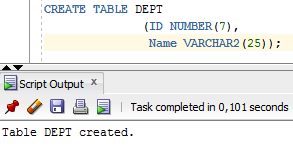
# Практическое занятие 9

* Создание новых таблиц
* Создание новой таблицы с помощью синтаксиса CREATETABLEAS
* Проверка существования таблицы
* Удаление таблиц

1. Создайте таблицу DEPT на основе приведенного ниже описания. Сохраните команду в файле lab9\_1.sql, а затем выполните файл для создания таблицы. Убедитесь в том, что таблица создана.

|  |  |  |
| --- | --- | --- |
| Имя столбца | ID | Name |
| Тип ключа |  |  |
| Nulls/Unique |  |  |
| Таблица FK |  |  |
| Столбец FK |  |  |
| Тип данных | NUMBER | VARCHAR2 |
| Длина | 7 | 25 |





CREATE TABLE DEPT

(ID NUMBER(7),

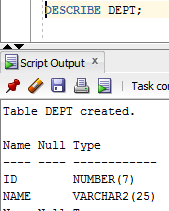
Name VARCHAR2(25));

CREATE TABLE DEPT

(ID NUMBER(7)

CONSTRAINT MYDEPT\_ID\_PK PRIMARY KEY,

Name VARCHAR2(25));



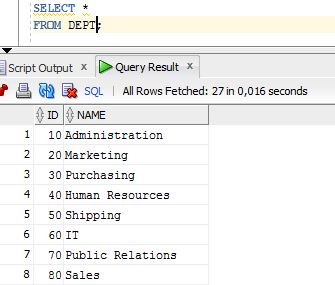
DESCRIBE DEPT;

1. Заполните таблицу DEPTданными из таблицы Departments. Включите только нужные столбцы.

INSERT INTO DEPT

SELECT DEPARTMENT\_ID, DEPARTMENT\_NAME

FROM DEPARTMENTS;



1. Создайте таблицу EMP на основе приведенного ниже описания. Сохраните команду в скрипт-файлеlab9\_3.sql, а затем выполните его, чтобы создать таблицу. Убедитесь в том, что таблица создана.

CREATE TABLE EMP

(ID NUMBER(7)

CONSTRAINT EMP\_ID\_PK PRIMARY KEY,

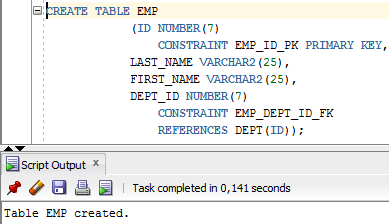
LAST\_NAME VARCHAR2(25),

FIRST\_NAME VARCHAR2(25),

DEPT\_ID NUMBER(7)

CONSTRAINT EMP\_DEPT\_ID\_FK

REFERENCES DEPT(ID));



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя столбца | ID | Last\_name | FIRST\_NAME | DEPT\_ID |
| Тип ключа |  |  |  |  |
| Nulls/Unique |  |  |  |  |
| Таблица FK |  |  |  |  |
| Столбец FK |  |  |  |  |
| Тип данных | NUMBER | VARCHAR2 | VARCHAR2 | NUMBER |
| Длина | 7 | 25 | 25 | 7 |



1. Создайте таблицу EMPLOYEES2, основанную на структуре таблицы EMPLOYEES, включив только столбцы EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARYи DEPARTMENT\_ID. Имена колонок в новой таблице должны быть ID, FIRST\_NAME, LAST\_NAME, SALARYи DEPT\_IDсоответственно.

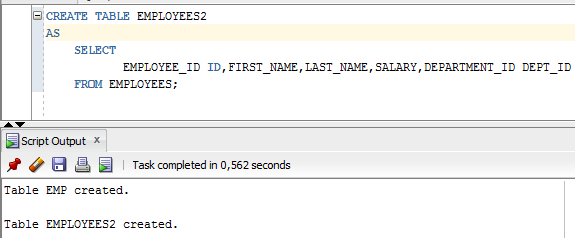
CREATE TABLE EMPLOYEES2

AS

SELECT

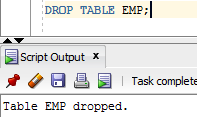
EMPLOYEE\_ID ID,FIRST\_NAME,LAST\_NAME,SALARY,DEPARTMENT\_ID DEPT\_ID

FROM EMPLOYEES;



1. Удалите таблицу EMP.

DROP TABLE EMP;



# Практическое занятие 10, Часть 1

* + Создание простого представления
  + Создание сложного представления
  + Создание представления с ограничением CHECK
  + Попытки изменить данные в представлении
  + Вывод определений представлений
  + Удаление представлений

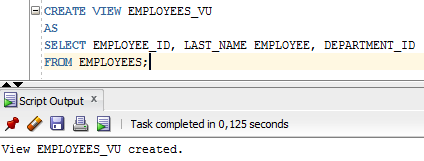
1. Создайте представление EMPLOYEES\_VU. Включите номер служащего, фамилию служащего и номер отдела из таблицы EMPLOYEES. Смените заголовок столбца с фамилией служащего на EMPLOYEE.

CREATE VIEW EMPLOYEES\_VU

AS

SELECT EMPLOYEE\_ID, LAST\_NAME EMPLOYEE, DEPARTMENT\_ID

FROM EMPLOYEES;

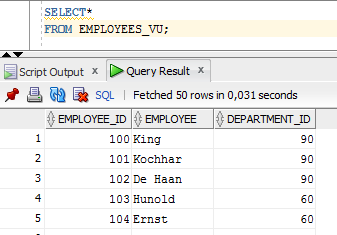


1. Выведите содержимое представления EMPLOYEE\_VU.



SELECT\*

FROM EMPLOYEES\_VU;



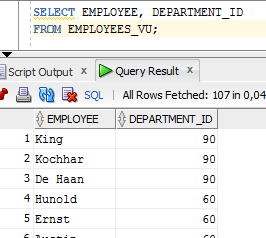
1. Используя своё представление EMPLOYEES\_VU, создайте запрос для вывода всех фамилий и номеров отделов служащих.



……

SELECT EMPLOYEE, DEPARTMENT\_ID

FROM EMPLOYEES\_VU;



1. Создайте представление DEPT50, содержащее номер служащего, фамилию и номер отдела для всех служащих отдела 50. Назовите столбцы представления EMPNO, EMPLOYEEи DEPTNO.Запретите операцию перевода служащего в другой отдел через представление.



CREATE VIEW DEPT50

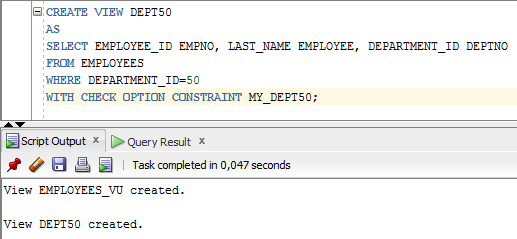
AS

SELECT EMPLOYEE\_ID EMPNO, LAST\_NAME EMPLOYEE, DEPARTMENT\_ID DEPTNO

FROM EMPLOYEES

WHERE DEPARTMENT\_ID=50

WITH CHECK OPTION CONSTRAINT MY\_DEPT50;

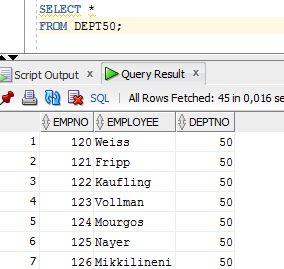


1. Выведите содержимое представления DEPT50.



SELECT \*

FROM DEPT50;

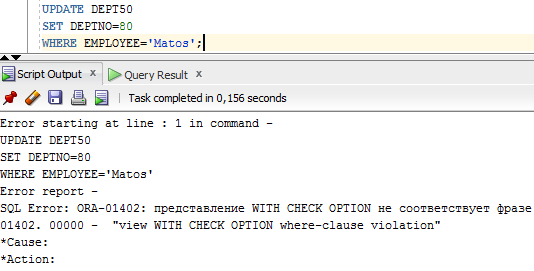


1. Попробуйте сменить номер отдела служащего по фамилии Matos на 80.

UPDATE DEPT50

SET DEPTNO=80

WHERE EMPLOYEE='Matos';



*Дополнительноезадание*

1. Создайте представление SALARY\_VU, включающее фамилию служащего, название отдела, оклад и категорию оклада для всех служащих. Используйтетаблицы EMPLOYEES, DEPARTMENTS и JOB\_GRADES. Соответственноназовитестолбцы Employee, Department, SalaryиGrade.

CREATE VIEW SALARY\_VU

AS SELECT

E.LAST\_NAME "Employee",D.DEPARTMENT\_NAME "Department",

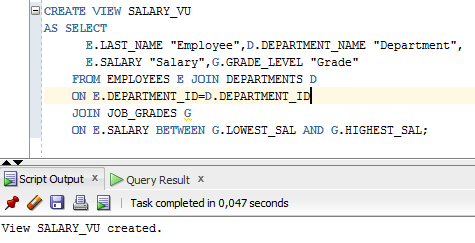
E.SALARY "Salary",G.GRADE\_LEVEL "Grade"

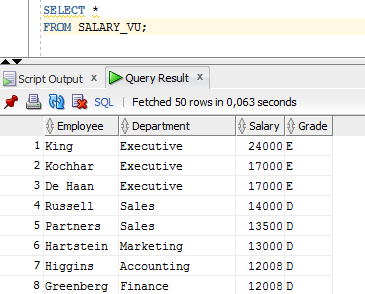
FROM EMPLOYEES E JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID=D.DEPARTMENT\_ID

JOIN JOB\_GRADES G

ON E.SALARY BETWEEN G.LOWEST\_SAL AND G.HIGHEST\_SAL;





# Практическое занятие 10, Часть 2

* + Создание последовательностей
  + Использование последовательностей
  + Создание неуникальных индексов

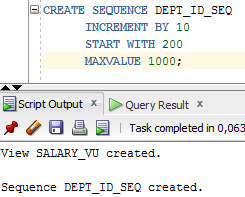
1. Создайте последовательность для столбца главного ключа таблицы DEPT. Последовательность должна начинаться с 200 и иметь максимальное значение 1000. шаг приращения значений – 10. Назовите последовательность DEPT\_ID\_SEQ.

CREATE SEQUENCE DEPT\_ID\_SEQ

INCREMENT BY 10

START WITH 200

MAXVALUE 1000;



1. Создайте скрипт lab12\_3.sql для вставки строки в таблицу DEPT. Обязательно используйте последовательность, созданную Вами для столбца ID. Добавьте в таблицу два отдела: Educationи Administration. Выполните файл, проверьте успешное выполнение вставки.

INSERT INTO DEPT

VALUES (DEPT\_ID\_SEQ.NEXTVAL, INITCAP('&DEPARTMENT\_NAME'));

INSERT INTO DEPT

VALUES (DEPT\_ID\_SEQ.NEXTVAL, INITCAP('Education'));

INSERT INTO DEPT

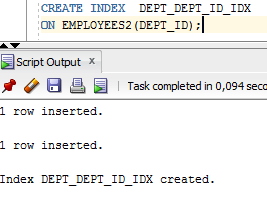
VALUES (DEPT\_ID\_SEQ.NEXTVAL, INITCAP('Administration'));



1. Создайте в таблице EMPLOYEES2 неуникальный индекс для столбца таблицы DEPT\_ID, имеющего ограничение FOREIGNKEY.

CREATE INDEX DEPT\_DEPT\_ID\_IDX

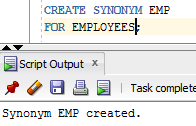
ON EMPLOYEES2(DEPT\_ID);



1. Создайте синоним для таблицы EMPLOYEES. Назовите его EMP.

CREATE SYNONYM EMP

FOR EMPLOYEES;



# Практическое занятие 11

* 1. Создайте скрипт, который позволит получать информацию об именах колонок, типах данных, допустимых объемах ячеек, а также о допустимости пустых ячеек. Пользователь должен иметь возможность вводить имя таблицы. Задайте псевдонимы для столбцов DATA\_PECISIONи DATA\_SCALE. Сохраните скрипт в файле.

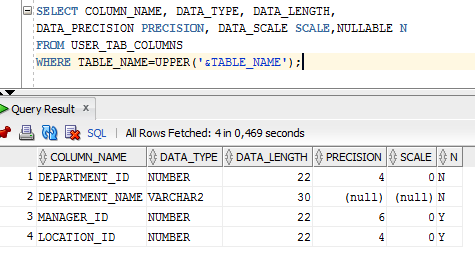
Например, если пользователь ввел DEPARTMENTS, он должен получить следующий результат:

SELECT COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH,

DATA\_PRECISION PRECISION, DATA\_SCALE SCALE,NULLABLE N

FROM USER\_TAB\_COLUMNS

WHERE TABLE\_NAME=UPPER('&TABLE\_NAME');



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **COLUMN\_NAME** | **DATA\_TYPE** | **DATA\_LENGTH** | **PRECISION** | **SCALE** | **N** |
| DEPARTMENT\_ID | NUMBER | 22 | 4 | 0 | N |
| DEPARTMENT\_NAME | VARCHAR2 | 30 |  |  | N |
| MANAGER\_ID | NUMBER | 22 | 6 | 0 | Y |
| LOCATION\_ID | NUMBER | 22 | 4 | 0 | Y |

* 1. Создайте скрипт для получения информации об именах колонок, именах ограничений, условиях поиска и статусе таблицы. Для получения необходимой информации Вы должны соединить таблицы USER\_CONSTRAINTSи USER\_CONS\_COLUMNS. Пользователь должен иметь возможность вводить имя таблицы. Сохраните скрипт в файле.

Например, если пользователь ввел DEPARTMENTS, он должен получить следующий результат:

SELECT CC.COLUMN\_NAME, C.CONSTRAINT\_NAME, C.CONSTRAINT\_TYPE C,

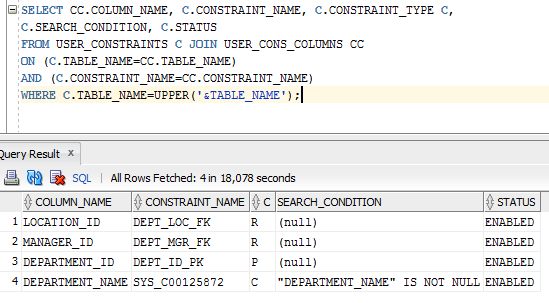
C.SEARCH\_CONDITION, C.STATUS

FROM USER\_CONSTRAINTS C JOIN USER\_CONS\_COLUMNS CC

ON (C.TABLE\_NAME=CC.TABLE\_NAME)

AND (C.CONSTRAINT\_NAME=CC.CONSTRAINT\_NAME)

WHERE C.TABLE\_NAME=UPPER('&TABLE\_NAME');

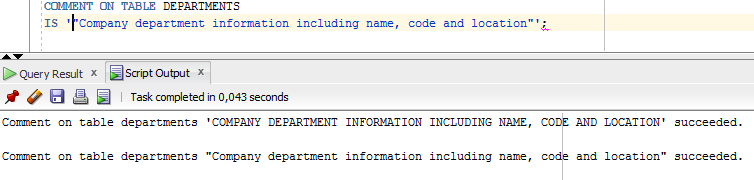


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN\_NAME** | **CONSTRAINT\_NAME** | **C** | **SEARCH\_CONDITION** | **STATUS** |
| DEPARTMENT\_ID | DEPT\_ID\_PK | P |  | ENABLED |
| LOCATION\_ID | DEPT\_LOC\_FK | R |  | ENABLED |
| MANAGER\_ID | DEPT\_MGR\_FK | bR |  | ENABLED |
| DEPARTMENT\_NAME | DEPT\_NAME\_NN | C | "DEPARTMENT\_NAME" IS NOT NULL | ENABLED |

* 1. Добавьте комментарий к таблице DEPARTMENTS. Затем обратитесь с запросом к представлению USER\_TAB\_COMMENTS, для проверки факта создания комментария.

COMMENT ON TABLE DEPARTMENTS

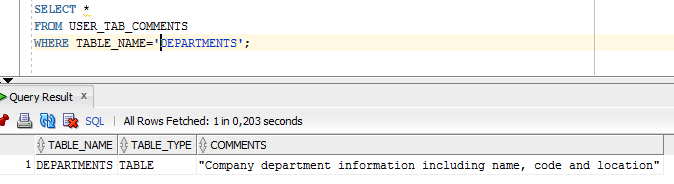
IS '"Company department information including name, code and location"';



SELECT \*

FROM USER\_TAB\_COMMENTS

WHERE TABLE\_NAME='DEPARTMENTS';

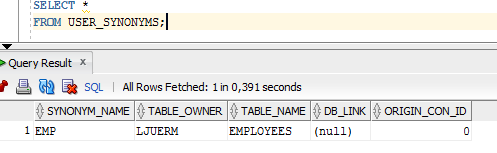


|  |
| --- |
| **COMMENTS** |
| Company department information including name, code and location |

* 1. Найдите имена всех синонимов в вашей схеме

SELECT \*

FROM USER\_SYNONYMS;



|  |  |  |  |
| --- | --- | --- | --- |
| **SYNONYM\_NAME** | **TABLE\_OWNER** | **TABLE\_NAME** | **DB\_LINK** |
| EMP | HR | EMPLOYEES |  |

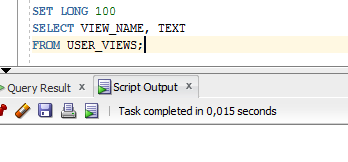
* 1. Из представления словаря данных USER\_VIEWSвыберите столбцы VIEW\_NAMEи TEXT.

**Примечание:**для просмотра содержимого столбца типа LONG выполните команду SETLONGn, где n – число выводимых запросом символов из столбца типа LONG.

SET LONG 100

SELECT VIEW\_NAME, TEXT

FROM USER\_VIEWS;





6. Найдите имена всех ваших последовательностей. Напишите запрос для получения следующей информации о ваших последовательностях: имя последовательности, максимальное значение, величина приращения, последнее значение. Сохранитескриптвфайле. Выполнитескрипт.lab11\_6

SELECT SEQUENCE\_NAME,

CASE WHEN MAX\_VALUE<10000 THEN to\_char(MAX\_VALUE)

ELSE to\_char(MAX\_VALUE, '9.9999EEEE')

END MAX\_VALUE, INCREMENT\_BY, LAST\_NUMBER

FROM USER\_SEQUENCES;

|  |  |  |  |
| --- | --- | --- | --- |
| **SEQUENCE\_NAME** | **MAX\_VALUE** | **INCREMENT\_BY** | **LAST\_NUMBER** |
| DEPARTMENTS\_SEQ | 9990 | 10 | 280 |
| EMPLOYEES\_SEQ | 1,0000E+27 | 1 | 207 |
| LOCATIONS\_SEQ | 9900 | 100 | 3300 |

**ОГЛАВЛЕНИЕ**

[Практическое занятие 1 1](#_Toc192567669)

[Практическое занятие 2 5](#_Toc192567670)

[Практическое занятие 3, часть 1 8](#_Toc192567671)

[Практическое занятие 3, часть 2 10](#_Toc192567672)

[Практическое занятие 4 13](#_Toc192567673)

[Практическое занятие 5 15](#_Toc192567674)

[Практическое занятие 6 19](#_Toc192567675)

[Практическое занятие 7 22](#_Toc192567676)

[Практическое занятие 8 23](#_Toc192567677)

[Практическое занятие 9 25](#_Toc192567678)

[Практическое занятие 10, Часть 1 26](#_Toc192567679)

[Практическое занятие 10, Часть 2 27](#_Toc192567680)

[Практическое занятие 11 28](#_Toc192567681)