PostgreSQL Project

- By Harsh Vats

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

QUERY: - SELECT first_name, last_name FROM customer;

	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
6	Elizabeth	Brown
7	Jennifer	Davis

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

QUERY: - 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;

	customer_id [PK] integer	first_name character varying (45)	last_name character varying (45)	total_spent numeric
1	184	Vivian	Ruiz	80.80
2	87	Wanda	Patterson	137.72
3	477	Dan	Paine	106.79
4	273	Priscilla	Lowe	130.72
5	550	Guy	Brownlee	151.69
6	51	Alice	Stewart	123.70

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

QUERY: - 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';

	category character varying (25)	film_count bigint
1	Family	69
2	Games	61
3	Animation	66
4	Classics	57
5	Documentary	68
6	New	63

Ques.4. Count the number of films in each category.

QUERY: - 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;

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

Ques.5. What is the total amount spent by each customer?

QUERY: - SELECT c.customer_id, c.first_name, c.last_name, SUM(p.amount) AS total_spent

FROM customer c

JOIN payment p ON c.customer_id = p.customer_id

GROUP BY c.customer_id;

	customer_id [PK] integer	first_name character varying (45)	last_name character varying (45)	total_spent numeric
1	184	Vivian	Ruiz	80.80
2	87	Wanda	Patterson	137.72
3	477	Dan	Paine	106.79
4	273	Priscilla	Lowe	130.72
5	550	Guy	Brownlee	151.69
6	51	Alice	Stewart	123.70

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

QUERY: - SELECT c.customer_id, c.first_name, c.last_name, SUM(p.amount) AS total_spent

FROM customer c

JOIN payment p ON c.customer_id = p.customer_id

GROUP BY c.customer_id

ORDER BY total_spent DESC

LIMIT 5;

	customer_id [PK] integer	first_name character varying (45)	last_name character varying (45)	total_spent numeric
1	148	Eleanor	Hunt	211.55
2	526	Karl	Seal	208.58
3	178	Marion	Snyder	194.61
4	137	Rhonda	Kennedy	191.62
5	144	Clara	Shaw	189.60

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

QUERY: - SELECT rental_id, rental_date, return_date FROM rental;

	rental_id [PK] integer	rental_date timestamp without time zone	return_date timestamp without time zone
1	2	2005-05-24 22:54:33	2005-05-28 19:40:33
2	3	2005-05-24 23:03:39	2005-06-01 22:12:39
3	4	2005-05-24 23:04:41	2005-06-03 01:43:41
4	5	2005-05-24 23:05:21	2005-06-02 04:33:21
5	6	2005-05-24 23:08:07	2005-05-27 01:32:07
6	7	2005-05-24 23:11:53	2005-05-29 20:34:53
7	8	2005-05-24 23:31:46	2005-05-27 23:33:46

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

QUERY: - SELECT s.first_name, s.last_name, st.store_id

FROM staff s

JOIN store st ON s.staff_id = st.manager_staff_id;

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

Ques.9. Find all customers living in 'California'.

QUERY: - 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';

	first_name character varying (45)	last_name character varying (45)
1	Betty	White
2	Renee	Lane
3	Jacob	Lance
4	Alice	Stewart
5	Rene	Mcalister
6	Rosa	Reynolds
7	Cassandra	Walters
8	Patricia	Johnson
9	Kristin	Johnston

Ques.10. Count how many customers are from each city.

QUERY: - 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;

	city character varying (50)	customer_count bigint
1	Southport	1
2	Taguig	1
3	Tokat	1
4	Atlixco	1
5	Mukateve	1
6	Pontianak	1
7	Gatineau	1
8	Saint-Denis	1

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

QUERY: - SELECT title, length

FROM film

WHERE length = (QUERY: - SELECT MAX(length) FROM film);

	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
6	Muscle Bright	185
7	Pond Seattle	185
8	Soldiers Evolution	185
9	Sweet Brotherhood	185
10	Worst Banger	185

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

QUERY: - 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';

	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

Ques.13. Find the number of rentals made each month.

QUERY: -

SELECT DATE_TRUNC('month', rental_date) AS month, COUNT(*) AS rental_count

FROM rental

GROUP BY month

ORDER BY month;

	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

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

QUERY: - 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';

	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
6	18500	1	2	2363	0.99	2007-02-18 12:02:25.996577
7	18501	1	1	3284	3.99	2007-02-21 04:53:11.996577
8	22680	1	2	10437	4.99	2007-03-01 07:19:30.996577

Ques.15. List all films that have never been rented.

QUERY: - SELECT f.title FROM film f

LEFT 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;

	title character varying (255)
1	Academy Dinosaur
2	Sky Miracle
3	Kill Brotherhood
4	Sister Freddy
5	Gladiator Westward
6	Floats Garden
7	Apollo Teen
8	Crystal Breaking

Ques.16. What is the average rental duration per category?

QUERY: - 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

	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
6	New	4.7460317460317460
7	Sports	4.7162162162162162
8	Children	5.0333333333333333

Ques.17. Which films were rented more than 50 times?

QUERY: - 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;

	title character varying (255)	rental_count bigint

Ques.18. List all employees hired after the year 2005.

QUERY: - SELECT first_name, last_name, last_update FROM staff

WHERE last_update> '2005-12-31';

	first_name character varying (45)	last_name character varying (45)	last_update timestamp without time zone
1	Mike	Hillyer	2006-05-16 16:13:11.79328
2	Jon	Stephens	2006-05-16 16:13:11.79328

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

QUERY: - SELECT s.first_name, s.last_name, COUNT(r.rental_id) AS rentals_processed

FROM staff s

JOIN rental r ON s.staff_id = r.staff_id

GROUP BY s.staff_id;

	first_name character varying (45)	last_name character varying (45)	rentals_processed bigint
1	Mike	Hillyer	8040
2	Jon	Stephens	8004

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

QUERY: - SELECT f.title, COUNT(r.rental_id) 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;

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

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

QUERY: - SELECT f.title, COUNT(r.rental_id) 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;

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

Ques.22. Show all films longer than 2 hours.

QUERY: - SELECT title, length

FROM film

WHERE length > 120;

	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
6	Alley Evolution	180
7	American Circus	129
8	Analyze Hoosiers	181

Ques.23. Find all rentals that were returned late.

QUERY: - SELECT rental_id, rental_date, return_date

FROM rental

WHERE return_date > rental_date + INTERVAL '3 days';

	rental_id [PK] integer	rental_date timestamp without time zone	return_date timestamp without time zone
1	2	2005-05-24 22:54:33	2005-05-28 19:40:33
2	3	2005-05-24 23:03:39	2005-06-01 22:12:39
3	4	2005-05-24 23:04:41	2005-06-03 01:43:41
4	5	2005-05-24 23:05:21	2005-06-02 04:33:21
5	7	2005-05-24 23:11:53	2005-05-29 20:34:53
6	8	2005-05-24 23:31:46	2005-05-27 23:33:46
7	9	2005-05-25 00:00:40	2005-05-28 00:22:40
8	10	2005-05-25 00:02:21	2005-05-31 22:44:21

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

QUERY: - 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;

	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
6	Chris	Brothers	22
7	Alice	Stewart	33
8	Kay	Caldwell	20

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

QUERY: - 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;

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

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

Query: - 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;

```
CREATE VIEW

Query returned successfully in 137 msec.
```

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

QUERY: - UPDATE customer

SET email = 'new_email@example.com'

WHERE customer_id = 1;

```
UPDATE 1

Query returned successfully in 118 msec.
```

Ques.28. Insert a new actor into the actor table.

QUERY: - INSERT INTO actor (first_name, last_name, last_update)

VALUES ('John', 'Doe', CURRENT_TIMESTAMP);

```
INSERT 0 1

Query returned successfully in 129 msec.
```

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

QUERY: - DELETE FROM rental

WHERE return_date IS NULL;

```
DELETE 0

Query returned successfully in 102 msec.
```

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

QUERY: - ALTER TABLE customer

ADD COLUMN age INT;

```
ALTER TABLE

Query returned successfully in 86 msec.
```

Ques.31. Create an index on the 'itle' column of the film table.

QUERY: - CREATE INDEX idx_film_title ON film(title);

```
CREATE INDEX

Query returned successfully in 97 msec.
```

Ques.32. Find the total revenue generated by each store.

QUERY: - 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;

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

Ques.33. What is the city with the highest number of rentals?

QUERY: - 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;

	city character varying (50)	rental_count bigint
1	Aurora	50

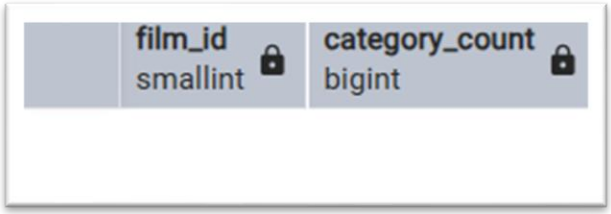
Ques.34. How many films belong to more than one category?

QUERY: - SELECT film_id, COUNT(category_id) AS category_count

FROM film_category

GROUP BY film_id

HAVING COUNT(category_id) > 1;



	film_id smallint	category_count bigint

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

QUERY: - SELECT DISTINCT c.email

QUERY: - 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;

	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

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

QUERY: - SELECT DISTINCT c.email

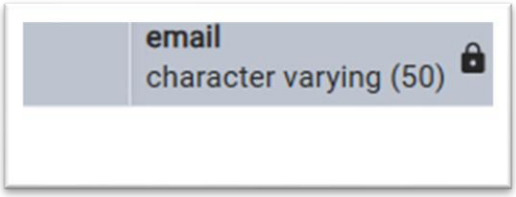
FROM customer c

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';



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

QUERY: - CREATE OR REPLACE FUNCTION get_customer_total_payment(cid INT)

RETURNS NUMERIC AS $$

DECLARE

total NUMERIC;

BEGIN

QUERY: - SELECT SUM(amount) INTO total

FROM payment

WHERE customer_id = cid;

RETURN total;

END;

$$ LANGUAGE plpgsql;

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

QUERY: - 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;

Ques.39. Show the customers who rented films in both ‘Action’ and 'Comedy' categories.

QUERY: - SELECT DISTINCT c.customer_id, c.first_name, c.last_name

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_category fc ON i.film_id = fc.film_id

JOIN category cat ON fc.category_id = cat.category_id

WHERE cat.name IN ('Action', 'Comedy')

GROUP BY c.customer_id, c.first_name, c.last_name

HAVING COUNT(DISTINCT cat.name) = 2;

	customer_id [PK] integer	first_name character varying (45)	last_name character varying (45)
1	1	Mary	Smith
2	3	Linda	Williams
3	4	Barbara	Jones
4	5	Elizabeth	Brown
5	6	Jennifer	Davis
6	7	Maria	Miller
7	10	Dorothy	Taylor
8	12	Nancy	Thomas

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

QUERY: - 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;

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

END