SQL

Structured Query Language

- Punith B

FUNCTIONS

Functions are the block of program to perform a particular task.

Functions have been categorized as:

- Aggregate Functions
- Character Functions
- Number Functions
- Date Functions
- Window Functions.

Aggregate Functions

This function takes multiple values as input and returns a single value as an output.

The types of Aggregate functions are:

MAX()	MAX() is used to obtain the maximum value of the given column. **Syntax:** MAX(column-name);
MIN()	MIN() is used to obtain the minimum value of the given column. **Syntax:** MIN(column-name);
SUM()	SUM() is used to obtain the total value of the given column. **Syntax:** SUM(column-name);
AVG()	AVG() is used to obtain the average value of the given column. **Syntax:** AVG(column-name);
COUNT()	COUNT() is used to obtain the number of values from the given column. **Syntax:** COUNT(*/column-name);

Note: Only COUNT() takes '*' as an argument.

REQ-1 : WAQTD THE MAXIMUM SALARY AND MINIMUM SAL GIVEN IN THE COMPANY.

```
SELECT MAX(SAL),MIN(SAL)
FROM EMP;
```

```
mysql> SELECT MAX(SAL),MIN(SAL)
    -> FROM EMP;
+-----------+-----------+
| MAX(SAL)  | MIN(SAL) |
+-----------+-----------+
| 500000.00 | 30000.00 |
+-----------+-----------+
1 row in set (0.00 sec)
```

REQ-2 : WAQTD THE TOTAL SALARY, MAX SALARY GIVEN TO ALL THE SALESMAN.

```
SELECT SUM(SAL),MAX(SAL)
FROM EMP
WHERE JOB='SALESMAN';
```

```
mysql> SELECT SUM(SAL),MAX(SAL)
    -> FROM EMP
    -> WHERE JOB='SALESMAN';
+-----------+-----------+
| SUM(SAL)  | MAX(SAL) |
+-----------+-----------+
| 130000.00 | 45000.00 |
+-----------+-----------+
1 row in set (0.01 sec)
```

REQ-3 : WAQTD THE AVG SALARY OBTAINED BY THE DEVELOPER.

```
SELECT AVG(SAL)
FROM EMP
WHERE JOB='DEVELOPER';
```

```
mysql> SELECT AVG(SAL)
    -> FROM EMP
    -> WHERE JOB='DEVELOPER';
+--------------+
| AVG(SAL)     |
+--------------+
| 31000.000000 |
+--------------+
1 row in set (0.00 sec)
```

REQ-4 : WAQTD THE NUMBER OF EMPS IN THE COMPANY.

```
SELECT COUNT(*)
FROM EMP;
```

```
mysql> SELECT COUNT(*)
    -> FROM EMP;
+----------+
| COUNT(*) |
+----------+
|       15 |
+----------+
1 row in set (0.01 sec)
```

Characteristics of Aggregate Functions

- Aggregate functions execute group by group.
- We cannot pass normal columns along with aggregate functions.
- We cannot nest aggregate functions.
- We cannot pass multiple columns inside a single function.
- We cannot pass aggregate function inside WHERE clause.
- Aggregate functions ignore NULL values.
- We can display group-by-expression along with aggregate functions.

GROUP BY clause

GROUP BY clause is used to group the records.

Syntax:

```
SELECT aggregate-functions/group-by-expressions
FROM table-name
[WHERE filter-condition]
GROUP BY column-name;
```

REQ-1 : WAQTD THE NUMBER OF EMPS WORKING IN EACH DEPT.

```
SELECT COUNT(*),DNO
FROM EMP
GROUP BY DNO;
```

```
mysql> SELECT COUNT(*),DNO
    -> FROM EMP
    -> GROUP BY DNO;
+----------+------+
| COUNT(*) | DNO  |
+----------+------+
|        4 |  113 |
|        1 |  114 |
|        4 |  111 |
|        4 |  110 |
|        2 |  112 |
+----------+------+
5 rows in set (0.01 sec)
```

REQ-2 : WAQTD THE NUMBER OF EMPS WHO ARE WORKING AS SALESMAN OR DISPATCHER IN EACH DEPT.

```
SELECT COUNT(*),DNO
FROM EMP
WHERE JOB IN ('SALESMAN','MANAGER')
GROUP BY DNO;
```

```
mysql> SELECT COUNT(*),DNO
    -> FROM EMP
    -> WHERE JOB IN ('SALESMAN','MANAGER')
    -> GROUP BY DNO;
+----------+------+
| COUNT(*) | DNO  |
+----------+------+
|        4 |  111 |
|        1 |  110 |
+----------+------+
2 rows in set (0.04 sec)
```

Characteristics of GROUP BY clause

- Aggregate functions execute group by group.
- We cannot pass normal columns along with aggregate functions.
- We cannot nest aggregate functions.
- We cannot pass multiple columns inside a single function.
- We cannot pass aggregate function inside WHERE clause.
- Aggregate functions ignore NULL values.
- We can display group-by-expression along with aggregate functions.

HAVING CLAUSE

HAVING clause is used to filter the group functions.

Syntax:

```
SELECT aggregate-functions/group-by-expressions
FROM table-name
[WHERE filter-condition]
GROUP BY column-name
HAVING filter-group-function;
```

REQ-1 : WAQTD THE NUMBER OF EMPS GETTING SALARY MORE THAN OR EQUAL TO 30000 AND MAX SALARY LESS THAN OR EQUAL TO 150000WORKING IN EACH DEPT.

```
SELECT COUNT(*),DNO
FROM EMP
WHERE SAL>=30000
GROUP BY DNO
HAVING MAX(SAL)<=150000;
```

```
mysql> SELECT COUNT(*),DNO
    -> FROM EMP
    -> WHERE SAL>=30000
    -> GROUP BY DNO
    -> HAVING MAX(SAL)<=150000;
+----------+------+
| COUNT(*) | DNO  |
+----------+------+
|        1 |  114 |
|        4 |  111 |
|        4 |  110 |
|        2 |  112 |
+----------+------+
4 rows in set (0.05 sec)
```

Characteristics of HAVING clause

- HAVING clause executes group by group.
- We can pass aggregate functions inside HAVING clause.
- It executes after GROUP BY clause.
- We cannot pass aggregate normal condition inside HAVING clause.

ORDER BY Clause

Order By clause is used to arrange the records either in ascending or descending order.

Syntax:

```
SELECT column-name
FROM table-name
ORDER BY column-name ASC/DESC;
```

REQ-1 :

```
mysql> SELECT FNAME,LNAME
    -> FROM EMP
    -> ORDER BY FNAME ASC;
+----------+-----------+
| FNAME    | LNAME     |
+----------+-----------+
| Abhijit  | Gowda     |
| Aman     | Rai       |
| Dharani  | Patil     |
| Fariya   | Taj       |
| Hema     | Shetty    |
| Jahnavi  | Naik      |
| Karan    | Bhat      |
| Kiran    | Raj       |
| Murali   | Krishnan  |
| Priya    | Shetty    |
| Rahul    | Mukharjee |
| Rashmi   | Gowda     |
| Sameer   | Khan      |
| Shivani  | Rai       |
| Siddarth | Patil     |
+----------+-----------+
15 rows in set (0.00 sec)
```

REQ-1 : WAQTD THE DETAILS OF EMP BASED ON THEIR SALARY FROM MAXIMUM TO MINIMUM.

```
SELECT *
FROM EMP
ORDER BY SAL DESC;
```

```
mysql> SELECT *
    -> FROM EMP
    -> ORDER BY SAL DESC;
+------+----------+-----------+------------+--------+-----------+------+------------+-----------+---------+------+------+
| EID  | FNAME    | LNAME     | DOB        | GENDER | JOB       | MGR  | DOJ        | SAL       | COMM    | DNO  | CID  |
+------+----------+-----------+------------+--------+-----------+------+------------+-----------+---------+------+------+
| 1601 | Siddarth | Patil     | 1985-11-24 | M      | Ceo       | NULL | 2016-01-16 | 500000.00 |    NULL |  113 | NULL |
| 1602 | Hema     | Shetty    | 1996-03-20 | F      | Hr        | 1601 | 2016-10-20 | 150000.00 |    NULL |  114 |  507 |
| 1702 | Sameer   | Khan      | 1995-04-20 | M      | Manager   | 1602 | 2017-07-07 | 120000.00 |    NULL |  110 | NULL |
| 1701 | Rahul    | Mukharjee | 1991-02-19 | M      | Manager   | 1602 | 2017-04-17 | 100000.00 |    NULL |  111 | NULL |
| 1902 | Abhijit  | Gowda     | 1997-12-25 | M      | Dispatcher| 1702 | 2019-12-28 |  50000.00 |    NULL |  110 |  505 |
| 1801 | Jahnavi  | Naik      | 1996-04-11 | F      | Dispatcher| 1702 | 2020-03-15 |  45000.00 | 1000.00 |  110 | NULL |
| 1901 | Shivani  | Rai       | 1998-11-07 | F      | Tester    | 1601 | 2019-12-12 |  45000.00 |    NULL |  113 |  502 |
| 1903 | Karan    | Bhat      | 1997-12-26 | M      | Salesman  | 1701 | 2019-12-26 |  45000.00 |    NULL |  111 | NULL |
| 2001 | Murali   | Krishnan  | 1998-06-08 | M      | Dispatcher| 1702 | 2020-03-15 |  45000.00 | 1000.00 |  110 | NULL |
| 2101 | Rashmi   | Gowda     | 1995-10-03 | F      | Salesman  | 1701 | 2021-01-02 |  45000.00 | 3000.00 |  111 | NULL |
| 2104 | Aman     | Rai       | 1998-08-15 | M      | Salesman  | 1701 | 2021-12-26 |  40000.00 |    NULL |  111 | NULL |
| 2102 | Fariya   | Taj       | 1999-01-03 | F      | Developer | 1601 | 2021-03-01 |  32000.00 | 3600.00 |  113 | NULL |
| 2103 | Priya    | Shetty    | 1998-03-20 | F      | Accountant| 1602 | 2021-05-01 |  32000.00 | 3600.00 |  112 | NULL |
| 2002 | Dharani  | Patil     | 1998-11-10 | F      | Developer | 1601 | 2021-06-20 |  30000.00 | 3000.00 |  113 | NULL |
| 2201 | Kiran    | Raj       | 1999-09-21 | M      | Accountant| 1602 | 2022-08-28 |  30000.00 | 3600.00 |  112 |  503 |
+------+----------+-----------+------------+--------+-----------+------+------------+-----------+---------+------+------+
15 rows in set (0.01 sec)
```

REQ-2 : WAQTD THE EMP FNAME, LNAME ACCORDING TO ALPHABETICAL ORDER.

```
SELECT FNAME,LNAME
FROM EMP
ORDER BY LNAME ASC;
```

```
mysql> SELECT FNAME,LNAME
    -> FROM EMP
    -> ORDER BY LNAME ASC;
+----------+-----------+
| FNAME    | LNAME     |
+----------+-----------+
| Karan    | Bhat      |
| Abhijit  | Gowda     |
| Rashmi   | Gowda     |
| Sameer   | Khan      |
| Murali   | Krishnan  |
| Rahul    | Mukharjee |
| Jahnavi  | Naik      |
| Siddarth | Patil     |
| Dharani  | Patil     |
| Shivani  | Rai       |
| Aman     | Rai       |
| Kiran    | Raj       |
| Hema     | Shetty    |
| Priya    | Shetty    |
| Fariya   | Taj       |
+----------+-----------+
15 rows in set (0.00 sec)
```

LIMIT and OFFSET

- LIMIT is used to return the specified number of records from the table.
- OFFSET is used to ignore the specified number of records from the table.

REQ-1 : WAQTD THE TOP 3 RECORDS FROM THE EMP TABLE.

```
SELECT *
FROM EMP
LIMIT 3;
```

```
mysql> SELECT *
    -> FROM EMP
    -> LIMIT 3;
+------+----------+-----------+------------+--------+---------+------+------------+-----------+------+------+------+
| EID  | FNAME    | LNAME     | DOB        | GENDER | JOB     | MGR  | DOJ        | SAL       | COMM | DNO  | CID  |
+------+----------+-----------+------------+--------+---------+------+------------+-----------+------+------+------+
| 1601 | Siddarth | Patil     | 1985-11-24 | M      | Ceo     | NULL | 2016-01-16 | 500000.00 | NULL |  113 | NULL |
| 1602 | Hema     | Shetty    | 1996-03-20 | F      | Hr      | 1601 | 2016-10-20 | 150000.00 | NULL |  114 |  507 |
| 1701 | Rahul    | Mukharjee | 1991-02-19 | M      | Manager | 1602 | 2017-04-17 | 100000.00 | NULL |  111 | NULL |
+------+----------+-----------+------------+--------+---------+------+------------+-----------+------+------+------+
3 rows in set (0.01 sec)
```

REQ-2 : WAQTD THE TOP 3RD RECORDS FROM THE EMP TABLE.

```
SELECT *
FROM EMP
LIMIT 1 OFFSET 2;
```

```
mysql> SELECT *
    -> FROM EMP
    -> LIMIT 1 OFFSET 2;
+------+-------+-----------+------------+--------+---------+------+------------+-----------+------+------+------+
| EID  | FNAME | LNAME     | DOB        | GENDER | JOB     | MGR  | DOJ        | SAL       | COMM | DNO  | CID  |
+------+-------+-----------+------------+--------+---------+------+------------+-----------+------+------+------+
| 1701 | Rahul | Mukharjee | 1991-02-19 | M      | Manager | 1602 | 2017-04-17 | 100000.00 | NULL |  111 | NULL |
+------+-------+-----------+------------+--------+---------+------+------------+-----------+------+------+------+
1 row in set (0.00 sec)
```