Exp No: 8
Date :20-10-2020

Built-in functions in MySQL

AIM

Familiarize various Built-in functions available in MySQL.

Theory and Examples

Aggregate Functions

An aggregate function performs a calculation on multiple values and returns a single value.

AVG()

AVG() function is an aggregate function that allows you to calculate the average value of a set.

Syntax :

SELECT AVG(expression)
FROM Table_name ;

Examples :

select avg(mark) 'avg_mark' from student;

```
mysql> select avg(mark) 'avg_mark' from student;
+----------+
| avg_mark |
+----------+
|  80.0000 |
+----------+
1 row in set (0.00 sec)
```

SUM()

The SUM() function is an aggregate function that allows you to calculate the sum of values in a set.

Syntax :

SELECT SUM(expression)
FROM Table_name ;

Examples :

select sum(mark) 'total_mark' from student;

```
mysql> select sum(mark) 'total_mark' from student;
+------------+
| total_mark |
+------------+
|        240 |
+------------+
1 row in set (0.00 sec)
```

MAX()

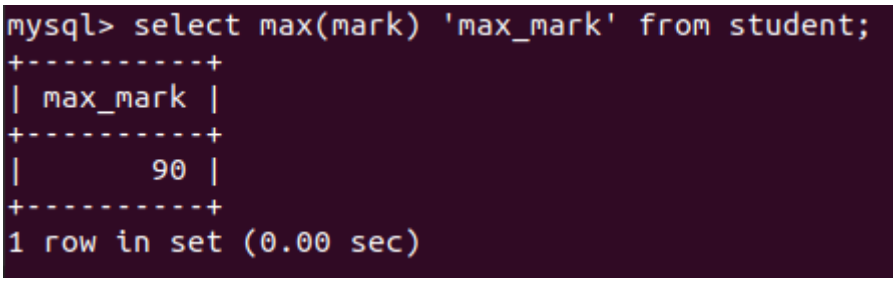
The MySQL MAX() function returns the maximum value in a set of values.

Syntax :

SELECT MAX(expression) FROM Table_name ;

Examples :

select max(mark) 'max_mark' from student;



MIN()

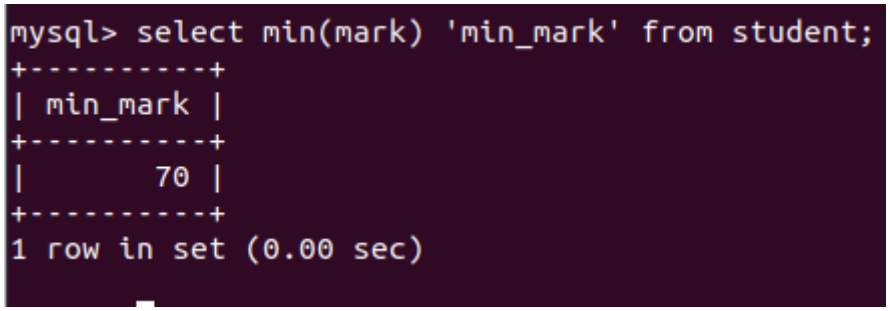
The MySQL MIN() function returns the minimum value in a set of values.

Syntax :

SELECT MIN(expression) FROM Table_name ;

Examples :

select min(mark) 'min_mark' from student;



MySQL String Functions

CONCAT()

Concatenate two or more strings into a single string.

Syntax:

SELECT concat(string1,string 2) FROM table_name;

Example:

select concat(fname,' ',lname) fullname from Student;

```
mysql> select concat(fname,' ',lname) fullname from Student;
+------------+
| fullname   |
+------------+
| raju ram   |
| Saju sam   |
| maya krish |
+------------+
3 rows in set (0.08 sec)
```

INSTR()

The INSTR function returns the position of the first occurrence of a substring in a string. If the substring is not found in the str, the INSTR function returns zero (0).

Syntax:

SELECT INSTR(str,substr);

Example:

select INSTR('anjali dileepkumar','dileep') position ;

```
mysql> select INSTR('anjali dileepkumar','dileep') position ;
+----------+
| position |
+----------+
|        8 |
+----------+
1 row in set (0.00 sec)
```

Returns the length of a string in bytes.

Syntax:

LENGTH(str);

Example:

select length('anjali dileepkumar') length ;

```
mysql> select length('anjali dileepkumar') length ;
+--------+
| length |
+--------+
|     18 |
+--------+
1 row in set (0.00 sec)
```

a)CHAR_LENGTH()

CHAR_LENGTH can be used if you want to return the number of characters(which could be different if you are using a multibyte character set).

Syntax:

CHAR_LENGTH(str);

Example:

select char_length('anjali dileepkumar') length ;

```
mysql> select char_length('anjali dileepkumar') length ;
+--------+
| length |
+--------+
|     18 |
+--------+
1 row in set (0.00 sec)
```

LOWER()

The LOWER() function accepts a string argument and returns the lowercase version of that string.

syntax :

LOWER(str);

Example:

select lower('ANJALI DILEEPKUMAR') lowercase;

```
mysql> select lower('ANJALI DILEEPKUMAR') lowercase;
+--------------------+
| lowercase          |
+--------------------+
| anjali dileepkumar |
+--------------------+
1 row in set (0.02 sec)
```

UPPER()

The UPPER() function returns the uppercase of a specified string argument.

syntax :

UPPER(str);

Example:

select upper('anjali dileepkumar') UpperCase;

```
mysql> select upper('anjali dileepkumar') UpperCase;
+--------------------+
| UpperCase          |
+--------------------+
| ANJALI DILEEPKUMAR |
+--------------------+
1 row in set (0.00 sec)
```

MySQL Date and time functions

CURDATE()

The CURDATE() function returns the current date as a value in the 'YYYY-MM-DD' format if it is used in a string context or YYYMMDD format if it is used in a numeric context.

syntax:

SELECT CURDATE();

Example:

SELECT CURDATE();

```
mysql> SELECT CURDATE();
+------------+
| CURDATE()  |
+------------+
| 2020-10-20 |
+------------+
1 row in set (0.06 sec)
```

DAYNAME()

MySQL DAYNAME function returns the name of a weekday for a specified date.

Syntax :

DAYNAME(date);

Examples:

select dayname('2020-10-20');

```
mysql> select dayname('2020-10-20');
+------------------------+
| dayname('2020-10-20') |
+------------------------+
| Tuesday                |
+------------------------+
1 row in set (0.06 sec)
```

MySQL LAST_DAY()

The LAST_DAY() function takes a DATE or DATETIME value and returns the last day of the month for the input date.

Syntax:

LAST_DAY(date);

Example:

select last_day('2020-10-20');

```
mysql> select last_day('2020-10-20');
+-------------------------+
| last_day('2020-10-20') |
+-------------------------+
| 2020-10-31              |
+-------------------------+
1 row in set (0.00 sec)
```

MySQL Comparison Functions

COALESCE()

It returns the first non-NULL arguments, which is very handy for substitution of NULL.

Syntax: COALESCE(value1,value2,...);

Example:

select rollno,fname,lname,dept,COALESCE(mark,'N/A') from Student;

```
mysql> select rollno,fname,lname,dept,COALESCE(mark,'N/A') from Student;
+--------+-------+---------+------+----------------------+
| rollno | fname | lname   | dept | COALESCE(mark,'N/A') |
+--------+-------+---------+------+----------------------+
|      1 | raju  | ram     | CS   | 90                   |
|      2 | Saju  | sam     | CS   | 90                   |
|      3 | maya  | krish   | CS   | 90                   |
|      4 | riya  | sathyan | CSE  | N/A                  |
+--------+-------+---------+------+----------------------+
4 rows in set (0.00 sec)
```

GREATEST () & LEAST()

It takes n arguments and returns the greatest and least values of the arguments respectively.

Syntax:

GREATEST(value1, value2, ...);

LEAST(value1,value2,...);

example:

SELECT company_id, LEAST(q1, q2, q3, q4) low, GREATEST(q1, q2, q3, q4) high FROM revenues;

```
+------------+--------+--------+
| company_id | low    | high   |
+------------+--------+--------+
|          1 | 100.00 | 130.00 |
|          2 | 250.00 | 310.00 |
|          3 |   NULL |   NULL |
+------------+--------+--------+
3 rows in set (0.00 sec)
```

ISNULL()

It returns 1 if the argument is NULL, otherwise, return zero.

Syntax :

ISNULL(expr)

example:

select * from Student where isnull(mark);

```
mysql> select * from Student where isnull(mark);
+--------+-------+---------+------+------+
| rollno | fname | lname   | dept | mark |
+--------+-------+---------+------+------+
|      4 | riya  | sathyan | CSE  | NULL |
+--------+-------+---------+------+------+
1 row in set (0.00 sec)
```

Math Functions

ABS()

It returns the absolute value of a number

Syntax :

ABS(n);

example:

select abs(-111);

```
mysql> select abs(-111);
+-----------+
| abs(-111) |
+-----------+
|       111 |
+-----------+
1 row in set (0.02 sec)
```

CEIL()

It returns the smallest integer value greater than or equal to the input number (n).

Syntax :

CEIL (numeric_expression);

Example:

select ceil(7.7777);

```
mysql> select abs(-111);
+-----------+
| abs(-111) |
+-----------+
|       111 |
+-----------+
1 row in set (0.02 sec)
```

FLOOR()

It returns the largest integer value not greater than the argument.

Syntax :

FLOOR(expression)

Example:

select floor(7.7777);

```
mysql> select floor(7.7777);
+---------------+
| floor(7.7777) |
+---------------+
|             7 |
+---------------+
1 row in set (0.00 sec)
```

ROUND()

Rounds a number to a specified number of decimal places.

Syntax :

ROUND(n,[d])

examples :

select round(44.444);

```
mysql> select round(44.444);
+---------------+
| round(44.444) |
+---------------+
|            44 |
+---------------+
1 row in set (0.00 sec)
```

TRUNCATE()

Truncates a number to a specified number of decimal places

syntax :

TRUNCATE(X,D)

In this syntax:

X is a literal number or a numeric expression to be truncated.

D is the number of decimal places to truncate to. If D is negative then the TRUNCATE() function causes D digits left of the decimal point of X to become zero. In case D is zero, then the return value has no decimal point.

Both X and D arguments are required.

Example:

select truncate(2.4444,1);

```
mysql> select truncate(2.4444,1);
+--------------------+
| truncate(2.4444,1) |
+--------------------+
|                2.4 |
+--------------------+
1 row in set (0.00 sec)
```