

A PRACTICAL FILE ON LAB ON ORACLE

SUBMITTED TO:

Mr. AJAY KUMAR

ASSISTANT PROFESSOR

SUBMITTED BY:

DHRUV GURSAHANI

0191BCA011

SHIFT: MORNING

SEMESTER: 3rd



BCA

(2019-2022)

INDEX

S.NO	ASSIGNMENT	SIGN	REMARKS
1	Display all the bill details of "24-jan-12" Display the bill details where total amount is greater than 6000. Display the bill details where total amount is less than 500.		
2	Add a column „number_of_items“ to BILL table. Rename the table „BILL“ to „BILL_DETAILS“. Update the bill amount of 24-JAN-12 Display the bill no. and total amount multiplied by 3.		
3	Display the owners who does not own a „Maruti“ Display the car details of car make „Hyundai“ Display model and owners from CAR Display the car details whose car colour is „RED“		
4	Rename the table „CAR“ to „CAR INFO“ Delete all the details of car „Hyundai“ Update the car color from „RED“ to „METALLIC RED“ Display the car details in ascending or descending order of colour.		
5	List all information about all employees. Display the name of all employees with their salary. List all employees name who is working with department no 20. List the name of all Salesman and Analyst. Display the details of those employees who have joined before the end of September 1981. List the employee's name, employee id who are Manager. List the name and job of all employees who are not clerk. List the name of employees whose employees no is 7369 ,752,7839, 7934 or 7788 List the employees details who does not belongs to department no 10 and 30 List the employees name and salary whose salary varies from 1000 to 2000. List employees name who have joined before 30 June-1981 and after 30 Dec-1981 List the Commission and name of employees who are availing the commission. List the name and designation of employee who does report any body List the details of employees whose salary is greater than 2000 and commission is Null. List the employees details whose name start with 'S'. List the employees name having 'I' as second character along with their Job. List the employees name and date of joining in the descending order of date of joining the column title should be 'Date of Joining'. List the employee name, salary, job, and department number and display it in ascending order of department no and descending order of salary. List the department no and total no of employees in each department. List the different job names available in emp table. List department no and total salary payable in each department. List the average salary and number of employees working in department no 20. List the job and number of employees in each job the result should be in descending order of number of employees. List the total salary, maximum salary, minimum salary and average salary of employee's job wise for department no 10 only. List of the average salary of each job excluding manager.		

	<p>List the average salary of each job within each department.</p> <p>List the average salary of all departments in which more than 5 people working.</p> <p>List the jobs of all employees where maximum salary is greater than equal to 5000.</p> <p>List the total salary average salary of the employees job wise for department no 20 and display only those rows having average salary greater than 1000.</p> <p>Retrieve records from any two tables without using any condition.</p> <p>Display the number of employees working in each department.</p> <p>Display the department information also even if no employee working to that department.</p> <p>List the name of manager along with employee's name, salary and manager code.</p> <p>List the employee's details who have joined the company before their manager.</p> <p>Display different job available in the department no 20 and 30 using set operator.</p> <p>Display all jobs commission in department no 20 and 30 by using set operator.</p> <p>Select all jobs unique in department no 20 by using set operator.</p> <p>List the employees name department no and department name from emp and dept table based on department number make sure that only those record select whose designation is manager.</p> <p>Display the entire department no from emp and dept table without eliminating the duplicate data.</p> <p>Fetch the employees no and employees name from emp table where department no is 10 and 30 and order the based-on employee name using set operator.</p>		
6	Write PL/SQL code to print the table of the number inserted by user.		
7	Write PL/SQL code to print factorial of the number entered by the user.		
8	Write a PL/SQL code to print Fibonacci series till the number entered by the user.		
9	Write a PL/SQL code to check that that the number entered by the user is prime or not.		
10	Write PL/SQL code to add two numbers.		
11	Write PL/SQL code to find that the number entered by user is even or odd.		
12	Write a code in PL/SQL using cursor to display the record of employees earning salary more than 3000.		
13	Create a trigger that restricts the insertion and updation of records having no_of_days less than zero.		
14	Write a procedure that will display doctor details who have treated diabetes patient.		

Q1-Create Table:

BILL (Bill_no, Bill_date, Amount, Tax, Total_amt)

Queries: (i) Display all the bill details of "24-jan-12"

(ii) Display the bill details where total amount is greater than 6000.

(iii) Display the bill details where total amount is less than 500.

Sol.

```
SQL> create table BILL
```

```
2  (Bill_No number(5,0),
3   Bill_Date date,
4   Amount number(10,2),
5   Tax number(5,0),
6   Total number(10,2));
```

Table created.

```
SQL> desc BILL
```

Name	Null?	Type
BILL_NO		NUMBER(5)
BILL_DATE		DATE
AMOUNT		NUMBER(10,2)
TAX		NUMBER(5)
TOTAL		NUMBER(10,2)

```
SQL> insert into BILL values(145,'03-Jan-19',5500,550,6050);
```

1 row created.

```
SQL> insert into BILL values(150,'10-Jan-19',100,10,110);
```

1 row created.

```
SQL> insert into BILL values(165,'24-Jan-12',900,90,990);
```

1 row created.

```
SQL> insert into BILL values(187,'24-Jan-12',10000,1000,11000);
```

1 row created.

```
SQL> insert into BILL values(130,'15-Mar-13',600,120,720);
```

1 row created.

```
SQL> insert into BILL values(210,'30-Mar-15',750,75,825);
```

1 row created.

```
SQL> select * from BILL;
```

BILL_NO	BILL_DATE	AMOUNT	TAX	TOTAL
145	03-JAN-19	5500	550	6050
150	10-JAN-19	100	10	110
165	24-JAN-12	900	90	990
187	24-JAN-12	10000	1000	11000
130	15-MAR-13	600	120	720
210	30-MAR-15	750	75	825

6 rows selected.

```
SQL> select * from BILL where Bill_Date='24-Jan-12';
```

BILL_NO	BILL_DATE	AMOUNT	TAX	TOTAL
165	24-JAN-12	900	90	990
187	24-JAN-12	10000	1000	11000

```
SQL> select * from BILL where Total>6000;
```

BILL_NO	BILL_DATE	AMOUNT	TAX	TOTAL
145	03-JAN-19	5500	550	6050
187	24-JAN-12	10000	1000	11000

```
SQL> select * from BILL where Total<500;
```

BILL_NO	BILL_DATE	AMOUNT	TAX	TOTAL
150	10-JAN-19	100	10	110

Q2- To perform queries for the table “BILL” using various DDL,DML and DQL commands.

- (i) Add a column „number_of_items“ to BILL table.
- (ii) Rename the table „BILL“ to „BILL_DETAILS“.
- (iii) Update the bill amount of 24-JAN-12.
- (iv) Display the bill no. and total amount multiplied by 3.

SOL.

```
SQL> alter table BILL
```

```
2 add(N_Items number(10));
```

Table altered.

```
SQL> desc BILL
```

Name	Null?	Type
BILL_NO		NUMBER(5)
BILL_DATE		DATE
AMOUNT		NUMBER(10,2)
TAX		NUMBER(5)
TOTAL		NUMBER(10,2)
N_ITEMS		NUMBER(10)

```
SQL> alter table BILL rename to BILL_DETAILS;
```

Table altered.

```
SQL> select * from BILL_DETAILS;
```

BILL_NO	BILL_DATE	AMOUNT	TAX	TOTAL	N_ITEMS
145	03-JAN-19	5500	550	6050	
150	10-JAN-19	100	10	110	
165	24-JAN-12	900	90	990	
187	24-JAN-12	10000	1000	11000	
130	15-MAR-13	600	120	720	
210	30-MAR-15	750	75	825	

```
SQL> update BILL_DETAILS
```

```
2 set total=total+10
```

```
3 where Bill_Date='24-Jan-12';
```

2 rows updated.

```
SQL> select * from BILL_DETAILS;
```

BILL_NO	BILL_DATE	AMOUNT	TAX	TOTAL	N_ITEMS
145	03-JAN-19	5500	550	6050	
150	10-JAN-19	100	10	110	
165	24-JAN-12	900	90	1000	
187	24-JAN-12	10000	1000	11010	
130	15-MAR-13	600	120	720	
210	30-MAR-15	750	75	825	

6 rows selected.

```
SQL> select Bill_No,Total*3 from BILL_DETAILS;
```

BILL_NO	TOTAL*3
145	18150
150	330
165	3000
187	33030
130	2160
210	2475

Q3- Create Table:

CAR(Regn_no, Make, Colour, Model, Owner)

Queries:

- (i) Display the owners who does not own a „Maruti“
- (ii) Display the car details of car make „Hyundai“
- (iii) Display model and owners from CAR
- (iv) Display the car details whose car colour is „RED“

SOL.

```
SQL> create table CAR
```

```
2  (Regn_No number(10),
3   Make varchar(25),
4   Colour varchar(10),
5   Model varchar(30),
6   Owner varchar(30));
```

Table created.

```
SQL> desc CAR
```

Name	Null?	Type
REGN_NO		NUMBER(10)
MAKE		VARCHAR2(25)
COLOUR		VARCHAR2(10)
MODEL		VARCHAR2(30)
OWNER		VARCHAR2(30)

```
SQL> insert into CAR values(35924,'Hyundai','White','Verna','Rakesh Mehta');
```

1 row created.

```
SQL> insert into CAR values(46424,'Maruti','Blue','Baleno','Shishir Ohri');
```

1 row created.

```
SQL> insert into CAR values(16254,'Nissan','Black','Micra','Tejas Gupta');
```

1 row created.

```
SQL> insert into CAR values(29584,'Ford','Silver','Ecosport','Aditya Singh');
```

1 row created.

```
SQL> insert into CAR values(13654,'Maruti','Red','Vitara Brezza','Avinash Kher');
```

1 row created.

```
SQL> insert into CAR values(16665,'Renault','Black','Duster','Mohan Das');
```

1 row created.

REGN_NO	MAKE	COLOUR	MODEL	OWNER
35924	Hyundai	White	Verna	Rakesh Mehta
46424	Maruti	Blue	Baleno	Shishir Ohri
16254	Nissan	Black	Micra	Tejas Gupta
29584	Ford	Silver	Ecosport	Aditya Singh
13654	Maruti	Red	Vitara Brezza	Avinash Kher
16665	Renault	Black	Duster	Mohan Das

```
SQL> select Owner from CAR where Make!='Maruti';
```

OWNER

Rakesh Mehta

Tejas Gupta

Aditya Singh

Mohan Das

```
SQL> select * from CAR where make='Hyundai';
```

REGN_NO	MAKE	COLOUR	MODEL	OWNER
35924	Hyundai	White	Verna	Rakesh Mehta

```
SQL> select Model,Owner from CAR;
```

MODEL	OWNER
Verna	Rakesh Mehta
Baleno	Shishir Ohri
Micra	Tejas Gupta
Ecosport	Aditya Singh
Vitara Brezza	Avinash Kher
Duster	Mohan Das

```
SQL> select * from CAR where Colour='Red';
```

REGN_NO	MAKE	COLOUR	MODEL	OWNER
13654	Maruti	Red	Vitara Brezza	Avinash Kher

Q4-To perform queries for the table “CAR” using various DDL,DML and DQL commands.

(i) Rename the table „CAR” to „CAR_INFORMATION”

(ii) Delete all the details of car „Hyundai”

(iii) Update the car color from „RED” to „METTALIC_RED”

(iv) Display the car details in ascending or descending order of colour.

SOL.

```
SQL> rename CAR to CAR_INFORMATION;
```

Table renamed.

```
SQL> delete from CAR_INFORMATION where make='Hyundai';
```

1 row deleted.

```
SQL> select * from CAR_INFORMATION;
```

REGN_NO	MAKE	COLOUR	MODEL	OWNER
46424	Maruti	Blue	Baleno	Shishir Ohri
16254	Nissan	Black	Micra	Tejas Gupta
29584	Ford	Silver	Ecosport	Aditya Singh
13654	Maruti	Red	Vitara Brezza	Avinash Kher
16665	Renault	Black	Duster	Mohan Das

```
SQL> update CAR_INFORMATION
```

```
2 set Colour='Metal_Red'
```

```
3 where Model='Vitara Brezza';
```

1 row updated.

```
SQL> select * from CAR_INFORMATION;
```

REGN_NO	MAKE	COLOUR	MODEL	OWNER
46424	Maruti	Blue	Baleno	Shishir Ohri
16254	Nissan	Black	Micra	Tejas Gupta
29584	Ford	Silver	Ecosport	Aditya Singh
13654	Maruti	Metal_Red	Vitara Brezza	Avinash Kher
16665	Renault	Black	Duster	Mohan Das

```
SQL> select * from CAR_INFORMATION order by Colour asc;
```

REGN_NO	MAKE	COLOUR	MODEL	OWNER
16254	Nissan	Black	Micra	Tejas Gupta
16665	Renault	Black	Duster	Mohan Das
46424	Maruti	Blue	Baleno	Shishir Ohri
13654	Maruti	Metal_Red	Vitara Brezza	Avinash Kher
29584	Ford	Silver	Ecosport	Aditya Singh

Q5.

1. Display all the information of the EMP table?

```
SQL> select * from emp;
```

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

2. Display the name of all employees with their salary.

```
SQL> select ename,sal from emp;
```

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

3. List all employees name who is working with department no 20.

```
SQL> select ename from emp where deptno=20;
```

ENAME

```

-----
SMITH
JONES
SCOTT
ADAMS
FORD

```

4. List the name of all Salesman and Analyst.

```
SQL> select ename,job from emp where job='SALESMAN' or job='ANALYST';
```

```

ENAME      JOB
-----
ALLEN      SALESMAN
WARD       SALESMAN
MARTIN     SALESMAN
SCOTT      ANALYST
TURNER     SALESMAN
FORD       ANALYST

```

5. List the name and job of all employees who are not clerk.

```
SQL> select ename,job from emp where job!='CLERK';
```

```

ENAME      JOB
-----
ALLEN      SALESMAN
WARD       SALESMAN
JONES      MANAGER
MARTIN     SALESMAN
BLAKE      MANAGER
CLARK      MANAGER
SCOTT      ANALYST
KING       PRESIDENT
TURNER     SALESMAN
FORD       ANALYST

```

6. List the name of employees whose employees no is 7369, 7521, 7839, 7934 or 7788

```
SQL> select ename,empno from emp where empno=7369 or empno=7521 or empno=7839
or empno=7934 or empno=7788;
```

```

ENAME      EMPNO
-----
SMITH      7369
WARD       7521
SCOTT      7788

```


KING	7839
MILLER	7934

7. List the employees details who does not belongs to department no 10 and 30

```
SQL> select * from emp where deptno!=10 and deptno!=30;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20

8. List the employees name and salary whose salary varies from 1000 to 2000.

```
SQL> select * from emp where sal between 1000 and 2000;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

9. . List employees name who have joined before 30 June-1981 and after 30 Dec-1981

```
SQL> select ename,hiredate from emp where hiredate<'30-Jun-81' or  
hiredate>'30-Dec-81';
```

ENAME	HIREDATE
SMITH	17-DEC-80
ALLEN	20-FEB-81
WARD	22-FEB-81
JONES	02-APR-81
BLAKE	01-MAY-81
CLARK	09-JUN-81
SCOTT	19-APR-87
ADAMS	23-MAY-87
MILLER	23-JAN-82

10. List the Commission and name of employees who are availing the commission.

```
SQL> select ename,comm from emp where comm!=0;
```

ENAME	COMM
ALLEN	300
WARD	500
MARTIN	1400

11. List the details of employees whose salary is greater than 2000 and commission is Null

```
SQL> select * from emp where sal>2000 or comm=null;
```

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

12. List the employees details whose name start with 'S'.

```
SQL> select * from emp where ename like 'S%';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20

13. List the employees name having 'I' as second character along with their Job.

```
SQL> select ename,job from emp where ename like '_I%';
```

ENAME	JOB
KING	PRESIDENT
MILLER	CLERK

14. List the employees name and date of joining in the descending order of date of joining the column title should be 'Date of Joining'.

```
SQL> select ename,hiredate AS date_of_joining from emp order by hiredate desc;
```

ENAME	DATE_OF_J
ADAMS	23-MAY-87
SCOTT	19-APR-87
MILLER	23-JAN-82
FORD	03-DEC-81
JAMES	03-DEC-81
KING	17-NOV-81
MARTIN	28-SEP-81
TURNER	08-SEP-81
CLARK	09-JUN-81
BLAKE	01-MAY-81
JONES	02-APR-81
WARD	22-FEB-81
ALLEN	20-FEB-81
SMITH	17-DEC-80

15. List the employee name, salary, job, and department number and display it in ascending order of department no and descending order of salary

```
SQL> select ename, sal, job, deptno from emp order by deptno asc, sal desc;
```

ENAME	SAL	JOB	DEPTNO
KING	5000	PRESIDENT	10
CLARK	2450	MANAGER	10
MILLER	1300	CLERK	10
SCOTT	3000	ANALYST	20
FORD	3000	ANALYST	20
JONES	2975	MANAGER	20
ADAMS	1100	CLERK	20
SMITH	800	CLERK	20
BLAKE	2850	MANAGER	30
ALLEN	1600	SALESMAN	30
TURNER	1500	SALESMAN	30
MARTIN	1250	SALESMAN	30
WARD	1250	SALESMAN	30
JAMES	950	CLERK	30

16. List the department no and total no of employees in each department.

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

DEPTNO	COUNT(*)
30	6

20	5
10	3

17. List the different job names available in emp table.

SQL> select job from emp group by job;

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

18. List department no and total salary payable in each department

SQL> select deptno,sum(sal) from emp group by deptno;

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

19. List the average salary and number of employees working in department no 20

SQL> select avg(sal),count(empno) from emp where deptno=20;

```
AVG(SAL) COUNT(EMPNO)
-----
2175      5
```

20. List the job and number of employees in each job the result should be in descending order of number of employees.

SQL> select job,count(job) from emp group by job order by count(job) desc;

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

21. List the total salary, maximum salary, minimum salary and average salary of employee's job wise for department no 10 only.

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

SUM(SAL)	MAX(SAL)	MIN(SAL)	AVG(SAL)	JOB
1300	1300	1300	1300	CLERK
5000	5000	5000	5000	PRESIDENT
2450	2450	2450	2450	MANAGER

22. List of the average salary of each job excluding manager.

```
SQL> select avg(sal),job from emp where job <> 'MANAGER' group by job;
```

AVG(SAL)	JOB
1037.5	CLERK
1400	SALESMAN
5000	PRESIDENT
3000	ANALYST

23. List the average salary of each job within each department.

```
SQL> select avg(sal),job,deptno from emp group by job,deptno;
```

AVG(SAL)	JOB	DEPTNO
2975	MANAGER	20
5000	PRESIDENT	10
1300	CLERK	10
1400	SALESMAN	30
3000	ANALYST	20
2850	MANAGER	30
2450	MANAGER	10
950	CLERK	30
950	CLERK	20

24. List the average salary of all departments in which more than 5 people working.

```
SQL> select avg(sal),deptno,count(deptno) from emp group by deptno having
count(deptno)>5;
```

AVG(SAL)	DEPTNO	COUNT(DEPTNO)

1566.66667

30

6

25. List the jobs of all employees where maximum salary is greater than equal to 5000.

```
SQL> select job,max(SAL) from emp group by job having max(SAL)>=5000;
```

JOB	MAX(SAL)
PRESIDENT	5000

26. List the total salary average salary of the employees job wise for department no 20 and display only those rows having average salary greater than 1000.

```
SQL> select sum(sal),avg(sal),job from emp where deptno=20 group by job having avg(sal)>1000;
```

SUM(SAL)	AVG(SAL)	JOB
2975	2975	MANAGER
6000	3000	ANALYST

27. Display the details of those employees who have joined before the end of September 1981.

```
SQL> select * from emp where hiredate<'30-SEP-81';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30

28. List the employee's name, employee id who are Manager.

```
SQL> select ename,empno from emp where job='MANAGER';
```

ENAME	EMPNO
JONES	7566

BLAKE	7698
CLARK	7782

29. List the name and designation of employee who does report any body.

```
SQL> select ename,job from emp where mgr IS NULL;
```

ENAME	JOB

KING	PRESIDENT

30. Display the number of employees working in each department. Display the department information also even if no employee working to that department.

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

DEPTNO	COUNT(ENAME)

10	3
20	5
30	6

31. List the name of manager along with employee's name, salary and manager code.

```
SQL> select w.ename,w.mgr,w.sal,m.ename,m.empno,m.sal from emp w, emp m where w.mgr=m.empno;
```

ENAME	MGR	SAL	ENAME	EMPNO	SAL

FORD	7566	3000	JONES	7566	2975
SCOTT	7566	3000	JONES	7566	2975
TURNER	7698	1500	BLAKE	7698	2850
ALLEN	7698	1600	BLAKE	7698	2850
WARD	7698	1250	BLAKE	7698	2850
JAMES	7698	950	BLAKE	7698	2850
MARTIN	7698	1250	BLAKE	7698	2850
MILLER	7782	1300	CLARK	7782	2450
ADAMS	7788	1100	SCOTT	7788	3000
BLAKE	7839	2850	KING	7839	5000
JONES	7839	2975	KING	7839	5000

ENAME	MGR	SAL	ENAME	EMPNO	SAL

CLARK	7839	2450	KING	7839	5000
SMITH	7902	800	FORD	7902	3000

32. List the employee's details who have joined the company before their manager.

```
SQL> select m.ename,m.hiredate,e.ename,e.hiredate from emp e, emp m where
m.mgr=e.empno and m.hiredate<e.hiredate;
```

ENAME	HIREDATE	ENAME	HIREDATE
ALLEN	20-FEB-81	BLAKE	01-MAY-81
WARD	22-FEB-81	BLAKE	01-MAY-81
BLAKE	01-MAY-81	KING	17-NOV-81
JONES	02-APR-81	KING	17-NOV-81
CLARK	09-JUN-81	KING	17-NOV-81
SMITH	17-DEC-80	FORD	03-DEC-81

6 rows selected.

33. Display different job available in the department no 20 and 30 using set operator.

```
SQL> select job from emp where deptno=20
2 UNION
3 select job from emp where deptno=30;
```

JOB
ANALYST
CLERK
MANAGER
SALESMAN

34. Display all jobs commission in department no 20 and 30 by using set operator.

```
SQL> select job,comm from emp where deptno=20
2 UNION
3 select job,comm from emp where deptno=30;
```

JOB	COMM
ANALYST	
CLERK	
MANAGER	
SALESMAN	0
SALESMAN	300
SALESMAN	500
SALESMAN	1400

35. Select all jobs unique in department no 20 and 30 by using set operator.

```
SQL> select job from emp where deptno=20
2 intersect
```



```
3 select job from emp where deptno=30;
```

```
JOB
```

```
-----
```

```
CLERK
```

```
MANAGER
```

36. List the employees name department no and department name from emp and dept table based on department number make sure that only those record select whose designation is manager

```
SQL> select e.* ,d.dname from emp e inner join dept d on e.deptno=d.deptno
where e.job='MANAGER';
```

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

37. Fetch the employees no and employees name from emp table where department no is 10 and 30 and order the based-on employee name using set operator.

```
SQL> select empno,ename from emp where deptno=10
2 UNION ALL
3 select empno,ename from emp where deptno=30 order by ename asc;
```

EMPNO	ENAME
7499	ALLEN
7698	BLAKE
7782	CLARK
7900	JAMES
7839	KING
7654	MARTIN
7934	MILLER
7844	TURNER
7521	WARD

```
9 rows selected.
```

38. Display the absolute value of -101

```
SQL> select abs(-101) from dual;
```

ABS(-101)
101

39. Calculate the remainder for given tow number (213, 9).

```
SQL> select remainder(213, 9)from dual;
```

```
REMAINDER(213,9)
```

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

40. Calculate the power of two number entered by user at runtime of query

```
SQL> select power(&a,&b) from dual;
```

```
Enter value for a: 2
```

```
Enter value for b: 3
```

```
old 1: select power(&a,&b) from dual
```

```
new 1: select power(2,3) from dual
```

```
POWER(2,3)
```

```
-----  
8
```

41. Display the all employees names along with the location of the character A in the Employees name from emp table where job is Clerk

```
SQL> select * from emp where ename LIKE '%A%' and job='CLERK';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30

42. Display the details of employee SCOTT

```
SQL> select * from emp where ename='SCOTT';
```

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

43. List all employee names, who have more than 20 years of experience in the company

```
SQL> select ename from emp where (sysdate-hiredate)>20;
```

```
ENAME
```

```
-----
```

```
SMITH
```

```
ALLEN
```

```
WARD
```

```
JONES
```

```
MARTIN
```

```
BLAKE
```

```
CLARK
```

SCOTT
KING
TURNER
ADAMS
JAMES
FORD
MILLER

14 rows selected.

44. Display the greatest and lowest value among salary and commission for all employees.

```
SQL> select max(sal),max(comm),min(sal),min(comm) from emp;
```

MAX(SAL)	MAX(COMM)	MIN(SAL)	MIN(COMM)
5000	1400	800	0

45. Display the smallest value of 16.0 and 59.2

```
SQL> select least(16.0,59.2) from dual;
```

LEAST(16.0,59.2)
16

46. Display the greatest integer value of 16.0 and 59.2

```
SQL> select greatest(16.0,59.2) from dual;
```

GREATEST(16.0,59.2)
59.2

Q6-Write PL/SQL code to print the table of the number inserted by user.

```
SQL> set serveroutput on
```

```
SQL> edit tables.sql
```

```
DECLARE
a integer;
i integer;
c integer;
BEGIN
dbms_output.put_line('enter the number: ');
a:=&a;
dbms_output.put_line('table for given number is: ');
i:=1;
WHILE i<10 LOOP
c:=a*i;
i:=i+1;
dbms_output.put_line(c);
END LOOP;
END;
/
```

```
SQL> @tables
```

```
Enter value for a: 10
```

```
old 7: a:=&a;
```

```
new 7: a:=10;
```

```
enter the number:
```

```
table for given number is:
```

10

20

30

40

50

60

70

80

90

PL/SQL procedure successfully completed.

Q7- Write PL/SQL code to print factorial of the number entered by the user.

```
SQL> edit factorial.sql
```

```
DECLARE
n number;
fac number:=1;
BEGIN
n:=&n;
WHILE n>0 LOOP
fac:=n*fac;
n:=n-1;
END LOOP;
dbms_output.put_line(fac);
END;
/
```

```
SQL> @factorial
```

```
Enter value for n: 5
```

```
old 5: n:=&n;
```

```
new 5: n:=5;
```

```
120
```

PL/SQL procedure successfully completed.

Q8-Write a PL/SQL code to print Fibonacci series till the number entered by the user.

```
SQL> edit fibonacci.sql
```

```
DECLARE
first integer:=0;
second integer:=1;
temp integer;
i integer;
n integer;
BEGIN
n:=&n;
dbms_output.put_line('Series: ');
dbms_output.put_line(first);
dbms_output.put_line(second);
FOR i in 2..n LOOP
temp:=first+second;
first:=second;
second:=temp;
dbms_output.put_line(temp);
END LOOP;
END;
/
```

```
SQL> @fibonacci
```

```
Enter value for n: 6
```

```
old 8: n:=&n;
```

```
new 8: n:=6;
```

```
Series:
```

0

1

1

2

3

5

8

PL/SQL procedure successfully completed.

Q9-Write a PL/SQL code to check that the number entered by the user is prime or not.

```
SQL> edit prime.sql
```

```
DECLARE
a integer;
i integer;
counter integer:=0;
BEGIN
dbms_output.put_line('enter the number: ');
a:=&a;
FOR i in 1..a LOOP
IF mod(a,i)=0 THEN
counter:=counter+1;
END IF;
END LOOP;
IF (counter>2) THEN
dbms_output.put_line('not a prime');
else
dbms_output.put_line('is prime');
END IF;
END;
/
```

```
SQL> @prime
```

```
Enter value for a: 23
```

```
old 7: a:=&a;
```

```
new 7: a:=23;
```

is prime

PL/SQL procedure successfully completed.

Q10-Write PL/SQL code to find that the number entered by user is even or odd.

```
SQL> set serveroutput on
```

```
SQL> edit odd.sql
```

```
DECLARE
a integer;
BEGIN
a:=&a;
IF mod(a,2)=0 then
dbms_output.put_line('this is even number');
ELSE
dbms_output.put_line('this is odd number');
END IF;
END;
/
```

```
SQL> @odd
```

```
Enter value for a: 20
```

```
old 4: a:=&a;
```

```
new 4: a:=20;
```

```
this is even number
```

PL/SQL procedure successfully completed.

Q11-Write PL/SQL code to add two numbers.

```
SQL> edit add.sql
```

```
DECLARE
a integer;
b integer;
c integer;
BEGIN
a:=&a;
b:=&b;
c:=a+b;
dbms_output.put_line('Sum-'|| c);
END;
/
```

```
SQL> @add
```

```
Enter value for a: 10
```

```
old 6: a:=&a;
```

```
new 6: a:=10;
```

```
Enter value for b: 20
```

```
old 7: b:=&b;
```

```
new 7: b:=20;
```

```
Sum-30
```

PL/SQL procedure successfully completed.

Q12-Write a code in PL/SQL using cursor to display the record of employees earning salary more than 3000.

```
SQL> edit c1.sql
```

```
DECLARE
e_id emp.empno%type;
e_name emp.ename%type;
e_job emp.job%type;
e_hiredate emp.hiredate%type;
e_sal emp.sal%type;
CURSOR c_emp IS
SELECT empno,ename,job,hiredate,sal
FROM emp
WHERE sal>3000;
BEGIN
OPEN c_emp;
FETCH c_emp INTO e_id,e_name,e_job,e_hiredate,e_sal;
dbms_output.put_line(e_id||' '||e_name||' '||e_job||' '||
e_hiredate||' '||e_sal);
CLOSE c_emp;
END;
/
```

```
SQL> @c1
```

```
7839 KING PRESIDENT 17-NOV-81 5000
```

PL/SQL procedure successfully completed.

Q13-Consider following Database

EMP2020(e_no,e_name,city,department)

Project20(p_no,p_name,status)

Emp_proj(e_no,p_no,no_of_days)

Create a trigger that restricts the insertion and updation of records having no_of_days less than zero.

```
SQL> set serveroutput on
```

```
SQL> edit p100.sql
```

```
CREATE OR REPLACE TRIGGER checkproject
BEFORE INSERT OR UPDATE ON emp_proj
FOR EACH ROW
BEGIN
IF:new.no_of_days<=0 THEN
raise_application_error(-20001,'no of days should be greater than 0');
END IF;
END;
/
```

```
SQL> @p100
```

```
Trigger created.
```

```
SQL> insert into emp_proj(e_no,p_no,no_of_days)values(1,10,0);
```

```
insert into emp_proj(e_no,p_no,no_of_days)values(1,10,0)
```

```
*
```

```
ERROR at line 1:
```

```
ORA-20001: no of days should be greater than 0
```

ORA-06512: at "SCOTT.CHECKPROJECT", line 3

ORA-04088: error during execution of trigger 'SCOTT.CHECKPROJECT'

Q14-Consider following relational database

Patient(p_no,p_name,p_addr)

Doctor(d_no,d_name,d_addr,d_city)

Patient_Doctor(p_no,d_no,disease,no_of_visits)

Write a procedure that will display doctor details who have treated diabetes patient.

```
SQL> set serveroutput on
```

```
SQL> edit p112.sql
```

```
CREATE OR REPLACE PROCEDURE p_printdoctor IS
CURSOR c_doctor IS select d.d_no,d.d_name from doctor d INNER JOIN
patient_doctor p ON d.d_no=p.d_no where p.diseases='diabetes';
r_doctor c_doctor%ROWTYPE;
BEGIN
OPEN c_doctor;
LOOP
FETCH c_doctor into r_doctor;
EXIT WHEN c_doctor%NOTFOUND;
dbms_output.put_line('Doctor Number: '||r_doctor.d_no);
dbms_output.put_line('Doctor Name: '||r_doctor.d_name);
END LOOP;
CLOSE c_doctor;
END;
```

/

```
SQL> @p112.sql
```


Procedure created.

SQL> edit p113.sql

BEGIN

p_printdoctor;

END;

/

SQL> @p113.sql

Doctor Number: 10

Doctor Name: Sarthak

Doctor Number: 20

Doctor Name: Prajwal

PL/SQL procedure successfully completed.