

Query

```
1 --PostgreSQL Practice Questions:  
2  
3 --Q1. List the first name and last name of all customers.  
4 SELECT first_name, last_name FROM customer;
```

Data Output Messages Notifications



Showing rows: 1 to 599 Page No: 1 of 1

	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
6	Jennifer	Davis
7	Maria	Miller
8	Susan	Wilson
9	Margaret	Moore
10	Dorothy	Taylor
11	Lisa	Anderson
12	Nancy	Thomas
13	Karen	Jackson
14	Betty	White
15	Helen	Harris
16	Sandra	Martin
17	Donna	Thompson
18	Carol	Garcia
19	Ruth	Martinez

Total rows: 599

Query complete 00:00:00.132

CRLF Ln 4, Col 1

Query History

```
6 --Q2. Find all the movies that are currently rented out.  
7 SELECT F.film_id, F.title FROM film F JOIN inventory I ON F.film_id = I.film_id  
8 JOIN rental R ON R.inventory_id = I.inventory_id  
9 WHERE R.return_date IS NULL;
```

Data Output Messages Notifications



Showing rows: 1 to 188 | Page No: 1 of 1 |

	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
6	17	Alone Trip
7	19	Amadeus Holy
8	21	American Circus
9	22	Amistad Midsummer
10	39	Armageddon Lost
11	50	Baked Cleopatra
12	53	Bang Kwai
13	57	Basic Easy
14	67	Berets Agent
15	79	Blade Polish
16	80	Blanket Beverly
17	86	Boogie Amelie
18	90	Boulevard Mob
19	91	Bound Cheaper

Total rows: 188 | Query complete 00:00:00.121

CRLF | Ln 7, Col 1

Query History

```
11 --Q3. Show the titles of all movies in the 'Action' category.  
12 SELECT F.film_id, title FROM film F JOIN film_category FC ON F.film_id = FC.film_id  
13 JOIN category C ON FC.category_id = C.category_id  
14 WHERE C.name = 'Action';
```

Data Output Messages Notifications

Showing rows: 1 to 64 Page No: 1 of 1

	film_id	title
	[PK] integer	character varying (255)
1	19	Amadeus Holy
2	21	American Circus
3	29	Antitrust Tomatoes
4	38	Ark Ridgemont
5	56	Barefoot Manchurian
6	67	Berets Agent
7	97	Bride Intrigue
8	105	Bull Shawshank
9	111	Caddyshack Jedi
10	115	Campus Remember
11	126	Casualties Encino
12	130	Celebrity Horn
13	162	Clueless Bucket
14	194	Crow Grease
15	205	Dances None
16	210	Darko Dorado
17	212	Darn Forrester
18	229	Devil Desire
19	250	Dragon Squad

Total rows: 64 Query complete 00:00:00.124

CRLF Ln 12, Col 1

Query History

```
16 --Q4. Count the number of films in each category.  
17 v SELECT name AS Category, COUNT(*) AS film_count  
18 FROM category JOIN film_category ON category.category_id = film_category.category_id  
19 GROUP BY Category ORDER BY film_count;  
20
```

Data Output Messages Notifications



Showing rows: 1 to 16 Page No: 1 of 1

	category character varying (25)	film_count bigint
1	Music	51
2	Horror	56
3	Travel	57
4	Classics	57
5	Comedy	58
6	Children	60
7	Sci-Fi	61
8	Games	61
9	Drama	62
10	New	63
11	Action	64
12	Animation	66
13	Documentary	68
14	Family	69
15	Foreign	73
16	Sports	74

Query History

```
21 --Q5. What is the total amount spent by each customer?
22 v SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name,
23   SUM(P.amount) AS Total_amount
24   FROM customer C JOIN payment P ON C.customer_id = P.customer_id
25   GROUP BY C.customer_id, Customer_name ORDER BY Total_amount;
```

Data Output Messages Notifications



Showing rows: 1 to 16 Page No: 1 of 1

	category character varying (25)	film_count bigint
1	Music	51
2	Horror	56
3	Travel	57
4	Classics	57
5	Comedy	58
6	Children	60
7	Sci-Fi	61
8	Games	61
9	Drama	62
10	New	63
11	Action	64
12	Animation	66
13	Documentary	68
14	Family	69
15	Foreign	73
16	Sports	74

Total rows: 16

Query complete 00:00:00.120

CRLF Ln 22, Col 1

Query

```
27 --Q6. Find the top 5 customers who spent the most.  
28 SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name,  
29 SUM(P.amount) AS Total_amount  
30 FROM customer C JOIN payment P ON C.customer_id = P.customer_id  
31 GROUP BY C.customer_id, Customer_name ORDER BY Total_amount DESC LIMIT(5);
```

Data Output Messages Notifications

Showing rows: 1 to 5 Page No: 1 of 1

	customer_id [PK] integer	customer_name text	total_amount 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

Successfully run. Total query runtime: 132 msec. 5 rows affected.

Total rows: 5

Query complete 00:00:00.132

CRLF Ln 28, Col 1

Query

```
33 --Q7. Display the rental date and return date for each rental.
34 SELECT rental_id, rental_date, return_date FROM rental;
```

Data Output



Showing rows: 1 to 1000 Page No: 1 of 17

	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
8	9	2005-05-25 00:00:40	2005-05-28 00:22:40
9	10	2005-05-25 00:02:21	2005-05-31 22:44:21
10	11	2005-05-25 00:09:02	2005-06-02 20:56:02
11	12	2005-05-25 00:19:27	2005-05-30 05:44:27
12	13	2005-05-25 00:22:55	2005-05-30 04:28:55
13	14	2005-05-25 00:31:15	2005-05-26 02:56:15
14	15	2005-05-25 00:39:22	2005-06-03 03:30:22
15	16	2005-05-25 00:43:11	2005-05-26 04:42:11
16	17	2005-05-25 01:06:36	2005-05-27 00:43:36
17	18	2005-05-25 01:10:47	2005-05-31 06:35:47
18	19	2005-05-25 01:17:24	2005-05-31 06:00:24
19	20	2005-05-25 01:48:41	2005-05-27 02:20:41
20	21	2005-05-25 01:59:46	2005-05-26 01:01:46

Total rows: 16049

Query complete 00:00:00.152

✓ Successfully run. Total query runtime: 152 msec. 16049 rows affected.

CRLF Ln 34, Col 1

Query

```
--Q8. List the names of staff members and the stores they manage.  
36  
37 SELECT S.staff_id, CONCAT(S.first_name, ' ', S.last_name) AS Staff_name, ST.store_id  
38 FROM staff S JOIN store ST  
39 ON S.staff_id = ST.manager_staff_id;
```

Data Output



Showing rows: 1 to 2 Page No: 1 of 1

	staff_id integer	staff_name text	store_id integer
1	1	Mike Hillyer	1
2	2	Jon Stephens	2

✓ Successfully run. Total query runtime: 100 msec. 2 rows affected.

Total rows: 2

Query complete 00:00:00.100

CRLF Ln 37, Col 1

Query

--Q9. Find all customers living in 'California'.

```
41 --Q9. Find all customers living in 'California'.
42 v SELECT C.customer_id, CONCAT(C.first_name, ' ', last_name) AS Customer_name, A.district AS Living_in, A.city_id
43 FROM customer C JOIN address A ON C.address_id = A.address_id
44 WHERE A.district = 'California';
```

Data Output



Showing rows: 1 to 9 Page No: 1 of 1

	customer_id integer	customer_name text	living_in character varying (20)	city_id smallint
1	2	Patricia Johnson	California	449
2	14	Betty White	California	120
3	51	Alice Stewart	California	171
4	112	Rosa Reynolds	California	292
5	182	Renee Lane	California	126
6	214	Kristin Johnston	California	508
7	269	Cassandra Walters	California	446
8	420	Jacob Lance	California	155
9	593	Rene Mcalister	California	177

Successfully run. Total query runtime: 85 msec. 9 rows affected.

Total rows: 9

Query complete 00:00:00.085

CRLF Ln 42, Col 1

Query

```
46 --Q10. Count how many customers are from each city.  
47 SELECT CT.city_id, CT.city, COUNT(*) AS Total_customer  
48 FROM city CT JOIN address A  
49 ON CT.city_id = A.city_id  
50 JOIN customer C ON A.address_id = C.address_id  
51 GROUP BY CT.city_id  
52 ORDER BY Total_customer;
```

Data Output Messages Notifications



Showing rows: 1 to 597



Page No:

1

of 1



	city_id [PK] integer	city character varying (50)	total_customer bigint
1	148	Duisburg	1
2	87	Botshabelo	1
3	477	Siegen	1
4	273	Klerksdorp	1
5	550	Udaipur	1
6	51	Balurghat	1
7	394	Pak Kret	1
8	272	Kitwe	1
9	70	Bergamo	1
10	190	guas Lindas de Gois	1
11	350	Mysore	1
12	539	Tonghae	1
13	554	Uluberia	1
14	278	Korolev	1
15	424	Quilmes	1
16	406	Phnom Penh	1

✓ Successfully run. Total query runtime: 127 msec. 597 rows affected. ✘

Total rows: 597 Query complete 00:00:00.127

CRLF Ln 47, Col 1

Query

```
--Q11. Find the film(s) with the longest duration.  
55 ✓ SELECT film_id, title, length FROM film  
56 WHERE length = (SELECT MAX(length) FROM film);
```

Data Output



Showing rows: 1 to 10 Page No: 1 of 1

	film_id [PK] integer	title character varying (255)	length smallint
1	141	Chicago North	185
2	182	Control Anthem	185
3	212	Darn Forrester	185
4	349	Gangs Pride	185
5	426	Home Pity	185
6	609	Muscle Bright	185
7	690	Pond Seattle	185
8	817	Soldiers Evolution	185
9	872	Sweet Brotherhood	185
10	991	Worst Banger	185

✓ Successfully run. Total query runtime: 124 msec. 10 rows affected.

Total rows: 10

Query complete 00:00:00.124

CRLF Ln 55, Col 1

Query

```
--Q12. Which actors appear in the film titled 'Alien Center'?
59 v SELECT A.actor_id, CONCAT(A.first_name, ' ', A.last_name) AS Actor_name, F.title
60 FROM actor A JOIN film_actor FA ON A.actor_id = FA.actor_id
61 JOIN film F ON FA.film_id = F.film_id
62 WHERE F.title = 'Alien Center';
```

Data Output



Showing rows: 1 to 6 Page No: 1 of 1

	actor_id	actor_name	title
1	36	Burt Dukakis	Alien Center
2	69	Kenneth Paltro...	Alien Center
3	105	Sidney Crowe	Alien Center
4	117	Renee Tracy	Alien Center
5	164	Humphrey Willis	Alien Center
6	170	Mena Hopper	Alien Center

Successfully run. Total query runtime: 95 msec. 6 rows affected.

Total rows: 6

Query complete 00:00:00.095

CRLF Ln 59, Col 1

Query

```
--Q13. Find the number of rentals made each month.  
65 SELECT TO_CHAR(rental_date,'MM') AS month, COUNT(*) AS Total_rentals  
66 FROM rental  
67 GROUP BY month  
68 ORDER BY month;
```

Data Output Messages Notifications



Showing rows: 1 to 5 Page No: 1 of 1

	month text	total_rentals bigint
1	02	182
2	05	1156
3	06	2311
4	07	6714
5	08	5686

✓ Successfully run. Total query runtime: 94 msec. 5 rows affected.

Total rows: 5 Query complete 00:00:00.094

CRLF Ln 65, Col 1

Query Query History

```
70 --Q14. Show all payments made by customer 'Mary Smith'.
71 SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name, P.payment_id, P.amount AS Payments
72 FROM payment P JOIN customer C
73 ON P.customer_id = C.customer_id
74 WHERE C.first_name = 'Mary' AND C.last_name = 'Smith';
```

Data Output Messages Notifications

Showing rows: 1 to 30 Page No: 1 of 1

	customer_id	customer_name	payment_id	payments numeric (5,2)
1	1	Mary Smith	18495	5.99
2	1	Mary Smith	18496	0.99
3	1	Mary Smith	18497	9.99
4	1	Mary Smith	18498	4.99
5	1	Mary Smith	18499	4.99
6	1	Mary Smith	18500	0.99
7	1	Mary Smith	18501	3.99
8	1	Mary Smith	22680	4.99
9	1	Mary Smith	22681	3.99
10	1	Mary Smith	22682	0.99
11	1	Mary Smith	22683	4.99
12	1	Mary Smith	22684	0.99
13	1	Mary Smith	22685	0.99
14	1	Mary Smith	22686	2.99
15	1	Mary Smith	22687	0.99
16	1	Mary Smith	22688	1.99
17	1	Mary Smith	22689	2.99
18	1	Mary Smith	22690	5.99

Total rows: 30

Query complete 00:00:00.089

Successfully run. Total query runtime: 89 msec. 30 rows affected.

CRLF Ln 71, Col 1

Query History

```
76 --Q15. List all films that have never been rented.  
77 v SELECT F.film_id, F.title, F.description, R.rental_id FROM film F JOIN inventory I  
78 ON F.film_id = I.film_id  
79 LEFT JOIN rental R ON I.inventory_id = R.inventory_id  
80 WHERE R.rental_id IS NULL;
```

Data Output Messages Notifications



Showing rows: 1 to 1 Page No: 1 of 1

	film_id	title	description	rental_id
	integer	character varying (255)	text	integer
1	1	Academy Dinosaur	A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies	[null]

✓ Successfully run. Total query runtime: 90 msec. 1 rows affected

Total rows: 1

Query complete 00:00:00.090

CRLF Ln 77, Col 1

Query History

```

82 --Q16. What is the average rental duration per category?
83 SELECT C.category_id, C.name AS Category, AVG(F.rental_duration) AS Avg_rental_duration
84 FROM category C JOIN film_category FC
85 ON C.category_id = FC.category_id
86 JOIN film F ON FC.film_id = F.film_id
87 GROUP BY Category, C.category_id
88 ORDER BY C.category_id;

```

Data Output Messages Notifications



Showing rows: 1 to 16 | Page No: 1 of 1 |

	category_id [PK] integer	category character varying (25)	avg_rental_duration numeric
1	1	Action	4.953125000000000
2	2	Animation	4.8939393939393939
3	3	Children	5.033333333333333
4	4	Classics	5.0701754385964912
5	5	Comedy	4.9310344827586207
6	6	Documentary	4.7647058823529412
7	7	Drama	5.0806451612903226
8	8	Family	5.1739130434782609
9	9	Foreign	5.1095890410958904
10	10	Games	5.0655737704918033
11	11	Horror	4.8571428571428571
12	12	Music	5.2352941176470588
13	13	New	4.7460317460317460
14	14	Sci-Fi	4.8852459016393443
15	15	Sports	4.7162162162162162
16	16	Travel	5.3508771929824561

✓ Successfully run. Total query runtime: 124 msec. 16 rows

Total rows: 16 | Query complete 00:00:00.124

CRLF | Ln 83, Col 1

Query Query History

```
90 --Q17. Which films were rented more than 50 times?  
91 SELECT F.title, COUNT(*) AS Total_rented  
92 FROM film F JOIN inventory I  
93 ON F.film_id = I.film_id  
94 JOIN rental R ON I.inventory_id = R.inventory_id  
95 GROUP BY F.title  
96 HAVING COUNT(*) > 50;  
--
```

Data Output Messages Notifications



	title	total_rented
	character varying (255)	bigint

Total rows: 0 | Query complete 00:00:00.112

CRLF | Ln 91, Col 1

Query

```
98 --Q18. List all employees hired after the year 2005.  
99 SELECT staff_id, CONCAT(first_name, ' ', last_name) AS Employee_name,  
100 TO_CHAR(last_update, 'YYYY') AS hired_year  
101 FROM staff WHERE last_update > '2005-12-31';  
102 --As the 'hired year' field is not available in the database, we utilized the last_update column from the staff table and  
103 --labeled it as 'hired Date' in the output.
```

Data Output



Showing rows: 1 to 2 Page No: 1 of 1

	staff_id [PK] integer	employee_name text	hired_year text
1	1	Mike Hillyer	2006
2	2	Jon Stephens	2006

✓ Successfully run. Total query runtime: 188 msec. 2 rows affected.

Total rows: 2

Query complete 00:00:00.188

CRLF Ln 99, Col 1

Query

```
--Q19. Show the number of rentals processed by each staff member.  
105  
106 SELECT S.staff_id, CONCAT(S.first_name, ' ', S.last_name) AS Staff_name, COUNT(R.rental_id) AS Total_rentals  
107 FROM staff S JOIN rental R  
108 ON S.staff_id = R.staff_id  
109 GROUP BY S.staff_id;
```

Data Output Messages Notifications



	staff_id [PK] integer	staff_name text	total_rentals bigint
1	1	Mike Hillyer	8045
2	2	Jon Stephens	8004

Showing rows: 1 to 2



Page No:

1

of 1



Total rows: 2 Query complete 00:00:00.105

✓ Successfully run. Total query runtime: 105 msec. 2 rows affected. ✘

CRLF Ln 106, Col 1

Query Query History

```
111 --Q20. Display all customers who have not made any payments.  
112 SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name  
113 FROM customer C JOIN payment P  
114 ON C.customer_id = P.payment_id  
115 WHERE P.payment_id IS NULL;  
116
```

Data Output Messages Notifications



	customer_id	customer_name
	[PK] integer	text

Total rows: 0

Query complete 00:00:00.105

✓ Successfully run. Total query runtime: 105 msec. 0 rows affected. ✘

CRLF Ln 111, Col 1

Query

Query History

```
117 --Q21. What is the most popular film (rented the most)?
118 SELECT F.film_id, F.title, COUNT(R.rental_id) AS Rental_count
119 FROM film F JOIN inventory I ON F.film_id = I.film_id
120 JOIN rental R ON I.inventory_id = R.inventory_id
121 GROUP BY F.film_id
122 ORDER BY Rental_count DESC
123 LIMIT(1);
```

Data Output



Showing rows: 1 to 1 Page No: 1 of 1

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

Total rows: 1

Query complete 00:00:00.109

✓ Successfully run. Total query runtime: 109 msec. 1 rows affected.

CRLF Ln 118, Col 1

Query

```
125 --Q22. Show all films longer than 2 hours.  
126 ✓ SELECT film_id, title, length FROM film  
127 WHERE length > 120  
128 ORDER BY length;
```

Data Output Messages Notifications



Showing rows: 1 to 457 Page No: 1 of 1

	film_id [PK] integer	title character varying (255)	length smallint
1	207	Dangerous Uptown	121
2	86	Boogie Amelie	121
3	403	Harry Idaho	121
4	93	Brannigan Sunrise	121
5	704	Pure Runner	121
6	37	Arizona Bang	121
7	658	Paris Weekend	121
8	490	Jumanji Blade	121
9	68	Betrayed Rear	122
10	218	Deceiver Betrayed	122
11	239	Dogma Family	122
12	452	Illusion Amelie	122
13	175	Confused Candles	122
14	297	Extraordinary Conquerer	122
15	142	Chicken Hellfighters	122
16	576	Mighty Luck	122
17	934	Vanilla Day	122
18	58	Beach Heartbreakers	122
19	436	Hours Range	122

✓ Successfully run. Total query runtime: 108 msec. 457 rows affected.

Total rows: 457

Query complete 00:00:00.108

CRLF Ln 126, Col 1

Query

```

130 --Q23. Find all rentals that were returned late.
131 SELECT R.rental_id, R.rental_date, R.return_date, F.rental_duration*INTERVAL '1 DAY' AS Duration
132 FROM rental R JOIN inventory I
133 ON R.inventory_id = I.inventory_id
134 JOIN film F ON I.film_id = F.film_id
135 WHERE R.return_date > R.rental_date + F.rental_duration*INTERVAL '1 DAY';

```

Data Output Messages Notifications

	rental_id [PK] integer	rental_date timestamp without time zone	return_date timestamp without time zone	duration interval
1	3	2005-05-24 23:03:39	2005-06-01 22:12:39	7 days
2	4	2005-05-24 23:04:41	2005-06-03 01:43:41	6 days
3	5	2005-05-24 23:05:21	2005-06-02 04:33:21	5 days
4	7	2005-05-24 23:11:53	2005-05-29 20:34:53	4 days
5	10	2005-05-25 00:02:21	2005-05-31 22:44:21	5 days
6	11	2005-05-25 00:09:02	2005-06-02 20:56:02	4 days
7	13	2005-05-25 00:22:55	2005-05-30 04:28:55	3 days
8	15	2005-05-25 00:39:22	2005-06-03 03:30:22	4 days
9	23	2005-05-25 02:40:21	2005-05-29 06:34:21	4 days
10	26	2005-05-25 03:36:50	2005-05-31 00:34:50	3 days
11	30	2005-05-25 04:01:32	2005-05-30 03:12:32	3 days
12	38	2005-05-25 04:47:44	2005-06-01 00:58:44	5 days
13	41	2005-05-25 05:12:29	2005-06-02 00:28:29	3 days
14	44	2005-05-25 05:53:23	2005-05-29 10:56:23	4 days
15	45	2005-05-25 05:59:39	2005-06-02 09:56:39	5 days
16	46	2005-05-25 06:04:08	2005-06-02 08:18:08	3 days
17	47	2005-05-25 06:05:20	2005-05-30 03:04:20	4 days

Showing rows: 1 to 1000

Page No: 1 of 9

✓ Successfully run. Total query runtime: 140 msec. 8121 rows affected. ✘

Total rows: 8121

Query complete 00:00:00.140

CRLF Ln 131, Col 1

Query

Query History



```
137 --Q24. List customers and the number of films they rented.  
138 SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name, COUNT(R.rental_id) AS Film_rented  
139 FROM customer C JOIN rental R  
140 ON C.customer_id = R.customer_id  
141 GROUP BY C.customer_id  
142 ORDER BY Film_rented;
```

Data Output Messages Notifications



Showing rows: 1 to 599

Page No:

1

of 1



	customer_id [PK] integer	customer_name text	film_rented bigint
1	318	Brian Wyman	12
2	281	Leona Obrien	14
3	61	Katherine Rivera	14
4	110	Tiffany Jordan	14
5	136	Anita Morales	15
6	248	Caroline Bowman	15
7	492	Lester Kraus	16
8	164	Joann Gardner	16
9	464	Jerome Kenyon	16
10	398	Antonio Meek	16
11	465	Floyd Gandy	17
12	555	Dwight Lombardi	17
13	48	Ann Evans	17
14	223	Melinda Fernandez	17
15	315	Kenneth Gooden	17
16	140	Eva Ramos	18
17	205	Eileen Carr	18

Successfully run. Total query runtime: 132 msec. 599 rows affected.

Total rows: 599

Query complete 00:00:00.132

CRLF Ln 138, Col 1

Query Query History

```
144 --Q25. Write a query to show top 3 rented film categories.  
145 SELECT C.category_id, C.name AS Film_category, COUNT(R.rental_id) AS Total_rented  
146 FROM rental R JOIN inventory I  
147 ON R.inventory_id = I.inventory_id  
148 JOIN film_category FC  
149 ON I.film_id = FC.film_id  
150 JOIN category C  
151 ON FC.category_id = C.category_id  
152 GROUP BY C.category_id  
153 ORDER BY Total_rented DESC  
154 LIMIT(3);
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No: 1 of 1

	category_id [PK] integer	film_category character varying (25)	total_rented bigint
1	15	Sports	1179
2	2	Animation	1166
3	1	Action	1112

Successfully run. Total query runtime: 111 msec. 3 rows affected.

Total rows: 3

Query complete 00:00:00.111

CRLF Ln 145, Col 1

Query

```
--Q26. Create a view that shows all customer names and their payment totals.  
156  
157 CREATE VIEW customer_payment_totals AS  
158 SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name, SUM(P.amount) AS Total_payment  
159 FROM customer C JOIN payment P  
160 ON C.customer_id = P.customer_id  
161 GROUP BY C.customer_id  
162 ORDER BY Total_payment;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 66 msec.

✓ Query returned successfully in 66 msec. ✘

Total rows: Query complete 00:00:00.066

CRLF Ln 157, Col 1

Query Query History

```
164 --Q27. Update a customer's email address given their ID.  
165 UPDATE customer  
166 SET email = 'newmail@gmail.com'  
167 WHERE customer_id = '1';
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 63 msec.

✓ Query returned successfully in 63 msec. X

Total rows: 1 Query complete 00:00:00.062

CREATE In 165 Col 1

Query Query History



```
169 --Q28. Insert a new actor into the actor table.  
170 ✓ INSERT INTO actor (first_name, last_name, last_update)  
171   VALUES ('Tony', 'Stark', CURRENT_TIMESTAMP);  
172
```



Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 69 msec.

✓ Query returned successfully in 69 msec. ✘

Total rows: Query complete 00:00:00.069

CRLF Ln 170, Col 1

Query Query History ↗

```
173 --Q29. Delete all records from the rentals table where return_date is NULL.  
174 ✓ DELETE FROM rental  
175 WHERE rental_date IS NULL;
```

Data Output Messages Notifications ↗

DELETE 0

Query returned successfully in 78 msec.

Total rows: Query complete 00:00:00.078

✓ Query returned successfully in 78 msec. ✘

CRLF Ln 174, Col 1

Query Query History

177 --Q30. Add a new column 'age' to the customer table.

178 ▾ **ALTER TABLE customer**

179 **ADD COLUMN age INTEGER;**

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 59 msec.

✓ Query returned successfully in 59 msec. X

Total rows: Query complete 00:00:00.059

CRLF Ln 178, Col 1

Query Query History 

181 --Q31. Create an index on the 'title' column of the film table.
182 ✓ **CREATE INDEX** index_title
183 **ON** film(title);

Data Output **Messages** Notifications 

CREATE INDEX

Query returned successfully in 69 msec.

✓ Query returned successfully in 69 msec. 

Total rows: 0 Query complete 00:00:00 069

CRI F In 182 Col 1

Query Query History

```
187 --Q32. Find the total revenue generated by each store.  
188 SELECT ST.store_id, SUM(P.amount) AS Total_revenue  
189 FROM store ST JOIN staff SF  
190 ON ST.store_id = SF.store_id  
191 JOIN payment P  
192 ON SF.staff_id = P.staff_id  
193 GROUP BY ST.store_id;
```

Data Output Messages Notifications

Showing rows: 1 to 2 Page No: 1 of 1

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

✓ Successfully run. Total query runtime: 99 msec. 2 rows affected.

Total rows: 2 Query complete 00:00:00.099

CRI F In 188 Col 1

Query

```
--Q33. What is the city with the highest number of rentals?  
196 SELECT CI.city_id, CI.city,COUNT(R.rental_id) AS Highest_rental  
197 FROM city CI JOIN address A  
198 ON CI.city_id = A.city_id  
199 JOIN customer C  
200 ON A.address_id = C.address_id  
201 JOIN rental R  
202 ON C.customer_id = R.customer_id  
203 GROUP BY CI.city_id  
204 ORDER BY Highest_rental DESC  
205 LIMIT(1);
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	city_id [PK] integer	city character varying (50)	highest_rental bigint
1	42	Aurora	50

Successfully run. Total query runtime: 170 msec. 1 rows affected.

Total rows: 1

Query complete 00:00:00.170

CRLF Ln 196, Col 1

Query Query History

```
207 --Q34. How many films belong to more than one category?  
208 SELECT COUNT(*) AS multi_category_film FROM (  
209     SELECT film_id FROM film_category  
210     GROUP BY film_id  
211     HAVING COUNT(category_id) > 1);  
212
```

Data Output Messages Notifications



Showing rows: 1 to 1 Page No: 1 of 1

	multi_category_film	bigint
1		0

Query

```
213 --Q35. List the top 10 actors by number of films they appeared in.  
214 SELECT A.actor_id, CONCAT(A.first_name, ' ', A.last_name) AS Actor_name, COUNT(FA.film_id) AS Total_films  
215 FROM actor A JOIN film_actor FA  
216 ON A.actor_id = FA.actor_id  
217 GROUP BY A.actor_id  
218 ORDER BY Total_films DESC  
219 LIMIT(10);
```

Data Output Messages Notifications



Showing rows: 1 to 10 Page No: 1 of 1

	actor_id [PK] integer	actor_name text	total_films bigint
1	107	Gina Degeneres	42
2	102	Walter Torn	41
3	198	Mary Keitel	40
4	181	Matthew Carrey	39
5	23	Sandra Kilmer	37
6	81	Scarlett Damon	36
7	144	Angela Witherspoon	35
8	158	Vivien Basinger	35
9	37	Val Bolger	35
10	60	Henry Berry	35

✓ Successfully run. Total query runtime: 127 msec. 10 rows affected.

Total rows: 10

Query complete 00:00:00.127

CRLF Ln 214, Col 1

Query Query History

```
221 --Q36. Retrieve the email addresses of customers who rented 'Matrix Revolutions'.
222 SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name, C.email
223 FROM customer C JOIN rental R
224 ON C.customer_id = R.customer_id
225 JOIN inventory I
226 ON R.inventory_id = I.inventory_id
227 JOIN film F
228 ON I.film_id = F.film_id
229 WHERE F.title = 'Matrix Revolutions';
```

Data Output Messages Notifications



	customer_id [PK] integer	customer_name	email
			character varying (50)

✓ Successfully run. Total query runtime: 125 msec. 0 rows affected. X

Total rows: 0 Query complete 00:00:00.125

CRLF Ln 222, Col 1

Query Query History



```
231 --Q37. Create a stored function to return customer payment total given their ID.  
232 ✓ CREATE OR REPLACE FUNCTION get_customer_total_payment(Cid INTEGER)  
233 RETURNS NUMERIC AS $$  
234 BEGIN  
235 RETURN (  
236 SELECT SUM(amount) FROM payment WHERE customer_id = Cid);  
237 END;  
238 $$ LANGUAGE plpgsql;
```

Data Output Messages Notifications



CREATE FUNCTION

Query returned successfully in 60 msec.

Total rows: Query complete 00:00:00.060

✓ Query returned successfully in 60 msec. X

CRLF Ln 232, Col 1

Query Query History

```
240 --Q38. Begin a transaction that updates stock and inserts a rental record.  
241 BEGIN;  
242 UPDATE inventory  
243 SET last_update = CURRENT_TIMESTAMP  
244 WHERE inventory_id = 1;  
245 INSERT INTO rental (rental_date, inventory_id, customer_id, staff_id, return_date)  
246 VALUES (CURRENT_TIMESTAMP, 1, 1, 1, NULL);  
247 COMMIT;
```

Data Output Messages Notifications

COMMIT

Query returned successfully in 65 msec.

✓ Query returned successfully in 65 msec.

Query Query History

```
249 --Q39. Show the customers who rented films in both 'Action' and 'Comedy' categories.
250 SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name
251 FROM customer C
252 WHERE EXISTS (
253     SELECT 1
254     FROM rental R JOIN inventory I
255     ON R.inventory_id = I.inventory_id
256     JOIN film_category FC
257     ON I.film_id = FC.film_id
258     JOIN category CA
259     ON FC.category_id = CA.category_id
260     WHERE CA.name = 'Action' AND C.customer_id = R.customer_id)
261 AND EXISTS (
262     SELECT 1
263     FROM rental R JOIN inventory I
264     ON R.inventory_id = I.inventory_id
265     JOIN film_category FC
266     ON I.film_id = FC.film_id
267     JOIN category CA
268     ON FC.category_id = CA.category_id
269     WHERE CA.name = 'Comedy' AND C.customer_id = R.customer_id)
270 ORDER BY C.customer_id;
```

Data Output Messages Notifications

Showing rows: 1 to 419 Page No: 1 of 1

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

Total rows: 419

Query complete 00:00:00.126

CRLF Ln 250, Col 1

Query Query History

```
272 --Q40. Find actors who have never acted in a film.  
273 SELECT A.actor_id, CONCAT(A.first_name, ' ', A.last_name) AS Actor_name  
274 FROM actor A LEFT JOIN film_actor FA  
275 ON A.actor_id = FA.actor_id  
276 WHERE FA.film_id IS NULL;
```

Data Output Messages Notifications



Showing rows: 1 to 1 Page No: 1 of 1

	actor_id [PK] integer	actor_name text
1	201	Tony Stark

✓ Successfully run. Total query runtime: 116 msec. 1 rows affected.

Total rows: 1

Query complete 00:00:00.116

CRLF Ln 273, Col 1