TỔNG QUAN VỀ KHOÁ HỌC

GIỚI THIỆU VỀ SQL

gitiho

TỔNG QUAN VỀ KHOÁ HỌC

gitiho

1 MỤC TIÊU

Sau khi hoàn thành khóa học SQL thành công, học viên sẽ đạt được các kỹ năng cơ bản cần thiết để có thể sử dụng được SQL. Cụ thể:

- Hiểu các nguyên tắc của Cơ sở dữ liệu quan hệ và Ngôn ngữ truy vấn dữ liệu có cấu trúc (SQL).
- Sử dụng câu lệnh Transact-SQL để cấu trúc và tùy chỉnh cơ sở dữ liệu (trong máy chủ MS SQL) cũng như để lưu trữ, truy xuất và thao tác dữ liệu trong cơ sở dữ liệu quan hệ.
- Kết nối SQL trên các công cụ trực quan hóa dữ liệu (BI và các công cụ báo cáo).

gitiho

2

YÊU CẦU ĐỐI VỚI HỌC VIÊN

Bằng cách tham gia khóa học SQL, học viên hiểu các điều khoản của chúng tôi và các điều kiện và đồng ý với:

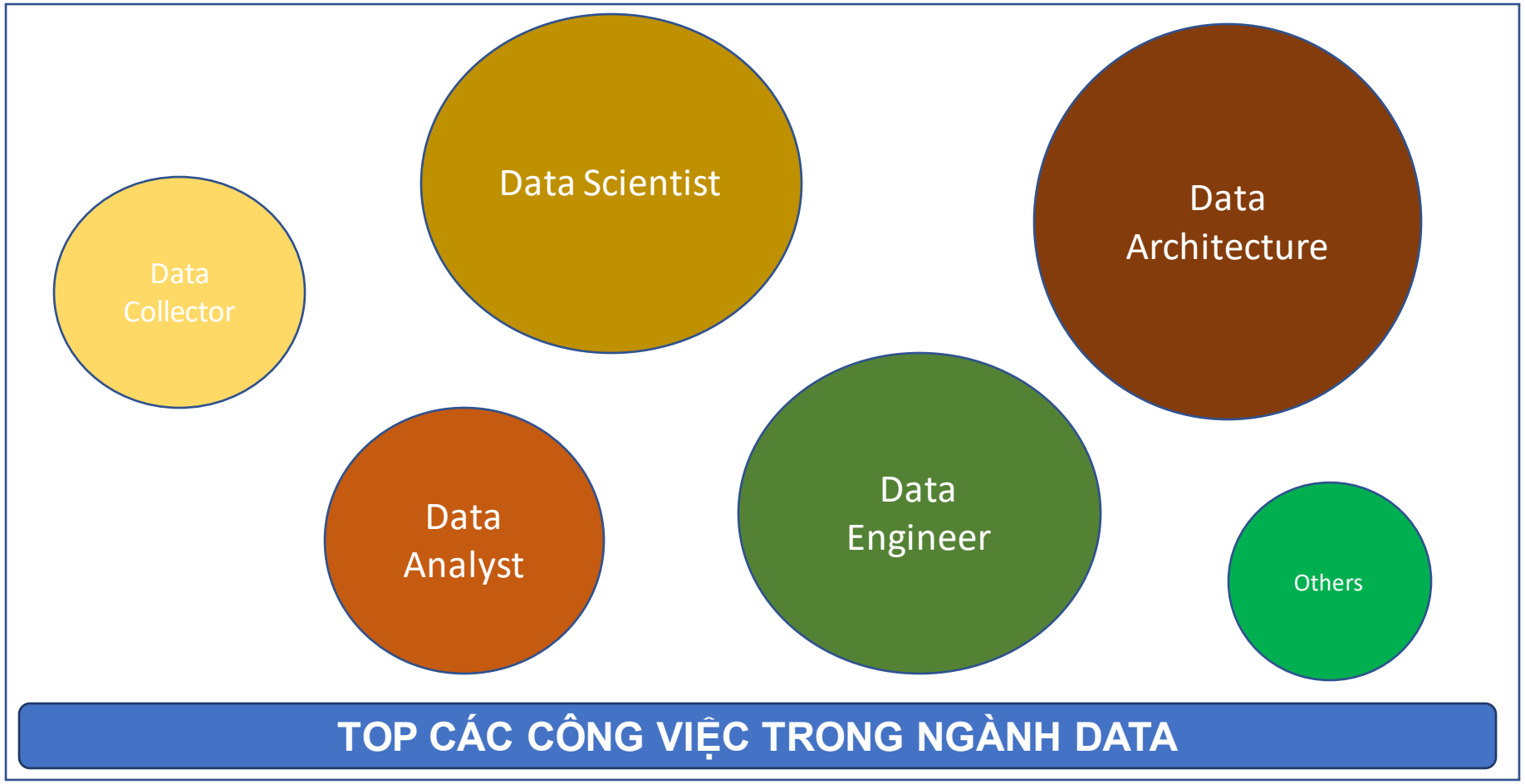
- Tham gia đầy đủ các giờ học
- Chú ý
- Làm bài tập thực hành

gitiho

GIỚI THIỆU VỀ SQL

gitiho

1 TẠI SAO NÊN HỌC SQL?



gitiho

2 SQL LÀ GÌ?

- SQL viết tắt của Structured Query Language, là ngôn ngữ truy vấn có cấu trúc, cho phép bạn truy cập và thao tác với các cơ sở dữ liệu để tạo, xóa, sửa đổi, trích xuất dữ liệu.
- SQL cũng là ngôn ngữ tiêu chuẩn cho các hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản trị cơ sở dữ liệu (RDBMS) như MySQL, MS Access, Oracle, Sybase, Informix, Postgres hay SQL Server đều lấy SQL làm ngôn ngữ cơ sở dữ liệu tiêu chuẩn.
- SQL là ngôn ngữ cơ sở dữ liệu, được sử dụng để tạo, xóa trong cơ sở dữ liệu, lấy các hàng và sửa đổi các hàng,...

Microsoft® SQL Server®

gitiho

2 SQL LÀ GÌ?

TÓM LẠI:

- SQL là ngôn ngữ truy vấn dữ liệu.
- SQL được thiết kế để quản lý dữ liệu trong một Hệ quản trị CSDL.
- Ví dụ: có thể dùng SQL để được sử dụng để truy xuất, tạo, thêm, xóa, sửa dữ liệu trong database,…

gitiho

3 HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Hệ quản trị CSDL là một hệ thống gồm một CSDL và các thao tác trên CSDL. Đó là hệ thống chương trình, công cụ cho phép quản lý và tương tác với CSDL.

Trên đó người dùng có thể định nghĩa, thao tác, và xử lí dữ liệu trong một CSDL để xuất ra những thông tin cần dùng.

Một số Hệ quản trị CSDL thường dùng như:

- **PostgreSQL**
- **MS SQL Server**
- **MySQL**
- **Oracle...**

Đều sử dụng SQL như là ngôn ngữ cơ sở dữ liệu chuẩn.

gitiho

4 ỨNG DỤNG CỦA SQL

- Xây dựng cơ sở dữ liệu thay cho việc lưu trữ nhiều file excel, csv trong nhiều folder. Không giới hạn số dòng như excel (1,04 triệu dòng). Lưu trữ trực quan.
- Sử dụng, gọi dữ liệu nhanh chóng, không phải mở file excel. Tốc độ cao và tận dụng được sức mạnh tối đa phần cứng.
- Phân tích dữ liệu (level cơ bản) với tốc độ cao.
- Có tương đối đầy đủ các hàm như trên excel …

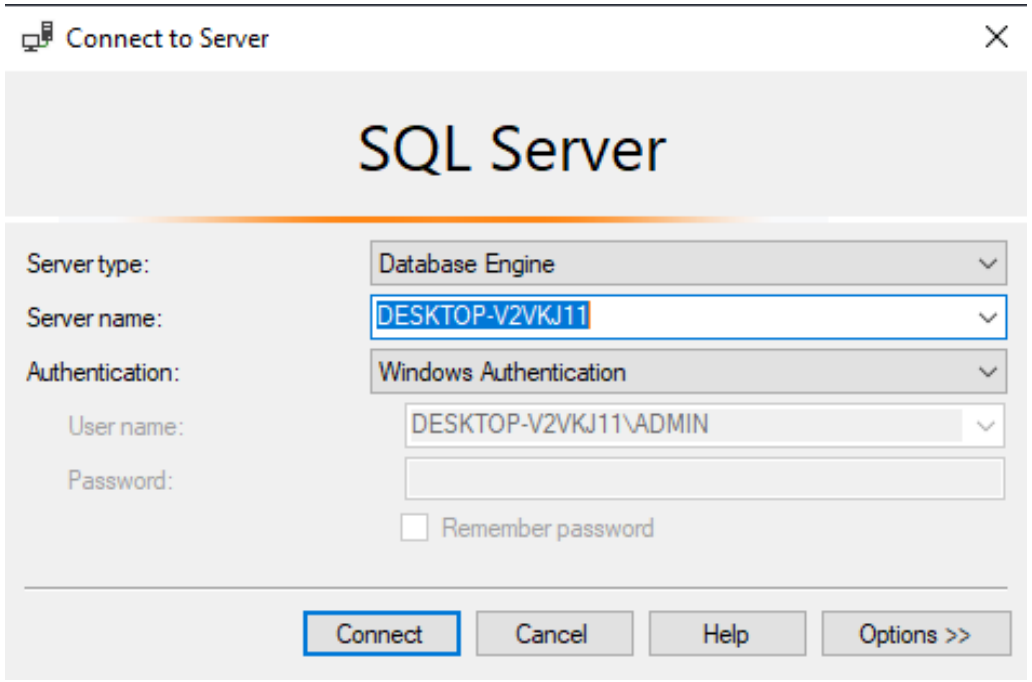
gitiho

5 GIỚI THIỆU VỀ GIAO DIỆN SQL SERVER

Sau khi đã tải SQL Server 2019 và SSMS

Lần đầu tiên truy cập SSMS sẽ có giao diện đăng nhập dựa trên tên My Computer như hình

Bước 1: => Click Connect để tiếp tục làm việc với SQL Server

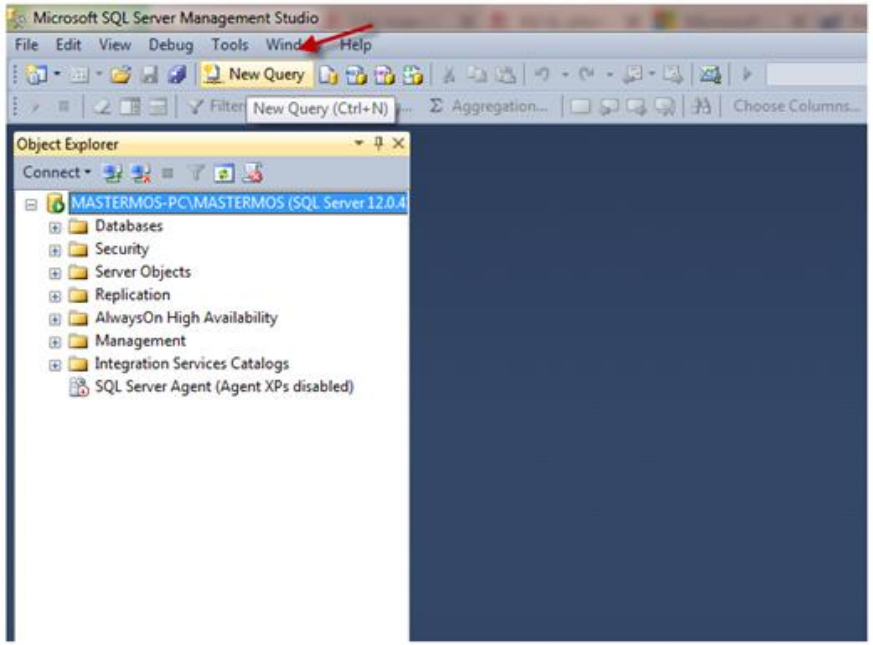


gitiho

5 GIỚI THIỆU VỀ GIAO DIỆN SQL SERVER

Bước 2: Sau khi đăng nhập thành công

Ấn vào nút **New Query** trên màn hình giao diện.



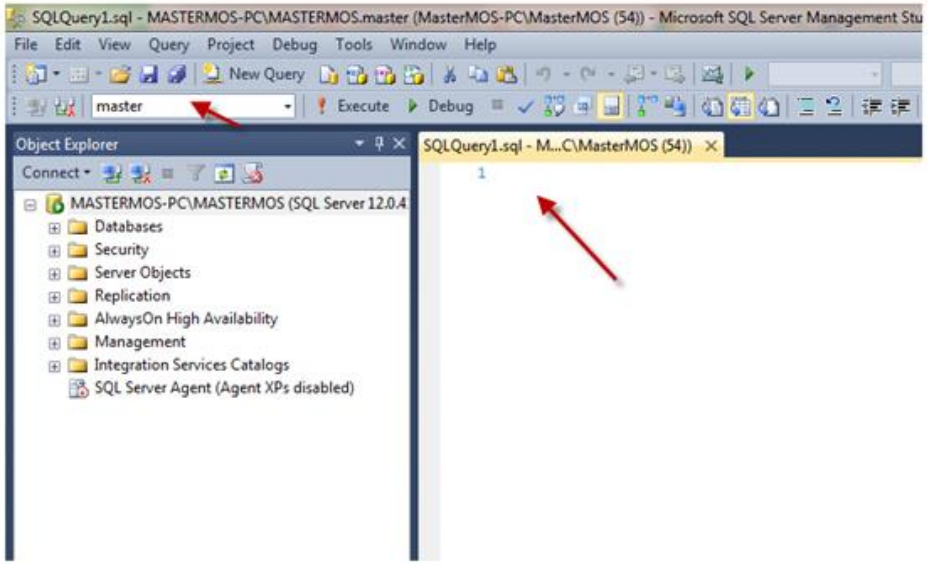
gitiho

5 GIỚI THIỆU VỀ GIAO DIỆN SQL SERVER

Bước 3: Sau khi ấn **New Query**

Một màn hình màu trắng hiện ra ở bên phải, trông giống như màn hình Microsoft Word, đây chính là nơi cho phép chúng ta viết các câu lệnh SQL để xử lý dữ liệu. Phần master ở phía trên gần thanh thực đơn chính là thể hiện nơi để chúng ta lựa chọn các Database, và Database mặc định khi mở SQL là master.

Ở phần này chúng ta cũng có thể chọn Database HocSQL mà chúng ta đã tạo ra theo hướng dẫn trong slide thực hành câu lệnh SQL để xử lý, phân tích dữ liệu.



gitiho

5 GIỚI THIỆU VỀ GIAO DIỆN SQL SERVER

Tổng quan giao diện làm việc:

1 – Thanh thực đơn chứa các tác vụ hay dùng để mở file SQL, lưu file SQL, các thao tác quản trị SQL Server

– Nút Execute, Stop, các nút để thao tác với câu lệnh SQL

2 – Phần bên trái là SQL Browser, nơi chứa sơ đồ hình cây các thành phần trong SQL Server từ mức SQL Server –> Database –> Table –> Column/Rows.

3 – Màn hình bên phải phía trên là nơi giúp chúng ta viết hàng nghìn câu lệnh SQL để tính toán, tổng hợp dữ liệu

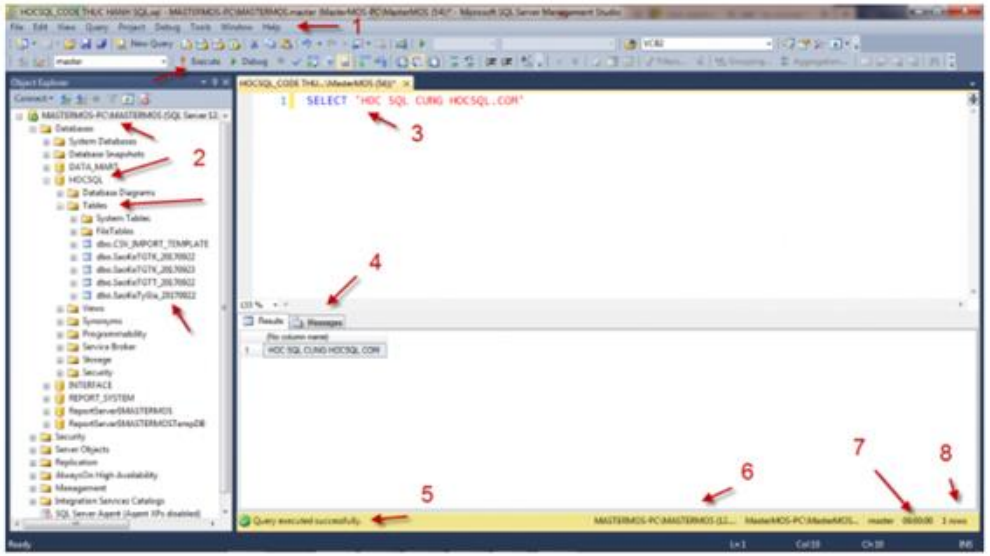
4 – Sau khi chạy câu lệnh SQL thì kết quả chạy câu lệnh sẽ hiển ở màn hình phía dưới bên phải. Kết quả câu lệnh có thể là dữ li sau khi tính toán, sau khi select hoặc là một câu thông báo việc chạy code SQL đã hoàn thành. Chúng ta sẽ xem chi tiết ở các bà sau.

5 – Câu lệnh chạy thành công sẽ có thông báo màu xanh "Query executed successfully", nếu không sẽ có thông báo lỗi ở đây.

6 – Tên máy tính và tên SQL Server

7 – Thời gian thực thi câu lệnh SQL

8 – Số dòng kết quả của câu lệnh SQL Select.



gitiho

6 CÁC ĐỐI TƯỢNG CHÍNH TRONG CSDL

Schema

a. Khái niệm:
Schema là một khái niệm mới được đưa vào SQL Server từ phiên bản 2005, dùng để gom nhóm các table có chung một đặc điểm nào đó để dễ dàng quản lý.
Nếu không sử dụng schema trong CSDL thì nó sẽ lấy schema mặc định là dbo.

b. Tính chất:
Trong một database thì tên của schema là duy nhất

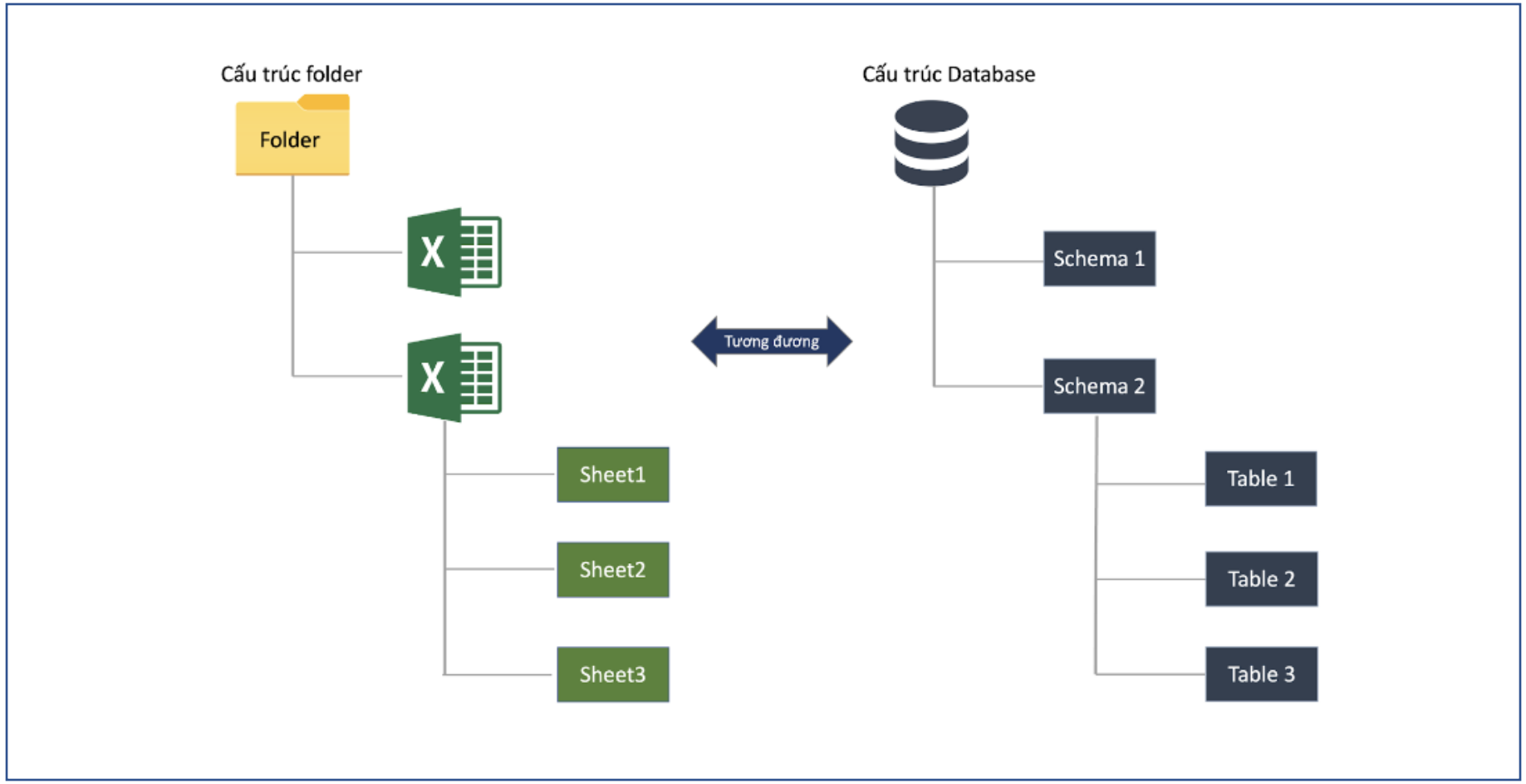
c. Ưu điểm:
Khi bạn phân nhóm các table lại thì sẽ rất dễ dàng quản lý, và bạn có thể phân quyền quản lý từng schema cho từng user khác nhau, đây chính là điểm mạnh của schema.

Ví dụ:
Trong lược đồ CSDL của bạn có hai loại table chính như sau:

- Các table về tin tức -> mình sẽ tạo schema tên là news gồm những table liên quan đến tin tức
- Các table hệ thống -> mình sẽ tạo schema tên là sys gồm những table liên quan đến hệ thống.

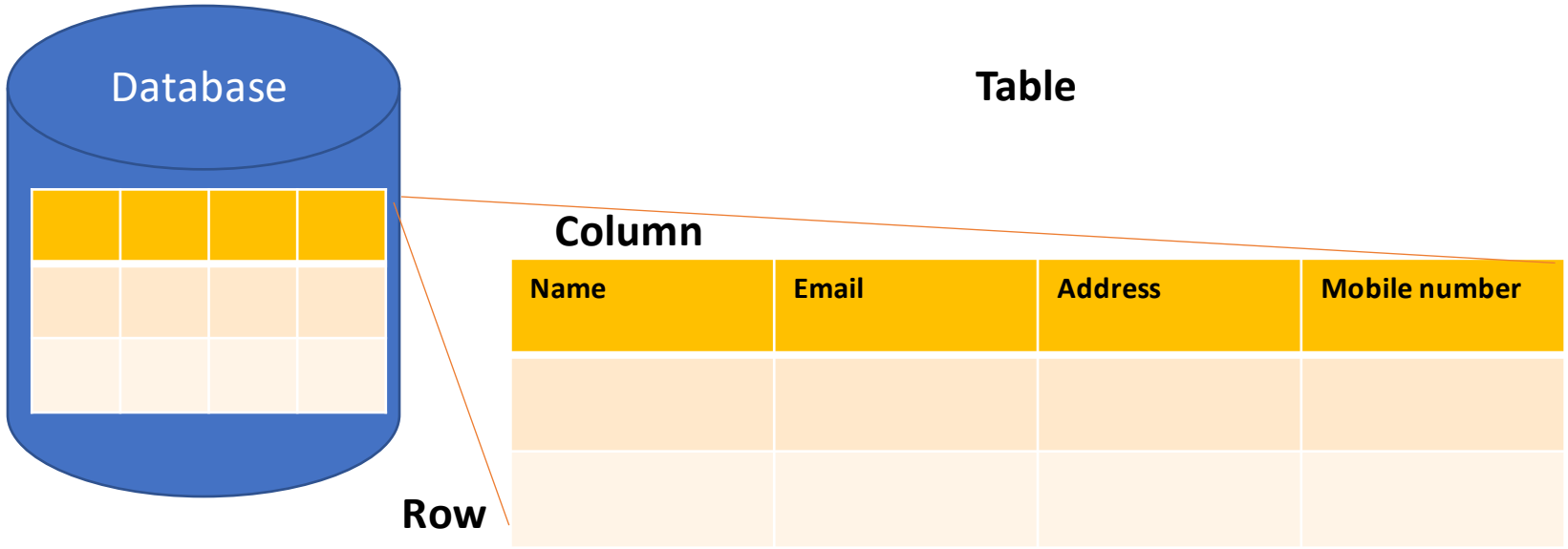
gitiho

6 CÁC ĐỐI TƯỢNG CHÍNH TRONG CSDL



gitiho

6 CÁC ĐỐI TƯỢNG CHÍNH TRONG CSDL



gitiho

BÀI 1. SQL DATABASE

gitiho

1 DATABASE LÀ GÌ?

Cơ sở dữ liệu trong SQL Server được tạo thành từ một tập hợp các bảng lưu trữ dữ liệu có cấu trúc cụ thể.

Một bảng chứa một tập hợp các hàng (còn được gọi là bản ghi) và cột (còn được gọi là thuộc tính).

Mỗi cột trong bảng được thiết kế để lưu trữ một loại thông tin nhất định

Ví dụ: Database Truong_hoc lưu trữ thông tin các bảng dữ liệu như Hoc_sinh, Giao_vien, Lop_hoc,...

gitiho

2 VAI TRÒ CỦA DATABASE?

Cơ sở dữ liệu đóng vai trò quan trọng trong các hoạt động của một tổ chức, cụ thể như:

- Lưu trữ thông tin một cách hệ thống
- Dữ liệu được sắp xếp có hệ thống và lưu trữ theo cấu trúc nhất định, mang tính nhất quán.
- Dễ dàng, thuận tiện trong việc thiết lập, tìm kiếm dữ liệu chính xác và nhanh chóng.

gitiho

3 TẠO, XÓA DATABASE

Khởi tạo database bằng giao diện
Step 1: Nhập chuột phải vào **Databases** > **New database**
Step 2: Nhập tên Database vào **Database Name**
Step 3: Click **OK**

Xóa database bằng giao diện
Step 1: Nhấp chuột phải vào **Database** cần xóa trong danh sách Databases bên trái > **Delete**
Step 2: Cửa sổ Delete Object hiển thị > đánh dấu vào ô Close existing connections > **OK** để chắc chắn xóa được

Khởi tạo database bằng query
Cú pháp:
CREATE DATABASE DATABASE_NAME

Example:
CREATE DATABASE HOC_SQL

Xóa database bằng query
Cú pháp:
DROP DATABASE DATABASE_NAME

Example:
DROP DATABASE HOC_SQL

gitiho

4 MỘT SỐ CHÚ Ý HỮU ÍCH

- Mỗi tên database là duy nhất
- Để thực thi câu lệnh có 2 cách: Bôi đen câu lệnh và ấn **F5** hoặc bôi đen câu lệnh và click **Execute** trên màn hình
- Muốn thực thi câu lệnh nào thì bôi đen câu lệnh đó
- Nếu ko bôi đen => thực thi tất cả câu lệnh trên file query

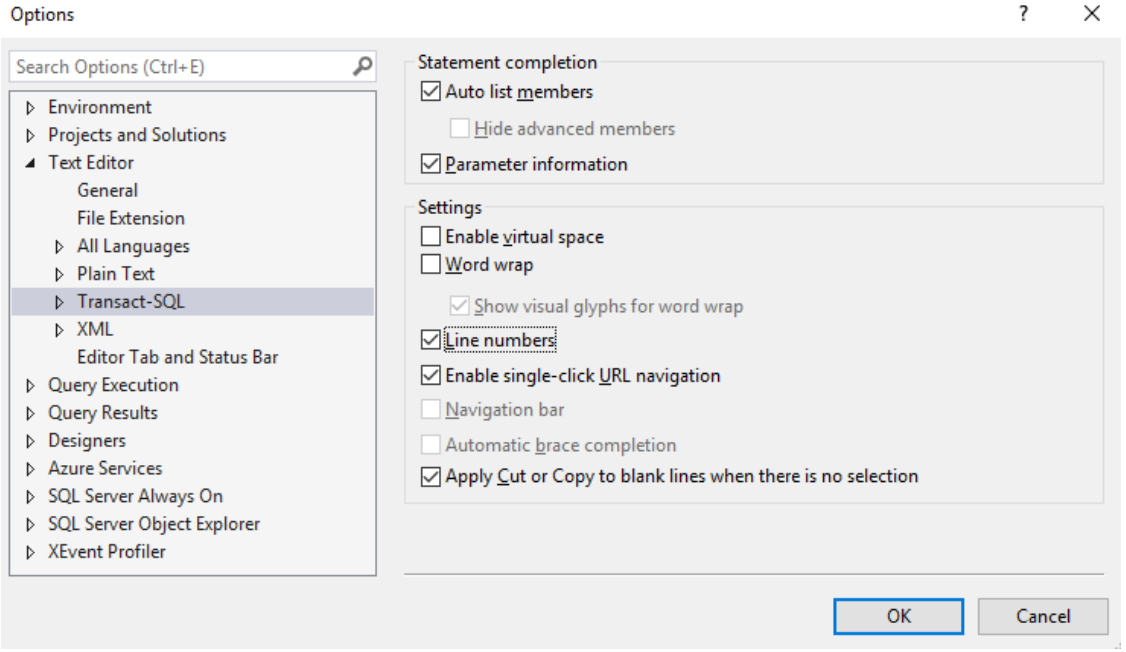
gitiho

4 MỘT SỐ CHÚ Ý HỮU ÍCH

1. Bật đánh dòng trong SQL:

Tool -> Option -> Text Editor -> Transact -SQL -> Line Number -> OK

=> Tác dụng phát hiện lỗi sai khi query bị báo lỗi



gitiho

BÀI 2. KIỂU DỮ LIỆU TRONG SQL

gitiho

1 CÁC KIỂU DỮ LIỆU THÔNG DỤNG TRONG SQL

Các kiểu dữ liệu thông dụng

- Kiểu dữ liệu chuỗi văn bản: **Varchar, Nvarchar**
- Kiểu dữ liệu: **Int, Float**
- Kiểu dữ ngày tháng: **Date, Datetime**

gitiho

2 CÁC KIỂU DỮ LIỆU THÔNG DỤNG TRONG SQL

Name	Type	Description
Integer	int	Số nguyên (Vd: 1, 2, 3, -3, -12,…)
Float	float	Số thực (Vd: 1.2 , 3.876,…)
Decimal	decimal(m,d)	Số thập phân, m là tổng số lượng các số còn d là số lượng các số nằm sau dấu phẩy. - m mặc định là 18 nếu không được chỉ định cụ thể. - d mặc định là 0 nếu không được chỉ định cụ thể. decimal(5,2) -> 123.00
Numeric	Numeric(m,d)	Số thập phân, tương tự decimal numeric(10,5) ->12345.12000
Date	date	Ngày (2019-01-15, May-26-2019,…
Datetime	datetime	Ngày + giờ (2019-01-15 20:39:42.03)
Character	char(n)	Kiểu ký tự (Không hỗ trợ Unicode), với “n” là số ký tự khai báo giới hạn Ví dụ: char(50) giới hạn 50 ký tự
Varcharacter	varchar(n)	Kiểu ký tự cấp phát động (Không hỗ trợ Unicode), với “n” là số ký tự khai báo giới hạn.
Nvarchar	nvarchar(n)	Kiểu ký tự cấp phát động (Hỗ trợ Unicode), với “n” là số ký tự khai báo giới hạn.

gitiho

BÀI 3. TOÁN TỬ TRONG SQL

gitiho

1 TOÁN TỬ TRONG SQL

Toán tử	Mô tả
Toán tử số học	
+	Cộng
-	Trừ
*	Nhân
/	Chia
%	Chia lấy số dư
Toán tử so sánh	
=	Bằng
<> ; !=	Khác
>	Lớn hơn
<	Bé hơn
>=	Lớn hơn hoặc bằng
<=	Bé hơn hoặc bằng

gitiho

2 TOÁN TỬ TRONG SQL

Toán tử	Mô tả
Toán tử logic	
AND	Trả về các bản ghi đáp ứng tất cả các điều kiện được phân tách bằng AND trong mệnh đề WHERE.
OR	Trả về các bản ghi đáp ứng tất cả các điều kiện được phân tách bằng OR trong mệnh đề WHERE.
NOT	Toán tử NOT đảo ngược ý nghĩa các toán tử logic đang sử dụng
[NOT] BETWEEN	Trả về các giá trị [NOT] trong một phạm vi nhất định
IN	Chỉ định nhiều giá trị trong mệnh đề WHERE (viết tắt cho nhiều điều kiện HOẶC)
IS [NOT] NULL	Trả về các bản ghi có giá trị [NOT] NULL trong các tập tin đã cho
[NOT] EXISTS	Toán tử EXISTS sử dụng để tìm kiếm sự tồn tại của 1 hàng trong 1 bảng được chỉ định đáp ứng 1 tiêu chí nhất định
[NOT] LIKE	Trả về các bản ghi [KHÔNG] khớp với một mẫu được chỉ định trong một cột.

gitiho

BÀI 4. SQL TABLE

gitiho

1 KHÁI NIỆM

- Bảng (table) là các đối tượng chứa tất cả dữ liệu trong cơ sở dữ liệu. Trong bảng, dữ liệu được tổ chức hợp lý theo định dạng hàng và cột tương tự như bảng tính.
- Mỗi hàng đại diện cho một bản ghi duy nhất và mỗi cột đại diện cho một trường trong bản ghi
- Một bảng chứa một tập hợp các hàng (còn được gọi là bản ghi) và cột (còn được gọi là thuộc tính).
- Ví dụ

Mã nhân viên	Tên	Địa chỉ	Chức danh	SĐT
1	A	Ha Noi	Nhan vien	0912345678

gitiho

2 TẠO TABLE

Khởi tạo bảng bằng giao diện

Step 1: Nhấp dấu (+) bên trái Database cần tạo Table > Tìm folder Table

Step 2: Nhấp chuột phải vào Table > New > Table… (hoặc New Table)

Step 3: Xuất hiện cửa sổ TenPC.TenDatabase – dbo.Table_1*

Step 4: Lưu Table vừa tạo bằng cách nhấn Biểu tượng SAVE hoặc phím tắt Ctrl + S > Cửa sổ Choose Name hiển thị. Nhập tên Table > OK

gitiho

3 TẠO TABLE

Khởi tạo bảng dữ liệu bằng câu lệnh

Lệnh **CREATE TABLE** trong SQL được sử dụng để tạo một bảng dữ liệu

Cú pháp:

```
CREATE TABLE table_name (
   column1 datatype,
   column2 datatype,
   column3 datatype,
  ....
);
```

Ví dụ:

```
CREATE TABLE Students (
student_id int,
last_name varchar(255),
first_name varchar(255),
email varchar(255),
address varchar(255)
);
```

gitiho

3 XÓA TABLE

Xóa bảng dữ liệu bằng câu lệnh

Lệnh **DROP TABLE** trong SQL được sử dụng để xóa một bảng và tất cả dữ liệu của bảng đó.

Ghi chú: Bạn phải thật cẩn thận trong khi sử dụng lệnh này, bởi vì một khi một bảng bị xóa thì tất cả thông tin có sẵn trong bảng đó cũng sẽ bị xóa vĩnh viễn.

Cú pháp:

```
DROP TABLE table_name
```

Ví dụ:

```
DROP TABLE Students
```

gitiho

4 THAY ĐỔI CẤU TRÚC BẢNG (THÊM, XÓA CỘT)

Thêm cột

Cú pháp:

ALTER TABLE *table_name*
ADD column_name datatype;

Ví dụ:

ALTER TABLE Students
ADD Phone_number int;

Xóa cột

Cú pháp:

ALTER TABLE table_name
DROP COLUMN column_name;

Ví dụ:

ALTER TABLE Students
DROP COLUMN Phone_number;

gitiho

4 BÀI TẬP THỰC HÀNH

Bài tập về nhà

Viết các câu lệnh theo các yêu cầu sau:

Bài 1: Khởi tạo database có tên là Homework

Bài 2: Tạo bảng thông tin nhân sự (staff_infor) trong Database Homework bao gồm các cột biết rằng:

STT int

Staff_name kiểu dữ liệu varchar độ rộng 25 ký tự

Staff_ID kiểu dữ liệu nvarchar độ rộng 50 ký tự

Salary kiểu dữ liệu dạng số (float)

Bài 3: Thêm cột Status kiểu dữ liệu nvarchar có độ rộng 25 ký tự vào bảng staff_infor

gitiho

BÀI 5. INSERT, DELETE, UPDATE TABLE (THÊM, XÓA, SỬA DỮ LIỆU TRONG SQL SERVER)

gitiho

1 LỆNH INSERT (THÊM) DỮ LIỆU

Lệnh **INSERT INTO** có 2 cú pháp cơ bản:

1. Chỉ định cả tên cột và giá trị được chèn:

INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...)
VALUES (*value1*, *value2*, *value3*, ...);

Ở đây, `col1, col2, col3,...colN` là tên các cột trong bảng mà bạn muốn chèn thêm dữ liệu. value1, value2, value3,...valueN là giá trị cần thêm vào tương ứng với các cột.

2.Nếu bạn muốn thêm giá trị vào tất cả các cột trong bảng => Không cần chỉ định tên các cột trong truy vấn
Tuy nhiên, phải đảm bảo thứ tự của các giá trị theo thứ tự với các cột trong bảng
Ở đây, cú pháp INSERT INTO sẽ như sau:

INSERT INTO *table_name*
VALUES (*value1*, *value2*, *value3*, ...);

`Ghi chú:` Bạn nhấn thấy sự khác biệt không? Ở cú pháp 2, bạn không cần liệt kê tên các cột trong bảng, chỉ cần đảm bảo rằng thứ tự các giá trị bạn muốn thêm vào đúng với thứ tự của các cột trong bảng là được.

gitiho

1 LỆNH INSERT (THÊM) DỮ LIỆU

Ví dụ:

Tạo bảng NHANVIEN như sau:

Results | Messages

	ID	TEN	TUOI	DIACHI	LUONG
1	1	Thanh	24	Haiphong	2000
2	2	Loan	26	Hanoi	1500
3	3	Nga	24	Hanam	2000
4	4	Manh	29	Hue	6500
5	5	Huy	28	Hatinh	8500
6	6	Cao	23	HCM	4500

gitiho

1 LỆNH INSERT (THÊM) DỮ LIỆU

Thực hành:

Tạo bảng NHANVIEN gồm các cột

```
CREATE TABLE NHANVIEN
(ID int,
TEN varchar(255),
TUOI int,
DIACHI varchar(255),
LUONG float )
```

Lệnh dưới đây sẽ tạo 6 bản ghi cho bảng NHANVIEN

```
INSERT INTO NHANVIEN(ID,TEN,TUOI,DIACHI,LUONG) VALUES (1, 'Thanh', 24, 'Haiphong', 2000.00 );
INSERT INTO NHANVIEN(ID,TEN,TUOI,DIACHI,LUONG) VALUES (2, 'Loan', 26, 'Hanoi', 1500.00 );
INSERT INTO NHANVIEN(ID,TEN,TUOI,DIACHI,LUONG) VALUES (3, 'Nga', 24, 'Hanam', 2000.00 );
INSERT INTO NHANVIEN(ID,TEN,TUOI,DIACHI,LUONG) VALUES (4, 'Manh', 29, 'Hue', 6500.00 );
INSERT INTO NHANVIEN(ID,TEN,TUOI,DIACHI,LUONG) VALUES (5, 'Huy', 28, 'Hatinh', 8500.00 );
INSERT INTO NHANVIEN(ID,TEN,TUOI,DIACHI,LUONG) VALUES (6, 'Cao', 23, 'HCM', 4500.00 );
```

Có thể sử dụng cú pháp 2 để thêm bản ghi thứ 7 cho bảng NHANVIEN như sau:

```
INSERT INTO NHANVIEN
VALUES (7, 'LAM', 29, 'Hanoi', 15000.00 );
```

gitiho

2 LỆNH DELETE (XÓA) DỮ LIỆU

Sử dụng **TRUNCATE** hoặc **DELETE** để xóa dữ liệu trong bảng

- Cú pháp với TRUNCATE : **TRUNCATE TABLE <Tên bảng>**
- Cú pháp với DELETE: **DELETE <Tên bảng> [where]**

GHI CHÚ: Cả hai câu lệnh này đều dùng để xóa dữ liệu trong 1 bảng tuy nhiên đối với DELETE bạn có thể xóa dữ liệu với điều kiện trong mệnh đề where.

Ví dụ 1: Xóa tất cả các dữ liệu trong bảng NHANVIEN vừa tạo

- Dùng TRUNCATE: TRUNCATE TABLE NHANVIEN
- Dùng DELETE: DELETE NHANVIEN

Ví dụ 2: Xóa tất cả các dữ liệu trong bảng NHANVIEN với điều kiện sau:

NOTE: Lưu ý xóa có điều kiện thì bạn chỉ có thể dùng DELETE chứ không thể dùng TRUNCATE

- Xóa dữ liệu với nhân viên có lượng nhỏ hơn 3000

=> DELETE NHANVIEN WHERE LUONG < 3000

- Xóa những nhân viên có địa chỉ không thuộc khu vực Hanoi và HCM

=> DELETE NHANVIEN WHERE DIACHI NOT IN ('Hanoi','HCM')

gitiho

3 LỆNH UPDATE (CẬP NHẬP) DỮ LIỆU

Tác dụng:

Câu lệnh UPDATE thay đổi hay cập nhật dữ liệu của một hay nhiều hàng trong một bảng dữ liệu.

Câu lệnh:

UPDATE <Tên bảng>
SET <col 1 = value 1>, <col 2 = value 2>,…,<col n = value n>
WHERE <điều kiện cập nhập>

Ví dụ 1: Cập nhập lương của tất cả nhân viên là 5000
=>

```
UPDATE NHANVIEN
SET LUONG = 5000
```

Ví dụ 2: Cập nhập tất cả những người nhỏ hơn 26 tuổi thành địa chỉ là HA NOI
=>

```
UPDATE NHANVIEN
SET DIACHI = 'HA NOI'
WHERE TUOI < 26
```

gitiho

4 BÀI TẬP THỰC HÀNH

1. Bạn hãy tạo một bảng với tên là manager với những thông tin bên dưới

manager_id	manager_name	manager_level	region	salary
111	Chris	2	Nunavut	370
112	William	3	West	240
113	Erin	3	Prarie	377
114	Sam	4	West	454
115	Pat	3	West	168

2. Thêm hai bản ghi cho bảng manager với thông tin như sau

manager_id	manager_name	manager_level	region	salary
116	Parker	1	Quebec	390
117	Robert	2	Prarie	407

3. Xóa tất cả những thông tin trong bảng manager với manager_level = 2

4. Cập nhập salary trong bảng manager là 500 với region là West và Quebec

4 ĐÁP ÁN

BT1. Bạn hãy tạo một bảng với tên là manager với những thông tin bên dưới:

Cách 1:

```
CREATE TABLE manager
(manager_id int,
manager_name varchar(255),
manager_level int,
region varchar(255),
salary float )

INSERT INTO manager (manager_id,manager_name,manager_level,region,salary) VALUES (111,'Chris',2,'Nunavut',370);
INSERT INTO manager (manager_id,manager_name,manager_level,region,salary) VALUES (112,'William',3,'West',240);
INSERT INTO manager (manager_id,manager_name,manager_level,region,salary) VALUES (113,'Erin',3,'Prarie',377);
INSERT INTO manager (manager_id,manager_name,manager_level,region,salary) VALUES (114,'Sam',4,'West',454);
INSERT INTO manager (manager_id,manager_name,manager_level,region,salary) VALUES (115,'Pat',3,'Ontario',168);
```

Cách 2:

```
CREATE TABLE manager
(manager_id int,
manager_name varchar(255),
manager_level int,
region varchar(255),
salary float )

INSERT INTO manager (manager_id,manager_name,manager_level,region,salary)
VALUES (111 ,'Chris' ,2 ,'Nunavut' ,370 );
 (112,'William',3,'West',240);
 (113,'Erin',3,'Prarie',377);
 (114,'Sam',4,'West',454);
 (115,'Pat',3,'Ontario',168);
```

4

ĐÁP ÁN

BT2. Thêm hai bản ghi cho bảng manager với thông tin như sau:

```
INSERT INTO manager VALUES (116,'Parker',1,'Quebec',390);
INSERT INTO manager VALUES (117,'Robert',2,'Prarie',407);
```

BT3. Xóa tất cả những thông tin trong bảng manager với manager_level = 2

```
DELETE manager WHERE manager_level = 2
```

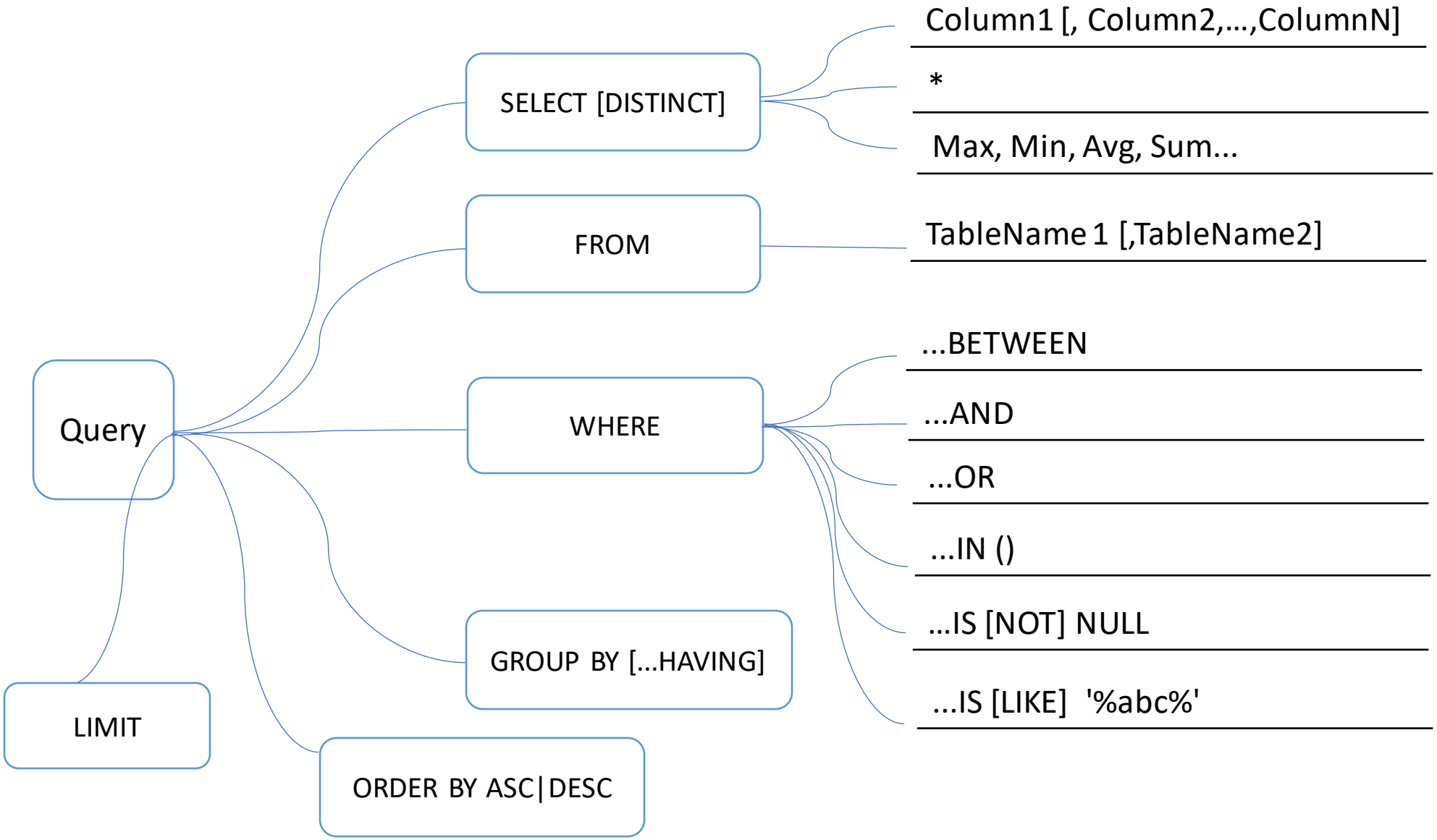
BT4. Cập nhập salary trong bảng manager là 500 với region là West và Quebec

```
UPDATE manager
SET salary = 500
WHERE region in ( 'West','Quebec')
```

BÀI 6. MỆNH ĐỀ SELECT

gitiho

1 GIỚI THIỆU SƠ ĐỒ QUERY CƠ BẢN



gitiho

2 SELECT FROM STATEMENT

Truy vấn dữ liệu cơ bản với toán tử SELECT

Câu lệnh **SELECT** trong SQL được sử dụng để lấy dữ liệu từ một bảng cơ sở dữ liệu trả về dữ liệu này dưới dạng một bảng kết quả. Các bảng kết quả này được gọi là tập kết quả (result-sets).

Cú pháp cơ bản của câu lệnh SELECT trong SQL như sau:

```
Select <Tên Cột>  <Bí Danh> <Biểu Thức>
From <Tên Bảng>;
```

- Select : Xác định những cột cần hiển thị
- From: Xác định bảng dữ liệu nguồn.

SELECT <Tên cột> FROM <Tên bảng>

1. Hiển thị tất cả các cột trong bảng

Ví dụ: Hiển thị tất cả các cột trong bảng Profiles

SELECT * FROM Profiles

2. Hiển thị các cột theo yêu cầu

Ví dụ: Hiển thị các cột Province, Region, Manager trong bảng Profiles

SELECT Province, Region, Manager FROM Profiles

MAGIC CODE INSTITUTE

2 SELECT FROM STATEMENT

Truy vấn dữ liệu cơ bản với toán tử SELECT

```
Select <Tên Cột> <Bí Danh> <Biểu Thức>
From <Tên Bảng>;
```

- Select : Xác định những cột cần hiển thị
- From: Xác định bảng dữ liệu nguồn.

SELECT <Bí danh> FROM <Tên bảng>

3. Hiển thị các cột bí danh (AS)

Ví dụ: Hiển thị bảng gồm cột Tinh (tương ứng cột Province), Khu_vuc (tương ứng cột Region) từ bảng Profiles

SELECT Provice AS Tinh, Region as Khu_vuc from Profiles

Thực hành: Hiển thị bảng gồm cột Ma_dat_hang (tương ứng cột order_id), Ngay_tra_hang (tương ứng cột returned_date) từ bảng Returns

MAGIC CODE INSTITUTE

5 SELECT FROM STATEMENT

SỬ DỤNG CÁC PHÉP TOÁN SỐ HỌC

Sử dụng các phép tính : + , - , * , / , %

Ví dụ: Hiển thị bảng kết quả gồm các cột:
order_id, order_date, order_quantity, value, profit , revenue từ bảng Orders.
Trong đó:
Revenue được tính bằng công thức: order_quantity* unit_price*(1- discount)

```
SELECT order_id, order_date, order_quantity, value, profit,
order_quantity* unit_price*(1- discount) as revenue
FROM Orders
```

MAGIC CODE INSTITUTE

2

BÀI TẬP THỰC HÀNH

Bài 1: Viết câu lệnh truy vấn dữ liệu từ bảng Orders

Hiển thị bảng kết quả gồm các cột: order_id, order_date, order_quantity, value, profit, revenue, total_cost

Trong đó:

- Công thức tính revenue là: order_quantity* unit_price*(1- discount)
- Công thức tính total_cost là: product_base_margin* unit_price+shipping_cost

BÀI 7. MỆNH ĐỀ SELECT DISTINCT

gitiho

1 SELECT DISTINCT STATEMENT

LOẠI BỎ NHỮNG DÒNG TRÙNG LẶP

```
SELECT DISTINCT column1, column2, colum3 FROM table_name
```

- Distinct: Loại bỏ những dòng (bản ghi) **trùng lặp**

Ví dụ: Hiển thị danh sách không trùng lặp của cột Manager trong bảng Profiles

```
SELECT DISTINCT Manager FROM Profiles
```

Hãy so sánh với kết quả câu truy vấn sau:

```
SELECT Manager FROM Profiles
```

Thực hành: Hãy viết câu truy vấn trả về danh sách không trùng lặp của cột Provice trong bảng Managers

2 BÀI TẬP THỰC HÀNH

Bài 1: Viết câu lệnh truy vấn trả về danh sách không trùng lặp các mức độ ưu tiên (order_priority) từ bảng Orders.

Bài 2: Viết câu lệnh truy vấn dữ liệu từ bảng Orders

Hiển thị bảng kết quả gồm các cột: customer_name, order_priority

Yêu cầu: Bảng kết quả không bao gồm các bản ghi trùng lặp

BÀI 8. MỆNH ĐỀ SELECT TOP

gitiho

1 SELECT TOP STATEMENT

GIỚI HẠN SỐ DÒNG KẾT QUẢ

```
SELECT TOP N [percent] [columns] FROM table_name
```

- TOP N: Trả về kết quả N dòng đầu tiên
- TOP N Percent: Trả về kết quả N% dòng đầu tiên

Ví dụ:

1. Hiển thị 10 dòng đầu tiên của bảng Orders

```
SELECT TOP 10 * FROM Orders
```

2. Hiển thị bảng gồm các cột order_id, order_date, order_priority, order_quantity, value từ bảng Orders. Giới hạn 20% dòng đầu tiên.

```
SELECT TOP 20 PERCENT order_id, order_date, order_priority, order_quantity, value
FROM Orders
```

Thực hành: Hãy viết câu truy vấn trả về 20 dòng đầu tiên của bảng gồm các cột order_id, order_date, order_priority từ bảng Orders.

MAGIC CODE INSTITUTE

2

BÀI TẬP THỰC HÀNH

Bài 1: Viết câu lệnh truy vấn dữ liệu từ bảng Orders

Hiển thị bảng kết quả gồm các cột: Order_id, order_date,customer_name, order_priority

Yêu cầu: Hiển thị 100 bản ghi đầu tiên

BÀI 9. MỆNH ĐỀ WHERE

gitiho

1 LỌC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

LỌC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

SELECT *column_name(s)* *FROM table_name* *WHERE column_name = value1;*	SELECT *column_name(s)* *FROM table_name* *WHERE column_name <> value1;*

1. Trả về kết quả BẰNG một giá trị xác định trong một cột

=> Sử dụng dấu "=" cho mệnh đề WHERE

Ví dụ: Viết câu truy vấn trả về các bản ghi từ bảng Orders với điều kiện cột order_priority='Medium'

| **SELECT * FROM Orders WHERE order_priority = 'Medium'**

2. Trả về kết quả KHÁC một giá trị xác định trong một cột
=> Sử dụng "!=" hoặc "<>" cho mệnh đề WHERE

Ví dụ: Viết câu truy vấn trả về các bản ghi từ bảng Orders với điều kiện cột order_priority không chứa giá trị 'High'

| **SELECT * FROM Orders WHERE order_priority <> 'High'**

MAGIC CODE INSTITUTE

1 LỌC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

LỌC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

SELECT *column_name(s)* *FROM table_name* *WHERE column_name IN (value1, value2, ...);*	SELECT *column_name(s)* *FROM table_name* *WHERE column_name NOT IN (value1, value2, ...);*

3. Trả về kết quả chứa danh sách nhiều giá trị xác định trong một cột

=> Sử dụng dấu "IN()" cho mệnh đề WHERE

Ví dụ: Viết câu truy vấn trả về các bản ghi từ bảng Orders với điều kiện cột order_priority gồm các giá trị cụ thể là 'Medium' , 'Low', 'High'

| **SELECT * FROM Orders WHERE order_priority IN ('Medium','Low','High')**

4. Trả về kết quả không bao gồm danh sách nhiều giá trị xác định trong một cột
=> Sử dụng "NOT IN()" cho mệnh đề WHERE

Ví dụ: Viết câu truy vấn trả về các bản ghi từ bảng Orders với điều kiện cột order_priority không bao gồm các giá trị là 'Medium', 'Low', 'High'

| **SELECT * FROM Orders WHERE order_priority NOT IN ('Medium', 'Low','High')**

MAGIC CODE INSTITUTE

1 LỌC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

LỌC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

SELECT *column_name(s)*
FROM table_name
WHERE column_name (not) like parttern;
Trong đó: 'a%' => Bắt đầu với ký tự a | '%a' => Kết thúc với ký tự a |'%a%' => Chứa ký tự a

3. Trả về kết quả theo điều kiện chưa xác định (điều kiện là chứa kí tự nào đó) trong một cột

=> Sử dụng "like" trong mệnh đề WHERE

VD1: Từ bảng Orders trả về 100 dòng kết quả đầu tiên với điều kiện cột shipping_mode chứa ký tự 'Air'

```
SELECT TOP 100 *
FROM Orders
WHERE shipping_mode like '%Air%'
```

VD2: Từ bảng Orders trả về các kết quả với điều kiện cột product_subcategory bắt đầu với ký tự "Co"

```
Select * from Orders where product_subcategory like 'Co%'
```

Lưu ý: Với điều kiện phủ định => sử dụng NOT LIKE

MAGIC CODE INSTITUTE

1 LỌC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

LỌC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

```
SELECT column_name(s)
FROM table_name
WHERE column_name is (not) null;
```

3. Lọc các kết quả theo điều kiện null

VD1: Từ bảng Returns, trả về các kết quả điều kiện cột status null (trống)

```
SELECT *
FROM Returns
WHERE status is null
```

VD2: Từ bảng Returns, trả về các kết quả điều kiện cột status không null (trống)

```
SELECT *
FROM Returns
WHERE status is not null
```

MAGIC CODE INSTITUTE

1 LỌC KẾT QUẢ SỬ DỤNG MỆNH ĐỀ WHERE

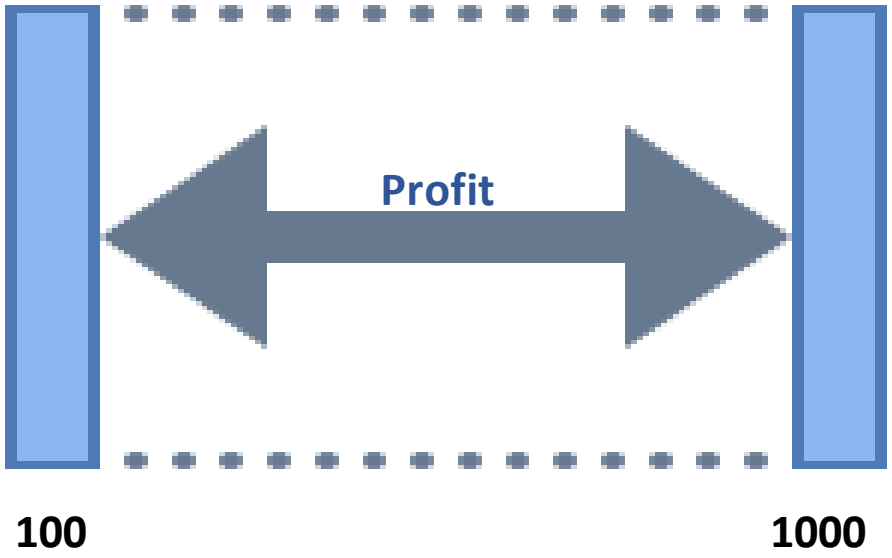
LỌC KẾT QUẢ THEO ĐIỀU KIỆN SỬ DỤNG MỆNH ĐỀ WHERE

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

4. Lọc kết quả trong một khoảng nào đó

VD1: Từ bảng Orders, trả về các kết quả điều kiện cột profit nằm trong khoảng (100,1000)

```
SELECT *
FROM Orders
WHERE profit between 100 and 1000
```



MAGIC CODE INSTITUTE

2 BÀI TẬP THỰC HÀNH

Bài 1: Từ bảng Orders. Viết các câu lệnh theo từng điều kiện sau:

1.1 region là “West”

1.2 order_priority không bao gồm “Critical”

1.3 order_prioritylà “High” hoặc order_priority là “Low” hoặc order_priority là “Medium” hoặc order_priority là “Not Specified”

1.4 province chứa từ “New”

1.5 shipping_mode không chứa từ “Air” và value nhỏ hơn 500

1.6 product_subcategory bắt đầu với từ “Co”

1.7 customer_segment kết thúc là “e” và order_quantity lớn hơn 10

BÀI 10. HÀM MAX, MIN, AVG, COUNT, SUM

gitiho

1 HÀM MAX() AND MIN()

SỬ DỤNG HÀM MIN() VÀ MAX()

SELECT MIN(*column_name*) FROM *table_name* *WHERE condition*;	SELECT MAX(*column_name*) FROM *table_name* *WHERE condition*;

VD1: Lấy giá trị nhỏ nhất của cột profit từ bảng Orders

```
SELECT MIN(profit)
FROM Orders
```

VD2: Lấy giá trị lớn nhất của cột Order_quantity từ bảng Orders

```
SELECT MAX(Order_quantity)
FROM Orders
```

MAGIC CODE INSTITUTE

8 HÀM COUNT(), AVG(), SUM()

SỬ DỤNG HÀM COUNT, AVG , SUM

SELECT COUNT ([DISTINCT] *column_name*) FROM *table_name* *WHERE condition*;	SELECT AVG(*column_name*) FROM *table_name* *WHERE condition*;	SELECT SUM(*column_name*) FROM *table_name* *WHERE condition*;

VD1: Đếm số dòng cột order_priority từ bảng Orders

```
SELECT COUNT(Order_priority)
FROM Orders
```

VD2: Tính giá trị trung bình profit từ bảng Orders

```
SELECT AVG(Profit)
FROM Orders
```

VD3: Tính tổng profit từ bảng Orders

```
SELECT SUM(Profit)
FROM Orders
```

MAGIC CODE INSTITUTE

BÀI 11. MỆNH ĐỀ GROUP BY

gitiho

1 BẢNG TỔNG HỢP VỚI GROUP BY

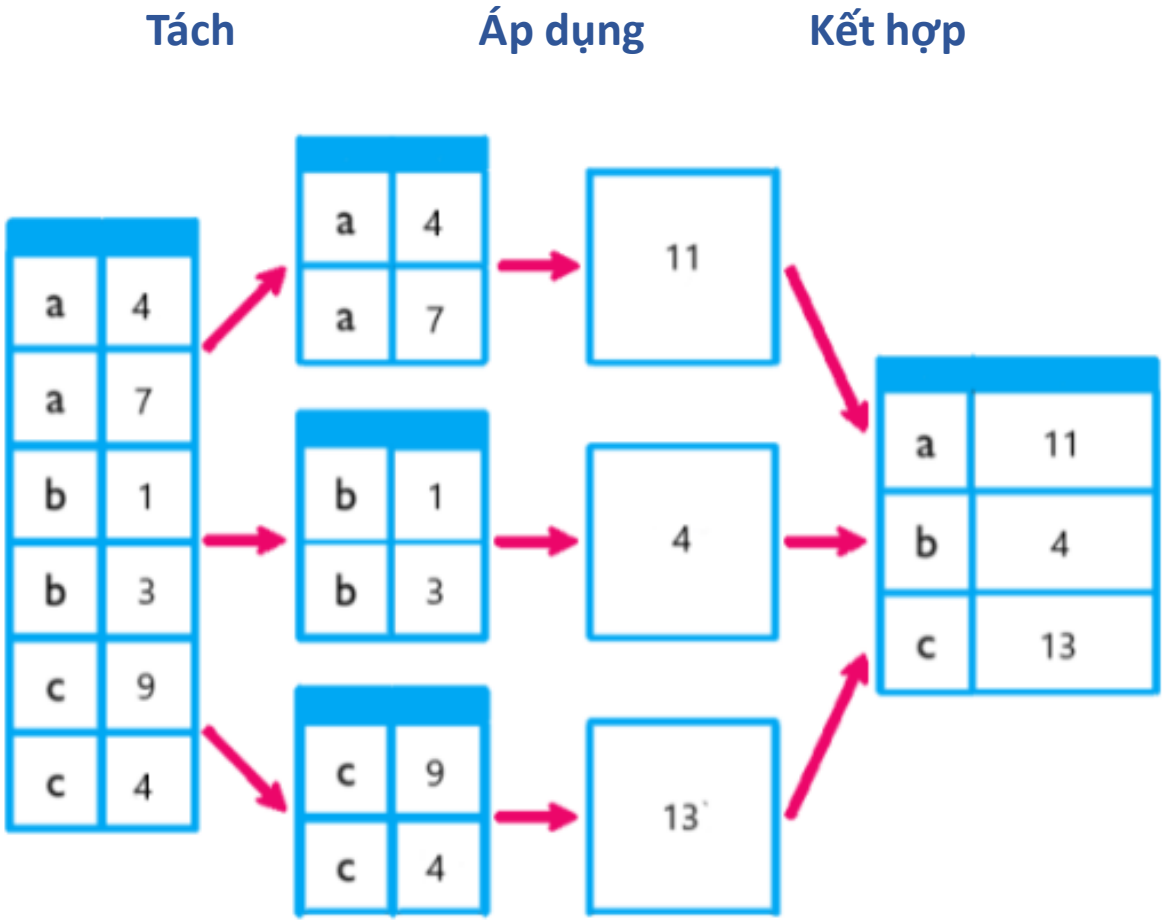
MÔ TẢ MỤC ĐÍCH SỬ DỤNG GROUP BY

Câu lệnh SQL Group By sử dụng chiến lược tách-áp-dụng-kết hợp.

1->Tách: Các nhóm khác nhau được chia theo giá trị của chúng.

2->Áp dụng: Hàm tổng hợp được áp dụng cho các giá trị của các nhóm này.

3->Kết hợp: Các giá trị được kết hợp trong một hàng.



MAGIC CODE INSTITUTE

1 BẢNG TỔNG HỢP VỚI GROUP BY

NHÓM KẾT QUẢ THEO MỘT HOẶC NHIỀU CỘT SỬ DỤNG GROUP BY

```
SELECT column_1, function_name(column_2)--bao gồm cột tạo mới bằng hàm tổng hợp
FROM table_name
GROUP BY column_name--nhóm các hàng để áp dụng hàm tổng hợp
ORDER BY column_name;--sắp xếp tăng/giảm nếu có
```

- Câu lệnh GROUP BY **nhóm các hàng** có **cùng giá trị** thành các hàng tóm tắt, như "tìm số lượng khách hàng ở mỗi quốc gia".
- Câu lệnh GROUP BY **thường được sử dụng với các hàm tổng hợp** (COUNT (), MAX (), MIN (), SUM (), AVG ()) để nhóm tập hợp kết quả theo một hoặc nhiều cột.

VD1: Trả về bảng tính tổng lợi nhuận (profit) theo các tỉnh (Province)

```
SELECT Province AS Tinh, SUM(Profit) as Tong_loi_nhuan
FROM Orders
GROUP BY Provine
```

Hãy mang tất cả các cột không áp dụng hàm tổng hợp ở SELECT để đặt vào GROUP BY

Bí quyết nè

MAGIC CODE INSTITUTE

2 BÀI TẬP THỰC HÀNH

Bài 3: Từ bảng Orders:

Viết lệnh truy vấn dữ liệu để tổng hợp kết quả tổng lợi nhuận (total_profit) theo từng tỉnh (province) và khách hàng (customer_name).

Kết quả trả về như bảng bên.

Results | Messages

	province	customer_name	total_profit
1	Alberta	Aaron Bergman	5881.6252
2	Nunavut	Aaron Bergman	8.572
3	British Columbia	Aaron Hawkins	3207.9784
4	Nova Scotia	Aaron Hawkins	261.1248
5	Ontario	Aaron Hawkins	112.2494
6	Quebec	Aaron Hawkins	1105.8156
7	Yukon	Aaron Hawkins	4889.796
8	Manitoba	Aaron Smayling	1727.8632
9	Quebec	Aaron Smayling	2718.2786
10	Alberta	Adam Bellavance	3218.6785
11	Ontario	Adam Bellavance	239.3556
12	British Columbia	Adam Hart	2897.6271
13	Yukon	Adam Hart	7778.5523
14	Alberta	Adam Shillingsburg	9076.5425
15	Alberta	Adrian Barton	13139.6214

BÀI 12. MỆNH ĐỀ HAVING

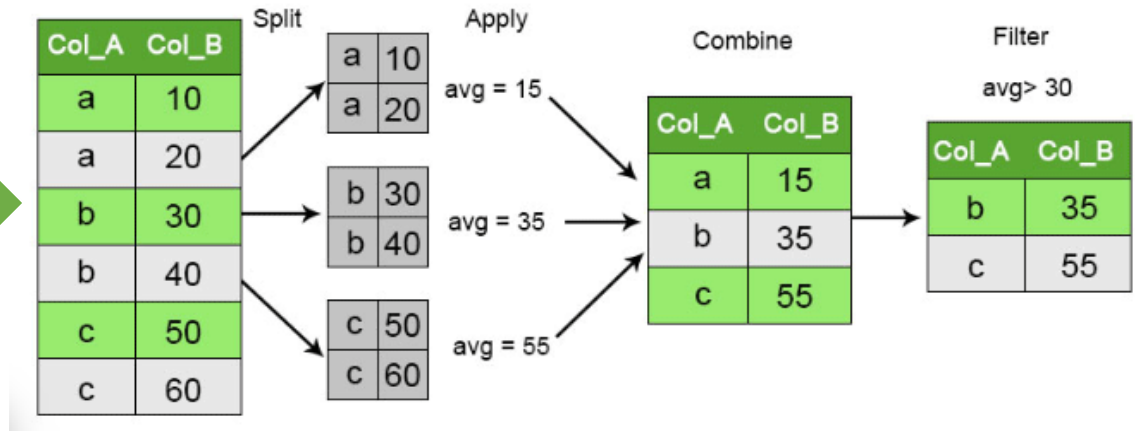
gitiho

1 LỌC KẾT QUẢ VỚI HAVING

MỆNH ĐỀ HAVING

```
SELECT <column_1, function_name(column_2),...>--Trong  select chứa hàm tổng hợp
FROM <table>
WHERE <condition>          -- Điều kiện áp dụng với cột có sẵn
GROUP BY <column(s)>       -- Nhóm các hàng để áp dụng hàm tổng hợp
HAVING <condition>         -- Hàm tổng hợp trong điều kiện
ORDER BY <column(s)>;      -- Xác định thứ tự tăng giảm (Tùy chọn)
```

SELECT Col_A, avg(Col_B) as Col_B
FROM MyTable
GROUP BY Col_A
HAVING avg(Col_B)>30



(*)Mệnh đề HAVING đã được thêm vào SQL vì từ khóa WHERE không thể được sử dụng với các hàm tổng hợp

gitiho

1 LỌC KẾT QUẢ VỚI HAVING

MỆNH ĐỀ HAVING

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
```

- **Mệnh đề HAVING đã được thêm vào SQL vì từ khóa WHERE không thể được sử dụng với các hàm tổng hợp.**

VD1: Trả về bảng tính tổng lợi nhuận (profit) theo các tỉnh (Province). Điều kiện là lấy các tỉnh có tổng lợi nhuận lớn hơn 10000

```
SELECT Province AS Tinh, SUM(Profit) as Tong_loi_nhuan
FROM Orders
GROUP BY Provine
HAVING SUM(profit) > 10000
```

gitiho

2 SO SÁNH WHERE VÀ HAVING

SỰ KHÁC NHAU GIỮA 2 MỆNH ĐỀ WHERE VÀ HAVING

WHERE	*HAVING*
Sử dụng để lọc điều kiện trên ***cột có sẵn***	*Sử dụng để lọc điều kiện của* ***hàm tổng hợp***

Bài tập thực hành:

1. Từ bảng Orders, lọc ra các kết quả với điều kiện cột Order_priority là "Low"

Output:

Results | Messages

	ID	order_id	order_date	order_priority	order_quantity	value	discou
1	1	3	2010-10-13 00:00:00.000	Low	6	224.2944	0.04
2	11	66	2009-01-19 00:00:00.000	Low	41	107.4528	0.09
3	14	70	2010-12-17 00:00:00.000	Low	48	86.6016	0.03
4	15	70	2010-12-17 00:00:00.000	Low	46	9001.763	0.05
5	18	129	2012-11-18 00:00:00.000	Low	4	23.5872	0.09

2. Từ bảng Orders, tạo bảng tính tổng số lượng sản phẩm (total_quantity) theo từng product_name. Lọc các kết quả có tổng số lượng sản phẩm lớn hơn 10

Output:

	product_name	total_quantity
1	Avery 510	382
2	Advantus Push Pins, Aluminum Head	344
3	Acco D-Ring Binder w/DublLock®	105
4	Staples Standard Envelopes	251
5	Canon PC1080F Personal Copier	180

gitiho

BÀI 13. MỆNH ĐỀ ORDER BY

gitiho

1 SẮP XẾP VỚI ORDER BY

SẮP XẾP KẾT QUẢ TĂNG DẦN/GIẢM DẦN SỬ DỤNG ORDER BY

```
SELECT column_name(s) --Các cột muốn hiển thị
FROM table_name
GROUP BY column_name(s) --Nhóm các hàng để áp dụng hàm tổng hợp (nếu có)
ORDER BY  column_name(s) ASC/DESC --Sắp xếp theo tứ thự tăng/giảm tùy chọn
```

- ASC: Tăng dần
- DESC: Giảm dần

VD1: Trả về bảng tính tổng lợi nhuận (profit) theo các tỉnh (Province) và sắp xếp theo thứ tự giảm dần

```
SELECT Province AS Tinh, SUM(Profit) as Tong_loi_nhuan
FROM Orders
GROUP BY Provine
ORDER BY SUM(Profit) DESC
```

gitiho

2

BÀI TẬP THỰC HÀNH

Bài 1: Viết câu lệnh truy vấn dữ liệu từ bảng Orders

Hiển thị bảng kết quả gồm tất cả các cột của bảng orders

Yêu cầu: Sắp xếp theo thứ tự profit giảm dần

Bài 2: Viết câu lệnh truy vấn dữ liệu từ bảng Orders

Tổng hợp tổng giá trị (total_value) theo từng khách hàng (customer_name)

Yêu cầu: Sắp xếp theo thứ tự tổng giá trị giảm dần

gitiho

THỨ TỰ CÁC MỆNH ĐỀ

Ở dạng đầy đủ, cú pháp của lệnh SELECT trong SQL Server sẽ là:

Cú pháp:

```
SELECT [ ALL | DISTINCT ]
[ TOP (n) [ PERCENT ]]
"biểu thức"
FROM “bảng”
[WHERE “điều kiện”]
[GROUP BY “biểu thức”]
[HAVING “điều kiện”]
[ORDER BY “biểu thức” [ ASC | DESC ]];
```

gitiho

BÀI 14. GHÉP BẢNG THEO CHIỀU NGANG (JOIN)

gitiho

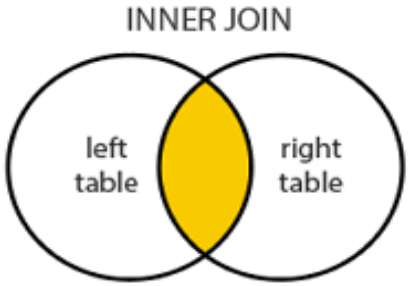
1 Giới thiệu Join trong SQL

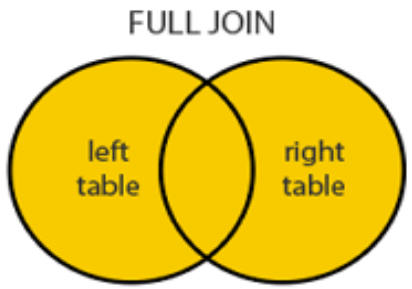
1. Join là gì?

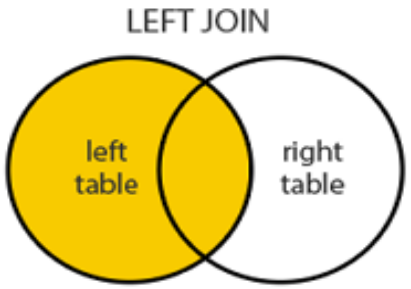
JOIN được sử dụng để kết hợp các dữ liệu từ hai hoặc nhiều bảng dựa trên mối liên quan giữa chúng

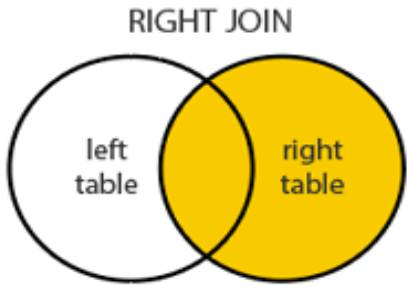
2. Các loại Join thường dùng

- **INNER JOIN** hay JOIN
- **LEFT OUTER JOIN** hay LEFT JOIN
- **RIGHT OUTER JOIN** hay RIGHT JOIN
- **FULL JOIN** hay FULL OUTER JOIN



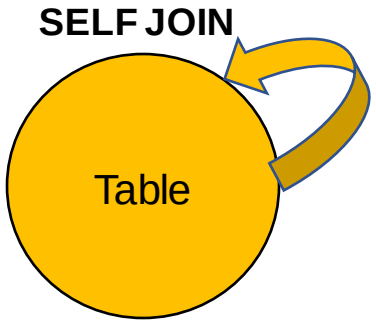


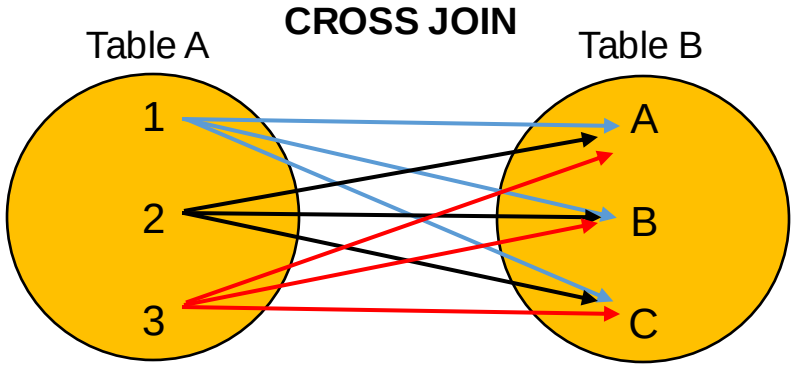




3. Một số loại Join khác

- **SELF JOIN**
- **CROSS JOIN**

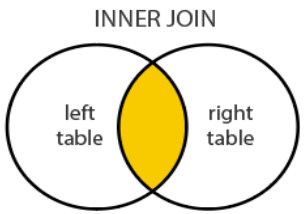




gitiho

2

Cách sử dụng và cấu trúc INNER JOIN



Kết quả:
Trả về các bản ghi có giá trị khớp trong cả hai bảng

Group	Value
A	2
B	3
C	4
D	1

INNER JOIN

Group	Order
C	6
D	7
E	8
F	9

Group	Value	Order
C	4	6
D	1	7

Cấu trúc câu lệnh mẫu:

```
SELECT T1.COLUMNS, T2.COLUMNS
FROM Table_A AS T1
INNER JOIN Table_B AS T2
ON T1.Key_column_name = T2.Key_column_name
```

Ví dụ:

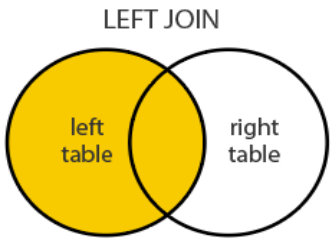
```
SELECT T1.manager_id
,T1.manager_name
,T2.region
FROM Managers as T1
INNER JOIN Profiles as T2
ON T1.manager_name=T2.manager
```

Bài tập thực hành:
Từ bảng Orders và Returns. Sử dụng inner join viết câu truy vấn bảng kết quả gồm những thông tin sau:
order_id, order_date, value, product_name, returned_date

gitiho

3

Cách sử dụng và cấu trúc LEFT JOIN



Kết quả:
Trả về tất cả các bản ghi từ bảng bên trái (A) và các bản ghi phù hợp từ bảng bên phải (B).
Kết quả là NULL từ bên phải, nếu không có kết quả phù hợp.

Group	Value
A	2
B	3
C	4
D	1

LEFT JOIN

Group	Order
C	6
D	7
E	8
F	9

Group	Value	Order
A	2	Null
B	3	Null
C	4	6
D	1	7

Cấu trúc câu lệnh mẫu:
SELECT T1.COLUMNS, T2.COLUMNS
FROM Table_A AS T1 --Bảng A sẽ là bảng gốc
LEFT JOIN Table_B AS T2 --Bảng B sẽ là bảng tham chiếu
ON T1.Key_column_name = T2.Key_column_name

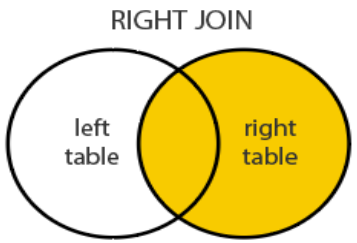
Ví dụ:
SELECT
T1.manager_id
,T1.manager_name
,T2.region
FROM Managers as T1
LEFT JOIN Profiles as T2
ON T1.manager_name=T2.manager

Bài tập thực hành:
Từ bảng Orders và Returns. Viết câu truy vấn sử dụng left join trả về bảng kết quả gồm các trường thông tin sau:
order_id, order_date, total_value, total_order_quantity, status, returned_date

gitiho

4

Cách sử dụng và cấu trúc RIGHT JOIN



Kết quả:
Trả về tất cả các bản ghi từ bảng bên phải (B) và các bản ghi phù hợp từ bảng bên trái (A).
Kết quả là NULL từ bên trái, nếu không có kết quả phù hợp.

Group	Value
A	2
B	3
C	4
D	1

RIGHT JOIN

Group	Order
C	6
D	7
E	8
F	9

Group	Order	Vlue
C	6	4
D	7	1
E	8	Null
F	9	Null

Cấu trúc câu lệnh mẫu:

```
SELECT T1.COLUMNS, T2.COLUMNS
FROM Table_A AS T1 --Bảng A sẽ là bảng tham chiếu
RIGHT JOIN Table_B AS T2 --Bảng B sẽ là bảng gốc
ON T1.Key_column_name = T2.Key_column_name
```

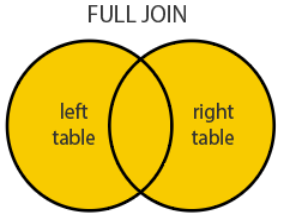
Ví dụ:

```
SELECT
T1.manager_id
,T1.manager_name
,T2.region
FROM Managers as T1
RIGHT JOIN Profiles as T2
ON T1.manager_name=T2.manager
```

Bài tập thực hành:
Từ bảng Orders và Returns. Viết câu truy vấn sử dụng right join trả về bảng kết quả gồm các trường thông tin sau:
order_id, order_date, total_value, total_order_quantity, status, returned_date

gitiho

5 Cách sử dụng và cấu trúc FULL OUTER JOIN



Kết quả:
Trả về tất cả các bản ghi khi có kết quả khớp trong bản ghi bảng trái (A) hoặc phải (B)

Group	Value
A	2
B	3
C	4
D	1

FULL OUTER JOIN

Group	Order
C	6
D	7
E	8
F	9

Group	Value	Order
A	2	Null
B	3	Null
C	4	6
D	1	7
E	Null	8
F	Null	9

Cấu trúc câu lệnh mẫu:
SELECT T1.COLUMNS, T2.COLUMNS
FROM Table_A AS T1
FULL OUTER JOIN Table_B AS T2
ON T1.Key_column_name = T2.Key_column_name

Ví dụ:
SELECT t1.manager_id, t1.manager_level, t1.manager_name, t1.manager_phone, t2.Manager, t2.province
FROM Managers as t1
FULL OUTER JOIN Profiles as t2
ON t1.manager_name = t2.manager

Bài tập thực hành:
Từ bảng Orders và Returns sử dụng Full Outer Join lấy ra thông tin order_id, order_date, total_value, total_order_quantity, status, returned_date

gitiho

7

Bài tập thực hành

BT1.

Từ bảng Orders và Returns tính tổng order_quantity, value, profit của các đơn hàng có trạng thái status = 'Returned' (kết quả chỉ lấy ra đơn hàng có trạng thái returned)

Lưu ý: Làm với 3 loại join (inner join, left join, right join)

Đáp án mẫu: (lấy các trường thông tin như trong bảng kết quả mẫu)

Results | Messages

	order_id_return	order_date	total_order_quantity	total_value	total_profit	returned_date
1	45127	2012-12-30 00:00:00.000	10	13.32	7.144	2013-01-13 00:00:00.000
2	42945	2012-12-29 00:00:00.000	45	165.3075	55.305	2013-01-12 00:00:00.000
3	29505	2012-12-27 00:00:00.000	22	37.9456	15.5624	2013-01-11 00:00:00.000
4	30469	2012-12-25 00:00:00.000	46	293.112	137.6924	2013-01-10 00:00:00.000
5	23619	2012-12-21 00:00:00.000	100	4246.1012	1908.7484	2013-01-05 00:00:00.000

BT2.

Sử dụng 2 bảng orders và profiles

Tính total_order_quantity, total_value,total_profit của từng manager

Gợi ý: Mối quan hệ giữa 2 bảng là cột province

Results | Messages

	manager	total_order_quantity	total_value	total_profit
1	Erin	51521	3467244.1212	1468952.2242
2	Pat	29311	2266301.2911	955382.580700001
3	Sam	46685	3022359.53560001	1231891.2602
4	Chris	15558	1092511.6531	447856.6088
5	William	71702	4892129.3638	2048438.9546

gitiho

8 Hướng dẫn giải bài tập về nhà

BT1.

Dùng inner join

```
Select o.order_id AS order_id_return,
o.order_date,
sum(o.order_quantity) AS total_order_quantity,
sum(o.value) AS total_value,
sum(o.profit) AS total_profit,
r.returned_date
from Orders AS o
inner join Returns AS r
on o.order_id=r.order_id
Group by o.order_id,o.order_date,r.returned_date
Order by o.order_date desc
```

Dùng left join

```
select o.order_id AS order_id_return,
o.order_date,
sum(o.order_quantity) AS total_order_quantity,
sum(o.value) AS total_value,
sum(o.profit) AS total_profit,
r.returned_date
from Returns AS r
left join Orders AS o
on o.order_id=r.order_id
Group by o.order_id,o.order_date,r.returned_date
Order by o.order_date desc
```

Dùng right join

```
select o.order_id AS order_id_return,
o.order_date,
sum(o.order_quantity) AS total_order_quantity,
sum(o.value)  AS total_value,
sum(o.profit)  AS total_profit,
r.returned_date
from Orders AS o
right join Returns AS r
on o.order_id=r.order_id
Group by o.order_id,o.order_date,r.returned_date
Order by o.order_date desc
```

BT2.

```
select p.manager,
sum(order_quantity) AS total_order_quantity,
sum(value) AS total_value,
sum(profit) AS total_profit
from Orders AS o
left join Profiles AS p
on o.province=p.province
group by p.manager
```

gitiho

BÀI 15. GHÉP BẢNG THEO DỌC (UNION)

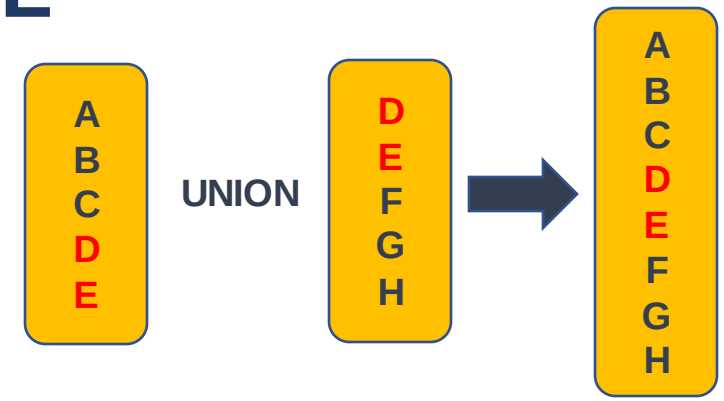
gitiho

6

Giới thiệu UNION Và UNION ALL

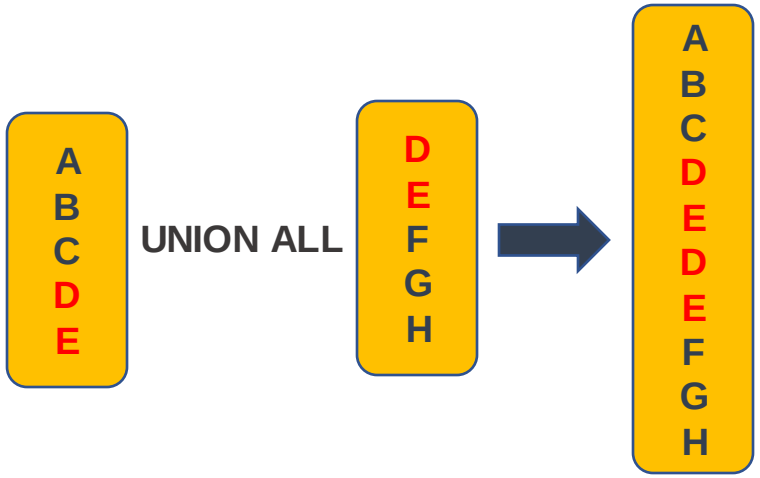
1. Định nghĩa

Union và Union All là phép nối các bảng có **cấu trúc giống nhau**. **Union All** sẽ lấy **tất cả bản ghi** của phép nối còn **Union** sẽ chỉ lấy **các bản ghi duy nhất**



2. Cấu trúc giống nhau là gì?

- Các bảng phải có cùng số cột
- Các cột cũng phải có các loại dữ liệu tương tự
- Các cột trong mỗi bảng phải theo thứ tự



3. Cấu trúc câu lệnh mẫu:

SELECT column1, column2, column3 FROM table1
UNION hoặc **UNION ALL**
SELECT column1, column2, column3 FROM tablle2

4. Ví dụ

SELECT t1.province, t1.region FROM Profiles t1
UNION ALL
SELECT t2.province, t2.region FROM Orders t2

Bài tập thực hành:

Dùng Union và Union All nối tên nhà quản lý từ 2 bảng Profiles với Managers sau đó rút ra nhận xét so sánh về kết quả số lượng bản ghi từ hai phép nối.

gitiho

2 Bài tập thực hành

BT1. Hiển thị bảng tính tổng lợi nhuận (total_profit)mỗi mức độ ưu tiên (order_priority) sử dụng union all

Kết quả như ảnh:

100 %

Results | Messages

	order_priority	total_profit
1	Not Specified	1152521.8831
2	Low	1327044.6468
3	High	1357097.7837
4	Medium	1203274.7473
5	Critical	1112582.5676

BT2.

Hiển thị bảng tính tổng lợi nhuận mỗi mức độ ưu tiên bao gồm dòng total như ảnh:

Results | Messages

	order_priority	total_profit
1	Not Specified	1152521.8831
2	Low	1327044.6468
3	High	1357097.7837
4	Medium	1203274.7473
5	Critical	1112582.5676
6	Total	6152521.62850002

gitiho

3 Hướng dẫn giải bài tập về nhà

BT1.

```
Select order_priority, sum(profit) AS total_profit
From Orders
Where order_priority = 'Not Specified'
Group by order_priority
Union all
Select order_priority, sum(profit) AS total_profit
From Orders
Where order_priority = 'Low'
Group by order_priority
Union all
Select order_priority, sum(profit) AS total_profit
From Orders
Where order_priority = 'High'
Group by order_priority
Union all
Select order_priority, sum(profit) AS total_profit
From Orders
Where order_priority = 'Medium'
Group by order_priority
Union all
Select order_priority, sum(profit) AS total_profit
From Orders
Where order_priority = 'Critical'
Group by order_priority
```

BT2.

```
Select order_priority, sum(profit) AS total_profit
From Orders
Group by order_priority

UNION ALL

Select 'Total' AS order_priority, sum(profit)
AS total_profit
From Orders
```

gitiho

BÀI 16. CÁC HÀM XỬ LÝ DỮ LIỆU

gitiho

1

Các hàm xử lí chuỗi

STT	Hàm	Cách dùng	Cấu trúc	Ví dụ
1	CHARINDEX()	Trả về vị trí của một chuỗi con trong một chuỗi	CHARINDEX(*substring*, *string*, *start*)	SELECT CHARINDEX('M', 'MAGIC CODE INSTITUTE') AS char_index
2	CONCAT()	Nối các chuỗi với nhau	CONCAT(string1, string2,..)	SELECT CONCAT('MAGIC','CODE','INSTITUTE') AS chars_combined
3	CONCAT_WS()	Thêm hai hoặc nhiều chuỗi với một dấu phân cách	CONCAT_WS(separator, string1, string2,..)	SELECT CONCAT_WS(' ', 'MAGIC', 'CODE', 'INSTITUTE')
4	LEN()	Đo chiều dài chuỗi	LEN(string)	SELECT LEN('MAGIC CODE INSTITUTE') AS char_length
5	LEFT()	Trích xuất một số ký tự từ một chuỗi (bắt đầu từ trái sang)	LEFT(string, number_of_chars)	SELECT LEFT('MAGIC CODE INSTITUTE', 5) AS extract_string
6	RIGHT()	Trích xuất một số ký tự từ một chuỗi (bắt đầu từ phải sang)	RIGHT(string, number_of_chars)	SELECT RIGHT('MAGIC CODE INSTITUTE', 9) AS extract_string
7	LTRIM() or RTRIM()	Loại bỏ khoảng trắng đầu chuỗi hoặc cuối chuỗi	LTRIM(string) RTRIM(string)	SELECT LTRIM(' MCI') AS Left_Trimmed SELECT RTRIM('MCI ') AS Right_Trimmed
8	PATINDEX()	Trả về vị trí của một mẫu trong một chuỗi	PATINDEX(%pattern%, string)	SELECT PATINDEX('%D%', 'MAGIC CODE INSTITUTE')

2 Các hàm xử lí chuỗi

STT	Hàm	Cách dùng	Cấu trúc	Ví dụ
9	REPLACE()	Thay thế tất cả các lần xuất hiện của một chuỗi con trong một chuỗi, với một chuỗi con mới	REPLACE(*string*, *old_string*, *new_string*)	SELECT REPLACE('MAGIC', 'MA', 'LO')
10	REPLICATE()	Lặp lại một chuỗi một số lần đã xác định	REPLICATE(string, times)	SELECT REPLICATE('MCI ', 3)
11	REVERSE()	Đảo ngược chuỗi	REVERSE(string)	SELECT REVERSE('MCI')
12	SUBSTRING()	Trích xuất một số ký tự từ một chuỗi	SUBSTRING(*string*, *start*, *length*)	SELECT SUBSTRING('MAGIC CODE INSTITUTE', 1, 5) AS extract_string
13	STUFF()	Xóa một phần của chuỗi rồi chèn một phần khác vào chuỗi, bắt đầu từ một vị trí được chỉ định	STUFF(*string*, *start*, *length*, *new_string*)	SELECT STUFF('MAGIC CODE INSTITUTE', 1, 2, 'LO')
14	LOWER()	Chuyển đổi chuỗi thành chữ thường	LOWER(string)	SELECT LOWER('MAGIC CODE INSTITUTE ')
15	UPPER()	Chuyển đổi chuối thành chữ in	UPPER(string)	SELECT LOWER('mci ')
16	STRING_AGG()	Là một hàm tổng hợp nối các hàng chuỗi thành một chuỗi duy nhất, được phân tách bằng dấu phân tách được chỉ định.	STRING_AGG (input_string, separator)	SELECT Product_category, STRING_AGG(Product_subcategory,';') AS Product_subcategory_list FROM Orders GROUP BY Product_category;

MAGIC CODE INSTITUTE

3 Các hàm xử lí thời gian

STT	Hàm	Cách dùng	Cấu trúc	Ví dụ
1	CURRENT_TIMESTAMP()	Trả về thời gian hiện tại	SELECT CURRENT_TIMESTAMP	SELECT CURRENT_TIMESTAMP
2	DATEADD()	Cộng thêm một khoảng thời gian	DATEADD(interval, number, date)	SELECT DATEADD(year, 1, '2020-03-10') as date_add
3	DATEDIFF()	Trả về khoảng cách giữa 2 ngày	DATEDIFF(interval, *date1*, *date2*)	SELECT DATEDIFF(year, '2021-03-10', '2020-03-10')
4	DATEFROMPARTS()	Trả về một ngày từ các phần được chỉ định (năm, tháng, ngày)	DATEFROMPARTS(year, month, day)	SELECT DATEFROMPARTS(2021, 10, 07) AS date_from_parts
5	DATEPART()	Trả về một phần đã xác định của một ngày (dưới dạng số nguyên)	DATEPART(interval, date)	SELECT DATEPART(year, '2021/12/01') AS date_part_int
6	DAY()	Trả về ngày trong ngày đã cho	DAY(date)	SELECT DAY('2021/05/23') AS get_day
7	MONTH()	Trả về tháng trong ngày đã cho	MONTH(date)	SELECT MONTH('2021/05/23') AS get_month
8	YEAR()	Trả về năm trong ngày đã cho	YEAR(date)	SELECT YEAR('2021/05/23') AS get_year

4 Các hàm chuyển đổi

STT	Hàm	Cách dùng	Cấu trúc	Ví dụ
1	CAST()	Chuyển đổi một giá trị (thuộc bất kỳ loại nào) thành một kiểu dữ liệu được chỉ định	CAST(expression AS data_type[length])	SELECT CAST('2021' as int) as new_datatype
2	CONVERT()	Chuyển đổi một giá trị (thuộc bất kỳ l oại nào) thành một kiểu dữ liệu được chỉ định	CONVERT(data_type[length], expression)	SELECT CONVERT(int, '2021') as new_datatype
3	STR()	Trả về dạng chuỗi	STR()	SELECT STR(2021)
4	ISNUMERIC()	Kiểm tra có dạng số hay không	ISNUMERIC()	SELECT ISNUMERIC(2021)
5	ISNULL()	Trả về giá trị đã chỉ định nếu biểu thức là NULL, nếu không trả về biểu thức	ISNULL()	SELECT ISNULL(NULL, 'MCI')

5 Mệnh đề Case When

- Hàm CASE đi qua các điều kiện đến khi một điều kiện Đúng, hàm này sẽ dừng và trả về kết quả
- Nếu không có điều kiện nào đúng, hàm này trả về giá trị trong mệnh đề ELSE
- Nếu không có ELSE và không có điều kiện nào đánh giá là đúng, hàm trả về NULL

=> Tương tự hàm if trong Excel

Cấu trúc hàm Case When:

CASE WHEN (điều kiện 1) THEN value_1
WHEN (điều kiện 1) THEN value_2
... ...
WHEN (điều kiện N) THEN value_N
ELSE value_0
END

Ví dụ:

SELECT product_category, product_subcategory,
CASE WHEN shipping_mode like '%Air%' then 'Air'
ELSE 'Other' **END** AS shipping_method
FROM Orders

Bài tập thực hành:

1. Từ bảng Orders: Nếu discount = 0 trả về 'No Discount' nếu không thì trả về 'Discount'

2. Từ bảng Orders sử dụng mệnh đề Case When theo điều kiện sau:

- Value lớn hơn 1000 trả về High
- Value từ 200 đến 1000 trả về Medium
- Value nhỏ hơn 200 trả về Low

Cột mới tạo ra được đặt tên là Range_Value

MAGIC CODE INSTITUTE

5 Bài tập thực hành

BT1. Tạo ra một bảng bao gồm các cột:
order_id, customer_name, product_category, product_subcategory, product_name, thickness
Thỏa mãn các điều kiện sau:
Product_subcategory = 'Pens & Art Supplies'
Product_name chứa từ 'Newell'
Giả sử rằng nếu product_name là "Newell 345", nó có nghĩa là độ dày của nó là 345 mm

Results | Messages

	order_id	customer_name	product_category	product_subcategory	product_name	THICKNESS
1	66	Arthur Gainer	Office Supplies	Pens & Art Supplies	Newell 340	340 mm
2	69	Jonathan Doherty	Office Supplies	Pens & Art Supplies	Newell 308	308 mm
3	225	Karen Ferguson	Office Supplies	Pens & Art Supplies	Newell 314	314 mm
4	706	Sarah Jordon	Office Supplies	Pens & Art Supplies	Newell 326	326 mm
5	1540	Don Miller	Office Supplies	Pens & Art Supplies	Newell 335	335 mm
6	1825	Aaron Smayling	Office Supplies	Pens & Art Supplies	Newell 310	310 mm
7	2209	Pete Armstrong	Office Supplies	Pens & Art Supplies	Newell 336	336 mm
8	2563	Brenda Bowman	Office Supplies	Pens & Art Supplies	Newell 315	315 mm
9	2915	Tim Brockman	Office Supplies	Pens & Art Supplies	Newell 315	315 mm
10	3078	Paul Lucas	Office Supplies	Pens & Art Supplies	Newell 340	340 mm
11	3458	Lisa DeChemey	Office Supplies	Pens & Art Supplies	Newell 326	326 mm
12	3492	Roy Skaria	Office Supplies	Pens & Art Supplies	Newell 318	318 mm
13	3521	Giulietta Baptist	Office Supplies	Pens & Art Supplies	Newell 336	336 mm
14	3553	Roy Skaria	Office Supplies	Pens & Art Supplies	Newell 343	343 mm

BT2. Tạo ra một bảng bao gồm các cột:
Year, count_orders, total_value, total_profit, total_quantity và sắp xếp theo thứ tự các năm giảm dần.

Results | Messages

	year	count_orders	total_value	total_profit	total_orrder_quantity
1	2012	1392	3674953.3207	1509559.5014	54449
2	2011	1317	3411676.39850001	1421788.8944	51564
3	2010	1393	3505310.1089	1450979.7442	54384
4	2009	1394	4148606.1367	1770193.4885	54380

BT3.
Từ bảng Managers tạo ra một bảng gồm các cột:
manager_name, manager_level, manager_phone, level
Cột level được tạo ra thỏa mãn điều kiện sau:
Nếu manager_level =1 trả về là 'Fresher'
Nếu manager_level = 2 và 3 trả về là 'Junior'
Nếu manager_level = 4 trả về là 'Senior'

Results | Messages

	manager_name	manager_level	manager_phone	level
1	Chris	2	613-555-0166	Junior
2	William	3	807-555-0118	Junior
3	Erin	3	306-555-0193	Junior
4	Sam	4	709-555-0139	Senior
5	Pat	3	613-555-0184	Junior
6	Parker	1	416-555-0117	Fresher
7	Robert	2	418-555-0186	Junior
8	Martin	2	250-555-0143	Junior
9	Lucas	1	506-555-0128	Fresher

BÀI 17. SUBQUERY-TRUY VẤN CON TRONG SQL

gitiho

1 Phương pháp sử dụng Subquery

Truy vấn con trong SQL (subquery) là gì?

Trong SQL Server, truy vấn con là một truy vấn nằm trong một truy vấn khác. Bạn có thể tạo các truy vấn trong lệnh SQL. Các truy vấn con này nằm trong mệnh đề **WHERE, FROM** hoặc **SELECT**

```
SELECT order_id,
       order_quantity,
       product_name
FROM Orders
WHERE order_quantity > (SELECT AVG(order_quantity)
                        FROM Orders)
```

Subquery

Lưu ý:

- Truy vấn con còn được gọi là **INNER QUERY** hay **INNER SELECT.**
- Truy vấn chính mà chứa truy vấn con được gọi là **OUTER QUERY** hay **OUTER SELECT.**

MAGIC CODE INSTITUTE

1 Phương pháp sử dụng Subquery

Có một vài quy tắc mà Sub query phải tuân theo:

- Sub query phải được đặt trong dấu ngoặc đơn.
- Không thể sử dụng lệnh ORDER BY trong sub query, mặc dù truy vấn chính có thể sử dụng ORDER BY.
- Lệnh GROUP BY được sử dụng bình thường trong 1 sub query.

2 VÍ DỤ

Sub query ở mệnh đề SELECT

```sql
Select
    order_priority,
    sum(profit) as total_profit,
    (select avg(profit) from orders) as avg_profit --Subquery
from orders
group by order_priority
```

	order_priority	total_profit	avg_profit
1	Not Specified	1152521.8831	732.530256994883
2	Low	1327044.6468	732.530256994883
3	High	1357097.7837	732.530256994883
4	Medium	1203274.7473	732.530256994883
5	Critical	1112582.5676	732.530256994883

2 VÍ DỤ

Sub query ở mệnh đề FROM

```
Select * from
    (
        Select
            order_priority,
            sum(profit) as total_profit
        from orders
        group by order_priority
    ) as A --Subquery
Where total_profit > 1200000
```

Giải thích:
Kết quả của câu truy vấn con này được gán bí danh là A, chúng ta có thể tiếp tục được sử dụng các câu lệnh truy vấn dữ liệu để truy vấn trực tiếp dữ liệu của bảng A này như các bảng có sẵn trong Database

	order_priority	total_profit
1	Low	1327044.6468
2	High	1357097.7837
3	Medium	1203274.7473

MAGIC CODE INSTITUTE

2 VÍ DỤ

Sub query ở mệnh đề WHERE

```
Select order_id, order_date, profit
from orders
where profit > (select avg(profit) from orders ) --Subquery
```

	order_id	order_date	profit
1	32	2011-07-15 00:00:00.000	777.0084
2	35	2011-10-22 00:00:00.000	886.3926
3	36	2011-11-02 00:00:00.000	1058.239
4	65	2011-03-17 00:00:00.000	1813.5972
5	70	2010-12-17 00:00:00.000	3405.2044
6	96	2009-04-16 00:00:00.000	2042.1272
7	132	2010-06-10 00:00:00.000	966.904
8	134	2012-04-30 00:00:00.000	1010.3311
9	230	2010-10-26 00:00:00.000	735.1516
10	261	2010-06-28 00:00:00.000	2447.6161
11	293	2012-10-01 00:00:00.000	3501.924
12	294	2010-07-15 00:00:00.000	1535.0285
13	322	2012-03-18 00:00:00.000	990.256
14	324	2012-03-19 00:00:00.000	1830.9075
15	359	2009-12-18 00:00:00.000	1031.123

3 BÀI TẬP THỰC HÀNH

BT1: Từ bảng dữ liệu **Orders**

Viết câu truy vấn tạo ra bảng kết quả gồm các cột: product_name, total_value, avg_value như bảng sau:

	product_name	total_value	avg_value
1	Avery 510	1374.2625	1755.03583340873
2	Advantus Push Pins, Aluminum Head	1913.814	1755.03583340873
3	Acco D-Ring Binder w/DublLock®	2140.138	1755.03583340873
4	Staples Standard Envelopes	1341.7296	1755.03583340873
5	Canon PC1080F Personal Copier	106672.2221	1755.03583340873
6	Desktop 3-Pocket Hot File®	3411.546	1755.03583340873
7	Prang Colored Pencils	613.284	1755.03583340873
8	Xerox 1933	1722.7612	1755.03583340873
9	Global Leather and Oak Executive Chair, Black	87482.8468	1755.03583340873
10	Canon PC940 Copier	203404.4798	1755.03583340873
11	Memorex 80 Minute CD-R Spindle, 100/Pack	8631.5148	1755.03583340873
12	Strathmore Photo Mount Cards	899.028	1755.03583340873
13	Eldon Expressions Punched Metal & Wood Desk Acce...	3275.3112	1755.03583340873
14	REDIFORM Incoming/Outgoing Call Register, 11" X 8 ...	829.6632	1755.03583340873
15	Acco Pressboard Covers with Storage Hooks, 14 7/8" x ...	1494.4567	1755.03583340873

Trong đó:

Product_name: là tên sản phẩm

Total_value: là tổng giá trị các đơn hàng của từng sản phẩm

Avg_value: là giá trị đơn hàng trung bình của tất cả các sản phẩm

Gợi ý: Sử dụng Subquery để tính avg_value trong mệnh đề SELECT

3 BÀI TẬP THỰC HÀNH

BT1: Từ bảng dữ liệu **Orders**

Viết câu truy vấn tạo ra bảng kết quả gồm các sản phẩm có tổng lợi nhuận > giá trị trung bình lợi nhuận của tất cả các đơn hàng. Bảng kết quả gồm các cột: product_name, total_value như bảng sau:

	product_name	total_value
1	Advantus Push Pins, Aluminum Head	1913.814
2	Acco D-Ring Binder w/DublLock®	2140.138
3	Canon PC1080F Personal Copier	106672.2221
4	Desktop 3-Pocket Hot File®	3411.546
5	Global Leather and Oak Executive Chair, Black	87482.8468
6	Canon PC940 Copier	203404.4798
7	Memorex 80 Minute CD-R Spindle, 100/Pack	8631.5148
8	Eldon Expressions Punched Metal & Wood Desk Acce...	3275.3112
9	Hon Pagoda™ Stacking Chairs	30127.1828
10	Imation 3.5" Diskettes, IBM Format, DS/HD, 10/Box, Ne...	3000.8706
11	Global Leather Task Chair, Black	29689.5008
12	Okidata ML184 Turbo Dot Matrix Printers	75999.255
13	Bush Cubix Collection Bookcases, Fully Assembled	19196.5326
14	Seth Thomas 12" Clock w/ Goldtone Case	6392.1168
15	Bevis Round Conference Table Top & Single Column B...	20993.9364

Gợi ý: Sử dụng Subquery để tạo ra bảng tổng lợi nhuận theo từng sản phẩm. Đặt bảng kết quả đó có bí danh là A. Sau đó truy vấn bảng A và lọc điều kiện trong mệnh đề WHERE

3 BÀI TẬP THỰC HÀNH

Gợi ý lời giải:

BT1:

```
select product_name, sum(value) as total_value,
(select avg(value) from orders ) as avg_value
from orders
group by product_name
```

BT2:

```
Select * from
    (
    select product_name, sum(value) as total_value
    from orders
    group by product_name
    ) A --Tính tổng giá trị theo từng sản phẩm trước
Where total_value > (select avg(value) from orders) --subquery để lấy giá trị trung bình tất cả các đơn hàng
```

BÀI 18. CTE - BIỂU THỨC BẢNG CHUNG TRONG SQL

gitiho

1 Phương pháp sử dụng CTE-Common Table Expression

CTE trong SQL Server là gì?

- CTE có thể được xem như **một bảng chứa dữ liệu tạm thời** từ câu lệnh được định nghĩa trong phạm vi của nó.
- CTE tương tự như một bảng dẫn xuất (derived table) ở chỗ nó không được lưu trữ như một đối tượng và chỉ kéo dài trong suốt thời gian của câu truy vấn. Không giống như bảng dẫn xuất, CTE có thể tự tham chiếu tới bản thân của nó và có thể tham chiếu nhiều lần trong một câu truy vấn.

WITH

CTE name

```
with engineers as (
  select *
  from employees
  where dept='Engineering'
)
select *
from engineers
where ...
```

CTE Body

CTE Usage

1 Phương pháp sử dụng CTE-Common Table Expression

Mục đích của CTE

- Tạo truy vấn đệ quy (recursive query).
- Thay thế View trong một số trường hợp.
- Sử dụng được nhiều CTE trong một truy vấn duy nhất

Ưu điểm của CTE

CTE có nhiều ưu điểm như khả năng đọc dữ liệu được cải thiện và dễ dàng bảo trì các truy vấn phức tạp. Các truy vấn có thể được phân thành các khối nhỏ, đơn giản. Những khối này được sử dụng để xây dựng các CTE phức tạp hơn cho đến khi tập hợp kết quả cuối cùng được tạo ra.

2 VÍ DỤ

```sql
with A as
(
select order_priority, sum(profit) as total_profit
from orders
group by order_priority
)
select * from A where total_profit > 1200000
```

A là một CTE

	order_priority	total_profit
1	Low	1327044.6468
2	High	1357097.7837
3	Medium	1203274.7473

3 BÀI TẬP THỰC HÀNH

BT1: Từ bảng dữ liệu **Orders**

Viết câu truy vấn tạo ra bảng kết quả để lấy danh sách các khách hàng có số lượng đơn hàng > 10. Kết quả như bảng dưới:

	customer_name	count_orders
1	Brian Moss	12
2	Sheri Gordon	12
3	Liz Carlisle	11
4	Doug Bickford	23
5	Eugene Moren	11
6	Tracy Collins	18
7	Arthur Prichep	13
8	Tracy Poddar	12
9	Justin Knight	19
10	Neil French	13
11	Anthony Johnson	12
12	Adam Bellavance	12

Yêu cầu: Sử dụng CTEs

3 BÀI TẬP THỰC HÀNH

Gợi ý lời giải:

BT1:

```
with A as --Tạo CTE A để tính tổng đơn hàng theo từng khách hàng trước
(
Select customer_name , count(order_id) as count_orders
from orders
group by customer_name
)
Select * from A where count_orders > 10 --Tiếp tục truy vấn với bảng A
```

3 BÀI TẬP THỰC HÀNH

BT1: Từ bảng dữ liệu **Orders**

Viết câu truy vấn tạo ra bảng kết quả để lấy danh sách các khách hàng có số lượng đơn hàng > 10. Kết quả như bảng dưới:

	customer_name	count_orders
1	Brian Moss	12
2	Sheri Gordon	12
3	Liz Carlisle	11
4	Doug Bickford	23
5	Eugene Moren	11
6	Tracy Collins	18
7	Arthur Prichep	13
8	Tracy Poddar	12
9	Justin Knight	19
10	Neil French	13
11	Anthony Johnson	12
12	Adam Bellavance	12

Yêu cầu: Sử dụng CTEs

BÀI 19. BẢNG TẠM TRONG SQL

gitiho

1 Phương pháp sử dụng BẢNG TẠM-TEMP TABLE

Bảng tạm trong SQL server là gì ?

Bảng tạm là các có cấu trúc và chức năng như một bảng cố định bình thường trong SQL Server. Nhưng thay vì tạo ra một bảng trong Database, bảng tạm được tạo ra và lưu trữ trong tempdb.

Đối với SQL Server có 2 dạng bảng tạm đó là:

- **Local temporary table (#Table1)**: Bảng tạm cục bộ, sử dụng để tạo ra bảng tạm và tồn tại trong kết nối của người dùng tạo ra bảng đó và sẽ bị huỷ khi ngắt kết nối.
- **Global temporary table (##Table2)**: Bảng tạm toàn cục, sử dụng để tạo ra bảng tạm và tồn tại đến khi nào tất cả các kết nối đến cơ sở dữ liệu làm việc đóng hết. Có thể sử dụng ở kết nối của người dùng khác.

2 Phương pháp sử dụng BẢNG TẠM-TEMP TABLE

Cú pháp tạo bảng tạm cục bộ

Local temporary table

CÁCH 1: Tạo bảng tạm bằng câu lệnh CREATE TABLE thì bảng được tạo sẽ là bảng rỗng. Muốn thêm dữ liệu phải sử dụng câu lệnh INSERT INTO để thêm dữ liệu vào bảng tạm đó.

```
CREATE TABLE #TempTable1
(
  ID INT,
  Name VARCHAR(10),
  DOB DATETIME
)
```

CÁCH 2: Tạo bảng tạm để lưu kết quả của một câu truy vấn

```
SELECT COLUMNS INTO #TempTable1 --Chỉ cần thêm INTO #TempTable1 trước FROM
FROM TABLE_NAME
...
```

Lưu ý: Bắt buộc có dấu # trước tên bảng tạm

Để xem bảng các bạn mở công cụ SSMS và chọn: **System Databases > tempdb > Temporary Tables**

MAGIC CODE INSTITUTE

2 Phương pháp sử dụng BẢNG TẠM-TEMP TABLE

Cú pháp tạo bảng tạm toàn cục

Global temporary table

CÁCH 1: Tạo bảng tạm bằng câu lệnh CREATE TABLE thì bảng được tạo sẽ là bảng rỗng. Muốn thêm dữ liệu phải sử dụng câu lệnh INSERT INTO để thêm dữ liệu vào bảng tạm đó.

```
CREATE TABLE ##TempTable1
(
  ID INT,
  Name VARCHAR(10),
  DOB DATETIME
)
```

CÁCH 2: Tạo bảng tạm để lưu kết quả của một câu truy vấn

```
SELECT COLUMNS INTO ##TempTable1 --Chỉ cần thêm INTO #TempTable1 trước FROM
FROM TABLE_NAME
...
```

Lưu ý: Bắt buộc có dấu ## trước tên bảng tạm

Để xem bảng các bạn mở công cụ SSMS và chọn: **System Databases > tempdb > Temporary Tables**

MAGIC CODE INSTITUTE

BÀI 20. CHỦ ĐỀ XẾP HẠNG ROW_NUMBER(), RANK(), DENSE_RANK()

gitiho

1 Chủ đề XẾP HẠNG (RANKING TOPIC)

Sử dụng hàm **ROW_NUMBER (), RANK() và DENSE_RANK()**

Phân biệt **ROW_NUMBER, RANK và DENSE_RANK**

Đầu tiên, bạn nhìn ví dụ sau:

```
SELECT Customer_name, order_id, Order_Date,
ROW_NUMBER() OVER(ORDER BY Order_Date ASC) AS [ROW_NUMBER],
RANK() OVER(ORDER BY Order_Date ASC) AS [RANK],
DENSE_RANK() OVER(ORDER BY Order_Date ASC) AS [DENSE_RANK]
FROM Orders
WHERE Customer_name = 'Tamara Chand';
```

Results | Messages

	Customer_name	order_id	Order_Date	ROW_NUMBER	RANK	DENSE_RANK
1	Tamara Chand	57415	2009-03-24 00:00:00.000	1	1	1
2	Tamara Chand	6592	2009-12-01 00:00:00.000	2	2	2
3	Tamara Chand	6592	2009-12-01 00:00:00.000	3	2	2
4	Tamara Chand	640	2010-01-22 00:00:00.000	4	4	3
5	Tamara Chand	640	2010-01-22 00:00:00.000	5	4	3
6	Tamara Chand	44960	2011-05-03 00:00:00.000	6	6	4
7	Tamara Chand	44960	2011-05-03 00:00:00.000	7	6	4
8	Tamara Chand	44960	2011-05-03 00:00:00.000	8	6	4

gitiho

1 Chủ đề XẾP HẠNG (RANKING TOPIC)

Results | Messages

	Customer_name	order_id	Order_Date	ROW_NUMBER	RANK	DENSE_RANK
1	Tamara Chand	57415	2009-03-24 00:00:00.000	1	1	1
2	Tamara Chand	6592	2009-12-01 00:00:00.000	2	2	2
3	Tamara Chand	6592	2009-12-01 00:00:00.000	3	2	2
4	Tamara Chand	640	2010-01-22 00:00:00.000	4	4	3
5	Tamara Chand	640	2010-01-22 00:00:00.000	5	4	3
6	Tamara Chand	44960	2011-05-03 00:00:00.000	6	6	4
7	Tamara Chand	44960	2011-05-03 00:00:00.000	7	6	4
8	Tamara Chand	44960	2011-05-03 00:00:00.000	8	6	4

Nhận xét:

Cả 3 hàm đều Order by **Order_Date**

Hàm ROW_NUMBER dùng để xếp hạng kết quả một cách tuần tự bắt đầu từ 1 và không quan tâm đến các giá trị giống nhau.

Tại dòng thứ 2 và dòng 3, giá trị Order_Date đều giống nhau. RANK và DENSE_RANK đều trả về giá trị là 2.

Tại dòng thứ 4, hàm RANK trực tiếp bắt kịp giá trị của ROW_NUMBER bằng cách bỏ đi giá trị 3. Còn DENSE_RANK thì trả về giá trị tiếp theo dựa trên dòng trước đó.

Tương tự với các dòng dưới

gitiho

1 Chủ đề XẾP HẠNG (RANKING TOPIC)

Results | Messages

	Customer_name	order_id	Order_Date	ROW_NUMBER	RANK	DENSE_RANK
1	Tamara Chand	57415	2009-03-24 00:00:00.000	1	1	1
2	Tamara Chand	6592	2009-12-01 00:00:00.000	2	2	2
3	Tamara Chand	6592	2009-12-01 00:00:00.000	3	2	2
4	Tamara Chand	640	2010-01-22 00:00:00.000	4	4	3
5	Tamara Chand	640	2010-01-22 00:00:00.000	5	4	3
6	Tamara Chand	44960	2011-05-03 00:00:00.000	6	6	4
7	Tamara Chand	44960	2011-05-03 00:00:00.000	7	6	4
8	Tamara Chand	44960	2011-05-03 00:00:00.000	8	6	4

Nhận xét:

Cả 3 hàm đều Order by **Order_Date**

Hàm ROW_NUMBER dùng để xếp hạng kết quả một cách tuần tự bắt đầu từ 1 và không quan tâm đến các giá trị giống nhau.

Tại dòng thứ 2 và dòng 3, giá trị Order_Date đều giống nhau. RANK và DENSE_RANK đều trả về giá trị là 2.

Tại dòng thứ 4, hàm RANK trực tiếp bắt kịp giá trị của ROW_NUMBER bằng cách bỏ đi giá trị 3. Còn DENSE_RANK thì trả về giá trị tiếp theo dựa trên dòng trước đó.

Tương tự với các dòng dưới

gitiho

1 Chủ đề XẾP HẠNG (RANKING TOPIC)

HÀM ROW_NUMBER()

```
ROW_NUMBER ( ) OVER ( [ PARTITION BY  Columns] ORDER BY Column ASC|DESC ) AS Row_num
```

- Lấy danh sách các đơn hàng (order_id) dựa vào ngày giao hàng (shipping_date) và cho biết đơn hàng đó là đơn hàng thứ mấy được ship đi.
- Kết quả hiển thị cần các cột: order_id, order_date, row_num (thứ tự ship hàng)

```
Select order_id, shipping_date,
ROW_NUMBER ( ) OVER ( ORDER BY shipping_date ASC )
AS Row_num
from orders
```

Results | Messages

	order_id	shipping_date	Row_num
1	28774	2009-01-02 00:00:00.000	1
2	37537	2009-01-02 00:00:00.000	2
3	44069	2009-01-02 00:00:00.000	3
4	13729	2009-01-03 00:00:00.000	4
5	9285	2009-01-04 00:00:00.000	5
6	9762	2009-01-04 00:00:00.000	6
7	7909	2009-01-04 00:00:00.000	7
8	44069	2009-01-04 00:00:00.000	8

gitiho

1 Chủ đề XẾP HẠNG (RANKING TOPIC)

HÀM DENSE_RANK()

```
DENSE_RANK ( ) OVER ( [ PARTITION BY Columns ] ORDER BY Columns ASC|DESC )  AS dense_rank_col
```

- Xếp hạng giá trị lợi nhuận tăng dần của mỗi đơn hàng ở mỗi tỉnh
- Kết quả hiển thị gồm các cột: province, order_id, profit, dense_rank_col (Cột xếp hạng)

```
SELECT
province, order_id, profit,
DENSE_RANK() OVER ( PARTITION BY province ORDER BY profit ASC )
AS dense_rank_col
from orders
```

Results | Messages

	province	order_id	profit	dense_rank_col
1	Alberta	55490	-38.2592	1
2	Alberta	9221	-37.0075	2
3	Alberta	54949	-33.9382	3
4	Alberta	52870	-21.0175	4
5	Alberta	774	-20.8208	5
6	Alberta	46533	-12.24	6
7	Alberta	38817	-2.9268	7
8	Alberta	27844	-2.632	8

gitiho

1 Chủ đề XẾP HẠNG (RANKING TOPIC)

HÀM RANK()

```
RANK ( ) OVER ( [ PARTITION BY Columns ] ORDER BY Columns ASC|DESC)
```

- Xếp hạng thứ bậc giá trị lợi nhuận tăng dần của mỗi đơn hàng ở mỗi tỉnh
- Kết quả hiển thị gồm các cột: province, order_id, profit, rank_col (Cột xếp hạng)

```
SELECT
province, order_id, profit,
RANK() OVER ( PARTITION BY province ORDER BY profit ASC )
AS rank_col
from orders
```

Results | Messages

	province	order_id	profit	dense_rank_col
1	Alberta	55490	-38.2592	1
2	Alberta	9221	-37.0075	2
3	Alberta	54949	-33.9382	3
4	Alberta	52870	-21.0175	4
5	Alberta	774	-20.8208	5
6	Alberta	46533	-12.24	6
7	Alberta	38817	-2.9268	7
8	Alberta	27844	-2.632	8

gitiho

2 THỰC HÀNH

Bài 1:

Lấy ra danh sách các đơn hàng (order_id) có lợi nhuận (profit) xếp hạng cao nhất (xếp hạng thứ nhất) của mỗi tỉnh

Gợi ý: Sử dụng dense_rank() để xếp hạng

Kết quả như ảnh:

Results | Messages

	province	order_id	profit	dense_rank_col
1	Alberta	29187	16074.0755	1
2	British Columbia	41895	16215.4636	1
3	Manitoba	48800	16495.4628	1
4	New Brunswick	29766	53764.8586	1
5	Newfoundland	14727	3349.4	1
6	Northwest Territories	52321	15193.4925	1
7	Nova Scotia	2208	13259.2886	1
8	Nunavut	1761	6813.3475	1

gitiho

2 THỰC HÀNH

Bài 2:

Lấy danh sách top 3 loại sản phẩm có tổng lợi nhuận cao nhất ở mỗi tỉnh

Gợi ý: Sử dụng row_number() để xếp hạng

Kết quả như ảnh:

Results | Messages

	province	product_category	total_profit	row_num
1	Alberta	Technology	409896.6965	1
2	Alberta	Furniture	171780.9896	2
3	Alberta	Office Supplies	151299.7148	3
4	British Columbia	Technology	357901.3043	1
5	British Columbia	Furniture	205175.4446	2
6	British Columbia	Office Supplies	199583.7439	3
7	Manitoba	Technology	245513.2356	1
8	Manitoba	Furniture	168960.2275	2

gitiho

3 TÓM LẠI

RANK()	DENSE_RANK()	ROW_NUMBER()
Xếp hạng theo thứ bậc tăng dần, các giá trị giống nhau có cùng thứ bậc, giá trị tiếp theo có thể nhảy n bậc nếu thứ bậc trước có n giá trị giống nhau	Xếp hạng theo thứ tự tăng dần, các giá trị giống nhau có thứ hạng bằng nhau.	Xếp hạng theo thứ tự tăng dần không quan tâm đến các giá trị giống nhau

Value	Rank	
1	1	
2	2	Cùng giá trị, cùng bậc
2	2	
3	4	Nhảy bậc
4	5	

Value	Dense_Rank	
1	1	
2	2	Cùng giá trị, cùng xếp hạng
2	2	
3	3	
4	4	

Value	Row_num
1	1
2	2
2	3
3	4
4	5

gitiho

Bài tập thực hành:

Bài 1: Sử dụng row_number ()
Lấy danh sách top 3 sản phẩm (product_name) có tổng lợi nhuận (total_profit) nhỏ nhất của mỗi loại sản phẩm (product_category).

Results | Messages

	product_category	product_name	total_profit	row_num
1	Furniture	DAX Wood Document Frame	52.795	1
2	Furniture	Eldon® Executive Woodline II Desk Accessories, Ma...	58.7349	2
3	Furniture	DAX Charcoal/Nickel-Tone Document Frame, 5 x 7	63.66	3
4	Office Supplies	Hoover® Commercial Lightweight Upright Vacuum	-175.4	1
5	Office Supplies	Hoover Portapower™ Portable Vacuum	-151.1776	2
6	Office Supplies	Bravo II™ Megaboss® 12-Amp Hard Body Upright, Re...	-134.37	3
7	Technology	Sony IBM Color Diskettes, 25/Pack	3.0494	1
8	Technology	Imation 3.5 IBM Diskettes, 10/Box	73.7886	2
9	Technology	DS/HD IBM Formatted Diskettes, 10/Pack - Staples	81.6726	3

Bài tập về nhà:

Bài 2: Sử dụng dense_rank ()
Xác định những sản phẩm (product_name) có tổng lợi nhuận xếp hạng 3 (từ trên xuống) của mỗi tỉnh

Results | Messages

	province	product_name	total_profit	denseRank
1	Alberta	Hewlett-Packard Business Color Inkjet 3000 [N, DTN]...	22616.0539	3
2	British Columbia	Canon imageCLASS 2200 Advanced Copier	16215.4636	3
3	Manitoba	Riverside Palais Royal Lawyers Bookcase, Royale Ch...	17891.7018	3
4	New Brunswick	Polycom ViewStation™ Adapter H323 Videoconferenc...	12776.942	3
5	Newfoundland	Bretford CR8500 Series Meeting Room Furniture	2864.585	3
6	Northwest Territories	SAFCO PlanMaster Heigh-Adjustable Drafting Table B...	20836.5165	3
7	Nova Scotia	Canon PC1060 Personal Laser Copier	11847.3404	3
8	Nunavut	Global Troy™ Executive Leather Low-Back Tilter	4633.114	3
9	Ontario	Bretford CR8500 Series Meeting Room Furniture	23108.8634	3
10	Prince Edward Island	Safco Contoured Stacking Chairs	6412.31	3
11	Quebec	Hewlett Packard LaserJet 3310 Copier	21838.6552	3

Bài tập về nhà:

Bài 3: Sử dụng rank ()

Xác định những sản phẩm (product_name) có tổng lợi nhuận xếp ví trí thứ 3 (từ dưới lên) của mỗi tỉnh

Results | Messages

	province	product_name	total_profit	Rankk
1	Alberta	Hoover Portapower™ Portable Vacuum	-20.8208	3
2	British Columbia	Eldon Wave Desk Accessories	-2.0949	3
3	Manitoba	Array® Parchment Paper, Assorted Colors	-3.142	3
4	New Brunswick	Tennsco Commercial Shelving	-21.5756	3
5	Newfoundland	Xerox 1984	-0.704800000000001	3
6	Northwest Territories	Imation 3.5 IBM Diskettes, 10/Box	1.162	3
7	Nova Scotia	Tennsco Lockers, Gray	-6.61480000000003	3
8	Nunavut	Eldon Base for stackable storage shelf, platinum	2.38239999999996	3
9	Ontario	Rogers Handheld Barrel Pencil Sharpener	-1.3628	3
10	Prince Edward Island	Recycled Eldon Regeneration Jumbo File	-1.4636	3
11	Quebec	Hoover Portapower™ Portable Vacuum	-4.73759999999999	3

BÀI 21. RÀNG BUỘC TRONG SQL

gitiho

1 KHÓA CHÍNH (PRIMARY KEY) & KHÓA NGOẠI (FOREIGN KEY)

Primary Key: Khóa chính được sử dụng để đảm bảo dữ liệu trong cột cụ thể là duy nhất. Đây là một cột không thể có giá trị NULL. Nó là một cột được tạo cụ thể bởi CSDL theo một trình tự xác định.

Poreign Key: Khóa ngoại là trường (cột) dữ liệu chứa các giá trị tham chiếu (trỏ đến) một khóa chính của bảng khác. Việc tạo ra các cột (trường) là khóa ngoại sẽ hình thành nên mối quan hệ giữa các bảng.

=> Bảng có khóa ngoại được gọi là bảng con và bảng có khóa chính được gọi là bảng tham chiếu hoặc bảng cha

Persons Table

PRIMARY KEY

PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

Orders Table

FOREIGN KEY

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

gitiho

1

KHÓA CHÍNH (PRIMARY KEY) & KHÓA NGOẠI (FOREIGN KEY)

Query như sau:

```
--Create Persons Table
CREATE TABLE Persons
 (
PersonID int NOT NULL PRIMARY KEY, --define primary key
LastName varchar (50),
 FirstName varchar(100),
Age int
)


 --Create Orders Table
CREATE TABLE Orders
 (
OrderID int NOT NULL PRIMARY KEY, --define primary key
OrderNumber int NOT NULL,
PersonID int FOREIGN KEY REFERENCES Persons(PersonID)--define foreign key
 )
```

gitiho

BÀI 22. KẾT NỐI SQL TỚI EXCEL & POWER BI

gitiho

1 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO EXCEL

Có hai kiểu lấy dữ liệu

- Một là lấy dữ liệu từ bảng có sẵn
- Hai là lấy dữ liệu từ câu truy vấn

Các bước cơ bản kết nối dữ liệu bao gồm:

B1: Chọn nguồn dữ liệu cần kết nối

B2: Điền các thống tin server để kết nối

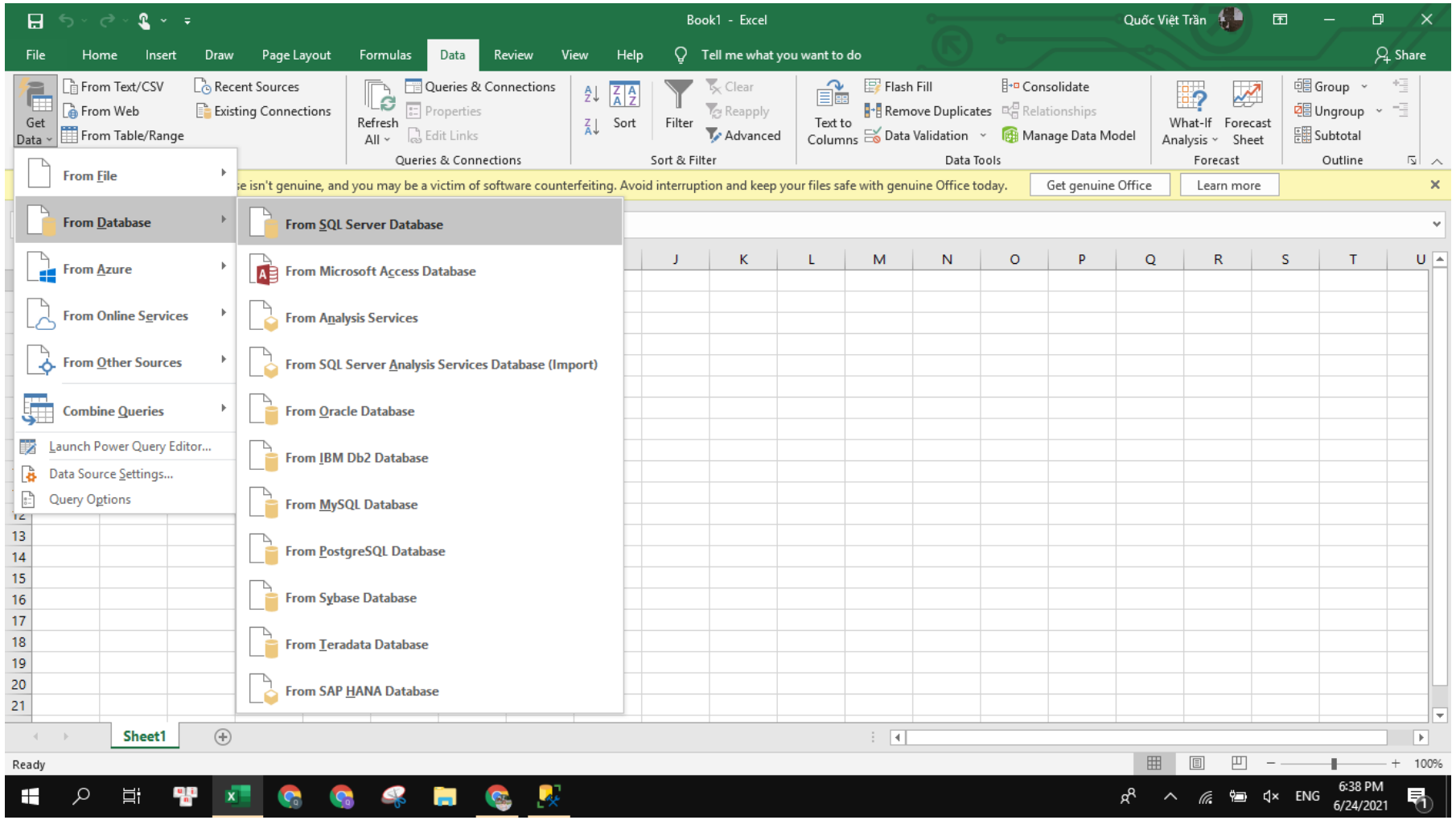
B3: Chọn database và bảng cần lấy hoặc câu truy vấn cần lấy dữ liệu

gitiho

1 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO EXCEL

1. Lấy dữ liệu từ SQL Server

Data -> Get Data -> From database -> From SQL Server database

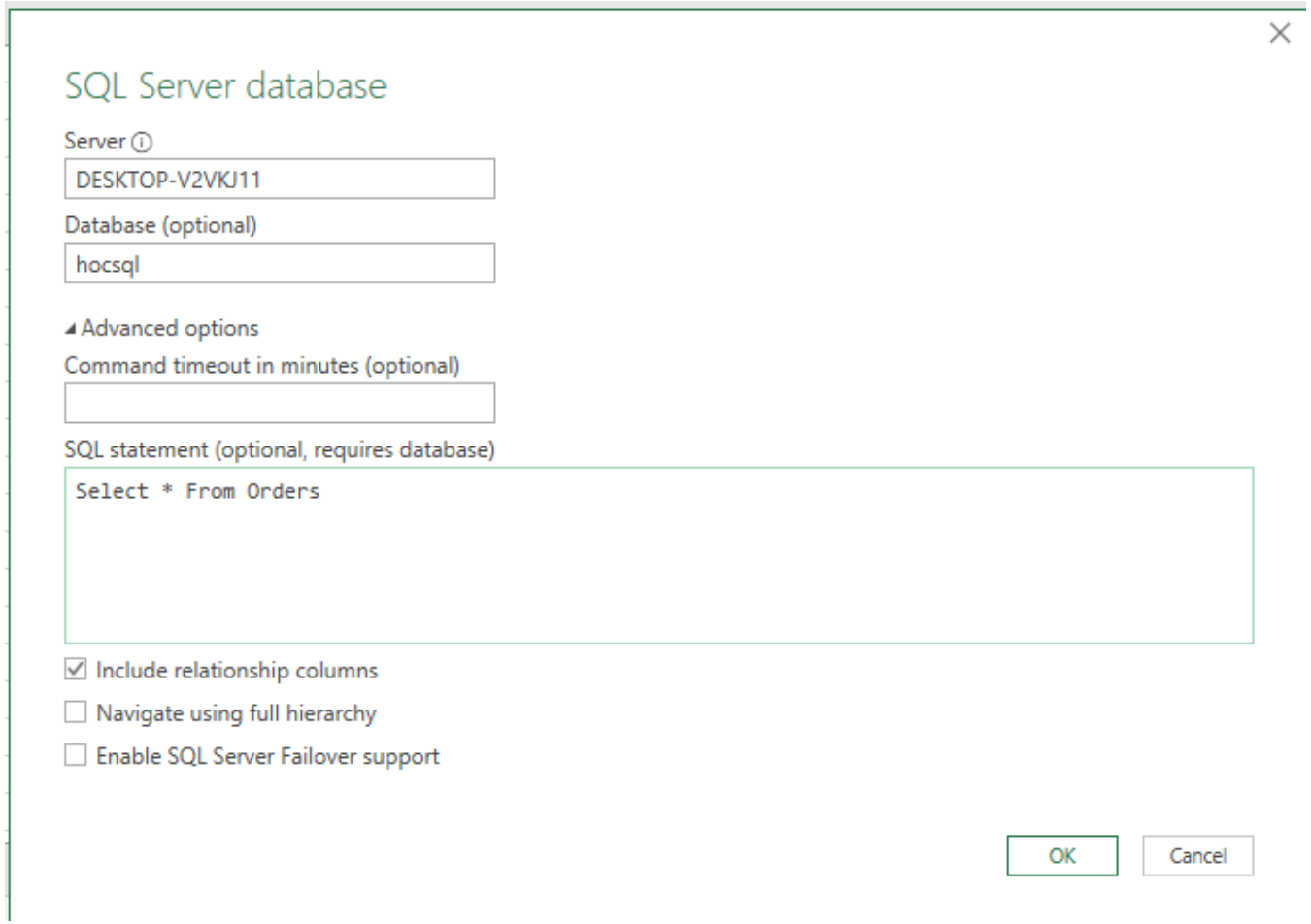


gitiho

1 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO EXCEL

1. Lấy dữ liệu từ SQL Server

Bước 2: Điền thông tin Server và Database -> Chọn Advance -> Điền câu truy vấn -> OK



gitiho

1 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO EXCEL

1. Lấy dữ liệu từ SQL Server

Bước 3: Kiểm tra bảng dữ liệu cần lấy từ query -> Load

DESKTOP-V2VKJ11: hocsql

ID	order_id	order_date	order_priority	order_quantity	value	discount	shipping_mode	profit	unit_price	sh
1	3	10/13/2010 12:00:00 AM	Low	6	224.2944	0.04	Regular Air	2.3824	38.94	
2	6	2/20/2012 12:00:00 AM	Not Specified	2	4.1184	0.01	Regular Air	-0.7296	2.08	
3	32	7/15/2011 12:00:00 AM	High	26	2600.0754	0.07	Regular Air	777.0084	107.53	
4	32	7/15/2011 12:00:00 AM	High	24	1548.2376	0.09	Delivery Truck	233.9584	70.89	
5	32	7/15/2011 12:00:00 AM	High	23	176.4192	0.04	Regular Air	61.1272	7.99	
6	32	7/15/2011 12:00:00 AM	High	15	121.824	0.04	Regular Air	12.583	8.46	
7	35	10/22/2011 12:00:00 AM	Not Specified	30	265.101	0.03	Regular Air	120.735	9.11	
8	35	10/22/2011 12:00:00 AM	Not Specified	14	2162.0214	0.01	Regular Air	886.3926	155.99	
9	36	11/2/2011 12:00:00 AM	Critical	46	2731.986	0.1	Regular Air	1058.239	65.99	
10	65	3/17/2011 12:00:00 AM	Critical	32	3631.1744	0.02	Regular Air	1813.5972	115.79	
11	66	1/19/2009 12:00:00 AM	Low	41	107.4528	0.09	Regular Air	40.628	2.88	
12	69	6/3/2009 12:00:00 AM	Not Specified	42	1182.1446	0.09	Regular Air	606.6382	30.93	
13	69	6/3/2009 12:00:00 AM	Not Specified	28	45.6288	0.03	Express Air	16.7048	1.68	
14	70	12/17/2010 12:00:00 AM	Low	48	86.6016	0.03	Regular Air	10.812	1.86	
15	70	12/17/2010 12:00:00 AM	Low	46	9001.763	0.05	Regular Air	3405.2044	205.99	
16	96	4/16/2009 12:00:00 AM	High	37	4615.0137	0.01	Regular Air	2042.1272	125.99	
17	97	1/28/2010 12:00:00 AM	Medium	26	72.8858	0.03	Regular Air	43.8326	2.89	
18	129	11/18/2012 12:00:00 AM	Low	4	23.5872	0.09	Regular Air	5.8068	6.48	
19	130	5/7/2012 12:00:00 AM	High	3	430.293	0.05	Express Air	244.1858	150.98	
20	130	5/7/2012 12:00:00 AM	High	29	539.1274	0.02	Regular Air	326.5493	18.97	

The data in the preview has been truncated due to size limits.

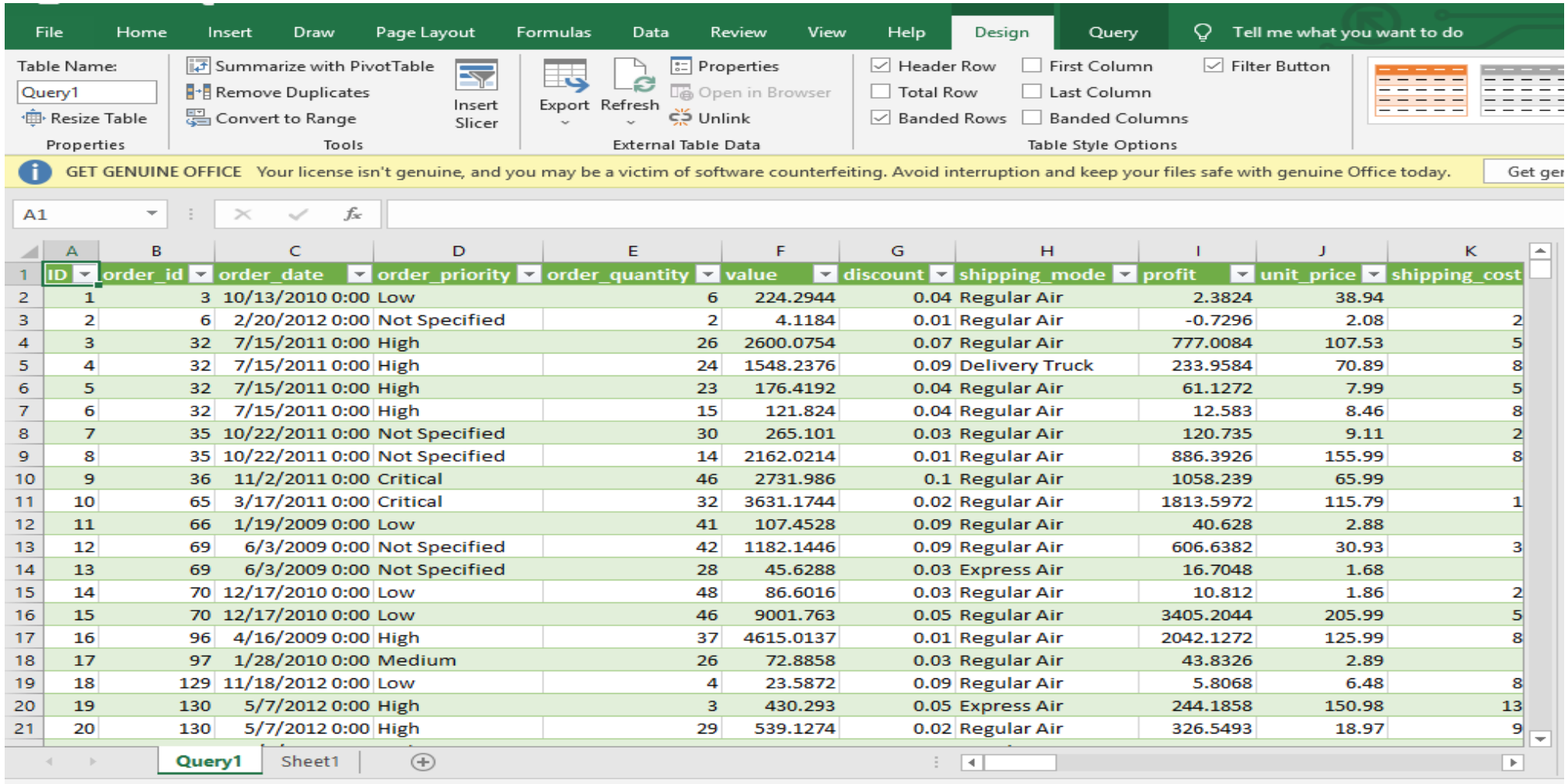
Load | Transform Data | Cancel

gitiho

1 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO EXCEL

1. Lấy dữ liệu từ SQL Server

Kết quả:



ID	order_id	order_date	order_priority	order_quantity	value	discount	shipping_mode	profit	unit_price	shipping_cost
1	3	10/13/2010 0:00	Low	6	224.2944	0.04	Regular Air	2.3824	38.94	
2	6	2/20/2012 0:00	Not Specified	2	4.1184	0.01	Regular Air	-0.7296	2.08	2
3	32	7/15/2011 0:00	High	26	2600.0754	0.07	Regular Air	777.0084	107.53	5
4	32	7/15/2011 0:00	High	24	1548.2376	0.09	Delivery Truck	233.9584	70.89	8
5	32	7/15/2011 0:00	High	23	176.4192	0.04	Regular Air	61.1272	7.99	5
6	32	7/15/2011 0:00	High	15	121.824	0.04	Regular Air	12.583	8.46	8
7	35	10/22/2011 0:00	Not Specified	30	265.101	0.03	Regular Air	120.735	9.11	2
8	35	10/22/2011 0:00	Not Specified	14	2162.0214	0.01	Regular Air	886.3926	155.99	8
9	36	11/2/2011 0:00	Critical	46	2731.986	0.1	Regular Air	1058.239	65.99	
10	65	3/17/2011 0:00	Critical	32	3631.1744	0.02	Regular Air	1813.5972	115.79	1
11	66	1/19/2009 0:00	Low	41	107.4528	0.09	Regular Air	40.628	2.88	
12	69	6/3/2009 0:00	Not Specified	42	1182.1446	0.09	Regular Air	606.6382	30.93	3
13	69	6/3/2009 0:00	Not Specified	28	45.6288	0.03	Express Air	16.7048	1.68	
14	70	12/17/2010 0:00	Low	48	86.6016	0.03	Regular Air	10.812	1.86	2
15	70	12/17/2010 0:00	Low	46	9001.763	0.05	Regular Air	3405.2044	205.99	5
16	96	4/16/2009 0:00	High	37	4615.0137	0.01	Regular Air	2042.1272	125.99	8
17	97	1/28/2010 0:00	Medium	26	72.8858	0.03	Regular Air	43.8326	2.89	
18	129	11/18/2012 0:00	Low	4	23.5872	0.09	Regular Air	5.8068	6.48	8
19	130	5/7/2012 0:00	High	3	430.293	0.05	Express Air	244.1858	150.98	13
20	130	5/7/2012 0:00	High	29	539.1274	0.02	Regular Air	326.5493	18.97	9

gitiho

2 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO POWER BI

Có hai kiểu lấy dữ liệu:

- Một là lấy dữ liệu từ bảng có sẵn
- Hai là lấy dữ liệu từ câu truy vấn

Các bước cơ bản kết nối dữ liệu:

B1: Chọn nguồn dữ liệu cần kết nối

B2: Điền các thông tin server để kết nối

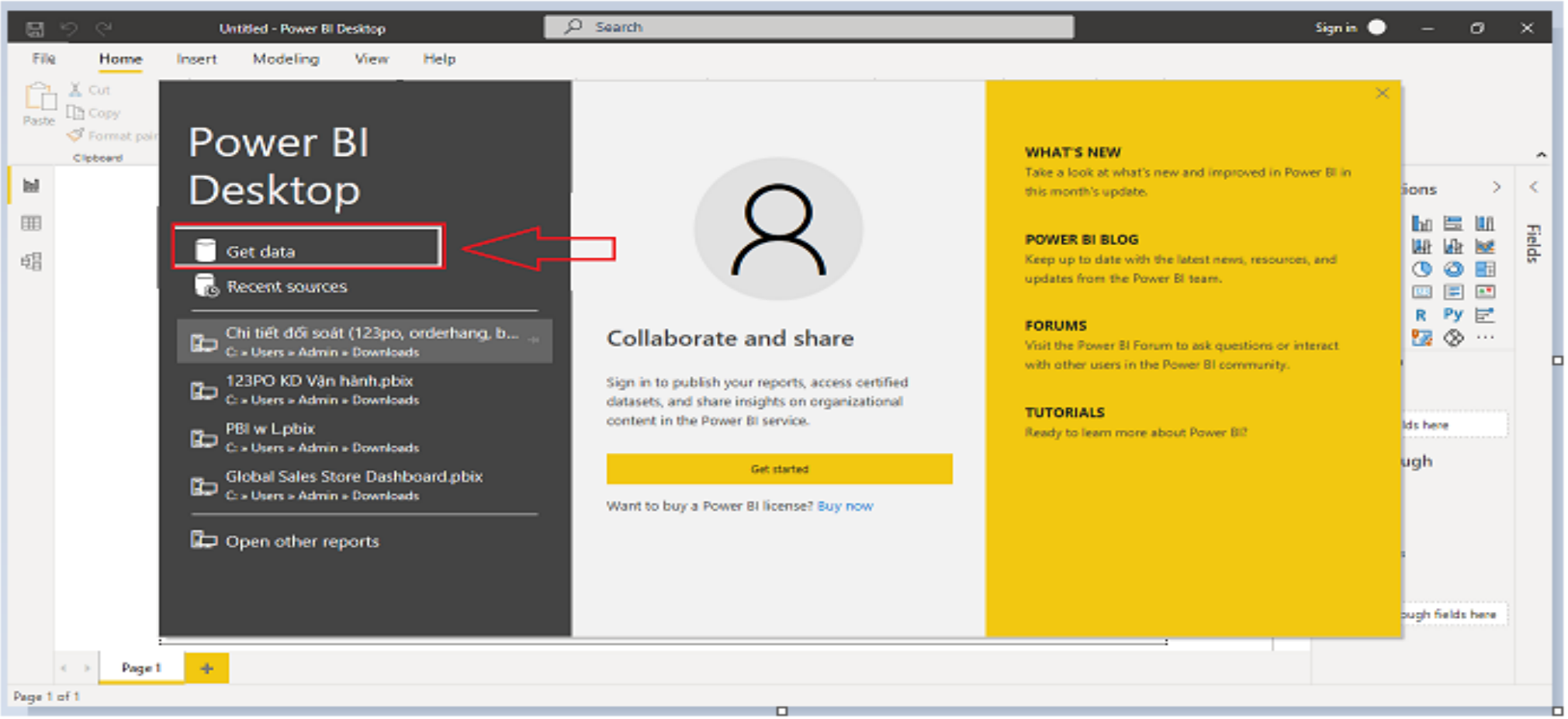
B3: Chọn database và bảng cần lấy hoặc câu truy vấn cần lấy dữ liệu

gitiho

2 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO POWER BI

Đầu tiên là bước chọn nguồn dữ liệu: có thể lấy từ SQL Server, My SQL, Azure, Big Query, Excel, Web...
Tuy nhiên trong trường hợp này là hướng dẫn kết nối với SQL Server:

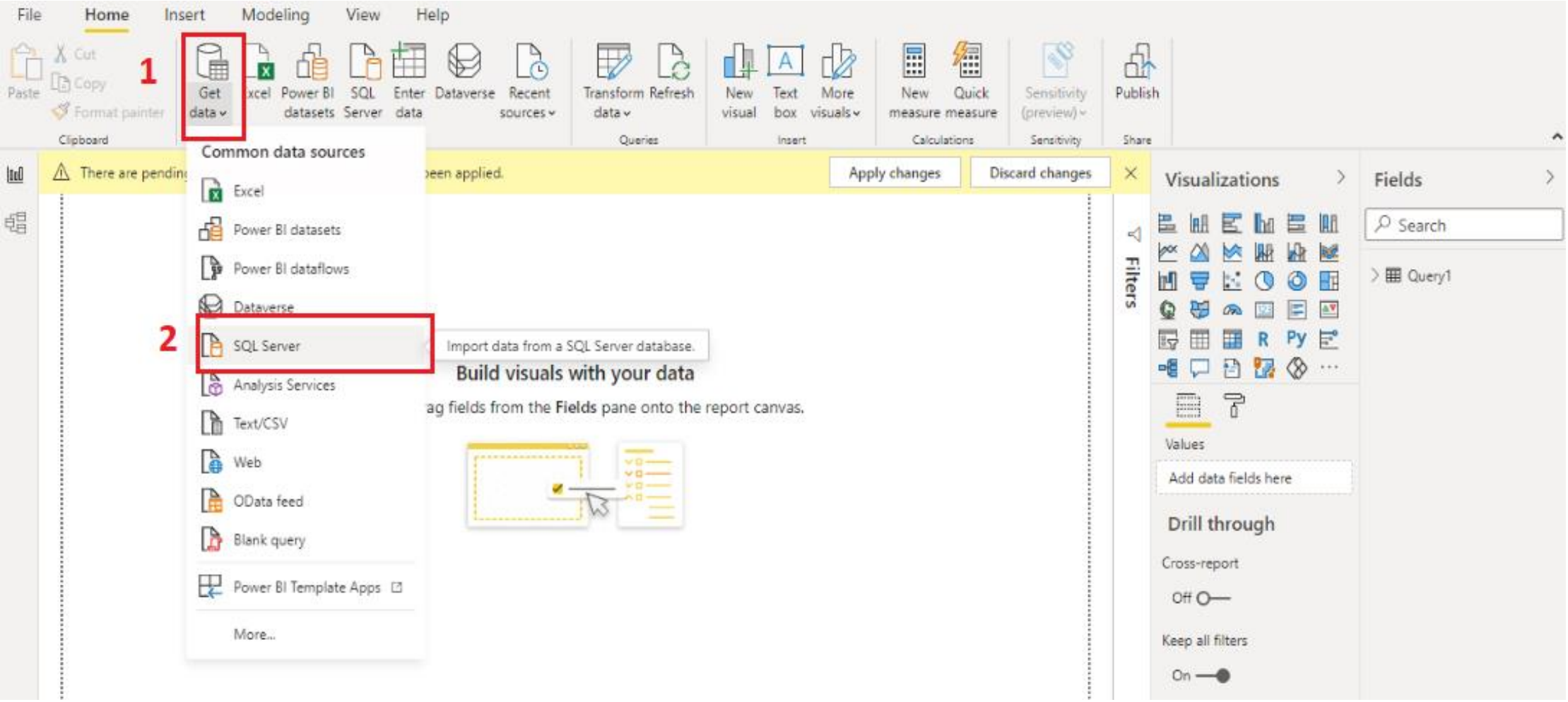
Sau khi bật PBI sẽ hiển thị hộp thoại này => Chọn Get Data



gitiho

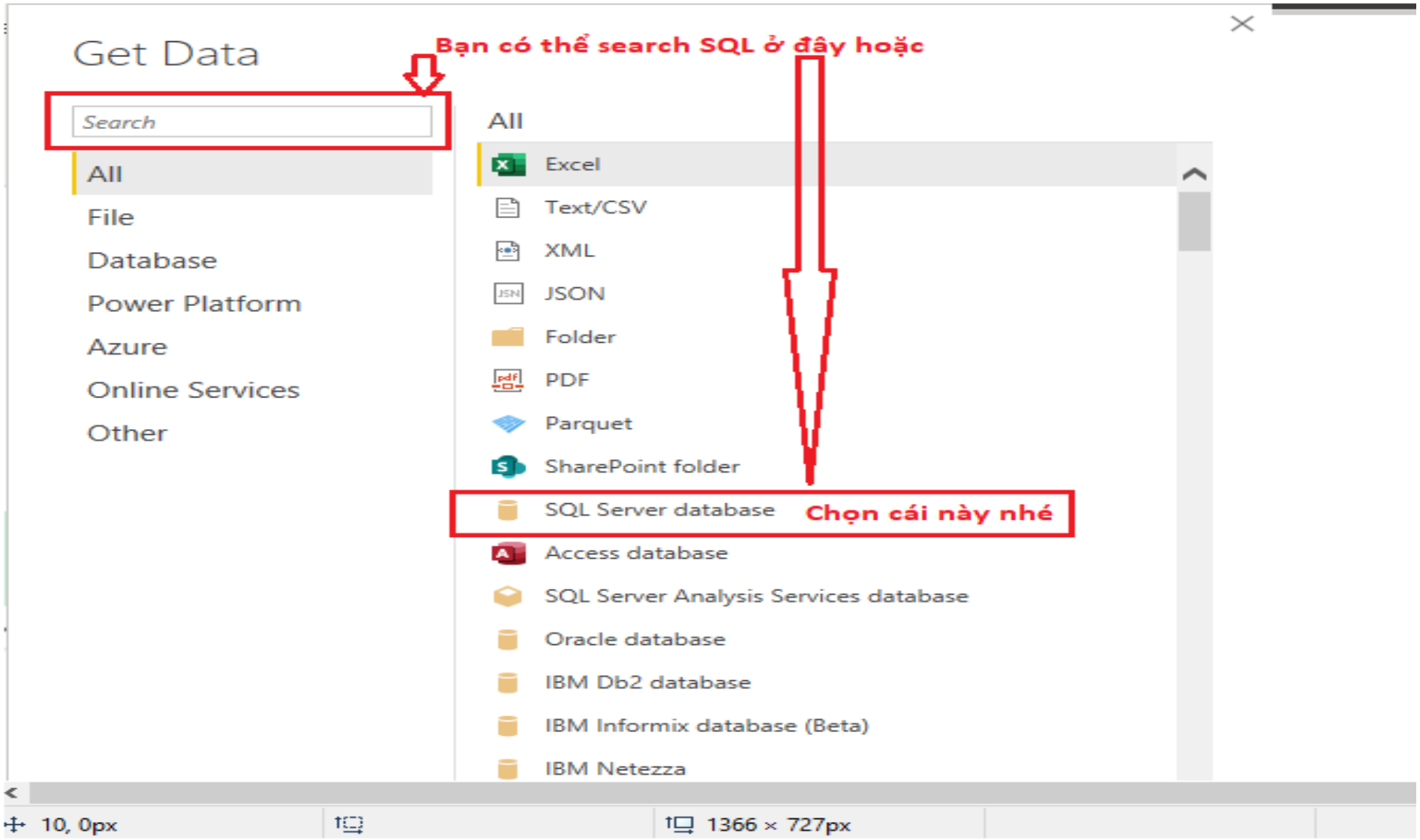
2 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO POWER BI

Hoặc làm theo hướng dẫn như hình bên dưới:



gitiho

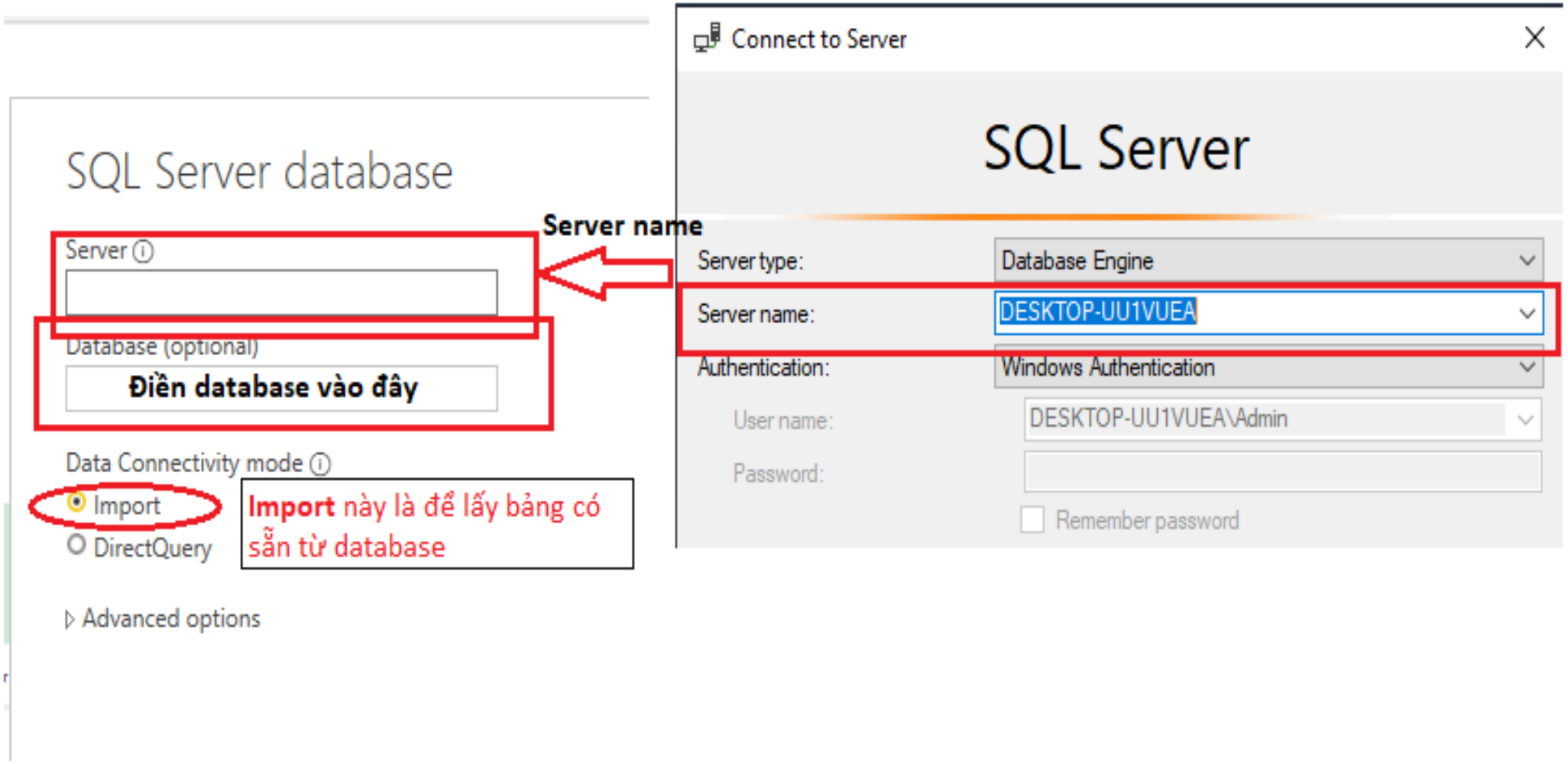
2 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO POWER BI



gitiho

2 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO POWER BI

Sau khi chọn nguồn cần kết nối, bạn sẽ điền thông tin về server và database. Nếu chọn bảng từ database có sẵn thì chọn import

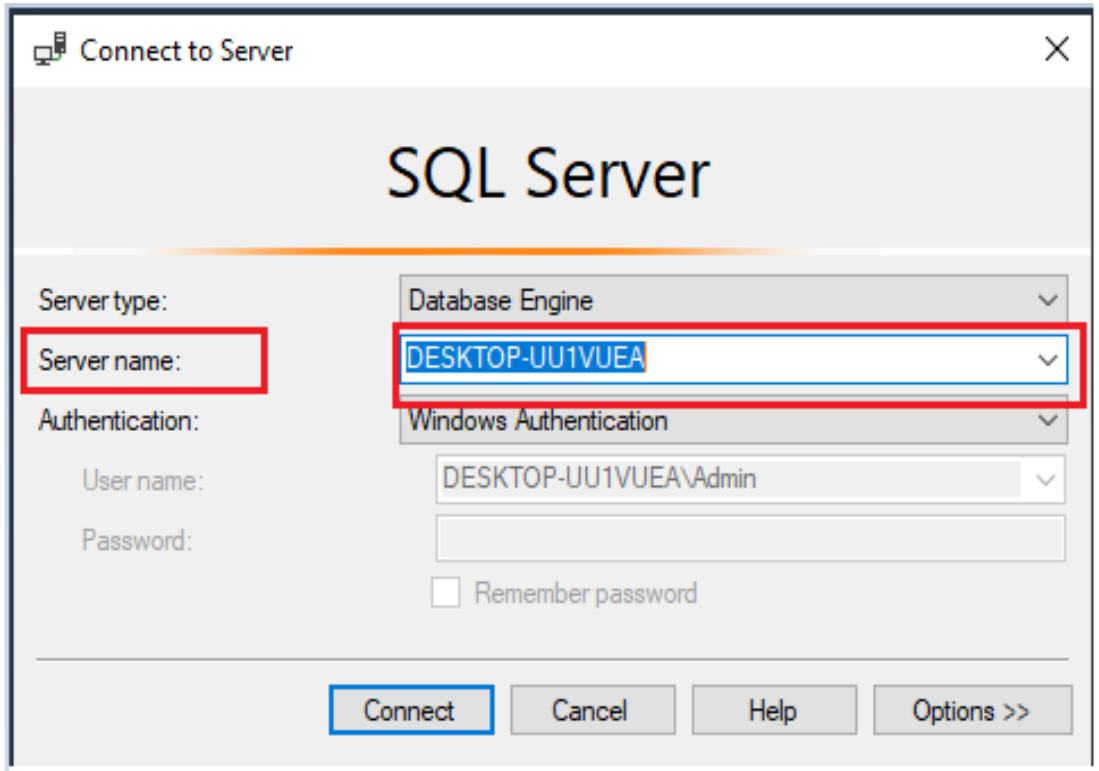


gitiho

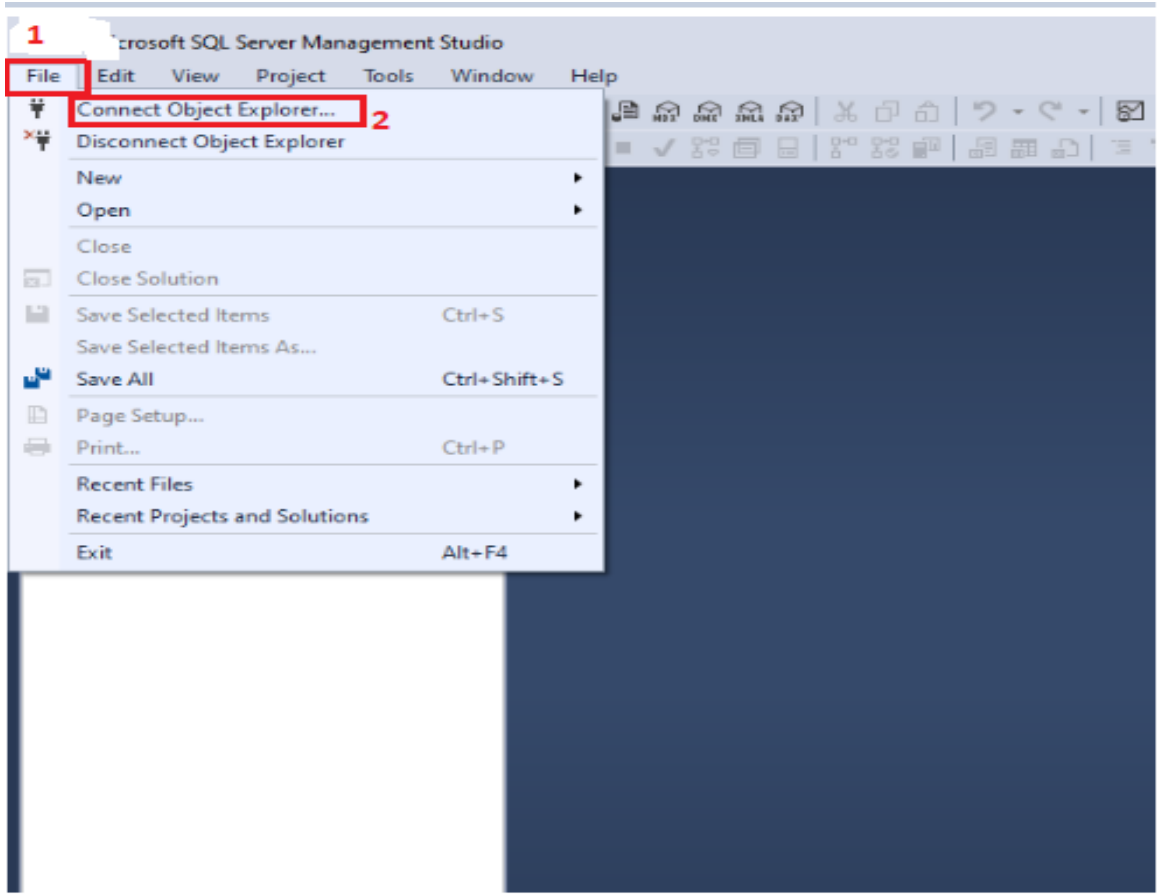
2 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO POWER BI

Bình thường khi làm việc bạn sẽ được cung cấp servername để kết nối. Tuy nhiên bạn có thể test thử kết nối trực tiếp với dữ liệu trên SQL Server của bạn. Xem hướng dẫn bên dưới.

Khi mở SSMS lên sẽ xuất hiện hộp thoại bên dưới, nó chứa serverame để có thể kết nối



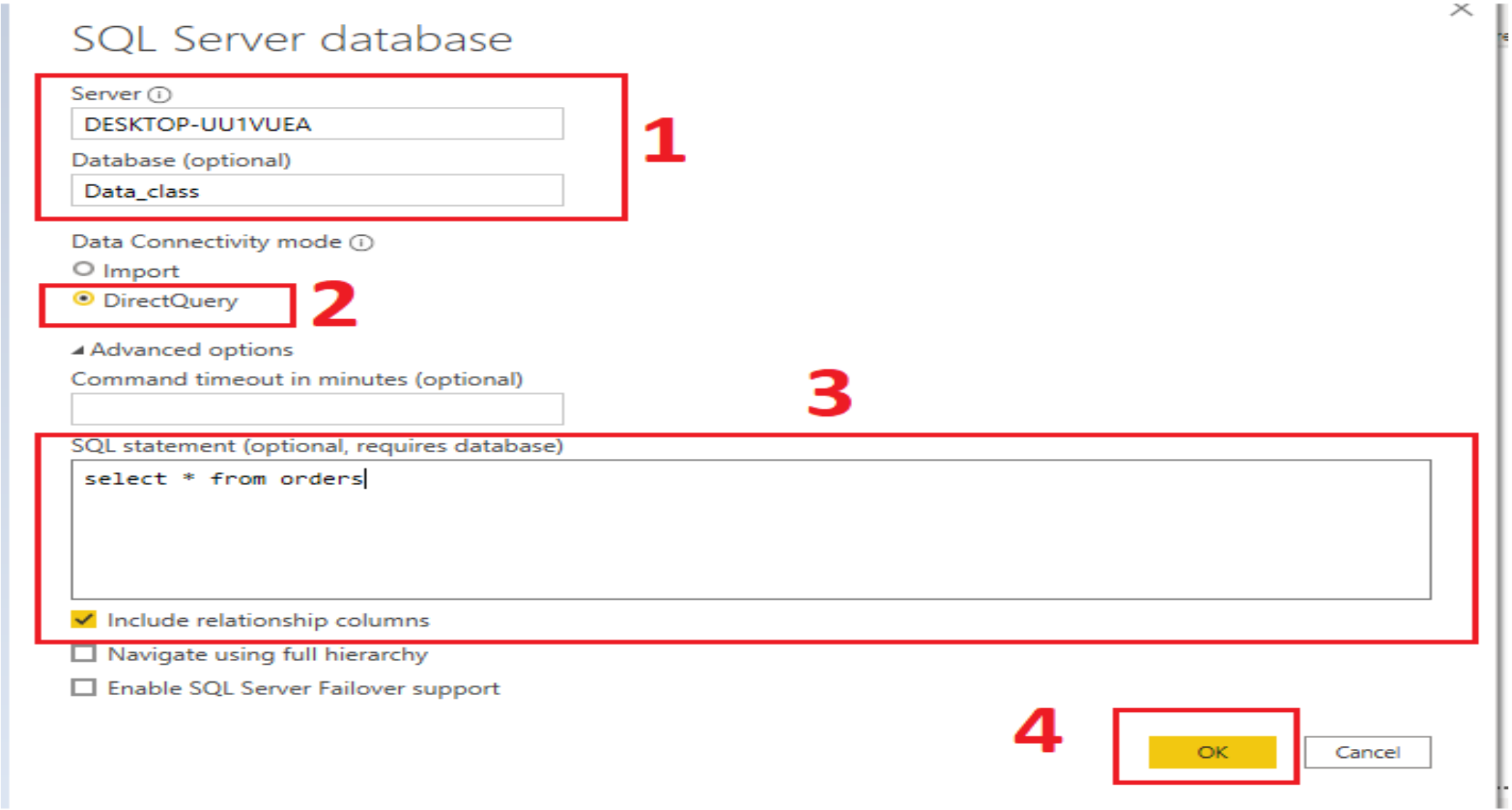
Hoặc có thể vào File/Connect Object Explorer



gitiho

2 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO POWER BI

Chọn DirectQuery nếu dữ liệu lấy từ một câu truy vấn



MAGIC CODE INSTITUTE

2 KẾT NỐI TỚI SQL SEVER ĐỂ LẤY DỮ LIỆU VÀO POWER BI

Sau đó sẽ xuất hiện một hộp thoại như hình => Chọn Load
=> Xong các bước kết nối với Power BI

DESKTOP-UU1VUEA: Data_class

ID	order_id	order_date	order_priority	order_quantity	value	discount	shipping_mode	profit	unit_price	sh
1	3	10/13/2010 12:00:00 AM	Low	6	224.2944	0.04	Regular Air	2.3824	38.94	
2	6	2/20/2012 12:00:00 AM	Not Specified	2	4.1184	0.01	Regular Air	-0.7296	2.08	
3	32	7/15/2011 12:00:00 AM	High	26	2600.0754	0.07	Regular Air	777.0084	107.53	
4	32	7/15/2011 12:00:00 AM	High	24	1548.2376	0.09	Delivery Truck	233.9584	70.89	
5	32	7/15/2011 12:00:00 AM	High	23	176.4192	0.04	Regular Air	61.1272	7.99	
6	32	7/15/2011 12:00:00 AM	High	15	121.824	0.04	Regular Air	12.583	8.46	
7	35	10/22/2011 12:00:00 AM	Not Specified	30	265.101	0.03	Regular Air	120.735	9.11	
8	35	10/22/2011 12:00:00 AM	Not Specified	14	2162.0214	0.01	Regular Air	886.3926	155.99	
9	36	11/2/2011 12:00:00 AM	Critical	46	2731.986	0.1	Regular Air	1058.239	65.99	
10	65	3/17/2011 12:00:00 AM	Critical	32	3631.1744	0.02	Regular Air	1813.5972	115.79	
11	66	1/19/2009 12:00:00 AM	Low	41	107.4528	0.09	Regular Air	40.628	2.88	
12	69	6/3/2009 12:00:00 AM	Not Specified	42	1182.1446	0.09	Regular Air	606.6382	30.93	
13	69	6/3/2009 12:00:00 AM	Not Specified	28	45.6288	0.03	Express Air	16.7048	1.68	
14	70	12/17/2010 12:00:00 AM	Low	48	86.6016	0.03	Regular Air	10.812	1.86	
15	70	12/17/2010 12:00:00 AM	Low	46	9001.763	0.05	Regular Air	3405.2044	205.99	
16	96	4/16/2009 12:00:00 AM	High	37	4615.0137	0.01	Regular Air	2042.1272	125.99	
17	97	1/28/2010 12:00:00 AM	Medium	26	72.8858	0.03	Regular Air	43.8326	2.89	
18	129	11/18/2012 12:00:00 AM	Low	4	23.5872	0.09	Regular Air	5.8068	6.48	
19	130	5/7/2012 12:00:00 AM	High	3	430.293	0.05	Express Air	244.1858	150.98	
20	130	5/7/2012 12:00:00 AM	High	29	539.1274	0.02	Regular Air	326.5493	18.97	

The data in the preview has been truncated due to size limits.

Load | Transform Data | Cancel

gitiho