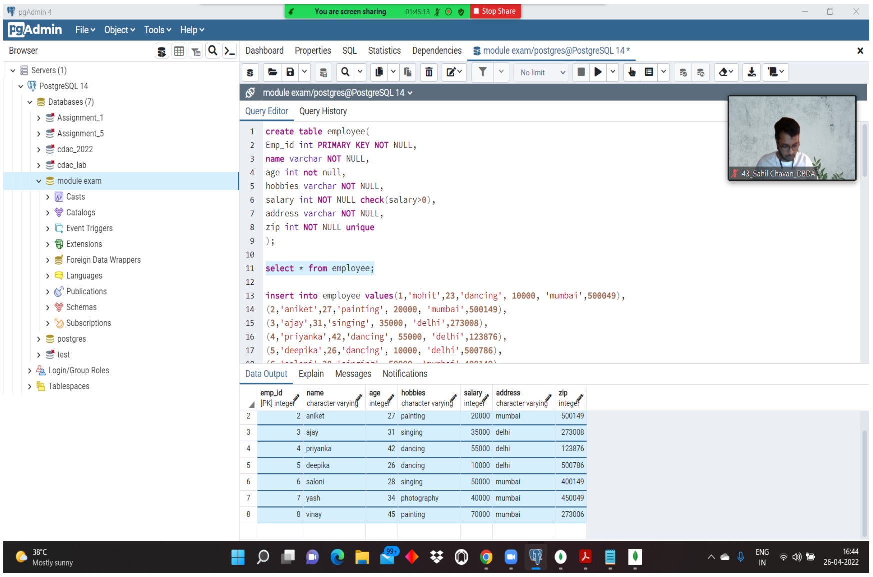
Q1 Create table employee,dept with following column and insert given data(3 Marks)

emp_id
Name - character varying
Age - Integer
hobbies character varying
salary integer
address character varying
zip integer

CONSTRAINT
Emp_id – Primary KEY
All columns not null
Salary should always greater than 0
Zip should be unique

ans-create table employee(
Emp_id int PRIMARY KEY NOT NULL,
name varchar NOT NULL,
age int not null,
hobbies varchar NOT NULL,
salary int NOT NULL check(salary>0),
address varchar NOT NULL,
zip int NOT NULL unique
);



```
create table employee(
Emp_id int PRIMARY KEY NOT NULL,
name varchar NOT NULL,
age int not null,
hobbies varchar NOT NULL,
salary int NOT NULL check(salary>0),
address varchar NOT NULL,
zip int NOT NULL unique
);

select * from employee;

insert into employee values(1,'mohit',23,'dancing', 10000, 'mumbai',500049),
(2,'aniket',27,'painting', 20000, 'mumbai',500149),
(3,'ajay',31,'singing', 35000, 'delhi',273008),
(4,'priyanka',42,'dancing', 55000, 'delhi',123876),
(5,'deepika',26,'dancing', 10000, 'delhi',500786),
(6,'saloni',28,'singing', 50000, 'mumbai',400149),
```

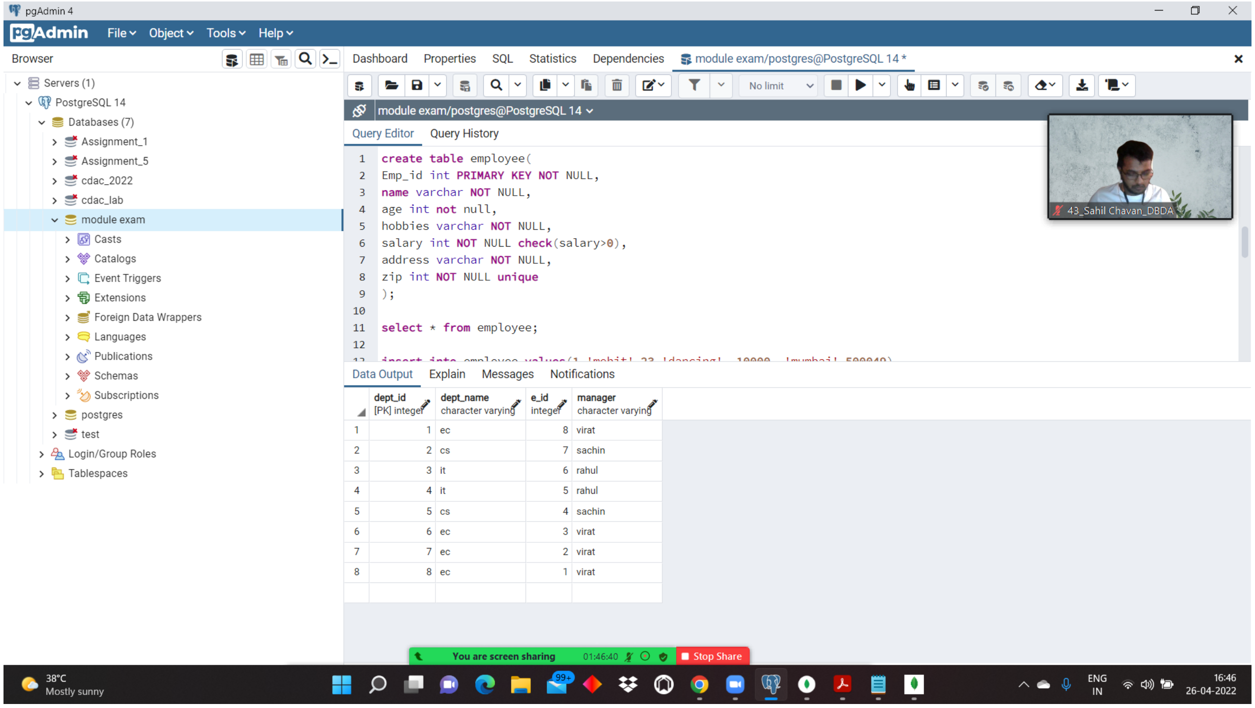
	emp_id [PK] integer	name character varying	age integer	hobbies character varying	salary integer	address character varying	zip integer
2	2	aniket	27	painting	20000	mumbai	500149
3	3	ajay	31	singing	35000	delhi	273008
4	4	priyanka	42	dancing	55000	delhi	123876
5	5	deepika	26	dancing	10000	delhi	500786
6	6	saloni	28	singing	50000	mumbai	400149
7	7	yash	34	photography	40000	mumbai	450049
8	8	vinay	45	painting	70000	mumbai	273006

Dept
dept_id integer

dept_name character varying
e_id integer
manager character varying

Constraints:
dept_id – Primary key
e_id - Foreign key to employee table (emp_id)
Q2 INSERT FOLLOWING DATA TO EMPLOYEE (3 Marks)
1,mohit,23,dancing, 10000, Mumbai,500049,
2,aniket,27,painting, 20000, mumbai,500149,
3,ajay,31,singing, 35000, delhi,273008,
4,priyanka,42,dancing, 55000, delhi,123876,
5,deepika,26,dancing, 10000, delhi,500786,
6,saloni,28,singing, 50000, Mumbai,400149,
7,yash,34,photography, 40000, Mumbai,450049,
8,vinay,45,painting, 70000, Mumbai,273006;
Dept table data
1,ec,8, virat,
2,cs,7, sachin,
3,it,6, rahul,
4,it,5, rahul,
5,cs,4, sachin,
6,ec,3, virat,
7,ec,2, virat,
8,ec,1, virat

```
ans-create table dept(
dept_id int primary key,
dept_name varchar,
e_id int references employee(emp_id),
manager varchar
);
```

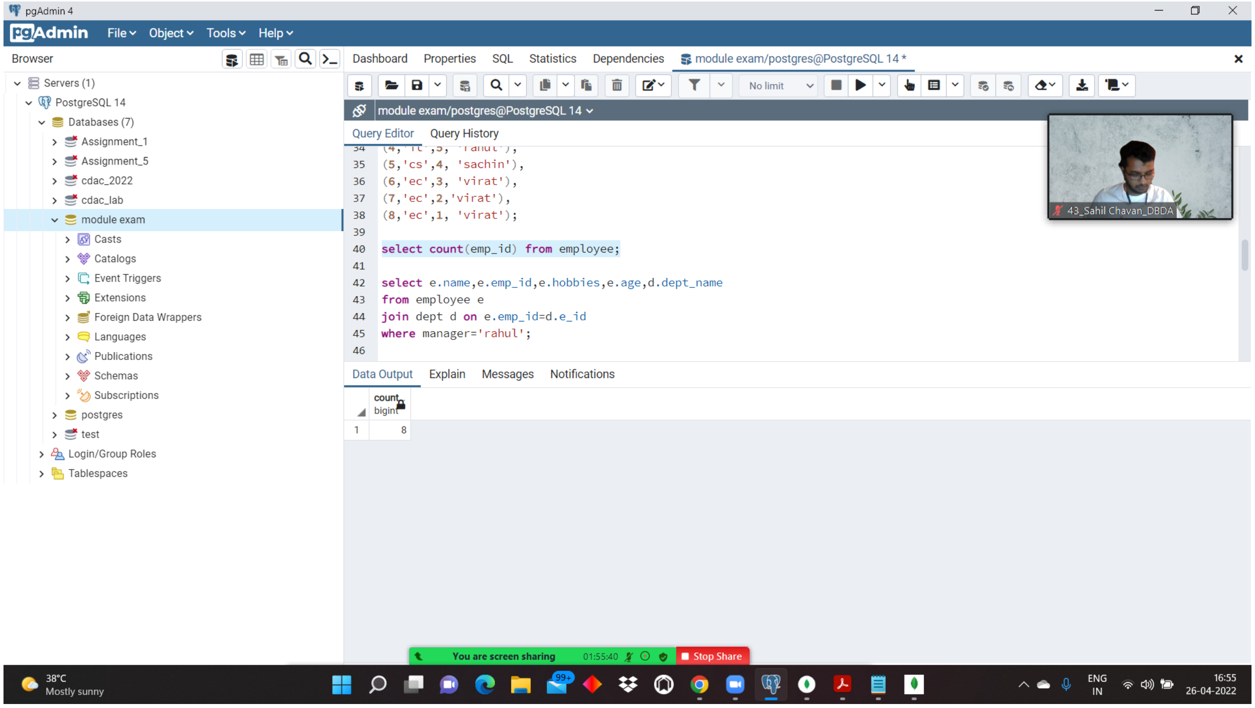


```sql
create table employee(
Emp_id int PRIMARY KEY NOT NULL,
name varchar NOT NULL,
age int not null,
hobbies varchar NOT NULL,
salary int NOT NULL check(salary>0),
address varchar NOT NULL,
zip int NOT NULL unique
);

select * from employee;

insert into employee values(1,'mohit',23,'dancing', 10000, 'mumbai',500049)
```

	dept_id [PK] integer	dept_name character varying	e_id integer	manager character varying
1	1	ec	8	virat
2	2	cs	7	sachin
3	3	it	6	rahul
4	4	it	5	rahul
5	5	cs	4	sachin
6	6	ec	3	virat
7	7	ec	2	virat
8	8	ec	1	virat

Write Down Following Queries (14 Marks)

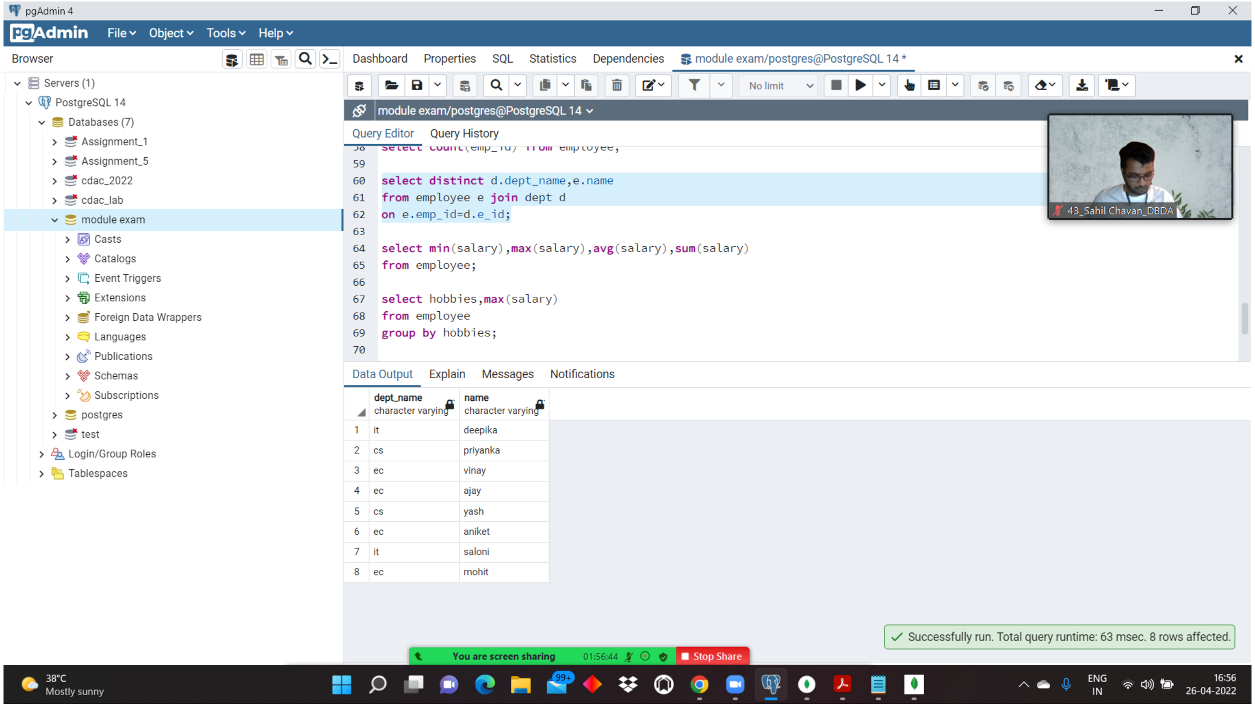
Write a Query to count No. of employees (2 Marks)



```sql
(4,'it',5, 'rahul'),
(5,'cs',4, 'sachin'),
(6,'ec',3, 'virat'),
(7,'ec',2,'virat'),
(8,'ec',1, 'virat');

select count(emp_id) from employee;

select e.name,e.emp_id,e.hobbies,e.age,d.dept_name
from employee e
join dept d on e.emp_id=d.e_id
where manager='rahul';

```

	count bigint
1	8

Write a Query to get unique department of employees (2 Marks)

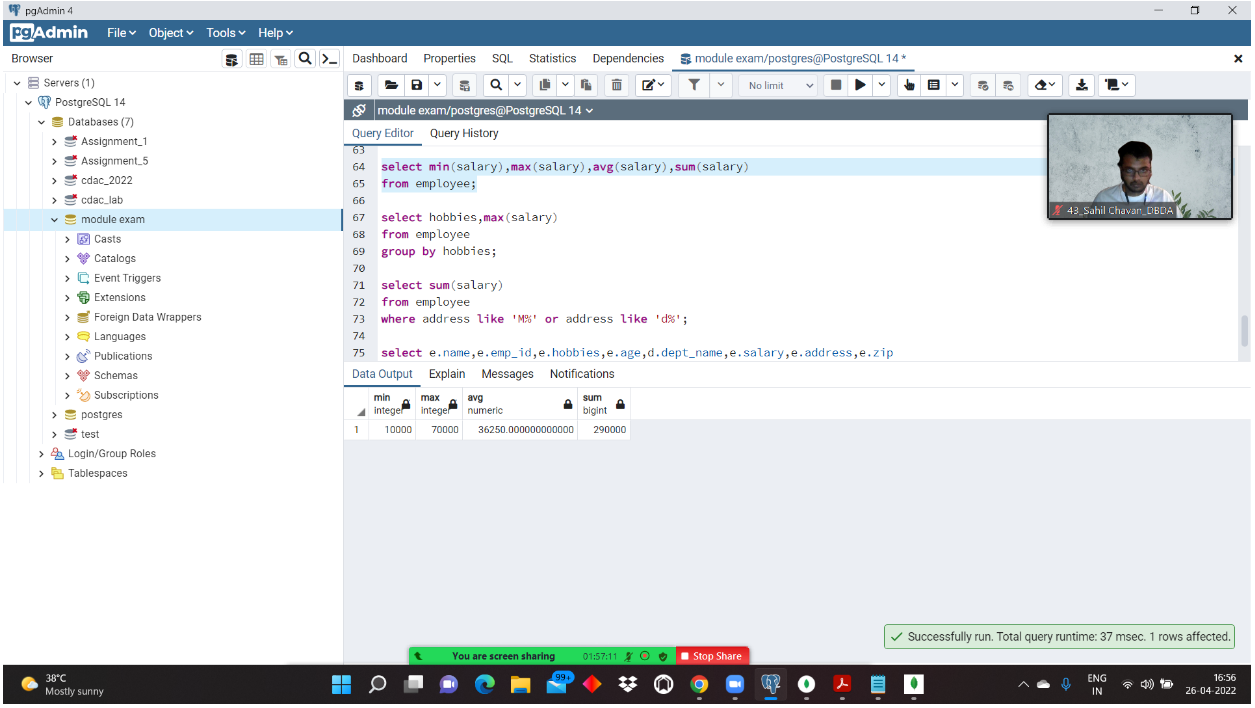


```sql
select count(emp_id) from employee;

select distinct d.dept_name,e.name
from employee e join dept d
on e.emp_id=d.e_id;

select min(salary),max(salary),avg(salary),sum(salary)
from employee;

select hobbies,max(salary)
from employee
group by hobbies;

```

	dept_name character varying	name character varying
1	it	deepika
2	cs	priyanka
3	ec	vinay
4	ec	ajay
5	cs	yash
6	ec	aniket
7	it	saloni
8	ec	mohit

✓ Successfully run. Total query runtime: 63 msec. 8 rows affected.

Write a Query to get min,max,avg,sum of salary for all employees (2 Marks)

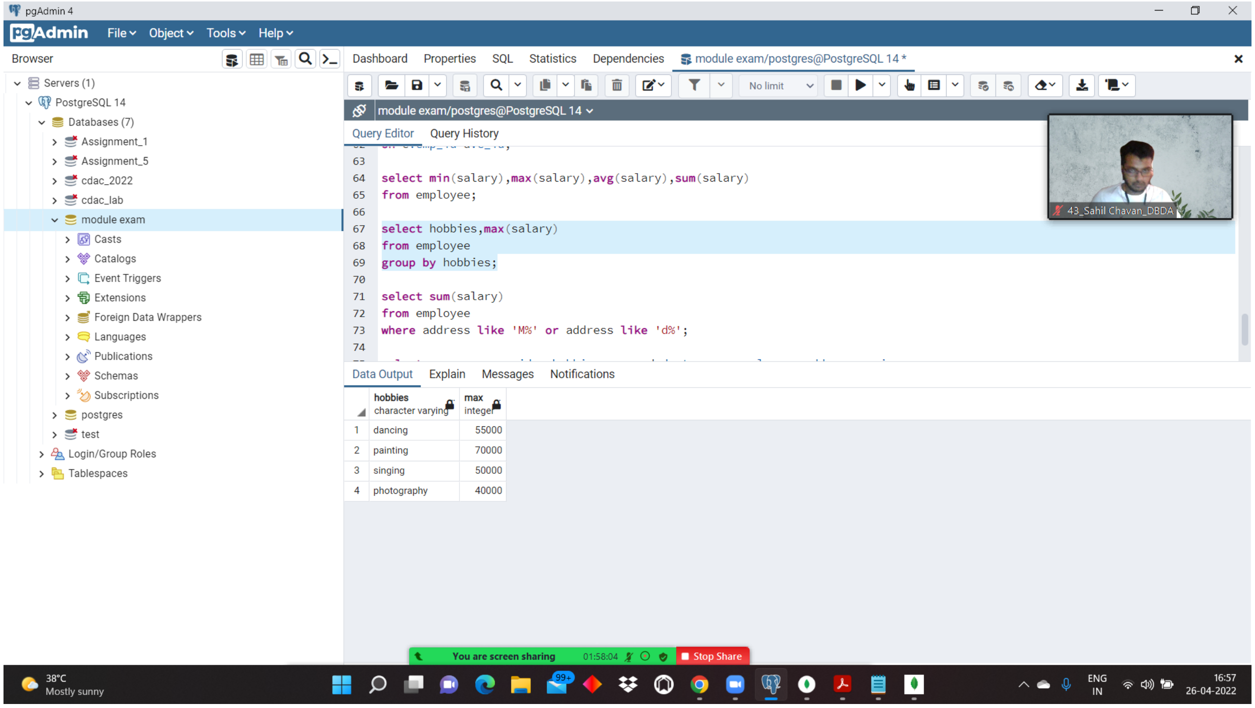


```sql

select min(salary),max(salary),avg(salary),sum(salary)
from employee;

select hobbies,max(salary)
from employee
group by hobbies;

select sum(salary)
from employee
where address like 'M%' or address like 'd%';

select e.name,e.emp_id,e.hobbies,e.age,d.dept_name,e.salary,e.address,e.zip
```

	min integer	max integer	avg numeric	sum bigint
1	10000	70000	36250.000000000000	290000

✓ Successfully run. Total query runtime: 37 msec. 1 rows affected.

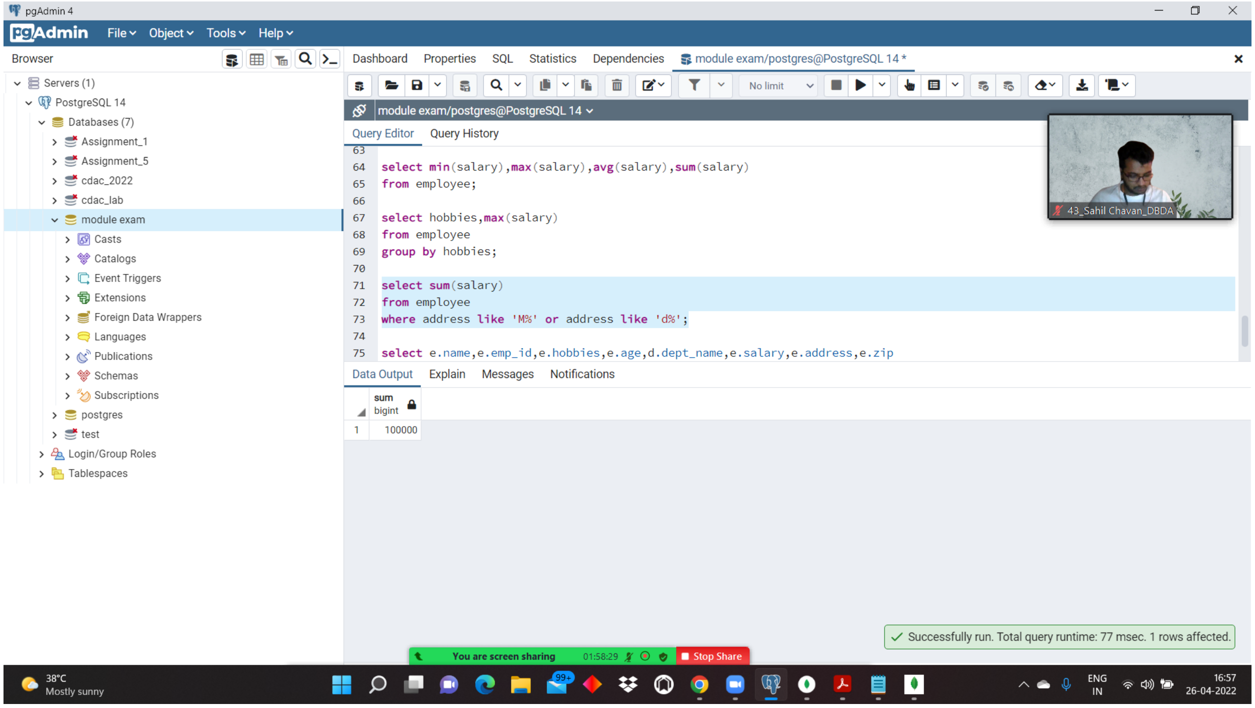
Write a Query to get highest salary of an individual based on hobbies (2 Marks)



```
select min(salary),max(salary),avg(salary),sum(salary)
from employee;

select hobbies,max(salary)
from employee
group by hobbies;

select sum(salary)
from employee
where address like 'M%' or address like 'd%';
```

	hobbies character varying	max integer
1	dancing	55000
2	painting	70000
3	singing	50000
4	photography	40000

Write a Query for sum of salary where address starts with 'M' or 'd' (2 Marks)

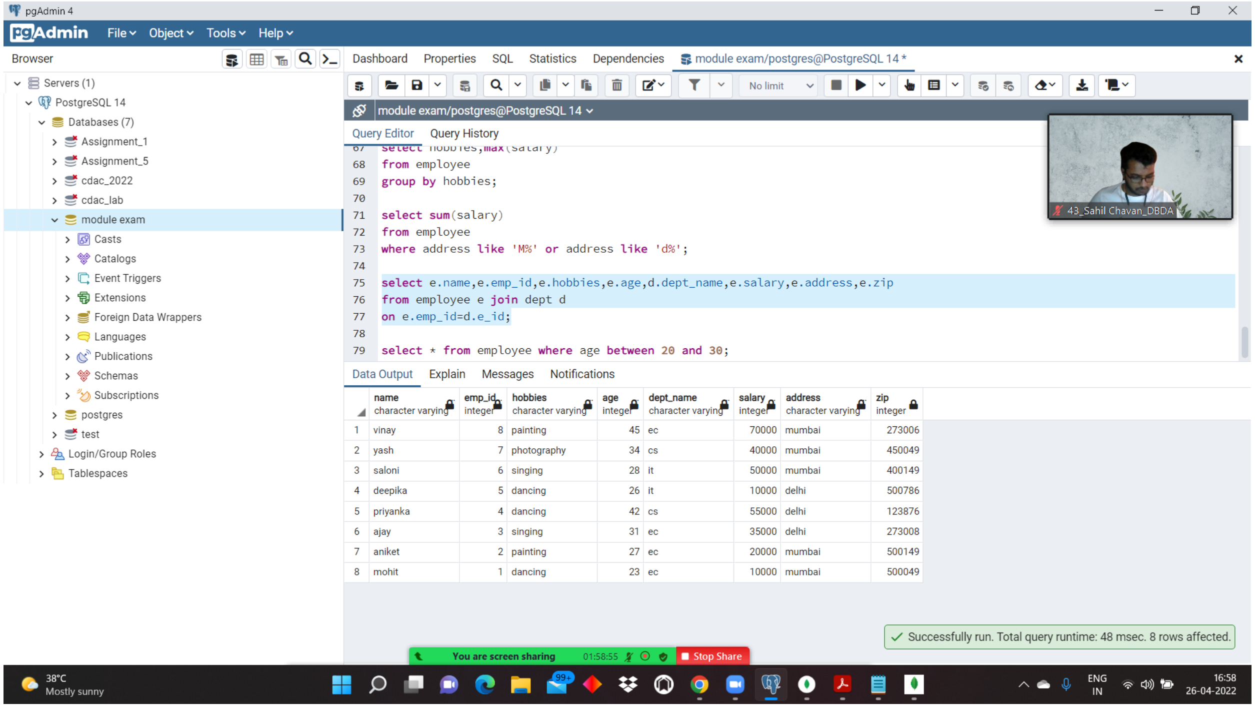


```
select min(salary),max(salary),avg(salary),sum(salary)
from employee;

select hobbies,max(salary)
from employee
group by hobbies;

select sum(salary)
from employee
where address like 'M%' or address like 'd%';

select e.name,e.emp_id,e.hobbies,e.age,d.dept_name,e.salary,e.address,e.zip
```

	sum bigint
1	100000

✓ Successfully run. Total query runtime: 77 msec. 1 rows affected.

Write a Query to Get all employee details with their department details(2 Marks)



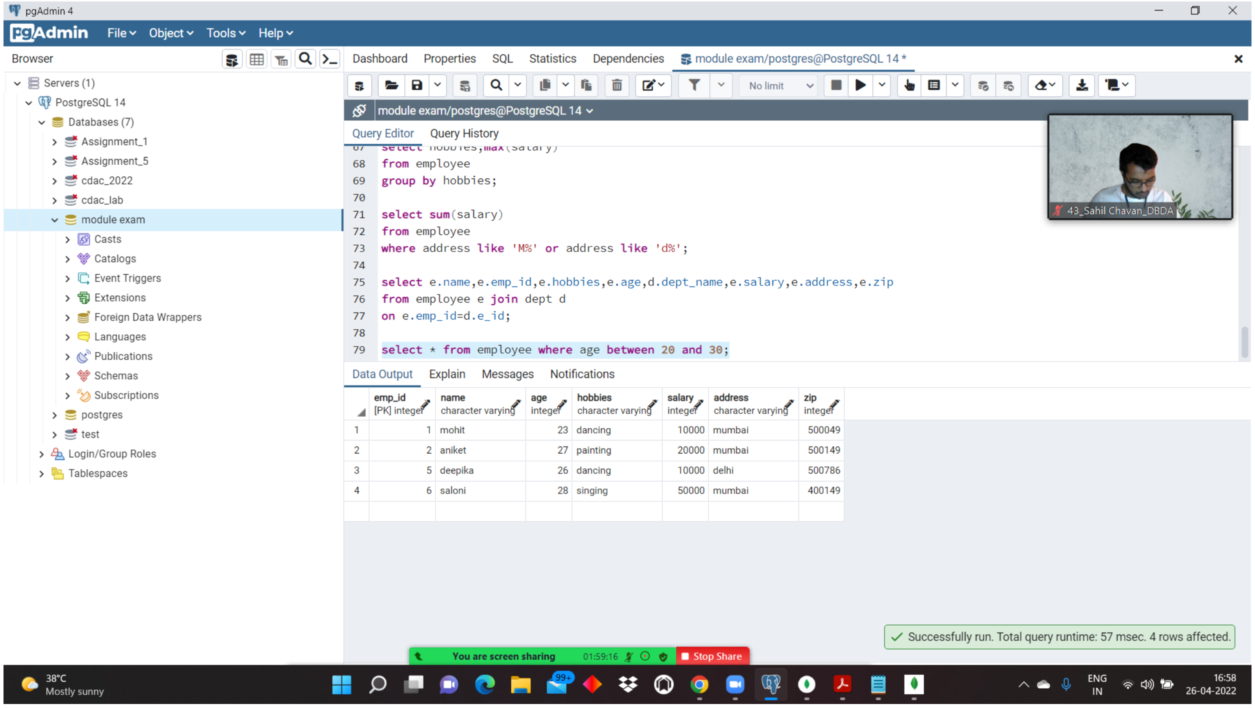
```
select hobbies,max(salary)
from employee
group by hobbies;

select sum(salary)
from employee
where address like 'M%' or address like 'd%';

select e.name,e.emp_id,e.hobbies,e.age,d.dept_name,e.salary,e.address,e.zip
from employee e join dept d
on e.emp_id=d.e_id;

select * from employee where age between 20 and 30;
```

	name character varying	emp_id integer	hobbies character varying	age integer	dept_name character varying	salary integer	address character varying	zip integer
1	vinay	8	painting	45	ec	70000	mumbai	273006
2	yash	7	photography	34	cs	40000	mumbai	450049
3	saloni	6	singing	28	it	50000	mumbai	400149
4	deepika	5	dancing	26	it	10000	delhi	500786
5	priyanka	4	dancing	42	cs	55000	delhi	123876
6	ajay	3	singing	31	ec	35000	delhi	273008
7	aniket	2	painting	27	ec	20000	mumbai	500149
8	mohit	1	dancing	23	ec	10000	mumbai	500049

✓ Successfully run. Total query runtime: 48 msec. 8 rows affected.

pgAdmin 4
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies module exam/postgres@PostgreSQL 14 *
module exam/postgres@PostgreSQL 14
Query Editor Query History
Data Output Explain Messages Notifications
43_Sahil Chavan_DBDA
You are screen sharing 01:58:55 Stop Share
38°C Mostly sunny
ENG IN 16:58 26-04-2022

Write a QUERY TO FIND employees age between 20 and 30 (2 Marks)



```
select hobbies,max(salary)
from employee
group by hobbies;

select sum(salary)
from employee
where address like 'M%' or address like 'd%';

select e.name,e.emp_id,e.hobbies,e.age,d.dept_name,e.salary,e.address,e.zip
from employee e join dept d
on e.emp_id=d.e_id;

select * from employee where age between 20 and 30;
```

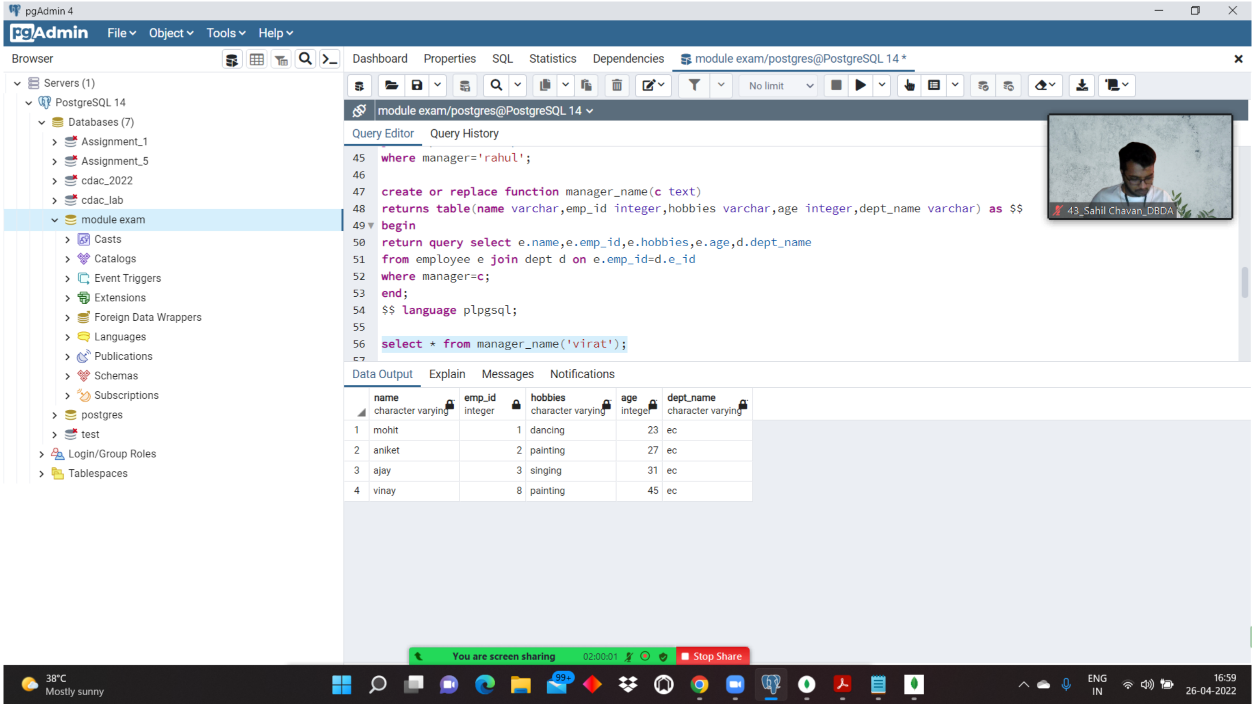
	emp_id [PK] integer	name character varying	age integer	hobbies character varying	salary integer	address character varying	zip integer
1	1	mohit	23	dancing	10000	mumbai	500049
2	2	aniket	27	painting	20000	mumbai	500149
3	5	deepika	26	dancing	10000	delhi	500786
4	6	saloni	28	singing	50000	mumbai	400149

✓ Successfully run. Total query runtime: 57 msec. 4 rows affected.

pgAdmin 4
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies module exam/postgres@PostgreSQL 14 *
module exam/postgres@PostgreSQL 14
Query Editor Query History
Data Output Explain Messages Notifications
43_Sahil Chavan_DBDA
You are screen sharing 01:59:16 Stop Share
38°C Mostly sunny
ENG IN 16:58 26-04-2022

Q3 Write a function to return name,emp_id,dept_name,hobbies,age by passing manager name (10 Marks)

MONGODB Section (10 Marks)



```
where manager='rahul';

create or replace function manager_name(c text)
returns table(name varchar,emp_id integer,hobbies varchar,age integer,dept_name varchar) as $$
begin
return query select e.name,e.emp_id,e.hobbies,e.age,d.dept_name
from employee e join dept d on e.emp_id=d.e_id
where manager=c;
end;
$$ language plpgsql;

select * from manager_name('virat');
```

	name character varying	emp_id integer	hobbies character varying	age integer	dept_name character varying
1	mohit	1	dancing	23	ec
2	aniket	2	painting	27	ec
3	ajay	3	singing	31	ec
4	vinay	8	painting	45	ec

CREATE MONGO DB COLLECTIONS with following details and insert data

--DB = mongo exam

--Collection = assignment,inventory

--assigment data

{ item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },
{ item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },
{ item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }

--inventory data

{ item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [14, 21] },
{ item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [14, 21] },
{ item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [14, 21] },
{ item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [22.85, 30] },
{ item: "postcard", qty: 45, tags: ["blue"], dim_cm: [10, 15.25] }

- use mongo exam

switched to db mongo

db.assignment.insertMany([{ item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } }, { item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } }, { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }])

```
db.inventory.insertMany([{ item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
... { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
... { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
... { item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [ 22.85, 30 ] },
... { item: "postcard", qty: 45, tags: ["blue"], dim_cm: [ 10, 15.25 ] }])
```

1. get assignment documents having tags = gray (2 Marks)

```
 db.assignment.find({tags:"gray"})
[
  {
    _id: ObjectId("6267ce7d105c89876319dc50"),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  }
]
```

2. Get inventory details whose dim_cm > 10 , sorted by qty descending order and print only 3 dcouments.(2 Marks)

```
db.inventory.find({dim_cm:{$gt:10}}).sort({qty:-1}).limit(3)
[
  {
    _id: ObjectId("6267cf5d105c89876319dc54"),
    item: 'paper',
    qty: 100,
    tags: [ 'red', 'blank', 'plain' ],
    dim_cm: [ 14, 21 ]
  },
  {
    _id: ObjectId("6267cf5d105c89876319dc55"),
    item: 'planner',
    qty: 75,
    tags: [ 'blank', 'red' ],
    dim_cm: [ 22.85, 30 ]
  },
  {
    _id: ObjectId("6267cf5d105c89876319dc53"),
    item: 'notebook',
    qty: 50,
    tags: [ 'red', 'blank' ],
    dim_cm: [ 14, 21 ]
  }
]
```

3. Create index on inventory in descending order of qty .(2 Marks)

```
db.inventory.createIndex({"qty":-1})
qty_-1
```

4. Querty to aggregare sum of qty in inventory collection(2 Marks)

```
 db.inventory.aggregate([{$group:{_id:"$qty",total:{$sum:1}}}])
[
  { _id: 100, total: 1 },
  { _id: 75, total: 1 },
  { _id: 25, total: 1 },
  { _id: 45, total: 1 },
  { _id: 50, total: 1 }
]
```

5. query to update inventory collection item name where qty:75 and dim_cm > 22. (2 Marks)

```
db.inventory.updateOne({$and:[{"qty":75},{dim_cm:{$gt:22}}]},{$set:{item:"name"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```