CHAPTER 4: Câu lệnh JOIN & UNION trong SQL 2
4.1. Kết hợp bảng sử dụng JOIN **2**
4.1.1. Khóa chính và khóa ngoại 2
4.1.1.1. Khóa chính (Primary Key – PK) 2
4.1.1.2. Khoá ngoại (Foreign Key - FK) 2
4.1.2. Các loại JOIN 3
4.1.2.1. Kết hợp điểm chung (Inner join) 3
4.1.2.2. Kết hợp trái (Left join) 6
4.1.2.3. Kết hợp phải (Right join) 8
4.1.2.4. Kết hợp chéo (Cross join) 10
4.1.2.5. Kết hợp tất cả (Outer join/Full Outer Join) 12
4.1.2.6. Kết hợp với nhiều hơn 2 bảng 13
4.1.2.7. Kết hợp với chính nó (Self Join) 15
4.2. Các lưu ý khi viết JOIN **16**
Lưu ý 1: Về Alias tên bảng 16
Lưu ý 2: Lỗi Ambigous column name 17
4.3. Kết hợp bảng sử dụng UNION **18**
4.3.1. UNION và UNION ALL 18
4.3.1.1. UNION 18
4.3.1.2. UNION ALL 22
4.3.2. Một số lý thuyết tập hợp (Set theory) khác 25
4.3.2.1. Intersect 25
4.3.2.2. Exception 27

© 2023 Datapot. All right reserved.

CHAPTER 4: Câu lệnh JOIN & UNION trong SQL

4.1. Kết hợp bảng sử dụng JOIN

Join là mệnh đề trong SQL, được dùng để kết nối dữ liệu từ hai hay nhiều bảng lại với nhau. JOIN cho phép truy vấn các cột dữ liệu từ nhiều bảng khác nhau để trả về trong cùng một tập kết quả.

JOIN thường được sử dụng để kết hợp dữ liệu trong cơ sở dữ liệu có quan hệ (Relational Database).

4.1.1. Khóa chính và khóa ngoại

4.1.1.1. Khóa chính (Primary Key – PK)

Khóa chính được sử dụng để định danh từng dòng dữ liệu trong bảng.

Đặc điểm:

- Khoá chính chứa giá trị không trùng lặp.
- Mỗi bảng chỉ có một khóa chính duy nhất (Khoá chính có thể là một trường hoặc tổ hợp các trường).
- Khóa chính không được chứa NULL.

4.1.1.2. Khoá ngoại (Foreign Key - FK)

Khóa ngoại chứa giá trị tham chiếu dùng để liên kết với khóa chính của bảng khác.

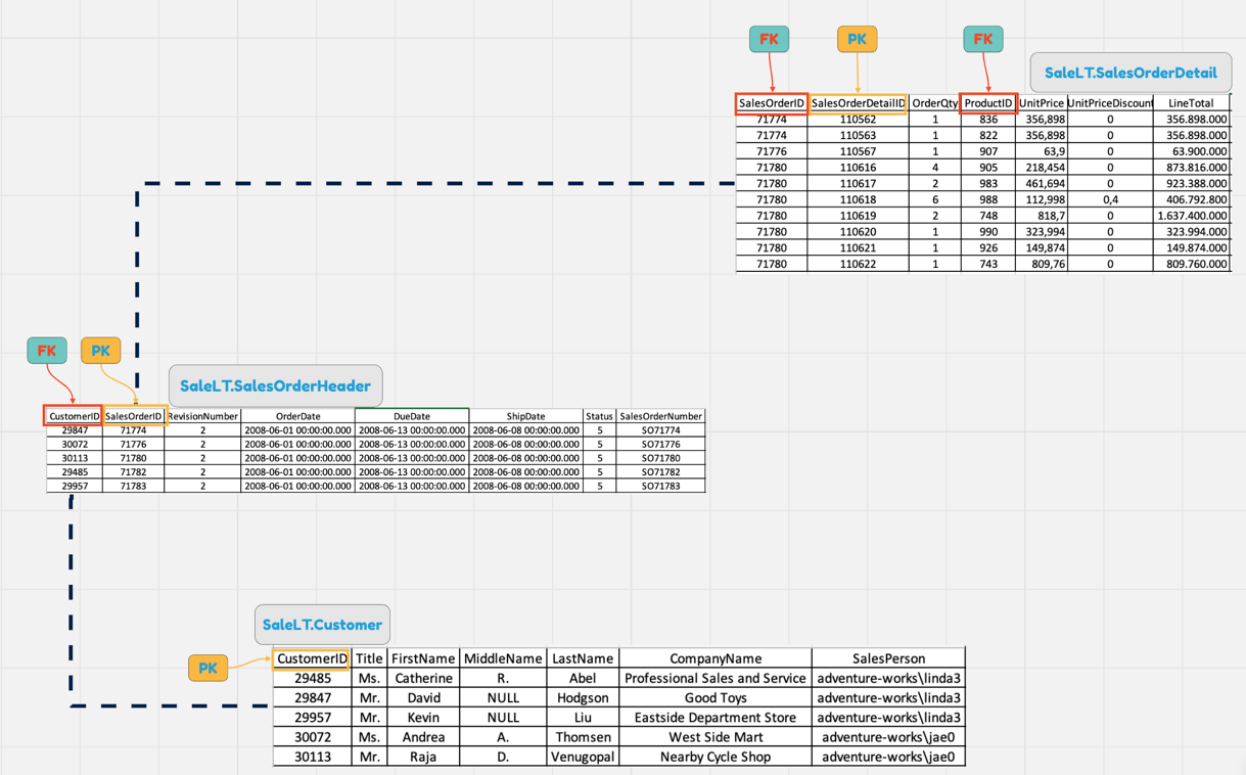
Đặc điểm:

- Khoá ngoại có thể là một trường hoặc tổ hợp trường.
- Một bảng dữ liệu có thể có nhiều khóa ngoại trong 1 bảng.
- Khóa ngoại có thể chứa (NULL).

Ví dụ:

Nhìn vào lược đồ (Schema) SaleLT:

© 2023 Datapot. All right reserved.



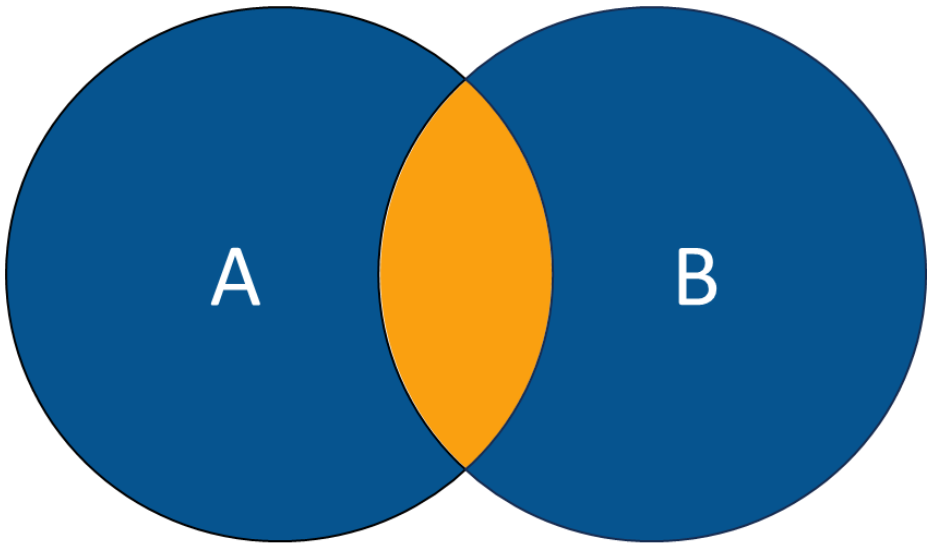
- Bảng Customer có PK là CustomerID.
- SalesOrderDetail có PK là SalesOrderDetailID.
- SalesOrderHeader có PK là SalesOrderID.
- SalesOrderDetail có FK là ProductID và SalesOrderID.
- SalesOrderHeader có FK là AddressID và CustomerID.

4.1.2. Các loại JOIN

4.1.2.1. Kết hợp điểm chung (Inner join)

INNER JOIN trả về kết quả là các bản ghi mà trường được join ở hai bảng khớp nhau, các bản ghi chỉ xuất hiện ở một trong hai bảng sẽ bị loại.

© 2023 Datapot. All right reserved.



INNER JOIN

Cú pháp của INNER JOIN:

```
SELECT
        A.Col_Name,
        B. Col_Name
FROM Table_A
INNER JOIN Table_B
On A.Key = B.Key
```

Ta có:

Bảng Customer gồm 847 hàng.

© 2023 Datapot. All right reserved.

```
USE adventureworks
SELECT *
FROM SalesLT.Customer
```

Results | Messages

	CustomerID	NameStyle	Title	FirstName	MiddleName	LastName
1	1	0	Mr.	Orlando	N.	Gee
2	2	0	Mr.	Keith	*NULL*	Harris
3	3	0	Ms.	Donna	F.	Carrera
4	4	0	Ms.	Janet	M.	Gates
5	5	0	Mr.	Lucy	*NULL*	Harring

0 UTF-8 CRLF SQL 847 rows MSSQL 00:00:00 sql-st.datapot.edu.vn : adventureworks

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng SalesLT.Customer.
- SELECT *: Truy vấn tất cả các hàng trong bảng dữ liệu.

Bảng CustomerAddress gồm 417 hàng.

```
USE adventureworks
SELECT *
FROM SalesLT.CustomerAddress
```

Results | Messages

	CustomerID	AddressID	AddressType	rowguid
1	29485	1086	Main Office	16765338-dbe4-4421-b5e9-3836…
2	29486	621	Main Office	22b3e910-14af-4ed5-8b4d-23bb…
3	29489	1069	Main Office	a095c88b-d7e6-4178-a078-2eca…
4	29490	887	Main Office	f12e1702-d897-4035-b614-0fe2…
5	29492	618	Main Office	5b3b3eb2-3f43-47ed-a20c-2369…

0 UTF-8 CRLF SQL 417 rows MSSQL 00:00:00 sql-st.datapot.edu.vn : adventureworks

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng SalesLT.CustomerAddress.
- SELECT *: Truy vấn tất cả các hàng trong bảng dữ liệu.

Ví dụ: INNER JOIN bảng Customer và CustomerAddress để truy vấn dữ liệu từ cột AddressType.

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    CST.CustomerID,
    CST.FirstName,
    CST.LastName,
    CTA.AddressID,
    CTA.AddressType
FROM SalesLT.Customer AS CST
INNER JOIN SalesLT.CustomerAddress AS CTA
ON CST.CustomerID = CTA.CustomerID
```

Results Messages

	CustomerID	FirstName	LastName	AddressID	AddressType
1	29485	Catherine	Abel	1086	Main Office
2	29486	Kim	Abercrombie	621	Main Office
3	29489	Frances	Adams	1069	Main Office
4	29490	Margaret	Smith	887	Main Office

0 Spaces: 4 UTF-8 CRLF SQL 417 rows MSSQL 00:00:00 sql-st.datapot.edu.vn : a

Giải thích câu lệnh truy vấn:

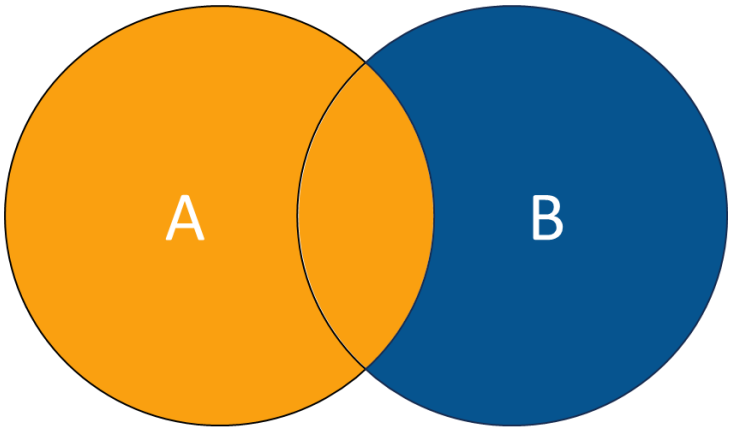
- FROM: Dữ liệu được lấy từ bảng SalesLT.Customer được gán dưới tên CST.
- INNER JOIN: Kết hợp điểm chung bảng Customer được gán tên CST với bảng CustomerAddress được gán tên CTA.
- ON: Khai báo điều kiện kết hợp bảng từ cột khoá chính CustomerID trong bảng Customer và khoá ngoại CustomerID trong bảng CustomerAddress.
- SELECT: Truy vấn các cột CST.CustomerID, CST.FirstName, CST.LastName, CTA.AddressID, CTA.AddressType.

4.1.2.2. Kết hợp trái (Left join)

Nếu bảng A LEFT JOIN với bảng B thì kết quả gồm các bản ghi có trong bảng A, với các bản ghi không có mặt trong bảng B thì các cột từ B được điền NULL. Các bản ghi chỉ có trong B mà không có trong A sẽ không được trả về.

Bảng được xác định là left trong phép JOIN là bảng được viết trước.

© 2023 Datapot. All right reserved.



LEFT JOIN

Cú pháp của LEFT JOIN:

```
SELECT
        A.Col_Name,
        B. Col_Name
FROM Table_A
LEFT JOIN Table_B
On A.Key = B.Key
```

© 2023 Datapot. All right reserved.

Ví dụ: Chọn bảng dữ liệu adventureworks, kết hợp trái giữa bảng Customer và Customer Address

```sql
USE adventureworks
SELECT
    CST.CustomerID,
    CST.FirstName,
    CST.LastName,
    CTA.AddressID,
    CTA.AddressType
FROM SalesLT.Customer AS CST
LEFT JOIN SalesLT.CustomerAddress AS CTA
ON CST.CustomerID = CTA.CustomerID
```

Results Messages

	CustomerID	FirstName	LastName	AddressID	AddressType
1	1	Orlando	Gee	NULL	NULL
2	2	Keith	Harris	NULL	NULL
3	3	Donna	Carreras	NULL	NULL
4	4	Janet	Gates	NULL	NULL

0 Ln 11, Col 1 Spaces: 4 UTF-8 CRLF SQL 857 rows MSSQL 00:00:00

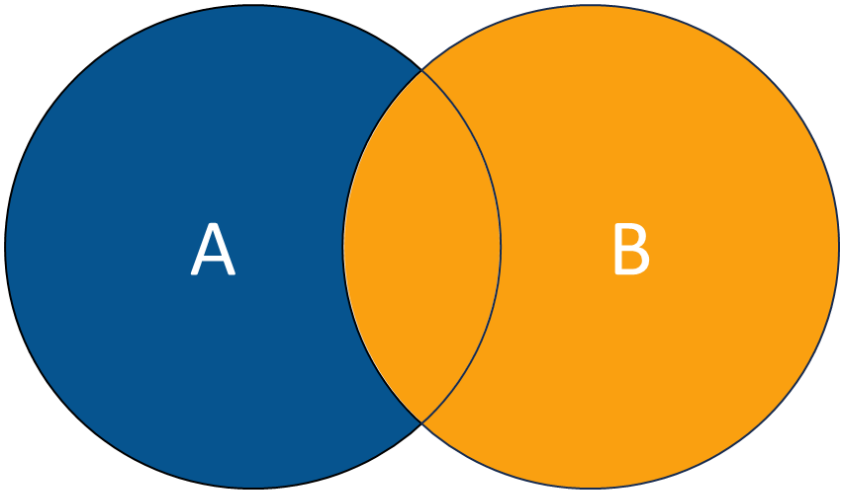
Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng SalesLT.Customer được gán dưới tên CST.
- LEFT JOIN: Kết hợp trái bảng Customer được gán tên CST với bảng CustomerAddress được gán tên CTA.
- ON: Khai báo điều kiện kết hợp bảng từ cột khoá chính CustomerID trong bảng Customer và khoá ngoại CustomerID trong bảng CustomerAddress.
- SELECT: Truy vấn các cột CST.CustomerID, CST.FirstName, CST.LastName, CTA.AddressID, CTA.AddressType.

4.1.2.3. Kết hợp phải (Right join)

Nếu bảng được kết hợp phải với bảng B thì kết quả gồm các bản ghi có trong bảng B. Với các bản ghi không có mặt trong bảng A thì các cột từ bảng A được trả về NULL. Các bản ghi chỉ có trong bảng A mà không có trong bảng B sẽ không được trả về.

© 2023 Datapot. All right reserved.



RIGHT JOIN

Cú pháp của RIGHT JOIN:

```
SELECT
    A. Col_Name,
    B. Col_Name
FROM Table_A
RIGHT JOIN Table_B
On A.Key = B.Key
```

Ví dụ: Chọn bảng dữ liệu adventureworks, kết hợp phải giữa bảng Customer và Customer Address.

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    CST.CustomerID,
    CST.FirstName,
    CST.LastName,
    CTA.AddressID,
    CTA.AddressType
FROM SalesLT.Customer AS CST
RIGHT JOIN SalesLT.CustomerAddress AS CTA
ON CST.CustomerID = CTA.CustomerID
```

Results Messages

	CustomerID	FirstName	LastName	AddressID	AddressType
1	29485	Catherine	Abel	1086	Main Office
2	29486	Kim	Abercrombie	621	Main Office
3	29489	Frances	Adams	1069	Main Office
4	29490	Margaret	Smith	887	Main Office

0 Ln 11, Col 1 Spaces: 4 UTF-8 CRLF SQL 417 rows MSSQL 00:00:00

Giải thích câu lệnh truy vấn:

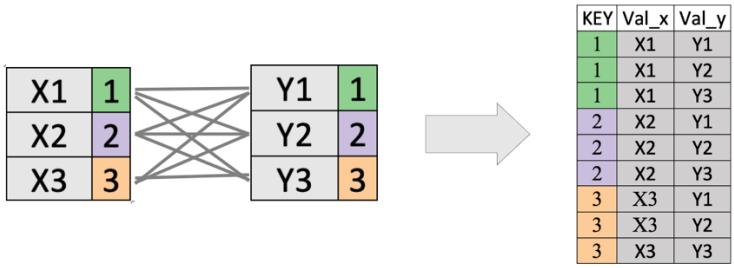
- FROM: Dữ liệu được lấy từ bảng SalesLT.Customer được gán dưới tên CST.
- RIGHT JOIN: Kết hợp phải bảng Customer được gán tên CST với bảng CustomerAddress được gán tên CTA.
- ON: Khai báo điều kiện kết hợp bảng từ cột khoá chính CustomerID trong bảng Customer và khoá ngoại CustomeID trong bảng bảng CustomerAddress.
- SELECT: Truy vấn các cột CST.CustomerID, CST.FirstName, CST.LastName, CTA.AddressID, CTA.AddressType.

4.1.2.4. Kết hợp chéo (Cross join)

Kết hợp chéo (Cross join) là kết hợp giữa các hàng của hai bảng với nhau, mỗi hàng trong bảng thứ nhất sẽ kết hợp với N hàng của bảng thứ hai. Kết quả của kết hợp chéo (Cross join) sẽ có số hàng bằng tích số của hai bảng.

Do kết quả trả ra của Cross join có thể rất lớn là tích số của hai bảng, cần cân nhắc sự cần thiết khi sử dụng kết hợp chéo (Cross join).

© 2023 Datapot. All right reserved.



Cú pháp của CROSS JOIN:

SELECT

FROM Table_A

CROSS JOIN Table_B

Ví dụ: Kết hợp bảng Customer (gồm 847 hàng) và CustomerAddress (gồm 417 hàng) bằng kết hợp chéo (Cross join).

Kết quả trả về là tích của 2 bảng là 353.199 hàng.

```sql
USE adventureworks
SELECT
    CST.CustomerID,
    CST.FirstName,
    CST.LastName,
    CTA.AddressID,
    CTA.AddressType
FROM SalesLT.Customer AS CST
CROSS JOIN SalesLT.CustomerAddress AS CTA
```

Results Messages

	CustomerID	FirstName	LastName	AddressID	AddressType
1	1	Orlando	Gee	1086	Main Office
2	2	Keith	Harris	1086	Main Office
3	3	Donna	Carreras	1086	Main Office
4	4	Janet	Gates	1086	Main Office

0 SQL 353,199 rows MSSQL 00:00:02 sql-st.datapot.edu.vn : adventureworks

Giải thích câu lệnh truy vấn:

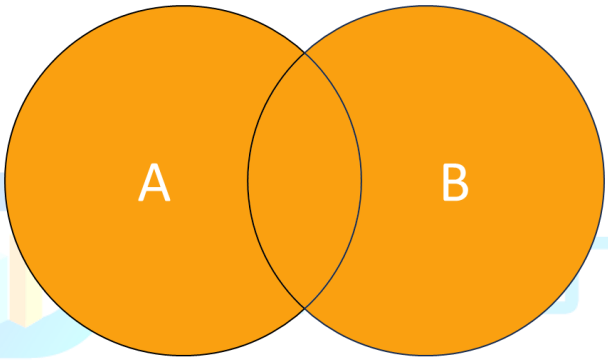
© 2023 Datapot. All right reserved.

- FROM: Dữ liệu được lấy từ bảng SalesLT.Customer được gán dưới tên CST.
- CROSS JOIN: Kết hợp chéo bảng Customer được gán tên CST với bảng CustomerAddress được gán tên CTA.
- SELECT: Truy vấn các cột CST.CustomerID, CST.FirstName, CST.LastName, CTA.AddressID, CTA.AddressType.

4.1.2.5. Kết hợp tất cả (Outer join/Full Outer Join)

Kết hợp chung (Outer join/Full Outer Join/FULL Join) là kết hợp tất cả các hàng với nhau.

Nếu không có sự trùng khớp giữa hai bảng với nhau, giá trị không xác định (NULL) sẽ được trả về cho các cột của bảng chứa các giá trị thiếu.



FULL OUTER JOIN

Cú pháp của FULL JOIN:

```
SELECT
        A.Col_Name,
        B. Col_Name
    FROM Table_A
    FULL JOIN Table_B
    On A.Key = B.Key
```

© 2023 Datapot. All right reserved.

Ví dụ: Kết hợp chung giữa bảng Customer và bảng (Table) CustomerAddress. Truy vấn các cột CustomerID, FirstNam, LastName, AddressID, AddressType.

```sql
USE adventureworks
SELECT
    CST.CustomerID,
    CST.FirstName,
    CST.LastName,
    CTA.AddressID,
    CTA.AddressType
FROM SalesLT.Customer AS CST
FULL JOIN SalesLT.CustomerAddress AS CTA
ON CST.CustomerID = CTA.CustomerID
```

Results Messages

	CustomerID	FirstName	LastName	AddressID	AddressType
1	1	Orlando	Gee	NULL	NULL
2	2	Keith	Harris	NULL	NULL
3	3	Donna	Carreras	NULL	NULL
4	4	Janet	Gates	NULL	NULL

0 CRLF SQL 857 rows MSSQL 00:00:00 sql-st.datapot.edu.vn : adventureworks

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng SalesLT.Customer được gán dưới tên CST.
- FULL JOIN: Kết hợp tất cả bản ghi của bảng Customer được gán tên CST với bảng CustomerAddress được gán tên CTA.
- ON: Khai báo điều kiện kết hợp bảng từ cột khoá chính CustomerID trong bảng Customer và khoá ngoại CustomerID trong bảng bảng CustomerAddress.
- SELECT: Truy vấn các cột CST.CustomerID, CST.FirstName, CST.LastName, CTA.AddressID, CTA.AddressType.

4.1.2.6. Kết hợp với nhiều hơn 2 bảng

JOIN nhiều hơn 2 bảng được phát triển từ phép JOIN thông thường. Thay vì chỉ JOIN 2 bảng thì chúng ta JOIN nhiều hơn 2 bảng.

Cú pháp của JOIN nhiều hơn 2 bảng:

SELECT

© 2023 Datapot. All right reserved.

Co_1,

Col_2,...

FROM Table 1 AS t1

LEFT/RIGHT/FULL/INNER JOIN Table 2 AS t2

ON t1.Key1 = t2.Key1

LEFT/RIGHT/FULL/INNER JOIN Table 3 AS t3

ON t1.Key2 = t3.Key2

Ví dụ: Từ 3 bảng trong bộ dữ liệu FactInternetSales, DimProduct, DimCustomer thuộc bộ dữ liệu AdventureWorksDW2019, truy vấn các cột ProductKey, FirstName, Color, SalesAmount .Với điều kiện, các đơn hàng có bán sản phẩm có Color = 'Red' và Customer FirstName bắt đầu bằng chữ A.

```sql
SELECT
    FIS.ProductKey,
    DC.FirstName,
    DP.Color,
    FIS.SalesAmount
FROM dbo.FactInternetSales AS FIS
LEFT JOIN dbo.DimProduct AS DP
ON DP.ProductKey = FIS.ProductKey
LEFT JOIN dbo.DimCustomer AS DC
ON FIS.CustomerKey = DC.CustomerKey
WHERE
    Color = 'Red' AND
    FirstName LIKE 'A%'
```

Results Messages

	ProductKey	FirstName	Color	SalesAmount
1	310	Albert	Red	3578.27
2	311	Abigail	Red	3578.27
3	312	Albert	Red	3578.27
4	311	Anne	Red	3578.27
5	313	Alyssa	Red	3578.27

© 2023 Datapot. All right reserved.

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng dbo.FactInternetSales được gán dưới tên FIS.
- WHERE: Lọc bản ghi thoả mãn các đơn hàng có bán sản phẩm có Color = 'Red' và Customer FirstName bắt đầu bằng chữ A.
- LEFT JOIN: Kết hợp trái bảng DimProduct được gán tên DP với bảng FactInternetSales được gán tên FIS.
- ON: Khai báo điều kiện kết hợp bảng từ cột khoá chính ProductKey trong bảng DimProduct và khoá ngoại ProductKey trong bảng FactInternetSales.
- LEFT JOIN: Sau khi kết hợp bảng FactInternetSales và DimProduct, tiếp tục kết hợp trái bảng DimCustomer được gán tên DC.
- ON: Khai báo điều kiện kết hợp bảng từ cột khoá chính CustomerKey trong bảng DimCustomer và khoá ngoại CustomerKey trong bảng FactInternetSales.
- SELECT: Truy vấn các cột FIS.ProductKey, DC.FirstName, DP.Color, FIS.SalesAmount.

4.1.2.7. Kết hợp với chính nó (Self Join)

Self Join về bản chất vẫn là phép Join thông thường, tuy nhiên thay vì kết hợp với một bảng khác thì sẽ sử dụng mối quan hệ về dữ liệu có sẵn ở trong bảng để Join với chính nó.

Self Join thường được sử dụng với dữ liệu có mối quan hệ phân cấp và phân tầng. Ví dụ như dữ liệu mô hình tổ chức (Khối - Phòng ban), dữ liệu nhân sự (Cấp quản lý - Cấp trực thuộc),...

Cú pháp của Self Join:

```
SELECT
        A.Col_Name,
        B.Col_Name, ..
FROM Table 1 AS A
LEFT/RIGHT/INNER/FULL JOIN Table 1 AS B
ON A.Key = B.Key
```

© 2023 Datapot. All right reserved.

Ví dụ: Ta có bảng dữ liệu dbo.DimEmployee thuộc bộ dữ liệu AdventureWorksDW2020. Thực hiện truy vấn tên người quản lý tương ứng với từng nhân viên (Sử dụng Self Join).

```sql
USE AdventureWorksDW2019
SELECT
    DE.EmployeeKey,
    DE.LastName AS Employee,
    PE.LastName AS Parent_Employee,
    DE.Title AS Title_Employee
FROM dbo.DimEmployee AS DE
LEFT JOIN dbo.DimEmployee AS PE
ON DE.ParentEmployeeKey = PE.EmployeeKey
ORDER BY EmployeeKey
```

Results Messages

	EmployeeKey	Employee	Parent_Employee	Title_Employee
1	1	Gilbert	Brown	Production Technician - WC60
2	2	Brown	Bradley	Marketing Assistant
3	3	Tamburello	Duffy	Engineering Manager
4	4	Walters	Tamburello	Senior Tool Designer
5	5	Walters	Tamburello	Senior Tool Designer
6	6	D'Hers	Cracium	Tool Designer

0 Spaces: 4 UTF-8 CRLF SQL 296 rows Choose SQL Language 00:00:00 sql-st.data

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng DimEmployee được gán tên DE.
- LEFT JOIN: Kết hợp trái bảng DimEmployee với bảng PE.
- ON: Khai báo điều kiện kết hợp từ cột ParentEmployeeKey trong bảng DimEmployee và cột EmployeeKey bảng PE.
- SELECT: Truy vấn các cột DE.EmployeeKey, DE.LastName, PE.LastName, DE.Title.

4.2. Các lưu ý khi viết JOIN

Lưu ý 1: Về Alias tên bảng

Khi thực hiện JOIN, ngầm định sẽ phải cần Alias tên bảng đem kết hợp. Khi Alias, hãy sử dụng các tên viết tắt và mang tính gợi nhớ đến bảng gốc.

© 2023 Datapot. All right reserved.

Nên	Không nên

`USE adventureworks`
`SELECT`
`  CST.CustomerID,`
`  CST.FirstName,`
`  CST.LastName,`
`  CTA.AddressID,`
`  CTA.AddressType`
`FROM SalesLT.Customer AS CST`
`LEFT JOIN SalesLT.CustomerAddress AS CTA`
`ON CST.CustomerID = CTA.CustomerID`
	`USE adventureworks`
	`SELECT`
	`  A.CustomerID,`
	`  A.FirstName,`
	`  A.LastName,`
	`  B.AddressID,`
	`  B.AddressType`
	`FROM SalesLT.Customer AS A`
	`LEFT JOIN SalesLT.CustomerAddress AS B`
	`ON A.CustomerID = B.CustomerID`

Lưu ý 2: Lỗi Ambigous column name

```
USE adventureworks
SELECT
    CustomerID
FROM SalesLT.Customer AS CST
LEFT JOIN SalesLT.CustomerAddress AS CTA
ON CST.CustomerID = CTA.CustomerID
```

```
es

6:17 PM     Started executing query at Line 15
            Msg 209, Level 16, State 1, Line 3
            Ambiguous column name 'CustomerID'.
            Total execution time: 00:00:00.036
```

Đây là một lỗi thường xuyên gặp khi mới thực hành viết phép JOIN. Nguyên nhân lỗi này từ việc gọi tên cột CustomerID nhưng tên cột này xuất hiện ở cả 2 bảng Customer và CustomerAddress. Vậy nên cần khai báo rõ ràng cột CustomerID đến từ bảng nào bằng cách khai báo tên bảng Alias đằng trước tên cột (Ví dụ: CST.CustomerID).

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    CST.CustomerID,
    FirstName
FROM SalesLT.Customer AS CST
LEFT JOIN SalesLT.CustomerAddress AS CTA
ON CST.CustomerID = CTA.CustomerID
```

Results Messages

	CustomerID	FirstName
1	1	Orlando
2	2	Keith
3	3	Donna
4	4	Janet
5	5	Lucy

Sau khi viết lại thành CST.CustomerID thì không còn lỗi như trước. Ngoài ra, đối với cột FirstName tuy không có tên Alias của bảng đằng trước nhưng không bị lỗi Ambigous là vì cột FirstName chỉ tồn tại duy nhất ở bảng Customer (CST).

Mặc dù vậy, vẫn khuyến khích sử dụng tên Alias bảng kèm đằng trước các cột để câu lệnh rõ ràng và mạch lạc hơn.

4.3. Kết hợp bảng sử dụng UNION

4.3.1. UNION và UNION ALL

4.3.1.1. UNION

UNION kết hợp các cột từ hai hay nhiều mệnh đề SELECT theo chiều dọc và không bao gồm các dòng trùng lặp.

© 2023 Datapot. All right reserved.

Đặc điểm:

- Khi UNION các câu lệnh SELECT cần trả về số cột dữ liệu bằng nhau.
- Các cột tương ứng cần có cùng kiểu dữ liệu.
- UNION sẽ gộp cả NULL.

Cú pháp của UNION:

```
SELECT
    Col_1,
    Col_2,...
FROM Table_1
UNION
SELECT
    Col_1,
    Col_2,...
FROM Table_2
```

Bộ dữ liệu bao gồm 4012 hàng.

© 2023 Datapot. All right reserved.

```sql
--Bộ dữ liệu gồm 4.021 hàng
USE AdventureWorksFull
SELECT
    OrderDate,
    TotalDue,
    'Purchase' AS Source
FROM Purchasing.PurchaseOrderHeader
```

Results Messages

	OrderDate	TotalDue	Source
1	2011-04-16 00:00:00.000	222.1492	Purchase
2	2011-04-16 00:00:00.000	300.6721	Purchase
3	2011-04-16 00:00:00.000	9776.2665	Purchase
4	2011-04-16 00:00:00.000	189.0395	Purchase
5	2011-04-30 00:00:00.000	22539.0165	Purchase

0 Spaces: 4 UTF-8 CRLF SQL 4,012 rows MSSQL

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng Purchasing.PurchaseOrderHeader.
- SELECT: Truy vấn các cột OrderDate, TotalDue, Source.

Bộ dữ liệu bao gồm 31.465 hàng.

© 2023 Datapot. All right reserved.

```sql
--Bộ dữ liệu gồm 31.465 hàng
USE AdventureWorksFull
SELECT
    OrderDate,
    TotalDue,
    'Sales' AS Source
FROM Sales.SalesOrderHeader
```

Results | Messages

	OrderDate	TotalDue	Source
1	2011-05-31 00:00:00.000	23153.2339	Sales
2	2011-05-31 00:00:00.000	1457.3288	Sales
3	2011-05-31 00:00:00.000	36865.8012	Sales
4	2011-05-31 00:00:00.000	32474.9324	Sales
5	2011-05-31 00:00:00.000	472.3108	Sales

0 Spaces: 4 UTF-8 CRLF SQL 31,465 rows MSSQL

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng Sales.SalesOrderHeader.
- SELECT: Truy vấn các cột OrderDate, TotalDue, Source.

Ví dụ: Kết hợp các đơn hàng từ Purchasing sources (Gồm 4.012 hàng) và Sales sources (Gồm 31.465 hàng) và loại bỏ giá trị trùng lặp.

Tổng số hàng trả về 29.224.

© 2023 Datapot. All right reserved.

```sql
-- Bộ dữ liệu kết hợp khi đã loại bỏ các giá trị trùng lặp
USE AdventureWorksFull
SELECT
    OrderDate,
    TotalDue,
    'Purchase' as Source
FROM Purchasing.PurchaseOrderHeader
UNION
SELECT
    OrderDate,
    TotalDue,
    'Sales' as Source
FROM Sales.SalesOrderHeader
```

Results grid

Results | Messages

	OrderDate	TotalDue	Source
1	2011-04-16 00:00:00.000	189,0395	Purchase
2	2011-04-16 00:00:00.000	222,1492	Purchase
3	2011-04-16 00:00:00.000	300,6721	Purchase
4	2011-04-16 00:00:00.000	9776,2665	Purchase
5	2011-04-30 00:00:00.000	766,1827	Purchase
6	2011-04-30 00:00:00.000	16164,0229	Purchase
7	2011-04-30 00:00:00.000	22539,0165	Purchase

Ln 13, Col 28 Spaces: 4 UTF-8 LF SQL 29.224 rows

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng PurchaseOrderHeader, SalesOrderHeader.
- UNION: Gộp các cột không bao gồm các dòng trùng lặp giữa 2 bảng với nhau.
- SELECT: Truy vấn các cột bao gồm OrderDate, TotalDue, Source.

4.3.1.2. UNION ALL

Tương tự như UNION, UNION ALL kết hợp cột từ hai hay nhiều mệnh đề SELECT theo chiều dọc. Tuy nhiên, không loại bỏ các dòng trùng lặp nếu có.

Đặc điểm:

- Khi UNION ALL các câu lệnh SELECT cần trả về số cột dữ liệu bằng nhau.
- Các cột tương ứng cần có cùng kiểu dữ liệu.
- UNION ALL sẽ gộp cả NULL.

© 2023 Datapot. All right reserved.

Cú pháp của UNION ALL:

```
SELECT
    Col_1,
    Col_2,...
FROM Table_1
UNION ALL
SELECT
    Col_1,
    Col_2,...
FROM Table_2
```

Ví dụ: Kết hợp các đơn hàng từ Purchasing sources và Sales sources, sử dụng gộp tất cả (UNION ALL).

Tổng số hàng trả về 35.477.

© 2023 Datapot. All right reserved.

```sql
USE AdventureWorksFull
SELECT
    OrderDate,
    TotalDue,
    'Purchase' as Source
FROM Purchasing.PurchaseOrderHeader
UNION ALL
SELECT
    OrderDate,
    TotalDue,
    'Sales' as Source
FROM Sales.SalesOrderHeader
```

Results Messages

	OrderDate	TotalDue	Source
1	2011-04-16 00:00:00.000	222,1492	Purchase
2	2011-04-16 00:00:00.000	300,6721	Purchase
3	2011-04-16 00:00:00.000	9776,2665	Purchase
4	2011-04-16 00:00:00.000	189,0395	Purchase
5	2011-04-30 00:00:00.000	22539,0165	Purchase
6	2011-04-30 00:00:00.000	16164,0229	Purchase
7	2011-04-30 00:00:00.000	64847,5328	Purchase

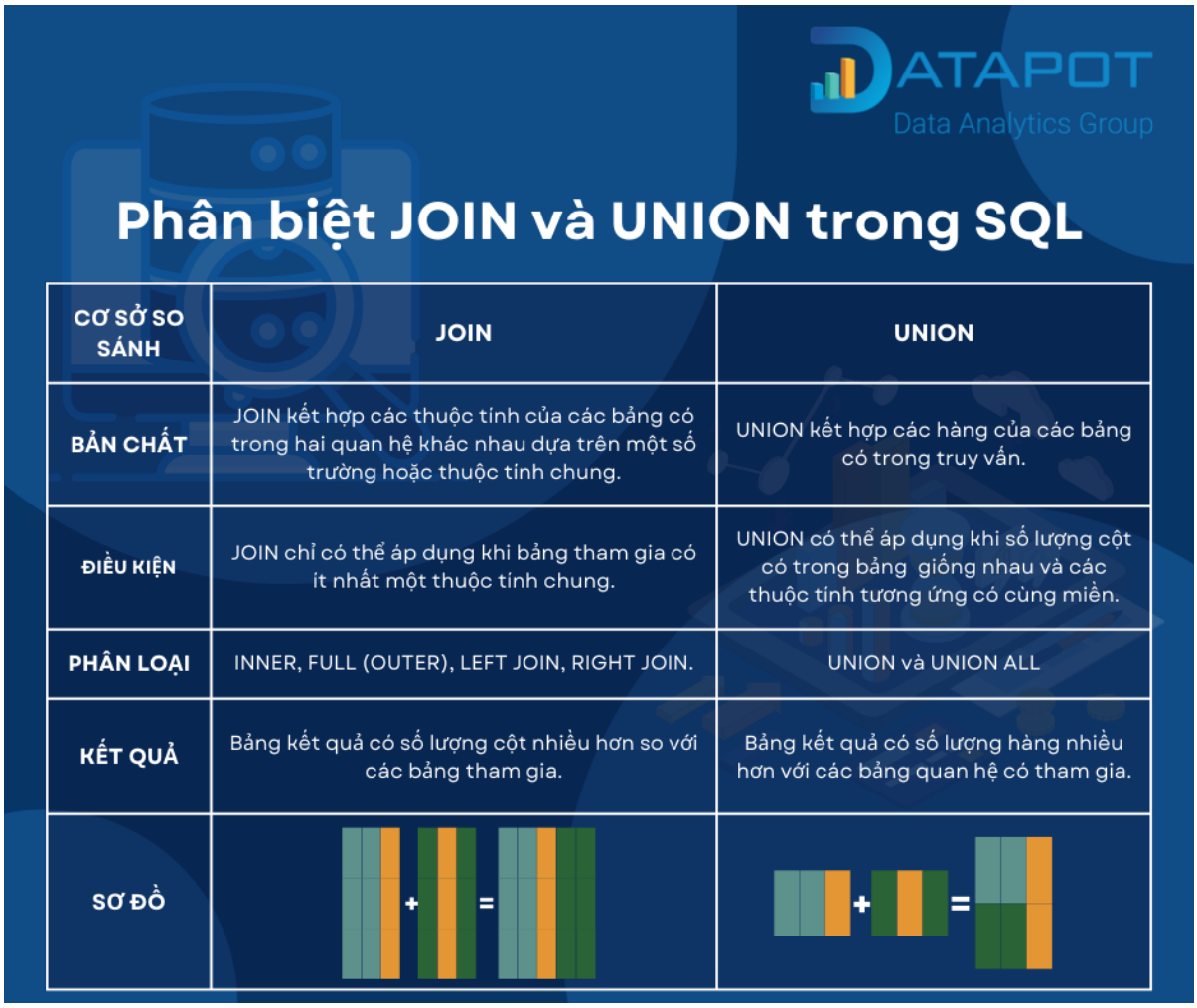
Ln 12, Col 28 Spaces: 4 UTF-8 LF SQL 35.477 rows

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng PurchaseOrderHeader, SalesOrderHeader.
- UNION ALL: Gộp các cột giữ nguyên các dòng trùng lặp giữa 2 bảng với nhau.
- SELECT: Truy vấn các cột bao gồm OrderDate, TotalDue, Source.

Các bạn có thể đọc thêm để phân biệt rõ ràng hơn giữa JOIN và UNION.

© 2023 Datapot. All right reserved.



4.3.2. Một số lý thuyết tập hợp (Set theory) khác

Bên cạnh UNION và UNION ALL được giới thiệu chính trong khoá DP080, bạn có thể tham khảo thêm các lý thuyết tập hợp khác là INTERSECT và EXCEPT.

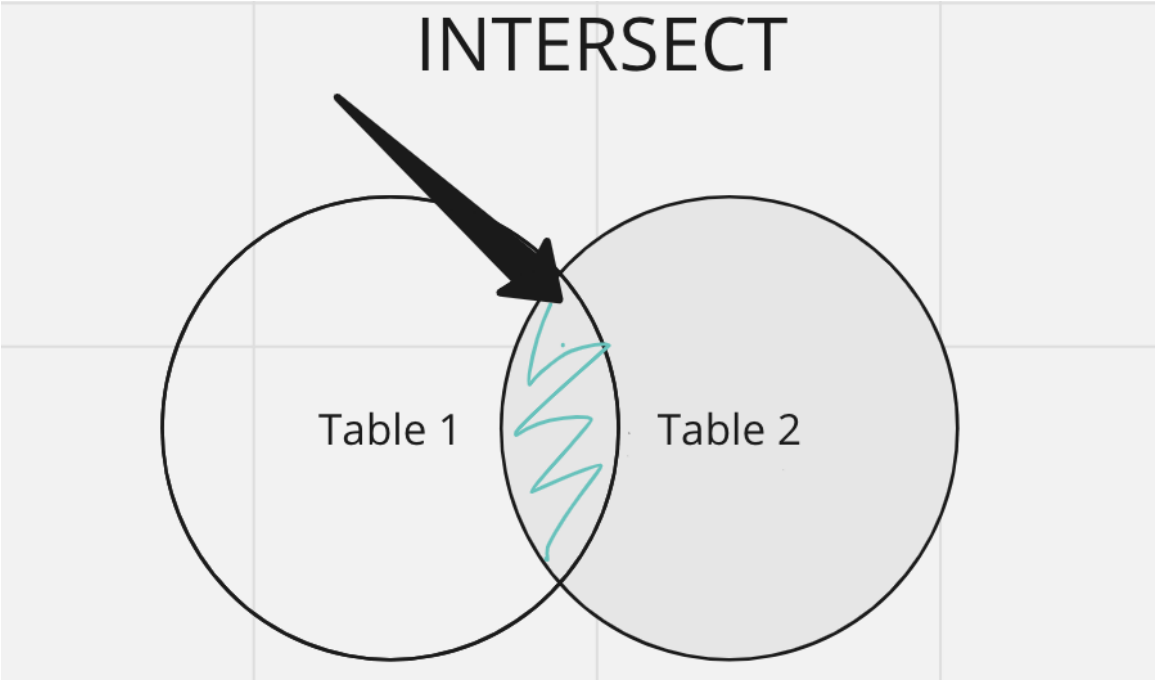
4.3.2.1. Intersect

INTERSECT truy vấn các bản ghi đồng thời xuất hiện trong 2 câu truy vấn đã chọn.

Đặc điểm

- INTERSECT có trả lại giá trị không tồn tại (NULL).
- INTERSECT không trả lại các giá trị trùng lặp.
- Tổng số cột được chọn của 2 bảng phải bằng nhau.

© 2023 Datapot. All right reserved.



Lưu ý: Tên cột và kiểu dữ liệu phải giống nhau trong 2 câu lệnh SELECT.

Cú pháp của INTERSECT:

```
SELECT
    Col_1,
    Col_2
    FROM Table_A
INTERSECT
SELECT
    Col_1,
    Col_2
    FROM Table_B
```

Ví dụ: Bạn cần truy vấn những SalesOrderID đồng thời xuất hiện trong 2 bảng là bảng (Table) SalesOrderDetail và bảng (Table) SalesOrderID.

Ta dùng INTERSECT để truy vấn dữ liệu đề bài yêu cầu:

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    SalesOrderID
FROM SalesLT.SalesOrderDetail
INTERSECT
SELECT
    SalesOrderID
FROM SalesLT.SalesOrderHeader
```

Results Messages

	SalesOrderID
1	71774
2	71776
3	71780
4	71782
5	71783

QL 32 rows MSSQL 00:00:00 sql-st.datapo

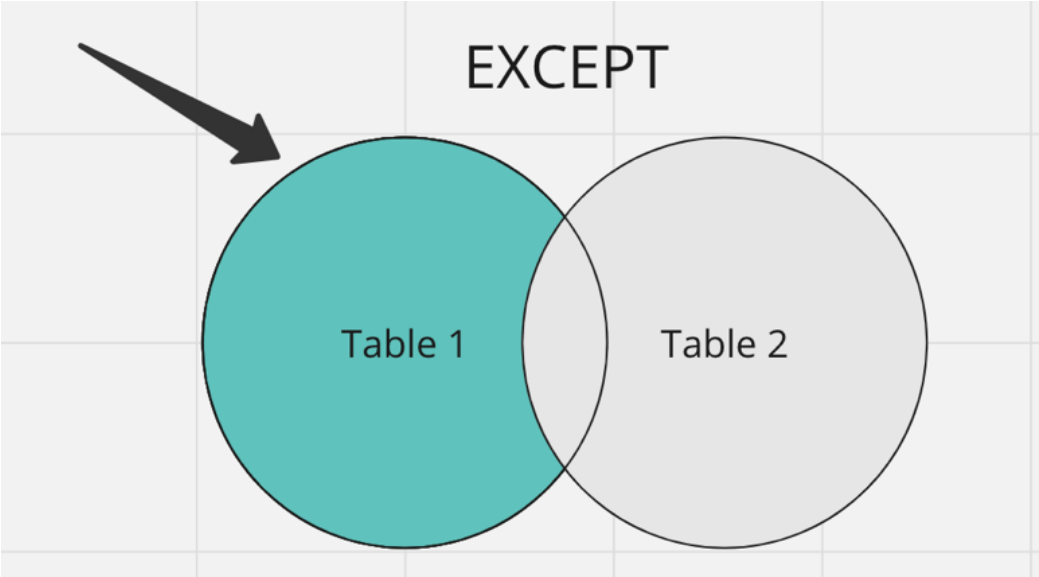
Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng SalesLT.SalesOrderDetail, SalesLT.SalesOrderHeader.
- INTERSECT: Truy vấn các bản ghi đồng thời xuất hiện tại 2 bảng SalesLT.SalesOrderDetail và SalesLT.SalesOrderHeader.
- SELECT: Truy vấn cột SalesOrderID.

4.3.2.2. Exception

Exception dùng để truy vấn ra các giá trị trong kết quả câu truy vấn bảng 1 và loại trừ kết quả trùng khớp với câu truy vấn bảng 2.

© 2023 Datapot. All right reserved.



Cú pháp của EXCEPT:

```
SELECT
        Col_1,
        Col_2
    FROM Table_A
    EXCEPT
    SELECT
        Col_1,
        Col_2
    FROM Table_B
```

Để hiểu rõ hơn ta cùng xem ví dụ sau:

Từ bảng (Table) Product gồm 295 hàng và bảng ProductAndDescription gồm 1.764 hàng bạn phải truy vấn ProductID chung của 2 bảng, ngoại trừ ProductID xuất hiện trong bảng vProductAnDescription.

Ta dùng từ khóa (Key Word) EXCEPT để thực hiện:

Do giá trị “907” là giá trị duy nhất có trong bảng ProductID nên sau khi thực hiện truy vấn ta có kết quả nhận được là 1 hàng với giá trị “907”.

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    ProductID
FROM SalesLT.Product
EXCEPT
SELECT
    ProductID
FROM SalesLT.vProductAndDescription
```

Results Messages

	ProductID
1	907

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được lấy từ bảng SalesLT.Product, SalesLT.vProductAndDescription.
- EXCEPT: Truy vấn những bản ghi có trong bảng SalesLT.Product và loại trừ các bản ghi có xuất hiện trong bản ghi SalesLT.vProductAndDescription.
- SELECT: Truy vấn cột ProductID.

Tài liệu tham khảo:

- Hai kiểu gộp bảng bằng JOIN và UNION trong SQL
- Đọc thêm về các loại JOIN
- Các ứng dụng khác của Self Join
- Tìm hiểu về No-Equi Join
- Tìm hiểu về toán tử tập hợp SET OPERATOR trong SQL

© 2023 Datapot. All right reserved.