Aggregate functions :

1. COUNT()

Example 1: Count the number of records in a table.

```sql
-- Example: Count the number of users in a 'users' table
SELECT COUNT(*) AS total_users
FROM users;
```

Example 2: Count the number of distinct values in a column.

```sql
-- Example: Count the number of unique cities in the 'users' table
SELECT COUNT(DISTINCT city) AS unique_cities
FROM users;
```

2. SUM()

Example 1: Calculate the total salary of all employees.

```sql
-- Example: Calculate the total salary of all employees in an 'employees' table
SELECT SUM(salary) AS total_salary
FROM employees;
```

Example 2: Sum a specific column based on a condition.

```sql
-- Example: Sum the sales amount for a specific product category
SELECT SUM(sales_amount) AS total_sales
FROM sales
WHERE product_category = 'Electronics';
```

3. AVG()

Example 1: Calculate the average age of users.

```sql
```

```sql
-- Example: Calculate the average age of users in a 'users' table
SELECT AVG(age) AS average_age
FROM users;
```

Example 2: Calculate the average rating of products.

```sql
-- Example: Calculate the average rating of products in a 'products' table
SELECT AVG(rating) AS average_rating
FROM products;
```

4. MIN()

Example 1: Find the minimum value in a column.

```sql
-- Example: Find the minimum salary in an 'employees' table
SELECT MIN(salary) AS min_salary
FROM employees;
```

Example 2: Find the earliest registration date in a user table.

```sql
-- Example: Find the earliest registration date in a 'users' table
SELECT MIN(registration_date) AS earliest_registration
FROM users;
```

5. MAX()

Example 1: Find the maximum value in a column.

```sql
-- Example: Find the maximum revenue in a 'sales' table
SELECT MAX(revenue) AS max_revenue
FROM sales;
```

Example 2: Find the latest order date in an orders table.

```sql
```

```sql
-- Example: Find the latest order date in an 'orders' table
SELECT MAX(order_date) AS latest_order_date
FROM orders;
```

Certainly! Let's continue with more examples for the remaining aggregate functions:

6. GROUP_CONCAT() (MySQL specific, for concatenating strings)

Example 1: Concatenate names of users by a specific condition.

```sql
-- Example: Concatenate names of users who are from the same city
SELECT city, GROUP_CONCAT(name) AS names_in_city
FROM users
GROUP BY city;
```

Example 2: Concatenate order IDs for each customer.

```sql
-- Example: Concatenate order IDs for each customer in an 'orders' table
SELECT customer_id, GROUP_CONCAT(order_id) AS order_ids
FROM orders
GROUP BY customer_id;
```

7. GROUP BY with COUNT()

Example 1: Count the number of orders for each product.

```sql
-- Example: Count the number of orders for each product in an 'order_items' table
SELECT product_id, COUNT(*) AS order_count
FROM order_items
GROUP BY product_id;
```

Example 2: Count the number of users in each age group.

```sql
-- Example: Count the number of users in each age group in a 'users' table
SELECT CASE
        WHEN age < 18 THEN 'Under 18'
```

```
        WHEN age BETWEEN 18 AND 30 THEN '18-30'
        WHEN age BETWEEN 31 AND 50 THEN '31-50'
        ELSE 'Over 50'
    END AS age_group,
    COUNT(*) AS user_count
FROM users
GROUP BY age_group;
```

8. HAVING Clause (filtering based on aggregate functions)

Example 1: Find products with more than 100 orders.

```sql
-- Example: Find products with more than 100 orders in an 'order_items' table
SELECT product_id, COUNT(*) AS order_count
FROM order_items
GROUP BY product_id
HAVING order_count > 100;
```

Example 2: Find cities with an average age greater than 35.

```sql
-- Example: Find cities with an average age greater than 35 in a 'users' table
SELECT city, AVG(age) AS average_age
FROM users
GROUP BY city
HAVING average_age > 35;
```

These additional examples should give you a more comprehensive understanding of how to use aggregate functions in SQL queries. Remember that the specific details may vary based on the database system you're using.