

BÀI SQL Cơ bản – CHỦ ĐỀ: HỆ THỐNG BÁN HÀNG / ĐƠN HÀNH

Bảng users

```
users (  
  
    id BIGINT PRIMARY KEY,  
  
    username VARCHAR(50),  
  
    full_name VARCHAR(100),  
  
    email VARCHAR(100),  
  
    department_id BIGINT,  
  
    status INT,      -- 1: active, 0: inactive  
  
    created_at TIMESTAMP  
  
)
```

Bảng departments

```
departments (  
  
    id BIGINT PRIMARY KEY,  
  
    name VARCHAR(100)  
  
)
```

Bảng documents

```
documents (  
    id BIGINT PRIMARY KEY,  
    title VARCHAR(255),  
    creator_id BIGINT,  
    status VARCHAR(20), -- draft, processing, completed  
    created_at TIMESTAMP  
)
```

Bảng document_logs

```
document_logs (  
    id BIGINT PRIMARY KEY,  
    document_id BIGINT,  
    action VARCHAR(50), -- create, approve, reject, view  
    user_id BIGINT,  
    created_at TIMESTAMP  
)
```

Bài test

□ Thời gian gợi ý: 30-40 phút

□□ Mục tiêu:

- SELECT, WHERE
 - JOIN cơ bản
 - COUNT, GROUP BY
 - ORDER BY
 - Hiểu dữ liệu nghiệp vụ
-

Câu 1: Lấy danh sách user đang hoạt động

Yêu cầu:

Lấy username, full_name, email của user có status = 1

Câu 2: User theo phòng ban

Yêu cầu:

Lấy danh sách user kèm tên phòng ban, sắp xếp theo tên phòng ban tăng dần

Câu 3: Đếm số user theo phòng ban

Yêu cầu:

Mỗi phòng ban có bao nhiêu user (chỉ tính user active)

Output:

department_name | total_users

Câu 4: Văn bản do user tạo

Yêu cầu:

Lấy danh sách văn bản (title, created_at) do user có username = 'admin' tạo

Câu 5: Thống kê văn bản theo trạng thái

Yêu cầu:

Đếm số văn bản theo từng status

Tiêu chí pass Cơ bản

- Viết được SELECT đúng
- JOIN không sai logic
- Không dùng SELECT *
- Kết quả đúng nghiệp vụ

BÀI TEST SQL – CHỦ ĐỀ: HỆ THỐNG BÁN HÀNG / ĐƠN HÀNG

I. MÔ TẢ CHUNG

Giả định CSDL: MySQL / PostgreSQL

1□ Bảng customers

```
customers (  
    id BIGINT PRIMARY KEY,  
    customer_code VARCHAR(50),  
    full_name VARCHAR(100),  
    email VARCHAR(100),  
    status INT,      -- 1: active, 0: inactive  
    created_at TIMESTAMP  
)
```

2□ Bảng products

```
products (  
    id BIGINT PRIMARY KEY,
```

```
product_code VARCHAR(50),  
  
name VARCHAR(255),  
  
price DECIMAL(12,2),  
  
status INT      -- 1: selling, 0: stopped  
  
)
```

3 □ Bảng orders

```
orders (  
  
    id BIGINT PRIMARY KEY,  
  
    order_code VARCHAR(50),  
  
    customer_id BIGINT,  
  
    order_date TIMESTAMP,  
  
    status VARCHAR(20) -- new, paid, cancelled, completed  
  
)
```

4 □ Bảng order_items

```
order_items (  
  

```

```
id BIGINT PRIMARY KEY,  
  
order_id BIGINT,  
  
product_id BIGINT,  
  
quantity INT,  
  
unit_price DECIMAL(12,2)  
  
)
```

5 □ Bảng payment_logs

```
payment_logs (  
  
id BIGINT PRIMARY KEY,  
  
order_id BIGINT,  
  
payment_method VARCHAR(50), -- cash, bank, momo  
  
amount DECIMAL(12,2),  
  
payment_time TIMESTAMP  
  
)
```

II. BÀI TEST SQL – MỨC ĐỘ CƠ BẢN

☐ Thời gian: 30–40 phút

☐ Mục tiêu:

- SELECT, WHERE
 - JOIN cơ bản
 - COUNT, SUM
 - GROUP BY
 - Hiểu nghiệp vụ bán hàng
-

☐ Câu 1: Khách hàng đang hoạt động

Yêu cầu:

Lấy danh sách customer_code, full_name, email của khách hàng có status = 1

☐☐ Câu 2: Danh sách đơn hàng kèm tên khách hàng

Yêu cầu:

Lấy `order_code`, `order_date`, `status`, `full_name`

Sắp xếp theo `order_date` giảm dần

☐☐ Câu 3: Thống kê số đơn theo trạng thái

Yêu cầu:

Đếm số lượng đơn theo từng `status`

☐☐ Câu 4: Sản phẩm trong một đơn hàng

Yêu cầu:

Với `order_code = 'ORD001'`, liệt kê:

`product_code` | `product_name` | `quantity` | `unit_price`

☐ Câu 5: Tổng tiền của mỗi đơn hàng

Yêu cầu:

Tính tổng tiền = $SUM(quantity * unit_price)$ cho từng đơn hàng

☐ Tiêu chí pass Cơ bản

- JOIN đúng
 - Tính toán chính xác
 - Không lạm dụng SELECT *
-

III. BÀI TEST SQL – MỨC ĐỘ NÂNG CAO

☐ Thời gian: 60-90 phút

☐ Mục tiêu:

- Subquery

- LEFT JOIN / NOT EXISTS
 - Kiểm thử dữ liệu
 - Phát hiện sai lệch nghiệp vụ
 - SQL tối ưu
-

☐☐ Câu 1: Khách hàng chưa từng đặt đơn

Yêu cầu:

Liệt kê khách hàng chưa có bất kỳ order nào

☐☐ Câu 2: Đơn hàng chưa được thanh toán

Yêu cầu:

Lấy danh sách đơn hàng không có bản ghi trong payment_logs

☐☐ Câu 3: Đơn hàng có tổng tiền khác tổng thanh toán

Yêu cầu:

Phát hiện các đơn hàng mà:

$SUM(order_items) \neq SUM(payment_logs.amount)$

→ Mục tiêu: kiểm thử sai lệch dữ liệu tài chính

☐☐ Câu 4: Sản phẩm bán chạy nhất

Yêu cầu:

Tìm sản phẩm có tổng số lượng bán ra cao nhất
(Chỉ tính đơn status = 'completed')

☐☐ Câu 5: Lần mua gần nhất của mỗi khách hàng

Yêu cầu:

Với mỗi khách hàng, lấy đơn hàng mới nhất

Output:

customer_code | full_name | last_order_date

→ Gợi ý:

- MAX(order_date)
 - CTE hoặc subquery
-

☐☐ Câu 6: Phát hiện dữ liệu bất thường

Yêu cầu:

Liệt kê:

1. order_items có product_id không tồn tại
2. orders có customer_id không tồn tại

→ Mục tiêu: Data Integrity Testing

☐☐ Câu 7 (Bonus – Tư duy hệ thống)

Câu hỏi lý thuyết:

1. Nên index những cột nào?

2. Vì sao không nên tính tổng tiền từ `orders.total_amount` nếu có `order_items`?

Phiên bản #1

Được tạo 2026-04-03 10:27:40 UTC bởi Nguyễn Thị Hiền

Được cập nhật 2026-04-03 10:32:26 UTC bởi Nguyễn Thị Hiền