SQL-1 DATE:

1.Write different types of sql queries using ddl and dml statement.

```
postgres=# create table student3(regno int,sname varchar(10),mark float);
CREATE TABLE
postgres=#  insert into student3 values(111,'anu',86);
INSERT 0 1
postgres=# insert into student3 values(112,'appu',83);
INSERT 0 1
postgres=# insert into student3 values(113,'anamika',79);
INSERT 0 1
postgres=# insert into student3 values(114,'raju',36);
INSERT 0 1
postgres=# insert into student3 values(115,'kichu',96);
INSERT 0 1
postgres=# select *from student3;
 regno |  sname  | mark
-------+---------+------
   111 | anu     |   86
   112 | appu    |   83
   113 | anamika |   79
   114 | raju    |   36
   115 | kichu   |   96
(5 rows)
```

```
postgres=# select*from student3;
 regno |  sname  | mark | rank
-------+---------+------+------
   111 | anu     |   86 |    1
   112 | appu    |   83 |    2
   113 | anamika |   79 |    3
   114 | raju    |   36 |    4
   115 | kichu   |   96 |    5
(5 rows)

postgres=# delete from student3 where mark=79;
DELETE 1
postgres=# select *from student3 order by regno asc;
 regno | sname | mark | rank
-------+-------+------+------
   111 | anu   |   86 |    1
   112 | appu  |   83 |    2
   114 | raju  |   36 |    4
   115 | kichu |   96 |    5
(4 rows)
postgres=# alter table student3 rename mark to score;
ALTER TABLE
postgres=# select*from student3;
 regno | sname | score | rank
-------+-------+-------+------
   111 | anu   |    86 |    1
   112 | appu  |    83 |    2
   114 | raju  |    36 |    4
   115 | kichu |    96 |    5
(4 rows)
```

2.Create and display view.

```
postgres=# CREATE VIEW STUDENT_DATA AS SELECT SNAME,SCORE FROM STUDENT3;
CREATE VIEW
postgres=# SELECT * FROM STUDENT_DATA;
 sname | score
-------+-------
 anu   |    86
 appu  |    83
 raju  |    36
 kichu |    96
(4 rows)
```

3.Write queries using joins

```
postgres=# CREATE  TABLE SD(SNO INT,NAME VARCHAR(10),PALCE VARCHAR(20));
CREATE TABLE
postgres=# INSERT INTO SD VALUES(115,'ACHU','CHENNAI');
INSERT 0 1
postgres=# INSERT INTO SD VALUES(116,'AMMU','COCHI');
INSERT 0 1
postgres=# INSERT INTO SD VALUES(116,'AMMU','ALUVA');
INSERT 0 1
postgres=# SELECT*FROM SD;
```

```
 sno | name |  palce

-----+------+---------

 115 | ACHU | CHENNAI

 116 | AMMU | COCHI

 116 | AMMU | ALUVA

(3 rows)


postgres=# SELECT * FROM SD INNER JOIN STUDENT3
ON(SD.SNO=STUDENT3.REGNO);

 sno | name |  palce  | regno | sname | score | rank

-----+------+---------+-------+-------+-------+------

 115 | ACHU | CHENNAI |   115 | kichu |    96 |    5

(1 row)


postgres=# SELECT*FROM SD LEFT OUTER JOIN STUDENT3
ON(SD.SNO<>STUDENT3.REGNO);

 sno | name |  palce  | regno | sname | score | rank

-----+------+---------+-------+-------+-------+------

 115 | ACHU | CHENNAI |   111 | anu   |    86 |    1

 115 | ACHU | CHENNAI |   112 | appu  |    83 |    2

 115 | ACHU | CHENNAI |   114 | raju  |    36 |    4

 116 | AMMU | COCHI   |   111 | anu   |    86 |    1

 116 | AMMU | COCHI   |   112 | appu  |    83 |    2

 116 | AMMU | COCHI   |   114 | raju  |    36 |    4

 116 | AMMU | COCHI   |   115 | kichu |    96 |    5

 116 | AMMU | ALUVA   |   111 | anu   |    86 |    1

 116 | AMMU | ALUVA   |   112 | appu  |    83 |    2
 116 | AMMU | ALUVA   |   114 | raju  |    36 |    4
 116 | AMMU | ALUVA   |   115 | kichu |    96 |    5
(11 rows)


postgres=# SELECT*FROM SD,STUDENT3 WHERE (SD.SNO=STUDENT3.REGNO);
 sno | name |  palce  | regno | sname | score | rank
-----+------+---------+-------+-------+-------+------
 115 | ACHU | CHENNAI |   115 | kichu |    96 |    5
(1 row)
```

4.WRITE QUERIES USING SUBQUIERIS.

```
postgres=# SELECT SNAME ,SCORE FROM STUDENT3 WHERE REGNO IN(SELECT SNO
FROM SD W
HERE SCORE>90);
 sname | score
-------+-------
 kichu |    96
(1 row)
```

5.WRITE PATTERN MATCHING QUERIES.

```
                ^
postgres=# SELECT SNAME FROM STUDENT3 WHERE SNAME LIKE 'a%';
 sname
-------
 anu
 appu
(2 rows)
```

6.QUERIES USING DIFFERENT DIFFERENT OPERATORS.

```
postgres=# SELECT SNAME FROM STUDENT3 WHERE SCORE<90;
 sname
-------
 anu
 appu
 raju
(3 rows)

postgres=# SELECT SNAME FROM STUDENT3 WHERE SCORE>90;
 sname
-------
 kichu
(1 row)

postgres=# SELECT SNAME FROM STUDENT3 WHERE SCORE<>90;
 sname
-------
 anu
 appu
 raju
 kichu
(4 rows)
```

7.WRITE QUERIES USING VARIOUS TYPES OF FUNCTIONS.

```
postgres=# SELECT AVG(SCORE) FROM STUDENT3;
  avg
```

```
-------
 75.25
(1 row)
postgres=# SELECT LPAD(SNAME::TEXT,10,'*')FROM STUD
    lpad
------------
 *******anu
 ******appu
 ******raju
 *****kichu
(4 rows)

postgres=# SELECT LOWER(SNAME)FROM STUDENT3;
 lower
-------
 anu
 appu
 raju
 kichu
(4 rows)

postgres=# SELECT MIN(SCORE)FROM STUDENT3;
 min
-----
  36
(1 row)
```

```
postgres=# SELECT MIN(SCORE)FROM STUDENT3;
 min
-----
  36
(1 row)

postgres=# SELECT MAX(SCORE)FROM STUDENT3;
 max
-----
  96
(1 row)

postgres=# SELECT COUNT(SCORE)FROM STUDENT3;
 count
-------
     4
(1 row)



postgres=# SELECT POWER(SCORE,2)FROM STUDENT3;
 power
-------
  7396
  6889
  1296
  9216
(4 rows)
```

8.QUERIES USING SEQUENCE.

```
postgres=# SELECT MIN(SCORE)FROM STUDENT3;
 min
-----
  36
(1 row)



postgres=# SELECT MAX(SCORE)FROM STUDENT3;
 max
-----
  96
(1 row)



postgres=# SELECT COUNT(SCORE)FROM STUDENT3;
 count
-------
     4
(1 row)

postgres=# SELECT POWER(SCORE,2)FROM STUDENT3;
 power
```

```
-------
  7396
  6889
  1296
  9216
(4 rows)
 postgres=# CREATE SEQUENCE NUMBER INCREMENT BY 1 START WITH 1 MAXVALUE
10 NO CYC
LE;
CREATE SEQUENCE
postgres=# SELECT * FROM NUMBER;
 sequence_name | last_value | start_value | increment_by | max_value | min_value
 | cache_value | log_cnt | is_cycled | is_called
---------------+------------+-------------+--------------+-----------+----------
-+-------------+---------+-----------+-----------
 number        |          1 |           1 |            1 |        10 |         1
 |           1 |       1 | f         | f
(1 row)
```

SQL-2 DATE:

Create a table students with fields sno,sname,sex,mark with sno as primary key and assign suitable constraints for each attribute.Insert five records into the table.

```
postgres=# CREATE TABLE STUDENT5(SNO INT PRIMARY KEY,SNAME VARCHAR(20)
NOT NULL,
SEX VARCHAR(20) NOT NULL,MARK FLOAT NOT NULL);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "student5_pkey" f
or table "student5"
```

```
CREATE TABLE
                  ^
postgres=# INSERT INTO STUDENT5 VALUES(10,'ROSY','F',250);
INSERT 0 1
postgres=# INSERT INTO STUDENT5 VALUES(11,'RAMU','M',240);
INSERT 0 1
postgres=# INSERT INTO STUDENT5 VALUES(21,'RAJU','F',250);
INSERT 0 1
postgres=# INSERT INTO STUDENT5 VALUES(23,'ARU','F',260);
INSERT 0 1
postgres=# INSERT INTO STUDENT5 VALUES(27,'ADHI','M',289);
INSERT 0 1postgres=# SELECT * FROM STUDENT5;
 sno | sname | sex | mark
-----+-------+-----+------
  10 | ROSY  | F   |  250
  11 | RAMU  | M   |  240
  21 | RAJU  | F   |  250
  23 | ARU   | F   |  260
  27 | ADHI  | M   |  289
(5 rows)
```

1.ALTER THE TABLE BY ADDING ONE MORE FIELD RANK.

```
postgres=# ALTER TABLE STUDENT5 ADD RANK INT;
ALTER TABLE
postgres=# UPDATE STUDENT5 SET RANK=1 WHERE SNO=10;
UPDATE 1
```

```
postgres=# UPDATE STUDENT5 SET RANK=2 WHERE SNO=11;
UPDATE 1
postgres=# UPDATE STUDENT5 SET RANK=3 WHERE SNO=21;
UPDATE 1
postgres=# UPDATE STUDENT5 SET RANK=4 WHERE SNO=23;
UPDATE 1
postgres=# UPDATE STUDENT5 SET RANK=5 WHERE SNO=27;
UPDATE 1
postgres=# SELECT*FROM STUDENT5;
 sno | sname | sex | mark | rank
-----+-------+-----+------+------
  10 | ROSY  | F   |  250 |    1
  11 | RAMU  | M   |  240 |    2
  21 | RAJU  | F   |  250 |    3
  23 | ARU   | F   |  260 |    4
  27 | ADHI  | M   |  289 |    5
(5 rows)
```

2.DISPLAY ALL BOY STUDENT S WITH THEIR NAME.

```
postgres=# SELECT SNAME FROM STUDENT5 WHERE SEX ='M';
 sname
-------
 RAMU
 ADHI
(2 rows)
```

3.FIND THE AVERAGE MARK.

```
postgres=# SELECT AVG(MARK) FROM STUDENT5;

  avg

-------

 257.8

(1 row)
```

4.CREATE A QUERY TO DISPLAY THE SNO AND SNAME FOR ALL STUDENTS WHO GOT MORE THAN THE AVERAGE MARK.SORTS THE RESULTS IN DESCENDING ORDER OF MARK.

```
postgres=# SELECT SNO,SNAME FROM STUDENT5 WHERE MARK>(SELECT AVG(MARK) FROM STUD

ENT5)ORDER BY MARK DESC;

 sno | sname

-----+-------

  27 | ADHI

  23 | ARU

(2 rows)
```

5.Create a sequence named ‘star’ to be used with student table’s primary key column-sno.The sequence should start with 10 & max value 99.

```
postgres=# CREATE SEQUENCE STAR1 INCREMENT BY 1 START WITH 10 MAXVALUE 99 NO CYC

LE;

CREATE SEQUENCE

postgres=# SELECT * FROM STAR1;

 sequence_name | last_value | start_value | increment_by | max_value | min_value

 | cache_value | log_cnt | is_cycled | is_called

---------------+------------+-------------+--------------+-----------+----------
```

```
-+-------------+---------+-----------+-----------

 star1       |       10 |        10 |          1 |      99 |       1

 |         1 |      1 | f        | f

(1 row)
```

6.Display girl student name for those who have marks greater than 40 and less than 20.

```
postgres=# SELECT SNAME FROM STUDENT5 WHERE SEX='F' AND(MARK>40 OR
MARK<20);

 sname

-------

 ROSY

 RAJU

 ARU

(3 rows)
```

SQL -3   DATE:

Create a table department with fields ename ,salary,dno,dname,place with dno as primary key .Insert five records into the table.

```
postgres=# create table department2(ename varchar(20),salary float,dno integer n

ot null primary key,dname varchar(20),place varchar(40));

NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "department2_pkey
```

```
" for table "department2"
CREATE TABLE
postgres=# insert into department2 values('sreena',10000,1,'hardware','banglore'
);
INSERT 0 1
postgres=# insert into department2 values('maya',10000,2,'software','banglore');


INSERT 0 1
postgres=# insert into department2 values('tony',25000,3,'md','chennai');
INSERT 0 1
postgres=# insert into department2 values('vijay',15000,4,'clerk','hydarabad');
INSERT 0 1
postgres=# select * from department2;
 ename  | salary | dno |  dname   |   place
--------+--------+-----+----------+-----------
 sreena |  10000 |   1 | hardware | banglore
 maya   |  10000 |   2 | software | banglore
 tony   |  25000 |   3 | md       | chennai
 vijay  |  15000 |   4 | clerk    | hydarabad
(4 rows)
```

1:Rename the field ‘place’with ‘city’.

```
postgres=# alter table department2 rename place to city;
ALTER TABLE
postgres=# select *from department2;
```

```
 ename  | salary | dno |  dname   |   city
--------+--------+-----+----------+-----------
 sreena |  10000 |   1 | hardware | banglore
 maya   |  10000 |   2 | software | banglore
 tony   |  25000 |   3 | md       | chennai
 vijay  |  15000 |   4 | clerk    | hydarabad
(4 rows)
```

2:Display the employees who got salary more than 10000 and less than 20000.

```
postgres=# select ename from department2 where salary >10000 and salary<20000;
 ename
-------
 vijay
(1 row)
```

3;Display total salary of the organization.

```
postgres=# select sum(salary) from department2;
  sum
-------
 60000
(1 row)
```

4:Display ename for those who are getting salary in between 10000 and 20000.

```
postgres=# select ename from department2 where salary between 10000 and 20000;
 ename
```

```
--------
 sreena
 maya
 vijay
(3 rows)
```

5:Create a view named ‘star’ with fieldname salry and place.

```
postgres=# create view starn as select ename,city from department2;
CREATE VIEW
postgres=# select * from starn;
 ename  |   city
--------+-----------
 sreena | banglore
 maya   | banglore
 tony   | chennai
 vijay  | hydarabad
(4 rows)
```

6:Dislay ename and salary ,salary rounded with 10 digits.

```
postgres=# select ename,round(salary::decimal,10)from department2;
 ename  |      round
--------+------------------
 sreena | 10000.0000000000
 maya   | 10000.0000000000
 tony   | 25000.0000000000
 vijay  | 15000.0000000000
(4 rows)
```

SQL-4 DATE:

Create a table emp with fields eno,ename,job,manager,salary,with eno as primary key.Insert five records into the table.

```
postgres=# create table emp12(eno int not null primary key,ename varchar(20),job
 varchar(20),manager varchar(20),salary float);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "emp12_pkey" for
table "emp12"
CREATE TABLE
```

```
postgres=# insert into emp12 values(20,'ammu','clerk','arun',25000);
INSERT 0 1
postgres=# insert into emp12 values(21,'appu','pune','maya',20000);
INSERT 0 1
postgres=# insert into emp12 values(22,'varun','accounting','sreya',33000);
INSERT 0 1
postgres=# insert into emp12 values(23,'athira','accounting','sreya',35000);
INSERT 0 1
postgres=# insert into emp12 values(24,'divya','clerk','arun',25500);
INSERT 0 1
postgres=# select * from emp12;
 eno | ename  |    job     | manager | salary
-----+--------+------------+---------+--------
  20 | ammu   | clerk      | arun    |  25000
  21 | appu   | pune       | maya    |  20000
  22 | varun  | accounting | sreya   |  33000
  23 | athira | accounting | sreya   |  35000
  24 | divya  | clerk      | arun    |  25500
(5 rows)
```

1.Display the ename and salary,salary with ascending order

```
postgres=# select ename,salary from emp12 order by salary asc;
 ename  | salary
--------+--------
 appu   |  20000
 ammu   |  25000
 divya  |  25500
 varun  |  33000
 athira |  35000
(5 rows)
```

2.Display ename and salary for eno=20

```
postgres=# select ename,salary from emp12 where eno=20;
 ename | salary
-------+--------
 ammu  |  25000
(1 row)
```

3.Create another table department with fields dno,salary,eno,dname,and place with eno as primary key.

```
postgres=# create table dept22(dno int,salary float,eno int not null primary key
,dname varchar(20),place varchar(20));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "dept22_pkey" for
 table "dept22"
CREATE TABLE
postgres=# insert into dept22 values(1,15000,22,'accounting','mumbai');
INSERT 0 1
postgres=# insert into dept22 values(2,16000,23,'sales','kolkata');
INSERT 0 1
postgres=# insert into dept22 values(3,17000,24,'sales','kolkata');
INSERT 0 1
postgres=# insert into dept22 values(4,18000,25,'accounting','delhi');
```

```
INSERT 0 1
postgres=# select *from dept22;
 dno | salary | eno |   dname    |  place
-----+--------+-----+------------+---------
   1 |  15000 |  22 | accounting | mumbai
   2 |  16000 |  23 | sales      | kolkata
   3 |  17000 |  24 | sales      | kolkata
   4 |  18000 |  25 | accounting | delhi
(4 rows)
```

4.Display the manager for the accounting Department(join)

```
postgres=# select distinct(manager)from emp12,dept22 where emp12.eno in(select e
no from dept22 where dname='accounting');
 manager
---------
 sreya
(1 row)
```

5.Write the queries using various group functions.

```
postgres=# select sum(salary) from emp12;
  sum
--------
 138500
(1 row)
```

6.Write the queries using various Number functions.

```
postgres=# select max(salary) from emp12;
  max
-------
```

```
 35000
(1 row)


postgres=# select min(salary) from emp12;
  min
-------
 20000
(1 row)

postgres=# select count(*) from emp12;
 count
-------
     5
(1 row)
postgres=# select count(ename) from emp12;
 count
-------
     5
(1 row)
postgres=# select power(salary,2) from emp12;
   power
------------
  625000000
  400000000
 1089000000
 1225000000
  650250000
(5 rows)

postgres=# select ceil(salary) from emp12 where eno=23;
 ceil
-------
 35000
(1 row)
postgres=# select floor(salary) from emp12 where eno=23;
 floor
-------
 35000
(1 row)
postgres=# select ename,round(salary::decimal,2) from emp12;
 ename  |  round
--------+----------
 ammu   | 25000.00
 appu   | 20000.00
 varun  | 33000.00
 athira | 35000.00
 divya  | 25500.00
(5 rows)
postgres=# select sqrt(salary) from emp12;
       sqrt
------------------
```

```
 158.113883008419
  141.42135623731
 181.659021245849
 187.082869338697
 159.687194226713
(5 rows)


postgres=# select abs(salary) from emp12;
  abs
-------
 25000
 20000
 33000
 35000
 25500
(5 rows)
```

SQL- 5 DATE:

Create table emp with fields eno,ename,job,manager ,salary with eno as primary key .Insert values into the table.

```
postgres=# create table emp10(eno int not null primary key,ename varchar(20),job
 varchar(20),manager varchar(20),salary int);
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "emp10_pkey" for
table "emp10"
CREATE TABLE
                 ^
postgres=# insert into emp10 values (110,'rose','accountant','harize',20000);
INSERT 0 1
postgres=# insert into emp10 values(111,'mathew','accountant','harize',21000);
INSERT 0 1
postgres=# insert into emp10 values(113,'sona','clerk','deepthi',25000);
```

```
INSERT 0 1

postgres=# insert into emp10 values (114,'jack','pune','hari',24000);

INSERT 0 1

postgres=# insert into emp10 values (115,'manu','security','deepu',15000);

INSERT 0 1

postgres=# select *from  emp10;

 eno | ename  |    job     | manager | salary

-----+--------+------------+---------+--------

 110 | rose   | accountant | harize  |  20000

 111 | mathew | accountant | harize  |  21000

 113 | sona   | clerk      | deepthi |  25000

 114 | jack   | pune       | hari    |  24000

 115 | manu   | security   | deepu   |  15000

(5 rows)
```

1:Display ename,salary from emp who are getting salary more than average salary of the organization.

```
postgres=# select ename ,salary from emp10 where salary>(select avg(salary)from

emp10);

 ename | salary

-------+--------

 sona  |  25000

 jack  |  24000

(2 rows)
```

2:Add 20% DA as extra salary to all employees.Labelthe column as new salary.

```
postgres=# alter table emp10 add new_salary float;

ALTER TABLE

postgres=# update emp10 set new_salary=salary+(.2*salary);

UPDATE 5

postgres=# select *from emp10;

 eno | ename  |    job     | manager | salary | new_salary

-----+--------+------------+---------+--------+------------

 110 | rose   | accountant | harize  |  20000 |      24000

 111 | mathew | accountant | harize  |  21000 |      25200

 113 | sona   | clerk      | deepthi |  25000 |      30000

 114 | jack   | pune       | hari    |  24000 |      28800

 115 | manu   | security   | deepu   |  15000 |      18000

(5 rows)
```

3:Create a query to display the eno and ename for all employees who earn more than the average salary.sorts the results in descending order of salary.

```
postgres=# select eno,ename from emp10 where salary>(select avg(salary)from emp1

0)order by salary desc;

 eno | ename

-----+-------

 113 | sona

 114 | jack

(2 rows)
```

4:Create a view called emp_view based on the eno,ename from emp table change the heading for the enmae to employee.

```
postgres=# create view emp10_view as select eno,ename "employee" from emp10;

CREATE VIEW

postgres=# select * from emp10_view;

 eno | employee

-----+----------

 110 | rose

 111 | mathew

 113 | sona

 114 | jack

 115 | manu

(5 rows)
```

5:write a query that will display the eno and ename for all employees who work in a department with   ^   ^

```
postgres=# select eno,ename from emp10 where ename like '%t%';

 eno | ename

-----+--------

 111 | mathew

(1 row)

                                                  ^

postgres=# create sequence eno_sequence increment by 10 start with 60 maxvalue 2

00 no cycle;

CREATE SEQUENCE

postgres=# select * from eno_sequence;

 sequence_name | last_value | start_value | increment_by | max_value | min_value
 | cache_value | log_cnt | is_cycled | is_called
---------------+------------+-------------+--------------+-----------+----------
-+-------------+---------+-----------+-----------
 eno_sequence  |         60 |          60 |           10 |       200 |         1
 |           1 |       1 | f         | f
(1 row)


postgres=# select sequence_name ,max_value,increment_by,last_value from eno_sequ
ence;
 sequence_name | max_value | increment_by | last_value
---------------+-----------+--------------+------------
 eno_sequence  |       200 |           10 |         60
(1 row)
```

SQL-6 DATE:

create a table department with fields ename,salary,dno,designation,dname,place with dno as primary key.insert five values into the table.

```
postgres=# create table department10(ename varchar(20),salary float,dno int prim
ary key,designation varchar(20),dname varchar(20),place varchar(20));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "department10_pke
y" for table "department10"
CREATE TABLE
postgres=# insert into department10 values('ammu',10000,1,'clerk','software','ka
nnur');
INSERT 0 1
postgres=# insert into department10 values('achu',15000,2,'pune','hardware','coc
h');
INSERT 0 1
postgres=# insert into department10 values('ramu',20000,3,'accountant','hardware
','kollam');
INSERT 0 1
postgres=# insert into department10 values('raju',10000,4,'clerk','softwaree','k
asargod');
INSERT 0 1
postgres=# insert into department10 values('nagil',50000,5,'manager','hardware',
'kasargod');
INSERT 0 1
postgres=# select*from department10;
 ename | salary | dno | designation |   dname   |  place
-------+--------+-----+-------------+-----------+----------
 ammu  |  10000 |   1 | clerk       | software  | kannur
 achu  |  15000 |   2 | pune        | hardware  | coch
 ramu  |  20000 |   3 | accountant  | hardware  | kollam
 raju  |  10000 |   4 | clerk       | softwaree | kasargod
 nagil |  50000 |   5 | manager     | hardware  | kasargod
(5 rows)
```

1.write the queries using various character functions in ename field.

```
postgres=# select lower(ename)from department10;
 lower
-------
 ammu
 achu
 ramu
 raju
 nagil
(5 rows)


postgres=# select rtrim(ename)from department10;
 rtrim
-------
 ammu
 achu
 ramu
 raju
 nagil
(5 rows)
postgres=# select length(ename)from department10 where salary=1000
 length
--------
      4
      4
(2 rows)
postgres=# select upper(ename)from department10 where designation=
 upper
-------
 AMMU
 RAJU
(2 rows)
postgres=# select btrim(ename)from department10;
 btrim
```

```
-------
 ammu
 achu
 ramu
 raju
 nagil
(5 rows)
```

2.create a query to display the employee number and name for all employees who earn more than the average salary.sort the result in descending order.

```
postgres=# select dno,ename from department10 where salary>(select avg(salary)fr
om department10)order by salary desc;
 dno | ename
-----+-------
   5 | nagil
(1 row)
```

3.display all employees who got salary between1000 and 20000.

```
                                                   ^
postgres=# select ename from department10 where salary between 10000 and 20000;
 ename
-------
 ammu
 achu
 ramu
 raju
(4 rows)
```

4.display ename,salary,designatin for those who got salary more than 5000 or his designation is ‘clerk.

```
postgres=# select ename,salary,designation from department10 where salary>5000 o
r designation='clerk';
 ename | salary | designation
-------+--------+-------------
 ammu  |  10000 | clerk
 achu  |  15000 | pune
 ramu  |  20000 | accountant
 raju  |  10000 | clerk
 nagil |  50000 | manager
(5 rows)
```

5.display ename and designatin those who are not a clerk or manager.

```
                                                ^
postgres=# select ename,designation from department10 where designation<>'c
 and designation<>'manager';
 ename | designation
-------+-------------
 achu  | pune
 ramu  | accountant
(2 rows)
```

6.display the names of all employees where the third letter of their name is an ‘g’.

```
postgres=# select ename from department10 where ename like'__g%';
 ename
-------
 Nagil
```

SQL-7 DATE:

Create table loan with fields loanno,cname,cid,bname assigning suitable constraints.Insert 5 records into the table

```
postgres=# create table loan2(loan_no int not null,ename varchar(20),cid int,bna

me varchar(20));

CREATE TABLE

postgres=# insert into loan2 values(100,'merry',1,'kallar');

INSERT 0 1

postgres=# insert into loan2 values(101,'lilly',2,'kannur');

INSERT 0 1

postgres=# insert into loan2 values(102,'appu',3,'calicut');

INSERT 0 1

postgres=# insert into loan2 values(103,'doppu',4,'alapuzha');

INSERT 0 1

postgres=# insert into loan2 values(104,'arun',5,'kasargod');

INSERT 0 1

postgres=# select*from loan2;

 loan_no | ename | cid |  bname

---------+-------+-----+----------

     100 | merry |   1 | kallar

     101 | lilly |   2 | kannur
     102 | appu  |   3 | calicut
     103 | doppu |   4 | alapuzha
     104 | arun  |   5 | kasargod
(5 rows)
```

1.calculate Rs.150 extra for all customers having loan.the added loan amount will display in a new column.

```
postgres=# alter table loan2 add newloan int;
ALTER TABLE
postgres=# update loan2 set newloan=150;
UPDATE 5
postgres=# select*from loan2;
 loan_no | ename | cid |  bname   | newloan
---------+-------+-----+----------+---------
     100 | merry |   1 | kallar   |     150
     101 | lilly |   2 | kannur   |     150
     102 | appu  |   3 | calicut  |     150
     103 | doppu |   4 | alapuzha |     150
     104 | arun  |   5 | kasargod |     150
(5 rows)
```

2.add one more field amount to loan table.dispaly cname for cid=2;

```
postgres=# alter table loan add amount float;;
ALTER TABLE
               ^
postgres=# select ename from loan2 where cid=2;
 ename
-------
```

```
 lilly
(1 row)
```

3.create table depositor with fields cid and sccno.

```
postgres=# create table depositor(cid int,accno int);
CREATE TABLE
postgres=# insert into depositor values(1,10000);
INSERT 0 1
postgres=# insert into depositor values(2,10001);
INSERT 0 1
postgres=# insert into depositor values(3,10002);
INSERT 0 1
postgres=# insert into depositor values(4,10003);
INSERT 0 1
postgres=# insert into depositor values(5,10004);
INSERT 0 1
postgres=# select*from depositor;
 cid | accno
-----+-------
   1 | 10000
   2 | 10001
   3 | 10002
   4 | 10003
   5 | 10004
(5 rows)
```

5.display loanno and ename of a customer who is residing in kannur city.

```
               ^
postgres=# select loan_no,ename from loan2 where cid=2;
 loan_no | ename
---------+-------
     101 | lilly
(1 row)
```

6.dispaly all information from loan table for loanno,2,4,5.

```
postgres=# select*from loan2 where cid in(2,4,5);
 loan_no | ename | cid |  bname   | newloan
---------+-------+-----+----------+---------
     101 | lilly |   2 | kannur   |     150
     103 | doppu |   4 | alapuzha |     150
     104 | arun  |   5 | kasargod |     150
(3 rows)
```