



### Topik

Pengenalan SQL SERVER dan perintah-perintah DDL

### Tujuan

Mahasiswa diharapkan dapat:

1. Menjalankan SQL (Data Definition Language (DDL)) berbasis console / SQL Text Editor
2. Memahami perintah-perintah SQL (Data Definition Language (DDL))
3. Memahami penggunaan dan pembuatan constraint pada tabel

### Tools yang digunakan

SQL Server Management Studio SQL Query Editor

### Pendahuluan

DDL atau Data Definition Language adalah bagian dari SQL yang digunakan untuk mendefinisikan data dan objek database. Didalam kategori ini terdapat perintah-perintah sebagai berikut:

Perintah	Keterangan
CREATE DATABASE	Membuat Database
CREATE TABLE	Membuat tabel
ALTER TABLE	Mengubah atau menyisipkan kolom ke dalam tabel
DROP TABLE	Menghapus tabel dari database

Ada beberapa tipe data yang sering digunakan pada MS. SQL Server :

Data Type	Category	Keterangan
Varchar [(n)]	Character Strings	Karakter data type yang memiliki ukuran panjang yang tetap. Maksimum 8000 karakter.
Char [(n)]		Sama seperti varchar hanya panjang ukuran konsisten.
Int	Exact Numerics	Tipe Number bilangan bulat Rangennya -2,147,483,648 s/d 2,147,483,647 (storage: 4 bytes)
Smallint		Tipe Number bilangan bulat Rangennya -32,768 s/d 32,767 (storage: 2 bytes)
Tinyint		Tipe Number bilangan bulat Rangennya 0 s/d 255 (storage: 1 bytes)
Decimal(p,[s])		Tipe data numerik yang memiliki batasan dan desimal.
Numeric(p,[s])		Equivalent dengan Decimal
Money		Tipe data yang merepresentasikan currency value

		...
SmallMoney		Sama seperti Money, hanya saja range nya lebih kecil
Date	Date And Time	Define tanggal, default literal format YYYY-MM-DD
DateTime		Define kombinasi tanggal dan waktu dengan base 24 jam

## **CREATE**

Instalasi sistem manajemen (DBMS) pada komputer memungkinkan anda untuk membuat dan mengelola banyak database independent. Untuk membuat database berikut querinya.

```
CREATE TABLE nm_tabel
(nm_field1 tippedata1 [NOT NULL | NULL]
[{nm_field2 tippedata2 [NOT NULL | NULL]}...])
```

**Nm\_tabel** adalah nama dari tabel yang akan anda buat, usahakan namanya sesuai dengan entity yang diwakilinya dan jangan lebih dari 128 karakter. Banyaknya tabel yang dapat ditampung oleh database dibatasi oleh banyaknya objek dalam database. Sebuah database setidaknya dapat menampung 2 Miliar objek termasuk tabel, view (jenis lain tabel), stored procedure, trigger, dan constraint. **Nm\_field1**, **nm\_field2** adalah nama field yang harus ada dalam sebuah tabel yang mewakili element entity tersebut. Tiap field akan menampung data dengan tipe data tertentu yang ditunjukkan oleh tippedata1, tippedata2 dan seterusnya. Opsi berikutnya adalah constraint (syarat) masing-masing tabel apakah boleh kosong (NULL) atau harus diisi (NOT NULL). Nilai default dari constraint ini adalah NULL yang artinya field tersebut boleh kosong atau tidak diisi.

## **Constraint**

Constraint merupakan batasan atau aturan yang ada pada tabel. SQL menyediakan beberapa tipe constraint berikut:

1. NOT NULL merupakan suatu kolom yang mendefinisikan dengan constraint NOT NULL. Kolom yang berfungsi sebagai kunci primer (Primary Key) otomatis tidak boleh NULL.
2. UNIQUE mendefinisikan suatu kolom bersifat unik, artinya antara satu data dengan data lain namanya tidak boleh sama, misal alamat email.
3. PRIMARY KEY Constraint PRIMARY KEY membentuk key yang unik untuk suatu tabel.
4. FOREIGN KEY Constraint didefinisikan pada suatu kolom yang ada pada suatu tabel, dimana kolom tersebut juga dimiliki oleh tabel yang lain sebagai suatu PRIMARY KEY bisa digunakan untuk menghubungkan antara dua tabel.
5. CHECK constraint yang satu ini mendefinisikan sebuah kondisi untuk data agar dapat masuk dalam field artinya tiap pemasukan data atau editing terhadap data record,

field yang kita masukkan akan selalu diperiksa apakah isinya ada diantara data-data yang kita masukkan, jika tidak ada maka SQL akan menampilkan pesan ERROR.

6. DEFAULT digunakan untuk mendefinisikan nilai default dari field yang mana ketika ada baris baru yang dimasukkan kedalam tabel nilai default dari field akan digunakan apabila tidak ada nilai yang diberikan padanya.

7. IDENTITY hanya menampung nilai bertipe data integer. Standar property ini sudah ditentukan oleh sistem.

CONTOH:

```
CREATE TABLE MATERIAL_GROUP (  
MaterialGroup varchar(4) PRIMARY KEY NOT NULL,  
Description varchar(30) NOT NULL  
)
```

## **ALTER**

Merubah tabel berarti merubah structure table. Pada perubahan structure table ini ada beberapa yang dapat dirubah dalam artian memodifikasi salah satu atau beberapa field pada table .

Queri ini digunakan untuk menambah field pada tabel

```
ALTER TABLE namatabel ADD nama_field tipe_data (lebar)
```

Queri ini digunakan untuk merubah field pada tabel

```
ALTER TABLE namatabel ALTER COLUMN nama_field tipe_data  
(lebar)
```

Queri ini digunakan untuk menghapus field pada tabel

```
ALTER TABLE namatabel DROP COLUMN nama_field.
```

## **DROP**

Perintah terakhir dari Data Definition Language, DROP memungkinkan kita untuk menghapus seluruh objek dalam database dari DBMS. Gunakan perintah ini dengan hati-hati, perlu diingat bahwa perintah DROP akan menghapus data keseluruhan struktur dari database. Querinya sebagai berikut.

```
DROP DATABASE namadatabase (Untuk menghapus database)
```

```
DROP TABLE namatabel (Untuk menghapus tabel)
```

## PRAKTIKUM

Langkah	Keterangan												
1.	<p>Buatlah tabel Barang</p> <table><tr><th>Nama Field</th><th>Tipe data</th></tr><tr><td>ID_Barang</td><td>Varchar</td></tr><tr><td>Nama_Barang</td><td>Varchar</td></tr><tr><td>Tanggal_terima</td><td>Datetime</td></tr><tr><td>Stok_Barang</td><td>Int</td></tr></table>	Nama Field	Tipe data	ID_Barang	Varchar	Nama_Barang	Varchar	Tanggal_terima	Datetime	Stok_Barang	Int		
Nama Field	Tipe data												
ID_Barang	Varchar												
Nama_Barang	Varchar												
Tanggal_terima	Datetime												
Stok_Barang	Int												
	<pre>CREATE TABLE Barang (   ID_Barang varchar(50) not null,   Nama_Barang varchar(255) not null,   Tanggal_terima datetime not null,   Stok_Barang int(11) null default 0,   constraint PK_Barang primary key   (ID_Barang) )</pre>												
	Catat hasilnya!												
2.	<p>Buatlah tabel Pembelian</p> <table><tr><th>Nama Field</th><th>Tipe Data</th></tr><tr><td>ID_Pembeli</td><td>Varchar</td></tr><tr><td>ID_Barang</td><td>Varchar</td></tr><tr><td>Tanggal_Beli</td><td>Datetime</td></tr><tr><td>Jumlah_Pembelian</td><td>Int</td></tr><tr><td>Nama_Pembelian</td><td>Varchar</td></tr></table>	Nama Field	Tipe Data	ID_Pembeli	Varchar	ID_Barang	Varchar	Tanggal_Beli	Datetime	Jumlah_Pembelian	Int	Nama_Pembelian	Varchar
Nama Field	Tipe Data												
ID_Pembeli	Varchar												
ID_Barang	Varchar												
Tanggal_Beli	Datetime												
Jumlah_Pembelian	Int												
Nama_Pembelian	Varchar												
	<pre>CREATE TABLE Pembelian (   ID_Pembeli varchar(50) not null,   ID_Barang varchar(50) null ,   Tanggal_Beli datetime not null,</pre>												

	<pre> Nama_Pembeli <b>varchar</b> (60) not null, Jumlah_Pembelian <b>int</b> null , <b>constraint</b> PK_Pembelian <b>primary key</b> (ID_Pembeli) ) </pre>
	Catat hasilnya!
3.	Tambahkan Foreign Key dengan menggunakan perintah ALTER
	<pre> <b>ALTER TABLE</b> Pembelian <b>ADD CONSTRAINT</b> FK_Pembelian_RELATION_Barang <b>foreign</b> <b>key</b> (ID_Barang) <b>REFERENCES</b> Barang (ID_Barang) </pre>
	Catat hasilnya!
4.	Tambahkan Field “Alamat_Pembeli” pada tabel Pembelian
	<pre> <b>ALTER TABLE</b> Pembelian <b>ADD</b> Alamat_Pembeli <b>varchar</b> (70) </pre>
	Catat hasilnya!
5.	Hapus kolom nama pembeli pada tabel Pembelian
	<pre> <b>ALTER TABLE</b> Pembelian <b>DROP COLUMN</b> Nama_Pembeli </pre>
	Catat hasilnya!
6.	Hapus tabel Barang
	<pre> <b>DROP TABLE</b> Barang </pre>
	Catat hasilnya!

## TUGAS

1. Buatlah database baru dengan perintah SQL dengan nama “Mahasiswa” yang memiliki dua tabel yaitu tabel “Jurusan” dan tabel “Biodata”!
2. Isikan kedua tabel dengan field masing-masing dibawah ini! Tabel  
Jurusan : KodJur, NamaJur, Ketua\_jurusan  
Tabel Biodata : NoMhs, KodJur, Nama, Nilai, Alamat, JK
3. Tentukan tipe data masing-masing field, Primary key serta Foreign Key nya agar kedua tabel saling berhubungan!

*~ Selamat Mengerjakan ~*