



TÀI LIỆU HƯỚNG DẪN THỰC HÀNH MÔN HỌC IT004 - CƠ SỞ DỮ LIỆU (DATABASE)

NỘI DUNG THỰC HÀNH TUẦN 4

Hướng dẫn thực hành

Lê Võ Đình Kha - khalvd@uit.edu.vn

GIỚI THIỆU NỘI DUNG THỰC HÀNH TUẦN 4



NỘI DUNG

1. Một số hàm mở rộng trong SQL.

- Hàm CAST(), hàm GETDATE()
- Hàm RIGHT(e, n), hàm LEFT(e, n)

2. Câu lệnh CASE...END trong SQL.

3. Sử dụng Alias kết hợp Subquery (truy vấn con).

4. Bài tập thực hành và hỏi đáp.

1

**MỘT SỐ HÀM MỞ RỘNG TRONG SQL
CAST(), GETDATE(), RIGHT(e, n) và LEFT(e, n)**

MỘT SỐ HÀM MỞ RỘNG TRONG SQL

Hàm CAST() trong SQL

- Hàm CAST() trong SQL được sử dụng để chuyển đổi dữ liệu từ một kiểu dữ liệu này sang kiểu dữ liệu khác.
- Ví dụ: Chuyển đổi từ kiểu dữ liệu **VARCHAR** sang **INT**, từ **DATETIME** sang **DATE**,...

- Cú pháp tổng quát:

CAST(<expression> **AS** <target_data_type>)

- **expression**: Biểu thức hoặc giá trị muốn chuyển đổi.
 - **target_data_type**: Kiểu dữ liệu muốn chuyển sang.
- Ví dụ

SELECT CAST ('2025-04-10' **AS** DATETIME) **AS** Ngay;

SELECT CAST (123.456 **AS** INT) **AS** SoNguyen;

MỘT SỐ HÀM MỞ RỘNG TRONG SQL

Hàm CAST() trong SQL

- Các kiểu dữ liệu có thể sử dụng trong hàm CAST()

Kiểu dữ liệu	Mô tả	Ví dụ
INT	Kiểu dữ liệu số nguyên, lưu trữ các giá trị số nguyên.	CAST (123 AS INT)
VARCHAR(n)	Chuỗi ký tự có độ dài tối đa là n. n là số lượng ký tự tối đa.	CAST ('Hello' AS VARCHAR(50))
DATE	Kiểu dữ liệu ngày tháng, lưu trữ ngày (yyyy-mm-dd).	CAST ('2024-11-14' AS DATE)
DATETIME	Kiểu dữ liệu ngày tháng và thời gian (yyyy-mm-dd hh:mm).	CAST ('2024-11-14 10:20:30' AS DATETIME)
DECIMAL(p,s)	Kiểu dữ liệu số thập phân với độ chính xác p và số chữ số sau dấu thập phân s.	CAST (123.456 AS DECIMAL(10,2))
NUMERIC(p,s)	Giống như DECIMAL, lưu trữ số thập phân với độ chính xác p và số chữ số sau dấu thập phân s.	CAST (123.456 AS NUMERIC(10,2))
FLOAT, REAL	Kiểu dữ liệu số thực, lưu trữ giá trị số thực với độ chính xác thấp.	CAST (123.456 AS FLOAT)
CHAR(n)	Chuỗi ký tự cố định, độ dài cố định n ký tự.	CAST ('Hello' AS CHAR(10))

MỘT SỐ HÀM MỞ RỘNG TRONG SQL

Hàm GETDATE() trong SQL

- Hàm GETDATE() là một hàm ngày giờ trong SQL Server, được sử dụng để trả về ngày và giờ hiện tại (ngày giờ hệ thống) tại thời điểm truy vấn được thực thi.
- **Cú pháp tổng quát:** `GETDATE()`
 - Hàm không nhận bất kỳ tham số nào.
 - Kết quả trả về có kiểu dữ liệu **datetime** (bao gồm cả ngày và giờ).
 - GETDATE() là hàm dành riêng cho SQL Server.
- **Ví dụ:** Lấy ngày giờ hiện tại
`SELECT GETDATE() AS NgayGioHienTai;`

MỘT SỐ HÀM MỞ RỘNG TRONG SQL

Hàm RIGHT() trong SQL

- Hàm RIGHT() được sử dụng để trích xuất một số ký tự từ bên phải (cuối cùng) của một chuỗi văn bản.
- **Cú pháp tổng quát:** `RIGHT(<chuỗi>, <số ký tự>)`
 - **<chuỗi>:** Chuỗi văn bản hoặc cột dữ liệu chuỗi muốn trích xuất ký tự.
 - **<số ký tự>:** Số lượng ký tự muốn trích xuất từ bên phải chuỗi.
- **Ví dụ:** Trích xuất 3 ký tự cuối của chuỗi 'abcdef'. **Kết quả: def**

`RIGHT('abcdef', 3)`

MỘT SỐ HÀM MỞ RỘNG TRONG SQL

Hàm LEFT() trong SQL

- Hàm LEFT() được sử dụng để trích xuất một số ký tự từ bên trái (đầu tiên) của một chuỗi văn bản.
- **Cú pháp tổng quát:** `LEFT(<chuỗi>, <số ký tự>)`
 - **<chuỗi>:** Chuỗi văn bản hoặc cột dữ liệu chuỗi muốn trích xuất ký tự.
 - **<số ký tự>:** Số lượng ký tự muốn trích xuất từ bên trái chuỗi.
- **Ví dụ:** Trích xuất 3 ký tự đầu của chuỗi 'abcdef'. **Kết quả: abc**

`LEFT('abcdef', 3)`

2

CÂU LỆNH CASE...END TRONG SQL

CÂU LỆNH CASE...END TRONG SQL

Câu lệnh CASE...END

- **CASE** là câu lệnh điều kiện trong **SQL**. Lệnh **CASE** được sử dụng để thực hiện các điều kiện có cấu trúc tương tự như câu lệnh **IF...ELSE** trong các ngôn ngữ lập trình.
- **Cú pháp tổng quát:**

```
CASE <biểu_thức>  
    WHEN <giá_trị_1> THEN <kết_quả_1>  
    WHEN <giá_trị_2> THEN <kết_quả_2>  
    ...  
    ELSE <kết_quả_khác>  
END AS <bí_danh>
```
- **WHEN condition:** Câu điều kiện cần kiểm tra. **THEN result:** Giá trị trả về nếu điều kiện thỏa mãn. **ELSE (tuỳ chọn):** Giá trị trả về nếu không có điều kiện nào thỏa mãn.

CÂU LỆNH CASE...END TRONG SQL

Câu lệnh CASE...END

- Ví dụ `SELECT MaKH, HoTen,`

`CASE`

`WHEN TongChiTieu >= 10000000 THEN 'VIP'`

`WHEN TongChiTieu >= 5000000 THEN 'Than thiet'`

`ELSE 'Thuong'`

`END AS PhanLoai`

`FROM KháchHang;`

- **TongChiTieu:** $\geq 10.000.000 \rightarrow$ Khách hàng **VIP**; $\geq 5.000.000 \rightarrow$ Khách hàng Thân thiết; $< 5.000.000 \rightarrow$ Khách hàng Thường.
- **Alias PhanLoai** là tên của cột hiển thị kết quả phân loại.

CÂU LỆNH CASE...END TRONG SQL

Câu lệnh CASE...END

- Ví dụ

```
SELECT MaNV, HoTen,  
       CASE GioiTinh  
         WHEN 1 THEN 'Nam'  
         WHEN 0 THEN 'Nu'  
         ELSE 'Khong xac dinh'  
       END AS PhanLoaiGioiTinh  
FROM NhanVien;
```

- Dựa vào giá trị của GioiTinh, hệ thống gán biểu thức:
- 1 → 'Nam'; 0 → 'Nữ'; Bất kỳ giá trị khác → 'Không xác định'

3

SỬ DỤNG ALIAS KẾT HỢP SUBQUERY (TRUY VẤN CON)

SỬ DỤNG ALIAS KẾT HỢP SUBQUERY

Lý thuyết

- **Alias (bí danh):** Là tên tạm đặt cho cột hoặc bảng trong truy vấn SQL, giúp dễ đọc và dễ sử dụng hơn trong các biểu thức phức tạp.
- **Subquery (truy vấn con):** Là một truy vấn SQL lồng bên trong một truy vấn khác, có thể nằm trong SELECT, FROM, WHERE, HAVING,...
- **Khi kết hợp alias và subquery, alias thường được dùng để:**
 - Đặt tên cho truy vấn con trong mệnh đề FROM.
 - Đặt tên cho cột kết quả trong SELECT.

SỬ DỤNG ALIAS KẾT HỢP SUBQUERY

Cú pháp tổng quát

- Subquery dùng trong FROM + Alias

```
SELECT Alias1.cot1, Alias1.cot2
FROM (
    SELECT ...
    FROM ...
    WHERE ...
) AS Alias
WHERE ....
```

SỬ DỤNG ALIAS KẾT HỢP SUBQUERY

Cú pháp tổng quát

- Subquery dùng trong SELECT + Alias

```
SELECT <cột chính>,  
      (SELECT <biểu thức>  
      FROM <bảng liên quan>  
      WHERE <điều kiện liên kết>) AS bí danh  
FROM <bảng chính> (AS <bí danh bảng>);
```


SỬ DỤNG ALIAS KẾT HỢP SUBQUERY

Ví dụ

- Câu truy vấn: Đếm số lần bán của từng sản phẩm

```
SELECT TenSP,  
       (SELECT count(*)  
        FROM CTHD  
        WHERE MaSP = SP.MaSP) AS SoLanBan  
FROM SanPham SP;
```

GIỚI THIỆU NỘI DUNG THỰC HÀNH TUẦN 4

4

BÀI TẬP THỰC HÀNH VÀ HỎI ĐÁP

BÀI TẬP THỰC HÀNH VÀ HỎI ĐÁP

Yêu cầu: (Xem thêm đề bài tại file Bài tập thực hành tuần 4)

Phần bài làm lưu trữ trên một file script có tên **<MSSV>_<HoVaTen>_BTTH4.sql** (trong đó **MSSV** là mã số sinh viên, **HoVaTen** là họ và tên của sinh viên).

Sử dụng các câu lệnh SQL trong công cụ **SQL Server Management Studio** để thực hiện yêu cầu. **Phần I, II, III** bài tập **Quản lý giáo vụ**.



HỎI ĐÁP



Liên hệ hỗ trợ

Lê Võ Đình Kha - khalvd@uit.edu.vn