



**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. \_\_\_\_\_IMRAN RIZWAN SHAIKH  
Roll No: \_\_\_\_\_109\_\_\_\_\_ Programme: BSc IT 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 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)**

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

## 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)

```
Run SQL Command Line

SQL> create table Student(sid int constraints pk primary key, sname varchar(20) not null, gender varchar(10) not null, d
ob date not null, remark varchar(20) default 'good', marks int check(marks>50), class varchar(10)not null, email varchar
(30) unique);

Table created.

SQL> desc Student
Name          Null?    Type
-----        -----
SID           NOT NULL NUMBER(38)
SNAME         NOT NULL VARCHAR2(20)
GENDER        NOT NULL VARCHAR2(10)
DOB           NOT NULL DATE
REMARK        VARCHAR2(20)
MARKS          NUMBER(38)
CLASS          NOT NULL VARCHAR2(10)
EMAIL          VARCHAR2(30)
```

b) Course(cid, cname, credits)

```
SQL> create table Course(cid int primary key, cname varchar(20) not null, credits int not null);

Table created.

SQL> desc Course
Name          Null?    Type
-----        -----
CID           NOT NULL NUMBER(38)
CNAME         NOT NULL VARCHAR2(20)
CREDITS       NOT NULL NUMBER(38)

SQL>
```

# PRACTICAL 1

2. Alter the structure of the Course table:

a) Modify datatype of cname.

```
Run SQL Command Line

SQL> desc Course
Name          Null?    Type
-----          -----
CID           NOT NULL NUMBER(38)
CNAME          NOT NULL VARCHAR2(20)
CREDITS        NOT NULL NUMBER(38)

SQL> alter table Course
2  modify cname varchar(30);

Table altered.

SQL> desc Course
Name          Null?    Type
-----          -----
CID           NOT NULL NUMBER(38)
CNAME          NOT NULL VARCHAR2(30)
CREDITS        NOT NULL 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> desc Course
Name          Null?    Type
-----          -----
CID           NOT NULL NUMBER(38)
CNAME          NOT NULL VARCHAR2(30)
CREDITS        NOT NULL NUMBER(38)
COURSEHOURS      NUMBER(38)

SQL>
```

# PRACTICAL 1

## c) Add a column cdesc.

```
Run SQL Command Line

SQL> desc Course
Name          Null?    Type
-----
CID           NOT NULL NUMBER(38)
CNAME          NOT NULL VARCHAR2(30)
CREDITS        NOT NULL NUMBER(38)
COURSEHOURS   NUMBER(38)

SQL> alter table Course
  2 add cdesc varchar(20) not null;

Table altered.

SQL> desc Course
Name          Null?    Type
-----
CID           NOT NULL NUMBER(38)
CNAME          NOT NULL VARCHAR2(30)
CREDITS        NOT NULL NUMBER(38)
COURSEHOURS   NUMBER(38)
CDESC          NOT NULL VARCHAR2(20)

SQL>
```

## 3. Alter the structure of the Student table:

### a) Add a column age with minimum age as 17.

```
Run SQL Command Line

SQL> desc Student;
Name          Null?    Type
-----
SID           NOT NULL NUMBER(38)
SNAME          NOT NULL VARCHAR2(20)
GENDER         NOT NULL VARCHAR2(10)
DOB            NOT NULL DATE
REMARK         VARCHAR2(20)
MARKS          NUMBER(38)
CLASS          NOT NULL VARCHAR2(10)
EMAIL          VARCHAR2(30)

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          NOT NULL VARCHAR2(20)
GENDER         NOT NULL VARCHAR2(10)
DOB            NOT NULL DATE
REMARK         VARCHAR2(20)
MARKS          NUMBER(38)
CLASS          NOT NULL VARCHAR2(10)
EMAIL          VARCHAR2(30)
AGE            NUMBER(38)

SQL>
```

# PRACTICAL 1

b) Delete the column dob.

```
Run SQL Command Line

SQL> desc Student;
Name          Null?    Type
-----
SID           NOT NULL NUMBER(38)
SNAME         NOT NULL VARCHAR2(20)
GENDER        NOT NULL VARCHAR2(10)
DOB           NOT NULL DATE
REMARK        VARCHAR2(20)
MARKS          NUMBER(38)
CLASS          VARCHAR2(10)
EMAIL          VARCHAR2(30)
AGE            NUMBER(38)

SQL> alter table Student
2 drop column dob;

Table altered.

SQL> desc Student;
Name          Null?    Type
-----
SID           NOT NULL NUMBER(38)
SNAME         NOT NULL VARCHAR2(20)
GENDER        NOT NULL VARCHAR2(10)
REMARK        VARCHAR2(20)
MARKS          NUMBER(38)
CLASS          VARCHAR2(10)
EMAIL          VARCHAR2(30)
AGE            NUMBER(38)

SQL>
```

c) Add a column phoneno.

```
Run SQL Command Line

SQL> alter table Student
2 add phoneno int;

Table altered.

SQL> desc Student;
Name          Null?    Type
-----
SID           NOT NULL NUMBER(38)
SNAME         NOT NULL VARCHAR2(20)
GENDER        NOT NULL VARCHAR2(10)
REMARK        VARCHAR2(20)
MARKS          NUMBER(38)
CLASS          VARCHAR2(10)
EMAIL          VARCHAR2(30)
AGE            NUMBER(38)
PHONENO       NUMBER(38)

SQL>
```

# PRACTICAL 1

d) Rename phoneno to contactno.

```
Run SQL Command Line

SQL> alter table Student
  2  rename column phoneno to contactno;

Table altered.

SQL> desc Student;
Name          Null?    Type
-----        -----
SID           NOT NULL NUMBER(38)
SNAME         NOT NULL VARCHAR2(20)
GENDER        NOT NULL VARCHAR2(10)
REMARK        VARCHAR2(20)
MARKS          NUMBER(38)
CLASS          NOT NULL VARCHAR2(10)
EMAIL          VARCHAR2(30)
AGE            NUMBER(38)
CONTACTNO      NUMBER(38)

SQL>
```

4. Rename Student table as Student\_details.

```
Run SQL Command Line

SQL> rename Student to Student_details;

Table renamed.

SQL> desc Student_details;
Name          Null?    Type
-----        -----
SID           NOT NULL NUMBER(38)
SNAME         NOT NULL VARCHAR2(20)
GENDER        NOT NULL VARCHAR2(10)
REMARK        VARCHAR2(20)
MARKS          NUMBER(38)
CLASS          NOT NULL VARCHAR2(10)
EMAIL          VARCHAR2(30)
AGE            NUMBER(38)
CONTACTNO      NUMBER(38)

SQL>
```

# PRACTICAL 1

5) Describe the structure of both tables.

Student\_detail Table

```
SQL> desc Student_details;
Name          Null?    Type
----- -----
SID           NOT NULL NUMBER(38)
SNAME         NOT NULL VARCHAR2(20)
GENDER        NOT NULL VARCHAR2(10)
REMARK        VARCHAR2(20)
MARKS         NUMBER(38)
CLASS         NOT NULL VARCHAR2(10)
EMAIL         VARCHAR2(30)
AGE           NUMBER(38)
CONTACTNO    NUMBER(38)

SQL>
```

Course Table

```
SQL> desc Course
Name          Null?    Type
----- -----
CID           NOT NULL NUMBER(38)
CNAME         NOT NULL VARCHAR2(30)
CREDITS       NOT NULL NUMBER(38)
COURSEHOURS  NUMBER(38)
CODESC        NOT NULL VARCHAR2(20)

SQL>
```

6) Drop the table Student\_detail and Course.

 Run SQL Command Line

```
SQL> drop table Student_details;
Table dropped.

SQL> drop table Course;
Table dropped.

SQL>
```

# PRACTICAL 1

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)

```
Run SQL Command Line

SQL> create table EMPLOYEE(Emp_no int primary key, E_name varchar(20) not null, E_address varchar(30) not null, E_ph_no int, Dept_no int not null, Dept_name varchar(30) not null, Job_id char(10) unique, Salary int check(Salary>=30000));
Table created.

SQL> desc EMPLOYEE
Name          Null?    Type
-----        -----
EMP_NO        NOT NULL NUMBER(38)
E_NAME        NOT NULL VARCHAR2(20)
E_ADDRESS    NOT NULL VARCHAR2(30)
E_PH_NO       NUMBER(38)
DEPT_NO       NOT NULL NUMBER(38)
DEPT_NAME    NOT NULL VARCHAR2(30)
JOB_ID        CHAR(10)
SALARY        NUMBER(38)

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        NOT NULL NUMBER(38)
E_NAME        NOT NULL VARCHAR2(20)
E_ADDRESS    NOT NULL VARCHAR2(30)
E_PH_NO       NUMBER(38)
DEPT_NO       NOT NULL NUMBER(38)
DEPT_NAME    NOT NULL VARCHAR2(30)
JOB_ID        CHAR(10)
SALARY        NUMBER(38)
HIREDATE     DATE
```

# PRACTICAL 1

## 3. Change the datatype of JOB\_ID from char to varchar2.

```
Run SQL Command Line

SQL> desc EMPLOYEE
Name          Null?    Type
-----          -----
EMP_NO        NOT NULL NUMBER(38)
E_NAME        NOT NULL VARCHAR2(20)
E_ADDRESS     NOT NULL VARCHAR2(30)
E_PH_NO       NUMBER(38)
DEPT_NO       NOT NULL NUMBER(38)
DEPT_NAME     NOT NULL VARCHAR2(30)
JOB_ID        CHAR(10)
SALARY        NUMBER(38)
HIREDATE      DATE

SQL> alter table EMPLOYEE
  2 modify Job_id varchar2(10);

Table altered.

SQL> desc EMPLOYEE
Name          Null?    Type
-----          -----
EMP_NO        NOT NULL NUMBER(38)
E_NAME        NOT NULL VARCHAR2(20)
E_ADDRESS     NOT NULL VARCHAR2(30)
E_PH_NO       NUMBER(38)
DEPT_NO       NOT NULL NUMBER(38)
DEPT_NAME     NOT NULL VARCHAR2(30)
JOB_ID        VARCHAR2(10)
SALARY        NUMBER(38)
HIREDATE      DATE

SQL>
```

## 4. Change the name of column/field Emp\_no to E\_no.

```
Run SQL Command Line

SQL> desc EMPLOYEE
Name          Null?    Type
-----          -----
EMP_NO        NOT NULL NUMBER(38)
E_NAME        NOT NULL VARCHAR2(20)
E_ADDRESS     NOT NULL VARCHAR2(30)
E_PH_NO       NUMBER(38)
DEPT_NO       NOT NULL NUMBER(38)
DEPT_NAME     NOT NULL VARCHAR2(30)
JOB_ID        VARCHAR2(10)
SALARY        NUMBER(38)
HIREDATE      DATE

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

Table altered.

SQL> desc EMPLOYEE
Name          Null?    Type
-----          -----
E_NO          NOT NULL NUMBER(38)
E_NAME        NOT NULL VARCHAR2(20)
E_ADDRESS     NOT NULL VARCHAR2(30)
E_PH_NO       NUMBER(38)
DEPT_NO       NOT NULL NUMBER(38)
DEPT_NAME     NOT NULL VARCHAR2(30)
JOB_ID        VARCHAR2(10)
SALARY        NUMBER(38)
HIREDATE      DATE
```

# PRACTICAL 1

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

```
Run SQL Command Line

SQL> desc EMPLOYEE
Name          Null?    Type
-----
E_NO          NOT NULL NUMBER(38)
E_NAME        NOT NULL VARCHAR2(20)
E_ADDRESS     NOT NULL VARCHAR2(30)
E_PH_NO       NUMBER(38)
DEPT_NO       NOT NULL NUMBER(38)
DEPT_NAME     NOT NULL VARCHAR2(30)
JOB_ID        VARCHAR2(10)
SALARY         NUMBER(38)
HIREDATE      DATE

SQL> alter table EMPLOYEE
2  modify Job_id varchar2(20);

Table altered.

SQL> desc EMPLOYEE
Name          Null?    Type
-----
E_NO          NOT NULL NUMBER(38)
E_NAME        NOT NULL VARCHAR2(20)
E_ADDRESS     NOT NULL VARCHAR2(30)
E_PH_NO       NUMBER(38)
DEPT_NO       NOT NULL NUMBER(38)
DEPT_NAME     NOT NULL VARCHAR2(30)
JOB_ID        VARCHAR2(20)
SALARY         NUMBER(38)
HIREDATE      DATE
```

C) Create the following table 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 varchar2(20) primary key, Department_Name varchar2(25));
Table created.

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

SQL>
```

# PRACTICAL 1

Instructor Table: Instructor\_id varchar2(20) primarykey, Department\_Id varchar2(20) Foreign key, 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'.

```
Run SQL Command Line

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

SQL> create table Instructor(Instructor_id varchar2(20) primary key, Department_Id varchar2(20) references Department(Department_Id), Last_Name varchar2(25), First_Name varchar2(200) not null, Telephone varchar2(20) unique, gender char(1) not null, city varchar(10) default 'MUMBAI');

Table created.

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

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              NOT NULL 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	-	-	-	-	✓

# PRACTICAL 1

## OUTPUT:

```
SQL> create table EMP(EMP int primary key ,ENAME varchar2(10) not null, JOB varchar2(9), MGR int,HIREDATE date,SAL number(7,2),Comm int,DEPTNO int,foreign key(DEPTNO)
references DEPT(DEPTNO));
Table created.

SQL> desc EMP;
Name          Null?    Type
-----  -----
EMP           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>
```

## Second Table:

### Table Name: DEPT

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

## OUTPUT:

```
Run SQL Command Line

SQL> create table DEPT(DEPTNO int primary key,DNAME varchar2(14) not null,LOC varchar2(13));

Table created.

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

SQL>
```

# PRACTICAL 2

## PRACTICAL 2: STUDY OF DATA MANIPULATION LANGUAGE STATEMENT

A) Insert the following records in above created table

EMP 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

### OUTPUT:

The screenshot shows the Oracle Database SQL Developer interface. At the top, there's a toolbar with icons for Undo, Redo, Cut, Copy, Paste, Find, Replace, and others. Below the toolbar is a menu bar with File, Edit, View, Tools, Help, and a Database connection dropdown.

The main area displays the following SQL code:

```
create table EMP(EMPNO int primary key, ENAME varchar2(10) not null, JOB varchar(9), MGR int, HIREDATE date, SAL number(7,2), COMM int, DEPTNO int);
desc EMP;
insert into EMP values(7839, 'KING', 'PRESIDENT', null,(to_date('17-Nov-81','dd-mm-yyyy')), 5000, null, 10);
insert into EMP values(7698, 'BLAKE', 'MANAGER', 7839,(to_date('01-May-81','dd-mm-yyyy')), 2850, null, 30);
insert into EMP values(7782, 'CLARK', 'MANAGER', 7839,(to_date('09-Jun-81','dd-mm-yyyy')), 2450, null, 10);
insert into EMP values(7566, 'JONES', 'MANAGER', 7839,(to_date('02-Apr-81','dd-mm-yyyy')), 2975, null, 10);
insert into EMP values(7788, 'SCOTT', 'ANALYST', 7566,(to_date('19-Apr-87','dd-mm-yyyy')), 3000, null, 20);
insert into EMP values(7902, 'FORD', 'ANALYST', 7566,(to_date('03-Dec-81','dd-mm-yyyy')), 3000, null, 20);
insert into EMP values(7369, 'SMITH', 'CLERK', 7902,(to_date('17-Dec-80','dd-mm-yyyy')), 800, null, 20);
insert into EMP values(7499, 'ALLEN', 'SALESMAN', 7698,(to_date('20-Feb-81','dd-mm-yyyy')), 1600, 300, 30);
insert into EMP values(7521, 'WARD', 'SALESMAN', 7698,(to_date('22-Feb-81','dd-mm-yyyy')), 1250, 500, 30);
insert into EMP values(7654, 'MARTIN', 'SALESMAN', 7698,(to_date('28-Sep-81','dd-mm-yyyy')), 1600, 1400, 30);
insert into EMP values(7844, 'TURNER', 'SALESMAN', 7698,(to_date('08-Sep-81','dd-mm-yyyy')), 1500, 0, 30);
```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, showing the following table output:

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	1600	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

At the bottom left, it says "14 rows returned in 0.00 seconds". At the bottom right, there's a "CSV Export" link.

# PRACTICAL 2

## DEPT TABLE

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

## OUTPUT:

 Run SQL Command Line

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

SQL> insert into DEPT values(10, 'ACCOUNTING', 'NEW YORK');

1 row created.

SQL> insert into DEPT values(20, 'RESEARCH', 'DALLAS');

1 row created.

SQL> insert into DEPT values(30, 'SALES', 'CHICAGO');

1 row created.

SQL> insert into DEPT values(40, 'OPERATIONS', 'BOSTON');

1 row created.

SQL> COMMIT;

Commit complete.

SQL> select * from DEPT;

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

SQL>
```

# PRACTICAL 2

## B) Update and Delete Queries.

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

 Run SQL Command Line

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

```
4 rows updated.
```

```
SQL> SELECT * FROM EMP;
```

EMP	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81-11-17	5000		10
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7566	JONES	MANAGER	7839	81-04-02	2975		20
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7902	FORD	ANALYST	7566	81-12-03	3000		20
7369	SMITH	CLERK	7902	80-12-17	1300		20
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
EMP	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	87-05-23	1600		20
7900	JAMES	CLERK	7698	81-12-03	1450		30
7934	MILLER	CLERK	7782	82-01-23	1800		10

```
14 rows selected.
```

```
SQL>
```

# PRACTICAL 2

## 2) Update the manager of James as CLARK.

```
Run SQL Command Line

SQL> update EMP
  2 set MGR=7839
  3 where ENAME='JAMES';

1 row updated.

SQL> select * from EMP;

EMP ENAME      JOB          MGR HIREDATE    SAL     COMM  DEPTNO
---- -----
 7839 KING       PRESIDENT   81-11-17  5000      10
 7698 BLAKE      MANAGER    7839 81-05-01  2850      30
 7782 CLARK      MANAGER    7839 81-06-09  2450      10
 7566 JONES      MANAGER    7839 81-04-02  2975      20
 7788 SCOTT      ANALYST    7566 87-04-19  3000      20
 7902 FORD       ANALYST    7566 81-12-03  3000      20
 7369 SMITH      CLERK      7902 80-12-17  1300      20
 7499 ALLEN      SALESMAN   7698 81-02-20  1600      300     30
 7521 WARD       SALESMAN   7698 81-02-22  1250      500     30
 7654 MARTIN    SALESMAN   7698 81-09-28  1250      1400    30
 7844 TURNER    SALESMAN   7698 81-09-08  1500      0       30

EMP ENAME      JOB          MGR HIREDATE    SAL     COMM  DEPTNO
---- -----
 7876 ADAMS      CLERK      7788 87-05-23  1600      20
 7900 JAMES      CLERK      7839 81-12-03  1450      30
 7934 MILLER    CLERK      7782 82-01-23  1800      10

14 rows selected.

SQL>
```

## 3) Change the role of Miller as Manager.

```
Run SQL Command Line

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

1 row updated.

SQL> select * from EMP;

EMP ENAME      JOB          MGR HIREDATE    SAL     COMM  DEPTNO
---- -----
 7839 KING       PRESIDENT   81-11-17  5000      10
 7698 BLAKE      MANAGER    7839 81-05-01  2850      30
 7782 CLARK      MANAGER    7839 81-06-09  2450      10
 7566 JONES      MANAGER    7839 81-04-02  2975      20
 7788 SCOTT      ANALYST    7566 87-04-19  3000      20
 7902 FORD       ANALYST    7566 81-12-03  3000      20
 7369 SMITH      CLERK      7902 80-12-17  1300      20
 7499 ALLEN      SALESMAN   7698 81-02-20  1600      300     30
 7521 WARD       SALESMAN   7698 81-02-22  1250      500     30
 7654 MARTIN    SALESMAN   7698 81-09-28  1250      1400    30
 7844 TURNER    SALESMAN   7698 81-09-08  1500      0       30

EMP ENAME      JOB          MGR HIREDATE    SAL     COMM  DEPTNO
---- -----
 7876 ADAMS      CLERK      7788 87-05-23  1600      20
 7900 JAMES      CLERK      7839 81-12-03  1450      30
 7934 MILLER    MANAGER    7782 82-01-23  1800      10

14 rows selected.

SQL>
```

## PRACTICAL 2

4) Delete the records of Manager.

 Run SQL Command Line

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

4 rows deleted.

```
SQL> select * from EMP;
```

EMP	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81-11-17	5000		10
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7902	FORD	ANALYST	7566	81-12-03	3000		20
7369	SMITH	CLERK	7902	80-12-17	1300		20
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
7876	ADAMS	CLERK	7788	87-05-23	1600		20
7900	JAMES	CLERK	7839	81-12-03	1450		30

10 rows selected.

SQL>

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

 Run SQL Command Line

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

10 rows deleted.

```
SQL> select * from EMP;
```

no rows selected

SQL>

**PRACTICAL 3**

A) Using emp table, perform the following queries:

1) *Display the details of all employees.*

**OUTPUT:**

Run SQL Command Line

SQL> select \* from EMP;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81-11-17	5000		10
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7566	JONES	MANAGER	7839	81-04-02	2975		20
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7902	FORD	ANALYST	7566	81-12-03	3000		20
7369	SMITH	CLERK	7902	80-12-17	800		20
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
7876	ADAMS	CLERK	7788	87-05-23	1100		20
7900	JAMES	CLERK	7698	81-12-03	950		30
7934	MILLER	CLERK	7788	82-01-23	1300		10

14 rows selected.

2) *Display the name and job for all employees.*

**OUTPUT:**

Run SQL Command Line

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
TURNER	SALESMAN
ADAMS	CLERK
JAMES	CLERK
MILLER	CLERK

14 rows selected.

**3) Display name and salary for all employees.**
 Run SQL Command Line

```
SQL> select ENAME,SAL as SALARY from EMP;
```

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

14 rows selected.

**4) Display the details of all employees who are earning salary greater than 2000.**
 Run SQL Command Line

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

EMP	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81-11-17	5000		10
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7566	JONES	MANAGER	7839	81-04-02	2975		20
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7902	FORD	ANALYST	7566	81-12-03	3000		20

6 rows selected.

**5) Display the details of all employees who are working as Manager.**
 Run SQL Command Line

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

EMP	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7566	JONES	MANAGER	7839	81-04-02	2975		20

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

 Run SQL Command Line

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

NAME

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

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

 Run SQL Command Line

```
SQL> select ENAME as NAME from EMP  
2 where (JOB='CLERK' and SAL>3000);
```

```
no rows selected
```

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

 Run SQL Command Line

```
SQL> select EMPNO,ENAME as NAME from EMP  
2 where COMM is not null;
```

EMPNO NAME

```
-----  
7499 ALLEN  
7521 WARD  
7654 MARTIN  
7844 TURNER
```

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

 Run SQL Command Line

```
SQL> select EMPNO,ENAME as NAME from EMP  
2 where COMM is null;
```

EMPNO NAME

```
-----  
7839 KING  
7698 BLAKE  
7782 CLARK  
7566 JONES  
7788 SCOTT  
7902 FORD  
7369 SMITH  
7876 ADAMS  
7900 JAMES  
7934 MILLER
```

```
10 rows selected.
```

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

 Run SQL Command Line

```
SQL> select ENAME as NAME from EMP  
  2 where JOB in('CLERK','SALESMAN','ANALYST') and SAL>2000;
```

NAME

-----

```
SCOTT  
FORD
```

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

 Run SQL Command Line

```
SQL> select ENAME as NAME from EMP  
  2 where JOB in('CLERK','SALESMAN','ANALYST');
```

NAME

-----

```
SCOTT  
FORD  
SMITH  
ALLEN  
WARD  
MARTIN  
TURNER  
ADAMS  
JAMES  
MILLER
```

```
10 rows selected.
```

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

Run SQL Command Line

```
SQL> select ENAME as NAME from EMP
  2 where DEPTNO in(10,20,30);

NAME
-----
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
ALLEN
WARD
MARTIN
TURNER

NAME
-----
ADAMS
JAMES
MILLER

14 rows selected.
```

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

Run SQL Command Line

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

EMPNO ENAME      JOB          MGR HIREDATE    SAL     COMM  DEPTNO
----- -----
 7499 ALLEN      SALESMAN    7698 81-02-20   1600    300   30
 7521 WARD       SALESMAN    7698 81-02-22   1250    500   30
 7654 MARTIN    SALESMAN    7698 81-09-28   1250   1400   30
 7844 TURNER    SALESMAN    7698 81-09-08   1500     0    30
 7876 ADAMS      CLERK       7788 87-05-23   1100
 7934 MILLER    CLERK       7782 82-01-23   1300

6 rows selected.
```

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

Run SQL Command Line

```
SQL> select * from EMP
  2 order by SAL asc;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80-12-17	800		20
7900	JAMES	CLERK	7698	81-12-03	950		30
7876	ADAMS	CLERK	7788	87-05-23	1100		20
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30
7934	MILLER	CLERK	7782	82-01-23	1300		10
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7566	JONES	MANAGER	7839	81-04-02	2975		20
7902	FORD	ANALYST	7566	81-12-03	3000		20
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7839	KING	PRESIDENT		81-11-17	5000		10

14 rows selected.

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

Run SQL Command Line

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

no rows selected

```
SQL>
```

- 16) List the employees who are either 'CLERK' or 'ANALYST' in the Desc order.

Run SQL Command Line

```
SQL> select * from EMP
  2 where JOB in('CLERK','ANALYST')
  3 order by JOB desc;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80-12-17	800		20
7900	JAMES	CLERK	7698	81-12-03	950		30
7934	MILLER	CLERK	7782	82-01-23	1300		10
7876	ADAMS	CLERK	7788	87-05-23	1100		20
7902	FORD	ANALYST	7566	81-12-03	3000		20
7788	SCOTT	ANALYST	7566	87-04-19	3000		20

6 rows selected.

## 17) List the employees who are working in Deptno 10 or 20.

 Run SQL Command Line

```
SQL> select * from EMP
2 where DEPTNO in(10,20);
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81-11-17	5000		10
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7566	JONES	MANAGER	7839	81-04-02	2975		20
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7902	FORD	ANALYST	7566	81-12-03	3000		20
7369	SMITH	CLERK	7902	80-12-17	800		20
7876	ADAMS	CLERK	7788	87-05-23	1100		20
7934	MILLER	CLERK	7782	82-01-23	1300		10

8 rows selected.

## 18) List the employees whose name have a character set 'll' together.

 Run SQL Command Line

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

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7934	MILLER	CLERK	7782	82-01-23	1300		10

## 19) List the employees in ascending order of their names.

 Run SQL Command Line

```
SQL> select * from EMP
2 order by ENAME asc;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	87-05-23	1100		20
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7902	FORD	ANALYST	7566	81-12-03	3000		20
7900	JAMES	CLERK	7698	81-12-03	950		30
7566	JONES	MANAGER	7839	81-04-02	2975		20
7839	KING	PRESIDENT		81-11-17	5000		10
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7934	MILLER	CLERK	7782	82-01-23	1300		10
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7369	SMITH	CLERK	7902	80-12-17	800		20
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30

14 rows selected.

**20) List the employees in descending order of their names.**

Run SQL Command Line

```
SQL> select * from EMP
  2 order by ENAME desc;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
7369	SMITH	CLERK	7902	80-12-17	800		20
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7934	MILLER	CLERK	7782	82-01-23	1300		10
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7839	KING	PRESIDENT		81-11-17	5000		10
7566	JONES	MANAGER	7839	81-04-02	2975		20
7900	JAMES	CLERK	7698	81-12-03	950		30
7902	FORD	ANALYST	7566	81-12-03	3000		20
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7876	ADAMS	CLERK	7782	87-05-23	1100		20

14 rows selected.

**21) List the employees who do not belong to Deptno 20.**

Run SQL Command Line

```
SQL> select * from EMP
  2 where DEPTNO <>20;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81-11-17	5000		10
7698	BLAKE	MANAGER	7839	81-05-01	2850		30
7782	CLARK	MANAGER	7839	81-06-09	2450		10
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
7900	JAMES	CLERK	7698	81-12-03	950		30
7934	MILLER	CLERK	7782	82-01-23	1300		10

9 rows selected.

**22) List all the employees except PRESIDENT and MANAGER.**

Run SQL Command Line

```
SQL> select * from EMP
  2 where JOB <> 'PRESIDENT' and JOB <> 'MANAGER';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	87-04-19	3000		20
7902	FORD	ANALYST	7566	81-12-03	3000		20
7369	SMITH	CLERK	7902	80-12-17	800		20
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	81-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	81-09-08	1500	0	30
7876	ADAMS	CLERK	7782	87-05-23	1100		20
7900	JAMES	CLERK	7698	81-12-03	950		30
7934	MILLER	CLERK	7782	82-01-23	1300		10

10 rows selected.

**23) List the employees whose name starts with A.**

Run SQL Command Line

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

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	81-02-20	1600	300	30
7876	ADAMS	CLERK	7788	87-05-23	1100		20

**24) List all the Clerks of Deptno 20.**

Run SQL Command Line

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

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80-12-17	800		20
7876	ADAMS	CLERK	7788	87-05-23	1100		20

**25) List the employees whose names ends with S.**

Run SQL Command Line

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

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	JONES	MANAGER	7839	81-04-02	2975		20
7876	ADAMS	CLERK	7788	87-05-23	1100		20
7900	JAMES	CLERK	7698	81-12-03	950		30

**26) List the employees who has name of exactly 4 characters.**

Run SQL Command Line

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

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81-11-17	5000		10
7902	FORD	ANALYST	7566	81-12-03	3000		20
7521	WARD	SALESMAN	7698	81-02-22	1250	500	30

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

 Run SQL Command Line

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

- 28) List the total salary of employees working as ANALYST.

 Run SQL Command Line

```
SQL> select sum(SAL) from EMP
  2 where JOB='ANALYST';
 
SUM(SAL)
-----
6000
```

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

 Run SQL Command Line

```
SQL> select min(SAL) from EMP;
 
MIN(SAL)
-----
800

SQL> select max(SAL) from EMP;
 
MAX(SAL)
-----
5000

SQL> select avg(SAL) from EMP;
 
AVG(SAL)
-----
2073,21429
```

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

 Run SQL Command Line

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

```
COUNT (DEPTNO)
```

```
-----  
3
```

B) Answer the following queries:

- 1) Display the total salary of employees department wise.

 Run SQL Command Line

```
SQL> select DEPTNO,sum(SAL)  
2 from EMP  
3 group by DEPTNO;
```

DEPTNO	SUM(SAL)
30	9400
20	10875
10	8750

```
SQL>
```

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

Run SQL Command Line

```
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

SQL>
```

3) Display the total number of employees with specific job.

Run SQL Command Line

```
SQL> select distinct job,count(*) from EMP
  2 group by JOB;

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

SQL> -
```

4) Display the total number of employees working in each department.

Run SQL Command Line

```
SQL> select distinct deptno,count(*) from EMP
  2 group by DEPTNO;

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

SQL> ■
```

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

Run SQL Command Line

```
SQL> select JOB,DEPTNO,sum(SAL) from EMP
  2 group by JOB,DEPTNO
  3 order by JOB;

  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

  8 rows selected.

SQL> ■
```

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

```
Run SQL Command Line  
SQL> select JOB,count(JOB),sum(SAL)  
  2  from EMP  
  3  group by JOB having count(*)>1;  
  
JOB      COUNT(JOB)    SUM(SAL)  
-----  -----  
CLERK          4      4150  
SALESMAN        4      5600  
MANAGER         3      8275  
ANALYST         2      6000  
  
SQL>
```

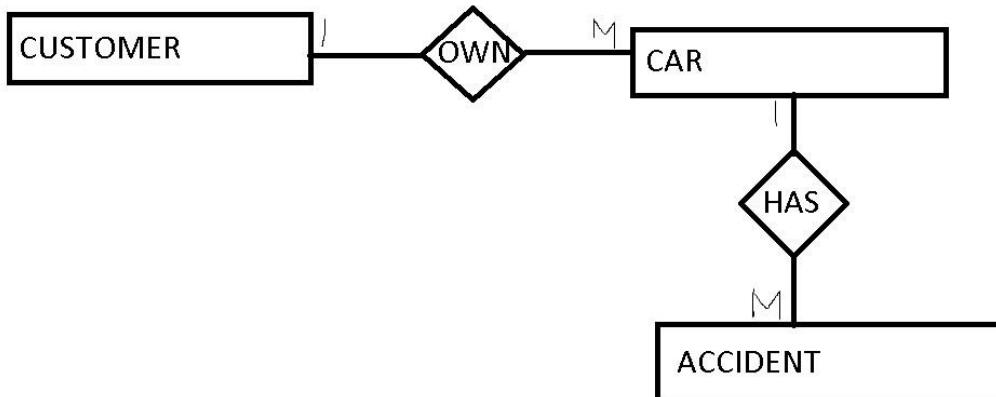
### 7) Display unique jobs of employees.

```
Run SQL Command Line  
SQL> select unique JOB from EMP;  
  
JOB  
-----  
CLERK  
SALESMAN  
PRESIDENT  
MANAGER  
ANALYST  
  
SQL> -
```

Questions on ER diagram

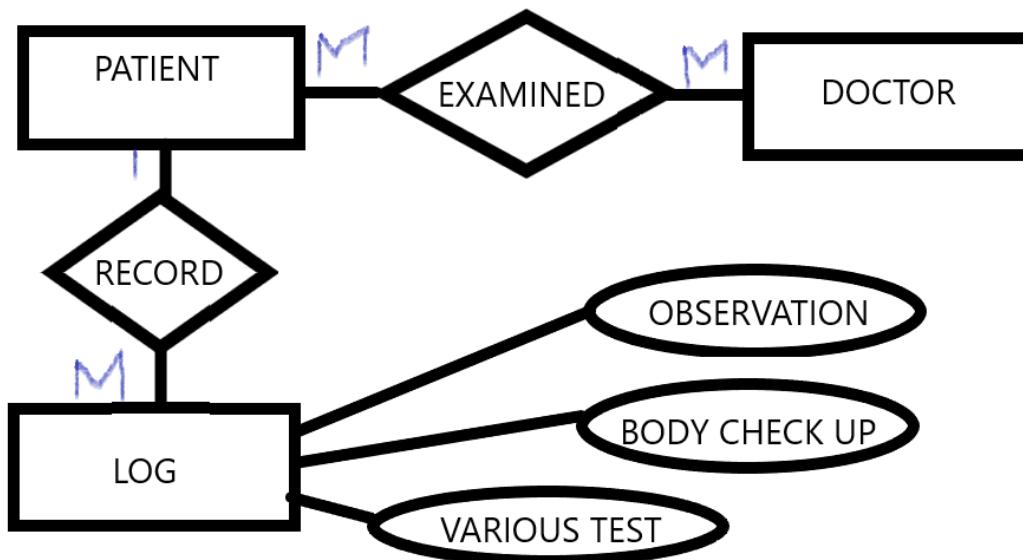
1. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

ER DIAGRAM:



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.

ER DIAGRAM:



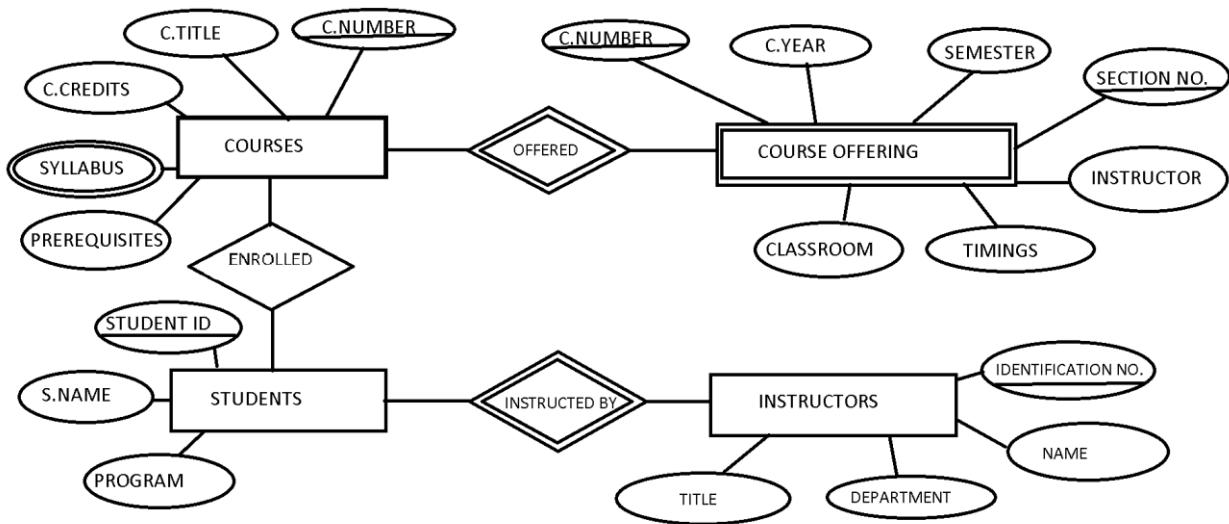
3. A university registrar's office maintains data about the following entities: (a)courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including

course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification 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 office .Document all assumptions that you make about the mapping constraints.

**E-R DIAGRAM:**

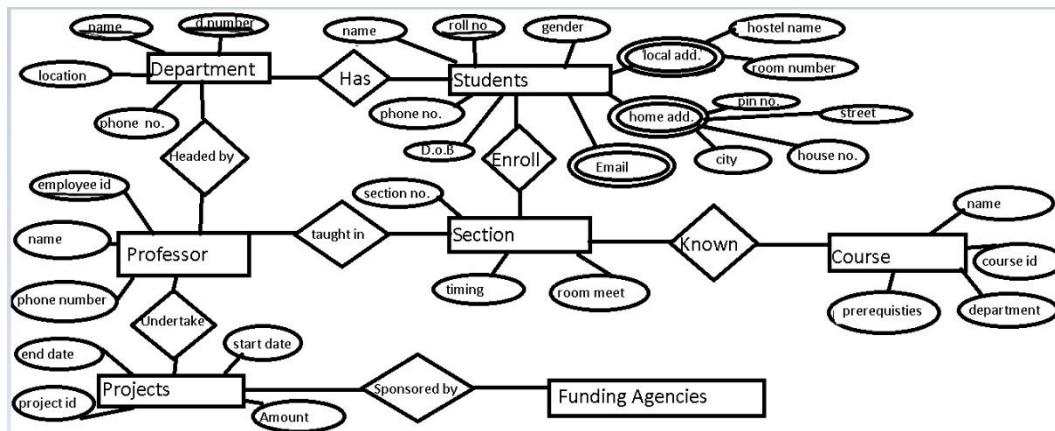


4. Draw the ER diagram for the given scenario

- 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 prerequisites 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 project id.

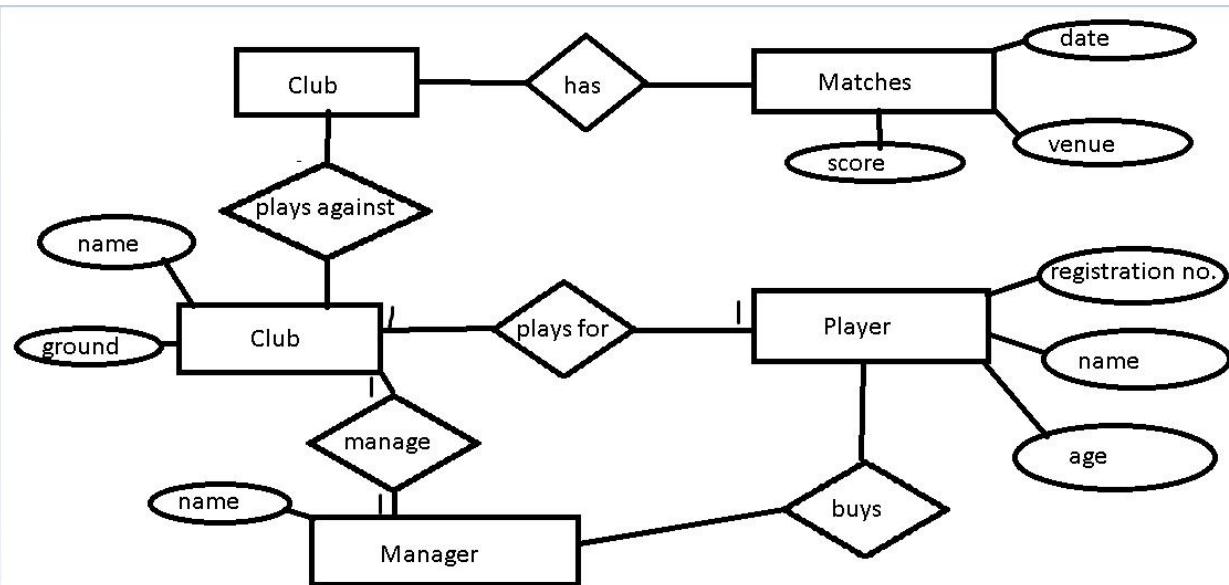
**ER DIAGRAM:**



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.

**ER DIAGRAM :**



# IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 5 DBMS

## PRACTICAL 5 Study of various type of JOINS

### 1. Inner join

```
SQL> select ENAME,SAL,LOC from EMP inner join DEPT  
2  on EMP.DEPTNO=DEPT.DEPTNO;  
  
ENAME          SAL   LOC  
-----  
KING           5000 NEW YORK  
BLAKE          2850 CHICAGO  
CLARK          2450 NEW YORK  
JONES          2975 DALLAS  
SCOTT          3000 DALLAS  
FORD            3000 DALLAS  
SMITH           800  DALLAS  
ALLEN          1600 CHICAGO  
WARD            1250 CHICAGO  
MARTIN          1250 CHICAGO  
TURNER          1500 CHICAGO  
ADAMS           1100 DALLAS  
JAMES            950  CHICAGO  
MILLER          1300 NEW YORK  
  
14 rows selected.
```

### 2. Outer join

#### right outer join:

```
SQL> select EMPNO, ENAME,SAL, imr.DEPTNO, DEPT.DEPTNO,LOC from imr right outer join DEPT  
2  on imr.DEPTNO=DEPT.DEPTNO;  
  
EMPNO ENAME      SAL    DEPTNO    DEPTNO LOC  
-----  
7839 KING        5000     10       10  NEW YORK  
7698 BLAKE       2850     30       30  CHICAGO  
7782 CLARK       2450     10       10  NEW YORK  
7566 JONES       2975     20       20  DALLAS  
7788 SCOTT       3000     20       20  DALLAS  
7902 FORD         3000     20       20  DALLAS  
7369 SMITH        800      20       20  DALLAS  
7499 ALLEN        1600     30       30  CHICAGO  
7521 WARD         1250     30       30  CHICAGO  
7654 MARTIN       1250     30       30  CHICAGO  
7844 TURNER       1500     30       30  CHICAGO  
7876 ADAMS        1100     20       20  DALLAS  
7900 JAMES         950      30       30  CHICAGO  
7934 MILLER       1300     10       10  NEW YORK  
                                40  BOSTON  
  
15 rows selected.
```

#### left outer join:

```
SQL> select EMPNO, ENAME,SAL, imr.DEPTNO, DEPT.DEPTNO,LOC from imr left outer join DEPT  
2  on imr.DEPTNO=DEPT.DEPTNO;  
  
EMPNO ENAME      SAL    DEPTNO    DEPTNO LOC  
-----  
7839 KING        5000     10       10  NEW YORK  
7698 BLAKE       2850     30       30  CHICAGO  
7782 CLARK       2450     10       10  NEW YORK  
7566 JONES       2975     20       20  DALLAS  
7788 SCOTT       3000     20       20  DALLAS  
7902 FORD         3000     20       20  DALLAS  
7369 SMITH        800      20       20  DALLAS  
7499 ALLEN        1600     30       30  CHICAGO  
7521 WARD         1250     30       30  CHICAGO  
7654 MARTIN       1250     30       30  CHICAGO  
7844 TURNER       1500     30       30  CHICAGO  
7876 ADAMS        1100     20       20  DALLAS  
7900 JAMES         950      30       30  CHICAGO  
7934 MILLER       1300     10       10  NEW YORK  
  
14 rows selected.
```

# IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 5 DBMS

## Full outer join:

```
SQL> select EMPNO, ENAME, SAL, imr.DEPTNO, DEPT.DEPTNO, LOC from imr full outer join DEPT
2 on imr.DEPTNO=DEPT.DEPTNO;

EMPNO ENAME      SAL   DEPTNO    DEPTNO LOC
-----  -----
7839 KING        5000    10        10  NEW YORK
7698 BLAKE       2850    30        30  CHICAGO
7782 CLARK       2450    10        10  NEW YORK
7566 JONES       2975    20        20  DALLAS
7788 SCOTT       3000    20        20  DALLAS
7902 FORD         3000    20        20  DALLAS
7369 SMITH        800     20        20  DALLAS
7499 ALLEN        1600    30        30  CHICAGO
7521 WARD         1250    30        30  CHICAGO
7654 MARTIN       1250    30        30  CHICAGO
7844 TURNER       1500    30        30  CHICAGO
7876 ADAMS        1100    20        20  DALLAS
7900 JAMES         950     30        30  CHICAGO
7934 MILLER       1300    10        10  NEW YORK
                                         40  BOSTON

15 rows selected.
```

## 3. Natural join

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

DEPTNO  EMPNO ENAME      JOB          MGR HIREDATE      SAL   COMM  DNAME      LOC
-----  -----
10      7839 KING    PRESIDENT    7839 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 TURNER   SALESMAN   7698 08-SEP-81  1500  0  SALES    CHICAGO
20      7876 ADAMS   CLERK      7788 23-MAY-87  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.
```

# IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 5 DBMS

## 4. Cross join

EMPLNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10	10	ACCOUNTING	NEW YORK
7848	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
7802	FORD	ANALYST	7566	03-DEC-81	3000	-	20	10	ACCOUNTING	NEW YORK
7869	SMITH	CLERK	7802	17-DEC-80	600	-	20	10	ACCOUNTING	NEW YORK
7888	ALLEN	SALESMAN	7888	20-FEB-81	1600	300	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7888	22-FEB-81	1250	500	30	10	ACCOUNTING	NEW YORK
7654	MARTIN	SALESMAN	7888	28-SEP-81	1250	1400	30	10	ACCOUNTING	NEW YORK
7844	TURNER	SALESMAN	7888	08-SEP-81	1500	0	30	10	ACCOUNTING	NEW YORK
7876	ADAMS	CLERK	7708	23-MAY-87	1100	-	20	10	ACCOUNTING	NEW YORK
7900	JAMES	CLERK	7888	03-DEC-81	950	-	30	10	ACCOUNTING	NEW YORK
7834	MILLER	CLERK	7702	23-JAN-82	1300	-	10	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10	20	RESEARCH	DALLAS
7888	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
7802	FORD	ANALYST	7566	03-DEC-81	3000	-	20	20	RESEARCH	DALLAS
7869	SMITH	CLERK	7802	17-DEC-80	600	-	20	20	RESEARCH	DALLAS
7888	ALLEN	SALESMAN	7888	20-FEB-81	1600	300	30	20	RESEARCH	DALLAS
7521	WARD	SALESMAN	7888	22-FEB-81	1250	500	30	20	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	7888	28-SEP-81	1250	1400	30	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7888	08-SEP-81	1500	0	30	20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7708	23-MAY-87	1100	-	20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7888	03-DEC-81	950	-	30	20	RESEARCH	DALLAS
7834	MILLER	CLERK	7702	23-JAN-82	1300	-	10	20	RESEARCH	DALLAS
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10	30	SALES	CHICAGO
7888	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
7802	FORD	ANALYST	7566	03-DEC-81	3000	-	20	30	SALES	CHICAGO
7869	SMITH	CLERK	7802	17-DEC-80	600	-	20	30	SALES	CHICAGO
7888	ALLEN	SALESMAN	7888	20-FEB-81	1600	300	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7888	22-FEB-81	1250	500	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7888	28-SEP-81	1250	1400	30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7888	08-SEP-81	1500	0	30	30	SALES	CHICAGO
7876	ADAMS	CLERK	7708	23-MAY-87	1100	-	20	30	SALES	CHICAGO
7900	JAMES	CLERK	7888	03-DEC-81	950	-	30	30	SALES	CHICAGO
7834	MILLER	CLERK	7702	23-JAN-82	1300	-	10	30	SALES	CHICAGO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10	40	OPERATIONS	BOSTON
7888	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
7802	FORD	ANALYST	7566	03-DEC-81	3000	-	20	40	OPERATIONS	BOSTON
7869	SMITH	CLERK	7802	17-DEC-80	600	-	20	40	OPERATIONS	BOSTON
7888	ALLEN	SALESMAN	7888	20-FEB-81	1600	300	30	40	OPERATIONS	BOSTON
7521	WARD	SALESMAN	7888	22-FEB-81	1250	500	30	40	OPERATIONS	BOSTON
7654	MARTIN	SALESMAN	7888	28-SEP-81	1250	1400	30	40	OPERATIONS	BOSTON
7844	TURNER	SALESMAN	7888	08-SEP-81	1500	0	30	40	OPERATIONS	BOSTON
7876	ADAMS	CLERK	7708	23-MAY-87	1100	-	20	40	OPERATIONS	BOSTON
7900	JAMES	CLERK	7888	03-DEC-81	950	-	30	40	OPERATIONS	BOSTON
7834	MILLER	CLERK	7702	23-JAN-82	1300	-	10	40	OPERATIONS	BOSTON

56 rows returned in 0.02 seconds [City Export](#)

## 5. Self join

```
SQL> select b.ename Employee,a.ename Manager from emp a,emp b  
2 where a.empno=b.mgr;  
  
EMPLOYEE      MANAGER  
-----  
BLAKE          KING  
CLARK          KING  
JONES          KING  
SCOTT          JONES  
FORD           JONES  
SMITH          FORD  
ALLEN          BLAKE  
WARD           BLAKE  
MARTIN         BLAKE  
TURNER         BLAKE  
ADAMS          SCOTT  
JAMES          BLAKE  
MILLER         CLARK  
  
13 rows selected.
```

## **SINGLE ROW FUNCTIONS**

### **1) Numeric**

Abs: ‘ABSOLUTE’-gives you the positive number in return

```
SQL> select abs(5) from dual;
      ABS(5)
-----
      5

SQL> select abs(-5) from dual;
      ABS(-5)
-----
      5

SQL>
```

Ceil :-gives you the greater whole number of the given specified number



The screenshot shows a SQL command line interface window titled "Run SQL Command Line". It contains two separate SQL statements. The first statement selects the ceiling of 12, resulting in 12. The second statement selects the ceiling of 12.56, resulting in 13. Both statements use the CEIL function and are run against a "dual" table.

```
Run SQL Command Line

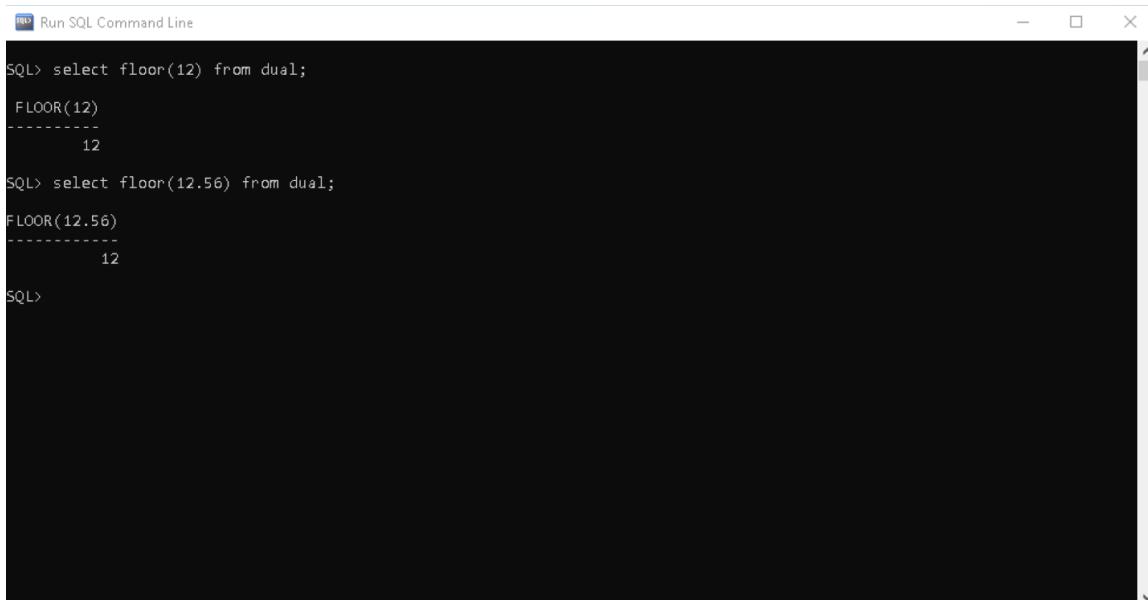
SQL> select ceil(12) from dual;
      CEIL(12)
-----
      12

SQL> select ceil(12.56) from dual;
      CEIL(12.56)
-----
      13

SQL>
```

Floor:-gives you the smaller whole number of the given specified number

\*CEIL and FLOOR specially works on decimal numbers\*



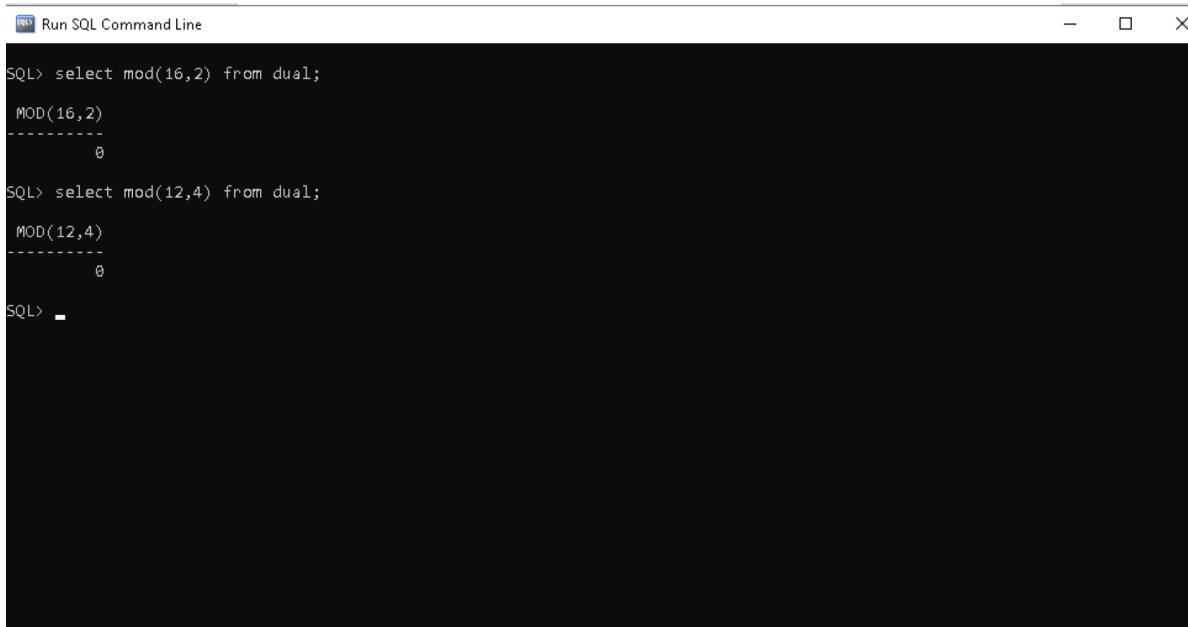
```
Run SQL Command Line
SQL> select floor(12) from dual;
FLOOR(12)
-----
12
SQL> select floor(12.56) from dual;
FLOOR(12.56)
-----
12
SQL>
```

SQRT: ‘SQUARE ROOT’-gives you the square root of a given number



```
Run SQL Command Line
SQL> select sqrt(4) from dual;
SQRT(4)
-----
2
SQL> select sqrt(25) from dual;
SQRT(25)
-----
5
SQL> -
```

MOD:-gives the remainder of the given numbers and it takes two parameters



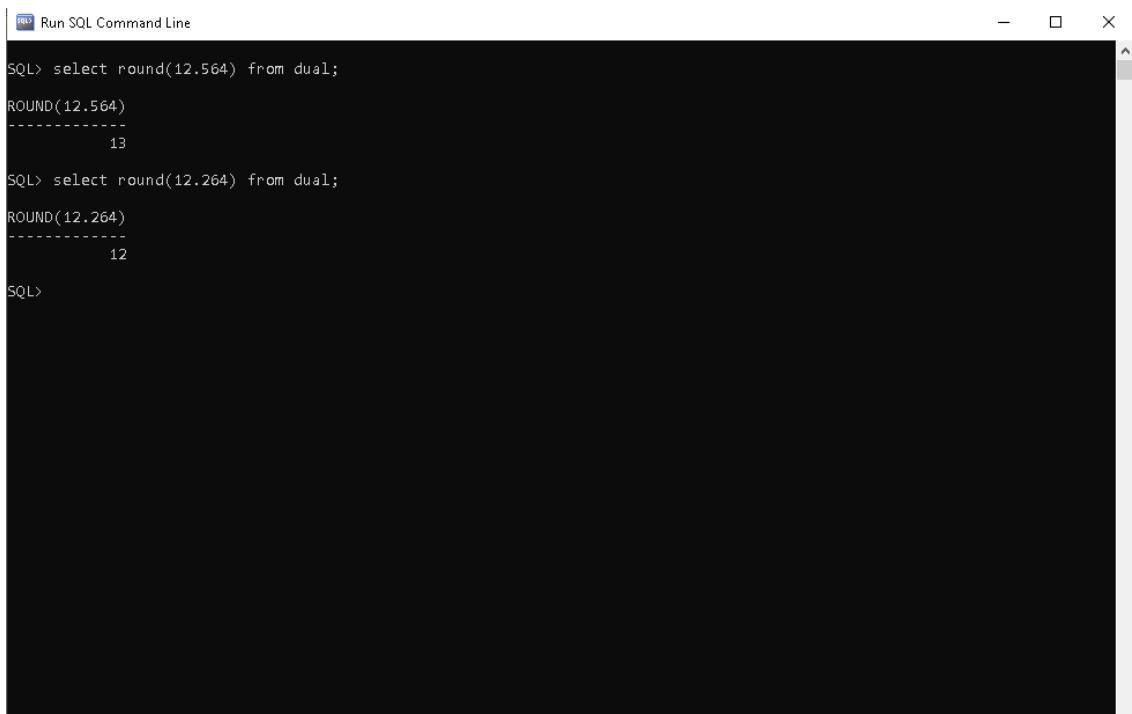
```
Run SQL Command Line

SQL> select mod(16,2) from dual;
MOD(16,2)
-----
0

SQL> select mod(12,4) from dual;
MOD(12,4)
-----
0

SQL>
```

ROUND:-gives you the round off of the number



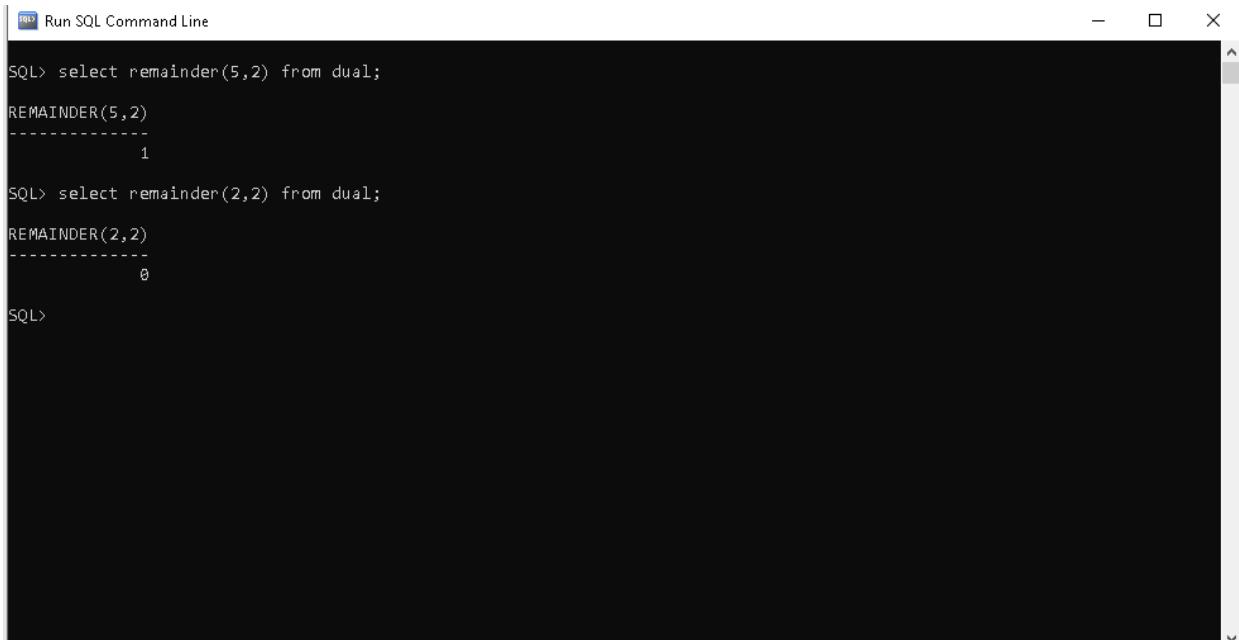
```
Run SQL Command Line

SQL> select round(12.564) from dual;
ROUND(12.564)
-----
13

SQL> select round(12.264) from dual;
ROUND(12.264)
-----
12

SQL>
```

**REMAINDER:** gives the remainder of the given numbers and it takes two parameters



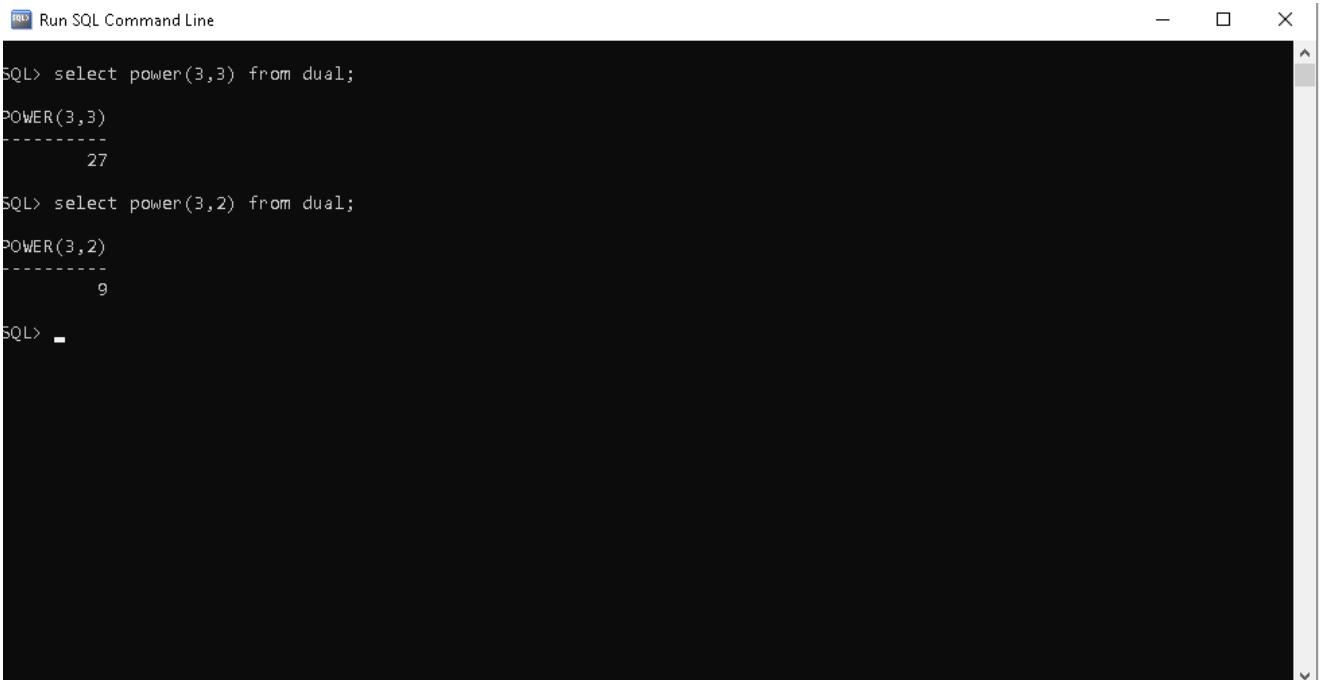
```
Run SQL Command Line

SQL> select remainder(5,2) from dual;
REMAINDER(5,2)
-----
1

SQL> select remainder(2,2) from dual;
REMAINDER(2,2)
-----
0

SQL>
```

**POWER:** gives you the power of the given number



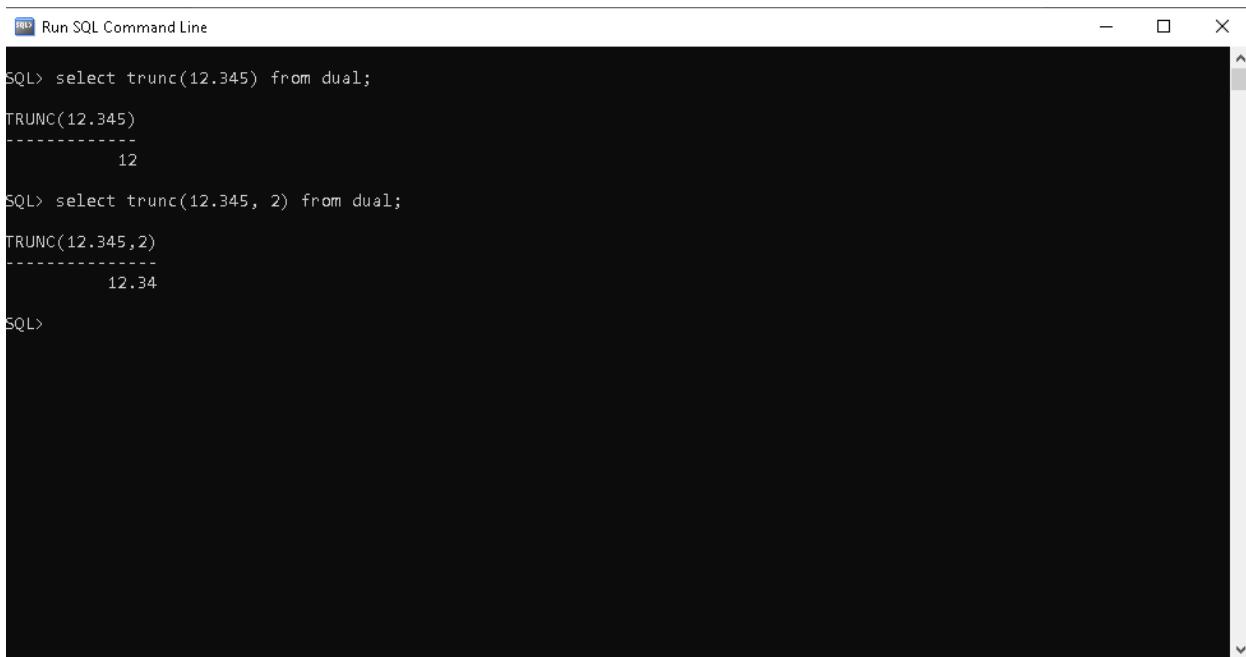
```
Run SQL Command Line

SQL> select power(3,3) from dual;
POWER(3,3)
-----
27

SQL> select power(3,2) from dual;
POWER(3,2)
-----
9

SQL>
```

TRUNC: ‘TRUNCET’-eliminate the values after decimal



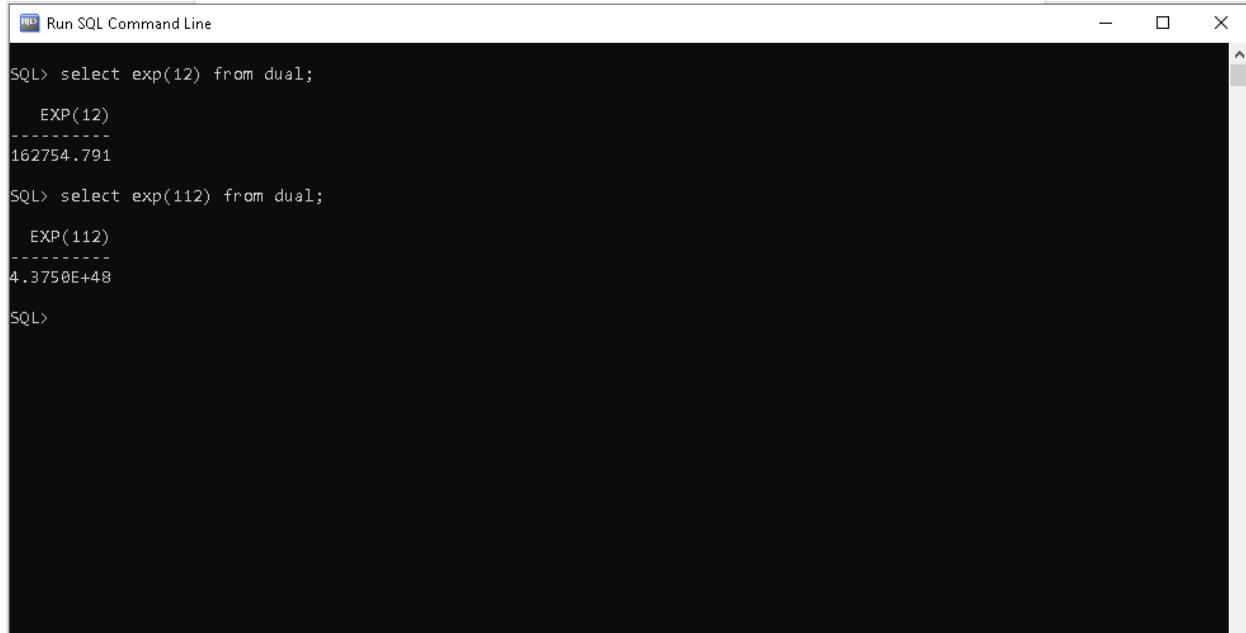
```
Run SQL Command Line

SQL> select trunc(12.345) from dual;
TRUNC(12.345)
-----
12

SQL> select trunc(12.345, 2) from dual;
TRUNC(12.345,2)
-----
12.34

SQL>
```

EXP: gives you the exponential of the number



```
Run SQL Command Line

SQL> select exp(12) from dual;
EXP(12)
-----
162754.791

SQL> select exp(112) from dual;
EXP(112)
-----
4.3750E+48

SQL>
```

## 2) Character

Lower: makes the string to lowercase

```
Run SQL Command Line

SQL> select lower('MY NAME IS IMRAN') from dual;
LOWER('MYNAMEISI
-----
my name is imran

SQL> select lower('tHIIs IS mY DbmS ProjEcT') from dual;
LOWER('THISISMYDBMSPROJ
-----
this is my dbms project

SQL>
```

Upper : makes the string to uppercase

```
Run SQL Command Line

SQL> select upper('my name is imran') from dual;
UPPER('MYNAMEISI
-----
MY NAME IS IMRAN

SQL> select upper('tHIIs IS mY DbmS ProjEcT') from dual;
UPPER('THISISMYDBMSPROJ
-----
THIS IS MY DBMS PROJECT

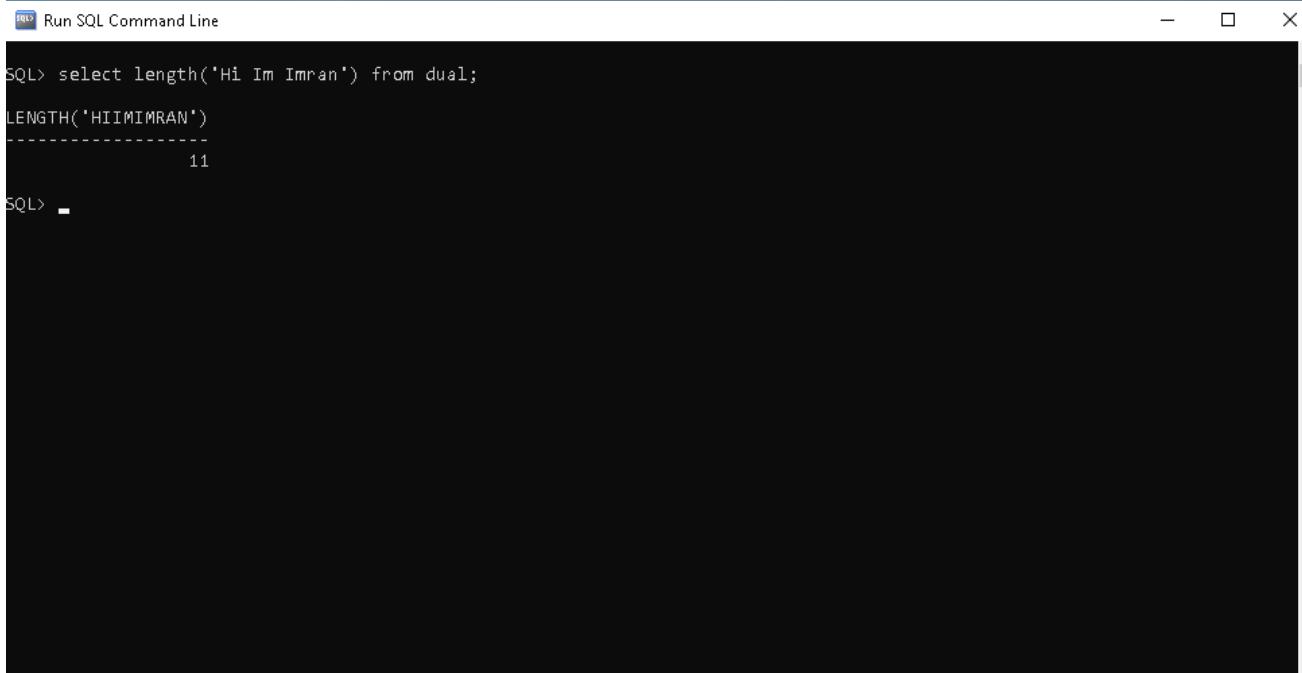
SQL> -
```

Initcap: makes the first letter capital



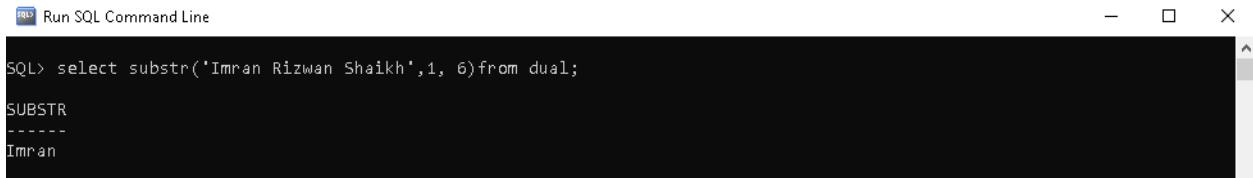
```
Run SQL Command Line
SQL> select initcap('i am imRan') from dual;
INITCAP('I
-----
I Am Imran
SQL>
```

Length: display the total length of the given string



```
Run SQL Command Line
SQL> select length('Hi Im Imran') from dual;
LENGTH('HIIMIMRAN')
-----
11
SQL>
```

Substr: takes three parameters and display specific string part



```
Run SQL Command Line
SQL> select substr('Imran Rizwan Shaikh',1, 6)from dual;
SUBSTR
-----
Imran
```

Concat: used to joins the given different string

```
SQL> select concat('How are you', ' Maam') from dual;
CONCAT('HOWAREYOU
-----
How are you Maam
```

Instr: used to finds the given word in string

```
SQL> select instr('How are you Maam','are') from dual;
INSTR('HOWAREYOUAMAAM','ARE')
-----
5
```

Trim: reduces the spaces in the string



The screenshot shows a Windows-style application window titled "Run SQL Command Line". Inside, three SQL queries are run against the "dual" table:

- The first query uses TRIM(LEADING) to remove leading spaces from the string "How are you Maam". The result is "How are you Maam".
- The second query uses TRIM(BOTH) to remove both leading and trailing spaces from the string "How are you Maam ". The result is "How are you Maam".
- The third query uses TRIM(TRAILING) to remove trailing spaces from the string "How are you Maam ". The result is "How are you Maam".

RTRIM: reduces the spaces in the string from right

LTRIM: reduces the spaces in the string from left

```
Run SQL Command Line

SQL> select rtrim ('How are you Maam      ') from dual;
RTRIM('HOWAREYOU
-----
How are you Maam

SQL> select rtrim ('How are you Maam   Im Imran   ') from dual;
RTRIM('HOWAREYOUAMAAMIMIMRAN'
-----
How are you Maam   Im Imran

SQL> select ltrim ('      How are you Maam   Im Imran   ') from dual;
LTRIM('HOWAREYOUAMAAMIMIMRAN')
-----
How are you Maam   Im Imran

SQL> select ltrim ('      I am Imran How are you all') from dual;
LTRIM('IAMIMRANHOWAREYOUA
-----
I am Imran How are you all

SQL>
```

**TRANSLATE:** used to change the letter by new letter

**REPLACE:** used to replace the whole word

```
Run SQL Command Line

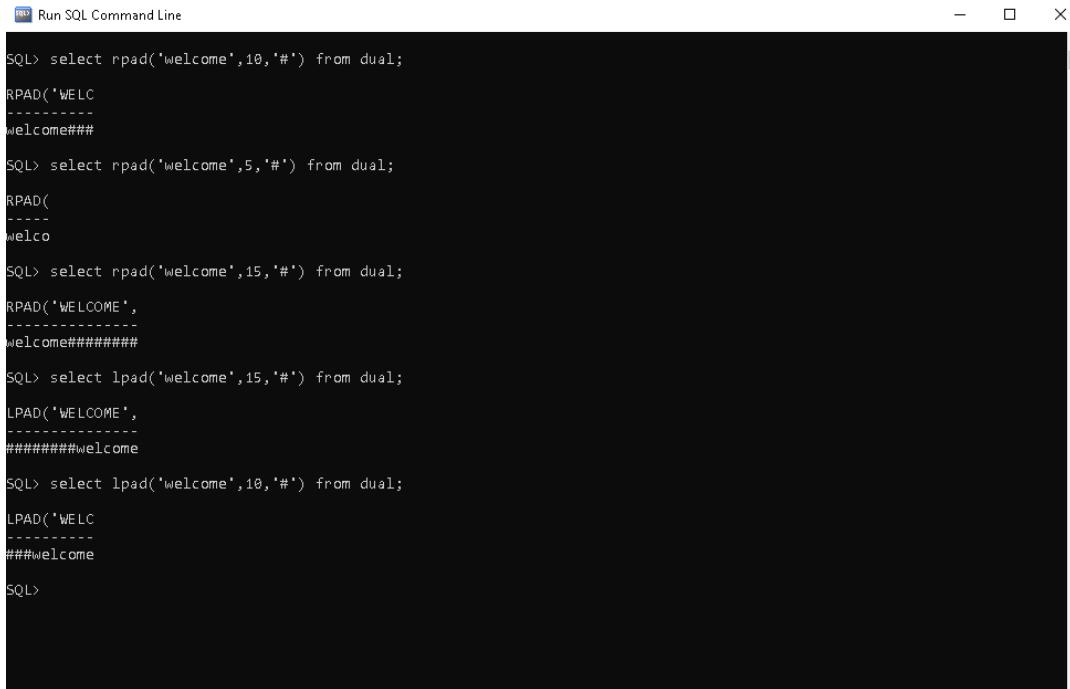
SQL> select translate('Imran is a good boy','iag','@#$') from dual;
TRANSLATE('IMRANISA
-----
Imr#n @s # $ood boy

SQL> select replace('Imran is a good boy','oo','$') from dual;
REPLACE('IMRANISAGO
-----
Imran is a g$$d boy

SQL>
```

**RPAD:** use to give space using symbols toward right

**LPAD:** use to give space using symbols towards left



```
Run SQL Command Line

SQL> select rpad('welcome',10,'#') from dual;
RPAD('WELC
-----
welcome###

SQL> select rpad('welcome',5,'#') from dual;
RPAD(
-----
welco

SQL> select rpad('welcome',15,'#') from dual;
RPAD('WELCOME',
-----
welcome#####

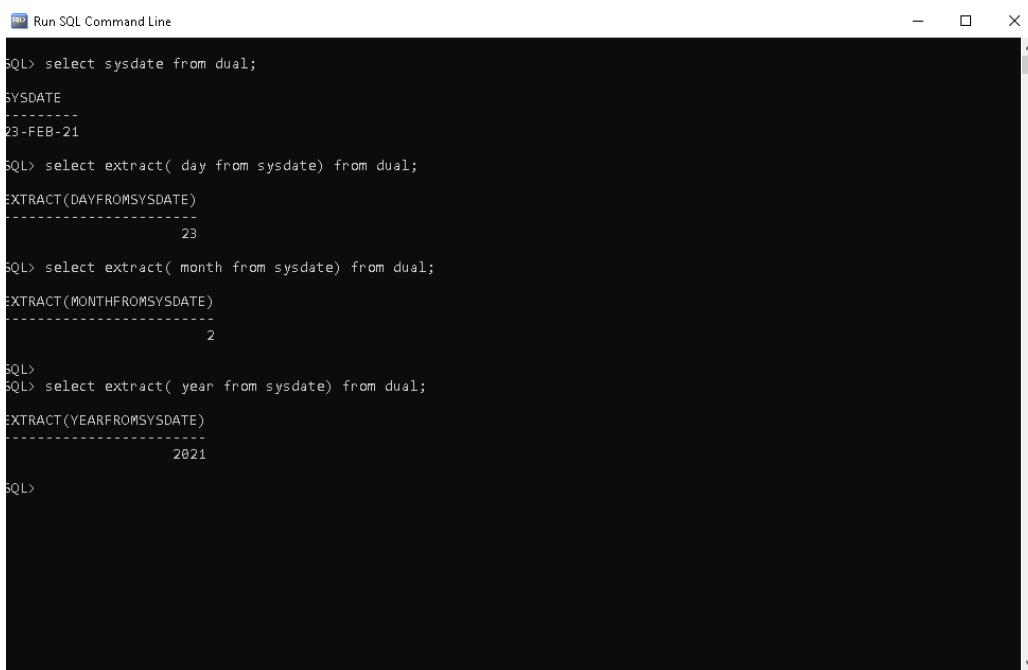
SQL> select lpad('welcome',15,'#') from dual;
LPAD('WELCOME',
-----
#####
welcome

SQL> select lpad('welcome',10,'#') from dual;
LPAD('WELC
-----
###welcome

SQL>
```

### 3) Date Function

**SYSDATE:** display the current date



```
Run SQL Command Line

SQL> select sysdate from dual;
SYSDATE
-----
23-FEB-21

SQL> select extract( day from sysdate) from dual;
EXTRACT(DAYFROMSYSDATE)
-----
23

SQL> select extract( month from sysdate) from dual;
EXTRACT(MONTHFROMSYSDATE)
-----
2

SQL> select extract( year from sysdate) from dual;
EXTRACT(YEARFROMSYSDATE)
-----
2021

SQL>
```

## NEXT\_DAY: used to find next days

```
Run SQL Command Line

SQL> select next_day(sysdate, 'sunday') from dual;
NEXT_DAY(
-----
28-FEB-21

SQL> select next_day(sysdate, 'Friday') from dual;
NEXT_DAY(
-----
26-FEB-21

SQL>
```

## LAST\_DAY: used to find last day

```
Run SQL Command Line

SQL> select last_day(sysdate) from dual;
LAST_DAY(
-----
28-FEB-21

SQL>
```

## ADD\_MONTHS: used to increase the months

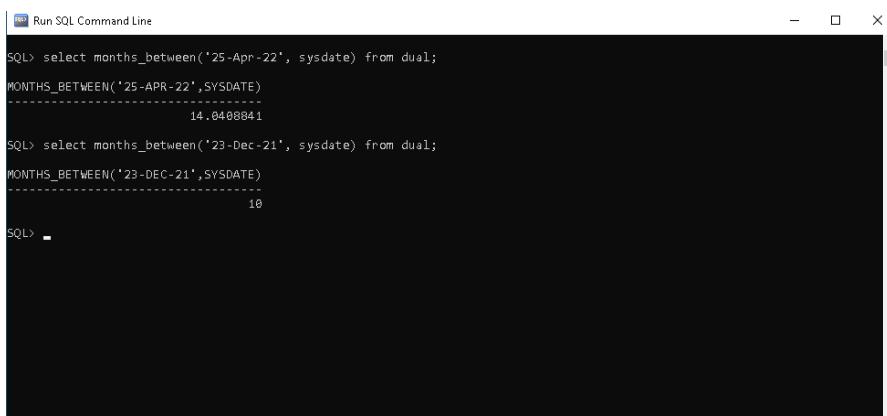
```
Run SQL Command Line

SQL> select add_months(sysdate, 3) from dual;
ADD_MONTH
-----
23-MAY-21

SQL> select add_months(sysdate, 8) from dual;
ADD_MONTH
-----
23-OCT-21

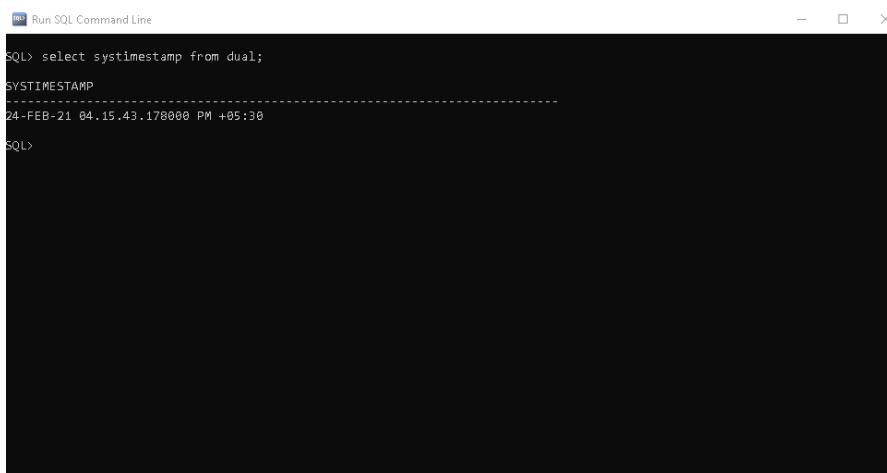
SQL>
```

**MONTHS\_BETWEEN:** used to find number of months between two dates



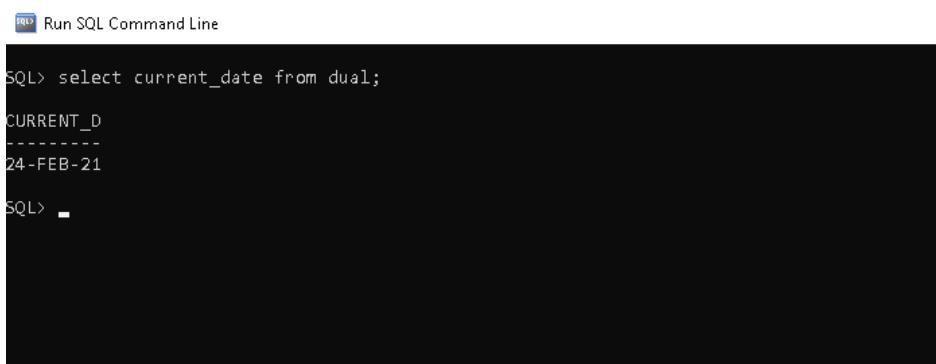
```
Run SQL Command Line
SQL> select months_between('25-Apr-22', sysdate) from dual;
MONTHS_BETWEEN('25-APR-22',SYSDATE)
-----
14.0408841
SQL> select months_between('23-Dec-21', sysdate) from dual;
MONTHS_BETWEEN('23-DEC-21',SYSDATE)
-----
10
SQL>
```

**SYSTIMESTAMP:** used to set time format



```
Run SQL Command Line
SQL> select systimestamp from dual;
SYSTIMESTAMP
-----
24-FEB-21 04.15.43.178000 PM +05:30
SQL>
```

**CURRENT\_DATE:** used to know current date



```
Run SQL Command Line
SQL> select current_date from dual;
CURRENT_DATE
-----
24-FEB-21
SQL>
```

## PRACTICAL 7: STUDY OF VARIOUS TYPES OF SET OPERATORS

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.)

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

```
Run SQL Command Line

SQL> create table Vendor(VEND_CODE int primary key);
Table created.

SQL> insert into Vendor values(123);
1 row created.

SQL> insert into Vendor values(124);
1 row created.

SQL> insert into Vendor values(125);
1 row created.

SQL> insert into Vendor values(126);
1 row created.

SQL> commit;
Commit complete.

SQL> select * from Vendor;
VEND_CODE
-----
123
124
125
126
```

# IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 7 DBMS



```
Run SQL Command Line
SQL> create table Product(PROD_CODE varchar2(10) primary key,VEND_CODE int not null,foreign key(VEND_CODE) references Vendor(VEND_CODE));
Table created.

SQL> Insert into Product values('ABC',125);
1 row created.

SQL> Insert into Product values('DEF',124);
1 row created.

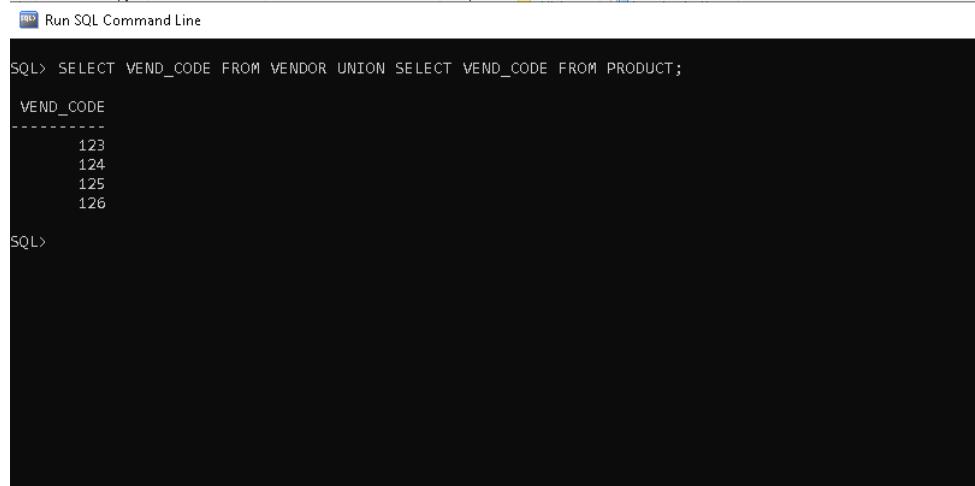
SQL> Insert into Product values('GHE',124);
1 row created.

SQL> Insert into Product values('JKL',123);
1 row created.

SQL> SELECT * FROM Product;
PROD_CODE      VEND_CODE
-----          -----
ABC            125
DEF            124
GHE            124
JKL            123

SQL>
```

a) A UNION query based on these two tables

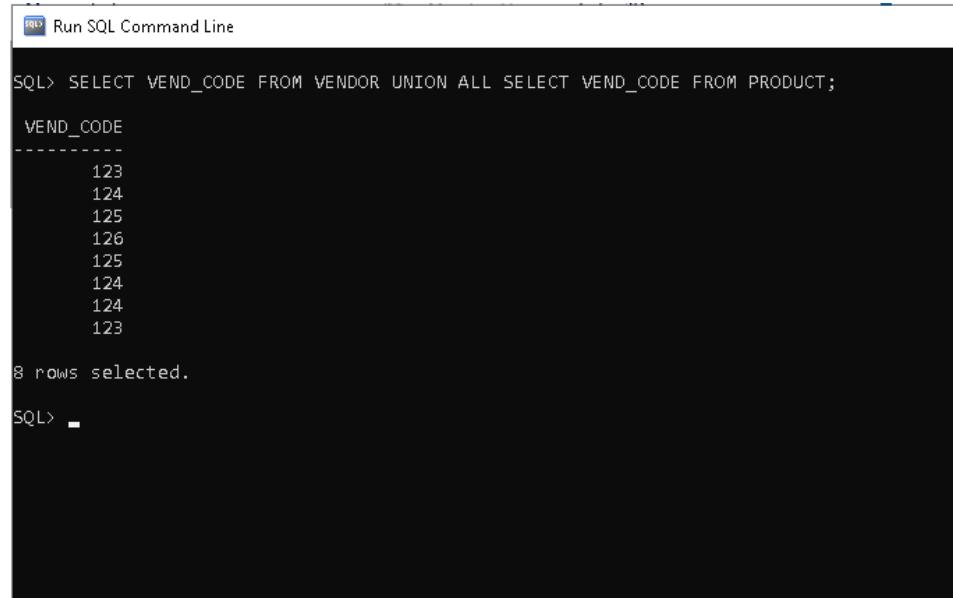


```
Run SQL Command Line
SQL> SELECT VEND_CODE FROM VENDOR UNION SELECT VEND_CODE FROM PRODUCT;

VEND_CODE
-----
123
124
125
126

SQL>
```

b) A UNION ALL query based on these two tables



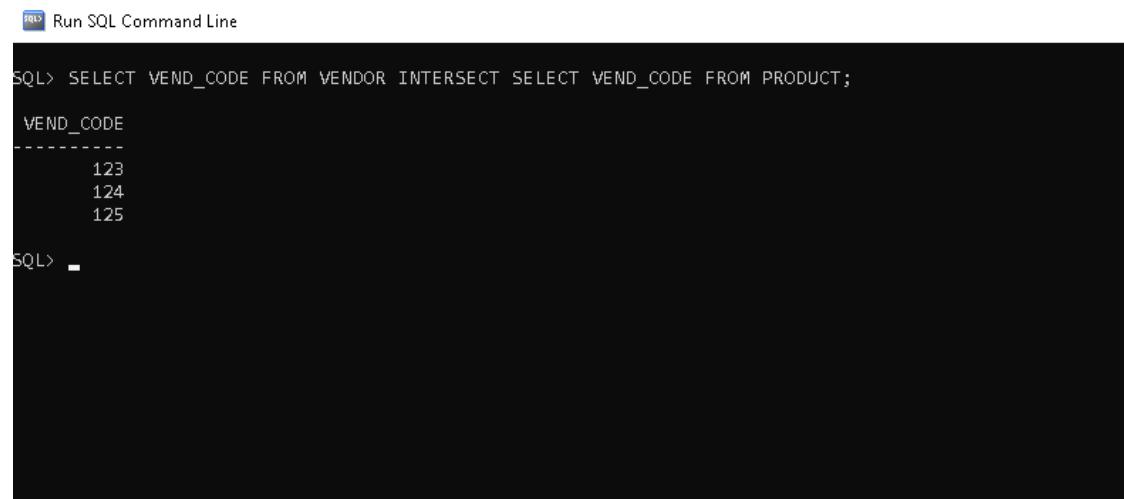
```
Run SQL Command Line
SQL> SELECT VEND_CODE FROM VENDOR UNION ALL SELECT VEND_CODE FROM PRODUCT;

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

8 rows selected.

SQL>
```

c) An INTERSECT query based on these two tables



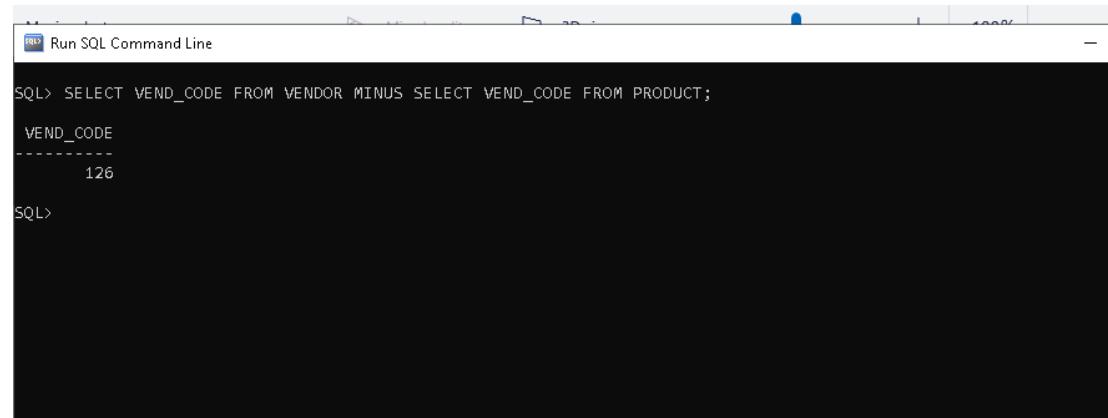
Run SQL Command Line

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

VEND_CODE
-----
123
124
125

SQL>
```

d) A MINUS query based on these two tables



Run SQL Command Line

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

VEND_CODE
-----
126

SQL>
```

**PRACTICAL 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’.**

 Run SQL Command Line

```
SQL> create view emp_hor as select * from EMP2
  2 where JOB='ANALYST';

View created.

SQL> select * from emp_hor;

  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

SQL>
```

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

## **IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 8 DBMS**

```
SQL> create view vwemp as select ENAME, JOB, SAL from EMP2;
View created.

SQL> select * from vwemp;

ENAME      JOB          SAL
-----      -----
KING        PRESIDENT    5000
BLAKE       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
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 group1(JOB,COUNT_EMP) as select JOB, count(*) from EMP2
  2 group by JOB;

View created.

SQL> select * from group1;

JOB          COUNT_EMP
-----          -----
CLERK            4
SALESMAN         4
PRESIDENT        1
MANAGER          3
ANALYST          2

SQL>
```

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

```
SQL> create view view1 as select ENAME, DNAME, LOC
  2  from EMP inner join DEPT
  3  on EMP.DEPTNO = DEPT.DEPTNO;

View created.

SQL> select * from view1;

ENAME      DNAME          LOC
-----      -----          -----
KING        ACCOUNTING    NEWYORK
BLAKE       SALES         CHICAGO
CLARK       ACCOUNTING    NEWYORK
JONES       RESEARCH     DALLAS
SCOTT       RESEARCH     DALLAS
FORD        RESEARCH     DALLAS
SMITH       RESEARCH     DALLAS
ALLEN       SALES         CHICAGO
WARD        SALES         CHICAGO
MARTIN      SALES         CHICAGO
TURNER      SALES         CHICAGO
ADAMS       RESEARCH     DALLAS
JAMES        SALES         CHICAGO
MILLER      ACCOUNTING    NEWYORK

14 rows selected.
```

## **IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 8 DBMS**

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

```
SQL> create view view2 as select ENAME, SAL, JOB from EMP2
  2 where DEPTNO=20;

View created.

SQL> select * from view2;

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

**6. Delete all the views created above.**

```
SQL> drop view emp_hor;

View dropped.

SQL> drop view vwemp;

View dropped.

SQL> drop view group1;

View dropped.

SQL> drop view view1;

View dropped.

SQL> drop view view2;

View dropped.

SQL>
```

## **PRACTICAL 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 * from EMP where SAL > (select SAL from EMP where EMPNO= 7566);
```

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

```
SQL>
```

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

```
SQL> select ename,sal,job from emp2 where job = (select job from emp2 where empno = 7369);
```

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

```
SQL>
```

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

```
SQL> select ename, sal from emp2 where sal < any(select sal from emp2 where job = 'CLERK');

ENAME          SAL
-----
SMITH          800
JAMES          950
ADAMS          1100
MARTIN         1250
WARD           1250

SQL>
```

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, empno, job from emp2 where job = (select job from emp2 where empno = 7900);

ENAME          SAL      EMPNO JOB
-----
SMITH          800     7369  CLERK
ADAMS          1100    7876  CLERK
JAMES          950     7900  CLERK
MILLER         1300    7934  CLERK

SQL>
```

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

Run SQL Command Line

```
SQL> select * from emp2 inner join dept2 on emp2.deptno = dept2.deptno where mgr in(select mgr from emp2 where mgr=7698);
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
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	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	30	SALES	CHICAGO
7900	JAMES	CLERK	7698	03-DEC-81	950		30	30	SALES	CHICAGO

```
SQL>
```

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

```
SQL> select * from emp2 where sal > any(select sal from emp2 where job ='MANAGER');
```

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
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7698	BLAKE	MANAGER	7839	01-MAY-31	2850		30

```
SQL>
```

## PRACTICAL 10: STUDY OF TRANSACTION (COMMIT/ROLLBACK), LOCKS

1. Perform Commit and Rollback on a table.

**COMMIT:** Commit is a type of save command. Using commit we can save our changes permanently. When we any write function we use commit. If we don't do commit, the data will be lost.

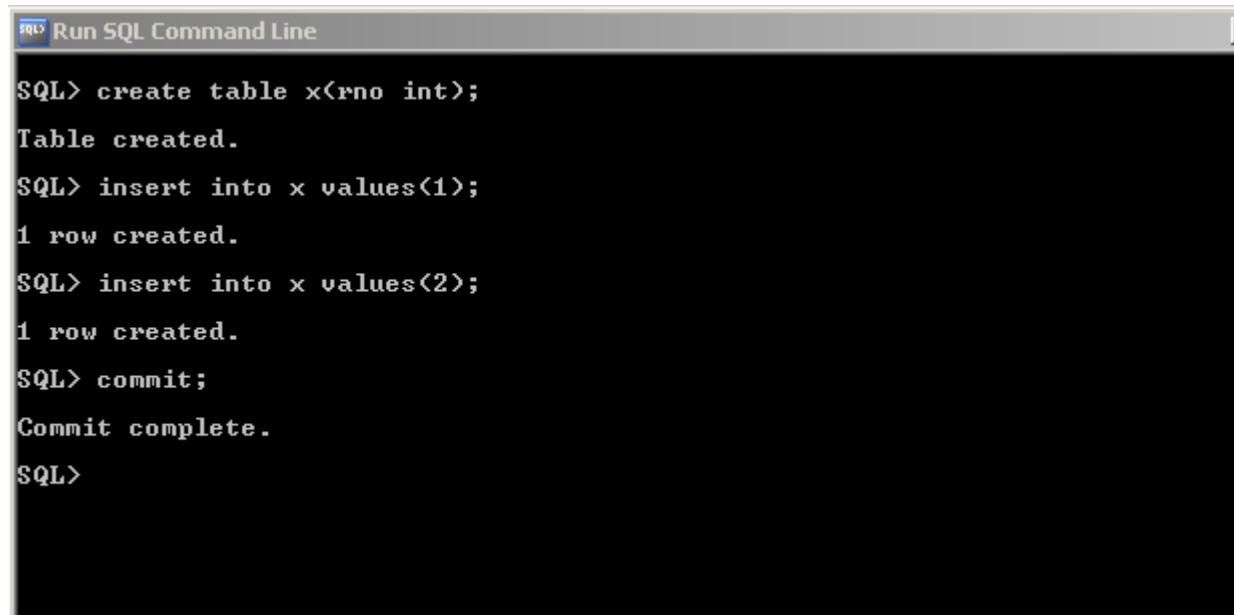
**ROLLBACK:** Rollback is a type of redo command. It is similarly like undo command.

For Example:

We have created a table x.

We have inserted 2 rows and run the commit command.

The changes will be saved.



```
Run SQL Command Line
SQL> create table x(rno int);
Table created.
SQL> insert into x values(1);
1 row created.
SQL> insert into x values(2);
1 row created.
SQL> commit;
Commit complete.
SQL>
```

By mistakenly if we run the wrong command, we can use rollback.

```
SQL> delete from x where rno = 2;
1 row deleted.

SQL> select * from x;
    RNO
-----
      1

SQL> rollback;
Rollback complete.

SQL> select * from x;
    RNO
-----
      1
      2

SQL>
```

## **IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 10 DBMS**

```
SQL> select * from x;
      RNO
-----
      1
      2

SQL> update x
  2  set rno = 10;
2 rows updated.

SQL> select * from x;
      RNO
-----
     10
     10

SQL> rollback;
Rollback complete.

SQL> select * from x;
      RNO
-----
      1
      2

SQL>
```

```
SQL> select * from x;
      RNO
-----
      1
      2

SQL> update x
  2  set rno = 10;
2 rows updated.

SQL> select * from x;
      RNO
-----
     10
     10

SQL> commit;
Commit complete.

SQL> select * from x;
      RNO
-----
     10
     10

SQL>
```

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

- a) We can lock table many times in share mode.

```
Run SQL Command Line
SQL> lock table emp2 in share mode;
Table(s) Locked.
SQL>
```

```
Run SQL Command Line
SQL> lock table emp2 in share mode;
Table(s) Locked.
SQL>
```

```
Run SQL Command Line
SQL> lock table emp2 in share mode;
Table(s) Locked.
SQL>
```

- b) Once we lock the table in share mode in one instance, if we want to lock the same table in exclusive mode in another instance then it will wait for the first instance to be closed.

```
Run SQL Command Line

SQL> lock table emp2 in share mode;

Table(s) Locked.

SQL>
```

```
Run SQL Command Line

SQL> lock table emp2 in exclusive mode;
```

Once the first instance will be closed it will locked in exclusive mode.

```
Run SQL Command Line

SQL> lock table emp2 in exclusive mode;

Table(s) Locked.

SQL>
```

c) We can't make the same table in exclusive mode again in another instances. It will wait for the first instance to be closed.

But we can make the same table in share mode in another instances. As we saw in example (a).

Run SQL Command Line

```
SQL> lock table emp2 in exclusive mode;
Table(s) Locked.
```

Run SQL Command Line

```
SQL> lock table emp2 in exclusive mode;
```

Once the first instance will be deleted, it will locked the table.

Run SQL Command Line

```
SQL> lock table emp2 in exclusive mode;
Table(s) Locked.
```

d) When we want to lock the table in share mode which is already in exclusive mode, so it will wait for first instances to be removed or deleted.

 Run SQL Command Line

```
SQL> lock table emp2 in exclusive mode;
```

```
Table(s) Locked.
```

```
SQL>
```

 Run SQL Command Line

```
SQL> lock table emp2 in share mode;
```

Once the first instance will be deleted, it will locked the table.

 Run SQL Command Line

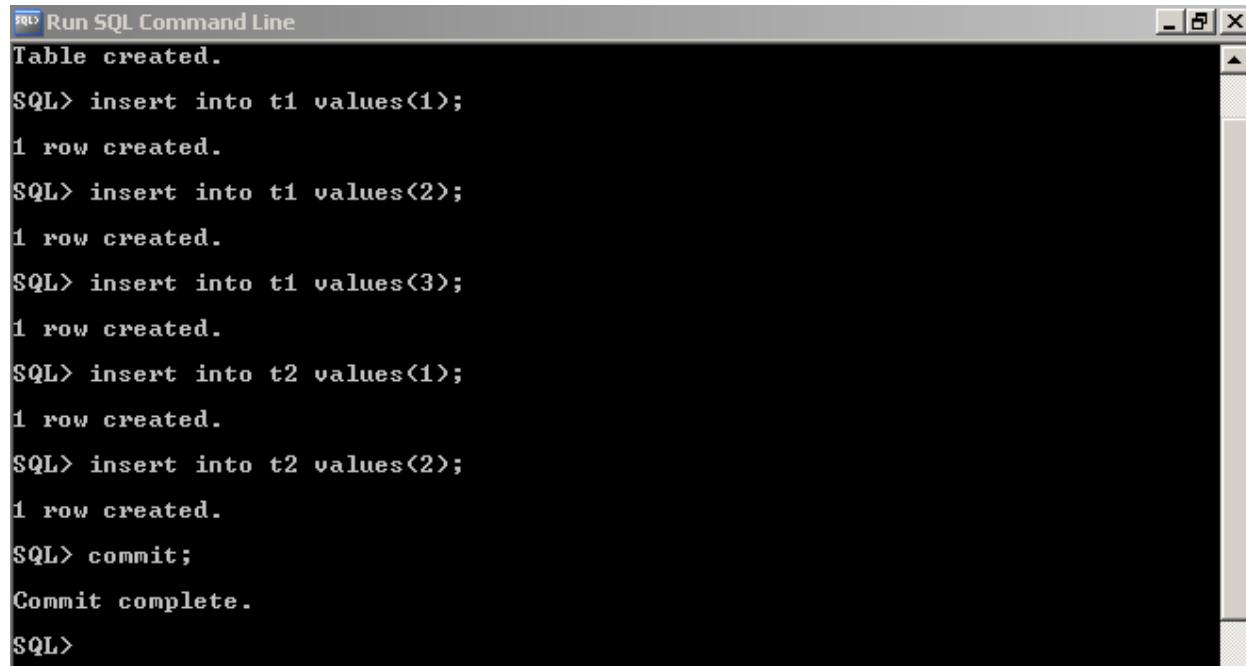
```
SQL> lock table emp2 in share mode;
```

```
Table(s) Locked.
```

```
SQL>
```

Implementing Deadlock.

Create two tables. Do commit.



```
Run SQL Command Line
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(1);
1 row created.

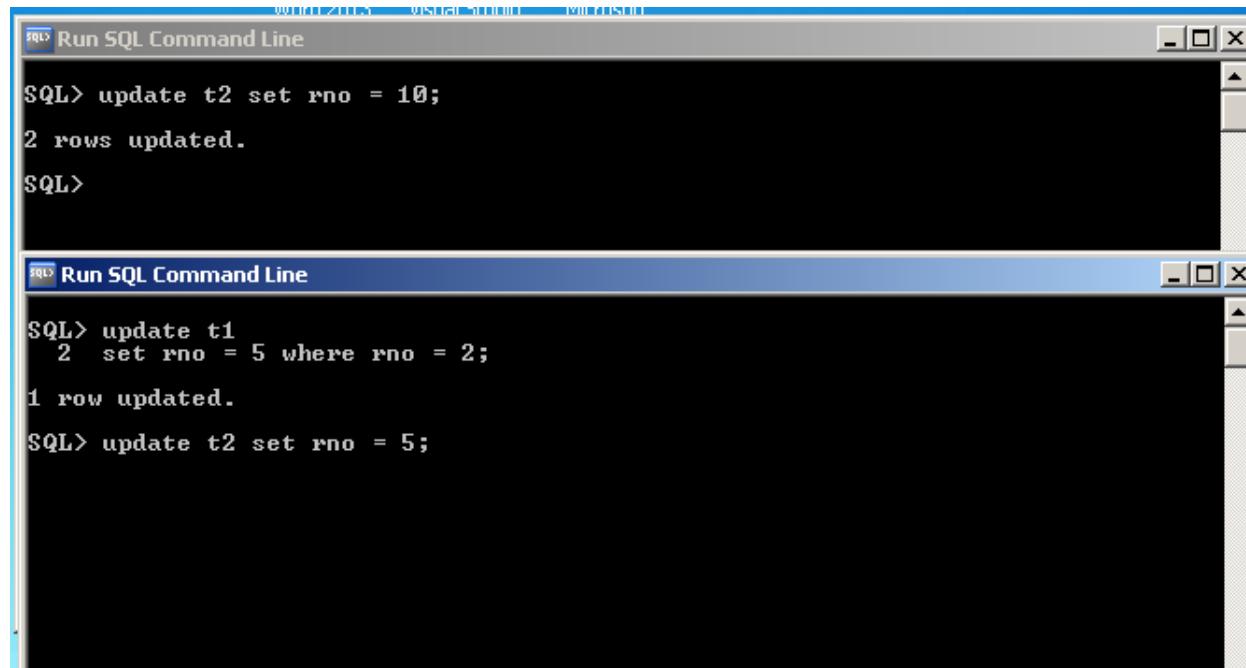
SQL> insert into t2 values(2);
1 row created.

SQL> commit;
Commit complete.

SQL>
```

One instance is holding table t1 and another instance is holding table t2.

When we try to update table t2 from instance 1 it will wait for another instance.



```
Run SQL Command Line
SQL> update t2 set rno = 10;
2 rows updated.

SQL>

Run SQL Command Line
SQL> update t1
2   set rno = 5 where rno = 2;
1 row updated.

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

# IMRAN RIZWAN SHAIKH 109 FYIT PRACTICAL 11 DBMS

If we try to update table 1 from instance 2 it will raise error in instance 1.

The screenshot shows two separate Oracle SQL Command Line windows. The top window, titled 'Run SQL Command Line', is connected to Instance 1. It contains the following session history:

```
SQL> update t1
  2  set rno = 5 where rno = 2;
1 row updated.

SQL> update t2 set rno = 5;
update t2 set rno = 5
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
```

The bottom window, also titled 'Run SQL Command Line', is connected to Instance 2. It contains the following session history:

```
SQL> update t2 set rno = 10;
2 rows updated.

SQL> update t1 set rno = 19;
```