



### Basic Commands & Syntax

#### Browsing and Information

```
SHOW DATABASES; -- List all databases
SHOW TABLES; -- List all tables in the current database
SHOW FIELDS FROM table_name; -- Describe table structure (columns)
DESCRIBE table_name; -- Alternative to SHOW FIELDS
SHOW CREATE TABLE table_name; -- Show the SQL statement used to create the table
SHOW PROCESSLIST; -- Display active MySQL processes
KILL process_number; -- Terminate a specific process
```

#### SELECT Statements

```
SELECT * FROM table_name; -- Select all columns from a table
SELECT column1, column2 FROM table_name; -- Select specific columns
SELECT * FROM table1, table2; -- Select all columns from multiple tables (cross join)
SELECT column1, column2 FROM table1, table2; - - Select specific columns from multiple tables
SELECT ... FROM ... WHERE condition; -- Add a WHERE clause to filter results
SELECT ... FROM ... WHERE condition GROUP BY column_name; -- Group results by a column
SELECT ... FROM ... WHERE condition GROUP BY column_name HAVING condition2; -- Filter grouped results using HAVING
SELECT ... FROM ... WHERE condition ORDER BY column1, column2; -- Sort results by columns
SELECT ... FROM ... WHERE condition ORDER BY column1, column2 DESC; -- Sort in descending order
SELECT ... FROM ... WHERE condition LIMIT 10; -- Limit the number of results
SELECT DISTINCT column1 FROM ...; -- Select unique values from a column
SELECT DISTINCT column1, column2 FROM ...; -- Select unique combinations of values
```

#### SELECT with JOIN

```
SELECT ... FROM table1 JOIN table2 ON table1.id = table2.id WHERE condition;
SELECT ... FROM table1 LEFT JOIN table2 ON table1.id = table2.id WHERE condition;
SELECT ... FROM table1 RIGHT JOIN table2 ON table1.id = table2.id WHERE condition;
SELECT ... FROM table1 JOIN (table2 JOIN table3 ON ...) ON ...;
```

#### Common Conditions

'field1 = value1'	Exact match
'field1 <> value1'	Not equal to value1
'field1 LIKE 'value %''	Matches 'value' followed by any characters
'field1 IS NULL'	Field is NULL
'field1 IS NOT NULL'	Field is not NULL
'field1 IN (value1, value2)'	Field is one of the specified values
'condition1 AND condition2'	Both conditions must be true
'condition1 OR condition2'	At least one condition must be true

### Database and Table Management

#### Database Operations

```
CREATE DATABASE DatabaseName; -- Create a new database
CREATE DATABASE DatabaseName CHARACTER SET utf8; -- Create with specific character set
USE DatabaseName; -- Select a database for use
DROP DATABASE DatabaseName; -- Delete a database
ALTER DATABASE DatabaseName CHARACTER SET utf8; -- Modify character set
```

#### Table Creation

```
CREATE TABLE table_name (column1 datatype, column2 datatype, ...);
CREATE TABLE table_name (column1 datatype, column2 datatype, INDEX (column));
CREATE TABLE table_name (column1 datatype, column2 datatype, PRIMARY KEY (column1));
CREATE TABLE table_name (column1 datatype, column2 datatype, PRIMARY KEY (column1, column2));
CREATE TABLE table_name (fk_column datatype, ..., FOREIGN KEY (fk_column) REFERENCES table2 (column2) ON UPDATE CASCADE ON DELETE SET NULL);
CREATE TABLE IF NOT EXISTS table_name (...);
CREATE TEMPORARY TABLE table_name (...);
```

#### Table Alteration

```
ALTER TABLE table_name MODIFY column_name datatype; -- Modify a column's data type
ALTER TABLE table_name MODIFY column_name datatype NOT NULL; - - Add NOT NULL constraint
ALTER TABLE table_name CHANGE old_name new_name datatype; -- Rename a column
ALTER TABLE table_name ALTER column_name SET DEFAULT value; - - Set a default value
ALTER TABLE table_name ALTER column_name DROP DEFAULT; -- Remove a default value
ALTER TABLE table_name ADD column_name datatype; -- Add a new column
ALTER TABLE table_name DROP column_name; -- Delete a column
ALTER TABLE table_name ADD INDEX (column_name); -- Add an index
```

#### Table Deletion

```
DROP TABLE table_name; -- Delete a table
DROP TABLE IF EXISTS table_name; -- Delete if exists
DROP TABLE table1, table2, ...; -- Delete multiple tables
```

### Data Manipulation

## INSERT Statements

```
INSERT INTO table_name (column1, column2)
VALUES (value1, value2); -- Insert data into
specified columns

INSERT INTO table_name VALUES (value1, value2,
...); -- Insert data into all columns (order
matters!)
```

```
INSERT INTO table_name (column1, column2)
VALUES
(value1_1, value2_1),
(value1_2, value2_2),
...
; -- Insert multiple rows
```

## DELETE Statements

```
DELETE FROM table_name; -- Delete all rows
from a table (be careful!)

TRUNCATE TABLE table_name; -- Faster way to
delete all rows (resets auto-increment)

DELETE FROM table_name WHERE condition; --
Delete specific rows based on a condition

DELETE table1, table2 FROM table1 INNER JOIN
table2 ON table1.id = table2.table1_id WHERE
condition; -- Delete rows from multiple tables
```

## UPDATE Statements

```
UPDATE table_name SET column1 = new_value
WHERE condition; -- Update a column's value

UPDATE table1 INNER JOIN table2 ON table1.id =
table2.table1_id SET table1.column1 =
new_value, table2.column2 = new_value2 WHERE
condition; -- Update multiple columns in
multiple tables
```

## User Management & Data Types

### User and Privileges

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'password'; -- Create a new
user

GRANT ALL PRIVILEGES ON database_name.* TO 'user'@'localhost'; -- Grant
all privileges

GRANT SELECT, INSERT ON database_name.table_name TO 'user'@'localhost'; --
Grant specific privileges

REVOKE ALL PRIVILEGES ON database_name.* FROM 'user'@'localhost'; --
Revoke all privileges

REVOKE SELECT ON database_name.table_name FROM 'user'@'localhost'; --
Revoke specific privilege

FLUSH PRIVILEGES; -- Reload grant tables
```

```
ALTER USER 'user'@'localhost' IDENTIFIED BY 'new_password'; -- Change user
password

DROP USER 'user'@'localhost'; -- Delete a user
```

Note: Host % indicates any host.

### Backup and Restore & Root Password Reset

#### Backup and Restore

```
mysqldump -u username -p database_name > backup.sql # Backup a database to
a SQL file

mysql -u username -p database_name < backup.sql # Restore a database from
a SQL file
```

```
mysqlcheck --all-databases # Check all databases for errors

mysqlcheck --all-databases --fast # Perform a quick check
```

### Main Data Types

Integer Types	TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT
Floating-Point Types	FLOAT(M,D), DOUBLE(M,D)
Date and Time Types	DATE, TIME, DATETIME, TIMESTAMP, YEAR
String Types	VARCHAR(size), TEXT, MEDIUMTEXT, LONGTEXT
Binary Types	BLOB, MEDIUMBLOB, LONGBLOB
ENUM Type	ENUM('value1', 'value2', ...)

### Resetting Root Password (If Forgotten)

This process might vary depending on your OS and MySQL version. Always consult the official documentation for the most accurate and safe procedure.

#### Steps (General Guide):

1. Stop the MySQL server: `sudo systemctl stop mysql` or `/etc/init.d/mysql stop`
2. Start the MySQL server with the `--skip-grant-tables` option: `sudo mysqld_safe --skip-grant-tables &`
3. Connect to the MySQL server as root without a password: `mysql -u root`
4. Update the root password: `UPDATE mysql.user SET authentication_string=PASSWORD('new_password') WHERE User='root';` (Older versions might use `password` instead of `authentication_string`)
5. Flush privileges and exit: `FLUSH PRIVILEGES;` then `exit;`
6. Restart the MySQL server normally: `sudo systemctl start mysql` or `/etc/init.d/mysql start`
7. Connect to MySQL with the new password.