

# Programming Database

## "Overview Select Statement"

---

### Objective:

1. Memahami Konsep perintah Select dalam Transaction (T-SQL) di SQL Server.
2. Manipulasi jumlah result set query dari total result set baris
3. Memahami konsep enquiry single dan multi table (Join).

### Prepare:

1. Download file **structureanddata.sql** yang disediakan di folder kerja anda.
2. Eksekusi file untuk Instalasi table, view yang akan digunakan selama praktikum

### Review and Case Study:

Bagian ini akan memberikan studi kasus dan review mengenai perintah yang terdapat dalam studi kasus yang diberikan.

#### 1. Enquiring data from single-table

Melakukan enquiry data pada satu source data (table) dan menampilkan beberapa/ seluruh informasi (atribut/ kolom), kita harus menggunakan perintah select untuk menampilkan kolom beserta dengan data yang terdapat didalamnya, dan mengkombinasikan dengan perintah from untuk merujuk sumber data yang digunakan.

##### Case Study:

Tampilkan informasi nama kontak (contactname) dan alamat (address) customer yang telah terdaftar?

##### Answer:

```
SELECT contactname, address  
FROM sales.Customers
```

##### Note:

Hindarilah menggunakan symbol \* untuk menampilkan informasi karena akan beban kerja enquiry membekak, dan gunakan pemanggilan nama kolom sesuai kebutuhan bisa mengoptimalkan perintah query anda.

Untuk melakukan penyaringan data, anda bisa menggunakan perintah WHERE dengan menggunakan kondisi tunggal atau jamak.

##### Case Study:

Tampilkan informasi nama kontak (contactname) dan alamat (address) customer yang tinggal di negara prancis?

##### Answer:

```
SELECT contactname, address  
FROM sales.Customers  
WHERE country = 'France'
```

##### Case Study:

Tampilkan informasi nama kontak (contactname) dan alamat (address) customer yang tinggal di kota Paris dan Nates di Negara prancis ?

##### Answer:

```
SELECT contactname, address  
FROM sales.Customers  
WHERE country = 'France' AND (city = 'paris' OR city = 'Nantes')
```

# Programming Database

## "Overview Select Statement"

Untuk melakukan pengelompokan data yang dengan melakukan agregat (fungsi yang menerima koleksi nilai dan mengembalikan nilai tunggal sebagai hasilnya).

Standar ISO untuk fungsi agregat:

SUM	Mengembalikan nilai total dari seluruh nilai pada kolom
AVG	Mengembalikan nilai rata-rata dari seluruh nilai kolom(total nilai / banyak data nilai).
COUNT	Mengembalikan banyak nilai dari seluruh nilai di sebuah kolom.
MAX	Mengembalikan nilai tertinggi yang terdapat didalam sebuah kolom.
MIN	Mengembalikan nilai terendah yang terdapat didalam sebuah kolom.

**Case Study:**

Tampilkan informasi jumlah pelanggan berdasarkan kota yang di tinggal oleh pelanggan di Negara Prancis ?

**Answer:**

```
SELECT city, COUNT(custid) AS countcust
FROM sales.Customers
WHERE country = 'France' AND (city = 'Paris' OR city = 'Nantes')
GROUP BY country, city
```

saat kita akan melakukan penyaringan data dengan menggunakan fungsi agregat, maka kita tidak bisa menggunakan perintah WHERE, karena perintah tersebut tidak bisa mendukung fungsi pencarian data dengan kombinasi fungsi agregat dan penyaringan dengan kombinasi fungsi agregat ialah perintah HAVING.

**Case Study:**

Tampilkan informasi jumlah pelanggan berdasarkan kota yang di tinggal oleh pelanggan di Negara Prancis dan yang jumlah pelanggannya diatas 1 pelanggan ?

**Answer:**

```
SELECT city, COUNT(custid) AS countcust
FROM sales.Customers
WHERE country = 'France' AND (city = 'Paris' OR city = 'Nantes')
GROUP BY country, city
HAVING COUNT(custid) > 1
```

Untuk melakukan membuat pengurutan informasi yang ditampilkan sesuai dengan kebutuhan, kita dapat menggunakan perintah ORDER BY dan dua pola pengurutan yang bisa digunakan oleh anda, yakni pengurutan secara ascending (kecil ke besar/terendah ke tertinggi) dan descending (besar ke kecil/ tertinggi ke terendah).

**Case Study:**

Tampilkan informasi nama kontak (contactname), kota (city) dan negara (country) customer yang telah terdaftar dan Tampilkan informasi dengan urutan city dari abjak tertinggi ke terendah dan negara dari abjak terkecil ke terbesar?

**Answer:**

```
SELECT contactname, city, country
FROM sales.Customers
WHERE city DESC, country
```

**Note:**

Dalam pengurutan secara ascending tidak perlu penulisan secara eksplisit secara default system akan melakukan pengurutan secara ascending untuk kolom yang ditunjuk dalam pengurutan tersebut.

# Programming Database

## "Overview Select Statement"

Dalam melakukan penyaringan jumlah result set baris yang didapatkan dalam proses enquiring data, kita bisa menggunakan beberapa fungsional yang disediakan oleh SQL Server, yakni menggunakan perintah **TOP** dan **OFFSET-FETCH**.

### Case Study:

Tampilkan informasi nama kontak (contactname) dan alamat (address) customer yang telah terdaftar dengan jumlah baris 10 teratas dari total result set?

### Answer:

```
SELECT TOP(5) contactname, address
FROM sales.Customers
WHERE country = 'France' OR country = 'Germany'
```

### Note:

Dalam penyaringan jumlah result set yang dikembalikan bisa menggunakan keyword **PERCENT** untuk penyaringan data berdasarkan Percentase dari total yang dikembalikan.

### Case Study:

Tampilkan informasi nama kontak (contactname) dan alamat (address) customer yang telah terdaftar dengan jumlah 10 persen baris teratas dari total result set?

### Answer:

```
SELECT TOP(10) PERCENT contactname, address
FROM sales.Customers
WHERE country = 'France' OR country = 'Germany'
```

### Case Study:

Tampilkan informasi nama kontak (contactname) dan alamat (address) customer yang telah terdaftar dimulai dari jumlah baris ke 4 s/d 7 dari total result set?

### Answer:

```
SELECT contactname, address
FROM sales.Customers
WHERE country = 'France' OR country = 'Germany'
ORDER BY contactname
OFFSET 3 ROWS FETCH NEXT 3 ROWS ONLY
```

### Note:

Saat menggunakan perintah **OFFSET-FETCH** harus diikutsertakan perintah **ORDER BY** untuk pengurutan untuk dapat menentukan nomor urut dan memudahkan dalam pengambilan informasi berdasarkan nomor baris yang terdapat dalam total result set.

Dalam memunculkan sebuah value dalam perintah **SELECT** kita bisa ubah dengan kondisional secara logic yang ditetapkan oleh anda, **CASE** bukanlah sebuah perintah seperti halnya **SELECT**, dan lainnya, melainkan sebuah ekspresi scalar sehingga ekspresi ini bisa diterapkan dalam perintah **SELECT**, **WHERE**, **HAVING** dan **CHECK Constraint** (selain **SELECT** akan dibahas di lain sesi). Cara kerja dari ekspresi ini sangatlah sederhana dengan membandingkan nilai yang nanti akan dicek terhadap kondisi yang telah ditentukan dan memiliki dua result value yakni: **true** dan **false**, setiap result value **true**, nilai akan dilakukan perubahan sesuai dengan result value yang ditentukan sebagai pengganti result set yang dibandingkan.

### Case Study:

Memunculkan data produk pada table **production.product** dengan menggunakan ekspresi **case** untuk memunculkan kategori produk berdasarkan nilai di kolom **cateoriesid**

# Programming Database

## "Overview Select Statement"

Answer:

```
SELECT productid, productname, categoryid,  
CASE categoryid  
  WHEN 1 THEN 'Beverages'  
  WHEN 2 THEN 'Condiments'  
  WHEN 3 THEN 'Confections'  
  WHEN 4 THEN 'Dairy Products'  
  WHEN 5 THEN 'Grains/Cereals'  
  WHEN 6 THEN 'Meat/Poultry'  
  WHEN 7 THEN 'Produce'  
  WHEN 8 THEN 'Seafood'  
  ELSE 'Unknown Category'  
END AS categoryname  
FROM Production.Products;
```

### 2. Enquiring data from Multi-table (JOIN)

Dalam enquiring data kita bisa saja menggunakan lebih dari satu data source, untuk enquiring dengan kondisi tersebut kita bisa menggunakan sebuah perintah yang dikenal dengan JOIN, dalam join anda bisa menggunakan tiga konsep yakni

Standar ISO untuk Join:

Cross	Melakukan penggabungan data dengan melakukan perkalian jumlah baris diantara dua data source atau lebih
INNER	Melakukan penggabungan data dengan melakukan filter data yang sama diantara dua data source atau lebih setelah dilakukan perkalian jumlah bais dalam kedua data source
Outer	Melakukan penggabungan data dengan melakukan filter data mengacu pada pada data source sebelah kiri(LEFT) atau sebelah kanan(RIGHT) diantara dua data source atau lebih setelah dilakukan perkalian jumlah bais dalam kedua data source

Case Study:

Memunculkan data produk pada table production.product dan nama kategori yang terdapat pada table production.categories untuk setiap produknya?

Answer:

```
SELECT prd.productid, prd.productname, prd.categoryid, cgs.categoryname  
FROM Production.Products AS Prd INNER JOIN Production.Categories AS cgs  
  ON Prd.categoryid = cgs.categoryid
```

Case Study:

Memunculkan nama customer dan perusahaan customer yang pernah dan belum pernah melakukan proses order yang mengacu pada sales.customers dan sales.orders ?

Answer:

```
SELECT c.contactname, c.companyname, o.orderid, o.orderdate, o.requireddate,  
o.shippeddate  
FROM Sales.Customers AS C LEFT OUTER JOIN Sales.Orders AS O  
  ON C.custid = O.custid;
```

# Programming Database

“Overview Select Statement”

---

Exercisel!

**Case Study:**

Kembali tiga negara pengapalan dengan rata-rata pengiriman tertinggi pada tahun 2007

Table yang digunakan : sales.orders table

Output yang diminta:

<u>shipcountry</u>	<u>avgfreight</u>
Austria	178.3642
Switzerland	117.1775
Sweden	105.16

Selection Data

