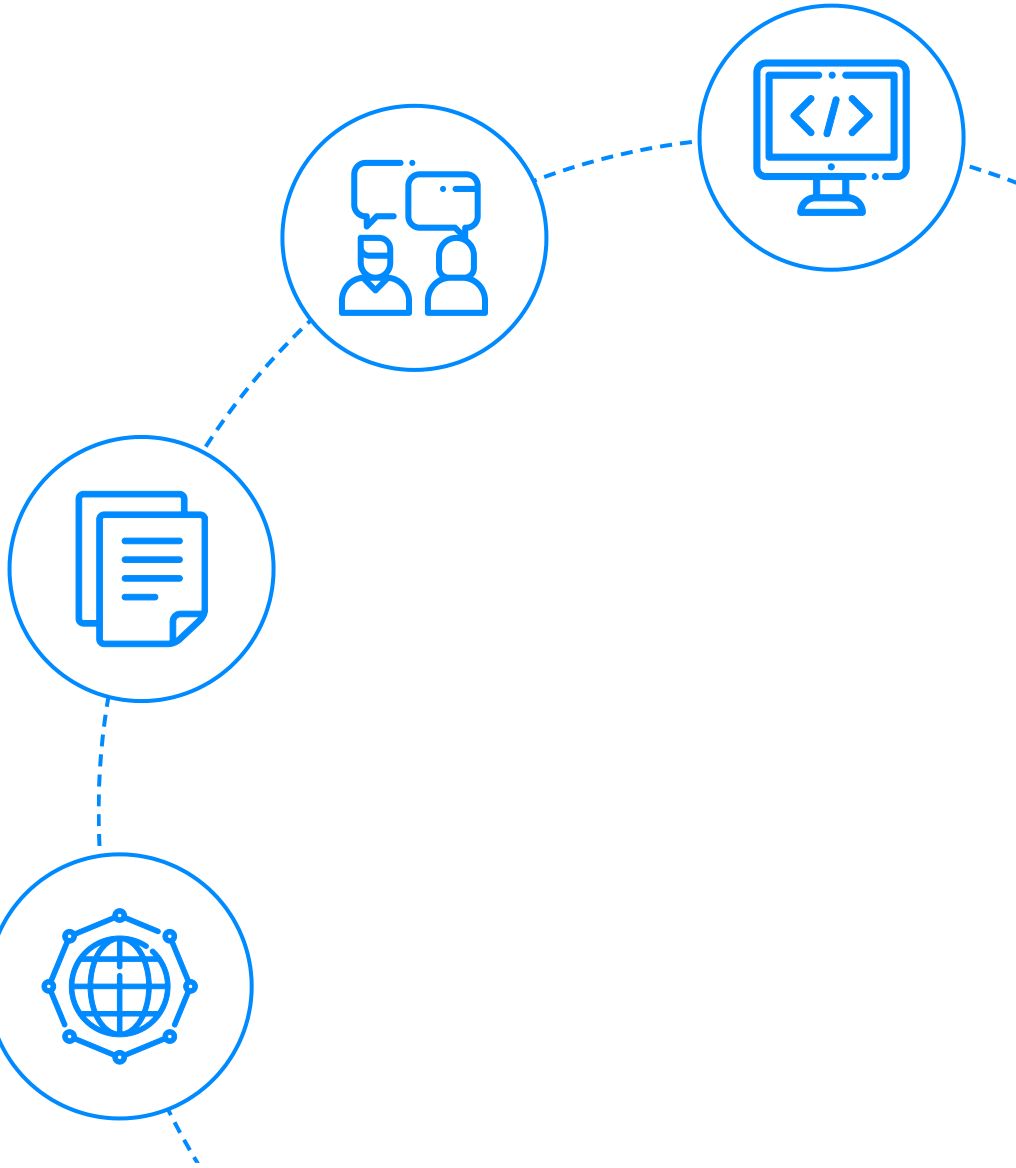
InterviewBit

MYSQL Cheat Sheet



To view the live version of the page, click here.

Contents

MYSQL Tutorial: Basics to Advanced

Let's get Started

MySQL is a fully open-source Relational Database Management System. It uses Structured Query Language (SQL) to manage the database. With only a few SQL statements, we can interact with MySQL using a simple programming language. SQL is a large group of statements that may be classified as sublanguages, typically: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). SQL includes data query, data manipulation (insert, update, and delete), data definition (schema creation and modification), and data access control.

MySql is platform flexible,i.e. can run on Windows, Linux and Mac. It can be easily used in both small-scale and large-scale businesses. SQL is largely based on Relational Algebra and Tuple Relational Calculus. Sun Microsystems (now Oracle Corporation) acquired MySQL AB, the Swedish company that owned and sponsored MySQL.

MySQL has stand-alone users that can interact with a MySQL database using SQL, but in more frequent instances, MySQL is used with other programs to create applications that require relational database expertise. LAMP is an acronym for Linux, Apache, MySQL, PHP/Python, and Perl/Python.LEMP is a stack which uses Linux, Nginx Server, MySQL, and PHP. Django Stack uses JS, Python, Django, and MySQL. MySQL is a component of this software stack.

MySQL is password encrypted which implies it is secure and lightweight. We can also implement a client /server architecture in MySQL. Transactions can be committed, rolled back and provides crash recovery. It also provides high performance, high flexibility, and high productivity.

We have made a cheat sheet, which will help you navigate and interact with MySQL.

INSTALLING MySQL

On Windows

1. Download the MYSQL installer from here: Install MySQL Installer. Execute the installer with administrator privileges.
2. Choose the appropriate setup type. Preferably Developer Default.
3. Complete the installation. This setup installs multiple MySQL products and the MySQL server is one of them.

On Linux

1. For distros that use apt (Debian based) run: sudo apt install mysql-server.
2. For distros that use yum, run : sudo yum install mysql-shell.
3. For distros that use dnf, run : sudo dnf install mysql-shell.

MYSQL Tutorial: Basics to Advanced

1. MYSQL COMMAND LINE COMMANDS

COMMAND	MEANING	SYNTAX
mysql	Allows user to connect to the MySQL CLI	`>MYSQL -U [USERNAME] -P;`
exit	Exits the MySQL CLI	`>EXIT;`
clear	Clears the MySQL shell	`>SYSTEM CLEAR;`
create user	Creates a new user	`>CREATE USER `NEWUSER`@`LOCALHOST` IDENT`
show user	Shows all user who have access to the MySQL Client	`>SELECT USER, HOST FROM MYSQL.USER;`
drop user	To delete an existing user	`> DROP USER 'USERNAME'@'LOCALHOST';`
grant all privileges	Assigns privileges to a MySQL user	`>GRANT ALL PRIVILEGES ON * . * TO 'USERNA`

2. MYSQL DATABASE COMMANDS (DATA DEFINITION LANGUAGE;DDL)

COMMAND	MEANING	SYNTAX
show database	Shows all the databases available in MySQL server.	`>SHOW DATABASE;`
create database	Creates a new database if it does not exist.	`>CREATE DATABASE DATABASENAME;`
drop database	To delete an existing database permanently.	`>DROP DATABASE  DATABASE_NAME`
alter database	Changes or modifies the characteristics of an existing database.	`>ALTER DATABASE [DATABASENAME] ALTEROP`
use database	Allow you to use a particular database or change from the current database to another database.	`>USE DATABASENAME;`

3. MySQL Table commands(DDL)

COMMAND	MEANING	SYNTAX
show tables	Shows all tables within the current database.	`>SHOW TABLES;`

create table	Creates a new table in the current database.	`>CREATE TABLE TABLENAME (`
		`COLUMN1 DATATYPE,`
		`COLUMN2 DATATYPE,`
		`COLUMN3 DATATYPE,`
		`....`
		`CONSTRAINTS ....`
		`);`

alter table (add column)	Adds a new column to an existing table.	`>ALTER TABLE TABLENAME ADD COLUMNNAME DATA`
alter table (drop column)	Deletes a column from an existing table.	`>ALTER TABLE TABLENAME DROP COLUMN COLUMNN`
alter table (alter column)	Alters an existing column in an already existing table.	`>ALTER TABLE TABLENAME` `ALTER COLUMN COLUMNNAME DATATYPE;`

4. MySQL DML(Data Manipulation Language) Commands

COMMAND	MEANING	SYNTAX
select *	Displays all rows in a table.	`>SELECT * FROM TABLENAME`
select * (multiple tables)	Displays all the rows of the cartesian product of the two tables	`>SELECT * FROM TABLENAME1,TABLENAME`
select columns	Select particular columns from table(s)	`>SELECT COLUMN1,COLUMN2 FROM TABLEN`
select with condition	Displays rows based on a particular condition	`> SELECT * FROM TABLENAME WHERE CON`
select with multiple conditions(AND)	Displays rows only when both the conditions are satisfied.	`> SELECT * FROM TABLENAME WHERE CON`

5. MySQL DATA TYPES

In MySQL just like other programming languages, each column, local variable, expression, and parameter has a related data type. A data type is an attribute that specifies the type of data that the object can hold.

- **String Data Types**

DATATYPE	DETAILS
CHAR(size)	Stores Alpha Numeric and special characters. Size varies from 0 to 255 characters.
VARCHAR(size)	Can contain letters, numbers, and characters that are of variable length (size). The size parameter specifies the column length in characters; it can be from 0 to 65535.
BINARY(size)	Similar to CHAR(). But it stores binary strings.
VARBINARY(size)	Similar to Binary() but the length is variable.
TINYBLOB	For Binary Large Objects. Max size=255 bytes.
TINYTEXT	Holds string of max length 255 characters.
TEXT(Size)	Stores a string of max length 65535 bytes.
BLOB	Stores Binary Large Objects up to 65535 bytes of data.
MEDIUMTEXT	Stores 2^8 times the characters as compared to TINYTEXT.
MEDIUMBLOB	Stores 2^8 times bytes as compared to TINYBLOB.
LONGTEXT	Stores 2^8 times the characters as compared to MEDIUMTEXT.

- **Numeric Data Types**

DATATYPE	DETAILS
BIT(size)	Stores a bit-value. The size parameter specifies the number of bits per value . The value is represented as a number of bits. The size parameter can hold a value from 1 to 64. The default value for size is 1.
TINYINT(size)	Stores very small int values. Signed ranges from -128 to 127. Unsigned ranges from 0 to 255. Size defines the maximum display width of 255.
BOOL	Zero is considered as false and one is considered as true.
BOOLEAN	Same as BOOL.
SMALLINT(size)	Stores a small integer. Signed ranges from -32768 to 32767. Unsigned ranges from 0 to 65535. Size defines the maximum display width of 255.
MEDIUMINT(size)	Stores a medium valued integer. Signed ranges from -8388608 to 8388607. Unsigned ranges from 0 to 16777215. Size defines the maximum display width of 255.
INT(size)	Stores a medium integer. Signed ranges from -2147483648 to 2147483647. Unsigned ranges from 0 to 4294967295. Size defines the maximum display width of 255.
INTEGER(size)	Same as INT(size)

- **Date and Time Data Types**

DATATYPE	DETAILS
DATE	Stores a date in the format: YYYY-MM-DD. Supports a range between '1000-01-01' to '9999-12-31'
DATETIME(fsp)	Combination of date and time in the format: YYYY-MM-DD hh:mm:ss. Supports a range between '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.
TIMESTAMP(fsp)	Stores a time stamp in the format YYYY-MM-DD hh:mm:ss UTC. Supports a range between '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC.
TIME(fsp)	Stores time in the format hh:mm:ss. Supports a range between '-838:59:59' to '838:59:59'
YEAR	Stores a year in four-digit format. Supports a range between 1901 to 2155 (includes 0000).

6. MySQL AGGREGATE FUNCTIONS

A function that performs an arithmetic operation on a set of values and returns a single value is called an aggregate function.

COMMAND	FUNCTION	SYNTAX
count()	Returns the number of rows, (including NULL)	`>SELECT COUNT(COLUMN_NAME) FROM TABLE_NAME WHERE CONDITION;`
sum()	Returns sum of all non NULL values.	`>SELECT SUM(COLUMN_NAME) FROM TABLE_NAME WHERE CONDITION;`
avg()	Returns average of all non NULL values.	`>SELECT AVG(COLUMN_NAME) FROM TABLE_NAME WHERE CONDITION;`
min()	Returns minimum value in the set.	`>SELECT MIN(COLUMN_NAME) FROM TABLE_NAME WHERE CONDITION;`
max()	Returns maximum value in the set.	`>SELECT MAX(COLUMN_NAME) FROM TABLE_NAME WHERE CONDITION;`
groutp_concat()	Concatenates values from multiple rows into one field.	`>SELECT COLUMN1, COLUMN2, ... GROUP_CONCAT ( DISTINCTCOLUMN1 ORDER BY .. ) FROM TABLE_NAME GROUP BY COLUMN2;`

7. INDEXES AND VIEWS IN MySQL

An Index retrieves data much faster than otherwise. Indexes speed up the query/search. A user cannot view an Index. Updating a table with an index takes more time because both table and index have to be updated.

The view is a virtual table which takes the result of an SQL query. Users can access a View. They have rows and columns similar to a table.

COMMAND	FUNCTION	SYNTAX
create index	Creates a new index from an existing table. Allows duplicate values.	`> CREATE INDEX indexname ON tablename (column1, column2, ...);`
create index unique	Similar to creating an index. But only allows unique values.	`>CREATE UNIQUE INDEX indexname ON tablename (column1, column2, ...);`
drop index	Deletes an existing index.	`> DROP INDEX INDEXNAME;`
rebuild index	Used to rebuild one or all indexes in a table if corrupted.	`>REINDEX INDEX INDEXNAME;`
create view	Creates a view if it doesn't exist.	`> CREATE VIEW VIEWNAME AS SELECT COLUMN1,C`

8. TRIGGERS IN MYSQL

Triggers are DBMS objects which are associated with tables. Triggers are fired when any one of the DML statements (INSERT, DELETE or UPDATE) is activated.

There are two types of triggers,

- Row Level Triggers: A trigger is an instruction that causes a row to trigger to be fired once for each row affected by an insert, update, or delete statement. The row trigger is fired automatically.
- Statement Level Trigger: Trigger is fired once regardless of the number of DML statements.

There are six types of triggers, namely,

- Before Insert: Activated before insertion.
- After Insert: Activated after insertion.
- Before Update: Activated before updating.
- After Update: Activated after updating.
- Before Delete: Activated before deletion.
- After Delete: Activated after deletion.

COMMAND	FUNCTION	SYNTAX
create trigger	Creates a new trigger on an existing table.	`>CREATE TRIGGER TRIGGERNAME BEFORE | AFTER INSERT | UPDATE| DELETE ON TABLENAME FOR EACH ROW TRIGGERBODY;`
drop trigger	Deletes an existing trigger.	`> DROP TRIGGER TRIGGERNAME;`
show all triggers	Displays all the triggers in the database.	`> SHOW TRIGGERS FROM | IN DATABASE_NAME WHERE SEARCH_CONDITION;`

9. STORED PROCEDURES AND FUNCTION

Procedures are reusable SQL codes that we store in a database. We can directly call procedures instead of writing the query again and again.

Functions are reusable code, which runs certain SQL commands and returns an appropriate value.

- **Syntax to create a new procedure.**

```
DELIMITER $$
CREATE PROCEDURE procedurename(parameterlist)
BEGIN
  body;
END $$
DELIMITER ;
```

- **Syntax to create a new function**

```
DELIMITER $$
CREATE FUNCTION functionname(parameterlist)
RETURNS datatype
NOT DETERMINISTIC
BEGIN
 %statements%
END $$

DELIMITER ;
```

COMMAND	FUNCTION	SYNTAX
drop procedure	Deletes an existing procedure.	> DROP PROCEDURE PROCEDURENAME;
show all procedures	Displays all the stored procedures in the database.	> SHOW PROCEDURE STATUS LIKE '%PATTERN' | WHERE CONDITION;
drop function	Deletes an existing stored function.	> DROP FUNCTION FUNCTIONNAME;
show stored functions	Displays all the stored functions.	> SHOW FUNCTION STATUS LIKE '%PATTERN' | WHERE CONDITION;

10. INBUILT FUNCTIONS IN MySQL

- **STRING FUNCTIONS**

Function	Description
ASCII	Returns the ASCII value of a character
CHAR_LENGTH	Returns the length of a string.
CHARACTER_LENGTH	Returns the length of a string
CONCAT	Concatenates two or more expressions.
CONCAT_WS	Concatenates with a separator.
FIELD	Returns the index of value in a list.
FIND_IN_SET	Returns the index of a string within a list.
FORMAT	Changes the format/representation.
INSERT	Inserts a string within a string at a given index.
INSTR	Returns the index of the first occurrence of a string in another one.
LCASE	Converts an entire string to lowercase.
LEFT	Extracts a length of characters from the left of a string.
LENGTH	Returns the string length in bytes.
	Returns the location of the first

- **NUMERIC FUNCTIONS**

Function	Description
ABS	Returns the absolute value.
ACOS	Returns the cosine inverse.
ASIN	Returns the sine inverse.
ATAN	Returns the tan inverse of one or two numbers.
ATAN2	Returns the tan inverse of two numbers.
AVG	Returns the mean value.
CEIL	Returns the smallest integer that is greater than or equal to the number
CEILING	Returns the smallest integer that is greater than or equal to the number
COS	Returns the cosine.
COT	Returns the cotangent.
COUNT	Returns the number of records returned by a query.
DEGREES	Converts angle in Radians to Degrees.
DIV	Integer division
EXP	Returns e raised to the power of value mentioned.
FLOOR	Returns the largest integer that is less than or equal to a number

- **MYSQL DATE FUNCTION**

Function	Description
ADDDATE	Adds a date interval and return the value.
ADDTIME	Adds a time interval and then returns the value.
CURDATE	Returns today's date
CURRENT_DATE	Same as CURDATE
CURRENT_TIME	Returns the time at the moment
CURRENT_TIMESTAMP	Returns date and time at the moment.
CURTIME	Returns time at the moment.
DATE	Picks up the date from an expression of Date/Time.
DATEDIFF	Returns number of days between two given dates.
DATE_ADD	Similar to ADDDATE
DATE_FORMAT	Changes the format in which Date is displayed.
DATE_SUB	Subtracts a time interval and returns the value.
DAY	Returns the weekday for today.
DAYNAME	Returns the weekday name for any date.

- **ADVANCED MYSQL FUNCTION**

Function	Description
BIN	Returns binary value of a given number.
BINARY	Converts a given string to a binary string.
CAST	Converts data from one data type to another.
COALESCE	Returns the first non-null value in a set or list.
CONV	Converts a number from one number-base system to another
CONVERT	Similar to CAST in working
CURRENT_USER	Returns the user name and host name for the MySQL account that is currently used.
DATABASE	Returns the name of the database currently in use.
IF	IF condition statement.
SESSION_USER	Returns the current MySQL user name and host name.
SYSTEM_USER	Similar to SESSION_USER.
USER	Similar to SESSION_USER.
VERSION	Returns the current version of the MySQL server installed.

CONCLUSION

By going through this cheat sheet, you would have got a decent understanding/revision of MySQL. More than memorizing syntax, do pay attention to practising them and solving problems.

Additional Resources

- MySQL Interview Questions
- MySQL Commands: Full List With Examples
- Top MySQL Features

Links to More Interview Questions

C Interview Questions	Php Interview Questions	C Sharp Interview Questions
Web Api Interview Questions	Hibernate Interview Questions	Node Js Interview Questions
Cpp Interview Questions	Oops Interview Questions	Devops Interview Questions
Machine Learning Interview Questions	Docker Interview Questions	Mysql Interview Questions
Css Interview Questions	Laravel Interview Questions	Asp Net Interview Questions
Django Interview Questions	Dot Net Interview Questions	Kubernetes Interview Questions
Operating System Interview Questions	React Native Interview Questions	Aws Interview Questions
Git Interview Questions	Java 8 Interview Questions	Mongodb Interview Questions
Dbms Interview Questions	Spring Boot Interview Questions	Power Bi Interview Questions
Pl Sql Interview Questions	Tableau Interview Questions	Linux Interview Questions
Ansible Interview Questions	Java Interview Questions	Jenkins Interview Questions