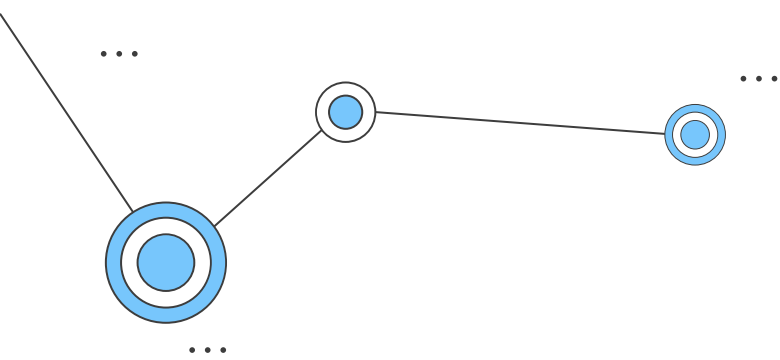


# BASIS DATA LANJUT

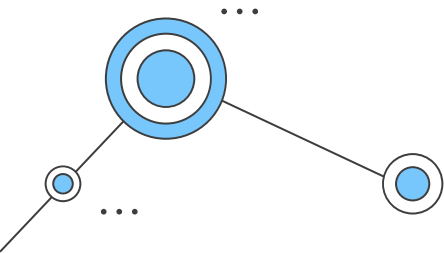
## Pertemuan 3

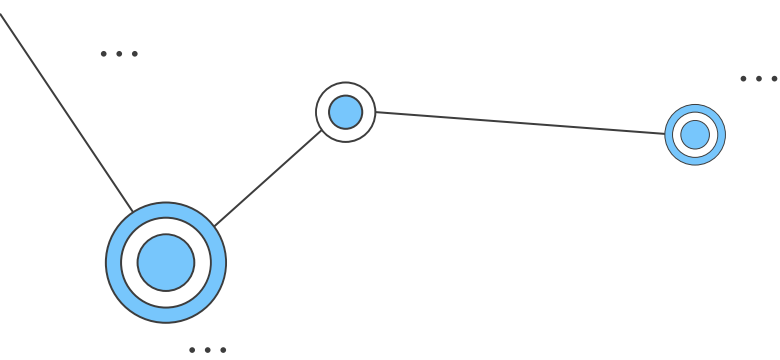
TIPE DATA, FUNGSI-FUNGSI BAWAAN,  
TABLE EXPRESSION



# OUTLINE

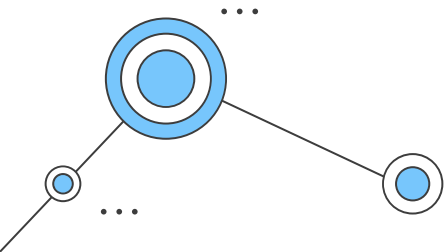
- Tipe Data
- Date & Time
- Fungsi Date & Time
- Data Karakter
- Fungsi Karakter
- View
- Common Table Expression (CTE)
- Table Valued Function (TVF)

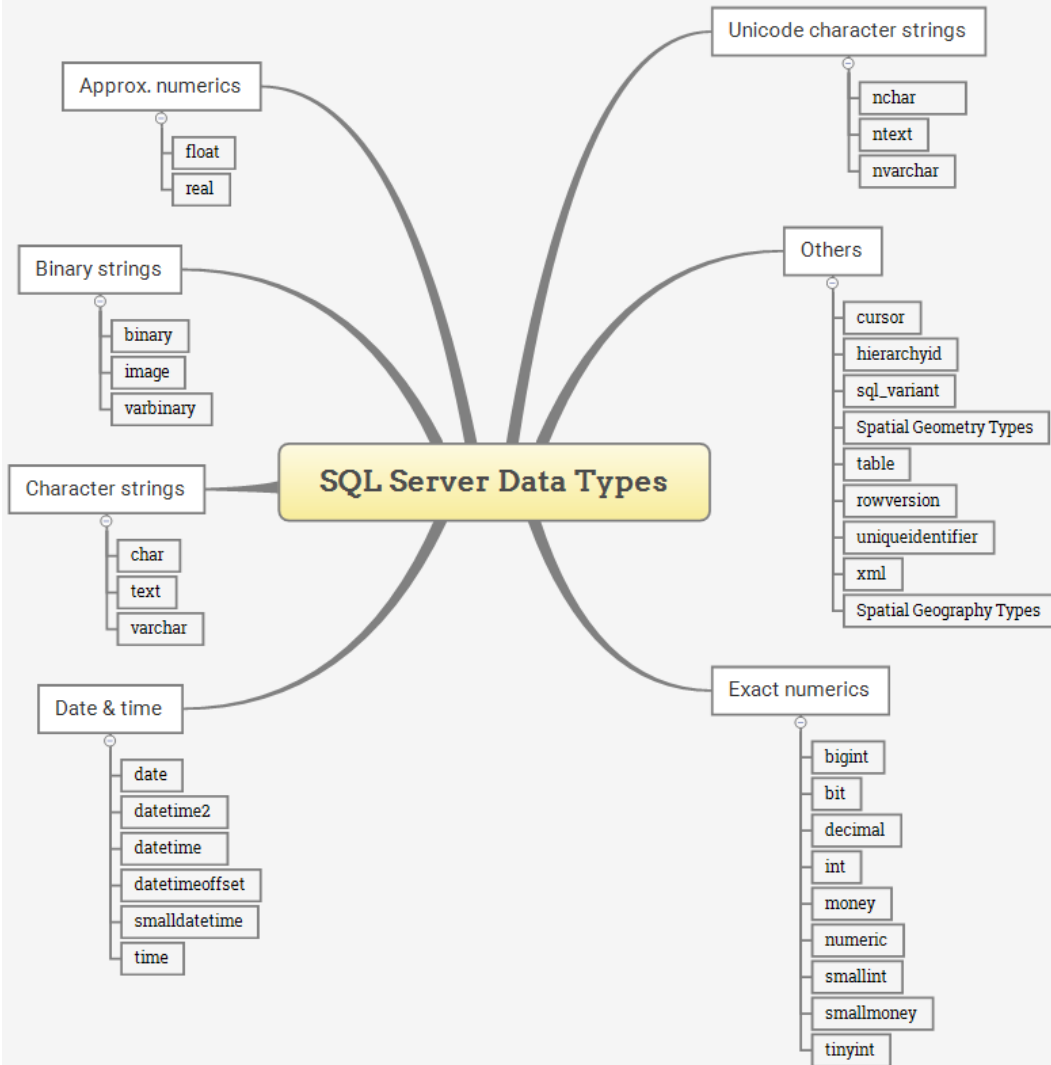




# Tipe Data

- Himpunan yang dapat ditemui pada data
- Dapat disebut sebagai kelompok data
- Mendefinisikan suatu kolom
- Menentukan batasan/kontrol terhadap data
- Menentukan memori yang digunakan





# TIPE DATA PADA SQL

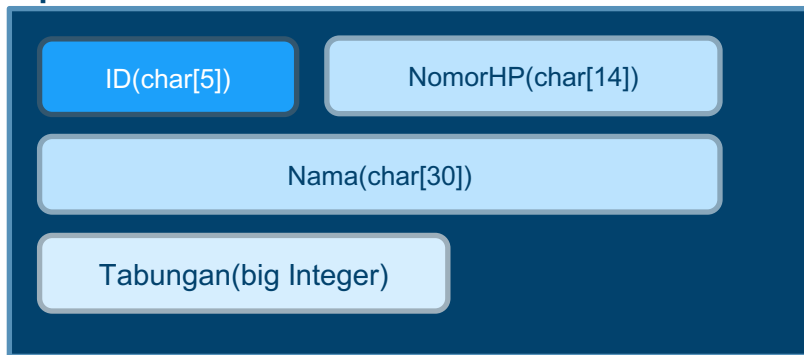
# EFISIENSI

**Tipe Data Sesuai = Efisien**  
**Tipe Data Tidak Sesuai = Tidak Efisien**

- Ketika membuat sebuah kolom dalam tabel, tipe data yang sesuai akan menghemat space storage
- Selain menghemat storage, juga akan membuat database lebih reliable.
- Tipe data yang sesuai juga akan lebih memastikan data yang masuk sesuai standar.
- Contoh :
  - Untuk membuat kolom Nomer Handphone maka hanya diperlukan paling banyak 15 karakter
  - Untuk menyimpan tanggal, gunakan tipe data datetime, sehingga format penanggalan akan seragam.

# ILUSTRASI

## Tipe Data Sesuai



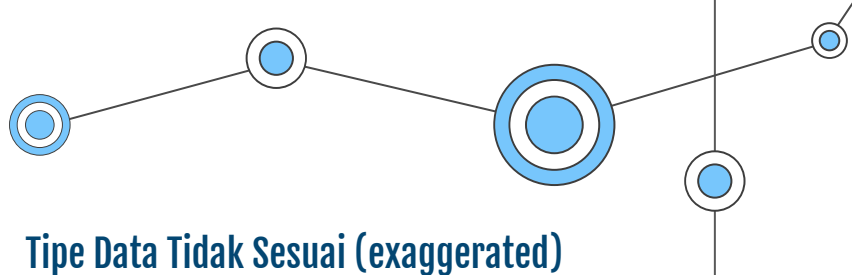
**1 record data**

**Total :**

**char = 5 + 14 + 30 = 49 byte**

**Big integer = 8 byte**

**TOTAL 1 record = 57 byte**



## Tipe Data Tidak Sesuai (exaggerated)

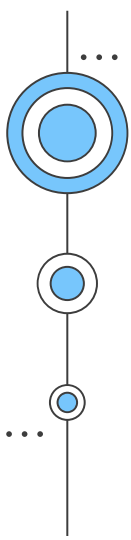
**1 record data**

**Total :**

**Char = 20 + 50 + 100 = 170byte**

**Decimal = 17 byte**

**TOTAL 1 record = 187 byte**



# Tipe data NUMERIC



# TIPE DATA NUMERIK

- Numerik adalah semua hal yang berbentuk **bilangan** dan **angka**.
- Data numerik ini dapat berupa minus (-) , positif ataupun bilangan bulat dan pecahan.
- Untuk menyimpan data dalam bentuk bilangan bulat negatif ataupun positif dapat menggunakan **INT**.

Data type	Range	Storage
<b>BIGINT</b>	$-2^{63}$ (-9,223,372,036,854,775,808) to $2^{63}-1$ (9,223,372,036,854,775,807)	8 Bytes
<b>INT</b>	$-2^{31}$ (-2,147,483,648) to $2^{31}-1$ (2,147,483,647)	4 Bytes
<b>SMALLINT</b>	$-2^{15}$ (-32,768) to $2^{15}-1$ (32,767)	2 Bytes
<b>TINYINT</b>	0 to 255	1 Byte



# TIPE DATA NUMERIK

- Sedangkan tipe data untuk menyimpan bilangan pecahan adalah sebagai berikut.

Type	Size
FLOAT	4 byte
DOUBLE	8 byte
DOUBLEPRECISION	8 byte
REAL	8 byte
DECIMAL (M,D)	M byte
NUMERIC (M,D)	M byte

## Contoh:

```
CREATE TABLE account (  
  accountNo integer,  
  balance numeric(8,2)  
);
```

The "balance" column can safely store the number 173226.62.

1 7 3 2 2 6 . 6 2

└──────────┘  
P

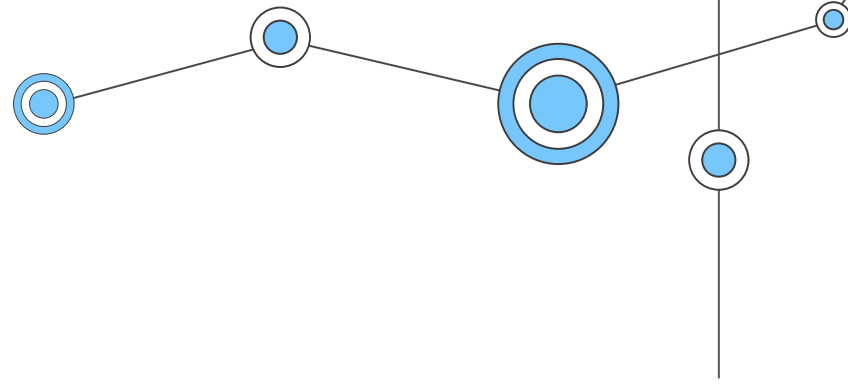
└──┘  
S

# Tipe data DATE & TIME

# TIPE DATA DATE & TIME

Data type	Format	Range	Accuracy	Storage size (bytes)	User-defined fractional second precision	Time zone offset
time	hh:mm:ss[.nnnnnnn]	00:00:00.0000000 through 23:59:59.9999999	100 nanoseconds	3 to 5	Yes	No
date	YYYY-MM-DD	0001-01-01 through 9999-12-31	1 day	3	No	No
smalldatetime	YYYY-MM-DD hh:mm:ss	1900-01-01 through 2079-06-06	1 minute	4	No	No
datetime	YYYY-MM-DD hh:mm:ss[.nnn]	1753-01-01 through 9999-12-31	0.00333 second	8	No	No
datetime2	YYYY-MM-DD hh:mm:ss[.nnnnnnn]	0001-01-01 00:00:00.0000000 through 9999-12-31 23:59:59.9999999	100 nanoseconds	6 to 8	Yes	No
datetimeoffset	YYYY-MM-DD hh:mm:ss[.nnnnnnn] [+ -]hh:mm	0001-01-01 00:00:00.0000000 through 9999-12-31 23:59:59.9999999 (in UTC)	100 nanoseconds	8 to 10	Yes	Yes

<https://docs.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-ver16>



# CONTOH QUERY DENGAN DATE

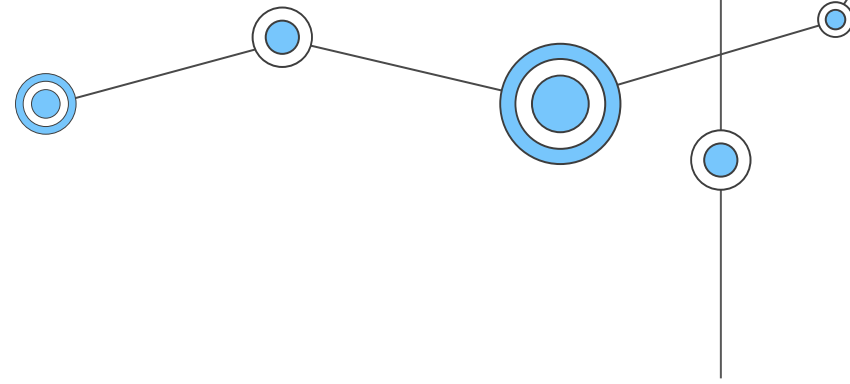
```
SELECT orderid, custid, empid, orderdate  
FROM Sales.Orders  
WHERE orderdate = '20070825';
```

	orderid	custid	empid	orderdate
1	10643	1	6	2007-08-25 00:00:00.000
2	10644	88	3	2007-08-25 00:00:00.000

DESKTOP-TDE45PK (14.0 RTM) | DESKTOP-TDE45PK\asus (54) | TSQL | 00:00:00 | 2 rows



# CONTOH QUERY DENGAN DATE



```
SELECT lastname, birthdate
FROM HR.Employees
WHERE
YEAR(birthdate) = 1973;
```

Results		Messages
	lastname	birthdate
1	Lew	1973-08-30 00:00:00.000
2	Suurs	1973-07-02 00:00:00.000

✓ Query executed successfully. | DESKTOP-EIPTP8V (15.0 RTM) | DESKTOP-EIPTP8V\milyu ... | TSQL | 00:00:00 | 2 rows





# FUNGSI DATE & TIME — `current_timestamp`

```
SELECT CURRENT_TIMESTAMP
```

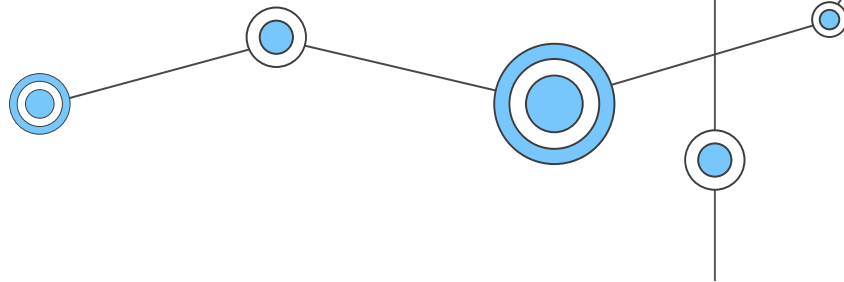
Results		Messages
	(No column name)	
1	2022-09-10 05:46:43.380	

✓ Query executed successfully. | DESKTOP-EIPTP8V (15.0 RTM)

- Mengembalikan nilai berupa tanggal dan waktu saat ini
- Merupakan fungsi SQL ANSI

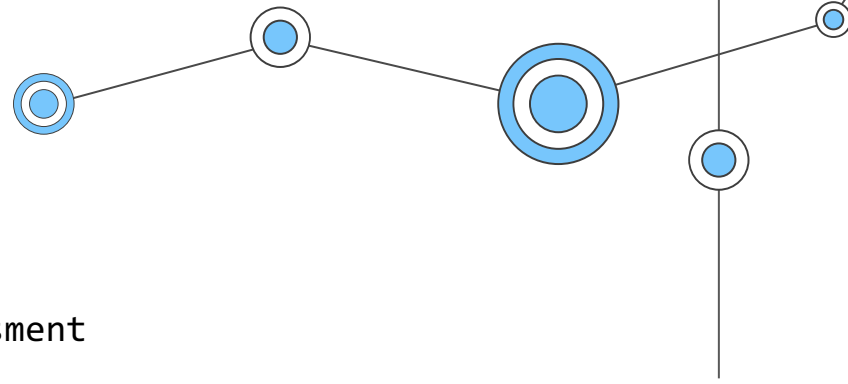
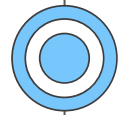


# FUNGSI DATE & TIME



Fungsi	Deskripsi
GETDATE()	Mengembalikan tanggal dan waktu
DATEPART()	Mengembalikan satu bagian dari tanggal / waktu
DATEADD()	Menambahkan atau mengurangi interval waktu yang ditentukan dari tanggal
DATEDIFF()	Mengembalikan waktu antara dua tanggal
CONVERT()	tanggal Menampilkan / data time dalam format yang berbeda





# FUNGSI DATE & TIME — dateadd()

SELECT

lastname, hiredate,

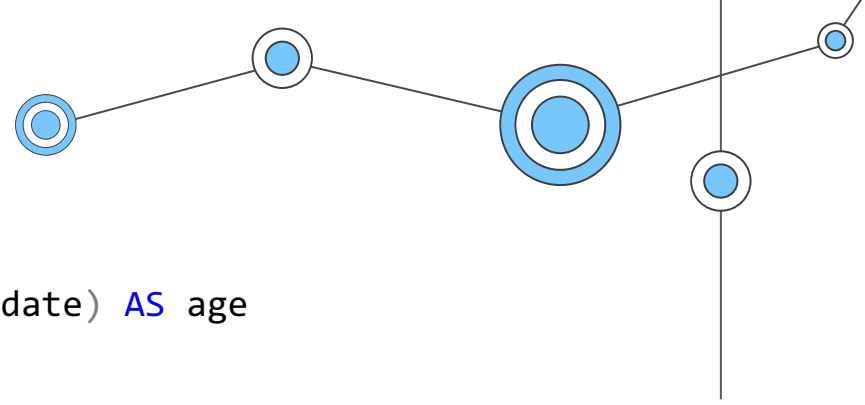
DATEADD(year, 1, hiredate) AS assesment

FROM HR.Employees

Results		Messages	
	lastname	hiredate	assesment
1	Davis	2002-05-01 00:00:00.000	2003-05-01 00:00:00.000
2	Funk	2002-08-14 00:00:00.000	2003-08-14 00:00:00.000
3	Lew	2002-04-01 00:00:00.000	2003-04-01 00:00:00.000
4	Peled	2003-05-03 00:00:00.000	2004-05-03 00:00:00.000
5	Buck	2003-10-17 00:00:00.000	2004-10-17 00:00:00.000
6	Suurs	2003-10-17 00:00:00.000	2004-10-17 00:00:00.000
7	King	2004-01-02 00:00:00.000	2005-01-02 00:00:00.000
8	Cameron	2004-03-05 00:00:00.000	2005-03-05 00:00:00.000
9	Dolgopyatova	2004-11-15 00:00:00.000	2005-11-15 00:00:00.000

SKTOP-TDE45PK (14.0 RTM) | DESKTOP-TDE45PK\asus (54) | TSQL | 00:00:00 | 9 rows





# FUNGSI DATE & TIME — datediff()

```
SELECT lastname,  
DATEDIFF(year, YEAR(CURRENT_TIMESTAMP), birthdate) AS age  
FROM HR.Employees
```

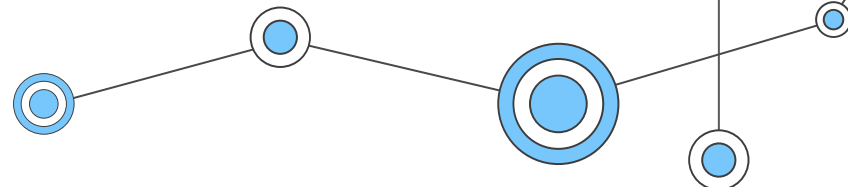
Results			Messages		
	lastname	age			
1	Davis	53			
2	Funk	57			
3	Lew	68			
4	Peled	42			
5	Buck	60			
6	Suurs	68			
7	King	65			
8	Cameron	63			
9	Dolgopyatova	71			

✓ Query executed successfully.

# CAST & CONVERT



# CAST & CONVERT



- CAST dan CONVERT adalah fitur dari server SQL yang diperlukan untuk konversi ekspresi dari satu jenis tipe data ke tipe data lainnya.
- CAST disarankan untuk konversi dasar. CONVERT disarankan untuk rutinitas spesifik datetime.
- CAST mempunyai waktu loading data yang lebih cepat dibandingkan dengan menggunakan CONVERT.





# CAST

- Digunakan pada klausa SELECT dan WHERE
- Merupakan fungsi standar ANSI
- Syntax:  
`CAST(<value> AS <datatype>)`
- Contoh:  
`SELECT CAST(25.65 AS int);`



# CONVERT

- Digunakan pada klausa SELECT dan WHERE
- Untuk konversi:
  - Date
  - Time
  - Numeric
  - XML
- Merupakan fungsi bawaan SQL SERVER
- Syntax:

**CONVERT(<datatype>,<value>,<optional\_style\_number>)**

- Syntax:

```
SELECT CONVERT(int, 25.65);
```

# KONVERSI FORMAT DATE

Without century	With century	Input/Output	Standard
0	100	mon dd yyyy hh:miAM/PM	Default
1	101	mm/dd/yyyy	US
2	102	yyyy.mm.dd	ANSI
3	103	dd/mm/yyyy	British/French
4	104	dd.mm.yyyy	German
5	105	dd-mm-yyyy	Italian
6	106	dd mon yyyy	-
7	107	Mon dd, yyyy	-
8	108	hh:mm:ss	-
9	109	mon dd yyyy hh:mi:ss:mmmAM (or PM)	Default + millisec

[https://www.w3schools.com/sql/func\\_sqlserver\\_convert.asp](https://www.w3schools.com/sql/func_sqlserver_convert.asp)

# CAST & CONVERT

- Konversi tipe data secara eksplisit

```
SELECT CAST('20220910' AS DATE)
```

```
SELECT CONVERT(DATETIME, '09/22/2022')
```

Results		Messages
	(No column name)	
1	2022-09-10	
	(No column name)	
1	2022-09-22 00:00:00.000	
✓ Query executed successfully.		DESKTOP-EIPTP8V (15.0 RTM)

# TIPE DATA KARAKTER



# TIPE DATA KARAKTER

Data type	Description	Max size	Storage
char(n)	Fixed width character string	8,000 characters	Defined width
varchar(n)	Variable width character string	8,000 characters	2 bytes + number of chars
varchar(max)	Variable width character string	1,073,741,824 characters	2 bytes + number of chars
text	Variable width character string	2GB of text data	4 bytes + number of chars
nchar	Fixed width Unicode string	4,000 characters	Defined width x 2
nvarchar	Variable width Unicode string	4,000 characters	
nvarchar(max)	Variable width Unicode string	536,870,912 characters	
ntext	Variable width Unicode string	2GB of text data	
binary(n)	Fixed width binary string	8,000 bytes	
varbinary	Variable width binary string	8,000 bytes	
varbinary(max)	Variable width binary string	2GB	
image	Variable width binary string	2GB	

# CONCATENATION

Menggabungkan karakter

- Menggunakan tanda +

```
SELECT empid, firstname + N ' ' + lastname AS FullName  
FROM HR.employees;
```

- Menggunakan CONCAT()

```
SELECT empid,  
       CONCAT (firstname, ' ', lastname) AS FullName  
FROM HR.employees;
```

Results			Messages		
	empid	FullName			
1	1	Sara Davis			
2	2	Don Funk			
3	3	Judy Lew			
4	4	Yael Peled			

# FUNGSI STRING

- **SUBSTRING()**

Mengembalikan sebagian string berdasarkan *starting point* dan jumlah karakter yang dikembalikan

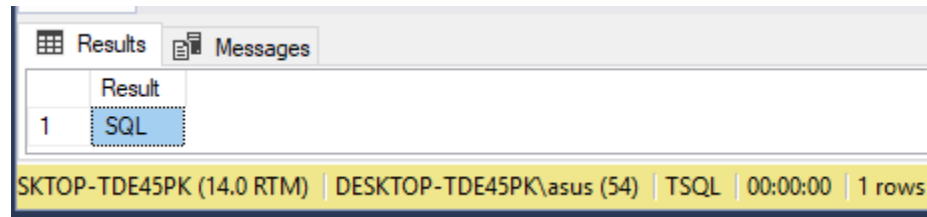
SUBSTRING(EXPRESSION, START, LENGTH)

Contoh:

SELECT

SUBSTRING ('Microsoft SQL SERVER', 11,3)

AS Result;



Results		Messages
	Result	
1	SQL	

SKTOP-TDE45PK (14.0 RTM) | DESKTOP-TDE45PK\asus (54) | TSQL | 00:00:00 | 1 rows

# FUNGSI STRING

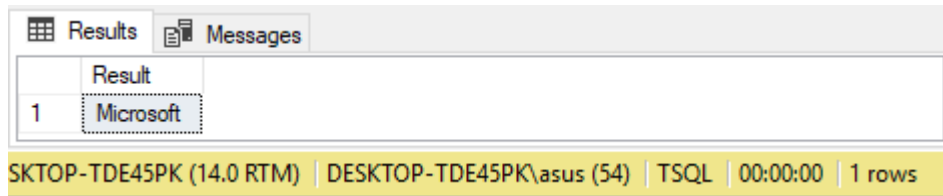
- **LEFT(), RIGHT()**

Mengembalikan sebagian string dalam jumlah tertentu, dimulai dari karakter string paling kiri/kanan

```
LEFT(EXPRESSION, integer_value)
```

Contoh:

```
SELECT  
LEFT ('Microsoft SQL SERVER', 9)  
AS Result;
```



Result
1 Microsoft

SKTOP-TDE45PK (14.0 RTM) | DESKTOP-TDE45PK\asus (54) | TSQL | 00:00:00 | 1 rows

# FUNGSI STRING

- **LEN(), DATALENGTH()**

Mengembalikan jumlah karakter pada string tertentu

LEN(String), DATALENGTH(String)

Contoh:

```
SELECT LEN('Microsoft SQL SERVER')AS Result;
```

```
SELECT DATALENGTH('Microsoft SQL SERVER')AS Result;
```

Results		Messages
	Result	
1	20	

SKTOP-TDE45PK (14.0 RTM) | DESKTOP-TDE45PK\asus (54) | TSQL | 00:00:00 | 1 rows

# LIKE

- Digunakan untuk membandingkan data berdasarkan pola tertentu
- Digunakan pada statemen WHERE
- Simbol % merepresentasikan string
- contoh.

```
SELECT categoryid, description
FROM Production.Categories
WHERE description LIKE '%ee%';
```

Results		Messages
	categoryid	description
1	1	Soft drinks, coffees, teas, beers, and ales
2	2	Sweet and savory sauces, relishes, spreads, and ...
3	3	Desserts, candies, and sweet breads
4	4	Cheeses
5	8	Seaweed and fish

DESKTOP-TDE45PK (14.0 RTM) | DESKTOP-TDE45PK\asus (54) | TSQL | 00:00:00 | 5 rows

# NULL

- NULL merepresentasikan nilai yang kosong
- Perlakuan ANSI untuk NULL values:

Hasil semua ekspresi yang bernilai NULL adalah NULL

- $2 + \text{NULL} = \text{NULL}$
- `'MyString: ' + NULL = NULL`
- Perbandingan nilai

`NULL = NULL` returns *false*

`NULL IS NULL` returns *true*

# NULL

- ISNULL menggantikan nilai NULL dengan suatu nilai

ISNULL elemen	Deskripsi
Ekspresi/ kolom yang ingin di cek	Mengembalikan ekspresi tersebut jika tidak bernilai NULL
Nilai pengganti	Menjadi kembalian jika ekspresi/kolom bernilai NULL



# NULL

```
Select custid, city, region, country  
FROM Sales.Customers;
```

	custid	city	region	country
1	1	Berlin	NULL	Germany
2	2	México D.F.	NULL	Mexico
3	3	México D.F.	NULL	Mexico
4	4	London	NULL	UK
5	5	Luleå	NULL	Sweden
6	6	Mannheim	NULL	Germany
7	7	Strasbourg	NULL	France
8	8	Madrid	NULL	Spain
9	9	Marseille	NULL	France
10	10	Tsawassen	BC	Canada
11	11	London	NULL	UK
12	12	Buenos Aires	NULL	Argentina

```
Select custid, city, ISNULL(region, 'N/A') AS region, country  
FROM Sales.Customers;
```

	custid	city	region	country
1	1	Berlin	N/A	Germany
2	2	México D.F.	N/A	Mexico
3	3	México D.F.	N/A	Mexico
4	4	London	N/A	UK
5	5	Luleå	N/A	Sweden
6	6	Mannheim	N/A	Germany
7	7	Strasbourg	N/A	France
8	8	Madrid	N/A	Spain
9	9	Marseille	N/A	France
10	10	Tsawassen	BC	Canada
11	11	London	N/A	UK
12	12	Buenos Aires	N/A	Argentina

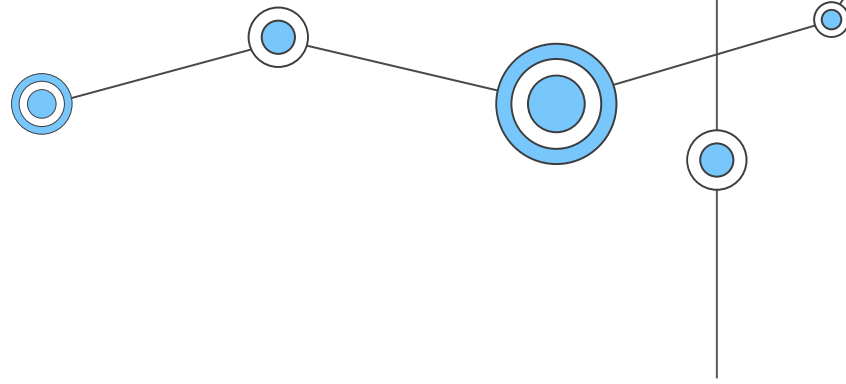
# LOGICAL FUNCTION



# LOGICAL FUNCTION

- Seleksi kondisi dengan IIF
- Memilih Item pada List dengan CHOOSE
- Fungsi logika baru pada SQL Server 2012
- Seperti ekspresi CASE dengan 2 kemungkinan kembalian
- Syntax:

```
SELECT IIF(<Boolean expression>,<value_if_TRUE>,  
<value_if_FALSE_or_UNKNOWN>;
```



# LOGICAL FUNCTION

- Contoh

```
SELECT productid, unitprice,  
IIF(unitprice > 50, 'high','low') AS pricepoint  
FROM Production.Products;
```

productid	unitprice	pricepoint
7	30.00	low
8	40.00	low
9	97.00	high
17	39.00	low
18	62.50	high

# Memilih Item pada List dengan CHOOSE

- CHOOSE mengembalikan sebuah item pada list yang dipilih berdasarkan nilai index
- Syntax:  
**SELECT CHOOSE(<index\_value>,<item1>, <item2>[,...]);**

Elemen CHOOSE	Deskripsi
Index_value	Nilai integer yang merepresentasikan posisi item pada list
Value_List (<item1>, <item2>,...)	List item dengan tipe data apa saja yang akan dikembalikan nilainya

# Memilih Item pada List dengan CHOOSE

```
SELECT CHOOSE (3, 'Beverages', 'Condiments', 'Confections') AS  
choose_result;
```

```
choose_result  
-----  
Confections
```

# Thanks!

Do you have any questions?



Team Teaching Matakuliah Basis Data Lanjut  
JTI POLINEMA