INDEX

Task- 1:

DDL commands (Create, Alter, Drop, Truncate)

1. Create a table EMP with the following structure.

Name	Type
EMPNO	NUMBER(6)
ENAME	VARCHAR2(20)
JOB	VARCHAR2(10)
MGR	NUMBER(4)
DEPTNO	NUMBER(3)
SAL	NUMBER(7,2)

2. Add a column commission to the emp table. Commission should be numeric with null values allowed.

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

4. Create dept table with the following structure.

NameType	
DEPTNO	NUMBER(2)
DNAME	VARCHAR2(10)
LOC	VARCHAR2(10)

DEPTNO as the primary key

5. Add constraints to the emp table that is empno as the primary key and deptno as the foreign key.

6. Add constraints to the emp table to check the empno value while entering (i.e) empno> 100. Salary value by default is 5000, otherwise it should accept the values from the user.
7. Add columns DOB to the emp table. Add and drop a column DOJ to the emp table.

Task- 2: DML COMMANDS (Insert, Update, Delete)

1. Insert 5 records into dept Insert few rows and truncate those from the emp1 table and also drop it.

2. Insert 11 records into emp table.

3. Update the emp table to set the value of commission of all employees to Rs1000/- who are working as managers.

4. Delete only those who are working as supervisors.
5. Delete the rows whose empno is 7599.

Task-3:TCL COMMANDS (Save Point, Rollback Commit).

Task- 4: DQL COMMAND (Select)- SQL Operators and Order by Clause

1.List the records in the emp table order by salary in descending order.

2.Display only those employees whose deptno is 30.
3.Display deptno from the table employee avoiding the duplicated values.
4.List all employee names, salary and 15% rise in salary. Label the column as pay hike.
5.Display the rows whose salary ranges from 15000 to 30000.
6.Display all the employees in dept 10 and 20 in alphabetical order of names.
7.List the employee names who do not earn commission.
8.Display all the details of the records with 5 character names with 'S' as starting character.
9Display joining date of all employees in the year of 1998.
10.List out the employee names whose salary is greater than 5000 and less than 6000

Task- 5: SQL Aggregate Functions, Group By clause, Having clause

1. Count the total records in the emp table.
2. Calculate the total and average salary of the employee.
3. Determine the max and min salary and rename the column as max-salary and min_salary.
4. Find number of departments in employee table.
5. Display job wise sum, average, max, min salaries.
6. Display maximum salaries of all the departments having maximum salary > 2000

7. Display job wise sum, avg, max, min salaries in department 10 having average salary is greater than 1000 and the result is ordered by sum of salary in descending order.

Task- 6: SQL Functions

1. Display the employee name concatenate with employee number.
2. Display half of employee name in upper case and half in lower case.
3. Display the month name of date "14-jul-09" in full.
4. Display the Date of joining of all employees in the format "dd-mm-yy".
5. Display the date two months after the Date of joining of employees.
6. Display the last date of that month in "05-Oct-09".
7. Display the rounded date in the year format, month format, day format in the employee

8. Display the commissions earned by employees. If they do not earn commission, display it as "No Commission".

Task- 7: Nested Queries

1. Find the third highest salary of an employee.

2. Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with 'M'.

4. Write a query to display information about employees who earn more than any employee in dept 30.

5. Display the employees who have the same job as Jones and whose salary is greater than or equal to the salary of Ford.

6. List out the employee names who get the salary greater than the maximum salaries of dept with dept no 20, 30.

7. Display the maximum salaries of the departments whose maximum salary is greater than 9000.

8. Create a table employee with the same structure as the table emp and insert rows into the table using select clauses.

9. Create a manager table from the emp table which should hold details only about the managers.

Task- 8:

Joins, Set Operators.

1. Display all the employees and the departments implementing a left outer join.

2. Display the employee name and department name in which they are working implementing a full outer join.

3. Write a query to display their employee names and their managers' name and salary for every employee.

4. Write a query to output the name, job, empno, deptname and location for each dept, even if there are no employees.

5. Display the details of those who draw the same salary.

Task- 9:Views

1. Create a view that displays the employee id, name and salary of employees who belong to 10th department.

2. Create a view with read only option that displays the employee name and their department name.
3. Display all the views generated.
4. Execute the DML commands on views created and drop them.

Task- 10: Practice on DCL commands,sequence and indexes.

Task- 11:

1. Write a PL/SQL code to retrieve the employee name, join date and designation of an employee whose number is given as input by the user.

2. Write a PL/SQL code to calculate tax of employee.
3. Write a PL/SQL program to display top ten employee details based on salary using cursors.

4. Write a PL/SQL program to update the commission values for all the employees' with salary less than 2000, by adding 1000 to the existing values.

Task- 12:

1. Write a trigger on employee table that shows the old and new values of employee name after updating on employee name.

2. Write a PL/SQL procedure for inserting, deleting and updating the employee table.

3. Write a PL/SQL function that accepts the department number and returns the total salary of that department.

Task- 13:

1. Write PL/SQL program to handle predefined exceptions.
2. Write PL/SQL program to handle user defined exception.

3. Write a PL/SQL code to create
 a. Package specification

 b. Package body to insert ,update, delete and
retrieve data on emp table.

Task-14: Table locking (Shared Lock and Exclusive lock)

TASK – 1

DDL COMMANDS (Create, Alter, Drop, Truncate)

1. Create a table EMP with the following structure.

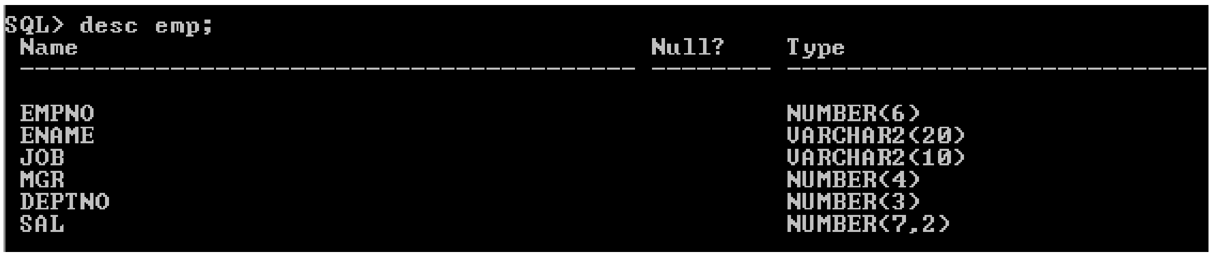
Name	Type
EMPNO	NUMBER(6)
ENAME	VARCHAR2(20)
JOB	VARCHAR2(10)
MGR	NUMBER(4)
DEPTNO	NUMBER(3)
SAL	NUMBER(7,2)

Query:

SQL>create table emp(empno number(6), ename varchar2(20), jobvarchar2(10), mgr number(4), deptno number(3), sal number(7,2));

Table created.

Output:



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

 EMPNO                                              NUMBER(6)
 ENAME                                              VARCHAR2(20)
 JOB                                                VARCHAR2(10)
 MGR                                                NUMBER(4)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
```

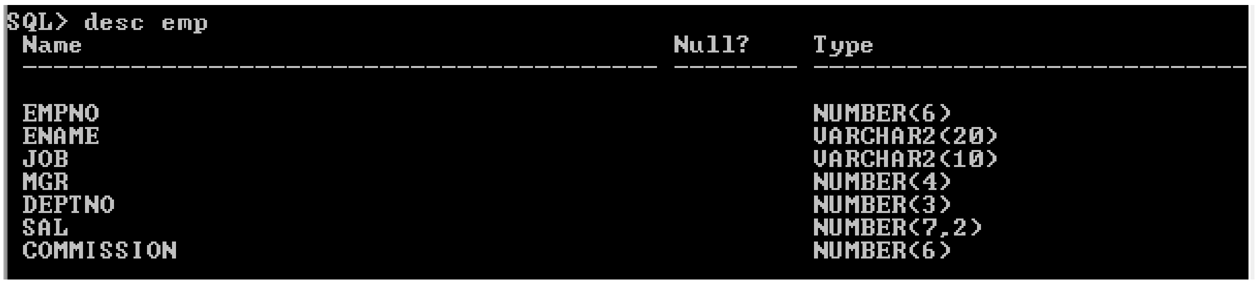
2. Add a column commission to the EMP table. Commission should be numeric with null values allowed.

Query:

SQL>Alter table empadd(commission number(4));

Output:

Table altered.



```
SQL> desc emp
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 EMPNO                                              NUMBER(6)
 ENAME                                              VARCHAR2(20)
 JOB                                                VARCHAR2(10)
 MGR                                                NUMBER(4)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
 COMMISSION                                         NUMBER(6)
```

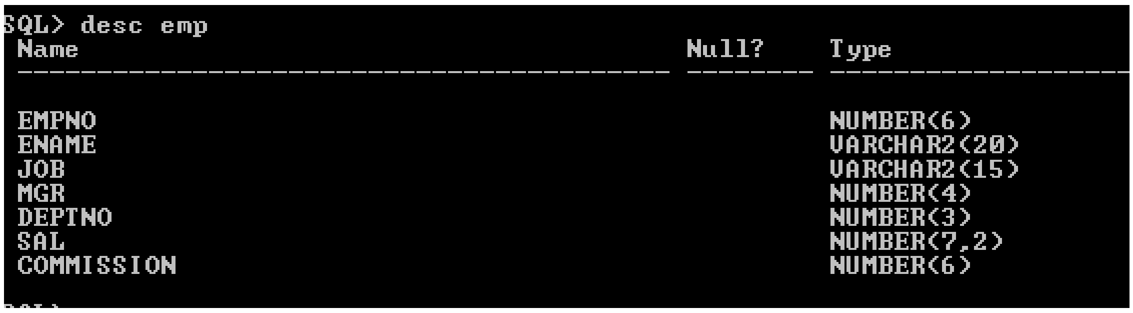
3. Modify the column width of the job field of emp table.

Query:

SQL>Alter table empmodify(job varchar2(15));

Output:

Table altered.



```
SQL> desc emp
 Name                                      Null?    Type
 ----------------------------------------- -------- --------------------

 EMPNO                                              NUMBER(6)
 ENAME                                              VARCHAR2(20)
 JOB                                                VARCHAR2(15)
 MGR                                                NUMBER(4)
 DEPTNO                                             NUMBER(3)
 SAL                                                NUMBER(7,2)
 COMMISSION                                         NUMBER(6)
```

4. Create dept table with the following structure.

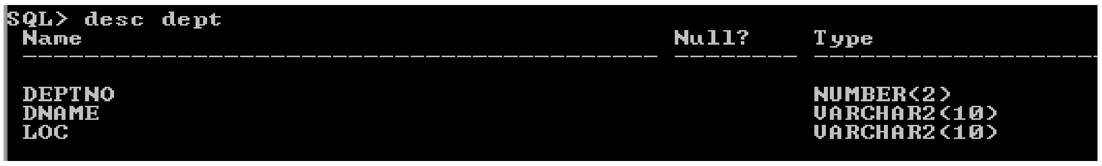
Name	Type
DEPTNO	NUMBER(2)
DNAME	VARCHAR2(10)
LOC	VARCHAR2(10)

Query:

SQL>create table dept(deptno number(2), dname varchar2(10), loc varchar2(10));

Output:

Table created.



```
SQL> desc dept
 Name                                      Null?    Type
 ----------------------------------------- -------- --------------------

 DEPTNO                                             NUMBER(2)
 DNAME                                              VARCHAR2(10)
 LOC                                                VARCHAR2(10)
```

5. Add constraint to the emp table that is empno as primary key and deptno as foreign key.

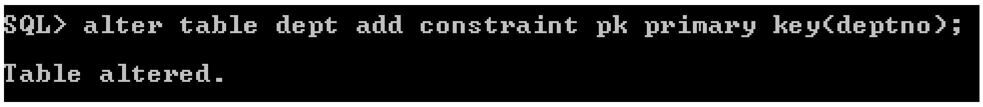
Query:

SQL>alter table emp add constraint emp_id_pk primary key(empno);

SQL>alter table dept add constraint pk primary key(deptno);

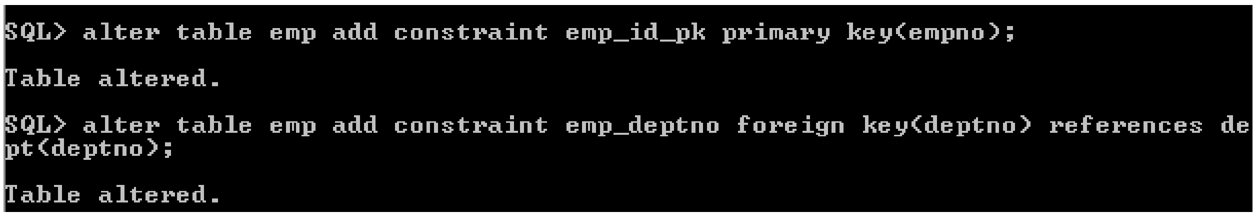
Output:

Table altered



```
SQL> alter table dept add constraint pk primary key(deptno);

Table altered.
```

SQL>Alter table emp add constraint emp_deptno_fk foreign key(deptno) references dept(deptno);



```
SQL> alter table emp add constraint emp_id_pk primary key(empno);

Table altered.

SQL> alter table emp add constraint emp_deptno foreign key(deptno) references de
pt(deptno);

Table altered.
```

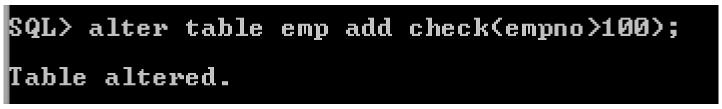
6. Add constraints to the emp table to check the empno value while entering i.eempno>100. Salary value by default is 5000, otherwise it should accept the values from the user.

Query:

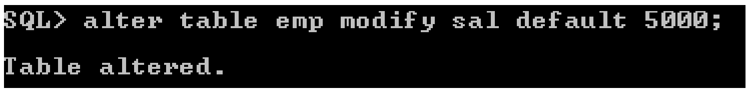
SQL>alter table emp add check (empno>100);

SQL>alter table emp modify sal default 5000;

Output:



```
SQL> alter table emp add check(empno>100);

Table altered.
```



```
SQL> alter table emp modify sal default 5000;

Table altered.
```

7. Add column DOB to the emp table Add and drop a column DOJ to the emp table.

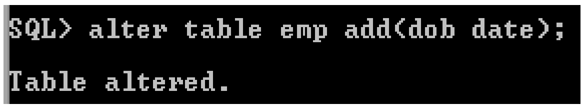
Query:

SQL>alter table emp add(dob date);

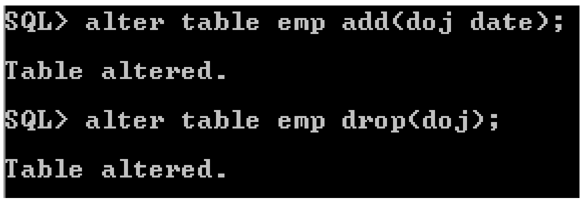
SQL>alter table emp add(doj date);

SQL>alter table emp drop(doj);

Output:



```
SQL> alter table emp add(dob date);

Table altered.
```



```
SQL> alter table emp add(doj date);
Table altered.
SQL> alter table emp drop(doj);
Table altered.
```

TASK – 2

DML COMMANDS (Insert, Select, Update, Delete)

1.Insert 5 records into dept table.Insert few rows and truncate those from emp1 table and also drop it. .

Query:

SQL>Insert into dept values(&deptno,’&dname’,’&loc’);

SQL>create table emp1 as select * from emp;

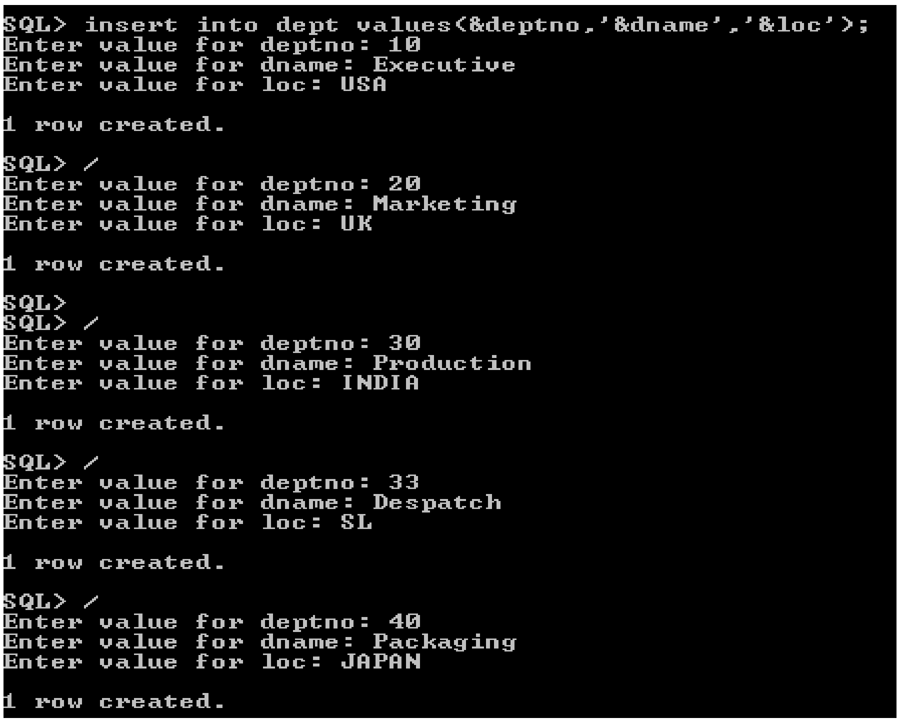
SQL>insert into emp1 values(7000, ’King’, ‘Pres’, 10, 20,10000,500, ’12-Jan-92’);

SQL>insert into emp1 values(7010, ‘Jack’, ‘VP’, 10, 30, 9000, 300, ’19-Jul-92’);

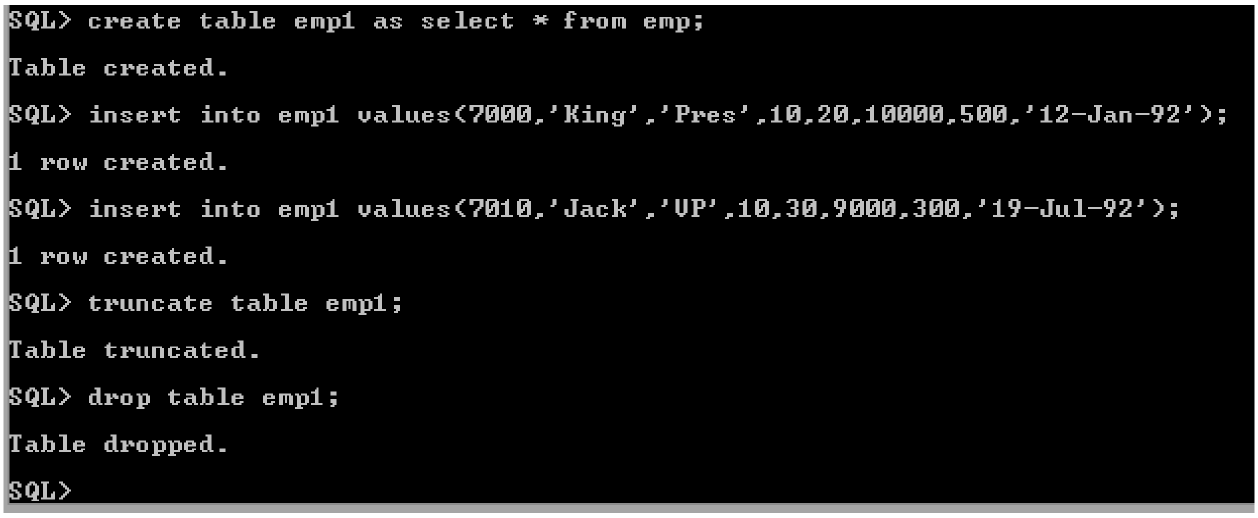
SQL>Truncate table emp1;

SQL>Drop table emp1;

Output:



```
SQL> insert into dept values(&deptno,'&dname','&loc');
Enter value for deptno: 10
Enter value for dname: Executive
Enter value for loc: USA

1 row created.

SQL> /
Enter value for deptno: 20
Enter value for dname: Marketing
Enter value for loc: UK

1 row created.

SQL>
SQL> /
Enter value for deptno: 30
Enter value for dname: Production
Enter value for loc: INDIA

1 row created.

SQL> /
Enter value for deptno: 33
Enter value for dname: Despatch
Enter value for loc: SL

1 row created.

SQL> /
Enter value for deptno: 40
Enter value for dname: Packaging
Enter value for loc: JAPAN

1 row created.
```



```
SQL> create table emp1 as select * from emp;

Table created.

SQL> insert into emp1 values(7000,'King','Pres',10,20,10000,500,'12-Jan-92');

1 row created.

SQL> insert into emp1 values(7010,'Jack','VP',10,30,9000,300,'19-Jul-92');

1 row created.

SQL> truncate table emp1;

Table truncated.

SQL> drop table emp1;

Table dropped.

SQL>
```

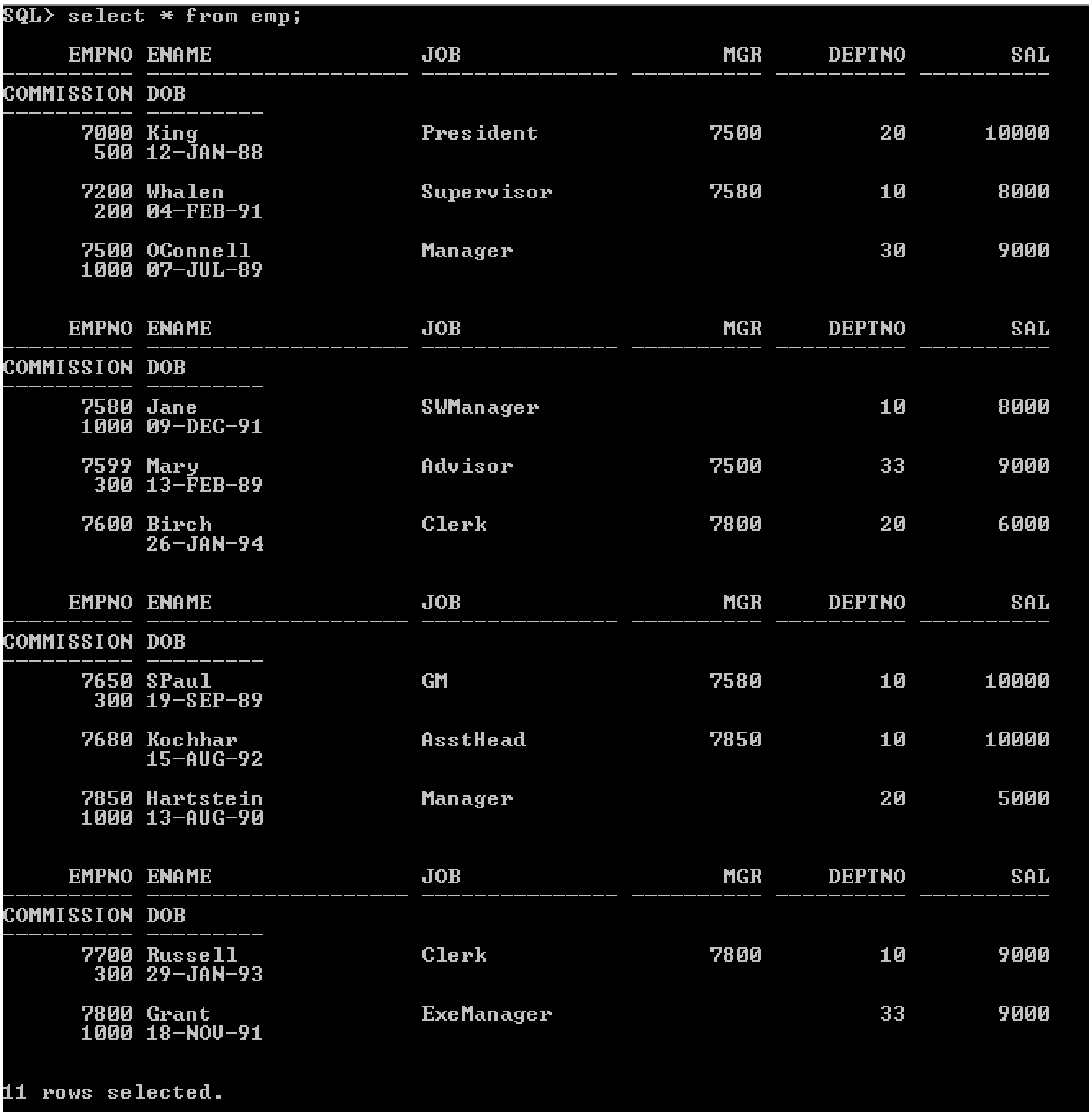
2. Insert 11 records into the emp table.

Query:

SQL>insert into empvalues(&no, ’&name’, ’&job’, &mgr, &deptno, &sal, &comm, ‘&dob’);

Note: Repeat execution of this statement for 11 times for 11 record insertions

Output:



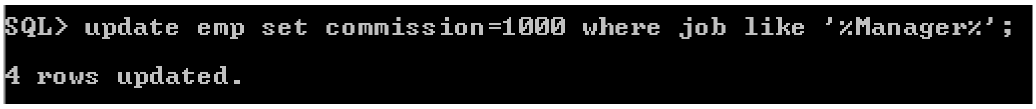
```
SQL> select * from emp;

     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7000 King                 President             7500         20      10000
       500 12-JAN-88

      7200 Whalen               Supervisor            7580         10       8000
       200 04-FEB-91

      7500 OConnell             Manager                            30       9000
      1000 07-JUL-89


     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7580 Jane                 SWManager                          10       8000
      1000 09-DEC-91

      7599 Mary                 Advisor               7500         33       9000
       300 13-FEB-89

      7600 Birch                Clerk                 7800         20       6000
           26-JAN-94


     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7650 SPaul                GM                    7580         10      10000
       300 19-SEP-89

      7680 Kochhar              AsstHead              7850         10      10000
           15-AUG-92

      7850 Hartstein            Manager                            20       5000
      1000 13-AUG-90


     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7700 Russell              Clerk                 7800         10       9000
       300 29-JAN-93

      7800 Grant                ExeManager                         33       9000
      1000 18-NOV-91


11 rows selected.
```

3. Update the emp table to set the default commission of all employees to Rs.1000 /- who are working as managers.

Query:

SQL>update emp set commission=1000 where job like '%Manager%';

Output:



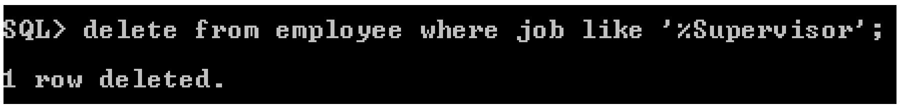
```
SQL> update emp set commission=1000 where job like '%Manager%';

4 rows updated.
```

4. Delete only those who are working as Supervisors.

Query:

SQL>delete from employee where job like '%Supervisor';

Output:



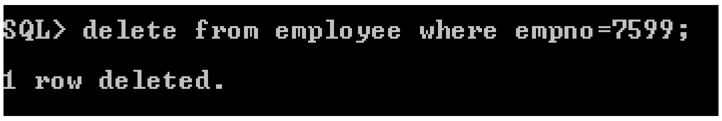
```
SQL> delete from employee where job like '%Supervisor';

1 row deleted.
```

5. Delete the rows whose empno is 759**9.**

Query:

SQL>delete from employee where empno=7599;

Output:



```
SQL> delete from employee where empno=7599;

1 row deleted.
```

TASK-3

TCL COMMANDS (Save Point, Rollback Commit).

Commit command is used to permanently save any transaction into the database.

When we use any DML command like Insert, Update and Delete the changes made by these commands are not permanent, until the current session is closed, the changes made by these commands can be rolled back.

To avoid that, we use Commit command to mark the changes as permanent.

Following is commit command's syntax,

Commit;

Roll back

This command restores the database to last committed state. It is also used with savepoint command to jump to a savepoint in an ongoing transaction.

If we have used the Update command to make some changes into the database, and realise that those changes were not required, then we can use the Rollback command to rollback those changes, if they were not commited using the Commit command.

Following is rollback command's syntax,

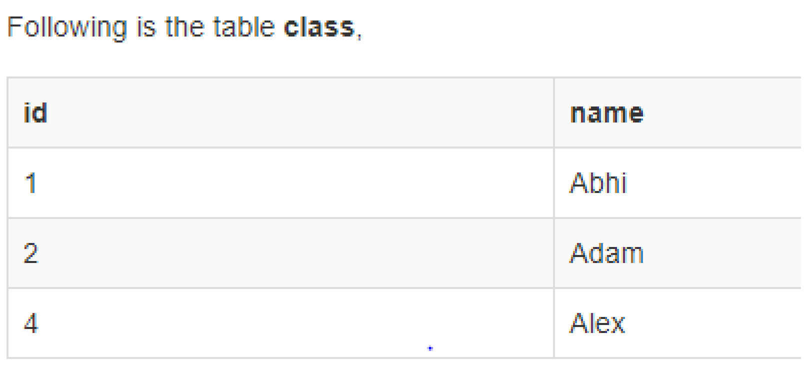
```
ROLLBACK TO savepoint_name;
```

Save Point

This command is used to temporarily save a transaction so that you can rollback to that point whenever required.

Syntax:

```
SAVEPOINT savepoint_name;
```



Following is the table **class**,

id	name
1	Abhi
2	Adam
4	Alex

Lets use some SQL queries on the above table and see the results.

```
INSERT INTO class VALUES(5, 'Rahul');

COMMIT;

UPDATE class SET name = 'Abhijit' WHERE id = '5';

SAVEPOINT A;

INSERT INTO class VALUES(6, 'Chris');

SAVEPOINT B;

INSERT INTO class VALUES(7, 'Bravo');

SAVEPOINT C;

SELECT * FROM class;
```

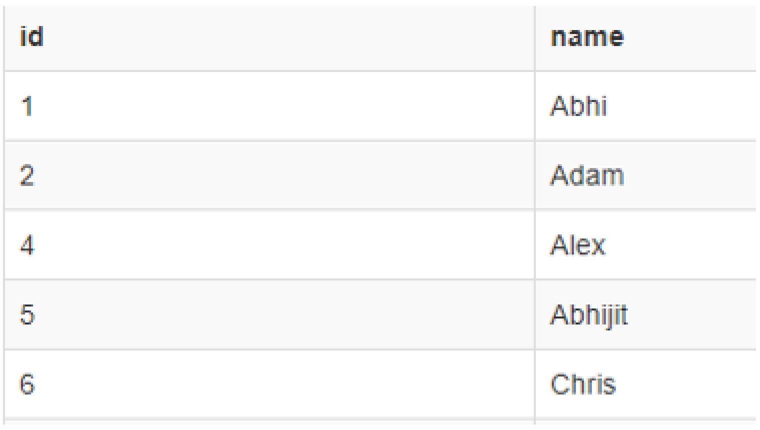
> **NOTE:** `SELECT` statement is used to show the data stored in the table.

The resultant table will look like,

id	name
1	Abhi
2	Adam
4	Alex
5	Abhijit
6	Chris
7	Bravo

Now let's use the `ROLLBACK` command to roll back the state of data to the **savepoint B**.

```
ROLLBACK TO B;

SELECT * FROM class;
```



id	name
1	Abhi
2	Adam
4	Alex
5	Abhijit
6	Chris

TASK – 4

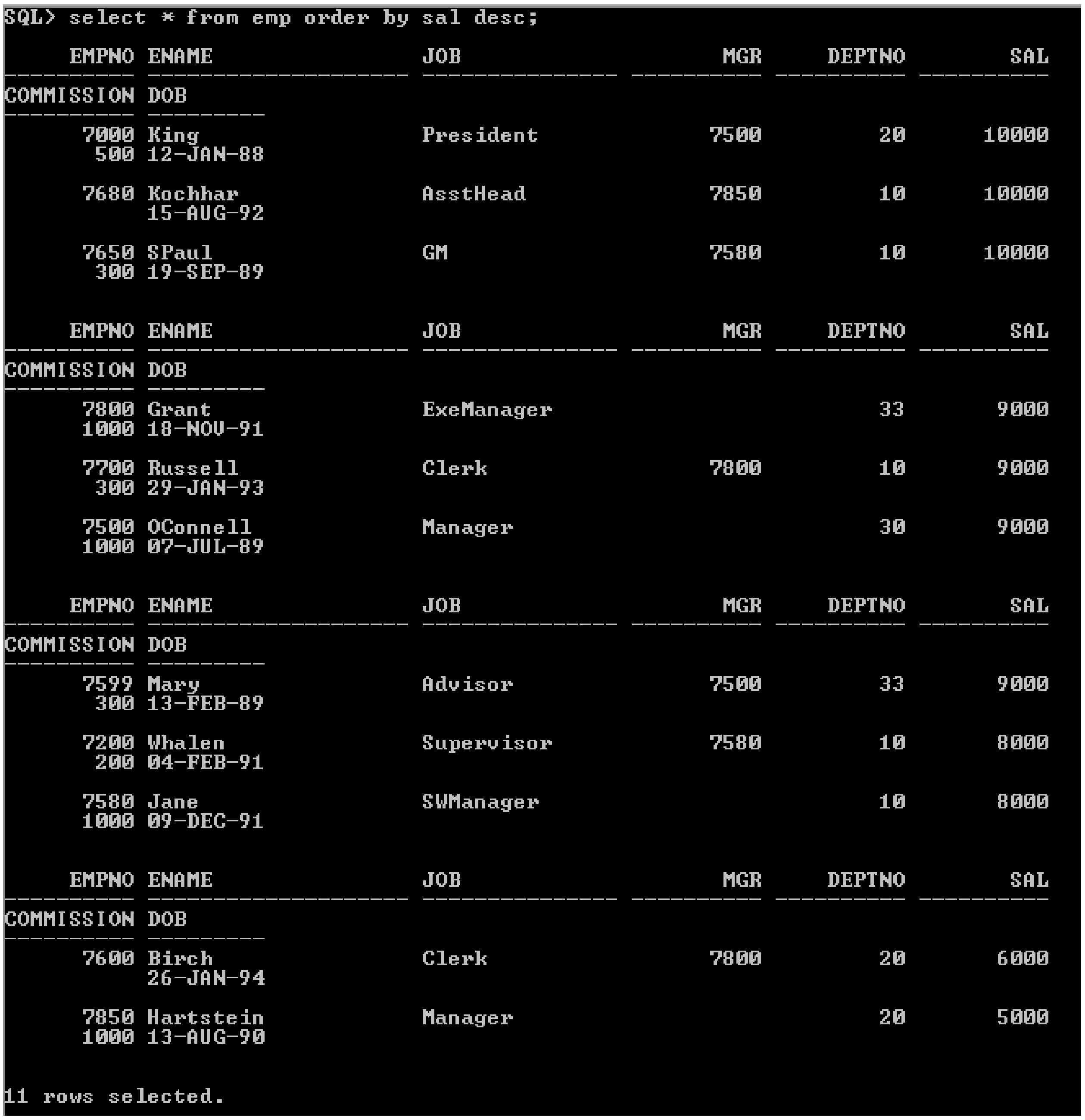
SQL Operators

1. List the records in the emp table order by salary in descending order.

Query:

SQL>select * from emp order by saldesc;

Output:



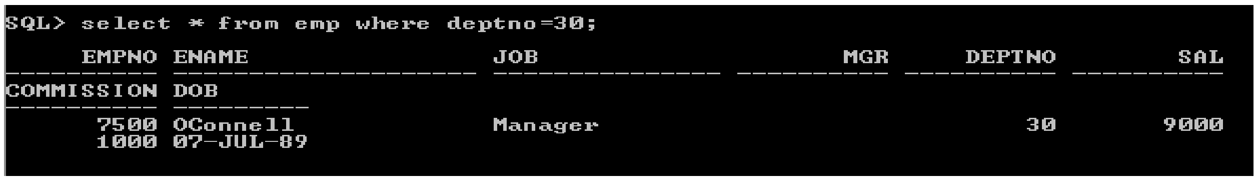
```
SQL> select * from emp order by sal desc;

     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7000 King                 President             7500         20      10000
       500 12-JAN-88

      7680 Kochhar              AsstHead              7850         10      10000
           15-AUG-92

      7650 SPaul                GM                    7580         10      10000
       300 19-SEP-89


     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7800 Grant                ExeManager                         33       9000
      1000 18-NOV-91

      7700 Russell              Clerk                 7800         10       9000
       300 29-JAN-93

      7500 OConnell             Manager                            30       9000
      1000 07-JUL-89


     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7599 Mary                 Advisor               7500         33       9000
       300 13-FEB-89

      7200 Whalen               Supervisor            7580         10       8000
       200 04-FEB-91

      7580 Jane                 SWManager                          10       8000
      1000 09-DEC-91


     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7600 Birch                Clerk                 7800         20       6000
           26-JAN-94

      7850 Hartstein            Manager                            20       5000
      1000 13-AUG-90


11 rows selected.
```

2. Display only those employees whose deptno is 30.

Query:

SQL>select * from emp where deptno=30;

Output:



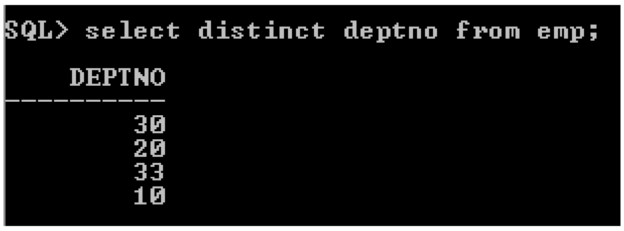
```
SQL> select * from emp where deptno=30;
     EMPNO ENAME                JOB                    MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7500 OConnell             Manager                            30       9000
      1000 07-JUL-89
```

3. Display deptno from the table employee avoiding the duplicate values.

Query:

SQL>select distinct deptno from emp;

Output:



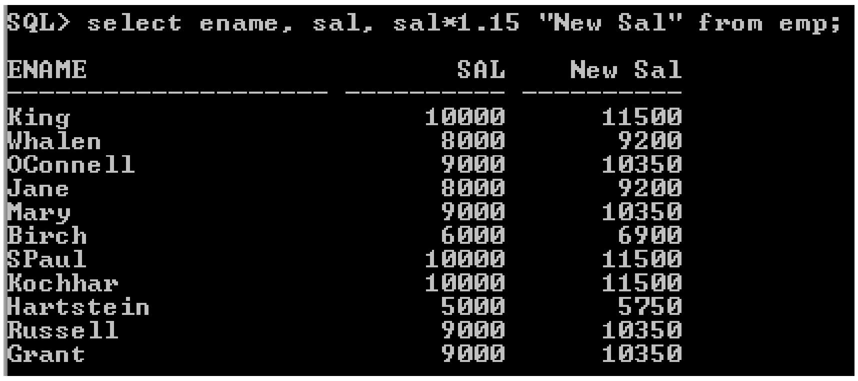
```
SQL> select distinct deptno from emp;

    DEPTNO
----------
        30
        20
        33
        10
```

4. List all employee names, salary and 15% rise in salary.Label the column as New Sal.

Query:

SQL>select ename, sal, (sal***1.**15) “New Sal” from emp;

Output:



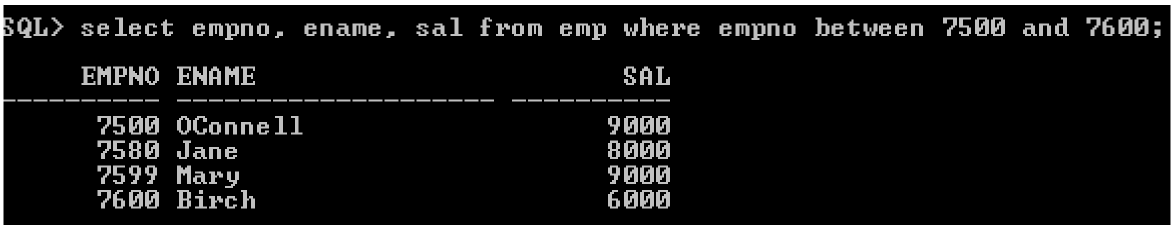
```
SQL> select ename, sal, sal*1.15 "New Sal" from emp;

ENAME                       SAL    New Sal
-------------------- ---------- ----------
King                      10000      11500
Whalen                     8000       9200
OConnell                   9000      10350
Jane                       8000       9200
Mary                       9000      10350
Birch                      6000       6900
SPaul                     10000      11500
Kochhar                   10000      11500
Hartstein                  5000       5750
Russell                    9000      10350
Grant                      9000      10350
```

5. Display the rows whose empno ranges from 7500 to 7600.

Query:

SQL>select empno, ename, sal from emp where empno between 7500 and 7600;

Output:

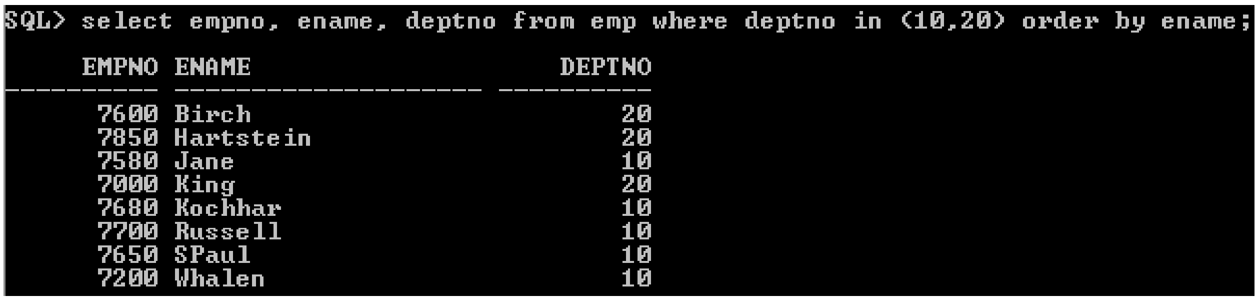

```
SQL> select empno, ename, sal from emp where empno between 7500 and 7600;

     EMPNO ENAME                       SAL
---------- -------------------- ----------
      7500 OConnell                   9000
      7580 Jane                       8000
      7599 Mary                       9000
      7600 Birch                      6000
```

6. Display all the employees in dept 10 and 20 in alphabetical order of names.

Query:

SQL>select empno, ename, deptno from emp where deptnoin(10,20) order by ename;

Output:

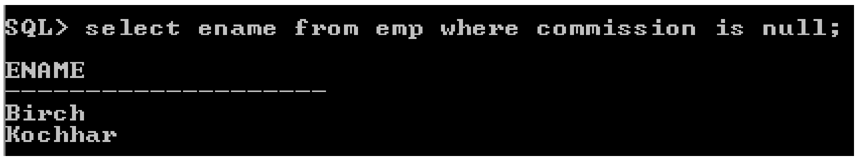

```
SQL> select empno, ename, deptno from emp where deptno in (10,20) order by ename;

     EMPNO ENAME                    DEPTNO
---------- -------------------- ----------
      7600 Birch                        20
      7850 Hartstein                    20
      7580 Jane                         10
      7000 King                         20
      7680 Kochhar                      10
      7700 Russell                      10
      7650 SPaul                        10
      7200 Whalen                       10
```

7. List the employe names who do not earn commission.

Query:

SQL>select empno, ename, sal from emp where commission is null;

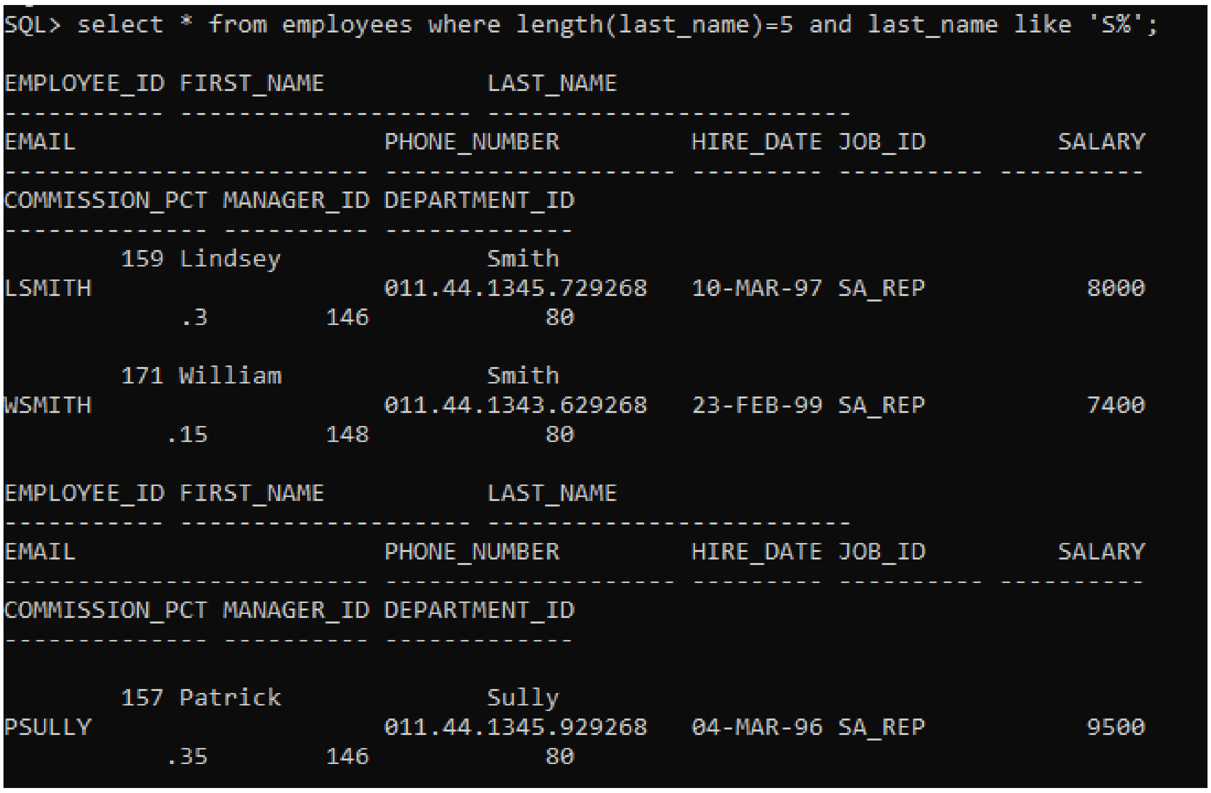
Output:


```
SQL> select ename from emp where commission is null;

ENAME
--------------------
Birch
Kochhar
```

8. Display all the details of the records with 5 character names with ‘S’ as starting character**.**

Query:

SQL>select * from employees where lengty(last_name)=5 and last_name like ‘s%’;

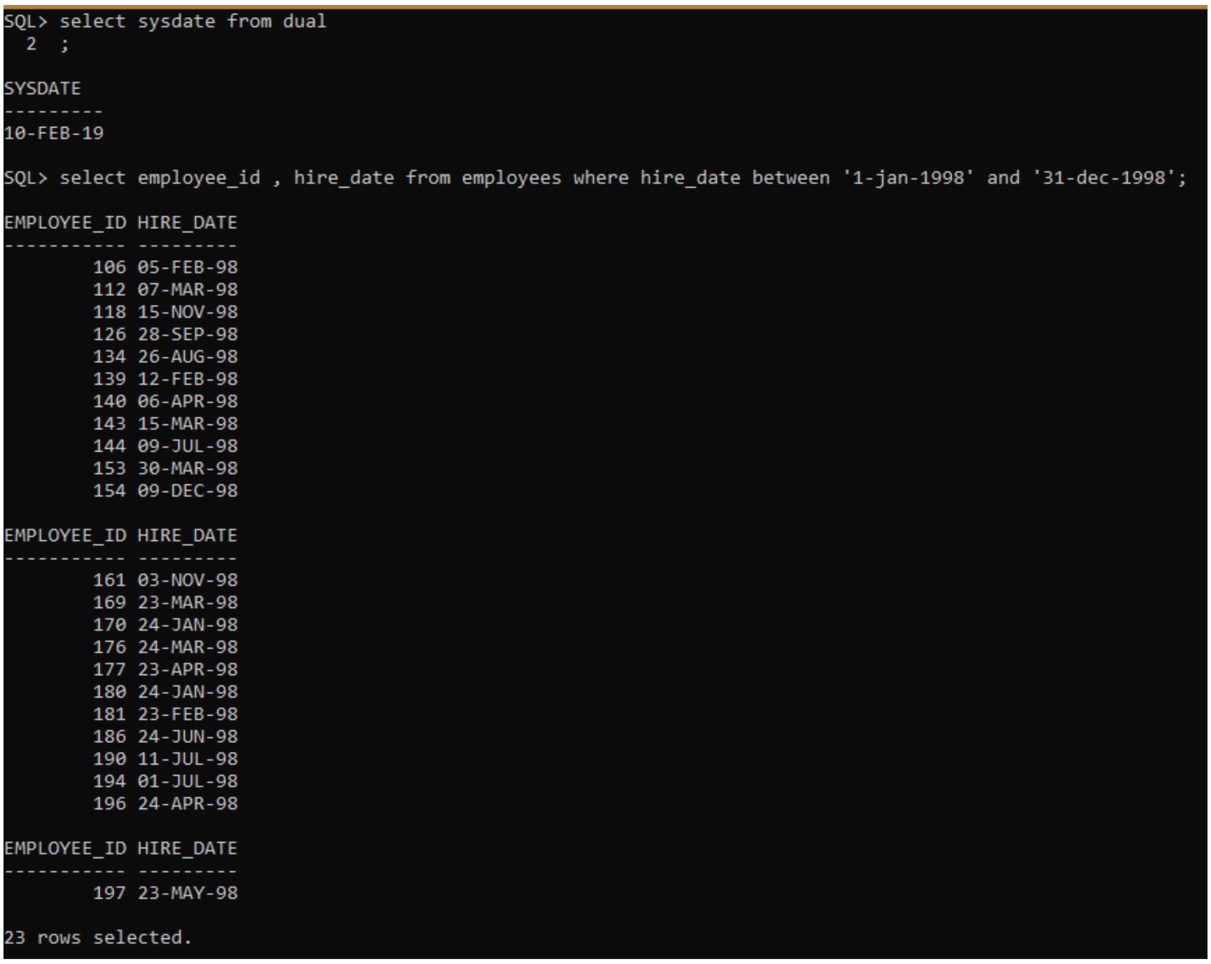


```
SQL> select * from employees where length(last_name)=5 and last_name like 'S%';

EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
EMAIL                     PHONE_NUMBER         HIRE_DATE JOB_ID         SALARY
------------------------- -------------------- --------- ---------- ----------
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-------------- ---------- -------------
        159 Lindsey              Smith
LSMITH                    011.44.1345.729268   10-MAR-97 SA_REP           8000
            .3        146            80

        171 William              Smith
WSMITH                    011.44.1343.629268   23-FEB-99 SA_REP           7400
           .15        148            80

EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
EMAIL                     PHONE_NUMBER         HIRE_DATE JOB_ID         SALARY
------------------------- -------------------- --------- ---------- ----------
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-------------- ---------- -------------

        157 Patrick              Sully
PSULLY                    011.44.1345.929268   04-MAR-96 SA_REP           9500
           .35        146            80
```

9.Display joining date of all employees in the year of 1998.

Query:

SQL>select employee_id ,hire_date from employees where hire_date between ‘1-jan-1998’ and ’31-dec-1998’;



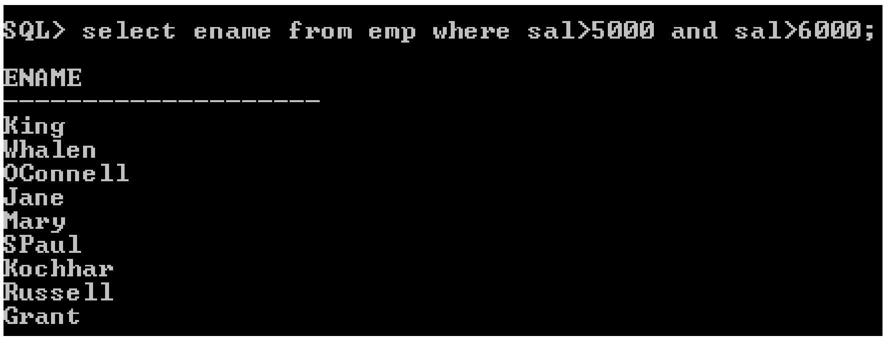
```
SQL> select sysdate from dual
  2  ;

SYSDATE
---------
10-FEB-19

SQL> select employee_id , hire_date from employees where hire_date between '1-jan-1998' and '31-dec-1998';

EMPLOYEE_ID HIRE_DATE
----------- ---------
        106 05-FEB-98
        112 07-MAR-98
        118 15-NOV-98
        126 28-SEP-98
        134 26-AUG-98
        139 12-FEB-98
        140 06-APR-98
        143 15-MAR-98
        144 09-JUL-98
        153 30-MAR-98
        154 09-DEC-98

EMPLOYEE_ID HIRE_DATE
----------- ---------
        161 03-NOV-98
        169 23-MAR-98
        170 24-JAN-98
        176 24-MAR-98
        177 23-APR-98
        180 24-JAN-98
        181 23-FEB-98
        186 24-JUN-98
        190 11-JUL-98
        194 01-JUL-98
        196 24-APR-98

EMPLOYEE_ID HIRE_DATE
----------- ---------
        197 23-MAY-98

23 rows selected.
```

10. List out the employee names whose salary is greater than 5000and lesser than6000.

Query:

SQL>select ename from emp where sal>5000 and sal>6000;

Output:



```
SQL> select ename from emp where sal>5000 and sal>6000;

ENAME
--------------------
King
Whalen
OConnell
Jane
Mary
SPaul
Kochhar
Russell
Grant
```

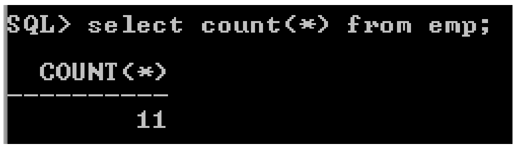
TASK – 5

SQL Aggregate Functions, Group By clause, Having clause

1. Count the total records in the emp table.

Query: select count(*) from emp;

Output:

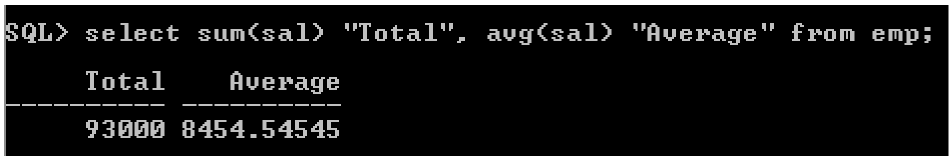


```
SQL> select count(*) from emp;

  COUNT(*)
----------
        11
```

2. Calculate the total and average salary of the employees.

Query: select sum(sal) “Total”, avg(sal) “Average” from emp;

Output:

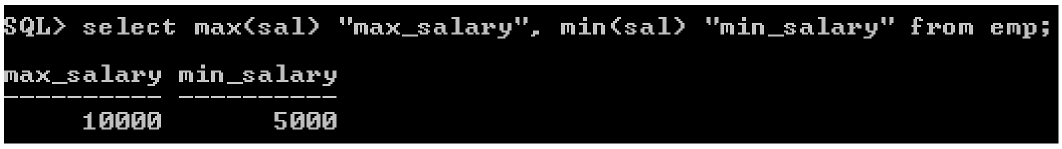


```
SQL> select sum(sal) "Total", avg(sal) "Average" from emp;

     Total    Average
---------- ----------
     93000 8454.54545
```

3. Determine the maximum and minimum salary of the employees and rename the columns max_salalary and min_salary.

Query: select max(sal) “max_salary”, min(sal) “min_salary” from emp;

Output:



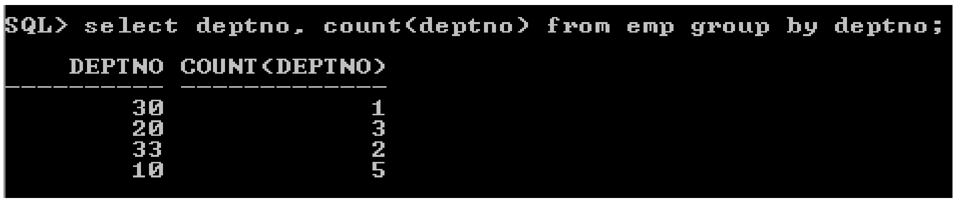
```
SQL> select max(sal) "max_salary", min(sal) "min_salary" from emp;

max_salary min_salary
---------- ----------
     10000       5000
```

4. Find the no.of departments in employee table.

Query: select deptno, count(deptno) from emp group by deptno;

Select count(Distinct deptno) from emp;

Output:

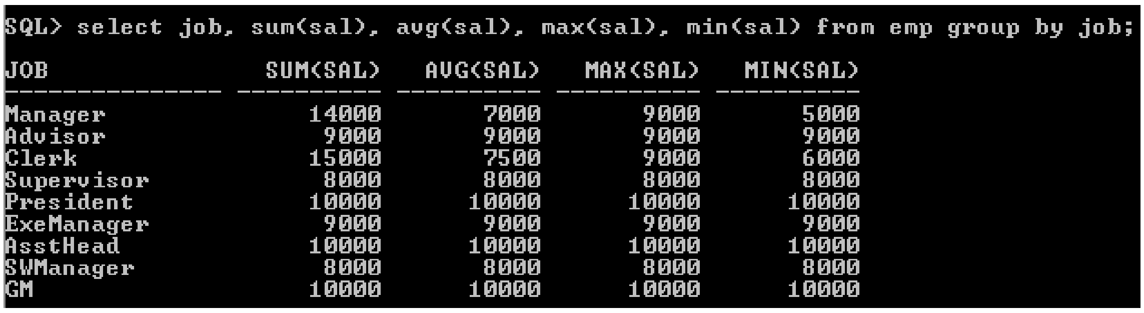


```
SQL> select deptno, count(deptno) from emp group by deptno;

    DEPTNO COUNT(DEPTNO)
---------- -------------
        30             1
        20             3
        33             2
        10             5
```

5. Display job wise sum, avg, max, min salaries.

Query: select job, sum(sal), avg(sal), max(sal), min(sal) from emp group by job;

Output:

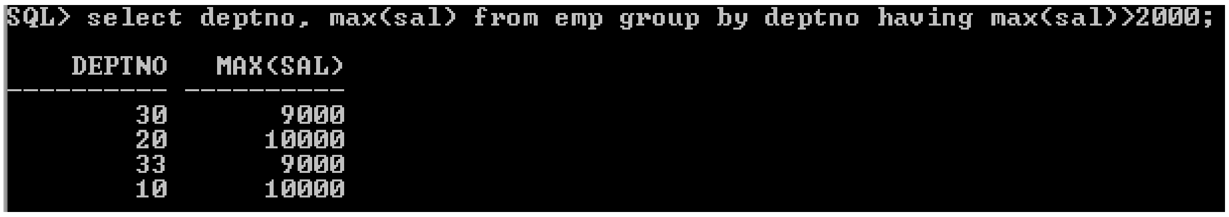


```
SQL> select job, sum(sal), avg(sal), max(sal), min(sal) from emp group by job;

JOB               SUM(SAL)   AVG(SAL)   MAX(SAL)   MIN(SAL)
--------------- ---------- ---------- ---------- ----------
Manager              14000       7000       9000       5000
Advisor               9000       9000       9000       9000
Clerk                15000       7500       9000       6000
Supervisor            8000       8000       8000       8000
President            10000      10000      10000      10000
ExeManager            9000       9000       9000       9000
AsstHead             10000      10000      10000      10000
SWManager             8000       8000       8000       8000
GM                   10000      10000      10000      10000
```

6. Display maximum salaries of all departments having maximum salary>2000.

Query: select deptno, max(sal) from emp group by deptno having max(sal)>2000;

Output:

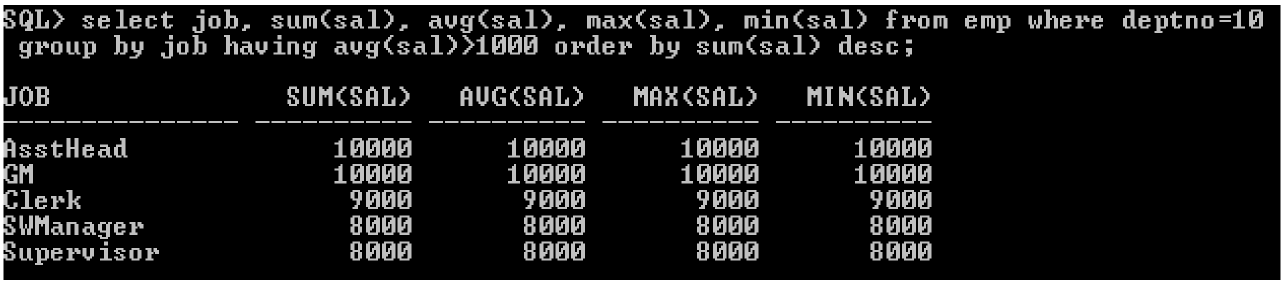


```
SQL> select deptno, max(sal) from emp group by deptno having max(sal)>2000;

    DEPTNO   MAX(SAL)
---------- ----------
        30       9000
        20      10000
        33       9000
        10      10000
```

7. Display job wise sum, avg, max and min salaries in department 10 having average salary>1000 and result is ordered by sum of salary in desc order.

Query: select job, sum(sal), avg(sal), max(sal), min(sal) from emp where deptno=10 group by job having avg(sal)>1000 order by sum(sal) desc;

Output:



```
SQL> select job, sum(sal), avg(sal), max(sal), min(sal) from emp where deptno=10
 group by job having avg(sal)>1000 order by sum(sal) desc;

JOB               SUM(SAL)   AVG(SAL)   MAX(SAL)   MIN(SAL)
--------------- ---------- ---------- ---------- ----------
AsstHead             10000      10000      10000      10000
GM                   10000      10000      10000      10000
Clerk                 9000       9000       9000       9000
SWManager             8000       8000       8000       8000
Supervisor            8000       8000       8000       8000
```

TASK - 6

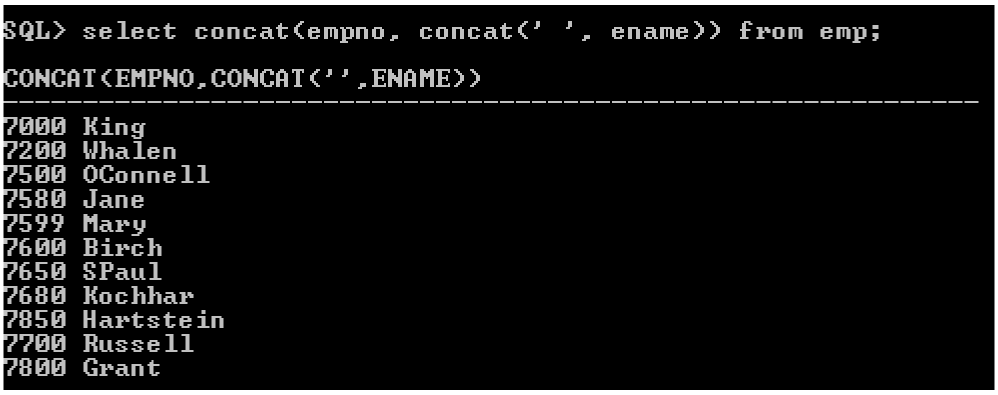
Exercise on SQL Functions

1. Display the employee name concatenated with empno.

Query:

select concat(empno, concat(' ', ename)) from emp;

Output:



```
SQL> select concat(empno, concat(' ', ename)) from emp;

CONCAT(EMPNO,CONCAT('',ENAME))
--------------------------------------------------------------
7000 King
7200 Whalen
7500 OConnell
7580 Jane
7599 Mary
7600 Birch
7650 SPaul
7680 Kochhar
7850 Hartstein
7700 Russell
7800 Grant
```

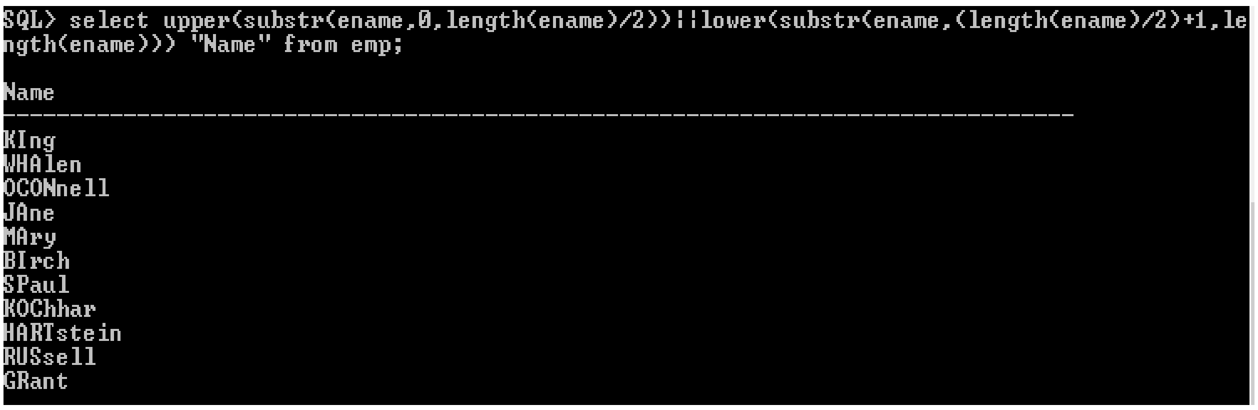
2. Display half of employee name in upper case and half in lower case.

Query:

Select

upper(substr(ename,0,length(ename)/2))||lower(substr(ename,(length(ename)/2)+1,length(ena me))) “Name” from emp;

Output:



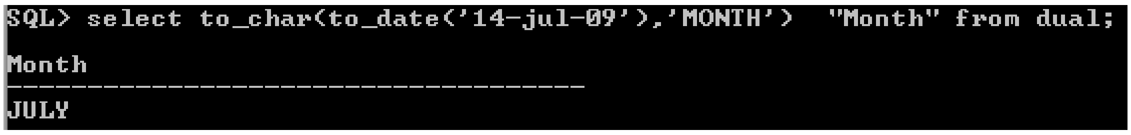
```
SQL> select upper(substr(ename,0,length(ename)/2))||lower(substr(ename,(length(ename)/2)+1,le
ngth(ename))) "Name" from emp;

Name
--------------------------------------------------------------------------------
KIng
WHAlen
OCONnell
JAne
MAry
BIrch
SPaul
KOChhar
HARTstein
RUSsell
GRant
```

3. Display the month name of date “14-jul-09” in full.

Query:

select to_char(to_date('14-jul-09'),'MONTH') “Month” from dual;

Output:



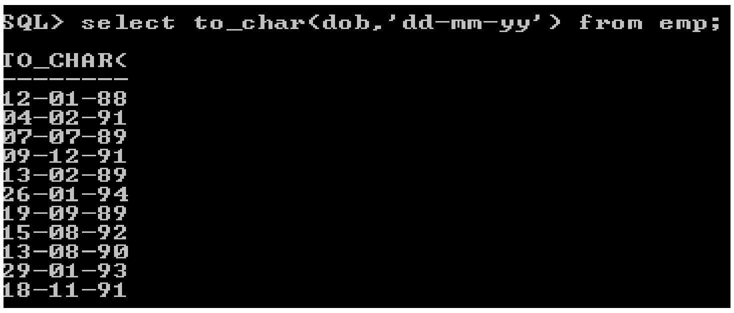
```
SQL> select to_char(to_date('14-jul-09'),'MONTH')  "Month" from dual;

Month
------------------------------------
JULY
```

4. Display the DOB of all employees in the format 'dd-mm-yy'.

Query:

select to_char(dob,'dd-mm-yy') from emp;

Output:



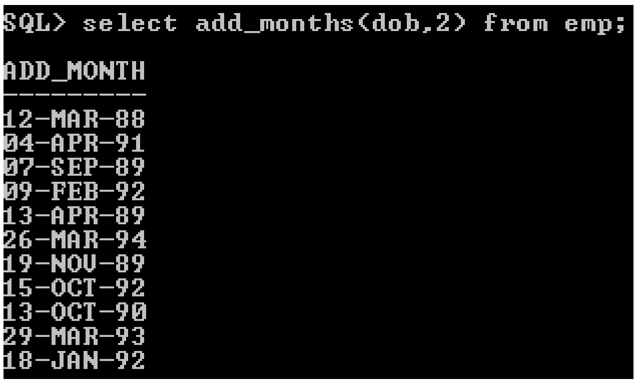
```
SQL> select to_char(dob,'dd-mm-yy') from emp;

TO_CHAR(
--------
12-01-88
04-02-91
07-07-89
09-12-91
13-02-89
26-01-94
19-09-89
15-08-92
13-08-90
29-01-93
18-11-91
```

5. Display the date two months after the DOB of employees.

Query:

select add_months(dob,2) from emp;

Output:



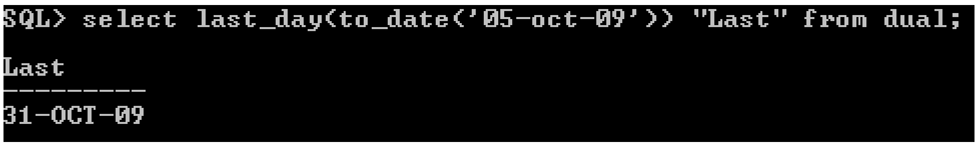
```
SQL> select add_months(dob,2) from emp;

ADD_MONTH
---------
12-MAR-88
04-APR-91
07-SEP-89
09-FEB-92
13-APR-89
26-MAR-94
19-NOV-89
15-OCT-92
13-OCT-90
29-MAR-93
18-JAN-92
```

6. Display the last date of that month in "05-Oct-09".

Query:

select last_day(to_date('05-oct-09')) "Last" from dual;

Output:



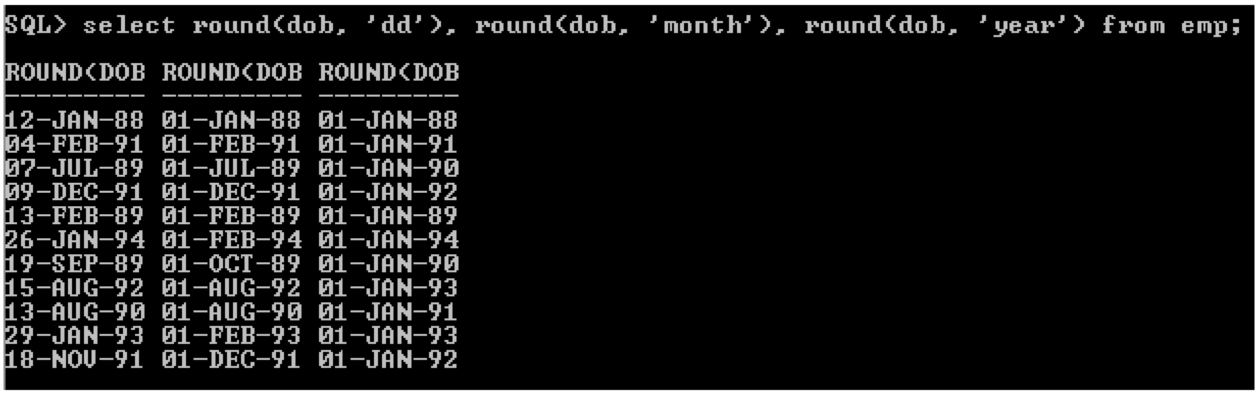
```
SQL> select last_day(to_date('05-oct-09')) "Last" from dual;

Last
---------
31-OCT-09
```

7. Display the rounded date in the year format, month format, day format in the employee.

Query:

select round(dob, 'dd'), round(dob, 'month'), round(dob, 'year') from emp;

Output:

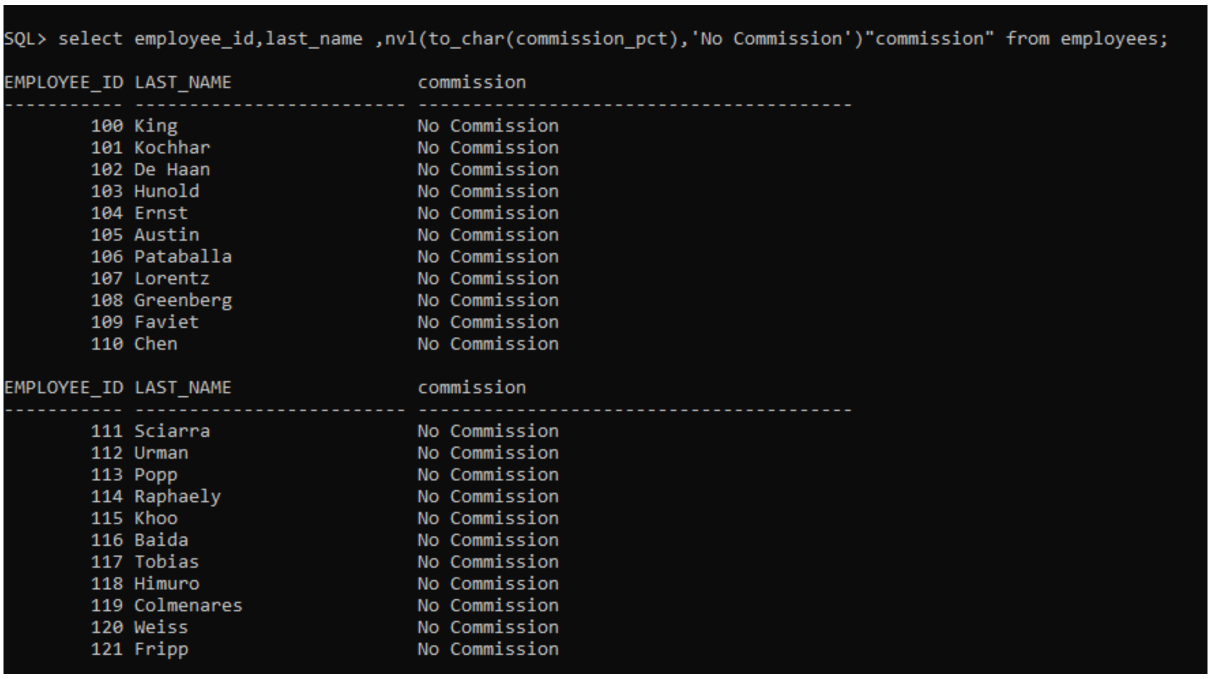


```
SQL> select round(dob, 'dd'), round(dob, 'month'), round(dob, 'year') from emp;

ROUND(DOB ROUND(DOB ROUND(DOB
--------- --------- ---------
12-JAN-88 01-JAN-88 01-JAN-88
04-FEB-91 01-FEB-91 01-JAN-91
07-JUL-89 01-JUL-89 01-JAN-90
09-DEC-91 01-DEC-91 01-JAN-92
13-FEB-89 01-FEB-89 01-JAN-89
26-JAN-94 01-FEB-94 01-JAN-94
19-SEP-89 01-OCT-89 01-JAN-90
15-AUG-92 01-AUG-92 01-JAN-93
13-AUG-90 01-AUG-90 01-JAN-91
29-JAN-93 01-FEB-93 01-JAN-93
18-NOV-91 01-DEC-91 01-JAN-92
```

8.Display the commissions earned by employees. If they do not earn commission, display it as “No Commission”.

Query:

select employee_id,last_name,nvl(to_char(commission_pct),’No Commission’)”commission” from employees;



```
SQL> select employee_id,last_name ,nvl(to_char(commission_pct),'No Commission')"commission" from employees;

EMPLOYEE_ID LAST_NAME                 commission
----------- ------------------------- ----------------------------------------
        100 King                      No Commission
        101 Kochhar                   No Commission
        102 De Haan                   No Commission
        103 Hunold                    No Commission
        104 Ernst                     No Commission
        105 Austin                    No Commission
        106 Pataballa                 No Commission
        107 Lorentz                   No Commission
        108 Greenberg                 No Commission
        109 Faviet                    No Commission
        110 Chen                      No Commission

EMPLOYEE_ID LAST_NAME                 commission
----------- ------------------------- ----------------------------------------
        111 Sciarra                   No Commission
        112 Urman                     No Commission
        113 Popp                      No Commission
        114 Raphaely                  No Commission
        115 Khoo                      No Commission
        116 Baida                     No Commission
        117 Tobias                    No Commission
        118 Himuro                    No Commission
        119 Colmenares                No Commission
        120 Weiss                     No Commission
        121 Fripp                     No Commission
```

TASK – 7

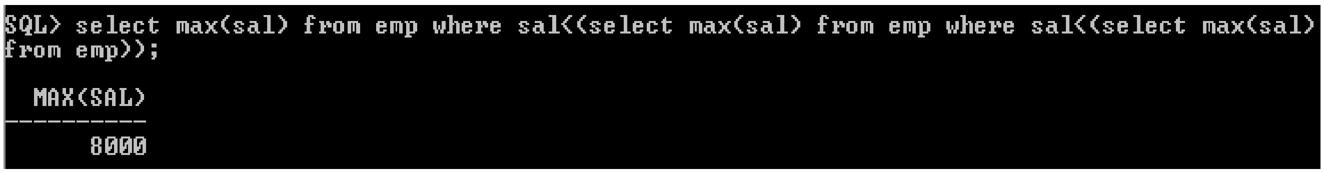
Nested Queries

1. Find the third highest salary of the employees.

Query:

select max(sal) from emp where sal<(select max(sal) from emp where sal<(select max(sal) from emp));

Output:



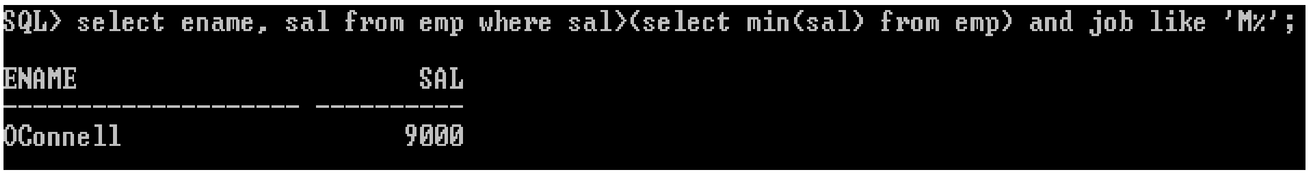
```
SQL> select max(sal) from emp where sal<(select max(sal) from emp where sal<(select max(sal)
from emp));

  MAX(SAL)
----------
      8000
```

2. Display all the employee names and salary whose salary is greater than the minimum salary and job title starts with 'M'.

Query:

select ename, sal from emp where sal>(select min(sal) from emp) and job like 'M%';

Output:



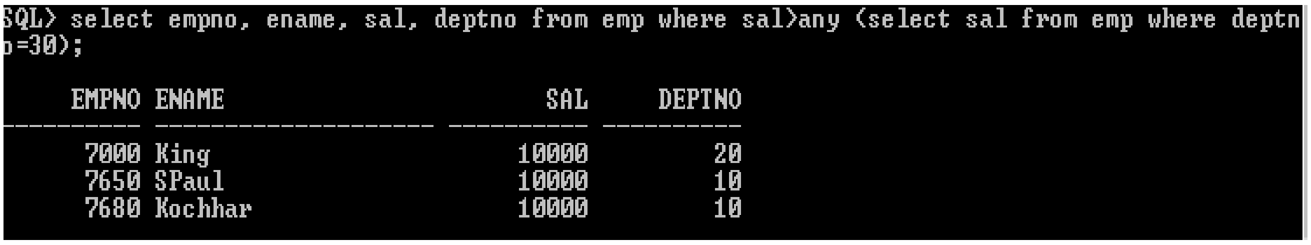
```
SQL> select ename, sal from emp where sal>(select min(sal) from emp) and job like 'M%';

ENAME                       SAL
-------------------- ----------
OConnell                   9000
```

3. Write a Query to display information about employees who earn more than any employee in department 30.

Query:

select empno, ename, sal, deptno from emp where sal>any (select sal from emp where deptno=30);

Output:



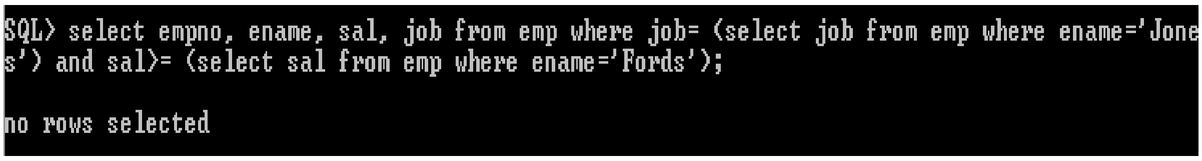
```
SQL> select empno, ename, sal, deptno from emp where sal>any (select sal from emp where deptn
o=30);

     EMPNO ENAME                       SAL     DEPTNO
---------- -------------------- ---------- ----------
      7000 King                      10000         20
      7650 SPaul                     10000         10
      7680 Kochhar                   10000         10
```

4. Display the employees who have the same job as Jones and whose salary>=Fords.

Query:

select empno, ename, sal, job from emp where job= (select job from emp where ename='Jones') and sal>= (select sal from emp where ename='Fords');

Output:



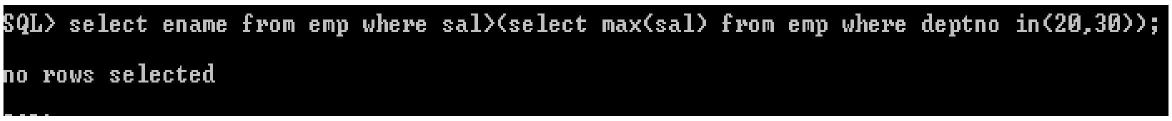
```
SQL> select empno, ename, sal, job from emp where job= (select job from emp where ename='Jone
s') and sal>= (select sal from emp where ename='Fords');

no rows selected
```

5. List out the employee names who get the salary> maximum salary of dept with deptno 20,30.

Query:

select ename from emp where sal>(select max(sal) from emp where deptno in(20,30));

Output:



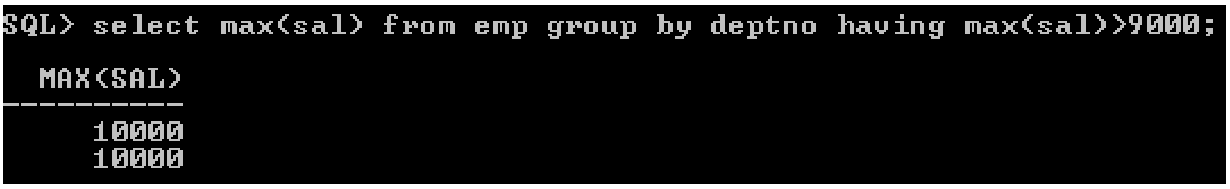
```
SQL> select ename from emp where sal>(select max(sal) from emp where deptno in(20,30));

no rows selected
```

6. Display the maximum salaries of the departments whose maximum salary>9000.

Query:

select max(sal) from emp group by deptno having max(sal)>9000;

Output:



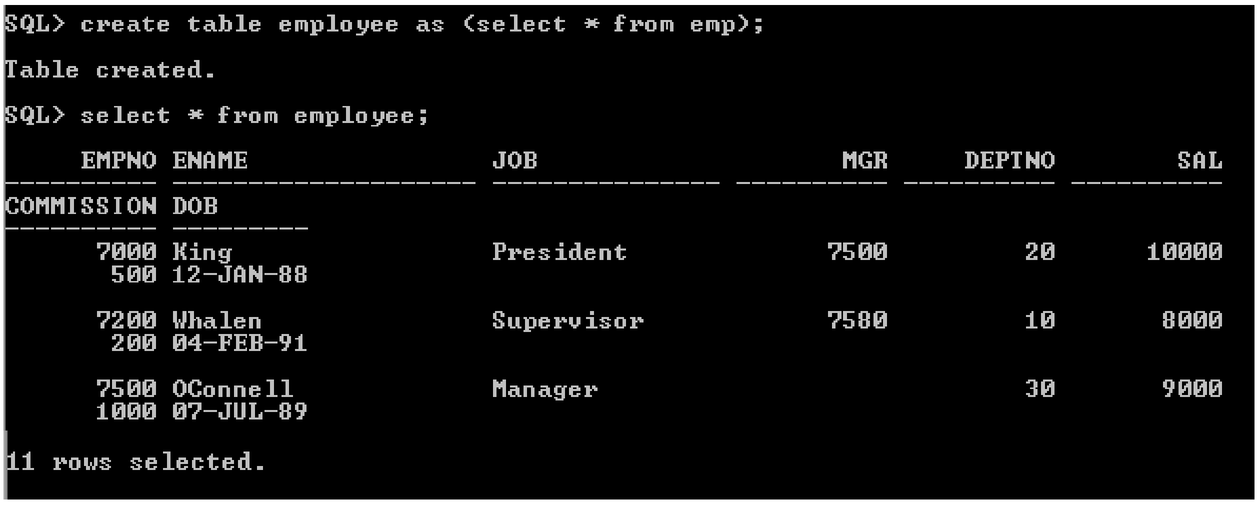
```
SQL> select max(sal) from emp group by deptno having max(sal)>9000;

  MAX(SAL)
----------
     10000
     10000
```

7. Create a table employee with the same structure as the table emp and insert rows into the table using select clauses.

Query:

SQL>create table employee as (select * from emp);

Output:

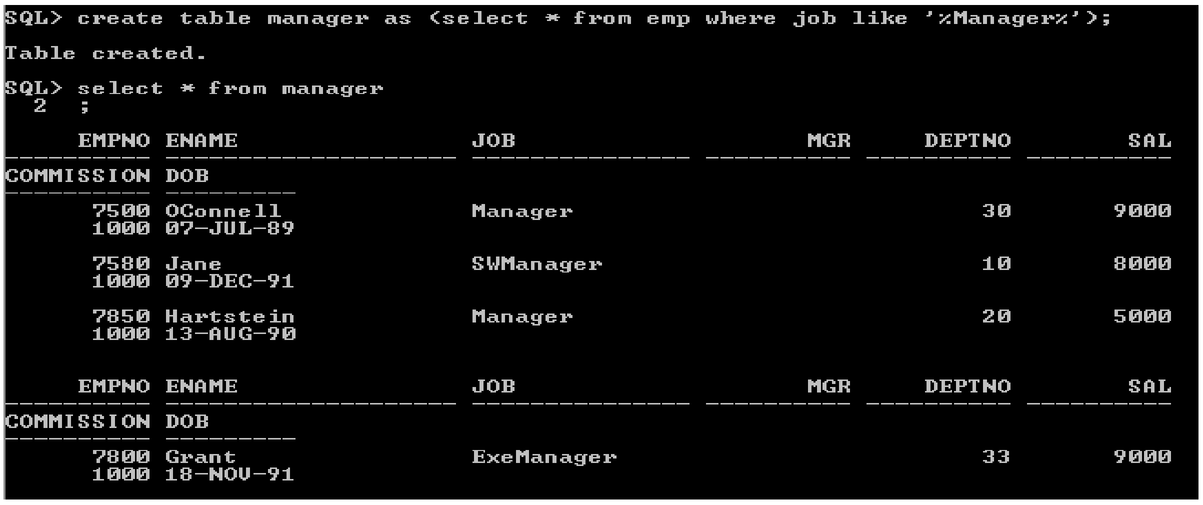


```
SQL> create table employee as (select * from emp);

Table created.

SQL> select * from employee;

     EMPNO ENAME                JOB                   MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7000 King                 President            7500         20      10000
       500 12-JAN-88

      7200 Whalen               Supervisor           7580         10       8000
       200 04-FEB-91

      7500 OConnell             Manager                           30       9000
      1000 07-JUL-89

11 rows selected.
```

8. Create a manager table from the emp table which should hold details only about managers.

Query:

SQL>create table manager as (select * from emp where job like '%Manager%');



```
SQL> create table manager as (select * from emp where job like '%Manager%');

Table created.

SQL> select * from manager
  2  ;

     EMPNO ENAME                JOB                   MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7500 OConnell             Manager                           30       9000
      1000 07-JUL-89

      7580 Jane                 SWManager                         10       8000
      1000 09-DEC-91

      7850 Hartstein            Manager                           20       5000
      1000 13-AUG-90

     EMPNO ENAME                JOB                   MGR     DEPTNO        SAL
---------- -------------------- --------------- ---------- ---------- ----------
COMMISSION DOB
---------- ---------
      7800 Grant                ExeManager                        33       9000
      1000 18-NOV-91
```

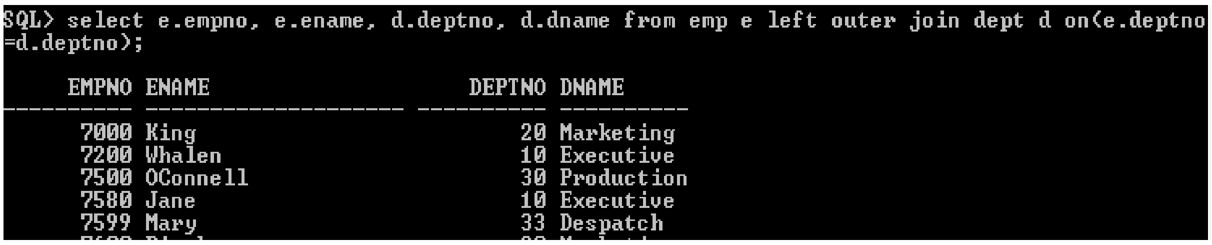
TASK – 8

Joins, Set Operators

1. Display all the employees and departments implementing left outer join.

Query: select e.empno, e.ename, d.deptno, d.dname from emp e left outer join dept d on(e.deptno=d.deptno);

Output:

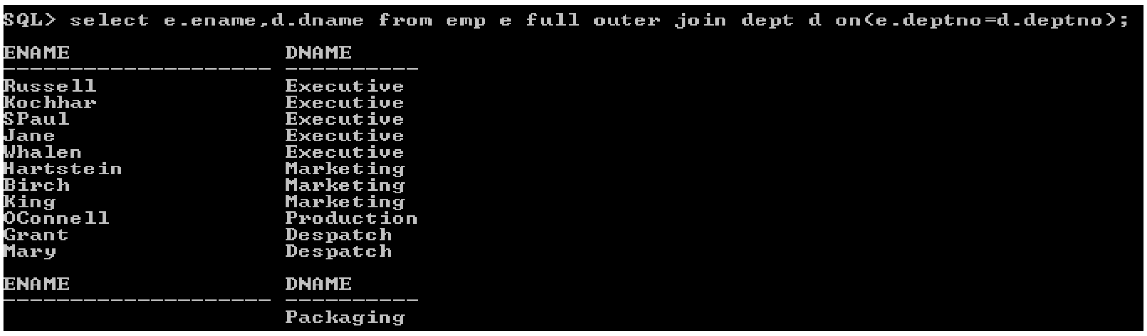


```
SQL> select e.empno, e.ename, d.deptno, d.dname from emp e left outer join dept d on(e.deptno
=d.deptno);

     EMPNO ENAME                    DEPTNO DNAME
---------- -------------------- ---------- ----------
      7000 King                         20 Marketing
      7200 Whalen                       10 Executive
      7500 OConnell                     30 Production
      7580 Jane                         10 Executive
      7599 Mary                         33 Despatch
```

2. Display the employee name and department name in which they are working implementing a full outer join.

Query: select e.ename,d.dname from emp e full outer join dept d on(e.deptno=d.deptno);

Output:

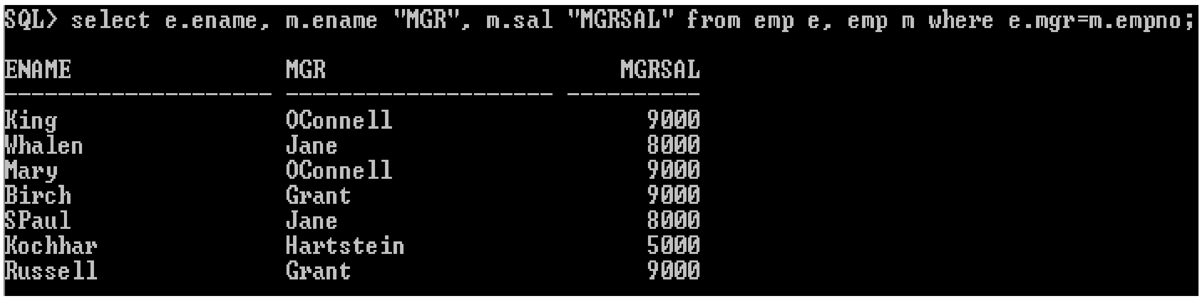


```
SQL> select e.ename,d.dname from emp e full outer join dept d on(e.deptno=d.deptno);

ENAME                DNAME
-------------------- ----------
Russell              Executive
Kochhar              Executive
SPaul                Executive
Jane                 Executive
Whalen               Executive
Hartstein            Marketing
Birch                Marketing
King                 Marketing
OConnell             Production
Grant                Despatch
Mary                 Despatch

ENAME                DNAME
-------------------- ----------
                     Packaging
```

3. Write a Query to display the employee name and manager’s name and salary for all employees.

Query: select e.ename, m.ename “MGR”, m.sal “MGRSAL” from emp e, emp m where e.mgr=m.empno;

Output:

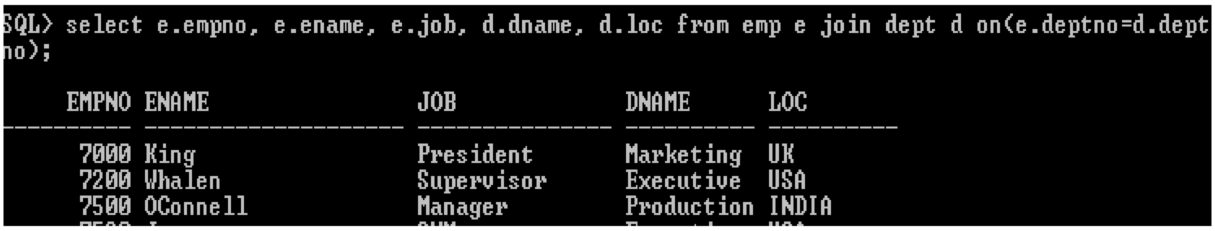


```
SQL> select e.ename, m.ename "MGR", m.sal "MGRSAL" from emp e, emp m where e.mgr=m.empno;

ENAME                MGR                      MGRSAL
-------------------- -------------------- ----------
King                 OConnell                   9000
Whalen               Jane                       8000
Mary                 OConnell                   9000
Birch                Grant                      9000
SPaul                Jane                       8000
Kochhar              Hartstein                  5000
Russell              Grant                      9000
```

4. Write a Query to Output the name, job, employee number, department name, location for each department even if there are no employees.

Query: select e.empno, e.ename, e.job, d.dname, d.loc from emp e join dept d on(e.deptno=d.deptno);

Output:

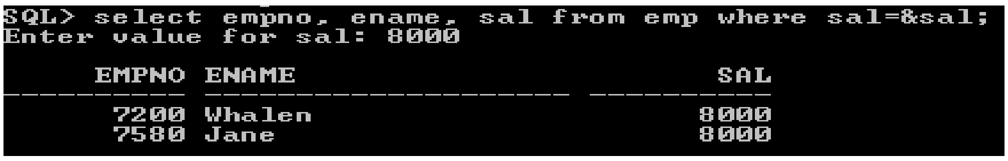


```
SQL> select e.empno, e.ename, e.job, d.dname, d.loc from emp e join dept d on(e.deptno=d.dept
no);

     EMPNO ENAME                JOB             DNAME      LOC
---------- -------------------- --------------- ---------- ----------
      7000 King                 President       Marketing  UK
      7200 Whalen               Supervisor      Executive  USA
      7500 OConnell             Manager         Production INDIA
```

5. Display the details of those who draw the same salary.

Query:select empno, ename, sal from emp where sal=&sal;

Output:



```
SQL> select empno, ename, sal from emp where sal=&sal;
Enter value for sal: 8000

     EMPNO ENAME                       SAL
---------- -------------------- ----------
      7200 Whalen                     8000
      7580 Jane                       8000
```

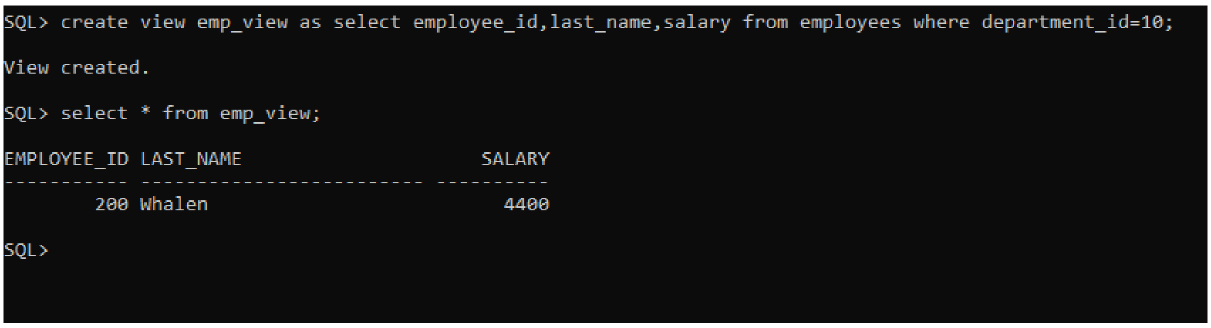
TASK – 9

Views

1.Create a view that displays the employee id, name and salary of employees who belong to 10th department.

Query:

Create view emp_view as select employee_id,last_name,salary from employees where department_id=10;

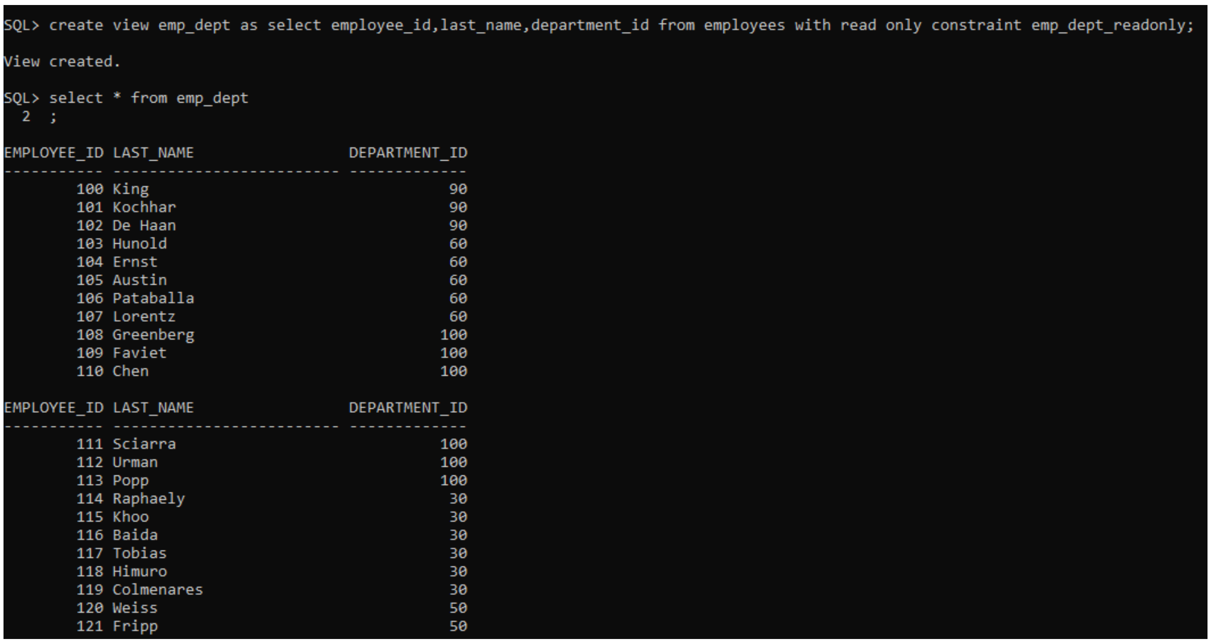


```
SQL> create view emp_view as select employee_id,last_name,salary from employees where department_id=10;

View created.

SQL> select * from emp_view;

EMPLOYEE_ID LAST_NAME                     SALARY
----------- ------------------------- ----------
        200 Whalen                          4400

SQL>
```

2.Create a view with read only option that displays the employee name and their department name

Query:

create view emp_dept as select employee_id,last_name,department_id from employees with read onlyh constraint emp_dept_readonly;



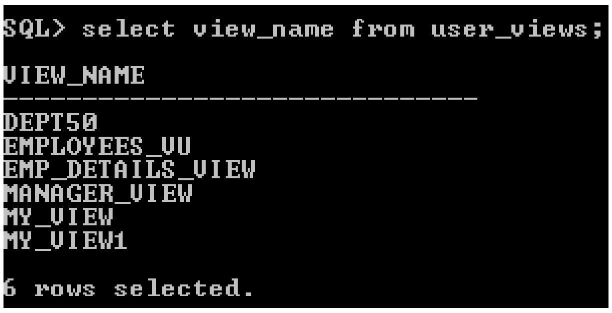
```
SQL> create view emp_dept as select employee_id,last_name,department_id from employees with read only constraint emp_dept_readonly;

View created.

SQL> select * from emp_dept
  2  ;

EMPLOYEE_ID LAST_NAME                 DEPARTMENT_ID
----------- ------------------------- -------------
        100 King                                 90
        101 Kochhar                              90
        102 De Haan                              90
        103 Hunold                               60
        104 Ernst                                60
        105 Austin                               60
        106 Pataballa                            60
        107 Lorentz                              60
        108 Greenberg                           100
        109 Faviet                              100
        110 Chen                                100

EMPLOYEE_ID LAST_NAME                 DEPARTMENT_ID
----------- ------------------------- -------------
        111 Sciarra                             100
        112 Urman                               100
        113 Popp                                100
        114 Raphaely                             30
        115 Khoo                                 30
        116 Baida                                30
        117 Tobias                               30
        118 Himuro                               30
        119 Colmenares                           30
        120 Weiss                                50
        121 Fripp                                50
```

3. Display all the views generated.

Query:

select view_name from user_views;

Output:



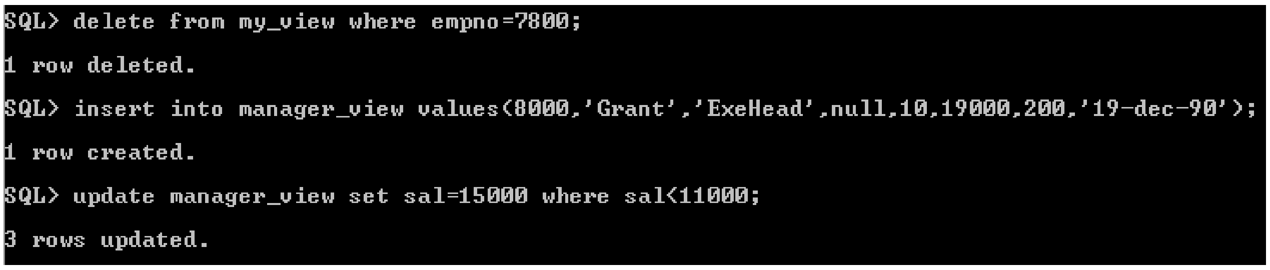
```
SQL> select view_name from user_views;

VIEW_NAME
------------------------------
DEPT50
EMPLOYEES_VU
EMP_DETAILS_VIEW
MANAGER_VIEW
MY_VIEW
MY_VIEW1

6 rows selected.
```

4. Execute the DML commands on the view created.and drop them.

Query:

- delete from my_view where empno=7900;
- insert into manager_view values(8000, 'Grant', 'ExeHead', null, 10, 19000, 200, '19-dec-90');
- update manager_view set sal=15000 where sal<11000;

Output:

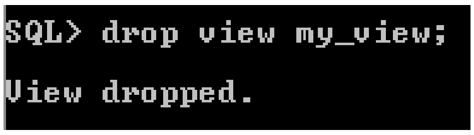


```
SQL> delete from my_view where empno=7800;

1 row deleted.

SQL> insert into manager_view values(8000,'Grant','ExeHead',null,10,19000,200,'19-dec-90');

1 row created.

SQL> update manager_view set sal=15000 where sal<11000;

3 rows updated.
```

Drop a view.

Query: drop view my_view;

Output:



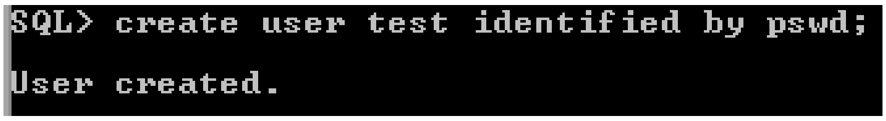
```
SQL> drop view my_view;

View dropped.
```

TASK – 10

Practices on DCL Commands

1. SQL>Create user test identified by pswd;

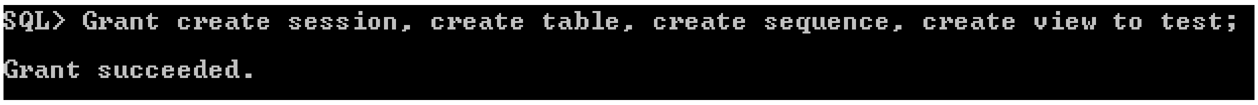
Output:



```
SQL> create user test identified by pswd;

User created.
```

2.SQL> Grant create session, create table, create sequence, create view to test;

Output:



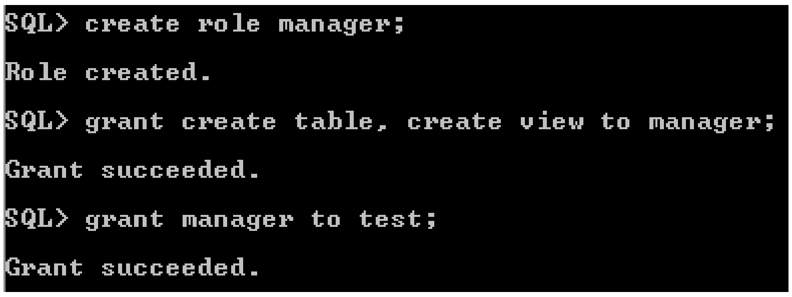
```
SQL> Grant create session, create table, create sequence, create view to test;

Grant succeeded.
```

3. SQL>Create role manager;

SQL>Grant create table, create view to manager;

SQL>Grant manager to test;

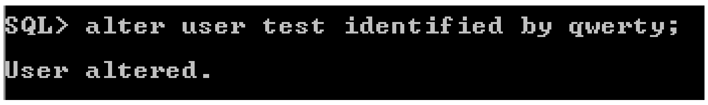
Output:



```
SQL> create role manager;

Role created.

SQL> grant create table, create view to manager;

Grant succeeded.

SQL> grant manager to test;

Grant succeeded.
```

4. SQL>Alter user test identified by qwerty;

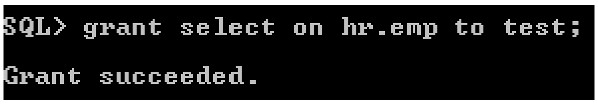
Output:



```
SQL> alter user test identified by qwerty;

User altered.
```

5. SQL>Grant select on employees to test;

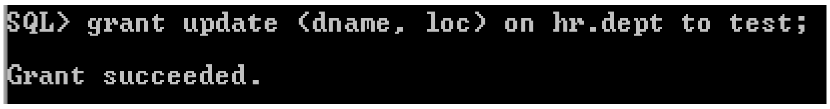
Output:



```
SQL> grant select on hr.emp to test;

Grant succeeded.
```

6. SQL>Grant update (department_name,location_id) on departments to test;

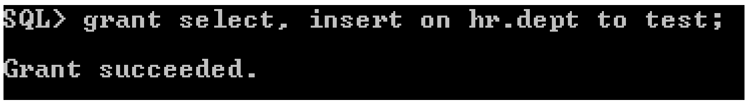
Output:



```
SQL> grant update (dname, loc) on hr.dept to test;

Grant succeeded.
```

7.SQL>Grant select,insert on hr.locations to test;

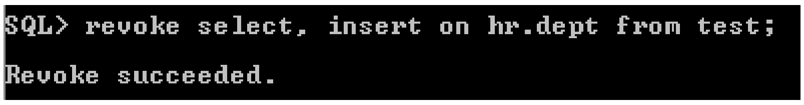
Output:



```
SQL> grant select, insert on hr.dept to test;

Grant succeeded.
```

8. SQL>Revoke select,insert on departments from test;

Output:



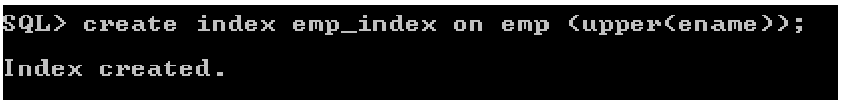
```
SQL> revoke select, insert on hr.dept from test;

Revoke succeeded.
```

INDEXES

1.Function based indexes:

SQL>create index emp_index on emp (upper(ename));

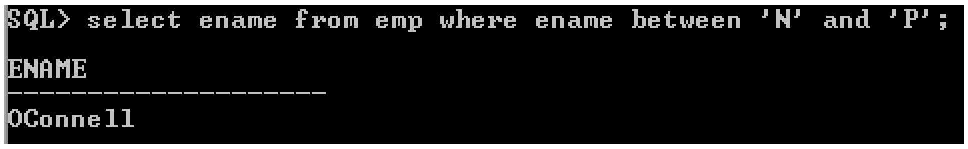
Output:



```
SQL> create index emp_index on emp (upper(ename));

Index created.
```

SQL>select employee_id,last_name,job_id from employees where last_name between 'N' and 'P';

Output:

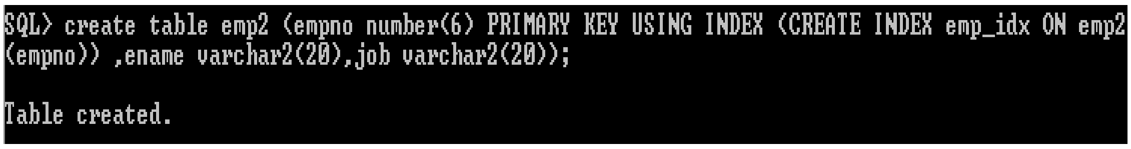


```
SQL> select ename from emp where ename between 'N' and 'P';

ENAME
--------------------
OConnell
```

2. Creating Index while creating Table.

SQL>create table emp2 (empnonumber(6) PRIMARY KEY USING INDEX (CREATE INDEX emp_idx ON emp2(empno)) ,ename varchar2(20),job varchar2(20));

Output:

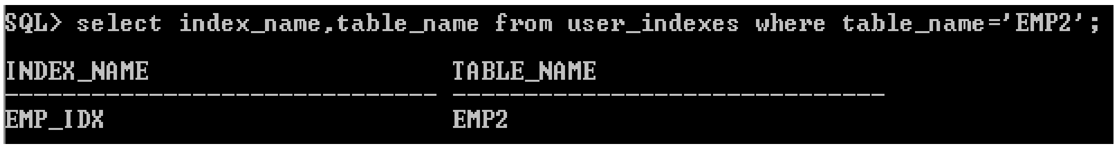


```
SQL> create table emp2 (empno number(6) PRIMARY KEY USING INDEX (CREATE INDEX emp_idx ON emp2
(empno)) ,ename varchar2(20),job varchar2(20));

Table created.
```

User-defined indexes:

SQL>select index_name,table_name from user_indexes where table_name='EMP2';

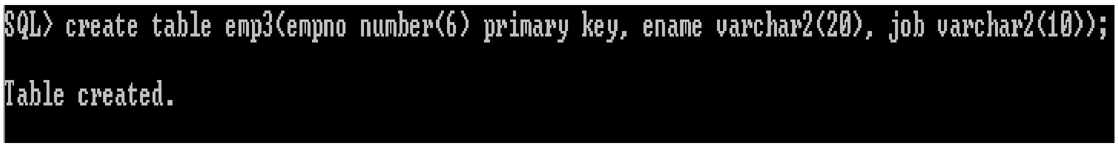
Output:



```
SQL> select index_name,table_name from user_indexes where table_name='EMP2';

INDEX_NAME                     TABLE_NAME
------------------------------ ------------------------------
EMP_IDX                        EMP2
```

3. create table emp3(empnonumber(6) primary key, ename varchar2(20), job varchar2(10));

Output:

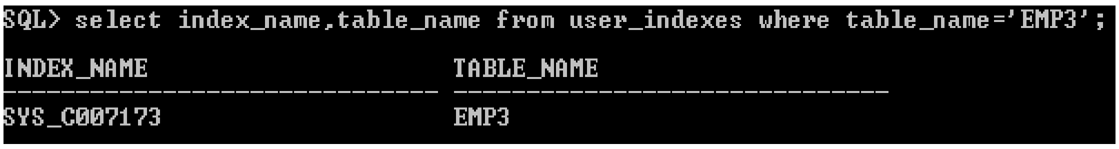


```
SQL> create table emp3(empno number(6) primary key, ename varchar2(20), job varchar2(10));

Table created.
```

Default indexes.

SQL>select index_name,table_name from user_indexes where table_name='EMP3';

Output:

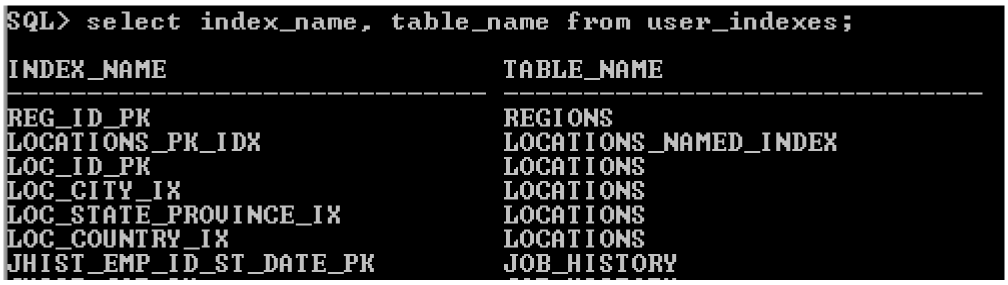


```
SQL> select index_name,table_name from user_indexes where table_name='EMP3';

INDEX_NAME                     TABLE_NAME
------------------------------ ------------------------------
SYS_C007173                    EMP3
```

4.Displaying all the indexes.

SQL>Select index_name, table_name from user_indexes;

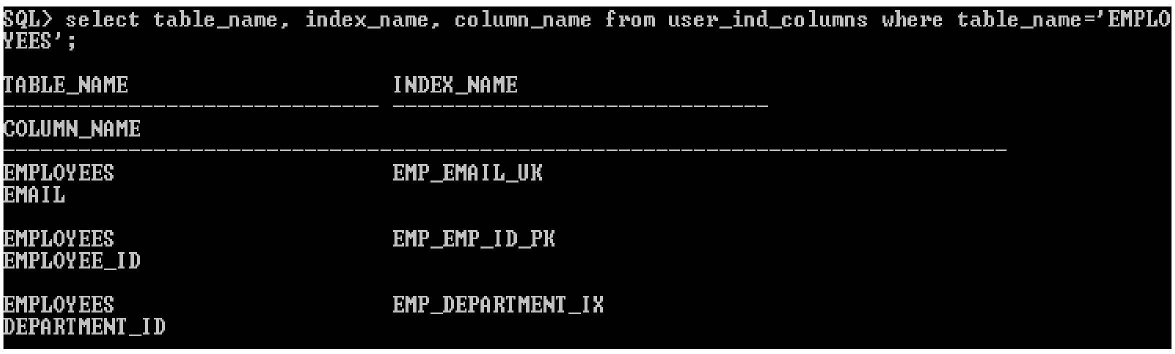
Output:



```
SQL> select index_name, table_name from user_indexes;

INDEX_NAME                     TABLE_NAME
------------------------------ ------------------------------
REG_ID_PK                      REGIONS
LOCATIONS_PK_IDX               LOCATIONS_NAMED_INDEX
LOC_ID_PK                      LOCATIONS
LOC_CITY_IX                    LOCATIONS
LOC_STATE_PROVINCE_IX          LOCATIONS
LOC_COUNTRY_IX                 LOCATIONS
JHIST_EMP_ID_ST_DATE_PK        JOB_HISTORY
```

4.SQL>select table_name, index_name, column_name from user_ind_columns where table_name='EMPLOYEES';

Output:

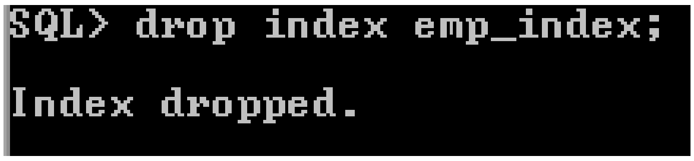


```
SQL> select table_name, index_name, column_name from user_ind_columns where table_name='EMPLO
YEES';

TABLE_NAME                     INDEX_NAME
------------------------------ ------------------------------
COLUMN_NAME
--------------------------------------------------------------------------------
EMPLOYEES                      EMP_EMAIL_UK
EMAIL

EMPLOYEES                      EMP_EMP_ID_PK
EMPLOYEE_ID

EMPLOYEES                      EMP_DEPARTMENT_IX
DEPARTMENT_ID
```

5. Dropping an index:

SQL> drop index emp_index;

Output:

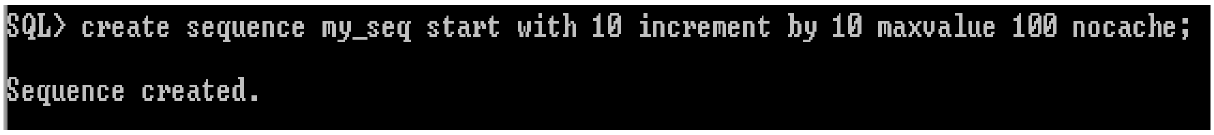


```
SQL> drop index emp_index;

Index dropped.
```

SEQUENCE

1. SQL>create sequence my_seq start with 10 increment by 10 maxvalue 100 nocache;

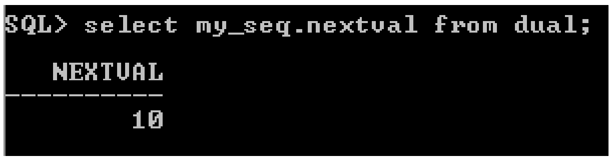
Output:



```
SQL> create sequence my_seq start with 10 increment by 10 maxvalue 100 nocache;

Sequence created.
```

2. SQL>select my_seq.nextval from dual;

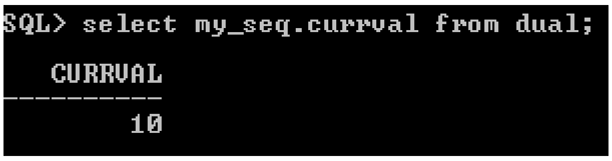
Output:



```
SQL> select my_seq.nextval from dual;

   NEXTVAL
----------
        10
```

3. SQL>select my_seq.currval from dual;

Output:

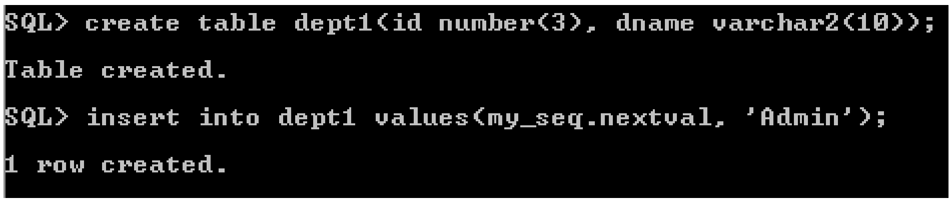


```
SQL> select my_seq.currval from dual;

   CURRVAL
----------
        10
```

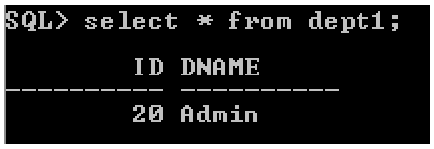
4. SQL>create table dept(deptno number(6),dname varchar2(20),loc varchar2(10));

SQL>insert into dept values(my_seq.nextval,'Executive','US');

SQL>insert into dept values(my_seq.nextval,'Marketing','UK');

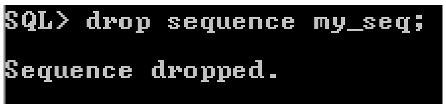


```
SQL> create table dept1(id number(3), dname varchar2(10));

Table created.

SQL> insert into dept1 values(my_seq.nextval, 'Admin');

1 row created.
```



```
SQL> select * from dept1;

        ID DNAME
---------- ----------
        20 Admin
```

5. SQL>drop sequence my_seq;



```
SQL> drop sequence my_seq;

Sequence dropped.
```

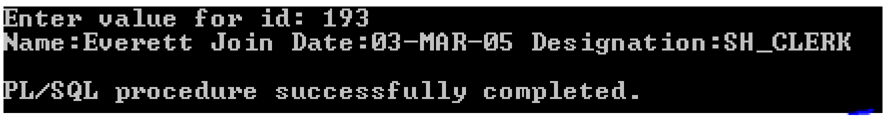
TASK - 11

1. Write a PL/SQL code to retrieve the employee name, join date and designation from employee database of an employee whose number is input by the user.

Program:

```
/*Employee details*/

DECLARE

v_name varchar2(25);

v_joindate date;

v_dsgn employees.job_id%type;

BEGIN

select last_name,hire_date,job_id into v_name,v_joindate,v_dsgn from employees where
employee_id=&id;

DBMS_OUTPUT.PUT_LINE('Name:'||v_name||' Join Date:'||v_joindate||'
Designation:'||v_dsgn);

END;

/
```

Output:



```
Enter value for id: 193
Name:Everett Join Date:03-MAR-05 Designation:SH_CLERK

PL/SQL procedure successfully completed.
```

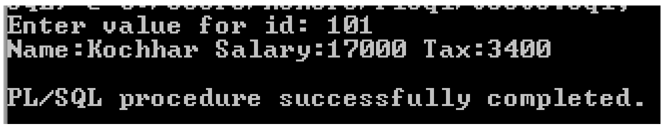
2. Write a PL/SQL code to calculate tax for an employee of an organization.

Program:

```
/*Calculate Tax*/

DECLARE

v_sal number(8);

v_tax number(8,3);
```

```
v_name varchar2(25);

BEGIN

select salary,last_name into v_sal,v_name from employees where employee_id=&id;

if v_sal<10000 then

 v_tax:=v_sal*0.1;

elsif v_sal between 10000 and 20000 then

 v_tax:=v_sal*0.2;

else

 v_tax:=v_sal*0.3;

END IF;

DBMS_OUTPUT.PUT_LINE('Name:'||v_name||' Salary:'||v_sal||'Tax:'||v_tax);

END;

/
```

Output:



```
Enter value for id: 101
Name:Kochhar Salary:17000 Tax:3400

PL/SQL procedure successfully completed.
```

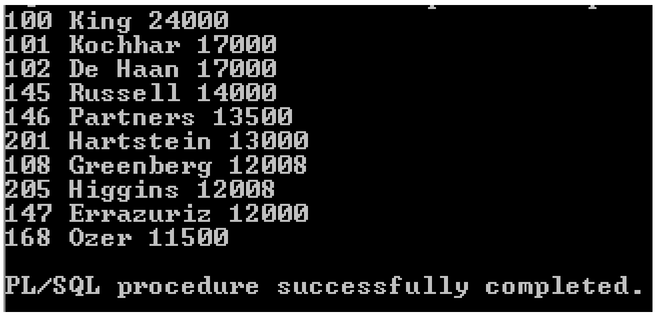
3. Write a PL/SQL program to display top 10 employee details based on salary using cursors.

Program:

```
/*Top 10 salary earning employee details*/

DECLARE

cursor c_emp_cursor is select employee_id, last_name, salary from employees order by
salary desc;

v_rec c_emp_cursor%rowtype;

v_i number(3):=0;

BEGIN

open c_emp_cursor;

loop
```

```
 v_i:=v_i+1;
 fetch c_emp_cursor into v_rec;
 exit when v_i>10;
 DBMS_OUTPUT.PUT_LINE(v_rec.employee_id||' '||v_rec.last_name||' '||v_rec.salary);
 END LOOP;
close c_emp_cursor;
END;
/
```

Output:



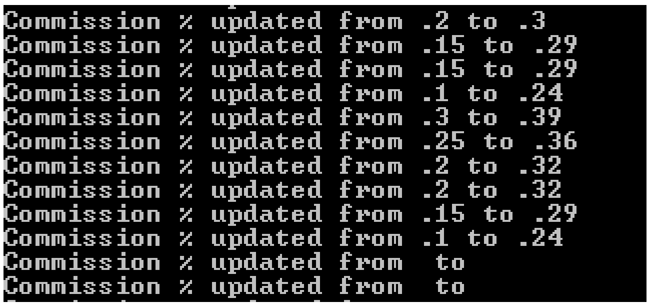
```
100 King 24000
101 Kochhar 17000
102 De Haan 17000
145 Russell 14000
146 Partners 13500
201 Hartstein 13000
108 Greenberg 12008
205 Higgins 12008
147 Errazuriz 12000
168 Ozer 11500

PL/SQL procedure successfully completed.
```

4. Write a PL/SQL program to update the commission values for all employees with salary less than 5000 by adding 1000 to existing employees.

Program:

```
/*Updation*/
declare
cursor c_emp is select salary,commission_pct from employees;
v_emp c_emp%rowtype;
v_temp number(7,2);
v_temp1 number;
BEGIN
open c_emp;
loop
 fetch c_emp into v_emp;
 exit when c_emp%notfound;
  v_temp1:=v_emp.commission_pct;
  v_temp:=(v_emp.salary*v_emp.commission_pct)+1000;
  v_temp:=v_temp/v_emp.salary;
  if(v_emp.salary<5000) then
  update employees set commission_pct=v_temp where   employee_id=v_temp.employee_id;
  end if;
  DBMS_OUTPUT.PUT_LINE('Commission % updated from '||v_temp1||' to '||v_temp);
 end loop;
END;
/
```

Output:



```
Commission % updated from .2 to .3
Commission % updated from .15 to .29
Commission % updated from .15 to .29
Commission % updated from .1 to .24
Commission % updated from .3 to .39
Commission % updated from .25 to .36
Commission % updated from .2 to .32
Commission % updated from .2 to .32
Commission % updated from .15 to .29
Commission % updated from .1 to .24
Commission % updated from  to
Commission % updated from  to
```

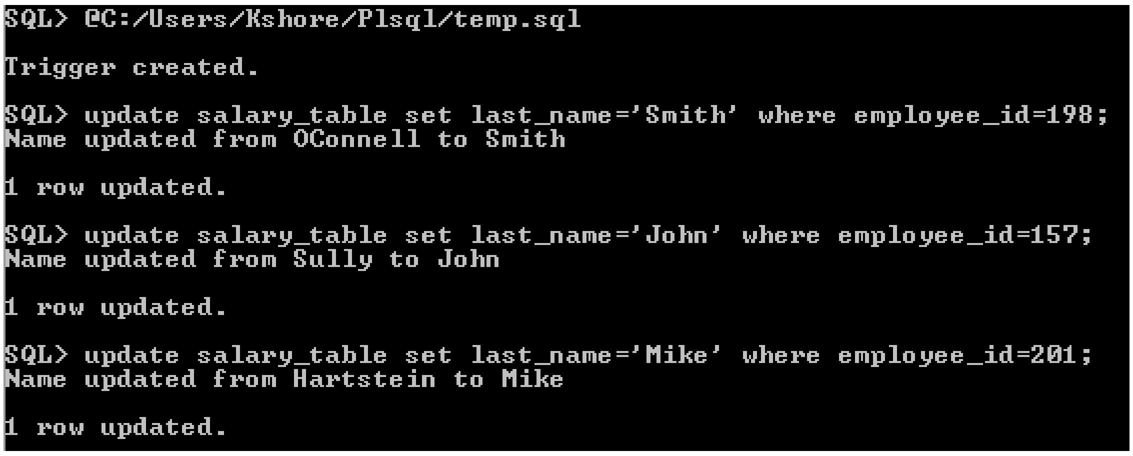
TASK – 12

1. Write a trigger on the employee table which shows the old values and new values of ename after any updations on ename on Employee table.

Program:

```
create or replace trigger t_emp_name after update of last_name on salary_table FOR EACH ROW

begin

DBMS_OUTPUT.PUT_LINE('Name updated from '||:OLD.last_name||' to '||:NEW.last_name);

END;

/
```

Output:



```
SQL> @C:/Users/Kshore/Plsql/temp.sql

Trigger created.

SQL> update salary_table set last_name='Smith' where employee_id=198;
Name updated from OConnell to Smith

1 row updated.

SQL> update salary_table set last_name='John' where employee_id=157;
Name updated from Sully to John

1 row updated.

SQL> update salary_table set last_name='Mike' where employee_id=201;
Name updated from Hartstein to Mike

1 row updated.
```

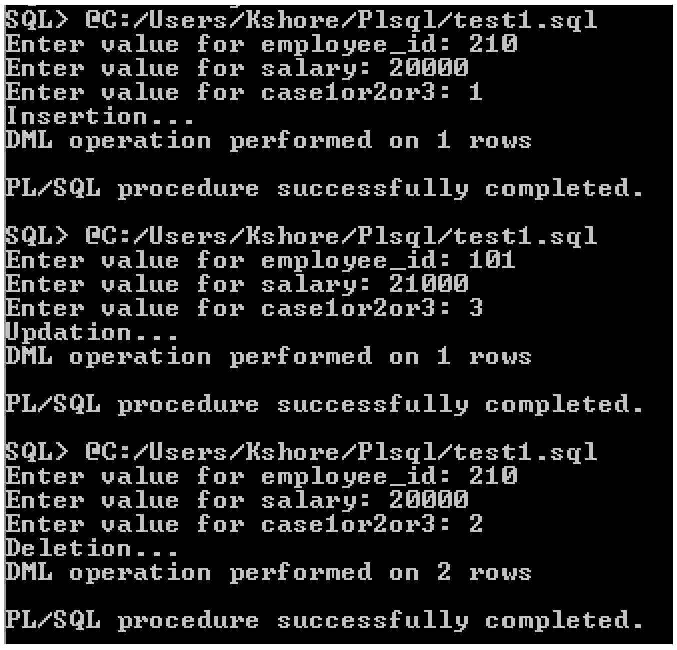
2. Write a PL/SQL procedure for inserting, deleting and updating in employee table.

Program:

```
create or replace procedure proc_dml (p_id emp.employee_id%type, p_sal number,p_case number)

is

BEGIN

case p_case

        when 1 then

            DBMS_OUTPUT.PUT_LINE('Insertion...');

                insert into emp(employee_id,last_name,email,hire_date,job_id) values(p_id,'Franco',’FJames’,'12-JAN-02','ST_CLERK');

        when 2 then
```

```
                DBMS_OUTPUT.PUT_LINE('Deletion...');
                delete from emp where employee_id=p_id;
        when 3 then
                DBMS_OUTPUT.PUT_LINE('Updation...');
                update emp set salary=p_sal where employee_id=p_id;
        end case;
DBMS_OUTPUT.PUT_LINE('DML operation performed on '||SQL%rowcount||' rows');
END;
/
DECLARE
v_id employees.employee_id%type:=&id;
v_sal employees.salary%type:=&sal;
v_case number:=&case1or2or3;
begin
proc_dml(v_id,v_sal,v_case);
END;
/
```

Output:



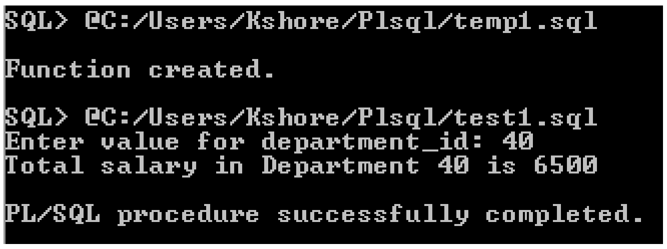
```
SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for employee_id: 210
Enter value for salary: 20000
Enter value for case1or2or3: 1
Insertion...
DML operation performed on 1 rows

PL/SQL procedure successfully completed.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for employee_id: 101
Enter value for salary: 21000
Enter value for case1or2or3: 3
Updation...
DML operation performed on 1 rows

PL/SQL procedure successfully completed.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for employee_id: 210
Enter value for salary: 20000
Enter value for case1or2or3: 2
Deletion...
DML operation performed on 2 rows

PL/SQL procedure successfully completed.
```

3. Write a PL/SQL function that accepts department number and returns the total salary of the department.

Program:

```
create function func_dept (p_dept number) return number is
v_total number;
BEGIN
select sum(salary) into v_total from employees where department_id=p_dept;
return v_total;
END;
/

DECLARE
v_dept number:=&department_id;
v_total number;
BEGIN
v_total:=func_dept(v_dept);
DBMS_OUTPUT.PUT_LINE('Total salary in Department '||v_dept||' is '||v_total);
END;
/
```

Output:



```
SQL> @C:/Users/Kshore/Plsql/temp1.sql

Function created.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for department_id: 40
Total salary in Department 40 is 6500

PL/SQL procedure successfully completed.
```

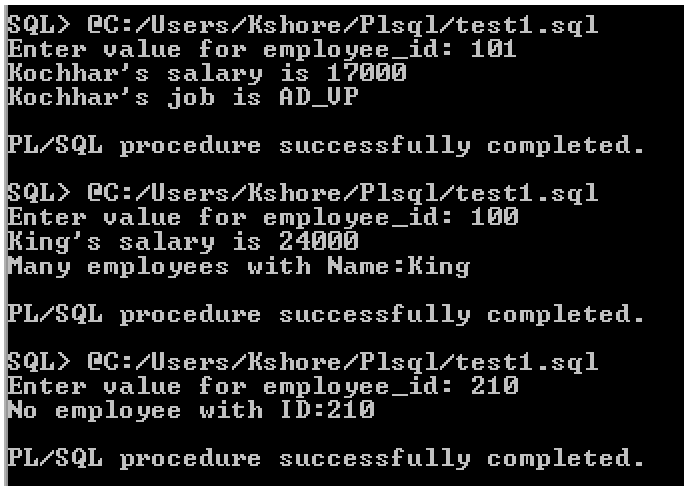
Task-13

1. Write a PL/SQL program to handle predefined exceptions.

Program:

```
declare
v_id number(6):=&employee_id;
v_sal employees.salary%type;
v_name employees.last_name%type;
v_job employees.job_id%type;
begin
select last_name, salary into v_name, v_sal from employees where employee_id=v_id;
DBMS_OUTPUT.PUT_LINE(v_name||q'['s salary is ]'||v_sal);
select job_id into v_job from employees where last_name=v_name;
DBMS_OUTPUT.PUT_LINE(v_name||q'['s job is ]'||v_job);
EXCEPTION
        when no_data_found then
                DBMS_OUTPUT.PUT_LINE('No employee with ID:'||v_id);
        when too_many_rows then
                DBMS_OUTPUT.PUT_LINE('Many employees with Name:'||v_name);
        when others then
                DBMS_OUTPUT.PUT_LINE('Some other error occured');
end;
/
```

Output:



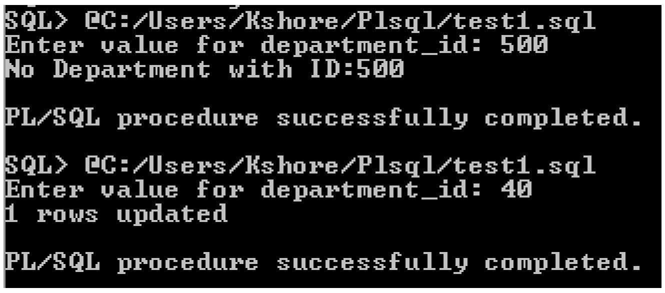
```
SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for employee_id: 101
Kochhar's salary is 17000
Kochhar's job is AD_VP

PL/SQL procedure successfully completed.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for employee_id: 100
King's salary is 24000
Many employees with Name:King

PL/SQL procedure successfully completed.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for employee_id: 210
No employee with ID:210

PL/SQL procedure successfully completed.
```

2. Write a PL/SQL program to handle user defined exception.

Program:

```
DECLARE
v_dept number:=&department_id;
e_nodept exception;
BEGIN
update employees set salary=salary+1050 where department_id=v_dept;
IF SQL%notfound then
  raise e_nodept;
ELSE
  DBMS_OUTPUT.PUT_LINE(SQL%rowcount||' rows updated');
END IF;
EXCEPTION
 when e_nodept then
        DBMS_OUTPUT.PUT_LINE('No Department with ID:'||v_dept)
END;
/
```

Output:



```
SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for department_id: 500
No Department with ID:500

PL/SQL procedure successfully completed.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for department_id: 40
1 rows updated

PL/SQL procedure successfully completed.
```

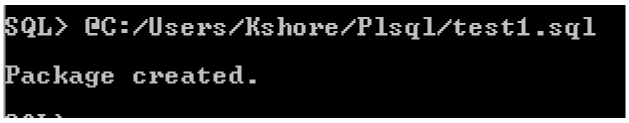
3)Write a PL/SQL code to create

a. Package specification.

Program:

```
create or replace package pack_dml is
        procedure proc_dml(p_id number,choice number);
END pack_dml;
/
```

Output:



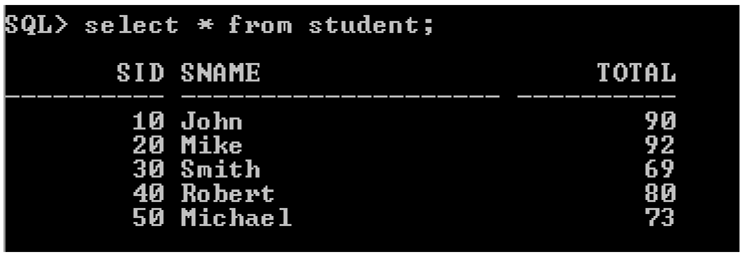
```
SQL> @C:/Users/Kshore/Plsql/test1.sql

Package created.
```

b. Package body for the insert, retrieve, update and delete operations on student table.

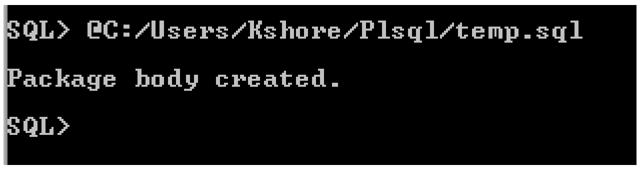
Program:

```
create or replace package body pack_dml is
        procedure proc_dml(p_id number,choice number) is
        v_name varchar2(20);
        v_total number;
        BEGIN
        case choice
                when 1 then
                DBMS_OUTPUT.PUT_LINE('Insertion...');
                insert into student values(p_id,'Franco',90);
                when 2 then
                DBMS_OUTPUT.PUT_LINE('Deletion...');
                delete from student where sid=p_id;
                when 3 then
                DBMS_OUTPUT.PUT_LINE('Updation...');
                update student set total=total+1 where sid=p_id;
                when 4 then
                select sname,total into v_name,v_total from student where sid=p_id;
                DBMS_OUTPUT.PUT_LINE('Total marks of '||v_name||' is '||v_total);
        end case;
        DBMS_OUTPUT.PUT_LINE('DML operation performed on '||SQL%rowcount||'
        rows');
END proc_dml;
END pack_dml;
/

BEGIN
pack_dml.proc_dml(&StudentID,&choice1or2or3or4);
END;
/
```

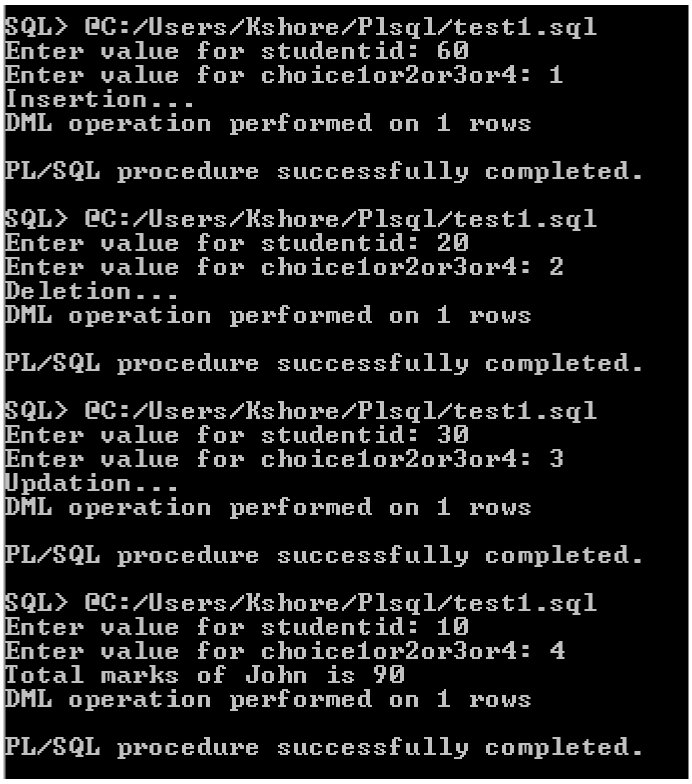
Output:



```
SQL> select * from student;

       SID SNAME                     TOTAL
---------- -------------------- ----------
        10 John                         90
        20 Mike                         92
        30 Smith                        69
        40 Robert                       80
        50 Michael                      73
```



```
SQL> @C:/Users/Kshore/Plsql/temp.sql

Package body created.

SQL>
```



```
SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for studentid: 60
Enter value for choice1or2or3or4: 1
Insertion...
DML operation performed on 1 rows

PL/SQL procedure successfully completed.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for studentid: 20
Enter value for choice1or2or3or4: 2
Deletion...
DML operation performed on 1 rows

PL/SQL procedure successfully completed.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for studentid: 30
Enter value for choice1or2or3or4: 3
Updation...
DML operation performed on 1 rows

PL/SQL procedure successfully completed.

SQL> @C:/Users/Kshore/Plsql/test1.sql
Enter value for studentid: 10
Enter value for choice1or2or3or4: 4
Total marks of John is 90
DML operation performed on 1 rows

PL/SQL procedure successfully completed.
```

Task-14:

Table Locking(Shared Lock and Exclusive Lock)

Problem Description
Oracle uses lock to control concurrent access to data. Locks are mechanisms intended to prevent destructive interaction between users accessing the same data. Table locks lock the entire tables, while row locks lock just selected rows. Thus locks are used to ensure data integrity while allowing max concurrent access to data by unlimited users.
Locks are used to achieve two important goals. 1. Data concurrency 2. Read consistency
Oracle lock is fully automatic and requires no user action .DBA locks the oracle data while executing SQL statement. This type of locking is called implicit locking. When a lock is put by user it is called explicit locking.

Types of locks
Two levels of lock that a DBA can apply. They are
1. Shared : Multi user can hold various share lock on a single resource
2. Exclusive: It prohibits all sharing of resources i.e. only one use has the sole ability to alter the resources until locks are released.

Syntax:
LOCK TABLE [Table name] IN { ROW SHARE | ROW EXCLUSIVE | SHARE UPDATE | SHARE | SHARE ROW EXCLUSIVE | EXCLUSIVE } MODE [NOWAIT]
ROW SHARE Row share locks all concurrent access to a table.
SHARE UPDATE They prohibit other users to lock entire table exclusively
ROW EXCLUSIVE Row exclusive locks the same as row share locks, but also prohibit locking in share mode. These locks are acquired when updating, inserting or deleting.

SHARE ROW EXCLUSIVE They are used to lock whole table to selective update and to allow other users to lock at row in the table but not lock the table in share mode or to update rows.
NO WAIT Indicates that you do not wish to wait if resources are unavailable.
All locks are released under the following circumstances:
- The transaction is committed successfully
- A rollback is performed
- A rollback to a save point will release locks set afterspecified save point
- Row-level-locks are not released by rolling back to a savepoint
- Data locks are released by log off

Input

Lock table emp in exclusive mode no wait;

Output
Table Locked

SQL PRACTICE QUERIES

1. DISPLAY THE DEPT INFORMATION FROM DEPARTMENT TABLE

```
SQL> SELECT * FROM DEPT;
```

2. DISPLAY THE DETAILS OF ALL EMPLOYEES.

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

3. DISPLAY THE NAME AND JOB FOR ALL EMPLOYEES.

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

4.DISPLAY THE NAME AND SALARY FOR ALL EMPLOYEES.

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

5.DISPLAY EMPLOYEE NUMBER AND TOTAL SALARY FOR EACH EMPLOYEE.

```
SQL>SELECT EMPNO,SAL + NVL(COMM,0) FROM EMP;
```

6.DISPLAY EMPLOYEE NAME AND ANNUAL SALARY FOR ALL EMPLOYEES

```
SQL>  SELECT EMPNO, (SAL + NVL(COMM,0))*12 "ANNUAL SAL" FROM EMP;
```

7. DISPLAY THE NAMES OF ALL EMPLOYEES WHO ARE WORKING IN DEPARTMENT NUMBER 10.

```
SQL > SELECT ENAME FROM EMP WHERE DEPTNO = 10;
```

8.DISPLAY THE NAMES OF ALL EMPLOYEES WHO ARE WORKING AS CLERKS AND DRAWING A SALARY MORE THAN 3000.

```
SQL> SELECT ENAME FROM EMP WHERE JOB = 'CLERK' AND SAL >3000;
```

9. DISPLAY EMPLOYEE NUMBER AND NAMES FOR EMPLOYEES WHO EARN COMMISSION.

```
SQL> SELECT EMPNO,ENAME FROM EMP WHERE COMM IS NOT NULL;
```

10. DISPLAY NAMES OF EMPLOYEES WHO DONOT EARN COMISSION.

SQL> SELECT ENAME FROM EMP WHERE COMM = 0 OR COMM IS NULL;

11.DISPLAY THE NAME OF EMPLOYEES WHO ARE WORKING AS CLERK,SALESMAN OR ANALYST AND DRAWING A SALARY MORE THAN 3000.

SQL> SELECT ENAME FROM EMP WHERE SAL >3000 AND JOB='CLERK' OR JOB = 'SALESMAN' OR JOB = 'ANALYST';

12.DISPLAY THE NAMES OF EMPLOYEES WHO ARE WORKING IN THE COMPANY FOR THE PAST

FIVE YEARS.

SQL> SELECT ENAME FROM EMP WHERE SYSDATE-HIREDATE > 5;

13. DISPLAY THE NAMES OF EMPLOYEES WHO HAVE JOINED THE COMPANY BEFORE 30THJUNE 90 AND AFTER 31ST DEC 90.

SQL> SELECT ENAME FROM EMP WHERE HIREDATE NOT BETWEEN '30-JUNE-1980' AND

'31-DEC-1981'

14.DISPLAY CURRENT DATE.

SQL> SELECT SYSDATE FROM DUAL;

15. DISPLAY THE LIST OF USERS IN YOUR DATABASE.

SQL> SELECT * FROM USER_USERS;

16.DISPLAY THE NAMES OF ALL TABLES FROM THE CURRENT USER.

SQL> SELECT * FROM USER_USERS;

17.DISPLAY THE NAME OF CURRENT USER.

SQL> SHOW USER;

18.DISPLAY THE NAMES OF EMPLOYEES WORKING IN DEPARTMENT NUMBER 10 OR 20 OR 40 OR

EMPLOYEES WORKING AS CLERKS , SALESMAN OR ANALYST.

SQL> SELECT ENAME FROM EMP WHERE DEPTNO = 10 OR DEPTNO =20 OR DEPTNO = 40 AND

JOB = 'CLERK' OR JOB = 'SALESMAN' OR JOB = 'ANALYST';

19.DISPLAY THE NAMES OF EMPLOYEES WHOSE NAMES START WITH ALPHABET S.

SQL> SELECT ENAME FROM EMP WHERE ENAME LIKE 'S%';

20. DISPLAY THE NAMES OF EMPLOYEE WHOSE NAME ENDS WITH ALPHABET S.

SQL> SELECT ENAME FROM EMP WHERE ENAME LIKE '%S';

21. DISPLAY THE NAMES OF EMPLOYEES WHOSE NAMES HAVE SECOND ALPHABET A IN THEIR

NAMES.

SQL> SELECT ENAME FROM EMP WHERE ENAME LIKE '_A%';

22. DISPLAY THE NAMES OF EMPLOYEES WHOSE NAME IS EXACTLY 5 CHARCTERS IN LENGTH.

SQL> SELECT ENAME FROM EMP WHERE LENGTH(ENAME)= 5;

23. DISPLAY THE NAMES OF EMPLOYEES WHO ARE NOT WORKING AS MANAGERS.

SQL> SELECT ENAME FROM EMP WHERE JOB != 'MANAGER';

24. DISPLAY THE NAMES OF EMPLOYEES WHO ARE NOT WORKING AS SALESMAN OR CLERK OR

ANALYST.

SQL> SELECT ENAME FROM EMP WHERE JOB != 'CLERK' OR JOB !='SALESMAN' OR

JOB != 'ANALYST';

25. DISPLAY ALL ROWS FROM EMP TABLE.

SQL> SELECT * FROM EMP;

26.DISPLAY THE TOTAL NUMBER OF EMPLOYEES WORKING IN THE COMPANY.

SQL> SELECT COUNT(*) FROM EMP;

27. DISPLAY THE TOTAL SALARY BEING PAID TO ALL EMPLOYEES.

SQL> SELECT SUM(SAL+NVL(COMM,0)) FROM EMP;

28.DISPLAY THE MAXIMUM SALARY FROM THE EMPLOYEE TABLE.

SQL> SELECT MAX(SAL+NVL(COMM,0)) FROM EMP;

29.DISPLAY THE MINIMUM SALARY FROM EMPLOYEE TABLE.

SQL> SELECT MIN(SAL+NVL(COMM,0)) FROM EMP;

30.DISPLAY THE AVERAGE SALARY FROM EMPLOYEE TABLE.

SQL> SELECT AVG(SAL+NVL(COMM,0)) FROM EMP;

31. DISPLAY THE MAXIMUM SALARY FROM EMPLOYEE TABLE BEING PAID TO CLERK.

SQL> SELECT MAX(SAL+NVL(COMM,0)) FROM EMP WHERE JOB = 'CLERK';

32 DISPLAY THE MAXIMUM SALARY FROM EMPLOYEE TABLE BEING PAID TO
DEPARTMENT NO 20.

SQL> SELECT MAX(SAL+NVL(COMM,0)) FROM EMP WHERE DEPTNO = 20;

33. DISPLAY THE MINIMUM SALARY FROM EMPLOYEE TABLE BEING PAID TO SALESMAN.

SQL> SELECT MIN(SAL+NVL(COMM,0)) FROM EMP WHERE JOB = 'SALESMAN';

34. DISPLAY THE AVERAGE SALARY FROM EMPLOYEE TABLE DRAWN BY MANAGERS.

SQL> SELECT AVG(SAL+NVL(COMM,0)) FROM EMP WHERE JOB = 'MANAGER';

35.DISPLAY THE TOTAL SALARY DRAWN BY ANALYST WORKING IN DEPT NO 30.

SQL> SELECT SUM(SAL+NVL(COMM,0)) FROM EMP WHERE JOB = 'ANALYST' AND DEPTNO = 30;

36. DISPLAY THE NAMES OF EMPLOYEES IN ASCENDING ORDER OF SALARY .

SQL> SELECT ENAME,SAL FROM EMP ORDER BY SAL;

37. DISPLAY THE NAMES OF EMPLOYEES IN DESCENDING ORDER OF SALARY.

SQL> SELECT * FROM EMP ORDER BY SAL DESC

38.DISPLAY THE DETAILS FROM EMP TABLE IN ORDER OF EMP NAME.

SQL> SELECT * FROM EMP ORDER BY ENAME;

39. DISPLAY EMPNO, ENAME, DEPTNO, AND SAL.SORT THE OUTPUT FIRST BASED ON NAME AND WITHIN NAME BY DEPTNO AND WITHIN DEPTNO BY SAL.

SQL> SELECT * FROM EMP ORDER BY ENAME,DEPTNO,SAL;

40. DISPLAY THE NAME OF THE EMPLOYEE ALONG WITH THEIR ANNUAL SALARY (SAL*12). THE NAME OF EMPLOYEE EARNING HIGHEST SALARY SHOULD APPEAR FIRST.

41. DISPLAY NAME,SAL,HRA,PF,DA,TOTALSAL FOR EACH EMPLOYEE.THE OUTPUT SHOULD BE IN THE ORDER OF TOTAL SAL ,HRA 15% OF SAL,DA 10% OF SAL , PF 5% OF SAL, TOTAL SAL WILL BE SAL+HRA+DA-PF.

42 DISPLAY THE DEPT NUMBERS AND TOTAL NUMBER OF EMPLOYEES IN EACH GROUP.

QL> SELECT DEPTNO,COUNT(*) FROM EMP GROUP BY DEPTNO;

43. DISPLAY VARIOUS JOBS AND TOTAL NUMBER OF EMPLOYEES WITHIN EACH JOB GROUP.

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

44. DISPLAY DEPT NUMBERS AND TOTAL SALARY FOR EACH DEPARTMENT.

SQL> SELECT DEPTNO,SUM(SAL+NVL(COMM,0)) FROM EMP GROUP BY DEPTNO;

45. DISPLAY DEPT NUMBERS AND MAXIMUM SALARY FOR EACH DEPARTMENT.

SQL> SELECT DEPTNO,MAX(SAL+NVL(COMM,0)) FROM EMP GROUP BY DEPTNO;

46. DISPLAY VARIOUS JOBS AND TOTAL SALARY FOR EACH JOB.

SQL> SELECT DISTINCT JOB,SUM(SAL+NVL(COMM,0)) FROM EMP GROUP BY JOB;

47.DISPLAY EACH JOB ALONG WITH MINIMUM SALARY BEING PAID IN EACH JOB GROUP.

SQL> SELECT DISTINCT JOB,MIN(SAL+NVL(COMM,0)) FROM EMP GROUP BY JOB;

48. DISPLAY THE DEPARTMENT NUMBERS WITH MORE THAN THREE EMPLOYEES IN EACH DEPT.

SQL> SELECT DEPTNO,COUNT(*) FROM EMP GROUP BY DEPTNO HAVING COUNT(*)>3;

49. DISPLAY THE VARIOUS JOBS ALONG WITH TOTAL SAL FOR EACH OF THE JOBS WHERE

TOTAL SAL > 40000.

SQL> SELECT DISTINCT(JOB),SUM(SAL+NVL(COMM,0)) FROM EMP GROUP BY JOB HAVING

SUM(SAL+NVL(COMM,0)) > 40000;

50. DISPLAY THE VARIOUS JOBS ALONG WITH TOTAL NUMBER OF EMPLOYEES IN EACH JOB.

SQL> SELECT DISTINCT(JOB),COUNT(*) FROM EMP GROUP BY JOB HAVING COUNT(*) >3;

51. DISPLAY THE NAME OF EMP WHO EARNS HIGHEST SAL.

SQL> SELECT ENAME FROM EMP WHERE SAL =(SELECT MAX(SAL) FROM EMP);

52. DISPLAY THE EMPLOYEE NUMBER AND NAME OF EMPLOYEE WORKING AS CLERK

AND EARNING HIGHEST SALARY AMONG CLERKS.

SQL> SELECT ENAME,EMPNO FROM EMP WHERE JOB = 'CLERK' AND SAL =

(SELECT MAX(SAL) FROM EMP WHERE JOB = 'CLERK');

53. DISPLAY THE NAMES OF THE SALESMAN WHO EARNS A SALARY MORE THAN THE HIGHES SALARY OF ANY CLERK.

SQL> SELECT ENAME FROM EMP WHERE JOB = 'SALESMAN' AND SAL>

(SELECT MAX(SAL) FROM EMP WHERE JOB = 'CLERK');

54. DISPLAY THE NAMES OF CLERKS WHO EARN SALARY MORE THAN THAT OF JAMES AND LESS THAN THAT OF SCOTT.

```
SQL> SELECT ENAME FROM EMP WHERE JOB = 'CLERK' AND
          SAL<(SELECT SAL FROM EMP WHERE ENAME LIKE 'SCOTT')
           AND SAL >(SELECT SAL FROM EMP WHERE ENAME LIKE 'JAMES');
```

55.DISPLAY THE NAMES OF EMPLOYEES WHO EARN A SAL MORE THAN THAT OF JAMES OR THAT

SALARY GREATER THAN THAT OF SCOTT.

SQL> SELECT ENAMEFROM EMP WHERE SAL>(SELECT SAL FROM EMP WHERE ENAME LIKE

'SCOTT');

56. DISPLAY THE NAMES OF THE EMPLOYEES WHO EARN HIGHEST SALARY IN THEIR RESPECTIVE DEPARTMENTS.

SQL> SELECT E.ENAME,E.SAL FROM EMP E WHERE E.SAL = (SELECT MAX(F.SAL) FROM EMP F GROUP BY F.DEPTNO HAVING E.DEPTNO = F.DEPTNO);

57. DISPLAY THE EMPLOYEE NAMES WHO ARE WORKING IN ACCOUNTING DEPT.

SQL> SELECT ENAME FROM EMP E,DEPT D WHERE E.DEPTNO = D.DEPTNO AND

DNAME= 'ACCOUNTING';

58. DISPLAY THE EMPLOYEE NAMES WHO ARE WORKING IN CHICAGO.

SQL> SELECT ENAME FROM EMP E, DEPT D WHERE E.DEPTNO = D.DEPTNO AND

LOC = 'CHICAGO';

59. DISPLAY THE JOB GROUPS HAVING TOTAL SALARY GREATER THAN THE MAXIMUM SALARY

FOR MANAGERS.

SQL> SELECT DISTINCT(JOB),SUM(SAL+NVL(COMM,0)) FROM EMP GROUP BY JOB

HAVING SUM(SAL+NVL(COMM,0)) > (SELECT MAX(SAL) FROM EMP WHERE JOB = 'MANAGER');

60. DISPLAY THE NAMES OF EMPLOYEES FROM DEPARTMENT NUMBER 10 WITH SALARY

GREATER THAN THAT OF ANY EMPLOYEE WORKING IN OTHER DEPARTMENTS.

SQL> SELECT ENAME FROM EMP WHERE DEPTNO = 10 AND

SAL > ANY(SELECT SAL FROM EMP WHERE DEPTNO != 10);

61. DISPLAY THE NAMES OF EMPLOYEE FROM DEPARTMENT NUMBER 10 WITH SALARY

GREATER THAN THAT OF ALL EMPLOYEE WORKING IN OTHER DEPARTMENTS.

SQL> SELECT ENAME FROM EMP WHERE DEPTNO = 10 AND

SAL >ALL(SELECT SAL FROM EMP WHERE DEPTNO != 10);

62. DISPLAY THE NAMES OF EMPLOYEES IN UPPER CASE.

SQL> SELECT UPPER(ENAME) FROM EMP;

63.DISPLAY THE NAMES OF EMPLOYEES IN LOWER CASE.

SQL> SELECT LOWER(ENAME) FROM EMP;

64. DISPLAY THE NAMES OF EMPLOYEES IN PROPER CASE.

SQL> SELECT INITCAP(ENAME) FROM EMP;

65.FIND OUT THE LENGTH OF YOUR NAME USING APPROPRIATE FUNCTION.

SQL> SELECT LENGTH('&NAME') FROM DUAL;

66. DISPLAY THE LENGTH OF ALL EMPLOYEES' NAMES.

SQL> SELECT LENGTH(ENAME) FROM EMP;

67. DISPLAY THE NAME OF THE EMPLOYEE CONCATENATE WITH EMPNO.

SQL> SELECT CONCAT(ENAME,EMPNO) FROM EMP;

68. USE APPROPRIATE FUNCTION AND EXTRACT 3 CHARACTERS STARTING FROM 2 CHARACTERS

FROM THE FOLLOWING STRING 'ORACLE' I.E THE OUTPUT SHOULD BE 'RAC'.

SQL> SELECT SUBSTR('ORACLE',2,3) FROM DUAL;

69. FIND THE FIRST OCCURENCE OF CHARACTER A FROM THE FOLLOWING STRING 'COMPUTER

MAINTAINENCE CORPORATION'

SQL> SELECT INSTR('COMPUTER MAINTENANCE CORPORATION','A') FROM DUAL;

70. REPLACE EVERY OCCURENCE OF ALPHABET A WITH B IN THE STRING ALLEN'S.

SQL> SELECT REPLACE('ALLEN','A','B') FROM DUAL;

71. DISPLAY THE INFORMATION FROM EMP TABLE.WHEREVER JOB 'MANAGER' IS FOUND ., IT SHOULD BE DISPLAYED AS BOSS.

```
SQL> SELECT JOB,DECODE(JOB,'MANAGER','BOSS',' ') FROM EMP;
```

72. DISPLAY EMPNO,ENAME,DEPTNO FROM EMP TABLE.INSTEAD OF DISPLAY DEPARTMENT

NUMBERS DISPLAY THE RELATED DEPARTMENT NAME.

```
SQL> SELECT E.EMPNO, E.ENAME, D.DNAME
  2  FROM EMP E, DEPT D
  3  WHERE E.DEPTNO = D.DEPTNO;
```

73. DISPLAY YOUR AGE IN DAYS.

```
SQL> SELECT MONTHS_BETWEEN('22-FEB-87',SYSDATE)*30 FROM DUAL;
```

74. DISPLAY YOUR AGE IN MONTHS.

```
SQL> SELECT MONTHS_BETWEEN('22-FEB-87',SYSDATE) FROM DUAL;
```

75. DISPLAY CURRENT DATE AS 15TH AUGUST FRIDAY NINETEEN FORTY SEVEN.

```
SQL> SELECT TO_CHAR(SYSDATE,'DDTH MONTH YEAR') FROM DUAL;
```

76. DISPLAY THE FOLLOWING OUTPUT FOR EACH ROW FROM EMP TABLE AS 'SCOTT HAS JOINED

THE COMPANY ON WEDNESDAY 13TH AUGUST NINETEEN NINETY'.

```
SQL> SELECT ENAME || 'HAS JOINED THE COMPANY ON'
          || TO_CHAR(HIREDATE,'DAY DDTH MONTH YEAR') FROM EMP;
```

77. DISPLAY THE COMMON JOBS FROM DEPARTMENT NUMBER 10 AND 20.

```
SQL> SELECT JOB FROM EMP WHERE DEPTNO = 10 AND DEPTNO =20;
```

78. DISPLAY THE JOBS FOUND IN DEPARTMENT NUMBER 10 AND 20 ELIMINATE DUPLICATE JOBS.

```
SQL> SELECT DISTINCT(JOB) FROM EMP WHERE DEPTNO = 10 AND DEPTNO =20;
```

79. DISPLAY THE DETAILS OF EMPLOYEES WHO ARE IN SALES DEPT AND GRADE IS 3.

```
SQL> SELECT E.ENAME,D.DNAME,S.GRADE FROM EMP E,DEPT D,SALGRADE S
```

WHERE E.DEPTNO = D.DEPTNO AND D.DNAME = 'SALES' AND S.GRADE = 3;

80. DISPLAY THOSE EMPLOYEES WHOSE NAME CONTAINS NOT LESS THAN 4 CHARS.

SQL> SELECT ENAME FROM EMP WHERE LENGTH(ENAME)>3;

81. DISPLAY THOSE DEPARTMENTS WHOSE NAME START WITH 'S' WHILE LOCATION NAME

END WITH 'O';

SQL> SELECT DNAME FROM DEPT WHERE DNAME LIKE 'S%' AND LOC LIKE '%O';

82. DISPLAY THOSE EMPLOYEES WHOSE MANAGER NAME IS JONES,AND ALSO DISPLAY THERE MANAGER NAME

SQL> SELECT E.ENAME EMPLOYEE ,M.ENAME MANAGER FROM EMP E,EMP M WHERE E.MGR=M.EMPNO AND M.ENAME='JONES';

83)DISPLAY NAME AND SALARY OF FORD IF HIS SAL IS EQUAL TO HIGH SAL OF HIS GRADE.

SQL>SELECT E.ENAME,E.SAL FROM EMP E,SALGRADE S WHERE E.ENAME='FORD' AND E.SAL=S.HISAL;

84)DISPLAY EMPLOYEE NAME,HIS JOB,HIS DEPT NAME,HIS MANAGER NAME,HIS GRADE AND MAKE OUT OF AN UNDER DEPT WISE.

SQL>SELECT E.ENAME EMPLOYEE,M.ENAME MANAGER,E.JOB,D.DNAME,S.GRADE

FROM EMP E,DEPT D,SALGRADE S,EMP M

GROUP BY M.ENAME,E.MGR,M.EMPNO,E.SAL,E.ENAME,E.JOB,D.DEPTNO,D.DNAME,S.GRADE,S.LOSAL,S.HISAL,E.DEPTNO HAVING E.MGR=M.EMPNO AND

E.DEPTNO=D.DEPTNO AND E.SAL BETWEEN S.LOSAL AND S.HISAL

85)LIST OUT ALL THE EMPLOYEES NAME,HIS JOB, HIS DEPT NAME AND SALARY GRADE FOR EVERY ONE IN THE COMPANY EXCEPT 'CLERK'.SORT ON SALARY.

SQL>SELECT E.ENAME EMPLOYEE,M.ENAME MANAGER,S.GRADE,D.DNAME FROM EMP E,EMP M,SALGRADE S,DEPT D

WHERE E.MGR=M.EMPNO AND E.ENAME!='CLARK' AND D.DEPTNO=E.DEPTNO AND E.SAL BETWEEN S.LOSAL AND S.HISAL ORDER BY E.SAL DESC ;

86)DISPLAY THE NAME OF THE EMPLOYEE WHO IS GETTING MAXIMUM SALSRY.

SQL> SELECT * FROM EMP WHERE SAL=(SELECT MAX(SAL) FROM EMP);

87)FIND OUT THE TOP FIVE EARNERS OF THE COMPANY.

SQL>SELECT * FROM EMP E WHERE 5>(SELECT COUNT(*) FROM EMP M WHERE E.SAL<M.SAL);

88)DISPLAY THOSE EMPLOYEES WHOSE SALARY IS EQUAL TO AVERAGE OF MAXIMUM AND MINIMUM SALARY.

SQL> SELECT * FROM EMP WHERE SAL IN (SELECT (MAX(SAL)+MIN(SAL))/2 FROM EMP);

89)DISPLAY COUNT OF EMPLOYEES IN EACH DEPARTMENT WHERE COUNT GREATER THAN OR EQUAL TO 3.

SQL> SELECT COUNT(*) FROM EMP WHERE 3<(SELECT COUNT(*) FROM EMP) GROUP BY DEPTNO;

90)DISPLAY DNAME WHERE ATLEAST 3 ARE WORKING AND DISPLAY ONLY DNAME.

SQL> SELECT D.DNAME FROM EMP E,DEPT D WHERE 3<(SELECT COUNT(*) FROM EMP) AND D.DEPTNO=E.DEPTNO GROUP BY D.DNAME,E.DEPTNO,D.DEPTNO;

91)DISPLAY NAME OF THOSE MANAGERS NAME WHOSE SALARY IS MORE THAN AVERAGE SALARY OF HIS EMPLOYEES.

SQL>SELECT E.ENAME FROM EMP E,EMP M WHERE E.SAL IN(SELECT AVG(SAL) FROM EMP) AND E.MGR=M.EMPNO GROUP BY E.MGR,E.ENAME,E.SAL,M.EMPNO

92)DISPLAY EMPLOYEE NAME,SAL,COMM,NET PAY FOR THOSE EMPLOYEES WHOSE NET PAY ARE GREATER THAN OR EQUAL TO ANY OTHER EMPLOYEE SALARY OF THE COMPANY.

SQL> SELECT ENAME,SAL,COMM,SAL+NVL(COMM,0) NET_PAY FROM EMP E

WHERE SAL+NVL(COMM,0)>ANY(SELECT SAL+NVL(COMM,0) FROM EMP M WHERE E.DEPTNO!=M.DEPTNO)

93)DISPLAY THOSE EMPLOYEES WHOSE SALARY IS LESS THAN HIS MANAGER BUT MORE THAN SALARY OF ANY OTHER MANAGER.

SQL>SELECT E.ENAME,E.SAL FROM EMP E WHERE E.SAL<(SELECT N.SAL FROM EMP N WHERE E.MGR=N.EMPNO) AND SAL>ANY(SELECT M.SAL FROM EMP M WHERE E.MGR!=M.EMPNO)

94)FIND OUT THE LEAST FIVE EARNERS.

SQL> SELECT * FROM EMP E WHERE 5>(SELECT COUNT(*) FROM EMP M WHERE E.SAL>M.SAL);

95)FIND OUT THE NO.OF EMPLOYEES WHOSE SALARY IS GREATER THAN THERE MANAGERS SALARY.

SQL> SELECT COUNT(*) FROM EMP E,EMP M WHERE E.MGR=M.EMPNO AND E.SAL>M.SAL;

96)DISPLAY THOSE MANAGERS WHO ARE NOT WORKING UNDER PRESIDENT BUT THEY ARE WORKING UNDER ANY OTHER MANAGER.

SQL>SELECT E.ENAME FROM EMP E WHERE MGR IN(SELECT M.EMPNO FROM EMP M WHERE M.ENAME!='KING' AND E.MGR=M.EMPNO);

97)DELETE THOSE DEPARTMENT WHERE NO EMPLOYEE WORKING.

SQL> DELETE EMP WHERE DEPTNO=(SELECT DEPTNO FROM EMP GROUP BY DEPTNO HAVING COUNT(DEPTNO)=0);

98)DELETE THOSE RECORDS FROM EMP TABLE WHOSE DEPTNO NOT AVAILABLE IN DEPT TABLE.

SQL> DELETE EMP WHERE DEPTNO NOT IN(SELECT DEPTNO FROM DEPT);

99)DISPLAY THOSE EARNERS WHOSE SALARY IS OUT OF THE GRADE AVAILABLE IN SALGRADE TABLE.

SQL> SELECT E.ENAME,E.SAL FROM EMP E,SALGRADE S WHERE E.SAL<LOSAL AND E.SAL>HISAL;

100) DISPLAY EMPLOYEE NAME,SAL,COMM WHOSE NET PAY IS GREATER THAN ANY OTHER IN TH COMPANY.

SQL> SELECT ENAME,SAL,COMM FROM EMP WHERE SAL+NVL(COMM,0)=(SELECT MAX(SAL+NVL(COMM,0)) FROM EMP);

101)DISPLAY THOSE EMPLOYEES WHO ARE GOING TO RETIRE ON 31-DEC-99.IF THE MAX JOB PERIOD IS 18 YEARS.

SQL>SELECT * FROM EMP WHERE MONTHS_BETWEEN('31-DEC-99',HIREDATE)>18*12;

102)DISPLAY THOSE EMPLOYEES WHOSE SALARY IS ODD VALUE.

SQL> SELECT * FROM EMP WHERE MOD(SAL,2)=1;

103)DISPLAY THOSE EMPLOYEES WHOSE SALARY CONTAINS ATLEAST 4 DIGITS.

```
SQL> SELECT * FROM EMP WHERE LENGTH(SAL)>=4;
```

104)DISPLAY THOSE EMPLOYEES WHOSE NAME CONTAINS A IN THEM.

```
SQL> SELECT * FROM EMP WHERE ENAME LIKE '%A%';
```

105)DISPLAY THOSE EMPLOYEES ,WHOSE FIRST 2 CHARACTERS FROM HIREDATE IS EQUAL TO LAST 2 CHARACTERS FROM SALARY.

```
SQL> SELECT * FROM EMP WHERE SUBSTR(HIREDATE,1,2) IN (SELECT
SUBSTR(SAL,LENGTH(SAL)-2,2) FROM EMP);
```

106)DISPLAY THOSE EMPLOYEES WHOSE 10% SALARY IS EQUAL TO THE YEAR OF JOINING.

```
SQL> SELECT * FROM EMP WHERE 0.1*SAL IN (SELECT
TO_NUMBER(TO_CHAR(HIREDATE,'YY')) FROM EMP);
```

107)DISPLAY THOSE EMPLOYEES WHO ARE WORKING IN SALES OR RESEARCH DEPT.

```
SQL> SELECT * FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE
DNAME IN('SALES','RESEARCH'));
```

108)DISPLAY THE GRADE OF JONES.

```
SQL>SELECT E.ENAME,E.SAL,S.GRADE FROM EMP E,SALGRADE S
  2  WHERE E.ENAME='JONES' AND E.SAL BETWEEN S.LOSAL AND S.HISAL;
```

109)DISPLAY THOSE EMPLOYEES WHO JOINED IN THE COMPANY BEFORE 15th OF THE MONTH.

```
SQL>SELECT * FROM EMP WHERE TO_NUMBER(TO_CHAR(HIREDATE,'DD'))<15;
```

110)DELETE THOSE EMPLOYEES WHERE NO.OF EMPLOYEES IN A PARTICULAR DEPT IS LESS THAN 3.

```
SQL> DELETE EMP WHERE 3>ANY(SELECT COUNT(*) FROM EMP GROUP BY DEPTNO);
```

111)DELETE THOSE EMPLOYEES WHO JOINED IN THE COMPANY 21 YEARS BACK FROM TODAY.

```
SQL> DELETE EMP WHERE MONTHS_BETWEEN(SYSDATE,HIREDATE)=21*12;
```

112)DISPLAY THE DEPARTMENT NAME WHERE THE NO.OF CHARACTERS OF WHICH IS EQUAL TO

THE NO. OF EMPLOYEES IN ANY OTHER DEPARTMENT.

SQL>SELECT DNAME FROM DEPT WHERE LENGTH(DNAME) IN (SELECT COUNT(*) FROM EMP GROUP BY DEPTNO);

113)DISPLAY THOSE EMPLOYEES WHO ARE WORKING AS MANAGERS.

SQL> SELECT * FROM EMP WHERE JOB='MANAGER';

114)COUNT THE NO.OF EMPLOYEES WHO ARE WORKING AS MANAGERS.

SQL> SELECT COUNT(*) FROM EMP GROUP BY JOB HAVING JOB='MANAGER';

115)DISPLAY THE NAMES OF THE DEPARTMENT THOSE EMPLOYEES WHO JOINED THE COMPANY

ON THE SAME DAY.

SQL>SELECT DNAME FROM DEPT WHERE DEPTNO IN(SELECT E.DEPTNO FROM EMP E

WHERE E.HIREDATE IN(SELECT M.HIREDATE FROM EMP M WHERE E.EMPNO!=M.EMPNO));

116)DISPLAY THE MANAGER WHO IS HAVING MAXIMUM NO.OF EMPLOYEES WORKING UNDER HIM.

SQL> SELECT ENAME FROM EMP WHERE EMPNO=(SELECT MGR FROM EMP GROUP BY

MGR HAVING COUNT(MGR)=(SELECT MAX(COUNT(MGR)) FROM EMP GROUP BY MGR));

117)LIST OUT THE EMPLOYEES NAME AND SALARY INCREASED BY 15% AND EXPRESSED

AS WHOLE NUMBER OF DOLLAR.

SQL>SELECT ENAME,ROUND(1.15*SAL/48,2)||'$' DOLLAR FROM EMP;

118)LIST ALL THE EMPLOYEES WITH HIREDATE IN THE FORMAT 'JUNE 04 1988'.

SQL> SELECT ENAME,TO_CHAR(HIREDATE,'MONTH DD YYYY') HIREDATE FROM EMP;

119)PRINT A LIST OF EMPLOYEES DISPLAYING 'LESS SALARY' IF SAL<1500.IF EXACTLY 1500

DISPLAY AS 'EXACT SALARY' AND IF GREATER THAN 1500 DISPLAY AS 'MORE SALARY'.

```
SQL>  SELECT ENAME,SAL||' LESS SALARY' FROM EMP WHERE SAL<1500
  UNION
  SELECT ENAME,SAL||' EXACT SALARY' FROM EMP WHERE SAL=1500
  UNION
  SELECT ENAME,SAL||' MORE SALARY' FROM EMP WHERE SAL>1500;
```

120)WRITE A QUERY TO CALCULATE THE LENGTH OF EMPLOYEE HAS BEEN WITH THE COMPANY.

```
SQL> SELECT ENAME,LENGTH(ENAME) LENGTH FROM EMP;
```

121)DISPLAY THOSE MANAGERS WHO ARE GETTING SALARY LESS THAN HIS EMPLOYEE.

```
SQL>SELECT E.ENAME,E.SAL FROM EMP E WHERE E.EMPNO IN
(SELECT M.MGR FROM EMP M WHERE M.MGR=E.EMPNO AND E.SAL<M.SAL)
```

122)PRINT THE DETAILS OF ALL THE EMPLOYEES WHO ARE SUB ORDINATES TO BLAKE.

```
SQL> SELECT * FROM EMP WHERE MGR=(SELECT EMPNO FROM EMP WHERE
ENAME='BLAKE');
```

123) DISPLAY THOSE EMPLOYEES WHOSE MANAGER NAME IS JONES AND ALSO WITH HIS

MANAGER NAME.

```
SQL> SELECT E.ENAME EMPLOYEE,M.ENAME MANAGER FROM EMP E,EMP M WHERE
M.EMPNO=E.MGR AND M.ENAME='JONES';
```

124)DEFINE VARIABLE REPRESENTING THE EXPRESSIONS USED TO CALCULATE ON EMPLOYEES

TOTAL ANNUAL REMUNERATION.

```
SQL> DEFINE REM=(SAL+NVL(COMM,0))*12 ON EMP;
SQL> SELECT ENAME,&REM FROM EMP;
```

125)FIND OUT THE AVG SAL AND AVG TOTAL REMUNERATION FOR EACH JOB TYPE.

```
SQL> SELECT AVG(SAL),AVG(SAL+NVL(COMM,0)) FROM EMP GROUP BY JOB;
```

126)LIST ENAME,JOB ANNUAL SAL,DEPTNO,DNAME AND GRADE WHO EARN MORE THAN

30000 PER YEAR AND WHO ARE NOT CLERKS.

```
SQL> SELECT E.ENAME,E.JOB,E.SAL*12 ANN_SAL,S.GRADE,D.DNAME FROM EMP E,SALGRADE S,DEPT D
         WHERE E.SAL*12>30000 AND E.DEPTNO=D.DEPTNO AND E.SAL BETWEEN S.LOSAL AND S.HISAL
          AND E.JOB!='CLERK';
```

127)FIND OUT THE JOB THAT WAS FILLED IN THE FIRST HALF OF 1983 AND THE SAME

JOB THAT WAS FILLED DURING THE SAME PERIOD ON 1984.

```
SQL> SELECT JOB FROM EMP WHERE TO_NUMBER(TO_CHAR(HIREDATE,'MM'))<6 AND
TO_NUMBER(TO_CHAR(HIREDATE,'YY'))=83
INTERSECT
SELECT JOB FROM EMP WHERE TO_NUMBER(TO_CHAR(HIREDATE,'MM'))<6
AND TO_NUMBER(TO_CHAR(HIREDATE,'YY'))=84;
```

128)LIST OUT THE LOWEST PAID EMPLOYEES WORKING FOR EACH MANAGER,EXCLUDE ANY

GROUPS WHERE MIN SAL IS LESS THAN 1000 SORT THE OUTPUT BY SAL.

```
SQL> SELECT E.ENAME,E.MGR,E.SAL FROM EMP E WHERE SAL IN
 (SELECT MIN(SAL) FROM EMP WHERE MGR=E.MGR) AND E.SAL>1000 ORDER BY SAL;
```

129)CHECK WHETHER ALL THE EMPLOYEE NO'S ARE INDEED UNIQUE.

```
SQL> SELECT COUNT(EMPNO),COUNT(DISTINCT(EMPNO)) FROM EMP
      HAVING COUNT(EMPNO)=COUNT(DISTINCT(EMPNO));
```

130.FIND OUT THE EMPLOYEES BY NAME AND NUMBER ALONG WITH THIER MANAGER'S NAME AND NUMBER ALSO DISPLAY NOMANAGER WHO HAS NOMANAGER .

```
SQL>SELECT E.EMPNO,E.ENAME,M.EMPNO,'MANAGER',M.ENAME MANAGERNAME FROM EMP E,EMP M WHERE E.MGR=M.EMPNO UNION SELECT T.EMPNO,T.ENAME,T.MGR,'NOMANAGER',X.ENAME FROM EMP T,EMP X WHERE T.MGR IS NULL AND T.EMPNO=X.EMPNO
```

```
SQL>SELECT * FROM EMP E WHERE SAL =(SELECT MIN(SAL) FROM EMP WHERE JOB=E.JOB);
```

132.FIND OUT MOST RECENTLY HIRED EMPLOYEE IN EACH DEPT

SQL>SELECT E.ENAME,E.DEPTNO FROM EMP E

WHERE E.HIREDATE=(SELECT MAX(HIREDATE) FROM EMP WHERE DEPTNO=E.DEPTNO)

133. DISPLAY ENAME AND SALARY FOR EACH EMPLOYEE WHO EARN A SALARY > THE

AVERAGE OF THEIR DEPARTMENT

SQL>SELECT E.ENAME,E.SAL,E.DEPTNO FROM EMP E WHERE

E.SAL > ANY(SELECT AVG(F.SAL) FROM EMP F WHERE E.DEPTNO=F.DEPTNO)ORDER BY E.DEPTNO

134. SELECT THE DEPARTMENT NUMBER WHERE THERE ARE NO EMPLOYEES

SQL>SELECT D.DEPTNO FROM DEPT D

WHERE (SELECT COUNT(*) FROM EMP E WHERE E.DEPTNO=D.DEPTNO) = 0;

135.IN WHICH YEAR DID MOST PEOPLE JOIN THE COMPANY .DISPLAY THE YEAR AND

NUMBER OF EMPLOYEES.

SQL>SELECT TO_CHAR(HIREDATE,'YYYY'),COUNT(*) FROM EMP

GROUP BY TO_CHAR(HIREDATE,'YYYY') HAVING COUNT(*)=(SELECT MAX(COUNT(*)) FROM EMP

GROUP BY TO_CHAR(HIREDATE,'YYYY'))

136. DISPLAY AVERAGE SAL FIGURE FOR THE DIPARTMENT

SQL>SELECT AVG(SAL) FROM EMP GROUP BY DEPTNO

137. EMPLOYEES WHO EARN MORE THAN LOWEST SAL IN DEPT NO 30

SQL> SELECT * FROM EMP WHERE SAL>(SELECT MIN(SAL) FROM EMP WHERE DEPTNO=30)

138. EMPLOYEES WHO EARN MORE THAN EVERY EMPLOYEE IN DEPT NO 30

SQL> SELECT * FROM EMP WHERE SAL>(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=30);

139. SELECT DEPT NAME ,DEPT NUMBER & SUM OF SALARIES.

```
SQL>  SELECT D.DNAME,D.DEPTNO,SUM(E.SAL) FROM EMP E,DEPT D
 GROUP BY D.DEPTNO,D.DNAME,E.DEPTNO
 HAVING D.DEPTNO=E.DEPTNO
```

140. FIND ALL DEPT'S WHICH HAVE MORE THAN 3 EMPLOYEES.

```
SQL> SELECT D.DNAME FROM DEPT D WHERE
3<ANY(SELECT COUNT(*) FROM EMP GROUP BY DEPTNO HAVING D.DEPTNO
```

141.DISPLAY HALF OF ENAME IN UPPER CASE AND HALF IN LOWER

```
SQL> SELECT
CONCAT(LOWER(SUBSTR(ENAME,1,LENGTH(ENAME)/2)),UPPER(SUBSTR(ENAME,LENG
TH(ENAME)/2+1,LENGT
H(ENAME)))) FROM EMP;
```

142.FIND OUT MOST RECENTLY HIRED EMPLOYEE IN EACH DEPT

```
SQL>SELECT E.ENAME,E.DEPTNO FROM EMP E
WHERE E.HIREDATE=(SELECT MAX(HIREDATE) FROM EMP WHERE
DEPTNO=E.DEPTNO)
```

143. CREATE A VIEW OF EMP TABLE

```
SQL> CREATE VIEW EMP2 AS SELECT * FROM EMP;
SQL> SELECT * FROM EMP2;
```

144.SELECT ENAME IF ENAME EXISTS MORE THAN ONCE

```
SQL> SELECT E.ENAME FROM EMP E WHERE 1<ANY (SELECT COUNT(F.ENAME) FROM
EMP  F
      GROUP BY  F.ENAME HAVING E.ENAME=F.ENAME )
```

145.DISPLAY ALL ENAMES IN REVERSE ORDER

```
   SELECT REVERSE(ENAME) FROM EMP
```

146.DISPLAY THOSE EMPLOYEES WHOOSE JOINED MONTH AND DATE ARE EQUAL

```
SQL> SELECT ENAME FROM EMP ,SALGRADE S WHERE
TO_CHAR(HIREDATE,'MM')=S.GRADE;
```

147.DISPLAY THOSE EMPLOYEES WHOOSE JOINING DATE IS AVAILABLE IN DEPT NO.

SQL> SELECT ENAME FROM EMP ,DEPT WHERE
TO_CHAR(HIREDATE,'DD')=DEPT.DEPTNO;

148.DISPLAY THE EMPLOYEES NAMES AS FOLLOWS .

A ALLEN,B BLAKE.

SQL> SELECT CONCAT(SUBSTR(ENAME,1,1),CONCAT(' ',ENAME)) FROM EMP

149.LIST OF EMPLOYEE NAME ,SAL ,PF FROM EMP

SQL>SELECT ENAME,SAL,&PF_AS_PERCENT_OF_SAL*SAL/100 "PF" FROM EMP

150.FIND OUT MOST RECENTLY HIRED EMPLOYEE IN EACH DEPT

SQL>SELECT ENAME,HIREDATE FROM EMP WHERE HIREDATE>

ANY(SELECT HIREDATE FROM EMP) GROUP BY HIREDATE,ENAME;

151)DISPLAY EMPLOYEE NAME,HIS JOB, HIS MANAGER.DISPLAY ALSO EMPLOYEES WHO ARE WITHOUT MANAGER.

SQL> SELECT E.ENAME,E.JOB,M.ENAME FROM EMP E,EMP M WHERE E.MGR=M.EMPNO OR E.MGR IS NULL;

152)DISPLAY NAMES OF THOSE MANAGERS WHOSE SALARY IS MAOR THAN AVERAGE SALARY OF THE COMPANY.

SQL>SELECT M.ENAME FROM EMP E,EMP M WHERE M.SAL IN(SELECT AVG(SAL) FROM EMP)

AND E.MGR=M.EMPNO

Yerragudipadu Subbarayudu
B.Tech,M.Tech,MISTE, IAENG,(Ph.D)
Department of IT/CSE.
Cell No: +91- 9059930490