EXERCISE.1

CREATE TABLE EMP (EMPNO INTEGER PRIMARY KEY,ENAME VARCHAR(20) NOT NULL,JOB VARCHAR(20) NOT NULL,MGR INTEGER,DEPTNO INTEGER,SAL INTEGER);

DESC EMP;

ALTER TABLE EMP ADD COMM INTEGER ;

DESC EMP;

ALTER TABLE EMP MODIFY JOB VARCHAR(30);

DESC EMP;

CREATE TABLE DEPT(DEPTNO INTEGER PRIMARY KEY,DNAME VARCHAR(20),LOC VARCHAR(40));

DESC DEPT;

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

DESC EMP;

ALTER TABLE EMP ADD CHECK (EMPNO>100);

ALTER TABLE EMP modify sal integer default 5000;

ALTER TABLE EMP ADD DOB VARCHAR(10);

DESC EMP;

EXERCISE.2

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

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

INSERT INTO DEPT VALUES(30, 'MAINTAINANCE','UNIT MAN BLOCK');

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

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

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7369,'SMITH','CLERK',7566,'17-DEC80',800,0,20);

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

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7499,'ALLEN','SALESMAN',7698,'20-FEB81',1600,300,30);

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7521,'WARD','SALESMAN',7698,'22-FEB82',1250,500,30);

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7566,'JONES','MANAGER',7839,'02-APR81',5975,500,20);

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7698,'BLAKE','MANAGER',7839,'01-MAY79',9850,1400,30);

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL , DEPTNO) VALUES(7611,'SCOTT','HOD',7839,'12-JUN76',3000,10);

INSERT INTO EMP(EMPNO, ENAME ,JOB ,DOB ,SAL , DEPTNO) VALUES(7839,'CLARK','CEO','16-MAR72',9900,10);

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7368,'FORD','SUPERVIS',7366,'17-DEC80',800,0,20);

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7599,'ALLEY','SALESMAN',7698,'20-FEB81',1600,300,30);

INSERT INTO EMP(EMPNO, ENAME ,JOB, MGR ,DOB ,SAL ,COMM, DEPTNO) VALUES(7421,'DRANK','CLERCK',7698,'22-JAN82',1250,500,30);

UPDATE EMP SET COMM=1000 WHERE JOB='MANAGER';

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));

INSERT INTO EMPLOYEE SELECT\*FROM EMP;

DELETE FROM EMPLOYEE WHERE JOB='SUPERVIS';

DELETE FROM EMPLOYEE WHERE EMPNO=7599;

SELECT \* FROM EMP ORDER BY SAL;

SELECT \* FROM EMP ORDER BY SAL DESC;

SELECT \* FROM EMP WHERE DEPTNO=30;

SELECT DISTINCT DEPTNO FROM EMP;

SELECT \* FROM EMP ORDER BY ENAME;

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

select \* from EMP where COMM=NULL ;

select ENAME,DNAME from EMP,DEPT where EMP.DEPTNO=DEPT.DEPTNO ;

EXERCISE.3

select \* from EMP where DEPTNO in(7369,7499);

select \* from EMPLOYEE where ENAME like "S%";

select \* from EMPLOYEE where ENAME not like "S%";

select \* from EMPLOYEE where EMPNO between 7500 and 7600 ;

Select \* from EMPLOYEE where EMPNO not between 7500 and 7600 ;

select sqrt(SAL) from EMP;

SELECT COUNT(\*) FROM EMP;

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

select min(SAL) "MIN\_SAL", MAX(SAL) "MAX\_SAL" from EMP;

SELECT SUM(SAL) FROM EMP;

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

select to\_date(DOB,'DD-MM-YY') from EMP;

select add\_months(DOB,2) from EMP;

select last\_day(’05-oct-09’) from dual;

select round(to\_date(dob),’month’) from emp;

select round(to\_date(dob),’year’) from emp;

select round(to\_date(dob),’day’) from emp;

select(sysdate-60) from dual;\*/

select ENAME ,SAL , SAL+0.15\* SAL from EMP;

select ENAME from EMP where ENAME like 'B%' or ENAME like 'C%';

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

select dname, count (ename) from emp, dept where emp.deptno=dept.deptno group by

dname

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

select ename from emp where mgr in(7602,7566,7789);

select count (distinct job) from emp;

select max(sal)-min(sal) from emp;

select count(distinct deptno) from emp;

select empname , dob from emp where to\_char (dob,'MON')='FEB';

select ENAME from EMP where ENAME LIKE ('S%') and ENAME LIKE('%H');

select ename from emp where sal>5000 or sal>6000;

EXERCISE.4

select ENAME,DNAME from EMP,DEPT where DNAME='MAINTAINANCE' OR DNAME='DEVELOPMENT' ;

SELECT ename FROM emp WHERE sal >(SELECT MIN(saL)FROM emp) AND JOB LIKE ('M%');

SELECT ename FROM EMP WHERE job =( SELECT job FROM emp WHERE eNAME='JONES');

SELECT \* FROM emp WHERE sal >ANY( SELECT sal FROM emp WHERE DEPTNO=30 );

SELECT \* FROM EMP WHERE job =( SELECT job FROM emp WHERE eNAME='JONES') AND SAL>=( SELECT sal FROM emp WHERE ENAME='FORD');

SELECT ename, job FROM emp WHERE DEPTNO=10 AND JOB IN(SELECT JOB FROM emp,dept WHERE EMP.DEPTNO=DEPT.DEPTNO AND Dname='MANAGEMENT');

SELECT \* FROM emp WHERE sal >(SELECT AVG(SAL)FROM emp);

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

SELECT \* FROM EMP WHERE job in (SELECT job FROM emp,dept WHERE emp.deptno=dept.deptno and LOC='MAIN BLOCK');

SELECT \* FROM emp WHERE DEPTNO=10 AND JOB IN(SELECT JOB FROM emp,dept WHERE EMP.DEPTNO=DEPT.DEPTNO AND Dname='development');

SELECT \* FROM EMP WHERE job =( SELECT job FROM emp WHERE eNAME='FORD') AND SAL=( SELECT SAL FROM emp WHERE eNAME='FORD');

SELECT \* FROM emp WHERE deptno=20 and job=ANY( SELECT job FROM emp WHERE DEPTNO=30 );

SELECT eNAME FROM emp WHERE sal >ANY( SELECT sal FROM emp WHERE DEPTNO IN (20,30));

select ename,dname from emp left join dept on emp.deptno=dept.deptno;

select ename,dname from emp right join dept on emp.deptno=dept.deptno;

select ename,dname from emp full outer join dept on emp.deptno=dept.deptno;

select ename,job,dname,loc from emp natural join dept;

EXERCISE.5

select deptno from dept union select deptno from accdept;

select deptno from dept union all select deptno from accdept;

select deptno from dept intersect select deptno from accdept;

select deptno from dept minus select deptno from accdept;

create view managers as select \* from employee where job='manager';

create view emps as select empno,ename,employee.deptno,dept.dname from employee,dept where employee.deptno=dept.deptno;

create view emps2 as select empno,ename,employee.deptno,dept.dname from employee,dept where employee.deptno=dept.deptno and job not in ('hod','ceo');

drop view managers;

EXERCISE.6

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

declare

a number(10);

b number(10);

begin

a:=&a;

b:=&b;

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

dbms\_output.put\_line(a);

dbms\_output.put\_line(b);

a:=a+b;

b:=a-b;

a:=a-b;

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

dbms\_output.put\_line(a);

dbms\_output.put\_line(b);

end;

OUTPUT:

SQL> @ SWAPPING.SQL

17 /

Enter value for a: 5

old 5: a:=&a;

new 5: a:=5;

Enter value for b: 3

old 6: b:=&b;

new 6: b:=3;

THE PREV VALUES OF A AND B WERE

5

3

THE VALUES OF A AND B ARE

3

5

PL/SQL procedure successfully completed.

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

declare

a number(10);

b number(10);

c number(10);

begin

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

dbms\_output.put\_line(a);

dbms\_output.put\_line(b);

a:=&a;

b:=&b;

c:=a;

a:=b;

b:=c;

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

dbms\_output.put\_line(a);

dbms\_output.put\_line(b);

end;

OUTPUT:

SQL> @ SWAPPING2.SQL

19 /

Enter value for a: 5

old 6: a:=&a;

new 6: a:=5;

Enter value for b: 3

old 7: b:=&b;

new 7: b:=3;

THE PREV VALUES OF A AND B WERE

5

3

THE VALUES OF A AND B ARE

3

5

PL/SQL procedure successfully completed.

Program 6.3:

Write a pl/sql program to find the largest of two numbers

declare

a number;

b number;

begin

a:=&a;

b:=&b;

if a=b then

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

elsif a>b then

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

else

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

end if;

end;

OUTPUT:

SQL> @ GREATESTOF2.sql

13 /

Enter value for a: 5

old 5: a:=&a;

new 5: a:=5;

Enter value for b: 2

old 6: b:=&b;

new 6: b:=2;

A IS GREATER

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

declare

java number(10);

dbms number(10);

co number(10);

se number(10); es

number(10); ppl

number(10); total

number(10); avgs

number(10); per

number(10);

begin

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

java:=&java;

dbms:=&dbms;

co:=&co;

se:=&se;

es:=&es;

ppl:=&ppl;

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

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

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

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

if per>75 then

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

elsif per>65 and per<75 then

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

elsif per>55 and per<65 then

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

else

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

end if;

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

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

end;

OUTPUT:

SQL> @ GRADE.sql

31 /

Enter value for java: 80

old 12: java:=&java;

new 12: java:=80;

Enter value for dbms: 70

old 13: dbms:=&dbms;

new 13: dbms:=70;

Enter value for co: 89

old 14: co:=&co;

new 14: co:=89;

Enter value for se: 72

old 15: se:=&se;

new 15: se:=72;

Enter value for es: 76

old 16: es:=&es;

new 16: es:=76;

Enter value for ppl: 71

old 17: ppl:=&ppl;

new 17: ppl:=71;

GRADE A

PERCENTAGE IS 76

TOTAL IS 458

PL/SQL procedure successfully completed.

Program 6.5:

Write a pl/sql program to find the sum of digits in a given number

declare

a number;

d number:=0;

sum1 number:=0;

begin

a:=&a;

while a>0

loop

d:=mod(a,10);

sum1:=sum1+d;

a:=trunc(a/10);

end loop;

dbms\_output.put\_line('sum is'|| sum1);

end;

OUTPUT:

SQL> @ SUMOFDIGITS.sql

16 /

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

declare

a number;

rev number;

d number;

begin

a:=&a;

rev:=0;

while a>0

loop

d:=mod(a,10);

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

a:=trunc(a/10);

end loop;

dbms\_output.put\_line('no is'|| rev);

end;

OUTPUT:

SQL> @ REVERSE2.sql

16 /

Enter value for a: 536

old 6: a:=&a;

new 6: a:=536;

no is635

PL/SQL procedure successfully completed.

Program 6.7:

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

declare

a number;

c number:=0;

i number;

begin

a:=&a;

for i in 1..a

loop

if mod(a,i)=0 then

c:=c+1;

end if;

end loop;

if c=2 then

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

else

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

end if;

end;

OUTPUT:

SQL> @ PRIME.SQL

19 /

Enter value for a: 11

old 6: a:=&a;

new 6: a:=11;

11is a prime number

PL/SQL procedure successfully completed.

Program 6.8:

Write a pl/sql program to find the factorial of a given number

declare

n number;

f number:=1;

begin

n:=&n;

for i in 1..n

loop

f:=f\*i;

end loop;

dbms\_output.put\_line('the factorial is'|| f);

end;

OUTPUT:

SQL> @ FACTORIAL.sql

12 /

Enter value for n: 5

old 5: n:=&n;

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 & area

TABLE NAME:AREAS

RADIUS AREA

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

Table created.

--PROGRAM

declare

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

radius number(5):=3;

area number(6,2);

begin

while radius<7 loop

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

insert into areas values(radius,area);

radius:=radius+1;

end loop;

end;

OUTPUT:

SQL> @ AREAOFCIRCLE.SQL

13 /

PL/SQL procedure successfully completed.

SQL> SELECT \* FROM AREAS;

RADIUS AREA

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

3 28.26

4 50.24

5 78.5

6 113.04

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));

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

SQL> select \* from acct;

ACCTNO NAME CUR\_BAL

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

777 sirius 10000

765 john 1000

855 sam 500

353 peter 800

--PROGRAM

declare

mano number(5);

mcb number(6,2);

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

fine number(6,2):=100.00;

begin

mano:=&mano;

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

if mcb<minibal then

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

end if;

end;

OUTPUT:

SQL> @ BANKACC.sql

13 /

Enter value for mano: 855

old 7: mano:=&mano;

new 7: mano:=855;

PL/SQL procedure successfully completed.

EXERCISE.7

A) Addition of two numbers

1 create or replace procedure addition(a number,b number) is

2 c number;

3 begin

4 c:=a+b;

5 dbms\_output.put\_line('the addition is'||c);

6\* end;

SQL> /

Procedure created.

SQL> exec addition(50,51);

the addition is101

B)To display a string using procedure

SQL> create or replace procedure hello(name varchar)

2 is

3 begin

4 dbms\_output.put\_line('hi hello'||name);

5 end;

6 /

SQL> exec hello('nivetha');

hi hellonivetha

PL/SQL procedure successfully completed.

SQL> exec hello(' nivetha');

hi hello nivetha

SQL> create or replace procedure hello

2 is

3 begin

4 dbms\_output.put\_line('Hi HELLO');

5\* end;

SQL> /

Procedure created.

SQL> exec hello

Hi HELLO

EXERCISE.8

SQL> create table employee(eno number(10),ename varchar(20),salary •

number(20));

Table created.

SQL> desc employee;

Name Null? Type

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

ENO NUMBER(10)

ENAME VARCHAR2(20)

SALARY NUMBER(20)

A.CREATE A TRIGGER TO CONVERT LOWERCASE TO UPPERCASE

SQL> create or replace trigger uppercase

2 before insert or update on employee

3 referencing new as n for each row

4 begin

5 :n.ename:=upper(:n.ename);

6 end;

7 /

Trigger created.

SQL> insert into employee values(11,'ravi',20000);

1 row created.

SQL> select \* from employee;

ENO ENAME SALARY

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

11 RAVI 20000

B.CREATE A TRIGGER TO CHECK THE SALARY IS ABOVE 1000

SQL> create or replace trigger salcondition

2 before insert or update on employee

3 referencing new as n for each row

4 begin

5 if(:n.salary<1000)then

6 raise\_application\_error(-20001,'salary must be greater than 1000');

7 end if;

8 end;

9 /

Trigger created.

SQL> insert into employee values(12,'ram',999);

insert into employee values(12,'ram',999)

\*

ERROR at line 1:

ORA-20001: salary must be greater than 1000

ORA-06512: at "SHAR.SALCONDITION", line 3

ORA-04088: error during execution of trigger 'SHAR.SALCONDITION'

SQL> insert into employee values(12,'ram',1001);

1 row created .

SQL> select \* from employee;

ENO ENAME SALARY

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

11 RAVI 20000

12 RAM 1001

C.CREATE A TRIGGER TO AVOID DELETION ON WEDNESDAY

SQL> create or replace trigger datedelete

2 before delete on employee

3 declare

4 Date1 char(5);

5 begin

6 Date1:=to\_char(sysdate,'dy');

7 if date1 in ('wed','WED')then

8 raise\_application\_error(-20002,'records cannot be deleted');

9 end if;

10 end;

11 /

Trigger created.

SQL> delete from employee where eno=11;

delete from employee where eno=11

ERROR at line 1:

ORA-20002: records cannot be deleted

ORA-06512: at "SHAR.DATEDELETE", line 6

ORA-04088: error during execution of trigger 'SHAR.DATEDELETE'

D.CREATE A TRIGGER TO AVOID DELETION OF PARTICULAR DATE

SQL> create or replace trigger namedelete

2 before delete on employee

3 referencing new as n for each row

4 begin

5 if rtrim(:n.ename)in('ravi','RAVI')then

6 raise\_application\_error(-20003,'record cannot be deleted for ravi');

7 end if;

8 end;

9 /

Trigger created.

SQL> delete from employee where ename='RAVI';

delete from employee where ename='RAVI'

\*

ERROR at line 1:

ORA-20002: records cannot be deleted for ravi

ORA-06512: at "SHAR.DATEDELETE", line 6

ORA-04088: error during execution of trigger 'SHAR.DATEDELETE'

E.CREATE A TRIGGER TO AVOID INSERTION OF PARTICULAR DATA

SQL> create or replace trigger inserteno

2 before insert on employee

3 referencing new as n for each row

4 begin

5 if(:n.eno=14)then

6 raise\_application\_error(-20003,'cannot insert this eno');

7 end if;

8\* end;

SQL> /

Trigger created.

SQL> insert into employee values(14,'seetha',20000);

insert into employee values(14,'seetha',20000)

\*

ERROR at line 1:

ORA-20003: cannot insert this eno

ORA-06512: at "SHAR.INSERTENO", line 3

ORA-04088: error during execution of trigger 'SHAR.INSERTENO'