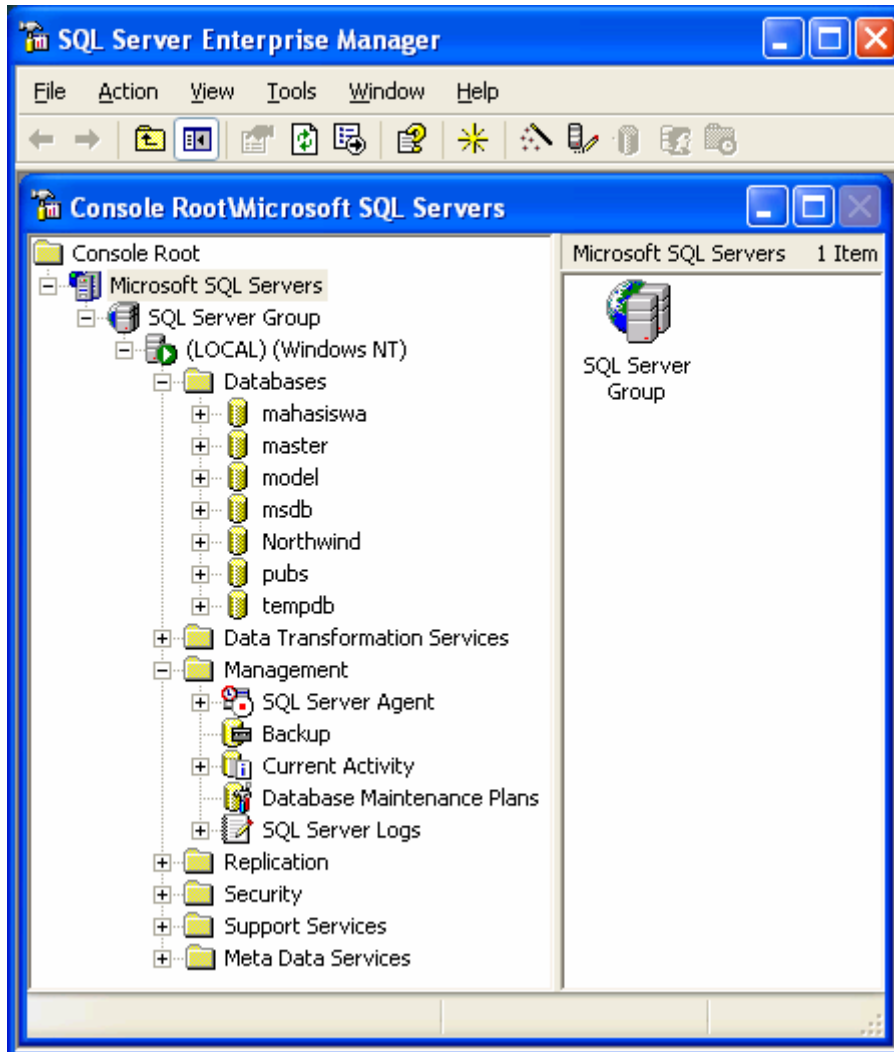


Database SQL Server

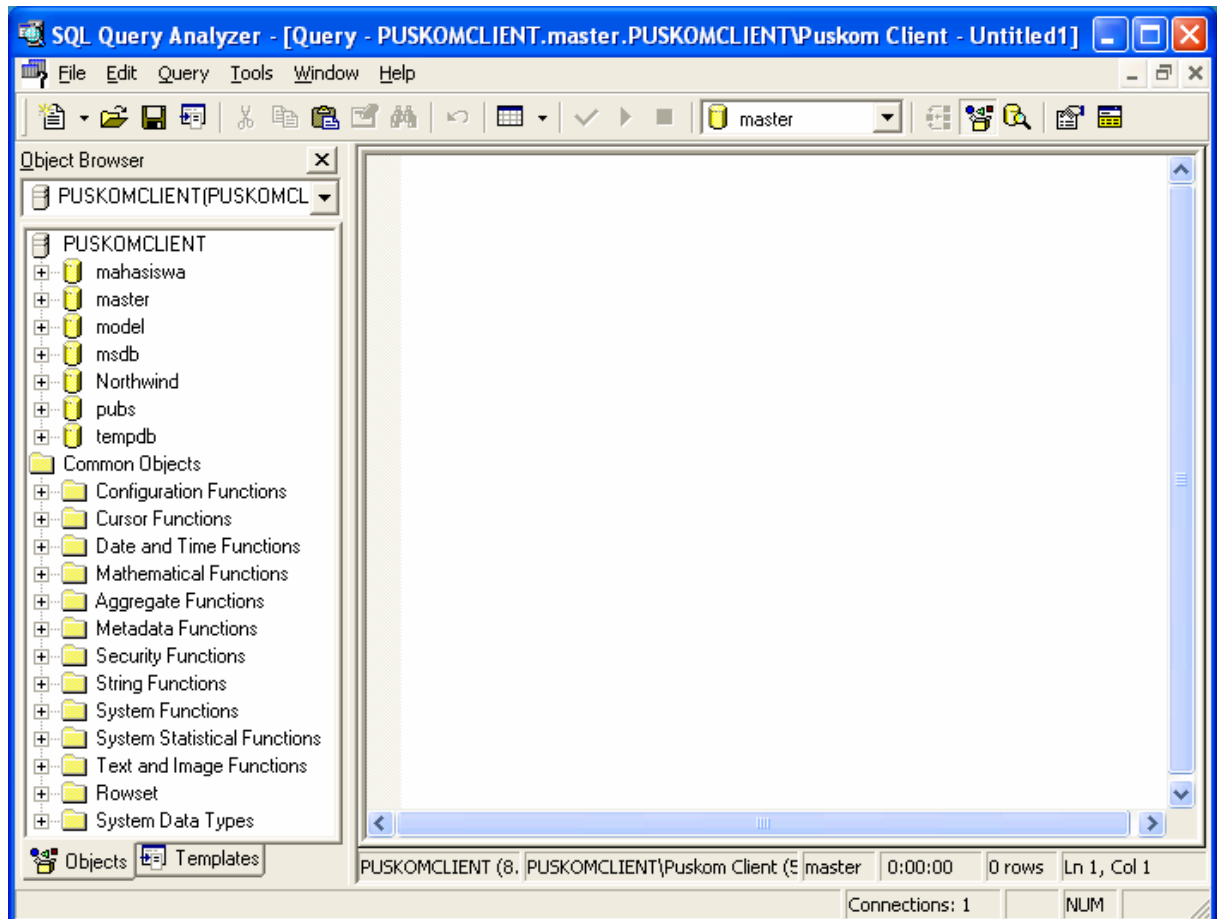
Database SQL Server dapat dibuat menggunakan Enterprise Manager atau Query Analyzer. Penggunaan Enterprise Manager pada SQL Server dapat membantu memudahkan DB owner untuk mengelola Database.

Tampilan pada Enterprise Manager



Enterprise Manager

Dan untuk editor yang berbasis SQL, dapat menggunakan Query Analyzer. Dalam query analyzer, kita dapat menuliskan perintah sql dan kemudian juga dapat dijalankan sehingga menghasilkan nilai keluaran dari sintak sql yang dituliskan.



Query Analyzer

Data Definition Language (DDL)

a. CREATE DATABASE

SINTAK :

CREATE DATABASE *database_name*

Contoh Penerapan :

```
create database mhs
```

Maka dalam Sql akan terbentuk atau terdapat sebuah database baru dengan nama *database Mhs* dengan contoh pesan hasil pembuatan database baru :

```
The CREATE DATABASE process is allocating 0.75 MB on disk 'mhs'.
The CREATE DATABASE process is allocating 0.49 MB on disk 'mhs_log'.
```

b. ALTER DATABASE

SINTAK :

ALTER DATABASE *database_name*

Contoh penerapan :

```
alter database mhs
modify name=mahasiswa
```

Perintah diatas digunakan untuk mengganti nama database menjadi 'Mahasiswa'. Contoh pesan nama database telah diganti.

The database name 'mahasiswa' has been set.

c. DROP DATABASE

SINTAK :

DROP DATABASE *database_name*

Contoh penggunaannya :

drop database mahasiswa

Dengan perintah yang kita ketikan diatas, maka database Mhs yang telah kita buat sebelumnya akan terhapus.

Tetapi kita tidak bisa menghapus database jika sedang dalam proses restore, terbuka untuk reading, dan writing oleh user dan tabel sedang direplikasi.

Contoh pesan hasil penghapusan :

Deleting database file 'C:\Program Files\Microsoft SQL Server\MSSQL\data\mhs_log.LDF'.

Deleting database file 'C:\Program Files\Microsoft SQL Server\MSSQL\data\mhs.mdf'.

d. CREATE TABLE

Dalam suatu database SQL server kita dapat menciptakan sekitar 2 juta tabel. Melalui statement CREATE TABLE kita dapat menentukan :

- Database dimana tabel dapat ditempatkan
- Owner dari tabel
- Table name – tidak boleh sama dengan tabel atau view lain dengan owner yang sama dalam database yang sama
- Menentukan jumlah column sampai dengan 1024 column
- PRIMARY KEY (optional)
- 1 s/d 250 constraint UNIQUE (optional)
- 1 s/d 153 constraint FOREIGN KEY (optional)
- 1 atau lebih constraint CHECK yang membatasi apa yang dapat dimasukan kedalam tabel (optional)
- FILEGROUPS dimana tabel disimpan (optional)

SINTAK :

CREATE TABLE *nama_tabel*

Contoh Penerapannya :

Nama tabel : **Mahasiswa**

Field	Tipe Data	Size
Nim	Varchar	7
Nama	Varchar	25
Alamat	Varchar	30
No_tlp	Varchar	12
Tempat_Ihr	Varchar	15
Tanggal_Ihr	Datetime	
Agama	Varchar	15
Gol_darah	Varchar	2

```

Create table Mahasiswa
( Nim varchar(7) not null,
  Nama varchar(25) not null,
  Alamat varchar(30) not null,
  No_telp varchar(12),
  Tempat_lhr varchar(15),
  Tanggal_lhr datetime,
  Agama varchar(15),
  Gol_darah varchar(2),
  Constraint MahasiswaPK primary key(nim)
)

```

PRIMARY KEY

Adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik suatu kejadian spesifik, tetapi juga mewakili setiap kejadian dari entity.

Contoh Penerapan PRIMARY KEY :

```

Create TABLE karyawan
( NIK varchar(5) constraint NikPK PRIMARY KEY,
  Nama varchar(20),
  Alamat varchar(30),
  No_telp varchar(12)
)

```

UNIQUE

UNIQUE digunakan untuk menjaga integritas entity data. Baris data pada column yang diberikan constraint UNIQUE tidak akan mempunyai duplikat. Tidak seperti PRIMARY KEY, SQL server memperkenalkan column dengan constraint UNIQUE mempunyai nilai Null.

Contoh Penerapannya :

```

Create TABLE karyawan2
( NIK varchar(5) PRIMARY KEY,
  Nama varchar(20) UNIQUE,
  Alamat varchar(30),
  No_telp varchar(12)
)

```

NOT NULL

SQL server tidak memperkenankan column dengan data kosong. Setiap data yang tidak diketahui datanya akan diisi dengan nilai Null. Jika pemilik tabel tidak menginginkan sebuah column diisi dengan nilai Null gunakan constraint Not Null.

Contoh Penerapannya :

```

Create TABLE karyawan3
( NIK varchar(5),
  Nama varchar(20) NOT NULL,
  Alamat varchar(30) NOT NULL,
  No_telp varchar(12)
)

```

CHECK

Check membantu user dalam melakukan entry data. Dengan CHECK kita dapat menentukan ruang nilai tertentu. Misalnya umur karyawan tidak mungkin dibawah 18 tahun dan tidak bisa lebih dari 65 tahun.

Maka Contoh Penerapannya :

```
Create TABLE karyawan4
( NIK varchar(5),
  Nama varchar(20) NOT NULL,
  Alamat varchar(30) NOT NULL,
  No_telp varchar(12),
  Umur int check (Umur >=18 AND Umur <= 65)
)
```

FOREIGN KEY

Foreign Key merupakan mitra dari Primary Key dalam bentuk relasi. Keberadaan Foreign Key dalam suatu tabel menyatakan bahwa field tersebut merupakan PRIMARY KEY pada tabel lain.

Contoh Penerapannya :

```
CREATE TABLE GajiBulanan
( GajiPokok SmallMoney DEFAULT NULL,
  Tunjangan SmallMoney DEFAULT NULL,
  TotalGaji As GajiPokok + Tunjangan,
  NIK varchar(5) PRIMARY KEY
)
```

```
Create TABLE karyawan5
( NIK varchar(5) UNIQUE,
  Nama varchar(20) NOT NULL,
  Alamat varchar(30) NOT NULL,
  No_telp varchar(12),
  Umur int check (Umur >=18 AND Umur <= 65),
  Constraint NIKPK FOREIGN KEY(NIK) REFERENCES GajiBulanan(NIK)
)
```

DEFAULT

Jika user tidak menentukannya, constraint default akan memberikan nilai yang disuplai oleh system SQL Server. Dalam contoh diatas, system akan memberikan nilai Null.

Contoh Lain Penerapannya :

```
Create TABLE karyawan6
( NIK varchar(5) UNIQUE,
  Nama varchar(20) NOT NULL,
  Alamat varchar(30) NOT NULL,
  No_telp varchar(12),
  Kota varchar(15) DEFAULT 'Bandung',
  Umur int check (Umur >=18 AND Umur <= 65),
  Constraint NIKPK FOREIGN KEY(NIK)
  REFERENCES GajiBulanan(NIK)
)
```

IDENTITY COLUMN

Merupakan sarana system dalam memberikan nomor urut secara otomatis pada setiap entry baris baru. Secara default noor awal yang diberikan adalah 1, dan inkremen 1 untuk setiap baris data.

Contoh Penerapannya :

```
CREATE TABLE Peserta  
( NoDaftar varchar(5) IDENTITY(1,1),  
  Nama varchar(20),  
  Alamat varchar(30)  
)
```

Maka pada tabel tersebut column NoDaftar dimulai dengan 1000, dan setiap diinput pendaftar baru maka nilai NoDaftar akan bertambah 1.

e. ALTER TABLE

Digunakan untuk memodifikasi table yang sudah ada, seperti menambah dan menghapus constraint.

Contoh Penerapannya :

Menambah, menghapus, dan melumpuhkan constraint

Menghapus constraint FOREIGN KEY dengan nama NIK pada table Karyawan yang terdapat dalam database Mhs.

```
ALTER TABLE Mhs.dbo.Karyawan  
DROP Constraint NIK
```

Menambah constraint FOREIGN KEY pada table Karyawan untuk membentuk relasi dengan table GajiBulanan.

```
ALTER TABLE Karyawan  
ADD Constraint KaryawanPK FOREIGN KEY (NIK) References GajiBulanan (NIK)
```

Melalui statement ALTER juga dapat melumpuhkan (disable) untuk sementara suatu constraint, misalnya untuk kebutuhan input data.

```
ALTER TABLE Karyawan  
NOCHECK Constraint NIK
```

Untuk mengaktifkan kembali constraint :

```
ALTER TABLE Karyawan  
CHECK Constraint NIK
```

Menambah, menghapus, dan mengubah column

Menambah column Discount pada table customer.

```
ALTER TABLE NorthWind.dbo.Customers  
ADD Discount Decimal(5,3) default 0  
CHECK (Discount between 0 AND 100)
```

ALTER juga bisa mengubah tipe data, ukuran, dan membuatnya menjadi nullable.

```
ALTER TABLE Customers  
ALTER COLUMN Discount(7,5) not null
```

Contoh menghapus column :

```
ALTER TABLE Customers  
DROP COLUMN Discount
```

Column dalam table tidak bisa dihapus jika kondisi berikut yang terjadi pada column yang akan dihapus :

- Disertakan dalam replication
- Digunakan dalam suatu index
- Menggunakan constraint CHECK, FOREIGN KEY, UNIQUE, atau PRIMARY KEY
- Digunakan sebagai kunci full_text table.

f. DROP TABLE

Digunakan untuk menghapus tabel yang sudah ada.

Sintak :

```
DROP TABLE table_nama
```

Contoh Penerapannya :

```
Uses Mhs  
DROP TABLE mahasiswa
```