

# Bài tập về Thiết kế Cơ sở Dữ liệu và Thực hành SQL

*Hoàng-Nguyên Vũ*

## Bài 1: Thiết kế Cơ sở Dữ liệu

Thiết kế cơ sở dữ liệu cho một hệ thống ngân hàng với các bảng sau:

- **KHÁCH HÀNG:**

- CIF: VARCHAR(20), PRIMARY KEY
- Ma\_NV: VARCHAR(20), FOREIGN KEY REFERENCES GIAO\_DICH\_VIEN(Ma\_nhan\_vien)
- Trang\_thai\_CIF: VARCHAR(10)
- Ngay\_tao\_CIF: DATE
- Chi\_nhanh\_tao\_CIF: VARCHAR(50)
- Ma\_dinh\_danh: VARCHAR(20)
- Loai\_giay\_to\_dinh\_danh: VARCHAR(20)
- Loai\_KH: VARCHAR(20)
- Ten\_KH: VARCHAR(100)
- Ngay\_sinh: DATE
- Gioi\_tinh: VARCHAR(10)
- Nghe\_nghiep: VARCHAR(50)
- Hoc\_van: VARCHAR(50)
- Tinh\_trang\_hon\_nhan: VARCHAR(20)
- Thanh\_pho: VARCHAR(50)
- Quan\_huyen: VARCHAR(50)
- Phuong\_xa: VARCHAR(50)
- Chi\_tiet: VARCHAR(100)
- SDT: VARCHAR(15)
- Email: VARCHAR(50)

- **GIAO DỊCH VIÊN:**

- Ma\_nhan\_vien: VARCHAR(20), PRIMARY KEY
- Ten\_NV: VARCHAR(100)
- Chuc\_vu: VARCHAR(50)
- SDT: VARCHAR(15)
- Thanh\_pho: VARCHAR(50)

- Quan\_huyen: VARCHAR(50)
- Phuong\_xa: VARCHAR(50)
- Dia\_chi\_chi\_tiet: VARCHAR(100)

- **TÀI KHOẢN NGÂN HÀNG:**

- So\_TK: VARCHAR(20), PRIMARY KEY
- Ten\_TK: VARCHAR(100)
- Loai\_TK: VARCHAR(20)
- Ngay\_lap: DATE
- So\_du\_TK: DECIMAL(18,2)
- Trang\_thai: VARCHAR(10)
- Ky\_han: VARCHAR(20), FOREIGN KEY REFERENCES KY\_HAN(Ky\_han)
- Ma\_NV\_tao\_TK: VARCHAR(20), FOREIGN KEY REFERENCES GIAO\_DICH\_VIEN(Ma\_nhan\_vien)

- **KỶ HẠN:**

- Ky\_han: VARCHAR(20), PRIMARY KEY
- Lai\_suat: DECIMAL(5,2)

- **GIAO DỊCH:**

- Ma\_lenh\_GD: VARCHAR(20), PRIMARY KEY
- Ma\_loai\_GD: VARCHAR(20), FOREIGN KEY REFERENCES LOAI\_GD(Ma\_loai\_GD)
- Hinh\_thuc\_chuyen: VARCHAR(20)
- Thoi\_diem\_GD: TIMESTAMP
- TK\_ghi\_no: VARCHAR(20), FOREIGN KEY REFERENCES TAI\_KHOAN\_NGAN\_HANG(So\_TK)
- TK\_ghi\_co: VARCHAR(20), FOREIGN KEY REFERENCES TAI\_KHOAN\_NGAN\_HANG(So\_TK)
- Trang\_thai: VARCHAR(10)
- So\_tien: DECIMAL(18,2)
- Noi\_dung: VARCHAR(255)
- Ten\_nguoi\_thu\_huong: VARCHAR(100)

- **LOẠI GD:**

- Ma\_loai\_GD: VARCHAR(20), PRIMARY KEY
- Ten\_loai\_GD: VARCHAR(50)

- **PHÍ DV:**

- Ma\_loai\_GD: VARCHAR(20), FOREIGN KEY REFERENCES LOAI\_GD(Ma\_loai\_GD)

- `Hinh_thuc_chuyen`: `VARCHAR(20)`
- `Phi_dich_vu`: `DECIMAL(18,2)`
- `PRIMARY KEY` (`Ma_loai_GD`, `Hinh_thuc_chuyen`)

- **TÀI KHOẢN DIGITAL BANKING:**

- `Ten_dang_nhap`: `VARCHAR(50)`, `PRIMARY KEY`
- `Mat_khau`: `VARCHAR(50)`
- `Thoi_gian_tao`: `TIMESTAMP`
- `Trang_thai`: `VARCHAR(10)`

## Bài 2: Thực hành các câu lệnh SQL

Thực hành trên cơ sở dữ liệu đã thiết kế:

1. **INSERT**: Thêm một khách hàng mới vào bảng KHÁCH HÀNG.

```
1 INSERT INTO KHACH_HANG
2 (....)
3 VALUES
4 (....);
```

2. **SELECT**: Lấy danh sách tất cả các khách hàng thuộc chi nhánh Ho Chi Minh.

```
1 SELECT * FROM KHACH_HANG
2 WHERE Chi_nhanh_tao_CIF = ....;
```

3. **UPDATE**: Cập nhật số dư tài khoản của một khách hàng.

```
1 UPDATE TAI_KHOAN_NGAN_HANG
2 # Your code here #;
3
```

4. **DELETE**: Xóa một giao dịch viên khỏi bảng GIAO DỊCH VIÊN.

```
1 DELETE FROM GIAO_DICH_VIEN
2 # Your code here #;
3
```

## Bài 3: Thực hành các câu lệnh SQL nâng cao

1. **JOIN**: Lấy danh sách tất cả các khách hàng kèm theo tên giao dịch viên đã tạo CIF của họ.

```
1 SELECT KH.Ten_KH, NV.Ten_NV
2 # Your code here #;
3
```

2. **GROUP BY:** Đếm số lượng khách hàng theo trạng thái CIF và hiển thị theo chi nhánh.

```
1 SELECT Chi_nhanh_taoCIF, Trang_thai_CIF,
2        COUNT(1) AS So_luong_KH
3 FROM KHACH_HANG
4      # Your code here #;
```

3. **HAVING:** Lọc các chi nhánh có số lượng khách hàng 'Active' lớn hơn 10.

```
1 SELECT Chi_nhanh_taoCIF, COUNT(1) AS So_luong_KH
2 FROM KHACH_HANG
3      # Your code here #;
```

4. **SUBQUERY:** Lấy danh sách các tài khoản ngân hàng mà số dư lớn hơn số dư trung bình của tất cả các tài khoản.

```
1 SELECT So_TK, Ten_TK, So_du_TK
2 FROM TAI_KHOAN_NGAN_HANG
3      # Your code here #;
```

5. **UPDATE với JOIN:** Cập nhật tên khách hàng trong bảng KHÁCH HÀNG dựa trên dữ liệu trong bảng GIAO DỊCH.

```
1 UPDATE KHACH_HANG KH
2 SET KH.Ten_KH = GD.Ten_nguoi_thu_huong
3 FROM GIAO_DICH GD
4      # Your code here #;
```

6. **DELETE với điều kiện phức tạp:** Xóa tất cả các giao dịch có số tiền dưới 100,000 VND và được tạo trong năm 2023.

```
1 DELETE FROM GIAO_DICH
2      # Your code here #;
```

7. **WINDOW FUNCTION:** Tính toán số dư lũy kế của mỗi tài khoản ngân hàng sau mỗi giao dịch.

```
1 SELECT TK_ghi_no, Thoi_diem_GD, So_tien,
2        SUM(So_tien) OVER
3          (PARTITION BY TK_ghi_no ORDER BY Thoi_diem_GD)
4        AS So_du_luy_ke
5 FROM GIAO_DICH
6      # Your code here #;
```

8. **COMMON TABLE EXPRESSION (CTE):** Tìm các khách hàng có số tài khoản ngân hàng nhiều nhất.

```

1  WITH CustomerAccountCount AS (
2      SELECT CIF, COUNT(*) AS So_luong_TK
3      FROM TAI_KHOAN_NGAN_HANG
4      GROUP BY CIF
5  )
6  SELECT CIF, So_luong_TK
7  FROM CustomerAccountCount
8  # Your code here #;
9

```

## Bài 4: Thực hành về Trigger với MySQL

1. Trigger để tự động cập nhật ngày lập tài khoản khi tạo mới tài khoản ngân hàng:

**Yêu cầu:** Tạo trigger để tự động gán giá trị Ngày lập là ngày hiện tại (CURRENT\_DATE()) khi một tài khoản ngân hàng mới được thêm vào bảng TAI\_KHOAN\_NGAN\_HANG.

```

1  # Your code here #
2

```

2. Trigger để kiểm tra số dư tài khoản không âm khi thực hiện giao dịch:

**Yêu cầu:** Tạo trigger để kiểm tra số dư tài khoản sau khi thực hiện giao dịch. Nếu số dư âm, không cho phép giao dịch và hiển thị thông báo lỗi.

```

1  CREATE TRIGGER trg_check_so_du_am
2  BEFORE INSERT ON GIAO_DICH
3  FOR EACH ROW
4  BEGIN
5      # Your code here #
6  END;
7

```

3. Trigger để ghi lại lịch sử thay đổi trạng thái CIF của khách hàng:

**Yêu cầu:** Tạo trigger để ghi lại lịch sử thay đổi trạng thái CIF của khách hàng vào một bảng lịch sử khi trạng thái CIF bị thay đổi.

```

1  CREATE TABLE LICH_SU_TRANG_THAI_CIF (
2      CIF VARCHAR(20),
3      Trang_thai_cu VARCHAR(10),
4      Trang_thai_moi VARCHAR(10),
5      Ngay_thay_doi DATE,
6      PRIMARY KEY (CIF, Ngay_thay_doi)
7  );
8
9  CREATE TRIGGER trg_log_trang_thai_cif
10 AFTER UPDATE ON KHACH_HANG
11 FOR EACH ROW
12 BEGIN
13     # Your code here #
14 END;
15

```

4. **Trigger để tự động cập nhật trạng thái tài khoản Digital Banking khi khóa tài khoản ngân hàng:**

**Yêu cầu:** Tạo trigger để tự động cập nhật trạng thái tài khoản Digital Banking thành 'Inactive' nếu tài khoản ngân hàng tương ứng bị khóa.

```
1 CREATE TRIGGER trg_inactive_digital_banking
2 AFTER UPDATE ON TAI_KHOAN_NGAN_HANG
3 FOR EACH ROW
4 BEGIN
5     # Your code here #
6 END;
```

5. **Trigger để tự động cập nhật lãi suất khi thay đổi kỳ hạn của tài khoản:**

**Yêu cầu:** Tạo trigger để tự động cập nhật lãi suất cho tài khoản ngân hàng khi kỳ hạn của tài khoản thay đổi.

```
1 CREATE TRIGGER trg_update_lai_suat
2 AFTER UPDATE ON TAI_KHOAN_NGAN_HANG
3 FOR EACH ROW
4 BEGIN
5     # Your code here #
6 END;
```