

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра \_\_\_\_\_

«Допущен к защите»  
Заведующий кафедрой КТ  
Курсалиев З. К., доцент, проф  
(Ф.И.О., ученая степень, звание)  
Ку «19» 05 2014 г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Обеспечение безопасности СУБД Oracle на примере базы данных Автоматизация проведения соревнований по фигурному катанию»

Специальность 5В070400 - Вычислительная техника и программное обеспечение

Выполнил (а) Байтенова А.А. ВТ-10-05  
(Фамилия и инициалы) группа

Научный руководитель Сабитова Б.Т., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Зрешева З.Д., с.с. преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
Зрешева «06» 05 2014 г.  
(подпись)

по безопасности жизнедеятельности:

Дриков И.Г., д.х.н., профессор  
(Фамилия и инициалы, ученая степень, звание)  
Дриков «24» 04 2014 г.  
(подпись)

по применению вычислительной техники:

Сабитова Б.Т., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.  
(подпись)

\_\_\_\_\_  
(Фамилия и инициалы, ученая степень, звание)  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.  
(подпись)

Нормоконтролер: Тучков Д.М.  
(Фамилия и инициалы, ученая степень, звание)  
Тучков «19» 05 2014 г.  
(подпись)

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

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет «Информационные технологии»  
Специальность «Вычислительная техника и программное обеспечение»  
Кафедра «Компьютерные технологии»

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Байтлечова Асель Асхарқызы  
(фамилия, имя, отчество)  
Тема проекта «Обеспечение безопасности СУБД Oracle на примере  
базы данных «Автоматизация проведения соревнований по фи-  
гурному катанию»

утверждена приказом ректора № 115 от «24» сентября 2013 г.  
Срок сдачи законченной работы «3» июня 2014 г.  
Исходные данные к проекту требуемые параметры результатов  
проектирования (исследования) и исходные данные объекта  
Книги по разработке баз данных, документация Oracle  
по обеспечению безопасности

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

1. проектирование базы данных
2. разработка клиентского приложения
3. обеспечение безопасности
4. Технико-экономическое обоснование
5. безопасность информации в сети





В данной дипломной работе согласно заданию была осуществлена безопасность СУБД Oracle на примере базы данных, позволяющей автоматизировать и значительно упростить проведения соревнований по фигурному катанию. Для обеспечения защиты базы данных было применено прозрачное шифрование TDE продукта Oracle Advanced Security, подключен протокол аутентификации Radius, обеспечен аудит базы данных. Данный программный комплекс может быть адаптирован на любом соревновании по фигурному катанию. Произведен расчет затрат на научно-исследовательскую работу. Рассмотрены вопросы безопасности жизнедеятельности, а именно расчет кондиционирования помещения и уровня шума.

### **АНДАТПА**

Бұл дипломдық жұмыста берілген міндеттерге сәйкес мәнерлеп сырғанау бойынша жарыстарды өткізуді автоматтандыруға және жеңілдетуге мүмкіндік беретін деректер базасының негізінде дерекқорлармен басқармау жүйесі Oracle қауіпсіздігі жүзеге асырылды. Деректер базасын қорғауды қамтамасыз ету үшін Oracle Advanced Security өнімінің TDE айқынды шифрлануы қолданылды, Radius сәйкестендіру хаттамасы енгізілді, деректер базасының аудиті қамтамасыз етілді. Осы бағдарламалық кешен мәнерлеп сырғанау бойынша кез келген жарыстарда жаттықтырылуы мүмкін. Ғылыми-зерттеу жұмысының шығындар есебі жүргізілді. Өміртіршілік қауіпсіздігі, соның ішінде бөлмелерді желдету мен айғай-шудың деңгейді есептеу мәселелері қарастырылды.

### **ANNOTATION**

In this graduating project, according to the task the securing a management system by Oracle database was carried, that gives an opportunity to automate and greatly simplify the competitions in figure skating. In order to provide the database security transparent data encryption of Oracle Advanced Security was applied, authentication protocol Radius was connected and the database auditing was provided. This software package can be adapted to any figure skating competition. Also the cost of research work was calculated. Life safety issues, namely payment and room conditioning noise were considered.

Введение	7
1 Выбор метода обеспечения безопасности	9
1.1 Краткий обзор средств шифрования Oracle Database	9
1.2 Прозрачное шифрование данных	11
1.2.1 Краткое описание управления ключами шифрования	12
1.2.2 Этапы внедрения	12
1.3 Протокол аутентификации Radius	15
2 Обеспечение безопасности СУБД Oracle	20
2.1 Прозрачное шифрование данных	20
2.1.1 Определение местоположения бумажника	22
2.1.2 Создание бумажника	22
2.1.3 Вопросы производительности	24
2.2 Настройка Radius	24
2.3 Аудит базы данных	28
3 Разработка базы данных	37
3.1 Постановка задачи	37
3.2 Анализ предметной области	38
3.3 Концептуальное проектирование	40
3.4 Логическое проектирование	41
3.4.1 Создание пользователей, логинов и задание им паролей	42
3.4.2 Схема отношений, составленные на языке определения данных ( ddl, data definition language)	43
3.5 Физическое проектирование	48
4 Создание клиентского приложения	50
5 Безопасность жизнедеятельности	54
5.1 Анализ потенциально опасных и вредных факторов, воздействующих на рабу технического оператора в процессе обслуживания и работы с системой	54
5.2 Расчетная часть	61
5.2 Вывод	66
6 Техничко-экономическое обоснование	67
6.1 Расчет затрат на разработку информационных технологий	67
6.2 Расчет цены программного продукта	75
6.3 Вывод	77
Заключение	78
Список использованной литературы	79
Приложение А	80
Приложение Б	82
Приложение В	85
Приложение Г	87
Приложение Д	88

## Введение

Любая деятельность в современном мире включает в себя множество задач по обеспечению безопасности и соблюдению законодательных норм.

В последние годы было множество случаев хищения персональных данных и мошеннических операций с использованием информации кредитных карт, приведших к убыткам в размере десятков миллионов долларов. Защита от таких видов угроз требует эффективных решений по безопасности. Университеты и медицинские организации повысили уровень безопасности данных, используемых для идентификации личности (PII), таких как номера социального страхования. Компании розничной торговли в настоящее время работают над обеспечением соответствия своих информационных систем требованиям PCI-DSS [1].

Самый большой кошмар любой организации: кто-то украл ленты с резервной копией вашей базы данных. Несомненно, была построена защищенная система, зашифрованы наиболее конфиденциальные ресурсы, размещены серверы баз данных за межсетевыми экранами. Но вор выбрал доступный ему способ: он взял ленты с резервной копией, чтобы, вероятно, скопировать базу данных на другом сервере, запустить экземпляр сервера этой базы данных, а затем не спеша просмотреть все ваши данные. Защита содержимого базы данных от такого воровства представляет собой не только хорошую практику; это – требование многих нормативно-правовых и нормативно-технических документов.

Целью данной дипломной работы является обеспечение безопасности системы управления базами данных на примере базы данных «Автоматизация проведения соревнований по фигурному катанию». Для обеспечения многоуровневой безопасности, как на уровне базы, так и на уровне сервера будут применены такие компоненты как Прозрачное шифрование данных (TDE), протокол аутентификации Radius, аудит базы данных.

База данных, в свою очередь, должна предусматривать отражение категорий фигуристов, тренеров, судей, имеющих в базе данных, каждое место (призовое и не призовое) фигуриста в рамках данного соревнования, а также учитывать информацию о том, каким судьей, какому фигуристу, какая оценка была поставлена за ту или иную программу. Информация об оценках в данной базе данных предназначена для самостоятельного просмотра фигуристом, тренером или судьей, но она также исключает возможность модификации, обновления или удаления данных ими. Таким образом, вводится необходимость разграничения прав доступа.

Для достижения поставленной цели база данных «Автоматизация проведения соревнований по фигурному катанию» должна предоставлять следующие возможности:

- хранение информации о фигуристах;
- хранение информации о тренерах;

- хранение информации о хореографах;
- хранение информации о судьях;
- хранение информации о технической части проведения соревнований;
- учет и автоматизация оценок фигуристов.

## **1 Выбор метода обеспечения безопасности**



## 1.1 Краткий обзор средств шифрования Oracle Database

Шифрование данных является ключевым компонентом при реализации принципа глубокой многоуровневой защиты, а также важным элементом защиты данных во время передачи и хранения. Впервые Oracle представил базу данных с программным интерфейсом (API) для шифрования данных в Oracle8i. На данный момент многие клиенты используют интерфейсы шифрования в базе данных Oracle для повышения безопасности конфиденциальных данных приложений. Достижение прозрачности чтения/записи зашифрованных данных посредством использования крипто-API, требует внедрения функции вызовов внутри самого приложения или использования предустановленного триггера БД [1].

Представления данных приложения также могут нуждаться в расшифровывании, прежде чем они будут переданы в приложение. Кроме того, управление ключами шифрования должно производиться программным путем.

Oracle Advanced Security Transparent Data Encryption (TDE), впервые представленная в Oracle Database 10g Release 2, является самым передовым решением в области шифрования. TDE обеспечивает встроенное управление ключами шифрования и полную прозрачность шифрования конфиденциальных данных. Используемый при этом механизм шифрования баз данных основан на использовании команд DDL, полностью исключая необходимость в изменении приложений, создании программных средств управления ключами шифрования, триггеров и представлений в базах данных.

Краткое описание Oracle Database Encryption представлено в таблице 1.

Т а б л и ц а 1 - Краткое описание Oracle Database Encryption

Комплекс функций	Набор программ DBMS Obfuscation (Oracle 8i и выше) SE & EE	DBMS CRYPTO (Oracle database 10g R1 и выше) SE & EE	Oracle advanced security transparent data encryption EE Only Option
Алгоритмы шифрования	DES, 3DES	DES, DES, AES, RC4, 3DES_2KEY(1)	3DES, AES (128, 192, и 256 бит)
Дополнительные формы	Не поддерживается	PKCS5, нули	PKCS5(2)
Режимы соединения на основе блочного шифра	CBC	CBC, CFB, ECB, OFB	CBC(2)

*Окончание таблицы 1*

Комплекс функций	Набор программ DBMS Obfuscation (Oracle 8i и выше) SE & EE	DBMS CRYPTO (Oracle database 10g R1 и выше) SE & EE	Oracle advanced security transparent data encryption EE Only Option
Режимы соединения на основе блочного шифра	CBC	CBC, CFB, ECB, OFB	CBC(2)
Шифровальные хэш-алгоритмы	MD5	SHA-1, MD4(1), MD5(1)	SHA-1(2)
Хэш-алгоритмы по ключу (MAC)	none supported	HMAC_MD5, HMAC_SH1	Не применяются
Шифровальный псевдослучайный генератор чисел	RAW, VARCHAR2	RAW, NUMBER, BINARY_INTEGER	Не применяется
Типы баз данных	RAW, VARCHAR2	RAW, CLOB, BLOB	Все кроме: OBJ., ADT, LOB

Oracle Advanced Security (OAS) обеспечивает секретность и конфиденциальность информации.

Oracle Advanced Security состоит из трех основных частей:

- шифрование трафика от клиента к серверу приложений (SSL);
- шифрование трафика от сервера приложений к базе данных (SQL Net);
- прозрачное шифрование данных на диске (Transparent Data Encryption)

[2].

Все соединения СУБД Oracle могут быть зашифрованы с помощью OAS.

Oracle Advanced Security реализует требования стандарта ISO 27001 в п. А.10.8.4, А.10.9.2, А.11.4.2, А.12.3.

OAS позволяет защищать все входящие и исходящие соединения СУБД Oracle. Для каждого соединения создается секретный ключ, обеспечивающий безопасность всего сетевого трафика. OAS делает невозможным скрытую модификацию, добавление или удаление части передаваемых данных [1].

Поддерживаются следующие алгоритмы шифрования и обеспечения целостности данных:

- AES (128, 192, 256).
- RC4 (40, 56, 128, 256).
- 3DES (2 и 3 ключа).

- MD5.
- SHA1.

Кроме того, в случае поддержки в информационной системе инфраструктуры открытого ключа PKI доступно SSL-шифрование. Простая конфигурация позволяет не изменять приложение. Включение шифрования и обеспечения целостности данных производится путем настройки сетевых конфигураций на стороне сервера и клиента. Т. к. нет необходимости менять приложение, эти способы обеспечения безопасности доступны практически для всех, а их конфигурирование и использование являются очень простыми.

Строгая аутентификация для Oracle Database 11g OAS обеспечивает возможность для организации использовать существующую инфраструктуру безопасности, например, Kerberos, PKI, RADIUS для осуществления строгой аутентификации в СУБД Oracle 11g. Связь протоколов аутентификации и Oracle Advanced Security представлена на рисунке 1.

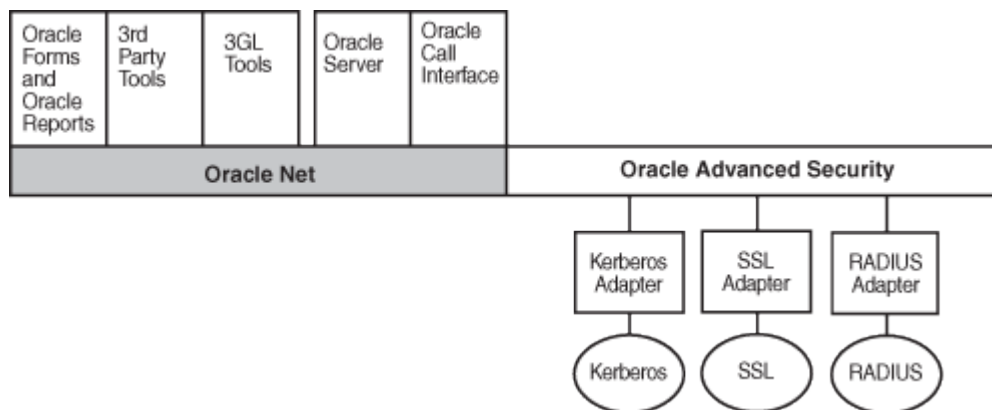


Рисунок 1 - Oracle Net Service с протоколами аутентификации

## 1.2 Прозрачное шифрование данных

Средствами TDE обеспечивается шифрование данных перед записью на диск и расшифровывание данных, прежде чем они возвращаются в приложение. Процесс зашифровывания и расшифровывания выполняется на уровне SQL и полностью прозрачен для прикладных программ и пользователей. Резервные копии баз данных, записанные на диск или магнитную ленту, будут содержать эти данные в зашифрованном виде. TDE, при необходимости, может быть использовано в сочетании с Oracle RMAN для зашифровывания всей СУБД Oracle в ходе резервирования на диски.

Преимущества прозрачного шифрования данных

- встроенное управление ключами шифрования;
- прозрачное шифрование защищаемых данных (по столбцам) в прикладных программах;
- прозрачное шифрование табличных пространств (Впервые в 11g);
- прозрачное шифрование файлов/LOBs (Впервые в 11g);

– интеграция аппаратного модуля безопасности (HSM) (Впервые в 11g).

### 1.2.1 Краткое описание управления ключами шифрования

TDE автоматически создает ключ шифрования, когда проводится зашифровывание данных столбца в таблице базы данных. Ключ шифрования - уникальный для каждой таблицы. Если в таблице зашифровывается более одного столбца, то для каждого из столбцов используется один и тот же ключ шифрования. Ключи шифрования для таблиц сохраняются в справочнике Oracle и зашифровываются при помощи первичного ключа (мастер-ключа) шифрования TDE. Первичный ключ шифрования сохраняется вне базы данных в «тубусе для ключей» Oracle Wallet (файл формата PKCS#12), который зашифрован с помощью пароля, определяемого администратором безопасности или DBA в процессе создания [1].

Новым в Oracle Database 11g Advanced Security является возможность сохранять первичный ключ в устройстве HSM, используя PKCS#11 интерфейс.

Краткое описание прозрачного шифрования данных представлено на рисунке 2.

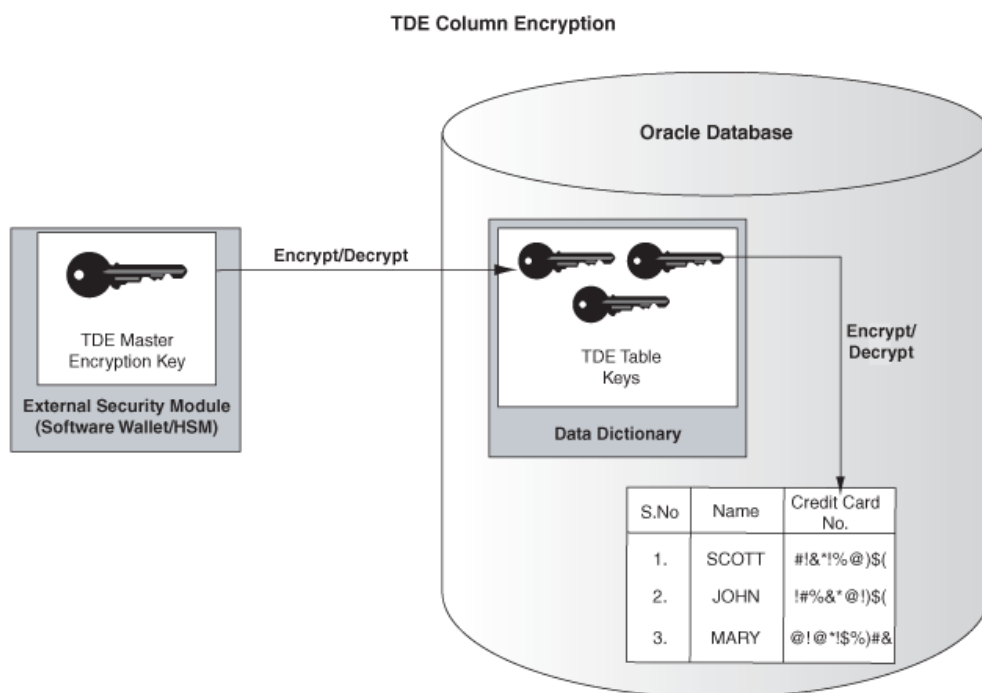


Рисунок 2 - Краткое описание прозрачного шифрования данных

### 1.2.2 Этапы внедрения

В связи с тем, что TDE прозрачно для существующих приложений (не требуются создавать триггеры и представления баз данных), процесс шифрования стал проще по сравнению с традиционными решениями на основе

использования API. Следующие шаги могут быть использованы для применения TDE:

- 1) инициализация мастер-ключа;
- 2) определение данных для зашифровывания (PII данные, кредитные карты);
- 3) проверка: поддерживает ли TDE выбранный тип данных, а также, не используются ли данные колонки в качестве внешних ключей;
- 4) зашифровывание конфиденциальных данных, используя TDE.

### **Инициализация мастер-ключа**

Мастер-ключ для каждой базы данных свой. Несмотря на это, любой мастер-ключ может быть скопирован во вторичную базу данных, если ранее он не был там использован. Прежде чем, зашифровывать данные таблицы, необходимо создать мастер-ключ. Для этого существует следующая команда:

```
SQL> alter system set key identified by "password";
```

Эта команда создает «тубус для ключей» (Oracle Wallet) и использует указанный в команде пароль для зашифровывания в соответствии со стандартом PKCS#5. Oracle Wallet сохраняет историю уже использованных мастерключей и делает их доступными, когда данные, зашифрованные старым ключом, считываются с резервных носителей.

До того как база данных сможет расшифровать табличные ключи для зашифровывания и расшифровывания прикладных данных, необходимо открыть «тубус для ключей» Wallet, который содержит первичный ключ шифрования. Конечно, можно запустить базу данных и работать в ней без открытия Wallet, но при попытке получить доступ к зашифрованным данным база данных выдаст сообщение об ошибке. Закрывать Wallet (и ограничивать тем самым доступ к данным) целесообразно только в ходе работ по техническому обслуживанию, когда доступ к базе данных разрешен специалисту, занимающемуся поддержкой.

Мастер-ключ может быть изменен при повторном запуске команды “alter system”.

```
SQL> alter system set key identified by "password";
```

При изменении мастер-ключа все табличные ключи в справочнике Oracle будут обновлены (расшифрованы и зашифрованы вновь). Стандарт безопасности данных PCI (DSS) 1.1 требует «обновлять ключи шифрования, не реже одного раза в год». Изменение мастер-ключа приведет к обновлению зашифрованных ключей столбцов, используя новый мастер-ключ, а зашифрованные данные в таблицах останутся без изменений.

Пароль Wallet может быть изменен независимо от мастер-ключа шифрования, он используется исключительно для зашифровывания данных

файла Wallet на диске. Для этого можно воспользоваться программой Oracle Wallet Manager или утилитой 'orapki'.

### **Определение конфиденциальных данных**

Процесс поиска конфиденциальных данных, таких как номера социального страхования и кредитных карт, может оказаться затруднительным, особенно в комплексных приложениях. Единственный метод, которым можно воспользоваться, – это поиск в справочнике Oracle по названию столбца и типу данных, которые часто используются для хранения подобной информации.

```
SQL> select column_name, table_name, data_type from dba_tab_cols
where column_name like '%SOCIAL%' or column_name like '%SSN%' or
column_name like '%SECNUM%' or column_name like "%SOC%" and
owner='<owner>';
```

### **Типы данных, поддерживаемых TDE**

TDE поддерживает наиболее распространенные типы данных. Они представлены в таблице 2.

Т а б л и ц а 2 - Типы данных, поддерживаемых TDE

VARCHAR2	CHAR	DATE
NUMBER	NVARCHAR2	NCHAR
RAW	RAW	SECUREFILES (LOBS)
BINARY_DOUBLE	BINARY_FLOAT	

TDE не может быть использован для зашифровывания столбцов, которые используются во внешнем ключе. Для того чтобы проверить, является ли колонка частью внешнего ключа, изучаются данные из справочника Oracle.

### **Зашифровывание данных с помощью TDE**

Чтобы зашифровать существующие столбцы в таблице:

```
SQL> alter table figure_skater modify ( fs_name encrypt);
```

В ходе выполнения транзакции по зашифровыванию поддерживается последовательное чтение данных.

Транзакции DML (вставить, обновить, удалить), выполняемые в течение транзакции зашифровывания, будут запрашивать 'изменения online'.

Transparent Data Encryption поддерживает работу с индексами, минимизируя дополнительный поиск в зашифрованном столбце.

Когда столбец с индексами готов к зашифровыванию, сначала рекомендуется удалить существующие индексы, зашифровать столбец, а затем заново создать индексы.

### **Смена ключа таблицы/столбца**

При выполнении команды "alter table" ключ таблицы или столбца, размер ключа и алгоритм могут быть независимо изменены:

```
SQL> ALTER TABLE judge REKEY;  
SQL> ALTER TABLE judge REKEY USING 'AES256';  
SQL> ALTER TABLE judge ENCRYPT USING 'AES128';
```

Изменение ключа таблицы или столбца приведет к повторению шифрования данных в таблице.

За время существования таблицы, ее данные могут фрагментироваться, расположиться в ином порядке, разделяться по категориям, увеличиваться в количестве и перемещаться внутри табличного пространства; это может привести к возникновению неиспользуемых блоков со старыми, недоступными копиями данных внутри файла базы данных. Когда шифруется существующий столбец, шифруется только его самая последняя "действующая" копия, при этом могут оставаться данные в незашифрованном виде в «старых» копиях данных столбца.

Для минимизации риска оставить конфиденциальные данные в незашифрованном виде при шифровании Oracle рекомендует создавать новое табличное пространство, перемещать таблицы приложений в него, и удалять старое табличное пространство.

## **1.3 Протокол аутентификации RADIUS**

Oracle Advanced Security снабжен протоколом RADIUS, который позволяет Oracle DataBase 11 соблюдать аутентификацию и полномочия, утвержденные на сервере RADIUS. Эта функция особенно полезна для фирм, которые заинтересованы в двухфакторной аутентификации, которая устанавливает личность на основе того, что она знает (пароль или ПИН – код) и том, чем она владеет (карту с переменным паролем), предоставленную неким производителем карт с переменным паролем. RADIUS (RFC #2138) это распространенная система, которая обеспечивает удаленный доступ к сетевым службам и давно признанна как отраслевой стандарт для удаленного и контролируемого доступа к сети. Полномочия пользователя в RADIUS и информация доступа определены на сервере RADIUS для того, чтобы дать возможность этому внешнему серверу выполнить аутентификацию, авторизацию и учет, когда это требуется [1].

RADIUS используется как протокол AAA:

- англ. *Authentication* - процесс, позволяющий аутентифицировать (проверить подлинность) субъекта по его идентификационным данным, например, по логину (имя пользователя, номер телефона и т. д.) и паролю;

- англ. *Authorization* - процесс, определяющий полномочия идентифицированного субъекта на доступ к определённым объектам или сервисам;

- англ. *Accounting* - процесс, позволяющий вести сбор сведений (учётных данных) об использованных ресурсах. Первичными данными (то есть, традиционно передаваемых по протоколу RADIUS) являются величины входящего и исходящего трафиков: в байтах/октетах (с недавних пор в гигабайтах). Однако протокол предусматривает передачу данных любого типа, что реализуется посредством VSA (Vendor Specific Attributes).

Работа Radius в среде Oracle показана на рисунке 3.

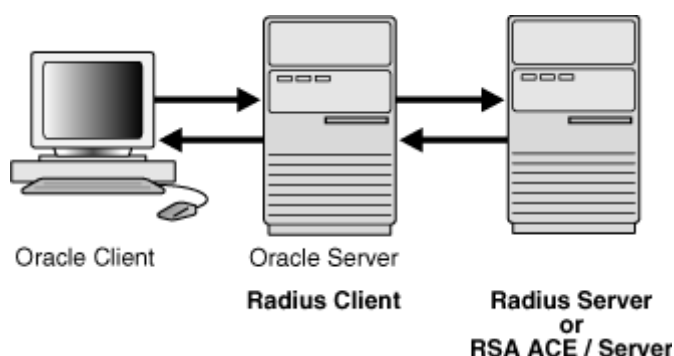


Рисунок 3 - Работа Radius в среде Oracle

Поддержка RADIUS в Oracle - это реализация протоколов клиента RADIUS (RADIUS Client), которые дают возможность базе данных предоставить пользователям RADIUS аутентификацию, авторизацию и службы учета. Они посылают запрос аутентификации на сервер RADIUS и действуют в соответствии с откликами сервера. Аутентификация может произойти в синхронном или асинхронном режиме аутентификации и является частью конфигурации Oracle для поддержки RADIUS [2].

Синхронный режим аутентификации показан на рисунке 4 и проходит по следующей последовательности:

1) пользователь авторизуется посредством кода (пароля) или других идентификационных данных. Клиентская система отправляет эти данные на Сервер Oracle;

2) сервер Oracle, выступая в роли RADIUS client передает данные с клиента Oracle на Сервер Radius;

3) сервер Radius, в свою очередь, передает данные на соответствующий аутентификационный сервер, например Smart Card или SecurID ACE для подтверждения;

4) аутентификационный сервер отправляет или разрешающее, или запрещающее доступ письмо обратно, на Сервер Radius;



- 5) сервер Radius перенаправляет этот ответ на Сервер Oracle/RADIUS client;
- 6) сервер Oracle/RADIUS client перенаправляет ответ обратно, клиенту Oracle.

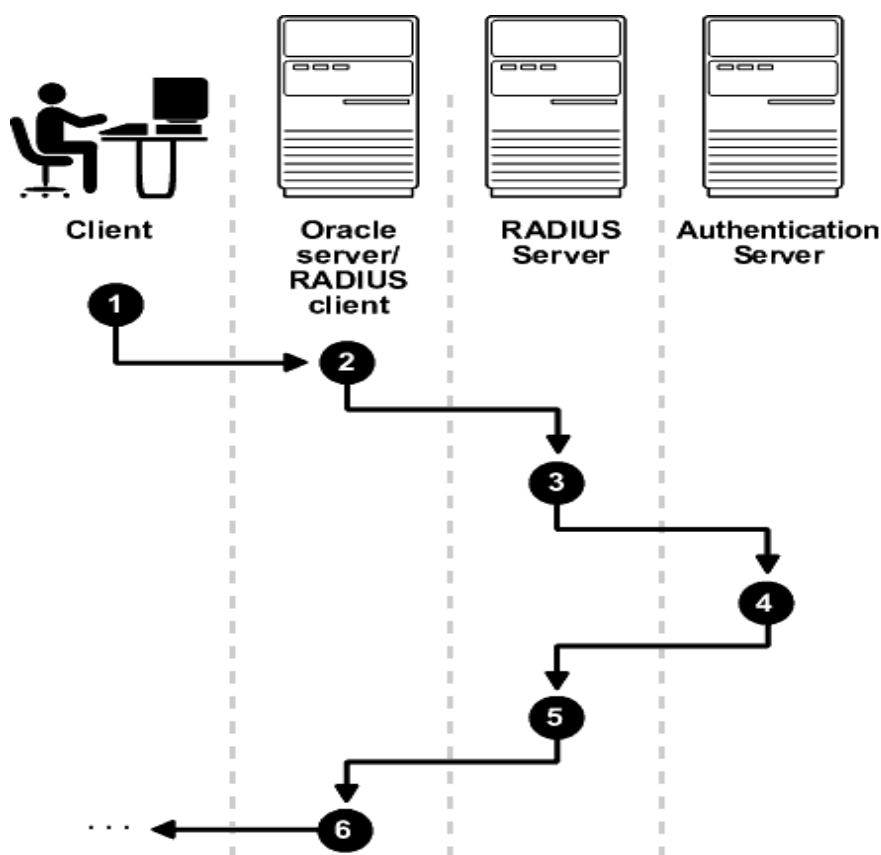


Рисунок 4 - Последовательность синхронной аутентификации

Асинхронный режим аутентификации показан на рисунке 5 и проходит по следующей последовательности:

- 1) пользователь отправляет запрос на вход. Клиентская система отправляет эти данные на Сервер Oracle;
- 2) сервер Oracle, выступая в роли RADIUS client передает данные с клиента Oracle на Сервер Radius;
- 3) сервер Radius, в свою очередь, передает данные на соответствующий аутентификационный сервер, например Smart Card или SecurID ACE;
- 4) аутентификационный сервер отправляет запрос на сервер Radius, который представляет собой случайный набор чисел;
- 5) сервер Radius отправляет данный запрос на Сервер Oracle/RADIUS client;
- 6) сервер Oracle/RADIUS client, в свою очередь, отправляет его клиенту Oracle;
- 7) пользователь должен ввести данный запрос в предложенное поле, далее клиент Oracle отправит данный запрос на сервер Oracle/RADIUS client;

- 8) сервер Oracle/RADIUS client отправляет ответ пользователя на сервер Radius;
- 9) сервер RADIUS отправляет ответ пользователя на аутентификационный сервер для подтверждения;
- 10) аутентификационный сервер отправляет или разрешающее, или запрещающее доступ письмо обратно, на Сервер Radius;
- 11) сервер Radius перенаправляет этот ответ на Сервер Oracle/RADIUS client;
- 12) сервер Oracle/RADIUS client перенаправляет ответ обратно, клиенту Oracle.

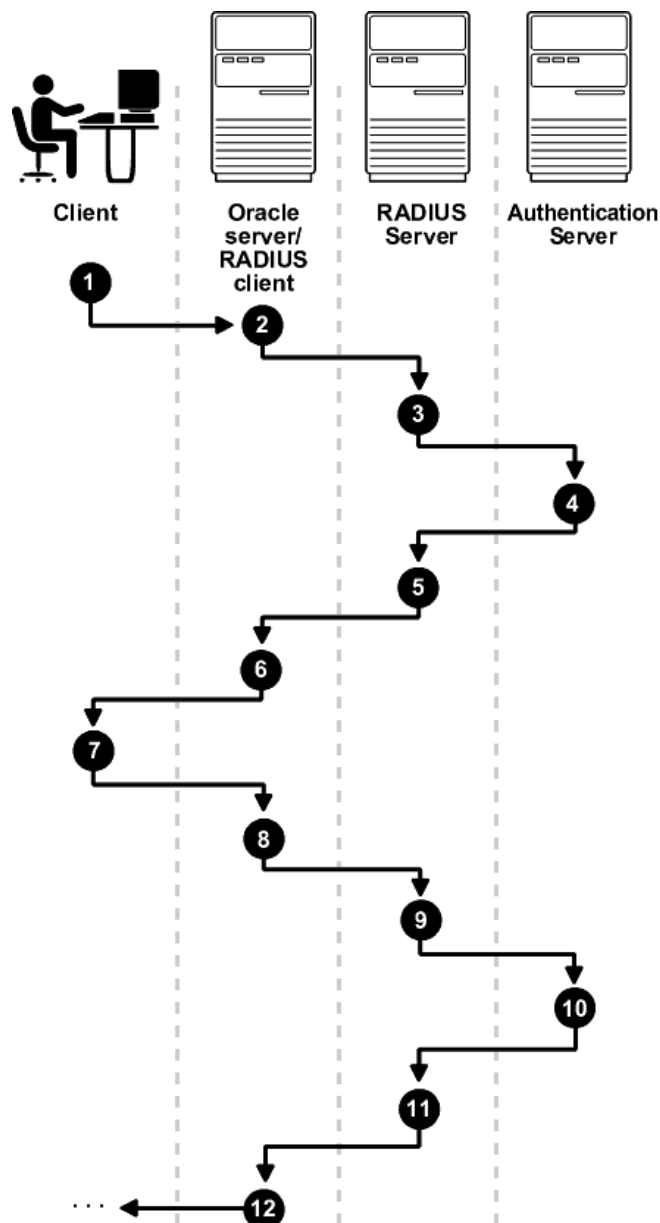


Рисунок 5 - Последовательность асинхронной аутентификации

Oracle Advanced Security предоставляет пользователям RADIUS аутентификацию, разрешения авторизации, сохраненные в RADIUS и основные службы учета при обращении к базе данных Oracle .

## 2 Обеспечение безопасности СУБД Oracle

### 2.1 Прозрачное шифрование данных

Самый большой кошмар любой организации: кто-то украл ленты с резервной копией вашей базы данных. Несомненно, была построена защищенная система, зашифрованы наиболее конфиденциальные ресурсы, размещены серверы баз данных за межсетевыми экранами. Но вор выбрал доступный ему способ: он взял ленты с резервной копией, чтобы, вероятно, скопировать базу данных на другом сервере, запустить экземпляр сервера этой базы данных, а затем не спеша просмотреть все ваши данные. Защита содержимого базы данных от такого воровства представляет собой не только хорошую практику; это - требование многих нормативно-правовых и нормативно-технических документов.

Одно решение состоит в том, чтобы зашифровать в базе данных конфиденциальные данные и хранить ключи шифрования в другом месте; без этих ключей любые украденные данные не будут иметь никакого значения. Тем не менее, необходимо найти баланс между двумя противоречащими понятиями: удобство доступа приложений к ключам шифрования и защита, требуемая для предотвращения воровства этих ключей. Причем соблюдение корпоративных и государственных нормативных требований предполагает немедленное принятие решения, без использования какого-либо сложного кодирования.

Новая функциональная возможность сервера Oracle Advanced Security позволяет сделать это: можно объявить столбец шифруемым, не написав при этом никаких строк кода приложения. Когда пользователи вставляют данные, сервер базы данных прозрачно шифрует эти данные и сохраняет их в столбце. Точно так же, когда пользователи выбирают этот столбец, сервер базы данных автоматически расшифровывает его. Так как все это делается прозрачно без какого-либо изменения кода приложения, эта функциональная возможность имеет соответствующее название: прозрачное шифрование данных (TDE, Transparent Data Encryption) [3].

Все, что необходимо сделать для шифрование - определить столбец, который будет шифроваться, и сервер Oracle Database 11g создаст криптографически стойкий ключ шифрования для таблицы, содержащей этот столбец, и зашифрует данные обычного текста в этом столбце, используя указанный вами алгоритм шифрования. Защита этого ключа таблицы имеет очень важное значение; сервер Oracle Database 11g шифрует его, используя главный ключ, который хранится в безопасном месте, называемом бумажником (wallet), который может быть файлом сервера базы данных. Зашифрованные ключи таблиц размещаются в словаре данных. Когда пользователь вставляет данные в столбец, определенный как зашифрованный, сервер Oracle Database 11g извлекает из бумажника главный ключ, расшифровывает ключ шифрования для этой таблицы, находящийся в словаре данных, использует этот ключ для

шифрования входного значения и сохраняет зашифрованные данные в базе данных, как показано на рисунке 6.

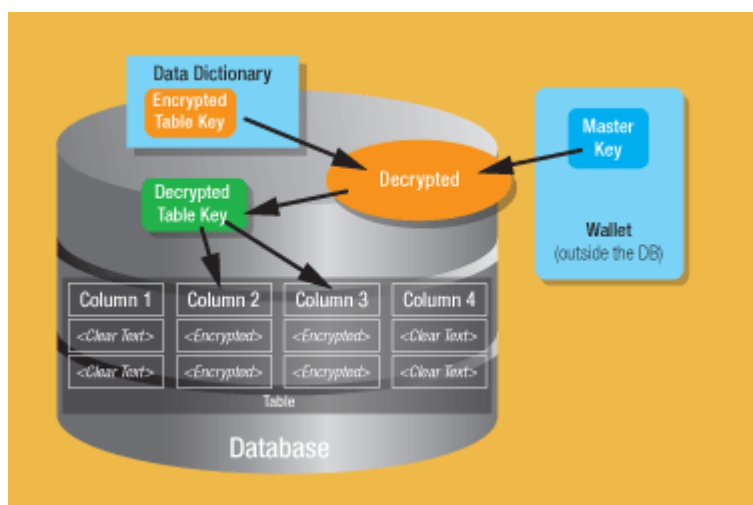


Рисунок 6 - Работа механизма прозрачного шифрования данных

Надписи на рисунке:

- *Data Dictionary* - словарь данных.
- *Encrypted Table Key* - зашифрованный ключ таблицы.
- *Master Key* - главный ключ.
- *Decrypted* - расшифрованный.
- *Wallet (outside the DB)* - бумажник (за пределами базы данных).
- *Decrypted Table Key* - расшифрованный ключ таблицы.
- *Column* - столбец.
- *Clear Text* - обычный текст.
- *Encrypted* - зашифрованный.
- *Table* - таблица.
- *Database* - база данных.

Когда пользователь выбирает зашифрованные столбцы, сервер Oracle Database 11g прозрачно извлекает из словаря данных зашифрованный ключ таблицы, а главный ключ - из бумажника и расшифровывает ключ таблицы. Затем сервер базы данных расшифровывает зашифрованные на диске данные и возвращает пользователю обычный текст.

Благодаря такому шифрованию, если данные будут украдены с диска, они не могут быть извлечены без главного ключа, который находится в бумажнике, не входящим в украденные данные. Даже если украден и бумажник, главный ключ не может быть извлечен из него без знания пароля бумажника. Следовательно, вор не сможет расшифровать данные, даже если он украл диски или копии файлов данных. Это удовлетворяет требованиям соответствия многим нормативным и руководящим документам. И все это было сделано без изменения приложения или написания сложной системы шифрования и управления ключами [3].

Перед тем, как начать использовать возможности TDE, вы должны определить местоположение бумажника, установить его пароль и открыть бумажник.

### 2.1.1 Определение местоположения бумажника

Перед включением возможностей TDE необходимо создать бумажник, в котором будет храниться главный ключ. По умолчанию бумажник создается в каталоге `$ORACLE_BASE/admin/$ORACLE_SID/wallet`. Так, если `$ORACLE_BASE` - `/u01/app/oracle`, а `$ORACLE_SID` - `SWBT4`, то бумажник будет храниться в каталоге `/u01/app/oracle/admin/swbt4/wallet`. Можно также выбрать другой каталог, указывая его в файле `sqlnet.ora`, который находится в каталоге `$ORACLE_HOME/network/admin`.

### 2.1.2 Создание бумажника

Теперь необходимо создать бумажник и установить пароль для доступа к нему. Для этого как пользователь с привилегией `ALTER SYSTEM` выполните следующий оператор:

```
alter system set encryption key authenticated by "wrl";
```

Этот оператор:

- создает бумажник в каталоге, который был определен в пункте 3.1.1;
- устанавливает пароль бумажника - "wrl";
- открывает бумажник для хранения и извлечения главного ключа средствами TDE.

Пароль зависит от регистра, и его необходимо заключать в двойные кавычки. Заметим, пароль "wrl" не показывается в виде обычного текста ни в каких динамических представлениях производительности или журнальных файлах.

Бумажник создается только один раз, поэтому больше не нужно повторять два предыдущих шага. Тем не менее, после запуска экземпляра сервера базы данных бумажник необходимо открывать явно. Когда создается бумажник (как выше в пункте 2.1.2), также открывается бумажник для работы с ним. После создания бумажника и установки его пароля каждый раз, когда открывается база данных, необходимо открывать и бумажник, используя его пароль:

```
alter system set encryption wallet open authenticated by "wrl";
```

Закрывать бумажник необходимо, используя оператор:

```
alter system set encryption wallet close;
```

Чтобы средства TDE работали, бумажник должен быть открытым. В противном случае, доступ будет получен, только ко всем незашифрованным столбцам, но не к зашифрованным.

Чтобы шифровать столбцы, используя средства TDE, все, что необходимо сделать, это добавить к определениям столбцов простое предложение - ENCRYPT. А до этого необходимо, какой будет использоваться тип шифрования и длина ключа.

```
SQL> CREATE TABLE Judge(  
  Judge_ID number Primary Key,  
  Judge_NAME varchar2(20) not null,  
  Judge_SURNAME varchar2(30),  
  Judge_GENDER number not null,  
  Judge_DATE_OF_BIRTH date not null);
```

В настоящее время все данные этой таблицы хранятся в виде обычного текста. Необходимо преобразовать столбцы judge\_name, judge\_surname, judge\_gender, чтобы они хранились в зашифрованном виде. Для этого необходимо выполнить операторы:

```
alter table judge modify (judge_name encrypt);  
alter table judge modify (judge_surname encrypt);  
alter table judge modify (judge_date_of_birth encrypt);
```

Этот оператор делает две вещи:

- создает ключ шифрования для таблицы;
- преобразовывает все значения столбцов в зашифрованный формат.

Этот оператор не изменяет тип данных или размер столбца, и он также не создает никаких триггеров или представлений.

По умолчанию для шифрования используется алгоритм AES (Advanced Encryption Standard, усовершенствованный стандарт шифрования) с 192-битовым ключом. Можно выбрать другой алгоритм, указывая в операторе соответствующее дополнительное предложение [3].

Например, чтобы использовать 128-битовое шифрование по алгоритму AES, вы можете выполнить оператор:

```
alter table judge modify (judge_name encrypt using 'AES128');
```

Также можно использовать предложения AES128, AES192, AES256 или 3DES168 (168-bit Triple DES , трехкратное применение алгоритма DES (Data Encryption Standard, стандарт шифрования данных) с 168-битовым ключом).

После шифрования столбца в описании таблицы появится следующее, как представлено на рисунке 7:

```

SQL> desc judge
Name                               Null?    Type
-----
JUDGE_ID                           NOT NULL NUMBER
JUDGE_NAME                          NOT NULL VARCHAR2(20) ENCRYPT
JUDGE_SURNAME                       VARCHAR2(30) ENCRYPT
JUDGE_GENDER                        NOT NULL NUMBER
JUDGE DATE OF BIRTH                 NOT NULL DATE ENCRYPT

```

Рисунок 7 - Описание таблицы «Judge» после шифрования

Следует обратить внимание на ключевое слово ENCRYPT, указанное после типа данных. Для поиска в базе данных зашифрованных столбцов можно воспользоваться представлением словаря данных DBA\_ENCRYPTED\_COLUMNS. (Средства TDE нельзя применять к таблицам схемы SYS.)

### 2.1.3 Вопросы производительности

Операции шифрования и дешифрования потребляют время центрального процессора, поэтому необходимо рассмотреть их влияние на производительность. Когда происходит обращение к незашифрованным столбцам таблицы, производительность несколько не отличается от производительности при работе с таблицами, для которых не используются средства TDE [3]. Когда же вы обращаетесь к зашифрованным столбцам, возникают небольшие накладные расходы на дешифрование во время выборки данных и шифрование во время вставки, так что вы можете захотеть шифровать столбцы выборочно. Если вам больше не нужно шифровать столбец, вы можете выключить шифрование следующим образом:

```
alter table judge modify (judge_name decrypt);
```

## 2.2 Настройка Radius

Для установки и настройки Radius выполняется следующая последовательность действий:

### Шаг 1 Установка Raduis.

### Шаг 2. А Конфигурация Radius на Oracle Client

- 1) Запуск Oracle Net Manager.  
Пуск, Программы, Oracle - HOME\_NAME, Configuration and Migration Tools, затем Net Manager.
- 2) Открыть вкладку Oracle Net Configuration, и, из вкладки Local, выбрать Profile.
- 3) Из выпадающего меню Naming выбрать Oracle Advanced Security. Появится окно настроек Oracle Advanced Security, представленное на рисунке 8.



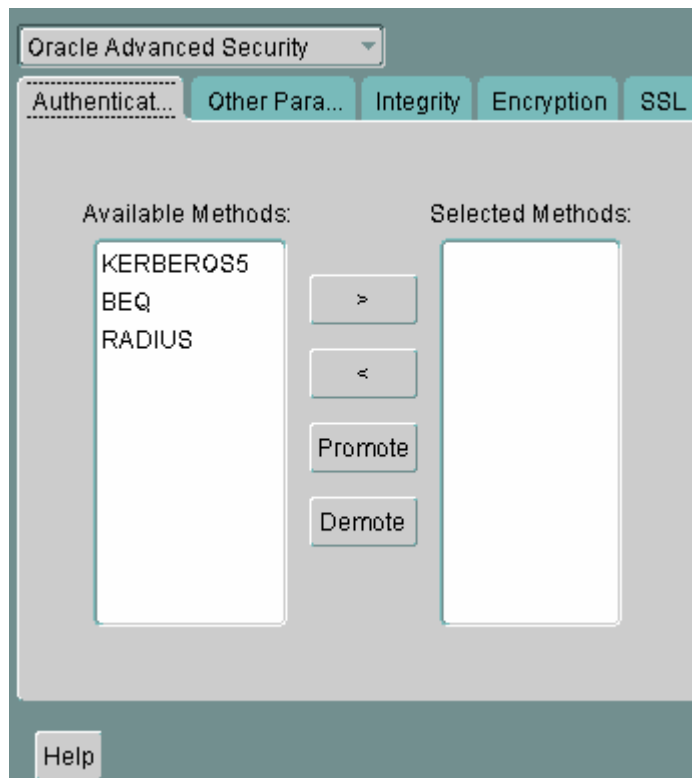


Рисунок 8 - Окно настроек Oracle Advanced Security.  
Вкладка Authentication

- 4) Выбрать вкладку Authentication (она должна быть выбрана по умолчанию).
- 5) Из доступных методов выбрать Radius.
- 6) Сохранить изменения [4].

Файл sqlnet.ora должен обновиться, должна добавиться следующая строка кода:

```
SQLNET.AUTHENTICATION_SERVICES=(RADIUS)
```

### **Шаг 2. Б Конфигурация Radius на Oracle Сервер.**

Выполняется та же последовательность действий (1-6), что и в шаге 2. А, только на Сервере Oracle.

- 7) Далее, в том же окне выбирается вкладка **Other Params** (рисунок 9);
- 8) Из выпадающего меню **Authentication Service** выбрать **Radius**;

Файл sqlnet.ora должен обновиться, должны добавиться следующие строки кода:

```
SQLNET.AUTHENTICATION_SERVICES=RADIUS  
SQLNET.RADIUS_AUTHENTICATION=RADIUS_server_{hostname/IP_address}
```

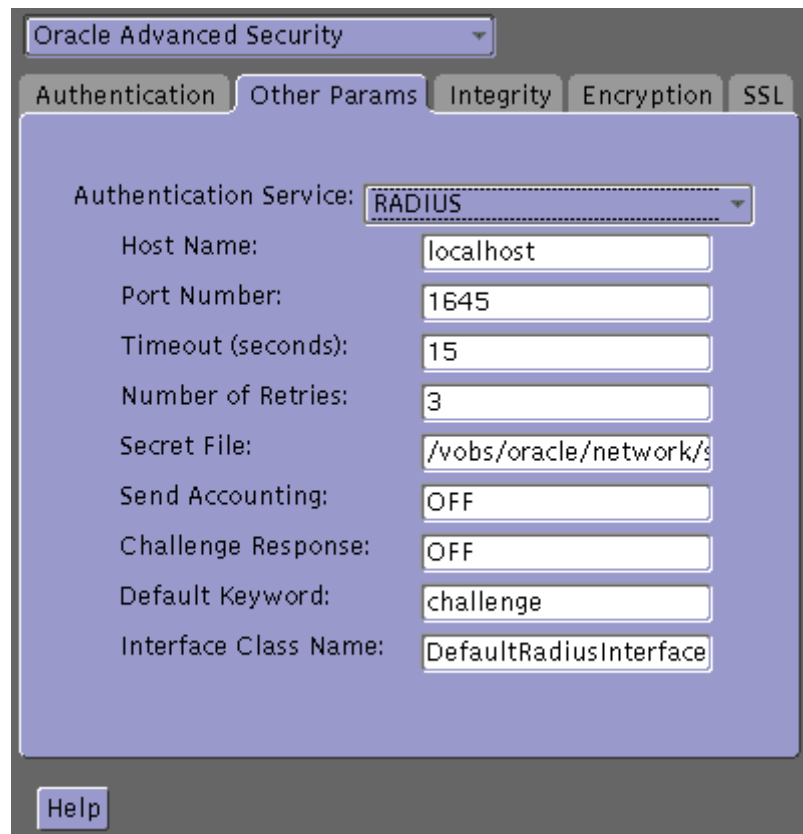


Рисунок 9 - Окно настроек Oracle Advanced Security

Для того, чтобы настроить параметры инициализации Oracle Database Server необходимо добавить следующие строки в файл `init.ora`:

```
OS_AUTHENT_PREFIX=""
```

По умолчанию файл `init.ora` находится в папке `ORACLE_HOME/dbs` в ОС Linux и в папке `ORACLE_HOME\database` в ОС Windows [4].

После чего необходимо перезагрузить базу данных, используя, к примеру, следующий код:

```
SQL> SHUTDOWN  
SQL> STARTUP
```

## **Шаг 2. В Конфигурация дополнительных опций Radius**

Для изменения настроек по умолчанию необходимо выполнить пункты 1-3 шага 2. А. Далее:

- 4) Выбрать вкладку **Other Params**, как показано на рисунке 10.
- 5) Из выпадающего меню **Authentication Service** выбрать **Radius**.
- 6) Изменить настройки по умолчанию по одному из полей, к примеру:
  - а) **Port Number**: задает порт прослушивания основного сервера RADIUS. Значение по умолчанию 1645.

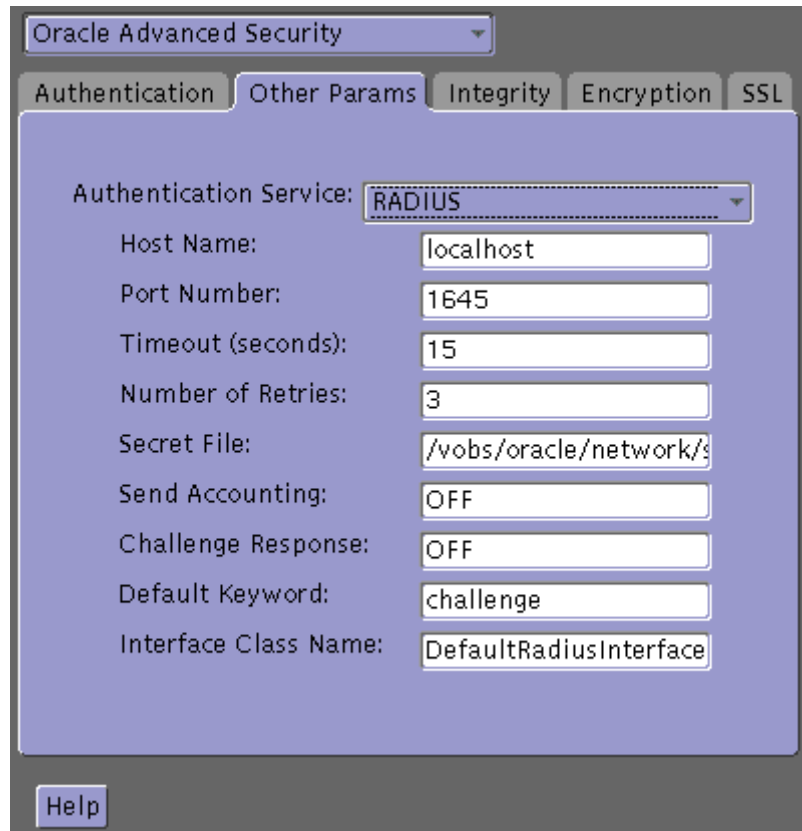


Рисунок 10 - Окно настроек Oracle Advanced Security.  
Вкладка Other Params

б) **Timeout (seconds):** Определяет время, в течение которого сервер базы данных Oracle ожидает ответа от основного сервера RADIUS. Значение по умолчанию 15 секунд.

в) **Number of Retries:** Определяет количество раз, которое сервер базы данных Oracle пытается доставить сообщения на основной сервер RADIUS. Значение по умолчанию 3 попытки.

7) Сохранить все изменения [4].

Файл sqlnet.ora должен обновиться, должны добавиться следующие строки кода:

```

SQLNET.RADIUS_AUTHENTICATION_PORT=(PORT)
SQLNET.RADIUS_AUTHENTICATION_TIMEOUT= (NUMBER OF SECONDS
TO WAIT FOR response)
SQLNET.RADIUS_AUTHENTICATION_RETRIES= (NUMBER OF TIMES TO
RE-SEND TO RADIUS server)
SQLNET.RADIUS_SECRET=(path/radius.key)

```

### Шаг 3. Создание пользователя и предоставление доступа

Для создания пользователя и предоставления ему прав доступа необходимо:

1) Запустить SQL\*Plus на сервере Radius и выполнить следующую последовательность команд:

```
SQL> CONNECT system@database_name;  
SQL> Enter password:  
SQL> CREATE USER username IDENTIFIED EXTERNALLY;  
SQL> GRANT CREATE SESSION TO USER username;  
SQL> EXIT
```

Данный пользователь автоматически создается на сервере базы данных Oracle.

### Шаг 4. Конфигурация авторизации Radius

Для того, чтобы пользователь мог подключиться к серверу базы данных Oracle необходимо добавить следующее:

1) Добавить параметр OS\_ROLE в файл init.ora и переключить его в состояние TRUE:

```
OS_ROLE=TRUE
```

2) Перезапустить базу данных, чтобы система могла применить изменения в файле init.ora . Команды для перезапуска могут иметь следующий вид:

```
SQL> SHUTDOWN  
SQL> STARTUP
```

### 2.3 Аудит базы данных

Аудит Oracle может помочь в определении неавторизованного доступа или внутреннего злоупотребления по отношению к информации содержащейся в базе данных.

Простой набор основных действий аудита должен быть активен все время. Необходимый минимум включает в себя отслеживание доступа пользователей, использование системных привилегий и изменение в структуре базы данных. Этот основной набор не покажет неудавшихся попыток доступа к специфическим данным, которые не должны быть доступны; тем не менее, он даст достаточно простой обзор "некорректного" доступа или использования привилегий. Если служащий подозревается в недозволенных действиях или ожидается атака, тогда может быть применен более детализованный аудит для

специфических таблиц. С точки зрения управления БД, аудит изменения данных для всех таблиц не так уж практичен и может повлиять на производительность системы в целом. Аудит доступа для изменения данных следует использовать для таблиц лишь имеющих особо важное значение (например, заработанная плата сотрудников в базе данных HR) [5].

Стандартные команды аудита позволяют контролировать все системные, и объектные привилегии доступа к любым таблицам или представлениям базы данных на `select`, `delete`, `insert` or `update`. Аудит может быть запущен как для успешных, так и для неуспешных попыток или для тех и других сразу. Как индивидуально для каждого пользователя, так и для всех пользователей сразу, он может выполняться на сессионном уровне или на уровне действия (доступа). На уровне действия - одна запись создается для одного действия, а на сессионном - одна запись для всех контролируемых операций одной сессии.

Часто аудит воспринимается как сложный и медленный. Причина этому обычное невежество. Если большинство из всех опций включены, тогда получающийся в результате журнал аудита может быть большим и трудным для интерпретации и управления. Кроме того, если аудит задействован на всех таблицах и представлениях базы, то это может повлиять и на производительность. Всякий раз, когда выполняемое действие контролируется аудитом в журнал вносится запись; очевидно, что чем интенсивнее используется аудит, тем больше записей будет записано в системное табличное пространство исключительно для аудита. В некоторых случаях это может привести к удвоению количества записей в базу данных: оригинальная запись и выполняемая для нее запись аудита.

Основное правило настройки аудита это простота и предусмотрительность. Необходимо выполнять аудит и детальный мониторинг только тех операций и объектов, информация о которых действительно необходима. Важно то, что с помощью простых отчетов можно выявить нарушения, среди действий, зафиксированных в журнале аудита. Отметим также, что при инсталляции Oracle, по умолчанию, аудит выключен, и Oracle не поставляется с какими-нибудь стандартными установками аудита по умолчанию или отчетами для анализа созданного журнала аудита [5].

Задачу аудита базы данных Oracle не следует ограничивать только лишь использованием команд аудита; так же успешно могут быть применены и другие технологии.

### **Аудит Oracle**

Все привилегии, которые могут быть предоставлены пользователю или роли базы данных могут быть проконтролированы. Сюда включено доступ на чтение, запись и удаление объектов на табличном уровне. Для более детализованного аудита можно задействовать триггеры.

### **Системные триггеры**

Эта возможность была представлена, начиная с Oracle 8, и разрешает выполнение операций триггера, когда имеет место системное событие. Сюда включены запуск и останов базы данных, попытки входа и выхода, создание,

изменение и удаление объектов схемы. С помощью автономных транзакций, можно записывать в журнал упомянутые системные события.

### **Update, delete и insert триггеры**

Это “вторая линия обороны”, которая позволяет понять действия пользователей на более детальном уровне. Для того, что бы отслеживать изменения в базе на уровне столбца и строки, можно написать триггеры, которые позволят полностью сохранять данные, до или после выполненного действия. Использование этого типа контроля очень ресурсоемко, так как создается и хранится много дополнительных записей. Кроме того, что существует еще один недостаток, связанный с этим методом - доступ на чтение нельзя отследить с помощью обычных триггеров базы данных.

### **Детализированный (Fine-grained) аудит**

Детализированный аудит решает проблему отслеживания доступа на чтение. Данная возможность основана на внутренних триггерах, срабатывающих, при разборе какой-нибудь части SQL-предложения. Это очень эффективно, так как SQL-предложение разбирается единожды для аудита и выполнения. Эта возможность использует предикаты, которые определены и проверяются каждый раз, когда происходит доступ к соответствующим объектам. Fine-grained аудит управляется PL/SQL пакетом который называется DBMS\_FGA. Созданная PL/SQL процедура выполняется каждый раз, когда выполняется, соответствующее ей, действие с предикатом. Этот метод позволяет контролировать не только DML-операции на уровне строк и столбцов, но и предложения чтения. Следует предостеречь читателей, в том, что для использования этой возможности необходим некоторый опыт программирования.

### **Системные журналы**

СУБД Oracle генерирует много журнальных файлов, и многие из них могут содержать полезную информацию для проведения аудита. Например, alert log используется для записи информации о запуске и остановке базы, а также о вносимых структурных изменениях, таких как добавление файла данных в базу [5].

В производственной базе данных никому из пользователей никогда не следует изменять структуру схемы. Администраторам баз данных следует вносить изменения в специально отведенное для этого время. Какие-либо другие изменения следует рассматривать как подозрительные. Наблюдение за структурными изменениями может включить индикаторы некорректного использования базы данных.

Аудит в Oracle разделен на три части:

- аудит таких выражений как CREATE TABLE или CREATE SESSION;
- аудит привилегий ALTER USER;
- аудит на объект на объектном уровне SELECT TABLE.

Записи аудита могут помещаться либо в аудиторскую таблицу базы данных, либо в аудиторский журнал операционной системы. Запись аудита в журнал операционной системы в некоторых случаях более защищена, но эта

возможность доступна не для всех платформ и ее специфика зависит от платформы. В этой статье в качестве места хранения для журнала аудита мы будем использовать базу данных [5].

Аудит включается для записи в базу данных добавлением следующей строки в файле *init.ora*. Символьная связь к нему обычно может быть найдена в *\$ORACLE\_HOME/dbs*.

```
audit_trail = db
```

После этого базу данных необходимо перезапустить. Простая проверка покажет, что аудит действительно включен.

```
select name, value from v$parameter where name like 'audit%';
```

Результат работы данного кода представлен на рисунке 11.

```
SQL> select name,value from v$parameter where name like 'audit%';

NAME
-----
VALUE
-----
audit_sys_operations
FALSE

audit_file_dest
/u01/app/oracle/admin/orcl/adump

audit_syslog_level

NAME
-----
VALUE
-----
audit_trail
DB
```

Рисунок 11 - Проверка работы аудита

Но контролируемые действия не отслеживаются до тех пор, пока эти действия не заданы явно; это верно, кроме случаев привилегированного доступа к базе данных, запуска и останова базы данных, структурных изменений, таких как добавление файла данных. Эти действия отслеживаются в файле операционной системы в *\$ORACLE\_HOME/rdbms/audit* до тех пор пока *audit\_file\_dest* не переопределено в файле *init.ora*. В Windows эти события появляются в Event Viewer.

Для того, что бы проверить наличие того, что какие-нибудь привилегии или выражения уже используются для аудита, необходимо сделать следующее:

```
select * from dba_stmt_audit_opts
union
select * from dba_priv_audit_opts;
```

Что бы найти какие объекты уже контролируются аудитом, необходимо запросить представление dba\_obj\_audit\_opts [5].

Что бы пользователь мог задать команду аудита, необходимым условием для него является наличие привилегии "AUDIT SYSTEM". Найти пользователей, которые имеют эту привилегию, можно выполнив следующее:

```
select * from dba_sys_privs  
where privilege like '%AUDIT%';
```

Данный скрипт выведет набор команд аудита в спул файл, который затем запустится для выполнения команд аудита.

```
set head off  
set feed off  
set pages 0  
spool aud.lis  
select 'audit '||name||';'  
from system_privilege_map  
where (name like 'CREATE%TABLE%'  
or name like 'CREATE%INDEX%'  
or name like 'CREATE%CLUSTER%'  
or name like 'CREATE%SEQUENCE%'  
or name like 'CREATE%PROCEDURE%'  
or name like 'CREATE%TRIGGER%'  
or name like 'CREATE%LIBRARY%')  
union  
select 'audit '||name||';'  
from system_privilege_map  
where (name like 'ALTER%TABLE%'  
or name like 'ALTER%INDEX%'  
or name like 'ALTER%CLUSTER%'  
or name like 'ALTER%SEQUENCE%'  
or name like 'ALTER%PROCEDURE%'  
or name like 'ALTER%TRIGGER%'  
or name like 'ALTER%LIBRARY%')  
union  
select 'audit '||name||';'  
from system_privilege_map  
where (name like 'DROP%TABLE%'  
or name like 'DROP%INDEX%'  
or name like 'DROP%CLUSTER%'  
or name like 'DROP%SEQUENCE%'
```



```

or name like 'DROP%PROCEDURE%'
or name like 'DROP%TRIGGER%'
or name like 'DROP%LIBRARY%')
union
select 'audit '||name||';'
from system_privilege_map
where (name like 'EXECUTE%INDEX%'
or name like 'EXECUTE%PROCEDURE%'
or name like 'EXECUTE%LIBRARY%')
/
spool off
@@aud.lis

```

Для создания команд аудита можно было бы использовать другой способ, через представления базы данных *dba\_sys\_privs*, использующий действительные разрешения пользователей. Этот способ может показаться лучшим решением и включает в себя меньше команд, но потенциально это бы не сработало для случаев, когда новые разрешения предоставлены пользователям. В этом случае, пришлось бы выполнять новые команды аудита после предоставления новых привилегий.

Сейчас, установки могут быть просмотрены при помощи этого SQL:

```

select audit_option,success,failure
from dba_stmt_audit_opts
union
select privilege,success,failure
from dba_priv_audit_opts

```

Каждый раз, когда пользователь пытается что-нибудь затронуть в базе данных, на что включен аудит, ядро Oracle проверяет действие и создает или обновляет (в случае одной записи для сессии) запись в таблице AUD\$, владельцем которой является пользователь SYS. Эта таблица по умолчанию находится в табличном пространстве SYSTEM.

AUD\$ - особенная таблица [словаря данных], так как только из нее пользователь SYS имеет право удалять из нее записи. Если журнал аудита включен и пишется в базу данных, то число записей в этой таблице необходимо внимательно контролировать что бы удостовериться что она не растет слишком быстро, и не заполнила все системное табличное пространство. Стратегия очистки нуждается в адаптации, что бы сохранять размер таблицы и если необходимо архивировать записи журнала аудита для будущего использования. Одна из тактик может состоять в том, что бы копировать записи в итоговую таблицу, созданную для выполнения спецпроверки с целью выявления злоупотреблений. Эти итоговые таблицы могут располагаться в отдельной базе

данных для увеличения защищенности. После копирования таблица sys.aud\$ может быть усечена [5].

Таблицу SYS.AUD\$ можно передвинуть в табличное пространство, отличное от SYSTEM. Только пользователи, которым предоставлен специальный доступ к таблице SYS.AUD\$ могут читать, изменять и удалять данные из нее. По умолчанию этими правами владеет SYS, но эти действия может выполнять любой другой пользователь, наделенный необходимыми правами. Существуют две специальные роли, которым разрешен доступ к таблице SYS.AUD\$ на select и delete, это роли DELETE\_CATALOG\_ROLE и SELECT\_CATALOG\_ROLE. Эти роли не следует предоставлять обычным пользователям.

Записи аудита могут быть просмотрены различными способами:

- путем выборки записей из SYS.AUD\$ - Это исходный журнал аудита;
- путем выборки записей из dba\_audit\_trail - Это представление DBA показывающее исходный журнал аудита;
- путем выборки записей из dba\_audit\_session - Это представление показывает только лишь события входа и выхода;

### **Неудачные попытки входа**

Они могут означать попытки атакующего получить неавторизованный доступ в базу данных. Нижеследующий SQL ярко демонстрирует это:

```
select count(*), username, terminal, to_char(timestamp, 'DD-MON-YYYY')  
from dba_audit_session  
where returncode<>0  
group by username, terminal, to_char(timestamp, 'DD-MON-YYYY');
```

### **Попытки доступа несуществующих пользователей в базу данных**

Одно интересное дополнение к приведенному выше SQL позволяет отыскивать попытки входа в систему под несуществующим пользователем. В этом случае тоже будут созданы записи аудита. Следующий SQL иллюстрирует это:

```
select username, terminal, to_char (timestamp, 'DD-MON-YYYY  
HH24:MI:SS')  
from dba_audit_session where returncode<>0  
and not exists (select 'x' from dba_users where dba_users.username=  
dba_audit_session.username)
```

Все попытки войти в систему под несуществующим пользователем следует проверять и расследовать каждый день.

### **Попытки доступа в базу данных в необычное время**

Следует выполнять проверки попыток доступа в базу данных во внерабочие часы. Им может оказаться обычная сверхурочная работа, но также

легко - неавторизованный доступ. Его можно проверить следующим выражением:

```
select username, terminal, action_name, returncode, to_char
(timestamp, 'DD-MON-YYYY HH24:MI:SS'), to_char(logoff_time, 'DD-MON-YYYY
HH24:MI:SS')
from dba_audit_session
where to_date (to_char (timestamp, 'HH24:MI:SS'), 'HH24:MI:SS') <
to_date ('08:00:00', 'HH24:MI:SS')
or to_date (to_char (timestamp, 'HH24:MI:SS'), 'HH24:MI:SS') > to_date
('19:30:00', 'HH24:MI:SS');
```

Приведенные выше SQL показывает любые соединения до 8:00 утра и после 7:30 вечера. Любые соединения, особенно те, которые выполнены привилегированными пользователями, такими как SYS или SYSTEM, должны быть исследованы. Особенное внимание следует обратить на то, откуда был произведен доступ. Например, если привилегированный доступ выполнен с машины, которая не находится в отделе администратора, администратор должен выяснить зачем он производился.

#### **Проверка пользователей, которые используют общую учетную запись в базе данных**

Следующее выражение SQL ищет пользователей, которые потенциально могут использовать общую учетную запись в базе данных:

```
select count(distinct(terminal)),username
from dba_audit_session
having count(distinct(terminal))>1
group by username
```

#### **Множественные попытки доступа под различными учетными записями с одного и того же терминала**

Данное выражение SQL довольно таки простое и к нему может быть добавлена группировка по дню, а также выведены пользователи для каждого терминала.

```
select count(distinct(username)),terminal
from dba_audit_session
having count(distinct(username))>1
group by terminal
```

Данный отчет показывает кого-либо пытающегося получить доступ перебором учетных записей и паролей, но сюда же могут попасть законопослушные пользователи, которые используют различные учетные

записи для различных аспектов своей работы. В любом случае администратору следует выяснить это в дальнейшем [5].

Настройка аудита это один из первых шагов для обеспечения безопасности базы данных. Использование аудита должно быть частью общего плана безопасности организации, в который входит и Oracle. Следует регулярно контролировать базу данных на неправильность конфигурации или наличие вновь обнаруженных уязвимостей, которые могут стать брешью в информационной безопасности системы.

Из-за своей сложной природы и большого числа различных параметров, сервер Oracle может быть по-разному настроен, однако, что бы наилучшим образом обеспечить безопасность необходимо всегда следовать принципу наименьших привилегий. Как только база данных станет частью общего плана безопасности и будет корректно сконфигурирована и регулярно проверяема, тогда аудит следует рассматривать как важную часть этой общей стратегии.

В основном, не стоит представлять какие-либо привилегии обычным пользователям в производственной базе данных, стоит удалить большинство привилегий PUBLIC, удалить, заблокировать или изменить пароли всех учетных записей по умолчанию. Стоит убедиться в том, что пользователи придерживаются политики безопасности при работе с паролями и включена функция управления паролями.

Важно, чтобы настройки аудита планировались с точки зрения производительности и удобства использования. Журнал аудита также должен был управляем.

Не менее важно то, что данные журнала аудита можно описывать в категориях защиты информации.

## **3 Разработка базы данных**

### **3.1 Постановка задачи**

В данной дипломной работе необходимо обеспечить безопасность базы данных автоматизации проведения соревнований по фигурному катанию. База должна предусматривать отражение категорий фигуристов, тренеров, судей, имеющих в базе данных, каждое место (призовое и не призовое) фигуриста в рамках данного соревнования, а также учитывать информацию о том, каким судьей, какому фигуристу, какая оценка была поставлена за ту или иную программу. Информация об оценках в данной базе данных предназначена для самостоятельного просмотра фигуристом, тренером или судьей, но она также исключает возможность модификации, обновления или удаления данных ими. Таким образом, вводится необходимость разграничения прав доступа.

Для достижения поставленной цели база данных «Автоматизация проведения соревнований по фигурному катанию» должна предоставлять следующие возможности:

- хранение информации о фигуристах;
- хранение информации о тренерах;
- хранение информации о хореографах;
- хранение информации о судьях;
- хранение информации о технической части проведения соревнований;
- учет и автоматизация оценок фигуристов.

При работе с базой технической специалист должен иметь возможности решать следующие задачи:

- 1) принимать заявки на участие и регистрировать их в системе;
- 2) размещать информацию о программах фигуристов;
- 3) фиксировать оценки фигуристов;
- 4) производить удаление фигуристов из базы. Данное явление может иметь место при распаде пар, окончании карьеры в любительском виде фигурного катания, отказе от участия в соревнование и т.д.

Фигурист, тренер, хореограф, судья имеют возможность решать следующие задачи:

- 1) просматривать список участников соревнования;
- 2) просматривать текущие и итоговые оценки, без возможности редактирования;
- 3) для выбранного фигуриста (судьи, тренера, хореографа) получить интересные сведения.

### 3.2 Анализ предметной области

Требуется разработать базу данных для автоматизации проведения соревнований по фигурному катанию. База содержит сведения о фигуристах, которые характеризуются следующими параметрами (реализовано в таблице «Фигурист (Figure\_Skater)»):

- уникальный код фигуриста;
- имя;
- фамилия;
- страна;
- пол;
- дата рождения;
- город проживания;
- рост;
- год начала карьеры;
- тренер;
- хореограф;
- категория фигурного катания.

Категория судей, оценивающих программы фигуристов, характеризуется следующими параметрами (реализовано в таблице «Судья (Judge)»):

- уникальный код судьи;
- имя;
- фамилия;
- пол;
- дата рождения.

Каждый фигурист (пара) имеет своего тренера и хореографа, которые в свою очередь могут тренировать нескольких фигуристов. Каждый тренер и хореограф имеют следующие характеристики (реализовано в таблицах «Тренер (Coach)» и «Хореограф (Choreographer)» соответственно):

- уникальный код тренера (хореографа);
- фамилия;
- имя.

В системе имеется информация о соревнованиях, которая представлена следующими характеристиками (реализовано в таблице «О Соревновании (About\_Competition)»):

- уникальный код соревнования;
- название соревнования;
- страна проведения;
- город проведения;
- адрес ледовой арены;
- дата начала соревнования;
- дата окончания соревнования.

В рамках соревнования фигуристы и судьи идентифицируются по заявкам. Информация о заявках представлена следующими характеристиками (реализовано в таблице «Заявка Фигуриста (Request\_Figure\_Skating)»):

- уникальный код заявки;
- уникальный код 1-го фигуриста;
- уникальный код 2-го фигуриста (может быть представлена пустым полем в случае одиночной категории фигурного катания);
- уникальный код соревнования.

В ходе проведения соревнования фигуристы представляют 2 вида программ: короткую и произвольную, которые оценивают судьи. Система судейства следующая: каждый судья ставит оценку каждому фигуристу (паре) за каждый элемент, исполненный в программе (от -3 до 3).

Далее выводится средний бал, который впоследствии суммируется с базовой стоимостью элемента. Перечень возможных элементов присутствует в системе (реализовано в таблице «Заявка Судья (Request\_Judge)»).

В системе также имеется информация о программах фигуристов, которая имеет следующие параметры (реализовано в таблице «О программе (About\_Program)»):

- уникальный код вида программы;
- уникальный код заявки;
- уникальный код соревнования;
- название программы;
- музыка.

Оценки за технические элементы по результатам программ хранятся в системе и имеют следующие характеристики (реализовано в таблице «Оценка за технику (Request\_Figure\_Skating)»):

- уникальный код судьи;
- уникальный код заявки фигуристов;
- уникальный код вида программы;
- уникальный код элемента;
- оценка за элемент.

По окончании каждого этапа соревнования итоговые оценки заносятся в систему, и определяется текущее место фигуриста (пары). По окончании всего соревнования подсчитываются итоговые оценки, путем суммирования результатов двух программ, подводятся результаты, присуждаются места и медали (реализовано в таблице «Результат (Result)»).

Во время начальной стадии разработки БД и концептуального проектирования мы определяем первоначальный набор сущностей. Эти сущности представляют собой наиболее важную информацию об объектах системы с точки зрения представления конечных пользователей и проектировщика. Некоторые сущности являются объектами реального мира, например, фигуристы, тренеры, судьи. Другие представляют собой

информацию о сущностях, например, элементы, категории фигурного катания. В данной работе будут использоваться сущности, представленные в таблице 3.

Т а б л и ц а 3 – Описание назначения таблиц

Имя сущности	Описание сущности	Тип сущностей
Segment	Данные о видах программ	Сильная
Gender	Данные о поле	Сильная
Medal	Данные о медали	Сильная
Category	Данные о категории фигурного катания	Сильная
Element	Данные об элементах, которые фигуристы исполняют в своих программах	Сильная
Figure_Skater	Данные о фигуристе	Слабая
Judge	Данные о судье	Слабая
Coach	Данные о тренере	Сильная
Choreographer	Данные о хореографе	Сильная
About_Compensation	Данные о соревновании	Сильная
Request_Figure_Skater	Данные о заявке фигуриста (-ов) на участие в соревновании	Слабая
Request_Judge	Данные о заявке судьи на судейство в соревновании	Слабая
About_Programm	Данные о программе, представляемой фигуристом(-ами)	Слабая
Technical_Element_Score	Данные об оценках фигуриста(-ов) за технические элементы	Слабая
Total_Segment_Score	Данные об оценках фигуриста(-ов) за виды программы	Слабая
Result	Данные об итоговой оценке фигуриста(-ов), местах и медалях.	Слабая

### 3.3 Концептуально проектирование

Учитывая особенности данной системы, рассмотренной в анализе предметной области, выделим таблицы и связи между ними, представленные на предварительной ER-диаграмме. Общая ER-диаграмма представлена на рисунке 12.

На рисунке 13 представлен окончательный вариант концептуальной модели базы данных, с учетом всех бизнес правил, предназначенной для проведения соревнований по фигурному катанию.



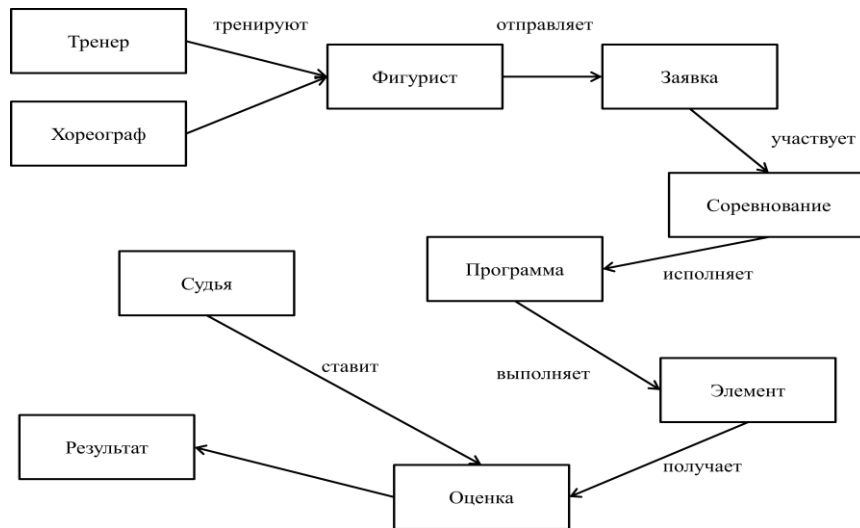


Рисунок 12 – Общая ER-диаграмма

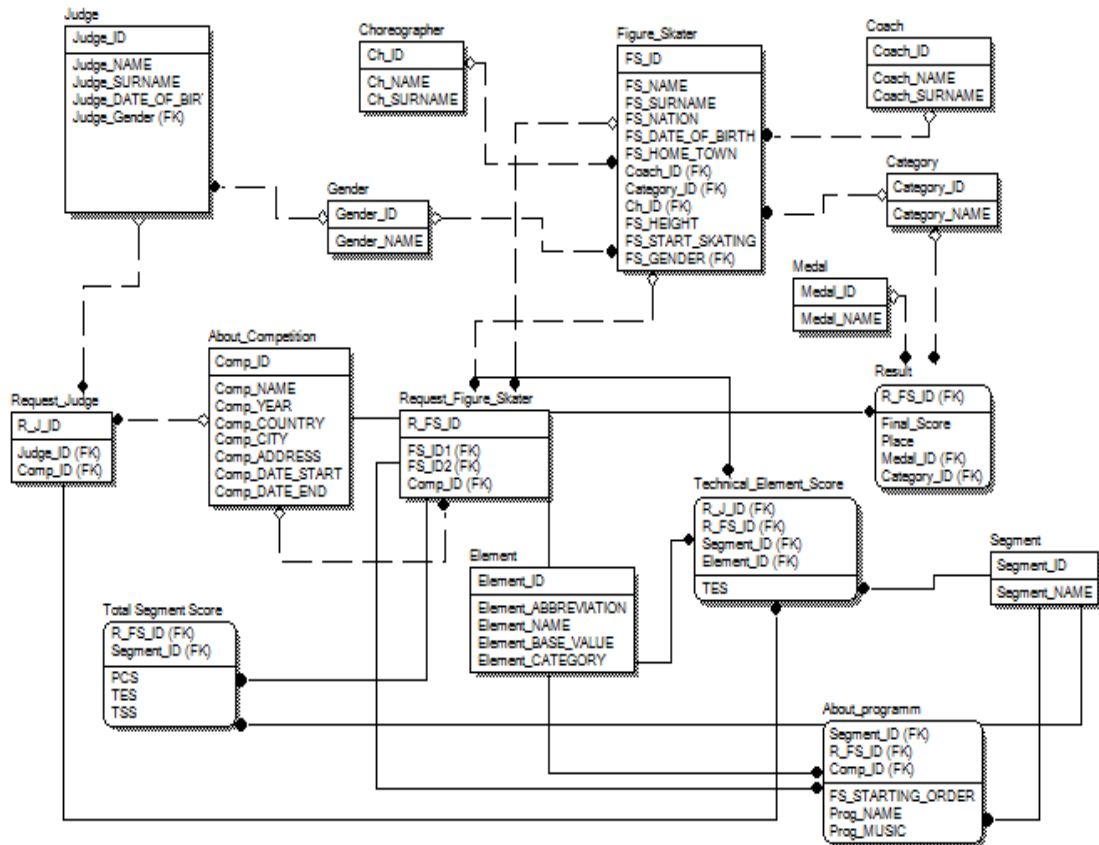


Рисунок 13 – Окончательный вариант концептуальной модели

### 3.4 Логическое проектирование

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

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.

### **3.4.1 Создание пользователей, логинов и задание им паролей**

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного 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. Для нашей базы данных на основании анализа предметной области создаются 3 учетные записи: администратора, судей и пользователя (фигурист, тренер, хореограф). Администратору дается полный контроль над БД, технические операторы могут добавлять, удалять и обновлять данные, а в права пользователя вносятся ограничения на изменение и обновление таблиц.

```
CREATE TABLESPACE fss  
DATAFILE '/u01/app/oracle/oradata/asel/fss.DAT'  
SIZE 150M REUSE AUTOEXTEND ON NEXT 2M MAXSIZE 500M;
```

```
create user fss identified by fss  
DEFAULT TABLESPACE fss  
TEMPORARY TABLESPACE TEMP;
```

```
alter user fss QUOTA unlimited ON fss;
```

```
grant create session to fss;  
grant create any table to fss;  
grant create any procedure to fss;  
grant create any trigger to fss;  
grant create any sequence to fss;  
grant create any view to fss;  
grant update any table to fss;  
grant drop any procedure to fss;  
grant drop any trigger to fss;  
grant drop any sequence to fss;  
grant drop any view to fss;  
grant delete any table to fss;
```

*grant alter any table to fss;*  
*grant alter any procedure to fss;*  
*grant alter any trigger to fss;*  
*grant alter any sequence to fss;*  
*grant alter session to fss;*  
*grant references on fss.Judge to fss;*  
*grant references on fss.Figure\_Skater to fss;*  
*grant references on fss.About\_competition to fss;*  
*grant references on fss.Request\_Judge to fss;*  
*grant references on fss.Request\_Figure\_Skater to fss;*  
*grant references on fss.About\_Programm to fss;*  
*grant references on fss.Technical\_Element\_Score to fss;*  
*grant references on fss.Total\_Segment\_Score to fss;*  
*grant references on fss.Updater to fss;*  
*grant references on fss.Result to fss;*

### **3.4.2 Схемы отношений, составленные на языке определения данных (ddl, data definition language)**

#### **Создание таблиц**

*CREATE TABLE Gender(  
 Gender\_ID number PRIMARY KEY,  
 Gender\_NAME VARCHAR2(10) NOT NULL);*

*CREATE TABLE Categor (  
 Category\_ID number PRIMARY KEY,  
 Category\_NAME VARCHAR2(40) NOT NULL);*

*CREATE TABLE Segment(  
 Segment\_ID number Primary key,  
 Segment\_NAME varchar2(30) not null);*

*CREATE TABLE Element(  
 Element\_ID number not null PRIMARY KEY,  
 Element\_ABBREVIATION varchar2(10) not null ,  
 Element\_NAME VARCHAR2(80) not null,  
 Element\_BASE\_VALUE float not null,  
 Element\_Category varchar2(15) not null);*

*CREATE TABLE Coach(  
 Coach\_ID number Primary Key,  
 Coach\_NAME varchar2(20) not null,*

*Coach\_SURNAME varchar2(30));*

*CREATE TABLE Choreographer(  
Ch\_ID number Primary Key,  
Ch\_NAME varchar2(20) not null,  
Ch\_SURNAME varchar2(30));*

*CREATE TABLE Judge(  
Judge\_ID number Primary Key,  
Judge\_NAME varchar2(20) not null,  
Judge\_SURNAME varchar2(30),  
Judge\_GENDER number not null,  
Judge\_DATE\_OF\_BIRTH date not null);*

*CREATE TABLE Figure\_Skater(  
FS\_ID number Primary Key,  
FS\_NAME varchar2(20) not null,  
FS\_SURNAME varchar2(30),  
FS\_NATION varchar2(5) not null,  
FS\_GENDER number not null,  
FS\_DATE\_OF\_BIRTH date not null,  
FS\_HOME\_TOWN varchar2(30) not null,  
FS\_HEIGHT number not null,  
FS\_START\_SKATING number not null,  
Coach\_ID number not null,  
Choreographer\_ID number not null,  
Category\_ID number not null);*

*CREATE TABLE About\_competition(  
Comp\_ID number Primary Key not null,  
Comp\_NAME varchar2(50) not null,  
Comp\_YEAR number not null,  
Comp\_COUNTRY varchar2(17) not null,  
Comp\_CITY varchar2(17) not null,  
Comp\_ADDRESS varchar2(50) not null,  
Comp\_DATE\_START date not null,  
Comp\_DATE\_END date not null);*

*CREATE TABLE Medal(  
Medal\_ID number Primary Key,  
Medal\_NAME varchar2(10) not null );*

*CREATE TABLE Request\_Judge(  
R\_J\_ID number PRIMARY KEY,*

*Judge\_ID number not null,  
Comp\_ID number not null );*

*CREATE TABLE Request\_Figure\_Skater(  
R\_FS\_ID number PRIMARY KEY,  
FS\_ID\_1 number not null,  
FS\_ID\_2 number null,  
Comp\_ID number not null );*

*CREATE TABLE About\_programm(  
Segment\_ID number not null,  
R\_FS\_ID number not null,  
Comp\_ID number not null,  
Prog\_NAME varchar2(50),  
Prog\_MUSIC\_BY varchar2(50));*

*CREATE TABLE Technical\_Element\_Score(  
R\_FS\_ID number not null,  
R\_J\_ID number not null,  
Segment\_ID number not null,  
Element\_ID number not null,  
TES number CHECK (TES in (-3,-2,-1,0,1,2,3)) not null ,  
CONSTRAINT PK\_Technical\_Element\_Score PRIMARY  
KEY(R\_FS\_ID,R\_J\_ID,Segment\_ID,Element\_ID));*

*CREATE TABLE UPDATER(  
U\_R\_FS\_ID number not null,  
U\_Segment\_ID number not null,  
U\_Element\_ID number not null,  
U\_TES float not null,  
CONSTRAINT PK\_UPDATER PRIMARY KEY(U\_R\_FS\_ID, U\_Segment\_ID,  
U\_Element\_ID));*

*CREATE TABLE Total\_Segment\_Score(  
R\_FS\_ID number not null,  
Segment\_ID number not null ,  
PCS float null,  
TES float not null,  
TSS float null,  
CONSTRAINT PK\_Total\_Segment\_Score PRIMARY KEY  
(R\_FS\_ID,Segment\_ID));*

*create table Result(*

```

R_FS_ID number not null Primary key,
Final_Score float not null,
Category_ID number not null,
Place number,
Medal_ID number REFERENCES Medal(Medal_ID));

alter table Judge
add constraint Judge_const foreign key (Judge_GENDER) references
Gender(Gender_ID);

alter table Figure_Skater
add constraint Figure_Skater_gender foreign key (FS_GENDER) references
Gender(Gender_ID);
alter table Figure_Skater
add constraint Figure_Skater_coach foreign key (Coach_ID) REFERENCES
Coach(Coach_ID);
alter table Figure_Skater
add constraint Figure_Skater_ch foreign key (Choreographer_ID )
REFERENCES Choreographer(Ch_ID);
alter table Figure_Skater
add constraint Figure_Skater_cat foreign key (Category_ID) REFERENCES
Categor(Category_ID);

alter table Request_Judge
add constraint Jreq foreign key (Judge_ID) REFERENCES Judge(Judge_ID);
alter table Request_Judge
add constraint Jcomp foreign key (Comp_ID) REFERENCES
About_competition(Comp_ID);

alter table Request_Figure_Skater
add constraint RFS_Comp foreign key (Comp_ID) REFERENCES
About_competition(Comp_ID);
alter table Request_Figure_Skater
add constraint RFS_FS1 foreign key (FS_ID_1) REFERENCES
Figure_Skater(FS_ID);
alter table Request_Figure_Skater
add constraint RFS_FS2 foreign key (FS_ID_2) REFERENCES
Figure_Skater(FS_ID);

alter table About_programm
add constraint AP_SEG foreign key (Segment_ID) REFERENCES
Segment(Segment_ID);
alter table About_programm

```

```

    add constraint AP_RF foreign key (R_FS_ID) REFERENCES
Request_Figure_Skater(R_FS_ID);
    alter table About_programm
    add constraint AP_Comp foreign key (Comp_ID) REFERENCES
About_competition(Comp_ID);

    alter table Technical_Element_Score
    add constraint TES_RFS foreign key (R_FS_ID) REFERENCES
Request_Figure_Skater(R_FS_ID);
    alter table Technical_Element_Score
    add constraint TES_RJ foreign key (R_J_ID) REFERENCES
Request_Judge(R_J_ID);
    alter table Technical_Element_Score
    add constraint TES_SEG foreign key (Segment_ID) REFERENCES
Segment(Segment_ID);
    alter table Technical_Element_Score
    add constraint TES_EL foreign key (Element_ID) REFERENCES
Element(Element_ID);

    alter table UPDATER
    add constraint UP_RFS foreign key (U_R_FS_ID) REFERENCES
Request_Figure_Skater(R_FS_ID);
    alter table UPDATER
    add constraint UP_SEG foreign key (U_Segment_ID) REFERENCES
Segment(Segment_ID);
    alter table UPDATER
    add constraint UP_EL foreign key (U_Element_ID) REFERENCES
Element(Element_ID);

    alter table Total_Segment_Score
    add constraint TSS_RFS foreign key (R_FS_ID) REFERENCES
Request_Figure_Skater(R_FS_ID);
    alter table Total_Segment_Score
    add constraint TSS_SEG foreign key (Segment_ID) REFERENCES
Segment(Segment_ID);

    alter table Result
    add constraint RES_RFS foreign key (R_FS_ID) REFERENCES
Request_Figure_Skater(R_FS_ID);
    alter table Result
    add constraint RES_CAT foreign key (Category_ID) REFERENCES
Categor(Category_ID);

```

```

alter table about_competition modify (comp_name encrypt using
'3DES168');
alter table about_competition modify (comp_year encrypt using
'3DES168');
alter table about_competition modify (comp_country encrypt using
'3DES168');
alter table about_competition modify (comp_city encrypt using
'3DES168');
alter table about_competition modify (comp_address encrypt using
'3DES168');
alter table about_competition modify (comp_date_start encrypt using
'3DES168');
alter table about_competition modify (comp_date_end encrypt using
'3DES168');

alter table about_programm modify (prog_name encrypt using
'3DES168');
alter table about_programm modify (prog_music_by encrypt using
'3DES168');

alter table judge modify (judge_name encrypt using '3DES168');
alter table judge modify (judge_surname encrypt using '3DES168');
alter table judge modify (judge_date_of_birth encrypt using '3DES168');

```

### **3.5 Физическое проектирование**

Для разработки базы данных была выбрана СУБД Oracle, которая является всеобъемлющим, интегрированным сквозным решением, которое наделяет пользователей организации безопасной, надежной, и продуктивной платформой для обработки промышленной информации и приложений, касающихся интеллектуальных ресурсов предприятия. Oracle предоставляет мощные, знакомые инструменты для профессионалов информационных технологий так же, как и для работников информационной сферы, уменьшая сложность создания, развёртывания, управления и использования данных предприятия и аналитических приложений на платформах от мобильных устройств до информационных систем предприятия. Благодаря исчерпывающему набору функций, взаимодействию с существующими системами и автоматизации типовых задач, Oracle 11 g предоставляет полное решение в области хранения данных для предприятий всех масштабов.

Диаграмма физической модели данной базы данных представлена на рисунке 14.



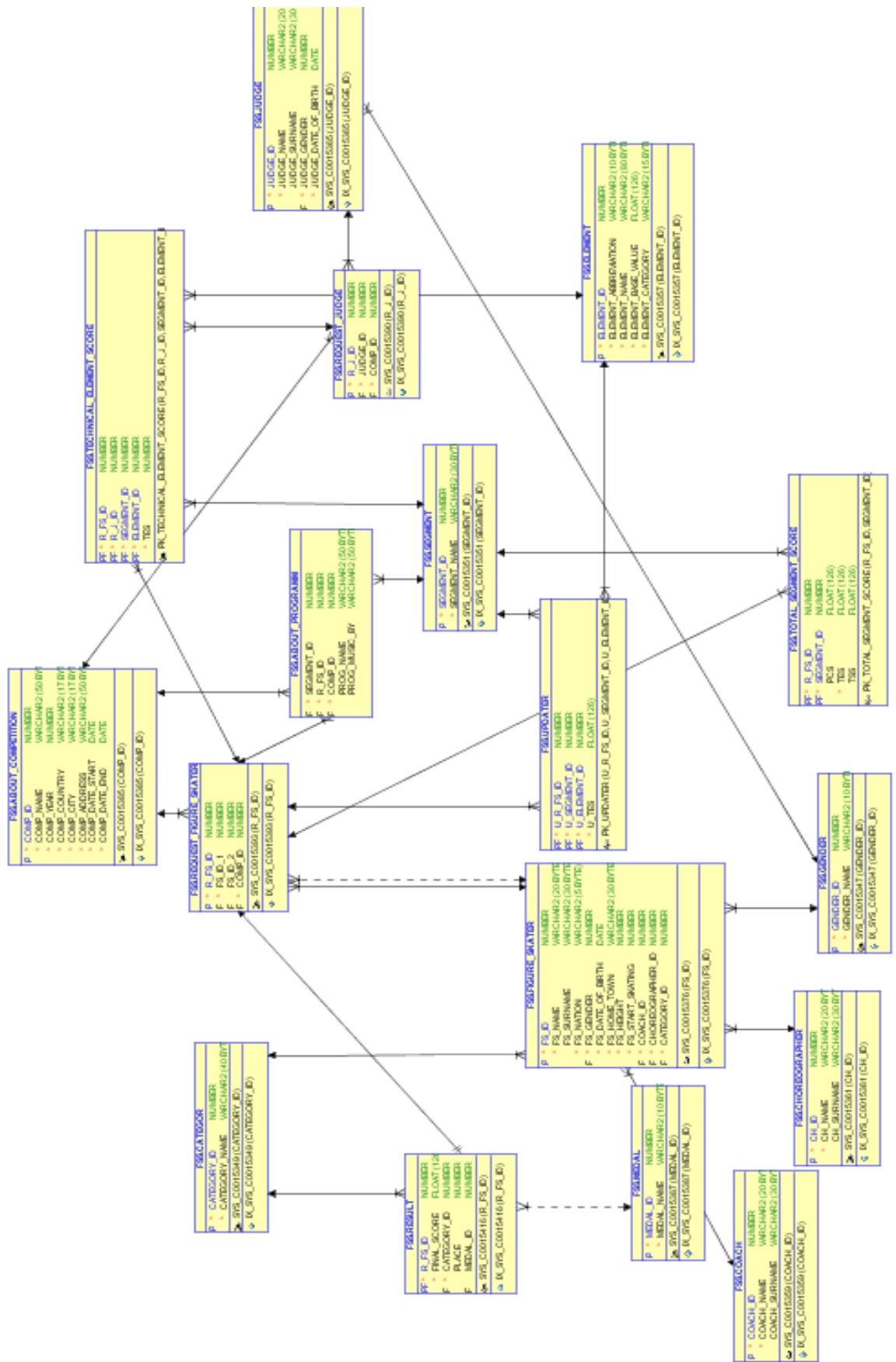


Рисунок 14 - Диаграмма физической модели данной базы данных

## 4 Создание клиентского приложения

Для создания клиентского приложения была выбрана среда разработки JDeveloper. **JDeveloper** - интегрированная среда разработки программного обеспечения, разработанная корпорацией Oracle. Предоставляет возможность для разработки на языках программирования Java, JavaScript, BPEL, PHP, SQL, PL/SQL и на языках разметки HTML, XML.

JDeveloper покрывает весь жизненный цикл разработки программного обеспечения от проектирования, кодирования, отладки, оптимизации и профилирования до его развёртывания. На рисунке 15 представлен полный жизненный цикл для разработки приложений.

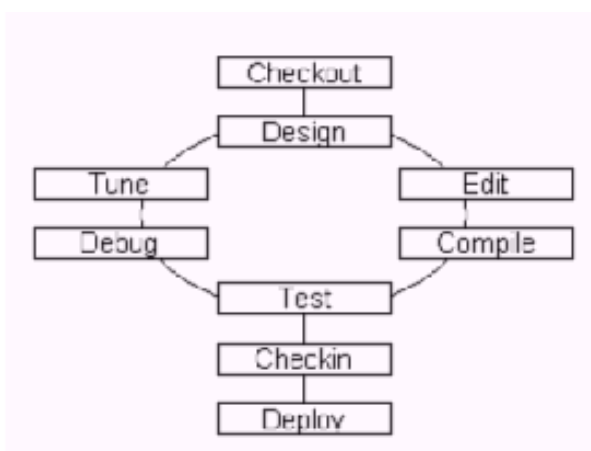


Рисунок 15 - Полный жизненный цикл для разработки приложений в среде JDeveloper

В типичном сценарии разработчик запускает JDeveloper, проверяет приложение с помощью системы управления исходным текстом и начинает цикл разработки. Разработчики UML-модели (UML modelers) помогают разработчику при конструировании приложения и, возможно, при генерации исходного текста. JDeveloper обеспечивает визарды (wizards) и редакторы – и визуальный, и кодоцентрированный (то есть, работающий с исходным текстом программы) – чтобы добавить функциональные возможности и различные инструментальные средства для компиляции, проверки, отладки и настройки приложения. Когда разработчик сочтет себя удовлетворенным, он может еще раз для проверки вернуть приложение назад в систему управления исходным текстом и, после достаточного испытания, развернуть его (приложение) в предназначенном ему месте.

Производитель отмечает в качестве основной задачи среды - максимальное использование возможностей визуального и декларативного подхода к разработке программного обеспечения в дополнение к удобной среде кодирования. Oracle JDeveloper интегрирована с Oracle ADF - Java EE-каркасом для создания коммерческих приложений на Java.

До версии 11g JDeveloper поставлялся в трёх редакциях (от младшей к старшей): *Java Edition*, *J2EE Edition* и *Studio Edition*. Каждая более старшая редакция включает возможности младшей, все редакции поставлялись бесплатно. JDeveloper 11g имеет только две редакции: *Java Edition* и *Studio Edition* (возможности J2EE Edition внесены в Studio Edition).

JDeveloper написана полностью на Java, поэтому работает на всех операционных системах, имеющих JDK.

Интерфейс данного приложения очень удобен для использования. С приложением могут работать два вида пользователей: пользователь и технический оператор. Техническому оператору предоставлены все права, а пользователи могут только просматривать информацию.

После запуска приложения открывается первая форма - форма входа, представленная на рисунке 16. На данном этапе производится авторизация пользователя (рисунок 17). При вводе неправильного пароля системы выдаст сообщение об ошибке (представлено на рисунке 18).

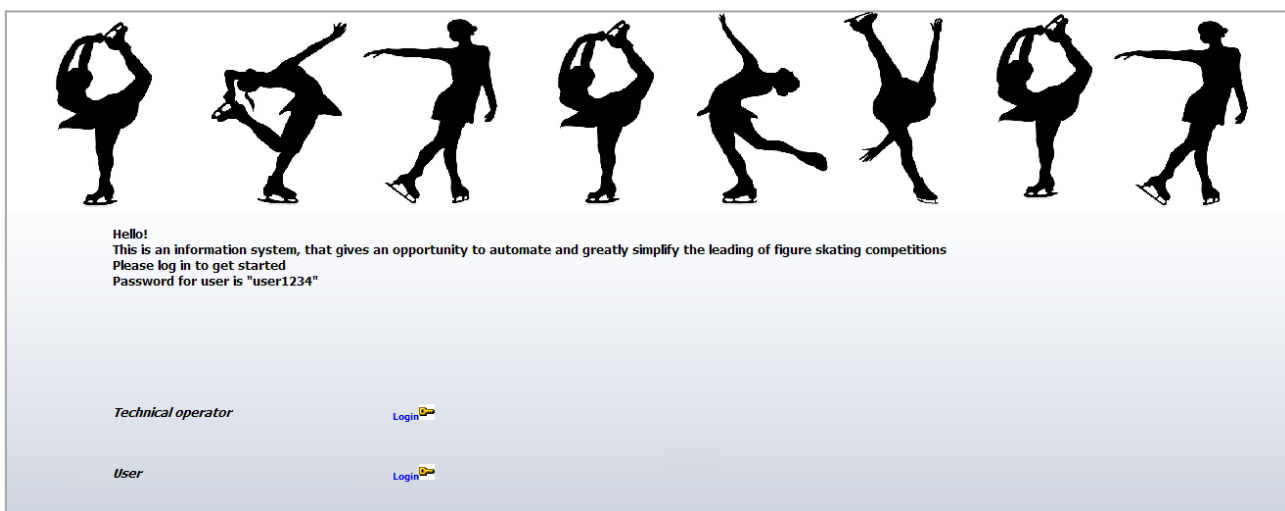


Рисунок 16 - Форма входа (Welcome page)

<b>Username:</b>	<input type="text" value="admin"/>
<b>Password:</b>	<input type="password" value="....."/>
<input type="button" value="Submit"/>	

Рисунок 17 - Авторизация пользователя. Ввод пароля

Invalid username or password

Рисунок 18 - Сообщение об ошибке при вводе неправильного пароля

При осуществлении входа под логином «Технический оператор (Techoper)» пароль представляет собой комбинацию - «techoper123». Технический оператор имеет права на всю базу данных. Он может изменять, добавлять и удалять любые данные из любой таблицы, контролировать систему учета оценок. При успешной авторизации под «Технический оператор (Techoper)» откроется форма, представленная на рисунке 19.

Рисунок 19 - Форма администратора

В форме, представленной на рисунке 19, технический оператор имеет право добавлять, изменять или удалять данные о Поле (Gender), Медали (Medal), Виде программы (Segment), Категории фигуриста (Category) и др. Кнопки для выполнения вышеописанных операций представлены на рисунке 20.



Рисунок 20 - Кнопки для добавления, удаления и сохранения данных

При осуществлении входа под логином «Пользователь (User)» пароль представляет собой комбинацию - «user1234». Пользователь может просматривать таблицы, разрешенные системой. При успешной авторизации под логином «Пользователь(User)» откроется форма пользователя, представленная на рисунке 21.



logout

About Competition Request FS Request Judge About Programm Result TSS **Final Results**

Place	CategoryName	FigureSkater	FinalScore	MedalName	CompName
1	Men	Stephane LAMBIEL	233,85	gold	European FS Cha...
2	Men	Evgeni PLUSHENKO	228,95	silver	European FS Cha...
3	Men	Florent AMODIO	194,25	bronze	European FS Cha...

Рисунок 21 - Форма пользователя

## **5 Безопасность жизнедеятельность**

### **5.1 Анализ потенциально опасных и вредных факторов, воздействующих на работу технического оператора в процессе обслуживания и работы с системой**

При работе с компьютером оператор подвергается воздействию ряда опасных и вредных производственных факторов: электромагнитных полей (диапазон радиочастот: ВЧ, УВЧ и СВЧ), инфракрасного и ионизирующего излучений, шума и вибрации, статического электричества и др [8].

Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей позы человека-оператора.

В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

#### **Освещение**

Правильно спроектированное и выполненное производственное освещение улучшает условия зрительной работы, снижает утомляемость, способствует повышению производительности труда, благотворно влияет на производственную среду, оказывая положительное психологическое воздействие на оператора, повышает безопасность труда и снижает травматизм.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности.

Согласно СНиП II-4-79 в помещениях вычислительных центров необходимо применить систему комбинированного освещения.

Требования к освещенности в помещениях, где установлены компьютеры, следующие: при выполнении зрительных работ высокой точности общая освещенность должна составлять 300лк, а комбинированная - 750лк; аналогичные требования при выполнении работ средней точности - 200 и 300лк соответственно.

Кроме того все поле зрения должно быть освещено достаточно равномерно – это основное гигиеническое требование. Иными словами, степень освещения помещения и яркость экрана компьютера должны быть примерно одинаковыми, т.к. яркий свет в районе периферийного зрения значительно увеличивает напряженность глаз и, как следствие, приводит к их быстрой утомляемости.

### Параметры микроклимата

В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. В санитарных нормах СН-245-71 установлены величины параметров микроклимата, создающие комфортные условия. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения (таблица 4) [8].

Объем помещений, в которых размещены работники вычислительных центров, не должен быть меньше 19,5м<sup>3</sup>/человека с учетом максимального числа одновременно работающих в смену. Нормы подачи свежего воздуха в помещения, где расположены компьютеры, приведены в таблице 5.

Т а б л и ц а 4 - Параметры микроклимата для помещений, где установлены компьютеры

Период года	Параметр микроклимата	Величина
Холодный	Температура воздуха в помещении	22...24°C
	Относительная влажность	40...60%
	Скорость движения воздуха	до 0,1м/с
Теплый	Температура воздуха в помещении	23...25°C
	Относительная влажность	40...60%
	Скорость движения воздуха	0,1...0,2 м/с

Т а б л и ц а 5 - Нормы подачи свежего воздуха в помещения, где расположены компьютеры

Характеристика помещения	Объемный расход подаваемого в помещение свежего воздуха, м <sup>3</sup> /на одного человека в час
Объем до 20м <sup>3</sup> на человека	Не менее 30
20...40м <sup>3</sup> на человека	Не менее 20
Более 40м <sup>3</sup> на человека	Естественная вентиляция

### Шум и вибрация

Шум ухудшает условия труда, оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Такие нарушения в работе ряда органов и систем организма человека могут вызвать негативные изменения в эмоциональном состоянии человека вплоть до стрессовых. Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума [выше 80 дБ (А)] на слух человека приводит к его частичной или полной потере [11].

В таблице 6 указаны предельные уровни звука в зависимости от категории тяжести и напряженности труда, являющиеся безопасными в отношении сохранения здоровья и работоспособности.

Т а б л и ц а 6 - Предельные уровни звука, дБ, на рабочих местах

Категория напряженности труда	Категория тяжести труда			
	I. Легкая	II. Средняя	III. Тяжелая	IV. Очень тяжелая
I. Мало напряженный	80	70	75	75
II. Умеренно напряженный	70	70	65	65
III. Напряженный	60	60	-	-
IV. Очень напряженный	50	50	-	-



Уровень шума на рабочем месте операторов видеоматериалов не должен превышать 50дБА, а в залах обработки информации на вычислительных машинах - 65дБА. Для снижения уровня шума стены и потолок помещений, где установлены компьютеры, могут быть облицованы звукопоглощающими материалами. Уровень вибрации в помещениях вычислительных центров может быть снижен путем установки оборудования на специальные виброизоляторы.

### **Электромагнитное излучения**

Большинство ученых считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Однако, исчерпывающих данных относительно опасности воздействия излучения от мониторов на работающих с компьютерами не существует и исследования в этом направлении продолжаются [8].

Допустимые значения параметров неионизирующих электромагнитных излучений от монитора компьютера представлены в таблице 7.

Т а б л и ц а 7 - Допустимые значения параметров неионизирующих электромагнитных излучений (в соответствии с СанПиН 2.2.2.542-96)

Наименование параметра	Допустимые значения
Напряженность электрической составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	10В/м
Напряженность магнитной составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	0,3А/м
Напряженность электростатического поля не должна превышать:	
для взрослых пользователей	20кВ/м
для детей дошкольных учреждений и учащихся средних специальных и высших учебных заведений	15кВ/м

Для снижения воздействия этих видов излучения рекомендуется применять мониторы с пониженным уровнем излучения (MPR-II, TCO-92, TCO-99), устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

## Требования к рабочему месту

Аспектами проектирования видеотерминальных рабочих мест, в частности, являются: высота рабочей поверхности, размеры пространства для ног, требования к расположению документов на рабочем месте (наличие и размеры подставки для документов, возможность различного размещения документов, расстояние от глаз пользователя до экрана, документа, клавиатуры и т.д.), характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места [12].

Главными элементами рабочего места программиста являются стол и кресло. Основным рабочим положением является положение сидя.

Рабочая поза сидя вызывает минимальное утомление программиста. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации.

Зона досягаемости рук в горизонтальной плоскости представлена на рисунке 22.

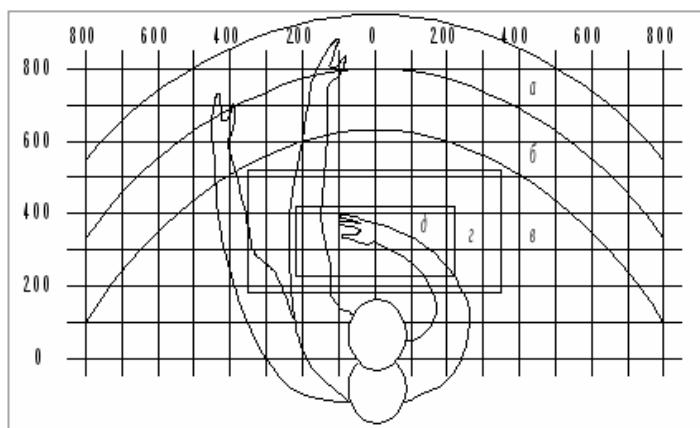


Рисунок 22 - Зоны досягаемости рук в горизонтальной плоскости

На рисунке 22:

- а) зона максимальной досягаемости;
- б) зона досягаемости пальцев при вытянутой руке;
- в) зона легкой досягаемости ладони;
- г) оптимальное пространство для грубой ручной работы;
- д) оптимальное пространство для тонкой ручной работы.

Оптимальное размещение предметов труда и документации в зонах досягаемости:

- дисплей размещается в зоне а (в центре);
- системный блок размещается в предусмотренной нише стола;
- клавиатура - в зоне г/д;
- «мышь» - в зоне в справа;
- сканер в зоне а/б (слева); принтер находится в зоне а (справа);

– документация, необходимая при работе - в зоне легкой досягаемости ладони – в;

– в выдвижных ящиках стола - литература, неиспользуемая постоянно.

На рисунке 23 показан пример размещения основных и периферийных составляющих ПК на рабочем столе программиста:

- 1) сканер;
- 2) монитор;
- 3) принтер;
- 4) поверхность рабочего стола;
- 5) клавиатура;
- 6) манипулятор типа «мышь».

Для комфортной работы стол должен удовлетворять следующим условиям [12]:

– высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники;

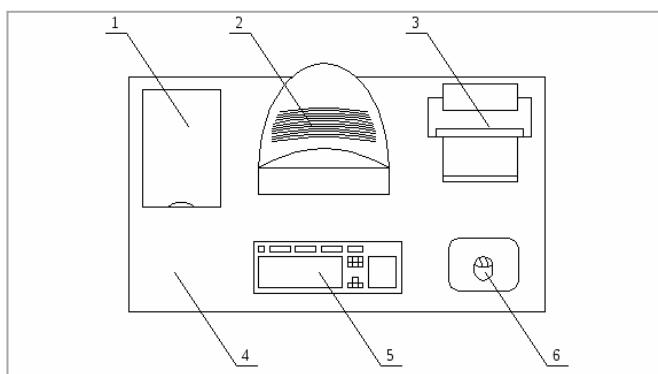


Рисунок 23. Размещение основных и периферийных составляющих ПК.

– нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги;

– поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста;

– конструкция стола должна предусматривать наличие выдвижных ящиков (не менее 3 для хранения документации, листингов, канцелярских принадлежностей);

– высота рабочей поверхности рекомендуется в пределах 680-760мм. Высота поверхности, на которую устанавливается клавиатура, должна быть около 650мм.

Большое значение придается характеристикам рабочего кресла. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420-550мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки - регулируемый.

Большое значение также придается правильной рабочей позе пользователя. При неудобной рабочей позе могут появиться боли в мышцах,

суставах и сухожилиях. Требования к рабочей позе пользователя видеотерминала следующие:

- голова не должна быть наклонена более чем на 20°;
- плечи должны быть расслаблены;
- локти - под углом 80°...100°;
- предплечья и кисти рук - в горизонтальном положении.

Причина неправильной позы пользователей обусловлена следующими факторами: нет хорошей подставки для документов, клавиатура находится слишком высоко, а документы - низко, некуда положить руки и кисти, недостаточно пространство для ног.

В целях преодоления указанных недостатков даются общие рекомендации: лучше передвижная клавиатура; должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и экрана, а также подставка для рук [12].

Во время пользования компьютером медики советуют устанавливать монитор на расстоянии 50-60 см от глаз. Специалисты также считают, что верхняя часть видеодисплея должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем когда он смотрит вниз. За счет этого площадь обзора значительно увеличивается, вызывая обезвоживание глаз. К тому же если экран установлен высоко, а глаза широко открыты, нарушается функция моргания. Это значит, что глаза не закрываются полностью, не омываются слезной жидкостью, не получают достаточного увлажнения, что приводит к их быстрой утомляемости.

### **Режим труда**

Как уже было неоднократно отмечено, при работе с персональным компьютером очень важную роль играет соблюдение правильного режима труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках [8].

В таблице 8 представлены сведения о регламентированных перерывах, которые необходимо делать при работе на компьютере, в зависимости от продолжительности рабочей смены, видов и категорий трудовой деятельности с ВДТ (видеодисплейный терминал) и ПЭВМ (в соответствии с СанПиН 2.2.2 542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ»).

Время перерывов дано при соблюдении указанных Санитарных правил и норм. При несоответствии фактических условий труда требованиям Санитарных правил и норм время регламентированных перерывов следует увеличить на 30%.

Т а б л и ц а 8 - Время регламентированных перерывов при работе на компьютере

Категория работы с ВДТ или ПЭВМ	Уровень нагрузки за рабочую смену при видах работы с ВДТ			Суммарное время регламентированных перерывов, мин	
	Группа А, количество знаков	Группа Б, количество знаков	Группа В, часов	При 8- часовой смене	При 12- часовой смене
І	до 20000	до 15000	до 2,0	30	70
ІІ	до 40000	до 30000	до 4,0	50	90
ІІІ	до 60000	до 40000	до 6,0	70	120

В соответствии со СанПиН 2.2.2 546-96 все виды трудовой деятельности, связанные с использованием компьютера, разделяются на три группы:

- группа А: работа по считыванию информации с экрана ВДТ или ПЭВМ с предварительным запросом;
- группа Б: работа по вводу информации;
- группа В: творческая работа в режиме диалога с ЭВМ.

Эффективность перерывов повышается при сочетании с производственной гимнастикой или организации специального помещения для отдыха персонала с удобной мягкой мебелью, аквариумом, зеленой зоной и т.п.

## 5.2 Расчетная часть

### Расчет уровня шума

Одним из неблагоприятных факторов производственной среды в информационно – вычислительного центра является высокий уровень шума, создаваемый печатными устройствами, оборудованием для кондиционирования воздуха, вентиляторами систем охлаждения в самих ЭВМ.

Для решения вопросов о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников [12]:

$$L_{\Sigma} = 10 \lg \sum_{i=1}^n 10^{0,1L_i} \quad (1)$$

где  $L_i$  – уровень звукового давления  $i$ -го источника шума;  
 $n$  – количество источников шума.

Полученные результаты расчета сравниваются с допустимым значением уровня шума для данного рабочего места. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в таблице 9.

Т а б л и ц а 9 - Уровни звукового давления различных источников

Источник шума	Уровень шума, дБ
Жесткий диск	40
Вентилятор	45
Монитор	17
Клавиатура	10
Принтер	45
Сканер	42

Обычно рабочее место оператора оснащено следующим оборудованием: винчестер в системном блоке, вентилятор(ы) систем охлаждения ПК, монитор, клавиатура, принтер и сканер.

Подставив значения уровня звукового давления для каждого вида оборудования в формулу (1), получим:

$$L_{\Sigma} = 10 \lg(10^4 + 10^{4,5} + 10^{1,7} + 10^1 + 10^{4,5} + 10^{4,2}) = 49,5 \text{ дБ}$$

Полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ (ГОСТ 12.1.003-83). И если учесть, что вряд ли такие периферийные устройства как сканер и принтер будут использоваться одновременно, то эта цифра будет еще ниже. Кроме того при работе принтера непосредственное присутствие оператора необязательно, т.к. принтер снабжен механизмом автоподачи листов.

### **Расчет кондиционирования**

Системы отопления и системы кондиционирования следует устанавливать так, чтобы ни теплый, ни холодный воздух не направлялся на людей. На производстве рекомендуется создавать динамический климат с определенными перепадами показателей. Температура воздуха у поверхности пола и на уровне головы не должна отличаться более, чем на 5 градусов. В производственных помещениях помимо естественной вентиляции

предусматривают приточно-вытяжную вентиляцию. Основным параметром, определяющим характеристики вентиляционной системы, является кратность обмена, т.е. сколько раз в час сменится воздух в помещении.

Расчет для помещения:

$V_{\text{вент}}$  - объем воздуха, необходимый для обмена;

$V_{\text{помещения}}$  - объем рабочего помещения.

Для расчета примем следующие размеры рабочего помещения:

– длина  $B = 7,35$  м;

– ширина  $A = 4,9$  м;

– высота  $H = 4,2$  м.

Соответственно объем помещения равен:

$$V_{\text{помещения}} = A \cdot B \cdot H = 151,3 \text{ м}^3$$

Необходимый для обмена объем воздуха  $V_{\text{вент}}$  определим исходя из уравнения теплового баланса:

$$V_{\text{вент}} \cdot C (t_{\text{уход}} - t_{\text{приход}}) \cdot Y = 3600 \cdot Q_{\text{избыт}} \quad (2)$$

где  $Q_{\text{избыт}}$  - избыточная теплота (Вт);

$C = 1000$  - удельная теплопроводность воздуха (Дж/кгК);

$Y = 1.2$  - плотность воздуха (мг/см).

Температура уходящего воздуха определяется по формуле:

$$t_{\text{уход}} = t_{\text{р.м.}} + (H - 2) \cdot t \quad (3)$$

где  $t = 1-5$  градусов - превышение  $t$  на 1 м высоты помещения;

$t_{\text{р.м.}} = 25$  градусов - температура на рабочем месте;

$H = 4,2$  м - высота помещения;

$t_{\text{приход}} = 18$  градусов.

Тогда, согласно формуле (3):

$$t_{\text{уход}} = 25 + (4.2 - 2) \cdot 2 = 29.4$$

$$Q_{\text{избыт}} = Q_{\text{изб.1}} + Q_{\text{изб.2}} + Q_{\text{изб.3}} \quad (4)$$

где  $Q_{\text{изб.1}}$  - избыток тепла от электрооборудования и освещения;

$Q_{\text{изб.2}}$  - теплопоступление от солнечной радиации;  
 $Q_{\text{изб.3}}$  - тепловыделения людей.

$$Q_{\text{изб.1}} = E \cdot p \quad (5)$$

где  $E$  - коэффициент потерь электроэнергии на теплоотвод ( $E = 0,55$  для освещения);

$p$  - мощность,

$$p = 40 \text{ Вт} \cdot 15 = 600 \text{ Вт.}$$

Тогда, согласно формуле (5):

$$Q_{\text{изб.1}} = 0,55 \cdot 600 = 330 \text{ Вт}$$

$$Q_{\text{изб.2}} = m \cdot S \cdot k \cdot Q_c \quad (6)$$

где  $m$  - число окон, примем  $m = 4$ ;  
 $S$  - площадь окна,  $S = 2,3 \cdot 2 = 4,6 \text{ м}^2$ ;  
 $k$  - коэффициент, учитывающий остекление.  
Для двойного остекления  $k = 0,6$ ;  
 $Q_c = 127 \text{ Вт/м}$  - теплопоступление от окон.

Тогда, согласно формуле (6):

$$Q_{\text{изб.2}} = 4,6 \cdot 4 \cdot 0,6 \cdot 127 = 1402 \text{ Вт}$$

$$Q_{\text{изб.3}} = n \cdot q \quad (7)$$

где  $q = 80 \text{ Вт/чел.}$ ;  
 $n$  - число людей, например,  $n = 15$ .

Тогда, согласно формуле (7):

$$Q_{\text{изб.3}} = 15 \cdot 80 = 1200 \text{ Вт}$$

Тогда, имея все данные, согласно формуле (4):

$$Q_{\text{избыт}} = 330 + 1402 + 1200 = 2932 \text{ Вт}$$

Из уравнения теплового баланса (2) следует:



$$V_{\text{вент}} = \frac{3600 \cdot 2932}{1000 \cdot (29,4 - 18)} = 926 \text{ м}^3$$

Оптимальным вариантом является кондиционирование воздуха, т.е. автоматическое поддержание его состояния в помещении в соответствии с определенными требованиями (заданная температура, влажность, подвижность воздуха) независимо от изменения состояния наружного воздуха и условий в самом помещении.

### Выбор вентилятора

Вентиляционная система состоит из следующих элементов:

1. Приточной камеры, в состав которой входят вентилятор с электродвигателем, калорифер для подогрева воздуха в холодное время года и жалюзная решетка для регулирования объема поступающего воздуха;
2. Круглого стального воздуховода длиной 1,5 м;
3. Воздухораспределителя для подачи воздуха в помещение.

Потери давления в вентиляционной системе определяются по формуле

$$H = R \cdot l + \frac{V^2 \cdot p}{2} \quad (8)$$

где  $H$  - потери давления, Па;

$R$  - удельные потери давления на трение в воздуховоде, Па/м;

$l$  - длина воздуховода, м;

$V$  - скорость воздуха, ( $V = 3 \text{ м/с}$ );

$p$  - плотность воздуха, ( $p = 1,2 \text{ кг/м}^3$ ).

Необходимый диаметр воздуховода для данной вентиляционной системы:

$$d = \frac{V}{900 \cdot V \cdot p} = \frac{926}{900 \cdot 3 \cdot 1,2} = 0,4 \text{ м}$$

Принимаем в качестве диаметра ближайшую большую стандартную величину - 0,45 м, при которой удельные потери давления на трение в воздуховоде -  $R=0,24 \text{ Па/м}$ .

Местные потери возникают в железной решетке ( $\xi=1,2$ ), воздухораспределителе ( $\xi=1,4$ ) и калорифере ( $\xi=2,2$ ). Отсюда, суммарный коэффициент местных потерь в системе:

$$\xi = 1,2 + 1,4 + 2,2 = 4,8$$

Тогда

$$H = 0,24 \cdot 0,15 + 4,8 \cdot \frac{9 \cdot 1,2}{2} = 26,28 \text{ Па}$$

С учетом 10 %-го запаса:

$$H = 110\% \cdot 26,28 = 28,01 \text{ Па}$$

$$V_{\text{вент}} = 110\% \cdot 1442 = 1586,2 \text{ м/ч}$$

По каталогу выбираем вентилятор осевой серии МЦ4: расход воздуха - 1600, давление - 40 Па, КПД - 65% , скорость вращения - 960 об/мин, диаметр колеса - 400 мм, мощность электродвигателя – 0,03 кВт.

### **5.3 Вывод**

В ходе описания данной части дипломной работы были проанализированы все вредные и потенциально-опасные факторы, влияющие на работу программиста, рассчитан уровень шума и кондиционирование помещения.

## 6 Технико-экономическое обоснование

### 6.1 Расчет затрат на разработку информационных технологий

Под информационными технологиями понимаются экономические информационные системы (ЭИС), программные продукты (ПП), информационные базы данных и т.д.

Расчет полных затрат на разработку проектного решения в виде информационных технологий ( $C_{\text{пi}}$ ) осуществляется по формуле:

$$C_{\text{пi}} = Z_{\text{фот}} + Z_{\text{сзи}} + M_i + A + P_{\text{ми}} + P_{\text{зи}} + P_{\text{ни}} \quad (9)$$

где  $Z_{\text{фот}}$  - общий фонд оплаты труда разработчиков, тенге;  
 $Z_{\text{сзи}}$  - отчисления по социальному налогу, тенге;  
 $M_i$  - затраты на материалы, тенге;  
 $A$  - амортизация;  
 $P_{\text{ми}}$  - затраты, связанные с эксплуатацией техники, тенге;  
 $P_{\text{зи}}$  - прочие затраты, тенге;  
 $P_{\text{ни}}$  - накладные расходы, тенге.

Размер фонда оплаты труда разработчиков ( $Z_{\text{фот}}$ ) рассчитывается по формуле:

$$Z_{\text{фот}} = Z_{\text{oi}} + Z_{\text{ди}} \quad (10)$$

где  $Z_{\text{oi}}$  - основная заработная плата, тенге;  
 $Z_{\text{ди}}$  - дополнительная заработная плата, тенге.

Затраты на оплату труда зависят от объема и трудоемкости разработки программного обеспечения.

Общий объем ( $V_0$ ) программного продукта определяется исходя из количества и объема функции, реализуемых программой:

$$V_0 = \sum_{j=1}^n V_i \quad (11)$$

где  $V_i$  - объем отдельной функции ПО;  
 $n$  - общее число функций.

Расчет уточненного объема ПО представлен в таблице 10.

$$V_0 \approx 10000$$

Т а б л и ц а 1 0 - Перечень и объем функций программного модуля

№ функции	Наименование (содержание)	Объем функции (LOC)	
		по каталогу $V_i$	уточненный $V_{vi}$
1	2	3	4
201	Генерация структуры базы данных	4300	1000
203	Формирование баз данных	2180	800
204	Обработка наборов и записей баз данных	2670	565
205	Обслуживание баз данных в пакетном режиме	1260	622
206	Обслуживание баз данных в интерактивном режиме	6950	571
207	Манипулирование данными	9550	3890
208	Организация поиска и поиск в базе данных	5480	1260
707	Графический вывод результатов	480	300
	Итого:	32870	9008

Общая трудоемкость небольших проектов рассчитывается по формуле:

$$T_0 = T_n \cdot K_c \cdot K_m \cdot K_n \quad (12)$$

где  $T_n$  - нормативная трудоемкость;  
 $K_c$  - коэффициент, учитывающий сложность ПО;  
 $K_t$  - поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей;  
 $K_n$  - коэффициент, учитывающий степень новизны ПО.

Коэффициент сложности определяется на основе данных представленных таблицы 11. Посредством коэффициента сложности учитываются затраты труда, связанные со сложностью ПП.

В разрабатываемом дипломном проекте  $K_i$ , за счет наличия у программного модуля одновременно трех характеристик:

- режим работы в реальном времени;
- управление удаленными объектами;
- существенное распараллеливание вычислений.

Принимаем  $K_c=0,18$ .

Коэффициент, учитывающий степень использования при разработке ПО стандартных модулей ( $K_m$ ).

Т а б л и ц а 11 – Дополнительные коэффициенты сложности ПО

Характеристика ПО	Значения $K_c$
1. Функционирование ПО в расширенной операционной среде (связь с другими ПО)	0,08
2. Интерактивный доступ	0,06
3. Обеспечение хранения, ведения и поиска данных в сложных структурах	0,07
4. Наличие у ПО одновременно нескольких характеристик по табл.Г4.1, приложение Г	0,12
4.1 2 характеристики	0,18
4.2 3 характеристики	0,26
4.3 Свыше 3-х характеристик	

Степень использования в разрабатываемом ПО стандартных модулей определяется их удельным весом в общем объеме проектируемого продукта. В данном дипломном проекте степень охвата реализуемых функций разрабатываемого ПО стандартными модулями, типовыми программами и ПО до 20%, следовательно  $K_m = 0,9$ .

Поправочный коэффициент, учитывающий новизну разрабатываемого ПО ( $K_n$ ) определяется на основе данных представленных в таблице 12 и составляет 1,0.

Т а б л и ц а 12 - Поправочные коэффициенты, учитывающие новизну ПО( $K_n$ )

Категория новизны	Степень новизны	Использование		Значение $K_n$
		На основе нового типа ПК	В среде новой ОС	
А	Принципиально новые ПО, не имеющие доступных аналогов	+	+	1,75
		-	+	1,6
		+	-	1,2
		-	-	1,0
Б	ПО, являющиеся развитием определенного параметрического ряда ПО	+	+	1,0
		-	-	0,9
		+	-	0,8
В	ПО, являющиеся развитием определенного параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации ПК и ОС	-	-	0,7

Нормативная трудоемкость ПО ( $T_n$ ) определяется на основе принятого в расчет  $V_y$  и категории сложности, которая уточняется с учетом сложности и

новизны проекта и степени использования стандартных модулей при разработке [13].

В соответствии с этим, согласно укрупненным нормам времени на разработку ПО ( $T_H$ ) в зависимости от уточненного объема ПО ( $V_0$ ) и группы сложности (Приложение В) : объем ПО(строки исходного кода, LOC) 10000, категория сложности ПО 1-я -  $T_H = 327$ , категория сложности ПО 39.

Следовательно  $T_0$  будет равно:

$$T_0 = 327 \cdot 0,18 \cdot 0,9 \cdot 1 = 52,97 \text{ (чел./дн.)}$$

Численность исполнителей проекта ( $Ч_p$ ) рассчитывается по формуле:

$$Ч_p = \frac{T_0}{T_p \cdot \Phi_{эф}} \quad (13)$$

где  $\Phi_{эф}$  - эффективный фонд времени работы одного работника в течение года (дн.);

$T_0$  - общая трудоемкость разработки проекта (чел./дн.);

$T_p$  - срок разработки проекта (лет).

Срок разработки проекта ( $T_p$ ) определяется по формуле:

$$T_p = \frac{T_0}{Ч_p \cdot \Phi_{эф}} \quad (14)$$

где  $Ч_p$  - плановое число разработчиков.

Эффективный фонд времени работы одного работника ( $\Phi_{эф}$ ) рассчитывается по формуле:

$$\Phi_{эф} = D_r - D_{п} - D_{в} - D_0 \quad (15)$$

где  $D_r$  - количество дней в году;

$D_{п}$  - количество праздничных дней в году;

$D_{в}$  - количество выходных дней в году;

$D_0$  - количество дней отпуска.

В соответствии производственным календарем на 2014 год.:  $D_r=365$ ;  $D_{п} = 14$  ;  $D_{в} = 103$ ;  $D_0 = 10$ , то по формуле (15) получим:

$$\Phi_{эф} = 365 - 14 - 103 - 10 = 228 \text{ дней}$$

Плановое число разработчиков  $Ч_p = 1$ , следовательно, по формуле (14):

$$T_p = 52,97 / (1 \cdot 228) = 0,23 \text{ года} = 84 \text{ дня}$$

Таким образом, согласно произведенным расчетам и в соответствии с формулой (13):

$$Ч = 52,97 / (0,23 \cdot 228) = 1 \text{ чел}$$

Основная заработная плата исполнителей на конкретное ПО рассчитывается по формуле:

$$\sum_{i=1}^n Z_{oi} = \sum T_{ci} \cdot T_{ч} \cdot \Phi_{п} \cdot K \quad (16)$$

где  $n$  - количество исполнителей, занятых разработкой конкретного ПО;  
 $T_{ci}$  - часовая тарифная ставка  $i$ -го исполнителя (тыс.тенге);  
 $\Phi_{п}$  - плановый фонд рабочего времени  $i$ -го исполнителя (дней), 84 дня;  
 $T_{ч}$  - количество часов работы в день (час), 8 часов;  
 $K$  - коэффициент премирования, составляет 1,2.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов-исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности (таблица 13).

Т а б л и ц а 13 - Сведения по работникам, задействованным в проекте

Специалист - Исполнитель	Количество, человек	Заработная плата в месяц, тенге
Программист	1	150 000
Итого		150 000

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленную при 40-часовой недельной норме рабочего времени расчетную среднемесячную норму рабочего времени в часах ( $\Phi_p$ ):

$$T_{ч} = \frac{T_{м}}{\Phi_{п}} \quad (17)$$

где  $T_{ч}$  - часовая тарифная ставка (тыс.тенге);  
 $T_{м}$  - месячная тарифная ставка (тыс.тенге).

Таким образом:

$$T_{\text{ч}} = \frac{150\,000}{168} = 892,9 \text{ тенге/час}$$

По формуле (16) можно рассчитать основную заработную плату исполнителей:

$$Z_{oi} = 892,9 \cdot 8 \cdot 84 \cdot 1,2 = 720\,034,56 \text{ тенге}$$

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

$$Z_{di} = Z_{oi} \cdot N_d / 100 \quad (18)$$

где  $N_d$  - коэффициент дополнительной заработной платы разработчиков 22%.

$$Z_{di} = 720\,034,56 \cdot 0,22 = 158\,407,6 \text{ тенге}$$

Социальный налог составляет 11% (ст. 358 п. 1 НК РК) от дохода работника, и рассчитывается по формуле:

$$Z_{czi} = (\text{ФОТ} - \text{ПО}) \cdot 11\% \quad (19)$$

где ПО - пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются:

$$\text{ПО} = \text{ФОТ} \cdot 10\%. \quad (20)$$

Таким образом:

$$\text{ПО} = 878\,442,16 \cdot 0,1 = 87\,844,216 \text{ тенге}$$

$$Z_{czi} = (878\,442,16 - 87\,844,216) \cdot 0,11 = 86\,965,77 \text{ тенге}$$

Затрат на материалы определяются по формуле:

$$M_i = (Z_{\text{осн}} \cdot N_{\text{мз}}) / 100\% \quad (21)$$

где  $N_{\text{мз}}$  - норма расхода материалов от основной заработной платы (3-5%).

$$M_i = 720\,034,56 \cdot 0,05 = 36\,001,73 \text{ тенге}$$



Амортизационные отчисления производятся по установленным нормам амортизации, выражаются, в процентах к балансовой стоимости оборудования и рассчитываются по формуле (22):

$$A = \frac{C_{\text{обор}} \cdot N_A \cdot N}{365 \cdot 100} \quad (22)$$

где  $N_A$  - норма амортизации (25 %);  
 $C_{\text{обор}}$  - первоначальная стоимость оборудования;  
 $N$  - фактический срок эксплуатации оборудования, 60 дней;

Данные по стоимости оборудования представлены в таблице 14.

Т а б л и ц а 14 – Стоимость оборудования одного ПК с периферией

Наименование	Цена с НДС, тенге	Цена без НДС, тенге
Монитор	32 000	28 160
Материнская плата	12 800	11 264
Процессор	30 000	26 400
Видеокарта	23 000	20 240
HDD	12 000	10 560
DVD-RW	6 500	5 720
CPU Cooler	5 500	4 840
Оперативная память	8 000	7 040
Клавиатура	1 000	880
Мышь	1 000	880
Принтер	8 000	7 040
Итого:	139800	123024

$$C_{\text{обор}} = 139\,800 \text{ тенге}$$

Тогда, согласно формуле (22):

$$A = \frac{139\,800 \cdot 25 \cdot 60}{365 \cdot 100} = 5745,2$$

Расходы по статье «Машинное время» ( $P_{mi}$ ) включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам (в машино-часах) на 100 строк исходного кода ( $H_{mb}$ ) машинного времени в зависимости от характера решаемых задач и типа ПК.

$$P_{mi} = C_{mi} \cdot (V_{oi}/100) \cdot H_{mb} \quad (23)$$

где  $C_{mi}$  - цена одного машино-часа (тыс.тенге);  
 $V_{oi}$  - общий объем ПО (строк исходного кода);  
 $H_{mb}$  - норматив расхода машинного времени на отладку 100 строк  
исходного кода (машино-часов).

Учитывая данные из Приложения Д, наименование подсистемы АС и ДОС - 1 и средний расход машинного времени, ч/100 сток кода составляет 12 [13].

Таким образом:

$$P_{Mi} = 892,9 \cdot (10\ 000 / 100) \cdot 11 = 982\ 190 \text{ тенге}$$

Расходы по статье «Прочие затраты» ( $P_{zi}$ ) на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате:

$$P_{zi} = Z_{oi} \cdot H_{pz} / 100 \quad (24)$$

где  $H_{pz}$  - норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20% .

Подставляем все данные в формулу (24) получаем:

$$P_{zi} = 720\ 034,56 \cdot 0,2 = 144\ 006,91 \text{ тенге}$$

Затраты по статье «Накладные расходы» ( $P_{ni}$ ), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных(экспериментальных) производств, а также с расходами на общехозяйственные нужды ( $P_{ni}$ ), относятся на конкретное ПО по нормативу ( $H_{pn}$ ) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации:

$$P_{ni} = Z_{oi} \cdot H_{pn} / 100\% \quad (25)$$

где  $P_{ni}$  - накладные расходы на конкретную ПО (тыс.тенге);

$H_{pn}$  - норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70% .

Подставляем все данные в формулу (25) получаем:

$$P_{ni} = 720\ 034,56 \cdot 0,7 = 504\ 024,19 \text{ тенге}$$

$$C_{ni} = 2\ 637\ 375,96 \text{ тенге}$$

Результаты выполненных расчетов представлены в виде таблицы 15 и рисунка 24 .

Т а б л и ц а 15 - Затраты на разработку

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{\text{ФОТ}}$	878 442,16	33,31
Социальный налог	$Z_{\text{сзи}}$	86 965,77	3,30
Материалы	$M_i$	36 001,73	1,37
Амортизация	$A$	5745,2	0,22
Машинное время	$P_{\text{ми}}$	982 190	37,24
Прочие затраты	$P_{\text{зи}}$	144 006,91	5,46
Накладные расходы	$P_{\text{ни}}$	504 024,19	19,11
Итого:		2 637 375,96	100,00

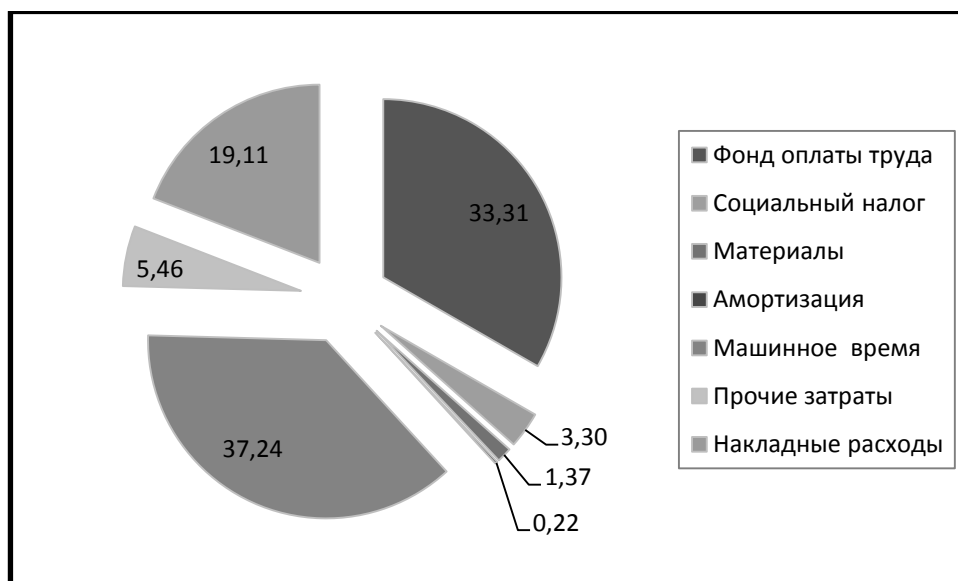


Рисунок 24 - Результаты выполнения расчетов

## 6.2 Расчет цены программного продукта

Расчет цены ПП в организационно-экономической части дипломной работы предлагается производить следующим образом:

– если ПП разработан одной организацией по заказу другой и не предназначен для тиражирования, то затраты на разработку ПП считаются его себестоимостью, и при формировании цены применяется затратный метод;

– если ПП предназначен для тиражирования, то конечная цена определяется путем экспертных оценок на основании ценностного подхода с учетом текущих цен конкурентов (если существуют аналогичные ПП).

Расчет цены ПП, который разработан одной организацией по заказу другой и не предназначен для тиражирования, осуществляется по формуле:

$$C_{ПП} = Z_{РПР} + П_n + НДС \quad (26)$$

где  $C_{ПП}$  - цена программного продукта, тенге;

$Z_{РПР}$  - затраты на разработку проектного решения, в данном случае программного продукта, тенге;

$П_n$  - планируемая прибыль, тенге;

$НДС$  - налог на добавленную стоимость, тенге.

Планируемая прибыль рассчитывается по формуле:

$$П_n = Z_{РПР} + R_{НПП} \quad (27)$$

где  $R_{НПП}$  - нормативная рентабельность ПП, определяемая организацией.

$НДС$ , начисленный на ПП, определяется следующим образом:

$$НДС = (Z_{РПР} + П_n) \cdot k_{НДС} \quad (28)$$

где  $k_{НДС}$  - ставка налога на добавленную стоимость.

Подставляем все значения в формулы (26) - (28) и получаем:

По формуле (27) учитывая, что  $Z_{РПР} = C_{пп}$ ,  $R_{НПП}$  - это процент рентабельности по отношению к себестоимости составляет 20%:

$$R_{НПП} = C_{пп} \cdot 0,20 = 2\,637\,375,96 \cdot 0,20 = 527\,475,19 \text{ тенге}$$

$$П_n = 2\,637\,375,96 + 527\,475,19 = 3\,164\,851,15 \text{ тенге}$$

Подставив данные в формуле (28) получаем:

$$НДС = (2\,637\,375,96 + 3\,164\,851,15) \cdot 0,12 = 696\,267,25 \text{ тенге}$$

Подставив данные в формуле (26) получаем:

$$C_{\text{III}} = 2\,637\,375,96 + 3\,164\,851,15 + 696\,267,25 = 6\,498\,494,36 \text{ тенге}$$

### **6.3 Вывод**

Стоимость разработки составила 2 637 375,96 тенге.

Наибольшую долю составляет машинное время.

Рыночная стоимость данного проекта (учитывая существующие аналоги) составляет 8 000 000 тенге. Таким образом, разработка и внедрение данного проекта выгодно не только с технической стороны, но и с финансовой.

Данная работа не преследует прибыли, главная цель автоматизация проведения соревнований по фигурному катанию.

## **Заключение**

В ходе выполнения данной дипломной работы была разработана база данных «Автоматизация проведения соревнований по фигурному катанию» и обеспечена ее безопасность.

При проектировании работы, были учтены все основные функции данной базы данных. Данная база данных проектировалась на Oracle 11g, так это наиболее удовлетворяющая по функциональным возможностям среда проектирования. В данной базе данных учтены права пользователей, и доступ к информации ограничен. Так, например, изменять, добавлять и удалять информацию из таблиц может только технический оператор. В то время как пользователь может только просматривать информацию.

База данных содержит множество триггеров и хранимых процедур, которые описаны выше, это позволяет ускорить процесс обработки информации, а так же упростить использование программы на большом количестве компьютеров. Система предполагает сопровождение программы на долгий период, так как создана с учетом вносимых желаемых изменений заказчика.

Для защиты был использован продукт Oracle Advanced Security, а именно, Прозрачное шифрование (TDE) и протокол аутентификации Radius.

## Список использованной литературы

- 1) Сайт [http://www.partner.fors.ru/registered/oracle\\_advanced\\_security](http://www.partner.fors.ru/registered/oracle_advanced_security).
- 2) Сайт <http://oracle.axoft.ru/catalog/product.php?ID=343>.
- 3) Арап Нанда. Прозрачное шифрование данных // Журнал Oracle Magazine, 2005.
- 4) Sumit Jeloka Oracle Database Advanced Security Administrator's Guide 11g Release 2 (11.2), 2014.
- 5) Пит Финниган. Введение в аудит-возможности Oracle. - Журнал Oracle Magazine, 2008.
- 6) Дубовцев В.А. Безопасность жизнедеятельности: Учеб. пособие для дипломников. - Киров: изд. КирПИ, 1992.
- 7) Мотузко Ф.Я. Охрана труда. - М.: Высшая школа, 1989.
- 8) Безопасность жизнедеятельности. /Под ред. Н.А. Белова. - М.: Знание, 2000.
- 9) Самгин Э.Б. Освещение рабочих мест. - М.: МИРЭА, 1989.
- 10) Справочная книга для проектирования электрического освещения. / Под ред. Г.Б. Кнорринга. - Л.: Энергия, 1976.
- 11) Борьба с шумом на производстве: Справочник / Е.Я. Юдин, Л.А. Борисов; Под общ. ред. Е.Я. Юдина. - М.: Машиностроение, 1985.
- 12) Зинченко В.П. Основы эргономики. - М.: МГУ, 1979.
- 13) Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 - Вычислительная техника и программное обеспечение Еркешева З.Д, Боканова Г.Ш. 2014

## Приложение А

### Классификация типов программного обеспечения

Таблица А1 - Классификация типов программного обеспечения

Код типа	Наименование типа ПО	Состав и содержание типа ПО
1.0	ПО общего назначения	1.1 ПО СУБД. 1.2 ПО систем ведения линейных файлов. 1.3 ПО ведения баз данных и линейных файлов. 1.4 ПО информационно-поисковых и информационно-справочных систем. 1.5 ПО ввода информации. 1.6 ПО мониторов телеобработки и сетей ПЭВМ. 1.7 ПО окружения СУБД, расширяющие возможности существующих СУБД. 1.8 ПО, расширяющие возможности обработки.
2.0	ПО технологии автоматизации программирования и проектирования АСУ	2.1 ПО автоматизации проектирования для автоматизации проектирования различных АСУ. 2.2 ПО технологии программирования. 2.3 ПО автоматизации программирования (для автоматизации процессов обработки и вывода информации). 2.4 ПО, расширяющие существующие языки программирования для повышения их компактности и простоты пользования. 2.5 ПО общего назначения, функционально-ориентированные. 2.6 ПО автоматического программирования. Реализуют различные классы экономико-математических методов и являются системами общего назначения, которые применяются в различных АСУ, для научно-технических расчетов и исследований.
3.0	ПО методоориентированных расчетов	3.1 ПО оптимизационных расчетов (обеспечивают решение различного класса задач оптимального планирования и управление производством). 3.2 ПО статистического анализа и прогнозирования (для прогнозирования ТЭП, спроса и т.д.). 3.3 ПО сетевого планирования и управления. 3.4 ПО общей математики. 3.5 ПО имитационного моделирования.



Продолжение приложения А

Окончание таблицы А1

Код типа	Наименование типа ПО	Состав и содержание типа ПО
4.0	ПО организации вычислительного процесса	<p>Автоматизация процесса ведения наборов данных, при обеспечении их надежного и систематизированного хранения.</p> <p>Повышение производительности ПЭВМ и пользователей ПО.</p> <p>Формирование и выдача отчетов о работе ПЭВМ.</p> <p>Оперативный контроль системы и ресурсов.</p> <p>Для автоматизации обработки экономических данных, при этом выделяются ПО, несущие функциональную нагрузку в АСУ.</p> <p>ПО данного типа выполнены в основном автономно.</p>
5.0	ПО функционального назначения	<p>5.1 ПО системы ПОП и СУП.</p> <p>5.2 ПО оперативного управления основным производством.</p> <p>5.3 ПО управления технической подготовкой производства.</p> <p>5.4 ПО бухгалтерского учета и управления финансами.</p> <p>5.5 ПО управления кадрами.</p> <p>5.6 ПО, не вошедшие ни в один из перечисленных типов ПО.</p>
4.0	ПО организации вычислительного процесса	<p>Автоматизация процесса ведения наборов данных, при обеспечении их надежного и систематизированного хранения.</p> <p>Повышение производительности ПЭВМ и пользователей ПО.</p> <p>Формирование и выдача отчетов о работе ПЭВМ.</p> <p>Оперативный контроль системы и ресурсов.</p> <p>Для автоматизации обработки экономических данных, при этом выделяются ПО, несущие функциональную нагрузку в АСУ.</p> <p>ПО данного типа выполнены в основном автономно.</p>
5.0	ПО функционального назначения	<p>5.1 ПО системы ПОП и СУП.</p> <p>5.2 ПО оперативного управления основным производством.</p> <p>5.3 ПО управления технической подготовкой производства.</p> <p>5.4 ПО бухгалтерского учета и управления финансами.</p> <p>5.5 ПО управления кадрами.</p> <p>5.6 ПО, не вошедшие ни в один из перечисленных типов ПО.</p>

## Приложение Б

### Каталог функций программного обеспечения

Таблица Б1 - Каталог функций программного обеспечения

№ п/п	Наименование (содержание) функций	Объем функций (строк исходного кода)		
		С использованием среды разработки приложений		
		Delphi (Borland)	C++ Builder (Borland)	Visual C++ (Microsoft)
<b>1. Ввод, анализ входной информации, генерация кодов и процессор входного языка</b>				
101	Организация ввода информации	100	110	150
102	Контроль, предварительная обработка и ввод информации	550		
103	Анализ входного языка (синтаксический и семантический)	630	660	980
104	Преобразование операторов входного языка и команды другого языка	1050	1050	980
105	Обработка входного заказа и формирование таблиц	750	900	1340
106	Преобразование входного языка в машин-ные команды (транслятор, препроцессор, макрогенератор)	1300	4300	5100
107	Синтаксический и семантический анализ входного языка и генерация кодов команд	8700	5400	5400
108	Процессор языка	3000	2300	2300
109	Организация ввода/вывода информации в интерактивном режиме	220	220	320
110	Организация ввода/вывода информации с сети терминалов	3680	3340	3200
110	Управление вводом/выводом	2700	2900	2400
<b>2. Формирование, введение и обслуживание баз данных</b>				
201	Генерация структуры базы данных	3450	3950	4300
202	Генерация структуры базы данных	1540	1610	2060
203	Формирование баз данных	1700	1750	2180
204	Обработка наборов и записей базы данных	2050	2350	2670
205	Обслуживание базы данных в пакетном режиме	1030	1100	1260
206	Обслуживание базы данных в интерактивном режиме	3800	4400	6950
207	Манипулирование данными	8400	8670	9550
208	Организация поиска и поиск в базе данных	5230	5460	5480
209	Реорганизация базы данных	130	190	220
210	Загрузки базы данных	3150	2950	2780

Продолжение приложения Б

Продолжение таблицы Б1

№ п/п	Наименование (содержание) функций	Объем функций (строк исходного кода)		
		С использованием среды разработки приложений		
		Delphi (Borland)	C++ Builder (Borland)	Visual C++ (Microsoft)
<b>3. Формирование и обработка файлов</b>				
301	Формирование последовательного файла	340	360	290
302	Сортировка файла	340	360	290
303	Автоматическая сортировка файлов	1040	1150	930
304	Обслуживание файлов	520	540	420
305	Обработка файлов	750	800	720
306	Обработка файлов в диалоговом режиме	2400	2600	3050
307	Совместная обработка группы файлов	4900	5300	6180
308	Управление файлами	5130	5380	5750
309	Формирование файла	1100	1080	1020
<b>4. Генерация программ и ПО, а также настройка ПО</b>				
401	Генерация рабочих программ	3680	4120	3360
402	Генерация программ по описанию пользователей	10870	12330	9880
403	Формирование служебных таблиц	570	620	1070
404	Система генерации ПО	5120	5340	4980
405	Система генерации ПО	250	300	370
<b>5. Управление ПО, компонентами ПО и внешними устройствами</b>				
501	Монитор ПО (управление работой компонентов)	350	360	740
502	Монитор системы (управление работой комплекса ПО)	3750	3880	7740
503	Управление внешними устройствами и объектами	6850	7340	5900
504	Обработка прерываний	890	730	540
505	Управление внешней памятью	250	210	200
506	Обработка ошибочных и сбойных ситуаций	430	410	410
507	Обеспечение интерфейса между компонентами	730	750	970
<b>6. Отладка прикладных программ, обмен информацией между магнитным диском и магнитной лентой, вспомогательные программные функции</b>				
601	Отладка прикладных программ в интерактивном режиме	4500	4700	4300
602	Обмен информацией между магнитным диском и магнитной лентой	-	-	-
603	Копирование наборов данных на магнитной ленте и восстановление	-	-	-

Продолжение приложения Б

Окончание таблицы Б1

№ п/п	Наименование (содержание) функций	Объем функций (строк исходного кода)		
		С использованием среды разработки приложений		
		Delphi (Borland)	C++ Builder (Borland)	Visual C++ (Microsoft)
6. Отладка прикладных программ, обмен информацией между магнитным диском и магнитной лентой, вспомогательные программные функции				
604	Справка и обучение	680	680	720
605	Вспомогательные и сервисные программы	460	490	580
7. Расчетные задачи, формирование и вывод на внешние носители (АЦПУ) документов сложной формы и файлов				
701	Матстатистика и прогнозирование	8370	9570	9320
702	Расчетные задачи (расчет режимов обр-ки)	12600	15300	14800
703	Расчет показателей	410	500	460
704	Процессор отчетов	1070	1230	3200
705	Формирование и вывод на внеш.носители	2650	2850	3500
706	Предварит.обработка и печать файлов	540	560	470
707	Графический вывод результатов	300		480
708	Интерактивный редактор текста	3800	3910	4540
709	Измерение состояния ресурсов в интерактивной системе	650	440	480

**Приложение В**  
**Укрупненные нормы времени на разработку ПО в зависимости от**  
**уточненного объема ПО и группы сложности ПО**

Таблица В1 – Укрупненные нормы времени на разработку ПО ( $T_H$ ) в зависимости от уточненного объема ПО ( $V_y$ ) и группы сложности ПО (чел./дн.)

Объем ПО(строки исходного кода, LOC)	Категории сложности ПО			Категории сложности ПО
	1-я	2-я	3-я	
200	—	—	21	1
300	—	—	23	2
400	—	—	25	3
500	—	—	27	4
600	—	33	28	5
700	—	36	30	6
800	—	38	32	7
900	—	40	34	8
1000	51	43	36	9
1200	54	45	38	10
1400	57	48	40	11
1600	60	50	42	12
1800	64	54	45	13
2000	68	57	48	14
2200	73	61	51	15
2400	76	64	54	16
2600	81	68	57	17
2800	86	72	60	18
3000	91	76	64	19
3200	97	81	68	20
3400	103	86	72	21
3600	110	92	77	22
3800	117	98	82	23
4000	124	104	87	24
4200	133	111	93	25
4400	141	118	99	26
4600	151	126	105	27
4800	160	134	112	28
5000	170	142	119	29
5500	182	152	127	30
6000	194	162	135	31
6500	206	172	144	32
7000	220	184	154	33
7500	235	196	164	34
8000	252	210	175	35
8500	268	224	187	36
9000	288	240	200	37
9500	307	256	214	38
10000	327	273	228	39
11000	349	291	243	40

Продолжение приложения В

Окончание таблицы В1

Объем ПО(строки исходного кода, LOC)	Категории сложности ПО			Категории сложности ПО
	1-я	2-я	3-я	
12000	374	312	260	41
13000	399	333	278	42
14000	427	356	297	43
15000	456	380	317	44
16000	487	406	339	45
18000	520	434	362	46
20000	556	464	387	47
22000	595	496	414	48
24000	636	530	442	49
26000	679	566	472	50
28000	727	606	505	51
30000	775	646	540	52
32000	830	692	577	53
34000	888	740	617	54
36000	950	792	660	55
38000	1016	847	706	56
40000	1087	906	755	57
42000	1161	968	807	58
44000	1242	1035	863	59
46000	1328	1107	923	60
48000	1420	1184	987	61
50000	1620	1267	1056	62

## Приложение Г

### Характеристики категорий сложности ПО

Таблица Г1 – Характеристики категорий сложности ПО

Категории сложности	Характеристики ПО
1	ПО, обладающие одной или несколькими из следующих характеристик: 1) Наличие сложного интеллектуального языкового интерфейса с пользователем. 2) Режим работы в реальном времени. 3) Обеспечение телекоммуникационной обработки данных и управление удаленными объектами. 4) Машинная графика. 5) Многомашинные комплексы. 6) Обеспечение существенного распараллеливания вычислений
2	ПО, обладающие одной или несколькими из следующих характеристик: 1) Оптимизационные расчеты. 2) Обеспечение настройки ПО на изменения структур входных и выходных данных. 3) Настройка ПО на нестандартную конфигурацию технических средств. 4) Обеспечение переносимости ПО. 5) Реализация особо сложных инженерных и научных расчетов.
3	ПО, не обладающие перечисленными выше характеристиками.

**Приложение Д**  
**Оценка значений среднего машинного времени на отладку 100 строк  
исходного кода без применения ПО**

Таблица Д1 – Оценка значений среднего машинного времени на отладку 100 строк исходного кода без применения ПО

Наименование подсистемы АС и СОД	Средний расход машинного времени, ч/100 строк кода
1. Общесистемные задачи: ведение линейных файлов, информационно-поисковые системы и информационно-справочные системы, сбор информации, ввод информации, расширение возможностей средств обработки данных, организация вычислительного процесса	12
2. Задачи расчетного характера	15
3. Оперативное управление производством, расчеты по ценообразованию	7
4. Техническая подготовка производства, транспортное, ремонтное, энергетическое и инструментальное обслуживание производства	8
5. Бухгалтерский учет, финансовые расчеты, учет пенсий и пособий, учет страховых операций, качество продукции	13
6. Управление кадрами	13
7. Техничко-экономическое планирование	13
8. Материально-техническое снабжение, реализация и сбыт готовой продукции	13