MySQL Syntax Cheat Sheet

# 1. Database Operations

* Create database:

CREATE DATABASE mydb;

* Show databases:

SHOW DATABASES;

* Use database:

USE mydb;

* Drop database:

DROP DATABASE mydb;

# 2. Table Operations

* Create table:

CREATE TABLE users (id INT PRIMARY KEY, name VARCHAR(50));

* Show tables:

SHOW TABLES;

* Describe table:

DESCRIBE users;

* Drop table:

DROP TABLE users;

* Alter table (add column):

ALTER TABLE users ADD email VARCHAR(100);

* Alter table (rename column):

ALTER TABLE users CHANGE name full\_name VARCHAR(50);

* Alter table (drop column):

ALTER TABLE users DROP COLUMN email;

# 3. Data Manipulation (DML)

* Insert data:

INSERT INTO users (id, name) VALUES (1, 'Alice');

* Select data:

SELECT \* FROM users;

* Select with condition:

SELECT \* FROM users WHERE name = 'Alice';

* Update data:

UPDATE users SET name = 'Bob' WHERE id = 1;

* Delete data:

DELETE FROM users WHERE id = 1;

# 4. Clauses and Operators

* WHERE:

SELECT \* FROM users WHERE name = 'Alice';

* AND/OR:

SELECT \* FROM users WHERE name = 'Alice' AND id = 1;

* IN:

SELECT \* FROM users WHERE name IN ('Alice', 'Bob');

* BETWEEN:

SELECT \* FROM orders WHERE date BETWEEN '2024-01-01' AND '2024-12-31';

* LIKE:

SELECT \* FROM users WHERE name LIKE 'A%';

* IS NULL / IS NOT NULL:

SELECT \* FROM users WHERE email IS NULL;

# 5. Joins

* INNER JOIN:

SELECT \* FROM orders INNER JOIN users ON orders.user\_id = users.id;

* LEFT JOIN:

SELECT \* FROM users LEFT JOIN orders ON users.id = orders.user\_id;

* RIGHT JOIN:

SELECT \* FROM orders RIGHT JOIN users ON orders.user\_id = users.id;

* FULL JOIN (emulated):

SELECT \* FROM users LEFT JOIN orders ON users.id = orders.user\_id UNION SELECT \* FROM users RIGHT JOIN orders ON users.id = orders.user\_id;

# 6. Grouping & Aggregates

* COUNT:

SELECT COUNT(\*) FROM users;

* GROUP BY:

SELECT department, COUNT(\*) FROM employees GROUP BY department;

* HAVING:

SELECT department, COUNT(\*) FROM employees GROUP BY department HAVING COUNT(\*) > 5;

# 7. Sorting & Limiting

* ORDER BY:

SELECT \* FROM users ORDER BY name ASC;

* LIMIT:

SELECT \* FROM users LIMIT 10;

* OFFSET:

SELECT \* FROM users LIMIT 10 OFFSET 20;

# 8. Indexing & Constraints

* Create index:

CREATE INDEX idx\_name ON users(name);

* Unique constraint:

CREATE TABLE users (id INT PRIMARY KEY, email VARCHAR(100) UNIQUE);

* Foreign key:

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

# 9. Stored Procedures & Functions

* Create procedure:

CREATE PROCEDURE GetUsers() BEGIN SELECT \* FROM users; END;

* Call procedure:

CALL GetUsers();

* Create function:

CREATE FUNCTION AddNumbers(a INT, b INT) RETURNS INT RETURN a + b;

# 10. User & Privileges

* Create user:

CREATE USER 'bob'@'localhost' IDENTIFIED BY 'pass';

* Grant privileges:

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

* Revoke privileges:

REVOKE ALL PRIVILEGES ON mydb.\* FROM 'bob'@'localhost';

* Show grants:

SHOW GRANTS FOR 'bob'@'localhost';