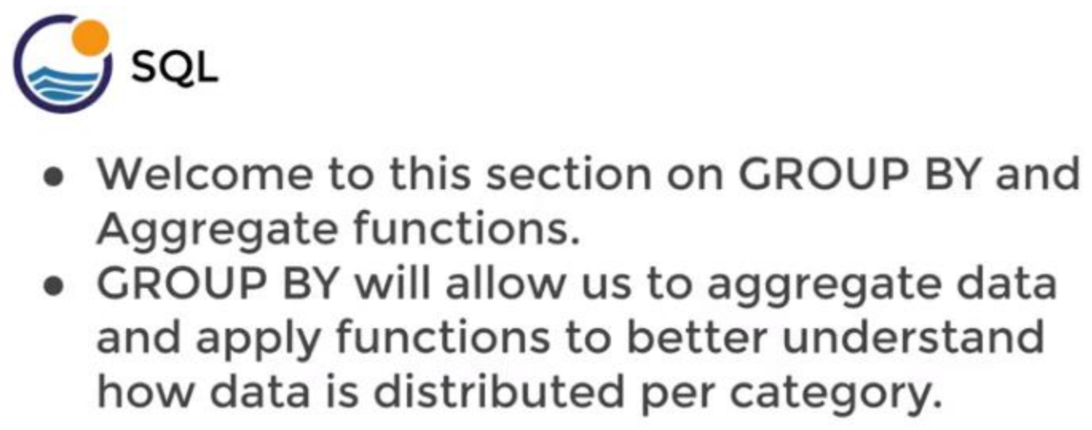
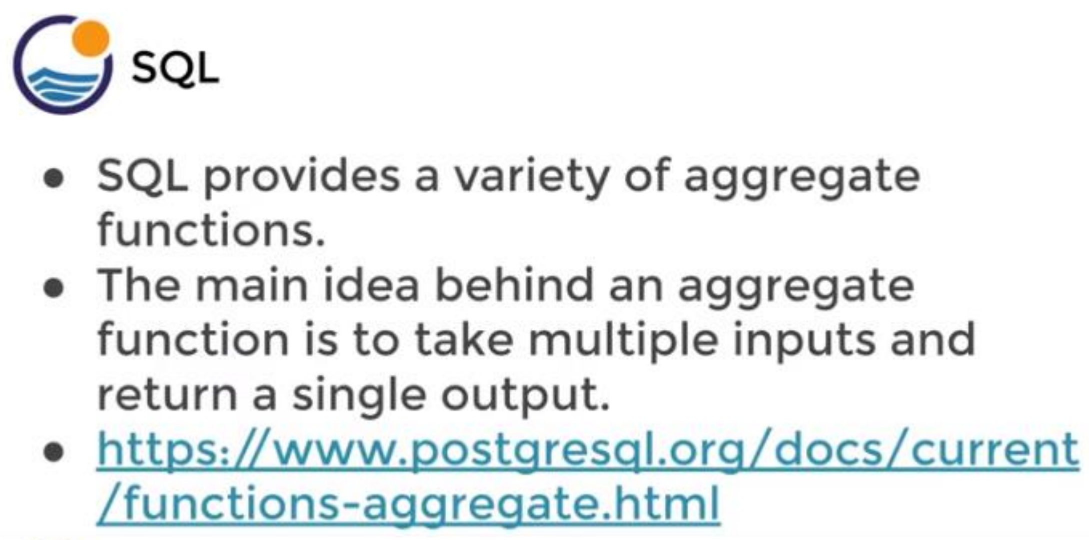
GROUP BY:



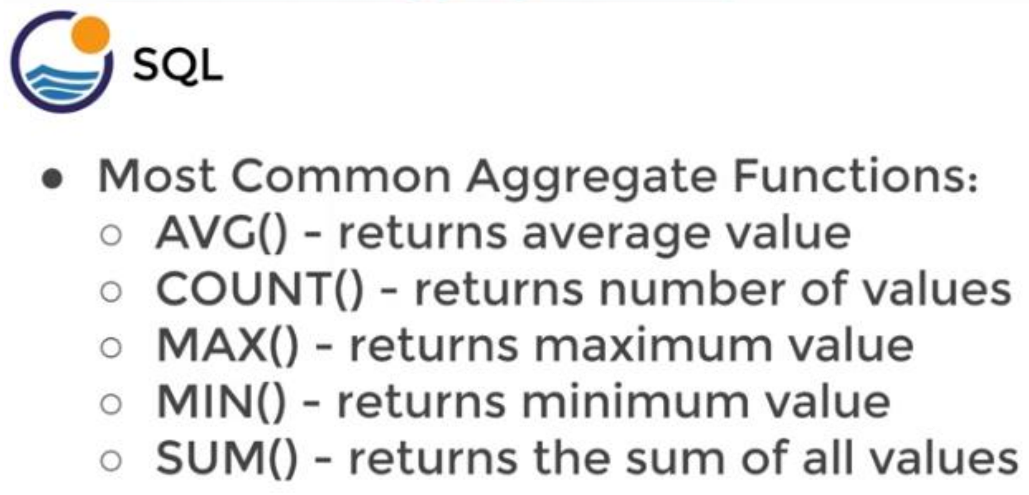
SQL

- Welcome to this section on GROUP BY and Aggregate functions.
- GROUP BY will allow us to aggregate data and apply functions to better understand how data is distributed per category.



SQL

- SQL provides a variety of aggregate functions.
- The main idea behind an aggregate function is to take multiple inputs and return a single output.
- https://www.postgresql.org/docs/current/functions-aggregate.html

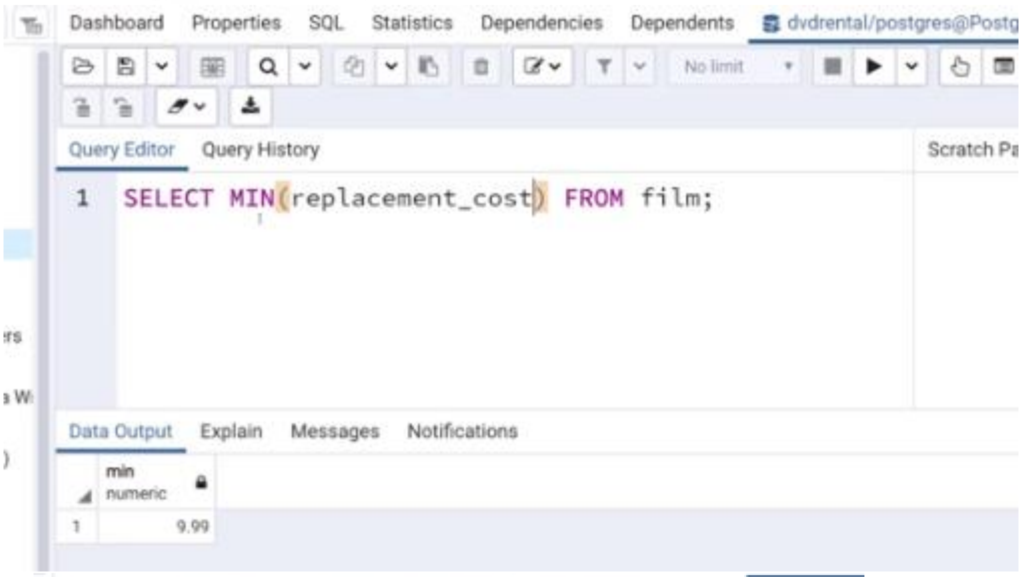


SQL

- Most Common Aggregate Functions:
  - AVG() - returns average value
  - COUNT() - returns number of values
  - MAX() - returns maximum value
  - MIN() - returns minimum value
  - SUM() - returns the sum of all values

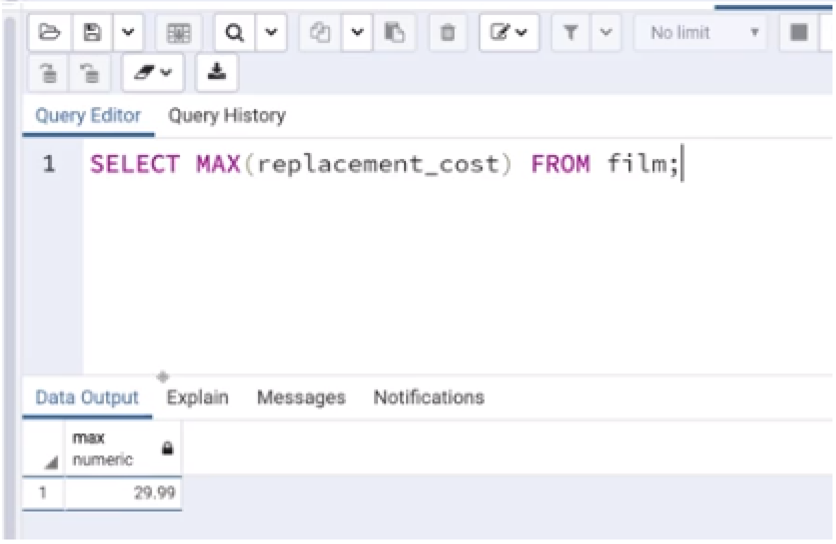
SQL

- Special Notes
  - AVG() returns a floating point value many decimal places (e.g. 2.342418…)
    - You can use ROUND() to specify precision after the decimal.
  - COUNT() simply returns the number of rows, which means by convention we just use COUNT(*)



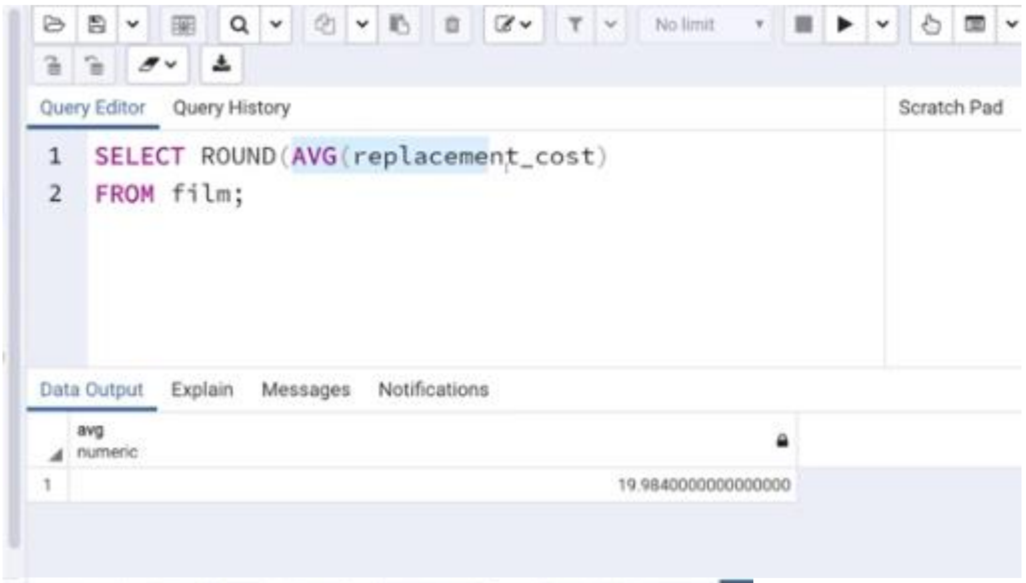
```
SELECT MIN(replacement_cost) FROM film;
```

```
  min
  numeric
1    9.99
```



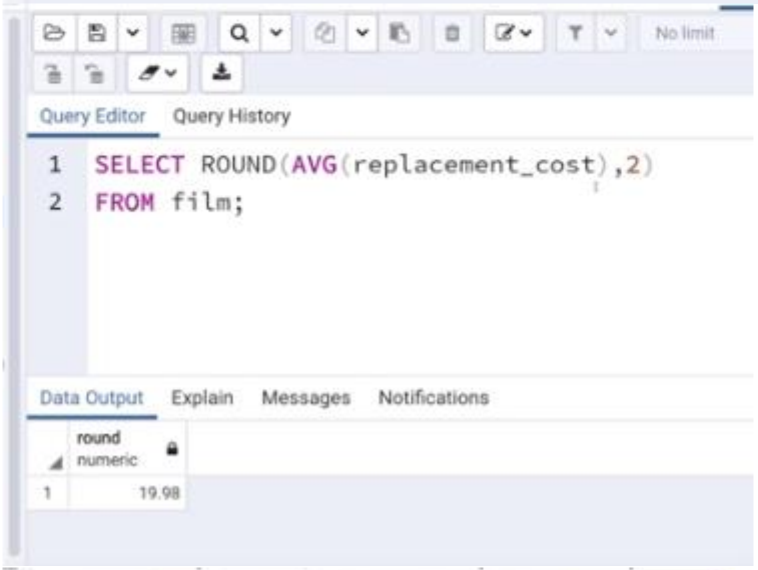
```
SELECT MAX(replacement_cost) FROM film;
```

```
  max
  numeric
1   29.99
```



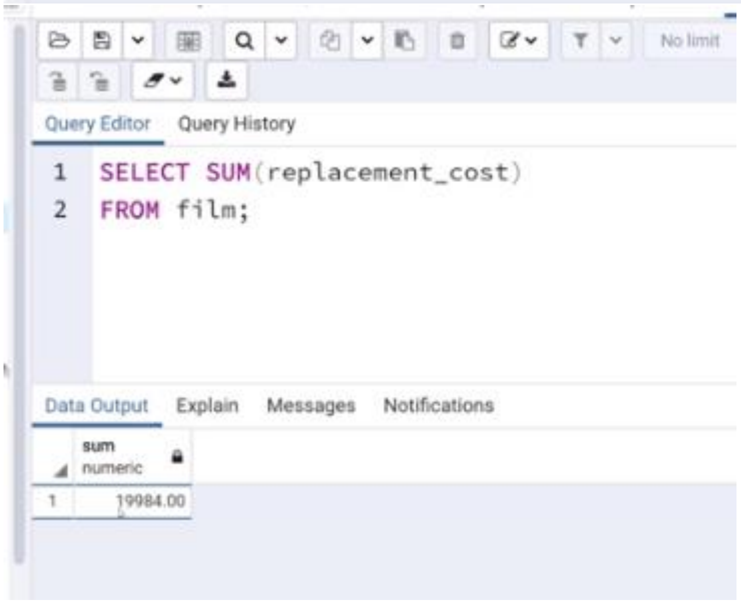
```sql
SELECT ROUND(AVG(replacement_cost)
FROM film;
```

```
avg
numeric
1    19.9840000000000000
```



```sql
SELECT ROUND(AVG(replacement_cost),2)
FROM film;
```

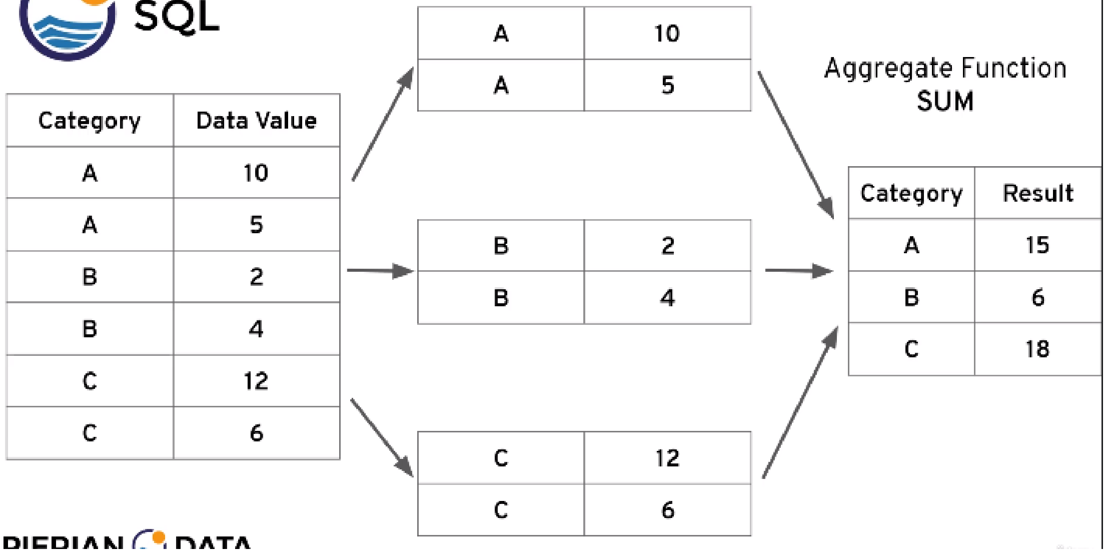
```
round
numeric
1    19.98
```



```sql
SELECT SUM(replacement_cost)
FROM film;
```

```
sum
numeric
1    19984.00
```

SQL



Category	Data Value
A	10
A	5
B	2
B	4
C	12
C	6

A	10
A	5

B	2
B	4

C	12
C	6

Category	Result
A	15
B	6
C	18

SQL

- SELECT category_col , AGG(data_col)
  FROM table
  GROUP BY category_col

SQL

- SELECT category_col , AGG(data_col)
  FROM table
  WHERE category_col != ‘A’
  GROUP BY category_col

- The GROUP BY clause must appear right after a FROM or WHERE statement.

SQL

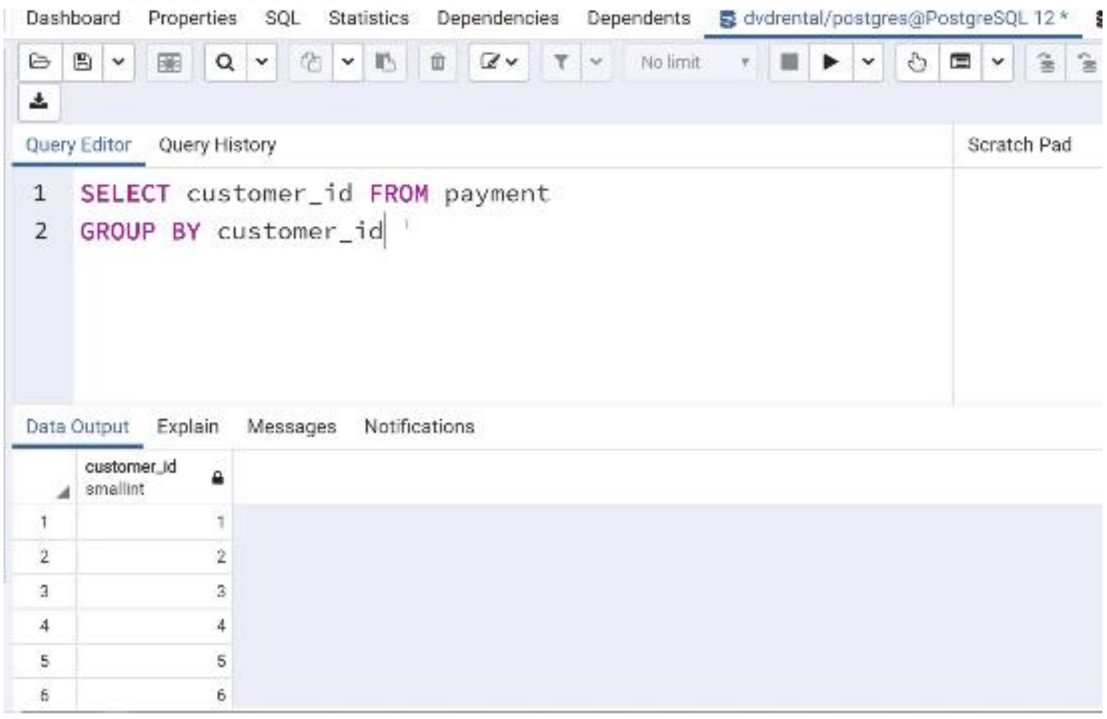
- SELECT category_col , AGG(data_col)
  FROM table
  GROUP BY category_col

- In the SELECT statement, columns must either have an aggregate function or be in the GROUP BY call.

SQL

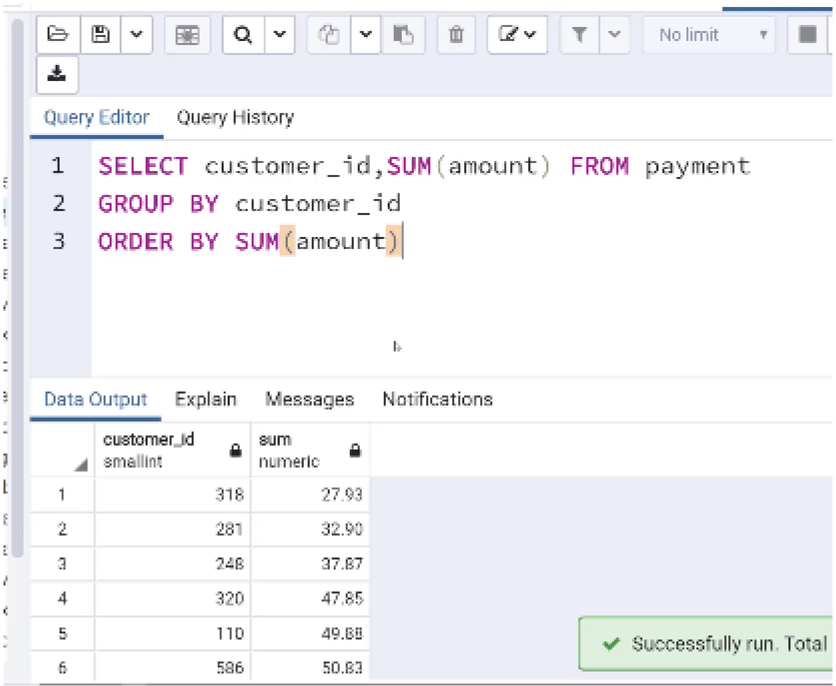
- SELECT company, SUM(sales)
  FROM finance_table
  GROUP BY company
  ORDER BY SUM(sales)

- If you want to sort results based on the aggregate, make sure to reference the entire function



```
SELECT customer_id FROM payment
GROUP BY customer_id
```

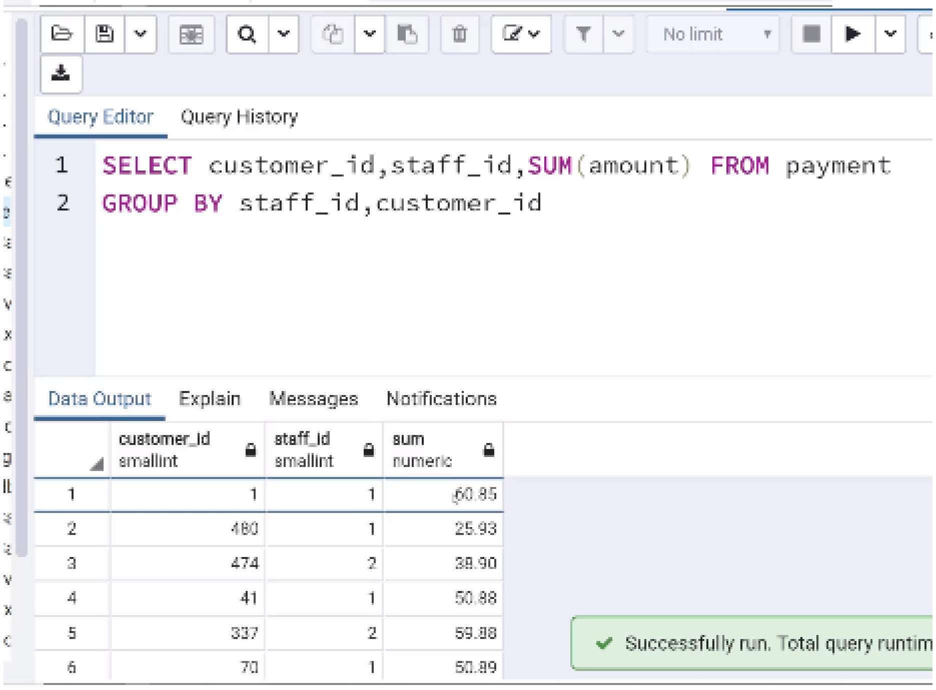
	customer_id smallint
1	1
2	2
3	3
4	4
5	5
6	6



```sql
SELECT customer_id,SUM(amount) FROM payment
GROUP BY customer_id
ORDER BY SUM(amount)
```

	customer_id smallint	sum numeric
1	318	27.93
2	281	32.90
3	248	37.87
4	320	47.85
5	110	49.88
6	586	50.83

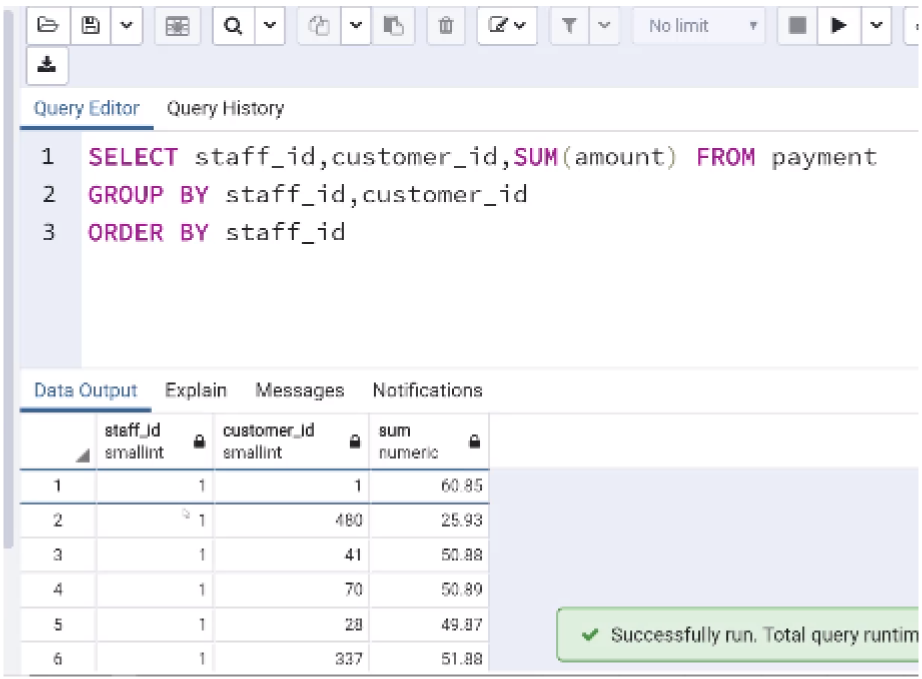
✓ Successfully run. Total



```sql
SELECT customer_id,staff_id,SUM(amount) FROM payment
GROUP BY staff_id,customer_id
```

	customer_id smallint	staff_id smallint	sum numeric
1	1	1	60.85
2	480	1	25.93
3	474	2	38.90
4	41	1	50.88
5	337	2	59.88
6	70	1	50.89

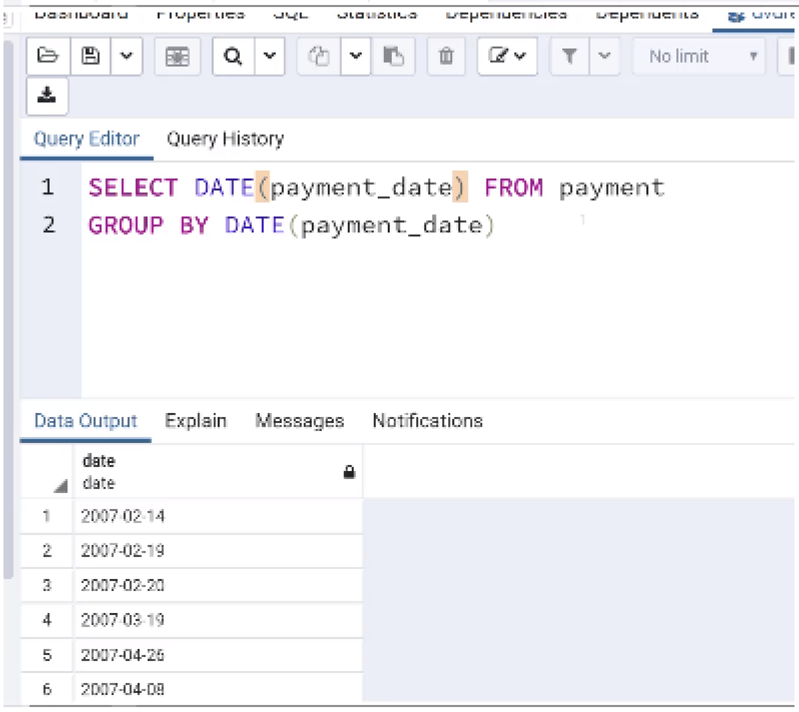
✓ Successfully run. Total query runtim



```sql
SELECT staff_id,customer_id,SUM(amount) FROM payment
GROUP BY staff_id,customer_id
ORDER BY staff_id
```

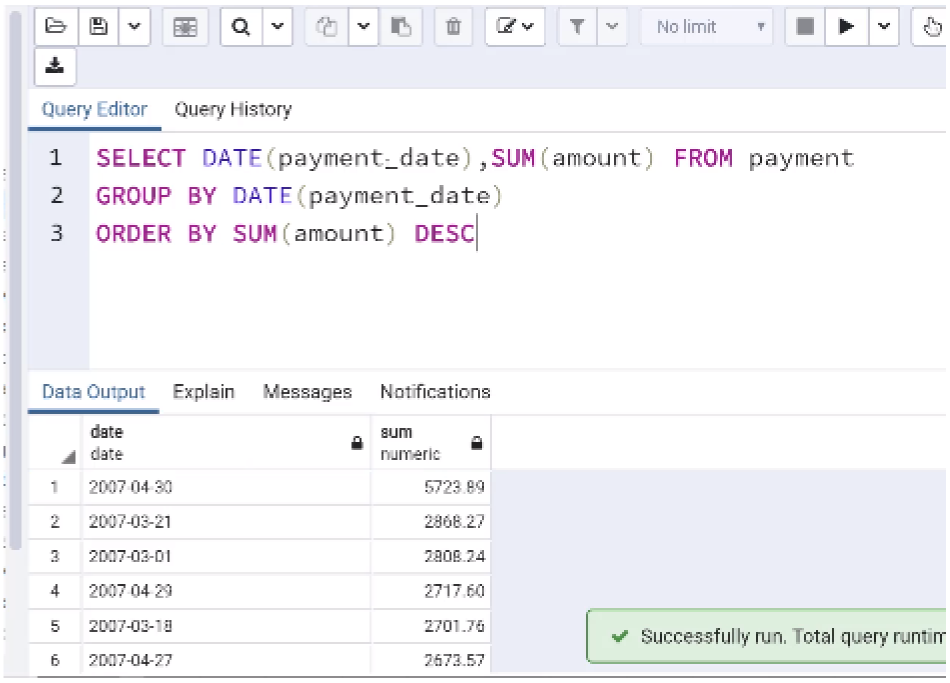
	staff_id smallint	customer_id smallint	sum numeric
1	1	1	60.85
2	1	460	25.93
3	1	41	50.88
4	1	70	50.89
5	1	28	49.87
6	1	337	51.88

Successfully run. Total query runtim



```sql
SELECT DATE(payment_date) FROM payment
GROUP BY DATE(payment_date)
```

	date date
1	2007-02-14
2	2007-02-19
3	2007-02-20
4	2007-03-19
5	2007-04-26
6	2007-04-08

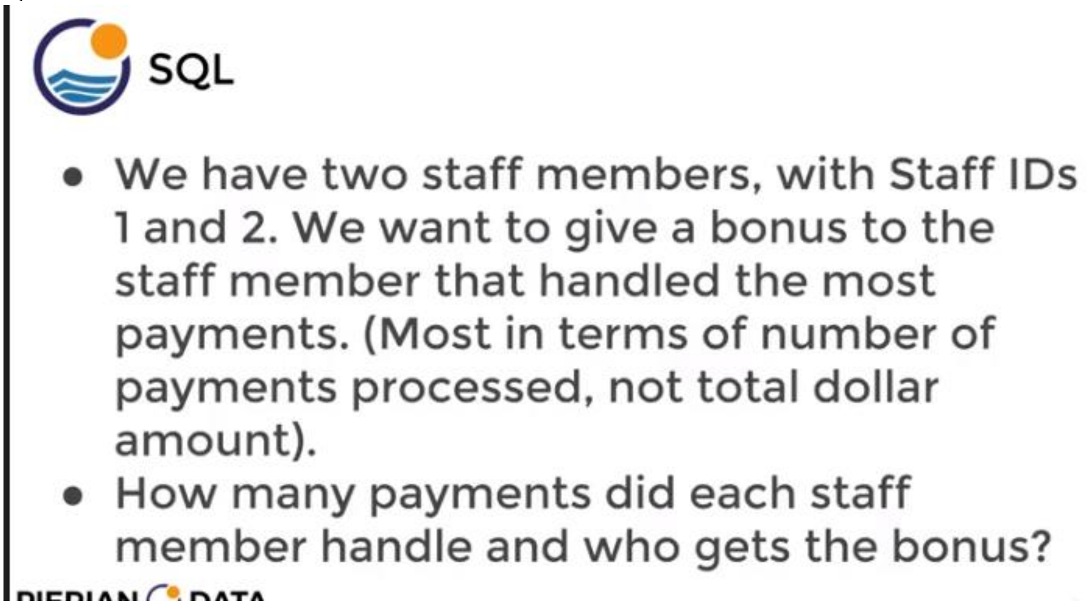


```sql
SELECT DATE(payment_date),SUM(amount) FROM payment
GROUP BY DATE(payment_date)
ORDER BY SUM(amount) DESC
```

	date date	sum numeric
1	2007-04-30	5723.89
2	2007-03-21	2868.27
3	2007-03-01	2808.24
4	2007-04-29	2717.60
5	2007-03-18	2701.76
6	2007-04-27	2673.57

Successfully run. Total query runtim

Q & A:



SQL

- We have two staff members, with Staff IDs 1 and 2. We want to give a bonus to the staff member that handled the most payments. (Most in terms of number of payments processed, not total dollar amount).
- How many payments did each staff member handle and who gets the bonus?

PIERIAN DATA

SQL

- Solution

  -
    ```
    SELECT staff_id,COUNT(amount)
    FROM payment
    GROUP BY staff_id
    ```

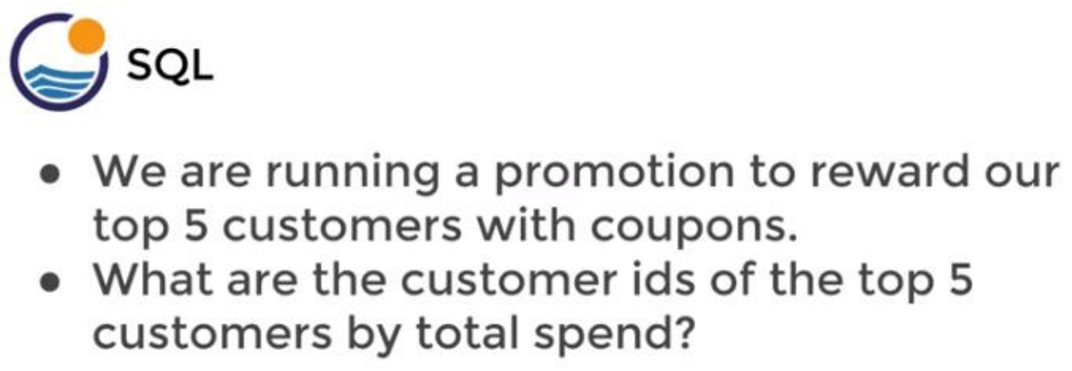
SQL

- Corporate HQ is conducting a study on the relationship between replacement cost and a movie MPAA rating (e.g. G, PG, R, etc…).
- What is the average replacement cost per MPAA rating?
  - Note: You may need to expand the AVG column to view correct results

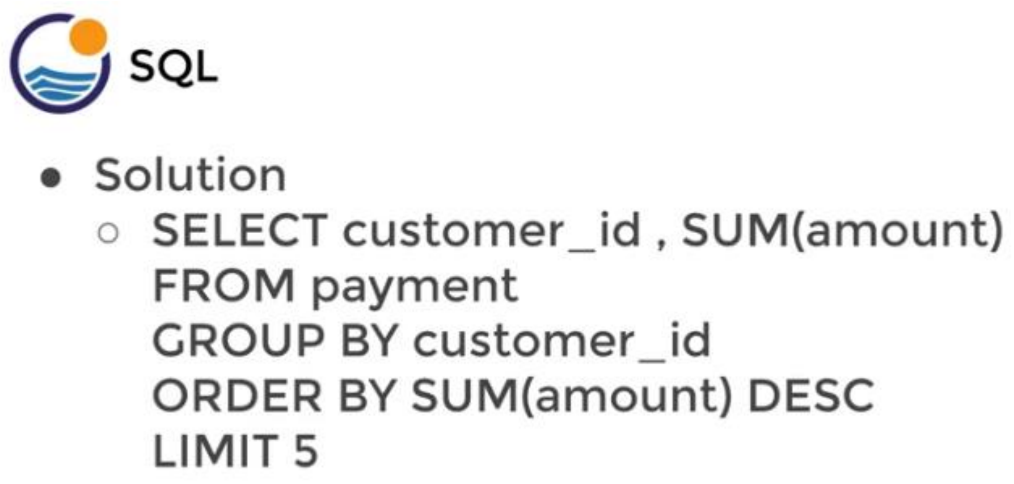
SQL

- Solution

  -
    ```
    SELECT rating , AVG(replacement_cost)
    FROM film
    GROUP BY rating
    ```


SQL

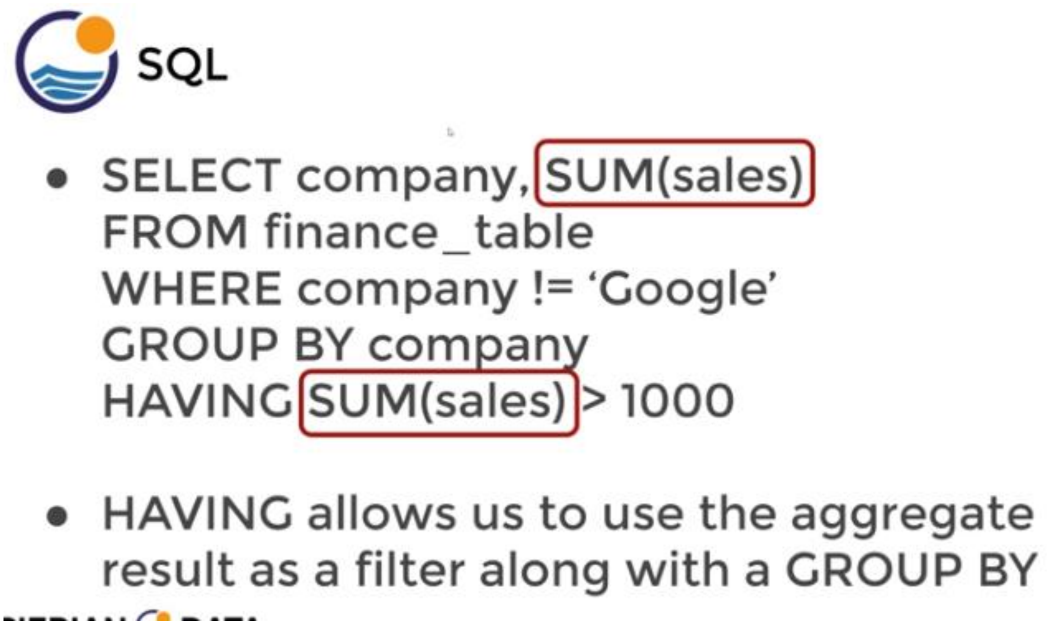
- We are running a promotion to reward our top 5 customers with coupons.
- What are the customer ids of the top 5 customers by total spend?


SQL

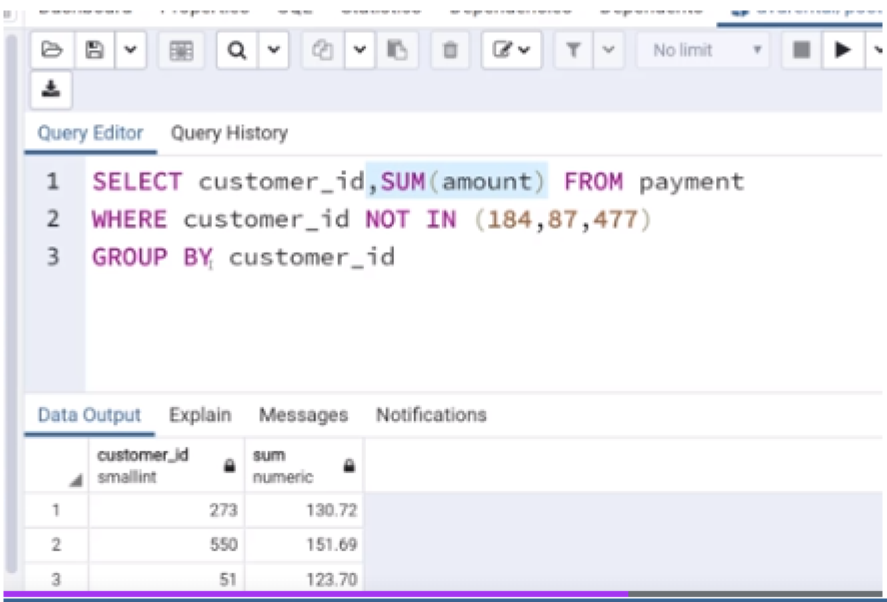
- Solution

  -
    ```
    SELECT customer_id , SUM(amount)
    FROM payment
    GROUP BY customer_id
    ORDER BY SUM(amount) DESC
    LIMIT 5
    ```

HAVING :

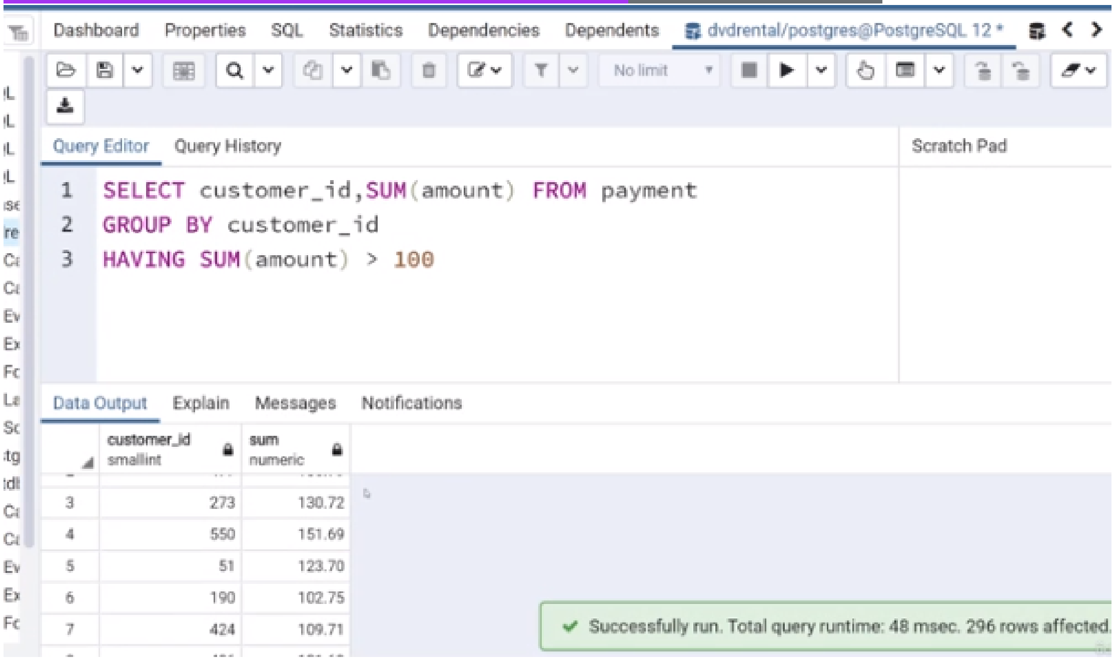

SQL

-
  ```
  SELECT company, SUM(sales)
  FROM finance_table
  WHERE company != 'Google'
  GROUP BY company
  HAVING SUM(sales) > 1000
  ```
- HAVING allows us to use the aggregate result as a filter along with a GROUP BY



```sql
SELECT customer_id,SUM(amount) FROM payment
WHERE customer_id NOT IN (184,87,477)
GROUP BY customer_id
```

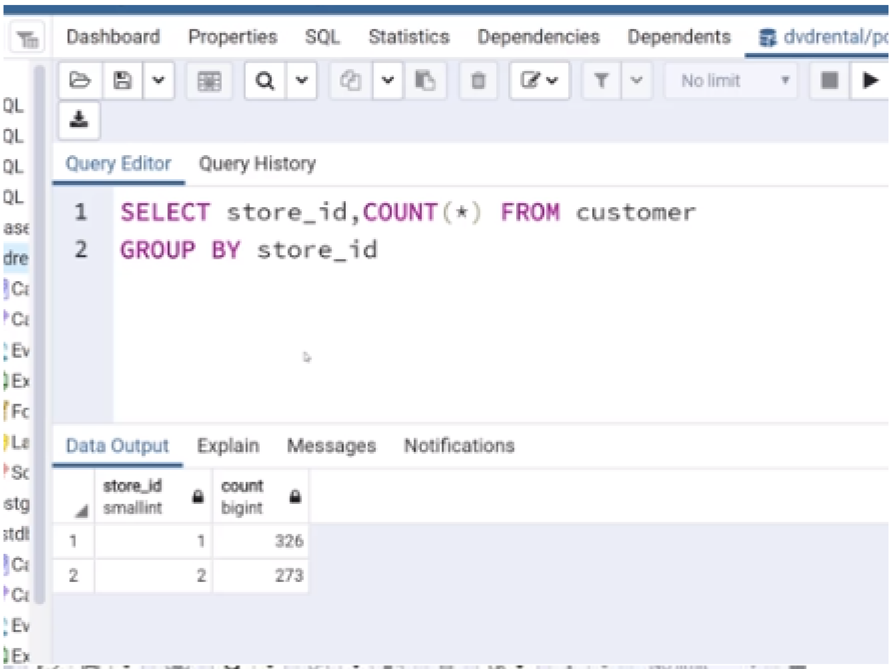
	customer_id smallint	sum numeric
1	273	130.72
2	550	151.69
3	51	123.70



```sql
SELECT customer_id,SUM(amount) FROM payment
GROUP BY customer_id
HAVING SUM(amount) > 100
```

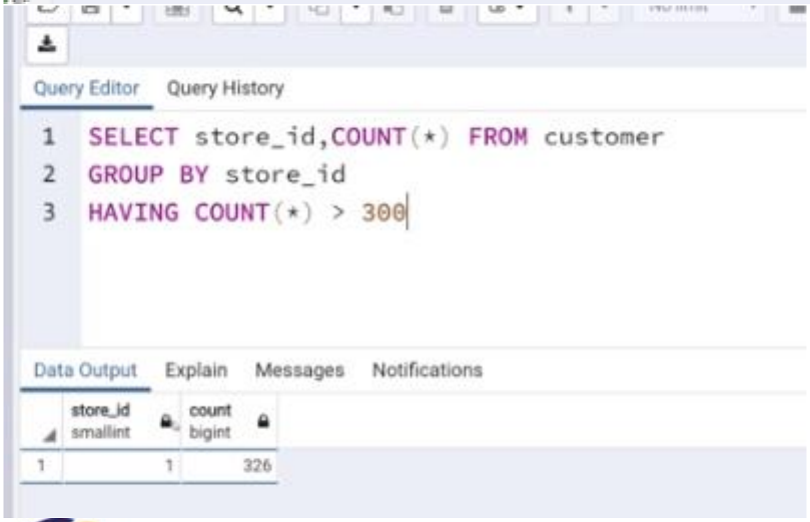
	customer_id smallint	sum numeric
3	273	130.72
4	550	151.69
5	51	123.70
6	190	102.75
7	424	109.71

Successfully run. Total query runtime: 48 msec. 296 rows affected.



```
SELECT store_id,COUNT(*) FROM customer
GROUP BY store_id
```

	store_id smallint	count bigint
1	1	326
2	2	273

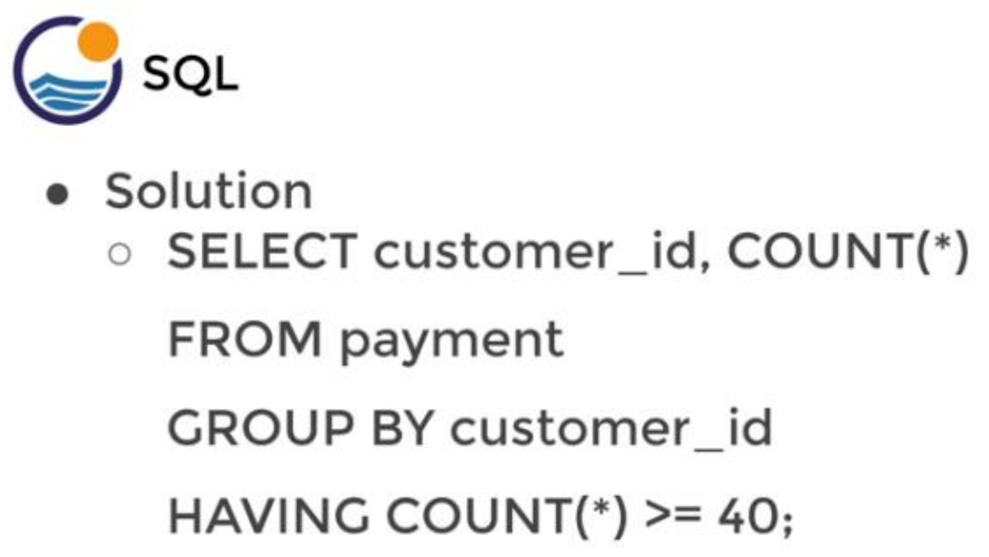


```
SELECT store_id,COUNT(*) FROM customer
GROUP BY store_id
HAVING COUNT(*) > 300
```

	store_id smallint	count bigint
1	1	326

SQL

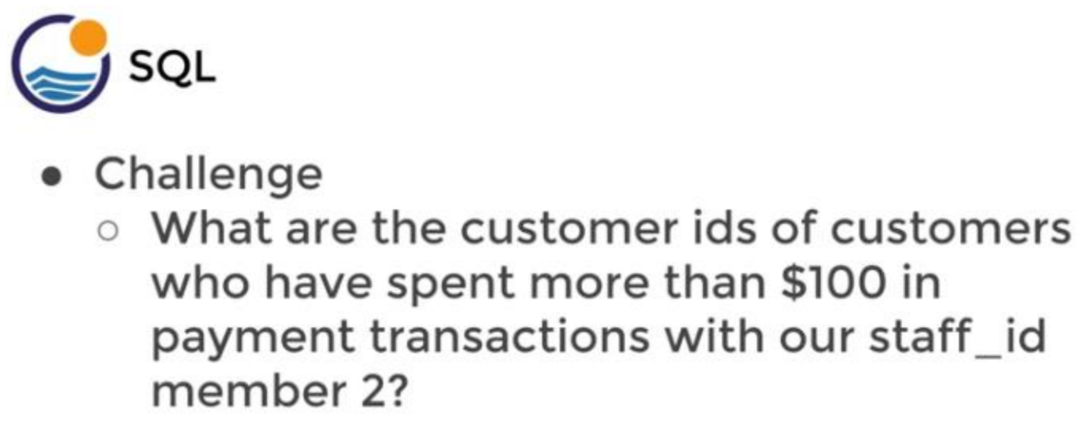
- Challenge
  - We are launching a platinum service for our most loyal customers. We will assign platinum status to customers that have had 40 or more transaction payments.
  - What customer_ids are eligible for platinum status?



SQL

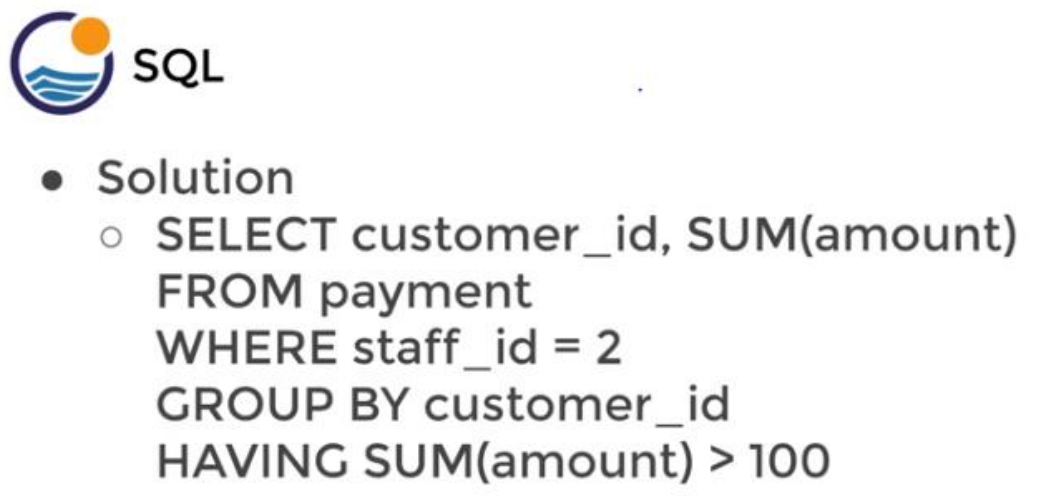
- Solution

  -
    ```
    SELECT customer_id, COUNT(*)

    FROM payment

    GROUP BY customer_id

    HAVING COUNT(*) >= 40;
    ```



SQL

- Challenge
  - What are the customer ids of customers who have spent more than $100 in payment transactions with our staff_id member 2?



SQL

- Solution

  -
    ```
    SELECT customer_id, SUM(amount)
    FROM payment
    WHERE staff_id = 2
    GROUP BY customer_id
    HAVING SUM(amount) > 100
    ```

ASSESSMENT:

ASSESSMENT TEST 1

COMPLETE THE FOLLOWING TASKS!

1. Return the customer IDs of customers who have spent at least $110 with the staff member who has an ID of 2.

The answer should be customers 187 and 148.

2. How many films begin with the letter J?

The answer should be 20.

3. What customer has the highest customer ID number whose name starts **with** an 'E' **and** has an address ID lower than 500?

The answer is Eddie Tomlin

1. Solution Below:

```
SELECT customer_id,SUM(amount)

FROM payment

WHERE staff_id = 2

GROUP BY customer_id

HAVING SUM(amount) > 110;
```

2. Solution Below:

```
SELECT COUNT(*) FROM film

WHERE title LIKE 'J%';
```

3. Solution Below:

```
SELECT first_name,last_name FROM customer

WHERE first_name LIKE 'E%'

AND address_id <500

ORDER BY customer_id DESC

LIMIT 1;
```