POSTGREY-SQL PROJECT : SURE PROED

Name : Tanvi Gunwant Pohankar
Batch : Data Science & Data Analytics [G8-DS]

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

No limit

Query | Query History

```
SELECT first_name, last_name FROM customer LIMIT 5;
```

Data Output | Messages | Notifications

	first_name character varying (45)	last_name character varying (45)
1	Jared	Ely
2	Mary	Smith
3	Patricia	Johnson
4	Linda	Williams
5	Barbara	Jones

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

No limit

Query | Query History

```
SELECT film.title
FROM rental
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
WHERE rental.return_date IS NULL LIMIT 5;
```

Data Output | Messages | Notifications

	title character varying (255)
1	Hyde Doctor
2	Hunger Roof
3	Frost Head
4	Frisco Forrest
5	Titans Jerk

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

dvd_rental/postgres@PostgreSQL 16

No limit

Query | Query History

```sql
SELECT f.title
FROM film f
JOIN film_category fc ON f.film_id = fc.film_id
JOIN category c ON fc.category_id = c.category_id
WHERE c.name = 'Action' LIMIT 5;
```

Data Output | Messages | Notifications

	title character varying (255)
1	Amadeus Holy
2	American Circus
3	Antitrust Tomatoes
4	Ark Ridgemont
5	Barefoot Manchurian

4) Count the number of films in each category.

dvd_rental/postgres@PostgreSQL 16

No limit

Query | Query History

```sql
SELECT c.name AS category, COUNT(*) AS film_count
FROM film_category fc
JOIN category c ON fc.category_id = c.category_id
GROUP BY c.name LIMIT 5;
```

Data Output | Messages | Notifications

	category character varying (25)	film_count bigint
1	Family	69
2	Games	61
3	Animation	66
4	Classics	57
5	Documentary	68

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

Query Query History

```
SELECT customer_id, SUM(amount) AS total_spent
FROM payment
GROUP BY customer_id LIMIT 5;
```

Data Output Messages Notifications

	customer_id smallint	total_spent numeric
1	1	114.70
2	2	123.74
3	3	130.76
4	4	81.78
5	5	134.65

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

Query Query History

```
SELECT customer_id, SUM(amount) AS total_spent
FROM payment
GROUP BY customer_id
ORDER BY total_spent DESC
LIMIT 5;
```

Data Output Messages Notifications

	customer_id smallint	total_spent numeric
1	1	114.70
2	2	123.74
3	3	130.76
4	4	81.78
5	5	134.65

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

Query Query History

```
SELECT rental_date, return_date FROM rental LIMIT 5;
```

Data Output Messages Notifications

	rental_date timestamp without time zone	return_date timestamp without time zone
1	2005-05-24 22:54:33	2005-05-28 19:40:33
2	2005-05-24 23:03:39	2005-06-01 22:12:39
3	2005-05-24 23:04:41	2005-06-03 01:43:41
4	2005-05-24 23:05:21	2005-06-02 04:33:21
5	2005-05-24 23:08:07	2005-05-27 01:32:07

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

Query Query History

```
SELECT s.first_name, s.last_name, s.store_id
FROM staff s;
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)	store_id smallint
1	Mike	Hillyer	1
2	Jon	Stephens	2

9) Find all customers living in 'California'.

Query Query History

```
SELECT c.first_name, c.last_name
FROM customer c
JOIN address a ON c.address_id = a.address_id
JOIN city ci ON a.city_id = ci.city_id
JOIN country co ON ci.country_id = co.country_id
WHERE a.district = 'California' LIMIT 5;
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)
1	Patricia	Johnson
2	Betty	White
3	Alice	Stewart
4	Rosa	Reynolds
5	Renee	Lane

10) Count how many customers are from each city.

Query Query History

```
SELECT ci.city, COUNT(*) AS customer_count
FROM customer c
JOIN address a ON c.address_id = a.address_id
JOIN city ci ON a.city_id = ci.city_id
GROUP BY ci.city LIMIT 5;
```

Data Output Messages Notifications

	city character varying (50)	customer_count bigint
1	Southport	1
2	Taguig	1
3	Tokat	1
4	Atlixco	1
5	Mukateve	1

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

Query Query History

```
SELECT title, length
FROM film
WHERE length = (SELECT MAX(length) FROM film) LIMIT 5;
```

Data Output Messages Notifications

	title character varying (255)	length smallint
1	Chicago North	185
2	Control Anthem	185
3	Darn Forrester	185
4	Gangs Pride	185
5	Home Pity	185

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

Query Query History

```
SELECT a.first_name, a.last_name
FROM actor a
JOIN film_actor fa ON a.actor_id = fa.actor_id
JOIN film f ON fa.film_id = f.film_id
WHERE f.title = 'Alien Center';
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)
1	Burt	Dukakis
2	Kenneth	Paltrow
3	Sidney	Crowe
4	Renee	Tracy
5	Humphrey	Willis
6	Mena	Hopper

13) Find the number of rentals made each month.

Query Query History

```
SELECT DATE_TRUNC('month', rental_date) AS month, COUNT(*) AS rental_count
FROM rental
GROUP BY month
ORDER BY month LIMIT 5;
```

Data Output Messages Notifications

	month timestamp without time zone	rental_count bigint
1	2005-05-01 00:00:00	1156
2	2005-06-01 00:00:00	2311
3	2005-07-01 00:00:00	6709
4	2005-08-01 00:00:00	5686
5	2006-02-01 00:00:00	182

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

Query Query History

```
SELECT p.*
FROM payment p
JOIN customer c ON p.customer_id = c.customer_id
WHERE c.first_name = 'Mary' AND c.last_name = 'Smith' LIMIT 5;
```

Data Output Messages Notifications

	payment_id [PK] integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp without time zone
1	18495	1	1	1185	5.99	2007-02-14 23:22:38.996577
2	18496	1	2	1422	0.99	2007-02-15 16:31:19.996577
3	18497	1	2	1476	9.99	2007-02-15 19:37:12.996577
4	18498	1	1	1725	4.99	2007-02-16 13:47:23.996577
5	18499	1	1	2308	4.99	2007-02-18 07:10:14.996577

15) List all films that have never been rented.

Query Query History

```
SELECT f.title
FROM film f
JOIN inventory i ON f.film_id = i.film_id
LEFT JOIN rental r ON i.inventory_id = r.inventory_id
WHERE r.rental_id IS NULL LIMIT 5;
```

Data Output Messages Notifications

	payment_id [PK] integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp without time zone
1	18495	1	1	1185	5.99	2007-02-14 23:22:38.996577
2	18496	1	2	1422	0.99	2007-02-15 16:31:19.996577
3	18497	1	2	1476	9.99	2007-02-15 19:37:12.996577
4	18498	1	1	1725	4.99	2007-02-16 13:47:23.996577
5	18499	1	1	2308	4.99	2007-02-18 07:10:14.996577

16) What is the average rental duration per category?

Query Query History

```
SELECT c.name, AVG(f.rental_duration) AS avg_duration
FROM film f
JOIN film_category fc ON f.film_id = fc.film_id
JOIN category c ON fc.category_id = c.category_id
GROUP BY c.name LIMIT 5;
```

Data Output Messages Notifications

	name character varying (25)	avg_duration numeric
1	Family	5.1739130434782609
2	Games	5.0655737704918033
3	Animation	4.8939393939393939
4	Classics	5.0701754385964912
5	Documentary	4.7647058823529412

17) Which films were rented more than 50 times?

Query Query History

```sql
SELECT f.title, COUNT(*) AS rental_count
FROM film f
JOIN inventory i ON f.film_id = i.film_id
JOIN rental r ON i.inventory_id = r.inventory_id
GROUP BY f.title
HAVING COUNT(*) > 50;
```

Data Output Messages Notifications

title character varying (255)	rental_count bigint

18) List all employees hired after the year 2005.

Query Query History

```sql
SELECT * FROM staff
WHERE last_update > '2005-12-31';
```

Data Output Messages Notifications

	f_id integer	first_name character varying (45)	last_name character varying (45)	address_id smallint	email character varying (50)	store_id smallint	active boolean	username character varying (16)	password character varying (40)	last_update timestamp without time zone	picture bytea
1	1	Mike	Hillyer	3	Mike.Hillyer@sakilastaff.com	1	true	Mike	8cb2237d0679ca88db6464eac60da96345513964	2006-05-16 16:13:11.79328	[binary dat...
2	2	Jon	Stephens	4	Jon.Stephens@sakilastaff.com	2	true	Jon	8cb2237d0679ca88db6464eac60da96345513964	2006-05-16 16:13:11.79328	[null]

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

Query Query History

```sql
SELECT staff_id, COUNT(*) AS rentals_processed
FROM rental
GROUP BY staff_id LIMIT 5;
```

Data Output Messages Notifications

	staff_id smallint	rentals_processed bigint
1	1	8041
2	2	8004

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

Query Query History

```sql
SELECT c.first_name, c.last_name
FROM customer c
LEFT JOIN payment p ON c.customer_id = p.customer_id
WHERE p.payment_id IS NULL;
```

Data Output Messages Notifications

first_name character varying (45)	last_name character varying (45)

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

Query Query History

```
SELECT f.title, COUNT(*) AS rental_count
FROM film f
JOIN inventory i ON f.film_id = i.film_id
JOIN rental r ON i.inventory_id = r.inventory_id
GROUP BY f.title
ORDER BY rental_count DESC
LIMIT 1;
```

Data Output Messages Notifications

	title character varying (255)	rental_count bigint
1	Bucket Brotherhood	34

22) Show all films longer than 2 hours.

Query Query History

```
SELECT title, length
FROM film
WHERE length > 120 LIMIT 5;
```

Data Output Messages Notifications

	title character varying (255)	length smallint
1	African Egg	130
2	Agent Truman	169
3	Alamo Videotape	126
4	Alaska Phantom	136
5	Ali Forever	150

23) Find all rentals that were returned late.

Query Query History

```
SELECT *
FROM rental
WHERE return_date > rental_date + INTERVAL '3 days'
LIMIT 5;
```

Data Output Messages Notifications

	rental_id [PK] integer	rental_date timestamp without time zone	inventory_id integer	customer_id smallint	return_date timestamp without time zone	staff_id smallint	last_update timestamp without time zone
1	2	2005-05-24 22:54:33	1525	459	2005-05-28 19:40:33	1	2006-02-16 02:30:53
2	3	2005-05-24 23:03:39	1711	408	2005-06-01 22:12:39	1	2006-02-16 02:30:53
3	4	2005-05-24 23:04:41	2452	333	2005-06-03 01:43:41	2	2006-02-16 02:30:53
4	5	2005-05-24 23:05:21	2079	222	2005-06-02 04:33:21	1	2006-02-16 02:30:53
5	7	2005-05-24 23:11:53	3995	269	2005-05-29 20:34:53	2	2006-02-16 02:30:53

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

Query Query History

```
SELECT c.first_name, c.last_name, COUNT(r.rental_id) AS total_rentals
FROM customer c
JOIN rental r ON c.customer_id = r.customer_id
GROUP BY c.customer_id LIMIT 5;
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)	total_rentals bigint
1	Wanda	Patterson	30
2	Vivian	Ruiz	23
3	Dan	Paine	22
4	Priscilla	Lowe	35
5	Guy	Brownlee	32

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

Query Query History

```
SELECT c.name AS category, COUNT(*) AS rental_count
FROM rental r
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film_category fc ON i.film_id = fc.film_id
JOIN category c ON fc.category_id = c.category_id
GROUP BY c.name
ORDER BY rental_count DESC
LIMIT 3;
```

Data Output Messages Notifications

	category character varying (25)	rental_count bigint
1	Sports	1179
2	Animation	1166
3	Action	1112

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

Query Query History

```
CREATE VIEW customer_payments AS
SELECT c.first_name, c.last_name, SUM(p.amount) AS total_payment
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY c.first_name, c.last_name LIMIT 5;
```

Data Output Messages Notifications

```
CREATE VIEW

Query returned successfully in 82 msec.
```

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

Query Query History

```
UPDATE customer
SET email = 'new_email@example.com'
WHERE customer_id = 1; -- change to actual ID
```

Data Output Messages Notifications

```
UPDATE 1

Query returned successfully in 84 msec.
```

28) Insert a new actor into the actor table.

Query Query History

```
INSERT INTO actor (first_name, last_name, last_update)
VALUES ('John', 'Doe', CURRENT_TIMESTAMP);
```

Data Output Messages Notifications

```
INSERT 0 1

Query returned successfully in 146 msec.
```

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

```
ALTER TABLE customer
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.

Query Query History

```sql
CREATE INDEX idx_film_title ON film(title);
```

Data Output Messages Notifications

```
CREATE INDEX

Query returned successfully in 84 msec.
```

32) Find the total revenue generated by each store.

Query Query History

```sql
SELECT s.store_id, SUM(p.amount) AS total_revenue
FROM store s
JOIN staff st ON s.store_id = st.store_id
JOIN payment p ON st.staff_id = p.staff_id
GROUP BY s.store_id;
```

Data Output Messages Notifications

	store_id [PK] integer	total_revenue numeric
1	1	30252.12
2	2	31059.92

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

Query Query History

```sql
SELECT ci.city, COUNT(*) AS rental_count
FROM rental r
JOIN customer c ON r.customer_id = c.customer_id
JOIN address a ON c.address_id = a.address_id
JOIN city ci ON a.city_id = ci.city_id
GROUP BY ci.city
ORDER BY rental_count DESC
LIMIT 1;
```

Data Output Messages Notifications

	city character varying (50)	rental_count bigint
1	Aurora	50

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

Query Query History

```sql
SELECT COUNT(*) AS multi_category_films
FROM (
    SELECT film_id
    FROM film_category
    GROUP BY film_id
    HAVING COUNT(*) > 1
) AS sub;
```

Data Output Messages Notifications

	multi_category_films bigint
1	0

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

```
SELECT a.first_name, a.last_name, COUNT(*) AS film_count
FROM actor a
JOIN film_actor fa ON a.actor_id = fa.actor_id
GROUP BY a.actor_id
ORDER BY film_count DESC
LIMIT 10;
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)	film_count bigint
1	Gina	Degeneres	42
2	Walter	Torn	41
3	Mary	Keitel	40
4	Matthew	Carrey	39
5	Sandra	Kilmer	37
6	Scarlett	Damon	36
7	Angela	Witherspoon	35
8	Vivien	Basinger	35
9	Val	Bolger	35
10	Henry	Berry	35

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

```
SELECT DISTINCT c.email
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
WHERE f.title = 'Matrix Revolutions';
```

Data Output Messages 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)
RETURNS NUMERIC AS $$
DECLARE
    total NUMERIC;
BEGIN
    SELECT SUM(amount) INTO total
    FROM payment
    WHERE customer_id = cid;
    RETURN total;
END;
$$ 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

```
BEGIN;

UPDATE inventory
SET last_update = CURRENT_TIMESTAMP
WHERE inventory_id = 1;

INSERT INTO rental (rental_date, inventory_id, customer_id, return_date, staff_id, last_update)
VALUES (CURRENT_TIMESTAMP, 1, 1, NULL, 1, CURRENT_TIMESTAMP);

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.

Query Query History

```
SELECT DISTINCT c.first_name, c.last_name
FROM customer c
WHERE c.customer_id IN (
    SELECT customer_id
    FROM rental r
    JOIN inventory i ON r.inventory_id = i.inventory_id
    JOIN film_category fc ON i.film_id = fc.film_id
    WHERE fc.category_id = (SELECT category_id FROM category WHERE name = 'Action')
)
AND c.customer_id IN (
    SELECT customer_id
    FROM rental r
    JOIN inventory i ON r.inventory_id = i.inventory_id
    JOIN film_category fc ON i.film_id = fc.film_id
    WHERE fc.category_id = (SELECT category_id FROM category WHERE name = 'Comedy')
);
```

Data Output Messages Notifications

	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

Total rows: 419 of 419 Query complete 00:00:00.084

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

Query Query History

```
SELECT a.first_name, a.last_name
FROM actor a
LEFT JOIN film_actor fa ON a.actor_id = fa.actor_id
WHERE fa.film_id IS NULL;
```

Data Output Messages Notifications

	first_name character varying (45)	last_name character varying (45)
1	John	Doe
2	John	Doe