205119045

Database Management Systems Lab

1. Data Definition Language (DDL) Commands

Problem 1.1:

create table EMP(EMPNO NUMBER(6),ENAME VARCHAR2(20) not null,JOB VARCHAR2(10) not null,MGR NUMBER(4),DEPTNO NUMBER(3),SAL NUMBER(7,2),constraint pk_emp primary key(empno));

Problem 1.2:

alter table EMP add COMMISSION NUMBER(7,2);

Problem 1.3:

alter table EMP modify(JOB varchar2(20));

Problem 1.4:

create table dept(DEPTNO NUMBER(2),DNAME VARCHAR2(10) not null,LOC VARCHAR2(10),constraint pk_dept primary key(deptno));

Problem 1.5:

alter table EMP add constraint fk_empdept foreign key(DEPTNO) references dept(DEPTNO);

Problem 1.6:

alter table EMP add constraint c1 check (EMPNO>100);

Problem 1.7:

alter table EMP modify SAL default 5000;

Problem 1.8:

alter table EMP add Dob DATE;

2. Data Manipulation Language (DDL) Commands

Problem 2.1:

```
alter table dept modify(DNAME varchar2(20));
alter table dept modify(LOC varchar2(20));
insert into dept values(10,'MANAGEMENT','MAIN BLOCK');
insert into dept values(20,'DEVELOPMENT','MANUFACTURING UNIT');
insert into dept values(30,'MAINTAINANCE','MAIN BLOCK');
insert into dept values(40,'TRANSPORT','ADMIN BLOCK');
insert into dept values(50,'SALES','HEAD OFFICE');
```

Problem 2.2:

```
insert into EMP values(7369,'SMITH','CLERK',7566,20,800,0,'17-DEC-1980');
insert into EMP values(7399,'ASANT','SALESMAN',7566,20,1600,300,'20-FEB-1981');
insert into EMP values(7499,'ALLEN','SALESMAN',7698,30,1600,300,'20-FEB-1981');
insert into EMP values(7521,'WARD','SALESMAN',7698,30,1250,500,'22-FEB-1982');
insert into EMP values(7566,'JONES','MANAGER',7839,20,5975,500,'02-APR-1981');
insert into EMP values(7698,'BLAKE','MANAGER',7839,30,9850,1400,'01-MAY-1979');
insert into EMP values(7611,'SCOTT','HOD',7839, 10,3000,NULL,'12-JUN-1976');
insert into EMP values(7839,'CLARK','CEO',NULL ,10,9900,NULL,'16-MAR-1972');
insert into EMP values(7368,'FORD','SUPERVIS',7366,20,800,0,'17-DEC-1980');
insert into EMP values(7599,'ALLEY','SALESMAN',7698,30,1600,300,'20-FEB-1981');
insert into EMP values(7421,'DRANK','CLERCK',7698,30,1250,500,'22-JAN-1982');
```

Problem 2.3:

```
update EMP set COMMISSION = 1000 where JOB='MANAGER';
```

Problem 2.4:

```
create table employee as select * from EMP;
```

Problem 2.5:

delete from employee where JOB='SUPERVISOR';

Problem 2.6:

delete from employee where EMPNO=7599;

Problem 2.7:

select * from employee order by SAL;

Problem 2.8:

select * from employee order by SAL desc;

Problem 2.9:

select * from employee where DEPTNO=30;

Problem 2.10:

select distinct DEPTNO from employee;

Problem 2.11:

select * from EMP order by ename;

Problem 2.12:

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

Problem 2.13:

select * from EMP where COMMISSION is null;

Problem 2.14:

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

3. In – Built Functions

Problem 3.1:

select * from emp where deptno=7369 or deptno=7499;

Problem 3.2:

select * from emp where substr(ename,1,1) in ('S');

Problem 3.3:

select * from emp where substr(ename,1,1) not in ('S');

Problem 3.4:

select * from emp where empno between 7500 and 7600;

Problem 3.5:

select * from emp where empno not between 7500 and 7600;

Problem 3.6:

select sqrt(sal) from emp;

Problem 3.7:

select count(*) from emp;

Problem 3.8:

select sum(sal),avg(sal) from emp;

Problem 3.9:

select max(sal) as max_salary,min(sal) as min_salary from emp;

Problem 3.10:

select sum(sal) from emp;

Problem 3.11:

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

Problem 3.12:

select to_char(to_date('14-jul-09'),'month') from dual;

Problem 3.13:

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

Problem 3.14:

select add_months(dob,2) from emp;

Problem 3.15:

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

Problem 3.16:

select to_char(round(to_date(dob),'day'),'yyyy-mm-dd') from emp;

Problem 3.17:

select (sysdate-60) from dual;

Problem 3.18:

select ename,sal,0.15*sal as raise from emp;

Problem 3.19:

select * from emp where substr(ename,1,1) in ('B','C');

Problem 3.20:

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

Problem 3.21:

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

Problem 3.22:

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

Problem 3.23:

select ename,mgr from emp where mgr in(77499,7566,7611);

Problem 3.24:

select count(distinct(job)) from emp;

Problem 3.25:

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

Problem 3.26:

select count(distinct(deptno)) from emp;

Problem 3.27:

select ename,dob from emp where to_char(dob,'MM') in ('02');

Problem 3.28:

select ename from emp where to_char(dob,'MM') in (extract(month from sysdate));

Problem 3.29:

select ename from emp where ename like 'S%H';

Problem 3.30:

select ename from emp where sal>6000;

4. Nested Queries and Joins

Problem 4.1:

select ENAME,DNAME from EMP,DEPT where emp.deptno=dept.deptno and (DNAME='MAINTAINANCE' OR DNAME='DEVELOPMENT');

Problem 4.2:

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

Problem 4.3:

select ename from emp where job=(select job from emp where ename='JONES') and ename not in ('JONES');

Problem 4.4:

select * from emp where sal>(select max(sal) from emp where deptno=30);

Problem 4.5:

select ename from emp where job=(select job from emp where ename='JONES') and sal>=(select sal from emp where ename='FORD') and ename not in ('JONES');

Problem 4.6:

select ename,job from emp where deptno=20 and job in(select job from dept,emp where dept.deptno=emp.deptno and dname = 'management');

Problem 4.7:

select ename,deptno,sal from emp e1 where sal > (select avg(sal) from emp e2 where e1.deptno=e2.deptno);

Problem 4.8:

select ename,job,dname from emp,dept where emp.deptno=dept.deptno;

Problem 4.9:

select ename from emp where job in(select job from emp,dept where emp.deptno=dept.deptno and loc='MAIN BLOCK') and deptno not in (select deptno from dept where loc='MAIN BLOCK');

Problem 4.10:

select ename from emp where deptno=10 and job in(select job from emp,dept where emp.deptno=dept.deptno and dname='DEVELOPMENT');

Problem 4.11:

select ename from emp where job=(select job from emp where ename='FORD') and sal=(select sal from emp where ename='FORD') and ename not in ('FORD');

Problem 4.12:

select dname from dept where (select count(*) from emp where job='SALESMAN' and dept.deptno=emp.deptno) >= 2;

Problem 4.13:

select ename from emp where deptno=20 and job in(select job from emp where deptno=30);

Problem 4.14:

select ename from emp where sal>(select max(sal) from emp where deptno=20 or deptno=30);

Problem 4.15:

select max(sal),dname from emp,dept where emp.deptno=dept.deptno and sal > 9000 group by dname;

Problem 4.16:

select max(sal),dname from emp,dept where emp.deptno=dept.deptno having min(sal)>1000 and min(sal)<5000 group by dname;

JOINS

Problem 4.17:

create table accdept as select * from dept where deptno in (10,20,30);

select dept.dname from dept,accdept where dept.deptno=accdept.deptno;

Problem 4.18:

select ename from emp where deptno in (select deptno from dept where dname not in (select dept.dname from dept,accdept where dept.deptno=accdept.deptno));

Problem 4.19:

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

Problem 4.20:

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

Problem 4.21:

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

Problem 4.22:

select a.ename as employee,b.ename as manager from emp a,emp b where a.mgr=b.empno;

Problem 4.23:

select a.ename as employee,b.sal as manager_salary from emp a,emp b where a.mgr=b.empno;

Problem 4.24:

select ename,job,empno,dname,loc from emp,dept where emp.deptno=dept.deptno;

Problem 4.25:

select a.empno,a.ename as employee,a.job,b.ename as manager from emp a,emp b where a.mgr=b.empno;

Problem 4.26:

select ename from emp where sal in (select sal from emp group by sal having count(*)>1);

5. SET Operators and Views

Problem 5.1:

select deptno from dept union select deptno from accdept;

Problem 5.2:

select deptno from dept union all select deptno from accdept;

Problem 5.3:

select deptno from dept intersect select deptno from accdept;

Problem 5.4:

select deptno from dept minus select deptno from accdept;

Problem 5.5:

create view MANAGERS as select * from emp where job='MANAGER';

Problem 5.6:

create view GENERALS as select emp.empno, emp.ename, emp.deptno, dept.dname from emp join dept on dept.deptno=emp.deptno;

Problem 5.7:

create view Al as select emp.empno, emp.ename, emp.deptno, dept.dname from emp join dept on emp.deptno=dept.deptno where emp.job<>'HOD' and emp.job<>'CEO';

Problem 5.8:

select view_name from user_view;

Problem 5.9:

insert into managers values(3311, 'Ankita', 'MANAGER', 6678, 20, 3222, 1000, '03-APR-81');

Problem 5.10:

drop view Al;

6. Control Structures

Problem 6.1:

Write a PL/SQL program to swap two numbers without 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

```
Enter value for a: 5
Enter value for b: 3
THE PREV VALUES OF A AND B WERE
 5
 3
THE VALUES OF A AND B ARE
 3
 5
```

Problem 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

```
Enter value for a: 5
Enter value for b: 3
THE PREVIOUS VALUES OF A AND B WERE
 15
 10
THE VALUES OF A AND B ARE
 10
 15
```

Problem 6.3:

Write a PL/SQL program to find 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

```
Enter value for a: 5
Enter value for b: 2
A IS GREATER
```

Problem 6.4:

Write a PL/SQL program to find total and average of 6 subjects and display the grade

```
declare
daa number(10);
dbms number(10);
os number(10);
rmt number(10);
python number(10);
c number(10);
total number(10);
per number(10);
begin
    dbms_output.put_line('ENTER THE MARKS');
    daa:=&daa;
    dbms:=&dbms;
    os:=&os;
    rmt:=&rmt;
    python:=&python;
    c:=&c;
    total:=(daa+dbms+os+rmt+python+c);
    per:=(total/600)*100;
    if daa<40 or dbms<40 or os<40 or rmt<40 or python<40 or c<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

```
Enter value for daa: 75
Enter value for dbms: 89
Enter value for os: 91
Enter value for rmt: 65
Enter value for python: 70
Enter value for c: 61
GRADE A
PERCENTAGE IS 75
TOTAL IS 451
```

Problem 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

```
 Enter value for a: 121
sum is:- 4
```

Problem 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

```
Enter value for a: 121
No is:- 635
```

Problem 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

```
Enter value for a: 11
11 is a prime number
```

Problem 6.8:

Write a PL/SQL program to find factorial of 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

```
Enter value for n: 6
The factorial is 720
```

Problem 6.9:

Write a PL/SQL program to calculate the area of circle for a value of radius varying from 3 to 7

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

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

```
SELECT * FROM AREAS;
RADIUS AREA
---------- ----------
    3       28.26
    4       50.24
    5       78.5
    6       113.04
```

Problem 6.10:

Write a PL/SQL program 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.

- ✓ create table acct(name varchar2(10),cur_bal number(10),
  acctno number(6,2));

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

7. Procedures and Functions

Problem 7.1:

```
Create or replace procedure salary(deptid number) as
    begin
       update emp set sal=sal+1000 where sal>5000 AND deptno=deptid;
    end;
```

Problem 7.2:

```
Create or replace procedure salary1(empid number) as
    begin
       update emp set sal=sal+sal*(0.1) where empno=empid;
    end;
```

Problem 7.3:

```
Create or replace procedure get_sal(dept number) as
    begin
        for s in (select * from emp where deptno = dept)
        loop
          dbms_output.put_line(s.sal);
        end loop;
     end;
```

Problem 7.4:

```
Create or replace procedure get_nature(dept number) as
    begin
      for s in (select * from emp where deptno = dept)
      loop
         dbms_output.put_line(s.job);
       end loop;
```

```
end;
```

Problem 7.5:

```
Create or replace procedure dep_name(deptid number)  as
    begin
      select dept.dname from dept,emp where emp.deptno=dept.deptno;
    end;
```

8. Triggers

Problem 8.1:

Write a trigger to ensure that DEPT TABLE does not contain duplicate of null values in DEPTNO column

```
CREATE OR REPLACE TRIGGER first
BEFORE INSERT on DEPT
for each row
DECLARE
   a number;
BEGIN
    if (:new.deptno is NULL)  then
       raise_application_error(-20001,'error::deptno cannot be NULL');
    else
       select count(*) into a from DEPT where deptno=:new.deptno;
        if(a=1)  then
          raise_application_error(-20002,'error::cannot have duplicate value);
        end if;
     end if;
  END;
```

Problem 8.2:

Write a trigger to carry out the following action: on deleting a deptno from DEPT table, all the records with that DEPTNO has to be deleted from the EMP table

```
CREATE OR REPLACE TRIGGER trgr_to_delete
BEFORE DELETE ON DEPT FOR EACH ROW
DECLARE
    CURSOR get_emp( p_deptno NUMBER ) IS
     select EMPNO, ENAME , JOB, MGR, SAL, COMM, DOB
     from EMP
    WHERE deptno=p_deptno;
```

```
BEGIN
        dbms_output.put_line( 'Delete dept = ' || :old.deptno );
        dbms_output.put_line( '- dept name = ' || :old.dname );
        dbms_output.put_line( '- dept loc  = ' || :old.loc );
   FOR get_emp_rec IN get_emp( :old.deptno ) LOOP
        dbms_output.put( '- emp ( ' || get_emp_rec.empno );
        dbms_output.put( ', ' || get_emp_rec.ename );
        dbms_output.put( ', ' || get_emp_rec.job );

        dbms_output.put( ', ' || get_emp_rec.mgr );
        dbms_output.put( ', ' || get_emp_rec.sal );
        dbms_output.put( ', ' || get_emp_rec.comm );

        dbms_output.put( ', ' || get_emp_rec.job );
       dbms_output.put_line( ' )' );
  END LOOP;
END;
```

Problem 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.

```
CREATE TRIGGER AfterDELETETrigger on [EMP Table]
FOR DELETE
AS
INSERT INTO [logtable](
             empno
            ,[ename]
            ,[job]
            ,[mgr]
            ,[deptno]
            ,[sal]
            ,[dob])
 SELECT  empno
              ,[ename]
              ,[job]
              ,[mgr]
              ,[deptno]
              ,[sal]
            ,[dob]
        ,CAST( SERVERPROPERTY('MachineName') AS VARCHAR2(50))
        ,CAST( SERVERPROPERTY('ServerName') AS VARCHAR2(50))
        ,GETDATE()
FROM DELETED;
PRINT 'We Successfully Fired the AFTER DELETE Triggers in SQL Server and inserted a values into log Table.'
GO
```