

--- CƠ SỞ DỮ LIỆU ---

Nội dung thực hành

Tuần 5

Truy vấn lồng

Tóm tắt nội dung bài thực hành:

Sử dụng cú pháp truy vấn lồng trong ngôn ngữ SQL.

Bộ môn **Hệ thống thông tin**

Khoa Công nghệ thông tin

ĐH Khoa học tự nhiên TP HCM



MỤC LỤC

1	Mục tiêu và tóm tắt nội dung.....	1
2	Hướng dẫn chi tiết.....	1
2.1	Giới thiệu truy vấn lồng.....	1
2.2	Các vị trí của câu truy vấn con:	2
2.2.1	Đặt tại mệnh đề SELECT:.....	2
2.2.2	Đặt tại mệnh đề FROM:	2
2.2.3	Đặt tại mệnh đề WHERE:.....	3
2.3	Truy vấn lồng phân cấp.....	3
2.3.1	Truy vấn lồng phân cấp với toán tử IN	3
2.3.2	Truy vấn lồng phân cấp với toán tử ALL.....	4
2.3.3	Truy vấn lồng phân cấp với toán tử ANY, SOME.....	5
2.4	Truy vấn lồng tương quan.....	6
2.4.1	Truy vấn lồng tương quan với EXISTS	6
2.4.2	Một số dạng khác của truy vấn lồng tương quan	7
2.5	Một câu truy vấn có thể giải bằng nhiều cách	8
2.6	Truy vấn lồng với các hàm kết hợp và gom nhóm	8
3	Bài tập	11

1 Mục tiêu và tóm tắt nội dung

Sau khi hoàn thành bài thực hành này sinh viên sẽ biết được:

- Khái niệm truy vấn lồng, truy vấn lồng phân cấp, truy vấn lồng tương quan
- Cách viết một câu truy vấn lồng phân cấp
- Sử dụng các toán tử IN, NOT IN, ALL, ANY, SOME
- Sử dụng các toán tử EXIST, NOT EXIST
- Truy vấn lồng phân cấp
- Truy vấn lồng tương quan.
- Truy vấn lồng kết hợp với gom nhóm

2 Hướng dẫn chi tiết

2.1 Giới thiệu truy vấn lồng

Định nghĩa: Truy vấn lồng là một câu truy vấn mà ở bên trong nội dung của nó có chứa một câu truy vấn con khác.

Cú pháp:

```
SELECT A  
  
FROM X  
  
WHERE ... ( SELECT B FROM Y WHERE ...) ...
```

Phân loại: Dựa vào đặc điểm của câu truy vấn con người ta phân truy vấn lồng thành 2 loại chính:

- *Truy vấn lồng phân cấp:* Khi nội dung của câu truy vấn con độc lập với câu truy vấn cha.

Cú pháp:

```
SELECT A  
  
FROM X  
  
WHERE ... (SELECT B, C FROM Y) ...
```

Ở ví dụ trên, câu truy vấn con **SELECT B FROM Y** không sử dụng bất kỳ thành phần nào của câu truy vấn cha. Do đó đây là một câu truy vấn lồng phân cấp.

- *Truy vấn lồng tương quan*: Khi nội dung của câu truy vấn con phụ thuộc vào câu truy vấn cha.

Cú pháp:

```
SELECT A  
  
FROM X  
  
WHERE ... (SELECT B, C FROM Y WHERE B = X.A) ...
```

Ở ví dụ này, câu truy vấn con **SELECT B, C FROM Y WHERE B = X.A** có sử dụng thành phần của câu truy vấn cha qua biểu thức so sánh **B = X.A**. Do đó, đây là một câu truy vấn lồng tương quan.

2.2 Các vị trí của câu truy vấn con:

Câu truy vấn con có thể nằm ở vị trí bất kỳ trong câu truy vấn cha. Câu truy vấn con có thể đặt tại mệnh đề **SELECT**, mệnh đề **FROM** hoặc thông thường nhất là ở mệnh đề **WHERE**.

2.2.1 Đặt tại mệnh đề SELECT:

Kết quả của câu truy vấn sẽ như là một giá trị của một thuộc tính.

Ví dụ 1: Với mỗi bộ môn, cho biết tên bộ môn và số lượng giáo viên của bộ môn đó.

```
SELECT BM.TENBM, ( SELECT COUNT(*)  
  
FROM GIAOVIEN GV  
  
WHERE GV.MABM = BM.MABM)  
  
FROM BOMON BM
```

2.2.2 Đặt tại mệnh đề FROM:

Kết quả của câu truy vấn sẽ xem như là một bảng dữ liệu, do vậy có thể truy vấn từ bảng dữ liệu này.

Ví dụ 2: Cho biết họ tên và lương của các giáo viên bộ môn HTTT

```
SELECT T.HOTEN, T.LNG
FROM (      SELECT MAGV, HOTEN, LUONG as LNG
            FROM GIAOVIEN
            WHERE MABM= 'HTTT') as T
```

2.2.3 Đặt tại mệnh đề WHERE:

Kết quả của câu truy vấn được sử dụng như một thành phần trong biểu thức điều kiện.

Ví dụ 3: Cho biết những giáo viên có lương lớn hơn lương của giáo viên có MAGV='001'

```
SELECT *
FROM GIAOVIEN
WHERE LUONG > (SELECT LUONG
               FROM GIAOVIEN
               WHERE MAGV= '001')
```

2.3 Truy vấn lồng phân cấp

2.3.1 Truy vấn lồng phân cấp với toán tử IN

Toán tử IN dùng để kiểm tra một giá trị có nằm trong một tập hợp nào đó hay không. Tập hợp đó có thể là kết quả của một câu truy vấn hoặc một tập hợp tường minh (xem lại chương 3 – Truy vấn cơ bản).

Cú pháp của biểu thức điều kiện: [THUỘC TÍNH] IN [TẬP HỢP]

```
...[THUỘC TÍNH] IN ( SELECT ...
                    FROM ...
                    WHERE ... )

Hoặc

...[THUỘC TÍNH] IN ( giá trị 1, giá trị 2, ..., giá trị n)
```

Chân trị của mệnh đề này là TRUE nếu giá trị của thuộc tính nằm trong tập hợp. Ngược lại mệnh đề sẽ cho chân trị FALSE.

Ví dụ 4: Cho biết họ tên những giáo viên mà không có một người thân nào.

```
SELECT HOTEN
FROM GIAOVIEN
WHERE MAGV NOT IN (SELECT MAGV FROM NGUOITHAN)
```

Ví dụ 5: Cho những giáo viên có tham gia đề tài

```
SELECT *
FROM GIAOVIEN
WHERE MAGV IN (SELECT MAGV FROM THAMGIADT)
```

2.3.2 Truy vấn lồng phân cấp với toán tử ALL

Toán tử ALL được sử dụng với các toán tử so sánh số học: >, <, >=, <=, ...

Cú pháp của biểu thức điều kiện:

[THUỘC TÍNH] [> | < | >= | <= | !=] ALL [TẬP HỢP]

Biểu thức điều kiện này cho chân trị đúng nếu giá trị của thuộc tính > | < | >= | <= | != mọi phần tử trong tập hợp.

Ví dụ 6: Cho những giáo viên có lương nhỏ nhất

```
SELECT *
FROM GIAOVIEN
WHERE LUONG <= ALL (SELECT LUONG FROM GIAOVIEN)
```

Ví dụ 7: Cho những giáo viên có lương cao hơn tất cả các giáo viên của bộ môn HTTT

```
SELECT *
FROM GIAOVIEN
```

```
WHERE LUONG>=ALL(SELECT LUONG
                    FROM GIAOVIEN
                    WHERE MABM= 'HTTT')
```

Ví dụ 8: Cho biết bộ môn (MABM) có đông giáo viên nhất

```
SELECT MABM
FROM GIAOVIEN
GROUP BY MABM
HAVING COUNT(*)>=ALL( SELECT COUNT(*)
                        FROM GIAOVIEN
                        GROUP BY MABM)
```

Ví dụ 9: Cho biết họ tên những giáo viên mà không có một người thân nào.
(Sử dụng ALL thay vì NOT IN)

```
SELECT HOTEN
FROM GIAOVIEN
GROUP BY MAGV != ALL ( SELECT MAGV
                        FROM NGUOITHAN)
```

2.3.3 Truy vấn lồng phân cấp với toán tử ANY, SOME

Cú pháp của sử dụng ANY và SOME tương tự với cú pháp sử dụng toán tử ALL.

ALL được sử dụng khi muốn giá trị của thuộc tính thỏa mãn với tất cả các phần tử trong tập hợp theo toán tử số học được sử dụng. Ngược lại là ANY: bất kỳ, SOME: một vài.

Ví dụ 10: Cho biết họ tên những giáo viên có tham gia đề tài. (Sử dụng = ANY thay vì IN)

```

SELECT HOTEN
FROM GIAOVIEN
WHERE MAGV = ANY (SELECT MAGV
                   FROM THAMGIADT)

```

2.4 Truy vấn lồng tương quan

2.4.1 Truy vấn lồng tương quan với EXISTS

Cú pháp sử dụng EXISTS:

- EXISTS thường được sử dụng trong biểu thức điều kiện:

```

... EXISTS (  SELECT ...
              FROM ...
              WHERE ...)

```

Chân trị của mệnh đề EXISTS là TRUE nếu kết quả của câu truy vấn con trả về từ một bộ trở lên. Ngược lại, nếu câu truy vấn con không trả ra dữ liệu thì mệnh đề EXISTS cho chân trị FALSE.

- EXISTS thường dùng trong câu truy vấn lồng tương quan.

Ví dụ 11: Cho biết các giáo viên có tham gia đề tài.

```

SELECT HOTEN
FROM GIAOVIEN GV
WHERE EXISTS (  SELECT *
                FROM THAMGIADT TG
                WHERE TG.MAGV = GV.MAGV)

```

Ví dụ 12: Cho biết các giáo viên không có người thân

```

SELECT HOTEN
FROM GIAOVIEN GV
WHERE NOT EXISTS ( SELECT *

```



```
FROM NGUOITHAN NT
WHERE NT.MAGV = GV.MAGV)
```

2.4.2 Một số dạng khác của truy vấn lồng tương quan

Ví dụ 14: Cho biết những giáo viên có lương lớn hơn lương trung bình của bộ môn mà giáo viên đó làm việc.

```
SELECT HOTEN
FROM GIAOVIEN GV1
WHERE GV1.LUONG > (
    SELECT AVG (GV2.LUONG)
    FROM GIAOVIEN GV2
    WHERE GV2.MABM = GV1.MABM)
```

Ví dụ 15: Cho biết những giáo viên có lương lớn nhất.

```
SELECT HOTEN
FROM GIAOVIEN GV1
WHERE (
    SELECT COUNT(*)
    FROM GIAOVIEN GV2
    WHERE GV2.LUONG > GV1.LUONG) = 0
```

Ví dụ 16: Cho biết những đề tài mà giáo viên '001' không tham gia.

```
SELECT MADT
FROM DETAI DT
WHERE NOT EXISTS ( SELECT *
    FROM THAMGIADT TG
    WHERE TG.MADT = DT.MADT AND MAGV= '001')
-----
SELECT MADT
```

```

FROM DETAI

WHERE MADT NOT IN (      SELECT MADT

                        FROM THAMGIADT

                        WHERE MAGV = '001')

```

2.5 Một câu truy vấn có thể giải bằng nhiều cách

Ví dụ 17: Cho biết họ tên những giáo viên có vai trò quản lý về mặt chuyên môn với các giáo viên khác.

```

-- C1: Sử dụng toán tử EXISTS

SELECT GV1.HOTEN
FROM GIAOVIEN GV1
WHERE EXISTS (      SELECT *

                    FROM GIAOVIEN GV2

                    WHERE GV2.GVQLCM = GV1.MAGV)

-- C2: Sử dụng toán tử IN

SELECT HOTEN
FROM GIAOVIEN
WHERE MAGV IN (      SELECT GVQLCM

                    FROM GIAOVIEN)

-- C3: Sử dụng phép kết thông thường, nhớ phải có từ khóa DISTINCT

SELECT DISTINCT GV2.HOTEN
FROM GIAOVIEN GV1, GIAOVIEN GV2
WHERE GV1.GVQLCM = GV2.MAGV

```

2.6 Truy vấn lồng với các hàm kết hợp và gom nhóm

Ví dụ 18: Cho biết những giáo viên có lương cao nhất

```

SELECT MABM

FROM GIAOVIEN

```

```

WHERE LUONG =(SELECT MAX(LUONG)
                FROM GIAOVIEN)

-----

SELECT MABM
FROM GIAOVIEN
WHERE LUONG >= ALL(SELECT LUONG
                   FROM GIAOVIEN)

```

Ví dụ 19: Cho biết những bộ môn (MABM) có đông giáo viên nhất.

```

SELECT MABM
FROM GIAOVIEN
GROUP BY MABM
HAVING COUNT(*) >= ALL ( SELECT COUNT(*)
                        FROM GIAOVIEN GV
                        GROUP BY GV.MABM)

-----

SELECT MABM
FROM GIAOVIEN
GROUP BY MABM
HAVING COUNT(*) = ( SELECT MAX (SLGV)
                   FROM ( SELECT COUNT(*) AS SLGV
                        FROM GIAOVIEN
                        GROUP BY MABM) AS T )

```

Ví dụ 20: Cho biết những tên bộ môn, họ tên của trưởng bộ môn và số lượng giáo viên của bộ môn có đông giáo viên nhất.

```

SELECT BM.TENBM, TBM.HOTEN, COUNT(*) AS SLGV
FROM GIAOVIEN GV, BOMON BM, GIAOVIEN TBM
WHERE GV.MABM = BM.MABM AND BM.TRUONGBM=TBM.MAGV
GROUP BY GV.MABM, TBM.HOTEN
HAVING COUNT(*) >= ALL (  SELECT COUNT(*)
                           FROM GIAOVIEN
                           GROUP BY MABM )

```

Ví dụ 21: Cho biết những giáo viên có lương lớn hơn mức lương trung bình của giáo viên bộ môn Hệ thống thông tin mà không trực thuộc bộ môn hệ thống thông tin.

```

SELECT      GV.HOTEN
FROM  GIAOVIEN GV, BOMON BM
WHERE      GV.MABM = BM.MABM AND
           BM.TENBM != N'Hệ thống thông tin' AND
           LUONG > (  SELECT AVG(GV.LUONG)
                     FROM GIAOVIEN GV, BOMON BM
                     WHERE GV.MABM=BM.MABM AND
                           BM.TENBM = N'Hệ thống thông tin' )

```

Ví dụ 22: Cho tên biết đề tài có đông giáo viên tham gia nhất.

```

SELECT      DT.MADT, DT.TENDT
FROM  DETAI      DT, THAMGIADT      TG
WHERE      DT.MADT = TG.MADT
GROUP BY      DT.MADT
HAVING      COUNT (DISTINCT MAGV) >= ALL
           (SELECT      COUNT(DISTINCT MAGV)
            FROM  THAMGIADT

```

3 Bài tập

Yêu cầu: Viết các câu truy vấn cho bài tập Quản lý đề tài bằng ngôn ngữ SQL.

Bài tập **Quản lý ĐỀ TÀI:**

- Q35. Cho biết mức lương cao nhất của các giảng viên.
- Q36. Cho biết những giáo viên có lương lớn nhất.
- Q37. Cho biết lương cao nhất trong bộ môn "HTTT".
- Q38. Cho biết tên giáo viên lớn tuổi nhất của bộ môn Hệ thống thông tin.
- Q39. Cho biết tên giáo viên nhỏ tuổi nhất khoa Công nghệ thông tin.
- Q40. Cho biết tên giáo viên và tên khoa của giáo viên có lương cao nhất.
- Q41. Cho biết những giáo viên có lương lớn nhất trong bộ môn của họ.
- Q42. Cho biết tên những đề tài mà giáo viên Nguyễn Hoài An chưa tham gia.
- Q43. Cho biết những đề tài mà giáo viên Nguyễn Hoài An chưa tham gia. Xuất ra tên đề tài, tên người chủ nhiệm đề tài.
- Q44. Cho biết tên những giáo viên khoa Công nghệ thông tin mà chưa tham gia đề tài nào.
- Q45. Tìm những giáo viên không tham gia bất kỳ đề tài nào
- Q46. Cho biết giáo viên có lương lớn hơn lương của giáo viên "Nguyễn Hoài An"
- Q47. Tìm những trưởng bộ môn tham gia tối thiểu 1 đề tài
- Q48. Tìm giáo viên trùng tên và cùng giới tính với giáo viên khác trong cùng bộ môn
- Q49. Tìm những giáo viên có lương lớn hơn lương của ít nhất một giáo viên bộ môn "Công nghệ phần mềm"
- Q50. Tìm những giáo viên có lương lớn hơn lương của tất cả giáo viên thuộc bộ môn "Hệ thống thông tin"
- Q51. Cho biết tên khoa có đông giáo viên nhất
- Q52. Cho biết họ tên giáo viên chủ nhiệm nhiều đề tài nhất
- Q53. Cho biết mã bộ môn có nhiều giáo viên nhất
- Q54. Cho biết tên giáo viên và tên bộ môn của giáo viên tham gia nhiều đề tài nhất.
- Q55. Cho biết tên giáo viên tham gia nhiều đề tài nhất của bộ môn HTTT.
- Q56. Cho biết tên giáo viên và tên bộ môn của giáo viên có nhiều người thân nhất.
- Q57. Cho biết tên trưởng bộ môn mà chủ nhiệm nhiều đề tài nhất.

HẾT

Bộ môn HTTT - Khoa CNTT - ĐH KHTN