

Connection & Database Management

MySQL is an open-source relational database management system (RDBMS).

- **Terminal Login:** `mysql -u username -p -h hostname -P 3306`
- **List Databases:** `SHOW DATABASES;`
- **Create / Drop Database:** `CREATE DATABASE mydb CHARACTER SET utf8mb4; / DROP DATABASE mydb;`
- **Select Database:** `USE mydb;`
- **Show Tables / Schema:** `SHOW TABLES; / DESCRIBE users; (or EXPLAIN users;)`

Data Definition Language (DDL)

Create Table

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  email VARCHAR(100) NOT NULL,
  age TINYINT UNSIGNED DEFAULT 0,
  created_at DATETIME DEFAULT
CURRENT_TIMESTAMP,
  INDEX idx_email (email)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Alter Table

- Add Column: `ALTER TABLE users ADD COLUMN phone VARCHAR(20) AFTER email;`
- Modify Column: `ALTER TABLE users MODIFY COLUMN username VARCHAR(100) NOT NULL;`
- Drop Column: `ALTER TABLE users DROP COLUMN phone;`
- Create/Drop Index: `CREATE INDEX idx_user_age ON users(age); / DROP INDEX idx_user_age ON users;`

Data Manipulation Language (DML)

INSERT Statements

```
INSERT INTO users (username, email, age)
```

```
VALUES ('alice', 'alice@example.com', 25),
       ('bob', 'bob@example.com', 30);
```

```
-- Upsert (Insert on duplicate key update)
INSERT INTO users (id, username, email)
VALUES (1, 'alice', 'a@a.com')
ON DUPLICATE KEY UPDATE email =
VALUES(email);
```

SELECT Queries

```
SELECT id, username, age FROM users
WHERE age ≥ 20 AND email LIKE
' %@gmail.com'
ORDER BY age DESC, created_at ASC
LIMIT 10 OFFSET 20; -- Page 3 (Rows 21-30)
```

UPDATE & DELETE

- `UPDATE users SET age = age + 1, email = 'new@a.com' WHERE id = 1;`
- `DELETE FROM users WHERE age < 18;`
- `TRUNCATE TABLE logs;` (High-speed table reset)

Joins & Aggregation

Joins

- **INNER JOIN:** Matches in both tables

```
SELECT u.username, p.title
FROM users u
INNER JOIN posts p ON u.id = p.user_id;
```
- **LEFT JOIN:** All rows from left table + matched right rows

```
SELECT u.username, COUNT(p.id) AS post_count
FROM users u
LEFT JOIN posts p ON u.id = p.user_id
GROUP BY u.id, u.username
HAVING post_count ≥ 5;
```

Built-in Functions & Aggregates

- **Aggregates:** `COUNT(*), SUM(val), AVG(val), MIN(val), MAX(val)`
- **Strings:** `CONCAT(a, ' ', b), SUBSTRING(str, pos, len), LENGTH(str)`

- **Date/Time:** `NOW(), CURDATE(), DATE_ADD(NOW(), INTERVAL 7 DAY), DATE_FORMAT(NOW(), '%Y-%m-%d')`
- **Conditionals:** `COALESCE(val, 'default'), IF(cond, true_val, false_val)`
- **CASE Expression:**

```
SELECT username,
CASE
  WHEN age ≥ 65 THEN 'Senior'
  WHEN age ≥ 20 THEN 'Adult'
  ELSE 'Youth'
END AS age_group
FROM users;
```

Transactions & Locks

```
START TRANSACTION;
UPDATE accounts SET balance = balance - 100 WHERE id = 1;
UPDATE accounts SET balance = balance + 100 WHERE id = 2;
COMMIT; -- Or ROLLBACK; to discard
```

Users & Permissions

- **Create User:** `CREATE USER 'dev'@'%' IDENTIFIED BY 'password123';`
- **Grant Privileges:** `GRANT ALL PRIVILEGES ON mydb.* TO 'dev'@'%';`
- **Revoke & Reload:** `REVOKE ALL ON mydb.* FROM 'dev'@'%'; / FLUSH PRIVILEGES;`

Backup & Restore (mysqldump)

- **Export:** `mysqldump -u root -p mydb > backup.sql`
- **Import:** `mysql -u root -p mydb < backup.sql`