



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

NỘI DUNG THỰC HÀNH TUẦN 3

Hướng dẫn thực hành

Lê Võ Đình Kha - khalvd@uit.edu.vn

GIỚI THIỆU NỘI DUNG THỰC HÀNH TUẦN 3



NỘI DUNG

1. Phép chia (Division Operation) trong SQL.

- Phép chia cơ bản (Basic Division).
- Phép chia có điều kiện (Conditional Division).
- Phép chia với NOT EXISTS.

2. Hàm tính toán và gom nhóm trong SQL.

3. Bài tập thực hành và hỏi đáp.

GIỚI THIỆU NỘI DUNG THỰC HÀNH TUẦN 3

1

PHÉP CHIA TRONG SQL

PHÉP CHIA TRONG SQL

Tổng quan về phép chia trong SQL

- Phép chia được sử dụng để truy vấn các đối tượng trong một bảng (gọi là bảng **R**) có liên hệ với tất cả các đối tượng trong một bảng khác (gọi là bảng **S**).
- Phép chia rất hữu ích khi muốn tìm các đối tượng thỏa mãn một điều kiện trên tất cả các giá trị từ một tập hợp.
- **Ví dụ:**
 - Tìm khách hàng đã mua **tất cả** các sản phẩm trong một danh sách.
 - Tìm nhân viên đã tham gia **tất cả** các dự án trong một phòng ban.

PHÉP CHIA TRONG SQL

Tổng quan về phép chia trong SQL

R	A	B	C	D	E
α	a	α		a	1
α	a	γ		a	1
α	a	γ	b		1
β	a	γ		a	1
β	a	γ	b		3
γ	a	γ	a		1
γ	a	γ	b		1
γ	a	β	b		1

S	D	E
b_i	a	1
	b	1

R÷S	A	B	C
a_i	α	a	γ
	γ	a	γ

- **Bảng R:** Đây là bảng lớn hơn, chứa các cột A, B, C, D, E.
- **Bảng S:** Bảng nhỏ hơn, chỉ có 2 cột D và E.
- Kết quả của phép chia R/S sẽ tìm ra các giá trị trong cột A, B, C của bảng R sao cho các giá trị này liên kết với tất cả các giá trị trong bảng S.

PHÉP CHIA TRONG SQL

Phép chia cơ bản (Basic Division)

- Phép chia cơ bản được sử dụng khi cần tìm những đối tượng từ bảng **R** liên quan đến tất cả các đối tượng trong bảng **S**.
- **Cú pháp tổng quát:**

```
SELECT B1.<Tên cột 1>,...  
FROM <Tên bảng 1> B1 JOIN <Tên bảng 3> B3 ON B1.<Tên cột 1> = B3.<Tên cột 1>  
GROUP BY B1.<Tên cột 1>  
HAVING COUNT(DISTINCT B3.<Tên cột 2>) = (SELECT COUNT(DISTINCT B2.<Tên cột 2>)  
FROM <Tên bảng 2> B2);
```

PHÉP CHIA TRONG SQL

Phép chia cơ bản (Basic Division)

- **Ví dụ 1: Tìm khách hàng đã mua tất cả các loại sản phẩm.**
 - ✓ KhachHang (MaKH, TenKH): Danh sách khách hàng.
 - ✓ SanPham (MaSP, TenSP): Danh sách sản phẩm.
 - ✓ MuaHang (MaKH, MaSP): Ghi lại mỗi lần khách hàng mua một sản phẩm.

```
SELECT KH.MaKH  
FROM KhachHang KH JOIN MuaHang MH ON KH.MaKH = MH.MaKH  
GROUP BY KH.MaKH  
HAVING COUNT(DISTINCT MH.MaSP) = (SELECT COUNT(*)  
FROM SanPham);
```

PHÉP CHIA TRONG SQL

Phép chia cơ bản (Basic Division)

- **Ví dụ 2: Tìm sinh viên đã đăng ký tất cả các môn học bắt buộc.**

- ✓ SinhVien (MaSV, TenSV): Danh sách sinh viên.
- ✓ MonHoc (MaMH, TenMH): Danh sách các môn học bắt buộc.
- ✓ DangKy (MaSV, MaMH): Ghi lại mỗi lần sinh viên đăng ký học một môn học.

```
SELECT SV.MaSV  
FROM SinhVien SV JOIN DangKy DK ON SV.MaSV = DK.MaSV  
GROUP BY SV.MaSV  
HAVING COUNT(DISTINCT DK.MaMH) = (SELECT COUNT(*)  
FROM MonHoc);
```

PHÉP CHIA TRONG SQL

Phép chia cơ bản (Basic Division)

- **Ví dụ 3: Tìm nhân viên đã tham gia tất cả các dự án.**

- ✓ NhanVien (MaNV, TenNV): Danh sách nhân viên.
- ✓ DuAn (MaDA, TenDA): Danh sách các dự án.
- ✓ ThamGia (MaNV, MaDA): Ghi lại mỗi lần nhân viên tham gia một dự án.

```
SELECT NV.MaNV  
FROM NhanVien NV JOIN ThamGia TG ON NV.MaNV = TG.MaNV  
GROUP BY NV.MaNV  
HAVING COUNT(DISTINCT TG.MaDA) = (SELECT COUNT(*)  
        FROM DuAn);
```

PHÉP CHIA TRONG SQL

Phép chia có điều kiện (Conditional Division)

- Là mở rộng phép chia cơ bản bằng cách thêm các điều kiện vào phép chia.
- **Cú pháp tổng quát:**

SELECT B1.<Tên cột 1>,...

FROM <Tên bảng 1> B1 **JOIN** <Tên bảng 3> B3 **ON** B1.<Tên cột 1> = B3.<Tên cột 1>
JOIN <Tên bảng 2> B2 **ON** B3.<Tên cột 2> = B2.<Tên cột 2>

WHERE <Điều kiện>

GROUP BY B1.<Tên cột 1>

HAVING COUNT(**DISTINCT** B3.<Tên cột 2>) = (**SELECT** COUNT(B2.<Tên cột 2>)

FROM <Tên bảng 2> B2

WHERE <Điều kiện>);

PHÉP CHIA TRONG SQL

Phép chia có điều kiện (Conditional Division)

- Ví dụ 4: Tìm khách hàng đã mua tất cả các loại sản phẩm thuộc danh mục 'Điện tử'.

SELECT KH.MaKH

FROM KhachHang KH **JOIN** MuaHang MH **ON** KH.MaKH = MH.MaKH

JOIN SanPham SP **ON** MH.MaSP = SP.MaSP

WHERE SP.DanhMuc = 'Điện tử'

GROUP BY KH.MaKH

HAVING COUNT(**DISTINCT** MH.MaSP) = (**SELECT COUNT(*)**

FROM SanPham

WHERE DanhMuc = 'Điện tử');

PHÉP CHIA TRONG SQL

Phép chia có điều kiện (Conditional Division)

- Ví dụ 5: Tìm sinh viên đã đăng ký tất cả môn học bắt buộc có số tín chỉ lớn hơn 3.

```
SELECT SV.MaSV
```

```
FROM SinhVien SV JOIN DangKy DK ON SV.MaSV = DK.MaSV
```

```
JOIN MonHoc MH ON MH.MaMH = DK.MaMH
```

```
WHERE MH.SoTinChi > 3
```

```
GROUP BY SV.MaSV
```

```
HAVING COUNT(DISTINCT DK.MaMH) = (SELECT COUNT(*)
```

```
FROM MonHoc
```

```
WHERE MH.SoTinChi > 3);
```

PHÉP CHIA TRONG SQL

Phép chia có điều kiện (Conditional Division)

- **Ví dụ 6:** Tìm nhân viên đã tham gia tất cả các dự án có thời gian hoàn thành dưới 6 tháng.

SELECT NV.MaNV

FROM NhanVien NV JOIN ThamGia TG ON NV.MaNV = TG.MaNV

JOIN DuAn DA ON DA.MaDA = TG.MaDA

WHERE DA.ThoiGianHoanThanh < 6

GROUP BY NV.MaNV

HAVING COUNT(DISTINCT TG.MaDA) = (SELECT COUNT(*)

FROM DuAn

WHERE DA.ThoiGianHoanThanh < 6);

PHÉP CHIA TRONG SQL

Phép chia với NOT EXISTS

- Phép chia với NOT EXISTS là cách thể hiện phép chia cơ bản bằng cách sử dụng điều kiện phủ định để kiểm tra sự tồn tại của các giá trị trong bảng con.
- **Mục đích chính** của phép chia với **NOT EXISTS** là để kiểm tra liệu có bất kỳ giá trị nào mà không thỏa mãn điều kiện hay không.
- Điều này cho phép thực hiện các truy vấn kiểu "**mua tất cả**", "**thực hiện tất cả các nhiệm vụ**", "**tham gia tất cả các sự kiện**",...

PHÉP CHIA TRONG SQL

Phép chia với NOT EXISTS

- Cú pháp tổng quát:

```
SELECT <Danh sách cột>
```

```
FROM <Tên bảng 1> B1
```

```
WHERE NOT EXISTS (SELECT *
```

```
    FROM <Tên bảng 2> B2
```

```
    WHERE NOT EXISTS (SELECT *
```

```
        FROM <Tên bảng 3> B3
```

```
        WHERE B3.<Thuộc tính> = B2.<Thuộc tính>
```

```
        AND B3.<Thuộc tính> = B1.<Thuộc tính>));
```

PHÉP CHIA TRONG SQL

Phép chia có điều kiện (Conditional Division)

- Ví dụ 7: Tìm khách hàng đã mua tất cả các sản phẩm thuộc danh mục 'Điện tử'.

```
SELECT KH.MaKH
```

```
FROM KhachHang KH
```

```
WHERE NOT EXISTS (SELECT *
```

```
    FROM SanPham SP
```

```
    WHERE SP.DanhMuc = 'Điện tử'
```

```
    AND NOT EXISTS (SELECT *
```

```
        FROM MuaHang MH
```

```
        WHERE KH.MaKH = MH.MaKH
```

```
        AND MH.MaSP = SP.MaSP));
```

PHÉP CHIA TRONG SQL

Phép chia có điều kiện (Conditional Division)

- Ví dụ 8: Tìm sinh viên đã đăng ký tất cả môn học bắt buộc có số tín chỉ lớn hơn 3.

```
SELECT SV.MaSV
```

```
FROM SinhVien SV
```

```
WHERE NOT EXISTS (SELECT *
```

```
    FROM MonHoc MH
```

```
    WHERE MH.SoTinChi > 3 AND NOT EXISTS (SELECT *
```

```
        FROM DangKy DK
```

```
        WHERE DK.MaMH = MH.MaMH
```

```
        AND DK.MaSV = SV.MaSV));
```

PHÉP CHIA TRONG SQL

Phép chia có điều kiện (Conditional Division)

- **Ví dụ 9:** Tìm nhân viên đã tham gia tất cả các dự án có thời gian hoàn thành dưới 6 tháng.

```
SELECT NV.MaNV  
FROM NhanVien NV  
WHERE NOT EXISTS (SELECT *  
                   FROM DuAn DA  
                   WHERE DA.ThoiGianHoanThanh < 6  
                   AND NOT EXISTS (SELECT *  
                                   FROM ThamGia TG  
                                   WHERE NV.MaNV = TG.MaNV AND  
                                         DA.MaDA = TG.MaDA));
```

PHÉP CHIA TRONG SQL

Các dấu hiệu nhận biết giữa ba loại phép chia

Loại phép chia	Dấu hiệu nhận biết
Phép chia cơ bản	<ul style="list-style-type: none">- GROUP BY và HAVING COUNT để đếm và so sánh số phần tử giữa các bảng.- Câu lệnh con sử dụng COUNT(DISTINCT) để đếm số lượng phần tử duy nhất.
Phép chia có điều kiện	<ul style="list-style-type: none">- Kết hợp điều kiện lọc trong WHERE với phép chia cơ bản.- Sử dụng GROUP BY và HAVING COUNT để kiểm tra số lượng phần tử khớp với điều kiện.
Phép chia với NOT EXISTS	<ul style="list-style-type: none">- Sử dụng hai lớp NOT EXISTS lồng nhau.- Kiểm tra sự tồn tại của các phần tử trong bảng phụ dựa trên điều kiện liên kết với bảng chính.- Không sử dụng GROUP BY.

GIỚI THIỆU NỘI DUNG THỰC HÀNH TUẦN 3

2

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

Chức năng các hàm tập hợp (Aggregate Functions)

Hàm	Mô tả	Ví dụ
COUNT(*)	Đếm tổng số hàng trong tập dữ liệu (kể cả giá trị NULL).	<code>SELECT COUNT(*) FROM NhanVien;</code>
COUNT(<thuộc tính>)	Đếm số giá trị khác NULL của thuộc tính	<code>SELECT COUNT(Luong) FROM NhanVien;</code>
COUNT(DISTINCT <thuộc tính>)	Đếm số giá trị khác nhau và khác NULL của thuộc tính	<code>SELECT COUNT(DISTINCT ChucVu) FROM NhanVien;</code>
SUM(x)	Tính tổng giá trị của cột x.	<code>SELECT SUM(Luong) FROM NhanVien;</code>
AVG(x)	Tính giá trị trung bình của cột x.	<code>SELECT AVG(Luong) FROM NhanVien;</code>
MAX(x)	Trả về giá trị lớn nhất của cột x.	<code>SELECT MAX(Luong) FROM NhanVien;</code>
MIN(x)	Trả về giá trị nhỏ nhất của cột x.	<code>SELECT MIN(Luong) FROM NhanVien;</code>

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

Ví dụ về truy vấn gom nhóm trong SQL

Tên SV	Điểm
Nam	9
Tuấn	9
Lộc	8,5
Linh	8
Trung	8
Minh	7,5
Đức	7

Bảng dữ liệu mẫu: SinhVien

- **Select Top 4:** Nam, Tuấn, Lộc, Linh.
- **Select Top 4 with ties:** Nam, Tuấn, Lộc, Linh, Trung.
- Sinh viên nằm trong 4 mức điểm cao nhất: Nam, Tuấn, Lộc, Linh, Trung, Minh

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

SELECT TOP

- Lệnh SELECT TOP được sử dụng để trả về một số lượng giới hạn các dòng dữ liệu từ bảng kết quả.

```
SELECT TOP <số lượng> <Danh sách các cột>  
FROM <Đơn vị>  
WHERE <Điều kiện>;
```

- Ví dụ: Trả về 4 dòng dữ liệu đầu tiên từ bảng NhanVien

```
SELECT TOP 4 HoTen, Luong  
FROM NhanVien;
```

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

SELECT TOP kết hợp với ORDER BY

- Khi kết hợp SELECT TOP với ORDER BY, có thể lấy các dòng dữ liệu xếp hạng theo một hoặc nhiều cột dựa trên thứ tự sắp xếp.

SELECT TOP <số lượng> <Đanh sách các cột>

FROM <Đanh sách các bảng>

WHERE <Điều kiện>

ORDER BY <Tên cột> (ASC | DESC);

- Ví dụ: Trả về 4 nhân viên có lương cao nhất từ bảng NhanVien

SELECT TOP 4 HoTen, Luong

FROM NhanVien

ORDER BY Luong DESC;

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

SELECT TOP WITH TIES

- Lệnh WITH TIES được sử dụng khi có nhiều dòng có giá trị bằng với dòng cuối cùng trong kết quả TOP.

```
SELECT TOP <số lượng> WITH TIES <Đanh sách các cột>
      FROM <Đanh sách các bảng>
      WHERE <Điều kiện>
      ORDER BY <Tên cột> (ASC | DESC);
```

- Ví dụ: Trả về 4 nhân viên có lương cao nhất. Nếu có nhiều nhân viên có cùng mức lương với vị trí thứ 4, sẽ lấy hết các nhân viên đó.

```
SELECT TOP 4 WITH TIES HoTen, Luong
      FROM NhanVien
      ORDER BY Luong DESC;
```

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

SELECT DISTINCT TOP

- DISTINCT kết hợp với TOP được sử dụng để lọc các giá trị trùng lặp trong kết quả và chỉ trả về các giá trị khác nhau.

```
SELECT DISTINCT TOP <số lượng> <Đanh sách các cột>
FROM <Đanh sách các bảng>
WHERE <Điều kiện>;
```

- Ví dụ: Trả về 4 chức vụ khác nhau đầu tiên từ bảng NhanVien

```
SELECT DISTINCT TOP 4 ChucVu
FROM NhanVien;
```

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

SELECT TOP kết hợp với GROUP BY và HAVING

```
SELECT TOP <số lượng> <Đanh sách các cột>
      FROM <Đanh sách các bảng>
      WHERE <Điều kiện>
      GROUP BY <Đanh sách các cột gom nhóm>
      HAVING <Điều kiện trên nhóm>
      ORDER BY <Tên cột> (ASC | DESC);
```

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

Chức năng các câu lệnh truy vấn gom nhóm trong SQL

Cú pháp	Mô tả	Ví dụ
SELECT TOP <số lượng>	Trả về số lượng dòng giới hạn từ bảng kết quả	SELECT TOP 4 * FROM NhanVien;
SELECT TOP <số lượng> ORDER BY <cột>	Trả về số lượng dòng giới hạn dựa trên thứ tự sắp xếp	SELECT TOP 4 * FROM NhanVien ORDER BY Luong DESC;
SELECT TOP <số lượng> WITH TIES	Trả về tất cả các dòng có giá trị bằng dòng cuối cùng trong kết quả TOP	SELECT TOP 4 WITH TIES * FROM NhanVien ORDER BY Luong DESC;
SELECT DISTINCT TOP <số lượng>	Trả về số lượng dòng khác nhau và không trùng lặp trong kết quả	SELECT DISTINCT TOP 4 ChucVu FROM NhanVien;
SELECT TOP <số lượng> GROUP BY <cột> HAVING <điều kiện>	Trả về số lượng dòng dựa trên các nhóm và điều kiện nhóm	SELECT TOP 4 MaPB, SUM(Luong) FROM NhanVien GROUP BY MaPB HAVING SUM(Luong) > 5000000;

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

Ví dụ tham khảo

- **Ví dụ:** Tìm mức lương cao nhất, thấp nhất, trung bình của mỗi phòng ban trong công ty

```
SELECT Phong, Max (Luong), Min (Luong), Avg (Luong)  
FROM NHANVIEN  
GROUP BY Phong;
```

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

Ví dụ tham khảo

- **Ví dụ:** Thống kê số lượng nhân viên của mỗi phòng. Lấy ra thông tin mã phòng, tên phòng, số lượng nhân viên.

```
SELECT PB.MaPH, TenPH, Count (MaNV) 'SLNV'  
FROM NHANVIEN NV, PHONGBAN PB  
WHERE NV.Phong = PB.MaPH  
GROUP BY MaPH, TenPH;
```

HÀM TÍNH TOÁN VÀ GOM NHÓM TRONG SQL

Ví dụ tham khảo

- **Ví dụ:** Tìm phòng ban có số lượng nhân viên lớn hơn 10. Lấy ra thông tin mã phòng, tên phòng, số lượng nhân viên. Sắp xếp theo số lượng nhân viên giảm dần.

```
SELECT PB.MaPH, TenPH, Count (MaNV) 'SLNV'  
FROM NHANVIEN NV, PHONGBAN PB  
WHERE NV.Phong = PB.MaPH  
GROUP BY MaPH, TenPH  
HAVING Count (MaNV) > 10  
ORDER BY Count (MaNV) DESC;
```

LỆNH TRUY VẤN SỬ DỤNG PHÉP KẾT NGOÀI

Câu lệnh truy vấn sử dụng phép kết ngoài

- Sử dụng kết ngoài trong một số trường hợp để **hạn chế** việc bỏ sót dữ liệu.
- Cách thức sử dụng tương tự inner join trong kết bằng.
 - Phép kết trái:**

Table1 left outer join Table2 on <điều kiện kết>

- Phép kết phải:**

Table1 right outer join Table2 on <điều kiện kết>

- Phép kết đầy đủ:**

Table1 full outer join Table2 on <điều kiện kết>

LỆNH TRUY VẤN SỬ DỤNG PHÉP KẾT NGOÀI

Ví dụ: Cho biết những mã nhân viên không tham gia đề án nào

```
SELECT NV.MaNV, HoTen  
FROM NHANVIEN NV left join PHANCONG PC  
on NV.MaNV = PC.MaNV  
WHERE MaDA IS NULL
```

GIỚI THIỆU NỘI DUNG THỰC HÀNH TUẦN 3

3

BÀI TẬP THỰC HÀNH VÀ HỎI ĐÁP

BÀI TẬP THỰC HÀNH VÀ HỎI ĐÁP

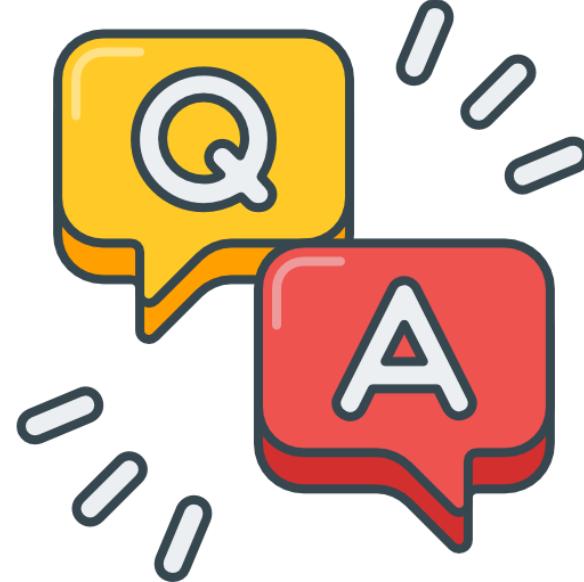
Yêu cầu: **(Xem thêm đề bài tại file Bài tập thực hành tuần 3)**

Phần bài làm lưu trữ trên một file script có tên **<MSV>_<HoVaTen>_BTTH3.sql** (trong đó **MSV** 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 III: Ngôn ngữ truy vấn dữ liệu**, bài tập **Quản lý bán hàng** từ câu 26 đến câu 53.



HỎI ĐÁP



Liên hệ hỗ trợ

Lê Võ Đình Kha - khalvd@uit.edu.vn