PostgreSQL Assignment Questions - Sure Pro Education

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

   ➔ Query: select first_name, last_name from customer;

   Output:

Data Output Messages Notifications

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

Total rows: 599 Query complete 00:00:01.029

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

   ➔ Query:
```
SELECT f.film_id, f.title
FROM film f
JOIN inventory i ON f.film_id = i.film_id
JOIN rental r ON i.inventory_id = r.inventory_id
WHERE r.return_date IS NULL;
```

   Output:

Data Output Messages Notifications

	film_id [PK] integer	title character varying (255)
1	1	Academy Dinosaur
2	2	Ace Goldfinger
3	4	Affair Prejudice
4	5	African Egg
5	13	Ali Forever

Total rows: 183 Query complete 00:00:00.676

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 LIKE '%Action%';
```

   Output:

Data Output Messages Notifications

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

Total rows: 64 Query complete 00:00:00.254

4. Count the number of films in each category.

➔ Query: SELECT c.name AS category, COUNT(f.film_id) AS film_count
FROM category c
JOIN film_category fc ON c.category_id = fc.category_id
JOIN film f ON fc.film_id = f.film_id
GROUP BY c.name;

Output:

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

Total rows: 16 Query complete 00:00:00.127

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

➔ Query: SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS name, SUM(p.amount) AS total FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY c.customer_id;

Output:

Data Output Messages Notifications

	customer_id [PK] integer	name text	total 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

Total rows: 599 Query complete 00:00:00.209

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

➔ Query: SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS name,
SUM(p.amount) AS total
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY c.customer_id Order by total DESC LIMIT 5 ;

Output:

Data Output Messages Notifications

	customer_id [PK] integer	name text	total 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

Total rows: 5 Query complete 00:00:00.183

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

➔ Query: select rental_id, rental_date, return_date from rental group by rental_id;

Output:

Data Output Messages Notifications

	rental_id [PK] integer	rental_date timestamp without time zone	return_date timestamp without time zone
1	11233	2005-08-02 13:06:11	2005-08-08 15:02:11
2	4790	2005-07-08 16:25:27	2005-07-11 11:35:27
3	3936	2005-07-06 21:15:03	2005-07-08 17:45:03
4	12502	2005-08-18 13:16:31	2005-08-19 17:47:31
5	5468	2005-07-09 23:06:09	2005-07-13 00:48:09

Total rows: 16044 Query complete 00:00:00.245

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

➔ Query: Select st.store_id, CONCAT(st.first_name, ' ', st.last_name) as names from staff st
join store s on st.store_id = s.store_id ;

Output:

Data Output Messages Notifications

	store_id smallint	names text
1	1	Mike Hillyer
2	2	Jon Stephens

9. Find all customers living in 'California'.

➔ Query: SELECT customer_id, CONCAT(first_name, ' ' ,last_name) as names
FROM customer
WHERE address_id IN (
SELECT address_id FROM address WHERE district = 'California');

Output:

Data Output Messages Notifications

	customer_id [PK] integer	names text
1	2	Patricia Johnson
2	14	Betty White
3	51	Alice Stewart
4	112	Rosa Reynolds
5	182	Renee Lane

Total rows: 9 Query complete 00:00:00.375

10. Count how many customers are from each city.

➔ Query: SELECT ci.city, COUNT(c.customer_id) AS total_customers
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;

Output:

Data Output Messages Notifications

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

Total rows: 597 Query complete 00:00:02.363

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

➔ Query: SELECT title, length
FROM film
WHERE length = (SELECT MAX(length) FROM film);

Output:

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

Total rows: 10 Query complete 00:00:00.407

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

➔ Query: select CONCAT(first_name, ' ', last_name) as 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';

Output:

	name text
1	Burt Dukakis
2	Kenneth Paltro...
3	Sidney Crowe
4	Renee Tracy
5	Humphrey Willis

Total rows: 6 Query cor

13. Find the number of rentals made each month.

➔ Query: SELECT EXTRACT(MONTH FROM rental_date) AS month, COUNT(rental_id)
AS total_rentals FROM rental
GROUP BY month;

Output:

Data Output Messages Notifications

	month numeric	total_rentals bigint
1	7	6709
2	8	5686
3	6	2311
4	2	182
5	5	1156

Total rows: 5 Query complete 00:00:00.294

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

➔ Query: select amount, payment_date from payment
join customer c on payment.customer_id = c.customer_id
where CONCAT(c.first_name, ' ', c.last_name) = 'Mary Smith';

Output:

Data Output Messages Notifications

	amount numeric (5,2)	payment_date timestamp without time zone
1	5.99	2007-02-14 23:22:38.996577
2	0.99	2007-02-15 16:31:19.996577
3	9.99	2007-02-15 19:37:12.996577
4	4.99	2007-02-16 13:47:23.996577
5	4.99	2007-02-18 07:10:14.996577

Total rows: 30 Query complete 00:00:00.195

15. List all films that have never been rented.

➔ Query: SELECT f.film_id, 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;

Output:

Data Output Messages Notifications

	film_id [PK] integer	title character varying (255)
1	1	Academy Dinosaur

16. What is the average rental duration per category?

➔ Query: select avg(f.rental_duration) as avg_rental_duration ,c.name as category from film f
join film_category fc on fc.film_id = f.film_id
join category c on fc.category_id = c.category_id
group by c.name;

Output:

Data Output Messages Notifications

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

Total rows: 16 Query complete 00:00:00.214

17. Which films were rented more than 50 times?

➔ Query: SELECT f.film_id, 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.film_id, f.title
HAVING COUNT(r.rental_id) > 50;

Output:

film_id [PK] integer	title character varying (255)	rental_count bigint

18. List all employees hired after the year 2005.

➔ Query: SELECT staff_id, first_name, last_name, last_update
FROM staff
WHERE last_update > '2005-12-31';

Output:

Data Output Messages Notifications

Showing rows: 1

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

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

➔ Query:
```
SELECT s.staff_id, CONCAT(s.first_name, ' ' ,s.last_name) as name , COUNT(r.rental_id)
AS total_rentals FROM staff s
JOIN rental r ON s.staff_id = r.staff_id
GROUP BY s.staff_id, s.first_name, s.last_name;
```

Output:

	staff_id [PK] integer	name text	total_rentals bigint
1	1	Mike Hillyer	8040
2	2	Jon Stephens	8004

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

➔ Query:
```
SELECT c.customer_id, CONCAT(c.first_name, ' ',c.last_name) as name
FROM customer c
left JOIN payment p ON c.customer_id = p.customer_id
WHERE p.payment_id IS NULL;
```

Output:

	customer_id [PK] integer	name text

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

➔ Query:
```
SELECT f.film_id, 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.film_id, f.title
order by rental_count desc limit 1;
```

Output:

	film_id [PK] integer	title character varying (255)	rental_count bigint
1	103	Bucket Brotherhood	34

22. Show all films longer than 2 hours.

➔ Query:
```
SELECT title, length
FROM film
WHERE length > '120';
```

Output:

	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

Total rows: 457 Query complete 00:00:00.256

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';
```

Output:

Data Output Messages Notifications

	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

Total rows: 11472 Query complete 00:00:00.315

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

➔ Query:
```
SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS 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, c.first_name, c.last_name;
```

Output:

Data Output Messages Notifications

	customer_id [PK] integer	name text	total_rentals bigint
1	87	Wanda Patterson	30
2	184	Vivian Ruiz	23
3	477	Dan Paine	22
4	273	Priscilla Lowe	35
5	550	Guy Brownlee	32

Total rows: 599 Query complete 00:00:00.426

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

➔ Query:
```
SELECT c.name AS category, COUNT(r.rental_id) AS
total_rentals FROM rental r
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film f ON i.film_id = f.film_id
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
ORDER BY total_rentals DESC
LIMIT 3;
```

Output:

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

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

➔ Query:
```
CREATE VIEW Customer_View AS
SELECT
CONCAT(c.first_name, ' ', c.last_name) AS name,
  SUM(p.amount) AS total_payment
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY c.customer_id, name;

select * from Customer_View;
```

Output:

	name text	total_payment numeric
1	Jeffrey Spear	65.83
2	Jesse Schilling	79.78
3	Jacqueline Long	146.68
4	Warren Sherrod	152.69
5	Scott Shelley	60.82

Total rows: 599 Query complete 00:00:00.254

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

➔ Query:
```
UPDATE customer
SET email = 'new_email@example.com'
WHERE customer_id = 1;
```

Output:
UPDATE 1 Query returned successfully in 162 msec.

598	599	2	Austin	Cintron	austin.cintron@sakilacustomer.org
599	1	1	Mary	Smith	new_email@example.com

28. Insert a new actor into the actor table.

➔ Query: Insert into actor values('201', 'John' , 'Thomas' , now()::timestamp(2));
Output:

Showing rows: 1

	actor_id [PK] integer	first_name character varying (45)	last_name character varying (45)	last_update timestamp without time zone
197	197	Reese	West	2013-05-26 14:47:57.62
198	198	Mary	Keitel	2013-05-26 14:47:57.62
199	199	Julia	Fawcett	2013-05-26 14:47:57.62
200	200	Thora	Temple	2013-05-26 14:47:57.62
201	201	John	Thomas	2025-07-12 16:09:34.858707

Total rows: 201 Query complete 00:00:00.201

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

➔ Query:
```
DELETE FROM rental
WHERE return_date IS NULL
AND rental_id NOT IN (SELECT rental_id FROM payment);
```

Output: DELETE 0 Query returned successfully in 171 msec.

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

➔ Query: alter table customer
add age int default 20 not null;
Output:

1 of 1

hout time zone	active integer	age integer
:49:45.738	1	20
:49:45.738	1	20
:49:45.738	1	20
:49:45.738	1	20
:49:45.738	1	20
:49:45.738	1	20

CRLF Ln 2, Col 24

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

➔ Query: create index film_title on film(title);
Output: relation "film_title" already exists

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;
```

Output:

	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:
```
SELECT ci.city, COUNT(r.rental_id) 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;
```

Output:

	city character varying (50)	rental_count bigint
1	Aurora	50

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

➔ Query:
```
SELECT COUNT(*) AS multi_category_films
FROM (
  SELECT film_id
  FROM film_category
  GROUP BY film_id
  HAVING COUNT(category_id) > 1 ) ;
```

Output:

	multi_category_films bigint
1	0

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

➔ Query:
```
SELECT
CONCAT(a.first_name, ' ', a.last_name) AS name,
  COUNT(f.film_id) AS no_of_films
FROM actor a
JOIN film_actor fa ON a.actor_id = fa.actor_id
JOIN film f ON f.film_id = fa.film_id
GROUP BY a.actor_id, a.first_name, a.last_name
ORDER BY no_of_films DESC
LIMIT 10;
```

Output:

	name text	no_of_films 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

Total rows: 10 Query complete 00:00:00.129

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';
```

Output:

	email character varying (50)

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

➔ Query:
```
CREATE FUNCTION get_customer_payment_total(cust_id INT)
RETURNS NUMERIC AS $$
BEGIN
  RETURN (
    SELECT SUM(amount)
    FROM payment
    WHERE customer_id = cust_id
  );
END;
$$ LANGUAGE plpgsql;

SELECT get_customer_payment_total(1);
```

Output:

	get_customer_payment_total numeric
1	114.70

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;

rollback;
```

Output: COMMIT

Query returned successfully in 81 msec.

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

➔ Query:
```
SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS 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;
```

Output:

	customer_id [PK] integer	name text
1	1	Mary Smith
2	3	Linda Williams
3	4	Barbara Jones
4	5	Elizabeth Brown
5	6	Jennifer Davis
6	7	Maria Miller

Total rows: 419 Query complete 00:00:00.141

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

➔ Query:
```
SELECT a.actor_id, CONCAT(a.first_name, ' ', a.last_name) AS name
FROM actor a
LEFT JOIN film_actor fa ON a.actor_id = fa.actor_id
WHERE fa.film_id IS NULL;
```

Output:

Data Output Messages Notificat

	actor_id [PK] integer	name text
1	201	John Thomas

-- END OF ASSIGNMENT --