

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХС

**Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ**

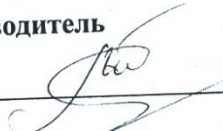
кафедра Информационных технологий

«Допущен к защите»
Заведующий кафедрой
Куралбаев З.К. д.ф.-м.н. профессор
Куз
« 21 » мая 2014 г.


ДИПЛОМНЫЙ ПРОЕКТ


На тему: «Разработка модуля отчетности в среде Oracle Business Intelligence Publisher
оптимизации нагрузки на АБИС»

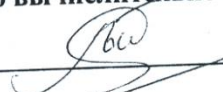
Специальность Вычислительная техника и программное обеспечение
Выполнила Рындина А. А. группа ВТу-11-1

Научный руководитель Шайхин Б.М. к.ф.-м.н., доцент
 « 14 » мая 2014 г.

Консультанты:
по экономической части: Еркешева З.Д. ст. преподаватель

 « 15 » апреля 2014 г.
по безопасности жизнедеятельности: Бегимбетова А.С. ст. преподаватель

 « 15 » апреля 2014 г.
по применению вычислительной техники: Шайхин Б.М. к.ф.-м.н., доцент

 « 14 » мая 2014 г.
Нормоконтролер: Тусупов Д.М. ассистент

 « 15 » мая 2014 г.

Рецензент: Мукимбеков М. Ж. д.т.н., доцент
« _____ » _____ 2014 г.

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Информационных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студентка Рындина Анастасия Андреевна

Тема проекта «Разработка модуля отчетности в среде Oracle Business Intelligence Publisher для оптимизации нагрузки на АБИС» утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «2» июня 2014 г.

Исходные данные к проекту, требуемые параметры результатов проектирования (исследования) и исходные данные объекта

- автоматизированная банковская информационная система;
- рабочая (фронтальная) система с собственной базой данных;
- таблицы фактов и измерений;
- сверточные данные;
- функциональные дизайны отчетов.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

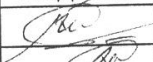

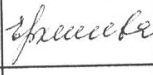

- рассмотреть теоретические вопросы, связанные с хранилищами данных и интегрированным комплексом аналитических инструментов – Oracle Business Intelligence;
- рассмотреть общие принципы организации процесса извлечения, преобразования и загрузки данных для хранилища данных;
- рассмотреть вопросы, непосредственно связанные с возможностями языка PL/SQL, созданием и применением пакетов, внешних процедур, курсоров и коллекций;
- рассмотреть возможности создания регламентированных отчетов в среде Oracle Business Intelligence;
- спроектировать шаблоны с помощью инструментов Oracle Business Intelligence и MS Office, сформировать запросы выгрузки данных их хранилища, рассмотреть способы доставки и представления результатов;
- провести обзор функциональности комплекса технологий – Oracle Business Intelligence;
- провести анализ предметных областей с помощью выгрузки в витрины данных.

Перечень графического материала (с точным указанием обязательных чертежей)

Рекомендуемая основная литература

- 1 Спирли Эрик. Корпоративные хранилища данных. Планирование, разработка, реализация. – М.: Вильямс, 2001. – т. 1.
- 2 Архипенков С.В., Голубев Д. К. Хранилища данных. От концепции внедрения. – М.: Диалог-МИФИ, 2002. – 528 с.
- 3 Darl Kuhn, Sam Alapati, Bill Padfield. Expert Indexing in Oracle Database 11g. Maximum Performance for your Database. – Publ.: Apress, 2011.
- 4 Архипенков С.Я. Аналитические системы на базе ORACLE Express OLAP. Проектирование, создание, сопровождение. – М.: Диалог–МИФИ, 2000. – 320 с.
- 5 Аносова Н.П., Бородин О.О. Распределенные базы и хранилища данных. Томск: ИНТУИТ, 2009.
- 6 Аруп Нанда, Стивен Фейерштейн. Oracle PL/SQL для администраторов баз данных. – СПб.: Символ-Плюс, 2012. – 69 с.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Общая часть	Шайхин Б.М.	28.03.2014	
Специальная часть	Шайхин Б.М.	08.04.2014	
Технико – экономическое обоснование проекта	Еркешева З.Д.	15.04.2014	
Безопасность жизнедеятельности, охрана труда и промышленная экология	Бегимбетова А.С.	15.04.2014	

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Потребность в анализе данных	26.03.2014	
2	Архитектура и компоненты хранилища данных	27.03.2014	
3	Основные схемы многомерной модели	01.04.2014	
4	Базовый процесс управления хранилищами данных – ETL	04.04.2014	
5	Централизованное хранилище данных с ETL.	07.04.2014	
6	Oracle Business Intelligence	09.04.2014	
7	Инструменты генерации запросов и отчетов.	11.04.2014	
8	Структура программы на PL/SQL.	14.04.2014	
9	Общая постановка задачи	15.04.2014	
10	Описание выгружаемых отчетов.	16.04.2014	
11	Связи фактов и измерений для выгружаемых отчетов	21.04.2014	
12	Требования и процесс создания пакета для отчета	22.04.2014	
13	Основные финансовые инструменты «Отчета о финансовых потоках и запасах».	23.04.2014	
14	Выгрузка коллекции в PL/SQL Developer.	24.04.2014	
15	Создание модели данных для отчетов	25.04.2014	
16	Создание отчета в среде Oracle Business Intelligence Publisher	28.04.2014	
17	Выгрузка отчета средствами Oracle Business Intelligence Publisher	29.04.2014	
18	Планирование запуска отчета	30.04.2014	
19	Витрины данных	05.05.2015	

Дата выдачи задания «18» сентября 2013 г.

Заведующий кафедрой _____

Куралбаев З.К.

Руководитель _____

Шайхин Б.М.

Задание принял к исполнению студент _____

Рындина А.А.

Аннотация

Дипломный проект посвящен разработке модуля оптимизации нагрузки автоматизированной банковской информационной системы (АБИС), для представления данных использовались графические средства – отчеты настраиваемые при помощи параметров в среде Oracle Business Intelligence Publisher, основанной на концепции хранилищ данных. Рассмотрены возможности, написания пакетов и курсоров в рамках базы данных Oracle Database 11g, расширяющие возможности работы с хранилищами данных, а также создание регламентированных отчетов и проектирования шаблонов. Сделан анализ характерных особенностей Oracle BI, в которые входит генерация логических запросов PL/SQL, также рассматриваются общие принципы организации процесса извлечения, преобразования и загрузки данных (Extract, Transform, Load) для ХД и интегрированный комплекс аналитических инструментов – Oracle Business Intelligence.

Андатпа

Диплом жобасы автоматтандырылған банктік ақпараттық жүйелерінің (АБАЖ) жүктемелерін оңтайландыру модулін әзірлеуге арналды, есеп деректер қоймасының тұжырымламасына негізделген Oracle Business Intelligence Publisher ортасында жасалды. Деректер қоймасымен жұмыс істеудің кеңейтілген мүмкіндіктері бар Oracle Database 11g деректер базасының шеңберінде пакеттер мен курсорларды жазу, сондай-ақ регламенті белгіленген есептер мен шаблондарды жобалау мүмкіндігі қаралды. PL/SQL логикалық сұраныстар генерациясына жататын Oracle BI ерекшеліктеріне талдау жасалынды, сондай-ақ ДҚ үшін деректерді табу, өзгерту және жүктеу (Extract, Transform, Load) процесін ұйымдастырудың жалпы принциптері мен Oracle Business Intelligence – аналитикалық аспаптардың интегралданған кешендері қарастырылды.

Annotation

Thesis project focuses on the development of automated optimization module load Banking Information System (ALIS), to represent data using graphic tools – reports using customizable parameters among Oracle Business Intelligence Publisher, based on the concept of data warehousing. The possibilities, writing packages and cursors in a database Oracle Database 11g, extend the capabilities of data warehousing, as well as the creation of regulated reports and design templates. The analysis of the characteristic features of Oracle BI, which include the generation of logical requests PL/SQL, also discusses the general principles of the process of extraction, transformation and loading (Extract, Transform, Load) for CD and integrated set of analytical tools – Oracle Business Intelligence.

Содержание

Аннотация	104
Введение.....	111
1 ОБЩАЯ ЧАСТЬ	114
1.1 Потребность в анализе данных	114
1.2 Хранилища данных	114
1.3 Архитектура и компоненты хранилища данных	115
1.4 Таблицы фактов и измерений в хранилищах данных	116
1.5 Основные схемы многомерной модели	118
1.6 Базовый процесс управления хранилищами данных – ETL	120
1.6.1 Извлечение данных в ETL	121
1.6.2 Преобразование данных	121
1.6.3 Загрузка данных	121
1.7 Централизованное хранилище данных с ETL.....	122
1.8 Расширенная модель ХД с витринами данных	124
1.9 Oracle Business Intelligence	126
1.10 Инструменты генерации запросов и отчетов	129
1.11 Структура программы на PL/SQL	130
1.12 Приведение типов	131
1.13 Коллекции	132
1.14 Объектно-ориентированное программирование в PL/SQL	134
1.15 Пакеты языка PL/SQL.....	137
1.16 Внешние процедуры	138
2 СПЕЦИАЛЬНАЯ ЧАСТЬ.....	140
2.1 Общая постановка задачи.....	140
2.2 Описание выгружаемых отчетов	141
2.2.1 Связи фактов и измерений для выгружаемых отчетов	141
2.2.2 Требования и процесс создания пакета для «Отчета о финансовых потоках и запасах»	150
2.2.3 Основные финансовые инструменты «Отчета о финансовых потоках и запасах»	152
2.2.4 Выгрузка коллекции в PL/SQL Developer	158

2.2.5	Требования и процесс создания пакета для «Отчета об остатках на балансовых счетах за вычетом резервов (провизий)»	160
2.2.6	Требования и процесс создания пакета для отчета «Основные показатели баланса»	165
2.3	Создание модели данных для отчетов	170
2.4	Создание отчета в среде Oracle Business Intelligence Publisher	175
2.5	Выгрузка отчета средствами Oracle Business Intelligence Publisher	177
2.6	Планирование запуска отчета	178
2.7	Витрины данных	181
3	ТЕХНИКО – ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА	184
3.1	Описание работы и обоснование необходимости	184
3.2	Трудовые ресурсы, используемые в работе	185
3.3	Оборудование и программное обеспечение, используемое в работе	185
3.4	Расчет стоимости разработки программного обеспечения	186
3.5	Сроки реализации проекта	187
3.5.1	Расчет затрат на оплату труда	188
3.5.2	Расчет затрат по социальному налогу	193
3.5.3	Расчет амортизационных отчислений	194
3.5.4	Расчет затрат на электроэнергию	195
3.5.5	Расчет затрат на накладные расходы	196
3.5.6	Расчет стоимости по всем статьям затрат	196
3.6	Цена интеллектуального труда	197
3.7	Вывод	198
4	БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ, ОХРАНА ТРУДА И ПРОМЫШЛЕННАЯ ЭКОЛОГИЯ	198
4.1	Описание работы и обоснование необходимости	198
4.2	Анализ условий труда в производственном помещении	199
4.3	Трудовые ресурсы и требования	200
4.4	Расчет освещенности на рабочем месте точечным методом	201
4.5	Вывод	207
	Заключение	208
	Перечень сокращений	210
	Список литературы	211
	Приложение А. Листинг созданных пакетов для отчетов	212
	Приложение Б. Выгружаемые отчеты из Oracle Business Intelligence	212

Введение

Хранилища данных – одна из самых актуальных и динамичных тем в современной индустрии информационных технологий.

Хранилище данных находится в центре всех ориентированных на приложения систем. Хранилище регулярно получает данные из этих систем и формирует сводное представление. Данные могут быть простой копией транзакционных данных (в этом случае их называют атомарными) или же подвергаться на пути от источника к пункту назначения (хранилищу) трансформации либо агрегированию. В данном диплом проекте хранилище данных работает с атомарными значениями. При этом в хранилище может помещаться только какое-то их подмножество, или же данные могут подвергаться конвертированию для того, чтобы обеспечить их совместимость с данными из других источников. Для обозначения процесса отсечения и извлечения данных обычно используются термины расслоение (slicing) и расщепление (dicing). Внутренняя структура хранилища данных построена так, чтобы запросы можно было легко создавать и эффективно выполнять.

Почти для всех успешно работающих приложений хранилищ данных используются выделенные серверы. Благодаря этим серверам запросы (типичные для хранилищ данных) влияют только на пользователей информационного сервиса и не замедляют критичные по времени операции.

Для множества людей хранилище данных – это некая совокупность данных объединенных из различных источников, структурированная и оптимизированная для доступа к ним при помощи средств создания запросов OLAP (on-line analytical processing – оперативной аналитической обработки). Для других хранилище данных – это фактически некая база данных, содержащая данные более чем из одного источника, собранные для целей управления информацией.

Понятие «хранение данных» возникло в середине 1980-х годов. И предназначалось для описания архитектурной модели потока данных от операционной системы к средствам поддержки принятия решений. Эта модель отвечает за различные задачи, ассоциированные с этим потоком и связанными с этим высокими затратами. Без такой архитектуры передаваемая управляющая информация обычно содержит большое количество избыточных данных. В больших корпорациях множественные проекты принятия решений обычно осуществляются независимо, каждый обслуживает различных пользователей, часто используя при этом те же самые данные. Процесс сбора, чистки и обобщения данных из различных, часто наследуемых, источников обычно дублировался для каждого проекта. Более того, существующие системы посещались повторно при каждом новом запросе, отличавшемся от предыдущего зачастую лишь оформлением данных.

Витрина данных обеспечивает идеальное решение, возможно, наиболее значительного конфликта при проектировании хранилищ данных – производительность против гибкости. Вообще, чем более упорядочена и гибка модель хранилища данных, тем ниже ее производительность при обработке запросов. Это обусловлено тем, что при запросе к нормализованной структуре обычно требуется гораздо больше действий по объединению таблиц, чем в случае оптимизированных структур. Направляя все запросы пользователя к витринам данных, и сохраняя при этом гибкую модель хранилища данных, проектировщики могут достичь гибкости и долговременной стабильности структуры хранилища данных, при оптимальной производительности обработки запросов пользователя.

Существует три основных подхода, дополняющих друг друга:

- статическая и динамическая отчетность. Наличие единого источника данных о бизнесе компании позволяет как с минимальными затратами строить статические отчеты, так и свободно создавать новые виды отчетов, предоставляющие новую точку зрения на бизнес компании. На рынке существует множество решений для построения отчетности от крупнейших мировых IT-производителей, полноценное применение которых возможно только при наличии полноценного хранилища данных. К подобным решениям относятся, например, Microsoft SQL Server: Reporting Services, Crystal Reports, Oracle Reports;

- интеллектуальный анализ данных (Data Mining и Business Intelligence). Наличие в хранилище всей совокупности данных об истории бизнеса компании позволяет аналитикам использовать современные средства интеллектуального анализа данных для поиска скрытых закономерностей в поведении клиентов/конкурентов. На рынке данных средств присутствует множество компаний, от лидеров IT-рынка с универсальными решениями, такими как Oracle BI или Microsoft SQL Server: Analysis Services, до небольших компаний со специализированными отраслевыми решениями;

- вычисление в реальном времени ключевых показателей эффективности (KPI). В современном бизнесе ситуация на рынке может меняться очень быстро. Своевременное отслеживание подобных изменений и адекватное реагирование на них является основой для выживания и роста компаний. Общепринятым средством отслеживания изменений, доступным для владельцев хранилищ данных, является создание набора считаемых в реальном времени ключевых показателей эффективности (KPI). Например, классическим KPI для банковской сферы может быть сумма средств, снятых с банкоматов некоторого района за последние сутки. Резкий скачок подобного показателя позволит своевременно отследить панику среди вкладчиков. Наличие полноценного хранилища данных позволяет компании реализовывать и поддерживать самые разнообразные KPI с минимальными трудовыми и временными затратами.

Специфика хранилища данных требует наличия у IT-специалистов, взявшихся за его реализацию, целого ряда специальных навыков и квалификаций.

Ниже перечислен список рисков, которые возникали при создании хранилища, и указаны навыки, необходимые для минимизации этих рисков:

- ориентация на имеющиеся технологии, а не на бизнес. Хранилище данных, чтобы быть успешным инструментом, должно отражать, прежде всего, бизнес-процессы заказчика. Это значит, что в разработке должны принимать самое непосредственное участие специалисты по бизнесу компании-заказчика. Попытка реализовать хранилище на основе расширения имеющейся IT-системы или сторонних разработок может привести к появлению еще одной из множества IT-систем, не решающей задач хранилища. Для минимизации данного риска исполнитель должен обладать опытом анализа бизнес-процессов, знанием предметной области и умением общаться со специалистами из других областей;

- ориентация на данные, а не на их использование. Хранилище данных должно не только содержать информацию, но и делать удобной ее анализ/использование. Для достижения этой цели при проектировании хранилища нужно понимать, для каких групп задач оно будет применяться. Для минимизации данного риска исполнитель должен обладать богатым опытом построения отчетности и технического анализа данных. Только в таком случае он сможет правильно предсказывать требования к хранилищу данных исходя из пожеланий его будущих пользователей;

- игнорирование сложности ETL процессов. Хранилище данных должно заполняться данными из разнообразных внешних источников, часто имеющих разную природу. Часть данных заказчика может храниться в реляционных базе данных (БД), часть – в Excel или же XML файлах. Данные разных систем могут дублировать друг друга или противоречить друг другу. При этом наполнение хранилища данным является не разовым процессом, а должно выполняться регулярно, для пополнения хранилища новыми данными. Данная комплексная задача может быть адекватно и надежно решена только за счет использования современных промышленных ETL-решений;

- отсутствие понимания используемой технологии хранения данных. Хранилище данных должно не просто хранить данные, оно должно успевать обновляться за заданные временные промежутки, а также удовлетворять требованиями заказчика по скорости генерации отчетов и выполнения запросов. Для соответствия данному требованию разработчик должен обладать богатым опытом реализации значительных баз данных на данной конкретной технологии.

1 ОБЩАЯ ЧАСТЬ

1.1 Потребность в анализе данных

Во всем мире в процессе своей деятельности организации накапливают или уже накопили очень большие объемы данных. Эти коллекции данных хранят в себе большие потенциальные возможности по извлечению новой, аналитической информации, на основе которой необходимо выявлять тенденции развития рынка, строить стратегию фирмы, находить новые решения, обуславливающие успешное развитие в условиях конкурентной борьбы. Для большинства фирм, у которых динамическое развитие такой анализ является неотъемлемой частью их повседневной деятельности, но большинство, только начинает приступать к нему всерьез.

Попытки строить системы принятия решений, которые обращались бы непосредственно к базам данных систем оперативной обработки транзакций, оказываются в большинстве случаев неудачными. Во-первых, аналитические запросы «конкурируют» с оперативными транзакциями, блокируя данные и вызывая нехватку ресурсов. Во-вторых, структура оперативных данных предназначена для эффективной поддержки коротких и частых транзакций и в силу этого слишком сложна для понимания конечными пользователями и, кроме того, не обеспечивает необходимой скорости выполнения аналитических запросов. В-третьих, в организации, как правило, функционирует несколько оперативных систем; каждая со своей базой данных. В этих базах используются различные структуры данных, единицы измерения, способы кодирования и т.д. Для конечного пользователя задача построения какого-либо сводного запроса по нескольким подобным базам данных практически неразрешима.

1.2 Хранилища данных

Для того чтобы обеспечить возможность анализа накопленных данных, организации стали создавать хранилища данных, которые представляют собой интегрированные коллекции данных, которые собраны из различных систем оперативного доступа к данным. Хранилища данных становятся основой для построения систем принятия решений. Несмотря на различия в подходах и реализациях, всем хранилищам данных свойственны следующие общие черты:

предметная ориентированность. Информация в хранилище данных организована в соответствии с основными аспектами деятельности предприятия (заказчики, продажи, склад и т.п.); это отличает хранилище

данных от БД, где данные организованы в соответствии с процессами (выписка счетов, отгрузка товара и т.п.). Предметная организация данных в хранилище способствует как значительному упрощению анализа, так и повышению скорости выполнения аналитических запросов. Выражается она, в частности, в использовании иных, чем в оперативных системах, схемах организации данных. В случае хранения данных в реляционной системы управления базами данных.(СУБД) применяется схема «звезды» (star) или «снежинки» (snowflake). Кроме того, данные могут храниться в специальной многомерной СУБД в n-мерных кубах;

- интегрированность. Исходные данные извлекаются из оперативных БД, проверяются, очищаются, приводятся к единому виду, в нужной степени агрегируются (то есть вычисляются суммарные показатели) и загружаются в хранилище. Такие интегрированные данные намного проще анализировать;

- привязка ко времени. Данные в хранилище всегда напрямую связаны с определенным периодом времени. Данные, выбранные их оперативных БД, накапливаются в хранилище в виде «исторических слоев», каждый из которых относится к конкретному периоду времени. Это позволяет анализировать тенденции в развитии бизнеса;

- неизменяемость. Попав в определенный «исторический слой» хранилища, данные уже никогда не будут изменены. Это также отличает хранилище от оперативной БД, в которой данные все время меняются, «дышат», и один и тот же запрос, выполненный дважды с интервалом в 10 минут, может дать разные результаты. Стабильность данных также облегчает их анализ.

1.3 Архитектура и компоненты хранилища данных

Англоязычный термин «Data Warehousing», который означает в широком смысле «создание, поддержку, управление и использование хранилища данных» и хорошо подтверждает тот факт, что речь идет о процессе. Цель этого процесса – непрерывная поставка необходимой информации нужным сотрудникам организации. Этот процесс подразумевает постоянное развитие, совершенствование, решение все новых задач и практически никогда не кончается, поэтому его нельзя уместить в более или менее четкие временные рамки, как это можно сделать для разработки традиционных систем оперативного доступа к данным.

Хранилища данных могут быть разбиты на два типа: корпоративные хранилища данных и киоски данных.

Корпоративные хранилища данных содержат информацию, относящуюся ко всей корпорации и собранную из множества оперативных источников для консолидированного анализа. Обычно такие хранилища охватывают целый ряд аспектов деятельности корпорации и используются для принятия как тактических, так и стратегических решений. Корпоративное хранилище содержит детальную и обобщающую

информацию; его объем может достигать от 50 Гбайт до одного или нескольких терабайт. Стоимость создания и поддержки корпоративных хранилищ может быть очень высокой. Обычно их созданием занимаются централизованные отделы информационных технологий, причем создаются они сверху вниз, то есть сначала проектируется общая схема, и только затем начинается заполнение данными. Такой процесс может занимать несколько лет.

Киоски данных содержат подмножество корпоративных данных и строятся для отделов или подразделений внутри организации. Киоски данных часто строятся силами самого отдела и охватывают конкретный аспект, интересующий сотрудников данного отдела. Киоск данных может получать данные из корпоративного хранилища или, что более распространено, данные могут поступать непосредственно из оперативных источников.

Киоски и хранилища данных строятся по сходным принципам и используют практически одни и те же технологии.

Основными компонентами хранилища данных являются следующие:

- оперативные источники данных;
- средства проектирования/разработки;
- средства переноса и трансформации данных;
- СУБД;
- средства доступа и анализа данных;
- средства администрирования ^[1].

1.4 Таблицы фактов и измерений в хранилищах данных

Таблица фактов – является основной таблицей хранилища данных. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться.

Обычно говорят о четырех наиболее часто встречающихся типах фактов. К ним относятся:

– факты, связанные с транзакциями (англ. Transaction facts). Они основаны на отдельных событиях (типичными примерами которых являются телефонный звонок или снятие денег со счета с помощью банкомата);

– факты, связанные с «моментальными снимками» (англ. Snapshot facts). Основаны на состоянии объекта (например, банковского счета) в определенные моменты времени, например на конец дня или месяца. Типичными примерами таких фактов являются объем продаж за день или дневная выручка;

– факты, связанные с элементами документа (англ. Line-item facts). Основаны на том или ином документе (например, счете за товар или услуги)

^[1] Архипенков С.В., Голубев Д. К. Хранилища данных. От концепции до внедрения. – М.: Диалог-МИФИ, 2002. – 528 с.

и содержат подробную информацию об элементах этого документа (например, количестве, цене, проценте скидки);

– факты, связанные с событиями или состоянием объекта (англ. Event or state facts). Представляют возникновение события без подробностей о нем.

Таблица фактов, как правило, содержит уникальный составной ключ, объединяющий первичные ключи таблиц измерений. Чаще всего это целочисленные значения либо значения типа «дата/время» в целочисленном формате – ведь таблица фактов может содержать сотни тысяч или даже миллионы записей, и хранить в ней повторяющиеся текстовые описания, как правило, невыгодно – лучше поместить их в меньшие по объему таблицы измерений. При этом как ключевые, так и некоторые не ключевые поля должны соответствовать будущим измерениям OLAP-куба. Помимо этого таблица фактов содержит одно или несколько числовых полей, на основании которых в дальнейшем будут получены агрегатные данные.

Для многомерного анализа пригодны таблицы фактов, содержащие как можно более подробные данные (то есть соответствующие членам нижних уровней иерархии соответствующих измерений). Бывает предпочтительнее взять за основу факты продажи товаров отдельным заказчикам, а не суммы продаж для разных стран – последние все равно будут вычислены OLAP-средством, в случае использования такового. Исключение можно сделать, пожалуй, только для клиентских OLAP-средств, поскольку в силу ряда ограничений они не могут манипулировать большими объемами данных.

Необходимо отметить, что в таблице фактов нет никаких сведений о том, как группировать записи при вычислении агрегатных данных. Например, в ней есть идентификаторы продуктов или клиентов, но отсутствует информация о том, к какой категории относится данный продукт или в каком городе находится данный клиент. Эти сведения, в дальнейшем используемые для построения иерархий в измерениях куба, содержатся в таблицах измерений. В случае построения отчетов напрямую из хранилища данных, минуя промежуточный шаг создания OLAP-кубов, также могут использоваться т.н. агрегатные таблицы фактов, содержащие более крупнозернистую информацию, например суммарные траты покупателя в выбранном магазине за месяц, вместо или в дополнение к детальной таблице фактов с подробной.

Таблица измерений – таблица в структуре многомерной базы данных, которая содержит атрибуты событий, сохраненных в таблице фактов. Атрибуты представляют собой текстовые или иные описания, логически объединенные в одно целое. Например, имя покупателя может являться атрибутом в таблице измерений покупателей, а наименование товара, в таблице измерений товаров. В то время как сумма транзакции является величиной аддитивной, и ее значение должно храниться в таблице фактов.

Таблица фактов связана с таблицами измерений с помощью внешнего ключа.

1.5 Основные схемы многомерной модели

Существуют несколько схем для многомерного моделирования данных. Две из них считаются основными: схема «звезда» и схема «снежинка». В более сложных случаях используются так называемые «многозвездочные» схемы или схема с несколькими таблицами фактов.

Схема «звезда» имеет одну таблицу фактов и несколько таблиц измерений. Схема «снежинка» имеет одну таблицу фактов и несколько нормализованных таблиц измерений.

Схема «звезда» имеет те же самые элементы, что диаграммы «сущность-связь». Это – сущности, атрибуты, первичные и внешние ключи, взаимосвязи, кардинальность связи. На рисунке 1.1 приведен пример схемы «звезда», созданной для учета продажи бакалейных товаров. Таблица фактов «Продажи бакалеи» (учет операций продажи бакалейных товаров торговой компании) имеет один первичный ключ «Номер счета», четыре внешних ключа (по числу измерений) и два параметра: «Количество» и «Стоимость товара».

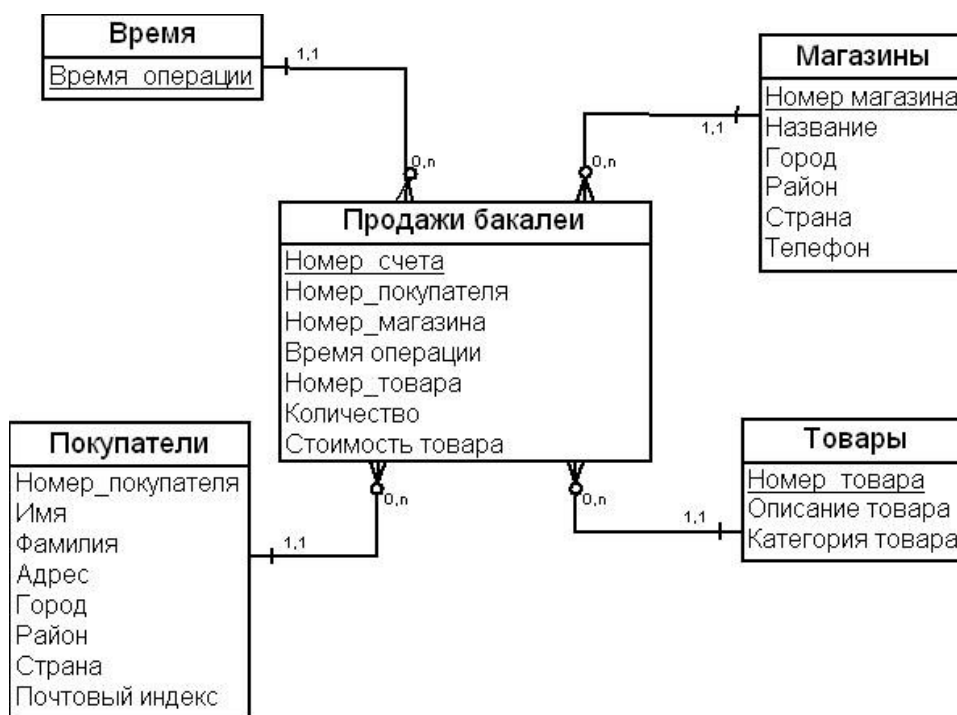


Рисунок 1.1 - Схема «звезда»

Измерение «Время» является одним из критических элементов модели хранилища данных (ХД). Если данные в OLTP-системах запросы фокусируются на текущем моменте времени, то в системах поддержки принятия решений, для которых проектируются и создаются ХД, запросы фокусируются на задачах анализа данных, а именно – как данные изменялись в различные периоды времени. Например, каков был объем продаж торговой

компании за последний квартал, месяц или тенденции в покупках в течение последнего квартала.

Измерение «Магазин» позволяет сгруппировать операции продаж по магазинам с учетом их географического положения.

Измерение «Товары» позволяет анализировать типовые схемы закупок товаров и отвечать на вопрос, какие товары, как правило, покупаются одновременно покупателями.

Измерение «Покупатели» позволяет анализировать покупки с учетом их частоты, географического положения и количества.

Таблицы измерений являются своеобразным путеводителем при выборке строк из таблицы фактов. Таблицы фактов имеют большое количество строк. Таблицы измерений, как правило, имеют гораздо меньшее число строк. Одним из главных преимуществ использования такой схемы является то, что производительность операции СУБД соединения таблицы фактов и таких таблиц соединений будет близка к оптимальной.

Схема «снежинка» (рисунок 1.2) добавляет иерархию в таблицы измерений. Например, измерение «Регион» группирует магазины по географическим регионам, измерение «Категория товара» группирует товары по категориям, измерение «Категория покупателей» группирует покупателей по категориям, а измерение «Период продаж» группирует продажи по периодам времени.

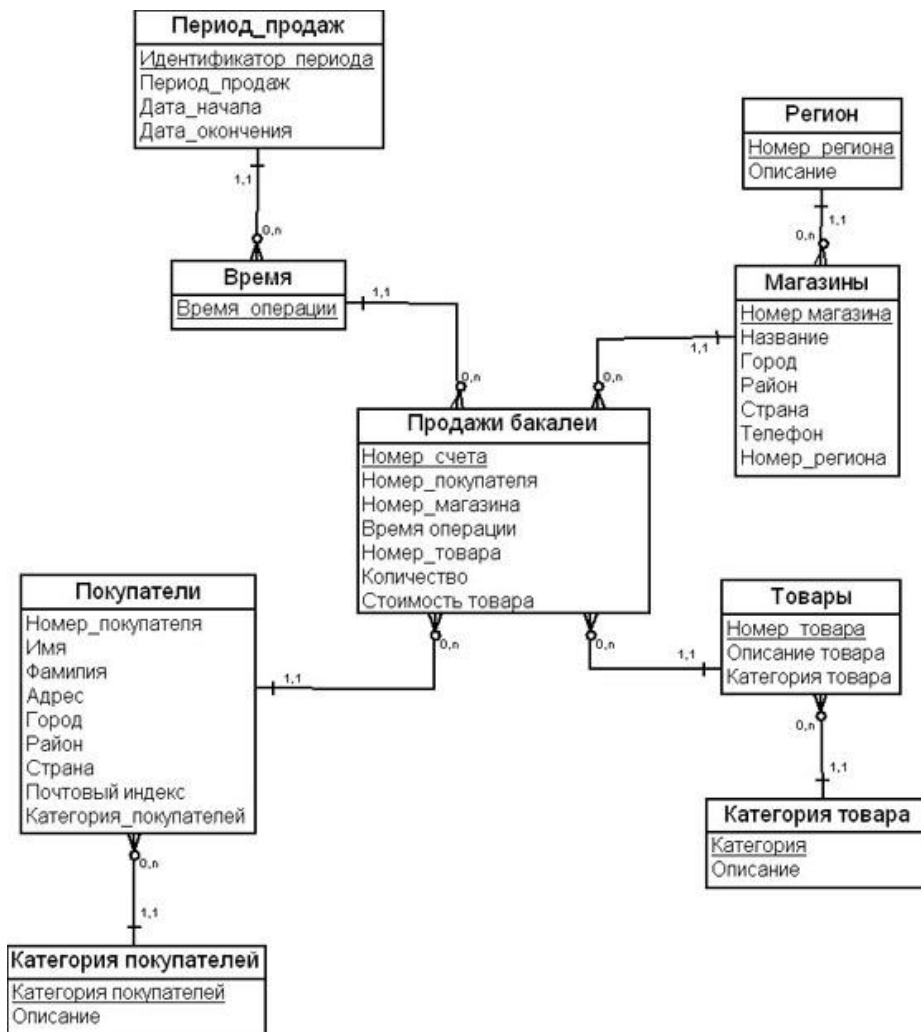


Рисунок 1.2 - Схема «снежинка»

Таким образом, использование иерархий превращает схему «звезда» в схему «снежинка». Можно расширять схемы «снежинка» за счет добавления в нее новых иерархий [2].

1.6 Базовый процесс управления хранилищами данных – ETL

ETL (от англ. Extract, Transform, Load – извлечение, преобразование, загрузка) – один из базовых процессов управления ХД, а также наименование класса утилит автоматизации этого процесса. ETL в узком смысле относится к технологиям консолидации данных, но современные решения, представленные на рынке, поддерживают помимо реализации, федерализации данных и консолидации, а также обмена данными.

Цель ETL-приложения состоит в том, чтобы своевременно предоставить необходимые данные его пользователям. Традиционно предприятия использовали программы ETL для передачи операционных

[2] Спирли Эрик. Корпоративные хранилища данных. Планирование, разработка, реализация. – М.: Изд.: Вильямс, 2001. – т. 1.

данных в системы бизнес-интеллекта, такие как ХД или киоски данных или переноса информации из унаследованных приложений в новые.

В целом приложения ETL извлекают информацию из исходной базы данных, преобразуют ее в формат, поддерживаемый базой данных назначения, а затем загружают в нее преобразованную информацию.

1.6.1 Извлечение данных в ETL

Начальным этапом процесса ETL является процедура извлечения записи из источников данных и подготовка их к процессу преобразования. При разработке процедуры извлечения данных, в первую очередь необходимо определить частоту выгрузки данных из OLTP-систем (Online Transaction Processing), или отдельных источников. В зависимости от объема извлекаемых данных, сложности доступа к ним и скорости работы оборудования выгрузка данных занимает определенное время, которое называется окном выгрузки.

Процедуру извлечения можно реализовать двумя способами:

- извлечение данных с помощью программных средств;
- извлечение данных средствами той системы, в которой они хранятся.

После извлечения данные помещаются в так называемую «промежуточную область», где для каждого источника данных создается своя таблица или отдельный файл, или и то и другое.

1.6.2 Преобразование данных

Цель этапа преобразования данных – подготовка данных к размещению в хранилище данных и приведение их к виду более удобному для последующего анализа. При этом должны учитываться некоторые, выдвигаемые аналитиком требования, в частности уровню качества данных.

Поэтому в процессе преобразования может быть задействован самый разнообразный инструментарий, начиная с простейших средств ручного редактирования данных до систем, реализующих сложные методы обработки и очистки данных. В процессе преобразования данных, в рамках ETL, чаще всего выполняются следующие операции:

- преобразование структуры данных;
- агрегирование данных;
- перевод значений;
- создание новых данных;
- очистка данных.

1.6.3 Загрузка данных

Процесс загрузки заключается в переносе данных из промежуточных таблиц в структуру хранилища данных. При очередной загрузке в хранилище данных переносится не вся информация из источников, а только та, которая была изменена в течение промежуточного времени, прошедшего с предыдущей загрузки. При этом выделяют два потока:

- поток добавления – в хранилище данных передается новая, ранее не существовавшая информация;

- поток обновления (добавления) – в хранилище данных передается информация, которая существовала ранее, но была изменена или дополнена.

Для распределения загружаемых данных на потоке используются средства данных. Они фиксируют состояние данных в некоторые моменты времени и определяют, какие данные были изменены или дополнены.

1.7 Централизованное хранилище данных с ETL

Виртуальные хранилища данных и независимые витрины показали, что для эффективной работы аналитических систем необходим единый репозиторий данных. Для наполнения этого репозитория необходимо извлечь, согласовать разнородные данные из различных источников и загрузить эти данные в репозиторий.

Средства извлечения, преобразования и загрузки данных (ETL) должны знать все об источниках данных: структуры хранящихся данных и их форматы, различия в алгоритмах обработки данных, смысл хранящихся данных, график выполнения обработки информации в транзакционных системах. Игнорирование этих данных о данных (метаданных) неизбежно приводит к ухудшению качества информации, загружаемой в хранилище. В результате пользователи теряют доверие к ХД, стараются получать информацию напрямую из источников, что приводит к неоправданным временным затратам специалистов, эксплуатирующих системы – источники данных.

Таким образом, информация об источниках данных должна использоваться средствами ETL. Поэтому средства ETL должны работать в тесной связке со средствами ведения метаданных.

При обработке извлеченных данных необходимо преобразовать их к единому виду. Поскольку основные данные хранятся в реляционных базах данных, нужно учесть различие в кодировке данных. Даты могут кодироваться в разных форматах; адреса могут использовать различные сокращения; кодировка продуктов может следовать различным номенклатурам. Первоначально информация о нормативно справочной информации (НСИ) заносилась в алгоритмы преобразования данных ETL. По мере роста числа источников данных объема обрабатываемых данных (он может достигать терабайтов в сутки), стало ясно, что необходимо отделить средства управления НСИ от средств ETL, и обеспечить их эффективное взаимодействие.

Средства ETL извлекают данные из источников, во взаимодействии со средствами ведения метаданных и НСИ преобразуют их к требуемым форматам и загружают в репозиторий данных. В качестве репозитория чаще всего выступает репозиторий хранилища данных, но также может быть и оперативный склад данных, и зоны временного хранения, и даже витрины данных. Поэтому одним из ключевых требований к средствам ETL является их способность взаимодействовать с различными системами (рисунок 1.3).



Рисунок 1.3 - Централизованное хранилище данных с ETL

Необходимость повышения оперативности предоставляемой аналитической информации и рост объемов обрабатываемых данных выставляют повышенные требования к производительности средств ETL и их масштабируемости. Поэтому средства ETL должны использовать различные схемы параллельных вычислений и уметь работать на высокопроизводительных системах различных архитектур.

Как видно, к средствам ETL предъявляются самые разные требования:

- необходимо собрать данные от разных систем – источников, даже если одна или несколько систем в результате сбоя не смогли в срок завершить свою работу и предоставить необходимые данные;
- полученная информация должна быть распознана и преобразована в соответствии с алгоритмами преобразования, а также с помощью систем ведения НСИ и метаданных;
- преобразованная информация должна быть загружена в зоны временного хранения, в хранилище данных, в оперативный склад данных (ОСД), в витрины данных, как того требует производственный процесс;

- средства ETL должны иметь высокую пропускную способность с тем, чтобы собирать и выгружать все возрастающие объемы данных;
- средства ETL должны обладать высокими вычислительными возможностями и масштабируемостью для сокращения времени обработки данных для уменьшения задержек в предоставлении данных;
- средства ETL должны предоставлять разнообразные инструменты извлечения данных в различных режимах работы – от пакетного сбора для систем, некритичных к временным задержкам, до инкрементальной обработки в режиме, близком к реальному времени.

В связи с этими, зачастую взаимоисключающими требованиями, проектирование и разработка средств ETL превращается в сложную задачу даже тогда, когда используются решения, предлагаемые на рынке.

1.8 Расширенная модель ХД с витринами данных

Прямая работа пользовательских программ с корпоративным хранилищем данных (КХД), допустима, если пользовательские запросы не препятствуют нормальному функционированию КХД, если между пользователями и КХД имеются высокоскоростные линии связи, или если случайный доступ ко всем данным не ведет к серьезным потерям. Администрирование прямого доступа пользователей к КХД представляет собой чрезвычайно сложную задачу. Например, пользователь одного подразделения имеет право доступа к данным другого подразделения только через 10 дней после получения этих данных. Или пользователь может видеть только агрегированные показатели, но не детальные данные. Существуют и другие, еще более запутанные правила доступа. Их ведение, учет и изменение приводит к неизбежным ошибкам, вызванным сочетанием сложных условий доступа.

Витрины данных, содержащие информацию, предназначенную для выделенной группы пользователей, значительно снижают риски нарушения требования информационной безопасности.

До сих пор серьезной проблемой для территориально распределенных организаций является качество линий связи. В случае обрыва или недостаточной пропускной способности удаленные пользователи лишаются доступа к информации, содержащейся в КХД. Решением являются удаленные витрины данных, которые заполняются либо в нерабочее время, либо инкрементально, по мере поступления информации, с использованием транспорта с гарантированной доставкой.

Разные пользовательские приложения нуждаются в различных форматах данных: многомерные кубы, ряды данных, двумерные массивы, реляционные таблицы, файлы в формате MS Excel, текстовые файлы с разделителями, XML-файлы и т.д. Никакая структура данных в КХД не может удовлетворить этим требованиям. Выходом является создание витрин,

чи структуры данных оптимизированы под специфические требования отдельных приложений (рисунок 1.4).



Рисунок 1.4 - Расширенная модель с витринами данных

Еще одной причиной необходимости создания витрин данных является требование к надежности КХД, которое часто определяется, как пять или четыре девятки. Это означает, что КХД может простаивать не более 5 минут в год или не более часа в год. Создание комплекса с такими характеристиками является сложной и весьма недешевой инженерной задачей. Требования к защите от терактов и стихийных бедствий еще более усложняют построение программно-технического комплекса и осуществление соответствующих организационных мероприятий. Чем сложнее такой комплекс, чем больше данных он хранит, тем выше его стоимость и сложнее его поддержка. Наличие витрин данных резко снижает нагрузку на КХД, как по количеству пользователей, так и по объему данных в хранилище, так как эти данные могут быть оптимизированы под хранение, а не под обслуживание запросов.

Если витрины наполняются напрямую из КХД, то фактическое количество пользователей снижается с сотен и тысяч до десятков витрин, которые и являются пользователями КХД. При использовании средств SRD (Sample, Restructure, Delivery – выборка, реструктуризация, доставка) количество пользователей сокращается до 1. В этом случае вся логика информационного снабжения витрин сосредотачивается в SRD. Витрины могут быть оптимизированы под обслуживание пользовательских запросов. Программно-технический комплекс КХД может быть оптимизирован исключительно под надежное, защищенное хранение данных.

Средства SRD также смягчают нагрузку на КХД за счет того, что разные витрины могут обращаться к одним и тем же данным, тогда как SRD

извлекает данные один раз, преобразует к различным форматам и доставляет в разные витрины данных.

1.9 Oracle Business Intelligence

Семейство решений Oracle Business Intelligence – это интегрированный комплекс аналитических инструментов, который разработан с целью обеспечить лучшее видение и понимание бизнеса широкому кругу пользователей и позволяет любому пользователю организации получить быстрый Web-доступ к актуальной информации.

Oracle Business Intelligence Suite объединяет несколько продуктов, которые могут использоваться как вместе, так и независимо друг от друга:

- Oracle BI Server – масштабируемый, высокопроизводительный сервер запросов и анализа, интегрирующий данные из множества неструктурированных, OLAP и готовых приложений-источников, разработанных как Oracle, так и другими производителями;

- Oracle BI Answers – мощный инструмент для выполнения произвольных запросов в Web-интерфейсе. При этом, пользователи работают с логическим представлением информации из различных источников данных;

- Oracle BI Interactive Dashboard – интерактивные информационные панели с широкими функциональными возможностями, построенные в Web-архитектуре и отображающие персонализированную информацию, которая помогает пользователям принимать точные и эффективные решения;

- Oracle BI Publisher – масштабируемый сервер формирования отчетов, позволяющий генерировать отчеты в разных форматах на основе данных из множества источников и рассылать их по различным каналам;

- Oracle BI Disconnected Analytics – решение, позволяющее пользователям получить доступ к возможностям Answers и Dashboards при работе на компьютерах в режиме офлайн;

- Oracle BI Office Plug-In – инструмент, позволяющий работать с аналитическим сервером при помощи Microsoft Word, Excel и Powerpoint;

- Oracle Mobile Application Designer - решение, позволяющее создавать отчеты, адаптированные для мобильных устройств на базе iOS, Android и т.д. Приложения создаются с помощью drag&drop интерфейса, не требуют знания языка SQL, могут использовать уже имеющиеся отчёты.

- Oracle Business Intelligence (OBIEE) (рисунок 1.5) поддерживает множество различных источников данных, среди которых реляционные базы, системы OLAP и файлы. Система объединяет в одном отчете данные из различных источников, делая это согласно заданным правилам.

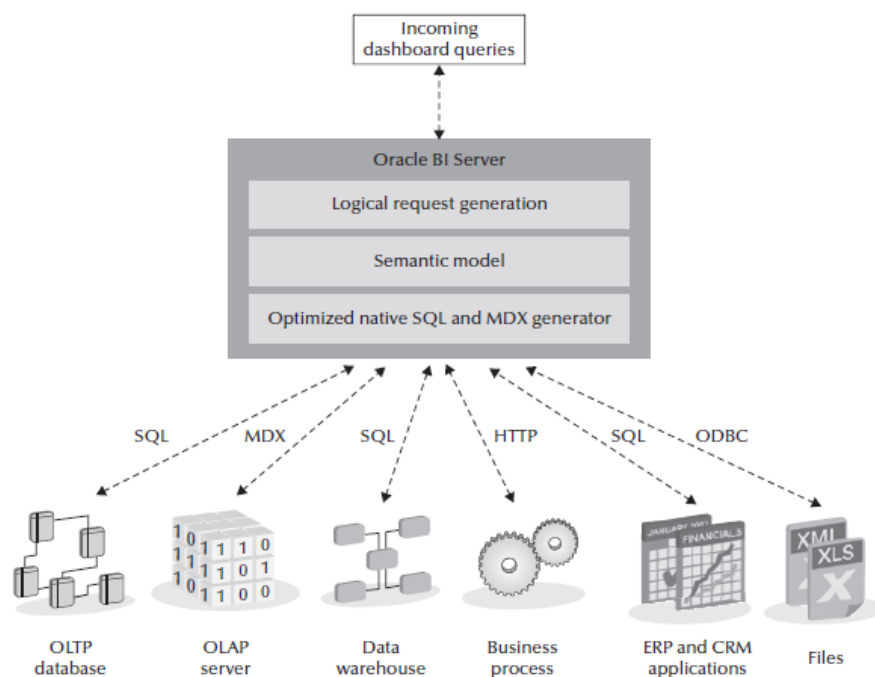


Рисунок 1.5 – Oracle Business Intelligence Suite

Архитектура Oracle Business Intelligence

В центре системы находится единая модель данных (репозиториум) – описание логической модели бизнес-области и привязку ее к физическим источникам данных. Модель состоит из трех слоев (рисунок 1.6) – презентационного, бизнес и физического. В бизнес-слое описывается логическая модель (которая понятна конечным пользователям), в форме многомерной модели – фактов и измерений, а также описывается привязка логических атрибутов к физическим источникам. Презентационный позволяет разбить логическую модель на несколько предметных областей и ограничить доступ пользователей к различным показателям и атрибутам. Физический слой содержит описание источников данных – таблиц, полей, ключей, кубов данных. Создание репозитория выполняет разработчик с помощью специальной программы – Oracle BI Administration Tool.

Когда пользователь открывает отчет, сервер презентаций генерирует запрос на языке Logical SQL к серверу BI. Сервер BI разбирает запрос, и переводит его в запросы к источникам данных на их «родных» языках – sql, mdx и т.п. После получения данных от источников сервер объединяет их, проводит различные действия над данными, и возвращает результат серверу презентаций. Сервер презентаций, в свою очередь, отрисовывает полученные данные в веб-интерфейсе или генерирует статичный отчет.

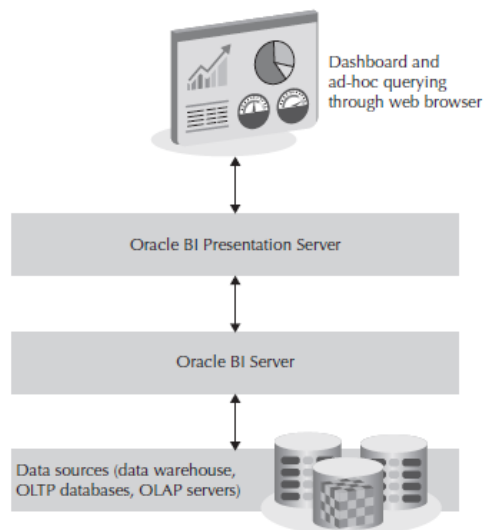


Рисунок 1.6 – Схема репозитория

Серверная часть ОВІЕЕ (рисунок 1.7) включает в себя несколько разрозненных компонентов, часть из которых управляется сервером приложений Weblogic, а другая часть существует как обычные программы и управляется компонентом oracle process manager. Также ОВІЕЕ использует для хранения части служебных данных реляционную базу [3].

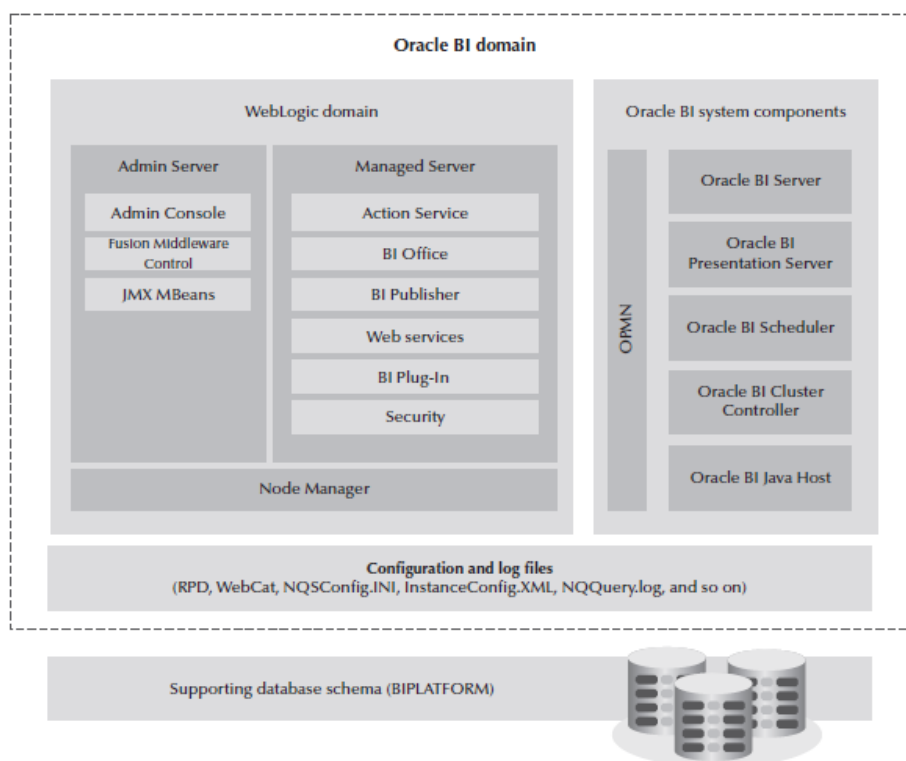


Рисунок 1.7 – Схема управление доменом ОВІЕЕ

[3] Сайт http://www.tadviser.ru/index.php/Продукт:Oracle_Business_Intelligence

1.10 Инструменты генерации запросов и отчетов

Основные возможности BI-системы развиваются по четырем основным направлениям: хранение данных, интеграция данных, анализ данных и представление данных.

Данные, используемые для бизнес-анализа, организуются в специальные хранилища (data warehouse, DW). Эти данные должны отражать текущую, реальную и полную картину бизнеса. Информация в хранилище данных (включая исторические данные) собирается из различных операционных (транзакционных) систем и структурируется специальным образом для более эффективного анализа и обработки запросов (в отличие от обычных баз данных, где информация организована таким образом, чтобы оптимизировать время обработки текущих транзакций).

Хранилища данных содержат огромные объемы информации, охватывающей все доступные стороны деятельности предприятия и позволяющие рассматривать все аспекты функционирования бизнеса в совокупности. Для решения более узких, конкретных задач из общего хранилища могут вычленяться подмножества данных – витрины данных.

Для всестороннего анализа данных в современных BI используются OLAP-инструменты (online analytical processing). Они позволяют рассматривать различные срезы данных, в том числе временные, позволяющие выявлять различные тренды и зависимости (по регионам, продуктам, клиентам и т.п.).

Для представления данных используются различные графические средства – отчеты и графики настраиваемые при помощи параметров.

Главная часть BI-инструментов делится на корпоративные BI-наборы и BI-платформы. Средства генерации запросов и отчетов в большой степени поглощаются и замещаются корпоративными BI-наборами. Многомерные OLAP-механизмы или серверы, а также реляционные OLAP-механизмы являются BI-инструментами и инфраструктурой для BI-платформ.

Большинство BI-инструментов применяются конечными пользователями для доступа, анализа и генерации отчетов по данным, которые чаще всего располагаются в ХД или ОСД.

Генераторы запросов и отчетов – типично «настольные» инструменты, предоставляющие пользователям доступ к базам данных, выполняющие некоторый анализ и формирующие отчеты. Запросы могут быть как незапланированными, так и иметь регламентный характер. Имеются системы генерации отчетов (серверные), которые поддерживают регламентные запросы и отчеты.

Настольные генераторы запросов и отчетов расширены также некоторыми облегченными возможностями OLAP. Развитые инструменты этой категории объединяют в себе возможности пакетной генерации регламентных отчетов и настольных генераторов запросов, рассылки отчетов

и их оперативного обновления, образуя так называемую корпоративную отчетность.

1.11 Структура программы на PL/SQL

PL/SQL – это процедурный блочно-структурированный язык. Он представляет собой расширение языка SQL и предназначен для работы с СУБД Oracle.

PL/SQL встроен в следующие СУБД: Oracle Database и IBM DB2. Также PL/SQL используется как встроенный язык для средства быстрой разработки Oracle Forms и инструмента разработки отчетов Oracle Reports.

PL/SQL предоставляет разработчику приложений и интерактивному пользователю следующие основные возможности:

- реализация подпрограмм как отдельных блоков, в том числе использование вложенных блоков;
- создание пакетов, процедур и функций, хранимых в базе данных;
- предоставление интерфейса для вызова внешних процедур;
- поддержка как типов данных SQL, так и типов, вводимых в PL/SQL;
- применение явного и неявного курсора, а также оператора цикла FOR для курсора;
- введение у переменных PL/SQL и курсоров атрибутов, которые позволяют ссылаться на тип данных или структуру элемента;
- введение типов коллекций и объектных типов;
- поддержка набора операторов управления и операторов цикла;
- реализация механизма обработки исключений.

Основной программной единицей PL/SQL является блок, который может содержать вложенные блоки, называемые иногда подблоками.

Блок позволяет объединять объявления и операторы, связанные общей логикой; может быть анонимным и именованным.

Блок состоит из трех основных частей:

- секция объявлений (необязательная часть);
- тело блока;
- обработчики исключений (необязательная часть).

```
[ <<label_name>> ]  
[DECLARE  
]  
BEGIN  
[EXCEPTION  
]  
END [label_name];
```

PL/SQL не чувствителен к регистру, кроме строковых переменных и констант. Каждая конструкция PL/SQL должна заканчиваться символом «;». Одна конструкция может быть расположена на нескольких строках ^[4].

1.12 Приведение типов

PL/SQL поддерживает явное и неявное приведение типов: явное приведение типов выполняется с помощью встроенных функций, а неявное - посредством PL/SQL при присвоении значения одного типа.

Значения переменных различных типов могут присваиваться друг другу в том случае, если они образованы из одного базового типа. Переменные могут иметь тип данных SQL или тип данных PL/SQL. Переменная объявляется в секциях объявлений блока PL/SQL, подпрограммы или пакета. Для объявления переменной после ее идентификатора следует указывать любой доступный тип данных.

Объявление переменной в PL/SQL может иметь следующие формы:

```
var_name type;  
var_name type := expr;  
var_name type DEFAULT expr;  
var_name type NOT NULL := expr;  
var_name type_var%TYPE;  
var_name type_var%TYPE := expr;  
var_name user.table.type_col%TYPE;  
var_name user.table.type_col%TYPE := expr;
```

Одновременно, при объявлении переменной, она может быть проинициализирована значением соответствующего типа. Выражение, находящееся справа от знака присваивания, может использовать ранее объявленные и проинициализированные переменные. PL/SQL требует, чтобы используемая ссылка была описана в программе выше места ее применения.

При объявлении переменной вместо оператора присваивания может указываться ключевое слово DEFAULT.

Объявляемая переменная может быть определена как NOT NULL. Такой переменной в дальнейшем нельзя присвоить значение NULL.

Переменным можно присваивать значения двумя способом – как с помощью оператора присваивания, так и как INTO-переменной.

При объявлении константы после идентификатора должно быть указано ключевое слово CONSTANT, а после идентификатора типа - указан оператор присваивания и значение константы.

Объявление константы может иметь следующее формальное описание:

```
const_name CONSTANT type :=value;
```

^[4] Сайт <http://datasql.ru/sqlandpol/11.htm>

Атрибут `%TYPE` позволяет объявлять переменную типа, соответствующего:

- типу другой переменной;
- типу столбца базы данных.

Атрибут `%ROWTYPE` позволяет объявлять переменную типа «запись», соответствующую строке таблицы. Переменная такого типа имеет поля, совпадающие с полями таблицы по имени и типу.

Такой тип значительно облегчает программирование операций со строками, позволяя выполнять выборку строки целиком в одну переменную типа «запись», а также предотвращает необходимость перепрограммирования блоков в случае изменения структуры таблицы.

Значения переменным, определенным с использованием атрибута `%ROWTYPE`, могут быть назначены как присваиванием значения одной записи другой записи, так и как `INTO`-переменным оператором `SELECT`.

```
DECLARE
    tbl1_rec1 tbl1%ROWTYPE;
    -- Для строки из таблицы tbl1
    tbl1_rec2 tbl1%ROWTYPE;
    CURSOR c1 IS SELECT * FROM tbl1;
    emp_rec2 c1%ROWTYPE;
    -- Для строки курсора c1,
    -- созданного из таблицы tbl1
    emp_rec3 c1%ROWTYPE;
BEGIN
    SELECT * INTO tbl1_rec1 FROM tbl1
        WHERE tbl1.f1=1;
    emp_rec2 := emp_rec1;
END
```

Переменная типа «запись» без квалификации именами полей может использоваться только для выборки значений, но ее нельзя применять для вставки или обновления значений строки ^[5].

1.13 Коллекции

Коллекцией называется упорядоченная группа элементов одного типа. Язык PL/SQL поддерживает три вида коллекций:

- вложенные таблицы (nested tables);
- индексированные таблицы;
- `varray`-массивы (variable-size arrays).

Доступ к элементу вложенной таблицы или массива осуществляется по его индексу, который указывается в скобках после имени переменной типа коллекции. Коллекция может быть передана в качестве параметра.

Коллекцию можно использовать:

^[5] Сайт <http://datasql.ru/sqlandpol/11.htm>

- для обмена с таблицами баз данных и столбцами данных;
- для передачи столбца данных из приложения клиента в хранимую процедуру или обратно.

Для создания коллекции следует определить тип коллекции – TABLE или VARRAY – и объявить переменную этого типа. Определение типа выполняется в секции объявлений блока PL/SQL, подпрограммы или пакета.

Определение типа вложенной таблицы может иметь следующее формальное описание:

```
TYPE type_name IS TABLE OF
element_type [NOT NULL];
```

Параметр type_name указывает имя определяемого типа, а element_type – это любой допустимый тип данных PL/SQL, исключая некоторые типы, в том числе VARRAY, TABLE, BOOLEAN, LONG, REF CURSOR и т.п.

Вложенную таблицу можно рассматривать как одномерный массив, в котором индексами служат значения целочисленного типа в диапазоне от 1 до 2147483647. Вложенная таблица может иметь пустые элементы, которые появляются после их удаления встроенной процедурой DELETE. Вложенная таблица может динамически увеличиваться.

Индексированные таблицы позволяют работать со столбцами как с единой переменной – массивом.

Определение индексированной таблицы (index-by tables) может иметь следующее формальное описание:

```
TYPE type_name IS TABLE
OF element_type [NOT NULL]
INDEX BY BINARY_INTEGER;
```

Индексированная таблица – это вариант вложенной таблицы, в которой элементы могут иметь произвольные целочисленные значения индексов. Такой тип данных очень удобен, если в качестве индекса использовать значение первичного ключа.

Определение типа Varray-массива может иметь следующее описание:

```
TYPE type_name IS
{VARRAY | VARYING ARRAY} (size_limit)
OF element_type [NOT NULL];
```

Параметр type_name указывает имя определяемого типа, size_limit – максимальное количество элементов, а element_type – это любой допустимый тип данных PL/SQL, исключая некоторые типы, такие как VARRAY, TABLE, BOOLEAN, LONG, REF CURSOR и т.п.

Максимальное количество элементов в Varray-массиве указывается при определении типа и не может изменяться динамически. Доступ к каждому

элементу Varray-массива осуществляется по индексу. Varray-массивы можно передавать в качестве параметров. Varray-массивы не могут иметь пустот, так как для них нет операции удаления произвольного элемента массива.

Для инициализации коллекции используется конструктор - автоматически создаваемая функция, одноименная с типом коллекции.

Конструктор создает коллекцию из значений переданного ему списка параметров. Конструктор может быть вызван как в секции объявлений через знак присваивания после указания типа, так и в теле программы. Вызов конструктора без параметров означает инициализацию коллекции как пустой.

```
DECLARE
CREATE TYPE rec_var1
AS VARRAY(3) OF num;
CREATE TYPE rec_var2
AS VARRAY(3) OF rec_obj;
  r1 rec_var1; r2 rec_var2;
BEGIN
  /* Инициализация коллекции
  из трех элементов */
  r1 := rec_var1 (2.0, 2.1, 2.2);
  /*Инициализация varray-массива,
  содержащего объекты типа rec_obj */
  r2 := rec_var2 (rec_obj(1, 100, 'fff'),
                rec_obj (2,110, 'ggg'),
                rec_obj (3,120, 'jjj'));
```

Оператор CREATE TYPE позволяет сохранить определяемый тип в базе данных.

Конструктор можно вызывать в любом месте, где допустим вызов функции. Для того чтобы добавить в таблицу базы данных строку, одно из полей которой имеет тип коллекции, следует использовать конструктор.

В PL/SQL реализован ряд встроенных методов для работы с коллекциями. Эти методы вызываются как `collection_name.method_name[(parameters)]`^[6].

1.14 Объектно-ориентированное программирование в PL/SQL

Объектным типом называется определяемый пользователем тип данных, который инкапсулирует структуру данных и подпрограммы.

Переменные, используемые в структуре данных объектного типа, называются атрибутами, или переменными объектного типа. Функции и процедуры, определяющие поведение объекта, называются методами.

При объявлении переменной объектного типа создается объект с атрибутами и методами, определяемым его типом.

^[6] Сайт <http://datasql.ru/sqlandpol/11.htm>

К преимуществам применения объектных типов можно отнести следующие их возможности:

- использование объектных типов позволяет перемещать код управления данными из блоков PL/SQL в методы объекта с помощью инкапсуляции данных;
- объектный тип ограничивает доступ к структуре объекта, предоставляя свои методы для работы с данными;
- объектные типы хорошо реализуются классами объектно-ориентированных языков программирования.

Объектный тип состоит из двух частей: спецификации и тела.

Спецификация типа доступна в клиентском приложении. Тело типа скрыто, но разработчик существующего в базе данных типа имеет возможность изменять реализацию методов.

Спецификация типа создается оператором CREATE TYPE, который, с некоторыми сокращениями, может иметь следующее формальное описание:

```
CREATE [OR REPLACE] TYPE [schema .]
type_name
{ { IS | AS } OBJECT }
[ { attribute datatype
  [sqlj_object_type_attr] } ] |
{ [ { [[NOT] OVERRIDING]
  [[NOT] FINAL] [[NOT] INSTANTIABLE]]
  { { MEMBER | STATIC }
    { procedure_spec | function_spec } |
    {{ MAP | ORDER } MEMBER function_spec}}}]
., :)
[[NOT] FINAL] [[NOT] INSTANTIABLE];
```

Определение функции (function_spec) указывается как

```
FUNCTION name (parameter datatype ., :)
{ RETURN datatype }
```

Определение процедуры (procedure_spec) указывается как:

```
PROCEDURE name (parameter datatype ., :)
```

Тело типа создается оператором CREATE TYPE BODY, который, с некоторыми сокращениями, может иметь следующее формальное описание:

```
[CREATE TYPE BODY type_name {IS | AS}
  { {MAP | ORDER}
    MEMBER function_body;
  | MEMBER {procedure_body |
    function_body};}
[MEMBER {procedure_body |
  function_body};]... END;]
```


Создание объектного типа выполняется в два этапа: сначала оператором `CREATE TYPE` создается спецификация типа, а затем оператором `CREATE TYPE BODY` создается тело типа.

```
-- Создание объектного типа
CREATE TYPE MyT AS OBJECT (
    r1 REAL,    -- Атрибуты типа
    r2 REAL,
    MEMBER FUNCTION plus (x MyT)
    RETURN MyT
);
CREATE TYPE BODY MyT AS
    MEMBER FUNCTION plus (x MyT)
    RETURN MyT IS
    BEGIN
        RETURN MyT (r1 + x.r1,
                    r2 + x.r2);
    END plus;
END;
-- Применение переменной объектного типа:
DECLARE- Создается объект c1 типа MyT
    c1 MyT;
BEGIN
    -- Вызов конструктора
    c1 := MyT (1,1); END;
```

На атрибут объектного типа накладываются следующие ограничения:

- типом атрибута может быть любой тип данных Oracle за исключением некоторых типов, включая типы `LONG`, `LONG RAW`, `NCHAR`, `NCLOB`, `NVARCHAR2`, `ROWID`, `BOOLEAN`, `PLS_INTEGER`, `RECORD`, `REF CURSOR`, `%TYPE` и `%ROWTYPE`, а также типы, определенные в пакете `PL/SQL`;

- при объявлении атрибута его нельзя инициализировать, используя оператор присваивания или ключевое слово `DEFAULT`;

- на атрибут не может быть наложено ограничение `NOT NULL`.

Типом атрибута может быть любой допустимый тип или другой объектный тип, называемый в этом случае вложенным объектным типом. Объектные типы могут применяться при создании таблиц как типы полей.

```
CREATE TYPE typ1 AS OBJECT (a1 NUMBER,
    MEMBER FUNCTION getf1 RETURN NUMBER);
-- Создание таблицы
CREATE TABLE tbl1(col typ1);
-- Вызов метода объектного типа
SELECT col.getf1() FROM tbl1;
```

Спецификация объектного типа должна включать объявление каждого общедоступного метода, реализация которого записывается в теле

объектного типа. Перед названием метода объектного типа при спецификации объектного типа и описании тела этого типа всегда указывается ключевое слово MEMBER.

Для квалификации атрибутов в методах объектного типа можно использовать ключевое слово SELF, рассматриваемое как ссылка на данный объект. Параметр SELF может быть указан первым параметром функции и процедуры и явным способом. Если параметр SELF явно не указывается, то для функции предполагается определение параметра с опцией IN (входной параметр), а для процедуры – с опцией IN OUT (входной-выходной параметр). Объектный тип может иметь перегружаемые методы.

В отличие от значений скалярных типов для значений объектных типов не существует единого правила их сравнения. В языке PL/SQL определены ключевые слова MAP и ORDER, которые позволяют специфицировать метод, определяющий правила сравнения значений объектного типа.

Объектный тип может иметь только один метод, выполняющий сравнение экземпляров данного типа, – MAP-метод или ORDER-метод. В зависимости от значения объектного типа MAP-метод можно рассматривать как некоторую функцию хеширования.

ORDER-метод – это функция с двумя параметрами (первый из которых является встроенным по умолчанию), возвращающая значение скалярного типа (DATE, NUMBER, VARCHAR2, CHARACTER или REAL), получаемое при сравнении первого параметра, всегда равного SELF, со вторым параметром, указывающим объект этого же типа.

В SQL-операторах всегда можно сравнивать два значения объектного типа на эквивалентность – полное совпадение значений всех атрибутов, но выполнять сравнение на упорядочивание можно только в том случае, если объектный тип имеет MAP-метод или ORDER-метод. Конструктор используется для инициализации и возвращения экземпляра объектного типа. При инициализации объекта вызывается конструктор со списком параметров, которые определяют атрибуты в порядке их объявления ^[7].

1.15 Пакеты языка PL/SQL

Пакет – это объект схемы, который объединяет логически зависимые типы PL/SQL, данные и подпрограммы. Пакет состоит из двух частей: спецификации пакета и тела пакета. В спецификации пакета объявляются доступные типы, переменные, константы, исключения, курсоры и подпрограммы. В теле пакета содержится определение курсоров и реализация подпрограмм.

Все элементы, объявляемые в теле пакета, невидимы для приложения, что позволяет скрывать от пользователя детали.

^[7] Сайт <http://datasql.ru/sqlandpol/13.htm>

Определение спецификации пакета выполняется оператором CREATE PACKAGE, который может иметь следующее формальное описание:

```
-- Спецификация (видимая часть)
CREATE PACKAGE name AS
-- Объявление общедоступных типов и переменных
-- Спецификация подпрограмм
END [name];
```

Определение тела пакета выполняется оператором CREATE PACKAGE BODY:

```
-- Тело пакета (скрытая часть)
CREATE PACKAGE BODY name AS
-- Объявление локальных типов и переменных
END [name];
```

Создание пакета со спецификацией и телом пакета:

```
-- Спецификация пакета
CREATE PACKAGE tbl_rows AS
  TYPE RecTBL1 IS RECORD
    (tbl1f1 INTEGER, tbl1f2 REAL);
  PROCEDURE insert_tbl1
    (f1 INTEGER, f2 REAL);
END tbl_rows;
-- Тело пакета
CREATE PACKAGE BODY tbl_rows AS
  PROCEDURE insert_tbl1
    (f1 INTEGER, f2 REAL) IS
  BEGIN
    INSERT INTO tbl1
      VALUES (f1, f2);
  END insert_tbl1;
END tbl_rows;
```

На общедоступные элементы пакета – типы, переменные и методы – можно ссылаться из триггеров, хранимых подпрограмм или OCI-приложений, используя следующий синтаксис: package_name.item_name.

Вызов пакетной процедуры из встроенного SQL может быть реализован в анонимном блоке, для выполнения которого используется оператор EXECUTE.

```
EXEC SQL EXECUTE
BEGIN
  tbl_rows.insert_tbl1(1, 200); END;
```

1.16 Внешние процедуры

Использование внешних процедур значительно расширяет функциональность сервера, предоставляя интерфейс для вызова программ, разработанных на других языках программирования, таких как C++ или Pascal, и созданных как DLL-библиотеки.

Во время выполнения блока PL/SQL динамически загружается DLL-библиотека и происходит вызов внешней процедуры как подпрограммы PL/SQL.

Внешние процедуры всегда выполняются в отдельном процессе. Вызовы внешних процедур можно выполнять из:

- анонимных блоков PL/SQL;
- методов, объявленных в объектном типе;
- хранимых или пакетных подпрограмм;
- триггеров базы данных;
- SQL-операторов, вызываемых только для пакетных подпрограмм.

Для применения внешней процедуры следует сначала определить используемую DLL-библиотеку, выполнив оператор `CREATE LIBRARY library_name {IS | AS} 'file_path';`.

Затем необходимо зарегистрировать внешнюю процедуру, указав в процедуре PL/SQL, используемой для опосредованного вызова внешней процедуры, место расположения процедуры, способ ее вызова и список передаваемых параметров.

Если DLL-библиотека `c_lib.dll` содержит функцию `f1_ext`, которая определена в C как `int c_ext_f1(int x_val, int y_val);`, то подпрограмма PL/SQL `reg_ext_f1`, выполняющая регистрацию этой внешней процедуры, может быть записана следующим образом:

```
CREATE FUNCTION reg_ext_f1(  
    x BINARY_INTEGER, y BINARY_INTEGER)  
    RETURN BINARY_INTEGER AS EXTERNAL  
    LIBRARY c_lib  
    - Имя внешней функции  
    NAME "c_ext_f1"  
    LANGUAGE C;
```

Для вызова внешней функции DLL-библиотеки из анонимного блока PL/SQL достаточно выполнить вызов функции PL/SQL, зарегистрированной как внешняя функция ^[8].

```
DECLARE  
    var1 BINARY_INTEGER;  
    x BINARY_INTEGER;  
    y BINARY_INTEGER;  
BEGIN  
    x:=1; y:=2;  
    -- Вызов внешней функции f1_ext:
```

[8] Сайт <http://datasql.ru/sqlandpol/13.htm>

```
var1 := reg_ext_f1(x, y);  
END;
```

2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Общая постановка задачи

Темой дипломного проекта является «Разработка модуля отчетности в среде Oracle Business Intelligence Publisher для оптимизации нагрузки на АБИС». Модуль отчетности в среде Oracle Business Intelligence Publisher – это сложная система, которая состоит из множества отчетов и моделей данных.

Модуль отчетности в среде Oracle Business Intelligence – обширный комплекс технологий, аналитических приложений технологических BI-платформ и хранилищ данных, позволяющий преобразовывать данные, накапливаемые в процессе операционной деятельности, в структурированную информацию для анализа и принятия решений. В Oracle BI присутствуют все средства для проведения анализа и подготовки отчетов любой сложности, обеспечения безопасности и гибкого разделения доступа к информации. Oracle BI – инструмент для формирования стратегической отчетности, который позволяет получать данные из разных источников, формировать различные макеты представления информации пользователю и передавать отформатированные отчеты широкому спектру устройств и инструментов. Шаблоны внешнего вида для отчетов BI Publisher могут быть разработаны с помощью таких широко распространенных и хорошо знакомых инструментов, как, например, Microsoft Word или Adobe Acrobat. Используя BI Publisher, можно формировать расписание выполнения отчетов и при необходимости передавать отформатированный результат нескольким адресатам.

В банке имеется рабочая (фронтальная) система с собственной базой данных; в ежедневном использовании БД начинает расти и выгрузка отчетности на прошедшие даты становится более долгой, что в целом тормозит работу автоматизированной банковской информационной системы Банка. (АБИС). Так как банк не может заставлять клиентов ожидать, было принято решение, внедрения ХД, которое позволит реализовать историзацию изменений в АБИС.

Благодаря реализации ХД появилась возможность перенесение выгрузки отчетности из АБИС в ХД. В связи с этим был открыт проект

«Хранилище данных». Реализация хранилища данных произведена на Oracle Database 11g. Для формирования отчетности (интерфейс) была развернута система Oracle Business Intelligence Publisher, которая позволяет обращаться к пакетам в ХД и выгружать коллекции в шаблоны, реализованные на стандартных приложениях с поддержкой BI Publisher (MS Office, Adobe Acrobat, HTML).

Задачей для дипломного проекта стала реализация отчетности и оптимизация выгрузки данных из ХД, с выгрузкой отчетов и анализом сверочных данных, что позволяет выявить возможные ошибки в ХД. И в дальнейшем предоставить полноценный продукт.

2.2 Описание выгружаемых отчетов

Создание отчета проходит в два этапа, реализация пакета для выгрузки данных в ХД и создание визуального представления в BI Publisher. Пакет должен выгружать коллекцию данных для обработки в BI Publisher. В пакете используются созданные для отчёта тип строки и тип таблицы (массив строк).

«Отчет о финансовых потоках и запасах», «Отчет об остатках на балансовых счетах за вычетом резервов (провизий)» и отчет «Основные показатели баланса» используются для предоставления в Национальный банк Казахстана. Период формирования «Отчета о финансовых потоках и запасах» раз в год, «Отчет об остатках на балансовых счетах за вычетом резервов (провизий)» и отчет «Основные показатели баланса» обновление требуется раз в месяц. Данные приводятся в целом по банку.

Формы отчетов представлены в приложении Б.

2.2.1 Связи фактов и измерений для выгружаемых отчетов

Для реализации данных отчетов используются таблица фактов главной книги (GL_ACC_F) (таблица 2.1) и измерения:

- счета главной книги (GL_LEDGER_ACCOUNT_D);
- план счетов (GL_PLAN_ACCOUNT_D);
- вид учета (GL_CALC_VARIETY_D);
- тип счета (GL_CALC_TYPE_D);
- тип счета (GL_ACCOUNT_TYPE_D);
- произвольно задаваемые пользователями атрибуты счетов для дальнейшего их использования в отчетности (GL_ACCOUNT_FLAG_D);
- вид активности/пассивности балансового счета (GL_ACCOUNT_AP_KIND_D);
- валюта (GL_CURRENCY_D);
- физическое лицо (G_NAT_PERSON_D);
- юридическое лицо (GL_JUR_PERSON_D).

Таблица фактов GL_ACC_F – содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться.

Таблицы измерений – таблицы в структуре хранилища данных, которые содержат атрибуты событий, сохраненных в таблице фактов.

Главная книга (ГК) – содержит остатки и обороты по типовому плану счетов (ТПС).

С помощью главной книги можно увидеть обороты выбранного счета за отчетный период с заданной детализацией. Также она позволяет разворачивать обороты по корреспондирующим счетам и выводить остатки по указанной детализации.

Т а б л и ц а 2.1 – Факты главной книги

Наименование поля	Комментарии	Связывающееся измерение	Поле
WORK_DATE	Операционный день		
ZO_SIGN	Признак зо		
INPUT_ACTIVE	Входящее значение в тенге по валюте		
INPUT_PASSIVE	Входящее значение в тенге по валюте		
TURN_DEBET_PLAN	Оборот по кредиту факт		
TURN_CREDIT_PLAN	Оборот по кредиту плановый		
TURN_DEBET_FACT	Оборот по дебету факт		
TURN_CREDIT_FACT	Оборот по дебету плановый		
OUTPUT_ACTIVE_FACT	Исходящий остаток активный		
OUTPUT_ACTIVE_PLAN	Исходящий остаток плановый		
OUTPUT_PASSIVE_FACT	Исходящей фактический остаток пассивное значение		
OUTPUT_PASSIVE_PLAN	Исходящей плановый остаток пассивное значение		
KZT_INPUT_ACTIVE	Входящее значение в тенге по валюте		
KZT_INPUT_PASSIVE	Входящее значение в тенге по валюте		
KZT_TURN_DEBET_PLAN	Оборот по кредиту факт		
KZT_TURN_CREDIT_PLAN	Оборот по кредиту плановый		
KZT_TURN_DEBET_FACT	Оборот по дебету факт		
KZT_TURN_CREDIT_FACT	Оборот по дебету плановый		
KZT_OUTPUT_ACTIVE_FACT	Исходящий остаток активный		
KZT_OUTPUT_ACTIVE_PLAN	Исходящий остаток плановый		
KZT_OUTPUT_PASSIVE_FACT	Исходящей фактический остаток пассивное значение		
KZT_OUTPUT_PASSIVE_PLAN	Исходящей плановый остаток пассивное значение		
INPUT	Входящее сальдо		

OUTPUT	Исходящее сальдо		
KZT_INPUT	Входящее сальдо		
KZT_OUTPUT	Исходящее сальдо		
GL_LEDGER_ACCOUNT_ID	Счета главной книги	GL_LEDGER_ACCOUNT_D	ID
GL_PLAN_ACCOUNT_ID	План счетов	GL_PLAN_ACCOUNT_D	ID
GL_CALC_VARIETY_ID	Вид учета	GL_CALC_VARIETY_D	ID
GL_CALC_TYPE_ID	Вид счета	GL_CALC_TYPE_D	ID
GL_ACCOUNT_TYPE_ID	Тип счета	GL_ACCOUNT_TYPE_D	ID

Окончание таблицы 2.1

Наименование поля	Комментарии	Связываемое измерение	Поле
GL_ACCOUNT_FLAG_ID	Произвольно задаваемые пользователями атрибуты счетов для дальнейшего их использования в отчетности	GL_ACCOUNT_FLAG_D	ID
GL_ACCOUNT_AP_KIND_ID	Вид активности, пассивности балансового счета	GL_ACCOUNT_AP_KIND_D	ID
GL_CURRENCY_ID	Валюта	GL_CURRENCY_D	ID
GL_NAT_PERSON_ID	Физическое лицо	GL_NAT_PERSON_D	ID
GL_JUR_PERSON_ID	Юридическое лицо	GL_JUR_PERSON_D	ID

Таблица 2.2 содержит измерения счетов главной книги, связующим полем является идентификатор ХД (ID).

Т а б л и ц а 2.2 – Измерения счетов (GL_LEDGER_ACCOUNT_D)

Наименование поля	Комментарии
ID	Идентификатор ХД
CODE_SYSTEM	ID счет главной Книги
CODE	Код
NAME	Наименование
SHORT_NAME	Краткое наименование
LEVEL_NUM	Номер уровня
ITEM_KIND	Тип элемента
IS_LAST	Признак последнего элемента в иерархии
BEGIN_DATE	Дата начала срока действия записи в ХД
END_DATE	Дата окончания срока действия записи в ХД

Таблица 2.3 содержит измерения плана счетов главной книги, связующим полем является идентификатор ХД (ID).

Т а б л и ц а 2.3 – Измерения плана счетов (GL_PLAN_ACCOUNT_D)

Наименование поля	Комментарии
CODE_SYSTEM	Код системы
CODE	Код
ID	Идентификатор ХД
NAME	Наименование
OPEN_DATE	Дата открытия
LEVEL_NUM	Номер уровня
ITEM_KIND	Тип элемента
BEGIN_DATE	Дата начала срока действия записи в ХД
END_DATE	Дата окончания срока действия записи в ХД

Таблица 2.4 содержит измерения видов учета главной книги, связующим полем является вид учета (ID).

Т а б л и ц а 2.4 – Измерения видов учета (GL_CALC_VARIETY_D)

Наименование поля	Комментарии
ID	Вид учёта id: 1 - стоимостной учёт; 2 - количественный учёт.
NAME	Вид учёта наименование

Таблица 2.5 содержит измерения видов счета главной книги, связующим полем является вид счета (ID).

Т а б л и ц а 2.5 – Измерения видов счета (GL_CALC_TYPE_D)

Наименование поля	Комментарии
ID	Вид счета id
NAME	Вид счета наименование

Таблица 2.6 содержит измерения типов счетов главной книги, связующим полем является тип счета (ID).

Типы счёта могут быть:

- 1 – счёт в платежной системе (коррсчет, внутрибанк. лицевой счёт);
- 2 – внутрибанковский лицевой счёт;
- 3 – клиентский лицевой счёт;
- 4 – зеркальный счёт;
- 50 – внебалансовый лицевой счёт.

Т а б л и ц а 2.6 – Измерения типов счета (GL_ACCOUNT_TYPE_D)

Наименование поля	Комментарии
CODE_SYSTEM	Системный код
BEGIN_DATE	Дата начала срока действия записи в ХД
END_DATE	Дата окончания срока действия записи в ХД
ID	Тип счета id

NAME	Тип счета наименование
------	------------------------

Таблица 2.7 содержит измерения, произвольно задаваемые пользователями атрибуты счетов для дальнейшего их использования в отчетности, связующим полем является атрибут (ID).

Т а б л и ц а 2.7 – Измерения атрибутов счетов (GL_ACCOUNT_FLAG_D)

Наименование поля	Комментарии
CODE_SYSTEM	Системный код
BEGIN_DATE	Дата начала срока действия записи в ХД
END_DATE	Дата окончания срока действия записи в ХД

Окончание таблицы 2.7

Наименование поля	Комментарии
ID	Атрибут id
CODE	Код
NAME	Наименование атрибута

Таблица 2.8 содержит измерения вида активности/пассивности балансового, связующим полем является вид (ID).

Т а б л и ц а 2.8 – Измерения балансового счета (GL_ACCOUNT_AP_KIND_D)

Наименование поля	Комментарии
ID	Вид id
NAME	Наименование

Таблица 2.9 содержит измерения валюты, связующим полем является валюта (ID).

Т а б л и ц а 2.9 – Измерения валюты (GL_CURRENCY_D)

Наименование поля	Комментарии
ID	Валюта id
CODE_SYSTEM	Системный код
NAME	Наименование
SHORT_NAME	Сокращенное наименование

Таблица 2.10 содержит измерения физического лица, связующим полем является идентификатор ХД (ID).

Т а б л и ц а 2.10 – Измерения физического лица (G_NAT_PERSON_D)

Наименование поля	Комментарии
ADDRESS_NAT	Адрес на нац. языке
PHONE	Телефон (дом.)
PHONE_CELL	Телефон (моб.)
FAX	Факс

E_MAIL	Email
HOUSEHOLD_CNT	Количество членов семьи
DEPENDANT_CNT	Количество иждивенцев
DEPENDANT_CNT_18	Количество иждивенцев после 18 лет
SOCIALSTS_CODE	Социальный статус код
SOCIALSTS_NAME	Социальный статус
NOTE	Примечание
IS_AFFILIATED	Аффилированный
BIRTH_PLACE	Место рождения
BIRTH_PLACE_NAT	Место рождения на национальном языке
IS_REPUBLIC_SITIZEN	Признак гражд. Республики

Продолжение таблицы 2.10

Наименование поля	Комментарии
WORK_ADDRESS	Адрес работы
WORK_PLACE	Место работы
APPOINTMENT	Должность
SENIORITY	Стаж работы общих лет
SENIORITY_LAST	Стаж работы последних лет
IS_ENTREPRENEUR	Частный предприниматель
AREA_REG_CODE	Адрес по прописке/Область код
AREA_REG_NAME	Адрес по прописке/Область
CITY_REG_CODE	Адрес по прописке/Город код
CITY_REG_NAME	Адрес по прописке/Город
CITY_FACT_CODE	Адрес фактического проживания/Город код
CITY_FACT_NAME	Адрес фактического проживания/Город
DISTRICT_REG	Адрес по прописке/Район
DISTRICT_FACT	Адрес фактического проживания/Район
STREET_REG	Адрес по прописке/Улица
STREET_FACT	Адрес фактического проживания/Улица
NUM_HOUSE_REG	Адрес по прописке/№ дома
NUM_HOUSE_FACT	Адрес фактического проживания/№ дома
MICRODIS_REG	Адрес по прописке/Микрорайон
MICRODIS_FACT	Адрес фактического проживания/Микрорайон
NUM_APART_REG	Адрес по прописке/№ квартиры
NUM_APART_FACT	Адрес фактического проживания/№ квартиры
POST_ADDRESS	Почтовый адрес (1- адрес по прописке, 2 - адрес фактического проживания)
REG_ADDRESSFL	(1- Адрес фактического проживания = Адрес по прописке, 0 - другое)
RESIDING_PERIOD_CODE	Период проживания
RESIDING_PERIOD_NAME	Период проживания
MLTRCARDFL	Признак наличия военного билета (0-нет, 1- есть)
SPOUSE_FM	Фамилия супруга/супруги
SPOUSE_NM	Имя супруга/супруги
SPOUSE_FT	Отчество супруга/супруги
SPOUSE_COUNTRY_CODE	Страна гражданства супруга/супруги код

SPOUSE_COUNTRY_NAME	Страна гражданства супруга/супруги
SPOUSE_SOCIALSTS_CODE	Социальный статус супруга/супруги код
SPOUSE_SOCIALSTS_NAME	Социальный статус супруга/супруги
MARRDOGFL	Признак наличия брачного контракта (0-нет, 1- есть)
MARRDOG	Документ о заключении брака
ORDER_RETURN_CODE	Порядок возврата займа код
ORDER_RETURN_NAME	Порядок возврата займа
BANK_CODE	Банк - держатель платежной карты код
BANK_NAME	Банк - держатель платежной карты
CHGCARD_CODE	Платежная карта код
CHGCARD_NAME	Платежная карта

Окончание таблицы 2.10

Наименование поля	Комментарии
CHGCARD_NUM	Номер платежной карты
CRDLIMIT_SUM	Сумма кредитного лимита
MULTICARDFL	Признак наличия нескольких платежных карт
SUM_LIABILITY	Сумма обязательства
BEGIN_DATE	Дата начала срока действия записи в ХД
END_DATE	Дата окончания срока действия записи в ХД
ID	Идентификатор ХД
CODE_SYSTEM	ID физическое лицо
FM	Фамилия
NM	Имя
FT	Отчество
FM_NAT	Фамилия на нац. языке
NM_NAT	Имя на нац. языке
FT_NAT	Отчество на нац. языке
FIO	ФИО
FIO_PREV	ФИО (прежнее)
DT	Дата рождения
FAMSTS_CODE	Код семейного положения
FAMSTS_NAME	Наименование семейного положения
SEX_CODE	Пол код
SEX_NAME	Пол наименование
NATION_CODE	Национальность код
NATION_NAME	Национальность наименование
COUNTRY_CODE	Страна код
COUNTRY_NAME	Страна наименование
EDUCATION_CODE	Образование код
EDUCATION_NAME	Образование наименование
WORK_TYPE_CODE	Тип работы код
WORK_TYPE_NAME	Тип работы наименование
ID_KIND_NAME	Вид удост. личности
ID_SERIAL	Серия документа
ID_NUM	Номер документа
ID_DATE	Дата выдачи документа

ID_DATE_END	Срок действия документа удостоверяющего личность
ID_ISSUER	Кем выдан. документ
ID_ISSUER_NAT	Кем выдан. документ на нац. языке
IDN	ИИН
PIND	Почтовый индекс
ADDRESS	Адрес

Таблица 2.11 содержит измерения юридического лица, связующим полем является идентификатор ХД (ID).

Т а б л и ц а 2.11 – Измерения юридического лица (GL_JUR_PERSON_D)

Наименование поля	Комментарии
ID	Идентификатор ХД
CODE_SYSTEM	ID Юридическое лицо
NAME	Наименование
NAME_NAT	Наименование на национальном языке
SHORT_NAME	Краткое наименование
PIND	Почтовый индекс
ADDRESS	Адрес
ADDRESS_NAT	Адрес на национальном языке
PHONE	Телефон
FAX	Факс
E_MAIL	EMail
BIK	БИК
BIK_HEAD	Головной БИК
BIK_NBRK	БИК в НБРК
ACCOUNT_CORR	Коррсчет
CHIEF_FM	Фамилия руководителя
CHIEF_NM	Имя руководителя
CHIEF_FT	Отчество руководителя
CHIEF_APPOINTMENT	Должность первого руководителя
MAIN_BK_FM	Фамилия Гл.Бух.
MAIN_BK_NM	Имя Гл. Бух.
MAIN_BK_FT	Отчество Гл. Буха.
OPEN_DATE	Дата регистрации
CLOSE_DATE	Дата закрытия
CRYPT_CODE_KCMR	Код в криптосети КЦМР
CRYPT_CODE_NBRK	Код в криптосети НБРК
NKCB_CODE	Код НКЦБ
RNN	РНН
IDN	БИН
IS_AFFILIATED	Аффилированный
IS_MEMBER_NB	Абон. критосети НБРК
IS_HAS_CRYPT_CODE	Имеет криптокод
IS_CAN_RECEIVE_MT102	Может принимать МТ102
NBRK_CODE	Код НБ РК

NOTES	Примечание
BEGIN_DATE	Дата начала срока действия записи в ХД

Также для реализации данных отчетов используются таблица фактов банковской формы Н700 (GL_700_F) (таблица 2.12) и измерения (таблица 2.13).

Банковская форма Н700 содержит остатки классифицированных счетов, т.е. которые скомпонованы по ТПС.

Т а б л и ц а 2.12 – Факты банковской формы (GL_700_F)

Наименование поля	Комментарии	Связывающееся измерение	Поле
GL_700_ID	Форма Н700 id	GL_700_D	ID
WORK_DATE	Дата		
VAL	Сумма		
ISZO	Признак 30		
ISAPP	Признак приложения		

Т а б л и ц а 2.13 – Измерения показателей формы (GL_700_D)

Наименование поля	Комментарии
ID	Идентификатор ХД
CODE_POKAZ	Код показателя
NAME_POKAZ	Наименование показателя
ISAPP	Признак приложения
BEGIN_DATE	Дата начала действия записи
END_DATE	Дата окончания действия записи

Также в отчете используются настроечные таблицы SET_REP_COLUMN_CALCULATION и SET_REPORT_COLUMN (таблицы 2.14, 2.15), которые используются для оптимизации типовых отчетов, которые содержат одинаковые запросы, только с разными данными параметрами. В данном случае имеется две настроечных таблицы, одна заполняет отчет строками (наименование и позиции), вторая включает в себя ТПС, сектор экономики и знак операции.

Т а б л и ц а 2.14 – Настроечная таблица для строк

Наименование поля	Комментарии
REPORT_ID	Отчет ID
COLUMN_ID	Столбец ID
CALCULATION_ID	Расчет ID
CALCULATION_SIGN	Признак расчета
IN_PA4CODE	Входящий код
OUT_PA4CODE	Исходящий код
IN_SECTOR	Входящий сектор

OUT_SECTOR	Исходящий сектор
------------	------------------

Т а б л и ц а 2.15 – Настроечная таблица для столбцов

Наименование поля	Комментарии
REPORT_ID	Отчет ID
COLUMN_ID	Колонка ID
COLUMN_NUM	Номер колонки
COLUMN_NAME	Название столбца
PARENT_COLUMN	Главный столбец
COLUMN_GROUP	Группа столбцов

2.2.2 Требования и процесс создания пакета для «Отчета о финансовых потоках и запасах»

Для того чтобы создать модель данных в Oracle Business Intelligence Publisher, необходимо реализовать пакет (приложение А) в Oracle SQL Developer интегрированной среде разработки на языках SQL и PL/SQL, ориентированная на применение в среде Oracle Database.

Обязательным требованием к отчету при его составлении является идентичность данных по запасам на начало и конец отчетного периода по балансовым данным с детализацией счетов.

Данные приводятся по всем финансовым инструментам, в разрезе секторов внутренней экономики, по видам валют.

По каждому финансовому инструменту отчета, запасы на конец отчетного периода должны быть равны запасам на начало отчетного периода плюс все изменения за отчетный период (изменения в результате проведения операций, переоценка стоимости активов, другие изменения).

В переменные vBeginDate и vEndDate записываются рабочие даты очень близкие или равные входящим параметрам aBeginDate и aEndDate, т. к. отчетные даты не могут быть выходными днями.

```

begin
select /*+ NO_PARALLEL*/ max(a.work_date)
  into vBeginDate
  from gl_working_date_d a
  where a.work_date < aBeginDate;

select /*+ NO_PARALLEL*/ max(a.work_date)
  into vEndDate
  from gl_working_date_d a
  where a.work_date <= aEndDate;

```

Измерение «Рабочая дата» является одним из критических элементов модели ХД. В системах поддержки принятия решений, для которых проектируются и создаются ХД, запросы фокусируются на задачах анализа данных, а именно – как данные изменялись в различные периоды времени.

В каждом запросе к ХД применяется хинт `/*+ NO_PARALLEL*/`, помогает отключить режим параллельного выполнения при отдельном выполнении подзапроса. Другими словами хинт – это ключевое слово, иногда с набором параметров, которое может повлиять на оптимизатор при составлении плана запроса. Другими словами, с помощью хинтов мы можем попытаться изменить способ получения и обработки данных.

Для каждого запроса (SELECT) создается новая строка (vRow) с названием значениями (в этом случае строки «Наличная валюта» и «1. Национальная» не имеет данных).

```
vRow := new TFINANCE_FLOW_RESERVE_R(1, 0, null, 'Наличная
валюта', null, null, null, null, null, null, null, null,
null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(2, 1, null, '1.
Национальная', null, null, null, null, null, null, null,
null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;
```

Все суммы указываются в тысячах тенге, для приведения сумм к тыс. тенге используется следующий алгоритм: все значения в отчете делятся на тысячу, после чего к ним применяют правила округления.

Для создания пакета используется команда:

```
create or replace package body REP_FIN_FLOW_RESERVE_perenos
is
-- объявление локальных типов
-- переменных
-- тела подпрограмм
end REP_FIN_FLOW_RESERVE_perenos;
```

В теле пакета объявляем переменные:

```
vTable TFINANCE_FLOW_RESERVE_T := TFINANCE_FLOW_RESERVE_T();
vRow TFINANCE_FLOW_RESERVE_R;
v3 Number(12);
v4 Number(12);
v5 Number(12);
v6 Number(12);
v7 Number(12);
v8 Number(12);
v9 Number(25);
v11 Number(12);
v12 Varchar2(255);
vBeginDate Date;
vEndDate Date;
```


Отчет о финансовых потоках и запасах состоит из двух выходящих форм финансовые активы и пассивы, для каждой формы написана соответствующая функция.

Далее в теле пакета необходимо создать функцию для выгрузки финансовых пассивов (GetFinanceAssets) с входящими параметрами даты на начало и конец периода.

```
function GetFinanceAssets(aBeginDate In Date, aEndDate In Date) return TFINANCE_FLOW_RESERVE_T;  
function GetFinanceLiabilities(aBeginDate In Date, aEndDate In Date) return TFINANCE_FLOW_RESERVE_T;  
end REP_FIN_FLOW_RESERVE_perenos;
```

2.2.3 Основные финансовые инструменты «Отчета о финансовых потоках и запасах»

При заполнении отчета следует отражать следующие финансовые инструменты.

Разбивка по секторам и видам валют осуществляется с использованием ТПС бухгалтерского учета с детализацией счетов для составления Главной бухгалтерской книги банков второго уровня и ипотечных организаций, утвержденного Постановлением Правления Национального Банка Республики Казахстан 27.12.2010 года №105.

Наличная валюта включает наличность в кассе, в обменных пунктах, в банкоматах, в вечерней кассе, банкноты и монеты в пути. В отчете отражается по графе 1 «Всего, тыс.тенге». Используются ТПС 1001, 1002, 1003, 1004, 1005.

```
select /*+ NO_PARALLEL*/ nvl(sum(val), 0)  
into v3  
from gl_700_f f, gl_700_d d  
where f.work_date = vBeginDate  
and f.gl_700_id = d.id  
and substr(d.code_pokaz, 1, 5) in ('10011', '10021',  
'10031', '10041', '10051')  
and substr(d.code_pokaz, 7, 1) = '1';
```

Переводимые депозиты на стороне активов: корреспондентские счета и начисленные доходы по ним, деньги в дорожных чеках в пути, деньги в дорожных чеках. Счета ТПС 1006, 1008, 1051, 1052, 1705, 1053, 1104.

```
select /*+ NO_PARALLEL*/ nvl(sum(val), 0)  
into v3  
from gl_700_f f, gl_700_d d  
where f.work_date = vBeginDate  
and f.gl_700_id = d.id
```

```

and substr(code_pokaz, 1, 4) in ('1006', '1008',
'1051', '1052', '1705', '1053', '1104')
and substr(code_pokaz, 6, 2) = '31';

```

На стороне пассивов: корреспондентские счета, текущие счета, карт-счета, начисленные расходы и просроченное вознаграждение по ним. Счета ТПС 2011, 2012, 2013, 2014, 2202, 2203, 2204, 2209, 2221, 2701, 2718, 2726, 2748.

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121',
'20131', '20141', '22021', '22031', '22041', '22091',
'22211', '27011', '27181', '27261', '27481')
and substr(d.code_pokaz, 6, 2) = '31';

```

Другие депозиты на стороне активов: аффинированные драгоценные металлы, размещенные на металлических счетах, обязательные резервы в Национальном Банке, все депозиты, кроме переводных, включающие вклады, размещенные на одну ночь, вклады до востребования, срочные вклады, непереводимые депозиты в иностранной валюте, вклады, являющиеся обеспечением обязательств банка, условные вклады, а также начисленные доходы, просроченная задолженность и просроченное вознаграждение по ним. Счета ТПС 1013, 1101, 1102, 1103, 1104, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1264, 1710, 1725, 1726, 1728.

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(code_pokaz, 1, 4) in ('1013', '1101',
'1102', '1103', '1104', '1251', '1252', '1253',
'1254', '1255', '1256', '1257', '1264', '1710', '1725',
'1726', '1728')
and substr(code_pokaz, 6, 2) = '31';

```

На стороне пассивов металлические счета в аффинированных драгоценных металлах, вклады до востребования, займы «овернайт» полученные, срочные вклады, условные вклады, деньги республиканского и местного бюджетов, вклады, привлеченные от банков-резидентов на одну ночь, вклады, являющиеся обеспечением обязательств, счета хранения указаний отправителей в соответствии с валютным законодательством Республики Казахстан, кредиторы по документарным расчетам, начисленные расходы и просроченное вознаграждение по перечисленным

инструментам, а также просроченное вознаграждение по корреспондентским счетам. Счета ТПС 2016, 2021, 2022, 2023, 2024, 2111, 2113, 2121, 2122, 2123, 2124, 2125, 2127, 2130, 2133, 2135, 2138, 2201, 2205, 2206, 2207, 2208, 2211, 2212, 2213, 2215, 2216, 2217, 2219, 2223, 2224, 2226, 2232, 2237, 2245, 2711, 2712, 2713, 2714, 2717, 2719, 2720, 2721, 2723, 2742, 2743, 2746, 2747, 2855.

```
select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211',
'20221', '20231', '20241', '21111', '21131', '21211',
'21221', '21231', '21241', '21251', '21271', '21301',
'21331', '21351', '21381', '22011', '22051', '22061',
'22071', '22081', '22111', '22121', '22131', '22151',
'22161', '22171', '22191', '22231', '22241', '22261',
'22321', '22371', '22451', '27111', '27121', '27131',
'27141', '27171', '27191', '27201', '27211', '27231',
'27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '31';
```

Ценные бумаги (ЦБ) (за исключением акций) на стороне активов: ценные бумаги, предназначенные для торговли, ценные бумаги, удерживаемые до погашения, ценные бумаги, имеющиеся в наличии для продажи, учтенные векселя, вознаграждение, начисленное предыдущими векселедержателями по учтенным векселям, опротестованные векселя, начисленные доходы, просроченная задолженность по приведенным инструментам. Счета ТПС 1201, 1202, 1405, 1406, 1425, 1452, 1459, 1481, 1485, 1744, 1745, 1746, 1750, 1752.

Данные по этому участку кода берутся из другой системы (статистика) из таблиц stat.s_pokaz@stat и stat.detail@stat через созданный db-link на стороне ХД @stat. Так как модуль «Ценные бумаги» в АБИС не реализован, и в ХД данных по ЦБ – нет.

```
select /*+ NO_PARALLEL*/ sum(case when
substr(pok.code_pokaz, 5, 3) = '131' then dt.znac else 0 end)
summ3,
sum(case when substr(pok.code_pokaz, 5, 3) = '141' then
dt.znac else 0 end) summ4,
sum(case when substr(pok.code_pokaz, 5, 3) = '151' then
dt.znac else 0 end) summ5,
sum(case when substr(pok.code_pokaz, 5, 3) = '191' then
dt.znac else 0 end) summ6,
sum(case when substr(pok.code_pokaz, 5, 3) = '181' then
dt.znac else 0 end) summ7,
sum(case when substr(pok.code_pokaz, 5, 3) in('111', '121')
then dt.znac else 0 end) summ8,
```

```

        sum(case when substr(pok.code_pokaz,5,3) in('161','171')
then dt.znac else 0 end) summ9,
        sum(case when substr(pok.code_pokaz,5,1) = '2' then
dt.znac else 0 end) summ11
    into v3, v4, v5, v6, v7, v8, v9, v11
    from stat.s_pokaz@stat pok,
        stat.detail@stat dt
    where pok.id_form = 15949
        and pok.dat_beg <= vBeginDate and (pok.dat_end is null
or pok.dat_end > vBeginDate)
        and pok.id = dt.id_pokaz
        and dt.d_report = vBeginDate
        and substr(pok.code_pokaz,1,4)
in('1201','1202','1405','1406','1425','1452','1459','1481','1485
','1744','1745','1746','1750','1752');

```

На стороне пассивов: обязательства по привилегированным акциям, выпущенные в обращение облигации, выпущенные в обращение прочие ценные бумаги, выкупленные облигации (за минусом), выкупленные субординированные облигации (за минусом), субординированные облигации, начисленные расходы, просроченное вознаграждение по приведенным инструментам. Счета ТПС 2301, 2303, 2306, 2405, 2406, 2730, 2744, 2756, 2863.

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
    into v9
    from gl_700_f f, gl_700_d d
    where f.work_date = vEndDate
        and f.gl_700_id = d.id
        and substr(d.code_pokaz, 1, 5) in ('23011', '23031',
'23061', '24051', '24061', '27301', '27441', '27561', '28631')
        and (substr(d.code_pokaz, 6, 2) = '61' or
substr(code_pokaz, 6, 2) = '71');

```

Кредиты и займы на стороне активов: счета по кредитным карточкам, операции обратного РЕПО, предоставленные займы «овердрафт», «овернайт», факторинговые, форфейтинговые операции, прочие займы, предоставленные клиентам, счета дебиторов по документарным расчетам, начисленные доходы и просроченное вознаграждение по указанным активам. Счета ТПС 1301, 1302, 1303, 1304, 1305, 1306, 1309, 1321, 1322, 1323, 1326, 1327, 1328, 1401, 1403, 1407, 1409, 1411, 1417, 1420, 1421, 1422, 1423, 1424, 1429, 1461, 1462, 1491, 1494, 1730, 1733, 1734, 1740, 1741, 1748, 1757, 1771, 1772, 1855.

```

select /*+ NO_PARALLEL*/ round(sum(case when substr(la.code,5,3)
= '131' then ka.kzt_output else 0 end)/1000) summ3,
        round(sum(case when substr(la.code,5,3) = '141' then
ka.kzt_output else 0 end)/1000) summ4,
        round(sum(case when substr(la.code,5,3) = '151' then
ka.kzt_output else 0 end)/1000) summ5,

```

```

round(sum(case when substr(la.code,5,3) = '191' then
ka.kzt_output else 0 end)/1000) summ6,
round(sum(case when substr(la.code,5,3) = '181' then
ka.kzt_output else 0 end)/1000) summ7,
round(sum(case when substr(la.code,5,3) in('111','121')
then ka.kzt_output else 0 end)/1000) summ8,
round(sum(case when substr(la.code,5,3) in('161','171')
then ka.kzt_output else 0 end)/1000) summ9,
round(sum(case when substr(la.code,5,1) = '2' then
ka.kzt_output else 0 end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and (ga.code in('1461','1748') or
(ga.code in ('1411', '1301', '1302', '1303',
'1304', '1305', '1306', '1309', '1321',
'1322', '1323', '1326', '1327', '1328',
'1401', '1403', '1407', '1409', '1420',
'1421', '1422', '1423', '1424', '1429',
'1462', '1491', '1494', '1730', '1733',
'1734', '1740', '1741', '1757', '1771',
'1772', '1855')
and exists(select /*+ NO_PARALLEL*/ *
from gl_acc_f a,
gl_plan_account_d b
where a.work_date = vBeginDate
and a.kzt_output <> 0
and a.gl_plan_account_id = b.id
and b.code = '1411'
and a.gl_nat_person_id =
ka.gl_nat_person_id)));

```

На стороне пассивов: займы полученные, полученный финансовый лизинг, начисленные расходы, просроченная задолженность, просроченное вознаграждение по ним, операции «РЕПО», субординированный долг, бессрочные финансовые инструменты, начисленные расходы по ним, вклады дочерних организаций специального назначения. Счета ТПС 2034, 2036, 2038, 2041, 2042, 2044, 2046, 2048, 2051, 2052, 2054, 2056, 2057, 2058, 2059, 2064, 2066, 2067, 2068, 2227, 2230, 2231, 2255, 2401, 2402, 2451, 2703, 2704, 2705, 2706, 2707, 2722, 2725, 2740, 2741, 2745, 2757.

```

select /*+ NO_PARALLEL*/ round(sum(case when
substr(la.code,5,3) = '131' then ka.kzt_output else 0
end)/1000) summ3,
round(sum(case when substr(la.code,5,3) = '141' then
ka.kzt_output else 0 end)/1000) summ4,

```

```

round(sum(case when substr(la.code,5,3) = '151' then
ka.kzt_output else 0 end)/1000) summ5,
round(sum(case when substr(la.code,5,3) = '191' then
ka.kzt_output else 0 end)/1000) summ6,
round(sum(case when substr(la.code,5,3) = '181' then
ka.kzt_output else 0 end)/1000) summ7,
round(sum(case when substr(la.code,5,3) in('111','121')
then ka.kzt_output else 0 end)/1000) summ8,
round(sum(case when substr(la.code,5,3) in('161','171')
then ka.kzt_output else 0 end)/1000) summ9,
round(sum(case when substr(la.code,5,1) = '2' then
ka.kzt_output else 0 end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code
in ('2034', '2036', '2038', '2041', '2042',
'2044', '2046', '2048', '2051', '2052', '2054',
'2056', '2057', '2058', '2059', '2064', '2066',
'2067', '2068', '2227', '2230', '2231', '2401',
'2402', '2451', '2703', '2704', '2705', '2706',
'2707', '2722', '2740', '2741', '2745', '2757');

```

Акции и другие формы участия в капитале на стороне пассивов: общие резервы (проезвизии), резервы переоценки, нераспределенный чистый доход (непокрытый убыток), счета капитала (объявленный уставный капитал, дополнительный оплаченный капитал, резервный капитал), выкупленные акции, выкупленные вклады и паи, счет корректировки на гиперинфляцию. В данную категорию необходимо включать также привилегированные акции, которые обеспечивают участие в распределении остаточной стоимости при ликвидации предприятия. Счета ТПС 3001, 3003, 3025, 3027, 3101, 3510, 3540, 3561, 3580, 3589, 3599.

```

select      /*+      NO_PARALLEL*/      round(sum(case      when
substr(la.code,5,3)      =      '131'      then      ka.kzt_output      else      0
end)/1000) summ3,
round(sum(case when substr(la.code,5,3) = '141'
or (substr(la.code,6,1) = '0' and substr(la.code,5,1) <>
'2')
then ka.kzt_output else 0      end)/1000) summ4,
round(sum(case when substr(la.code,5,3) = '151' then
ka.kzt_output else 0 end)/1000) summ5,
round(sum(case when substr(la.code,5,3) = '191' then
ka.kzt_output else 0 end)/1000) summ6,
round(sum(case when substr(la.code,5,3) = '181' then
ka.kzt_output else 0 end)/1000) summ7,

```

```

round(sum(case when substr(la.code,5,3) in('111','121')
then ka.kzt_output else 0 end)/1000) summ8,
round(sum(case when substr(la.code,5,3) in('161','171')
then ka.kzt_output else 0 end)/1000) summ9,
round(sum(case when substr(la.code,5,1) = '2' then
ka.kzt_output else 0 end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code in('3001', '3003', '3025',
'3027', '3101', '3510', '3540', '3561',
'3580', '3589', '3599');

```

Для каждого запроса создаются строки vRow (массив строк), куда выводятся переменные, в данном запросе переменные v3, v4, v5, v6, v7, v8, v9, v11 имеют значения, в остальных переменных необходимо указывать null.

```

vRow := new TFINANCE_FLOW_RESERVE_R(237, 2, 187, 'объем на
начало отчетного периода', null, null, v3, v4, v5, v6, v7,
v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

Остальные запросы должны быть расшифрованы и построены по аналогичному алгоритму, где меняются только ТПС. Также имеются строки итогов, которые собираются суммой или разностью некоторых полей. Записываются в массив данных аналогичным способом, что и переменные из выборок.

На счетах отражаются изменения в активах и пассивах между начальным и заключительным балансом. Финансовые потоки подразделяются на отдельные компоненты: операции (поступление средств/приобретение активов и списание средств/погашение стоимости приобретенных активов), переоценку стоимости (оценочные изменения стоимости) и другие изменения стоимости финансовых активов.

2.2.4 Выгрузка коллекции в PL/SQL Developer

После создания пакета, производится выгрузка коллекции инструментами PL/SQL Developer, для контроля данных и формирования выгружаемого шаблона.

Пакет состоит из двух процедур «GetFinanceAssets» и «GetFinanceLiabilities», с входящими параметрами vBeginDate – «дата начала периода» и vEndDate – «дата конца периода».

С помощью процедуры GetFinanceAssets выгружаются данные по финансовым потокам за период в один год (2012) с 01.01.2012 по 31.12.2012 г. (рисунок 2.1).

После названия выгружаемого пакета указываем название процедуры «GetFinanceAssets» и параметры в формате соответствующего типа данных, в нашем случае - дата будет иметь формат to_date ('01.01.2012','dd.mm.yyyy'). Потом указываем тип таблиц строк.

```
select * from
table(cast(REP_FIN_FLOW_RESERVE_perenos.GetFinanceAssets
(to_date('01.01.2012','dd.mm.yyyy'),
to_date('31.12.2012','dd.mm.yyyy')) as
TFINANCE_FLOW_RESERVE_T))
```

The screenshot shows the PL/SQL Developer interface. The top part displays the SQL query used to execute the procedure. The bottom part shows the output table with the following data:

RECORD_NO	RECORD_NAME	TOTAL	FINANCE_SECTOR	NATIONAL_BANK	BANK	OTHER_FINANCE_ORG	HOUSEKEEPING
1	Наличная валюта	...					
2	1. Национальная	...					
3	1 объем на начало отчетного периода	441789	441789	441789	0		0
4	2 изменения в результате проведения операций	332213	332213	332213	0		0
5	3 другие изменения (укажите вид)	...					
6	4 объем на конец отчетного периода	774002	774002	774002	0		0
7	2. Иностранная	...					
8	5 объем на начало отчетного периода	9272	0	0	0		0
9	6 изменения в результате проведения операций	5074	0	0	0		0
10	7 оценочные изменения стоимости	381	0	0	0		0
11	8 другие изменения (укажите вид)	...					
12	9 объем на конец отчетного периода	14727	0	0	0		0
13	Переводимые депозиты	...					
14	1. В национальной валюте	...					
15	10 объем на начало отчетного периода	7915073	7915073	7915058	15		0
16	11 изменения в результате проведения операций	-5738759	-5738759	-5738756	-3		0
17	12 оценочные изменения стоимости	0	0	0	0		0
18	13 другие изменения (укажите вид)	...					

Рисунок 2.1 – Выгрузка данных в PL/SQL Developer с помощью процедуры «GetFinanceAssets»

С помощью процедуры «GetFinanceLiabilities» выгружаются данные по финансовым резервам за период в один год (2012) с 01.01.2012 по 31.12.2012 г (рисунок 2.2).

```
select * from
table(cast(REP_FIN_FLOW_RESERVE_perenos.GetFinanceLiabilities
(to_date('01.01.2012','dd.mm.yyyy'),
to_date('31.12.2012','dd.mm.yyyy')) as
TFINANCE_FLOW_RESERVE_T))
```


The screenshot shows the PL/SQL Developer interface. At the top, there is a menu bar (Edit, Session, Debug, Tools, Reports, Deployment, Window, Help) and a toolbar. Below that, the SQL editor contains the following query:

```
select * from table (cast (REP_FIN_FLOW_RESERVE_perenos.GetFinanceLiabilities(to_date('01.01.2012','dd.mm.yyyy'),
to_date('31.12.2012','dd.mm.yyyy') ) as TFINANCE_FLOW_RESERVE_T)
```

Below the SQL editor, the 'Output' tab displays a table with the following data:

RECORD_NO	RECORD_NAME	TOTAL	FINANCE_SECTOR	NATIONAL_BANK	BANK	OTHER_FINANCE_ORG	HOUSEKEEPING
1	Переводимые депозиты	...					
2	1. В национальной валюте	...					
3	объем на начало отчетного периода	642591	0	0	0	0	0
4	изменения в результате проведения операций	739077	0	0	0	0	0
5	оценочные изменения стоимости	0	0	0	0	0	0
6	другие изменения (укажите вид)	...					
7	объем на конец отчетного периода	1381668	0	0	0	0	0
8	116 расходы к оплате	...					
9	2. В иностранной валюте	...					
10	объем на начало отчетного периода	...					
11	изменения в результате проведения операций	...					
12	119 оценочные изменения стоимости	...					
13	120 другие изменения (укажите вид)	...					
14	объем на конец отчетного периода	...					
15	122 расходы к оплате	...					
16	Другие депозиты	...					
17	1. В национальной валюте	...					
18	123 объем на начало отчетного периода	72562377	0	0	0	0	0

Рисунок 2.2 – Выгрузка данных в PL/SQL Developer с помощью процедуры «GetFinanceLiabilities»

2.2.5 Требования и процесс создания пакета для «Отчета об остатках на балансовых счетах за вычетом резервов (провизий)»

Для того чтобы создать модель данных в Oracle Business Intelligence Publisher, необходимо реализовать пакет (приложение А) в Oracle SQL Developer интегрированной среде разработки.

В данном «Отчете об остатках на балансовых счетах за вычетом резервов (провизий)» указываются данные по активам, обязательствам и структуре капитала. Для этого отчета используется один входящий параметр `p_Date`, т. к. отчет обновляется раз в месяц.

В реализации отчета используется курсор для оптимизации самого кода, что бы было достаточно просто обращаться к сбору итогов, не через таблицы, а только к тем данным, которые попали в курсор.

Под курсором в Oracle понимается получаемый при выполнении запроса результирующий набор и связанный с ним указатель текущей записи. Для объявления явного курсора используется оператор `CURSOR`, который, в данном отчете имеет следующее описание:

```
cursor emp_cur is -- Создание курсора
```

Явный курсор объявляется, а неявный курсор не требует объявления.

Для запросов, возвращающих более одной строки, можно использовать только явный курсор.

Курсор может быть объявлен в секциях объявлений любого блока PL/SQL, подпрограммы или пакета.

Оператор `CURSOR` выполняет объявление явного курсора.

Цикл `FOR` с курсором выполняет следующие действия:

- неявно объявляет переменную цикла как запись `%ROWTYPE`;

- открывает курсор;
 - при каждой итерации извлекает следующую строку из результирующего набора в поля неявно объявленной записи;
 - по достижении конца результирующего набора закрывает курсор.
- Сбор итогов с помощью курсора:

```
for rec in emp_cur loop
  if trim(rec.column_num) in ('3.1') then
    val_1 := val_1 + rec.val;
  end if;
  if rec.column_num in ('4.1') then
    val_2 := val_2 + rec.val;
  end if;
  if rec.column_num in ('11.2') then
    val_3 := val_3 + rec.val;
  end if;
  ...
end loop;
```

Строка «Активы» – итог по строкам 14,15,18,19,20.

```
if rec.column_num in ('14','15','18','19','20') then --
АКТИВЫ
  val_13 := val_13 + rec.val;
end if;
```

Корреспондентские счета (с учетом начисленного вознаграждения и за вычетом резервов (провизий) – итог по строке 3.1.

```
if trim(rec.column_num) in ('3.1') then --3
  val_1 := val_1 + rec.val;
end if;
```

Вклады, размещенные в других банках – итог по строкам 4.1.

```
if rec.column_num in ('4.1') then --4
  val_2 := val_2 + rec.val;
end if;
```

Займы, предоставленные физическим лицам (за вычетом резервов (провизий)) – итог по строкам 11.2.

```
if rec.column_num in ('11.2') then --11
  val_3 := val_3 + rec.val; end if;
```

Вклады, привлеченные от физических лиц – итог по строкам 28.1, 28.4.

```
if rec.column_num in ('28.1.1') then --28.1
```

```

    val_5 := val_5 + rec.val;
end if;
if rec.column_num in ('28.4.1', '28.4.2') then --28.4
    val_6 := val_6 + rec.val; end if;

```

Итого обязательства, влекущие расход – итог по строкам 30, 31.

```

if rec.column_num in ('30', '31') then --35
    val_7 := val_7 + rec.val; end if;

```

Собственный капитал – итог по строкам 40,41, 46, 44.

```

if rec.column_num in ('40', '41', '46', '44') then --39
    val_8 := val_8 + rec.val; end if;

```

Нераспределенный чистый доход и резерв капитала – итог по строкам 42.1,42.2,42.3.

```

if rec.column_num in ('42.1','42.2','42.3') then --42
    val_12 := val_12 + rec.val; end if;

```

В данном запросе (рисунок 2.3) выводим код (ID), ТПС (CODE_POKAZ) и наименование (NAME_POKAZ). Создаем связь по полям CALCULATION_ID из таблицы SET_REP_COLUMN_CALCULATION, и по ID из GL_700_D (ГК). При выполнении запроса выводиться ТПС и его название, для получения данных по активам, обязательствам и структуре капитала (таблица 2.16).

```

select distinct t.calculation_id, d.code_pokaz, d.name_pokaz
from set_rep_column_calculation t,
gl_700_d d
where t.calculation_id = d.id
and d.code_pokaz in ('1100', '1051',
'1052', '1250', '1450', '1750', '1461', '1400',
'1740', '1741', '1100', '1051', '1052',
'1250', '1450', '1750', '1461', '1000',
'1651', '1652', '1653', '1654', '1658',
'1659', '1660', '1850', '1790', '1710',
'1725', '1746', '1748', '1602', '1857',
'1746', '1725', '1710', '1748', '1790',
'1428', '1451', '1692', '1693', '1694',
'1698', '1699', '2204', '2207191', '223891',
'2207291', '2238291', '2213', '2036', '2703',
'2770', '2850', '2703', '3001', '3101',
'3580', '3599', '3510', '3561', '3400')

```

SQL Output Statistics

```

select distinct t.calculation_id, d.code_pokaz, d.name_pokaz
from set_rep_column_calculation t, gl_700_d d
where t.calculation_id = d.id
and d.code_pokaz in ('1100','1051','1052','1250','1450','1750',
                    '1451','1450','1740','1741','1100','1051','1052')

```

	CALCULATION_ID	CODE_POKAZ	NAME_POKAZ
1	43	1725	Начисленные доходы по вкладам, размещенным в других банках
2	663	3561	Резервы переоценки стоимости финансовых активов, имеющихся в
3	690	1052	Корреспондентские счета в других банках
4	703	1750	Просроченное вознаграждение по ценным бумагам
5	33	1653	Компьютерное оборудование
6	45	1741	Просроченное вознаграждение по займам и финансовому лизингу,
7	377	1850	Прочие дебиторы
8	390	3101	Дополнительный оплаченный капитал
9	569	1428	Резервы (проевизии) по займам и финансовому лизингу, предоставле
10	712	1857	Отложенные налоговые активы
11	44	1740	Начисленные доходы по займам и финансовому лизингу, предостав
12	215	1698	Начисленная амортизация по транспортным средствам
13	366	1100	Требования к Национальному Банку Республики Казахстан
14	451	1790	Предоплата вознаграждения и расходов
15	509	3001	Уставный капитал – простые акции
16	561	1660	Создаваемые (разрабатываемые) нематериальные активы
17	695	2703	Начисленные расходы по займам, полученным от Правительства Ре
18	629	2204	Текущие счета физических лиц
19	34	1654	Прочие основные средства
20	214	1658	Транспортные средства

Рисунок 2.3 – Выгрузка из настроечных таблиц

Т а б л и ц а 2.16 – Описание ТПС

ID	ТПС	Название
43	1725	Начисленные доходы по вкладам, размещенным в других банках
663	3561	Резервы переоценки стоимости финансовых активов, имеющихся в наличии для продажи
690	1052	Корреспондентские счета в других банках
703	1750	Просроченное вознаграждение по ценным бумагам
33	1653	Компьютерное оборудование
45	1741	Просроченное вознаграждение по займам и финансовому лизингу, предоставленным клиентам
377	1850	Прочие дебиторы
390	3101	Дополнительный оплаченный капитал
569	1428	Резервы (проевизии) по займам и финансовому лизингу, предоставленным клиентам
712	1857	Отложенные налоговые активы
44	1740	Начисленные доходы по займам и финансовому лизингу, предоставленным клиентам
215	1698	Начисленная амортизация по транспортным средствам
366	1100	Требования к Национальному Банку Республики Казахстан
451	1790	Предоплата вознаграждения и расходов
509	3001	Уставный капитал – простые акции
561	1660	Создаваемые (разрабатываемые) нематериальные активы
695	2703	Начисленные расходы по займам, полученным от Правительства Республики Казахстан, местных исполнительных органов Республики Казахстан и национального управляющего холдинга

629	2204	Текущие счета физических лиц
34	1654	Прочие основные средства
214	1658	Транспортные средства
376	1710	Начисленные доходы по вкладам, размещенным в Национальном Банке Республики Казахстан
449	1748	Начисленные доходы по операциям «обратное РЕПО» с ценными бумагами
699	1451	Резервы (провизии) на покрытие убытков по ценным бумагам, имеющимся в наличии для продажи
636	2213	Вклад, являющийся обеспечением обязательств физических лиц
31	1651	Строящиеся (устанавливаемые) основные средства
37	1692	Начисленная амортизация по зданиям и сооружениям
40	1699	Начисленная амортизация по нематериальным активам
369	1250	Вклады, размещенные в других банках
392	3510	Резервный капитал
578	1450	Ценные бумаги, имеющиеся в наличии для продажи
584	3599	Нераспределенная чистая прибыль (непокрытый убыток)
662	3580	Нераспределенная чистая прибыль (непокрытый убыток) прошлых лет
694	2036	Долгосрочные займы, полученные от Правительства Республики Казахстан, местных исполнительных органов Республики Казахстан и национального управляющего холдинга
9	1051	Корреспондентский счет в Национальном Банке Республики Казахстан
32	1652	Земля, здания и сооружения
371	1400	Требования к клиентам
702	1461	Операции «обратное РЕПО» с ценными бумагами
38	1693	Начисленная амортизация по компьютерному оборудованию

Окончание таблицы 2.16

ID	ТПС	Название
39	1694	Начисленная амортизация по прочим основным средствам
384	2850	Прочие кредиторы
448	1746	Начисленные доходы по ценным бумагам, имеющимся в наличии для продажи
29	1602	Прочие товарно-материальные запасы
36	1659	Нематериальные активы
364	1000	Деньги

Создание пакета REP_AG006 с функцией GET_REPORT:

```
create or replace package REP_AG006 is function
get_report(p_date date) return T_AG006_TAB end REP_AG006;
```

После создания пакета, производится выгрузка коллекции инструментами PL/SQL Developer, для контроля данных и формирования выгружаемого шаблона. Для выгрузки коллекции (рисунок 2.4) используем код:

```
select * from
table(cast(rep_ag006.get_report(to_date('31.01.2014',
'DD.MM.YYYY')) as T_AG006_TAB ))
```

	ADATE	POS	NAME	A2
1	31.01.14		Активы	369795393
2	31.01.14		Вклады в Национальном Банке Республики Казахстан	11365053
3	31.01.14	3	Корреспондентские счета (с учетом начисленного вознаграждения и за вычетом резервов (р...	14
4	31.01.14	3.1	– у резидентов Республики Казахстан (за вычетом резервов (провизий))	14
5	31.01.14	3.2	– у нерезидентов Республики Казахстан (за вычетом резервов (провизий))	0
6	31.01.14	4	Вклады, размещенные в других банках (за вычетом резервов (провизий)), в том числе	13116506
7	31.01.14	4.1	– у резидентов Республики Казахстан (за вычетом резервов (провизий))	13116506
8	31.01.14	4.2	– у нерезидентов Республики Казахстан (за вычетом резервов (провизий))	0
9	31.01.14	5	Займы, предоставленные банкам и организациям, осуществляющим отдельные виды банк...	0
10	31.01.14	5.1	– резидентам Республики Казахстан (за вычетом резервов (провизий))	0
11	31.01.14	5.2	– нерезидентам Республики Казахстан (за вычетом резервов (провизий))	0
12	31.01.14	6	Ценные бумаги, учитываемые по справедливой стоимости через прибыль или убыток (за вы...	0
13	31.01.14	7	Ценные бумаги, имеющиеся в наличии для продажи (за вычетом резервов (провизий))	144365420
14	31.01.14	8	Ценные бумаги, удерживаемые до погашения (за вычетом резервов (провизий))	0
15	31.01.14	9	Операции "Обратное РЕПО" с ценными бумагами (за вычетом резервов (провизий))	47642040
16	31.01.14	10	Займы, предоставленные юридическим лицам (за вычетом резервов (провизий))	0
17	31.01.14	11	Займы, предоставленные физическим лицам (за вычетом резервов (провизий)), в том числе...	145317495
18	31.01.14	11.1	на потребительские цели (за вычетом резервов (провизий))	0
19	31.01.14	11.2	на строительство, покупку и (или) ремонт жилья (за вычетом резервов (провизий))	145317495
20	31.01.14	11.3	прочие займы (за вычетом резервов (провизий))	0
21	31.01.14	12	Инвестиции в капитал (за вычетом резервов (провизий))	0
22	31.01.14	13	Субординированный долг	0
23	31.01.14	14	Итого активы, приносящие доход	361806528

Рисунок 2.4 – Выгрузка данных в PL/SQL Developer REP_AG006

2.2.6 Требования и процесс создания пакета для отчета «Основные показатели баланса»

Для того чтобы создать модель данных в Oracle Business Intelligence Publisher, необходимо реализовать пакет (приложение А) в Oracle SQL Developer интегрированной среде разработки.

В данном отчете «Основные показатели баланса» указываются данные по бухгалтерскому балансу. Периодичность обновления 1 раз в месяц. Данный отчет используется для предоставления лицам Правления Банка.

```
create or replace package body REP_AG041 is
function get_report(p_date in date) return T_AG041_TAB is
```

Для этого отчета используется один входящий параметр P_DATE, т. к. отчет обновляется раз в месяц.

В реализации отчета используется курсор для оптимизации кода.

```
cursor emp_cur is -- Создание курсора
```

Все суммы указываются в тысячах тенге, для приведение сумм к тыс. тенге используется следующий алгоритм: все значения в отчете делятся на тысячу, после чего к ним применяют правила округления.

```
select round(sum(gf.output) / 1000, 0)
```

```

from gl_acc_f          gf,
     gl_account_d      gl,
     gl_plan_account_d pl
     where gf.gl_account_id = gl.id
           and gf.work_date = p_date

```

НДС в зачет (счет 1851 НДС):

```

select round(sum(gf.output) / 1000, 0)
from gl_acc_f          gf,
     gl_account_d      gl,
     gl_plan_account_d pl
     where gf.gl_account_id = gl.id
           and gf.work_date = p_date
           and gl.name like '%НДС%'
           and pl.id = gf.gl_plan_account_id
           and pl.code = '1851'

```

Начисленный НДС (счет 2851):

```

select round(sum(gf.output) / 1000, 0)
from gl_acc_f          gf,
     gl_account_d      gl,
     gl_plan_account_d pl
     where gf.gl_account_id = gl.id
           and gf.work_date = p_date
           and gl.name like '%НДС%'
           and pl.id = gf.gl_plan_account_id
           and pl.code = '2851'

```

Доходы по амортизации дисконта по вкладу, размещенному АО «Банк Центр Кредит», АО «БТА Банк», АО «Цеснабанк» (счет 4266):

```

select round(sum(gf.output) / 1000, 0)
from gl_acc_f          gf,
     gl_account_d      gl,
     gl_plan_account_d pl
     where gf.gl_account_id = gl.id
           and gf.work_date = p_date
           and gl.name like '%Центр%Кредит%'
           and pl.id = gf.gl_plan_account_id
           and pl.code = '4266'

```

Доходы по амортизации дисконта по займам, предоставленным клиентам (счет 4434):

```

select round(sum(gf.output) / 1000, 0)
from gl_acc_f          gf,
     gl_account_d      gl,
     gl_plan_account_d pl
     where gf.gl_account_id = gl.id
           and gf.work_date = p_date
           and gl.name not like '%(CC)%'

```

```

and pl.id = gf.gl_plan_account_id
and pl.code = '4434'

```

Самортизированный расход по займам, выданным за счет займа от АО «ФНБ Самрук-Казына», АО «Цеснабанк», по Госпрограмме за счет бюджетных средств из уставного капитала, в том числе по эффективной ставке (счет 5921):

```

select round(sum(gf.output) / 1000, 0)
  from gl_acc_f          gf,
       gl_account_d     gl,
       gl_plan_account_d pl
  where gf.gl_account_id = gl.id
        and gf.work_date = p_date
        and gl.name like '%Самрук%'
        and pl.id = gf.gl_plan_account_id
        and pl.code = '5921'

```

Итоги собираться с помощью цикла, который проходит по всем значениям, которые попали в курсор.

```

for rec in emp_cur loop
  my_row := new T_AG041_ROW(p_date,
                           rec.column_num,
                           rec.parent_column,
                           rec.column_name,
                           rec.val);

  my_res.extend;
  my_res(my_res.count) := my_row;
end loop; -- цикл закрывается

```

После того, как итоги были собраны, все значения возвращаются.

```

return my_res;

```

В данном запросе указываются номер счета (рисунок 2.5) для получения данных по активам, обязательствам и структуре капитала. Номер счета берётся из настроечных таблиц (таблица 2.17).

```

select distinct t.calculation_id, d.code_pokaz, d.name_pokaz
  from set_rep_column_calculation t, gl_700_d d
  where t.calculation_id = d.id
        and d.code_pokaz in ('1051', '1103', '1428',
                              1434', '1451', '1265', '1851', '1480', '2036',
                              '2070', '2238', '2851', '3001', '3100', '3400',
                              '3510', '3561', '3580', '3599', '4200', '4450',
                              '4465', '4510', '4709', '4710', '4730', '4480',
                              '4266', '4440', '4434', '5060', '5921')

```

Т а б л и ц а 2.17 – Описание ТПС

ID	ТПС	Название
471	4730	Реализованные доходы от переоценки
651	4480	Доходы, связанные с получением вознаграждения по ценным бумагам, удерживаемым до погашения
524	1434	Дисконт по займам, предоставленным клиентам
527	2070	Дисконт по полученным займам
663	3561	Резервы переоценки стоимости финансовых активов, имеющих в наличии для продажи
516	4710	Нереализованный доход от прочей переоценки
569	1428	Резервы (провизии) по займам и финансовому лизингу, предоставленным клиентам
529	2238	Премия по вкладам, привлеченным от клиентов
585	4450	Доходы, связанные с получением вознаграждения по ценным бумагам, имеющимся в наличии для продажи
509	3001	Уставный капитал – простые акции
80	3100	Дополнительный капитал
692	4200	Доходы, связанные с получением вознаграждения по ценным бумагам, учитываемым по справедливой стоимости через прибыль или убыток
677	4709	Нереализованный доход от изменения стоимости ценных бумаг, учитываемых по справедливой стоимости через прибыль или убыток
699	1451	Резервы (провизии) на покрытие убытков по ценным бумагам, имеющимся в наличии для продажи
523	1265	Дисконт по вкладам, размещенным в других банках
392	3510	Резервный капитал
9	1051	Корреспондентский счет в Национальном Банке Республики Казахстан

Окончание таблицы 2.17

ID	ТПС	Название
694	2036	Долгосрочные займы, полученные от Правительства Республики Казахстан, местных исполнительных органов Республики Казахстан и национального управляющего холдинга
662	3580	Нераспределенная чистая прибыль (непокрытый убыток) прошлых лет
584	3599	Нераспределенная чистая прибыль (непокрытый убыток)
367	1103	Срочные вклады в Национальном Банке Республики Казахстан
102	4510	Доходы по купле-продаже ценных бумаг
535	4440	Доходы по амортизации премии по вкладам, привлеченным от клиентов

SQL		Output	Statistics
<pre>select distinct t.calculation_id, d.code_pokaz, d.name_pokaz from set_rep_column_calculation t, gl_700_d d where t.calculation_id = d.id and d.code_pokaz in ('1051','1103', '1428','1434','1451','1265', '1851','1480','2036', '2070', '2238','2851', '3001','3100','3400', '3510','3561', '3580', '3599', '4200', '4450', '4465', '4510', '4709','4710','4730', '4480','4266','4440', '4434', '5060', '5921')</pre>			
	CALCULATION_ID	CODE_POKAZ	NAME_POKAZ
1	471	4730	Реализованные доходы от переоценки
2	651	4480	Доходы, связанные с получением вознаграждения по ценным бумагам
3	524	1434	Дисконт по займам, предоставленным клиентам
4	527	2070	Дисконт по полученным займам
5	663	3561	Резервы переоценки стоимости финансовых активов, имеющих в
6	516	4710	Нереализованный доход от прочей переоценки
7	569	1428	Резервы (провизии) по займам и финансовому лизингу, предоставл
8	529	2238	Премия по вкладам, привлеченным от клиентов
9	585	4450	Доходы, связанные с получением вознаграждения по ценным бумагам
10	509	3001	Уставный капитал – простые акции
11	80	3100	Дополнительный капитал
12	692	4200	Доходы, связанные с получением вознаграждения по ценным бумагам
13	677	4709	Нереализованный доход от изменения стоимости ценных бумаг, учи
14	699	1451	Резервы (провизии) на покрытие убытков по ценным бумагам, имен
15	523	1265	Дисконт по вкладам, размещенным в других банках
16	392	3510	Резервный капитал
17	9	1051	Корреспондентский счет в Национальном Банке Республики Казахст
18	694	2036	Долгосрочные займы, полученные от Правительства Республики Ка
19	662	3580	Нераспределенная чистая прибыль (непокрытый убыток) прошлых

Рисунок 2.5 – Выгрузка из настроечных таблиц

После создания пакета, производится выгрузка коллекции инструментами PL/SQL Developer, для контроля данных и формирования выгружаемого шаблона. С помощью процедуры «GET_REPORT» выгружаются данные основным показателям баланса, период на конец месяца.

Для выгрузки коллекции (рисунок 2.6) используем код:

```
select * from table(cast(rep_ag041.get_report(to_date
('31.01.2014', 'DD.MM.YYYY')) as T_AG041_TAB ))
```

SQL Output Statistics

```
select * from table(cast(rep_ag041.get_report(to_date('31.01.2014', 'DD.MM.YYYY')) as T_AGO41_TAB ))
```

	ADATE	N	P_NAME	NAME	SALDO
1	31.01.14			ИТОГО, в том числе:	369795393
2	31.01.14	1	Раздел I Активы	Корр.счет в НБРК	2365053
3	31.01.14	2	Раздел I Активы	Срочные вклады в Национальном Банке РК	9000000
4	31.01.14	3	Раздел I Активы	Требования к клиентам, в том числе:	144820803
5	31.01.14	4	Раздел I Активы	1) Резервы (провизии) по займам и финансовому лизингу, предоста	-2117220
6	31.01.14	5	Раздел I Активы	2) Дисконт по займам, предоставленным клиентам	-3558666
7	31.01.14	6	Раздел I Активы	Ценные бумаги по рыночной стоимости, в том числе:	144365420
8	31.01.14	7	Раздел I Активы	1) Резервы (провизии) на покрытие убытков по ценным бумагам	-1592460
9	31.01.14	8	Раздел I Активы	Межбанковские депозиты, в том числе:	13116506
10	31.01.14	9	Раздел I Активы	1) Дисконт по вкладам, размещенным в других банках	-198922
11	31.01.14	10	Раздел I Активы	НДС в зачет	6519
12	31.01.14	11	Раздел I Активы	Ценные бумаги удерживаемые до погашения	0
13	31.01.14			ИТОГО, в том числе:	270800473
14	31.01.14	12	Раздел II Обязательства	Долгосрочные займы, полученные от Правительства РК и местных и	69200000
15	31.01.14	13	Раздел II Обязательства	Дисконт по полученным займам от АО "ФНБ Самрук-Казына"и Пра	-10325360
16	31.01.14	14	Раздел II Обязательства	Обязательства перед клиентами, в том числе:	208748115
17	31.01.14	15	Раздел II Обязательства	Текущие счета физических лиц	1906397
18	31.01.14	16	Раздел II Обязательства	1) Премия по вкладам, привлеченным от клиентов	959332
19	31.01.14	17	Раздел II Обязательства	Начисленный НДС	4845
20	31.01.14			ИТОГО, в том числе:	98994920
21	31.01.14	18	Раздел III Собственный капитал	Уставный капитал - простые акции	78300000
22	31.01.14	19	Раздел III Собственный капитал	Дополнительный капитал	12704693
23	31.01.14	20	Раздел III Собственный капитал	Динамические резервы	1106269
24	31.01.14	21	Раздел III Собственный капитал	Резервный капитал	2283335
25	31.01.14	22	Раздел III Собственный капитал	Резервы переоценки стоимости финансовых активов, имеющихс	-1316500

Рисунок 2.6 – Выгрузка данных в PL/SQL Developer REP_AG041

2.3 Создание модели данных для отчетов

Пункт меню «Создать» содержит основной функционал для работы с Oracle Business Intelligence с отчетами. Для создания модели данных во вкладке «Создать» необходимо выбрать «Модель данных» (рисунок 2.7).

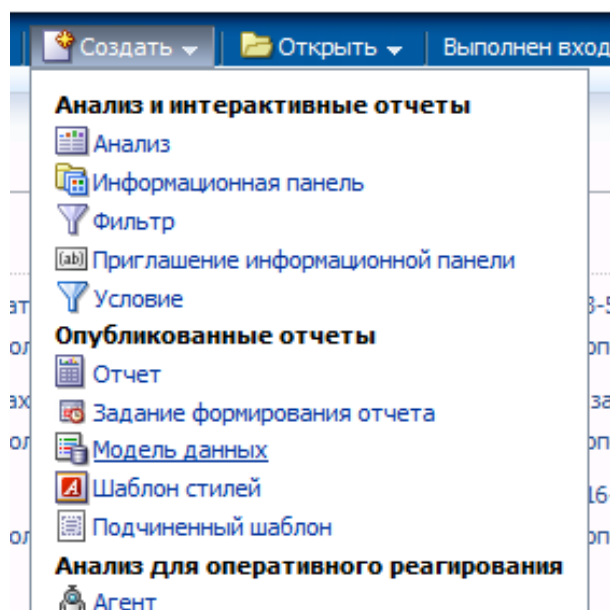


Рисунок 2.7 – Создание модели данных

При выборе «Модель данных» откроется форма новой модели данных (рисунок 2.8), где необходимо создать параметры и sql-запросы для выгрузки коллекций через BI Publisher.

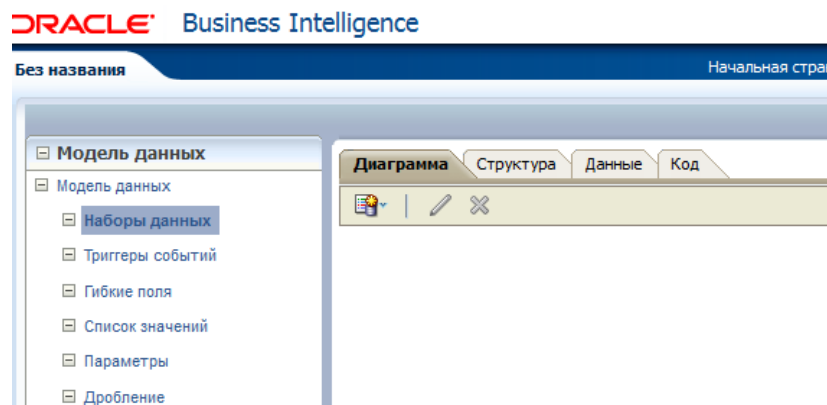


Рисунок 2.8 – Форма модели данных

Модель данных включает в себя «Наборы данных», «Триггеры событий», «Гибкие поля», «Список значений», «Параметры», «Дробление». «Наборы данных» - это запрос (или несколько запросов) по выгрузки коллекций. «Список значений» – включает в себя запросы и/или списки значений, формирования параметров. «Параметры» – это те данные, от которых зависит выгрузка той или иной коллекции. Можно создавать, как параметры с ручным вводом, можно создавать параметры с «списком значений», что позволяет избегать не корректных вводимых параметров пользователем.

Для создания параметров необходимо зайти во вкладку «Параметры». Нажать на «зелёный плюс» и добавить: название, тип данных, значение по умолчанию, тип параметра, размещение строк и задать порядок (рисунок 2.9).

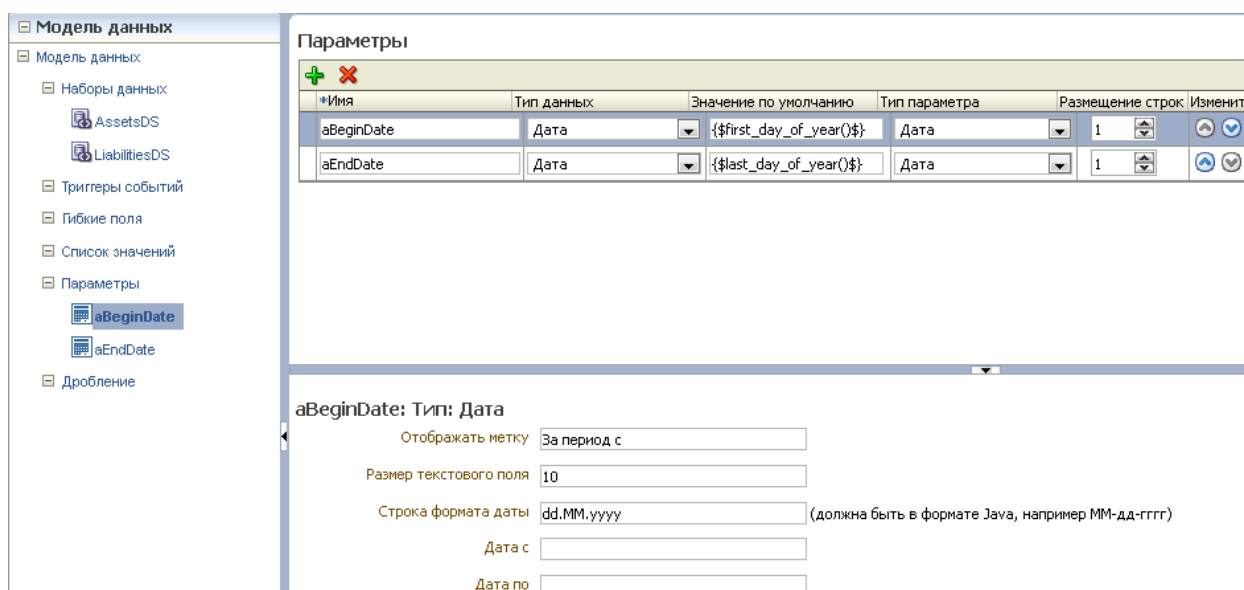


Рисунок 2.9 – Параметры модели данных

В нашем случае создается два входящих параметра `aBeginDate` с типом данных – дата и `aEndDate` с типом данных – дата, т. к. при выгрузке пакета мы задаем период в формате даты. Строка формата даты должна быть в формате Java `dd.MM.yyyy`.

Чтобы создать вызов коллекции в BI Publisher необходимо войти во вкладку «Наборы данных» и на панели выбрать «Создать новый запрос» -> «Запрос SQL» (рисунок 2.10).

Окно с запросом будет содержать имя набора данных, источник данных и тип SQL (по умолчанию «Standard SQL»).

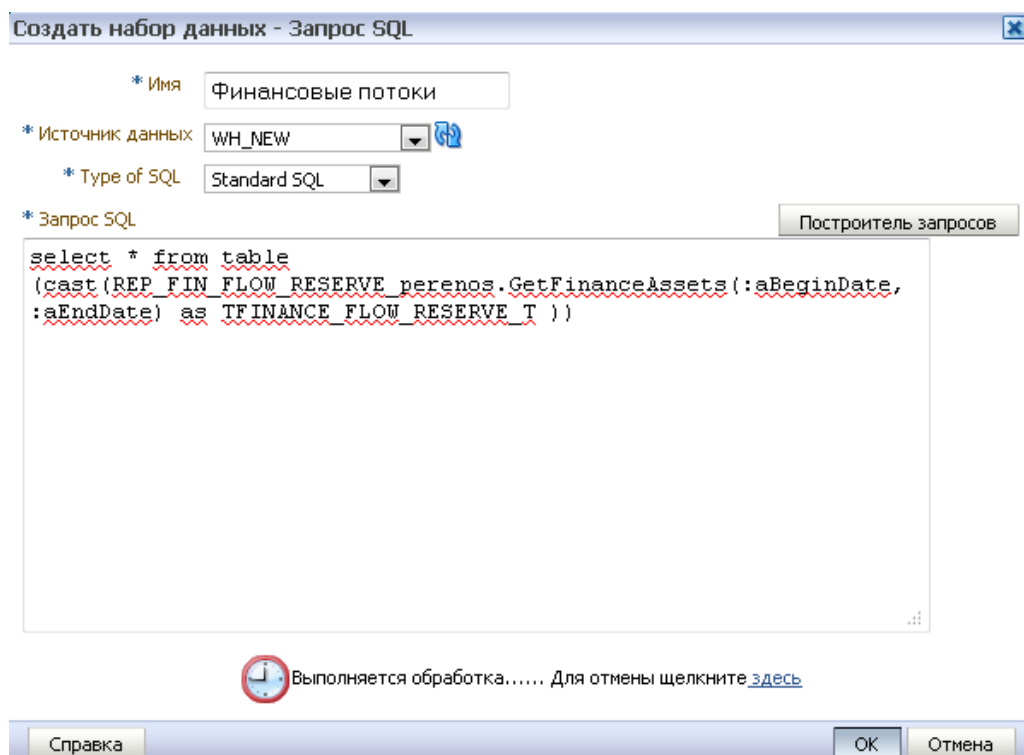


Рисунок 2.10 – Создание запроса «Финансовые потоки»

В поле «Запрос» вводим код с помощью которого выгружается первая коллекция с помощью функции «GetFinanceAssets» в PL/SQL Developer:

```
select * from table
(cast(REP_FIN_FLOW_RESERVE_perenos.GetFinanceAssets
(:aBeginDate, :aEndDate) as TFINANCE_FLOW_RESERVE_T ))
```

В поле «Источник данных» - выбираем базу данных, в которой создана данная коллекция.

Далее добавляем еще набор данных и в поле «Запрос» вводим код с помощью которого выгружается вторая коллекция с помощью функции «GetFinanceLiabilities» в PL/SQL Developer (рисунок 2.11):

```
select * from table
(cast(REP_FIN_FLOW_RESERVE_perenos.GetFinanceLiabilities
(:aBeginDate, :aEndDate) as TFINANCE_FLOW_RESERVE_T))
```

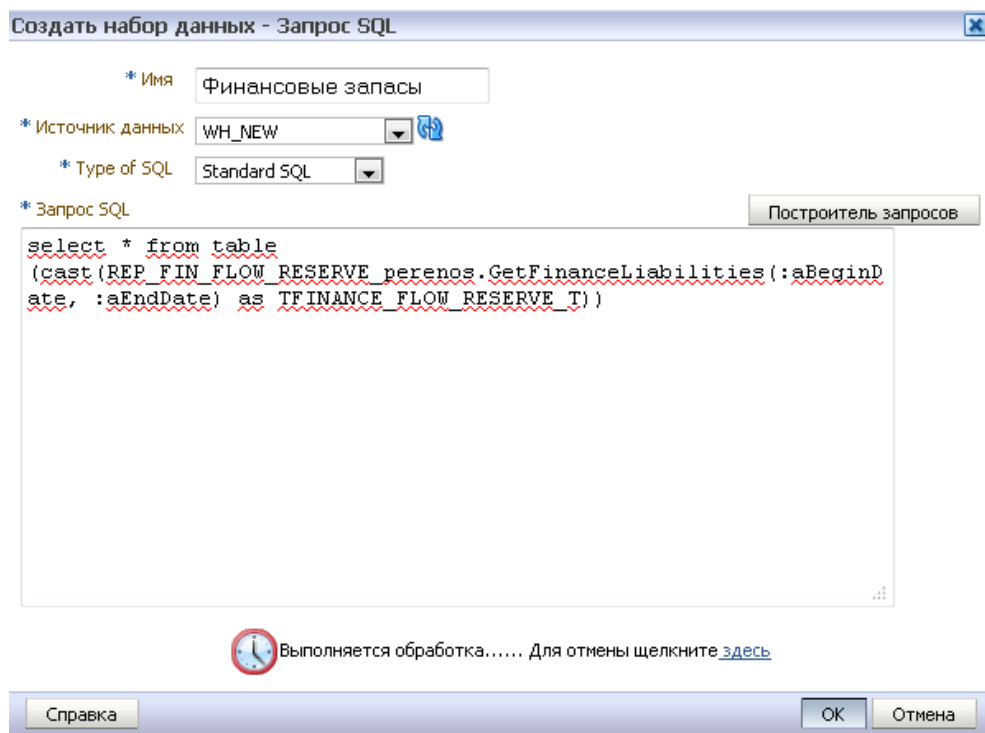


Рисунок 2.11 – Создание запроса «Финансовые запасы»

После создания наборов данных появится форма всех наборов (диаграмма) (рисунок 2.12), с уникальным наименованием.

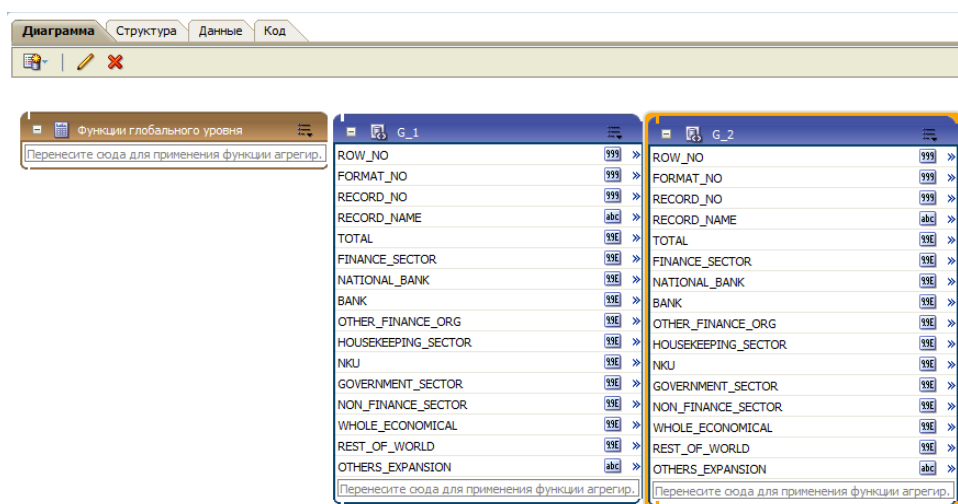


Рисунок 2.12 – Диаграмма набора данных

После того, как запросы сохранены в модели данных, необходимо сделать выгрузку общего набора данных в формате XML. Это делается

переходом на панель «Данные», задаем дату начала отчетного периода и дату конца отчетного периода, и нажимаем на кнопку «Представление» (рисунок 2.13), и после выгрузки представления нажимаем «Сохранить как образец данных».

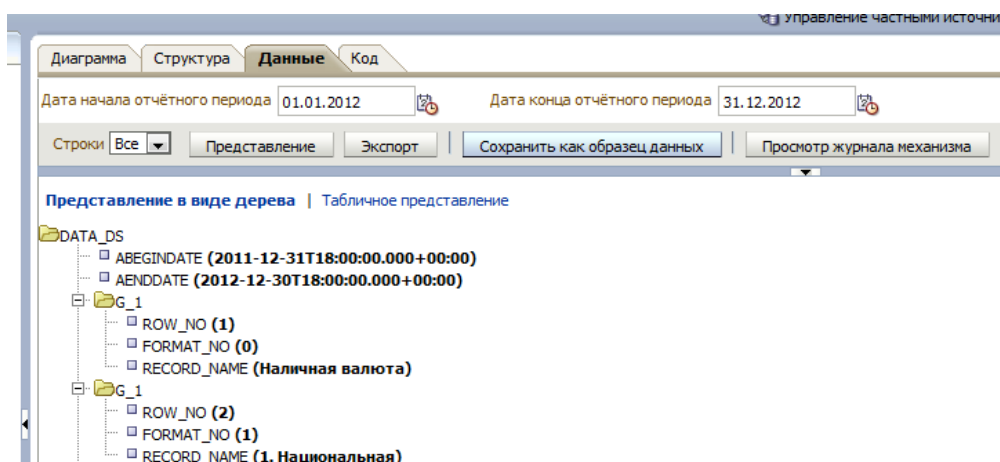


Рисунок 2.13 – Представление в виде дерева

Далее переходим в самую «Модель данных», где указаны ее свойства (рисунок 2.14):

- краткое описание;
- стандартный источник данных;
- пакет по умолчанию БД Oracle;
- рез. копия из БД;
- параметры выходящих данных XML (включаем тэги параметров);
- отображение тэгов XML (выбираем верхний регистр).

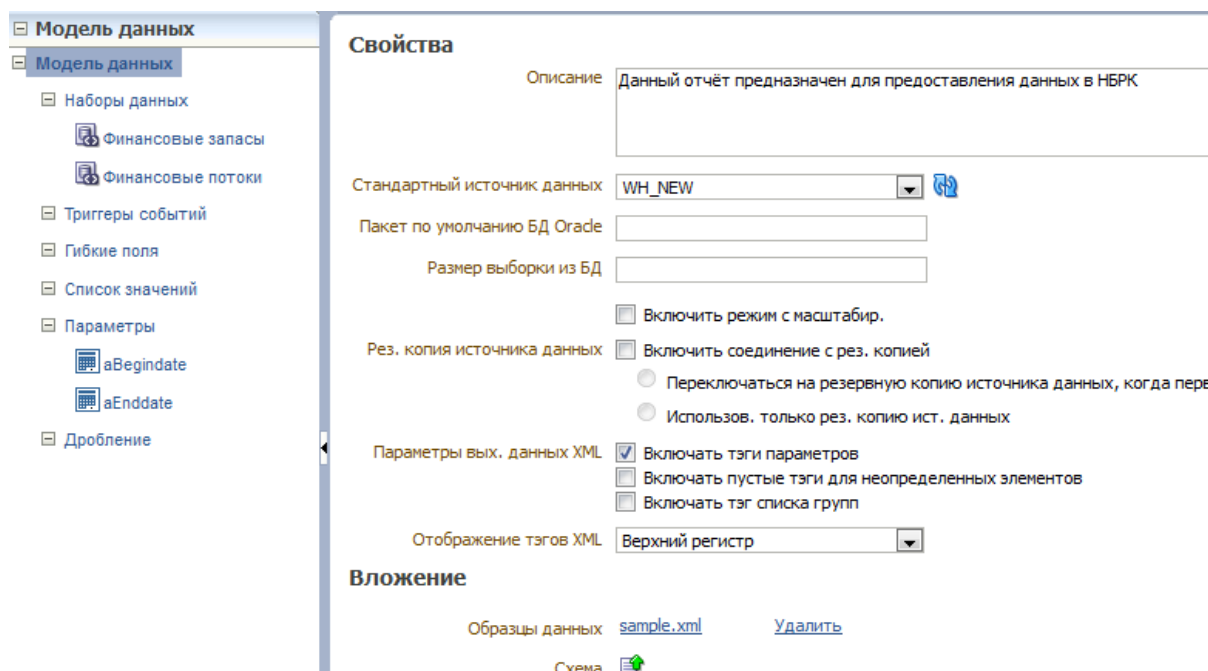


Рисунок 2.14 – Свойства модели данных

Во «Вложении», где находится образец данных с названием файла, необходимо сохранить файл себе на компьютер, для того что бы потом привязать к шаблону созданному с помощью MS Office (рисунок 2.15).

С этим XML-файлом не связана ни одна таблица стилей. Ниже показано дерево элементов.

```
-<!--
  Generated by Oracle BI Publisher 11.1.1.7.140225 -Dataengine, datamodel:/-elstal_borlas/_temp/30e1cd92-62e5-498b-b2ff-149954482d66.xdm
-->
-<DATA_DS>
  <ABEGINDATE>2011-12-31T18:00:00.000+00:00</ABEGINDATE>
  <AENDDATE>2012-12-30T18:00:00.000+00:00</AENDDATE>
  -<G_1>
    <ROW_NO>1</ROW_NO>
    <FORMAT_NO>0</FORMAT_NO>
    <RECORD_NAME>Наличная валюта</RECORD_NAME>
  </G_1>
  -<G_1>
    <ROW_NO>2</ROW_NO>
    <FORMAT_NO>1</FORMAT_NO>
    <RECORD_NAME>1. Национальная</RECORD_NAME>
  </G_1>
  -<G_1>
    <ROW_NO>3</ROW_NO>
    <FORMAT_NO>2</FORMAT_NO>
    <RECORD_NO>1</RECORD_NO>
    <RECORD_NAME>объем на начало отчетного периода</RECORD_NAME>
```

Рисунок 2.15 – Выгруженный XML файл

2.4 Создание отчета в среде Oracle Business Intelligence Publisher

После создания модели данных, можно перейти к созданию самого «Отчета о финансовых потоках и запасах». Для создания отчета будем использовать существующую модель данных (рисунок 2.16).

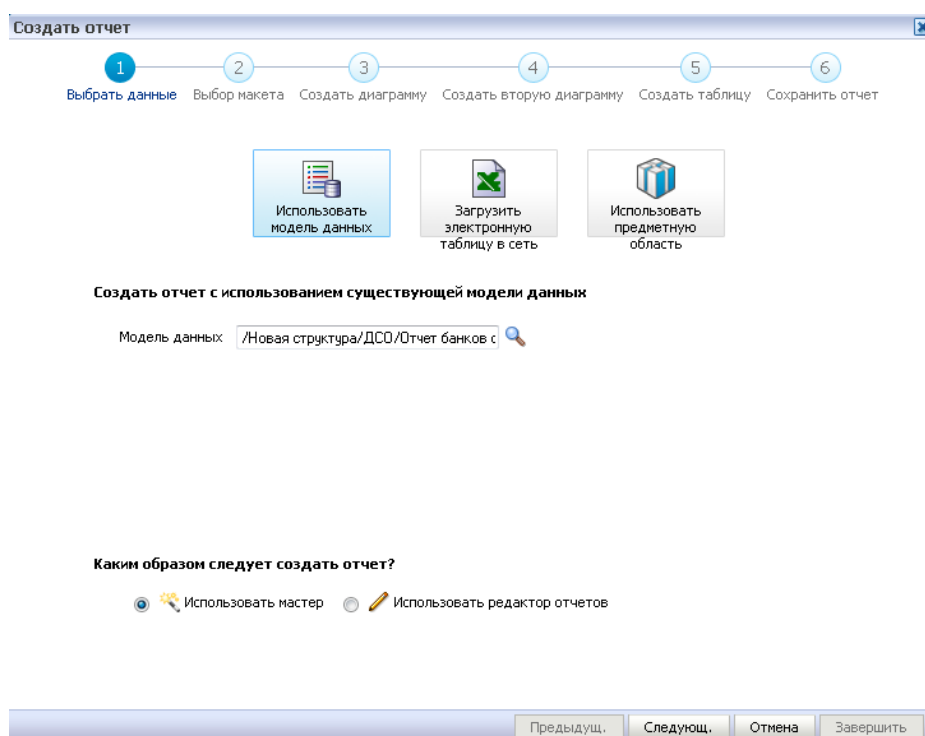


Рисунок 2.16 – Первый шаг для создания отчета

Для того что бы создать таблицу, нужные поля перетащим из источника данных для создания сводной таблице (рисунок 2.17), и можно будет посмотреть образец данных. Также можно отобразить общие итоги.

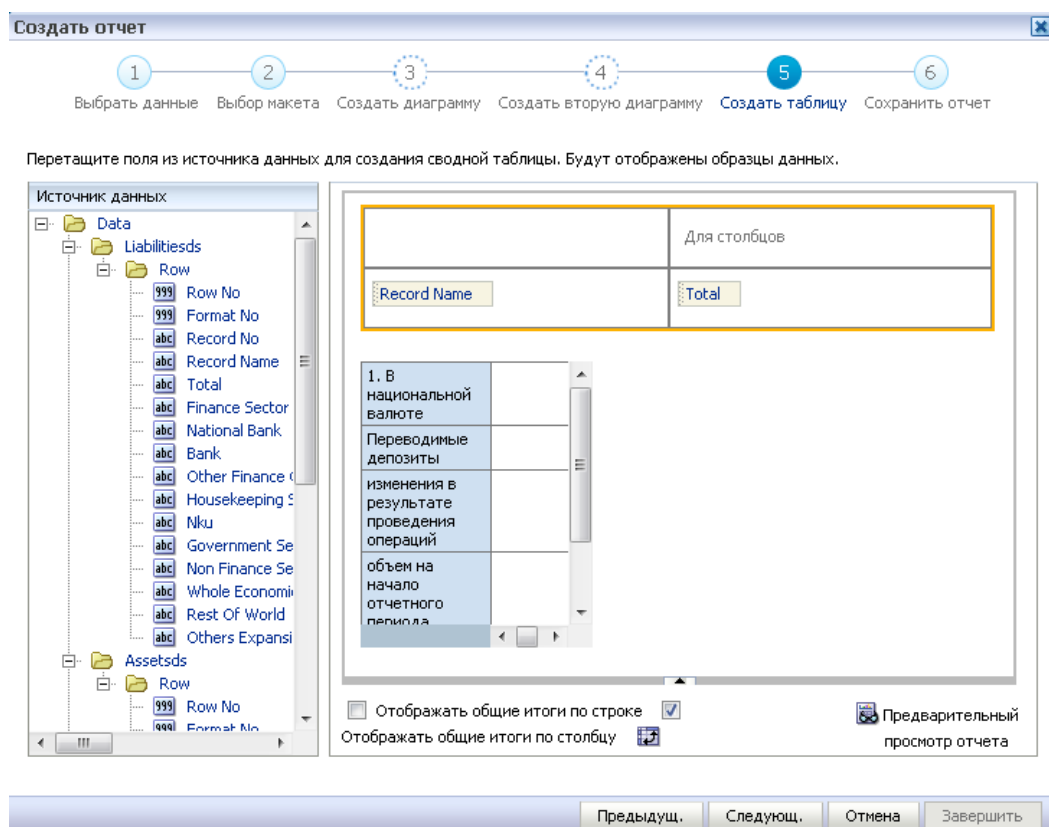


Рисунок 2.17– Создание сводной таблицы

Существует еще один способ создания отчета, для этого необходимо создать шаблон средствами MS Office (MS Word, MS Excel) и загрузить его в сеть (рисунок 2.18) и прикрепить к нему существующую модель данных.

Шаблон выглядит как обычный документ Microsoft Word и Excel, в которых кроме текста хранится информация о конфигурации панелей инструментов и параметры, определяющие вид документа и поведение программы. Настроив шаблон, можно создавать на его основе файлы, которые наследуют имеющиеся в шаблоне текст и объекты, стили и оформление.

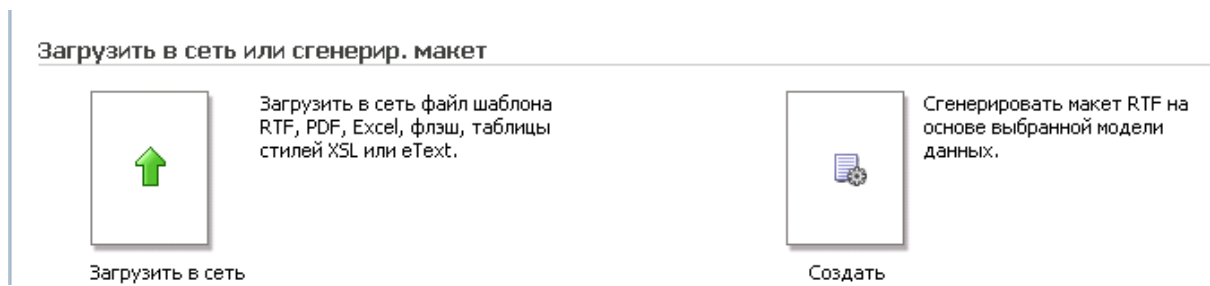


Рисунок 2.18 – Загрузка в сеть файла шаблона

Указываем имя макета, загружаем файл шаблона, выбираем тип шаблона и региональные настройки (рисунок 2.19).

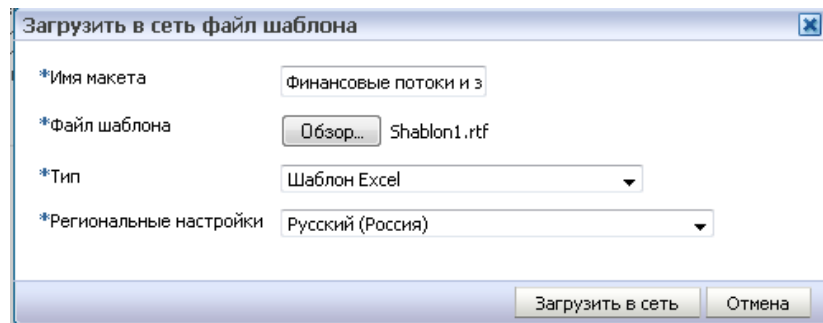


Рисунок 2.19 – Загрузка в сеть файл шаблона

2.5 Выгрузка отчета средствами Oracle Business Intelligence Publisher

После создания модели данных и самого «Отчета о финансовых потоках и запасах», в папке содержатся два файла с моделью данных и самим отчетом (рисунок 2.20).

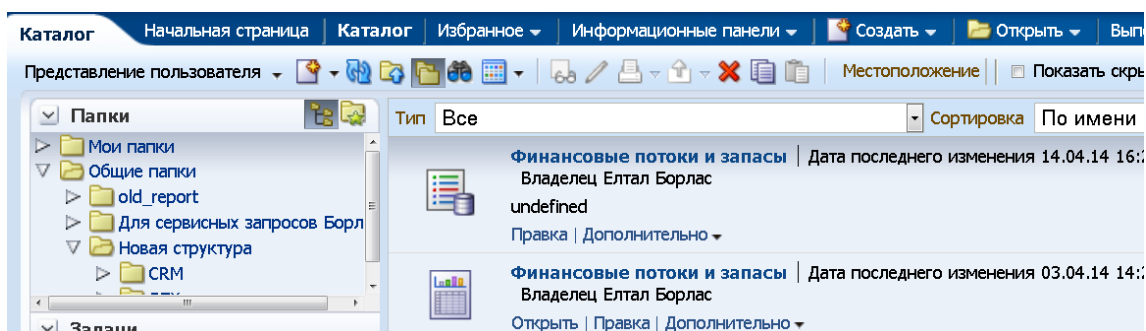


Рисунок 2.20 – Папка с отчетом и моделью данных

Для открытия отчета переходим непосредственно в сам отчет с помощью «Открыть». Откроется форма для выгрузки отчета (рисунок 2.21).

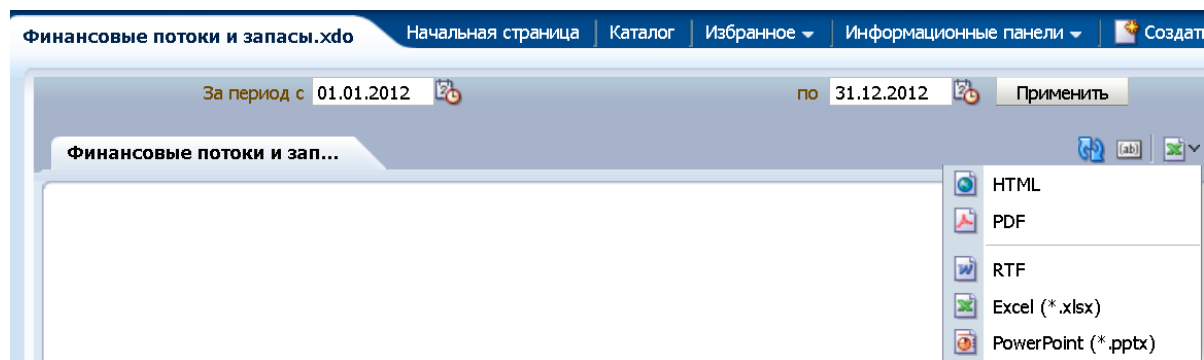


Рисунок 2.21 – Форма для выгрузки отчета

Форма содержит поля ввода входящих параметров, в нашем случае «за период с» и «по», потому что данный отчет считается годовым. Отчет можно выгрузить в форматах HTML, PDF, RTF, EXEL, Power Point. Формат HTML (рисунок 2.22) выгружается в браузер.

П/П	Наименование показателей	Всего, тыс. тенге	В том числе:									Расшифровка изданных изменений и стоимости	
			по секторам экономики:										
			Финансовый сектор	Из него:			Сектор домашних хозяйств	НКУ, обслуживающие домашние хозяйства (НКУОДХ)	Сектор органов государственной власти	Нефтегазовый сектор	В целом по экономике		Остальное (нераспределенное)
3	Банки	Другие финансовые организации (небанковские)	4	5	6	7						8	
A	B	1 (10+11)	2 (3+4+5)	3	4	5	6	7	8	9	10 (2+6+7+8+9)	11	12
	Национальная валюта												
	1. Национальная												
1	объем на начало отчетного периода	441,789	441,789	441,789							441,789		
2	изменения в результате проведения операций	332,213	332,213	332,213							332,213		
3	другие изменения (укажите вид)												
4	объем на конец отчетного периода	774,002	774,002	774,002							774,002		
	2. Иностранная												
5	объем на начало отчетного периода	9,272										9,272	
6	изменения в результате проведения операций	5,074										5,074	
7	оценочные изменения стоимости	381										381	

Рисунок 2.22 – Выгруженный отчет в формате HTML

2.6 Планирование запуска отчета

Планировщик задач (Oracle BI Scheduler), являющийся ядром модуля доставки отчетов, обладает широким функционалом благодаря тому, что поддерживает возможность вызова дополнительных модулей на языке Java.

Для планирования отчета, необходимо нажать «Дополнительно», и в выпадающем меню нажать «План» (рисунок 2.23).

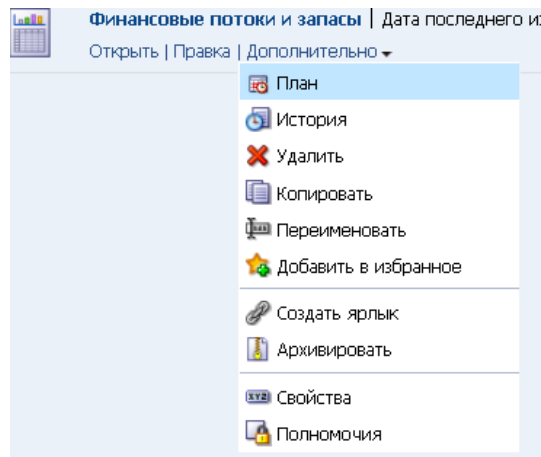


Рисунок 2.23 – Дополнительные функции отчета
Вкладка «Общие сведения» содержит название отчета и его входящие параметры (рисунок 2.24).

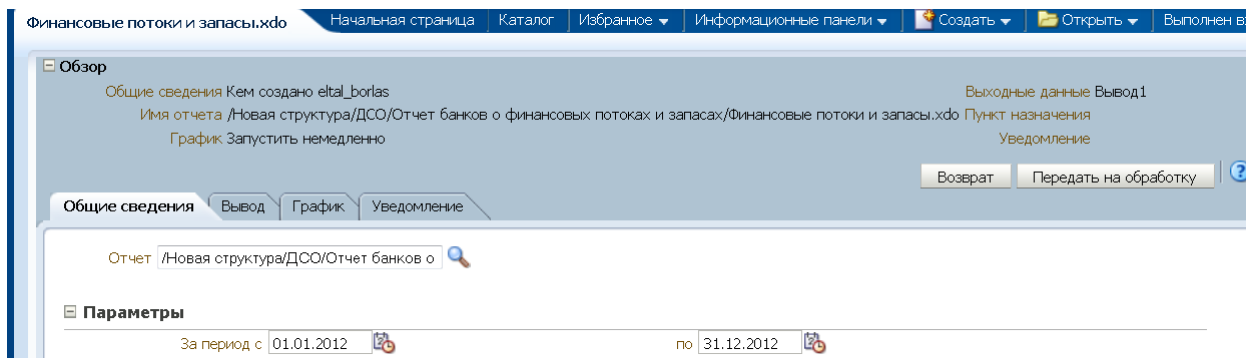


Рисунок 2.24 – Общие сведения

Вкладка «Вывод» содержит шаблоны для отчета, форматы для выгрузки, региональные настройки, часовой пояс. Так же можно задать тип пункта назначения (электронная почта, принтер, факс, File Transfer Protocol (FTP) или Web – папка (рисунок 2.25)).

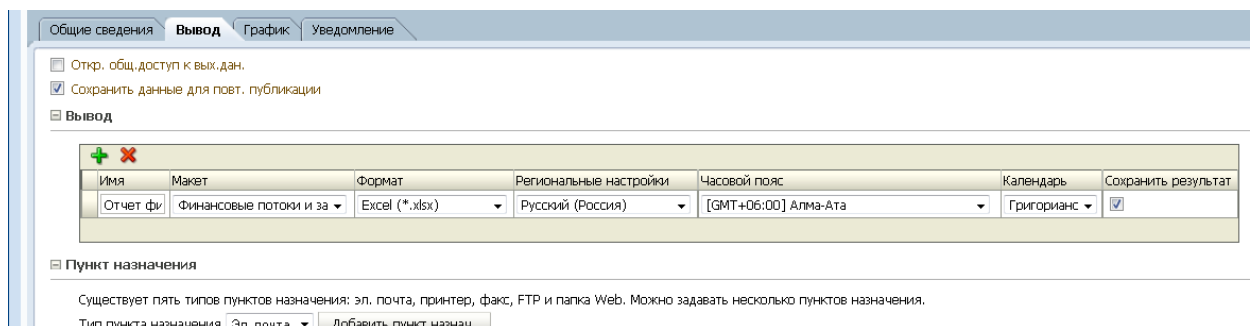


Рисунок 2.25 – Параметры вывода

Вкладка «График» содержит график выполнения отчета, можно задать периодичность выполнения, или задать число и время запуска (рисунок 2.26).

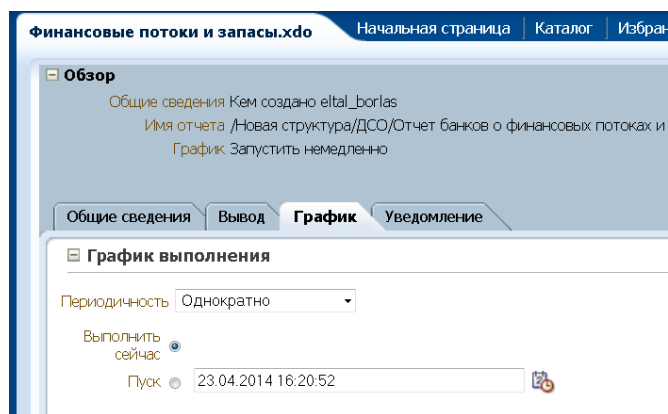


Рисунок 2.26 – График выполнения отчета

Для передачи отчета на обработку нужно нажать «Передать на обработку» и указать имя задания формирования отчета (рисунок 2.27)

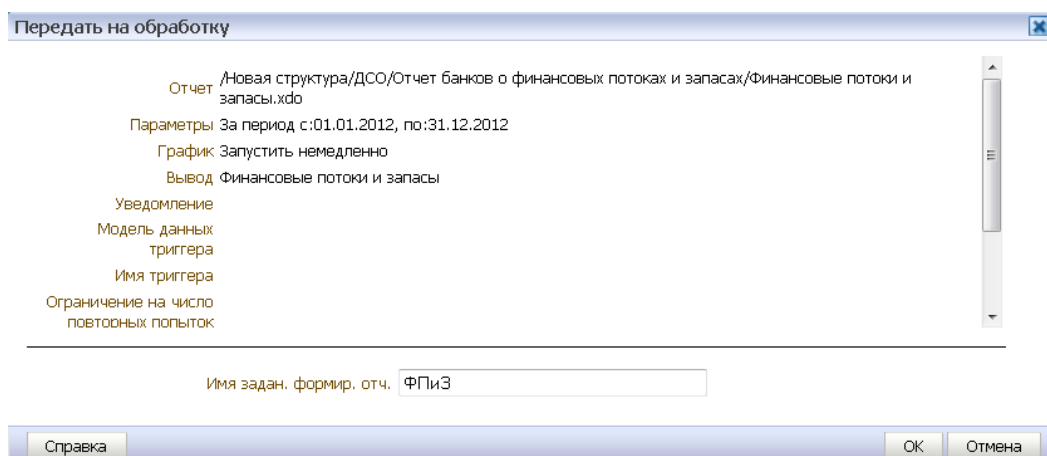


Рисунок 2.27 – Передача на обработку

Для просмотра и управления заданиями на «Начальной странице» в разделе «Обзор/Управление» нужно перейти по ссылке «Задания выполнения отчетов» (рисунок 2.28). В вкладке «Управление историей заданий» можно задать различные фильтры по имени отчета, по пути к отчету, по расписанию, времени начала, времени окончания и по состоянию (к примеру показывать только те отчеты, которые выполнены успешно).

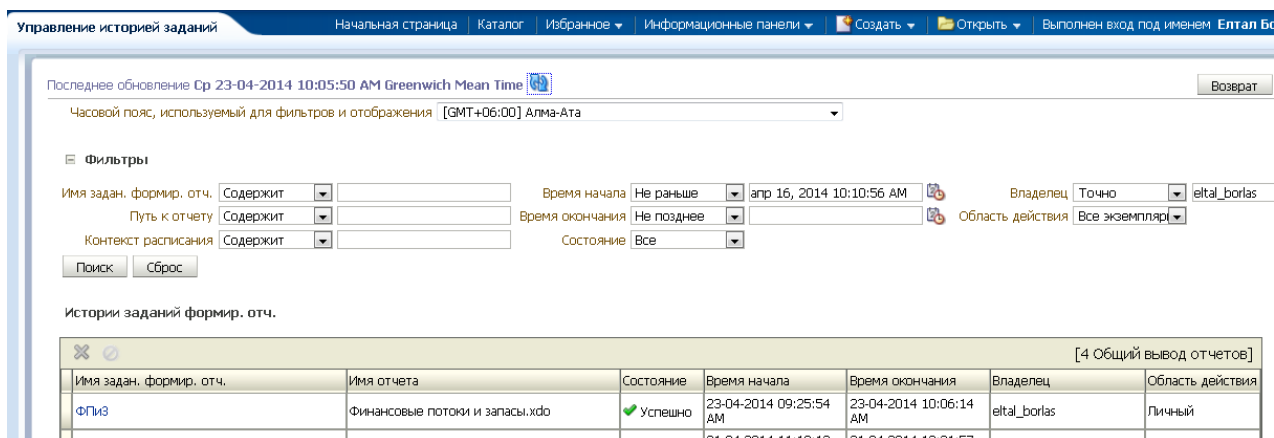


Рисунок 2.28 – Управление историей заданий

2.7 Витрины данных

Облегченным вариантом ХД является витрина данных (ВД) – это набор тематически связанных БД, содержащие информацию, относящуюся к отдельным аспектам деятельности организации. ВД существенно меньше по объему, чем ХД, и для его реализации не требуется особо мощная вычислительная техника.

ВД – это аналитическая структура, которая поддерживает область работы бизнес-процесса или отдела. Сотрудники отдела обобщают требования к информации и приспособливают каждую витрину к своим нуждам. Затем они обеспечивают персонал, работающий с информацией, средствами интерактивной отчетности.

Для того что бы выгрузить витрину, необходимо в меню «Создать» (рисунок 2.29) выбрать «Анализ» (рисунок 2.30), далее выбрать предметную область – «Витрина Главная Книга - счета». Фактовыми данными считаются обороты и остатки на определенную дату.

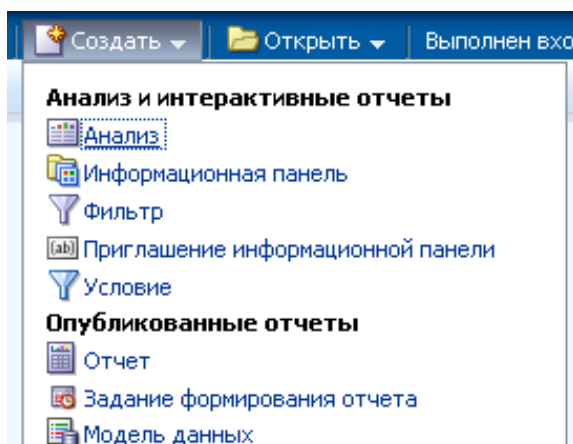


Рисунок 2.29 – Пункт меню «Создать»

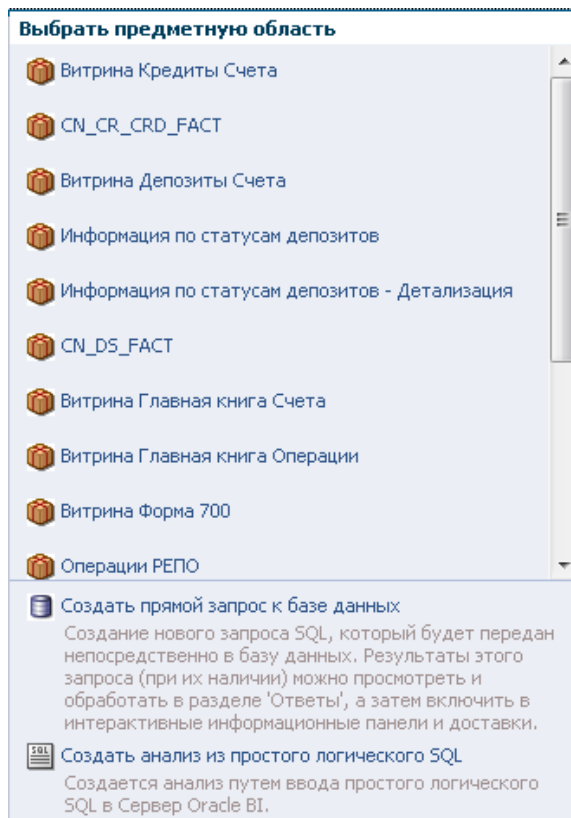


Рисунок 2.30 – Выбор предметной области

Фактовыми данными являются:

- состояние позиции балансового счёта;
- входящий остаток;
- входящее значение в тенге по валюте;
- входящее значение в тенге по валюте;
- входящий активный остаток в тенге;
- входящий пассивный остаток в тенге;
- исходящий остаток в тенге;
- исходящий активный остаток фактический в тенге;
- исходящий пассивный остаток фактический в тенге;
- обороты фактические по кредиту в тенге;
- обороты фактические по дебету в тенге;
- исходящий остаток;
- исходящий активный остаток фактический;
- исходящий пассивный остаток фактический;
- оборот по кредиту фактический;
- оборот по дебету фактический;
- признак 30;
- входящий остаток в тенге;
- операционные дни.

Для выгрузки витрины «Главная книга Счета» в разрезе плана счетов и программы кредитования, создаем критерии выгрузки (рисунок 2.31):

- операционные дни (дата операционного дня с фильтром «Дата операционного дня равно/находится в 28.02.2014»);
- план счетов (наименование);
- код плана счетов (счета ГК, наименование);
- тип счета (наименование типа счета);
- лицевые счета (код ЕБКБ);
- состояние позиции балансового счета (входящий остаток, исходящий остаток, признак з0).

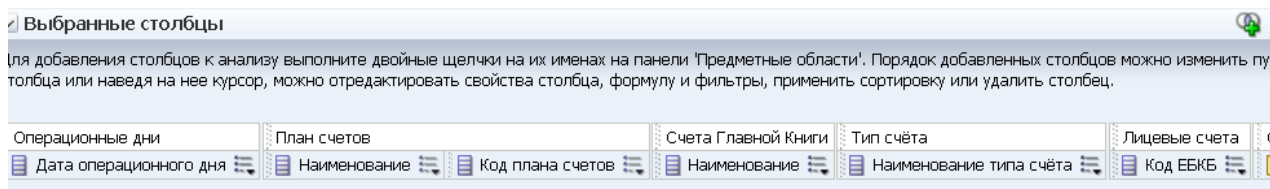


Рисунок 2.31 – Выбранные столбцы

Для добавления фильтров к критериям анализа можно нажать параметр «Фильтр» для конкретного столбца на панели «Выбранные столбцы» или нажать кнопку фильтра в заголовке панели «Фильтр» (рисунок 2.31).

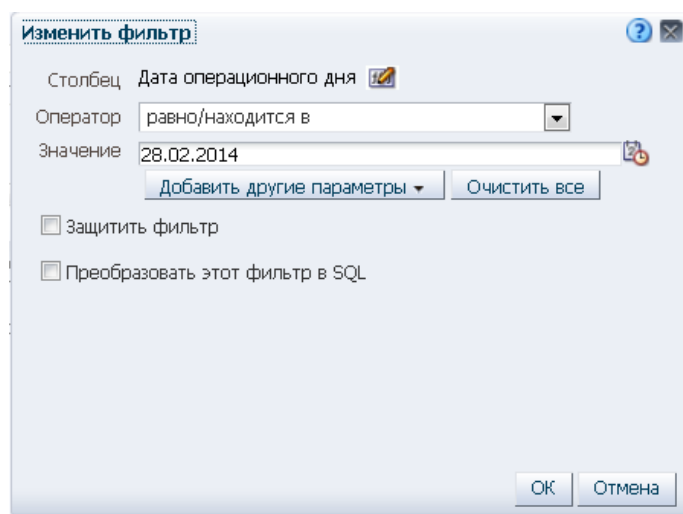


Рисунок 2.31 – Установка фильтров

Для просмотра результата переходим на вкладку «Результаты» (рисунок 2.32).

Дата операционного дня	Наименование	Код плана счетов	Наименование	Наименование типа счета	Код ЕБКБ	Входящий остаток	Исходящий остаток	Признак ЭО
28.02.14	Расходы, связанные с выплатой вознаграждения по краткосрочным займам, полученным от других банков	5054	Расходы, связанные с выплатой вознаграждения по краткосрочным займам, полученным от других банков	Внутрибанковский лицевой счет	50199212125	0,00	0,00	0
		5742	Административные расходы	Внутрибанковский лицевой счет	50131989891	8016000,00	8016000,00	0
					50199155561	13428,77	13428,77	0
					50199161622	0,00	0,00	0
					50199233331	0,00	0,00	0
					50199233332	0,00	0,00	0
					50199236361	0,00	0,00	0
					50199241411	94353,84	94353,84	0
					50199241412	276124,62	276124,62	0
					50199241413	0,00	0,00	0
				50199242420	0,00	0,00	0	
				50199242422	0,00	0,00	0	

Рисунок 2.32 – Результаты выгрузки

Для экспорта данного анализа выбираем один из формата (рисунок 2.33).

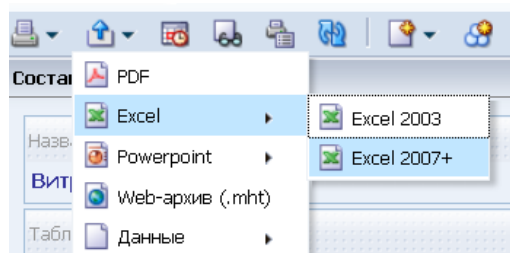


Рисунок 2.33 – Экспорт данных

3 ТЕХНИКО – ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

3.1 Описание работы и обоснование необходимости

Тема данного дипломного проекта – «Разработка модуля отчетности в среде Oracle Business Intelligence Publisher для оптимизации нагрузки на АБИС».

Данный дипломный проект предназначен для того, что бы автоматизировать выгрузку отчетных данных в системе Oracle Business Intelligence Publisher, которая является масштабируемым сервером формирования отчетов, позволяющий генерировать отчеты в разных форматах на основе данных из хранилища, с разработкой индивидуальных шаблонов и внедрением в промышленную эксплуатацию.

Дипломный проект разработан с целью обеспечить лучшее видение и понимание бизнеса широкому кругу пользователей и позволяет любому

пользователю организации получить быстрый Web-доступ к актуальной информации.

В данном разделе приводится рассмотрение экономической составляющей реализации данного дипломного проекта, отражающей временные, трудовые и финансовые затраты на проект.

Важнейшими требованиями являются соблюдение рыночного подхода к оценке любых технических мероприятий, учет конкретных условий и особенностей решения задач для дипломного проекта. Конечным критерием оценки является экономическая и социальная эффективность выполненной разработки в целом.

3.2 Трудовые ресурсы, используемые в работе

В данном дипломном проекте используется интеллектуальный труд, который осуществляется главным образом с помощью умственных усилий.

В проекте задействованы (таблица 3.1):

- руководитель;
- программист-разработчик;
- консультант по экономической части;
- консультант по части БЖД.

Т а б л и ц а 3.1 – Данные о заработной плате

Исполнитель	Количество человек	Зароботная плата, тенге
Руководитель	1	300000
Программист-разработчик	1	220000
Консультант по части «Экономика»	1	60000

Окончание таблицы 3.1

Исполнитель	Количество человек	Зароботная плата, тенге
Консультант по части «БЖД»	1	60000
Итого	4	640000

3.3 Оборудование и программное обеспечение, используемое в работе

Необходимо зафиксировать оборудование, которое задействовано в разработке данного дипломного проекта, так как оно составляет не малую долю затрат. Далее найти амортизацию по мере физического или морального износа от итоговой стоимости используемого модуля отчетности при разработке.

Для разработки, тестирования и внедрения данного программного модуля, используется следующее оборудование:

- ноутбук ASUS N550JV /Intel Core i7 4700HQ(2.4Ghz)/8192Mb/1000Gb;
- мышка ASUS WX470 Laser Black USB.

Данные о количестве каждого вида оборудования и стоимости представлены в таблице 3.2.

Т а б л и ц а 3.2 – Используемое оборудование и программное обеспечение

Наименование материала	Единица измерения	Количество	Цена за единицу в тенге	Сумма в тенге (с НДС)	Сумма в тенге (без НДС)
ноутбук ASUS N550JV/Intel Core i7 4700HQ(2.4Ghz) /8192Mb/1000Gb	штук	1	239 600	239 600	213928
мышка ASUS WX470 Laser Black USB	штук	1	9 130	9 130	8151
Oracle Business Intelligence	штук	1	218400	218400	195000
Windows 7 Professional	штук	1	35600	35600	31785
PL/SQL Developer 10.0	штук	1	11650	11650	10401
Итого:				514380	459265

Сумма налога на добавленную стоимость (НДС) – 12%, подлежащая уплате в соответствии с настоящей статьей, определяется путем применения ставки, предусмотренной пунктом 1 статьи 268 настоящего Налогового кодекса Республики Казахстан, к размеру облагаемого оборота.

Общая стоимость расходов на оборудование и программное обеспечение (ПО) составляет 459265 тенге.

3.4 Расчет стоимости разработки программного обеспечения

Расчет осуществляется по калькуляционным статьям расходов. Калькуляция даёт возможность определить плановую или фактическую себестоимость. Прежде всего необходимо рассчитать себестоимость на протяжении всего жизненного цикла. Себестоимость – это сумма расходов, связанных с производством продукции (производственная себестоимость), и расходов по ее реализации (коммерческие расходы).

Себестоимость ПО складывается из следующих статей затрат:

- оплата труда;
- социальный налог;
- амортизационные отчисления;
- расходы на электроэнергию;
- накладные расходы.

Себестоимость разработки проекта определяется по формуле (3.1).

$$C = \text{ФОТ} + C_{\text{н}} + A + C_{\text{э}} + H \quad (3.1)$$

где ФОТ – фонд оплаты труда;
 $C_{\text{н}}$ – социальный налог;
 A – амортизационные начисления;
 $C_{\text{э}}$ – расходы на электроэнергию;
 H – накладные расходы ^[9].

3.5 Сроки реализации проекта

Разработка модуля отчетности в среде Oracle BI Publisher для оптимизации нагрузки на АБИС включает в себя следующие этапы:

- 1 этап: анализ компонент и основных возможностей BI Publisher как средства формирования отчетов;
- 2 этап: создание компонент и основных возможностей BI Publisher как средства формирования отчетов;
- 3 этап: создание моделей данных с помощью редактора моделей данных (Data Model Editor) на базе Oracle Database 11g;
- 4 этап: создание параметров и списков допустимых значений для моделей данных;
- 5 этап: создание отчетов BI Publisher на основе моделей данных;
- 6 этап: создание макетов внешнего вида отчета с использованием редактора макетов (Layout Editor);
- 7 этап: тестирование модуля отчетности;
- 8 этап: оформление отчета.

Т а б л и ц а 3.3 – Этапы и сроки реализации модуля

Перечень работ		Недели от начала работ									
		1	2	3	4	5	6	7	8	9	10
1 этап	Постановка задачи	■									
	Анализ компонент и основных возможностей BI Publisher										
	Изучение литературы		■								
2 этап	Создание компонент и основных возможностей BI Publisher		■								
3 этап	Изучение редактора моделей данных (Data Model Editor) на базе Oracle Database 11g			■							
	Создание моделей данных с помощью редактора моделей данных				■						
этап	Создание параметров для моделей данных				■						

^[9] Николаева С.А. Принципы формирования и калькулирования себестоимости. – М.: Изд.: Аналитика, 2009.

	Создание списков допустимых значений для моделей данных					■					
5 этап	Создание отчетов BI Publisher на основе моделей данных						■				
6 этап	Создание макетов внешнего вида отчета с использованием редактора макетов (Layout Editor)							■			
7 этап	Тестирование модуля отчетности								■		
	Отладка недочетов									■	
	Внедрение										■
8 этап	Подготовка раздела «Экономическая часть»										■
	Подготовка раздела «БЖД»										■
	Проверка и сдача отчетов										■

3.5.1 Расчет затрат на оплату труда

Для расчета затрат на заработную плату необходимы следующие данные:

- численность за действенного персонала;
- среднемесячная заработная плата каждого работника;
- длительность разработки проекта и каждого вида выполненных работ;
- трудоемкость.

В процессе разработки данного программного обеспечения участвует четыре человека.

Месячная заработная плата сотрудников:

- программист-разработчик – 220000 тенге;
- руководитель – 300000 тенге;
- консультант по части «Экономика» – 60000 тенге;
- консультант по части «БЖД» – 60000 тенге.

Зарботную плату за один час рассчитаем по формуле (3.2)

$$D = \frac{ЗП_{\text{м}}}{D_{\text{р}} * Ч_{\text{р}}} \quad (3.2)$$

где $ЗП_{\text{м}}$ – ежемесячный размер заработной платы;

$D_{\text{р}}$ – количество рабочих дней в месяце (21 рабочий день);

$Ч_{\text{р}}$ – количество часов раб. дня (при 8-ми часовом рабочем дне);

- 1) Зарботная плата программиста-разработчика за один час составляет:

$$D = \frac{220000}{21 * 8} = \frac{220000}{168} = 1309,5 \text{ тенге/час.}$$

2) Заработная плата руководителя за один час составляет:

$$D = \frac{300000}{21 * 8} = \frac{300000}{168} = 1785,7 \text{ тенге/час.}$$

3) Заработная плата консультанта по части «Экономика» за один час составляет:

$$D = \frac{60000}{21 * 8} = \frac{60000}{168} = 357,14 \text{ тенге/час.}$$

4) Заработная плата консультанта по части «БЖД» за один час составляет:

$$D = \frac{60000}{21 * 8} = \frac{60000}{168} = 357,14 \text{ тенге/час.}$$

Длительность цикла в днях по каждому виду работ укрупнено, определяем по формуле (3.3).

$$t_n = \frac{T}{q_n * z * K} \quad (3.3)$$

где T – трудоемкость этапа, норма-час;

q_n – количество исполнителей по этапу;

z – продолжительность рабочего дня, $z = 8$ часов;

K – коэффициент выполнения норм времени, $K = 1,1$.

Полученная величина t_n округляется в большую сторону до целых дней.

Руководитель – постановка задачи:

$$t_1 = \frac{20}{1 * 8 * 1,1} = \frac{20}{8,8} \approx 3 \text{ дн.}$$

Программист – разработчик – анализ компонент и основных возможностей VI Publisher:

$$t_2 = \frac{10}{1 * 8 * 1,1} = \frac{10}{8,8} \approx 2 \text{ дн.}$$

Программист – разработчик – изучение литературы:

$$t_3 = \frac{10}{1 * 8 * 1,1} = \frac{10}{8,8} \approx 2 \text{ дн.}$$

Программист – разработчик – создание компонент и основных возможностей BI Publisher:

$$t_4 = \frac{20}{1 * 8 * 1,1} = \frac{20}{8,8} \approx 3 \text{ дн.}$$

Программист – разработчик – изучение редактора моделей данных:

$$t_5 = \frac{10}{1 * 8 * 1,1} = \frac{10}{8,8} \approx 2 \text{ дн.}$$

Программист – разработчик – создание моделей данных с помощью редактора моделей данных:

$$t_6 = \frac{20}{1 * 8 * 1,1} = \frac{20}{8,8} \approx 3 \text{ дн.}$$

Программист – разработчик – создание параметров для моделей данных:

$$t_7 = \frac{30}{1 * 8 * 1,1} = \frac{30}{8,8} \approx 4 \text{ дн.}$$

Программист – разработчик – создание списков допустимых значений для моделей данных:

$$t_8 = \frac{8}{1 * 8 * 1,1} = \frac{8}{8,8} \approx 1 \text{ дн.}$$

Программист – разработчик – создание отчетов BI Publisher на основе моделей данных:

$$t_9 = \frac{40}{1 * 8 * 1,1} = \frac{40}{8,8} \approx 5 \text{ дн.}$$

Программист – разработчик – создание макетов внешнего вида отчета с использованием редактора макетов:

$$t_{10} = \frac{40}{1 * 8 * 1,1} = \frac{40}{8,8} \approx 5 \text{ дн.}$$

Руководитель и программист-разработчик – тестирование модуля отчетности:

$$t_{11} = \frac{8}{1 * 8 * 1,1} = \frac{8}{8,8} \approx 1 \text{ дн.}$$

Программист – разработчик – отладка недочетов:

$$t_{12} = \frac{30}{1 * 8 * 1,1} = \frac{30}{8,8} \approx 4 \text{ дн.}$$

Программист – разработчик – внедрение:

$$t_{13} = \frac{40}{1 * 8 * 1,1} = \frac{40}{8,8} \approx 5 \text{ дн.}$$

Консультант по части «Экономика» – подготовка раздела «Экономика»:

$$t_{14} = \frac{8}{1 * 8 * 1,1} = \frac{8}{8,8} \approx 1 \text{ дн.}$$

Программист – разработчик – подготовка раздела «Экономика»:

$$t_{15} = \frac{10}{1 * 8 * 1,1} = \frac{10}{8,8} \approx 2 \text{ дн.}$$

Консультант по части «БЖД» – подготовка раздела «БЖД»:

$$t_{16} = \frac{8}{1 * 8 * 1,1} = \frac{8}{8,8} \approx 1 \text{ дн.}$$

Программист – разработчик – подготовка раздела «БЖД»:

$$t_{17} = \frac{8}{1 * 8 * 1,1} = \frac{8}{8,8} \approx 1 \text{ дн.}$$

Программист – разработчик – проверка и сдача отчета:

$$t_{18} = \frac{40}{1 * 8 * 1,1} = \frac{40}{8,8} \approx 5 \text{ дн.}$$

Сводные результаты расчета затрат на основную заработную плату работников, задействованных в проекте – «Разработка модуля отчетности в среде Oracle Business Intelligence Publisher для оптимизации нагрузки на АБИС», представлены в таблице 3.4.

Т а б л и ц а 3.4 – Результаты расчета затрат на основную заработную плату

Наименование содержания работы	Исполнитель	Трудоемкость		Длительность цикла, дни	Зарплата за час работы, тенге	Сумма заработной платы, тенге
		Нормы часы	% от общей трудоемкости			
Постановка задачи	Руководитель	20	5,5	3	1785,7	35750
Анализ компонент и основных возможностей BI Publisher	Программист – разработчик	10	2,7	2	1309,5	13095
Изучение литературы	Программист – разработчик	10	2,7	2	1309,5	13095
Создание компонент и основных возможностей BI Publisher	Программист – разработчик	20	5,5	3	1309,5	26190
Изучение редактора моделей данных(Data Model Editor) на базе Oracle Database 11g	Программист – разработчик	10	2,7	2	1309,5	13095
Создание моделей данных с помощью редактора моделей данных	Программист – разработчик	20	5,5	3	1309,5	26190
Создание параметров для моделей данных	Программист – разработчик	30	8,3	4	1309,5	39285
Создание списков допустимых значений для моделей данных	Программист – разработчик	8	2,2	1	1309,5	10476
Создание отчетов BI Publisher на основе моделей данных	Программист – разработчик	40	11,1	5	1309,5	52380

Окончание таблицы 3.4

Наименование содержания работы	Исполнитель	Трудоемкость	ост	ь	та	за	час	або	тно	й	пла
--------------------------------	-------------	--------------	-----	---	----	----	-----	-----	-----	---	-----

		Нормы часы	% от общей трудоемкости			
Создание макетов внешнего вида отчета с использованием редактора макетов (Layout Editor)	Программист – разработчик	40	11,1	5	1309,5	52380
Тестирование модуля отчетности	Программист – разработчик	8	2,2	1	1309,5	10476
Отладка недочетов	Программист – разработчик	30	8,3	4	1309,5	39285
Внедрение	Программист – разработчик	40	11,1	5	1309,5	52380
Подготовка раздела «Экономическая часть»	Консультант по части «Экономика»	8	2,2	1	357,14	2857,12
Подготовка раздела «Экономическая часть»	Программист – разработчик	10	2,7	2	1309,5	13095
Подготовка раздела «БЖД»	Консультант по части «БЖД»	8	2,2	1	357,14	2857,12
Подготовка раздела «БЖД»	Программист – разработчик	8	2,2	1	1309,5	10476
Проверка и сдача отчетов	Программист – разработчик	40	11,1	5	1309,5	52380
Итого	–	360	100	50	–	465742,24

Дополнительная заработная плата составляет 10% от основной заработной платы и составляет:

$$Z_{\text{доп}} = 465742,24 * 0,1 = 46574,224 \text{ тенге.}$$

Таким образом, суммарный фонд оплаты труда исполнителей составит:

$$\text{ФОТ} = 465742,24 + 46574,224 = 512316,5 \text{ тенге.}$$

3.5.2 Расчет затрат по социальному налогу

Согласно разделу 12 «Социальный налог» статья 358 «Ставки налога» Налогового кодекса Республики Казахстан по состоянию на 01.01.2014 г. –

для юридических лиц-резидентов Республики Казахстан, а также нерезидентов, осуществляющих деятельность в Республике Казахстан через постоянное учреждение, филиалов и представительств иностранных юридических лиц, уплачивают социальный налог по ставке 11 % от сумм облагаемых доходов, выплаченных работодателем.

Социальный налог рассчитывается по формуле (3.4).

$$C_n = (\text{ФОТ}-\text{ПО}) * 11\% \quad (3.4)$$

где ФОТ – фонд оплаты труда;

ПО – пенсионные отчисления.

Пенсионные отчисления вычисляются по формуле (3.5).

$$\text{ПО} = \text{ФОТ} * 10\% \quad (3.5)$$

$$\text{ПО} = 512316,5 * 10\% = 51231,65 \text{ тенге.}$$

Таким образом, отчисления на социальный налог составляют:

$$C_n = (512316,5 - 51231,65) * 11\% = 461084,8 * 11\% = 50719,33 \text{ тенге.}$$

3.5.3 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле (3.6).

$$A_i = N_A * C_{\text{ПЕР}} * N / 100 * 12 * n \quad (3.6)$$

где N_A – норма амортизации;

$C_{\text{ПЕР}}$ – первоначальная стоимость оборудования;

N – количество дней на выполнение работ;

n – количество дней в рабочем месяце.

Норма амортизации N_A на компьютерную технику составляет 40% от стоимости всего оборудования.

Таким образом, амортизационные отчисления по используемому оборудованию, в соответствии с формулой (3.6) составят:

– на ноутбук:

$$A_1 = \frac{40 * 210848 * 50}{100 * 12 * 21} = \frac{421696000}{25200} = 16733,9 \text{ тенге;}$$

– на мышь:

$$A_2 = \frac{40 * 8034 * 50}{100 * 12 * 21} = \frac{16068000}{25200} = 637,6 \text{ тенге.}$$

Сводные результаты расчета амортизационных отчислений предоставлены в таблице 3.5.

Т а б л и ц а 3.5 – Сведения по затратам на амортизацию

Наименование оборудования	Количество	Норма амортизации, %	Сумма амортизации, тенге	Цена за единицу, тенге
ноутбук ASUS N550JV/Intel Core i7 4700HQ(2.4Ghz) /8192Mb/1000Gb	1	40	16733,9	210848
мышка ASUS WX470 Laser Black USB	1	40	637,6	8034
Итого	–	–	17371,5	–

3.5.4 Расчет затрат на электроэнергию

В процессе разработки данного дипломного проекта, используется электрооборудование, поэтому необходимо рассчитать затраты на электроэнергию. Затраты на электроэнергию для производственных нужд включают в себя расходы электроэнергии на оборудование и дополнительные нужды и рассчитывается по формуле (3.7).

$$\mathcal{E} = \mathcal{Z}_{\text{эл.+эн.+обор.}} + \mathcal{Z}_{\text{доп.нуж.}} \quad (3.7)$$

где $\mathcal{Z}_{\text{эл.+эн.+обор.}}$ – затраты на электроэнергию оборудования;

$\mathcal{Z}_{\text{доп.нуж.}}$ – затраты электроэнергии на дополнительные нужды.

Расходы по электроэнергии на оборудование рассчитываются по формуле (3.8).

$$\mathcal{Z}_{\text{эл.+эн.+обор.}} = W * T * S * K_{\text{исп}} \quad (3.8)$$

где W – потребляемая мощность, Вт;

T – время работы;

S – тариф (1 кВт = 21,71 тг/кВт * ч – без НДС);

$K_{\text{исп}}$ – коэффициент использования ($K_{\text{исп}} = 0,9$).

$$\mathcal{Z}_{\text{эл.+эн.+обор.1}} = 0,8 * 400 * 21,71 * 0,9 = 6252,48 \text{ тенге};$$

$$\mathcal{Z}_{\text{эл.+эн.+обор.2}} = 0,6 * 40 * 21,71 * 0,9 = 468,94 \text{ тенге.}$$

Суммарные затраты на электроэнергию основного оборудования составляют:

$$\mathcal{Z}_{\text{эл.+эн.+обор.}} = 6252,48 + 468,94 = 6721,42 \text{ тенге.}$$

Затраты на дополнительные нужды берутся по укрупненному показателю в размере 5% от затрат на оборудование и составляют:

$$Z_{\text{доп.нуж.}} = 6721,42 * 0,05 = 336,07 \text{ тенге.}$$

Таким образом, суммарные затраты на электроэнергию составляют:

$$C_{\Sigma} = 6721,42 + 336,07 = 7057,49 \text{ тенге.}$$

Сводные результаты расчета затрат на электроэнергию представлены в таблице 3.6.

Т а б л и ц а 3.6 – Затраты на электроэнергию

Наименование приборов	W, Вт	Число рабочих дней	K _{исп}	Время работы прибора, час	ΣW кВт/ч	Сумма, тенге
Ноутбук	0,8	50	0,9	400	320	6252,48
Принтер	0,6	5	0,9	40	24	336,07
Итого	–	–	–	–	344	7057,49

3.5.5 Расчет затрат на накладные расходы

Накладные расходы сопутствующие на разработку проекта составляют 25% от общей суммы затрат и рассчитываются по формуле (3.9).

$$H = \text{ФОТ} * 0,25 \quad (3.9)$$

В итоге, согласно формуле (3.9), накладные расходы будут равны:

$$H = 512316,5 * 0,25 = 128079,125 \text{ тенге.}$$

3.5.6 Расчет стоимости по всем статьям затрат

В соответствии с произведенными расчетами по статьям затрат себестоимость проекта, согласно формуле 3.1, будет равна:

$$C = 512316,5 + 50719,33 + 17371,5 + 7529,46 + 128079,125 = 716015,9 \text{ тенге.}$$

Сводные результаты расчета себестоимости разработки программного продукта представлены в таблице 3.7 и на рисунке 3.1.

Т а б л и ц а 3.7 – Стоимость разработки модуля отчетности

Наименование статей затрат	Сумма, тенге	Структура затрат %
----------------------------	--------------	--------------------

ФОТ	512316,5	43,86%
Отчисления на социальные нужды	50719,33	4,34%
Амортизация	17371,5	1,49%
Затраты на электроэнергию	7057,49	0,60%
Программное обеспечение	459265	39,31%
Накладные расходы	128079,125	10,96%
Итого	1174808,945	100

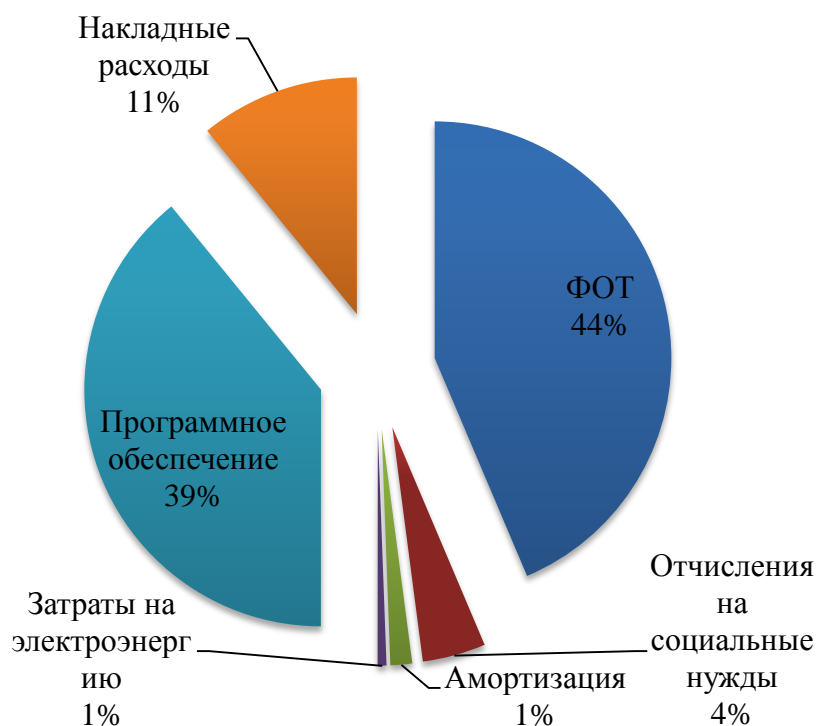


Рисунок 3.1 – Структура себестоимости модуля отчетности.

3.6 Цена интеллектуального труда

Цена реализации программного продукта складывается из себестоимости и чистого дохода, и вычисляется по формуле (3.10).

$$Ц = С * П \quad (3.10)$$

где С – себестоимость продукта;

П – чистый доход.

Первоначальная цена рассчитывается через рентабельность проекта. Учитывая, что желаемый уровень рентабельности для отрасли телекоммуникации составляет 25%, применим следующую формулу (3.11).

$$Ц_{п} = С * \left(1 + \frac{Р}{100}\right) \quad (3.11)$$

где Р – рентабельность (25%).

Согласно формуле (3.10) первоначальная цена будет равна:

$$C_{\text{п}} = 1174808,945 * \left(1 + \frac{25}{100}\right) = 1468511,181 \text{ тенге.}$$

Цена реализации готовой продукции рассчитывается по формуле (3.12).

$$C_{\text{р}} = C_{\text{п}} + \text{НДС} \quad (3.12)$$

На сегодняшний день налог на добавленную стоимость в Республики Казахстан составляет 12%, следовательно, рассчитывается по формуле (3.13).

$$\text{НДС} = \frac{12}{100} * C_{\text{п}} \quad (3.13)$$

$$\text{НДС} = \frac{12}{100} * 1468511,181 = 176221,34 \text{ тенге} \quad (3.13)$$

Согласно формуле (3.12) цена реализации готовой продукции составит:

$$C_{\text{р}} = 1468511,181 + 176221,34 = 1644732,52 \text{ тенге.}$$

3.7 Вывод

Себестоимость программного продукта «Разработка модуля отчетности в среде Oracle Business Intelligence Publisher для оптимизации нагрузки на АБИС» составила 1174808,945 тенге, наибольшую долю в себестоимости разработки составляет фонд оплаты труда – 512317 тенге (43,86%) и затраты на программное обеспечение 459265 (38,73%), а цена реализации продукта - 1644732,52 тенге. На основании проведенных расчетов можно сделать вывод, что разрабатываемый программный продукт является экономически эффективным и выгоден для использования.

4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ, ОХРАНА ТРУДА И ПРОМЫШЛЕННАЯ ЭКОЛОГИЯ

4.1 Описание работы и обоснование необходимости

Тема данного дипломного проекта – «Разработка модуля отчетности в среде Oracle Business Intelligence Publisher для оптимизации нагрузки на АБИС».

Данный дипломный проект предназначен для того, что бы автоматизировать выгрузку отчетных данных в системе Oracle Business Intelligence Publisher, которая является масштабируемым сервером формирования отчетов, позволяющий генерировать отчеты в разных форматах на основе данных из хранилища.

Дипломный проект разработан с целью обеспечить лучшее видение и понимание бизнеса широкому кругу пользователей и позволяет любому пользователю организации получить быстрый Web-доступ к актуальной информации.

Модулем отчетности пользуется широкий круг пользователей, который все свое рабочее время проводит за персональным компьютером, поэтому необходимо обеспечить правильные условия искусственного освещения. Потому что неправильное освещение, является одной из главных причин зрительного дискомфорта при работе на персональном компьютере.

Очень важно, чтобы помещение для установки компьютеров отвечало необходимым гигиеническим требованиям, соблюдение которых способствовало бы оптимизации рабочего процесса и сохранению здоровья.

В данном разделе приводится рассмотрение условий искусственного освещения.

4.2 Анализ условий труда в производственном помещении

Дипломный проект разрабатывался в одиннадцатизэтажном здании, на одиннадцатом этаже, в офисе открытой планировки (open plan office), которые у нас называются «опенспейсами» и «ньюзрумами».

Для Казахстана опенспейс – новинка, которая входит в современную жизнь последние 11–16 лет. Добираясь до работы в тесноте общественного транспорта и дорожных пробок, снова оказываемся в толпе, только офисной.

Оказываясь в опенспейсе, человек подвергается сильному эмоциональному воздействию: забытые на столах телефоны, как назло звонящие в это время, постоянный гул, отсутствие личного пространства, суета т.д. Отсюда, как утверждают некоторые эксперты, возникают проблемы с концентрацией, стрессы, снижение работоспособности. Ученые также обратили внимание, что чем старше человек, тем больше личного пространства ему нужно.

Но опенспейс помогает человеку побороть такой социальный страх, как деятельность на публике, многочасовое нахождение в окружении малознакомых людей.

Площадь опенспейса составляет 600 квадратных метров, 20 метров длина, 30 ширина и высота потолка 5 метров. В общем на этаже расположены столовая (13,5 кв.м.), с/у (6 кв. м.), кабинет 1 (33,5 кв. м.),

кабинет 2 (15,1 кв. м.), кабинет 3 (15,5 кв. м.) и серверная (10,5 кв.м.). Поэтому при заданном объеме помещения опенспейса необходимо обеспечить условия искусственного освещения.

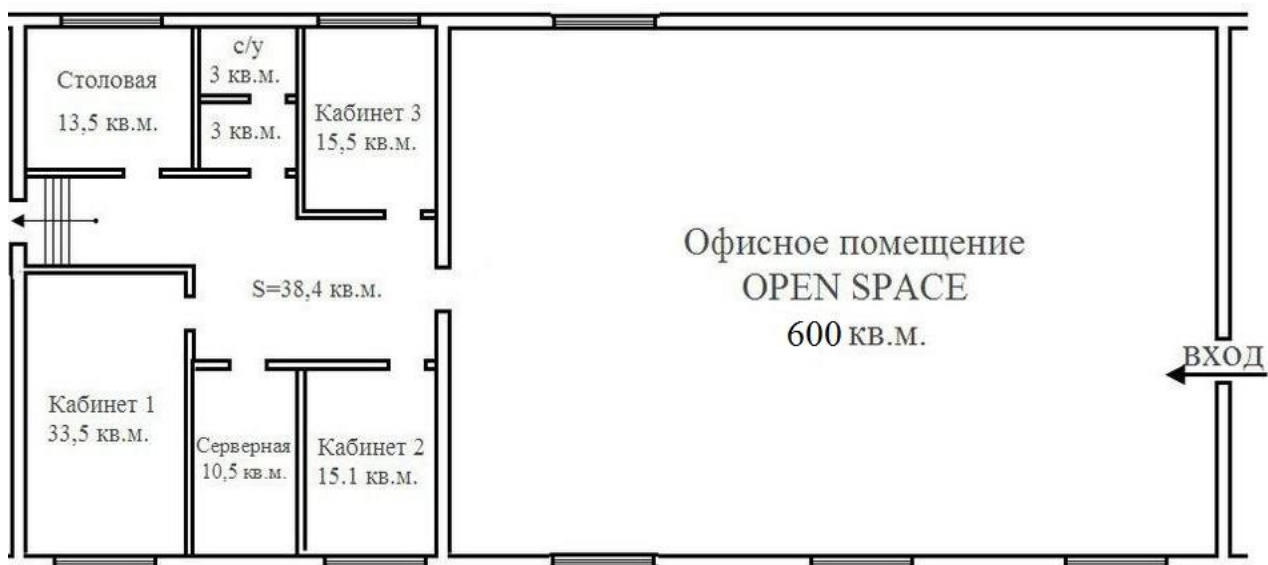


Рисунок 4.1 – План опенспейса

4.3 Трудовые ресурсы и требования

Необходимо разработать систему искусственного освещения для офиса открытой планировки с исходными данными, которые представлены в таблице 4.1.

В этапы разработки систему искусственного освещения для опенспейса входят:

- составить существующую схему размещения светильников;
- рассчитать освещенность на рабочем месте точечным методом;
- привести нормативные значения освещенности согласно СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования;
- произвести реконструкцию системы освещения производственного помещения;
- привести планируемую схему расположения светильников.

Т а б л и ц а 4.1 – Исходные данные

Наименование	Измерения
Помещение	Офис открытой планировки (open plan office)
Габариты	20х30х5
Тип светильника	ДРЛ-250 (дуговая ртутная лампа)
Количество светильников	18 шт

Разряд зрительной работы	IV,а
Коэффициенты отражения	
$\rho_{ном}$	30
$\rho_{ст}$	10
$\rho_{пол}$	10

4.4 Расчет освещенности на рабочем месте точечным методом

Расчет искусственного освещения в помещениях можно производить следующими четырьмя методами: точечным, ватт (по таблицам удельной мощности), графическим и методом коэффициента использования светового потока.

Точечный метод применяется для расчета осветительной установки при локализованном размещении светильников. Этим методом можно определить освещение наклонных плоскостей, а также проверить расчет равномерного общего освещения (без учета отраженного светового потока).

Точечный метод расчета освещения применяется для расчета общего равномерного и локализованного освещения, местного освещения независимо от расположения освещаемой поверхности при светильниках прямого света. Согласно данной методики освещенность определяется в каждой точке рассчитываемой поверхности, относительно каждого источника освещения.

Используют для расчета неравномерного освещения: общего локализованного, местного, наклонных поверхностей, наружного. Необходимый световой поток осветительной установки определяют исходя из условия, что в любой точке освещаемой поверхности освещенность должна быть не меньше нормированной, даже в конце срока службы источника света.

Определим расчётную высоту подвеса, если:

- высота свеса светильников – $h_{св} = 0,3$ м;
- рабочая поверхность над полом – $h_{р.пов.} = 0,7$ м;
- высота офиса открытой планировки (open plan office) – $h = 5$ м.

Тогда расчётную высоту определим по формуле (4.1).

$$h_{расч} = H - h_{св.} - h_{р.пов.} \quad (4.1)$$

$$h_{расч} = 5 - 0,3 - 0,7 = 4 \text{ м.}$$

Общая схема расчетной высоты ($h_{расч}$) показана на рисунке 4.2.

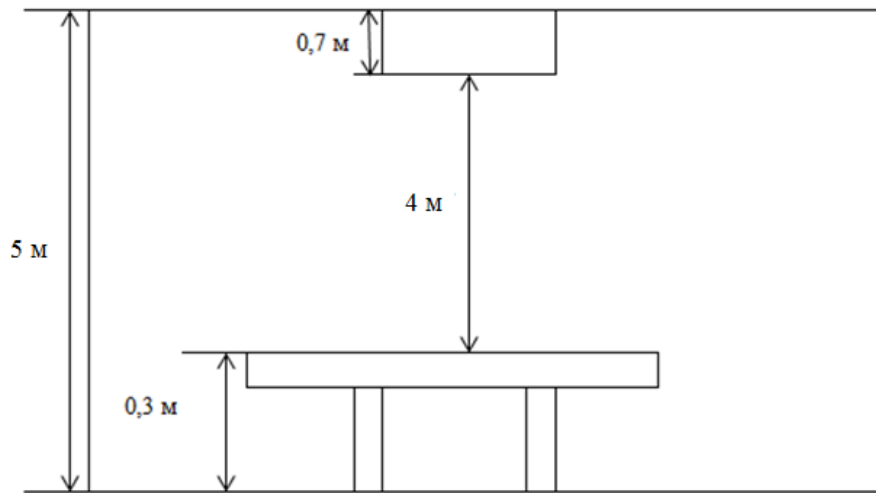


Рисунок 4.2 – Общая схема расчетной высоты ($h_{расч}$)

Определяем расстояние между светильниками, с помощью формул (4.2, 4.3) учитывая, что $\lambda=1,2\div 2,0$.

$$L_{A,B} = \lambda * h_p \quad (4.2)$$

$$L_A = 1,2 * 4 = 4,85 \text{ м};$$

$$L_B = 1,2 * 4 = 4,85 \text{ м}.$$

$$l_{a,b} = (0,3 \div 0,5) * L_{A,B} \quad (4.3)$$

$$l_a = 0,5 * 4,85 = 2,42 \text{ м};$$

$$l_b = 0,3 * 4,85 = 1,45 \text{ м}.$$

Для нахождения суммарной освещенности, необходимо нарисовать схему расположения светильников (рисунок 4.3).

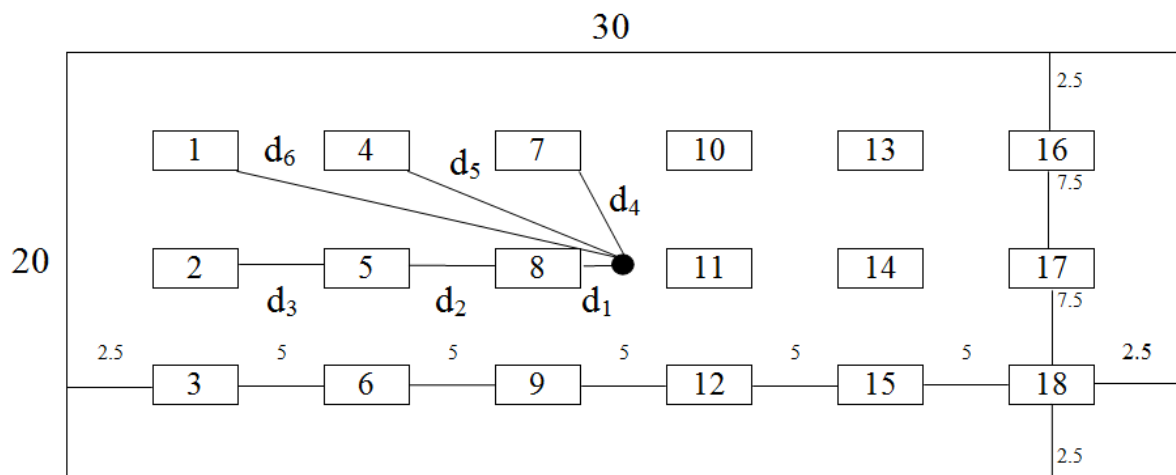


Рисунок 4.3 – Расчетная схема точечного метода

Намечаем контрольную точку А. Для нее определяем суммарную условную освещенность всех светильников следующим образом: находим проекцию расстояния на потолок от точки А до светильника – d , далее определяем угол между потолком и прямой d , и по этому углу находим условную освещенность с помощью формул (4.4 – 4.9).

$$d_1 = \frac{L_A}{2} \quad (4.4)$$

$$d_1 = \frac{4.85}{2} = 2.425 \text{ м};$$

$$d_2 = L_A * 1.5 \quad (4.5)$$

$$d_2 = 4.85 * 1.5 = 7.275 \text{ м};$$

$$d_3 = L_A * 2.5 \quad (4.6)$$

$$d_3 = 4.85 * 2.5 = 12.125 \text{ м};$$

$$d_4 = \sqrt{\left(\frac{L_A}{2}\right)^2 + (L_B)^2} \quad (4.7)$$

$$d_4 = \sqrt{\left(\frac{4.85}{2}\right)^2 + (4.85)^2} = 5.42 \text{ м};$$

$$d_5 = \sqrt{(L_A * 1.5)^2 + (L_B)^2} \quad (4.8)$$

$$d_5 = \sqrt{(4.85 * 1.5)^2 + (4.85)^2} = 8.74 \text{ м};$$

$$d_6 = \sqrt{(L_A * 2.5)^2 + (L_B)^2} \quad (4.9)$$

$$d_6 = \sqrt{(4.85 * 2.5)^2 + (4.85)^2} = 13.06 \text{ м}.$$

Далее определяем угол по формуле (4.10) между высотой потолка и соответствующим отрезком.

$$\text{tg}\alpha_i = \frac{d_i}{h_{\text{расч}}} \quad (4.10)$$

$$\begin{aligned} \operatorname{tg}_{\alpha_1} &= \frac{2.425}{4} = 0.6 \rightarrow \alpha_1 = 31^\circ \\ \operatorname{tg}_{\alpha_2} &= \frac{7.275}{4} = 1.81 \rightarrow \alpha_2 = 61^\circ \\ \operatorname{tg}_{\alpha_3} &= \frac{12.125}{4} = 3 \rightarrow \alpha_3 = 72^\circ \\ \operatorname{tg}_{\alpha_4} &= \frac{5.42}{4} = 1.35 \rightarrow \alpha_4 = 53^\circ \\ \operatorname{tg}_{\alpha_5} &= \frac{8.74}{4} = 2.18 \rightarrow \alpha_5 = 65^\circ \\ \operatorname{tg}_{\alpha_6} &= \frac{13.06}{4} = 3.26 \rightarrow \alpha_6 = 73^\circ \end{aligned}$$

По найденным углам находим силу света от каждого источника по таблице 4.2 (методом экстраполяции).

В методическом плане основным инструментом любого прогноза является метод экстраполяции.

Процедура экстраполяции – это чисто механический прием, следовательно, большое значение здесь имеет расчет доверительного интервала, т.е. диапазона отклонения полученной прогнозной оценки. Доверительный интервал рассчитывается двумя способами: формальным и неформальным. Формальный основан на применении специальных математических формул, а неформальный – на использовании экспертных оценок, заключений. В нашем случае мы будем использовать неформальный способ.

Т а б л и ц а 4.2 – Светораспределение светильника

Тип св-ка	Сила света I_α , кд в направлении угла α										
	0	5	15	25	35	45	55	65	75	85	90
ДРЛ	431	390	380	340	305	297	185	101	80	40	7

$$\alpha_1 = 31^\circ - I_{\alpha_1} = 200 \text{ кд};$$

$$\alpha_2 = 61^\circ - I_{\alpha_2} = 120 \text{ кд};$$

$$\alpha_3 = 72^\circ - I_{\alpha_3} = 97 \text{ кд};$$

$$\alpha_4 = 53^\circ - I_{\alpha_4} = 165 \text{ кд};$$

$$\alpha_5 = 65^\circ - I_{\alpha_5} = 101 \text{ кд};$$

$$\alpha_6 = 73^\circ - I_{\alpha_6} = 95 \text{ кд}.$$

Освещенность помещения относительно контрольной точки от каждого источника определяется по формуле (4.11).

$$E_\Gamma = \frac{I_1 * \cos^3 \alpha_1}{h_p^2} \quad (4.11)$$

$$E_1 = \frac{200 * 0.630}{16} = 7.88 \text{ лк};$$

$$E_2 = \frac{120 * 0.114}{16} = 0.86 \text{ лк};$$

$$E_3 = \frac{97 * 0.0294}{16} = 0.18 \text{ лк};$$

$$E_4 = \frac{165 * 0.218}{16} = 2.25 \text{ лк};$$

$$E_5 = \frac{101 * 0.075}{16} = 0,47 \text{ лк};$$

$$E_6 = \frac{95 * 0.0249}{16} = 0,15 \text{ лк}.$$

Суммарная освещенность вычисляется по формуле (4.12).

$$E = \frac{\mu * F}{1000 * K_3} * \sum_{i=1}^n E_i \quad (4.12)$$

где μ – коэффициент, учитывающий действие «удаленных» светильников (1,1 ÷ 1,2);

F – световой поток лампы (для ДРЛ-250 равен 13000);

K_3 – коэффициент запаса (для данного случая равен 1,4).

$$\sum_{i=1}^6 E_i = 11.79;$$

$$E = \frac{13000 * 1.1}{1000 * 1.4} * 11,79 = 120.43 \text{ лк}.$$

Нормированная освещенность составляет 300 лк, а получившаяся освещенность немного меньше. Освещенность на рабочем месте считается недостаточной, поэтому необходимо производить реконструкцию системы освещения производственного помещения.

Произведем реконструкцию, применяя метод коэффициента использования.

Определим индекс помещения по формуле (4.13).

$$i = \frac{L * B}{h_p * (L + B)} \quad (4.13)$$

$$i = \frac{20 * 30}{4 * (20 + 30)} = 3.$$

Коэффициент использования светового потока с лампами ДРЛ, учитывая найденный индекс помещения и заданные коэффициенты отражения: $\eta = 70\%$.

n – коэффициент использования, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен и потолка. Значение η определим по таблице коэффициентов использования различных светильников.

Рассчитаем необходимое количество ламп по формуле (4.14).

$$N = \frac{E * S * Z * K}{F * \eta} \quad (4.14)$$

где Z – коэффициент неравномерности освещения, равный $1,1 \div 1,2$;

K_z – коэффициент запаса, принимаемый равным $1,4$ для заданного типа помещения из таблицы.

$$N = \frac{300 * 600 * 1,1 * 1,4}{13000 * 0,7} = 30 \text{ (шт.)}$$

Увеличиваем количество ламп до 30. Теперь схема светильников будет иметь вид (рисунок 4.4):

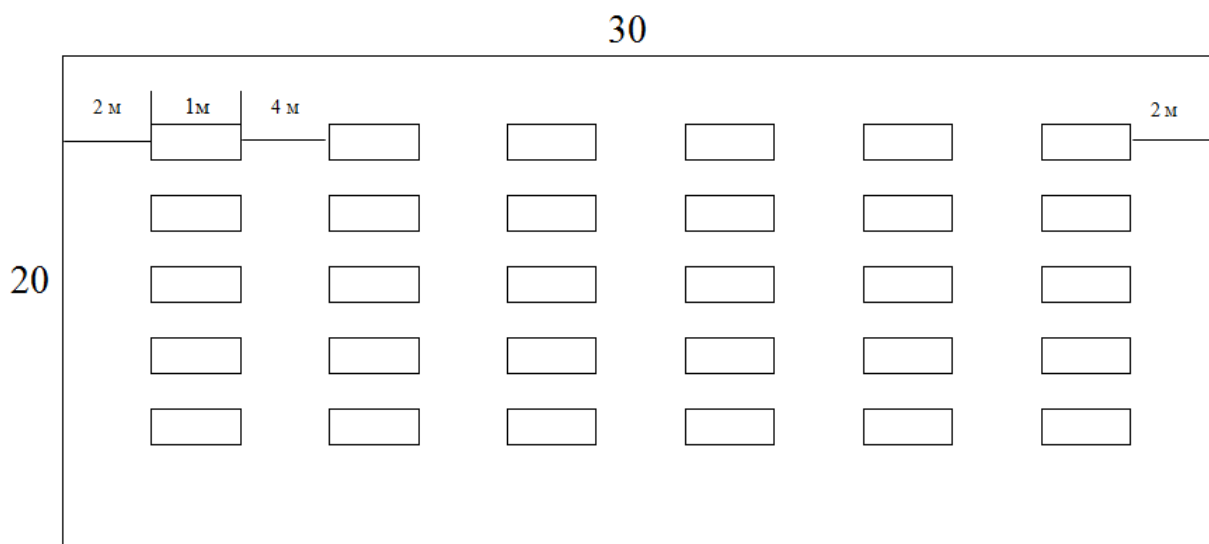


Рисунок 4.4 – Реконструированная схема расположения светильников

Для определения расстояний между светильниками необходимо учесть длину светильников: $a = 1$ м. Расстояние между светильниками будет равно:

$$L_a = \frac{30 - 6 * 1}{6} = 4 \text{ м.}$$

Расстояние между светильником и стеной будет равно:

$$l_a = \frac{4}{2} = 2 \text{ м.}$$

Для обеспечения необходимой освещенности помещения опенспейса с параметрами 20x30x5 необходимо увеличить количество светильников типа ДРЛ-250 с 18 до 30 штук.

4.5 Вывод

В данном разделе был произведен расчёт точечным методом, при заданном объёме помещения опенспейса, указанный тип светильников не применим для обеспечения необходимого уровня освещенности. Расчётная освещённость получается ниже установленной для разряда зрительной работы IV, а.

Расчет точечным методом позволяет делать анализ расчета на уровне номинальной освещенности, и основным недостатком этого метода является то, что нельзя сказать, насколько эффективно используются светильники.

Реконструкция освещенности дает обратную картину. Она позволяет определить, насколько эффективно и экономично можно использовать те или иные светильники, т.е. позволяет определить номинальную мощность.

Таким образом, была произведена реконструкция искусственного освещения производственного помещения путем увеличения количества светильников.

Заключение

Хранилище данных – прекрасный способ решения проблем, описанных в данном дипломном проекте. Можно считать, что хранилище данных расположено в центре всех ориентированных на приложения систем организации.

Многие компании на сегодняшний день используют разнообразные инструменты и приложения для бизнес - аналитики, позволяющие собирать и анализировать информацию из разных источников, а также обмениваться ею с пользователями. Однако, такие инструменты неудобны в применении, имеют функциональные ограничения, плохо масштабируются при увеличении объема данных и количества пользователей и сложны в управлении. В результате, руководители компаний приходят к выводу, что их организации все еще не располагают актуальной информацией, необходимой для принятия быстрых и

эффективных решений, несмотря на постоянные инвестиции в технологии бизнес-аналитики. Oracle Business Intelligence создан для того, чтобы удовлетворить требованиям в отношении корпоративных решений для бизнес-аналитики нового класса. Этот комплексный и интегрированный пакет аналитических инструментов разработан с целью обеспечить лучшее видение и понимание бизнеса широкому кругу пользователей и позволяет любому пользователю организации иметь быстрый Web-доступ к точной и оперативной, релевантной аналитике, необходимой для принятия важных и эффективных решений.

Бизнес-пользователи могут быстро и легко создавать отчеты с помощью Oracle BI Publisher, устанавливать формат данных для отчета, планировать задачи по формированию отчетов и их публикации, указывать способ вывода и канал доставки отчета, а также публиковать отчет в репозитории документов.

Использование Oracle BI Publisher:

- создание нового отчета. Новые отчеты создаются в web-интерфейсе. Разработчик отчета имеет доступ к различным информационным источникам, включая Oracle BI Server, и может сформировать запрос на данные, которые необходимо опубликовать в виде отчета.

- проектирование шаблона отчета. После того, как специфицированы данные для отчета, необходимо создать шаблон разметки отчета. Для разработки шаблон можно использовать стандартные офисные инструменты, такие как Microsoft Word, Adobe Acrobat или Microsoft Excel.

- формат и способы доставки отчетов. Для каждого отчета можно определить пользователей и роли, имеющие доступ к отчету, а также способы доставки отчетов. Пользователь может указать несколько шаблонов и каналов доставки для одного и того же отчета.

– генерация отчетов по расписанию. Дополнительно можно указать, будет ли отчет сформирован сразу же или позже, назначить генерацию отчета по расписанию, просматривать существующие отчеты и их историю.

Oracle BI Publisher является высокопроизводительным инструментом, позволяющим формировать отчеты разных форматов на основе информации из различных источников данных и распространять такие отчеты по различным каналам доставки. Он поддерживает высокопроизводительную печать, генерацию и распространение отчетов по расписанию.

В дипломном проекте с помощью инструментов SQL Developer рассматриваются возможности языка PL/SQL и написания пакетов и курсоров в рамках Oracle Database 11g, расширяющими возможности работы с хранилищами данных. Использование новых возможностей языка в работе с хранилищами данных: использование новых возможностей SQL Developer, таких как компиляция и поиск ошибок в PL/SQL и создание отчетов.

Перечень сокращений

BI – business intelligence.

DW – data warehouse.

FTP – file transfer protocol.

OBIEE – oracle business intelligence.

OLTP – online transaction processing.

OLAP – on-line analytical processing.

SRD – sample restructure delivery.

АБИС – автоматизированная банковская информационная система.

БД – база данных.

ВД – витрина данных.

ГК – главная книга.

ДРЛ – дуговая ртутная лампа.

КХД – корпоративное хранилище данных.

НДС – налог на добавленную стоимость

НСИ – нормативно справочная информация.

ОСД – оперативный склад данных.

СУБД – системы управления базами данных.

ТПС – типовой план счетов.

ФОТ – фонд оплаты труда.

ХД – хранилище данных.

ЦБ – ценные бумаги.

ПО – программное обеспечени.

Список литературы

1. Архипенков С.В., Голубев Д. К. Хранилища данных. От концепции до внедрения. – М.: Диалог-МИФИ, 2002. – 528 с.
2. Спирли Эрик. Корпоративные хранилища данных. Планирование, разработка, реализация. – М.: Изд.: Вильямс, 2001. – т. 1.
3. Сайт <http://datasql.ru/sqlandpol/11.htm>
4. Сайт http://www.tadviser.ru/index.php/Продукт:Oracle_Business_Intelligence
5. Николаева С.А. Принципы формирования и калькулирования себестоимости. – М.: Изд.: Аналитика, 2009.
6. Лазаревич М.И. Себестоимость продукции на предприятии / М. И. Лазаревич // Экономика. Финансы. Управление, 2008. № 2 с 35-40.
7. Марголин А.М. Инвестиции: Учебник. – М.: РАГС, 2010.
8. Ример М.И., Касатов А.Д. Экономическая оценка инвестиций. СПб.: – Изд.: «Питер», 2008.
9. Darl Kuhn, Sam Alapati, Bill Padfield. Expert Indexing in Oracle Database 11g: Maximum Performance for your Database. Publ.: Apress, 2011.
10. Ralph Kimball. The Data Warehouse Toolkit: Building the Web-Enabled Data Warehouse// John Wiley & Sons, 2000. № 28. 59 p.
11. Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат. – Алматы: АИЭС, 2009. – 20 с.
12. Аналитические системы на базе ORACLE Express OLAP. Проектирование, создание, сопровождение. – М.: Диалог-МИФИ, 2000. – 320 с.
13. Аносова Н.П., Бородин О.О. Распределенные базы и хранилища данных. – Томск: ИНТУИТ, 2009.
14. Аруп Нанда, Стивен Фейерштейн. Oracle PL/SQL для администраторов баз данных. – СПб.: Символ-Плюс, 2012. – 69 с.
15. Байзакова А.А., Санатова Т.С. Охрана труда. Методические указания к выполнению расчётно-графических работ (для студентов всех форм обучения всех специальностей). – Алматы: АИЭС, 2005.–17 с.
16. Гринвальд Р., Стаковьяк Р., Стерн Д. Oracle 11g. Основы. – СПб.: Символ-плюс, 2009. – т. 1.
17. Дональд Э. Кнут. Искусство программирования. Основные алгоритмы. 3-е изд.: – М.: Вильямс, 2000.
18. Сайт <http://www.firststeps.ru/sql/oracle/oracle1.html>
19. Санжей Мишра, Алан Бьюли. Секреты Oracle SQL. Изд.: Символ-Плюс, 2006.

Приложение А. Листинг созданных пакетов для отчетов

Создание пакета «Финансовые потоки и запасы»

```
create or replace package body REP_FIN_FLOW_RESERVE_perenos is

    -- ФИНАНСОВЫЕ АКТИВЫ
    function GetFinanceAssets(aBeginDate In Date, aEndDate In Date) return
    TFINANCE_FLOW_RESERVE_T is

vTable TFINANCE_FLOW_RESERVE_T := TFINANCE_FLOW_RESERVE_T();
    vRow TFINANCE_FLOW_RESERVE_R;
    v3 Number(12);
    v4 Number(12);
    v5 Number(12);
    v6 Number(12);
    v7 Number(12);
    v8 Number(12);
    v9 Number(25);
    v11 Number(12);
    v12 Varchar2(255);
    vBeginDate Date;
    vEndDate Date;

begin
select /*+ NO_PARALLEL*/ max(a.work_date)
    into vBeginDate
    from gl_working_date_d a
    where a.work_date < aBeginDate;

select /*+ NO_PARALLEL*/ max(a.work_date)
    into vEndDate
    from gl_working_date_d a
    where a.work_date <= aEndDate;

    vRow := new TFINANCE_FLOW_RESERVE_R(1, 0, null, 'Наличная валюта', null,
null, null, null, null, null, null, null, null, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(2, 1, null, '1. Национальная', null,
null, null, null, null, null, null, null, null, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
    into v3
    from gl_700_f f, gl_700_d d
    where f.work_date = vBeginDate
        and f.gl_700_id = d.id
        and substr(d.code_pokaz, 1, 5) in ('10011', '10021', '10031',
'10041', '10051')
        and substr(d.code_pokaz, 7, 1) = '1';

    vRow := new TFINANCE_FLOW_RESERVE_R(3, 2, 1, 'объем на начало отчетного
периода', null, null, v3, 0, 0, 0, 0, 0, 0, null, 0, null);
```

```

vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(4, 2, 2, 'изменения в результате
проведения операций', null, null, null, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(5, 2, 3, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('10011', '10021', '10031',
'10041', '10051')
and substr(d.code_pokaz, 7, 1) = '1';

vRow := new TFINANCE_FLOW_RESERVE_R(6, 2, 4, 'объем на конец отчетного
периода', null, null, v3, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(4);
vRow.format_no := 2;
vRow.national_bank := vTable(6).national_bank - vTable(3).national_bank;
vTable(4) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(7, 1, null, '2. Иностранная', null,
null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v11
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('10012', '10022', '10032',
'10042', '10052')
and substr(d.code_pokaz, 7, 1) != '1';

vRow := new TFINANCE_FLOW_RESERVE_R(8, 2, 5, 'объем на начало отчетного
периода', null, null, 0, 0, 0, 0, 0, 0, 0, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(9, 2, 6, 'изменения в результате
проведения операций', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ sum(case
when ga.code = '5703' then
ka.turn_debet_fact * -1
else
ka.turn_debet_fact
end)/1000

```

```

        into v11
    from gl_acc_f ka,
        gl_working_date_d wd,
        GL_PLAN_ACCOUNT_D ga
    where wd.work_date between aBeginDate and aEndDate
        and wd.work_date = ka.work_date
        and ka.turn_debet_fact <> 0
        and ka.gl_plan_account_id = ga.id
        and ga.code in ('4703', '5703');

    vRow := new TFINANCE_FLOW_RESERVE_R(10, 2, 7, 'оценочные изменения
стоимости', null, null, 0, 0, 0, 0, 0, 0, null, v11, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(11, 2, 8, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
    into v11
    from gl_700_f f, gl_700_d d
    where f.work_date = vEndDate
        and f.gl_700_id = d.id
        and substr(d.code_pokaz, 1, 5) in ('10012', '10022', '10032',
'10042', '10052')
        and substr(d.code_pokaz, 7, 1) != '1';

    vRow := new TFINANCE_FLOW_RESERVE_R(12, 2, 9, 'объем на конец отчетного
периода', null, null, 0, 0, 0, 0, 0, 0, null, v11, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := vTable(9);
    vRow.rest_of_world := vTable(12).rest_of_world - (vTable(8).rest_of_world
+ vTable(10).rest_of_world);
    vTable(9) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(13, 0, null, 'Переводимые депозиты',
null, null, null, null, null, null, null, null, null, null, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(14, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
    into v3
    from gl_700_f f, gl_700_d d
    where f.work_date = vBeginDate
        and f.gl_700_id = d.id
        and substr(code_pokaz, 1, 4) in ('1006', '1008', '1051', '1052',
'1705', '1053', '1104')
        and substr(code_pokaz, 6, 2) = '31';

    select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
    into v4
    from gl_700_f f, gl_700_d d

```

```

where f.work_date = vBeginDate
      and f.gl_700_id = d.id
      and substr(code_pokaz, 1, 4) in ('1006', '1008', '1051', '1052',
'1705', '1053', '1104')
      and substr(code_pokaz, 6, 2) = '41';
vRow := new TFINANCE_FLOW_RESERVE_R(15, 2, 10, 'объем на начало отчетного
периода', null, null, v3, v4, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(16, 2, 11, 'изменения в результате
проведения операций', null, null, null, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(17, 2, 12, 'оценочные изменения
стоимости', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(18, 2, 13, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(code_pokaz, 1, 4) in ('1006', '1008', '1051', '1052',
'1705', '1053', '1104')
      and substr(code_pokaz, 6, 2) = '31';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v4
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(code_pokaz, 1, 4) in ('1006', '1008', '1051', '1052',
'1705', '1053', '1104')
      and substr(code_pokaz, 6, 2) = '41';

vRow := new TFINANCE_FLOW_RESERVE_R(19, 2, 14, 'объем на конец отчетного
периода', null, null, v3, v4, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(20, 2, 15, 'доходы к получению',
null, null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(16);
vRow.format no := 2;
vRow.national_bank := vTable(19).national_bank -
vTable(15).national_bank;
vRow.bank := vTable(19).bank - vTable(15).bank;
vTable(16) := vRow;

```



```

vRow := new TFINANCE_FLOW_RESERVE_R(21, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(22, 2, 16, 'объем на начало отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(23, 2, 17, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(24, 2, 18, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(25, 2, 19, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(26, 2, 20, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(27, 2, 21, 'доходы к получению',
null, null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(28, 0, null, 'Другие депозиты', null,
null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(29, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work date = vBeginDate
and f.gl_700_id = d.id
and substr(code_pokaz, 1, 4) in ('1013', '1101', '1102', '1103',
'1104', '1251', '1252', '1253', '1254', '1255', '1256', '1257', '1264',
'1710', '1725', '1726', '1728')
and substr(code_pokaz, 6, 2) = '31';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)

```

```

into v4
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
      and f.gl_700_id = d.id
      and substr(code_pokaz, 1, 4) in ('1013', '1101', '1102', '1103',
'1104', '1251', '1252', '1253', '1254', '1255', '1256', '1257', '1264',
'1710', '1725', '1726', '1728')
      and substr(code_pokaz, 6, 2) = '41';

vRow := new TFINANCE_FLOW_RESERVE_R(30, 2, 22, 'объем на начало отчетного
периода', null, null, v3, v4, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(31, 2, 23, 'изменения в результате
проведения операций', null, null, null, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(32, 2, 24, 'оценочные изменения
стоимости', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(33, 2, 25, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(code_pokaz, 1, 4) in ('1013', '1101', '1102', '1103',
'1104', '1251', '1252', '1253', '1254', '1255', '1256', '1257', '1264',
'1710', '1725', '1726', '1728')
      and substr(code_pokaz, 6, 2) = '31';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v4
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(code_pokaz, 1, 4) in ('1013', '1101', '1102', '1103',
'1104', '1251', '1252', '1253', '1254', '1255', '1256', '1257', '1264',
'1710', '1725', '1726', '1728')
      and substr(code_pokaz, 6, 2) = '41';

vRow := new TFINANCE_FLOW_RESERVE_R(34, 2, 26, 'объем на конец отчетного
периода', null, null, v3, v4, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(code_pokaz, 1, 4) = '1710'
      and substr(code_pokaz, 6, 2) = '31';

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v4
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(code_pokaz, 1, 4) = '1725'
      and substr(code_pokaz, 6, 2) = '41';

vRow := new TFINANCE_FLOW_RESERVE_R(35, 2, 27, 'доходы к получению',
null, null, v3, v4, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(31);
vRow.format_no := 2;
vRow.national_bank := vTable(34).national_bank -
vTable(30).national_bank;
vRow.bank := vTable(34).bank - vTable(30).bank;
vTable(31) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(36, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(37, 2, 28, 'объем на начало отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(38, 2, 29, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(39, 2, 30, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(40, 2, 31, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(41, 2, 32, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(42, 2, 33, 'доходы к получению',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(43, 0, null, 'Ценные бумаги (за
исключением акции)', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(44, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

--34
select /*+ NO_PARALLEL*/ sum(case
      when substr(pok.code_pokaz,5,3) = '131' then dt.znac else 0
    end) summ3,
sum(case
      when substr(pok.code_pokaz,5,3) = '141' then dt.znac else 0
    end) summ4,
sum(case
      when substr(pok.code_pokaz,5,3) = '151' then dt.znac else 0
    end) summ5,
sum(case
      when substr(pok.code_pokaz,5,3) = '191' then dt.znac else 0
    end) summ6,
sum(case
      when substr(pok.code_pokaz,5,3) = '181' then dt.znac else 0
    end) summ7,
sum(case
      when substr(pok.code_pokaz,5,3) in('111','121') then dt.znac
else 0
      end) summ8,
sum(case
      when substr(pok.code_pokaz,5,3) in('161','171') then dt.znac
else 0
      end) summ9,
sum(case
      when substr(pok.code_pokaz,5,1) = '2' then dt.znac else 0
    end) summl1
into v3, v4, v5, v6, v7, v8, v9, v11
from stat.s_pokaz@stat pok,
stat.detail@stat dt
where pok.id_form = 15949
      and pok.dat_beg <= vBeginDate and (pok.dat_end is null or
pok.dat_end > vBeginDate)
      and pok.id = dt.id_pokaz
      and dt.d_report = vBeginDate
      and
      substr(pok.code_pokaz,1,4)
in('1201','1202','1405','1406','1425','1452','1459','1481','1485','1744','174
5','1746','1750','1752');

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(45, 2, 34, 'объем на начало отчетного
периода', null, null, nvl(v3,0), nvl(v4,0), nvl(v5,0), nvl(v6,0), nvl(v7,0),
nvl(v8,0), nvl(v9,0), null, nvl(v11,0), null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(46, 2, 35, 'нетто покупка (включая
новые выпуски) за отчетный период', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(47, 2, 36, 'разница в оценке
стоимости', null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--37
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code = '7130'
and la.code_system in(2468,2537,2551,2552);

select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) - v3 summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) - v4 summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) - v5 summ5,
round(sum(case

```

```

                                when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
                                end)/1000) - v6 summ6,
round(sum(case
                                when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
                                end)/1000) - v7 summ7,
round(sum(case
                                when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
                                end)/1000) - v8 summ8,
round(sum(case
                                when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
                                end)/1000) - v9 summ9,
round(sum(case
                                when substr(la.code,5,1) = '2' then ka.kzt_output else
0
                                end)/1000) - v11 summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code = '7130'
and la.code_system in(2468,2537,2551,2552);

vRow := new TFINANCE_FLOW_RESERVE_R(48, 2, 37, 'другие изменения (укажите
вид)', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--38
select /*+ NO_PARALLEL*/ sum(case
                                when substr(pok.code_pokaz,5,3) = '131' then dt.znac else 0
                                end) summ3,
sum(case
                                when substr(pok.code_pokaz,5,3) = '141' then dt.znac else 0
                                end) summ4,
sum(case
                                when substr(pok.code_pokaz,5,3) = '151' then dt.znac else 0
                                end) summ5,
sum(case
                                when substr(pok.code_pokaz,5,3) = '191' then dt.znac else 0
                                end) summ6,
sum(case
                                when substr(pok.code_pokaz,5,3) = '181' then dt.znac else 0
                                end) summ7,
sum(case
                                when substr(pok.code_pokaz,5,3) in('111','121') then dt.znac
else 0
                                end) summ8,
sum(case
                                when substr(pok.code_pokaz,5,3) in('161','171') then dt.znac
else 0
                                end) summ9,
sum(case
                                when substr(pok.code_pokaz,5,1) = '2' then dt.znac else 0
                                end) summ11
into v3, v4, v5, v6, v7, v8, v9, v11

```

```

from stat.s_pokaz@stat pok,
stat.detail@stat dt
where pok.id_form = 15949 --H700
and pok.dat_beg <= vEndDate and (pok.dat_end is null or pok.dat_end
> vEndDate)
and pok.id = dt.id_pokaz
and dt.d_report = vEndDate
and
substr(pok.code_pokaz,1,4)
in('1201','1202','1405','1406','1425','1452','1459','1481','1485','1744','174
5','1746','1750','1752');

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(49, 2, 38, 'объем на конец отчетного
периода', null, null, nvl(v3,0), nvl(v4,0), nvl(v5,0), nvl(v6,0), nvl(v7,0),
nvl(v8,0), nvl(v9,0), null, nvl(v11,0), null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := vTable(46);
vRow.format_no := 2;
vRow.national_bank := vTable(49).national_bank -
vTable(48).national_bank - vTable(45).national_bank;
vRow.bank := vTable(49).bank -
vTable(48).bank - vTable(45).bank;
vRow.other_finance_org := vTable(49).other_finance_org -
vTable(48).other_finance_org - vTable(45).other_finance_org;
vRow.housekeeping_sector := vTable(49).housekeeping_sector -
vTable(48).housekeeping_sector - vTable(45).housekeeping_sector;
vRow.nku := vTable(49).nku -
vTable(48).nku - vTable(45).nku;
vRow.government_sector := vTable(49).government_sector -
vTable(48).government_sector - vTable(45).government_sector;
vRow.non_finance_sector := vTable(49).non_finance_sector -
vTable(48).non_finance_sector - vTable(45).non_finance_sector;
vRow.rest_of_world := vTable(49).rest_of_world -
vTable(48).rest_of_world - vTable(45).rest_of_world;
vTable(46) := vRow;

```

```

--39
select /*+ NO_PARALLEL*/ sum(case
when substr(pok.code_pokaz,5,3) = '131' then dt.znac else 0
end) summ3,
sum(case
when substr(pok.code_pokaz,5,3) = '141' then dt.znac else 0
end) summ4,
sum(case
when substr(pok.code_pokaz,5,3) = '151' then dt.znac else 0
end) summ5,
sum(case
when substr(pok.code_pokaz,5,3) = '191' then dt.znac else 0
end) summ6,
sum(case
when substr(pok.code_pokaz,5,3) = '181' then dt.znac else 0
end) summ7,
sum(case
when substr(pok.code_pokaz,5,3) in('111','121') then dt.znac
else 0
end) summ8,
sum(case
when substr(pok.code_pokaz,5,3) in('161','171') then dt.znac
else 0
end) summ9,
sum(case
when substr(pok.code_pokaz,5,1) = '2' then dt.znac else 0

```

```

        end) summl1
into v3, v4, v5, v6, v7, v8, v9, v11
from stat.s_pokaz@stat pok,
     stat.detail@stat dt
where pok.id_form = 15949 --H700
     and pok.dat_beg <= vEndDate and (pok.dat_end is null or pok.dat_end
> vEndDate)
     and pok.id = dt.id_pokaz
     and dt.d_report = vEndDate
     and substr(pok.code_pokaz,1,4) in('1746','1750');

vRow := new TFINANCE_FLOW_RESERVE_R(50, 2, 39, 'проценты (дивиденды) к
получению за отчетный период', null, null, nvl(v3,0), nvl(v4,0), nvl(v5,0),
nvl(v6,0), nvl(v7,0), nvl(v8,0), nvl(v9,0), null, nvl(v11,0), null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(51, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(52, 2, 40, 'объем на начало отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(53, 2, 41, 'нетто покупка (включая
новые выпуски) за отчетный период', null, null, null, null, null, null,
null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(54, 2, 42, 'разница в оценке
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(55, 2, 43, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(56, 2, 44, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(57, 2, 45, 'проценты (дивиденды) к
получению за отчетный период', null, null, null, null, null, null, null,
null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(58, 0, null, 'Кредиты и займы', null,
null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```



```

vRow := new TFINANCE_FLOW_RESERVE_R(59, -1, null, '1. Краткосрочные',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(60, 1, null, '1.1 В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--46
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and (ga.code in('1461','1748')
or
(ga.code
in('1411','1301','1302','1303','1304','1305','1306','1309','1321','1322','132
3','1326',
'1327','1328','1401','1403','1407','1409','1420','1421','1422','1423','1424',
'1429',
'1462','1491','1494','1730','1733','1734','1740','1741','1757','1771','1772',
'1855'))

```

```

*
        and exists(select /*+ NO_PARALLEL INDEX(a GL_ACC_F_FK5)*/
                    from gl_acc_f a,
                        gl_plan_account_d b
                    where a.work_date = vBeginDate
                          and a.kzt_output <> 0
                          and a.gl_plan_account_id = b.id
                          and b.code = '1411'
                          and          a.gl_nat_person_id          =
ka.gl_nat_person_id));

    vRow := new TFINANCE_FLOW_RESERVE_R(61, 2, 46, 'объем на начало отчетного
периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(62, 2, 47, 'изменения в результате
проведения операций', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(63, 2, 48, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(64, 2, 49, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

--50
select /*+ NO_PARALLEL*/ round(sum(case
                                when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
                                end)/1000) summ3,
round(sum(case
            when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
            end)/1000) summ4,
round(sum(case
            when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
            end)/1000) summ5,
round(sum(case
            when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
            end)/1000) summ6,
round(sum(case
            when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
            end)/1000) summ7,
ka.kzt_output else 0
round(sum(case
            when substr(la.code,5,3) in('111','121') then
end)/1000) summ8,
round(sum(case
            when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
            end)/1000) summ9,

```

```

round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and (ga.code in('1461','1748')
or
in('1411','1301','1302','1303','1304','1305','1306','1309','1321','1322','1323','1326',
'1327','1328','1401','1403','1407','1409','1420','1421','1422','1423','1424',
'1429',
'1462','1491','1494','1730','1733','1734','1740','1741','1757','1771','1772',
'1855')
and exists (select /* NO_PARALLEL INDEX(a GL_ACC_F_FK5)*/ *
from gl_acc_f a,
gl_plan_account_d b
where a.work_date = vEndDate
and a.kzt_output <> 0
and a.gl_plan_account_id = b.id
and b.code = '1411'
and a.gl_nat_person_id =
ka.gl_nat_person_id)));

vRow := new TFINANCE_FLOW_RESERVE_R(65, 2, 50, 'объем на конец отчетного
периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(62);
vRow.NATIONAL_BANK := vTable(65).NATIONAL_BANK -
vTable(61).NATIONAL_BANK;
vRow.BANK := vTable(65).BANK - vTable(61).BANK;
vRow.OTHER_FINANCE_ORG := vTable(65).OTHER_FINANCE_ORG -
vTable(61).OTHER_FINANCE_ORG;
vRow.HOUSEKEEPING_SECTOR := vTable(65).HOUSEKEEPING_SECTOR -
vTable(61).HOUSEKEEPING_SECTOR;
vRow.NKU := vTable(65).NKU - vTable(61).NKU;
vRow.GOVERNMENT_SECTOR := vTable(65).GOVERNMENT_SECTOR -
vTable(61).GOVERNMENT_SECTOR;
vRow.NON_FINANCE_SECTOR := vTable(65).NON_FINANCE_SECTOR -
vTable(61).NON_FINANCE_SECTOR;
vRow.REST_OF_WORLD := vTable(65).REST_OF_WORLD -
vTable(61).REST_OF_WORLD;
vTable(62) := vRow;

--51
select /* NO PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) summ4,

```

```

round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from
gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and (ga.code in('1748')
or (ga.code in('1740','1741')
and exists(select /*+ NO_PARALLEL INDEX(a GL_ACC_F_FK5)*/ *
from gl_acc_f a,
gl_plan_account_d b
where a.work_date = vEndDate
and a.kzt_output <> 0
and a.gl_plan_account_id = b.id
and a.gl_nat_person_id =
ka.gl_nat_person_id
and b.code = '1411')));

vRow := new TFINANCE_FLOW_RESERVE_R(66, 2, 51, 'проценты к получению за
отчетный период (по графику)', null, null, v3, v4, v5, v6, v7, v8, v9, null,
v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

-- 52
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) - v3 summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) - v4 summ4,
round(sum(case

```

```

                                when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
                                end)/1000) - v5 summ5,
round(sum(case
                                when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
                                end)/1000) - v6 summ6,
round(sum(case
                                when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
                                end)/1000) - v7 summ7,
round(sum(case
                                when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
                                end)/1000) - v8 summ8,
round(sum(case
                                when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
                                end)/1000) - v9 summ9,
round(sum(case
                                when substr(la.code,5,1) = '2' then ka.kzt_output else
0
                                end)/1000) - v11 summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and (ga.code in('1748')
or (ga.code in('1740','1741')
and exists(select /*+ NO_PARALLEL INDEX(a GL_ACC_F_FK5)*/
*
                                from gl_acc_f a,
gl_plan_account_d b
                                where a.work_date =
to_date('30.12.2011','dd.mm.yyyy')
                                and a.kzt_output <> 0
                                and a.gl_plan_account_id = b.id
                                and b.code = '1411'
                                and a.gl_nat_person_id =
ka.gl_nat_person_id)));

select /*+ NO_PARALLEL*/
round(sum(decode(ga.code,'4465',ka.turn_credit_fact,0))/1000) + v5,
round(sum(decode(ga.code,'4411',ka.turn_credit_fact,0))/1000) + v6
into v5, v6
from gl_acc_f ka,
gl_working_date_d wd,
gl_plan_account_d ga
where wd.work_date between aBeginDate and aEndDate
and wd.work_date = ka.work_date
and (ka.turn_debet_fact <> 0 or ka.turn_credit_fact <> 0)
and ka.gl_plan_account_id = ga.id
and ga.code in('4411','4465')
and ga.name not like '%штраф, пеня%';

vRow := new TFINANCE_FLOW_RESERVE_R(67, 2, 52, 'проценты фактически
полученные', null, null, 0, 0, v5, v6, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(68, 1, null, '1.2 В иностранной
валюте', null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(69, 2, 53, 'объем на начало отчетного
периода', null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(70, 2, 54, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(71, 2, 55, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(72, 2, 56, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(73, 2, 57, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(74, 2, 58, 'проценты к получению за
отчетный период (по графику)', null, null, null, null, null, null, null,
null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(75, 2, 59, 'проценты фактически
полученные', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(76, -1, null, '2. Долгосрочные',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(77, 1, null, '2.1 В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--60
select /*+ NO_PARALLEL*/ round(sum(case

```

```

                when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
                end)/1000) summ3,
round(sum(case
                when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
                end)/1000) summ4,
round(sum(case
                when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
                end)/1000) summ5,
round(sum(case
                when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
                end)/1000) summ6,
round(sum(case
                when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
                end)/1000) summ7,
round(sum(case
                when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
                end)/1000) summ8,
round(sum(case
                when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
                end)/1000) summ9,
round(sum(case
                when substr(la.code,5,1) = '2' then ka.kzt_output else
0
                end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
     gl_plan_account_d ga,
     gl_ledger_account_d la
where ka.work_date = vBeginDate
     and ka.kzt_output <> 0
     and ka.gl_plan_account_id = ga.id
     and la.id = ka.gl_ledger_account_id
     and ga.code
in('1417','1301','1302','1303','1304','1305','1306','1309','1321','1322','132
3','1326',

'1327','1328','1401','1403','1407','1409','1420','1421','1422','1423','1424',
'1429',

'1462','1491','1494','1730','1733','1734','1740','1741','1757','1771','1772',
'1855')
     and not exists(select /* NO_PARALLEL INDEX(a GL_ACC_F_FK5)*/ *
                    from gl_acc_f a,
                    gl_plan_account_d b
                    where a.work_date = vBeginDate
                    and a.kzt_output <> 0
                    and a.gl_plan_account_id = b.id
                    and b.code = '1411'
                    and a.gl_nat_person_id =
ka.gl_nat_person_id);

vRow := new TFINANCE_FLOW_RESERVE_R(78, 2, 60, 'объем на начало отчетного
периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(79, 2, 61, 'изменения в результате
проведения операций', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

dbms_output.put_line(v3 || ' ' || v4 || ' ' || v5 || ' ' || v6 || '
' || v7 || ' ' || v8 || ' ' || v9 || ' ' || v11); -- проверка текущего
значения в строке

vRow := new TFINANCE_FLOW_RESERVE_R(80, 2, 62, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--63
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
ka.kzt_output else 0
when substr(la.code,5,3) in('111','121') then
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code = '7130'
and la.code_system not in(2468,2537,2551,2552);

select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) - v3 summ3,

```



```

round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) - v4 summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) - v5 summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) - v6 summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) - v7 summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) - v8 summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) - v9 summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) - v11 summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code = '7130'
and la.code_system not in(2468,2537,2551,2552);

v12 := null;
if (v6 != 0) then
v12 := 'Другие изменения в объеме финансовых активов, не отнесенные к
другим категориям - списание кредиторами безнадежных долгов';
end if;

vRow := new TFINANCE_FLOW_RESERVE_R(81, 2, 63, 'другие изменения (укажите
вид)', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, v12);
vTable.Extend;
vTable(vTable.Count) := vRow;

-- 64
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,

```

```

round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code
in('1417','1301','1302','1303','1304','1305','1306','1309','1321','1322','132
3','1326',
'1327','1328','1401','1403','1407','1409','1420','1421','1422','1423','1424',
'1429',
'1462','1491','1494','1730','1733','1734','1740','1741','1757','1771','1772',
'1855')
and not exists(select /* NO_PARALLEL INDEX(a GL_ACC_F_FK5)*/
*
from gl_acc_f a,
gl_plan_account_d b
where a.work_date =
to_date('29.12.2012','dd.mm.yyyy')
and a.kzt_output <> 0
and a.gl_plan_account_id = b.id
and b.code = '1411'
and a.gl_nat_person_id =
ka.gl_nat_person_id);

vRow := new TFINANCE_FLOW_RESERVE_R(82, 2, 64, 'объем на конец отчетного
периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(79);
vRow.NATIONAL_BANK := vTable(82).NATIONAL_BANK -
vTable(78).NATIONAL_BANK - vTable(81).NATIONAL_BANK;
vRow.BANK := vTable(82).BANK -
vTable(78).BANK - vTable(81).BANK;
vRow.OTHER_FINANCE_ORG := vTable(82).OTHER_FINANCE_ORG -
vTable(78).OTHER_FINANCE_ORG - vTable(81).OTHER_FINANCE_ORG;

```

```

vRow.HOUSEKEEPING_SECTOR      := vTable(82).HOUSEKEEPING_SECTOR -
vTable(78).HOUSEKEEPING_SECTOR - vTable(81).HOUSEKEEPING_SECTOR;
vRow.NKU                       := vTable(82).NKU -
vTable(78).NKU                 - vTable(81).NKU;
vRow.GOVERNMENT_SECTOR        := vTable(82).GOVERNMENT_SECTOR -
vTable(78).GOVERNMENT_SECTOR - vTable(81).GOVERNMENT_SECTOR;
vRow.NON_FINANCE_SECTOR       := vTable(82).NON_FINANCE_SECTOR -
vTable(78).NON_FINANCE_SECTOR - vTable(81).NON_FINANCE_SECTOR;
vRow.REST_OF_WORLD            := vTable(82).REST_OF_WORLD -
vTable(78).REST_OF_WORLD      - vTable(81).REST_OF_WORLD;
vTable(79) := vRow;

--65
select /*+ NO_PARALLEL*/ round(sum(case
      when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
      end)/1000) summ3,
round(sum(case
      when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
      end)/1000) summ4,
round(sum(case
      when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
      end)/1000) summ5,
round(sum(case
      when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
      end)/1000) summ6,
round(sum(case
      when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
      end)/1000) summ7,
round(sum(case
      when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
      end)/1000) summ8,
round(sum(case
      when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
      end)/1000) summ9,
round(sum(case
      when substr(la.code,5,1) = '2' then ka.kzt_output else
0
      end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code in('1740','1741')
and not exists (select /*+ NO_PARALLEL INDEX(a GL_ACC_F_FK5)*/ *
      from gl_acc_f a,
      gl_plan_account_d b
      where a.work_date = vEndDate
      and a.kzt_output <> 0
      and a.gl_plan_account_id = b.id
      and b.code = '1411'
      and a.gl_nat_person_id =
ka.gl_nat_person_id);

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(83, 2, 65, 'проценты к получению за
отчетный период (по графику)', null, null, v3, v4, v5, v6, v7, v8, v9, null,
v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;
--66
select /*+ NO_PARALLEL*/ round(sum(case
                                when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
                                end)/1000) - v3 summ3,
round(sum(case
            when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
            end)/1000) - v4 summ4,
round(sum(case
            when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
            end)/1000) - v5 summ5,
round(sum(case
            when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
            end)/1000) - v6 summ6,
round(sum(case
            when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
            end)/1000) - v7 summ7,
ka.kzt_output else 0
            when substr(la.code,5,3) in('111','121') then
            end)/1000) - v8 summ8,
round(sum(case
            when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
            end)/1000) - v9 summ9,
round(sum(case
            when substr(la.code,5,1) = '2' then ka.kzt_output else
0
            end)/1000) - v11 summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code in('1740','1741')

and not exists(select /*+ NO_PARALLEL INDEX(a GL_ACC_F_FK5)*/
a.gl_nat_person_id
                from gl_acc_f a,
                gl_plan_account_d b
                where a.work_date = vEndDate
                and a.kzt_output <> 0
                and a.gl_plan_account_id = b.id
                and b.code = '1411'
                and a.gl_nat_person_id =
ka.gl_nat_person_id);

select /*+ NO_PARALLEL*/ round(sum(ka.turn_credit_fact)/1000) + v6
into v6
from gl_acc_f ka,

```

```

        gl_working_date_d wd,
        gl_plan_account_d ga
    where wd.work_date between aBeginDate and aEndDate
        and wd.work_date = ka.work_date
        and (ka.turn_debet_fact <> 0 or ka.turn_credit_fact <> 0)
        and ka.gl_plan_account_id = ga.id
        and ga.code = '4417'
        and ga.name not like '%штраф, пеня%';

    vRow := new TFINANCE_FLOW_RESERVE_R(84, 2, 66, 'проценты фактически
полученные', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(85, 1, null, '2.2 В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(86, 2, 67, 'объем на начало отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(87, 2, 68, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(88, 2, 69, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(89, 2, 70, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(90, 2, 71, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(91, 2, 72, 'проценты к получению за
отчетный период (по графику)', null, null, null, null, null, null, null,
null, null, null, null, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(92, 2, 73, 'проценты фактически
полученные', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(93, 0, null, 'Производные финансовые
инструменты', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(94, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(95, 2, 74, 'объем на начало отчетного
периода', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(96, 2, 75, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(97, 2, 76, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(98, 2, 77, 'другие изменения (укажите
вид)', null, null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(99, 2, 78, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(100, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(101, 2, 79, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(102, 2, 80, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(103, 2, 81, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null, null,
null);

```

```

vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(104, 2, 82, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(105, 2, 83, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(106, 0, null, 'Акции и другие формы
участия в капитале', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(107, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(108, 2, 84, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(109, 2, 85, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(110, 2, 86, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(111, 2, 87, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(112, 2, 88, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(113, 2, 89, 'доходы к получению',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(114, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(115, 2, 90, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(116, 2, 91, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(117, 2, 92, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(118, 2, 93, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(119, 2, 94, 'объем на конец отчетного
периода', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(120, 2, 95, 'доходы к получению',
null, null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(121, 0, null, 'Другая дебиторская
задолженность (требования)', null, null, null, null, null, null, null, null,
null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(122, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--96
select /*+ NO_PARALLEL*/ round(sum(case
                                when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
                                end)/1000) summ3,
round(sum(case
            when substr(la.code,5,3) = '141'
            or (ga.code = '1265' and substr(la.code,5,3) =
'000') then ka.kzt_output else 0
            end)/1000) summ4,

```



```

round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191'
or (ga.code in('1434','1854')) and
substr(la.code,5,3) = '000') then ka.kzt_output else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121')
or (ga.code = '1857' and substr(la.code,5,3) =
'000') then ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code
in('1105','1106','1205','1206','1265','1266','1267','1312','1313','1330','133
1','1353',
'1432','1433','1434','1435','1445','1453','1454','1482','1483','1492','1493',
'1551',
'1552','1735','1747','1749','1755','1756','1792','1793','1799','1811','1812',
'1813',
'1814','1815','1816','1817','1818','1819','1820','1821','1822','1823','1824',
'1825',
'1826','1827','1831','1832','1833','1834','1835','1836','1837','1838','1839',
'1840',
'1841','1842','1843','1844','1851','1852','1853','1854','1856','1857','1860',
'1861',
'1864','1867','1870','1879','1880','1894');

vRow := new TFINANCE_FLOW_RESERVE_R(123, 2, 96, 'объем на начало
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(124, 2, 97, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;

```

```

vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(125, 2, 98, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(126, 2, 99, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--100
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141'
or (ga.code = '1265' and substr(la.code,5,3) =
'000') then ka.kzt_output else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191'
or (ga.code in('1434','1854') and
substr(la.code,5,3) = '000') then ka.kzt_output else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121')
or (ga.code = '1857' and substr(la.code,5,3) =
'000') then ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la,
gl_account_d gl
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and gl.id = ka.gl_account_id
and length(gl.code) = 20
and ka.work_date between gl.begin_date and gl.end_date
and ka.work_date between ga.begin_date and ga.end_date

```

```

        and ka.work_date between la.begin_date and la.end_date
        and
        ga.code
in('1105','1106','1205','1206','1265','1266','1267','1312','1313','1330','133
1','1353',

'1432','1433','1434','1435','1445','1453','1454','1482','1483','1492','1493',
'1551',

'1552','1735','1747','1749','1755','1756','1792','1793','1799','1811','1812',
'1813',

'1814','1815','1816','1817','1818','1819','1820','1821','1822','1823','1824',
'1825',

'1826','1827','1831','1832','1833','1834','1835','1836','1837','1838','1839',
'1840',

'1841','1842','1843','1844','1851','1852','1853','1854','1856','1857','1860',
'1861',

        '1864','1867','1870','1879','1880','1894');

    vRow := new TFINANCE_FLOW_RESERVE_R(127, 2, 100, 'объем на конец
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := vTable(124);
    vRow.format_no := 2;
    vRow.national_bank := vTable(127).national_bank -
vTable(123).national_bank;
    vRow.bank := vTable(127).bank - vTable(123).bank;
    vRow.other_finance_org := vTable(127).other_finance_org -
vTable(123).other_finance_org;
    vRow.housekeeping_sector := vTable(127).housekeeping_sector -
vTable(123).housekeeping_sector;
    vRow.nku := vTable(127).nku - vTable(123).nku;
    vRow.government_sector := vTable(127).government_sector -
vTable(123).government_sector;
    vRow.non_finance_sector := vTable(127).non_finance_sector -
vTable(123).non_finance_sector;
    vRow.rest_of_world := vTable(127).rest_of_world -
vTable(123).rest_of_world;
    vTable(124) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(128, 2, 101, 'проценты к получению за
отчетный период', null, null, null, null, null, null, null, null, null,
null, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(129, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

    vRow := new TFINANCE_FLOW_RESERVE_R(130, 2, 102, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null);
    vTable.Extend;
    vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(131, 2, 103, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(132, 2, 104, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(133, 2, 105, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(134, 2, 106, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(135, 2, 107, 'проценты к получению за
отчетный период', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(136, 3, null, 'Справочно (без
разбивки по секторам)', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(137, 0, 108, 'специальные резервы
(проекции) на покрытие убытков', null, null, null, null, null, null,
null, null, null, null, null);

select /*+ NO_PARALLEL*/ (sum(a.output_active_fact) -
sum(a.output_passive_fact)) / 1000
into vRow.total
from gl_acc_f a,
gl_plan_account_d b
where a.work_date = vEndDate
and a.gl_plan_account_id = b.id
and b.code in ('1428', '1451', '1878', '3300', '3301', '3302',
'3303', '3304', '3305', '3306', '3307', '3308', '3309', '3310', '3311',
'3312', '3313', '3314', '3315', '3316')
and b.is_last = 1;

vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(138, 0, 109, 'счета положительной
корректировки стоимости финансовых инструментов', null, null, null,
null, null, null, null, null, null, null);

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into vRow.total
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate

```

```

        and substr(d.code_pokaz, 1, 4) in ('1208', '1260', '1262', '1310',
'1324', '1430', '1456', '2035', '2045', '2055', '2128', '2136', '2233',
'2235');

vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(139, 0, 110, 'счета отрицательной
корректировки стоимости финансовых инструментов', null, null, null, null,
null, null, null, null, null, null, null);

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into vRow.total
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(d.code_pokaz, 1, 4) in ('1209', '1261', '1263', '1311',
'1325', '1431', '1457', '2037', '2047', '2065', '2129', '2137', '2234',
'2236');

vTable.Extend;
vTable(vTable.Count) := vRow;
for i in 1..135 loop
    vRow := vTable(i);
    vRow.finance_sector      := vRow.national_bank      + vRow.bank      +
vRow.other_finance_org;
    vRow.whole_economical := vRow.finance_sector + vRow.housekeeping_sector
+ vRow.nku + vRow.government_sector + vRow.non_finance_sector;
    vRow.total := vRow.whole_economical + vRow.rest_of_world;
    vTable(i) := vRow;
end loop;

return vTable;
end;

-- ФИНАНСОВЫЕ ПАССИВЫ
function GetFinanceLiabilities(aBeginDate In Date, aEndDate In Date) return
TFINANCE_FLOW_RESERVE_T is
vTable TFINANCE_FLOW_RESERVE_T := TFINANCE_FLOW_RESERVE_T();
vRow TFINANCE_FLOW_RESERVE_R;
v3 Number(12);
v4 Number(12);
v5 Number(12);
v6 Number(12);
v7 Number(12);
v8 Number(12);
v9 Number(12);
v11 Number(12);
vTemp Number(12);
vBeginDate Date;
vEndDate Date;

begin
select /*+ NO_PARALLEL*/ max(a.work_date)
into vBeginDate
from gl_working_date_d a
where a.work_date < aBeginDate;

select /*+ NO_PARALLEL*/ max(a.work_date)
into vEndDate
from gl_working_date_d a
where a.work_date <= aEndDate;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(140, 0, null, 'Переводимые депозиты',
null, null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(141, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '31';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v4
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '41';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v5
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '51';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v6
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '91';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v7
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '81';

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v8
from gl_700_f f, gl_700_d d
where f.work_date =vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '11';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v9
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and (substr(d.code_pokaz, 6, 2) = '61' or substr(d.code_pokaz, 6,
2) = '71');

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v11
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20112', '20122', '20132',
'20142', '22022', '22032', '22042', '22092', '22212', '27012', '27182',
'27262', '27482')
and substr(d.code_pokaz, 7, 1) = '1';

vRow := new TFINANCE_FLOW_RESERVE_R(142, 2, 111, 'объем на начало
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(143, 2, 112, 'изменения в результате
проведения операций', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(144, 2, 113, 'оценочные изменения
стоимости', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(145, 2, 114, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '31';

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v4
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '41';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v5
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '51';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v6
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '91';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v7
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '81';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v8
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and substr(d.code_pokaz, 6, 2) = '11';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v9
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20111', '20121', '20131',
'20141', '22021', '22031', '22041', '22091', '22211', '27011', '27181',
'27261', '27481')
and (substr(d.code_pokaz, 6, 2) = '61' or substr(d.code_pokaz, 6,
2) = '71');

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)

```



```

into v11
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(d.code_pokaz, 1, 5) in ('20112', '20122', '20132',
'20142', '22022', '22032', '22042', '22092', '22212', '27012', '27182',
'27262', '27482')
      and substr(d.code_pokaz, 7, 1) = '1';

vRow := new TFINANCE_FLOW_RESERVE_R(146, 2, 115, 'объем на конец
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(147, 2, 116, 'расходы к оплате',
null, null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(4);
vRow.format_no := 2;
vRow.national_bank := vTable(7).national_bank - vTable(3).national_bank;
vRow.bank := vTable(7).bank - vTable(3).bank;
vRow.other_finance_org := vTable(7).other_finance_org -
vTable(3).other_finance_org;
vRow.housekeeping_sector := vTable(7).housekeeping_sector -
vTable(3).housekeeping_sector;
vRow.nku := vTable(7).nku - vTable(3).nku;
vRow.government_sector := vTable(7).government_sector -
vTable(3).government_sector;
vRow.non_finance_sector := vTable(7).non_finance_sector -
vTable(3).non_finance_sector;
vRow.rest_of_world := vTable(7).rest_of_world - vTable(3).rest_of_world;
vTable(4) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(148, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(149, 2, 117, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(150, 2, 118, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(151, 2, 119, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(152, 2, 120, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;

```

```

vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(153, 2, 121, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(154, 2, 122, 'расходы к оплате',
null, null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(155, 0, null, 'Другие депозиты',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(156, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '31';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v4
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '41';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v5
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',

```

```

'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
    and substr(d.code_pokaz, 6, 2) = '51';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v6
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
    and f.gl_700_id = d.id
    and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
    and substr(d.code_pokaz, 6, 2) = '91';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v7
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
    and f.gl_700_id = d.id
    and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
    and substr(d.code_pokaz, 6, 2) = '81';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v8
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
    and f.gl_700_id = d.id
    and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
    and substr(d.code_pokaz, 6, 2) = '11';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v9
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
    and f.gl_700_id = d.id
    and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
    and (substr(d.code_pokaz, 6, 2) = '61' or substr(d.code_pokaz, 6,
2) = '71');

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)

```

```

into v11
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20162', '20212', '20222',
'20232', '20242', '21112', '21132', '21212', '21222', '21232', '21242',
'21252', '21272', '21302', '21332', '21352', '21382', '22012', '22052',
'22062', '22072', '22082', '22112', '22122', '22132', '22152', '22162',
'22172', '22192', '22232', '22242', '22262', '22322', '22372', '22452',
'27112', '27122', '27132', '27142', '27172', '27192', '27202', '27212',
'27232', '27422', '27432', '27462', '27472', '28552')
and substr(d.code_pokaz, 7, 1) = '1' ;

vRow := new TFINANCE_FLOW_RESERVE_R(157, 2, 123, 'объем на начало
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(158, 2, 124, 'изменения в результате
проведения операций', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(159, 2, 125, 'оценочные изменения
стоимости', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(160, 2, 126, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '31';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v4
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '41';

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v5
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '51';

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v6
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '91';

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v7
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '81';

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v8
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
and substr(d.code_pokaz, 6, 2) = '11';

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v9
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id

```

```

        and substr(d.code_pokaz, 1, 5) in ('20161', '20211', '20221',
'20231', '20241', '21111', '21131', '21211', '21221', '21231', '21241',
'21251', '21271', '21301', '21331', '21351', '21381', '22011', '22051',
'22061', '22071', '22081', '22111', '22121', '22131', '22151', '22161',
'22171', '22191', '22231', '22241', '22261', '22321', '22371', '22451',
'27111', '27121', '27131', '27141', '27171', '27191', '27201', '27211',
'27231', '27421', '27431', '27461', '27471', '28551')
        and (substr(d.code_pokaz, 6, 2) = '61' or substr(d.code_pokaz, 6,
2) = '71');

```

```

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v11
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(d.code_pokaz, 1, 5) in ('20162', '20212', '20222',
'20232', '20242', '21112', '21132', '21212', '21222', '21232', '21242',
'21252', '21272', '21302', '21332', '21352', '21382', '22012', '22052',
'22062', '22072', '22082', '22112', '22122', '22132', '22152', '22162',
'22172', '22192', '22232', '22242', '22262', '22322', '22372', '22452',
'27112', '27122', '27132', '27142', '27172', '27192', '27202', '27212',
'27232', '27422', '27432', '27462', '27472', '28552')
      and substr(d.code_pokaz, 7, 1) = '1' ;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(161, 2, 127, 'объем на конец
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(162, 2, 128, 'расходы к оплате',
null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := vTable(19);
vRow.format_no := 2;
vRow.national_bank := vTable(22).national_bank -
vTable(18).national_bank;
vRow.bank := vTable(22).bank - vTable(18).bank;
vRow.other_finance_org := vTable(22).other_finance_org -
vTable(18).other_finance_org;
vRow.housekeeping_sector := vTable(22).housekeeping_sector -
vTable(18).housekeeping_sector;
vRow.nku := vTable(22).nku - vTable(18).nku;
vRow.government_sector := vTable(22).government_sector -
vTable(18).government_sector;
vRow.non_finance_sector := vTable(22).non_finance_sector -
vTable(18).non_finance_sector;
vRow.rest_of_world := vTable(22).rest_of_world -
vTable(18).rest_of_world;
vTable(19) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(163, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(164, 2, 129, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(165, 2, 130, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(166, 2, 131, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(167, 2, 132, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(168, 2, 133, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(169, 2, 134, 'расходы к оплате',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(170, 0, null, 'Ценные бумаги (за
исключением акции)', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(171, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v9
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
and (substr(d.code_pokaz, 6, 2) = '61' or substr(code_pokaz, 6, 2)
= '71');

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v11
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23012', '23032', '23062',
'24052', '24062', '27302', '27442', '27562', '28632')
and substr(d.code_pokaz, 6, 2) = '1';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v3

```

```

from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
and substr(d.code_pokaz, 6, 2) = '31';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v4
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
and substr(d.code_pokaz, 6, 2) = '41';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v5
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
and substr(d.code_pokaz, 6, 2) = '51';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v6
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
and substr(d.code_pokaz, 6, 2) = '91';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v7
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
and substr(d.code_pokaz, 6, 2) = '81';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v8
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
and substr(d.code_pokaz, 6, 2) = '11';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v9
from gl_700_f f, gl_700_d d
where f.work_date = vBeginDate
and f.gl_700_id = d.id
and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
and (substr(d.code_pokaz, 6, 2) = '61' or substr(d.code_pokaz, 6,
2) = '71');

```



```

        select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
        into v11
        from gl_700_f f, gl_700_d d
        where f.work_date = vBeginDate
              and f.gl_700_id = d.id
              and substr(d.code_pokaz, 1, 5) in ('23012', '23032', '23062',
'24052', '24062', '27302', '27442', '27562', '28632')
              and substr(d.code_pokaz, 7, 1) = '1';

        vRow := new TFINANCE_FLOW_RESERVE_R(172, 2, 135, 'объем на начало
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
        vTable.Extend;
        vTable(vTable.Count) := vRow;

        vRow := new TFINANCE_FLOW_RESERVE_R(173, 2, 136, 'покупка (включая новые
выпуски) за отчетный период', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0,
null);
        vTable.Extend;
        vTable(vTable.Count) := vRow;

        vRow := new TFINANCE_FLOW_RESERVE_R(174, 2, 137, 'продажа (включая выкуп)
за отчетный период', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
        vTable.Extend;
        vTable(vTable.Count) := vRow;

        vRow := new TFINANCE_FLOW_RESERVE_R(175, 2, 138, 'разница в оценке
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
        vTable.Extend;
        vTable(vTable.Count) := vRow;

        vRow := new TFINANCE_FLOW_RESERVE_R(176, 2, 139, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null, null,
null, null);
        vTable.Extend;
        vTable(vTable.Count) := vRow;

        select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
        into v3
        from gl_700_f f, gl_700_d d
        where f.work_date = vEndDate
              and f.gl_700_id = d.id
              and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
              and substr(d.code_pokaz, 6, 2) = '31';

        select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
        into v4
        from gl_700_f f, gl_700_d d
        where f.work_date = vEndDate
              and f.gl_700_id = d.id
              and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
              and substr(d.code_pokaz, 6, 2) = '41';

        select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
        into v5
        from gl_700_f f, gl_700_d d
        where f.work_date = vEndDate
              and f.gl_700_id = d.id
              and substr(d.code_pokaz, 1, 5) in('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')

```

```

        and substr(d.code_pokaz, 6, 2) = '51';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v6
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(d.code_pokaz, 1, 5) in('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
      and substr(d.code_pokaz, 6, 2) = '91';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v7
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
      and substr(d.code_pokaz, 6, 2) = '81';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v8
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
      and substr(d.code_pokaz, 6, 2) = '11';

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v9
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(d.code_pokaz, 1, 5) in ('23011', '23031', '23061',
'24051', '24061', '27301', '27441', '27561', '28631')
      and (substr(d.code_pokaz, 6, 2) = '61' or substr(d.code_pokaz, 6,
2) = '71');

select /*+ NO_PARALLEL*/ nvl(sum(val), 0)
into v11
from gl_700_f f, gl_700_d d
where f.work_date = vEndDate
      and f.gl_700_id = d.id
      and substr(d.code_pokaz, 1, 5) in ('23012', '23032', '23062',
'24052', '24062', '27302', '27442', '27562', '28632')
      and substr(d.code_pokaz, 7, 1) = '1';

vRow := new TFINANCE_FLOW_RESERVE_R(177, 2, 140, 'объем на конец
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(34);
vRow.format no := 2;
vRow.national_bank := vTable(38).national_bank -
vTable(33).national_bank;
vRow.bank := vTable(38).bank - vTable(33).bank;
vRow.other_finance_org := vTable(38).other_finance_org -
vTable(33).other_finance_org;
vRow.housekeeping_sector := vTable(38).housekeeping_sector -
vTable(33).housekeeping_sector;

```

```

vRow.nku := vTable(38).nku - vTable(33).nku;
vRow.government_sector := vTable(38).government_sector -
vTable(33).government_sector;
vRow.non_finance_sector := vTable(38).non_finance_sector -
vTable(33).non_finance_sector;
vRow.rest_of_world := vTable(38).rest_of_world -
vTable(33).rest_of_world;
vTable(34) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(178, 2, 141, 'проценты (дивиденды) к
уплате за отчетный период', null, null, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(179, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(180, 2, 142, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(181, 2, 143, 'покупка (включая новые
выпуски) за отчетный период', null, null, null, null, null, null, null,
null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(182, 2, 144, 'продажа (включая выкуп)
за отчетный период', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(183, 2, 145, 'разница в оценке
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(184, 2, 146, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(185, 2, 147, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(186, 2, 148, 'проценты (дивиденды) к
уплате за отчетный период', null, null, null, null, null, null, null,
null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(187, 0, null, 'Кредиты и займы',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(188, -1, null, '1. Краткосрочные',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(189, 1, null, '1.1 В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--149
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code in('2255','2725');

vRow := new TFINANCE_FLOW_RESERVE_R(190, 2, 149, 'объем на начало
отчетного периода', null, null, nvl(v3,0), nvl(v4,0), nvl(v5,0), nvl(v6,0),
nvl(v7,0), nvl(v8,0), nvl(v9,0), null, nvl(v11,0), null);
vTable.Extend;

```

```

vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(191, 2, 150, 'изменения в результате
проведения операций', null, null, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(192, 2, 151, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(193, 2, 152, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--153
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code in('2255','2725');

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(194, 2, 153, 'объем на конец
отчетного
периода', null, null, nvl(v3,0), nvl(v4,0), nvl(v5,0), nvl(v6,0), nvl(v7,0), nvl(v8,0
), nvl(v9,0), null, nvl(v11,0), null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(52);
vRow.format_no := 2;
vRow.national_bank := vTable(55).national_bank -
vTable(51).national_bank;
vRow.bank := vTable(55).bank - vTable(51).bank;
vRow.other_finance_org := vTable(55).other_finance_org -
vTable(51).other_finance_org;
vRow.housekeeping_sector := vTable(55).housekeeping_sector -
vTable(51).housekeeping_sector;
vRow.nku := vTable(55).nku - vTable(51).nku;
vRow.government_sector := vTable(55).government_sector -
vTable(51).government_sector;
vRow.non_finance_sector := vTable(55).non_finance_sector -
vTable(51).non_finance_sector;
vRow.rest_of_world := vTable(55).rest_of_world -
vTable(51).rest_of_world;
vTable(52) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(195, 2, 154, 'проценты к уплате за
отчетный период (по графику)', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(196, 2, 155, 'проценты фактически
уплаченные', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(197, 1, null, '1.2 В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(198, 2, 156, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(199, 2, 157, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(200, 2, 158, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(201, 2, 159, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null, null,
null, null);

```

```

vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(202, 2, 160, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(203, 2, 161, 'проценты к уплате за
отчетный период (по графику)', null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(204, 2, 162, 'проценты фактически
уплаченные', null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(205, -1, null, '2. Долгосрочные',
null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(206, 1, null, '2.1 В национальной
валюте', null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--163
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
ka.kzt_output else 0
when substr(la.code,5,3) in('111','121') then
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case

```

```

                                when substr(la.code,5,1) = '2' then ka.kzt_output else
0
                                end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
     gl_plan_account_d ga,
     gl_ledger_account_d la
where ka.work_date = to_date('30.12.2011','dd.mm.yyyy')
      and ka.kzt_output <> 0
      and ka.gl_plan_account_id = ga.id
      and la.id = ka.gl_ledger_account_id
      and
                                                                ga.code
in('2034','2036','2038','2041','2042','2044','2046','2048','2051','2052','205
4','2056',

'2057','2058','2059','2064','2066','2067','2068','2227','2230','2231','2401',
'2402',

'2451','2703','2704','2705','2706','2707','2722','2740','2741','2745','2757')
;
vRow := new TFINANCE_FLOW_RESERVE_R(207, 2, 163, 'объем на начало
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(208, 2, 164, 'изменения в результате
проведения операций', null, null, 0, 0, 0, 0, 0, 0, 0, null, 0, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(209, 2, 165, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(210, 2, 166, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--167
select /*+ NO_PARALLEL*/ round(sum(case
                                when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
                                end)/1000) summ3,
round(sum(case
                                when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
                                end)/1000) summ4,
round(sum(case
                                when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
                                end)/1000) summ5,
round(sum(case
                                when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
                                end)/1000) summ6,
round(sum(case
                                when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
                                end)/1000) summ7,

```



```

        round(sum(case
            when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
            end)/1000) summ8,
        round(sum(case
            when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
            end)/1000) summ9,
        round(sum(case
            when substr(la.code,5,1) = '2' then ka.kzt_output else
0
            end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
     gl_plan_account_d ga,
     gl_ledger_account_d la
where ka.work_date = vEndDate
     and ka.kzt_output <> 0
     and ka.gl_plan_account_id = ga.id
     and la.id = ka.gl_ledger_account_id
     and ga.code
in('2034','2036','2038','2041','2042','2044','2046','2048','2051','2052','205
4','2056',

'2057','2058','2059','2064','2066','2067','2068','2227','2230','2231','2401',
'2402',

'2451','2703','2704','2705','2706','2707','2722','2740','2741','2745','2757')
;

vRow := new TFINANCE_FLOW_RESERVE_R(211, 2, 167, 'объем на конец
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := vTable(69);
vRow.format_no := 2;
vRow.national_bank := vTable(72).national_bank -
vTable(68).national_bank;
vRow.bank := vTable(72).bank - vTable(68).bank;
vRow.other_finance_org := vTable(72).other_finance_org -
vTable(68).other_finance_org;
vRow.housekeeping_sector := vTable(72).housekeeping_sector -
vTable(68).housekeeping_sector;
vRow.nku := vTable(72).nku - vTable(68).nku;
vRow.government_sector := vTable(72).government_sector -
vTable(68).government_sector;
vRow.non_finance_sector := vTable(72).non_finance_sector -
vTable(68).non_finance_sector;
vRow.rest_of_world := vTable(72).rest_of_world -
vTable(68).rest_of_world;
vTable(69) := vRow;

--168
select /*+ NO PARALLEL*/ round(sum(case
            when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
            end)/1000) summ3,
        round(sum(case
            when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
            end)/1000) summ4,

```

```

round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code in('2703');

vRow := new TFINANCE_FLOW_RESERVE_R(212, 2, 168, 'проценты к уплате за
отчетный период (по графику)', null, null, v3, v4, v5, v6, v7, v8, v9, null,
v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--169
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) - v3 summ3,
round(sum(case
when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
end)/1000) - v4 summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) - v5 summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) - v6 summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) - v7 summ7,
round(sum(case

```

```

        when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
        end)/1000) - v8 summ8,
        round(sum(case
        when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
        end)/1000) - v9 summ9,
        round(sum(case
        when substr(la.code,5,1) = '2' then ka.kzt_output else
0
        end)/1000) - v11 summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code in('2703');

select /*+ NO_PARALLEL*/ v8 + round(sum(ka.turn_credit_fact)/1000)
into v8
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la,
gl_account_d gl
where ka.work_date = vEndDate
and ka.turn_credit_fact <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code = '5036'
and gl.id = ka.gl_account_id
and ka.work_date between gl.begin_date and gl.end_date
and gl.name like '%Правительства%';

select /*+ NO_PARALLEL*/ v9 + round(sum(ka.turn_credit_fact)/1000)
into v9
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la,
gl_account_d gl
where ka.work_date = vEndDate
and ka.turn_credit_fact <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code = '5036'
and gl.id = ka.gl_account_id
and ka.work_date between gl.begin_date and gl.end_date
and gl.name like '%Сампук%';

vRow := new TFINANCE_FLOW_RESERVE_R(213, 2, 169, 'проценты фактически
уплаченные', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(214, 1, null, '2.2 В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(215, 2, 170, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(216, 2, 171, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(217, 2, 172, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(218, 2, 173, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(219, 2, 174, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(220, 2, 175, 'проценты к уплате за
отчетный период (по графику)', null, null, null, null, null, null, null,
null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(221, 2, 176, 'проценты фактически
уплаченные', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(222, 0, null, 'Производные финансовые
инструменты', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(223, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(224, 2, 177, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(225, 2, 178, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);

```

```

vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(226, 2, 179, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(227, 2, 180, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(228, 2, 181, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(229, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(230, 2, 182, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(231, 2, 183, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(232, 2, 184, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(233, 2, 185, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(234, 2, 186, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(235, 0, null, 'Акции и другие формы
участия в капитале', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(236, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--187
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141'
or (substr(la.code,6,1) = '0' and
substr(la.code,5,1) <> '2') then ka.kzt_output else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code
in('3001','3003','3025','3027','3101','3510','3540','3561','3580','3589','359
9'));

vRow := new TFINANCE_FLOW_RESERVE_R(237, 2, 187, 'объем на начало
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(238, 2, 188, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := new TFINANCE_FLOW_RESERVE_R(239, 2, 189, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(240, 2, 190, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--191
select /*+ NO_PARALLEL*/ round(sum(case
when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
end)/1000) summ3,
round(sum(case
when substr(la.code,5,3) = '141'
or (substr(la.code,6,1) = '0' and
substr(la.code,5,1) <> '2') then ka.kzt_output else 0
end)/1000) summ4,
round(sum(case
when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
end)/1000) summ5,
round(sum(case
when substr(la.code,5,3) = '191' then ka.kzt_output
else 0
end)/1000) summ6,
round(sum(case
when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
end)/1000) summ7,
round(sum(case
when substr(la.code,5,3) in('111','121') then
ka.kzt_output else 0
end)/1000) summ8,
round(sum(case
when substr(la.code,5,3) in('161','171') then
ka.kzt_output else 0
end)/1000) summ9,
round(sum(case
when substr(la.code,5,1) = '2' then ka.kzt_output else
0
end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vEndDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and ga.code
in('3001','3003','3025','3027','3101','3510','3540','3561','3580','3589','359
9');

vRow := new TFINANCE_FLOW_RESERVE_R(241, 2, 191, 'объем на конец
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

```

```

vRow := vTable(99);
vRow.format_no := 2;
vRow.national_bank := vTable(102).national_bank -
vTable(98).national_bank;
vRow.bank := vTable(102).bank - vTable(98).bank;
vRow.other_finance_org := vTable(102).other_finance_org -
vTable(98).other_finance_org;
vRow.housekeeping_sector := vTable(102).housekeeping_sector -
vTable(98).housekeeping_sector;
vRow.nku := vTable(102).nku - vTable(98).nku;
vRow.government_sector := vTable(102).government_sector -
vTable(98).government_sector;
vRow.non_finance_sector := vTable(102).non_finance_sector -
vTable(98).non_finance_sector;
vRow.rest_of_world := vTable(102).rest_of_world -
vTable(98).rest_of_world;
vTable(99) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(242, 2, 192, 'расходы к оплате',
null, null, null, null, null, null, null, null, null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(243, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(244, 2, 193, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(245, 2, 194, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(246, 2, 195, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(247, 2, 196, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(248, 2, 197, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(249, 2, 198, 'расходы к оплате',
null, null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;

```



```

vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(250, 0, null, 'Другая кредиторская
задолженность', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(251, 1, null, '1. В национальной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--199
select /*+ NO_PARALLEL*/ round(sum(case
                                when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
                                end)/1000) summ3,
round(sum(case
            when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
            end)/1000) summ4,
round(sum(case
            when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
            end)/1000) summ5,
round(sum(case
            when substr(la.code,5,3) = '191'
              or (ga.code in('2854','2861') and
substr(la.code,5,3) = '000') then ka.kzt_output else 0
            end)/1000) summ6,
round(sum(case
            when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
            end)/1000) summ7,
round(sum(case
            when substr(la.code,5,3) in('111','121')
              or (ga.code in('2857') and
substr(la.code,5,3)='000') then ka.kzt_output else 0
            end)/1000) summ8,
round(sum(case
            when substr(la.code,5,3) in('161','171')
              or (ga.code in('2070') and substr(la.code,5,3) =
'000') then ka.kzt_output else 0
            end)/1000) summ9,
round(sum(case
            when substr(la.code,5,1) = '2' then ka.kzt_output else
0
            end)/1000) summ11
into v3, v4, v5, v6, v7, v8, v9, v11
from gl_acc_f ka,
gl_plan_account_d ga,
gl_ledger_account_d la
where ka.work_date = vBeginDate
and ka.kzt_output <> 0
and ka.gl_plan_account_id = ga.id
and la.id = ka.gl_ledger_account_id
and
ga.code
in('2069','2070','2131','2139','2140','2151','2152','2153','2210','2225','222
8','2237',

```

```

'2238','2239','2240','2304','2305','2403','2404','2551','2552','2715','2731',
'2749',

'2755','2770','2792','2793','2794','2799','2811','2812','2813','2814','2815',
'2816',

'2817','2818','2819','2820','2831','2832','2833','2834','2835','2836','2838',
'2839',

'2851','2852','2853','2854','2856','2857','2860','2861','2862','2864','2867',
'2868',

                                '2869','2870','2880','2894');

vRow := new TFINANCE_FLOW_RESERVE_R(252, 2, 199, 'объем на начало
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(253, 2, 200, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(254, 2, 201, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(255, 2, 202, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

--203
select /*+ NO_PARALLEL*/ round(sum(case
                                when substr(la.code,5,3) = '131' then ka.kzt_output
else 0
                                end)/1000) summ3,
round(sum(case
                                when substr(la.code,5,3) = '141' then ka.kzt_output
else 0
                                end)/1000) summ4,
round(sum(case
                                when substr(la.code,5,3) = '151' then ka.kzt_output
else 0
                                end)/1000) summ5,
round(sum(case
                                when substr(la.code,5,3) = '191'
                                or      (ga.code      in('2854','2861'))      and
substr(la.code,5,3) = '000') then ka.kzt_output else 0
                                end)/1000) summ6,
round(sum(case
                                when substr(la.code,5,3) = '181' then ka.kzt_output
else 0
                                end)/1000) summ7,
round(sum(case
                                when substr(la.code,5,3) in('111','121')

```

```

                                or          (ga.code          in('2857'))          and
substr(la.code,5,3)='000') then ka.kzt_output else 0
                                end)/1000) summ8,
                                round(sum(case
                                    when substr(la.code,5,3) in('161','171')
                                        or (ga.code in('2070')) and substr(la.code,5,3) =
'000') then ka.kzt_output else 0
                                    end)/1000) summ9,
                                round(sum(case
                                    when substr(la.code,5,1) = '2' then ka.kzt_output else
0
                                    end)/1000) summ11
                                into v3, v4, v5, v6, v7, v8, v9, v11
                                from gl_acc_f ka,
                                    gl_plan_account_d ga,
                                    gl_ledger_account_d la
                                where ka.work_date = vEndDate
                                    and ka.kzt_output <> 0
                                    and ka.gl_plan_account_id = ga.id
                                    and la.id = ka.gl_ledger_account_id
                                    and
                                ga.code
in('2069','2070','2131','2139','2140','2151','2152','2153','2210','2225','222
8','2237',

'2238','2239','2240','2304','2305','2403','2404','2551','2552','2715','2731',
'2749',

'2755','2770','2792','2793','2794','2799','2811','2812','2813','2814','2815',
'2816',

'2817','2818','2819','2820','2831','2832','2833','2834','2835','2836','2838',
'2839',

'2851','2852','2853','2854','2856','2857','2860','2861','2862','2864','2867',
'2868',

                                '2869','2870','2880','2894');

                                vRow := new TFINANCE_FLOW_RESERVE_R(256, 2, 203, 'объем на конец
отчетного периода', null, null, v3, v4, v5, v6, v7, v8, v9, null, v11, null);
                                vTable.Extend;
                                vTable(vTable.Count) := vRow;

                                vRow := vTable(114);
                                vRow.format_no := 2;
                                vRow.national_bank := vTable(117).national_bank -
vTable(113).national_bank;
                                vRow.bank := vTable(117).bank - vTable(113).bank;
                                vRow.other_finance_org := vTable(117).other_finance_org -
vTable(113).other_finance_org;
                                vRow.housekeeping_sector := vTable(117).housekeeping_sector -
vTable(113).housekeeping_sector;
                                vRow.nku := vTable(117).nku - vTable(113).nku;
                                vRow.government_sector := vTable(117).government_sector -
vTable(113).government_sector;
                                vRow.non_finance_sector := vTable(117).non_finance_sector -
vTable(113).non_finance_sector;
                                vRow.rest_of_world := vTable(117).rest_of_world -
vTable(113).rest_of_world;
                                vTable(114) := vRow;

                                vRow := new TFINANCE_FLOW_RESERVE_R(257, 2, 204, 'проценты к уплате за
отчетный период', null, null, null, null, null, null, null, null, null, null, null, null);

```

```

vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(258, 1, null, '2. В иностранной
валюте', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(259, 2, 205, 'объем на начало
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(260, 2, 206, 'изменения в результате
проведения операций', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(261, 2, 207, 'оценочные изменения
стоимости', null, null, null, null, null, null, null, null, null, null,
null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(262, 2, 208, 'другие изменения
(укажите вид)', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(263, 2, 209, 'объем на конец
отчетного периода', null, null, null, null, null, null, null, null,
null, null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

vRow := new TFINANCE_FLOW_RESERVE_R(264, 2, 210, 'проценты к уплате за
отчетный период', null, null, null, null, null, null, null, null, null,
null, null);
vTable.Extend;
vTable(vTable.Count) := vRow;

for i in 1..vTable.Count loop
vRow := vTable(i);
vRow.finance_sector := vRow.national_bank + vRow.bank +
vRow.other_finance_org;
vRow.whole_economical := vRow.finance_sector + vRow.housekeeping_sector
+ vRow.nku + vRow.government_sector + vRow.non_finance_sector;
vRow.total := vRow.whole_economical + vRow.rest_of_world;
vTable(i) := vRow;
end loop;

return vTable;
end;

begin
-- Initialization
null;
end REP_FIN_FLOW_RESERVE_perenos;

```

Создание типа таблицы строк

```
CREATE OR REPLACE TYPE "TFINANCE_FLOW_RESERVE_T" as table of
TFINANCE_FLOW_RESERVE_R
```

Создание типа строки

```
CREATE OR REPLACE TYPE "TFINANCE_FLOW_RESERVE_R"
as object
(
  ROW_NO          Number(3),
  FORMAT_NO       Number(1),
  RECORD_NO       Number(3),
  RECORD_NAME     Varchar2(255),
  TOTAL           Number(12),
  FINANCE_SECTOR  Number(12),
  NATIONAL_BANK   Number(12),
  BANK            Number(12),
  OTHER_FINANCE_ORG Number(12),
  HOUSEKEEPING_SECTOR Number(12),
  NKU             Number(12),
  GOVERNMENT_SECTOR Number(12),
  NON_FINANCE_SECTOR Number(12),
  WHOLE_ECONOMICAL Number(12),
  REST_OF_WORLD   Number(12),
  OTHERS_EXPANSION Varchar2(255)
)
```

Создание пакета «Сведения по остаткам на балансовых счетах за вычетом специальных резервов (проvizий)»

```
create or replace package body REP_AG006 is

function get_report(p_date date) return T_AG006_TAB is
  my_res T_AG006_TAB := T_AG006_TAB();
  my_row T_AG006_ROW := T_AG006_ROW(null, null, null, null);

cursor emp_cur is
  select rc.column_num,
         rc.column_name,
         sum((select sum(case
                    when cc.calculation_sign = '+' then
                      abs(gf.val)
                    else
                      abs(gf.val) * (-1)
                    end) val
              from gl_700_f gf, gl_700_d gd
              where gf.work_date = p_date
                    and gf.gl_700_id = cc.calculation_id
                    and gd.id = cc.calculation_id)) val
  from set_rep_column_calculation cc, set_report_column rc
  where rc.report_id = cc.report_id
        and cc.column_id = rc.column_id
        and rc.report_id = 2
        and cc.report_id = rc.report_id
  group by rc.column_id,
           rc.column_num,
           rc.column_name
  order by rc.column_id;

val_1 number := 0;
```

```

val_2 number := 0;
val_3 number := 0;
val_4 number := 0;
val_5 number := 0;
val_6 number := 0;
val_7 number := 0;
val_8 number := 0;
val_9 number := 0;
val_10 number := 0;
val_11 number := 0;
val_12 number := 0;
val_13 number := 0;

begin
  --ИТОГИ--
  for rec in emp_cur loop

    if trim(rec.column_num) in ('3.1') then --3
      val_1 := val_1 + rec.val;
    end if;
    if rec.column_num in ('4.1') then --4
      val_2 := val_2 + rec.val;
    end if;
    if rec.column_num in ('11.2') then --11
      val_3 := val_3 + rec.val;
    end if;
    if rec.column_num in ('28.1.1') then --28.1
      val_5 := val_5 + rec.val;
    end if;
    if rec.column_num in ('28.4.1', '28.4.2') then --28.4
      val_6 := val_6 + rec.val;
    end if;
    if rec.column_num in ('30', '31') then --35
      val_7 := val_7 + rec.val;
    end if;
    if rec.column_num in ('40', '41', '46', '44') then --39
      val_8 := val_8 + rec.val;
    end if;
    if rec.column_num in ('37') then --23
      val_9 := val_9 + rec.val;
    end if;
    if rec.column_num in ('42.1','42.2','42.3') then --42
      val_12 := val_12 + rec.val;
    end if;
    if rec.column_num in ('14','15','18','19','20') then --АКТИВЫ
      val_13 := val_13 + rec.val;
    end if;
  end loop;
  val_10 := val_5 + val_6;           --28
  val_7 := val_7 + val_10;         --35
  val_9 := val_9 + val_7;         --23
  val_8 := val_8 + val_12;        --39
  val_11 := val_9 + val_8;        --49
  val_13 := val_13 + val_4;       --АКТИВЫ
  for rec in emp_cur loop
    my row := new T AG006 ROW(p date,
                             rec.column_num,
                             rec.column_name,
                             nvl(case
                                when rec.column_num = '3' then
                                  val_1
                                when rec.column_num = '4' then
                                  val_2

```

```

        when rec.column_num = '11' then
            val_3
        when rec.column_num = '23' then
            val_9
        when rec.column_num = '28' then
            val_10
        when rec.column_num = '28.1' then
            val_5
        when rec.column_num = '28.4' then
            val_6
        when rec.column_num = '35' then
            val_7
        when rec.column_num = '39' then
            val_8
            when rec.column_num = '42' then
                val_12
        when rec.column_num = '49' then
            val_11
        when trim(rec.column_name) = 'Активы' then
            val_13
        else rec.val
    end,0));

    my_res.extend;
    my_res(my_res.count) := my_row;
end loop;
return my_res;
end;
end REP_AG006;

```

Создание типа таблицы строк

```
CREATE OR REPLACE TYPE "T_AG006_TAB" as table of T_AG006_ROW
```

Создание типа строки

```
CREATE OR REPLACE TYPE "T_AG006_ROW" as object
(
    aDate    Varchar2(2000),
    pos      varchar2(10),
    name     varchar2(2000),
    a2       number
)

```

Создание пакета «Основные показатели баланса»

```

create or replace package REP_AG041 is
    function get_report(p_date date) return T_AG041_TAB;
end REP_AG041;

create or replace package body REP_AG041 is
    function get_report(p_date in date) return T_AG041_TAB is
        my_res T_AG041_TAB := T_AG041_TAB();
        my_row T_AG041_ROW := T_AG041_ROW(NULL, NULL, NULL, NULL, NULL);

        cursor emp_cur is
            select rc.column_num,
                   rc.parent_column,
                   rc.column_name,
                   nvl((case
                       when rc.column_id not in
                           (11, 19, 30, 31, 32, 34, 35, 41, 42, 43) then
                           sum((select sum(case

```

```

        when cc.calculation_sign = '+' then
            gf.val
        else
            gf.val * (-1)
        end) val
    from gl_700_f gf, gl_700_d gd
    where gf.work_date = p_date
        and gf.gl_700_id = cc.calculation_id
        and gd.id = cc.calculation_id)
when rc.column_id = 11 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f gf,
            gl_account_d gl,
            gl_plan_account_d pl
        where gf.gl_account_id = gl.id
            and gf.work_date = p_date
            and gl.name like '%НДС%'
            and pl.id = gf.gl_plan_account_id
            and pl.code = '1851'))
when rc.column_id = 19 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f gf,
            gl_account_d gl,
            gl_plan_account_d pl
        where gf.gl_account_id = gl.id
            and gf.work_date = p_date
            and gl.name like '%НДС%'
            and pl.id = gf.gl_plan_account_id
            and pl.code = '2851'))
when rc.column_id = 30 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f gf,
            gl_account_d gl,
            gl_plan_account_d pl
        where gf.gl_account_id = gl.id
            and gf.work_date = p_date
            and gl.name like '%Центр%Кредит%'
            and pl.id = gf.gl_plan_account_id
            and pl.code = '4266'))
when rc.column_id = 31 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f gf,
            gl_account_d gl,
            gl_plan_account_d pl
        where gf.gl_account_id = gl.id
            and gf.work_date = p_date
            and gl.name like '%БТА%'
            and pl.id = gf.gl_plan_account_id
            and pl.code = '4266'))
when rc.column_id = 32 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f gf,
            gl_account_d gl,
            gl_plan_account_d pl
        where gf.gl_account_id = gl.id
            and gf.work date = p date
            and gl.name like '%Цесна%'
            and pl.id = gf.gl_plan_account_id
            and pl.code = '4266'))
when rc.column_id = 34 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f gf,
            gl_account_d gl,

```



```

        gl_plan_account_d pl
    where gf.gl_account_id = gl.id
        and gf.work_date = p_date
        and gl.name not like '%(CC)%'
        and pl.id = gf.gl_plan_account_id
        and pl.code = '4434'))
when rc.column_id = 35 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f          gf,
             gl_account_d      gl,
             gl_plan_account_d pl
    where gf.gl_account_id = gl.id
        and gf.work_date = p_date
        and gl.name like '%(CC)%'
        and pl.id = gf.gl_plan_account_id
        and pl.code = '4434'))
when rc.column_id = 41 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f          gf,
             gl_account_d      gl,
             gl_plan_account_d pl
    where gf.gl_account_id = gl.id
        and gf.work_date = p_date
        and gl.name like '%Самрук%'
        and pl.id = gf.gl_plan_account_id
        and pl.code = '5921'))
when rc.column_id = 42 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f          gf,
             gl_account_d      gl,
             gl_plan_account_d pl
    where gf.gl_account_id = gl.id
        and gf.work_date = p_date
        and gl.name like '%Щесна%'
        and pl.id = gf.gl_plan_account_id
        and pl.code = '5921'))
when rc.column_id = 43 then
    sum((select round(sum(gf.output) / 1000, 0)
        from gl_acc_f          gf,
             gl_account_d      gl,
             gl_plan_account_d pl
    where gf.gl_account_id = gl.id
        and gf.work_date = p_date
        and gl.name like '%(CC)%'
        and pl.id = gf.gl_plan_account_id
        and pl.code = '5921'))
    end),0) val
    from set_rep_column_calculation cc, set_report_column rc
where rc.report_id = cc.report_id
    and cc.column_id = rc.column_id
    and rc.report_id = 1
    and cc.report_id = rc.report_id
group by rc.column_id,
         rc.column_num,
         rc.parent_column,
         rc.column_name
order by rc.column_id;

```

```

begin
for          rec          in          emp_cur          loop

```

```
my_row := new T_AG041_ROW(p_date,
                          rec.column_num,
                          rec.parent_column,
                          rec.column_name,
                          rec.val);

my_res.extend;
my_res(my_res.count) := my_row;
end loop;
return my_res;
end;
END REP_AG041;
```

Создание типа таблицы строк

```
CREATE OR REPLACE TYPE T_AG041_TAB as table of T_AG041_ROW
```

Создание типа строки

```
CREATE OR REPLACE TYPE T_AG041_ROW as object
(
  aDate   Varchar2(2000),
  N       Number,
  p_name  Varchar2(2000),
  name    Varchar2(2000),
  saldo   Number
)
)
```

Приложение Б

Выгружаемые отчеты из Oracle Business Intelligence

Отчет об остатках на балансовых счетах за вычетом резервов (провизий) по состоянию на 01/01/2014

(наименование банка)

(в тысячах тенге)

Наименование		Всего	из них в иностранной валюте
А		1	2
1	Активы	355 161 978	
2	Вклады в Национальном Банке Республики Казахстан	14 712 303	
3	Корреспондентские счета (с учетом начисленного вознаграждения и за вычетом резервов (провизий), в том числе:	51	
3.1	– у резидентов Республики Казахстан (за вычетом резервов (провизий))	51	
3.2	- у нерезидентов Республики Казахстан (за вычетом резервов (провизий))		
4	Вклады, размещенные в других банках (за вычетом резервов (провизий)), в том числе	10 611 552	
4.1	- у резидентов Республики Казахстан (за вычетом резервов (провизий))	10 611 552	
4.2	- у нерезидентов Республики Казахстан (за вычетом резервов (провизий))		
5	Займы, предоставленные банкам и организациям, осуществляющим отдельные виды банковских операций (за вычетом (провизий)), в том числе		
5.1	- резидентам Республики Казахстан (за вычетом резервов (провизий))		
5.2	- нерезидентам Республики Казахстан (за вычетом резервов (провизий))		
6	Ценные бумаги, учитываемые по справедливой стоимости через прибыль или убыток (за вычетом резервов (провизий))		
7	Ценные бумаги, имеющиеся в наличии для продажи (за вычетом резервов (провизий))	147 120 110	
8	Ценные бумаги, удерживаемые до погашения (за вычетом резервов (провизий))		
9	Операции "Обратное РЕПО" с ценными бумагами (за вычетом резервов (провизий))	31 605 022	
10	Займы, предоставленные юридическим лицам (за вычетом резервов (провизий))		

Продолжение приложения Б

11	Займы, предоставленные физическим лицам (за вычетом резервов (провизий)), в том числе	143 791 724	
11.1	на потребительские цели (за вычетом резервов (провизий))		
11.2	на строительство, покупку и (или) ремонт жилья (за вычетом резервов (провизий))	143 791 724	
11.3	прочие займы (за вычетом резервов (провизий))		
12	Инвестиции в капитал (за вычетом резервов (провизий))		
13	Субординированный долг		
14	Итого активы, приносящие доход	347 840 762	
15	Деньги	722 155	
16	Аффинированные драгоценные металлы		
17	Обязательные резервы в Национальном Банке Республики Казахстан		
18	Основные средства (за вычетом амортизации)	2 958 421	
19	Нематериальные активы (за вычетом амортизации)	450 577	
20	Прочие активы, в том числе	3 190 063	
20.1	- требования по производным финансовым инструментам		
20.2	- требования по операциям спот		
20.3	- отсроченный подоходный налог	761 161	
20.4	- начисленные проценты к получению	2 224 726	
20.5	- предоплата (расходы)	23 198	
21	Справочно: резервы (провизии)	3 742 443	
22	Справочно: амортизация	1 125 372	
23	Обязательства	256 908 389	
24	Обязательства перед Национальным Банком Республики Казахстан		
25	Обязательства перед банками и организациями, осуществляющими отдельные виды банковских операций, в том числе		
25.1	- корреспонденские счета		
25.1.1	резидентов Республики Казахстан		
25.1.2	нерезидентов Республики Казахстан		
25.2	- займы, полученные		
25.2.1	от резидентов Республики Казахстан		
25.2.2	от нерезидентов Республики Казахстан		
25.3	- вклады, привлеченные		
25.3.1	от резидентов Республики Казахстан		
25.3.1.1	до востребования		
25.3.1.2	срочные		
25.3.1.3	условные		
25.3.2	от нерезидентов Республики Казахстан		
25.3.2.1	до востребования		
25.3.2.2	срочные		

Продолжение приложения Б

25.3.2.3	условные		
26	Займы, полученные от иностранных центральных банков		
27	Займы, полученные от международных финансовых организаций		
28	Вклады, привлеченные от физических лиц, в том числе	172 049 137	
28.1	- текущие и карт-счета	2 076 188	
28.1.1	от резидентов Республики Казахстан	2 076 188	
28.1.2	от нерезидентов Республики Казахстан		
28.2	- вклады до востребования		
28.2.1	от резидентов Республики Казахстан		
28.2.2	от нерезидентов Республики Казахстан		
28.3	- условные вклады		
28.3.1	от резидентов Республики Казахстан		
28.3.2	от нерезидентов Республики Казахстан		
28.4	- срочные вклады	169 972 949	
28.4.1	от резидентов Республики Казахстан	169 967 953	
28.4.2	от нерезидентов Республики Казахстан	4 996	
29	Вклады, привлеченные от юридических лиц, в том числе		
29.1	- текущие и карт-счета		
29.1.1	от резидентов Республики Казахстан		
29.1.2	от нерезидентов Республики Казахстан		
29.2	- вклады до востребования		
29.2.1	от резидентов Республики Казахстан		
29.2.2	от нерезидентов Республики Казахстан		
29.3	- условные вклады		
29.3.1	от резидентов Республики Казахстан		
29.3.2	от нерезидентов Республики Казахстан		
29.4	- срочные вклады		
29.4.1	от резидентов Республики Казахстан		
29.4.2	от нерезидентов Республики Казахстан		
29.5	- вклады дочерних организаций специального назначения		
29.5.1	от резидентов Республики Казахстан		
29.5.2	от нерезидентов Республики Казахстан		
30	Деньги, принятые в качестве обеспечения (заклад, задаток) обязательств клиентов	23 500 494	
31	Займы, полученные от Правительства Республики Казахстан и местных органов власти Республики Казахстан	58 771 221	
32	Ценные бумаги, выпущенные в обращение		
33	Операции "РЕПО" с ценными бумагами		
33.1	с резидентами Республики Казахстан		
33.2	с нерезидентами Республики Казахстан		

Продолжение приложения Б

34	Субординированный долг		
34.1	от резидентов Республики Казахстан		
34.2	от нерезидентов Республики Казахстан		
35	Итого обязательства, влекущие расход	254 320 852	
36	Резервы (провизии) на покрытие убытков по условным обязательствам		
37	Прочие обязательства, в том числе	2 587 537	
37.1	обязательства по производным финансовым инструментам		
37.2	обязательства по операциям спот		
37.3	обязательства по подоходному налогу		
37.4	начисленные проценты к оплате	108 347	
37.5	предоплата (доходы)		
38	Справочно: обязательства банка перед родительской организацией по полученным займам		
39	Собственный капитал	98 253 589	
40	Уставный капитал	78 300 000	
41	Дополнительный капитал	12 704 693	
42	Нераспределенный чистый доход и резерв капитала:	7 449 094	
42.1	. доход (непокрытый убыток) предыдущих лет	614 198	
42.2	. доход (непокрытый убыток) текущего года	4 551 561	
42.3	. резерв капитала	2 283 335	
43	Резервы переоценки основных средств		
44	Резервы переоценки стоимости ценных бумаг, имеющих в наличии для продажи	-1 306 467	
45	Резервы (провизии) на общепанковские риски		
46	Динамические резервы	1 106 269	
47	Резервы по прочей переоценке		
48	Выкупленные банком собственные акции		
49	Итого обязательства и капитал	355 161 978	
50	Справочно: активы, номинированные в тенге, индексированные к иностранной валюте		
51	Справочно: обязательства, номинированные в тенге, индексированные к иностранной валюте		
52	Справочно: объявленные дивиденды по простым акциям за финансовый год		
53	Справочно: количество работников, занятых полный рабочий день	895	

Продолжение приложения Б

Основные показатели баланса

(в тысячах тенге)

№ п/п	Показатели баланса	Дата, по состоянию на	Дата, по состоянию на
		01.01.2014	01.02.2014
1	2	3	4
		5	
	ИТОГО, в том числе:	355 161 978	369 795 393
1	Корр.счет в НБРК	5 712 303	2 365 053
2	Срочные вклады в Национальном Банке РК	9 000 000	9 000 000
3	Требования к клиентам, в том числе:	143 308 333	144 820 803
4	1) Резервы (провизии) по займам и финансовому лизингу, предоставленным клиентам	-2 142 333	-2 117 220
5	2) Дисконт по займам, предоставленным клиентам	-3 596 648**	-3 558 666**
6	Ценные бумаги по рыночной стоимости, в том числе:	147 082 348	144 365 420
7	1) Резервы (провизии) на покрытие убытков по ценным бумагам	-1 592 460	-1 592 460
8	Межбанковские депозиты, в том числе:	10 611 552	13 116 506
9	1) Дисконт по вкладам, размещенным в других банках	-203 876	-198 922
10	НДС в зачет	0	6 519
11	Ценные бумаги удерживаемые до погашения	0	0
	ИТОГО, в том числе:	256 908 389	270 800 473
12	Долгосрочные займы, полученные от Правительства РК и местных исполнительных органов РК и национального управляющего холдинга	69 200 000	69 200 000
13	Дисконт по полученным займам от АО "ФНБ Самрук-Казына" и Правительства РК	-10 428 779	-10 325 360
14	Обязательства перед клиентами, в том числе:	195 549 631	208 748 115
15	1) Премия по вкладам, привлеченным от клиентов	939 253	959 332
16	Начисленный НДС	4 326	4 845

Продолжение приложения Б

	Раздел III Собственный капитал	ИТОГО, в том числе:	98 253 589	98 994 920	
17		Уставный капитал - простые акции	78 300 000	78 300 000	
18		Дополнительный капитал	12 704 693	12 704 693	
19		Динамические резервы	1 106 269	1 106 269	
20		Резервный капитал	2 283 335	2 283 335	
21		Резервы переоценки стоимости финансовых активов, имеющих в наличии для продажи	-1 306 467	-1 316 500	
22		Нераспределенная прибыль (непокрытый убыток) прошлых лет	614 198	5 165 759	
23		Нераспределенная прибыль	4 551 561	751 364	
		IV Раздел Доходы	ИТОГО, в том числе:	18 858 708	1 905 001
24	Доходы от ЦБ		7 552 358	754 292	
25	Доходы по амортизации дисконта по вкладу, размещенному АО "Банк Центр Кредит"		20 572	1 775	
26	Доходы по амортизации дисконта по вкладу, размещенному АО "БТА Банк"		0	0	
27	Доходы по амортизации дисконта по вкладу, размещенному АО "Цеснабанк"		20 988	3 179	
28	Доходы от займов, в том числе:		9 205 415	929 060	
29	1) Доходы по амортизации дисконта по займам, предоставленным клиентам		403 685	40 604	
30	2) Доходы по амортизации дисконта по займам ГП 2008-2010 г.г., предоставленным клиентам за счет займа АО "ФНБ Самрук-Казына" и Правительства РК, в том числе по эффективной ставке (СС)		798 685	65 152	
31	Комиссионные доходы		622 979	27 495	
32	Доходы от восстановления резервов (провизий)		113 179	38 040	
	V Раздел Расходы		ИТОГО, в том числе:	12 861 612	1 153 637
33			Расходы по амортизации дисконта по полученным займам от АО "ФНБ "Самрук-Казына"	1 237 819	103 419
34			Ассигнования на обеспечение	726 896	14 088
35			Самортизированный расход по займам, выданным за счет займа от АО "ФНБ Самрук-Казына"	0	0
36		Самортизированный расход по вкладу, размещенному в АО "Цеснабанк"	112 222	0	

Продолжение приложения Б

37		Самортизированный расход по выданным займам по Госпрограмме за счет бюджетных средств из уставного капитала, в том числе по эффективной ставке (СС)	381 368	40 606
38	VII Раздел Счета меморандума	ИТОГО, в том числе:	321 491 035	325 599 677

Продолжение приложения Б

Отчет банков о финансовых потоках и запасах

за _____ ГОД

Раздел I. ФИНАНСОВЫЕ АКТИВЫ

П/П	Наименование показателей	Всего, тыс. тенге	В том числе:										Расшифровка вида других изменений стоимости
			по секторам экономики:								В целом по экономике	Остальной мир (нерезиденты)	
			Финансовый сектор	Из него:			Сектор домашних хозяйств	НКУ, обслуживающие домашние хозяйства (НКУОДХ)	Сектор органов государственного управления	Нефинансовый сектор			
Национальный Банк	Банки	Другие финансовые организации (небанковские)		6	7	8					9	10 (2+6+7+8+9)	11
А	В	1 (10+11)	2 (3+4+5)	3	4	5	6	7	8	9	10 (2+6+7+8+9)	11	12
	Наличная валюта												
	<i>1. Национальная</i>												
1	объем на начало отчетного периода	251 577	251 577	251 577							251 577		
2	изменения в результате проведения операций	190 212	190 212	190 212							190 212		
3	другие изменения (укажите вид)												
4	объем на конец отчетного периода	441 789	441 789	441 789							441 789		
	<i>2. Иностранная</i>												
5	объем на начало отчетного периода	21 331										21 331	
6	изменения в результате проведения операций	-12 560										-12 560	
7	оценочные изменения стоимости	501										501	
8	другие изменения (укажите вид)												
9	объем на конец отчетного периода	9 272										9 272	
	Переводимые депозиты												
	<i>1. В национальной валюте</i>												
10	объем на начало отчетного периода	2 062 331	2 062 331	2 057 928	4 403						2 062 331		
11	изменения в результате проведения операций	5 852 742	5 852 742	5 857 130	-4 388						5 852 742		
12	оценочные изменения стоимости												
13	другие изменения (укажите вид)												

Продолжение приложения Б

	вид)												
14	объем на конец отчетного периода	7 915 073	7 915 073	7 915 058	15							7 915 073	
15	доходы к получению												
	<i>2. В иностранной валюте</i>												
16	объем на начало отчетного периода												
17	изменения в результате проведения операций												
18	оценочные изменения стоимости												
19	другие изменения (укажите вид)												
20	объем на конец отчетного периода												
21	доходы к получению												
	Другие депозиты												
	<i>1. В национальной валюте</i>												
22	объем на начало отчетного периода	10 441 146	10 441 146	8 400 361	2 040 785							10 441 146	
23	изменения в результате проведения операций	10 088 472	10 088 472	8 606 472	1 482 000							10 088 472	
24	оценочные изменения стоимости												
25	другие изменения (укажите вид)												
26	объем на конец отчетного периода	20 529 618	20 529 618	17 006 833	3 522 785							20 529 618	
27	доходы к получению	43 607	43 607	6 833	36 774							43 607	
	<i>2. В иностранной валюте</i>												
28	объем на начало отчетного периода												
29	изменения в результате проведения операций												
30	оценочные изменения стоимости												
31	другие изменения (укажите вид)												
32	объем на конец отчетного периода												
33	доходы к получению												
	Ценные бумаги (за исключением акций)												
	<i>1. В национальной валюте</i>												
34	объем на начало отчетного периода	24 537 745	2 935 572		2 537 147	398 425			19 911 630	1 690 543		24 537 745	
35	нетто покупка (включая новые выпуски) за отчетный период	70 556 980	16 669 817	16 555 736	193 355	-79 274			52 118 654	1 768 509		70 556 980	
36	разница в оценке стоимости												
37	другие изменения (укажите вид)												
38	объем на конец отчетного периода	95 094 725	19 605 389	16 555 736	2 730 502	319 151			72 030 284	3 459 052		95 094 725	

Продолжение приложения Б

39	проценты (дивиденды) к получению за отчетный период	977 433	45 404		41 354	4 050			881 977	50 052	977 433		
	<i>2. В иностранной валюте</i>												
40	объем на начало отчетного периода												
41	нетто покупка (включая новые выпуски) за отчетный период												
42	разница в оценке стоимости												
43	другие изменения (укажите вид)												
44	объем на конец отчетного периода												
45	проценты (дивиденды) к получению за отчетный период												
	Кредиты и займы												
	1. Краткосрочные												
	<i>1.1 В национальной валюте</i>												
46	объем на начало отчетного периода	5 534	0				5 534				5 534		
47	изменения в результате проведения операций	4 303 066	4 305 375			4 305 375	-2 309				4 303 066		
48	оценочные изменения стоимости												
49	другие изменения (укажите вид)												
50	объем на конец отчетного периода	4 308 600	4 305 375			4 305 375	3 225				4 308 600		
51	проценты к получению за отчетный период (по графику)	376	371			371	5				376		
52	проценты фактически полученные	42 481	42 402			42 402	79				42 481		
	<i>1.2 В иностранной валюте</i>												
53	объем на начало отчетного периода												
54	изменения в результате проведения операций												
55	оценочные изменения стоимости												
56	другие изменения (укажите вид)												
57	объем на конец отчетного периода												
58	проценты к получению за отчетный период (по графику)												
59	проценты фактически полученные												
	2. Долгосрочные												
	<i>2.1 В национальной валюте</i>												
60	объем на начало отчетного периода	67 936 547					67 936 547				67 936 547		
61	изменения в результате проведения операций	18 846 552					18 846 552				18 846 552		
62	оценочные изменения стоимости												

Продолжение приложения Б

63	другие изменения (укажите вид)	-145 859											Другие изменения в объеме финансовых активов, не отнесенные к другим категориям - списание кредиторам и безнадежных долгов.
64	объем на конец отчетного периода	86 637 240					86 637 240					86 637 240	
65	проценты к получению за отчетный период (по графику)	261 123					261 123					261 123	
66	проценты фактически полученные	3 949 691					3 949 691					3 949 691	
	<i>2.2 В иностранной валюте</i>												
67	объем на начало отчетного периода												
68	изменения в результате проведения операций												
69	оценочные изменения стоимости												
70	другие изменения (укажите вид)												
71	объем на конец отчетного периода												
72	проценты к получению за отчетный период (по графику)												
73	проценты фактически полученные												
	Производные финансовые инструменты												
	<i>1. В национальной валюте</i>												
74	объем на начало отчетного периода												
75	изменения в результате проведения операций												
76	оценочные изменения стоимости												
77	другие изменения (укажите вид)												
78	объем на конец отчетного периода												
	<i>2. В иностранной валюте</i>												
79	объем на начало отчетного периода												
80	изменения в результате проведения операций												
81	оценочные изменения стоимости												
82	другие изменения (укажите вид)												

Продолжение приложения Б

	вид)												
83	объем на конец отчетного периода												
	Акции и другие формы участия в капитале												
	<i>1. В национальной валюте</i>												
84	объем на начало отчетного периода												
85	изменения в результате проведения операций												
86	оценочные изменения стоимости												
87	другие изменения (укажите вид)												
88	объем на конец отчетного периода												
89	доходы к получению												
	<i>2. В иностранной валюте</i>												
90	объем на начало отчетного периода												
91	изменения в результате проведения операций												
92	оценочные изменения стоимости												
93	другие изменения (укажите вид)												
94	объем на конец отчетного периода												
95	доходы к получению												
	Другая дебиторская задолженность (требования)												
	<i>1. В национальной валюте</i>												
96	объем на начало отчетного периода	-1 836 444	17 198		13 952	3 246	-1 523 401	216	-35 020	-302 456	-1 843 463	7 019	
97	изменения в результате проведения операций	-1 051 407	-189 663	-118 504	-74 663	3 504	-1 465 814	45	563 511	46 456	-1 045 465	-5 942	
98	оценочные изменения стоимости												
99	другие изменения (укажите вид)												
100	объем на конец отчетного периода	-2 887 851	-172 465	-118 504	-60 711	6 750	-2 989 215	261	528 491	-256 000	-2 888 928	1 077	
101	проценты к получению за отчетный период												
	<i>2. В иностранной валюте</i>												
102	объем на начало отчетного периода												
103	изменения в результате проведения операций												
104	оценочные изменения стоимости												
105	другие изменения (укажите вид)												
106	объем на конец отчетного периода												

Продолжение приложения Б

107	проценты к получению за отчетный период												
	<i>Справочно (без разбивки по секторам):</i>												
108	специальные резервы (проvisions) на покрытие убытков	-1 788 266	x	x	x	x	x	x	x	x	x	x	x
109	счета положительной корректировки стоимости финансовых инструментов	1 378 264	x	x	x	x	x	x	x	x	x	x	x
110	счета отрицательной корректировки стоимости финансовых инструментов	-594 024	x	x	x	x	x	x	x	x	x	x	x

Раздел II. ФИНАНСОВЫЕ ПАССИВЫ

П/П	Наименование показателей	Всего, тыс. тенге	В том числе:										Расшифровка вида других изменений стоимости
			по секторам экономики:										
			Финансовый сектор	Из него:			Сектор домашних хозяйств	НКУ, обслуживающие домашние хозяйства (НКУОДХ)	Сектор органов государственного управления	Нефинансовый сектор	В целом по экономике	Остальной мир (нерезиденты)	
Национальный Банк	Банки второго уровня	Другие финансовые организации (небанковские)											
	А	1 (10+11)	2 (3+4+5)	3	4	5	6	7	8	9	10 (2+6+7+8+9)	11	
	Переводимые депозиты												
	<i>1. В национальной валюте</i>												
111	объем на начало отчетного периода	376 895					376 895				376 895		
112	изменения в результате проведения операций	265 696					265 696				265 696		
113	оценочные изменения стоимости												
114	другие изменения (укажите вид)												
115	объем на конец отчетного периода	642 591					642 591				642 591		
116	расходы к оплате												
	<i>2. В иностранной валюте</i>												
117	объем на начало отчетного периода												
118	изменения в результате проведения операций												
119	оценочные изменения стоимости												
120	другие изменения (укажите вид)												
121	объем на конец отчетного периода												

Продолжение приложения Б

122	расходы к оплате												
	Другие депозиты												
	<i>1. В национальной валюте</i>												
123	объем на начало отчетного периода	39 673 539				39 670 234				39 670 234	3 305		
124	изменения в результате проведения операций	32 888 838				32 891 413				32 891 413	-2 575		
125	оценочные изменения стоимости												
126	другие изменения (укажите вид)												
127	объем на конец отчетного периода	72 562 377				72 561 647				72 561 647	730		
128	расходы к оплате												
	<i>2. В иностранной валюте</i>												
129	объем на начало отчетного периода												
130	изменения в результате проведения операций												
131	оценочные изменения стоимости												
132	другие изменения (укажите вид)												
133	объем на конец отчетного периода												
134	расходы к оплате												
	Ценные бумаги (за исключением акций)												
	<i>1. В национальной валюте</i>												
135	объем на начало отчетного периода												
136	покупка (включая новые выпуски) за отчетный период												
137	продажа (включая выкуп) за отчетный период												
138	разница в оценке стоимости												
139	другие изменения (укажите вид)												
140	объем на конец отчетного периода												
141	проценты (дивиденды) к уплате за отчетный период												
	<i>2. В иностранной валюте</i>												
142	объем на начало отчетного периода												
143	покупка (включая новые выпуски) за отчетный период												
144	продажа (включая выкуп) за отчетный период												
145	разница в оценке стоимости												
146	другие изменения (укажите вид)												
147	объем на конец отчетного периода												
148	проценты (дивиденды) к уплате за отчетный период												

Продолжение приложения Б

	Кредиты и займы												
	1. Краткосрочные												
	<i>1.1 В национальной валюте</i>												
149	объем на начало отчетного периода	1 500 015	1 500 015			1 500 015					1 500 015		
150	изменения в результате проведения операций	-1 500 015	-1 500 015			-1 500 015					-1 500 015		
151	оценочные изменения стоимости												
152	другие изменения (укажите вид)												
153	объем на конец отчетного периода	0	0			0					0		
154	проценты к уплате за отчетный период (по графику)												
155	проценты фактически уплаченные	2 016	2 016			2 016					2 016		
	<i>1.2 В иностранной валюте</i>												
156	объем на начало отчетного периода												
157	изменения в результате проведения операций												
158	оценочные изменения стоимости												
159	другие изменения (укажите вид)												
160	объем на конец отчетного периода												
161	проценты к уплате за отчетный период (по графику)												
162	проценты фактически уплаченные												
	2. Долгосрочные												
	<i>2.1 В национальной валюте</i>												
163	объем на начало отчетного периода	38 049 911						23 638 533	14 411 378		38 049 911		
164	изменения в результате проведения операций	19 054 049							19 054 049		19 054 049		
165	оценочные изменения стоимости												
166	другие изменения												
167	объем на конец отчетного периода	57 103 960						23 638 533	33 465 427		57 103 960		
168	проценты к уплате за отчетный период (по графику)	103 960						78 533	25 427		103 960		
169	проценты фактически уплаченные	340 764						157 067	183 697		340 764		
	<i>2.2 В иностранной валюте</i>												
170	объем на начало отчетного периода												
171	изменения в результате проведения операций												
172	оценочные изменения стоимости												
173	другие изменения (укажите												

Продолжение приложения Б

	вид)													
174	объем на конец отчетного периода													
175	проценты к уплате за отчетный период (по графику)													
176	проценты фактически уплаченные													
	Производные финансовые инструменты													
	<i>2.1 В национальной валюте</i>													
177	объем на начало отчетного периода													
178	изменения в результате проведения операций													
179	оценочные изменения стоимости													
180	другие изменения													
181	объем на конец отчетного периода													
	<i>2.2 В иностранной валюте</i>													
182	объем на начало отчетного периода													
183	изменения в результате проведения операций													
184	оценочные изменения стоимости													
185	другие изменения													
186	объем на конец отчетного периода													
	Акции и другие формы участия в капитале													
	<i>1. В национальной валюте</i>													
187	объем на начало отчетного периода	25 049 815	25 049 815		25 049 815							25 049 815		
188	изменения в результате проведения операций	64 209 447	64 209 447		64 209 447							64 209 447		
189	оценочные изменения стоимости													
190	другие изменения (укажите вид)													
191	объем на конец отчетного периода	89 259 262	89 259 262		89 259 262							89 259 262		
192	расходы к оплате													
	<i>2. В иностранной валюте</i>													
193	объем на начало отчетного периода													
194	изменения в результате проведения операций													
195	оценочные изменения стоимости													
196	другие изменения (укажите вид)													
197	объем на конец отчетного периода													

Продолжение приложения Б

198	расходы к оплате												
	Другая кредиторская задолженность												
	<i>1. В национальной валюте</i>												
199	объем на начало отчетного периода	-3 461 892	58 162	234	159	57 769	901 667	35	10 433	-4 434 304	-3 464 007	2 115	
200	изменения в результате проведения операций	-3 303 236	14 126	168	38	13 920	614 763	-35	24 721	-3 955 935	-3 302 360	-876	
201	оценочные изменения стоимости												
202	другие изменения (укажите вид)												
203	объем на конец отчетного периода	-6 765 128	72 288	402	197	71 689	1 516 430	0	35 154	-8 390 239	-6 766 367	1 239	
204	проценты к уплате за отчетный период												
	<i>2. В иностранной валюте</i>												
205	объем на начало отчетного периода												
206	изменения в результате проведения операций												
207	оценочные изменения стоимости												
208	другие изменения (укажите вид)												
209	объем на конец отчетного периода												
210	проценты к уплате за отчетный период												