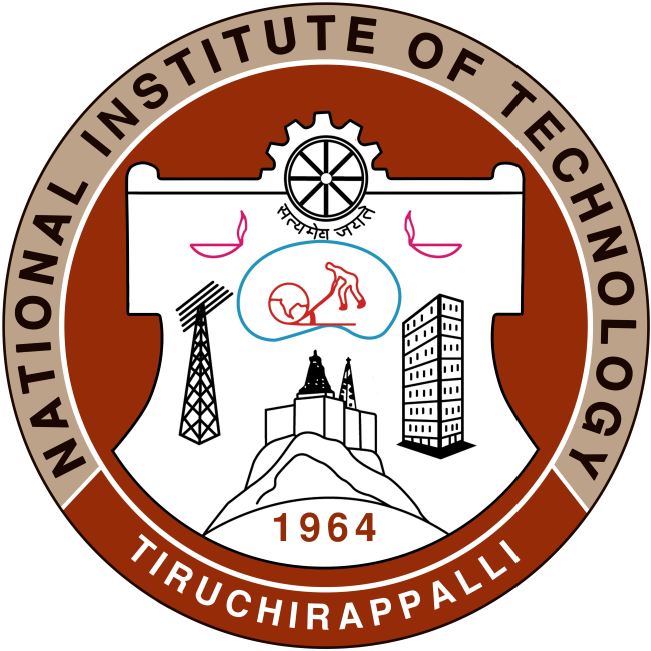
**NATIONAL INSTITUTE OF** **TECHNOLOGY, TIRUCHIRAPPALLI**

**Tamil Nadu-620015**



# *‘Database Management System’*

**LAB EXCERCISE**

**Submitted To:**  **Submitted By:**

**Dr. R. Balaji Ganesh Sumit Parmar**

**Roll No. – 205119102**

**MCA - II Semester ‘B’**

SQL\*Plus: Release 12.1.0.2.0 Production on Thu Jan 30 11:08:33 2020

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

Enter user-name: system

Enter password:

ERROR:

ORA-01017: invalid username/password; logon denied

Enter user-name: system

Enter password:

ERROR:

ORA-01005: null password given; logon denied

Enter user-name: system

Enter password:

Last Successful login time: Thu Jan 30 2020 09:38:17 +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.**

SQL> CREATE TABLE EMP (EMPNO NUMBER(6),EMPNAME VARCHAR(20) NOT NULL,JOB VARCHAR(10) NOT NULL,MGR NUMBER(8),DEPTNO NUMBER(6),SAL NUMBER(7,2))

2 ;

CREATE TABLE EMP (EMPNO NUMBER(6),EMPNAME VARCHAR(20) NOT NULL,JOB VARCHAR(10) NOT NULL,MGR NUMBER(8),DEPTNO NUMBER(6),SAL NUMBER(7,2))

\*

ERROR at line 1:

ORA-00955: name is already used by an existing object

SQL> CREATE TABLE EMP (EMPNO NUMBER(6),EMPNAME VARCHAR(20) NOT NULL,JOB VARCHAR(10) NOT NULL,MGR NUMBER(8),DEPTNO NUMBER(6),SAL NUMBER(7,2));

Table created.

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

SQL> ALTER TABLE EMP ADD COMMI NUMBER(5);

ALTER TABLE EMP ADD COMMI NUMBER(5)

\*

ERROR at line 1:

ORA-01430: column being added already exists in table

SQL>

SQL> ALTER TABLE EMP ADD COMMI NUMBER(5);

Table altered.

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

SQL>

SQL> ALTER TABLE EMP MODIFY JOB VARCHAR(12);

Table altered.

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

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

Deptno as the primarykey

SQL> create table dept(deptno number(20)primary key,dname varchar2(10),loc varchar2(10));

create table dept(deptno number(20)primary key,dname varchar2(10),loc varchar2(10))

\*

ERROR at line 1:

ORA-00955: name is already used by an existing object

SQL> drop table dept;

Table dropped.

SQL>

SQL> CREATE TABLE DEPT (DEPTNO NUMBER(5) PRIMARY KEY,DNAME VARCHAR(20),DLOC VARCHAR(20));

Table created.

**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 KE;

ALTER TABLE EMP MODIFY EMPNO INT PRIMARY KE

\*

ERROR at line 1:

ORA-00905: missing keyword

SQL>

SQL> ALTER TABLE EMP MODIFY EMPNO INT PRIMARY KEY;

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 constraint empno check(emp>100);

alter table emp add constraint empno check(emp>100)

\*

ERROR at line 1:

ORA-00904: "EMP": invalid identifier

SQL>

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 FLOT DEFAULT 5000 NOT NULL;

ALTER TABLE EMP MODIFY SAL FLOT DEFAULT 5000 NOT NULL

\*

ERROR at line 1:

ORA-01442: column to be modified to NOT NULL is already NOT NULL

SQL>

SQL> ALTER TABLE EMP MODIFY SAL FLOAT DEFAULT 5000 NOT NULL;

Table altered.

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

SQL> ALTER TABLE EMP ADD\_DOB DATE;

ALTER TABLE EMP ADD\_DOB DATE

\*

ERROR at line 1:

ORA-01735: invalid ALTER TABLE option

SQL> ALTER TABLE EMP AD\_O\_DOB DATE;

ALTER TABLE EMP AD\_O\_DOB DATE

\*

ERROR at line 1:

ORA-01735: invalid ALTER TABLE option

SQL>

SQL> ALTER TABLE EMP ADD DOB DATE;

Table altered.

**Exercise 2**

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

SQL>

SQL> INSERT INTO DEPT VALUES(10,'MANAGEMENT','MAIN BLOCK');

1 row created.

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

1 row created.

SQL> INSERT INTO DEPT VALUES(30,'MAINTAINANCE','MAIN BLOCK');

1 row created.

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

1 row created.

SQL> INSERT INTO DEPT VALUES(50,'SALES','HEAD OFFICE');

1 row created.

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

SQL>

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7369,'SMITH','CLERK',7566,'17-DEC80',800,0,20);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO) VALUES(7399,'ASANT','SALESMAN',7566,'20-FEB81',1600,300,20);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7499,'ALLEN','SALESMAN',7698,'20-FEB81',1600,300,30);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7521,'WARD','SALESMAN',7698,'22-FEB82',1250,500,30);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7566,'JONES','MANAGER',7839,'02-APR81',5975,500,20);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7698,'BLAKE','MANAGER',7839,'01-MAY79',9850,1400,30);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7611,'SCOTT','HOD',7839,'12-JUN76',3000,NULL,10);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB ,MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7839,'CLARK','CEO',NULL,'16-MAR72',9900,NULL,10);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7368,'FORD','SUPERVIS',7366,'17-DEC80',800,0,20);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7599,'ALLEY','SALESMAN',7698,'20-FEB81',1600,300,30);

1 row created.

SQL> INSERT INTO EMP(EMPNO, EMPNAME ,JOB, MGR ,DOB ,SAL ,COMMI, DEPTNO)

2 VALUES(7421,'DRANK','CLERCK',7698,'22-JAN82',1250,500,30);

1 row created.

**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 COMMI=1000 WHERE JOB=MANAGER;

UPDATE EMP SET COMMI=1000 WHERE JOB=MANAGER

\*

ERROR at line 1:

ORA-00904: "MANAGER": invalid identifier

SQL>

SQL> UPDATE EMP SET COMMI=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 (EMPNO INTEGER PRIMARY KEY,ENAME VARCHAR(20) NOT NULL,JOB VARCHAR(30) NOT NULL,MGR ITEGER,DEPTNO INTEGER,SAL INTEGER,COMM INTGER,DOB VARCHAR(10));

CREATE TABLE EMPLOYEE (EMPNO INTEGER PRIMARY KEY,ENAME VARCHAR(20) NOT NULL,JOB VARCHAR(30) NOT NULL,MGR ITEGER,DEPTNO INTEGER,SAL INTEGER,COMM INTGER,DOB VARCHAR(10))

\*

ERROR at line 1:

ORA-00902: invalid datatype

SQL>

SQL> CREATE TABLE EMPLOYEE (EMPNO INTEGER PRIMARY KEY,ENAME VARCHAR(20) NOT NULL,JOB VARCHAR(30) NOT NULL,MGR INTEGER,DEPTNO INTEGER,SAL INTEGER,COMM INTEGER,DOB VARCHAR(10));

Table created.

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

SQL> DELETE FROM EMPLOYEE WHERE JOB='SUPERVIS;

ERROR:

ORA-01756: quoted string not properly terminated

SQL>

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 EMP.NO=7599;

DELETE FROM EMPLOYEE WHERE EMP.NO=7599

\*

ERROR at line 1:

ORA-00904: "EMP"."NO": invalid identifier

SQL>

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>

SQL> SELECT \* FROM EMP ORDER BY SAL;

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7369 SMITH CLERK 7566 20 800

0 17-DEC-80

7368 FORD SUPERVIS 7366 20 800

0 17-DEC-80

7521 WARD SALESMAN 7698 30 1250

500 22-FEB-82

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7421 DRANK CLERCK 7698 30 1250

500 22-JAN-82

7399 ASANT SALESMAN 7566 20 1600

300 20-FEB-81

7499 ALLEN SALESMAN 7698 30 1600

300 20-FEB-81

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7599 ALLEY SALESMAN 7698 30 1600

300 20-FEB-81

7611 SCOTT HOD 7839 10 3000

12-JUN-76

7566 JONES MANAGER 7839 20 5975

1000 02-APR-81

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7698 BLAKE MANAGER 7839 30 9850

1000 01-MAY-79

7839 CLARK CEO 10 9900

16-MAR-72

11 rows selected.

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

SQL>

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

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7839 CLARK CEO 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 EMPNAME JOB MGR DEPTNO SAL

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

COMMI 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 EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7599 ALLEY SALESMAN 7698 30 1600

300 20-FEB-81

7421 DRANK CLERCK 7698 30 1250

500 22-JAN-82

7521 WARD SALESMAN 7698 30 1250

500 22-FEB-82

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7369 SMITH CLERK 7566 20 800

0 17-DEC-80

7368 FORD SUPERVIS 7366 20 800

0 17-DEC-80

11 rows selected.

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

SQL>

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

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI 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

1000 01-MAY-79

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7599 ALLEY SALESMAN 7698 30 1600

300 20-FEB-81

7421 DRANK CLERCK 7698 30 1250

500 22-JAN-82

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

SQL>

SQL> SELECT DISTINCT DEPTNO FROM EMP;

DEPTNO

----------

30

10

20

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

SQL>

SQL> SELECT \* FROM EMP ORDER BY EMPNAME;

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI 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 EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7698 BLAKE MANAGER 7839 30 9850

1000 01-MAY-79

7839 CLARK CEO 10 9900

16-MAR-72

7421 DRANK CLERCK 7698 30 1250

500 22-JAN-82

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI 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 EMPNAME JOB MGR DEPTNO SAL

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

COMMI 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';

create table manager as select \* from emp where job='manager'

\*

ERROR at line 1:

ORA-00955: name is already used by an existing object

SQL> create table manager as( select \* from emp) where job='manager';

create table manager as( select \* from emp) where job='manager'

\*

ERROR at line 1:

ORA-00933: SQL command not properly ended

SQL> drop table manager;

Table dropped.

SQL>

SQL> create table manager as select \* from EMP where JOB='MANAGER';

Table created.

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

SQL>

SQL> select \* from emp where commi IS null;

EMPNO EMPNAME JOB MGR DEPTNO SAL

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

COMMI DOB

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

7611 SCOTT HOD 7839 10 3000

12-JUN-76

7839 CLARK CEO 10 9900

16-MAR-72

**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.

SQL>

**Exercise 3**

SQL\*Plus: Release 12.1.0.2.0 Production on Thu Feb 6 09:24:02 2020

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

Enter user-name: system

Enter password:

Last Successful login time: Fri Jan 31 2020 14:36:49 +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

SQL> desc emp

Name Null? Type

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

EMPNO NOT NULL NUMBER(6)

ENAME NOT NULL VARCHAR2(20)

JOB NOT NULL VARCHAR2(20)

MGR NUMBER(4)

DEPTNO NUMBER(3)

SAL NUMBER(7,2)

COMMISSION NOT NULL VARCHAR2(10)

DOB VARCHAR2(10)

SQL> desc employee

Name Null? Type

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

EMPNO NUMBER(6)

ENAME NOT NULL VARCHAR2(20)

JOB NOT NULL VARCHAR2(20)

MGR NUMBER(4)

DEPTNO NUMBER(3)

SAL NUMBER(7,2)

COMMISSION NOT NULL VARCHAR2(10)

DOB VARCHAR2(10)

SQL> desc dept

Name Null? Type

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

DEPTNO NOT NULL NUMBER(20)

DNAME VARCHAR2(30)

LOC VARCHAR2(30)

SQL> select ename from employee where deptno=7369 or 7499;

select ename from employee where deptno=7369 or 7499

\*

ERROR at line 1:

ORA-00920: invalid relational operator

SQL> select ename from employee where deptno=7369 and 7499;

select ename from employee where deptno=7369 and 7499

\*

ERROR at line 1:

ORA-00920: invalid relational operator

SQL> select ename from employee where deptno=7369,7499;

select ename from employee where deptno=7369,7499

\*

ERROR at line 1:

ORA-00933: SQL command not properly ended

SQL> select ename from employee where deptno=7369,7499;

select ename from employee where deptno=7369,7499

\*

ERROR at line 1:

ORA-00933: SQL command not properly ended

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

SQL> select ename from employee where deptno=7369 and deptno=7499;

no rows selected

SQL> select ename from employee where deptno =7369 and deptno=7499;

no rows selected

**Problem 3.2: Display all the details of the records whose employee name starts** **with ‘S’.**

SQL> select \* from emp 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 0 12-jun-76

**Problem 3.3: Display all the details of the records whose employee name does not** **starts with ‘S’.**

SQL> select \* from emp 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

1600 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 0 16-mar-72

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.

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

SQL> select \* from emp where empno>7500 and empno<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

7599 alley salesman 7698 30

1600 300 20-feb-81

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

SQL> select \* from employee where empno>7500 and empno<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

SQL> select \* from employee where not empno>7500 and not empno<7600;

no rows selected

SQL> select \* from emp where not empno>7500 and not empno<7600;

no rows selected

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

SQL> select sqrt(sal) from employee;

SQRT(SAL)

----------

28.2842712

40

40

35.3553391

77.2981242

99.2471662

54.7722558

99.4987437

28.2842712

35.3553391

40.620192

SQRT(SAL)

----------

74.4983221

12 rows selected.

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

SQL> select count\* from emp;

select count\* from emp

\*

ERROR at line 1:

ORA-00936: missing expression

SQL> select count(\*) from emp;

COUNT(\*)

----------

11

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

SQL> select round(sum(sal),0)"sum",round(avg(sal),0)"average" from emp;

sum average

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

37625 3420

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

SQL> select round(max(sal),0)"MAX\_SALARY",round(min(sal),0

2 select round(max(sal),0)"MAX\_SALARY",round(min(sal),0)"MIN\_SALARY" from emp;

select round(max(sal),0)"MAX\_SALARY",round(min(sal),0)"MIN\_SALARY" from emp

\*

ERROR at line 2:

ORA-00907: missing right parenthesis

SQL> select round(max(sal),0)"MAX\_SALARY",round(min(sal),0)"MIN\_SALARY" from emp;

MAX\_SALARY MIN\_SALARY

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

9900 800

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

SQL> select sum(sal) from emp;

SUM(SAL)

----------

37625

SQL> select sum(sal) from emp;

SUM(SAL)

----------

37625

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

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

SUM(SAL)

----------

6050

800

1250

9900

800

15825

3000

7 rows selected.

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

JOB SUM(SAL)

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

salesman 6050

clerk 800

clerck 1250

ceo 9900

supervis 800

manager 15825

hod 3000

7 rows selected.

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

SQL> select convert("14-jul-09",char);

select convert("14-jul-09",char)

\*

ERROR at line 1:

ORA-00936: missing expression

SQL> select monthname("14-jul-09");

select monthname("14-jul-09")

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

SQL> select monthname(14-jul-09);

select monthname(14-jul-09)

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

SQL> select declare @date datetime2='14-jul-09';

select declare @date datetime2='14-jul-09'

\*

ERROR at line 1:

ORA-01729: database link name expected

SQL> select to\_char(to\_date('14-jul-09'),'month') from dual;

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

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

july

SQL> select ename from employee where deptno =7369 or deptno=7499;

no rows selected

SQL> select ename from employee where deptno =20 or deptno=30;

ENAME

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

smith

asant

allen

ward

jones

blake

ford

drank

prank

9 rows selected.

SQL> select month('14-jul-09');

select month('14-jul-09')

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

SQL> select monthname('14-jul-09');

select monthname('14-jul-09')

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

**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;

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.

SQL> select dateadd(month,2,dob) as dateadd;

select dateadd(month,2,dob) as dateadd

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

SQL> select dateadd(month,2,dob) as dateadd from emp;

select dateadd(month,2,dob) as dateadd from emp

\*

ERROR at line 1:

ORA-00904: "DATEADD": invalid identifier

SQL> select addmonth(dob,2) from emp;

select addmonth(dob,2) from emp

\*

ERROR at line 1:

ORA-00904: "ADDMONTH": invalid identifier

SQL> select addmon(dob,2) from emp;

select addmon(dob,2) from emp

\*

ERROR at line 1:

ORA-00904: "ADDMON": invalid identifier

SQL> select add\_month(dob,2) from emp;

select add\_month(dob,2) from emp

\*

ERROR at line 1:

ORA-00904: "ADD\_MONTH": invalid identifier

SQL> select ADD\_MONTH(dob,2) from emp;

select ADD\_MONTH(dob,2) from emp

\*

ERROR at line 1:

ORA-00904: "ADD\_MONTH": invalid identifier

SQL> select ADD\_MONTH(2,dob) from emp;

select ADD\_MONTH(2,dob) from emp

\*

ERROR at line 1:

ORA-00904: "ADD\_MONTH": invalid identifier

SQL> select add\_month(dob,2) from employee;

select add\_month(dob,2) from employee

\*

ERROR at line 1:

ORA-00904: "ADD\_MONTH": invalid identifier

SQL> select add\_months(dob,2) from employee;

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

22-MAR-82

02-MAR-88

ADD\_MONTH

---------

27-MAR-98

12 rows selected.

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

SQL> select last\_day('05-oct-09') from emp;

LAST\_DAY(

---------

31-OCT-09

31-OCT-09

31-OCT-09

31-OCT-09

31-OCT-09

31-OCT-09

31-OCT-09

31-OCT-09

31-OCT-09

31-OCT-09

31-OCT-09

11 rows selected.

SQL> select last\_day('05-oct-09') ;

select last\_day('05-oct-09')

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

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

SQL> select round(to\_date(dob

2 select round(to\_date(dob)

3 select round(to\_date(dob);

select round(to\_date(dob)

\*

ERROR at line 2:

ORA-00907: missing right parenthesis

SQL> select round(to\_date(dob),'month') round(to\_date(dob),'year') from emp;

select round(to\_date(dob),'month') round(to\_date(dob),'year') from emp

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

SQL> select round(to\_date('16-sept-2015'),'month') round(to\_date('16-sept-2015),'year') from emp;

ERROR:

ORA-01756: quoted string not properly terminated

SQL> select round(to\_date('16-sept-2015'),'month') "month" round(to\_date('16-sept-2015),'year') "year" from emp;

ERROR:

ORA-01756: quoted string not properly terminated

SQL> select round(to\_date('16-sept-2015'),'month') "month", round(to\_date('16-sept-2015),'year') "year" from emp;

ERROR:

ORA-01756: quoted string not properly terminated

SQL> select round(to\_date('16-sept-2015'),'month') "month", round(to\_date('16-sept-2015),'year') "year" from emp;

ERROR:

ORA-01756: quoted string not properly terminated

SQL> select round(to\_date('16-sept-2015'),'month') "new month", round(to\_date('16-sept-2015),'year') "new year" from emp;

ERROR:

ORA-01756: quoted string not properly terminated

SQL> select round(to\_date(dob,'dd-mm-yyyy'),'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,'dd-mm-yyyy'),'year')frm emp;

select round(to\_date(dob,'dd-mm-yyyy'),'year')frm emp

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

SQL> select round(to\_date(dob,'dd-mm-yyyy'),'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(todate(dob,'dd-mm-yyyy'),'day')from emp;

select round(todate(dob,'dd-mm-yyyy'),'day')from emp

\*

ERROR at line 1:

ORA-00904: "TODATE": invalid identifier

SQL> select round(to\_date(dob,'dd-mm-yyyy'),'day')from emp;

ROUND(TO\_

---------

17-DEC-80

18-FEB-81

18-FEB-81

24-FEB-82

01-APR-81

02-MAY-79

09-JUN-76

15-MAR-72

17-DEC-80

18-FEB-81

20-JAN-82

11 rows selected.

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

SQL> select date\_sub("2017-06-15"),interval 60 day);

select date\_sub("2017-06-15"),interval 60 day)

\*

ERROR at line 1:

ORA-00923: FROM keyword not found where expected

SQL> select add\_months(dob,-2) from emp;

ADD\_MONTH

---------

17-OCT-80

20-DEC-80

20-DEC-80

22-DEC-81

02-FEB-81

01-MAR-79

12-APR-76

16-JAN-72

17-OCT-80

20-DEC-80

22-NOV-81

11 rows selected.

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

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 0 12-jun-76

7839 clark ceo 0 10

9900 0 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 3.18: List all employee names , salary and 15% rise in salary.**

SQL> select ename,sal,sal+sal\*.15 from emp;

ENAME SAL SAL+SAL\*.15

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

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 \* from emp where ename like 'b%'or ename like 'c%';

EMPNO ENAME JOB MGR DEPTNO

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

SAL COMMISSION DOB

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

7698 blake manager 7839 30

9850 1000 01-may-79

7839 clark ceo 0 10

9900 0 16-mar-72

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

SQL> select ename,sal from emp where min(sal);

select ename,sal from emp where min(sal)

\*

ERROR at line 1:

ORA-00934: group function is not allowed here

SQL> select ename,sal from emp where min(sal) groupby manager;

select ename,sal from emp where min(sal) groupby manager

\*

ERROR at line 1:

ORA-00934: group function is not allowed here

SQL> select ename,min(sal) from emp groupby manager;

select ename,min(sal) from emp groupby manager

\*

ERROR at line 1:

ORA-00933: SQL command not properly ended

SQL> select ename,min(sal) from emp groupby job;

select ename,min(sal) from emp groupby job

\*

ERROR at line 1:

ORA-00933: SQL command not properly ended

SQL> select ename,min(sal) from emp group by mgr;

select ename,min(sal) from emp group by mgr

\*

ERROR at line 1:

ORA-00979: not a GROUP BY expression

SQL> select ename,sal from emp where min(Sal) group by mgr;

select ename,sal from emp where min(Sal) group by mgr

\*

ERROR at line 1:

ORA-00934: group function is not allowed here

SQL> select ename,mgr from emp where min(Sal) group by mgr;

select ename,mgr from emp where min(Sal) group by mgr

\*

ERROR at line 1:

ORA-00934: group function is not allowed here

SQL> select ENAME,SAL,MGR from EMP where SAL in (select min(SAL) from EMP group by MGR);

EMPNAME SAL MGR

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

SMITH 800 7566

WARD 1250 7698

SCOTT 3000 7839

CLARK 9900

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 (empname) from emp, dept where emp.deptno=dept.deptno group by dname;

DNAME COUNT(EMPNAME)

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

MAINTAINANCE 5

DEVELOPMENT 4

MANAGEMENT 2

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

SQL> select empname from emp where length(empname)<=5;

EMPNAME

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

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 empname from emp where mgr in (7399,7698,7566);

EMPNAME

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

SMITH

ASANT

ALLEN

WARD

ALLEY

DRANK

6 rows selected.

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

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 empname,dob from emp where to\_char(dob,'mon')='feb';

EMPNAME 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 EMPname from EMP where to\_char(dob,'MON') like to\_char(sysdate, 'MON');

no rows selected

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

SQL> select empname from emp where empname like 'S%H';

EMPNAME

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

SMITH

**Problem 3.30: List out the employee names whose salary is greater than** **5000,6000**

SQL> select empname from emp where sal>5000;

EMPNAME

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

JONES

BLAKE

CLARK

SQL>

**Exercise 4**

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

SQL>

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

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB DEPTNO DNAME

DLOC

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

7369 SMITH CLERK 7566

20 800 0 17-DEC-80 20 DEVELOPMENT MANUFACTURING UNIT

7399 ASANT SALESMAN 7566

20 1600 300 20-FEB-81 20 DEVELOPMENT MANUFACTURING UNIT

7499 ALLEN SALESMAN 7698

30 1600 300 20-FEB-81 30 MAINTAINANCE MAIN BLOCK

7521 WARD SALESMAN 7698

30 1250 500 22-FEB-82 30 MAINTAINANCE MAIN BLOCK

7566 JONES MANAGER 7839

20 5975 1000 02-APR-81 20 DEVELOPMENT MANUFACTURING UNIT

7698 BLAKE MANAGER 7839

30 9850 1000 01-MAY-79 30 MAINTAINANCE MAIN BLOCK

7368 FORD SUPERVIS 7366

20 800 0 17-DEC-80 20 DEVELOPMENT MANUFACTURING UNIT

7599 ALLEY SALESMAN 7698

30 1600 300 20-FEB-81 30 MAINTAINANCE MAIN BLOCK

7421 DRANK CLERCK 7698

30 1250 500 22-JAN-82 30 MAINTAINANCE MAIN BLOCK

9 rows selected.

**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>

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

EMPNAME SAL

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

JONES 5975

BLAKE 9850

SQL>

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

SQL> > SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNAME='JONES');

SP2-0734: unknown command beginning "> SELECT \*..." - rest of line ignored.

SQL>

SQL> SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNAME='JONES');

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7566 JONES MANAGER 7839

20 5975 1000 02-APR-81

7698 BLAKE MANAGER 7839

30 9850 1000 01-MAY-79

SQL>

**Problem 4.4: Issue a query to display information about employees who earn more** **than any employee in dept 30.**

SQL>

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

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7839 CLARK CEO

10 9900 16-MAR-72

SQL>

**Problem 4.5: Display the employees who have the same job as jones and whose** **salary >= fords.**

SQL>

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

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7566 JONES MANAGER 7839

20 5975 1000 02-APR-81

7698 BLAKE MANAGER 7839

30 9850 1000 01-MAY-79

SQL>

**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 EMPNAME,JOB FROM EMP WHERE DEPTNO=20 AND JOB=ANY(SELECT JOB FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPTNO AND D.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>

SQL> SELECT \* FROM EMP OUTER WHERE SAL>(SELECT AVG(SAL) FROM EMP WHERE DEPTNO=OUTER.DEPTNO);

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7566 JONES MANAGER 7839

20 5975 1000 02-APR-81

7698 BLAKE MANAGER 7839

30 9850 1000 01-MAY-79

7839 CLARK CEO

10 9900 16-MAR-72

SQL>

**Problem 4.8: Write a query that would display the empname, job where each** **employee works and the name of their dept.**

SQL>

SQL> SELECT EMPNAME,JOB,DNAME FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPT;

SELECT EMPNAME,JOB,DNAME FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPT

\*

ERROR at line 1:

ORA-00904: "D"."DEPT": invalid identifier

SQL> SELECT EMPNAME,JOB,DNAME FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPTNO;

EMPNAME 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>

SQL> SELECT \* FROM EMP WHERE JOB=ANY(SELECT E.JOB FROM DEPT D,EMP E WHERE D.DEPTNO=E.DEPTNO AND DLOC='MAIN BLOCK') AND DEPTNO!=(SELECT DEPTNO FROM DEPT WHERE DLOC='MAIN BLOCK');

SELECT \* FROM EMP WHERE JOB=ANY(SELECT E.JOB FROM DEPT D,EMP E WHERE D.DEPTNO=E.DEPTNO AND DLOC='MAIN BLOCK') AND DEPTNO!=(SELECT DEPTNO FROM DEPT WHERE DLOC='MAIN BLOCK')

\*

ERROR at line 1:

ORA-01427: single-row subquery returns more than one row

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

SQL>

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

no rows selected

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

SQL>

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

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7368 FORD SUPERVIS 7366

20 800 0 17-DEC-80

SQL>

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

SQL>

SQL> SELECT DNAME FROM DEPT WHERE DEPTNO=ANY(SELECT DEPTNO FROM (SELECT COUNT(JOB) AS NO,DEPTNO FROM EMP WHERE JOB='SALESMAN' GROUP BY DEPTNO) WHERE NO>=2);

DNAME

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

MAINTAINANCE

**Problem 4.13: Write a query to list the employees in dept 20 with the same job as** **anyone in dept 30.**

SQL>

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

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7399 ASANT SALESMAN 7566 20 1600 300 20-FEB-81

7566 JONES MANAGER 7839 20 5975 1000 02-APR-81

SQL>

**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>

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

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7839 CLARK CEO 10 9900

16-MAR-72

7698 BLAKE MANAGER 7839 30 9850 1000 01-MAY-79

SQL>

**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

SQL>

**Problem 4.16: Display the maximum salaries of the departments whose** **minimum salary is greater than 1000 and lesser than 5000.**

SQL>

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

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

\*

ERROR at line 1:

ORA-00904: "EMPE": invalid identifier

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

MAX(SAL)

----------

1250

1250

1600

1600

3000

1600

6 rows selected.

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

SQL>

SQL> SELECT A.DNAME FROM DEPT D,ACCDEPT A WHERE D.DEPTNO=A.DEPTNO;

no rows selected

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

SQL>

SQL> SELECT EMPNAME FROM EMP WHERE DEPTNO!=ANY(SELECT DEPTNO FROM ACCDEPT);

no rows selected

**LEFTOUT-JOIN**

~~~~~~~~~~~~

**Problem 4.19: Display all the employees and the departments implementing a left outer join.**

SQL>

SQL> SELECT \* FROM EMP LEFT JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB DEPTNO DNAME DLOC

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

7611 SCOTT HOD 7839 10 3000

12-JUN-76 10 MANAGEMENT MAIN BLOCK

7839 CLARK CEO 10 9900

16-MAR-72 10 MANAGEMENT MAIN BLOCK

7369 SMITH CLERK 7566 20 800 0 17-DEC-80 20 DEVELOPMENT MANUFACTURING UNIT

7399 ASANT SALESMAN 7566 20 1600 300 20-FEB-81 20 DEVELOPMENT MANUFACTURING UNIT

7566 JONES MANAGER 7839 20 5975 1000 02-APR-81 20 DEVELOPMENT MANUFACTURING UNIT

7368 FORD SUPERVIS 7366 20 800 0 17-DEC-80 20 DEVELOPMENT MANUFACTURING UNIT

7499 ALLEN SALESMAN 7698 30 1600 300 20-FEB-81 30 MAINTAINANCE MAIN BLOCK

7521 WARD SALESMAN 7698 30 1250 500 22-FEB-82 30 MAINTAINANCE MAIN BLOCK

7698 BLAKE MANAGER 7839 30 9850 1000 01-MAY-79 30 MAINTAINANCE MAIN BLOCK

7599 ALLEY SALESMAN 7698 30 1600 300 20-FEB-81 30 MAINTAINANCE MAIN BLOCK

7421 DRANK CLERCK 7698 30 1250 500 22-JAN-82 30 MAINTAINANCE MAIN BLOCK

11 rows selected.

**RIGHTOUTER-JOIN**

**~~~~~~~~~~~~~~~**

**Problem 4.20: Display the employee name and department name in which they are working implementing a right outer join.**

SQL>

SQL> SELECT \* FROM EMP RIGHT JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB DEPTNO DNAME DLOC

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

7369 SMITH CLERK 7566 20 800 0 17-DEC-80 20 DEVELOPMENT MANUFACTURING UNIT

7399 ASANT SALESMAN 7566 20 1600 300 20-FEB-81 20 DEVELOPMENT MANUFACTURING UNIT

7499 ALLEN SALESMAN 7698 30 1600 300 20-FEB-81 30 MAINTAINANCE MAIN BLOCK

7521 WARD SALESMAN 7698 30 1250 500 22-FEB-82 30 MAINTAINANCE MAIN BLOCK

7566 JONES MANAGER 7839 20 5975 1000 02-APR-81 20 DEVELOPMENT MANUFACTURING UNIT

7698 BLAKE MANAGER 7839 30 9850 1000 01-MAY-79 30 MAINTAINANCE MAIN BLOCK

7611 SCOTT HOD 7839 10 3000

12-JUN-76 10 MANAGEMENT MAIN BLOCK

7839 CLARK CEO 10 9900

16-MAR-72 10 MANAGEMENT MAIN BLOCK

7368 FORD SUPERVIS 7366 20 800 0 17-DEC-80 20 DEVELOPMENT MANUFACTURING UNIT

7599 ALLEY SALESMAN 7698 30 1600 300 20-FEB-81 30 MAINTAINANCE MAIN BLOCK

7421 DRANK CLERCK 7698 30 1250 500 22-JAN-82 30 MAINTAINANCE MAIN BLOCK

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB DEPTNO DNAME DLOC

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

50 SALES HEAD OFFICE

40 TRANSPORT ADMIN BLOCK

13 rows selected.

**FULLOUTER-JOIN**

**~~~~~~~~~~~~~~**

**Problem 4.21: Display the employee name and department name in which they are working implementing a full outer join.**

SQL> SELECT \* FROM EMP FULL JON DEPT ON DEPT.DEPTNO=EMP.DEPTNO;

SELECT \* FROM EMP FULL JON DEPT ON DEPT.DEPTNO=EMP.DEPTNO

\*

ERROR at line 1:

ORA-00933: SQL command not properly ended

SQL>

SQL> SELECT \* FROM EMP FULL JOIN DEPT ON DEPT.DEPTNO=EMP.DEPTNO;

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB DEPTNO DNAME DLOC

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

7369 SMITH CLERK 7566 20 800 0 17-DEC-80 20 DEVELOPMENT MANUFACTURING UNIT

7399 ASANT SALESMAN 7566 20 1600 300 20-FEB-81 20 DEVELOPMENT MANUFACTURING UNIT

7499 ALLEN SALESMAN 7698 30 1600 300 20-FEB-81 30 MAINTAINANCE MAIN BLOCK

7521 WARD SALESMAN 7698 30 1250 500 22-FEB-82 30 MAINTAINANCE MAIN BLOCK

7566 JONES MANAGER 7839 20 5975 1000 02-APR-81 20 DEVELOPMENT MANUFACTURING UNIT

7698 BLAKE MANAGER 7839 30 9850 1000 01-MAY-79 30 MAINTAINANCE MAIN BLOCK

7611 SCOTT HOD 7839 10 3000

12-JUN-76 10 MANAGEMENT MAIN BLOCK

7839 CLARK CEO 10 9900

16-MAR-72 10 MANAGEMENT MAIN BLOCK

7368 FORD SUPERVIS 7366 20 800 0 17-DEC-80 20 DEVELOPMENT MANUFACTURING UNIT

7599 ALLEY SALESMAN 7698 30 1600 300 20-FEB-81 30 MAINTAINANCE MAIN BLOCK

7421 DRANK CLERCK 7698 30 1250 500 22-JAN-82 30 MAINTAINANCE MAIN BLOCK

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB DEPTNO DNAME DLOC

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

50 SALES HEAD OFFICE

40 TRANSPORT ADMIN BLOCK

13 rows selected.

SQL>

**SELFJOIN** **~~~~~~~~~~~**

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

SQL>

SQL> SELECT E.EMPNAME,M.EMPNAME FROM EMP E,EMP M WHERE E.MGR=M.EMPNO;

EMPNAME EMPNAME

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

SMITH JONES

ASANT JONES

ALLEN BLAKE

WARD BLAKE

JONES CLARK

BLAKE CLARK

SCOTT CLARK

ALLEY BLAKE

DRANK BLAKE

9 rows selected.

**Problem 4.23: Write a query to display their employee names and their managers** **salary for every employee .**

SQL> SELECT E.EMPNAME,M.SAL FROM EMP E,EM WHERE E.MGR=M.EMPNO;

SELECT E.EMPNAME,M.SAL FROM EMP E,EM WHERE E.MGR=M.EMPNO

\*

ERROR at line 1:

ORA-00942: table or view does not exist

SQL> SELECT E.EMPNAME,M.SAL FROM EMP E,EMP M WHERE E.MGR=M.EMPNO;

EMPNAME SAL

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

SMITH 5975

ASANT 5975

ALLEN 9850

WARD 9850

JONES 9900

BLAKE 9900

SCOTT 9900

ALLEY 9850

DRANK 9850

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 E.EMPNAME,E.JOB,E.EMPNO,D.DNAME,D.DLOC FROM EMP E, DEPT D WHERE E.DEPTNO=E.DEPTN D.DEPTNO=E.DEPTNO;

SELECT E.EMPNAME,E.JOB,E.EMPNO,D.DNAME,D.DLOC FROM EMP E, DEPT D WHERE E.DEPTNO=E.DEPTN D.DEPTNO=E.DEPTNO

\*

ERROR at line 1:

ORA-00933: SQL command not properly ended

SQL>

SQL> SELECT E.EMPNAME,E.JOB,E.EMPNO,D.DNAME,D.DLOC FROM EMP E, DEPT D WHERE E.DEPTNO=E.DEPTNO AND D.DEPTNO=E.DEPTNO;

EMPNAME JOB EMPNO DNAME DLOC

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

SMITH CLERK 7369 DEVELOPMENT MANUFACTURING UNIT

ASANT SALESMAN 7399 DEVELOPMENT MANUFACTURING UNIT

ALLEN SALESMAN 7499 MAINTAINANCE MAIN BLOCK

WARD SALESMAN 7521 MAINTAINANCE MAIN BLOCK

JONES MANAGER 7566 DEVELOPMENT MANUFACTURING UNIT

BLAKE MANAGER 7698 MAINTAINANCE MAIN BLOCK

SCOTT HOD 7611 MANAGEMENT MAIN BLOCK

CLARK CEO 7839 MANAGEMENT MAIN BLOCK

FORD SUPERVIS 7368 DEVELOPMENT MANUFACTURING UNIT

ALLEY SALESMAN 7599 MAINTAINANCE MAIN BLOCK

DRANK CLERCK 7421 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>

SQL> SELECT E.EMPNO,E.EMPNAME,E.JOB,M.EMPNAME FROM EMP E,EMP M WHERE E.MGR=M.EMPNO;

EMPNO EMPNAME JOB EMPNAME

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

7369 SMITH CLERK JONES

7399 ASANT SALESMAN JONES

7499 ALLEN SALESMAN BLAKE

7521 WARD SALESMAN BLAKE

7566 JONES MANAGER CLARK

7698 BLAKE MANAGER CLARK

7611 SCOTT HOD CLARK

7599 ALLEY SALESMAN BLAKE

7421 DRANK CLERCK BLAKE

9 rows selected.

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

SQL>

SQL> SELECT E.EMPNAME,P.EMPNAME FROM EMP E,EMP P WHERE E.SAL=P.SAL AND E.EMPNAME!=P.EMPNAME;

EMPNAME EMPNAME

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

FORD SMITH

ALLEN ASANT

ALLEY ASANT

ASANT ALLEN

ALLEY ALLEN

DRANK WARD

SMITH FORD

ASANT ALLEY

ALLEN ALLEY

WARD DRANK

10 rows selected.

SQL>

SQL>

**Exercise 5**

**Problem 5.1: Display all the dept numbers available with the dept and accdept tables avoiding duplicates.**

SQL> SELECT DEPTNO FROM DEPT UNION SELECT DEPTNO FROM ACCDPT;

SELECT DEPTNO FROM DEPT UNION SELECT DEPTNO FROM ACCDPT

\*

ERROR at line 1:

ORA-00942: table or view does not exist

SQL>

SQL>

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

DEPTNO

----------

10

20

30

40

50

SQL>

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

SQL>

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

DEPTNO

----------

10

20

30

40

50

SQL>

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

SQL>

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

no rows selected

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

SQL>

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

DEPTNO

----------

10

20

30

40

50

SQL>

**Problem 5.5: The organization wants to display only the details of the employees those who are managers.( horizontal portioning)**

SQL>

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

View created.

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

SQL> SELECT \* FROM MANGERS;

SELECT \* FROM MANGERS

\*

ERROR at line 1:

ORA-00942: table or view does not exist

SQL> SELECT \* FROM MANAGERS;

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7566 JONES MANAGER 7839 20 5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30 9850 1000 01-MAY-79

SQL>

**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>

SQL>CREATE VIEW EMP\_ALL AS SELECT E.EMPNO,E.EMPNAME,D.DEPTNO,D.DNAME FROM EMP E, DEPT D WHERE E.DEPTNO=D.DEPTNO AND E.JOB NOT IN('HOD','CEO');

View created.

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

SQL> SELECT \* FROM EMP\_ALL;

EMPNO EMPNAME DEPTNO DNAME

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

7369 SMITH 20 DEVELOPMENT

7399 ASANT 20 DEVELOPMENT

7499 ALLEN 30 MAINTAINANCE

7521 WARD 30 MAINTAINANCE

7566 JONES 20 DEVELOPMENT

7698 BLAKE 30 MAINTAINANCE

7368 FORD 20 DEVELOPMENT

7599 ALLEY 30 MAINTAINANCE

7421 DRANK 30 MAINTAINANCE

9 rows selected.

SQL>

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

SQL> SELECT \* FROM MANAGERS;

EMPNO EMPNAME JOB MGR DEPTNO SAL COMMI DOB

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

7566 JONES MANAGER 7839 20 5975 1000 02-APR-81

7698 BLAKE MANAGER 7839 30 9850 1000 01-MAY-79

SQL>

**Problem 5.10: Drop a view.**

SQL> DROP VIEW EMP\_AL;

DROP VIEW EMP\_AL

\*

ERROR at line 1:

ORA-00942: table or view does not exist

SQL>

SQL> DROP VIEW EMP\_ALL;

View dropped.

SQL>

**Excersie6**

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

SQL> declare

2 a number(10);

3 b number(10);

4 begin

5 a:=&a;

6 b:=&b;

7 dbms\_output.put\_line('THE PREV VALUES OF A AND B WERE');

8 dbms\_output.put\_line(a);

9 dbms\_output.put\_line(b);

10 a:=a+b;

11 b:=a-b;

12 a:=a-b;

13 dbms\_output.put\_line('THE VALUES OF A AND B ARE');

14 dbms\_output.put\_line(a);

15 dbms\_output.put\_line(b);

16 end;

17 /

Enter value for a: 2

old 5: a:=&a;

new 5: a:=2;

Enter value for b: 4

old 6: b:=&b;

new 6: b:=4;

PL/SQL procedure successfully completed.

SQL> SET SERVEROUT ON;

SQL> declare

2 a number(10);

3 b number(10);

4 begin

5 a:=&a;

6 b:=&b;

7 dbms\_output.put\_line('THE PREV VALUES OF A AND B WERE');

8 dbms\_output.put\_line(a);

9 dbms\_output.put\_line(b);

10 a:=a+b;

11 b:=a-b;

12 a:=a-b;

13 dbms\_output.put\_line('THE VALUES OF A AND B ARE');

14 dbms\_output.put\_line(a);

15 dbms\_output.put\_line(b);

16 end;

17 /

Enter value for a: 4

old 5: a:=&a;

new 5: a:=4;

Enter value for b: 5

old 6: b:=&b;

new 6: b:=5;

THE PREV VALUES OF A AND B WERE

4

5

THE VALUES OF A AND B ARE

5

4

PL/SQL procedure successfully completed.

SQL>

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

SQL>

SQL> a number(10);

16\* end;number(10)

SQL> b number(10);

SP2-0734: unknown command beginning "b number(1..." - rest of line ignored.

SQL> c number(10);

16\* end;n

SQL> begin

2 a:=&a;

3 b:=&b;

4 dbms\_output.put\_line('THE PREV VALUES OF A AND B WERE');

5 dbms\_output.put\_line(a);

6 dbms\_output.put\_line(b);

7 c:=a;

8 a:=b;

9 b:=c;

10 dbms\_output.put\_line('THE VALUES OF A AND B ARE');

11 dbms\_output.put\_line(a);

12 dbms\_output.put\_line(b);

13 end;

14 /

Enter value for a:

old 2: a:=&a;

new 2: a:=;

Enter value for b: 5

old 3: b:=&b;

new 3: b:=5;

a:=;

\*

ERROR at line 2:

ORA-06550: line 2, column 4:

PLS-00103: Encountered the symbol ";" when expecting one of the following:

( - + case mod new not null <an identifier>

<a double-quoted delimited-identifier> <a bind variable>

continue avg count current exists max min prior sql stddev

sum variance execute forall merge time timestamp interval

date <a string literal with character set specification>

<a number> <a single-quoted SQL string> pipe

<an alternatively-quoted string literal with character set specification>

<an alternatively

SQL> declare

2 a number(10);

3 b number(10);

4 c number(10);

5 begin

6 a:=&a;

7 b:=&b;

8 dbms\_output.put\_line('THE PREV VALUES OF A AND B WERE');

9 dbms\_output.put\_line(a);

10 dbms\_output.put\_line(b);

11 c:=a;

12 a:=b;

13 b:=c;

14 dbms\_output.put\_line('THE VALUES OF A AND B ARE');

15 dbms\_output.put\_line(a);

16 dbms\_output.put\_line(b);

17 end;

18 /

Enter value for a:

old 6: a:=&a;

new 6: a:=;

Enter value for b: 5

old 7: b:=&b;

new 7: b:=5;

a:=;

\*

ERROR at line 6:

ORA-06550: line 6, column 4:

PLS-00103: Encountered the symbol ";" when expecting one of the following:

( - + case mod new not null <an identifier>

<a double-quoted delimited-identifier> <a bind variable>

continue avg count current exists max min prior sql stddev

sum variance execute forall merge time timestamp interval

date <a string literal with character set specification>

<a number> <a single-quoted SQL string> pipe

<an alternatively-quoted string literal with character set specification>

<an alternatively

SQL> declare

2 a number(10);

3 b number(10);

4 c number(10);

5 begin

6 a:=&a;

7 b:=&b;

8 dbms\_output.put\_line('THE PREV VALUES OF A AND B WERE');

9 dbms\_output.put\_line(a);

10 dbms\_output.put\_line(b);

11 c:=a;

12 a:=b;

13 b:=c;

14 dbms\_output.put\_line('THE VALUES OF A AND B ARE');

15 dbms\_output.put\_line(a);

16 dbms\_output.put\_line(b);

17 end;

18 /

Enter value for a: 74

old 6: a:=&a;

new 6: a:=74;

Enter value for b: 98

old 7: b:=&b;

new 7: b:=98;

THE PREV VALUES OF A AND B WERE

74

98

THE VALUES OF A AND B ARE

98

74

PL/SQL procedure successfully completed.

SQL>

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

SQL> declare

2 a number;

3 b number;

4 begin

5 a:=&a;

6 b:=&b;

7 if a=b then

8 dbms\_output.put\_line('BOTH ARE EQUAL');

9 elsif a>b then

10 dbms\_output.put\_line('A IS GREATER');

11 else

12 dbms\_output.put\_line('B IS GREATER');

13 end if;

14 end;

15 /

Enter value for a: 5

old 5: a:=&a;

new 5: a:=5;

Enter value for b: 7

old 6: b:=&b;

new 6: b:=7;

B IS GREATER

PL/SQL procedure successfully completed.

SQL>

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

SQL> declare

2 java number(10);

3 dbms number(10);

4 co number(10);

5 se number(10); es

6 number(10); ppl

7 number(10); total

8 number(10); avgs

9 number(10); per

10 number(10);

11 dbms\_output.put\_line('ENTER THE MARKS');

12 begin

13 java:=&java;

14 dbms:=&dbms;

15 co:=&co;

16 se:=&se;

17 es:=&es;

18 ppl:=&ppl;

19 total:=(java+dbms+co+se+es+ppl);

20 per:=(total/600)\*100;

21 if java<40 or dbms<40 or co<40 or se<40 or es<40 or ppl<40 then

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

23 elsif per>75 then

24 dbms\_output.put\_line('GRADE A');

25 elsif per>65 and per<75 then

26 dbms\_output.put\_line('GRADE B');

27 elsif per>55 and per<65 then

28 dbms\_output.put\_line('GRADE C');

29 else

30 dbms\_output.put\_line('INVALID INPUT');

31 end if;

32 dbms\_output.put\_line('PERCENTAGE IS '||per);

33 dbms\_output.put\_line('TOTAL IS '||total);

34 end;

35 /

Enter value for java: 12

old 13: java:=&java;

new 13: java:=12;

Enter value for dbms: 54

old 14: dbms:=&dbms;

new 14: dbms:=54;

Enter value for co: 46

old 15: co:=&co;

new 15: co:=46;

Enter value for se: 59

old 16: se:=&se;

new 16: se:=59;

Enter value for es: 49

old 17: es:=&es;

new 17: es:=49;

Enter value for ppl: 57

old 18: ppl:=&ppl;

new 18: ppl:=57;

dbms\_output.put\_line('ENTER THE MARKS');

\*

ERROR at line 11:

ORA-06550: line 11, column 12:

PLS-00103: Encountered the symbol "." when expecting one of the following:

constant exception <an identifier>

<a double-quoted delimited-identifier> table columns long

double ref char time timestamp interval date binary national

character nchar

The symbol "<an identifier>" was substituted for "." to continue.

SQL> declare

2 java number(10);

3 dbms number(10);

4 co number(10);

5 se number(10);

6 es number(10);

7 ppl number(10);

8 total number(10);

9 avgs number(10);

10 per number(10);

11 dbms\_output.put\_line('ENTER THE MARKS');

12 begin

13 java:=&java;

14 dbms:=&dbms;

15 co:=&co;

16 se:=&se;

17 es:=&es;

18 ppl:=&ppl;

19 total:=(java+dbms+co+se+es+ppl);

20 per:=(total/600)\*100;

21 if java<40 or dbms<40 or co<40 or se<40 or es<40 or ppl<40 then

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

23 elsif per>75 then

24 dbms\_output.put\_line('GRADE A');

25 elsif per>65 and per<75 then

26 dbms\_output.put\_line('GRADE B');

27 elsif per>55 and per<65 then

28 dbms\_output.put\_line('GRADE C');

29 else

30 dbms\_output.put\_line('INVALID INPUT');

31 end if;

32 dbms\_output.put\_line('PERCENTAGE IS '||per);

33 dbms\_output.put\_line('TOTAL IS '||total);

34 end;

35 /

Enter value for java: 54

old 13: java:=&java;

new 13: java:=54;

Enter value for dbms: 98

old 14: dbms:=&dbms;

new 14: dbms:=98;

Enter value for co: 57

old 15: co:=&co;

new 15: co:=57;

Enter value for se: 56

old 16: se:=&se;

new 16: se:=56;

Enter value for es: 52

old 17: es:=&es;

new 17: es:=52;

Enter value for ppl: 65

old 18: ppl:=&ppl;

new 18: ppl:=65;

dbms\_output.put\_line('ENTER THE MARKS');

\*

ERROR at line 11:

ORA-06550: line 11, column 12:

PLS-00103: Encountered the symbol "." when expecting one of the following:

constant exception <an identifier>

<a double-quoted delimited-identifier> table columns long

double ref char time timestamp interval date binary national

character nchar

The symbol "<an identifier>" was substituted for "." to continue.

SQL> declare

2 java number(10);

3 dbms number(10);

4 co number(10);

5 se number(10);

6 es number(10);

7 ppl number(10);

8 total number(10);

9 avgs number(10);

10 per number(10);

11 begin

12 dbms\_output.put\_line('ENTER THE MARKS');

13 java:=&java;

14 dbms:=&dbms;

15 co:=&co;

16 se:=&se;

17 es:=&es;

18 ppl:=&ppl;

19 total:=(java+dbms+co+se+es+ppl);

20 per:=(total/600)\*100;

21 if java<40 or dbms<40 or co<40 or se<40 or es<40 or ppl<40 then

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

23 elsif per>75 then

24 dbms\_output.put\_line('GRADE A');

25 elsif per>65 and per<75 then

26 dbms\_output.put\_line('GRADE B');

27 elsif per>55 and per<65 then

28 dbms\_output.put\_line('GRADE C');

29 else

30 dbms\_output.put\_line('INVALID INPUT');

31 end if;

32 dbms\_output.put\_line('PERCENTAGE IS '||per);

33 dbms\_output.put\_line('TOTAL IS '||total);

34 end;

35 /

Enter value for java: 54

old 13: java:=&java;

new 13: java:=54;

Enter value for dbms: 47

old 14: dbms:=&dbms;

new 14: dbms:=47;

Enter value for co: 46

old 15: co:=&co;

new 15: co:=46;

Enter value for se: 65

old 16: se:=&se;

new 16: se:=65;

Enter value for es: 42

old 17: es:=&es;

new 17: es:=42;

Enter value for ppl: 44

old 18: ppl:=&ppl;

new 18: ppl:=44;

ENTER THE MARKS

INVALID INPUT

PERCENTAGE IS 50

TOTAL IS 298

PL/SQL procedure successfully completed.

SQL>

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

SQL> declare

2 a number;

3 d number:=0;

4 sum1 number:=0;

5 begin

6 a:=&a;

7 while a>0

8 loop

9 d:=mod(a,10);

10 sum1:=sum1+d;

11 a:=trunc(a/10);

12 end loop;

13 dbms\_output.put\_line('SUM = '|| sum1);

14 end;

15 /

Enter value for a: declare

old 6: a:=&a;

new 6: a:=declare;

a:=declare;

\*

ERROR at line 6:

ORA-06550: line 6, column 4:

PLS-00103: Encountered the symbol "DECLARE" when expecting one of the following:

( - + case mod new not null <an identifier>

<a double-quoted delimited-identifier> <a bind variable>

continue avg count current exists max min prior sql stddev

sum variance execute forall merge time timestamp interval

date <a string literal with character set specification>

<a number> <a single-quoted SQL string> pipe

<an alternatively-quoted string literal with character set specification>

<an alterna

SQL> a number;

6\* a:=&a;number

SQL> d number:=0;

SP2-0734: unknown command beginning "d number:=..." - rest of line ignored.

SQL> sum1 number:=0;

SP2-0734: unknown command beginning "sum1 numbe..." - rest of line ignored.

SQL> begin

2 a:=&a;

3 while a>0

4 loop

5 d:=mod(a,10);

6 sum1:=sum1+d;

7 a:=trunc(a/10);

8 end loop;

9 dbms\_output.put\_line('SUM = '|| sum1);

10 end;

11 /

Enter value for a: 546

old 2: a:=&a;

new 2: a:=546;

a:=546;

\*

ERROR at line 2:

ORA-06550: line 2, column 1:

PLS-00201: identifier 'A' must be declared

ORA-06550: line 2, column 1:

PL/SQL: Statement ignored

ORA-06550: line 3, column 7:

PLS-00201: identifier 'A' must be declared

ORA-06550: line 3, column 1:

PL/SQL: Statement ignored

ORA-06550: line 9, column 33:

PLS-00201: identifier 'SUM1' must be declared

ORA-06550: line 9, column 1:

PL/SQL: Statement ignored

SQL> declare

2 a number;

3 d number:=0;

4 sum1 number:=0;

5 begin

6 a:=&a;

7 while a>0

8 loop

9 d:=mod(a,10);

10 sum1:=sum1+d;

11 a:=trunc(a/10);

12 end loop;

13 dbms\_output.put\_line('SUM = '|| sum1);

14 end;

15 /

Enter value for a: 546

old 6: a:=&a;

new 6: a:=546;

SUM = 15

PL/SQL procedure successfully completed.

SQL>

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

SQL> declare

2 a number;

3 rev number;

4 d number;

5 begin

6 a:=&a;

7 rev:=0;

8 while a>0

9 loop

10 d:=mod(a,10);

11 rev:=(rev\*10)+d;

12 a:=trunc(a/10);

13 end loop;

14 dbms\_output.put\_line('REVERSE NUMBER = '|| rev);

15 end;

16 /

Enter value for a: 56

old 6: a:=&a;

new 6: a:=56;

REVERSE NUMBER = 65

PL/SQL procedure successfully completed.

SQL>

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

SQL> declare

2 a number;

3 c number:=0;

4 i number;

5 begin

6 a:=&a;

7 for i in 1..a

8 loop

9 if mod(a,i)=0 then

10 c:=c+1;

11 end if;

12 end loop;

13 if c=2 then

14 dbms\_output.put\_line(a ||' is a prime number');

15 else

16 dbms\_output.put\_line(a ||' is not a prime number');

17 end if;

18 end;

19 /

Enter value for a: 45

old 6: a:=&a;

new 6: a:=45;

45 is not a prime number

PL/SQL procedure successfully completed.

SQL>

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

SQL> declare

2 n number;

3 f number:=1;

4 begin

5 n:=&n;

6 for i in 1..n

7 loop

8 f:=f\*i;

9 end loop;

10 dbms\_output.put\_line('Factorial '|| n ||' is '|| f);

11 end;

12 /

Enter value for n: 8

old 5: n:=&n;

new 5: n:=8;

Factorial 8 is 40320

PL/SQL procedure successfully completed.

SQL>

**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.**

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

Table created.

SQL>

SQL> declare

2 pi constant number(4,2):=3.14;

3 radius number(5):=3;

4 area number(6,2);

5 begin

6 while radius<7 loop

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

8 insert into areas values(radius,area);

9 radius:=radius+1;

10 end loop;

11 end;

12 /

PL/SQL procedure successfully completed.

SQL> SELECT \* FROM area;

SELECT \* FROM area

\*

ERROR at line 1:

ORA-04044: procedure, function, package, or type is not allowed here

SQL> SELECT \* FROM AREAS;

RADIUS AREA

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

3 28.26

4 50.24

5 78.5

6 113.04

3 28.26

4 50.24

5 78.5

6 113.04

8 rows selected.

**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> declare

2 mano number(5);

3 mcb number(6,2);

4 minibal constant number(7,2):=1000.00;

5 fine number(6,2):=100.00;

6 begin

7 mano:=&mano;

8 select cur\_bal into mcb from acct where acctno=mano;

9 if mcb<minibal then

10 update acct set cur\_bal=cur\_bal-fine where acctno=mano;

11 end if;

12 end;

13 /

Enter value for mano: 98

old 7: mano:=&mano;

new 7: mano:=98;

declare

\*

ERROR at line 1:

ORA-01403: no data found

ORA-06512: at line 8

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

insert into stud values('&sname',&rollno,&marks);

select \* from acct;

SQL>declare

2 mano number(5);

3 mcb number(6,2);

4 minibal constant number(7,2):=1000.00;

5 fine number(6,2):=100.00;

6 begin

7 mano:=&mano;

8 select cur\_bal into mcb from acct where acctno=mano;

9 if mcb<minibal then

10 update acct set cur\_bal=cur\_bal-fine where acctno=mano;

11 end if;

12 end;

13 /

**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**

**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 EP SET SAL=SAL+1000 WHERE SAL>5000 AND DEPTNO=DEPTID;

4 END;

5 /

Warning: Procedure created with compilation errors.

SQL>

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

2 BEGIN

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

4 END;

5 /

Warning: Procedure created with compilation errors.

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.

**Program7.2 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>

SQL> EXE SALARY1(7369);

SP2-0734: unknown command beginning "EXE SALARY..." - rest of line ignored.

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.

**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 LOP

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

6 END;

7

8

9

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.

**Program7.4 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.

**Program7.5 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**

**Program8.1 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 /

Warning: Trigger created with compilation errors.

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.

**Excercise8.2 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.

**Exercise 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>

SQL> SELECT \* FROM LG;

SELECT \* FROM LG

\*

ERROR at line 1:

ORA-00942: table or view does not exist

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 EMO = 7369;

DELETE FROM EMP WHERE EMO = 7369

\*

ERROR at line 1:

ORA-00904: "EMO": invalid identifier

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

**--THANKYOU--**

**--END--**