

# Database

It is defined as a collection of interrelated data stored together to serve multiple applications.

## MySQL Elements

MySQL has certain elements that play an important role in querying a database.

### Literals

Literals refer to a fixed data value

```
17 #It is a numeric literal
"Harry" #It is a text literal
12.5 #It is a real literal
```

### Data Types

Data types are means to identify the type of data.

```
#Numeric
```

```
INT -- Integer data type
TINYINT
SMALLINT
MEDIUMINT
BIGINT
```

```
FLOAT(M,D) -- Floating point data type
DOUBLE(M,D) -- Double data type also stores decimal values
DECIMAL(M,D) -- Decimal data type
```

```
#Data and Time
```

```
DATE -- Date data type (YYYY-MM-DD)
DATETIME -- It's a date and time combination (YYYY-MM-DD HH:MM:SS)
TIME -- It stores time (HH:MM:SS)
```

```
#String/Text
```

```
CHAR(M) -- Character data type
```

```
VARCHAR(M) -- Variable character data type
```

```
BLOB or TEXT
```

## NULL Values

If a column has no value, then it is said to be NULL

## Comments

A comment is a text that is not executed.

```
/* This is a multi-line  
comment in MySQL */
```

```
# It is a single-line comment
```

```
-- It is also a single-line comment
```

## MySQL Simple Calculations

You can perform simple calculations in MySQL, just by using the Select command, there's no need to select any particular database to perform these commands.

### Addition

It will add two numbers

```
Select 5+8;
```

### Subtraction

It will subtract the second number from first

```
Select 15-5;
```

### Multiplication

It will give the product of supplied numbers

```
Select 5*5;
```

## Division

It will divide the number

```
Select 24/4;
```

```
-- SQL is not a case-sensitive language
```

## Accessing Database

These commands allow one to check all the databases and tables

### Show command

It will show all the databases in the system

```
Show databases;
```

It will show all the tables in a selected database

```
show tables;
```

### Use command

It will start using the specified database i.e. now you can create tables in the selected database

```
use database_name;
```

## Creating tables

These commands allow you to create the table in MySQL

### Create table command

This query is used to create a table in the selected database

```
Create table <table-name>
(<column_name> <data_type>,
<column_name> <data_type>,
<column_name> <data_type>);
```

## Insert command

It will add data into the selected table

```
Insert into <table_name> [<column-list>]
Values (<value1>,<value2>...);
```

## Inserting NULL values

This query will add NULL value in the col3 of the selected table

```
Inset into <table-name> (col1, col2,col3)
Values (val1,val2,NULL);
```

## Inserting Dates

It will add the following data into the selected column of the table

```
Insert into <table_name> (<col_name>)
Values ('2021-12-10');
```

## Select Command

A select query is used to fetch the data from the database

### Selecting All Data

It will retrieve all the data of the selected table

```
Select * From <table_name>;
```

### Selecting Particular Rows

It will retrieve all the data of the row that will satisfy the condition

```
Select * from <table_name>  
Where <condition_to_satisfy>;
```

## Selecting Particular Columns

It will retrieve data of selected columns that will satisfy the condition

```
Select column1, column2 from <table_name>  
Where <condition_to_satisfy>;
```

## DISTINCT Keyword

It will retrieve only distinct data i.e. duplicate data rows will get eliminated

```
Select DISTINCT <column_name> from <table_name>;
```

## ALL Keyword

It will retrieve all the data of the selected column

```
Select ALL <column_name> from <table_name>;
```

## Column Aliases

It is used to give a temporary name to a table or a column in a table for the purpose of a particular query

```
Select <column1>,<column2> AS <new_name>  
From <table_name>;
```

## Condition Based on a Range

It will only retrieve data of those columns whose values will fall between value1 and value2 (both inclusive)

```
Select <col1>, <col2>  
From <table_name>  
Where <value1> Between <value2>;
```

## Condition Based on a List

```
Select * from <table_name>
Where <column_name> IN (<val1>,<val2>,<val3>);
```

```
"Select * from <table_name>
Where <column_name> NOT IN (<val1>,<val2>,<val3>);"
```

## Condition Based on Pattern Match

```
Select <col1>,<col2>
From <table_name>
Where <column> LIKE 'Ha%';
```

```
Select <col1>,<col2>
From <table_name>
Where <column> LIKE 'Ha__y%';
```

## Searching NULL

It returns data that contains a NULL value in them

```
Select <column1>, <column2>
From <table_name> Where <Val> IS NULL;
```

## SQL Constraints

SQL constraints are the rules or checks enforced on the data columns of a table

### NOT NULL

It will create a table with NOT NULL constraint to its first column

```
Create table <table_name>
( <col1> <data_type> NOT NULL,
<col2> <data_type>,
<col3> <data_type>);
```

### DEFAULT

DEFAULT constraint provides a default value to a column

```
Create table <table_name>
( <col1> <data_type> DEFAULT 50,
  <col2> <data_type>,
  <col3> <data_type>);
```

## UNIQUE

UNIQUE constraint ensures that all values in the column are different

```
Create table <table_name>
( <col1> <data_type> UNIQUE,
  <col2> <data_type>,
  <col3> <data_type>);
```

## CHECK

CHECK constraint ensures that all values in a column satisfy certain conditions

```
Create table <table_name>
( <col1> <data_type> CHECK (condition),
  <col2> <data_type>,
  <col3> <data_type>);
```

## Primary Key

Primary key is used to uniquely identify each row in a table

```
Create table <table_name>
( <col1> <data_type> Primary Key,
  <col2> <data_type>,
  <col3> <data_type>);
```

## Foreign Key

```
CREATE TABLE Orders (
  OrderID int NOT NULL,
  OrderNumber int NOT NULL,
  PersonID int,
```

```
PRIMARY KEY (OrderID),  
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);
```

## Viewing Table Structure

### Desc or Describe command

It allows you to see the table structure

```
Desc <table_name>;
```

## Modifying Data

### Update Command

It will update the values of selected columns

```
Update <table_name>  
SET <col1> = <new_value>, <col2> = <new_value>  
Where <condition>;
```

## Deleting Data

### Delete Command

It will delete the entire row that will satisfy the condition

```
Delete From <table_name>  
Where <condition>;
```

## Ordering Records

Order by clause is used to sort the data in ascending or descending order of specified column

### order by clause

It will return records in the ascending order of the specified column name's data

```
Select * from <table_name> order by <column_name>;
```

It will return records in the descending order of the specified column name's data

```
Select * from <table_name> order by <column_name> DESC;
```

## Ordering data on multiple columns

It will return records in the ascending order of column1 and descending order of column2

```
Select * From <table_name> order by <column1> ASC, <column2> DESC;
```

## Grouping Result

It is used to arrange identical data into groups so that aggregate functions can work on them

### Group by clause

It allows you to group two or more columns and then you can perform aggregate function on them

```
Select <column>, Count(*) from <table_name> group by <column>;
```

### Having clause

Having clause is used to put conditions on groups

```
Select avg(<column>), sum(<column>) from <table_name> group by <column_name>
```

## Altering Table

These commands allow you to change the structure of the table

### To Add New Column

It will add a new column in your table

```
Alter Table <table_name>  
Add <new_column>;
```

### To Modify Old Column

It will update the data type or size of old column

```
Alter Table <table_name>  
Modify <old_column_name> [<new_data_type><size>];
```

## To Change Name of Column

It will change the name of the old column in the table

```
Alter Table Change <old_column_name> <new_column_name><data_type>;
```

## Dropping Table

### DROP command

It will delete the complete table from the database

```
Drop table <table_name>;
```

## MySQL Functions:

There are many functions in MySQL that perform some task or operation and return a single value

### Text/String Functions

Text function work on strings

#### Char Function

It returns the character for each integer passed

```
Select Char(72,97,114,114,121);
```

#### Concat Function

It concatenates two strings

```
Select Concat("Harry","Bhai");
```

## Lower/Lcase

It converts a string into lowercase

```
Select Lower("Harry Bhai");
```

## Upper/Ucase

It converts a string into uppercase

```
Select Upper("CodeWithHarry");
```

## Substr

It extracts a substring from a given string

```
Select Substr(string,m,n);
```

## Trim

It removes leading and trailing spaces from a given string

```
Select Trim(leading ' ' FROM ' Harry Bhai');
```

## Instr

It searches for given second string into the given first string

```
Select Instr(String1,String2);
```

## Length

It returns the length of given string in bytes

```
Select Length(String)
```

## Numeric Functions

Numeric function works on numerical data and returns a single output

## MOD

It returns modulus of two numbers

```
Select MOD(11,4);
```

## Power

It returns the number m raised to the nth power

```
Select Power(m,n);
```

## Round

It returns a number rounded off number

```
Select Round(15.193,1);
```

## Sqrt

It returns the square root of a given number

```
Select Sqrt(144);
```

## Truncate

It returns a number with some digits truncated

```
Select Truncate(15.75,1);
```

## Date/Time Functions

These are used to fetch the current date and time and allow you to perform several operations on them

### Curdate Function

It returns the current date

```
Select Curdate();
```

## Date Function

It extracts the date part of the expression

```
Select Date('2021-12-10 12:00:00');
```

## Month Function

It returns the month from the date passed

```
Select Month(date);
```

## Day Function

It returns the day part of a date

```
Select Day(date);
```

## Year Function

It returns the year part of a date

```
Select Year(date);
```

## Now Function

It returns the current date and time

```
Select now();
```

## Sysdate Function

It returns the time at which function executes

```
Select sysdate();
```

## Aggregate Functions

Aggregate functions or multiple row functions work on multiple data and returns a single result

## AVG Function

It calculates the average of given data

```
Select AVG(<column_name>) "Alias Name" from <table_name>;
```

## COUNT Function

It counts the number of rows in a given column

```
Select Count(<column_name>) "Alias Name" from <table_name>;
```

## MAX Function

It returns the maximum value from a given column

```
Select Max(<column_name>) "Alias Name" from <table_name>;
```

## MIN Function

It returns the minimum value from a given column

```
Select Min(<column_name>) "Alias Name" from <table_name>;
```

## SUM Function

It returns the sum of values in given column

```
Select Sum(<column_name>) "Alias Name" from <table_name>;
```

## MySQL Joins

Join clause is used to combine or merge rows from two or more tables based on a related attribute

### INNER JOIN

It returns all rows from multiple tables where the join condition is satisfied. It is the most common type of join.

```
SELECT columns FROM table1 INNER JOIN table2 ON table1.column = table2.column
```

## LEFT OUTER JOIN

It returns all rows from the left-hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

```
SELECT columns FROM table1 LEFT [OUTER] JOIN table2 ON table1.column = table
```

## RIGHT OUTER JOIN

It returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where the join condition is satisfied

```
SELECT columns FROM table1 RIGHT [OUTER] JOIN table2 ON table1.column = tabl
```

## FULL JOIN

It combines the results of both left and right outer joins

```
SELECT column_name FROM table1 FULL OUTER JOIN table2 ON table1.column_name
```

## SELF JOIN

In this join, table is joined with itself

```
SELECT column_name FROM table1 T1, table1 T2 WHERE condition;
```