

5. SQL FUNCTION

Obyektif :

Setelah menyelesaikan bahasan ini, diharapkan dapat melakukan hal berikut:

1. Menyebutkan dan menjelaskan berbagai tipe function yang tersedia pada SQL
 2. Menggunakan character, number dan date function pada statement SELECT
 3. Menjelaskan kegunaan conversion function
-

5.1. Tipe SQL Function

Fungsi merupakan sebuah rutin yang melakukan operasi tertentu dan mengembalikan suatu hasil. Fungsi bisa menerima argumen yang akan digunakan di dalam proses. Ada 2 tipe berbeda pada fungsi SQL, yaitu :

- Fungsi satu baris / scalar (Single-Row Functions).
- Fungsi Group / Aggregate (Multiple-Row Function).

Single-Row Function

Fungsi ini mengembalikan hasil dari setiap baris di dalam bentuk table / view. **Single-Row Functions** menerima satu atau lebih argumen dan menghasilkan satu nilai tiap baris. Argumen dapat berupa :

- User-supplied constant
- Nilai variabel
- Nama kolom
- Ekspresi

Ada beberapa tipe yang berbeda dari single-row functions, yaitu :

1. Character
2. Number
3. Date
4. Conversion

Multiple-Row Functions

Mengembalikan hanya satu hasil ke dalam group baris. Fungsi ini menggunakan kelompok baris untuk memberi satu hasil setiap kelompok baris, dapat berupa nested.

Syntax :

Functions_name (*column / expression*, [arg1, arg2, ...])

Keterangan :

Functions_name : nama fungsi
Column : nama kolom database
Expression : untai karakter atau ekspresi yang dikalkulasi
arg1, arg2 : argumen yang akan digunakan oleh fungsi

5.1.1. Character Functions

Single-row character functions menerima data karakter sebagai masukan dan dapat menghasilkan nilai karakter dan bilangan. Fungsi character dapat dibagi menjadi :

- Case conversion function
- Character manipulation function

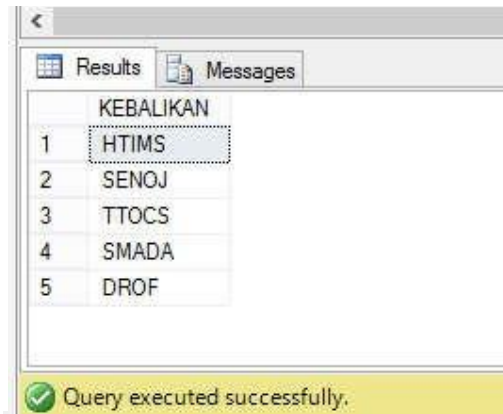
Fungsi Mengubah Bentuk Huruf (Case Conversion Function)

Beberapa case conversion functions, yaitu :

- LOWER (*char_expr*) : Mengubah untai karakter huruf menjadi huruf kecil.
- UPPER (*char_expr*) : Mengubah untai karakter huruf menjadi huruf besar.
- REVERSE (*char_expr*) : Menghasilkan karakter dalam bentuk terbalik.

Contoh :

```
➤ SELECT REVERSE (ENAME) AS "KEBALIKAN"  
FROM EMP  
WHERE DEPTNO=20
```



The screenshot shows a database query result window with a tab labeled 'Results'. The window displays a table with the following data:

| | KEBALIKAN |
|---|-----------|
| 1 | HTIMS |
| 2 | SENOJ |
| 3 | TTOCS |
| 4 | SMADA |
| 5 | DROF |

Below the table, a status bar indicates 'Query executed successfully.' with a green checkmark icon.

Operator Gabungan

Anda dapat menghubungkan / menggabungkan kolom dengan kolom lain, ekspresi aritmatika atau nilai konstanta untuk membentuk sebuah ekspresi karakter menggunakan operator gabungan (+). Kolom yang terletak setelah operator (+), digabungkan untuk membuat kolom tunggal.

Fungsi Manipulasi Karakter (Character Manipulation Function)

Beberapa character manipulation functions adalah :

- SUBSTRING : Mengambil string yang panjangnya ditentukan Bentuk penulisan untuk SUBSTR dan TRIM :

SUBSTRING (*nama_field* FROM *start_index* [FOR *length*]) □ LTRIM

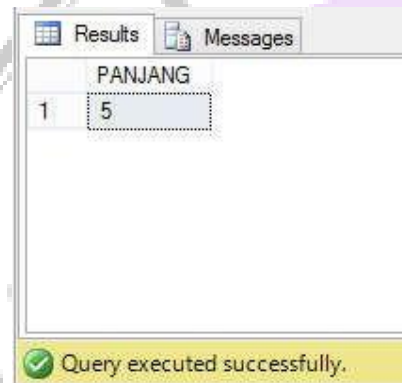
(*char_expr*) : Menghasilkan data tanpa didahului spasi kosong.

- RTRIM (*char_expr*) : Menghasilkan data tanpa diakhiri spasi kosong.
- LEN(*char_expr*) : Menunjukkan panjang dari string dengan angka.
- ASCII (*expression*) : Menghasilkan nilai ASCII dari ekspresi karakter.
- STR : Mengkonversi tipe data numeric ke karakter.
- CHAR (*expr_int*) : Menghasilkan karakter yang sesuai dengan nilai kode ASCII

- **DIFFERENCE** (char_expr) : Membandingkan dua string dan menguji kesamaan diantara keduanya pada skala 1 – 4, dimana 4 adalah pencocokan yang paling baik.
- **RIGTH** (char_expr, expr_int) : Menghasilkan bagian karakter dari kanan.
- **LEFT** ((char_expr, expr_int) : Menghasilkan bagian karakter dari kiri.
- **REPLICATE** (char_expr, expr_int) : Menghasilkan karakter sesuai dengan jumlah perkalian yang ditentukan.

Contoh

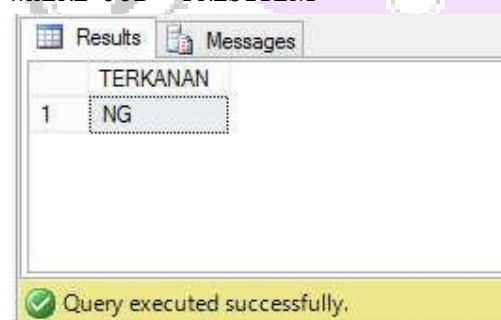
- `select LEN (ENAME) AS "PANJANG" FROM EMP WHERE EMPNO=7566`



| PANJANG | |
|---------|---|
| 1 | 5 |

Query executed successfully.

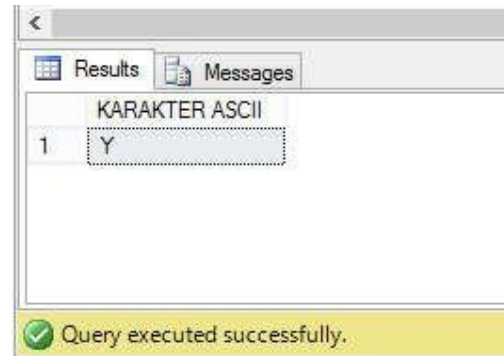
- `SELECT RIGHT (ENAME, 2) AS "TERKANAN"`
`FROM EMP`
`WHERE JOB='PRESIDENT'`



| TERKANAN | |
|----------|----|
| 1 | NG |

Query executed successfully.

➤ `SELECT CHAR(89) AS "KARAKTER ASCII"`



5.1.2. Number Functions

Fungsi number menerima masukan numeric dan menghasilkan nilai numeric.

Bentuk Umum :

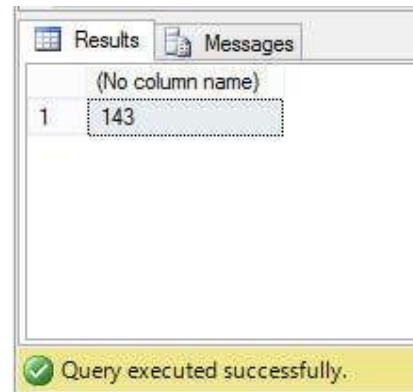
`SELECT nama_fungsi <parameter>`

Beberapa Fungsi Numerik :

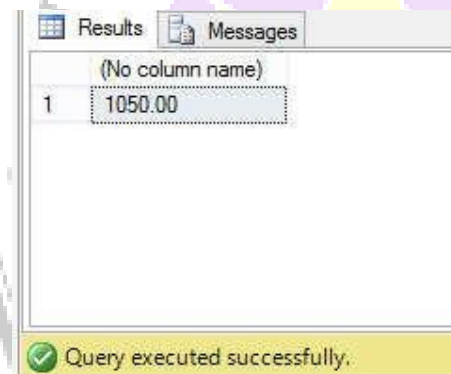
- **ABS** : menghasilkan nilai mutlak.
- **ROUND** : membulatkan ekspresi_numeric ke nilai presisi sesuai dengan eksponen_integer yang ditentukan.
- **CEILING** : menghasilkan integer terkecil yang lebih besar dari atau sama dengan nilai yang di tentukan.
- **EXP** : menghasilkan nilai eksponen dari nilai yang ditentukan.
- **FLOOR** : menghasilkan integer terbesar yang kuang dari atau sama dengan nilai yang ditentukan.
- **POWER** (ekspresi_numeric, pangkat) : menghasilkan nilai ekspresi_numeric ke pangkat.
- **RAND** (ekspresi_integer) : menghasilkan angka random bertipe data float antara 0 dan 1 dengan pilihan menggunakan eksponen_integer untuk seed.

CONTOH

- `SELECT CEILING (142.234)`



- `SELECT ABS ([SAL]-200)
FROM EMP
WHERE ENAME= 'WARD'`



5.1.3. Date Functions

SQL Server menyimpan tanggal pada format numeric internal, mewakili abad, tahun, bulan, tanggal, jam, menit dan detik.

Bentuk Umum Fungsi Tanggal :

`SELECT Fungsi_Tanggal (parameter)`

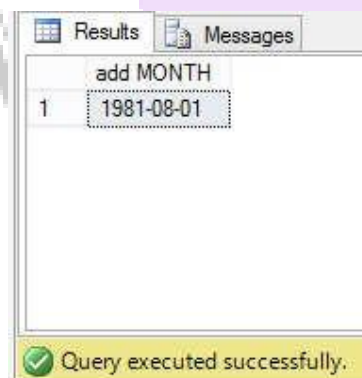
Beberapa Fungsi Tanggal SQL Server :

- a. DATEADD (*bagian_tgl, bil, tgl*) : Menambah bagian tanggal sesuai dengan besar bilangan ke dalam tanggal.
- b. DATEDIFF (*bagian_tgl, tgl1, tgl2*) : Menghasilkan nilai bagian tanggal diantara dua tanggal.
- c. DATENAME (*bagian_tgl, tgl*) : Menghasilkan character string dari bagian tanggal.
- d. DATEPART (*bagian_tgl, tgl*) : Menghasilkan nilai integer dari bagian tanggal.
- e. GETDATE () : Menghasilkan tanggal dan waktu sekarang.

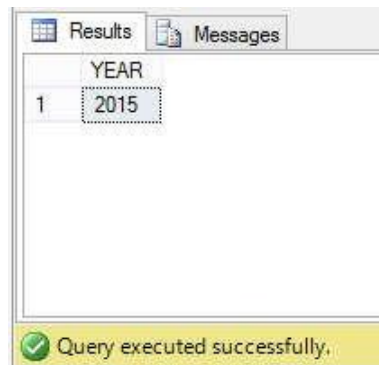
Bagian tanggal digunakan dengan fungsi tanggal untuk menentukan elemen nilai tanggal untuk aritmatika tanggal.

Contoh

- SELECT DATEDIFF (yy, ord_date, getdate()) FROM
- SELECT DATEADD (MONTH, 3, HIREDATE) AS "add MONTH" FROM EMP WHERE ENAME='BLAKE'



➤ `SELECT YEAR(getdate()) as "YEAR"`



| | YEAR |
|---|------|
| 1 | 2015 |

➤ `SELECT DATENAME (YY, GETDATE ()) AS "PART"`



| | part |
|---|------|
| 1 | 2015 |

Tabel. 5.1. Nilai Bagian Tanggal (datepart)

| Bagian_ Tanggal | Singkatan | Nilai |
|-----------------|-----------|---------------------|
| Tahun | yy | 1753 – 9999 |
| Quarter | Q | 1 – 4 |
| Month | mm | 1 – 12 |
| Day Of Year | dy | 1 – 366 |
| Day | dd | 1 – 31 |
| Week | wk | 1 – 54 |
| Week Day | dw | 1 – 7 (1 = Sunday) |
| Hour | hh | 0 – 23 |
| Minute | mi | 0 – 59 |
| Second | ss | 0 – 69 |
| Milisecond | ms | 0 – 999 |

5.1.4. Conversion Functions

Ada 2 Conversion Function, yaitu :

□ CAST □ CONVERT

Kedua function diatas mempunyai fungsi yang sama yaitu untuk mengkonversi nilai suatu field dari satu tipe menjadi tipe data yang lain.

Syntax :

CAST (expression **AS** *data_type*)

CONVERT (*data_type* [(length), expression [style]])

CAST merupakan SQL standard function yang dapat mengkonversi dari :

- Number To Character
- Character To Number
- Date To Character
- Character To Date

Contoh:

CONVERT (numeric, '10')

CONVERT (char, '10')

CONVERT (datetime, '04-JUL-97')

CONVERT (char, GETDATE())

CONVERT (char, GETDATE(), 'dd/mm/yy')

EXTRACT : mengekstrak tanggal, bulan atau tahun dari field dengan tipe data Date, Time / Time Stamp.

Syntax :

Extract (*Year / Month / Day* From *nama_tabel*) Contoh :

Select SaleDate,

Extract (Year From SaleDate) As Tahun

Extract (Month From SaleDate) As Bulan

Extract (Day From SaleDate) As Tanggal

From Orders