**# MySQL Cheat Sheet**

> Help with SQL commands to interact with a MySQL database

**## MySQL Locations**

\* Mac             *\*/usr/local/mysql/bin\**

\* Windows         *\*/Program Files/MySQL/MySQL \_version\_/bin\**

\* Xampp           *\*/xampp/mysql/bin\**

**## Add mysql to your PATH**

# Current Session

export PATH=${PATH}:/usr/local/mysql/bin

# Permanantly

echo 'export PATH="/usr/local/mysql/bin:$PATH"' >> ~/.bash\_profile

On Windows - https://www.qualitestgroup.com/resources/knowledge-center/how-to-guide/add-mysql-path-windows/

**## Login**

mysql -u root -p

**## Show Users**

SELECT User, Host FROM mysql.user;

**## Create User**

CREATE USER 'someuser'@'localhost' IDENTIFIED BY 'somepassword';

**## Grant All Priveleges On All Databases**

GRANT ALL PRIVILEGES ON \* . \* TO 'someuser'@'localhost';

FLUSH PRIVILEGES;

**## Show Grants**

SHOW GRANTS FOR 'someuser'@'localhost';

**## Remove Grants**

REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'someuser'@'localhost';

**## Delete User**

DROP USER 'someuser'@'localhost';

**## Exit**

exit;

**## Show Databases**

SHOW DATABASES

**## Create Database**

CREATE DATABASE acme;

**## Delete Database**

DROP DATABASE acme;

**## Select Database**

USE acme;

**## Create Table**

CREATE TABLE users(

id INT AUTO\_INCREMENT,

   first\_name VARCHAR(100),

   last\_name VARCHAR(100),

   email VARCHAR(50),

   password VARCHAR(20),

   location VARCHAR(100),

   dept VARCHAR(100),

   is\_admin TINYINT(1),

   register\_date DATETIME,

   PRIMARY KEY(id)

);

**## Delete / Drop Table**

DROP TABLE tablename;

**## Show Tables**

SHOW TABLES;

**## Insert Row / Record**

INSERT INTO users (first\_name, last\_name, email, password, location, dept, is\_admin, register\_date) values ('Brad', 'Traversy', 'brad@gmail.com', '123456','Massachusetts', 'development', 1, now());

**## Insert Multiple Rows**

INSERT INTO users (first\_name, last\_name, email, password, location, dept,  is\_admin, register\_date) values ('Fred', 'Smith', 'fred@gmail.com', '123456', 'New York', 'design', 0, now()), ('Sara', 'Watson', 'sara@gmail.com', '123456', 'New York', 'design', 0, now()),('Will', 'Jackson', 'will@yahoo.com', '123456', 'Rhode Island', 'development', 1, now()),('Paula', 'Johnson', 'paula@yahoo.com', '123456', 'Massachusetts', 'sales', 0, now()),('Tom', 'Spears', 'tom@yahoo.com', '123456', 'Massachusetts', 'sales', 0, now());

**## Select**

SELECT \* FROM users;

SELECT first\_name, last\_name FROM users;

**## Where Clause**

SELECT \* FROM users WHERE location='Massachusetts';

SELECT \* FROM users WHERE location='Massachusetts' AND dept='sales';

SELECT \* FROM users WHERE is\_admin = 1;

SELECT \* FROM users WHERE is\_admin > 0;

**## Delete Row**

DELETE FROM users WHERE id = 6;

**## Update Row**

UPDATE users SET email = 'freddy@gmail.com' WHERE id = 2;

**## Add New Column**

ALTER TABLE users ADD age VARCHAR(3);

**## Modify Column**

ALTER TABLE users MODIFY COLUMN age INT(3);

**## Order By (Sort)**

SELECT \* FROM users ORDER BY last\_name ASC;

SELECT \* FROM users ORDER BY last\_name DESC;

**## Concatenate Columns**

SELECT CONCAT(first\_name, ' ', last\_name) AS 'Name', dept FROM users;

**## Select Distinct Rows**

SELECT DISTINCT location FROM users;

**## Between (Select Range)**

SELECT \* FROM users WHERE age BETWEEN 20 AND 25;

**## Like (Searching)**

SELECT \* FROM users WHERE dept LIKE 'd%';

SELECT \* FROM users WHERE dept LIKE 'dev%';

SELECT \* FROM users WHERE dept LIKE '%t';

SELECT \* FROM users WHERE dept LIKE '%e%';

**## Not Like**

SELECT \* FROM users WHERE dept NOT LIKE 'd%';

**## IN**

SELECT \* FROM users WHERE dept IN ('design', 'sales');

**## Create & Remove Index**

CREATE INDEX LIndex On users(location);

DROP INDEX LIndex ON users;

**## New Table With Foreign Key (Posts)**

CREATE TABLE posts(

id INT AUTO\_INCREMENT,

   user\_id INT,

   title VARCHAR(100),

   body TEXT,

   publish\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

   PRIMARY KEY(id),

   FOREIGN KEY (user\_id) REFERENCES users(id)

);

**## Add Data to Posts Table**

INSERT INTO posts(user\_id, title, body) VALUES (1, 'Post One', 'This is post one'),(3, 'Post Two', 'This is post two'),(1, 'Post Three', 'This is post three'),(2, 'Post Four', 'This is post four'),(5, 'Post Five', 'This is post five'),(4, 'Post Six', 'This is post six'),(2, 'Post Seven', 'This is post seven'),(1, 'Post Eight', 'This is post eight'),(3, 'Post Nine', 'This is post none'),(4, 'Post Ten', 'This is post ten');

**## INNER JOIN**

SELECT

  users.first\_name,

  users.last\_name,

  posts.title,

  posts.publish\_date

FROM users

INNER JOIN posts

ON users.id = posts.user\_id

ORDER BY posts.title;

**## New Table With 2 Foriegn Keys**

CREATE TABLE comments(

  id INT AUTO\_INCREMENT,

    post\_id INT,

    user\_id INT,

    body TEXT,

    publish\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

    PRIMARY KEY(id),

    FOREIGN KEY(user\_id) references users(id),

    FOREIGN KEY(post\_id) references posts(id)

);

**## Add Data to Comments Table**

INSERT INTO comments(post\_id, user\_id, body) VALUES (1, 3, 'This is comment one'),(2, 1, 'This is comment two'),(5, 3, 'This is comment three'),(2, 4, 'This is comment four'),(1, 2, 'This is comment five'),(3, 1, 'This is comment six'),(3, 2, 'This is comment six'),(5, 4, 'This is comment seven'),(2, 3, 'This is comment seven');

**## Left Join**

SELECT

comments.body,

posts.title

FROM comments

LEFT JOIN posts ON posts.id = comments.post\_id

ORDER BY posts.title;

**## Join Multiple Tables**

SELECT

comments.body,

posts.title,

users.first\_name,

users.last\_name

FROM comments

INNER JOIN posts on posts.id = comments.post\_id

INNER JOIN users on users.id = comments.user\_id

ORDER BY posts.title;

**## Aggregate Functions**

SELECT COUNT(id) FROM users;

SELECT MAX(age) FROM users;

SELECT MIN(age) FROM users;

SELECT SUM(age) FROM users;

SELECT UCASE(first\_name), LCASE(last\_name) FROM users;

**## Group By**

SELECT age, COUNT(age) FROM users GROUP BY age;

SELECT age, COUNT(age) FROM users WHERE age > 20 GROUP BY age;

SELECT age, COUNT(age) FROM users GROUP BY age HAVING count(age) >=2;