4 GROUP BY Statements

4.1 MIN MAX SUM AVG aggregate functions.

Eg:- SELECT AVG(amount) FROM payment;
— do —ROUND (AVG (amount), 2)— . do —

SELECT MIN(amount) FROM payment;
SELECT amount FROM payment; }
ORDER BY amount; }

SELECT MAX (amount) FROM payment;

SELECT AVG (amount) FROM payment;

SELECT SUM (amount) FROM payment;

GROUP BY

→ The GROUPBY clause divides the rows returned from the SELECT Statement into groups.

→ For each group, you Can apply an aggregate function;

Eg:- → Calculating the sum of items
→ Count the number of items in the groups

Syntax

SELECT Column_1, aggregate_function (Column_2)
FROM table_name
GROUP BY Column_1;

Eg:- SELECT Cst-id * FROM Payment
GROUP BY Custer-id;

```sql
SELECT   Cust_id , SUM (amount)
FROM   payment.
GROUP BY   Cust_id .
ORDER BY   SUM (amount) DESC;


SELECT   Staff_id , COUNT (Payment_id)
FROM   payment.
GROUP BY   staff_id;


SELECT   rating , COUNT (rating)
FROM   film
GROUP BY   rating


SELECT * FROM film.
SELECT rental_duration , COUNT (rental_duration)
FROM film
GROUPBY rental_duration;
```

4.3   HAVING Statement

→ We often use the HAVING Clause in Conjunction with the GROUP BY Clause to filter group rows that do not satisfy a specified Condition.

Syntax

```
SELECT Column-1, aggregate_function(Column-2)
FROM table_name
GROUP BY Column-1
HAVING Condition;
```

→ ~~Having~~ The HAVING Clause sets the Condition for group rows Created by the GROUP BY clause after the GROUP BY Clause applies; while the where clause sets the Condition for individual rows before GROUP BY Clause applies. => This is the difference b/w the two

Eg:-

```
SELECT Cust_id, SUM(amount)
FROM payment
GROUPBY Cust-id.
HAVING SUM(amount) > 200;
```

```
SELECT store_id, COUNT(customer_id)
FROM Customer
GROUP BY Store_id
HAVING COUNT(customer_id) > 300
```

```
SELECT * FROM film.
SELECT rating, rental_rate. AVG(rental_rate)
FROM film
WHERE rating IN ('R','G','PG');
GROUP BY rating;
HAVING AVG(rental_rate) < 3.
```