-- Set - 4 : -

****

11. Write a function which will display total number of projects based on status (pass status as parameter).

Ans.

```
CREATE OR REPLACE FUNCTION SET4Q11(status1 in varchar)
return number
is
   total number;
BEGIN
   select count(*) into total from project where status = status1;
   return total;
END;
```

Output : -

```
Function created.
```

```
declare
   total number(1);
begin
   total:=SET4Q11('complete');
   dbms_output.put_line(total);
end;
```

Output : -

```
Statement processed.
2
```

12. Write a procedure that will display list of projects which is going to start today.

Ans.

```
select * from project;
update project set start_date = '13-jan-2022' where project_id = 0003;
CREATE OR REPLACE PROCEDURE set4Q12
AS
CURSOR C12 IS
  select project_id, type_of_project from project where TO_DATE(start_date, 'dd-mm-yy') =
TO_DATE(SYSDATE, 'dd-mm-yy');
  PID project.project_id%type;
  PNAME project.type_of_project%type;
BEGIN
  open C12;
    loop
      fetch C12 into PID,PNAME;
        exit when C12%notfound;
        dbms_output.put_line(pid||'-'||pname);
```

```
   end loop;
  close C12;
END;
```

Output : -

```
Procedure created.
```

```
EXECUTE set4Q12;
```

Output : -

```
Statement processed.
3-sales
```

13. Write a trigger which do not allow insertion/updation/deletion into Project table if status type is ‘pending’

Ans.

```
CREATE OR REPLACE TRIGGER CHECKSTATUS
  BEFORE INSERT or UPDATE
  on project
  FOR EACH ROW
BEGIN
  IF(:new.status='pending')THEN
    RAISE_APPLICATION_ERROR(-20000,'NOT ALLOWED SATUSE PADING');
  END IF;
END;
```

Output : -

```
Trigger created.
```

-- Set - 5 : -

****

15. Write a trigger which do not allow to insert or update student record if mobile_no length is less than 10 digits.

Ans.

```
CREATE OR REPLACE TRIGGER lingthMobile
  BEFORE insert or update
  on student
  FOR EACH ROW

DECLARE
  leng number(5);
BEGIN
  leng := length(:new.mobile);
  IF(leng != 10) THEN
    RAISE_APPLICATION_ERROR(-20000,'Mobile Number length must be 10 digit');
  END IF;
END;
```

Output : -

```
Trigger created.
```

16. Write a PL/SQL block which will count total number of student's gender wise.

Male Students: 999 students

Female Students: 999 students

Ans.

```
DECLARE
   male number(2);
   female number(2);
BEGIN
   select count(*) into male from student where gender ='male';
   dbms_output.put_line('male student is '||male);

   select count(*) into female from student where gender ='female';
   dbms_output.put_line('male student is '||female);
END;
```

Output : -

```
Statement processed.
male student is 6
male student is 4
```

-- Set - 6 : -

****

12. Write a PL/SQL block to print the following report (Symptoms wise print total number of medicine given)

Ans.

select m.mno,d.symptom1,d.symptom2,d.symptom3 from medicine m,disease d, treatment t where t.mno=m.mno and t.dname=d.dname;

```
DECLARE
   mno medicine.mno%type;
   s1 disease.symptom1%type;
   s2 disease.symptom2%type;
   s3 disease.symptom3%type;
   cursor cs is select m.mno,d.symptom1,d.symptom2,d.symptom3 from medicine m,disease d, treatment t
where t.mno=m.mno and t.dname=d.dname;
BEGIN
   dbms_output.put_line('Medicine | symptom1 | symptom2 | symptom3 ');
   OPEN cs;
      LOOP
         FETCH cs INTO mno,s1,s2,s3;
            EXIT WHEN cs%notfound;
            dbms_output.put_line(mno||' | '||s1||' | '||s2||' | '||s3);
      END LOOP;
   CLOSE cs;
END;
```

Output : -

```
Statement processed.
Medicine | symptom1   | symptom2   | symptom3
   20    | mind blind | low energy | -
   50    | cold       | fever      | hotness
   50    | cold       | fever      | low energy
   40    | cold       | fever      | cough
   10    | cold       | fever      | hotness
   30    | cold       | fever      | cough
   40    | body pain  | fever      | low energy
```

13. Write a trigger which does not allow to insert or update treatment table if AVG_CURE_TIME is less than 1.

Ans.

```
CREATE OR REPLACE TRIGGER checktime
   BEFORE INSERT OR UPDATE
   ON treatment
   FOR EACH ROW

BEGIN
   IF (:new.avg_cure_time <= 1) THEN
      RAISE_APPLICATION_ERROR(-20000,'YOU CAN NOT INSERT OR UPDATE LESS THEN 1');
   END IF;
END;
```

Output : -

```
      Trigger created.
```

-- Set - 7 : -

****

13. Write a PL/SQL procedure which will display records in the following format

Ans.

```
CREATE OR REPLACE PROCEDURE Set7q13
AS
cursor cur is select i.ino,i.iname,i.expire_date,ci.qty_p,i.price,i.shop_name from item i,cust_item ci where
i.ino=ci.ino;
   ino item.ino%type;
   iname item.iname%type;
   expire_date item.expire_date%type;
   qty_p cust_item.qty_p%type;
   price item.price%type;
   shop_name item.shop_name%type;
   total number(10);
   gtotal number(10) := 0;
BEGIN
   dbms_output.put_line('Today's date is '||sysdate());
   dbms_output.put_line('Item No'||'|'||'Item Name'||'|'||'Expire Date'||'|'||'Quantity'||'|'||'Price'||'|'||'Total
RS'||'|'||'Shop name');
   open cur;
      loop
         fetch cur into ino, iname, expire_date, qty_p, price, shop_name;
         exit when cur%notfound;
         dbms_output.put_line(ino||' | '||iname||' | '||expire_date||' | '||qty_p||' | '||price||' | '||qty_p*price||' |
'||shop_name);
         gtotal := gtotal + (qty_p*price);
      end loop;
   close cur;
   dbms_output.put_line('Grand Total Rs. '||gtotal);
END;
```

Output : -

```
Procedure created.
```

```
execute  Set7q13;
```

Output : -

```
Statement processed.
Today's date is 13-JAN-22
Item No| Item Name |Expire Date| Quantity| Price |Total RS |Shop name
101    |   fan     | 30-DEC-22 |    5    | 1500  |  7500   | vihar shop
105    |   printer | 18-MAR-30 |    2    | 150000|  300000 | croma shop
102    |   fruit   | 03-JUN-21 |    10   | 150   |  1500   | collection shop
103    |   pendrive| 10-JAN-23 |    2    | 5000  |  10000  | driven shop
104    |   laptop  | 21-NOV-25 |    1    | 51000 |  51000  | amazon shop

                                        Grand Total Rs. 370000
```

14. Write a trigger which do not allow insertion / updation / deletion of Item details on Sunday.

Ans.

```
CREATE OR REPLACE TRIGGER chkday
   BEFORE insert or update or delete
   on customer
   FOR EACH ROW

BEGIN
   if to_char(sysdate,'DAY') = 'sunday' then
      raise_application_error(-20101,'No any modify on today');
   end if;
END;
```

Output : -

```
Trigger created.
```

-- Set - 8 : -

****

6. Write a PL/SQL function which will count total number of day's horror movie last longer.

Ans.

```
CREATE OR REPLACE FUNCTION SetQ6 (movie in varchar)
RETURN number
AS
   days number(3);
BEGIN
   select (date_of_cl-date_of_arr) into days from curent where mid = (select mid from movie where mname
= movie);
   return days;
END;
```

Output : -

```
Function created.
```

```
DECLARE
   total_days number(3);
BEGIN
   total_days := SetQ6('Avengers');
   dbms_output.put_line('total number of days is '||total_days);
END;
```

Output : -

```
Statement processed.
total number of days is 16
```

7. Write a PL/SQL procedure that will display movie which is going to release today.

Ans.

```
CREATE OR REPLACE PROCEDURE SetQ7
AS
   mname movie.mname%type;
BEGIN
   select mname into mname from movie where to_char(date_of_rel) = to_char(sysdate);
   dbms_output.put_line('Today release movie is ' || mname);
END;
```

Output : -

```
Procedure created.
```

```
update movie set date_of_rel='13-jan-2022' where mid='201';
execute SetQ7;
```

Output : -

```
Statement processed.
Today release movie is Avengers
```

8. Write a trigger which will not allow to insert/update in current table if Date_of_arrival is less than date_of_closure.

Ans.

```
CREATE OR REPLACE TRIGGER checkdate
```

```
   BEFORE insert or update
   on curent
   FOR EACH ROW
BEGIN
   IF (:new.date_of_arr < :new.date_of_cl) THEN
      RAISE_APPLICATION_ERROR(-20000,'ARRIVAL DATE IS LESS THEN CLOSER DATE');
   END IF;
END;
```

Output : -

```
Trigger created.
```

-- Set - 9 : -

****

Create the database EXAM and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.
APPLICANT (AID, ANAME, ADDR, ABIRTH_DT)
ENTRANCE_TEST (ETID, ETNAME, MAX_SCORE, CUT_SCORE)
ETEST_CENTRE (ETCID, LOCATION, INCHARGE, CAPACITY)
ETEST_DETAILS (AID, ETID, ETCID, ETEST_DT, SCORE)
(This database is for a common entrance test which is being conducted at a number of centers and can be taken by an applicant on any day except holidays)
Ans.

```
create table applicant
(
  aid varchar(20) primary key,
  aname varchar(30) not null,
  addr varchar(50),
  adob date
);

create table etest
(
  etid number(5) primary key,
  etname varchar(30) not null,
  max_score number(10,2),
  cut_score number(10,2)
);

create table etcenter
(
  etcid number(5) primary key,
  location varchar(50),
  incharge varchar(50),
  capacity number(10)
);

create table etdetail
(
  aid varchar(20) references applicant(aid) ON DELETE CASCADE,
  etid number(5) references etest(etid) ON DELETE CASCADE,
  etcid number(5) references etcenter(etcid) ON DELETE CASCADE,
  etest_dt date,
  score number(10,2)
);

-------------------------------------------------------
CREATE SEQUENCE seqid
start with 1123
increment by 1
minvalue 1123
maxvalue 2000
```

```
cycle;

CREATE OR REPLACE TRIGGER chkaid
  BEFORE INSERT
  on applicant
  for EACH ROW

DECLARE
  id varchar(20);
BEGIN
  id := 'A'||:new.aid;
  :new.aid := id;
END;
```

----------INSERT QUERY----------

Applicant: -

```
insert into applicant values(seqid.nextval,'babu','sanivari','28-feb-2020');
insert into applicant values(seqid.nextval,'batli','surat','02-dec-2021');
insert into applicant values(seqid.nextval,'urvi','chok','05-jan-2020');
insert into applicant values(seqid.nextval,'rics','katargam','15-oct-2020');
insert into applicant values(seqid.nextval,'miten','goregav','10-sep-2021');
insert into applicant values(seqid.nextval,'ajay','gadhipur','01-oct-2021');
select * from applicant;
```

Etest: -

```
insert into etest values(101,'upsc',200,100);
insert into etest values(102,'gpsc',100,50);
insert into etest values(103,'jee',250,150);
insert into etest values(104,'neet',500,250);
insert into etest values(105,'account',100,100);
select * from etest;
```

Etcenter: -

```
insert into etcenter values(001,'amroli','babulal',500);
insert into etcenter values(002,'bhagal','popatlal',200);
insert into etcenter values(003,'chok','champkalal',400);
insert into etcenter values(004,'goregav','jethalal',300);
insert into etcenter values(005,'mumbai','munna bhai',150);
insert into etcenter values(006,'chambut','tiwari shet',230);

select * from etcenter;
```

Etdetail: -

```
insert into etdetail values('A1123',101,001,'11-dec-2021',100);
insert into etdetail values('A1124',103,005,'02-may-2020',200);
insert into etdetail values('A1126',104,004,'10-sep-2019',80);
insert into etdetail values('A1125',105,003, '23-mar-2017',90);
insert into etdetail values('A1127',102,002,'15-mar-2021',200);
insert into etdetail values('A1128',101,001,'15-mar-2021',100);
select * from etdetail;
```

---

1. **Modify the APPLICANT table so that every applicant id has an 'A' before its value. E.g. if value is '1123', it should become 'A1123'.**
   Display test center details where no tests were conducted.
   Display details about applicants who have the same score as that of Ajaykumar in 'ORACLE FUNDAMENTALS'.

Ans. select * from etcenter where etcid not in (select ETCID from etdetail);

ETCID	LOCATION	INCHARGE	CAPACITY
6	chambut	tiwari shet	230

select a.aname, atd.score from applicant a,etdetail atd where atd.score in (select atd.score from applicant a, etest at, etdetail atd where a.aname='ajay' and a.aid=atd.aid and at.etname='upsc') and a.aid=atd.aid;

ANAME	SCORE
babu	100
ajay	100

2. **Display details of applicants who appeared for all tests.**

Ans. select a.* from applicant a,etdetail etd,etest et where etd.aid = a.aid and et.etid = etd.etid;

AID	ANAME	ADDR	ADOB
A1123	babu	sanivari	28-FEB-20
A1124	batli	surat	02-DEC-21
A1126	rics	katargam	15-OCT-20
A1125	urvi	chok	05-JAN-20
A1127	miten	goregav	10-SEP-21
A1128	ajay	gadhipur	01-OCT-21

3. **Display those tests where no applicant has failed.**

Ans. select * from etest where etid not in (select ed.etid from etdetail ed,etest et where ed.etid=et.etid and score<cut_score);

ETID	ETNAME	MAX_SCORE	CUT_SCORE
101	upsc	200	100
102	gpsc	100	50
103	jee	250	150

4. **Display details of entrance test centers which had full attendance between 1st Oct 15 and 15th Oct 16.**

Ans. select etcid from etcenter where (etcid,capacity) in (select etcid,count(*) from etdetail where etest_dt>'15-mar-2021' and etest_dt<'15-mar-2021' group by etcid);

Output : -

```
no data found
```

5. **Display details of the applicants who scored more than the cut score in the tests they appeared in.**

Ans. select ed.etid,cut_score,score from etdetail ed, etest et where ed.etid=et.etid and score>cut_score;

ETID	CUT_SCORE	SCORE
103	150	200
102	50	200

6. **Display average and maximum score test wise of tests conducted at Mumbai.**

Ans. select max(atd.score),avg(atd.score) from etcenter atc,etdetail atd,etest at where atc.location='mumbai' and atc.etcid=atd.etcid;

MAX(ATD.SCORE)	AVG(ATD.SCORE)
200	200

7. **Display the number of applicants who have appeared for each test, test center wise.**

Ans. select ed.etid,ed.etcid,count(aid) from etdetail ed group by ed.etid,ed.etcid order by etid;

ETID	ETCID	COUNT(AID)
101	1	2
102	2	1
103	5	1
104	4	1
105	3	1

8. **Display details about test centers where no tests have been conducted.**

Ans. select * from etcenter where etcid not in (select etcid from etdetail);

ETCID	LOCATION	INCHARGE	CAPACITY
6	chambut	tiwari shet	230

9. **For tests, which have been conducted between 2-3-17 and 23-4-17, show details of the tests as well as the test centre.**

Ans. select at.*,atc.* from etest at,etcenter atc,etdetail atd where etest_dt between '2-mar-17' and '23-apr-17' and atd.etid = at.etid and atc.etcid = atd.etcid;

ETID	ETNAME	MAX_SCORE	CUT_SCORE	ETCID	LOCATION	INCHARGE	CAPACITY
105	account	100	100	3	chok	champkalal	400

10. How many applicants appeared in the 'ORACLE FUNDAMENTALS' test at Chennai in the month of February?

Ans. select count(a.aid) from applicant a,etest at,etcenter atc,etdetail atd where (a.aid,at.etid,atc.etcid) in (select aid,etid,etcid from etdetail where TO_CHAR(etest_dt,'mon') = 'feb') and at.etname='upsc' and atc.location = 'chennai' and a.aid=atd.aid;

COUNT(A.AID)
0

11. Display details about applicants who appeared for tests in the same month as the month in which they were born.

Ans. select a.* from applicant a,etdetail atd where to_char(adob,'mm') = to_char(etest_dt,'mm') and a.aid = atd.aid;

Output : -

```
no data found
```

12. Display the details about APPLICANTS who have scored the highest in each test, test centre wise.

Ans. select a.aid from applicant a,etdetail atd where (atd.etid,atd.score) in (select etid,max(score) from etdetail group by etid) and atd.aid=a.aid;

AID
A1123
A1124
A1126
A1125
A1127
A1128

13. Design a read only view, which has details about applicants and the tests that he has appeared for.

Ans. create view et_view as select a.aid,aname,et.etid,et.etname from applicant a, etdetail ed,etest et where a.aid=ed.aid and et.etid=ed.etid;

Output : -

```
View created.
```

select * from et_view;

AID	ANAME	ETID	ETNAME
A1123	babu	101	upsc
A1124	batli	103	jee
A1125	urvi	105	account
A1126	rics	104	neet
A1127	miten	102	gpsc
A1128	ajay	101	upsc

14. Write a procedure which will print maximum score centre wise.

Ans.

```
CREATE OR REPLACE PROCEDURE SetQ914
AS
   cursor cur is select etcid,max(score) from etdetail group by etcid order by etcid;
   center etcenter.etcid%type;
   score etdetail.score%type;
BEGIN
   open cur;
      loop
         fetch cur into center,score;
         exit when cur%notfound;
         dbms_output.put_line('center : '||center||', Max Score : '||score);
      end loop;
   close cur;
END;
```

Output : -

```
Procedure created.
```

```
EXECUTE SetQ914
```

Output : -

```
Statement processed.
center : 1, Max Score : 100
center : 2, Max Score : 200
center : 3, Max Score : 90
center : 4, Max Score : 80
center : 5, Max Score : 200
```

15. Write a procedure which will print details of entrance test.

Ans.

```
CREATE OR REPLACE PROCEDURE SET9Q15
AS
   cursor c is select etc.etcid,a.aid,etd.etest_dt,etd.score from applicant a,etest et,etcenter   etc,etdetail etd
where
            et.etname='SSC' and a.aid=etd.aid and et.etid=etd.etid and etc.etcid=etd.etcid;

   center etcenter.etcid%type;
   aid applicant.aid%type;
   edate etdetail.etest_dt%type;
```

```
   score etdetail.score%type;

BEGIN
    dbms_output.put_line('-----------------------------------------------------------------------------------------------
');
   open c;
      loop
         fetch c into center,aid,edate,score;
         exit when c%notfound;
         dbms_output.put_line('center name : '||center||' candidate id : '||aid||' date : '||edate||' score :
'||score);
      end loop;
   close c;
   dbms_output.put_line('-----------------------------------------------------------------------------------------------');
END;
```

Output : -

```
Procedure created.
```

```
execute SET9Q15;
```

16. Write a trigger which do not allow insertion / updation / deletion of Enterance test details on Sunday.

Ans.

```
CREATE OR REPLACE TRIGGER check_day
   BEFORE insert or update or delete
   on etdetail
   FOR EACH ROW
BEGIN
   if to_char(sysdate,'DAY') = 'sunday' then
      raise_application_error(-20101,'No any modify on today');
   end if;
END;
```

Output : -

```
Trigger created.
```

-- Set - 10 : -

****

Create the database BUS TRANSPORT and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.
CATEGORY (CAT_CODE, CATDESC)
ROUTEMASTER (ROUTENO, ORIGIN, DESTINATION, FARE, DISTANCE, CAPACITY, DAY, CAT_CODE)
TICKETHEADER (TICKETNO, DATEOFISSUE, DATEOFTRAVEL, BOAR DPLACE, ROUTENO)
TICKET DETAILS (TICKETNO, NAME, SEX, AGE, FARE)
ADD THE FOLLOWING CONSTRAINTS:

1. **DELUXE, SUPERDELUXE, SUPERFAST AND NORMAL ARE THE CATDESC**
2. **ORIGIN AND DESTINATION CANNOT BE SAME,**
3. **CAPACITY SHOULD BE>0 AND <=60**

Ans.

Category: -

```
CREATE TABLE category(
   cat_code number(5) primary key,
   catdec varchar(30) check (catdec in('deluxe','superdeluxe','superfast','normal'))
);

insert into category values (101,'normal');
insert into category values (102,'superdeluxe');
insert into category values (103,'deluxe');
insert into category values (104,'superdeluxe');
insert into category values (105,'superfast');

select * from category;
```

Routemaster: -

```
CREATE TABLE routemaster(
   routeno number(5) primary key,
   origin varchar(20),
   destination varchar(20),
   fare number(5),
   distance number(10,2),
   capacity number(10) check (capacity>0 and capacity<=60),
   day varchar(20),
   cat_code number(5) references category(cat_code)
);
alter table routemaster add constraint chk_dest check (origin!=destination);
insert into routemaster values (001,'surat','valsad',400,400,24,'monday',102);
insert into routemaster values (002,'surat','rajkot',700,470,55,'satuarday',103);
insert into routemaster values (003,'surat','navsari',600,300,56,'thursday',105);
insert into routemaster values (004,'surat','bardoli',500,500,60,'tuesday',101);
insert into routemaster values (005,'surat','bilimora',400,450,40,'sunday',104);
select * from routemaster;
```

Ticketheader: -

```
CREATE TABLE ticketheader(
   ticketno number(5)primary key,
   dateofissue date,
   dateoftravel date,
   boardplace varchar(20),
   routeno number(5) references routemaster(routeno)
);
insert into ticketheader values (1,'10-sep-2021','11-sep-2021','goregav',001);
insert into ticketheader values (2,'10-jun-2021','12-jul-2021','chok',004);
insert into ticketheader values (3,'01-may-2021','05-may-2021','amroli',005);
insert into ticketheader values (4,'12-aug-2021','15-aug-2021','surat station',003);
insert into ticketheader values (5,'29-oct-2021','30-oct-2021','amroli',005);
select * from ticketheader;
```

Ticketdetail: -

```
CREATE TABLE ticketdetail (
   ticketno number(5) references ticketheader(ticketno),
   name varchar(30),
   sex varchar(30),
   age number(3),
   fare number(5)
);
insert into ticketdetail values (1,'urvi','female',16,550);
insert into ticketdetail values (2,'rics','male',24,400);
insert into ticketdetail values (3,'miten','male',22,500);
insert into ticketdetail values (4,'janu','female',23,350);
insert into ticketdetail values (5,'devu','female',24,300);
select * from ticketdetail;
```

---

1. **Display the total number of people traveled on each ticket group by ticket no 23.**

   Ans. select count(*) from ticketdetail where ticketno = 3 group by ticketno;

COUNT(*)
1

2. **Give the total collection of fare for each route.**

   Ans. select th.routeno,sum(td.fare) from ticketdetail td,ticketheader th where td.ticketno=th.ticketno group by th.routeno;

ROUTENO	SUM(TD.FARE)
1	550
4	400
5	800
3	350

3. **Give the number of months between issue date and travel date of each ticket issued.**

Ans. select ticketno, MONTHS_BETWEEN(dateoftravel,dateofissue) from ticketheader;

TICKETNO	MONTHS_BETWEEN(DATEOFTRAVEL,DATEOFISSUE)
1	.032258064516129032258064516129032258064 5
2	1.064516129032258064516129032258064516 13
3	.129032258064516129032258064516129032258 1
4	.096774193548387096774193548387096774193 5
5	.032258064516129032258064516129032258064 5

4. **Count number of person boarding from the same place and same route.**

Ans. select count(ticketno) from ticketheader group by (boardplace,routeno) having count(ticketno)>1;

COUNT(TICKETNO)
2

5. **Display count of person who has traveled in each category.**

Ans. select count(th.ticketno) from ticketheader th,routemaster rm,category c where c.cat_code=rm.cat_code and rm.routeno=th.routeno;

COUNT(TH.TICKETNO)
5

6. **Write a trigger which allow to insert or update the bus capacity only greater than zero and less than 60.**

Ans.

```
CREATE OR REPLACE TRIGGER checkcapapcity
   BEFORE insert or update
   on routemaster
   FOR EACH ROW
BEGIN
   IF (:new.capacity<0 and :new.capacity>=60)  THEN
      RAISE_APPLICATION_ERROR(-20000,'Your capacity greater than 0-60');
   END IF;
END;
```

Output : -

```
Trigger created.
```

7. **Write a Procedure which will print tour details, a driver is going to take it. (pass route_no as parameter)**

Ans.

```
CREATE OR REPLACE PROCEDURE SetQ107 (r IN number)
AS
   cursor cur is select r.routeno,r.origin,r.destination,th.dateoftravel,r.day,r.capacity from routemaster
r,ticketheader th
   where r.routeno=r.routeno and r.routeno=th.routeno;
   routeno routemaster.routeno%type;
```

```
   origin routemaster.origin%type;
   destination routemaster.destination%type;
   day routemaster.day%type;
   capacity routemaster.capacity%type;
   dateoftravel ticketheader.dateoftravel%type;
BEGIN
   open cur;
      loop
         fetch cur into routeno,origin,destination,dateoftravel,day,capacity;
         exit when cur%notfound;
         dbms_output.put_line('Route No : '||routeno);
          dbms_output.put_line('Origin | Destination | Dateoftravel | Day | Capacity');
         dbms_output.put_line(origin||' '||destination||' '||dateoftravel||' '||day||' '||capacity);
      end loop;
   close cur;
END;
```

Output : -

```
Procedure created.
```

```
execute SetQ107(01);
```

Output : -

```
Statement processed.
Route No : 1
Origin | Destination | Dateoftravel | Day | Capacity
surat valsad 11-SEP-21 monday 24
Route No : 4
Origin | Destination | Dateoftravel | Day | Capacity
surat bardoli 12-JUL-21 tuesday 60
Route No : 5
Origin | Destination | Dateoftravel | Day | Capacity
surat bilimora 05-MAY-21 sunday 40
Route No : 3
Origin | Destination | Dateoftravel | Day | Capacity
surat navsari 15-AUG-21 thursday 56
Route No : 5
Origin | Destination | Dateoftravel | Day | Capacity
surat bilimora 30-OCT-21 sunday 40
```

-- Set - 11 : -

****

Create the database BUS TRANSPORT and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.

TRAIN _MASTER:

```
create table train_master
(
  trainno varchar(6) primary key,
  trainname varchar(25) not null,
  arrivaltime date not null,
  departuretime date not null,
  noofhour number(10,2) not null,
  sourcestation varchar(25) not null,
  endstation varchar(25) not null
);
insert into train_master values('T1DN','rajdhani','4-jan-2022','4-jan-2022',7,'surat station','anad stataion');
insert into train_master values('T2UP','bullet','22-dec-2021','22-dec-2021',5,'ahmedabad station','vapi stataion');
insert into train_master values('T3DN','express','7-jan-2019','7-jan-2019',6,'valsad station','rajkot stataion');
insert into train_master values('T4UP','local','15-feb-2022','15-feb-2022',2,'surat station','bardoli stataion');
insert into train_master values('T5DN','chennai express','1-jan-2022','1-jan-2022',28,'surat station','chennai stataion');
select * from train_master;
```

PASSENGER_DETAILS:

```
create table passenger_details
(
  ticketno number(5) primary key,
  trainno varchar(6) references train_master(trainno) ON DELETE CASCADE,
  seatno number(2) not null,
  passangername varchar(35) not null,
  age number(2) not null,
  gender char(1) check(gender in ('M','F')),
  travelldate date,
  class varchar(4) check (class in('IA','IIA','IIIA','IC','II'))
);
insert into passenger_details values(001,'T2UP',21,'kharak shing','56','M','22-dec-2021','IA');
insert into passenger_details values(002,'T1DN',05,'babu bhaiya','65','M','4-jan-2022','II');
insert into passenger_details values(003,'T4UP',15,'tiwari sheth','40','M','13-jan-2022','IIIA');
insert into passenger_details values(004,'T3DN',45,'chameli ben','15','F','15-sep-2021','IC');
insert into passenger_details values(005,'T5DN',50,'kalin bhiya','45','M','05-dec-2020','IIA');
select * from passenger_details;
```

TRAIN_SEAT_MASTER:

```
create table train_seat_master
(
  trainno varchar(6) references train_master(trainno) ON DELETE CASCADE,
  class varchar(4)  check (class in('IA','IIA','IIIA','IC','II')),
```

```
  total_seat number(2) check (total_seat >=25 and total_seat <=90)
);
insert into train_seat_master values('T1DN','IC','40');
insert into train_seat_master values('T5DN','IIIA','55');
insert into train_seat_master values('T2UP','IA','45');
insert into train_seat_master values('T4UP','IIA','35');
insert into train_seat_master values('T3DN','II','25');
select * from train_seat_master;
```

TRAIN_DAY_MASTER:

```
create table train_day_master
(
  trainno varchar(6) references train_master(trainno) ON DELETE CASCADE,
  day varchar(3) check (day in('mon','tue','wed','thu','fri','sat','sun'))
);
insert into train_day_master values('T1DN','mon');
insert into train_day_master values('T2UP','tue');
insert into train_day_master values('T5DN','fri');
insert into train_day_master values('T4UP','sat');
insert into train_day_master values('T3DN','sun');
select * from train_day_master;
```

---

7. Write a procedure which will print all train details going from Baroda to Banglore.
Ans.

```
set serveroutput on;
CREATE OR REPLACE PROCEDURE set11Q7
as
  cursor  tr_cur is select trainno,trainname from train_master where sourcestation = 'surat station' and
endstation = 'bardoli stataion';
  trainno train_master.trainno%type;
  trainname train_master.trainname%type;
BEGIN
  open tr_cur;
    loop
      fetch tr_cur into trainno,trainname;
        exit when tr_cur%notfound;
        dbms_output.put_line(trainno ||' - '||trainname);
    end loop;
  close tr_cur;
END;
```

Output : -

```
Procedure created.
```

```
execute set11Q7;
```

Output : -

```
Statement processed.
T4UP - local
```

8. **Write a function which will print arrival time and departure time for a given train. (pass train no as a parameter)**

Ans.

```
CREATE OR REPLACE FUNCTION Set11Q8 (tno In varchar2)
RETURN number
IS
   arrivaltime train_master.arrivaltime%type;
   departuretime train_master.departuretime%type;
BEGIN
   select arrivaltime,departuretime into arrivaltime,departuretime from train_master where trainno = tno;
   dbms_output.put_line('Arrival time : '||arrivaltime);
   dbms_output.put_line('Departure time : '||departuretime);
   return 0;
END;
```

Output : -

```
Function created.
```

```
DECLARE
   a number(2);
BEGIN
   a := Set11Q8('T1DN');
END;
```

Output : -

```
Statement processed.
Arrival time : 04-JAN-22
Departure time : 04-JAN-22
```

9. **Write a trigger which do not allow to insert or update passenger record if age is greater than 100.**

Ans.

```
CREATE OR REPLACE TRIGGER set11Q9
  BEFORE insert or update
  on passenger_details
  FOR EACH ROW
BEGIN
  IF(:new.age >= 100) THEN
    RAISE_APPLICATION_ERROR(-20000, 'Do not enter pssaanger age above 100.....');
  END IF;
END;
```

Output : -

```
Trigger created.
```

-- Set - 12 : -

****

CUSTOMER(cid, fname, lname, city, country, phone)
ORDER (oid, oDate, oNumber, cid, oTotalAmount)

Ans.

```
create table customer(
   cid number(5)primary key,
   fname varchar(20),
   lname varchar(20),
   city varchar(30),
   country varchar(20),
   phone number(11)
);
insert into customer values (1,'urvi','vaghasiya','surat','india',9879776);
insert into customer values (2,'rics','gangani','ahemdabad','india',874562);
insert into customer values (3,'janu','barvaliya','tokio','japan',5845578);
insert into customer values (4,'bindu','vaghasiya','new york','america',9679776);
insert into customer values (5,'pratik','savani','wuhan','china',125894);
create table ordr(
   oid number(5)primary key,
   odate date,
   onumber number(5),
   cid number(5) references customer(cid),
   ototal_amount number(10,2)
);
insert into ordr values (1,'11-jan-2022',50,1,5000);
insert into ordr values (2,'01-mar-2021',60,2,15000);
insert into ordr values (3,'30-dec-2021',150,3,25000);
insert into ordr values (4,'09-nov-2022',550,4,55000);
insert into ordr values (5,'23-may-2020',250,5,10000);
insert into ordr values (6,'23-may-2020',250,5,10000);
```

4. Create a trigger that executes whenever country is updated in CUSTOMER table.

Ans.

```
CREATE OR REPLACE TRIGGER set12Q4
   BEFORE update
   ON customer
   FOR EACH ROW

BEGIN
      :new.city := :old.city;
      RAISE_APPLICATION_ERROR(-20000,'You can not modify city of customer...');
END;
update customer set city='' where cid=2;
```

Output : -

```
Trigger created.
```

5. Create a function to return customer with maximum orders.

Ans.

```
CREATE OR REPLACE FUNCTION SET12Q5
RETURN number
AS
```

```
   total number(2);
BEGIN
   select cid into total from ordr where onumber = (select max(onumber) from ordr) group by cid;
   return total;
END;
```

Output : -

```
Function created.
```

```
DECLARE
   cus number(2);
BEGIN
   cus := SET12Q5();
   dbms_output.put_line('maximum order customer id : '||cus);
END;
```

Output : -

```
Statement processed.
maximum order customer id : 4
```

6. Create a procedure to display month names of dates of ORDER table. The month names should be unique.

Ans.

```
CREATE OR REPLACE PROCEDURE SET12Q6
AS
   cursor c is select to_char(odate,'mon') from ordr group by TO_CHAR(odate,'mon');
   mon varchar(5);
BEGIN
   open c;
      loop
         fetch c into mon;
         exit when c%notfound;
         dbms_output.put_line('->'||mon);
      end loop;
   close c;
END;
```

Output : -

```
Procedure created.
```

```
execute SET12Q6;
```

Output : -

```
Statement processed.
->mar
->may
->jan
->dec
->nov
```

-- Set - 14 : -

****

1.Write a PLSQL block which will print Employee list (Empno and Name)

EMP (empno, empnm, empadd, salary, date_birth, joindt, deptno)

Ans.

```
CREATE TABLE emp(
   empno number(5)primary key,
   empnm varchar(20),
   empadd varchar(50),
   salary number(10,2),
   bod date,
   joindt date,
   deptno number(5)
 );
insert into emp values (1,'urvi','mota varachha',10000,'11-jan-2001','20-mar-2020',5);
insert into emp values (2,'janu','hirabag',15000,'1-jun-2000','10-apr-2019',4);
insert into emp values (3,'rics','hazira',20000,'15-jul-1990','11-may-2018',3);
insert into emp values (4,'jinu','katargam',25000,'30-nov-1995','31-jul-2021',2);
insert into emp values (5,'vihana','jakatnaka',30000,'05-dec-1999','25-jun-2020',1);
select * from emp;
```

```
Declare
   empno emp.empno%type;
   empnm emp.empnm%type;
   empadd emp.empadd%type;
   salary emp.salary%type;
   bod emp.bod%type;
   joindt emp.joindt%type;
   deptno emp.deptno%type;

   cursor c is select * from emp;
BEGIN
   open c;
      loop
         fetch c into empno,empnm,empadd,salary,bod,joindt,deptno;
         exit when c%notfound;
         dbms_output.put_line(empno||'|'||empnm||'|'||empadd||'|'||salary||'|'||bod||'|'||joindt||'|'||deptno);
      end loop;
   close c;
END;
```

Output : -

```
Statement processed.
1|urvi|mota varachha|10000|11-JAN-01|20-MAR-20|5
2|janu|hirabag|15000|01-JUN-00|10-APR-19|4
3|rics|hazira|20000|15-JUL-90|11-MAY-18|3
4|jinu|katargam|25000|30-NOV-95|31-JUL-21|2
5|vihana|jakatnaka|30000|05-DEC-99|25-JUN-20|1
```

2. Write a function that returns total number of incomplete jobs, using table
JOB (jobid, type_of_job, status)
Ans.

```
CREATE TABLE job(
   jobid number(5)primary key,
   type_of_job varchar(20),
   status number(5)
);
insert into job values (1,'boss',1);
insert into job values (2,'Auditor',0);
insert into job values (3,'IPS',1);
insert into job values (4,'hod',0);
insert into job values (5,'clerk',1);
select * from job;
```

```
CREATE OR REPLACE FUNCTION SET14Q2
RETURN number
AS
   no_of_job number(2);
BEGIN
   select count(jobid) into no_of_job from job where status = 0;
   return no_of_job;
END;
```

Output : -

```
Function created.
```

```
DECLARE

BEGIN
   dbms_output.put_line('INCOMPLETE JOB : '||SET14Q2());
END;
```

Output : -

```
Statement processed.
INCOMPLETE JOB : 2
```

3. Write a function which displays the number of items whose weight fall between a given ranges for a particular color using table
ITEM (itemno, name, color, weight)
Ans.

```
CREATE TABLE item(
   itemno number(5)primary key,
   name varchar(20),
   color varchar(10),
   weight number(5)
);
insert into item values (1,'laptop','black',50);
insert into item values (2,'PC','blue',100);
insert into item values (3,'mouse','white',5);
insert into item values (4,'keyboard','brown',10);
insert into item values (5,'pendrive','silver',2);
```

```
select * from item;
```

```
CREATE OR REPLACE FUNCTION SET14Q3 (color IN varchar,s IN number,e IN number)
RETURN number
AS
   tt number(2);
BEGIN
   select count(itemno) into tt from item where color='white' and weight>s AND weight<e;
   return tt;
END;
```
Output : -

```
Function created.
```

```
DECLARE
   total number(2);
BEGIN
   total := SET14Q3('black',1,100);
   dbms_output.put_line('Total item as per your requirment : '||total);
END;
```
Output : -

```
Statement processed.
Total item as per your requirment : 1
```

4. Write a procedure to display top five highest paid workers who are specialized in 'PAINTING" using table
WORKER (workerid, name, wage_per_hour, specialized_in, manager_id)
Ans.
```
CREATE TABLE worker(
   wid number(5)primary key,
   name varchar(20),
   wage_per_hour number(5),
   specialized_in varchar(30),
   manager_id number(5)
);
insert into worker values (1,'urvi',500,'painting',100);
insert into worker values (2,'rics',400,'machines',200);
insert into worker values (3,'janu',600,'painting',300);
insert into worker values (4,'savani',300,'machines',400);
insert into worker values (5,'bindu',700,'painting',500);
select * from worker;
```

```
CREATE OR REPLACE PROCEDURE SET14Q4
AS
   cursor c is select wid,name,wage_per_hour from worker where specialized_in='painting' order by
wage_per_hour desc FETCH FIRST 5 ROWS ONLY;
   wid worker.wid%type;
   name worker.name%type;
   wage_per_hour worker.wage_per_hour%type;
BEGIN
   open c;
      loop
```

```
        fetch c into wid,name,wage_per_hour;
        exit when c%notfound;
        dbms_output.put_line('Wroker id : '||wid);
        dbms_output.put_line('Wroker Name : '||name);
        dbms_output.put_line('Wroker salary per hour : '||wage_per_hour);
        dbms_output.put_line('----------------------------');
      end loop;
   close c;
END;
```

Output : -

```
Procedure created.
```

EXECUTE SET14Q4;

Output : -

```
Statement processed.
Wroker id : 5
Wroker Name : bindu
Wroker salary per hour : 700
----------------------------
Wroker id : 3
Wroker Name : janu
Wroker salary per hour : 600
----------------------------
Wroker id : 1
Wroker Name : urvi
Wroker salary per hour : 500
----------------------------
```

-- Set - 15 : -

****

EMP (empno, empnm, empadd, salary, date_birth, joindt, deptno)
DEPT (deptno, deptnm)
Ans.

```
create table emp
(
   empno number(5) primary key,
   empnm varchar(20),
   empadd varchar(50),
   salary number(10,2),
   dob date,
   joindt date,
   deptno number(5) references dept(deptno)
);
insert into emp values(1,'rics','choal',150,'14-jan-2021','11-jan-2022',101);
insert into emp values(2,'urvi','mumbai',5000,'11-jan-2022','10-apr-2015',103);
insert into emp values(3,'janu','surat',10000,'15-dec-2021','11-may-201',104);
insert into emp values(4,'nainsi','navasari',25000,'05-oct-2021','31-jul-2014',102);
insert into emp values(5,'abhi','bilimora',15000,'2-jan-2022','25-sep-2023',105);
select * from emp;
create table dept
(
   deptno number(5)primary key,
   deptnm varchar(20)
);
insert into dept values(101,'manage');
insert into dept values(102,'account');
insert into dept values(103,'sales');
insert into dept values(104,'finanace');
insert into dept values(105,'marketing');
select * from dept;
```

```
DECLARE
   cursor cur is select e.empno,empnm,(TO_CHAR(sysdate,'yyyy')-TO_CHAR(e.joindt,'yyyy')) from emp
e,dept d where deptnm='sales' and d.deptno=e.deptno;
   empno emp.empno%type;
   empnm emp.empnm%type;
   yy number(5);
BEGIN
   open cur;
      loop
         fetch cur into empno,empnm,yy;
         exit when cur%notfound;
         dbms_output.put_line('emp no : '||empno||'     emp name : '||empnm||'       year : '||yy);
      end loop;
   close cur;
END;
```

Output : -

```
Statement processed.
emp no : 2     emp name : urvi       year : 7
```

-- Set - 16 : -

****

EMPMAST (empno, name, pfno, empbasic, deptno, designation)
DEPT (DNO, DNAME)
Rules: HRA = 15% of basic
DA = 50% of basic
Medical = 100
PF = 8.33%of basic
Print Salary slip. Design your own format

Ans.

```
create table empmast
(
   empno number(5) primary key,
   name varchar(20),
   pfno number(5),
   empbasic number(10,2),
   deptno number(5) references dept(dno),
   designation varchar(30)
);
insert into empmast values (501,'rsju',5001,6000,1,'manager');
insert into empmast values (502,'shyam',5002,20000,2,'member');
insert into empmast values (503,'babu',5003,6500,3,'peon');
insert into empmast values (504,'mukesh',5004,20000,4,'manager');
insert into empmast values (505,'suresh',5005,3500,5,'clerk');
select * from empmast
```

```
create table dept
(
   dno number(5) primary key,
   dname varchar(30)
);
insert into dept values (1,'manages');
insert into dept values (2,'sales');
insert into dept values (3,'account');
insert into dept values (4,'production');
insert into dept values (5,'marketing');
select * from dept;
```

```
DECLARE
   cursor cur is select empno,empbasic from empmast;
   empno empmast.empno%type;
   salary empmast.empbasic%type;
   HRA number(5);
   DA number(5);
   Medical number(5) := 100;
   PF number(5);
```

```
   Gsalary number(5);
BEGIN
   open cur;
      loop
         fetch cur into empno,salary;
         exit when cur%notfound;
         dbms_output.put_line('EMPLOYEE NO : '||empno);
         HRA := (salary*15)/100;
         DA := (salary*50)/100;
         PF := (salary*8.33)/100;
         Gsalary := salary+HRA+DA-Medical-PF;
         dbms_output.put_line('HRA |  DA  |  Medical  |  PF');
         dbms_output.put_line(HRA||' | '||DA||' |  '||Medical||'      | '||PF);
         dbms_output.put_line('Gross Salary is '||Gsalary);
         dbms_output.put_line('**************************************');
      end loop;
   close cur;
END;
```

Output : -

```
Statement processed.
EMPLOYEE NO : 501
HRA |  DA  |  Medical  |  PF
900 | 3000 |  100      | 500
Gross Salary is 9300
**************************************
EMPLOYEE NO : 502
HRA |  DA  |  Medical  |  PF
3000 | 10000 |  100      | 1666
Gross Salary is 31234
**************************************
EMPLOYEE NO : 503
HRA |  DA  |  Medical  |  PF
975 | 3250 |  100      | 541
Gross Salary is 10084
**************************************
EMPLOYEE NO : 504
HRA |  DA  |  Medical  |  PF
3000 | 10000 |  100      | 1666
Gross Salary is 31234
**************************************
EMPLOYEE NO : 505
HRA |  DA  |  Medical  |  PF
525 | 1750 |  100      | 292
Gross Salary is 5383
**************************************
```

-- Set - 17 : -

****

Consider the Bank schema as
ACCOUNT (AC_NO, NAME, AC_TYPE, BALANCE_AMT, BALANCE_DATE)
TRANSACTION (AC_NO, DATE, TR_TYPE, AMOUNT, PREV_BALANCE, REMARK)
Note: 1. AC_type may be S for saving or C for current, 2. TR_type may be D for deposit or W for withdrawal.
a. Write a procedure to print the Bank Transaction details by passing from and to dates.

Ans.

```
create table account
(
   acno number(5) primary key,
   name varchar(20),
   type varchar(20),
   balance number(10,2),
   baladate date
);

insert into account values (1,'boi','saving',10000,'12-jan-2022');
insert into account values (2,'sbi','current',15000,'3-dec-2021');
insert into account values (3,'icici','saving',30000,'11-jan-2022');
insert into account values (4,'hdfc','current',25000,'05-may-2022');
insert into account values (5,'bob','saving',35000,'10-sep-2022');

select * from account;
```

```
create table transaction(
   acno number(5) references account(acno),
   dt date,
   type varchar(30),
   amt number(10,2),
   pre_bal number(10,2),
   remark varchar(20)
);

insert into transaction values (1,'12-jan-2022','bank transfer',5500,25000,'pqr');
insert into transaction values (2,'11-jan-2022','phone pay',4000,45000,'abc');
insert into transaction values (3,'10-jan-2022','paytm',2050,20000,'xyz');
insert into transaction values (4,'11-jan-2022','Gpay',5700,10000,'mno');
insert into transaction values(5,'22-dec-2021','ATM',5000,20000,'ade');

select * from transaction;
```

```
CREATE OR REPLACE PROCEDURE set17Q1 (frm IN date,tto IN date)
AS
```

```
   cursor c is select acno,dt,type,amt from transaction where dt between frm and tto;
   acno account.acno%type;
   tdate transaction.dt%type;
   ttype transaction.type%type;
   tamt transaction.amt%type;
   cnt number(2) := 1;
BEGIN
   open c;
      loop
         fetch c into acno,tdate,ttype,tamt;
         exit when c%notfound;
         dbms_output.put_line('('||cnt||')');
         dbms_output.put_line('Account no    : '||acno||'       Date : '||tdate);
         dbms_output.put_line('Account Type  : '||ttype);
         dbms_output.put_line('Amount        : '||tamt);
         dbms_output.put_line('--------------------------------------------->');
         cnt := cnt + 1;
      end loop;
   close c;
END;
```

Output : -

```
Procedure created.
```

```
EXECUTE set17Q1('05-jan-2022','10-jan-2022');
```

Output : -

```
Statement processed.
(1)
Account no    : 3       Date : 10-JAN-22
Account Type  : paytm
Amount        : 2050
--------------------------------------------->
```

-- Set - 18 : -

****

BRANCH (branch_no, area, city)
MEMBERS (mno, name branch_no, salary, manager_no)
Note: Manager can be from one of the members.
Ans.

```
create table branch
(
   branch_no number(5) primary key,
   area varchar(20),
   city varchar(20)
);
insert into branch values(1,'bhagal','surat');
insert into branch values(2,'chok','rajkot');
insert into branch values(3,'shanivari','vapi');
insert into branch values(4,'amroli','surat');
insert into branch values(5,'katargam','ahmedabad');
select * from branch;
```

```
create table members
(
   mno number(5) primary key,
   name varchar(20),
   branch_no number(5) references branch(branch_no),
   salary number(10,2),
   manager_no number(5) references manager(manager_no)
);
insert into members values (101,'flex',1,5000,01);
insert into members values (102,'dom',4,15000,02);
insert into members values (103,'max',5,20000,03);
insert into members values (104,'bludy',2,10000,02);
insert into members values (105,'bob',3,40000,01);
select * from members;
```

```
create table manager
(
   manager_no number(5) primary key,
   salary number(10,2)
);
insert into manager values (01,10000);
insert into manager values (02,20000);
insert into manager values (03,30000);

select * from manager;
```

---

1. Write a procedure which list the name of members who earns more than that of his managers.

Ans.

```
CREATE OR REPLACE PROCEDURE Set18Q1
AS
   cursor cur is select mno,name from members ms,manager mg where ms.salary>mg.salary and
ms.manager_no=mg.manager_no;
   mno members.mno%type;
   name members.name%type;
BEGIN
   open cur;
      loop
         fetch cur into mno,name;
         exit when cur%notfound;
         dbms_output.put_line('Member No : '||mno);
         dbms_output.put_line('Member Name : '||name);
         dbms_output.put_line('************************');
      end loop;
   close cur;
END;
```

Output : -

```
Procedure created.
```

```
EXECUTE Set18Q1;
```

Output : -

```
Statement processed.
Member No : 105
Member Name : bob
************************
```

2. Write a procedure which gives details of employee having maximum salary branch wise.

Ans.

```
CREATE OR REPLACE PROCEDURE Set18Q2
AS
   cursor cur is select branch_no,max(salary) from members group by branch_no;
   branch_no branch.branch_no%type;
   salary members.salary%type;
BEGIN
   dbms_output.put_line('Branch ID     Employee Salary(max)');
   open cur;
      loop
         fetch cur into branch_no,salary;
         IF cur%notfound THEN
            EXIT;
         END IF;
         dbms_output.put_line(branch_no||'                '||salary);
      end loop;
   close cur;
END;
```

Output : -

```
Procedure created.
```

EXECUTE Set18Q2;

Output : -

```
Statement processed.
Branch ID     Employee Salary(max)
1                  5000
2                  10000
4                  15000
5                  20000
3                  40000
```

-- Set - 19 : -

****

Employee (eid, fname, lname, salary)

```
create table employee
(
   eid number(5)primary key,
   fname varchar(30),
   lname varchar(30),
   salary number(10,2)
);

insert into employee values(101,'urvi','vaghasiya',5000);
insert into employee values(102,'rics','gangani',10000);
insert into employee values(103,'janu','barvaliya',25000);
insert into employee values(104,'krina','gajera',10000);
insert into employee values(105,'pratik','savani',6000);

select * from employee;
```

---

1. **Use a Cursor for Loop inside a function to calculate and return total paid salary to all employees by the company.**

Ans.

```
CREATE OR REPLACE FUNCTION Set19Q1
RETURN number
AS
   cursor cur is select sum(salary) from employee;
   total number(5);
BEGIN
   open cur;
      loop
         fetch cur into total;
         exit when cur%notfound;
         return total;
      end loop;
   close cur;
END;
```

Output : -

```
Function created.
```

```
DECLARE
   total_salary number(5);
BEGIN
   total_salary := Set19Q1;
   dbms_output.put_line('Total paid salary by department : '||total_salary||' Rs.');
END;
```

Output : -

```
Statement processed.
Total paid salary by department : 56000 Rs.
```

2. **Modify the function created above to become a procedure and display the total paid salary from the procedure itself. Instead of calculating for all employees, calculate only for those employees whose name starts from a character passed as parameter to the procedure and hence to the cursor.**

Ans.

```
CREATE OR REPLACE PROCEDURE Set19Q2 (str IN varchar)
AS
   cursor cur is select sum(salary) from employee where fname LIKE str||'%';
   total_s number(5);
BEGIN
   open cur;
      loop
         fetch cur into total_s;
         if cur%notfound then
            exit;
         end if;
         dbms_output.put_line('Total paid salary by department : '||total_s||' Rs.');
      end loop;
   close cur;
END;
```

Output : -

```
Procedure created.
```

```
EXECUTE Set19Q2('');
```

Output : -

```
Statement processed.
Total paid salary by department : 56000 Rs.
```

-- Set - 20 : -

****

Consider the DUAL and data dictionary tables/views to solve the following Queries.

1. **Find out the names of all the tables, views and constraints associated with current tables in the system.**

Ans. select OWNER,CONSTRAINT_NAME ,COLUMN_NAME,POSITION from User_cons_columns where TABLE_NAME='person';

2. **Write a query to add 15 days to the current date.**

Ans. select sysdate,sysdate+15 as adddate from dual;

SYSDATE	ADDDATE
16-JAN-22	31-JAN-22

3. **Write a query to Add and subtract 5 months from the current month.**

Ans. select add_months(sysdate,5)AS add_month_five,add_months(sysdate,-5) AS sub_month_five from dual;

ADD_MONTH_FIVE	SUB_MONTH_FIVE
16-JUN-22	16-AUG-21

4. **Find out the ASCII equivalent of character 'M'.**

Ans. select ASCII('M') from dual;

ASCII('M')
77

5. **Find out the character equivalent of ASCII 67, 65 and 84.**

Ans. select chr(67),chr(65),chr(84) from dual;

CHR(67)	CHR(65)	CHR(84)
C	A	T

6. **Write a query to find the last day of the month.**

Ans. select last_day(sysdate) from dual;

LAST_DAY(SYSDATE)
31-JAN-22

7. **Find out how many days are left in the current month.**

Ans. select last_day(sysdate)-sysdate AS Left_Days from dual;

LEFT_DAYS
15

8. **Write a query to calculate the Date difference between current date and 20/05/2015.**

Ans. SELECT trunc(sysdate) - TO_date('20/05/2015', 'dd/mm/yyyy') DAYS FROM DUAL;

DAYS
2433

9. **Write a query to Calculate the number of months between current date and 03/03/2016.**

Ans. SELECT MONTHS_BETWEEN (sysdate,'03-mar-2016') "Months" FROM dual;

Months
70.424168533452807646356033452807646356 03

10. **Find out the second occurrence of ‘or’ from third position in the string ‘corporate floor**

Ans. select INSTR('Corporate Floor','or', 3, 2) from dual;

INSTR('CORPORATEFLOOR','OR',3,2)
14

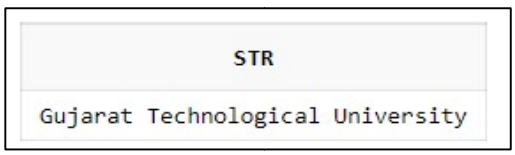
11. **Find out log to the base 3 of 81.**

Ans. select ceil(log(3,81)) AS LOG from dual;

LOG
4

12. **Convert the string ‘gujarat technological university’ so that first character of each work is in capital.**

Ans. select INITCAP('gujarat technological university') AS STR from dual;



STR
Gujarat Technological University

13. **Convert the string ‘jack and jue’ Into ‘black and blue’.**

Ans. select REPLACE('jack and jue','j','bl') AS REPLCE from dual;

REPLCE
black and blue

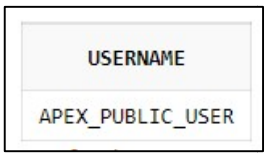
14. Round off the date 27-July-2016 to the current year.

Ans. select ROUND (TO_DATE ('27-jul-16'),'YY')-366 from dual;

ROUND(TO_DATE('27-JUL-16'),'YY')-366
01-JAN-16

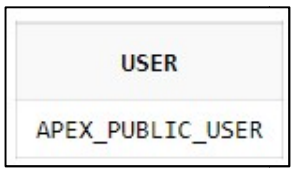
15. Find out the user name and user id off currently logged on user.

Ans. SELECT USERNAME FROM V$SESSION;



USERNAME
APEX_PUBLIC_USER

select user from dual;



USER
APEX_PUBLIC_USER