MY SQL

Connect to MySQL Using mysql command-line client

mysql is a command-line client program that allows you to interact with MySQL in the interactive and non-interactive mode.

The mysql command-line client is typically located in the bin directory of the MySQL’s installation folder.

To invoke the mysql program, you just simply navigate to the bin directory of the MySQL’s installation folder and type:

mysql

If the mysql program is already in the PATH, you can simply invoke it using mysql command. To connect to the MySQL Server, you use this command:

shell>mysql -u root -p

Code language: SQL (Structured Query Language) (sql)

-u root means that you connect to the MySQL Server using the user account root.

-p instructs mysql to prompt for a password.

You type the password for the user account root and press Enter: Enter password: \*\*\*\*\*\*\*\*

Code language: SQL (Structured Query Language) (sql)

If everything is OK, you will connect to the MySQL Server with the following command: mysql>

Code language: SQL (Structured Query Language) (sql)

To display the databases in the current server, you use the [SHOW DATABASES](https://www.mysqltutorial.org/mysql-show-databases/) statement: mysql> show databases;

List of all MySQL commands:

Note that all te t commands must be ﬁrst on line and end with ';'

? (\?) Synonym for `help'.

clear (\c) Clear the current input statement.

connect (\r) Reconnect to the server. Optional arguments are db and host. delimiter (\d) Set statement delimiter.

ego (\G) Send command to mysql server, display result vertically. e it (\q) E it mysql. Same as quit.

go (\g) Send command to mysql server.

help (\h) Display this help.

notee (\t) Don't write into outﬁle. print (\p) Print current command.

prompt (\R) Change your mysql prompt. quit (\q) Quit mysql.

rehash (\ ) Rebuild completion hash.

source (\.) Eecute an SQL script ﬁle. Takes a ﬁle name as an argument. status (\s) Get status information from the server.

system (\!) Eecute a system shell command.

tee (\T) Set outﬁle [to\_outﬁle]. Append everything into given outﬁle. use (\u) Use another database. Takes database name as argument.

charsets.(\C) Switch to another charset. Might be needed for processing binlog with multi-byte warnings (\W) Show warnings after every statement.

nowarning (\w) Don't show warnings after every statement. resetconnection(\) Clean session conte t.

qpuicekryu\_pa. ttributes Sets string parameters (name1 value1 name2 value2 ...) for the ne t query to

mysql> SELECT VERSION(), CURRENT\_DATE;

+-----------+--------------+

| VERSION() | CURRENT\_DATE |

+-----------+--------------+

| 8.0.26 | 2021-11-16 |

+-----------+--------------+

1 row in set (0.04 sec)

you can use [**mysql**](https://dev.mysql.com/doc/refman/8.0/en/mysql.html) as a simple calculator:

mysql> SELECT SIN(PI()/4), (4+1)\*5;

+------------------+---------+

| SIN(PI()/4) | (4+1)\*5 |

+------------------+---------+

| 0.70710678118655 | 25 |

+------------------+---------+

1 row in set (0.02 sec)

mysql> SELECT VERSION(); SELECT NOW();

+-----------+

| VERSION() |

+-----------+

| 8.0.26 |

+-----------+

1 row in set (0.00 sec)

+---------------------+

| NOW() |

+---------------------+

| 2021-11-16 12:00:59 |

+---------------------+

1 row in set (0.00 sec)

mysql> SELECT

-> USER()

-> ,

-> CURRENT\_DATE;

+---------------+--------------+

| USER() | CURRENT\_DATE |

+---------------+--------------+

| jon@localhost | 2018-08-24 |

+---------------+--------------+

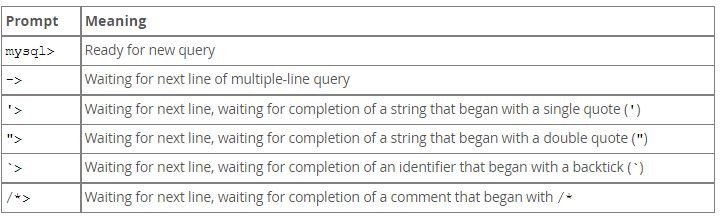
If you decide you do not want to e ecute a query that you are in the process of entering, cancel it by typing \c:

mysql> SELECT

-> USER()

-> \c

mysql>



mysql> SHOW DATABASES;

+----------+

| Database |

+----------+

| mysql |

| test |

| tmp |

+----------+

mysql> create schema ‘abduldb’;

New schema will be created with the name ‘abduldb’

Your database needs to be created only once, but you must select it for use each time you begin a **mysql** session.

You can do this by issuing a USE statement.

mysql> use abduldb;

Database changed

mysql> select \* from branch; Empty set (0.00 sec)

mysql> show tables;

+-------------------+

| Tables\_in\_abduldb |

+-------------------+

| branch |

+-------------------+

1. row in set (0.02 sec)

mysql> describe branch;

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | E tra |

+------------+-------------+------+-----+---------+-------+

| idbranch | int | NO | PRI | NULL | |

| branchName | varchar(45) | NO | | NULL | |

+------------+-------------+------+-----+---------+-------+

1. rows in set (0.00 sec)

mysql> drop table student;

Query OK, 0 rows affected (2.66 sec)

syntax:

DROP DATABASE *databasename*;

Drop Database abduldb;

Command to create a schema or (DB)

Synta : CREATE SCHEMA < name of the schema>; CREATE SCHEMA `abduldb` ;

The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

The FOREIGN KEY constraint is used to link data between tables.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the [PRIMARY KEY](https://www.w3schools.com/mySQl/mysql_primarykey.asp) in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

create table branch( branchid int, branchname varchar(200), primary key(branchid));

create table student(id int not null, name varchar(300) not null, address varchar(400), branchid int,

primary key(id),

# foreign key(branchid) references branch(branchid));

To retrieve data from a table we use select statement. Synta :

Select < columns > from <table name> where < condition to get what u want> select \* from student;

select \* from branch;

select name, branchid from student;

select name, address, branchid from student; select \* from student where branchid=1;

select \* from student where branchid =1 or address='Bangalore'; select \* from student where branchid =3 and address='Hyderabad'; select \* from student where branchid in(4,2);

A join is used to retrieve data from more than 1 table.

It is a technique to get data from more than one tables which are linked by a common column known as foreign key.

Types of joins:

Inner join

Left join Right Join Cross join

The inner join clause joins two tables based on a condition which is known as a join predicate.

The inner join clause compares each row from the ﬁrst table with every row from the second table.

In other words, the inner join clause includes only matching rows from both tables. SELECT column\_list FROM table\_1

INNER JOIN table\_2 ON join\_condition;

select t.idno, t.tr\_name, t.gender, t.stipend, a.idno 'assingmentID', a.title, a.start\_date from trainee t

inner join assignment a on t.as\_idno = a.idno;

The left join selects all data from the left table whether there are matching rows exist in the right table or not.

In case there are no matching rows from the right table found, the left join uses NULLs for columns of the row from the right table in the result set.

SELECT column\_list FROM table\_1

LEFT JOIN table\_2 ON join\_condition;

The right join clause selects all rows from the right table and matches rows in the left table. If a row from the right table does not have matching rows from the left table, the column of the left table will have NULL in the ﬁnal result set.

SELECT column\_list FROM table\_1

RIGHT JOIN table\_2 ON join\_condition;

The cross join makes a Cartesian product of rows from the joined tables. The cross join combines each row from the ﬁrst table with every row from the right table to make the result set.

Suppose the ﬁrst table has n rows and the second table has m rows. The cross join that joins the tables will return n m rows.

SELECT select\_list FROM table\_1 CROSS JOIN table\_2;

Update statement used to edit e isting data in the table. Synta :

UPDATE table\_name SET

column\_name1 = e pr1, column\_name2 = e pr2, ... [WHERE condition];

E ample:

update assignment set title='As\_ABC1', start\_date='2022-09-11' where idno=1;

Delete command is used to delete row set from the table. DELETE FROM table\_name

WHERE condition;

If you omit the WHERE clause, the DELETE statement will delete all rows in the table.

Alter statement is used to add a column, alter a column, rename a column, drop a column and rename a table.

Adding a column:

ALTER TABLE table\_name

ADD new\_column\_name column\_deﬁnition [FIRST | AFTER column\_name]

\* You can add a column after an existing column (ATER column\_name) or as the ﬁrst column (FIRST). If you omit this clause, the column is appended at the end of the column list of the table.

ALTER TABLE table\_name

ADD new\_column\_name column\_deﬁnition [FIRST | AFTER column\_name],

ADD new\_column\_name column\_deﬁnition [FIRST | AFTER column\_name],

...;

ALTER TABLE table\_name

MODIFY column\_name column\_deﬁnition [ FIRST | AFTER column\_name];

E : Alter Table Emp

Modify designation not null;

( designation column will be changed to not null)

ALTER TABLE table\_name

CHANGE COLUMN original\_name new\_name column\_deﬁnition [FIRST | AFTER column\_name];

To see the table structure: Describe Table Name:

ALTER TABLE table\_name DROP COLUMN column\_name;

ALTER TABLE table\_name RENAME TO new\_table\_name;

Subquery is called an inner query while the query that contains the subquery is called an outer query. A subquery can be used anywhere that expression is used and must be closed in parentheses.

Subquery is a query nested within another query such as SELECT, INSERT, UPDATE or DELETE. Also, a subquery can be nested within another subquery.

Example:

Select name, stipend, address, gender from trainee

where as\_id = (select idno from assignment where title=’As\_ABC’);