

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Кемеровский государственный университет»  
Институт цифры  
Кафедра цифровых технологий

## ОТЧЕТ

о прохождении Производственной практики.  
Технологическая (проектно-технологическая) практика

выполнил обучающийся

Корбин Егор Константинович

Курс 4

Группа МОА-195

Форма обучения очная

направление подготовки 02.03.03 Математическое обеспечение и  
администрирование информационных систем.  
направленность (профиль) подготовки «Информационные системы и базы  
данных».

Руководитель практики:  
канд. техн. наук, доцент,  
доцент кафедры цифровых технологий,  
Завозкин С.Ю.



Работа защищена:

“ 22 ” 02 2023 г.

с оценкой хорошо

## КАЛЕНДАРНЫЙ ПЛАН

№ п/п	Наименования этапов (разделов) практики	Календарные сроки (даты выполнения)
1	Небольшая лекция об архитектуре БД Oracle	01.02.2023
2	Установка и настройка БД. Установка SQL Developer.	02.02.2023 - 06.02.2023
3	Лекция по мультиконтейнерной архитектуре	06.02.2023
4	Создать пользователя с правами, достаточными для подключения к БД (на уровне CDB). Создать пользователя, обладающего достаточными правами (без переизбытка) для обращений к представлению v\$database (на уровне CDB). Создать суперпользователя в CDB. Подготовить запрос, с помощью которого можно увидеть созданных пользователей.	06.02.2023 - 07.02.2023
5	Создать подключаемую БД, сделать её доступной для чтения-записи.	07.02.2023
6	Создать суперпользователя в PDB. Создать произвольную таблицу суперпользователем в PDB. Создать другого пользователя в PDB, выдать необходимые и достаточные права для удаления таблицы, созданной первым пользователем.	07.02.2023 - 08.02.2023
7	Необходимый шаг для обеспечения полноценного восстановления БД	08.02.2023
8	Установите схему HR - схема с демонстрационными данными (в ранее созданной PDB). Создать пользователя, обладающего правами на чтение таблиц схемы HR. Если выбран вариант с проектированием и созданием собственных таблиц (в своей схеме), необходимо также создать другого пользователя, обладающего правами на чтение созданных таблиц.	08.02.2023 - 10.02.2023
9	Настроить резервное копирование с учётом того, что при сбое в бд будет потеряно данных не больше, чем за последние 6 часов.	13.02.2023
10	Создание джобов - создать таблицу с двумя полями - целочисленное и дата. Создать джоб, который будет	13.02.2023 - 14.02.2023

	выполнять вставку в данную таблицу целого числа, а также текущую дату и время. При каждой вставке целое число должно возрастать на 1. Выполнение данного джоба должно осуществляться один раз в час. Задание выполняется в рамках созданной PDB.	
11	Выполнить удаление объектов схемы HR. Выполнить восстановление из существующих бэкапов.	15.02.2023 - 16.02.2023
12	Выступление с презентацией, в которой будет описано, чем Вы занимались, чему научились, что понравилось/не понравилось, пожелания.	20.02.2023

## Содержание

<b>ВВЕДЕНИЕ .....</b>	<b>5</b>
<b>ОСНОВНАЯ ЧАСТЬ .....</b>	<b>7</b>
<b>Общая характеристика компании ООО “Софт Инжиниринг” .....</b>	<b>7</b>
<b>Выполнение задач, предусмотренных планом .....</b>	<b>9</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>26</b>
<b>СПИСОК ЛИТЕРАТУРЫ .....</b>	<b>27</b>

## ВВЕДЕНИЕ

Современные базы данных – это сложные многофункциональные программные системы, работающие в открытой распределенной среде. Они уже сегодня доступны для использования в деловой сфере и выступают не просто в качестве технических и научных решений, но как завершенные продукты, предоставляющие разработчикам мощные средства управления данными и богатый инструментарий для создания прикладных программ и систем.

Администрирование предусматривает осуществление функций, направленных на обеспечение надежного и эффективного функционирования системы баз данных, адекватности содержания базы данных информационным потребностям пользователей, отображения в базе данных актуального состояния предметной области, кроме того система баз данных может иметь много пользователей. Исходя из этого, есть необходимость в персонале, обеспечивающего администрирование базы данных.

Такая потребность в квалифицированных кадрах является следствием централизованного характера управления данными в таких системах, постоянно требующего поиска компромисса между противоречивыми требованиями к системе в социальной пользовательской среде. Хотя такая необходимость и признавалась на ранних стадиях развития технологии баз данных, четкое понимание и структуризация функций персонала, занятого администрированием, сложилось только вместе с признанием многоуровневой архитектуры СУБД (система управления базами данных). СУБД — это комплекс языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями. Она обеспечивает хранение, защиту данных и взаимодействие пользователя с БД, позволяя пользователям производить поиск, сортировку и выборку информации в базе данных, а некоторым пользователям — добавлять, удалять и изменять записи в БД. Лидирующие позиции среди СУБД занимает – Oracle Database, объектно-реляционная СУБД, созданная компанией Oracle.

Актуальностью является то, что базы данных составляют в настоящее время основу компьютерного обеспечения информационных процессов, входящих практически во все сферы деятельности, следовательно, есть необходимость в персонале, обеспечивающим управление данными в системе БД. Действительно, процессы обработки информации имеют общую природу и полагаются на описание фрагментов реальности, выраженное в виде совокупности взаимосвязанных данных. Базы данных являются эффективным средством представления структур данных и манипулирования ими. Концепция баз данных предполагает использование интегрированных средств хранения информации, позволяющих обеспечить централизованное управление данными и обслуживание ими многих пользователей. При этом БД должна поддерживаться в среде электронно-вычислительная машина (ЭВМ) единым программным обеспечением, называемым системой управления базами данных. СУБД вместе с прикладными программами называют банком данных.

Объект исследования – СУБД Oracle Database.

Предмет исследования – использование СУБД Oracle Database в деятельности IT-компаний.

Целью практики является наработка практических навыков в СУБД Oracle и формирование опыта трудовой деятельности в компании ООО “Софт Инжиниринг” в отделе баз данных.

Для достижения поставленной цели были определены следующие задачи:

- ознакомление с компанией,
- прослушивание лекций,
- изучение принципа работы архитектуры Oracle Multitenant,
- выполнение задач, выданных руководителем профильной организации,
- изучение теоретического материала для лучшего выполнения задач.

Производственная практика проходила в компании ООО “Софт Инжиниринг” в отделе баз данных.

## ОСНОВНАЯ ЧАСТЬ

### Общая характеристика компании ООО “Софт Инжиниринг”

Компания является крупнейшим IT-разработчиком в регионе. Одна из приоритетных задач компании – удовлетворение бизнес-потребностей клиента. Штат сотрудников состоит из 140 человек и продолжает увеличиваться. Причём штат сотрудников не сосредоточен конкретно в Кемеровской области, в компании трудятся люди и из других регионов.

Юридический адрес: 650002, Кемеровская область, город Кемерово, Бакинский переулок, строение 15, помещение 1.

Основные виды деятельности приведены в таблице 1.

Таблица 1 – Основные виды деятельности компании

Название трудовой деятельности	Вид профессиональной деятельности
Разработка программного обеспечения	Корпоративные приложения
	1С разработка
	Веб-разработка (бэкенд и фронтенд)
	Мобильная разработка на iOS и Android
Аналитика и разработка технического задания	Выявление требований к программному продукту
	Составление спецификаций на разработку программного продукта
	Макетирование и прототипирование интерфейсов
Аудит IT-инфраструктуры предприятия	Исследование и анализ
	Составление экспертной оценки состояния
	Формирование рекомендаций по оптимизации

За время своего десятилетнего развития компания создала 78 реализованных продуктов, среди которых:

- цифровая платформа «Кузбасс Онлайн»,
- мобильное приложение «Запись к врачу»,
- Система мультипостинга «Запости»,
- Новостной вестник A42.RU.

### **1) Цифровая платформа «Кузбасс Онлайн»**

Интерактивная платформа для создания комфортной среды в городах Кузбасса за счет открытого диалога горожан, органов местного самоуправления и организаций, обслуживающих город. При помощи сайта и мобильного приложения «Кузбасс Онлайн» каждый житель может принимать активное участие в развитии родного региона.

### **2) Мобильное приложение «Запись к врачу»**

Мобильное приложение позволяет записаться на прием к врачам в государственные и частные медицинские учреждения без ожидания ответа администратора в формате 24/7.

### **3) Система мультипостинга «Запости»**

Веб-сайт автоматизирует отложенную публикацию и позволяет консолидировано публиковать записи в социальные сети (ВКонтакте, Одноклассники, Телеграм-каналы).

### **4) Новостной вестник A42.RU**

Региональный сайт Кузбасса, повествует читателям обо всём, что происходит в регионе и за его пределами. Публикуются свежие и проверенные новости, развлекательные и аналитические материалы.

В трудовой деятельности особое внимание уделяется индивидуальному развитию сотрудников. Компания дважды в год проводит праздник, в котором оценивается профессиональный рост сотрудника. Данное событие влияет на поднятие заработной платы. Кроме профессионального развития, компания способствует развитию личных качеств сотрудников. Желание прийти на помощь, порядочность, взаимное доверие и поддержка – всё это создаёт ощущение полной вовлеченности в рабочий процесс.



## Выполнение задач, предусмотренных планом

1) Создать пользователя с правами, достаточными для подключения к БД (на уровне CDB).

```
CREATE USER C##EAGLE IDENTIFIED BY ORCL;  
GRANT CREATE SESSION TO C##EAGLE;
```

2) Создать пользователя, обладающего достаточными правами (без переизбытка) для обращений к представлению v\$database (на уровне CDB).

```
CREATE USER C##WHITE_EAGLE IDENTIFIED BY ORCL;  
GRANT CREATE SESSION TO C##WHITE_EAGLE;  
GRANT SELECT ON V_$DATABASE TO C##WHITE_EAGLE;
```

3) Создать суперпользователя в CDB.

```
CREATE USER C##SUPER_EAGLE IDENTIFIED BY ORCL;  
GRANT CREATE SESSION TO C##SUPER_EAGLE;  
GRANT DBA TO C##SUPER_EAGLE;
```

4) Подготовить запрос, с помощью которого можно увидеть созданных пользователей.

Запрос и его результат изображены в соответствии с рисунками 1 и 2.

```
SQL> select username, created  
2  from dba_users  
3  where created between to_date('2023/02/01', 'yyyy/mm/dd') and to_date('2023/02/28', 'yyyy/mm/dd')  
4  order by username;  
  
USERNAME  
-----  
CREATED  
-----  
C##EAGLE  
07-FEB-23  
  
C##SUPER_EAGLE  
07-FEB-23  
  
C##WHITE_EAGLE  
07-FEB-23  
  
SQL> set linesize 250  
SQL> _
```

Рис. 1 – Запрос с использованием дат

```
SQL> select username, oracle_maintained
2  from dba_users
3  where oracle_maintained like '%N%'
4  order by username;

USERNAME
-----
C##EAGLE
C##SUPER_EAGLE
C##WHITE_EAGLE
SQL>
```

*Рис. 2 – Запрос с использованием оператора WHERE*

5) Создать подключаемую БД, сделать её доступной для чтения-записи.

Создание подключаемой БД и перевод в режим чтения-записи изображено в соответствии с рисунком 3.

```
SQL> show user
USER is "SYS"
SQL>
SQL> create pluggable database STDB1 admin user STDB1DBA identified by oracle
2  default tablespace USERS
3  datafile 'C:\APP\STUDENT\PRODUCT\21C\ORADATA\XE\STDB1\STDB1_users01.dbf' size 250m autoextend on
4  file_name_convert=('C:\APP\STUDENT\PRODUCT\21C\ORADATA\XE\pdbseed', 'C:\APP\STUDENT\PRODUCT\21C\ORADATA\XE\STDB1');

Pluggable database created.

SQL> show pdbs

  CON_ID CON_NAME              OPEN MODE RESTRICTED
  -----
2 PDB$SEED                   READ ONLY NO
3 XEPDB1                     READ WRITE NO
4 STDB1                      MOUNTED

SQL> alter pluggable database STDB1 open;

Pluggable database altered.

SQL> show pdbs

  CON_ID CON_NAME              OPEN MODE RESTRICTED
  -----
2 PDB$SEED                   READ ONLY NO
3 XEPDB1                     READ WRITE NO
4 STDB1                      READ WRITE NO

SQL>
```

*Рис. 3 – Создание PDB с режимом чтения-записи*

6) Создать суперпользователя в PDB.

```
ALTER SESSION SET CONTAINER = STDB1;
CREATE USER PDB_EAGLE IDENTIFIED BY ORACLE;
GRANT CREATE SESSION TO PDB_EAGLE;
GRANT DBA TO PDB_EAGLE;
```

Проверка пользователя отображена на рисунке 4.

```
SQL> connect PDB_Eagle/oracle@localhost/STDB1;
Connected.
SQL> show user
USER is "PDB_EAGLE"
SQL> show con_name;

CON_NAME
-----
STDB1
SQL>
```

*Рис. 4 – Проверка пользователя*

7) Создать произвольную таблицу суперпользователем в PDB.

Создание таблицы показано в соответствии с рисунком 5.

```
SQL> show user
USER is "PDB_EAGLE"
SQL> show con_name

CON_NAME
-----
STDB1
SQL> CREATE TABLE customers
  2  ( customer_id number(10) NOT NULL,
  3  customer_name varchar2(50) NOT NULL,
  4  city varchar2(50));

Table created.
```

*Рис. 5 – Создание таблицы*

8) Создать другого пользователя в PDB, выдать необходимые и достаточные права для удаления таблицы, созданной первым пользователем.

```
ALTER SESSION SET CONTAINER = STDB1;
CONNECT PDB_EAGLE/ORACLE@LOCALHOST/STDB1;
CREATE USER USER_DROP IDENTIFIED BY ORCL;
GRANT CREATE SESSION TO USER_DROP;
GRANT DROP ANY TABLE TO USER_DROP;
```

Удаление таблицы показано в соответствии с рисунком 6.

```
SQL> connect user_drop/orcl@localhost/STDB1;
Connected.
SQL> show user
USER is "USER_DROP"
SQL> drop table PDB_EAGLE.CUSTOMERS;

Table dropped.

SQL> _
```

*Рис. 6 – Удаление таблицы*

9) Включение архивных логов.

Сделаем проверку архивных логов в соответствии с рисунком 7.

```

SQL> show user
USER is "SYS"
SQL> show con_name

CON_NAME
-----
CDB$ROOT
SQL> SELECT LOG_MODE FROM SYS.V$DATABASE;

LOG_MODE
-----
NOARCHIVELOG

SQL> ARCHIVE LOG LIST;
Database log mode                No Archive Mode
Automatic archival                Disabled
Archive destination              C:\app\student\product\21c\homes\OraDB21Home2\RDBMS
Oldest online log sequence       5
Current log sequence             7

```

*Рис. 7 – Проверка активации*

Включение архивных логов произведено в соответствии с рисунками 8 и 9.

```

SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='location=C:\app\student\product\21c\homes\OraDB21Home2\RDBMS' SCOPE=spfile;
System altered.

SQL> ALTER SYSTEM SET LOG_ARCHIVE_FORMAT='%t_%s_%r.arc' SCOPE=spfile;
System altered.

SQL> SHUTDOWN IMMEDIATE;
Database closed.
Database dismounted.
ORACLE instance shut down.

```

*Рис. 8 – Включение логов (1)*

```

SQL> STARTUP MOUNT;
ORACLE instance started.

Total System Global Area 1291844832 bytes
Fixed Size                 9854176 bytes
Variable Size             603979776 bytes
Database Buffers          671088640 bytes
Redo Buffers              6922240 bytes
Database mounted.
SQL> ALTER DATABASE ARCHIVELOG;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

SQL> SELECT LOG_MODE FROM SYS.V$DATABASE;

LOG_MODE
-----
ARCHIVELOG

SQL> ARCHIVE LOG LIST;
Database log mode                Archive Mode
Automatic archival                Enabled
Archive destination              C:\app\student\product\21c\homes\OraDB21Home2\RDBMS
Oldest online log sequence       5
Next log sequence to archive    7
Current log sequence            7
SQL>

```

*Рис. 9 – Включение логов (2)*

10) Установите схему HR - схема с демонстрационными данными (в ранее созданной PDB). Создать пользователя, обладающего правами на чтение таблиц схемы HR. Если выбран вариант с проектированием и созданием собственных таблиц (в своей схеме), необходимо также создать другого пользователя, обладающего правами на чтение созданных таблиц.

Выбран вариант с проектированием и созданием собственных таблиц. Спроектированная схема показана на рисунке 10.

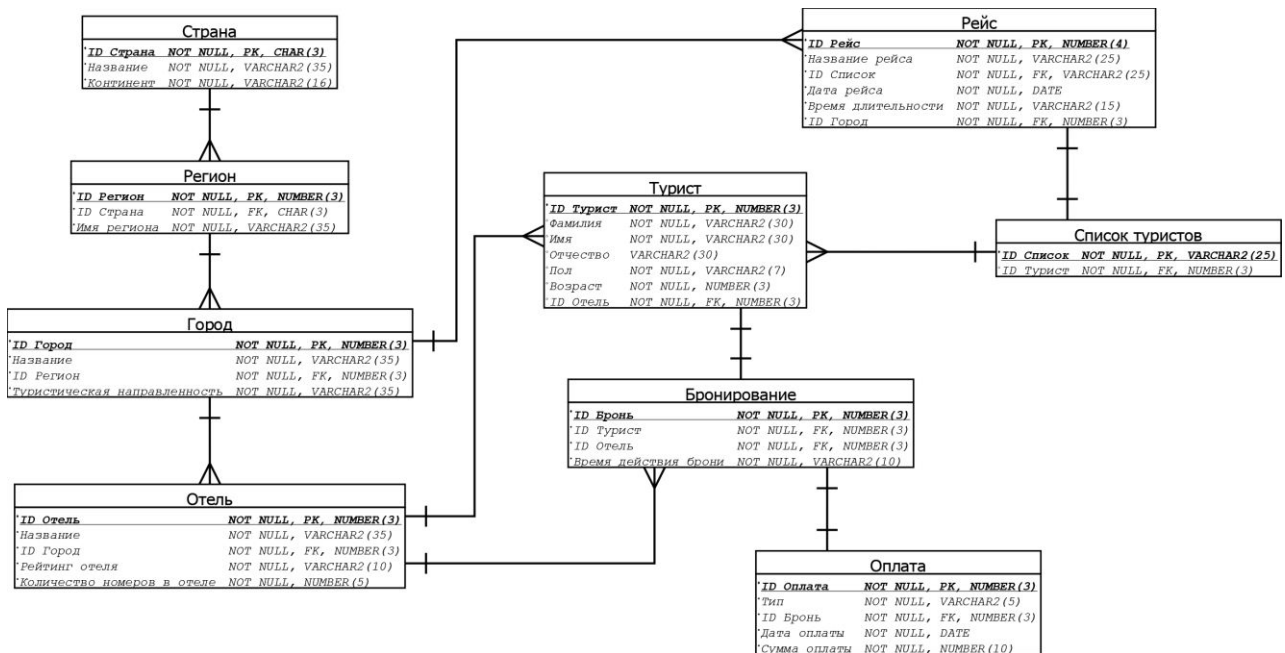


Рис. 10 – Физическая модель данных

Скрипты генерации таблиц.

```

CREATE TABLE COUNTRY_T (
  ID_COUNTRY CHAR(3) CONSTRAINT COUNTRY_T_ID_PK PRIMARY KEY,
  NAME_COUNTRY VARCHAR2(35) NOT NULL,
  CONTINENT VARCHAR2(16) NOT NULL);
    
```

```

CREATE TABLE REGION_T (
  ID_REGION NUMBER(3) CONSTRAINT REGION_T_ID_PK PRIMARY KEY,
  COUNTRY_ID CHAR(3)
  CONSTRAINT REGION_T_COUNTRY_ID_NN NOT NULL
  CONSTRAINT REGION_T_COUNTRY_ID_FK REFERENCES
  COUNTRY_T(ID_COUNTRY),
  NAME_REGION VARCHAR2(35) NOT NULL);
    
```

```

CREATE TABLE CITY_T (
  ID_CITY NUMBER(3) CONSTRAINT CITY_T_ID_PK PRIMARY KEY,
  NAME_CITY VARCHAR2(35) NOT NULL,
  REGION_ID NUMBER(3)
    
```

CONSTRAINT CITY_T_REGION_ID_NN NOT NULL CONSTRAINT CITY_T_REGION_ID_FK REGION_T(ID_REGION), TRAVEL_FOCUS VARCHAR2(35) NOT NULL);	REFERENCES
---	------------

```

CREATE TABLE LIST_OF_TOURISTS_T (
ID_LIST VARCHAR2(25) CONSTRAINT LIST_T_ID_PK PRIMARY KEY,
TOURIST_ID NUMBER(3)
CONSTRAINT LIST_T_TOURIST_ID_NN NOT NULL
CONSTRAINT LIST_T_TOURIST_ID_FK REFERENCES
TOURIST_T(ID_TOURIST));

```

```

CREATE TABLE VOYAGE_T (
ID_VOYAGE NUMBER(4) CONSTRAINT VOYAGE_T_ID_PK PRIMARY KEY,
NAME_VOYAGE VARCHAR2(25) NOT NULL,
LIST_ID VARCHAR2(25)
CONSTRAINT VOYAGE_T_LIST_ID_NN NOT NULL
CONSTRAINT VOYAGE_T_LIST_ID_FK REFERENCES
LIST_OF_TOURISTS_T(ID_LIST),
VOYAGE_DATE DATE NOT NULL,
DURATION_TIME VARCHAR2(15) NOT NULL,
CITY_ID NUMBER(3)
CONSTRAINT VOYAGE_T_CITY_ID_NN NOT NULL
CONSTRAINT VOYAGE_T_CITY_ID_FK REFERENCES CITY_T(ID_CITY));

```

Скрипты проверки создания ограничений.

```

set linesize 250
COLUMN table_name FORMAT A25
COLUMN column_name FORMAT A25
COLUMN constraint_name FORMAT A25
COLUMN constraint_type FORMAT A25

```

```

SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'COUNTRY_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;

```

```

SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'REGION_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;

```

```

SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'CITY_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;

```

```
SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'HOTEL_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;
```

```
SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'TOURIST_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;
```

```
SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'BOOKING_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;
```

```
SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'PAYMENT_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;
```

```
SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'LIST_OF_TOURISTS_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;
```

```
SELECT      cols.table_name,      cols.column_name,      cons.constraint_name,
cons.constraint_type
FROM all_constraints cons, all_cons_columns cols
WHERE cols.table_name = 'VOYAGE_T'
AND cons.constraint_name = cols.constraint_name
AND cons.owner = cols.owner;
```



Некоторые скрипты генерации данных в таблицах отображены в соответствии с рисунком 11.

```
INSERT INTO PDB_EAGLE.CITY_T
(ID_CITY, NAME_CITY, REGION_ID, TRAVEL_FOCUS)
VALUES
(39, 'Karyes', 9, 'Pilgrimage');

INSERT INTO PDB_EAGLE.CITY_T
(ID_CITY, NAME_CITY, REGION_ID, TRAVEL_FOCUS)
VALUES
(40, 'Beziers', 17, 'Local holidays');

INSERT INTO PDB_EAGLE.CITY_T
(ID_CITY, NAME_CITY, REGION_ID, TRAVEL_FOCUS)
VALUES
(41, 'Kuopio', 14, 'Thermal springs');

INSERT INTO PDB_EAGLE.BOOKING_T
(ID_BOOKING, TOURIST_ID, HOTEL_ID, VALIDITY_PERIOD)
VALUES
(32, 99, 59, '6 HOURS');

INSERT INTO PDB_EAGLE.BOOKING_T
(ID_BOOKING, TOURIST_ID, HOTEL_ID, VALIDITY_PERIOD)
VALUES
(33, 98, 60, '12 HOURS');

INSERT INTO PDB_EAGLE.BOOKING_T
(ID_BOOKING, TOURIST_ID, HOTEL_ID, VALIDITY_PERIOD)
VALUES
(34, 97, 60, '12 HOURS');

INSERT INTO PDB_EAGLE.BOOKING_T
(ID_BOOKING, TOURIST_ID, HOTEL_ID, VALIDITY_PERIOD)
VALUES
(35, 96, 60, '24 HOURS');

INSERT INTO PDB_EAGLE.TOURIST_T
(ID_TOURIST, SURNAME, NAME, PATRONIM, SEX, AGE, HOTEL_ID)
VALUES
(93, 'Levett', 'Vanessa', '', 'Female', 65, 60);

INSERT INTO PDB_EAGLE.TOURIST_T
(ID_TOURIST, SURNAME, NAME, PATRONIM, SEX, AGE, HOTEL_ID)
VALUES
(94, 'Costenko', 'Vyacheslav', 'Danilovich', 'Male', 21, 60);

INSERT INTO PDB_EAGLE.TOURIST_T
(ID_TOURIST, SURNAME, NAME, PATRONIM, SEX, AGE, HOTEL_ID)
VALUES
(95, 'Tusch-Letz', 'Stanislav', '', 'Male', 28, 60);

INSERT INTO PDB_EAGLE.TOURIST_T
(ID_TOURIST, SURNAME, NAME, PATRONIM, SEX, AGE, HOTEL_ID)
VALUES
(96, 'Trubach', 'Nikolai', 'Andreevich', 'Male', 38, 60);
```

*Рис. 11 – Генерация данных*

Создание другого пользователя, обладающего правами на чтение созданных таблиц.

```
create user read_table identified by orcl;
grant create session to read_table;
grant select on COUNTRY_T to read_table;
grant select on REGION_T to read_table;
grant select on CITY_T to read_table;
grant select on HOTEL_T to read_table;
grant select on TOURIST_T to read_table;
grant select on BOOKING_T to read_table;
grant select on PAYMENT_T to read_table;
grant select on LIST_OF_TOURISTS_T to read_table;
grant select on VOYAGE_T to read_table;
```

Проведём проверку в соответствии с рисунком 12.

```
SQL> show user
USER is "READ_TABLE"
SQL> select * from PDB_Eagle.PAYMENT_T;

ID_PAYMENT TYPE      BOOKING_ID PAYMENT_D PAYMENT_AMOUNT
-----
60 Cash          52 17-DEC-15      1000
61 Cash          41 20-MAY-16       500
62 Cash          42 13-JAN-13       750
63 Card          43 30-JUN-14      6300
64 Card          44 23-JUL-16      2800
65 Cash          34 10-MAR-13      3600
66 Card          35 19-SEP-18       410

7 rows selected.

SQL> _
```

*Рис 12. – Проверка таблицы*

11) Настроить резервное копирование с учётом того, что при сбое в базе данных будет потеряно данных не больше, чем за последние 6 часов.

Настройка скриптов RMAN [5].

```
report schema;
show all;
run {
configure device type disk backup type to copy;
configure device type disk backup type to backupset;
backup incremental level 0 database plus archivelog tag "level 0";
backup current controlfile spfile;}
exit;
```

```
report schema;
show all;
run {
backup incremental level 1 database plus archivelog tag "level 1";}
exit;
```

Создание сценариев запуска файлов RMAN показано в соответствии с рисунком 13 [8].

```

1 cd C:\app\MacSquizzzy\product\21c\dbhomeXE\bin\
2 rman TARGET / LOG C:\app\MacSquizzzy\product\21c\oradata\XE\backup\backup_one.log
3 @C:\app\MacSquizzzy\product\21c\oradata\XE\level_one.rman

start_zero.bat start_one.bat level_zero.rman level_one.rman новый 1 новый 2

1 cd C:\app\MacSquizzzy\product\21c\dbhomeXE\bin\
2 rman TARGET / LOG C:\app\MacSquizzzy\product\21c\oradata\XE\backup\backup_zero.log
3 @C:\app\MacSquizzzy\product\21c\oradata\XE\level_zero.rman

```

Рис. 13 – Сценарии запуска

Для копирования с нулевым и первым уровнем создаются следующие программы и задачи [7].

```

BEGIN
  DBMS_SCHEDULER.CREATE_PROGRAM(
    program_name => 'stud_practice_backup_zero',
    program_type => 'executable',
    program_action => 'C:\app\MacSquizzzy\product\21c\oradata\XE\start_zero.bat',
    enabled      => TRUE);
END;
/

```

```

BEGIN
  DBMS_SCHEDULER.SET_ATTRIBUTE('stud_practice_backup_zero', 'detached',
TRUE);
  DBMS_SCHEDULER.CREATE_JOB(
    job_name      => 'stud_backup_zero',
    program_name  => 'stud_practice_backup_zero',
    repeat_interval => 'FREQ=WEEKLY; BYDAY=MON');

  DBMS_SCHEDULER.ENABLE('stud_backup_zero');
END;
/

```

```

BEGIN
  DBMS_SCHEDULER.CREATE_PROGRAM(
    program_name => 'stud_practice_backup_one',
    program_type => 'executable',
    program_action => 'C:\app\MacSquizzzy\product\21c\oradata\XE\start_one.bat',
    enabled      => TRUE);
END;
/

```

```

BEGIN
  DBMS_SCHEDULER.SET_ATTRIBUTE('stud_practice_backup_one', 'detached',
TRUE);
  DBMS_SCHEDULER.CREATE_JOB(
    job_name      => 'stud_backup_one',
    program_name  => 'stud_practice_backup_one',
    repeat_interval    => 'FREQ=HOURLY;          INTERVAL=6;
BYDAY=TUE,WED,THU,FRI,SAT,SUN');

    DBMS_SCHEDULER.ENABLE('stud_backup_one');
END;
/

```

Проведём проверку, используя следующий запрос и получим результат в соответствии с рисунком 14.

```

SELECT          OWNER,          JOB_NAME,          ENABLED,
NEXT_RUN_DATE,RAISE_EVENTS,LAST_START_DATE

FROM dba_scheduler_jobs;

```

OWNER	JOB_NAME	ENABL	NEXT_RUN_DATE	RAISE_EVENTS	LAST_START_DATE
SYS	FGR\$AUTOPURGE_JOB	FALSE			
SYS	LOAD_OPATCH_INVENTORY	FALSE			
SYS	XMLDB_NFS_CLEANUP_JOB	FALSE			
SYS	STUD_BACKUP_ZERO	TRUE	20-FEB-23 04.09.07.321000 PM +07:00		
SYS	STUD_BACKUP_ONE	TRUE	15-FEB-23 04.09.23.075000 PM +07:00		15-FEB-23 04.09.23.106000 PM +07:00
ORACLE_OCM	MGMT_CONFIG_JOB	TRUE	16-FEB-23 01.01.01.277000 AM +07:00		15-FEB-23 03.23.33.265000 AM +07:00
OWNER	JOB_NAME	ENABL	NEXT_RUN_DATE	RAISE_EVENTS	LAST_START_DATE
ORACLE_OCM	MGMT_STATS_CONFIG_JOB	TRUE	01-MAR-23 01.01.01.338000 AM +07:00		01-FEB-23 10.22.33.001000 AM +07:00

23 rows selected.

SQL> \_

*Рис. 14 – Проверка созданных задач*

12) Создание джобов - создать таблицу с двумя полями - целочисленное и дата. Создать джоб, который будет выполнять вставку в данную таблицу целого числа, а также текущую дату и время. При каждой вставке целое число должно возрастать на 1. Выполнение данного джоба должно осуществляться один раз в час. Задание выполняется в рамках созданной PDB.

Заготовил таблицу.

```
CREATE TABLE TABLE_JOB (  
  INCREMENT_T NUMBER(10) CONSTRAINT TABLE_JOB_ID_PK PRIMARY  
  KEY,  
  UPDATE_DATE TIMESTAMP);
```

Заготовил последовательность для первичного ключа.

```
CREATE SEQUENCE DRAGOSTEA  
  START WITH 1  
  INCREMENT BY 1  
  NOCACHE  
  NOCYCLE;
```

Создаём задание для выполнения условия задачи.

```
BEGIN  
  DBMS_SCHEDULER.CREATE_JOB(  
    JOB_NAME => 'test_job_updater',  
    JOB_TYPE => 'PLSQL_BLOCK',  
    JOB_ACTION => 'INSERT INTO TABLE_JOB  
(INCREMENT_T,UPDATE_DATE) VALUES  
(DRAGOSTEA.nextval,CURRENT_TIMESTAMP);',  
    REPEAT_INTERVAL => 'FREQ=HOURLY; INTERVAL=1');  
  
  DBMS_SCHEDULER.ENABLE('test_job_updater');  
END;  
/
```

Работа приведённых выше инструкций показана на рисунке 15 и 16 [6].

```
SQL> alter session set container = STDB1;  
  
Session altered.  
  
SQL> CREATE TABLE TABLE_JOB (  
  2 INCREMENT_T NUMBER(10) CONSTRAINT TABLE_JOB_ID_PK PRIMARY KEY,  
  3 UPDATE_DATE TIMESTAMP);  
  
Table created.  
  
SQL> CREATE SEQUENCE DRAGOSTEA  
  2 START WITH 1  
  3 INCREMENT BY 1  
  4 NOCACHE  
  5 NOCYCLE;  
  
Sequence created.  
  
SQL> BEGIN  
  2 DBMS_SCHEDULER.CREATE_JOB(  
  3 JOB_NAME => 'test_job_updater',  
  4 JOB_TYPE => 'PLSQL_BLOCK',  
  5 JOB_ACTION => 'INSERT INTO TABLE_JOB (INCREMENT_T,UPDATE_DATE) VALUES (DRAGOSTEA.nextval,CURRENT_TIMESTAMP);',  
  6 REPEAT_INTERVAL => 'FREQ=HOURLY; INTERVAL=1');  
  7  
  8 DBMS_SCHEDULER.ENABLE('test_job_updater');  
  9 END;  
 10 /  
  
PL/SQL procedure successfully completed.
```

Рис. 15 – Создание задачи для таблицы

```

SQL> COLUMN NEXT_RUN_DATE FORMAT A20
SQL> SELECT owner, job_name, enabled,NEXT_RUN_DATE,RAISE_EVENTS, LAST_START_DATE
  2  from dba_scheduler_jobs
  3  where JOB_NAME='TEST_JOB_UPDATER';

```

OWNER	JOB_NAME	ENABL	NEXT_RUN_DATE	RAISE_EVENTS	LAST_START_DATE
SYS	TEST_JOB_UPDATER	TRUE	17-FEB-23 11.13.24.0 88000 AM US/MOUNTAIN		17-FEB-23 10.13.24.0 88000 AM US/MOUNTAIN

```

SQL> select * from TABLE_JOB;

```

INCREMENT_T	UPDATE_DATE
1	18-FEB-23 12.13.24.119000 AM

```

SQL>

```

*Рис. 16 – Проверка работоспособности*

13) Выполнить удаление объектов схемы HR. Выполнить восстановление из существующих бэкапов.

Выполнение данного задания продемонстрировано рядом рисунков 17, 18, 19, 20, 21, 22 [9].

```

RMAN> run {
2> configure device type disk backup type to copy;
3> configure device type disk backup type to backupset;
4> backup incremental level 0 database plus archivelog tag "level 0";
5> backup current controlfile spfile;}

```

using target database control file instead of recovery catalog  
old RMAN configuration parameters:  
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO BACKUPSET PARALLELISM 1;  
new RMAN configuration parameters:  
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY PARALLELISM 1;  
new RMAN configuration parameters are successfully stored

old RMAN configuration parameters:  
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY PARALLELISM 1;  
new RMAN configuration parameters:  
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO BACKUPSET PARALLELISM 1;  
new RMAN configuration parameters are successfully stored

Starting backup at 24-FEB-23  
current log archived  
allocated channel: ORA\_DISK\_1  
channel ORA\_DISK\_1: SID=285 device type=DISK  
channel ORA\_DISK\_1: starting archived log backup set  
channel ORA\_DISK\_1: specifying archived log(s) in backup set  
input archived log thread=1 sequence=2 RECID=58 STAMP=1129249295  
input archived log thread=1 sequence=3 RECID=59 STAMP=1129326456  
input archived log thread=1 sequence=4 RECID=60 STAMP=1129326752  
input archived log thread=1 sequence=5 RECID=61 STAMP=1129493536  
input archived log thread=1 sequence=6 RECID=62 STAMP=1129646449  
input archived log thread=1 sequence=7 RECID=63 STAMP=1129669897  
channel ORA\_DISK\_1: starting piece 1 at 24-FEB-23  
channel ORA\_DISK\_1: finished piece 1 at 24-FEB-23  
piece handle=C:\APP\MACSQUIZZY\PRODUCT\21C\DBHOME\XE\DATABASE\841LAP8E\_260\_1\_1 tag=LEVEL 0 comment=NONE  
channel ORA\_DISK\_1: backup set complete, elapsed time: 00:00:25  
channel ORA\_DISK\_1: starting archived log backup set  
channel ORA\_DISK\_1: specifying archived log(s) in backup set  
input archived log thread=1 sequence=1 RECID=57 STAMP=1129158547

*Рис. 17 – Выполнение резервного копирования*

```

SQL> COLUMN TABLE_NAME FORMAT A25
SQL> COLUMN OWNER FORMAT A25
SQL> select table_name, owner from dba_tables
  2  where owner like '%PDB_EAGLE%';

TABLE_NAME                                OWNER
-----
COUNTRY_T                                PDB_EAGLE
REGION_T                                PDB_EAGLE
CITY_T                                  PDB_EAGLE
HOTEL_T                                 PDB_EAGLE
TOURIST_T                               PDB_EAGLE
BOOKING_T                               PDB_EAGLE
VOYAGE_T                                PDB_EAGLE
PAYMENT_T                               PDB_EAGLE
LIST_OF_TOURISTS_T                     PDB_EAGLE
CUSTOMERS                              PDB_EAGLE
TEST                                    PDB_EAGLE

11 rows selected.

SQL> commit;

Commit complete.

SQL> select current_scn from v$database;
select current_scn from v$database
*
ERROR at line 1:
ORA-00903: invalid table name

SQL> select current_scn from v$database;

CURRENT_SCN
-----
16039923

SQL>

```

Рис. 18 – Получение текущего скана базы данных

```

SQL> drop table PDB_EAGLE.COUNTRY_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.REGION_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.CITY_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.HOTEL_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.TOURIST_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.BOOKING_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.VOYAGE_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.PAYMENT_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.LIST_OF_TOURISTS_T cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.CUSTOMERS cascade constraints;
Table dropped.

SQL> drop table PDB_EAGLE.TEST cascade constraints;
Table dropped.

SQL> select table_name, owner from dba_tables
  2  where owner like '%PDB_EAGLE%';

no rows selected

SQL> conn / as sysdba
Connected.
SQL> alter system switch logfile;

System altered.

SQL>

```

Рис. 19 – Удаление схемы

```

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1610608752 bytes
Fixed Size                  9855088 bytes
Variable Size               973078528 bytes
Database Buffers            620756992 bytes
Redo Buffers                 6918144 bytes
Database mounted.
SQL> select status from v$instance;

STATUS
-----
MOUNTED

SQL>

```

*Рис. 20 – Первый шаг восстановления*

```

RMAN> run {
2> set until scn = 16039923;
3> restore database;
4> recover database;}

executing command: SET until clause

Starting restore at 24-FEB-23
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=26 device type=DISK
channel ORA_DISK_1: restoring datafile 00016 to C:\APP\MACSQUIZZY\
channel ORA_DISK_1: restoring datafile 00016 to C:\APP\MACSQUIZZY\
channel ORA_DISK_1: reading from backup piece C:\APP\MACSQUIZZY\P
channel ORA_DISK_1: piece handle=C:\APP\MACSQUIZZY\PRODUCT\21C\DB
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:25
Finished restore at 24-FEB-23

Starting recover at 24-FEB-23
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:05

Finished recover at 24-FEB-23

RMAN> _

```

*Рис. 21 – Второй шаг восстановления*



```

SQL> alter database open resetlogs;

Database altered.

SQL> show pdbs


```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	XEPDB1	READ WRITE	NO
4	STDB1	MOUNTED	

```

SQL> alter pluggable database STDB1 open;

Pluggable database altered.

SQL> alter session set container = STDB1;

Session altered.

SQL> select table_name, owner from dba_tables
       2 where owner like '%PDB_EAGLE%';


```

TABLE_NAME	OWNER
COUNTRY_T	PDB_EAGLE
REGION_T	PDB_EAGLE
CITY_T	PDB_EAGLE
HOTEL_T	PDB_EAGLE
TOURIST_T	PDB_EAGLE
BOOKING_T	PDB_EAGLE
VOYAGE_T	PDB_EAGLE
PAYMENT_T	PDB_EAGLE
LIST_OF_TOURISTS_T	PDB_EAGLE
CUSTOMERS	PDB_EAGLE
TEST	PDB_EAGLE

```

11 rows selected.

SQL>

```

Рис. 22 – Проверка процесса восстановления

## **ЗАКЛЮЧЕНИЕ**

В результате прохождения практики цель была достигнута, все задачи решены в полном объёме, профессиональные компетенции приобретены. На основании пройденной практики можно сделать следующие выводы.

Администрирование базами данных предусматривает выполнение функций, направленных на обеспечение надежного и эффективного функционирования системы баз данных, адекватности содержания базы данных информационным потребностям пользователей, отображения в базе данных актуального состояния предметной области.

Администратор БД отвечает за целостность информационных ресурсов компании. На нем лежит ответственность по созданию, обновлению и сохранности связанных между собой резервных копий файлов, исходя из задач компании. Этот человек должен в мельчайших подробностях знать существующие механизмы восстановления программного обеспечения БД, обязан учитывать текущие и перспективные информационные требования предметной области. Координация действий по сбору сведений, проектированию и эксплуатации базы данных, а также по обеспечению защиты данных – неотъемлемая часть компетенций администратора БД.

Правильная реализация функций администрирования базы данных существенно улучшает контроль и управление ресурсами данных предметной области. С этой точки зрения функции администратора БД являются больше управляющими, нежели техническими. Принципы работы и функции определяются подходом к данным как к ресурсам организации, поэтому решение проблем, связанных с администрированием начинается с установления общих принципов эксплуатации СУБД.

Знания, умения, навыки, полученные за период практики, явились достаточным стимулом для активной работы в освоении будущей специальности, позволили практически реализовать теоретически изученные моменты, получить первый профессиональный опыт работы и сформировать

общее представление о специфике деятельности компании ООО “Софт Инжиниринг”.

## СПИСОК ЛИТЕРАТУРЫ

1. Urbano R., Ashdown L. Oracle® Multitenant Administrator's Guide [Electronic resource]. – URL: [https://docs.oracle.com/cd/F19136\\_01/multi/multitenant-administrators-guide.pdf](https://docs.oracle.com/cd/F19136_01/multi/multitenant-administrators-guide.pdf) (date of treatment: 24.02.2023)
2. How to grant access to v\$ views (v\$session, v\$instance): сайт. – URL: <https://techgoeasy.com/how-to-grant-access-to-v-views/> (дата обращения: 24.02.2023). – Текст: электронный.
3. How To Enable/Disable Archive Log Mode In Oracle Database: сайт. – URL: <https://dbaclass.com/article/how-to-enabledisable-archive-log-mode-in-oracle-database/> (дата обращения: 24.02.2023). – Текст: электронный.
4. Database Backup and Recovery User's Guide: сайт. – URL: <https://docs.oracle.com/database/121/BRADV/rcmconfb.htm#BRADV8002> (дата обращения: 24.02.2023). – Текст: электронный.
5. RMAN configuration and backup: сайт. – URL: <https://www.cerebrosql.com/post/rman-configuration-and-backup> (дата обращения: 24.02.2023). – Текст: электронный.
6. Oracle: Scheduler Jobs in Oracle Database: сайт. – URL: <https://www.ktexperts.com/oracle-scheduler-jobs-in-oracle-database/> (дата обращения: 24.02.2023). – Текст: электронный.
7. Использование планировщика Oracle Scheduler: сайт. – URL: <https://oracle-patches.com/oracle/begin/использование-планировщика-oracle-scheduler> (дата обращения: 24.02.2023). – Текст: электронный.
8. Oracle: Scheduling Rman backup jobs through DBMS\_SCHEDULER.: сайт. – URL: [https://www.ktexperts.com/oracle-scheduling-rman-backup-jobs-through-dbms\\_scheduler/](https://www.ktexperts.com/oracle-scheduling-rman-backup-jobs-through-dbms_scheduler/) (дата обращения: 24.02.2023). – Текст: электронный.

9. Oracle Database - Инкарнации базы данных: сайт. – URL: <https://oracle-dba.ru/database/backup-and-restore/rman/rman-incarnations-sample/> (дата обращения: 24.02.2023). – Текст: электронный.