

Tham Khảo Nhanh MySQL

Databases, bảng, CRUD, joins, indexes, users

Cơ Sở Dữ Liệu & Bảng

Quản Lý Database

```
CREATE DATABASE shop;
USE shop;
SHOW DATABASES;
DROP DATABASE shop;
```

Tạo Bảng

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  email VARCHAR(255) UNIQUE NOT NULL,
  age INT,
  active BOOLEAN DEFAULT TRUE,
  created TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Sửa Đổi Bảng

```
ALTER TABLE users ADD COLUMN city VARCHAR(100);
ALTER TABLE users MODIFY COLUMN age TINYINT;
ALTER TABLE users DROP COLUMN city;
ALTER TABLE users RENAME TO customers;
TRUNCATE TABLE users; -- remove all rows, keep table
DROP TABLE users;
```

CRUD

INSERT

```
INSERT INTO users (name, email, age)
VALUES ('Alice', 'alice@example.com', 30);
```

```
INSERT INTO users (name, email)
VALUES ('Bob', 'bob@example.com'),
('Carol', 'carol@example.com');
```

SELECT

```
SELECT * FROM users;
SELECT name, email FROM users WHERE age > 25;
SELECT * FROM users ORDER BY name ASC LIMIT 10;
SELECT * FROM users WHERE name LIKE 'A%';
SELECT DISTINCT city FROM users;
```

UPDATE

```
UPDATE users SET age = 31 WHERE name = 'Alice';
UPDATE users SET active = FALSE WHERE age < 18;
```

DELETE

```
DELETE FROM users WHERE name = 'Alice';
DELETE FROM users WHERE active = FALSE;
```

Lọc & Sắp Xếp

Mệnh Đề WHERE

```
WHERE age BETWEEN 18 AND 30
WHERE name IN ('Alice', 'Bob', 'Carol')
WHERE email IS NOT NULL
WHERE name LIKE '%smith%' -- contains
WHERE name REGEXP '^A' -- regex
```

Sắp Xếp & Giới Hạn

```
ORDER BY age DESC, name ASC
LIMIT 10
LIMIT 10 OFFSET 20 -- pagination
LIMIT 20, 10 -- offset, count
```

Hàm Tổng Hợp

COUNT(*)	Đếm tất cả hàng
COUNT(col)	Đếm giá trị không NULL
SUM(col)	Tổng giá trị số
AVG(col)	Giá trị trung bình
MAX(col) / MIN(col)	Giá trị lớn nhất / nhỏ nhất
GROUP_CONCAT(col)	Nối chuỗi trong nhóm

Joins

Các Loại JOIN

```
-- INNER JOIN
SELECT u.name, o.total
FROM users u
JOIN orders o ON u.id = o.user_id;

-- LEFT JOIN
SELECT u.name, o.total
FROM users u
LEFT JOIN orders o ON u.id = o.user_id;

-- Multiple joins
SELECT u.name, p.title, c.body
FROM users u
JOIN posts p ON p.user_id = u.id
JOIN comments c ON c.post_id = p.id;
```

Các Loại JOIN

INNER JOIN	Chỉ các hàng có match ở cả hai bảng
LEFT JOIN	Tất cả từ bảng trái, NULL nếu không match
RIGHT JOIN	Tất cả từ bảng phải, NULL nếu không match
CROSS JOIN	Tích Cartesian

GROUP BY & HAVING

Tổng Hợp Theo Nhóm

```
SELECT city, COUNT(*) AS total
FROM users
GROUP BY city
ORDER BY total DESC;

SELECT user_id, SUM(amount) AS revenue
FROM orders
GROUP BY user_id
HAVING revenue > 1000;
```

Subqueries

Subquery Cơ Bản

```
SELECT name FROM users
WHERE id IN (SELECT user_id FROM orders);

SELECT name,
(SELECT COUNT(*) FROM orders WHERE user_id = u.id) AS order_count
FROM users u;
```

CTEs

```
WITH active_users AS (
  SELECT * FROM users WHERE active = TRUE
)
SELECT * FROM active_users WHERE age > 25;
```

Indexes

Tạo & Quản Lý Index

```
CREATE INDEX idx_email ON users (email);
CREATE UNIQUE INDEX idx_unique_email ON users (email);
CREATE INDEX idx_city_age ON users (city, age);
SHOW INDEXES FROM users;
DROP INDEX idx_email ON users;
```

EXPLAIN

```
EXPLAIN SELECT * FROM users WHERE email = 'alice@example.com';
EXPLAIN ANALYZE SELECT ...; -- MySQL 8+
```

Stored Procedures & Triggers

Stored Procedure

```
DELIMITER //
CREATE PROCEDURE GetUserOrders(IN uid INT)
BEGIN
  SELECT * FROM orders WHERE user_id = uid;
END //
DELIMITER ;

CALL GetUserOrders(42);
```

Trigger

```
CREATE TRIGGER before_insert_users
BEFORE INSERT ON users
FOR EACH ROW
SET NEW.created = NOW();
```

Users & Quyền

Quản Lý User

```
CREATE USER 'alice'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON shop.* TO 'alice'@'localhost';
GRANT SELECT, INSERT ON shop.users TO 'bob'@'%';
REVOKE INSERT ON shop.users FROM 'bob'@'%';
DROP USER 'alice'@'localhost';
FLUSH PRIVILEGES;
```

Quyền Phổ Biến

ALL PRIVILEGES	Toàn quyền trên database
SELECT / INSERT / UPDATE / DELETE	Quyền DML
CREATE / DROP / ALTER	Quyền DDL
GRANT OPTION	Cho phép cấp quyền cho người khác

Mẫu Phổ Biến

Upsert

```
INSERT INTO users (id, name, email)
VALUES (1, 'Alice', 'alice@example.com')
ON DUPLICATE KEY UPDATE
  name = VALUES(name),
  email = VALUES(email);
```

JSON (MySQL 5.7+)

```
SELECT JSON_EXTRACT(data, '$.name') FROM logs;
SELECT data->>'$.name' FROM logs; -- unquoted
WHERE data->>'$.status' = 'active'
```

Window Functions (MySQL 8)

```
SELECT name, salary,
RANK() OVER (PARTITION BY dept ORDER BY salary DESC) AS rnk,
SUM(salary) OVER (PARTITION BY dept) AS dept_total
FROM employees;
```