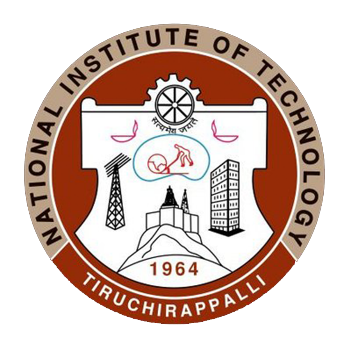
**NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI-620015**



**Department of Computer Applications**

**LABORATORY RECORD BOOK**

**DBMS LAB**

**CA 702**

**CONTINUOUS ASSESSMENT DOCUMENT**

*Submitted To: Submitted By:*

**Dr. U. Vignesh Akshay Jain**

**205119007**

SQL\*Plus: Release 12.1.0.2.0 Production on Sat Jan 25 09:28:01 2020

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

Enter user-name: system

Enter password:

Last Successful login time: Thu Jan 23 2020 09:56:11 +05:30

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

EXERCISE-1

**Problem 1.1: Create a table called 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)**

SQL> CREATE TABLE EMP(EMPNO NUMBER(6) NOT NULL, ENAME VARCHAR2(20), JOB VARCHAR2(10), MGR NUMBER(4) NOT NULL, DEPTNO NUMBER(3) NOT NULL, SAL NUMBER(7,2) NOT NULL);

Table created.

SQL> DESC EMP;

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NOT NULL NUMBER(6)

ENAME VARCHAR2(20)

JOB VARCHAR2(10)

MGR NOT NULL NUMBER(4)

DEPTNO NOT NULL NUMBER(3)

SAL NOT NULL NUMBER(7,2)

**Problem 1.2: Add a column commission to the emp table** **Commission numeric null allowed.**

SQL> ALTER TABLE EMP ADD COMMISSION INTEGER;

Table altered.

**Problem 1.3: Modify the column width of the job field of emp table.**

SQL> ALTER TABLE EMP MODIFY JOB VARCHAR(20);

Table altered.

**Problem 1.4: Create dept table with the following structure.**

**Name Type**

**--------------------------------- --------------------------- DEPTNO NUMBER(2) DNAME VARCHAR2(10) LOC VARCHAR2(10)**

SQL> CREATE TABLE DEPT(DEPTNO NUMBER(2), DNAME VARCHAR2(10), LOC VARCHAR2(10), PRIMARY KEY(DEPTNO));

Table created.

SQL> DESC DEPT;

Name Null? Type

----------------------------------------- -------- ----------------------------

DEPTNO NOT NULL NUMBER(2)

DNAME VARCHAR2(10)

LOC VARCHAR2(10)

**Problem 1.5: Add constraints to the emp table that empno as the primary key and**

**deptno as the foreign key.**

SQL> ALTER TABLE EMP MODIFY EMPNO INT PRIMARY KEY;

Table altered.

SQL> ALTER TABLE EMP ADD FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO);

Table altered.

**Problem 1.6: Add constraints to the emp table to check the empno value while**

**entering (i.e) empno > 100.**

SQL> ALTER TABLE EMP ADD CHECK (EMPNO>100);

Table altered.

**Problem 1.7: Salary value by default is 5000, otherwise as entered values**

SQL> ALTER TABLE EMP MODIFY SAL DEFAULT '5000';

Table altered.

**Problem 1.8: Add columns Dob to the emp table.**

SQL> ALTER TABLE EMP ADD DOB DATE;

Table altered.

SQL> DESC EMP;

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NOT NULL NUMBER(6)

ENAME VARCHAR2(20)

JOB VARCHAR2(20)

MGR NOT NULL NUMBER(4)

DEPTNO NOT NULL NUMBER(3)

SAL NOT NULL NUMBER(7,2)

COMMISSION NUMBER(38)

DOB DATE

SQL> DESC DEPT;

Name Null? Type

----------------------------------------- -------- ----------------------------

DEPTNO NOT NULL NUMBER(2)

DNAME VARCHAR2(10)

LOC VARCHAR2(10)

EXERCISE-2

**Problem 2.1: Insert 3 records into dept table.**

SQL> INSERT INTO DEPT VALUES(&DEPTNO, &DEPTNAME, &LOC);

Enter value for deptno: 10

Enter value for deptname: 'MANAGEMENT'

Enter value for loc: 'MAIN BLOCK'

old 1: INSERT INTO DEPT VALUES(&DEPTNO, &DEPTNAME, &LOC)

new 1: INSERT INTO DEPT VALUES(10, 'MANAGEMENT', 'MAIN BLOCK')

1 row created.

SQL> /

Enter value for deptno: 20

Enter value for deptname: 'DEVELOPMENT'

Enter value for loc: 'MANUFACTURING UNIT'

old 1: INSERT INTO DEPT VALUES(&DEPTNO, &DEPTNAME, &LOC)

new 1: INSERT INTO DEPT VALUES(20, 'DEVELOPMENT', 'MANUFACTURING UNIT')

INSERT INTO DEPT VALUES(20, 'DEVELOPMENT', 'MANUFACTURING UNIT')

\*

ERROR at line 1:

ORA-12899: value too large for column "SYSTEM"."DEPT"."DNAME" (actual: 11,

maximum: 10)

SQL> ALTER TABLE DEPT MODIFY LOC VARCHAR2(30);

Table altered.

SQL> INSERT INTO DEPT VALUES(&DEPTNO, &DEPTNAME, &LOC);

Enter value for deptno: 20

Enter value for deptname: 'DEVELOPMENT'

Enter value for loc: 'MANUFACTURING UNIT'

old 1: INSERT INTO DEPT VALUES(&DEPTNO, &DEPTNAME, &LOC)

new 1: INSERT INTO DEPT VALUES(20, 'DEVELOPMENT', 'MANUFACTURING UNIT')

INSERT INTO DEPT VALUES(20, 'DEVELOPMENT', 'MANUFACTURING UNIT')

\*

ERROR at line 1:

ORA-12899: value too large for column "SYSTEM"."DEPT"."DNAME" (actual: 11,

maximum: 10)

SQL> ALTER TABLE DEPT MODIFY DNAME VARCHAR2(30);

Table altered.

SQL> INSERT INTO DEPT VALUES(&DEPTNO, &DEPTNAME, &LOC);

Enter value for deptno: 20

Enter value for deptname: 'DEVELOPMENT'

Enter value for loc: 'MANUFACTURING UNIT'

old 1: INSERT INTO DEPT VALUES(&DEPTNO, &DEPTNAME, &LOC)

new 1: INSERT INTO DEPT VALUES(20, 'DEVELOPMENT', 'MANUFACTURING UNIT')

1 row created.

SQL> /

Enter value for deptno: 30

Enter value for deptname: 'MAINTAINANCE'

Enter value for loc: 'MAIN BLOCK'

old 1: INSERT INTO DEPT VALUES(&DEPTNO, &DEPTNAME, &LOC)

new 1: INSERT INTO DEPT VALUES(30, 'MAINTAINANCE', 'MAIN BLOCK')

1 row created.

SQL> SELECT \* FROM DEPT;

DEPTNO DNAME LOC

---------- ------------------------------ -----------------------------------------

10 MANAGEMENT MAIN BLOCK

20 DEVELOPMENT MANUFACTURING UNIT

30 MAINTAINANCE MAIN BLOCK

SQL> DESC EMP;

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NOT NULL NUMBER(6)

ENAME VARCHAR2(20)

JOB VARCHAR2(20)

MGR NOT NULL NUMBER(4)

DEPTNO NOT NULL NUMBER(3)

SAL NOT NULL NUMBER(7,2)

COMMISSION NUMBER(38)

DOB DATE

**Problem 2.2: Insert 10 records into emp table.**

SQL> INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB);

Enter value for empno: 7369

Enter value for ename: 'SMITH'

Enter value for job: 'CLERK'

Enter value for mgr: 7566

Enter value for deptno: 20

Enter value for sal: 800

Enter value for commission: 0

Enter value for dob: '17-DEC-80'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7369, 'SMITH', 'CLERK', 7566, 20, 800, 0, '17-DEC-80')

1 row created.

SQL> /

Enter value for empno: 7399

Enter value for ename: 'ASANT'

Enter value for job: 'SALESMAN'

Enter value for mgr: 7566

Enter value for deptno: 20

Enter value for sal: 1600

Enter value for commission: 300

Enter value for dob: '20-FEB-81'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7399, 'ASANT', 'SALESMAN', 7566, 20, 1600, 300, '20-FEB-81')

1 row created.

SQL> /

Enter value for empno: 7499

Enter value for ename: 'ALLEN'

Enter value for job: 'SALESMAN'

Enter value for mgr: 7698

Enter value for deptno: 30

Enter value for sal: 1600

Enter value for commission: 300

Enter value for dob: '20-FEB-81'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7499, 'ALLEN', 'SALESMAN', 7698, 30, 300, 300, '20-FEB-81')

1 row created.

SQL> /

Enter value for empno: 7521

Enter value for ename: 'WARD'

Enter value for job: 'SALESMAN'

Enter value for mgr: 7698

Enter value for deptno: 30

Enter value for sal: 1250

Enter value for commission: 500

Enter value for dob: '22-FEB-82'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7521, 'WARD', 'SALESMAN', 7698, 30, 1250, 500, '22-FEB-82')

1 row created.

SQL> /

Enter value for empno: 7566

Enter value for ename: 'JONES'

Enter value for job: 'MANAGER'

Enter value for mgr: 7839

Enter value for deptno: 20

Enter value for sal: 5975

Enter value for commission: 500

Enter value for dob: '02-APR-81'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7566, 'JONES', 'MANAGER', 7839, 20, 5975, 500, '02-APR-81')

1 row created.

SQL> /

Enter value for empno: 7698

Enter value for ename: 'BLAKE'

Enter value for job: 'MANAGER'

Enter value for mgr: 7839

Enter value for deptno: 30

Enter value for sal: 9850

Enter value for commission: 1400

Enter value for dob: '01-MAY-79'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7698, 'BLAKE', 'MANAGER', 7839, 30, 9850, 1400, '01-MAY-79')

1 row created.

SQL> /

Enter value for empno: 7611

Enter value for ename: 'SCOTT'

Enter value for job: 'HOD'

Enter value for mgr: 7839

Enter value for deptno: 10

Enter value for sal: 3000

Enter value for commission: NULL

Enter value for dob: '12-JUN-76'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7611, 'SCOTT', 'HOD', 7839, 10, 3000, NULL, '12-JUN-76')

1 row created.

SQL> /

Enter value for empno: 7839

Enter value for ename: 'CLARK'

Enter value for job: 'CEO'

Enter value for mgr: 0

Enter value for deptno: 10

Enter value for sal: 9900

Enter value for commission: NULL

Enter value for dob: '16-MAR-72'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7839, 'CLARK', 'CEO', 0, 10, 9900, NULL, '16-MAR-72')

1 row created.

SQL> /

Enter value for empno: 7368

Enter value for ename: 'FORD'

Enter value for job: 'SUPERVIS'

Enter value for mgr: 7366

Enter value for deptno: 20

Enter value for sal: 800

Enter value for commission: 0

Enter value for dob: '17-DEC-80'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7368, 'FORD', 'SUPERVIS', 7366, 20, 800, 0, '17-DEC-80')

1 row created.

SQL> /

Enter value for empno: 7599

Enter value for ename: 'ALLEY'

Enter value for job: 'SALESMAN'

Enter value for mgr: 7698

Enter value for deptno: 30

Enter value for sal: 1600

Enter value for commission: 300

Enter value for dob: '20-FEB-81'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7599, 'ALLEY', 'SALESMAN', 7698, 30, 1600, 300, '20-FEB-81')

1 row created.

SQL> /

Enter value for empno: 7421

Enter value for ename: 'DRANK'

Enter value for job: 'CLERCK'

Enter value for mgr: 7698

Enter value for deptno: 30

Enter value for sal: 1250

Enter value for commission: 500

Enter value for dob: '22-JAN-82'

old 1: INSERT INTO EMP VALUES(&EMPNO, &ENAME, &JOB, &MGR, &DEPTNO, &SAL, &COMMISSION, &DOB)

new 1: INSERT INTO EMP VALUES(7421, 'DRANK', 'CLERCK', 7698, 30, 1250, 500, '22-JAN-82')

1 row created.

SQL> SELECT \* FROM EMP;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

300 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

5975 500 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1400 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

11 rows selected.

**Problem 2.3: Update the emp table to set the default commission of all employees**

**to Rs 1000/- who are working as managers**

SQL> UPDATE EMP SET COMMISSION='1000' WHERE JOB='MANAGER';

2 rows updated.

**Problem 2.4: Create a pseudo table employee with the same structure as the table**

**emp and insert rows into the table using select clauses.**

SQL> CREATE TABLE EMPLOYEE AS SELECT \* FROM EMP;

Table created.

SQL> DESC EMPLOYEE;

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NOT NULL NUMBER(6)

ENAME VARCHAR2(20)

JOB VARCHAR2(20)

MGR NOT NULL NUMBER(4)

DEPTNO NOT NULL NUMBER(3)

SAL NOT NULL NUMBER(7,2)

COMMISSION NUMBER(38)

DOB DATE

SQL> SELECT \* FROM EMPLOYEE;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

300 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

11 rows selected.

**Problem 2.5: Delete only those who are working as supervisors.**

SQL> DELETE FROM EMPLOYEE WHERE JOB='SUPERVIS';

1 row deleted.

**Problem 2.6: Delete the rows whose empno is 7599.**

SQL> DELETE FROM EMPLOYEE WHERE EMPNO=7599;

1 row deleted.

**Problem 2.7: List the records in the emp table orderby salary in ascending order.**

SQL> SELECT \* FROM EMPLOYEE ORDER BY SAL;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7499 ALLEN SALESMAN 7698 30

300 300 20-FEB-81

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7611 SCOTT HOD 7839 10

3000 12-JUN-76

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

7839 CLARK CEO 0 10

9900 16-MAR-72

9 rows selected.

**Problem 2.8: List the records in the emp table orderby salary in descending order.**

SQL> SELECT \* FROM EMPLOYEE ORDER BY SAL DESC;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7839 CLARK CEO 0 10

9900 16-MAR-72

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7499 ALLEN SALESMAN 7698 30

300 300 20-FEB-81

9 rows selected.

**Problem 2.9: Display only those employees whose deptno is 30.**

SQL> SELECT \* FROM EMPLOYEE WHERE DEPTNO=30;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COM DOB

---------- ---------- ---------

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7698 BLAKE MANAGER 7839 30

9850 1400 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COM DOB

---------- ---------- ---------

7421 DRANK CLERK 7698 30

1250 500 22-JAN-82

**Problem 2.10: Display deptno from the table employee avoiding the duplicated**

**values.**

SQL> SELECT DISTINCT DEPTNO FROM EMPLOYEE;

DEPTNO

----------

30

20

10

**Problem 2.11: List the records in sorted order of their employees**

SQL> SELECT \* FROM EMP ORDER BY ENAME;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

7839 CLARK CEO 0 10

9900 16-MAR-72

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

7611 SCOTT HOD 7839 10

3000 12-JUN-76

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

11 rows selected.

**Problem 2.12: create a manager table from the emp table which should hold**

**details aonly about the managers.**

SQL> CREATE TABLE MANAGER AS SELECT \* FROM EMP WHERE JOB='MANAGER';

Table created.

SQL> DESC MANAGER;

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NOT NULL NUMBER(6)

ENAME VARCHAR2(20)

JOB VARCHAR2(20)

MGR NOT NULL NUMBER(4)

DEPTNO NOT NULL NUMBER(3)

SAL NOT NULL NUMBER(7,2)

COMMISSION NUMBER(38)

DOB DATE

**Problem 2.13: List the employee names whose commission is null**

SQL> SELECT \* FROM EMP WHERE COMMISSION=NULL;

no rows selected

**Problem 2.14: List the employee names and the department name in which they**

**are working.**

SQL> SELECT ENAME,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;

ENAME DNAME

-------------------- ------------------------------

SMITH DEVELOPMENT

ASANT DEVELOPMENT

ALLEN MAINTAINANCE

WARD MAINTAINANCE

JONES DEVELOPMENT

BLAKE MAINTAINANCE

SCOTT MANAGEMENT

CLARK MANAGEMENT

FORD DEVELOPMENT

ALLEY MAINTAINANCE

DRANK MAINTAINANCE

11 rows selected.

EXERCISE-3

**Problem 3.1: Select all employees from department numbers 7369,7499.**

SQL> SELECT \* FROM EMP WHERE EMPNO IN (7369,7499);

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7499 ALLEN SALESMAN 7698 30

300 300 20-FEB-81

**Problem 3.2: Display all the details of the records whose employee name starts**

**with ‘S’.**

SQL> SELECT \* FROM EMPLOYEE WHERE ENAME LIKE 'S%';

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7611 SCOTT HOD 7839 10

3000 12-JUN-76

**Problem 3.3: Display all the details of the records whose employee name does not**

**starts with ‘S’.**

SQL> SELECT \* FROM EMPLOYEE WHERE ENAME NOT LIKE 'S%';

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

300 300 20-FEB-81

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

7839 CLARK CEO 0 10

9900 16-MAR-72

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

7 rows selected.

**Problem 3.4: Display the rows whose empno ranges from 7500 to 7600.**

SQL> SELECT \* FROM EMPLOYEE WHERE EMPNO BETWEEN 7500 AND 7600;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

**Problem 3.5: Display the rows whose empno not in range from 7500 to 7600.**

SQL> SELECT \* FROM EMPLOYEE WHERE EMPNO NOT BETWEEN 7500 AND 7600;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

7 rows selected.

**Problem 3.6: Calculate the square root of the salary of all employees.**

SQL> SELECT SQRT(SAL) FROM EMP;

SQRT(SAL)

----------

28.2842712

40

17.3205081

35.3553391

77.2981242

99.2471662

54.7722558

99.4987437

35.3553391

9 rows selected.

**Problem 3.7: Count the total records in the emp table.**

SQL> SELECT COUNT(\*) FROM EMP;

COUNT(\*)

----------

11

**Problem 3.8: Calculate the total and average salary amount of the emptable.**

SQL> SELECT SUM(SAL) , AVG(SAL) FROM EMP;

SUM(SAL) AVG(SAL)

---------- ----------

37625 3420.45455

**Problem 3.9: Determine the max and min salary and rename the column as**

**max\_salary and min\_salary.**

SQL> SELECT MIN(SAL) "min\_sal", MAX(SAL) "max\_sal" FROM EMP;

min\_sal max\_sal

---------- ----------

800 9900

**Problem 3.10: Display total salary spent for employees.**

SQL> SELECT SUM(SAL) FROM EMP;

SUM(SAL)

----------

37625

**Problem 3.11: Display total salary spent for each job category.**

SQL> SELECT JOB, SUM(SAL) FROM EMP GROUP BY JOB;

JOB SUM(SAL)

-------------------- ----------

CEO 9900

SUPERVIS 800

CLERK 800

SALESMAN 6050

MANAGER 15825

HOD 3000

CLERCK 1250

7 rows selected.

**Problem 3.12: Display the month name of date “14-jul-09” in full.**

SQL> SELECT TO\_CHAR(TO\_DATE('14-JUL-09'),'MONTH') FROM DUAL;

TO\_CHAR(TO\_DATE('14-JUL-09'),'MONTH'

------------------------------------

JULY

**Problem 3.13: Display the Dob of all employees in the format “dd-mm-yy”.**

SQL> SELECT TO\_DATE(DOB, 'DD-MM-YY') FROM EMP;

TO\_DATE(D

---------

17-DEC-80

20-FEB-81

20-FEB-81

22-FEB-82

02-APR-81

01-MAY-79

12-JUN-76

16-MAR-72

17-DEC-80

20-FEB-81

22-JAN-82

11 rows selected.

**Problem 3.14: Display the date two months after the Dob of employees.**

SQL> SELECT ADD\_MONTHS(dob,2) from EMP;

ADD\_MONTH

---------

17-FEB-81

20-APR-81

20-APR-81

22-APR-82

02-JUN-81

01-JUL-79

12-AUG-76

16-MAY-72

17-FEB-81

20-APR-81

22-MAR-82

11 rows selected.

**Problem 3.15: Display the last date of that month in “05-Oct-09”.**

SQL> SELECT LAST\_DAY ('05-OCT-09') FROM DUAL;

LAST\_DAY(

---------

31-OCT-09

**Problem 3.16: Display the rounded date in the year format, month format, day**

**format in the employees.**

SQL> SELECT ROUND (TO\_DATE(DOB), 'YEAR') FROM EMP;

ROUND(TO\_

---------

01-JAN-81

01-JAN-81

01-JAN-81

01-JAN-82

01-JAN-81

01-JAN-79

01-JAN-76

01-JAN-72

01-JAN-81

01-JAN-81

01-JAN-82

11 rows selected.

SQL> SELECT ROUND (TO\_DATE(DOB), 'MONTH') FROM EMP;

ROUND(TO\_

---------

01-JAN-81

01-MAR-81

01-MAR-81

01-MAR-82

01-APR-81

01-MAY-79

01-JUN-76

01-APR-72

01-JAN-81

01-MAR-81

01-FEB-82

11 rows selected.

SQL> SELECT ROUND (TO\_DATE(DOB), 'DAY') FROM EMP;

ROUND(TO\_

---------

14-DEC-80

22-FEB-81

22-FEB-81

21-FEB-82

05-APR-81

29-APR-79

13-JUN-76

19-MAR-72

14-DEC-80

22-FEB-81

24-JAN-82

11 rows selected.

**Problem 3.17: Display the date 60 days before current date.**

SQL> SELECT (SYSDATE-60) FROM DUAL;

(SYSDATE-

---------

08-DEC-19

**Problem 3.18: List all employee names , salary and 15% rise in salary.**

SQL> SELECT ENAME,SAL,SAL+0.15\*SAL FROM EMP;

ENAME SAL SAL+0.15\*SAL

-------------------- ---------- ------------

SMITH 800 920

ASANT 1600 1840

ALLEN 1600 1840

WARD 1250 1437.5

JONES 5975 6871.25

BLAKE 9850 11327.5

SCOTT 3000 3450

CLARK 9900 11385

FORD 800 920

ALLEY 1600 1840

DRANK 1250 1437.5

11 rows selected.

**Problem 3.19: List all employees which starts with either B or C.**

SQL> SELECT ENAME FROM EMP WHERE ENAME LIKE 'B%' OR ENAME LIKE 'C%';

ENAME

--------------------

BLAKE

CLARK

**Problem 3.20: Display lowest paid employee details under each manager.**

SQL> SELECT ENAME,SAL,MGR FROM EMP WHERE SAL IN (SELECT MIN(SAL) FROM EMP GROUP BY MGR);

ENAME SAL MGR

-------------------- ---------- ----------

SMITH 800 7566

WARD 1250 7698

SCOTT 3000 7839

CLARK 9900 0

FORD 800 7366

DRANK 1250 7698

6 rows selected.

**Problem 3.21: Display number of employees working in each department and their**

**department name.**

SQL> SELECT DNAME, COUNT(ENAME) FROM EMP, DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO GROUP BY DNAME;

DNAME COUNT(ENAME)

------------------------------ ------------

MAINTAINANCE 5

MANAGEMENT 2

DEVELOPMENT 4

**Problem 3.22: Display the employee names whose name contains up to 5**

**characters.**

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

ENAME

--------------------

SMITH

ASANT

ALLEN

WARD

JONES

BLAKE

SCOTT

CLARK

FORD

ALLEY

DRANK

11 rows selected.

**Problem 3.23: List all employee names and their manager whose manager is**

**77499 or 7566 0r 7611.**

SQL> SELECT ENAME FROM EMP WHERE MGR IN(77499,7566,7611);

ENAME

--------------------

SMITH

ASANT

**Problem3.24: Find how many job titles are available in employee table.**

**Solution:**

SQL> SELECT COUNT(DISTINCT JOB) FROM EMP;

COUNT(DISTINCTJOB)

------------------

7

**Problem 3.25 : What is the difference between maximum and minimum salaries**

**of employees in the organization?**

SQL> SELECT MAX(SAL)-MIN(SAL) FROM EMP;

MAX(SAL)-MIN(SAL)

-----------------

9100

**Problem 3.26: Find no.of dept in employee table.**

SQL> SELECT COUNT(DISTINCT DEPTNO) FROM EMP;

COUNT(DISTINCTDEPTNO)

---------------------

3

**Problem 3.27: Display the names and dob of all employees who were born in**

**Feburary**.

SQL> SELECT ENAME, DOB FROM EMP WHERE TO\_CHAR(DOB,'MON')='FEB';

ENAME DOB

-------------------- ---------

ASANT 20-FEB-81

ALLEN 20-FEB-81

WARD 22-FEB-82

ALLEY 20-FEB-81

**Problem 3.28: List out the employee names who will celebrate their birthdays**

**during current month.**

SQL> SELECT ENAME FROM EMP WHERE TO\_CHAR(DOB,'MON') LIKE TO\_CHAR(SYSDATE,'MON');

ENAME

--------------------

ASANT

ALLEN

WARD

ALLEY

**Problem 3.29: List out the employee names whose names starts with s and ends**

**with h.**

SQL> SELECT ENAME FROM EMP WHERE ENAME LIKE('S%') AND ENAME LIKE ('%H');

ENAME

--------------------

SMITH

**Problem 3.30: List out the employee names whose salary is greater than**

**5000,6000**

SQL> SELECT ENAME FROM EMP WHERE SAL>ANY(5000,6000);

ENAME

--------------------

JONES

BLAKE

CLARK

**EXERCISE 4**

**Problem 4.1: Select all employees from ‘maintainance’ and ‘development’ dept**.

SQL> SELECT ENAME,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND (DNAME='MAINTAINANCE' OR DNAME='DEVELOPMENT');

ENAME DNAME

-------------------- ------------------------------

SMITH DEVELOPMENT

ASANT DEVELOPMENT

ALLEN MAINTAINANCE

WARD MAINTAINANCE

JONES DEVELOPMENT

BLAKE MAINTAINANCE

FORD DEVELOPMENT

ALLEY MAINTAINANCE

DRANK MAINTAINANCE

**Problem 4.2: Display all employee names and salary whose salary is greater**

**than minimum salary of the company and job title starts with ‘M’.**

SQL> SELECT ENAME FROM EMP WHERE SAL>(SELECT MIN(SAL) FROM EMP) AND JOB LIKE('M%');

ENAME

--------------------

JONES

BLAKE

**Problem 4.3: Issue a query to find all the employees who work in the same job as**

**jones.**

SQL> SELECT ENAME FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='JONES');

ENAME

--------------------

JONES

BLAKE

**Problem 4.4: Issue a query to display information about employees who earn more**

**than any employee in dept 30.**

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

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7839 CLARK CEO 0 10

9900 16-MAR-72

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7 rows selected.

**Problem 4.5: Display the employees who have the same job as jones and whose**

**salary &gt;= fords.**

SQL> SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='JONES') AND SAL>=(SELECT SAL FROM EMP WHERE ENAME='FORD');

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

**Problem 4.6: Write a query to display the name and job of all employees in dept**

**20 who have a job that someone in the Management dept as well.**

SQL> SELECT ENAME, JOB FROM EMP WHERE DEPTNO=20 AND JOB IN(SELECT JOB FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND DNAME='MANAGEMENT');

no rows selected

**Problem 4.7: Issue a query to list all the employees who salary is > the average**

**salary of their own dept.**

SQL> SELECT \* FROM EMP WHERE SAL >(SELECT AVG(SAL) FROM EMP);

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

7839 CLARK CEO 0 10

9900 16-MAR-72

**Problem 4.8: Write a query that would display the empname, job where each**

**employee works and the name of their dept.**

SQL> SELECT ENAME,JOB,DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;

ENAME JOB DNAME

-------------------- -------------------- ------------------------------

SMITH CLERK DEVELOPMENT

ASANT SALESMAN DEVELOPMENT

ALLEN SALESMAN MAINTAINANCE

WARD SALESMAN MAINTAINANCE

JONES MANAGER DEVELOPMENT

BLAKE MANAGER MAINTAINANCE

SCOTT HOD MANAGEMENT

CLARK CEO MANAGEMENT

FORD SUPERVIS DEVELOPMENT

ALLEY SALESMAN MAINTAINANCE

DRANK CLERCK MAINTAINANCE

11 rows selected.

**Problem 4.9: Write a query to list the employees having the same job as**

**employees located in ‘ mainblock’.(use multiple subquery)**

SQL> SELECT \* FROM EMP WHERE JOB IN (SELECT JOB FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND LOC='MAIN BLOCK');

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

9 rows selected.

**Problem 4.10: Write a query to list the employees in dept 10 with the same job as**

**anyone in the development dept.**

SQL> SELECT \* FROM EMP WHERE DEPTNO=10 AND JOB IN (SELECT JOB FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO AND DNAME='DEVELPOMENT');

no rows selected

**Problem 4.11: Write a query to list the employees with the same job and salary as**

**‘ford’.**

SQL> SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='FORD') AND SAL=(SELECT SAL FROM EMP WHERE ENAME='FORD');

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

**Problem 4.12: Write a query to list all depts. with at least 2 salesman.**

SQL> SELECT DNAME,JOB FROM DEPT,EMP WHERE EMP.JOB='SALESMAN';

DNAME JOB

------------------------------ --------------------

MANAGEMENT SALESMAN

MANAGEMENT SALESMAN

MANAGEMENT SALESMAN

MANAGEMENT SALESMAN

DEVELOPMENT SALESMAN

DEVELOPMENT SALESMAN

DEVELOPMENT SALESMAN

DEVELOPMENT SALESMAN

MAINTAINANCE SALESMAN

MAINTAINANCE SALESMAN

MAINTAINANCE SALESMAN

DNAME JOB

------------------------------ --------------------

MAINTAINANCE SALESMAN

12 rows selected.

**Problem 4.13: Write a query to list the employees in dept 20 with the same job as**

**anyone in dept 30.**

SQL> SELECT \* FROM EMP WHERE DEPTNO=20 AND JOB=ANY(SELECT JOB FROM EMP WHERE DEPTNO=30);

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

**Problem 4.14: List out the employee names who get the salary greater than the**

**maximum salaries of dept with dept no 20,30**

SQL> SELECT \* FROM EMP WHERE SAL>ANY(SELECT MAX(SAL) FROM EMP WHERE DEPTNO=20 OR DEPTNO=30);

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7839 CLARK CEO 0 10

9900 16-MAR-72

**Problem 4.15: Display the maximum salaries of the departments whose**

**maximum salary is greater than 9000.**

SQL> SELECT MAX(SAL) FROM EMP GROUP BY DEPTNO HAVING MAX(SAL)>9000;

MAX(SAL)

----------

9850

9900

**Problem 4.16: Display the maximum salaries of the departments whose**

**minimum salary is greater than 1000 and lesser than 5000.**

SQL> SELECT MAX(SAL) FROM EMP HAVING MIN(SAL)>1000 AND MIN(SAL)<5000 GROUP BY ENAME;

MAX(SAL)

----------

1250

1250

1600

1600

3000

1600

6 rows selected.

**Create the following table :**

**AccDept.( Accredited Department by quality council)**

**DNAME DEPTNO DCity**

**-------- ---------- ------- -- ---------- -------------- 10 MANAGEMENT MAIN BLOCK**

**20 DEVELOPMENT MANUFACTURING UNIT**

**30 MAINTAINANCE MAIN BLOCK**

SQL> CREATE TABLE AccDept AS SELECT \* FROM DEPT WHERE DEPTNO IN (10,20,30);

Table created.

**Problem 4.17: Display the departments that are accredited by the quality council.**

SQL> SELECT DEPT.DNAME FROM DEPT, AccDept WHERE DEPT.DEPTNO=AccDept.DEPTNO;

DNAME

------------------------------

MANAGEMENT

DEVELOPMENT

MAINTAINANCE

**Problem 4.18: Display the employees of departments which are not accredited by**

**the quality council**

SQL> SELECT ENAME FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME IN (SELECT DEPT.DNAME FROM DEPT, AccDept WHERE DEPT.DEPTNO=AccDept.DEPTNO));

ENAME

--------------------

SCOTT

CLARK

SMITH

ASANT

JONES

FORD

ALLEN

WARD

BLAKE

ALLEY

DRANK

11 rows selected.

**Problem 4.19: Display all the employees and the departments implementing a**

**left outer join.**

SQL> SELECT ENAME,DNAME FROM EMP LEFT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;

ENAME DNAME

-------------------- ------------------------------

SCOTT MANAGEMENT

CLARK MANAGEMENT

SMITH DEVELOPMENT

ASANT DEVELOPMENT

JONES DEVELOPMENT

FORD DEVELOPMENT

ALLEN MAINTAINANCE

WARD MAINTAINANCE

BLAKE MAINTAINANCE

ALLEY MAINTAINANCE

DRANK MAINTAINANCE

11 rows selected.

**Problem 4.20: Display the employee name and department name in which they**

**are working implementing a right outer join.**

SQL> SELECT ENAME,DNAME FROM EMP RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;

ENAME DNAME

-------------------- ------------------------------

SMITH DEVELOPMENT

ASANT DEVELOPMENT

ALLEN MAINTAINANCE

WARD MAINTAINANCE

JONES DEVELOPMENT

BLAKE MAINTAINANCE

SCOTT MANAGEMENT

CLARK MANAGEMENT

FORD DEVELOPMENT

ALLEY MAINTAINANCE

DRANK MAINTAINANCE

11 rows selected.

**Problem 4.21: Display the employee name and department name in which they**

**are working implementing a full outer join.**

SQL> SELECT ENAME,DNAME FROM EMP FULL OUTER JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;

ENAME DNAME

-------------------- ------------------------------

SMITH DEVELOPMENT

ASANT DEVELOPMENT

ALLEN MAINTAINANCE

WARD MAINTAINANCE

JONES DEVELOPMENT

BLAKE MAINTAINANCE

SCOTT MANAGEMENT

CLARK MANAGEMENT

FORD DEVELOPMENT

ALLEY MAINTAINANCE

DRANK MAINTAINANCE

11 rows selected.

**Problem 4.22: Write a query to display their employee names and their managers**

**name.**

SQL> SELECT A.ENAME AS EMPLOYEE, B.ENAME AS MANAGER FROM EMP A, EMP B WHERE A.MGR=B.EMPNO;

EMPLOYEE MANAGER

-------------------- --------------------

SMITH JONES

ASANT JONES

ALLEN BLAKE

WARD BLAKE

ALLEY BLAKE

DRANK BLAKE

JONES CLARK

BLAKE CLARK

SCOTT CLARK

9 rows selected.

**Problem 4.23: Write a query to display their employee names and their managers**

**salary for every employee .**

SQL> SELECT A.ENAME AS EMPLOYEE, B.SAL AS MANAGER\_SALARY FROM EMP A, EMP B WHERE A.MGR=B.EMPNO;

EMPLOYEE MANAGER\_SALARY

-------------------- --------------

SMITH 5975

ASANT 5975

ALLEN 9850

WARD 9850

ALLEY 9850

DRANK 9850

JONES 9900

BLAKE 9900

SCOTT 9900

9 rows selected.

**Problem 4.24: Write a query to output the name , job, empno, deptname and**

**location for each dept, even if there are no employees.**

SQL> SELECT ENAME,JOB,DNAME,LOC FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;

ENAME JOB DNAME

-------------------- -------------------- ------------------------------

LOC

------------------------------

SMITH CLERK DEVELOPMENT

MANUFACTURING UNIT

ASANT SALESMAN DEVELOPMENT

MANUFACTURING UNIT

ALLEN SALESMAN MAINTAINANCE

MAIN BLOCK

ENAME JOB DNAME

-------------------- -------------------- ------------------------------

LOC

------------------------------

WARD SALESMAN MAINTAINANCE

MAIN BLOCK

JONES MANAGER DEVELOPMENT

MANUFACTURING UNIT

BLAKE MANAGER MAINTAINANCE

MAIN BLOCK

ENAME JOB DNAME

-------------------- -------------------- ------------------------------

LOC

------------------------------

SCOTT HOD MANAGEMENT

MAIN BLOCK

CLARK CEO MANAGEMENT

MAIN BLOCK

FORD SUPERVIS DEVELOPMENT

MANUFACTURING UNIT

ENAME JOB DNAME

-------------------- -------------------- ------------------------------

LOC

------------------------------

ALLEY SALESMAN MAINTAINANCE

MAIN BLOCK

DRANK CLERCK MAINTAINANCE

MAIN BLOCK

11 rows selected.

**Problem 4.25: Find the name of the manager for each employee. Include** **the**

**following in the output: empno, empname, job and his manager’s name.**

SQL> SELECT A.EMPNO,A.ENAME AS EMPLOYEE,A.JOB,B.ENAME AS MANAGER FROM EMP A,EMP B WHERE A.MGR=B.EMPNO;

EMPNO EMPLOYEE JOB MANAGER

---------- -------------------- -------------------- --------------------

7369 SMITH CLERK JONES

7399 ASANT SALESMAN JONES

7499 ALLEN SALESMAN BLAKE

7521 WARD SALESMAN BLAKE

7599 ALLEY SALESMAN BLAKE

7421 DRANK CLERCK BLAKE

7566 JONES MANAGER CLARK

7698 BLAKE MANAGER CLARK

7611 SCOTT HOD CLARK

9 rows selected.

**Problem 4.26: Display the details of those who draw the same salary.**

SQL> SELECT ENAME FROM EMP WHERE SAL IN (SELECT SAL FROM EMP GROUP BY SAL HAVING COUNT(\*)>1);

ENAME

--------------------

ASANT

ALLEN

ALLEY

SMITH

FORD

WARD

DRANK

**EXERCISE 5**

**Problem 5.1: Display all the dept numbers available with the dept and accdept**

**tables avoiding duplicates.**

SQL> SELECT DISTINCT DEPTNO FROM DEPT UNION SELECT DISTINCT DEPTNO FROM AccDept;

DEPTNO

----------

10

20

30

**Problem 5.2: Display all the dept numbers available with the dept and accdept**

SQL> SELECT DEPTNO FROM DEPT UNION ALL SELECT DEPTNO FROM ACCDEPT;

DEPTNO

----------

10

20

30

10

20

30

6 rows selected.

**Problem 5.3: Display dept no available in both the dept and acc dept tables.**

SQL> SELECT DEPTNO FROM DEPT INTERSECT SELECT DEPTNO FROM ACCDEPT;

DEPTNO

----------

10

20

30

**Problem 5.4: Display all the dept numbers available in dept and not in accdept**

**tables.**

SQL> SELECT DEPTNO FROM DEPT MINUS SELECT DEPTNO FROM AccDept;

no rows selected

**Problem 5.5: The organization wants to display only the details of the employees**

**those who are managers.( horizontal portioning)**

SQL> CREATE VIEW view\_MANAGER AS SELECT \* FROM EMP WHERE JOB='MANAGER';

View created.

SQL> SELECT \* FROM view\_MANAGER;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7566 JONES MANAGER 7839 20

5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

**Problem 5.6: The organization wants to display only the details like empno,empname,deptno,deptname of the employees. (vertical portioning)**

SQL> CREATE VIEW GENERAL AS SELECT EMPNO,ENAME,DEPTNO FROM EMP;

View created.

SQL> SELECT \* FROM GENERAL;

EMPNO ENAME DEPTNO

---------- -------------------- ----------

7369 SMITH 20

7399 ASANT 20

7499 ALLEN 30

7521 WARD 30

7566 JONES 20

7698 BLAKE 30

7611 SCOTT 10

7839 CLARK 10

7368 FORD 20

7599 ALLEY 30

7421 DRANK 30

11 rows selected.

**Problem 5.7: The organization wants to display only the details like empno, empname, deptno, deptname of the all the employees except the HOD and CEO. (full portioning)**

SQL> CREATE VIEW VIEWALL AS SELECT EMPNO,ENAME,DEPTNO FROM EMP WHERE JOB NOT IN('HOD', 'CEO');

View created.

SQL> SELECT \* FROM VIEWALL;

EMPNO ENAME DEPTNO

---------- -------------------- ----------

7369 SMITH 20

7399 ASANT 20

7499 ALLEN 30

7521 WARD 30

7566 JONES 20

7698 BLAKE 30

7368 FORD 20

7599 ALLEY 30

7421 DRANK 30

9 rows selected.

**Problem 5.8: Display all the views generated.**

SQL> SELECT \* FROM TAB;

TNAME

--------------------------------------------------------------------------------

TABTYPE CLUSTERID

------- ----------

ACCDEPT

TABLE

AQ$DEF$\_AQCALL

VIEW

AQ$DEF$\_AQERROR

VIEW

TNAME

--------------------------------------------------------------------------------

TABTYPE CLUSTERID

------- ----------

AQ$\_DEF$\_AQCALL\_F

VIEW

AQ$\_DEF$\_AQERROR\_F

VIEW

AQ$\_INTERNET\_AGENTS

TABLE

TNAME

--------------------------------------------------------------------------------

TABTYPE CLUSTERID

------- ----------

AQ$\_INTERNET\_AGENT\_PRIVS

TABLE

AQ$\_QUEUES

TABLE

AQ$\_QUEUE\_TABLES

TABLE

TNAME

--------------------------------------------------------------------------------

TABTYPE CLUSTERID

------- ----------

AQ$\_SCHEDULES

TABLE

CATALOG

SYNONYM

COL

SYNONYM

TNAME

--------------------------------------------------------------------------------

TABTYPE CLUSTERID

------- ----------

DEF$\_AQCALL

TABLE

DEF$\_AQERROR

TABLE

DEF$\_CALLDEST

TABLE

TNAME

--------------------------------------------------------------------------------

TABTYPE CLUSTERID

------- ----------

DEF$\_DEFAULTDEST

TABLE

DEF$\_DESTINATION

TABLE

DEF$\_ERROR

TABLE

TNAME

--------------------------------------------------------------------------------

TABTYPE CLUSTERID

------- ----------

DEF$\_LOB

TABLE

DEF$\_ORIGIN

TABLE

DEF$\_PROPAGATOR

TABLE

…….

…….

…….

210 rows selected.

**Problem 5.9: Execute the DML commands on the view created.**

SQL> SELECT \* FROM GENERAL;

EMPNO ENAME DEPTNO

---------- -------------------- ----------

7369 SMITH 20

7399 ASANT 20

7499 ALLEN 30

7521 WARD 30

7566 JONES 20

7698 BLAKE 30

7611 SCOTT 10

7839 CLARK 10

7368 FORD 20

7599 ALLEY 30

7421 DRANK 30

11 rows selected.

**Problem 5.10: Drop a view.**

SQL> DROP VIEW GENERAL;

View dropped.

**EXERCISE 6**

**Program 6.1:Write a pl/sql program to swap two numbers with out taking third variable**

SQL> set serveroutput on;

SQL> declare

2 a number;

3 b number;

4 begin

5 a:=5;

6 b:=7;

7 dbms\_output.put\_line('before swapping: ');

8 dbms\_output.put\_line('a='||a||' b='||b);

9 a:=a+b;

10 b:=a-b;

11 a:=a-b;

12 dbms\_output.put\_line('after swapping: ');

13 dbms\_output.put\_line('a='||a||' b='||b);

14 end;

15 /

before swapping:

a=5 b=7

after swapping:

a=7 b=5

PL/SQL procedure successfully completed.

**Program 6.2:Write a pl/sql program to swap two numbers by taking third variable**

SQL> declare

2 a number;

3 b number;

4 temp number;

5 begin

6 a:=5;

7 b:=7;

8 dbms\_output.put\_line('before swapping: ');

9 dbms\_output.put\_line('a='||a||' b='||b);

10 temp:=a;

11 a:=b;

12 b:=temp;

13 dbms\_output.put\_line('after swapping: ');

14 dbms\_output.put\_line('a='||a||' b='||b);

15 end;

16 /

before swapping:

a=5 b=7

after swapping:

a=7 b=5

PL/SQL procedure successfully completed.

**Program 6.3: Write a pl/sql program to find the largest of two numbers**

SQL> DECLARE

2 N NUMBER;

3 M NUMBER;

4 BEGIN

5 N:=&NUMBER;

6 M:=&NUMBER;

7 IF N>M THEN

8 DBMS\_OUTPUT.PUT\_LINE(''|| N ||' IS GREATER THAN '|| M ||'');

9 ELSE

10 DBMS\_OUTPUT.PUT\_LINE(''|| M ||' IS GREATER THAN '|| N ||'');

11 END IF;

12 END;

13 /

Enter value for number: 45

old 11: N:=&NUMBER;

new 11: N:=45;

Enter value for number: 67

old 15: M:=&NUMBER;

new 15: M:=67;

67 IS GREATER THAN 45

PL/SQL procedure successfully completed.

**Program 6.4:Write a pl/sql program to find the total and average of 6 subjects and display the grade**

SQL> DECLARE

2

3 a NUMBER;

4 b NUMBER;

5 c NUMBER;

6 d NUMBER;

7 e NUMBER;

8 f NUMBER;

9 sumOf6 NUMBER;

10 avgOf6 NUMBER;

14 BEGIN

16 a:=&NUMBER;

17 b:=&NUMBER;

18 c:=&NUMBER;

19 d:=&NUMBER;

20 e:=&NUMBER;

21 f:=&NUMBER;

30 sumOf6 := a + b + c + d + e + f;

33 avgOf6 := sumOf6 / 3;

36 dbms\_output.Put\_line('Sum = '

37 ||sumOf6);

38

39 dbms\_output.Put\_line('Average = '

40 ||avgOf6);

42 if(sumOf6>50) then

43 dbms\_output.put\_line('FAIL');

44 elsif(avgOf6>=90) then dbms\_output.put\_line('A+');

45 elsif(avgOf6>=80) then dbms\_output.put\_line('A');

46 elsif(avgOf6>=70) then dbms\_output.put\_line('B');

47 elsif(avgOf6>=60) then dbms\_output.put\_line('C');

48 else

49 dbms\_output.put\_line('D');

50 end if;

51 END;

54 /

Enter value for number: 10

old 16: a:=&NUMBER;

new 16: a:=10;

Enter value for number: 20

old 17: b:=&NUMBER;

new 17: b:=20;

Enter value for number: 30

old 18: c:=&NUMBER;

new 18: c:=30;

Enter value for number: 40

old 19: d:=&NUMBER;

new 19: d:=40;

Enter value for number: 50

old 20: e:=&NUMBER;

new 20: e:=50;

Enter value for number: 60

old 21: f:=&NUMBER;

new 21: f:=60;

Sum = 210

Average = 70

FAIL

**Program 6.5:Write a pl/sql program to find the sum of digits in a given number**

SQL> DECLARE

2 n INTEGER;

3 temp\_sum INTEGER;

4 r INTEGER;

5 BEGIN

6 n :=&n;

7 temp\_sum := 0;

8 WHILE n <> 0 LOOP

9 r := MOD(n, 10);

10 temp\_sum := temp\_sum + r;

11 n := Trunc(n / 10);

12 END LOOP;

13 dbms\_output.Put\_line('sum of digits = '

14 || temp\_sum);

15 END;

16 /

Enter value for n: 12345

old 6: n :=&n;

new 6: n :=12345;

sum of digits = 15

PL/SQL procedure successfully completed.

**Program 6.6:Write a pl/sql program to display the number in reverse order**

SQL> declare

2 n number;

3 i number;

4 rev number:=0;

5 r number;

6

7 begin

8 n:=&n;

9

10 while n>0

11 loop

12 r:=mod(n,10);

13 rev:=(rev\*10)+r;

14 n:=trunc(n/10);

15 end loop;

16

17 dbms\_output.put\_line('reverse is '||rev);

18

19 end;

20 /

Enter value for n: 12345

old 8: n:=&n;

new 8: n:=12345;

reverse is 54321

PL/SQL procedure successfully completed**.**

**Program 6.7:Write a pl/sql program to check whether the given number is prime or not**

SQL> declare

2 n number;

3 i number;

4 flag number;

5 begin

6 i:=2;

7 flag:=1;

8 n:=&n;

9 for i in 2..n/2

10 loop

11 if mod(n,i)=0

12 then

13 flag:=0;

14 exit;

15 end if;

16 end loop;

17 if flag=1

18 then

19 dbms\_output.put\_line('prime');

20 else

21 dbms\_output.put\_line('not prime');

22 end if;

23 end;

24 /

Enter value for n: 5

old 8: n:=&n;

new 8: n:=5;

prime

PL/SQL procedure successfully completed.

**Program 6.8: Write a pl/sql program to find the factorial of a given number**

SQL> declare

2 n number;

3 fac number:=1;

4 i number;

5 begin

6 n:=&n;

7 for i in 1..n

8 loop

9 fac:=fac\*i;

10 end loop;

11 dbms\_output.put\_line('factorial='||fac);

12 end;

13 /

Enter value for n: 4

old 6: n:=&n;

new 6: n:=4;

factorial=24

PL/SQL procedure successfully completed.

**Program 6.9:Write a pl/sql code block to calculate the area of a circle for a value of radius varying** **from 3 to 7.** **Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius &amp; area**

**TABLE NAME:AREAS**

**RADIUS AREA**

SQL> create table areas(radius number(10),area number(6,2));

Table created.

SQL> DECLARE

2 pi constant number(3,2):=3.14;

3 radius number(2);

4 area number(10,2);

5 BEGIN

6 radius:=3;

7 while(radius<=7)

8 LOOP

9 area:=pi\*power(radius,2);

10 Insert into areas values(radius,area);

11 radius:=radius+1;

12 END LOOP;

13 dbms\_output.put\_line('Records are successfully inserted');

14 END;

15 /

Records are successfully inserted

PL/SQL procedure successfully completed.

SQL> select \* from areas;

RADIUS AREA

---------- ----------

3 28.26

4 50.24

5 78.5

6 113.04

7 153.86

**Program 6.10: write a pl/sql code block that will accept an account number from the** **user,check if the users balance is less than minimum balance,only then deduct rs.100/- from the** **balance.this process is fired on the acct table.**

SQL> create table acct(name varchar2(10),cur\_bal number(10),acctno number(6,2));

Table created.

SQL> insert into acct values(&name,&cur\_bal,&acctno);

Enter value for name: 'sirius'

Enter value for cur\_bal: 10000

Enter value for acctno: 777

old 1: insert into acct values(&name,&cur\_bal,&acctno)

new 1: insert into acct values('sirius',10000,777)

1 row created.

SQL> /

Enter value for name: 'john'

Enter value for cur\_bal: 1000

Enter value for acctno: 765

old 1: insert into acct values(&name,&cur\_bal,&acctno)

new 1: insert into acct values('john',1000,765)

1 row created.

SQL> /

Enter value for name: 'sam'

Enter value for cur\_bal: 500

Enter value for acctno: 855

old 1: insert into acct values(&name,&cur\_bal,&acctno)

new 1: insert into acct values('sam',500,855)

1 row created.

SQL> /

Enter value for name: 'peter'

Enter value for cur\_bal: 800

Enter value for acctno: 353

old 1: insert into acct values(&name,&cur\_bal,&acctno)

new 1: insert into acct values('peter',800,353)

1 row created.

SQL> DECLARE

2 xacctno number(5);

3 xminbal number(5):=1000;

4 xbalance number(5);

5 BEGIN

6 xacctno:=&xacctno;

7 select cur\_bal into xbalance from acct where acctno=xacctno;

8 IF(xbalance < xminbal) THEN

9 update acct set cur\_bal=cur\_bal-100 where acctno=xacctno;

10 xbalance:=xbalance-100;

11 dbms\_output.put\_line('Rs 100 is deducted and current balance is'||xbalance);

12 ELSE

13 dbms\_output.put\_line('Current balance is'||xbalance);

14 END IF;

15 END;

16 /

Enter value for xacctno: 353

old 6: xacctno:=&xacctno;

new 6: xacctno:=353;

Rs 100 is deducted and current balance is700

PL/SQL procedure successfully completed.

**EXERCISE 7**

**Program 7.1 Write a procedure to add an amount of Rs.1000 for the employees whose salaries**

**is greater than 5000 and who belongs to the deptno passed as an argument.**

SQL> CREATE OR REPLACE PROCEDURE SALARY(DEPTID NUMBER) AS

2 BEGIN

3 UPDATE EMP SET SAL=SAL+1000 WHERE SAL>5000 AND DEPTNO=DEPTID;

4 END;

5 /

Procedure created.

SQL> SELECT \* FROM EMP;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

6975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

9850 1000 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

11 rows selected.

SQL> EXEC SALARY(30);

PL/SQL procedure successfully completed.

SQL> SELECT \* FROM EMP;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

6975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

10850 1000 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

11 rows selected.

**Program 7.2 Write a PL/SQL block to update the salary of the employee with a 10% increase**

**whose empno is to be passed as an argument for the procedure.**

SQL> CREATE OR REPLACE PROCEDURE SALARY1(EMPID NUMBER) AS

2 BEGIN

3 UPDATE EMP SET SAL=SAL+SAL\*(0.1) WHERE EMPNO=EMPID;

4 END;

5 /

Procedure created.

SQL> SELECT \* FROM EMP;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

800 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

6975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

10850 1000 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

11 rows selected.

SQL> EXEC SALARY1(7369);

PL/SQL procedure successfully completed.

SQL> SELECT \* FROM EMP;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

880 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

6975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

10850 1000 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

11 rows selected.

**Program 7.3 Write a function to find the salary of the employee who is working in the deptno**

**20(to be passed as an argument).**

SQL> CREATE OR REPLACE PROCEDURE GET\_SAL(DEPT NUMBER) AS

2 BEGIN

3 FOR S IN (SELECT \* FROM EMP WHERE DEPTNO = DEPT)

4 LOOP

5 DBMS\_OUTPUT.PUT\_LINE(S.SAL);

6 END LOOP;

7 END;

8 /

Procedure created.

SQL> EXEC GET\_SAL(20);

880

1600

6975

800

PL/SQL procedure successfully completed.

**Program 7.4 Write a function to find the nature of job of the employee whose deptno is 20(to be**

**passed as an argument)**

SQL> CREATE OR REPLACE PROCEDURE GET\_NATURE(DEPT NUMBER) AS

2 BEGIN

3 FOR S IN (SELECT \* FROM EMP WHERE DEPTNO = DEPT)

4 LOOP

5 DBMS\_OUTPUT.PUT\_LINE(S.JOB);

6 END LOOP;

7 END;

8 /

Procedure created.

SQL> EXEC GET\_NATURE(20);

CLERK

SALESMAN

MANAGER

SUPERVIS

PL/SQL procedure successfully completed.

**Program 7.5 Write a PL/SQL block to obtain the department name of the employee who works**

**for deptno 30.**

SQL> CREATE OR REPLACE PROCEDURE DEP\_NAME(DEPTID NUMBER) AS

2 BEGIN

3 FOR DEP IN (SELECT \* FROM DEPT WHERE DEPTNO = DEPTID)

4 LOOP

5 DBMS\_OUTPUT.PUT\_LINE(DEP.DNAME);

6 END LOOP;

7 END;

8 /

Procedure created.

SQL> EXEC DEP\_NAME(30);

MAINTAINANCE

PL/SQL procedure successfully completed.

**EXERCISE 8**

**Program 8.1 Write a Trigger to ensure that DEPT TABLE does not contain duplicate of null**

**values in DEPTNO column.**

SQL> CREATE OR REPLACE TRIGGER FIRST

2 BEFORE INSERT ON DEPT

3 FOR EACH ROW

4 DECLARE

5 A NUMBER;

6 BEGIN

7 IF (:NEW.DEPTNO IS NULL) THEN

8 RAISE\_APPLICATION\_ERROR(-20001,’ERROR::DEPTNO CANNOT BE NULL’);

9 ELSE

10 SELECT COUNT(\*) INTO A FROM DEPT WHERE DEPTNO=:NEW.DEPTNO;

11 IF(A=1) THEN

12 RAISE\_APPLICATION\_ERROR(-20002,’ERROR::CANNOT HAVE DUPLICATE VALUE’);

13 END IF;

14 END IF;

15 END;

16 /

Trigger created.

SQL> SELECT \* FROM DEPT;

DEPTNO DNAME LOC

---------- ------------------------------ ------------------------------

10 MANAGEMENT MAIN BLOCK

20 DEVELOPMENT MANUFACTURING UNIT

30 MAINTAINANCE MAIN BLOCK

SQL> INSERT INTO DEPT VALUES(10, 'FINANCE' ,'ADMIN BLOCK');

INSERT INTO DEPT VALUES(10, 'FINANCE' ,'ADMIN BLOCK')

\*

ERROR at line 1:

ORA-20002: ERROR::CANNOT HAVE DUPLICATE VALUE

ORA-06512: at "SYSTEM.FIRST", line 9

ORA-04088: error during execution of trigger 'SYSTEM.FIRST'

SQL> INSERT INTO DEPT VALUES(40, 'FINANCE' ,'ADMIN BLOCK');

1 row created.

**Program 8.2 Write a Trigger to carry out the following action: on deleting a deptno from dept**

**table , all the records with that deptno has to be deleted from the emp table**

SQL> CREATE OR REPLACE TRIGGER TRGR\_TO\_DELETE

2 BEFORE DELETE ON DEPT FOR EACH ROW

3 DECLARE

4 CURSOR GET\_EMP( P\_DEPTNO NUMBER ) IS

5 SELECT EMPNO, ENAME , JOB, MGR, SAL, COMMISSION, DOB

6 FROM EMP

7 WHERE DEPTNO=P\_DEPTNO;

8 BEGIN

9 DBMS\_OUTPUT.PUT\_LINE( 'DELETE DEPT = ' || :OLD.DEPTNO );

10 DBMS\_OUTPUT.PUT\_LINE( '- DEPT NAME = ' || :OLD.DNAME );

11 DBMS\_OUTPUT.PUT\_LINE( '- DEPT LOC = ' || :OLD.LOC );

12 FOR GET\_EMP\_REC IN GET\_EMP( :OLD.DEPTNO ) LOOP

13 DBMS\_OUTPUT.PUT( '- EMP ( ' || GET\_EMP\_REC.EMPNO );

14 DBMS\_OUTPUT.PUT( ', ' || GET\_EMP\_REC.ENAME );

15 DBMS\_OUTPUT.PUT( ', ' || GET\_EMP\_REC.JOB );

16 DBMS\_OUTPUT.PUT( ', ' || GET\_EMP\_REC.MGR );

17 DBMS\_OUTPUT.PUT( ', ' || GET\_EMP\_REC.SAL );

18 DBMS\_OUTPUT.PUT( ', ' || GET\_EMP\_REC.COMMISSION );

19 DBMS\_OUTPUT.PUT( ', ' || GET\_EMP\_REC.JOB );

20 DBMS\_OUTPUT.PUT\_LINE( ' )' );

21 DELETE FROM EMP WHERE DEPTNO = :OLD.DEPTNO;

22 END LOOP;

23 END;

24 /

Trigger created.

SQL> SELECT \* FROM EMP;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

880 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

6975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

10850 1000 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7611 SCOTT HOD 7839 10

3000 12-JUN-76

7839 CLARK CEO 0 10

9900 16-MAR-72

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

11 rows selected.

SQL> SELECT \* FROM DEPT;

DEPTNO DNAME LOC

---------- ------------------------------ ------------------------------

20 DEVELOPMENT MANUFACTURING UNIT

30 MAINTAINANCE MAIN BLOCK

40 FINANCE ADMIN BLOCK

10 MAINTAINANCE MAIN BLOCK

SQL> DELETE FROM DEPT WHERE DEPTNO = 10;

DELETE DEPT = 10

- DEPT NAME = MAINTAINANCE

- DEPT LOC = MAIN BLOCK

- EMP ( 7611, SCOTT, HOD, 7839, 3000, , HOD )

- EMP ( 7839, CLARK, CEO, 0, 9900, , CEO )

1 row deleted.

SQL> SELECT \* FROM EMP;

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7369 SMITH CLERK 7566 20

880 0 17-DEC-80

7399 ASANT SALESMAN 7566 20

1600 300 20-FEB-81

7499 ALLEN SALESMAN 7698 30

1600 300 20-FEB-81

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7521 WARD SALESMAN 7698 30

1250 500 22-FEB-82

7566 JONES MANAGER 7839 20

6975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30

10850 1000 01-MAY-79

EMPNO ENAME JOB MGR DEPTNO

---------- -------------------- -------------------- ---------- ----------

SAL COMMISSION DOB

---------- ---------- ---------

7368 FORD SUPERVIS 7366 20

800 0 17-DEC-80

7599 ALLEY SALESMAN 7698 30

1600 300 20-FEB-81

7421 DRANK CLERCK 7698 30

1250 500 22-JAN-82

9 rows selected.

SQL> SELECT \* FROM DEPT;

DEPTNO DNAME LOC

---------- ------------------------------ ------------------------------

20 DEVELOPMENT MANUFACTURING UNIT

30 MAINTAINANCE MAIN BLOCK

40 FINANCE ADMIN BLOCK

3 rows selected.

**Program 8.3 Write a Trigger to carry out the following action: on deleting any records from the**

**emp table,the same values must be inserted into the log table.**

SQL> CREATE TABLE LOG (DELETED\_EMPNO NUMBER(38), DELETED\_EMPNAME VARCHAR2(20), DELETED\_JOB VARCHAR2(12), DELETED\_MGR VARCHAR2(12), DELETED\_DEPTNO NUMBER(8), DELETED\_SAL NUMBER(6), DELETED\_COMMI FLOAT(126), DELETED\_DOB DATE, DATE1 DATE);

Table created.

SQL> SELECT \* FROM LOG;

no rows selected

SQL> CREATE OR REPLACE TRIGGER EMP\_AFTER\_DELETE

2 AFTER DELETE ON EMP FOR EACH ROW

3 DECLARE

4 CURSOR GET\_EMP( P\_EMPNO NUMBER ) IS

5 SELECT EMPNO, ENAME , JOB, MGR, DEPTNO,SAL, COMMISSION, DOB

6 FROM EMP

7 WHERE EMPNO=P\_EMPNO;

8 BEGIN

9 INSERT INTO LOG (DELETED\_EMPNO, DELETED\_EMPNAME, DELETED\_JOB, DELETED\_MGR, DELETED\_DEPTNO, DELETED\_SAL, DELETED\_COMMI, DELETED\_DOB, DATE1)

10 VALUES (:OLD.EMPNO, :OLD.ENAME, :OLD.JOB, :OLD.MGR, :OLD.DEPTNO, :OLD.SAL, :OLD.COMMISSION, :OLD.DOB, SYSDATE());

11 DBMS\_OUTPUT.PUT\_LINE( 'RECORD OF EMPLOYEE WITH EMPLOYEE ID ' || :OLD.DEPTNO || 'HAS BEEN DELETED FROM EMP FROM AND HAS BEEN INSERTED TO LOG TABLE.' );

12 END;

13 Trigger created.

SQL> DELETE FROM EMP WHERE EMPNO = 7369;

RECORD OF EMPLOYEE WITH EMPLOYEE ID 20HAS BEEN DELETED FROM EMP FROM AND HAS BEEN INSERTED TO LOG TABLE.

1 row deleted.

SQL> SELECT \* FROM LOG;

DELETED\_EMPNO DELETED\_EMPNAME DELETED\_JOB DELETED\_MGR DELETED\_DEPTNO

------------- -------------------- ------------ ------------ --------------

DELETED\_SAL DELETED\_COMMI DELETED\_D DATE1

----------- ------------- --------- ---------

7369 SMITH CLERK 7566 20

880 0 17-DEC-80 14-JUL-20

**--END--**