PostgreSQL Subquery

Query Editor | Query History | Explain | Messages | Notifications

```
--Now, we can get films whose rental rate is higher than the average rental rate:

SELECT
     film_id,
     title,
     rental_rate
FROM
     film
WHERE
     rental_rate > 2.98;
```

Data Output

	film_id [PK] integer	title character varying (255)	rental_rate numeric (4,2)
1	133	Chamber Italian	4.99
2	384	Grosse Wonderful	4.99
3	8	Airport Pollock	4.99
4	98	Bright Encounters	4.99
5	2	Ace Goldfinger	4.99
6	3	Adaptation Holes	2.99
7	4	Affair Prejudice	2.99
8	5	African Egg	2.99
9	6	Agent Truman	2.99
10	7	Airplane Sierra	4.99
11	9	Alabama Devil	2.99
12	10	Aladdin Calendar	4.99
13	13	Ali Forever	4.99

```
--To construct a subquery, we put the second query in brackets and use it in the WHERE clause as an expression:

SELECT
    film_id,
    title,
    rental_rate
FROM
    film
WHERE
    rental_rate > (
        SELECT
            AVG (rental_rate)
        FROM
            film
    );
```

Data Output

	film_id [PK] integer	title character varying (255)	rental_rate numeric (4,2)
1	133	Chamber Italian	4.99
2	384	Grosse Wonderful	4.99
3	8	Airport Pollock	4.99
4	98	Bright Encounters	4.99
5	2	Ace Goldfinger	4.99
6	3	Adaptation Holes	2.99
7	4	Affair Prejudice	2.99
8	5	African Egg	2.99
9	6	Agent Truman	2.99

```
--to get films that have the returned date between 2005-05-29 and 2005-05-30, you use the following query:

SELECT
    inventory.film_id
FROM
    rental
INNER JOIN inventory ON inventory.inventory_id = rental.inventory_id
WHERE
    return_date BETWEEN '2005-05-29'
AND '2005-05-30';
```

Data Output

	film_id smallint
1	15
2	19
3	45
4	50
5	52
6	54
7	68

Query Editor | Query History | Explain | Messages | Notifications

```sql
--It returns multiple rows so we can use this query as a subquery in the WHERE clause of a query as follows:

SELECT
    film_id,
    title
FROM
    film
WHERE
    film_id IN (
        SELECT
            inventory.film_id
        FROM
            rental
```

Data Output

	film_id [PK] integer	title character varying (255)
1	307	Fellowship Autumn
2	255	Driving Polish
3	388	Gunfight Moon
4	130	Celebrity Horn
5	563	Massacre Usual
6	397	Hanky October
7	898	Tourist Pelican
8	228	Detective Vision
9	347	Games Bowfinger
10	1000	Zorro Ark

```sql
--The EXISTS operator only cares about the number of rows returned from the subquery,
--not the content of the rows, therefore, the common coding convention of EXISTS operator is as follows:

SELECT
    first_name,
    last_name
FROM
    customer
WHERE
    EXISTS (
        SELECT
            1
        FROM
            payment
        WHERE
            payment.customer_id = customer.customer_id
    );
```

Data Output

	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