SQL DATABASE RELATED

QUESTION-2

EmployeeMaster(Emp_1. No, Name, Job, Join_Date,
Salary, commission, dept_no)

```
create table employeemaster
(
emp_no number(3) primary key,
name varchar2(10),   job
varchar2(10),  join_date date,
salary number(10),
commission number(10),
dept_no number(3) references department(dept_no)
);
```

employeemaster:

```
insert into employeemaster values (101,'abhi','clerk','17-dec-12',20000,null,1);
insert into employeemaster values (102,'ravi','operator','20-nov-90',30000,500,2);
insert into employeemaster values (103,'raju','salesman','18-oct-96',40000,null,3);
insert into employeemaster values (104,'rahi','manager','11-sep-99',20000,null,4);
insert into employeemaster values (105,'lee','clerk','14-aug-01',10000,200,1); insert
into employeemaster values (106,'liba','manager','13-jul-07',30000,null,2); insert
into employeemaster values
(107,'abhijit','assistant','11-mar98',50000,500,3); insert into employeemaster
values (108,'lisa','clerk','10-feb-7',10000,100,4); insert into employeemaster
values (109,'liya','dba','16-jan-09',10000,null,null);
```

2. Department(Dept_No, Dept_name, Location)

```
create table department
(
dept_no number(3) primary key,    dept_name
varchar2(10),
location varchar2(10)
);
```

department:

```
insert into department values (1,'account','varacha');
insert into department values (2,'computer','katargam');
insert into department values (3,'sales','vesu'); insert
into department values (4,'research','kamrej'); insert
into department values (5,'commerce','vesu');
```

Quries:

1.Select the name and salary of all employee who are "Manager".

ANS:

select name,salary from employeemaster where job='manager';

OUTPUT:

name salary
---------- ----------- rahi
20000
liba 30000

2. list the employees whose salary are less than 5000.

ANS:

select name from employeemaster where salary<5000;

OUTPUT:

no rows selected

3. list of employee whose commission is greater than their salary.

ANS:

select name from employeemaster where commission>salary;

OUTPUT:

no rows selected

4. list of employee who do not get commission.

ANS:

select name from employeemaster where commission=0;

OUTPUT:

no rows selected

5. list the name of employees whose Join date is month of December.

ANS:

select name from employeemaster where to_char(join_date,'mon')='dec';

OUTPUT: name
---------- abhi

6. find the employee name whose name length is 6 character.

ANS:

select name from employeemaster where length(name)='7';

OUTPUT:

name
---------- abhijit

7. find the total number of clerk in department number=5.

ANS:

```
select job ,count(job)"NO"  from employeemaster e,department d  where job='clerk' and e.dept_no=d.dept_no group by job;
```

OUTPUT:

```
NO       job
--------  -------- 3
clerk
```

QUESTION-3

Tables:

1. Product(pid,pname,ptype,price,qty)

```
create table product
(
pid number(5) primary key,
pname varchar2(10),
ptype varchar2(10),    price
number(5),  qty number(5)
);
```

Product:

```
insert into product values (101,'xyz','a',500,50);
insert into product values (102,'abc','b',400,40);
insert into product values (103,'pqr','a',250,55);  insert
into product values (104,'mno','c',400,60);
insert into product values (105,'mla','d',550,45);
```

2. Salesman(sid,name,area,DOJ,target)

```
create table salesman
(
sid number(5) primary key,
name varchar2(10),    area
varchar2(10),    doj date,
target number(5)
);
```

salesman:

```
insert into salesman values (1,'aneri','varachha','25-feb-2015',50);  insert
into salesman values (2,'vijay','katargam','17-feb-2018',30);  insert into
salesman values (3,'mihita','punagam','05-mar-2012',50);  insert into
salesman values (4,'rudra','vesu','15-jan-2017',55);
```

```
insert into salesman values (5,'hiten','vesu','08-apr-2015',40);
```

3. Sales(pid,sid,saledate,qty)

```
create table sales
(
pid number(5) references product(pid),   sid
number(5) references salesman(sid),
salesdate date,  qty
number(5)
);
```

sales:

```
insert into sales values (101,5,'25-feb-2022',40);
insert into sales values (102,4,'17-mar-2021',50);
insert into sales values (103,3,'05-jan-2022',50);  insert
into sales values (104,2,'15-apr-2020',45);  insert into
sales values (105,1,'10-jul-2020',55);
```

Constraints:

1. pid and sid must be declared as primary key while creating table.Create appropriate relationship between tables.
2. Qty field must not null.

Queries:

1. Display total qty sold of product “xyz”.

ANS:

```
select qty from product where pname='xyz';
```

OUTPUT:

```
 QTY
---------- 50
```

2. Count total products sale by each salesman.

ANS:

```
select count(target) total_target from salesman;
```

OUTPUT:

```
total_target
---------------------          5
```

3.Display salesman who sold more products than given target.

ANS:

```
select s.name  from salesman s,sales e where e.qty>s.target and s.sid=e.sid;
```

OUTPUT:

```
name
---------- aneri
vijay
```

4.Display salesman who have joined in current month.

ANS:

```
select name,doj from salesman where to_char(doj,'mon')='feb';
```

OUTPUT:

```
name       doj
---------- --------- aneri
25-feb-15  vijay
17-feb-18
```

5. Display salesman who have joined in current month.

ANS:

```
select name,doj from salesman where to_char(doj,'mon')='feb';
```

```
name       doj
---------- --------- aneri
25-FEB-15  vijay
17-FEB-18
```

6. Count total products of each type.

ANS:

```
select count(ptype) from product;
```

OUTPUT:

```
count(ptype)
--------------------          5
```

7. Display salesman whose name contain more than 5 charcters and joined in current year.

ANS:

```
select name,doj from salesman where length(name)>='5'and to_char(doj,'yyyy')=2023;
```

OUTPUT:

```
no rows selected
```

9. Remove the column area from salesman table.

ANS:

alter table salesman drop column area;

OUTPUT:

sid name doj target --------
---------- ---------- ---------- ----------
1 aneri 25-feb-15 50
2 vijay 17-feb-18 30
3 mihita 05-mar-12 50 4 rudra 15-jan-17 55
5 hiten 08-apr-15 4 QUESTION-5

Constraints:
1. transtype:Issue,Deposit

Tables:
1.Bookmaster(bookid,title,author,publisher,price,noofcopy)

```
create table bookmaster
(
bookid number(3) primary key,
title varchar2(20),
author varchar2(20),
publisher varchar2(20),
price number(5),
noofcopy number(5)
);
```

bookmaster:

```
insert into bookmaster values (101,'soft_eng','meet','vijay',1000,90);
insert into bookmaster values (102,'eng','kevin','tata_mcgrawhill',250,40);
insert into bookmaster values (103,'c++','rudra','leeya',600,80); insert
into bookmaster values (104,'let_us_c','hiten','hiren',900,100); insert into
bookmaster values (105,'php','vinas','dharm',500,40); insert into
bookmaster values (106,'engn','meet','vijay',1000,90); insert into
bookmaster values (107,'soft_eng','hiten','vijay',1000,90);
```

2. Studentmaster(studid,name,class,mobile,email)

```
create table studentmaster
(
studid number(3) primary key,
name varchar2(20), class
varchar2(5), mobile
number(10),
email varchar2(20)
);
```

studentmaster:

```
insert into studentmaster values (101,'abhi','fybca',9345685216,'abhi01@yahoo.com');  insert
into studentmaster values (171,'meet','fybba',9345681234,'meet25@gmail.com');  insert into
studentmaster values (172,'aneri','sybca',9123485214,'vijay17@gmail.com');  insert into
studentmaster values (138,'neel','sybba',7845685215,'neel15@yahoo.com');  insert into
studentmaster values (315,'hiren','tybca',9159685852,'hiten@gmail.com');
```

3. Transaction(transid,studid,bookid,transtype,transdate)

```
create table transaction
(
transid number(3) primary key,    studid number(3)
references studentmaster(studid),    bookid
number(3) references bookmaster(bookid),
transtype    varchar2(10),
transdate                date
check(transtype="issue"
or transtype="deposit")
);
```

transaction:

```
insert into transaction values (111,101,101,'issue','12-jan-14');
insert into transaction values (112,171,102,'deposit','15-feb-16');
insert into transaction values (122,172,103,'deposit','05-mar-15');
insert into transaction values (222,138,104,'issue','21-feb-15');
insert into transaction values (113,315,105,'deposit','07-jan-18');
insert into transaction values (133,101,101,'issue','01-feb-15'); insert
into transaction values (333,171,102,'issue','10-mar-16');  insert into
transaction values (114,172,101,'deposit','25-jan-14');  insert into
transaction values (144,138,104,'issue','15-feb-15');  insert into
transaction values (444,315,105,'issue','30-jan-16');
```

Queries:

1. Display all student information from studentmaster.

ANS:

```
select * from studentmaster;
```

OUTPUT:

```
studid      name        class      mobile               email
---------- ----------- ----- -------- -- ----------- ------------------------------ 101
abhi            fybca     9345685216      abhi01@yahoo.com
171       aneri          fybba     9345681234      meet25@gmail.com
172       vijay          sybca     9123485214      vijay17@gmail.com
138       neel           sybba     7845685215      neel15@yahoo.com
315       hiren          tybca     9159685852      hiten@gmail.com
```

2. Display all student information who has “yahoo.com” in email.

ANS:

```
select * from studentmaster where email like '%yahoo.com';
```

OUTPUT:

studid	name	class	mobile	email
101	abhi	fybca	9345685216	abhi01@yahoo.com
138	neel	sybba	7845685215	neel15@yahoo.com

3. Display the book of “Tata McGrawhill” publication.

ANS:

```
select title from bookmaster where publisher='tata_mcgrawhill';
```

OUTPUT:

title
---------- eng

4. Display the total number of book issue in year 2015 of title “Let Us C”.

ANS:

select count(t.bookid) total_booid from transaction t,bookmaster m where m.title='let_us_c' and m.bookid=t.bookid and to_char(transdate,'YY')=15;

OUTPUT:

total_bookid
---------- 2

5. Display the students whose name starts with “A”.

ANS:

select name from studentmaster where name like 'a%';

OUTPUT:

name
---------- abhi

6. Display details of books issued by students of class “FYBCA”.

ANS:

select * from transaction t,studentmaster s where s.class='fybca' and s.studid=t.studid;

OUTPUT:

t_id	s_id	bookid	t_type	transdate	studid	name	class	mobile	email
111	101	101	issue	12-JAN-14	101	abhi	fybca	9345685216	abhi01@yahoo.com
133	101	101	issue	01-FEB-15	101	abhi	fybca	9345685216	abhi01@yahoo.com

7. Display total number of books of each author in sorted order.

ANS:

select author,count(title) NO_OF_BOOKS from bookmaster group by author order by count(title);

OUTPUT:
author NO_OF_BOOKS

---

rudra 1 vinas
1 kevin 1 meet
2 hiten 2

8. Display students who have not issued any book yet.

ANS:

```
select s.name from transaction t,studentmaster s where t.transtype='null' and s.studid=t.studid;
```

OUTPUT:

no rows selected

9. Display the student who issue maximum book of Title “Software Engineering”.

ANS:

```
select s.name from studentmaster s,transaction t,bookmaster b where s.studid=t.studid and b.bookid=t.bookid and
t.transtype='issue' and b.title in (select max(title) from bookmaster where title='soft_eng' );
```

OUTPUT:

name
---
abhi abhi

10. Display the students whose mobile number starts with “78”.

ANS:

```
select name from studentmaster where mobile like '78%';
```

OUTPUT:

name
---
neel

PL/SQL PROGRAMS

5. Write Pl/sql Program to accept user salary & CALCULATION OF NET SALARY and Display it.

Da =15% HRA =41%
Basic salary <3000 Pf = 5 %
Basic salary <=5000 Pf = 7 %
Basic salary <=8000 Pf = 8 %
Basic salary >=8000 Pf = 10 %
Net salary:=basic + da + hra –pf

ANS:

```
set serveroutput on;
declare     basic number;
da number;     hra
number;     pf number;
netsalary number; begin
basic:=&basic;
da:=basic * (30/100);
hra:=basic * (10/100);
if (basic < 8000)     then
       pf:=basic * (8/100);
       elsif (basic >= 8000 and basic <= 16000)
then
       pf:=basic * (10/100);

end if;
       netsalary:=basic + da + hra -pf;

dbms_output.put_line('Providend Fund : ' || pf);
dbms_output.put_line('Net salary : ' || netsalary); end;
/
```

OUTPUT:

```
Enter value for basic: 8000
old   8:     basic:=&basic; new
8:     basic:=8000; Providend
Fund : 800
Net salary : 10400

PL/SQL procedure successfully completed.
```

6. Write pl/sql Program to accept 3 numbers from user and Display MAXIMUM OF 3 NUMBERS.

ANS:

```
set serveroutput on;
DECLARE
            a NUMBER := 46;
            b NUMBER := 67;
            c NUMBER := 21;
BEGIN
```

```
        IF a > b
        AND a > c THEN
        dbms_output.Put_line('Greatest number is '||a);
        ELSIF b > a
        AND b > c THEN
        dbms_output.Put_line('Greatest number is '||b);
        ELSE
        dbms_output.Put_line('Greatest number is '||c);
        END IF;
END;
/
```

OUTPUT:

Greatest number is 67

PL/SQL procedure successfully completed.

9. Write pl/sql Program find out whether the given number is a prime number or not

ANS:

```
set serveroutput on;
declare     num
number(05);     p
number(02) := 0; begin
num := &num;     for i
in 2..num-1 loop
        if mod(num,i) = 0 then
p := 1;
        end if;
end loop;     if
p = 0 then
        dbms_output.put_line(num||' is prime number.');
else
        dbms_output.put_line(num||' is not prime number.');
end if; end;
```

```
/
```

OUTPUT:

```
Enter value for num: 11
old 5:      num := &num;
new 5:      num := 11; 11
is prime number.

PL/SQL procedure successfully completed.
```

10.Write pl/sql Program to Display area of circle from 10 numbers (o/p : 4,16,36,64,100)

ANS:

```
SET SERVEROUTPUT ON;
declare
            pi constant number(4,2):=3.14;
radius number(5);               area
number(14,2);
begin
            radius:=1;
            dbms_output.put_line('radius    area');
while radius<=10
            loop
            area:=pi*power(radius,2);
            dbms_output.put_line(radius||'        '||area);
radius:=radius+1;               end loop; end;
/
```

OUTPUT:

```
radius    area -------
-------
1         3.14
2         12.56
3         28.26
4         50.24
5         78.5
6         113.04 7         153.86
8          200.96
```

```
9       254.34
10      314

PL/SQL procedure successfully completed.
```

14. create table emp (no,name.salary) with input 5 records in table to design pl/sql Program that accept emp no from user and check whether salary greater than 5000 then increment 1000 rs in emp table.

ANS:

```
create table emp
(
no      number(05),
name varchar(15),
salary number(05)
);

insert into emp values(1,'Krina',10000);
insert into emp values(2,'Shruti',3000); insert
into emp values(3,'Rutvik',5000); insert into
emp values(4,'Niyati',8000); insert into emp
values(5,'Parth',13000); insert into emp
values(6,'ansh',18000);
insert into emp values (7,'priya',4500);

set serveroutput on; declare
eno emp.no%type;    esalary
emp.salary%type;        begin
eno := &eno;
    select salary into esalary from emp where eno = no;
if esalary < 5000 then
        update emp set salary = salary + 1000 where eno = no;
dbms_output.put_line('Salary is updated.');      end if;
end;
/
```

OUTPUT:

```
Enter value for eno: 7
old 5:      eno := &eno;
new 5:      eno := 7;
Salary is updated.

PL/SQL procedure successfully completed.
```

15.Write a pl/sql code block to calculate the area of a circle for a value of radius from 1 to 7.Store the radius and the corresponding values of calculated area in an empty table named areas consist of two columns radius and area.

ANS:

```
create table areas
(
radius number(4,2), area
number(6,2)
);

set serveroutput on; declare
    a areas%rowtype; begin
    for i in 1..7 loop
a.radius := i;
        a.area := 2 * 3.14 * i;          insert into
areas(radius,area) values(a.radius,a.area);      end loop;
end;
/
```

OUTPUT:

```
select * from areas;

   radius       area ----------
----------
1       6.28
2       12.56         3      18.84
4       25.12
5       31.4
6       37.68
7       43.96

7 rows selected
```

16.Write pl/sql code block accept empno from user and display record of particular employee details. Emp(empno,name,salary)

ANS:

```
set serveroutput on; declare
    tblemp emp%rowtype; begin
```

```
    tblemp.no := &no;
    select * into tblemp from emp where no = tblemp.no;
    dbms_output.put_line('No = '||tblemp.no||'     Name = '||tblemp.name||'    Salary = '||tblemp.salary); end;
/

create table emp
(
no      number(05),
name varchar(15),
salary number(05)
);

insert into emp values(1,'Krina',10000);
insert into emp values(2,'Shruti',3000); insert
into emp values(3,'Rutvik',5000); insert into
emp values(4,'Niyati',8000); insert into emp
values(5,'Parth',13000); insert into emp
values(6,'ansh',18000);
insert into emp values(7,'priya',1000);
```

OUTPUT:

```
Enter value for no: 3 old   4:
tblemp.no := &no; new    4:
tblemp.no := 3;
No = 3     Name = Rutvik    Salary = 5000

PL/SQL procedure successfully completed.
```

18. Wrte a pl/sql code block that will accept an employee number from the user and deduct salary by rs 500 from the inputted employee if employee has a salary is less than 5000 after salary is deducted then display user defined message ‘salary less then 5000’ also display message if salary is greater than 5000. Table Emp(empno,name,salary).

ANS:

```
set serveroutput on; declare
    esalary  emp.salary%type;
eno    emp.no%type;    begin
eno := &eno;
    select salary into esalary from emp where eno = no;
esalary := esalary - 500;      if esalary < 5000 then
        dbms_output.put_line('Salary is less then 5000.');
else
```

```
        dbms_output.put_line('Salary is greater then 5000.');
end if; end;
/
```

OUTPUT:

```
Enter value for eno: 2
old 5:      eno := &eno;
new 5:      eno := 2;
Salary is less then 5000.

PL/SQL procedure successfully completed.
```

PLSQL CURSOR RELATED PROGRAMS

1.Write a program display all department records.department(d_no,d_name,D_location).

Table:department1

```
create table department1
(
d_no number(5) primary key,
d_name varchar2(10),  d_location
varchar2(10)
);
```

department1:

```
insert into department1 values (1,'meet','surat');
insert into department1 values (2,'vijay','vapi'); insert
into department1 values (3,'kevin','valsad');  insert
into department1 values (4,'rudra','surat');
insert into department1 values (5,'ansh','amreli');

declare
  cursor mycursor is select * from department1;
  rec mycursor%rowtype;
begin    for rec in
mycursor
  loop       dbms_output.put_line('d_no:'||rec.d_no);
dbms_output.put_line('d_name:'||rec.d_name);
```

```
dbms_output.put_line('d_location:'||rec.d_location);
end loop;
end;
/
```

OUTPUT:

```
d_no:1
d_name:meet
d_location:surat
d_no:2 d_name:vijay
d_location:vapi
d_no:3
d_name:kevin
d_location:valsad
d_no:4
d_name:rudra
d_location:surat
d_no:5  d_name:ansh
d_location:amreli
```

4. Write a program to display top 3 highest salary employee.Table: emp(eno,ename,salary)

ANS:

```
create table emp3
(
eno number(5) primary key,   ename varchar2(10),  salary
number(5)
);

emp3:  insert into emp3
values(1,'meet',90000); insert into emp3
values(2,'vijay',80000);  insert into
emp3 values(3,'kevin',70000);  insert
into emp3 values(4,'aneri',85000);
insert into emp3 values(5,'ansh',75000);

select * from emp3;

declare
  cursor mycursor1 is select eno,ename,salary from emp3 order by salary desc;
rec mycursor1%rowtype;
  i number(2);
begin
  i:=1;    for rec in
mycursor1
  loop
dbms_output.put_line('no:'||rec.eno);
dbms_output.put_line('name:'||rec.ename);
```

```
dbms_output.put_line('salary:'||rec.salary);
i:=i+1;        if i=4 then              exit;
     end if;
end loop;  end;
/
```

OUTPUT:

```
no:1 name:meet
salary:90000
no:4 name:aneri
salary:85000 no:2
name:vijay
salary:80000
```

5. Write a pl/sql block that take input as account number, transaction amount,transaction type (credit, debit) from user and take action appropriate according to transaction type. Also implement user-define exception that will raise when user try to withdraw greater amount than current balance.Table: account_master(actno,curbal)

ANS:

```
create table account_master
(
actno number(5) primary key,   curbal
number(10)
);

insert into account_master values(101,80000);  insert
into account_master values(102,70000);  insert into
account_master values(103,60000);  insert into
account_master values(104,55000);  insert into
account_master values(105,40000);

declare      acctno
number(5);      bal
number(10);      trans_amt
number(5);      oper
char(1);    more_than_bal
exception;
begin
acctno:=&acctno;
trans_amt:=&trans_amt;
  oper:='&oper';
    select curbal into bal   from account_master
where actno=acctno;        if oper='w' then
if trans_amt<bal then
       update account_master set curbal=curbal-trans_amt where actno=acctno;
dbms_output.put_line('transaction complit');              else
```

```
        raise more_than_bal;
      end if;        elsif
oper='d' then
        update account_master set curbal=curbal+trans_amt where actno=acctno;
dbms_output.put_line('transaction complit');       end if;  exception
when more_than_bal then
       dbms_output.put_line('more than bal');
 when no_data_found then
       dbms_output.put_line('number not availble');  end;
/
```

OUTPUT:

```
Enter value for acctno: 101
Enter value for trans_amt: 5000

Enter value for oper: w  transaction
complit
```

6. Write a pl/sql block that will print all details of employee of given employee id. If employee is not existing then handle appropriate exception. Table: emp(eno,ename,salary)

ANS:

```
create table emp4
(
empno number(5) primary key,    name
varchar2(10),
salary number(5)
);

emp4:
insert into emp4 values (1,'aneri',80000);  insert
into emp4 values (2,'vijay',70000);  insert into
emp4 values (3,'kevin',50000);  insert into
emp4 values (4,'rudra',60000);
insert into emp4 values (5,'ansh',50000);

declare
  eno number(5);
  ename varchar2(10);
esalary number(5);
begin    eno:=&eno;
select empno,name,salary into eno,ename,esalary from emp4 where empno=eno;
dbms_output.put_line('NO:'||eno);          dbms_output.put_line('NAME:'||ename);
dbms_output.put_line('SALARY:'||esalary);

exception
  when NO_DATA_FOUND then
     dbms_output.put_line('Data not available');  end;
```

```
/
```

OUTPUT:

```
Enter value for eno: 1
NO:1
NAME:aneri
SALARY:80000
create table student1
(
rno number(3) primary key,
m1 number(5),    m2
number(5),  m3 number(5)
);

Table:result create table result
(
rno number(3) references student1(rno),    total
number(5),
per number(5)
);

declare      rno
number(3);      m1
number(5);      m2
number(5);       m3
number(5);      total
number(5);      per
number(5);  begin
rno:=&rno;
m1:=&m1;
m2:=&m2;
m3:=&m3;
total:=m1+m2+m3;
   dbms_output.put_line('TOTAL :'||total);
per:=total/3;
dbms_output.put_line('PER   :'||per);
    if(m1>=35 and m2>=35 and m3>=35) then
          dbms_output.put_line('roll no '||rno ||' is pass in All Subject');
insert into student1 values(rno,m1,m2,m3);                    insert
into result values(rno,total,per);   else      dbms_output.put_line('roll
no '||rno ||' is fail');         end if;    end;
/
```

OUTPUT:

```
Enter value for rno: 2
Enter value for m1: 80
Enter value for m2: 98
```

```
Enter value for m3: 90  TOTAL
:268 PER   :89
roll no 2 is pass in All Subject

RNO        TOTAL      PER
---------- ---------- ----------
1          285        95
2          268        89
```