MALAD KANDIVALI EDUCATION SOCIETY’S
NAGINDAS KHANDWALA COLLEGE OF COMMERCE,
ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS
KHANDWALA COLLEGE OF SCIENCE
MALAD [W], MUMBAI – 64
(AUTONOMOUS)

(Reaccredited ‘A’ Grade by NAAC)
(AFFILIATED TO UNIVERSITY OF MUMBAI)
(ISO 9001:2015)

CERTIFICATE

Name: Mr./Ms._Aniket_Prajapati___________________________________

Roll No: _44______ **Programme**: BSc - CS **Semester**: II

This is certified to be a bonafide record of practical works done by the above student in the college laboratory for the course **Database Management Systems I** (Course Code: **2023UISPR**) for the partial fulfillment of Second Semester of BSc IT/CS during the academic year 2020-2021.

The journal work is the original study work that has been duly approved in the year 2020-2021 by the undersigned.

____________________ ____________________

External Examiner **Subject-In-Charge**
(Ms.Sweety Garg)

Date of Examination: (College Stamp)

Name: Aniket Prajapati **Roll No: 44**

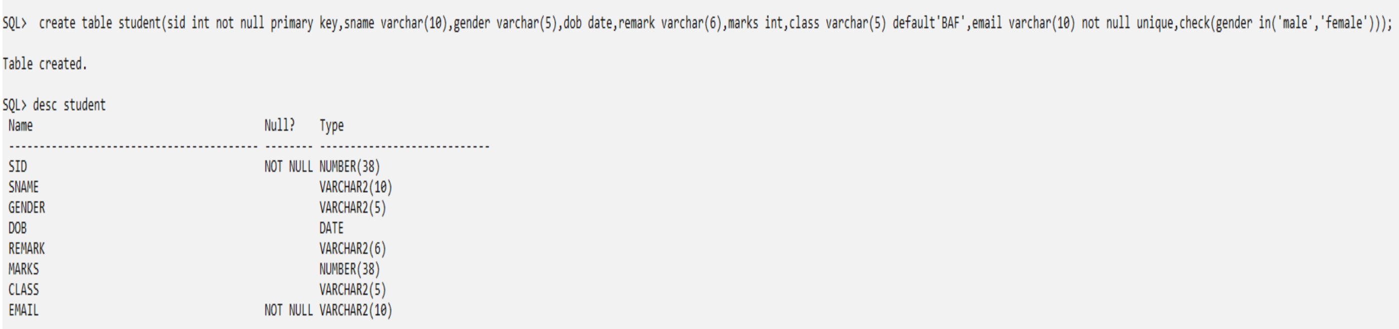
Sr. No.	DATE	TITLE	SIGN
1.	28/1/21	Study of Data Definition Language Statement	
2.	4/2/21	Study of Data Manipulation Language Statement	
3.	12/2/21	Study of SELECT Statement.	
4.	18/2/21	Draw ER diagram for given scenario/project/case study	
5.	4/3/21	Study of various type of JOINS	
6.	18/3/21	Study of different functions	
7.	18/3/21	Study of various types of SET OPERATORS	
8.	25/3/21	Study of various types of views	
9.	25/3/21	Study of subqueries with all its clauses	
10.	8/4/21	Study of Transaction (Commit/ Rollback), Locks	
11.	8/4/21	Implementing deadlocks	

Practical 1: Study of Data Definition Language Statement

A) Write the query for the following

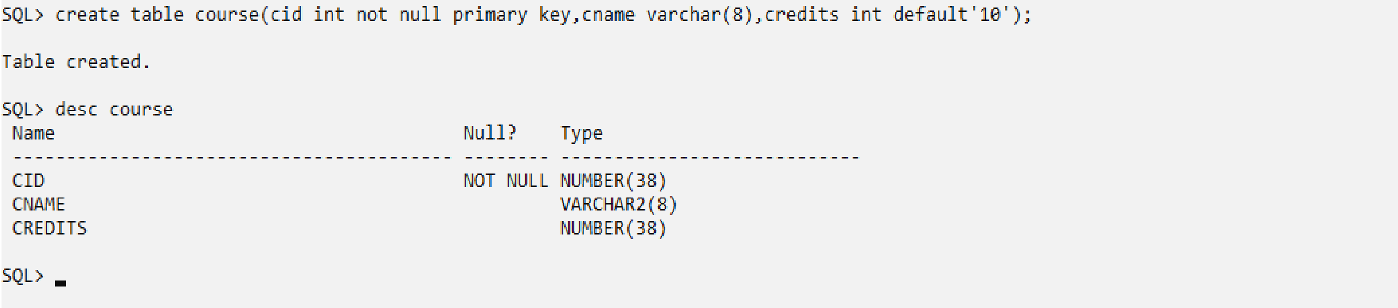
1) Create the following tables and include the necessary constraints NOT NULL, DEFAULT, CHECK, PRIMARY KEY, UNIQUE.

a) Student (sid, sname, gender, dob, remark,marks, class, email)



```
SQL>  create table student(sid int not null primary key,sname varchar(10),gender varchar(5),dob date,remark varchar(6),marks int,class varchar(5) default'BAF',email varchar(10) not null unique,check(gender in('male','female')));

Table created.

SQL> desc student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(38)
 SNAME                                              VARCHAR2(10)
 GENDER                                             VARCHAR2(5)
 DOB                                                DATE
 REMARK                                             VARCHAR2(6)
 MARKS                                              NUMBER(38)
 CLASS                                              VARCHAR2(5)
 EMAIL                                     NOT NULL VARCHAR2(10)
```

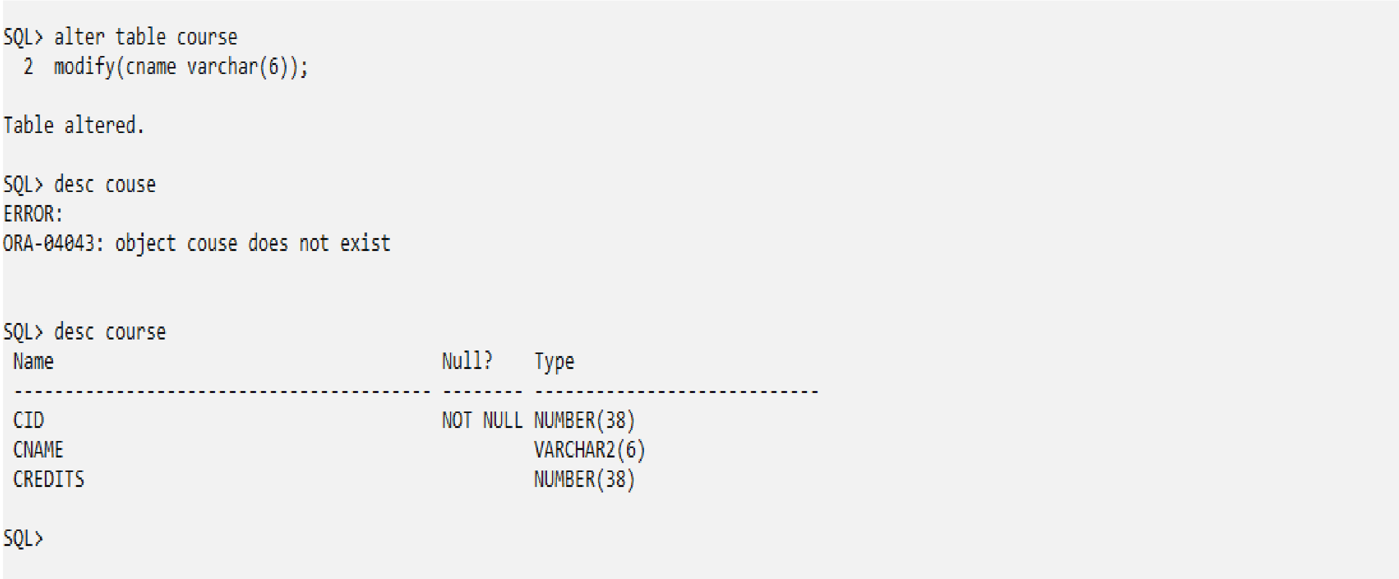
b) Course (cid, cname, credits)



```
SQL> create table course(cid int not null primary key,cname varchar(8),credits int default'10');

Table created.

SQL> desc course
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL NUMBER(38)
 CNAME                                              VARCHAR2(8)
 CREDITS                                            NUMBER(38)

SQL> _
```

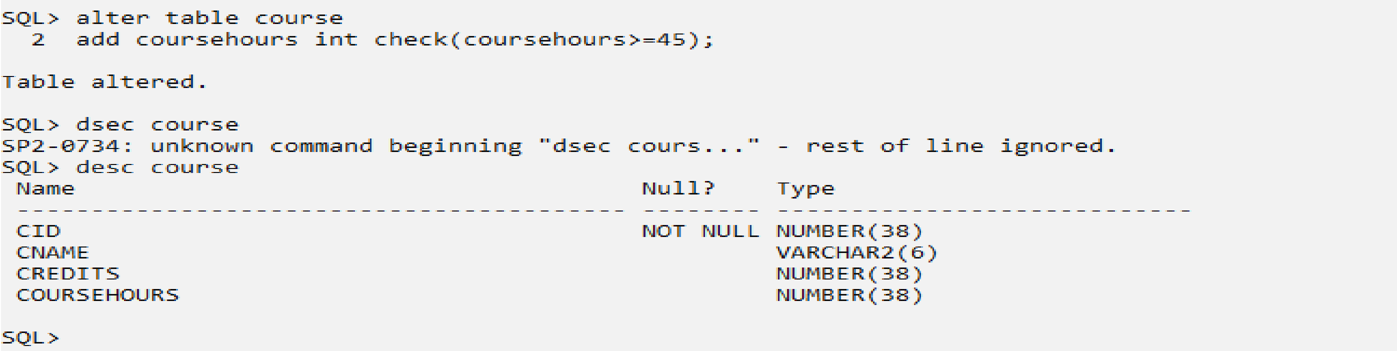
2) Alter the structure of the Course table

   a) Modify datatype of cname.



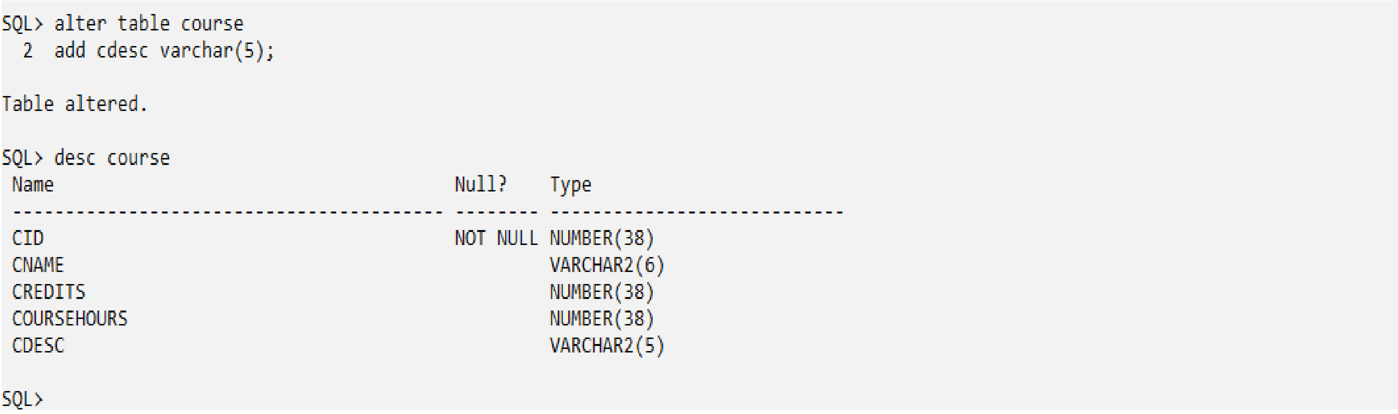
```
SQL> alter table course
  2  modify(cname varchar(6));

Table altered.

SQL> desc couse
ERROR:
ORA-04043: object couse does not exist


SQL> desc course
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL NUMBER(38)
 CNAME                                              VARCHAR2(6)
 CREDITS                                            NUMBER(38)

SQL>
```

b) Add a column coursehours with minimum course hours greater than 45.



```
SQL> alter table course
  2  add coursehours int check(coursehours>=45);

Table altered.

SQL> dsec course
SP2-0734: unknown command beginning "dsec cours..." - rest of line ignored.
SQL> desc course
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL NUMBER(38)
 CNAME                                              VARCHAR2(6)
 CREDITS                                            NUMBER(38)
 COURSEHOURS                                        NUMBER(38)

SQL>
```

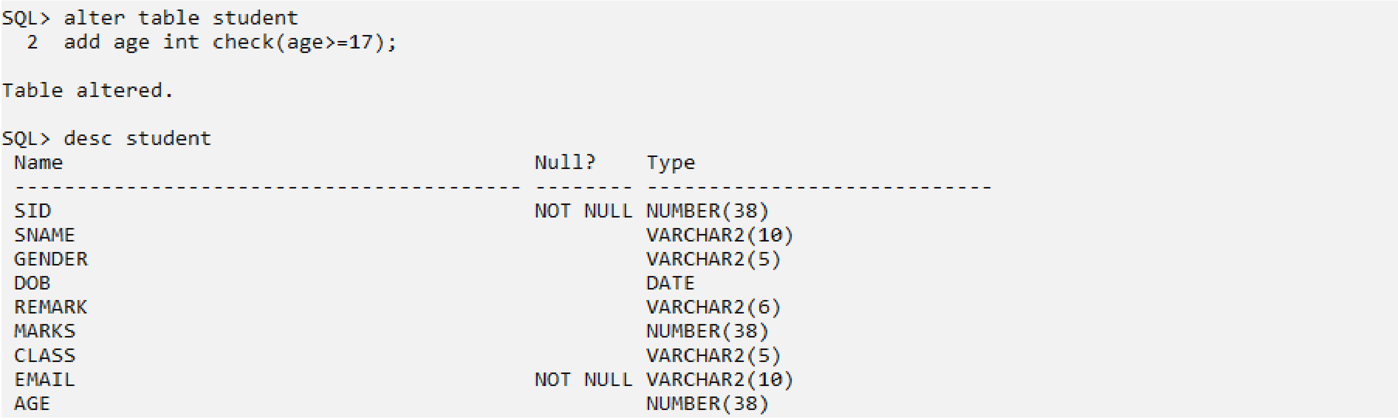
c) Add a column cdesc



```
SQL> alter table course
  2  add cdesc varchar(5);

Table altered.

SQL> desc course
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL NUMBER(38)
 CNAME                                              VARCHAR2(6)
 CREDITS                                            NUMBER(38)
 COURSEHOURS                                        NUMBER(38)
 CDESC                                              VARCHAR2(5)

SQL>
```

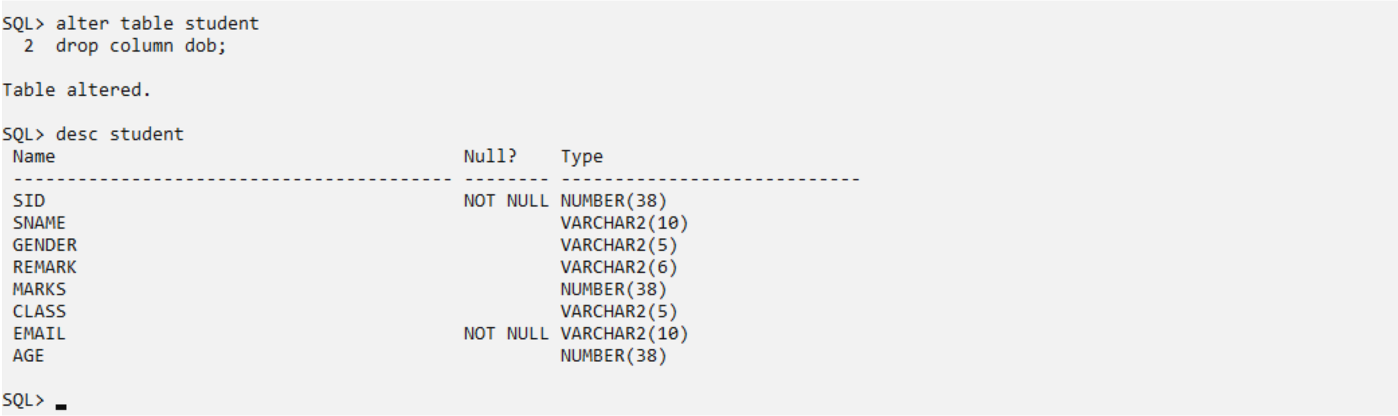
3) Alter the structure of Student Table

a) Add column age with minimum age as 17.



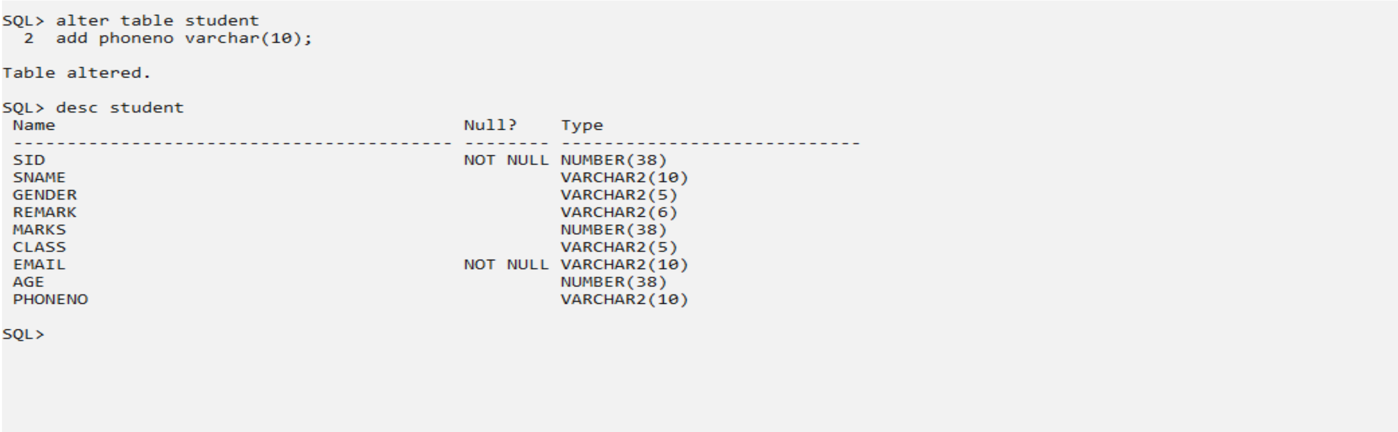
```
SQL> alter table student
  2  add age int check(age>=17);

Table altered.

SQL> desc student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(38)
 SNAME                                              VARCHAR2(10)
 GENDER                                             VARCHAR2(5)
 DOB                                                DATE
 REMARK                                             VARCHAR2(6)
 MARKS                                              NUMBER(38)
 CLASS                                              VARCHAR2(5)
 EMAIL                                     NOT NULL VARCHAR2(10)
 AGE                                                NUMBER(38)
```

b) Delete the column dob



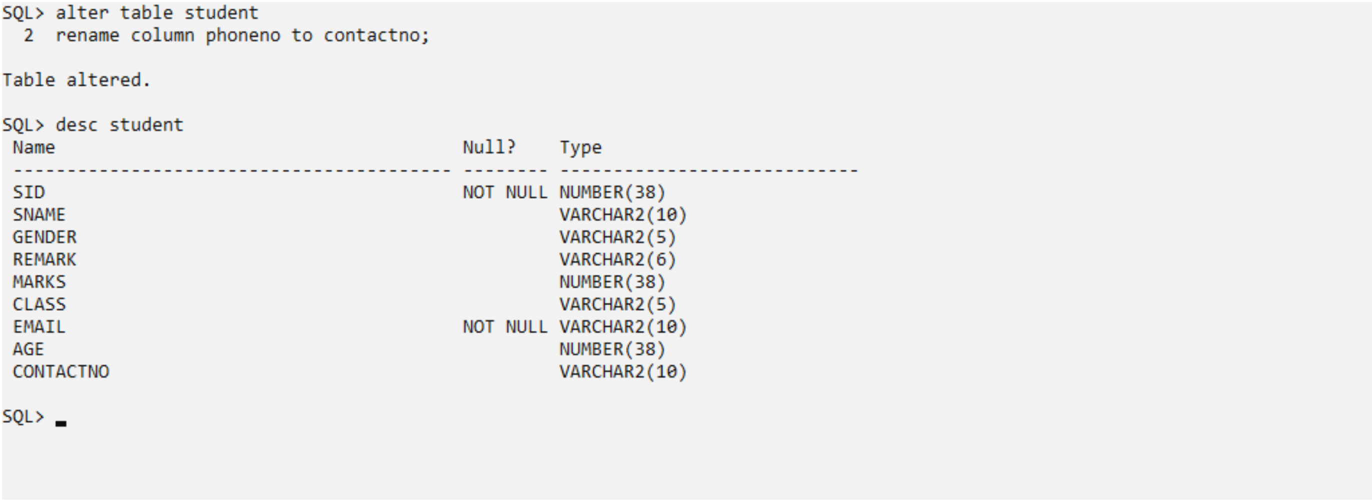
```
SQL> alter table student
  2  drop column dob;

Table altered.

SQL> desc student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(38)
 SNAME                                              VARCHAR2(10)
 GENDER                                             VARCHAR2(5)
 REMARK                                             VARCHAR2(6)
 MARKS                                              NUMBER(38)
 CLASS                                              VARCHAR2(5)
 EMAIL                                     NOT NULL VARCHAR2(10)
 AGE                                                NUMBER(38)

SQL> _
```

c) Add a column phoneno.



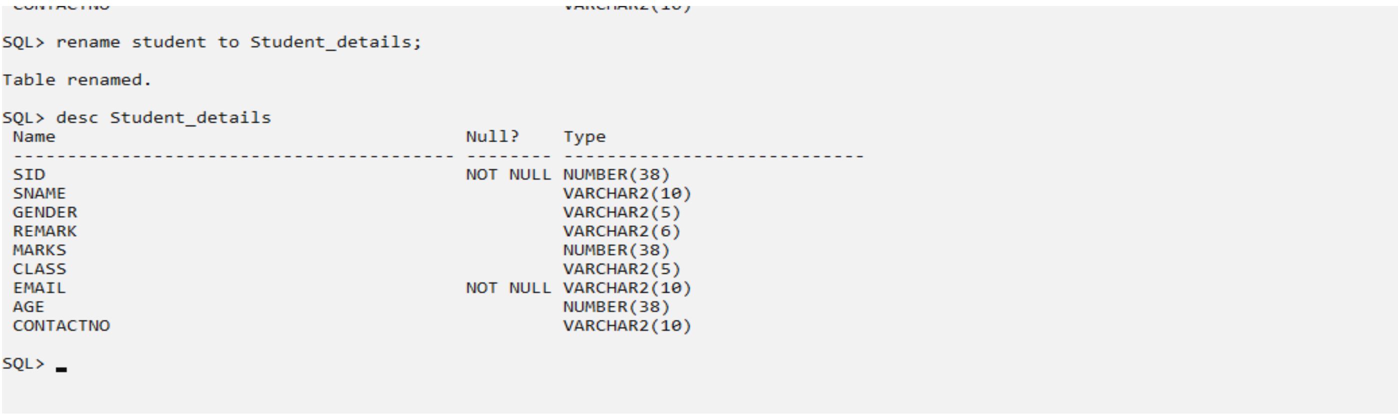
```
SQL> alter table student
  2  add phoneno varchar(10);

Table altered.

SQL> desc student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(38)
 SNAME                                              VARCHAR2(10)
 GENDER                                             VARCHAR2(5)
 REMARK                                             VARCHAR2(6)
 MARKS                                              NUMBER(38)
 CLASS                                              VARCHAR2(5)
 EMAIL                                     NOT NULL VARCHAR2(10)
 AGE                                                NUMBER(38)
 PHONENO                                            VARCHAR2(10)

SQL>
```

d) Rename phoneno to contactno



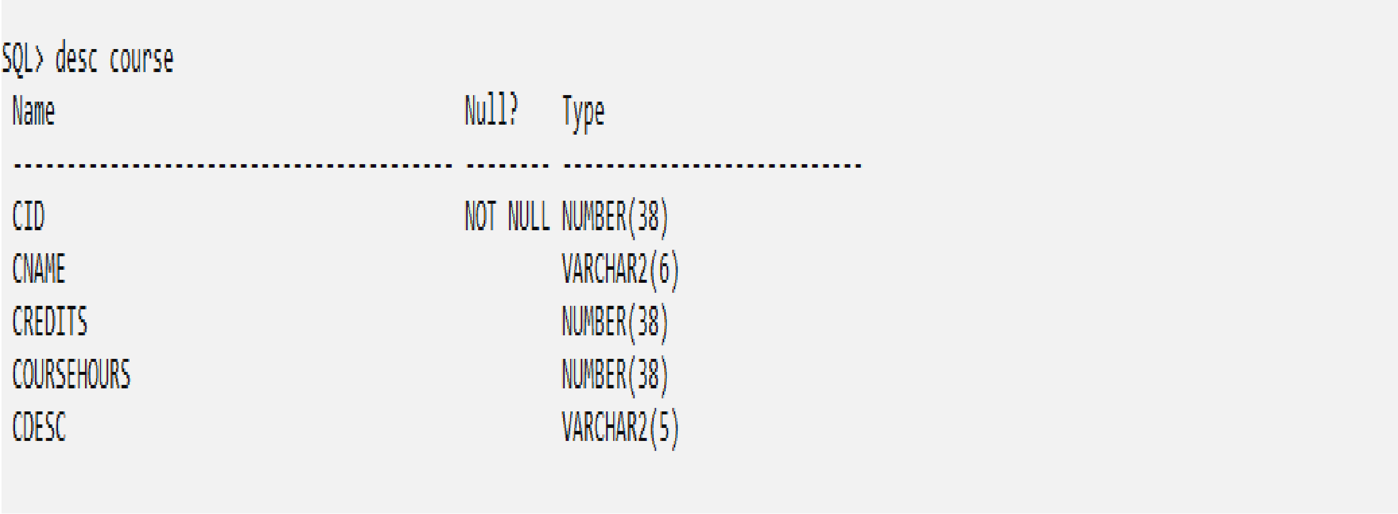
```
SQL> alter table student
  2  rename column phoneno to contactno;

Table altered.

SQL> desc student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(38)
 SNAME                                              VARCHAR2(10)
 GENDER                                             VARCHAR2(5)
 REMARK                                             VARCHAR2(6)
 MARKS                                              NUMBER(38)
 CLASS                                              VARCHAR2(5)
 EMAIL                                     NOT NULL VARCHAR2(10)
 AGE                                                NUMBER(38)
 CONTACTNO                                          VARCHAR2(10)

SQL> _
```

4) Rename Student table as Student_details.

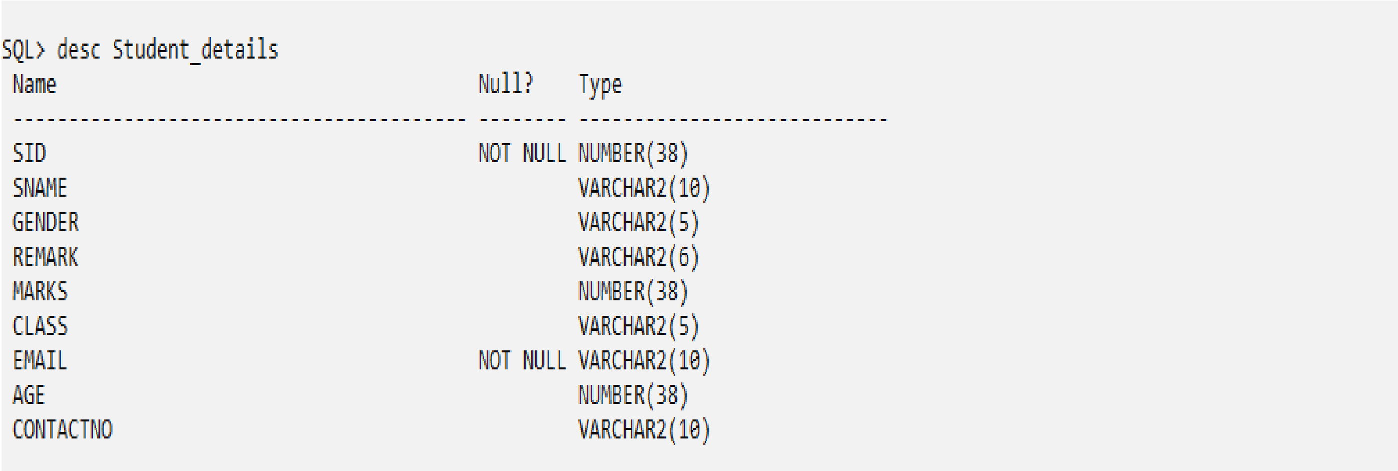


```
 CONTACTNO                                          VARCHAR2(10)

SQL> rename student to Student_details;

Table renamed.

SQL> desc Student_details
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(38)
 SNAME                                              VARCHAR2(10)
 GENDER                                             VARCHAR2(5)
 REMARK                                             VARCHAR2(6)
 MARKS                                              NUMBER(38)
 CLASS                                              VARCHAR2(5)
 EMAIL                                     NOT NULL VARCHAR2(10)
 AGE                                                NUMBER(38)
 CONTACTNO                                          VARCHAR2(10)

SQL> _
```

5) Describe the structure of both the tables.

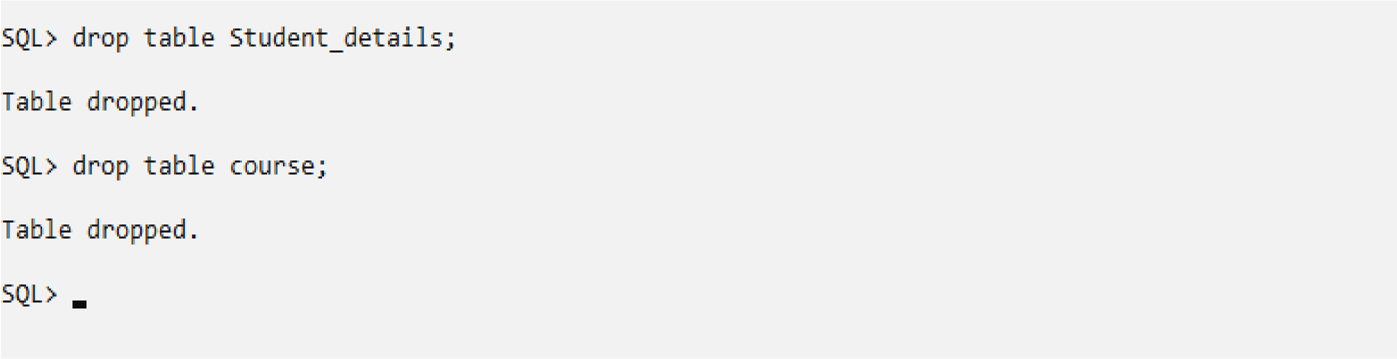


```
SQL> desc course
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL NUMBER(38)
 CNAME                                              VARCHAR2(6)
 CREDITS                                            NUMBER(38)
 COURSEHOURS                                        NUMBER(38)
 CDESC                                              VARCHAR2(5)
```



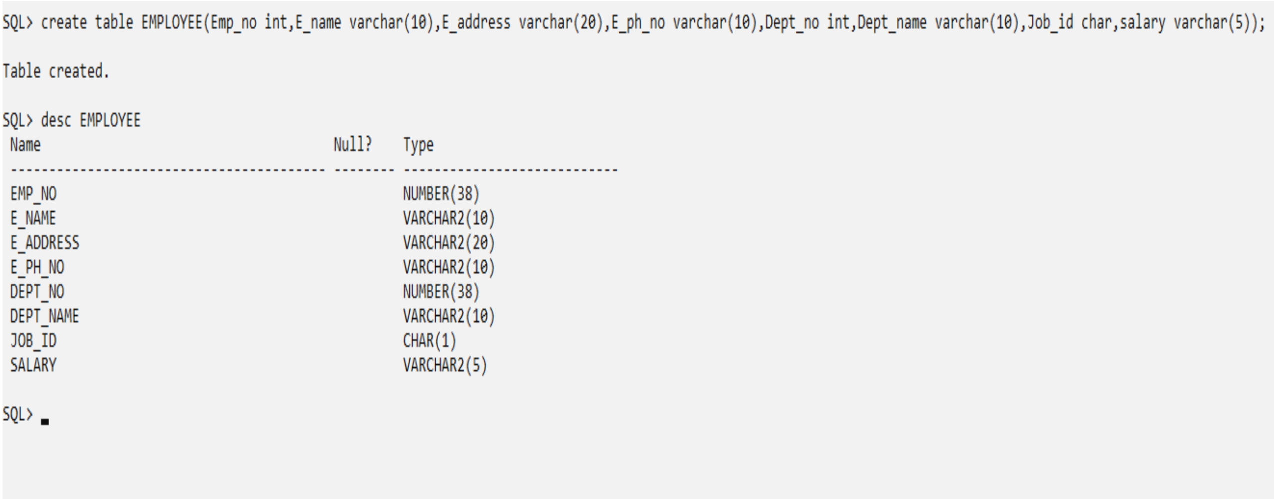
```
SQL> desc Student_details
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(38)
 SNAME                                              VARCHAR2(10)
 GENDER                                             VARCHAR2(5)
 REMARK                                             VARCHAR2(6)
 MARKS                                              NUMBER(38)
 CLASS                                              VARCHAR2(5)
 EMAIL                                     NOT NULL VARCHAR2(10)
 AGE                                                NUMBER(38)
 CONTACTNO                                          VARCHAR2(10)
```

6) Drop the table student_details and Course.



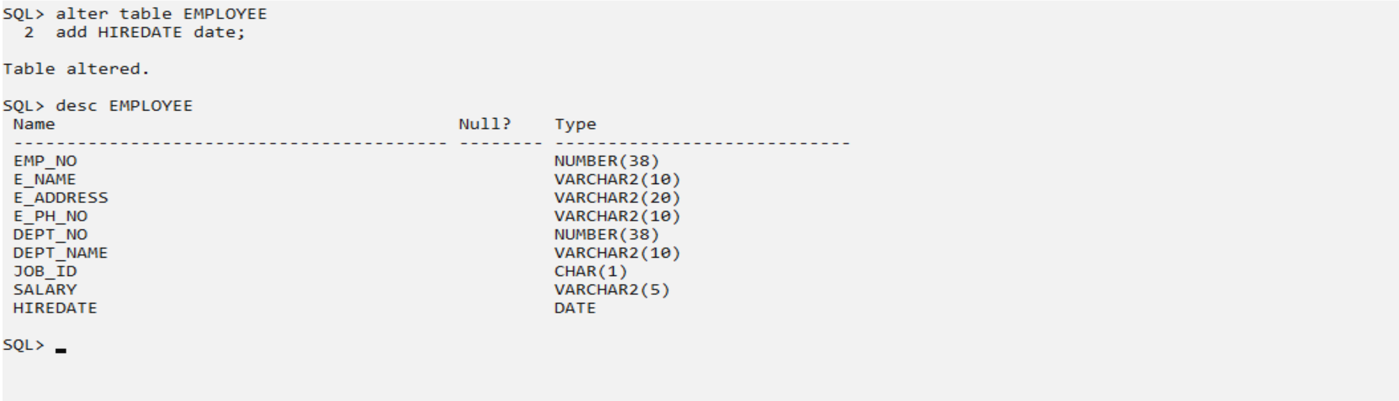
```
SQL> drop table Student_details;

Table dropped.

SQL> drop table course;

Table dropped.

SQL>
```

B) 1. Create a table EMPLOYEE with following attributes and specific data types and constraints required
(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary)



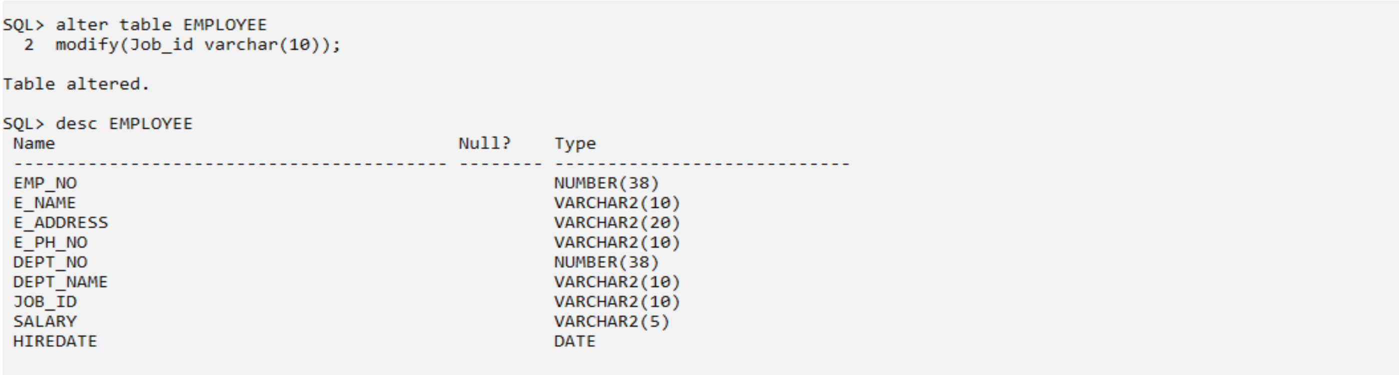
```
SQL> create table EMPLOYEE(Emp_no int,E_name varchar(10),E_address varchar(20),E_ph_no varchar(10),Dept_no int,Dept_name varchar(10),Job_id char,salary varchar(5));

Table created.

SQL> desc EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_NO                                             NUMBER(38)
 E_NAME                                             VARCHAR2(10)
 E_ADDRESS                                          VARCHAR2(20)
 E_PH_NO                                            VARCHAR2(10)
 DEPT_NO                                            NUMBER(38)
 DEPT_NAME                                          VARCHAR2(10)
 JOB_ID                                             CHAR(1)
 SALARY                                             VARCHAR2(5)

SQL>
```

2. Add a new column HIREDATE to the existing relation.



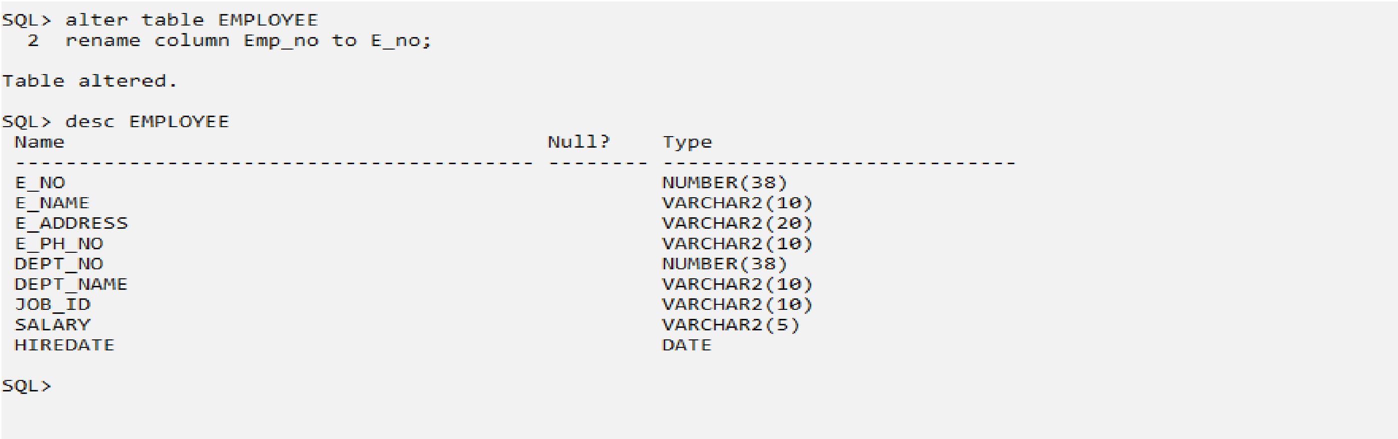
```
SQL> alter table EMPLOYEE
  2  add HIREDATE date;

Table altered.

SQL> desc EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_NO                                             NUMBER(38)
 E_NAME                                             VARCHAR2(10)
 E_ADDRESS                                          VARCHAR2(20)
 E_PH_NO                                            VARCHAR2(10)
 DEPT_NO                                            NUMBER(38)
 DEPT_NAME                                          VARCHAR2(10)
 JOB_ID                                             CHAR(1)
 SALARY                                             VARCHAR2(5)
 HIREDATE                                           DATE

SQL>
```

3. Change the datatype of JOB_ID from char to varchar2.



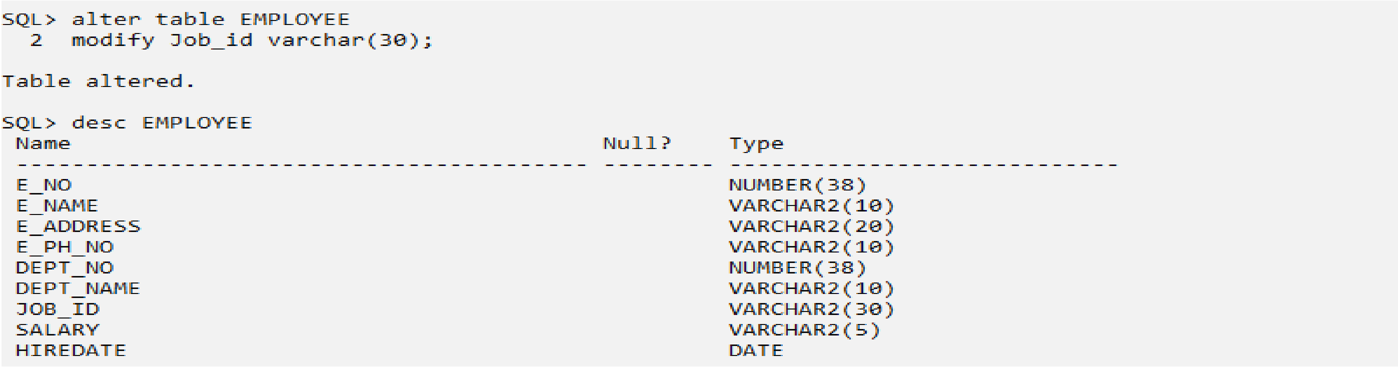
```
SQL> alter table EMPLOYEE
  2  modify(Job_id varchar(10));

Table altered.

SQL> desc EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_NO                                             NUMBER(38)
 E_NAME                                             VARCHAR2(10)
 E_ADDRESS                                          VARCHAR2(20)
 E_PH_NO                                            VARCHAR2(10)
 DEPT_NO                                            NUMBER(38)
 DEPT_NAME                                          VARCHAR2(10)
 JOB_ID                                             VARCHAR2(10)
 SALARY                                             VARCHAR2(5)
 HIREDATE                                           DATE
```

4. Change the name of column/field Emp_no to E_no.



```
SQL> alter table EMPLOYEE
  2  rename column Emp_no to E_no;

Table altered.

SQL> desc EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 E_NO                                               NUMBER(38)
 E_NAME                                             VARCHAR2(10)
 E_ADDRESS                                          VARCHAR2(20)
 E_PH_NO                                            VARCHAR2(10)
 DEPT_NO                                            NUMBER(38)
 DEPT_NAME                                          VARCHAR2(10)
 JOB_ID                                             VARCHAR2(10)
 SALARY                                             VARCHAR2(5)
 HIREDATE                                           DATE

SQL>
```

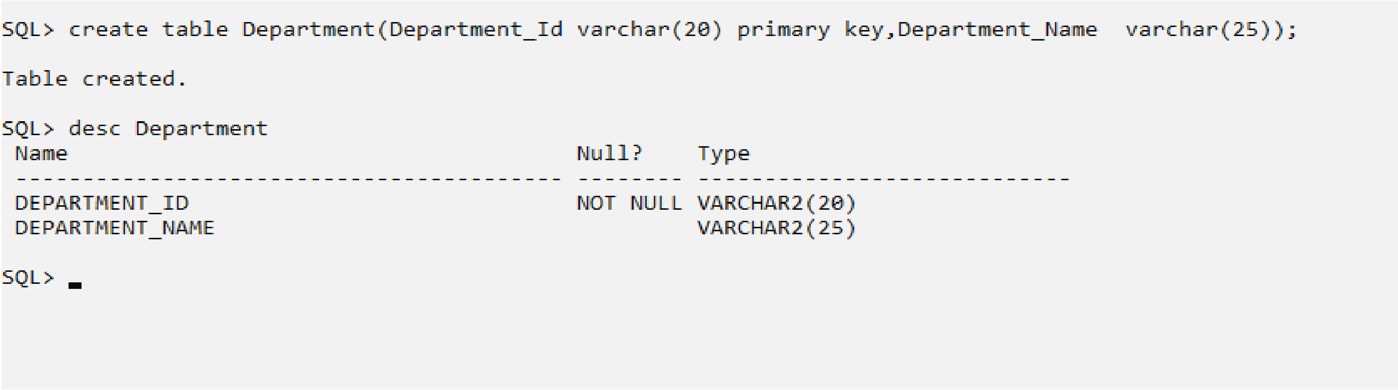
5. Modify the column width of the job field of emp table.



```
SQL> alter table EMPLOYEE
  2  modify Job_id varchar(30);

Table altered.

SQL> desc EMPLOYEE
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 E_NO                                               NUMBER(38)
 E_NAME                                             VARCHAR2(10)
 E_ADDRESS                                          VARCHAR2(20)
 E_PH_NO                                            VARCHAR2(10)
 DEPT_NO                                            NUMBER(38)
 DEPT_NAME                                          VARCHAR2(10)
 JOB_ID                                             VARCHAR2(30)
 SALARY                                             VARCHAR2(5)
 HIREDATE                                           DATE
```

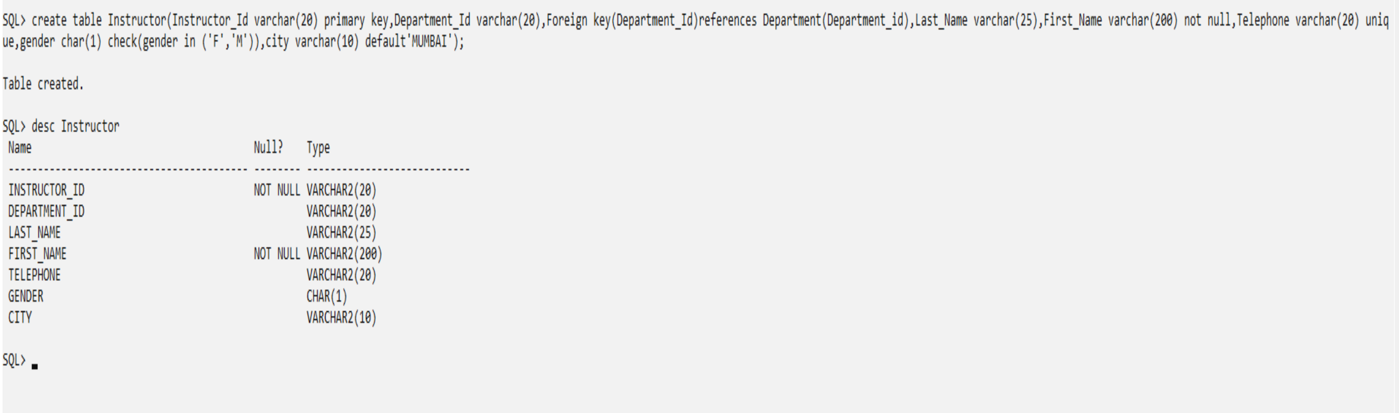
C) Create the following tables with specified attributes and constraints

Department Table: Department_Id varchar2(20) primarykey, Department_Name varchar2(25) with required data.



```
SQL> create table Department(Department_Id varchar(20) primary key,Department_Name  varchar(25));

Table created.

SQL> desc Department
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPARTMENT_ID                             NOT NULL VARCHAR2(20)
 DEPARTMENT_NAME                                    VARCHAR2(25)

SQL>
```

Instructor Table: Instructor_id varchar2(20) primary key,foreign key, Department_Id varchar2(20) ,Last_Name varchar2(25), First_Name varchar2(200) must have value, Telephone varchar2(20) must be unique, gender char(1) must be either ‘F’or ‘M’,city varchar(10) default value must be ‘MUMBAI’

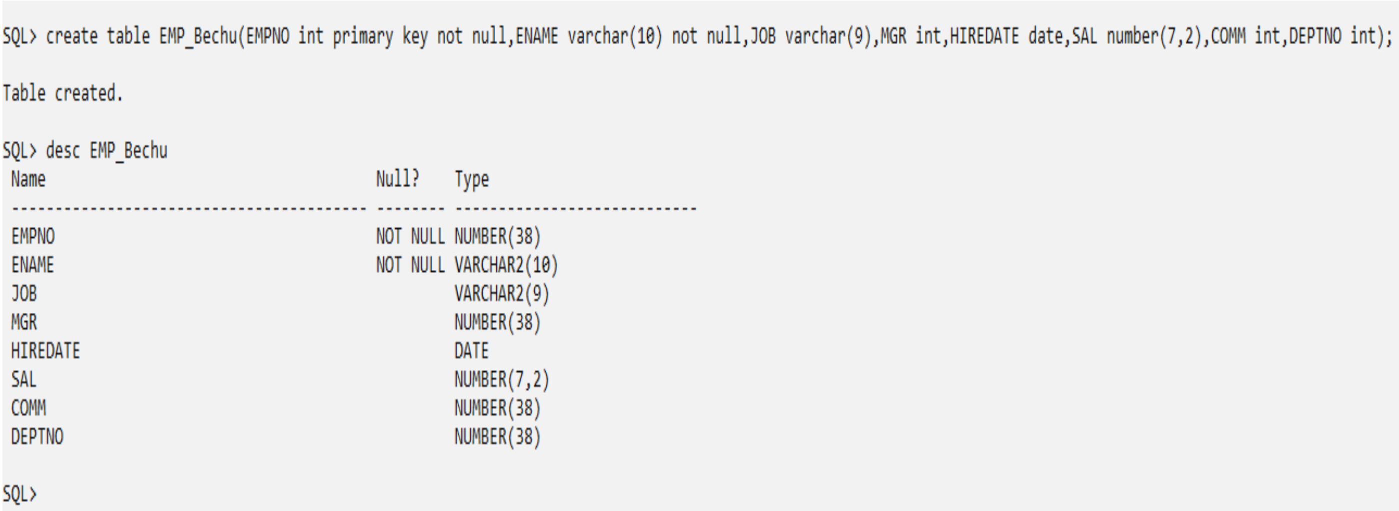


```
SQL> create table Instructor(Instructor_Id varchar(20) primary key,Department_Id varchar(20),Foreign key(Department_Id)references Department(Department_id),Last_Name varchar(25),First_Name varchar(200) not null,Telephone varchar(20) uniq
ue,gender char(1) check(gender in ('F','M')),city varchar(10) default'MUMBAI');

Table created.

SQL> desc Instructor
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 INSTRUCTOR_ID                             NOT NULL VARCHAR2(20)
 DEPARTMENT_ID                                      VARCHAR2(20)
 LAST_NAME                                          VARCHAR2(25)
 FIRST_NAME                                NOT NULL VARCHAR2(200)
 TELEPHONE                                          VARCHAR2(20)
 GENDER                                             CHAR(1)
 CITY                                               VARCHAR2(10)

SQL>
```

D) Create the following described below:

Table Name: EMP

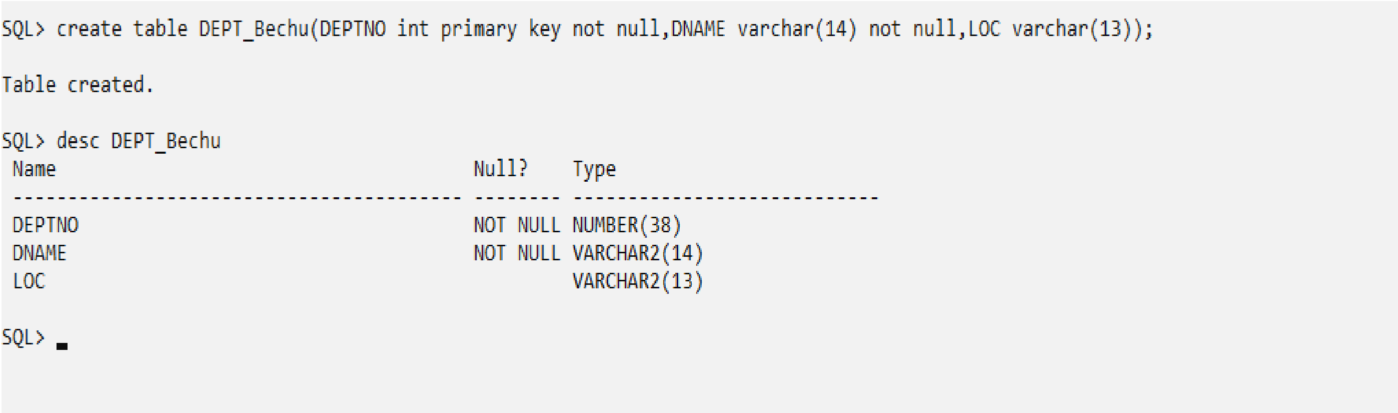
Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPNO	Int	-	-	-	Yes	-
ENAME	Varchar2	10	-	-	-	No
JOB	Varchar2	9	-	-	-	✓
MGR	Int	-	-	-	-	✓
HIREDATE	Date	-	-	-	-	✓
SAL	Number	-	7	2	-	✓
COMM	Int	-	-	-	-	✓
DEPTNO	Int	-	-	-	-	✓



```
SQL> create table EMP_Bechu(EMPNO int primary key not null,ENAME varchar(10) not null,JOB varchar(9),MGR int,HIREDATE date,SAL number(7,2),COMM int,DEPTNO int);

Table created.

SQL> desc EMP_Bechu
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(38)
 ENAME                                     NOT NULL VARCHAR2(10)
 JOB                                                VARCHAR2(9)
 MGR                                                NUMBER(38)
 HIREDATE                                           DATE
 SAL                                                NUMBER(7,2)
 COMM                                               NUMBER(38)
 DEPTNO                                             NUMBER(38)

SQL>
```

Table Name: DEPT

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
DEPTNO	Int	-	-	-	Yes	-
DNAME	Varchar2	14	-	-	-	No
LOC	Varchar2	13	-	-	-	✓



```
SQL> create table DEPT_Bechu(DEPTNO int primary key not null,DNAME varchar(14) not null,LOC varchar(13));

Table created.

SQL> desc DEPT_Bechu
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPTNO                                    NOT NULL NUMBER(38)
 DNAME                                     NOT NULL VARCHAR2(14)
 LOC                                                VARCHAR2(13)

SQL> _
```

Practical no:2

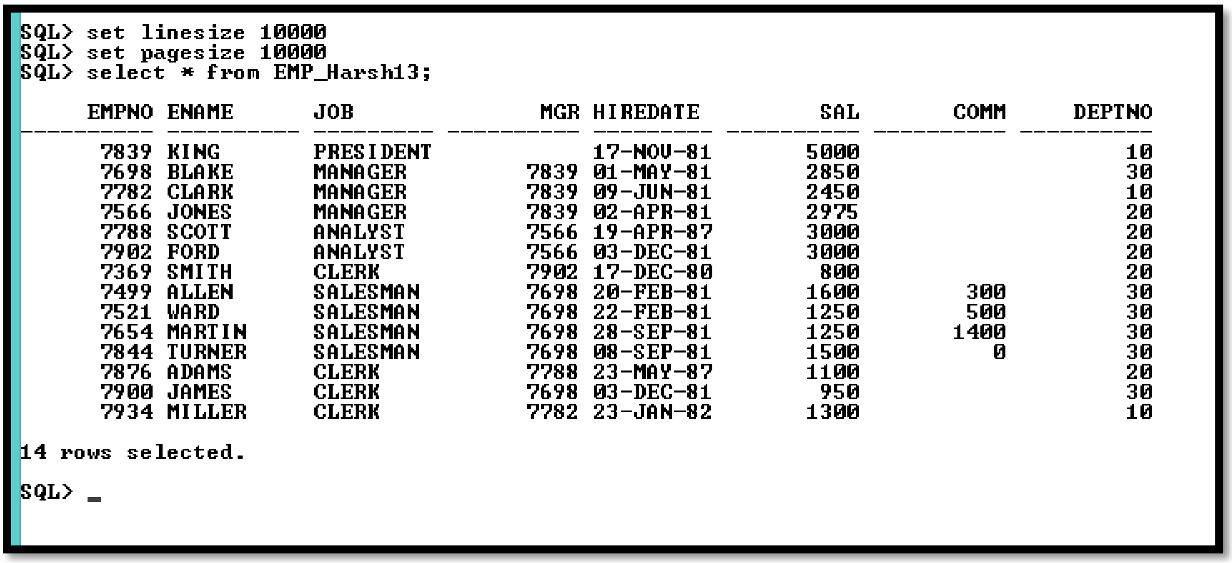
Study of Data Manipulation Language Statement

A) Insert the following records in above created table

EMP_Aniket44_TABLE

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-Nov-81	5000		10
7698	BLAKE	MANAGER	7839	01-May-81	2850		30
7782	CLARK	MANAGER	7839	09-Jun-81	2450		10
7566	JONES	MANAGER	7839	02-Apr-81	2975		20
7788	SCOTT	ANALYST	7566	19-Apr-87	3000		20
7902	FORD	ANALYST	7566	03-Dec-81	3000		20
7369	SMITH	CLERK	7902	17-Dec-80	800		20
7499	ALLEN	SALESMAN	7698	20-Feb-81	1600	300	30
7521	WARD	SALESMAN	7698	22-Feb-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	1400	30
7844	TURNER	SALESMAN	7698	08-Sep-81	1500	0	30
7876	ADAMS	CLERK	7788	23-May-87	1100		20
7900	JAMES	CLERK	7698	03-Dec-81	950		30
7934	MILLER	CLERK	7782	23-Jan-82	1300		10

EMP_Aniket44 Table Record List:

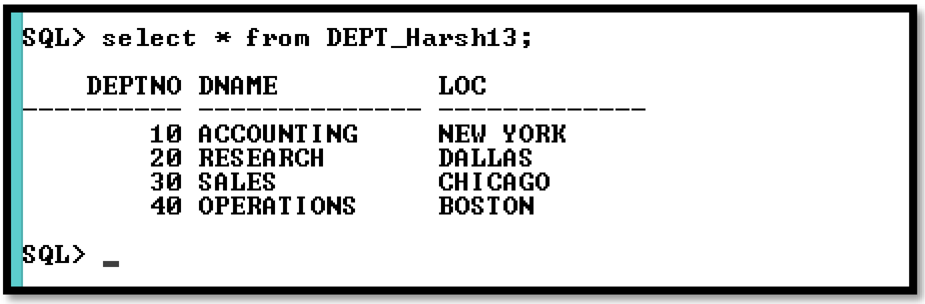


```
SQL> set linesize 10000
SQL> set pagesize 10000
SQL> select * from EMP_Harsh13;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.

SQL> _
```

<u>DEPT_Aniket44_TABLE</u>

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

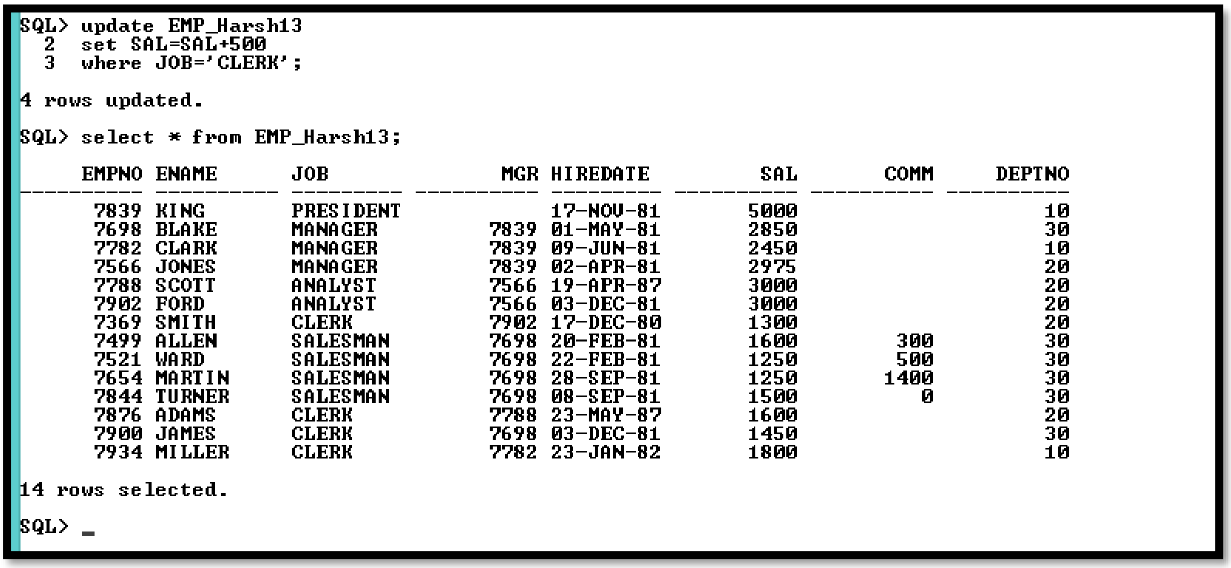
<u>DEPT_Aniket44 Table Record List:</u>



```
SQL> select * from DEPT_Harsh13;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

SQL> _
```

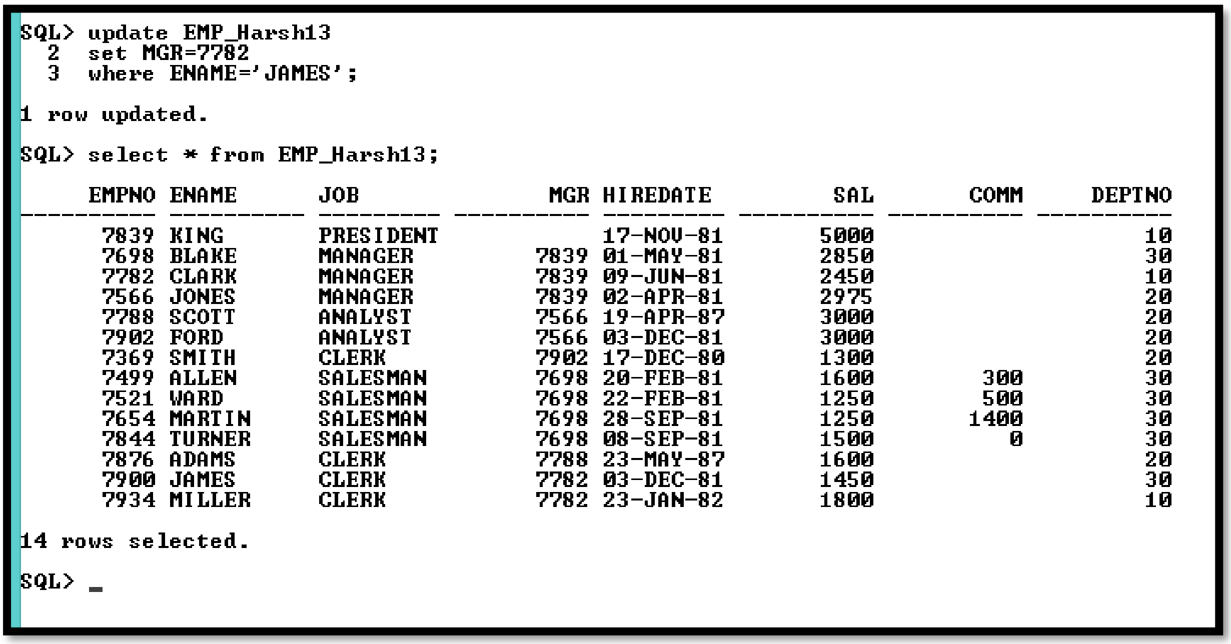
B) Update and Delete Queries

1) Update the salary of employees working as CLERK by 500.



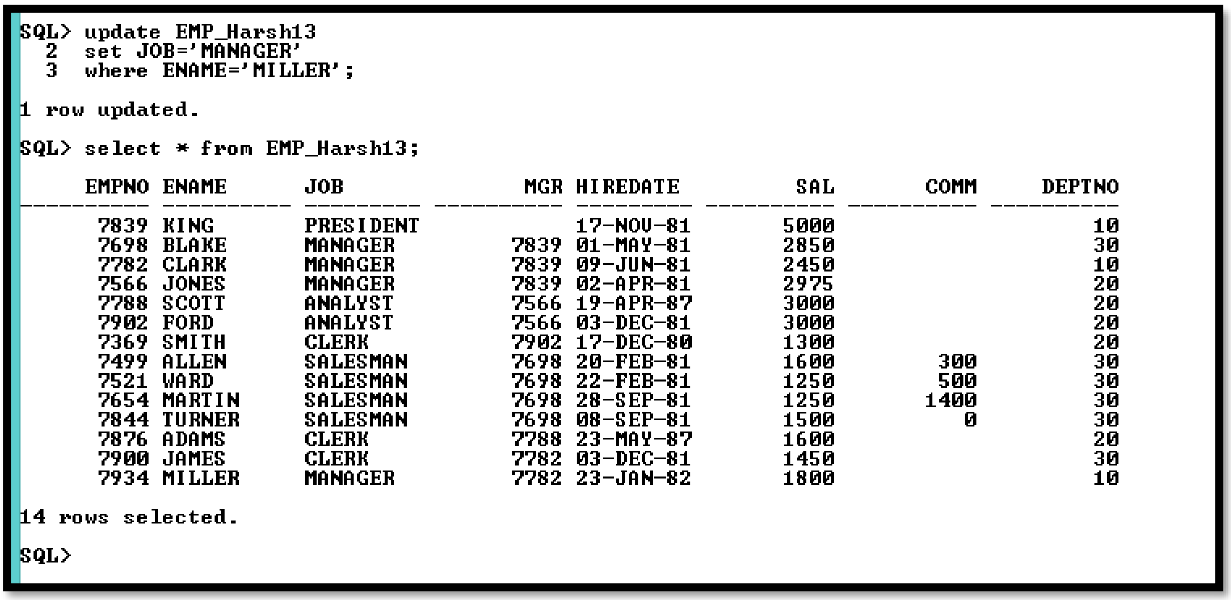
```
SQL> update EMP_Harsh13
  2  set SAL=SAL+500
  3  where JOB='CLERK';

4 rows updated.

SQL> select * from EMP_Harsh13;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80       1300                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1600                    20
      7900 JAMES      CLERK           7698 03-DEC-81       1450                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1800                    10

14 rows selected.

SQL> _
```

2) Update the manager of James as CLARK.



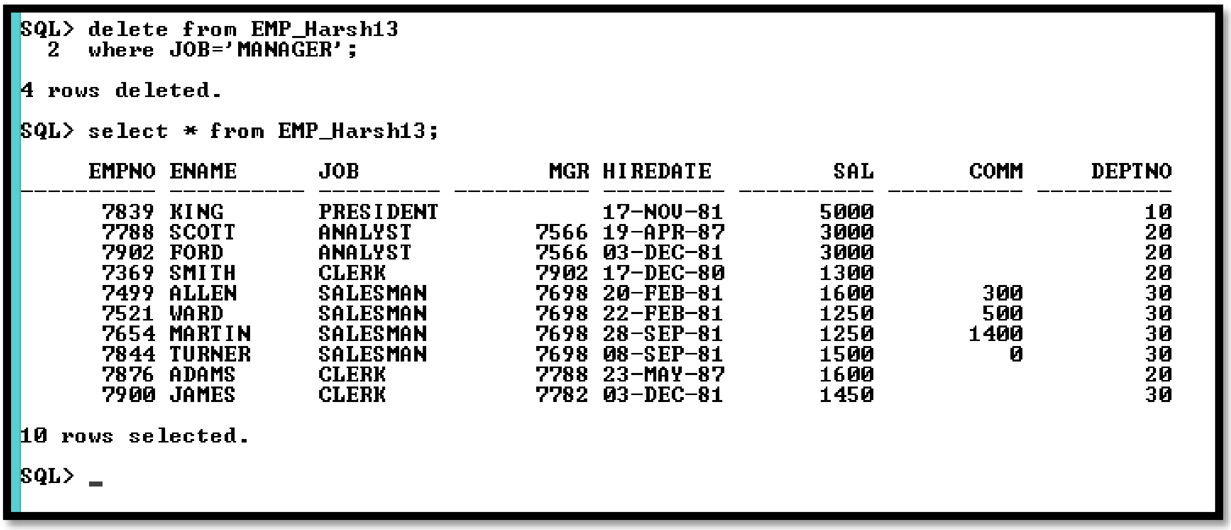
```
SQL> update EMP_Harsh13
  2  set MGR=7782
  3  where ENAME='JAMES';

1 row updated.

SQL> select * from EMP_Harsh13;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80       1300                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1600                    20
      7900 JAMES      CLERK           7782 03-DEC-81       1450                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1800                    10

14 rows selected.

SQL> _
```

3) Change the role of Miller as MANAGER.



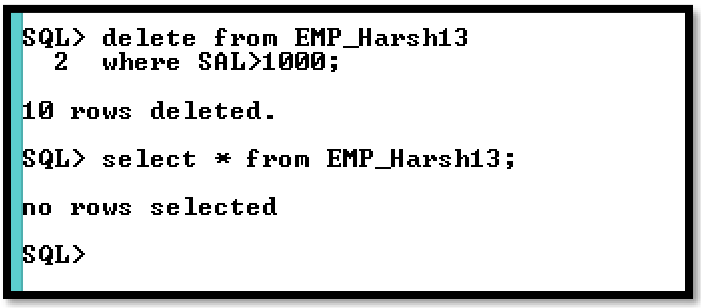
```
SQL> update EMP_Harsh13
  2  set JOB='MANAGER'
  3  where ENAME='MILLER';

1 row updated.

SQL> select * from EMP_Harsh13;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80       1300                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1600                    20
      7900 JAMES      CLERK           7782 03-DEC-81       1450                    30
      7934 MILLER     MANAGER         7782 23-JAN-82       1800                    10

14 rows selected.

SQL>
```

4) Delete the records of Manager.



```
SQL> delete from EMP_Harsh13
  2  where JOB='MANAGER';

4 rows deleted.

SQL> select * from EMP_Harsh13;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80       1300                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1600                    20
      7900 JAMES      CLERK           7782 03-DEC-81       1450                    30

10 rows selected.

SQL> _
```

5) Delete the salary when records is greater than 1000.

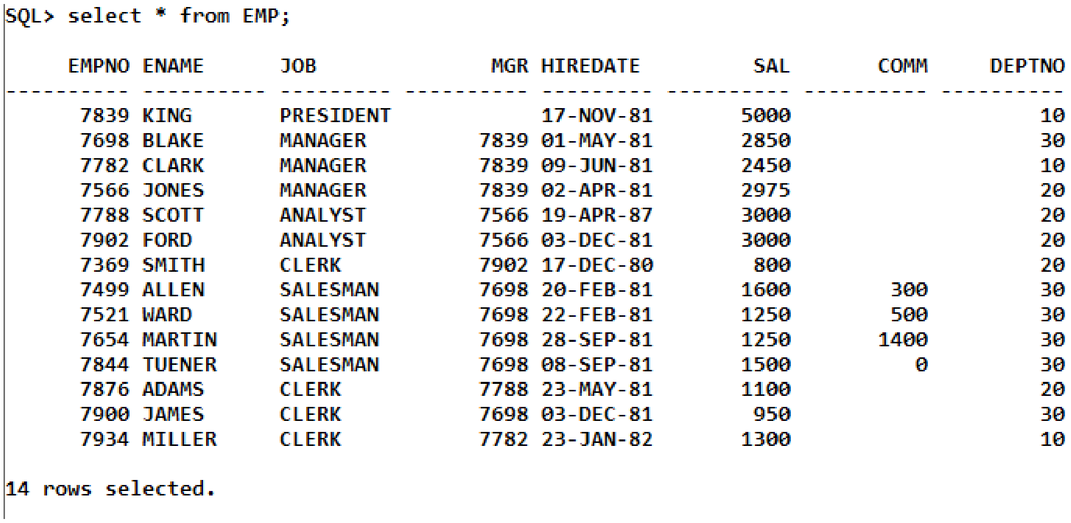


```
SQL> delete from EMP_Harsh13
  2  where SAL>1000;

10 rows deleted.

SQL> select * from EMP_Harsh13;

no rows selected

SQL>
```

DBMS Practical 3

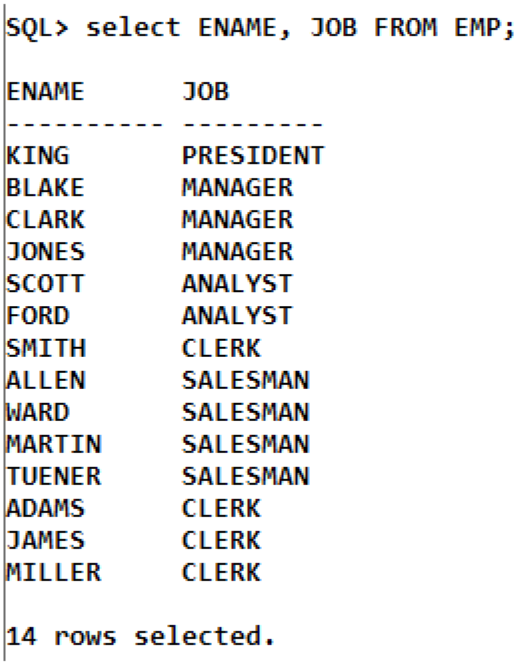
1. Using emp table, perform the following queries:

- Display the details of all employees.



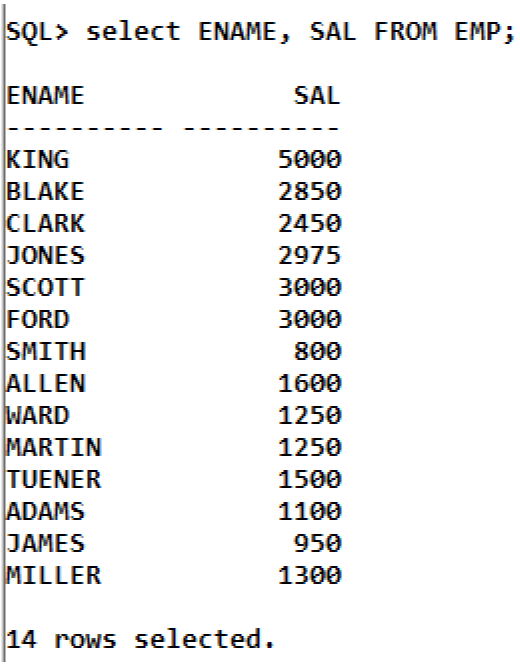
```
SQL> select * from EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

- Display the name and job for all employees.



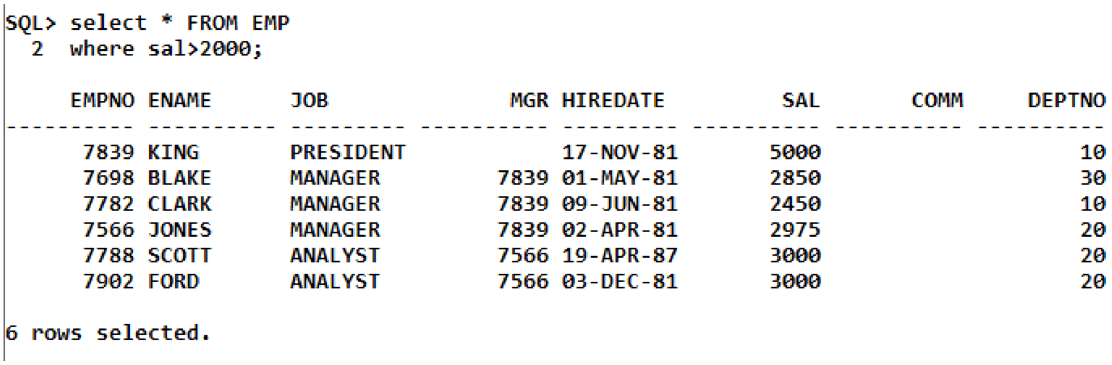
```
SQL> select ENAME, JOB FROM EMP;

ENAME      JOB
---------- ---------
KING       PRESIDENT
BLAKE      MANAGER
CLARK      MANAGER
JONES      MANAGER
SCOTT      ANALYST
FORD       ANALYST
SMITH      CLERK
ALLEN      SALESMAN
WARD       SALESMAN
MARTIN     SALESMAN
TUENER     SALESMAN
ADAMS      CLERK
JAMES      CLERK
MILLER     CLERK

14 rows selected.
```

- Display name and salary for all employees.



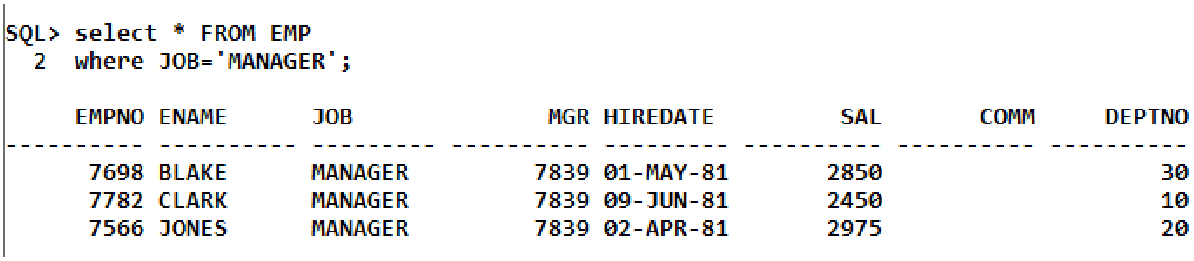
```
SQL> select ENAME, SAL FROM EMP;

ENAME             SAL
---------- ----------
KING             5000
BLAKE            2850
CLARK            2450
JONES            2975
SCOTT            3000
FORD             3000
SMITH             800
ALLEN            1600
WARD             1250
MARTIN           1250
TUENER           1500
ADAMS            1100
JAMES             950
MILLER           1300

14 rows selected.
```

- Display the details of all employees who are earning salary greater than 2000.



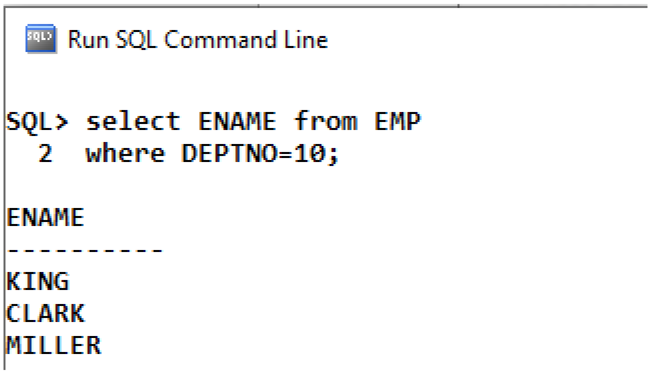
```
SQL> select * FROM EMP
  2  where sal>2000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

6 rows selected.
```

- Display the details of all employees who are working as Manager.



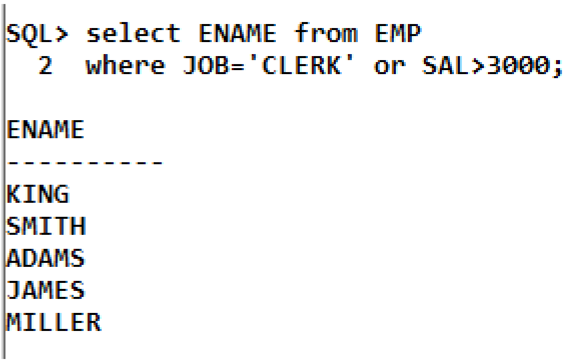
```
SQL> select * FROM EMP
  2  where JOB='MANAGER';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
```

- Display the names of all employees who are working in department number 10.



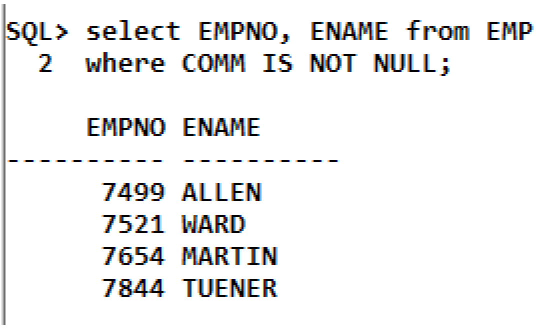
```
SQL> select ENAME from EMP
  2  where DEPTNO=10;

ENAME
----------
KING
CLARK
MILLER
```

- Display the names of all employees working as clerk and drawing a salary more than 3000.



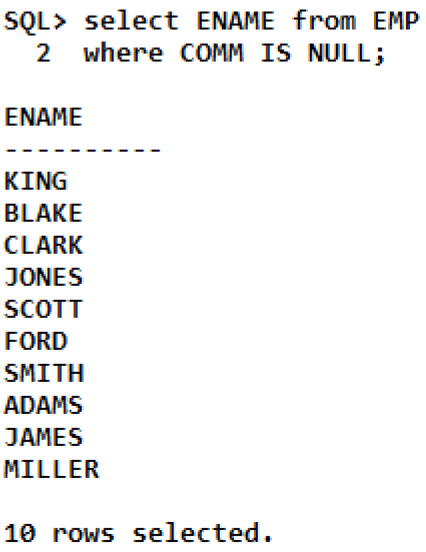
```
SQL> select ENAME from EMP
  2  where JOB='CLERK' or SAL>3000;

ENAME
----------
KING
SMITH
ADAMS
JAMES
MILLER
```

- Display employee number and names for employees who earn commission.



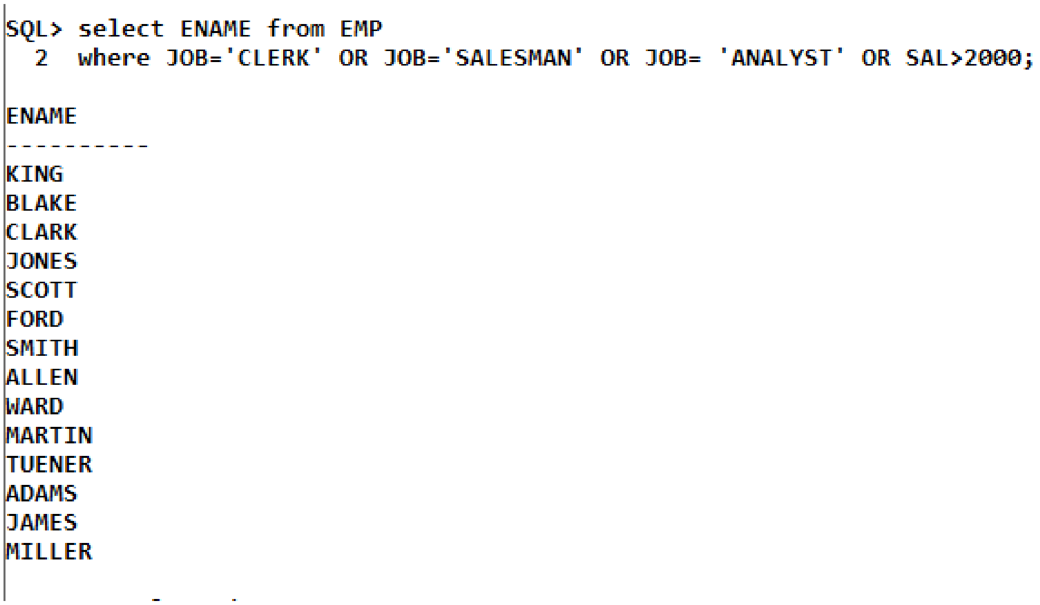
```
SQL> select EMPNO, ENAME from EMP
  2  where COMM IS NOT NULL;

     EMPNO ENAME
---------- ----------
      7499 ALLEN
      7521 WARD
      7654 MARTIN
      7844 TUENER
```

- Display names of employees who do not earn any commission.



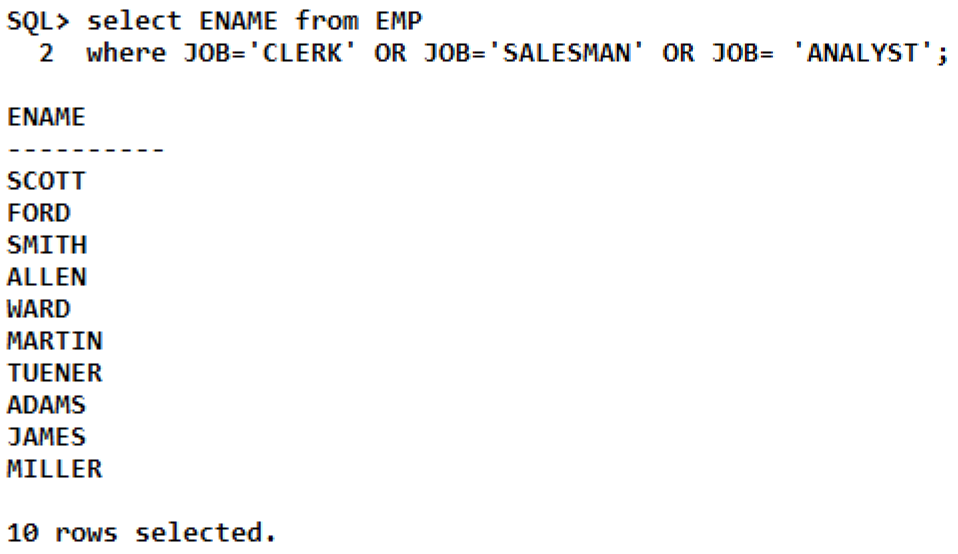
```
SQL> select ENAME from EMP
  2  where COMM IS NULL;

ENAME
----------
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
ADAMS
JAMES
MILLER

10 rows selected.
```

- Display the names of employees who are working as clerk, salesman or analyst and drawing a salary more than 2000.



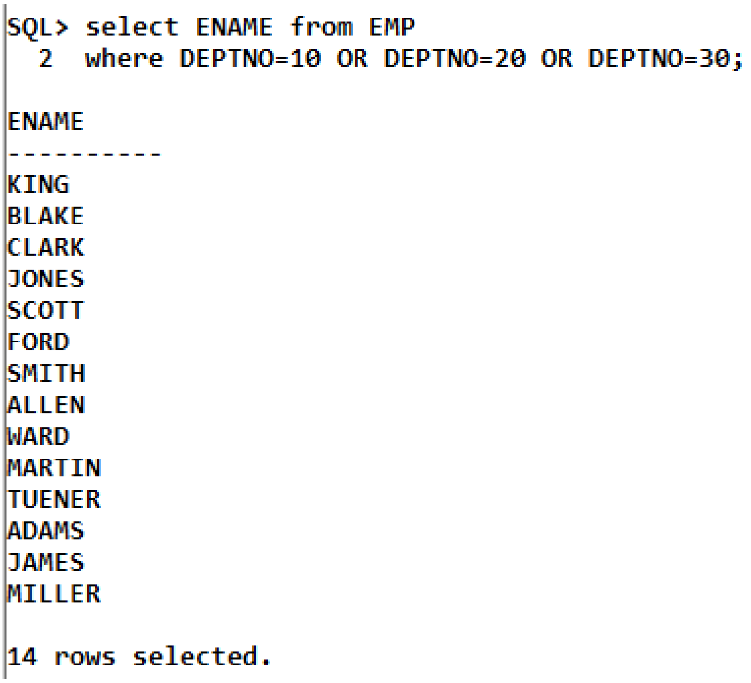
```
SQL> select ENAME from EMP
  2  where JOB='CLERK' OR JOB='SALESMAN' OR JOB= 'ANALYST' OR SAL>2000;

ENAME
----------
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
ALLEN
WARD
MARTIN
TUENER
ADAMS
JAMES
MILLER
```

- Display the names of employees who are working as clerk, salesman or analyst.



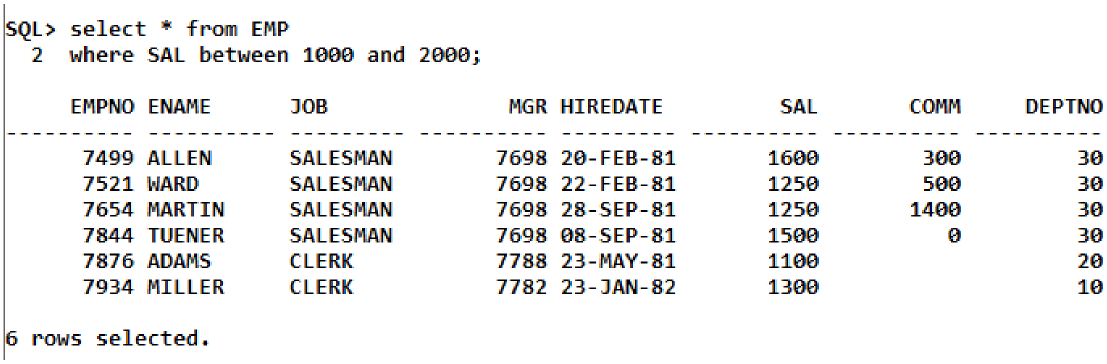
```
SQL> select ENAME from EMP
  2  where JOB='CLERK' OR JOB='SALESMAN' OR JOB= 'ANALYST';

ENAME
----------
SCOTT
FORD
SMITH
ALLEN
WARD
MARTIN
TUENER
ADAMS
JAMES
MILLER

10 rows selected.
```

- Display the names of employees working in department number 10 or 20 or 30.



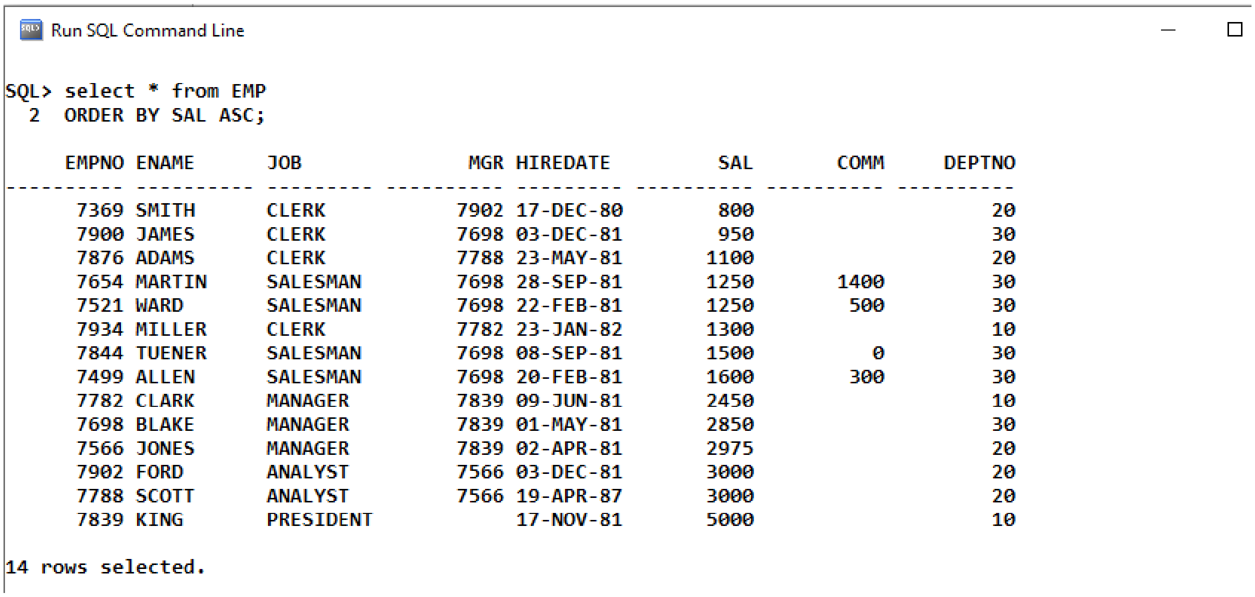
```
SQL> select ENAME from EMP
  2  where DEPTNO=10 OR DEPTNO=20 OR DEPTNO=30;

ENAME
----------
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
ALLEN
WARD
MARTIN
TUENER
ADAMS
JAMES
MILLER

14 rows selected.
```

- Display the details of employees whose salary lies in the range of 1000 and 2000.



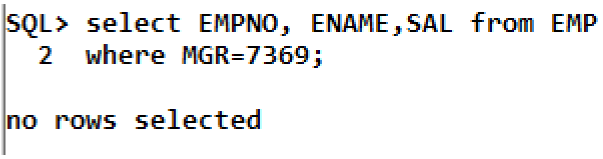
```
SQL> select * from EMP
  2  where SAL between 1000 and 2000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

6 rows selected.
```

- List the employees in the ascending order of their salaries.



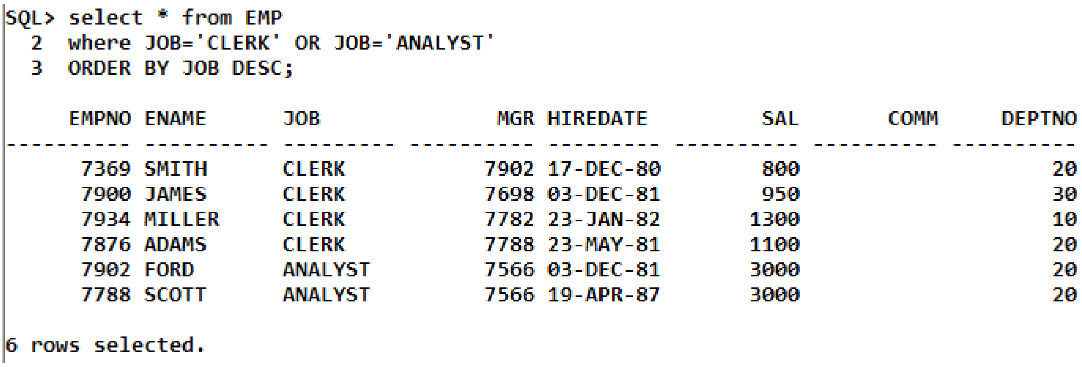
```
SQL> select * from EMP
  2  ORDER BY SAL ASC;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10

14 rows selected.
```

- List the Empno, Ename, Sal of all emps working for Mgr 7369.



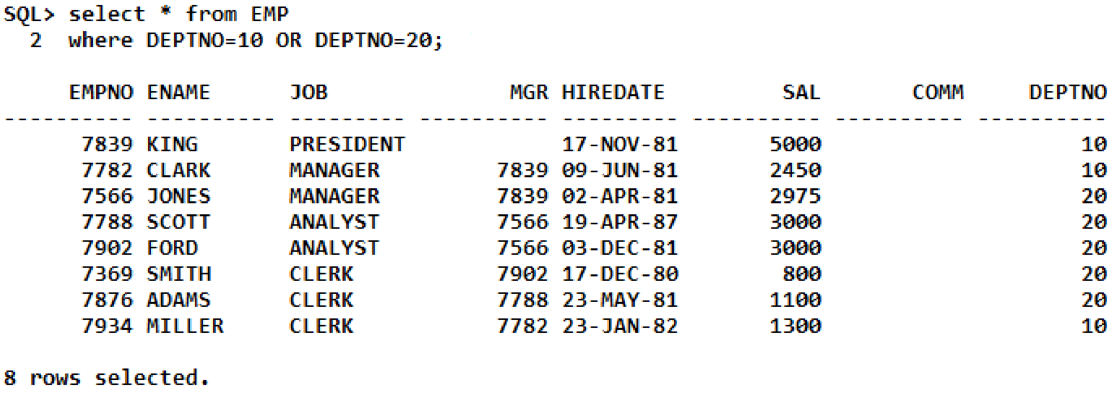
```
SQL> select EMPNO, ENAME,SAL from EMP
  2  where MGR=7369;

no rows selected
```

- List the employees who are either ‘CLERK’ or ‘ANALYST’ in the Desc order.



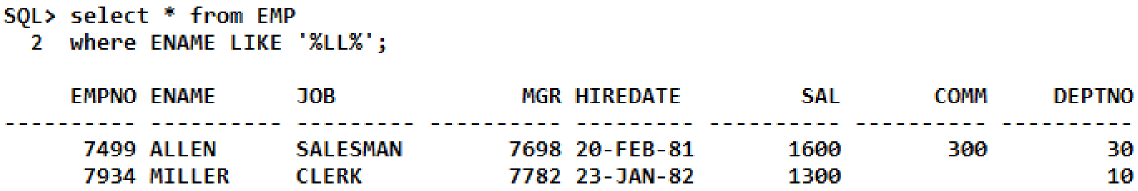
```
SQL> select * from EMP
  2  where JOB='CLERK' OR JOB='ANALYST'
  3  ORDER BY JOB DESC;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20

6 rows selected.
```

- List the employees who are working in Deptno 10 or 20.



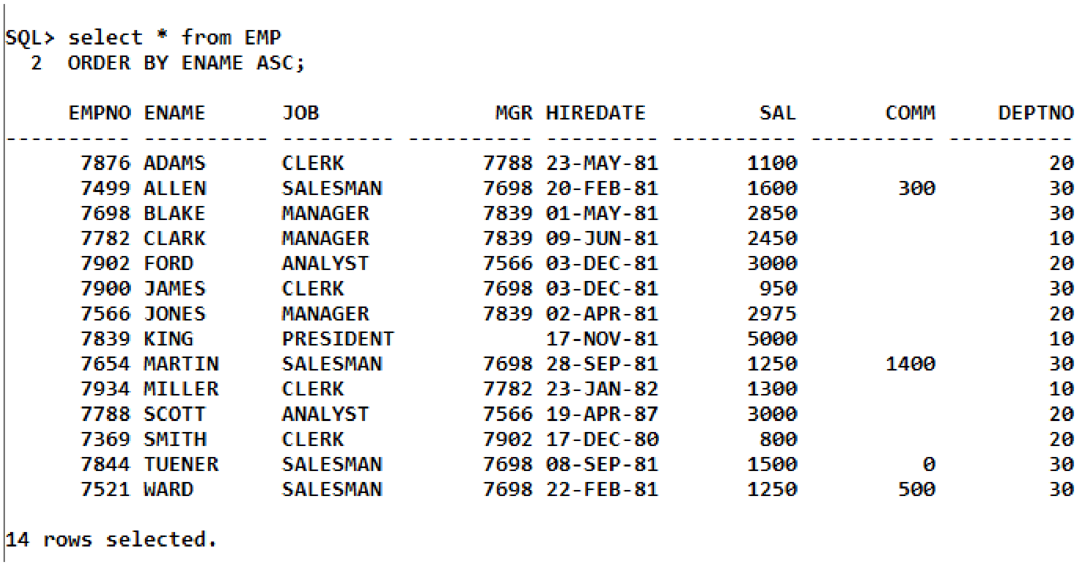
```
SQL> select * from EMP
  2  where DEPTNO=10 OR DEPTNO=20;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

8 rows selected.
```

- List the employees whose name have a character set ‘ll’ together.



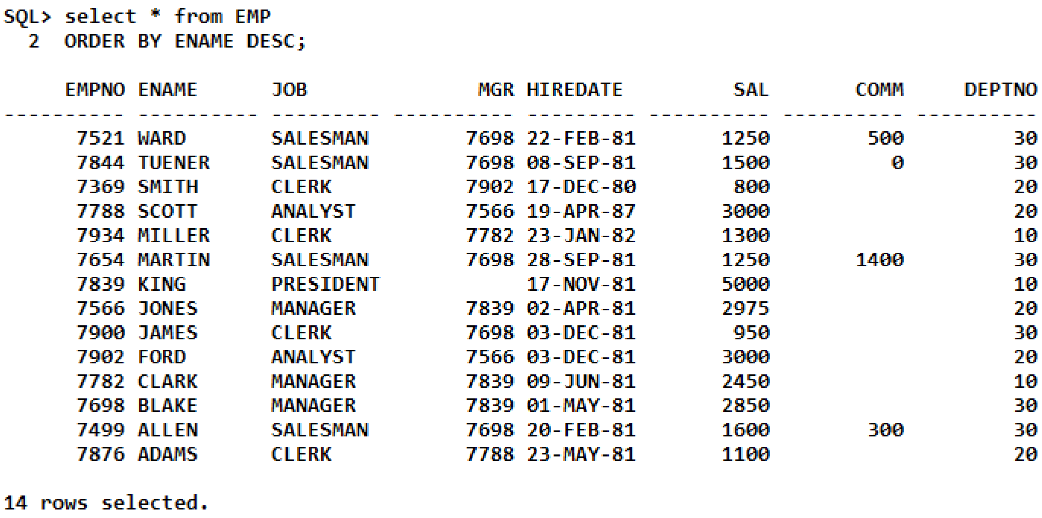
```
SQL> select * from EMP
  2  where ENAME LIKE '%LL%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

- List the employees in ascending order of their names.



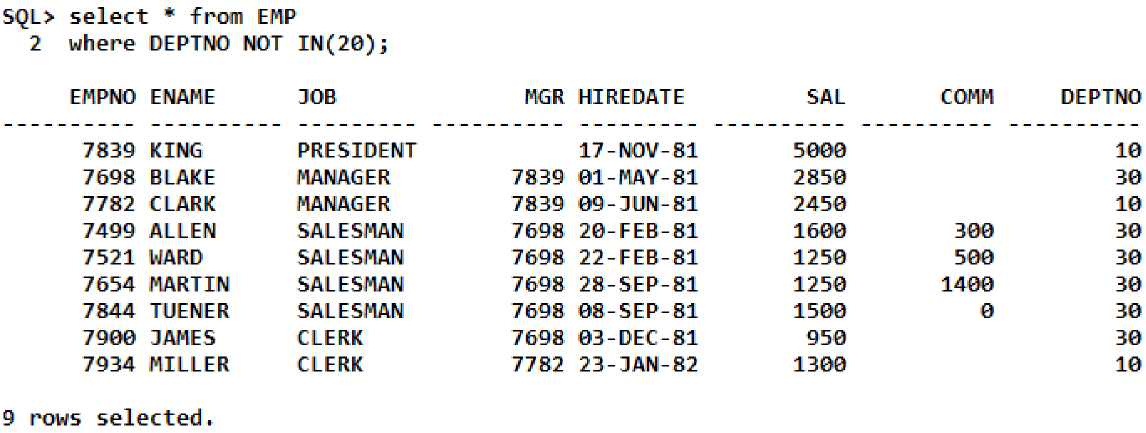
```
SQL> select * from EMP
  2  ORDER BY ENAME ASC;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30

14 rows selected.
```

- List the employees in descending order of their names.



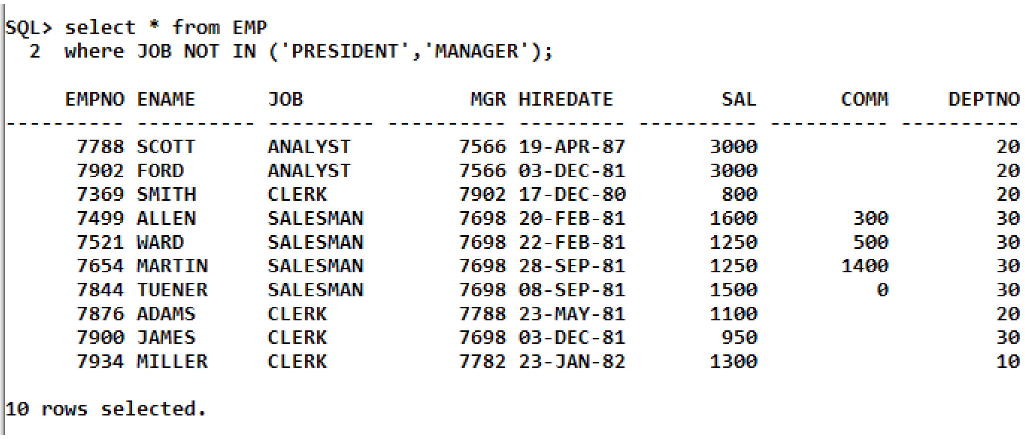
```
SQL> select * from EMP
  2  ORDER BY ENAME DESC;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20

14 rows selected.
```

- List the employees who do not belong to Deptno 20.



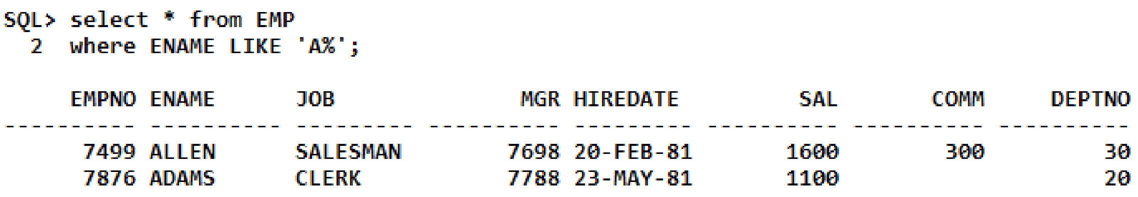
```
SQL> select * from EMP
  2  where DEPTNO NOT IN(20);

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

9 rows selected.
```

- List all the employees except PRESIDENT and MANAGER.



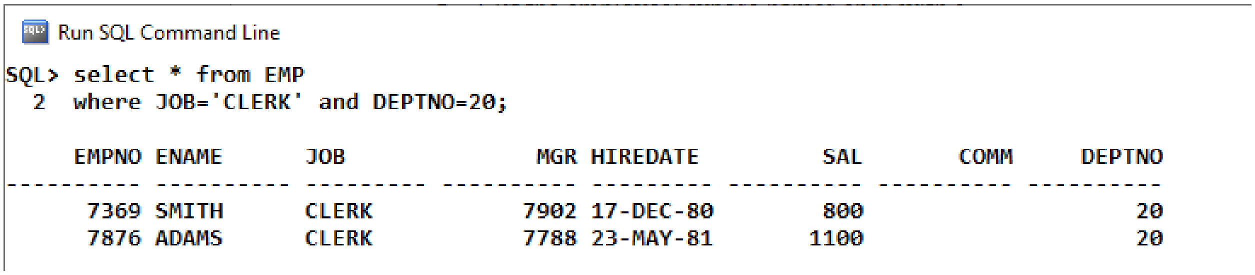
```
SQL> select * from EMP
  2  where JOB NOT IN ('PRESIDENT','MANAGER');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

10 rows selected.
```

- List the employees whose name starts with A.



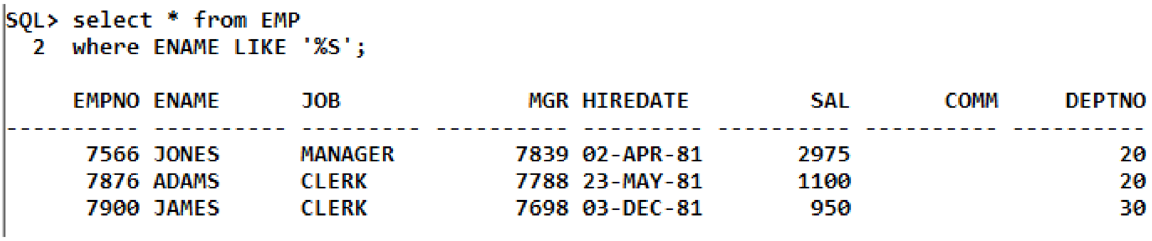
```
SQL> select * from EMP
  2  where ENAME LIKE 'A%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
```

- List all the Clerks of Deptno 20.



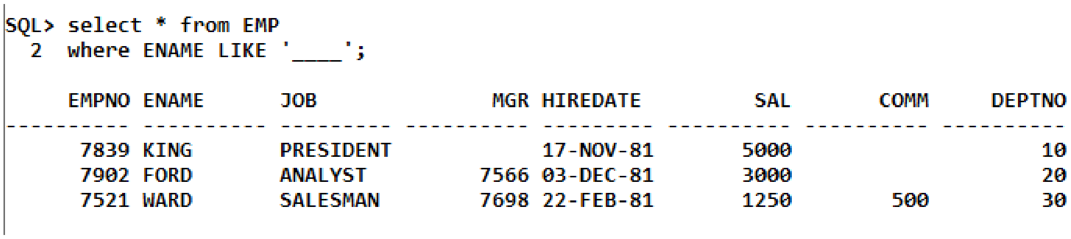
```
SQL> select * from EMP
  2  where JOB='CLERK' and DEPTNO=20;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
```

- List the employees whose names ends with S.



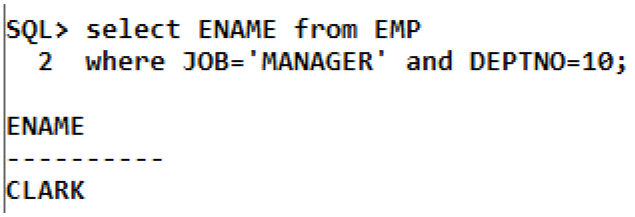
```
SQL> select * from EMP
  2  where ENAME LIKE '%S';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
```

- List the employees who has name of exactly 4 characters.



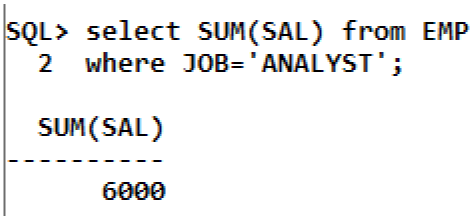
```
SQL> select * from EMP
  2  where ENAME LIKE '____';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
```

- List the names of the employees who are working as MANAGER in department 10.



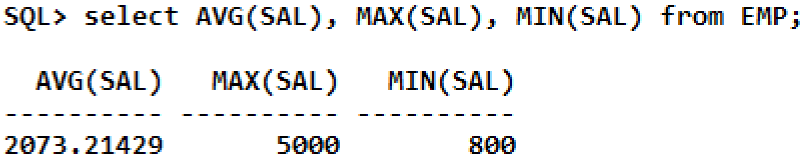
```
SQL> select ENAME from EMP
  2  where JOB='MANAGER' and DEPTNO=10;

ENAME
----------
CLARK
```

- List the total salary of employees working as ANALYST.



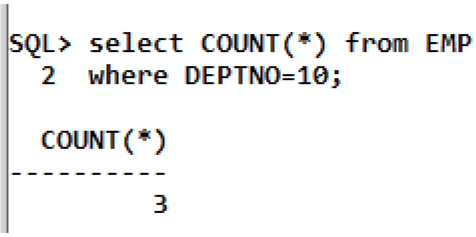
```
SQL> select SUM(SAL) from EMP
  2  where JOB='ANALYST';

  SUM(SAL)
----------
      6000
```

- List the minimum, maximum and average salary of the employees.



```
SQL> select AVG(SAL), MAX(SAL), MIN(SAL) from EMP;

  AVG(SAL)   MAX(SAL)   MIN(SAL)
---------- ---------- ----------
2073.21429       5000        800
```

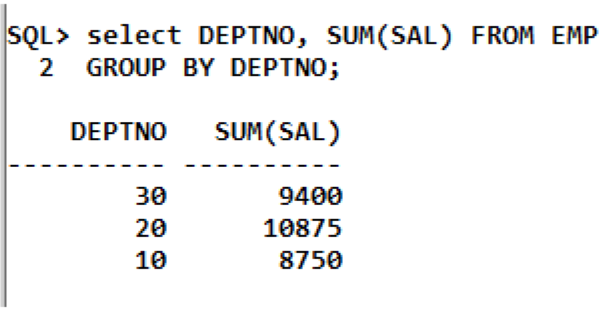
- List the total number of employees working in department 10.



```
SQL> select COUNT(*) from EMP
  2  where DEPTNO=10;

  COUNT(*)
----------
         3
```

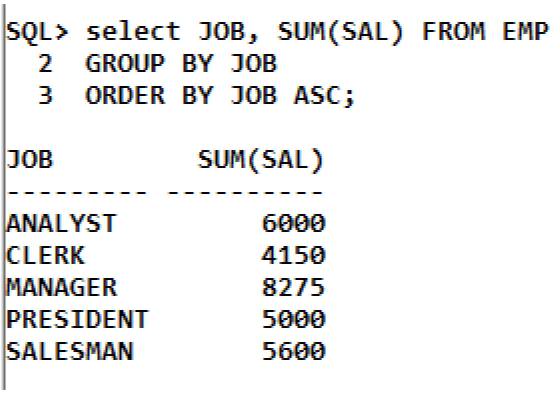
2. Answer the following queries:

- Display the total salary of employees department wise.



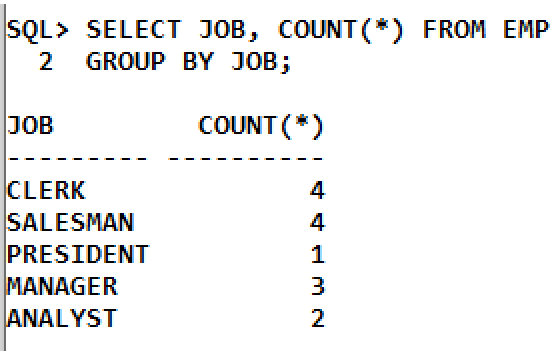
```
SQL> select DEPTNO, SUM(SAL) FROM EMP
  2  GROUP BY DEPTNO;

    DEPTNO   SUM(SAL)
---------- ----------
        30       9400
        20      10875
        10       8750
```

- Display the total salary of employees job wise in ascending order of job.



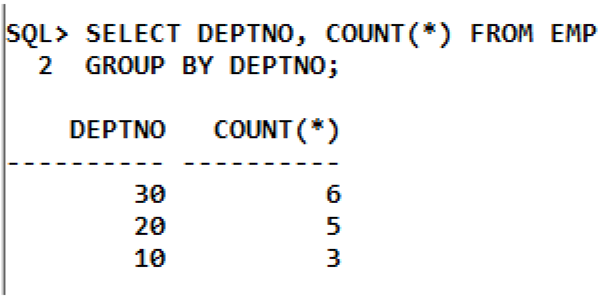
```
SQL> select JOB, SUM(SAL) FROM EMP
  2  GROUP BY JOB
  3  ORDER BY JOB ASC;

JOB         SUM(SAL)
--------- ----------
ANALYST         6000
CLERK           4150
MANAGER         8275
PRESIDENT       5000
SALESMAN        5600
```

- Display the total number of employees with specific job.



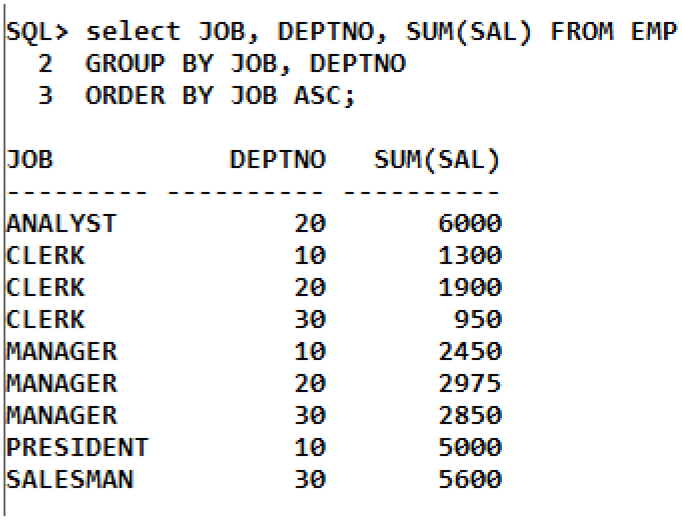
```
SQL> SELECT JOB, COUNT(*) FROM EMP
  2  GROUP BY JOB;

JOB         COUNT(*)
--------- ----------
CLERK              4
SALESMAN           4
PRESIDENT          1
MANAGER            3
ANALYST            2
```

- Display the total number of employees working in each department.



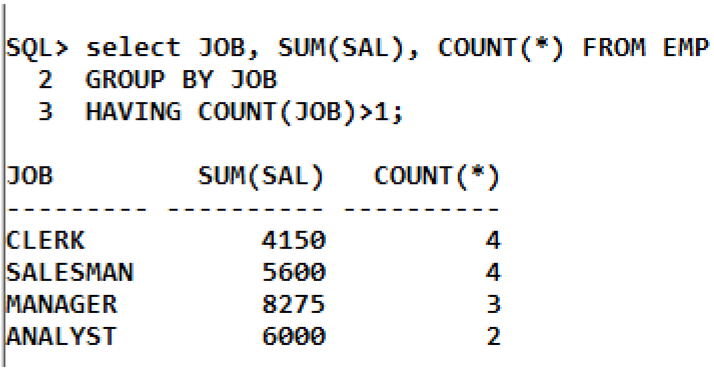
```
SQL> SELECT DEPTNO, COUNT(*) FROM EMP
  2  GROUP BY DEPTNO;

    DEPTNO   COUNT(*)
---------- ----------
        30          6
        20          5
        10          3
```

- Display the total salary of employees specific to job and department in ascending order of job.



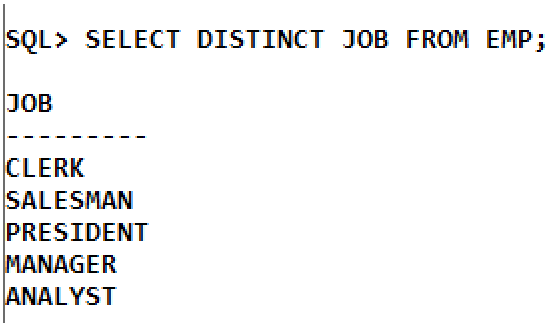
```
SQL> select JOB, DEPTNO, SUM(SAL) FROM EMP
  2  GROUP BY JOB, DEPTNO
  3  ORDER BY JOB ASC;

JOB           DEPTNO   SUM(SAL)
--------- ---------- ----------
ANALYST           20       6000
CLERK             10       1300
CLERK             20       1900
CLERK             30        950
MANAGER           10       2450
MANAGER           20       2975
MANAGER           30       2850
PRESIDENT         10       5000
SALESMAN          30       5600
```

- Display the total salary of the employees specific to job when employee count is greater than 1.



```
SQL> select JOB, SUM(SAL), COUNT(*) FROM EMP
  2  GROUP BY JOB
  3  HAVING COUNT(JOB)>1;

JOB         SUM(SAL)   COUNT(*)
--------- ---------- ----------
CLERK           4150          4
SALESMAN        5600          4
MANAGER         8275          3
ANALYST         6000          2
```

- Display unique jobs of employees.



```
SQL> SELECT DISTINCT JOB FROM EMP;

JOB
---------
CLERK
SALESMAN
PRESIDENT
MANAGER
ANALYST
```

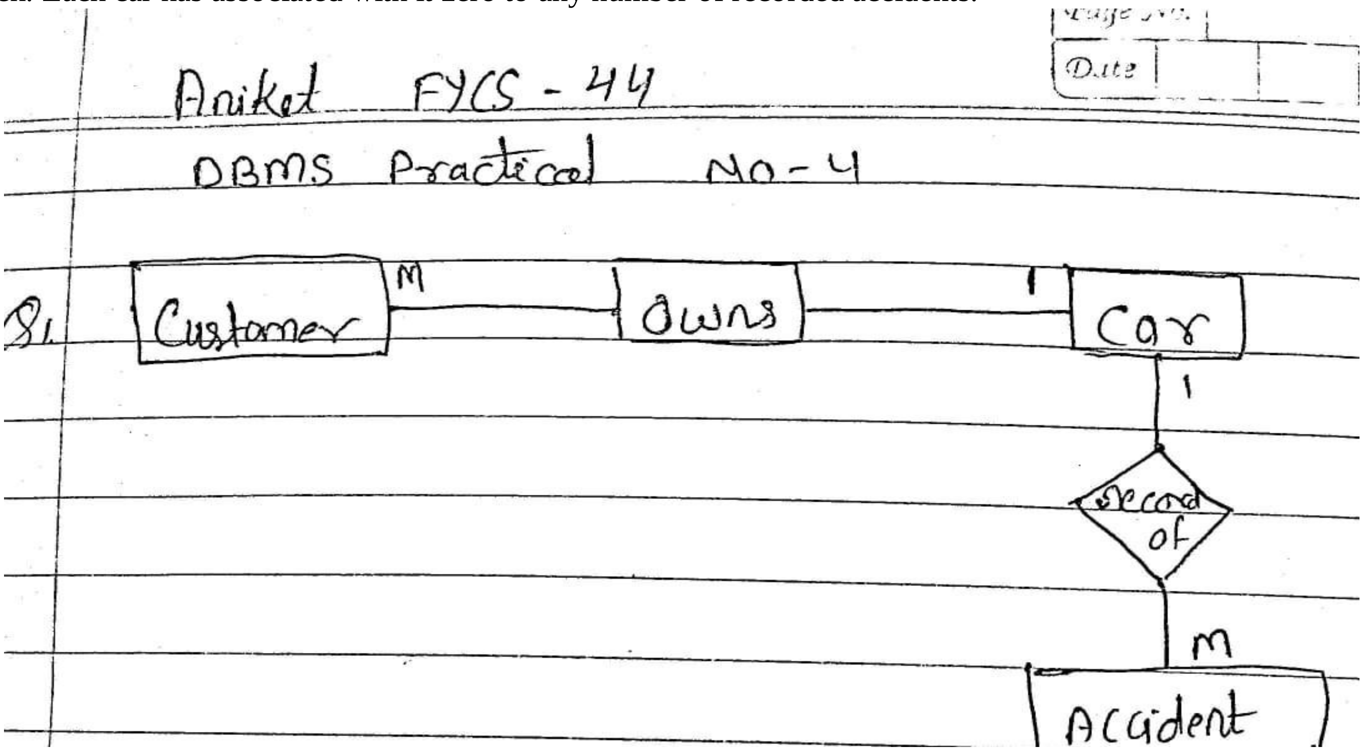
Aniket Prajapati

FYCS - 44

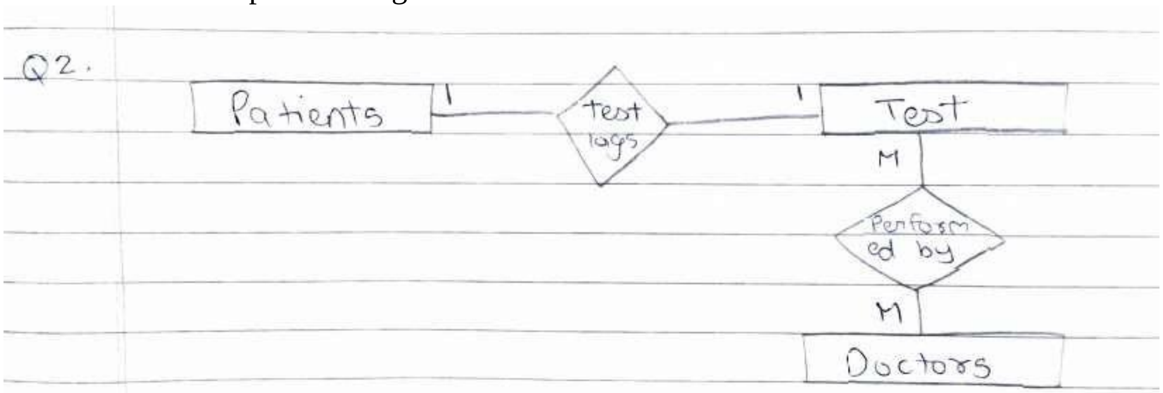
PRACTICAL NO. 4

Questions on ER diagram

1. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.



2. Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

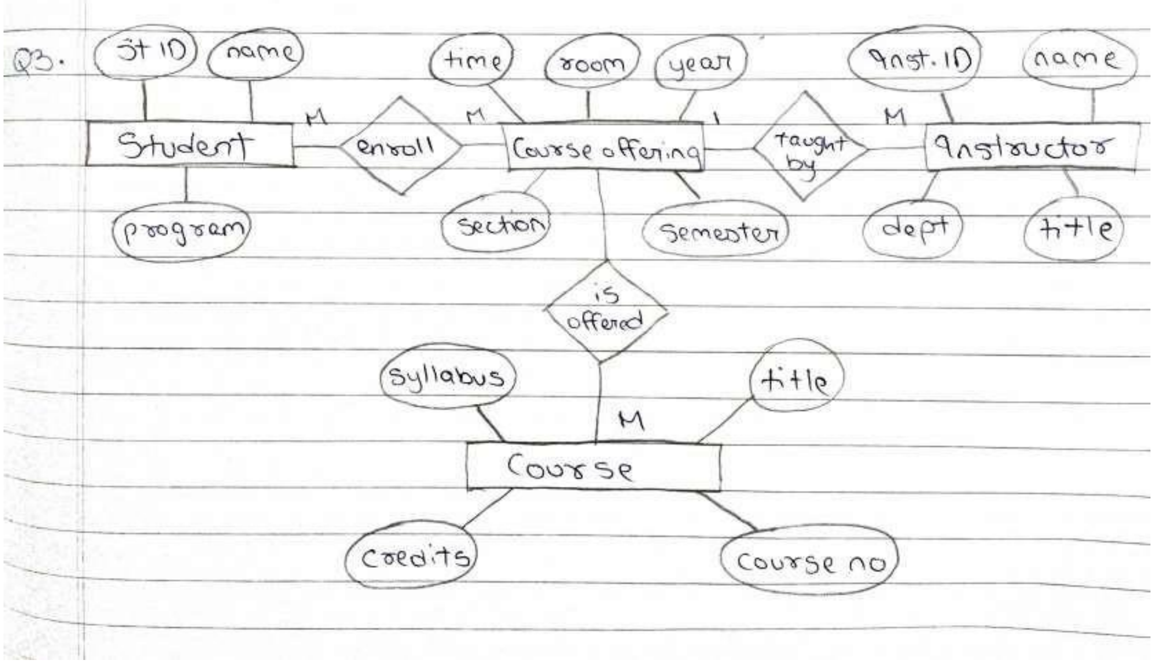


3. A university registrar.s of_ce maintains data about the following entities: (a)courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including

student-id, name, and program; and (d) instructors, including identi-cation number, name, department, and title.

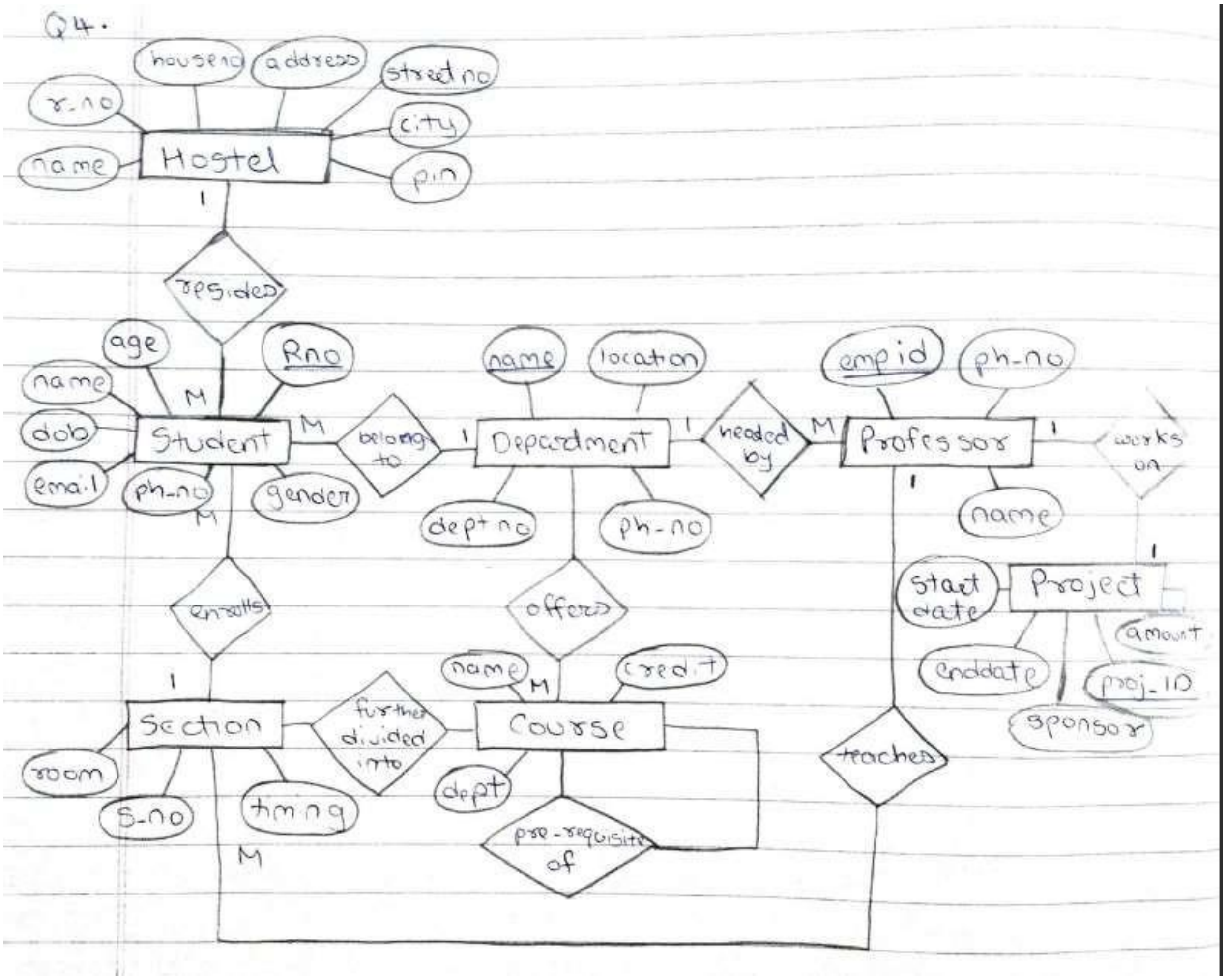
Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled.

Construct an E-R diagram for the registrar.s of_ce.Document all assumptions that you make about the mapping constraints.



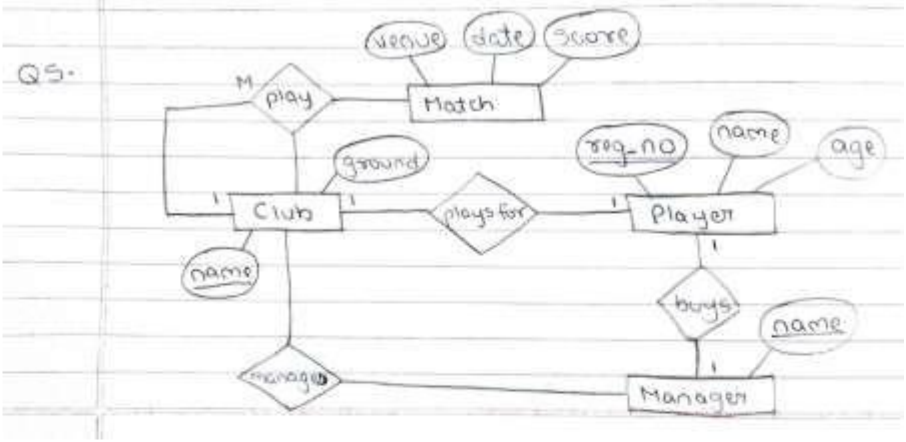
4. Draw the ER diagram for the given scenario

- In an educational institute, there are several departments and students belong to one of them. Each department has a unique department number, a unique name, a location, a phone number and is headed by a professor.
- Professors have a unique employee id, name, phoneno. We like to keep track of the following details regarding students: name, unique roll no, gender, phone number, date of birth, age and one or more email addresses.
- Students have a local address consisting of the hostel name and the room number. They also have home address consisting of house number, street, city and pin. It is assumed that all students reside in the hostels.
- A course taught in a semester of the year is called a section. There can be several sections of the same course in a semester; these are identified by the section number. Each section is taught by a different professor and has its own timings and a room to meet.
- Students enroll for several sections in a semester. Each course has a name, number of credits and the department that offers it. A course may have other courses as pre-requisites i.e courses to be completed before it can be enrolled in.
- Professors also undertake research projects. These are sponsored by funding agencies and have a specific start date, end date and amount of money given. More than one professor can be involved in a project. Also a professor may be simultaneously working on several projects. A project has a unique projectid.



5. Draw the ER diagram for the given scenario.

“A football club has a name and a ground and is made up of players. A player can play for only one club and a manager, represented by his name manages a club. A footballer has a registration number, name and age. A club manager also buys players. Each club plays against each other club in the league and matches have a date, venue and score.



PRACTICAL NO. 5

1. **INNER JOIN:** The INNER JOIN keyword selects all rows from both the tables as long as the satisfies.

SYNTAX:

SELECT table1.column1,table1.column2,table2.column1,….

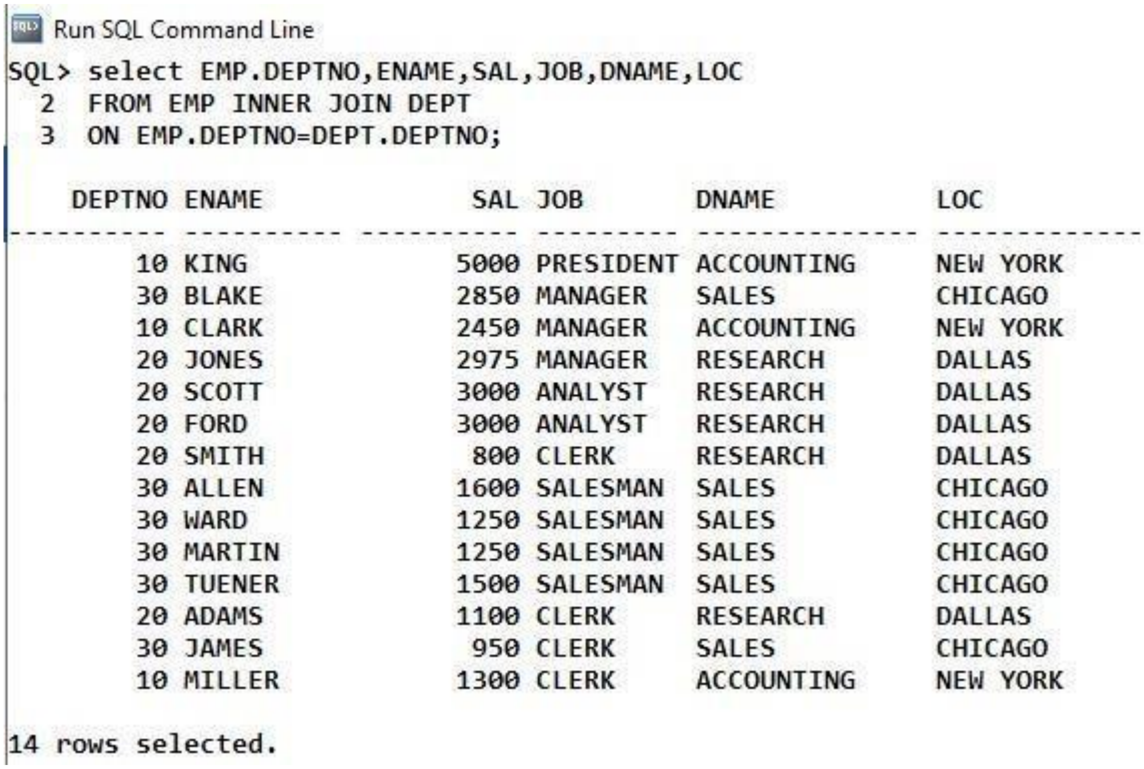
FROM table1 INNER JOIN table2

ON table1.matching_column=table2.matching_column; Example:

SELECT EMP.DEPTNO,ENAME, SAL,JOB,DNAME,LOC

FROM EMP INNER JOIN DEPT

ON EMP.DEPTNO=DEPT.DEPTNO;



```
SQL> select EMP.DEPTNO,ENAME,SAL,JOB,DNAME,LOC
  2  FROM EMP INNER JOIN DEPT
  3  ON EMP.DEPTNO=DEPT.DEPTNO;

    DEPTNO ENAME             SAL JOB       DNAME          LOC
---------- ---------- ---------- --------- -------------- -------------
        10 KING             5000 PRESIDENT ACCOUNTING     NEW YORK
        30 BLAKE            2850 MANAGER   SALES          CHICAGO
        10 CLARK            2450 MANAGER   ACCOUNTING     NEW YORK
        20 JONES            2975 MANAGER   RESEARCH       DALLAS
        20 SCOTT            3000 ANALYST   RESEARCH       DALLAS
        20 FORD             3000 ANALYST   RESEARCH       DALLAS
        20 SMITH             800 CLERK     RESEARCH       DALLAS
        30 ALLEN            1600 SALESMAN  SALES          CHICAGO
        30 WARD             1250 SALESMAN  SALES          CHICAGO
        30 MARTIN           1250 SALESMAN  SALES          CHICAGO
        30 TUENER           1500 SALESMAN  SALES          CHICAGO
        20 ADAMS            1100 CLERK     RESEARCH       DALLAS
        30 JAMES             950 CLERK     SALES          CHICAGO
        10 MILLER           1300 CLERK     ACCOUNTING     NEW YORK

14 rows selected.
```

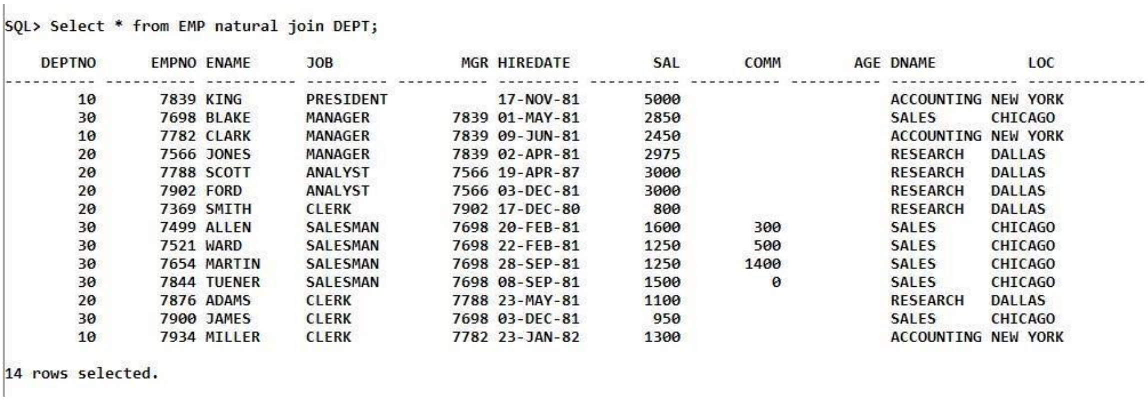
2. **NATURAL JOIN:** A Natural Join is a type equi join which occurs implicitly by comparing all the same names columns in both tables. The join result has only one column for each pair of equally named columns

SYNTAX:

select * From table1 natural join table2;

Example:

Select * from EMP natural join DEPT;



```
SQL> Select * from EMP natural join DEPT;

    DEPTNO      EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM        AGE DNAME      LOC
---------- ---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ---------- -------------
        10       7839 KING       PRESIDENT            17-NOV-81       5000                       ACCOUNTING NEW YORK
        30       7698 BLAKE      MANAGER         7839 01-MAY-81       2850                       SALES      CHICAGO
        10       7782 CLARK      MANAGER         7839 09-JUN-81       2450                       ACCOUNTING NEW YORK
        20       7566 JONES      MANAGER         7839 02-APR-81       2975                       RESEARCH   DALLAS
        20       7788 SCOTT      ANALYST         7566 19-APR-87       3000                       RESEARCH   DALLAS
        20       7902 FORD       ANALYST         7566 03-DEC-81       3000                       RESEARCH   DALLAS
        20       7369 SMITH      CLERK           7902 17-DEC-80        800                       RESEARCH   DALLAS
        30       7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300            SALES      CHICAGO
        30       7521 WARD       SALESMAN        7698 22-FEB-81       1250        500            SALES      CHICAGO
        30       7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400            SALES      CHICAGO
        30       7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0            SALES      CHICAGO
        20       7876 ADAMS      CLERK           7788 23-MAY-81       1100                       RESEARCH   DALLAS
        30       7900 JAMES      CLERK           7698 03-DEC-81        950                       SALES      CHICAGO
        10       7934 MILLER     CLERK           7782 23-JAN-82       1300                       ACCOUNTING NEW YORK

14 rows selected.
```

3. OUTER JOIN:

Theta Join, Equijoin, and Natural Join are called inner joins. An inner join includes only those tuples with matching attributes and the rest are discarded in the resulting relation. Therefore, we need to use outer joins to include all the tuples from the participating relations in the resulting relation. There are three kinds of outer joins − left outer join, right outer join, and full outer join.

A) RIGHT OUTER JOIN:

SYNTAX:

SELECT table1.column1, table2.column2....

FROM table1

RIGHT JOIN table2

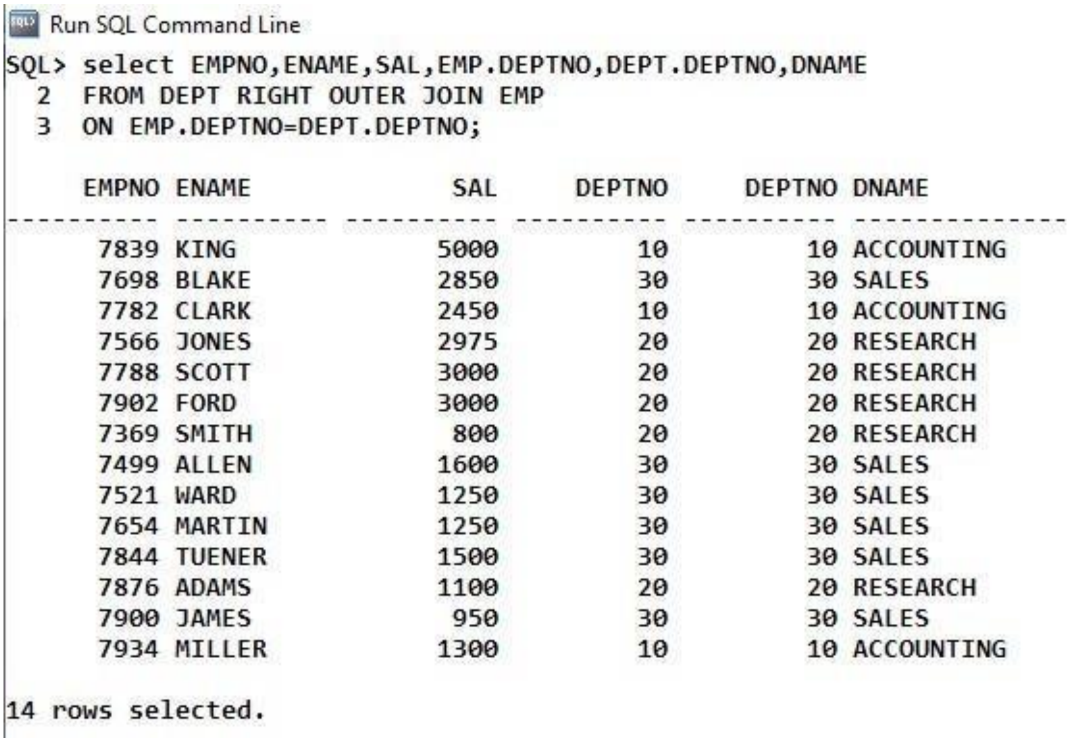
ON table1.column_field = table2.column_field;

EXAMPLE:

SELECT EMPNO,ENAMESAL,EMP.DEPTNO,DEPT.DEPTNO,DNAME

FROM DEPT RIGHT OUTER JOIN EMP

ON EMP.DEPTNO=DEPT.DEPTNO;



```
SQL> select EMPNO,ENAME,SAL,EMP.DEPTNO,DEPT.DEPTNO,DNAME
  2  FROM DEPT RIGHT OUTER JOIN EMP
  3  ON EMP.DEPTNO=DEPT.DEPTNO;

     EMPNO ENAME             SAL     DEPTNO     DEPTNO DNAME
---------- ---------- ---------- ---------- ---------- --------------
      7839 KING             5000         10         10 ACCOUNTING
      7698 BLAKE            2850         30         30 SALES
      7782 CLARK            2450         10         10 ACCOUNTING
      7566 JONES            2975         20         20 RESEARCH
      7788 SCOTT            3000         20         20 RESEARCH
      7902 FORD             3000         20         20 RESEARCH
      7369 SMITH             800         20         20 RESEARCH
      7499 ALLEN            1600         30         30 SALES
      7521 WARD             1250         30         30 SALES
      7654 MARTIN           1250         30         30 SALES
      7844 TUENER           1500         30         30 SALES
      7876 ADAMS            1100         20         20 RESEARCH
      7900 JAMES             950         30         30 SALES
      7934 MILLER           1300         10         10 ACCOUNTING

14 rows selected.
```

B) LEFT OUTER JOIN :

SYNTAX:

SELECT table1.column1, table2.column2....

FROM table1

LEFT JOIN table2

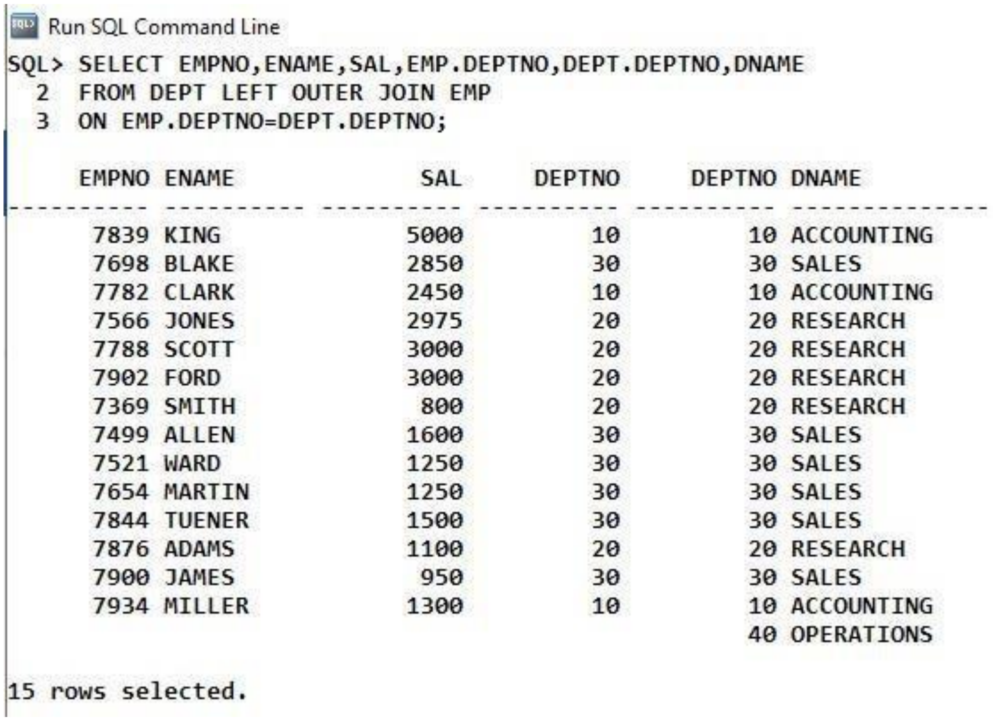
ON table1.column_field = table2.column_field;

EXAMPLE:

SELECT EMPNO,ENAME,SAL,EMP.DEPTNO,DEPT.DEPTNO,DNAME

FROM DEPT LEFT OUTER JOIN EMP

ON EMP.DEPTNO=DEPT.DEPTNO;



```
SQL> SELECT EMPNO,ENAME,SAL,EMP.DEPTNO,DEPT.DEPTNO,DNAME
  2  FROM DEPT LEFT OUTER JOIN EMP
  3  ON EMP.DEPTNO=DEPT.DEPTNO;

     EMPNO ENAME             SAL     DEPTNO     DEPTNO DNAME
---------- ---------- ---------- ---------- ---------- --------------
      7839 KING             5000         10         10 ACCOUNTING
      7698 BLAKE            2850         30         30 SALES
      7782 CLARK            2450         10         10 ACCOUNTING
      7566 JONES            2975         20         20 RESEARCH
      7788 SCOTT            3000         20         20 RESEARCH
      7902 FORD             3000         20         20 RESEARCH
      7369 SMITH             800         20         20 RESEARCH
      7499 ALLEN            1600         30         30 SALES
      7521 WARD             1250         30         30 SALES
      7654 MARTIN           1250         30         30 SALES
      7844 TUENER           1500         30         30 SALES
      7876 ADAMS            1100         20         20 RESEARCH
      7900 JAMES             950         30         30 SALES
      7934 MILLER           1300         10         10 ACCOUNTING
                                                    40 OPERATIONS

15 rows selected.
```

C) FULL OUTER JOIN:

SYNTAX:

SELECT table1.column1, table2.column2....

FROM table1

FULL JOIN table2

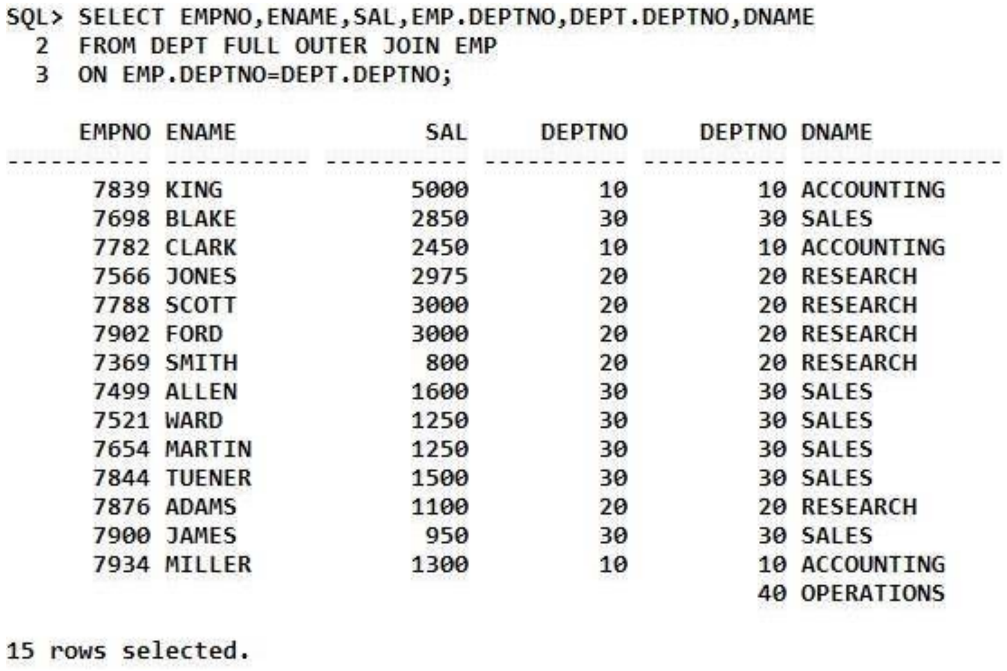
ON table1.column_field = table2.column_field;

EXAMPLE:

SELECT EMPNO,ENAME,SAL,EMP.DEPTNO,DEPT.DEPTNO,DNAME

FROM DEPT FULL OUTER JOIN EMP

ON EMP.DEPTNO=DEPT.DEPTNO;



```
SQL> SELECT EMPNO,ENAME,SAL,EMP.DEPTNO,DEPT.DEPTNO,DNAME
  2  FROM DEPT FULL OUTER JOIN EMP
  3  ON EMP.DEPTNO=DEPT.DEPTNO;

     EMPNO ENAME             SAL     DEPTNO     DEPTNO DNAME
---------- ---------- ---------- ---------- ---------- --------------
      7839 KING             5000         10         10 ACCOUNTING
      7698 BLAKE            2850         30         30 SALES
      7782 CLARK            2450         10         10 ACCOUNTING
      7566 JONES            2975         20         20 RESEARCH
      7788 SCOTT            3000         20         20 RESEARCH
      7902 FORD             3000         20         20 RESEARCH
      7369 SMITH             800         20         20 RESEARCH
      7499 ALLEN            1600         30         30 SALES
      7521 WARD             1250         30         30 SALES
      7654 MARTIN           1250         30         30 SALES
      7844 TUENER           1500         30         30 SALES
      7876 ADAMS            1100         20         20 RESEARCH
      7900 JAMES             950         30         30 SALES
      7934 MILLER           1300         10         10 ACCOUNTING
                                                    40 OPERATIONS

15 rows selected.
```

4. **CROSS JOIN:** When each row of first table is combined with each row from the second table, known as Cartesian join or cross join.

SYNTAX:

SELECT * FROM TABLE1 CROSS JOIN TABLE2

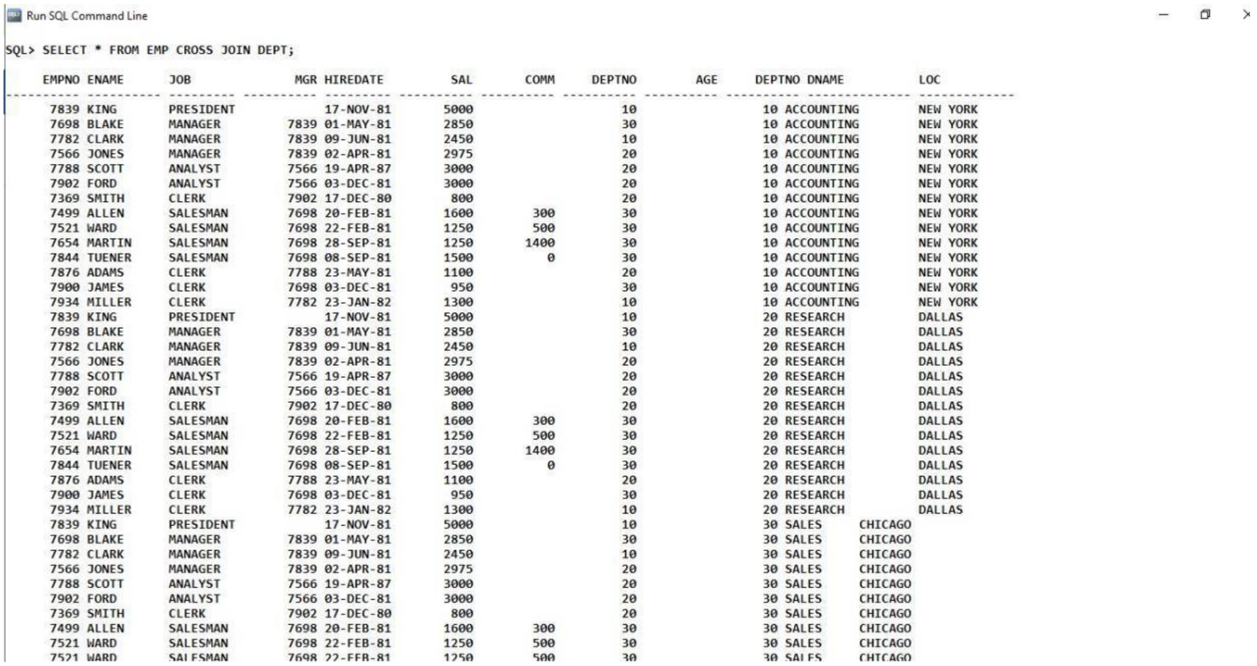
OR

SELECT * FROM TABLE1 , TABLE2 EXAMPLE:

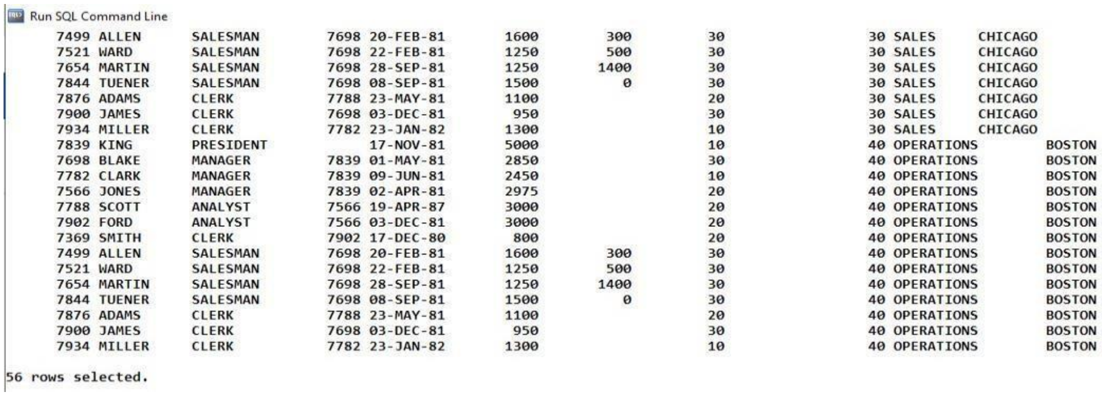
SELECT * FROM EMP CROSS JOIN DEPT;

OR

SELECT * FROM EMP, DEPT



```
SQL> SELECT * FROM EMP CROSS JOIN DEPT;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO        AGE     DEPTNO DNAME          LOC
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ---------- ---------- -------------- -------------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10                    10 ACCOUNTING     NEW YORK
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30                    10 ACCOUNTING     NEW YORK
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10                    10 ACCOUNTING     NEW YORK
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20                    10 ACCOUNTING     NEW YORK
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20                    10 ACCOUNTING     NEW YORK
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20                    10 ACCOUNTING     NEW YORK
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20                    10 ACCOUNTING     NEW YORK
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30                    10 ACCOUNTING     NEW YORK
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30                    10 ACCOUNTING     NEW YORK
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30                    10 ACCOUNTING     NEW YORK
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30                    10 ACCOUNTING     NEW YORK
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20                    10 ACCOUNTING     NEW YORK
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30                    10 ACCOUNTING     NEW YORK
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10                    10 ACCOUNTING     NEW YORK
      7839 KING       PRESIDENT            17-NOV-81       5000                    10                    20 RESEARCH       DALLAS
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30                    20 RESEARCH       DALLAS
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10                    20 RESEARCH       DALLAS
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20                    20 RESEARCH       DALLAS
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20                    20 RESEARCH       DALLAS
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20                    20 RESEARCH       DALLAS
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20                    20 RESEARCH       DALLAS
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30                    20 RESEARCH       DALLAS
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30                    20 RESEARCH       DALLAS
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30                    20 RESEARCH       DALLAS
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30                    20 RESEARCH       DALLAS
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20                    20 RESEARCH       DALLAS
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30                    20 RESEARCH       DALLAS
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10                    20 RESEARCH       DALLAS
      7839 KING       PRESIDENT            17-NOV-81       5000                    10                    30 SALES     CHICAGO
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30                    30 SALES     CHICAGO
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10                    30 SALES     CHICAGO
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20                    30 SALES     CHICAGO
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20                    30 SALES     CHICAGO
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20                    30 SALES     CHICAGO
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20                    30 SALES     CHICAGO
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30                    30 SALES     CHICAGO
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30                    30 SALES     CHICAGO
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30                    30 SALES     CHICAGO
```



```
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30                    30 SALES      CHICAGO
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30                    30 SALES      CHICAGO
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30                    30 SALES      CHICAGO
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30                    30 SALES      CHICAGO
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20                    30 SALES      CHICAGO
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30                    30 SALES      CHICAGO
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10                    30 SALES      CHICAGO
      7839 KING       PRESIDENT            17-NOV-81       5000                    10                    40 OPERATIONS        BOSTON
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30                    40 OPERATIONS        BOSTON
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10                    40 OPERATIONS        BOSTON
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20                    40 OPERATIONS        BOSTON
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20                    40 OPERATIONS        BOSTON
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20                    40 OPERATIONS        BOSTON
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20                    40 OPERATIONS        BOSTON
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30                    40 OPERATIONS        BOSTON
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30                    40 OPERATIONS        BOSTON
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30                    40 OPERATIONS        BOSTON
      7844 TUENER     SALESMAN        7698 08-SEP-81       1500          0         30                    40 OPERATIONS        BOSTON
      7876 ADAMS      CLERK           7788 23-MAY-81       1100                    20                    40 OPERATIONS        BOSTON
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30                    40 OPERATIONS        BOSTON
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10                    40 OPERATIONS        BOSTON

56 rows selected.
```

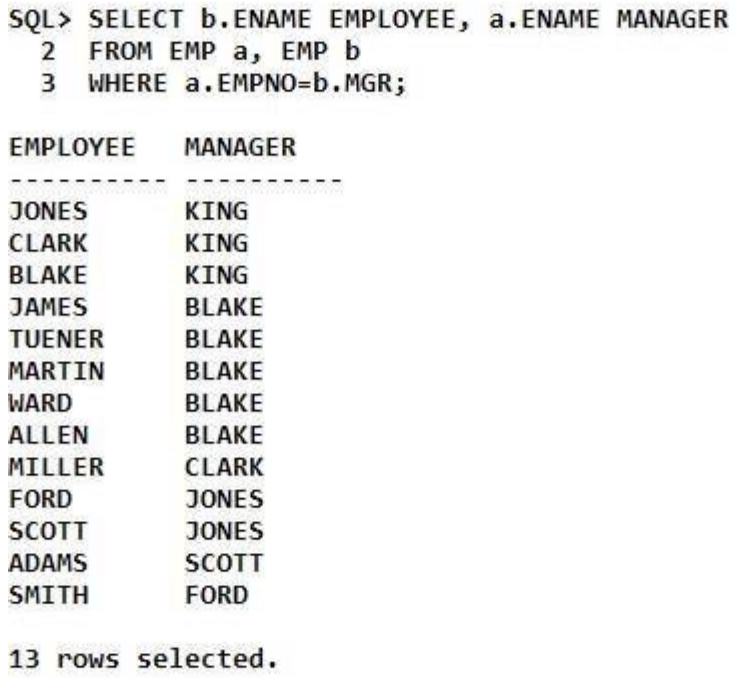
5. **SELF JOIN:** The SQL **SELF JOIN** is used to join a table to itself as if the table were two tables; temporarily renaming at least one table in the SQL statement.

SYNTAX:

```
SELECT a.column_name, b.column_name

FROM table1 a, table1 b

WHERE a.common_field = b.common_field;
```

EXAMPLE:

```
SELECT a.ENAME, b.ENAME

FROM EMP a, EMP BWHERE a.EMPNO = b.DEPTNO;
```



```
SQL> SELECT b.ENAME EMPLOYEE, a.ENAME MANAGER
  2  FROM EMP a, EMP b
  3  WHERE a.EMPNO=b.MGR;

EMPLOYEE   MANAGER
---------- ----------
JONES      KING
CLARK      KING
BLAKE      KING
JAMES      BLAKE
TUENER     BLAKE
MARTIN     BLAKE
WARD       BLAKE
ALLEN      BLAKE
MILLER     CLARK
FORD       JONES
SCOTT      JONES
ADAMS      SCOTT
SMITH      FORD

13 rows selected.
```

Aniket

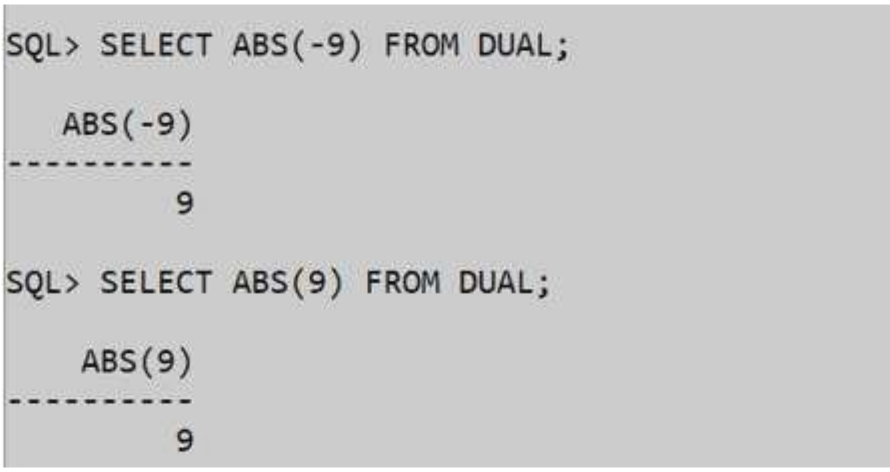
Prajapati

FYCS – 44

PRACTICAL NO. 6

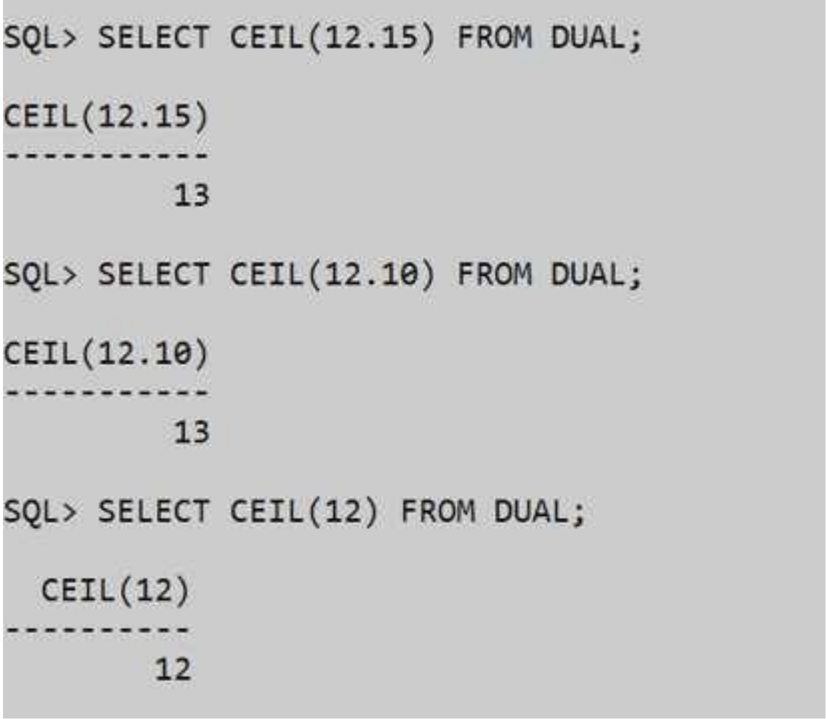
- NUMERICAL:

1. ABS



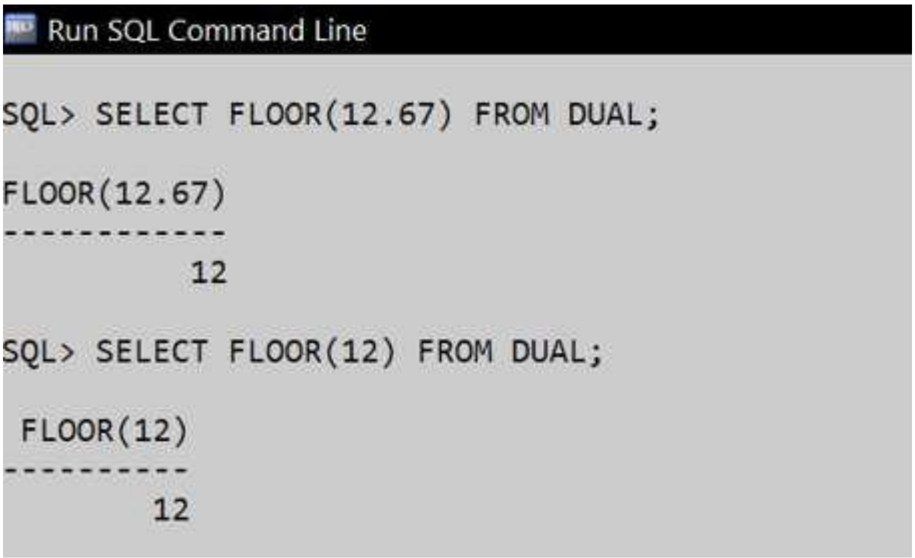
```
SQL> SELECT ABS(-9) FROM DUAL;

   ABS(-9)
----------
         9

SQL> SELECT ABS(9) FROM DUAL;

    ABS(9)
----------
         9
```

2. CEIL



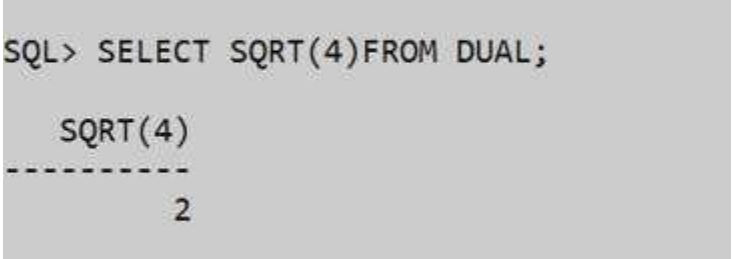
```
SQL> SELECT CEIL(12.15) FROM DUAL;

CEIL(12.15)
-----------
         13

SQL> SELECT CEIL(12.10) FROM DUAL;

CEIL(12.10)
-----------
         13

SQL> SELECT CEIL(12) FROM DUAL;

  CEIL(12)
----------
        12
```

3. FLOOR



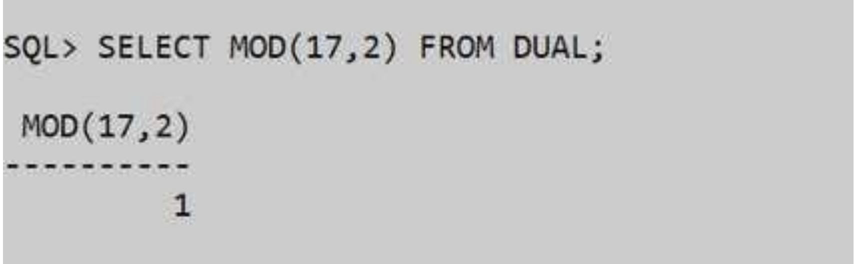
```
SQL> SELECT FLOOR(12.67) FROM DUAL;

FLOOR(12.67)
------------
          12

SQL> SELECT FLOOR(12) FROM DUAL;

 FLOOR(12)
----------
        12
```

4. SQRT



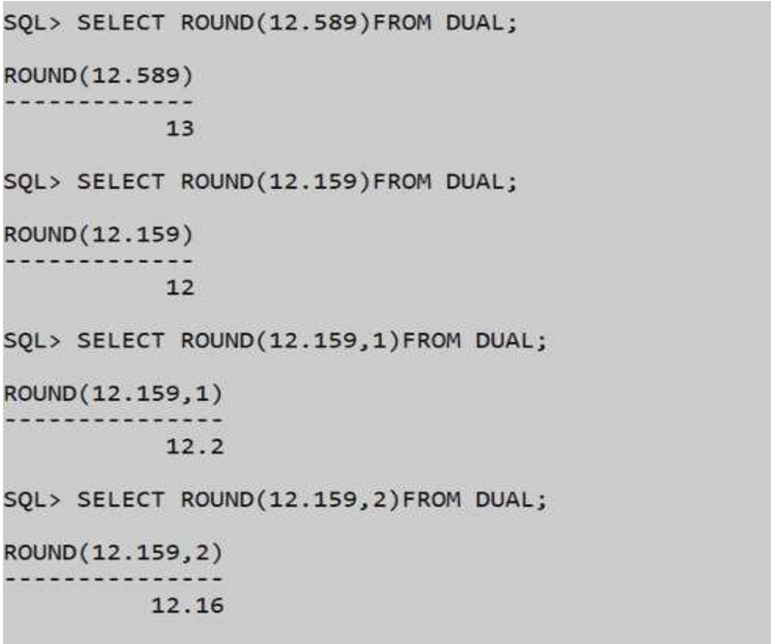
```
SQL> SELECT SQRT(4)FROM DUAL;

   SQRT(4)
----------
         2
```

5. MOD



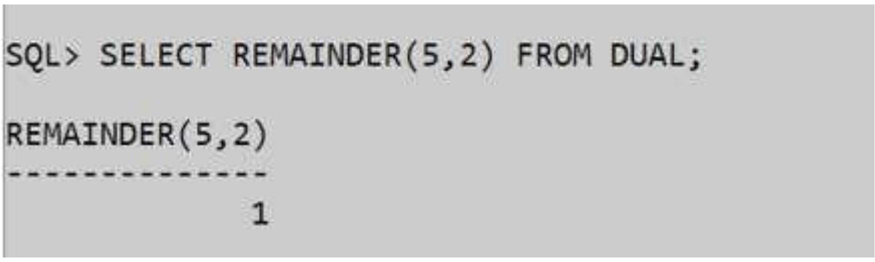
```
SQL> SELECT MOD(17,2) FROM DUAL;

 MOD(17,2)
----------
         1
```

6. ROUND



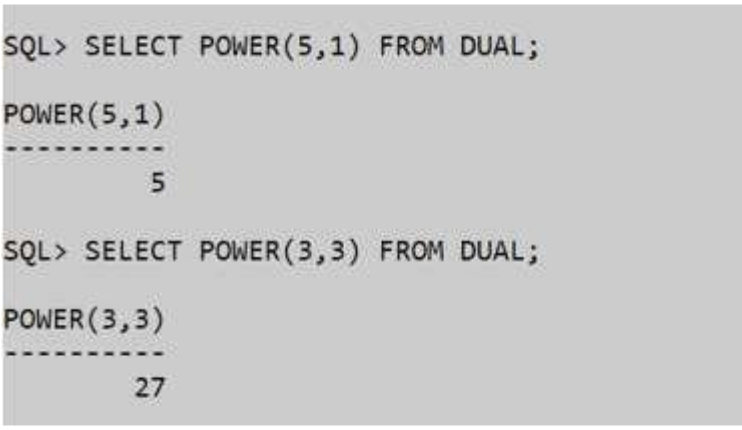
```
SQL> SELECT ROUND(12.589)FROM DUAL;

ROUND(12.589)
-------------
           13

SQL> SELECT ROUND(12.159)FROM DUAL;

ROUND(12.159)
-------------
           12

SQL> SELECT ROUND(12.159,1)FROM DUAL;

ROUND(12.159,1)
---------------
           12.2

SQL> SELECT ROUND(12.159,2)FROM DUAL;

ROUND(12.159,2)
---------------
          12.16
```

7. REMAIDER



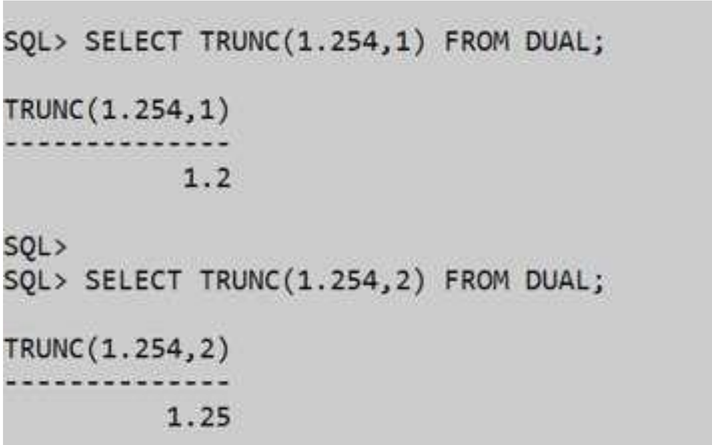
```
SQL> SELECT REMAINDER(5,2) FROM DUAL;

REMAINDER(5,2)
--------------
             1
```

8. POWER



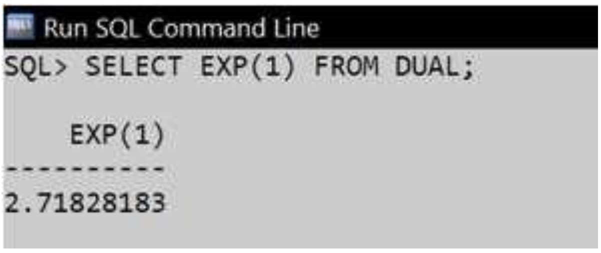
```
SQL> SELECT POWER(5,1) FROM DUAL;

POWER(5,1)
----------
         5

SQL> SELECT POWER(3,3) FROM DUAL;

POWER(3,3)
----------
        27
```

9. TRUNC



```
SQL> SELECT TRUNC(1.254,1) FROM DUAL;

TRUNC(1.254,1)
--------------
           1.2

SQL>
SQL> SELECT TRUNC(1.254,2) FROM DUAL;

TRUNC(1.254,2)
--------------
          1.25
```

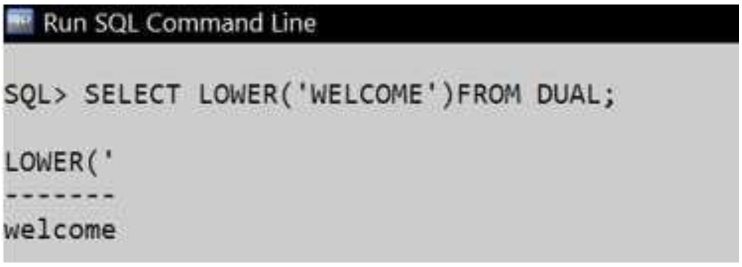
10. EXP



```
SQL> SELECT EXP(1) FROM DUAL;

    EXP(1)
----------
2.71828183
```

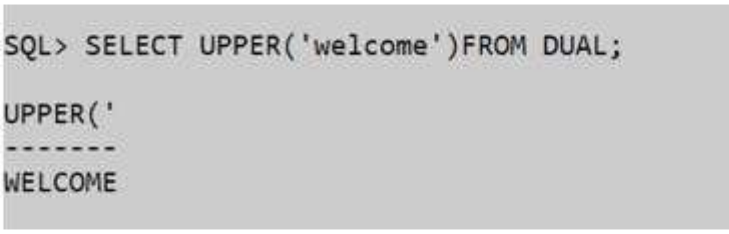
CHARACTER:

1. LOWER



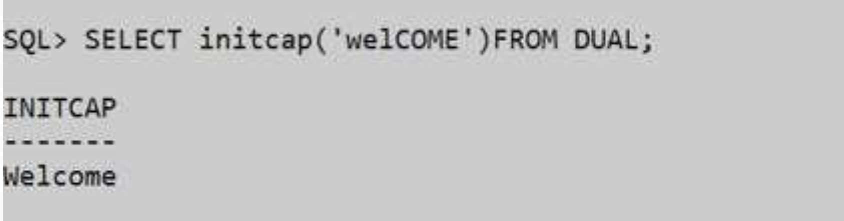
```
SQL> SELECT LOWER('WELCOME')FROM DUAL;

LOWER('
-------
welcome
```

2. UPPER



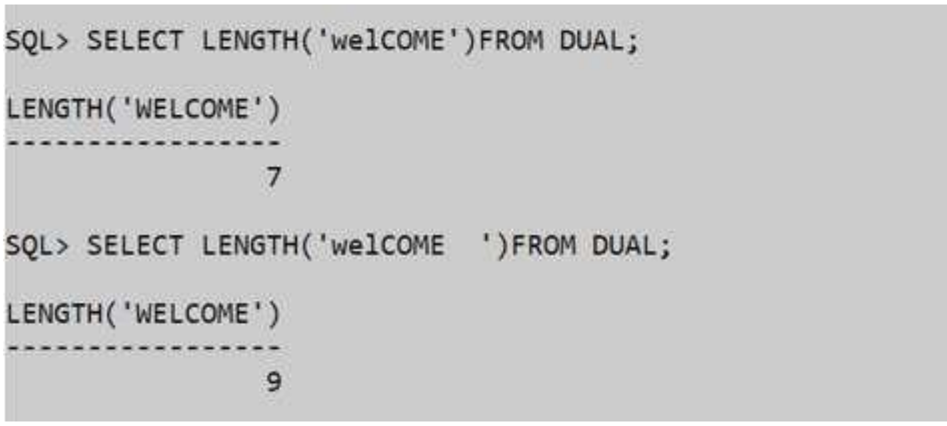
```
SQL> SELECT UPPER('welcome')FROM DUAL;

UPPER('
-------
WELCOME
```

3. INITCAP



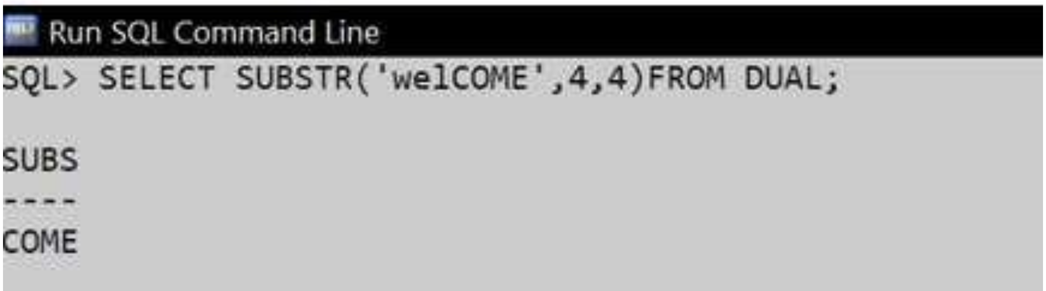
```
SQL> SELECT initcap('welCOME')FROM DUAL;

INITCAP
-------
Welcome
```

4. LENGTH



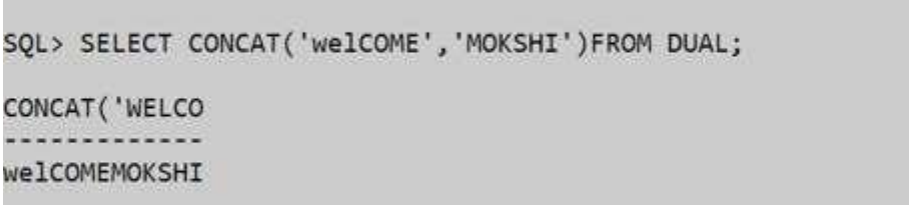
```
SQL> SELECT LENGTH('welCOME')FROM DUAL;

LENGTH('WELCOME')
-----------------
                7

SQL> SELECT LENGTH('welCOME  ')FROM DUAL;

LENGTH('WELCOME')
-----------------
                9
```

5. SUBSTR



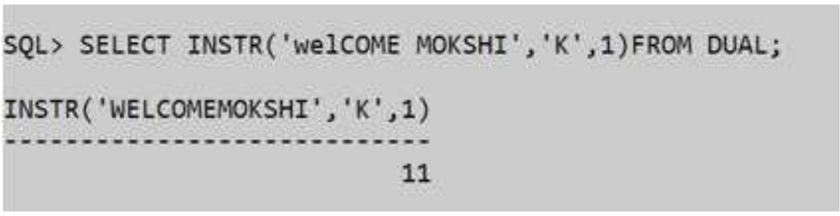
```
SQL> SELECT SUBSTR('welCOME',4,4)FROM DUAL;

SUBS
----
COME
```

6. CONCAT



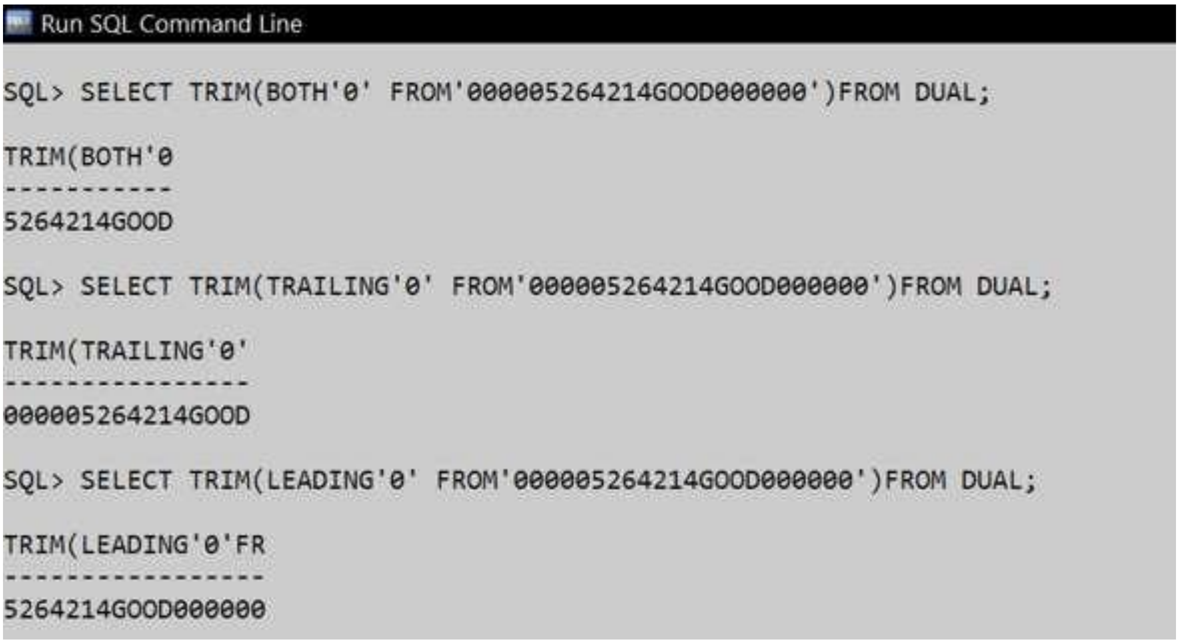
```
SQL> SELECT CONCAT('welCOME','MOKSHI')FROM DUAL;

CONCAT('WELCO
-------------
welCOMEMOKSHI
```

7. INSTR



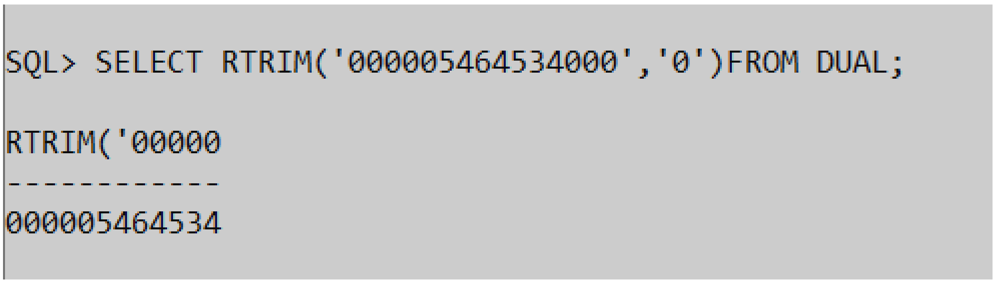
```
SQL> SELECT INSTR('welCOME MOKSHI','K',1)FROM DUAL;

INSTR('WELCOMEMOKSHI','K',1)
----------------------------
                          11
```

8. TRIM



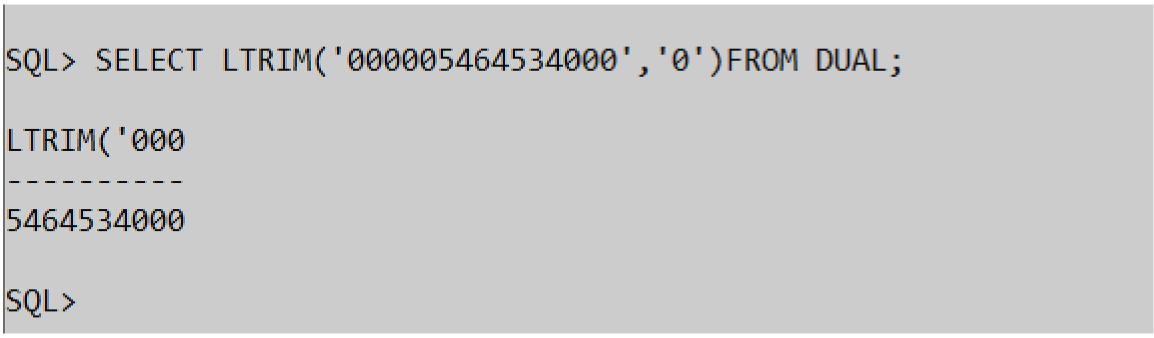
```
SQL> SELECT TRIM(BOTH'0' FROM'000005264214GOOD000000')FROM DUAL;

TRIM(BOTH'0
-----------
5264214GOOD

SQL> SELECT TRIM(TRAILING'0' FROM'000005264214GOOD000000')FROM DUAL;

TRIM(TRAILING'0'
----------------
000005264214GOOD

SQL> SELECT TRIM(LEADING'0' FROM'000005264214GOOD000000')FROM DUAL;

TRIM(LEADING'0'FR
-----------------
5264214GOOD000000
```

9. RTRIM



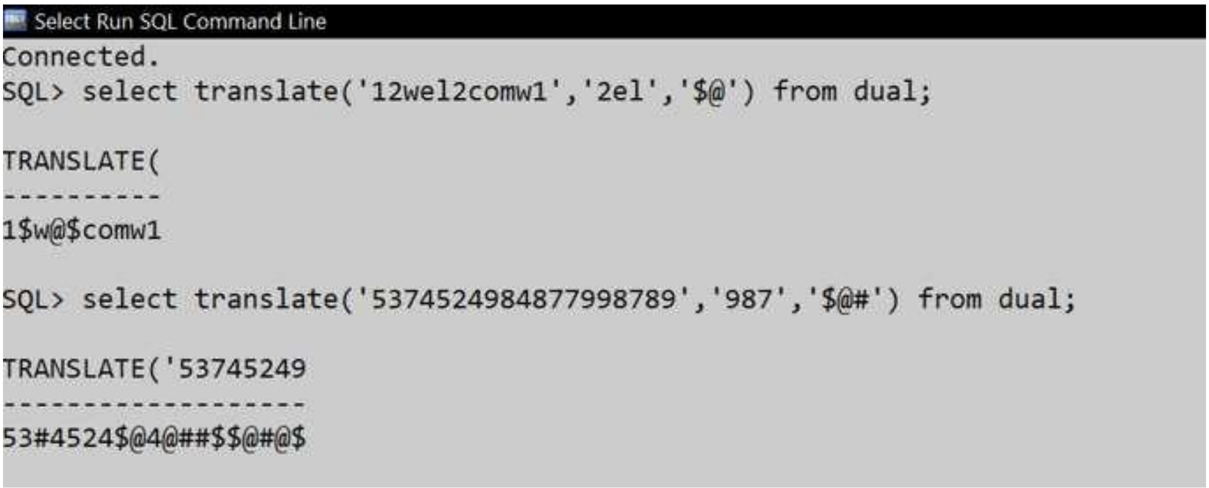
```
SQL> SELECT RTRIM('000005464534000','0')FROM DUAL;

RTRIM('00000
------------
000005464534
```

10. LTRIM



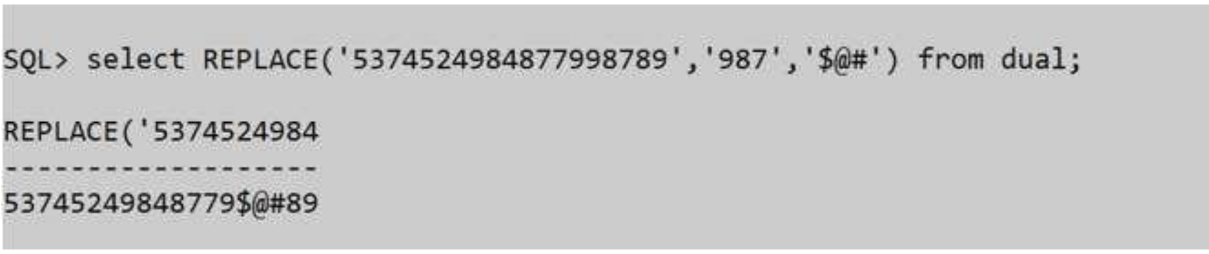
```
SQL> SELECT LTRIM('000005464534000','0')FROM DUAL;

LTRIM('000
----------
5464534000

SQL>
```

11. TRANSLATE



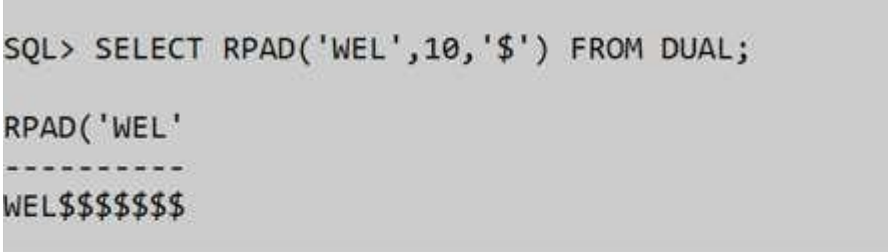
```
Connected.
SQL> select translate('12wel2comw1','2el','$@') from dual;

TRANSLATE(
----------
1$w@$comw1

SQL> select translate('5374524984877998789','987','$@#') from dual;

TRANSLATE('53745249
-------------------
53#4524$@4@##$$@#@$
```

12. REPLACE



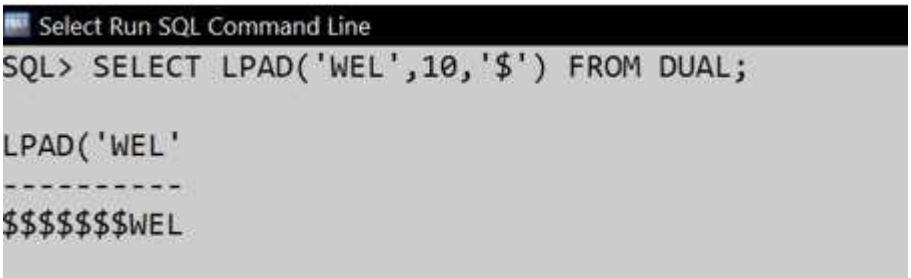
```
SQL> select REPLACE('5374524984877998789','987','$@#') from dual;

REPLACE('5374524984
-------------------
53745249848779$@#89
```

13. RPAD



```
SQL> SELECT RPAD('WEL',10,'$') FROM DUAL;

RPAD('WEL'
----------
WEL$$$$$$$
```

14. LPAD

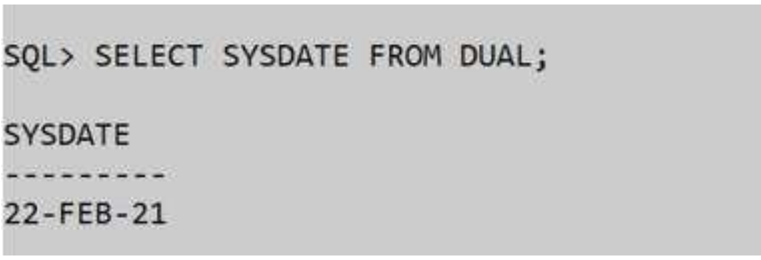


Select Run SQL Command Line

```
SQL> SELECT LPAD('WEL',10,'$') FROM DUAL;

LPAD('WEL'
----------
$$$$$$$WEL
```

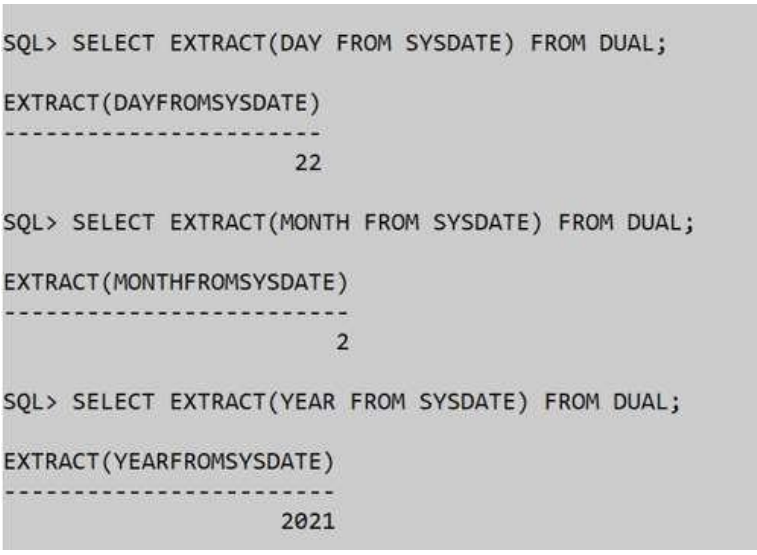
DATE FUNCTION:

1. SYSDATE



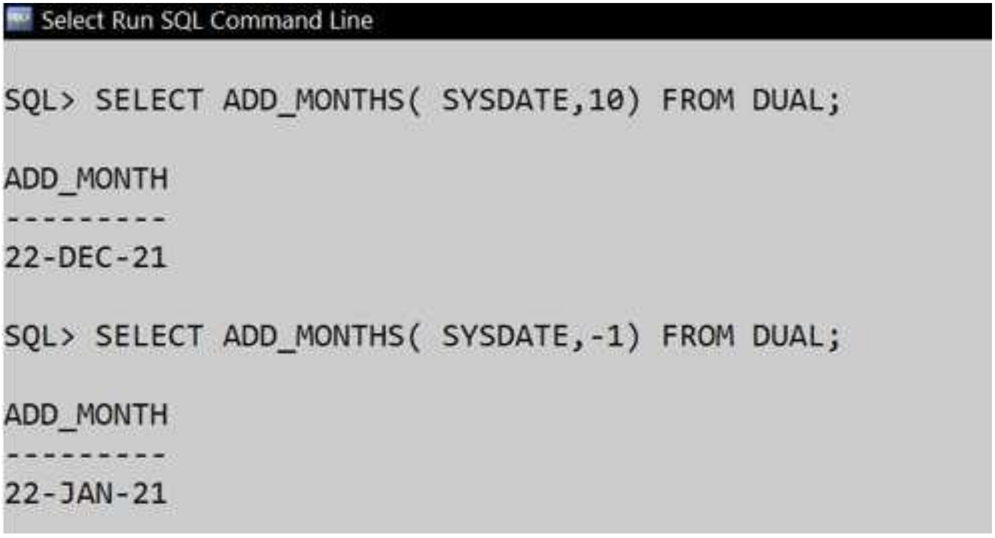
```
SQL> SELECT SYSDATE FROM DUAL;

SYSDATE
---------
22-FEB-21
```

2. ECTRACT



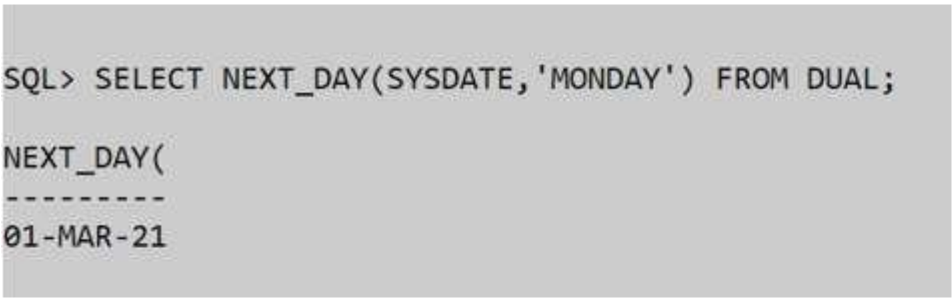
```
SQL> SELECT EXTRACT(DAY FROM SYSDATE) FROM DUAL;

EXTRACT(DAYFROMSYSDATE)
-----------------------
                     22

SQL> SELECT EXTRACT(MONTH FROM SYSDATE) FROM DUAL;

EXTRACT(MONTHFROMSYSDATE)
-------------------------
                        2

SQL> SELECT EXTRACT(YEAR FROM SYSDATE) FROM DUAL;

EXTRACT(YEARFROMSYSDATE)
------------------------
                    2021
```

3. ADD_MONTHS



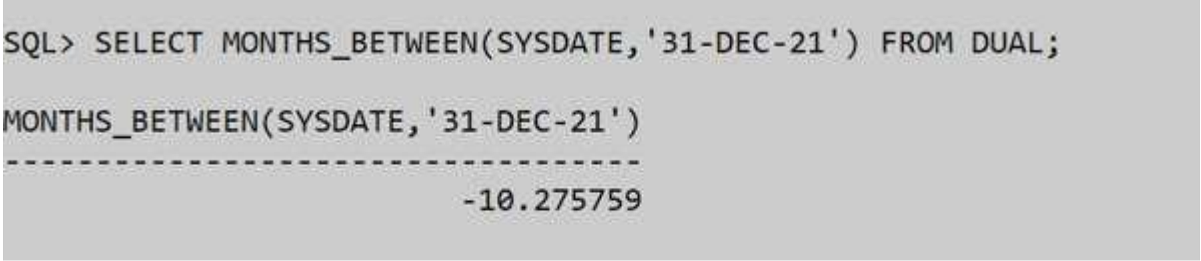
```
SQL> SELECT ADD_MONTHS( SYSDATE,10) FROM DUAL;

ADD_MONTH
---------
22-DEC-21

SQL> SELECT ADD_MONTHS( SYSDATE,-1) FROM DUAL;

ADD_MONTH
---------
22-JAN-21
```

4. NEXT_DAY



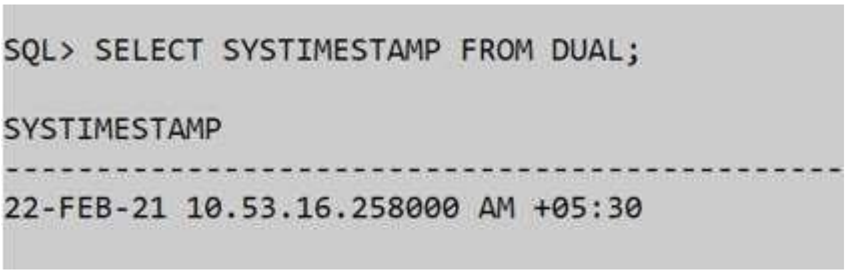
```
SQL> SELECT NEXT_DAY(SYSDATE,'MONDAY') FROM DUAL;

NEXT_DAY(
---------
01-MAR-21
```

5. MONTHS_BETWEEN



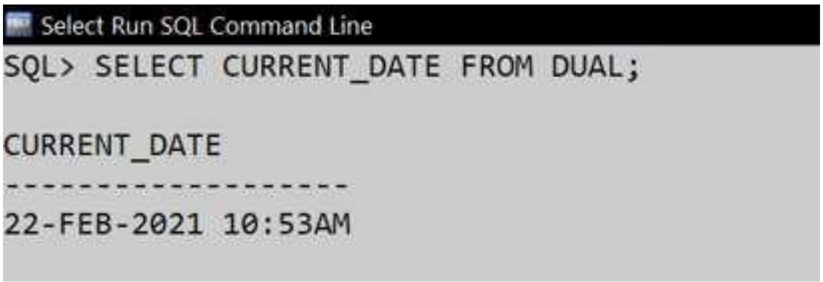
```
SQL> SELECT MONTHS_BETWEEN(SYSDATE,'31-DEC-21') FROM DUAL;

MONTHS_BETWEEN(SYSDATE,'31-DEC-21')
-----------------------------------
                         -10.275759
```

6. SYSTIMESTAMP



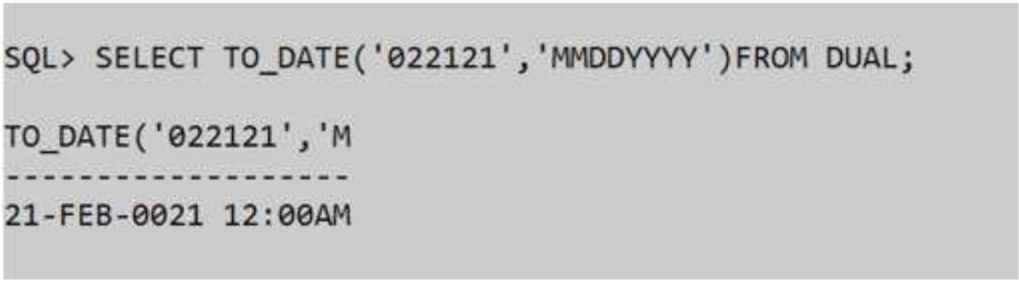
```
SQL> SELECT SYSTIMESTAMP FROM DUAL;

SYSTIMESTAMP
---------------------------------------------------------------------------
22-FEB-21 10.53.16.258000 AM +05:30
```

7. CURRENT_DATE



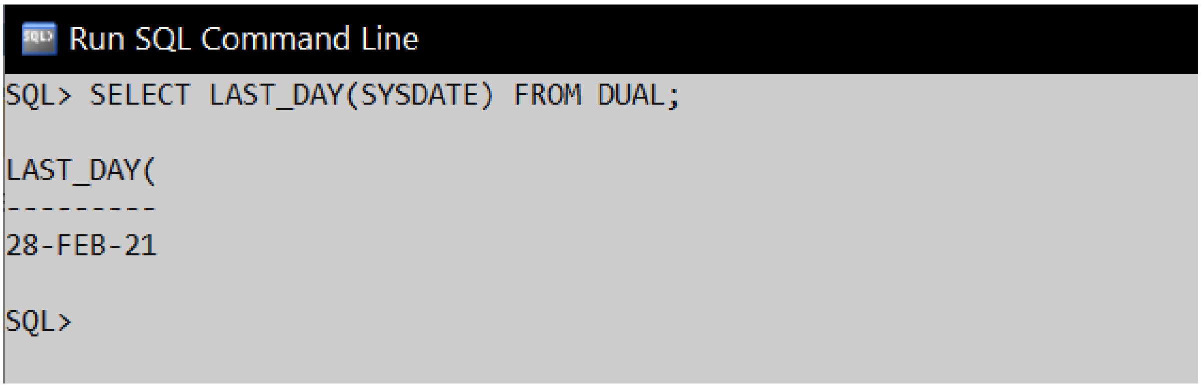
```
SQL> SELECT CURRENT_DATE FROM DUAL;

CURRENT_DATE
--------------------
22-FEB-2021 10:53AM
```

8. TO_DATE



```
SQL> SELECT TO_DATE('022121','MMDDYYYY')FROM DUAL;

TO_DATE('022121','M
-------------------
21-FEB-0021 12:00AM
```

9. LAST_DATE



```
SQL> SELECT LAST_DAY(SYSDATE) FROM DUAL;

LAST_DAY(
---------
28-FEB-21

SQL>
```

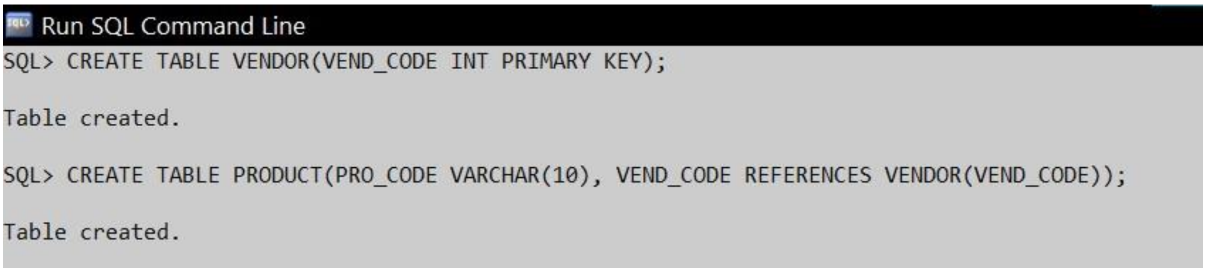
Aniket

Prajapati

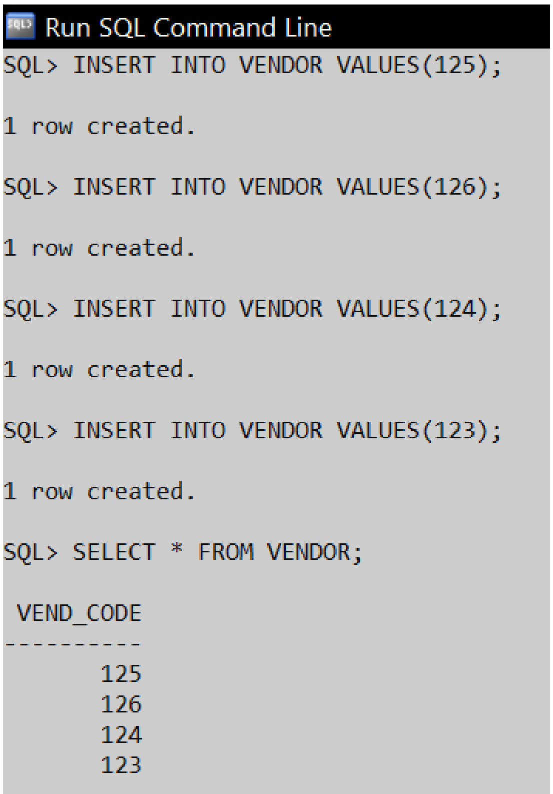
FYCS – 44

PRATICAL NO. 7

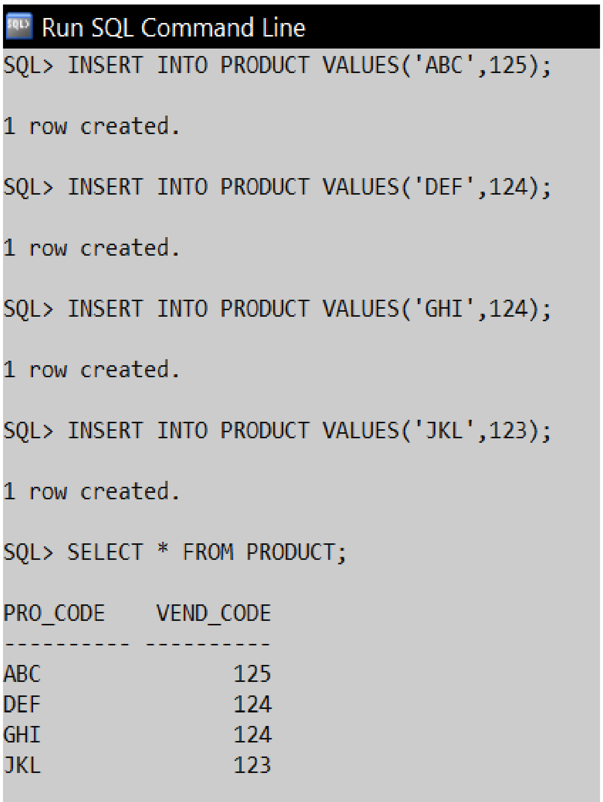
Suppose that a Product table contains two attributes, PROD_CODE and VEND_CODE. The values for the PROD_CODE are: ABC, DEF, GHI and JKL. These are matched by the following values for the VEND_CODE: 125, 124, 124 and 123, respectively (e.g., PROD_CODE value ABC corresponds to VEND_CODE value 125). The Vendor table contains a single attribute, VEND_CODE, with values 123, 124, 125 and 126. (The VEND_CODE attribute in the Product table is a foreign key to the VEND_CODE in the Vendor table.)



```
SQL> CREATE TABLE VENDOR(VEND_CODE INT PRIMARY KEY);

Table created.

SQL> CREATE TABLE PRODUCT(PRO_CODE VARCHAR(10), VEND_CODE REFERENCES VENDOR(VEND_CODE));

Table created.
```



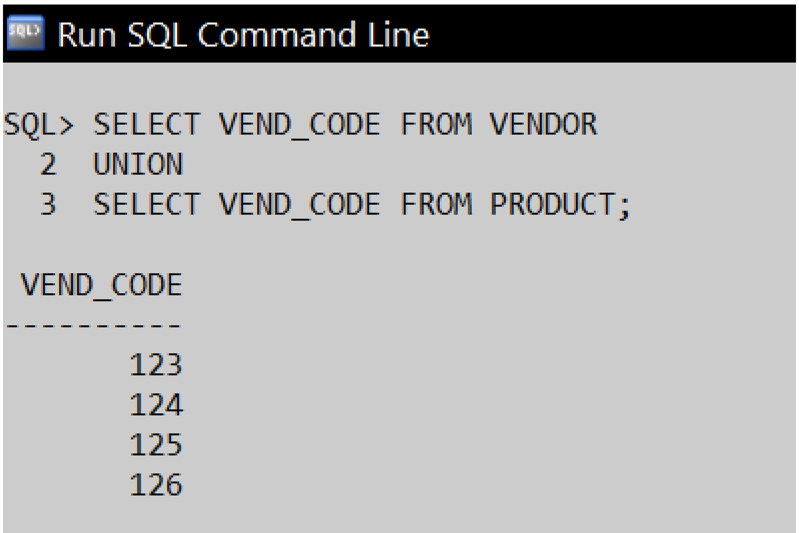
```
SQL> INSERT INTO VENDOR VALUES(125);

1 row created.

SQL> INSERT INTO VENDOR VALUES(126);

1 row created.

SQL> INSERT INTO VENDOR VALUES(124);

1 row created.

SQL> INSERT INTO VENDOR VALUES(123);

1 row created.

SQL> SELECT * FROM VENDOR;

 VEND_CODE
----------
       125
       126
       124
       123
```



```
SQL> INSERT INTO PRODUCT VALUES('ABC',125);

1 row created.

SQL> INSERT INTO PRODUCT VALUES('DEF',124);

1 row created.

SQL> INSERT INTO PRODUCT VALUES('GHI',124);

1 row created.

SQL> INSERT INTO PRODUCT VALUES('JKL',123);

1 row created.

SQL> SELECT * FROM PRODUCT;

PRO_CODE    VEND_CODE
---------- ----------
ABC               125
DEF               124
GHI               124
JKL               123
```

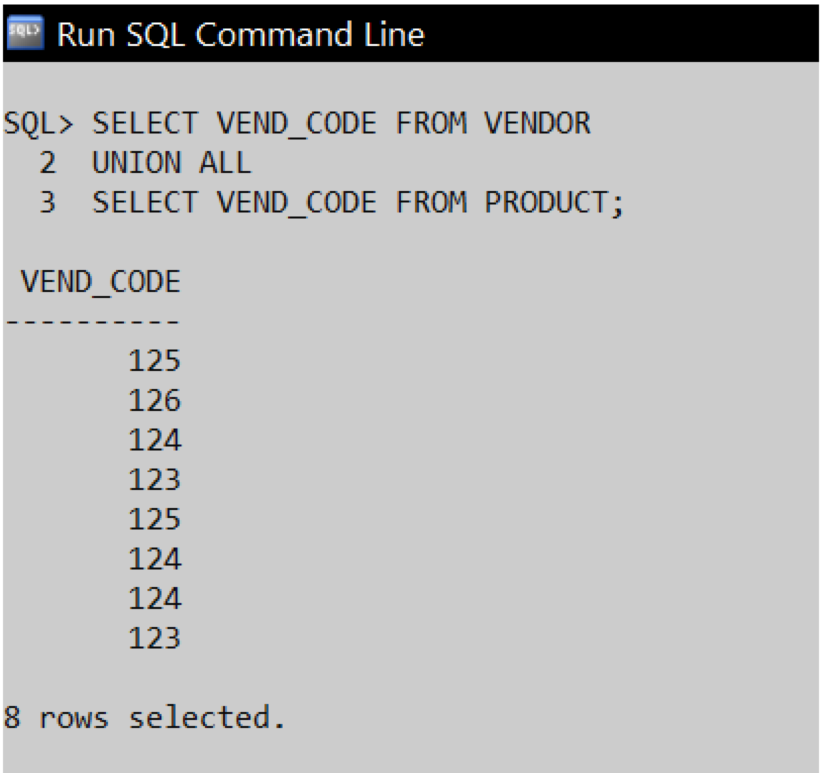
Given the information, what would be the query output for the following? Show values.

a) A UNION query based on these two tables.



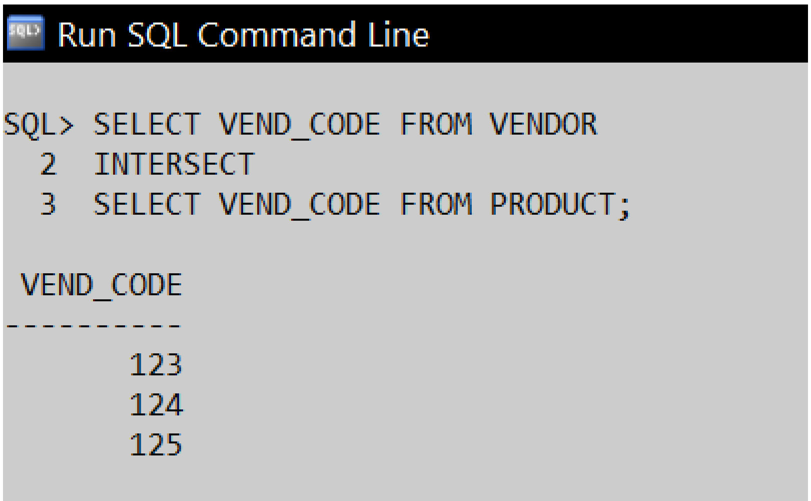
```
SQL> SELECT VEND_CODE FROM VENDOR
  2  UNION
  3  SELECT VEND_CODE FROM PRODUCT;

 VEND_CODE
----------
       123
       124
       125
       126
```

b) A UNION ALL query based on these two tables



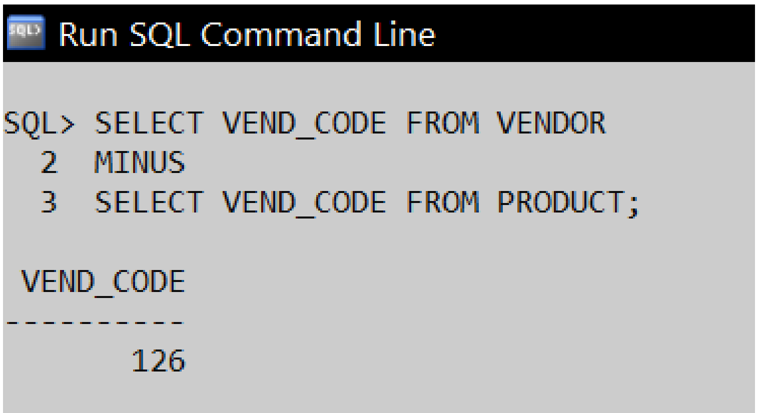
```
SQL> SELECT VEND_CODE FROM VENDOR
  2  UNION ALL
  3  SELECT VEND_CODE FROM PRODUCT;

 VEND_CODE
----------
       125
       126
       124
       123
       125
       124
       124
       123

8 rows selected.
```

c) An INTERSECT query based on these two tables



```
SQL> SELECT VEND_CODE FROM VENDOR
  2  INTERSECT
  3  SELECT VEND_CODE FROM PRODUCT;

 VEND_CODE
----------
       123
       124
       125
```

d) A MINUS query based on these two tables



```
SQL> SELECT VEND_CODE FROM VENDOR
  2  MINUS
  3  SELECT VEND_CODE FROM PRODUCT;

 VEND_CODE
----------
       126
```

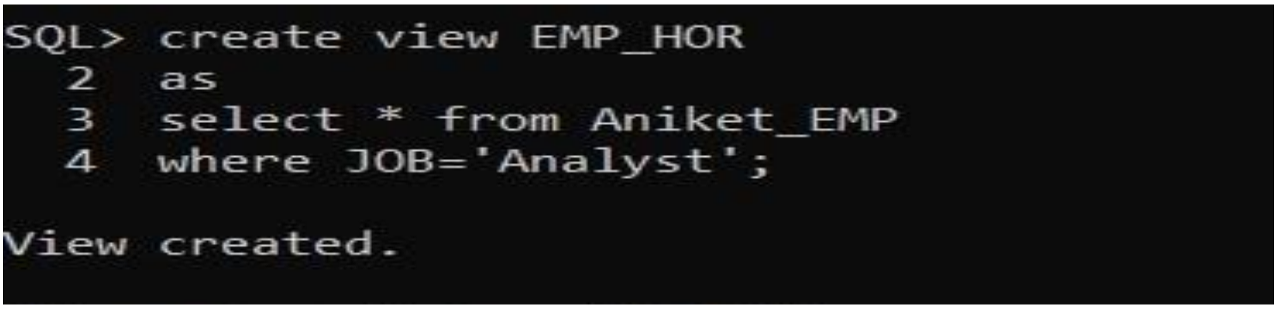
Aniket Prajapati
FYCS-44

Practical No. 8

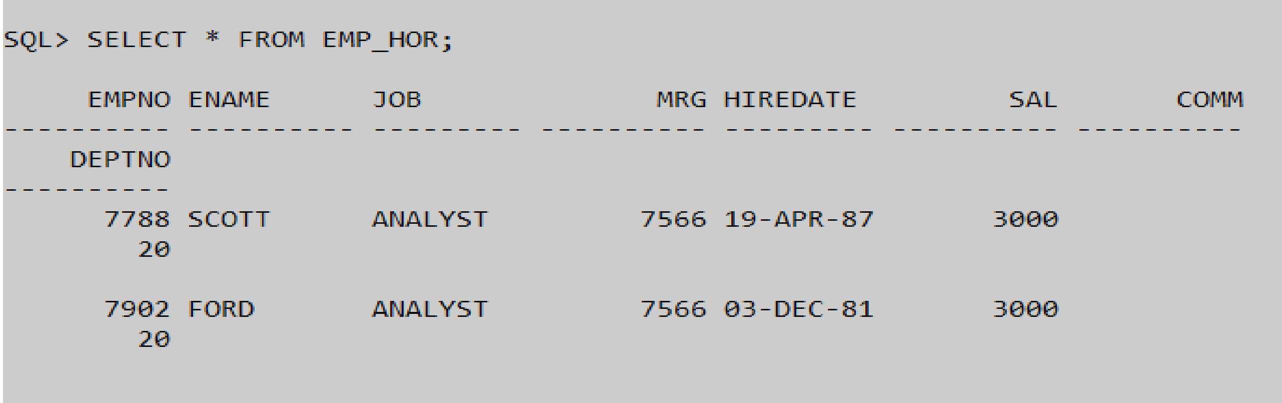
Study of various types of views

Considering Emp and Dept table, perform the following:

1. Create a view named emp_hor with the job titled as ‘ANALYST’.

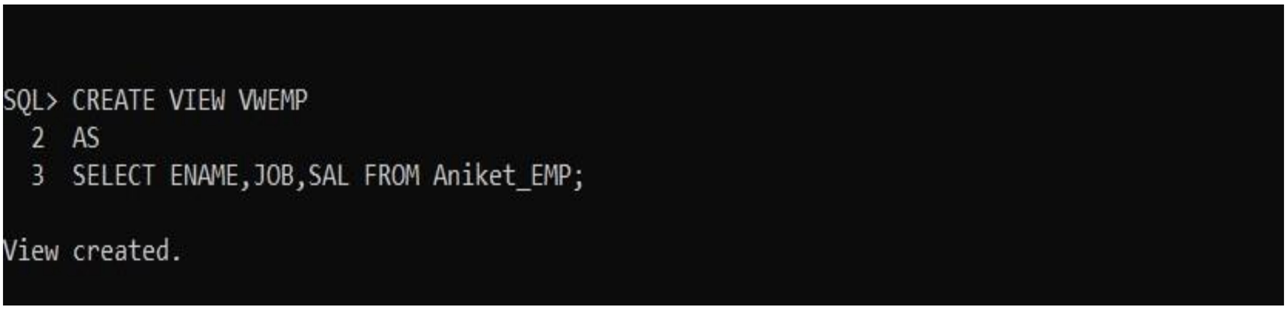


```
SQL> create view EMP_HOR
  2  as
  3  select * from Aniket_EMP
  4  where JOB='Analyst';

View created.
```

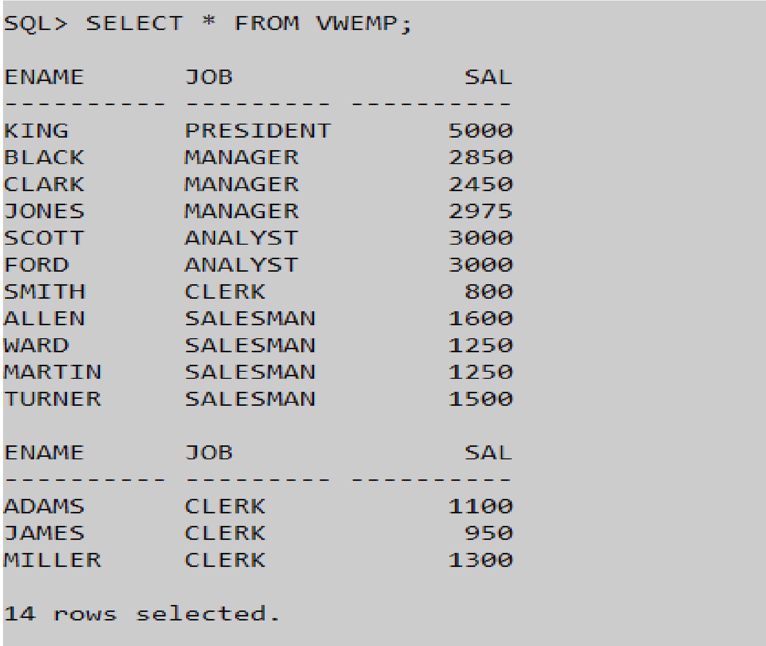


```
SQL> SELECT * FROM EMP_HOR;

     EMPNO ENAME      JOB              MRG HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000
        20

      7902 FORD       ANALYST         7566 03-DEC-81       3000
        20
```

2)Create a view named vwemp specifying name of employees, job and their salary.

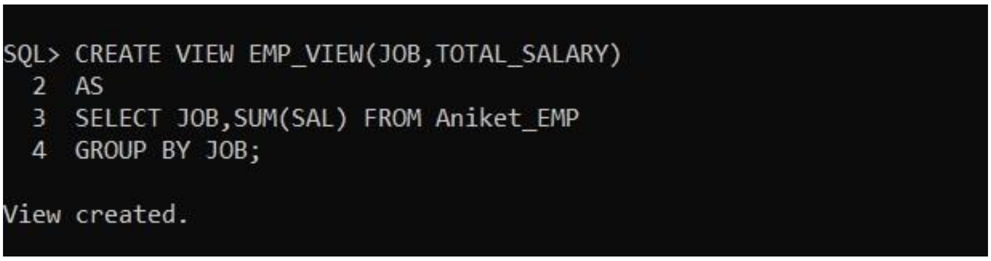


```
SQL> CREATE VIEW VWEMP
  2  AS
  3  SELECT ENAME,JOB,SAL FROM Aniket_EMP;

View created.
```

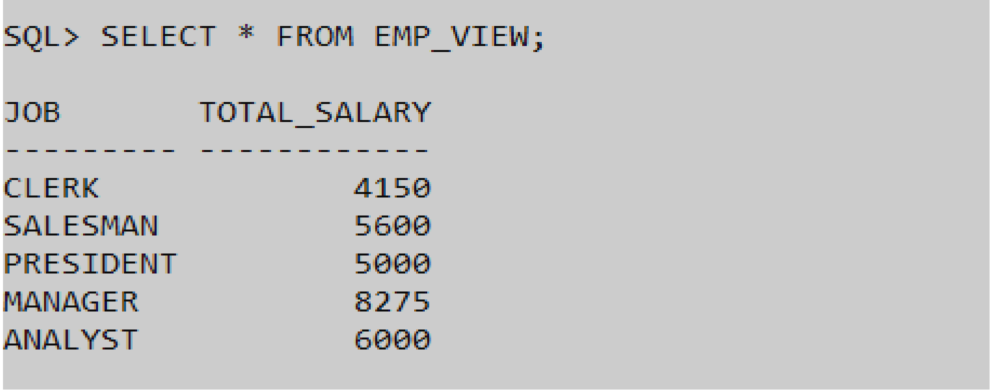


```
SQL> SELECT * FROM VWEMP;

ENAME      JOB              SAL
---------- --------- ----------
KING       PRESIDENT       5000
BLACK      MANAGER         2850
CLARK      MANAGER         2450
JONES      MANAGER         2975
SCOTT      ANALYST         3000
FORD       ANALYST         3000
SMITH      CLERK            800
ALLEN      SALESMAN        1600
WARD       SALESMAN        1250
MARTIN     SALESMAN        1250
TURNER     SALESMAN        1500

ENAME      JOB              SAL
---------- --------- ----------
ADAMS      CLERK           1100
JAMES      CLERK            950
MILLER     CLERK           1300

14 rows selected.
```

3) Create a view displaying total salary on the basis of the jobs.

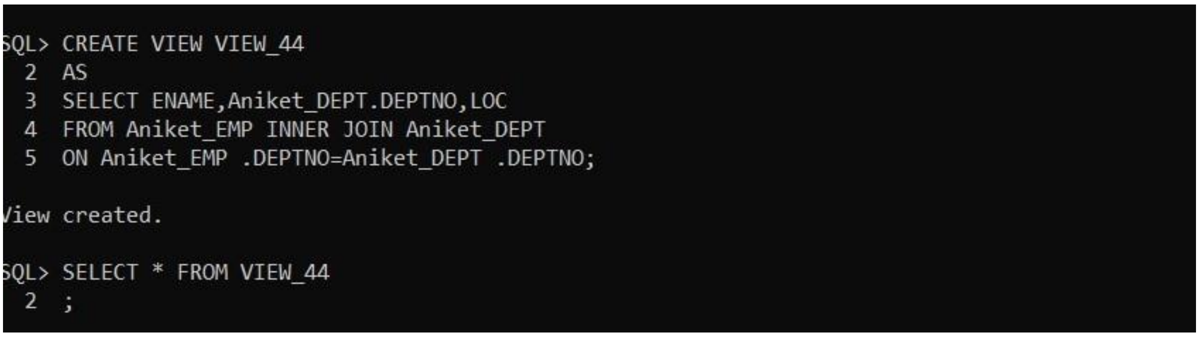


```
SQL> CREATE VIEW EMP_VIEW(JOB,TOTAL_SALARY)
  2  AS
  3  SELECT JOB,SUM(SAL) FROM Aniket_EMP
  4  GROUP BY JOB;

View created.
```

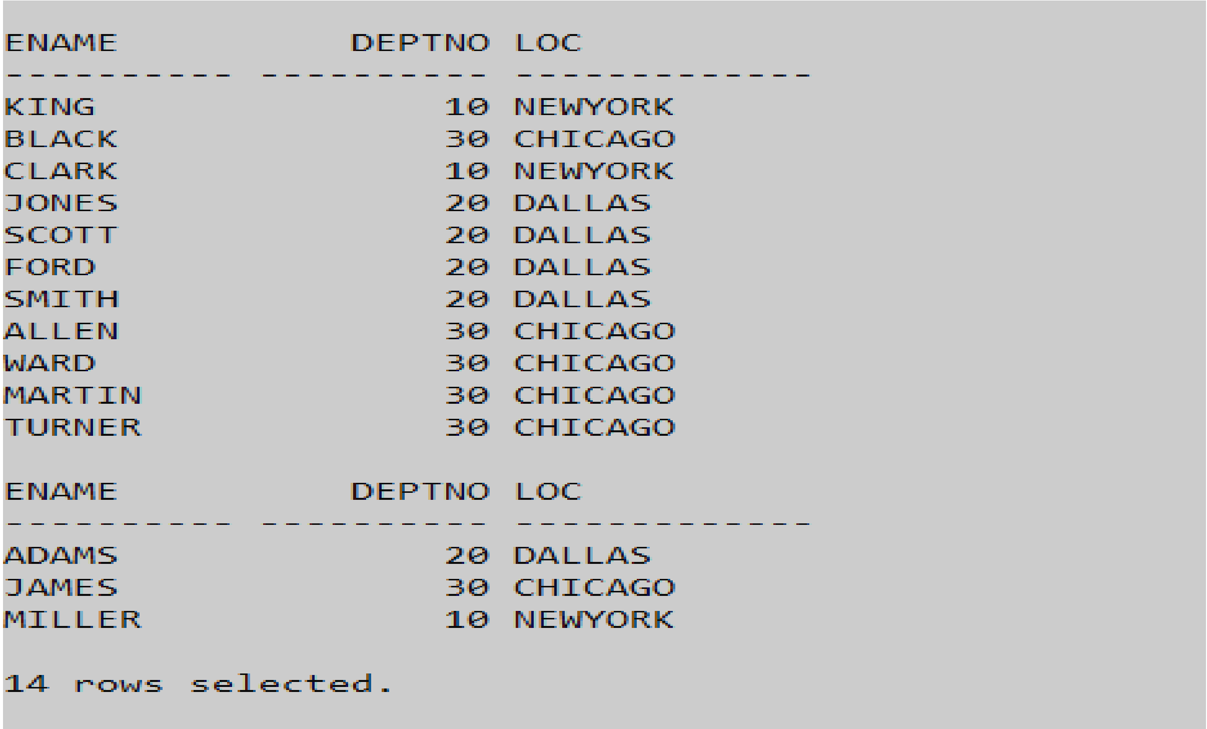


```
SQL> SELECT * FROM EMP_VIEW;

JOB       TOTAL_SALARY
--------- ------------
CLERK             4150
SALESMAN          5600
PRESIDENT         5000
MANAGER           8275
ANALYST           6000
```

4) Create a view with contains name of employee, dept and the location of the employees.

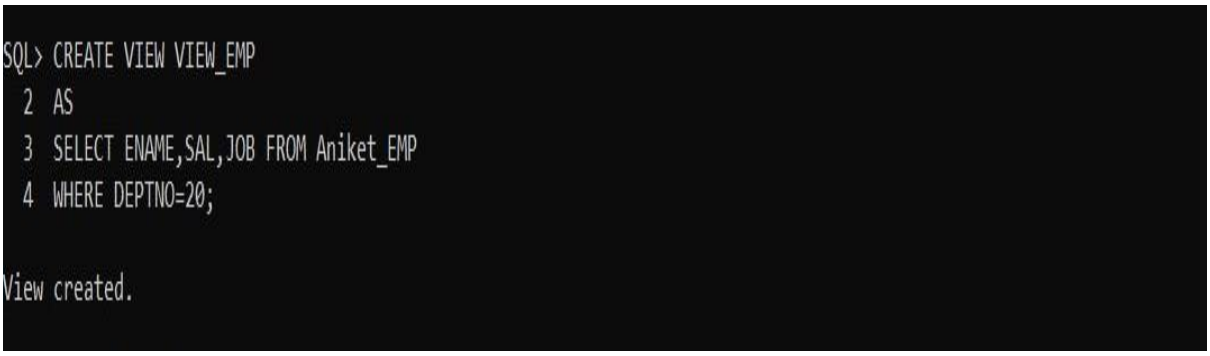


```
SQL> CREATE VIEW VIEW_44
  2  AS
  3  SELECT ENAME,Aniket_DEPT.DEPTNO,LOC
  4  FROM Aniket_EMP INNER JOIN Aniket_DEPT
  5  ON Aniket_EMP .DEPTNO=Aniket_DEPT .DEPTNO;

View created.

SQL> SELECT * FROM VIEW_44
  2  ;
```

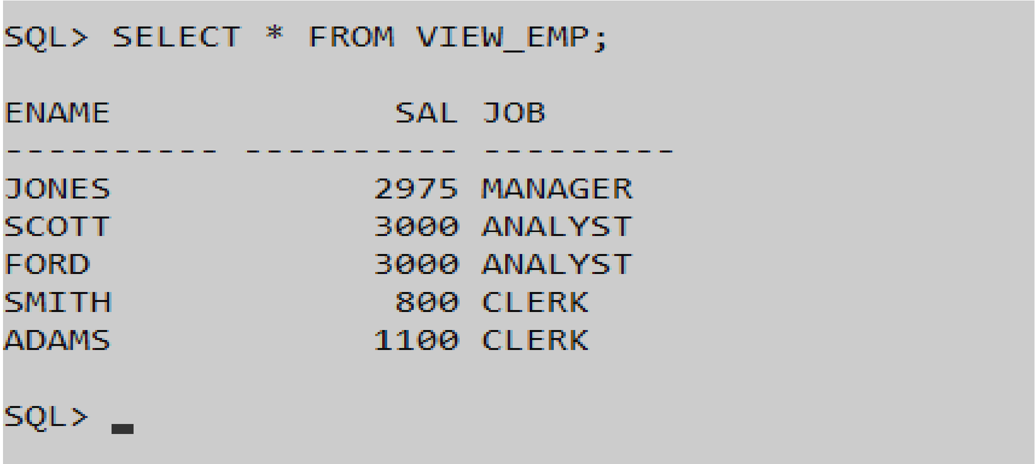


```
ENAME          DEPTNO LOC
---------- ---------- -------------
KING               10 NEWYORK
BLACK              30 CHICAGO
CLARK              10 NEWYORK
JONES              20 DALLAS
SCOTT              20 DALLAS
FORD               20 DALLAS
SMITH              20 DALLAS
ALLEN              30 CHICAGO
WARD               30 CHICAGO
MARTIN             30 CHICAGO
TURNER             30 CHICAGO

ENAME          DEPTNO LOC
---------- ---------- -------------
ADAMS              20 DALLAS
JAMES              30 CHICAGO
MILLER             10 NEWYORK

14 rows selected.
```

5) Create a view to display the name of the employees with their salary and job who belongs to department 20.

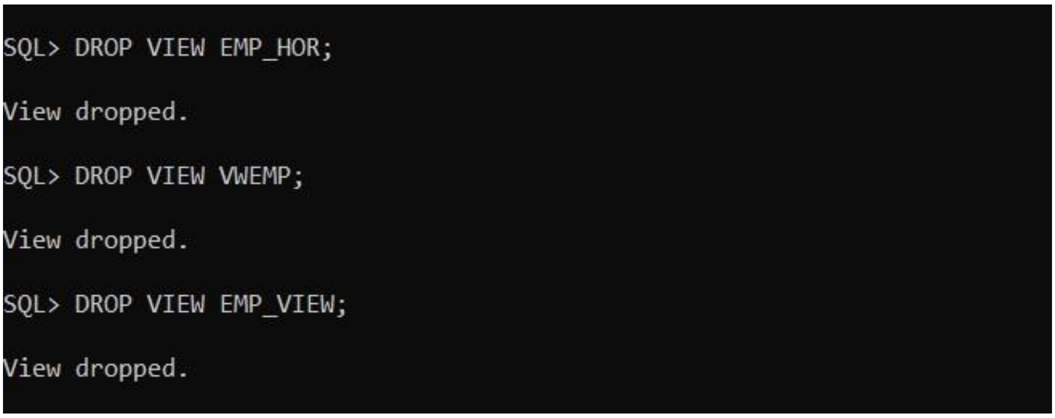


```
SQL> CREATE VIEW VIEW_EMP
  2  AS
  3  SELECT ENAME,SAL,JOB FROM Aniket_EMP
  4  WHERE DEPTNO=20;

View created.
```

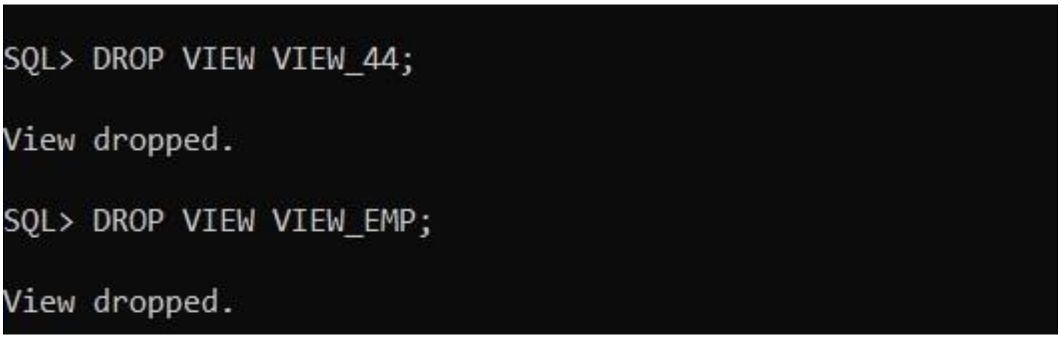


```
SQL> SELECT * FROM VIEW_EMP;

ENAME             SAL JOB
---------- ---------- ---------
JONES            2975 MANAGER
SCOTT            3000 ANALYST
FORD             3000 ANALYST
SMITH             800 CLERK
ADAMS            1100 CLERK

SQL> _
```

6) Delete all the views created above.



```
SQL> DROP VIEW EMP_HOR;

View dropped.

SQL> DROP VIEW VWEMP;

View dropped.

SQL> DROP VIEW EMP_VIEW;

View dropped.
```



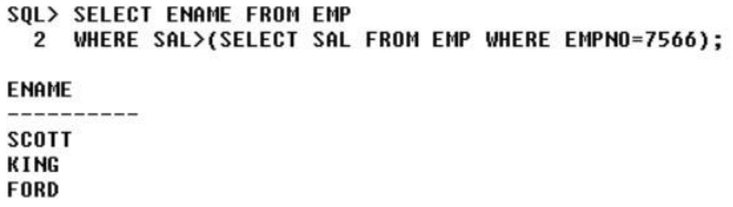
```
SQL> DROP VIEW VIEW_44;

View dropped.

SQL> DROP VIEW VIEW_EMP;

View dropped.
```

Aniket Prajapati
FYCS-44

Practical No 9

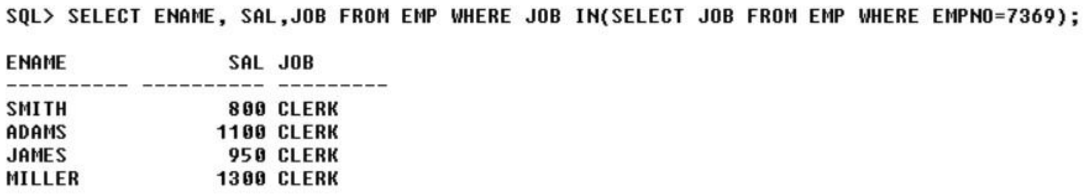
Study of subqueries with all its clauses

1. Display the employee name whose salary is greater than the salary of employee 7566.



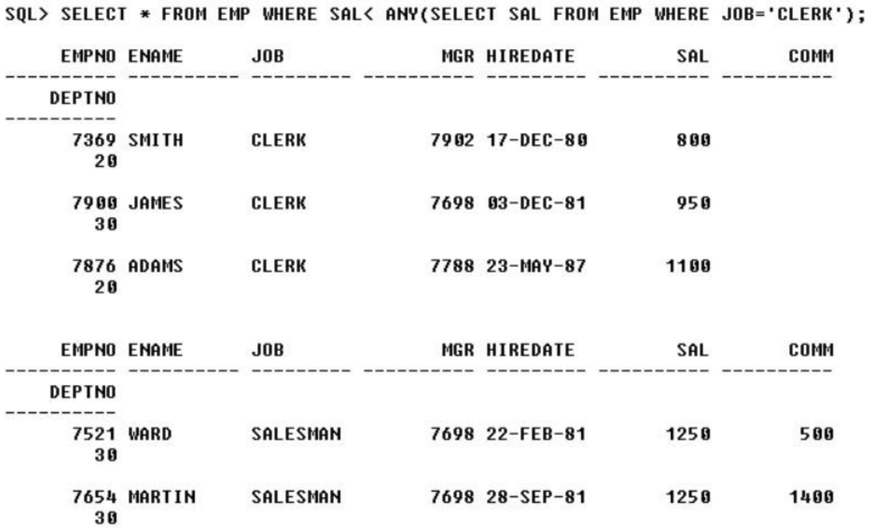
```
SQL> SELECT ENAME FROM EMP
  2  WHERE SAL>(SELECT SAL FROM EMP WHERE EMPNO=7566);

ENAME
----------
SCOTT
KING
FORD
```

2.Display the employee name, sal, job of the employee whose job is similar to the employee 7369.



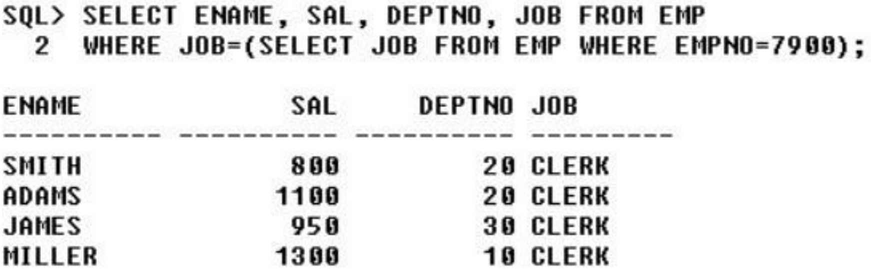
```
SQL> SELECT ENAME, SAL,JOB FROM EMP WHERE JOB IN(SELECT JOB FROM EMP WHERE EMPNO=7369);

ENAME             SAL JOB
---------- ---------- ---------
SMITH             800 CLERK
ADAMS            1100 CLERK
JAMES             950 CLERK
MILLER           1300 CLERK
```

3.Display the employee name with the salary less than any salary of job type CLERK.



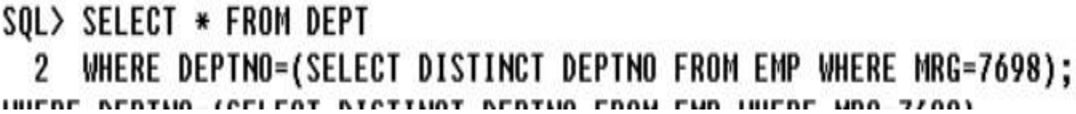
```
SQL> SELECT * FROM EMP WHERE SAL< ANY(SELECT SAL FROM EMP WHERE JOB='CLERK');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7369 SMITH      CLERK           7902 17-DEC-80        800
        20

      7900 JAMES      CLERK           7698 03-DEC-81        950
        30

      7876 ADAMS      CLERK           7788 23-MAY-87       1100
        20


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500
        30

      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400
        30
```

4 Display the employee name, salary, department id, job id for those employees who works in the same designation as the employee works whose id is 7900.

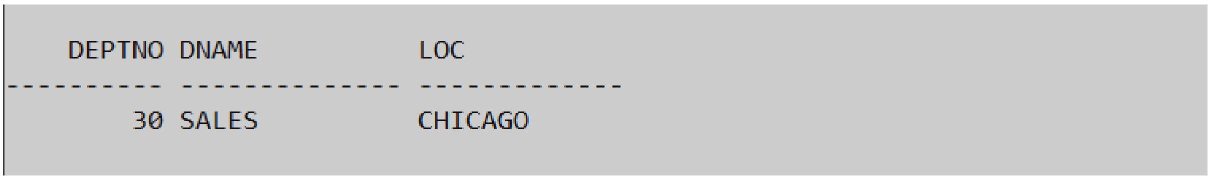


```
SQL> SELECT ENAME, SAL, DEPTNO, JOB FROM EMP
  2  WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNO=7900);

ENAME             SAL     DEPTNO JOB
---------- ---------- ---------- ---------
SMITH             800         20 CLERK
ADAMS            1100         20 CLERK
JAMES             950         30 CLERK
MILLER           1300         10 CLERK
```

5.Display the detail of department whose manager Ecode='7698'.



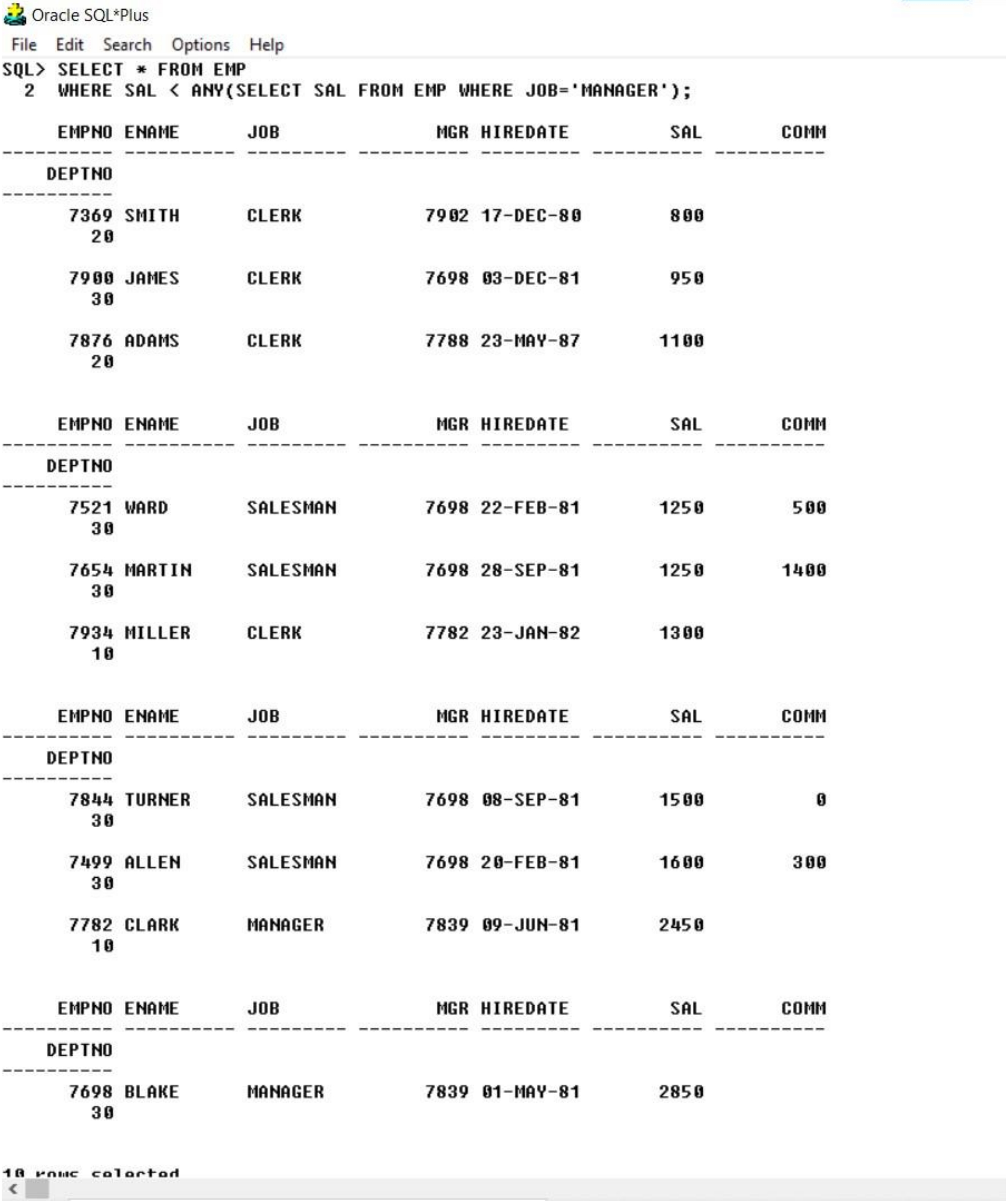
```
SQL> SELECT * FROM DEPT
  2  WHERE DEPTNO=(SELECT DISTINCT DEPTNO FROM EMP WHERE MRG=7698);
WHERE DEPTNO=(SELECT DISTINCT DEPTNO FROM EMP WHERE MRG=7698)
```



```
    DEPTNO DNAME          LOC
---------- -------------- -------------
        30 SALES          CHICAGO
```

6. Display the employees whose salary is greater than any MANAGER.

Aniket Prajapati
FYCS-44



```
SQL> SELECT * FROM EMP
  2  WHERE SAL < ANY(SELECT SAL FROM EMP WHERE JOB='MANAGER');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7369 SMITH      CLERK           7902 17-DEC-80        800
        20

      7900 JAMES      CLERK           7698 03-DEC-81        950
        30

      7876 ADAMS      CLERK           7788 23-MAY-87       1100
        20


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500
        30

      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400
        30

      7934 MILLER     CLERK           7782 23-JAN-82       1300
        10


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0
        30

      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300
        30

      7782 CLARK      MANAGER         7839 09-JUN-81       2450
        10


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850
        30


10 rows selected
```

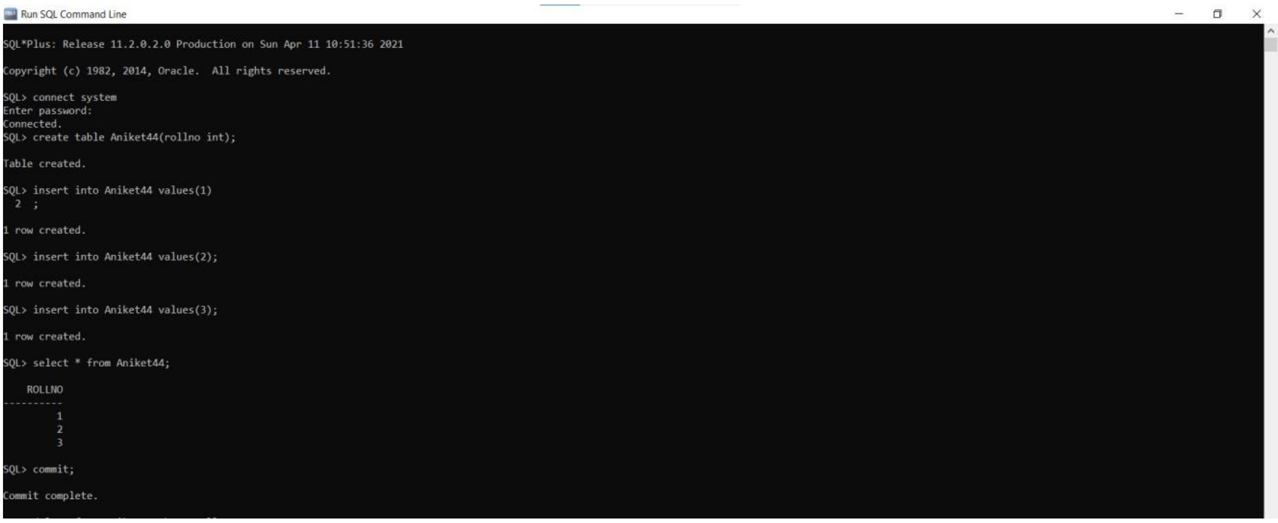
Aniket Prajapati

FYCS-44

Practical No 10

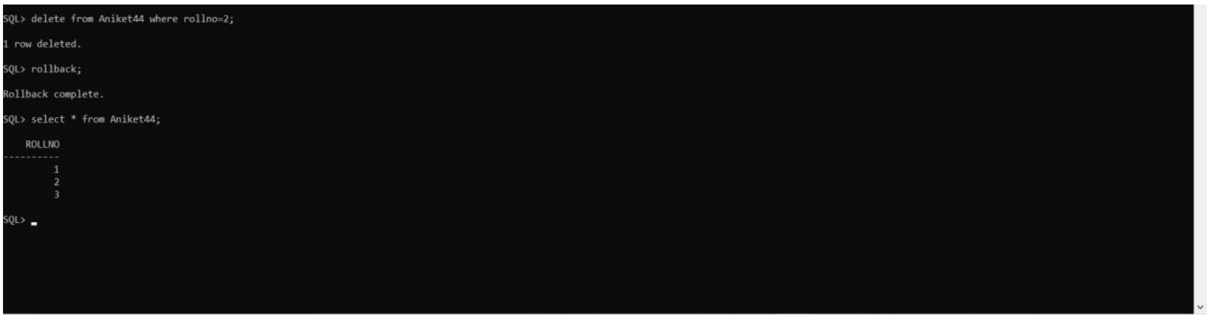
Study of Transaction (Commit/ Rollback), Locks

1. Perform Commit and Rollback on a table.
   Commit:



```
SQL*Plus: Release 11.2.0.2.0 Production on Sun Apr 11 10:51:36 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect system
Enter password:
Connected.
SQL> create table Aniket44(rollno int);

Table created.

SQL> insert into Aniket44 values(1)
  2  ;

1 row created.

SQL> insert into Aniket44 values(2);

1 row created.

SQL> insert into Aniket44 values(3);

1 row created.

SQL> select * from Aniket44;

    ROLLNO
----------
         1
         2
         3

SQL> commit;

Commit complete.
```

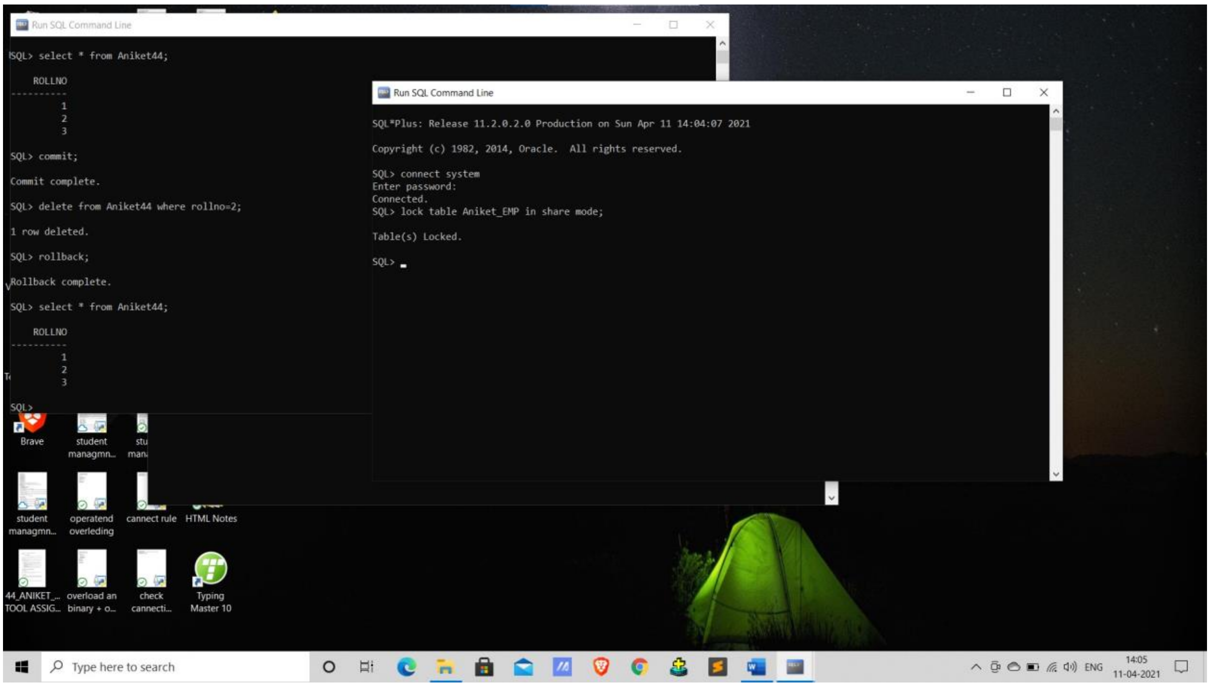
Rollback:



```
SQL> delete from Aniket44 where rollno=2;

1 row deleted.

SQL> rollback;

Rollback complete.

SQL> select * from Aniket44;

    ROLLNO
----------
         1
         2
         3

SQL>
```

2. Implementation of Share and Exclusive Lock Mode in employee table

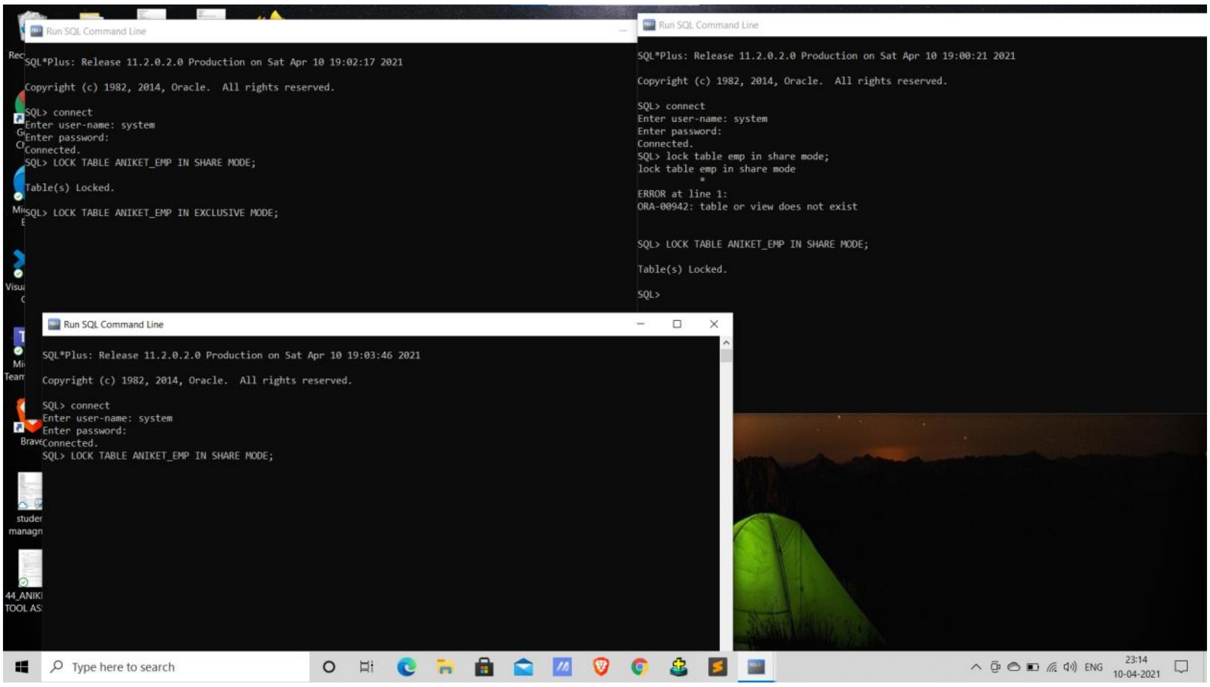
Shared Mode:



```
SQL> select * from Aniket44;

   ROLLNO
---------
        1
        2
        3

SQL> commit;

Commit complete.

SQL> delete from Aniket44 where rollno=2;

1 row deleted.

SQL> rollback;

Rollback complete.

SQL> select * from Aniket44;

   ROLLNO
---------
        1
        2
        3

SQL>
```

```
SQL*Plus: Release 11.2.0.2.0 Production on Sun Apr 11 14:04:07 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect system
Enter password:
Connected.
SQL> lock table Aniket_EMP in share mode;

Table(s) Locked.

SQL> _
```

Exclusive Lock:



```
SQL*Plus: Release 11.2.0.2.0 Production on Sat Apr 10 19:02:17 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> LOCK TABLE ANIKET_EMP IN SHARE MODE;

Table(s) Locked.

SQL> LOCK TABLE ANIKET_EMP IN EXCLUSIVE MODE;
```

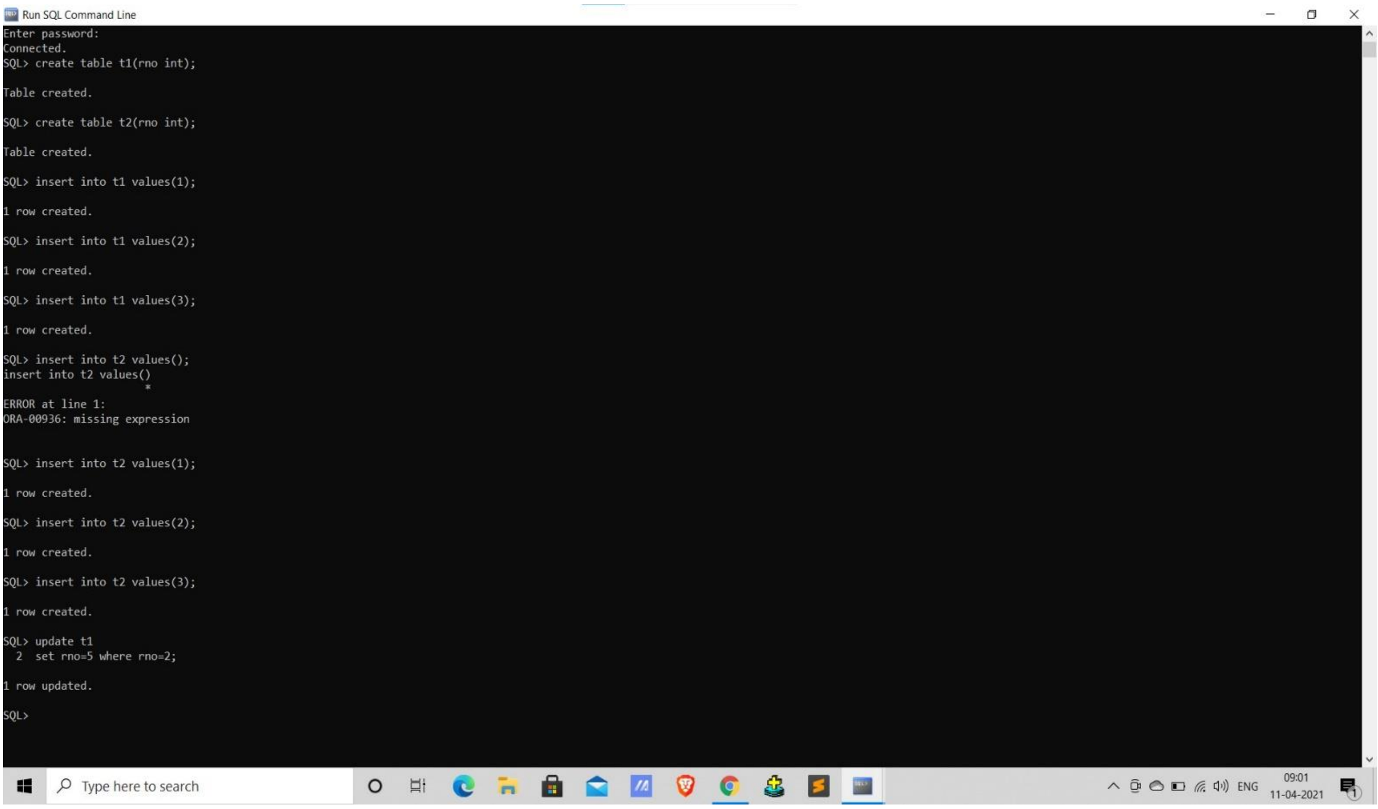
```
SQL*Plus: Release 11.2.0.2.0 Production on Sat Apr 10 19:00:21 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> lock table emp in share mode;
lock table emp in share mode
           *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> LOCK TABLE ANIKET_EMP IN SHARE MODE;

Table(s) Locked.

SQL>
```

```
SQL*Plus: Release 11.2.0.2.0 Production on Sat Apr 10 19:03:46 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> LOCK TABLE ANIKET_EMP IN SHARE MODE;
```

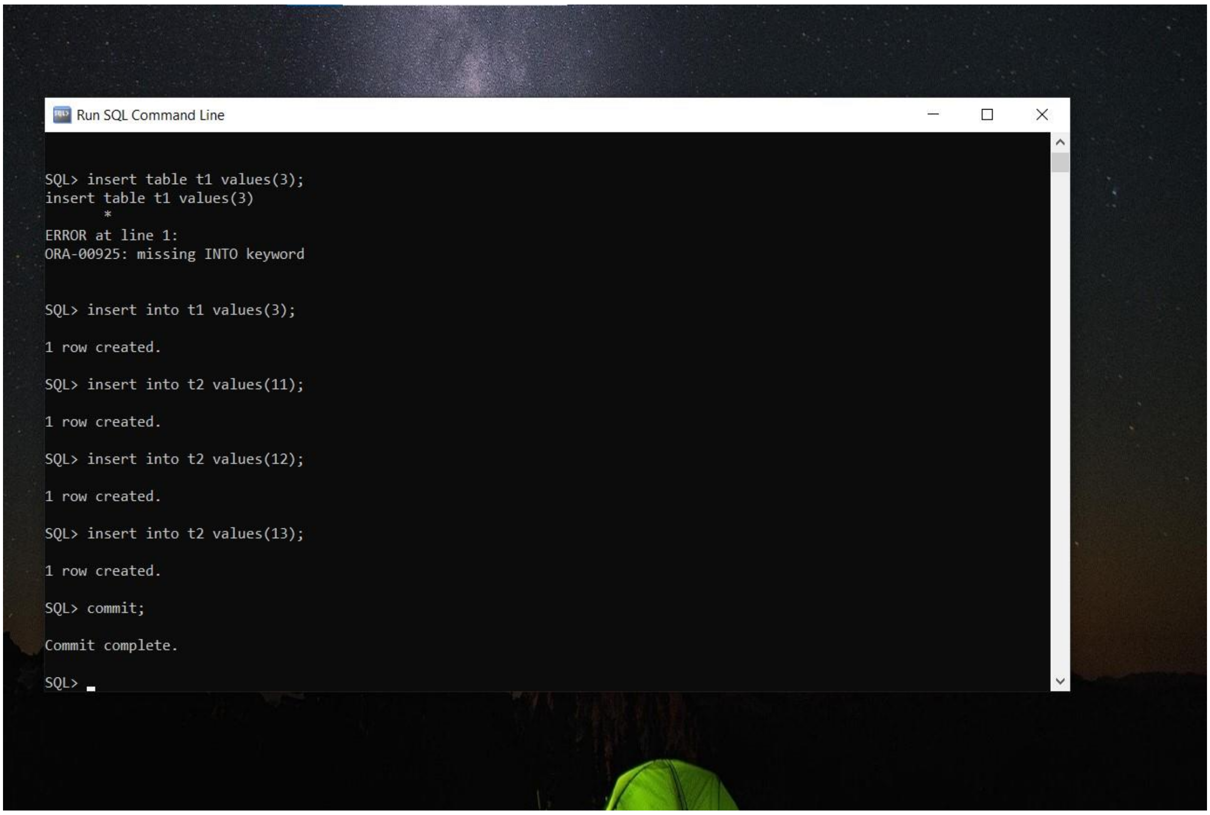
PRACTICAL-11

NAME:- Aniket Prajapati

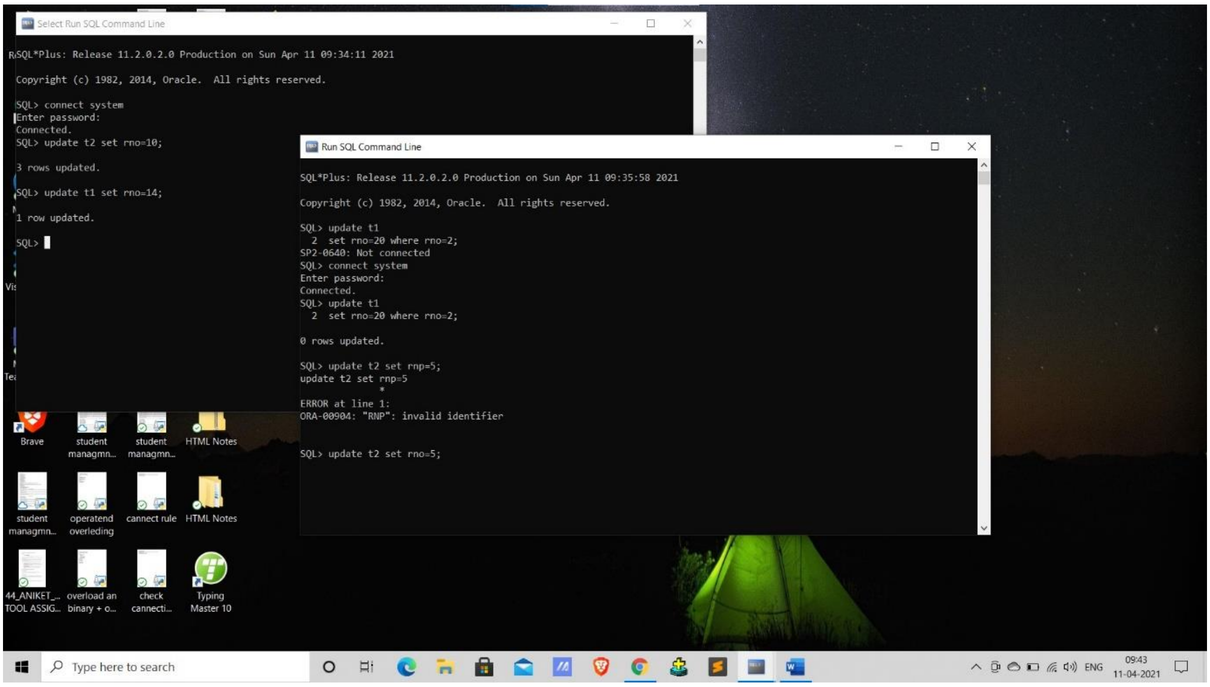
Roll No:-44



```
Enter password:
Connected.
SQL> create table t1(rno int);

Table created.

SQL> create table t2(rno int);

Table created.

SQL> insert into t1 values(1);

1 row created.

SQL> insert into t1 values(2);

1 row created.

SQL> insert into t1 values(3);

1 row created.

SQL> insert into t2 values();
insert into t2 values()
                      *
ERROR at line 1:
ORA-00936: missing expression


SQL> insert into t2 values(1);

1 row created.

SQL> insert into t2 values(2);

1 row created.

SQL> insert into t2 values(3);

1 row created.

SQL> update t1
  2  set rno=5 where rno=2;

1 row updated.

SQL>
```



```
SQL> insert table t1 values(3);
insert table t1 values(3)
       *
ERROR at line 1:
ORA-00925: missing INTO keyword


SQL> insert into t1 values(3);

1 row created.

SQL> insert into t2 values(11);

1 row created.

SQL> insert into t2 values(12);

1 row created.

SQL> insert into t2 values(13);

1 row created.

SQL> commit;

Commit complete.

SQL>
```



```
SQL*Plus: Release 11.2.0.2.0 Production on Sun Apr 11 09:34:11 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect system
Enter password:
Connected.
SQL> update t2 set rno=10;

3 rows updated.

SQL> update t1 set rno=14;

1 row updated.

SQL>
```

```
SQL*Plus: Release 11.2.0.2.0 Production on Sun Apr 11 09:35:58 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> update t1
  2  set rno=20 where rno=2;
SP2-0640: Not connected
SQL> connect system
Enter password:
Connected.
SQL> update t1
  2  set rno=20 where rno=2;

0 rows updated.

SQL> update t2 set rnp=5;
update t2 set rnp=5
              *
ERROR at line 1:
ORA-00904: "RNP": invalid identifier


SQL> update t2 set rno=5;
```