Database Management System - II

Functions

- In MySQL, a function is a stored program that you can pass parameters into and then return a value.
- MySQL comes bundled with a number of built in functions. Built in functions are simply functions come already implemented in the MySQL server. These functions allow us to perform different types of manipulations on the data. The built in functions can be basically categorized into the following most used categories.
  - **Strings functions** – operate on string data types
  - **Numeric functions** – operate on numeric data types
  - **Date functions** – operate on date data types
  - **Aggregate functions** – operate on all of the above data types and produce summarized result sets.

1. Distinct Values

The MySQL DISTINCT clause is used to remove duplicates from the result set. The DISTINCT clause can only be used with SELECT statements.

Syntax:

- SELECT DISTINCT expressions
- FROM tables
- [WHERE conditions];

WHERE conditions: It is optional. It specifies the conditions that must be met for the records to be selected.

Example : Select distinct name from student;

2. String Functions

1) ASCII()
2) CHAR()
3) CONCAT()
1)
4) CONCAT_WS()
5) INSTR()
6) INSERT()
7) LCASE()
8) UCASE()
9) REPLACE()
10) REVERSE()
11) LENGTH()
12) SUBSTR()
13) REPEAT()
14) LEFT()
15) RIGHT()

1. ASCII() : The ASCII() function returns the ASCII value for the specific character.

- **Syntax** : ASCII (character/ column name)
- **Example** : Select ASCII(name) as new_name from student;

2. CHAR() : The CHAR() function returns the character value for the specific Ascii Value

- **Syntax :** char (ascii value)
- **Example :** select char(67) ;

3. CONCAT_WS() : Adds two or more expressions together with a separator.

- **Syntax**:CONCAT_WS(separator,expression1 ,expression2,expression3,.)
- **Example** :

a) SELECT CONCAT_WS("-", "MySQL", "is", "fun!") AS Concatenated String;

b) SELECT CONCAT_WS(" ", Address, PostalCode, City) AS Address FROM Customers;

4. CONCAT() : The CONCAT() function adds two or more expressions together.

Syntax : CONCAT(expression1, expression2, expression3,...)

Example : select concat(name , ' ' , city) as new from student;

5.INSTR(): The INSTR() function returns the position of the first occurrence of a string in another string.

- **Syntax** : INSTR(string1, string2)
- **Example** : SELECT INSTR("W3Schools.com", "COM") AS MatchPosition;

6. INSERT() : Inserts a string within a string at the specified position and for a certain number of characters

- **Syntax** : INSERT(string, position, length, string2)
- **Example** : select insert(rollno , 1 , 2 ,name) as new from student;

7. REPLACE() : The REPLACE() function replaces all occurrences of a substring within a string, with a new substring.

- **Syntax :** REPLACE(string, substring, new_string)
- **Example :** select replace(city , 'bb' , 'cc') from student;

8. REVERSE() : The REVERSE() function reverses a string and returns the result.

- **Syntax :** REVERSE(string)
- **Example :** select reverse(city) from student;

9. LCASE() : The LCASE() function converts a string to lower-case.

- **Syntax :** LCASE(text)
- **Example :** Select Lcase(city) as lowercase_new from student;

10. UCASE() : The UCASE() function converts a string to upper-case.

- **Syntax :** UCASE(text)
- **Example :** Select ucase(city) as uppercase_new from student;

11. LENGTH() : The LENGTH() function returns the length of a string.

- **Syntax :** LENGTH(string)
- **Example :** select length(city) from student

12. SUBSTR() : The SUBSTR() function extracts a substring from a string.

- **Syntax :** SUBSTR(string, start, length)
- **Example :** select substr(city,2,5) from student

13. REPEAT() : Repeats a string as many times as specified

- **Syntax :** REPEAT(string, number)
- **Example :** SELECT REPEAT(CustomerName, 2) FROM Customers;

  OR SELECT REPEAT(“HELLO ”, 4);

14. RIGHT() : Extracts a number of characters from a string (starting from right)

- **Syntax** : RIGHT(string, number_of_chars)
- **Example** : SELECT RIGHT("SQL Tutorial is cool", 4) AS ExtractString;

OR

SELECT RIGHT(CustomerName, 5) AS ExtractString FROM Customers;

15. LEFT() : The LEFT() function extracts a number of characters from a string (starting from left).

- **Syntax :** LEFT(string, number_of_chars)
- **Example :** SELECT LEFT("SQL Tutorial", 3) AS ExtractString;

OR

SELECT LEFT(CustomerName, 5) AS ExtractString FROM Customers;

3. Sorting data functions

- The ORDER BY keyword is used to sort the result-set in ascending or descending order.
- The ORDER BY keyword sorts the records in ascending order by default.
- To sort the records in descending order, use the DESC keyword.

- **Syntax :**
  - SELECT
  - select_list
  - FROM
  - table_name
  - ORDER BY
  - column1 [ASC|DESC],
  - column2 [ASC|DESC],

- **Example :**
  - select * from student order by name desc;
  - select * from student order by name;

4. Math Functions

AVG()
CEIL()
COUNT()
FLOOR()
MAX()
MIN()
MOD()
POW()
ROUND()
SQRT()
SUM()
GREATEST()
LEAST()
PI()
COS()
SIN()

1) 1. AVG() : The AVG() function returns the average value of an expression

Syntax : AVG(expression)

Example : SELECT AVG(Price) AS AveragePrice FROM Products;

2) CEIL() : The CEIL() function returns the smallest integer value that is bigger than or equal to a number.

Syntax : CEIL(number)

Example : SELECT CEIL(25.75); output : 26

3) COUNT() : The COUNT() function returns the number of records returned by a select query. NULL values are not counted.

Syntax : COUNT(expression)

Example : SELECT COUNT(ProductID) AS newProduct FROM Products;

4) FLOOR() : The FLOOR() function returns the largest integer value that is smaller than or equal to a number.

Syntax : FLOOR()

Example : SELECT FLOOR(25.75); - output : 25

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

Syntax : MAX(expression)

Example : SELECT MAX(Price) AS LargestPrice FROM Products;

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

Syntax : MIN(expression)

Example : SELECT MIN(Price) AS LargestPrice FROM Products;

7) MOD() : The MOD() function returns the remainder of a number divided by another number.

Syntax : MOD(x, y)

Example : SELECT MOD(18, 4); or select mod(age,2) as new_age from student;

8) POW() : The POW() function returns the value of a number raised to the power of another number.

Syntax : POW(x, y)

Example : SELECT POW(18, 4); or select pow(age,2) as new_age from student;

9) ROUND() : The ROUND() function rounds a number to a specified number of decimal places.

Syntax : ROUND(number, decimals)

Example : SELECT ProductName, Price, ROUND(Price, 1) AS RoundedPrice FROM Products; or SELECT ROUND(135.375, 2);

10) SQRT() : The SQRT() function returns the square root of a number.

Syntax : SQRT(number)

Example : SELECT SQRT(13);

11) SUM(): The SUM() function calculates the sum of a set of values.

Syntax : SUM(expression)

Example : SELECT SUM(Quantity) AS TotalOrdered FROM OrderDetails;

12) GREATEST() :Returns the greatest value of the list of arguments.

SYNTAX : GREATEST(arg1, arg2, arg3, ...)

Example : SELECT GREATEST(3, 12, 34, 8, 25);

OR

SELECT GREATEST("flipkart.com", "microsoft.com", "apple.com");

13) LEAST(): Returns the smallest value of the list of arguments.

Syntax : LEAST(arg1, arg2, arg3, ...)

Example : SELECT LEAST(3, 12, 34, 8, 25);

OR

SELECT LEAST("Flipkart.com", "microsoft.com", "apple.com");

14) PI() :Returns the value of PI.

SYNTAX : PI()

Example : SELECT PI();

15) COS(): The COS() function returns the cosine of a number.

Syntax : COS(number)

Example : SELECT COS(2);

16) SIN():The SIN() function returns the sine of a number.

Syntax : SIN(number)

Example : SELECT SIN(2);

5. TO_NUMBER Function

- The TO_NUMBER function can convert a number or a character expression representing a number value to a DECIMAL data type.
- The TO_NUMBER function has this syntax:
- TO_NUMBER(char_expr)
- TO_NUMBER(num_expr)

Example : SELECT TO_NUMBER('$100.00') from mytab;

6. Date Functions

1. Date & Time Format

DATE_FORMAT() : The DATE_FORMAT() function formats a date as specified.A list of format specifiers given bellow may be used to format a date. The '%' is required before the format specifier characters.

Syntax : DATE_FORMAT(date, format)

Example : SELECT DATE_FORMAT("2017-06-15", "%Y");

or SELECT DATE_FORMAT("2017-06-15", "%M %d %Y");

or SELECT DATE_FORMAT(BirthDate, "%W %M %e %Y") FROM Employees;

%a	Abbreviated weekday name (Sun to Sat)
%W	Weekday name in full (Sunday to Saturday)
%b	Abbreviated month name (Jan to Dec)
%M	Month name in full (January to December)
%c	Numeric month name (0 to 12)
%T	Time in 24 hour format (hh:mm:ss)
%H	Hour (00 to 23)
%i	Minutes (00 to 59)
%S	Seconds (00 to 59)
%Y	Year as a numeric, 4-digit value
%y	Year as a numeric, 2-digit value
%e	Day of the month as a numeric value (0 to 31)
%U	Week where Sunday is the first day of the week (00 to 53)

2. Sysdate , Now ,Current Date

Sysdate() : This function returns the current date and time as a value in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS format, depending on whether the function is used in a string or numeric context. It returns the date and time in the current time zone.

Syntax : SELECT SYSDATE();

Now() : This function is a synonym for SYSDATE().

Syntax : SELECT NOW();

Current Date() : This function returns the current date, without any time, as a value in 'YYYY-MM-DD' or YYYYMMDD format, depending on whether the function is used in a string or numeric context. It returns the date in the current time zone.

Syntax : SELECT CURDATE();

2. ADDDATE() : The ADDDATE() function adds a time/date interval to a date and then returns the date.

Syntax

ADDDATE(date, INTERVAL value addunit **OR** ADDDATE(date, days)

Example:

Add 15 minutes to a date and return the date:

```
SELECT ADDDATE("2017-06-15 09:34:21", INTERVAL 15 MINUTE);
```

Subtract 2 months to a date and return the date:

```
SELECT ADDDATE("2017-06-15", INTERVAL -2 MONTH);
```

Add 10 days to a date and return the date:

```
SELECT ADDDATE("2017-06-15", INTERVAL 10 DAY);
```

3. DATEDIFF() : The DATEDIFF() function returns the number of days between two date values.

Syntax

```
DATEDIFF(date1, date2)
```

Example

Return the number of days between two date values:

```
SELECT DATEDIFF("2017-01-01", "2016-12-24");
```

4. DAY() : The DAY() function returns the day of the month for a given date (a number from 1 to 31).

Syntax

```
DAY(date)
```

Example

Return the day of the month for a date:

```
SELECT DAY("2017-06-15");
```

Return the day of the month for the current system date:

```
SELECT DAY(CURDATE());
```

5. DAYNAME() : function returns the weekday name for a given date.

   Syntax : DAYNAME(date)

6. The DAYOFWEEK() function returns the weekday index for a given date (a number from 1 to 7).**Note: 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday.**

   Syntax : DAYOFWEEK(date)

7. The DAYOFYEAR() function returns the day of the year for a given date (a number from 1 to 366).

   Syntax : DAYOFYEAR(date)

8. The EXTRACT() function extracts a part from a given date.

Syntax : EXTRACT(part FROM date)

Example :

Extract the month from a date:

```
SELECT EXTRACT(MONTH FROM "2017-06-15");
```

We can extract many things like : **DAY , WEEK ,MONTH , QUARTER ,YEAR , YEAR_MONTH**

9. The TIMESTAMPDIFF() function extracts a part DIFFERENCE from a given dateS.

Syntax : TIMESTAMPDIFF(part, date1 , date2)

Example :

Extract the month from a dates:

```
SELECT TIMESTAMPDIFF(MONTH, "2017-06-15", "2023-06-15");
```

We can extract many PARTS like :

DAY , WEEK , MONTH , QUARTER , YEAR