**LAB 03 SQL Functions**

**Difference Between IN , Between**

**Clear mysql commandline by \! cls;**

# LIMIT

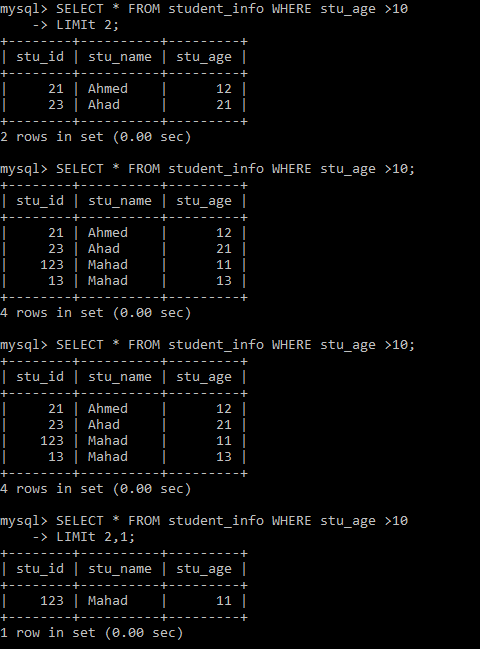
The **LIMIT** clause is used in the **SELECT** statement to force the number of records in a result set. The **LIMIT** clause accepts two arguments: **offset** and/or **count**. The values of both arguments must be zero or positive integer.

* **Offset** specifies the offset of the first row to return. The offset of the first row is 0, not 1.
* **Count** specifies the maximum number of rows to return.
* **SELECT** col\_names(s) **FROM** tb\_name

**LIMIT** offset, count;

* **SELECT** col\_names(s) **FROM** tb\_name

**LIMIT** count;

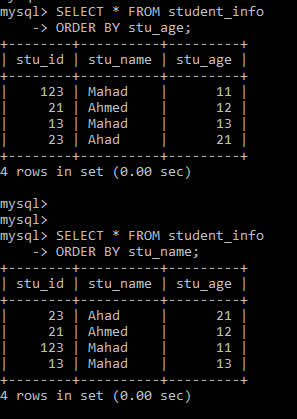


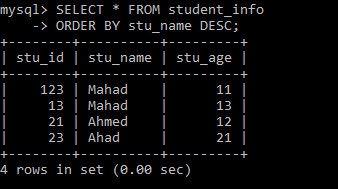
# ORDER BY

The **ORDER BY** clause in **SELECT** statement is used to sort the result set in ascending or descending order. The **ORDER BY** clause sorts the records in ascending order by default.

* **SELECT** col\_names(s) **FROM** tb\_name

**ORDER BY** col\_name(s) **ASC**/**DESC**;





# OBJECTIVE(S)

* Learn column placement
* Learn string functions
* Learn math functions
* Learn SQL aggregate functions
* Learn about grouping rows

# NEW COLUMN PLACEMENT

If we wish to alter the table by adding new columns, we can decide where to place them in the table. The following keywords allow us to decide their placement.

## FIRST

The new column is added as the first column of the table.

* **ALTER TABLE** tb\_name **ADD** col\_name datatype(size) **FIRST**;

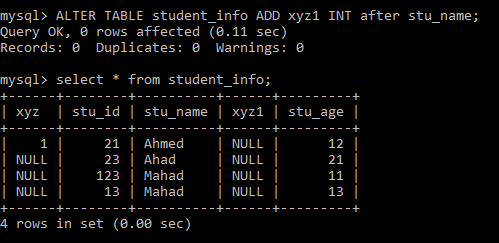
## AFTER

This allows us to place the column after another existing column in the table.

* **ALTER TABLE** tb\_name **ADD** new\_col\_name datatype(size) **AFTER** col\_name;

|  |
| --- |
| TASK  * Add a new column ***sno*** at the beginning of the students table. * Add a new column ***guardian\_name*** before ***semester***. |

# 



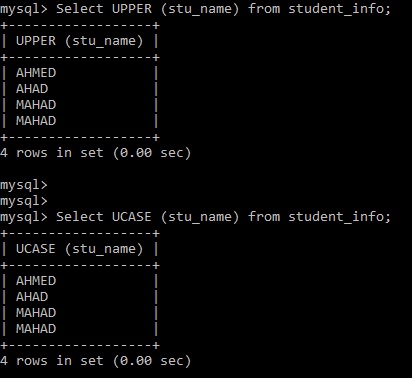
# STRING FUNCTIONS

MySQL allows us to manipulate string data using several functions some of which are given below.

## UPPER

The **UPPER** or **UCASE** function converts all the characters of a string into uppercase.

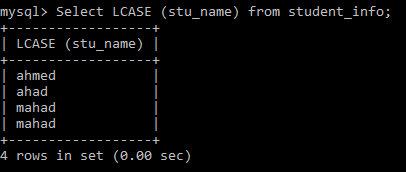
* SELECT **UPPER**(col\_name) **FROM** tb\_name;
* SELECT **UCASE**(col\_name) **FROM** tb\_name;



## LOWER

The **LOWER** or **LCASE** function converts all the characters of a string into lowercase.

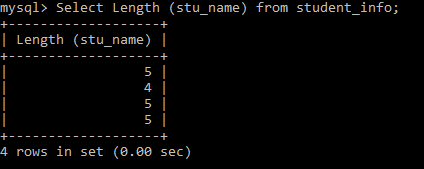
* SELECT **LOWER**(col\_name) **FROM** tb\_name;
* SELECT **LCASE**(col\_name) **FROM** tb\_name;



## LENGTH

The **LENGTH** function returns the length of the data of a particular column.

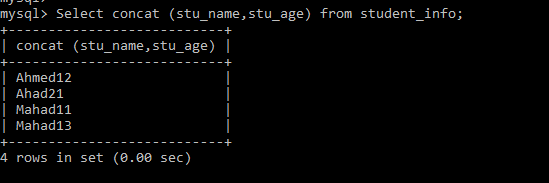
* **SELECT LENGTH**(col\_name) **FROM** tb\_name;



## CONCAT

The **CONCAT** function adds (concatenates) two or more strings. It returns NULL if any field contains NULL.

* **SELECT CONCAT**(col1\_name, col2\_name) **FROM** tb\_name;
* **SELECT CONCAT**(col1\_name, “sample string”) **FROM** tb\_name;



## SUBSTR

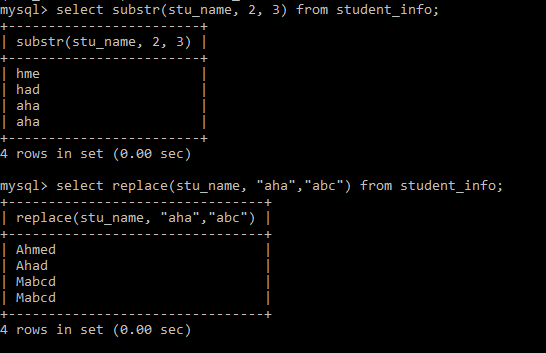
The **SUBSTR** function returns a substring of length *len* starting at position *pos* in the data.

* **SELECT SUBSTR**(col\_name, pos, len) **FROM** tb\_name;

## REPLACE

The **REPLACE** function replaces all instances of a substring within a string.

* **SELECT REPLACE**(col\_name, “substring”, “replacement”) **FROM** tb\_name;



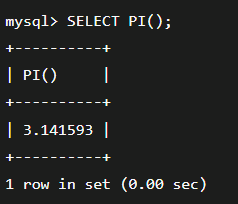
|  |
| --- |
| TASK  * Display the names of students in upper case. * Display the names of students in the following format: **SMITH, J.** * Display the names and departments of students in the following format: **John Smith, CS**. * Display the names and departments of all the students. Replace all instances of **CS** with **CSE**. Label appropriately. |

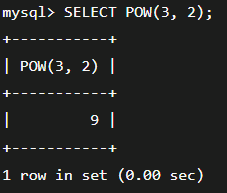
# MATH FUNCTIONS

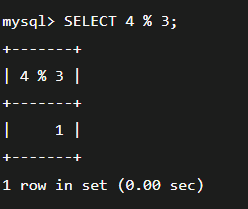
MySQL also allows us to perform certain mathematical operations on the data of a table. Some functions for these operations are given below. The general syntax of these functions (unless otherwise specified) is:

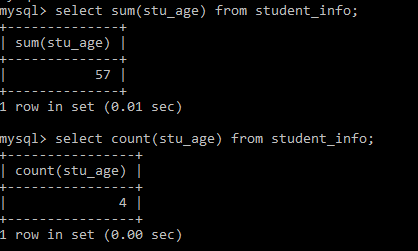
* **SELECT FUNCTION\_NAME**(col\_name) **FROM** tb\_name;

| FUNCTION NAME | DESCRIPTION |
| --- | --- |
| ABS | Returns the absolute value of a number. |
| CEIL | Synonym: **CEILING**  Returns the smallest integer value which is greater than the number provided as the argument. |
| FLOOR | Returns the smallest integer which is lower than the number provided as the argument. |
| DEGREES | Converts radians into degrees. |
| RADIANS | Converts degrees into radians. |
| EXP | Returns the value of , where is the argument. E.g. **EXP**() |
| LN | Returns the natural logarithm of the argument. |
| LOG2 | Returns the logarithm of the argument to the base 2. |
| LOG10 | Returns the logarithm of the argument to the base 10. |
| POW | Synonym: **POWER**(M,N)  **POW**(M,N) returns the value of . |
| ROUND | **ROUND**(N,D) rounds the number N to D decimal places. |
| SQRT | Returns the square root of the argument. |
| TRUNCATE | **TRUNCATE**(N,D) trims the number N to D decimal places. |
| TRIGONOMETRIC FUNCTIONS | These allow us to perform trigonometric operations on the data of a table. Some common trigonometric functions are **COS()**, **SIN()**, **TAN()**, **ACOS()**, **ASIN()**, and **ATAN()**. |









|  |
| --- |
| TASK  * Find the absolute value of **-273.15**. * Convert into degrees. * Convert **120o** into radians. * Calculate . Round off to four decimal places. * Truncate the value of to four decimal places. * Calculate the distance between the following coordinates: **(-1, -3)** and **(2, -7)**. * Display the range of percentage for each student. E.g. 95 – 96. Label appropriately. |

# DATE/TIME FUNCTIONS

MySQL provides several functions for date/time calculations some of which are listed below. The general syntax of these functions (unless otherwise specified) is:

* **SELECT FUNCTION\_NAME**(col\_name) **FROM** tb\_name;

| FUNCTION NAME | DESCRIPTION |
| --- | --- |
| YEAR | Extracts the year (YYYY) from a date datatype field. |
| MONTH | Extracts the month (MM) from a date datatype field. |
| DAYOFMONTH | Extracts the day (DD) from a date datatype field. |
| CURDATE | Gives the current date in YYYY-MM-DD format. |
| TIMESTAMPDIFF | **TIMESTAMPDIFF**(unit, date1, date2) can be used to calculate the time elapsed between date1 and date2 in the specified units. Unit is used to indicate how we want the result to be expressed. It can take the value YEAR, MONTH, or DAY each of which will return the result in the appropriate unit. |

|  |
| --- |
| TASK  * Display the number of months until your next birthday. Label appropriately. * Display the year of birth and age of each student. Label appropriately. |

# 

