

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Автоматизированная информационная система анализа финансового состояния страховой компании

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Бакарова А.П. ВТ-10-03
(Фамилия и инициалы) группа

Научный руководитель Сатимова С.Т. доцент, к.т.н. [подпись]
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Ерсеиева З.Д. асс. профессор.
(Фамилия и инициалы, ученая степень, звание)
Ерсеиева « 14 » 05 2014 г.
(подпись)

по безопасности жизнедеятельности:

Приходько И.Г. д.х.н. профессор
(Фамилия и инициалы, ученая степень, звание)
[подпись] « 14 » 05 2014 г.
(подпись)

по применению вычислительной техники:

Сатимова С.Т. доцент, к.т.н. [подпись]
(Фамилия и инициалы, ученая степень, звание)
[подпись] « 29 » 05 2014 г.
(подпись)

Нормоконтролер: Тусупов Д.М.
(Фамилия и инициалы, ученая степень, звание)

[подпись] « 27 » мая 2014 г.
(подпись)

Рецензент: _____
(Фамилия и инициалы, ученая степень, звание)
« _____ » _____ 20__ г.
(подпись)

Алматы 2014 г.

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационные Технологии
Специальность Вычислительная техника и программное обеспечение
Кафедра компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Баканова Анна Павловна
(фамилия, имя, отчество)

Тема проекта Автоматизированная информационная система анализа финансового состояния страховой компании

утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «__» _____ 20__ г.

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

Разработать автоматизированную информационную систему для работы с управленческой ответственностью страховой компании. Система должна быть понятной обычному пользователю и удобной в обращении.

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

Требуется разработка автоматизированной информационной системы, предназначенной для сбора, хранения, анализа и предоставления оперативной финансовой информации руководству, необходимой для дальнейшего планирования, измерения объема, суммарного до- сегментов продаж и достижения более высоких результа- тов бизнеса. Система должна работать с управленчес- кой ответственностью страховой компании и проводить ее ана- лиз, что подразумевает собой расчет экономических показателей, параметров и коэффициентов, применяе- мых при дальнейшем стратегическом планировании компании.

Перечень графического материала (с точным указанием обязательных чертежей)

- 1) Схема работы АИС
- 2) Схема функционирования системы
- 3) Структура информационного хранилища
- 4) Схема центрального хранилища данных и многих ветвей
- 5) Диаграмма компонентов, диаграмма прецедентов
- 6) Диаграмма классов, диаграмма развертывания
- 7) Диаграмма ER-диаграмма
- 8) Сегменты ER-диаграммы для бизнес-правил
- 9) Диаграмма физической модели хранилища данных
- 10) Структура связности программного продукта
- 11) Изображение работы программы.

Рекомендуемая основная литература

- 1) Сатишова, Е.Г. Проектирование баз данных. Методические указания к выполнению лабораторных работ. - Алматы: АИИЭ, 2009.
- 2) Информационно-аналитические системы поддержки принятия решений в страховании / Дмитрий Евровский - Ри-насовые газета ЭКСПО, 2006
- 3) Э. Сирли. Корпоративные хранилища - М.: Издательский дом "Вильямс", 2001
- 4) А.И. Кудинев. Хранилища данных. Цикл лабораторных работ. Часть 1, 2 - Томск: ТПУ - 2008
- 5) П. Род, К. Коржел. Системы баз данных: проектирование, реализация, управление - 5-е издание - СПб, 2004.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Жеңелме	Бресеева З. Д.	18.04 - 14.05.14	Бресеева
Нормоконтроль	Тусупов Д.М.	27.05.2014	Тусупов
Техзадание Н.Г.	БНД	11.04 - 14.05.14	БНД
Сатишова Е.Г.	Дипломный руководитель	21.12.13 - 29.05.14	Сатишова

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1.	Оформление целей и задач АИС	16.01.14	
2.	Определение специфики работы академической системы	07.02.14	
3.	Определение специфики работы страховой компании, анализ управленческой отчетности	12.02.14	В течение прохождения производственной практики в страховой компании
4.	Начальная разработка ходовых графиков, определение основных возможностей, построение диаграмм	18.02.14	"London-Almaty"
5.	Проектирование ХД, определение бизнес правил, разработка и построение ER-диаграммы	21.02.14	
6.	Архитектурное и функциональное проектирование	15.03.14	
7.	Определение языка программирования и среды разработки языка базисного интерпретатора	22.03.14	
8.	Оформление технического экономического обоснования проекта	09.04.14	
9.	Оформление технического решения безопасности труда	13.04.14	
10.	Исправление ошибок в разработке системы	15.04.14	
11.	Оформление отчета	05.05.14	
12.	Подготовка к защите, создание презентации	17.05.14	

Дата выдачи задания « ___ » _____ 20__ г.

Заведующий кафедрой _____
(подпись) (Фамилия и инициалы)

Руководитель _____
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент _____
(подпись) (Фамилия и инициалы)

Аннотация

В данном дипломном проекте проводится разработка автоматизированной информационной системы анализа финансового состояния страховой компании.

Разработана система, для автоматизации работы финансового аналитика страховой компании с управленческой отчетностью. Она позволяет проводить анализ финансового состояния страховой компании на основе управленческой отчетности. Система создана с использованием новейших технологий и специфики сферы страхования и отвечает всем современным требованиям к программному обеспечению.

Кроме этого в двух последних главах рассматриваются вопросы безопасности жизнедеятельности, приводится технико-экономическое обоснование и рассчитывается цена разработки проекта.

Андатпа

Бұл диплом жобасында сақтандыру компаниясының қаржы жағдайын талдаудың автоматтандырылған ақпараттық жүйелері әзірленді.

Сақтандыру компаниясының қаржы талдамасын автоматтандыру үшін басқарушылық есептемесі бар жүйе жасалды. Бұл жүйе басқарушылық есептеме негізінде сақтандыру компаниясының қаржы жағдайына талдау жүргізуге мүмкіндік береді. Жүйе жаңа технологияларды және сақтандыру саласының ерекшеліктерін қолдана отырып жасалды және бағдарламалық камтамасыз етудің барлық қазіргі талаптарына жауап береді.

Сонымен қатар, жұмыстың соңғы екі тарауында өміртіршілік қауіпсіздігі мәселелері қарастырылды, техника-экономикалық дәйектемелері келтірілді және жобаны әзірлеу құны есептелді.

Abstract

In this thesis-project is to develop an automated information system of the analysis of a financial condition of the insurance company.

Developed system for automation of financial analyst of insurance company management reporting. It allows the analysis of a financial condition of the insurance company based on management accounts. The system is built using latest technology and specifics of the insurance industry and meets all modern requirements to the software.

In addition, in the last two chapters deals with the issues of safety, provides a feasibility study and calculation of the cost of project development.

Содержание

Введение.....	Ошибка! Закладка не определена.
1 Требования к разрабатываемой информационной системе	16
1.1 Назначение и цели создания системы.....	16
1.2 Разработка аналитической системы.....	17
1.3 Требования к разрабатываемой информационной системе	21
1.4 Специфика разработки данного проекта	30
2 Разработка аналитической системы	34
2.1 Постановка задачи	34
2.2 Этап начальной разработки ХД.....	34
2.2.1 Анализ предметной области	34
2.2.2 Базовые сущности хранилища данных	35
2.2.3 UML диаграммы.....	36
2.2.4 Диаграмма компонентов	36
2.2.5 Диаграмма прецедентов	37
2.2.6 Диаграмма классов.....	37
2.2.7 Диаграмма развертывания	38
2.3 Проектирование ХД.....	39
2.3.1 Концептуальное проектирование.....	39
2.3.2 Бизнес правила	40
2.3.3 Разработка и построение подробной ER-диаграммы на основании бизнес правил.....	43
2.3.4 Анализ информационных задач и круга пользователей системы.....	44
2.3.5 Определение атрибутов каждой сущности	44
2.4 Составление реляционных отношений.....	45
2.5 Расчет места для хранения ХД	46
2.6 Используемые языки программирования.....	49
2.6.1 Среда разработки Visual studio	49
2.6.2 Язык С#	49
2.7 Расчётная часть.....	50
2.7.1 Логическое проектирование	50
2.7.2 Физическое проектирование.....	51
2.8 Описание интерфейса приложения и его функциональности.....	64
3 Техничко-экономическое обоснование проекта.....	71
3.1 Описание работы и обоснование необходимости	71
3.2 Трудовые ресурсы, используемые в работе	71
3.3 Расчет затрат на амортизацию	78
3.4 Цена программного продукта	81
4 Технические решения безопасности труда	83
4.1 Анализ потенциально опасных и вредных производственных факторов проектируемого объекта, воздействующих на персонал.....	83
4.2 Анализ помещения.....	83
4.3 Характеристика оборудования	85

4.4 Мероприятия, обеспечивающие безопасность при работе на компьютере	87
4.5 Расчет искусственного освещения помещения.....	88
4.6 Расчет системы кондиционирования офиса.....	90
Заключение	94
Список литературы	941
Приложение А	952

Введение

Современный этап развития рыночных отношений в российской экономике (первое десятилетие XXI века) характеризуется началом экономического подъема. Период времени быстрых, в значительной мере интуитивных, импровизационных, а зачастую и силовых решений меняется на зону продуманных, просчитанных выводов и решений – оперативных, инвестиционных. Большое значение имеет субъективный фактор, под которым будем понимать влияние на ход бизнес-процессов работников предприятия и в особенности лиц, принимающих решения.

Для выработки и принятия соответствующих складывающейся обстановке решений необходимы информация и знания, которые должны удовлетворять требованиям полноты, достоверности, своевременности (актуальности), полезности. Основополагающую роль в подготовке принятия решений играет его обоснование по имеющейся информации. Ее, как правило, получают из различных внутренних и внешних источников. В интересах выработки адекватного решения используются внутренние информационные ресурсы, которые складываются из отражения деятельности (функционирования) объекта в документах, других видах и способах сбора, обработки, хранения информации.

Таким образом, границы информационного пространства как отображения деятельности предприятия и его взаимодействия с внешней средой, в рамках которого принимаются решения, выходят далеко за пределы предприятия. Объемы информации, необходимой и используемой при принятии решений, достигают десятков и сотен мегабайт, а в крупных корпоративных и общегосударственных системах и терабайт (10¹² байт). Информация характеризуется многоплановостью, сложностью отображаемых объектов и систем, а также связей между объектами, явлениями и процессами, скрытостью закономерностей.

Эти обстоятельства вынуждают использовать имеющиеся в настоящее время весьма развитые программно-технические средства. Широкое и эффективное применение этих средств стало одним из факторов выживаемости и успеха предприятия в условиях острой конкурентной борьбы. Получили широкое распространение автоматизированные информационные системы, которые в последние годы чаще называют информационные системы, подразумевая, что без автоматизации их просто невозможно представить.

Проблема анализа исходной информации для принятия решений оказалась настолько серьезной, что появилось отдельное направление или вид информационных систем – информационно-аналитические системы (ИАС), под которыми понимают комплекс аппаратных, программных средств, информационных ресурсов, методик, которые используются для обеспечения автоматизации аналитических работ в целях обоснования принятия управленческих решений и других возможных применений.

Основной целью дипломного проекта является разработка автоматизированной информационной системы, предназначенной для сбора, хранения, анализа и предоставления оперативной финансовой информации руководству, необходимой для дальнейшего планирования, снижения объёма бумажного документооборота и др.

Для создания автоматизированной информационной системы были выбраны Microsoft Visual Studio 2012 и Oracle SQL Developer. Oracle SQL Developer – интегрированная среда разработки на языках SQL и PL/SQL, ориентированная на применение в среде Oracle Database. Microsoft Visual Studio – линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

Основной задачей данной АИС является то, что она должна работать с управленческой отчётностью предприятия и анализировать, что позволит применять её в дальнейшем стратегическом планировании страховой компании. Благодаря её использованию, существенно сокращается время, затрачиваемое на обработку отчётной информации для предоставления руководителю. Это достигается путем произведения больших производственных расчётов и хранения данных в зависимости от типа.

Требования к разрабатываемой информационной системе

1.1 Назначение и цели создания системы

Важнейшая задача руководителей информационных служб и высшего руководства в сфере информатизации заключается в том, чтобы взаимно увязать деятельность подчиненных им ИТ-подразделений с задачами бизнес подразделений организации. Однако большинство руководителей, будучи техническими специалистами, не владеют в полной мере искусством формирования и воплощения в жизнь организационной стратегии. Кроме того, сама организация унаследованных систем не позволяет легко интегрировать данные из разных систем для предоставления потребителям новой информации. Поэтому, увязать информационные технологии и бизнес для предоставления бизнес-подразделениям новой информации за счет внесения изменений в оперативные системы – сложная задача. ИТ-специалист может обозначить некоторую ситуацию, как очевидно требующую изменений, но он не знает, как выбрать стратегию или реализовать изменения, связанные с существующей технологией.

В настоящее время наблюдается повсеместное внедрение АИС в сферу управления предприятием. Это вполне понятно, так как автоматизация управления приносит наиболее ощутимый эффект. В сфере автоматизации крупных предприятий эти системы получили название корпоративных информационных систем. К предприятиям такого типа можно отнести, например, промышленные предприятия с разветвленной структурой производства, торгово-закупочные объединения, предприятия энергоснабжения и др.

Современными тенденциями страхового рынка является консолидация, слияние, укрупнение компаний, и в тоже время есть законодательное разделение страховых операций между отдельными юридическими лицами, например законодательное требование о разделении бизнеса страхования жизни и рисков видов страхования. Информационные технологии позволяют создать единое информационное пространство и исключить влияние организационных барьеров на эффективность работы компании или холдинга.

Таким образом, для решения задач развития компании, увеличения доли рынка, разработки новых страховых продуктов, повышения объема оборотных средств и прибыли от их инвестирования страховой компании необходимо анализировать огромные массивы информации, создавать модели и прогнозы страхового и финансового рынка, что в современной ситуации без применения ИТ совершенно невозможно.

У современной крупной страховой компании, предлагающей широкий ассортимент страховых продуктов и юридическим, и физическим лицам, в работе – миллионы договоров, данные которых необходимо, во-первых, хранить и обрабатывать, вести учет денег, документов, операций и т.д.

Перед страховыми компаниями встает задача максимально эффективного распределения имеющихся массивов, в том числе посредством облачных технологий в случае, если это позволяют требования безопасности и конфиденциальности информации. Обеспечение повсеместной возможности мгновенного доступа к данным с любого типа устройств значительно повысит эффективность работы страховщиков. Автоматизация задач учета в большинстве страховых компаний уже тем или иным образом реализована (или начаты проекты внедрения учетной системы). Второй круг задач связан с анализом этих данных: необходимо данные, разложить по категориям (видам страхования, правилам страхования, продуктам, объектам и т.д.) и выбрать ценную для бизнеса информацию. При этом нужно иметь в виду, что каждый договор обычно сопровождается 8-10 учетных записей: когда заключен договор, когда он вступает в силу, каков график платежей, перестраховывается ли договор (где, когда, в какой доле), а в случае убытка – записи по урегулированию убытка, выплатам и т.д., то есть миллион договоров порождает на порядок больше учетных записей в системе (это без учета бухгалтерии, где каждая учетная запись порождает несколько проводок). И всю эту информацию надо проанализировать, для того чтобы делать обоснованные выводы об убыточности, прибыльности того или иного продукта, клиента, направления развития. Понятно, что без применения информационных технологий такую работу полноценно выполнить невозможно.

Основной целью всех прикладных задач ИТ является достижение максимальной эффективности интеллектуального труда на основе оптимального распределения рабочего времени в интересах реализации поставленных задач. Автоматизация основных функций и процессов в деятельности страховой компании позволяет не только повысить производительность труда персонала и освободить высококвалифицированных специалистов от выполнения многих рутинных операций, но и создать необходимые условия для широкого внедрения маркетингового инструментария в интересах дальнейшего развития бизнеса.

Реальные условия функционирования компании требуют объективного и всестороннего финансового анализа процессов совершаемых коммерческих операций. Объективная необходимость выполнения такого анализа в страховой организации обусловлена высокой социальной значимостью. Финансовый анализ организации позволяет выявить эффективность функционирования экономического субъекта, недостатки в работе, причины их возникновения и позволяет на основе полученных результатов выработать конкретные рекомендации по оптимизации его деятельности.

1.2 Разработка аналитической системы

Профессиональное управление финансами неизбежно требует глубокого анализа, позволяющего наиболее точно оценить неопределенность ситуации с помощью современных методов исследования. В связи с этим существенно

возрастает приоритет и роль финансового анализа, основным содержанием которого является комплексное системное изучение финансового состояния предприятия. Качество принимаемых управленческих решений на уровне хозяйствующего субъекта зависит от качества их аналитического обоснования [1]. Получение большого числа ключевых информативных показателей, дающих объективную оценку финансового состояния предприятия, является основной целью финансового анализа. К тому же их предназначение – обеспечить информацией руководителей различных уровней внутрифирменного управления, ответственных за достижение конкретных производственных целей [2]. Информация, необходимая для принятия оперативных управленческих решений, в первую очередь, относится к издержкам производства и поэтому должна поступать в максимально короткий срок. Она обобщается или детализируется в соответствии с потребностями управления и формируется с учетом задач перспективного развития организации.

Выбор направления и цели каждого нового проекта, который осуществлялся подразделениями ИТ на заре их истории, казался совершенно ясным. Просто выполнение некоторой задачи или функции, требующей больших затрат ручного труда, необходимо было возложить на автоматизированную систему, а экономия за счет снижения трудозатрат и ускорения выполнения этой функции или задачи служила достаточно "очевидным" обоснованием целесообразности проекта. Сбор данных вручную необходимо было заменить автоматизированным процессом, поддерживающим оперативные требования некоторого подразделения компании. К сожалению, многие из этих проектов были основаны на весьма слабом взаимодействии между бригадой разработчиков информационной системы (ИС) и основными подразделениями компаний. При этом успех или неудача проекта зависели от уровня понимания бизнес-процессов, которым обладала бригада разработчиков ИС. Во многих случаях разработчики имели достаточно опыта в той области деятельности компании, которая подлежала автоматизации, и конечный продукт, передаваемый подразделению компании, имел потрясающий успех. В других случаях разработчики прилагали огромные усилия для проведения необходимого анализа бизнес-процессов, и конечный продукт, поставляемый соответствующему подразделению, также отвечал ожиданиям пользователей. Во всех остальных случаях разработчики обладали весьма смутным представлением о функциональных требованиях; они прилагали огромные усилия, однако конечный продукт, поставляемый пользователям функциональных подразделений, оставлял желать лучшего.

Поскольку неявно предполагалось, что автоматизации подлежали именно бизнес-процессы, используемые организацией, то именно они и служили отправной точкой анализа системы для специалистов бригады разработчиков ИС. Данные, необходимые для поддержки существующих бизнес-процессов, анализировались применительно к каждому отдельному бизнес-процессу и включались в соответствующий проект сегмента системы или в подсистему.

Поскольку весь проект рассматривался под углом зрения отдельных бизнес-процессов, на этом этапе развития ИТ еще не нашел широкого признания тот факт, что данные в организации можно использовать совершенно различными, причем заранее не предсказуемыми способами. Понадобилось также определенное время и на признание еще одного важного для нас факта: если по мере развития той или иной сферы деловой деятельности соответствующие бизнес-процессы могут подвергаться радикальным изменениям, то данные, необходимые компаниям для ведения деловых операций, остаются практически неизменными. Например, банки в течение многих лет ведут учет данных о вложении и снятии финансовых средств, балансе и клиентах. В интересах маркетинга финансовых услуг виды банковских счетов и правила их ведения изменились, однако данные, необходимые для обслуживания счетов, остались неизменными. До тех пор пока этот перелом в сознании не произошел, верх одерживал подход к автоматизации, ориентированный на процессы.

Создание информационно-аналитической системы всегда направлено на решение задач управления во всех его аспектах - управления финансовыми, кадровыми, техническими ресурсами и т.д. Поэтому, в конечном счете, отдача от внедрения информационно-аналитической системы будет состоять в резком повышении эффективности управления: оперативного принятия сбалансированных решений, возможности долговременного планирования и прогнозирования и т.д. Эффективность управления достигается, в том числе, и за счет существенного возрастания обоснованности принятых решений на основе больших выборок точных данных и огромного аналитического аппарата. Следует подчеркнуть, что информационно-аналитическая система - это не готовый продукт или семейство продуктов. Успех реализации информационно-аналитической системы зависит от правильного выбора и максимально тесной интеграции используемых для его реализации средств.

Еще один аспект анализа деятельности страховой компании связан с планированием развития компании, разработкой новых страховых продуктов и договоров, выбором направлений инвестирования средств и т.д. В этом случае накопленная в системе информация - это те «исторические» данные, которые совместно с комплексом внешней информации ложатся в основу прогнозирования ситуации и выработки решений по ее изменению с целью достижения более высоких показателей бизнеса (прибыли, объемов продаж и пр.). В дальнейшем эти решения реализуются, накапливаются новые фактические данные, которые вновь анализируются для внесения необходимых корректирующих изменений в деятельность компании. Это своего рода цикл, который эффективно поддерживают информационно-аналитические системы и хранилища данных (Рисунок 1.1).

Развитие ИС, ориентированных на нужды функциональных подразделений организаций, привело к разработке компьютерных систем, которые можно образно назвать информационными "бункерами" или "скважинами". Эти системы могли предоставить детализированные данные по

определенной сфере бизнеса, но не были способны интегрировать информацию из других сфер.

Построенная им математическая теория реляционных баз данных на несколько лет опередила появление реальных прототипов программных продуктов, знаменовавших собой начало новой технологической эры в области обработки данных — эры систем управления реляционными базами данных (СУРБД). Оригинальная теория реляционных баз данных Кодда включает некоторые строгие математические понятия, например идею домена, которая до сих пор не нашла воплощения в промышленном программном обеспечении (ПО) систем управления базами данных. До настоящего времени некоторые большие операционные системы используют технологию плоских файлов или аналогичные технологии. Если говорить прямо, то теория опередила реальную практику на много лет.

Конечно, с тех пор и теория и практика обработки данных шагнули далеко вперед, однако теория не смогла преодолеть существующих в действительности противоречий между стремлением к эффективной обработке данных, с одной стороны, и ограниченным бюджетом и необходимостью поддержки бизнес-процессов в рабочем состоянии — с другой. Поэтому сегодняшняя практика по-прежнему отстает от теории. Одной из особенностей технологии реляционных баз данных, которая обещала облегчить жизнь ИТ-специалистам, было снижение объема избыточных данных. Общие данные можно было поместить в общую таблицу или информационный объект, а специализированные или повторяющиеся данные — в связанные информационные объекты. Многие функциональные подсистемы могли использовать одни и те же данные для различных функциональных потребностей. Данные, общие для различных функциональных подсистем, могли храниться и сопровождаться в одном информационном объекте, и все функциональные подсистемы могли обращаться за необходимой информацией к этому единственному объекту. Появление технологии реляционных баз данных вместе с запоминающими устройствами прямого доступа сулило революцию в области ИТ, в частности в подходах к архитектурным решениям для информационных систем. Однако то, что великолепно выглядело в теории, оказалось не так просто реализовать на практике. Проекты информационных систем, обслуживающих бизнес-процессы организаций, опирались на технологию плоских файлов.

Потребители корпоративных данных из функциональных подразделений теперь повторяли путь, пройденный ИТ-подразделениями, и разрабатывали собственную неупорядоченную среду принятия решений. Служащие разных подразделений создавали собственные базы данных со своей структурой; определения данных и способы вычисления значений полей различались от одного пользователя к другому. Эти новые базы данных содержали данные из внешних источников, причем разные пользователи могли выбирать различные источники. Поскольку хранящиеся в ПК пользовательские данные одвергались автоматизированному обновлению, они быстро устаревали. выводы.



Рисунок 1.1 – Цикл реализации информационных решений

1.3 Требования к разрабатываемой информационной системе

Основными требованиями к АИС являются:

- поддержка высокой скорости;
- поддержка внутренней непротиворечивости данных;
- автоматизированный ввод информации по первичным документам;
- ввод информации в реальном времени;
- автоматическое получение необходимой аналитической и бухгалтерской информации;
- возможность получения и сравнения данных;
- наличие удобных утилит просмотра и работы с данными, наглядность представления информации;
- полнота и достоверность хранимой информации;
- поддержка качественного процесса пополнения данных.

Схема работы системы, учитывая вышеперечисленные требования представлена на рисунке 1.2.

Порядок работы АФС выглядит примерно следующим образом. Сведения отчетности вводятся в хранилище данных и программа проводит тщательный анализ на предмет непротиворечивости показателей. После ввода и контроля исходных данных выполняется их преобразование к сопоставимому виду.

Для выполнения поставленных задач необходим инструментарий для преобразования больших объемов детализированных данных в форму, которая удобна для стратегического планирования и реорганизации бизнеса и необходима специалисту, ответственному за принятие решений. При этом происходит слияние из разных источников различных сведений в требуемую предметно-ориентированную форму с использованием различных методов анализа.

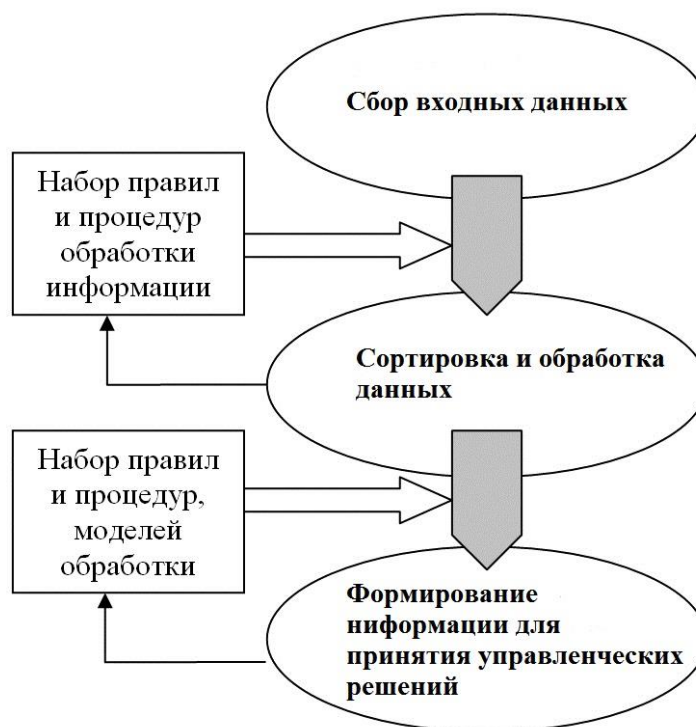


Рисунок 1.2 – Схема работы АИС

Учитывая развитие современных технологий, предлагается новое решение – хранилище данных. Особенность этой новой технологии в том, что она предлагает среду накопления данных, которая не только надежна, но по сравнению с распределенными СУБД и оптимальна в отношении доступа к данным и манипулирования ими. Хранилище информации предназначено для хранения, оперативного получения и анализа интегрированной информации по всем видам деятельности организации.

Основной задачей данной проектируемой системы является повышение эффективности работы финансовых аналитиков по управлению финансами страховой организации. Финансовый аналитик в страховой организации оценивает, как управленческие решения и планы руководства скажутся на финансовом состоянии страхового бизнеса. В его должностные обязанности входит анализ информации о финансовой деятельности компании для прогнозирования экономических условий с целью дальнейшего использования при принятии инвестиционных решений анализ финансовой информации о деятельности компании для расчета будущих доходов и расходов, сбор и анализ информации, необходимой для последующего составления финансовых отчетов компании и вычисления финансовых показателей ее деятельности [3].

Значительную часть работы финансового аналитика берет на себя разрабатываемая система. Можно также заметить, что большое значение имеет человеческий фактор, в силу которого результативная информация может быть некорректной, несвоевременной, или по каким-либо причинам утеряна. Помимо этого, намного выгоднее использование среды накопления данных,

оптимизированной для выполнения сложных аналитических запросов управленческого персонала. Запросами управленческого персонала в данном случае являются расчёт и получение необходимых финансовых показателей, а также составление проанализированной отчётности в виде представлений. К тому же, в условиях быстро меняющегося рынка, нестабильной политической обстановки, ужесточающейся конкуренции к актуальности деловой информации, и ее достоверности предъявляются новые требования. Процесс производства информации должен укладываться в жесткие временные рамки. Именно автоматизированная система, на основе хранилища данных может и должна обеспечить требуемые временные характеристики. После того, как традиционная система обработки данных реализована и начинает функционировать, она становится ровно таким же самостоятельным объектом, как и любой производственный процесс. В этом смысле данные обладают ровно теми же свойствами и характеристиками, что и любой промышленный продукт: сроком годности, местом складирования (хранения), совместимостью с данными из других производств, рыночной стоимостью, транспортабельностью, комплектностью, ремонтпригодностью и т.д.

В такой системе данные легко суммируются в произвольных разрезах, в том числе для мгновенного ответа на вопросы, возникающие спонтанно; важные цифры выделяются так, что их невозможно пропустить; серии данных иллюстрируются наглядными диаграммами и графиками. Любой возможный отчет, строящийся как комбинация имеющихся данных, может быть получен немедленно. Эти свойства новой технологии меняют стиль работы руководителя и создают условия для принятия более взвешенных решений. Ценность собранных данных бывает очень высока, а создание всех вариаций отчетов, которые интересуют разных потребителей информации, может стоить столь дорого и отнимать так много времени, что к моменту получения отчетов данные устареют [4].

Использование информационных технологий предоставляет следующие преимущества:

- а) Экономия рабочего времени.
- б) Отсутствие дублирования.
- в) Избежание ошибок.
- г) Гарантия оперативного обновления форматов представления информации в электронном виде.
- д) Возможность получения информационной выписки.
- е) Оперативное информирование.
- ж) Повышение оперативности обработки информации, исключение технических ошибок.

Ядром информационной системы являются хранимые данные. Совокупность информационных объектов, информационно отображающих свойства системы и протекающие в ней процессы, называют информационным пространством. Характерным свойством информационного пространства является его структурированность. Это означает, что выделены его элементы,

установлены связи между ними, введены обозначения, элементы и связи упорядочены. Свойство структурированности в разных видах информационных пространств может быть выражено в разной степени. Высокий уровень обеспечивает возможность представления информации в виде документов и манипулирования данными с помощью программно-технических средств информационных систем. Поэтому одним из требований разрабатываемой системы является предоставление пользователю (финансовому аналитику) данных в читабельном виде, соответствующему бухгалтерскому виду. Информация должна быть представлена в виде готового отчёта, на основании которого можно сделать выводы о работе компании в короткие сроки.

В качестве элементов структуры информационного пространства выступают единицы информации. Это понятие, которое рассматривается в теории экономических информационных систем (ЭИС), выражает сущностное или смысловое наполнение элемента ИП. Под единицей информации понимают «набор символов, которому придается определенный смысл». Рассматривается система единиц информации, которая имеет довольно сложную иерархическую структуру. Выделяют несколько уровней единиц информации в зависимости от смыслового (семантического) значения, его наполненности. По возрастанию содержательности понятия определены следующие единицы информации: реквизит и составная единица информации (СЕИ), которая включает в себя такие единицы как показатель и база данных.

Элементарной единицей информации нижнего уровня является реквизит. Это информационное отображение свойства объекта, какого-либо процесса или явления. Сообщения состоят из определений свойств объектов, предметов, явлений, складывающихся некоторым образом из соответствующих реквизитов. Следует заметить, что синонимом понятия реквизит является атрибут, широко используемый в литературе по базам данных термин.

Отсюда составная единица информации собирается из набора соответствующих определению данного объекта реквизитов и представляет собой информационное отображение объекта или его части. Разновидностью составной единицы информации является показатель. Это сложное понятие, которое является основной идеей данной автоматизированной системы. Имеются его различные определения. Одни авторы подчеркивают сущностный смысл или характер, привязанный к предметной области, в частности экономической. Другие исходят из формально-структурного подхода, ориентированного на структуризацию содержащейся в показателе информации в целях приспособления его структуры для эффективного использования в информационной системе. Результаты такой структуризации используются и в информационно-аналитических системах. То есть составляется масштабная система расчётов и показателей [5]. Для поддержания данных в актуальном состоянии пользователи тратили много времени на сопровождение собственных баз данных. Если они не обновляли свои локальные данные, то из неверных данных делались неправильные. Корпоративные пользователи, попытавшиеся работать с неструктурированной унаследованной

информационной средой, сразу же столкнулись с проблемой понимания взаимосвязи между бункерами данных. Не существовало единого способа идентификации записей, общих для всех бункеров данных. Отсутствие общего взгляда на то, какие элементы данных в разных бункерах считать одинаковыми, приводило в замешательство как корпоративных пользователей, так и представителей ИТ-подразделений. Специалисты функциональных подразделений обвиняли во всем специалистов ИТ-подразделений, поскольку два разных отчета, которые должны были вырабатывать одинаковые значения для одинаковых показателей, предоставляли совершенно разные цифры. Это приводило к тому, что лица, принимающие или готовящие решения в организации, докладывали на совещании о совершенно разных значениях, относящихся к одним и тем же показателям.

Основной причиной недоразумений была хаотичная, неструктурированная среда поддержки принятия решений. Если бы организация смотрела вперед и заранее планировала отчеты, то одной процедуры или набора общих процедур по извлечению данных было бы достаточно для получения необходимых данных для всех отчетов. Данные могли бы храниться централизованно. Однако, поскольку большинство отчетов разрабатывалось независимо, спланировать централизованное хранение данных было невозможно. Проницательный читатель может подумать, что реляционные базы данных наверняка должны были решить проблему несогласованности данных или, образно говоря, проблему "хаоса данных". Хотя технология реляционных баз данных во многом помогла при разработке новых функциональных подсистем, она не решила вопроса бункеров данных и стихийного развития систем.

Функциональные подсистемы по-прежнему разрабатывались безотносительно друг к другу. Не существовало общего проекта, не создавалось общих систем уникальных идентификаторов или схем кодирования, и технология реляционных баз данных была не в силах решить эту проблему.

Таким образом, в сфере ИТ складывалось все более тяжелое положение. Нарастание сложности сопровождения макаронных программ извлечения данных и генерации отчетов шло достаточно медленно для того, чтобы кто-нибудь спохватился и забил тревогу. На биологических факультетах университетов часто можно услышать историю про лягушку. Если лягушку поместить в посудину с горячей водой, то она, спасаясь, выпрыгнет из нее. Но если лягушку поместить в посудину с холодной водой, а затем медленно нагревать ее, вполне возможно, что она сварится. Точно так же ситуация в сфере ИТ медленно "разогревалась", и внезапно технологическое сообщество обнаружило, что стало слишком горячо.

При перенесении данных из оперативной системы в хранилище перед загрузкой они преобразуются. Различного рода несоответствия в кодировании, типах данных и других "свойствах", присущих исходной системе, устраняются. Это также отличный повод для анализа данных исходной системы и приведения в соответствие всех расхождений реального состояния данных с их

типами и кодами, представленными в документации. Вообще говоря, построение хранилища данных открывает возможность избавиться от нежелательных "свойств" оперативной системы.

Другим важным свойством, отличающим хранилище данных от оперативной системы, является то, что оно не разрушается. В то время как оперативная система выполняет над хранимыми данными операции обновления, удаления и вставки, в хранилище помещается большой объем данных, которые, будучи раз загруженными, уже никогда больше не подвергаются каким-либо изменениям. Характерной особенностью хранилища данных является то, что два разных корпоративных пользователя, выполняющих один и тот же запрос к хранилищу данных в разное время, получают один и тот же результат. Это исключает ситуации, при которых незапланированное извлечение данных и генерация отчетов приводят к различным результатам.

Еще одна особенность хранилища данных – независимость от времени. Если оперативная система содержит только текущие данные, то системы хранилищ данных содержат как исторические данные, так и данные, которые имели статус текущих при последней загрузке хранилища. Временные рамки данных, содержащихся в хранилище, изменяются в широких пределах в зависимости от типа системы.

Естественным способом преодоления непредсказуемости в развитии как систем поддержки принятия решений, так и систем обработки транзакций является включение в жизненный цикл программного обеспечения фаз планирования и анализа. Проекты, связанные с программным обеспечением, терпят неудачу из-за недопонимания вопросов планирования, рамок и масштабов проекта, системных требований или характера данных. Многие практики разработали собственные методы, направленные на то, чтобы справиться с каждой из этих причин, приводящих к неудачам в проектировании.

Конструирование корпоративной модели данных для крупной организации – монументальная задача, и большая часть этих усилий пропала даром. Проблема заключалась в масштабах проекта, недостатках планирования, а также в подвижном, ускользающем характере цели. Что касается крупномасштабных проектов, то в этом случае весьма сложно достичь ощутимых результатов, прежде чем спонсор начнет предполагать, что проект обречен на неудачу. Для хорошего плана характерно наличие нескольких промежуточных результатов в виде прототипов или отдельных поставляемых заказчику компонентов, представляющих реальную ценность для организации. Еще одна отличительная черта хорошего плана и умелого управления проектом – возможность получить готовые компоненты прежде, чем существенно изменится модель данных. Таким образом, успешный проект по моделированию корпоративных данных обычно завершается вместе с созданием моделей для части предприятия.

Одним из подходов, позволивших снизить объем избыточных данных в организации, стала методология, в основу которой была положена так называемая модель "сущность-связь". В литературе эта методология получила название *ER-моделирования (entity-relationship (ER) modeling)*. Если говорить очень коротко, то модель "сущность-связь" была разработана для моделирования данных, относящихся к некоторым предметным областям или объектам, которые представляют интерес для организации, атрибутов, которыми обладают эти объекты, и отношений между этими объектами.

Объекты представляются в модели сущностями, атрибуты – полями, а отношения – связями между сущностями. Можно также сказать, что моделирование в терминах "сущность-связь" представляет собой определенный способ мышления. Поскольку в рассматриваемой модели данных каждый атрибут или поле присутствует единожды, то различные значения в разных полях никак не могут представлять одни и те же данные. Построенная на основе ER-модели компьютерная система для обновления данных, относящихся к определенному экземпляру объекта, просто обновляет одно поле реляционной базы данных. Например, почтовый адрес клиента хранится в определенной ячейке, представляющей собой то единственное место, в котором следует при необходимости обновить почтовый адрес клиента. Это позволяет избежать проблемы нерациональной загрузки системного процессора (system processing unit – SPU) или центрального процессора (central processing unit – CPU) избыточными заданиями. В конечном итоге обработка и обновление данных ускоряются. Все эти трудности, связанные с бункерами данных, неуправляемостью среды оперативных данных и среды поддержки принятия решений, сделали *корпоративную модель данных (enterprise data model)* весьма популярной, однако и ее постигла неудача. Главная идея, лежащая в основе корпоративной модели данных, состояла в том, чтобы начинать проект с инвентаризации всех данных, которые используются в компьютерной системе. Данные о данных, так называемые *метаданные*, должны были размещаться в специальном хранилище – *репозитории*.

После составления реестра всех полей данных можно было двигаться дальше: осуществлять процесс идентификации отношений между данными в различных мастер-записях и полями, содержащими ту же информацию.

Перспективные потребности организации данных также можно было спланировать и учесть в корпоративной модели. Конструирование корпоративной модели данных для крупной организации — монументальная задача, и большая часть этих усилий пропала даром. Проблема заключалась в масштабах проекта, недостатках планирования, а также в подвижном, ускользающем характере цели. Что касается крупномасштабных проектов, то в этом случае весьма сложно достичь ощутимых результатов, прежде чем спонсор начнет предполагать, что проект обречен на неудачу. Для хорошего плана характерно наличие нескольких промежуточных результатов в виде прототипов или отдельных поставляемых заказчику компонентов, представляющих реальную ценность для организации. Еще одна отличительная

черта хорошего плана и умелого управления проектом – возможность получить готовые компоненты прежде, чем существенно изменится модель данных. Таким образом, успешный проект по моделированию корпоративных данных обычно завершается вместе с созданием моделей для части предприятия. Для больших организаций эти модели, какой бы степени детализации они не достигали, никогда нельзя назвать корпоративными.

Конечно, учитывая все, что нам известно о методологиях, моделях и инструментальных средствах, применяемых в ИТ, можно предположить, что организации больше не строят новых информационных систем, организованных вокруг бункеров данных. Но это неверно! Небольшие организации, особенно быстрорастущие компании, строят новые системы в спешке, не планируя общего взгляда. Это свойственно, в частности, молодым компаниям, которые еще не развили в достаточной мере корпоративную культуру и не наладили координацию и бригадную работу различных функциональных подразделений. Чтобы проект хранилища данных был успешным, зачастую важно разработать высокоуровневую корпоративную модель данных. Эта модель может также способствовать росту небольших компаний. Цель модели – убедиться в том, что разработчики не повторяют пройденный ИТ-подразделениями путь создания нового набора бункеров данных. Чтобы избежать создания теперь уже бункеров хранилищ данных, следует разрабатывать модель предприятия достаточной глубины для выявления связей в системе транзакций и системе поддержки принятия решений.

Методы, используемые при моделировании данных систем обработки транзакций, отличаются от методов, применяемых при моделировании данных систем поддержки принятия решений. Если при создании нормализованной модели типа "сущность-связь" и проектировании соответствующей реляционной базы данных основная цель состоит в увеличении производительности операций обновления данных, то создание модели данных системы поддержки принятия решений преследует цель повышения эффективности обработки запросов к базе данных. Эффективность выполнения запросов выступает одной из главных задач при создании систем поддержки принятия решений, поскольку им свойственна скорее массовая загрузка данных, нежели их периодическое обновление.

Поскольку при разработке систем поддержки принятия решений преследуется цель повышения скорости обработки запросов, то обычно при этом используется организация данных, отличная от применяемой в случае систем обработки транзакций. Запросы, адресуемые системе поддержки принятия решений, касаются множества фактов и измерений (аспектов), имеющих определенный смысл для бизнес-процессов. Факты обычно представляют собой числовые параметры, например объем сбыта, а обычно используемые измерения носят дискретный характер и отображают такие бизнес-факторы компаний, как торговый персонал, товары, время и географические регионы. Уникальная для систем поддержки принятия решений

модель данных носит название схема типа "звезда" (star schema) или размерностная модель данных (dimensional data model), которая представлена на рисунке 1.3.

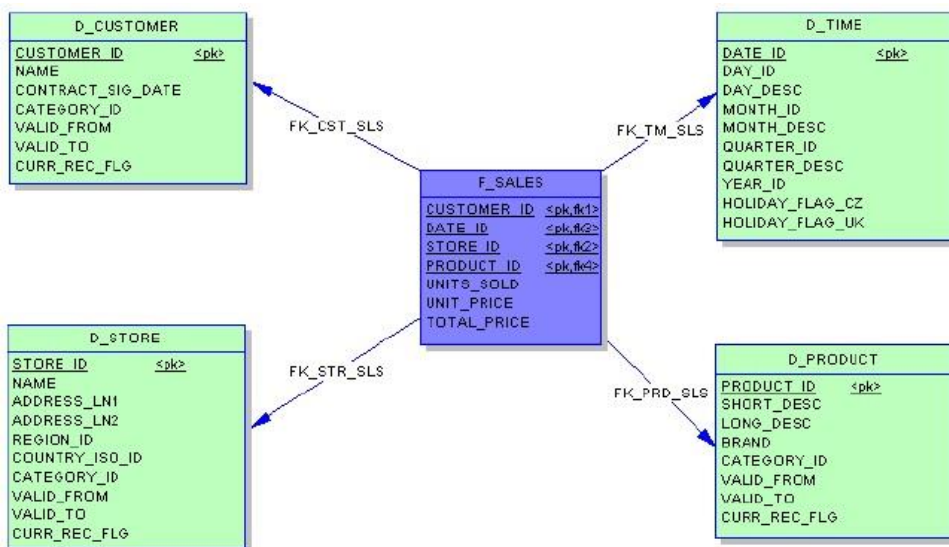


Рисунок 1.3 – Схема типа "звезда"

С практической точки зрения наилучшим подходом к построению такой модели данных является высокоуровневый анализ использования данных как в системе обработки транзакций, так и в системе поддержки принятия решений. Результатом I наполнения этой задачи должен быть документ, отображающий данные оперативной системы в систему поддержки принятия решений. Если эту задачу выполнить в начале этапа планирования, можно гарантировать, что в основу разработки системы поддержки принятия решений положен проект базы данных, приспособленной к получению и загрузке данных из оперативной системы. Чтобы получать и загружать данные из оперативной системы, соответствующие поля должны существовать в ее базе данных; кроме того, эти поля должны быть заполнены, т.е. обладать значениями. Типы и длина полей в обеих системах также должны совпадать. Более того, если оперативная система и система поддержки принятия решений могут использовать одинаковые схемы кодирования, это уменьшает длительность циклов загрузки данных и сложность, избавив разработчиков и системных администраторов от лишней головной боли. Иными словами, согласованность полей двух систем повышает устойчивость и облегчает сопровождение информационного фонда предприятия. С помощью приведенной схемы и небольших объяснений любой руководитель – от главного финансового администратора и главного управляющего по информации до менеджера по линии продуктов – может понять связь между данными и организационной структурой.

1.4 Специфика разработки данного проекта

Вся проблема аналитической подготовки принятия решений имеет следующие аспекты:

- извлечение из многих источников разнородных данных, представленных в различных форматах и приведение их к единому формату и единой структуре;
- организация хранения и предоставления пользователям необходимой для принятия решений информации;
- собственно анализ, в том числе оперативный и интеллектуальный, и подготовка плановой или регулярной оценки состояния управляемого объекта в виде бумажных документов или экранных форм;
- подготовка результатов оперативного и интеллектуального анализа для эффективного их восприятия потребителями и принятия на основе адекватных решений.

Аспект, касающийся сбора и хранения информации с сопутствующей доработкой, оформился в концепцию информационных хранилищ (Data Warehouse). Эта концепция состоит в том, что сведения о деятельности предприятия или иного объекта хозяйственной или иной деятельности накапливаются в течение длительного периода времени (годы) в информационном хранилище по определенным правилам. Накопленные данные используются в различных временных режимах для анализа, как источник данных для разного рода отчетности, работы с партнерами (Reporting) и обоснования управленческих решений.

Процессы продвижения и использования данных проходят несколько этапов:

- Этап извлечения, преобразования и загрузки данных. На основе принятой системы показателей, характеризующих деятельность предприятия, подлежащих анализу и использованию в процессе принятия решений, и необходимых при разработке каких-либо документов, организуется сбор необходимых данных в хранилище и прорабатываются пути непосредственного извлечения в экстренных случаях необходимых детальных данных из первичных источников; этому этапу предшествует работа по созданию необходимой структуры перекачиваемых данных.

- Этап накопления, обеспечения готовности данных к использованию. По мере накопления в соответствующих зонах памяти выполняется периодическая загрузка данных из функциональных (транзакционных) подсистем интегрированной информационной системы (ИС) или автономных ИС, поддерживается необходимый уровень качества данных; в отдельных случаях допускается внеплановая загрузка по ситуации.

- Этап применения данных, содержащихся в хранилище, и извлекаемых напрямую из первичных источников. Для обеспечения процесса управления предприятием или другим объектом данные используются в трех основных режимах – создания плановых отчетных и других документов (Reporting),

оперативного анализа в незапланированных ситуациях (OLAP-анализ), интеллектуального или углубленного анализа (Data mining). Накопленные в хранилище данные могут быть использованы в специальных программах, обеспечивающих развитие бизнеса. Этапы работы системы представлены на рисунке 1.4.



Рисунок 1.4 – Схема функционирования системы

Хранилище данных предлагает среду накопления данных, оптимизированную для выполнения сложных аналитических запросов управленческого персонала. важным свойством, отличающим хранилище данных от оперативной системы, является то, что оно не разрушается. В то время как оперативная система выполняет над хранимыми данными операции обновления, удаления и вставки, в хранилище помещается большой объем данных, которые, будучи раз загруженными, уже никогда больше не подвергаются каким-либо изменениям. Еще одна особенность хранилища данных – независимость от времени. Если оперативная система содержит только текущие данные, то системы хранилищ данных содержат как исторические данные, так и данные, которые имели статус текущих при последней загрузке хранилища. Временные рамки данных, содержащихся в хранилище, изменяются в широких пределах в зависимости от типа системы. Однако обычно временные рамки данных, находящихся в хранилище, лежат в пределах от 15-ти месяцев до пяти лет. Данные большей давности, как правило, переносятся в архив. Характерной особенностью хранилища данных является то, что два разных корпоративных пользователя, выполняющих один и тот же запрос к хранилищу данных в разное время, получают один и тот же результат. Это исключает ситуации, при которых незапланированное извлечение данных и генерация отчетов приводят к различным результатам. Схема информационного хранилища представлена на рисунке 1.5.

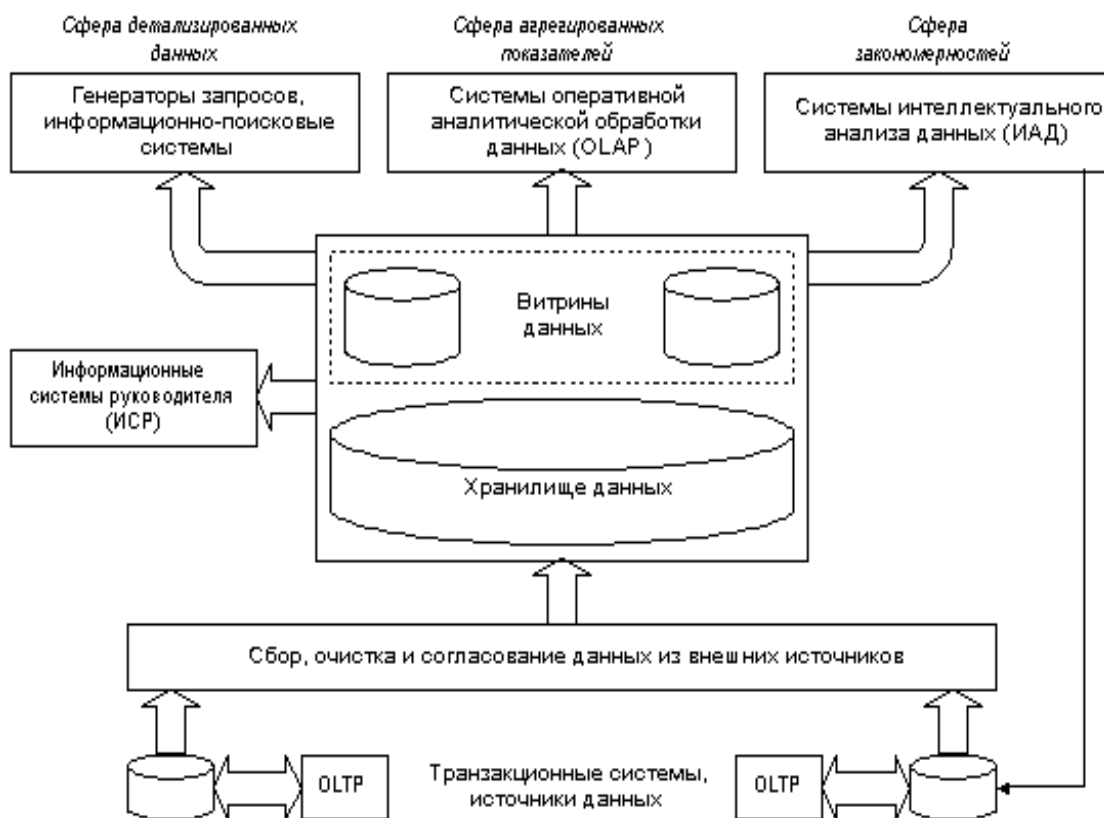


Рисунок 1.5 – Структура информационного хранилища

С хранилищами данных тесно связана еще одна структура под названием витрина данных (data mart). Иногда эти структуры хранения данных называют также киосками данных. Во многом аналогично тому, как выбор и наличие товаров в продуктовом киоске значительно уступает супермаркету, витрина данных представляет собой хранилище данных в миниатюре. Как правило, витрина содержит значительно меньше данных, охватывает всего несколько предметных областей и имеет более короткую историю. Витрины данных можно представить в виде логически или физически разделенных подмножеств хранилищ данных. Обычно они строятся для обслуживания нужд определенной группы пользователей. Подобно хранилищам данных, витрины содержат как детализированные оперативные, так и агрегированные данные. Схема работы хранилища данных представлена на рисунке 1.6.

В связи с большим объемом и сложностью проблемы собственно аспект анализа имеет два направления – оперативный анализ данных (информации), широко распространена аббревиатура англоязычного названия – On-Line Analytical Processing – OLAP. Основной задачей оперативного или OLAP-анализа является быстрое (в пределах секунд) извлечение необходимой аналитики для обоснования или принятия решения информации.



Рисунок 1.6 – Схема центрального хранилища данных и многих витрин.

Интеллектуальный анализ информации имеет также широко распространенное в русской специальной литературе англоязычное название Data mining. Предназначен для фундаментального исследования проблем в той или иной предметной области. Требования по времени менее жестки, но используются более сложные методики. Ставятся, как правило, задачи и получают результаты стратегического значения. При решении сложных задач в режиме Data mining приходится использовать весьма мощные специальные программные средства или инструменты.

Аспекты проблемы анализа и необходимые для их разрешения функции нашли выражение в соответствующих программных продуктах. Соответственно средства автоматизации анализа представлены в различных видах. Имеются комплексные информационно-аналитические системы, выполняющие в той или иной степени функции в соответствии с рассмотренными аспектами. Представленный программный продукт является целевой программной системой, выполняющие в увеличенном объеме, расширенном составе и повышенной сложности различные функции, например оперативного или интеллектуального анализа. ИАС информационно подпитывают системы поддержки принятия решений (DSS – Decision Support System).

Как правило, все инструментальные средства, предназначенные для автоматизации аналитических работ, приспособлены для обработки многомерных массивов информации; имеют также возможность импорта/экспорта данных в другие операционные среды, развитые средства визуального двумерного (2D) и трехмерного (3D) представления информации. В данном случае, функции ввода/вывода реализованы средствами применяемых технологий.

2 Разработка аналитической системы

2.1 Постановка задачи

Требуется разработать автоматизированную информационную систему анализа финансового состояния страховой компании, которая позволила бы улучшить работу компании и в частности, отдела аналитики.

Автоматизированная система позволит обрабатывать и хранить информацию управленческой отчётности страховой компании, а также анализируемую информацию о её состоянии.

Пользователь, т.е. финансовый аналитик, сможет просматривать данные бухгалтерского баланса, отчёта о прибылях и убытках, проводить оценку ликвидности, рентабельности, вероятности банкротства организации, за счёт расчёта коэффициентов, производить вычисление показателей относительной и абсолютной финансовой устойчивости, деловой активности страховой компании и т.д., а также осуществлять вертикальный и горизонтальный анализ.

В дополнительной информации можно будет увидеть данные о количестве сотрудников, договоров переданных в страхование и перестрахование, средней тарифной ставке, количестве застрахованных объектов и др.

Администратор может вносить новых данные бухгалтерского баланса, отчёта о прибылях и убытках, а также дополнительную вспомогательную информацию для расчётов.

2.2 Этап начальной разработки ХД

2.2.1 Анализ предметной области

Страховая компания – это юридически оформленная единица предпринимательской деятельности, которая берет на себя обязательства страховщика и имеет для этого соответствующую лицензию.

Деятельность страховой компании заключается в формировании на основании договоров с юридическими и физическими лицами (через продажу страховых полисов) специальных денежных фондов, из которых осуществляются выплаты страхователям денежных средств в обусловленных размерах в случае наступления определенных событий (страховых случаев).

Финансовый анализ – изучение основных показателей, параметров, коэффициентов и мультипликаторов, дающих объективную оценку финансового состояния организаций и стоимости акций компаний с целью принятия решений о размещении капитала.

В финансово-аналитической системе производится загрузка и хранение основных данных управленческой отчётности, которая представляет собой бухгалтерский баланс организации, а также отчёт о прибылях и убытках. На

основании этих данных проводится анализ финансового состояния страховой компании, а также рассчитываются показатели, параметры, коэффициенты и мультипликаторы для оценки финансового состояния организации и стоимости акций компаний с целью принятия решений о размещении капитала самостоятельно. Проводимый системой анализ финансового состояния компании может быть двух видов: горизонтальный и вертикальный. Параметры, которые рассчитываются в системе включают в себя группировку активов и пассивов, коэффициенты вероятности банкротства, коэффициенты ликвидности баланса, показатели финансовой устойчивости, относительные показатели финансовой устойчивости, показатели оценки достаточности страховых резервов, показатели зависимости от перестраховщика, показатели рентабельности и деловой активности компании.

2.2.2 Базовые сущности хранилища данных

Вся информация в хранилище данных делится по двум направлениям: загружаемые данные управленческой отчётности и рассчитываемые на основе неё показатели финансового анализа. Эти два вида определяются сущностью «NAPR», которая содержит следующие атрибуты:

- ID направления.
- Название направления.

Вся система основана на обработке показателей различного вида, так как показателями являются не только рассчитываемые системой данные, но и все загружаемые в неё значения управленческой отчётности. Поэтому все параметры необходимо разделить по видам, для чего создана сущность «VID», которая содержит следующие атрибуты:

- ID вида.
- Название вида.

Для определения и объединения всех показателей создана сущность «POKAZATEL», содержащая атрибуты:

- ID показателя.
- ID направления.
- ID вида.
- Название показателя.

В хранилище данных автоматизированной системы анализа финансового состояния страховой компании в первую очередь происходит загрузка данных бухгалтерского баланса организации. Сущность «BALANCE» содержит следующие атрибуты:

- ID показателя.
- Сумма.
- Дата.

Отчёт о прибылях и убытках также отражает состояние управленческой отчётности компании. Сущность «OPIU» содержит следующие атрибуты:

- ID показателя.

- Сумма.
- Дата.

Значение всех рассчитанных показателей формируется в процессе работы системы в общую сущность под названием «CALC», которая содержит атрибуты:

- ID показателя.
- Значение.
- Дата.

В системе также требуется загрузка дополнительных вспомогательных параметров, необходимых для расчёта показателей финансового анализа. Эти данные определены в сущность «SUPPORT», чьими атрибутами являются:

- ID показателя.
- Значение.
- Дата.

2.2.3 UML диаграммы

Для моделирования статических объектов в объектно-ориентированной концепции UML существует 12 диаграмм, но так как большинство объектов хранилища разрабатываемой системы являются статическими, используем самые основные:

- а) Диаграмма компонентов.
- б) Диаграмма прецедентов.
- в) Диаграмма классов.
- г) Диаграмма развертывания.

2.2.4 Диаграмма компонентов

Диаграмма компонентов показывает набор компонентов и отношений между ними.

Компонент – это физически заменяемая часть системы, которая имеет набор интерфейсов и обеспечивает их реализацию.

Исполняемый модуль в виде пользовательского приложения FinInsuranse.exe обращается за получением или добавлением информации к хранилищу данных FinInsuranse.mdf аналитической системы. Но обращается не напрямую, а через Oracle SQL Developer и OraClient11g. Таким образом, Oracle SQL Developer и OraClient11g является интерфейсом между программой и хранилищем данных, обеспечивая их взаимодействие. Диаграмма компонентов представлена на рисунке 2.1.

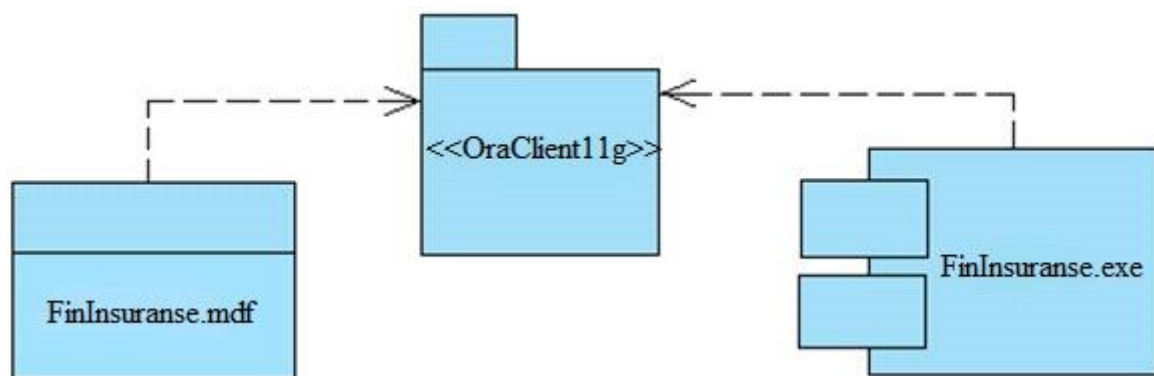


Рисунок 2.1 – Диаграмма компонентов

2.2.5 Диаграмма прецедентов

Для характеристики взаимодействия пользователей с хранилищем данных применяется диаграмма прецедентов или вариантов использования. Диаграмма прецедентов представлена на рисунке 2.2.



Рисунок 2.2 – Диаграмма прецедентов

Из нее видно, что пользоваться хранилищем может только одна группа людей, у которых нет возможности редактировать данные, так как ядром системы является хранилище данных.

2.2.6 Диаграмма классов

Диаграмма классов, показанная на рисунке 2.3, в UML является частным случаем ER-диаграммы. ER-диаграммы используются для логического проектирования баз данных. Главное их отличие: в ER-диаграмме уделяется внимание структуре данных, а в диаграмме классов – поведению классов.

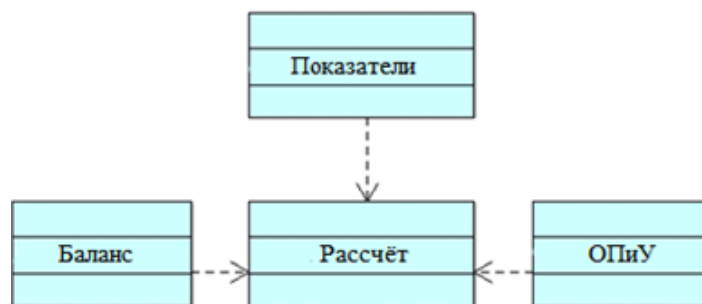


Рисунок 2.3 – Диаграмма классов

2.2.7 Диаграмма развертывания

Диаграмма развертывания, представленная на рисунке 2.4, Deployment diagram в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать приложение диаграмма развертывания должна показывать, какие аппаратные компоненты ("узлы") существуют (например сервер базы данных, сервер приложения), какие программные компоненты ("артефакты") работают на каждом узле (например, приложение, хранилище данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, OLEDB, RMI). Диаграмма развертывания представлена на рисунке 2.4.

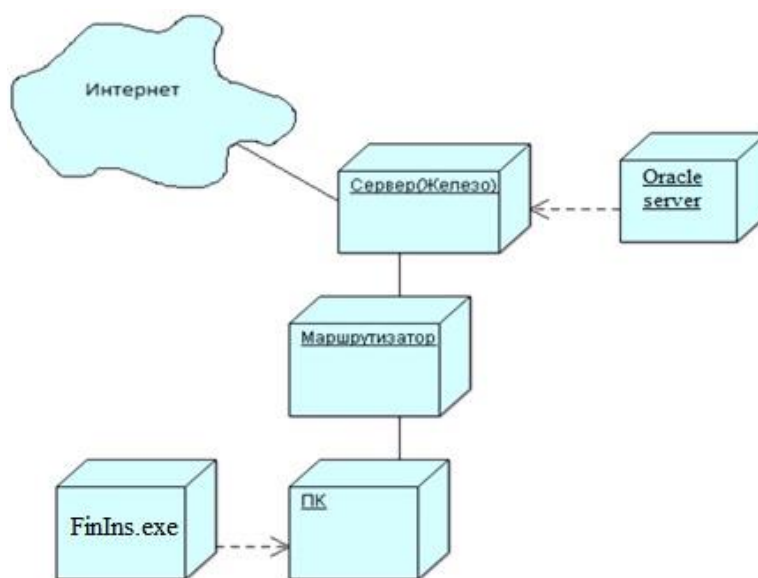


Рисунок 2.4 – Диаграмма развертывания

Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь подузлы, которые представляются как вложенные прямоугольные параллелепипеды. Один узел диаграммы развертывания может

концептуально представлять множество физических узлов, таких как кластер серверов баз данных.

Существует два типа узлов:

- узел устройства;
- узел среды выполнения.

Узлы устройств – это физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения, такие как обычные ПК, мобильные телефоны. Узел среды выполнения - это программный вычислительный ресурс, который работает внутри внешнего узла и который предоставляет собой сервис, выполняющий другие исполняемые программные элементы.

Диаграмма развертывания нашей системы очень проста, так как не требует специального оборудования и может устанавливаться на любой ПК в компьютерных классах.

2.3 Проектирование ХД

2.3.1 Концептуальное проектирование

Учитывая особенности данной системы, рассмотренной в анализе предметной области, выделим основные таблицы и связи между ними, предоставленные на предварительной ER-диаграмме, которая приведена на рисунке 2.5.

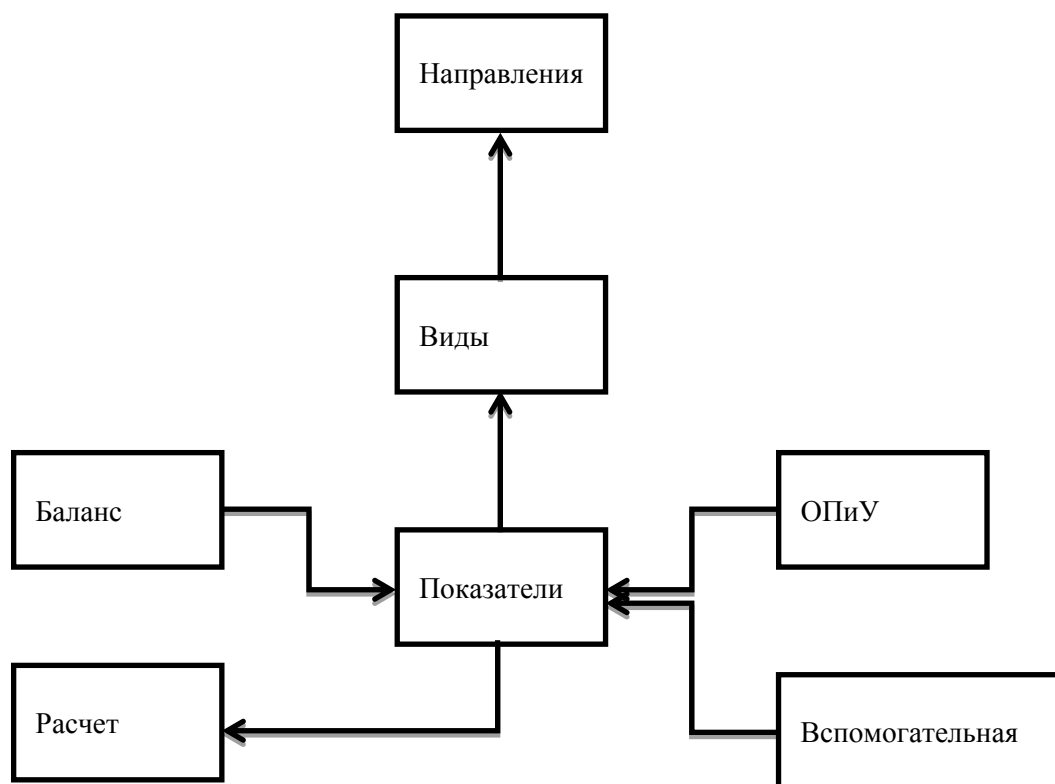


Рисунок 2.5 – Общая ER-диаграмма

На основе этапа начальной разработки сформируем бизнес-правила. Затем определяем связи между сущностями, в основном опираясь на описание операций. Точнее связи сущности базируются на бизнес-правилах, построенных на основе подробного описания операций.

2.3.2 Бизнес правила

В результате разработки первоначальной ER - модели, необходимо определить связи между сущностями, опираясь на операции. Если сказать точнее, то связи между сущностями базируются на основе бизнес правил:

1) В систему не могут загружаться данные бухгалтерского баланса с одинаковыми суммами и одной и той же датой.

2) В систему не могут загружаться данные отчёта о прибылях и убытках с одинаковыми суммами и одной и той же датой.

3) В систему не могут загружаться вспомогательные данные с одинаковыми значениями и одной и той же датой.

4) В системе не может содержаться рассчитываемая информация с одинаковой суммой и одной и той же датой.

5) Все показатели можно разделить на два направления: баланс и расчет. Одному показателю соответствует только одно направление. В то время как каждому направлению соответствует множество показателей (Рисунок 2.6).

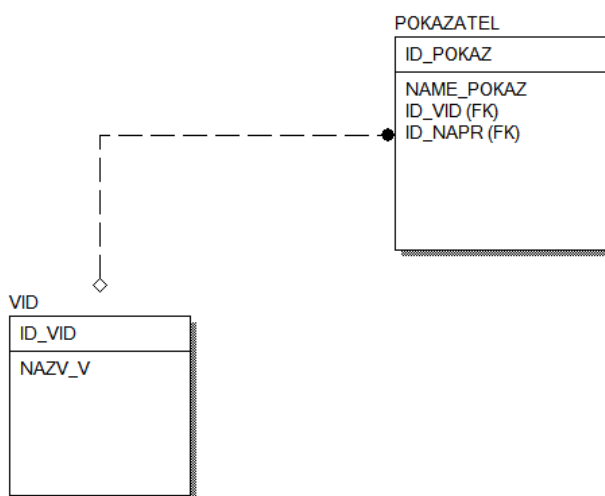


Рисунок 2.6 – Сегмент ER -диаграммы для бизнес-правила 5

6) Все показатели можно разделить на семнадцать видов. Одному показателю соответствует только одно направление. В то время как каждому направлению соответствует множество показателей (Рисунок 2.7).

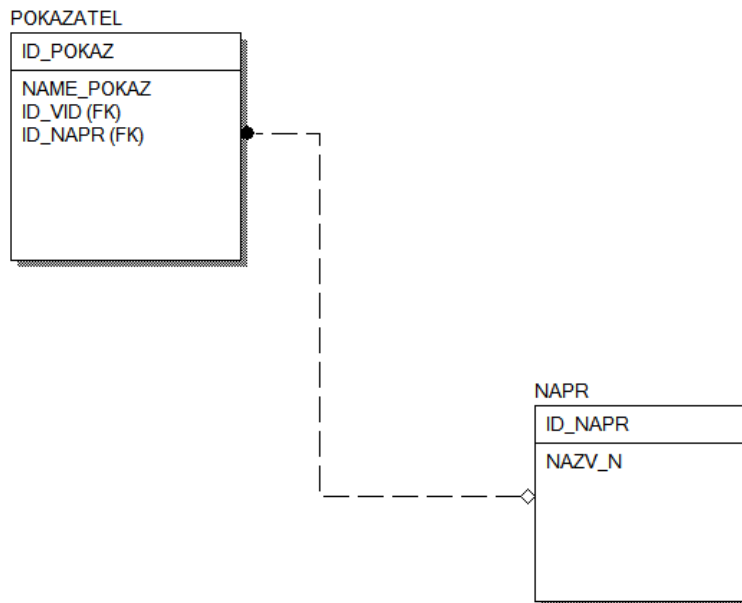


Рисунок 2.7 – Сегмент ER -диаграммы для бизнес-правила 6

7) Каждому значению, загружаемому в таблицу «BALANCE» соответствует только один вид, одно направление и ID показателя. В то время как под это соответствие попадает множество значений в таблице «BALANCE» (Рисунок 2.8).

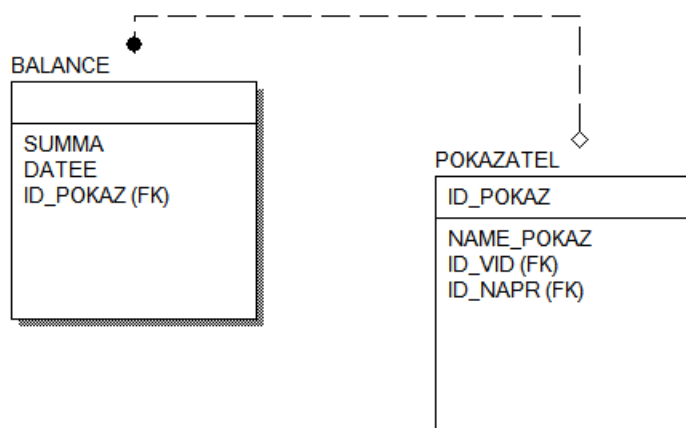


Рисунок 2.8 – Сегмент ER -диаграммы для бизнес-правила 7

8) Каждому значению, загружаемому в таблицу «ОПИУ» соответствует только один вид, одно направление и ID показателя. В то время как под это соответствие попадает множество значений в таблице «ОПИУ» (Рисунок 2.9).

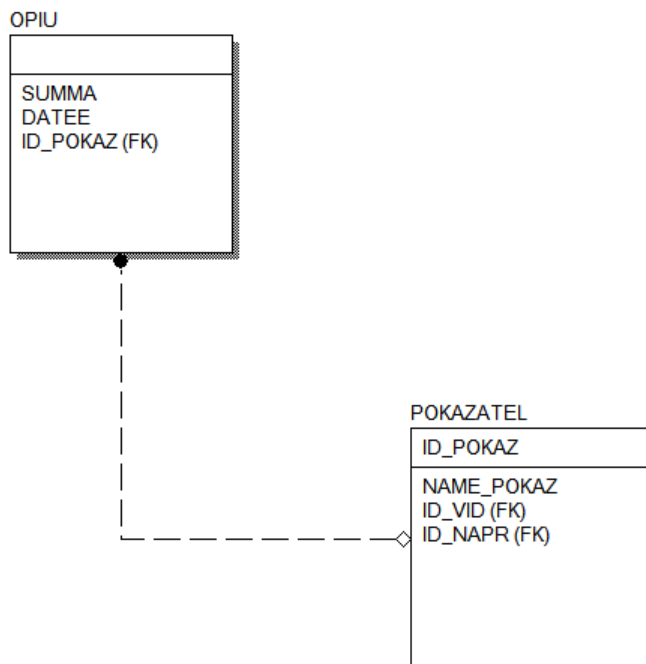


Рисунок 2.9 – Сегмент ER -диаграммы для бизнес-правила 8

9) Каждому значению, загружаемому в таблицу «SUPPORT» соответствует только один вид, одно направление и ID показателя. В то время как под это соответствие попадает множество значений в таблице «SUPPORT» (Рисунок 2.10).

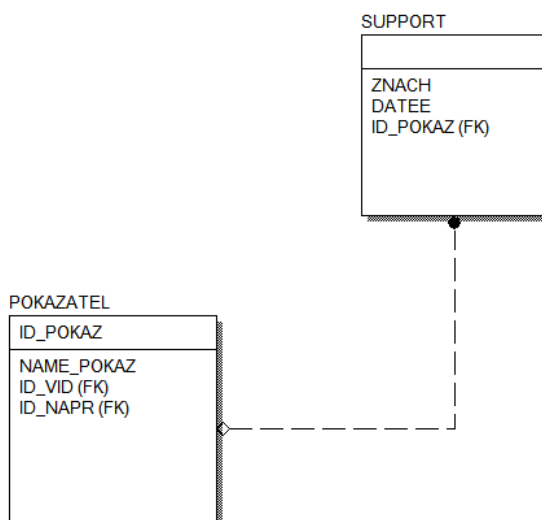


Рисунок 2.10 – Сегмент ER -диаграммы для бизнес-правила 9

10) Каждому рассчитываемому значению соответствует одно направление, вид и ID показателя. В то время как под это соответствие попадает множество рассчитываемых значений в таблице «CALC» (Рисунок 2.11).

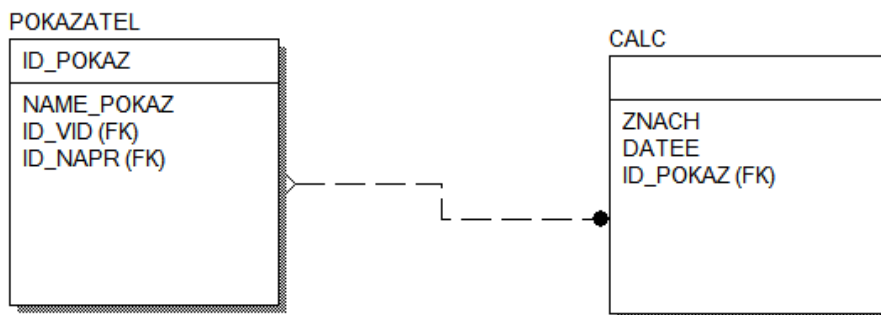


Рисунок 2.11 – Сегмент ER -диаграммы для бизнес - правила 10

2.3.3 Разработка и построение подробной ER-диаграммы на основании бизнес правил

Концептуальная модель аналитической информационной системы представлена на рисунке 2.12.

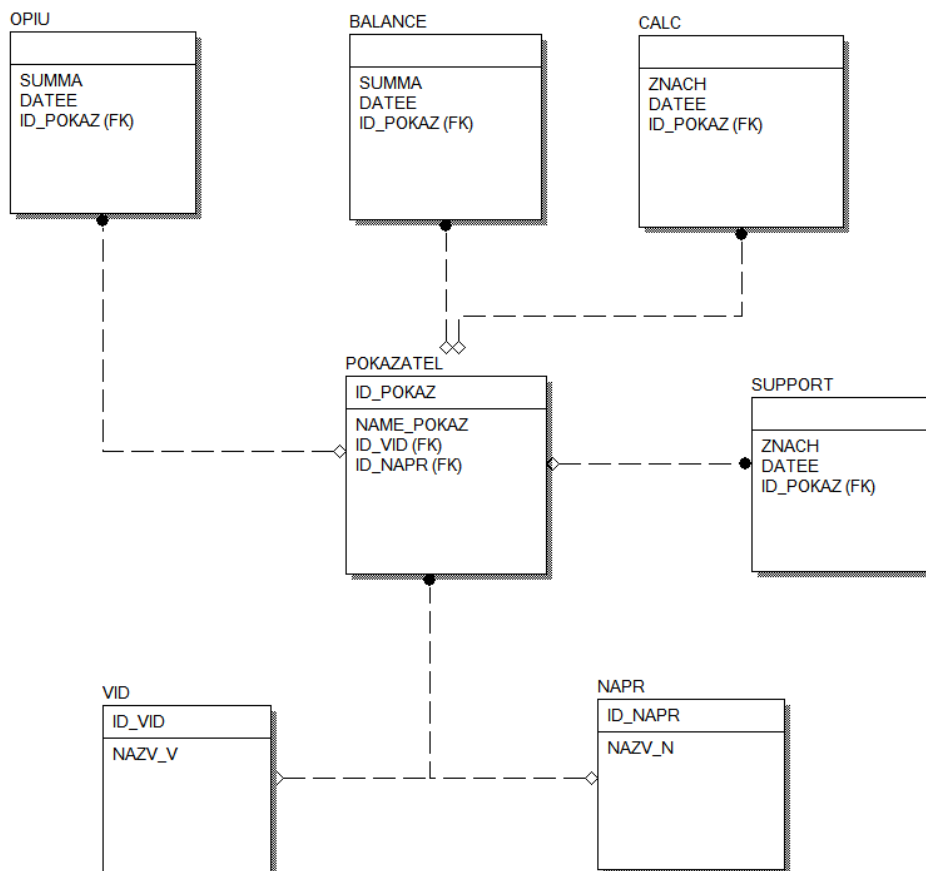


Рисунок 2.12 – Окончательный вариант концептуальной модели

2.3.4 Анализ информационных задач и круга пользователей системы

С данной ИС может работать только пользователь – аналитик.

При работе с системой пользователь должен иметь возможность решать следующие задачи:

- Загружать новые данные бухгалтерского баланса в систему.
- Загружать новые данные отчёта о прибылях и убытках в систему.
- Вносить в систему вспомогательную дополнительную информацию по организации.
- Производить расчёт всех необходимых показателей, параметров и коэффициентов.
- Производить вертикальный и горизонтальный анализ управленческой отчётности.
- Выгружать значения показателей по их видам в отчёты.
- Выгружать данные вертикального и горизонтального анализа управленческой отчётности в отчёты.
- Просматривать управленческую отчётность компании.
- Просматривать данные рассчитанных показателей по видам.
- Просматривать данные вертикального и горизонтального анализа управленческой отчётности.
- Совершать поиск информации по дате.

2.3.5 Определение атрибутов каждой сущности

Описание сущностей разработанной ER-модели и их атрибутов представлено в таблице 2.1.

Т а б л и ц а 2.1 – Определение атрибутов каждой сущности

Сегмент ER модели	Описание
NAPR ID_NAPR NAZV_N	Таблица NAPR (Направление) содержит следующие данные: ID_NAPR – идентификатор направления NAZV_N – название направления
VID ID_VID NAZV_V	Таблица VID (Вид) содержит следующие данные: ID_VID – идентификатор вида NAZV_V – название вида
POKAZATEL ID_POKAZ NAME_POKAZ ID_VID (FK) ID_NAPR (FK)	Таблица POKAZATEL (Показатели, содержащиеся в системе) содержит следующие данные: ID_POKAZ – идентификатор показателя NAME_POKAZ – название показателя ID_VID - идентификатор вида ID_NAPR - идентификатор направления

Сегмент ER модели	Описание			
<p>OPIU</p> <table border="1"> <tr> <td>SUMMA</td> </tr> <tr> <td>DATEE</td> </tr> <tr> <td>ID_POKAZ (FK)</td> </tr> </table>	SUMMA	DATEE	ID_POKAZ (FK)	<p>Таблица OPIU (Загружаемые данные отчёта о прибылях и убытках) содержит следующие данные:</p> <p>ID_POKAZ – идентификатор показателя</p> <p>SUMMA – сумма статьи отчёта</p> <p>DATEE – дата загрузки данных</p>
SUMMA				
DATEE				
ID_POKAZ (FK)				
<p>BALANCE</p> <table border="1"> <tr> <td>SUMMA</td> </tr> <tr> <td>DATEE</td> </tr> <tr> <td>ID_POKAZ (FK)</td> </tr> </table>	SUMMA	DATEE	ID_POKAZ (FK)	<p>Таблица BALANCE (Загружаемые данные отчёта баланса) содержит следующие данные:</p> <p>ID_POKAZ – идентификатор показателя</p> <p>SUMMA – сумма статьи отчёта</p> <p>DATEE – дата загрузки данных</p>
SUMMA				
DATEE				
ID_POKAZ (FK)				
<p>SUPPORT</p> <table border="1"> <tr> <td>ZNACH</td> </tr> <tr> <td>DATEE</td> </tr> <tr> <td>ID_POKAZ (FK)</td> </tr> </table>	ZNACH	DATEE	ID_POKAZ (FK)	<p>Таблица SUPPORT (Загружаемые дополнительные вспомогательные данные) содержит следующие данные:</p> <p>ID_POKAZ – идентификатор показателя</p> <p>ZNACH – значение показателя</p> <p>DATEE – дата загрузки данных</p>
ZNACH				
DATEE				
ID_POKAZ (FK)				
<p>CALC</p> <table border="1"> <tr> <td>ZNACH</td> </tr> <tr> <td>DATEE</td> </tr> <tr> <td>ID_POKAZ (FK)</td> </tr> </table>	ZNACH	DATEE	ID_POKAZ (FK)	<p>Таблица CALC (Расчитанные показатели) содержит следующие данные:</p> <p>ID_POKAZ – идентификатор показателя</p> <p>ZNACH – значение показателя</p> <p>DATEE – дата загрузки данных</p>
ZNACH				
DATEE				
ID_POKAZ (FK)				

2.4 Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть).

Отношения приведены в таблицах 2.2 – 2.8. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Обязательное поле для краткости обозначено not null, необязательное – null. Типы данных обозначаются так: number – числовой, varchar2 – символьный, date – дата.

Т а б л и ц а 2.2 – NAPR

Содержание поля	Имя поля	Тип, длина	Примечание
ID Направления	ID_NAPR	number(5)	Первичный ключ
Название направления	NAZV_N	varchar2(10)	Null

Т а б л и ц а 2.3 – VID

Содержание поля	Имя поля	Тип, длина	Примечание
ID Вида	ID_VID	number(5)	Первичный ключ
Название вида	NAZV_V	varchar2(80)	null

Т а б л и ц а 2.4 – POKAZATEL

Содержание поля	Имя поля	Тип, длина	Примечание
ID показателя	ID_POKAZ	number(5)	Первичный ключ
Содержание поля	Имя поля	Тип, длина	Примечание
ID показателя	ID_POKAZ	number(5)	Первичный ключ
Содержание поля	Имя поля	Тип, длина	Примечание

Т а б л и ц а 2.5 – OPIU

Содержание поля	Имя поля	Тип, длина	Примечание
ID показателя	ID_POKAZ	number(5)	Первичный ключ
Сумма статьи	SUMMA	number(12)	null
Дата загрузки	DATEE	date	null

Т а б л и ц а 2.6 – BALANCE

Содержание поля	Имя поля	Тип, длина	Примечание
ID показателя	ID_POKAZ	number(5)	Первичный ключ
Сумма статьи	SUMMA	number(12)	null
Дата загрузки	DATEE	date	null

Т а б л и ц а 2.7 – SUPPORT

Содержание поля	Имя поля	Тип, длина	Примечание
ID показателя	ID_POKAZ	number(5)	Первичный ключ
Значение показателя	ZNACH	number(10)	null
Дата загрузки	DATEE	date	null

Т а б л и ц а 2.8 – CALC

Содержание поля	Имя поля	Тип, длина	Примечание
ID показателя	ID_POKAZ	number(5)	Первичный ключ
Значение показателя	ZNACH	number(12)	null
Дата	DATEE	date	null

2.5 Расчет места для хранения ХД

Перед запуском ХД необходимо выставить необходимые размеры первоначальной ХД и ее роста.

На этом этапе, необходимо знать какой объем памяти будет занимать создаваемое хранилище данных. Объем внешней памяти, необходимый для функционирования системы, складывается из двух составляющих: память, занимаемая модулями СУБД (ядро, утилиты, вспомогательные программы), и память, отводимая под данные (МД). Наиболее существенным обычно является МД. Объем памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объемом самих данных, поэтому может не учитываться. В проекте рассчитывается предполагаемый максимальный объем памяти занимаемой ХД. Расчет физической памяти приводится в таблицах 2.9 – 2.15.

Т а б л и ц а 2.9 – Расчет физической памяти для таблицы NAPR

Имя поля	Тип, длина	Длина (байт)
ID_NAPR	number(5)	5
NAZV_N	varchar2(10)	10

Общая длина строки: 15 байт

Число строк: ~ 2

Общий объем требуемой памяти: ~ 30 байт

Т а б л и ц а 2.10 – Расчет физической памяти для таблицы VID

Имя поля	Тип, длина	Длина (байт)
ID_VID	number(5)	5
NAZV_V	varchar2(80)	80

Общая длина строки: 85 байт

Число строк: ~ 17

Общий объем требуемой памяти: ~ 1445 байт

Т а б л и ц а 2.11 – Расчет физической памяти для таблицы POKAZATEL

Имя поля	Тип, длина	Примечание
ID_POKAZ	number(5)	5
ID_NAPR	number(5)	5
ID_VID	number(5)	5
NAZV_V	varchar2(80)	80

Общая длина строки: 95 байт

Число строк: ~ 154

Общий объем требуемой памяти: ~ 14630 байт

Т а б л и ц а 2.12 – Расчет физической памяти для таблицы OPIU

Имя поля	Тип, длина	Примечание
ID_POKAZ	number(5)	5
SUMMA	number(12)	12
DATEE	date	8

Общая длина строки: 25 байт

Число строк: ~ 20 500 000

Общий объем требуемой памяти: ~ 512 500 000 байт

Т а б л и ц а 2.13 – Расчет физической памяти для таблицы BALANCE

Имя поля	Тип, длина	Примечание
ID_POKAZ	number(5)	5
SUMMA	number(12)	12
DATEE	date	8

Общая длина строки: 25 байт

Число строк: ~ 20 500 000

Общий объем требуемой памяти: ~ 512 500 000 байт

Т а б л и ц а 2.14 – Расчет физической памяти для таблицы SUPPORT

Имя поля	Тип, длина	Примечание
ID_POKAZ	number(5)	5
ZNACH	number(10)	10
DATEE	date	8

Общая длина строки: 23 байт

Число строк: ~ 100 000

Общий объем требуемой памяти: ~ 2 500 000 байт

Т а б л и ц а 2.15 – Расчет физической памяти для таблицы CALC

Имя поля	Тип, длина	Примечание
ID_POKAZ	number(5)	5
ZNACH	number(12)	12
DATEE	date	8

Общая длина строки: 25 байт

Число строк: ~ 40 500 000

Общий объем требуемой памяти: ~ 1 012 500 000 байт

Таким образом, из полученных данным можно рассчитать приблизительный максимальный объем хранилища данных. Объем будет равен сумме объемов всех таблиц:

$$M_d = 30 + 1445 + 14630 + 512500000 + 512500000 + 2500000 + 1012500000 = 1,9 \text{ (Гб)}.$$

Объём памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объёмом самих данных, поэтому может не учитываться. Требуемый объём оперативной памяти определяется на основании анализа интенсивности запросов и объёма результирующих данных.

Исходя из данных, полученных во время физического проектирования можно сделать вывод, что размер данных ХД за первые 3 года работы базы будет равен 1,9 Гб. Через шесть лет этот размер будет увеличен в 2 раза. Перед запуском ХД необходимо выставить необходимые размеры первоначальной ХД и ее роста.

2.6 Используемые языки программирования

2.6.1 Среда разработки Visual studio

Microsoft Visual Studio – линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Microsoft Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

2.6.2 Язык C#

Язык программирования C# был создан в конце 1990-х годов и стал частью общей .NET-стратегии Microsoft. Впервые он увидел свет в качестве б-версии в середине 2000 года. Главным архитектором C# был Андерс Хейлсберг (Anders Hejlsberg) – один из ведущих специалистов в области языков программирования, получивший признание во всем мире. Достаточно сказать, что в 1980-х он был автором весьма успешного продукта Turbo Pascal, изящная реализация которого установила стандарт для всех будущих компиляторов. C# непосредственно связан с C, C++ и Java. И это не случайно. Эти три языка -- самые популярные и самые любимые языки программирования в мире. Более того, почти все профессиональные программисты сегодня знают C и C++, и большинство знает Java. Поскольку C# построен на прочном, понятном фундаменте, то переход от этих "фундаментальных" языков к "надстройке" происходит без особых усилий со стороны программистов. Так как Андерс

Хейлсберг не собирался изобретать свое "колесо", он сосредоточился на введении усовершенствований и новшеств.

Свой синтаксис С# во многом унаследовал от С++ и Java. Разработчики, имеющие опыт написания приложений на этих языках, найдут в С# много знакомых черт. Но вместе с тем он является во многом новаторским - атрибуты, делегаты и события, прекрасно вписанные в общую идеологию языка, прочно заняли место в сердцах .NET - разработчиков. Их введение позволило применять принципиально новые приемы программирования. От С язык С# унаследовал синтаксис, многие ключевые слова и операторы. Кроме того, С# построен на улучшенной объектной модели, определенной в С++. С# и Java связаны между собой несколько сложнее. Как упоминалось выше, Java также является потомком С и С++. У него тоже общий с ними синтаксис и сходная объектная модель. Подобно Java С# предназначен для создания переносимого кода.

2.7 Расчётная часть

2.7.1 Логическое проектирование

Задача логического проектирования состоит в создании реляционных табличных структур на языке DDL.

Создание табличного пространства:

```
Create tablespace Diplom datafile
'D:\Anna\OEL5_7\BD\Diplom.dat' size 100M reuse autoextend on
next 5M maxsize 200M
/
create user Diplom identified by Diplom
default tablespace Diplom quota 40M on Diplom
temporary tablespace temp
```

В приложении А представлен полный код создания базы данных. Ниже представлен код создания таблиц:

```
create table NAPR(
ID_Napr number not null primary key,
Nazv_N varchar2(10));
/
create table VID(
ID_VID number not null primary key,
Nazv_V varchar2(80));
/
create table POKAZATEL(
ID_POKAZ number not null primary key,
ID_Napr number references NAPR (ID_Napr),
ID_VID number references VID (ID_VID),
Name_Pokaz varchar2(80));
```

```

/
create table BALANCE(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Summa number(12,0),
Datee date);
/
create table OPiU(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Summa number(12,0),
Datee date);
/
create table SUPPORT(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Znach number(10,2),
Datee date);
/
create table CALC(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Znach number(12,2),
Datee date);

```

2.7.2 Физическое проектирование

Обоснование выбора СУБД

Для разработки хранилища данных была выбрана ОРСУБД Oracle Database 11.2.0.3. Oracle Database 11g помогает заказчикам снизить затраты на информационные технологии и повысить качество услуг за счет консолидации в облака баз данных и готовые системы, например, Oracle Exadata и Oracle Database Appliance. Это быстрая, надежная, безопасная и легкая в управлении система, подходящая для выполнения всех задач, связанных с базами данных, в том числе с приложениями для предприятий, банками данных и анализом крупных объемов данных.

В результате выполнения созданных скриптов, получим реальное ХД. Диаграмма физической модели хранилища данных представлена на рисунке 2.13.

Меры по обеспечению безопасности

Одной из наиболее важных частей в БД является разработка прав доступа к ней, т.к. нужна защита от несанкционированного доступа и защита от доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа.

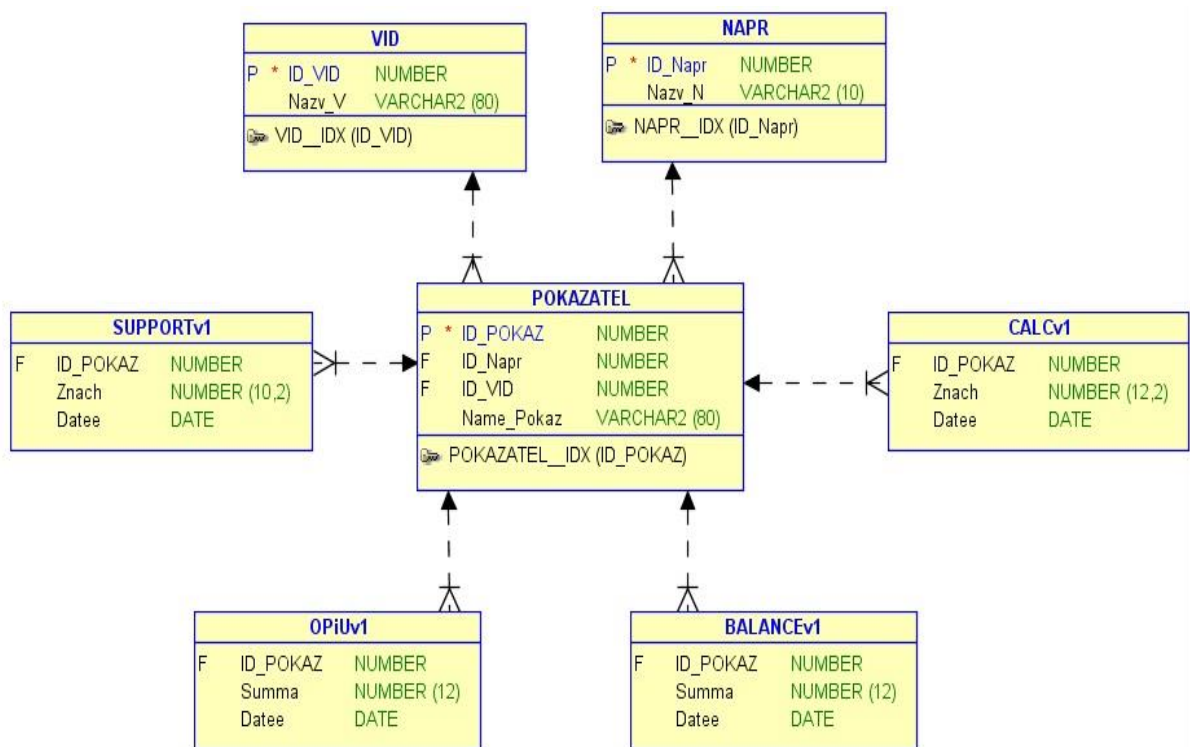


Рисунок 2.13 – Диаграмма физической модели хранилища данных

Создание пользователей, логинов и задание им паролей

Создание пользователя analytk:

```
create user analytk identified by analytk
default tablespace analytk quota 40M on analytk
temporary tablespace temp
```

Создание роли analytk:

```
create role analytk;
```

Присвоение пользователю analytk роль analytk:

```
grant create session to analytk;
/
grant select on NAPR to analytk;
/
grant select on VID to analytk;
/
grant select on POKAZATEL to analytk;
/
grant select on OPIU to analytk;
/
grant select on BALANCE to analytk;
/
grant select on SUPPORT to analytk;
/
```

```
grant select on CALC to analityk;
/
grant analityk to analityk;
```

Создание процедур, представлений и триггеров

Триггер на запрет ввода данных с одинаковыми значениями и датой загрузки в таблицу «BALANCE»:

```
Create Or Replace Trigger Balance
Before Update Or Insert Of ID_POKAZ On BALANCE
for each row
begin
if :NEW.ID_POKAZ <1110 or :NEW.ID_POKAZ>1700 then
raise_application_error (-20500,'Neverniy kod stroky!');
end if;
end;
```

Триггер на запрет ввода данных с одинаковыми значениями и датой загрузки в таблицу «BALANCE»:

```
Create Or Replace Trigger OPIU
Before Update Or Insert Of ID_POKAZ On OPIU
for each row
begin
if :NEW.ID_POKAZ <10 or :NEW.ID_POKAZ>300 then
raise_application_error (-20500,'Neverniy kod stroky!');
end if;
end;
```

Процедура группировки статей актива бухгалтерского баланса. Результаты работы представлены на рисунке 2.14:

```
Create or replace procedure AP (v_datte in date:='&p_datte')
is
A1 number;
A2 number;
A3 number;
A4 number;
invest number;
zdaniya number;
deng number;
Nezavstr number;
DZ_strah number;
DZ_perestrah number;
DZ_mengod number;
DZ_bolgod number;
Zapasy number;
Dr_aktivy number;
NMA number;
Dolya_v_perestrez number;
```

```

Osn_sred number;
begin
Select Summa into invest from Balance where ID_POKAZ=1120 and
DATEE=v_datte;
Select Summa into zdaniya from Balance where ID_POKAZ=1122 and
DATEE=v_datte;
Select Summa into deng from Balance where ID_POKAZ=1260 and
DATEE=v_datte;
Select Summa into Nezavstr from Balance where ID_POKAZ=1240
and DATEE=v_datte;
Select Summa into DZ_strah from Balance where ID_POKAZ=1190
and DATEE=v_datte;
Select Summa into DZ_perestrah from Balance where
ID_POKAZ=1200 and DATEE=v_datte;
Select Summa into DZ_mengod from Balance where ID_POKAZ=1220
and DATEE=v_datte;
Select Summa into Zapasy from Balance where ID_POKAZ=1250 and
DATEE=v_datte;
Select Summa into Dr_aktivy from Balance where ID_POKAZ=1270
and DATEE=v_datte;
Select Summa into NMA from Balance where ID_POKAZ=1110 and
DATEE=v_datte;
Select Summa into Dolya_v_perestrez from Balance where
ID_POKAZ=1160 and DATEE=v_datte;
Select Summa into Osn_sred from Balance where ID_POKAZ=1230
and DATEE=v_datte;
Select Summa into DZ_bolgod from Balance where ID_POKAZ=1210
and DATEE=v_datte;
A1:=invest-ZDANIYA+DENG+NEZAVSTR;
A2:=DZ_STRAH+DZ_PERESTRAH+DZ_MENGOD;
A3:=ZAPASY+DR_AKTIVY;
A4:=NMA+ZDANIYA+NEZAVSTR+DOLYA_V_PERESTREZ+DZ_STRAH+DZ_bolgod+
OSN_SRED;
Insert into CALC values (3000,A1,v_datte);
Insert into CALC values (3001,A2,v_datte);
Insert into CALC values (3002,A3,v_datte);
Insert into CALC values (3003,A4,v_datte);
commit;
end AP;

```

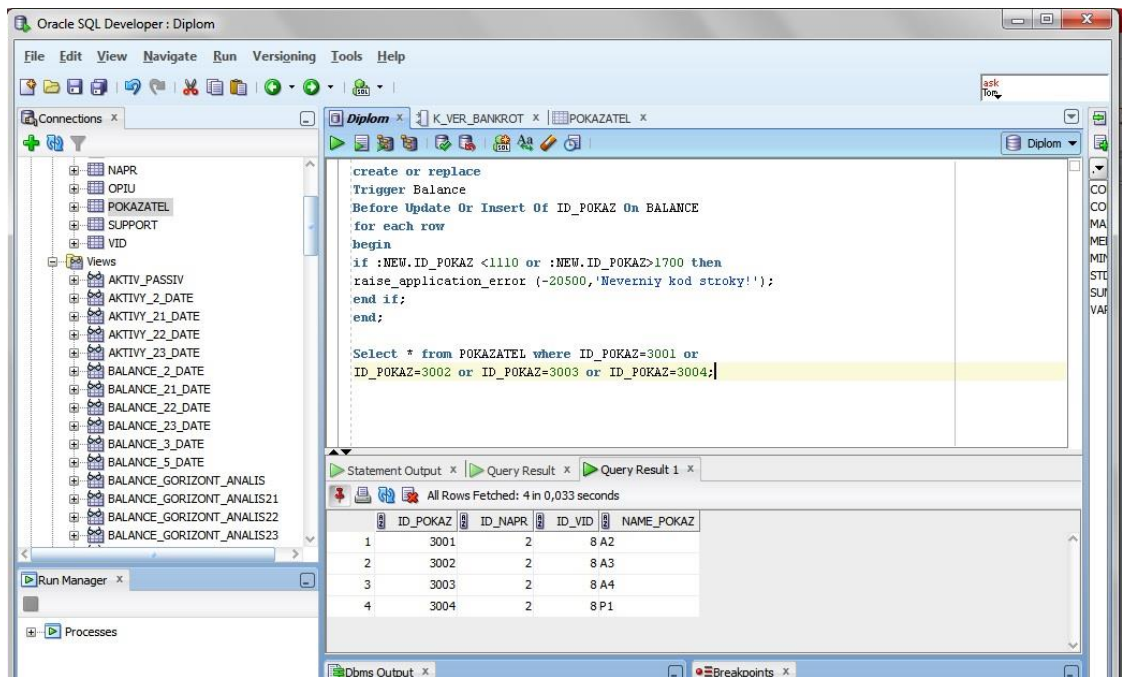



Рисунок 2.14 – работа процедуры группировки активов компании

Процедура расчета коэффициентов вероятности банкротства:

```

Create or replace procedure K_ver_bankrot (v_datte in
date:='&p_datte')
is
K_sootn_sob_zajem number;
K_tekush_likv number;
K_obesp_sobst_sr number;
Z_schet number;
v_1 number;
v_2 number;
v_3 number;
v_4 number;
v_5 number;
v_6 number;
v_7 number;
v_8 number;
v_9 number;
v_10 number;
v_11 number;
v_12 number;
v_13 number;
v_14 number;
v_15 number;
v_16 number;
v_17 number;
v_18 number;
v_19 number;
v_20 number;
v_21 number;

```

```

v_22 number;
v_23 number;
v_24 number;
v_25 number;
v_26 number;
v_27 number;
begin
SELECT SUMMA into v_1 from BALANCE where ID_POKAZ=1490 and
DATEE=v_datte;
Select SUMMA into v_2 from BALANCE where ID_POKAZ=1665 and
DATEE=v_datte;
Select SUMMA into v_3 from BALANCE where ID_POKAZ=1675 and
DATEE=v_datte;
Select SUMMA into v_4 from BALANCE where ID_POKAZ=1680 and
DATEE=v_datte;
Select SUMMA into v_5 from BALANCE where ID_POKAZ=1681 and
DATEE=v_datte;
Select SUMMA into v_6 from BALANCE where ID_POKAZ=1300 and
DATEE=v_datte;
SELECT SUMMA into v_7 from BALANCE where ID_POKAZ=1190 and
DATEE=v_datte;
Select SUMMA into v_8 from BALANCE where ID_POKAZ=1200 and
DATEE=v_datte;
Select SUMMA into v_9 from BALANCE where ID_POKAZ=1210 and
DATEE=v_datte;
Select SUMMA into v_10 from BALANCE where ID_POKAZ=1220 and
DATEE=v_datte;
Select SUMMA into v_11 from BALANCE where ID_POKAZ=1240 and
DATEE=v_datte;
Select SUMMA into v_12 from BALANCE where ID_POKAZ=1250 and
DATEE=v_datte;
Select SUMMA into v_13 from BALANCE where ID_POKAZ=1260 and
DATEE=v_datte;
Select SUMMA into v_14 from BALANCE where ID_POKAZ=1270 and
DATEE=v_datte;
Select SUMMA into v_15 from BALANCE where ID_POKAZ=1640 and
DATEE=v_datte;
Select SUMMA into v_16 from BALANCE where ID_POKAZ=1650 and
DATEE=v_datte;
Select SUMMA into v_17 from BALANCE where ID_POKAZ=1660 and
DATEE=v_datte;
Select SUMMA into v_18 from BALANCE where ID_POKAZ=1620 and
DATEE=v_datte;
Select SUMMA into v_19 from BALANCE where ID_POKAZ=1630 and
DATEE=v_datte;
SELECT SUMMA into v_20 from BALANCE where ID_POKAZ=1110 and
DATEE=v_datte;
Select SUMMA into v_21 from BALANCE where ID_POKAZ=1120 and
DATEE=v_datte;
Select SUMMA into v_22 from BALANCE where ID_POKAZ=1230 and
DATEE=v_datte;

```

```

SELECT SUMMA into v_23 from BALANCE where ID_POKAZ=1470 and
DATEE=v_datte;
Select SUMMA into v_24 from OPIU where ID_POKAZ=250 and
DATEE=v_datte;
Select SUMMA into v_25 from OPIU where ID_POKAZ=80 and
DATEE=v_datte;
Select SUMMA into v_26 from OPIU where ID_POKAZ=10 and
DATEE=v_datte;
Select SUMMA into v_27 from BALANCE where ID_POKAZ=1410 and
DATEE=v_datte;
K_sootn_sob_zaem:=ROUND((v_1+v_2+v_3+v_4+v_5)/(v_6-
(v_1+v_2+v_3+v_4+v_5)),3);
K_tekush_likv:=ROUND((v_7+v_8+v_9+v_10+v_11+v_12+v_13+v_14)/(v
_15+v_16+v_17+v_18+v_19),3);
K_obesp_sobst_sr:=ROUND((v_1+v_2+v_3+v_4+v_5-v_20-v_21-
v_22)/(v_7+v_8+v_9+v_10+v_11+v_12+v_13+v_14),3);
Z_schet:=ROUND((1.2*(v_7+v_8+v_9+v_10+v_11+v_12+v_13+v_14)+1.4
*v_23+3.3*v_24+0.99*(v_25+v_26)+0.6*v_27)/(v_6-v_23*v_6),3);
Insert into CALC values (3011,K_sootn_sob_zaem,v_datte);
Insert into CALC values (3012,K_tekush_likv,v_datte);
Insert into CALC values (3013,K_obesp_sobst_sr,v_datte);
Insert into CALC values (3014,Z_schet,v_datte);
commit;
end K_ver_bankrot;

```

Процедура расчета показателей финансовой устойчивости:

```

Create or replace procedure Fin_ustoych (v_datte in
date:='&p_datte')
is
K_avtonomyy number;
K_fin_zavisim number;
K_fin_leveridj number;
K_manevr_sobstv_sr number;
K_rent_sobstv_sr number;
SK number;
VB number;
SR number;
Ob number;
DZP number;
Z number;
DS number;
begin
Select SUMMA into SK from BALANCE where ID_POKAZ=1490 and
DATEE=v_datte;
Select SUMMA into VB from BALANCE where ID_POKAZ=1700 and
DATEE=v_datte;
Select SUMMA into SR from BALANCE where ID_POKAZ=1590 and
DATEE=v_datte;
Select SUMMA into Ob from BALANCE where ID_POKAZ=1690 and
DATEE=v_datte;

```

```

Select SUMMA into DZP from BALANCE where ID_POKAZ=1220 and
DATEE=v_datte;
Select SUMMA into Z from BALANCE where ID_POKAZ=1250 and
DATEE=v_datte;
Select SUMMA into DS from BALANCE where ID_POKAZ=1260 and
DATEE=v_datte;
K_avtonomy:=ROUND((SK/VB),3);
K_fin_zavisim:=ROUND(((SR+Ob)/VB),3);
K_fin_leveridj:=ROUND((SR/SK),3);
K_manevr_sobstv_sr:=ROUND(((DZP+Z+DS)/SK),3);
Insert into CALC values (3023,K_avtonomy,v_datte);
Insert into CALC values (3024,K_fin_zavisim,v_datte);
Insert into CALC values (3025,K_fin_leveridj,v_datte);
Insert into CALC values (3026,K_manevr_sobstv_sr,v_datte);
commit;
end Fin_ustoych;

```

Процедуры расчета параметров оценки резервов перестрахования:

```

Create or replace procedure Reservy_Perestrakh (v_datte in
date:='&p_datte')
is
Dost_strakh_rezerv number;
Ud_ves_prem_v_perest number;
Uch_perest_v_sost_ubytk number;
Uch_perest_v_strakh_rez number;
SR number;
v_67 number;
v_68 number;
v_69 number;
v_70 number;
v_71 number;
v_72 number;
v_73 number;
v_74 number;
v_75 number;
v_76 number;
v_77 number;
v_78 number;
v_79 number;
v_80 number;
v_81 number;
v_82 number;
begin
Select SUMMA into SR from BALANCE where ID_POKAZ=1590 and
DATEE=v_datte;
Select SUMMA into v_67 from BALANCE where ID_POKAZ=1160 and
DATEE=v_datte;
Select SUMMA into v_68 from BALANCE where ID_POKAZ=1170 and
DATEE=v_datte;
Select SUMMA into v_69 from BALANCE where ID_POKAZ=1180 and
DATEE=v_datte;

```

```

Select SUMMA into v_70 from OPIU where ID_POKAZ=10 and
DATEE=v_datte;
Select SUMMA into v_71 from OPIU where ID_POKAZ=80 and
DATEE=v_datte;
Select SUMMA into v_72 from OPIU where ID_POKAZ=50 and
DATEE=v_datte;
Select SUMMA into v_73 from OPIU where ID_POKAZ=160 and
DATEE=v_datte;
Select SUMMA into v_74 from OPIU where ID_POKAZ=12 and
DATEE=v_datte;
Select SUMMA into v_75 from OPIU where ID_POKAZ=82 and
DATEE=v_datte;
Select SUMMA into v_76 from OPIU where ID_POKAZ=80 and
DATEE=v_datte;
Select SUMMA into v_77 from OPIU where ID_POKAZ=10 and
DATEE=v_datte;
Select SUMMA into v_78 from OPIU where ID_POKAZ=112 and
DATEE=v_datte;
Select SUMMA into v_79 from OPIU where ID_POKAZ=100 and
DATEE=v_datte;
Select SUMMA into v_80 from BALANCE where ID_POKAZ=1510 and
DATEE=v_datte;
Select SUMMA into v_81 from BALANCE where ID_POKAZ=1520 and
DATEE=v_datte;
Select SUMMA into v_82 from BALANCE where ID_POKAZ=1530 and
DATEE=v_datte;
Dost_strakh_rezerv:=ROUND(((SR-(v_67+v_68+v_69))/(v_70+v_71-
(v_72+v_73))),3);
Ud_ves_prem_v_perest:=ROUND(((v_74+v_75)/(v_77+v_76)),3);
Uch_perest_v_sost_ubytk:=ROUND(((v_67+v_68+v_69)/v_79),3);
Uch_perest_v_strakh_rez:=ROUND(((v_67+v_68+v_69)/(v_80+v_81+v_
82)),3);
Insert into CALC values (3049,Dost_strakh_rezerv,v_datte);
Insert into CALC values (3050,Ud_ves_prem_v_perest,v_datte);
Insert into CALC values
(3051,Uch_perest_v_sost_ubytk,v_datte);
Insert into CALC values
(3052,Uch_perest_v_strakh_rez,v_datte);
commit;
end Reservy_Perestrakh;

```

Представление, отражающее оценку страховых резервов (Рисунок 2.15):

```

Create view Otsenka_strakh_rez_Bugalter as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.14', sm, Null)) "01.07.14",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '01.07.13', sm, Null)) "01.07.13"
from

```

```
(Select CC.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P,CALC CC
where P.ID_VID=13 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc) GROUP BY NAME_POKAZ ;
```

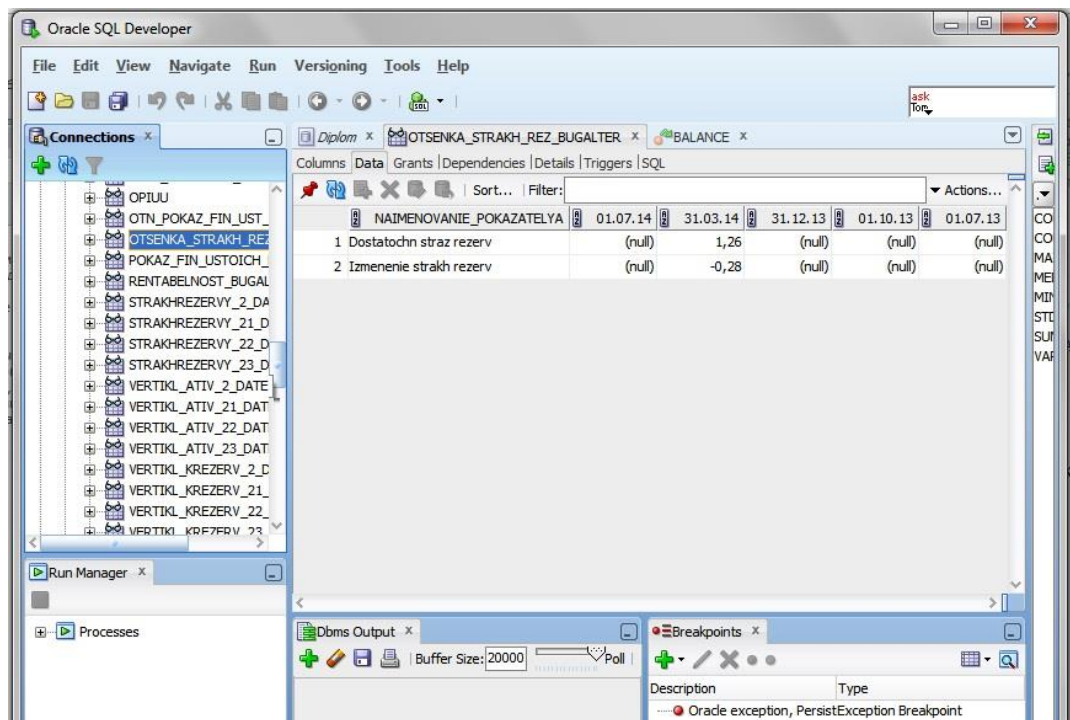


Рисунок 2.15 – Оценка страховых резервов

Представление, отражающее зависимость компании от перестраховщиков
(Рисунок 2.16):

```
Create view Zavisim_ot_perestrakh_Bugalter as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.14', sm, Null)) "01.07.14",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '01.07.13', sm, Null)) "01.07.13"
from
(Select CC.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P,CALC CC
where P.ID_VID=14 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc) GROUP BY NAME_POKAZ ;
```

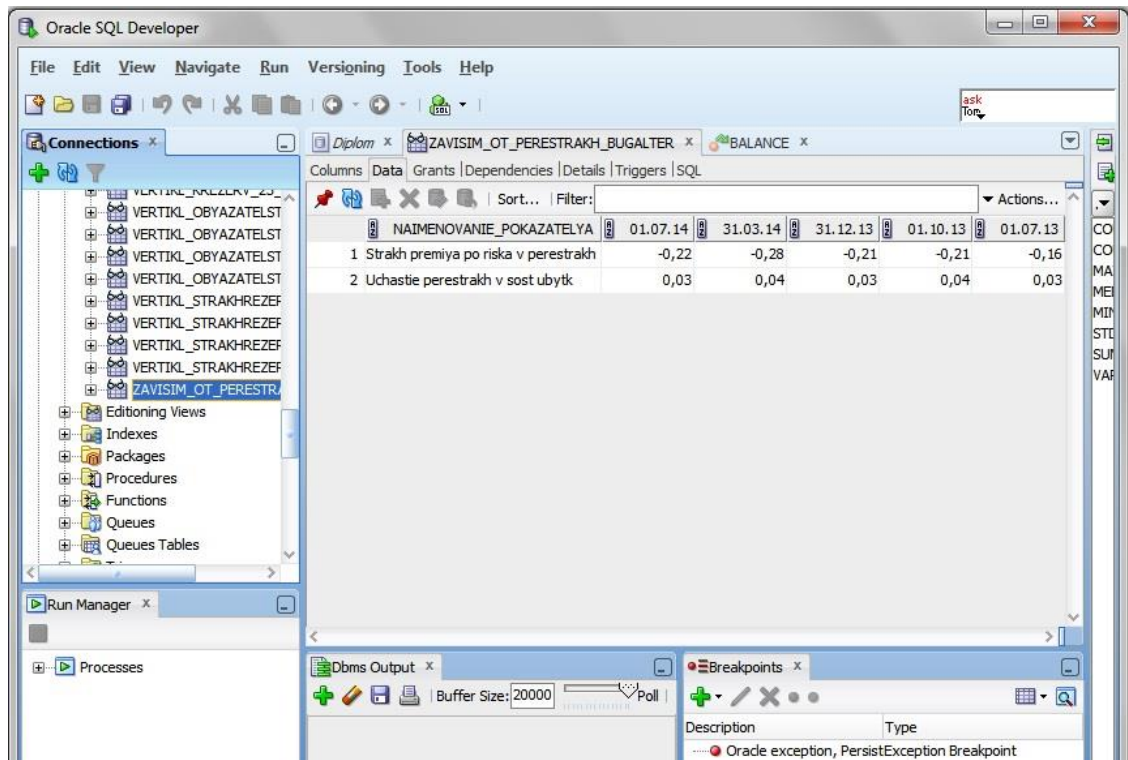


Рисунок 2.16 – Зависимость компании от перестраховщиков

Два взаимосвязанных представления отображающие горизонтальный анализ данных за два периода (Рисунок 2.17):

```

Create or replace view Balance_2_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '01.07.14', sm, Null)) "01.07.14"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where (P.ID_VID=1 or P.ID_VID=2 or P.ID_VID=3 or P.ID_VID=4)
and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ Asc)
GROUP BY NAME_POKAZ ;
/

Create or replace view Balance_Gorizont_analis as
Select
Naimenovanie_pokazatelya, "31.03.14",
"01.07.14", "01.07.14"-"31.03.14" Absolut_otklonen,
Round((( "01.07.14"-"31.03.14") *100/"31.03.14"), 2)
Procent_otklonen from BALANCE_2_DATE Where "31.03.14"<>0;

```

NAIMENOVANIE_POKAZATELYA	31.03.14	01.07.14	ABSOLUT_OTKLONEN	PROCENT_OTKLONEN
1 Nematerialnye aktivy	200	185	-15	-7,5
2 Proch DZ<12	195	160	-35	-17,95
3 Osnovnye sredstva	6141	6041	-100	-1,63
4 Rezervnyy kapital	542	500	-42	-7,75
5 Rezervy nezarabotannye premii	8806	8800	-6	-0,07
6 ITOGO strakhovye rezervy	12999	13109	110	0,85
7 ITOGO kapital i rezervy	7973	8003	30	0,38
8 Neraspribyly proshlykh let	3273	3300	27	0,82
9 Rezervy ubytkov	2600	2300	-300	-11,54
10 Drugie strakhovye rezervy	1600	1300	-300	-18,75
11 Proch obyazatelstva	2104	2304	200	9,51
12 ITOGO aktivy	28520	27920	-600	-2,1
13 Balans aktivy	28520	27920	-600	-2,1
14 Investitsii v tom chisle:	5457	5607	150	2,75
15 Dolya perestrakh-v v rezerv ubytkov	63	73	10	15,87
16 Zapasy	1386	1586	200	14,43
17 Inye aktivy	686	706	20	2,92
18 DZ po operats perestrakh	15	13	-2	-13,33
19 KZ po operats perestrakh	10	12	2	20
20 Dolya pestrakh-v v RNP	368	268	-100	-27,17
21 Dolya pestrakh-v v TSP	1205	1300	95	7,88

Рисунок 2.17 – Горизонтальный анализ данных за два периода

Представление, отображающее вертикальный анализ бухгалтерского баланса за два периода (Рисунок 2.18):

```

Create or replace view Vertikal_Ativ_22_date as
Select      Naimenovanie_pokazatelya,      "01.10.13",
ROUND(("01.10.13"*100/(Select  "01.10.13"
from AKTIVY_22_DATE where  Naimenovanie_pokazatelya='ITOGO
aktivy')),2)      Udeln_ves_aktiva,      "31.12.13",
ROUND(("31.12.13"*100/(Select  "31.12.13"
from AKTIVY_22_DATE where  Naimenovanie_pokazatelya='ITOGO
aktivy')),2) Udeln_ves_aktiva_1 FROM AKTIVY_22_DATE;

```


NAIMENOVANIE_POKAZATELYA	31.03.14	01.07.14	ABSOLUT_OTKLONEN	PROCENT_OTKLONEN
1 Nematerialnye aktivy	200	185	-15	-7,5
2 Proch DZ < 12	195	160	-35	-17,95
3 Osnovnye sredstva	6141	6041	-100	-1,63
4 Rezervnyy kapital	542	500	-42	-7,75
5 Rezervy nezarabotannye premii	8806	8800	-6	-0,07
6 ITOGO strakhovye rezervy	12999	13109	110	0,85
7 ITOGO kapital i rezervy	7973	8003	30	0,38
8 Nerazspr pribyl proshlykh let	3273	3300	27	0,82
9 Rezervy ubytkov	2600	2300	-300	-11,54
10 Drugie strakhovye rezervy	1600	1300	-300	-18,75
11 Proch obyazatelstva	2104	2304	200	9,51
12 ITOGO aktivy	28520	27920	-600	-2,1
13 Balans aktivy	28520	27920	-600	-2,1
14 Investitsii v tom chisle:	5457	5607	150	2,75
15 Dolya perestrakh-v v rezerv ubytkov	63	73	10	15,87
16 Zapasy	1386	1586	200	14,43
17 Inye aktivy	686	706	20	2,92
18 DZ po operats perestrakh	15	13	-2	-13,33
19 KZ po operats perestrakh	10	12	2	20
20 Dolya pestrakh-v v RNP	368	268	-100	-27,17
21 Dopolnitelnye sredstva	1200	1200	0	0,00

Рисунок 2.18 – Вертикальный анализ бухгалтерского баланса

Два взаимосвязанных представления, отображающие горизонтальный анализ отчёта о прибылях и убытках (Рисунок 2.19):

```

Create or replace view OPIU_2_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '01.07.14', sm, Null)) "01.07.14"
from
(Select  OP.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, OP.SUMMA sm
from POKAZATEL P,OPIU OP
where (P.ID_VID=5 or P.ID_VID=6 or P.ID_VID=7) and
P.ID_POKAZ=OP.ID_POKAZ order by P.ID_POKAZ desc)
GROUP BY NAME_POKAZ,ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view OPIU_Gorizont_analis as
Select
Naimenovanie_pokazatelya, "31.03.14",
"01.07.14","01.07.14"- "31.03.14" Absolut_otklonen,
Round((( "01.07.14"- "31.03.14") *100/"31.03.14"),2)
Procent_otklonen from OPIU_2_date Where "31.03.14"<>0;

```

NAIMENOVANIE_POKAZATELYA	31.03.14	01.07.14	ABSOLUT_OTKLOMEN	PROCENT_OTKLOMEN
1 Strakh premii - netto perestrakh	992	1000	8	0,81
2 strakhovye premii	992	1000	8	0,81
3 Dokhody po investitsiyam	90	100	10	11,11
4 Oplach ubytki - netto perestrakh	-50	-40	10	-20
5 Izmen rez po strakh zh - netto perestrakh	-954	-944	10	-1,05
6 Rez oper po strakh zhizni	78	80	2	2,56
7 Strakh premii - netto perestrakh	5752	5730	-22	-0,38
8 strakhovye premii	7654	7634	-20	-0,26
9 peredannye perestrahovshchikam	-1902	-1900	2	-0,11
10 Izmen rez nezarab pr - netto perestrakh	1262	1202	-60	-4,75
11 izmen rez nezarab pr - vsego	1597	1600	3	0,19
12 izmen doli perestrakh v rezerve	-335	-305	30	-8,96
13 Sost ubytki - netto perestrakh	-1542	-1552	-10	0,65
14 Oplach ubytki - netto perestrakh	-1338	-1388	-50	3,74
15 oplach ubytki - vsego	-1385	-1300	85	-6,14
16 dolya perestrahovshchikov	47	50	3	6,38
17 Izmen rez ubytkov - netto perestrakh	-204	-214	-10	4,9
18 izmen rezervov ubytkov - vsego	-207	-217	-10	4,83
19 izm doli perestrakh-v v rezervakh	3	5	2	66,67
20 Otschisl v rez predopr-kh meropr	2	(null)	(null)	(null)
21 Dokhody na ustoychivost perestrakh	2142	2142	0	0,00

Рисунок 2.19 – Горизонтальный анализ бухгалтерского баланса за два периода

2.8 Описание интерфейса приложения и его функциональности

Данное приложение разработано с учетом интуитивно понятного интерфейса, а так же с наиболее удобным функционалом. Листинг программы представлен в приложении В.

Приложение дает право пользователю загружать данные в хранилище, производить расчет параметров, проводить анализ имеющихся данных и выводить информацию в качестве отчётов. В случае неверного пароля или логина выходит сообщение. На рисунке 2.20 представлена форма входа в систему.

Рисунок 2.20 – Форма входа в систему

Работа аналитика с систему начинается с авторизации. С приложением может работать 1 вид пользователя: пользователь.

После авторизации сотруднику откроется основная форма программы. В неё сотрудник может начать такую работу как: загрузка новых данных, удаление старых, перейти к работе с данными отчётности или расчетными данными. Данная форма представлена на рисунке 2.21.

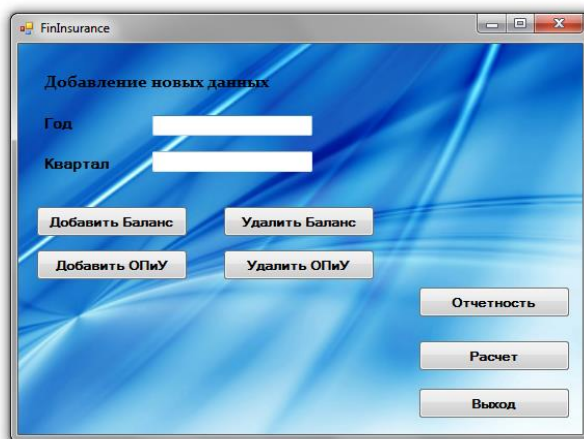


Рисунок 2.14 – Форма первоначальной работы с системой

Перейдя в следующую форму, мы видим, как пользователь может работать с данными управленческой отчётности. Он может выбрать год и квартал, когда была загружен баланс или отчёт о прибылях и убытках и посмотреть их. Также он может посмотреть сводный баланс и отчёт о прибылях и убытках. А также произведя различные виды анализа просмотреть полученные данные кнопкой «Обзор данных», предварительно отметив галочкой нужный отчёт. Отметив несколько видов отчётов можно произвести выгрузку данных с помощью кнопки «Выгрузить» (Рисунок 2.22).

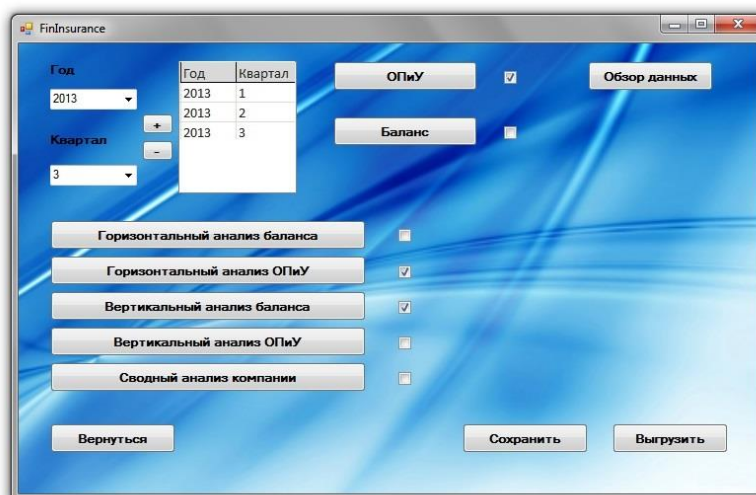


Рисунок 2.22 – Форма работы с управленческой отчётностью

Вернувшись обратно с помощью кнопки «Вернуться» пользователь попадает снова в первую форму, а оттуда может перейти непосредственно к расчёту параметров загруженных показателей. Предназначенная для этого форма представлена на рисунке 2.23. В форме работы с расчетами параметров и показателей сотрудник может также выбрать необходимые год и квартал, а затем произвести расчет и группировку всех показателей. Также он может просмотреть необходимую для этого вспомогательную информацию или добавить её.

The screenshot shows the 'FinInsurance' application window. It features a blue-themed interface with several interactive elements:

- Year Selection:** A dropdown menu set to '2013'.
- Quarter Selection:** A dropdown menu set to '2', with '+' and '-' buttons for navigation.
- Table:** A small table with columns 'Год' (Year) and 'Квартал' (Quarter). It contains three rows: (2013, 1), (2013, 2), and (2013, 3).
- Buttons:** 'Обзор данных' (Data Overview), 'Вспомогательные данные' (Auxiliary data), and 'Добавить вспомогательные данные' (Add auxiliary data).
- Checkboxes for Indicators:**
 - Вероятность банкротства (Probability of bankruptcy)
 - Кoeffициенты ликвидности (Liquidity coefficients)
 - Показатели финансовой устойчивости (Financial stability indicators)
 - Относительные показатели фин. устойч. (Relative financial stability indicators)
 - Достаточность страховых резервов (Adequacy of insurance reserves)
 - Зависимость от перестраховщиков (Dependence on reinsurers)
 - Показатели рентабельности (Profitability indicators)
 - Деловая активность страховой (Insurance business activity)
 - Группировка АП (AP grouping)
- Bottom Buttons:** 'Вернуться' (Return), 'Сохранить' (Save), and 'Выгрузить' (Export).

Рисунок 2.23 – Форма работы с расчетами параметров и показателей

В конце работы можно просмотреть выбранные данные, предварительно пометив галочкой нужный отчёт. Выбранные отчёты можно также получить из системы с помощью кнопки «Выгрузить».

Выбрав нужные года, отметив галочкой необходимый отчёт, пользователь может просмотреть данные. Форма просмотра информации представлена на рисунке 2.24.

NAIMENOVANIE POKAZATELYA	01.07.2014	31.03.2014	31.12.2013	01.10.2013	01.07.2014
Nematerialnye aktivy	185	200	115	194	113
Proch DZ<12	160	195	130	195	133
Osnovnye sredstva	6041	6141	6341	6199	6343
Rezervnyy kapital	500	542	530	590	500
Rezervy nezarabotan premii	8800	8806	8300	8096	8330
ITOGO strakhovye rezervy	13109	12999	13309	1299	13339
Kratkosr zaimy i kredity	30		30	30	10
Dohody budush periodov	0		0	0	0
Proch DZ >12	0	0	0	0	0
ITOGO kapital i rezervy	8003	7973	3003	759	3033
Rezervy po strakh zhizni	0	0	0	0	0
Zaimy i kredity >12	0	0	0	0	0
Rezervy predupredit meropriyat	0	0	0	0	0
Depo premiy u perestrahovateley	0	0	0	0	0
Neraspr pribyl proshlykh let	3300	3273	3310	3093	3330

Рисунок 2.24 – Форма просмотра и поиска данных

В форме представления данных возможен поиск информации по году, месяцу, дате, а также ключевым словам (рисунки 2.25, 2.26).

NAIMENOVANIE POKAZATELYA	01.10.2013
Nematerialnye aktivy	113
Proch DZ<12	133
Osnovnye sredstva	6343
Rezervnyy kapital	500
Rezervy nezarabotan premii	8330
ITOGO strakhovye rezervy	13339
Kratkosr zaimy i kredity	10
Dohody budush periodov	0
Proch DZ >12	0
ITOGO kapital i rezervy	3033
Rezervy po strakh zhizni	0
Zaimy i kredity >12	0
Rezervy predupredit meropriyat	0
Depo premiy u perestrahovateley	0
Neraspr pribyl proshlykh let	3330

Рисунок 2.2515 – Поиск данных бухгалтерского баланса по дате

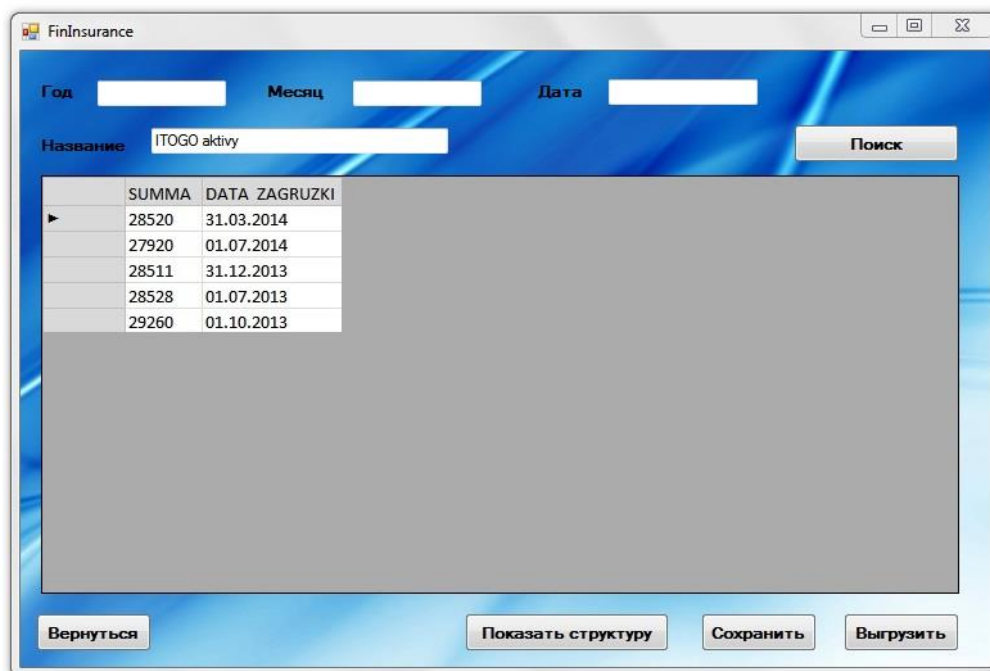


Рисунок 2.166 – Поиск данных бухгалтерского баланса по показателю

На рисунках 2.27–2.30 представлены формы, содержащие различную информацию, хранящуюся в системе

FinInsurance

Год Месяц Дата

Название

NAIMENOVANIE POKAZATELYA	01.07.2014	31.03.2014	31.12.2013	01.10.2013	01.07.2014
▶ peredannye perestrahovshchikam	-1900	-1902	-1950	-1350	-1250
Izmen rez po strakh zh - netto peres	-944	-954	-964	-974	-904
Izm rez nezarab pr - netto perestrah	1202	1262	1252	1752	1252
oplach ubytki - vsego	-1300	-1385	-1350	-1350	-1050
Vnerealiz dokhod krome doots fin vlozh	0				
Nalog na pribyl i inye obyaz plat	-500	-487	-550	-570	-270
Dokhody po investitsiyam	100	90	110	91	81
Sost ubytki - netto perestrah	-1552	-1542	-1502	-1902	-2002
Oplach ubytki - netto perestrah	-1388	-1338	-1308	-1508	-2008
dolya perestrahovshchikov	50	47	30	35	35
Raskhody po investitsiyam	0				
Operats raskh krome svyaz s invest	-230	-229	-250	-230	-220
Pribyl (ubytok) do nalogooblozh	1800	1811	1850	1950	2050
Strakh premii - netto perestrah	1000	992	1010	910	810
izmen rez nezarab pr - vsego	1600	1597	1630	1330	1230

Рисунок 2.177 – Данные отчёта о прибылях и убытках

Год Месяц Дата

Название

NAIMENOVANIE POKAZATELYA	01.07.2014	31.03.2014	31.12.2013	01.10.2013	01.07.2013
▶ Koeff urovnya dokhod po invest		0,05			
Koeff rent strakh deyat	0,45	0,57	0,42	0,45	0,41
Koeff rentab kapital	0,16	0,17	0,45	1,92	0,68
Koeff raskhodov		-0,47			
Koeff eff invest deyat		-0,17			
Koeff doli perestrakh		-0,28			
Koeff ubytochn	-0,24	-0,24	-0,23	-0,28	-0,3
Koeff rentab		0,18			

Рисунок 2.188 – Данные о состоянии рентабельности компании за последние пять лет

Год Месяц Дата

Название

DATA ZAGRUZKY	A1	A2	A3	A4	P1	P2	P3	P4
▶ 01.07.2014	18707	173	2292	6226	5964	13122	20	8
31.12.2013	18607	140	1692	6456	5665	13319	30	3
01.10.2013	18357	211	1982	6375	5242	1225	20	
31.03.2014	18512	195	1872	6210	5504	12899	20	7
01.07.2013	18630	146	1683	6456	5342	13352	10	3

Рисунок 2.199 – Группировка активов и пассивов организации, сравнение по годам

NAIMENOVANIE POKAZATELYA	31.03.2014	01.07.2014	ABSOLUTNOE OTKLOНENIE	%
Strakh premii - netto perestrakh	992	1000	8	0,81
strakhovye premii	992	1000	8	0,81
Dokhody po investitsiyam	90	100	10	11,11
Oplach ubytki - netto perestrakh	-50	-40	10	-20
Izmen rez po strakh zh - netto perestrakh	-954	-944	10	-1,05
Rez oper po strakh zhizni	78	80	2	2,56
Strakh premii - netto perestrakh	5752	5730	-22	-0,38
strakhovye premii	7654	7634	-20	-0,26
peredannye perestrakhovshchikam	-1902	-1900	2	-0,11
Izm rez nezarab pr - netto perestrakh	1262	1202	-60	-4,75
izmen rez nezarak pr - vsego	1597	1600	3	0,19
izmen doli perestrakh v rezerve	-335	-305	30	-8,96
Sost ubytki - netto perestrakh	-1542	-1552	-10	0,65
Oplach ubytki - netto perestrakh	-1338	-1388	-50	3,74
oplach ubytki - vsego	-1385	-1300	85	-6,14

Рисунок 2.3020 – Горизонтальный анализ бухгалтерского баланса компании

При просмотре данных предоставляется возможность ознакомиться со структурой отчёта, нажав кнопку «Показать структуру» (Рисунок 2.31)

STATYA BALANCE	NAPRAVLENIE	VID	NAZVANIE POKAZATELYA
1110	Balance	Aktivny	Nematerialnye aktivy
1120	Balance	Aktivny	Investitsii v tom chisle:
1122	Balance	Aktivny	zdaniya
1130	Balance	Aktivny	Finansovye vlozheniya
1150	Balance	Aktivny	Depo premiy u perestrakhovateley
1160	Balance	Aktivny	Dolya perestrakh-v v rezerv po strakh zhizni
1170	Balance	Aktivny	Dolya pestrakh-v v RNP
1180	Balance	Aktivny	Dolya perestrakh-v v rezerv ubytkov
1190	Balance	Aktivny	DZ po operats strakh
1200	Balance	Aktivny	DZ po operats perestrakh
1210	Balance	Aktivny	Proch DZ >12
1220	Balance	Aktivny	Proch DZ <12
1224	Balance	Aktivny	Zadolzh uch-v po vzosam v UK
1230	Balance	Aktivny	Osnovnye sredstva
1240	Balance	Aktivny	Nezaversh stroitelstvo
1250	Balance	Aktivny	Zapasy
1290	Balance	Aktivny	ITOGO aktivy

Рисунок 2.3121 – Форма структуры отчёты

Технико-экономическое обоснование проекта

2.9 Описание работы и обоснование необходимости

Тема данного дипломного проекта - «Автоматизированная информационная система анализа финансового состояния страховой компании».

Цель данного проекта - разработать систему, для автоматизации работы финансового аналитика с управленческой отчётностью, позволяющую на основании внесенных в систему данных проводить анализ финансового состояния страховой компании, рассчитывать показатели, параметры, коэффициенты и мультипликаторы для оценки финансового состояния организации и стоимости акций компаний с целью принятия решений о размещении капитала самостоятельно.

В настоящее время разработка и внедрение таких систем в экономике в целом практикуется не часто, а особенно в области страхования. Это обусловлено большой ценой на аналитические системы, сложностью внедрения, недостаточной технической обеспеченностью современных компаний. К тому же в область страхования является специфической областью поэтому для разработки программных продуктов в данной сфере необходимо привлечение высококвалифицированных разработчиков в сфере IT, а также консультантов по экономическому и финансовому направлению. Всё это затрудняет и замедляет процесс усовершенствования оснащённости техническими средствами страховых компаний и экономической отрасли в общем виде. На сегодняшний день пройден лишь этап первоначальной автоматизации организаций, но не разрабатываются серьёзные системы, которые могли бы проводить детальный анализ финансового состояния компании, и её дальнейшего развития. До настоящего времени разработкой такого продукта в специфике страховой компании не занимались, он является совершенно новым, отвечающим всем современным стандартам.

В данном разделе приводится рассмотрение экономической составляющей реализации данного продукта, отражающее временные, трудовые и финансовые затраты на проект.

2.10 Трудовые ресурсы, используемые в работе

Разработка многомодульного программного продукта – сложный и трудоемкий процесс, требующий наряду с интеллектуальными, техническими затратами и финансовыми затратами. Поэтому необходимым является произведение расчета стоимости этой разработки. Затраты на разработку данного программного комплекса определяется по формуле

$$C = \Phi OT + C_H + A + Э + C_{np} + H \quad (3.1)$$

где ФОТ – фонд оплаты труда;

C_n – социальный налог;

A – амортизационные отчисления;

Э – затраты на электроэнергию;

$C_{пр}$ - прочие расходы;

H – накладные расходы.

Необходимый фонд оплаты труда рассчитывается по формуле

$$ФОТ = Z_{осн} + Z_{доп} \quad (3.2)$$

где $Z_{осн}$ - основная заработная плата;

$Z_{доп}$ - дополнительная заработная плата.

В процессе работы задействованы:

1. Руководитель – постановка задачи, разработка основных бизнес правил для работы проекта;

2. Инженер-разработчик – разработка сайта офис регистратора, проектирование базы данных, сопровождение системы;

3. Консультанты по экономической части и по части ОБЖД.

Общее количество сотрудников и их заработная плата представлены в таблице 3.1.

Т а б л и ц а 3.1 – Количество задействованных в проекте работников, и их заработная плата

Исполнитель	Количество, чел	Месячная заработная плата, тенге
Руководитель	1	80000
Консультант по части “Экономика”	1	75000
Консультант по части “Безопасность жизнедеятельности”	1	60000
Инженер-разработчик	1	125000
Итого	4	340000

Труд сотрудников института, задействованных в работе, оплачивается согласно положению о заработной плате АУЭС, труд программиста-разработчика принят условно, на договорной основе в размере 120000 тенге.

Базовый показатель для определения составляющих затрат труда вычисляется по формуле

$$Q = q * c \quad (3.3)$$

где Q – условное число команд;

$q=4900$ – коэффициент, учитывающий условное число команд в зависимости от типа задачи;

$c=1,26$ – коэффициент, учитывающий новизну и сложность программы.

$$Q = 4900 * 1,26 = 6174.$$

Выбор значения коэффициента q был произведен из таблицы 3.2.

Т а б л и ц а 3.2 – Значения коэффициента q

Тип задачи	Пределы изменений коэффициента
Задачи учета	От 1400 до 1500
Задачи оперативного управления	От 1500 до 1700
Задачи планирования	от 3000 до 3500
Много вариантыные задачи	от 4500 до 5000
Комплексные задачи	от 5000 до 5500

По степени новизны программные продукты могут быть отнесены к одной из 4-х групп:

- группа А - разработка принципиально новых задач;
- группа Б - разработка оригинальных программ;
- группа В - разработка программ с использованием типовых решений.
- группа Г - разовая типовая задача.

Выбор коэффициента c производится из таблицы 3.3, на пересечении групп сложности и степени новизны.

Т а б л и ц а 3.3 – Коэффициенты расчета трудоемкости

Язык программ.	Группа сложности	Степень новизны				Коэффициент В
		А	Б	В	Г	
Высокого уровня	1	1,38	1,26	1,15	0,69	1,2
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,5
Низкого уровня	1	1,58	1,45	1,32	0,79	1,2
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,5

При разработке многомодульного программного продукта используется техническое задание, согласно которому выполнение работ происходит последовательно по заданным пунктам, с учетом сроков их исполнения. График выполнения работ по разработке программного продукта представлен в таблице 3.4.

Таблица 3.4 – График выполнения работ по разработке программного продукта

Код работы	Наименование работы	Ожидаемая длительность, дни	Обозначение
1	Подготовка описания задачи	2	ТПЗ
2	Описание задачи	3	ТОЗ
3	Разработка алгоритма	5	ТА
4	Установка и запуск локального сервера	3	ТЛС
5	Проектирование и создание базы данных	6	ТБД
6	Разработка модулей приложения	6	ТМ
7	Разработка и создание интерфейса системы	5	ТИН
8	Разработка основной части программы	6	ТОСН
9	Тестирование базы данных и основных модулей с целью выявления ошибок	2	ТТЕСТ
10	Составление технической документации	3	ТТЕХ
11	Подготовка раздела “Экономика”	9	ТЭ
12	Подготовка раздела “Безопасность жизнедеятельности”	9	ТБЖ

Так как участники, задействованные в проекте, работают в различные промежутки времени, в течение которого реализуется проект, необходимо произвести расчет дневной и почасовой оплаты труда.

Заработная плата каждого работника за один рабочий день рассчитывается по формуле

$$D = \frac{O}{n} \quad (3.3)$$

где O – оклад работника в тенге;

n – количество дней в рабочем месяце (это 24 дня – шестидневная рабочая неделя):

– для руководителя:

$$D = \frac{80000}{24} = 3333 \text{ тенге/день};$$

– для консультанта по части “Экономика”:

$$D = \frac{75000}{24} = 3125 \text{ тенге/день};$$

– для консультанта по части “БЖД”:

$$D = \frac{60000}{24} = 2500 \text{ тенге/день};$$

– для инженера разработчика:

$$D = \frac{125000}{24} = 5200 \text{ тенге/день}.$$

Заработная плата за один час рассчитывается по формуле

$$H = \frac{D}{z} \quad (3.4)$$

где D – заработная плата работника за один рабочий день,

z – количество часов рабочего дня (8 часов):

– для руководителя:

$$H = \frac{3333}{8} = 416,6 \text{ тенге/час};$$

– для консультанта по части “Экономика”:

$$H = \frac{3125}{8} = 390,6 \text{ тенге/час};$$

– для консультанта по части “БЖД”:

$$H = \frac{2500}{8} = 312,5 \text{ тенге/час};$$

– для инженера разработчика:

$$H = \frac{5200}{8} = 650 \text{ тенге/час}.$$

Время рассчитывается в человеко-часах, причем $T_{ПЗ}$ берется по фактически отработанному времени, а время остальных этапов определяется расчётно, по условному числу команд Q .

Определяем время, затраченное на каждый этап создания программного продукта:

1. $T_{ПЗ}$ (время на подготовку описания задачи), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов): $T_{ПЗ} = 32 \text{ чел/час}$

2. $T_{ОЗ}$ (время на описание задачи) определяется по формуле

$$T_{O3} = \frac{Q * B}{50 * K} \quad (3.5)$$

где $B=1,2$ – коэффициент учета изменений задачи, коэффициент B в зависимости от сложности задачи и числа изменений выбирается в интервале от 1,2 до 1,5 (таблица 3.5);

$K=1,2$ – коэффициент, учитывающий квалификацию программиста.

$$T_{O3} = \frac{6174 * 1,5}{50 * 1,2} = 148,2 \text{ чел/час}$$

Т а б л и ц а 3 . 5 – Коэффициенты квалификации программиста

Опыт работы	Коэффициент квалификации
До двух лет	0.8
2-3 года	1
3-5 лет	1.1 – 1.2
5-7 лет	1.3 – 1.4
более 7 лет	1.5 – 1.6

3. T_A (время на разработку алгоритма) рассчитываем по формуле

$$T_A = \frac{Q}{50 * K} \quad (3.6)$$

$$T_A = \frac{6174}{50 * 1,2} = 123,5 \text{ чел/час}$$

4. $T_{ЛС}$ (время на установку и запуск локального сервера) определяется аналогично T_A

$$T_{ЛС} = 123,5 \text{ чел/час}$$

5. $T_{БД}$ (время на проектирование и создание БД) определяется по формуле

$$T_{БД} = \frac{3500 * 1,5}{50 * 1,2} = 87,5 \text{ чел/час}$$

6. T_M (время на разработку модулей приложения) определяется по формуле

$$T_M = \frac{3500 * 1,5}{50 * 1,2} = 87,5 \text{ чел/час}$$

7. $T_{ИН}$ (время отладки и тестирования программы) определяется по формуле

$$T_{ИН} = \frac{6174 \cdot 1,5}{50 \cdot 1,2} = 154,4 \text{ чел/час}$$

8. $T_{ОСН}$ (время на разработку основной части), определяется по формуле

$$T_{ОСН} = \frac{4500 \cdot 1,5}{50 \cdot 1,2} = 112,5 \text{ чел/час}$$

9. $T_{ТЕСТ}$ (время на тестирование), определяется по формуле

$$T_{ТЕСТ} = \frac{3000 \cdot 1,5}{50 \cdot 1,2} = 75 \text{ чел/час}$$

10. $T_{ТЕХ}$ (время на составление технической документации), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_{ТЕХ} = 30 \text{ чел/час}$$

11. $T_{Э}, T_{БЖ}$ (время на подготовку разделов Экономика и БЖ), берется по факту и составляет (от 7-и до 10-ти дней по 8 часов)

$$T_{Э} = T_{БЖ} = 60 \text{ чел/час}$$

12. т.к. в проекте задействованы руководители, заработная плата которых отличается, то необходимо рассчитать время $T_{РВК}$. Время берется среднее и оно равно

$$T_{Р} = 80 \text{ чел/час};$$

$$T_{Э} = T_{БЖ} = 22 \text{ чел/час.}$$

Суммарные затраты труда рассчитываются как сумма составных затрат труда по формуле

$$T_{СТ} = T_{ИВ} + T_{ОЗ} + T_{А} + T_{ЛС} + T_{БД} + T_{М} + T_{ИН} + T_{ОСН} + T_{ТЕСТ} + T_{ТЕХ} + T_{Э} + T_{БЖ} \quad (3.7)$$

$$T_{СТ} = 32 + 148,2 + 123,5 + 123,5 + 87,5 + 87,5 + 154,4 + 112,5 + 75 + 30 + 60 + 60 = 1094 \text{ чел/час}$$

Таким образом, суммарная основная заработная плата составит

$$Z_{OCH} = Z_{ИНДЖ} + Z_{РУК} \quad (3.8)$$

$$Z_{OCH} = 1094 * 650 + 22 * 312,5 + 22 * 390,6 + 80 * 416,6 = 759896 \text{ тенге}$$

Дополнительная заработная плата в среднем определяется в размере 10% от основной заработной платы и рассчитывается по формуле

$$Z_{ДОП} = Z_{OCH} * 10\% \quad (3.9)$$

$$Z_{ДОП} = 759896 * 10\% = 75990 \text{ тенге}$$

Общий фонд оплаты труда согласно формуле 3.2

$$\Phi OT = 759896 + 75990 = 835886 \text{ тенге};$$

Социальный налог составляет 11% от ФОТ и рассчитывается по формуле

$$C_H = (\Phi OT - \text{ПО}) * 11\% \quad (3.10)$$

где ПО (пенсионные отчисления) составляют 10% от ФОТ и рассчитываются по формуле

$$\text{ПО} = \Phi OT * 10\% \quad (3.11)$$

Размер пенсионных отчислений согласно формуле 3.11 составляет

$$\text{ПО} = 835886 * 10\% = 83589 \text{ тенге}$$

Отчисления по социальному налогу согласно формуле 3.10

$$C_H = (835886 - 83589) * 11\% = 82752,7 \text{ тенге}$$

2.11 Расчет затрат на амортизацию

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются, в процентах к балансовой стоимости оборудования и рассчитываются по формуле

$$A = \frac{C_{обор} * H_A * N}{100 * 12 * t} \quad (3.12)$$

где H_A – норма амортизации;

$C_{обор}$ – первоначальная стоимость оборудования;

N – количество дней на выполнение работ;
 t – общее время использования персонального компьютера.
 Норма амортизации H_A , рассчитывается по формуле

$$H_A = \frac{C_{ОБОР} - C_{ЛИКВ}}{T_{НОРМ} * C_{ОБОР}} * 100\% \quad (3.13)$$

где $C_{ЛИКВ}$ – ликвидационная стоимость, составляет 5.61% от стоимости оборудования;

$T_{НОРМ}$ – нормативный срок службы (для персонального компьютера – 4 года).

Используемое оборудование представлено в таблице 3.6.

Т а б л и ц а 3 . 6 – Оборудование, используемое в разработке

Наименование	Модель	Стоимость,тг
Процессор	HP P6-2419ER D7B84EA	132 910
Монитор	Samsung S22C150NS	23 660
Клавиатура	A4tech KRS-8572	1 810
Компьютерная мышь	HP H2C21AA	860
Модем	D-link DSL-2500	1 810
Итого		161050

Так как ликвидационная стоимость составляет 5.61%, следовательно

$$C_{ЛИКВ} = 0,056 * C_{ОБОР};$$

$$C_{ЛИКВ} = 0,056 * 161050 = 9034,9 \text{ тг.}$$

Общее время использования персонального компьютера учитывает лишь время работы на компьютере и рассчитывается по формуле

$$t = T_{ЛС} + T_{БД} + T_M + T_{ИН} + T_{ОСН} + T_{ТЕСТ} \quad (3.14)$$

$$t = 123,5 + 87,5 + 87,5 + 154,4 + 112,5 + 75 = 640,4 \text{ чел/час} = 80 \text{ дней};$$

$$H_A = \frac{161050 - 9034,9}{4 * 161050} = 23,5 \%;$$

$$A = \frac{5387,8}{100 * 12 * 80} = 5387,8 \text{ тенге.}$$

Затраты на электроэнергию вычисляется по формуле

$$\mathcal{E} = M * k_z * T * C_{кВт\cdotч} \quad (3.15)$$

где M – мощность ЭВМ (600 Вт=0,6 кВт);

k_z – коэффициент загрузки (0.8);

$C_{кВт-ч}$ – 14,935 тг– стоимость 1 кВт-час электроэнергии;

T – время работы.

$$\mathcal{E} = 0,6 * 0,8 * 14,935 * 640,4 = 4591 \text{ тенге}$$

Расходы на материалы и комплектующие, используемые в процессе написания программного продукта ($C_{мик}$), а также затраты на техническое обслуживание и ремонт ($C_{ТО}$) составляют, соответственно, 2,06% и 2,266% от стоимости оборудования – формулы

$$C_{мик} = 0,0206 * C_{обор} \quad (3.16)$$

$$C_{то} = 0,02266 * C_{обор} \quad (3.17)$$

$$C_{мик} = 0,0206 * 161050 = 3318 \text{ тенге,}$$

$$C_{ТО} = 0,02266 * 161050 = 3649 \text{ тенге.}$$

$$C_{пр} = C_{мик} * C_{то} \quad (3.18)$$

$$C_{ПР} = 3318 + 3649 = 6967 \text{ тенге}$$

Накладные расходы, связанные с управлением и обслуживанием, содержанием и эксплуатацией оборудования и прочими дополнительными затратами на обеспечение процессов производства и обращения, составляют 50% от всех затрат, вычисляются по формуле

$$H = (\Phi OT + C_H + A + \mathcal{E} + C_{пр}) * 0,5 \quad (3.19)$$

$$H = (835886 + 82752,7 + 5387,8 + 4591 + 6967) = 467792,3 \text{ тенге}$$

Таким образом, затраты на разработку данного программного комплекса согласно формуле 3.1 равны:

$$C = 835886 + 82752,7 + 5387,8 + 4591 + 6967 + 467792,3 = 1396409,8 \text{ тенге}$$

Сводные результаты расчета себестоимости программного продукта представлены в таблице 3.7 и на рисунке 3.1, с указанием статьи расходов, суммы и их доли в общей стоимости разработки:

Т а б л и ц а 3 . 7 – Результирующая таблица себестоимости

Статья расходов	Сумма, д.ед.	В процентах от общей суммы,%
ФОТ	835886	60
СН	82752,7	6
А	5387,8	0
Э	4591	0
Спр	6967	1
Н	467792,3	33
Итого:	1396409,8	100

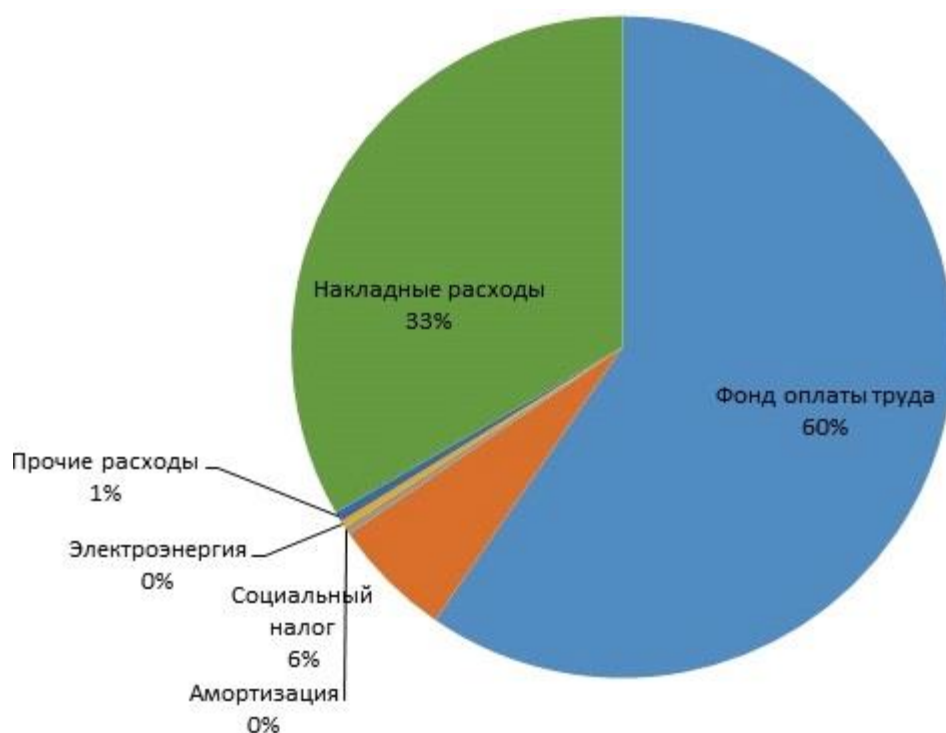


Рисунок 3.1 – Структура себестоимости программного обеспечения

2.12 Цена программного продукта

Минимальная цена программного продукта рассчитывается по следующей формуле

$$C_{min} = (C_{пп} + C_{мд} + C_3)(1 + P/100) \quad (3.20)$$

где $C_{пп}$, д.ед – себестоимость разработки программного продукта;

$C_{мд}$, д.ед – цена магнитного диска;

C_3 , д.ед – стоимость;

P ,% – планируемый процент рентабельности по отношению к себестоимости.

Тогда по формуле 3.20

$$C_{\min} = (1396409,8 + 193,8 + 142,8) * (1 + 0,1938) = 1671052 \text{ тг.}$$

Далее определяется цена реализации с учётом налога на добавленную стоимость (НДС)

$$C_p = C_{\min} + \text{НДС} \quad (3.21)$$

Ставка НДС на 2013 год, согласно статьи 268 Налогового кодекса РК составляет 12 %

$$C_p = 1671052 + 200526 = 1871578 \text{ тг.}$$

Таким образом, сравнивая стоимость другого программного обеспечения, которое является таким же по сложности создания и реализации, с учётом тех же показателей трудоёмкости, можно сделать вывод, что цена разрабатываемой мной автоматизированной системы полностью отвечает всем требованиям, вполне адекватна, не преувеличена и не занижена. К примеру, стоимость только лицензии на продукт Oracle составляет 1800000тг. Учитывая техническую поддержку продукта, а также разработку приложения цена может быть увеличена в половину, то есть доходить до 3000000тг. При таких условиях цена 1871578тг является приемлемой для рынка программного обеспечения. Себестоимость данного продукта сформирована в основном за счёт фонда оплаты труда и накладных расходов, что составляет 835886тг. И 467792,3тг. В процентном соотношении представляет 60% и 33% соответственно. Затраты на электроэнергию, амортизацию, а также прочие затраты влияют на стоимость разработки незначительно.

3 Технические решения безопасности труда

3.1 Анализ потенциально опасных и вредных производственных факторов проектируемого объекта, воздействующих на персонал

Опасные и вредные производственные факторы по природе возникновения делятся на следующие группы:

- физические;
- химические;
- психофизиологические;
- биологические.

Создание дипломной работы «Автоматизированная информационная система анализа финансового состояния страховой компании» проводится в помещении (кабинете) без повышенной опасности, относящиеся к специальной категории. Кабинет сухой, температура в нем поддерживалась на уровне 20°C – 25°C с влажностью в пределах 60%. Условия в кабинете соответствуют ГОСТу: нет сырости и токопроводящей пыли.

Данная работа представляет собой программу для страховой компании. Поэтому, для работы с этой программой, требуется персональный компьютер.

К химически опасным факторам постоянно действующим на программиста относятся следующие: возникновение, в результате ионизации воздуха при работе компьютера, активных частиц.

Биологические вредные производственные факторы в данном помещении отсутствуют.

К психологически вредным факторам, воздействующим на оператора в течение его рабочей смены можно отнести следующие:

- нервно-эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора.

Далее более подробно рассмотрены опасные и вредные факторы, воздействующие на программиста, возникшие в связи с разработкой данной системы.

3.2 Анализ помещения

Анализ помещения:

- Рабочее помещение находится на 9 этаже.
- Тип аудитории: офис.
- Помещение представляет собой две соединенные комнату, размеры которой 5x10x3 (ширина-длина).
- Источники света: 4 люминесцентные лампы.
- Одно окно (2x4).

- Окраска кабинета и мебели по стандарту должна способствовать созданию благоприятных условий для зрительного восприятия.
- Для защиты от избыточной яркости с окон могут быть применены жалюзи.
- В помещении сидят 6 человек. Работают в дневную смену. Режим работы с 8:20 до 18:00.
- Схема помещения приведена на рисунке 4.1.

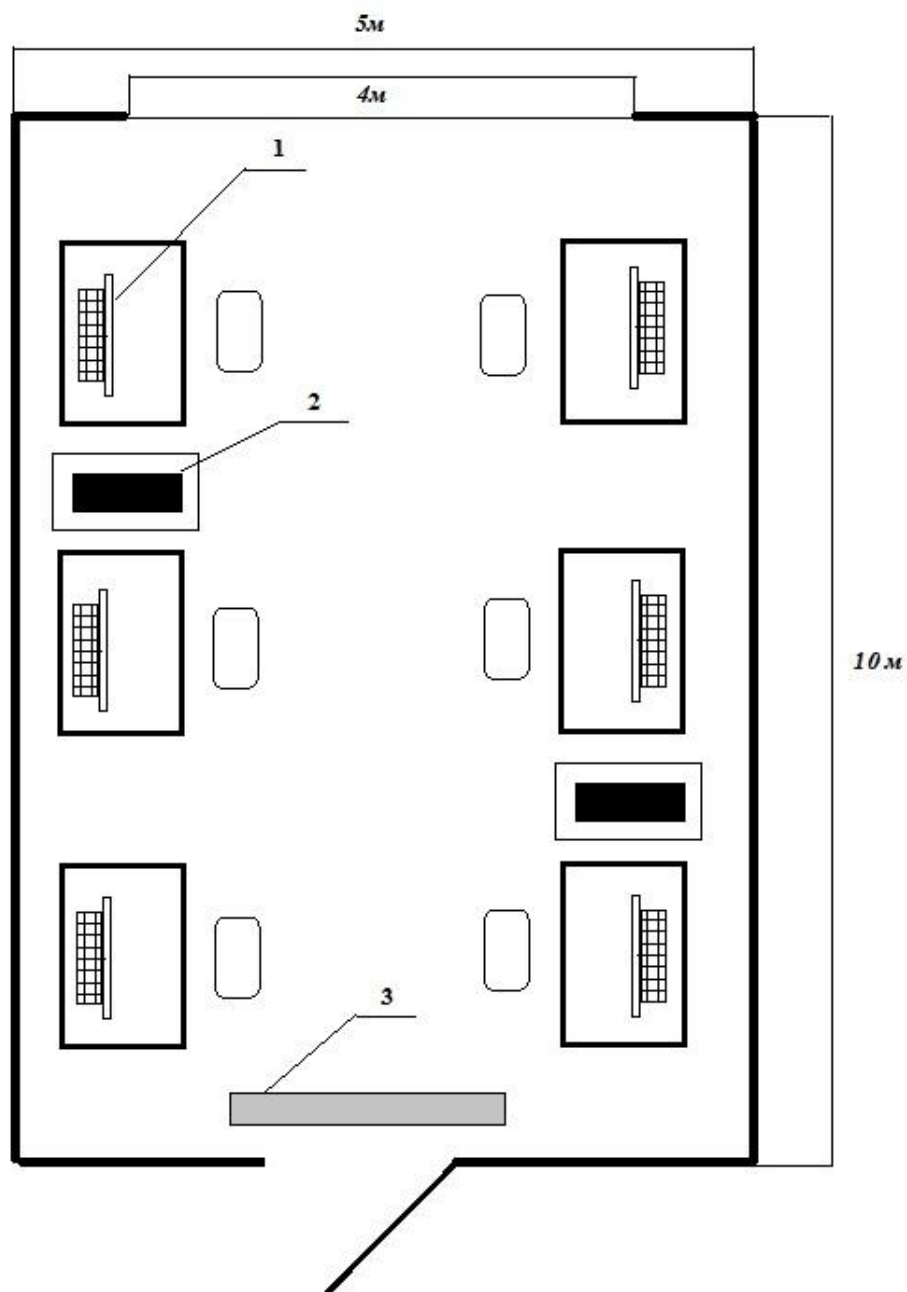


Рисунок 4.1 – План помещения
1 – ПК, 2 – принтер/сканер, 3 – кондиционер.

3.3 Характеристика оборудования

В таблице 4.1 представлены характеристики используемого оборудования.

Т а б л и ц а 4 . 1 – Оборудование и его характеристики

Наименование оборудования	Характеристики
Стационарный компьютер	Windows 7, Intel® Atom (TM) CPU D525 @1.80 GHz, 2048МБ, 320ГБ, Intel GMA HD, DOS, 15.6", DVD±RW (DL), VGA, HDMI, RJ-45, Super Multi Dual Layer, Intel HD Graphics) Количество – 3 шт., Мощность – 40Вт;
Принтеры, сканеры или множительные аппараты	МФУ Canon MF 4430 Adf “3d 1”, i-sensys MF 4340d Количество 2-шт.
Кондиционер настенный Gree -09 GW	Мощность по холоду 16.7 кВт Мощность компрессора 4.5 кВт Мощность электронагревателя 6.6 кВт

Работа с персональным компьютером – это воспроизведение визуальной информации на дисплее, которая должна быстро и точно восприниматься пользователем.

Основным фактором, влияющим на производительность труда людей, работающих с компьютером, являются комфортные и безопасные условия труда.

Условия труда пользователя, работающего с персональным компьютером, определяются:

- особенностями организации рабочего места;
- условиями производственной среды (освещением, микроклиматом, шумом, электромагнитными и электростатическими полями, визуальными эргономическими параметрами дисплея и т. д.);
- характеристиками информационного взаимодействия человека и персональных электронно-вычислительных машин.

При выполнении работ на персональном компьютере (ПК) могут иметь место следующие факторы:

- повышенная температура поверхностей ПК;
- повышенная или пониженная температура воздуха рабочей зоны;
- выделение в воздух рабочей зоны ряда химических веществ;
- повышенная или пониженная влажность воздуха;
- повышенный или пониженный уровень отрицательных и положительных аэроионов;
- повышенное значение напряжения в электрической цепи, замыкание;
- повышенный уровень статического электричества;
- повышенный уровень электромагнитных излучений;
- повышенная напряженность электрического поля;
- отсутствие или недостаток естественного света;

- недостаточная искусственная освещенность рабочей зоны;
- повышенная яркость света;
- повышенная контрастность;
- прямая и отраженная блескость;
- зрительное напряжение;
- монотонность трудового процесса;
- нервно-эмоциональные перегрузки.

Повышенные статические и динамические нагрузки у пользователей ПК приводят к жалобам на боли в спине, шейном отделе позвоночника и руках. Из всех недомоганий, обусловленных работой на компьютерах, чаще встречаются те, которые связаны с использованием клавиатуры. В период выполнения операций ввода данных количество мелких стереотипных движений кистей и пальцев рук за смену может превысить 60 тыс., что в соответствии с гигиенической классификацией труда относится к категории вредных и опасных. Поскольку каждое нажатие на клавишу сопряжено с сокращением мышц, сухожилия непрерывно скользят вдоль костей и соприкасаются с тканями, вследствие чего могут развиваться болезненные воспалительные процессы.

К числу факторов, ухудшающих состояние здоровья пользователей компьютерной техники, относятся электромагнитное и электростатическое поля, акустический шум, изменение ионного состава воздуха и параметров микроклимата в помещении. Немаловажную роль играют эргономические параметры расположения экрана монитора (дисплея), состояние освещенности на рабочем месте, параметры мебели и характеристики помещения, где расположена компьютерная техника.

К физическим вредным и опасным факторам относятся: повышенные уровни электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; повышенный уровень статического электричества и запыленности воздуха рабочей зоны; повышенное содержание положительных аэронов и пониженное содержание отрицательных аэронов в воздухе рабочей зоны; повышенный уровень блескости и ослепленности; неравномерность распределения яркости в поле зрения; повышенная яркость светового изображения; повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.

Химические вредные и опасные факторы следующие: повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола и формальдегида.

Психологические вредные и опасные факторы: напряжение зрения и внимания; интеллектуальные, эмоциональные и длительные статические нагрузки; монотонность труда; большой объем информации, обрабатываемый в единицу времени; нерациональная организация рабочего места.

Тепловое воздействие ПК характеризуется повышением температуры тела, локальным избирательным нагревом клеток, тканей и органов вследствие перехода ПК в тепловую энергию. Интенсивность нагрева зависит от

количества поглощенной энергии и скорости оттока тепла от облучаемых участков тела. Отток тепла затруднен в органах и тканях с плохим кровоснабжением. К ним в первую очередь относится хрусталик глаза, вследствие чего возможно развитие катаракты. Тепловому воздействию ЭМП подвергаются также паренхиматозные органы (печень, поджелудочная железа) и полые органы, содержащие жидкость (мочевой пузырь, желудок). Нагревание их может вызвать обострение хронических заболеваний.

3.4 Мероприятия, обеспечивающие безопасность при работе на компьютере

Организация рабочего места в соответствии с требованиями эргономики.

Под рабочим местом условно понимают зону, оснащенную необходимыми техническими средствами, где работник или группа работников постоянно или временно выполняют одну работу или операцию.

Правильная планировка рабочих мест может существенно снизить действие целого ряда неблагоприятных факторов, действующих на работающего с ПЭВМ человека.

Санитарными правилами предусмотрены следующие нормы.

Помещения с ВДТ (видео-дисплейные терминалы) и ПЭВМ должны иметь естественное и искусственное освещение.

Естественное освещение должно осуществляться через светопроемы, ориентированные преимущественно на север и северо-восток и обеспечивать коэффициент естественной освещенности (КЕО) не ниже 1,2.

Для внутренней отделки интерьера помещений с ВДТ и ПЭВМ должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка – 0,7 – 0,8; для стен – 0,5 – 0,6; для пола – 0,3 – 0,5.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ВДТ и ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

Рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии 600 – 700 мм, но не ближе 500.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии 100 – 300 мм от края, обращенного к пользователю или на специальной,

регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

Рациональная организация режима труда и отдыха. Для предупреждения заболеваний, связанных с работой на компьютере необходима рациональная организация труда и отдыха, которая нормируется в соответствии с санитарными правилами.

Режимы труда и отдыха при работе с ПК должны организовываться в зависимости от вида и категории трудовой деятельности.

Для обеспечения оптимальной работоспособности, сохранения здоровья пользователей, а также предупреждения развития переутомления на протяжении рабочей смены (учебных занятий) должны устанавливаться регламентированные перерывы.

3.5 Расчет искусственного освещения помещения

Рациональное освещение в помещении, предназначенном для работы с ПЭВМ создается при наличии как естественного, так и искусственного освещения.

Недостаточное освещение приводит к сильному напряжению глаз, быстрой утомляемости, близорукости, снижению качества работы, увеличению брака. Яркое освещение раздражает сетчатку глаза, ослепляет, глаза быстро устают, растёт производственный травматизм.

Искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые, по сравнению с лампами накаливания, имеют существенные преимущества:

- по спектральному составу света они близки к дневному, естественному свету;
- обладают более высоким КПД (в 1,5-2 раза выше, чем КПД ламп накаливания);
- обладают повышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);
- более длительный срок службы.

Для создания нормальных условий, на рабочем месте проводят нормирование освещенности в зависимости от размеров объекта различения, контраста объекта с фоном. Определение нормированной освещенности ведется по разрядам и подразрядам выполняемых работ. Для работ, выполняемых разработчиком, отводится четвертый разряд и подразряд "Б". Минимальное значение нормированной освещенности согласно СНиП 23-05-95 $E_{\min}=200$ Лк для общей системы освещения.

Рассчитаем общее освещение офисного помещения размером 10x5.

Нормируемая освещенность - 300 лк.

В помещении принята система общего освещения люминесцентными лампами ЛБ (белого цвета), мощностью 40 Вт и световым потоком $\Phi_{л} = 3750$

Лк, диаметром 40 мм. и длиной со штырьками 1213,6 мм. Высота рабочей поверхности $h_p=0,8$ м.

Индекс помещения, определяется соотношением размеров освещаемого помещения

$$i = \frac{A \cdot B}{h(A + B)} \quad (4.1)$$

Определим необходимое расстояние между светильниками

$$L = \lambda \times h \quad (4.2)$$

где $\lambda = 1,2 \div 1,4$.

Высота подвеса светильников над рабочей поверхностью

$$h = H - h_{\text{раб}} \quad (4.3)$$

По этим данным находим, что необходимое расстояние между светильниками равно

$$L = 2,2 * 1,3 = 2,86 = 3 \text{ м}$$

Тогда индекс помещения определим, как

$$i = \frac{A \cdot B}{h(A + B)} \quad (4.4)$$

$$i = 10 * 5 / 2,2 * 15 = 1,515 \text{ м}$$

В качестве светильника возьмем ЛД-65 рассчитанный на две лампы мощностью 40 Вт, диаметром 40 мм и длиной со штырьками 1213,6 мм. Световой поток лампы ЛД-65 40 Вт составляет 3750 Лк, световой поток, излучаемый светильником равен

$$\Phi_{\text{СВ}} = \Phi_{\text{Л}} * 2 \quad (4.5)$$

$$\Phi_{\text{СВ}} = 3750 * 2 = 7500 \text{ лм}$$

Определим число светильников по формуле (4.6)

$$N = \frac{E \times K_3 \times S \times z}{n \times \Phi_{\text{Л}} \times \eta} \quad (4.6)$$

где S – площадь помещения, $S = 50 \text{ м}^2$;

K_3 – коэффициент запаса, $K_3 = 1,5$;

E – заданная минимальная освещенность, $E = 300 \text{ лк}$;

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

N – количество ламп в светильнике, $n = 2$;

$\Phi_{\text{л}}$ – световой поток выбранной лампы, $\Phi_{\text{СВ}} = 3750 \text{ лм}$;

η – коэффициент использования, $\eta = 0,61$.

$$N = \frac{300 * 1,5 * 50 * 1,2}{2 * 3750 * 0,61} = \frac{27000}{4575} = 5,90 \cong 6 \text{ шт.}$$

Размещаем 2 ряда светильников по 3 светильника в ряду с расстоянием между светильниками в ряду 3 метра. Всего для создания нормируемой освещенности 300 лк необходимо 8 ламп ЛД-65 мощностью 40 Вт.

Тогда расчетный световой поток будет определяться по формуле

$$F_p = \frac{E \cdot K_3 \cdot S \cdot z}{N \cdot \eta} \quad (4.7)$$

$$F_p = \frac{300 * 1,5 * 10 * 5 * 1,1}{6 * 2 * 0,61} = \frac{24750}{7,32} = 3381 \text{ лм}$$

Проверка светового потока

$$\Delta F = \frac{F_l - F_p}{F_l} * 100\% \quad (4.8)$$

$$\Delta F = \frac{3750 - 3381}{3750} * 100\% = 0,098 = 9,8\%$$

Полученная величина находится в пределах $-10\% \leq \Delta F \leq +20\%$, значит перерасчет светового потока не требуется.

3.6 Расчет системы кондиционирования офиса

Приведем расчет системы кондиционирования в помещени. Кондиционирование обеспечивает наилучшее микроклимата в помещении и условия работы точной и чувствительной аппаратуры, и должно выполняться в соответствии с главой СНиП 11-33-75 "Отопление, вентиляция и кондиционирование воздуха".

Определим количество воздуха L м³/ч, которое необходимо вывести за один час из помещения, чтобы вместе с ним удалить избыток тепла Q_{H36} , определяется по формуле

$$L = \frac{Q_{\text{изб}}}{C_D * t * \gamma_B} \quad (4.9)$$

где C_B – теплоемкость сухого воздуха, ккал/кг ($C_B = 0,24$ ккал/кг град);

$t = t_{\text{ух}} - t_{\text{вх}}$ при расчетах возьмем $t=10^\circ\text{C}$;

γ_B – плотность уходящего воздуха, определяемая в зависимости от температуры, кг/м³ (при расчетах принимается $\gamma_B=1,20$ кг/м³).

Определим избыточное тепло $Q_{\text{изб}}$ ккал/ч

$$Q_{\text{изб}} = Q_{\text{п}} - Q_{\text{от}} \quad (4.10)$$

где $Q_{\text{п}}$ – количества тепла поступающего в воздух помещения, ккал/ч;

$Q_{\text{от}}$ – теплоотдача в окружающую среду через наружные ограждения (в теплое время года, при расчетах можно принять нулю).

Количество тепловыделений $Q_{\text{п}}$ зависит от мощности оборудования, числа работающих людей и тепла, которое вносится в помещение через оконные проемы

$$Q_{\text{п}} = Q_{\text{сб}} + Q_{\text{л}} + Q_{\text{р}} + Q_{\text{осв}} \quad (4.11)$$

где $Q_{\text{сб}}$ – тепло, выделяемое производственным оборудованием, ккал/ч;

$Q_{\text{л}}$ – тепло выделяемое людьми, ккал/ч;

$Q_{\text{р}}$ – тепло, вносимое солнечной радиацией, ккал/ч.

Тепло, выделяемое производственным оборудованием в рабочем помещении, определяется из соотношения

$$Q_{\text{сб}} = 860 * P_{\text{об}} * n * N \quad (4.12)$$

где 860 тепловой эквивалент 1 кВт·ч, то есть тепло, эквивалентное 1 кВт·ч электрической энергии;

$P_{\text{об}}$ – мощность, потребляемая оборудованием $P_{\text{об}} = 350$ Вт;

n – коэффициент перехода тепла в помещение, $n=0,75$;

N – количество производственного оборудования.

$$Q_{\text{сб}} = 860 * 0,35 * 0,75 * 6 = 1354,5 \text{ ккал/ч}$$

Тепло, вносимое солнечной, радиацией, определяется из соотношения

$$Q_{\text{р}} = m * F * g_{\text{ост}} \quad (4.13)$$

где m – количество окон в помещении;

F – площадь одного окна $F=8\text{м}^2$;

$g_{\text{ост}}$ – солнечная радиация через остекленную поверхность, то есть количество тепла, вносимое за 1ч через остекление площадью в 1 м².

$$Q_p = 1 \cdot 8 \cdot 145 = 1160 \text{ ккал/ч}$$

Количество тепла, выделяемого находящимися в помещении людьми

$$Q_l = N \cdot q_l \quad (4.14)$$

где N – количество людей в помещении;

q_l , ккал/ч – тепловыделения одного человека $q_l = 60$.

$$Q_l = 6 \cdot 60 = 360 \text{ ккал/ч}$$

Количество тепла, выделяемого системой освещения

$$Q_{осв} = 860 \cdot P_{осв} \cdot \alpha \cdot b \cdot \cos(\varphi) \quad (4.15)$$

где α – коэф.перевода электрической энергии в тепловую,

люминесцентные лампы $\alpha = 0,46 - 0,48$;

b – коэффициент одновременности работы (при работе всех светильников $b = 1$);

$\cos(\varphi) = 0,7 - 0,8$ – коэффициент мощности;

$P_{осв}$, кВт – мощность осветительной установки.

$$Q_{осв} = 860 \cdot 0,48 \cdot 0,8 \cdot 0,04 \cdot 8 = 105,68 \text{ ккал/ч}$$

Тогда тепловыделение составит

$$Q_{ИЗБ} = 1354,5 + 1160 + 360 + 105,68 = 2980,18 \text{ ккал/ч}$$

Необходимый воздухообмен рассчитаем

$$L = \frac{2980,18}{0,24 \cdot 10 \cdot 1,20} = 1035 \text{ м}^3/\text{ч}$$

Отношение количества воздуха, поступающего в помещение за один час, к объему помещения называется кратностью воздухообмена

$$K = \frac{L}{V_{п}} \quad (4.16)$$

где $V_{п}$ – объем помещения

$$V_{п} = 10 \cdot 5 \cdot 3 = 150 \text{ м}^3,$$

$$K = \frac{1035}{150} = 6,9.$$

Находим требуемую производительность кондиционера

$$W_K = k_3 * L \quad (4.17)$$

где k_3 – коэффициент запаса, $k_3 = 1,3 \div 2,0$.

$$W_K = 2 * 1035 = 2070 \text{ м}^3/\text{ч}.$$

Исходя из расчетов в помещение с оборудованием, для соблюдения требуемых параметров микроклимата следует установить один кондиционер с производительностью не менее 2070 м³/ч. Выбранный кондиционер York EANS 09 FS-R. Расход воздуха (внутр./внешн.), м³/ч: 950/3310. Размещение светильников в рабочем помещении представлено на рисунке 4.2.

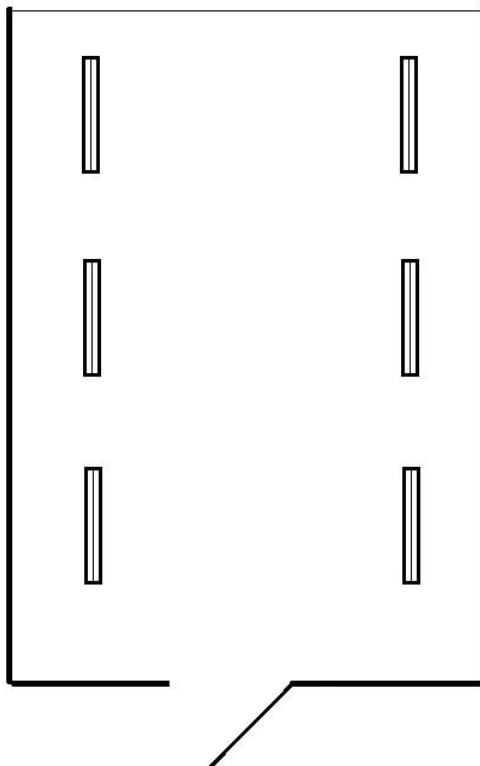


Рисунок 4.2 – Размещение светильников в рабочем помещении

В результате проделанных расчетов просчитаны необходимые меры безопасности и условия труда инженера инфокоммуникационные оборудования, которые соответствуют стандартам СНиП, и ГОСТ. Произведена необходимая освещенность рабочей поверхности.

Заключение

В результате дипломной работы была создана автоматизированная информационная система анализа финансового состояния страховой компании.

Программа предназначена для автоматизации работы финансового аналитика с управленческой отчетностью, позволяющая на основании внесенных в систему данных проводить анализ финансового состояния страховой компании, рассчитывать показатели, параметры, коэффициенты и мультипликаторы для оценки финансового состояния организации и стоимости акций компаний с целью принятия решений о размещении капитала самостоятельно.

В настоящее время разработка и внедрение таких систем в экономике в целом практикуется не часто, а особенно в области страхования. Это обусловлено большой ценой на аналитические системы, сложностью внедрения, недостаточной технической обеспеченностью современных компаний. К тому же в область страхования является специфической областью поэтому для разработки программных продуктов в данной сфере необходимо привлечение высококвалифицированных разработчиков в сфере IT, а также консультантов по экономическому и финансовому направлению. Всё это затрудняет и замедляет процесс усовершенствования оснащённости техническими средствами страховых компаний и экономической отрасли в общем виде. На сегодняшний день пройден лишь этап первоначальной автоматизации организаций, но не разрабатываются серьёзные системы, которые могли бы проводить детальный анализ финансового состояния компании, и её дальнейшего развития. До настоящего времени разработкой такого продукта в специфике страховой компании не занимались, он является совершенно новым, отвечающим всем современным стандартам.

Данная система представляет собой хранилище данных, из которого пользователь имеет возможность получать проанализированную за несколько лет информацию. Благодаря использованию данной системы существенно сокращается время работы финансового отдела и увеличивается его работа. Уменьшается время на подготовку и предоставление отчётной информации руководителю компании, а следовательно эффективность работы самого руководителя по дальнейшему экономическому планированию увеличивается. Во много раз увеличивается точность предоставляемых данных за счёт исключения «человеческого фактора» в работе с важными данными.

Простой интерфейс программы позволяет быстро понять принцип её работы и максимально использовать все её функции.

Список литературы

- 1 Генник Дж. Oracle SQL*plus, Карманный справочник. – 2-е изд. –СПб.: Питер, 2004. – 189 с.
- 2 Сатимова Е.Г. Проектирование баз данных. Методические указания к выполнению лабораторных работ (для студентов всех специальностей). – Алматы: АИЭС, 2009.
- 3 Зобнин Б.Б. Моделирование систем: Конспект лекций. – Екатеринбург, 2007.
- 4 Информационные технологии управления: Учебное пособие для ВУЗов / под ред. Г.А. Титоренко. – М.:ЮНИТИ-ДАНА, 2005.
- 5 Иванов В.И. Информационные системы. – М.: АРТ, 2009.
- 6 Информационно-аналитические системы поддержки принятия решений в страховании / Дмитрий Бобровский. – Финансовая газета ЭКСПО, 2006
- 7 Семенов М.И. и другие. Автоматизированные информационные технологии в экономике: Учебник. – М.: Финансы и статистика, 2006.
- 8 Рудикова Л.В. Базы данных. Разработка приложений. – СПб: БХВ-Петербург, 2006.
- 9 Барановская Т.П., Лойко В.И. и др. Информационные системы и технологии в экономике: Учебник. – М.: Финансы и статистика, 2010.
- 10 Э.Спирли. Корпоративные хранилища. – М.: Издательский дом «Вильямс», 2001.
- 11 Bob Griesemer. Oracle Warehouse Builder 11g. – Packt, 2009.
- 12 А.И.Кудинов. Хранилища данных. Цикл лабораторных работ. Часть 1. – Томск: ТПУ, 2007.
- 13 А.И.Кудинов. Хранилища данных. Цикл лабораторных работ. Часть 2. – Томск: ТПУ, 2008.
- 14 Грэм Мэлколм. Бизнес анализ в SQL Server 2008, 2007.
- 15 Питер Роб, Карлос Коронел. Системы баз данных: проектирование, реализация и управление. – 5-е издание. – СПб: ВНУ Санкт-Петербург, 2004 .
- 16 ГОСТ 12.1.005-88. Санитарно-гигиенические требования к воздуху рабочей зоны.
- 17 З.Д. Еркешева, Г.Ш. Боканова. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. – Алматы: АУЭС, 2013. – 40 с.
- 18 ГОСТ ИСО/МЭК 12119. Информационная технология. Пакеты программ. Требование к качеству и тестирование, 2010.
- 19 ГОСТ 28195-89. Оценка качества программных средств. Общие положения, 2011.

Приложение А

```
create user Diplom identified by Diplom
default tablespace Diplom quota 40M on Diplom
temporary tablespace temp
/
alter user Diplom QUOTA unlimited ON Diplom;
/
commit
/
grant create session to Diplom with admin option
/
grant create table to Diplom
/
grant create procedure to Diplom
/
grant create trigger to Diplom
/
grant create sequence to Diplom
/
grant create view to Diplom
/
grant CREATE SESSION to Diplom
/
grant CREATE JOB to Diplom
/
grant CREATE ANY JOB to Diplom
/
grant MANAGE SCHEDULER to Diplom
/
grant CREATE EXTERNAL JOB to Diplom
/
commit
/
create table NAPR(
ID_Napr number not null primary key,
Nazv_N varchar2(10));
/
create table VID(
ID_VID number not null primary key,
Nazv_V varchar2(80));
/
create table POKAZATEL(
ID_POKAZ number not null primary key,
ID_Napr number references NAPR (ID_Napr),
ID_VID number references VID (ID_VID),
Name_Pokaz varchar2(80));
/
create table BALANCE(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Summa number(12,0),
```

```

Datee date);
/
create table OPiU(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Summa number(12,0),
Datee date);
/
create table SUPPORT(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Znach number(10,2),
Datee date);
/
create table CAPITAL(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Ustav number(12,0),
Dobav number(12,0),
Izyat number(12,0),
Rezerv number(12,0),
Pereocen number(12,0),
Neraspr number(12,0),
Itogo number(12,0),
Datee date);
/
create table CALC(
ID_POKAZ number references POKAZATEL (ID_POKAZ),
Znach number(12,2),
Datee date);
/
commit;
/
Create or replace procedure PA (v_datte in date:='&p_datte')
is
P1 number;
P2 number;
P3 number;
P4 number;
KZ_poop number;
KZ_perestr number;
KZ_proch number;
Itogo_tretiy number;
KZ_depo_pr number;
KZ_kratkosroch number;
Dolg_zaimy number;
Itogo_vtoroy number;
Nepokr_ubitk number;
Doh_budush_per number;
Reserv_predupr_mer number;
Rezerv_predst_rash number;
begin
Select Summa into KZ_poop from Balance where ID_POKAZ=1640 and
DATEE=v_datte;

```

```

Select Summa into KZ_perestr from Balance where ID_POKAZ=1650
and DATEE=v_datte;
Select Summa into KZ_proch from Balance where ID_POKAZ=1660
and DATEE=v_datte;
Select Summa into Itogo_tretiy from Balance where
ID_POKAZ=1590 and DATEE=v_datte;
Select Summa into KZ_depo_pr from Balance where ID_POKAZ=1190
and DATEE=v_datte;
Select Summa into KZ_kratkosroch from Balance where
ID_POKAZ=1200 and DATEE=v_datte;
Select Summa into Dolg_zaimy from Balance where ID_POKAZ=1615
and DATEE=v_datte;
Select Summa into Itogo_vtoroy from Balance where
ID_POKAZ=1490 and DATEE=v_datte;
Select Summa into Nepokr_ubitk from Balance where
ID_POKAZ=1475 and DATEE=v_datte;
Select Summa into Doh_budush_per from Balance where
ID_POKAZ=1665 and DATEE=v_datte;
Select Summa into Reserv_predupr_mer from Balance where
ID_POKAZ=1681 and DATEE=v_datte;
Select Summa into Rezerv_predst_rash from Balance where
ID_POKAZ=1680 and DATEE=v_datte;
P1:=KZ_poop+KZ_perestr+KZ_proch;
P2:=Itogo_tretiy+KZ_depo_pr+KZ_kratkosroch;
P3:=Dolg_zaimy;
P4:=Itogo_vtoroy+Nepokr_ubitk+Doh_budush_per+Reserv_predupr_me
r+Rezerv_predst_rash;
Insert into CALC values (3004,P1,v_datte);
Insert into CALC values (3005,P2,v_datte);
Insert into CALC values (3006,P3,v_datte);
Insert into CALC values (3007,P4,v_datte);
commit;
end PA;
/
Execute PA ('31.12.13');
/
Create or replace procedure A_P (v_datte in date:='&p_datte')
is
v_A1_P1 number;
v_A2_P2 number;
v_A3_P3 number;
A1 number;
A2 number;
A3 number;
P1 number;
P2 number;
P3 number;
begin
Select ZNACH into A1 from CALC where ID_POKAZ=3000 and
DATEE=v_datte;
Select ZNACH into A2 from CALC where ID_POKAZ=3001 and
DATEE=v_datte;

```

```

Select ZNACH into A3 from CALC where ID_POKAZ=3002 and
DATEE=v_datte;
Select ZNACH into P1 from CALC where ID_POKAZ=3004 and
DATEE=v_datte;
Select ZNACH into P2 from CALC where ID_POKAZ=3005 and
DATEE=v_datte;
Select ZNACH into P3 from CALC where ID_POKAZ=3006 and
DATEE=v_datte;
v_A1_P1:=A1-P1;
v_A2_P2:=A2-P2;
v_A3_P3:=A3-P3;
Insert into CALC values (3008,v_A1_P1,v_datte);
Insert into CALC values (3009,v_A2_P2,v_datte);
Insert into CALC values (3010,v_A3_P3,v_datte);
commit;
end A_P;
/
Execute A_P ('01.10.13');
/
Create or replace procedure K_likv_balance (v_datte in
date:='&p_datte')
is
K_abs_likv number;
K_pokritiya number;
K_bistr_likv number;
K_ceny_likv number;
K_likv_obsh number;
K_persp_plat number;
K_zadolj number;
K_platej_obsh number;
A1 number;
A2 number;
A3 number;
A4 number;
P1 number;
P2 number;
P3 number;
begin
Select ZNACH into A1 from CALC where ID_POKAZ=3000 and
DATEE=v_datte;
Select ZNACH into A2 from CALC where ID_POKAZ=3001 and
DATEE=v_datte;
Select ZNACH into A3 from CALC where ID_POKAZ=3002 and
DATEE=v_datte;
Select ZNACH into A4 from CALC where ID_POKAZ=3003 and
DATEE=v_datte;
Select ZNACH into P1 from CALC where ID_POKAZ=3004 and
DATEE=v_datte;
Select ZNACH into P2 from CALC where ID_POKAZ=3005 and
DATEE=v_datte;
Select ZNACH into P3 from CALC where ID_POKAZ=3006 and
DATEE=v_datte;

```

```

K_abs_likv:=A1/P1 + P2;
K_pokritiya:=(A1+A2+A3)/(P1+P2);
K_bistr_likv:=(A1+A2)/(P1+P2);
K_ceny_likv:=(A1+A2+A3+A4)/(P1+P2+P3);
K_likv_obsh:=(A1+0.5*A2+0.3*A3)/(P1+0.2*P2+0.3*P3);
K_persp_plat:=P3/A3;
K_zadolj:=P3/(A1+A2+A3+A4);
K_platej_obsh:=(P2+P3)/(A3+A4);
Insert into CALC values (3015,K_abs_likv,v_datte);
Insert into CALC values (3016,K_pokritiya,v_datte);
Insert into CALC values (3022,K_bistr_likv,v_datte);
Insert into CALC values (3017,K_ceny_likv,v_datte);
Insert into CALC values (3018,K_likv_obsh,v_datte);
Insert into CALC values (3019,K_persp_plat,v_datte);
Insert into CALC values (3020,K_zadolj,v_datte);
Insert into CALC values (3021,K_platej_obsh,v_datte);
commit;
end K_likv_balance;
/
Execute K_likv_balance ('01.07.14');
/
Create or replace procedure Fin_ustoych2 (v_datte in
date:='&p_datte')
is
Otn_tekh_rezer_SK_Netto number;
Otn_netto_Sob_sred number;
Otn_sob_sred_Netto number;
Otn_tekh_rez_Sob_sred number;
Otn_tekh_rez_Netto number;
Otn_strach_vznos_Obsh_vz number;
Otn_obyaz_Likv_aktiv number;
Otn_DZ_strach_vzns_Sob_sr number;
Otn_KZ_k_Aktiv number;
Dohod_inv_k_V_inv number;
Dohod_inv_k_aktiv number;
Dohod_inv_k_SK number;
Otn_strakh_vzns_Pribyl number;
Pokaz_chist_ubytok number;
v_28 number;
v_29 number;
v_30 number;
v_31 number;
v_32 number;
v_33 number;
v_34 number;
v_35 number;
v_36 number;
v_37 number;
v_38 number;
v_39 number;
v_40 number;
v_41 number;

```

```

v_42 number;
v_43 number;
v_44 number;
v_45 number;
v_46 number;
v_47 number;
v_48 number;
v_49 number;
v_50 number;
v_51 number;
v_52 number;
v_53 number;
v_54 number;
v_55 number;
v_56 number;
begin
Select Summa into v_28 from Balance where ID_POKAZ=1590 and
DATEE=v_datte;
Select Summa into v_29 from Balance where ID_POKAZ=1490 and
DATEE=v_datte;
Select Summa into v_30 from Balance where ID_POKAZ=1190 and
DATEE=v_datte;
Select Summa into v_31 from Balance where ID_POKAZ=1200 and
DATEE=v_datte;
Select Summa into v_32 from Balance where ID_POKAZ=1520 and
DATEE=v_datte;
Select Summa into v_33 from Balance where ID_POKAZ=1530 and
DATEE=v_datte;
Select Summa into v_34 from OPIU where ID_POKAZ=80 and
DATEE=v_datte;
Select Summa into v_35 from OPIU where ID_POKAZ=12 and
DATEE=v_datte;
Select Summa into v_36 from OPIU where ID_POKAZ=11 and
DATEE=v_datte;
Select Summa into v_37 from OPIU where ID_POKAZ=82 and
DATEE=v_datte;
Select Summa into v_38 from OPIU where ID_POKAZ=81 and
DATEE=v_datte;
Select Summa into v_39 from Balance where ID_POKAZ=1210 and
DATEE=v_datte;
Select Summa into v_40 from Balance where ID_POKAZ=1220 and
DATEE=v_datte;
Select Summa into v_41 from Balance where ID_POKAZ=1230 and
DATEE=v_datte;
Select Summa into v_42 from Balance where ID_POKAZ=1620 and
DATEE=v_datte;
Select Summa into v_43 from Balance where ID_POKAZ=1630 and
DATEE=v_datte;
Select Summa into v_44 from Balance where ID_POKAZ=1640 and
DATEE=v_datte;
Select Summa into v_45 from Balance where ID_POKAZ=1650 and
DATEE=v_datte;

```

```

Select Summa into v_46 from Balance where ID_POKAZ=1660 and
DATEE=v_datte;
Select Summa into v_47 from Balance where ID_POKAZ=1300 and
DATEE=v_datte;
Select Summa into v_48 from OPIU where ID_POKAZ=180 and
DATEE=v_datte;
Select Summa into v_49 from Balance where ID_POKAZ=1120 and
DATEE=v_datte;
Select Summa into v_50 from Balance where ID_POKAZ=1290 and
DATEE=v_datte;
Select Summa into v_51 from OPIU where ID_POKAZ=10 and
DATEE=v_datte;
Select Summa into v_53 from OPIU where ID_POKAZ=20 and
DATEE=v_datte;
Select Summa into v_52 from Balance where ID_POKAZ=1470 and
DATEE=v_datte;
Select Summa into v_54 from OPIU where ID_POKAZ=111 and
DATEE=v_datte;
Select Summa into v_55 from Balance where ID_POKAZ=1530 and
DATEE=v_datte;
Select Summa into v_56 from Balance where ID_POKAZ=1260 and
DATEE=v_datte;
Otn_tekh_rezer_Sk_netto:=ROUND(((v_28+v_29-v_30-
v_31)/(v_32+v_33+v_34)),2)*100;
Otn_netto_Sob_sred:=ROUND((v_34/(v_29-v_30-v_31-v_52)),2)*100;
if v_34=0 then Otn_strakh_vzns_Pribyl:=0;
else Otn_sob_sred_Netto:=ROUND(((v_29-v_30-v_31-
v_52)/v_34),2)*100;
end if;
Otn_tekh_rez_Sob_sred:=ROUND((v_28/(v_29-v_30-v_31-
v_52)),2)*100;
if v_34=0 then Otn_strakh_vzns_Pribyl:=0;
else Otn_tekh_rez_Netto:=ROUND((v_28/v_34),2)*100;
end if;
Otn_strach_vznos_Obsh_vz:=ROUND(((v_35+v_37)/(v_36+v_38)),2)*1
00;
if v_56=0 then Otn_strakh_vzns_Pribyl:=0;
else
Otn_obyaz_Likv_aktiv:=ROUND(((v_44+v_45+v_46)/v_56),2)*100;
end if;
Otn_DZ_strach_vzns_Sob_sr:=ROUND(((v_30+v_31+v_39+v_40+v_41)/(
v_29-v_30-v_31-v_52)),2)*100;
Otn_KZ_k_Aktiv:=ROUND(((v_42+v_43+v_44+v_45+v_46)/(v_47-
v_52)),2)*100;
if v_49=0 then Otn_strakh_vzns_Pribyl:=0;
else Dohod_inv_k_V_inv:=ROUND((v_53/v_49),2)*100;
end if;
if v_50=0 then Otn_strakh_vzns_Pribyl:=0;
else Dohod_inv_k_aktiv:=ROUND((v_53/v_50),2)*100;
end if;
if v_29=0 then Otn_strakh_vzns_Pribyl:=0;
else Dohod_inv_k_SK:=ROUND((v_53/v_29),2)*100;

```



```

end if;
if v_52=0 then Otn_strakh_vzns_Pribyl:=0;
else Otn_strakh_vzns_Pribyl:=ROUND(((v_51+v_34)/v_52),2)*100;
end if;
Pokaz_chist_ubytok:=ROUND(((v_54+v_55)/(v_34+v_51)),2)*100;
Insert into CALC values
(3028,Otn_tekh_rezer_Sk_netto,v_datte);
Insert into CALC values (3029,Otn_netto_Sob_sred,v_datte);
Insert into CALC values (3030,Otn_sob_sred_Netto,v_datte);
Insert into CALC values (3031,Otn_tekh_rez_Sob_sred,v_datte);
Insert into CALC values (3032,Otn_tekh_rez_Netto,v_datte);
Insert into CALC values
(3033,Otn_strach_vznos_Obsh_vz,v_datte);
Insert into CALC values (3034,Otn_obyaz_Likv_aktiv,v_datte);
Insert into CALC values
(3035,Otn_DZ_strach_vzns_Sob_sr,v_datte);
Insert into CALC values (3036,Otn_KZ_k_Aktiv,v_datte);
Insert into CALC values (3037,Dohod_inv_k_V_inv,v_datte);
Insert into CALC values (3038,Dohod_inv_k_aktiv,v_datte);
Insert into CALC values (3039,Dohod_inv_k_SK,v_datte);
Insert into CALC values (3040,Otn_strakh_vzns_Pribyl,v_datte);
Insert into CALC values (3041,Pokaz_chist_ubytok,v_datte);
commit;
end Fin_ustoych2;
/
Execute Fin_ustoych2 ('01.07.14');
/
Create or replace procedure Otn_Fin_ustoych (v_datte in
date:='&p_datte')
is
Koeff_SK number;
Koeff_zaem_kap number;
Koeff_obsh_likv number;
Koeff_srochn_likv number;
Koeff_Konshin number;
SK number;
Zaem number;
v_56 number;
v_57 number;
v_58 number;
v_59 number;
v_60 number;
v_61 number;
v_62 number;
v_63 number;
v_64 number;
v_65 number;
v_66 number;
Obyaz number;
v_q number;
v_n number;
Visoko_Likv_Akt number;

```

```

begin
Select SUMMA into SK from BALANCE where ID_POKAZ=1490 and
DATEE=v_datte;
Select SUMMA into v_56 from BALANCE where ID_POKAZ=1640 and
DATEE=v_datte;
Select SUMMA into v_57 from BALANCE where ID_POKAZ=1650 and
DATEE=v_datte;
Select SUMMA into v_58 from BALANCE where ID_POKAZ=1660 and
DATEE=v_datte;
Select SUMMA into v_59 from BALANCE where ID_POKAZ=1620 and
DATEE=v_datte;
Select SUMMA into v_60 from BALANCE where ID_POKAZ=1630 and
DATEE=v_datte;
Select SUMMA into Obyaz from BALANCE where ID_POKAZ=1690 and
DATEE=v_datte;
Select SUMMA into v_61 from BALANCE where ID_POKAZ=1250 and
DATEE=v_datte;
Select SUMMA into v_62 from BALANCE where ID_POKAZ=1190 and
DATEE=v_datte;
Select SUMMA into v_63 from BALANCE where ID_POKAZ=1200 and
DATEE=v_datte;
Select SUMMA into v_64 from BALANCE where ID_POKAZ=1210 and
DATEE=v_datte;
Select SUMMA into v_65 from BALANCE where ID_POKAZ=1220 and
DATEE=v_datte;
Select SUMMA into v_66 from BALANCE where ID_POKAZ=1260 and
DATEE=v_datte;
Select SUMMA into Visoko_Likv_Akt from BALANCE where
ID_POKAZ=1130 and DATEE=v_datte;
Select ZNACH into v_q from SUPPORT where ID_POKAZ=4001 and
DATEE=v_datte;
Select ZNACH into v_n from SUPPORT where ID_POKAZ=4002 and
DATEE=v_datte;
Koeff_SK:=ROUND((SK*100/(Obyaz+SK)),3);
Koeff_zaem_kap:=ROUND((v_56+v_57+v_58+v_59+v_60)*100/(v_56+v_5
7+v_58+v_59+v_60+SK),3);
if v_60=0 then Koeff_obsh_likv:=0;
else
Koeff_obsh_likv:=ROUND(((v_61+v_62+v_63+v_64+v_65+v_66)/v_60),
3);
end if;

if v_60=0 then Koeff_srochn_likv:=0;
else Koeff_srochn_likv:=ROUND((v_66/v_60),3);
end if;
/
Koeff_Konshin:=ROUND(((1-v_q)/(v_q*v_n)),3);
/
Insert into CALC values (3044,Koeff_SK,v_datte);
Insert into CALC values (3045,Koeff_zaem_kap,v_datte);
Insert into CALC values (3046,Koeff_obsh_likv,v_datte);
Insert into CALC values (3047,Koeff_srochn_likv,v_datte);

```

```

Insert into CALC values (3048,Koeff_Konshin,v_datte);
commit;
end Otn_Fin_ustoych;
/
Execute Otn_Fin_ustoych ('01.07.14');
/
Create or replace procedure Reservy_Perestrakh (v_datte in
date:='&p_datte')
is
Dost_strakh_rezerv number;
Ud_ves_prem_v_perest number;
Uch_perest_v_sost_ubyt number;
Uch_perest_v_strakh_rez number;
SR number;
v_67 number;
v_68 number;
v_69 number;
v_70 number;
v_71 number;
v_72 number;
v_73 number;
v_74 number;
v_75 number;
v_76 number;
v_77 number;
v_78 number;
v_79 number;
v_80 number;
v_81 number;
v_82 number;
begin
Select SUMMA into SR from BALANCE where ID_POKAZ=1590 and
DATEE=v_datte;
Select SUMMA into v_67 from BALANCE where ID_POKAZ=1160 and
DATEE=v_datte;
Select SUMMA into v_68 from BALANCE where ID_POKAZ=1170 and
DATEE=v_datte;
Select SUMMA into v_69 from BALANCE where ID_POKAZ=1180 and
DATEE=v_datte;
Select SUMMA into v_70 from OPIU where ID_POKAZ=10 and
DATEE=v_datte;
Select SUMMA into v_71 from OPIU where ID_POKAZ=80 and
DATEE=v_datte;
Select SUMMA into v_72 from OPIU where ID_POKAZ=50 and
DATEE=v_datte;
Select SUMMA into v_73 from OPIU where ID_POKAZ=160 and
DATEE=v_datte;
Select SUMMA into v_74 from OPIU where ID_POKAZ=12 and
DATEE=v_datte;
Select SUMMA into v_75 from OPIU where ID_POKAZ=82 and
DATEE=v_datte;

```

```

Select SUMMA into v_76 from OPIU where ID_POKAZ=80 and
DATEE=v_datte;
Select SUMMA into v_77 from OPIU where ID_POKAZ=10 and
DATEE=v_datte;
Select SUMMA into v_78 from OPIU where ID_POKAZ=112 and
DATEE=v_datte;
Select SUMMA into v_79 from OPIU where ID_POKAZ=100 and
DATEE=v_datte;
Select SUMMA into v_80 from BALANCE where ID_POKAZ=1510 and
DATEE=v_datte;
Select SUMMA into v_81 from BALANCE where ID_POKAZ=1520 and
DATEE=v_datte;
Select SUMMA into v_82 from BALANCE where ID_POKAZ=1530 and
DATEE=v_datte;
Dost_strakh_rezerv:=ROUND(((SR-(v_67+v_68+v_69))/(v_70+v_71-
(v_72+v_73))),3);
Ud_ves_prem_v_perest:=ROUND(((v_74+v_75)/(v_77+v_76)),3);
Uch_perest_v_sost_ubytk:=ROUND(((v_67+v_68+v_69)/v_79),3);
Uch_perest_v_strakh_rez:=ROUND(((v_67+v_68+v_69)/(v_80+v_81+v_
82)),3);
Insert into CALC values (3049,Dost_strakh_rezerv,v_datte);
Insert into CALC values (3050,Ud_ves_prem_v_perest,v_datte);
Insert into CALC values
(3051,Uch_perest_v_sost_ubytk,v_datte);
Insert into CALC values
(3052,Uch_perest_v_strakh_rez,v_datte);
commit;
end Reservy_Perestrakh;
/
Execute Reservy_Perestrakh ('01.07.14');
/
Create or replace procedure Rentabelnost (v_datte in
date:='&p_datte')
is
Koeff_ubitochn number;
Koeff_doly_perestr number;
Koeff_raskh number;
Koeff_Ur_dohod_inv number;
Koeff_eff_inv_deyat number;
Koeff_rentab number;
Koeff_rentab_kapital number;
Koeff_rentab_strah number;
SR number;
v_83 number;
v_84 number;
v_85 number;
v_86 number;
v_87 number;
v_88 number;
v_89 number;
v_90 number;
v_91 number;

```

```

v_92 number;
v_93 number;
v_94 number;
v_95 number;
SK number;
Pn number;
RVD number;
v_96 number;
begin
Select SUMMA into v_83 from OPIU where ID_POKAZ=100 and
DATEE=v_datte;
Select SUMMA into v_84 from OPIU where ID_POKAZ=112 and
DATEE=v_datte;
Select SUMMA into v_85 from OPIU where ID_POKAZ=10 and
DATEE=v_datte;
Select SUMMA into v_86 from OPIU where ID_POKAZ=80 and
DATEE=v_datte;
Select SUMMA into v_87 from OPIU where ID_POKAZ=12 and
DATEE=v_datte;
Select SUMMA into v_88 from OPIU where ID_POKAZ=82 and
DATEE=v_datte;
Select SUMMA into v_89 from OPIU where ID_POKAZ=50 and
DATEE=v_datte;
Select SUMMA into v_90 from OPIU where ID_POKAZ=160 and
DATEE=v_datte;
Select SUMMA into v_91 from OPIU where ID_POKAZ=20 and
DATEE=v_datte;
Select SUMMA into v_92 from OPIU where ID_POKAZ=180 and
DATEE=v_datte;
Select SUMMA into v_93 from OPIU where ID_POKAZ=60 and
DATEE=v_datte;
Select SUMMA into v_94 from OPIU where ID_POKAZ=190 and
DATEE=v_datte;
Select SUMMA into v_95 from OPIU where ID_POKAZ=120 and
DATEE=v_datte;
Select SUMMA into SK from BALANCE where ID_POKAZ=1490 and
DATEE=v_datte;
Select SUMMA into Pn from OPIU where ID_POKAZ=270 and
DATEE=v_datte;
Select SUMMA into RVD from OPIU where ID_POKAZ=170 and
DATEE=v_datte;
Select SUMMA into v_96 from OPIU where ID_POKAZ=130 and
DATEE=v_datte;
Koeff_ubitochn:=(v_83-v_84)/(v_85+v_86);
Koeff_doly_perestr:=(v_87+v_88)/(v_85+v_86);
Koeff_raskh:=(v_89+v_90)/(v_85+v_86);
Koeff_Ur_dohod_inv:=(v_91+v_92-(v_93+v_94))/(v_85+v_86);
Koeff_eff_inv_deyat:=((v_92+v_95)-(v_93+v_94))/(v_95+v_96);
Koeff_rentab:=1+Koeff_Ur_dohod_inv+(Koeff_ubitochn+Koeff_eff_i
nv_deyat+Koeff_raskh);
Koeff_rentab_kapital:=Pn/SK;
Koeff_rentab_strah:=Pn/RVD;

```

```

Insert into CALC values (3054,Koeff_ubitochn,v_datte);
Insert into CALC values (3055,Koeff_doly_perestr,v_datte);
Insert into CALC values (3056,Koeff_raskh,v_datte);
Insert into CALC values (3057,Koeff_Ur_dohod_inv,v_datte);
Insert into CALC values (3058,Koeff_eff_inv_deyat,v_datte);
Insert into CALC values (3059,Koeff_rentab,v_datte);
Insert into CALC values (3060,Koeff_rentab_kapital,v_datte);
Insert into CALC values (3061,Koeff_rentab_strah,v_datte);
commit;
end Rentabelnost;
/
Execute Rentabelnost ('01.07.14');
/
Create or replace procedure Delo_aktivn (v_datte in
date:='&p_datte')
is
Koeff_obor_akt number;
Koeff_obor_SK number;
Obor_inv_akt number;
Obem_strakh_prem number;
Obem_strakh_vypl number;
Akt number;
SK number;
v_97 number;
v_98 number;
v_99 number;
v_100 number;
v_101 number;
v_102 number;
v_103 number;
v_104 number;
begin
Select SUMMA into v_97 from OPIU where ID_POKAZ=10 and
DATEE=v_datte;
Select SUMMA into v_98 from OPIU where ID_POKAZ=80 and
DATEE=v_datte;
Select SUMMA into Akt from BALANCE where ID_POKAZ=1290 and
DATEE=v_datte;
Select SUMMA into SK from BALANCE where ID_POKAZ=1490 and
DATEE=v_datte;
Select SUMMA into v_101 from BALANCE where ID_POKAZ=1120 and
DATEE=v_datte;
Select SUMMA into v_102 from BALANCE where ID_POKAZ=1130 and
DATEE=v_datte;
Select SUMMA into v_99 from OPIU where ID_POKAZ=20 and
DATEE=v_datte;
Select SUMMA into v_100 from OPIU where ID_POKAZ=180 and
DATEE=v_datte;
Select SUMMA into v_103 from OPIU where ID_POKAZ=30 and
DATEE=v_datte;
Select SUMMA into v_104 from OPIU where ID_POKAZ=110 and
DATEE=v_datte;

```

```

Koeff_obor_akt:=ROUND(((v_97+v_98)/AKT),3);
Koeff_obor_SK:=ROUND(((v_97+v_98)/SK),3);
Obor_inv_akt:=ROUND(((v_99+v_100)/(v_101+v_102)),3);
Obem_strakh_prem:=ROUND(((v_97+v_98)),3);
Obem_strakh_vypl:=ROUND(((v_103+v_104)),3);
Insert into CALC values (3062,Koeff_obor_akt,v_datte);
Insert into CALC values (3063,Koeff_obor_SK,v_datte);
Insert into CALC values (3064,Obor_inv_akt,v_datte);
Insert into CALC values (3065,Obem_strakh_prem,v_datte);
Insert into CALC values (3066,Obem_strakh_vypl,v_datte);
commit;
end Delo_aktivn;
/
Execute Delo_aktivn ('01.07.14');
/
Create view Aktiv_Passiv as
Select
DATEE,
min (decode(mn,3000,sm,Null)) A1,
min (decode(mn,3001,sm,Null)) A2,
min (decode(mn,3002,sm,Null)) A3,
min (decode(mn,3003,sm,Null)) A4,
min (decode(mn,3004,sm,Null)) P1,
min (decode(mn,3005,sm,Null)) P2,
min (decode(mn,3006,sm,Null)) P3,
min (decode(mn,3007,sm,Null)) P4,
min (decode(mn,3008,sm,Null)) A1_MINUS_P1,
min (decode(mn,3009,sm,Null)) A2_MINUS_P2,
min (decode(mn,3010,sm,Null)) A3_MINUS_P3
from
(Select CC.DATEE, P.ID_POKAZ mn, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P,CALC CC
where P.ID_VID=8 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc) GROUP BY DATEE ;
/
Select * from Aktiv_Passiv;
/
CREATE OR REPLACE TRIGGER povtoren
BEFORE INSERT ON CALC
for each row
BEGIN
IF :new.ZNACH=:old.ZNACH and :new.DATEE=:old.DATEE then
raise_application_error (-20040, 'Takie dannie uje est');
else Insert into CALC
values (:new.ID_POKAZ,:new.ZNACH,:new.DATEE);
END IF;
END;
/
Create view Delovaya_aktivnost as
Select
DATEE,
min (decode(mn,3062,sm,Null)) K_oborachivaemost_Aktivov,

```

```

min (decode(mn,3063,sm,Null)) K_oborachivaemost_SK,
min (decode(mn,3064,sm,Null)) K_oborachivaemost_Investiciy,
min (decode(mn,3065,sm,Null)) V_strakhovih_premiy,
min (decode(mn,3066,sm,Null)) V_strakhovih_viplat
from
(Select  CC.DATEE, P.ID_POKAZ mn, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P,CALC CC
where P.ID_VID=17 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc)  GROUP BY DATEE ;
/
Select * from Delovaya_aktivnost;
/
Select  P.NAME_POKAZ Nazvanie_pokazatelya,P.ID_POKAZ
Znachenie_pokazatelya, CC.DATEE DATE_svodky  from POKAZATEL
P,CALC CC
where P.ID_VID=17 and P.ID_POKAZ=CC.ID_POKAZ and
CC.DATEE='&p_datte' order by P.ID_POKAZ Asc
/
Create view Delovaya_aktivnost_glavnoe as
Select  P.NAME_POKAZ Nazvanie_pokazatelya,P.ID_POKAZ
Znachenie_pokazatelya, CC.DATEE DATE_svodky  from POKAZATEL
P,CALC CC
where P.ID_VID=17 and P.ID_POKAZ=CC.ID_POKAZ  order by
P.ID_POKAZ Asc
/
Create view Delovaya_aktivnost_Bugalter as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.14', sm,Null)) Iun_2014,
min (decode(mn, '31.03.14', sm,Null)) Mart_2014,
min (decode(mn, '31.12.13', sm,Null)) Dekabr_2013,
min (decode(mn, '01.10.13', sm,Null)) Oktyabr_2013,
min (decode(mn, '01.07.13', sm,Null)) Iun_2013
from
(Select  CC.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P,CALC CC
where P.ID_VID=17 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc)  GROUP BY NAME_POKAZ ;
/
Create view Rentabelnost_Bugalter as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.14', sm,Null)) "01.07.14",
min (decode(mn, '31.03.14', sm,Null)) "31.03.14",
min (decode(mn, '31.12.13', sm,Null)) "31.12.13",
min (decode(mn, '01.10.13', sm,Null)) "01.10.13",
min (decode(mn, '01.07.13', sm,Null)) "01.07.13"
from
(Select  CC.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P,CALC CC
where P.ID_VID=16 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc)  GROUP BY NAME_POKAZ ;

```



```

/
Create view Delovaya_aktivnost_Bugalter as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.14', sm, Null)) "01.07.14",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '01.07.13', sm, Null)) "01.07.13"
from
(Select CC.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P, CALC CC
where P.ID_VID=17 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc) GROUP BY NAME_POKAZ ;
/

Create view Koeff_likvidn_balansa_Bugalter as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.14', sm, Null)) "01.07.14",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '01.07.13', sm, Null)) "01.07.13"
from
(Select CC.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P, CALC CC
where P.ID_VID=13 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc) GROUP BY NAME_POKAZ ;
/

Create view Pokaz_fin_ustoich_Bugalter as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.14', sm, Null)) "01.07.14",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '01.07.13', sm, Null)) "01.07.13"
from
(Select CC.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P, CALC CC
where P.ID_VID=11 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc) GROUP BY NAME_POKAZ ;
/

Create view Otn_pokaz_fin_ust_Bugalter as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.14', sm, Null)) "01.07.14",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '01.07.13', sm, Null)) "01.07.13"
from

```

```

(Select  CC.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, CC.ZNACH sm
from POKAZATEL P,CALC CC
where P.ID_VID=12 and P.ID_POKAZ=CC.ID_POKAZ order by
P.ID_POKAZ Asc)  GROUP BY NAME_POKAZ ;
/
Create or replace view Balance_22_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.10.13', sm,Null)) "01.10.13",
min (decode(mn, '31.12.13', sm,Null)) "31.12.13"
from
(Select  B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P,Balance B
where (P.ID_VID=1 or P.ID_VID=2 or P.ID_VID=3 or P.ID_VID=4)
and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ Asc)
GROUP BY NAME_POKAZ;
/
Create or replace view Balance_Gorizont_analis22 as
Select Naimenovanie_pokazatelya, "01.10.13",
"31.12.13","31.12.13"-"01.10.13" Absolut_otklonen,
Round(((("31.12.13"-"01.10.13")*100/"01.10.13"),2)
Procent_otklonen from BALANCE_22_DATE Where "01.10.13"<>0;
/
Create or replace view Balance_23_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.13', sm,Null)) "01.07.13",
min (decode(mn, '01.10.13', sm,Null)) "01.10.13"
from
(Select  B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P,Balance B
where (P.ID_VID=1 or P.ID_VID=2 or P.ID_VID=3 or P.ID_VID=4)
and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ Asc)
GROUP BY NAME_POKAZ;
/
Create or replace view Balance_Gorizont_analis23 as
Select Naimenovanie_pokazatelya, "01.07.13",
"01.10.13","01.10.13"-"01.07.13" Absolut_otklonen,
Round(((("01.10.13"-"01.07.13")*100/"01.07.13"),2)
Procent_otklonen from BALANCE_23_DATE Where "01.07.13"<>0;
/
Create or replace view Aktivy_2_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.03.14', sm,Null)) "31.03.14",
min (decode(mn, '01.07.14', sm,Null)) "01.07.14"
from
(Select  B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P,Balance B
where P.ID_VID=1 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ,ID_POKAZ order by ID_POKAZ asc ;

```

```

/
Create or replace view Vertikl_Ativ_2_date as
Select Naimenovanie_pokazatelya, "31.03.14",
ROUND(("31.03.14"*100/(Select "31.03.14"
from AKTIVY_2_DATE where Naimenovanie_pokazatelya='ITOGO
aktivy')),2) Udeln_ves_aktiva, "01.07.14",
ROUND(("01.07.14"*100/(Select "01.07.14"
from AKTIVY_2_DATE where Naimenovanie_pokazatelya='ITOGO
aktivy')),2) Udeln_ves_aktiva FROM AKTIVY_2_DATE;
/
Create or replace view KapitalRezerv_2_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '01.07.14', sm, Null)) "01.07.14"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=2 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_KRezerv_2_date as
Select Naimenovanie_pokazatelya, "31.03.14",
ROUND(("31.03.14"*100/(Select "31.03.14"
from KapitalRezerv_2_date where
Naimenovanie_pokazatelya='ITOGO kapital i rezervy')),2)
Udeln_ves_aktiva, "01.07.14", ROUND(("01.07.14"*100/(Select
"01.07.14"
from KapitalRezerv_2_date where
Naimenovanie_pokazatelya='ITOGO kapital i rezervy')),2)
Udeln_ves_aktiva_1 FROM KapitalRezerv_2_date;
/
Create or replace view StrakhRezervy_2_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '01.07.14', sm, Null)) "01.07.14"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=3 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/

Create or replace view Vertikl_StrakhRezervy_2_date as
Select Naimenovanie_pokazatelya, "31.03.14",
ROUND(("31.03.14"*100/(Select "31.03.14"
from StrakhRezervy_2_date where
Naimenovanie_pokazatelya='ITOGO strakhovye rezervy')),2)

```

```

Udeln_ves_aktiva, "01.07.14", ROUND(("01.07.14"*100/(Select
"01.07.14"
from StrakhRezervy_2_date where
Naimenovanie_pokazatelya='ITOGO strakhovye rezervy')),2)
Udeln_ves_aktiva_1 FROM StrakhRezervy_2_date;
/
Create or replace view Obyazatelstva_2_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.03.14', sm, Null)) "31.03.14",
min (decode(mn, '01.07.14', sm, Null)) "01.07.14"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=4 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_Obyazatelstva_2_date as
Select Naimenovanie_pokazatelya, "31.03.14",
ROUND(("31.03.14"*100/(Select "31.03.14"
from Obyazatelstva_2_date where
Naimenovanie_pokazatelya='ITOGO obyazatelstva')),2)
Udeln_ves_aktiva, "01.07.14", ROUND(("01.07.14"*100/(Select
"01.07.14"
from Obyazatelstva_2_date where
Naimenovanie_pokazatelya='ITOGO obyazatelstva')),2)
Udeln_ves_aktiva_1 FROM Obyazatelstva_2_date;
/
Create or replace view Aktivy_21_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=1 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_Ativ_21_date as
Select Naimenovanie_pokazatelya, "31.12.13",
ROUND(("31.03.14"*100/(Select "31.12.13"
from AKTIVY_21_DATE where Naimenovanie_pokazatelya='ITOGO
aktivy')),2) Udeln_ves_aktiva, "31.03.14",
ROUND(("31.03.14"*100/(Select "31.03.14"
from AKTIVY_21_DATE where Naimenovanie_pokazatelya='ITOGO
aktivy')),2) Udeln_ves_aktiva_1 FROM AKTIVY_21_DATE;
/
Create or replace view KapitalRezerv_21_date as
Select

```

```

NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=2 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_KRezerv_21_date as
Select Naimenovanie_pokazatelya, "31.12.13",
ROUND(("31.12.13"*100/(Select "31.12.13"
from KapitalRezerv_21_date where
Naimenovanie_pokazatelya='ITOGO kapital i rezervy')),2)
Udeln_ves_aktiva, "31.03.14", ROUND(("31.03.14"*100/(Select
"31.03.14"
from KapitalRezerv_21_date where
Naimenovanie_pokazatelya='ITOGO kapital i rezervy')),2)
Udeln_ves_aktiva_1 FROM KapitalRezerv_21_date;
/
Create or replace view StrakhRezervy_21_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=3 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_StrakhRezervy_21_date as
Select Naimenovanie_pokazatelya, "31.12.13",
ROUND(("31.12.13"*100/(Select "31.12.13"
from StrakhRezervy_21_date where
Naimenovanie_pokazatelya='ITOGO strakhovye rezervy')),2)
Udeln_ves_aktiva, "31.03.14", ROUND(("31.03.14"*100/(Select
"31.03.14"
from StrakhRezervy_21_date where
Naimenovanie_pokazatelya='ITOGO strakhovye rezervy')),2)
Udeln_ves_aktiva_1 FROM StrakhRezervy_21_date;
/
Create or replace view Obyazatelstva_21_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '31.12.13', sm, Null)) "31.12.13",
min (decode(mn, '31.03.14', sm, Null)) "31.03.14"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B

```

```

where P.ID_VID=4 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ,ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_Obyazatelstva_21_date as
Select Naimenovanie_pokazatelya, "31.12.13",
ROUND(("31.12.13"*100/(Select "31.12.13"
from Obyazatelstva_21_date where
Naimenovanie_pokazatelya='ITOGO obyazatelstva')),2)
Udeln_ves_aktiva, "31.03.14", ROUND(("31.03.14"*100/(Select
"31.03.14"
from Obyazatelstva_21_date where
Naimenovanie_pokazatelya='ITOGO obyazatelstva')),2)
Udeln_ves_aktiva_1 FROM Obyazatelstva_21_date;
/
Create or replace view Aktivny_22_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P,Balance B
where P.ID_VID=1 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ,ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_Aktiv_22_date as
Select Naimenovanie_pokazatelya, "01.10.13",
ROUND(("01.10.13"*100/(Select "01.10.13"
from AKTIVY_22_DATE where Naimenovanie_pokazatelya='ITOGO
aktivny')),2) Udeln_ves_aktiva, "31.12.13",
ROUND(("31.12.13"*100/(Select "31.12.13"
from AKTIVY_22_DATE where Naimenovanie_pokazatelya='ITOGO
aktivny')),2) Udeln_ves_aktiva_1 FROM AKTIVY_22_DATE;
/
Create or replace view KapitalRezerv_22_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P,Balance B
where P.ID_VID=2 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ,ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_KRezerv_22_date as
Select Naimenovanie_pokazatelya, "01.10.13",
ROUND(("01.10.13"*100/(Select "01.10.13"

```

```

from KapitalRezerv_22_date where
Naimenovanie_pokazatelya='ITOGO kapital i rezervy')),2)
Udeln_ves_aktiva, "31.12.13", ROUND(("31.12.13"*100/(Select
"31.12.13"
from KapitalRezerv_22_date where
Naimenovanie_pokazatelya='ITOGO kapital i rezervy')),2)
Udeln_ves_aktiva_1 FROM KapitalRezerv_22_date;
/
Create or replace view StrakhRezervy_22_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=3 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_StrakhRezervy_22_date as
Select Naimenovanie_pokazatelya, "01.10.13",
ROUND(("01.10.13"*100/(Select "01.10.13"
from StrakhRezervy_22_date where
Naimenovanie_pokazatelya='ITOGO strakhovye rezervy')),2)
Udeln_ves_aktiva, "31.12.13", ROUND(("31.12.13"*100/(Select
"31.12.13"
from StrakhRezervy_22_date where
Naimenovanie_pokazatelya='ITOGO strakhovye rezervy')),2)
Udeln_ves_aktiva_1 FROM StrakhRezervy_22_date;
/
Create or replace view Obyazatelstva_22_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=4 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_Obyazatelstva_22_date as
Select Naimenovanie_pokazatelya, "01.10.13",
ROUND(("01.10.13"*100/(Select "01.10.13"
from Obyazatelstva_22_date where
Naimenovanie_pokazatelya='ITOGO obyazatelstva')),2)
Udeln_ves_aktiva, "31.12.13", ROUND(("31.12.13"*100/(Select
"31.12.13"

```

```

from Obyazatelstva_22_date where
Naimenovanie_pokazatelya='ITOGO obyazatelstva')),2)
Udeln_ves_aktiva_1 FROM Obyazatelstva_22_date;
/
Create or replace view Aktivy_23_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.13', sm, Null)) "01.07.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=1 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_Ativ_23_date as
Select Naimenovanie_pokazatelya, "01.07.13",
ROUND(("01.07.13"*100/(Select "01.07.13"
from AKTIVY_23_DATE where Naimenovanie_pokazatelya='ITOGO
aktivy')),2) Udeln_ves_aktiva, "01.10.13",
ROUND(("01.10.13"*100/(Select "01.10.13"
from AKTIVY_23_DATE where Naimenovanie_pokazatelya='ITOGO
aktivy')),2) Udeln_ves_aktiva_1 FROM AKTIVY_23_DATE;
/
Create or replace view KapitalRezerv_23_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.13', sm, Null)) "01.07.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=2 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_KRezerv_23_date as
Select Naimenovanie_pokazatelya, "01.07.13",
ROUND(("01.07.13"*100/(Select "01.07.13"
from KapitalRezerv_23_date where
Naimenovanie_pokazatelya='ITOGO kapital i rezervy')),2)
Udeln_ves_aktiva, "01.10.13", ROUND(("01.10.13"*100/(Select
"01.10.13"
from KapitalRezerv_23_date where
Naimenovanie_pokazatelya='ITOGO kapital i rezervy')),2)
Udeln_ves_aktiva_1 FROM KapitalRezerv_23_date;
/
Create or replace view StrakhRezervy_23_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.13', sm, Null)) "01.07.13",

```



```

min (decode(mn, '01.10.13', sm, Null)) "01.10.13"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=3 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_StrakhRezervy_23_date as
Select Naimenovanie_pokazatelya, "01.07.13",
ROUND(("01.07.13"*100/(Select "01.07.13"
from StrakhRezervy_23_date where
Naimenovanie_pokazatelya='ITOGO strakhovye rezervy')),2)
Udeln_ves_aktiva, "01.10.13", ROUND(("01.10.13"*100/(Select
"01.10.13"
from StrakhRezervy_23_date where
Naimenovanie_pokazatelya='ITOGO strakhovye rezervy')),2)
Udeln_ves_aktiva_1 FROM StrakhRezervy_23_date;
/
Create or replace view Obyazatelstva_23_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.07.13', sm, Null)) "01.07.13",
min (decode(mn, '01.10.13', sm, Null)) "01.10.13"
from
(Select B.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, B.SUMMA sm from
POKAZATEL P, Balance B
where P.ID_VID=4 and P.ID_POKAZ=B.ID_POKAZ order by P.ID_POKAZ
desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;
/
Create or replace view Vertikl_Obyazatelstva_23_date as
Select Naimenovanie_pokazatelya, "01.07.13",
ROUND(("01.07.13"*100/(Select "01.07.13"
from Obyazatelstva_23_date where
Naimenovanie_pokazatelya='ITOGO obyazatelstva')),2)
Udeln_ves_aktiva, "01.10.13", ROUND(("01.10.13"*100/(Select
"01.10.13"
from Obyazatelstva_23_date where
Naimenovanie_pokazatelya='ITOGO obyazatelstva')),2)
Udeln_ves_aktiva_1 FROM Obyazatelstva_23_date;
Create or replace view OPIU_22_date as
Select
NAME_POKAZ Naimenovanie_pokazatelya,
min (decode(mn, '01.10.13', sm, Null)) "01.10.13",
min (decode(mn, '31.12.13', sm, Null)) "31.12.13"
from
(Select OP.DATEE mn, P.ID_POKAZ, P.NAME_POKAZ, OP.SUMMA sm
from POKAZATEL P, OPIU OP
where (P.ID_VID=5 or P.ID_VID=6 or P.ID_VID=7) and
P.ID_POKAZ=OP.ID_POKAZ order by P.ID_POKAZ desc)
GROUP BY NAME_POKAZ, ID_POKAZ order by ID_POKAZ asc ;

```