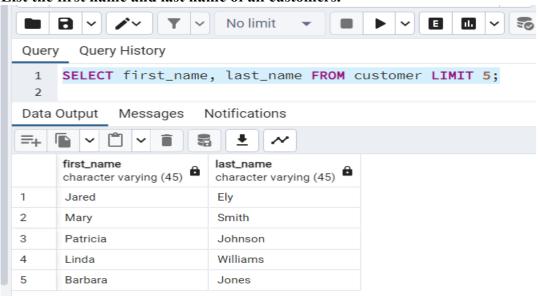
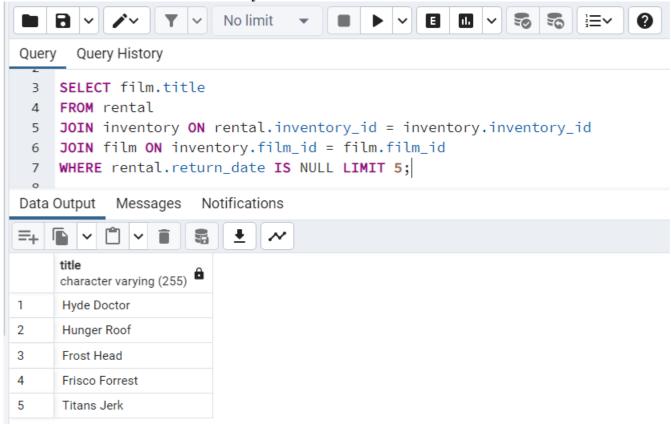
Name: Tanvi Gunwant Pohankar

Batch: Data Science & Data Analytics [G8-DS]

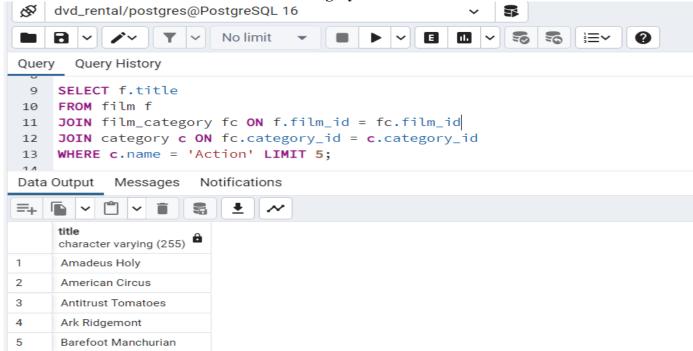
1) List the first name and last name of all customers.



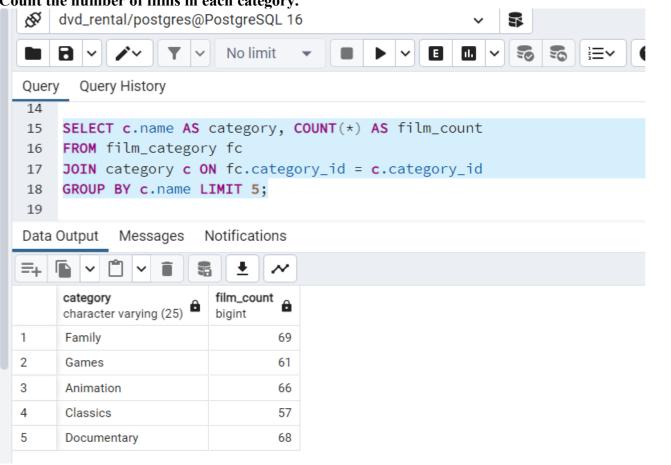
2) Find all the movies that are currently rented out.



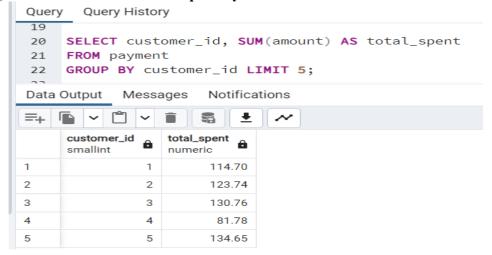
3) Show the titles of all movies in the 'Action' category.



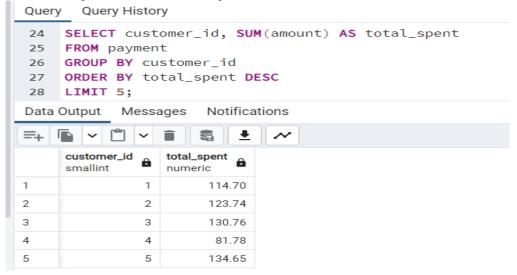
4) Count the number of films in each category.



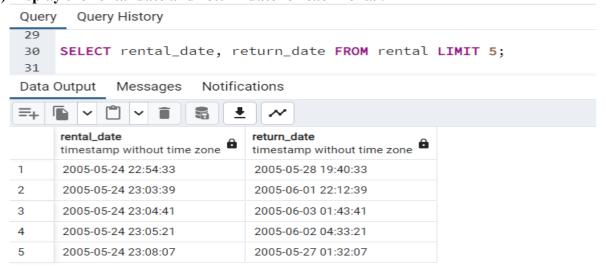
5) What is the total amount spent by each customer?



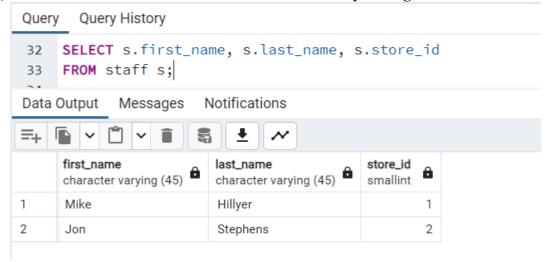
6) Find the top 5 customers who spent the most.



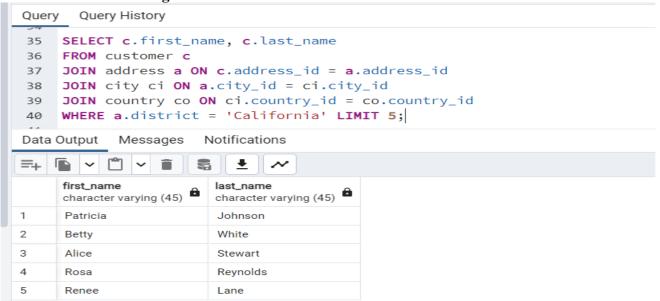
7) Display the rental date and return date for each rental.



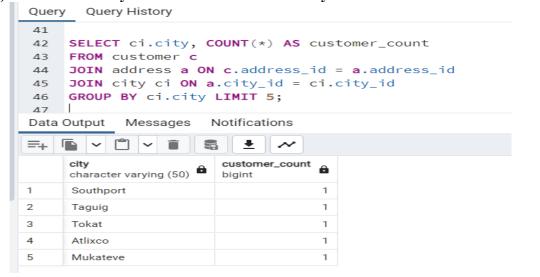
8) List the names of staff members and the stores they manage.



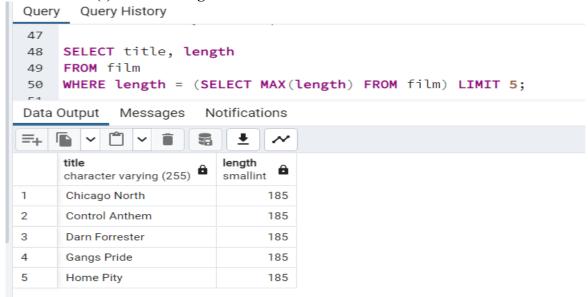
9) Find all customers living in 'California'.



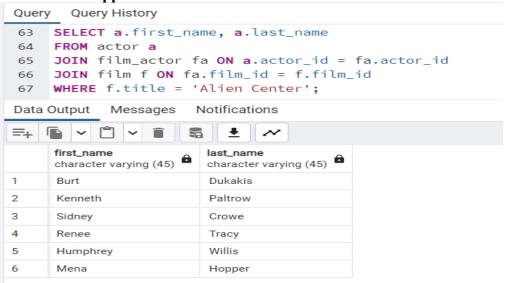
10) Count how many customers are from each city.



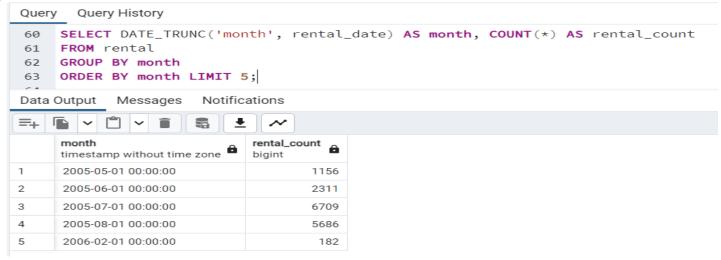
11) Find the film(s) with the longest duration.



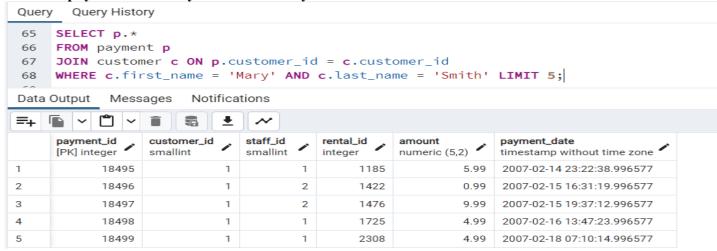
12) Which actors appear in the film titled 'Alien Center'?



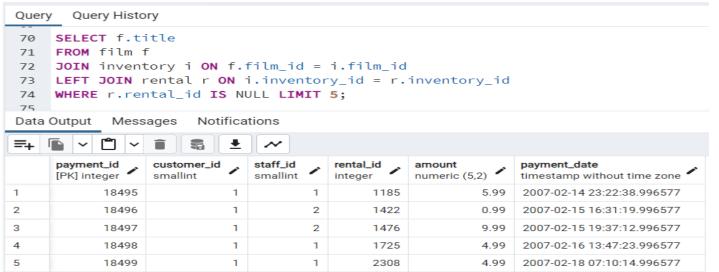
13) Find the number of rentals made each month.



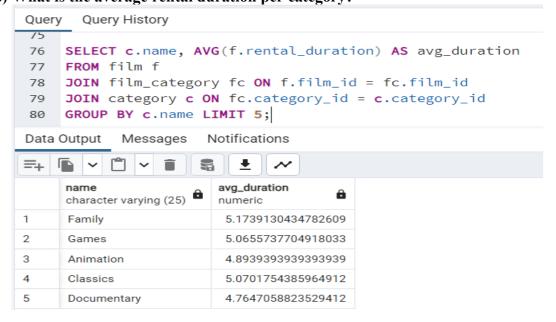
14) Show all payments made by customer 'Mary Smith'.



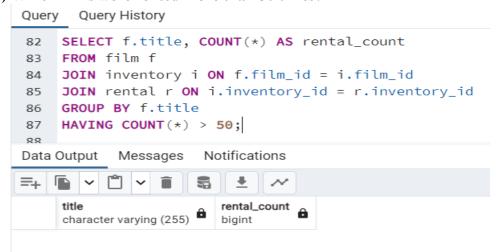
15) List all films that have never been rented.



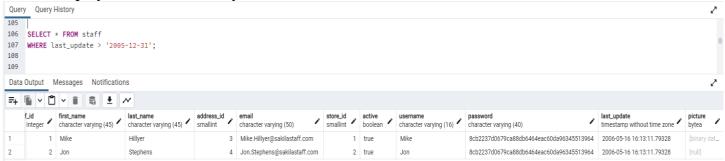
16) What is the average rental duration per category?



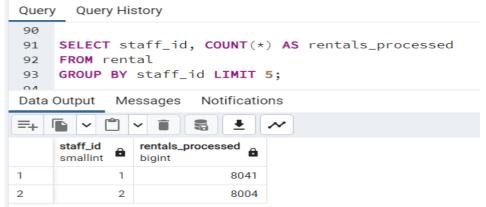
17) Which films were rented more than 50 times?



18) List all employees hired after the year 2005.



19) Show the number of rentals processed by each staff member.



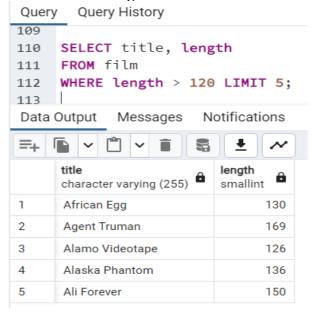
20) Display all customers who have not made any payments.

```
Query History
Query
9/1
     SELECT c.first_name, c.last_name
95
96
     FROM customer c
     LEFT JOIN payment p ON c.customer_id = p.customer_id
97
    WHERE p.payment_id IS NULL;
                       Notifications
Data Output
            Messages
=+
    #
                 first_name
                        last_name
     character varying (45)
                        character varying (45)
```

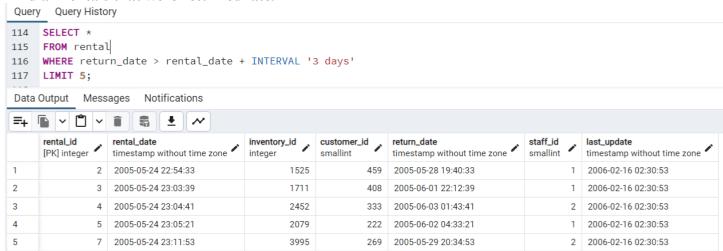
21) What is the most popular film (rented the most)?



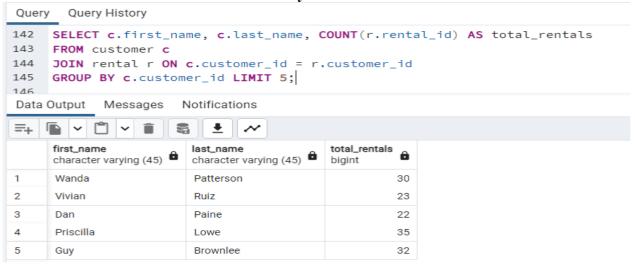
22) Show all films longer than 2 hours.



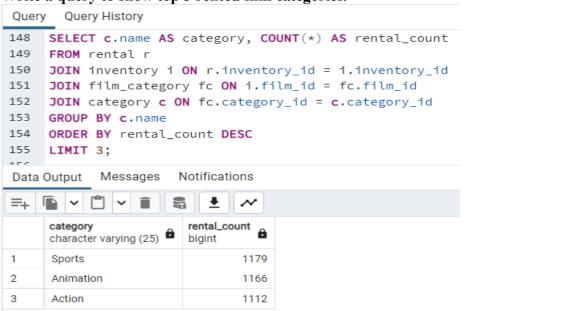
23) Find all rentals that were returned late.



24) List customers and the number of films they rented.



25) Write a query to show top 3 rented film categories.



26) Create a view that shows all customer names and their payment totals.

```
Query History
Query
111
     CREATE VIEW customer_payments AS
134
     SELECT c.first_name, c.last_name, SUM(p.amount) AS total_payment
135
     FROM customer c
136
     JOIN payment p ON c.customer_id = p.customer_id
137
     GROUP BY c.first_name, c.last_name LIMIT 5;
138
139
                       Notifications
Data Output
            Messages
CREATE VIEW
Query returned successfully in 82 msec.
```

27) Update a customer's email address given their ID.

```
Query Query History

139
140    UPDATE customer
141    SET email = 'new_email@example.com'
142    WHERE customer_id = 1; -- change to actual ID
143    Data Output    Messages    Notifications

UPDATE 1

Query returned successfully in 84 msec.
```

28) Insert a new actor into the actor table.

29) Delete all records from the rentals table where return_date is NULL.

```
Query Query History

DELETE FROM payment
WHERE rental_id IN (
SELECT rental_id FROM rental WHERE return_date IS NULL
);

DELETE FROM rental
WHERE return_date IS NULL;

Data Output Messages Notifications

DELETE 187

Query returned successfully in 49 msec.
```

30) Add a new column 'age' to the customer table.

```
Query Query History

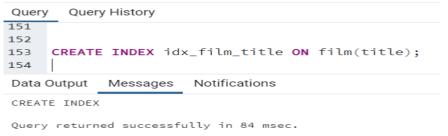
148
149    ALTER TABLE customer
150    ADD COLUMN age INTEGER;

Data Output    Messages    Notifications

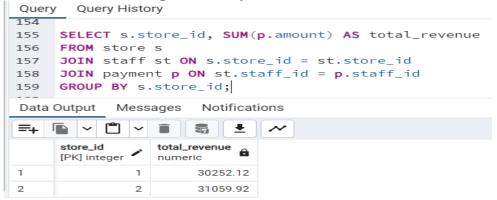
ALTER TABLE

Query returned successfully in 171 msec.
```

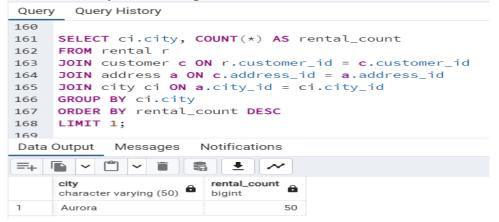
31) Create an index on the 'title' column of the film table.



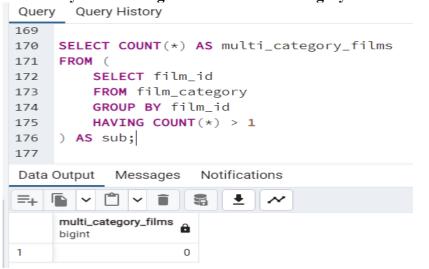
32) Find the total revenue generated by each store.



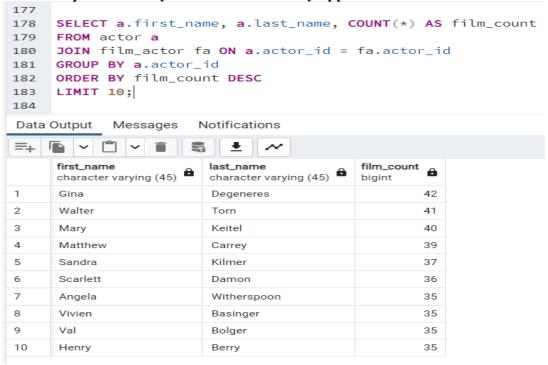
33) What is the city with the highest number of rentals?



34) How many films belong to more than one category?



35) List the top 10 actors by number of films they appeared in.



36) Retrieve the email addresses of customers who rented 'Matrix Revolutions'.

```
184
185
    SELECT DISTINCT c.email
186
    FROM customer c
     JOIN rental r ON c.customer_id = r.customer_id
     JOIN inventory i ON r.inventory_id = i.inventory_id
     JOIN film f ON i.film_id = f.film_id
190
     WHERE f.title = 'Matrix Revolutions';
191
            Messages
Data Output
                      Notifications
                          email
     character varying (50)
```

37) Create a stored function to return customer payment total given their ID.

```
CREATE OR REPLACE FUNCTION get_customer_total_payment(cid INT)
192
     RETURNS NUMERIC AS $$
193
194
     DECLARE
195
         total NUMERIC;
196 ▼ BEGIN
197
         SELECT SUM(amount) INTO total
         FROM payment
         WHERE customer_id = cid;
200
         RETURN total;
201
    END;
202 $$ LANGUAGE plpgsql;
Data Output
           Messages
                       Notifications
CREATE FUNCTION
Query returned successfully in 96 msec.
```

38) Begin a transaction that updates stock and inserts a rental record.

```
Query Query History

236

BEGIN;

237

238

UPDATE inventory

SET last_update = CURRENT_TIMESTAMP

240

WHERE inventory_id = 1;

241

242

INSERT INTO rental (rental_date, inventory_id, customer_id, return_date, staff_id, last_update)

VALUES (CURRENT_TIMESTAMP, 1, 1, NULL, 1, CURRENT_TIMESTAMP);

244

245

COMMIT;

Data Output Messages Notifications

COMMIT

Query returned successfully in 73 msec.
```

39) Show the customers who rented films in both 'Action' and 'Comedy' categories.



	first_name character varying (45)	last_name character varying (45)
1	Sue	Peters
2	Kimberly	Lee
3	Hilda	Hopkins
4	Colleen	Burton
5	Jeremy	Hurtado
6	Mitchell	Westmoreland
7	Maria	Miller

40) Find actors who have never acted in a film.

