```
mysql> SELECT FirstName, LastName, (Bonus+Salary) as Total_Sal
    -> FROM faculty
    -> WHERE (Bonus+ Salary)<50000;
+-----------+----------+-----------+
| FirstName | LastName | Total_Sal |
+-----------+----------+-----------+
| Aradhya   | Choudhury |     43800 |
| Animesh   | Yadav     |     49400 |
+-----------+----------+-----------+
2 rows in set (0.00 sec)
```

## OPERATORS IN SQL*PLUS

| Type | Symbol / Keyword | Where to use |
|---|---|---|
| Arithmetic | + , - , * , / | To manipulate numerical column values, WHERE clause |
| Comparison | =, !=, <, <=, >, >=, between, not between, in, not in, like, not like | WHERE clause |
| Logical | and, or, not | WHERE clause, Combining two queries |

**Functions:**

- **Single Row Functions**
- **Group functions**

**Single Row Functions**

- SQL supplies a rich library of in-built functions which can be employed for various tasks. The essential capabilities of a functions can be the **case conversion of strings**, **in-string** or **substring operations, mathematical computations on numeric data**, and **date operations on date type values**.
- 
- SQL Functions optionally take arguments from the user and mandatorily return a value.

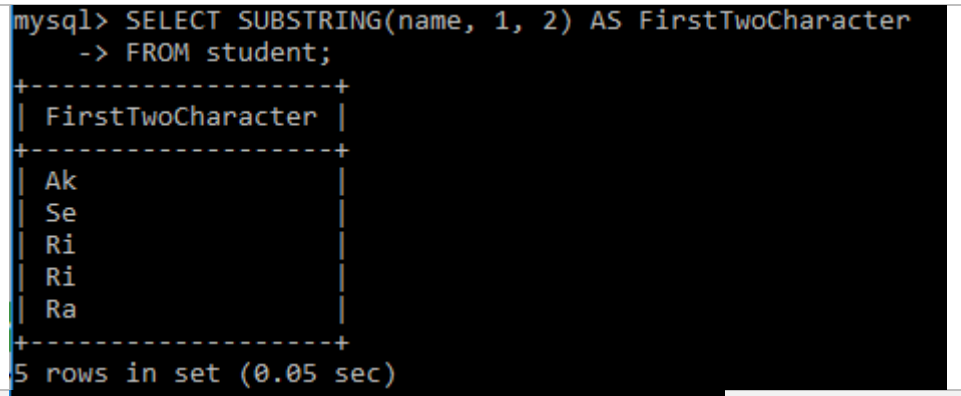
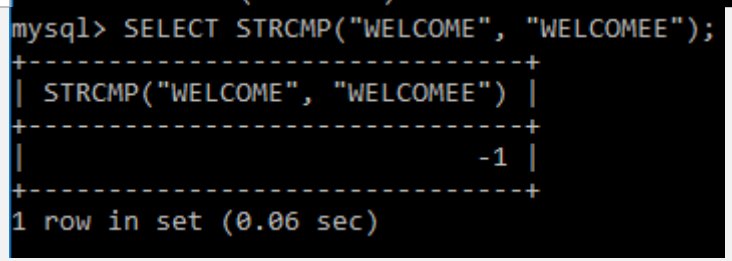
On a broader category, there are two types of functions :-

**Single Row functions** - Single row functions are the one who work on single row and return one output per row. For example, length and case conversion functions are single row functions.

**Multiple Row functions** - Multiple row functions work upon group of rows and return one result for the complete set of rows. They are also known as Group Functions.

**Single row functions are used in SELECT command and included in WHERE clause, Order by clause**

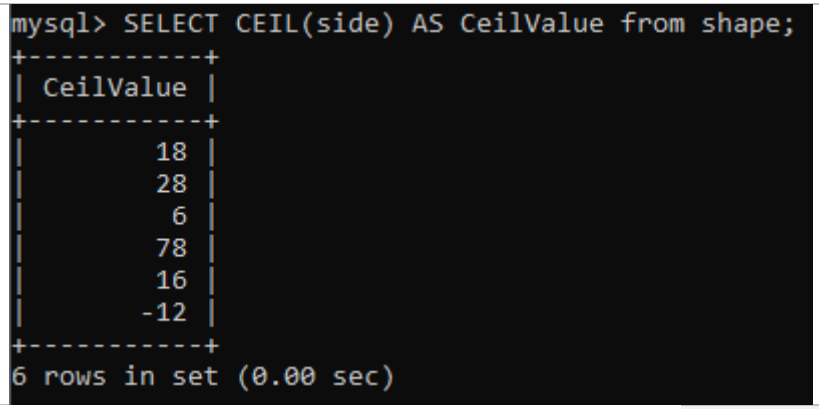
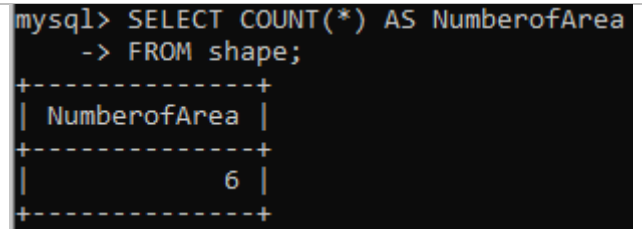
| | Single Row Functions: String Functions | | |
|---|---|---|---|
| **35** | UPPER / UCASE | SELECT UPPER(columnname)<br>FROM tablename<br>WHERE condition; | mysql> SELECT UPPER(name), LOWER(course)<br>-> FROM student<br>-> WHERE student_id = 6;<br>+-------------+---------------+<br>\| UPPER(name) \| LOWER(course) \|<br>+-------------+---------------+<br>\| RAKESH \| cse \|<br>+-------------+---------------+<br>1 row in set (0.23 sec) |
| | LOWER | SELECT LOWER(columnname)<br>FROM tablename<br>WHERE condition; | |
| | TRIM()- Removes leading and trailing spaces from a string. | SELECT TRIM("string to trim") AS TrimmedString; | mysql> SELECT TRIM(" WELCOME ") AS TrimmedString;<br>+---------------+<br>\| TrimmedString \|<br>+---------------+<br>\| WELCOME \|<br>+---------------+<br>1 row in set (0.08 sec) |

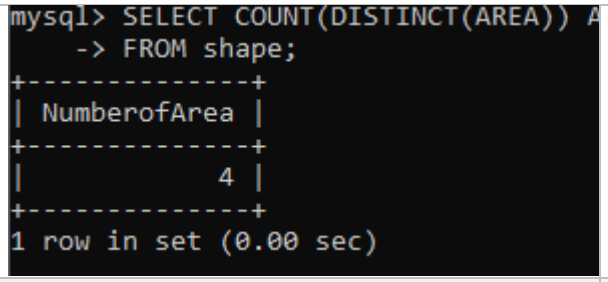
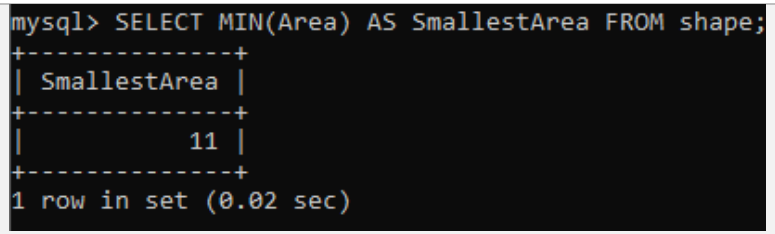
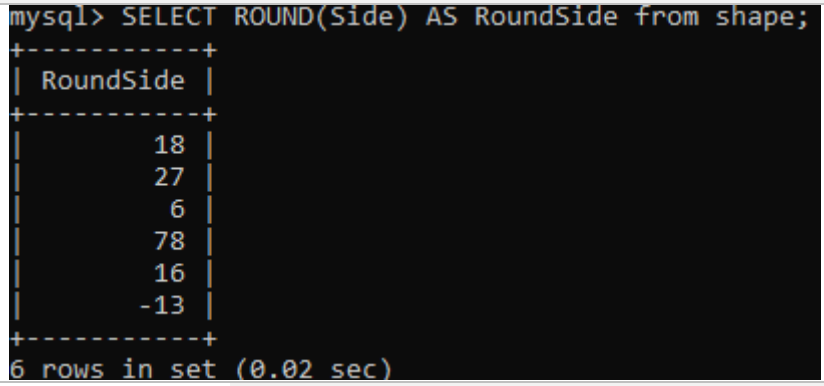
| | | | |
|---|---|---|---|
| | SUBSTRING() | SUBSTRING(*string*, *start*, *length*)<br>Start - +ve or –ve.<br>Length – number of character from start |  |
| | STRCMP() | `STRCMP(string1, string2)`<br>• If *string1* = *string2*, this function returns 0<br>• If *string1* < *string2*, this function returns -1<br>• If *string1* > *string2*, this function returns 1 |  |

| Function Name | Description |
|---|---|
| ASCII | Returns the ASCII value for the specific character |
| CHAR_LENGTH | Returns the length of a string (in characters) |
| CHARACTER_LENGTH | Returns the length of a string (in characters) |
| CONCAT | Adds two or more expressions together |
| CONCAT_WS | Adds two or more expressions together with a separator |
| FIELD | Returns the index position of a value in a list of values |
| FIND_IN_SET | Returns the position of a string within a list of strings |
| FORMAT | Formats a number to a format like "#,###,###.##", rounded to a specified number of decimal places |
| INSERT | Inserts a string within a string at the specified position and for a certain number of characters |
| INSTR | Returns the position of the first occurrence of a string in another string |
| LCASE | Converts a string to lower-case |
| LEFT | Extracts a number of characters from a string (starting from left) |
| LENGTH | Returns the length of a string (in bytes) |
| LOCATE | Returns the position of the first occurrence of a substring in a string |
| LOWER | Converts a string to lower-case |
| LPAD | Left-pads a string with another string, to a certain length |
| LTRIM | Removes leading spaces from a string |
| MID | Extracts a substring from a string (starting at any position) |
| POSITION | Returns the position of the first occurrence of a substring in a string |
| REPEAT | Repeats a string as many times as specified |
| REPLACE | Replaces all occurrences of a substring within a string, with a new substring |
| REVERSE | Reverses a string and returns the result |
| RIGHT | Extracts a number of characters from a string (starting from right) |
| RPAD | Right-pads a string with another string, to a certain length |
| RTRIM | Removes trailing spaces from a string |
| SPACE | Returns a string of the specified number of space characters |
| STRCMP | Compares two strings |
| SUBSTR | Extracts a substring from a string (starting at any position) |
| SUBSTRING | Extracts a substring from a string (starting at any position) |
| SUBSTRING_INDEX | Returns a substring of a string before a specified number of delimiter occurs |
| TRIM | Removes leading and trailing spaces from a string |
| UCASE | Converts a string to upper-case |
| UPPER | Converts a string to upper-case |

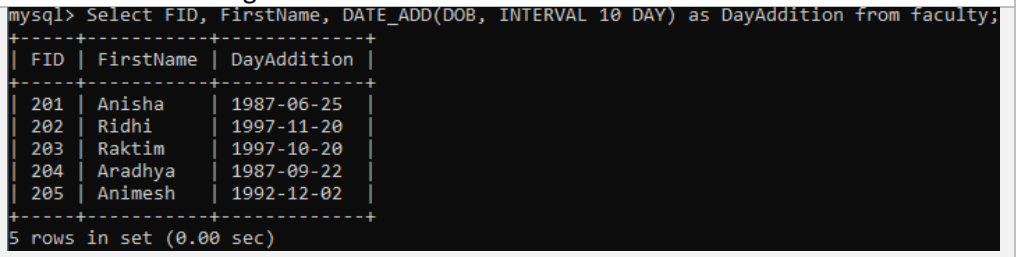
| 36 | Multiple Row Function : Numeric Functions | |
|---|---|---|
| | ABS | Returns the absolute value of a number |
| | ACOS | Returns the arc cosine of a number |
| | ASIN | Returns the arc sine of a number |
| | ATAN | Returns the arc tangent of one or two numbers |
| | ATAN2 | Returns the arc tangent of two numbers |
| | AVG | Returns the average value of an expression |
| | CEIL | Returns the smallest integer value that is >= to a number |
| | CEILING | Returns the smallest integer value that is >= to a number |
| | COS | Returns the cosine of a number |
| | COT | Returns the cotangent of a number |
| | COUNT | Returns the number of records returned by a select query |
| | DEGREES | Converts a value in radians to degrees |
| | DIV | Used for integer division |
| | EXP | Returns e raised to the power of a specified number |
| | FLOOR | Returns the largest integer value that is <= to a number |
| | LN | Returns the natural logarithm of a number |
| | LOG | Returns the natural logarithm of a number, or the logarithm of a number to a specified base |
| | LOG10 | Returns the natural logarithm of a number to base 10 |
| | LOG2 | Returns the natural logarithm of a number to base 2 |
| | MAX | Returns the maximum value in a set of values |
| | MIN | Returns the minimum value in a set of values |
| | MOD | Returns the remainder of a number divided by another number |
| | PI | Returns the value of PI |
| | POW | Returns the value of a number raised to the power of another number |
| | POWER | Returns the value of a number raised to the power of another number |
| | RADIANS | Converts a degree value into radians |
| | RAND | Returns a random number |
| | ROUND | Rounds a number to a specified number of decimal places |
| | SIGN | Returns the sign of a number |
| | SIN | Returns the sine of a number |
| | SQRT | Returns the square root of a number |
| | SUM | Calculates the sum of a set of values |
| | TAN | Returns the tangent of a number |
| | TRUNCATE | Truncates a number to the specified number of decimal places |

| | | | |
|---|---|---|---|
| | | Create table shape(<br>size int NOT NULL AUTO_INCREMENT,<br>area int,<br>perimeter int,<br>side decimal(4,2),<br>PRIMARY KEY(size)<br>); | INSERT into shape(area,perimeter, side) values (25,34,17.82);<br>INSERT into shape(area,perimeter, side) values (25,34,27.33);<br>INSERT into shape(area,perimeter, side) values (11,45,5.75);<br>INSERT into shape(area,perimeter, side) values (23,59,77.75);<br>INSERT into shape(area,perimeter, side r) values (16,20,15.75);<br>INSERT into shape(area,perimeter, side r) values (25,34,-12.75); |
| | mysql> select * from shape;<br>+------+------+-----------+--------+<br>\| size \| area \| perimeter \| side \|<br>+------+------+-----------+--------+<br>\| 1 \| 25 \| 34 \| 17.82 \|<br>\| 2 \| 25 \| 34 \| 27.33 \|<br>\| 3 \| 11 \| 45 \| 5.75 \|<br>\| 4 \| 23 \| 59 \| 77.75 \|<br>\| 5 \| 16 \| 20 \| 15.75 \|<br>\| 6 \| 25 \| 34 \| -12.75 \|<br>+------+------+-----------+--------+<br>6 rows in set (0.00 sec) | | |
| **37** | ABS() | SELECT ABS(Side) AS Absolute FROM shape; | mysql> SELECT ABS(Side) AS Absolute FROM shape;<br>+----------+<br>\| Absolute \|<br>+----------+<br>\| 17.82 \|<br>\| 27.33 \|<br>\| 5.75 \|<br>\| 77.75 \|<br>\| 15.75 \|<br>\| 12.75 \|<br>+----------+<br>6 rows in set (0.03 sec) |

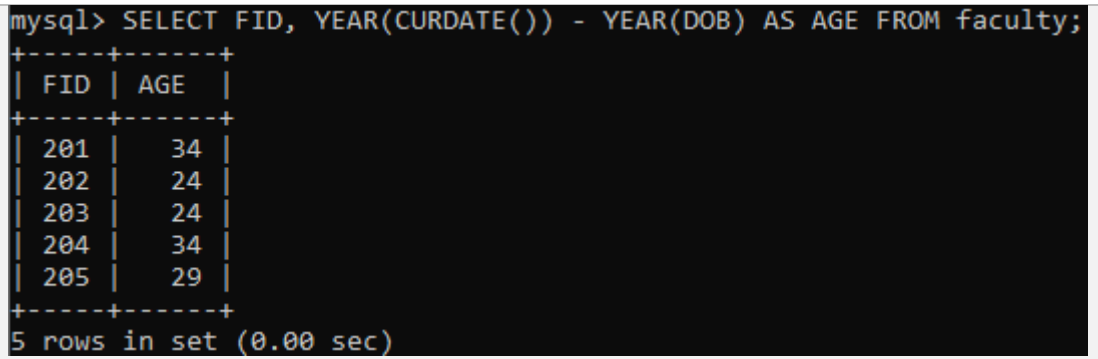
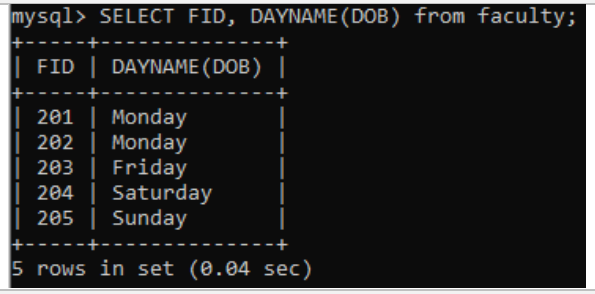
| | | | |
|---|---|---|---|
| **38** | CEILING() | SELECT CEIL(side) AS CeilValue from shape; |  |
| **39** | COUNT() | • The **COUNT** function is used to count rows or values of a column that do not contain a NULL value.<br>• The COUNT function returns a numeric value.<br>• **Syntax:**<br>SELECT **COUNT (*) / (DISTINCT / ALL (COLUMN NAME)**)<br>FROM TBLNAME;<br>• The **DISTINCT** command cannot be used with **COUNT(*).**<br>• Because Count(*) will count the column with the duplicate values. | |
| | | |  <br>Select count(ALL course) |

| | | | |
|---|---|---|---|
| | |  | |
| **40** | MIN() | SELECT MIN(Area) AS SmallestArea<br>from shape; |  |
| **41** | ROUND() | SELECT ROUND(Side) AS RoundSide from shape; |  |
| **42** | **Date Functions** | | |
| | ADDTIME Adds a time interval to a time/datetime and then returns the time/datetime<br>CURDATE Returns the current date<br>CURRENT_DATE Returns the current date<br>CURRENT_TIME Returns the current time<br>CURRENT_TIMESTAMP Returns the current date and time<br>CURTIME Returns the current time<br>DATE Extracts the date part from a datetime expression<br>DATEDIFF Returns the number of days between two date values | | |

| | | |
|---|---|---|
| | DATE_ADD | Adds a time/date interval to a date and then returns the date |
| | DATE_FORMAT | Formats a date |
| | DATE_SUB | Subtracts a time/date interval from a date and then returns the date |
| | DAY | Returns the day of the month for a given date |
| | DAYNAME | Returns the weekday name for a given date |
| | DAYOFMONTH | Returns the day of the month for a given date |
| | DAYOFWEEK | Returns the weekday index for a given date |
| | DAYOFYEAR | Returns the day of the year for a given date |
| | EXTRACT | Extracts a part from a given date |
| | FROM_DAYS | Returns a date from a numeric datevalue |
| | HOUR | Returns the hour part for a given date |
| | LAST_DAY | Extracts the last day of the month for a given date |
| | LOCALTIME | Returns the current date and time |
| | LOCALTIMESTAMP | Returns the current date and time |
| | MAKEDATE | Creates and returns a date based on a year and a number of days value |
| | MAKETIME | Creates and returns a time based on an hour, minute, and second value |
| | MICROSECOND | Returns the microsecond part of a time/datetime |
| | MINUTE | Returns the minute part of a time/datetime |
| | MONTH | Returns the month part for a given date |
| | MONTHNAME | Returns the name of the month for a given date |
| | NOW | Returns the current date and time |
| | PERIOD_ADD | Adds a specified number of months to a period |
| | PERIOD_DIFF | Returns the difference between two periods |
| | QUARTER | Returns the quarter of the year for a given date value |
| | SECOND | Returns the seconds part of a time/datetime |
| | SEC_TO_TIME | Returns a time value based on the specified seconds |
| | STR_TO_DATE | Returns a date based on a string and a format |
| | SUBDATE | Subtracts a time/date interval from a date and then returns the date |
| | SUBTIME | Subtracts a time interval from a datetime and then returns the time/datetime |
| | SYSDATE | Returns the current date and time |
| | TIME | Extracts the time part from a given time/datetime |
| | TIME_FORMAT | Formats a time by a specified format |
| | TIME_TO_SEC | Converts a time value into seconds |
| | TIMEDIFF | Returns the difference between two time/datetime expressions |

| | | | |
|---|---|---|---|
| | TIMESTAMP | Returns a datetime value based on a date or datetime value | |
| | TO_DAYS | Returns the number of days between a date and date "0000-00-00" | |
| | WEEK | Returns the week number for a given date | |
| | WEEKDAY | Returns the weekday number for a given date | |
| | WEEKOFYEAR | Returns the week number for a given date | |
| | YEAR | Returns the year part for a given date | |
| | YEARWEEK | Returns the year and week number for a given date | |
| **43** | Date_Add() | date Required. The date to be modified<br>days Required. The number of days to add to date<br>value Required. The value of the time/date interval to add. Both positive and negative values are allowed<br>addunit Required.<br><br>The type of interval to add. Can be one of the following values: MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR , SECOND_MICROSECOND, MINUTE_MICROSECOND, MINUTE_SECOND, HOUR_MICROSECOND<br>HOUR_SECOND, HOUR_MINUTE, DAY_MICROSECOND, DAY_SECOND, DAY_MINUTE, DAY_HOUR, YEAR_MONTH); |  |
| **44** | DATE_FORMAT() | Fpormat Can be one or a combination of the following values:<br>Format Description<br>%a Abbreviated weekday name (Sun to Sat)<br>%b Abbreviated month name (Jan to Dec)<br>%c Numeric month name (0 to 12)<br>%D Day of the month as a numeric value, followed by suffix (1st, 2nd, 3rd, ...)<br>%d Day of the month as a numeric value (01 to 31)<br>%e Day of the month as a numeric value (0 to 31)<br>%f Microseconds (000000 to 999999)<br>%H Hour (00 to 23)<br>%h Hour (00 to 12) | |

| | | |
|---|---|---|
| | | %I Hour (00 to 12)<br>%i Minutes (00 to 59)<br>%j Day of the year (001 to 366)<br>%k Hour (0 to 23)<br>%l Hour (1 to 12)<br>%M Month name in full (January to December)<br>%m Month name as a numeric value (00 to 12)<br>%p AM or PM<br>%r Time in 12 hour AM or PM format (hh:mm:ss AM/PM)<br>%S Seconds (00 to 59)<br>%s Seconds (00 to 59)<br>%T Time in 24 hour format (hh:mm:ss)<br>%U Week where Sunday is the first day of the week (00 to 53)<br>%u Week where Monday is the first day of the week (00 to 53)<br>%V Week where Sunday is the first day of the week (01 to 53). Used with %X<br>%v Week where Monday is the first day of the week (01 to 53). Used with %x<br>%W Weekday name in full (Sunday to Saturday)<br>%w Day of the week where Sunday=0 and Saturday=6<br>%X Year for the week where Sunday is the first day of the week. Used with %V<br>%x Year for the week where Monday is the first day of the week. Used with %v<br>%Y Year as a numeric, 4-digit value<br>%y Year as a numeric, 2-digit value<br><br>`mysql> SELECT DOB, DATE_FORMAT(DOB, "%j %a %M %d %Y") from faculty;`<br>`+------------+-----------------------------------+`<br>`\| DOB \| DATE_FORMAT(DOB, "%j %a %M %d %Y") \|`<br>`+------------+-----------------------------------+`<br>`\| 1987-06-15 \| 166 Mon June 15 1987 \|`<br>`\| 1997-11-10 \| 314 Mon November 10 1997 \|`<br>`\| 1997-10-10 \| 283 Fri October 10 1997 \|`<br>`\| 1987-09-12 \| 255 Sat September 12 1987 \|`<br>`\| 1992-11-22 \| 327 Sun November 22 1992 \|`<br>`+------------+-----------------------------------+`<br>`5 rows in set (0.00 sec)` |

| | | |
|---|---|---|
| **45** | CURDATE() |  |
| | DAYNAME() |  |
| | | |