

SQL THAM KHẢO NHANH

SELECT, JOIN, subquery, index, transaction

SELECT

```
SELECT * FROM users;
SELECT name, email FROM users;
SELECT DISTINCT city FROM users;
SELECT name AS full_name FROM users;
```

WHERE

Toán Tử So Sánh

= <> (!=) Bằng / không bằng

< > <= >= Toán tử so sánh

AND OR NOT Toán tử logic

IS NULL / IS NOT NULL Kiểm tra null

Khớp Pattern

```
SELECT * FROM users WHERE name LIKE 'A%';
-- % = any chars, = single char
SELECT * FROM users WHERE age IN (20, 25, 30);
SELECT * FROM users WHERE age BETWEEN 18 AND 30;
```

JOIN

Các Loại Join

INNER JOIN Hàng khớp trong cả hai bảng

LEFT JOIN Tất cả hàng bên trái + khớp bên phải

RIGHT JOIN Tất cả hàng bên phải + khớp bên trái

FULL OUTER JOIN Tất cả hàng từ cả hai bảng

CROSS JOIN Tích Descartes của hai bảng

Cú Pháp Join

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

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

INSERT / UPDATE / DELETE

Insert

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

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

Update

```
UPDATE users SET email = 'new@ex.com'
WHERE id = 1;
```

Delete

```
DELETE FROM users WHERE id = 1;
DELETE FROM users; -- delete all rows
```

CREATE TABLE

Cú Pháp

```
CREATE TABLE users (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT NOT NULL,
  email TEXT UNIQUE,
  age INTEGER DEFAULT 0,
  score REAL
);
```

Kiểu Dữ Liệu Phổ Biến

INTEGER Số nguyên

REAL Số thực dấu phẩy động

TEXT Chuỗi / dữ liệu văn bản

BLOB Dữ liệu nhị phân

BOOLEAN TRUE / FALSE (lưu dưới dạng 0/1)

DATE / DATETIME Giá trị ngày và timestamp

Ràng Buộc

PRIMARY KEY Định danh hàng duy nhất

NOT NULL Bắt buộc phải có giá trị

UNIQUE Không có giá trị trùng lặp

DEFAULT val Giá trị mặc định nếu bỏ qua

CHECK (expr) Quy tắc xác thực tùy chỉnh

FOREIGN KEY Tham chiếu đến bảng khác

Hàm Tổng Hợp

COUNT (*) Số hàng

COUNT (col) Giá trị không null trong cột

SUM (col) Tổng cột số

AVG (col) Trung bình cột số

MIN (col) Giá trị nhỏ nhất

MAX (col) Giá trị lớn nhất

Ví Dụ

```
SELECT COUNT(*) AS total,
AVG(age) AS avg_age,
MAX(score) AS top_score
FROM users;
```

GROUP BY / HAVING

```
SELECT city, COUNT(*) AS num_users
FROM users
GROUP BY city;
```

```
SELECT city, AVG(age) AS avg_age
FROM users
GROUP BY city
HAVING AVG(age) > 25;
```

WHERE lọc hàng trước khi nhóm; HAVING lọc nhóm sau khi tổng hợp

ORDER BY / LIMIT

```
SELECT * FROM users ORDER BY name ASC;
SELECT * FROM users ORDER BY age DESC;
SELECT * FROM users
ORDER BY city, name
LIMIT 10 OFFSET 20; -- skip 20, take 10
```

Subquery

Trong Mệnh Đề WHERE

```
SELECT name FROM users
WHERE id IN (
  SELECT user_id FROM orders
  WHERE total > 100
);
```

Bảng Dẫn Xuất

```
SELECT city, avg_age FROM (
  SELECT city, AVG(age) AS avg_age
  FROM users GROUP BY city
) WHERE avg_age > 30;
```

Index

```
CREATE INDEX idx_name ON users(name);
-- % = any chars, = single char
CREATE UNIQUE INDEX idx_email
ON users(email);
DROP INDEX idx_name;
```

Khi Nào Dùng Index

Cột trong WHERE Tăng tốc lọc

Cột trong JOIN ON Tăng tốc tra cứu join

Cột trong ORDER BY Tăng tốc sắp xếp

Cột có cardinality cao Nhiều giá trị duy nhất sẽ lợi nhất