59- Most common aggregation functions:

- MIN()
- MAX()
- SUM()
- AVG()
- COUNT()

60- If not using GROUP BY (grouping) we cannot select aggregated column and non-aggregated column togather in same one query.

61- We can select (display) multiple aggregated column in single query.

62- Display the total amount(column) of payment and average amount(column) of payment rounded to 2 decimal places from the table(payment).

```
SELECT
SUM(amount) as "Total Amount"
,ROUND(AVG(amount),2) as Average_Amount
FROM payment
```

63- Display miniimum, maximum, sum, average replacement cost (column) of films from table(film).

```
SELECT
MIN(replacement_cost) as min_replacement_cost
, MAX(replacement_cost) as max_replacement_cost
, SUM(replacement_cost) as sum_replacement_cost
, ROUND(AVG(replacement_cost),2) as avg_replacement_cost
FROM film
```

64- GROUP BY clause comes after WHERE clause if WHERE clause is present.

65- Display total amount(column) spend by each customer(cutomer_id column) from table(payment) in descending order of total amount spend.

```
SELECT
customer_id
, SUM(amount) as total_amount
FROM payment
GROUP BY customer_id
ORDER BY SUM(amount) DESC
```

65- Display which of two staff member is responsible for more count of payments and total sum of payments made by customer from table(payment) in descending order of sum and count.
(do not count payments with amount=0)

```
SELECT
staff_id
,COUNT(amount)
,SUM(amount)
FROM payment
WHERE amount <> 0
GROUP BY staff_id
ORDER BY SUM(amount) DESC, COUNT(amount) DESC
```

66- Display the amount of payment and number of payment between each customer & staff pair from the table(payment), order it in descending order of amount.

```
SELECT
staff_id
,customer_id
,COUNT(payment_id) as no_of_payments
,SUM(amount) as sum_of_payments
FROM payment
GROUP BY staff_id,customer_id
ORDER BY sum_of_payments DESC
```

67- NOTE: function DATE(’2020-01-25 03:10:19.996577+05:30’) returns ’2020-01-25’.

68- Query to display the staff_id and amount of employee who had maximum payment amount in a single day.

```
SELECT
staff_id
,DATE(payment_date) AS day
,SUM(amount) AS sum_amount
FROM payment
GROUP BY staff_id, day
ORDER BY sum_amount DESC
LIMIT 1
```

68- Query to display the staff_id and amount of employee who had maximum number of payments in a single day.(do not count payments with amount=0)

```
SELECT
staff_id
,DATE(payment_date) AS day
,COUNT(payment_id) AS payment_count
FROM payment
WHERE amount <> 0
GROUP BY staff_id,day
ORDER BY payment_count DESC
LIMIT 1
```

69- HAVING clause is used to filter data when GROUP BY and/or Aggregation functions are used.

70- Display customer_id(column) and sum amount(column) of customers who have spent a sum total greater than 100 from table(payment).

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