

Trường Du lịch **ĐẠI HỌC HUẾ**

Học phần:

**Ngôn ngữ truy vấn có cấu trúc
(SQL - Structured Query Language)**

Giảng viên: ThS. Lê Văn Hòa
Khoa QLSK&CNTT – Trường Du lịch

HUHT School of Hospitality and Tourism

Trường Du lịch **ĐẠI HỌC HUẾ**

Chương 2:

NGÔN NGỮ THAO TÁC DỮ LIỆU

HUHT School of Hospitality and Tourism

Trường Du lịch **ĐẠI HỌC HUẾ**

Nội dung chính

1. Cú pháp câu lệnh SELECT
2. Câu lệnh SELECT đơn giản
3. Sắp xếp kết quả truy vấn
4. Các loại phép nối
5. Gom nhóm dữ liệu với GROUP BY
6. Truy vấn con
7. Phép toán UNION

HUHT School of Hospitality and Tourism 3

Trường Du lịch **ĐẠI HỌC HUẾ**

1. Cú pháp câu lệnh SELECT

```
SELECT [ALL|DISTINCT] [TOP (n) [PERCENT]]  
list_of_data_items  
[INTO name_of_new_table]  
FROM table(s)  
[WHERE search_conditions]  
[GROUP BY list_of_columns]  
[HAVING search_conditions]  
[ORDER BY list_of_sorted_columns]
```

- Tất cả các mệnh đề phải đúng thứ tự trong cú pháp.
- Mệnh đề **SELECT** và **FROM** bắt buộc phải có.
- Các mệnh đề còn lại tùy chọn.

HUHT School of Hospitality and Tourism 4

Trường Du lịch **ĐẠI HỌC HUẾ**

Kết quả truy vấn

- Mỗi câu lệnh SELECT tạo ra kết quả truy vấn là một bảng bao gồm:
 - Một hay nhiều cột (column).
 - Không hoặc nhiều dòng, bản ghi (row)

HUHT School of Hospitality and Tourism 5

Trường Du lịch **ĐẠI HỌC HUẾ**

Kết quả truy vấn

```
SELECT MaSinhVien, HoDem, Ten,
       NgaySinh
FROM SINHVIEN
```

masinhvien	hodem	ten	ngaysinh
DL01	Dương Minh	Hùng	1993-05-15
DL02	Lê Thị Thịnh	Vương	1992-08-25
DL03	Thị	Nở	1991-12-29
DL04	Văn Bá Thành	Nhân	1994-03-11
DL05	Du Ái	Nghĩa	1994-08-05
DL06	Văn	Cao	1991-12-25
DL07	Lệnh Hồ	Xung	1991-05-09

HUHT School of Hospitality and Tourism 6

Trường Du lịch **ĐẠI HỌC HUẾ**

2. Câu lệnh SELECT đơn giản

- Câu lệnh SELECT đơn giản nhất truy vấn dữ liệu từ duy nhất một bảng trong CSDL:
 - SELECT:** Danh sách các cột được hiển thị trong kết quả truy vấn.
 - FROM:** Bảng lưu trữ dữ liệu cần truy xuất
 - WHERE:** Điều kiện được sử dụng để lựa chọn dữ liệu từ bảng.

HUHT School of Hospitality and Tourism 7

Trường Du lịch **ĐẠI HỌC HUẾ**

2. Câu lệnh SELECT đơn giản

- Ví dụ:

```
SELECT MaSinhVien,
       HoDem, Ten,
       NgaySinh
FROM SINHVIEN
WHERE YEAR(NgaySinh) = '1991'
```

HUHT School of Hospitality and Tourism 8

Trường Du lịch **ĐẠI HỌC HUẾ**

Các mục danh sách chọn

- Mệnh đề SELECT chỉ định danh sách các mục được chọn và được phân cách bằng dấu phẩy “,”.
- Một mục được chọn có thể là một trong những mục sau:
 - Tên cột
 - Hằng số
 - Biểu thức

Ví dụ:

```

SELECT MaSinhVien,
      N'Họ và tên: ',
      CONCAT(HoDem, ' ', Ten) AS 'Họ và tên'
FROM SINHVIEN
  
```

9

Trường Du lịch **ĐẠI HỌC HUẾ**

Các mục danh sách chọn

- SQL cho phép sử dụng dấu hoa thị (*) thay cho “tất cả các cột” trong danh sách chọn:

```
SELECT * FROM SINHVIEN
```

- Chúng ta có thể đổi tên của bất kỳ biểu thức, cột nào trong danh sách chọn bằng cách sử dụng bí danh (Alias):

```

SELECT MaSinhVien AS 'Mã sinh viên',
       CONCAT(HoDem, ' ', Ten) AS 'Họ và tên',
       NgaySinh 'Ngày sinh'
  
```

FROM SINHVIEN

Mã sinh viên	Họ và tên	Ngày sinh
DL01	Dương Minh Hùng	1993-05-15
DL02	Lê Thị Hạnh	1992-08-25
DL03	Thị Nở	1991-12-29
DL04	Văn Bá Thành Nhân	1994-03-11

10

Trường Du lịch **ĐẠI HỌC HUẾ**

Các mục danh sách chọn

- Một truy vấn có thể bao gồm các cột được tính toán mà giá trị được tính từ một biểu thức SQL.

```

SELECT MaSinhVien AS 'Mã sinh viên',
       CONCAT(HoDem, ' ', Ten) AS 'Họ và tên',
       NgaySinh 'Ngày sinh'
  
```

FROM SINHVIEN

Biểu thức SQL

Mã sinh viên	Họ và tên	Ngày sinh
DL01	Dương Minh Hùng	1993-05-15
DL02	Lê Thị Hạnh	1992-08-25
DL03	Thị Nở	1991-12-29
DL04	Văn Bá Thành Nhân	1994-03-11

11

Trường Du lịch **ĐẠI HỌC HUẾ**

Các mục danh sách chọn

- Câu lệnh **CASE ... WHEN** được sử dụng trong danh sách chọn để kiểm tra danh sách các điều kiện và trả về một trong các biểu thức kết quả có thể có.
- Trường hợp 1:**

```

CASE input_expression
    WHEN when_expression1 THEN result1
    WHEN when_expression2 THEN result2
    [...n]
    [ELSE else_result]
END
  
```

12

Trường Du lịch **ĐẠI HỌC HUẾ**

Các mục danh sách chọn

Ví dụ:

```
SELECT MaSinhVien, HoDem, Ten,
CASE GioiTinh
    WHEN 1 THEN N'Nam'
    WHEN 0 THEN N'Nữ'
    ELSE N'Chưa nhập Dữ liệu'
END AS 'Giới tính'
FROM SINHVIEN
```

HUHT School of Hospitality and Tourism 13

Trường Du lịch **ĐẠI HỌC HUẾ**

Các mục danh sách chọn

- **Trường hợp 2:**

```
CASE
    WHEN boolean_expression1 THEN result1
    WHEN boolean_expression2 THEN result2
    [...]
    [ELSE else_result]
END
```

HUHT School of Hospitality and Tourism 14

Trường Du lịch **ĐẠI HỌC HUẾ**

Các mục danh sách chọn

- Ví dụ:

```
SELECT MaSinhVien, HoDem, Ten,
CASE
    WHEN GioiTinh = 1 THEN N'Nam'
    WHEN GioiTinh = 0 THEN N'Nữ'
    ELSE N'Chưa nhập Dữ liệu'
END AS 'Giới tính'
FROM SINHVIEN
```

HUHT School of Hospitality and Tourism 15

Trường Du lịch **ĐẠI HỌC HUẾ**

Loại bỏ các hàng trùng lặp

- Từ khóa **DISTINCT**, được viết trước danh sách chọn, được sử dụng để loại bỏ các bản ghi trùng lặp trong kết quả truy vấn.

<code>SELECT NoiSinh FROM SINHVIEN</code>	<code>SELECT DISTINCT NoiSinh FROM SINHVIEN</code>
NoiSinh TP.Huế Quảng Trị Huế Thừa Thiên Huế Thành Phố Huế Huế Quảng Trị Đà Nẵng TT Huế TT Huế Đà Nẵng Đà Nẵng Quảng Bình Hà Nội	NoiSinh Đà Nẵng Hà Nội Huế Quảng Bình Quảng Trị Thành Phố Huế Thừa Thiên Huế TP Huế Hà Nội Hà Nội

HUHT School of Hospitality and Tourism 16

Trường Du lịch **ĐẠI HỌC HUẾ**

Mệnh đề SELECT TOP

- Mệnh đề **SELECT TOP** giới hạn các bản ghi được trả về trong kết quả truy vấn để chỉ định một số bản ghi hoặc tỷ lệ phần trăm bản ghi.
- Ví dụ:**
Liệt kê 5 sinh viên đầu tiên

```
SELECT TOP 5 MaSinhVien, Hodem, Ten, NgaySinh
FROM SINHVIEN
```

 Hiển thị 5% số sinh viên từ bảng Sinh viên

```
SELECT TOP 5 PERCENT MaSinhVien, Hodem, Ten
FROM SINHVIEN
```
- Lưu ý:** Nên sử dụng mệnh đề **ORDER BY** kết hợp với mệnh đề TOP để có được câu lệnh truy vấn tốt nhất.

HUHT School of Hospitality and Tourism 17

Trường Du lịch **ĐẠI HỌC HUẾ**

Mệnh đề FROM

- Theo sau là tên của bảng chứa dữ liệu được truy xuất bởi truy vấn (được gọi là bảng nguồn của truy vấn)
- Bí danh (**Alias**) có thể được sử dụng để đổi tên bảng trong câu lệnh SELECT.
- Ví dụ:**

```
SELECT L.MaLop, TenLop,
COUNT(MaSinhVien) AS 'Tổng số SV'
FROM LOP L LEFT JOIN SINHVIEN AS S
ON L.MaLop = S.MaLop
GROUP BY L.MaLop, TenLop
```

S là bí danh
của bảng Sinh viên

HUHT School of Hospitality and Tourism 18

Trường Du lịch **ĐẠI HỌC HUẾ**

Mệnh đề WHERE

- Mệnh đề **WHERE** bao gồm từ khóa WHERE theo sau là điều kiện tìm kiếm để chỉ định các bản ghi sẽ được truy xuất.
- Điều kiện tìm kiếm có thể trả về một trong ba kết quả cho mỗi bản ghi trong kết quả truy vấn:
 - Nếu điều kiện tìm kiếm là **TRUE**, bản ghi sẽ được đưa vào kết quả truy vấn.
 - Nếu điều kiện tìm kiếm là **FALSE**, bản ghi sẽ bị loại trừ khỏi kết quả truy vấn.
 - Nếu điều kiện tìm kiếm là **NULL**, bản ghi sẽ bị loại trừ khỏi kết quả truy vấn.

HUHT School of Hospitality and Tourism 19

Trường Du lịch **ĐẠI HỌC HUẾ**

Mệnh đề WHERE

- Ví dụ:**

```
SELECT MaSinhVien, HoDem, Ten, NgaySinh
FROM SINHVIEN
WHERE YEAR(NgaySinh) = '1991'
```

Điều kiện tìm kiếm

masinhvien	hodem	ten	ngaysinh
DL03	Thị	Nở	1991-12-29
DL06	Văn	Cao	1991-12-25
DL07	Lê Hỗ	Xung	1991-05-09
KD03	Lê Thị Hồng	Nga	1991-07-24
TM04	Đỗ Thị Ngọc	Huyền	1991-03-29
TM07	Nguyễn	Tam	1991-06-25

HUHT School of Hospitality and Tourism 20

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Mệnh đề WHERE

SQL cung cấp năm loại điều kiện tìm kiếm:

- Kiểm tra so sánh:** So sánh giá trị của một biểu thức với giá trị của một biểu thức khác.
- Kiểm tra phạm vi:** Kiểm tra xem giá trị của một biểu thức có nằm trong phạm vi giá trị được chỉ định hay không.
- Kiểm tra thuộc tập giá trị:** Kiểm tra xem giá trị của một biểu thức có khớp với một hoặc một bộ giá trị hay không.
- Kiểm tra đối sánh mẫu:** Kiểm tra xem giá trị của biểu thức chứa dữ liệu chuỗi có khớp với mẫu được chỉ định hay không.
- Kiểm tra giá trị Null:** Kiểm tra xem một cột có giá trị NULL hay không.

HUHT School of Hospitality and Tourism 21

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Kiểm tra so sánh

- Trong kiểm tra so sánh, SQL tính toán và so sánh các giá trị của hai biểu thức.
- SQL cung cấp sáu cách khác nhau để so sánh hai biểu thức:

=	Bằng
<>	Không bằng
>	Lớn hơn
>=	Lớn hơn hoặc bằng
<	Nhỏ hơn
<=	Nhỏ hơn hoặc bằng

HUHT School of Hospitality and Tourism 22

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Kiểm tra so sánh

- Ví dụ:**

```
SELECT MaSinhVien, HoDem, Ten, NgaySinh
FROM SINHVIEN
WHERE MONTH(NgaySinh) < 7
```

Kiểm tra so sánh

HUHT School of Hospitality and Tourism 23

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Điều kiện tìm kiếm kết hợp

- Chúng ta có thể kết hợp các điều kiện tìm kiếm đơn giản để tạo thành những điều kiện phức tạp hơn.
- SQL cung cấp ba phép toán logic như sau:
 - AND:** Được sử dụng để kết hợp hai hay nhiều điều kiện tìm kiếm đều phải trả về kết quả TRUE.
 - OR:** Được sử dụng để kết hợp hai hay nhiều điều kiện tìm kiếm trong đó có ít nhất một điều kiện trả về kết quả TRUE.
 - NOT:** Sử dụng từ khóa này để chọn các bản ghi có điều kiện tìm kiếm trả về kết quả FALSE.

Ví dụ:

```
SELECT MaSinhVien, HoDem, Ten, NgaySinh
FROM SINHVIEN
WHERE MONTH(NgaySinh) < 7 AND YEAR(Ngaysinh)=1991
```

HUHT School of Hospitality and Tourism 24

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Điều kiện tìm kiếm kết hợp

A	B	A AND B	A OR B	NOT A
TRUE	TRUE	?	?	?
TRUE	FALSE	?	?	?
FALSE	TRUE	?	?	?
FALSE	FALSE	?	?	?

HUHT School of Hospitality and Tourism 25

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Điều kiện tìm kiếm kết hợp

A	B	A AND B	A OR B	NOT A
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE

HUHT School of Hospitality and Tourism 26

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Kiểm tra phạm vi

- SQL cung cấp phép toán **BETWEEN .. AND** để kiểm tra phạm vi trong điều kiện tìm kiếm.
- Kiểm tra phạm vi sử dụng để kiểm tra xem giá trị dữ liệu có nằm giữa (hoặc không nằm giữa) hai giá trị được chỉ định:
 - x **BETWEEN a AND b** → $(a \leq x \leq b)$
 - x **NOT BETWEEN a AND b** → $(x < a \text{ và } x > b)$

HUHT School of Hospitality and Tourism 27

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Kiểm tra phạm vi

- Ví dụ:

```
SELECT MaSinhVien, HoDem, Ten, NgaySinh
FROM SINHVIEN
WHERE MONTH(NgaySinh) BETWEEN 3 AND 7
```

masinhvien	hodem	ten	ngaysinh
DL01	Dương Minh	Hùng	1993-05-15
DL04	Văn Bá Thành	Nhân	1994-03-11
DL07	Lênh Hồ	Xung	1991-05-09
KD03	Lé Thị Hồng	Nga	1991-07-24
KD05	Nguyễn Dương	Thiện	1992-04-21
TM02	Võ Thị Thu	Hằng	1993-05-21
TM03	Trần Thị Khánh	Hòa	1992-06-24
TM04	Đỗ Thị Ngọc	Huyền	1991-03-29
TM07	Nguyễn	Tam	1991-06-25
TM08	Dư	Tuấn	1992-05-29

HUHT School of Hospitality and Tourism 28

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Kiểm tra thuộc tập giá trị

- Toán tử **IN** để kiểm tra xem một giá trị dữ liệu có khớp với một trong danh sách các giá trị được chỉ định hay không.
- Chúng ta cũng có thể kiểm tra xem một giá trị dữ liệu có nằm ngoài danh sách các giá trị được chỉ định hay không bằng cách sử dụng toán tử **NOT IN**.

– $X \text{ IN } (a, b, c) \rightarrow (x = a) \text{ hay } (x = b) \text{ hay } (x = c)$
– $X \text{ NOT IN } (a, b, c) \rightarrow (x \neq a) \text{ và } (x \neq b) \text{ và } (x \neq c)$

HUHT School of Hospitality and Tourism 29

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Kiểm tra thuộc tập giá trị

- Ví dụ:** Hai câu truy vấn sau là tương tự nhau

```
SELECT CONCAT(HoDem, ' ', Ten) AS 'Họ và tên',
NoiSinh
FROM SINHVIEN
WHERE NoiSinh = N'Quảng Trị' OR
NoiSinh = N'TT Huế' OR
NoiSinh = N'Hà Nội'
```

```
SELECT CONCAT(HoDem, ' ', Ten) AS 'Họ và tên',
NoiSinh
FROM SINHVIEN
WHERE NoiSinh IN (N'Quảng Trị', N'TT Huế', N'Hà
Nội')
```

HUHT School of Hospitality and Tourism 30

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Kiểm tra đối sánh mẫu

- Kiểm tra đối sánh mẫu (**LIKE**) sử dụng để kiểm tra xem giá trị dữ liệu trong một cột có khớp với mẫu được chỉ định hay không.
- Mẫu là một chuỗi có thể bao gồm một hoặc nhiều ký tự đại diện.

Ví dụ:

- Tên bắt đầu với chuỗi "Lê" ("Lê Văn Bình", "Lê Lợi",...)
- Tên có chứa chuỗi "Thị" ("Võ Thị Sáu", "Võ Thị Ngọc Nhi", "Thị Nở", "Lê Thị Thịnh Vượng",...)

HUHT School of Hospitality and Tourism 31

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

Kiểm tra đối sánh mẫu

Ký tự đại diện	Mô tả	Ví dụ
%	Đại diện cho không hoặc nhiều ký tự	LIKE N'Lê%' "Lê Văn Bình", "Lê"
_	Đại diện duy nhất một ký tự	LIKE 'K5_QTKD' "K5QTKD", "K59QTKD"
[]	Đại diện bất kỳ ký tự trong khoảng được chỉ định, [a-f] hoặc [a,b,c,d,e,f]	LIKE '[a-f]anh' "banh", "canh", "danh"
[^]	Đại diện bất kỳ ký tự không nằm trong khoảng được chỉ định, [a-f] hoặc [a,b,c,d,e,f]	LIKE '[^a-f]anh' "hanh", "lanh",....

HUHT School of Hospitality and Tourism 32

ĐẠI HỌC HUẾ

Trường Du lịch

Kiểm tra đối sánh mẫu

- Ví dụ:

```
SELECT HoDem, Ten, NoiSinh, NoiSinh
FROM SINHVIEN
WHERE HoDem LIKE N'Văn%' AND
      NoiSinh LIKE N'%Huế'
```

HoDem	Ten	NoiSinh
Văn Bá Thành	Nhân	Thừa Thiên Huế
Văn	Cao	Huế

HUHT School of Hospitality and Tourism 33

ĐẠI HỌC HUẾ

Trường Du lịch

Kiểm tra đối sánh mẫu

- Ví dụ:

```
SELECT HoDem, Ten, NoiSinh
FROM SINHVIEN
WHERE HoDem NOT LIKE N'%Thị%' AND
      HoDem NOT LIKE N'%Võ%' AND
      NoiSinh NOT LIKE N'%Huế'
```

HoDem	Ten	NoiSinh
Lênh Hö	Xung	Quảng Trị
Nguyễn Dương	Thiện	Đà Nẵng
Dương	Quá	Đà Nẵng
Nguyễn Du Ngọc	Ánh	Hà Nội
Nguyễn	Tam	Quảng Bình
Dư	Tuấn	Đà Nẵng

HUHT School of Hospitality and Tourism 34

ĐẠI HỌC HUẾ

Trường Du lịch

Kiểm tra giá trị NULL

- Chúng ta không thể kiểm tra giá trị **NULL** bằng các toán tử so sánh, chẳng hạn như =, <, >,...
- Thay vào đó, chúng ta sẽ phải sử dụng toán tử **IS NULL** và **IS NOT NULL**
- Cú pháp

```
WHERE expression IS [NOT] NULL
```

Ví dụ:

```
SELECT CONCAT(HoDem, ' ', Ten) AS 'Họ và
tên', NoiSinh
FROM SINHVIEN
WHERE NoiSinh IS NULL
```

HUHT School of Hospitality and Tourism 35

ĐẠI HỌC HUẾ

Trường Du lịch

3. Sắp xếp kết quả truy vấn

- Chúng ta có thể yêu cầu SQL sắp xếp kết quả của một truy vấn bằng cách sử dụng mệnh đề **ORDER BY** trong câu lệnh SELECT.
- Mệnh đề ORDER BY bao gồm từ khóa ORDER BY, sau đó là danh sách các biểu thức sắp xếp được phân tách bằng dấu phẩy.
- Theo mặc định, SQL sắp xếp dữ liệu theo trình tự tăng dần. Để yêu cầu sắp xếp theo trình tự giảm dần, chúng ta sử dụng từ khóa **DESC**.
- Lưu ý:** Chúng ta có thể sử dụng từ khóa **ASC** để chỉ định sắp xếp tăng dần, nhưng vì đó là trình tự sắp xếp mặc định nên từ khóa thường bị bỏ qua.

HUHT School of Hospitality and Tourism 36

Trường Du lịch **ĐẠI HỌC HUẾ**

3. Sắp xếp kết quả truy vấn

- Ví dụ:

```
SELECT TOP 5 CONCAT(HoDem, ' ', Ten) AS
    'Họ và tên', NgaySinh
FROM SINHVIEN
ORDER BY NgaySinh DESC
```

Họ và tên	ngaysinh
Dư Ái Nghĩa	1994-08-05
Văn Bá Thành Nhân	1994-03-11
Nguyễn Dư Ngọc Ánh	1994-02-26
Nguyễn Thị Huế	1993-08-05
Võ Thị Thu Hằng	1993-05-21

HUHT 37

Trường Du lịch **ĐẠI HỌC HUẾ**

3. Sắp xếp kết quả truy vấn

- Ví dụ:

```
SELECT TOP 6 CONCAT(HoDem, ' ', Ten) AS
    'Họ và tên', NgaySinh, DiemMon1,
    (DiemMon1 + DiemMon2 + DiemMon3)/3 AS DiemTB
FROM SINHVIEN S INNER JOIN DIEMTS D
    ON S.MaSinhVien = D.MaSinhVien
WHERE DiemMon1 > 7
ORDER BY DiemMon1 DESC, DiemTB DESC
```

Họ và tên	ngaysinh	DiemMon1	DiemTB
Võ Văn Hậu	1992-09-21	9.4	9.266666
Dương Minh Hùng	1993-05-15	9.1	8.900000
Đỗ Thị Ngọc Huyền	1991-03-29	7.9	9.033333
Văn Bá Thành Nhân	1994-03-11	7.8	7.700000
Nguyễn Dương Thị Hiền	1992-04-21	7.8	5.233333
Lê Thị Thanh Hằng	1992-08-25	7.5	7.500000

HUHT 38

Trường Du lịch **ĐẠI HỌC HUẾ**

4. Các loại phép nối

- Mệnh đề **JOIN** được sử dụng để kết hợp các bản ghi từ hai hoặc nhiều bảng trong CSDL.
- Phép nối cho phép chúng ta có thể truy xuất dữ liệu từ hai hoặc nhiều bảng dựa trên mối quan hệ logic giữa các bảng.
- Các phép nối cho biết cách SQL nên sử dụng dữ liệu từ một bảng để chọn các bản ghi trong bảng khác.

HUHT School of Hospitality and Tourism 39

Trường Du lịch **ĐẠI HỌC HUẾ**

4. Các loại phép nối

Products table

ProductID	ProductName	CategoryID	UnitPrice
1	Chai	1	18
2	Aniseed Syrup	2	10
3	Chef Anton's Cajun Seasoning	2	22
4	Chef Anton's Gumbo Mix	2	21.35
5	Teatime Chocolate Biscuits	4	9.2
6	Sir Rodney's Marmalade	4	81
7	Sir Rodney's Scones	4	10
8	Chang	1	19

Categories table

CategoryID	CategoryName
1	Beverages
2	Condiments
3	Dairy Products
4	Confections

Products JOIN Categories ON Products.CategoryID = Categories.CategoryID

ProductID	ProductName	CategoryID	UnitPrice	CategoryID	CategoryName
1	Chai	1	18	1	Beverages
2	Aniseed Syrup	2	10	1	Beverages
3	Chef Anton's Cajun Seasoning	2	22	2	Condiments
4	Chef Anton's Gumbo Mix	2	21.35	2	Condiments
5	Teatime Chocolate Biscuits	4	9.2	4	Confections
6	Sir Rodney's Marmalade	4	81	4	Confections
7	Sir Rodney's Scones	4	10	4	Confections
8	Chang	1	19	1	Beverages

HUHT School of Hospitality and Tourism 40

Trường Du lịch **ĐẠI HỌC HUẾ**

4. Các loại phép nối

- Cú pháp:

```

SELECT *          Các bảng tham gia
FROM LOPSINHVIEN AS L
INNER JOIN        Từ khóa JOIN
Kiểu kết       SINHVIEN AS S
nối           ON L.MaLopSV = S.MaLopSV
                           Điều kiện nối
  
```

HUHT School of Hospitality and Tourism 41

Trường Du lịch **ĐẠI HỌC HUẾ**

4. Các loại phép nối

- Điều kiện nối, được viết sau từ khóa **ON**, xác định cách hai bảng có liên quan với nhau thực hiện kết nối trong một truy vấn:
 - Chỉ định cột từ mỗi bảng sẽ được sử dụng cho phép nối. Một điều kiện kết hợp diễn hình chỉ định một khóa ngoại từ một bảng và khóa liên quan của nó trong bảng khác.
 - Chỉ định một toán tử logic (=, <>, <,> =, <=,...) được sử dụng để so sánh các giá trị từ các cột.
- Lưu ý:** Tên cột bị trùng giữa hai hoặc nhiều bảng trong truy vấn thi phải chỉ định rõ ràng theo đúng cú pháp:
Tên bảng.Tên cột

HUHT School of Hospitality and Tourism 42

Trường Du lịch **ĐẠI HỌC HUẾ**

4. Các loại phép nối

- Ví dụ:

```

SELECT MaSinhVien, CONCAT(HoDem, ' ', Ten) AS
      'Họ và tên', NgaySinh, L.MaLop, TenLop
FROM SINHVIEN S INNER JOIN LOP L
      ON S.MaLop = L.MaLop
WHERE YEAR(NgaySinh) = 1991

SELECT L.MaLop, TenLop
FROM SINHVIEN S RIGHT JOIN LOP L
      ON S.MaLop = L.MaLop
GROUP By L.MaLop, TenLop
HAVING COUNT(MaSinhVien) = 0
  
```

HUHT School of Hospitality and Tourism 43

Trường Du lịch **ĐẠI HỌC HUẾ**

4. Các loại phép nối

- Các kiểu phép nối:
 - [INNER] JOIN
 - LEFT [OUTER] JOIN
 - RIGHT [OUTER] JOIN
 - FULL [OUTER] JOIN
 - CROSS JOIN (or CARTESIAN JOIN)
 - SELF JOIN

INNER JOIN	LEFT JOIN	RIGHT JOIN	FULL OUTER JOIN
table1 table2	table1 table2	table1 table2	table1 table2

HUHT School of Hospitality and Tourism 44

Trường Du lịch

4. Các loại phép nối

- [INNER] JOIN**

INNER JOIN

Customers table

CustomerID	Name	Address
1	Dall Gerlack	005 Sage Plaza
2	Ambrose Aldren	63 Dwight Hill
3	Benton Strzaker	1 Oakridge Street
4	Myrtle Creagh	1 Hanover Trail
5	Isis Beecham	74586 Graceland Way

Orders table

OrderID	OrderDate	CustomerID	Amount
11	2018-01-01	5	1000
12	2018-02-12	2	2000
13	2018-03-05	2	3000
14	2018-04-11	0	5000
15	2018-04-15	0	1500

Customers JOIN Orders
ON Customer.CustomerID = Orders.CustomerID

CustomerID	Name	Address	OrderID	OrderDate	CustomerID	Amount
2	Ambrose Aldren	63 Dwight Hill	12	2018-02-12	2	2000
2	Ambrose Aldren	63 Dwight Hill	13	2018-03-05	2	3000
5	Isis Beecham	74586 Graceland Way	11	2018-01-01	5	1000

HUH 45

Trường Du lịch

4. Các loại phép nối

- LEFT [OUTER] JOIN**

LEFT JOIN

Customers table

CustomerID	Name	Address
1	Dall Gerlack	005 Sage Plaza
2	Ambrose Aldren	63 Dwight Hill
3	Benton Strzaker	1 Oakridge Street
4	Myrtle Creagh	1 Hanover Trail
5	Isis Beecham	74586 Graceland Way

Orders table

OrderID	OrderDate	CustomerID	Amount
11	2018-01-01	5	1000
12	2018-02-12	2	2000
13	2018-03-05	2	3000
14	2018-04-11	0	5000
15	2018-04-15	0	1500

Customers LEFT JOIN Orders
ON Customer.CustomerID = Orders.CustomerID

CustomerID	Name	Address	OrderID	OrderDate	CustomerID	Amount
1	Dall Gerlack	005 Sage Plaza	NULL	NULL	NULL	NULL
2	Ambrose Aldren	63 Dwight Hill	12	2018-02-12	2	2000
2	Ambrose Aldren	63 Dwight Hill	13	2018-03-05	2	3000
3	Benton Strzaker	1 Oakridge Street	NULL	NULL	NULL	NULL
4	Myrtle Creagh	1 Hanover Trail	NULL	NULL	NULL	NULL
5	Isis Beecham	74586 Graceland Way	11	2018-01-01	5	1000

HUH 46

Trường Du lịch

4. Các loại phép nối

- RIGHT [OUTER] JOIN**

RIGHT JOIN

Customers table

CustomerID	Name	Address
1	Dall Gerlack	005 Sage Plaza
2	Ambrose Aldren	63 Dwight Hill
3	Benton Strzaker	1 Oakridge Street
4	Myrtle Creagh	1 Hanover Trail
5	Isis Beecham	74586 Graceland Way

Orders table

OrderID	OrderDate	CustomerID	Amount
11	2018-01-01	5	1000
12	2018-02-12	2	2000
13	2018-03-05	2	3000
14	2018-04-11	0	5000
15	2018-04-15	0	1500

Customers RIGHT JOIN Orders
ON Customer.CustomerID = Orders.CustomerID

CustomerID	Name	Address	OrderID	OrderDate	CustomerID	Amount
5	Isis Beecham	74586 Graceland Way	11	2018-01-01	5	1000
2	Ambrose Aldren	63 Dwight Hill	12	2018-02-12	2	2000
2	Ambrose Aldren	63 Dwight Hill	13	2018-03-05	2	3000
NULL	NULL	NULL	14	2018-04-11	0	5000
NULL	NULL	NULL	15	2018-04-15	0	1500

HUH 47

Trường Du lịch

4. Các loại phép nối

- FULL [OUTER] JOIN**

FULL OUTER JOIN

Customers table

CustomerID	Name	Address
1	Dall Gerlack	005 Sage Plaza
2	Ambrose Aldren	63 Dwight Hill
3	Benton Strzaker	1 Oakridge Street
4	Myrtle Creagh	1 Hanover Trail
5	Isis Beecham	74586 Graceland Way

Orders table

OrderID	OrderDate	CustomerID	Amount
11	2018-01-01	5	1000
12	2018-02-12	2	2000
13	2018-03-05	2	3000
14	2018-04-11	0	5000
15	2018-04-15	0	1500

Customers FULL JOIN Orders
ON Customer.CustomerID = Orders.CustomerID

CustomerID	Name	Address	OrderID	OrderDate	CustomerID	Amount
1	Dall Gerlack	005 Sage Plaza	NULL	NULL	NULL	NULL
2	Ambrose Aldren	63 Dwight Hill	12	2018-02-12	2	2000
2	Ambrose Aldren	63 Dwight Hill	13	2018-03-05	2	3000
3	Benton Strzaker	1 Oakridge Street	NULL	NULL	NULL	NULL
4	Myrtle Creagh	1 Hanover Trail	NULL	NULL	NULL	NULL
5	Isis Beecham	74586 Graceland Way	11	2018-01-01	5	1000
NULL	NULL	NULL	14	2018-04-11	0	5000
NULL	NULL	NULL	15	2018-04-15	0	1500

HUH 48

ĐẠI HỌC HUẾ

Trường Du lịch

4. Các loại phép nối

- CROSS JOIN

CustomerID	Name	Address
1	Dall Gefack	005 Sage Plaza
2	Ambrose Aldren	63 Dwight Hill
3	Benton Strazaker	1 Oakridge Street
4	Myrtle Creagh	1 Hanover Trail
5	Iesi Beecham	74586 Graceland Way

Customers table

CustomerID	Name	Address
1	Dall Gefack	005 Sage Plaza
2	Ambrose Aldren	63 Dwight Hill
3	Benton Strazaker	1 Oakridge Street
4	Myrtle Creagh	1 Hanover Trail
5	Iesi Beecham	74586 Graceland Way

Customers CROSS JOIN Orders

OrderID	OrderDate	CustomerID	Amount
11	2018-01-01	5	1000
12	2018-02-12	2	2000
13	2018-03-05	2	3000
14	2018-04-11	0	5000
15	2018-04-15	0	1500

Orders table

OrderID	OrderDate	CustomerID	Amount
1	2018-01-01	5	1000
2	2018-02-12	2	2000
3	2018-03-05	2	3000
4	2018-04-11	0	5000
5	2018-04-15	0	1500

49

ĐẠI HỌC HUẾ

Trường Du lịch

4. Các loại phép nối

- SELF JOIN

– Self JOIN được sử dụng để nối một bảng với chính nó.

– Chúng ta phải sử dụng bí danh để đổi tên ít nhất một bảng trong câu lệnh SQL.

- Ví dụ:

```
SELECT *  
FROM Orders AS o1 JOIN Orders as o2  
ON o1.Amount > o2.Amount
```

HUHT School of Hospitality and Tourism 50

ĐẠI HỌC HUẾ

Trường Du lịch

5. Gom nhóm dữ liệu với GROUP BY

Mỗi quốc gia có bao nhiêu người?
Tuổi trung bình của mỗi quốc gia?
Tuổi thấp nhất và cao nhất của mỗi quốc gia?

FullName	Country	Age
Walliw Ziems	England	28
Eli Altimas	USA	30
Luce Tumility	England	25
Manuel Sharpie	Canada	26
Rania Borsi	England	27
Libby Sturley	USA	29
Marella Honywill	Canada	30
Anthe Bolzen	England	29
Harold Edson	USA	28
Roseanne Dodge	USA	26

SELECT Country,
COUNT(*) AS 'Count of People',
AVG(Age) AS 'Average Age',
MIN(Age) AS 'Min Age',
MAX(Age) AS 'Max Age'
FROM People
GROUP BY Country

Country	Count of People	Average Age	Min Age	Max Age
Canada	2	28	26	30
England	4	27	25	29
USA	4	28	26	30

HI

ĐẠI HỌC HUẾ

Trường Du lịch

5. Gom nhóm dữ liệu với GROUP BY

- Truy vấn gom nhóm hoạt động như thế nào?

FullName	Country	Age
Walliw Ziems	England	28
Eli Altimas	USA	30
Luce Tumility	England	25
Manuel Sharpie	Canada	26
Rania Borsi	England	27
Libby Sturley	USA	29
Marella Honywill	Canada	30
Anthe Bolzen	England	29
Harold Edson	USA	28
Roseanne Dodge	USA	26

Tuổi trung bình của mỗi quốc gia?

SELECT Country,
AVG(Age) 'Average Age'
FROM People
GROUP BY Country

Country	Average Age
Canada	28
England	27
USA	28

AVG (Age)

HUH 52

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

- Hàm gộp (Aggregate Function):**
 - SQL cho phép chúng ta tóm tắt dữ liệu từ CSDL thông qua một tập các hàm gộp.
 - Một hàm gộp lấy toàn bộ cột dữ liệu làm đối số của nó và tạo ra một giá trị duy nhất tóm tắt cột.

Ví dụ:

```
SELECT L.Malop, TenLop,
       COUNT(MaSinhVien) AS 'Tổng số SV'
  FROM LOP L LEFT JOIN SINHVIEN S
    ON L.Malop = S.Malop
 GROUP BY L.Malop, TenLop
```

Malop	TenLop	Tổng số SV
K45HDL	Lớp K45HDL	6
K45QTKD	Lớp K45QTKD	0
K45TMBT	Lớp K45TMBT	5
K46HDL	Lớp K46HDL	0
K46QTKD	Lớp K46QTKD	5

HUHT School of Hospitality and Tourism 53

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

CÚ PHÁP	Ý NGHĨA
SUM([DISTINCT] expr)	Hàm trả về tổng các giá trị của cột, biểu thức kiểu số. Tùy chọn DISTINCT có thể được sử dụng để tính tổng các giá trị phân biệt. Hàm trả về giá trị NULL nếu không có bản ghi thỏa mãn điều kiện (nếu có).
AVG([DISTINCT] expr)	Hàm trả về giá trị trung bình của cột hay biểu thức kiểu số trong các bản ghi. Tùy chọn DISTINCT có thể được sử dụng để tính trung bình của các giá trị phân biệt. Hàm trả về giá trị NULL nếu không có bản ghi thỏa mãn điều kiện (nếu có).

HUHT School of Hospitality and Tourism 54

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

CÚ PHÁP	Ý NGHĨA
COUNT([DISTINCT] expr)	Hàm trả về số lượng bản ghi của cột, biểu thức có giá trị khác rỗng. Tùy chọn DISTINCT có thể được sử dụng để đếm các giá trị phân biệt. Hàm trả về giá trị 0 nếu không có bản ghi thỏa mãn điều kiện (nếu có).
COUNT(*)	Hàm trả về số lượng bản ghi được truy xuất trong câu lệnh SELECT, hàm này chấp nhận giá trị rỗng (NULL). Hàm trả về giá trị NULL nếu không có bản ghi thỏa mãn điều kiện (nếu có).

HUHT School of Hospitality and Tourism 55

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

CÚ PHÁP	Ý NGHĨA
MAX(expr)	Hàm trả về giá trị lớn nhất của cột hay biểu thức. Hàm trả về giá trị NULL nếu không có bản ghi thỏa mãn điều kiện (nếu có).
MIN(expr)	Hàm trả về giá trị nhỏ nhất của cột hay biểu thức. Hàm trả về giá trị NULL nếu không có bản ghi thỏa mãn điều kiện (nếu có).

HUHT School of Hospitality and Tourism 56

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

- Mệnh đề GROUP BY
 - Mệnh đề **GROUP BY** thường được sử dụng với các hàm gộp (**COUNT, MAX, MIN, SUM, AVG**) để nhóm các kết quả truy vấn theo một hoặc nhiều cột.
 - Truy vấn chứa mệnh đề GROUP BY được gọi là truy vấn gom nhóm vì nó gom nhóm dữ liệu từ các bảng nguồn của nó và tạo ra một bản ghi tóm tắt duy nhất cho mỗi nhóm.
 - Các cột có tên trong mệnh đề GROUP BY được gọi là các cột gom nhóm của truy vấn, vì chúng xác định cách các bản ghi được chia thành các nhóm.

HUHT School of Hospitality and Tourism 57

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

- Ví dụ:

```
SELECT L.MaLop, TenLop,
       COUNT(MaSinhVien) AS 'Tổng số SV'
  FROM LOP L LEFT JOIN SINHVIEN S
    ON L.MaLop = S.MaLop
 GROUP BY L.MaLop, TenLop
```

MaLop	TenLop	Tổng số SV
K45HDDL	Lớp K45HDDL	6
K45QTKD	Lớp K45QTKD	0
K45TMĐT	Lớp K45TMĐT	5
K46HDDL	Lớp K46HDDL	0
K46QTKD	Lớp K46QTKD	5

HUHT 58

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

Một số lưu ý đối với các truy vấn gom nhóm

- Trong mệnh đề SELECT có thể là:
 - Hằng số
 - Một hàm gộp, tạo ra một giá trị duy nhất tóm tắt các bản ghi trong nhóm.
 - Cột gom nhóm, theo định nghĩa có cùng giá trị trong mọi bản ghi của nhóm.
- Tất cả các cột xuất hiện trong danh sách chọn (ngoại trừ các cột xuất hiện dưới dạng đối số của các hàm gộp) phải xuất hiện sau mệnh đề GROUP BY.

HUHT School of Hospitality and Tourism 59

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

Giới hạn kết quả với mệnh đề HAVING

- Mệnh đề **WHERE** chỉ định điều kiện trên các cột đã chọn nhưng không cho phép sử dụng các hàm gộp (**COUNT, MAX, MIN, SUM, AVG**).
- Mệnh đề **HAVING** cho phép chúng ta chỉ định các điều kiện lọc dữ liệu gom nhóm khi trong mệnh đề sử dụng các hàm gộp (**COUNT, MAX, MIN, SUM, AVG**).

HUHT School of Hospitality and Tourism 60

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

- Ví dụ:

```
SELECT L.MaLop, TenLop,
       COUNT(MaSinhVien) AS 'Tổng số SV'
  FROM LOP L LEFT JOIN SINHVIEN S
    ON L.MaLop = S.MaLop
 GROUP BY L.MaLop, TenLop
 HAVING COUNT(MaSinhVien) >= 5
```

MaLop	TenLop	Tổng số SV
K45HDDL	Lớp K45HDDL	7
K45TMDT	Lớp K45TMDT	8
K46QTKD	Lớp K46QTKD	6

HUHT School of Hospitality and Tourism 61

Trường Du lịch **ĐẠI HỌC HUẾ**

5. Gom nhóm dữ liệu với GROUP BY

- Ví dụ:

```
SELECT L.MaLop, TenLop,
       COUNT(MaSinhVien) 'Tổng số SV'
  FROM LOP L LEFT JOIN SINHVIEN S
    ON L.MaLop = S.MaLop
 WHERE L.MaLop LIKE '%HDDL'
 GROUP BY L.MaLop, TenLop
 HAVING COUNT(MaSinhVien) >= 5
```

MaLop	TenLop	Tổng số SV
K45HDDL	Lớp K45HDDL	7

HUHT School of Hospitality and Tourism 62

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Truy vấn con là truy vấn được lồng bên trong câu lệnh SELECT, INSERT, UPDATE, DELETE, hoặc bên trong một truy vấn con khác.
- Kết quả của truy vấn con được sử dụng để xác định kết quả của truy vấn cấp cao hơn có chứa truy vấn con.

Ví dụ:

```
SELECT S.MaSinhVien, HoDem, Ten, DiemMon1
  FROM SINHVIEN S INNER JOIN DIEMTS D
    ON S.MaSinhVien = D.MaSinhVien
 WHERE DiemMon1 > (SELECT AVG(DiemMon1)
                      FROM DIEMTS)
```

Truy vấn cha

Truy vấn con

HUHT School of Hospitality and Tourism 63

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

Ví dụ:

```
SELECT S.MaSinhVien, HoDem, Ten, DiemMon1
  FROM SINHVIEN S INNER JOIN DIEMTS D
    ON S.MaSinhVien = D.MaSinhVien
 WHERE DiemMon1 > (SELECT AVG(DiemMon1)
                      FROM DIEMTS)
```

Truy vấn cha

Truy vấn con

MaSinhVien	HoDem	Ten	DiemMon1
DL01	Dương Minh	Hùng	9.1
DL02	Lê Thanh	Vương	7.5
DL04	Văn Bá Thành	Nhân	7.8
KD01	Nguyễn Thị	Huệ	7.4
KD03	Lê Thị Hồng	Nga	7.4
KD05	Nguyễn Dương	Thiện	7.8
TM01	Võ Văn	Hậu	9.4
TM04	Đỗ Thị Ngọc	Huyền	7.9
TM05	Nguyễn Dư Ng...	Ánh	7.1

HUHT School of Hospitality and Tourism 64

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Truy vấn con luôn được đặt trong dấu ngoặc đơn và nó có thể có các thành phần sau:
 - Mệnh đề bắt buộc: SELECT, FROM
 - Mệnh đề tùy chọn: WHERE, GROUP BY, HAVING, ORDER BY.

Ví dụ:

```
SELECT S.MaSinhVien, HoDem, Ten, DiemMon1
FROM SINHVIEN S INNER JOIN DIEMTS D
ON S.MaSinhVien = D.MaSinhVien
WHERE DiemMon1 = (SELECT MAX(DiemMon1)
                   FROM DIEMTS)
```

HUHT School of Hospitality and Tourism 65

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Điều kiện tìm kiếm truy vấn con:**
 - Kiểm tra so sánh trong truy vấn con:** So sánh giá trị của một biểu thức với một giá trị duy nhất được tạo ra bởi một truy vấn con.
 - Kiểm tra thuộc tập giá trị truy vấn con:** Kiểm tra xem giá trị của một biểu thức có khớp với một trong các tập giá trị được tạo bởi truy vấn con hay không.
 - Kiểm tra sự tồn tại:** Kiểm tra xem một truy vấn con có tạo ra bất kỳ bản ghi nào trong kết quả truy vấn không.
 - Kiểm tra so sánh định lượng:** So sánh giá trị của một biểu thức với từng bộ giá trị được tạo ra bởi một truy vấn con.

HUHT School of Hospitality and Tourism 66

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- CSDL Quản lý bán hàng**

```

    erDiagram
        class OrderDetails {
            string OrderDetailId
            string OrderId
            string ProductId
            number Quantity
        }
        class Products {
            string ProductId
            string ProductName
            string SupplierId
            string CategoryId
            number Unit
            number Price
        }
        class Customers {
            string CustomerId
            string CustomerName
            string ContactName
            string Address
            string City
            string PostalCode
            string Country
        }
        class Orders {
            string OrderId
            string CustomerId
            string EmployeeId
            string OrderDate
            string ShipperId
        }
        class Suppliers {
            string SupplierId
            string SupplierName
            string ContactName
            string Address
            string City
            string PostalCode
            string Country
            string Phone
        }
        class Categories {
            string CategoryId
            string CategoryName
            string Description
        }
        class Employees {
            string EmployeeId
            string LastName
            string FirstName
            string BirthDate
            string Photo
            string Notes
        }
        class Shippers {
            string ShipperId
            string ShipperName
            string Phone
        }

        OrderDetails }o--o{ Products : "N"
        OrderDetails }o--o{ Customers : "N"
        OrderDetails }o--o{ Orders : "N"
        Products }o--o{ Suppliers : "N"
        Products }o--o{ Categories : "N"
        Customers }o--o{ Orders : "N"
        Employees }o--o{ Orders : "N"
        Employees }o--o{ Shippers : "N"
        Categories }o--o{ Employees : "N"
        Shippers }o--o{ Employees : "N"
    
```

HUHT School of Hospitality and Tourism 67

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Kiểm tra so sánh truy vấn con:** Truy vấn con trong kiểm tra so sánh (=; >; <; >=, <=, <>) phải tạo ra một giá trị duy nhất với kiểu dữ liệu thích hợp (Kết quả truy vấn con phải tạo ra một bản ghi chứa chính xác một cột).

Ví dụ: Liệt kê tất cả các nhà cung cấp (Suppliers) có cùng quốc gia với nhân viên (Customers) có ID nhân viên (CustomerID) là 1.

Suppliers
SupplierID SupplierName ContactName Address City PostalCode Country Phone

Customers
CustomerID CustomerName ContactName Address City PostalCode Country Phone

HUHT School of Hospitality and Tourism 68

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Ví dụ:** Liệt kê tất cả các nhà cung cấp (Suppliers) có cùng quốc gia với nhân viên (Customers) có ID nhân viên (CustomerID) là 1.

```
SELECT * FROM Suppliers
WHERE Country = (SELECT Country
                  FROM Customers
                  WHERE CustomerID = 1)
```

Suppliers	
SupplierID	SupplierName
ContactName	
Address	
City	
PostalCode	
Country	
Phone	

Customers	
CustomerID	CustomerName
ContactName	
Address	
City	
PostalCode	
Country	

H School of Hospitality and Tourism 69

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Kiểm tra thuộc tập giá trị (IN, NOT IN):** Nếu truy vấn con tạo ra 0 hoặc nhiều bản ghi và có chính xác một cột, truy vấn con này có thể được sử dụng làm danh sách trong kiểm tra thuộc tập giá trị.
- Ví dụ:** Liệt kê tất cả các khách hàng (Customers) có đơn đặt hàng (Orders) vào tháng 7 năm 2016, ngày đặt hàng dựa vào cột OrderDate.

Customers	
CustomerID	CustomerName
ContactName	
Address	
City	
PostalCode	
Country	

Orders	
OrderID	CustomerID
EmployeeID	
OrderDate	
ShipperID	

HUHT School of Hospitality and Tourism 70

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Ví dụ:** Liệt kê tất cả các khách hàng (Customers) có đơn đặt hàng (Orders) vào tháng 7 năm 2016, ngày đặt hàng dựa vào cột OrderDate.

```
SELECT * FROM Customers
WHERE CustomerID IN (SELECT CustomerID
                      FROM Orders
                      WHERE OrderDate LIKE '2016-07-%')
```

Customers	
CustomerID	CustomerName
ContactName	
Address	
City	
PostalCode	
Country	

Orders	
OrderID	CustomerID
EmployeeID	
OrderDate	
ShipperID	

HUHT School of Hos 71

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Kiểm tra sự tồn tại:** Kiểm tra xem một truy vấn có tạo ra bất kỳ bản ghi nào trong kết quả truy vấn không.
- Ví dụ:** Liệt kê các sản phẩm có tồn tại ít nhất một đơn đặt hàng với số lượng (Quantity) lớn hơn 100.

```

classDiagram
    class Products {
        ProductID
        ProductName
        SupplierID
        CategoryID
        Unit
        Price
    }
    class OrderDetails {
        OrderDetailID
        OrderID
        ProductID
        Quantity
    }
    Products "1" --> "M" OrderDetails
  
```

Products	
ProductID	ProductName
SupplierID	
CategoryID	
Unit	
Price	

OrderDetails	
OrderDetailID	OrderID
ProductID	
Quantity	

HUHT School of Hospitality and Tourism 72

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Ví dụ:** Liệt kê các sản phẩm có tồn tại ít nhất một đơn đặt hàng với số lượng (Quantity) lớn hơn 100.

```
SELECT *  
FROM Products p  
WHERE EXISTS (SELECT *  
               FROM OrderDetails  
               WHERE ProductID = p.ProductID  
                 AND Quantity > 100)
```

HUHT School of Hospitality and Tourism 73

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Kiểm tra định lượng với từ khóa ANY:**
 - Kiểm tra **ANY** được sử dụng kết hợp với một trong sáu toán tử so sánh trong SQL để so sánh một giá trị đơn với một cột giá trị do truy vấn con tạo ra.
 - Để thực hiện kiểm tra, SQL sử dụng toán tử so sánh được chỉ định để so sánh giá trị kiểm tra với từng giá trị trong cột, từng giá trị một. Nếu bất kỳ phép so sánh riêng lẻ nào trả về kết quả TRUE, thì phép kiểm tra ANY trả về kết quả TRUE.

HUHT School of Hospitality and Tourism 74

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Kiểm tra định lượng với từ khóa ANY:**
- Ví dụ:** Liệt kê tất cả khách hàng (Customers) có cùng quốc gia với bất kỳ nhà cung cấp (Suppliers) nào.

HUHT School of Hospitality and Tourism 75

Trường Du lịch **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Ví dụ:** Liệt kê tất cả khách hàng (Customers) có cùng quốc gia với bất kỳ nhà cung cấp (Suppliers) nào.

```
SELECT * FROM Customers  
WHERE Country = ANY (SELECT Country  
                      FROM Suppliers)
```

HUHT School of Hospitality and Tourism 76

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

6. Truy vấn con

Kiểm tra định lượng với từ khóa ALL:

- Kiểm tra **ALL** được sử dụng cùng với một trong sáu toán tử so sánh trong SQL để so sánh một giá trị kiểm tra đơn với một cột giá trị do truy vấn con tạo ra.
- Để thực hiện kiểm tra, SQL sử dụng toán tử so sánh được chỉ định để so sánh giá trị kiểm tra với từng giá trị trong cột, từng giá trị một. Nếu tất cả các phép so sánh riêng lẻ trả về kết quả TRUE, thì phép kiểm tra ALL sẽ trả về kết quả TRUE.

HUHT School of Hospitality and Tourism 77

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

6. Truy vấn con

Kiểm tra định lượng với từ khóa ALL:

- Ví dụ:** Hiển thị các nhà cung cấp có số lượng sản phẩm lớn nhất.

```
SELECT p.SupplierID,
       supplierName,
       COUNT(p.ProductID) AS CountOfProducts
FROM Suppliers AS s
JOIN Products AS p
      ON s.SupplierID = p.SupplierID
GROUP BY p.SupplierID, supplierName
HAVING COUNT(p.ProductID)
      >= ALL(SELECT Count(ProductID)
              FROM Products GROUP BY SupplierID)
```

HUHT School of Hospitality and Tourism 78

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Một truy vấn con có thể được sử dụng trong mệnh đề FROM của truy vấn bên ngoài.

```
SELECT t1.CategoryID, t1.CategoryName,
       t1.Count_Of_Products,
       t2.Avg_of_UnitPrice
FROM (
    SELECT c.CategoryID, c.CategoryName,
           COUNT(p.ProductID) AS Count_of_Products
    FROM Categories AS c
    JOIN Products AS p ON c.CategoryID = p.CategoryID
    GROUP BY c.CategoryID, c.CategoryName
) AS t1
JOIN (
    SELECT CategoryID,
           AVG(UnitPrice) AS Avg_of_UnitPrice
    FROM Products
    GROUP BY CategoryID
) AS t2 ON t1.CategoryID = t2.CategoryID
```

HUHT School of Hospitality and Tourism 79

 **Trường Du lịch** **ĐẠI HỌC HUẾ**

6. Truy vấn con

- Truy vấn con dưới dạng biểu thức:** Nếu truy vấn con tạo ra chính xác một cột và không hoặc một bản ghi, nó có thể được sử dụng ở bất kỳ nơi nào cho phép một biểu thức.

Ví dụ:

```
SELECT p.ProductID,
       p.ProductName,
       p.CategoryID,
       (
           SELECT CategoryName
           FROM Categories
           WHERE CategoryID = p.CategoryID
       ) AS CategoryName
FROM Products AS p
```

HUHT School of Hospitality and Tourism 80



Trường Du lịch **ĐẠI HỌC HUẾ**

Phép toán UNION

- **Union** là toán tử sử dụng để ghép kết quả của hai hay nhiều câu truy vấn dạng SELECT lại với nhau thành một tập kết quả duy nhất.
- **Cú pháp:**
*SELECT ...
UNION
SELECT ...
[UNION
SELECT ...]*

HUHT School of Hospitality and Tourism 81



Trường Du lịch **ĐẠI HỌC HUẾ**

Phép toán UNION

- **Ví dụ:**

```
SELECT MaSinhVien AS 'Mã sinh viên',
CONCAT(HoDem, ' ', Ten) AS 'Họ và tên',
NgaySinh 'Ngày sinh'
FROM SINHVIEN
WHERE MaLop LIKE 'K43%'
UNION
SELECT MaSinhVien AS 'Mã sinh viên',
CONCAT(HoDem, ' ', Ten) AS 'Họ và tên',
NgaySinh 'Ngày sinh'
FROM SINHVIEN HO LE
```

HUHT School of Hospitality and Tourism 82