MySQL Cheat Sheet

A quick reference guide for commonly used MySQL commands and syntax, covering data definition, data manipulation, user management, and more.

Basic SQL Commands

Data Definition Language (DDL)

CREATE DATABASE: Creates a new database.

```
CREATE DATABASE database_name;
```

DROP DATABASE: Deletes an existing database.

```
DROP DATABASE database_name;
```

CREATE TABLE: Creates a new table.

```
CREATE TABLE table_name (
    column1 datatype constraints,
    column2 datatype constraints,
    ...
);
```

ALTER TABLE: Modifies an existing table structure.

```
ALTER TABLE table_name ADD column_name datatype;
ALTER TABLE table_name DROP COLUMN column_name;
ALTER TABLE table_name MODIFY COLUMN column_name datatype;
```

DROP TABLE: Deletes a table.

```
DROP TABLE table_name;
```

TRUNCATE TABLE: Removes all rows from a table.

```
TRUNCATE TABLE table_name;
```

Data Manipulation Language (DML)

SELECT: Retrieves data from one or more tables.

```
SELECT column1, column2 FROM table_name WHERE condition;
```

INSERT: Adds new rows to a table.

```
INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

UPDATE: Modifies existing data in a table.

```
UPDATE table_name SET column1 = value1 WHERE condition;
```

DELETE: Removes rows from a table.

```
DELETE FROM table_name WHERE condition;
```

REPLACE: Deletes and inserts new rows, if a row with the same primary key or unique index exists.

```
REPLACE INTO table_name (column1, column2) VALUES (value1, value2);
```

Data Control Language (DCL)

GRANT: Grants privileges to users.

```
GRANT privilege ON database.table TO 'user'@'host';
```

REVOKE: Revokes privileges from users.

```
REVOKE privilege ON database.table FROM 'user'@'host';
```

Common SQL Clauses and Operators

WHERE Clause

Filters records based on a condition.

```
SELECT * FROM table_name WHERE column_name = 'value';
```

Common operators: `=`, `!=`, `>`, `<`, `>=`, `<=`, `LIKE`, `BETWEEN`, `IN`.

LIKE: Pattern matching.

`%` - Represents zero or more characters.
`_` - Represents a single character.

```
SELECT * FROM table_name WHERE column_name LIKE 'a%'; -- Starts with 'a'
```

BETWEEN: Specifies a range.

```
SELECT * FROM table_name WHERE column_name BETWEEN 10 AND 20;
```

IN: Specifies a set of values.

```
SELECT * FROM table_name WHERE column_name IN ('value1', 'value2');
```

ORDER BY Clause

Sorts the result set.

```
SELECT * FROM table_name ORDER BY column_name ASC|DESC;
```

`ASC`: Ascending order (default).
`DESC`: Descending order.

GROUP BY Clause

Groups rows that have the same values into summary rows.

```
SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name;
```

Often used with aggregate functions like `COUNT`, `SUM`, `AVG`, `MIN`, `MAX`.

HAVING: Filters groups based on a condition.

```
SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name HAVING COUNT(*) > 5;
```

LIMIT Clause

Limits the number of rows returned.

```
SELECT * FROM table_name LIMIT number;
SELECT * FROM table_name LIMIT offset, number;
```

`offset`: Specifies the offset of the first row to return.
`number`: Specifies the maximum number of rows to return.

Joins and Subqueries

JOIN Operations

INNER JOIN: Returns rows when there is a match in both tables.

```
SELECT * FROM table1 INNER JOIN table2 ON table1.column_name =
table2.column_name;
```

LEFT JOIN: Returns all rows from the left table, and the matched rows from the right table. If there is no match, the result is NULL on the right side.

```
SELECT * FROM table1 LEFT JOIN table2 ON table1.column_name =
table2.column_name;
```

RIGHT JOIN: Returns all rows from the right table, and the matched rows from the left table. If there is no match, the result is NULL on the left side.

```
SELECT * FROM table1 RIGHT JOIN table2 ON table1.column_name =
table2.column_name;
```

FULL OUTER JOIN: Returns all rows when there is a match in one of the tables. Note: MySQL does not directly support `FULL OUTER JOIN`, but it can be emulated using `UNION`.

```
SELECT * FROM table1 LEFT JOIN table2 ON table1.column_name =
table2.column_name
UNION
SELECT * FROM table1 RIGHT JOIN table2 ON table1.column_name =
table2.column_name;
```

CROSS JOIN: Returns the Cartesian product of the tables. Each row from the first table is combined with each row from the second table.

```
SELECT * FROM table1 CROSS JOIN table2;
```

Subqueries

A query nested inside another query.

```
SELECT * FROM table_name WHERE column_name IN (SELECT
column_name FROM another_table);
```

Can be used in `SELECT`, `WHERE`, `FROM` clauses.

Types: Scalar, Column, Row, Table subqueries.

User Management

User Account Management

CREATE USER: Creates a new MySQL user.

```
CREATE USER 'user'@'host' IDENTIFIED BY 'password';
```

DROP USER: Deletes a MySQL user.

```
DROP USER 'user'@'host';
```

RENAME USER: Renames a MySQL user.

```
RENAME USER 'old_user'@'host' TO 'new_user'@'host';
```

SET PASSWORD: Sets or changes the password for a MySQL user.

```
SET PASSWORD FOR 'user'@'host' = PASSWORD('new_password');
```

SHOW GRANTS: Displays the privileges granted to a MySQL user.

```
SHOW GRANTS FOR 'user'@'host';
```

Privilege Management

GRANT: Grants privileges to a user.

```
GRANT SELECT, INSERT ON database.table TO 'user'@'host';
GRANT ALL PRIVILEGES ON *.* TO 'user'@'host';
```

REVOKE: Revokes privileges from a user.

```
REVOKE SELECT, INSERT ON database.table FROM 'user'@'host';
REVOKE ALL PRIVILEGES ON *.* FROM 'user'@'host';
```

FLUSH PRIVILEGES: Reloads the grant tables after making changes to privileges.

```
FLUSH PRIVILEGES;
```