

Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Е. А. Блинова, Л. С. Мороз**

# **АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ И ПРИЛОЖЕНИЙ**

**Учебно-методическое пособие к выполнению  
курсового проекта для студентов специальности  
1-40 05 01-03 «Информационные системы и технологии  
(издательско-полиграфический комплекс)»**

Минск 2018

УДК 004.658(075.8)  
ББК 32.973я73  
Б69

Рассмотрено и рекомендовано к изданию редакционно-издательским советом Белорусского государственного технологического университета.

**Р е ц е н з е н т ы :**

кандидат технических наук, доцент кафедры информационных технологий автоматизированных систем

Белорусского государственного университета информатики и радиоэлектроники *О. В. Герман*;

кандидат технических наук, доцент, доцент кафедры информатики и веб-дизайна Белорусского государственного технологического университета *А. А. Дятко*

**Блинова, Е. А.**

Б69      Администрирование баз данных и приложений : учеб.-метод. пособие к выполнению курсового проекта для студентов специальности 1-40 05 01-03 «Информационные системы и технологии (издательско-полиграфический комплекс)» / Е. А. Блинова, Л. С. Мороз. – Минск : БГТУ, 2018. – 70 с.

Учебно-методическое пособие предназначено для студентов, выполняющих курсовой проект по дисциплине «Администрирование баз данных и приложений», и содержит план работы, требования к курсовому проекту, описание структуры пояснительной записки и демонстрационный пример. Результатом курсового проекта являются разработанные студентом база данных в СУБД Oracle с применением какой-либо технологии и приложения для демонстрации возможностей разработанной базы данных.

УДК 004.658(075.8)  
ББК 32.973я73

© УО «Белорусский государственный  
технологический университет», 2018  
© Блинова Е. А., Мороз Л. С., 2018

## Оглавление

Введение .....	6
1 Требования к курсовому проекту .....	7
1.1 Этапы и сроки выполнения курсового проекта .....	7
1.2 Минимальные требования к курсовому проекту .....	8
1.3 Дополнительные требования к курсовому проекту .....	8
2 Основные разделы пояснительной записки .....	9
2.1 Структура пояснительной записки .....	9
2.2 Титульный лист .....	9
2.3 Задание на курсовой проект .....	9
2.4 Содержание пояснительной записки .....	10
2.5 Введение .....	10
2.6 Основная часть пояснительной записки .....	10
2.6.1 Постановка задачи .....	10
2.6.2 Установка и настройка сервера СУБД .....	11
2.6.2.1 Установка и настройка сервера СУБД Oracle 12c .....	11
2.6.2.2 Установка и настройка мультиарендной архитектуры сервера СУБД Oracle 12c .....	20
2.6.2.3 Файлы, обслуживающие экземпляр СУБД Oracle 12c .....	23
2.6.2.4 Файлы данных экземпляра сервера СУБД Oracle 12c .....	26
2.6.3 Проектирование инфраструктуры базы данных и порядок авторизации пользователей к базе данных .....	29
2.6.3.1 Проектирование табличных пространств .....	29
2.6.3.2 Создание ролей .....	31
2.6.3.3 Назначение системных привилегий .....	32
2.6.3.4 Создание профилей безопасности .....	33
2.6.3.5 Создание пользователей и назначение ролей .....	35
2.6.3.6 Выдача объектных привилегий пользователям .....	36
2.6.4 Проектирование и реализация объектов базы данных .....	38
2.6.4.1 Проектирование и реализация таблиц и ограничений целостности .....	38
2.6.4.2 Проектирование и реализация последовательностей .....	40

2.6.4.3 Проектирование и реализация представлений .....	40
2.6.4.4 Проектирование и реализация индексов .....	40
2.6.4.5 Проектирование и реализация пакетов, группирующих хранимые процедуры и функции .....	42
2.6.4.6 Проектирование и реализация триггеров .....	43
2.6.5 Изучение и применение технологии в созданной базе данных .....	44
2.6.6 Импорт и генерация тестовых данных .....	44
2.6.7 Резервное копирование и восстановление данных .....	45
2.6.8 Разработка демонстрационного приложения .....	45
2.7 Заключение .....	46
2.8 Графический материал .....	46
2.9 Список использованных источников .....	46
2.10 Приложения .....	46
3 Оформление пояснительной записки .....	47
3.1 Общие требования .....	47
3.2 Структурные элементы записки .....	47
3.3 Нумерация страниц .....	48
3.4 Перечисления .....	48
3.5 Изложение текста .....	49
3.6 Формулы .....	49
3.7 Примечания .....	50
3.8 Рисунки .....	50
3.9 Таблицы .....	50
3.10 Ссылки .....	51
3.11 Приложения .....	52
3.12 Список использованных источников .....	52
4 Примерная тематика курсовых проектов .....	53
4.1 Общие замечания .....	53
4.2 Список областей применения баз данных .....	53
4.3 Примерный список рекомендуемых технологий для самостоятельного изучения .....	54
5 Демонстрационный пример .....	55
5.1 Постановка задачи .....	55
5.2 Проектирование и разработка базы данных .....	55
5.3 Проектирование и разработка хранимых процедур .....	56
5.4 Проектирование и разработка триггеров .....	57
5.5 Способы резервного копирования в Oracle .....	57

5.5.1 Общие сведения.....	57
5.5.2 Горячее резервирование с использованием утилиты RMAN.....	58
5.5.3 Утилита Handy Backup.....	62
5.5.4 Утилита Iperius Backup .....	62
5.5.5 Приложение для горячего резервирования SMA Backup .....	63
Приложение А (обязательное). Образец оформления титульного листа пояснительной записки .....	65
Приложение Б (обязательное). Образец оформления задания на курсовой проект.....	66
Литература .....	69

## Введение

Целью курсового проекта является освоение навыков проектирования и администрирования базы данных (БД). В процессе выполнения работы студент должен применить теоретические знания, полученные при изучении дисциплины «Администрирование баз данных и приложений», спроектировав базу данных, самостоятельно освоить новые технологии, применяемые в СУБД, и создать приложение для демонстрации возможностей разработанной базы данных.

Задание предполагает:

- 1) установку и настройку сервера СУБД Oracle 12c;
- 2) проектирование инфраструктуры базы данных;
- 3) создание необходимых объектов;
- 4) загрузку тестовых данных для демонстрации и проведения оптимизации;
- 5) самостоятельное изучение и применение определенной технологии в созданной базе данных;
- 6) самостоятельное изучение и применение резервного копирования и восстановления данных в созданной базе данных;
- 7) разработку небольшого приложения для демонстрации.

Для успешной защиты курсового проекта студент должен:

- установить и настроить сервер СУБД;
- спроектировать инфраструктуру базы данных;
- создать необходимые объекты;
- импортировать или сгенерировать тестовые данные для демонстрации;
- исследовать и при необходимости оптимизировать структуру запросов к базе данных;
- самостоятельно изучить и применить определенную технологию в созданной базе данных;
- выполнить резервное копирование и восстановление данных;
- разработать небольшое приложение для демонстрации;
- подготовить пояснительную записку к курсовому проекту.

Предлагаемое учебно-методическое пособие содержит теоретический материал, требования к курсовому проекту (раздел 1), структуру и описание содержимого пояснительной записки (раздел 2), правила ее оформления (раздел 3), примерный список областей применения баз данных и технологий (раздел 4) и демонстрационный пример (раздел 5).

## 1 Требования к курсовому проекту

### 1.1 Этапы и сроки выполнения курсового проекта

Тематика курсовых проектов представлена отдельным списком (см. раздел 4). Курсовой проект разрабатывается студентами в сроки, предусмотренные графиком учебного процесса. На выполнение курсового проекта в учебном плане предусмотрено 40 часов. Курсовой проект должен быть выполнен в течение 11 учебных недель со следующим распределением объемов выполнения задания (в процентах, с нарастающим итогом).

Неделя	1	2	3	4	5	6	7	8	9	10	11
%	5	15	25	35	45	55	65	75	85	95	100

Предполагается, что в качестве СУБД выбирается Oracle 12с, однако по согласованию с преподавателем СУБД может быть выбрана и другая. Необходимо также определить область решения задачи, совместно с преподавателем сформулировать решаемые бизнес-задачи и выбрать технологию для самостоятельного изучения. В задании должны быть перечислены бизнес-задачи, для решения которых создается база данных. После согласования задания должен быть составлен подписанный студентом и преподавателем лист задания (в двух экземплярах) для подписи и утверждения у заведующего кафедрой. Один экземпляр остается на кафедре у преподавателя для контроля выполнения курсового проекта, второй служит студенту техническим заданием.

Для консультаций по выполнению проекта установлено специальное время консультаций. В процессе консультаций у преподавателя могут возникнуть замечания, которые следует устранить до защиты работы. Во время проведения текущих аттестаций отмечается объем выполненного задания.

Законченный курсовой проект должен быть представлен для защиты на кафедру за месяц до начала сессии. При защите курсового проекта предоставляются пояснительная записка к курсовому проекту и электронный носитель CD/DVD, содержащий пояснительную записку, SQL-скрипты, файлы базы данных и приложения. За месяц до сессии на кафедре вывешивается список дат защиты курсовых проектов. Для защиты курсового проекта необходимо записаться на удобную дату.

Курсовые проекты, претендующие на высокую оценку, направляются на открытую защиту, где могут присутствовать все желающие: студенты и преподаватели.

## **1.2 Минимальные требования к курсовому проекту**

Для получения положительной оценки за курсовой проект необходимо выполнить следующее:

- 1) выбрать область решения задачи и технологию;
- 2) установить и настроить сервер СУБД Oracle 12c;
- 3) спроектировать инфраструктуру базы данных;
- 4) разработать порядок авторизации в базе данных;
- 5) создать объекты базы данных;
- 6) импортировать и/или сгенерировать тестовые данные для демонстрации;
- 7) самостоятельно изучить и применить определенную технологию в созданной базе данных;
- 8) выполнить резервное копирование и восстановление данных;
- 9) подготовить пояснительную записку к курсовому проекту.

Требования к технологии:

- самостоятельно изучить и применить выбранную технологию в созданной базе данных;
- описать в пояснительной записке особенности применения технологии.

Требования по оптимизации:

- 1) исследовать и при необходимости оптимизировать структуру запросов к базе данных;
- 2) создать необходимые объекты для ускорения поиска в базе данных.

## **1.3 Дополнительные требования к курсовому проекту**

Для повышения оценки за курсовой проект необходимо:

- выбрать сложную бизнес-задачу (по согласованию с преподавателем) и решить ее наиболее рациональным способом;
- создать план резервного копирования и восстановления данных;
- разработать полноценное приложение для демонстрации работы;
- подготовить пояснительную записку к курсовому проекту на высоком уровне.



## **2 Основные разделы пояснительной записки**

Общий объем пояснительной записки курсового проекта должен составлять примерно 50–60 страниц текста, включая приложения.

### **2.1 Структура пояснительной записки**

Пояснительная записка состоит:

- 1) из титульного листа;
- 2) задания на курсовой проект;
- 3) реферата;
- 4) содержания;
- 5) введения;
- 6) основной части пояснительной записки;
- 7) заключения;
- 8) списка использованных источников;
- 9) графического материала;
- 10) приложений.

### **2.2 Титульный лист**

Титульный лист является первой страницей пояснительной записки. Исчисление страниц пояснительной записки начинается с титульного листа, номер страницы на котором не ставится. Пример оформления приведен в приложении А.

### **2.3 Задание на курсовой проект**

Задание на курсовой проект формулируется преподавателем при участии студента и включает:

- тему проекта;
- требования к работе;
- исходные данные;
- срок сдачи законченного проекта;
- содержание пояснительной записки;
- перечень графического материала;
- календарный план выполнения работы.

Пример задания на курсовое проектирование приведен в приложении Б.

## **2.4 Содержание пояснительной записки**

Содержание должно включать в себя: введение, наименование всех разделов и подразделов, пунктов (если последние имеют наименование), заключение, список использованных источников и приложений с указанием номеров страниц, на которых они размещены.

## **2.5 Введение**

Введение – это небольшой обзор курсового проекта. Во введении следует указать цель выполнения курсового проекта, сформулировать задачи для достижения цели, кратко описать содержание пояснительной записки со ссылкой на ее разделы.

## **2.6 Основная часть пояснительной записки**

Основная часть пояснительной записки должна включать следующие разделы и подразделы.

### **2.6.1 Постановка задачи**

В настоящем разделе необходимо обосновать выбор области решения бизнес-задачи и применяемой технологии.

Примерный список областей применения баз данных приведен в разделе 4.

При формулировании задания должно быть подробно указано, какие именно бизнес-задачи могут быть решены при использовании базы данных. Например, в случае автоматизации работы магазина необходимо указать в задании:

- 1) процедуры поставки товара на склад магазина;
- 2) процедуры возврата товара;
- 3) процедуры оплаты товара, в том числе со скидкой;
- 4) процедуры получения остатков товара на складе;
- 5) анализ свойств покупателей: частота покупок, наличие дорогостоящих покупок и т. д.;
- 6) управление работой продавцов (смены, ответственность за определенные участки и др.).

Список решаемых бизнес-задач должен быть согласован с преподавателем. Чем больше бизнес-задач могут быть решены при помощи базы данных, тем лучше, и, соответственно, выше оценка. Если какая-то

из указанных бизнес-задач не была решена, работа может быть признана невыполненной. Если какая-то из указанных бизнес-задач была решена не полностью, работа может быть признана выполненной не полностью, и пропорционально снижена оценка.

Предполагается, что будет использоваться СУБД Oracle 12c. Однако по согласованию с преподавателем можно использовать и другую реляционную СУБД, а в случае сравнения применения реляционного и нереляционного решения будут использованы оба типа СУБД.

При выборе технологии необходимо учитывать, целесообразно ли применение данной технологии при решении именно такой задачи. Например, для задачи, предполагающей создание общей базы данных для распределенного предприятия, может быть использована технология репликации или технология работы с пространственными данными.

Примерный список технологий приведен в разделе 4.

В пояснительной записке следует обосновать выбор технологии, применяемой для решения выбранной бизнес-задачи. Необходимо также описать существующие аналогичные решения. Достаточно трех решений при их наличии. Ссылки на аналогичные решения следует указать в списке использованных источников.

## **2.6.2 Установка и настройка сервера СУБД**

Необходимо установить и настроить сервер СУБД Oracle 12c и среду выполнения приложения на университетском сервере (за деталями следует обращаться к преподавателю), в облачной инфраструктуре или на личном ноутбуке (по выбору). В пояснительной записке требуется указать, какой сервер (облачный, виртуальный) используется.

В случае если выбрана не СУБД Oracle 12c, то по согласованию с преподавателем необходимо указать краткие этапы установки и настройки сервера СУБД.

### **2.6.2.1 Установка и настройка сервера СУБД Oracle 12c**

Далее указаны этапы установки сервера Oracle 12c на операционную систему Windows Server 2012.

Дистрибутив сервера Oracle 12c можно найти на сервере Oracle (рисунок 2.1).

Процесс установки начинается с установок обновлений (рисунок 2.2).

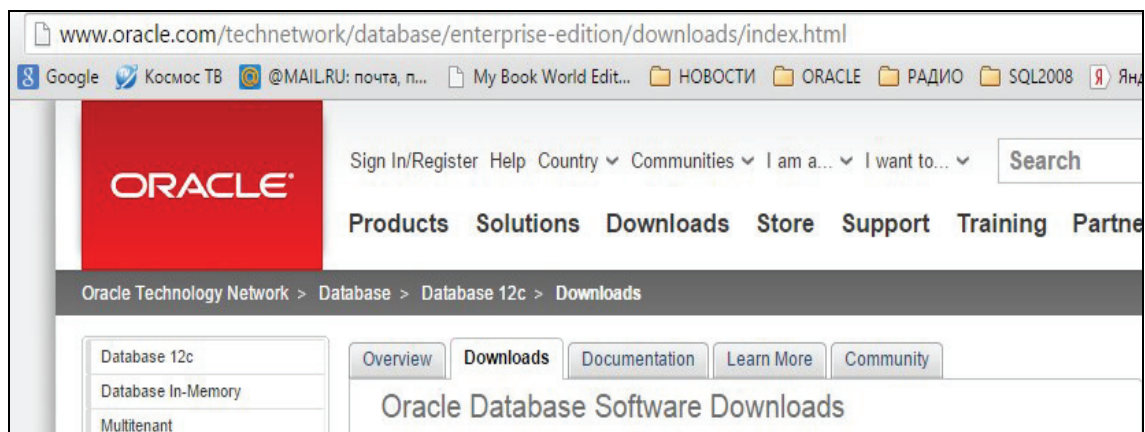


Рисунок 2.1 – Дистрибутивы Oracle

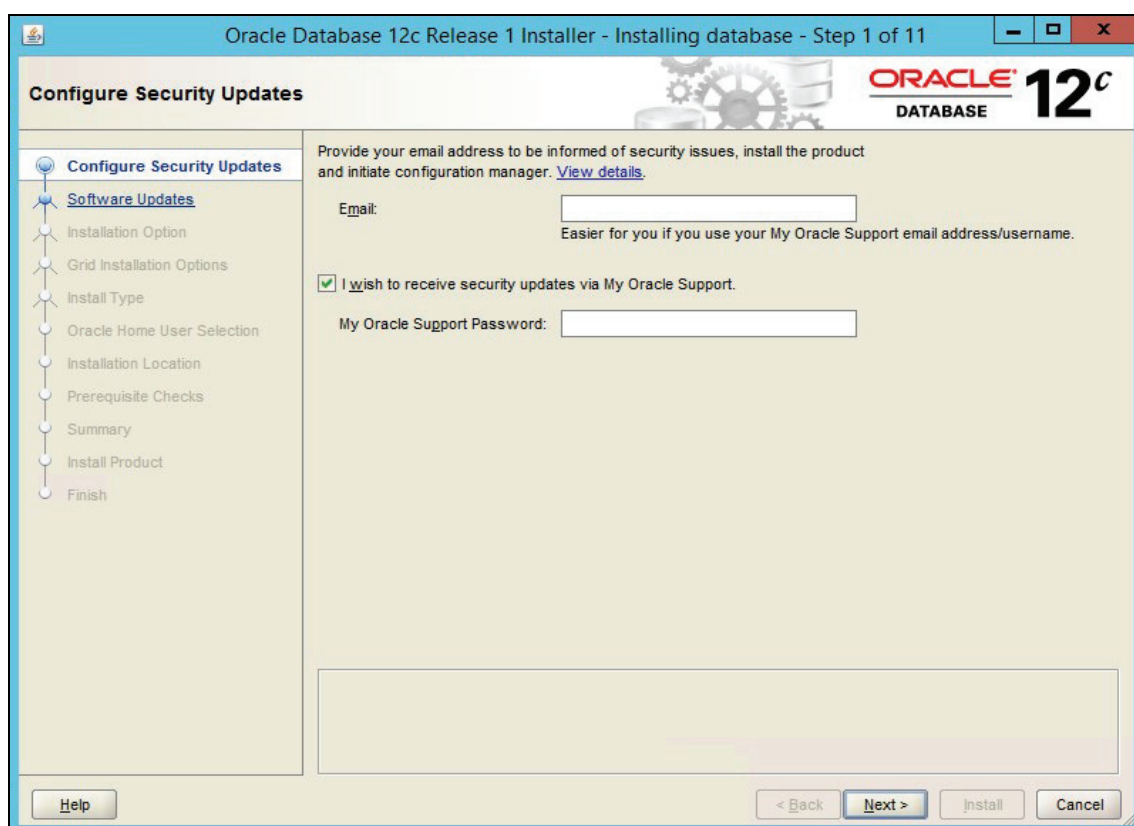


Рисунок 2.2 – Установка обновлений Oracle

Далее собираются необходимые сведения для установки. Первый шаг – следует выбрать, производится ли только установка программного обеспечения или еще и конфигурация базы данных (рисунок 2.3).

Затем необходимо указать конфигурацию Server Class сервера Oracle (рисунок 2.4).

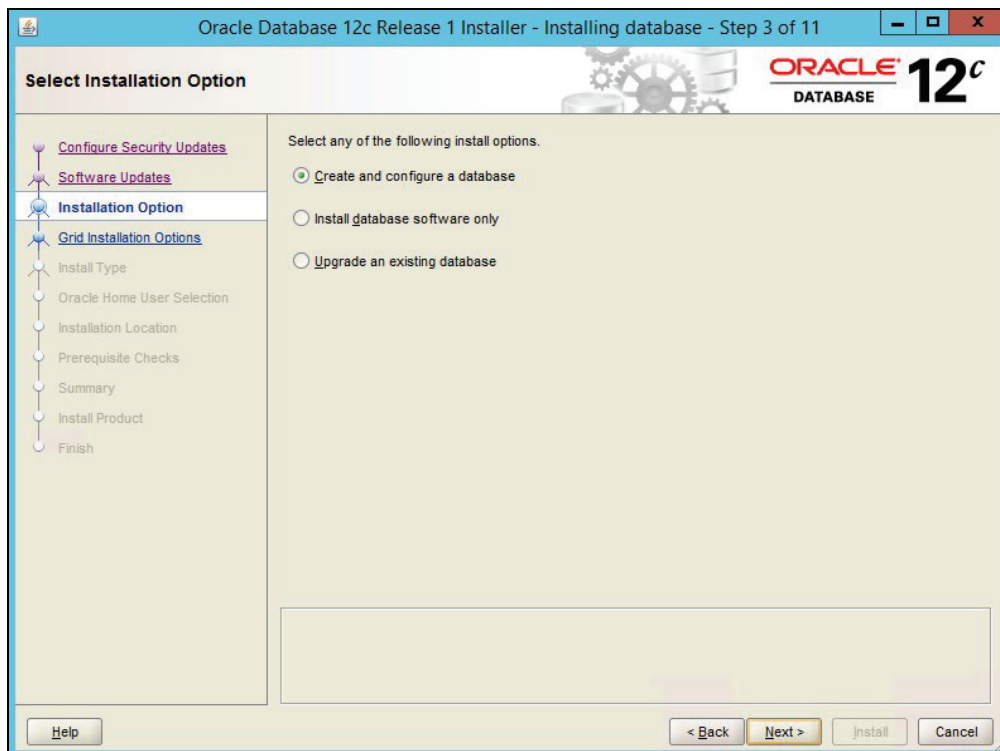


Рисунок 2.3 – Создание базы данных Oracle

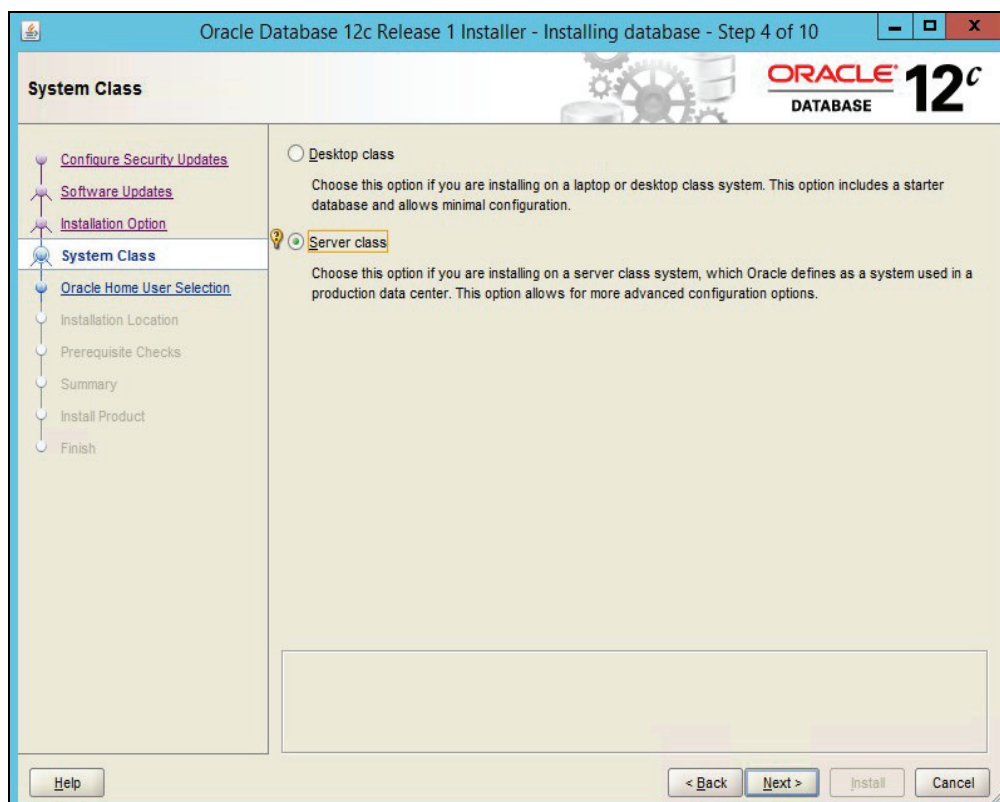


Рисунок 2.4 – Выбор конфигурации сервера Oracle

Следует определить, будет ли устанавливаемый сервер отдельным сервером или частью кластера (набора узлов, между которыми распределяется нагрузка) (рисунок 2.5). Необходимо указать, какой будет выбран режим установки: более или менее подробный (рисунок 2.6).

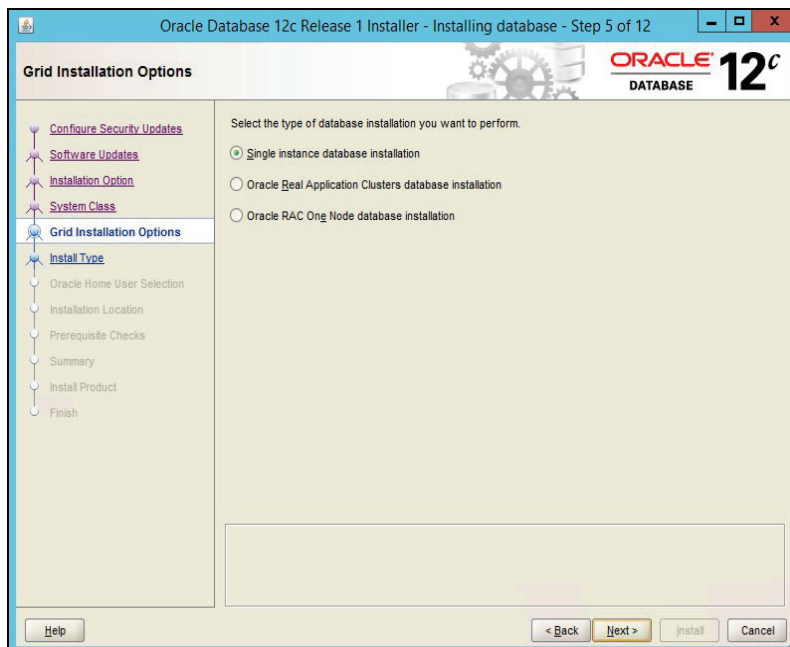


Рисунок 2.5 – Выбор кластерной конфигурации сервера Oracle

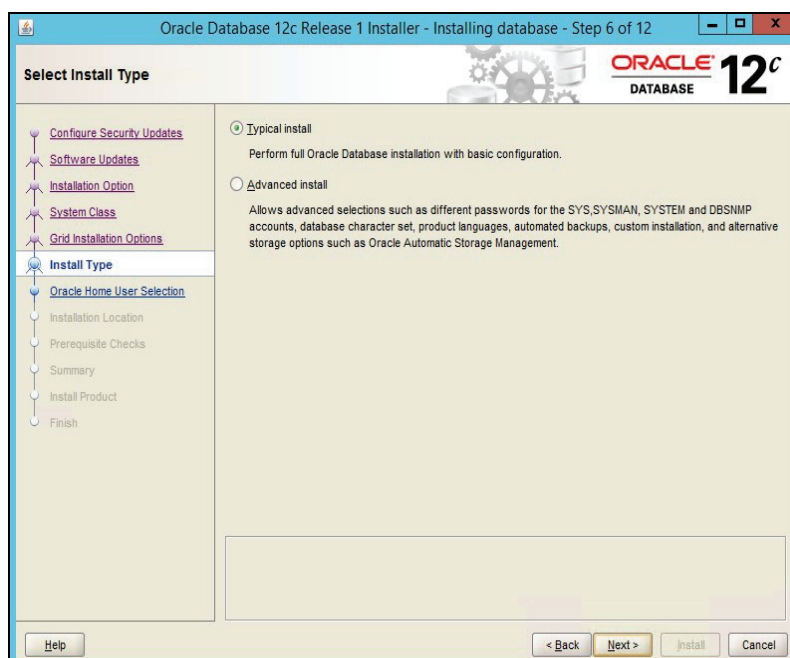


Рисунок 2.6 – Выбор режима установки сервера Oracle

Следует ввести для ОС Windows имя и пароль пользователя, под которым будет запускаться сервер Oracle (рисунок 2.7). Необходимо задать основные параметры сервера: место хранения данных, глобальное имя базы данных, тип установки сервера (рисунок 2.8).

Рисунок 2.7 – Создание пользователя для запуска сервера Oracle

Рисунок 2.8 – Основные параметры сервера Oracle



После сбора всех сведений о параметрах сервера выводится итоговое окно установки (рисунок 2.9).

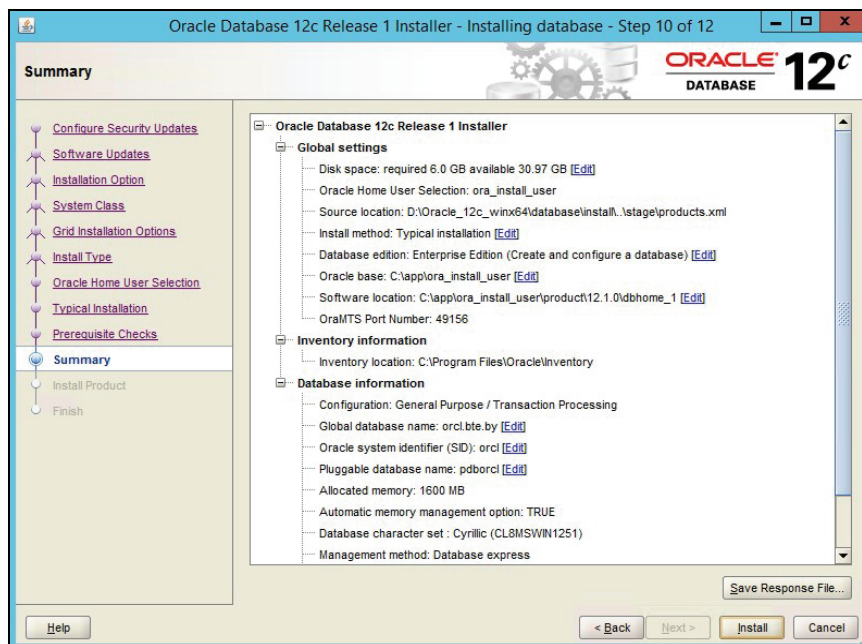


Рисунок 2.9 – Итоговое окно мастера установки сервера Oracle

В процессе установки требуется установить пароли для администраторов. При появлении следующего окна необходимо нажать кнопку Password Management (рисунок 2.10) и установить пароли для администраторов (рисунок 2.11).

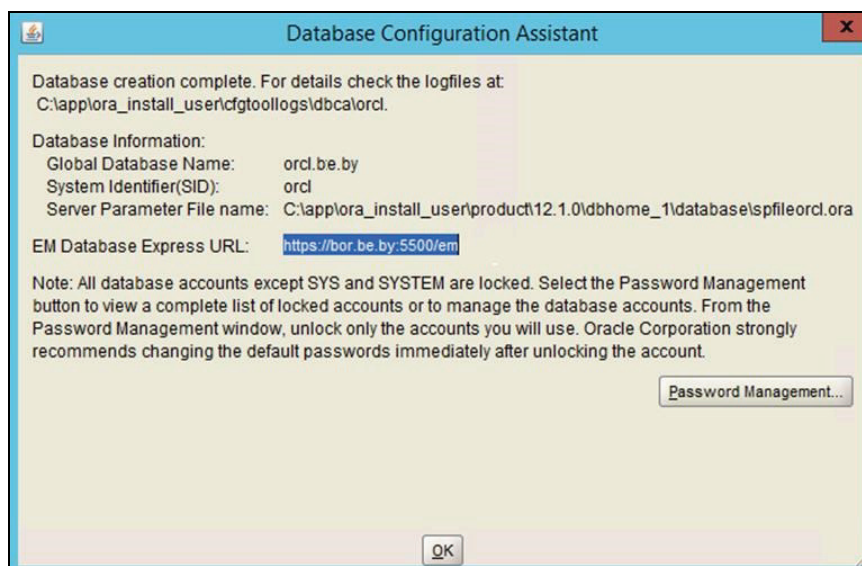


Рисунок 2.10 – Окно конфигурации сервера Oracle



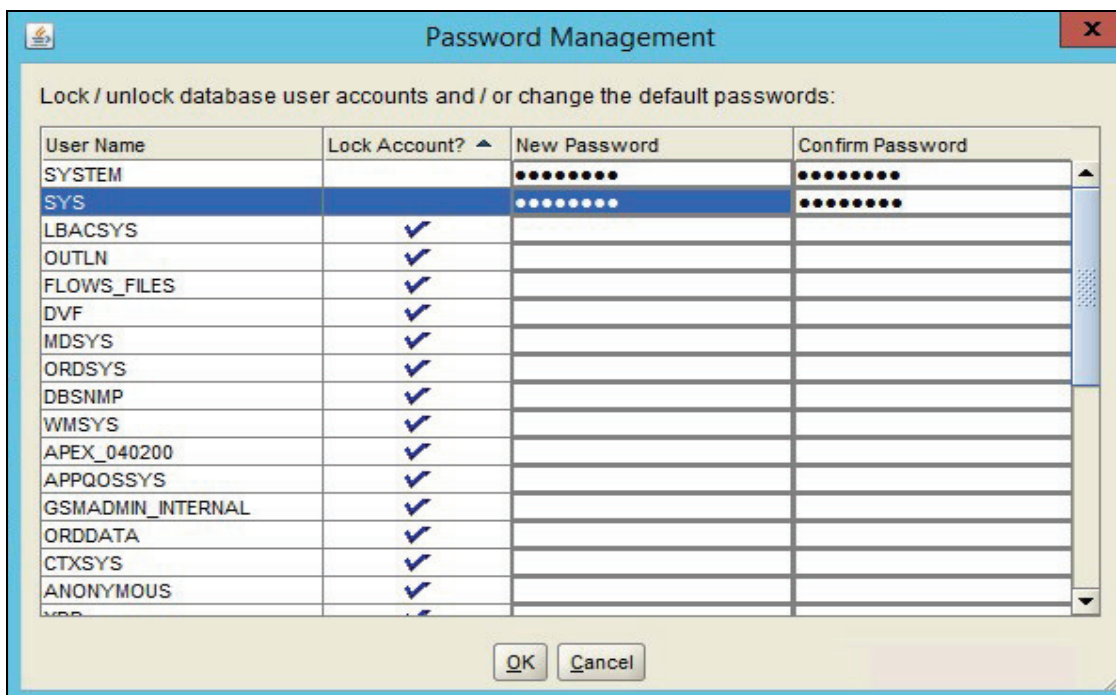


Рисунок 2.11 – Установка паролей администраторов сервера Oracle

После окончания установки следует проверить работу установленного сервера. Проверка осуществляется по следующему плану:

- проверить запущенные сервисы;
- проверить реестр;
- проверить пользователи и группы;
- установить соединение с базой данных.

Проверить запущенные сервисы можно в утилите Службы, сервисы Oracle должны быть запущены (рисунок 2.12).

Optimize drives	Helps the c...	Manual	Local Sy
OracleJobSchedulerORCL		Disabled	.\ora_in
OracleOraDB12Home1MTSRecoveryService	Running	Automatic	.\ora_in
OracleOraDB12Home1TNSListener	Running	Automatic	.\ora_in
OracleRemExecServiceV2	Running	Manual	Local Sy
OracleServiceORCL	Running	Automatic	.\ora_in
OracleVssWriterORCL	Running	Automatic	.\ora_in
Performance Counter DLL Host	Enables rem...	Manual	Local Se

Рисунок 2.12 – Проверка запущенных сервисов сервера Oracle

В реестре можно проверить, куда установлен сервер Oracle и его основные параметры (рисунок 2.13).

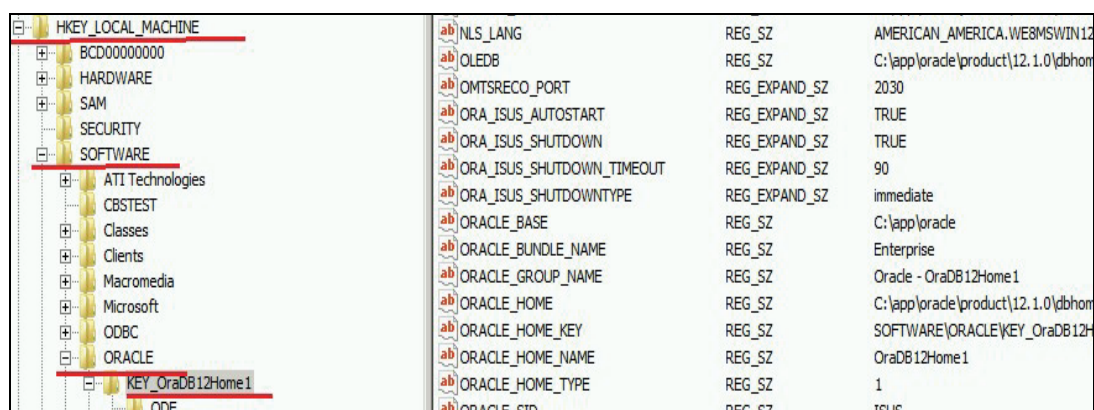


Рисунок 2.13 – Параметры сервера Oracle в редакторе реестра

В настройках локальных пользователей и групп следует убедиться, что созданы все необходимые группы сервера Oracle. Для выполнения административных работ под текущим пользователем Windows необходимо добавить его в группу ora\_dba (рисунок 2.14).

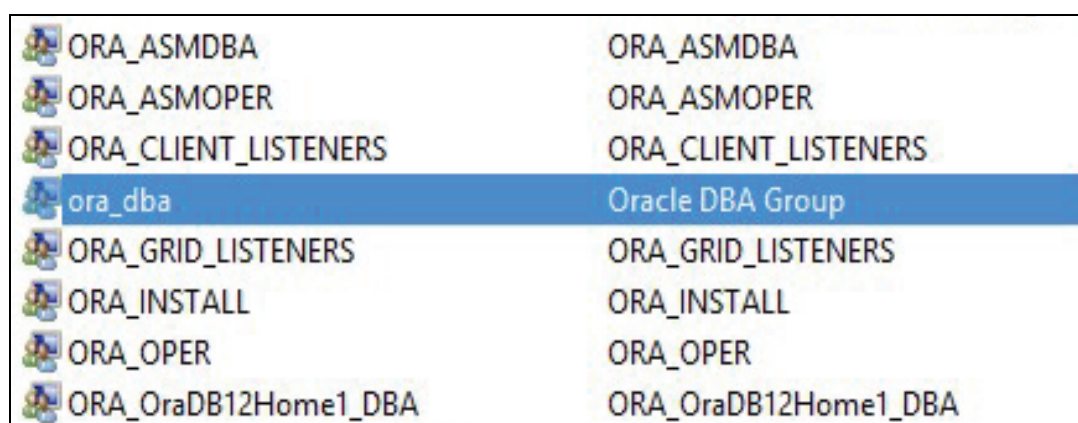


Рисунок 2.14 – Пользователи сервера Oracle

В заключение следует установить соединение с сервером из клиентского приложения. В качестве клиентских приложений будем использовать Oracle SQL Developer – интегрированную среду разработки на языках SQL и PL/SQL и администрирования баз данных и утилиту командной строки SQLPlus. Продемонстрируем соединение из SQL Developer и SQLPlus.

В SQL Developer необходимо указать имя соединения, имя пользователя, пароль, тип соединения, имя компьютера, на котором установлен сервер Oracle, или его IP-адрес и нажать кнопку Test. При успешном соединении с сервером статус соединения изменится на Success (рисунок 2.15).

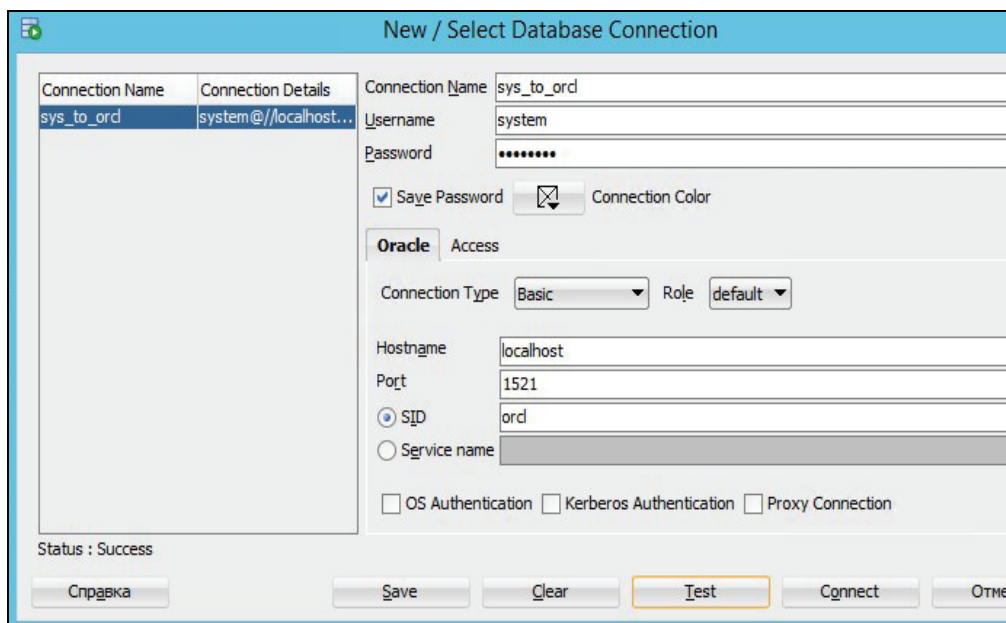


Рисунок 2.15 – Установка соединения с сервером Oracle

Далее следует установить соединение с сервером из утилиты командной строки SQLPlus. Можно установить соединение без указания имени пользователя и пароля, тогда соединение будет установлено от имени текущего пользователя Windows (рисунок 2.16).

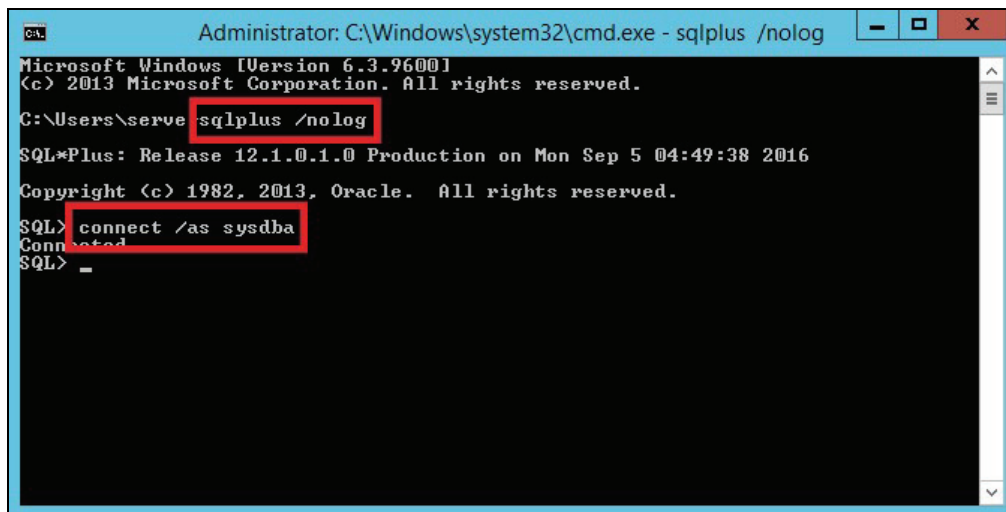


Рисунок 2.16 – Установка соединения с сервером Oracle без указания имени пользователя и пароля

Также можно установить соединение с указанием имени пользователя и пароля, тогда соединение будет установлено от имени данного пользователя (рисунок 2.17).

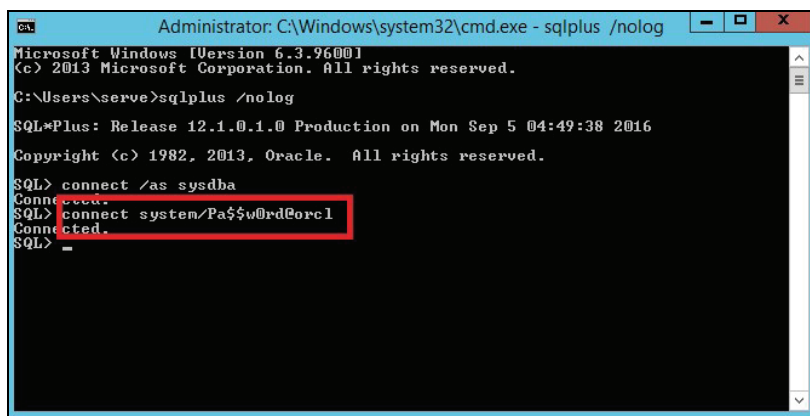


Рисунок 2.17 – Установка соединения с сервером Oracle с указанием имени пользователя и пароля

### 2.6.2.2 Установка и настройка мультиарендной архитектуры сервера СУБД Oracle 12c

Oracle Multitenant – технология, позволяющая запустить несколько независимых подключаемых баз данных в рамках одного экземпляра. Каждая подключаемая база данных имеет свой набор объектов, но при этом у них общая SGA и один набор серверных процессов. Базы данных изолированы друг от друга и не конфликтуют между собой. Для выполнения курсовой работы необходимо создать подключаемую базу данных. Продemonстрируем создание подключаемой базы данных. Для этого необходимо запустить утилиту Database Configuration Assistant (рисунок 2.18).

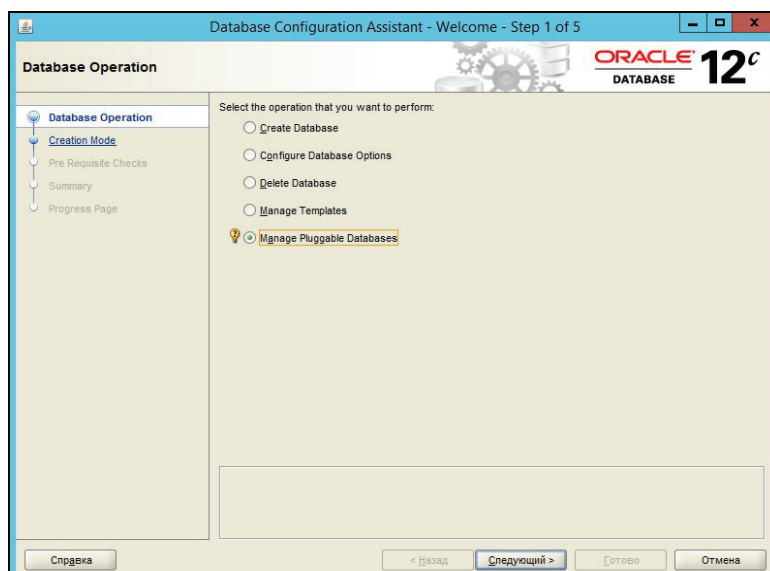


Рисунок 2.18 – Запуск утилиты Database Configuration Assistant

После чего следует указать, что будет создаваться подключаемая база данных (рисунок 2.19). Затем необходимо выбрать экземпляр, на котором будут создаваться подключаемые базы данных (рисунок 2.20).

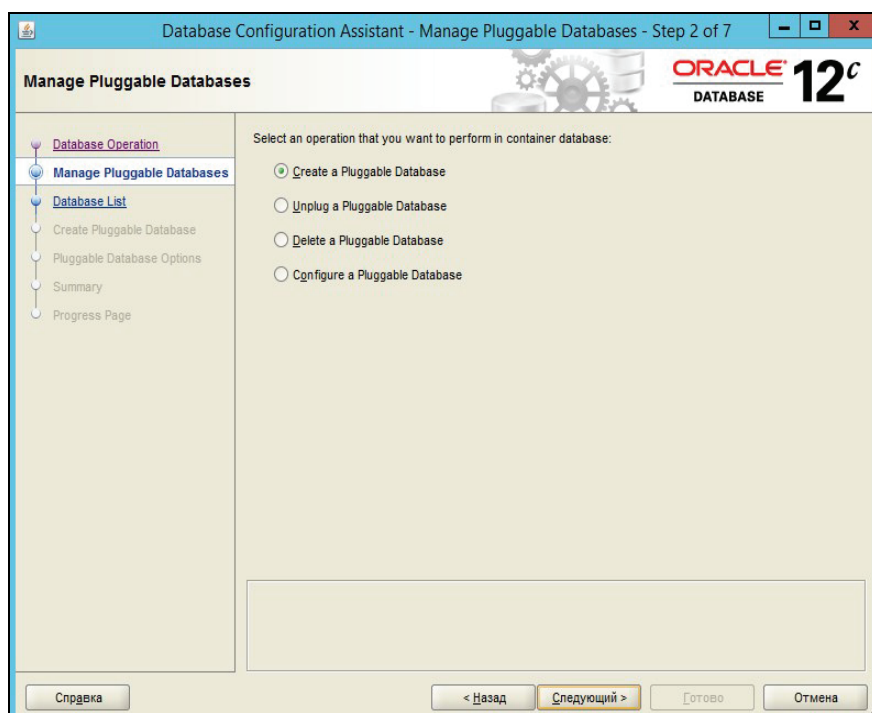


Рисунок 2.19 – Создание подключаемой базы данных

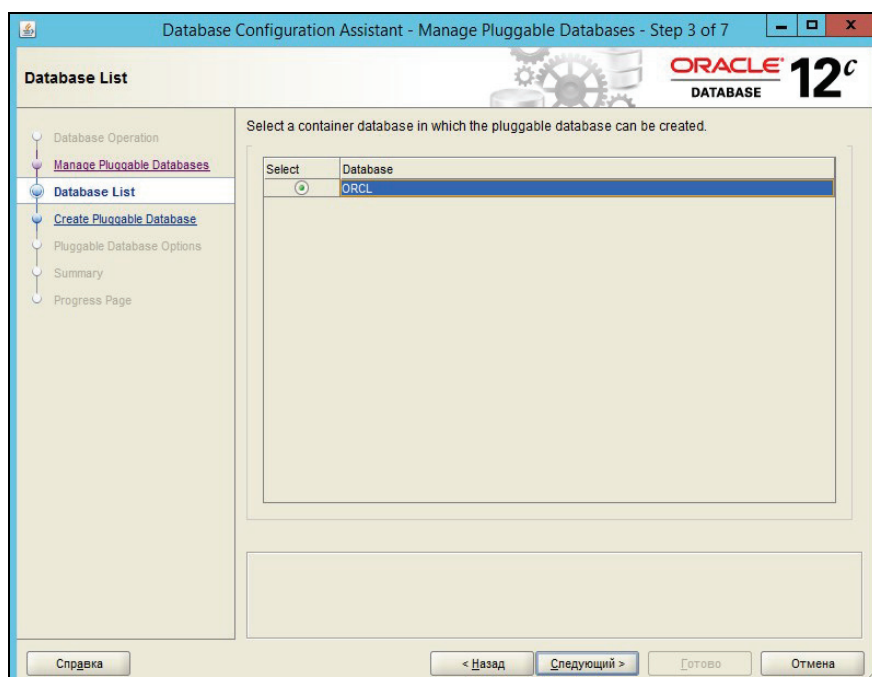


Рисунок 2.20 – Выбор экземпляра



Следует указать способ создания подключаемой базы данных – из архива или из набора файлов (рисунок 2.21).

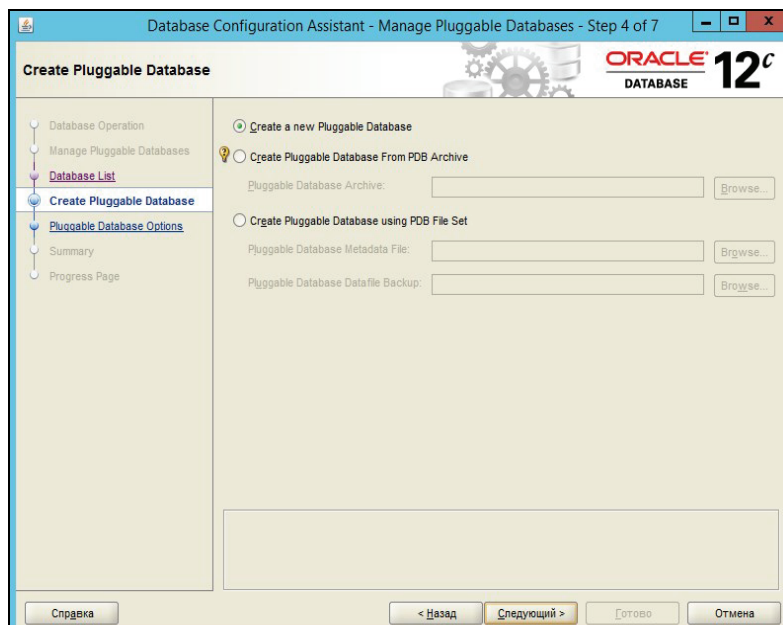


Рисунок 2.21 – Выбор источника для подключаемой базы данных

Необходимо выбрать месторасположение файлов базы данных и наделить пользователя административными полномочиями (рисунок 2.22).

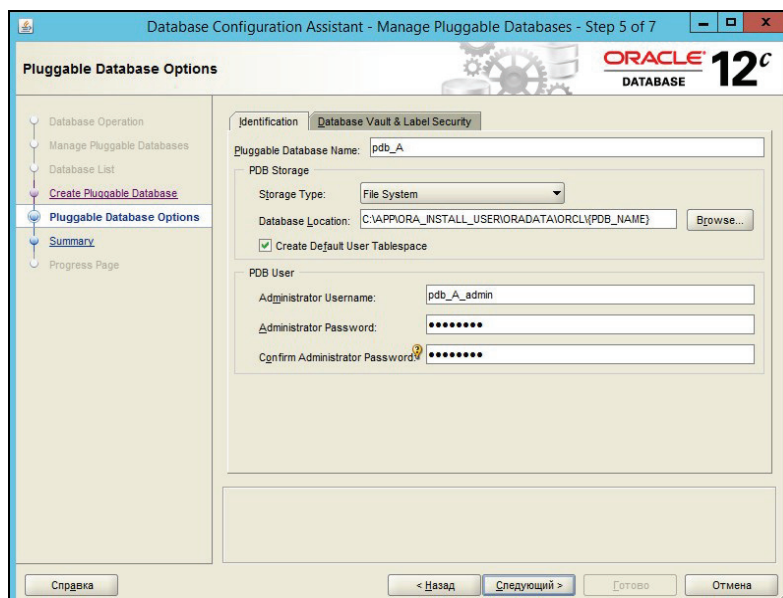


Рисунок 2.22 – Выбор свойств базы данных и администратора

В результате выполнения мастера база данных создана (рисунок 2.23).

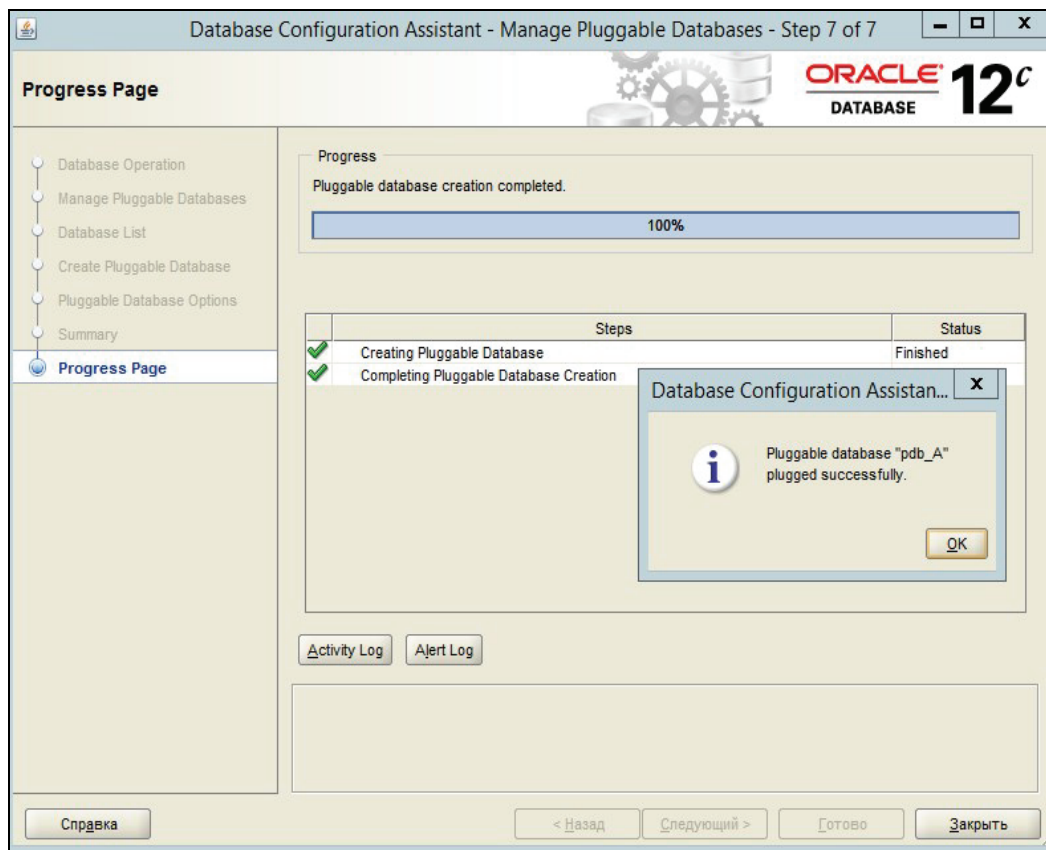


Рисунок 2.23 – Итоговое окно создания базы данных

### 2.6.2.3 Файлы, обслуживающие экземпляр СУБД Oracle 12c

Чтобы запустить сервер, используется набор служебных файлов Oracle:

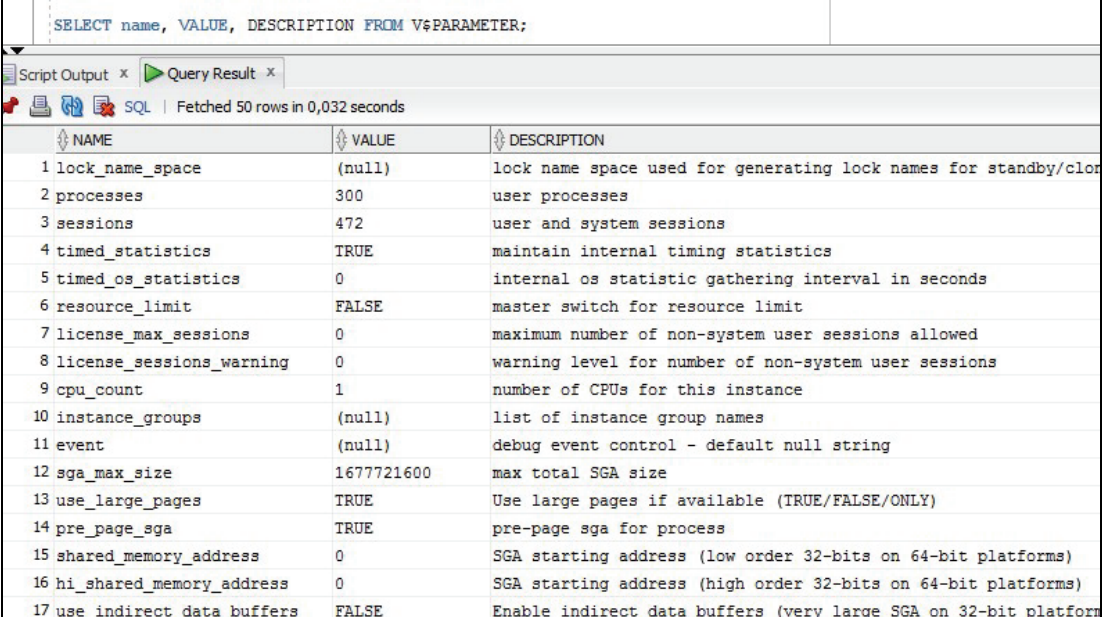
- 1) файл параметров;
- 2) управляющие файлы;
- 3) файл паролей;
- 4) файлы журналов работы.

Файлы параметров предназначены для конфигурирования действий Oracle при старте.

Файл параметров сервера (SPFILE) – двоичный файл, который может быть записан и считан сервером базы данных и не должен редактироваться вручную. Он находится на сервере, на котором выполняется экземпляр Oracle. Этот файл не изменяется при завершении работы экземпляра и его запуске. Именем по умолчанию для этого

файла является `spfile.ora`. Текстовый файл параметров инициализации может быть прочитан сервером базы данных, но сервер не может записывать в него. Именем по умолчанию для этого файла является `init.ora`. Изменения в файле параметров могут быть произведены командой `ALTER SYSTEM ... SCOPE=SPFILE`, тогда изменение параметров сохраняется в `SPFILE` и будет применяться при следующем старте Oracle.

Просмотреть значения параметров можно при помощи следующего динамического представления словаря (рисунок 2.24).



The screenshot shows a SQL\*Plus window with the query `SELECT name, VALUE, DESCRIPTION FROM V$PARAMETER;` executed. The output is a table with 17 rows, each representing a database parameter. The columns are NAME, VALUE, and DESCRIPTION. The parameters listed include lock\_name\_space, processes, sessions, timed\_statistics, timed\_os\_statistics, resource\_limit, license\_max\_sessions, license\_sessions\_warning, cpu\_count, instance\_groups, event, sga\_max\_size, use\_large\_pages, pre\_page\_sga, shared\_memory\_address, hi\_shared\_memory\_address, and use\_indirect\_data\_buffers.

NAME	VALUE	DESCRIPTION
lock_name_space	(null)	lock name space used for generating lock names for standby/clon
processes	300	user processes
sessions	472	user and system sessions
timed_statistics	TRUE	maintain internal timing statistics
timed_os_statistics	0	internal os statistic gathering interval in seconds
resource_limit	FALSE	master switch for resource limit
license_max_sessions	0	maximum number of non-system user sessions allowed
license_sessions_warning	0	warning level for number of non-system user sessions
cpu_count	1	number of CPUs for this instance
instance_groups	(null)	list of instance group names
event	(null)	debug event control - default null string
sga_max_size	1677721600	max total SGA size
use_large_pages	TRUE	Use large pages if available (TRUE/FALSE/ONLY)
pre_page_sga	TRUE	pre-page sga for process
shared_memory_address	0	SGA starting address (low order 32-bits on 64-bit platforms)
hi_shared_memory_address	0	SGA starting address (high order 32-bits on 64-bit platforms)
use_indirect_data_buffers	FALSE	Enable indirect data buffers (very large SGA on 32-bit platform)

Рисунок 2.24 – Просмотр параметров экземпляра

Управляющие файлы – это файлы, содержащие имена и местоположение основных физических файлов базы данных и некоторых параметров. Управляющие файлы содержат данные о физической структуре базы данных. Эти файлы являются критическими по отношению к базе данных. Без них невозможно открыть файлы данных, чтобы получить доступ к данным в базе данных. Они могут также содержать метаданные, связанные с резервными копиями.

По умолчанию для надежности создается два управляющих файла, хотя сервер может стартовать и с одним. Обычно для надежности их размещают на разных дисковых носителях. Размещение управляющих файлов можно увидеть из следующего представления словаря (рисунок 2.25).



SELECT NAME FROM V\$CONTROLFILE;	
Script Output x	Query Result x
SQL   All Rows Fetched: 2 in 0,078 seconds	
NAME	
1	C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\CONTROL01.CTL
2	C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\CONTROL02.CTL

Рисунок 2.25 – Местонахождение управляющих файлов

Файл паролей используется для аутентификации администраторов в задачах создания базы данных или запуска и остановки сервера. Аутентификация всех остальных пользователей выполняется внутри самой базы данных, но, поскольку база данных может быть выключена или не смонтирована, для этих случаев требуется иная форма аутентификации администратора. В связи со слишком большими привилегиями, предоставляемыми посредством этого файла, он должен храниться в защищенном каталоге, который недоступен ни для кого, за исключением администраторов базы данных и администраторов операционной системы.

Список пользователей, которым предоставлен административный доступ, можно получить из представления (рисунок 2.26).

SELECT * FROM V\$PWFILE_USERS;							
USERNAME	SYSDBA	SYSOPER	SYSASM	SYSBACKUP	SYSDBG	SYSKM	CON_ID
1 SYS	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	0
2 SYSDBG	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	1
3 SYSBACKUP	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	1
4 SYSKM	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	1

Рисунок 2.26 – Список привилегированных пользователей базы данных

Файл предупреждений базы данных является хронологическим журналом сообщений об ошибках или отклонениях в работе сервера, таких как:

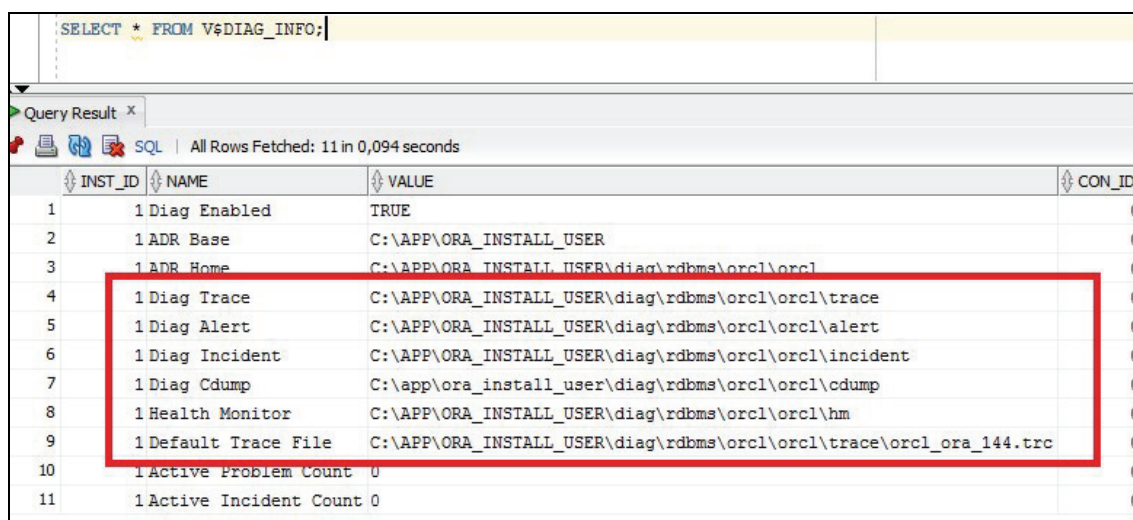
- использование при запуске каких-либо параметров инициализации не по умолчанию;
- все внутренние ошибки, которые возникли в процессе работы;
- все административные операции, такие как создание, изменение, удаление объектов и старт или остановка сервера;

– возникновение ошибок автоматического обновления материализованных представлений.

База данных Oracle использует журнал предупреждений, чтобы вести учет таких событий. Если административная операция успешна, сообщение отмечается в журнале предупреждений как «завершено» с меткой времени. Следует просматривать журнал, чтобы изучить не критические ошибки и информационные сообщения. Поскольку файл может вырасти до очень большого размера, необходимо периодически делать резервную копию файла предупреждений и удалять текущий файл.

Журнал предупреждений ведется в xml-формате и может быть просмотрен в браузере, текстовом редакторе или специализированном программном обеспечении для анализа файлов журналов.

Просмотреть месторасположение журналов предупреждений можно при помощи следующего представления словаря (рисунок 2.27).



INST_ID	NAME	VALUE	CON_ID
1	1 Diag Enabled	TRUE	0
2	1 ADR Base	C:\APP\ORA_INSTALL_USER	0
3	1 ADR Home	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl	0
4	1 Diag Trace	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl\trace	0
5	1 Diag Alert	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl>alert	0
6	1 Diag Incident	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl\incident	0
7	1 Diag Cdump	C:\app\ora_install_user\diag\rdbms\orcl\orcl\cdump	0
8	1 Health Monitor	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl\hm	0
9	1 Default Trace File	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl\trace\orcl_ora_144.trc	0
10	1 Active Problem Count	0	0
11	1 Active Incident Count	0	0

Рисунок 2.27 – Месторасположение файлов журналов предупреждений

#### 2.6.2.4 Файлы данных экземпляра сервера СУБД Oracle 12c

Все данные в Oracle сохраняются в файлах данных. Все таблицы, представления, индексы, триггеры, последовательности, программы на PL/SQL находятся в файлах данных. В каждой базе данных Oracle имеется по крайней мере один файл данных и файл журнала повторного выполнения. Каждый файл данных может быть связан только с одной базой данных. Файл данных состоит из заголовка и совокупности блоков. Заголовок файла данных содержит идентификатор базы данных, номер и имя файла, тип файла, SCN создания и состояния файла и пр.

Оперативные файлы журналов повтора предназначены для записи всех изменений, выполненных над данными. Используются для хранения на диске информации для повторного выполнения операций. Каждая база данных должна иметь не менее двух оперативных файлов журналов повтора. Текущий файл постепенно заполняется, и после его заполнения сервер переходит к записи в следующий файл. Поскольку файлы повтора необходимы для выполнения восстановления базы данных и являются критичными, их объединяют в группы. Запись происходит одновременно в файлы одной группы.

Просмотреть месторасположение журналов повторного выполнения можно при помощи следующего представления словаря (рисунок 2.28).

SELECT * FROM V\$LOGFILE;					
GROUP#	STATUS	TYPE	MEMBER	IS_RECOVERY_I	
1	3 (null)	ONLINE	C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO03.LOG	NO	
2	2 (null)	ONLINE	C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO02.LOG	NO	
3	1 (null)	ONLINE	C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO01.LOG	NO	

Рисунок 2.28 – Месторасположение файлов журналов повтора

Как только оперативный файл журнала повтора заполнен, экземпляр сервера Oracle начинает запись в следующий файл. Эта операция повторяется, и информация в оперативных файлах журнала многократно перезаписывается. Если необходимо сохранить историю изменений, нужно, чтобы их копия после переключения журналов сохранялась. Для этого достаточно перевести работу базы данных в режим работы ARCHIVELOG.

Состояние архивирования журналов повторного выполнения можно увидеть из следующего представления словаря (рисунок 2.29).

SELECT INSTANCE_NAME, ARCHIVER, ACTIVE_STATE FROM V\$INSTANCE;					
INSTANCE_NAME	ARCHIVER	ACTIVE_STATE			
1 orcl	STOPPED	NORMAL			

Рисунок 2.29 – Архивирование журналов в состоянии «Выключено»

Просмотреть режим архивирования журналов повторного выполнения можно при помощи следующего представления словаря (рисунок 2.30).

The screenshot shows the Oracle SQL Developer interface. The query editor contains the SQL statement: `SELECT NAME, LOG_MODE FROM V$DATABASE;|`. Below the editor, the 'Query Result' tab is active, displaying the results of the query. The status bar indicates 'All Rows Fetched: 1 in 0,031 seconds'.

	NAME	LOG_MODE
1	ORCL	ARCHIVELOG

Рисунок 2.30 – Архивирование журналов в состоянии «Включено»

Просмотреть, какие из журналов повторного выполнения были заархивированы, можно при помощи следующего представления словаря (рисунок 2.31).

The screenshot shows the Oracle SQL Developer interface. The query editor contains the SQL statement: `SELECT * FROM V$LOG;|`. Below the editor, the 'Query Result' tab is active, displaying the results of the query. The status bar indicates 'All Rows Fetched: 3 in 0,063 seconds'.

GROUP#	THREAD#	SEQUENCE#	BYTES	BLOCKSIZE	MEMBERS	ARCHIVED	STATUS	FIRST_CHANGE#	FIRST_TIME	NEXT
1	1	1	125 52428800	512	1	NO	CURRENT	3864455	27.09.16	28147
2	2	1	123 52428800	512	1	YES	INACTIVE	3864363	27.09.16	
3	3	1	124 52428800	512	1	YES	INACTIVE	3864438	27.09.16	

Рисунок 2.31 – Архивирование журналов повторного выполнения

Архивные файлы журналов повторного выполнения жизненно важны при восстановлении. Файлы журналов повтора должны применяться к базе данных последовательно. Если один из архивных файлов журналов повтора пропущен, то остальные архивные файлы журналов не могут быть использованы.

Необходимо хранить все архивные файлы журналов повторного выполнения с момента выполнения последней резервной копии. Резервные копии базы данных следует делать по установленному плану резервного копирования и восстановления.

### **2.6.3 Проектирование инфраструктуры базы данных и порядок авторизации пользователей к базе данных**

Физическая реализация базы данных может существенно повлиять на способность базы данных удовлетворять требованиям к хранению и производительности. Проектирование физической реализации базы данных включает планирование файловых групп, применение секционирования для управления большими таблицами и возможное использование сжатия.

Создание инфраструктуры базы данных состоит:

- 1) из проектирования табличных пространств;
- 2) создания ролей;
- 3) назначения системных привилегий;
- 4) создания профилей безопасности;
- 5) создания пользователей;
- 6) назначения пользователям ролей;
- 7) создания объектов базы данных;
- 8) назначения ролям объектных привилегий.

#### **2.6.3.1 Проектирование табличных пространств**

С одной стороны, табличное пространство – логическая структура хранения данных, а с другой – контейнер сегментов. С одним табличным пространством связаны один или несколько файлов операционной системы, с каждым файлом может быть связано только одно табличное пространство.

При установке экземпляра сервера Oracle создаются некоторые табличные пространства:

- SYSTEM;
- SYSAUX;
- USERS;
- UNDO;
- TEMP.

В табличном пространстве SYSTEM хранится словарь данных Oracle, т. е. набор таблиц, которые содержат метаданные о различных компонентах базы данных. Сервер не позволяет обращаться к таблицам словаря данных напрямую, а создает представления на базе этих таблиц, к которым могут обращаться пользователи.

Табличное пространство SYSAUX служит вспомогательным табличным пространством по отношению к табличному пространству SYSTEM. В нем содержатся такие сведения, как, например, расписание

работы планировщика, история работы оптимизатора, реализация пространственных данных, поиск в тексте и пр.

Табличное пространство USERS – табличное пространство для хранения пользовательских данных по умолчанию.

Табличное пространство UNDO служит для хранения данных отмены, которые используются для возврата измененных данных в исходное состояние.

Табличное пространство TEMP предназначено для хранения объектов, существующих на протяжении пользовательского сеанса.

Табличные пространства бывают трех видов:

1) PERMANENT TABLESPACE – табличное пространство для постоянных данных. Их может быть несколько для пользователя. Пользователь может создавать свои объекты в любом из таких табличных пространств, если у него есть соответствующие привилегии. Одно табличное пространство может быть указано в качестве пространства по умолчанию;

2) TEMPORARY TABLESPACE – табличное пространство для временных данных приписывается одному или нескольким пользователям, которые могут там размещать свои временные данные;

3) UNDO TABLESPACE – табличное пространство для данных отката. Таких пространств может быть несколько, но активным является всегда только одно (указывается в конфигурационном файле SPFILE.ORA).

Создаются постоянные табличные пространства при помощи команды CREATE TABLESPACE (рисунок 2.32).

```
CREATE TABLESPACE ts_ea
  DATAFILE 'C:\app\Tablespaces\ts_EA.dbf'
  SIZE 10 m
  AUTOEXTEND ON NEXT 500K
  MAXSIZE 100M
  EXTENT MANAGEMENT LOCAL;
```

Рисунок 2.32 – Создание постоянного табличного пространства

При создании табличного пространства обычно указывается (рисунок 2.33):

- его имя;
- тип;



- имя файла или нескольких файлов, которые его составляют;
- исходный размер;
- приращение размера файла;
- максимальный размер, если это необходимо.

```
CREATE TEMPORARY TABLESPACE ts_TEMP_ea
  TEMPFILE 'C:\app\Tablespaces\ts__TEMP_EA.dbf'
  SIZE 10 m
  AUTOEXTEND ON NEXT 500K
  MAXSIZE 100M
  EXTENT MANAGEMENT LOCAL;
```

Рисунок 2.33 – Создание временного табличного пространства

### 2.6.3.2 Создание ролей

После аутентификации пользователя в базе данных ему разрешается выполнять в ней действия над данными. Не имея никаких полномочий, пользователь не может выполнять действия с базой данных, например, даже установить соединение с ней. В то же время большинству пользователей не требуется создавать, изменять или удалять таблицы и другие объекты базы данных, поэтому привилегии на создание и удаление таблиц им тоже ни к чему.

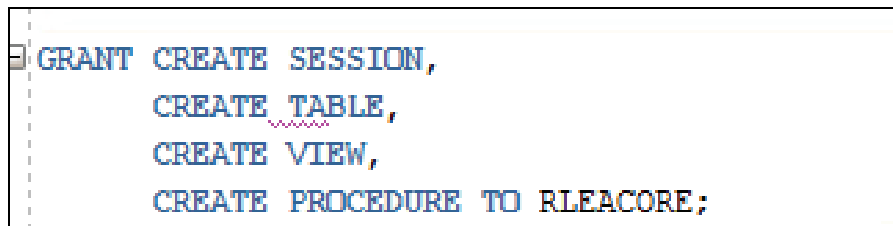
Привилегии могут назначаться как по отдельности, так и набором. Роль – это именованный набор привилегий. Использование ролей существенно облегчает управление привилегиями. Вместо того, чтобы предоставлять десятки полномочий каждому пользователю, можно создать несколько типичных ролей, наделить их необходимыми привилегиями и назначить пользователям их роли. При необходимости и забрать одну роль проще, чем отбирать привилегии по списку.

Создаются, изменяются и удаляются роли стандартными операторами DDL – CREATE, ALTER и DELETE (рисунок 2.34). И полномочия, и роли предоставляются пользователям оператором GRANT, а отбираются оператором REVOKE.

```
CREATE ROLE RLEACORE;
```

Рисунок 2.34 – Создание роли

Привилегии подразделяются на системные и объектные. Системные привилегии предоставляются администратором базы данных и дают пользователю возможность выполнять системную операцию определенного вида. Объектная привилегия позволяет пользователю выполнять операцию определенного вида с конкретным объектом базы данных, например с таблицей, с представлением или с хранимой процедурой. Каждый объект имеет определенный набор предоставляемых привилегий (рисунок 2.35).



```
GRANT CREATE SESSION,  
CREATE TABLE,  
CREATE VIEW,  
CREATE PROCEDURE TO ROLECORE;
```

Рисунок 2.35 – Предоставление привилегий роли

### 2.6.3.3 Назначение системных привилегий

Администратор базы данных имеет высокоуровневые системные привилегии для ряда задач:

- 1) управление сервером;
- 2) управление пользователями;
- 3) управление объектами.

К привилегированным пользователям уровня администратора относятся следующие основные системные пользователи Oracle:

– SYS – предопределенный привилегированный пользователь ранга администратора базы данных, который является владельцем ключевых ресурсов базы данных Oracle;

– SYSTEM – предопределенный привилегированный пользователь, которому принадлежат ключевые ресурсы базы данных Oracle.

Системная роль администратора базы данных называется DBA. Это предопределенная роль, которая автоматически создается для каждой базы данных Oracle и содержит все системные привилегии, кроме специальных системных привилегий SYSDBA и SYSOPER. SYSDBA и SYSOPER – специальные привилегии администратора, которые позволяют выполнять базовые задачи администрирования: запуск или остановка экземпляра базы данных; создание, удаление, открытие или монтирование базы и др.

Привилегии могут быть указаны при подключении пользователя к базе данных.



Перечислим кратко основные системные привилегии:

- 1) CREATE PROCEDURE – можно создать в схеме пользователя хранимую процедуру, функцию или пакет;
- 2) CREATE VIEW – можно создать в схеме пользователя представление;
- 3) CREATE SEQUENCE – можно создать в схеме пользователя последовательность;
- 4) CREATE TABLE – можно создать в схеме пользователя таблицы;
- 5) CREATE SESSION – можно установить соединение с базой данных;
- 6) CREATE ANY TABLE – можно создавать таблицы в любой схеме;
- 7) SELECT ANY TABLE – можно запрашивать данные таблицы, представления или материализованного представления в любой схеме;
- 8) BACKUP ANY TABLE – можно выполнять резервное копирование любой таблицы в любой схеме с помощью служебной программы экспорта;
- 9) DROP ANY TABLE – можно удалить таблицу в любой схеме;
- 10) DROP USER – можно удалить другого пользователя;
- 11) CREATE USER – можно создать других пользователей.

#### 2.6.3.4 Создание профилей безопасности

Назначение профиля безопасности – управление использованием пароля и ограничение количества используемых ресурсов сессии. Для создания профиля применяется команда CREATE PROFILE, в которой устанавливаются необходимые ограничения (рисунок 2.36). Неустановленные значения будут выставлены в зависимости от значений в профиле DEFAULT.

```
CREATE PROFILE PFEACORE LIMIT
PASSWORD_LIFE_TIME 180 -- количество дней жизни пароля
SESSIONS_PER_USER 3 -- количество сессий для пользователя
FAILED_LOGIN_ATTEMPTS 7 -- количество попыток входа
PASSWORD_LOCK_TIME 1 -- количество дней блокирования после ошибок
PASSWORD_REUSE_TIME 10 -- через сколько дней можно повторить пароль
PASSWORD_GRACE_TIME DEFAULT -- количество дней предупреждений о смене пароля
CONNECT_TIME 180 -- время соединения, минут
IDLE_TIME 30 -- количество минут простоя
```

Рисунок 2.36 – Создание профиля безопасности

Могут быть установлены следующие ограничения:

- FAILED\_LOGIN\_ATTEMPTS – устанавливает число последовательных неправильных вводов пароля перед блокировкой аккаунта;
- PASSWORD\_LOCK\_TIME – устанавливает количество в днях до разблокировки аккаунта после блокировки FAILED\_LOGIN\_ATTEMPTS;
- PASSWORD\_LIFE\_TIME – количество в днях до окончания жизни пароля, после этого пароль еще может быть активным какое-то время в зависимости от параметра PASSWORD\_GRACE\_TIME;
- PASSWORD\_GRACE\_TIME – количество в днях после первой успешной попытки подключения после окончания жизни пароля;
- PASSWORD\_REUSE\_TIME – количество дней перед тем, как пароль может быть снова использован;
- PASSWORD\_REUSE\_MAX – устанавливает, сколько раз может быть использован один и тот же пароль;
- PASSWORD\_VERIFY\_FUNCTION – функция проверки нового пароля на необходимый уровень безопасности;
- SESSION\_PER\_USER – количество параллельных подключений, которые может создать один и тот же аккаунт;
- CPU\_PER\_SESSION – время процессора в сентисекундах, которое серверный процесс сессии может использовать до принудительного завершения;
- CPU\_PER\_CALL – время процессора в сентисекундах, которое серверный процесс может использовать для выполнения одной SQL-команды до принудительной остановки выполнения запроса;
- LOGICAL\_READS\_PER\_SESSION – количество блоков, которое может считать сессия перед принудительным отключением сессии;
- LOGICAL\_READS\_PER\_CALL – количество блоков, которое может считать запрос перед принудительной остановкой выполнения запроса;
- PRIVATE\_SGA – количество килобайт, которое сессия может использовать для подключенных сессий;
- CONNECT\_TIME – максимальная длительность сессии в минутах;
- IDLE\_TIME – длительность бездействия в минутах, по достижении которой сессия принудительно отключается.

Когда сессия отключается из-за достижения лимита, неподтвержденная транзакция отменяется, изменения, сделанные запросом, отменяются.

Профиль нельзя удалить, если он назначен какому-либо пользователю. Для удаления профиля можно либо сначала перевести пользователей на другой профиль, либо назначить директиву CASCADE, использование которой автоматически переназначит пользователей на профиль DEFAULT.

#### 2.6.3.5 Создание пользователей и назначение ролей

Пользователь базы данных является учетной записью, через которую можно подключиться к базе данных и получить доступ к данным при наличии соответствующих привилегий.

Администратор базы данных создает пользователя, и на момент создания пользователь не имеет никаких привилегий. Затем администратор базы данных может предоставить этому пользователю определенные привилегии. Этими привилегиями определяется, что именно пользователь может делать на уровне базы данных.

Пользователь создается при помощи команды CREATE USER (рисунок 2.37), параметры пользователя могут быть изменены посредством команды ALTER USER, удаление пользователя производится командой DROP USER.

```
CREATE USER EACORE IDENTIFIED BY 12345
DEFAULT TABLESPACE TBLSP_EA QUOTA UNLIMITED ON TBLSP_EA
TEMPORARY TABLESPACE TBLSP_TEMP_EA
PROFILE PFEACORE
ACCOUNT UNLOCK
PASSWORD EXPIRE
```

Рисунок 2.37 – Создание пользователя

При создании пользователя указывается:

- 1) имя пользователя;
- 2) первоначальный пароль пользователя;
- 3) табличные пространства по умолчанию;
- 4) профиль безопасности;
- 5) дополнительные параметры.

Имя создаваемого пользователя может содержать только символы из набора символов базы данных и должно соответствовать правилам именования. Имя общего пользователя должно начинаться с C## или c##, а имя локального пользователя не должно начинаться с C## или c##.

Пароль указывается для входа в базу данных. Пароли чувствительны к регистру. Пароли могут содержать любые однобайтовые, многобайтовые или специальные символы или любую их комбинацию из набора символов базы данных, за исключением знака двойной кавычки («») и символа возврата. Если пароль содержит символ, отличный от алфавитно-цифрового символа, знак подчеркивания (\_), знак доллара (\$) или знак фунта (#), то этот символ должен быть заключен в двойные кавычки.

Необходимо указывать табличное пространство по умолчанию для объектов, созданных пользователем, иначе объекты пользователя хранятся в табличном пространстве базы данных по умолчанию. Если для базы данных не указано табличное пространство по умолчанию, объекты пользователя хранятся в табличном пространстве SYSTEM.

Выделение квоты позволяет указать максимальный объем пространства, который пользователь может получить в табличном пространстве.

Профиль безопасности ограничивает количество ресурсов базы данных, которые пользователь может использовать. Если профиль не указан, тогда Oracle использует профиль по умолчанию.

Если указаны ключевые слова PASSWORD EXPIRE, то пользователь будет обязан изменить пароль до того, как сможет войти в базу данных.

Если указаны ключевые слова ACCOUNT LOCK, то учетная запись пользователя будет заблокирована. По умолчанию установлено ACCOUNT UNLOCK.

#### **2.6.3.6 Выдача объектных привилегий пользователям**

Привилегия – это право выполнять конкретный тип предложений SQL, или право доступа к объекту другого пользователя. Oracle имеет два вида привилегий: системные и объектные. Привилегии назначаются оператором GRANT и отзываются оператором REVOKE.

Объектная привилегия – это привилегия или право на выполнение определенного действия над отдельной таблицей, последовательностью, процедурой или отдельным представлением.

Владелец объекта обладает всеми привилегиями на свой объект. Владелец может назначить любую объектную привилегию на любой объект схемы, которым он владеет, любому другому пользователю или любой роли.

Если операция предоставления привилегий содержит ключевые слова WITH GRANT OPTION, тогда объектную привилегию можно и

дальше передавать другим пользователям, иначе только получивший привилегии пользователь может пользоваться этой привилегией и не может предоставлять ее другим пользователям.

Каждый объект имеет определенный набор предоставляемых привилегий. В таблице 2.1 перечисляются привилегии для различных объектов.

Таблица 2.1 – Возможные объектные привилегии

Привилегия	TABLE	VIEW	SEQUENCE	PROCEDURE
ALTER	+		+	
DELETE	+	+		
EXECUTE				+
INDEX	+			
INSERT	+	+		
REFERENCES	+			
SELECT	+	+	+	
UPDATE	+	+		

Перечислим свойства объектных привилегий:

- SELECT – можно выполнять запросы для таблицы;
- INSERT – можно добавлять строки в таблицу;
- UPDATE – можно изменять существующие значения в строках таблицы, причем можно ограничить множеством столбцов таблицы;
- DELETE – можно удалять строки из таблицы;
- ALTER – можно изменять структуру только таблицы или последовательности;
- INDEX – можно создавать индекс для ранее описанной таблицы;
- EXECUTE – можно выполнять хранимую процедуру или функцию;
- REFERENCES – можно определить внешний ключ, который использует один или несколько столбцов таблицы в качестве родительского ключа.

Обозначения ALL и ALL PRIVILEGES в команде GRANT или REVOKE назначают или отзывают все возможные объектные привилегии для объекта.

В пояснительной записке требуется указать размещение файлов базы данных, ее файлов и табличных пространств.

Кроме того, необходимо определить порядок аутентификации и авторизации пользователей, в том числе:

- 1) создать необходимые роли;
- 2) предоставить ролям необходимые привилегии;
- 3) создать профили безопасности;
- 4) создать пользователей и назначить им роли.

Обратим внимание на то, что пользователям должны быть предоставлены только необходимые привилегии, т. е. обычный пользователь базы данных не имеет доступа к системному каталогу и не может изменять структуру базы данных.

В пояснительной записке требуется привести все созданные роли и пользователей базы данных с указанием привилегий.

#### **2.6.4 Проектирование и реализация объектов базы данных**

Для решения бизнес-задач, сформулированных в задании на курсовой проект, необходимо спроектировать логическую схему базы данных.

Схема базы данных – это ее структура, которая определяет таблицы, поля в каждой таблице, с указанием их названия, типа, обязательности и ограничения целостности, в том числе первичные и внешние ключи. Следует проверить спроектированную схему на соответствие третьей нормальной форме.

Затем необходимо создать следующие объекты базы данных:

- последовательности;
- представления;
- индексы;
- хранимые процедуры и функции, объединенные в пакеты;
- триггеры.

Схема базы данных должна быть приведена в разделе графического материала.

##### **2.6.4.1 Проектирование и реализация таблиц и ограничений целостности**

В системе управления реляционными базами данных пользовательские и системные данные хранятся в таблицах. Каждая таблица состоит из набора строк, которые описывают сущности, и набора столбцов, которые содержат атрибуты объекта. Проектирование таблиц – одна из наиболее важных задач, которые выполняет разработчик базы данных, поскольку неправильный дизайн таблиц приводит к неэффективности запроса данных.

Ограничения целостности в реляционных базах данных позволяют применять важные бизнес-правила к таблицам базы данных. Для выполнения бизнес-правил можно также использовать и логику приложения, однако ограничения целостности гарантируют, что вставки, обновления и удаления данных таблиц будут соответствовать определенным правилам. В то же время логика приложения позволяет отклонить или подтвердить данные без обращения к содержимому таблицы. Таким образом, выбрать, какой метод выполнения бизнес-правил должен быть использован, необходимо на основе конкретной задачи.

Для таблиц в Oracle 12c можно установить различные типы ограничений:

- 1) ограничения NOT NULL;
- 2) ограничения первичного ключа;
- 3) проверочные ограничения;
- 4) ограничения уникальности;
- 5) ссылочные ограничения целостности.

По умолчанию значения NULL разрешены. Если пустые значения недопустимы, для них необходимо задавать ограничение NOT NULL.

При необходимости идентифицировать значения строк уникальным образом создается первичный ключ по значению столбца или набору столбцов. Столбец, на котором определен первичный ключ, должен быть уникальным и не NULL.

Проверочные ограничения применяются для обеспечения соответствия данных столбца определенным параметрам. Ограничение уникальности гарантирует уникальность строк в реляционной таблице. Ограничения ссылочной целостности гарантируют, что значения определенных важных столбцов будут иметь смысл. В таблице могут существовать несколько ограничений таких видов.

Дополнительно можно указать, когда именно должно проверяться ограничение целостности в процессе обработки данных – немедленно (NOT DEFERRABLE) или проверка будет отложена (DEFERRABLE). В первом случае база данных проверит ограничения перед изменением данных, во втором – проверка ограничения откладывается до окончания транзакции.

Необходимо выполнить проектирование таблиц с указанием правильных типов данных и ограничений целостности, в том числе первичных и внешних ключей.

В пояснительной записке должны быть перечислены все таблицы с указанием типов данных и ограничений целостности и кратко описано, данные о каких объектах находятся в этой таблице.

#### **2.6.4.2 Проектирование и реализация последовательностей**

Последовательность – это объект базы данных, который генерирует целые числа в соответствии с правилами, установленными во время его создания. Для последовательности можно указывать как положительные, так и отрицательные целые числа. В системах баз данных последовательности применяют для самых разных целей, но в основном для автоматической генерации первичных ключей. К первичному ключу таблицы последовательность все же не привязана и может использоваться коллективно.

Необходимо выполнить проектирование и реализацию последовательностей. В пояснительной записке должны быть перечислены все созданные последовательности и указано их назначение.

#### **2.6.4.3 Проектирование и реализация представлений**

Представление – это поименованный SELECT-запрос, который хранится и может использоваться в других запросах точно так же, как и таблица. Представления упрощают дизайн базы данных, формируя слой абстракции и скрывая сложность соединения таблиц. Представления также являются способом защиты данных, предоставляя пользователям разрешения использовать требуемые данные, при этом не позволяя манипулировать исходными объектами, содержащими весь набор данных.

Необходимо выполнить проектирование и реализацию представлений. В пояснительной записке должны быть перечислены все созданные представления и указано их назначение.

#### **2.6.4.4 Проектирование и реализация индексов**

Индексы используются для улучшения производительности запросов или обеспечения уникальности. Индексы позволяют находить строку с определенным значением столбца, просматривая при этом лишь небольшую часть общего объема строк таблицы. Таким образом, правильное использование индексов сокращает до минимума количество дорогостоящих операций ввода-вывода.

Применение индексов представляет собой, с одной стороны, ускорение получения результатов запросов, а с другой – замедление обновлений и вставок данных.

При создании индексов следует руководствоваться следующими правилами:

- если таблицы используются в основном для выборки информации, то можно создать много индексов – по всем столбцам, по которым выполняется поиск;



- если в таблице производится большое количество вставок, обновлений и удалений, то лучше использовать только индекс по первичному ключу;
- индекс применим, если нужно обеспечить доступ одновременно не более чем к 5% данных таблицы;
- желательно избегать создания индексов для сравнительно небольших таблиц;
- следует создавать первичные ключи для всех таблиц;
- рекомендуется индексировать столбцы, участвующие в много-табличных операциях соединения;
- следует индексировать столбцы, участвующие в операциях ORDER BY и GROUP BY или в операциях UNION и DISTINCT, включающих сортировку;
- в составных индексах первым столбцом ключа должен быть столбец, в котором количество строк с одинаковым значением минимально.

На фазе проектирования базы данных после анализа требований приложения необходимо выбрать правильный тип индекса:

1) индексы на основе В-деревьев структурированы в форме обратного дерева. Если количество уровней одинаково от каждого листового блока до корневого узла, то такое дерево называется сбалансированным;

2) битовые индексы используют битовые карты для указания значения индексированного столбца. Битовые индексы состоят из битового потока (единиц и нулей) для каждого столбца индекса. Это лучший индекс для столбца, где мало уникальных записей в таблице при большом размере таблицы;

3) индексы с реверсированным ключом – это индексы В-деревьев, причем байты данных ключевого столбца при индексации меняют порядок на противоположный;

4) индексы на основе функций предварительно вычисляют значения функций по заданному столбцу и сохраняют результат в индексе, используются для конструкций WHERE, содержащих вызовы функций.

Данные индекса постоянно изменяются из-за DML-действий, связанных с его таблицей, становятся слишком большими и разреженными, если удаляется много строк, так как пространство, занятое удаленными значениями, повторно индексом не используется. За счет периодического применения команды REBUILD можно реорганизовать индексы и сделать их более компактными и эффективными.

В пояснительной записке должны быть перечислены все созданные индексы и кратко указан их тип и назначение.

#### **2.6.4.5 Проектирование и реализация пакетов, группирующих хранимые процедуры и функции**

Пакет – это объект базы данных, который группирует логически связанные типы, программные объекты и подпрограммы PL/SQL. Пакеты обычно состоят из двух частей, спецификации и тела. Спецификация пакета объявляет типы, переменные, константы, исключения, курсоры и подпрограммы, доступные для использования в пакете. Тело пакета полностью определяет курсоры и подпрограммы, реализуя спецификацию пакета. В отличие от подпрограмм пакеты нельзя вызывать, передавать им параметры или вкладывать их друг в друга.

Пакеты предлагают следующие преимущества:

- модульность – пакеты позволяют инкапсулировать логически связанные типы, объекты и подпрограммы в поименованный модуль PL/SQL;

- облегчение проектирования – для проектирования приложения надо изначально знать только спецификацию пакета. Спецификация может быть написана и скомпилирована без тела пакета. После того как спецификация откомпилирована, хранимые подпрограммы, обращающиеся к пакету, также могут быть откомпилированы;

- скрытие информации – с помощью пакетов можно указывать, какие типы, объекты и подпрограммы являются видимыми и доступными (*public*) или скрытыми (*private*). Пакет скрывает определение личной процедуры, что упрощает сопровождение и защищает целостность базы данных;

- расширенная функциональность – пакетированные общие переменные и курсоры продолжают существовать в течение всей сессии. Поэтому они могут совместно использоваться всеми процедурами, выполняющимися в данном окружении. Кроме того, через них можно передавать данные между транзакциями без необходимости записывать такие данные в базу данных;

- улучшенная производительность – при вызове процедуры пакета в первый раз в память загружается весь пакет. Поэтому последующие вызовы других подпрограмм этого пакета не требуют операций ввода-вывода. Помимо этого, пакеты останавливают каскадные зависимости и тем самым избегают излишних перекомпиляций. Например, когда вы изменяете определение независимой функции, Oracle должен перекомпилировать все хранимые подпрограммы, которые вызывают

эту функцию. Однако когда вы изменяете определение пакетированной функции, перекомпиляция вызывающих подпрограмм не требуется, потому что они не зависят от тела пакета.

Процедуры и функции – это подпрограммы, которые используются для инкапсуляции часто выполняемой логики. Вместо того чтобы повторять логику подпрограммы во многих местах, код может вызывать процедуру или функцию. Это делает код более удобным для обслуживания и легче для отладки. Процедуры также являются способом организации удобного доступа к данным из приложения, предоставляя приложению вызывать процедуру для использования требуемых данных, при этом не позволяя манипулировать исходными объектами, содержащими весь набор данных.

Необходимо выполнить проектирование и реализацию хранимых процедур с параметрами:

- 1) для реализации бизнес-логики задачи;
- 2) вставки данных;
- 3) обновления данных;
- 4) удаления данных;
- 5) выборки данных.

Выборка данных подразумевает, что может быть получен как один объект, так и их набор. В результате выполнения выборки при помощи процедуры должен быть получен набор данных, не требующий дополнительной обработки, например, каких-либо вычислений или сортировки.

В процедурах должна быть реализована обработка ошибок. При необходимости должна быть использована обработка бизнес-логики в явной транзакции и установлен необходимый уровень изоляции транзакций.

Также следует выполнить проектирование и реализацию функций, определенных пользователем, для реализации бизнес-логики задачи. При необходимости могут быть использованы как скалярные, так и табличные функции.

В пояснительной записке должны быть перечислены все процедуры и функции с описанием входных и выходных параметров (при их наличии) и возвращаемых значений с указанием типов данных и оговорено, что именно выполняет каждая хранимая процедура или функция, определенная пользователем.

#### **2.6.4.6 Проектирование и реализация триггеров**

Триггеры языка манипуляции данными (DML) – это мощные инструменты, которые можно использовать для обеспечения доменной

целостности, контроля сущности реляционных данных и бизнес-логики. Обеспечение целостности помогает создавать надежные приложения.

Следует выполнить проектирование и реализацию необходимых триггеров для поддержания доменной целостности, контроля сущности реляционных данных и реализации бизнес-логики базы данных. Рекомендуется создать триггеры аудита. При необходимости можно использовать DDL-триггеры.

В пояснительной записке должны быть перечислены все триггеры и указано, что именно выполняет каждый триггер и по какому условию.

### **2.6.5 Изучение и применение технологии в созданной базе данных**

Необходимо изучить и развернуто описать избранную технологию для СУБД. В записке должно быть подробно отражено следующее:

- формальное описание назначения технологии;
- примеры приложений с использованием данной технологии;
- описание использования технологии в базе данных.

В приложении должны быть приведены листинги процедур, использующих эту технологию.

### **2.6.6 Импорт и генерация тестовых данных**

Для демонстрации функционирования базы данных и оптимизации запросов к ней необходимо произвести импорт и экспорт данных. Данные могут импортироваться и/или дополнительно генерироваться, причем импорт и экспорт может производиться для данных следующих форматов:

- 1) табличных данных;
- 2) данных в формате XML или JSON;
- 3) данных в текстовом формате;
- 4) с использованием команд SQL\*Loader или утилиты Import.

Рекомендуется создать процедуры импорта и экспорта данных и привести их в пояснительной записке. Листинги процедур следует разместить в приложении.

Также данные могут быть сгенерированы любым удобным для вас способом. Для проведения оптимизации запросов в базе данных необходимо, чтобы количество строк в таблицах, к которым выполняются запросы, было не менее 100 000 строк.

Обратим внимание на то, что при загрузке больших объемов данных может потребоваться значительное время на загрузку данных, каждую строку которых нужно проверять на предмет нарушения целостности. Более эффективная стратегия предусматривает отключение ограничений целостности (DISABLE VALIDATE), загрузку данных и последующую проверку корректности загруженных данных. После завершения загрузки ограничения снова включаются (ENABLE VALIDATE). Когда ограничение отключается, база данных уничтожает индекс.

Для оптимизации запросов в базе данных следует провести исследование SELECT-запросов и создать необходимые индексы и/или переформулировать запросы.

В пояснительной записке необходимо описать планы выполнения запросов до оптимизации, предложенные меры и состояние после оптимизации.

### **2.6.7 Резервное копирование и восстановление данных**

Следует описать схему резервного копирования и восстановления данных с учетом решаемой бизнес-задачи.

В самом простом случае необходимо выполнить полное резервное копирование данных и восстановление из резервной копии. В пояснительной записке следует описать скрипты создания резервной копии и восстановления из нее. При использовании специализированных утилит важно указать порядок использования утилиты.

Для получения более высокой оценки необходимо создать план резервного копирования и восстановления данных. Следует принимать во внимание различные аспекты потери данных: сбой хранения, сбой при передаче, умышленные или неумышленные действия третьих лиц. Также необходимо учитывать режим работы с данными: требуется ли режим работы 24/7, сколько времени может составлять допустимый простой и на какой момент времени должны быть восстановлены данные. Все эти аспекты должны быть описаны в пояснительной записке.

Во время защиты может потребоваться продемонстрировать потерю данных и их восстановление.

### **2.6.8 Разработка демонстрационного приложения**

Для демонстрации работы необходимо представить результаты курсового проекта в приложении.

Основная цель разработки демонстрационного приложения – показать все стороны решения задачи и выбранную технологию. Среду

разработки, язык, структуру и внешний вид приложения студент определяет самостоятельно.

Доступ к данным базы данных осуществляется только при помощи вызова соответствующих процедур и/или функций.

В пояснительной записке приводится краткое описание приложения. Структура, UML-диаграмма, основные классы могут быть вынесены в приложение.

## **2.7 Заключение**

В заключении формулируются краткие выводы по результатам выполненной работы. Приводятся количественные и качественные характеристики реализации базы данных: решенные бизнес-задачи, результаты применения избранной технологии в базе данных и т. п.

## **2.8 Графический материал**

Графический материал выполняется на листе формата А4 по ГОСТ 2.301 и представляет собой логическую схему базы данных.

## **2.9 Список использованных источников**

Необходимо перечислить книги, статьи, электронные ресурсы, которые были использованы при выполнении работы. Информация о правилах оформления этого списка приведена в пункте 3.12. Список использованных источников должен содержать не менее пяти наименований.

## **2.10 Приложения**

Приложения содержат объемные материалы или материалы справочного характера, такие как схема базы данных, блок-схемы алгоритмов решения бизнес-задач, тексты SQL-скриптов, UML-диаграмму приложения, краткое описание классов приложения и т. д.

### **3 Оформление пояснительной записки**

#### **3.1 Общие требования**

При оформлении пояснительной записки следует руководствоваться требованиями СТП БГТУ 001-2010 Проекты (работы) дипломные. Требования и порядок подготовки, представление к защите и защиты.

Пояснительная записка к курсовому проекту оформляется в соответствии с правилами оформления текстовых документов, изложенными в ГОСТ 2.105–1995 Единая система конструкторской документации. Общие требования к текстовым документам, и с правилами оформления курсовых проектов, приведенными в данном издании.

Текст записки должен быть напечатан на одной стороне листа формата А4. Используемый шрифт – Times New Roman, размер шрифта – 14 пт, межстрочный интервал – одинарный. Цвет шрифта – черный. Абзацный отступ – 1,25 см. В тексте после знаков препинания обязательно ставится пробел. Нельзя сокращать слова (кроме сокращений, установленных правилами орфографии). Графический материал оформляется на листах формата А3 с рамкой.

Текст следует печатать, соблюдая поля: правое –  $(10 \pm 1)$  мм; верхнее –  $(20 \pm 1)$  мм; левое –  $(23 \pm 1)$  мм; нижнее –  $(15 \pm 1)$  мм. При наличии на листе рамки и основной надписи по форме 2 расстояние между верхней границей основной надписи и последней строкой текста, если лист полностью заполняется текстом, должно составлять 10–15 мм.

#### **3.2 Структурные элементы записки**

Структурные элементы записки: «Содержание», «Реферат», «Введение», «Заключение», «Список использованных источников», «Графический материал», а также каждый из основных разделов и каждое из приложений следует начинать с нового листа.

Заголовки разделов, подразделов, а также элементов текста «Содержание», «Реферат», «Введение», «Заключение», «Список использованных источников», «Графический материал» следует записывать в начале соответствующих страниц строчными буквами, кроме первой прописной, с абзацного отступа полужирным шрифтом и отделять от текста интервалом в 18 пт. Точку между последней цифрой номера заголовка и текстом заголовка не ставят. Также не ставят точку в конце

заголовка. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, то их разделяют точкой.

Текст основной части делят на разделы, разделы – на подразделы, подразделы – на пункты и т. д. Разделы нумеруются арабскими цифрами, точка в конце не ставится; подразделы нумеруются в пределах раздела. Номер раздела отделяется от номера подраздела точкой.

Запрещено переносить заголовки подразделов и пунктов со страницы на страницу, а также записывать их в конце текста, если после указанных заголовков размещается менее двух строк излагаемого материала.

### **3.3 Нумерация страниц**

Нумерация страниц пояснительной записки должна быть сквозной. На титульном листе (первой странице пояснительной записки) номер не указывается. Номер проставляется арабской цифрой без точки в правом верхнем углу страницы.

Графический материал, размещенный на листе формата А3, учитывается как одна страница.

### **3.4 Перечисления**

В тексте пояснительной записки могут быть использованы перечисления. Пункты перечисления записывают после двоеточия в виде списка, каждый с абзачного отступа. Перед каждым пунктом нумерованного списка перечисления следует ставить тире. При необходимости ссылки в тексте на один или несколько пунктов списка перечисление оформляют в виде маркированного списка, каждый пункт начинают со строчной буквы русского алфавита (за исключением г, ё, з, й, о, ь, ы, ь) с проставленной после нее круглой скобкой. Для дальнейшей детализации перечислений (сложные перечисления) необходимо использовать арабские цифры с проставленными после них круглыми скобками. Запись подчиненных пунктов сложного перечисления выполняют с абзачными отступами по отношению к основному.

Ниже приведен пример выполнения простого перечисления.

Перечислим объекты базы данных, используемые в настоящем курсовом проекте:

- таблицы;
- представления;



- индексы;
- хранимые процедуры и функции, определенные пользователем;
- триггеры.

### 3.5 Изложение текста

Текст пояснительной записки должен быть кратким, четким и не допускать различных толкований. В тексте пояснительной записки не допускается:

- 1) применять обороты разговорной речи;
- 2) использовать для одного и того же понятия различные термины;
- 3) употреблять сокращения слов, кроме установленных правилами орфографии русского языка.

Перечень допускаемых сокращений русских слов установлен в ГОСТ 2.316–2008 и ГОСТ 7.12–1993, белорусских – в СТБ 7.12–2001.

Если в пояснительной записке принята особая система сокращения слов или наименований, то в ней должен быть приведен перечень принятых сокращений.

### 3.6 Формулы

Формулы располагаются в тексте в отдельных строках, по центру строки. Формулы нумеруются арабскими цифрами в пределах раздела. Номера формул включают последовательно номер раздела и порядковый номер формулы, разделенные точкой. Номера формул записываются на уровне формулы в круглых скобках справа в конце строки.

Пояснения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой. Пояснения должны начинаться со слова «где». Далее следует описание каждого символа в той последовательности, в которой они приведены в формуле по одному в строке.

Размер шрифта символов в формулах и уравнениях должен соответствовать размеру основного шрифта текста. Размер индексов при основных символах в формулах и уравнениях – 9 пт.

Ссылки на формулы, ранее приведенные в тексте пояснительной записки, а также на формулы в приложениях необходимо выполнять с использованием их номера, например: «...по формуле (2.8)...», «...расчетная зависимость (А.6)...».

### 3.7 Примечания

Примечания приводятся в пояснительной записке, если необходимы пояснения по содержанию текста, таблиц или иллюстраций.

Примечания необходимо помещать непосредственно после текстового материала (рекомендуется в конце пункта, подпункта), таблицы или графического материала, к которым они относятся. Если примечание одно, то после слова «Примечание» следует ставить тире, а за ним с прописной буквы набирать текст примечания. Одно примечание не нумеруется. Несколько примечаний нумеруют по порядку арабскими цифрами. Номер примечания от его текста точкой не отделяют. Примечание к таблице необходимо помещать в конце таблицы над обозначающей ее окончание чертой.

Текст примечаний рекомендуется печатать размером шрифта 12 пт.

### 3.8 Рисунки

Все рисунки (схемы, графики, структурные схемы и др.) должны располагаться непосредственно после ссылки на них в тексте. Рисунок заверстывают так, чтобы его удобно было смотреть без поворота листа или с поворотом по часовой стрелке. Рисунки нумеруются арабскими цифрами в пределах раздела и приложения. Номер рисунка включает номер раздела (обозначение приложения) и порядковый номер по разделу (приложению), разделенные точкой, например: Рисунок 2.5; Рисунок А.1.

Рисунки должны иметь наименование. При необходимости рисунки снабжаются пояснительными данными (подрисуночным текстом). Слово «Рисунок» с порядковым номером и наименование рисунка помещают после пояснительных данных и записывают следующим образом:

Рисунок 6.3 – Результат резервного копирования

Размер шрифта заголовков и подрисуночных надписей иллюстраций должен соответствовать размеру основного шрифта текста. Рисунок отделяют от текста интервалом в 14 пт.

Не допускается отрыв (перенос со страницы на страницу) рисунка от подрисуночного текста.

### 3.9 Таблицы

Таблицы нумеруются арабскими цифрами в пределах раздела. Номер таблицы включает номер раздела (обозначение приложения) и порядковый номер по разделу (приложению), разделенные точкой,

например: Таблица 3.4; Таблица А.2. Запись заголовка таблицы начинают со слова «Таблица», первую букву которого размещают над левым углом таблицы, затем пишут ее номер, а далее через тире строчными буквами, кроме первой прописной, название таблицы.

Пример:

Таблица 3.4 – Список процедур и функций

--	--	--	--

Если таблица выходит за размер листа, то ее делят на части. При переносе части таблицы на другой лист заголовок помещают только над первой частью, над остальными частями пишут «Продолжение таблицы» с указанием номера таблицы. Шапку таблицы при переносе части таблицы повторяют. Заголовки строк и столбцов пишут с прописной буквы.

Таблицу располагают в пояснительной записке непосредственно после текста, в котором она упоминается. Допускается выполнять таблицы, размещая их вдоль длинной стороны листа таким образом, чтобы таблица читалась при повороте листа на 90° по часовой стрелке.

Размер шрифта заголовков и текста таблиц должен соответствовать размеру основного шрифта текста. Таблицу следует отделять от текста интервалом в 14 пт. Таблицы слева, справа и снизу ограничивают линиями.

### 3.10 Ссылки

Ссылки на разделы, подразделы, перечисления, таблицы, иллюстрации, формулы и приложения пояснительной записки следует оформлять по следующим примерам:

- «...структура базы данных, описанная в разделе 2...»;
- «...по пункту б) перечисления...»;
- «...символы приведены в таблице 1.1...»;
- «...изображено на рисунке 3.8...»;
- «...текст процедуры представлен в приложении Д...».

Ссылку на литературный источник выполняют с указанием порядкового номера источника, под которым он внесен в «Список использованных источников» пояснительной записки. Номер ссылки проставляется арабскими цифрами в квадратных скобках, например, [3], [1, 2, 5].

Пример ссылки на источники приведен ниже:

- 1) «...согласно п. 3.4 стандарта [7]...»;
- 2) «...на рисунке 3.2 учебника [5]...».

### 3.11 Приложения

Каждое приложение начинается с нового листа с указанием наверху посередине страницы слова «Приложение» с первой прописной буквы и его обозначения.

Приложения по ГОСТ 2.105–1995 обозначаются заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ь. После слова «Приложение» пишется буква, идентифицирующая его последовательность. Если в документе одно приложение, оно обозначается «Приложение А».

Приложения должны иметь заголовки, которые записывают симметрично тексту с прописной буквы в следующей строке. Затем через одну пустую строчку следует текст приложения.

### 3.12 Список использованных источников

Источники – это книги, учебники, статьи из научных журналов и интернета и т. д., использованные при выполнении курсового проекта. Источники в списке располагаются в порядке ссылок на них в тексте пояснительной записки или по алфавиту, нумеруются арабскими цифрами без точки и печатаются с абзацного отступа, при этом дается библиографическое описание каждого источника в соответствии с ГОСТ 7.1–2003, ГОСТ 7.12–1993.

Примеры оформления списка использованных источников приведены ниже.

Характеристика источника	Пример оформления
Один, два или три автора	Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. – М.: Вильямс, 2005. – 328 с.
	Мишра, С. Oracle SQL / С. Мишра. – СПб.: Символ, 2003. – 360 с.
	Кайт, Т. Oracle для профессионалов: архитектура и методики программирования / Т. Кайт, Д. Кун. – 3-е изд. – М.: Вильямс, 2017. – 960 с.
Четыре и более авторов	Oracle PL/SQL для профессионалов: практические решения / К. МакДональд [и др.]; пер. с англ. – СПб.: Диасофт, 2015. – 560 с.

## **4 Примерная тематика курсовых проектов**

### **4.1 Общие замечания**

Для написания курсового проекта вам потребуется выбрать тематику задачи и технологию СУБД для самостоятельного изучения.

Возможен выбор других СУБД, кроме Oracle 12c, по согласованию с преподавателем.

Важно проконсультироваться с преподавателем по выбору задачи и технологии. Возможно, что для решения некоторых задач потребуются специфические технологии.

### **4.2 Список областей применения баз данных**

Можно разработать базу данных:

- для авиакомпании;
- железнодорожного вокзала;
- агентства недвижимости;
- финансово-консалтингового агентства;
- службы занятости;
- промышленного предприятия;
- страховой компании;
- структуры законодательной власти;
- сети санаторно-курортных учреждений;
- сети кинотеатров;
- сети супермаркетов;
- службы доставки;
- гостиничного комплекса;
- сети предприятий общественного питания;
- сети образовательных учреждений разного уровня (школы, колледжи, училища, учреждения высшего образования и т. д.);
- культурно-развлекательного учреждения;
- издательства, занимающегося выпуском, например, газет, журналов;
- системы тестирования;
- туристического агентства;
- сети сотовой связи;
- электронной площадки по предоставлению услуг;

- аукционной площадки;
- составления расписания занятий в учреждении высшего образования;
- планирования совместных действий (поездки и т. д.);
- медицинского центра;
- физкультурно-оздоровительного центра;
- магазина автомобилей;
- магазина средств связи;
- магазина бытовой техники и т. д.

#### **4.3 Примерный список рекомендуемых технологий для самостоятельного изучения**

Если не указано иное, то предполагается, что технология может быть реализована для выбранной СУБД (MS SQL Server, Oracle, PostgreSQL), если это возможно. Предлагаются следующие технологии для самостоятельного изучения и применения в курсовом проекте:

- 1) применение технологии Oracle 12c Flashback;
- 2) применение СУБД Oracle TimesTen;
- 3) применение средств трассировки в СУБД Oracle 12c;
- 4) применение CRM-системы Oracle Apex;
- 5) применение возможностей аудита сервера СУБД;
- 6) установка и настройка Oracle RAC СУБД Oracle 12c;
- 7) реализация холодного копирования сервера СУБД Oracle 12c;
- 8) реализация горячего копирования сервера СУБД Oracle 12c;
- 9) применение Java для разработки программных объектов в СУБД Oracle 12c;
- 10) применение мультимедийных типов данных в базе данных СУБД Oracle 12c;
- 11) применение пространственных типов данных в базе данных СУБД Oracle 12c;
- 12) обработка графов в СУБД Oracle 12c;
- 13) шифрование и маскирование данных в СУБД Oracle 12c;
- 14) применение полнотекстового поиска в СУБД Oracle 12c;
- 15) репликация данных между серверами СУБД Oracle 12c;
- 16) настройка системы безопасности сервера СУБД Oracle 12c;
- 17) применение средств мониторинга состояния СУБД Oracle 12c;
- 18) применение облачной версии СУБД Oracle 12c;
- 19) сравнение SQL- и NoSQL-решений.

## 5 Демонстрационный пример

В демонстрационном примере приводятся выдержки из пояснительной записки к курсовому проекту студента III курса. Пример специально сокращен и предназначен только для справки. Количество объектов базы данных указано не полностью. Некоторые разделы намеренно отсутствуют, остальные значительно сокращены. Приложения с SQL-скриптами не приводятся. При выполнении курсового проекта вы должны руководствоваться своей постановкой задачи.

### 5.1 Постановка задачи

Целью данного курсового проекта является анализ возможностей горячего резервного копирования Oracle. Для достижения цели поставлены следующие задачи:

- проектирование и разработка базы данных «Перевозки»;
- изучение существующих способов горячего резервного копирования в СУБД Oracle 12c;
- разработка программного средства для резервного копирования базы данных Oracle.

### 5.2 Проектирование и разработка базы данных

Информация о таблицах базы данных приведена в таблице 5.1.

Таблица 5.1 – Таблицы базы данных

№ п/п	Наименование таблицы	Пояснение
1	Город	Справочник пунктов отправления, назначения и промежуточных остановок
2	Пассажир	Данные о пассажирах
3	Водитель	Данные о водителях
4	Транспортное_средство	Данные о транспортных средствах
5	Маршрут	Информация обо всех маршрутах
6	Рейс	Информация о запланированных рейсах
7	Билет	Информация о проданных билетах

Таблица Рейс хранит информацию обо всех запланированных рейсах. Структура таблицы представлена в таблице 5.2.

Таблица 5.2 – Столбцы таблицы Рейс и их описание

№ п/п	Наименование столбца	Пояснение
1	Номер_рейса	Уникальный номер рейса
2	Дата_отправления	Дата выполнения рейса с указанием даты/времени
3	Дата_прибытия	Дата прибытия рейса с указанием даты/времени
4	Маршрут	Ссылка на таблицу Маршрут
5	Транспортное_средство	Ссылка на таблицу Транспортное_средство
6	Водитель	Ссылка на таблицу Водитель
7	Место_отправления	Место отправления рейса (платформа), обязательное
8	Место_прибытия	Место прибытия рейса (платформа), обязательное
9	Примечания	Примечания (необязательное)

Скрипты по созданию таблиц базы данных приведены в приложении.

### 5.3 Проектирование и разработка хранимых процедур

Для доступа к данным и решения бизнес-задач были реализованы процедуры и функции, представленные в таблице 5.3.

Таблица 5.3 – Список процедур и функций и их описания

№ п/п	Наименование процедуры или функции	Пояснение
1	City_Create	Процедура, которая добавляет данные в таблицу Город
2	Route_Inner_Cities	Функция, позволяющая получить промежуточные города по номеру рейса
3	Client_Routes	Функция, позволяющая получить маршруты определенного пассажира за определенный период времени



Все скрипты по созданию процедур и функций базы данных приведены в приложении.

## 5.4 Проектирование и разработка триггеров

Для реализации целостности данных и решения бизнес-задач были спроектированы следующие триггеры (таблица 5.4).

Таблица 5.4 – Список триггеров и их описания

№ п/п	Наименование триггера	Пояснение
1	Ticket_Create	Обновление количества свободных мест на рейс при покупке билета

Все скрипты по созданию триггеров базы данных приведены в приложении.

## 5.5 Способы резервного копирования в Oracle

### 5.5.1 Общие сведения

Когда база данных находится в рабочем состоянии, содержимое файлов базы данных постоянно изменяется. Корректируются файлы базы данных и управляющие файлы, записываются и архивируются журналы повтора.

Операции резервного копирования базы данных, выполняемые под управлением пользователя, помещают каждое табличное пространство в режим горячего резервного копирования и создают резервные копии файлов данных. После выполнения резервных копий файлов данных табличные пространства возвращаются обратно в свое обычное состояние.

После создания этих горячих резервных копий открытой базы данных появятся файлы, которые можно восстановить в случае сбоя.

Выполнение резервных копий на открытой базе данных позволяет пользователям и приложениям использовать всю базу данных во время операций резервного копирования.

Доступность базы данных во время проведения операции ее резервного копирования является главным и очевидным преимуществом

горячего резервного копирования над холодным копированием. Однако это влечет и ряд недостатков. Резервные файлы данных, полученные в результате открытого резервирования, не согласованы, что является следствием работы пользователей и приложений с базой данных в момент ее копирования. Следовательно, для компенсации этой несогласованности требуется предусматривать специальные механизмы. Также сам механизм копирования, не предотвращающий доступ пользователей к копируемым данным, требует некоторых процедур и, соответственно, ресурсов сервера.

Прежде всего необходимо отметить, что для работы любого механизма горячего резервирования используются архивы журналов повторного выполнения. По умолчанию Oracle не архивирует устаревшие журналы повторного выполнения, поэтому следует самостоятельно включить архивирование журналов.

### **5.5.2 Горячее резервирование с использованием утилиты RMAN**

Менеджер восстановления (Recovery Manager, RMAN) – инструмент Oracle, который можно использовать для резервного копирования, восстановления и возвращения к предшествующему состоянию баз данных Oracle.

RMAN является интерпретатором командных строк, преобразующим команды пользователя, выполняемые на базе данных, для копирования и восстановления данных. Каждое резервное копирование, созданное с использованием RMAN, сохраняется, а информация о резервной копии записывается в управляющий файл той базы данных, для которой была создана резервная копия, и в каталог восстановления (Recovery Catalog).

Используя различные представления базы данных или команды RMAN, можно обращаться с запросом к управляющему файлу или к каталогу восстановления, чтобы показать состояние полученных резервных копий. RMAN называется «восстановлением под управлением сервера», поскольку он управляет большинством аспектов резервного копирования, восстановления и возвращения к предшествующему состоянию.

На рисунке 5.1 продемонстрирован процесс сохранения текущего состояния базы данных.

Чтобы эмулировать потерю данных, удалим несколько файлов (рисунки 5.2 и 5.3).

```

Administrator: Command Prompt - rman target/
RMAN> backup database;

Starting backup at 21-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=24 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=C:\APP\ADMINISTRATOR\ORADATA\ORCL\SYSTEM01.DBF
input datafile file number=00003 name=C:\APP\ADMINISTRATOR\ORADATA\ORCL\SYSAUX01.DBF
input datafile file number=00005 name=C:\APP\ADMINISTRATOR\ORADATA\ORCL\UNDOTBS01.DBF
input datafile file number=00015 name=C:\TBS\RMAN1.DBF
input datafile file number=00006 name=C:\APP\ADMINISTRATOR\ORADATA\ORCL\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 21-MAY-16
channel ORA_DISK_1: finished piece 1 at 21-MAY-16
piece handle=C:\APP\ADMINISTRATOR\FAST_RECOVERY_AREA\ORCL\BACKUPSET\2016_05_21\O1_MF_NNNDP_TAG20160521T104519_CN04JJB4_.BKP tag=TAG20160521T104519 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:14
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00011 name=C:\APP\ADMINISTRATOR\ORADATA\ORCL\KMU_PDB\SYSAUX01.DBF
input datafile file number=00010 name=C:\APP\ADMINISTRATOR\ORADATA\ORCL\KMU_PDB\

```

Рисунок 5.1 – Процесс бекапа базы данных

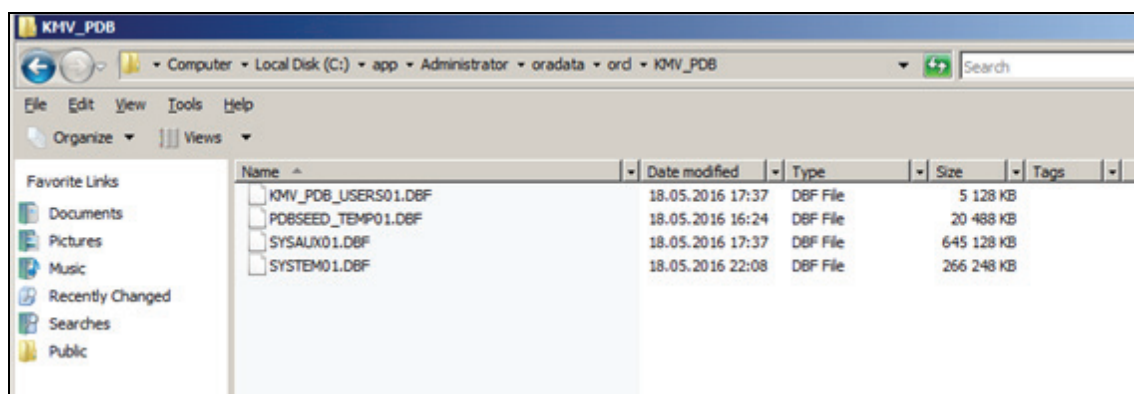


Рисунок 5.2 – Файлы данных

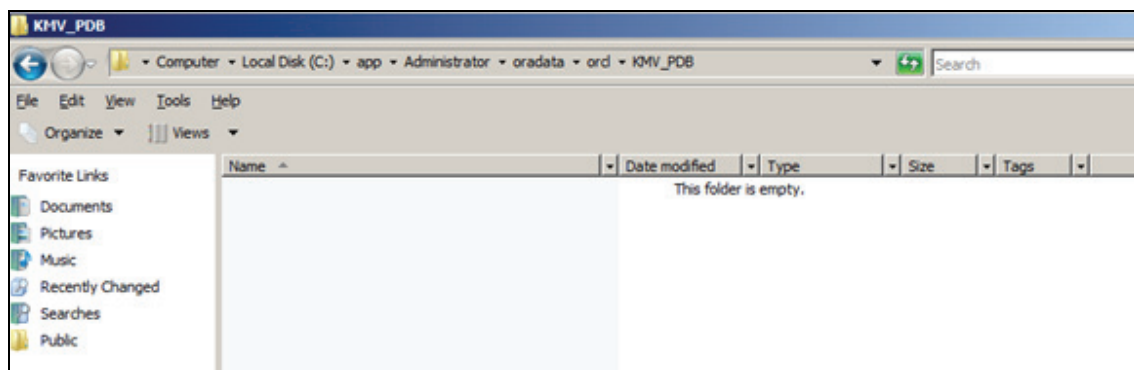


Рисунок 5.3 – Окно с удаленными файлами данных

Далее восстановим базу данных при помощи команды RESTORE DATABASE (рисунок 5.4).

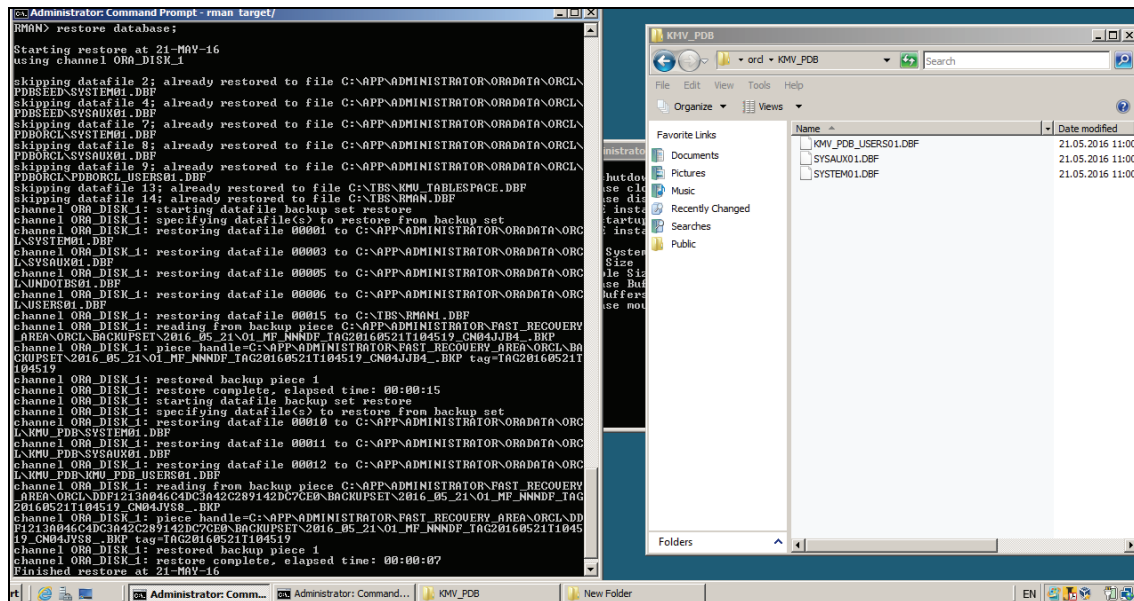


Рисунок 5.4 – Консольное окно с процессом восстановления базы данных и папка с восстановленной PDB

Как видно, RMAN восстановил утерянные файлы. Попробуем открыть сервер и PDB (рисунок 5.5).

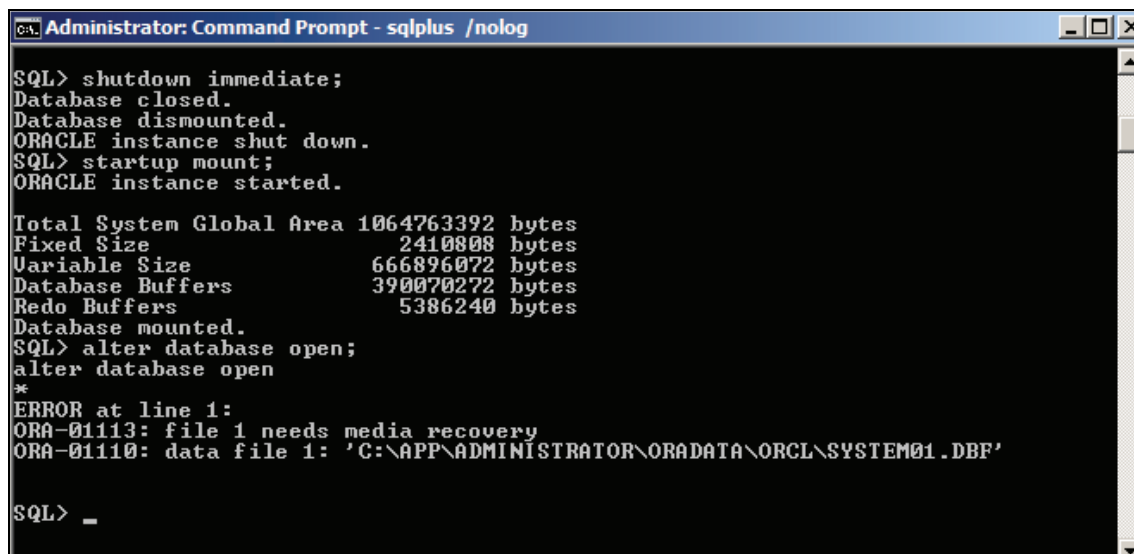
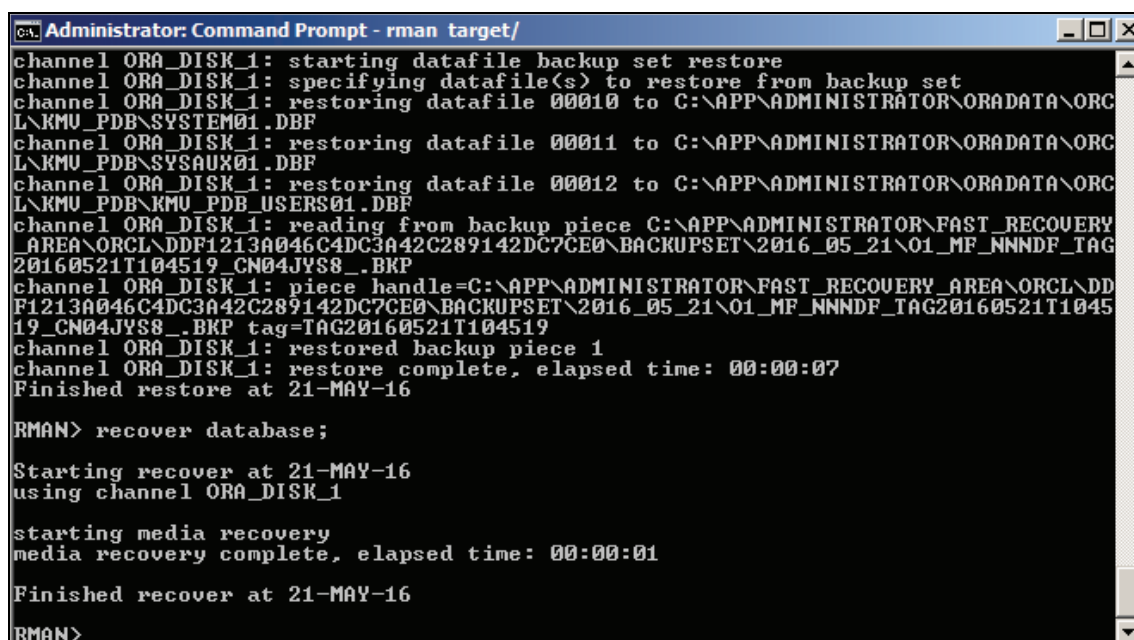


Рисунок 5.5 – Попытка открытия инстанса Oracle

Открыть базу данных не получается, проблема с файлом SYSTEM01.DBF. Снова выполним команду RECOVER DATABASE (рисунок 5.6).



```
Administrator: Command Prompt - rman target/
channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00010 to C:\APP\ADMINISTRATOR\ORADATA\ORCL\KMU_PDB\SYSTEM01.DBF
channel ORA_DISK_1: restoring datafile 00011 to C:\APP\ADMINISTRATOR\ORADATA\ORCL\KMU_PDB\SYSAUX01.DBF
channel ORA_DISK_1: restoring datafile 00012 to C:\APP\ADMINISTRATOR\ORADATA\ORCL\KMU_PDB\KMU_PDB_USERS01.DBF
channel ORA_DISK_1: reading from backup piece C:\APP\ADMINISTRATOR\FAST_RECOVERY_AREA\ORCL\DDF1213A046C4DC3A42C289142DC7CE0\BACKUPSET\2016_05_21\01_MF_NNDF_TAG20160521T104519_CN04JVS8_.BKP
channel ORA_DISK_1: piece handle=C:\APP\ADMINISTRATOR\FAST_RECOVERY_AREA\ORCL\DDF1213A046C4DC3A42C289142DC7CE0\BACKUPSET\2016_05_21\01_MF_NNDF_TAG20160521T104519_CN04JVS8_.BKP tag=TAG20160521T104519
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:07
Finished restore at 21-MAY-16

RMAN> recover database;

Starting recover at 21-MAY-16
using channel ORA_DISK_1

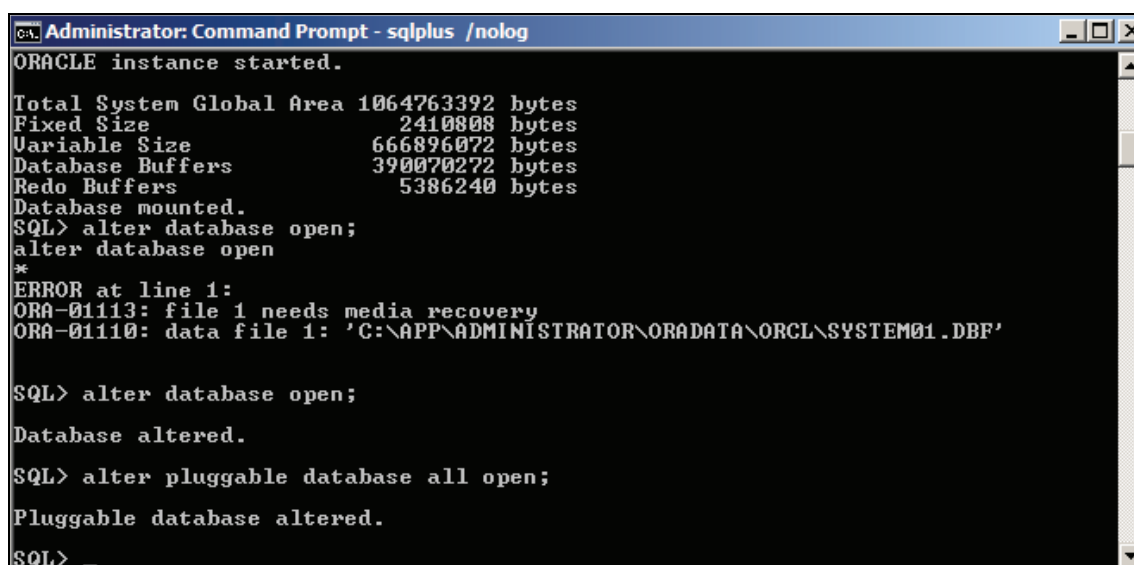
starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 21-MAY-16

RMAN> _
```

Рисунок 5.6 – Восстановление остальных файлов через RMAN

Опять попытаемся открыть сервер и PDB. Как видим, все прошло успешно (рисунок 5.7).



```
Administrator: Command Prompt - sqlplus /nolog
ORACLE instance started.

Total System Global Area 1064763392 bytes
Fixed Size                2410880 bytes
Variable Size             666896072 bytes
Database Buffers          390070272 bytes
Redo Buffers              5386240 bytes
Database mounted.
SQL> alter database open;
alter database open
*
ERROR at line 1:
ORA-01113: file 1 needs media recovery
ORA-01110: data file 1: 'C:\APP\ADMINISTRATOR\ORADATA\ORCL\SYSTEM01.DBF'

SQL> alter database open;
Database altered.

SQL> alter pluggable database all open;
Pluggable database altered.

SQL> _
```

Рисунок 5.7 – Успешное открытие инстанса

### 5.5.3 Утилита Handy Backup

Handy Backup – программа, которая разработана компанией «Новософт», позволяет осуществлять резервное копирование, восстановление и синхронизацию данных из различных источников (рисунк 5.8).

Резервное копирование данных может производиться по сложному расписанию, включая еженедельное и ежемесячное, и отправлять уведомления о бекапах администратору по e-mail.

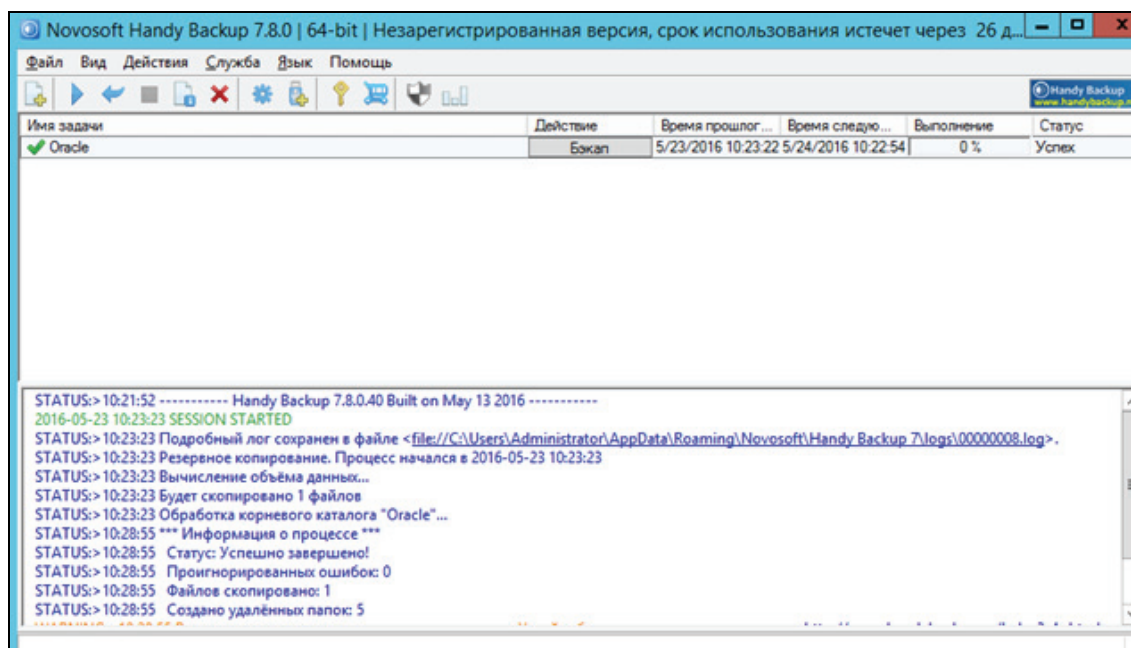


Рисунок 5.8 – Успешное выполнение бекапа

### 5.5.4 Утилита Iperius Backup

Iperius Backup позволяет проводить автоматическое резервное копирование файлов, папок и серверов на любые носители информации, такие как NAS, внешние диски USB, носители RDX и компьютеры в сети. С помощью данной программы можно надежно защитить ценные данные и при необходимости восстановить их в случае вирусных атак, системных и других сбоев (рисунок 5.9).

Изучения этих утилит показало, что они все используют RMAN для своих резервных копий. На основе данных утилит было разработано приложение SMA Backup для горячего резервирования баз данных как по расписанию, так и немедленно, с проверкой целостности резервных копий.



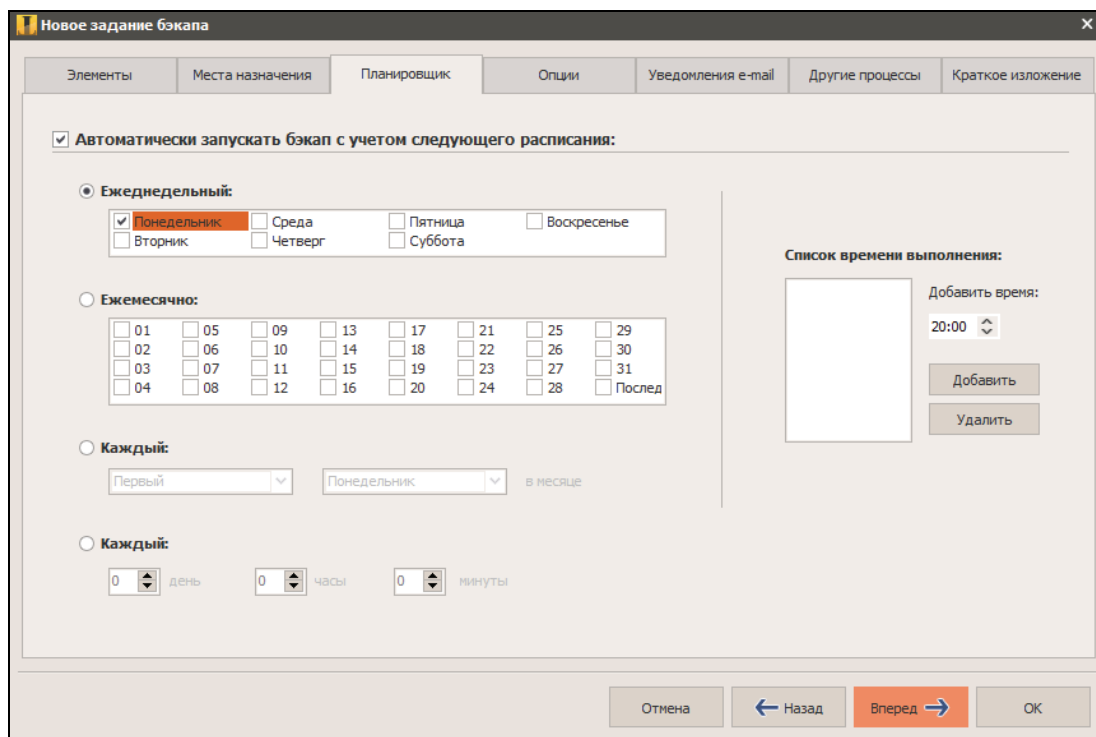


Рисунок 5.9 – Создание задания для бекапа базы данных

### 5.5.5 Приложение для горячего резервирования SMA Backup

Для пользования программой нужны минимальные знания утилиты RMAN и режимов работы сервера Oracle. Продемонстрируем выполнение резервных копий базы данных (рисунок 5.10).

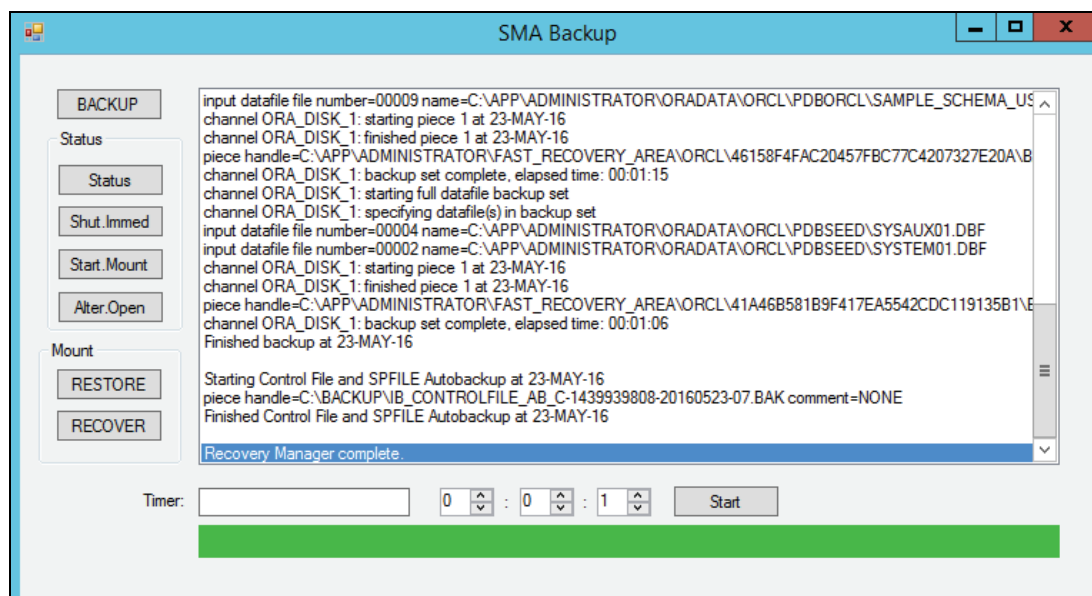


Рисунок 5.10 – Процесс бекапа

Резервное копирование прошло успешно, теперь удалим файлы базы данных и попытаемся восстановиться (рисунок 5.11).

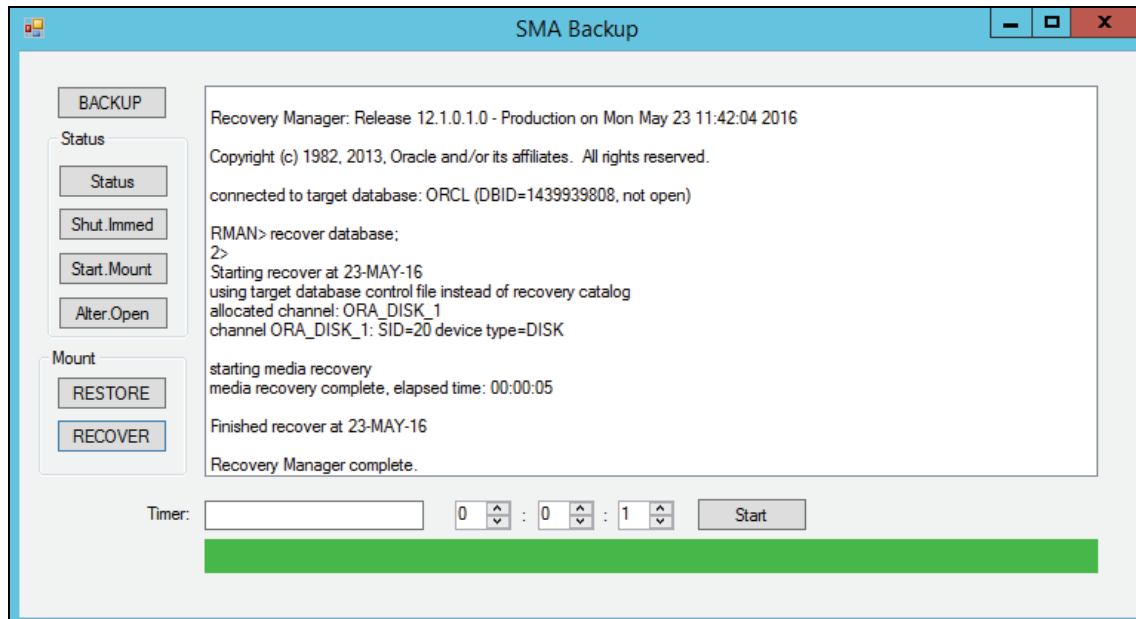


Рисунок 5.11 – Восстановление базы данных

Как видно, восстановление файлов прошло успешно.



## Приложение А (обязательное)

### Образец оформления титульного листа пояснительной записки

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра информационных систем и технологий  
Специальность 1-40 05 01-03 «Информационные системы и техноло-  
гии (издательско-полиграфический комплекс)»  
Специализация «Программирование интернет-приложений»

### ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Выполнил студент \_\_\_\_\_  
(Ф.И.О. студента)

Руководитель проекта \_\_\_\_\_  
(учен. степень, звание, должность, подпись, Ф.И.О. руководителя)

Заведующий кафедрой \_\_\_\_\_  
(учен. степень, звание, должность, подпись, Ф.И.О.)

Консультанты: \_\_\_\_\_  
(учен. степень, звание, должность, подпись, Ф.И.О.)

\_\_\_\_\_  
\_\_\_\_\_

Курсовой проект защищен с оценкой \_\_\_\_\_

Минск 201\_

## Приложение Б (обязательное)

### Образец оформления задания на курсовой проект

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра информационных систем и технологий  
Специальность 1-40 05 01-03 «Информационные системы и техно-  
логии (издательско-полиграфический комплекс)»  
Специализация «Программирование интернет-приложений»

«Утверждаю»  
Заведующий кафедрой

\_\_\_\_\_ подпись \_\_\_\_\_ инициалы и фамилия  
« \_\_\_\_ » \_\_\_\_\_ 201\_\_ г.

### ЗАДАНИЕ на курсовой проект по дисциплине «Администрирование баз данных и приложений» студента

\_\_\_\_\_ инициалы и фамилия студента

1. Тема проекта \_\_\_\_\_
2. Срок сдачи студентом законченного проекта: « \_\_\_\_ » мая 201\_\_ г.
3. Требования к работе:
  - 3.1. Функционально должны быть выполнены следующие бизнес-задачи: \_\_\_\_\_
  - 3.2. Прочие требования:
    - 3.2.1. Доступ к данным должен осуществляться только через соответствующие процедуры.
    - 3.2.2. Должен быть проведен импорт данных из файлов следующего формата \_\_\_\_\_, экспорт данных в формат \_\_\_\_\_.

3.2.3. Необходимо протестировать производительность базы данных (на таблицах, содержащих не менее 100 000 строк) и внести изменения в структуру в случае необходимости.

3.2.4. Необходимо при решении бизнес-задачи применить технологию базы данных согласно выбранной теме.

3.2.5. Листинги скриптов должны содержать комментарии.

4. Содержание расчетно-пояснительной записки:

4.1. Введение.

4.2. Постановка задачи (выбор СУБД, область решения бизнес-задачи, используемая технология, аналоги).

4.3. Установка и настройка сервера СУБД.

4.4. Проектирование инфраструктуры базы данных (пользователи, роли, файлы и пр.).

4.5. Проектирование и реализация необходимых объектов базы данных (таблицы, представления, индексы, хранимые процедуры и функции, триггеры и пр.).

4.6. Описание применения технологии в созданной базе данных.

4.7. Импорт и генерация тестовых данных.

4.8. Резервное копирование и восстановление данных.

4.9. Демонстрационное приложение.

4.10. Заключение.

4.11. Список использованных источников.

4.12. Приложения.

5. Форма представления выполненного курсового проекта:

5.1. Теоретическая часть курсового проекта должна быть представлена в формате MS Word согласно \_\_\_\_\_.

5.2. Необходимые схемы, диаграммы и рисунки допускается делать в MS Office Visio или PrintScreen.

5.3. Листинги скриптов представляются в приложении.

5.4. К записке необходимо приложить CD (DVD), который должен содержать: пояснительную записку, SQL-скрипты, файлы базы данных и приложение.

6. Дата выдачи задания «\_\_\_» февраля 201\_\_ г.

Руководитель \_\_\_\_\_  
подпись инициалы и фамилия преподавателя

Задание принял к исполнению \_\_\_\_\_  
подпись инициалы и фамилия студента

### Календарный план

№ п/п	Наименование этапов курсового проекта	Срок выполнения этапов
1	Аналитический обзор литературы по теме проекта	
2	Проектирование инфраструктуры и модели базы данных	
3	Проектирование и реализация объектов базы данных	
4	Применение технологии в базе данных	
5	Импорт и экспорт данных, тестирование производительности	
6	Резервное копирование и восстановление данных базы данных	
7	Разработка демонстрационного приложения	
8	Оформление пояснительной записки	

## Литература

- 1 Грюнвальд, Р. Oracle. Справочник / Р. Грюнвальд, Д. Крейнс. – СПб.: Символ-Плюс, 2005. – 976 с.
- 2 Каучмен, Д. Oracle Certified Professional. Подготовка администратора баз данных / Д. Каучмен, У. Швинн. – М.: Лори, 2009. – 868 с.
- 3 Кайт, Т. Oracle для профессионалов. В 2 кн. Кн. 1. Архитектура и основные особенности / Т. Кайт. – СПб.: ДиаСофтЮП, 2005. – 656 с.
- 4 Кайт, Т. Oracle для профессионалов. В 2 кн. Кн. 2. Расширение возможностей и защита / Т. Кайт. – СПб.: ДиаСофтЮП, 2005. – 816 с.
- 5 Генник, Дж. Oracle SQL\*Plus. Карманный справочник / Дж. Генник. – СПб.: Питер, 2004. – 188 с.
- 6 Oracle [Электронный ресурс] / Oracle Corporation. – Redwood Shores, 2012. – Режим доступа: <http://www.oracle.com>. – Дата доступа: 18.09.2017.
- 7 Oracle DBA Forum [Электронный ресурс] / Jelsoft Enterprises Ltd. – М., 2012. – Режим доступа: <http://odba.ru>. – Дата доступа: 18.09.2017.

Учебное издание

**Блинова** Евгения Александровна  
**Мороз** Леонарда Станиславовна

**АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ  
И ПРИЛОЖЕНИЙ**

Учебно-методическое пособие

Редактор *Е. С. Ватеичкина*  
Компьютерная верстка *Е. С. Ватеичкина*  
Корректор *Е. С. Ватеичкина*

Издатель:

УО «Белорусский государственный технологический университет».

Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий

№ 1/127 от 20.03.2014.

Ул. Свердлова, 13а, 220006, г. Минск.