**Rockbuster Stealth PostgreSQL queries:**

**Contents:**



## 1. Selecting

Rockbuster/postgres@PostgreSQL 14

Query Editor | Query History

```
1  SELECT * FROM category
```

Data Output | Explain | Messages | Notifications

| | category_id [PK] integer | name character varying (25) | last_update timestamp without time zone |
|---|---|---|---|
| 1 | 1 | Action | 2006-02-15 09:46:27 |
| 2 | 2 | Animation | 2006-02-15 09:46:27 |
| 3 | 3 | Children | 2006-02-15 09:46:27 |
| 4 | 4 | Classics | 2006-02-15 09:46:27 |
| 5 | 5 | Comedy | 2006-02-15 09:46:27 |
| 6 | 6 | Documentary | 2006-02-15 09:46:27 |
| 7 | 7 | Drama | 2006-02-15 09:46:27 |
| 8 | 8 | Family | 2006-02-15 09:46:27 |
| 9 | 9 | Foreign | 2006-02-15 09:46:27 |
| 10 | 10 | Games | 2006-02-15 09:46:27 |
| 11 | 11 | Horror | 2006-02-15 09:46:27 |
| 12 | 12 | Music | 2006-02-15 09:46:27 |
| 13 | 13 | New | 2006-02-15 09:46:27 |
| 14 | 14 | Sci-Fi | 2006-02-15 09:46:27 |
| 15 | 15 | Sports | 2006-02-15 09:46:27 |
| 16 | 16 | Travel | 2006-02-15 09:46:27 |

## 2. Inserting rows

```sql
INSERT INTO category(category_id,name,last_update)
VALUES(21,'War',current_timestamp)
```

```sql
INSERT INTO category(category_id,name,last_update)
VALUES(20,'Romance',current_timestamp)
```

```sql
INSERT INTO category(category_id,name,last_update)
VALUES(19,'Mystery',current_timestamp)
```

```sql
INSERT INTO category(category_id,name,last_update)
VALUES(18,'Crime',current_timestamp)
```

```sql
INSERT INTO category(category_id,name,last_update)
VALUES(17,'Thriller',current_timestamp)
```

## 3. Updating and deleting

```sql
UPDATE film_category
SET category_id = 17
WHERE film_id = 5
```

```sql
DELETE FROM category
WHERE category_id = 19
```

## 4. Creating Tables

Query Editor  Query History

```sql
CREATE TABLE employees
(
    employee_id serial NOT NULL,
    name VARCHAR(50),
    contact_number INT,
    designation_id INT,
    last_update TIMESTAMP NOT NULL DEFAULT NOW(),
    CONSTRAINT employees_pkey PRIMARY KEY (employee_id)
);
```

Data Output  Explain  Messages  Notifications

| employee_id [PK] integer | name character varying (50) | contact_number integer | designation_id integer | last_update timestamp without time zone |
|---|---|---|---|---|
| | | | | |

## 5. Ordering

```sql
SELECT * FROM film
ORDER BY
    title,
    release_year DESC,
    rental_rate DESC;

```

## 6. Aggregating and Grouping

```sql
SELECT rating,
        AVG(rental_rate) AS avg_rental_rate,
        MIN(rental_duration) AS min_rental_duration,
        MAX(rental_duration) AS max_rental_duration
FROM film
GROUP BY rating

```

Data Output Explain Messages Notifications

| | rating mpaa_rating | avg_rental_rate numeric | min_rental_duration smallint | max_rental_duration smallint |
|---|---|---|---|---|
| 1 | R | 2.9387179487179487 | 3 | 7 |
| 2 | NC-17 | 2.970952380952381 | 3 | 7 |
| 3 | G | 2.888876404494382 | 3 | 7 |
| 4 | PG | 3.0518556701030928 | 3 | 7 |
| 5 | PG-13 | 3.034843049327354 | 3 | 7 |

## 7. Using 'CASE'

```sql
SELECT rating,
        MIN(replacement_cost) AS min_replacement_cost,
        MAX(replacement_cost) AS max_replacement_cost
FROM film
GROUP BY rating
ORDER BY CASE WHEN rating = 'G' THEN 1
                WHEN rating = 'PG' THEN 2
                WHEN rating = 'PG-13' THEN 3
                WHEN rating = 'R' THEN 4
                WHEN rating = 'NC-17' THEN 5
            END;

```

Data Output Explain Messages Notifications

| | rating mpaa_rating | min_replacement_cost numeric | max_replacement_cost numeric |
|---|---|---|---|
| 1 | G | 9.99 | 29.99 |
| 2 | PG | 9.99 | 29.99 |
| 3 | PG-13 | 9.99 | 29.99 |
| 4 | R | 9.99 | 29.99 |
| 5 | NC-17 | 9.99 | 29.99 |

## 8. Checking for duplicate records

Rockbuster/postgres@PostgreSQL 14

Query Editor | Query History

```
SELECT title,
    release_year,
    language_id,
    rental_duration,
    COUNT(*)
FROM film
GROUP BY title,
    release_year,
    language_id,
    rental_duration
HAVING COUNT(*) >1;
```

Data Output | Explain | Messages | Notifications

| title character varying (255) | release_year integer | language_id smallint | rental_duration smallint | count bigint |
|---|---|---|---|---|

## 9. Creating view for unique values

Rockbuster/postgres@PostgreSQL 14

Query Editor | Query History

```
CREATE VIEW unique_film AS
SELECT title,
    release_year,
    language_id,
    rental_duration
FROM film
GROUP BY title,
    release_year,
    language_id,
    rental_duration
```

Data Output | Explain | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 115 msec.
```

## 10. Checking for missing data

Rockbuster/postgres@PostgreSQL 14

Query Editor | Query History | Scratch Pad

```
SELECT
    COUNT(customer_id) AS id_count,
    COUNT(store_id) AS store_count,
    COUNT(first_name) AS first_count,
    COUNT(last_name) AS last_count,
    COUNT(email) AS email_count,
    COUNT(address_id) AS address_count,
    COUNT(activebool) AS bool_count,
    COUNT(create_date) AS create_count,
    COUNT(last_update) AS update_count,
    COUNT(active) AS active_count
FROM customer
```

Data Output | Explain | Messages | Notifications

| | id_count bigint | store_count bigint | first_count bigint | last_count bigint | email_count bigint | address_count bigint | bool_count bigint | create_count bigint | update_count bigint | active_count bigint |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 599 | 599 | 599 | 599 | 599 | 599 | 599 | 599 | 599 | 599 |

## 11. Using inner joins

Rockbuster/postgres@PostgreSQL 14

Query Editor  Query History

```
SELECT
    A.country,
    COUNT(customer_id)
FROM country A
INNER JOIN city B ON A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
GROUP BY A.country
ORDER BY COUNT(customer_id) DESC
LIMIT 10;
```

Data Output  Explain  Messages  Notifications

| | country character varying (50) | count bigint |
|---|---|---|
| 1 | India | 60 |
| 2 | China | 53 |
| 3 | United States | 36 |
| 4 | Japan | 31 |
| 5 | Mexico | 30 |
| 6 | Brazil | 28 |
| 7 | Russian Federation | 28 |
| 8 | Philippines | 20 |
| 9 | Turkey | 15 |
| 10 | Indonesia | 14 |

## 12. Using inner joins with 'WHERE'

Rockbuster/postgres@PostgreSQL 14

Query Editor  Query History

```
SELECT
    B.city,
    A.country,
    COUNT(customer_id)
FROM country A
INNER JOIN city B ON A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
WHERE country IN
    ('India',
     'China',
     'United States',
     'Japan',
     'Mexico',
     'Brazil',
     'Russian Federation',
     'Phillipines',
     'Turkey',
     'Indonesia')
GROUP BY A.country, B.city
ORDER BY COUNT(customer_id) DESC
LIMIT 10;
```

Data Output  Explain  Messages  Notifications

| | city character varying (50) | country character varying (50) | count bigint |
|---|---|---|---|
| 1 | Aurora | United States | 2 |
| 2 | Acua | Mexico | 1 |
| 3 | Citrus Heights | United States | 1 |
| 4 | Iwaki | Japan | 1 |
| 5 | Ambattur | India | 1 |
| 6 | Shanwei | China | 1 |
| 7 | So Leopoldo | Brazil | 1 |
| 8 | Teboksary | Russian Federation | 1 |
| 9 | Tianjin | China | 1 |
| 10 | Cianjur | Indonesia | 1 |

## 13. Using subqueries

```sql
SELECT
    country.country,
    COUNT(DISTINCT customer.customer_id) AS all_customer_count,
    COUNT(DISTINCT top_5.customer_id) AS top_customer_count
FROM country
INNER JOIN city ON country.country_id = city.country_id
INNER JOIN address ON city.city_id = address.city_id
INNER JOIN customer ON address.address_id = customer.address_id
LEFT JOIN
    (SELECT
        E.customer_id,
        D.first_name,
        D.last_name,
        A.country,
        B.city,
        SUM(E.amount) AS total_paid
    FROM country A
    INNER JOIN city B ON A.country_id = B.country_id
    INNER JOIN address C ON B.city_id = C.city_id
    INNER JOIN customer D ON C.address_id = D.address_id
    INNER JOIN payment E ON D.customer_id = E.customer_id
    WHERE country IN
        ('India','China','United States','Japan','Mexico',
         'Brazil','Russian Federation','Phillipines','Turkey','Indonesia')
        AND city IN
        ('Aurora','Acua','Citrus Heights','Iwaki','Ambatttur','Shanwei',
         'So Leopoldo','Teboksary','Tianjin','Cianjur')
    GROUP BY
        E.customer_id,
        D.first_name,
        D.last_name,
        A.country,
        B.city
    ORDER BY total_paid DESC
    LIMIT 5) AS top_5 ON top_5.country = country.country
GROUP BY country.country
ORDER BY top_customer_count DESC
```

Data Output | Explain | Messages | Notifications

| | country character varying (50) | all_customer_count bigint | top_customer_count bigint |
|---|---|---|---|
| 1 | United States | 36 | 2 |
| 2 | Mexico | 30 | 1 |
| 3 | Japan | 31 | 1 |
| 4 | China | 53 | 1 |
| 5 | Anguilla | 1 | 0 |

## 14. Using Common Table Expressions

Rockbuster/postgres@PostgreSQL 14

Query Editor  Query History

```
WITH top_5_cte (amount) AS
    (SELECT
        E.customer_id,
        D.first_name,
        D.last_name,
        A.country,
        B.city,
        SUM(E.amount) AS total_paid
    FROM country A
    INNER JOIN city B ON A.country_id = B.country_id
    INNER JOIN address C ON B.city_id = C.city_id
    INNER JOIN customer D ON C.address_id = D.address_id
    INNER JOIN payment E ON D.customer_id = E.customer_id
    WHERE country IN
        ('India','China','United States','Japan','Mexico',
         'Brazil','Russian Federation','Phillipines','Turkey','Indonesia')
        AND city IN
        ('Aurora','Acua','Citrus Heights','Iwaki','Ambatttur','Shanwei',
         'So Leopoldo','Teboksary','Tianjin','Cianjur')
    GROUP BY
        E.customer_id,
        D.first_name,
        D.last_name,
        A.country,
        B.city
    ORDER BY total_paid DESC
    LIMIT 5)
SELECT AVG(total_paid) AS average_top_5
FROM top_5_cte
```

Data Output  Explain  Messages  Notifications

| | average_top_5 numeric |
|---|---|
| 1 | 102.5560000000000000 |