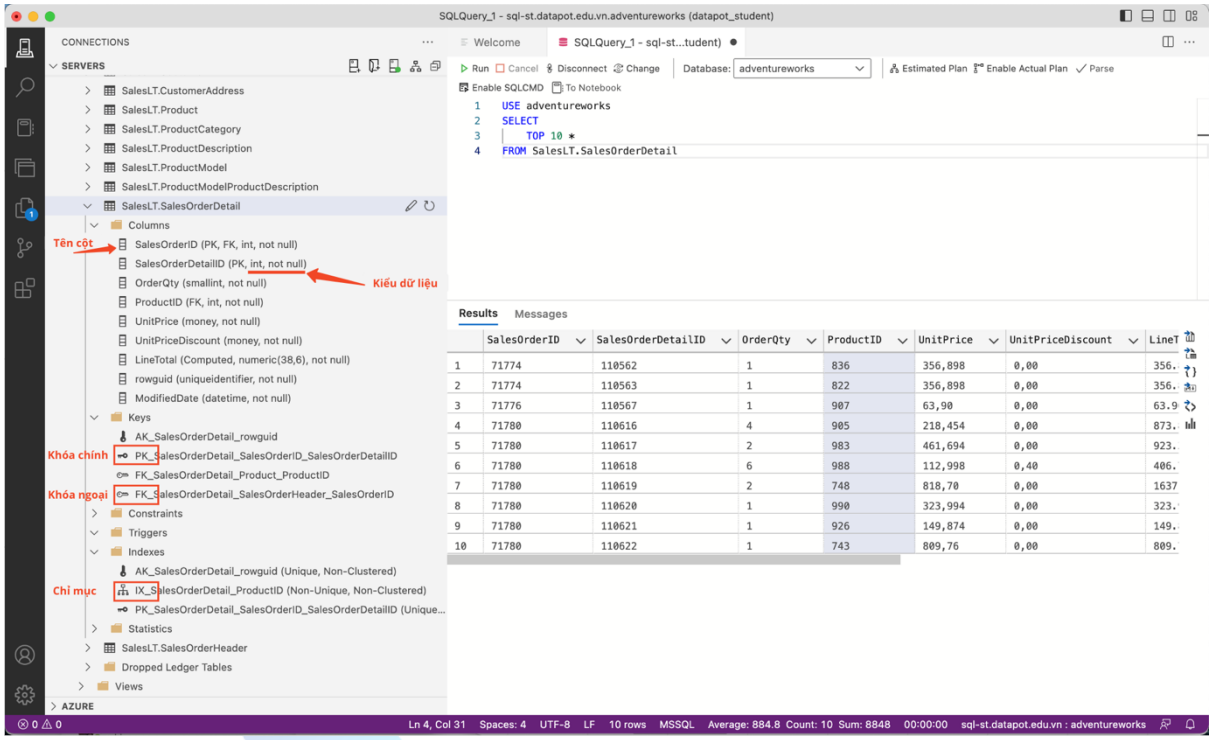
Chapter 2: Kiểu dữ liệu của cột và các hàm làm việc với kiểu dữ liệu 2
2.1. Cách xem thuộc tính của cột (Column properties) 2
2.1.1. Tên cột 2
2.1.2. Kiểu dữ liệu 2
2.1.3. Chỉ mục (Index - IX) 2
2.1.4. Một số các ràng buộc (Constraints) được ứng dụng vào các cột hoặc các bảng 2
2.2. Kiểu dữ liệu trong SQL Server (Data type) 3
2.2.1. Dữ liệu dạng số. 4
2.2.1.1. Dữ liệu dạng số xấp xỉ 4
2.2.1.2. Dữ liệu dạng số chính xác 4
2.2.2. Dữ liệu dạng ký tự 5
2.2.2.1. Dữ liệu dạng chuỗi ký tự 5
2.2.2.2. Dữ liệu dạng chuỗi nhị phân 5
2.2.2.3. Dữ liệu dạng ký tự Unicode 5
2.2.3. Dữ liệu dạng ngày & giờ 6
2.2.4. Trạng thái NULL của dữ liệu 7
2.3. Kiểm tra kiểu dữ liệu của cột trong bảng 8
2.4. Các hàm làm việc tương ứng kiểu dữ liệu 10
2.4.1. Các hàm chuyển đổi kiểu dữ liệu 10
2.4.1.1. Hàm chuyển đổi kiểu dữ liệu 11
2.4.1.2. Xử lý dữ liệu dạng chuỗi 19
2.4.1.3. Xử lý và thao tác với dữ liệu dạng ngày (Date manipulations) 28
2.4.1.4. Xử lý các dữ liệu không tồn tại hoặc dữ liệu không có sẵn (NULL) 29
2.5. Nhóm hàm định dạng dữ liệu (Formatting data) 33
2.5.1. Định dạng ngày (Date formats) 33
2.5.2. Làm tròn số (Rounding) 34

© 2023 Datapot. All right reserved.

Chapter 2: Kiểu dữ liệu của cột và các hàm làm việc với kiểu dữ liệu

2.1. Cách xem thuộc tính của cột (Column properties)



2.1.1. Tên cột

Tên của cột đã chọn.

2.1.2. Kiểu dữ liệu

Hiện thị kiểu dữ liệu của cột đã chọn.

2.1.3. Chỉ mục (Index - IX)

Chỉ mục dùng để truy vấn nhanh hơn, giúp giảm số lượng trang dữ liệu cần được đọc để truy vấn dữ liệu trong một câu lệnh SQL.

2.1.4. Một số các ràng buộc (Constraints) được ứng dụng vào các cột hoặc các bảng

- Giá trị không tồn tại hoặc không xác định (NOT NULL)

  Là các cột không có giá trị không tồn tại (NULL VALUE).

- Giá trị duy nhất (UNIQUE)

  Tất cả các giá trị trong cột đều khác nhau.

© 2023 Datapot. All right reserved.

- Khóa chính (PRIMARY KEY)

  Khóa chính là sự kết hợp của giá trị không tồn tại (NULL VALUE) và giá trị duy nhất (Unique). Tồn tại duy nhất một khóa chính trong một bảng dữ liệu.

- Khóa ngoại (FOREIGN KEY)

  Khóa ngoại là là một hoặc nhiều cột trong bảng mà chứa giá trị tham chiếu liên kết với với khóa chính của bảng khác.

  Khóa chính và khóa ngoại sẽ được nêu trong chapter 3 của textbook này.

- Kiểm tra (CHECK)

  Đảm bảo giá trị trong cột thỏa mãn các điều kiện nhất định.

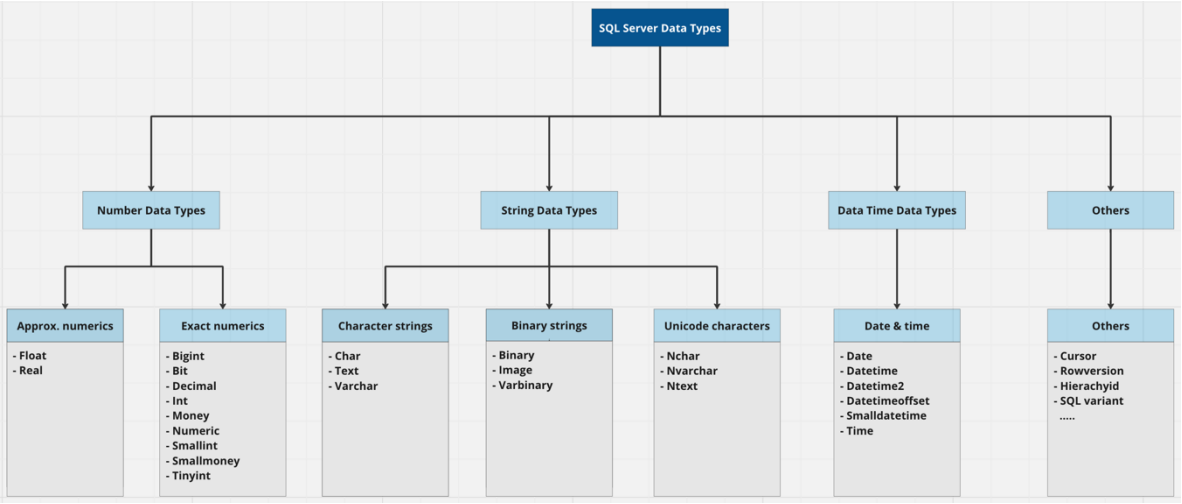
- DEFAULT

  Đặt một giá trị mặc định cho cột nếu cột không có giá trị nào được chỉ định.

- Tạo chỉ mục (CREATE INDEX)

  Giúp truy vấn dữ liệu nhanh chóng hơn.

2.2. Kiểu dữ liệu trong SQL Server (Data type)



Một số kiểu dữ liệu thường gặp:

© 2023 Datapot. All right reserved.

2.2.1. Dữ liệu dạng số.

2.2.1.1. Dữ liệu dạng số xấp xỉ

- FLOAT(N): Định dạng dữ liệu được sử dụng để lưu trữ các số thực.

  FLOAT(10): 10 là số byte được sử dụng để lưu trữ phần thập phân.

- REAL: Được sử dụng để lưu trữ số thực, ít chính xác hơn FLOAT().

2.2.1.2. Dữ liệu dạng số chính xác

- DECIMAL(p,s): Được sử dụng để lưu trữ số thập phân có độ chính xác cố định.

  p là tổng chữ số.

  s là số chữ số sau số thập phân.

- NUMERIC(p,s): Kiểu dữ liệu số nguyên

  p là tổng chữ số.

  s là số chữ số thập phân.

To Notebook

```sql
SELECT
    CAST(243.87989 AS DECIMAL(6,3)),
    CAST(243.87989 AS NUMERIC(6,3))
```

Results | Messages

	(No column name)	(No column name)
1	243.880	243.880

Giải thích câu lệnh truy vấn:

- SELECT: Hiển thị kết quả truy vấn.

CAST(243.87989 AS DECIMAL(6, 3)): Chuyển 243.87989 về DECIMAL có tổng 6 chữ số và 3 số thập phân.

CAST(243.87989 AS NUMERIC(6, 3)): Chuyển 243.87989 về NUMERIC có tổng 6 chữ số và 3 số thập phân.

Hàm CAST sẽ được nhắc tới tại mục 2.4.1.1. của Textbook này.

© 2023 Datapot. All right reserved.

- INT, BIGINT, SMALLINT, TINYINT đều lưu trữ kiểu dữ liệu số nguyên, tuy nhiên sẽ khác nhau về dung lượng và khoảng số nguyên.

Kiểu dữ liệu	Khoảng số nguyên	Dung lượng lưu trữ
bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	8 Bytes
int	-2,147,483,648 to 2,147,483,647	4 Bytes
smallint	-32,768 to 32,767	2 Bytes
tinyint	-32,768 to 32,767	1 Byte

2.2.2. Dữ liệu dạng ký tự

2.2.2.1. Dữ liệu dạng chuỗi ký tự

- CHAR[(n | max)]: Lưu trữ chuỗi ký tự có kích thước cố định, tương ứng n.

CHAR(10) Hệ thống cơ sở dữ liệu có thể lưu trữ 10 ký tự sử dụng mã hóa single byte, nhưng ít hơn 10 ký tự khi sử dụng mã hóa multiple byte.

CHAR(MAX) lưu trữ chuỗi ký tự không giới hạn.

- VARCHAR [(n | max)]: Dữ liệu chuỗi có kích thước linh hoạt, kích thước tương ứng n hoặc max.

VARCHAR(255) sẽ lưu trữ chuỗi ký tự có tối đa 255 ký tự, và chỉ sử dụng bộ nhớ tương ứng với độ dài của chuỗi.

VARCHAR(MAX) lưu trữ chuỗi ký tự không giới hạn.

- TEXT: Lưu trữ các chuỗi ký tự lớn, định dạng TEXT không có tham biến, nhưng sẽ lưu trữ tối đa trong phạm vi khoảng 2GB, trong khi đó VARCHAR có thể lưu trữ với phạm vi ngắn hoản khoảng 8000 byte.

2.2.2.2. Dữ liệu dạng chuỗi nhị phân

- BINARY: Dùng để lưu trữ các dãy byte có độ dài cố định.

2.2.2.3. Dữ liệu dạng ký tự Unicode

N được viết ở đầu mỗi định dạng cho phép lưu trữ ký tự Unicode.

© 2023 Datapot. All right reserved.

- NCHAR [(n | max)]: Lưu trữ chuỗi có độ dài cố định, có kích thước tương ứng n hoặc max và cho phép lưu trữ dưới dạng Unicode.

NCHAR(10) Hệ thống cơ sở dữ liệu lưu trữ 2 lần 10 byte.

NCHAR(MAX) Hệ thống lưu trữ chuỗi ký tự Unicode không giới hạn.

- NVARCHAR [(n | max)]: Lưu trữ chuỗi ký tự có độ dài linh hoạt, có kích thước tương ứng n hoặc max, cho phép Unicode.

NVARCHAR(50): Lưu trữ chuỗi ký tự Unicode có độ dài tối đa là 50 ký tự.

NVARCHAR(MAX): Lưu trữ chuỗi ký tự Unicode có độ dài không giới hạn.

- NTEXT: Chuỗi ký tự có độ dài linh hoạt, tối đa lưu trữ khoảng 1073741823 byte, cho phép Unicode.

2.2.3. Dữ liệu dạng ngày & giờ

Kiểu dữ liệu	Format	Khoảng lưu trữ	Độ chính xác
Time	hh:mm:ss[.nnnnnnn]	00:00:00.0000000 đến 23:59:59.9999999	100 nano giây
Date	YYYY-MM-DD	0001-01-01 đến 9999-12-31	1 ngày
Datetimeoffset	YYYY-MM-DD hh:mm:ss[.nnnnnnn] [+|-]hh:mm	0001-01-01 00:00:00.0000000 đến 9999-12-31 23:59:59.9999999 (UTC)	100 nano giây
Smalldatetime	YYYY-MM-DD hh:mm:ss	1900-01-01 đến 2079-06-06	1 phút
Datetime	YYYY-MM-DD hh:mm:ss[.nnn]	1753-01-01 đến 9999-12-31	100 nano giây
Datetime2	YYYY-MM-DD hh:mm:ss[.nnnnnnn]	0001-01-01 00:00:00.0000000	100 nano giây

© 2023 Datapot. All right reserved.

		đến 9999-12-31 23:59:59.9999999	

2.2.4. Trạng thái NULL của dữ liệu

NULL là dữ liệu không xác định hoặc dữ liệu không có sẵn, NULL không được coi là giá trị.

Bất cứ một phép toán nào thực hiện với NULL đều sẽ trả ra NULL. Khi lấy NULL để so sánh với các toán tử như “>”, “<”, “=”, kết quả sẽ trả về kết qua false.

```sql
SELECT
    StandardCost,
    StandardCost * 0.2 AS StandardCostDiscount,
    ProductAlternateKey,
    Class,
    ProductAlternateKey + Class AS ProductName
FROM dbo.DimProduct
```

Results Messages

	StandardCost	StandardCostDiscount	ProductAlternateKey	Class	ProductName
205	NULL	NULL	SS-2985	NULL	NULL
206	NULL	NULL	ST-9828	NULL	NULL
207	NULL	NULL	TO-2301	NULL	NULL
208	NULL	NULL	TP-0923	NULL	NULL
209	NULL	NULL	RC-0291	NULL	NULL
210	NULL	NULL	FR-R92B-58	H	FR-R92B-58H
211	NULL	NULL	FR-R92R-58	H	FR-R92R-58H
212	12,0278	2.40556	HL-U509-R	NULL	NULL
213	13,8782	2.77564	HL-U509-R	NULL	NULL
214	13,0863	2.61726	HL-U509-R	NULL	NULL
215	12,0278	2.40556	HL-U509	NULL	NULL
216	13,8782	2.77564	HL-U509	NULL	NULL

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ dbo.DimProduct.
- SELECT: Truy vấn các cột được chọn: StandardCose, StandardCostDiscount. ProductAlternateKey, Class, ProductName.

  NULL là trạng thái đáng chú ý của dữ liệu do nó có thể gây ra sai lệnh trong tính toán.

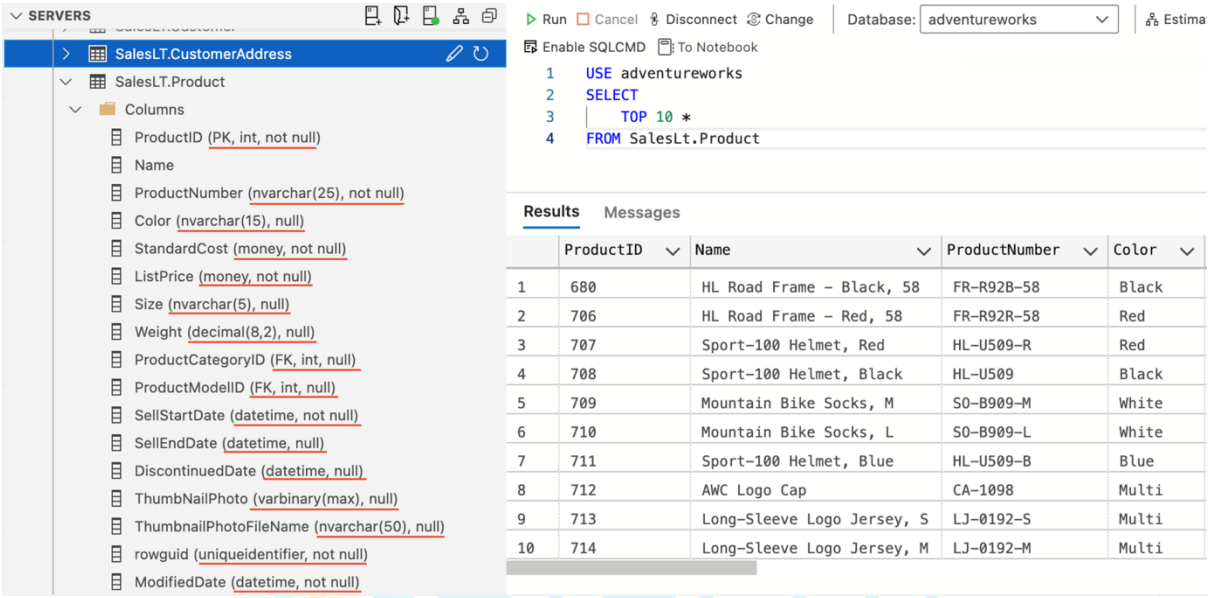
  Sử dụng * kết hợp giữa NULL và giá trị nhất định sẽ trả ra NULL.

© 2023 Datapot. All right reserved.

Sử dụng toán tử + kết hợp giữa NULL và giá trị nhất định sẽ trả ra NULL.

2.3. Kiểm tra kiểu dữ liệu của cột trong bảng

Bạn có thể kiểm tra kiểu dữ liệu trong bảng bằng cách ấn chuột vào thư mục Columns, bạn sẽ thấy định dạng dữ liệu được lưu trữ ở bên góc tay trái.



Hoặc bạn có thể sử dụng cú pháp sau để kiểm tra kiểu dữ liệu:

```
SELECT
    Col_name,
    ...
FROM INFORMATION_SCHEMA.COLUMNS
```

Để để kiểm tra kiểu dữ liệu trong bảng SalesOrderHeader thuộc bộ dữ liệu AdventureworksFULL. Ta sử dụng câu truy vấn sau:

© 2023 Datapot. All right reserved.

```sql
USE AdventureworksFull
SELECT
    COLUMN_NAME,
    DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE
    (TABLE_SCHEMA = 'Sales')
    AND (TABLE_NAME = 'SalesOrderHeader')
ORDER BY DATA_TYPE
```

Results Messages

	COLUMN_NAME	DATA_TYPE
1	OnlineOrderFlag	bit
2	OrderDate	datetime
3	DueDate	datetime
4	ShipDate	datetime
5	ModifiedDate	datetime
6	SalesOrderID	int
7	CurrencyRateID	int
8	CustomerID	int
9	SalesPersonID	int
10	TerritoryID	int

Giải thích câu lệnh truy vấn:

- FROM INFORMATION_SCHEMA.COLUMNS: Bảng mà cần truy vấn
- WHERE (TABLE_SCHEMA = 'Sales') AND (TABLE_NAME = 'SalesOrderHeader'): Là điều kiện để lọc dữ liệu. Điều kiện ở đây là truy vấn thông tin có trong bảng SalesOferHeader thuộc lược đồ Sales.
- SELECT: Dùng để chọn các cột muốn xuất hiện trong kết quả truy vấn.

© 2023 Datapot. All right reserved.

- ORDER BY DATA_TYPE: Kết quả sẽ được sắp xếp theo DATA_TYPE thứ tự từ A-Z.
- Để đối chiếu, ta sử dụng câu lệnh sau để xem các thông tin trong bảng SalesOrderHeader.

Ta có thể thấy:

- SalesOderID được lưu trữ dưới dạng int.
- OderDate, DueDate, ShipDate được lưu trữ dưới dạng datetime.

Tương tự với các cột khác.

```
USE AdventureWorksFull
SELECT TOP 10 *
FROM Sales.SalesOrderHeader
```

Results | Messages

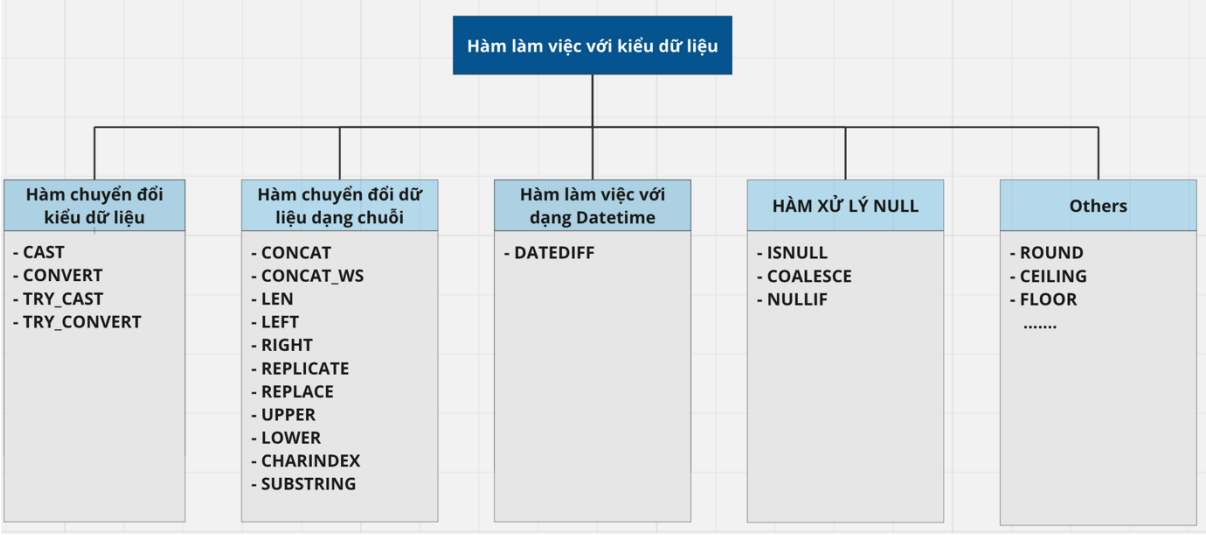
	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOr
1	43659	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
2	43660	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
3	43661	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
4	43662	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
5	43663	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
6	43664	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
7	43665	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
8	43666	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0

0 Ln 4, Col 1 Spaces: 4 UTF-8 CRLF SQL 10 rows MSSQL 00:00:00 sql-st.datapot.edu.vn : AdventureWorksFull

2.4. Các hàm làm việc tương ứng kiểu dữ liệu

2.4.1. Các hàm chuyển đổi kiểu dữ liệu

Một số loại kiểu hàm chuyển đổi dữ liệu chính trong SQL.



© 2023 Datapot. All right reserved.

Trong quá trình truy vấn dữ liệu, bạn sẽ thường xuyên phải update kiểu dữ liệu (Data types). Dưới đây là một số hàm giúp chuyển đổi kiểu dữ liệu:

2.4.1.1. Hàm chuyển đổi kiểu dữ liệu

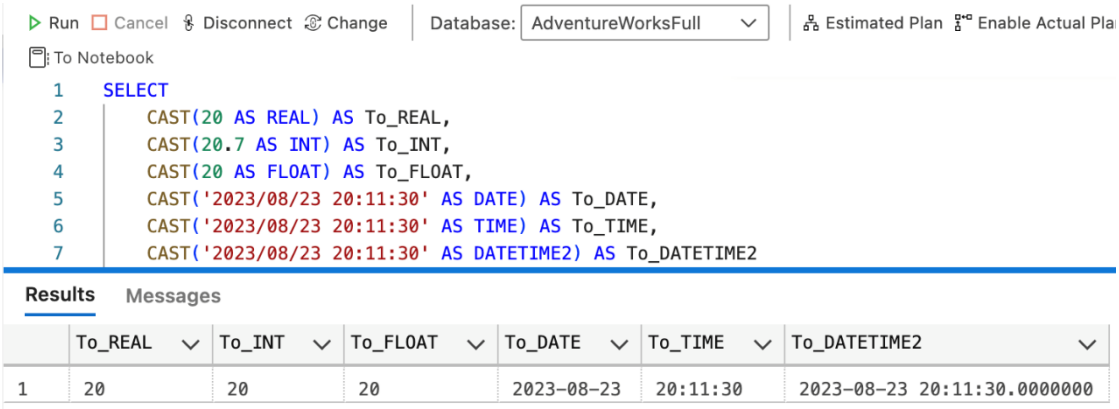
- **Hàm CAST**

Hàm cho phép chuyển đổi từ một kiểu dữ liệu sang một kiểu dữ liệu khác.

Cú pháp của hàm CAST:

CAST(Col/Expression AS Data type)

- Col/Expression: Cột/biểu thức cần chuyển đổi kiểu dữ liệu.
- Data type: Kiểu dữ liệu mà bạn muốn chuyển Col/Expression thành.



Giải thích câu lệnh truy vấn:

- SELECT: dùng để truy vấn các cột được chọn.
- CAST: được dùng để ép kiểu dữ liệu.

CAST(20 AS REAL) ép kiểu của giá trị 20 thành REAL.

CAST(20.7 AS INT) ép kiểu của giá trị 20.7 thành INT.

CAST(20 AS FLOAT) ép kiểu của giá trị 20 thành FLOAT.

CAST('2023/08/23 20:14:30' AS DATE) chuyển từ định dạng ngày/tháng/năm thời gian thành định dạng ngày.

© 2023 Datapot. All right reserved.

CAST('2023/08/23 20:14:30' AS TIME) chuyển từ định dạng ngày/tháng/năm thời gian thành định dạng thời gian.

CAST('2023/08/23 20:14:30' AS DATETIME2) chuyển từ định dạng ngày/tháng/năm thời gian thành định dạng ngày/tháng/năm thời gian cụ thể hơn.

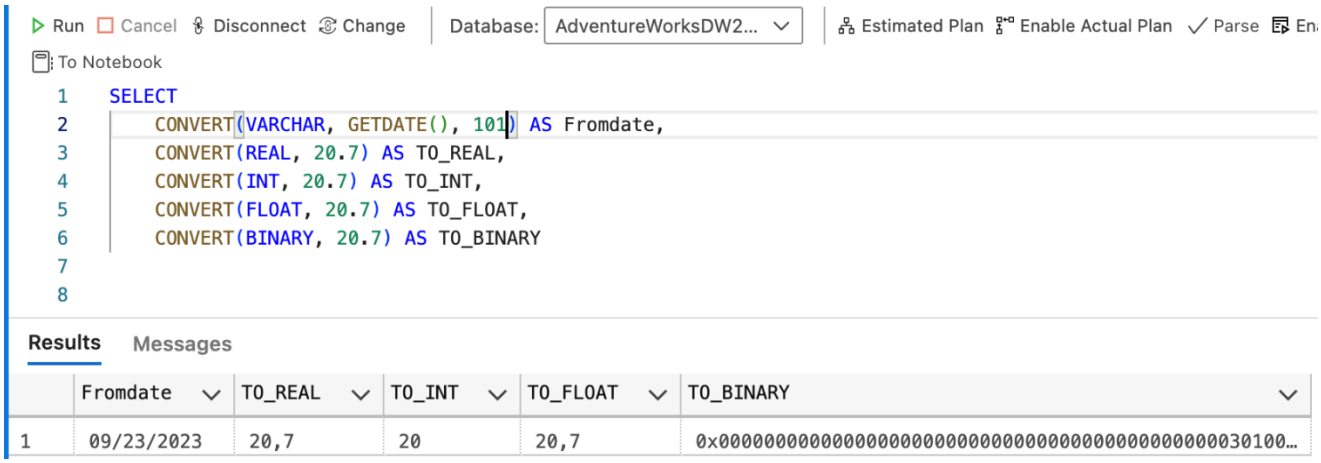
- **Hàm CONVERT**

Tương tự như hàm CAST, hàm CONVERT dùng để chuyển đổi từ một kiểu dữ liệu sang một kiểu dữ liệu khác. Tuy nhiên, hàm CONVERT hỗ trợ nhiều kiểu dữ liệu khác nhau (Binary; XML;...), chuyển định dạng ngày tháng.

Cú pháp của hàm CONVERT:

CONVERT(target_data_type, expression, style)

- Target_data_type: Kiểu dữ liệu mà bạn muốn chuyển đổi giá trị/biểu thức cần chuyển đổi.
- Expression: Giá trị hoặc biểu thức cần chuyển đổi.
- Style (Không bắt buộc): Thường áp dụng cho các chuyển đổi liên quan đến ngày tháng. Nó xác định cách định dạng ngày tháng được hiển thị hoặc chấp nhận.



Giải thích câu lệnh truy vấn:

- SELECT: Dùng để truy vấn các cột được chọn.

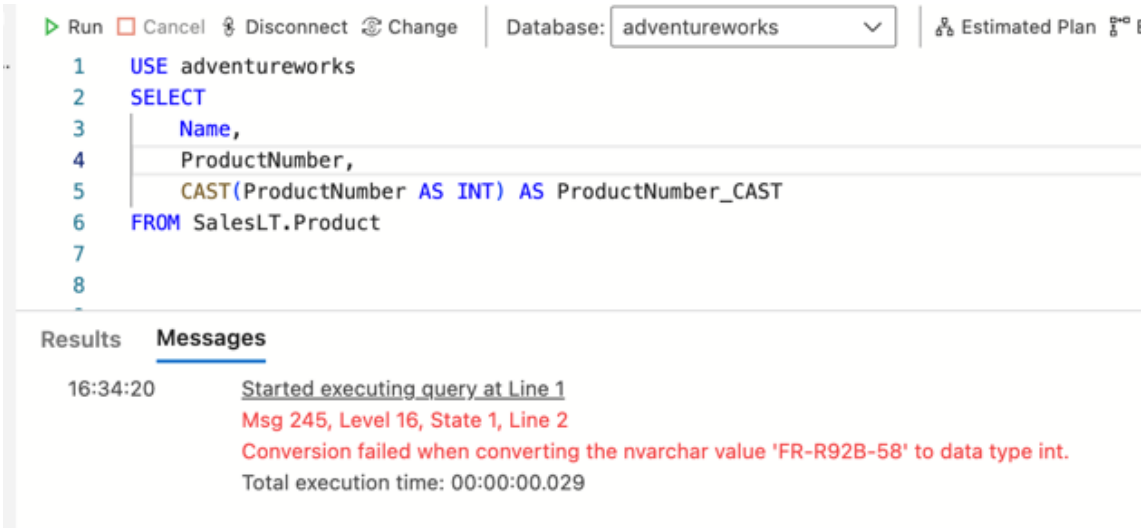
© 2023 Datapot. All right reserved.

- CONVERT(VARCHAR, GETDATE(), 101):
  - Kiểu dữ liệu muốn chuyển đổi thành kiểu VARCHAR.
  - Giá trị cần chuyển đổi là thời gian của ngày hiện tại, ta sử dụng hàm GETDATE() để lấy ngày hiện tại (23/09/2023)
  - Chuyển đổi dưới định dạng tháng/ngày/năm.
- CONVERT(REAL, 20.7) AS TO_REAL: Chuyển đổi kiểu dữ liệu sang kiểu REAL.
- CONVERT(INT, 20.7) AS TO_INT: Chuyển đổi kiểu dữ liệu sang kiểu INT.
- CONVERT(FLOAT, 20.7) AS TO_FLOAT: Chuyển đổi kiểu dữ liệu sang kiểu FLOAT.
- CONVERT(BINARY, 20.7) AS TO_BINARY: Chuyển đổi kiểu dữ liệu sang kiểu BINARY.

○ **Hàm TRY_CAST**

Cũng tương tự như CAST, TRY_CAST dùng để chuyển đổi từ một kiểu dữ liệu sang một kiểu dữ liệu khác. Tuy nhiên với những giá trị không chuyển đổi được kiểu dữ liệu, hàm TRY_CAST sẽ trả về dữ liệu không tồn tại (NULL).

Ví dụ: Từ bảng SalesLT.Product thuộc bộ dữ liệu adventureworks, truy vấn các cột sau: Name, ProductNumber và chuyển kiểu dữ liệu của cột ProductNumver thành dạng INT và lưu dưới tên ProductNumber_CAST



© 2023 Datapot. All right reserved.

Giải thích câu lệnh truy vấn:

- SELECT: Dùng để truy vấn các cột Name, ProductNumber, ProductNumber_CAST
- CAST(ProductNumber AS INT): Chuyển đổi kiểu dữ liệu của ProductNumber thành định dạng INT.

Tuy nhiên, việc chuyển kiểu dữ liệu của cột ProductNumber thành định dạng INT bị lỗi. Vì vậy, hàm CAST() trả về lỗi truy vấn.

Trong khi đó, vẫn với ví dụ trên, ta sử dụng hàm TRY_CAST:

Run Cancel Disconnect Change Database: adventureworks Estimated Pla

```
USE adventureworks
SELECT
    Name,
    ProductNumber,
    TRY_CAST(ProductNumber AS INT) AS ProductNumber_TRY_CAST
FROM SalesLT.Product
```

Results Messages

	Name	ProductNumber	ProductNumber_TRY_CAST
1	HL Road Frame - Black, 58	FR-R92B-58	NULL
2	HL Road Frame - Red, 58	FR-R92R-58	NULL
3	Sport-100 Helmet, Red	HL-U509-R	NULL
4	Sport-100 Helmet, Black	HL-U509	NULL
5	Mountain Bike Socks, M	SO-B909-M	NULL
6	Mountain Bike Socks, L	SO-B909-L	NULL
7	Sport-100 Helmet, Blue	HL-U509-B	NULL
8	AWC Logo Cap	CA-1098	NULL
9	Long-Sleeve Logo Jersey, S	LJ-0192-S	NULL

Giải thích câu lệnh truy vấn:

- SELECT: Dùng để truy vấn các cột Name, ProductNumber, ProductNumber_CAST
- TRY_CAST(ProductNumber AS INT): Chuyển đổi kiểu dữ liệu của ProductNumber thành định dạng INT.

© 2023 Datapot. All right reserved.

Tuy nhiên, việc chuyển kiểu dữ liệu của cột ProductNumber (Dạng Varchar) thành định dạng INT là không thể. Vì vậy, hàm TRY_CAST() trả về NULL, thay vì báo lỗi như hàm CAST thông thường.

- **Hàm TRY_CONVERT**

Tương tự như hàm CONVERT, TRY_CONVERT cho phép chuyển đổi từ một kiểu dữ liệu sang một kiểu dữ liệu khác. Tuy nhiên, khi chuyển đổi không thành công, hàm TRY_CONVERT trả về NULL thay vì lỗi như hàm CAST.

Cú pháp hàm TRY_CONVERT:

TRY_CONVERT(expression AS data_type)

- Expression: Giá trị hoặc biểu thức mà bạn muốn chuyển đổi sang kiểu dữ liệu mới.
- Data_type: Kiểu dữ liệu mà bạn muốn chuyển đổi.

Enable SQLCMD To Notebook

```sql
SELECT
    CONVERT(INT, 20.7),
    TRY_CONVERT(INT, 'abc')
```

Results Messages

	(No column name)	(No column name)
1	20	NULL

Giải thích câu lệnh truy vấn:

- SELECT: Dùng để truy vấn các cột được chọn.
- CONVERT(): Dùng để chuyển kiểu định dạng của 20.7 thành INT.
- TRY_CONVERT: Dùng để chuyển kiểu định dạng của 'abc' thành INT.

Có thể thấy, hàm TRY_CONVERT trả về NULL khi chuyển đổi giá trị không thành công.

© 2023 Datapot. All right reserved.

```
SELECT
    CONVERT(INT, 'abc')
```

Results | Messages

20:20:22 Started executing query at Line 1
Msg 245, Level 16, State 1, Line 1
Conversion failed when converting the varchar value 'abc' to data type int.
Total execution time: 00:00:00.034

Giải thích câu lệnh truy vấn:

- SELECT: Dùng để truy vấn các cột được chọn.
- Hàm CONVERT sẽ lỗi khi chuyển kiểu định dạng của 'abc' thành INT.

Hàm CONVERT trả về lỗi truy vấn khi chuyển đổi giá trị không thành công.

Sau đây là một số tham số về định dạng chỉ ngày, thời gian, ngày&giờ phổ biến:

- Định dạng chỉ ngày.

Định dạng #	Truy vấn	Định dạng	Kết quả
1	CONVERT(VARCHAR, GETDATE(), 1)	MM/DD/YY	08/28/23
2	CONVERT(VARCHAR, GETDATE(), 2)	YY.MM.DD	23/08/28
3	CONVERT(VARCHAR, GETDATE(), 3)	DD/MM/YY	28/08/23
4	CONVERT(VARCHAR, GETDATE(), 4)	DD.MM.YY	28.08.23
5	CONVERT(VARCHAR, GETDATE(), 5)	DD-MM-YY	28-08-23

© 2023 Datapot. All right reserved.

```sql
SELECT
    CONVERT(VARCHAR, GETDATE(), 1) AS [MM/DD/YY],
    CONVERT(VARCHAR, GETDATE(), 2) AS [YY.MM.DD],
    CONVERT(VARCHAR, GETDATE(), 3) AS [DD/MM/YY],
    CONVERT(VARCHAR, GETDATE(), 4) AS [DD.MM.YY],
    CONVERT(VARCHAR, GETDATE(), 5) AS [DD-MM-YY]
```

Results | Messages

	MM/DD/YY	YY.MM.DD	DD/MM/YY	DD.MM.YY	DD-MM-YY
1	08/28/23	23.08.28	28/08/23	28.08.23	28-08-23

- Định dạng chỉ thời gian:

Định dạng #	Truy vấn	Định dạng	Kết quả
8	CONVERT(VARCHAR, GETDATE(), 8)	HH:MM:SS	09:34:11
14	CONVERT(VARCHAR, GETDATE(), 14)	HH:MM:SS:NNN	09:34:11:363
24	CONVERT(VARCHAR, GETDATE(), 24)	HH:MM:SS	09:34:11
108	CONVERT(VARCHAR, GETDATE(), 108)	DD.MM.YY	09:34:11
114	CONVERT(VARCHAR, GETDATE(), 114)	HH:MM:SS:NNN	09:34:11:363

© 2023 Datapot. All right reserved.

Welcome | SQLQuery_1 - sql-st...tudent)

Run | Cancel | Disconnect | Change | Database: AdventureWorksFull | Estimated Plan

```sql
SELECT
    CONVERT(VARCHAR, GETDATE(), 8) AS [HH:MM:SS],
    CONVERT(VARCHAR, GETDATE(), 14) AS [HH:MM:SS:NNN],
    CONVERT(VARCHAR, GETDATE(), 24) AS [HH:MM:SS],
    CONVERT(VARCHAR, GETDATE(), 108) AS [DD.MM.YY],
    CONVERT(VARCHAR, GETDATE(), 114) AS [HH:MM:SS:NNN]
```

Results | Messages

HH:MM:SS	HH:MM:SS:NNN	HH:MM:SS	DD.MM.YY	HH:MM:SS:NNN
09:34:11	09:34:11:363	09:34:11	09:34:11	09:34:11:363

- Định dạng chỉ ngày & giờ.

Định dạng #	Truy vấn	Định dạng	Kết quả
0	CONVERT(VARCHAR, GETDATE(), 0)	MON DD YYYY HH:MM AM/PM	Aug 28 2023 9:43AM
9	CONVERT(VARCHAR, GETDATE(), 9)	MON DD YYYY HH:MM:SS:NNN AM/PM	Aug 28 2023 9:43:51:963AM
13	CONVERT(VARCHAR, GETDATE(), 13)	DD MON YYYY HH:MM:SS:NNN AM/PM	28 Aug 2023 09:43:51:963
20	CONVERT(VARCHAR, GETDATE(), 20)	YYYY-MM-DD HH:MM:SS	2023-08-28 09:43:51
109	CONVERT(VARCHAR, GETDATE(), 109)	MON DD YYYY HH:MM:SS:NNN AM/PM	Aug 28 2023 9:43:51:963AM

© 2023 Datapot. All right reserved.

```sql
SELECT
    CONVERT(VARCHAR, GETDATE(), 0) AS [MON DD YYYY HH:MM AM/PM],
    CONVERT(VARCHAR, GETDATE(), 9) AS [DD MON YYYY HH:MM:SS:NNN AM/PM],
    CONVERT(VARCHAR, GETDATE(), 13) AS [DD MON YYYY HH:MM:SS:NNN AM/PM],
    CONVERT(VARCHAR, GETDATE(), 20) AS [YYYY-MM-DD HH:MM:SS],
    CONVERT(VARCHAR, GETDATE(), 109) AS [MON DD YYYY HH:MM:SS:NNN AM/PM]
```

Results | Messages

	MON DD YYYY HH:...	DD MON YYYY HH:MM:SS:N...	DD MON YYYY HH:MM:S...	YYYY-MM-DD HH:...	MON DD YYYY HH:MM:SS:NNN AM/PM
1	Oct 12 2023 9:31AM	Oct 12 2023 9:31:15:243AM	12 Oct 2023 09:31:15:243	2023-10-12 09:31:15	Oct 12 2023 9:31:15:243AM

0 Ln 7, Col 5 Spaces: 4 UTF-8 CRLF SQL 1 rows MSSQL 00:00:00 sql-st.datapot.edu.vn : AdventureWorksFull

2.4.1.2. Xử lý dữ liệu dạng chuỗi

- **Hàm CONCAT**

Hàm CONCAT được sử dụng để kết hợp nhiều chuỗi lại với nhau và tạo ra một chuỗi duy nhất.

Cùng là kết hợp, tuy nhiên khi kết hợp nhiều chuỗi bao gồm cả NULL

Hàm CONCAT sẽ vẫn kết hợp chuỗi và bỏ qua NULL.

Một cách làm khác để kết hợp chuỗi là sử dụng dấu “+”.

Cú pháp của hàm CONCAT:

CONCAT(string1, string2, ...)

- string1, string2: Các tham số là chuỗi hoặc biểu thức chuỗi muốn nối lại với nhau.

Ví dụ: Từ bảng dữ liệu SalesLT.Customer thuộc bộ dữ liệu adventureworks, truy vấn các cột FirstName, LastName và kết hợp 2 cột trên lưu dưới tên FullName bằng hàm CONCAT và toán tử “+”.

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    FirstName,
    LastName,
    FirstName + ' ' + LastName AS FullName,
    CONCAT(FirstName, ' ', LastName) AS Concat_FullName
FROM SalesLT.Customer
```

Results | Messages

	FirstName	LastName	FullName	Concat_FullName
1	Orlando	Gee	Orlando Gee	Orlando Gee
2	Keith	Harris	Keith Harris	Keith Harris
3	Donna	Carreras	Donna Carreras	Donna Carreras
4	Janet	Gates	Janet Gates	Janet Gates
5	Lucy	Harrington	Lucy Harrington	Lucy Harrington

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu truy vấn được lấy từ bảng SalesLT.Customer.
- SELECT: truy vấn các cột được chọn: FirstName, LastName và FullName được kết hợp từ 2 cột trên.

Hàm CONCAT và toán tử "+" được sử dụng để kết hợp 2 cột FirstName, LastName với nhau.

- **Hàm CONCAT_WS**

Tương tự như hàm CONCAT, hàm CONCAT_WS sẽ kết hợp nhiều chuỗi bằng cách sử dụng một phân tách thành một chuỗi duy nhất. Ngoài ra, khi kết hợp các cột có NULL, hàm CONCAT_WS sẽ vấn kết hợp chuỗi và bỏ qua NULL.

Cú pháp của hàm CONCAT_WS:

CONCAT(separator, col_1, col_2, ...)

- separator: Chuỗi phân tách mà bạn muốn sử dụng để ngăn cách các chuỗi với nhau.
- col_1, col_2: Các tham số là chuỗi hoặc biểu thức chuỗi mà bạn muốn nối với nhau sau khi được phân tách.

© 2023 Datapot. All right reserved.

Ví dụ: Từ bảng dữ liệu SalesLT.Customer, truy vấn các cột Title, FirstName, LastName và kết hợp 3 cột trên lưu dưới tên FullName bằng hàm CONCAT_WS.

Parse Enable SQLCMD To Notebook

```sql
USE adventureworks
SELECT
    Title,
    FirstName,
    LastName,
    CONCAT_WS(' ', Title, FirstName, LastName) AS FullName
FROM SalesLT.Customer
```

Results Messages

	Title	FirstName	LastName	FullName
1	Mr.	Orlando	Gee	Mr. Orlando Gee
2	Mr.	Keith	Harris	Mr. Keith Harris
3	Ms.	Donna	Carreras	Ms. Donna Carreras
4	Ms.	Janet	Gates	Ms. Janet Gates
5	Mr.	Lucy	Harrington	Mr. Lucy Harrington
6	Ms.	Rosmarie	Carroll	Ms. Rosmarie Carroll
7	Mr.	Dominic	Gash	Mr. Dominic Gash
8	Ms.	Kathleen	Garza	Ms. Kathleen Garza
9	Ms.	Katherine	Harding	Ms. Katherine Harding

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu truy vấn được lấy từ bảng SalesLT.Customer.
- SELECT: Truy vấn các cột được chọn: Title, FirstName, LastName và FullName được kết hợp từ 3 cột trên.

Thay vì phải điền các chuỗi phân tách như hàm CONCAT, bạn chỉ cần điều chuỗi phân tách 1 lần trong hàm CONCAT_WS.

- **Hàm LEN**

Hàm LEN được sử dụng dùng để tính độ dài của chuỗi ký tự.

Cú pháp của hàm LEN:

LEN(string)

- string: chuỗi ký tự mà bạn muốn tính độ dài.

© 2023 Datapot. All right reserved.

Ví dụ: Từ bảng dữ liệu SalesLT.Customer thuộc bộ dữ liệu adventureworks, truy vấn các cột FirstName và tính độ dài của tên khách hàng.

Estimated Plan Enable Actual Plan Parse Enal

```
1  USE adventureworks
2  SELECT
3      FirstName,
4      LEN(FirstName) AS FirstNameLen
5  FROM SalesLT.Customer
```

Results Messages

	FirstName	FirstNameLen
1	Orlando	7
2	Keith	5
3	Donna	5
4	Janet	5
5	Lucy	4

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu truy vấn được lấy từ bảng SalesLT.Customer.
- SELECT: truy vấn các cột được chọn: FirstName, FirstNameLen.

Hàm LEN tính toán độ dài của tên khách hàng.

- **Hàm LEFT & RIGHT**

Hàm LEFT dùng để trích xuất một số ký tự cụ thể từ bên trái của ký tự.

Hàm RIGHT dùng để trích xuất một số ký tự cụ thể từ bên phải của ký tự.

Cú pháp của hàm LEFT và hàm RIGHT:

LEFT/RIGHT(string, num_characters)

- string: Chuỗi ký tự mà bạn muốn trích xuất từ bên trái/phải.

© 2023 Datapot. All right reserved.

- length: Số lượng ký tự mà bạn muốn trích xuất từ phía trái/phải của chuỗi.

Ví dụ: Truy vấn 2 ký tự từ bên trái và 2 ký tự bên phải của từ 'Hello'.

```
1  SELECT
2      LEFT('Hello', 2),
3      RIGHT('Hello', 2)
```

Results | Messages

	(No column name)	(No column name)
1	He	lo

Giải thích câu lệnh truy vấn.

- SELECT: Hiển thị kết quả truy vấn.

LEFT(): trả về 2 ký tự từ bên trái qua phải của từ 'Hello'.

RIGHT(): trả về 2 ký tự từ bên phải qua trái của từ 'Hello'.

- **Hàm REPLICATE**

Hàm REPLICATE được sử dụng để lặp lại một chuỗi với số lần nhất định.

Cú pháp hàm REPLICATE:

REPLICATE(string, n_times)

- string: Chuỗi hoặc ký tự muốn lặp lại.
- n_times: Số lần lặp lại chuỗi hoặc ký tự.

Ví dụ: Từ bảng SalesLT. Customer thuộc bộ dữ liệu adventureworks, truy vấn LastName và LastName2 lặp lại thông tin LastName 2 lần.

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    LastName,
    REPLICATE(LastName, 2) AS LastName2
FROM SalesLT.Customer
```

Results | Messages

	LastName	LastName2
1	Gee	GeeGee
2	Harris	HarrisHarris
3	Carreras	CarrerasCarreras
4	Gates	GatesGates
5	Harrington	HarringtonHarrington

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ bảng SalesLT.Customer.
- SELECT: Truy vấn các cột LastName, LastName2.

Hàm REPLICATE lặp lại giá trị của cột LastName 2 lần.

- **Hàm REPLACE**

Hàm REPLACE thay thế một chuỗi ký tự trong chuỗi ký tự gốc.

Cú pháp hàm REPLACE:

REPLACE(string, to_replace, replacement)

- string: Chuỗi ký tự gốc.
- to_replace: Chuỗi ký tự được thay thế trong ký tự gốc.
- replacement: Chuỗi ký tự thay thế.

Ví dụ: Từ bảng SalesLT.Product thuộc bộ dữ liệu adventureworks, truy vấn các cột Name, ProductNumber, và thay thế ký tự 'BK' trong cột ProductNumber thành ký tự 'XX'.

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    Name,
    ProductNumber,
    REPLACE(ProductNumber, 'BK-', 'XX-') AS NewProductNumber
FROM SalesLT.Product
```

Results | Messages

	Name	ProductNumber	NewProductNumber
62	Road-650 Black, 60	BK-R50B-60	XX-R50B-60
63	Road-650 Black, 62	BK-R50B-62	XX-R50B-62
64	Road-650 Black, 44	BK-R50B-44	XX-R50B-44
65	Road-650 Black, 48	BK-R50B-48	XX-R50B-48
66	Road-650 Black, 52	BK-R50B-52	XX-R50B-52
67	Mountain-100 Silver, 38	BK-M82S-38	XX-M82S-38
68	Mountain-100 Silver, 42	BK-M82S-42	XX-M82S-42

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ bảng SalesLT.Product.
- SELECT: Truy vấn các cột Name, ProductNumber, NewProductNumber.
  Hàm REPLACE thay đổi ký tự 'BK' trong ký tự gốc tại cột ProductNumber thánh ký tự 'XX'.

- **Hàm UPPER**

Hàm UPPER cho phép chuyển đổi chữ cái sang dạng viết hoa

Cú pháp của hàm UPPER:

UPPER(nvarchar)

- **Hàm LOWER**

Hàm LOWER cho phép chuyển đổi chữ cái sang dạng viết thường

Cú pháp của hàm LOWER:

© 2023 Datapot. All right reserved.

LOWER(nvarchar)

Ví dụ: Bạn muốn viết hoa FirstName, viết thường Middle Name.

```
USE adventureworks
SELECT
    UPPER(FirstName) AS Upper_FirstName,
    LOWER(MiddleName) AS Lower_MiddleName
FROM SalesLT.Customer
```

Results grid

Results | Messages

	Upper_FirstName	Lower_MiddleName
1	ORLANDO	n.
2	KEITH	*NULL*
3	DONNA	f.
4	JANET	m.
5	LUCY	*NULL*
6	ROSMARIE	j.
7	DOMINIC	p.

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ bảng SalesLT.Customer.
- SELECT: Truy vấn các cột Upper_FirstName và Lower_MiddleName.

  Hàm UPPER chuyển đổi FirstName thành chữ cái viết hoa.

  Hàm LOWER chuyển đổi MiddleName thành chữ cái viết thường.

- **Hàm CHARINDEX:**

CHARINDEX được dùng để tìm kiếm chuỗi ký tự trong chuỗi ký tự gốc và trả về vị trí đầu tiên của chuỗi khớp nếu có. Ngược lại, nếu không tìm thấy chuỗi ký tự con trong chuỗi ký tự gốc, hàm CHARINDEX sẽ trả về NULL.

Cú pháp của CHARINDEX:

CHARINDEX(substr, str, [start_location])

- substr: Chuỗi ký tự tìm kiếm trong chuỗi gốc.
- str: Chuỗi gốc.
- start_location: Vị trí bắt đầu tìm kiếm (Nếu có).

Ví dụ: Đếm ra có bao nhiêu ký tự đứng trước '@' của cột EmailAddress.

© 2023 Datapot. All right reserved.

```sql
SELECT
    FirstName,
    EmailAddress,
    CHARINDEX('@', EmailAddress) AS position_@
FROM SalesLT.Customer
```

Results | Messages

	FirstName	EmailAddress	position_@
1	Orlando	*NULL*	*NULL*
2	Keith	keith0@adventure-works.com	7
3	Donna	donna0@adventure-works.com	7
4	Janet	janet1@adventure-works.com	7
5	Lucy	lucy0@adventure-works.com	6

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ bảng SalesLT.Customer.
- SELECT: Truy vấn các cột FirstName, EmailAddress và position_@. Hàm CHARINDEX đếm vị trí của '@' trong chuỗi ký tự EmailAddress.

○ **Hàm SUBSTRING**

Hàm SUBSTRING dùng để trích xuất một phần của chuỗi ký tự từ một vị trí bắt đầu và độ dài cụ thể.

Cú pháp của SUBSTRING:

SUBSTRING(input_string, start_position, length)

- input_string: Chuỗi ký tự cần trích xuất.
- start_position: Vị trí bắt đầu trích xuất.
- length: Độ dài phần cần trích xuất.

Ví dụ: Truy vấn tên người bán hàng từ cột SalesPerson bằng SUBSTRING.

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    Title,
    LastName,
    CompanyName,
    SalesPerson,
    SUBSTRING(SalesPerson, 17, 10) AS NAME_SalesPerson
FROM SalesLT.Customer
```

Results | Messages

	Title	LastName	CompanyName	SalesPerson	NAME_SalesPerson
1	Mr.	Gee	A Bike Store	adventure-works\pamela0	pamela0
2	Mr.	Harris	Progressive Sports	adventure-works\david8	david8
3	Ms.	Carreras	Advanced Bike Components	adventure-works\jillian0	jillian0
4	Ms.	Gates	Modular Cycle Systems	adventure-works\jillian0	jillian0
5	Mr.	Harrington	Metropolitan Sports Supply	adventure-works\shu0	shu0

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ bảng SalesLT.Customer.
- SELECT: Truy vấn các cột Title, LastName, CompanyName, SalesPerson, NAME_SalesPerson.
  Hàm SUBSTRING trích xuất tên người bán hàng.

2.4.1.3. Xử lý và thao tác với dữ liệu dạng ngày (Date manipulations)

Xử lý và thao tác với dữ liệu dạng ngày là một trong những việc phổ biến khi truy vấn dữ liệu.

○ **Hàm DATEDIFF**

Hàm DATEDIFF được sử dụng để tính khoảng thời gian giữa hai thời điểm thời gian. Hàm này trả về một giá trị số nguyên, đại diện cho sự khác biệt giữa các ngày hoặc thời điểm dựa trên một đơn vị thời gian cụ thể (Ví dụ: ngày, tháng, năm).

Cú pháp của DATEDIFF

DATEDIFF(date_part, start_date, end_date)

- Date_part: Đơn vị thời gian mà bạn muốn tìm khoảng các, có thể là năm , tháng, ngày,...
- Start_date: Ngày hoặc thời điểm bắt đầu khoảng thời gian.
- End_date: Ngày hoặc thời điểm kết thúc khoảng thời gian.

© 2023 Datapot. All right reserved.

Ví dụ: Từ bảng SalesOderHear thuộc bộ dữ liệu adventureworks, truy vấn SalesOrderID, OrderDate, ShipDate, Khoảng cách thời gian Order và Shipping theo Ngày/Tuần/Tháng.

```sql
USE AdventureworksFull
SELECT
    SalesOrderID,
    OrderDate,
    ShipDate,
    -- diff in days change
    DATEDIFF(DAY, OrderDate, ShipDate) AS DayDiff,
    -- diff in week change
    DATEDIFF(WEEK, OrderDate, ShipDate) AS WeekDiff,
    -- diff in months change
    DATEDIFF(MONTH, OrderDate, ShipDate) AS MonthDiff
FROM Sales.SalesOrderHeader
```

Results | Messages

	SalesOrderID	OrderDate	ShipDate	DayDiff	WeekDiff	MonthDiff
1	43659	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
2	43660	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
3	43661	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
4	43662	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
5	43663	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
6	43664	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
7	43665	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
8	43666	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
9	43667	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
10	43668	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
11	43669	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1
12	43670	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	7	1	1

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ bảng SalesLT.SalesOrderHeader.
- SELECT: Truy vấn các cột SalesOrderID, OrderDate, ShipDate, Khoảng cách thời gian Order và Shipping theo ngày/tuần/tháng.

DATEDIFF() tính khoảng thời gian giữa thời điểm OrderDate và ShipDate theo mốc ngày/tuần/tháng.

2.4.1.4. Xử lý các dữ liệu không tồn tại hoặc dữ liệu không có sẵn (NULL)

Trên phần 2.2.4 đã nêu khái niệm về NULL. Sau đây ta cùng tìm hiểu một số các hàm xử lý NULL.

- **Hàm ISNULL**

Hàm ISNULL trả về một giá trị được chỉ định nếu biểu thức là NULL. Nếu biểu thức không phải là NULL, hàm này trả về chính biểu thức đó.

© 2023 Datapot. All right reserved.

Cú pháp hàm ISNULL:

ISNULL(expression, value)

- expression: Biểu thức cần kiểm tra là NULL hay không
- value: Giá trị sẽ được trả về nếu biểu thức là NULL.

Ví dụ: Từ bảng SalesLT.Product thuộc bộ dữ liệu adventureworks, sử dụng ISNULL để so sánh kết quả trong cột Color. Nếu cột Color có NULL, trả về tên NoColor.

Parse | Enable SQLCMD | To Notebook

```
1  USE adventureworks
2  SELECT
3      ISNULL(Color, 'NoColor')
4  FROM SalesLT.Product
5
```

Results | Messages

	(No column name)
224	NoColor
225	NoColor
226	NoColor
227	NoColor
228	NoColor
229	NoColor
230	NoColor
231	Silver/Black
232	Silver/Black

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ SalesLT.Product.
- SELECT: Truy vấn các cột được chọn.

  Hàm ISNULL trả về NoColor khi biểu thức là NULL.

○ **Hàm COALESCE**

© 2023 Datapot. All right reserved.

Hàm COALESCE là được sử dụng để trả về giá trị đầu tiên không phải NULL từ một danh sách các biểu thức.

Cú pháp hàm COALESCE:

COALESCE(expression1, expression2, …)

- expression1, expression2: Danh sách các biểu thức bạn muốn kiểm tra và trả về giá trị đầu tiên không phải là NULL.

Ví dụ: Từ bảng Customer, truy vấn cột Class, ProductLine và ProductClass được tạo nên bởi cột Class và ProductLine. Với tên không xác định hoặc không có sẵn (NULL), hiển thị với giá trị 'Unknown'.

Run Cancel Disconnect Change Database: AdventureWorksDW2...
To Notebook

```sql
SELECT
    Class,
    ProductLine,
    COALESCE(Class, ProductLine, 'Unknown') AS ProductClass
FROM dbo.DimProduct
```

Results Messages

	Class	ProductLine	ProductClass
198	L	NULL	L
199	NULL	NULL	Unknown
200	NULL	NULL	Unknown
201	NULL	NULL	Unknown
202	NULL	NULL	Unknown
203	NULL	NULL	Unknown
204	NULL	NULL	Unknown
205	NULL	NULL	Unknown
206	NULL	NULL	Unknown
207	NULL	NULL	Unknown
208	NULL	NULL	Unknown
209	NULL	NULL	Unknown
210	H	R	H
211	H	R	H
212	NULL	S	S
213	NULL	S	S
214	NULL	S	S

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu được truy vấn từ bảng SalesLT.Customer.

© 2023 Datapot. All right reserved.

- SELECT: Truy vấn các cột được chọn: Class, ProductLine và ProductClass và cột kết hợp từ 2 cột trên.
  Hàm COALESCE: kết hợp 2 cột Class và ProductLine. Khi có NULL xuất hiện tại biểu thức tham chiều đầu tiên (Cột Class), kết quả trả về tên 'Unknown'.
  Hàng 198 có biểu thức thứ 1 không NULL nên COALESCE trả về giá trị biểu thức đầu tiên L.
  Hàng 200 có biểu thức thứ 1, 2 NULL nên COALESCE trả về 'Unknown'.
  Hàng 210 có biểu thức thứ 1 không NULL nên COALESCE trả về giá trị biểu thức thứ 1 H.
  Hàng 212 có biểu thức thứ 2 không NULL nên COALESCE trả về giá trị biểu thức thứ 2 S.

- **Hàm NULLIF**

Hàm NULLIF được sử dụng để so sánh hai giá trị và trả về NULL nếu chúng bằng nhau.

Cú pháp hàm NULLIF:

NULLIF(expression1, expression2)

- expression1: Giá trị bạn muốn so sánh.
- expression2: Giá trị mà bạn muốn so sánh với expression1.

Ví dụ: so sánh giữa giá trị 5 và 5, 10 và 5 bằng hàm NULLIFF

```
1  SELECT
2      NULLIF(5, 5),
3      NULLIF(10, 5)
```

Results Messages

	(No column name)	(No column name)
1	NULL	10

© 2023 Datapot. All right reserved.

Giải thích câu lệnh truy vấn:

- SELECT: Truy vấn các cột được chọn.
  Hàm NULLIF trả về NULL khi 2 giá trị so sánh bằng nhau.
  Hàm NULLIF trả về giá trị đầu tiên khi 2 giá trị so sánh khác nhau.

2.5. Nhóm hàm định dạng dữ liệu (Formatting data)

2.5.1. Định dạng ngày (Date formats)

Hầu hết cơ sở dữ liệu sẽ có hàm (Function) để giúp định dạng ngày. Cần thống nhất định dạng ngày (Date formats) để đảm bảo tính nhất quán trong cơ sở dữ liệu.

Ví dụ: Định dạng ngày của Việt Nam và Mỹ:

- Việt Nam: DD/MM/YYYY
- Mỹ: MM/DD/YYYY

Vì vậy, cần hiểu cơ sở dữ liệu của mình đang lưu trữ dưới định dạng ngày (Date formats) nào để đảm bảo tính nhất quán trong việc lưu trữ.

- **Hàm FORMAT:**

Hàm FORMAT được dùng để định dạng lại giá trị ngày tháng hoặc số dưới dạng chuỗi theo các định dạng tùy chỉnh.

Cú pháp của FORMAT:

FORMAT(col_name, format, [culture])

- col_name: giá trị cần định dạng.
- format: Dạng fomat mà bạn muốn áp dụng, có thể là định dạng thời gian hoặc định dạng số.
- culture (tùy chọn): Mã vùng để xác định cách định dạng, mặc định là 'NULL'.

Ví dụ: Format 24/09/2023 theo định dạng ngày, tháng, năm. Format giá trị 20.3 dưới dạng phần trăm và format 24/09/2023 theo định dạng ngày

© 2023 Datapot. All right reserved.

Run Cancel Disconnect Change Database: AdventureWorksFull

Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

```sql
SELECT
    FORMAT(2023-09-24, 'dd/MM/yyyy') AS Dateformat,
    FORMAT(20.3, 'P') AS Numberformat,
    FORMAT(2023-09-24, 'dd/MM/yyyy', 'fr-FR') AS Dateformatwithculture
```

Results Messages

	Dateformat	Numberformat	Dateformatwithculture
1	dd/MM/yyyy	2,030.00%	dd/MM/yyyy

Giải thích câu lệnh truy vấn:

- SELECT: Hiển thị các cột đã chọn: Dateformat, Numberformat và Dateformatwithculture.
  FORMAT(2023-09-24, 'dd/MM/yyyy') giá trị ngày tháng năm được định dạng lại dưới format ngày/tháng/năm.
  FORMAT(20.3, 'P') giá trị 20.3 được định dạng dưới format phần trăm.
  FORMAT(2023-09-24, 'dd/MM/yyyy', 'fr-FR') giá trị ngày tháng năm được định dạng lại dưới format ngày/tháng/năm, không bao gồm giờ, phút, giây sử dụng mã vùng là 'fr-FR' (tiếng Pháp - Pháp).

2.5.2. Làm tròn số (Rounding)

Một số hàm làm tròn số trong trong SQL:

- ROUND(): Làm tròn tới chứ số thập phân (Float).
- CEILING(): Làm tròn lên.
- FLOOR(): Làm tròn xuống.

Ví dụ: Từ bảng Product thuộc bộ dữ liệu adventureworks, truy vấn Name, StandardCost và sử dụng các hàm làm tròn với StandardCost (ROUND, CEILING, FLOOR).

© 2023 Datapot. All right reserved.

```sql
USE adventureworks
SELECT
    Name,
    StandardCost,
    ROUND(StandardCost, 3) AS Rounded,
    CEILING(StandardCost) AS Rounded_up,
    FLOOR(StandardCost) AS Rounded_down
FROM SalesLT.Product
```

Results Messages

	Name	StandardCost	Rounded	Rounded_up	Rounded_down
1	HL Road Frame - Black, 58	1059,31	1059,31	1060,00	1059,00
2	HL Road Frame - Red, 58	1059,31	1059,31	1060,00	1059,00
3	Sport-100 Helmet, Red	13,0863	13,086	14,00	13,00
4	Sport-100 Helmet, Black	13,0863	13,086	14,00	13,00
5	Mountain Bike Socks, M	3,3963	3,396	4,00	3,00
6	Mountain Bike Socks, L	3,3963	3,396	4,00	3,00
7	Sport-100 Helmet, Blue	13,0863	13,086	14,00	13,00

Giải thích câu lệnh truy vấn:

- FROM: Dữ liệu truy vấn từ bảng SalesLT.Product.
- SELECT: Truy vấn các cột được chọn: Name, StandardCost, Rounded, Rounded_up, Rounded_down.

  Hàm ROUND: Làm tròn StandardCost tới chữ số thập phân thứ 3.

  Hàm CEILING: Làm tròn StandardCost lên.

  Hàm FLOOR: Làm tròn StandardCost xuống.

Trên đây chỉ là một số hàm, định dạng kiểu dữ liệu phổ biến, các bạn có thể tham khảo thêm các hàm, định dạng kiểu dữ liệu khác tại tài liệu của Microsoft.

Tài liệu tham khảo

- NULL là gì và cách xử lý trong SQL Server
- 3 lỗi thường gặp khi xử lý NULL trong SQL

© 2023 Datapot. All right reserved.