

Implementasi Administrator Jaringan dan Database

Toni Wijanarko Adi Putra, M.Kom



Implementasi Administrator Jaringan dan Database

Oleh : Toni Wijanarko Adi Putra., M.Kom.

ISBN 978-623-8120-01-7 (PDF)



Implementasi Administrator Jaringan dan Database

Penulis:

Toni Wijanarko Adi Putra., M.Kom.

ISBN : 978-623-8120-01-7

Editor:

Indra Ava Dianta, S.Kom., M.T.

Penyunting :

Danang, S.Kom., M.T.

Desain Sampul dan Tata Letak :

Nuris Dwi Setiawan, S.Kom., M.T.

Penerbit :

Yayasan Prima Agus Teknik

Redaksi: Jln Majapahit No 605 Semarang

Tlpn. (024) 6723456

Fax . 024-6710144

Email: penerbit_ypat@stekom.ac.id

Distributor Tunggal:

UNIVERSITAS STEKOM

Jln Majapahit No 605 Semarang

Tlpn. (024) 6723456

Fax . 024-6710144

Email: info@stekom.ac.id

Hak Cipta dilindungi Undang undang

Dilarang memperbanyak karya Tulis ini dalam bentuk dan dengan cara apapun tanpa ijin tertulis dan penerbit.

KATA PENGANTAR

Terima kasih kepada Tuhan Yang Maha Esa atas kasih dan karunia-Nya, karena hanya bimbingan dan rahmat-Nya buku ini bisa selesai dengan lancar tanpa terhambat oleh sesuatu apapun.

Terimakasih kembali saya ucapkan kepada Mahasiswa yang telah membantu dalam hal pengumpulan gambar, teman teman dosen yang telah memberikan masukan berupa kritik dan saran serta semua pihak yang tidak bisa saya sebutkan satu persatu yang telah membantu sehingga tersusunnya buku ini.

Buku ini adalah panduan praktis dan pengenalan Microsoft SQL Server, serta panduan referensi untuk bahasa Transac-SQL yang digunakan oleh produk ini. Ada 18 bab yang membahas mengenai modul utama dan sumber daya dari program ini, serta panduan referensi untuk fungsi dan perintah dari bahasa Transact-SQL SQL Server adalah sebuah produk besar yang memiliki banyak sumber daya dan keistimewaan. Walaupun antarmuka grafisnya menjadikannya mudah dioperasikan, produk ini adalah produk yang sangat kompleks dan sangat rinci (sama seperti database perusahaan lain), dan hanya dengan banyak latihan dan waktu pembaca akan mendapatkan pengetahuan yang mendalam mengenai SQL Server, Karena karakteristik ini, kami memilih untuk membahas sumber daya utama dan item-item baru dari SQL Server pada buku ini.

Semoga dengan kehadiran buku ini dapat memberikan manfaat yang sebesar besarnya khususnya bagi Mahasiswa yang mengambil matakuliah Database Server dengan SQL dan para pembaca pada umumnya.

Penulis

Toni Wijanarko Adi Putra, S.Kom., M.Kom

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL	ii
KATA PENGANTAR	iv
DAFTAR ISI	vi
BAB 1 Gambaran Singkat SQL Server	1
1. Arsitektur Client/Server	1
2. Arsitektur SQL Server	2
3. Komponen SQL Server	3
a. Database	3
b. Tabel	5
c. Database Diagram	5
d. Indeks	6
e. View	6
f. Stored Procedure	7
g. Trigger	7
h. Full-Text Indexes	8
4. Wizard	8
5. Kesimpulan	9
BAB 2 Membuat Database	10
1. Membuat Database secara Manual.....	10
2. Membuat Database dengan Wizard	12
3. Menentukan Rencana Pemeliharaan Untuk Database	16
a. Database Maintenance Plan Wizard	16
b. Membuat Rencana Pemeliharaan Lain	22
4. Membuat Database Menggunakan Transact-SQL	22
5. Mengubah Karakteristik Database	23
6. Menghapus Database	23
7. Melihat Komponen Database	23
8. Kesimpulan	24
BAB 3 Tabel	25
1. Karakteristik dari Tabel	25
a. Filegroup	25

b. Tipe Data	26
c. Sinonim	27
d. Batasan	27
2. Tabel Contoh	28
3. Membuat Tabel Dengan Enterprise Manager	29
4. Karakteristik Kolom	29
a. Kolom Pengenal	31
b. Nilai Null	31
c. Kunci Primer	31
d. Kunci Asing	32
e. Batasan UNIQUE	32
f. Batasan CHECK	32
5. Menggunakan Tabel	32
a. Menghapus Tabel	32
b. Mengganti Nama Tabel	33
c. Melihat Data Tabel	33
6. Menyunting Data di Dalam Tabel dengan Enterprise Manager	35
a. Menambah Kolom Baru Ke Dalam Tabel	35
b. Menghapus Kolom Dari Tabel	35
c. Melihat Properti Dari Tabel	36
7. Membuat Tabel Authors	37
a. Membuat Kunci Primer	38
8. Kesimpulan	39
BAB 4 Diagram Database	40
1. Membuat Diagram	41
2. Menggunakan Diagram	43
a. Membuka Diagram	43
b. Mencetak Diagram	44
c. Keluar Dari Diagram	46
3. Menggunakan Tabel Pada Diagram Database	46
a. Melihat Properti Tabel	46
b. Mengubah Tampilan dari Tabel	49
c. Menambahkan Tabel Ke Dalam Diagram	50
d. Menghapus Tabel Dari Diagram	52
e. Menambah Tabel-Tabel Yang Berhubungan	53
f. Menyertakan Catatan	55

g. Membuat Tabel Baru	56
h. Membuat Hubungan Antartabel	57
4. Kesimpulan	59
BAB 5 Query Analyzer	60
1. Berhubungan dengan Server	60
2. Layar Kerja	61
a. Mengubah Ukuran Panel	62
b. Menjalankan Perintah dengan Beberapa Baris	62
3. Menjalankan Query	63
a. Melihat Query dengan Grid	64
b. Melihat Query Execution Plan	65
c. Mengatur Tampilan Query Execution Plan	67
4. Menganalisa Query	67
5. Menyimpan dan Membuka Query	68
6. Menyunting Query	69
7. Menyimpan Results Panel	69
a. Mencetak Query dan Hasilnya	70
b. Mengubah Konfigurasi Tampilan	70
8. Menampilkan Help Online	71
9. Kesimpulan	72
BAB 6 Dasar-Dasar SQL	73
1. SQL Server dan SQL	73
2. Dasar-Dasar SQL	74
a. Jenis-Jenis Deklarasi SQL	74
b. DDL (Data Definition Language)	74
c. DML (Data Manipulation Language)	75
d. Tipe-Tipe Data	75
e. Ekspresi dan Operator	76
f. Perintah SELECT	78
g. Contoh dari SELECT	80
h. Fungsi	83
i. Perintah CREATE TABLE	87
j. Perintah INSERT	89
k. Perintah DELETE	92
l. Perintah UPDATE	93

m. Perintah DROP TABLE	95
3. Ada sebuah tabel Perintah IF EXISTS	96
4. Kesimpulan	98
BAB 7 Indeks	99
1. Jenis-Jenis Indeks	99
2. Fill Factor	100
3. Membuat Indeks	101
4. Membuat Indeks pada Saat Pembuatan Tabel	101
a. Membuat Indeks pada Tabel yang Sudah Ada	104
b. Membuat Indeks dengan Wizard	105
c. Merekonstruksi Indeks	108
5. Melihat Indeks	108
6. Menghapus Indeks	109
7. Kesimpulan	110
BAB 8 View	111
1. Membuat View	112
2. Menyimpan View	115
3. Memeriksa View	115
4. Menyunting View	117
a. Mencegah Tampilan Kolom	117
b. Mengubah Urutan Kolom	118
c. Menyisipkan Kolom Baru	119
d. Mengatur Panel	119
5. Mengubah Nama View	120
6. Menghapus View	120
7. Kesimpulan	120
BAB 9 Prosedur	121
1. Struktur Prosedur	121
2. Jenis-Jenis Prosedur	122
3. Komponen dari Prosedur	122
a. Parameter	122
b. Kode yang Dihasilkan	123
c. Kode Nilai	123
4. Membuat Prosedur	124

5. Menjalankan Prosedur	125
6. Membuat Prosedur dengan Parameter	127
7. Mengubah Prosedur	128
8. Menggunakan Nilai Standar sebagai Parameter	129
9. Mengkompilasi Ulang Prosedur	130
10. Menghapus Prosedur	130
11. Prosedur Sistem	131
12. Kesimpulan	137
BAB 10 Trigger	138
1. Komponen dari Trigger	138
2. Batasan Trigger	139
3. Membuat Trigger	139
a. Membuat Trigger dengan Enterprise Manager	140
b. Membuat Trigger dengan Query Analyzer	143
4. Tabel Inserted dan Deleted	145
5. Mengubah Trigger	147
6. Membuat Replikasi Baris dengan Trigge	147
7. Melihat Trigger dari Tabel	151
8. Menghapus Trigger	153
9. Kesimpulan	153
BAB 11 Replikasi	154
1. Model Replikasi	154
2. Jenis-Jenis Replikasi	154
a. Snapshot	154
b. Transactional	154
c. Merge	154
3. Berlangganan Publikasi	155
4. Contoh Replikasi Transactional	155
5. Mendefinisikan Publikasi	160
6. Mengirimkan Publikasi	168
7. Memeriksa Replikasi	175
8. Menguji Replikasi	175
9. Kesimpulan	176

BAB 12 Replikasi Merge	177
1. Mendefinisikan Replikasi	178
2. Mengirimkan Publikasi	186
3. Memeriksa Publikasi	192
4. Menguji Replikasi	193
a. Menggunakan Query Analyzer	193
5. Kesimpulan	200
BAB 13 Data Warehousing dan OLAP	201
1. Apa itu Data Warehouse?	202
a. Data yang Terkonsolidasi dan Konsisten	202
b. Data yang Berorientasi pada Topik Tertentu	203
c. Data yang Bersifat Historis	203
d. Data Read-Only	203
2. Data Warehousing dan OLAP	204
3. Memperkenalkan Microsoft Decision Support Services	205
a. Kemudahan Penggunaan	205
b. Model Data yang Fleksibel	208
c. Skalabilitas	209
d. Arsitektur Server	210
4. Arsitektur Client	212
5. Membuat dan Membangun Data Warehouse dan Sistem OLAP	213
6. Kesimpulan	213
BAB 14 Contoh Praktis OLAP Manager	214
1. Mengkonfigurasi DSN	214
2. Menjalankan OLAP Manager	216
3. Mengkonfigurasi Sumber Data	218
4. Membuat Dimensi Bersama	220
a. Membuat Dimensi Product	223
b. Membuat Dimensi Store	226
5. Membangun Kubus	228
a. Membuat Dimensi dengan Cepat	230
6. Menyunting Kubus	231
a. Menambahkan Dimensi Baru	232
b. Mengubah Nama Dimensi	234
c. Menambah Aturan	235

7. Merancang Data Koleksi	238
8. Menjelajah dengan Kubus	241
9. Kesimpulan	243
BAB 15 Data Transformation Services	245
1. Konsep	245
a. Mengimpor dan Mengekspor Data	245
b. Mentransformasi Data	245
c. Arsitektur	245
2. Membuat Paket DTS	246
3. Membuat Paket DTS	247
4. Mentransformasi Data	252
a. Customer Dimension	258
b. Geography Dimension	260
c. Products Dimension	263
d. Tabel Orders Fact	265
5. Menjalankah Paket DTS	269
6. Kesimpulan	272
BAB 16 Contoh Praktis DTS	273
1. Menggunakan Export Wizard	273
2. Kesimpulan	278
BAB 17 Program Bantu SQL Server	279
1. Gambaran Singkat	279
a. Microsoft Management Console	279
b. SQL Server Client Network Utility	279
c. SQL Server Enterprise Manager	281
d. SQL Server Network Utility	281
e. Performance Monitor	282
f. SQL Server Profiler	282
g. SQL Server Query Analyzer	282
h. SQL Server Service Manager	283
i. SQL Server Setup	283
j. Version Upgrade Wizard	283
2. SQL Server Profiler	283
a. Mendaftarkan Server	284

b. Membuat Trace	284
c. Filter	286
d. Menjalankan Trace	286
e. Menyimpan Trace	286
3. SQL Server Service Manager	286
4. Performance Monitor	287
5. Kesimpulan	289
BAB 18 Mem-backup dan Merestorasi Data	290
1. Jenis-Jenis Backup	290
2. Membuat Backup dengan Enterprise Manager	290
3. Mengubah Database	292
4. Merestorasi Backup	293
5. Menjadwalkan Backup	295
6. Membuat Backup Diferensial	296
7. Merestorasi Backup	299
a. Merestorasi Backup Lengkap	300
b. Merestorasi Backup Differensial	302
8. Kesimpulan	304
DAFTAR PUSTAKA	305

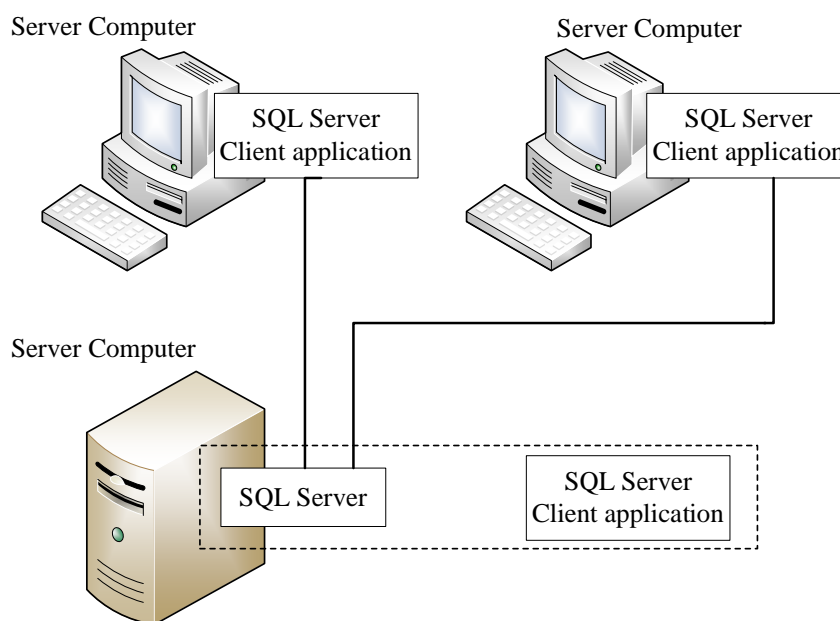
BAB 1

Gambaran Singkat SQL Server

Bab ini membahas mengenai konsep-konsep dasar yang terdapat pada SQL Server dan keistimewaan-keistimewaannya.

Arsitektur Client/Server

SQL Server adalah basis data relasional yang mendukung aplikasi dengan arsitektur klien/server. Basis data berada di komputer server dan dibagikan di antara banyak pengguna yang menjalankan aplikasi di komputer lokal atau klien. Arsitektur ini memberikan integritas data yang tinggi karena semua pengguna bekerja dengan data yang sama. Pemantauan informasi yang ditambahkan ke database berlaku untuk semua pengguna. Arsitektur klien/server secara signifikan mengurangi lalu lintas jaringan karena hanya menyediakan informasi yang diminta oleh pengguna. Misalnya, jika pencarian database dengan 100.000 record hanya mengembalikan 3 record, 3 record tersebut dikirim melalui jaringan ke komputer klien. Dalam sistem tradisional, semua 100.000 catatan dikirim melalui jaringan. Ini menyederhanakan tugas pemeliharaan seperti mencadangkan dan memulihkan data karena data hanya ada di satu tempat. Gambar 1.1 di bawah ini adalah contoh penggunaan dasar arsitektur ini:

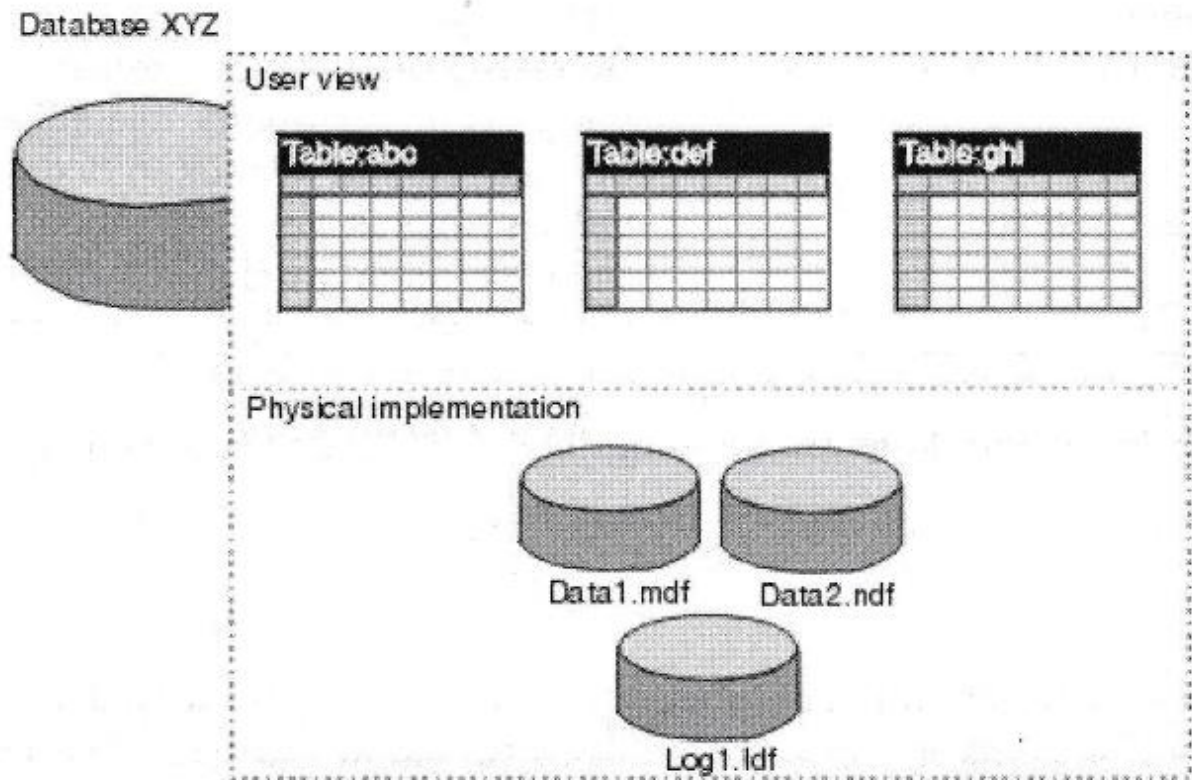


Gambar 1.1 Arsitektur Client/Server

Arsitektur SQL Server

Database SQL Server dibagi menjadi beberapa komponen logis seperti tabel, tampilan, dan elemen lain yang terlihat oleh pengguna. Barang-barang ini secara fisik disimpan dalam dua atau lebih file di disk. Format file atau lokasi tempat penulisan elemen logis ini tidak diketahui oleh pengguna sistem.

SQL Server berisi banyak basis data yang digunakan oleh banyak pengguna. Perusahaan mungkin memiliki database tunggal yang digunakan oleh banyak pengguna di beberapa departemen, atau perusahaan mungkin memiliki banyak database yang hanya digunakan oleh pengguna tertentu di setiap departemen. Gambar 1.2 di bawah menunjukkan tampilan pengguna sebagai tiga tabel dan implementasi fisik dari tabel tersebut menggunakan file data, direktori, dan log.



Gambar 1.2 Arsitektur SQL Server

Setelah database dibuat, pengguna bisa mendapatkan izin yang diberikan kepada mereka. Ini memungkinkan SQL Server untuk menyimpan banyak basis data dan membatasi akses ke setiap basis data untuk pengguna tertentu.

Komponen SQL Server

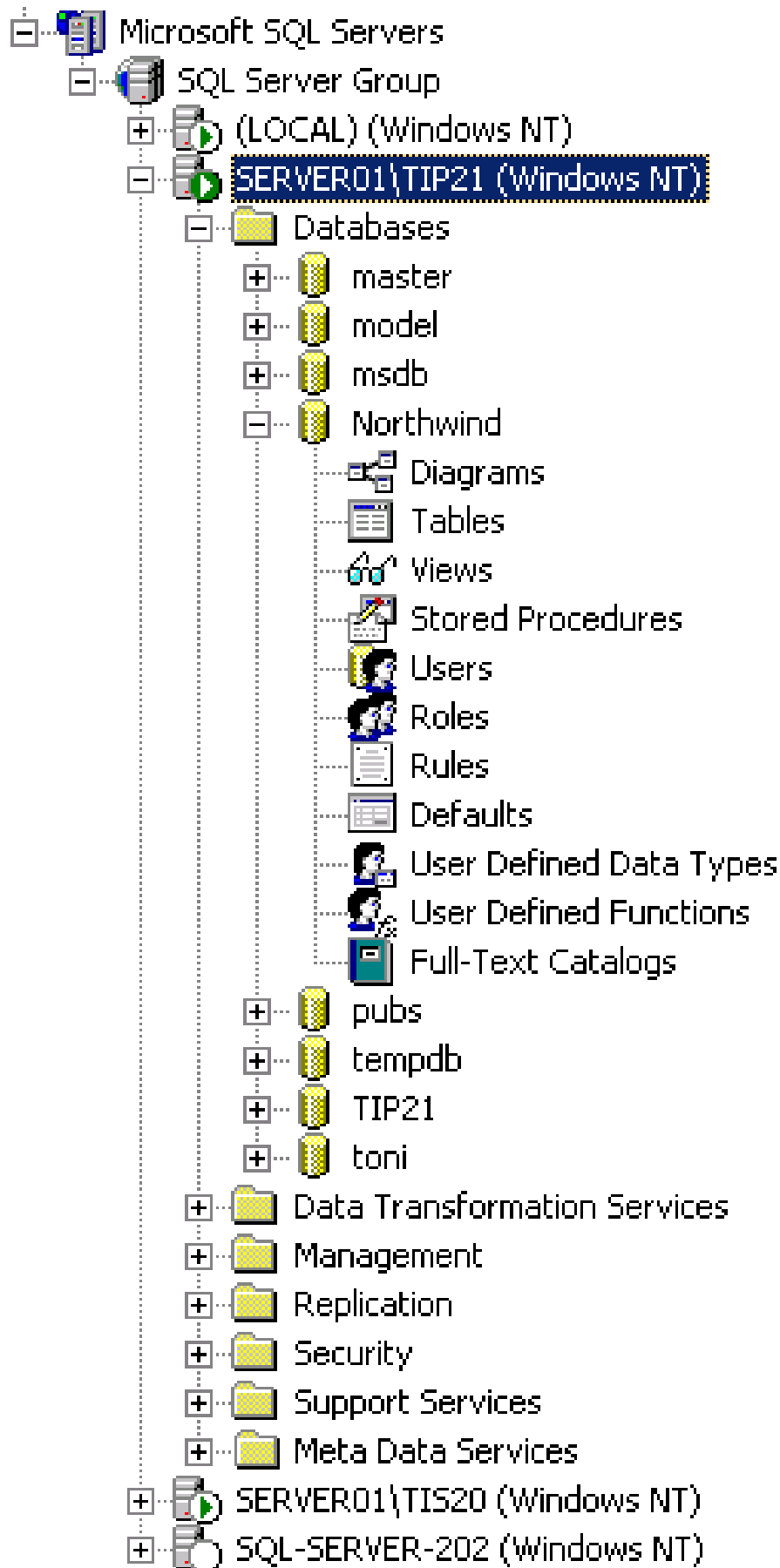
Tabel berikut menunjukkan komponen utama SQL Server.

Komponen	Keterangan
Database	Berisi objek yang digunakan untuk mewakili, menyimpan, dan mengakses data.
Tabel	Menyimpan baris atau rekaman dan hubungannya dengan tabel lain.
Database Diagram	Menampilkan objek database secara grafis sehingga dapat dimanipulasi tanpa menggunakan bahasa Transact-SQL.
Indeks	File tambahan yang meningkatkan kecepatan akses ke baris tabel.
View	Menyediakan cara untuk melihat berbagai data dalam satu atau beberapa tabel.
Stored Procedure	Program Transact-SQL disimpan di server yang melakukan tugas tertentu.
Trigger	Serangkaian prosedur tersimpan khusus yang berjalan secara otomatis saat operasi tertentu dilakukan di tabel.
Full-Text Indexes	Indeks khusus yang memudahkan pencarian di kolom dengan tipe data Varchar dan Teks.

Di bawah ini adalah **penjelasan** lebih rinci **dari item** di atas.

Database

Database adalah kumpulan tabel, tampilan, indeks, pemicu, prosedur, dan objek lainnya. SQL Server 7 dapat menghosting banyak basis data. Gambar 1.3 di bawah menunjukkan sebuah server bernama Server01\TIP21 berisi delapan database bernama master, model, msdb, northwind, pubs, tempdb, TIP21, dan toni.



Gambar 1.3 Database

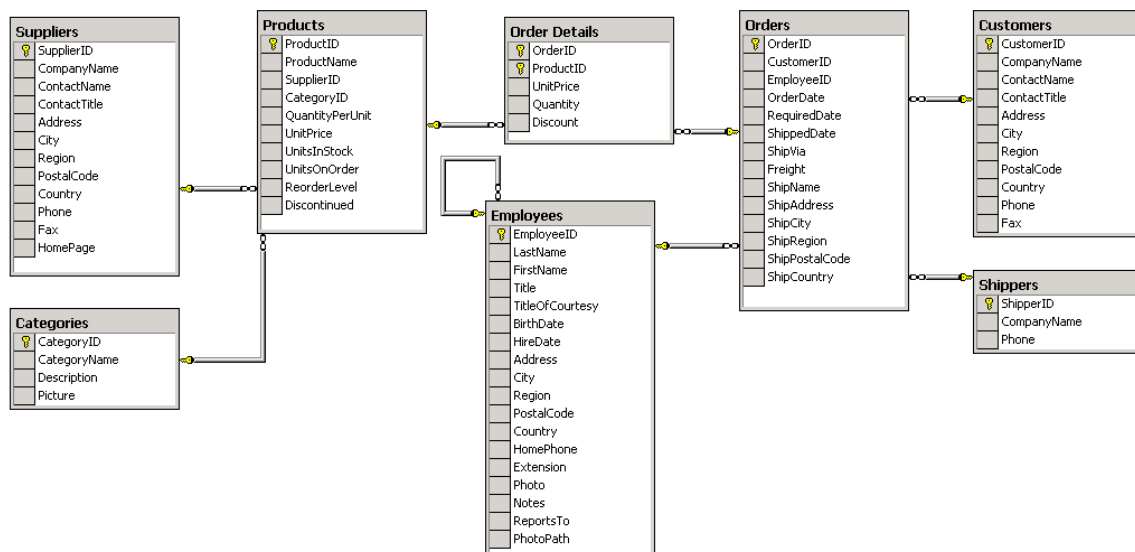
Perhatikan bahwa database Northwind menampilkan komponen-komponen yang dipelihara olehnya seperti tabel, view, dan sebagainya.

Tabel

Tabel adalah jantung dari sebuah database. Tabel menyimpan data yang dikelompokkan dalam baris dan kolom seperti spreadsheet. Setiap baris mewakili catatan dan setiap kolom adalah atribut atau bidang. Setiap bidang berisi beberapa informasi. Misalnya, tipe data bidang yang disebut "Nilai" adalah Numerik, sedangkan bidang "Nama" berisi data dengan tipe data "Teks". Bidang array dapat memiliki batasan pada konten yang dapat disimpan di bidang. Misalnya, bidang bernama Jumlah tidak dapat menyimpan teks, hanya dapat menyimpan angka, dan tidak dapat menerima nilai kurang dari 1 atau lebih besar dari 100.

Database Diagram

SQL Server dapat membuat skema database. Bagan ini adalah representasi grafis dari tabel, indeks, dan tampilan yang disimpan dalam database dan dapat diedit menggunakan teknik drag-and-drop dan interaksi dengan kotak dialog. Ini memungkinkan banyak tugas dilakukan tanpa menggunakan bahasa Transact-SQL, seperti mengubah properti fisik database atau tabelnya. SQL Server dapat membuat skema database. Bagan ini adalah representasi grafis dari tabel, indeks, dan tampilan yang disimpan dalam database dan dapat diedit menggunakan teknik drag-and-drop dan interaksi dengan kotak dialog. Ini memungkinkan banyak tugas dilakukan tanpa menggunakan bahasa Transact-SQL, seperti mengubah properti fisik database atau tabelnya.

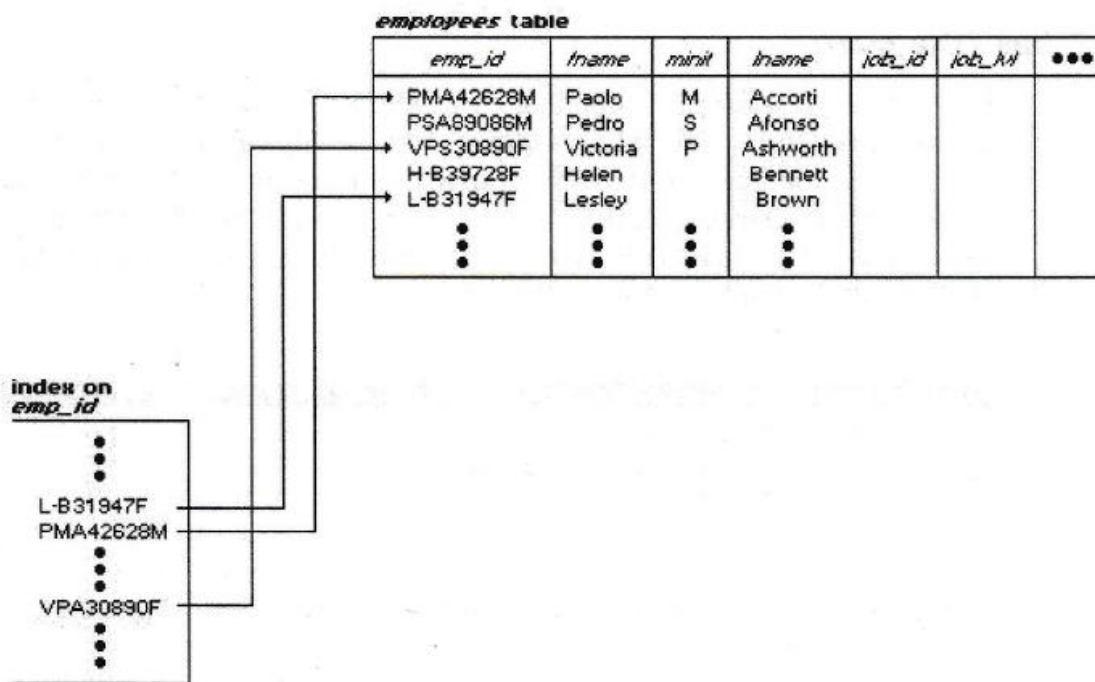


Gambar 1.4 Skema Database

Indeks

Indeks adalah jenis file khusus yang bekerja dengan tabel. Tujuannya adalah untuk mempercepat akses ke rekaman atau grup rekaman tertentu. Indeks dapat dibandingkan dengan indeks sebuah buku, di mana kita tidak perlu membaca seluruh buku untuk menemukan halaman tertentu, kita hanya perlu menggunakan indeks untuk menemukan subjek dan nomor halaman.

Pada database Pubs, tabel Employees memiliki indeks berdasarkan kolom emp_id, seperti terlihat pada Gambar 1.5 di bawah ini.



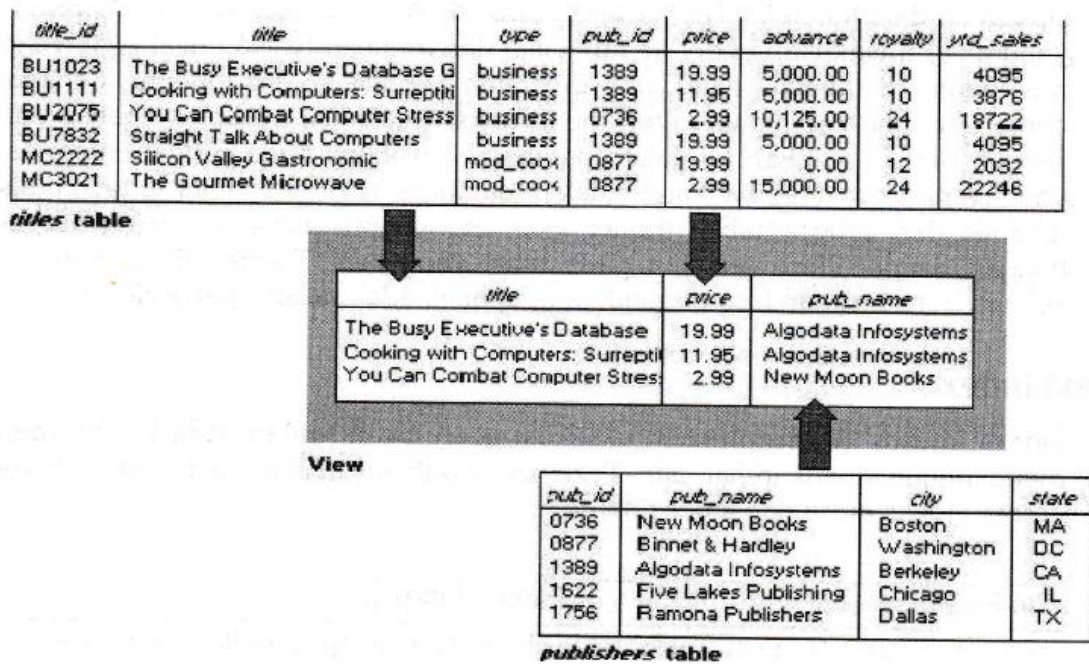
Gambar 1.5 Indeks

Saat mencari tabel Karyawan, server mengenali kolom kunci dan mencari indeks yang pada dasarnya berisi salinan isi kolom *emp_id* dan alamat barisnya di tabel. Beberapa indeks dibuat secara otomatis, mis. B. kolom primary key. Situasi optimal adalah membuat indeks pada setiap kolom dalam database untuk meningkatkan kinerja. Tetapi setiap p-index yang dibuat menggunakan ruang disk dan bahkan tidak digunakan secara teratur.

View

View adalah tabel virtual yang isinya ditentukan oleh query di database. Tampilan ini bukan tabel fisik tetapi sekumpulan instruksi yang mengembalikan hasil kumpulan

data. Gambar 1.5 berikut ini menunjukkan dua tabel yang berfungsi sebagai dasar di dalam pembuatan sebuah query.



Gambar 1.5 View

Prosedur Tersimpan

Data basis data hanya dapat diakses dengan menjalankan perintah Transact-SQL. Saat membuat aplikasi yang berfungsi sebagai antarmuka basis data, pengembang dapat memilih untuk membuat program SQL disimpan secara lokal dan dikirim ke server untuk dieksekusi di sana, atau untuk membuat dan memelihara program di server itu sendiri, dalam prosedur tersimpan yang dapat dipicu oleh program di klien. Prosedur tersimpan dapat menerima parameter (nilai yang diteruskan ke prosedur) untuk memprosesnya. Namun, tidak seperti fungsi, prosedur tidak mengembalikan nilai apa pun. Setelah prosedur dibuat, itu dapat digunakan oleh aplikasi apa pun yang memiliki akses basis data. Prosedur dibuat dengan perintah Transact-SQL `CREATE PROCEDURE` dan dimodifikasi dengan perintah `ALTER PROCEDURE`.

Trigger

Pemicu adalah prosedur tersimpan yang dijalankan secara otomatis ketika data dalam tabel berubah karena eksekusi perintah SQL `INSERT`, `UPDATE`, atau `DELETE`. Salah satu kegunaannya yang paling umum adalah untuk menerapkan kendala yang lebih

kompleks daripada yang diperbolehkan oleh kendala CHECK, membatasi informasi yang akan dimasukkan ke dalam kolom. Pemicu dapat dibuat dengan pernyataan INSERT, yang menanyakan tabel lain dan mengembalikan nilai logika yang membatasi data yang ditetapkan ke kolom tertentu. Misalnya, pemicu dapat dibuat untuk menjalankan salinan, sehingga jika satu baris dimasukkan ke dalam basis data Z, baris dengan informasi yang sama akan ditambahkan ke basis data Y. Atau jika satu baris dihapus dari satu tabel, pemicunya akan menjadi dihancurkan di tempat lain. baris yang terkait dengan baris ini di tabel lain. Pemicu dibuat sebagai transaksi dan dapat dibatalkan jika masalah terdeteksi.

Full-Text Index

Indeks khusus ini memungkinkan untuk melakukan kueri berdasarkan kolom yang berisi jenis Varchar dan Teks. Berikut adalah perbedaan utama antara indeks normal dan indeks teks lengkap:

Tabel 1.1 Indeks

Indeks biasa	Indeks full-text
Dibuat dan dihapus melalui perintah SQL.	Dibuat dan dihapus menggunakan prosedur tersimpan.
Mungkin ada beberapa indeks di setiap tabel.	Hanya ada satu indeks teks lengkap per tabel.
Perbarui secara otomatis saat bidang dalam tabel berubah.	Pembaruan hanya berdasarkan permintaan.

Wizard

SQL Server menggunakan wizard untuk melakukan beberapa tugas administratif. Tanpa wizard, tugas ini harus dilakukan melalui perintah Transact-SQL. Wizard menampilkan rangkaian kotak dialog yang berinteraksi dengan pengguna, meminta mereka untuk memasukkan informasi dan kemudian menggunakan informasi tersebut untuk melakukan tugas tertentu. Berikut adalah daftar pembantu utama:

Tabel 1.2 Wizard

Wizard	Fungsi
Konfigurasi penerbitan dan distribusi	Konfigurasi database untuk replikasi.

Create Alert	Jalankan lansiran yang dihasilkan.
Create Backup	Buat cadangan database.
Create Database	Melakukan pembuatan database.
Create Diagram	Melakukan pembuatan diagram database.
Create Index	Lakukan pembuatan indeks.
Create Job	Lakukan pembuatan pekerjaan.
Create New Data Service	Lakukan penginstalan dan konfigurasi sumber data ODBC.
Create Publication	Lakukan pembuatan publikasi untuk disalin.
Create SQL Server Login	Terapkan pembuatan akses login untuk pengguna.
Create Stored Procedures	Terapkan proses pembuatan untuk menambah, menghapus, dan memperbarui baris.
Create Trace	Lakukan pembuatan jejak.
Create View	Jalankan pembuatan tampilan.
Database Maintenance Plan	Lakukan pembuatan file pemeliharaan.
Disable Publishing and Distribution	Jalankan menonaktifkan penerbitan dan penyalinan skema.
DTS Export	Jalankan build ekspor.
DTS Import	Jalankan build impor data.
Full-Text Indexing	Jalankan proses pembuatan indeks teks lengkap.
Index Tuning	Jalankan proses peningkatan stat.
Make Master Server	Jalankan proses pembuatan server inti.
Make Target Server	Jalankan proses pembuatan server target.
Pull Subscription	Jalankan prosedur pemulihan data di server replika.
Push Subscription	Proses yang berjalan mengirimkan data ke server untuk replikasi.
Register Servers Setup	Jalankan proses pendaftaran server SQL.
Setup	Jalankan proses instalasi dan konfigurasi SQL Server.
Web Assistant	Lakukan langkah-langkah build yang diperlukan untuk membuat halaman web berdasarkan konten tabel atau data web yang diimpor.

Kesimpulan

Bab ini memberikan gambaran singkat mengenai arsitektur SQL Server beserta komponen-komponennya. Pada bab-bab berikutnya, kami akan menampilkan cara membuat masing-masing komponen tersebut, dimulai dengan database.

BAB 2

Membuat Database

Membuat data base, terutama struktur dasarnya, adalah pekerjaan yang sangat sederhana pada SQL Server karena SQL Server sudah menggunakan antarmuka grafis. Langkah-langkah yang dilakukan hanya membutuhkan tidak lebih dari dua menit, baik Anda membuatnya secara manual, dengan bantuan wizard, atau dengan perintah Transact-SQL. Membuat database melibatkan penyediaan karakteristik fisik database, seperti ukuran, tingkat pertumbuhan, nama, ID pemilik, dan ID grup.

Ada tiga file yang membentuk database: file utama, sub file dan file log. Berikut penjelasan dari masing-masing file:

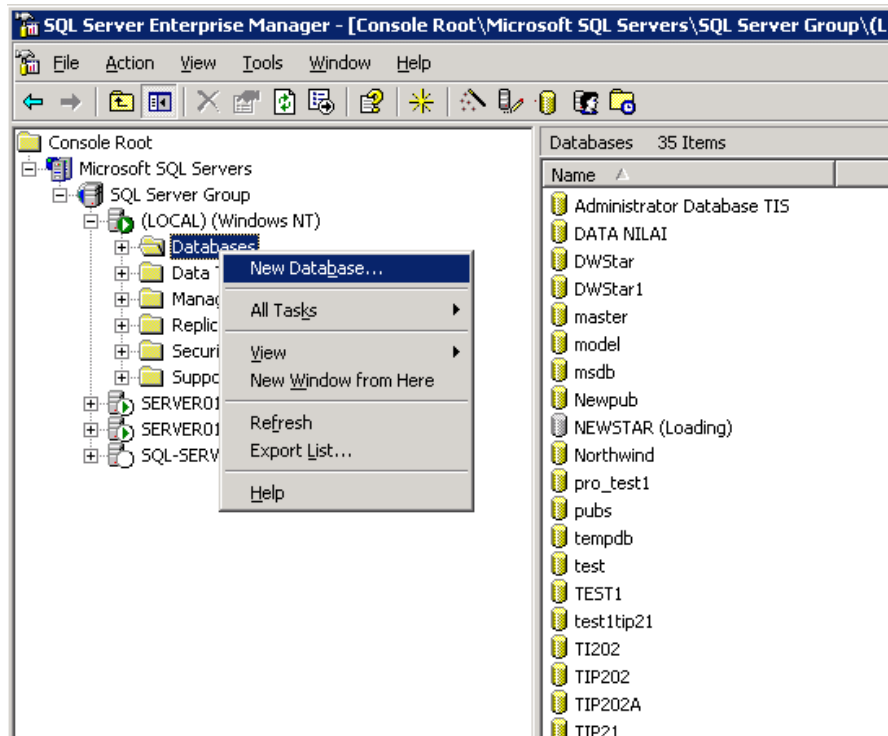
1. **Primer:** File ini berisi informasi yang diperlukan untuk memuat pintasan menu dan menginisialisasi database, serta untuk menyimpan data. Ini adalah file default untuk setiap database yang Anda buat.
2. **Sekunder:** File sekunder hanya ada jika file utama tidak cukup besar untuk menampung semua file data. Bergantung pada ukuran database, Anda mungkin memerlukan lebih dari satu file tambahan ini.
3. **Log:** File log digunakan untuk memulihkan database. Setidaknya ada satu file log di setiap database dan file log kedua selalu dapat dibuat. Ukuran file log minimum adalah 512 KB.

Saat database dibuat, semua filenya diberi nilai nol ditugaskan Misalnya, semuanya diisi dengan nilai nol untuk menghapus semua data yang tersisa di hard drive dengan menghapus file.

Database dapat dibuat secara manual menggunakan Enterprise Manager, Create Database Wizard, atau Transact-SQL.

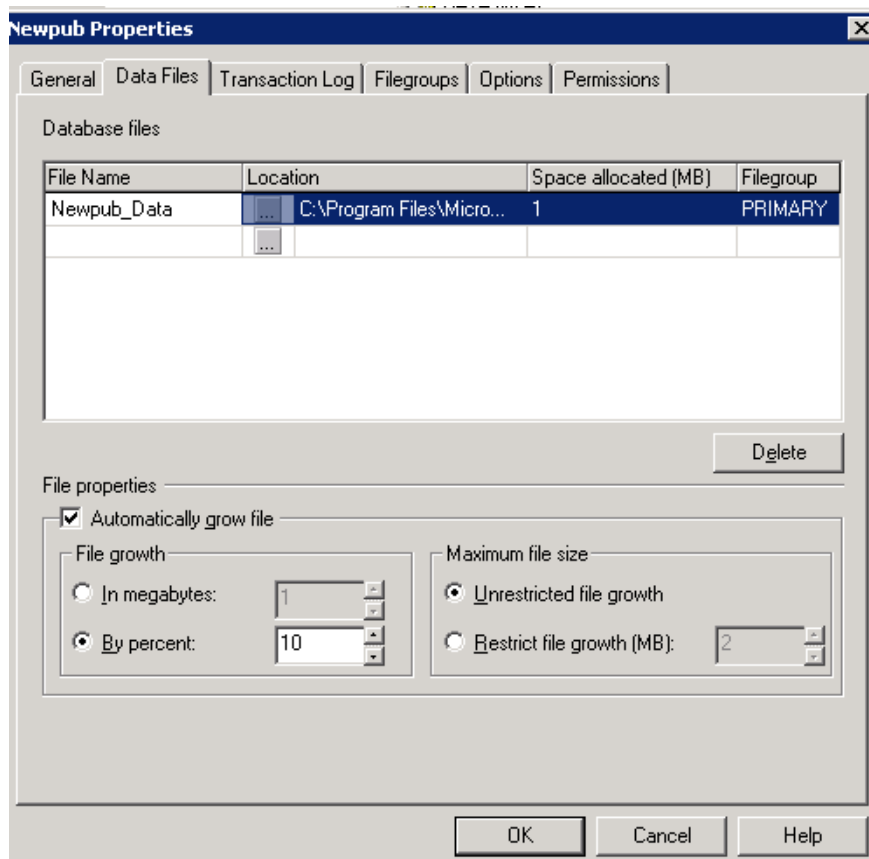
Membuat Database secara Manual

Untuk membuat database secara manual, ikuti langkah-langkah berikut: Di Enterprise Manager, klik ikon Microsoft SQL Server dan buka server. Klik kanan Folder Database kemudian pilih New Database seperti pada Gambar 2.1 di bawah ini:



Gambar 2.1 Database Baru

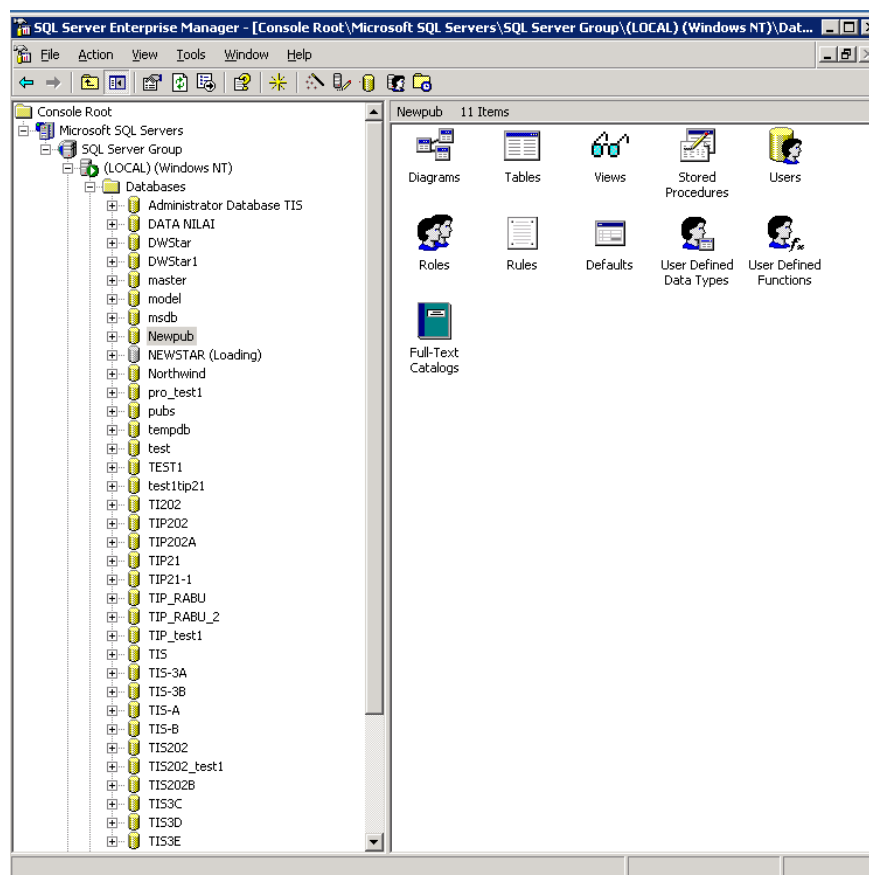
Masukkan nama database ini di kotak dialog yang muncul. Pada contoh ini, ketikkanlah **newpub**.



Gambar 2.2 New Database newpub

File database utama dan log peristiwa dibuat menggunakan nama yang Anda berikan sebagai awalan. Pada contoh kita, nama-nama ini adalah **newpub_data.mdf** dan **newpub.ldf**. Ukuran awal dari file utama dan file log didasarkan kepada database model. Sebelum menekan tombol **OK**, perhatikanlah tab Options.

Di tab Pengaturan Anda dapat menentukan jenis akses ke database. Opsi " DBO Use Only" hanya mengizinkan penulis database untuk menggunakannya. Opsi Single User memungkinkan hanya satu pengguna untuk mengakses database, mencegah akses bersama. Kembalilah ke tab General dan klik OK. Database akan dibuat dengan menggunakan nilai-nilai default.



Gambar 2.3 Database newpub

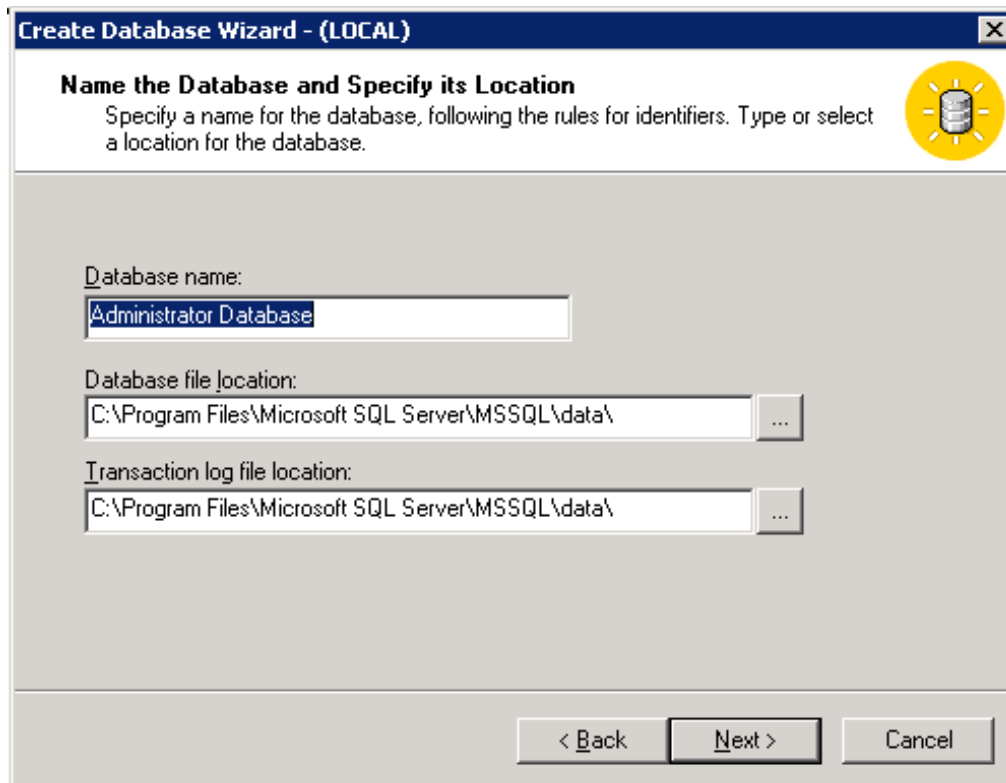
Pada Gambar 2.3, database Newpub yang baru saja dibuat ditampilkan dengan database Northwind dan Pubs.

Membuat Database dengan Wizard

Menggunakan database wizard sangatlah sederhana. Wizard dapat dimulai dengan perintah menu Ekstra | penyihir diaktifkan. Perintah ini membuka jendela di mana Anda dapat

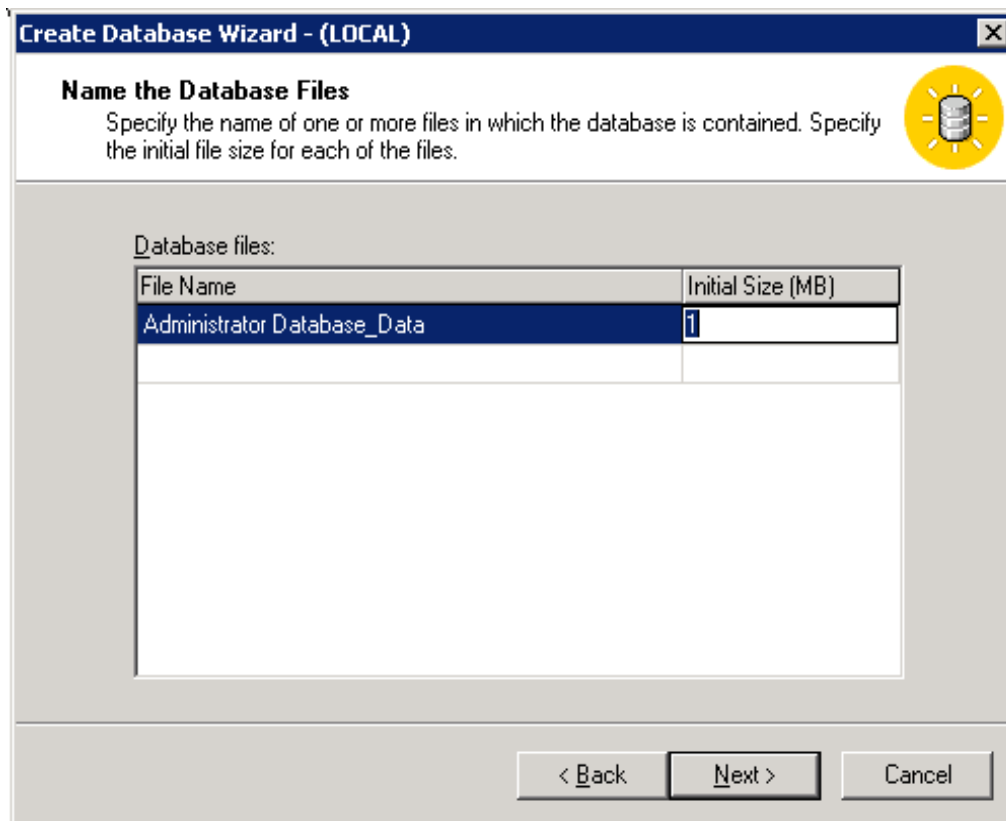
mengklik Database dan kemudian mengklik dua kali Wisaya Buat Database. Anda juga dapat mengaktifkan wizard dengan mengklik kanan folder Databases di panel kiri dan memilih Tools | pilih Asisten dari menu.

Layar pertama wizard hanya berisi informasi. Tekan Berikutnya untuk melanjutkan. Di layar kedua, masukkan nama dan terima atau tentukan lokasi berbeda untuk database sebelum mengklik Berikutnya.



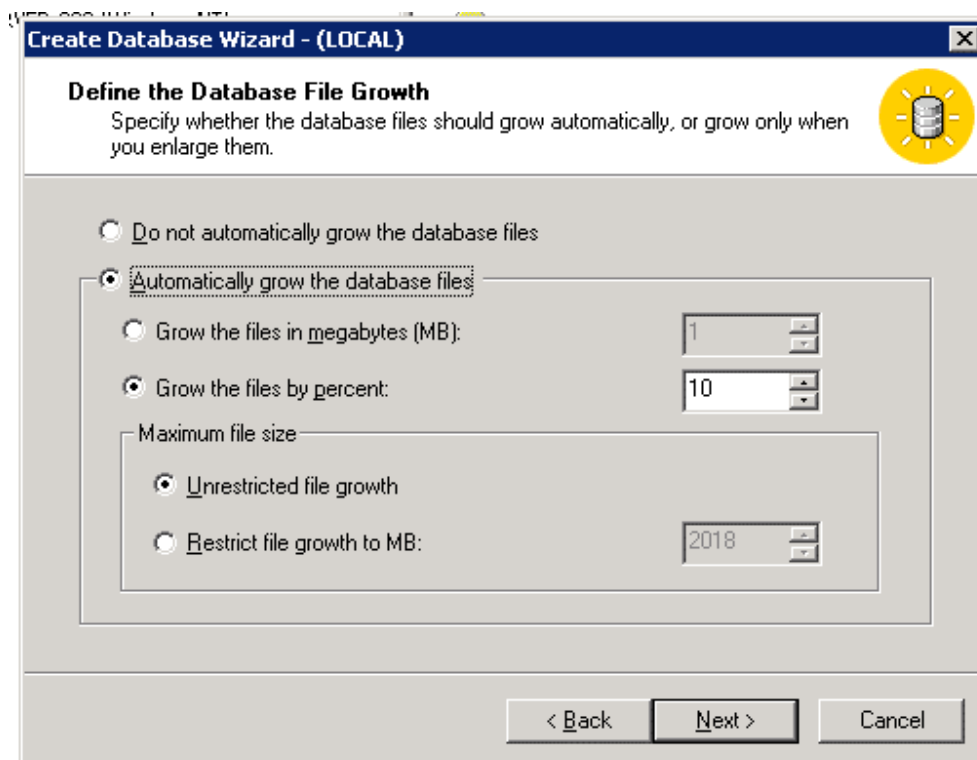
Gambar 2.4 Create Database Wizard

Pada langkah selanjutnya Anda dapat menerima nilai yang diberikan untuk nama dan ukuran file database atau memasukkan ukuran awal file database.



Gambar 2.5 Database initial Size

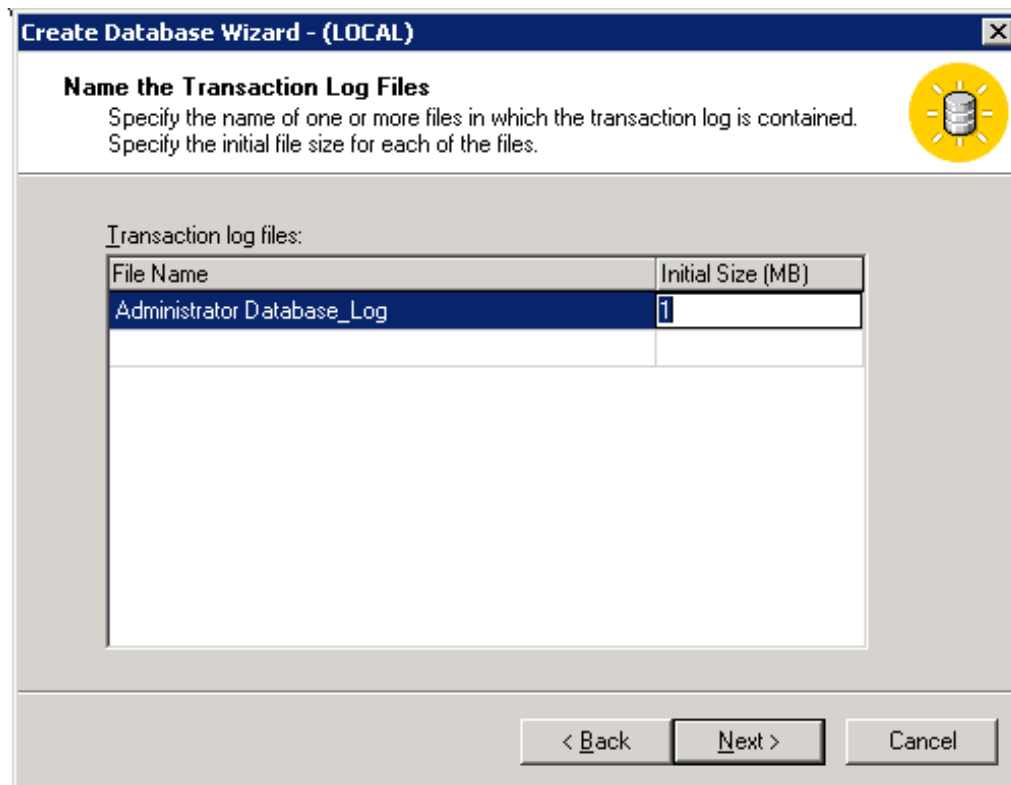
Setelah mengklik Berikutnya, Anda harus memasukkan informasi tentang pertumbuhan file database.



Gambar 2.6 Pertumbuhan file basis data

Secara default, database secara otomatis tumbuh dengan kelipatan 10 persen dari ukuran aslinya. Anda dapat mengubah ini sehingga database bertambah dalam megabita alih-alih dalam persentase dengan memilih opsi Tumbuhkan file dalam megabita. Standarnya adalah menumbuhkan satu megabyte dalam satu waktu. Ukuran maksimum database tidak dibatasi kecuali oleh ruang disk fisik yang tersedia. Jika Anda ingin menentukan ukuran maksimum, Anda harus memilih Batasi pertumbuhan file dan tentukan ukuran akhir yang dapat dicapai database. Tekan Berikutnya.

Pada langkah selanjutnya, seperti yang dijelaskan di halaman berikutnya, Anda harus menentukan ukuran file log peristiwa. Wizard memberikan nilai default sebesar 1 MB. Tekan Berikutnya dan ulangi proses ini dengan file log.



Gambar 2.7 Log file database

Layar akan meminta Anda untuk menentukan jenis pertumbuhan dan ukuran file maksimum. Masukkan informasi sesuai keinginan. Dalam contoh ini kita masih mendapatkan nilai yang dikembalikan. Klik Next dan sebuah jendela akan muncul meringkas properti dari database yang akan dibuat, seperti yang ditunjukkan di bawah ini.



Gambar 2.8 Completing the create database wizard

Klik Finish untuk membuat database. Wizard kemudian menanyakan apakah Anda ingin membuat rencana pemeliharaan untuk database saat ini. Jika Anda menjawab ya, rencana pemeliharaan basis data terpandu akan dimulai.

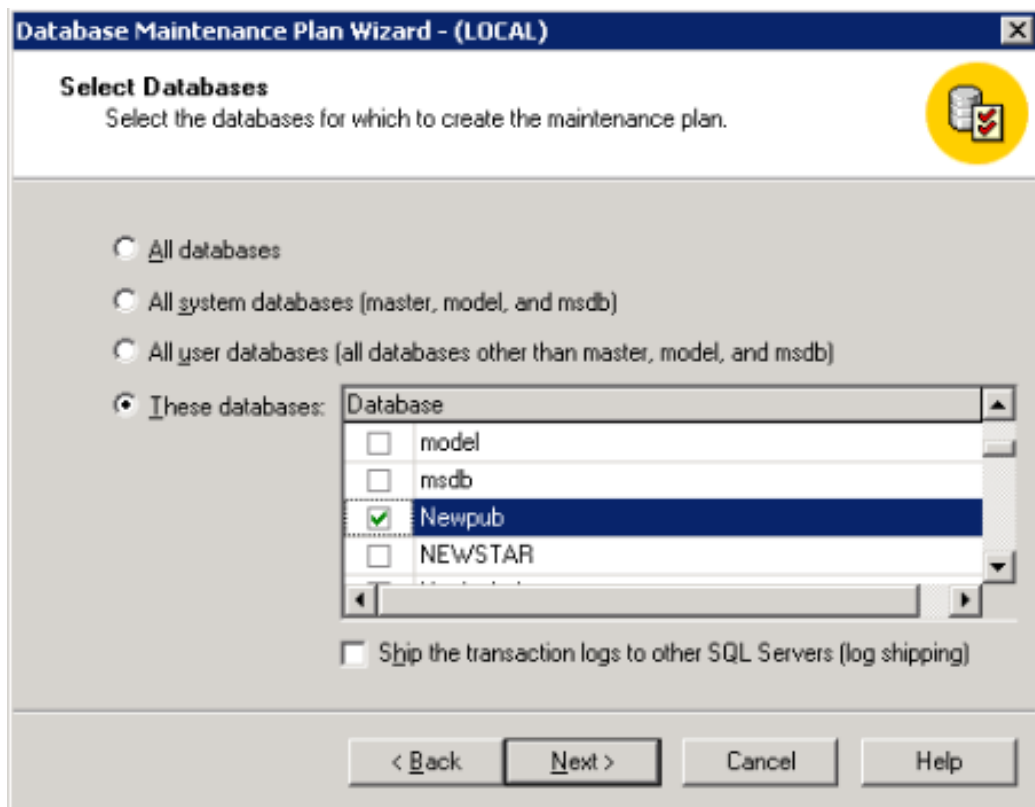
Menentukan Rencana Pemeliharaan Untuk Database

Setelah database dibuat, Anda harus membuat rencana pemeliharaannya. Melalui wizard, Anda bisa menentukan beberapa tugas pemeliharaan preventif yang ditujukan untuk meningkatkan kinerja atau keamanan dari database. Tugas-tugas seperti pemeriksaan integritas dari file dan indeks, frekuensi backup, serta pembuatan laporan adalah beberapa tugas yang akan dijalankan secara otomatis oleh rencana pemeliharaan. Bagian ini akan menjelaskan cara menentukan pemeliharaan database yang baru saja kita buat.

Database Maintenance Plan Wizard

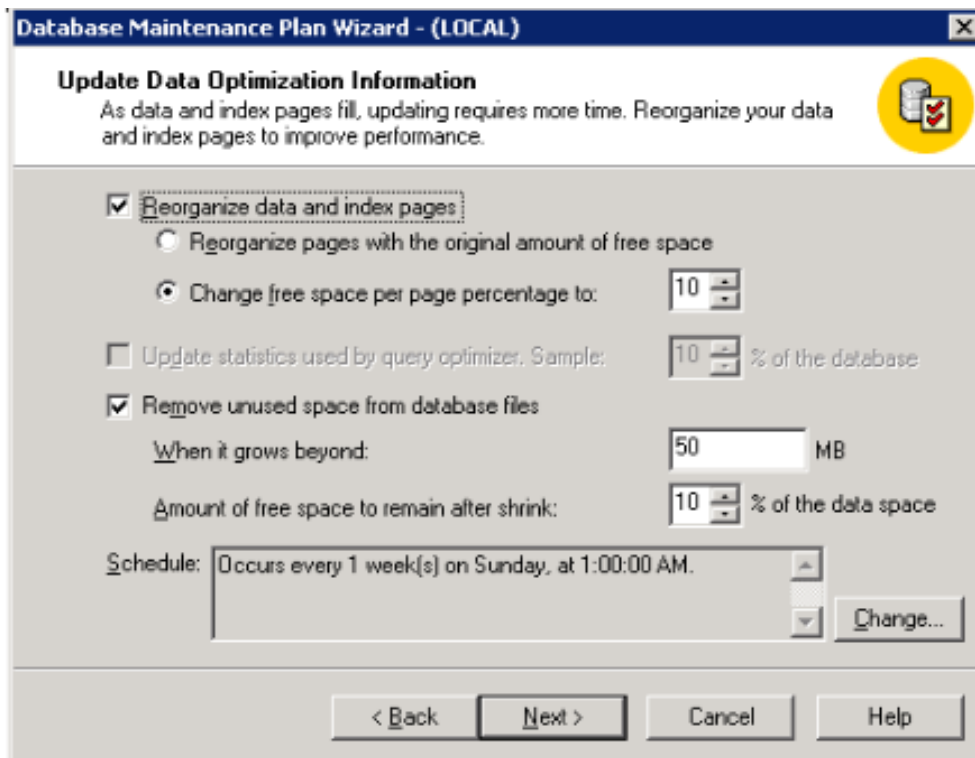
Layar pertama wizard hanya berisi informasi informasional. Setelah menekan Next, Anda akan melihat layar di mana Anda harus mengetikkan nama dari database yang akan menjadi bagian dari rencana pemeliharaan. Database Newpub2 dipilih secara default, karena wizard yang membuatnya menjalankan Maintenance Wizard. Di sini Anda bisa memilih All Databases untuk menyertakan semua file, termasuk database sistem, di dalam rencana

pemeliharaan. Anda juga bisa memilih All System Databases atau All User Data bases.



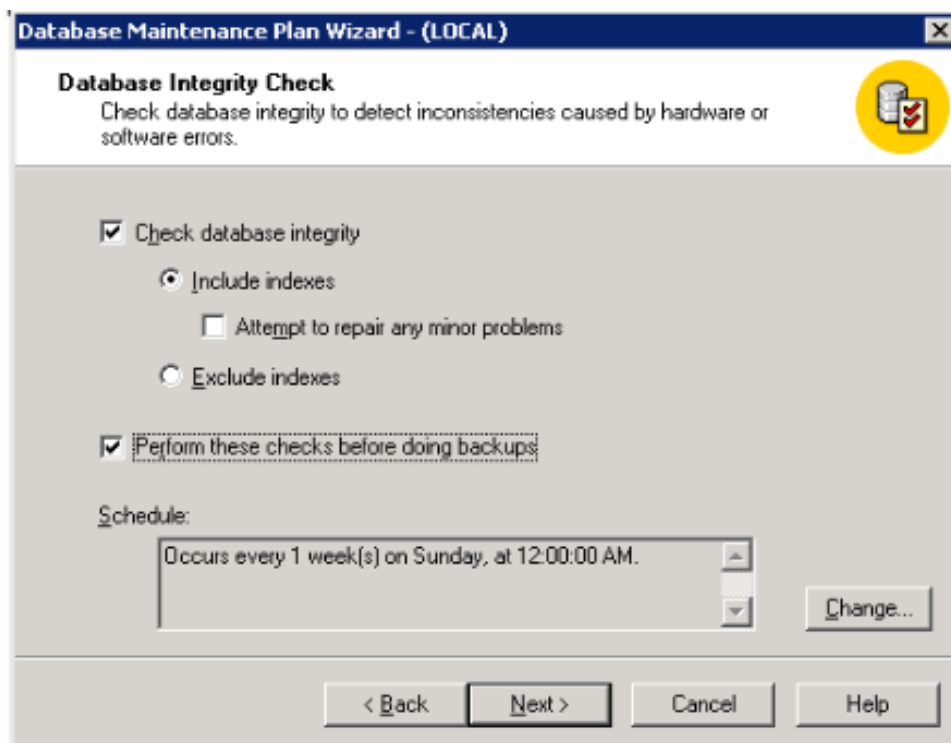
Gambar 2.9 Select Database

Apabila Anda menekan Next, akan muncul layar optimisasi update. Pilihan di dalamnya membantu Anda mengorganisir ulang data dan indeks untuk meningkatkan kinerja. Pastikan semua kotak dipilih untuk contoh ini. Lalu tekan Next.



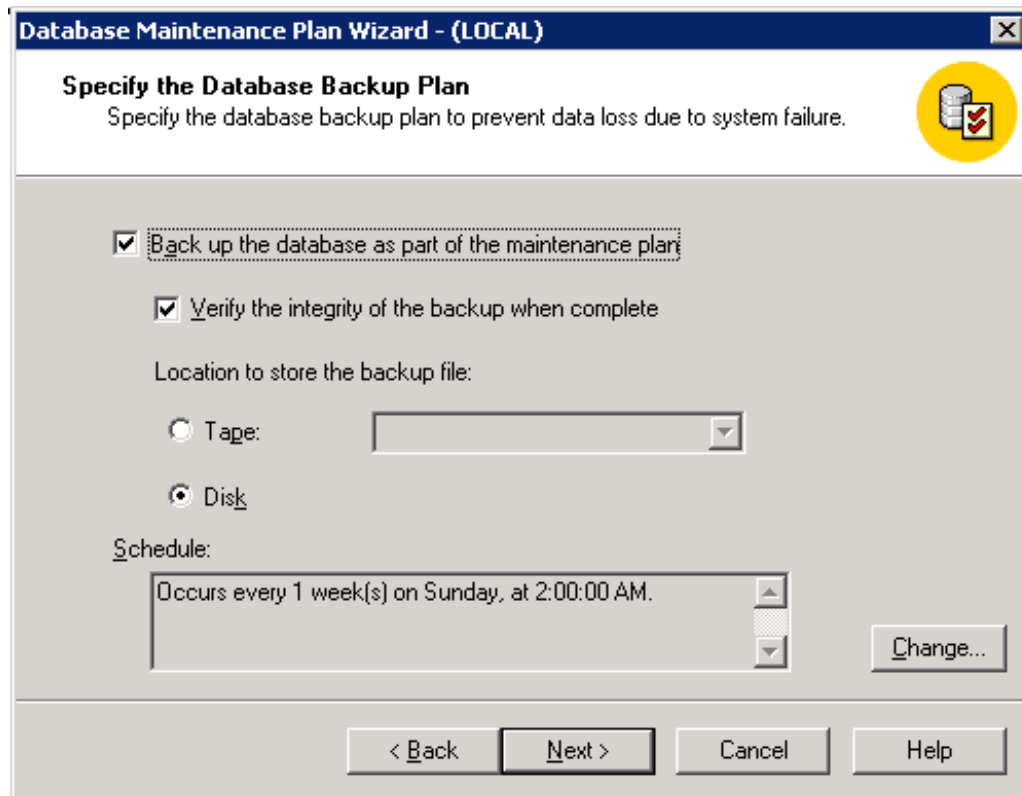
Gambar 2.30 Optimisasi Update

Pada langkah berikutnya, tentukanlah uji integritas data yang akan dilakukan. Pilih tes-tes yang sesuai dan tentukan pada kotak Schedule kapan tes-tes tersebut harus dilakukan, seperti ditunjukkan di bawah ini.



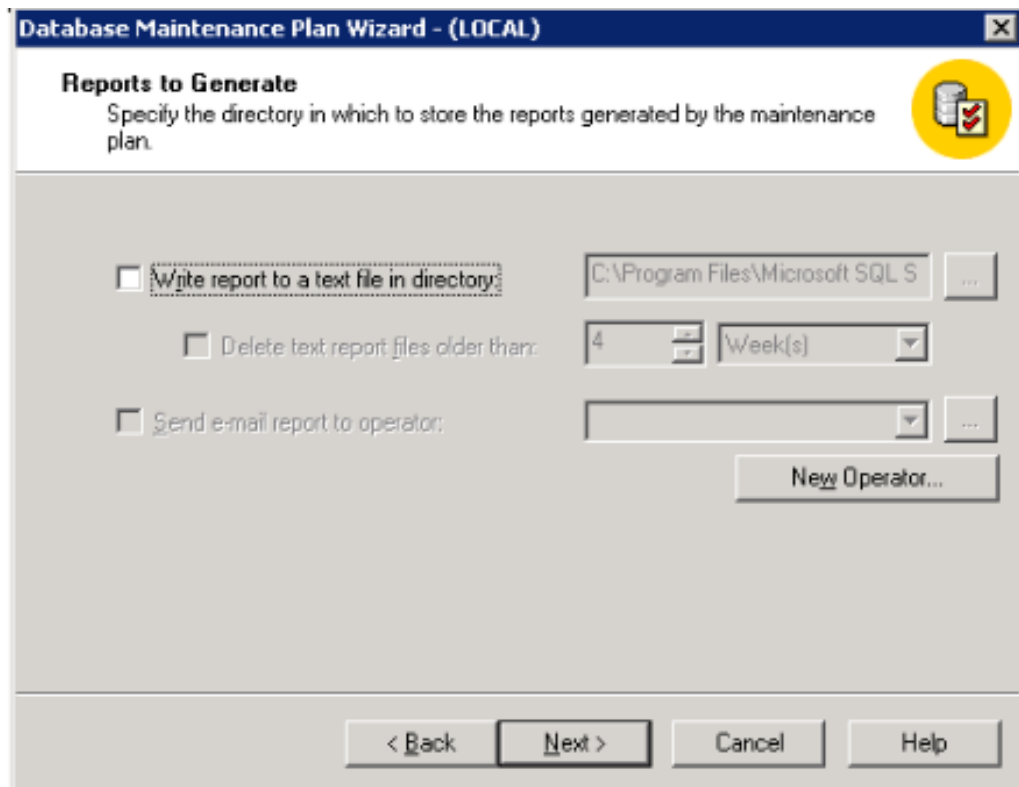
Gambar 2.31 Uji Integritas Data

Tekan Next. Pada layar berikutnya, wizard ingin mengetahui apakah ia bisa menjalankan program backup sebagai bagian dari rencana pemeliharaan. Anda harus menentukan konfigurasi backup pada langkah ini.



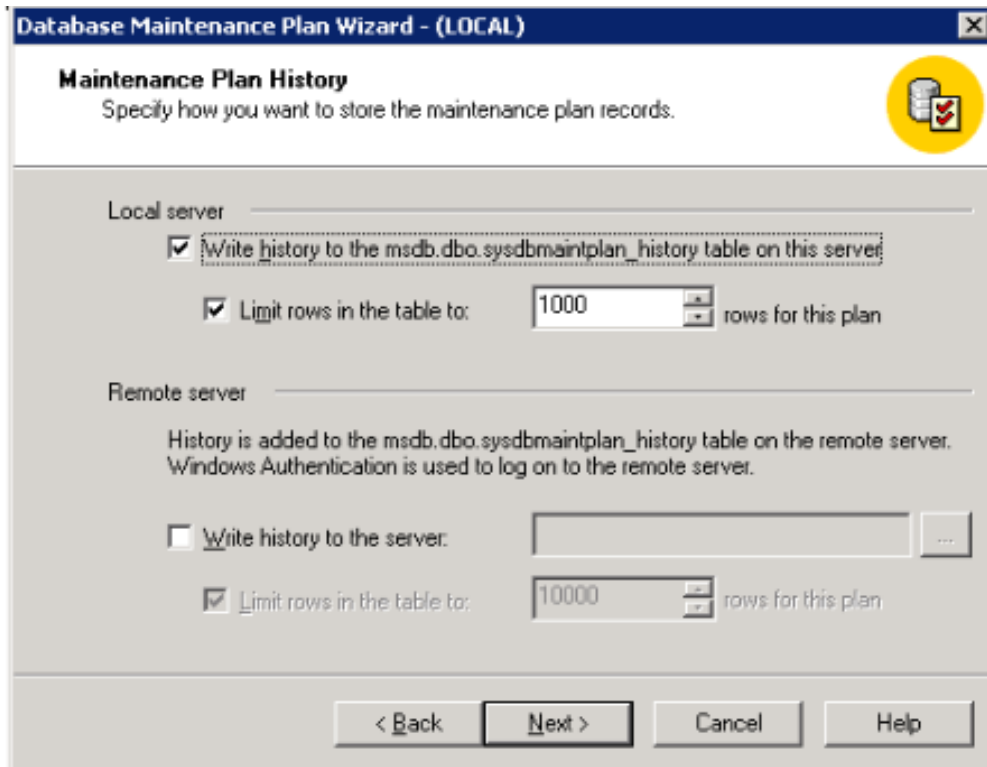
Gambar 2.32 Backup

Tekan Next. Sebuah layar akan muncul, serupa dengan layar sebelumnya yang berisikan eksekusi dari backup file log. Tekan Next. Pada layar berikutnya, seperti gambar di bawah ini, akan meminta otorisasi untuk menghasilkan laporan.



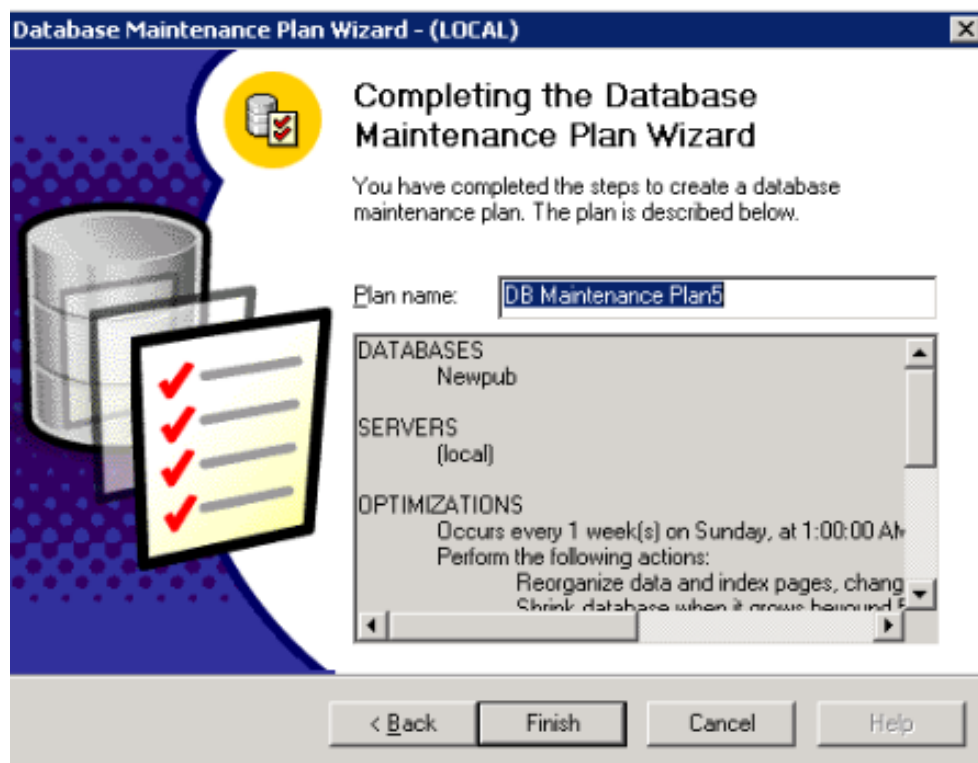
Gambar 2.33 Laporan

Jika Anda ingin menghasilkan laporan, pilihlah jenis laporan dan tentukan di mana laporan tersebut harus dibuat, serta seberapa sering laporan itu harus dihapus . Berikutnya, wizard ingin mengetahui apakah sejarah pemeliharaan harus disimpan di dalam server di mana pemeliharaan dilakukan atau disimpan pada server lain. Jika Anda memilih server lain, ketikkanlah lokasinya. Pada setiap pilihan, Anda bisa membatasi jumlah baris untuk tabel pemeliharaan.



Gambar 2.34 History

Tekan Next. Layar terakhir akan muncul, menampilkan ringkasan dari operasi-operasi yang akan dilakukan. Di sini Anda bisa mengubah nama untuk rencana pemeliharaan atau menerima nama yang telah diberikan.



Gambar 2.35

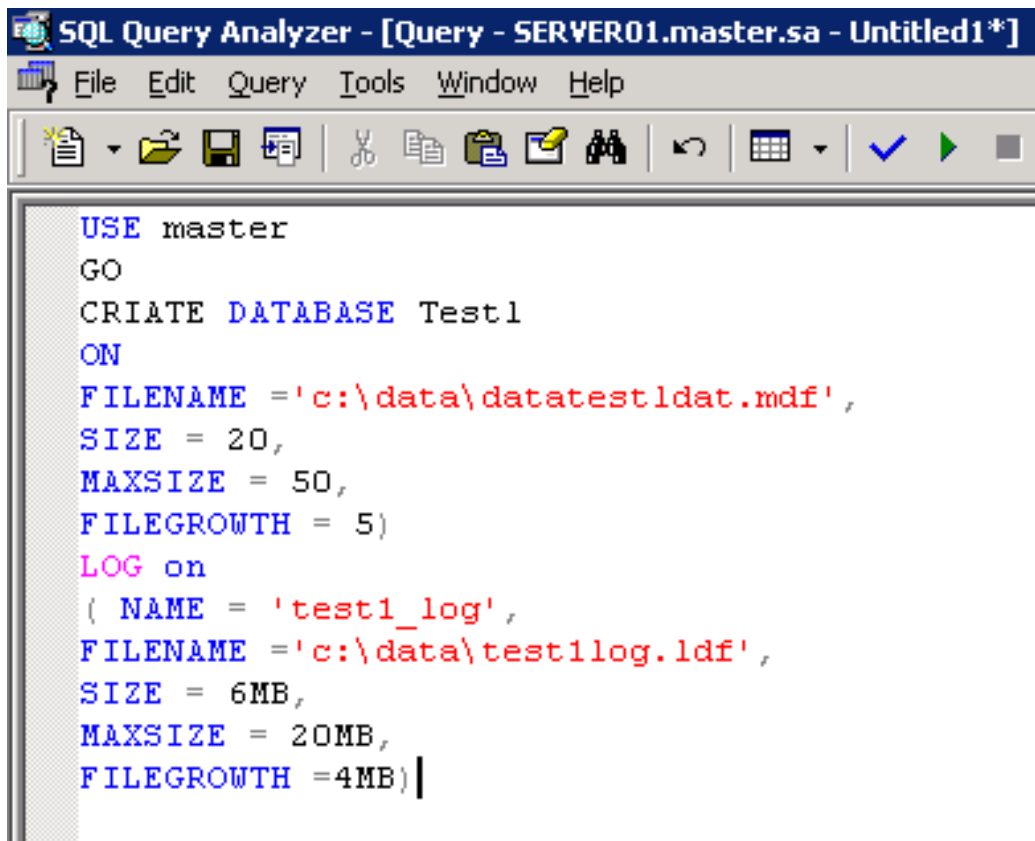
Tekan Next. Layar terakhir akan muncul, menampilkan ringkasan dari operasi-operasi yang akan Anda lakukan. Apabila Anda menekan tombol Finish, rencana pemeliharaan akan dibuat dengan karakteristik yang telah Anda pilih di atas.

Membuat Rencana Pemeliharaan Lain

Database Maintenance Plan Wizard bisa dijalankan kapan saja melalui menu Tools | Wizard. Apabila kotak Wizard dibuka, klik Management dan pilih Database Maintenance Plan Wizard. Anda bisa membuat beberapa rencana pemeliharaan berdasarkan pada kebutuhan dan karakteristik dari database Anda.

Membuat Database Menggunakan Transact-SQL

Database juga bisa dibuat dengan menggunakan program CREATE DATABASE. Di bawah ini adalah contoh dari perintah untuk membuat database bernama Test1. File primer bernama Test1_dat. Ukuran awalnya 20 MB dan ukuran maksimalnya 50 MB, dengan kenaikan pertumbuhan sebesar 5 MB. File log memiliki ukuran awal 6 MB dan ukuran maksimal 20 MB, dan akan tumbuh dengan kenaikan 4 MB.



```
USE master
GO
CREATE DATABASE Test1
ON
FILENAME = 'c:\data\datatest1dat.mdf',
SIZE = 20,
MAXSIZE = 50,
FILEGROWTH = 5)
LOG on
( NAME = 'test1_log',
FILENAME = 'c:\data\test1log.ldf',
SIZE = 6MB,
MAXSIZE = 20MB,
FILEGROWTH = 4MB)
```

Gambar 2.36 Create Database

Mengubah Karakteristik Database

Setelah database dibuat, Anda bisa mengubah karakteristiknya melalui kotak dialog Properties. Untuk mengaktifkan kotak dialog ini, klik kanan pada nama dari data base lalu pilih Properties pada menu.

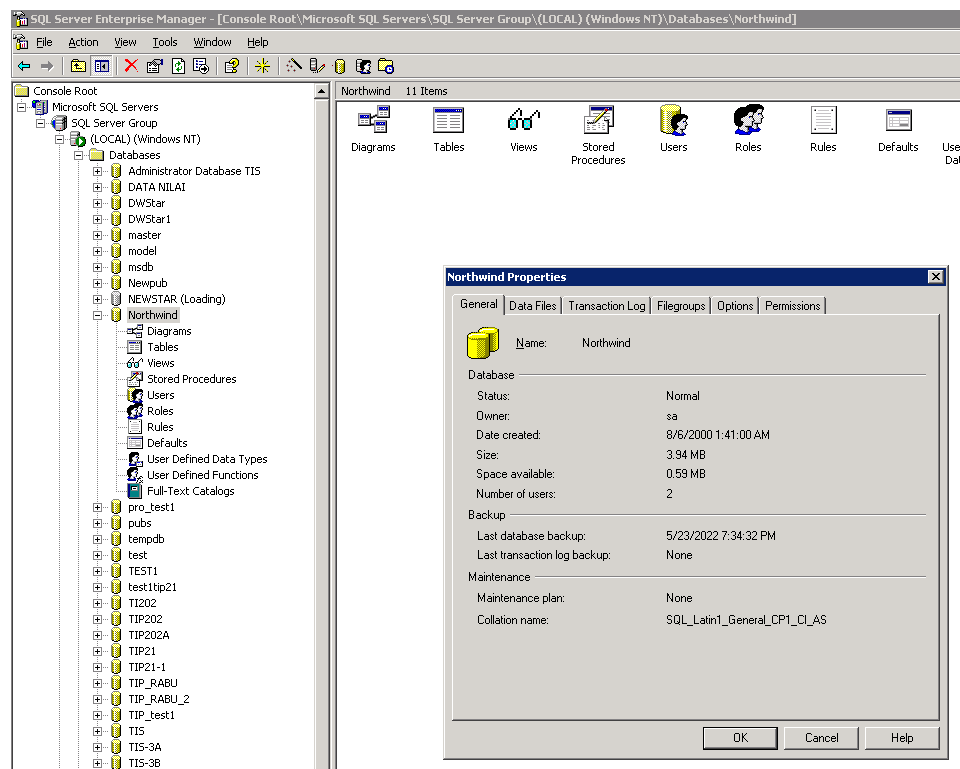
Kotak dialog ini sama seperti yang digunakan untuk membuat database, tetapi dengan tambahan tab Permissions. Tab ini berfungsi untuk memberikan atau mencabut hak-hak database untuk user yang terdaftar.

Menghapus Database

Untuk menghapus database dan semua isinya, aktifkan menu di sebelah kanan dengan mengklik nama database dan pilih Hapus. Kotak dialog konfirmasi akan muncul. Jika Anda memilih Ya, database akan dihapus dan tidak dapat dipulihkan.

Melihat Komponen Database

Setelah database dibuat, Anda bisa melihat karakteristik fisiknya, seperti misalnya ukuran file dan ukuran log, atau jumlah tabel dan baris pertabel, melalui panel pemantau.

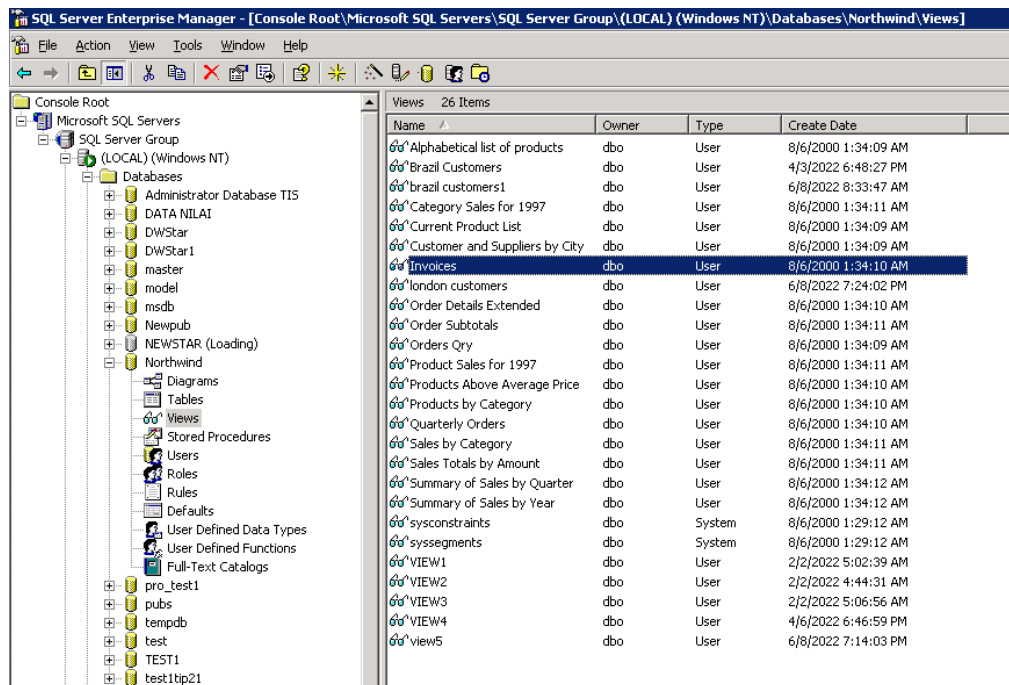


Gambar 2.37 Komponen Database

Untuk melihat informasi ini, klik ganda pada nama database. Sebagai contoh, kami akan

melakukan operasi ini dengan file Northwind. Pada panel kanan, baris yang terdapat di bagian atas menampilkan lima buah topik yaitu: General, File Space, Log Space, Table/Index Space, dan Database Diagram. Apabila Anda mengklik salah satunya, informasi yang sesuai akan muncul. Pada gambar di atas, tabel Northwind ditampilkan dengan field, jumlah baris, dan ukurannya.

Untuk melihat komponen tertentu dari database, klik pada folder yang sesuai pada pohon lalu klik pada item di panel keterangan. Sebagai contoh, untuk mengakses isi dari sebuah view, Anda harus mengklik pada folder SQL Server Views. Jika database sudah memiliki view yang dibuat sebelumnya, view tersebut akan ditampilkan pada panel kanan, seperti ditunjukkan pada gambar 2.38.



Gambar 2.38 View

Untuk membuka, mengedit, atau melihat salah satu tampilan, klik kanan nama tampilan. Menu konteks menampilkan semua opsi yang tersedia untuk item ini. Prosedur ini sama untuk semua objek database.

Kesimpulan

Bab ini menampilkan cara membuat database secara manual, menggunakan wizard untuk membuat database dan perintah Transact-SQL. Di sini juga dibahas cara menentukan rencana pemeliharaan. Bab berikutnya akan membahas mengenai tabel.

BAB 3

Tabel

Tabel adalah jantung dari database. Mengetahui cara membuat dan memelihara tabel adalah pengetahuan yang paling penting bagi para user database. Untuk membuat tabel, Anda harus mengetahui konsep dan karakteristik dari database yang digunakan.

Proses membuat tabel harus dimulai hanya setelah database didefinisikan dengan seksama. Banyak pengembang yang memiliki kebiasaan buruk dengan mendefinisikan database secara "online", misalnya seperti membuka modul yang membuat tabel lalu mulai memikirkan tentang database. Hal ini bisa menyebabkan banyak masalah.

Karakteristik dari Tabel

Dengan SQL Server Anda dapat membuat tabel permanen dan tabel sementara. Tabel sementara adalah tabel yang dibuat di memori sedangkan tabel permanen adalah tabel yang ditulis ke disk.

Tabel SQL Server dapat berisi hingga 1024 kolom. Setiap kolom harus dibuat dengan menentukan informasi dasarnya seperti nama, jenis data yang akan disimpan, definisi tambahan dari nilai default, dan batasan yang diterapkan pada kontennya.

Nama kolom harus unik untuk database. Jadi tabel yang sama tidak boleh memiliki dua kolom bernama status. Namun, tabel yang berbeda dalam database mungkin berisi kolom dengan nama yang sama. Sebagai contoh, dua tabel bernama Clients dan Suppliers masing-masing bisa mengandung data informasi alamat di dalam kolom bernama Street Address, City, dan State.

Nama tabel harus unik di antara tabel yang dibuat pemilik. Namun, database mungkin berisi tabel dengan nama yang sama yang dimiliki oleh pengguna yang berbeda.

Filegroup

Saat tabel dibuat, secara otomatis diasosiasikan dengan filegroup default. Namun, jika tabel berisi bidang bertipe Teks, Ntext, atau Gambar, mereka dapat ditempatkan di grup file yang berbeda dari tabel lainnya.

Tipe Data

Setiap kolom dalam tabel harus memiliki tipe data tertentu. Dengan kata lain, Anda harus menentukan bahwa kolom menyimpan jenis data tertentu, seperti teks, angka, tanggal, atau gambar. SQL Server dapat menggunakan beberapa tipe data dan juga memungkinkan pengguna untuk membuat tipe data mereka sendiri.

Data yang disimpan dalam tabel SQL Server dapat berupa tipe berikut:

Integer	
Bit	Integer dengan nilai 0 atau 1.
Int	Nilai integer dengan nilai antara -2^{31} (-2.147.483.648) sampai $2^{31}-1$ (2.147.483.647).
Smallint	2^{15} (-32.768) sampai $2^{16}-1$ (32.767).
Tinyint	Integer antara 0 dan 255.
Decimal Atau Numeric	Angka dengan akurasi tetap antara -10^{38} sampai $10^{38}-1$.
Money	Nilai data moneter dari -2^{63} (-922.377.203.685.477,5808) sampai $2^{63}-1$ (922.377.203.685.477,5807), dengan tingkat akurasi hingga seperseratusribu dari unit moneter.
Smallmoney	-274.748,3648 sampai 274.748,3647, dengan tingkat akurasi hingga seperseratusribu dari unit moneter.
Float	-1.79E+308 sampai 1.79E+308.
Real	-3.40E+38 sampai 3.40E+38.
Datetime	1 Januari 1753 sampai 31 Desember 9999, dengan akurasi sampai 3,33 milidetik.
Smalldatetime	1 Januari 1900 sampai 6 Juni 2079, dengan ketelitian hingga 1 menit.

Rupa rupa

Cursor	Referensi ke cursor.
Timestamp	Angka eksklusif yang dikenali oleh database.
Uniqueid	Pengenal global yang eksklusif.

String

Char	Field tetap dengan ukuran maksimal sebesar 8000 byte.
Varchar	Field tetap dengan ukuran maksimal sebesar 8000 byte.
Text	Variabel dengan ukuran hingga $2^{31}-1$ (2.147.483.647) byte.

Unicode String

Nchar	Karakter unicode dengan ukuran tetap hingga 4000 byte.
Nvarchar	Karakter unicode dengan ukuran bervariasi hingga 4000 byte.
Ntext	Variabel dengan ukuran sampai $2^{30}-1$ (1.073.741.823) byte.

Binary String

Binary	Ukuran tetap hingga 8000 byte.
Varbinary	Ukuran bervariasi hingga 8000 byte.
Image	Ukuran bervariasi hingga 2^{31-7} (2.147.483.647) byte.

Sinonim

Beberapa sinonim telah diciptakan agar kompatibel dengan bahasa SQL-92.

Sinonim	Setara dengan
binary varying	varbinary
char varying	varchar
character	char
character	char(l)
character(n)	char(n)
character varying(n)	varchar(n)
dec	decimal
double precision	float
float[(n)] for n = 1 - 7	real
float[(n)] for n = 8 - 15	float
integer	int
national character(n)	nchar(n)
national char(n)	nchar(n)
national character varying(n)	nvarchar(n)
national char varying(n)	nvarchar(n)
national text	ntext
numeric	decimal

Batasan

Kendala adalah cara untuk memastikan integritas database dengan memfilter data yang dimasukkan ke dalam kolom tabel. Batasan adalah mekanisme mendasar untuk memastikan integritas lapangan. Kendala lebih diutamakan daripada pemicu, aturan, dan default. Ada lima macam limit, yaitu:

NOT NULL menunjukkan bahwa kolom tidak dapat dibatalkan.

CHECK membatasi nilai yang dapat dituliskan ke kolom saat menguji suatu kondisi. Nilai Boolean didasarkan pada konten yang ditetapkan ke kolom. Jika kondisi yang akan diuji adalah BENAR, kolom dapat diberi nilai yang dimasukkan.

UNIQUE memaksa kolom untuk menggunakan nilai unik. Dengan kata lain, kolom kendala **UNIQUE** tidak dapat memiliki dua catatan dengan konten yang sama. Sebagai contoh, kode produk atau nomor kesejahteraan harus bersifat eksklusif. Di sisi lain, isi dari field negara

bagian atau kota bisa digandakan.

PRIMARY KEY membuat tabel, kolom, atau kombinasi kolom dengan kunci utama atau nilai kunci utama yang harus unik dalam tabel untuk mengidentifikasi baris. Kolom dengan batasan kunci primer tidak boleh berisi nilai nol. Jika primary key terdiri dari lebih dari satu kolom, hanya salah satunya yang dianggap sebagai primary key. Kolom lain yang menyusun kunci disebut kunci kandidat.

FOREIGN KEY mendefinisikan hubungan antar tabel. Ini adalah bidang di tabel aktif yang menunjuk ke bidang kunci di tabel lain. Menggunakan kunci asing dapat mencegah baris dihapus dari tabel aktif saat ada referensi ke bidang kunci dari tabel eksternal. Pembatasan seperti itu mencegah penyisipan nilai ke dalam tabel yang tidak ada di bidang kunci tabel bersama eksternal.

Tabel Contoh

Pada bab ini, kita akan membuat sebuah tabel dengan data penulis, serta sebuah tabel dengan data buku, dan tabel ketiga yang akan digunakan untuk menggabungkan tabel penulis dan buku. Tabel berikut dibuat dalam database Newpub. Tabel Authors harus memiliki struktur seperti yang ditunjukkan pada halaman berikut ini, dengan kolom au_id sebagai kunci primer, dengan nilai eksklusif yang didefinisikan oleh ID.

Column	Name	Type	Size	Null?	Default
1	au_id	Int	10	no	
2	au_name	varchar	30	no	
3	au_phone	varchar	18	yes	
4	au_email	varchar	25	yes	
5	au_addr	varchar	40	yes	
6	au_city	varchar	30	yes	
7	au_state	char	2	yes	
8	au_zip	char	8	yes	
9	au_country	char	15	yes	'Brazil'
10	active	bit	1		

Field Active menunjukkan apakah penulis pernah menerbitkan buku.

Tabel Books harus mengandung field-field seperti di bawah ini. Kolom book_id harus menjadi kunci primer dengan nilai eksklusif yang didefinisikan oleh ID.

Column	Name	Type	Size	null?	Default value
1	book_id	ID	4	no	
2	title	varchar	50	no	
3	price	money	8	no	
4	category	char	10	no	
5	comments	varchar	100	yes	'not available'
6	publiday	datetime	8	yes	

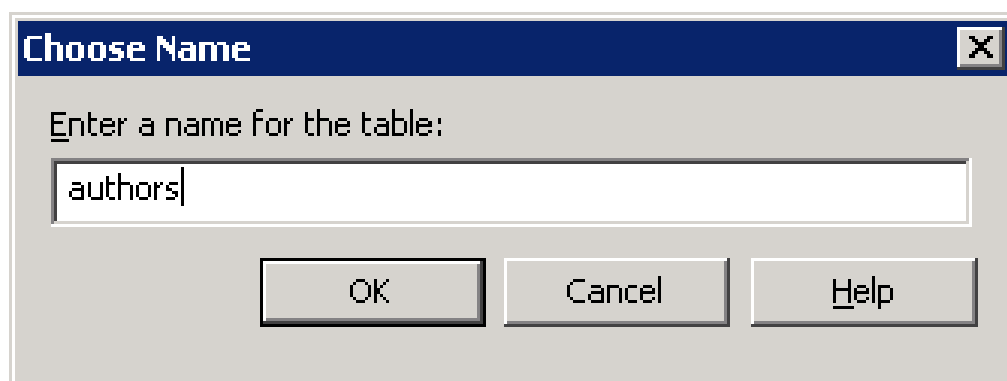
Tabel Autbook harus mengandung struktur berikut ini, dan kolom book_id harus menjadi kunci primer:

Column	Name	Type	Size	Null?
1	book_id	ID	4	no
2	au_id	ID	10	no

Membuat Tabet Dengan Enterprise Manager

Ikutilah langkah-langkah berikut ini untuk membuat tabel:

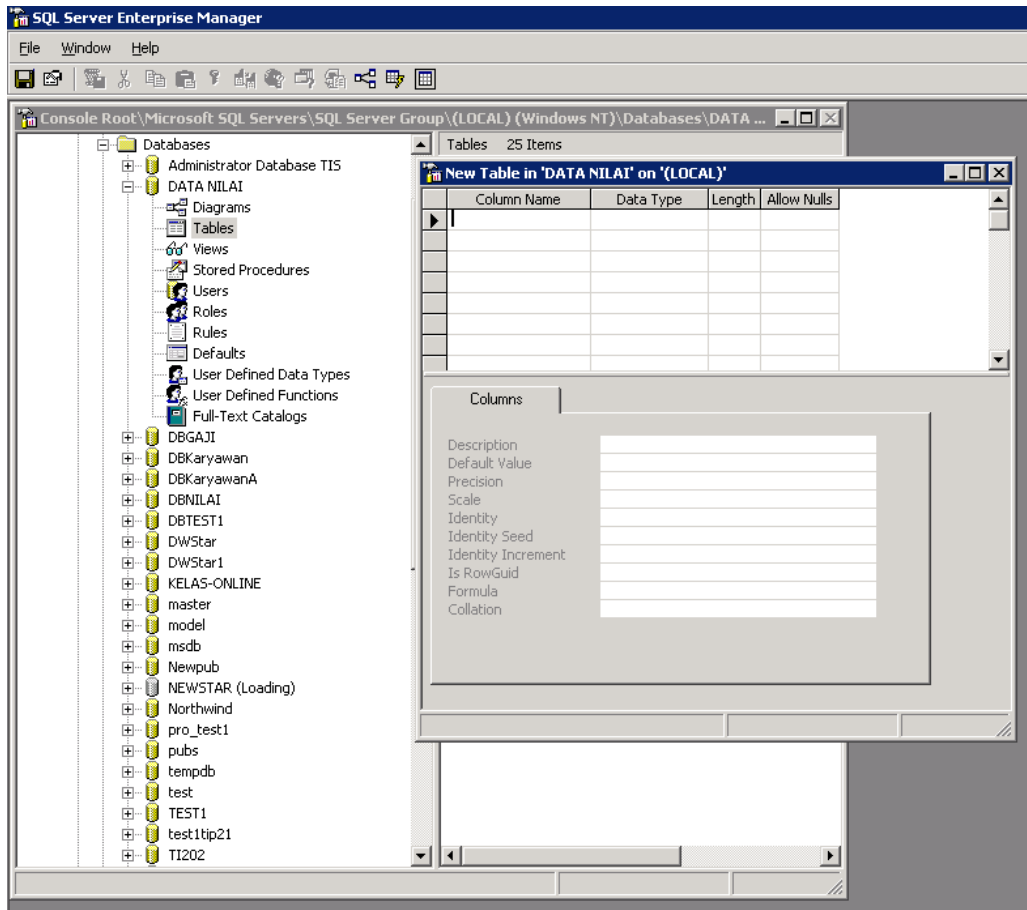
Luaskan folder Database dan database tempat Anda ingin membuat tabel. Kemudian klik kanan folder Tabel dan pilih Tabel Baru. Dialog yang ditunjukkan di bawah ini menanyakan nama tabel. Dalam contoh ini, tuliskan faktor-faktornya. Tekan OK. Kemudian, sebuah jendela terbuka di mana Anda dapat menentukan properti kolom.



Gambar 3.1 Nama table

Karakteristik Kolom

Jendela berikut ini harus digunakan untuk menentukan karakteristik dari kolom-kolom di dalam tabel yang kita buat:



Gambar 3.2 Karakteristik Kolom

Berikut adalah penjelasan setiap kolom dalam jendela tersebut:

- Column Name Menentukan nama kolom atau bidang.
- Datatype Menunjukkan jenis data yang disimpan oleh kolom.
- Length Menentukan ukuran bidang.
- Precision Menentukan jumlah digit dalam angka.
- Scale Menunjukkan jumlah pecahan.
- Allow Nulls Saat kolom ini dipilih, SQL Server mengizinkan nilai null untuk kolom tersebut.
- Default Value Digunakan untuk menetapkan nilai default untuk kolom.
- Identify Jika dipilih, kolom ini menunjukkan bahwa bidang tersebut memiliki nomor identifikasi yang dihasilkan secara otomatis.
- Identity Seed Menentukan nilai awal dari record pertama.
- Identity Increment Menentukan nilai insert dari record baru.
- Is RowGuid Membuat teks unik di tingkat kolom global. Sebuah tabel hanya dapat memiliki satu kolom dari tipe ini dan harus dibuat dengan tipe

data Uniqueidentifier.

Untuk field numerik, Length menunjukkan jumlah byte yang digunakan untuk menyimpan semua angka. Untuk field karakter atau Unicode, ini adalah jumlah karakter. Untuk field Binary, Varbinary, dan Image, ini menunjukkan jumlah byte .

Kolom Pengenal

Tabel dapat berisi kolom ID. Kolom jenis ini hanya berisi satu nilai yang mengidentifikasi baris dalam tabel. Jenis bidang ini ideal untuk membuat kode rekaman secara otomatis. Sebagai contoh, sebuah tabel berisikan pesanan penjualan bisa mengandung sebuah field bernama ped_num yang menyimpan hanya satu pengenal dalam urutan pesanan. Melalui properti Identity, SQL Server menghasilkan nilai yang menaik untuk setiap baris yang ditambahkan ke dalam tabel. Anda bisa menentukan nilai awal (seed) dan nilai yang akan digunakan untuk menaikkan counter ini. Apabila tidak ditentukan, angka 1 akan digunakan sebagai nilai default. Hanya satu kolom per tabel yang bisa memiliki properti ini. Tipe data dari kolom haruslah Int atau Smallint; ia tidak mengizinkan nilai Null.

Nilai Null

Beberapa tabel bisa mengandung kolom-kolom yang menerima nilai Null, menunjukkan bahwa kolom tersebut belum menerima isi apa pun. Field yang harus diisi dibuat dengan mematikan pilihan Allow Nulls.

Kunci Primer

Satu atau lebih kolom pada tabel yang memiliki baris yang eksklusif di dalam tabel akan membentuk kunci primer. Spesifikasi dari kunci primer memastikan integritas dari tabel. Kolom yang membentuk kunci primer tidak mengandung nilai Null. Bila menentukan nilai primer, SQL Server akan membuat indeks tertentu untuk memastikan bahwa kunci tersebut bersifat eksklusif.

Pada tabel kita, au_id dan book_id adalah kunci primer dari tabel Authors dan Books. Tabel Autbook memiliki kunci primer yang dibentuk dari kolom au_id dan book_id . Kunci primer juga dapat digunakan bersama-sama untuk menghubungkan antar tabel.

Kunci Asing

Kunci asing adalah kolom atau kombinasi kolom yang digunakan untuk menentukan tautan antara dua tabel. Tautan atau hubungan ini dibuat dengan membuat bidang yang sama untuk kedua tabel. Penggunaannya sangat umum untuk menjaga keutuhan hubungan antara dua tabel. Sebagai contoh, Anda memiliki tabel Employee dengan field bernama Department. Field ini berhubungan dengan field dep_id pada tabel lain bernama Dept yang memiliki data dari departemen perusahaan. Jika tidak ada kunci asing, semua orang bisa menghapus baris dari departemen pada tabel Dept apabila ada beberapa pegawai departemen tersebut yang ditentukan di dalam record Anda. Kunci asing hanya mengizinkan penyisipan nilai-nilai yang ditemukan pada field-field yang berhubungan pada tabel asing.

Walaupun sama, kunci asing tidak perlu berhubungan dengan kunci primer pada tabel lain. Tetapi kunci asing harus merujuk kepada kolom yang mengandung isi yang eksklusif.

Batasan UNIQUE

Selain kunci primer, kita bisa membuat batasan berjenis UNIQUE pada kolom lain. Apabila batasan jenis ini dibuat, SQL Server akan membuat sebuah indeks eksklusif untuk memastikan bahwa data tidak digandakan.

Batasan CHECK

Batasan berjenis CHECK membatasi nilai yang diterima oleh kolom. Pemeriksaan validitas data dilakukan dengan membandingkan data yang dimasukkan dengan daftar nilai atau analisa dari ekspresi logikal, sebagai contoh harga > 1 dan < 100 .

Menggunakan Tabel

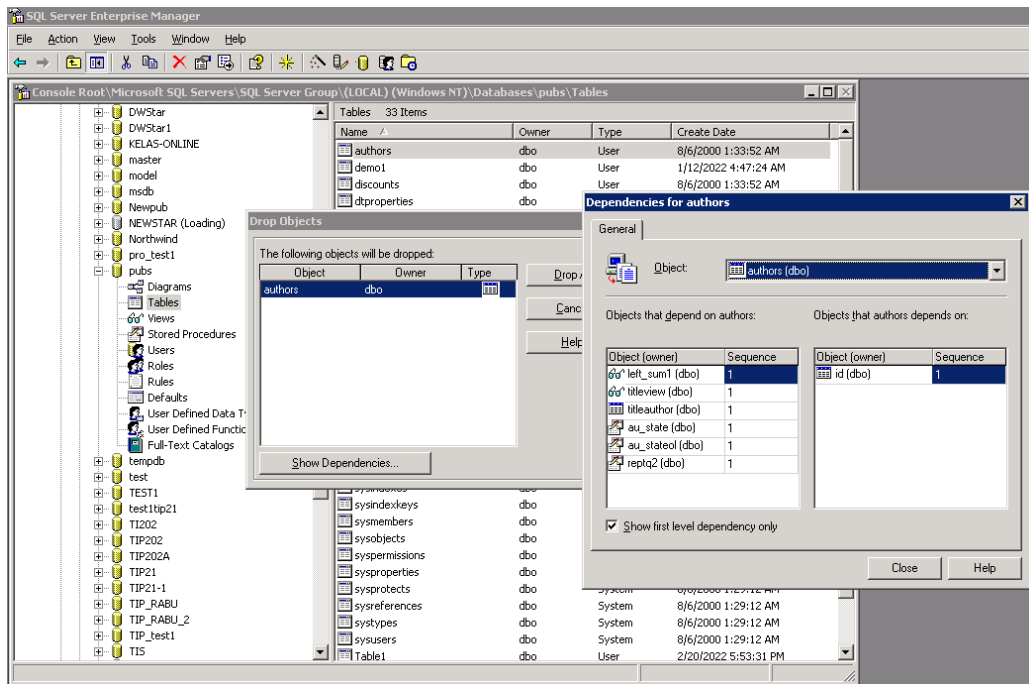
Mari kita lihat operasi-operasi dasar yang dilakukan pada tabel.

Menghapus Tabel

Anda bisa menghapus tabel dari database dengan langkah-langkah berikut ini: Perbesarlah folder Databases dan database yang mengandung tabel-tabel yang ingin dihapus. Lalu perbesarlah folder Tables. Klik kanan nama tabel yang ingin Anda hapus, lalu klik Hapus. Kotak dialog Drop Objects akan muncul.

Untuk menghapus tabel, tekan tombol Drop All. (Jangan benar-benar menghapus tabel pada contoh ini). Untuk melihat semua ketergantungan yang dimiliki oleh tabel, tekan Show

Dependencies. Jika ada ketergantungan, sebuah kotak dialog akan muncul. Untuk kembali ke layar sebelumnya, tekan Close.



Gambar 3.3 Menghapus Tabel

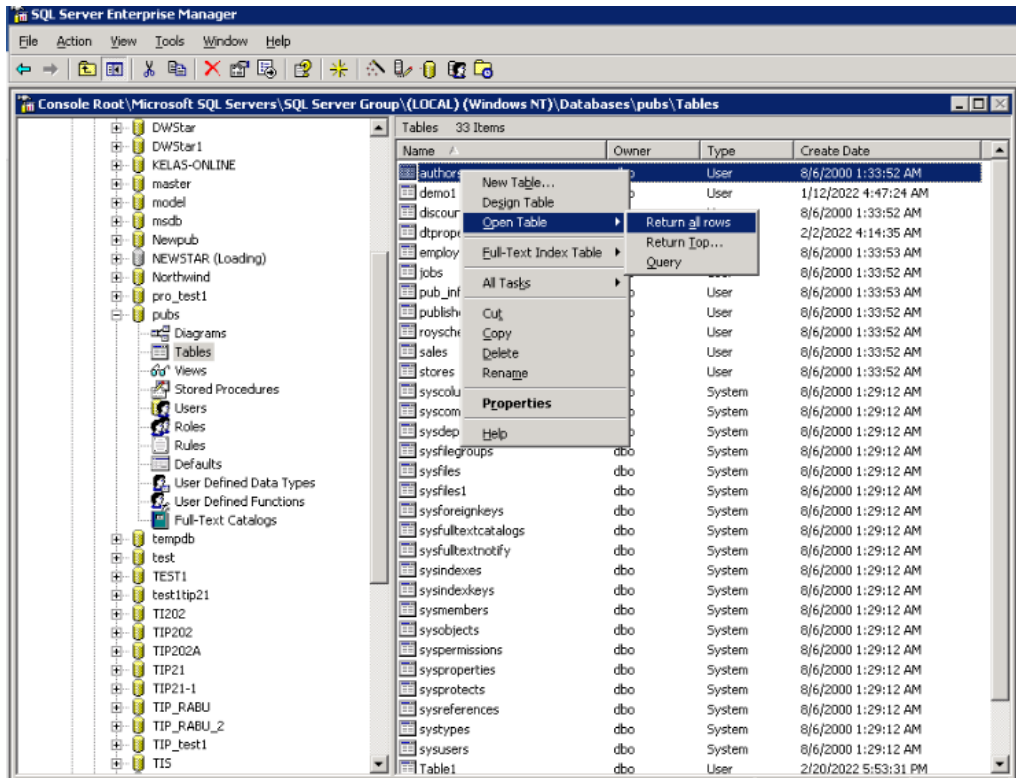
Ganti nama tabel

Luaskan folder Database dan database yang berisi tabel yang ingin Anda ganti namanya. Kemudian perluas folder Tabel. Klik kanan tabel yang ingin Anda ganti namanya.

Masukkan nama baru untuk tabel, lalu klik di luar area masukan.

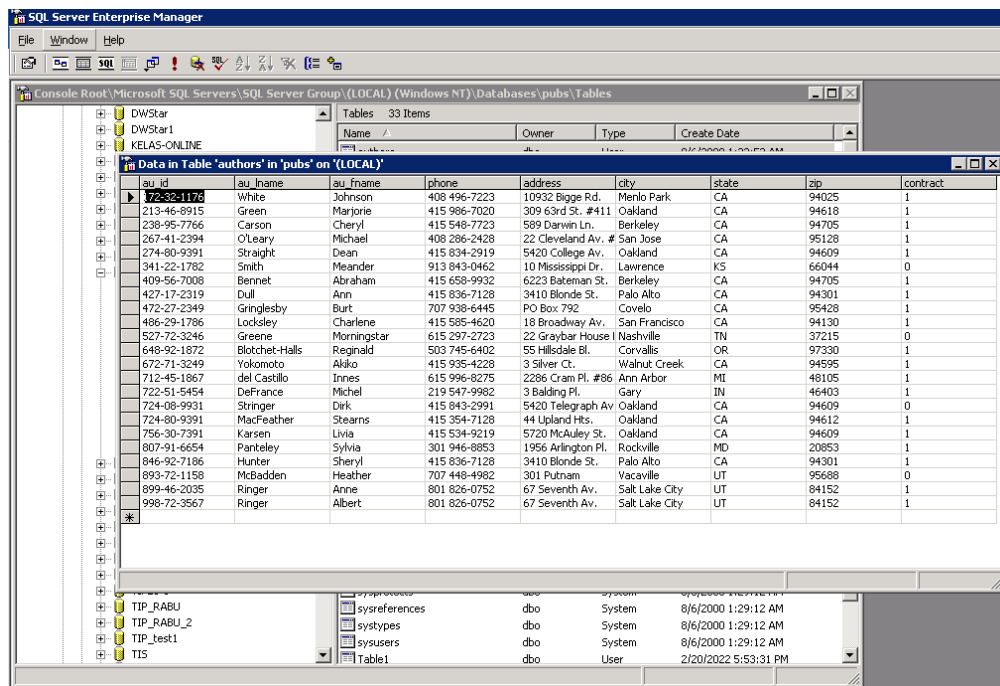
Melihat Data Tabel

Di panel data, klik kanan nama tabel yang ingin Anda lihat. Pilih OpenTables. Klik Kembali semua baris untuk menampilkan semua baris dalam tabel, atau klik Kembali ke atas untuk menentukan jumlah baris yang akan ditampilkan.



Gambar 3.4 Melihat Data Tabel

Sebuah jendela baru akan terbuka, menampilkan record-record pada tabel.



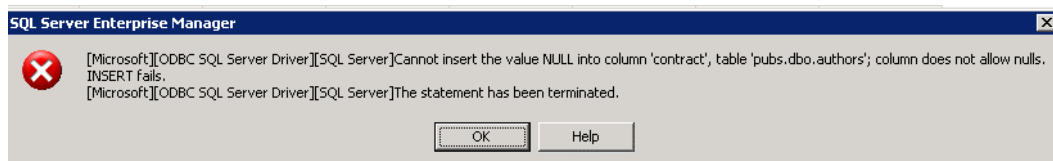
Gambar 3.5 Isi Data Tabel

Pada jendela ini, Anda bisa menggunakan scroll bar untuk melihat lebih banyak baris atau kolom.

Mengedit data dalam tabel dengan Enterprise Manager

Jika Anda ingin mengedit data dalam bidang tabel, ikuti prosedur yang sama seperti saat Anda menampilkan tabel. Saat catatan yang ingin Anda edit terlihat, klik bidang yang ingin Anda edit dan masukkan konten baru.

Ingatlah bahwa batasan yang ditempatkan pada bidang ini harus dipatuhi saat mengedit catatan. Jika Anda mencoba menyisipkan record baru pada tabel Authors dari database Pubs, dan tidak mengetikkan isi yang tepat kedalam field Contract, pesan berikut ini akan muncul:



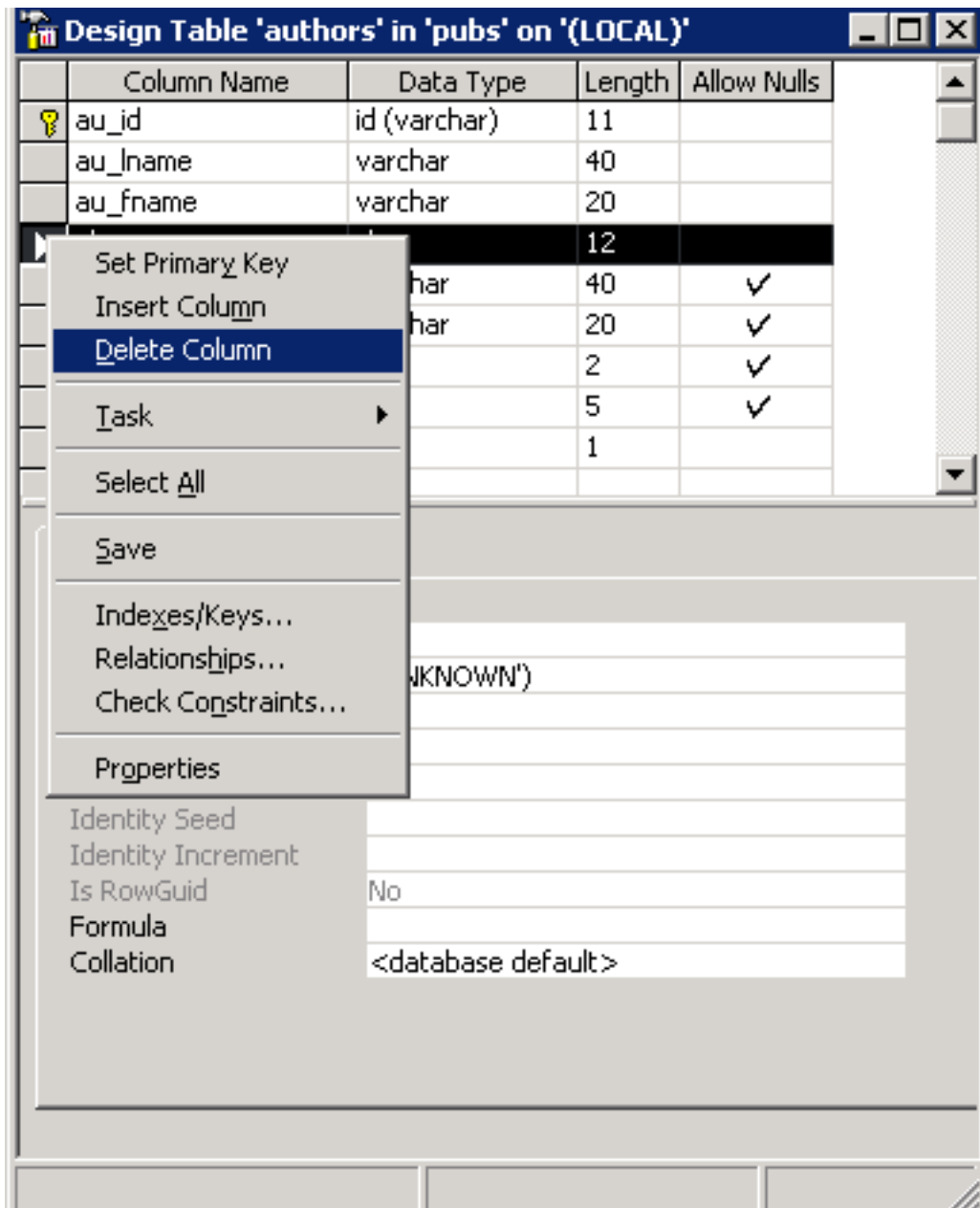
Gambar 3.6 Error menyunting data dalam table

Menambah Kotom Baru Ke Datam Tabel

Luaskan folder Database dan database yang berisi tabel yang ingin Anda ubah. Kemudian perluas folder Tabel. Klik kanan tabel yang ingin Anda ubah. Klik pada opsi Tabel Desain. Pilih kolom di sebelah kanan kolom baru, lalu klik kanan nama kolom. Pilih Tambahkan Kolom. Setelah memasukkan informasi, tekan Simpan. Anda dapat membatalkan penghapusan baris saat mengetik dengan menekan tombol Esc.

Menghapus Kolom Dari Tabel

Perbesarlah folder Databases dan database yang ingin Anda sunting. Lalu perbesarlah folder Tables. Klik kanan nama tabel yang ingin Anda ubah. Klik pada opsi Tabel Desain. Pilih kolom yang ingin Anda hapus, lalu klik Hapus Kolom.

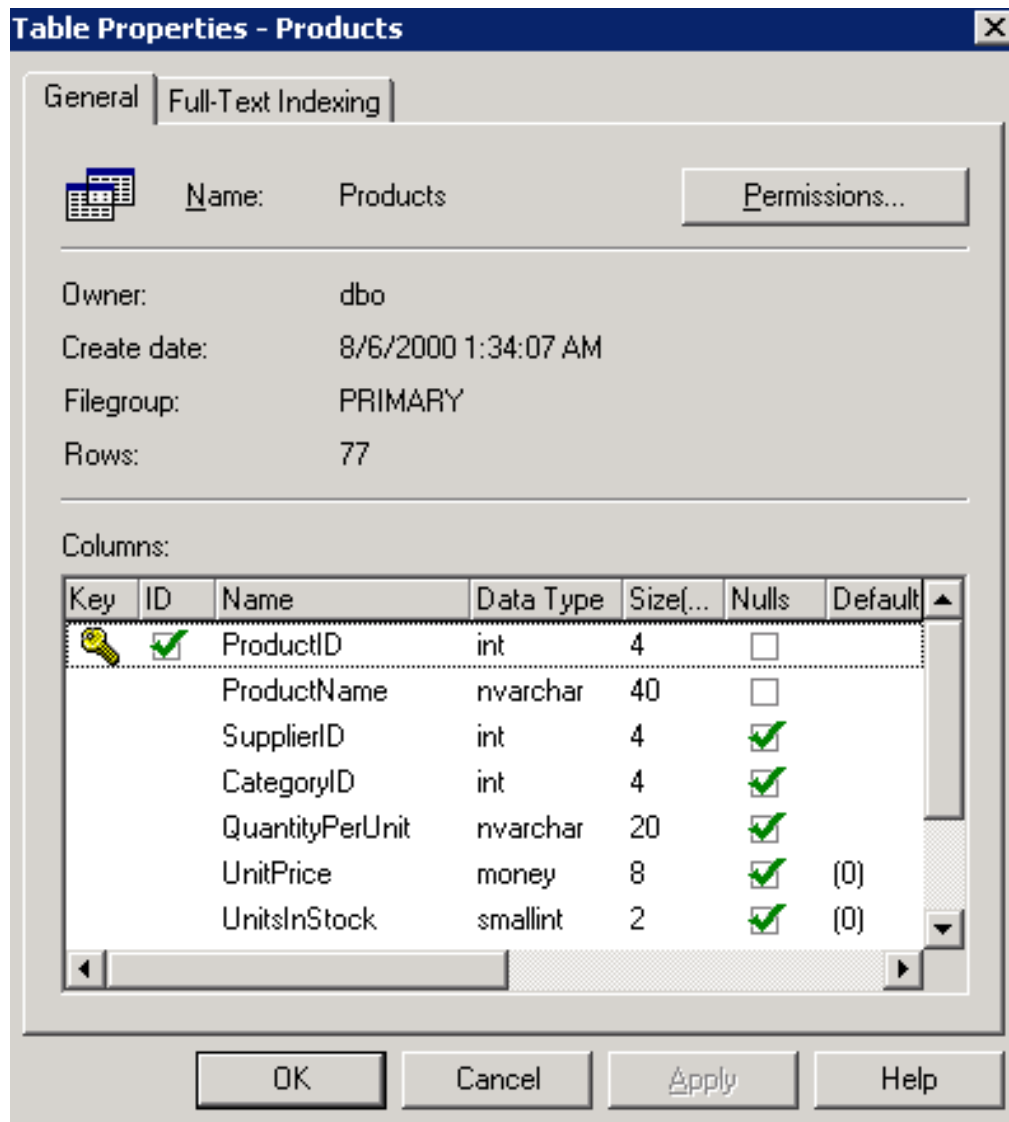


Gambar 3.7 Menghapus Kolom Dari Tabel

Saat kolom dihapus, tekan Simpan. Anda tidak dapat menghapus kolom yang digunakan oleh indeks, batasan CHECK atau UNIQUE, atau kunci utama dan asing.

Lihat Properti di Tabel

Untuk mengetahui struktur dari tabel serta perincian mengenai kolom-kolomnya, Anda perlu mengaktifkan kotak dialog Properties. Perbesarlah folder Data bases dan database yang mengandung tabel. Lalu perbesarlah folder Tables. Klik dengan tombol mouse kanan pada nama tabel, lalu klik pada pilihan Properties. Layar di bawah menampilkan properti dari tabel Products pada database Northwind.

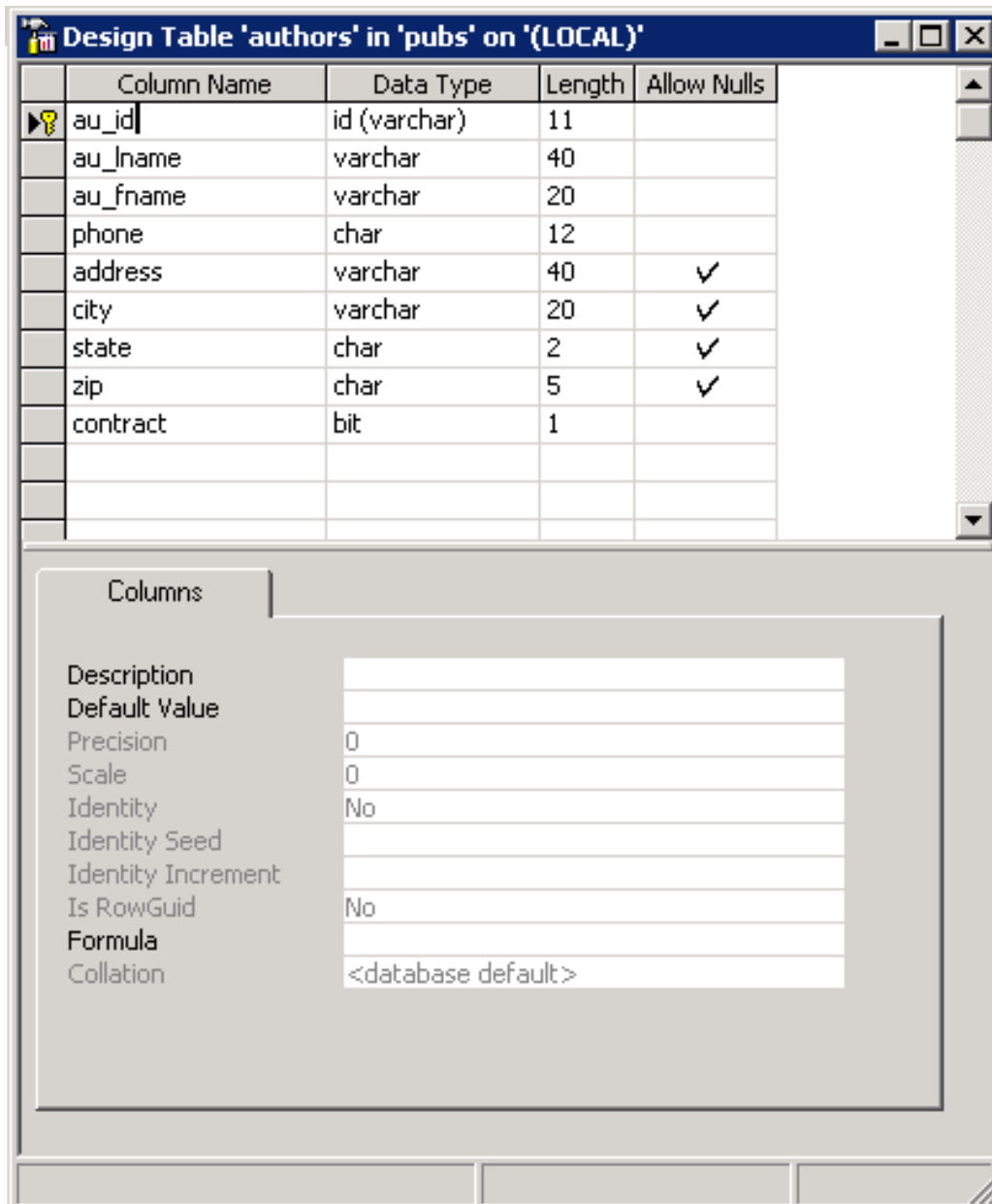


Gambar 3.8 Melihat Properti dari Tabel

Selain melihat properti dari kolom-kolom, Anda juga mengetahui izin akses untuk tabel dengan menekan Permissions. Untuk menutupnya, tekan OK .

Membuat Tabel Authors

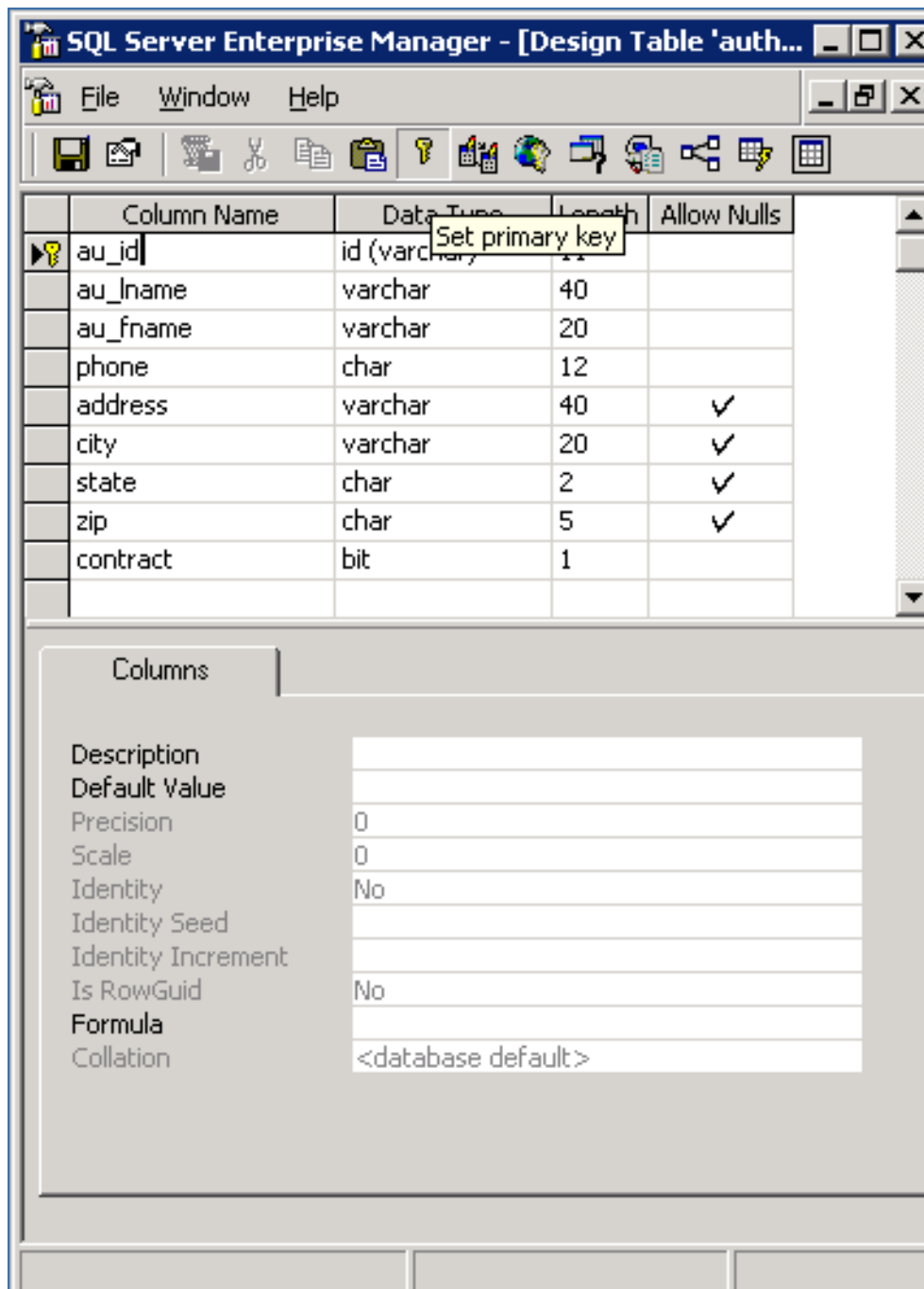
Mari kita buat tabel Authors di dalam database Newpub. Untuk melakukan ini, perbesarlah folder Databases, Klik kanan Tabel, lalu klik Tabel Baru. Masukkan penulis di kotak dialog pertama dan isi informasi di tabel penulis seperti yang ditunjukkan di bawah ini. Kata Brazil harus ditutup dengan tanda kutip tunggal. Kalau tidak, akan muncul error saat tabel ini ditulis.



Gambar 3.9 Membuat Tabel Authors

Membuat Kunci Primer

Kunci utama tabel ini adalah bidang au_id. Untuk menentukan kunci utama, klik sel di sebelah kiri nama kolom yang sesuai dengan bidang yang Anda inginkan. Kemudian klik tombol Set Master Key.



Gambar 3.10 Membuat Kunci Primer

Untuk menyelesaikan membuat tabel, tekan Save. Buatlah tabel contoh Books dan Autobook dengan menggunakan teknik yang sama.

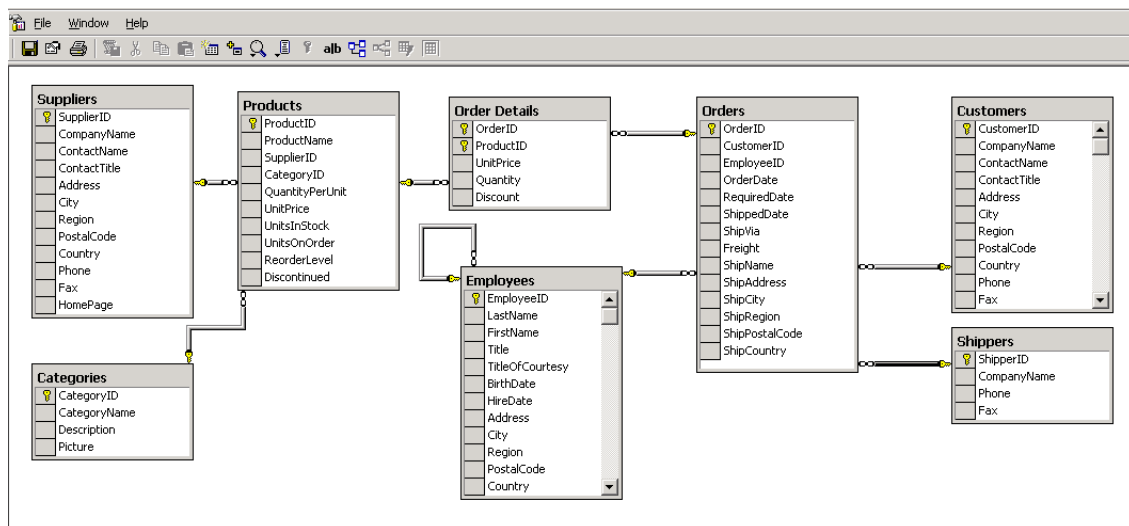
Kesimpulan

Dalam bab ini, Anda mempelajari cara membuat tabel, mengeditnya, dan menambahkan data. Pada bab selanjutnya kita akan melihat bagaimana membuat hubungan antar tabel dan bagaimana menggunakan diagram database.

BAB 4

Diagram Basis Data

Diagram database adalah representasi grafis dari database. Diagram database memungkinkan Anda untuk mengelola objek database menggunakan antarmuka pengguna grafis dan menyeret mouse untuk melakukan tugas yang biasanya dilakukan dengan perintah SQL. Tugas ini termasuk menambahkan atau menghapus tabel, pemicu, prosedur, batasan kolom tabel, dan hubungan antar tabel. Berikut adalah contoh diagram database Northwind:



Gambar 4.1 Diagram Basis Data

Anda dapat melihat tabel database dan hubungan di antara mereka. Semua tabel yang ditampilkan di sini hanyalah referensi ke tabel yang disimpan secara fisik di database. Beberapa bagan dapat dibuat untuk database yang sama. Berikut adalah beberapa tugas yang dapat dilakukan pada grafik:

- Lihat struktur dan hubungan tabel
- Buat beberapa tampilan bagian dari database yang kompleks
- Percobaan dengan struktur database tanpa mengubah database
- Mengubah struktur database

Anda juga bisa mengubah karakteristik tabel dengan menggunakan diagram. Perubahan ini secara otomatis tercermin dalam bagan lain tempat tabel ditampilkan. Itu juga dapat digunakan untuk menghapus tabel. Perubahan yang dilakukan pada tabel dalam diagram tidak

mempengaruhi tabel fisik hingga diagram ditulis. Bagan memungkinkan Anda untuk membuat perubahan pada beberapa tabel dan hanya memilih tabel tempat perubahan Anda ditulis. Tindakan yang dijelaskan di bawah ini dapat menyebabkan tabel dibuat ulang saat bagan disimpan:

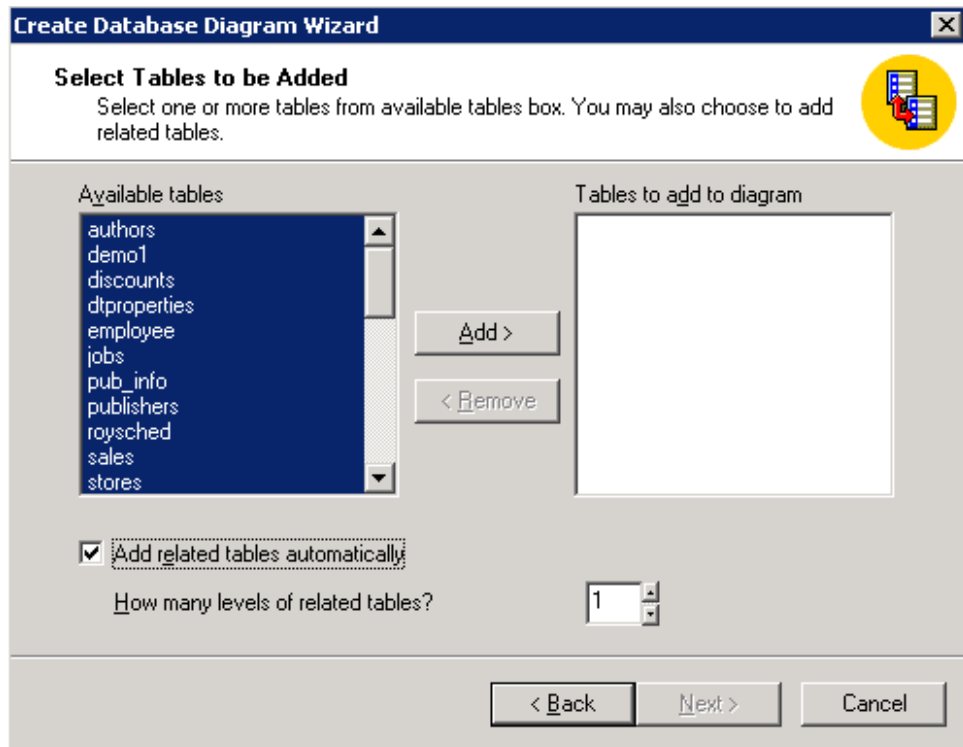
- Menambahkan array non-null ke akhir array aktif
- Tambahkan kolom ke badan tabel
- Susun ulang kolom yang ada
- Hapus kolom yang ada

Membuat Diagram

Untuk membuat bagan, perluas database yang diinginkan. Kemudian klik kanan Database Diagram dan pilih New Database Diagram. Ini mengaktifkan panduan pembuatan bagan. Buat bagan

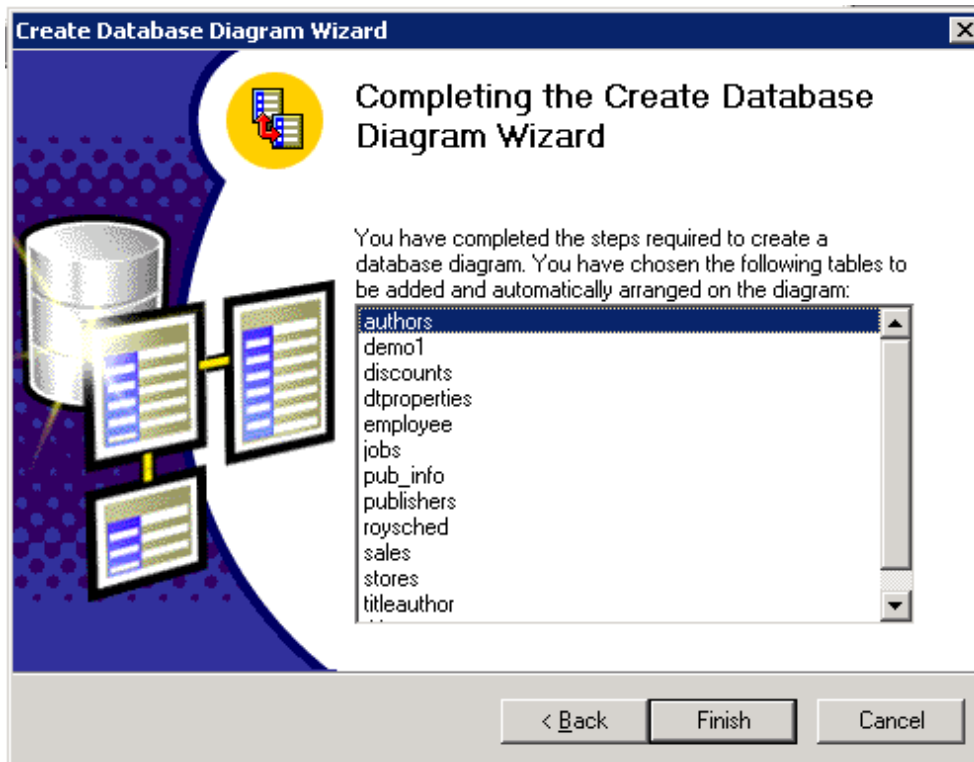
Untuk membuat bagan, perluas database yang diinginkan. Kemudian klik kanan Database Diagram dan pilih New Database Diagram. Ini mengaktifkan panduan pembuatan bagan.

Sekarang mari kita buat diagram untuk database Pubs. Setelah mengikuti langkah-langkah di atas, layar wizard pertama akan muncul. Klik pada layar tersebut dan lanjutkan. Pilihlah semua tabel pada kotak dialog berikutnya, seperti ditunjukkan di bawah ini. Tekan Add.



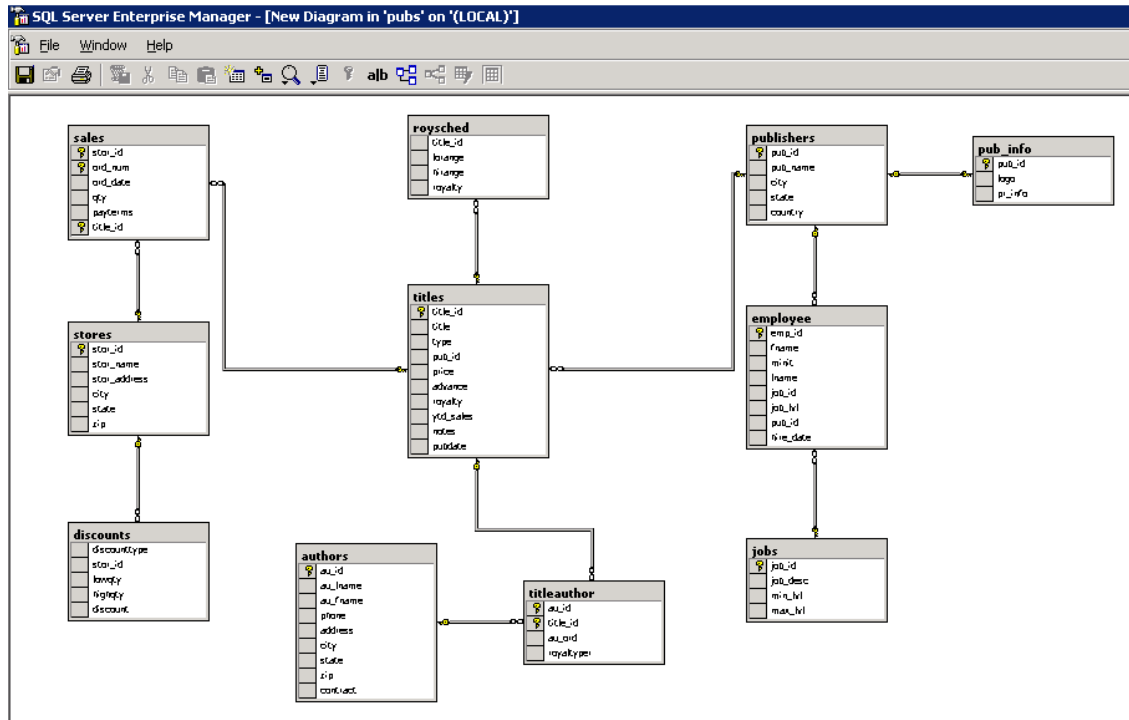
Gambar 4.2 Membuat Diagram Database

Tekan Berikutnya. Wizard menampilkan tabel untuk ditambahkan di layar berikutnya. Tekan Selesai.



Gambar 4.3 Membuat Diagram Database

Setelah Anda menekan Next, program menampilkan nama-nama tabel yang sudah dipilih. Klik Selesai untuk membuat bagan. Bagan kemudian akan muncul di ruang kerja seperti yang ditunjukkan di bawah ini.



Gambar 4.4 Diagram Database

Simpan grafik dengan mengklik tombol Simpan (ikon disk) dan masukkan nama untuk grafik. Kita akan memberinya nama **diagram02** untuk contoh ini.

Menggunakan Diagram

Bagian berikut ini menunjukkan cara melakukan beberapa operasi dasar dengan diagram.

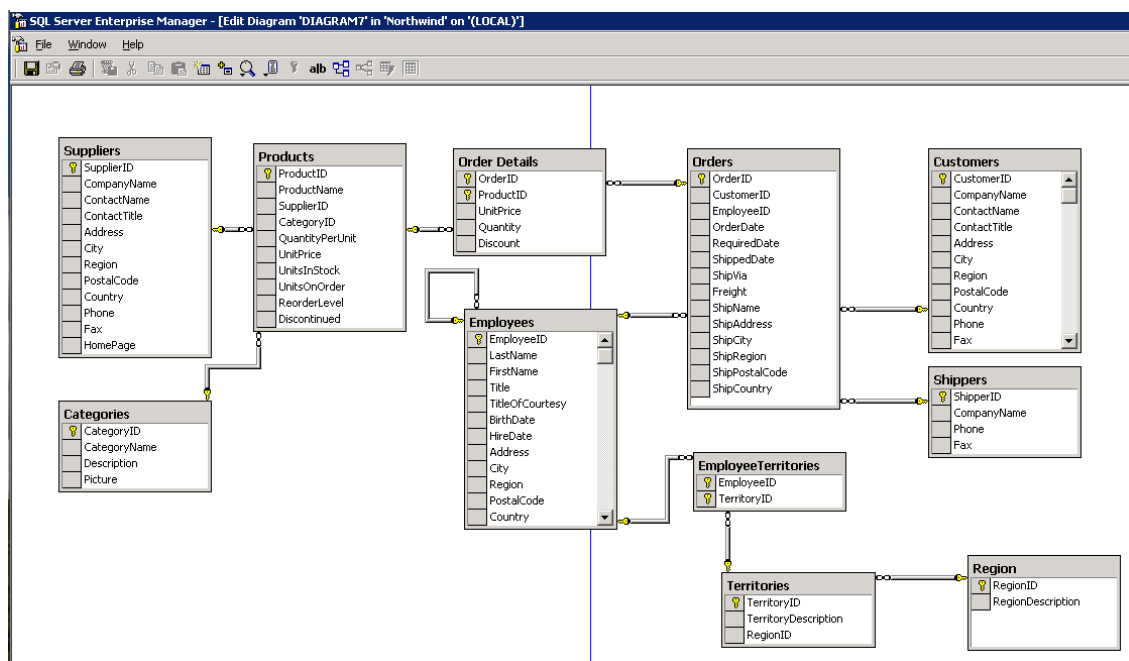
Membuka Diagram

Untuk membuka diagram yang sudah ada, perluas folder Server dan Database terlebih dahulu. Klik pada folder Diagram Database. Bagan yang ada ditampilkan di panel kanan. Klik dua kali diagram yang ingin Anda buka, atau klik kanan dan pilih Diagram Desain dari menu yang muncul.

Operasi ini akan membuka diagram database Northwind yang ditunjukkan di awal dari bab ini.

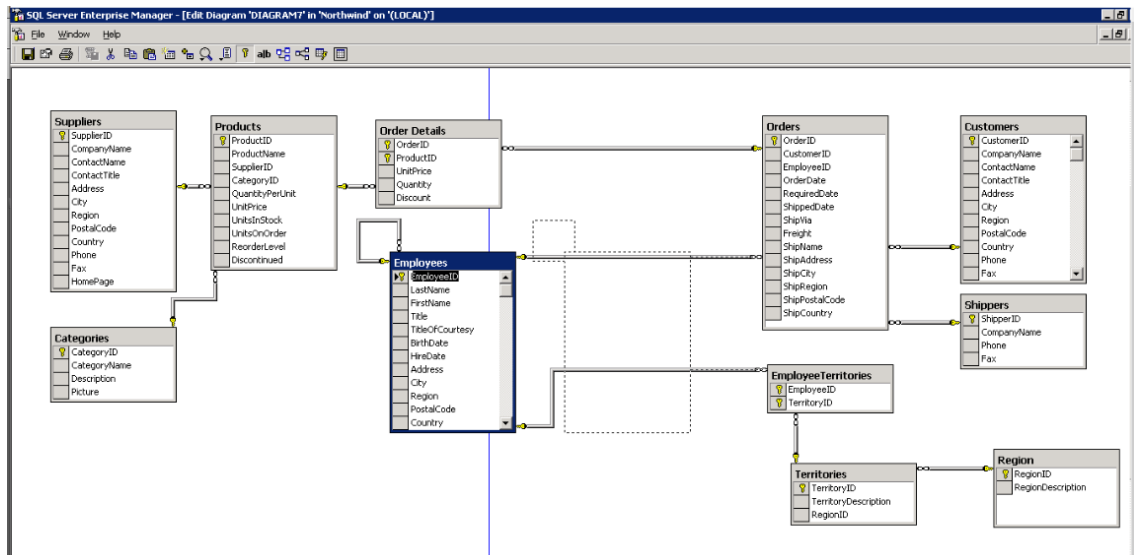
Mencetak Diagram

Diagram basis data adalah cara yang baik untuk mendokumentasikan basis data. Anda dapat mencetak bagan dengan mengklik tombol Cetak atau opsi Cetak di menu konteks. Tentukan di mana jeda halaman akan terjadi sebelum mencetak, karena surat dengan banyak tabel terkadang membutuhkan dua halaman atau lebih. Untuk menampilkan jeda baris, aktifkan menu konteks dengan meng-klik kanan tempat kosong di bagan, lalu pilih View Page Breaks. Untuk melihat seluruh bagan dan mengetahui berapa banyak ruang yang digunakan pada halaman, klik opsi zoom dan pilih persentase yang lebih kecil. Gambar berikut ditampilkan pada tingkat zoom 50%.



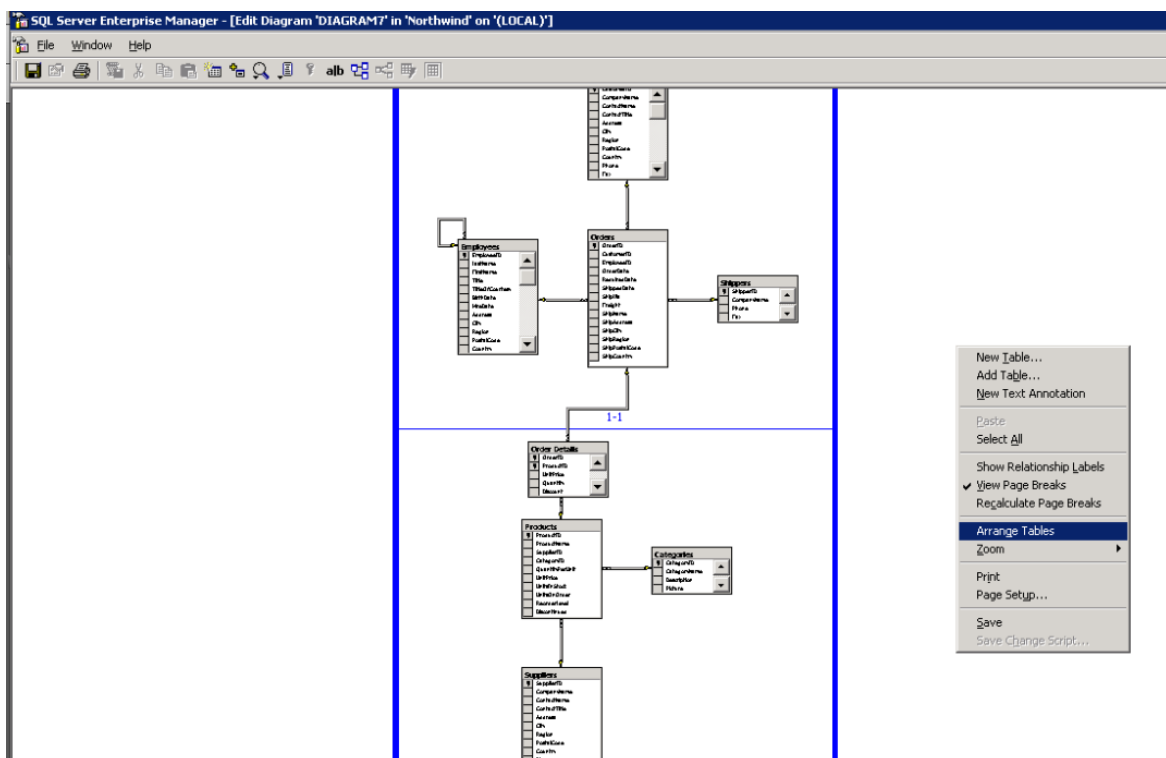
Gambar 4.5 Mencetak Diagram

Garis tebal di sekeliling diagram adalah pembatas halaman. Garis vertikal di tengah-tengah layar adalah pemenggalan halaman. Perhatikan dalam kasus ini bahwa ada dua tabel yang akan dibagi menjadi dua. Anda bisa secara manual mengatur penempatan tabel dengan mengklik pada baris judul tabel dan menggesernya ke posisi yang diinginkan, seperti ditunjukkan di bawah ini.



Gambar 4.6 Menata Diagram

Daripada melakukan pengaturan secara manual, Anda bisa menggunakan pilihan **Arrange Tables** pada menu shortcut. Perintah ini berfungsi mengorganisir ulang tabel-tabel secara otomatis dan menjaga agar pemenggalan halaman tidak membagi tabel apabila mungkin. Gambar berikutnya menunjukkan diagram dengan tingkat perbesaran 25% setelah tabel-tabel diatur.



Gambar 4.7 Arrange Tables Diagram

Keluar Dari Diagram

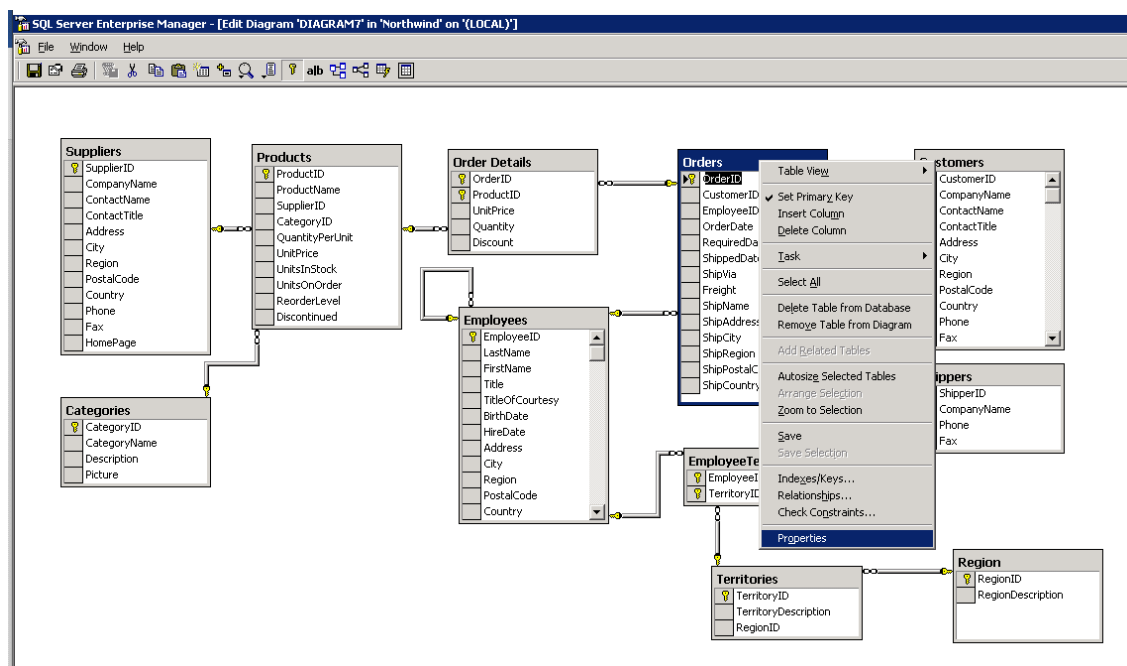
Untuk menutup diagram tanpa menyimpan perubahan yang telah dilakukan, klik pada tombol Close (tanda X pada baris judul) lalu pilih No pada kotak dialog. Jika Anda menjawab Yes, semua perubahan yang dibuat pada layout diagram akan ditulis dan akan mempengaruhi database.

Menggunakan Tabel Pada Diagram Database

Bagian berikut ini menjelaskan cara menggunakan tabel di dalam diagram database. Tugas-tugas ini meliputi membuat, menambah, dan menghapus tabel, serta membuat hubungan antartabel.

Melihat Properti Tabel

Anda dapat melihat properti tabel dan melakukan sebagian besar operasi terkait dengan mengklik kanan bilah judulnya. Menu popup dengan semua opsi untuk tabel itu akan muncul.

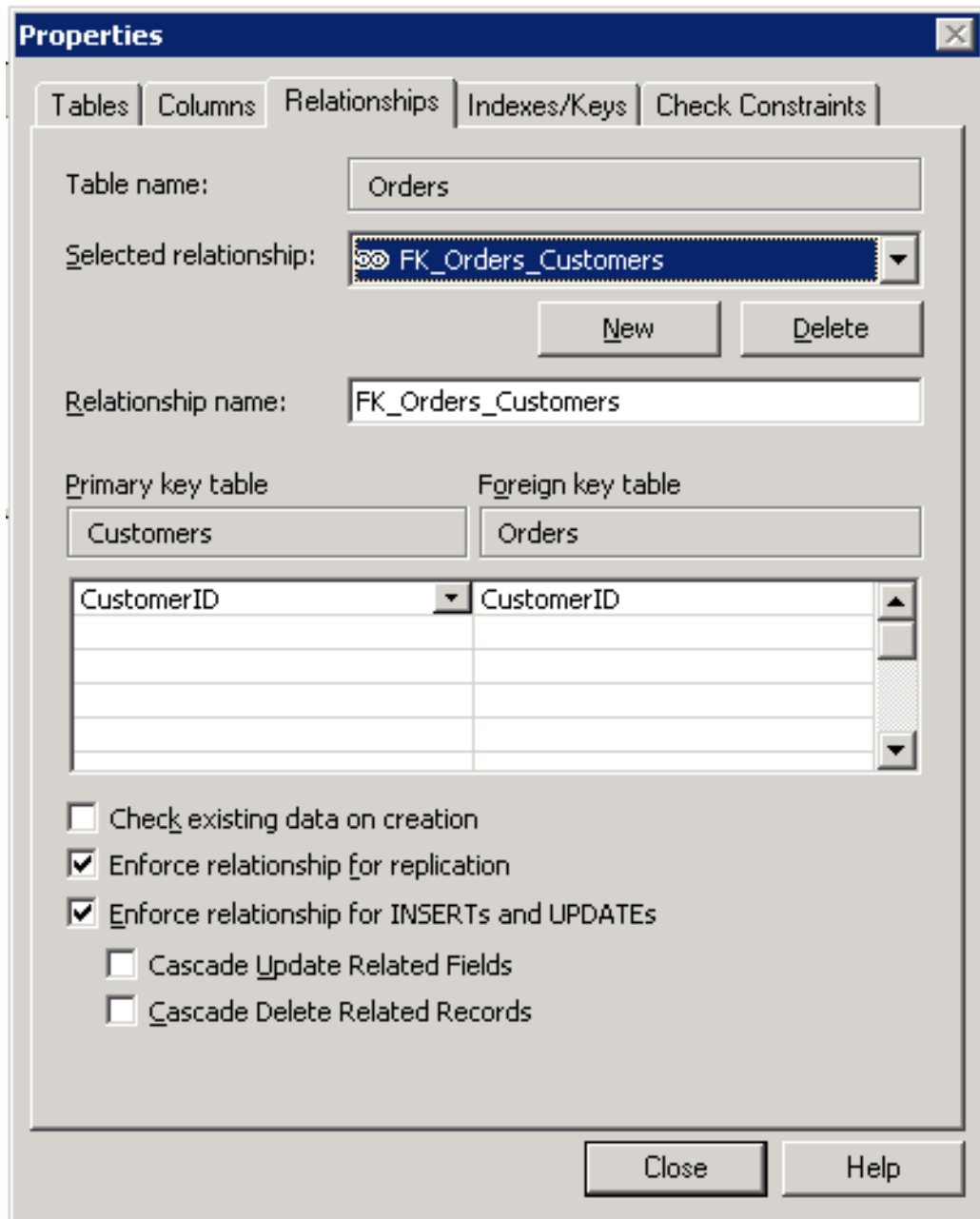


Gambar 4.8 Melihat Properties Table

Item terakhir adalah Properties. Karena tabel Orders yang dipilih, apabila Anda mengklik pada item ini, sebuah kotak dialog akan muncul dan menampilkan properti dari tabel tersebut. Pada kotak dialog ini terdapat tiga buah tab. Tab Relationship menampilkan hubungan antara tabel ini dan tabel lain. Tab ini menampilkan tabel-tabel yang dirujuk pada

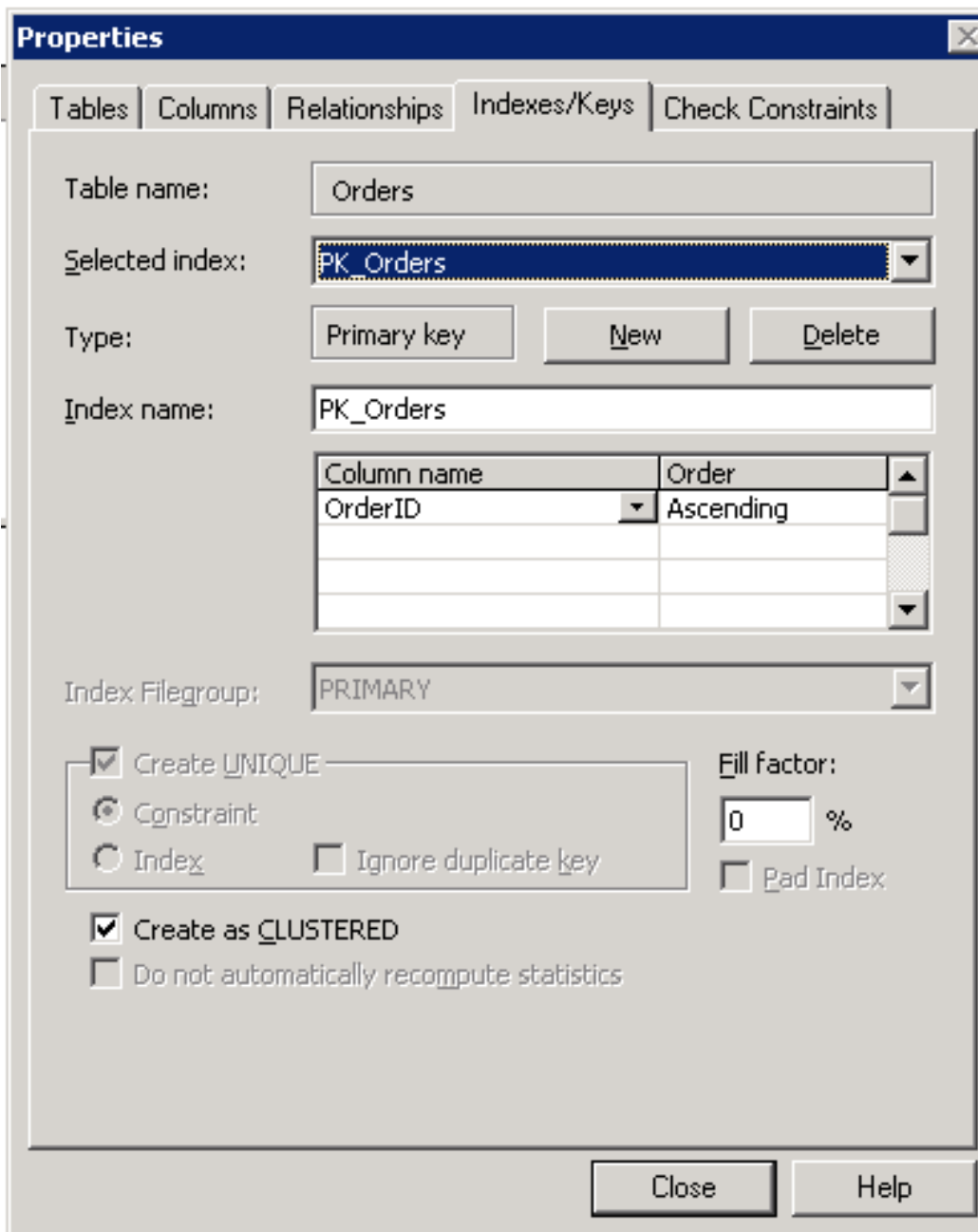
kunci asing dan tabel-tabel yang merujuknya.

Sebagai contoh, tabel Orders berhubungan melalui field CustomerId dengan tabel Customers yang memiliki field CustomerID sebagai kunci primernya, seperti ditunjukkan pada Gambar 4.9.



Gambar 4.9 Tampilan Properties Relationships

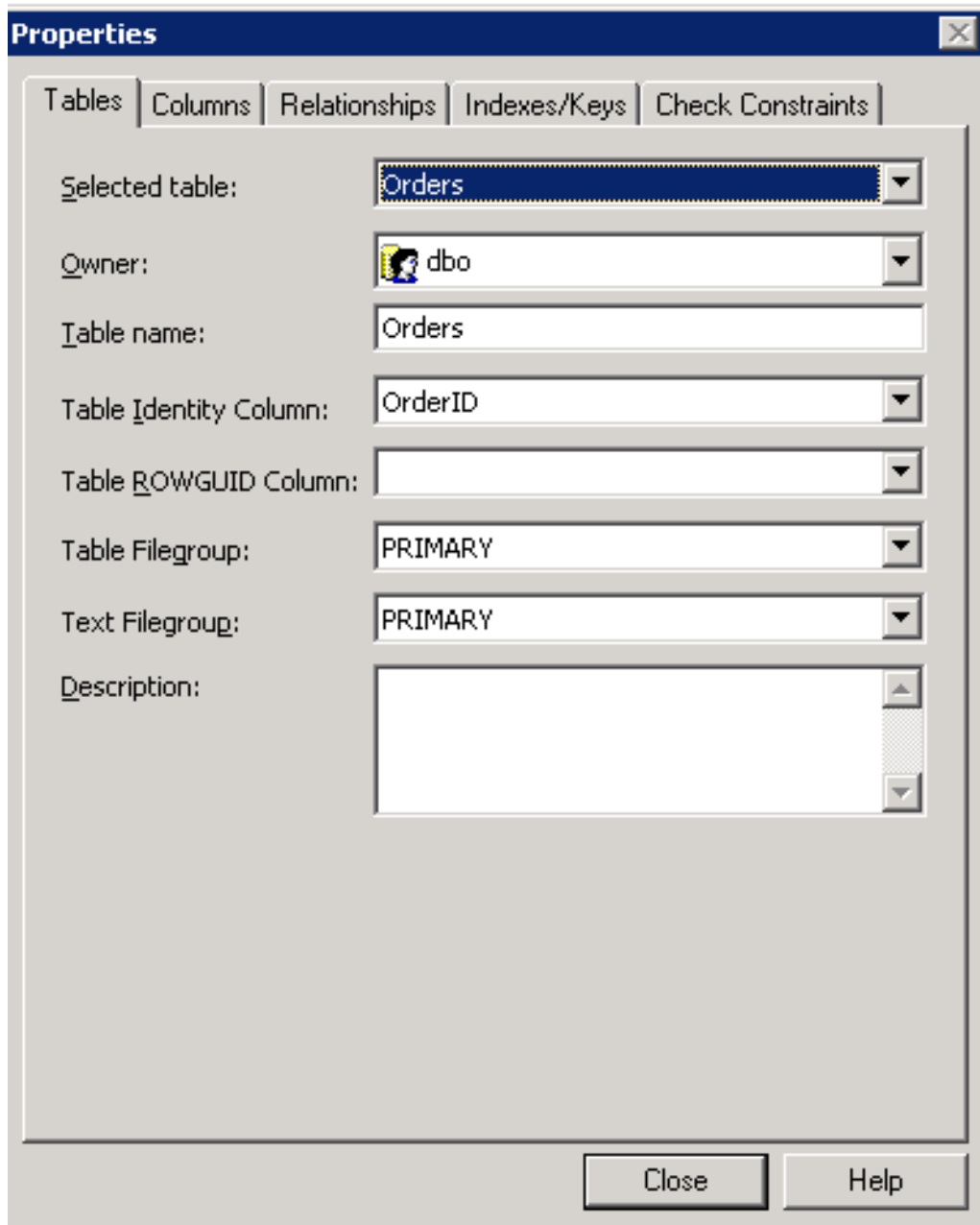
Tab Indexes/Keys mengizinkan Anda melihat dan membuat file indeks dari tabel. Pada contoh ini, kolom OrderID digunakan sebagai kunci primer tabel dan nama dari indeks yang berhubungan dengannya adalah PK_Orders .



Gambar 4.10 Tampilan Properties Indexes/Keys

Anda bisa membuat indeks-indeks baru dengan menggunakan tombol New. Untuk menghapus indeks , pilihlah indeks lalu tekan tombol Delete .

Tab Tables menampilkan informasi seperti misalnya nama dari tabel, lokasi file dari tabel, serta batasan-batasanya. Melalui field Selected Table Anda bisa memilih tabel-tabel lain di dalam diagram. Gambar berikut ini menampilkan tabel Products yang mengandung batasan yang membatasi nilai dari kolom UnitPrice.



Gambar 4.10 Tampilan Properties Tables

Mengubah Tampilan dari Tabel

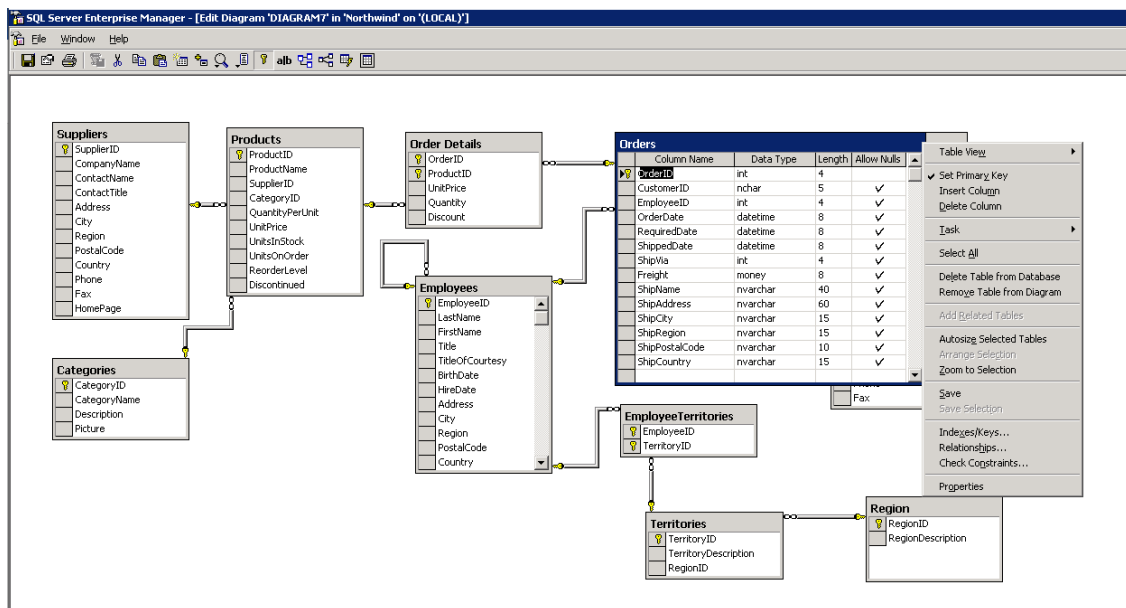
Secara default, diagram database menampilkan tabel sebagai kisi persegi panjang dengan nama kolom tabel. Anda dapat mengubah tampilan ini menggunakan menu konteks. Setiap tabel dapat dilihat dalam format berikut:

- Properti kolom: menunjukkan kisi dengan semua kolom tabel sebagai properti, misalnya ukuran, apakah kolom menerima nilai nol, tipe data, dll. Saat tabel baru ditambahkan ke bagan, format ini secara otomatis diaktifkan.
- Nama kolom: tampilkan semua nama kolom. Ini adalah tampilan default. Jika tabel

yang sudah ada ditambahkan ke bagan, tabel tersebut akan ditampilkan dalam format ini.

- Kunci: Menampilkan nama semua kolom yang merupakan kunci utama atau asing, atau kolom dengan properti unik.
- Name Only: hanya ada satu baris judul pada tabel.
- Custom: menampilkan kolom-kolom yang ditentukan oleh user saja.

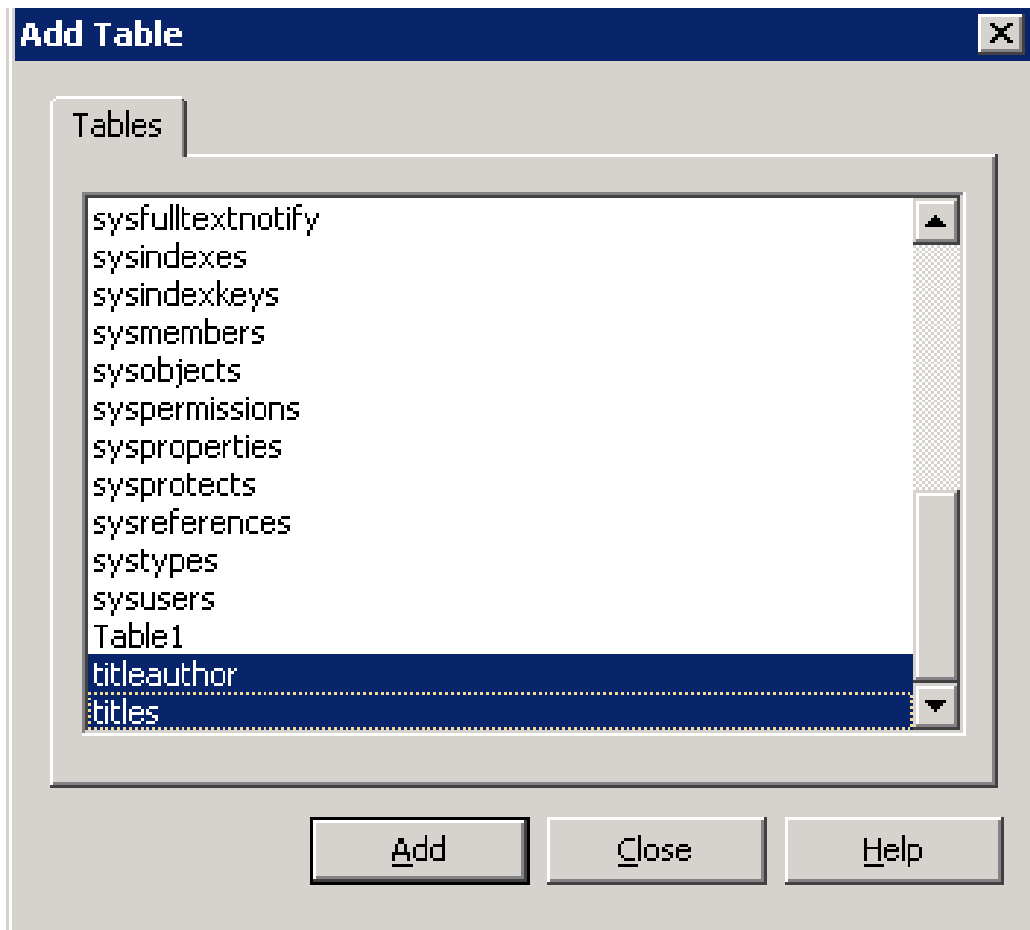
Gambar berikut ini menampilkan format Column Properties yang dipilih untuk tabel Orders.



Gambar 4.11 Mengubah Tampilan Tables

Menambahkan Tabel Ke Dalam Diagram

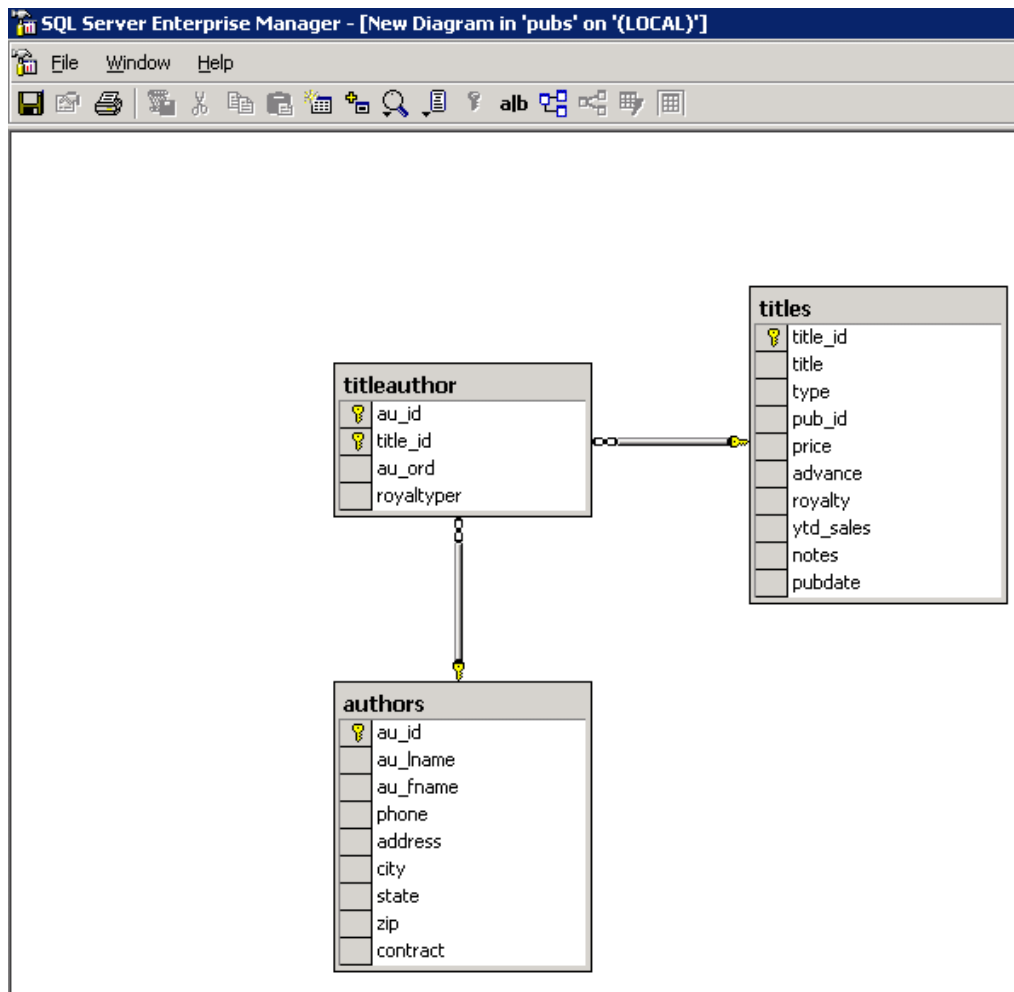
Di dalam bagian membuat diagram, kita telah menggunakan wizard untuk membuat diagram dengan semua tabel-tabel data. Kini kita akan membuat sebuah diagram kecil dan menambahkan tabel secara manual. Jalankanlah Create Diagram Wizard dan tekan Cancel pada layar pertama. Kemudian klik kanan pada ruang kerja dan pilih Add Table. Sebuah kotak dialog akan muncul, menampilkan tabel-tabel dari database. Kita akan membuat sebuah diagram yang mengandung tabel Authors, Titleauthor, dan Titles. Pertama-tama pilihlah tabel Authors dan tekan tombol Add.



Gambar 4.12 Menambah Tables

Tabel yang dipilih akan segera ditampilkan pada daerah kerja. Lalu pilihlah Titleauthor dan Titles secara serentak, dengan menekan tombol Shift sambil mengklik pada nama-nama tabel. Setelah dipilih, tekanlah Add. Tabel-tabel tersebut akan saling tumpang-tindih pada daerah kerja. Anda bisa mengatur posisinya dengan mengaktifkan menu shortcut dan memilih Arrange Tables.

Perhatikan bagaimana diagram ditam pilkan setelah tugas-tugas di atas dilakukan. Perhatikan bagaimana hubungan antartabel telah dibuat se.cara otomatis.

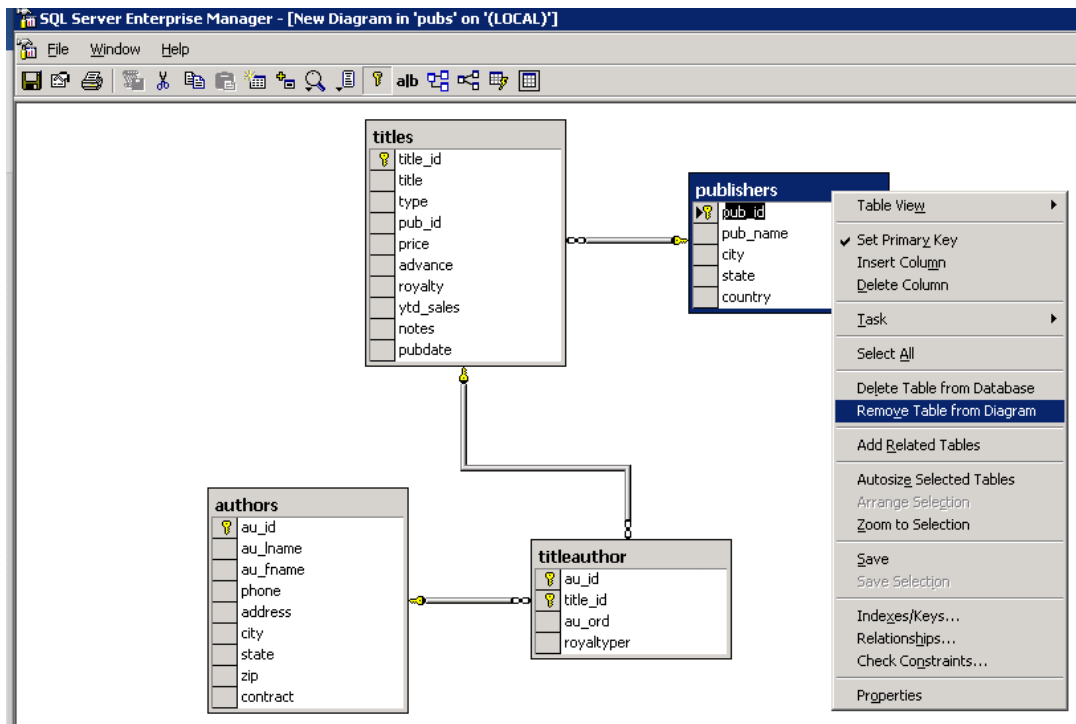


Gambar 4.13 Menambah 3 Tables

Untuk menyelesaikan contoh ini, tambahkanlah tabel Publishers ke dalam diagram dan simpanlah dengan nama diagram01.

Menghapus Tabel Dari Diagram

Untuk menghapus tabel dari diagram, klik kanan bilah judul tabel dan pilih Remove Table From Diagram. Seperti yang Anda lihat pada gambar di bawah, kami menghapus tabel terakhir yang ditambahkan.

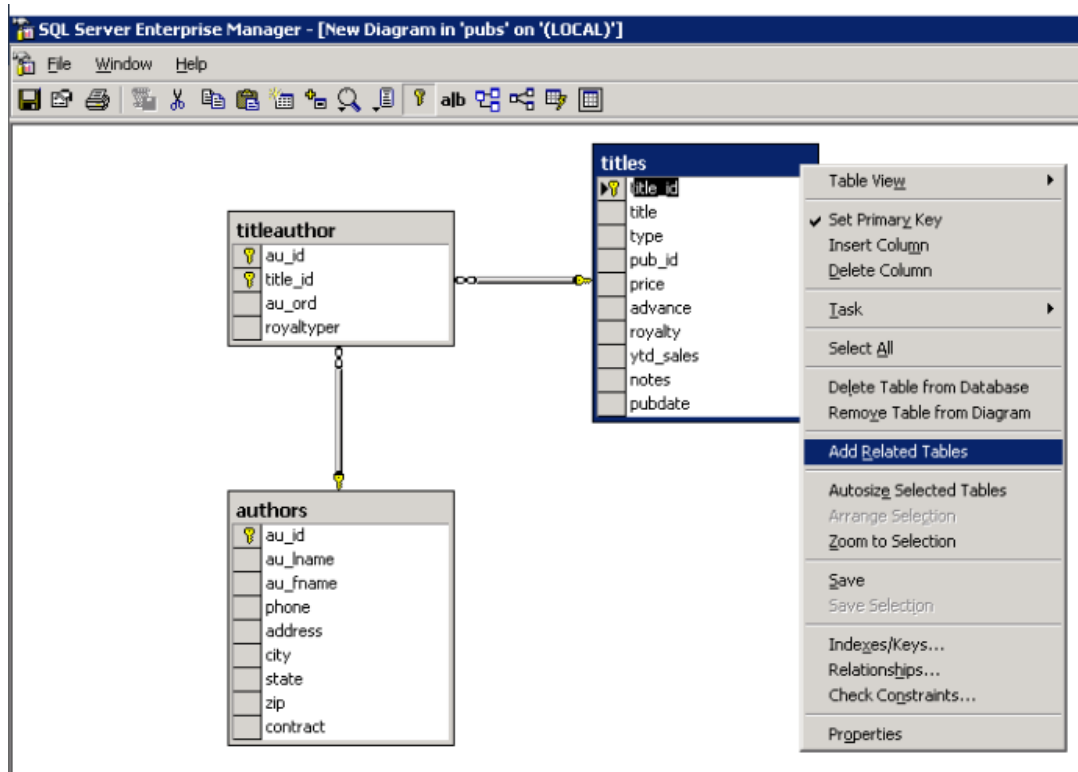


Gambar 4.14 Menghapus Tables

Apa bila Anda memilih Delete Table dari pilihan Data base, sebuah pesan akan muncul dan menyatakan bahwa tabel secara fisik akan dihapus dari data base. Hati-hati sewaktu melakukan operasi ini.

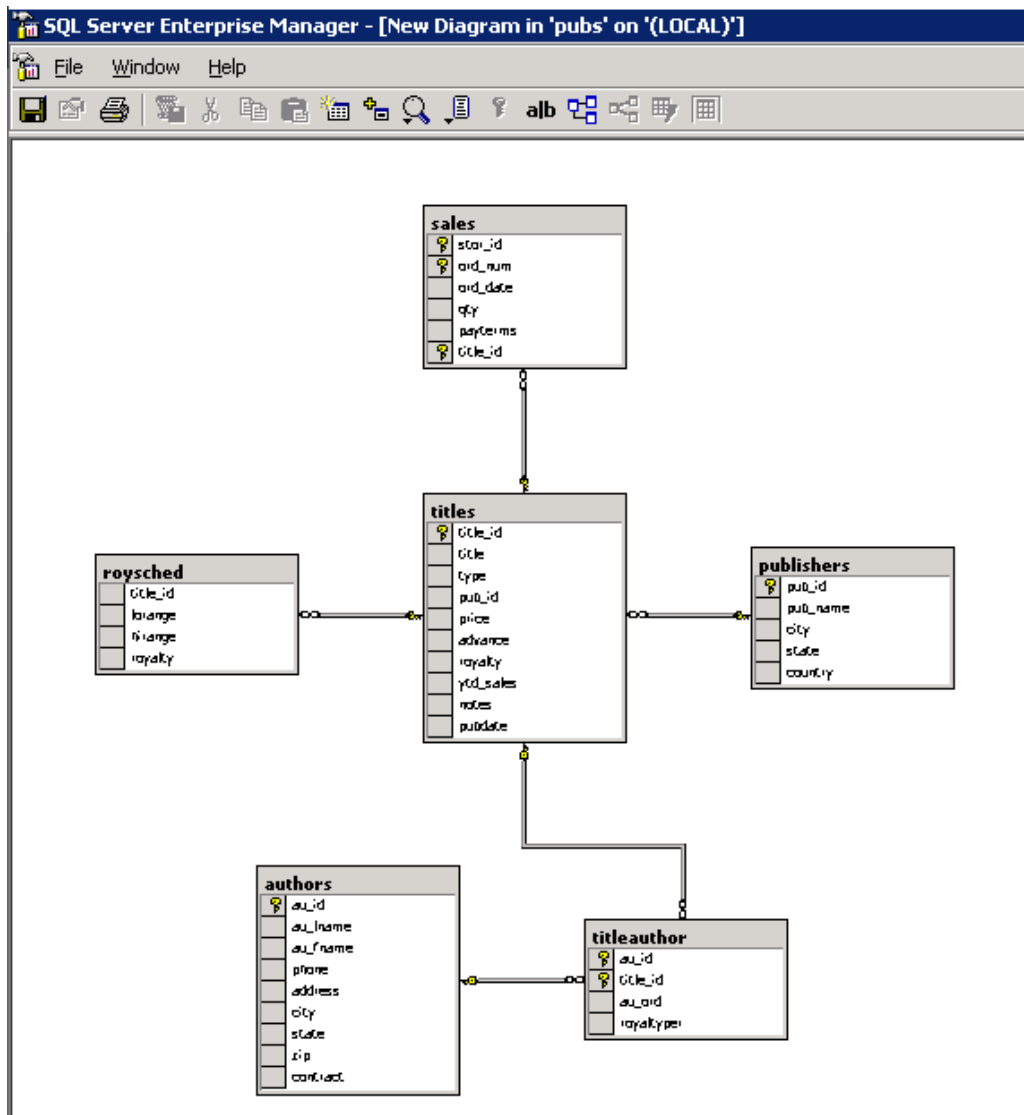
Menambah Tabel-Tabel Yang Berhubungan

Diagram database mengizinkan semua tabel yang berhubungan dengan tabel lain disertakan sekaligus. Sebagai contoh, tabel Titles berhubungan dengan tabel Publishers, Sales, dan Roysched, selain dengan kedua tombol yang sudah berhubungan (Titleauthor dan Authors). Untuk menambahkan tabel-tabel lain yang berhubungan, klik pada tabel di mana semua link berasal lalu pilih Add Related Tables.



Gambar 4.15 Menambah Tables

Tabel-tabel yang berhubungan akan ditampilkan pada diagram. Gunakanlah pilihan Arrange Tables agar diagram ditampilkan seperti gambar berikut ini.

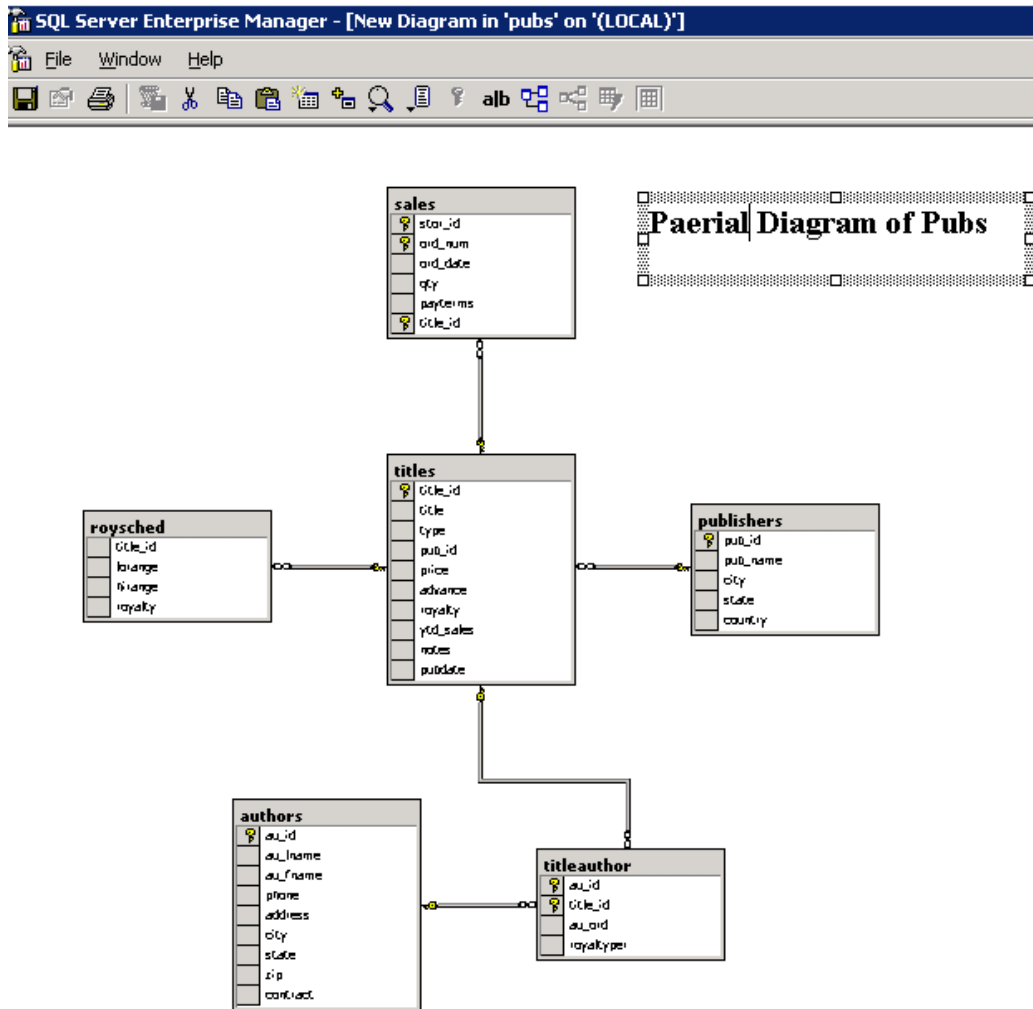


Gambar 4.16 Menambah Tables

Menyertakan Catatan

Diagram data base mengizinkan Anda menyertakan catatan atau komentar di dalam kotak teks pada daerah kerja. Untuk melakukannya, klik kanan tempat Anda ingin menambahkan teks, lalu pilih Komentar Teks Baru.

Sebuah kotak akan muncul. Ketikkanlah teks yang Anda inginkan di dalam kotak tersebut. Anda bisa memperbesar atau memperkecil ukuran kotak dan mengatur teks dengan mengklik dan menggeser sudut-sudutnya.



Gambar 4.17 Menambah Catatan

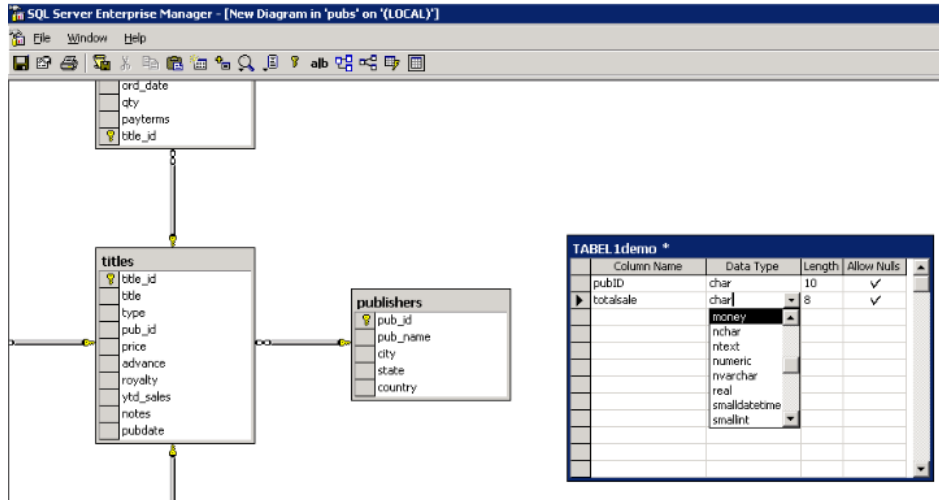
Membuat Tabel Baru

Untuk membuat tabel baru, tekanlah tombol New Table pada toolbar. Ketikkan TABLE1 demo pada kotak dialog lalu tekan OK.



Gambar 4.18 Membuat Tabel Baru

Grid untuk membuat tabel akan muncul pada daerah kerja. Anda hanya perlu menentukan karakteristik dari tabel yang baru tersebut. Sebagai contoh, kita akan membuat dua kolom seperti ditunjukkan pada Gambar 4.19 berikut ini:

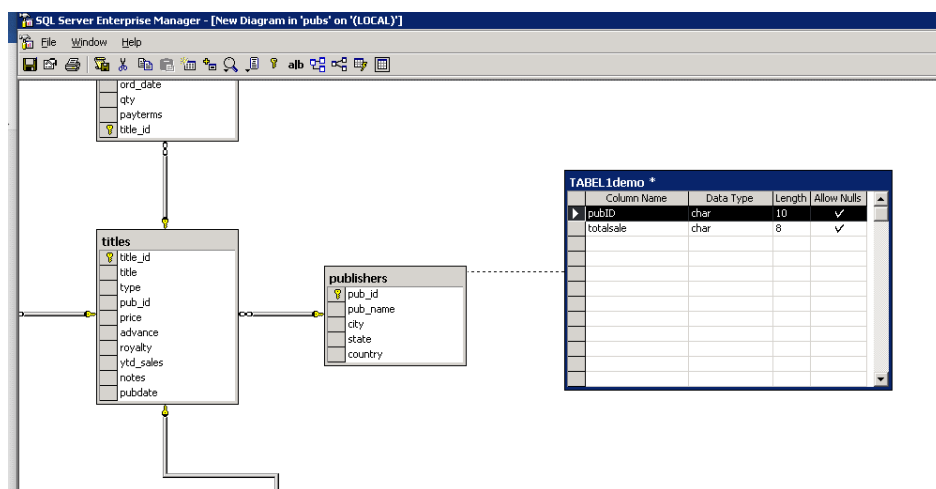


Gambar 4.19 Membuat Tabel Baru

Untuk membuat kunci primer, klik dengan tombol mouse kanan pada baris pertama dari grid, dan klik pada Set Primary Key pada toolbar.

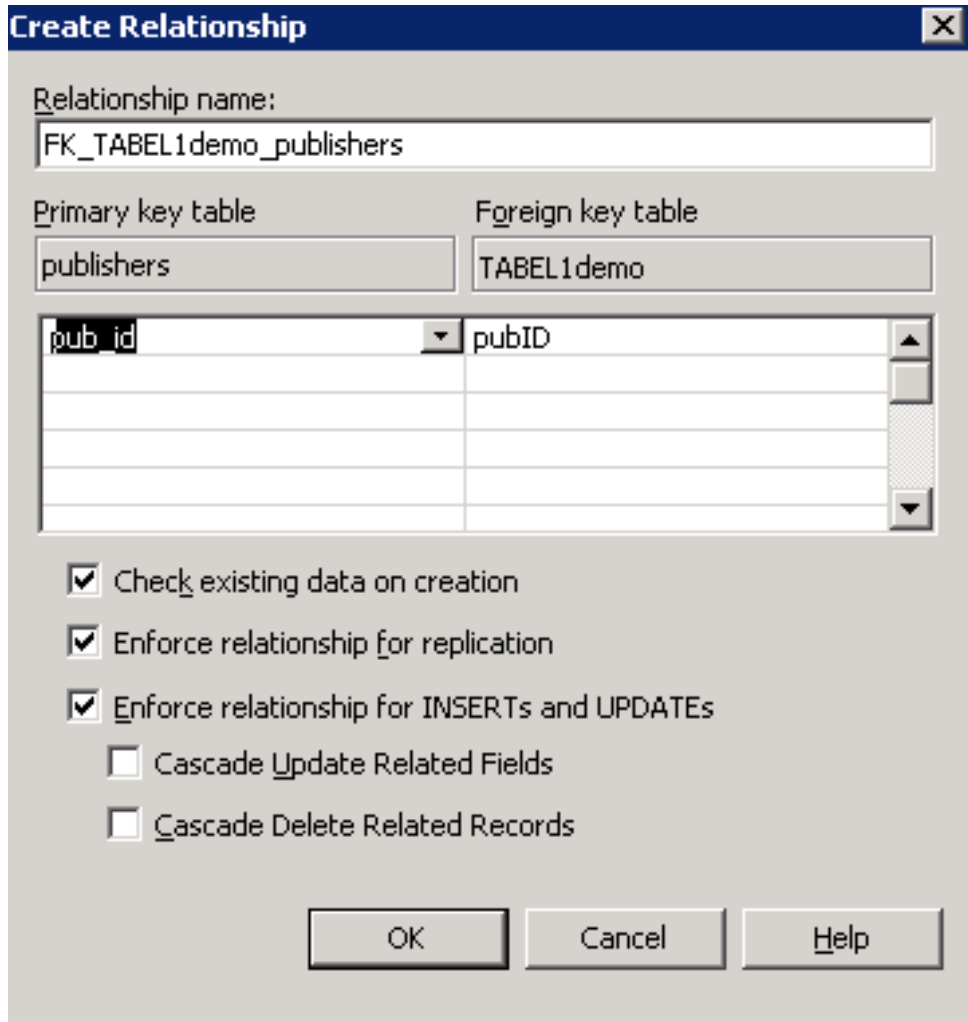
Membuat Hubungan Antartabel

Kini setelah tabel baru dibuat, kita akan menghubungkannya dengan tabel Titles. Proses ini sangat sederhana. Pilih baris yang menentukan kolom kunci utama di tabel baru, lalu gulir ke nama kolom terkait di tabel lainnya.



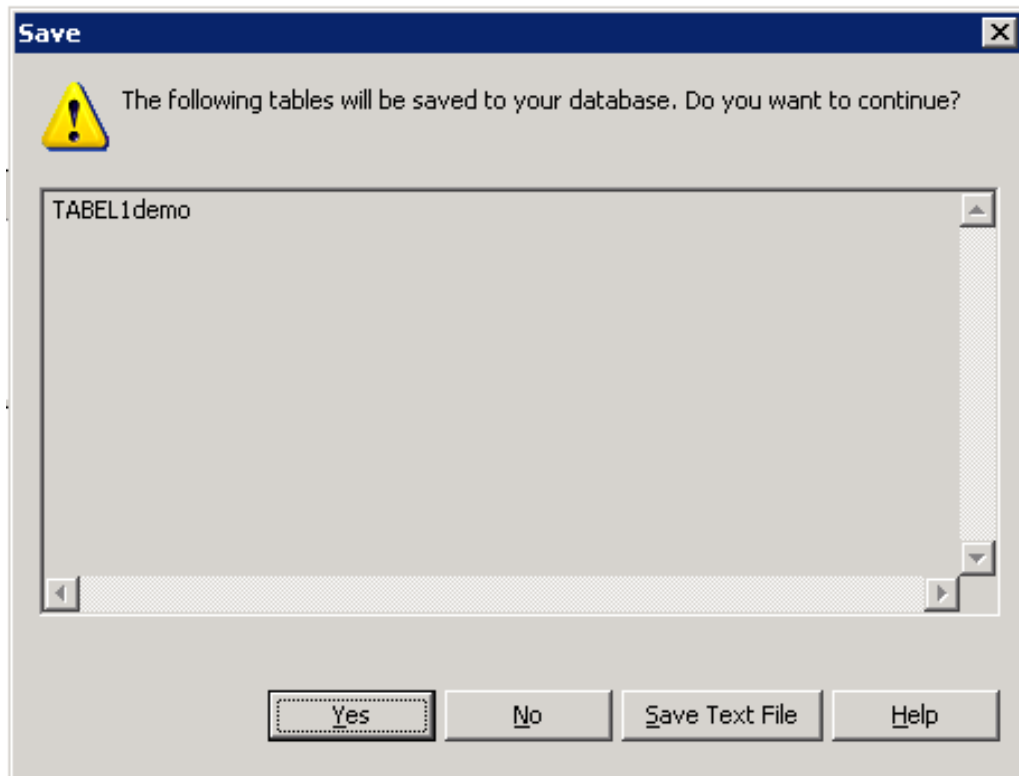
Gambar 4.20 Membuat Hubungan Antar Tabel

Disini, kita akan memilih pubId dari tabel TABLE1demo dan menggesernya ke kolom pub_id dari tabel Titles. Operasi ini akan membuka sebuah kotak dialog baru, di mana Anda bisa menekan OK untuk memastikan pembuatan hubungan antartabel.



Gambar 4.21 Membuat Hubungan Antar Tabel

Saat Anda menekan OK, program akan menampilkan kotak dialog yang memberi tahu Anda bahwa tabel baru akan dibuat secara fisik di database.



Gambar 4.22 Menyimpan Tabel

Pada contoh kita ini, kita tidak perlu menyimpan perubahan yang telah dibuat, jadi tekanlah No.

Kesimpulan

Pada bab ini Anda telah melihat bahwa diagram database bisa sangat bermanfaat untuk membuat model dan prototipe dari database, selain membantu Anda dalam membuat dan memelihara tabel-tabel. Pada bab berikutnya, kita akan membahas mengenai sarana dasar yang digunakan untuk bekerja dengan SQL Server, yaitu Query Analyzer.

BAB 5

Query Analyzer

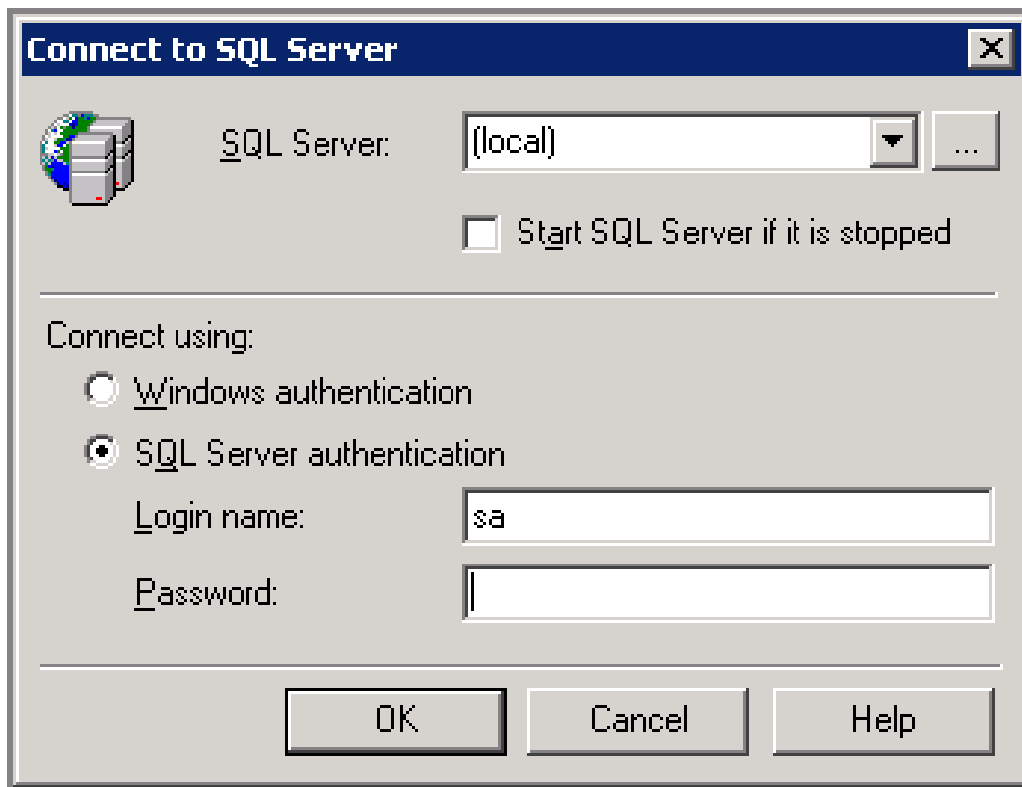
Dalam bab ini, kita akan memeriksa salah satu alat SQL Server dan memahami cara kerjanya. SQL Server Query Analyzer adalah alat antarmuka pengguna grafis yang melakukan beberapa fungsi yang terkait dengan pemulihan kueri dan pengoptimalan dalam database. Fitur-fitur ini meliputi:

- Query database SQL Server menggunakan perintah SQL untuk menampilkan teks atau grid.
- Tunjukkan representasi grafis dari langkah-langkah untuk menyelesaikan survei
- Melakukan analisis kinerja indeks
- Gunakan bantuan online untuk bahasa Transact-SQL
- Jalankan skrip Transact-SQL dari prosedur tersimpan

Query Analyzer adalah pilihan terbaik jika Anda ingin berlatih menggunakan bahasa Transact-SQL karena Anda dapat menjalankan perintah dan langsung melihat hasilnya. Selain itu, Query Analyzer menyediakan akses langsung ke pernyataan Transact-SQL jika Anda mengalami masalah dengan sintaks.

Berhubungan dengan Server

Setelah program dimulai, sebuah dialog akan muncul di mana Anda harus menentukan server yang ingin Anda sambungkan. Dialog ini juga dapat diakses melalui File | Connect.



Gambar 5.1 Menghubungkan ke Server

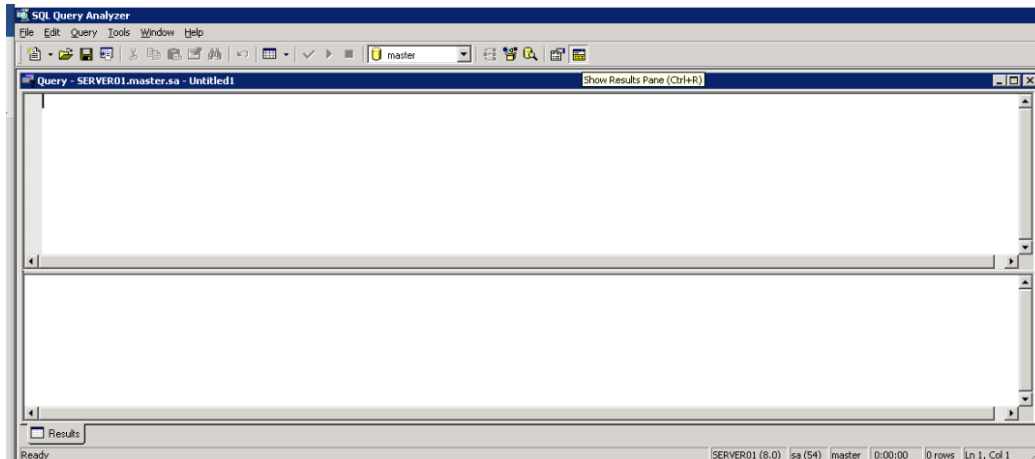
Pada kotak SQL Server, ketikkanlah nama dari server yang diinginkan, atau klik pada tombol Browse (tombol dengan tiga titik) untuk mengakses daftar server network. (Windows 95 tidak memiliki pilihan Browse).

Pilihlah metode autentikasi pada panel Connection Information. Pilihlah Use Windows NT Authentication untuk melakukan hubungan dengan user ID dan password Windows NT atau pilihlah SQL Sever Authentication untuk mengetikkan user ID dan password yang dikenali oleh SQL Server.

Dalam hal ini, satu-satunya server yang tersedia adalah ntserver. Layar kerja dari Query Analyzer akan muncul setelah Anda menekan OK.

Layar Kerja

Ada dua panel pada layar kerja Query Analyzer. Jika Anda hanya melihat satu panel, tekanlah tombol Hide Results (di sebelah field Data base), lalu bukalah panel kedua. Pada panel atas yaitu Query Panel, ketikkanlah perintah SQL atau skrip yang ingin dijalankan. Panel bawah menampilkan hasil dari query Anda atau pesan error dan pesan peringatan.



Gambar 5.2 Layar Kerja

Toolbar memiliki dua tombol yang setara dengan pilihan menu. Ada juga daftar dari data base yang terdapat di dalam server. Anda bisa memilih database dengan mengklik pada tanda panah pada kotak data base untuk membuka daftar lalu nama dari data base yang diinginkan. Pada contoh ini, kita akan bekerja dengan data base Pubs. Kita akan menggunakan beberapa perintah SQL untuk melihat bagaimana cara kerja dari Query Analyzer. Perintah-perintah ini dijelaskan lebih detail pada bab berikutnya dan sintaks lengkapnya dapat dilihat pada Lampiran A. Sebelum melanjutkan, pilihlah database Pubs pada field Database dan jadikanlah ini database default.

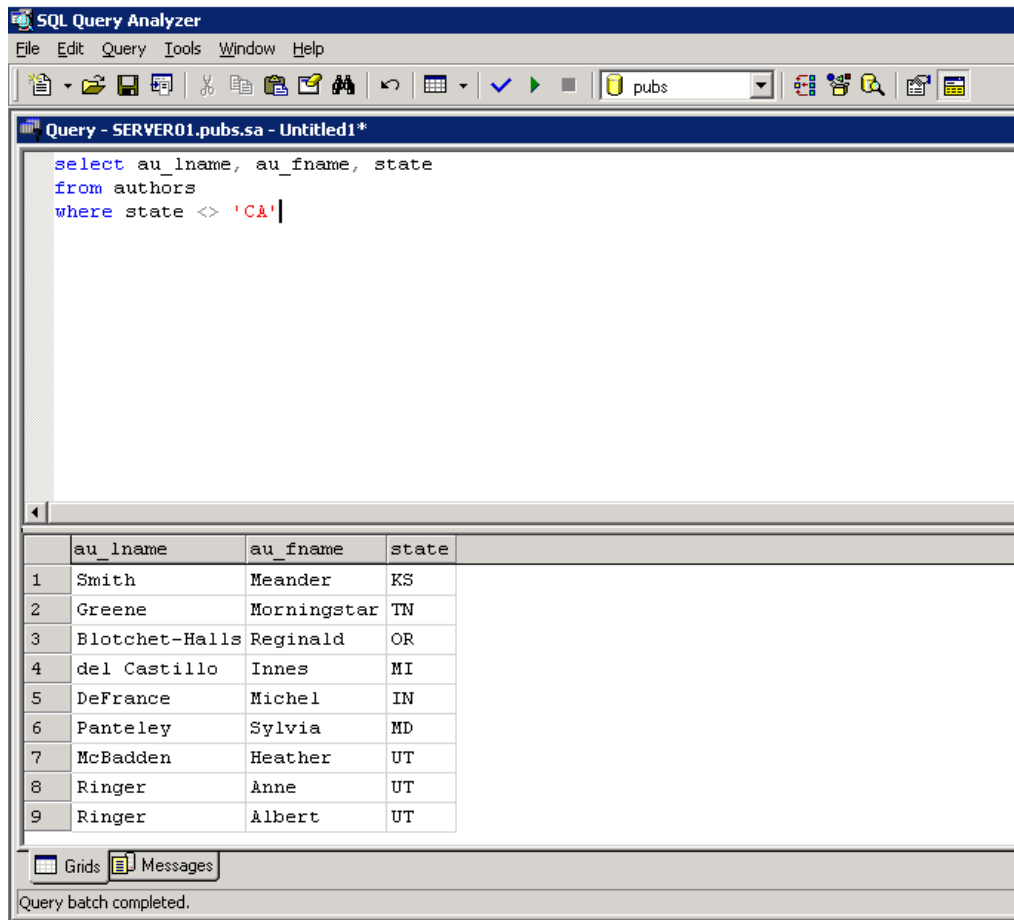
Mengubah Ukuran Panel

Anda bisa memperbesar atau memperkecil ukuran kedua panel dengan mengklik dan menggeser baris pembagi yang memisahkannya. Sewaktu memposisikan kursor pada baris pembagi, format akan berubah. Anda lalu bisa menggeser dan mengubah ukuran jendelanya.

Menjalankan Perintah dengan Beberapa Baris

Perintah SQL bisa dibagi menjadi beberapa baris agar mudah dibaca. Untuk menuju ke baris berikutnya, tekanlah Enter di akhir baris. Setelah Anda selesai mengetikkan seluruh perintah, query bisa dijalankan dengan menggunakan salah satu cara yang dijelaskan pada bagian berikutnya.

Perintah yang ditampilkan di bawah ini berfungsi memilih field au_lname, au_fname, dan State dari tabel Authors hanya untuk record-record yang memiliki field State selain "CA".

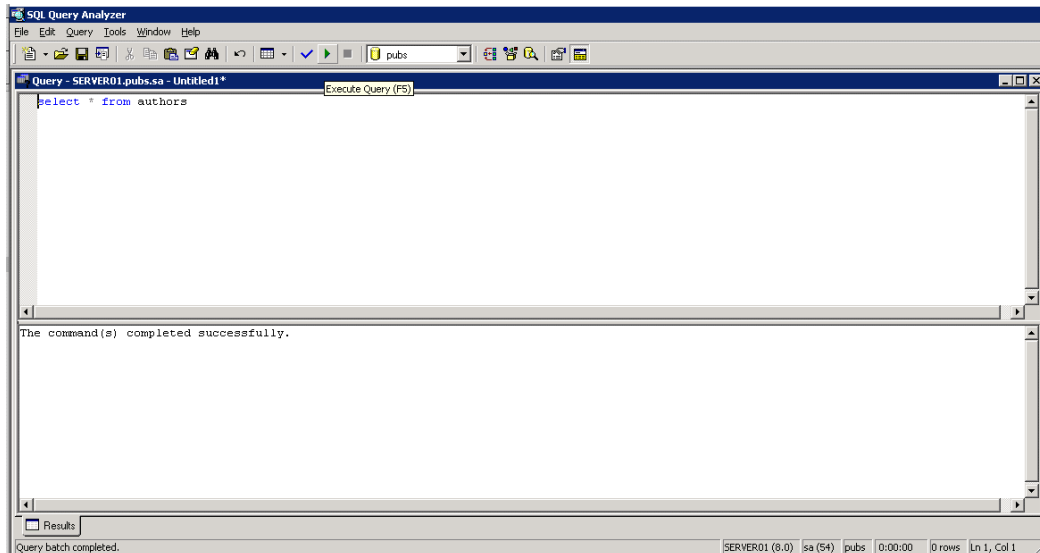


Gambar 5.3 Menjalankan Perintah

Pada contoh ini, kita tidak mempedulikan apakah isi field harus mengandung huruf kapital; semua kemunculan "ca " atau "CA"

Menjalankan Query

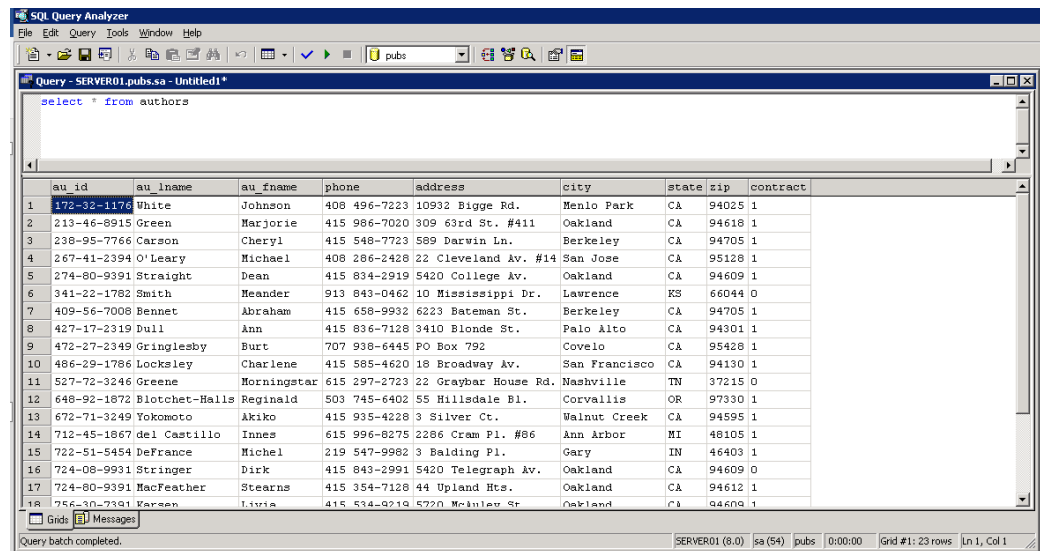
Pastikan Anda telah membuat database Pubs sebagai database default. Pada Query Panel, ketikkanlah perintah `select* from authors`. Perhatikan pada saat Anda mengetik, kata "select" dan "from" akan menjadi berwarna biru. Ini disebabkan karena kata-kata ini adalah bagian dari sintaks perintah SELECT, yang merupakan perintah utama yang digunakan untuk mengambil field dan baris pada database. Authors adalah salah satu tabel yang terdapat di dalam database Pubs. Untuk menjalankan dan melihat hasil dari query, Anda bisa menggunakan tombol Execute Query seperti ditampilkan pada di bawah ini, pilihan Execute Query pada menu Query, atau kombinasi tombol `Ctrl +e`.



Gambar 5.3 Menjalankan Query

Melihat Query dengan Grid

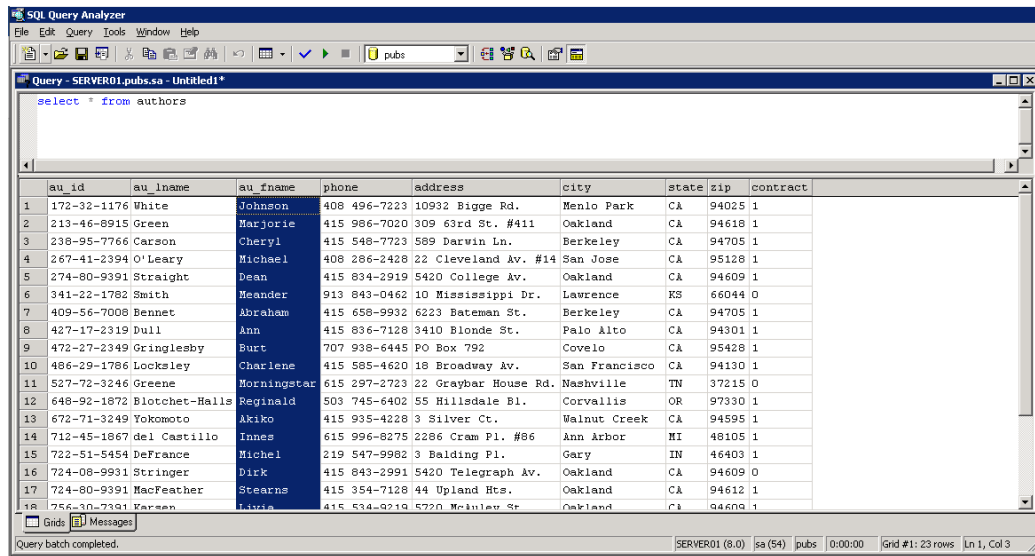
Daripada menggunakan tombol Execute Query, Anda bisa menggunakan tombol atau pilihan Execute Query Into Grid. Tombol ini memberikan hasil yang sama, tetapi ia menampilkan data dalam format tabular, termasuk nama dari kolom-kolom, serta memanfaatkan ruang pada layar, seperti pada gambar 5.4 ini.



Gambar 5.4 Melihat Query dengan Grid

Bekerja dengan grid jauh lebih menarik karena kita bisa mengatur lebar dari kolom dan mengorganisir urutan tampilan. Untuk melakukan ini, klik dengan tombol mouse kanan pada salah satu kolom untuk menampilkan menu shortcut, lalu pilihlah Sort Column. Di

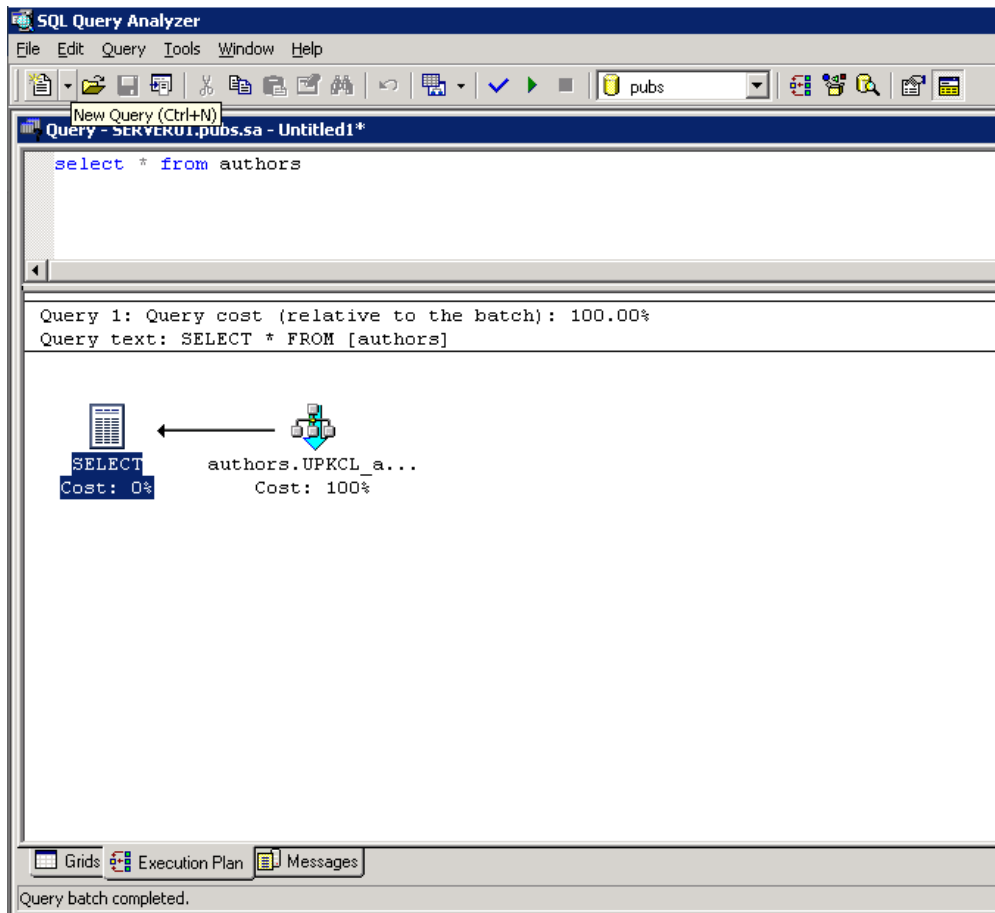
bawah ini adalah contoh sebelumnya setelah diorganisir berdasarkan kolom au_lname.



Gambar 5.5 Sort Query dengan Grid

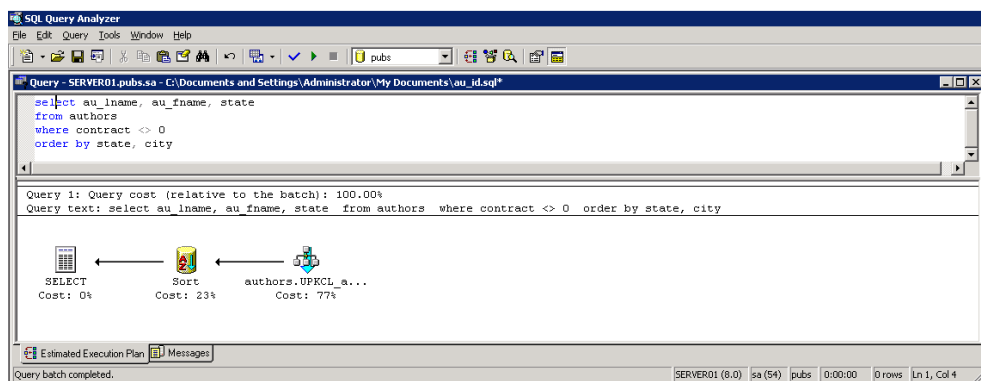
Melihat Query Execution Plan

Salah satu tugas terpenting dari Query Analyzer adalah analisis kueri. Dari tombol Survey Execution Plan kita bisa melihat semua langkah dan prosedur dalam melakukan survey. Beberapa instruksi memerlukan beberapa operasi logis dan fisik yang memerlukan perangkat keras dan waktu CPU. Jika Anda ingin melihat langkah-langkah eksekusi kueri secara grafis, Anda perlu mengklik tombol Rencana Eksekusi Kueri. Bidang hasil menunjukkan simbol dari operasi yang dilakukan.



Gambar 5.6 Melihat Query Execution Plan

Dalam kasus ini, tabel Pubs .authors telah dibaca seluruhnya. Langkah-langkah yang lebih banyak dibutuhkan apa bila perintah menjadi semakin kompleks. Gambar berikutnya menampilkan perintah `SELECT` dengan dua klausa tambahan. Klausa `WHERE` berfungsi menyaring record-record di mana field `Contract` tidak bernilai nol, dan klausa `ORDER BY` mengorganisir hasil secara alfabetis berdasarkan negara bagian dan kota.

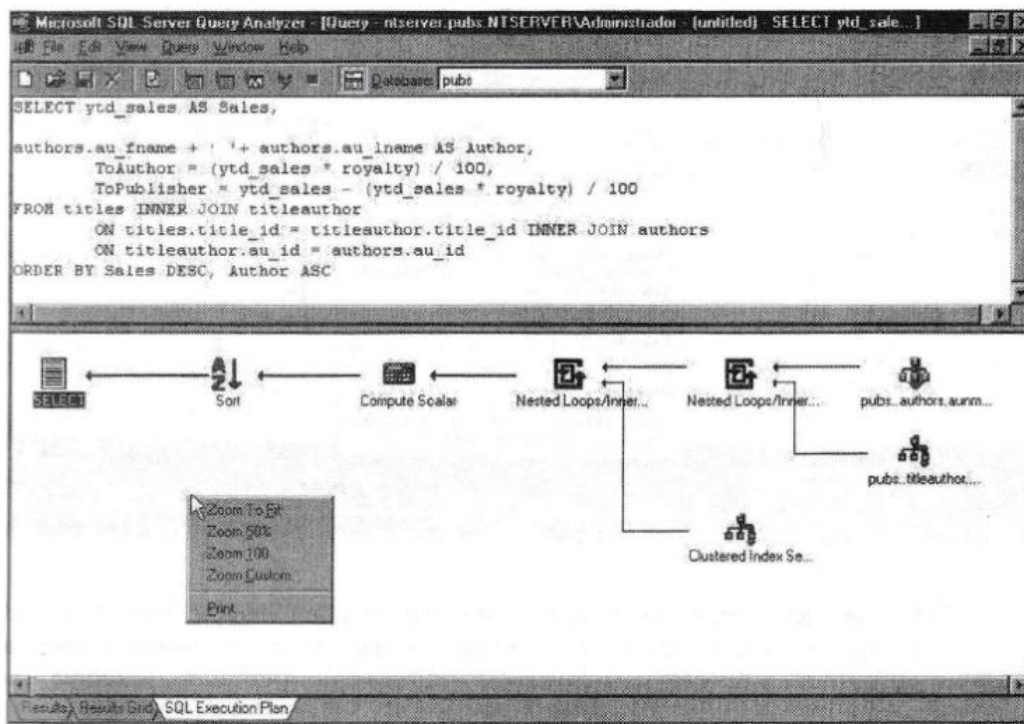


Gambar 5.7 Melihat Query Execution Plan

Perhatikan pada panel bawah, ada dua operasi baru yang ditampilkan.

Mengatur Tampilan Query Execution Plan

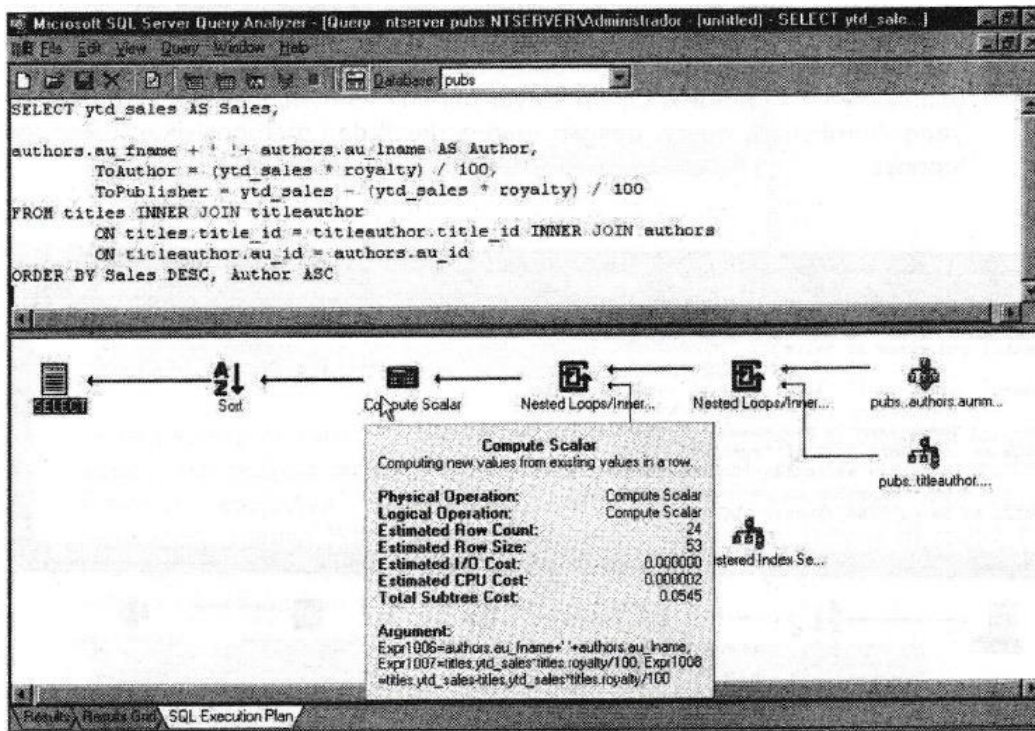
Query yang lebih kompleks membutuhkan beberapa operasi yang tidak selalu ditampilkan secara lengkap pada panel bawah. Untuk melihat tampilan global dari operasi yang dilakukan, Anda bisa memperkecil ukuran dari icon-icon. Operasi ini bisa dilakukan melalui menu shortcut, yang bisa ditampilkan dengan mengklik kanan pada panel bawah. Pada menu ini, pilihan pertama yaitu Zoom To Fit harus dipilih. Layar berikutnya menampilkan semua elemen yang membentuk query, dengan memperkecil dan memposisikan ulang icon- iconnya.



Gambar 5.8 Mengatur Tampilan Query Execution Plan

Menganalisa Query

Setelah survei selesai, panel hasil dapat diakses untuk mendapatkan informasi tentang setiap fungsi yang dilakukan. Tempatkan penunjuk tetikus di atas ikon untuk melihat deskripsi properti dari tindakan yang dilakukan.



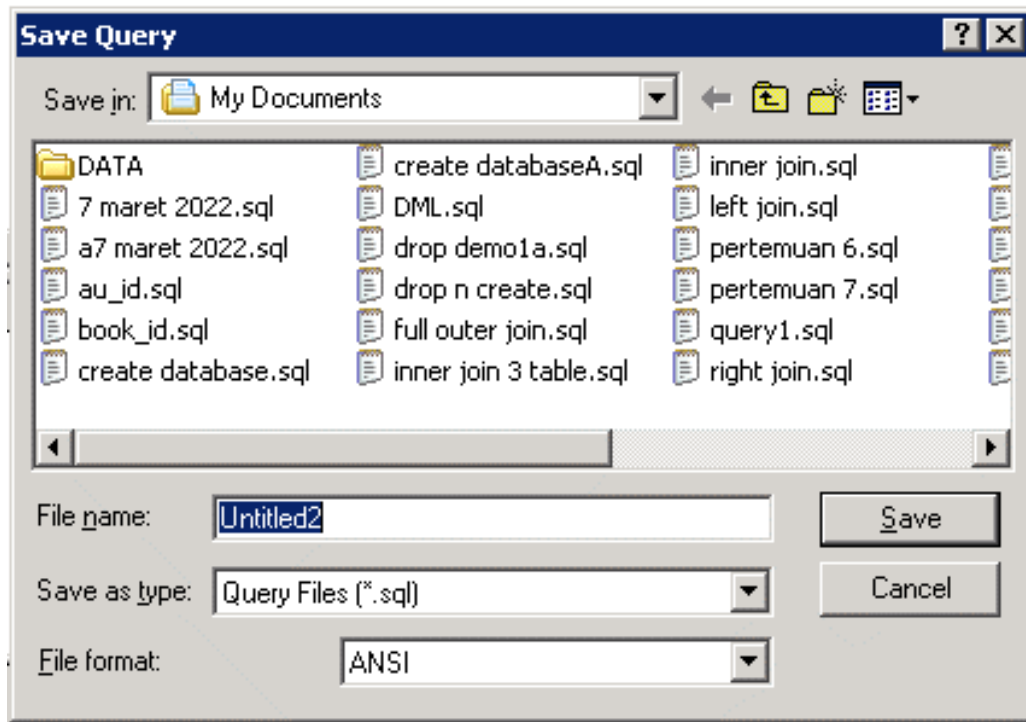
Gambar 5.9 Menganalisa Query

Gambar di atas menunjukkan informasi tentang perhitungan skalar yang dilakukan oleh operasi matematika internal kueri. Jika aksi membutuhkan lebih banyak perhatian, judulnya berwarna merah.

Menyimpan dan Membuka Query

Teks kueri atau skrip dapat ditulis ke disk untuk dieksekusi atau dibuka nanti. Menjalankan kueri yang sama sering kali dapat menghemat banyak waktu Anda.

Klik File | Save untuk menyimpan survei. Pilih tempat Anda ingin menyimpan survei dan masukkan nama survei di bidang Nama file. Di bidang Format file, pilih format karakter yang ingin Anda gunakan untuk menyimpan survei.



Gambar 5.10 Menyimpan dan Membuka Query

Simpanlah query pada kode halaman yang digunakan pada Windows jika ingin dijalankan di kemudian hari dalam operasi batch dari baris perintah. Simpanlah query dalam format ANSI apa bila Anda ingin menggunakannya lagi di dalam Query Analyzer. Simpanlah query dalam format Unicode apabila ia menggunakan karakter asing khusus.

Untuk me-load query, klik pada File I Open. Pilihlah lokasi di mana query ini disimpan, lalu masukkan nama query pada field File Name. Ingat bahwa Anda harus menambahkan ekstensi.sql pada nama file query.

Kelima query terakhir yang disimpan akan ditampilkan di akhir dari menu File.

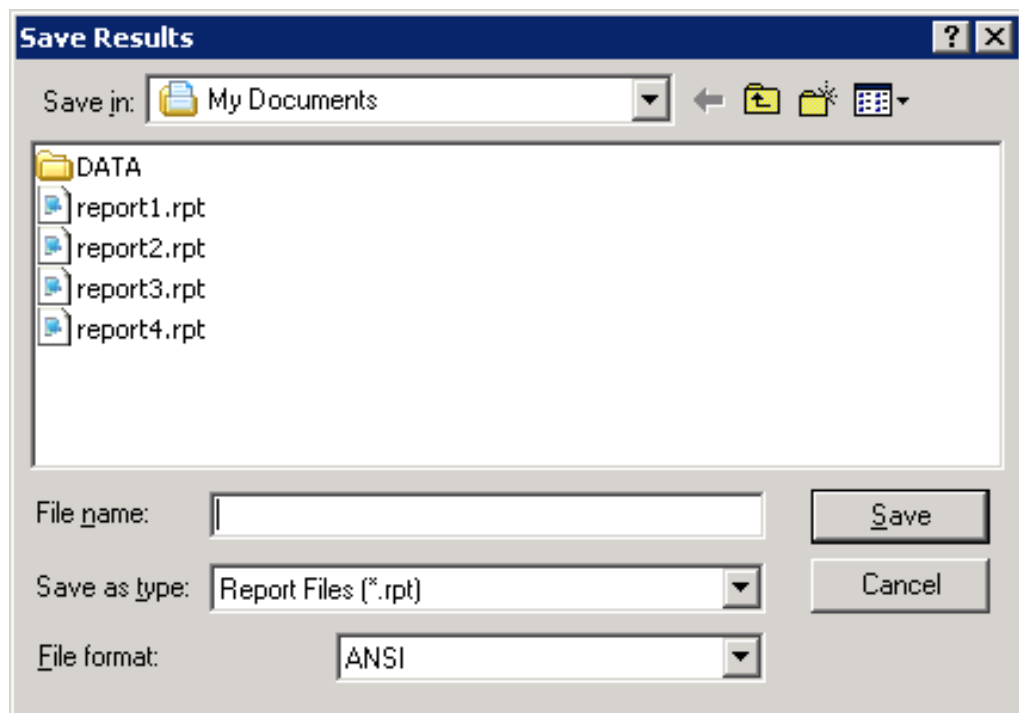
Menyunting Query

Pada menu Edit, Anda bisa melakukan operasi dasar seperti menyunting teks dari query. Dengan perintah Find, Anda bisa menentukan teks yang ingin dicari di dalam baris-baris perintah. Dengan perintah Replace, Anda bisa menggantikan teks. Perintah Go To berfungsi untuk menentukan nomor baris dari query di mana Anda ingin memposisikan kursor. Ingat bahwa beberapa skrip bisa mengandung banyak baris.

Menyimpan Results Panel

Anda dapat menyimpan konten hasil panel jika dalam format raster, lalu mengekspor

data ke program lain. Untuk melakukannya, pilih jendela hasil (tekan Shift+F6 untuk mengaktifkannya). Klik Berkas | Simpan sebagai. Kemudian masukkan nama file ini di bidang Nama file. Di Pemisah Kolom, pilih Dipisahkan Koma (CSV) untuk memisahkan bidang dengan koma, atau pilih Tab Dipisahkan untuk menggunakan karakter tabel. Klik tombol Simpan.



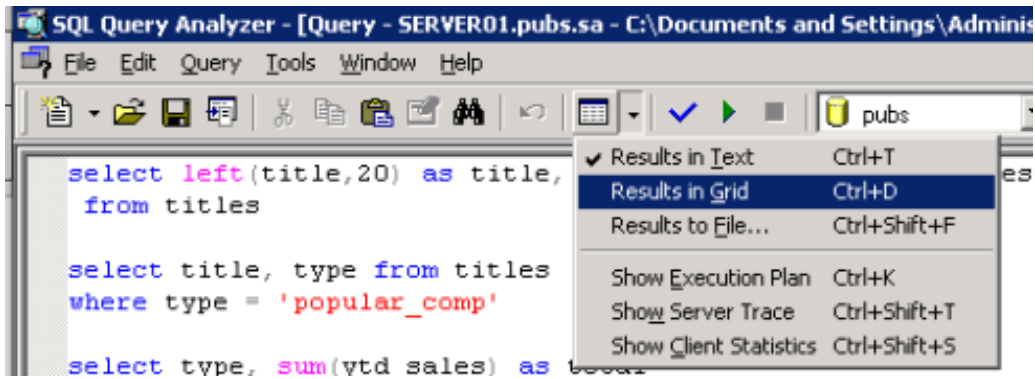
Gambar 5.11 Menyimpan Results Panel

Mencetak Query dan Hasilnya

Anda dapat mencetak teks pertanyaan (skrip) atau konten jendela hasil. Aktifkan panel yang ingin Anda cetak, lalu klik perintah Print pada menu File.

Mengubah Konfigurasi Tampilan

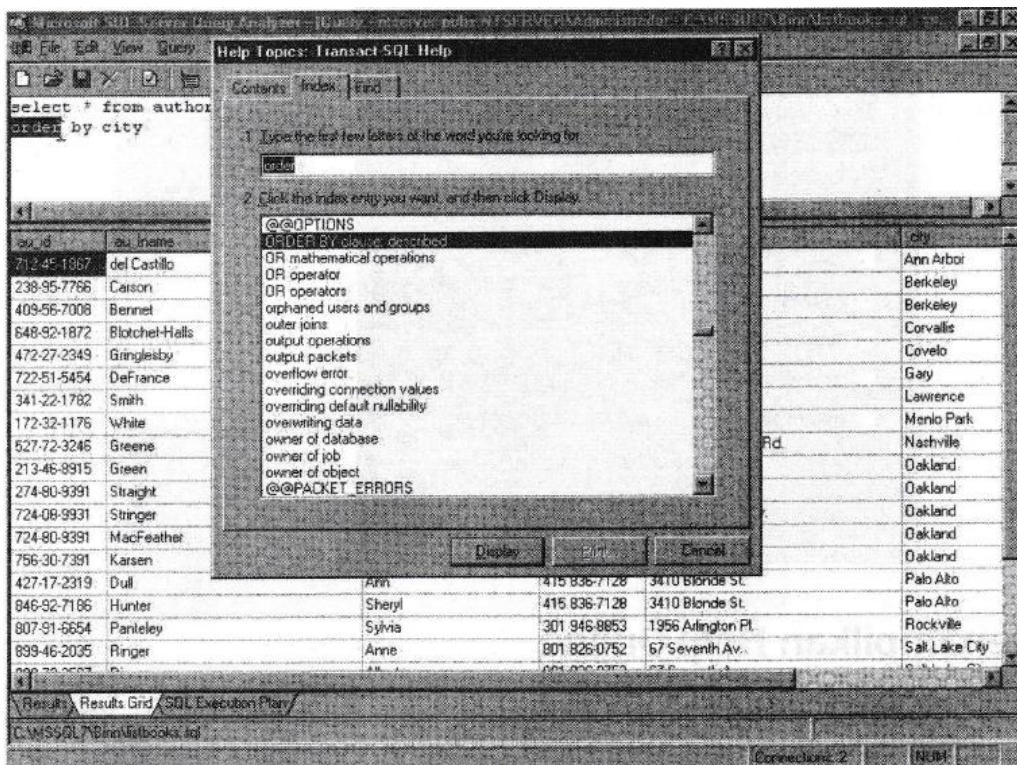
Melalui pilihan Set Options pada menu Query, Anda bisa mengubah format dari panel hasil eksekusi. Klik pada tab yang sesuai lalu ubahlah item-item sesuai keinginan Anda.



Gambar 5.12 Mengubah Konfigurasi Tampilan

Menampilkan Help Online

Jika Anda tidak mengetahui sintaks perintah SQL, pilih kata kunci yang Anda inginkan, lalu tekan Shift+F1. Fungsi ini mengaktifkan bantuan online dan menampilkan deskripsi kata yang dipilih seperti yang ditunjukkan di bawah ini. Query Analyzer adalah salah satu sarana yang paling penting bagi pengembang yang ingin mengakses dan mengubah tabel pada data base melalui perintah-perintah SQL. Operasinya sangat sederhana karena penggunaan antarmuka grafis. Bab berikutnya akan memperkenalkan secara singkat bahasa SQL.



Gambar 5.13 Menampilkan Help Online

Kesimpulan

Query Analyzer adalah salah satu sarana yang paling penting bagi pengembang yang ingin mengakses dan mengubah tabel pada database melalui perintah-perintah SQL. Operasinya sangat sederhana karena penggunaan antarmuka grafis. Bab berikutnya akan memperkenalkan secara singkat bahasa SQL.

BAB 6

Dasar-Dasar SQL

Bahasa SQL adalah alat dasar yang diperlukan untuk mengakses data dalam database relasional. Menguasai bahasa SQL adalah langkah pertama dan terpenting dalam memahami database relasional. Dalam beberapa tahun terakhir, bahasa SQL telah meninggalkan mainframe dan mendarat di desktop. Perubahan ini karena basis data relasional telah mengembangkan platformnya dan sekarang ada versi yang dapat berjalan di desktop dan mainframe. Selain itu, internet menjadi sarana untuk mempopulerkan bahasa ini. Karena bahasa tersebut memiliki struktur yang cocok untuk arsitektur klien-server, semakin banyak aplikasi dan halaman yang mengakses basis data relasional dibangun dalam SQL.

Bab ini tidak berniat menjadikan Anda pakar SQL. Tujuan dari bab ini adalah memberikan pengenalan singkat mengenai bahasa ini, agar Anda bisa melakukan query pertama dengan menggunakan SQL. Kami menyarankan Anda membaca buku yang ditulis khusus mengenai SQL jika Anda ingin menjadi seorang profesional.

SQL Server dan SQL

Walaupun kini ada standar bahasa SQL yang selalu berkembang, sebagian besar dari vendor data base telah-membuat standar mereka sendiri untuk mengeksplorasi karakteristik dari data base, seperti dalam kasus SQL Server dan Transact-SQL. Perintah-perintah yang ditunjukkan pada daftar berikut ini hanyalah beberapa tambahan dari SQL Server ke dalam standar SQL.

ALTER DATABASE	CREATE TRIGGER
ALTER PROCEDURE	CREATE VIEW
ALTER TABLE	DROP DATABASE
ALTER TRIGGER	DROP DEFAULT
ALTER VIEW	DROP INDEX
CREATE CLUSTER	DROP PROCEDURE
CREATE DATABASE	DROP RULE
CREATE DEFAULT	DROP STATISTICS
CREATE INDEX	DROP TABLE
CREATE PROCEDURE	DROP TRIGGER
CREATE RULE	SET
CREATE SCHEMA	TRUNCATE TABLE
CREATE TABLE	

Beberapa dari perintah ini dijelaskan secara rinci pada bab ini dan bab berikutnya. Tetapi kita akan lebih konsentrasi kepada perintah-perintah utama dari bahasa SQL.

Dasar-Dasar SQL

Dengan adanya lusinan perintah dan fungsi, pembaca akan bisa melakukan sebagian besar aktivitas yang berhubungan dengan penelitian dan pemanipulasian database. Tabel di bawah ini menampilkan perintah-perintah dasar yang akan dibahas pada bab ini. Sebagian besar tugas-tugas yang dilakukan oleh perintah-perintah ini bisa dijalankan melalui sarana grafis SQL Server, dan ini merupakan bantuan yang besar untuk para user. Tetapi menguasai dasar dari bahasa ini juga akan sangat membantu di dalam mempelajari SQL Server.

Perintah	Fungsi
SELECT	SUM()
INSERT	AVG()
DELETE	MAX()
UPDATE	MIN()
CREATE TABLE	COUNT()
CREATE VIEW	SYSDATE()
CREATE PROCEDURE	

Jenis-Jenis Deklarasi SQL

Pernyataan atau perintah SQL diklasifikasikan menjadi dua kategori berdasarkan cara kerjanya, yaitu DDL dan DML.

DDL (Data Definition Language)

DDL atau Data Definition Language adalah bagian dari SQL yang digunakan untuk mendefinisikan objek data dan database. Entri dibuat di database SQL Server saat perintah ini digunakan. Kelas ini memiliki perintah berikut:

Perintah	Keterangan
CREATE TABLE	Membuat tabel
CREATE INDEX	Membuat indeks
ALTER TABLE	Mengubah/menyisipkan kolom dalam tabel
DROP TABLE	Menghapus tabel dari database
DROP INDEX	Menghapus indeks
GRANT	Memberikan izin akses kepada user

DML (Data Manipulation Language)

DML atau Data Manipulation Language adalah bagian dari SQL yang digunakan untuk mengambil dan memanipulasi data. Perintah-perintah ini bertanggung jawab atas kueri dan perubahan pada tabel. Tabel berikut mencantumkan beberapa perintah paling penting dalam kategori ini:

Perintah	Keterangan
SELECT	Ini adalah perintah utama dari bahasa ini. Digunakan untuk memilih data dari tabel atau view.
INSERT	Menyisipkan baris dari tabel.
DELETE	Menghapus baris dari tabel.
UPDATE	Mengubah isi dari kolom (field) pada tabel.
COMMIT	Menuliskan perubahan ke dalam disk.
ROLLBACK	Membatalkan perubahan yang dilakukan setelah perintah COMMIT yang terakhir.

Tipe-Tipe Data

Sebelum mempelajari lebih lanjut tentang perintah SQL, Anda perlu mengetahui jenis data apa yang dapat disimpan dalam database SQL Server. Tipe data ini digunakan saat Anda membuat tabel. Anda harus memberikan informasi tertentu, mis. Misalnya nama tabel dan nama tiap field, tipe data yang akan disimpan di dalamnya, dll.

Di SQL Server, data yang disimpan dalam tabel dapat memiliki tipe sesuai dengan tabel berikut.

Integer	
Bit	Integer dengan nilai 0 atau 1.
Int	Nilai integer dengan nilai antara -2^{31} (-2.147.483.648) sampai $2^{31}-1$ (2.147.483.647).
Smallint	2^{15} (-32.768) sampai $2^{15}-1$ (32.767).
Tinyint	Integer antara 0 dan 255.
Decimal atau Numeric	Angka dengan akurasi tetap antara $-10^{38}-1$ sampai $10^{38}-1$.
Money	Nilai data moneter dari -2^{63} (-922.377.203.685.477 ,5808) sampai $2^{63}-1$
	(-922.377.203.685.477,5807), dengan tingkat akurasi hingga sepersepuluhribu dari unit moneter.
Smallmoney	-214.748,3648 sampai 214.748,3647, dengan tingkat akurasi hingga sepersepuluhribu dari unit moneter.
Float	-1.79E+308 sampai 1.79E+308.

Real	-3.40E+38 sampai 3.40E+38.
Datetime	1 Januari 1753 sampai 31 Desember 9999, dengan akurasi sampai 3,33 milidetik.
Smalldatetime	1 Januari 1900 sampai 6 Juni 2079, dengan ketelitian hingga 1 menit.

Rupa-rupa

Cursor	Referensi ke kursor.
Timestamp	Angka eksklus yang dikenali oleh database.
Uniqueidentifier	Pengenal global yang eksklusif.

String

Char	Field tetap dengan ukuran maksimal sebesar 8000 byte.
Varchar	Field tetap dengan ukuran maksimal sebesar 8000 byte.
Text	Variabel dengan ukuran hingga $2^{31}-1$ (2.147.483.647) byte.

Unicode String

Nchar	Karakter unicode dengan ukuran tetap hingga 4000 byte.
Nvarchar	Karakter unicode dengan ukuran bervariasi hingga 4000 byte.
Ntext	Variabel dengan ukuran sampai $2^{30}-1$ (1.073.741.823) byte.

Binary String

Binary	Ukuran tetap hingga 8000 byte.
Varbinary	Ukuran bervariasi hingga 8000 byte.
Image	Ukuran bervariasi hingga $2^{31}-1$ (2.147.483.647) byte.

Ekspresi dan Operator

SQL Server mendukung semua operator yang ditemukan di sebagian besar bahasa pemrograman. Operator beroperasi pada operan, yaitu. H. Elemen dianalisis oleh operator. Ada dua jenis operator, operator unary yang hanya bekerja pada satu operan dan operator biner yang bekerja pada dua atau lebih operan.

Urutan Operator

Jika ekspresi berisi lebih dari satu tipe operator, operator akan dievaluasi dengan urutan sebagai berikut:

- + (positif), - (negatif), ~ (bitwise NOT)
- * (perkalian), / (pembagian), % (modulus)
- + (penjumlahan), + (penggabungan), - (pengurangan)
- =, >, <, >=, <=, <>, !=, !>, !< (perbandingan)
- ^ (bitwise Exclusive OR), & (bitwise AND), | (bitwise OR)

- NOT
- AND
- ALL, ANY , BETWEEN , IN, LIKE, OR, SOME
- = (penugasan)

Operator

Jenis operator berikut ini didukung oleh SQL Server:

Aritmetika	
Perkalian	*
Pembagian	/
Pengurangan	-
Penjumlahan	+
Modulus	%
Karakter	
Penggabungan	
Perbandingan	
Sama dengan	=
Tidak sama dengan	<>
Lebih besar dari	>
Lebih kecil dari	<
Lebih besar atau sama dengan	>=
Lebih kecil atau sama dengan	<=
Tidak kurang dari	!<
Tidak lebih dari	!>
Tidak sama dengan	!=
Logik	
NOT	Membalikkan nilai boolean
AND	Bernilai benar apabila kedua ekspresi benar
OR	Bernilai benar apabila salah satu ekspresi benar
BETWEEN	Bernilai benar apabila operand berada di dalam jangkauan
LIKE	Bernilai benar apabila operand menemukan standar
IN	Bernilai benar apabila beberapa item bernilai benar
SOME	Bernilai benar apabila item apapun bernilai benar
ANY	Bernilai benar apabila beberapa item bernilai benar
ALL	Bernilai benar apabila semua bernilai benar

Bitwise

(operasi manipulasi bit antara dua ekspresi integer)

& (bitwise AND)

| (bitwise OR)

^ (bitwise Exclusive OR)

Unary

+ (positif)

Nilai numerik positif

- (negatif)

Nilai numerik negatif

~ (bitwise NOT)

Mengembalikan nilai komplemen dari angka

Perintah SELECT

Urutan ini dikenal. Meskipun Anda belum menggunakannya dalam buku ini, perintah ini adalah dasar dari semua contoh yang telah kita lihat sejauh ini. Perintah ini adalah dasar dari bahasa SQL.

```
SELECT [ALL | DISTINCT] [TOP n [PERCENT] [WITH TIES]] <select_list>[INTO
new_table][FROM<table_sources>][WHERE <search_conditions>][[GROUP BY[ALL]
group_by_expression [...n]] [HAVING <Search_conditions>] [WITH{CUBE |
ROLLUP}]]][ORDER BY {column_name [ASC | DESC]} [...n]] [ COMPUTE {{AVG |
COUNT | MAX | MIN | SUM} (expression)} [...n] [ BY expression [...n]]][FOR
BROWSE][OPTION (<query_hints> ) ]
<Select_list> :: = {[<table_or_view> | table_alias .]* | {column_name | expression |
IDENTITYCOL | ROWGUIDCOL }[.[AS]column_alias]new_column_name =
IDENTITY(data_type , seed, increment) | GROUPING (column_name) | {table_name |
table_alias}.RANK | column_alias = expression | expression column_name }[ ,...n]
```

Di bawah ini adalah beberapa parameter utamanya:

DISTINCT mengembalikan hanya sebuah salinan dari masing-masing baris yang digandakan, seperti misalnya baris-baris yang memiliki isi yang sama.

Contoh berikut ini mengembalikan hasil dari setiap negara bagian yang ditemukan pada tabel Authors. Walaupun ada beberapa baris dengan negara bagian "ca ", hanya satu yang akan dihasilkan .

Contoh:

```
SOL> select distinct state from authors
```

```
state
```

```
CA
```

IN
KS
MD
MI
OR
TN
UT
(8 row(s) affected)

ALL adalah nilai default . Berfungsi menampilkan semua tabel yang dipilih .

* memilih semua kolom dari tabel atau view yang ditentukan pada FROM.

<expression> memilih sebuah ekspresi, biasanya nama dari kolom atau ekspresi yang dibentuk oleh satu tabel atau ditentukan pada snapshot.

WHERE menyaring baris-baris yang akan ditampilkan oleh query. Tanpa klausa ini, semua baris akan ditampilkan.

GROUP BY mengelompokkan baris-baris yang dipilih berdasarkan nilai dari <expr> untuk setiap baris, dan hanya mengembalikan satu baris dengan informasi untuk setiap kelompok. Sebagai contoh, pada tabel Titles kita bisa menggunakan klausa ini untuk mengelompokkan atau menampilkan informasi mengenai baris editorial.

HAVING menyaring baris-baris yang dihasilkan, menampilkan baris-baris yang ekspresinya bernilai True saja. Biasanya ini digunakan dengan GROUP BY untuk menyaring baris-baris kelompok.

ORDER BY column_name mengurutkan baris-baris yang dihasilkan dengan perintah SELECT berdasarkan kolom yang ditentukan.

COMPUTE menampilkan nilai yang dihasilkan dari penggunaan fungsi yang diterapkan ke dalam proses dari subgrup. Kolom klausa COMPUTE harus muncul pada daftar field-field yang dipilih. Hasil yang diberikan akan ditampilkan sebagai baris tambahan pada hasil query.

<Column_alias> mem buat alias untuk sebuah kolom.

<table_alias> membuat alias untuk sebuah tabel, view, atau snapshot.

UNION menampilkan semua baris-baris eksklusif yang dipilih oleh query.

UNION ALL menampilkan semua baris yang dipilih oleh query, termasuk duplikatnya.

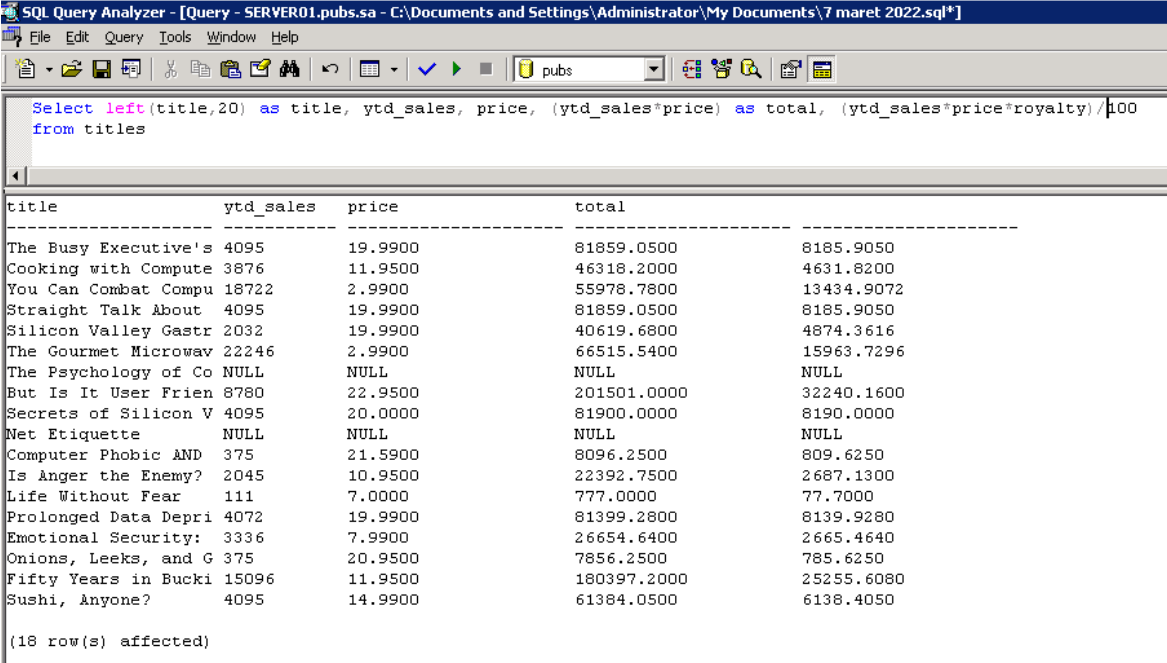
MINUS memilih semua baris yang dipilih oleh query pertama, tetapi bukan oleh query kedua.

column_name adalah ekspresi, biasanya sebuah field di mana baris-baris akan diurutkan. ASC adalah nilai defaultnya dan menggunakan urutan menaik. DESC menggunakan urutan menurun. Semua kolom yang ditentukan oleh ORDER BY atau GROUP BY harus ditentukan pada daftar kolom dan ekspresi dari perintah SELECT.

Contoh dari SELECT

Contoh berikut ini menampilkan penggunaan fungsi LEFT() untuk membatasi jumlah teks yang ditampilkan pada field Title. Contoh ini juga menampilkan penggunaan klausa AS untuk membuat judul untuk kolom yang dibentuk oleh ekspresi tertentu, dan penggunaan kolom-kolom yang dihitung dari kolom lain .

```
Select left(title,20) as title, ytd_sales, price, (ytd_sales*price) as total,  
(ytd_sales*price*royalty)/100  
from titles
```



The screenshot shows the SQL Query Analyzer interface with a query window containing the following SQL statement:

```
Select left(title,20) as title, ytd_sales, price, (ytd_sales*price) as total, (ytd_sales*price*royalty)/100  
from titles
```

The results window displays the following data:

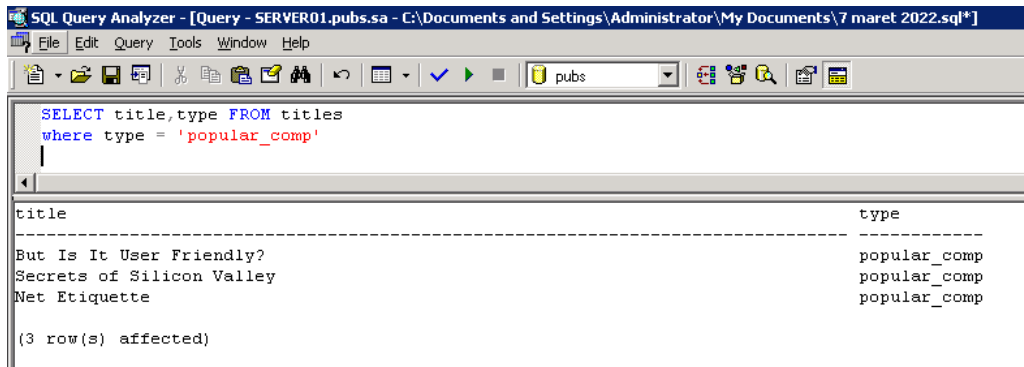
title	ytd_sales	price	total	
The Busy Executive's	4095	19.9900	81859.0500	8185.9050
Cooking with Compute	3876	11.9500	46318.2000	4631.8200
You Can Combat Compu	18722	2.9900	55978.7800	13434.9072
Straight Talk About	4095	19.9900	81859.0500	8185.9050
Silicon Valley Gastr	2032	19.9900	40619.6800	4874.3616
The Gourmet Microwav	22246	2.9900	66515.5400	15963.7296
The Psychology of Co	NULL	NULL	NULL	NULL
But Is It User Frien	8780	22.9500	201501.0000	32240.1600
Secrets of Silicon V	4095	20.0000	81900.0000	8190.0000
Net Etiquette	NULL	NULL	NULL	NULL
Computer Phobic AND	375	21.5900	8096.2500	809.6250
Is Anger the Enemy?	2045	10.9500	22392.7500	2687.1300
Life Without Fear	111	7.0000	777.0000	77.7000
Prolonged Data Depri	4072	19.9900	81399.2800	8139.9280
Emotional Security:	3336	7.9900	26654.6400	2665.4640
Onions, Leeks, and G	375	20.9500	7856.2500	785.6250
Fifty Years in Bucki	15096	11.9500	180397.2000	25255.6080
Sushi, Anyone?	4095	14.9900	61384.0500	6138.4050

(18 row(s) affected)

Gambar 6.1 Contoh Select

Contoh berikutnya menampilkan penggunaan klausa WHERE untuk menyaring record-record khusus untuk jenis buku komputer.

```
SELECT title,type FROM titles  
where type = 'popular_comp'
```



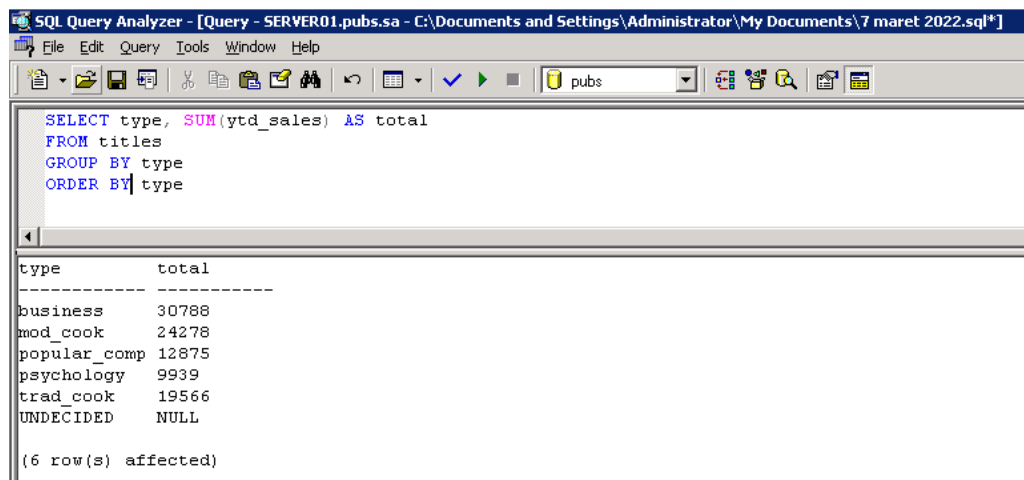
Gambar 6.2 Contoh Select From

Contoh berikut menampilkan penggunaan GROUP BY dan ORDER BY untuk menampilkan hasil penjualan dari masing-masing jenis buku.

```

SELECT type, SUM(ytd_sales) AS total
FROM titles
GROUP BY type
ORDER BY type

```

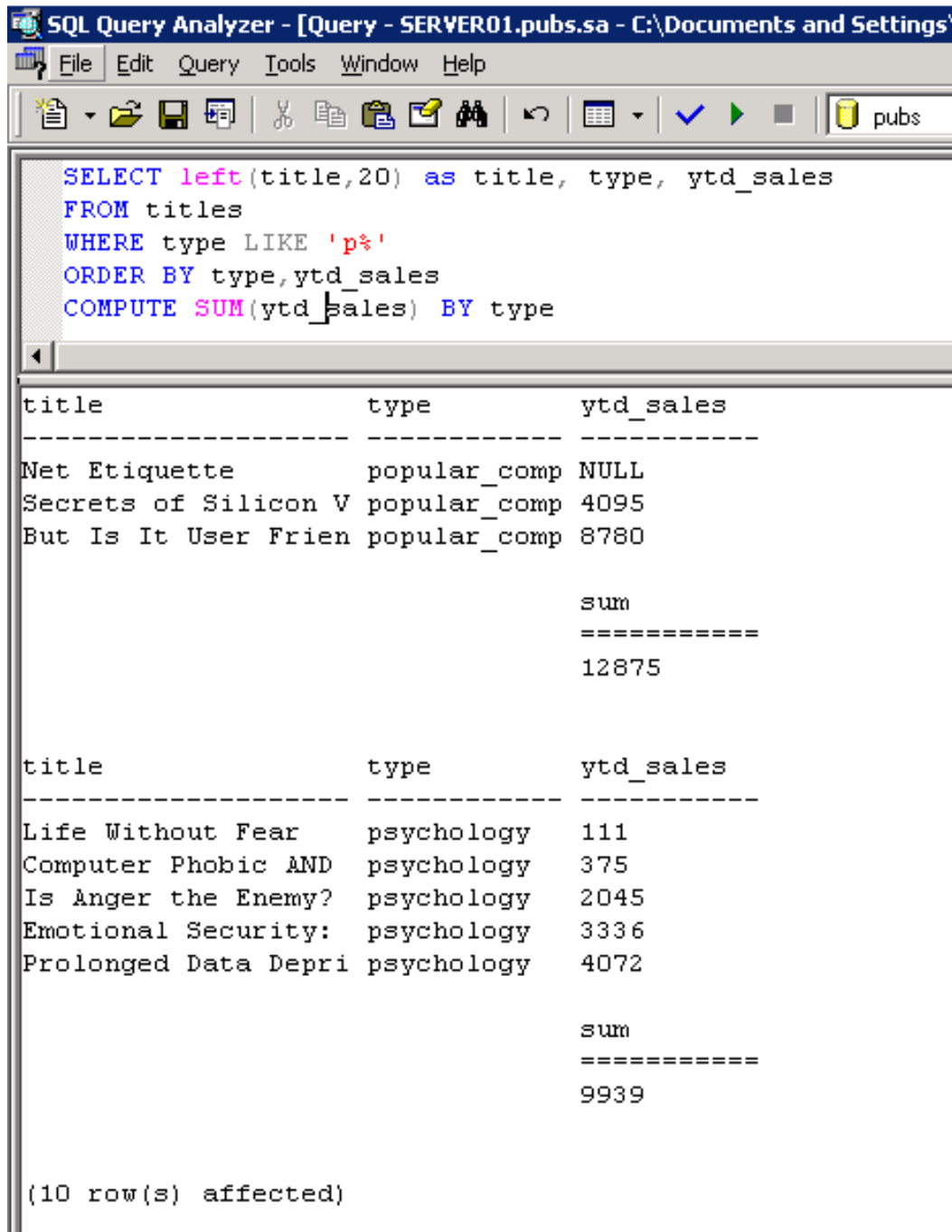


Gambar 6.3 Contoh GROUP BY dan ORDER BY

Contoh berikut menampilkan penggunaan klausa COMPUTE untuk membuat total untuk masing-masing jendela buku yang diawali dengan huruf "p". Klausa LIKE menggunakan karakter "%" sebagai wildcard, yang dalam contoh ini akan menerima semua karakter setelah huruf "p". Pada perintah ini, semua jenis buku dikelompokkan berdasarkan kolom Type dan diletakkan dalam urutan menaik berdasarkan penjualan tahunan. Klausa COMPUTE menambahkan field ytd_sales dari semua record yang memiliki tipe yang sama.

Perintah ini menunjukkan nilai ini setelah menampilkan kelompok.

```
SELECT left(title,20) as title, type, ytd_sales
FROM titles
WHERE type LIKE 'p%'
ORDER BY type,ytd_sales
COMPUTE SUM(ytd_sales) BY type
```



Gambar 6.4 Contoh klausa COMPUTE

Fungsi

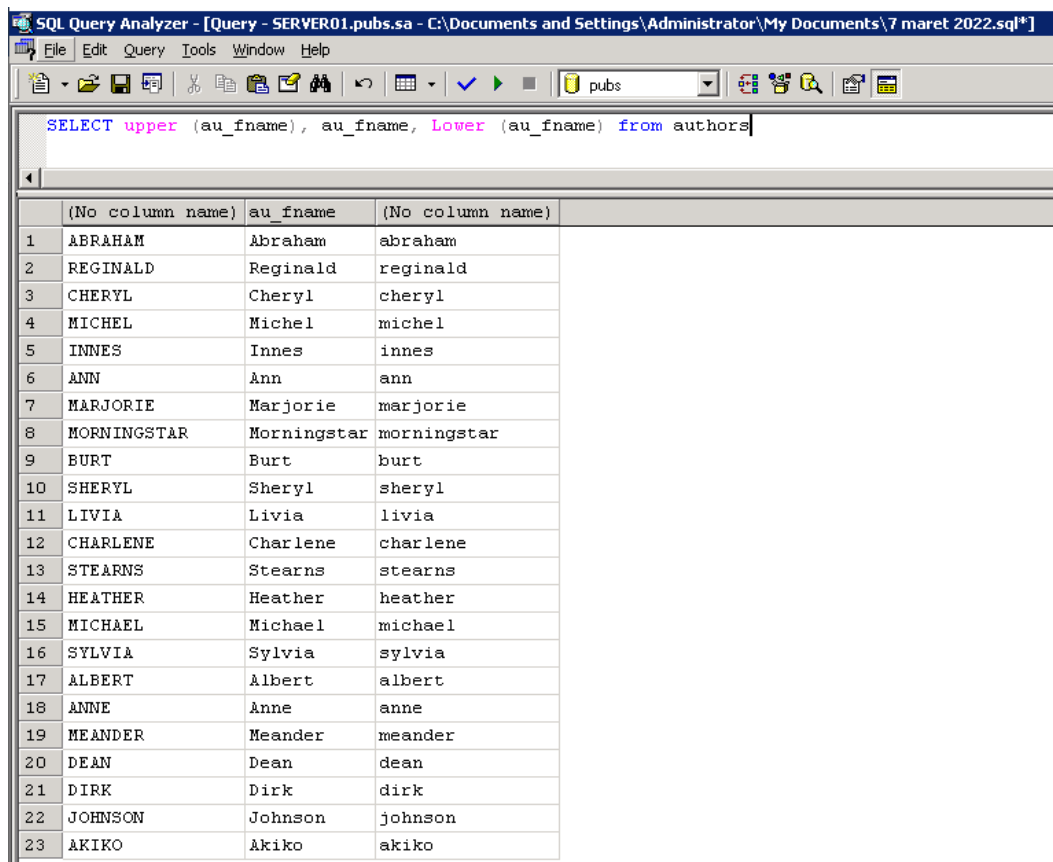
Fungsi SQL adalah sebuah rutin yang melakukan operasi tertentu dan mengembalikan suatu hasil. Fungsi sama seperti prosedur, kecuali bahwa prosedur tidak mengembalikan suatu hasil. Fungsi bisa menerima argumen yang akan digunakan di dalam proses.

Sintaks dasar dari sebuah fungsi adalah:

Funcname (arg1, arg2, ...ARGn)

Funcname adalah nama dari fungsi, seperti misalnya ABS, LOWER, dan sebagainya. arg1 dan arg2 adalah parameter opsional untuk fungsi.

Fungsi-fungsi SQL dibagi ke dalam fungsi satu baris atau skalar, dan fungsi grup atau aggregate. Fungsi satu baris mengembalikan hasil dari setiap baris di dalam bentuk tabel atau view. Fungsi grup mengembalikan hanya satu hasil ke dalam grup baris. Sebagai contoh, fungsi LOWER() mengubah argumennya menjadi huruf kecil semua, dan fungsi UPPER() mengubah argumennya menjadi huruf besar semua. Pada contoh berikut ini, apabila fungsi dijalankan, ia akan mengeksekusi semua baris yang terdapat pada tabel.



The screenshot shows a window titled "SQL Query Analyzer - [Query - SERVER01.pubs.sa - C:\Documents and Settings\Administrator\My Documents\7 maret 2022.sql*]". The query editor contains the following SQL statement:

```
SELECT upper (au_fname), au_fname, Lower (au_fname) from authors
```

The results are displayed in a table with the following data:

	(No column name)	au_fname	(No column name)
1	ABRAHAM	Abraham	abraham
2	REGINALD	Reginald	reginald
3	CHERYL	Cheryl	cheryl
4	MICHEL	Michel	michel
5	INNES	Innes	innes
6	ANN	Ann	ann
7	MARJORIE	Marjorie	marjorie
8	MORNINGSTAR	Morningstar	morningstar
9	BURT	Burt	burt
10	SHERYL	Sheryl	sheryl
11	LIVIA	Livia	livia
12	CHARLENE	Charlene	charlene
13	STEARNS	Stearns	stearns
14	HEATHER	Heather	heather
15	MICHAEL	Michael	michael
16	SYLVIA	Sylvia	sylvia
17	ALBERT	Albert	albert
18	ANNE	Anne	anne
19	MEANDER	Meander	meander
20	DEAN	Dean	dean
21	DIRK	Dirk	dirk
22	JOHNSON	Johnson	johnson
23	AKIKO	Akiko	akiko

Gambar 6.5 Contoh Funcname

Fungsi-fungsi ini bisa digunakan pada perintah **SELECT** yang mengandung klausa **WHERE**, **START WITH**, atau **CONNECT BY**. Tetapi fungsi-fungsi ini tidak bisa digunakan dengan **GROUP BY**.

Fungsi group memproses sekelompok baris dari tabel dan mengembalikan hasilnya. Secara default, sebagian besar dari fungsi-fungsi ini mengeksekusi semua baris dari tabel. Fungsi ini menerima parameter berikut ini:

DISTINCT Menentukan bahwa fungsi hanya mengeksekusi baris dengan nilai-nilai yang berbeda.

ALL Membuat fungsi mengeksekusi semua baris, termasuk baris yang mengandung data duplikat. Ini adalah parameter default.

Dengan pengecualian fungsi **COUNT()**, semua fungsi lain mengabaikan nilai **Null**.

Fungsi-fungsi grup ini adalah:

AVG Mengembalikan nilai rata-rata aritmetik dari *n*.

Sintaks: **AVG** ([**DISTINCT** | **ALL**] *n*)

COUNT Mengembalikan jumlah baris dari query. *Expr* merepresentasikan field atau ekspresi non-Null

Sintaks: **COUNT** ({* | [**DISTINCT** | **ALL**] *expr*})

MAX Mengembalikan nilai maksimal dari *expr*.

Sintaks: **MAX** ([**DISTINCT** | **ALL**] *expr*)

MIN Mengembalikan nilai minimal dari *expr*.

Sintaks: **MIN** ([**DISTINCT** | **ALL**] *expr*)

Berikut ini adalah beberapa contoh dari fungsi-fungsi di atas. Sebelumnya Anda harus membuka Query Analyzer dan memilih database Pubs. Perhatikan isi yang lengkap dari tabel **Titles** yang digunakan pada contoh-contoh ini; tabel ini mengandung 18 record.

	title_id	title	type	pub_id	price	advance	royalty	ytd_sales
1	BU1032	The Busy Executive's Database Guide	business	1389	19.9900	5000.0000	10	4095
2	BU1111	Cooking with Computers: Surreptitious ...	business	1389	11.9500	5000.0000	10	3876
3	BU2075	You Can Combat Computer Stress!	business	0736	2.9900	10125.0000	24	18722
4	BU7832	Straight Talk About Computers	business	1389	19.9900	5000.0000	10	4095
5	MC2222	Silicon Valley Gastronomic Treats	mod_cook	0877	19.9900	.0000	12	2032
6	MC3021	The Gourmet Microwave	mod_cook	0877	2.9900	15000.0000	24	22246
7	MC3026	The Psychology of Computer Cooking	UNDECIDED	0877	NULL	NULL	NULL	NULL
8	PC1035	But Is It User Friendly?	popular_comp	1389	22.9500	7000.0000	16	8780
9	PC8888	Secrets of Silicon Valley	popular_comp	1389	20.0000	8000.0000	10	4095
10	PC9999	Net Etiquette	popular_comp	1389	NULL	NULL	NULL	NULL
11	PS1372	Computer Phobic AND Non-Phobic Individ...	psychology	0877	21.5900	7000.0000	10	375
12	PS2091	Is Anger the Enemy?	psychology	0736	10.9500	2275.0000	12	2045
13	PS2106	Life Without Fear	psychology	0736	7.0000	6000.0000	10	111
14	PS3333	Prolonged Data Deprivation: Four Case ...	psychology	0736	19.9900	2000.0000	10	4072
15	PS7777	Emotional Security: A New Algorithm	psychology	0736	7.9900	4000.0000	10	3336
16	TC3218	Onions, Leeks, and Garlic: Cooking Sec...	trad_cook	0877	20.9500	7000.0000	10	375
17	TC4203	Fifty Years in Buckingham Palace Kitchens	trad_cook	0877	11.9500	4000.0000	14	15096
18	TC7777	Sushi, Anyone?	trad_cook	0877	14.9900	8000.0000	10	4095

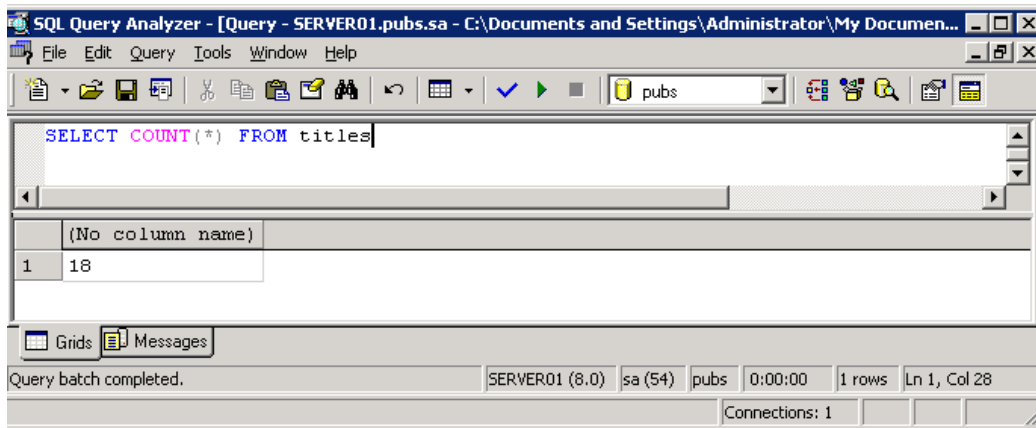
Gambar 6.6 Contoh Fungsi Select

Gambar berikut menunjukkan tiga kolom yang berisi nilai rata-rata, minimum, dan maksimum bidang ytd_sales (jumlah buku yang terjual selama setahun) dari Tabel Judul di database Pubs.

	(No column name)	(No column name)	(No column name)
1	6090	111	22246

Gambar 6.7 Contoh avg, min, max

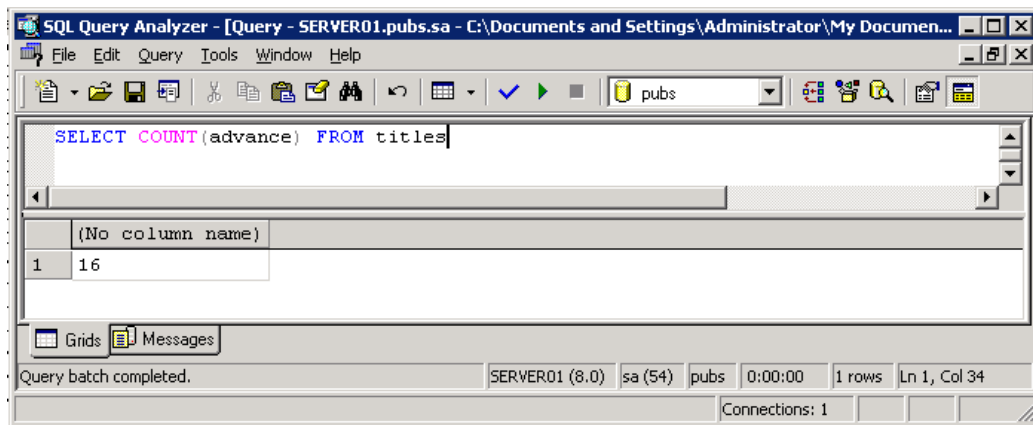
Contoh berikut ini menampilkan jumlah total record di dalam tabel Titles.
 SELECT COUNT(*) FROM titles



Gambar 6.7 Contoh SELECT COUNT

Contoh berikut ini menampilkan baris-baris yang nilainya selain Null pada field Advance.

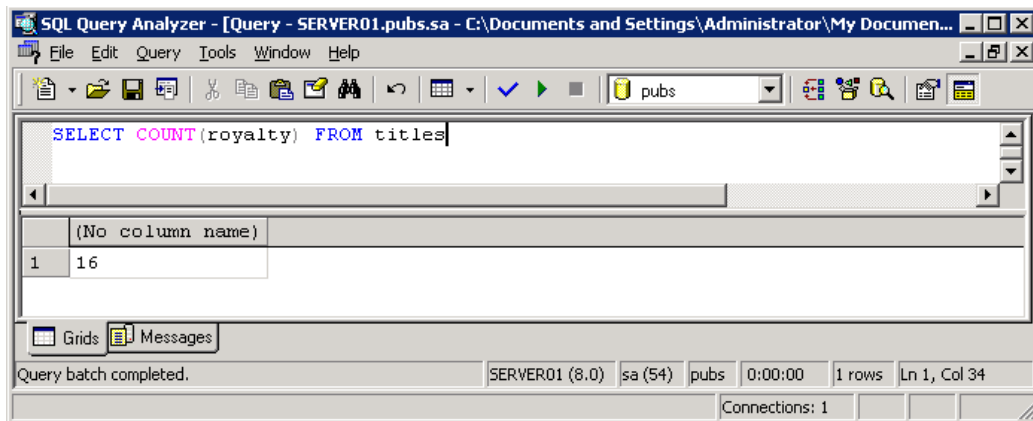
`SELECT COUNT(advance) FROM titles`



Gambar 6.8 Contoh SELECT COUNT

Contoh berikut ini menampilkan berapa banyak buku yang royaltinya telah dibayarkan, yaitu yang barisnya memiliki nilai selain nilai Null pada kolom ini.

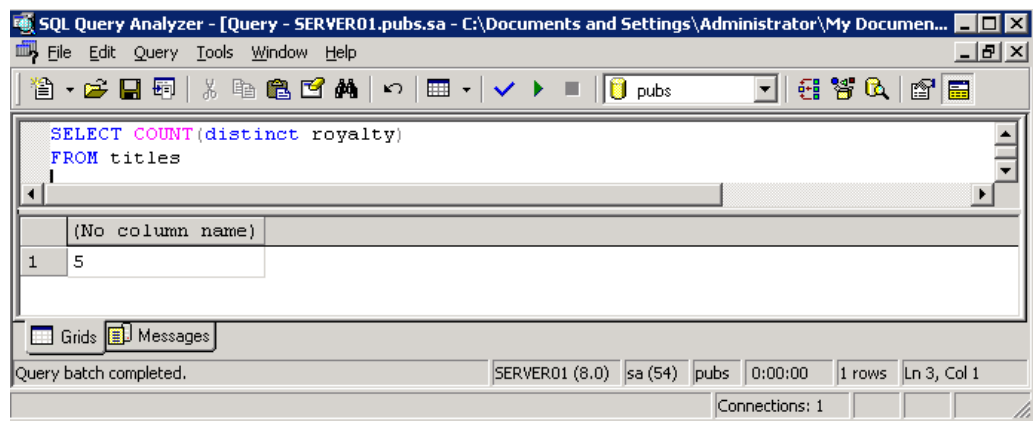
`SELECT COUNT(royalty) FROM titles`



Gambar 6.9 Contoh SELECT COUNT

Contoh berikut ini menampilkan berapa banyak jenis royalti yang ada dan menampilkan jumlah nilai yang berbeda pada kolom Royalty.

```
SELECT COUNT(distinct royalty)
FROM titles
```



Gambar 6.10 Contoh SELECT COUNT

Perintah CREATE TABLE

Perintah CREATE TABLE membuat tabel baru di database. SQL Server memungkinkan hingga 2 miliar tabel di setiap database dan hingga 1024 kolom di setiap tabel. Jumlah baris dalam tabel tidak dibatasi, kecuali ruang memori fisik yang tersedia. Jumlah maksimum byte per baris adalah 8092. Tabel akan dibuat di database yang sedang aktif kecuali jika Anda menentukan nama database yang berbeda. Perintah CREATE TABLE membuat tabel baru di database. SQL Server memungkinkan hingga 2 miliar tabel di setiap database dan hingga 1024 kolom di setiap tabel. Jumlah baris dalam tabel tidak dibatasi, kecuali ruang memori fisik yang tersedia. Jumlah maksimum byte per baris adalah 8092.

Tabel akan dibuat di database yang sedang aktif kecuali jika Anda menentukan nama database yang berbeda.

```
CREATE TABLE table_name({<column_definition>| column_name AS  
computed_column_expression | <table_constraint>}[, ...n])[ON {filegroup |  
DEFAULT}][TEXTIMAGE_ON {filegroup | DEFAULT}]
```

table_name adalah nama dari tabel yang Anda buat.

column_definition menentukan tipe data di dalam kolom. Hingga sebanyak 1024 buah kolom dapat dibuat.

Ini adalah sintaks untuk mendefinisikan kolom:

```
<column_definition> ::= {column_name data_type}[NULL | NOT  
NULL][IDENTITY[(seed, increment)[NOT FOR REPLICATION]  
]][ROWGUIDCOL][<column_constraint>::= [CONSTRAINT constraint_name]  
{PRIMARY KEY | UNIQUE}[CLUSTERED | NONCLUSTERED] [WITH  
[FILLFACTOR = fillfactor]][ON {filegroup | DEFAULT}] | [FOREIGN  
KEY]REFERENCES ref_table [(ref_column)] [NOT FOR REPLICATION]| DEFAULT  
constant_expression | CHECK [NOT FOR REPLICATION] (logical_expression)} [ ...n]
```

IDENTITAS menunjukkan bahwa bidang hanya memiliki satu nilai, yang dihasilkan secara otomatis oleh sistem setiap kali catatan baru dimasukkan. Properti Identity ini bisa diberikan kepada kolom Tinyint, Smallint, Int, Decimal (p,O), atau Numeric (p,O). Hanya satu kolom identitas yang bisa dibuat untuk satu tabel. Batasan DEFAULT tidak bisa digunakan dengan kolom identitas. Anda harus menentukan dasar dan kenaikannya, atau tidak dua-duanya. Jika tidak ditentukan, nilai defaultnya adalah (1,1).

seed adalah nilai yang digunakan untuk baris paling pertama yang di-load ke dalam tabel.

increment adalah nilai kenaikan yang ditambahkan ke dalam nilai identitas dari baris sebelumnya yang di-load.

column_constraint sifatnya opsional dan berfungsi menunjukkan kendali integritas yang diterapkan ke dalam kolom, seperti misalnya isinya tidak boleh kosong.

DEFAULT sifatnya opsional dan berfungsi menunjukkan nilai default untuk kolom.

table_constraint sifatnya opsional dan berfungsi menunjukkan kendali integritas yang diterapkan ke dalam tabel sebagai kunci primer.

Ini adalah sintaks untuk table_constraint:

```
<table_constraint> ::= [CONSTRAINT constraint_name] [{PRIMARY KEY | UNIQUE}
[CLUSTERED | NONCLUSTERED] { ( column[, ...n] ) } [ WITH [FILLFACTOR =
fillfactor] ] [ON {filegroup | DEFAULT} ] ] | FOREIGN KEY [(column[,n])]
REFERENCES ref_table [(ref_column[,n])] [NOT FOR REPLICATION] | CHECK [NOT
FOR REPLICATION] (search_conditions)}
```

Contoh

```
CREATE TABLE demo2
(code int,
name char(20),
state char(2)CONSTRAINT default_name DEFAULT ('CA')
)
```

Perintah INSERT

Perintah INSERT menambahkan baris baru ke tabel dan mengisi kolom dengan nilai tertentu. Anda tidak harus memasukkan konten di semua bidang. Sintaks dasarnya adalah sebagai berikut:

```
INSERT [INTO].{table_source}> { { [(column_list)] VALUES ({DEFAULT |
constant_expression }[,...n] ) | select_statement | execute_statement } | DEFAULT VALUES
}
```

INTO menentukan nama tabel yang menerima data.

column_list adalah daftar kolom yang menerima data. Daftar ini harus diapit dan dipisahkan dengan koma. Apa bila nilai yang ditetapkan pada **VALUES** tidak dalam urutan yang sama dengan kolom, pilihan **column_list** ini harus digunakan untuk menentukan bahwa kolom-kolom akan menerima nilai semacam itu.

VALUES menentukan daftar nilai yang akan disisipkan.

DEFAULT menentukan nilai default untuk kolom. Apabila nilai default tidak didefinisikan, ia akan menggunakan nilai Null.

constant_expression adalah sebuah nilai literal, ekspresi, atau variabel.

Contoh berikut ini menggunakan klausa **DEFAULT** pada field State. Salah satu dari record yang disisipkan menggunakan klausa **DEFAULT VALUES** untuk menghasilkan isi untuk semua field. Perhatikan bahwa karena mereka tidak memiliki isi standar, beberapa field akan mengandung nilai Null.

```

CREATE TABLE demo2
(code int,
name char(20),
state char(2) DEFAULT T('CA')
)

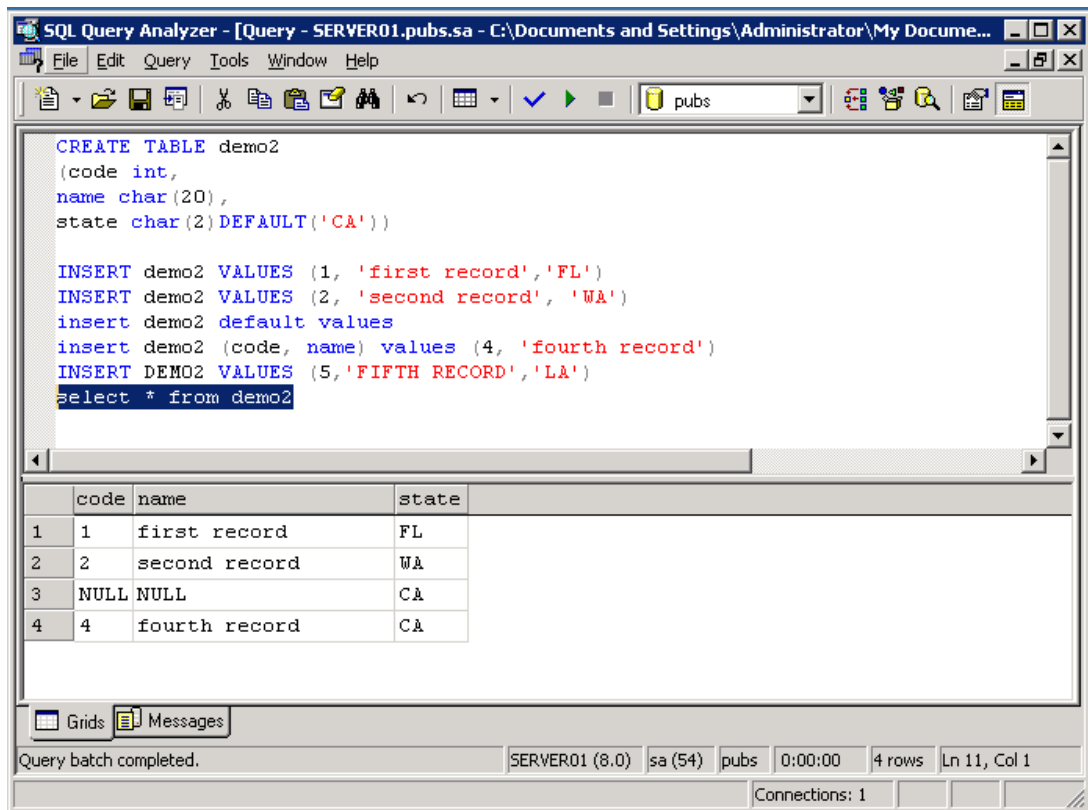
```

```

INSERT demo2 VALUES (1, 'first record', "FL")
INSERT demo2 VALUES (2, 'second record', "WA")
insert demo2 default values
insert demo2 (code, name) values (4, 'fourth record')
INSERT DEMO2 VALUES (5, "FIFTH RECORD", "LA")
select * from demo2

```

Hasilnya adalah:



Gambar 6.11 Contoh CREATE TABLE

Contoh berikut ini menampilkan penggunaan klausa CHECK untuk membatasi informasi yang bisa didistribusikan ke dalam field State. Hanya negara bagian yang terdapat di dalam daftar yang akan diterima di dalam field ini.

```

drop table demo2
CREATE TABLE demo2
(code int,
name char(20),
state char(2) CHECK (STATE IN ("CA","FL","TX","CO"))) DEFAULT('CA')
)

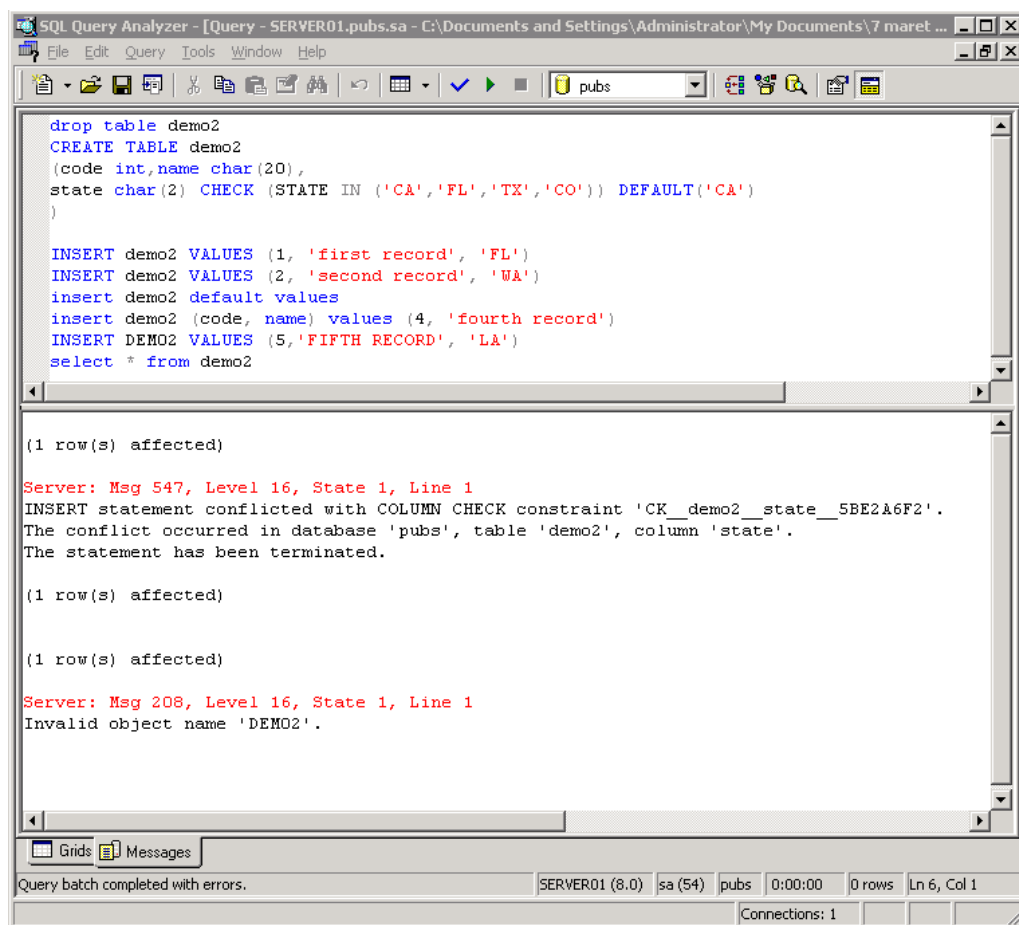
```

```

INSERT demo2 VALUES (1, 'first record', "FL")
INSERT demo2 VALUES (2, 'second record', "WA")
insert demo2 default values
insert demo2 (code, name) values (4, 'fourth record')
INSERT DEMO2 VALUES (5,"FIFTH RECORD", "LA")
select * from demo2

```

Hasilnya adalah:



Gambar 6.12 Contoh DROP TABLE

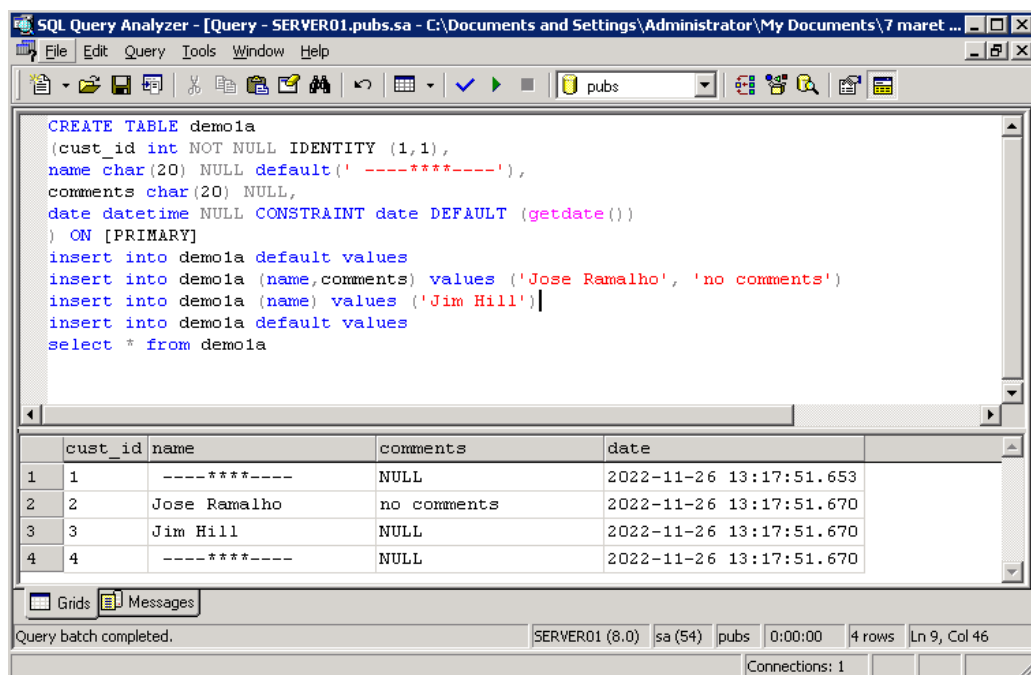
Perhatikan bahwa record kedua dan kelima tidak disertakan karena field State tidak memenuhi syarat yang ditentukan untuk field ini.

```

CREATE TABLE demo1a
(
cust_id int NOT NULL IDENTITY (1,1),
name char(20) NULL default("----***-----"),
comments char(20) NULL,
date datetime NULL CONSTRAINT date DEFAULT (getdate())
) ON [PRIMARY]
insert into demo1a default values
insert into demo1a (name,comments) values ('Jose Ramalho', 'no comments')
insert into demo1a (name) values ('Jim Hill')
insert into demo1a default values
select * from demo1a

```

Perhatikan bahwa isi dari field cust_id dan date secara otomatis dihasilkan oleh sistem. Klausula IDENTITY secara otomatis menaikkan isi dari field cust_id dan fungsi GETDATE() menghasilkan tanggal dan jam dari sistem.

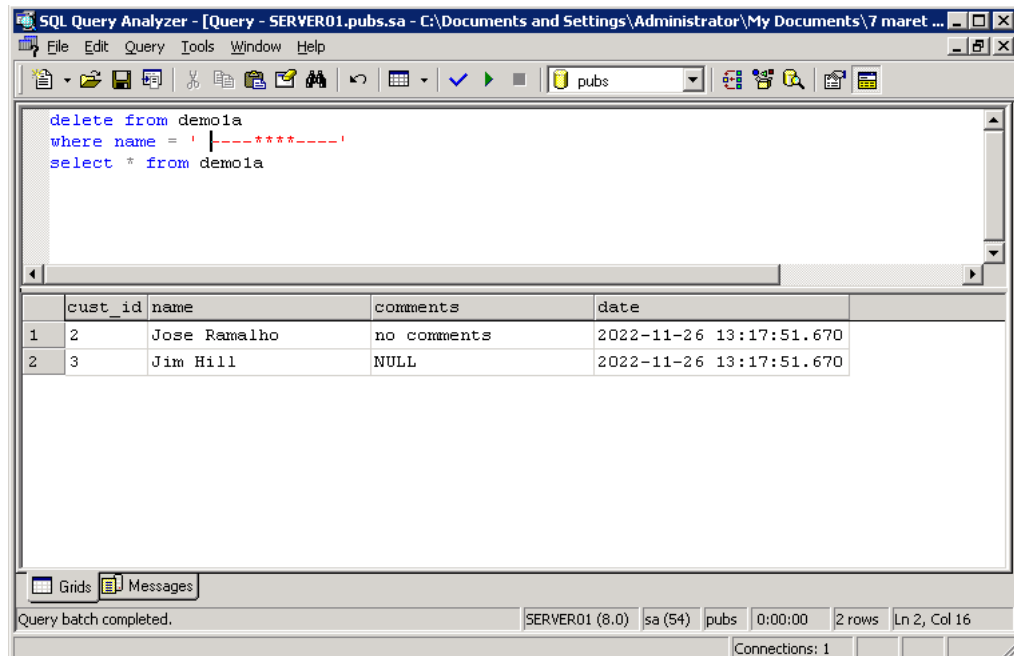


Gambar 6.13 Contoh IDENTITY

Perintah DELETE

Perintah ini menghapus baris dari tabel atau tampilan yang memenuhi kondisi tertentu. Sintaks dasar: DELETE FROM scheme.table WHERE condition. Sebagai contoh, untuk menghapus baris-baris dari tabel demo1a yang baru saja kita buat dengan nama field "----***-----" kita bisa menggunakan perintah berikut ini:

```
delete from demo1a where name="----****----"
select * from demo1a
```



Gambar 6.14 Contoh Perintah DELETE

Perintah UPDATE

Perintah UPDATE memperbarui atau me-refresh satu atau beberapa bidang dari satu baris atau grup baris dalam tabel atau tampilan. Isi setiap bidang dapat diatur menggunakan pernyataan SET. Jika lebih dari satu kolom diperbarui, pasangan kolom=nilai harus dipisahkan dengan koma. Bidang identitas tidak dapat diperbarui.

Sintaks dasar:

```
UPDATE .table SET column=value WHERE condition
```

```
UPDATE {<table_or_view>} SET {column_name = {expression | DEFAULT}| @ variable
= expression} [,...n] [FROM {<table_or_view> | select_statement} [AS] table_alias [
(column_alias [,...m]) ] | <table_or_view> CROSS JOIN <table_or_view> | INNER
[<join_hints>] JOIN <table_or_view> ON <join_condition> | <rowset_function>}, ...n]]
[WHERE <search_conditions> | CURRENT OF {{{[GLOBAL] cursor_name } |
cursor_variable_name}}] [OPTION (<query_hints>, [,...n])] SET ,menentukan daftar kolom
yang akan di-update.
```

column_name adalah nama kolom yang akan di-update. Jika ia merujuk kepada tabel selain

tabel yang sedang aktif , parameter ini harus diawali dengan nama dari tabel/database.

expression adalah isi baru dari kolom; ia bisa merupakan hasil dari subselect yang mengembalikan hanya satu nilai.

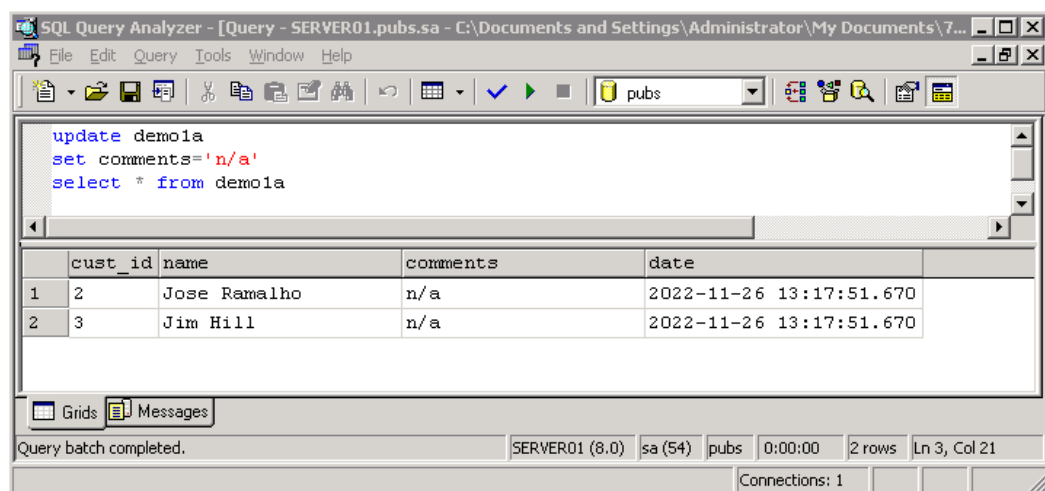
DEFAULT menunjukkan isi saat ini yang harus digantikan dengan nilai default yang sebelumnya telah didefinisikan sewaktu tabel dibuat.

FROM menentukan tabel atau view lain yang akan digunakan untuk menyediakan kriteria update untuk operasi ini.

WHERE menentukan kondisi yang harus dipenuhi untuk membatasi jumlah baris yang akan di-update .

Perhatikanlah contoh di bawah ini di mana isi dari field Comments diubah menjadi "n/a" untuk semua baris.

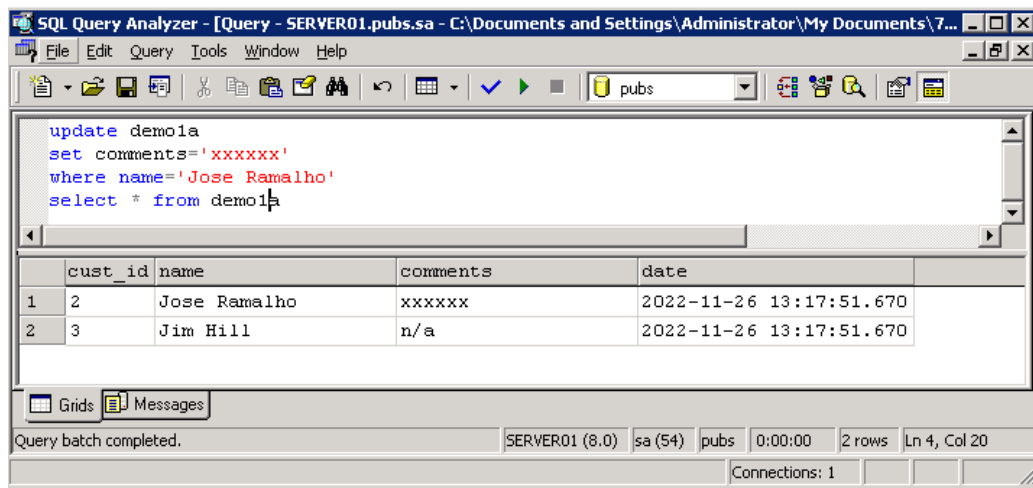
```
update demola
set comments='n/a'
select * from demola
```



Gambar 6.15 Contoh Perintah UPDATE

Pada contoh berikutnya, kita menggunakan klausa WHERE untuk menyaring perubahan.

```
update demola
set comments='xxxxxx'
where name='Jose Ramalho'
select * from demola
```



Gambar 6.16 Contoh Perintah UPDATE

Perintah DROP TABLE

Untuk menghapus tabel dari database, gunakan perintah DROP TABLE. Perintah ini secara fisik menghapus tabel dari database dan menghapus struktur dan datanya.

Sintaks: DROP TABLE nama_tabel

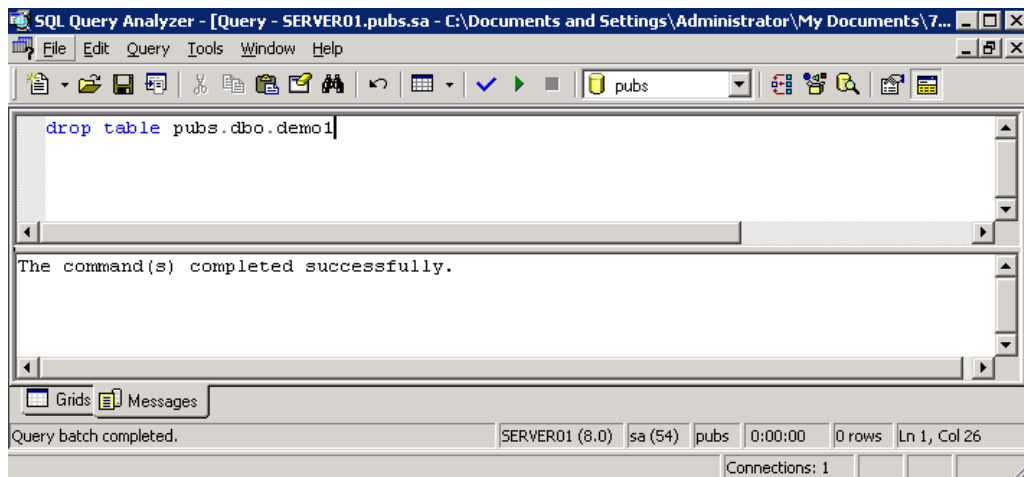
Contoh berikut ini menghapus tabel demo2 dari database yang sedang aktif:

```
DROP TABLE demo2
```

Untuk menghapus tabel di database lain, Anda harus menentukan nama database diikuti dengan ekstensi .dbo dan nama tabel yang akan dihapus.

Sebagai contoh: DROP TABLE pubs.dbo.demo1

Gambar berikut ini menunjukkan bahwa data base master sedang aktif dan kita ingin menghapus tabel demo1 dari database Pubs .



Gambar 6.17 Contoh Perintah DROP TABLE

Ada sebuah tabel Perintah IF EXISTS

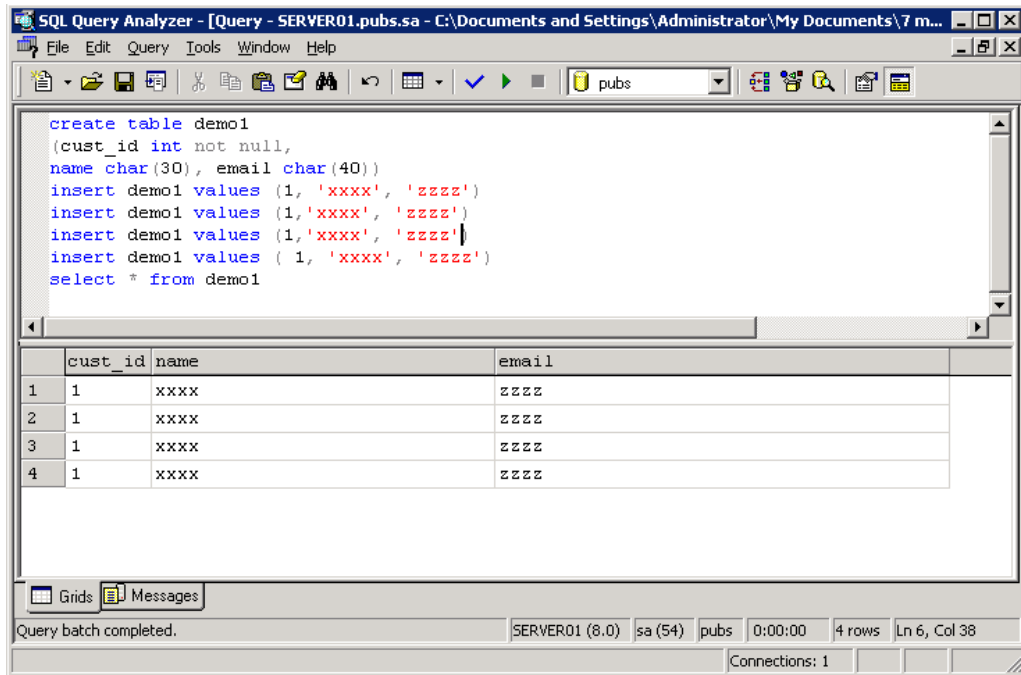
Sistem bernama INFORMATION_SCHEMA.TABLES WHERE, dan salah satu dari kolomnya (table_name) mengandung nama dari tabel-tabel yang ada. Melalui perintah IF EXISTS, kita bisa memeriksa apakah tabel tertentu memang ada, sehingga kita bisa menghapusnya sebelum membuatnya. Ini adalah contoh dari perintah ini:

```
IF EXISTS (SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME = 'demo1 ') DROP TABLE demo1
```

Kini kita akan membuat sebuah tabel menyisipkan beberapa record ke dalamnya melalui Query Analyzer.

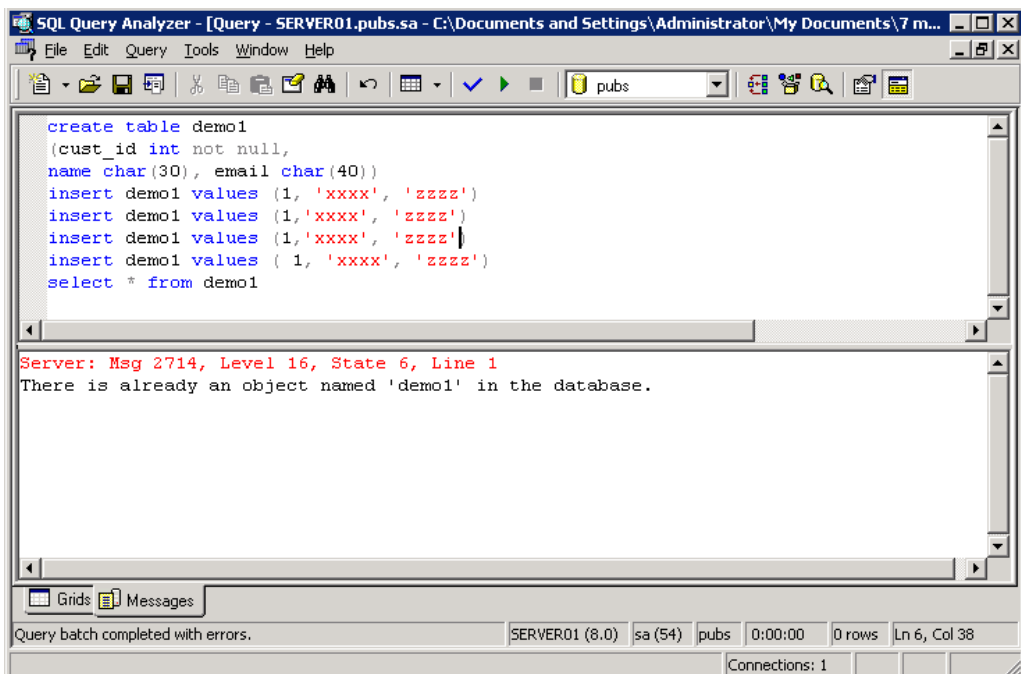
```
create table demo1
(cust_id int not null,
name char(30), email char(40))
insert demo1 values (1, 'xxxx', 'zzzz')
insert demo1 values (1,'xxxx', 'zzzz')
insert demo1 values (1,'xxxx', 'zzzz')
insert demo1 values ( 1, 'xxxx', 'zzzz')
select * from demo1
```

Hasil dari perintah ini adalah :



Gambar 6.18 Contoh Perintah CREATE TABLE

Tetapi jika Anda mencoba menjalankannya kembali, Anda akan mendapatkan pesan error berikut ini:



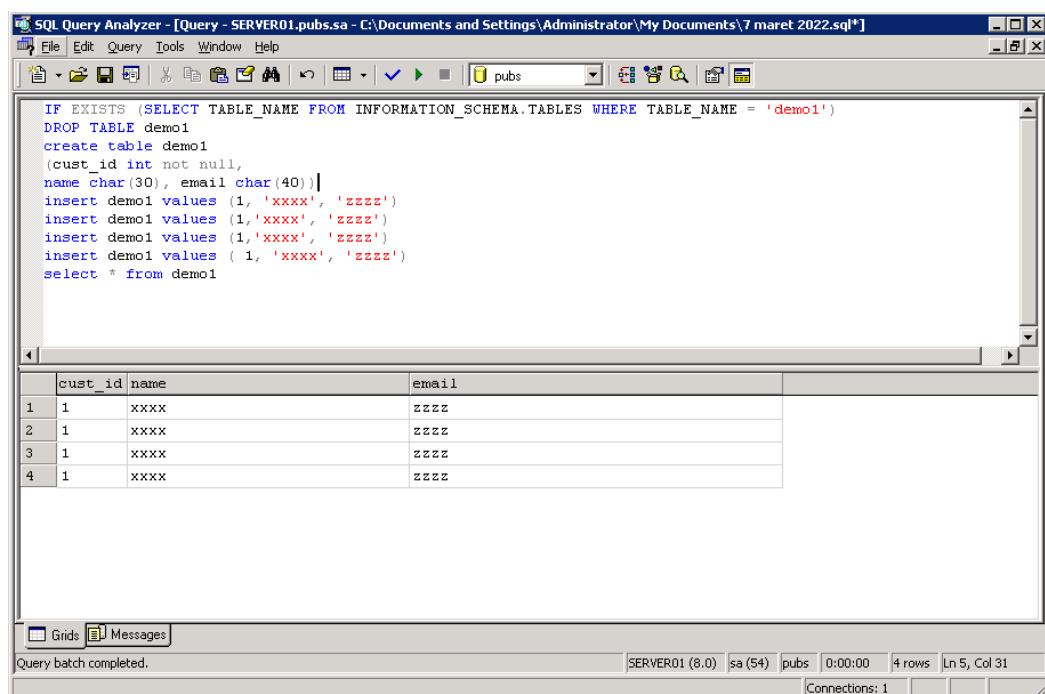
Gambar 6.19 Contoh Perintah ERROR CREATE TABLE

Dengan mengubah skrip dan menyertakan perintah IF, kita bisa menjalankannya tanpa masalah. Tabel yang sudah ada akan selalu dihapus.

```

IF EXISTS (SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME = 'demo1')
DROP TABLE demo1
create table demo1
(cust_id int not null,
name char(30), email char(40))
insert demo1 values (1, 'xxxx', 'zzzz')
insert demo1 values (1,'xxxx', 'zzzz')
insert demo1 values (1,'xxxx', 'zzzz')
insert demo1 values ( 1, 'xxxx', 'zzzz')
select * from demo1

```



Gambar 6.20 Contoh Perintah IF EXISTS CREATE TABLE

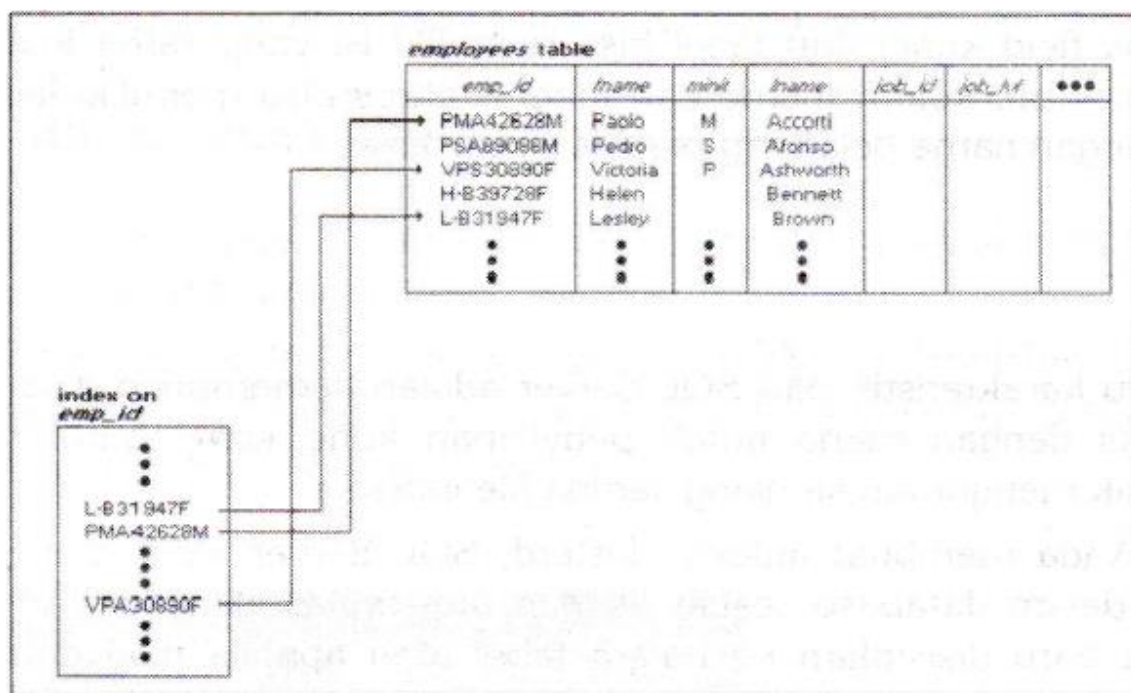
Kesimpulan

Bab ini memberikan pengenalan singkat kepada bahasa SQL. Dengan informasi yang terdapat di sini, Anda bisa melakukan query, menjelajahi, dan bahkan membuat tabel pada banyak data base. Pada beberapa bab berikutnya, kita akan melihat bagaimana Enterprise Manager dapat mengahatkan pekerjaan manual dalam membuat dan mengelola tabel seperti yang Anda lihat di sini. Bab berikutnya membahas mengenai cara bekerja dengan indeks.

BAB 7

Indeks

Di Bab 1, Anda telah mempelajari bahwa indeks adalah file tambahan yang terkait dengan tabel yang tugasnya mempercepat akses ke baris dalam tabel. Indeks terdiri dari kunci berdasarkan isi dari satu atau lebih kolom tabel. Gambar 7.1 di bawah menunjukkan tabel Employees yang mengandung indeks berdasarkan kepada kolom emp_id.



Gambar 7.1 Indeks

Jenis-Jenis Indeks

SQL Server menggunakan dua jenis indeks: berkelompok dan tidak berkelompok

1. **Clustered:** Indeks berkerumun mengatur baris tabel dalam urutan tertentu, seperti menurut abjad atau numerik. Misalnya, dalam urutan abjad, catatan untuk penulis bernama Bennet berada di atas catatan penulis bernama Yokomoto. Jenis indeks ini sempurna bila tabel tidak sering diperbarui karena bila konten bidang kunci berubah, tabel disusun ulang. Satu tabel hanya dapat memiliki satu indeks berkerumun. Kita bahkan bisa mengatakan bahwa baris dari tabel itu adalah indeks, seperti dalam kasus buku telepon di mana baris dengan nama berfungsi sebagai nama data dan indeksnya sekaligus. Dengan kata lain,

untuk mencari nomor telepon Jose Ramalho, Anda harus mencari entri Ramalho, lalu Jose di dalam daftar, lalu mendapatkan nomor teleponnya yang berada di sebelah nama tersebut.

2. **Non-clustered:** Indeks nonclustered memiliki struktur yang berbeda dari array. Urutan fisik baris dalam tabel tidak mengikuti urutan folder. Indeks non-cluster mirip dengan indeks akhir buku. Data (baris tabel) ditempatkan di satu tempat dan indeks di tempat lain. Informasi dalam file indeks disimpan dalam urutan berbeda yang tidak mengikuti urutan file indeks berkerumun. Setiap tabel dapat memiliki beberapa indeks jenis ini.

Indeks juga dapat didefinisikan sebagai unik, sehingga tidak boleh memiliki lebih dari satu baris dengan konten yang sama. Jika indeks tidak dibuat dengan opsi ini, bidang kunci tabel dapat memiliki konten yang sama berkali-kali. Sebagai contoh, kolom lname pada Authors Table dapat berisi lebih dari satu record dengan nama belakang dari penulis yang sama.

Fill Factor

Salah satu karakteristik dari SQL Server adalah kemampuan dalam membuat file indeks dengan ruang untuk penyisipan kunci-kunci baru, tanpa harus secara fisik mengorganisir ulang semua file indeks.

Apabila Anda membuat indeks clusterd, SQL Server akan menyimpan data tabel ke dalam database sesuai dengan urutan-urutan dari field kunci. Apabila ada baris baru disisipkan ke dalam tabel atau apabila nilai dari field kunci berubah, Anda perlu mengorganisir ulang penyimpanan dari baris-baris ini pada halaman database. Apabila halaman sudah penuh, server memindahkan sekitar separuh dari baris dari halaman tersebut ke halaman yang baru. Ini disebut dengan istilah page split, dan bergantung dari ukuran tabel, ia bisa sangat mempengaruhi kinerja terutama apabila ada operasi update yang intensif pada field kunci. Selama proses pembuatan indeks, apakah dengan menggunakan Enterprise Manager atau perintah Transact-SQL, Anda bisa menggunakan pilihan Fill Factor untuk memberikan ruang kosong di dalam halaman. Jadi penambahan atau perubahan yang dilakukan pada baris tabel di kemudian hari bisa di masukkan ke dalam halaman, tanpa harus mengorganisir ulang. Fill Factor memiliki nilai antara 1 sampai 100, dan menentukan berapa besar ruang kosong yang harus berada di dalam halaman. Nilai 100 menunjukkan bahwa halaman bisa diisi dengan penuh. Nilai ini hanya bisa digunakan apabila tabel digunakan untuk query. Nilai yang lebih kecil dari 100 akan menyisakan lebih banyak ruang yang tidak terpakai di dalam halaman untuk penambahan di kemudian hari.

Membuat Indeks

Indeks dibuat dalam filegroup yang sama dengan tabel. Alternatifnya, Anda dapat membuat indeks yang tidak terkunci pada grup file yang berbeda untuk meningkatkan kinerja dengan menggunakan media akses fisik yang berbeda.

Ada dua cara di dalam membuat indeks. Salah satunya adalah menggunakan perintah Transact-SQL yang disebut CREATE INDEX, yang lain adalah dengan menggunakan Enterprise Manager. Indeks bisa dibuat baik selama pembuatan tabel atau pada tabel yang sudah ada.

Membuat Indeks pada Saat Pembuatan Tabel

Luaskan folder Databases dan folder database tempat Anda ingin membuat tabel/indeks. Klik kanan folder Tabel, lalu pilih Tabel Baru. Di kotak dialog Pilih Nama yang muncul, masukkan nama untuk tabel ini. Untuk contoh ini, ketikkanlah test1.



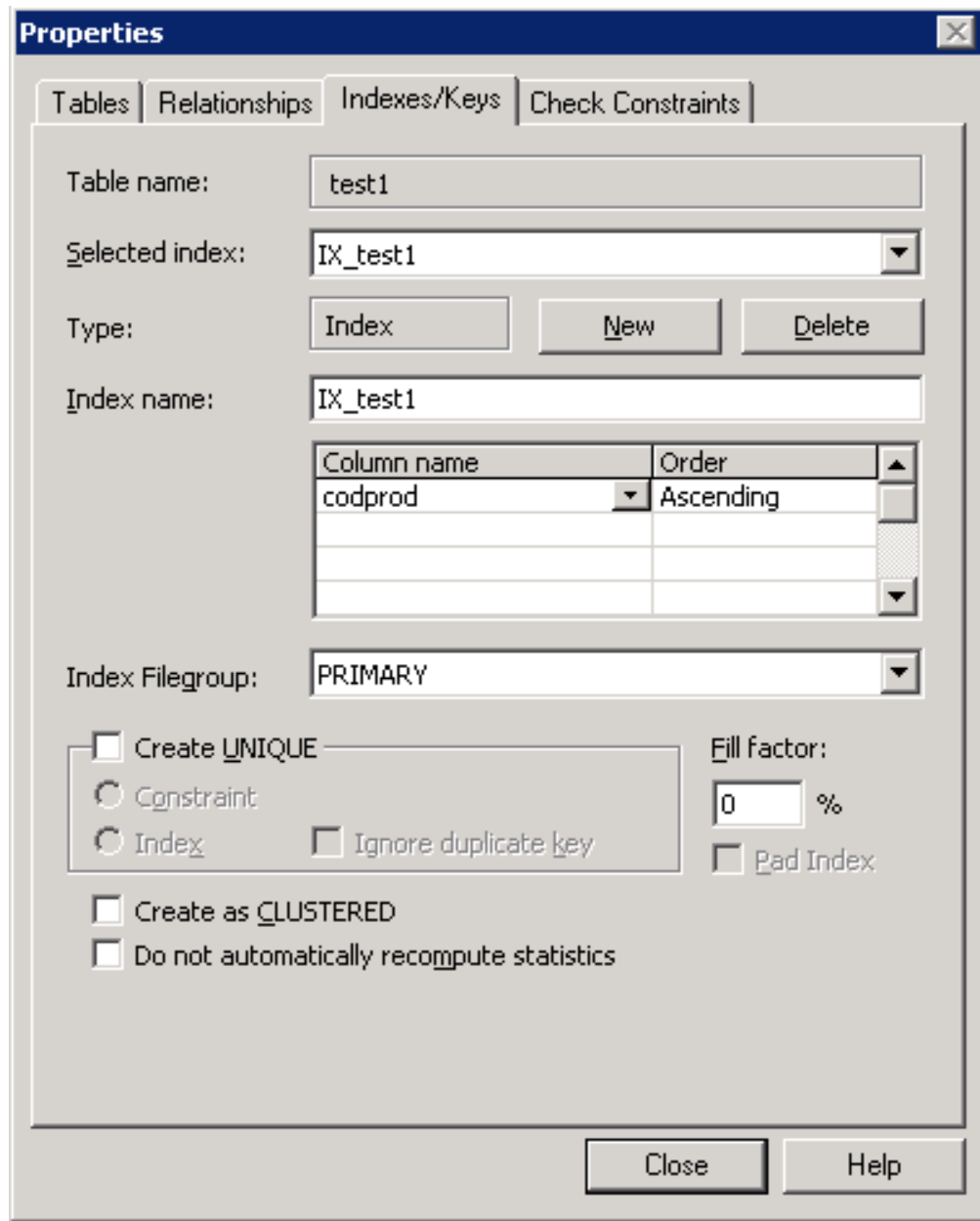
Gambar 7.2 Membuat Indeks

Isilah field-field yang sesuai dengan kolom tabel. Sebagai contoh, kita akan membuat tiga buah kolom dengan menggunakan nilai-nilai pada gambar di bawah ini. Perhatikan bahwa kolom Allow Nulls pada baris pertama tidak dipilih .

	Column Name	Data Type	Length	Allow Nulls
	codprod	char	5	
▶	nameprod	char	30	✓
	cost	int	4	✓

Gambar 7.3 Membuat Indeks

Klik kanan kolom mana saja dan pilih Properties. Pada tab Indexes/Keys dari kotak dialog Properties, klik tombol New. Di bidang Nama kolom, pilih kolom (atau beberapa kolom) yang akan membentuk indeks. Dalam contoh ini kami memilih codprod.



Gambar 7.4 Membuat Indeks lewat Properties

Nama indeks dapat diubah. Masukkan nama lain di bidang Nama Indeks. Di bidang File Filegroup, Anda dapat memilih filegroup yang berbeda untuk membuat folder. Untuk membuat indeks dengan tipe unik, pilih bidang Create UNIQUE, lalu pilih apakah bidang dibatasi oleh indeks atau oleh batasan tertentu. Untuk membuat indeks tipe tergugus, pilih kotak centang Create CLUSTERED, lalu pilih salah satu opsi.

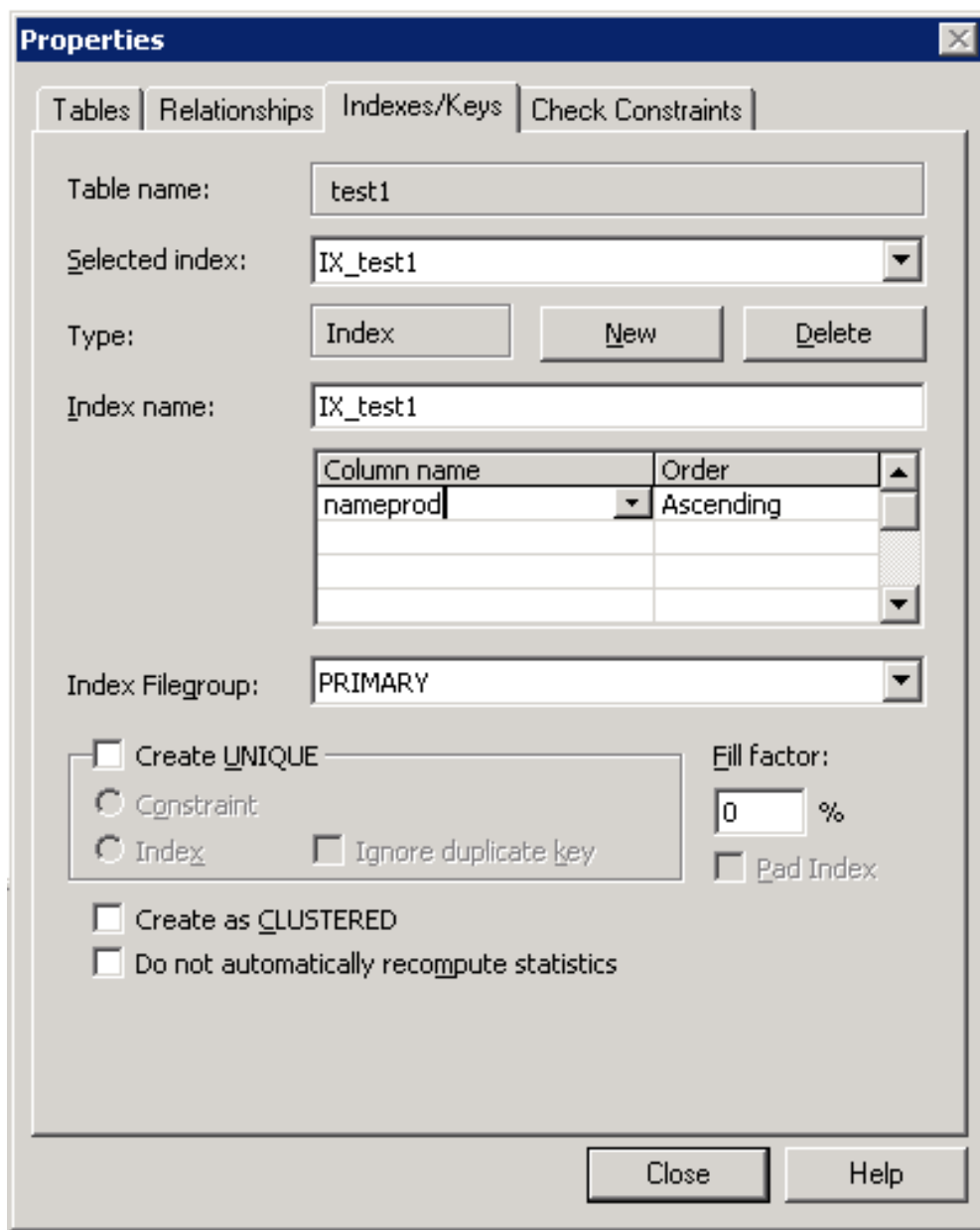
Tutup kotak dialog untuk menyelesaikan pembuatan indeks. Perhatikan bahwa pada tabel ini, kita hanya membuat sebuah indeks berdasarkan kepada salah satu dari fieldnya, dan kita tidak membuat kunci primer.

Membuat Indeks di Tabel Ada

Kami sekarang akan membuat indeks baru di tabel test1 yang baru saja kami buat. Untuk membuat indeks baru pada tabel yang sudah ada, perluas database lalu folder Tabel.

Klik kanan nama tabel dan pilih Design Table. Klik kanan di kolom mana saja dan pilih Properties. Kotak dialog Properti muncul.

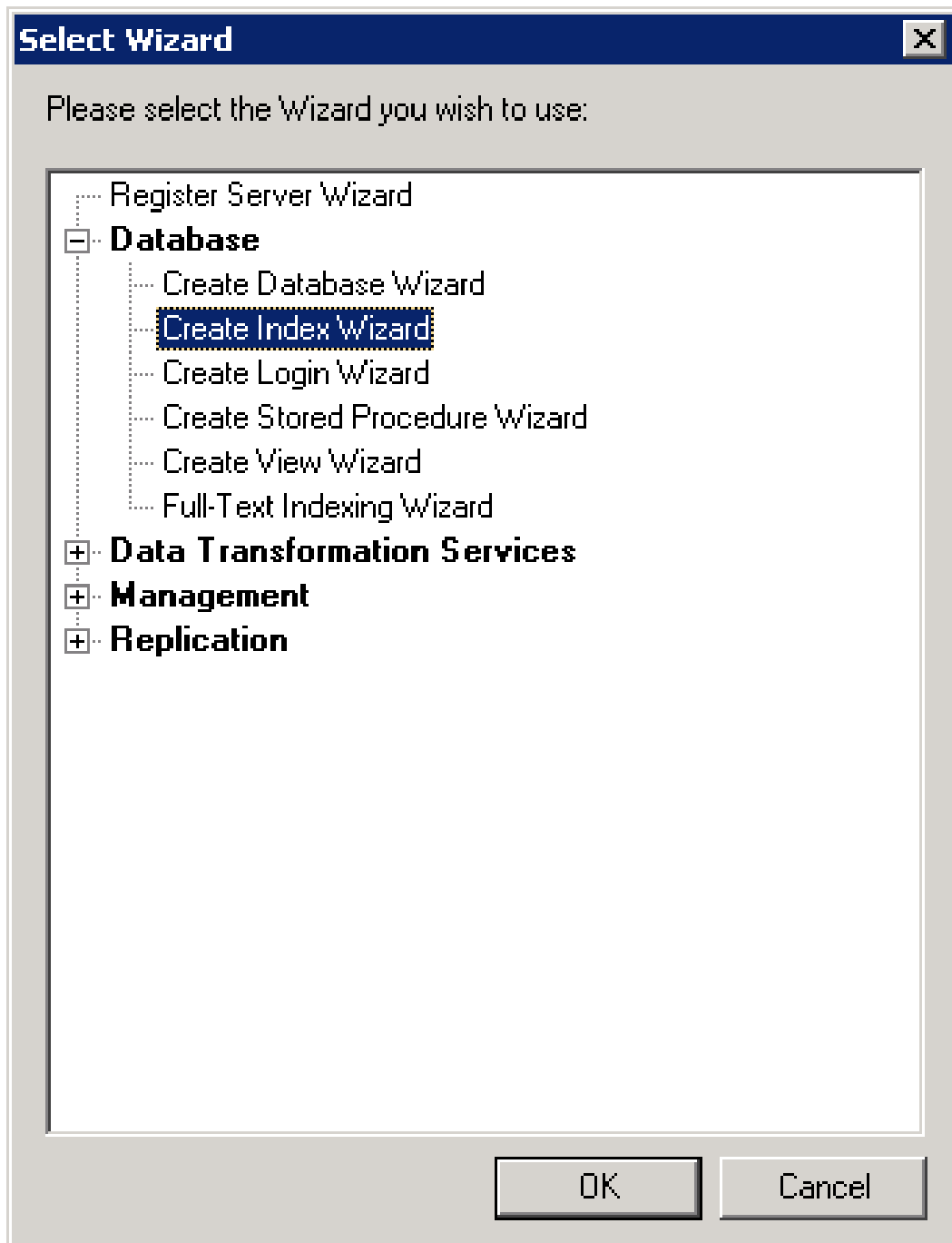
Pada tab Indeks/Kunci, klik Baru. Di bidang Nama kolom, pilih kolom atau grup kolom untuk menjadi bagian dari indeks. Pada contoh ini, kita akan membuat sebuah indeks bernama name (ketikkan ini pada field Index Name), berdasarkan pada kolom Nameprod, lalu pilihlah Create as CLUSTERED.



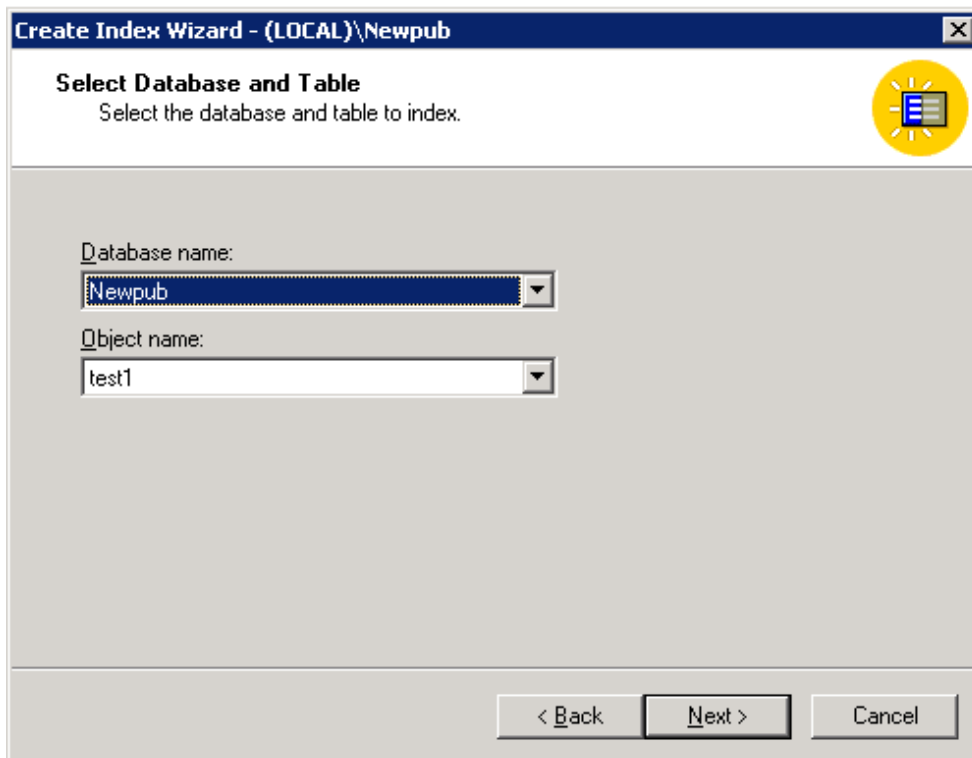
Gambar 7.5 Membuat Indeks di table yang sudah ada

Membuat Indeks menggunakan Wizard

Untuk membuat folder menggunakan wizard, klik kanan folder database dan pilih Tools. Perluas database, lalu klik dua kali Wisaya Buat Direktori. Ikuti instruksi wizard seperti yang dijelaskan di bawah ini. Di layar pertama, pilih database dan tabel untuk membuat indeks. Tekan Berikutnya.

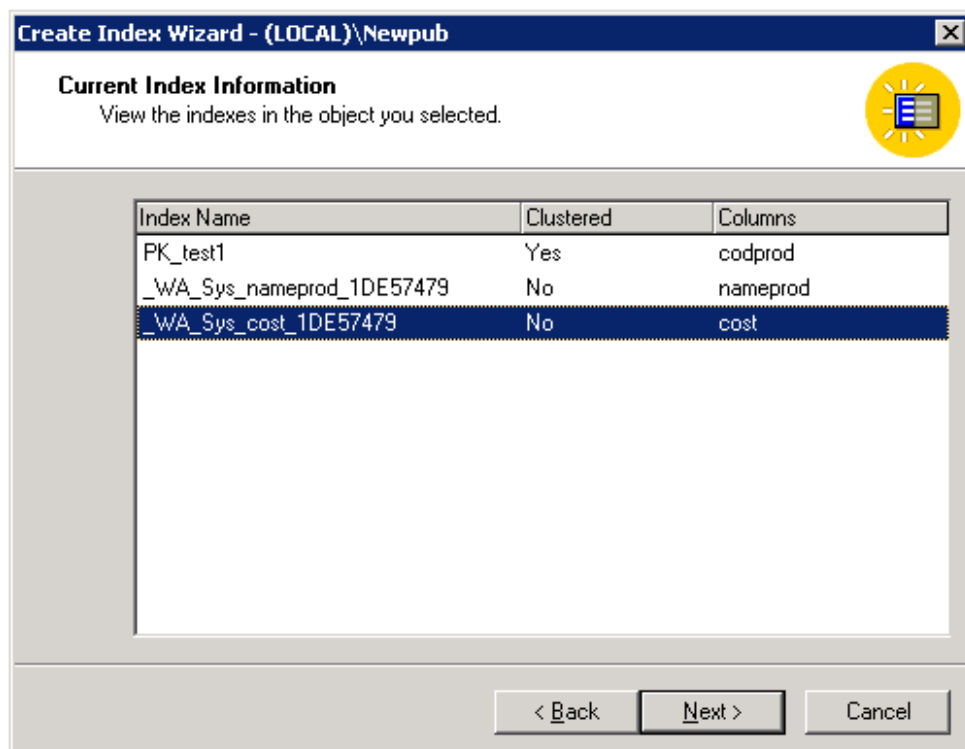


Gambar 7.6 Membuat Indeks Wizard



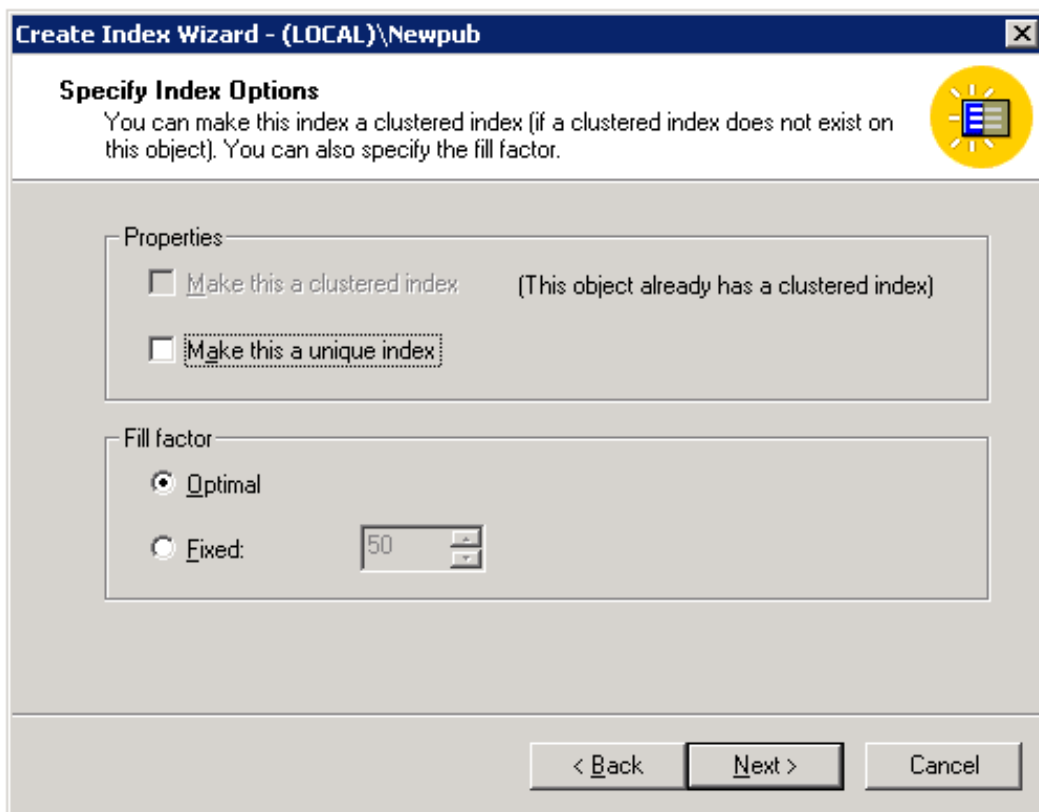
Gambar 7.7 Membuat Indeks Wizard-Database name

Pada langkah selanjutnya, wizard akan menampilkan indeks yang ada. Tekan Next. Di kotak dialog berikutnya, pilih kolom yang membentuk indeks.



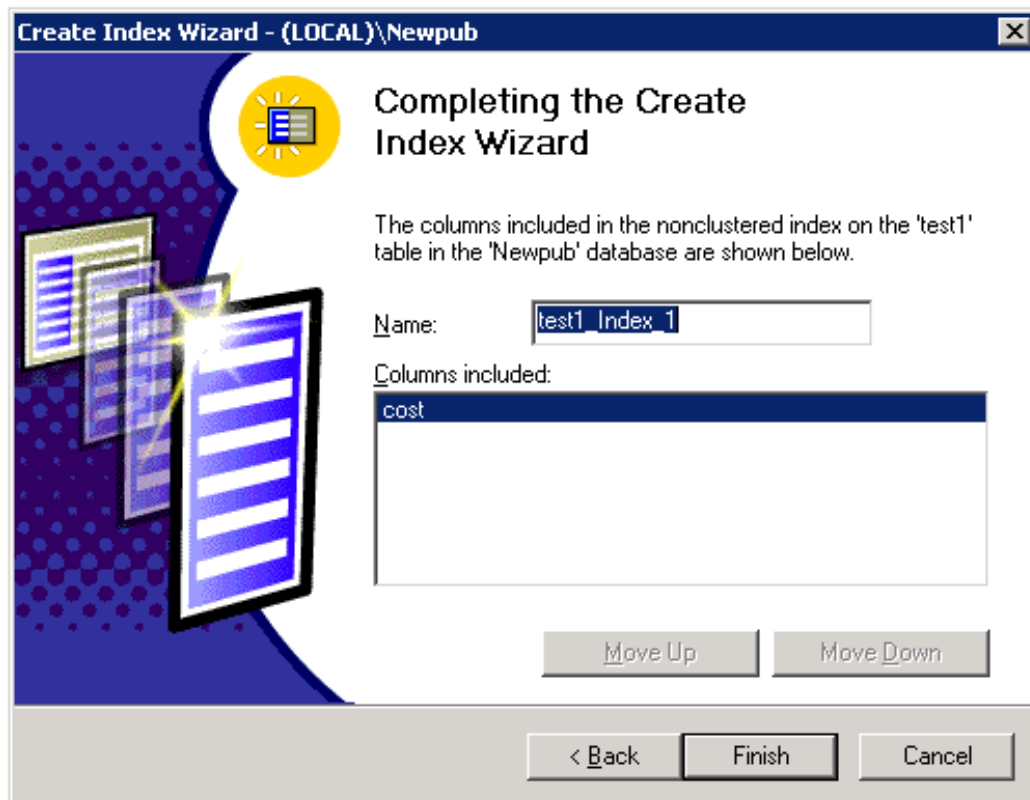
Gambar 7.7 Membuat Indeks Wizard-cost

Pada contoh ini, pilihlah kolom Cost lalu tekan Next. Pada titik ini Anda dapat menentukan nilai tetap untuk pengaturan faktor isian atau mengubah jenis indeks menjadi unik. Anda juga dapat menentukan pembuatan indeks berkerumun. Karena kita sudah membuat indeks bertipe clustered untuk tabel ini, pilihan untuk itu akan dinonaktifkan. Tekan Next.



Gambar 7.8 Specity Index Options

Pada langkah terakhir Anda dapat mengubah nama indeks dan menampilkan kolom yang merupakan bagian dari indeks. Jika ada lebih dari satu kolom, Anda dapat mengubah urutan kolom menggunakan tombol Atas dan Bawah. Klik tombol Selesai untuk membuat folder.



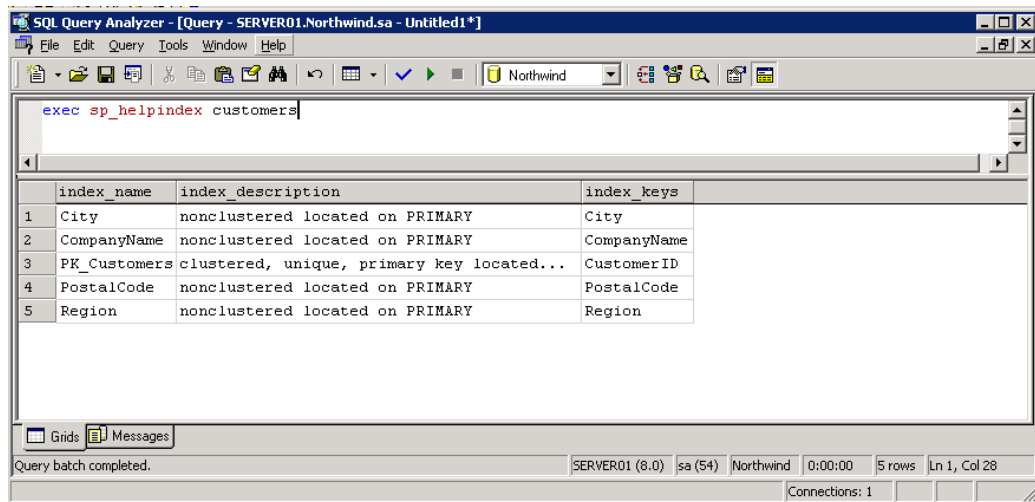
Gambar 7.9 Completing Create Index Wizard

Merekonstruksi Indeks

Indeks secara fisik ditulis ke halaman indeks dalam database. Halaman direktori dibangun secara berurutan dengan penunjuk yang menunjukkan lokasi di halaman berikutnya. Saat bidang kunci dan indeks berubah, halaman indeks menjadi terfragmentasi. Kami menyarankan Anda membangun kembali indeks menggunakan perintah CREATE INDEX dan WITH DROP_EXISTING, yang menggunakan catatan yang ada untuk mempercepat proses.

Melihat Indeks

Untuk melihat indeks yang sudah ada di tabel, gunakan prosedur sp_helpindex. Prosedur ini melakukan query di dalam tabel sistem sysindexes, yang mengandung informasi mengenai semua indeks. Prosedur ini harus dijalankan di dalam Query Analyzer. Sebagai contoh, untuk menjalankan prosedur ini untuk tabel Customers pada database Northwind, bukalah Query Analyzer, pilih database Northwind, lalu jalankan perintah exec sp_helpindex customers, seperti pada gambar 7.10.

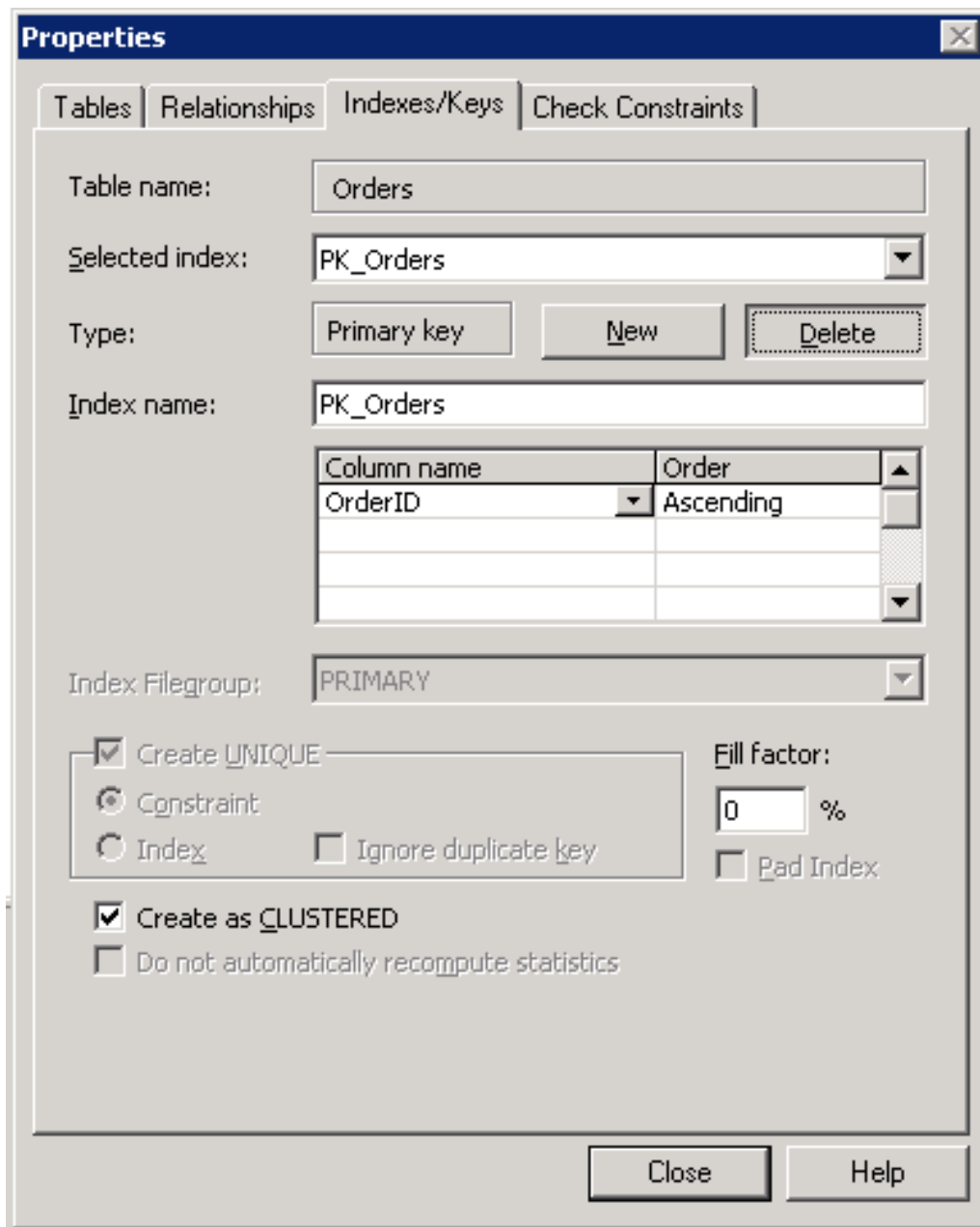


Gambar 7.10 Melihat Indeks

Menghapus Indeks

Setelah membuat folder, Anda dapat menghapusnya saat tidak digunakan lagi. Indeks sementara atau yang tidak diperlukan dapat dihapus menggunakan langkah-langkah berikut:

Perluas database dan folder Tabel. Di panel informasi, klik kanan tombol yang ingin Anda hapus indeksinya, lalu pilih Design Table dari menu konteks yang muncul. Klik kanan kolom mana pun, lalu pilih Properti. Kotak dialog Properti muncul. Pada tab Indeks/Kunci, dalam kotak Indeks Terpilih, pilih indeks yang ingin Anda hapus, lalu klik tombol Hapus. Layar di bawah ini menampilkan operasi di atas yang telah dilakukan pada tabel Orders pada database Northwind.



Gambar 7.11 Menghapus Indeks

Jika kita menekan tombol Delete, indeks ShippedDate akan dihapus. Untuk membatalkan operasi ini, tutuplah kotak dialog.

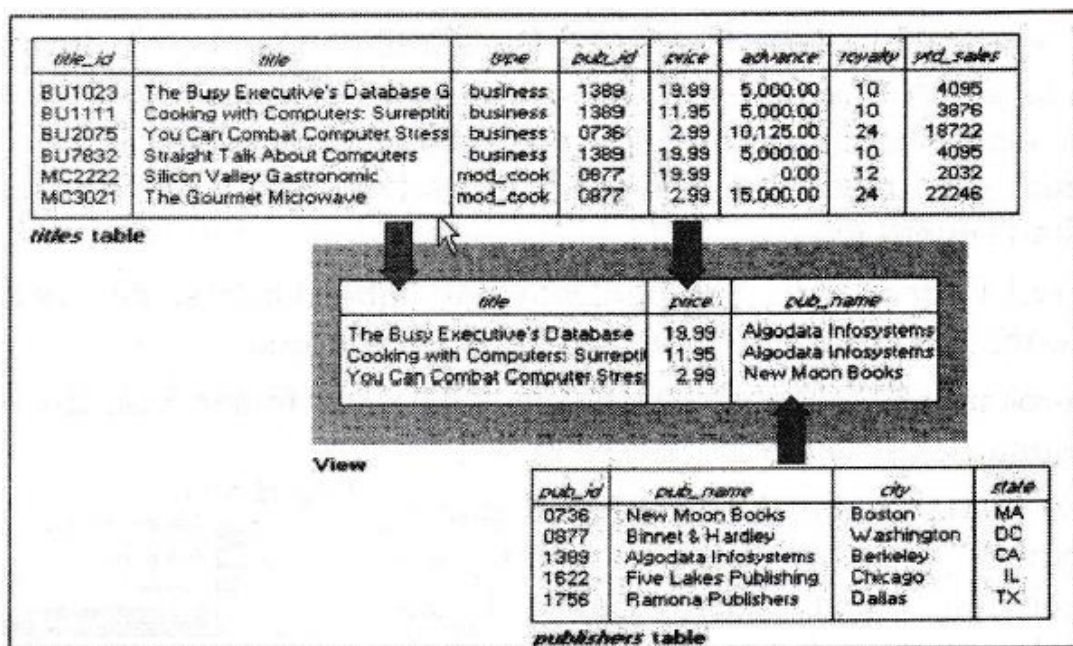
Kesimpulan

Pada bab ini, Anda telah melihat jenis-jenis indeks yang dikelola oleh SQL Server. Anda telah melihat cara membuat indeks dengan menggunakan Create Index Wizard dan kotak dialog Properties, serta cara melihat dan menghapusnya. Pada bab berikutnya, kita akan belajar cara membuat view database.

BAB 8

View

Tampilan adalah tabel virtual yang isinya ditentukan oleh kueri basis data. Tampilan bukanlah larik fisik, tetapi serangkaian pernyataan yang menghasilkan sekumpulan data. Gambar di bawah ini menampilkan dua buah tabel yang berfungsi sebagai dasar di dalam pembuatan view.



Gambar 8.1 View

Menggunakan tampilan sangat berguna saat Anda ingin fokus pada data tertentu dalam database. Bayangkan database perusahaan yang digunakan oleh banyak pengguna di berbagai departemen. Informasi yang diproses oleh tim penjualan sangat berbeda dengan informasi yang digunakan oleh departemen akuntansi dan pemasaran. Menggunakan tampilan hanya dapat memberikan informasi yang Anda perlukan, apakah itu berasal dari satu atau beberapa tabel database.

Tampilan tersebut memungkinkan banyak pengguna yang berbeda untuk melihat data yang sama pada pembesaran yang berbeda. Tampilan memungkinkan Anda menggabungkan data untuk memenuhi kebutuhan pengguna tertentu dan bahkan dapat diekspor ke aplikasi lain.

Membuat View

Tampilan dapat dibuat dengan perintah Transact-SQL yaitu CREATE VIEW, atau dengan editor tampilan yaitu Enterprise Manager.

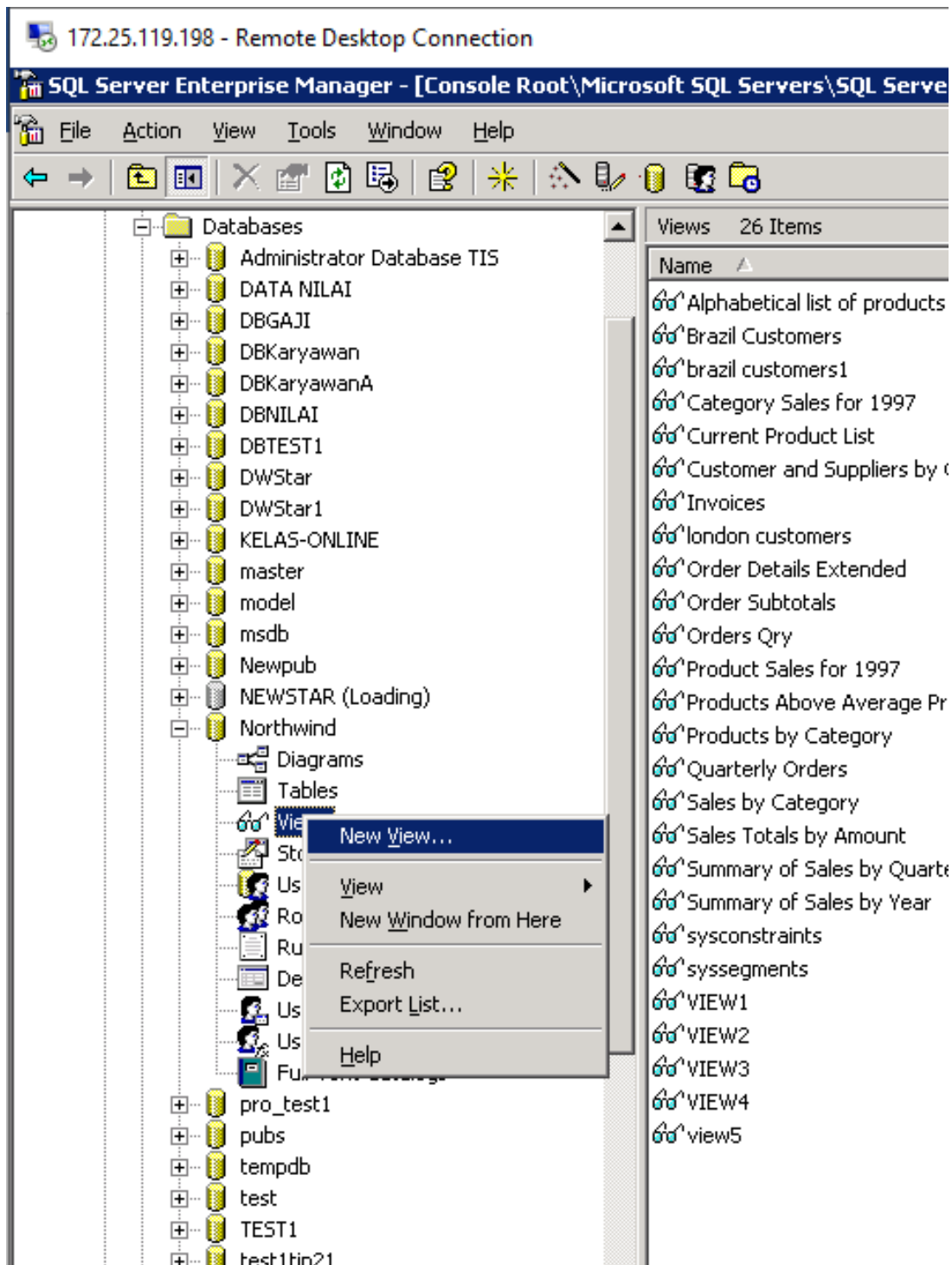
Saat membuat tampilan, perhatikan hal-hal berikut:

- Tampilan hanya dapat dibuat pada database yang digunakan.
- Anda tidak dapat mengaitkan pemicu, aturan, dan nilai default dengan tampilan.
- Tampilan dapat menggunakan data dari tampilan lain.
- Tampilan dapat direferensikan menggunakan prosedur tersimpan. □ Anda tidak dapat membuat indeks untuk tampilan.
- Jika ada beberapa kolom dengan nama yang sama dalam tampilan, kolom tersebut harus berupa alias dan diawali dengan nama tabel tempat kolom tersebut berada.
- Kolom tampilan sesuai dengan kolom tabel asli.

Kita akan membuat view di dalam data base Northwind yang berfungsi menindaklanjuti pelanggan yang pesanannya telah dikirim. View semacam ini akan menggunakan field-field dari tabel Orders dan Customers. Strukturnya akan seperti ini:

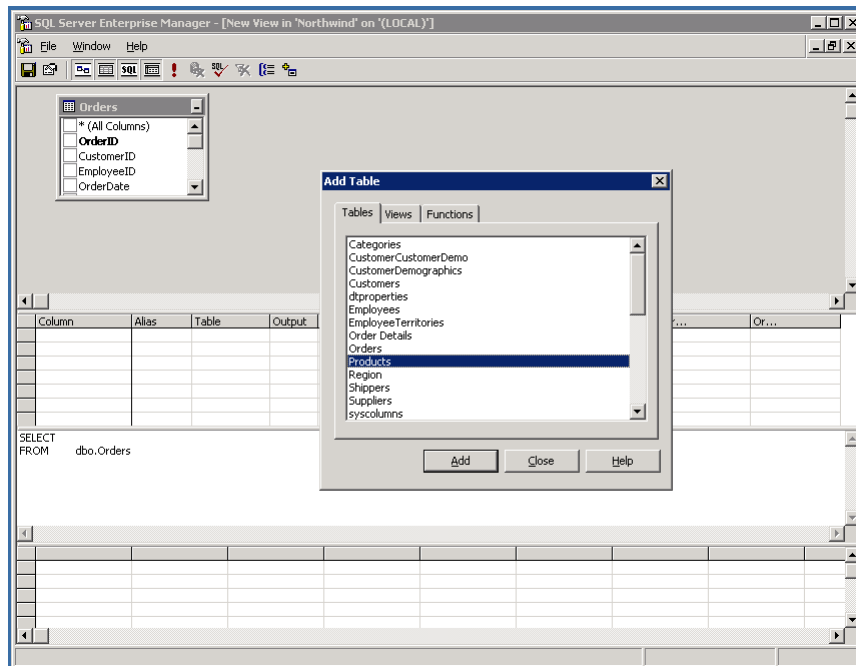
Field OrderId dan ShippedDate dari tabel Orders, dan field Company Name, ContactName, dan Phone dari tabel Customers.

Perbesarlah data base Northwind, klik pada folder SQL Server Views, lalu klik kanan pada New SQL Server View. Klik pada tombol Add Table. Ini adalah tombol terakhir pada toolbar dan memiliki tanda plus. Apabila kotak dialog muncul dan menam pilkan tabel-tabel data base, pilihlah Orders lalu klik pada tombol Add.



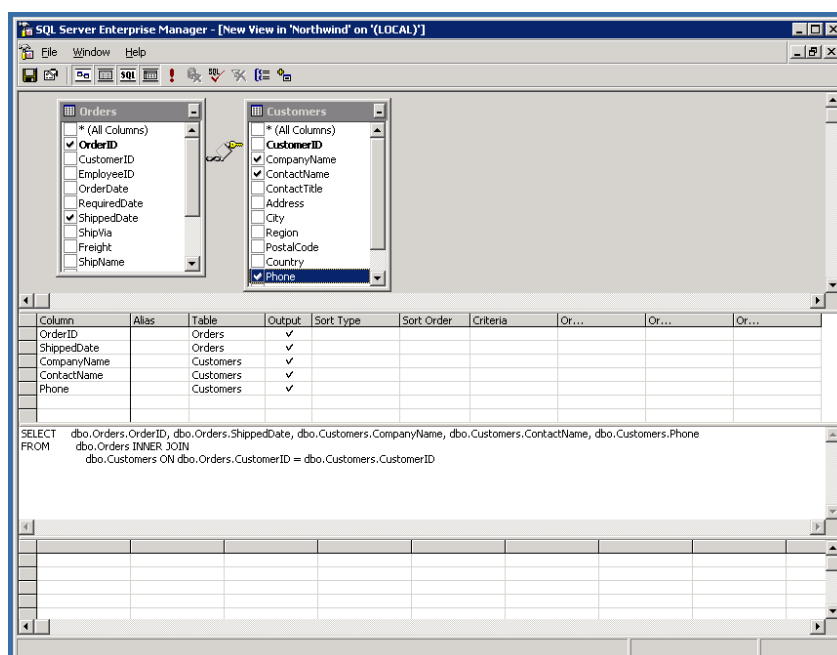
Gambar 8.2 Membuat View

Ulangilah operasi ini dengan tabel Customers. Perhatikan bahwa tabel-tabel ini dihubungkan, karena mereka terhubung berdasarkan field customer_id. Tekanlah tombol Close untuk menutup kotak dialog.



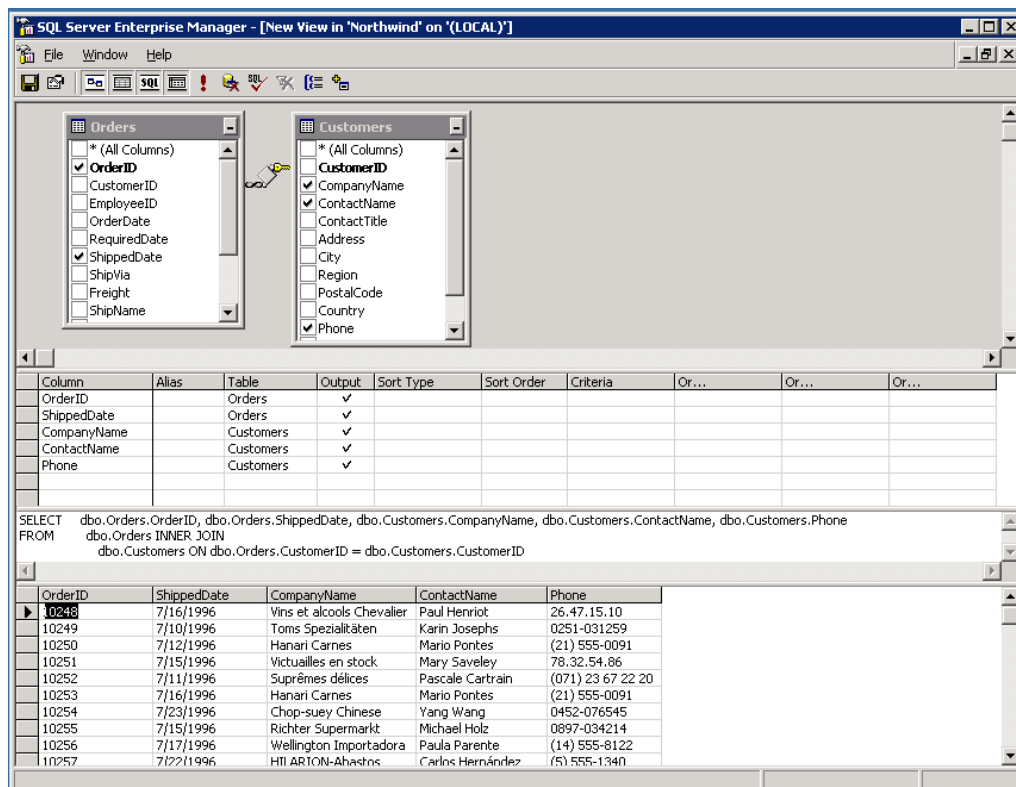
Gambar 8.3 Menambah table View

Langkah berikutnya adalah memilih kolom-kolom yang akan ditampilkan oleh view dari kedua tabel. Klik pada pembatas bawah dari kedua tabel, dan perbesarlah daerah tampilan field. Pilihlah tampilan field. Pilihlah field-field seperti pada gambar 8.4. Urut-urutan pemilihan field sangatlah penting, karena hal ini akan menentukan urutan di mana kolom akan muncul. Pilihlah field-field dalam urutan berikut ini: OrderID, ShippedDate, Company Name, ContactName, Phone.



Gambar 8.4 Menambah 2 table untuk View

Perhatikan bahwa saat setiap kolom dipilih, perintah SELECT yang dibutuhkan untuk eksekusinya akan dituliskan pada panel utama. Kini jalankanlah query tersebut dengan menekan tombol Run (tombol dengan tanda seru pada toolbar). Hasil dari view akan muncul pada panel bawah, seperti pada gambar 8.5.



Gambar 8.5 Menampilkan View

Menyimpan View

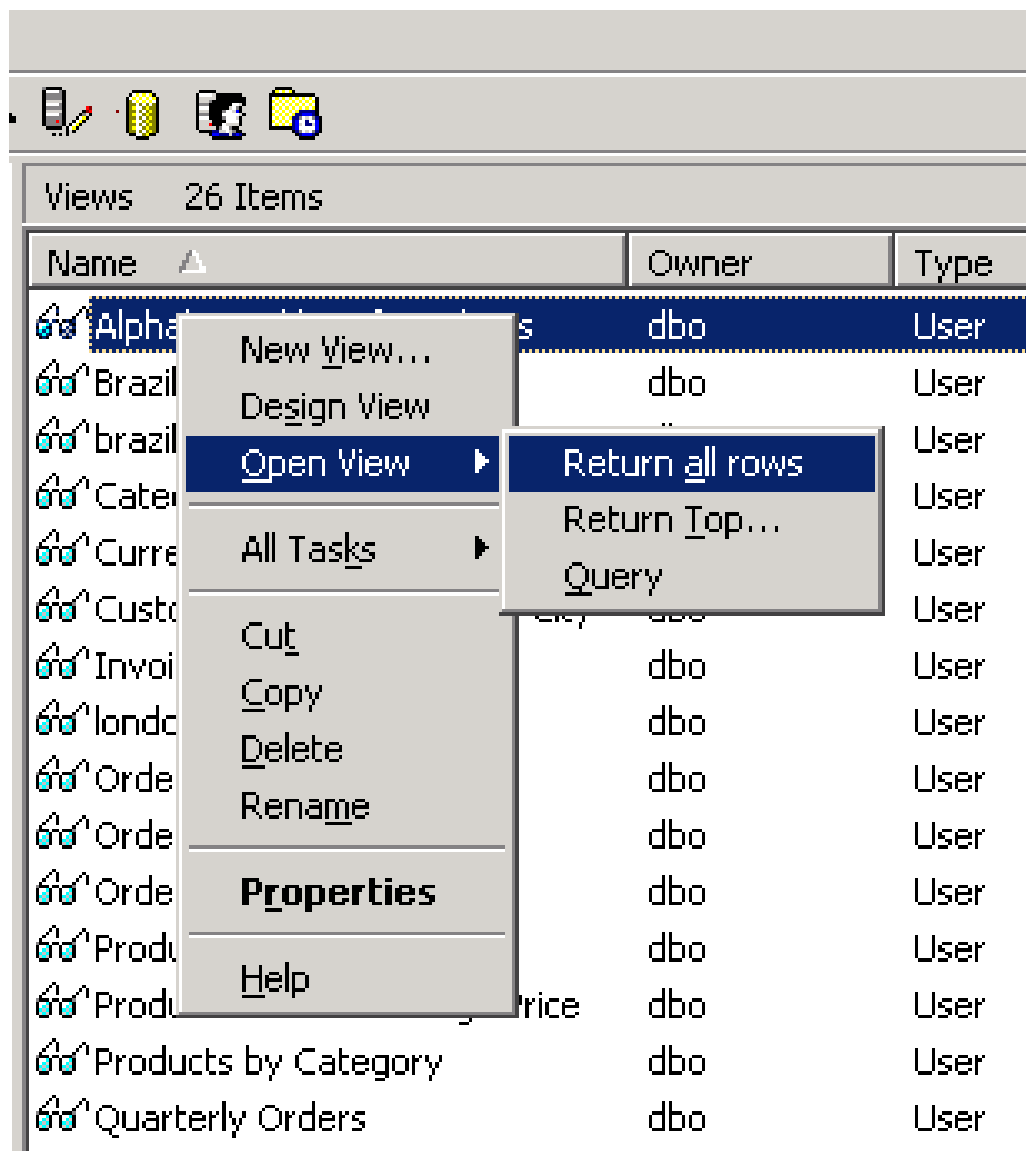
Untuk menyimpan view, tekanlah tombol Save (tombol di ujung paling kiri dari toolbar), ketikkan nama untuk view tersebut (untuk contoh ini, ketikkanlah viewtest1), lalu tekan OK. Tampilan baru ditampilkan bersama dengan tampilan lainnya di area informasi. Tutuplah editor view dengan mengklik tombol Close (X) .

Setelah dibuat, tampilan dapat dilihat, diedit, atau dihapus menggunakan menu konteks, yang dapat diaktifkan dengan mengklik kanan nama tampilan. Fungsi-fungsi ini dijelaskan secara lebih rinci di bagian selanjutnya.

Memeriksa View

Di sisi pengguna, tampilan hanyalah sebuah tabel, seperti tabel lainnya. Oleh karena itu, untuk melihat konten, Anda hanya perlu membuka SQL Server View lalu pilih Return all

rows untuk melihat semua baris, atau pilih Back to top dan tentukan berapa banyak baris yang ingin Anda lihat.



Gambar 8.6 Menjalankan View

Untuk menutup view dan kembali ke panel utama dari Enterprise Manager, klik pada tombol Close di sudut kanan atas.

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel
1	Chai	1	1	10 boxes x 20 bags	18	39	0	10
2	Chang	1	1	24 - 12 oz bottles	19	17	40	25
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13	70	25
4	Chef Anton's Cajun	2	2	48 - 6 oz jars	22	53	0	0
6	Grandma's Boysenb	2	2	12 - 8 oz jars	25	120	0	25
7	Uncle Bob's Organic	3	7	12 - 1 lb pkgs.	30	15	0	10
8	Northwoods Cranb	3	2	12 - 12 oz jars	40	6	0	0
10	Ikura	4	8	12 - 200 ml jars	31	31	0	0
11	Queso Cabrales	5	4	1 kg pkg.	21	22	30	30
12	Queso Manchego L	5	4	10 - 500 g pkgs.	38	86	0	0
13	Konbu	6	8	2 kg box	6	24	0	5
14	Tofu	6	7	40 - 100 g pkgs.	23.25	35	0	0
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39	0	5
16	Pavlova	7	3	32 - 500 g boxes	17.45	29	0	10
18	Carnarvon Tigers	7	8	16 kg pkg.	62.5	42	0	0
19	Teatime Chocolate	8	3	10 boxes x 12 piec	9.2	25	0	5
20	Sir Rodney's Marm	8	3	30 gift boxes	81	40	0	0
21	Sir Rodney's Scone	8	3	24 pkgs. x 4 pieces	10	3	40	5
22	Gustaf's Knäckebrö	9	5	24 - 500 g pkgs.	21	104	0	25
23	Tunnbröd	9	5	12 - 250 g pkgs.	9	61	0	25
25	NuNuCa Nuß-Noug	11	3	20 - 450 g glasses	14	76	0	30
26	Gumbär Gummibärc	11	3	100 - 250 g bags	31.23	15	0	0
27	Schoggi Schokolade	11	3	100 - 100 g pieces	43.9	49	0	30
30	Nord-Ost Matjeshe	13	8	10 - 200 g glasses	25.89	10	0	15
31	Gorgonzola Telino	14	4	12 - 100 g pkgs	12.5	0	70	20
32	Mascarpone Fabioli	14	4	24 - 200 g pkgs.	32	9	40	25
33	Geitost	15	4	500 g	2.5	112	0	20
34	Sasquatch Ale	16	1	24 - 12 oz bottles	14	111	0	15
35	Steeleye Stout	16	1	24 - 12 oz bottles	18	20	0	15
36	Inlagd Sill	17	8	24 - 250 g jars	19	112	0	20
37	Gravad lax	17	8	12 - 500 g pkgs.	26	11	50	25
38	Côte de Blaye	18	1	12 - 75 cl bottles	263.5	17	0	15
39	Chartreuse verte	18	1	750 cc per bottle	18	69	0	5
40	Boston Crab Meat	19	8	24 - 4 oz tins	18.4	123	0	30
41	Jack's New England	19	8	12 - 12 oz cans	9.65	85	0	10
43	Ipoh Coffee	20	1	16 - 500 g tins	46	17	10	25
44	Gula Malacca	20	2	20 - 2 kg bags	19.45	27	0	15
45	Donnerstag	21	8	16 pkgs	9.5	5	70	15

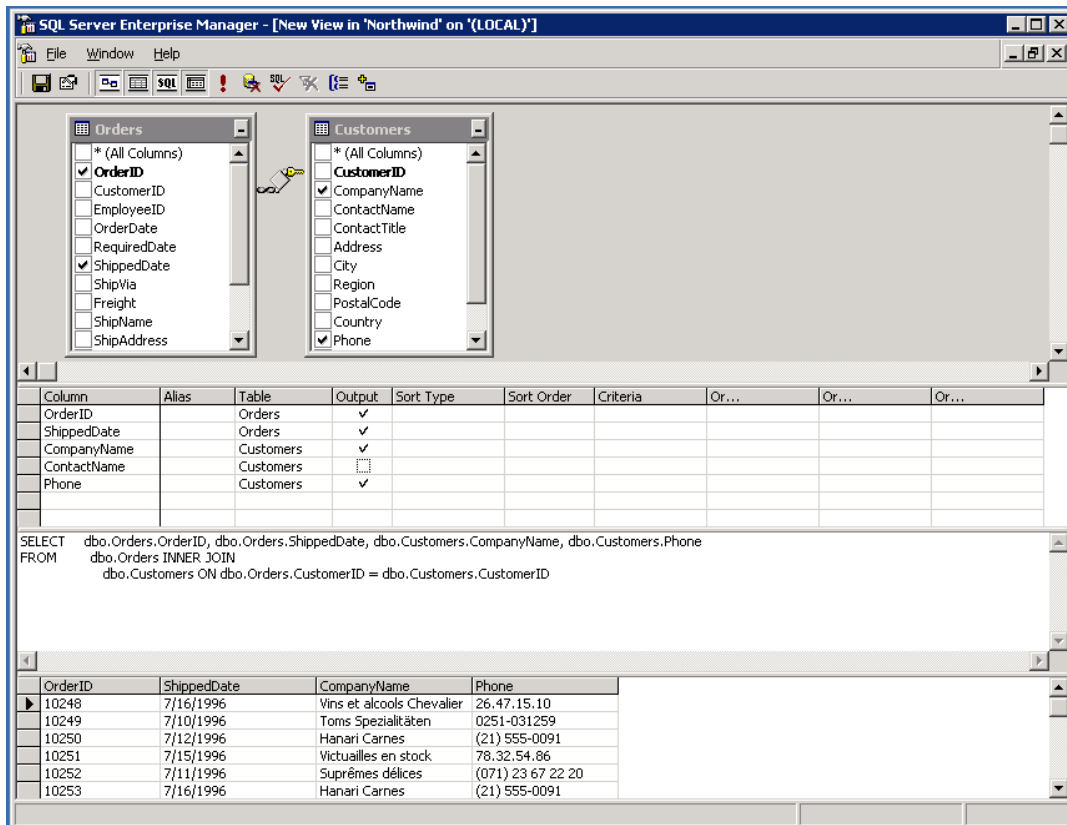
Gambar 8.7 Hasil View

Menyunting View

Perintah tampilan desain SQL Server membuka editor tampilan. Di sana Anda dapat membuat perubahan yang diperlukan. Perubahan ini dapat menghapus kolom secara permanen dan mengubah urutan tampilan kolom.

Mencegah Tampilan Kolom

Untuk mencegah menampilkan beberapa kolom tertentu, pada panel atas Anda harus membatalkan pilihan kolom-kolom yang tidak ingin ditampilkan. Gambar berikut ini menampilkan kolom ContactName yang dibatalkan pilihannya. Untuk melihat view tanpa kolom tersebut, tekanlah tombol Run. Panel pada bagian bawah menampilkan hasilnya.



Gambar 8.8 Memilih Tampilan Hasil View

Anda bisa mencegah tampilan kolom-kolom dan memelihara semua kolom di dalam daftar kolom-kolom yang dipilih. Tetapi jika Anda menyimpan view dengan kolom tidak dipilih, apabila nanti Anda me-load view ini, kolom tersebut tidak akan menjadi bagian dari view lagi.

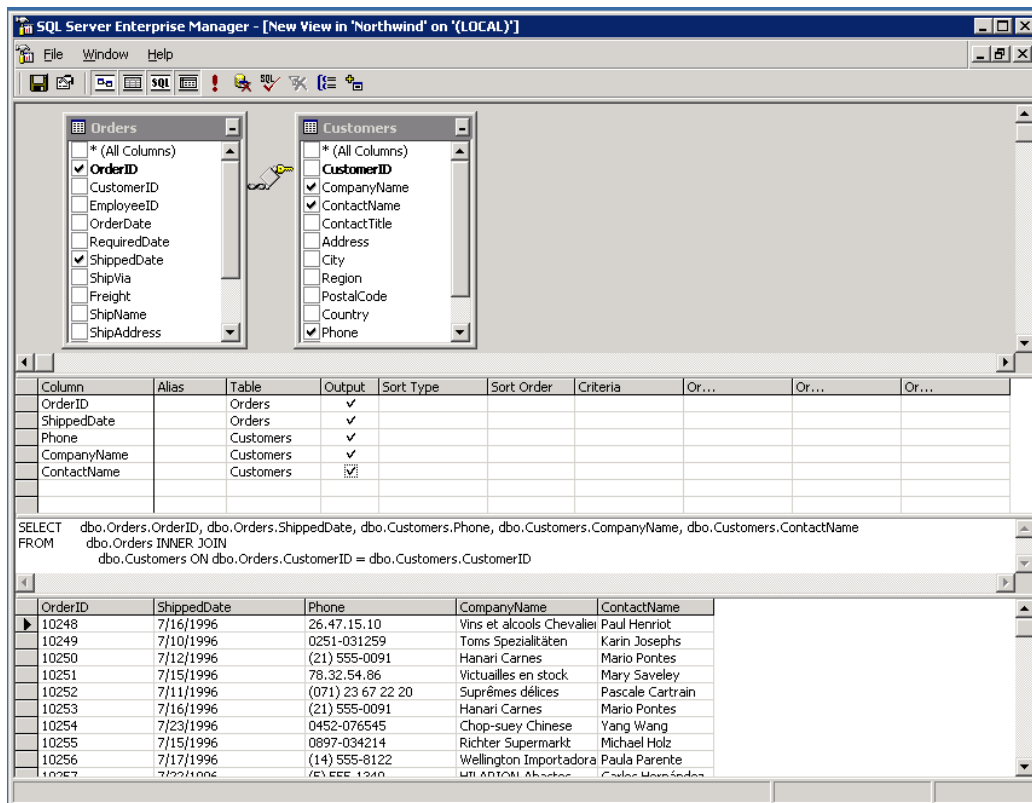
Mengubah Urutan Kolom

Untuk mengubah urutan kolom, Anda bisa menggeser kolom untuk memindahkannya ke posisi yang baru. Sebagai contoh, untuk membuat kolom Phone muncul pada kolom ketiga, pilihlah nama kolom tersebut, klik pada tombol abu-abu di sebelah kiri namanya, lalu geserlah ke atas garis kolom CompanyName seperti ditunjukkan di bawah ini.

Column	Alias	Table	Output	Sort Type	Sort Order	Criteria	Or...
OrderID		Orders	✓				
ShippedDate		Orders	✓				
CompanyName		Customers	✓				
ContactName		Customers					
Phone		Customers	✓				

Gambar 8.9 Mengubah Urutan Hasil View

Untuk melihat perubahannya, Anda perlu menjalankan kembali view. Inilah hasilnya :



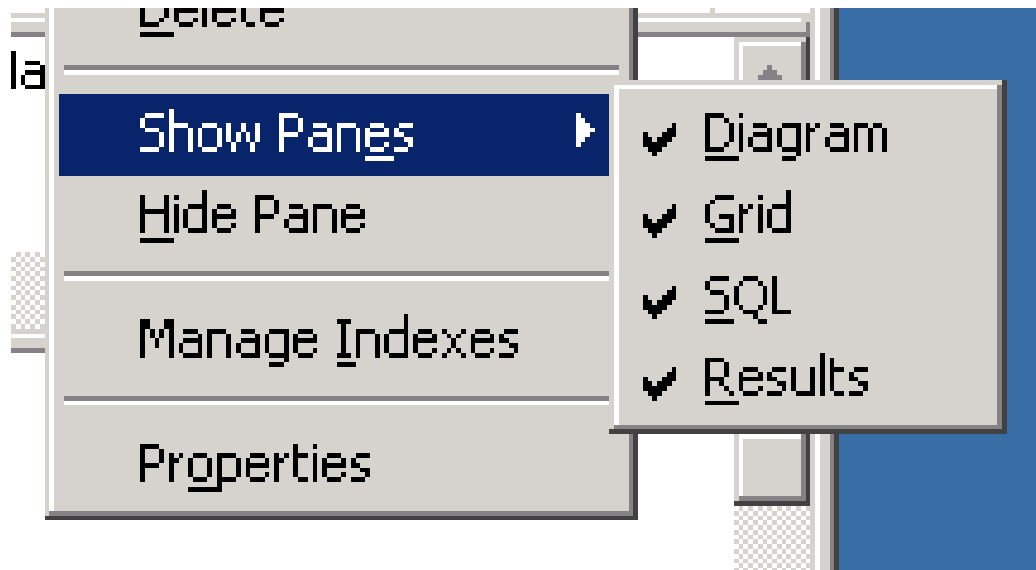
Gambar 8.10 Hasil Urutan View

Menyisipkan Kolom Baru

Untuk menyisipkan kolom baru di dalam view, pilihlah kolom yang diinginkan pada panel atas. Kolom yang baru ini akan selalu disisipkan pada posisi terakhir. Anda bisa memindahkannya ke tempat lain pada view dengan menggunakan cara yang telah dijelaskan sebelumnya.

Mengatur Panel

Anda bisa mengubah ukuran panel dengan mengklik dan menggeser garis-garis yang memisahkannya. Untuk menyembunyikan panel, atau untuk menampilkannya kembali setelah disembunyikan, klik tombol mouse kanan pada sebuah panel, pilih Show Panes, lalu pilih atau matikan pilihan pada panel yang ingin disembunyikan atau ditampilkan.



Gambar 8.11 Mengatur Tampilan Panes

Mengubah Nama View

Anda hanya dapat mengubah nama tampilan yang dibuat pengguna. Anda dapat mengganti nama tampilan menggunakan menu konteks yang muncul saat Anda mengklik kanan nama tampilan, atau dengan menekan F2 saat tampilan dipilih di panel informasi.

Menghapus View

Jika tampilan dihapus, tabel asli tidak terpengaruh. Hanya definisi yang dihilangkan. Untuk menghapus tampilan, klik kanan nama tampilan yang ingin Anda hapus, lalu tekan Hapus saat tampilan dipilih di panel informasi. Ini akan membuka dialog tempat Anda dapat mengonfirmasi untuk menghapus tabel.

Kesimpulan

Pada bab ini, Anda telah melihat betapa bermanfaatnya view sebagai sarana untuk memberikan gambaran yang berbeda dari informasi yang sama pada user yang berbeda. Anda telah melihat cara membuat, mengubah, mengganti nama, serta menghapus view. Pada bab berikutnya, kita akan membahas topik tingkat lanjut: prosedur tersimpan.

BAB 9

Prosedur

Sebuah prosedur pada dasarnya adalah sebuah program yang ditulis dalam bahasa Transact-SQL dan disimpan dalam database SQL Server. Prosedur dibentuk oleh perintah, variabel, dan proses logis yang terkandung dalam SQL. Prosedur dapat dilakukan secara manual atau dengan bantuan program lain.

Prosedur tersimpan sangat berguna dalam lingkungan klien-server untuk kinerja dan pemeliharaan. Karena banyak program dapat menggunakan satu metode, administrasi sistem menjadi lebih mudah karena perubahan segera tercermin ke semua pengguna. Setelah operasi dijalankan di server, akses selanjutnya cepat karena rencana eksekusi disimpan di memori. Seperti bahasa lainnya, prosedur SQL Server dapat menerima parameter input dan meneruskan nilai tertentu sebagai parameter output ke program yang menjalankannya. Prosedur dapat memanggil prosedur lain. Itu juga dapat menghasilkan nilai status yang menunjukkan apakah itu berhasil diselesaikan atau menghasilkan kesalahan.

Meskipun prosedur mengembalikan parameter keluaran, mereka berbeda dari fungsi karena tidak mengembalikan nilai yang dapat diubah namanya dan tidak dapat digunakan dalam ekspresi. Prosedur dapat dibuat dengan perintah Transact-SQL CREATE PROCEDURE dan dimodifikasi dengan perintah ALTER PROCEDURE.

Struktur Prosedur

Saat Anda membuat prosedur, Anda harus menentukan nama prosedur, parameternya, dan isi yang berisi perintah Transact-SQL untuk dijalankan. Contoh berikut ini menam pilkan prosedur yang menerima judul buku sebagai parameternya, lalu mengembalikan jumlah buku yang dijual dalam setahun. Penjelasan dari kode program terdapat di bawahnya. Bab 9

1. CREATE PROCEDURE get_sales_for_title
2. @title varchar (80),
3. @ytd-sales int OUTPUT AS
4. SELECT @ytd_sales = ytd_sales
5. FROM titles
6. WHERE title = @title
7. RETURN
8. GO

Baris	Arti
1	Membuat prosedur bernama <code>get_sales_for_title</code>
2	Membuat parameter input bernama <code>@ title</code>
3	Membuat parameter output bernama <code>@ ytd_sales</code>
4-6	Badan dari prosedur
7	Selesai menjalankan prosedur

Jenis-Jenis Prosedur

Prosedur dapat dibagi menjadi prosedur lokal yang disimpan di server lokal atau prosedur jarak jauh yang disimpan di server lain. Prosedur juga dapat diklasifikasikan sebagai prosedur yang ditentukan pengguna atau prosedur sistem yang telah dibuat sebelumnya dan merupakan bagian dari server. Pada akhir dari bab ini, akan ditampilkan daftar dari prosedur sistem.

Komponen dari Prosedur

Kini mari kita lihat komponen-komponen dari suatu prosedur. Komponen ini adalah parameter dan kode yang dihasilkan .

Parameter

Parameter memungkinkan komunikasi antara proses dan dunia luar. Ketika sebuah program melakukan operasi, ia dapat meneruskan nilainya dalam bentuk parameter. Prosedur dapat menggunakan nilai-nilai ini untuk melakukan tugasnya. Sebagai contoh, Anda bisa mengirimkan kode produk yang akan digunakan oleh perintah `SELECT` untuk menyaring query. Prosedur dapat berisi hingga 255 parameter dan kode hasil.

Nama dan Tipe Data

Parameter harus memiliki nama yang bermakna dan selalu dimulai dengan tanda `@`. Anda juga harus menyertakan definisi tipe data. Contoh berikut membuat parameter bernama kode tipe `Varchard` dan ukuran 8.

```
CREATE PROCEDURE test1
@code varchar(8)
```

Arah dari Parameter

Semua parameter yang dihasilkan dianggap sebagai parameter input, yaitu mereka menerima data dari program yang menjalankan prosedur. Dengan menambahkan kata OUTPUT ke definisi parameter, prosedur dapat mengembalikan nilai parameter ke program pemanggil.

```
CREATE PROCEDURE test1
@code varchar(8)
@totalcode int OUTPUT
```

Kode yang Dihasilkan

Prosedur bisa mengembalikan hasil nilai bertipe integer yang disebut return code, untuk menunjukkan apakah eksekusi dari prosedur tersebut berhasil. Kode ini harus dibuat sebagai parameter output dengan pilihan OUTPUT, dan harus dikembalikan kepada program yang memanggil prosedur tersebut melalui perintah RETURN. User bisa menggunakan nilai-nilai positif ini dan membuat kode-kode berdasarkan kebutuhannya. Sebagai contoh :

Nilai	Arti
0	Prosedur dijalankan dengan sukses
1	Parameter input tidak ditentukan
2	Isi parameter salah

Perhatikanlah contoh di bawah ini:

```
CREATE-PF/OCEDURE test2
@code varchar(B) = NULL,
AS
IF @code IS NULL
BEGIN
PRINT "ERROR: Parameter not entered."
RETURN(1)
END
```

Kode Nilai

Nilai 0 menunjukkan bahwa prosedur dijalankan dengan sukses. Nilai negatif (dari-99 sampai-1) menunjukkan alasan dari kegagalan prosedur. Hanya 14 nilai pertama yang digunakan.

Nilai	Arti
0	Prosedur dijalankan dengan sukses
-1	Objek hilang
-2	Error tipe data
-3	Proses menemukan dead lock
-4	Error izin
-5	Error sintaks
-6	Error rupa-rupa
-7	Error sumber daya
-8	Error internal tidak fatal
-9	Batasan sistem telah tercapai
-10	Error internal fatal
-11	Error internal fatal
-12	Tabel atau indeks rusak
-14	Error pada perangkat keras ditemukan

Membuat Prosedur

Prosedur dibuat dengan menggunakan perintah `CREATE PROCEDURE`. Proses pembuatan yang sesungguhnya bisa dilakukan dengan Enterprise Manager, Query Analyzer, atau Stored Procedures Wizard. Dengan pengecualian dari perintah `CREATE`, hampir semua perintah SQL bisa disertakan di dalam prosedur. Ini adalah perintah-perintah yang tidak boleh digunakan di dalam prosedur :

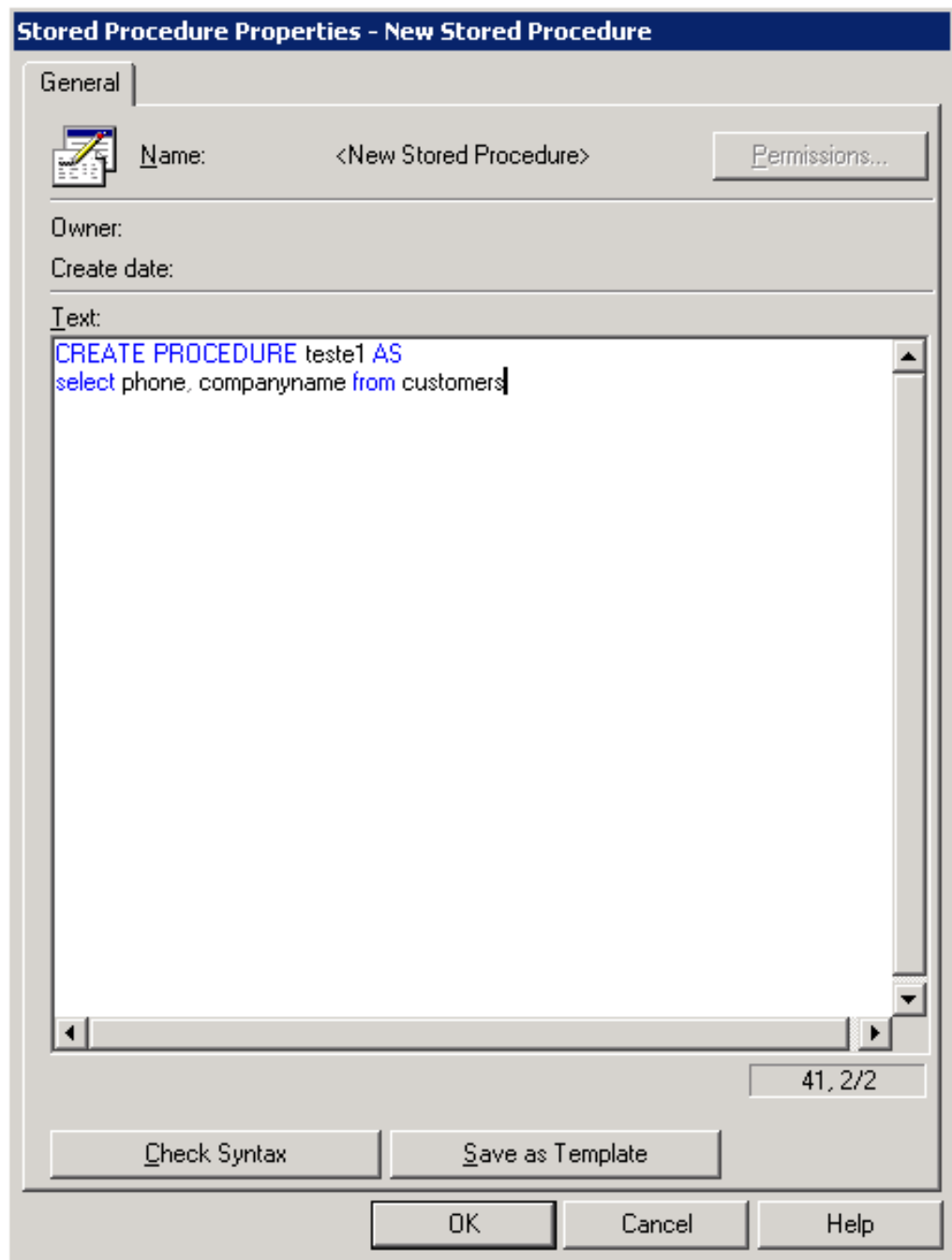
<code>CREATE DEFAULT</code>	<code>CREATE TRIGGER</code>
<code>CREATE PROCEDURE</code>	<code>CREATE VIEW</code>
<code>CREATE RULE</code>	

Apabila prosedur memanggil prosedur lain, prosedur kedua bisa mengakse semua objek yang dibuat oleh prosedur yang pertama. Apabila prosedur yang disimpan di tempat yang terpisah dijalankan, perubahan yang diubah oleh prosedur tersebut pada remote server tidak bisa dimundurkan. Anda bisa membuat prosedur sementara dengan menambahkan `#` dan `##` sebelum nama dari prosedur tersebut.

Untuk contoh, kita akan membuat sebuah prosedur pada database Northwind. Pertama-tama perbesarlah folder dari database Northwind, lalu klik tombol mouse kanan pada folder Stored Procedures. Pada menu shortcut, pilihlah New Stored Procedure. Sebuah kotak dialog akan dibuka, di mana Anda bisa mengetikkan perintah-perintah dari prosedur tersebut.

Perhatikanlah prosedur yang dibuat pada gambar di bawah ini. Prosedur ini akan menampilkan isi dari kolom Phone dan CompanyName dari tabel Customers. Nama dari

prosedur bisa ditentukan di dalam tanda kutip atau tanda kurung.



Gambar 9.1 Create Procedure

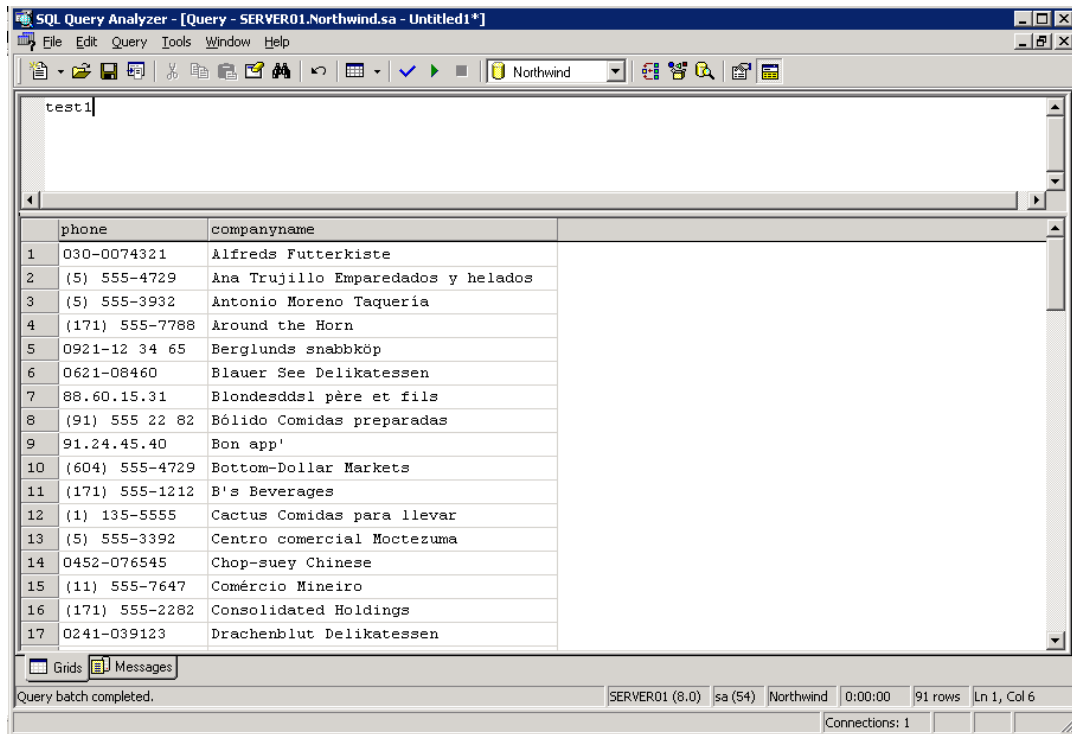
Tekanlah tombol OK untuk menyimpan prosedur tersebut. Berikutnya akan muncul daftar prosedur yang terdapat pada database Northwind.

Menjalankan Prosedur

Prosedur bisa dijalankan dengan perintah EXECUTE atau EXEC. Selain kode hasil,

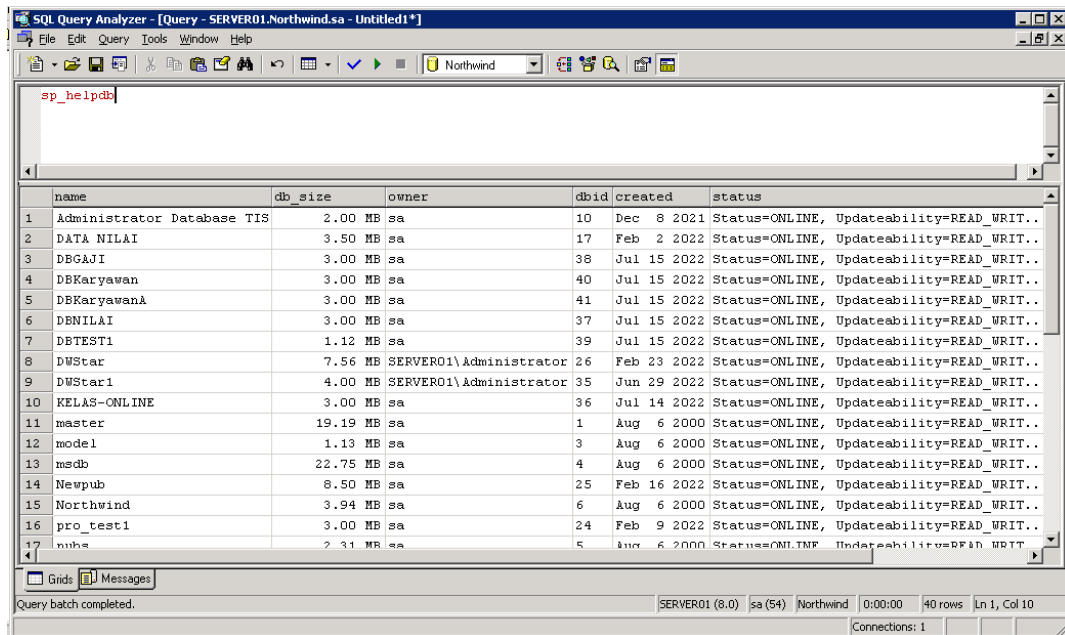
prosedur bisa mengem balikan set-set data yang dibuat oleh perintah SELECT dan pesan-pesan error yang dihasilkan karena proses SQL Server yang salah. Pesan-pesan yang dihasilkan oleh perintah PRINT juga bisa dikembalikan.

Dengan menggunakan Query Analyzer, pilihlah database pada field Database, lalu ketikkan nama dari prosedur dan tekan tombol Run. Dalam hal ini, perintah EXECUTE sifatnya pilihan. Ini adalah hasil dari prosedur yang baru saja kita buat:



Gambar 9.2 Menjalankan Procedure

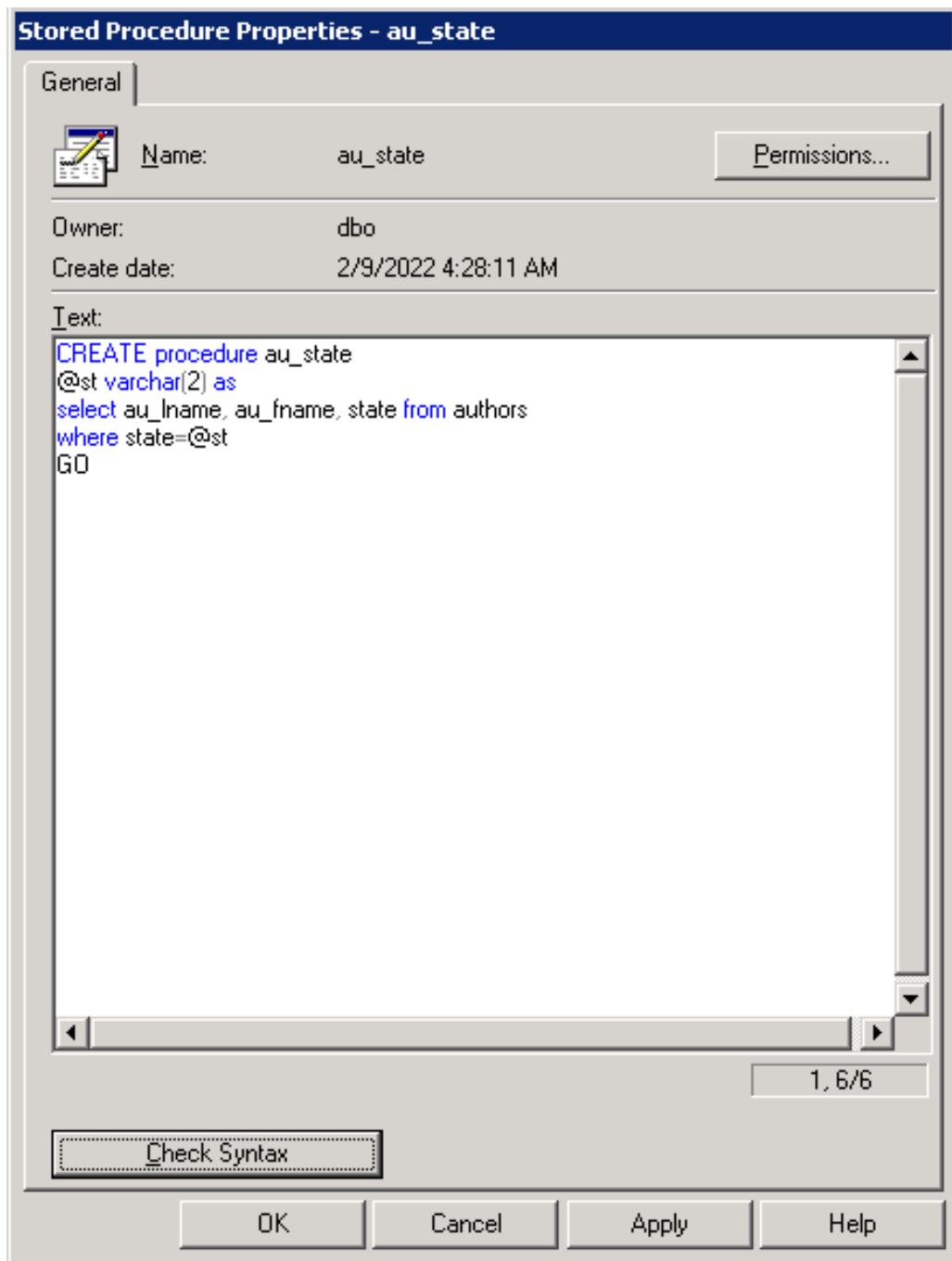
Eksekusi dari prosedur sistem juga sama. Tentukanlah namanya dan semua parameter yang digunakannya. Gambar di bawah ini menampilkan eksekusi dari prosedur sistem bernama sp_helpdb yang memberikan informasi mengenai database di dalam server.



Gambar 9.3 Menjalankan Procedure Tersimpan

Membuat Prosedur dengan Parameter

Menyertakan parameter di dalam prosedur sangatlah mudah. Anda hanya perlu menentukan nama dari parameter tersebut dan tipe datanya. Apabila ada lebih dari satu parameter, parameter tersebut harus dipisahkan dengan tanda koma. Sebagai contoh, kita akan membuat sebuah prosedur bernama `au_state` di dalam database Pubs yang menggunakan parameter penulis yang tinggal di negara bagian tertentu. Ini adalah kode untuk prosedur ini:



Gambar 9.4 Membuat Prosedur dengan Parameter

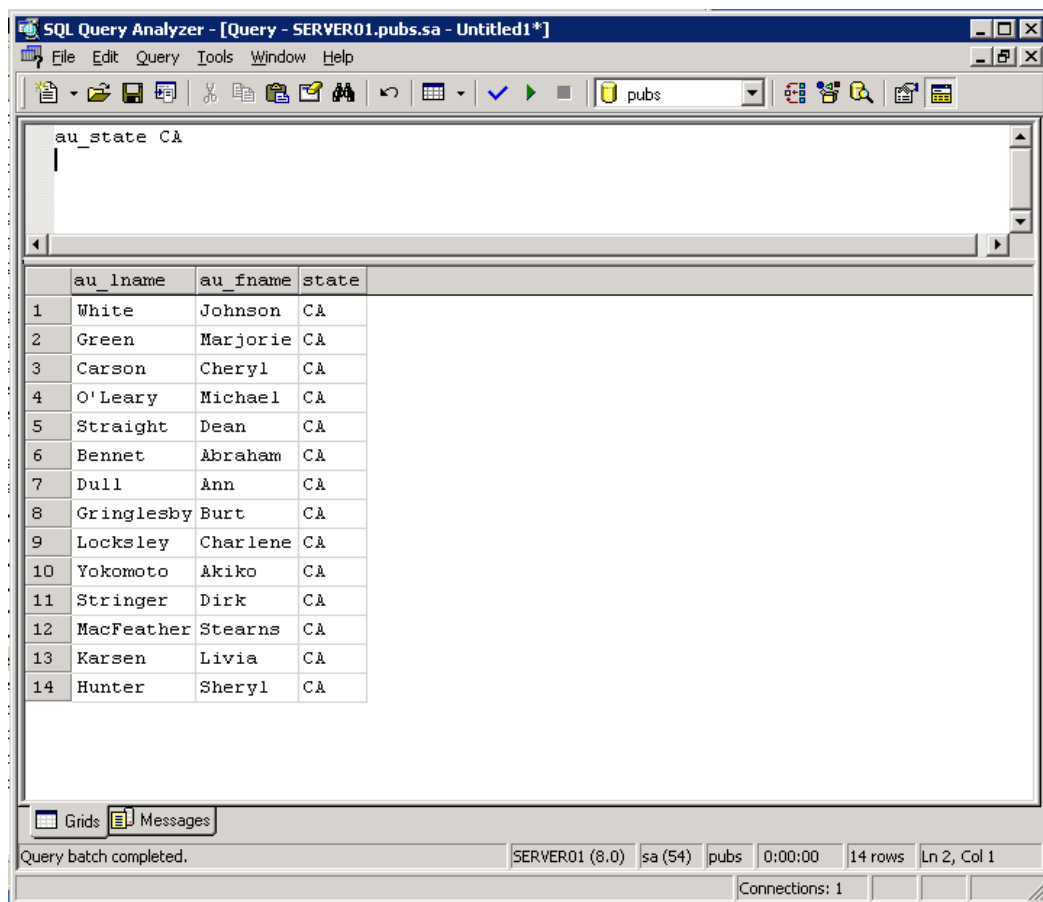
Jalankan perintah `exec au_state CA` untuk memberikan hasil semua penulis di negara bagian California.

Mengubah Prosedur

Untuk mengubah prosedur, Anda harus menggunakan perintah `ALTER PROCEDURE` diikuti dengan nama prosedur dan perubahan yang diperlukan. Pada contoh berikut ini, kita akan mengubah prosedur `au_state` untuk menyertakan sebuah kolom tambahan.

```
ALTER PROCEDURE au_state
@st varchar(2) AS
SELECT au_lname, au_fname, state from authors
where state=@st
```

Untuk menghemat pengetikan, copy teks untuk pembuatan prosedur, sisipkanlah pada daerah pengetikan, lalu buatlah perubahan yang dibutuhkan. Apabila prosedur ini dijalankan dengan parameter CA, ia akan menampilkan kolom baru seperti ditunjukkan di bawah ini.



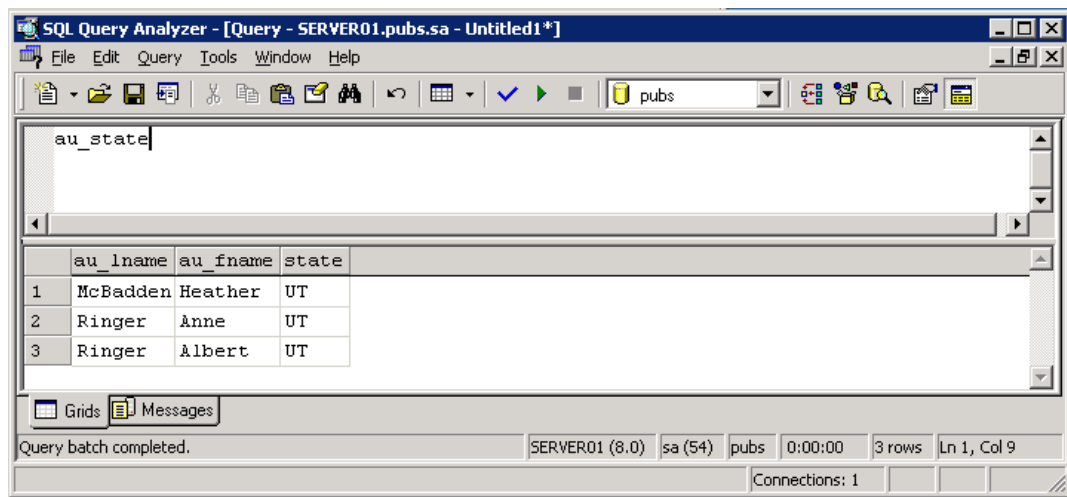
Gambar 9.5 Menjalankan Prosedur dengan Parameter CA

Menggunakan Nilai Standar sebagai Parameter

Nilai standar bisa dimasukkan sebagai parameter pada perintah CREATE PROCEDURE. Nilai ini digunakan apabila user tidak menentukan nilai tertentu sebagai parameter. Ubahlah prosedur au_state dengan menggunakan perintah ALTER PROCEDURE, dan gunakanlah nilai standar sebagai parameter negara bagian.

```
ALTER PROCEDURE au_state
@st varchar(2)='UT' AS
SELECT au_lname , au_fname, state from authors
where state=@ st
```

Ini adalah hasil dari eksekusi prosedur di atas apabila tidak ada parameter yang ditentukan:



Gambar 9.6 Menjalankan Prosedur Menggunakan Nilai Standar sebagai Parameter

Mengkompilasi Ulang Prosedur

Prosedur secara otomatis dikompilasi pada saat pertama kali ia dieksekusi. Apabila ada indeks baru yang dibuat dan berhubungan dengan prosedur, prosedur tersebut tidak akan dikompilasi ulang secara otomatis. Proses kompilasi ulang hanya akan terjadi apabila isi dari tabel yang digunakan oleh prosedur diubah. Untuk memaksa kompilasi ulang, Anda bisa menggunakan prosedur sistem `sp_recom` pile.

Contoh : stored procedure_recompile au_state

Menghapus Prosedur

Prosedur dapat dijatuhkan menggunakan perintah `DROP PROCEDURE` seperti yang ditunjukkan di bawah ini atau melalui Enterprise Manager. Untuk menghapus tindakan, pilih nama tindakan di area informasi lalu tekan tombol Hapus atau gunakan opsi Hapus di menu konteks.

Contoh : `DROP PROCEDURE test1`

Prosedur Sistem

SQL Server memiliki lebih dari 100 prosedur khusus yang menjalankan tugas-tugas administratif. Prosedur-prosedur sistem ini bisa dijalankan dari database mana saja. Prosedur sistem ini disimpan di dalam database Master.

Prosedur ini dapat mudah dikenali, karena namanya selalu diawali dengan kata "sp_". Jika seorang user membuat prosedur dengan awalan dan nama yang mirip dengan prosedur sistem, prosedur tersebut tidak akan dijalankan. Prosedur sistem dikelompokkan berdasarkan kategori berikut ini:

Kategori	Keterangan
Catalog Procedures	Mengimplementasikan fungsi kamus data ODBC
Cursor Procedures	Mengimplementasikan fungsi kursor
Distributed Queries Procedures	Mengelola query terdistribusi
SOL Agent Procedures	Menentukan event-event
Replication Procedures	Mengelola replikasi
Security Procedures	Mengelola sumber daya keamanan
System Procedures	Melakukan fungsi-fungsi pemeliharaan

Kategori	Keterangan
Web Assistant Procedures	Digunakan oleh Web Assistant Wizard
General Extended Procedures	Mengelola antarmuka dengan program eksternal
SOL Mail Extended Procedures	Menjalankan operasi e-mail dengan SOL
SOL Server Profiler Extended Procedures	Mendukung operasi SOL Server Profiler
OLE Automation Procedures	Mendukung operasi otomatisasi OLE
Data Transformation Services Procedures	Mendukung operasi DTS

Berikut ini adalah daftar prosedur sistem berdasarkan kategorinya. Nama-nama dari sebagian besar prosedur sistem ini sudah menjelaskan fungsinya. Keterangan yang lengkap dari prosedur-prosedur ini bisa ditemukan pada help online.

Catalog Procedures

sp_column_privileges	sp_special_columns
sp_columns	sp_sproc_columns
sp_databases	sp_statistics
sp_fkeys	sp_stored_procedures
sp_pkeys	sp_table_privileges
sp_server_info	sp_tables

Cursor Procedures

sp_cursor_list	sp_describe_cursor_tables
sp_describe_cursor_columns	

Distributed Queries Procedures

sp_addlinkedserver	sp_indexes
sp_addlinkedsrvlogin	sp_linkedservers
sp_catalogs	sp_primarykeys
sp_column_privileges_ex	sp_serveroption
sp_columns_ex	sp_table_privileges_ex
sp_droplinkedsrvlogin	sp_tables_ex
sp_foreignkeys	

SOL Server Agent Procedures

sp_add_alert	sp_help_jobschedule
sp_add_category	sp_help_jobserver
sp_add_job	sp_help_jobstep
sp_add_jobschedule	sp_help_notification
sp_add_jobserver	sp_help_operator
sp_add_jobstep	sp_helphistory
sp_add_notification	sp_help_targetserver
sp_add_operator	sp_help_targetservergroup
sp_add_targetservergroup	sp_helptask
sp_add_targetsvrgrp_member	sp_managejobs_by_login
sp_addtask	sp_msx_defect
sp_applyjob_to_targets	sp_msx_enlist
sp_delete_alert	sp_post_msx_operation
sp_delete_category	sp_purgejobhistory
sp_deletejob	sp_purgehistory
sp_deletejobschedule	sp_reassigntask
sp_deletejobserver	sp_removejob_from_targets
sp_deletejobstep	sp_resync_targetserver
sp_delete_notification	sp_startjob
sp_delete_operator	sp_stopjob
sp_delete_targetserver	sp_update_alert
sp_delete_targetservergroup	sp_update_category
sp_delete_targetsvrgrp_member	sp_update_job
sp_droptask	sp_update_jobschedule
sp_help_alert	sp_update_jobstep
sp_help_category	sp_update_notification
sp_help_downloadlist	sp_update_operator
sp_helpjob	sp_update_targetservergroup
sp_helpjobhistory	sp_updatetask

Replication Procedures

sp_add_agent_parameter	sp_enumcustomresolvers
sp_add_agent_profile	sp_enumfullsubscribers
sp_addarticle	sp_generatefilters
sp_adddistpublisher	sp_getmergedeletetype
sp_adddistributiondb	sp_get_distributor
sp_adddistributor	sp_grant_publication_access
sp_addmergearticle	sp_help_agent_default
sp_addmergefilter	sp_help_agent_parameter
sp_addmergepublication	sp_help_agent_profile
sp_addmergepullsubscription	sp_help_publication_access
sp_addmergepullsubscription_agent	sp_helparticle
sp_addmergesubscription	sp_helparticlecolumns
sp_addpublication	sp_helpdistpublisher
sp_addpublication_snapshot	sp_helpdistributiondb
sp_addpublisher	sp_helpdistributor
sp_addpullsubscription	sp_helpmergearticle
sp_addpullsubscription_agent	sp_helpmergearticleconflicts
sp_addsubscriber	sp_helpmergeconflictrows
sp_addsubscriber_schedule	sp_helpmergedeleteconflictrows
sp_addsubscription	sp_helpmergefilter
sp_addsynctriggers	sp_helpmergepublication
sp_article_validation	sp_helpmergepullsubscription
sp_articlecolumn	sp_helpmergesubscription
sp_articlefilter	sp_helppublication
sp_articlesynctrprocs	sp_helppublication_snapshot
sp_articleview	sp_helppullsubscription
sp_browsereplcmds	sp_helpreplicationdb
sp_changearticle	sp_helpreplicationdboption
sp_changedistpublisher	sp_helpreplicationoption
sp_changedistributiondb	sp_helpsubscriber
sp_changedistributor_password	sp_helpsubscriberinfo
sp_changedistributor_property	sp_helpsubscription
sp_changemergearticle	sp_helpsubscription_properties
sp_changemergefilter	sp_link_publication
sp_changemergepublication	sp_mergedummyupdate
sp_changemergepullsubscription	sp_mergesubscription_cleanup
sp_changemergesubscription	sp_publication_validation
sp_changepublication	sp_refreshsubscriptions
sp_changesubscriber	sp_reinitmergepullsubscription
sp_changesubscriber_schedule	sp_reinitmergesubscription
sp_changesubstatus	sp_reinitpullsubscription

sp_changemergearticle	sp_helpsubscription_properties
sp_changemergefilter	sp_link_publication
sp_changemergepublication	sp_merGEDummyupdate
sp_changemergepullsubscription	sp_mergesubscription_cleanup
sp_changemergesubscription	sp_publication_validation
sp_changepublication	sp_refreshsubscriptions
sp_changesubscriber	sp_reinitmergepullsubscription
sp_changesubscriber_schedule	sp_reinitmergesubscription
sp_changesubstatus	sp_reinitpullsubscription
sp_change_subscription_properties	sp_removedbreplication
sp_deletemergeconflictrow	sp_replcmds
sp_distcounters	sp_replcounters
sp_drop_agent_parameter	sp_repldone
sp_drop_agent_profile	sp_replflush
sp_droparticle	sp_replicationdboption
sp_dropdistributiondb	sp_replicationoption
sp_dropdistributor	sp_replication_agent_checkup
sp_dropmergearticle	sp_replshowcmds
sp_dropmergefilter	sp_repltrans
sp_dropmergepublication	sp_revoke_publication_access
sp_dropmergepullsubscription	sp_script_synctran_commands
sp_dropmergesubscription	sp_scriptdelproc
sp_droppublication	sp_scriptinsproc
sp_droppullsubscription	sp_scriptupdproc
sp_dropsubscriber	sp_subscription_cleanup
sp_dropsubscription	sp_table_validation
sp_dumpparamcmd	

Security Procedures

sp_addalias	sp_droprole
sp_addapprole	sp_droprolemember
sp_addgroup	sp_dropserver
sp_addlinkedrvlogin	sp_dropsvrolemember
sp_addlogin	sp_dropuser
sp_addremotelogin	sp_grantdbaccess
sp_addrole	sp_grantlogin
sp_addrolemember	sp_helpdbfixedrole
sp_addserver	sp_helpgroup
sp_addsvrolemember	sp_helplogins
sp_adduser	sp_helpntgroup
sp_approlepassword	sp_helpremotelogin
sp_change_users_login	sp_helprole
sp_changedbowner	sp_helprolemember

sp_changegroup	sp_helpprotect
sp_changeobjectowner	sp_helpsrvrole
sp_dbfixedrolepermission	sp_helpsrvrolemember
sp_defaultdb	sp_helpuser
sp_defaultlanguage	sp_password
sp_denylogin	sp_remoteoption
sp_dropalias	sp_revokedbaccess
sp_dropapprole	sp_revokelgin
sp_dropgroup	sp_setapprole
sp_droplinkedsvrlogin	sp_srvrolepermission
sp_droplogin	sp_validatelogins
sp_dropremotelgin	

System Procedures

sp_addextendedproc	sp_helpdb
sp_addmessage	sp_helpdevice
sp_addtype	sp_helpextendedproc
sp_addumpdevice	sp_helpfile
sp_altermessage	sp_helpfilegroup
sp_autostats	sp_help_fulltext_catalogs
sp_attach_db	sp_help_fulltext_catalogs_cursor
sp_attach_single_file_db	sp_help_fulltext_columns
sp_bindefault	sp_help_fulltext_columns_cursor
sp_bindrule	sp_help_fulltext_tables
sp_bindsession	sp_help_fulltext_tables_cursor
sp_certify_removable	sp_helpindex
sp_configure	sp_helplanguage
sp_create_removable	sp_helpserver
sp_createstats	sp_helpsort
sp_datatype_info	sp_helptext
sp_dbcmptlevel	sp_helptrigger
sp_dboption	sp_indexoption
sp_depends	sp_lock
sp_detach_db	sp_monitor
sp_dropdevice	sp_processmail
sp_dropextendedproc	sp_procoption
sp_dropmessage	sp_recompile
sp_droptype	sp_refreshview
sp_dsninfo	sp_rename
sp_enumdsn	sp_renamedb
sp_executesql	sp_serveroption
sp_getbindtoken	sp_setnetname
sp_fulltext_catalog	sp_spaceused

sp_fulltext_column	sp_tableoption
sp_fulltext_database	sp_unbindefault
sp_fulltext_service	sp_unbindrule
sp_fulltext_table	sp_updatestats
sp_help	sp_validname
sp_helpconstraint	sp_who

Web Assistant Procedures

sp_dropwebtask	sp_makewebtask
sp_enumcodepages	sp_runwebtask

General Extended Procedures

xp_cmdshell	xp_revokelogin
xp_enumgroups	xp_snmp_getstate
xp_findnextmsg	xp_snmp_raisetrap
xp_grantlogin	xp_sprintf
xp_loginconfig	xp_sqlinventory
xp_logininfo	xp_sscanf
xp_logevent	xp_sqlmaint
xp_msver	

SOL Mail Extended Procedures

xp_deletemail	xp_sendmail
xp_findnextmsg	xp_startmail
xp_readmail	xp_stopmail

SOL Server Profiler Extended Procedures

xp_sqltrace	xp_trace_getuserfilter
xp_trace_addnewqueue	xp_trace_getwritefilter
xp_trace_deletequeuedefinition	xp_trace_loadqueuedefinition
xp_trace_destroyqueue	xp_trace_pausequeue
xp_trace_enumqueuedefname	xp_trace_restartqueue
xp_trace_enumqueuehandles	xp_trace_savequeuedefinition
xp_trace_eventclassrequired	xp_trace_setappfilter
xp_trace_generate_event	xp_trace_setconnectionidfilter
xp_trace_getappfilter	xp_trace_setcpufilter
xp_trace_getconnectionidfilter	xp_trace_setdbidfilter
xp_trace_getcpufilter	xp_trace_setdurationfilter
xp_trace_getdbidfilter	xp_trace_seteventclassrequired
xp_trace_getdurationfilter	xp_trace_seteventfilter
xp_trace_geteventfilter	xp_trace_sethostfilter

xp_trace_geteventnames	xp_trace_sethpidfilter
xp_trace_getevents	xp_trace_setindidfilter
xp_trace_gethostfilter	xp_trace_setntdmfilter
xp_trace_gethpidfilter	xp_trace_setntnmfilter
xp_trace_getindidfilter	xp_trace_setobjidfilter
xp_trace_getntdmfilter	xp_trace_setqueueautostart
xp_trace_getntnmfilter	xp_trace_setqueuecreateinfo
xp_trace_getobjidfilter	xp_trace_setqueuedestination
xp_trace_getqueueautostart	xp_trace_setreadfilter
xp_trace_getqueuedestination	xp_trace_setserverfilter
xp_trace_getqueueproperties	xp_trace_setseverityfilter
xp_trace_getreadfilter	xp_trace_setspidfilter
xp_trace_getserverfilter	xp_trace_settextfilter
xp_trace_getseverityfilter	xp_trace_setuserfilter
xp_trace_getspidfilter	xp_trace_setwritefilter
xp_trace_gettextfilter	

OLE Automation Extended Stored Procedures

sp_OACreate	sp_OAMethod
sp_OADestroy	sp_OASetProperty
sp_OAGetErrorInfo	sp_OAStop
sp_OAGetProperty	Object Hierarchy Syntax

Kesimpulan

Bab ini memberikan pengenalan yang singkat mengenai prosedur tersimpan, sebuah topik yang membutuhkan perhatian khusus pada SQL Server. Pada bab berikutnya, kita akan mempelajari cara bekerja dengan trigger, yaitu sebuah prosedur khusus yang secara otomatis akan dijalankan apabila muncul suatu event atau kejadian tertentu.

BAB 10

Trigger

Sebagian besar aplikasi yang dirancang untuk mengelola database desktop atau database perusahaan juga bertanggung jawab mengelola aturan bisnis sistem. Sebagai contoh, apabila ada sebuah produk yang kuantitas minimalnya sudah tercapai pada sistem kontrol inventori, program harus "memicu" (trigger) sebuah rutin yang membutuhkan pembelian lot baru serta pembuatan laporan kontrol. Jika database diakses oleh sarana yang mengizinkan perubahan atau oleh sistem lain yang tidak melakukan kontrol ini, inventori produk bisa jatuh di bawah kuantitas minimal tanpa ada pemberitahuan.

Salah satu fitur menarik dari SQL Server adalah mekanisme kontrol yang disebut pemicu basis data. Pemicu seperti serangkaian pernyataan Transact-SQL yang dijalankan secara otomatis saat pernyataan INSERT, DELETE, atau UPDATE dijalankan di atas meja.

Penggunaan utama trigger adalah untuk membuat metode validasi dan pembatasan akses pada database, mis. B. Rutinitas keselamatan. Alih-alih memberikan kontrol ke aplikasi, kita bisa mengontrol tabel itu sendiri dengan pemicu, jadi manipulasi database bisa jauh lebih aman. Aplikasi yang dapat dijalankan menggunakan trigger antara lain:

- Buat isi kolom dari kolom lain.
- Buat mekanisme validasi yang menyertakan kueri terhadap beberapa tabel.
- Buat log untuk mencatat penggunaan tabel.
- Perbarui spreadsheet lain bila ada penambahan atau perubahan lain pada spreadsheet yang aktif.

Komponen dari Trigger

Pemicu terdiri dari dua bagian:

- Perintah SQL untuk mengaktifkan pemicu. Perintah INSERT, DELETE, dan UPDATE dapat mengaktifkan pemicu. Trigger yang sama bisa diaktifkan apabila ada lebih dari satu aksi yang terjadi. Dengan kata lain, trigger bisa diaktifkan apabila perintah INSERT dijalankan, atau apabila perintah UPDATE dan DELETE dijalankan.
- Tindakan dilakukan dengan menggunakan pemicu. Pemicu mengeksekusi blok PL/SQL.

Batasan Trigger

Berikut ini adalah batasan dan pertimbangan untuk menggunakan pemicu:

- Pemicu dapat menjalankan perintah di tubuhnya atau mengaktifkan tindakan dan pemicu lain untuk melakukan tugas tertentu.
- Setiap perintah SET dapat diatur untuk memicu. Perintah ini tetap aktif saat pemicu sedang berjalan.
- Anda tidak dapat membuat pemicu pada tampilan. Tapi saat penglihatan digunakan, pelatuk alas tiang menyala secara normal.
- Apabila trigger dijalankan, hasilnya akan dikembalikan kepada aplikasi yang memanggilnya. Untuk menghindari mengembalikan hasil, jangan gunakan perintah SELECT yang mengembalikan hasil atau menjalankan atribut dari isi ke dalam variabel.
- Perintah TRUNCATE TABLE tidak bisa dicegat oleh trigger dengan tipe DELETE. Perintah WRITETEXT tidak bisa mengaktifkan trigger.
- Perintah-perintah Transact-SQL yang ditampikan di bawah ini tidak bisa digunakan di dalam trigger :

ALTER DATABASE	ALTER PROCEDURE	ALTER TABLE
ALTER TRIGGER	ALTER VIEW	CREATE DATABASE
CREATE DEFAULT	CREATE INDEX	CREATE PROCEDURE
CREATE RULE	CREATE SCHEMA	CREATE TABLE
CREATE TRIGGER	CREATE VIEW	DENY
DISK INIT	DISK RESIZE	DROP DATABASE
DROP DEFAULT	DROP INDEX	DROP PROCEDURE
DROP RULE	DROP TABLE	DROP TRIGGER
DROP VIEW	GRANT	LOAD DATABASE
LOAD LOG	RESTORE DATABASE	RESTORE LOG
REVOKE	RECONFIGURE	TRUNCATE TABLE
UPDATE STATISTICS		

Membuat Trigger

Trigger bisa dibuat dengan Enterprise Manager atau Query Analyzer. Dengan apapun Anda membuatnya, Anda perlu mengetikkan perintah-perintah SQL. Pertama-tama, mari kita lihat pada sintaks perintah SQL yang telah disederhanakan. Lihatlah panduan referensi pada Lampiran A untuk sintaks yang lebih lengkap.

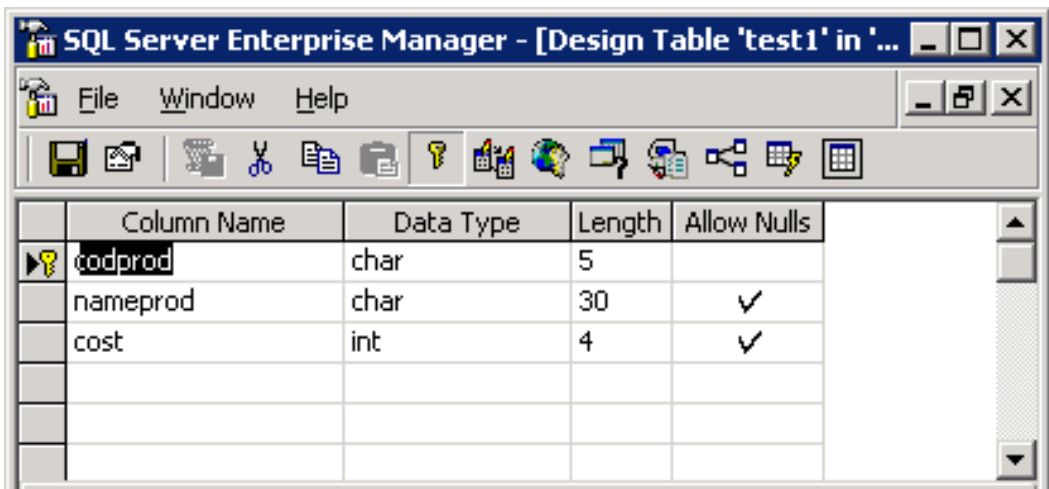
```
CREATE TRIGGER trigger_name ON table_name
FOR [INSERT/, DELETE/, UPDATE]
AS commands
```

ON tablename menentukan tabel atau skema tempat pemicu dibuat.

FOR harus diikuti dengan jenis perintah yang memicu pemicu.

AS memulai badan pemicu dengan perintah yang dapat dieksekusi.

Pada bab ini, kita akan membuat dua buah tabel yang strukturnya seperti terlihat pada gambar di bawah ini. Tabel ini dibuat di database Newpub2.



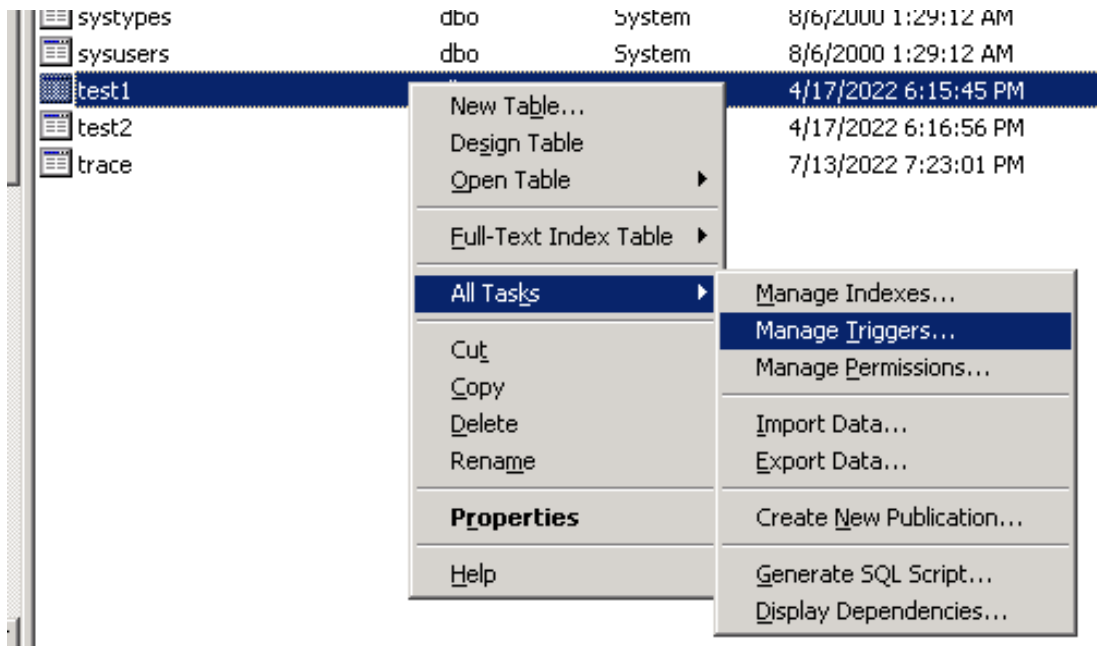
	Column Name	Data Type	Length	Allow Nulls
🔑	codprod	char	5	
	nameprod	char	30	✓
	cost	int	4	✓

Gambar 10.1 Tabel Test1

Tabel yang pertama bernama Test1 dan tabel yang kedua bernama Test2. Semua trigger akan dibuat di dalam tabel Test1.

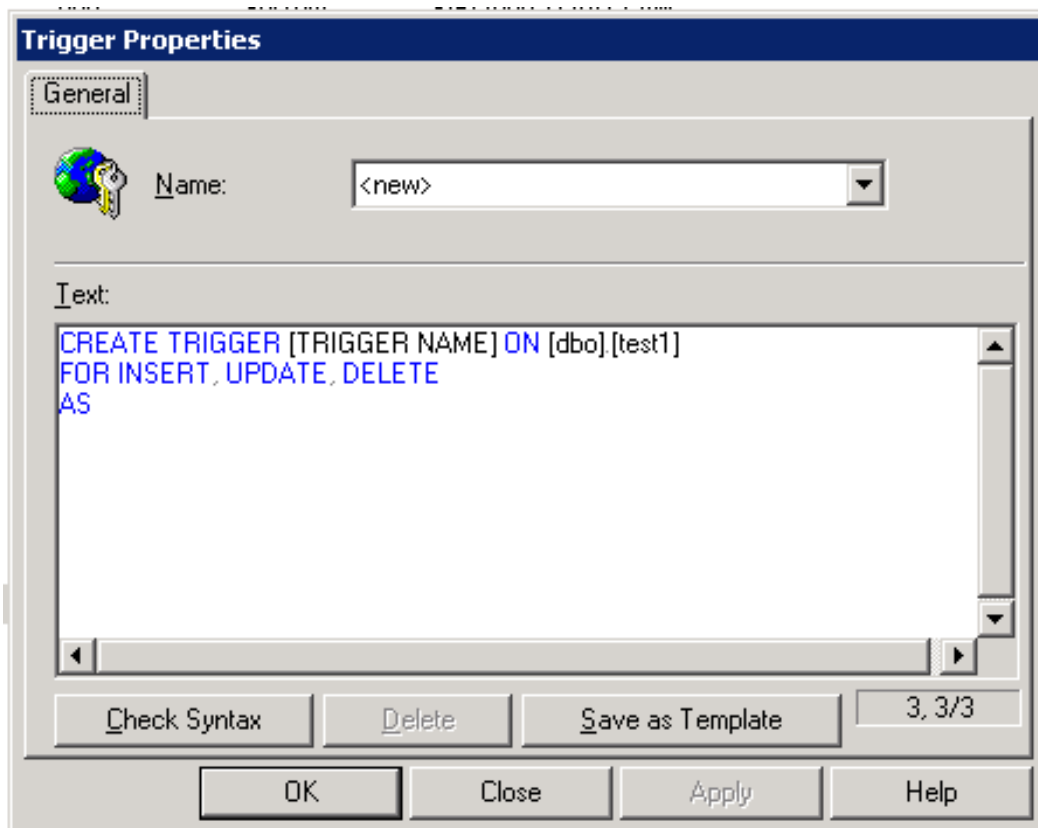
Membuat Trigger dengan Enterprise Manager

Perbesarlah database dan folder yang mengandung tabel. Kemudian klik kanan pada nama tabel yang ingin Anda buat pemicunya. Pilihlah Task pada menu dan pilih Manage Triggers. Prosedur ini juga dapat digunakan untuk mengubah atau menghapus trigger.



Gambar 10.2 Manage Triggers

Langkah ini akan menampilkan kotak dialog Trigger Properties.



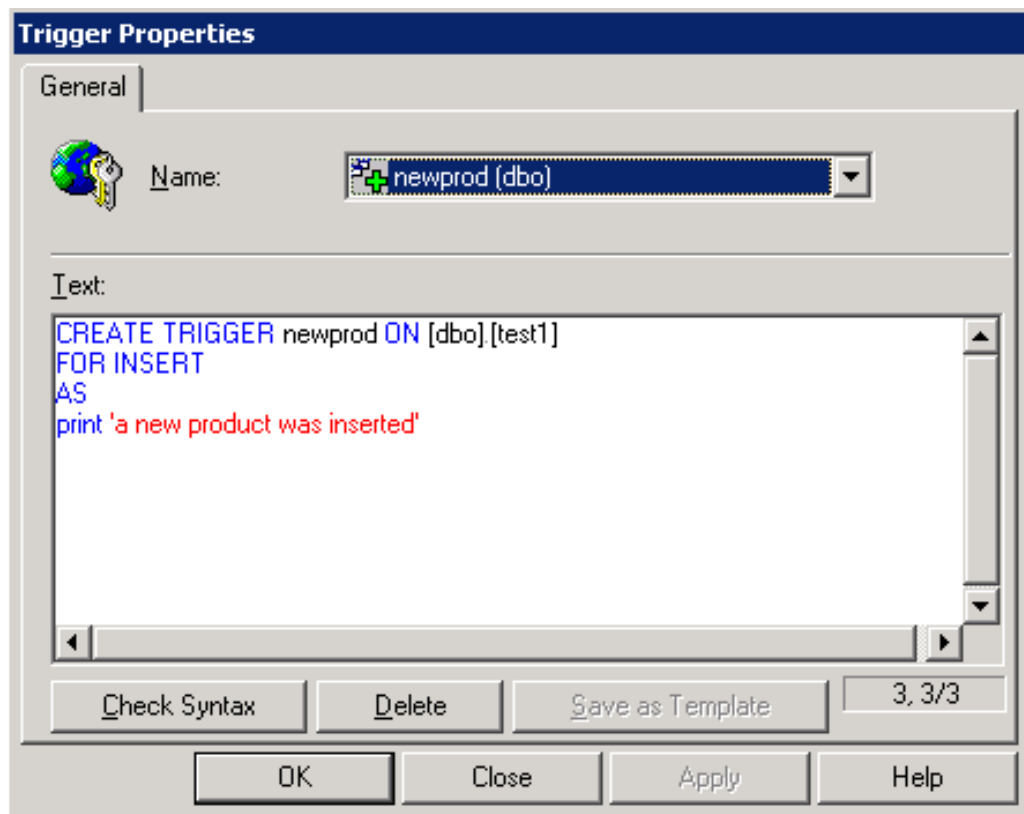
Gambar 10.3 Trigger Properties

Kotak dialog ini menampilkan panel utama dengan struktur dasar dari trigger. Anda

harus mengganti isi di dalam kurung [trigger name] dengan nama dari trigger yang akan dibuat. Perhatikan pada gambar di bawah ini, bahwa nama dari trigger akan muncul pada tempat yang tepat.

Pada baris yang kedua, pilihlah jenis dari aksi trigger, yaitu perintah yang akan mengaktifkannya. Defaultnya adalah INSERT, DELETE, dan UPDATE. Setelah kata AS, tentukanlah blok perintah yang akan dijalankan.

Trigger pertama yang akan dibuat akan menampilkan pesan pada layar yang mengatakan bahwa ada sebuah record yang disisipkan ke dalam tabel. Ini akan menunjukkan bahwa trigger benar-benar diaktifkan oleh perintah INSERT. Ketikkan perintah PRINT seperti ditunjukkan pada layar berikutnya, lalu tekan tombol OK untuk menyimpan trigger.

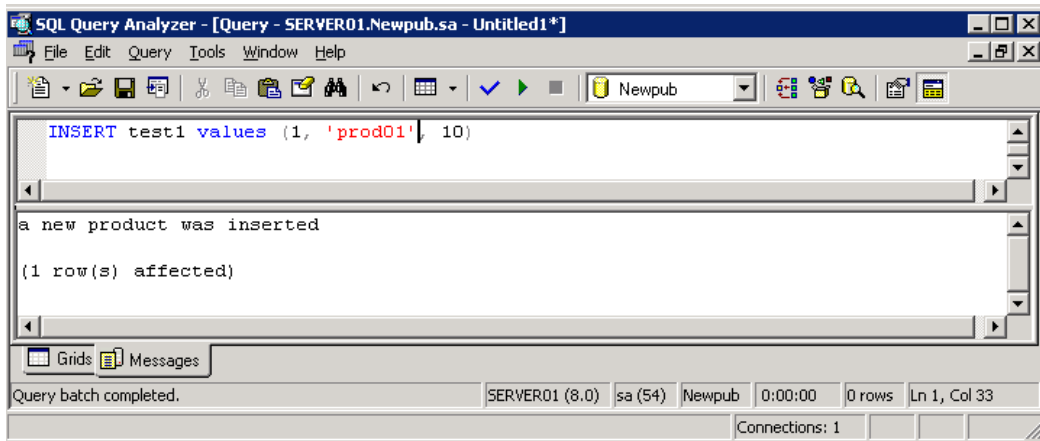


Gambar 10.3 Trigger INSERT

Kini kita akan menguji fungsi dari trigger ini dengan menggunakan Query Analyzer. Kita akan menyisipkan sebuah baris di dalam tabel Test1. Setelah me-load program, pilihlah database Newpub2 lalu ketikkan perintah berikut ini:

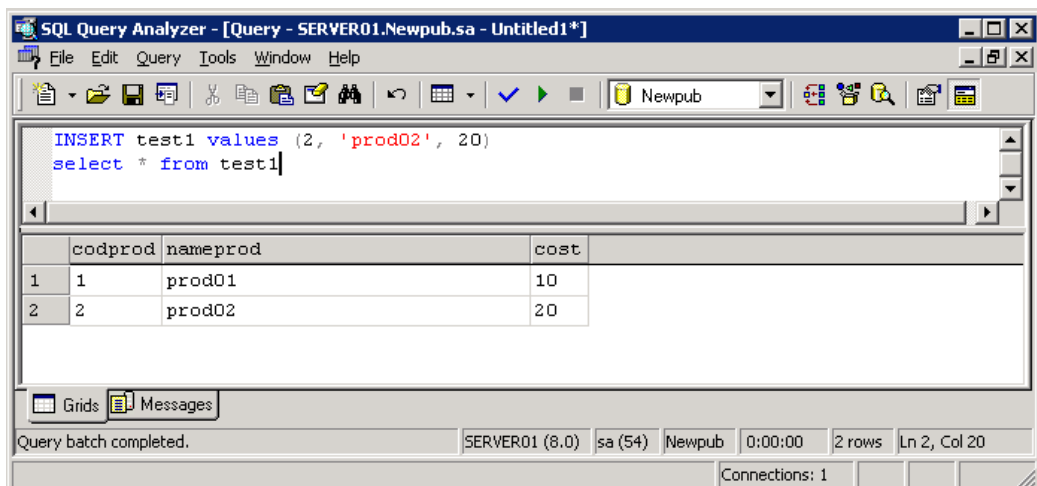
```
INSERT test1 values (1, "prod01", 10)
```

Perhatikanlah hasilnya pada gambar 10.3.



Gambar 10.3 Hasil Trigger INSERT

Kini kita akan menyisipkan sebuah record baru dan menampilkan isi dari tabel:

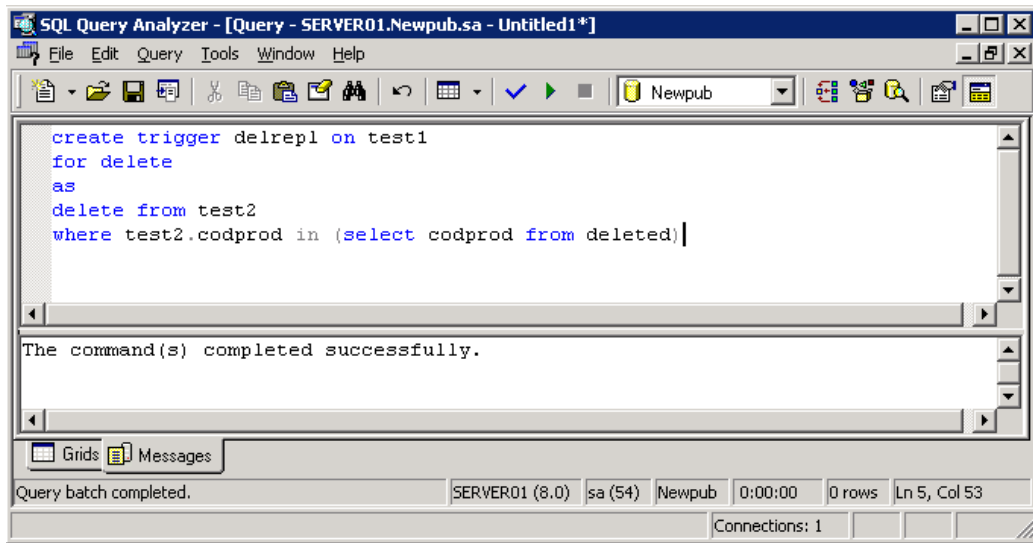


Gambar 10.4 Menambahkan Trigger INSERT

Pada bagian berikutnya, kita akan membuat dua trigger tambahan. Salah satunya akan menampilkan sebuah pesan apabila ada baris yang dihapus, dan trigger lain akan memberi tahu bahwa ada baris yang telah diubah.

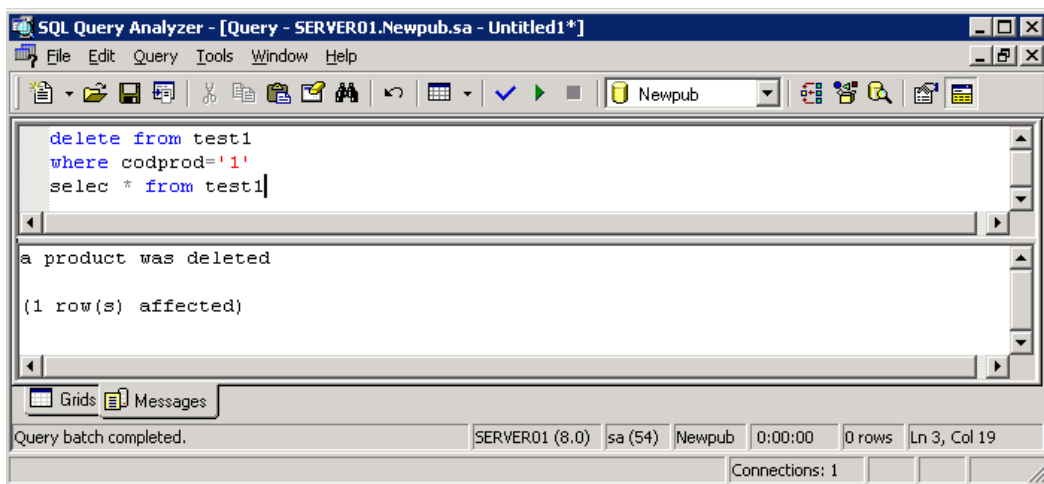
Membuat Trigger dengan Query Analyzer

Membuat trigger sama mudahnya dengan Query Analyzer. Pilihlah database tempat tabel berada dengan menggunakan field Database, ketikkan teks dari trigger, lalu tekan tombol Run.



Gambar 10.5 Create Trigger DELETE

Jika tidak ada error, sebuah pesan akan ditampilkan dan memberi tahu bahwa perintah sudah dilakukan dengan sukses. Jika ada syntax error, sebuah pesan error akan ditampilkan. Dalam hal ini, kembalilah ke panel atas dan perbaikilah perintah tersebut. Untuk menguji trigger, mari kita hapus record untuk prod01.



Gambar 10.6 Hasil Trigger DELETE

Perhatikan bahwa perintah SELECT hanya akan mengembalikan record yang tersisa di dalam tabel. Sisipkan kembali record untuk prod01 di dalam tabel. Kini mari kita buat trigger yang akan diaktifkan apabila perintah UPDATE dijalankan. Kode trigger tersebut adalah:

```

CREATE TRIGGER updprod ON test1
FOR UPDATE
AS
print "a product was updated"

```

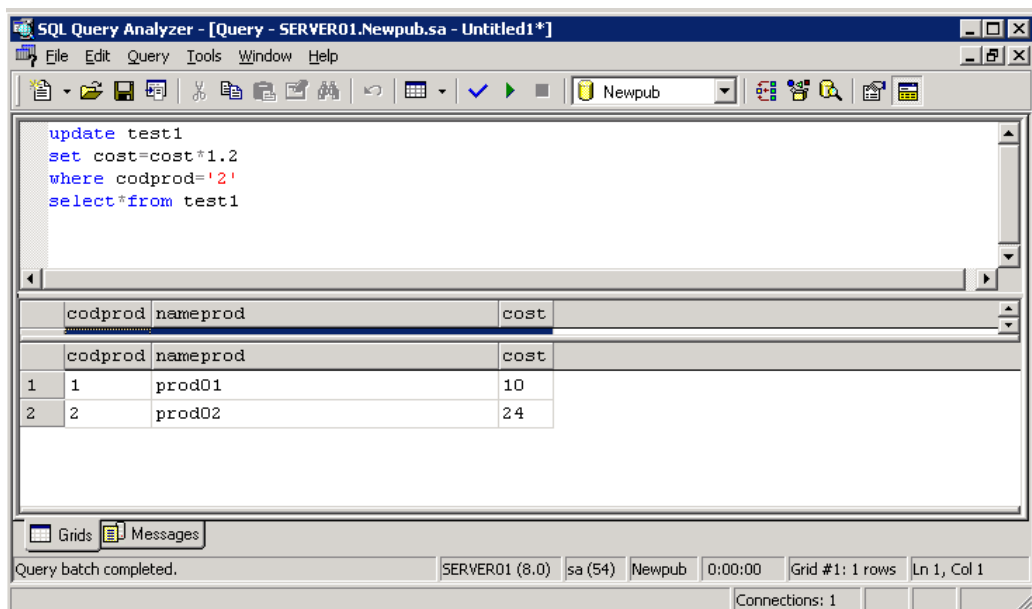
Kini jalankanlah kode berikut ini pada Query Analyzer untuk memperbesar nilai dari field Cost pada record codprod 2 sebanyak 20 persen.

```

update test1
set cost=cost*1.2
where codprod='2'
select*from test1

```

Dan lihatlah hasilnya:



Gambar 10.7 Hasil Trigger UPDATE

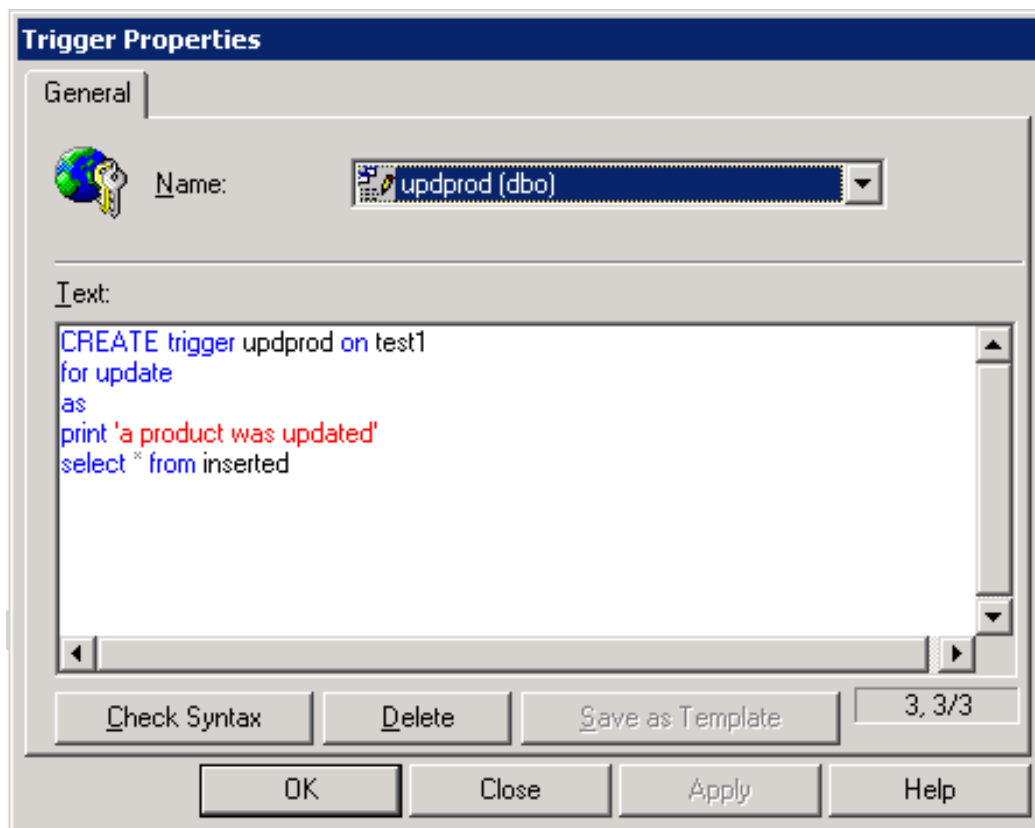
Tabel Inserted dan Deleted

Saat pemicu berjalan, SQL Server membuat dua tabel sementara yang hanya terlihat saat pemicu diaktifkan. Tab pertama disebut "Ditambahkan" dan yang kedua disebut "Dihapus". Saat perintah INSERT atau UPDATE dijalankan, record yang dibuat atau dimodifikasi akan disalin ke dalam tabel yang disisipkan. Saat perintah DELETE dijalankan, baris yang dihapus akan disalin ke tabel Dihapus.

Tabel-tabel ini sangat berguna apabila Anda ingin menambah, menghapus, atau mengubah baris-baris selama proses menjalankan trigger. Tabel ini sangat berguna untuk

meng-copy baris- baris dari satu tabel ke tabel lain.

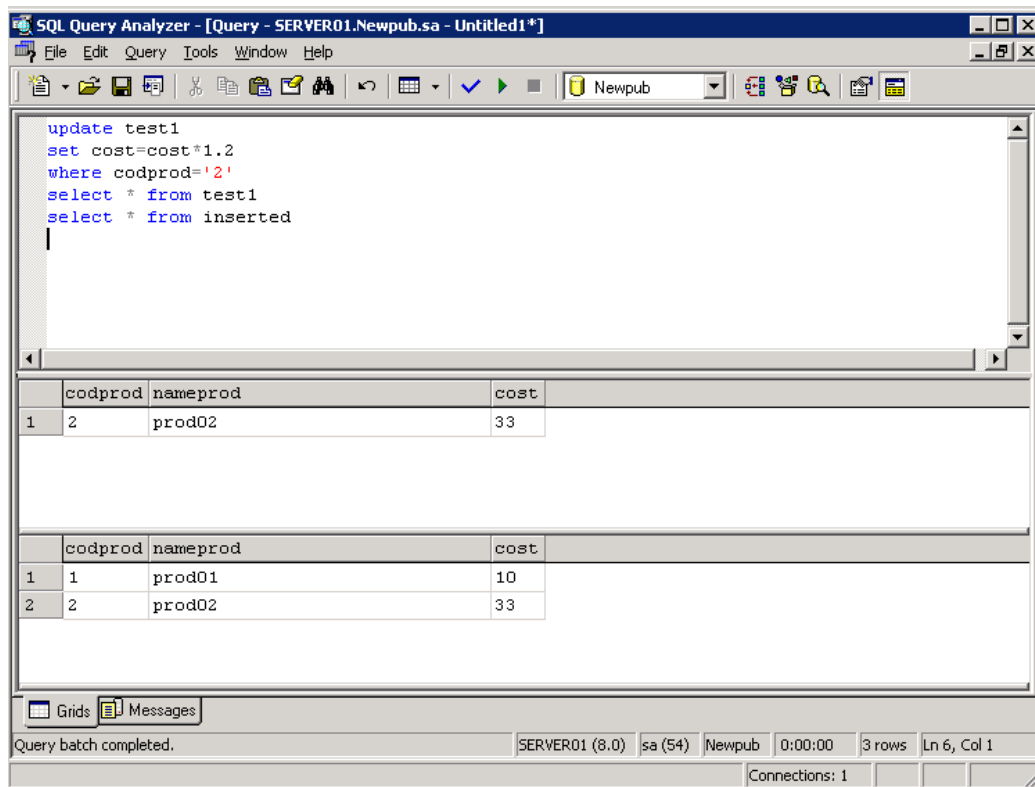
Untuk menampilkan keberadaan dari tabel ini, kita akan mengubah trigger updprod seperti ditunjukkan pada gambar berikut ini. Setelah mengaktifkan kotak dialog Trigger Properties, pilihlah nama dari trigger pada field Name dan sisipkanlah perintah SELECT di akhir dari teks trigger tersebut.



Gambar 10.8 Inserted

Tekan OK untuk menyimpan perubahan. Setelah menjalankah contoh berikut ini, ubahlah trigger updprod dan hapuslah baris ini.

Kode berikut ini meng-update record dengan prod02 dan menampilkan hasil melalui pemilihan baris-baris dari tabel Inserted, yang hanya akan menampilkan record yang diubah dan Test1. Trigger menampilkan pesan "A product was updated" dan isi dari tabel Inserted.



Gambar 10.9 Update Inserted

Tampilan ini dibuat oleh pernyataan SQL dari trigger updprod. Perhatikan bahwa error akan terjadi apabila kita mencoba menampilkan data dari tabel Inserted setelah trigger dijalankan, karena tabel Inserted sudah tidak ada lagi.

Mengubah Trigger

Trigger bisa diubah secara langsung dengan menggunakan perintah ALTER TRIGGER seperti ditunjukkan di bawah ini. Pilihan lain adalah dengan menghapus trigger dan membuatnya kembali. Operasi ini juga bisa dilakukan pada kotak dialog Trigger Properties.

Sintaks: ALTER TRIGGER trigger_name

Membuat Replikasi Baris dengan Trigge

Menggunakan tabel pemicu sementara, kita dapat membuat mekanisme replikasi cepat dari satu tabel ke tabel lainnya. Kita dapat menyisipkan, memodifikasi, atau menghapus record di tabel lain saat mereka melakukan operasi di tabel aktif.

Kami membuat tiga pemicu untuk mengulangi atau menggandakan operasi dari array test1 ke array test2. Pemicu pertama mengulangi penyisipan catatan. Nama itu benar. Kode

untuk pemicu ini ditampilkan di bawah. Gunakan salah satu metode di atas untuk membuat pemicu.

```
CREATE TRIGGER [insrepl] ON test1
FOR insert AS
insert into test2
select codprod, nameprod, cost from inserted
```

Dalam pemicu ini, kami menggunakan perintah INSERT INTO dan SELECT untuk mendapatkan bidang dari tabel yang disisipkan. Pemicu berikutnya mengulangi proses penghapusan. Kami menggunakan klausa DELETE FROM dan klausa WHERE untuk memfilter catatan yang akan dihapus dengan mencari tabel Dihapus, yang dalam hal ini hanya berisi catatan yang baru saja dihapus.

```
CREATE TRIGGER [delrepl] ON test1
FOR DELETE
as
delete from test2
where test2.codprod in (select codprod from deleted)
```

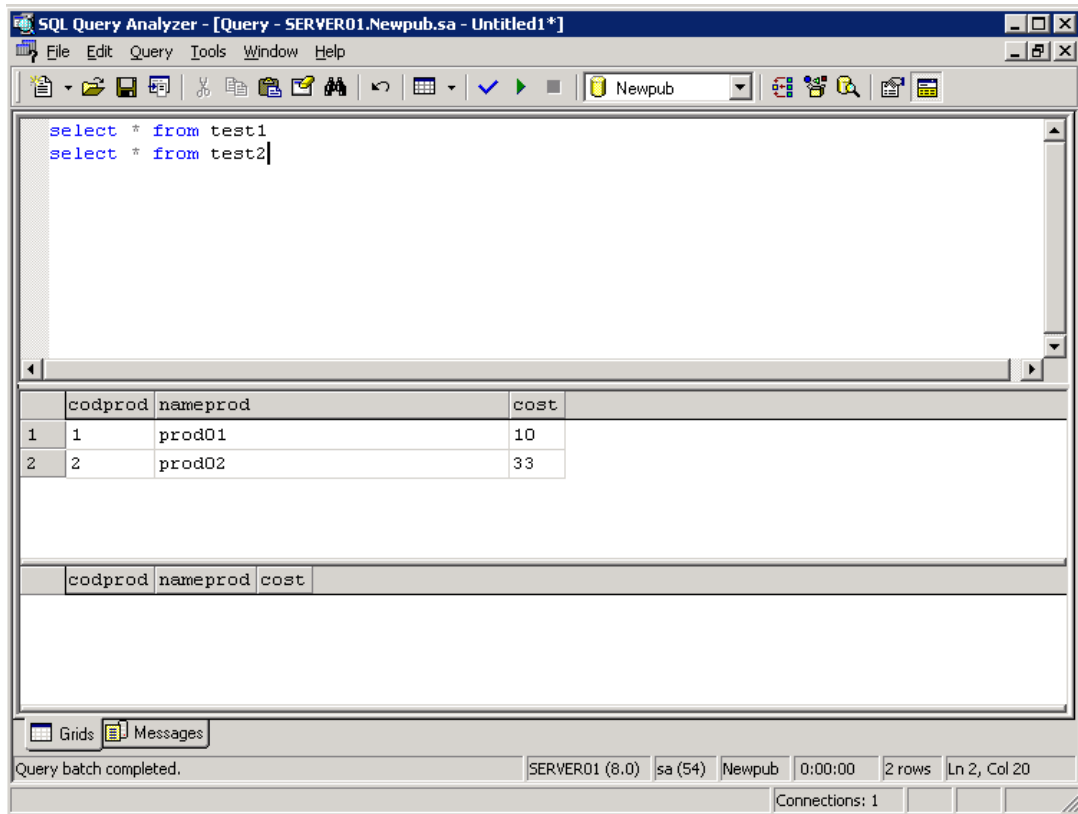
Pemicu berikut mengulangi proses pembaruan yang dilakukan di Test1. Di dalamnya kita meng-update kolom Cost dari Test2 dengan isi dari kolom Cost dari tabel Inserted (ingat bahwa INSERT dan UPDATE menggunakan tabel Inserted yang sama). Klausa WHERE juga digunakan untuk memperbarui catatan yang kodenya hanya ditemukan di tabel yang disisipkan.

```
CREATE TRIGGER [updrepl] ON test1
FOR update
AS
update test2
set cost=(select cost from inserted)
where test2.codprod in (select codprod from inserted)
```

Setelah membuat trigger, ini saatnya untuk mengujinya. Pertama-tama, mari kita lihat bagaimana kedua tabel menjalankan perintah SELECT ini:

Select * from test1
Select * from test2

Ini adalah hasil yang didapatkan pada Query Analyzer:

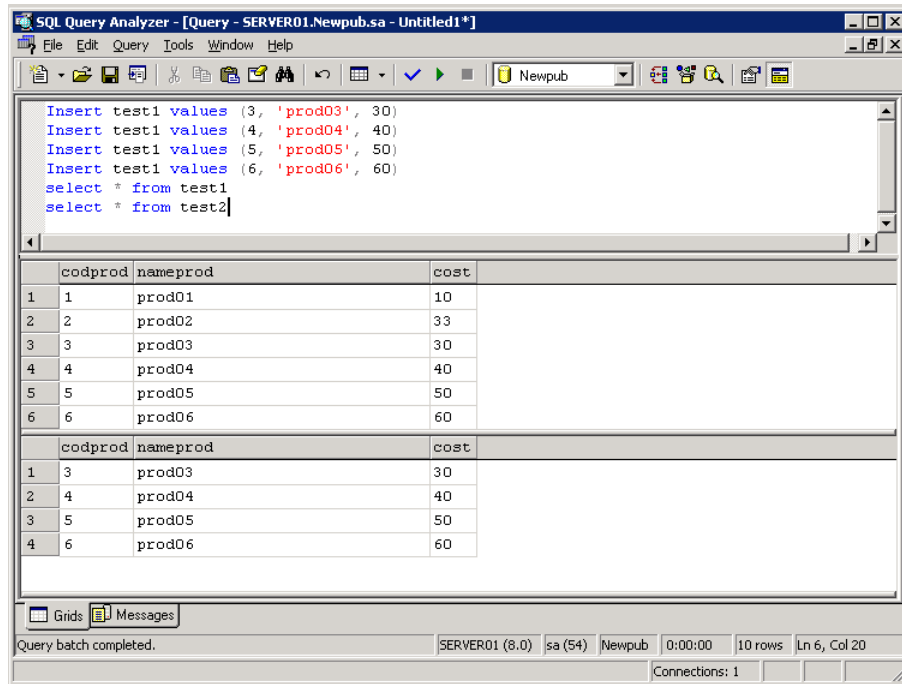


Gambar 10.10 Melihat isi table test1 dan test2

Kini kita akan menambahkan kode berikut ini ke dalam tabel Test1 dan melihat bagaimana ia mempengaruhi kedua tabel:

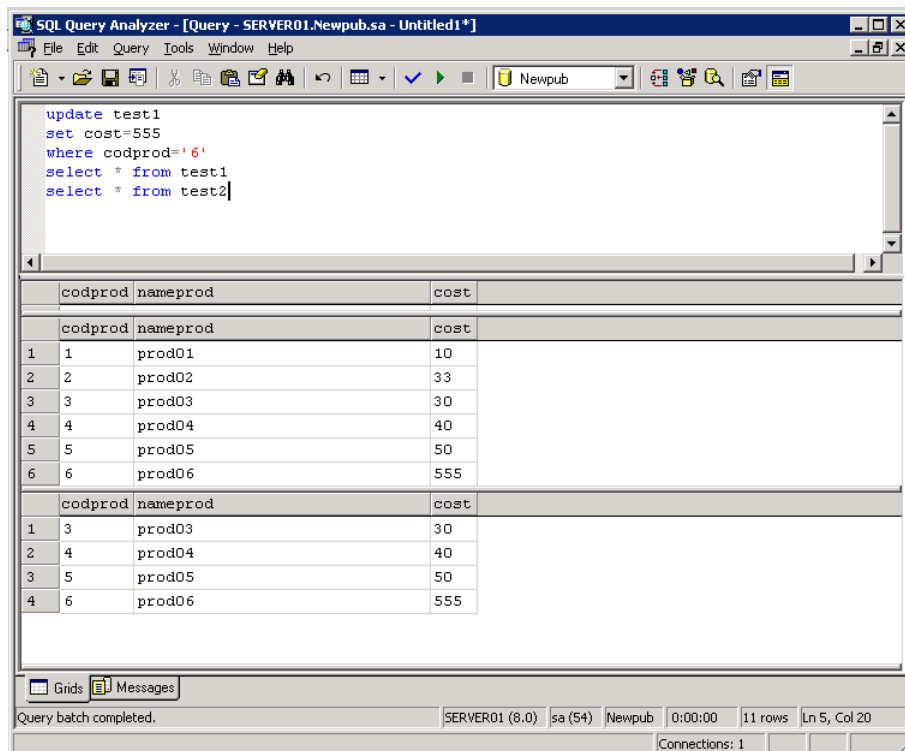
Insert test1 values (3, 'prod03', 30)
Insert test1 values (4, 'prod04', 40)
Insert test1 values (5, 'prod05', 50)
Insert test1 values (6, 'prod06', 60)

Ini adalah hasil yang didapatkan pada Query Analyzer:



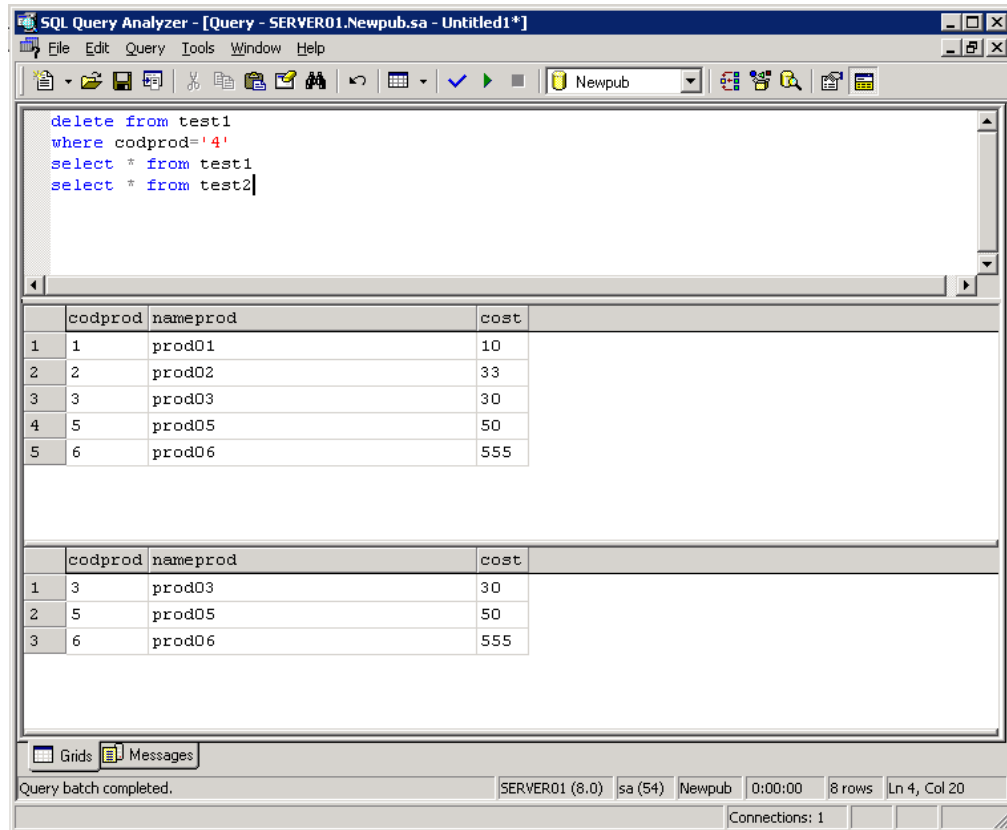
Gambar 10.11 Hasil input table test1 dan hasil test2

Perhatikan bahwa Test2 memiliki empat baris yang sama seperti yang disisipkan ke dalam Test1. Kini mari kita uji replikasi perubahan. Ubahlah nilai dari field Cost untuk prod6 dari 60 menjadi 555, dan jalankan kode yang ditampilkan pada panel atas dari gambar di bawah ini. Perhatikan bahwa proses replikasi telah berjalan dengan benar.



Gambar 10.12 Hasil update table test1 dan hasil test2

Uji yang terakhir mereplikasi penghapusan baris. Kita akan menghapus produk dengan codprod4.



Gambar 10.13 Hasil delete table test1 dan hasil test2

Sekali lagi proses replikasi berjalan dengan sukses.

Melihat Trigger dari Tabel

Melalui prosedur `sp_helptrigger`, Anda bisa menemukan tipe-tipe trigger yang didefinisikan pada tabel di dalam database yang sedang aktif.

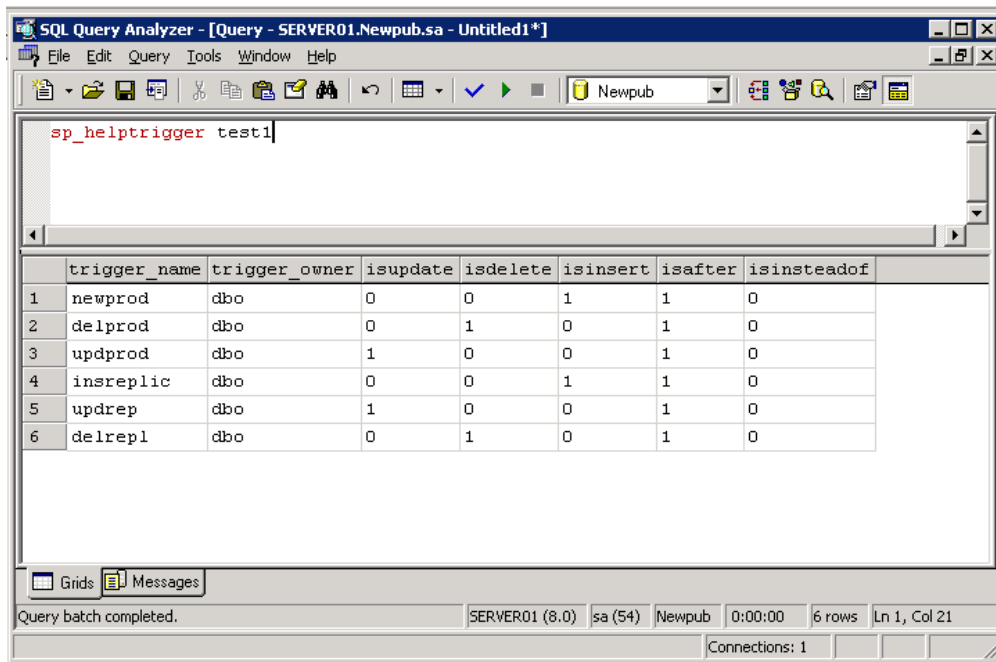
Sintaks: `sp_helptrigger [@tablename=] 'table' [,@triggertype=] 'type'`

`[@tablename =] 'table'` adalah nama dari tabel di mana trigger dibuat.

`[@triggertype =] 'type'` adalah tipe dari trigger di mana Anda ingin informasi lebih banyak.

Tipe yang dibolehkan sebagai parameternya adalah DELETE, INSERT, dan UPDATE.

Contoh: `sp_helptrigger test1`

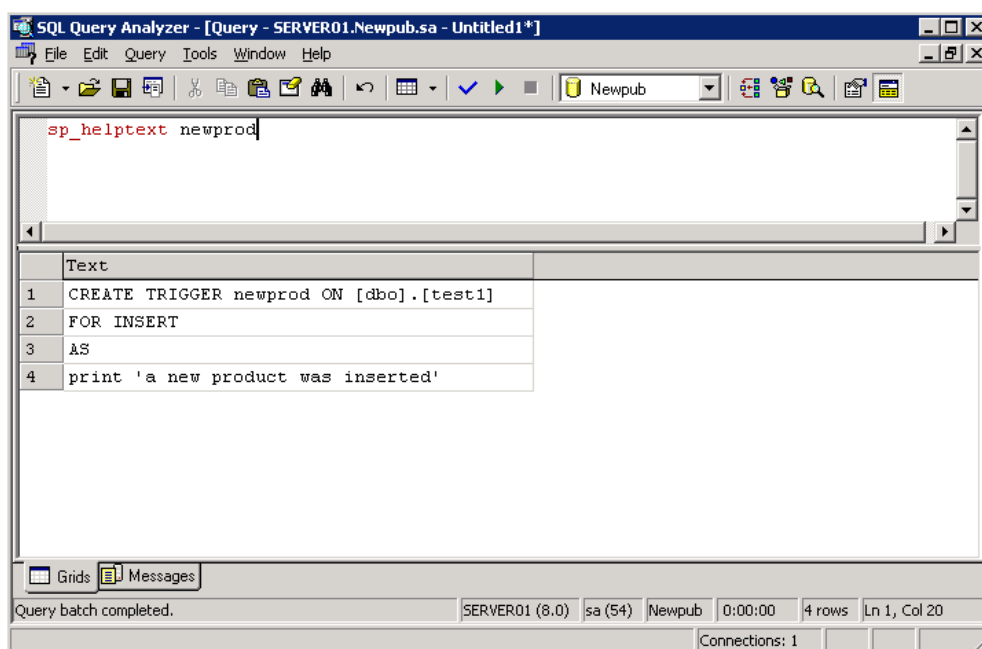


Gambar 10.14 sp_helptrigger test1

Untuk melihat teks dari trigger, gunakanlah prosedur sp_helptext. Prosedur ini menghasilkan teks dari objek yang ditentukan. Objek ini bisa berupa trigger, prosedur, view, aturan, atau default. Sintaks: sp_helptext [@ objname='name'

[@objname =) 'name' adalah nama dari objek di mana Anda ingin informasi lebih banyak.

Perhatikan kedua prosedur yang diterapkan kepada tabel Test1 dan salah satu dari trigger.



Gambar 10.14 sp_helptext newprod

Menghapus Trigger

Untuk menghapus trigger, pertama-tama perbesarlah folder Databases, perbesarlah database di mana terdapat tabel yang mengandung trigger, lalu klik Tables. Pada panel detail, klik kanan tabel yang mengandung trigger, tunjuk ke Tasks, lalu klik Manage Triggers. Pada daftar Name, pilihlah nama trigger yang ingin dihapus, lalu pilih Delete. Anda juga bisa menggunakan perintah SQL DROP TRIGGER:

Sintaks: DROP TRIGGER trigger_name

Perintah ini akan menghapus struktur trigger dari database dan menghapus hak-hak yang diberikan kepada user lain. Jika tabel yang mengandung trigger dihapus, trigger juga akan dihapus dari dalam database.

Anda juga bisa menggunakan kotak dialog Trigger Properties untuk menghapus trigger. Pilihlah nama trigger pada kotak Name, tekan tombol Delete, lalu pastikan penghapusan.

Kesimpulan

Bab ini membahas beberapa fungsi yang dapat diberikan oleh trigger untuk para pengembang program. Menjelajahi kemampuan penuh dari trigger adalah salah satu tantangan terbesar di dalam mempelajari SQL Server. Pada bab berikutnya, kita akan membahas mengenai sarana replikasi yang terdapat pada SQL Server.

BAB 11

Replikasi

Replikasi adalah teknik yang sangat penting dalam lingkungan database perusahaan. Replikasi memungkinkan Anda berbagi database dan proses di seluruh perusahaan. Informasi basis data dapat digandakan dan disalin ke beberapa lokasi dalam organisasi. Basis data terdistribusi ini dapat disinkronkan agar selalu memiliki nilai yang sama, mis. B. Daftar harga disinkronkan antara semua cabang.

Model Replikasi

SQL Server mengadopsi model replikasi berdasarkan pada model "publish and subscribe" yang diperkenalkan pada versi 6.0. Penerbit, distributor, pelanggan, artikel, dan publikasi merancang model ini, selain tipe push and pull. Model ini mengizinkan replikasi pada database heterogen melalui penggunaan ODBC atau OLEDB.

Jenis-Jenis Replikasi

SQL Server mendukung tiga jenis replikasi, yaitu snapshot, transactional, dan merge.

Snapshot

Replikasi snapshot membuat "snapshot" dari database yang direplikasi untuk kliennya, yang menerima salinan lengkap data, bukan hanya perubahannya. Replikasi ini sempurna ketika database tidak terus berubah dan untuk pengguna yang tidak ingin mengupdate tabel.

Transactional

Replikasi transaksional memungkinkan tabel dan prosedur untuk direplikasi. Replikasi ini memungkinkan pemfilteran data yang dipublikasikan. Rilis ini menggunakan log untuk mencatat perubahan pada item (seperti tabel) sejak rilis terakhir dan memeriksa perintah INSERT, UPDATE, dan DELETE. Pesanan disimpan dalam database distribusi dan kemudian dikirim ke pelanggan.

Merge

Ketik replikasi peta melacak perubahan ke basis data sumber dan menyinkronkan nilai antara editor (penerbit) dan pelanggan. Perubahan pada database target juga mengubah database sumber dan sebaliknya.

Untuk menerbitkan beberapa data, Anda harus membuat publikasi dengan memilih tabel dan prosedur serta ketersediaannya untuk pelanggan. Objek dalam publikasi, seperti tabel atau baris, disebut artikel. Prosedur dapat dilihat sebagai artikel reproduksi dari semua jenis asosiasi.

Berlangganan Publikasi

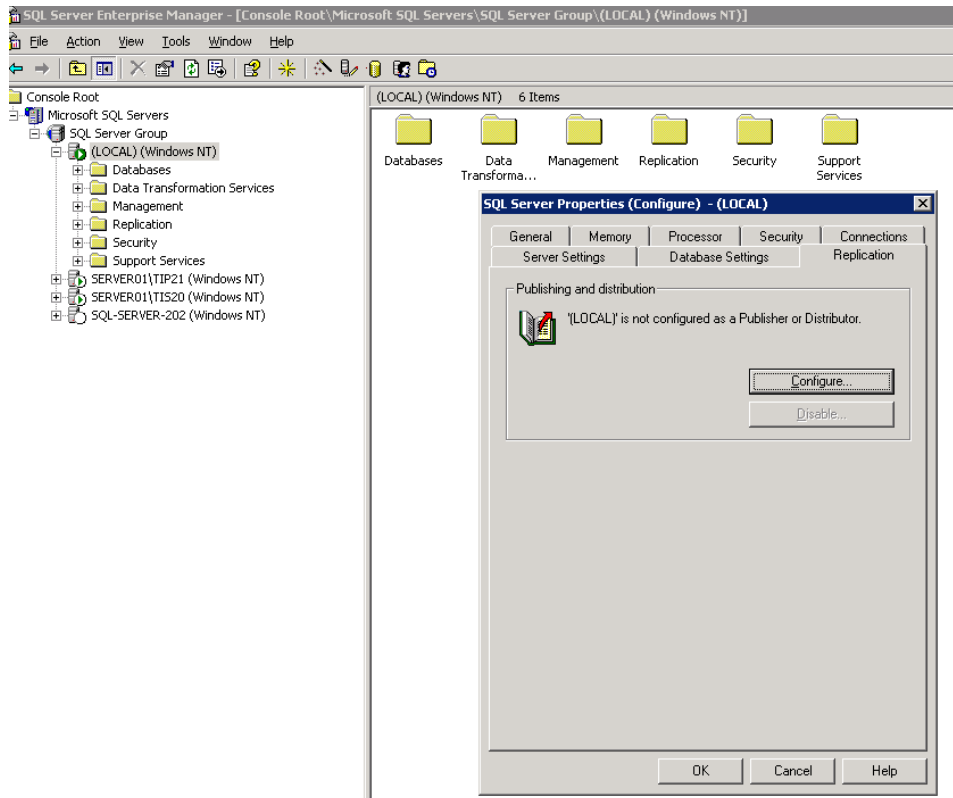
Jika Anda ingin menerima publikasi, Anda harus berlangganan. Anda harus menentukan database untuk menerima rilis ini. Ada dua jenis edisi:

Pull dan Push. Langganan tipe push diterapkan saat penerbit (editor) dikelola secara terpusat. Dalam hal ini, contoh salinan publikasi akan dikirimkan atau diteruskan ke pelanggan. Langganan tipe tarik diimplementasikan saat pelanggan dikelola secara terdesentralisasi. Orang ini mendapat salinan publikasi dari penerbit. Setelah tabel publikasi dan tabel target ditentukan, keduanya harus disinkronkan agar proses kontrol publikasi berfungsi dengan benar.

Bagian berikut ini menjelaskan pembuatan replikasi transactional secara langkah demi langkah. Kita akan membuat replikasi dari database Northwind.

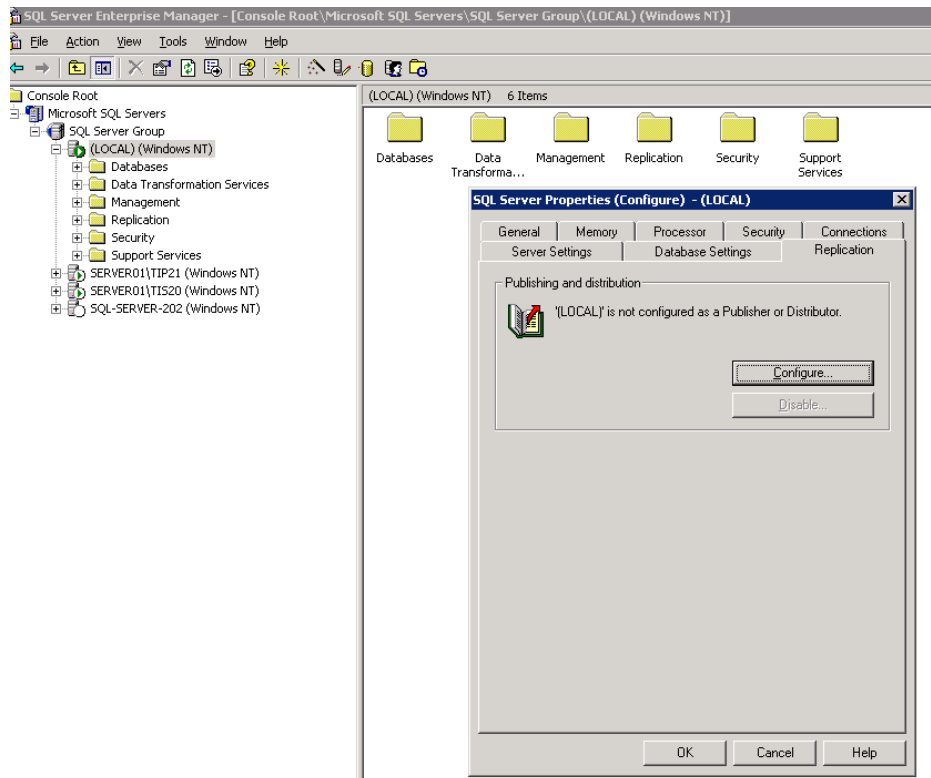
Contoh Replikasi Transactional

Bukalah Enterprise Manager dan klik ganda pada nama server. Pilihlah Replicate Data pada panel kanan.



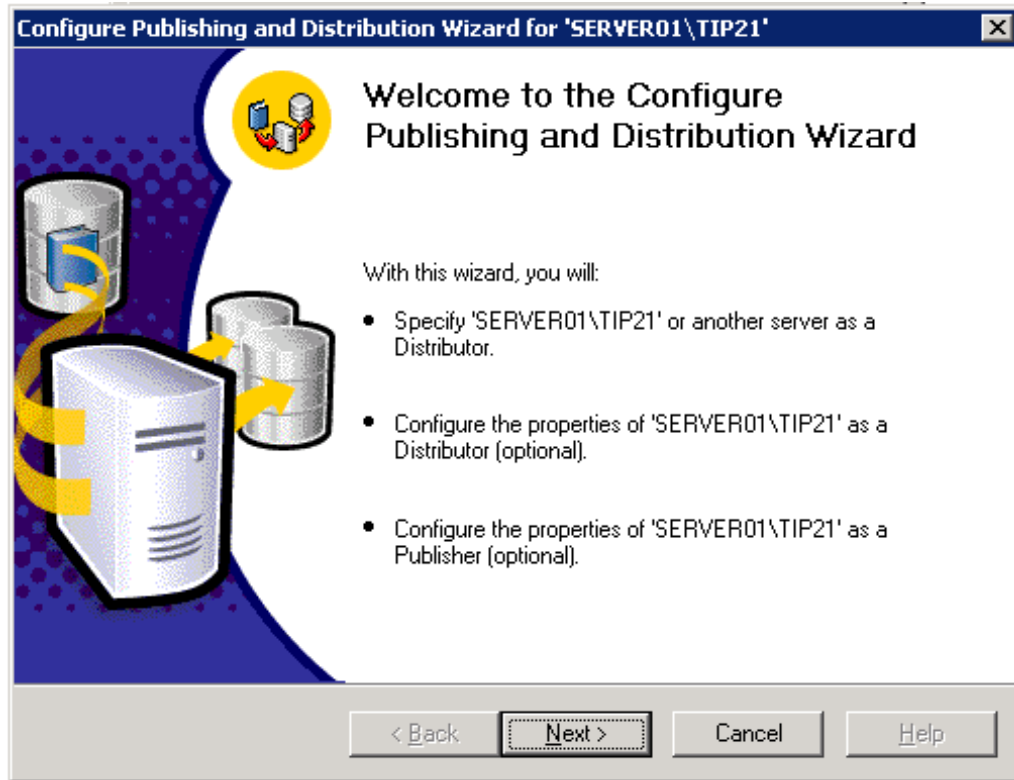
Gambar 11.1 Replication

Pada layar berikutnya, pilihlah Configure Replication.



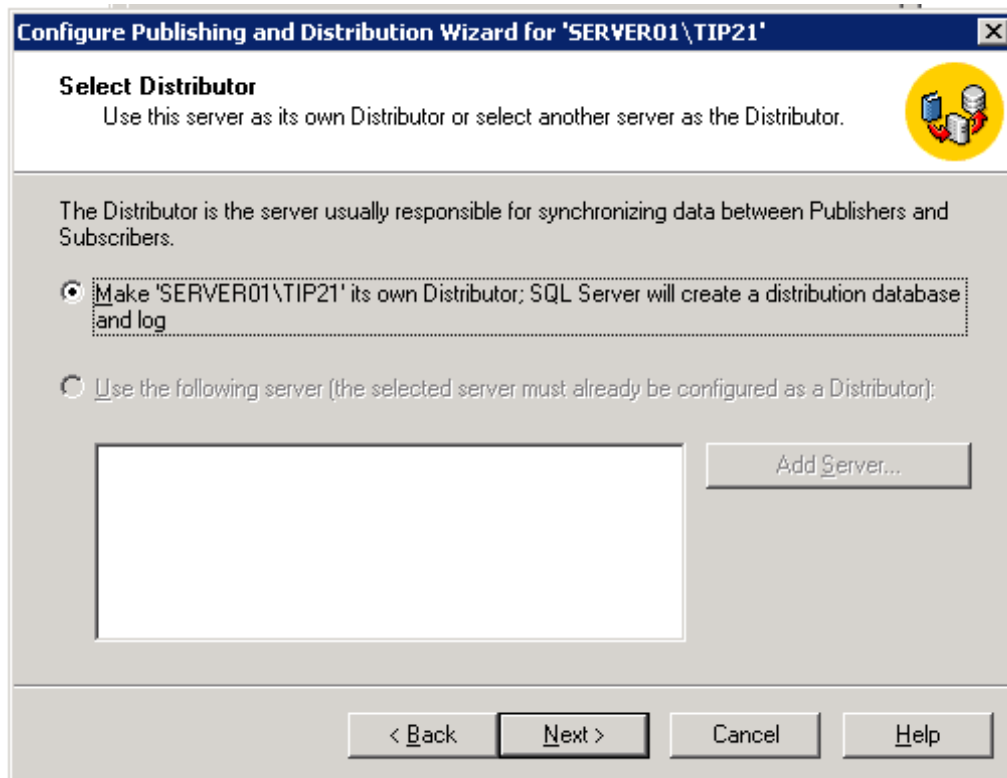
Gambar 11.2 Configure Replication

Ini akan mengaktifkan panduan persiapan Penerbitan dan Distribusi, yang akan memandu Anda melalui langkah-langkah yang diperlukan.



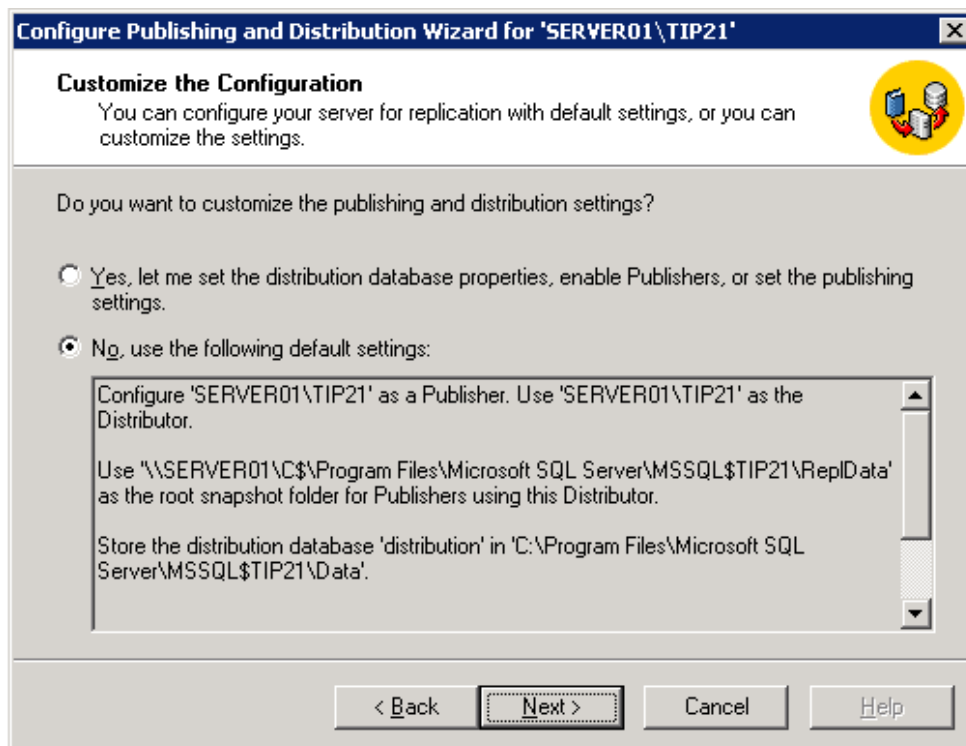
Gambar 11.3 Configure Publishing

Klik Next. Lalu pilih database yang didistribusikan, yang dalam contoh ini akan tetap berada pada server lokal.



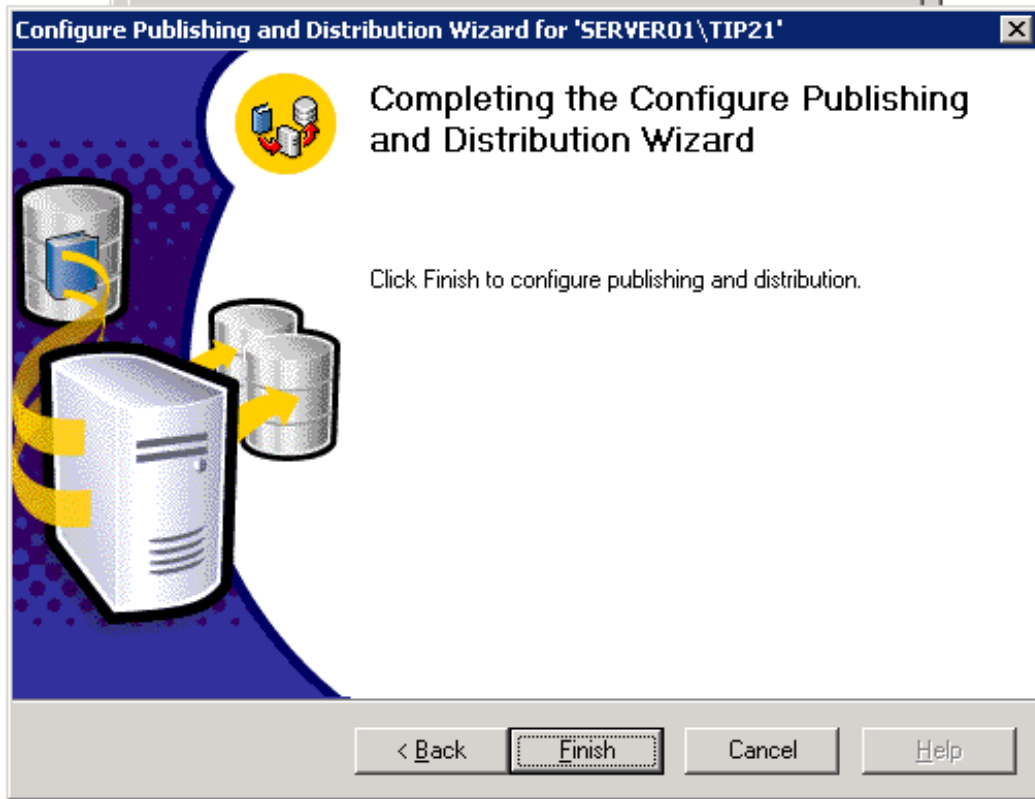
Gambar 11.4 Configure Publishing Default

Klik Next. Akan muncul layar dengan konfigurasi. Kita akan menggunakan setting default yang telah disediakan.



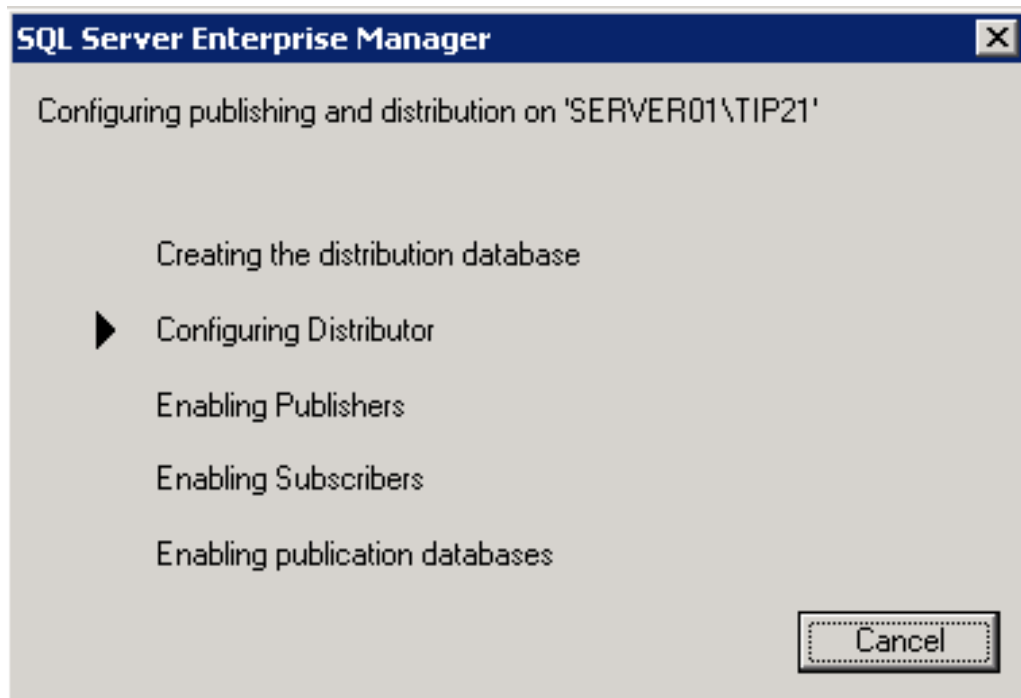
Gambar 11.5 Customize the Configure

Terimalah nilai-nilai default lalu klik Next. Wizard akan melengkapi tugasnya dan mengizinkan Anda menentukan database yang akan dipublikasikan dan didistribusikan. Tekan tombol Finish untuk memulai proses konfigurasi.



Gambar 11.6 Completing the Configure Publishing

Sebuah kotak dialog akan muncul dan menampilkan kemajuan dari proses, seperti pada gambar 11.7.

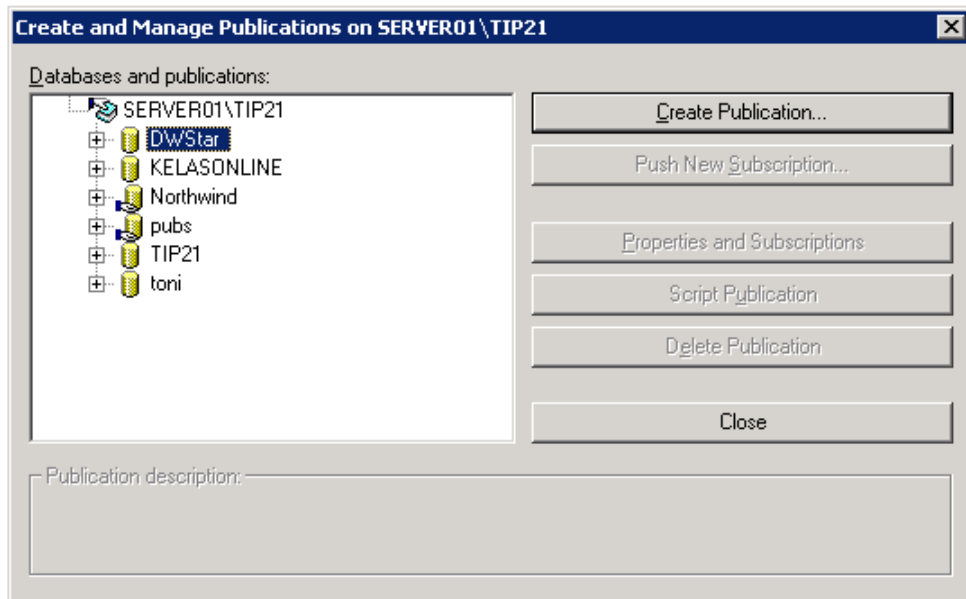


Gambar 11.7 Configure Publishing

Sebuah pesan yang memastikan eksekusi tugas akan muncul pada bagian akhir. Tekan tombol OK untuk melanjutkan. Jika SQL Server Agent tidak dikonfigurasi untuk berjalan apabila komputer dinyalakan, kotak dialog akan meminta Anda mengkonfigurasikannya. Apabila sudah selesai, tekan tombol Close .

Mendefinisikan Publikasi

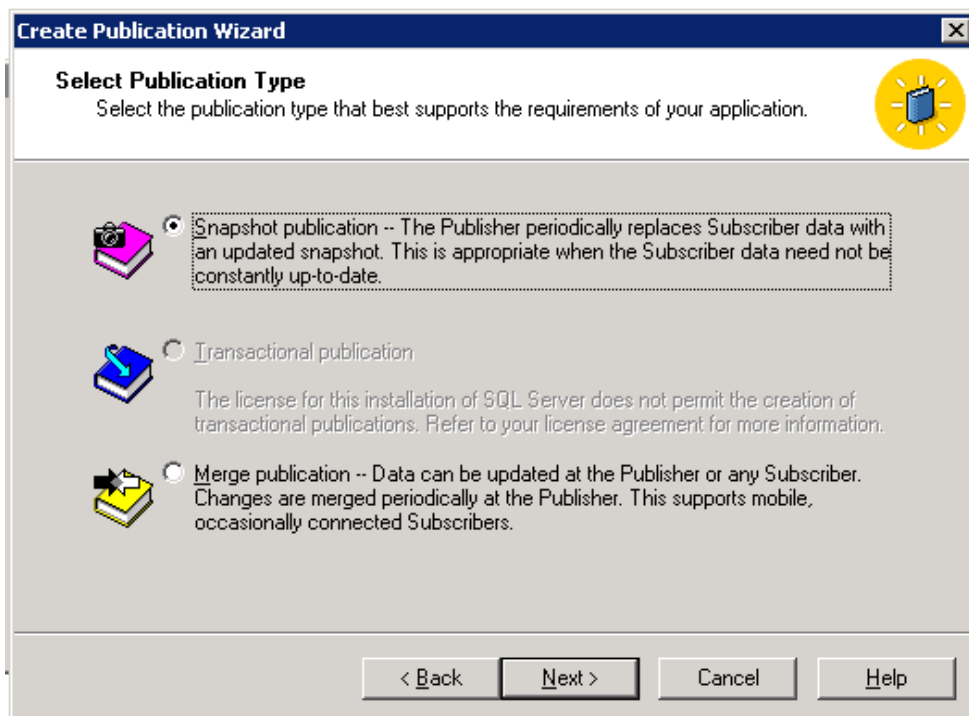
Klik pada item Create or Manage Publication pada panel kanan. Kotak dialog Create and Manage Publication akan muncul.



Gambar 11.8 Mendefinisikan Publikasi

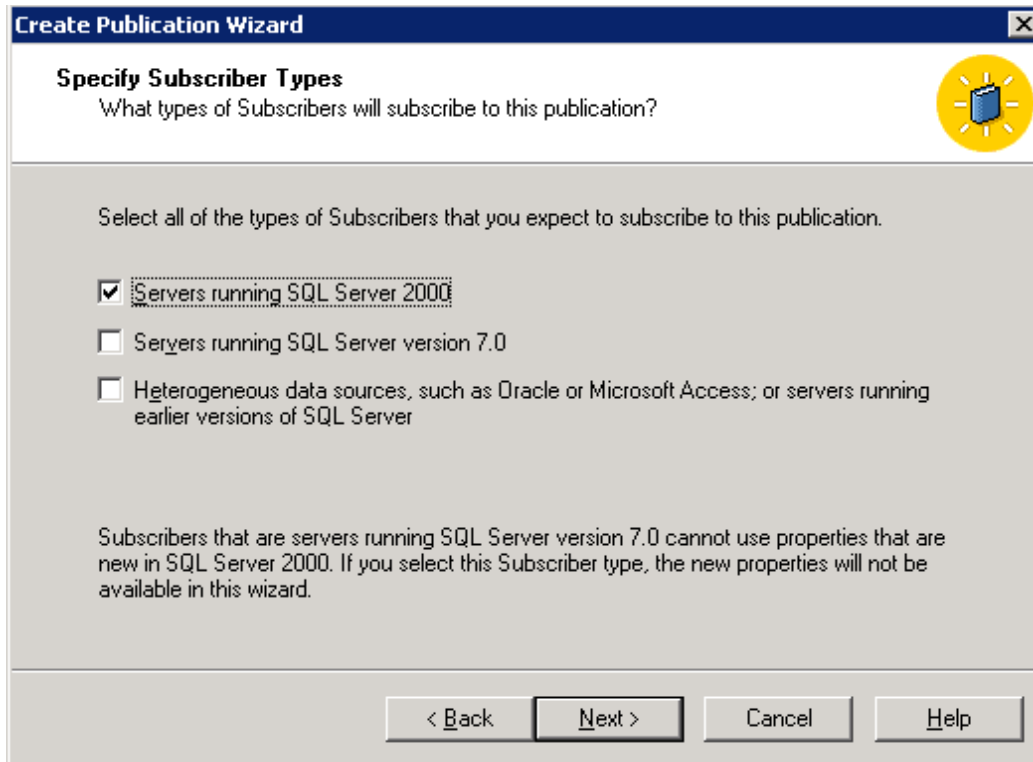
Pilihlah database Northwind lalu klik pada tombol Create Publication. Create Publication Wizard akan dijalankan.

Tekan tombol Next pada layar pertama. Pada langkah kedua, Anda bisa memilih jenis publikasi yang diinginkan dari pilihan yang ada. Pilihlah Transactional Publication lalu tekan Next.



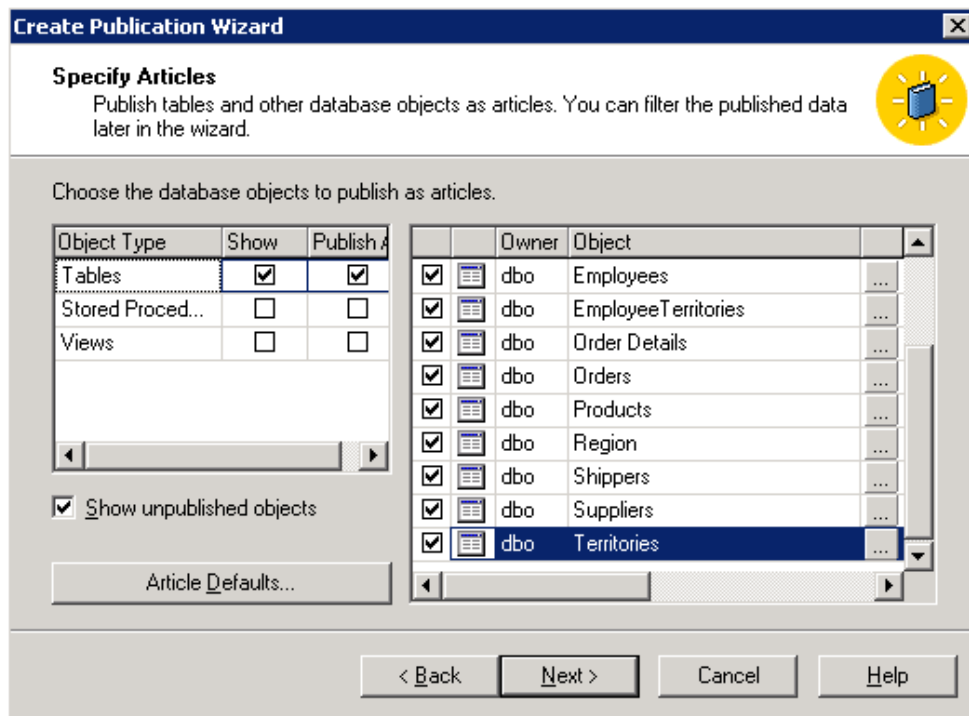
Gambar 11.9 Memilih Type Publikasi

Anda akan ditanyakan mengenai langganan yang di-update secara langsung. Terimalah nilai default untuk tidak mengaktifkan langganan yang di-update secara langsung, lalu tekan Next.



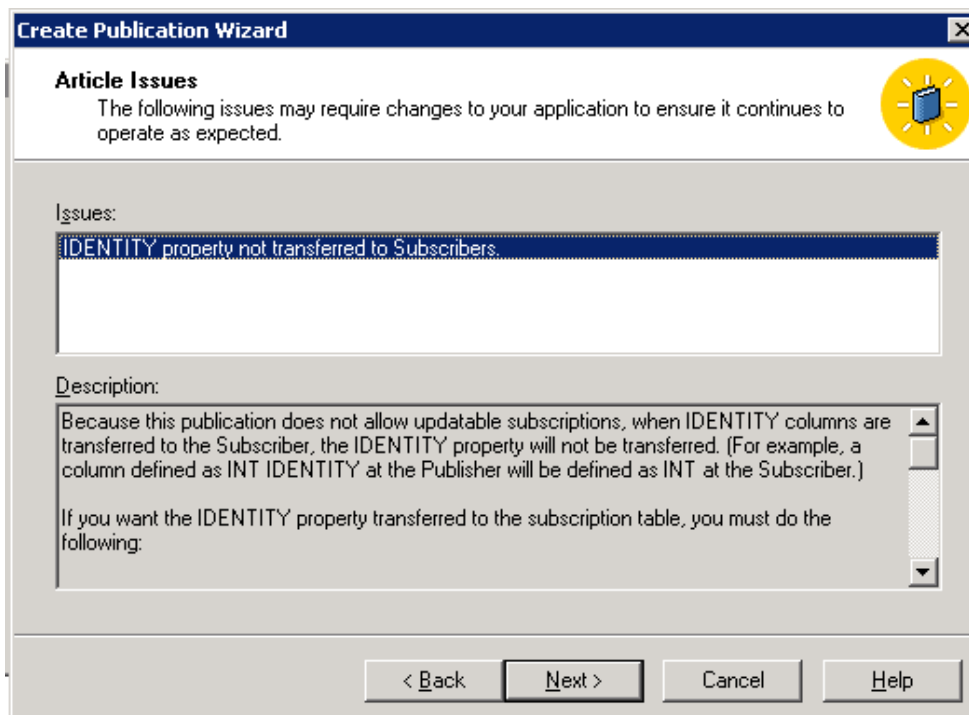
Gambar 11.10 Create Publication Wizard

SQL Server kemudian akan meminta Anda menentukan pelanggan yang diizinkan. Anda bisa menentukan pelanggan yang tidak menjalankan SQL. Dalam hal ini, pilihlah All Subscribers lalu tekan Next .



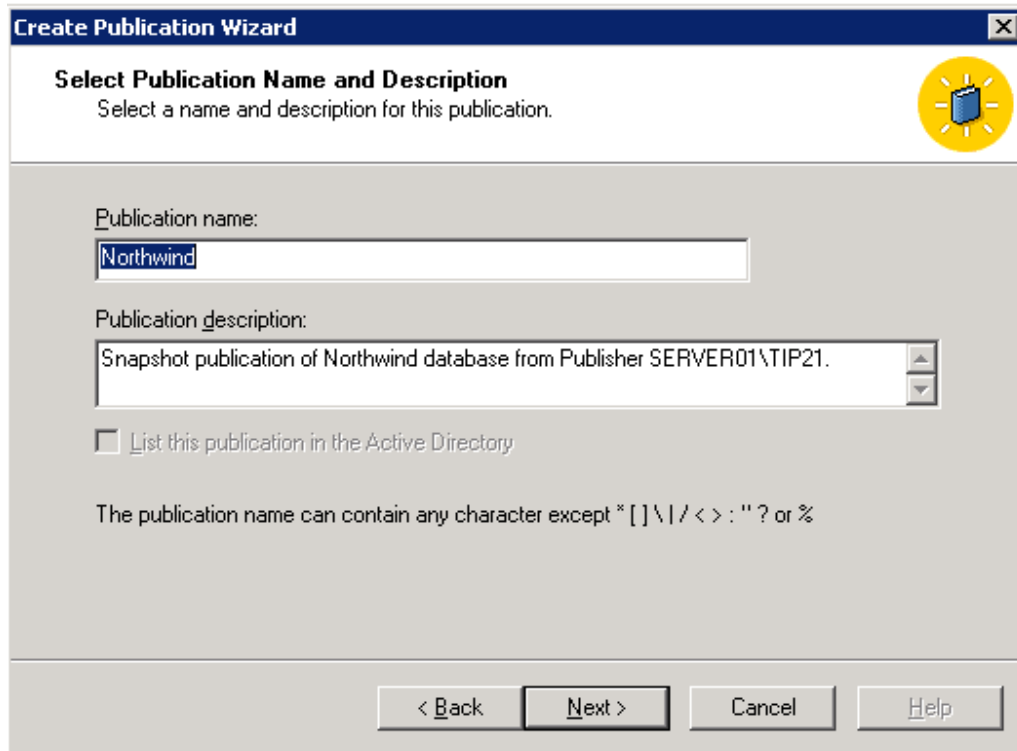
Gambar 11.11 All Subscribers

Kini Anda perlu memilih artikel atau objek yang akan dipublikasikan. Semua tabel yang tersedia akan ditam pilkan. Pada contoh ini, kita akan memilih semuanya. Oleh karena itu, pilihlah tombol Publish All, lalu tekan Next.



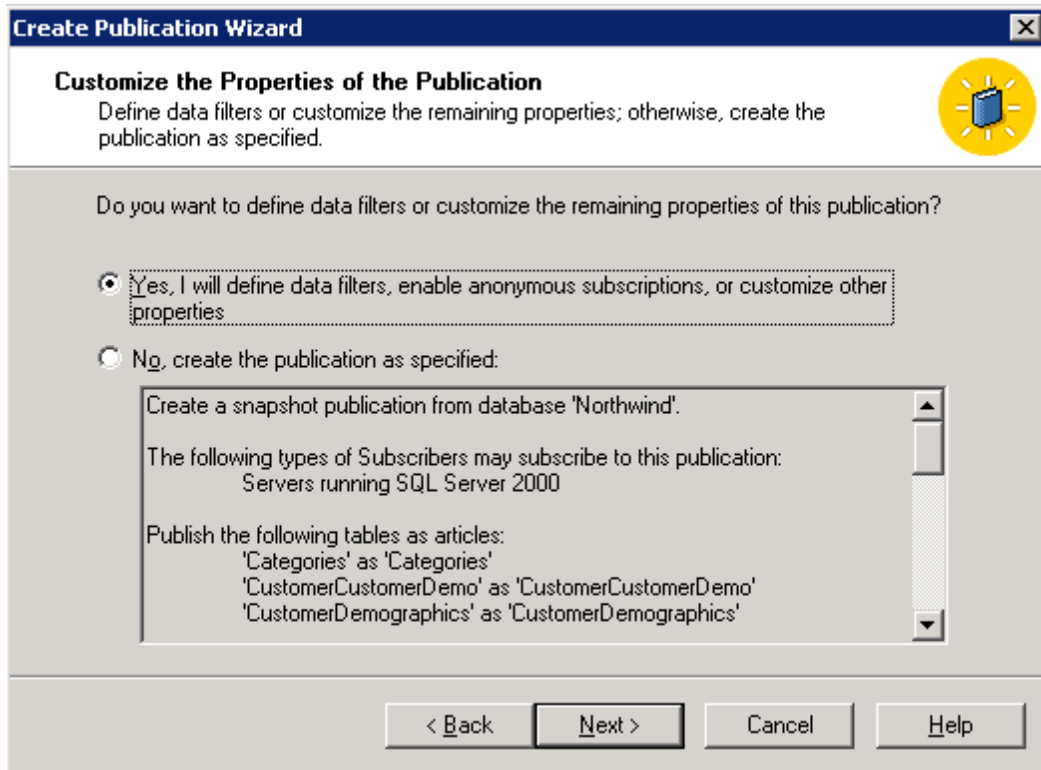
Gambar 11.12 Article Issue

Pada langkah berikutnya, ketikkanlah nama untuk replikasi. Tambahkan `_Tran` pada nama Northwind, lalu tekan Next.



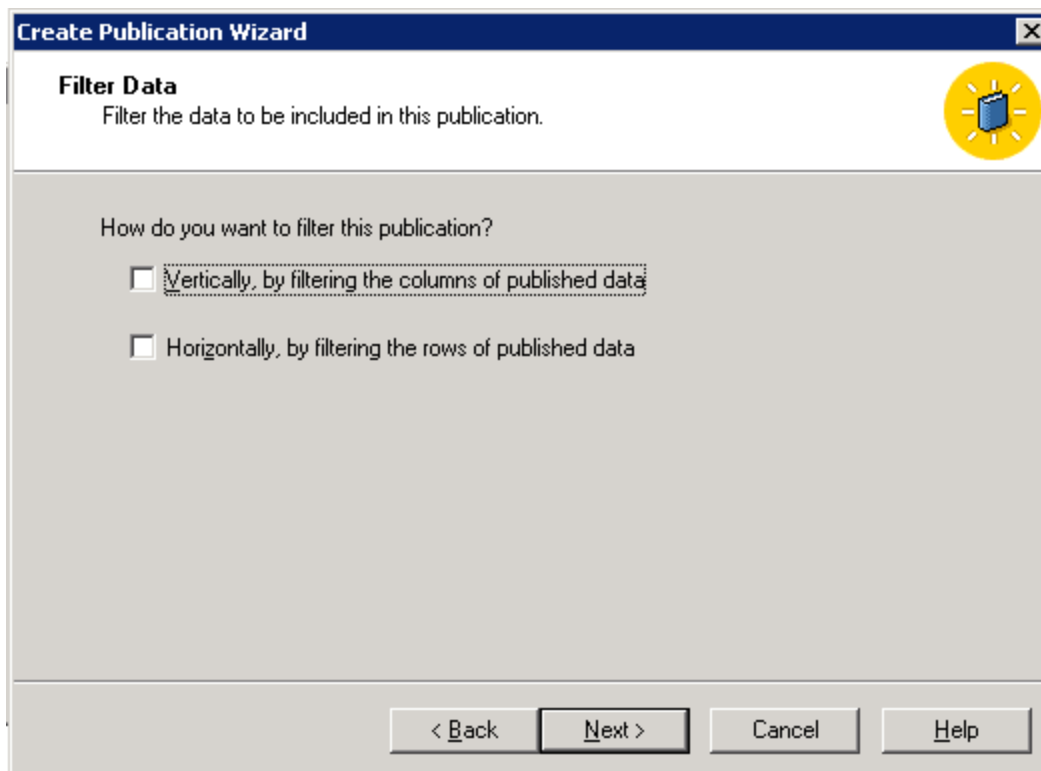
Gambar 11.13 Publication Name

Pada langkah berikutnya, Anda bisa mengubah replikasi mempublikasikannya tanpa filter. Klik Yes lalu tekan tombol Next.



Gambar 11.14 Mengubah Replikasi

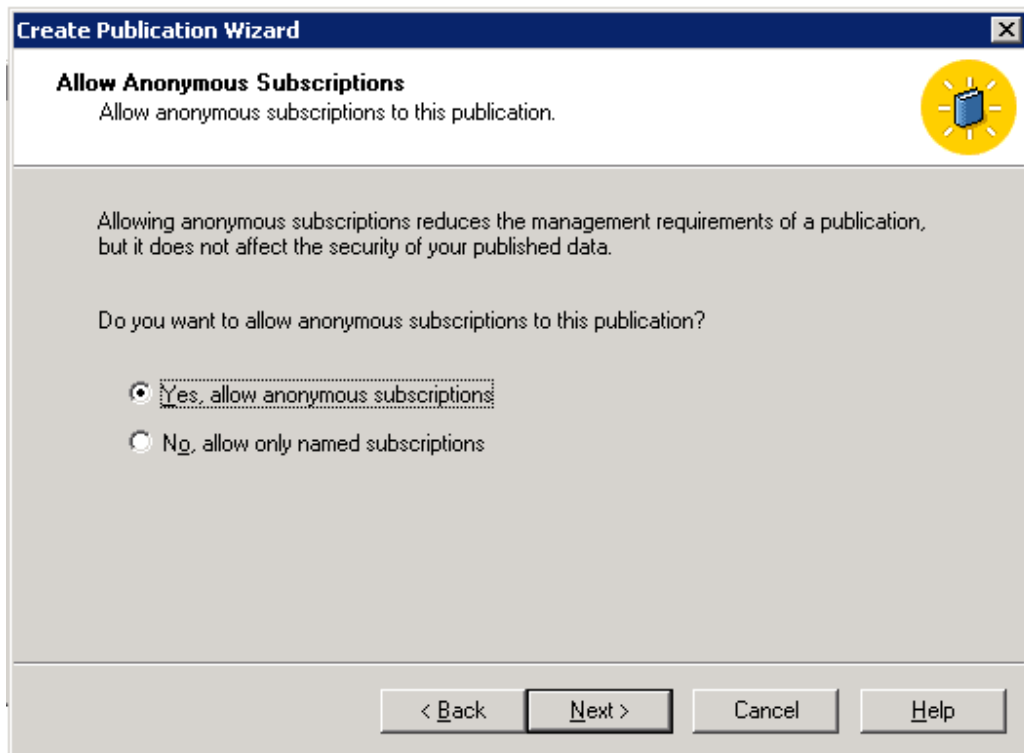
Pada kotak dialog berikutnya, pilihlah No, lalu tekan tombol Next.



Gambar 11.15 Filter Data

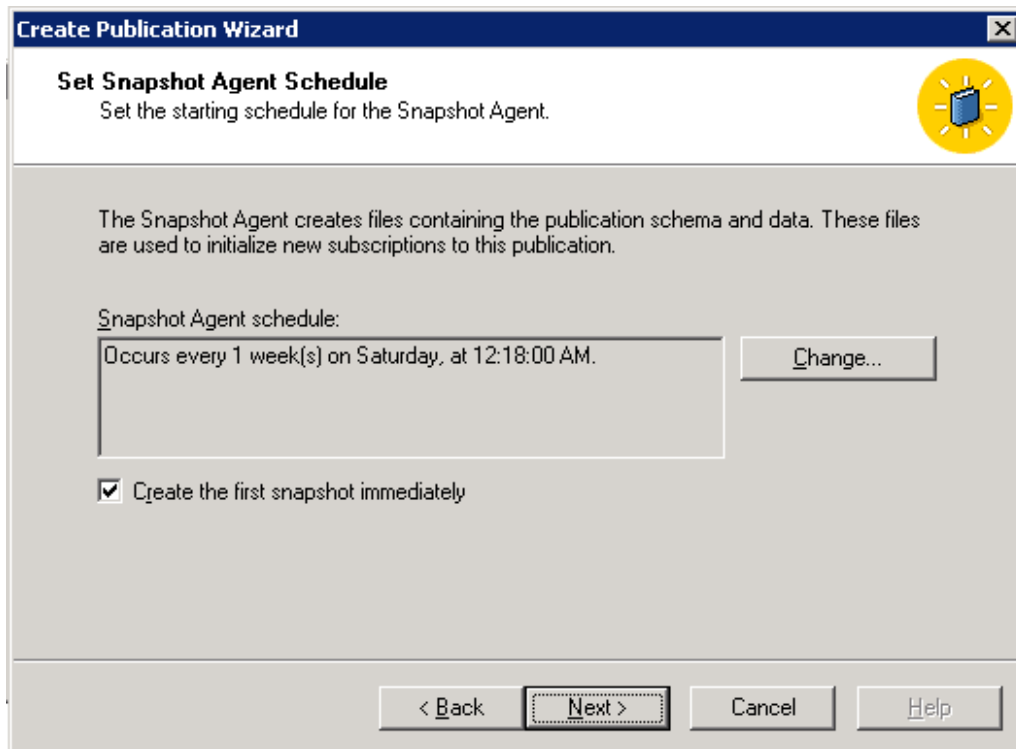
Pilihlah Yes atau No, tergantung pada apakah Anda ingin mempublikasikan semua data atau seluruh tabel. Klik Next.

Berikutnya, SQL akan menanyakan apakah Anda akan mengizinkan pelanggan anonymous. Pilihlah Yes, lalu klik Next.



Gambar 11.16 Akses

Layar berikutnya akan mengkonfigurasi Snapshot Agent. Tentukan bahwa Anda ingin snapshot langsung, lalu klik pada tombol Next.



Gambar 11.17 Schedule

Terimalah nilai default yang disarankan.



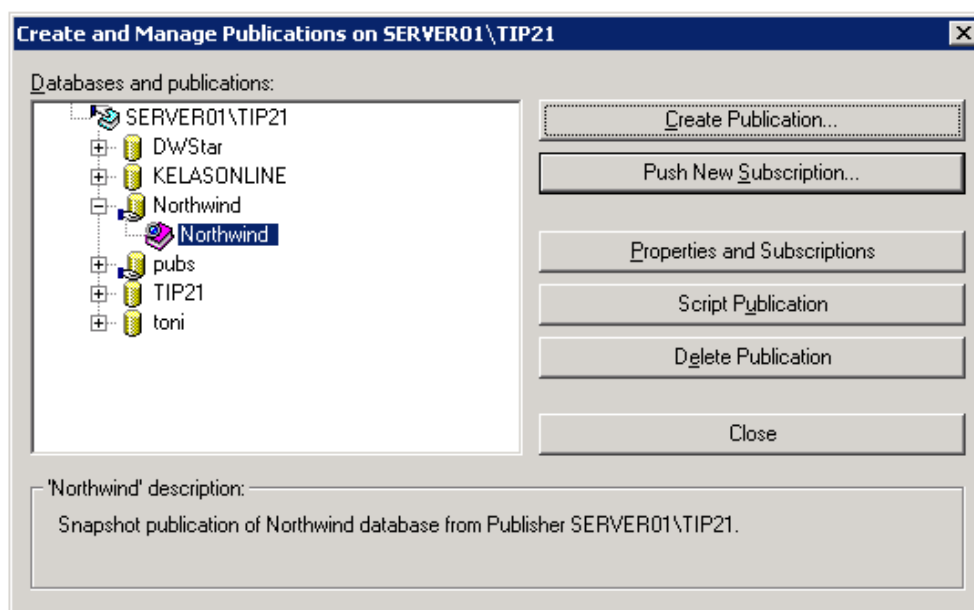
Gambar 11.18 Completing the Create Publication Wizard

Tekan Finish. Layar wizard terakhir akan menampilkan rangkuman dari yang akan dilakukan. Klik OK untuk menutupnya.

Mengirimkan Publikasi

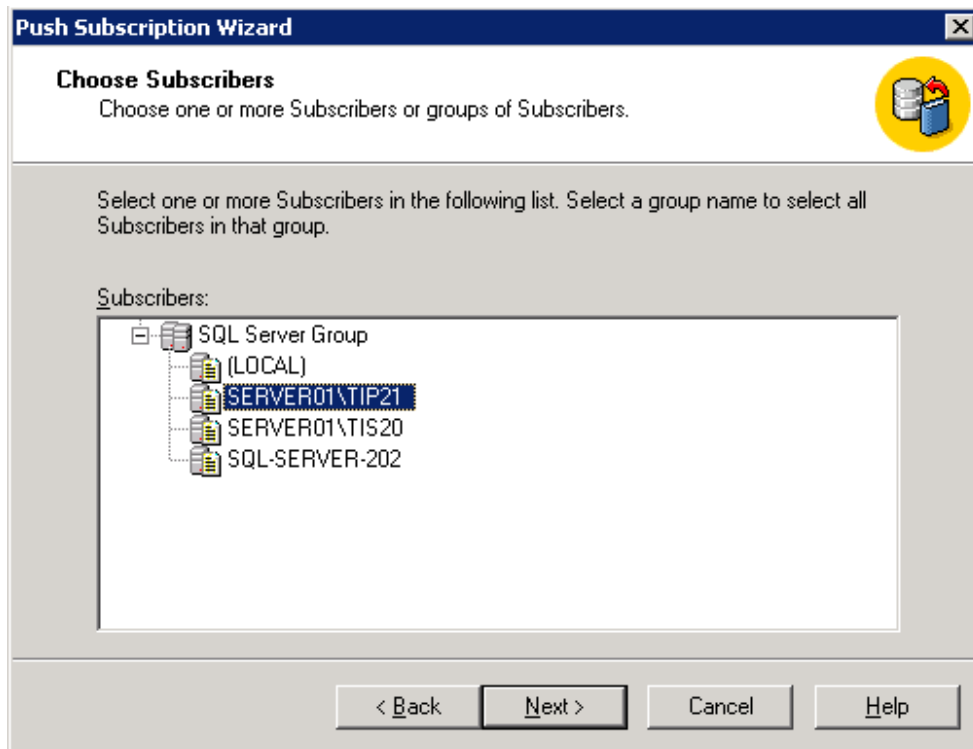
Langkah berikutnya adalah mengirimkan publikasi. Kini kita akan mengkonfigurasi pelanggan untuk publikasi dan membuat data base lain yang akan diisi dengan snapshot awal yang baru saja kita buat.

Pada kotak dialog Create and Manage Publications, klik pada tombol Push New Subscription.



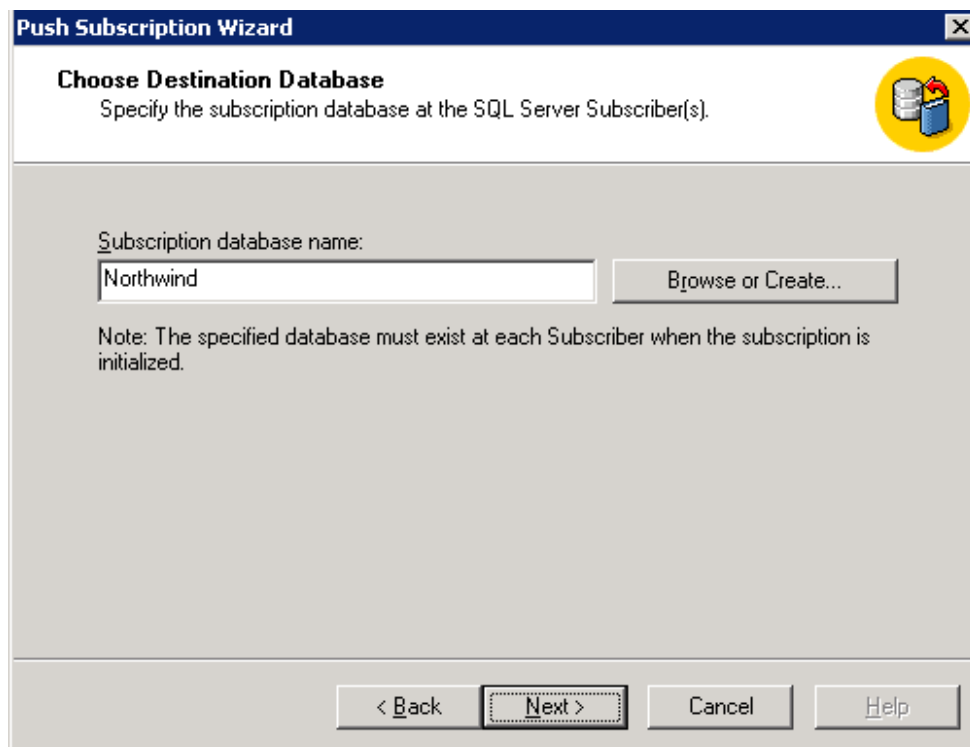
Gambar 11.19 Mengirimkan Publikasi

Push Subscription Wizard akan dijalankan. Klik pada tombol Next pada layar pertama yang muncul. Lalu klik pada Subscribers dan pilih server Anda dari daftar. Pada contoh kita, nama servernya adalah NTSERVER02. Jika Anda mengikuti contoh ini, pilihlah nama dari server Anda.



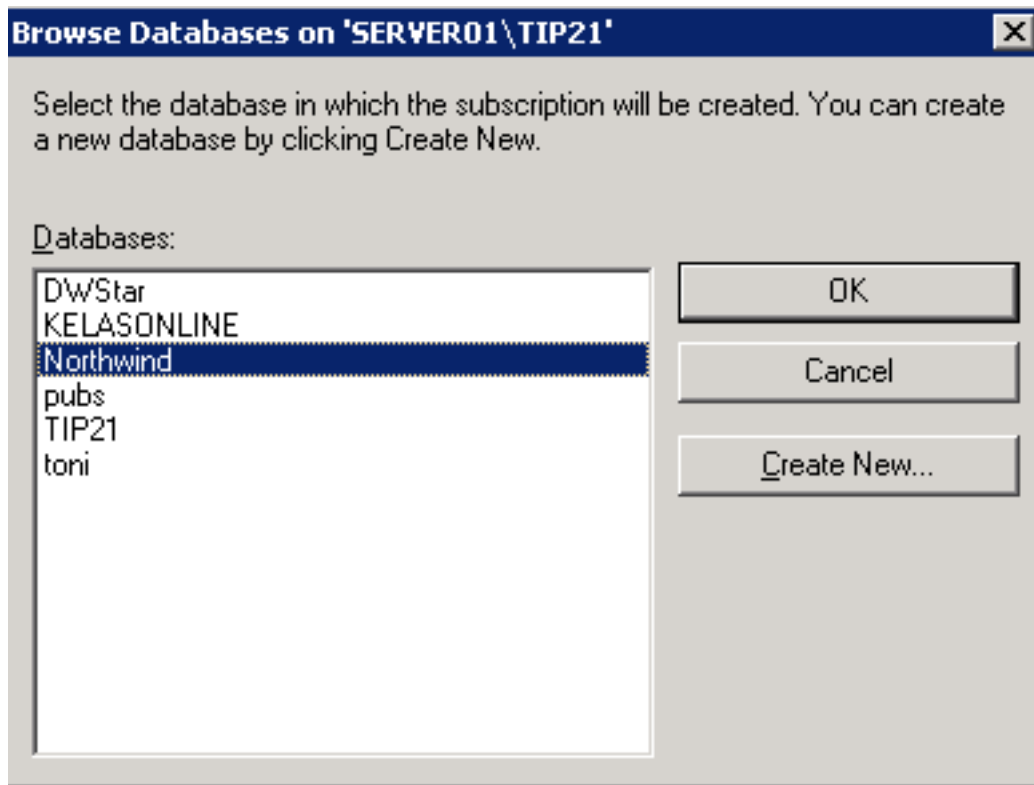
Gambar 11.20 Memilih Subscribers

Klik Next, lalu pilihlah database tujuan. Anda bisa mengetikkan namanya seperti ditunjukkan di bawah ini:



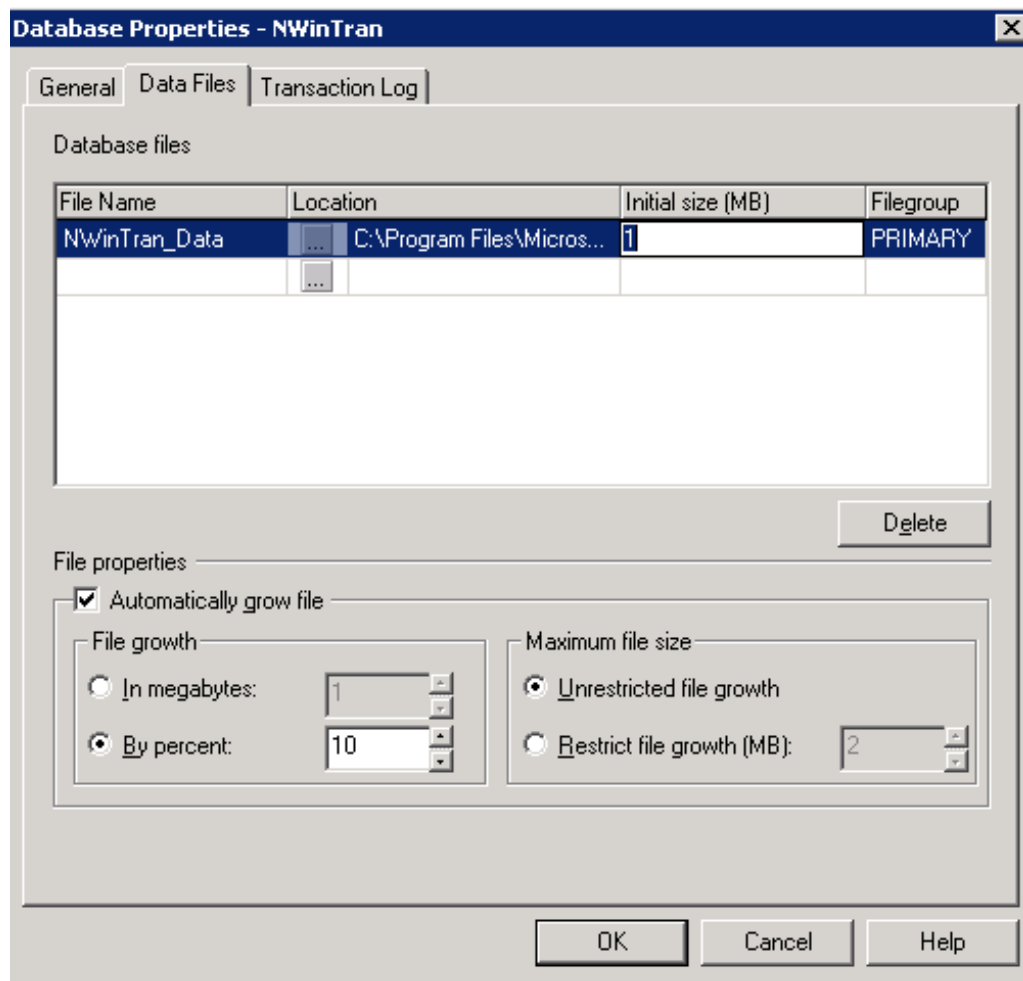
Gambar 11.21 Nama Database

Atau mengklik Browse Databases, yang akan menampilkan sebuah kotak dialog:



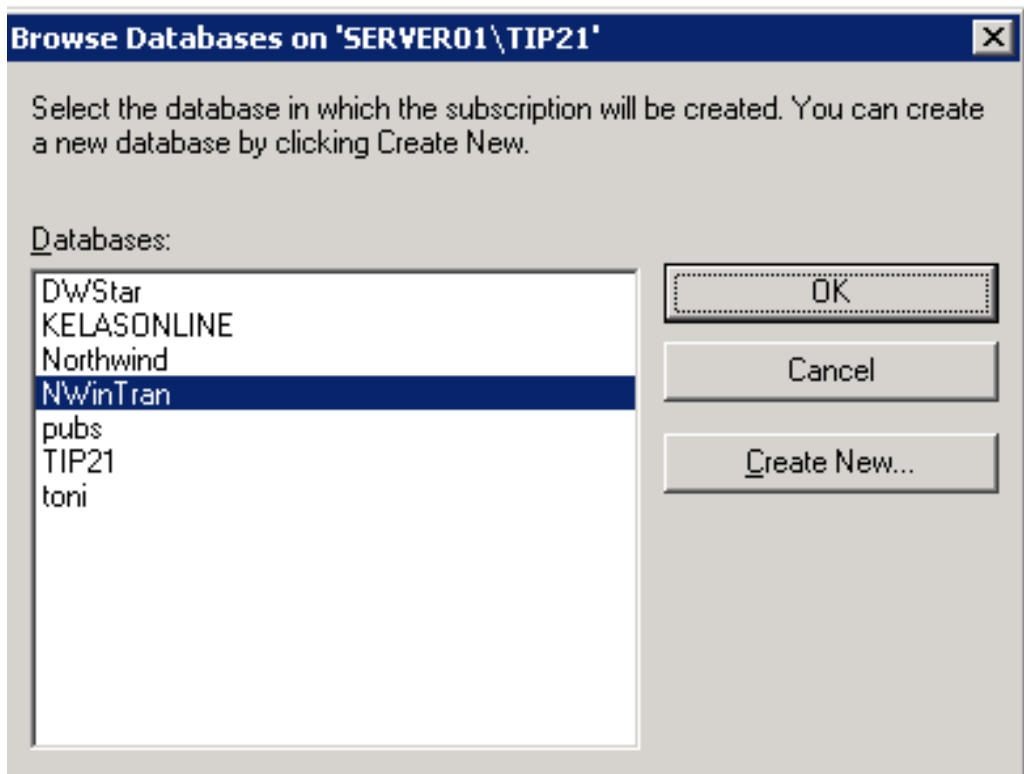
Gambar 11.22 Broswe Database

Pada contoh ini, kita perlu membuat database lain. Untuk melakukan itu, klik pada Create New. Ketikkan NwinTran pada field Name.



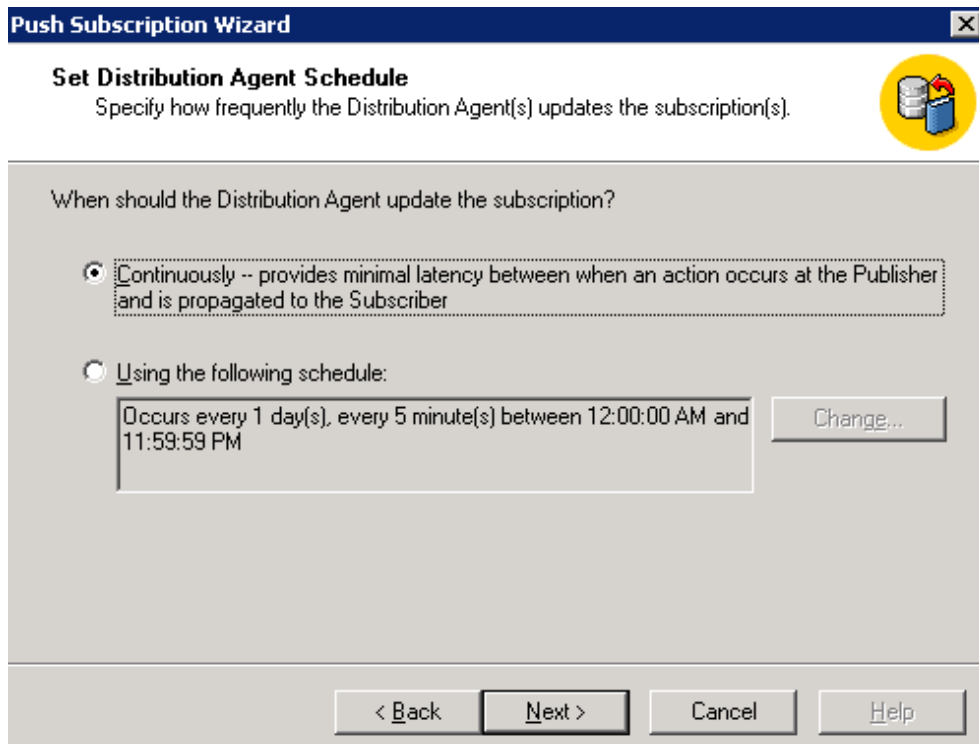
Gambar 11.22 Broswe Database

Tekan OK untuk kembali ke layar sebelumnya. Pilihlah database yang baru saja Anda buat, lalu tekan OK .



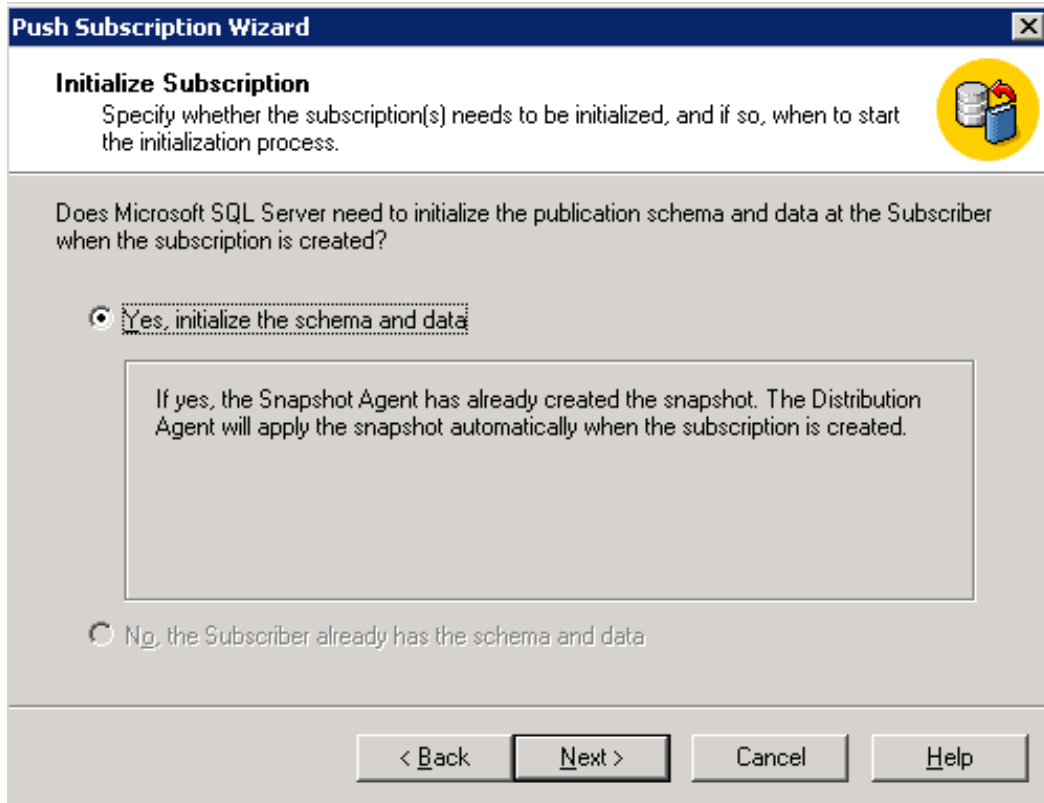
Gambar 11.23 NWinTran Database

Tekan Next akan muncul. Berikutnya, tentukanlah kapan publikasi tersebut akan di-update. Pilihlah Continuously lalu tekan Next.



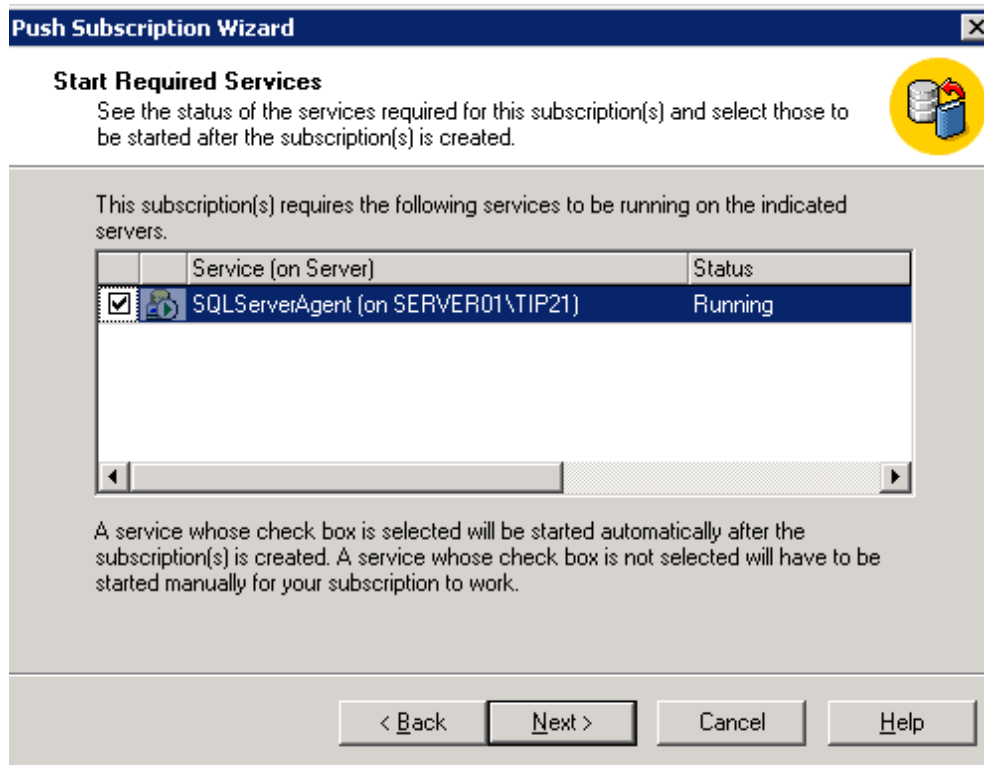
Gambar 11.24 Distribution Agent

Karena kita baru saja membuat database pelanggan, kita akan memilih untuk langsung menjalankan proses inialisasi. Lalu tekan Next.



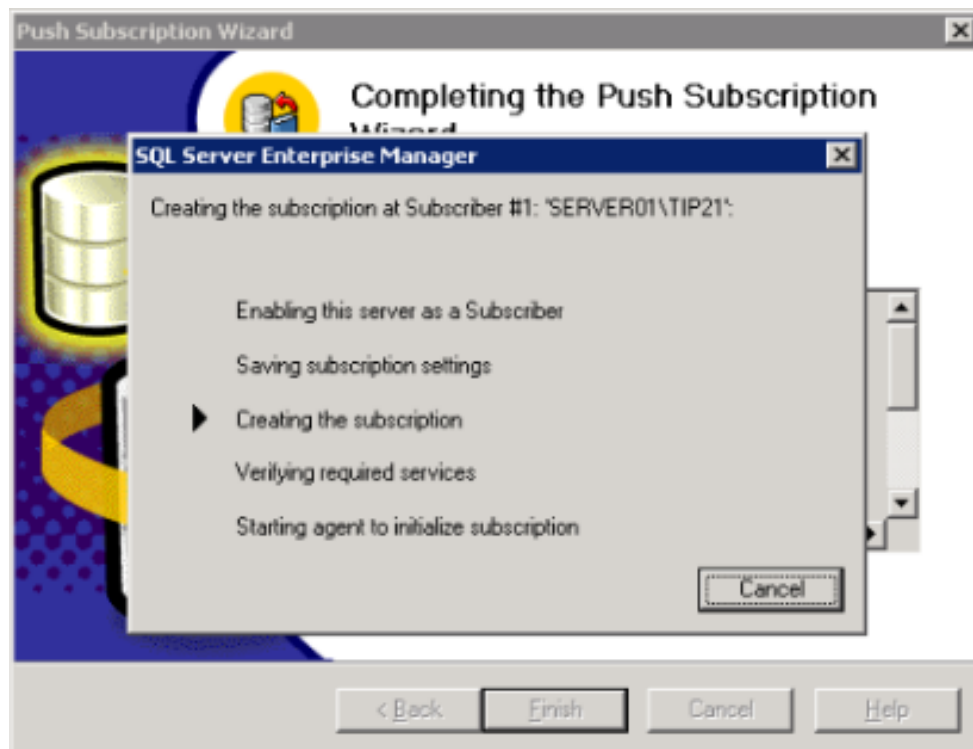
Gambar 11.25 Inialisasi Subscription

SQL Server Agent harus di-load agar operasi ini berjalan. Gambar berikut ini menampilkan statusnya.



Gambar 11.26 Memulai Inisialisasi Subscription

Klik Next, lalu klik Finish. Setelah beberapa saat, akan muncul pesanya menampilkan kemajuan dari operasi

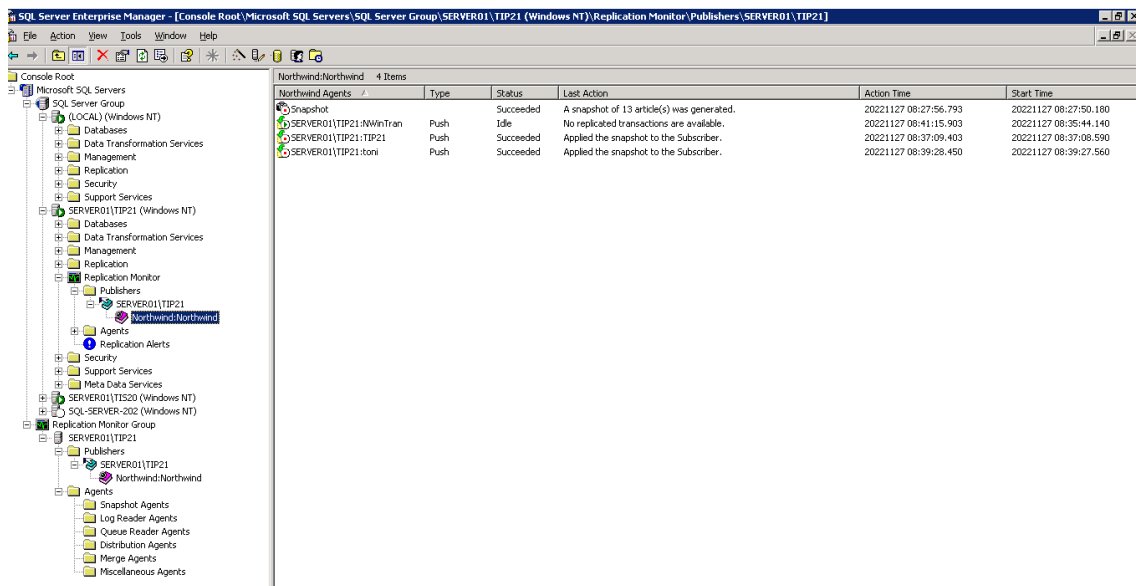


Gambar 11.27 Start Inisialisasi Subscription

Klik tombol Close.

Memeriksa Replikasi

Anda bisa melihat status dari replikasi dengan mengklik pada item Replication Monitor pada pohon database Enterprise Manager serta memperbesar item-itemnya sampai Anda menemukan publikasi, seperti ditampilkan pada gambar 11.28.



Gambar 11.28 Memeriksa Replikasi

Jika Anda mengklik beberapa teks pada panel kanan, mendapatkan lebih banyak informasi mengenai status dari sedang dilakukan.

Menguji Replikasi

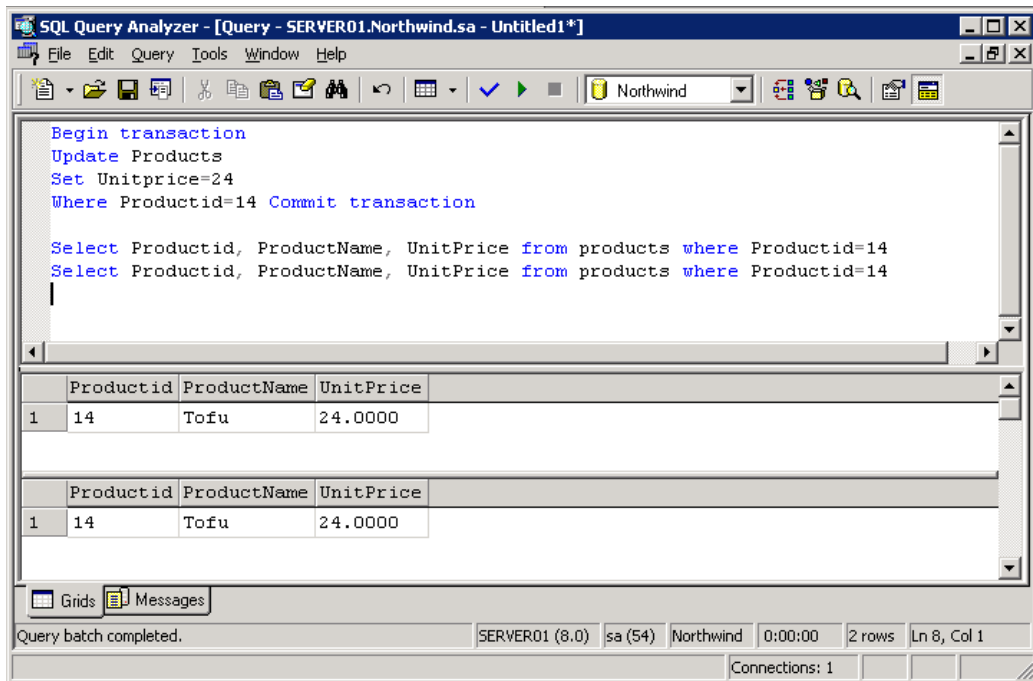
Mari kita menggunakan Query Analyzer untuk menguji replikasi. Setelah membuka dan berhubungan ke server, kita akan meng-update harga produk dan melihat apa yang terjadi. Lalu ubahlah isi dari tabel Product dengan skrip berikut ini:

```
Begin transaction  
Update Northwind.Products  
Set Unitprice=24  
Where ProductId=14 Commit transaction
```

Setelah itu, buatlah sebuah query baru dan jalankan:

Select ProductID, ProductName, UnitPrice from Northwind.products where ProductID=14
Select ProductID, ProductName, UnitPrice from NWinTran.products where ProductID=14

Anda akan melihat bahwa kini nilai produk akan sama pada kedua tabel.



Gambar 11.29 Menguji Replikasi

Kesimpulan

Walaupun ini merupakan proses yang rumit, replikasi itu sendiri merupakan konsep yang sederhana. Bab ini memberikan contoh mengenai replikasi transactional. Bab berikutnya menunjukkan cara membuat replikasi tipe merge.

BAB 12

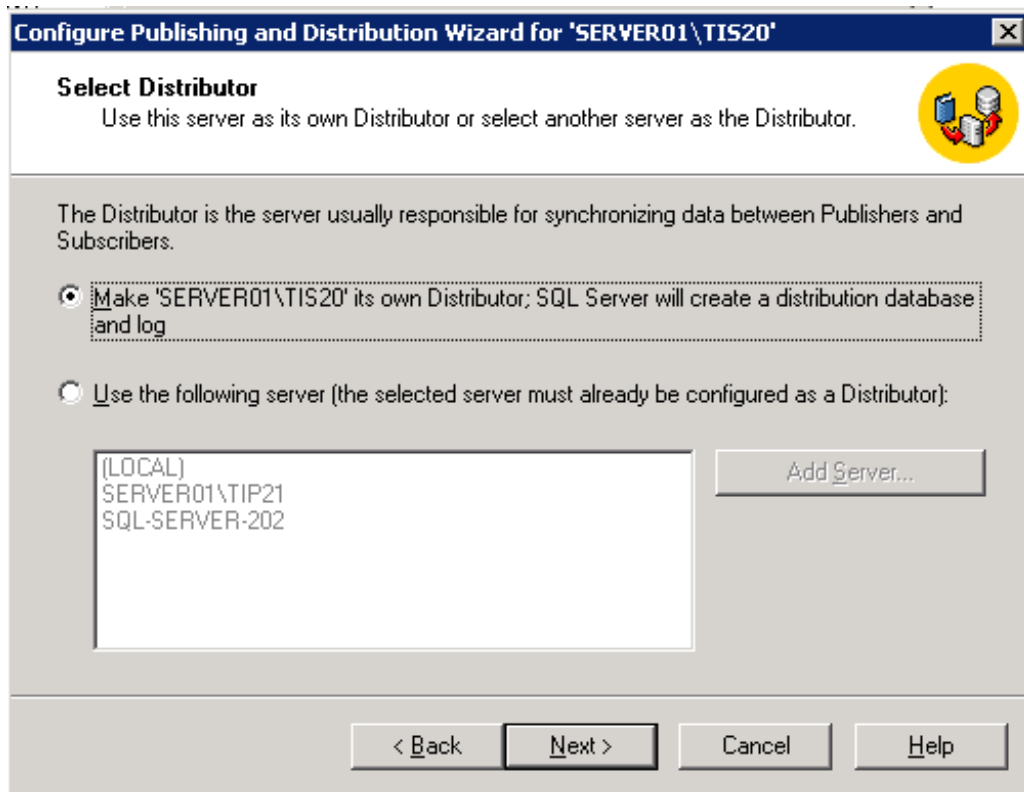
Replikasi Merge

Bab ini menunjukkan cara membuat replikasi gabungan di mana perubahan yang dibuat ke database sumber atau database target dapat diperbarui bolak-balik. Jika Anda telah melakukan langkah-langkah dalam membuat replikasi transactional, Anda bisa melompati ke bagian tentang mendefinisikan replikasi. Pada bagian pertama, kita akan mengkonfigurasi replikasi dari awal.

Bukalah Enterprise Manager dan klik pada nama server. Pada panel kanan, pilih Replication Data. Pada layar berikutnya, pilih Configure Replication. Ini akan membuka Configure Publishing and Distribution Wizard. Jika Anda sebelumnya pernah membuat replikasi, akan muncul layar yang berbeda. Jika ini terjadi, tutuplah dan bacalah bagian mengenai mendefinisikan replikasi.

Layar selamat datang wizard menyambut Anda dan memandu Anda melalui langkah-langkah yang diperlukan untuk menyiapkan publikasi Anda. klik berikutnya. Di layar berikutnya, pilih database distribusi, yang dalam contoh ini akan tetap berada di server lokal.

klik berikutnya. Layar konfigurasi akan muncul seperti yang ditunjukkan pada gambar di bawah ini. Kami akan menggunakan nilai default yang disediakan dengan mengklik tombol Berikutnya.

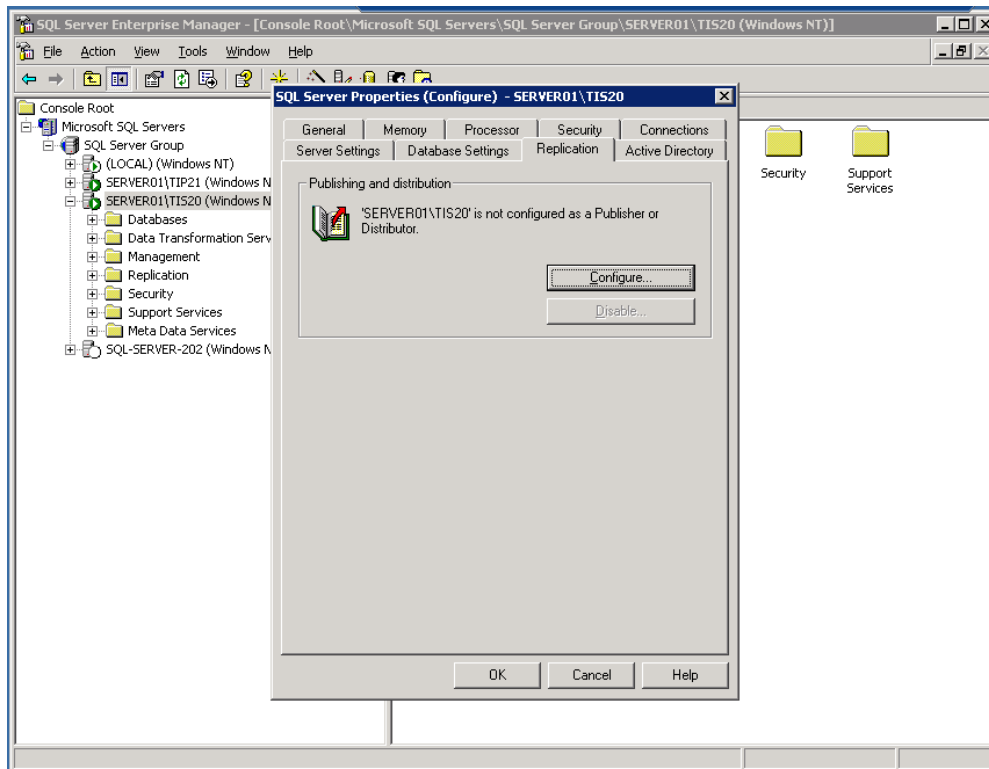


Gambar 12.1 Replikasi Merge

Wizard akan melengkapi tugas dan mengizinkan Anda menentukan database yang akan dipublikasikan dan didistribusikan. Lalu tekan tombol Finish. Kotak dialog menampilkan kemajuan dari operasi ini. Apabila sudah selesai, tutuplah tombol Close.

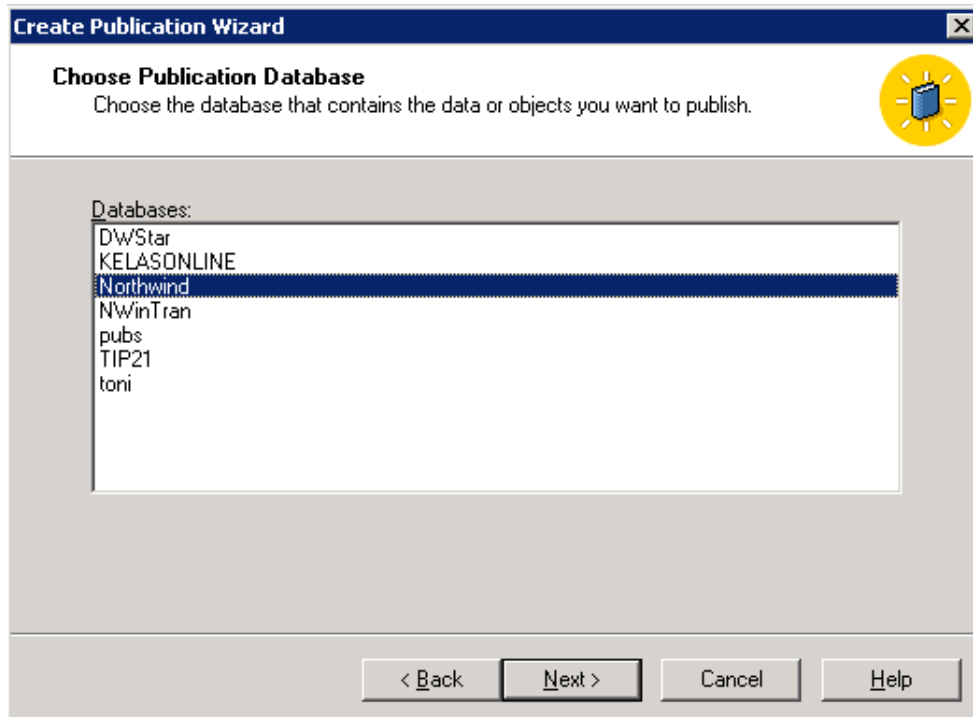
Mendefinisikan Replikasi

Pada layar dengan menu yang mengandung tugas-tugas replikasi, klik pada item Create or Manage a Publication.



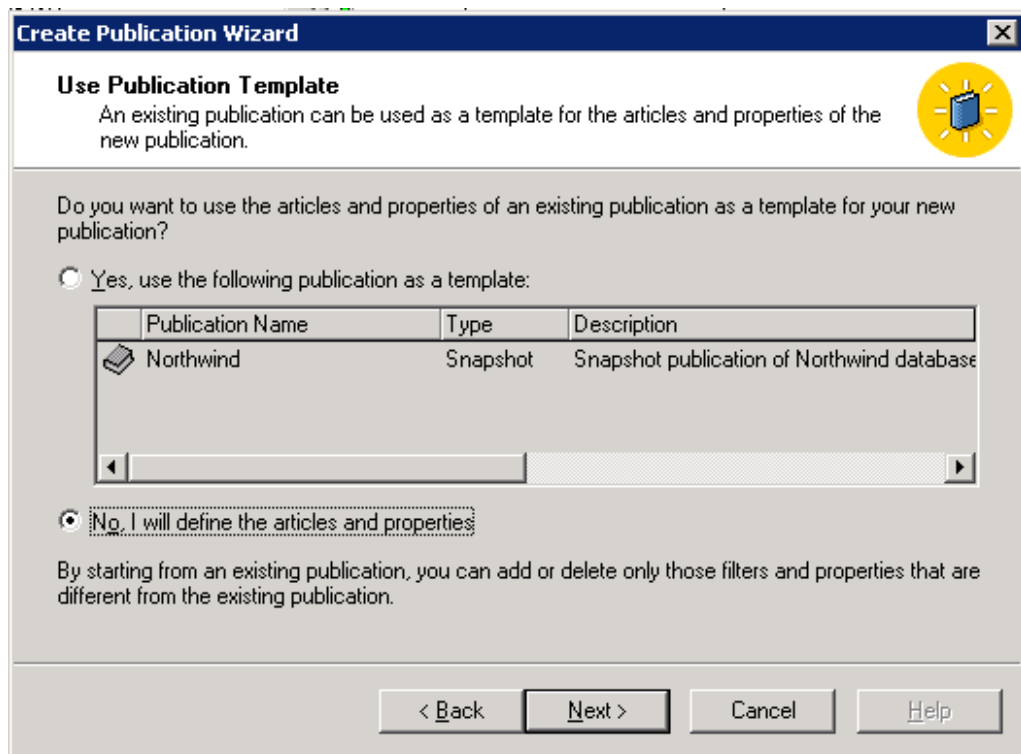
Gambar 12.2 Mendefinisikan Replikasi Merge

Kotak dialog Create or Manage a Publication akan ditampilkan. Pilihlah database Northwind lalu klik pada tombol Create Publication.



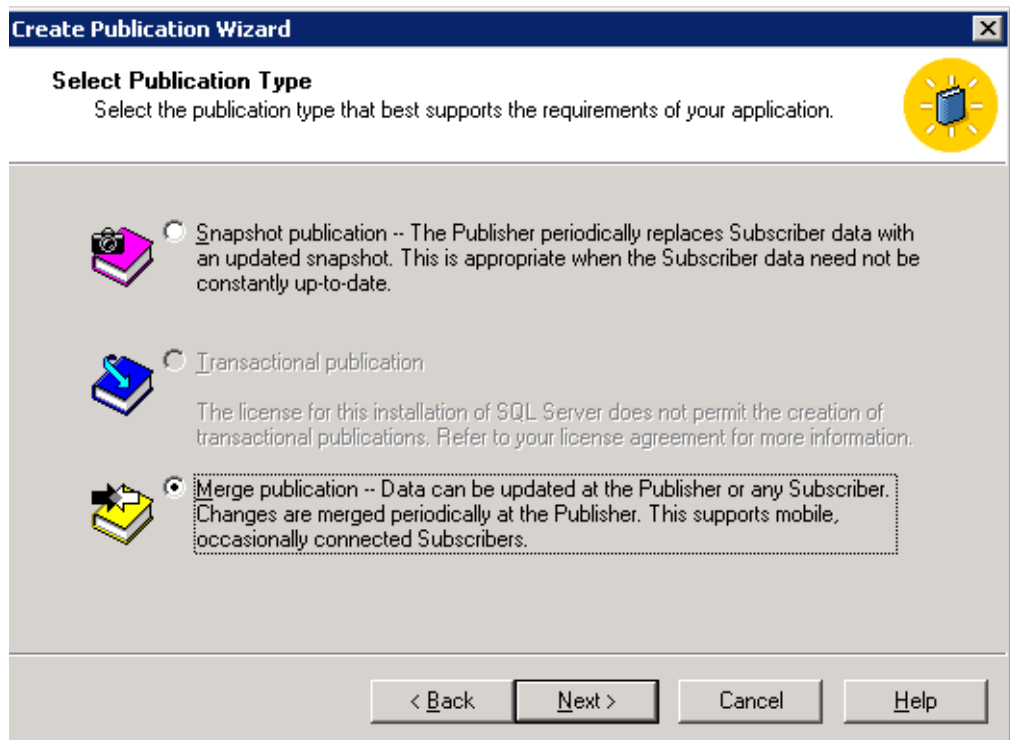
Gambar 12.3 Create or Manage a Publication

Langkah ini akan mengaktifkan Create Publication Wizard. Tekan Next untuk menuju ke layar kedua. Jika Anda pernah membuat publikasi lain, sebuah kotak dialog akan menanyakan apakah Anda ingin menggunakannya sebagai template atau pola. Dalam hal ini, pilihlah No dan tekan Next.



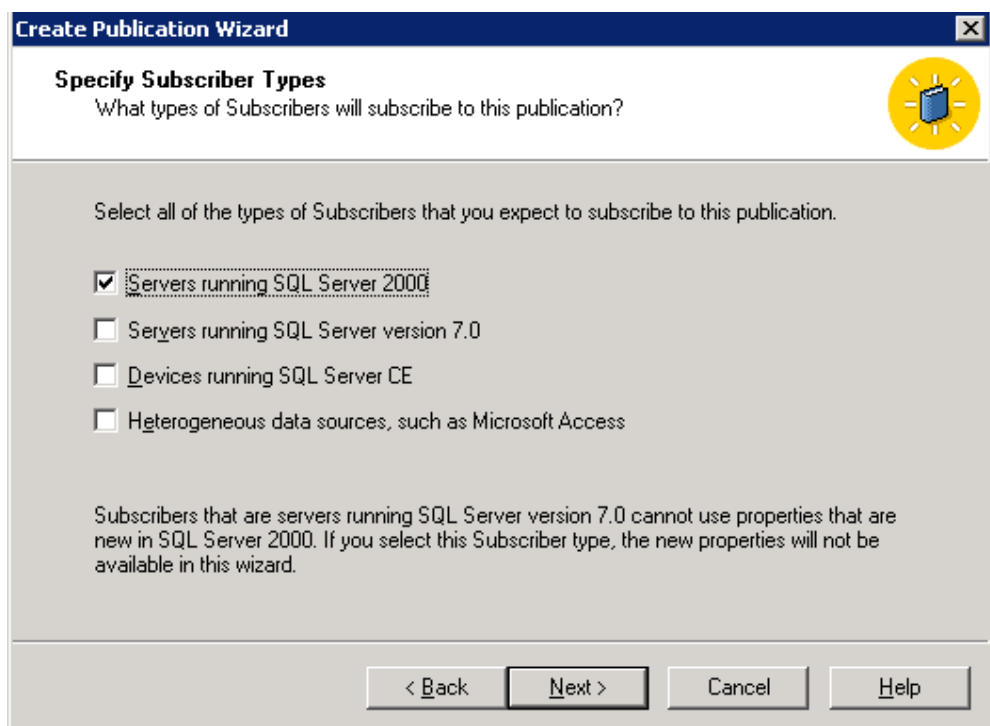
Gambar 12.4 Create Publication Wizard

Anda bisa memilih jenis publikasi yang ada. Pada contoh ini, pilihlah Merge Publication lalu tekan Next.



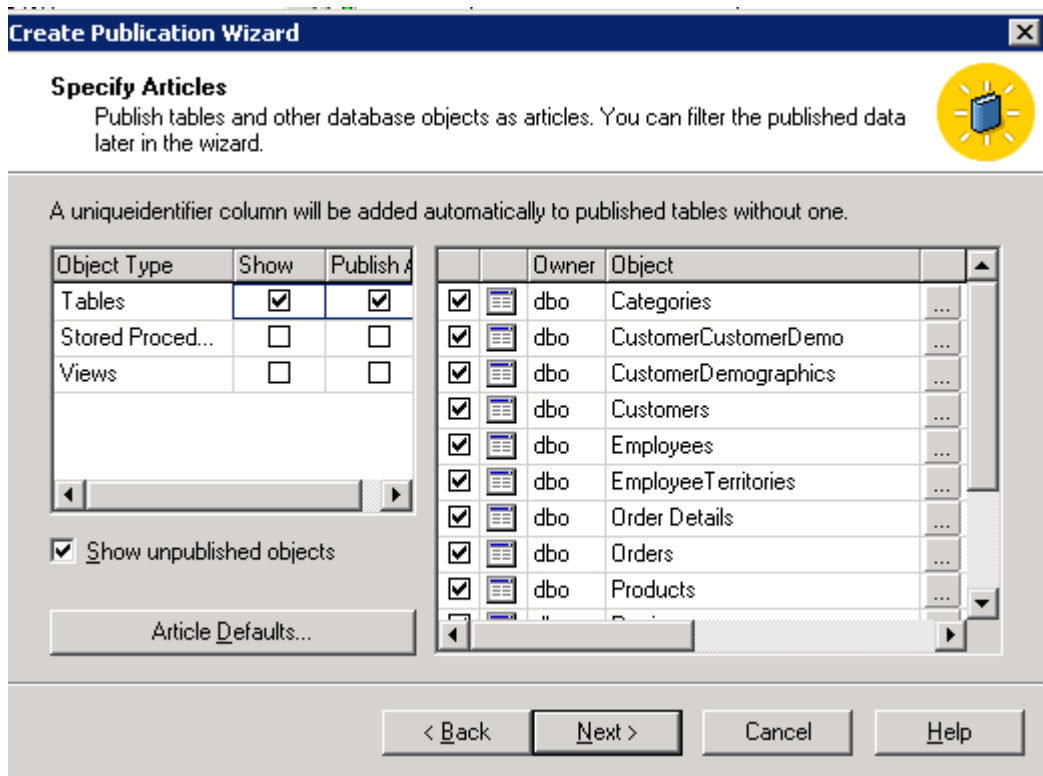
Gambar 12.5 Jenis Publikasi

SQL mengizinkan penggunaan pelanggan yang tidak berada di dalam database SQL. Tetapi, pada contoh ini kita akan memilih All Subscribers, yang menyatakan semua pelanggan adalah bagian dari database SQL Server.



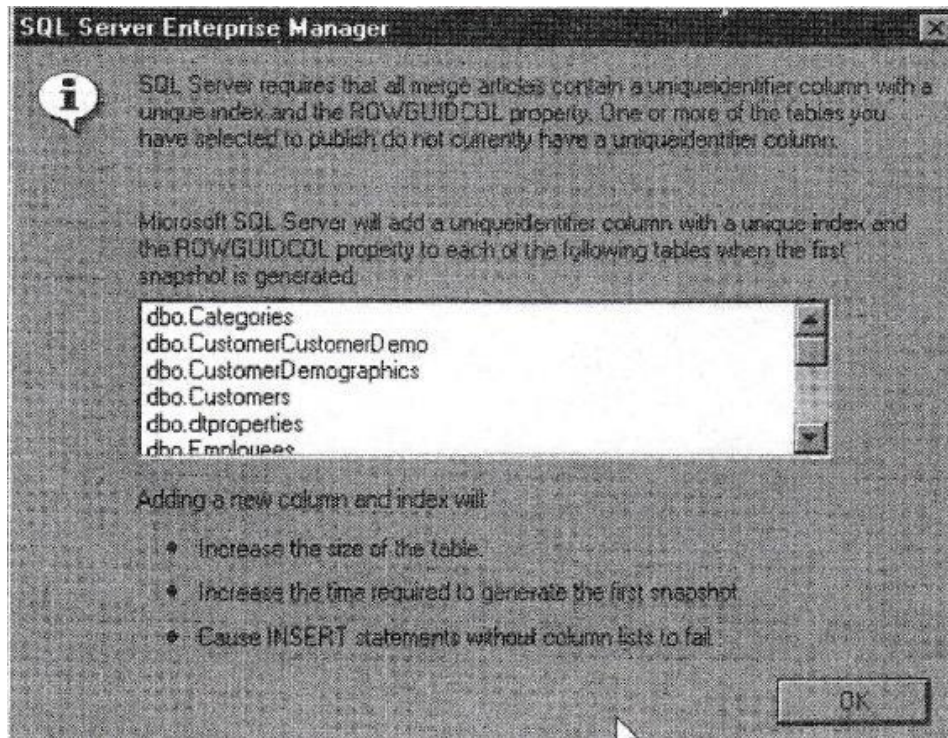
Gambar 12.6 Jenis Subscribers

Tekan tombol Next untuk melanjutkan. Langkah berikutnya adalah mendefinisikan artikel atau objek yang akan dipublikasikan. Semua tabel yang tersedia akan ditampilkan. Pada contoh ini, kita akan memilih semuanya. Untuk melakukan itu, tekanlah Publish All lalu tekan Next.



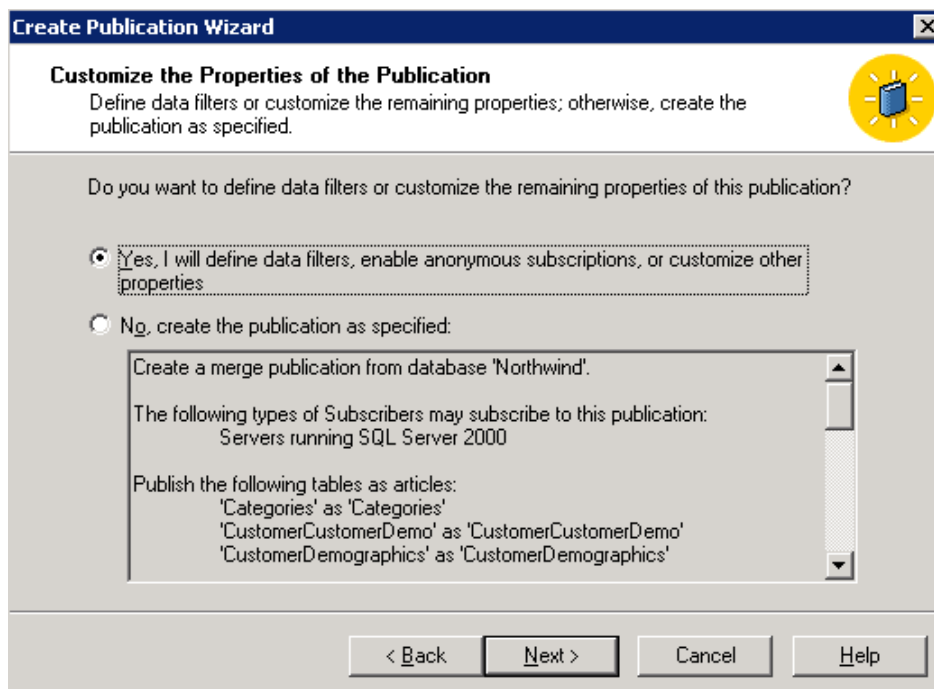
Gambar 12.7 Jenis Subscribers

Pada replikasi merge, SQL Server membutuhkan semua tabel memiliki pengenal yang unik. Ia akan membuat pengenal yang unik untuk tabel-tabel yang tidak memilikinya. SQL Server akan memberi tahu Anda soal ini dan juga memberi tahu bahwa ukuran tabel Anda akan menjadi lebih besar karenanya. Tekan OK.



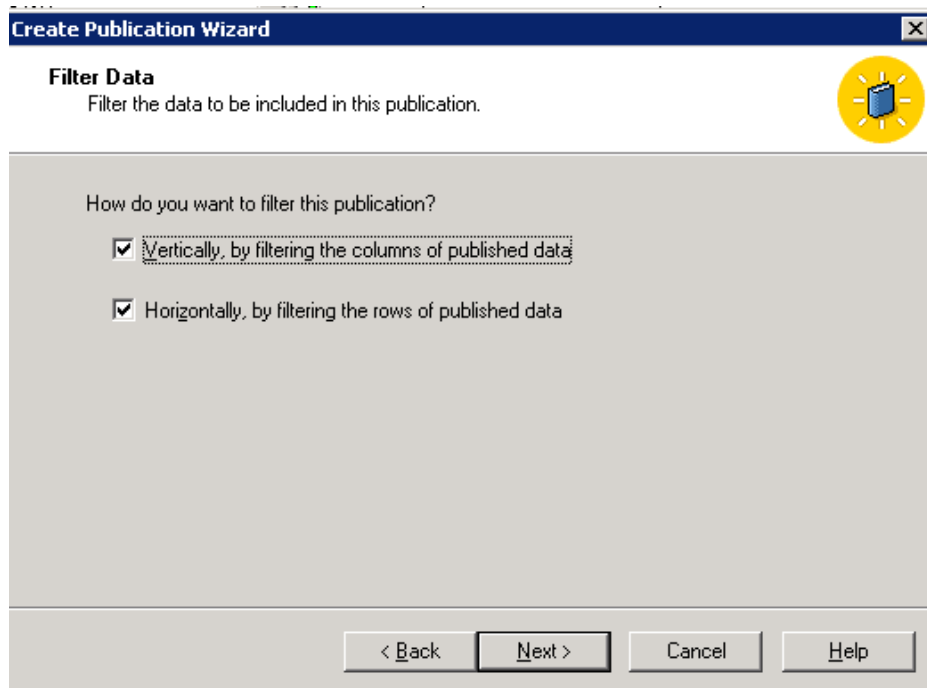
Gambar 12.8 Subscribers

Pada langkah berikutnya, ketikkanlah nama untuk replikasi ini. Tambahkan kata_Merge di belakang kata Northwind, lalu tekan Next untuk melanjutkan. Berikutnya Anda bisa melakukan replikasi atau publikasi tanpa filter. Klik Yes lalu tekan Next.



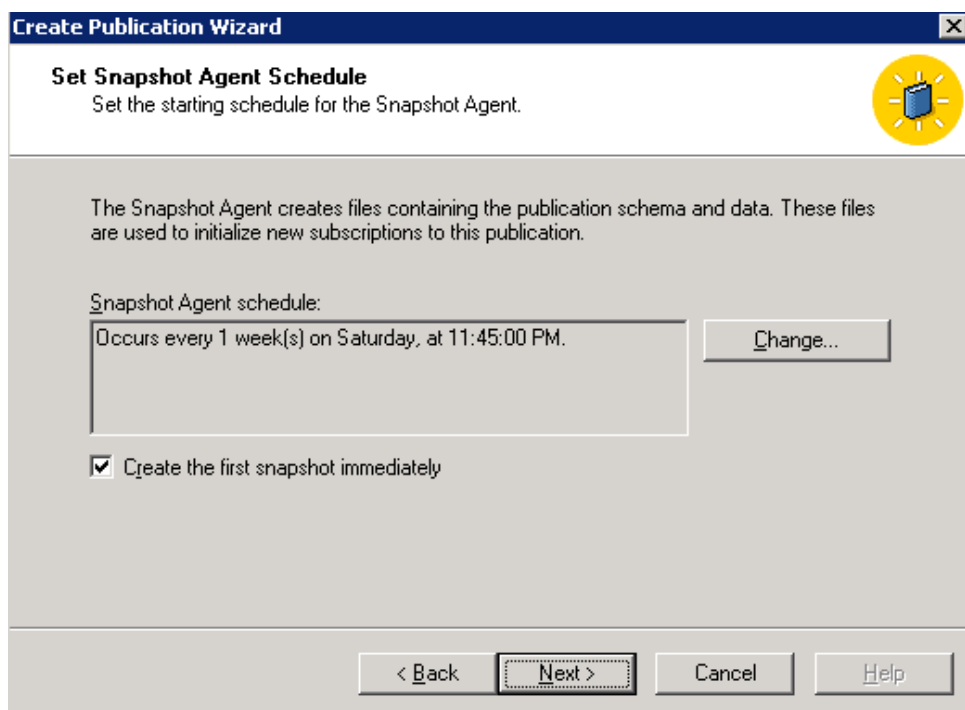
Gambar 12.9 Replikasi atau Publikasi

Lalu pilih No, dan tekan tombol Next.



Gambar 12.10 Filter Data

Kini kita akan mengizinkan pelanggan anonymous dengan memilih Yes dan menekan tombol Next. Layar penjadwalan dari Snapshot Agent akan ditampilkan.



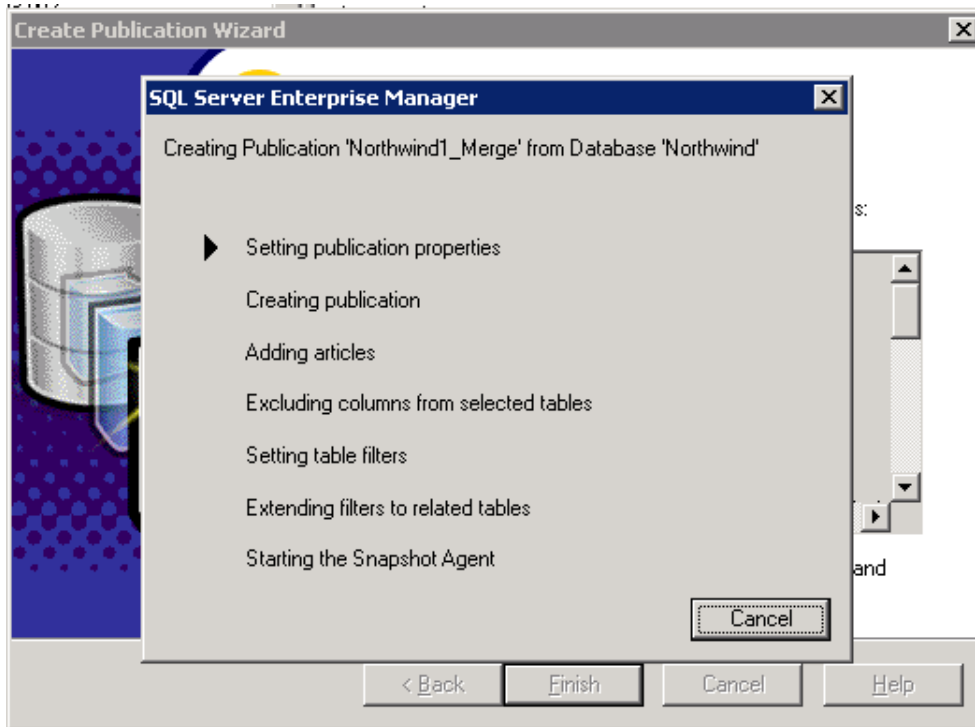
Gambar 12.11 Schedule

Setelah Anda menekan Next pada layar penjadwalan, Anda akan melihat sebuah layar yang menampilkan pilihan-pilihan yang telah Anda pilih. Tekan tombol Finish, dan proses pembuatan akan segera dijalankan.



Gambar 12.12 Completing the Create Publication Wizard

Sebuah kotak yang menampilkan kemajuan dari tugas-tugas akan ditampilkan.

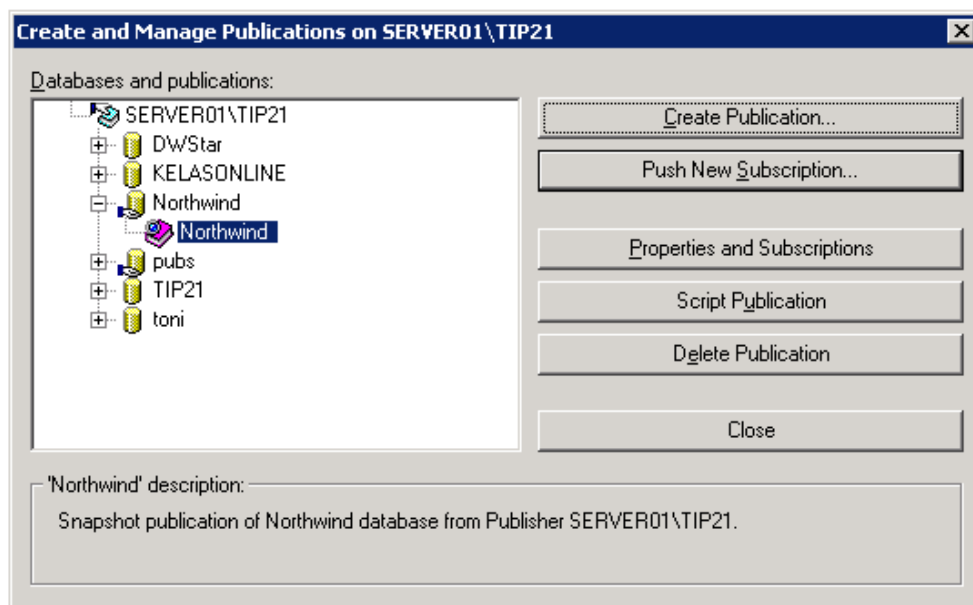


Gambar 12.13 Completing the Create Publication Wizard

Kemudian sebuah jendela akan memberi tahu bahwa proses sudah selesai.

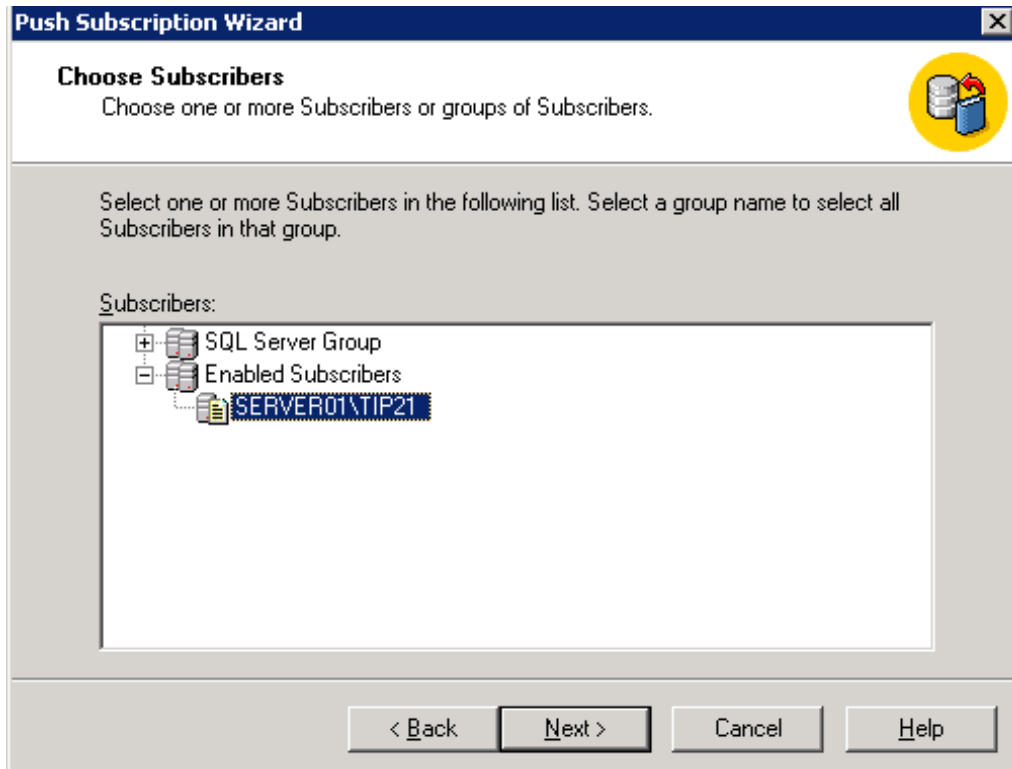
Mengirimkan Publikasi

Apabila Anda menekan tombol Close atau OK, kotak dialog Create and Manage a Publication akan muncul. Pilihlah Northwind_Merge lalu klik tombol Push New Subscription.



Gambar 12.14 Mengirimkan Publikasi

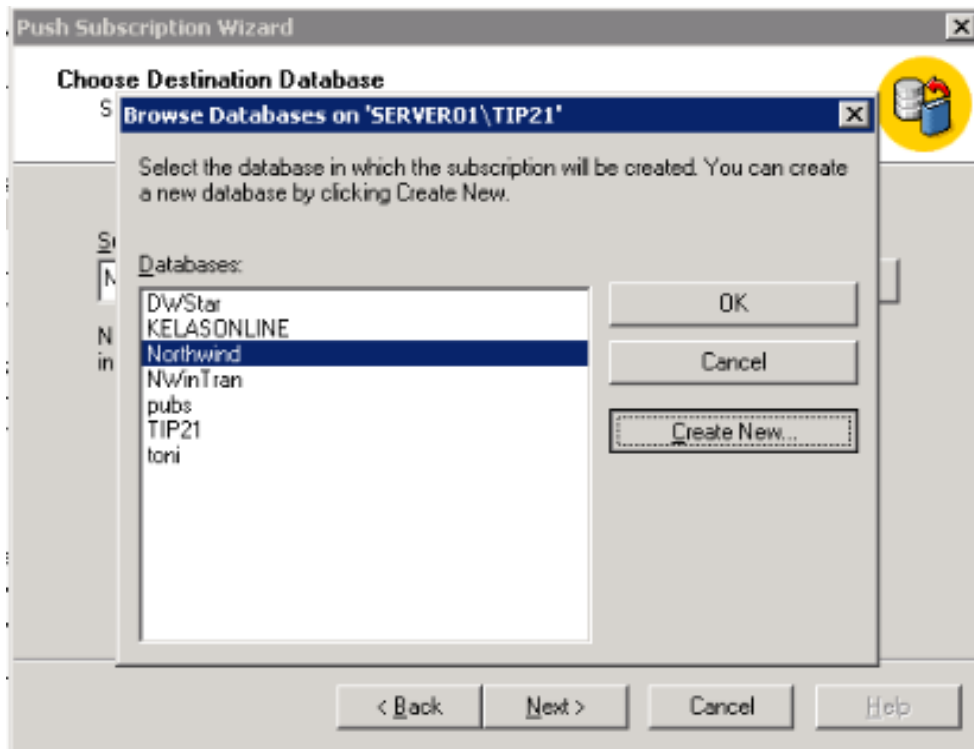
Ini akan mengaktifkan Push Subscription Wizard. Pada layar pertama, klik Next. Lalu klik Enabled Subscribers dan pilih server Anda pada daftar.



Gambar 12.15 Mengirimkan Publikasi Wizard

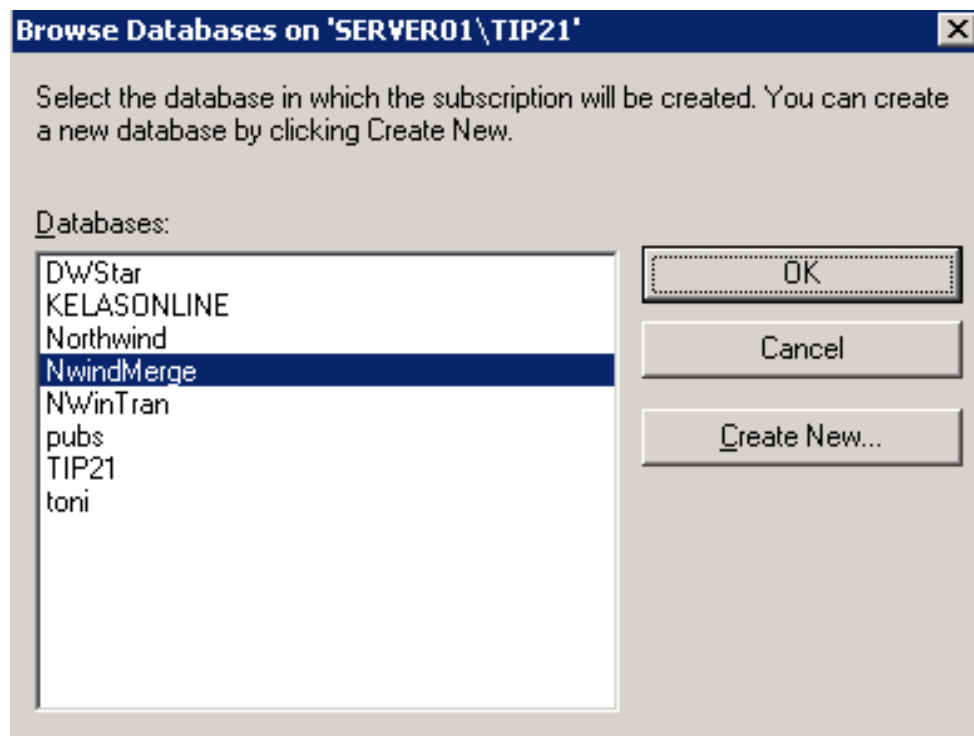
Setelah memilih server, klik pada Next dan pilih database tujuan.

Pada contoh ini, kita perlu membuat database lain. Untuk melakukan itu, klik pada Browse Databases untuk menampilkan kotak dialog di bawah ini, lalu klik Create New.



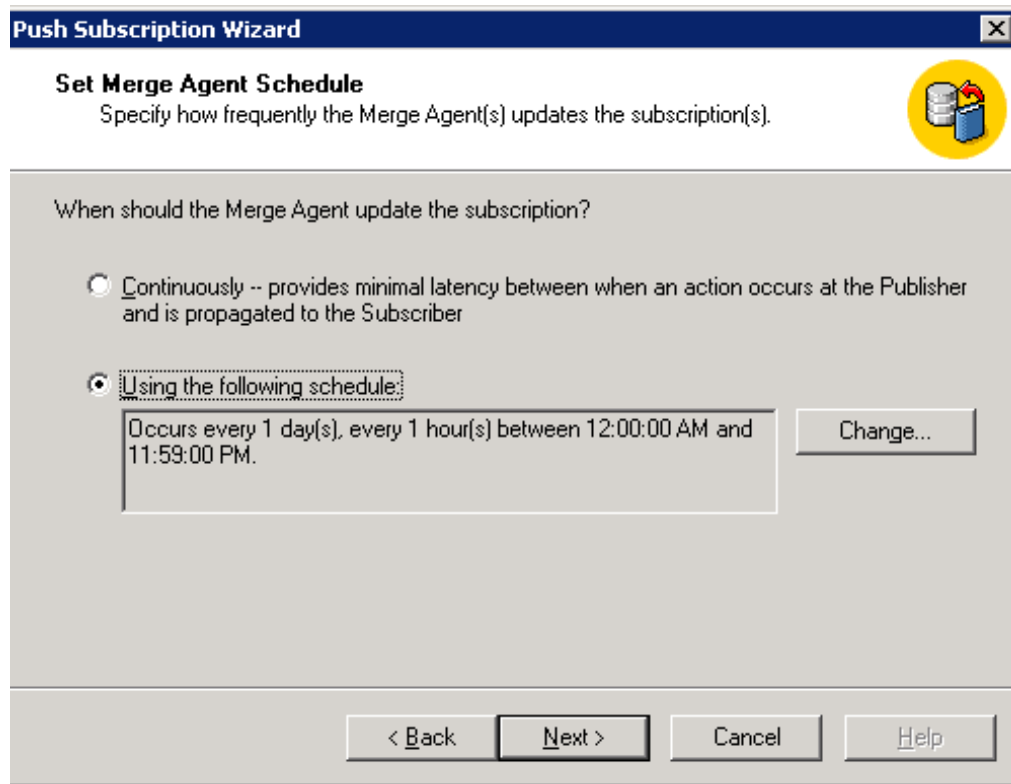
Gambar 12.16 Browse Database

Ketikkan NwindMerge sebagai nama dari database yang baru ini, lalu tekan tombol OK. Ini akan mengembalikan Anda ke layar sebelumnya. Pilihlah database ini, lalu tekan OK.



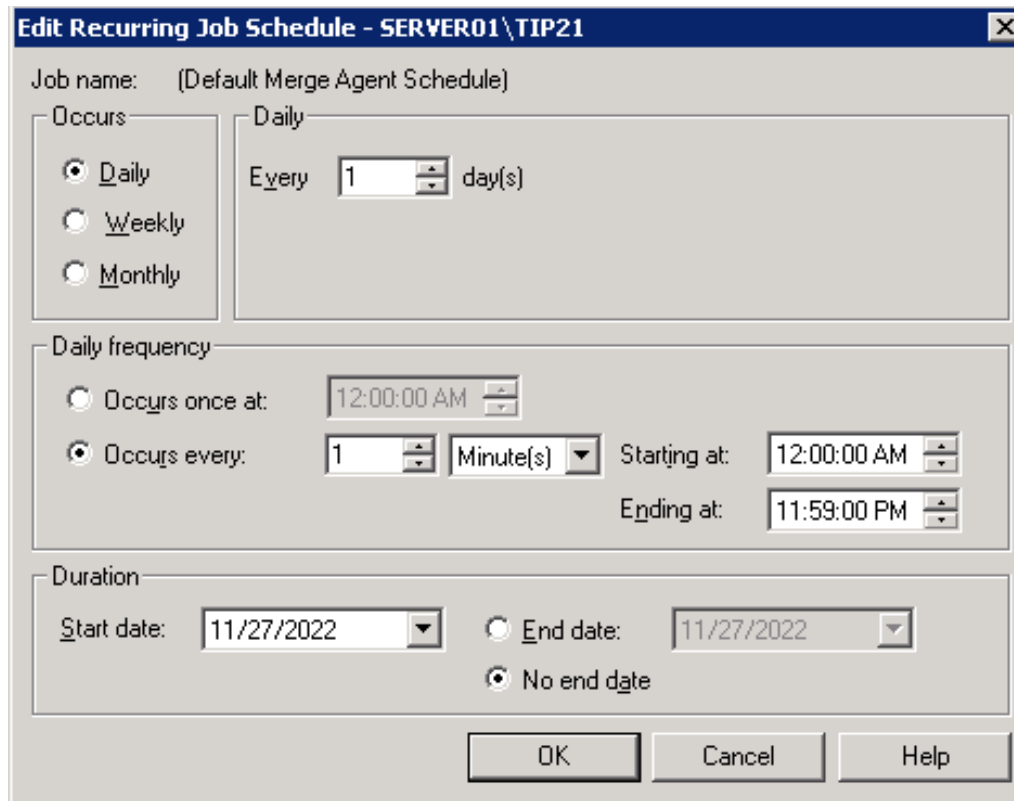
Gambar 12.17 Memilih Database

Tekan Next untuk menampilkan kotak dialog baru. Di sini Anda akan menentukan apakah publikasi tersebut akan di-update. Untuk mengubah frekuensi update, gunakanlah Change.



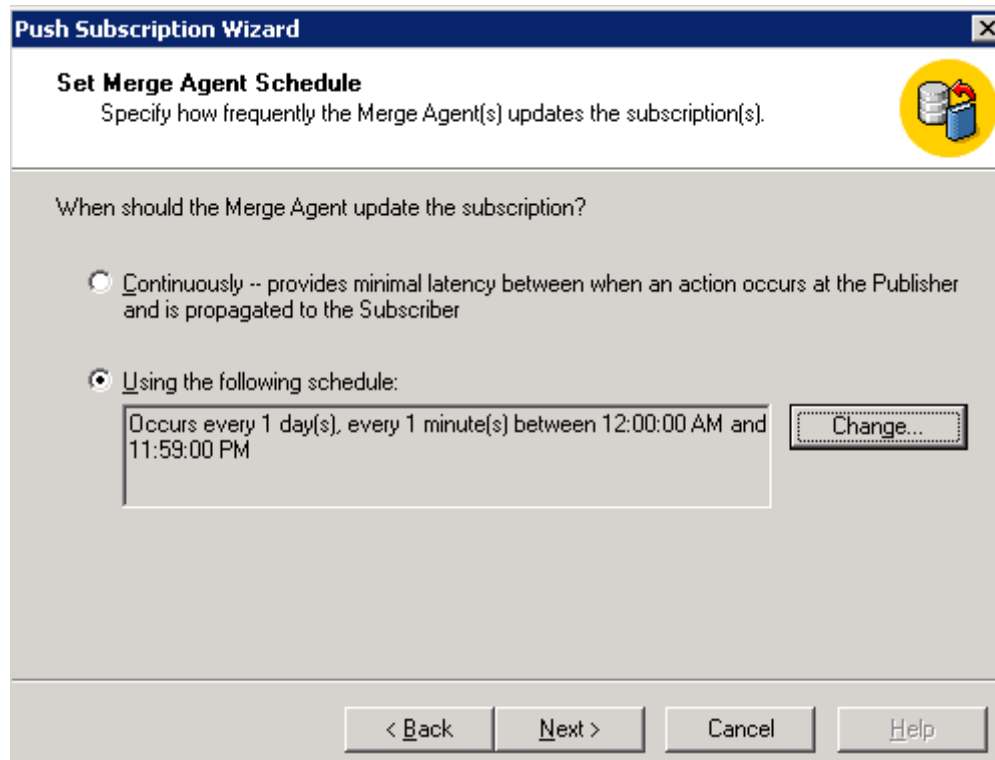
Gambar 12.18 Set Merge Agent Schedule

Pada contoh ini, kita akan mengubah frekuensinya menjadi 1 menit. Klik pada tombol Change untuk menampilkan kotak dialog, lalu ubahlah field Occurs Every dari Hours menjadi Minutes.



Gambar 12.19 Edit Schedule

Lalu klik OK. Klik Next untuk menerima perubahan.



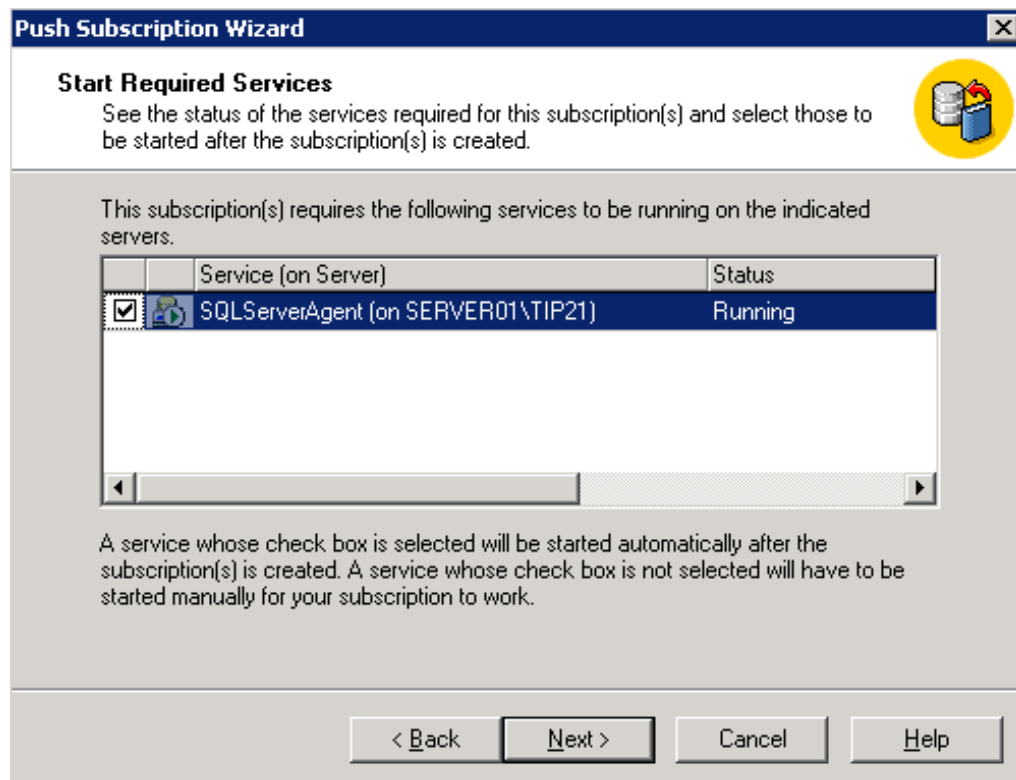
Gambar 12.20 Menggunakan Schedule

Karena kita baru saja membuat database pelanggan, kita akan segera melakukan proses inisialisasi. Pilih Yes lalu tekan Next.



Gambar 12.21 Inisialisasi

Tekan Next untuk melanjutkan. Pada langkah berikutnya, SQL Server Agent harus di-load. Layar berikutnya menampilkan statusnya.

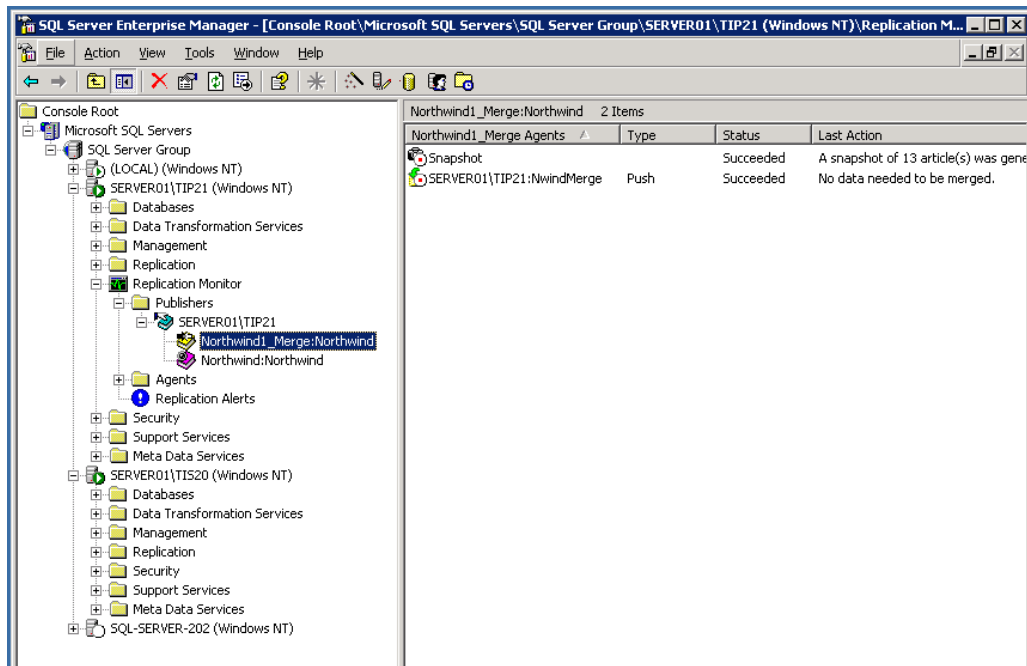


Gambar 12.22 Menjalankan Agent

Jika status kolom tidak menampilkan kata "Running," berarti SQL Server Agent harus diaktifkan sebelum Anda melanjutkan. Klik Next lalu klik Finish. Sebuah pesan akan menunjukkan bahwa proses telah selesai. Untuk menyelesaikan proses, klik tombol Close. Untuk kembali ke kotak dialog Create and Manage a Publication, tekan tombol Close .

Memeriksa Publikasi

Untuk melihat status dari replikasi, klik pada item Replication Monitor dan perbesarlah item-itemnya sampai Anda menemukan publikasi yang telah Anda buat, seperti ditunjukkan di bawah ini.



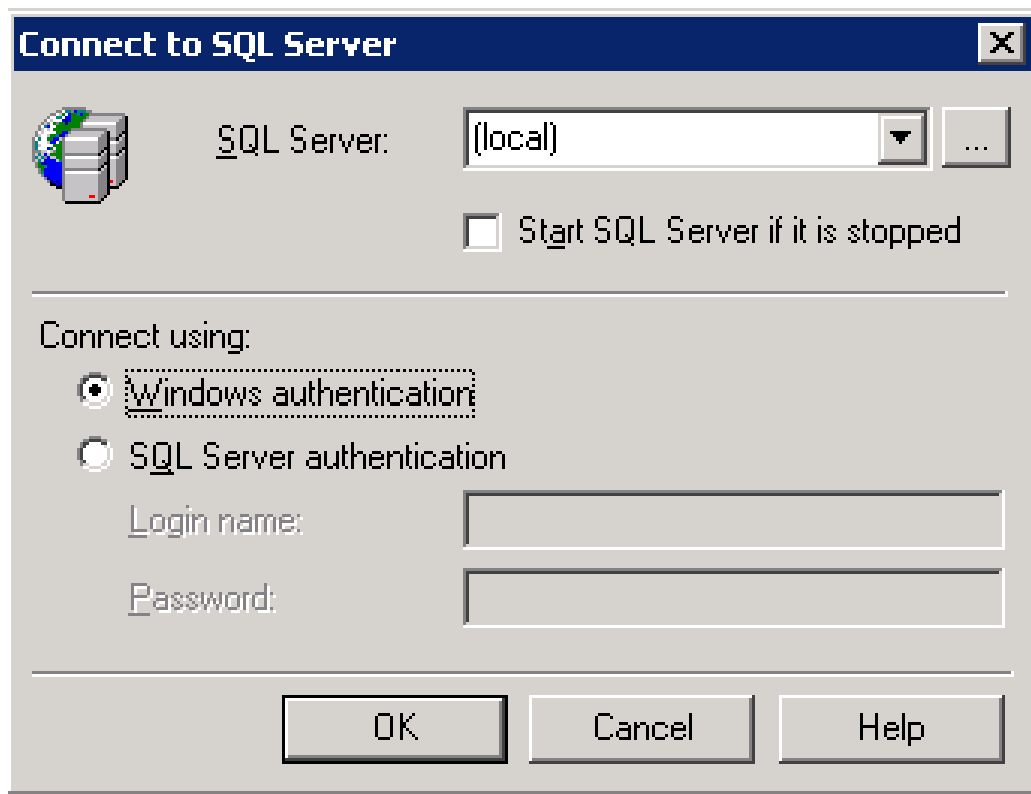
Gambar 12.23 Memeriksa Publikasi

Menguji Replikasi

Untuk menguji replikasi, kita akan membuat beberapa perubahan pada database Publishers dan melihat apakah mereka sudah benar. Untuk melakukan ini, kita akan menggunakan Query Analyzer dan menjalankan beberapa kode singkat. Setelah setiap eksekusi, kita akan lihat apa yang terjadi pada tabel yang sedang aktif dan menggunakan Conflict Solver untuk menghapus konflik yang ada.

Menggunakan Query Analyzer

Query Analyzer harus dibuka dari menu SQL. Pada contoh ini, pilihlah Use Windows NT Authentication. Tekan OK.

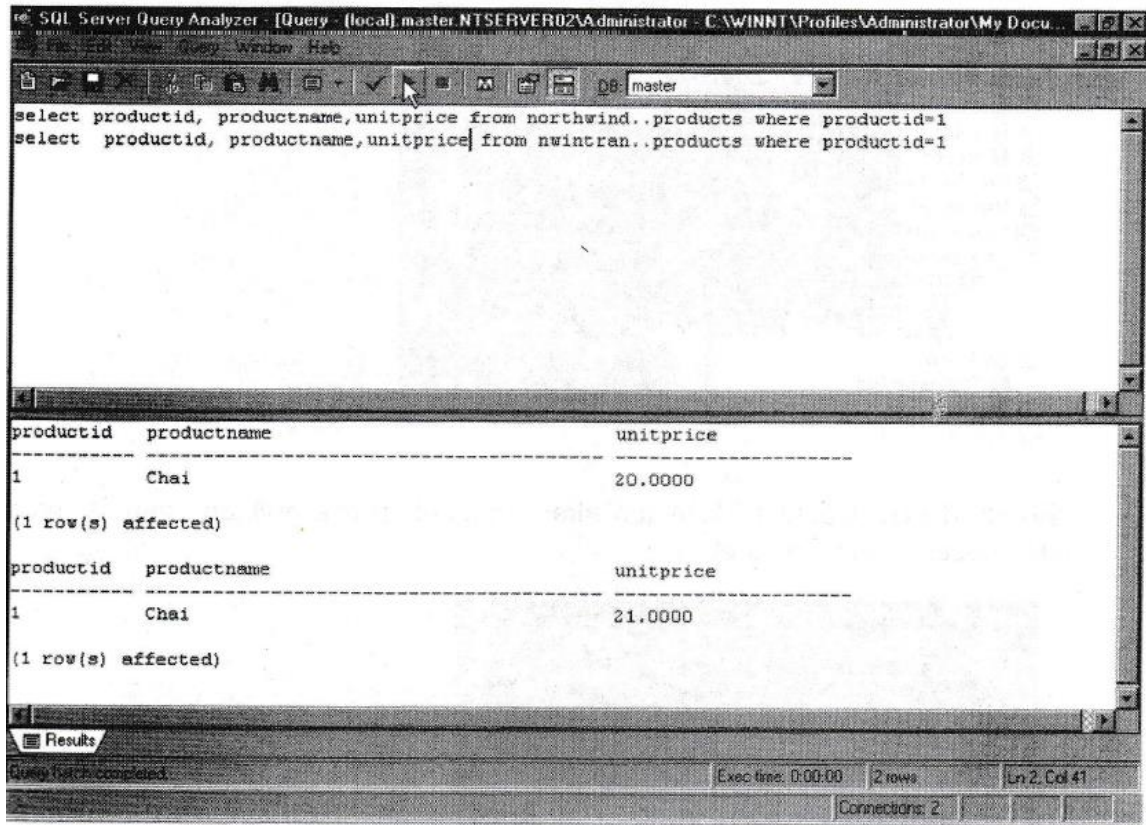


Gambar 12.24 Menggunakan Query Analyzer

Pada layar Query Analyzer, masukkan kode berikut:

```
begin tran
  update northwind.products
    set unitprice=20 where productid= 1
  update nwindmerge ..products
    set unitprice=21 where productid= 1
commit tran
```

Kode ini menunjukkan bahwa harga unit produk dengan productid 1 akan diperbesar menjadi 20 pada penerbit dan 21 pada pelanggan, sehingga akan menimbulkan konflik. Tekan tombol Run untuk menjalankan perintah ini. Kini saatnya melihat bagaimana record akan ditampilkan pada kedua database. Jika kita menjalankan kode berikut ini, kita akan melihat bahwa harga produk pada kedua database akan sama.

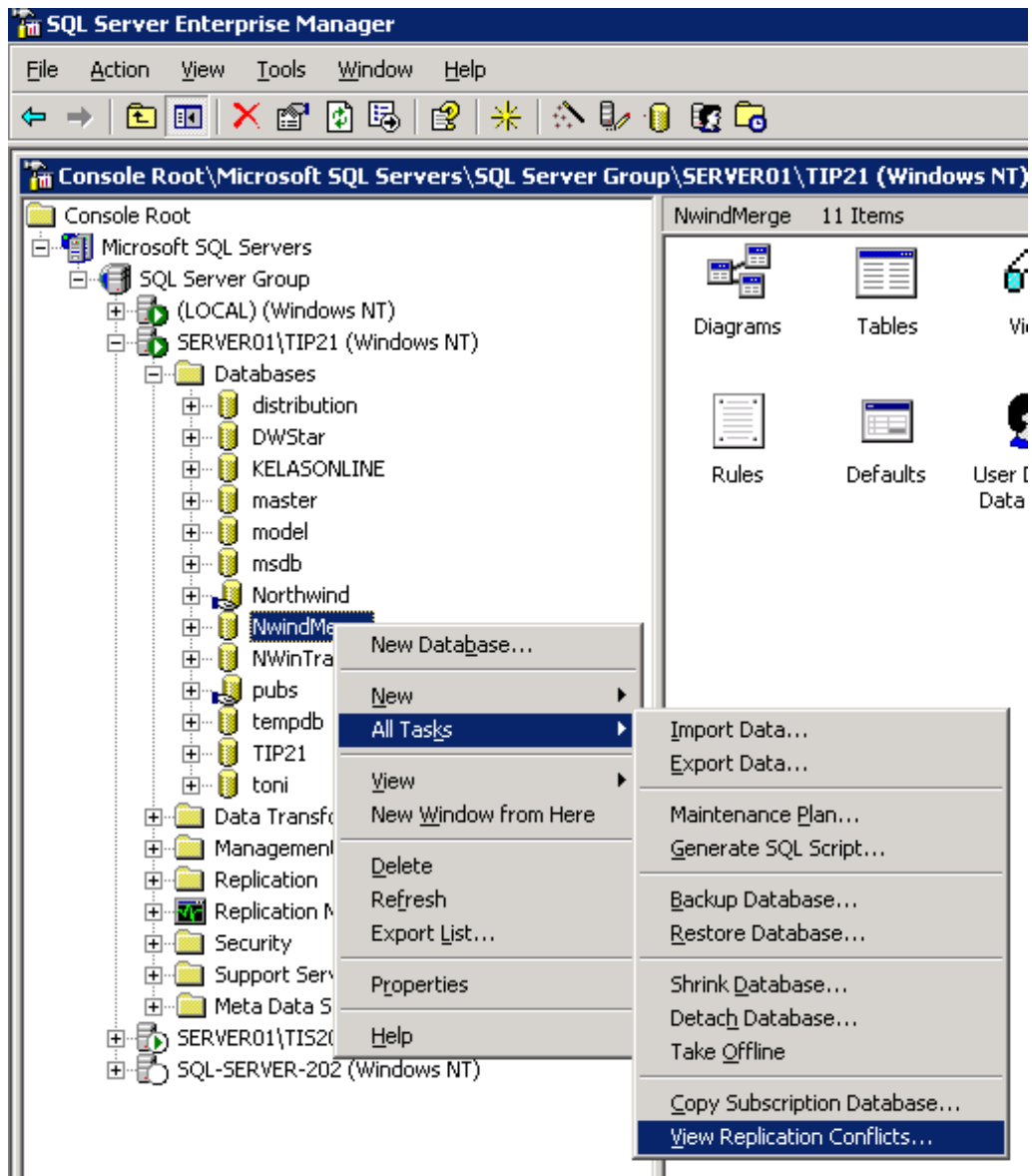


Gambar 12.25 Hasil Query Analyzer

Ini menunjukkan bahwa database Northwind menghapus nilai dari produk di dalam database NwindMerge.

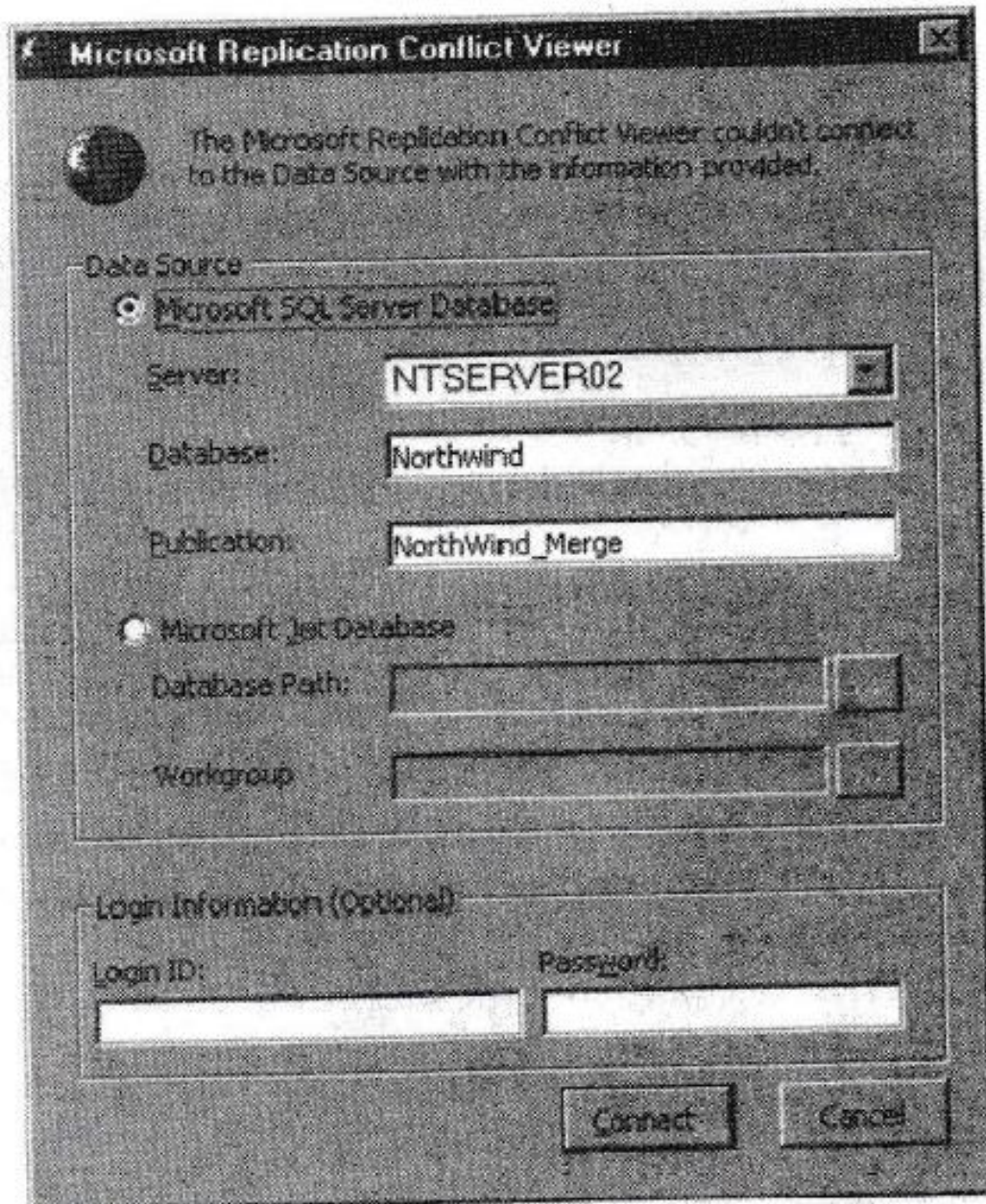
Untuk mengatasi konflik ini, gunakanlah Enterprise Manager. Pilihlah server yang digunakan, lalu klik tombol mouse kanan pada database NwindMerge.

Pada menu yang muncul, klik pada All Task, lalu pilih View Replication Conflicts.



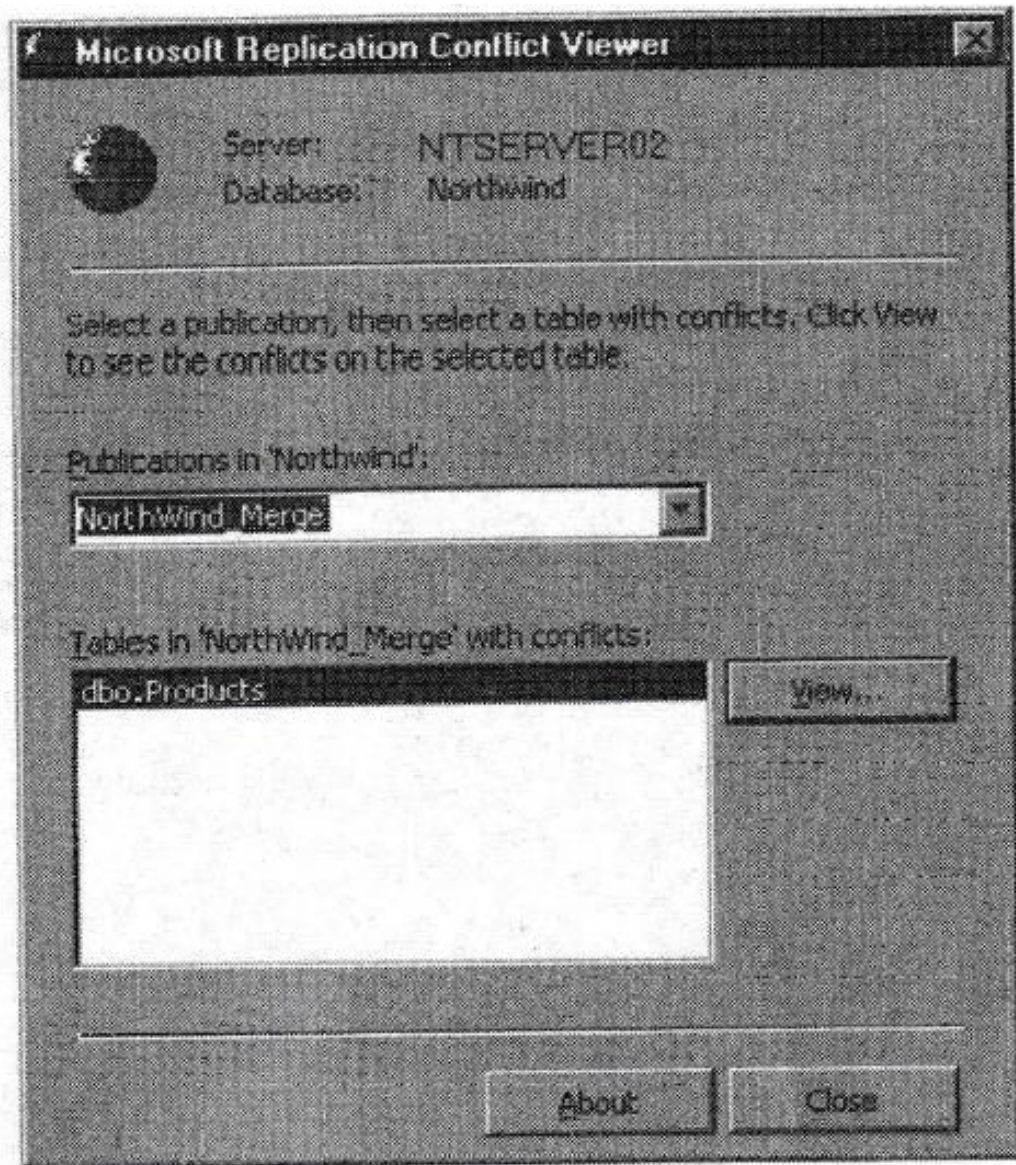
Gambar 12.26 View Replication Conflicts

Kotak dialog Conflict Manager akan muncul menampilkan data di dalam database. Tekan Connect.



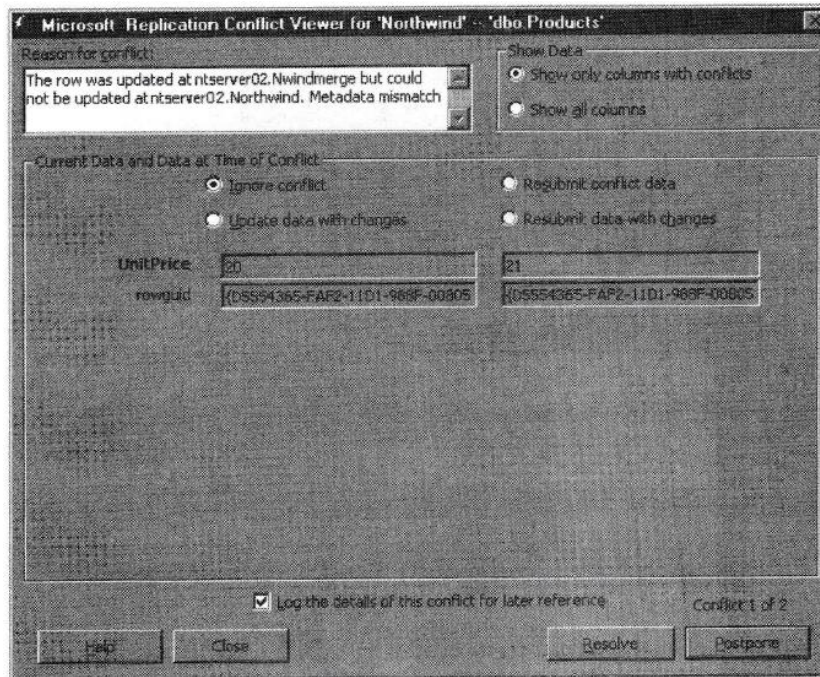
Gambar 12.27 View Replication Conflicts

Layar berikutnya menunjukkan bahwa tabel dbo.Products memiliki beberapa masalah.



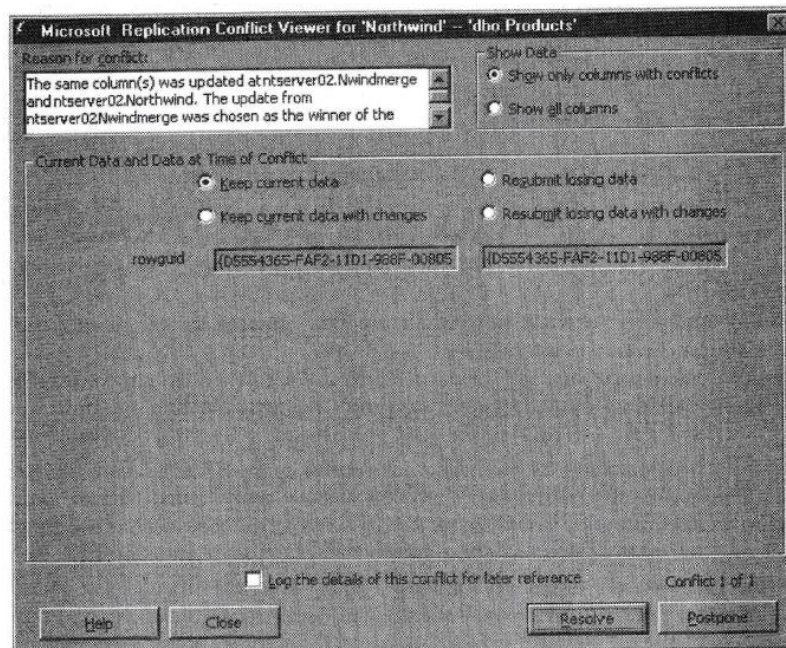
Gambar 12.28 Select Replication Conflicts

Untuk mengatasi masalah ini, klik tombol View. Layar berikutnya menampilkan alasan dari konflik ini.



Gambar 12.29 Hasil Conflicts

Perhatikan bahwa penerbit memenangkan proses update. Kini kita bisa memilih siapa yang akan menang. Perubahan yang dilakukan akan direplikasikan ke seluruh topologi. Dalam hal ini, kita akan menyimpan perubahan yang dibuat oleh database penerbit. Untuk melakukan itu, tekanlah tombol Resolve. Perhatikanlah konflik pada daerah Reason for Conflict.



Gambar 12.30 Hasil Conflicts

Tutuplah kotak dialog dengan menekan tombol Close.

Kesimpulan

Seperti yang telah Anda lihat, replikasi hanya membutuhkan perencanaan dan mempertimbangkan bagaimana cara user menggunakan sistem agar dapat memberikan replikasi yang terbaik. Pada bab berikutnya, kita akan membahas mengenai data warehousing dan posisi mereka dalam sistem penunjang keputusan, salah satu kekuatan dari SQL Server.

BAB 13

Data Warehousing dan OLAP

Gudang data berfungsi sebagai dasar dari sistem pendukung keputusan. Gudang data dirancang untuk memecahkan beberapa masalah yang dihadapi ketika organisasi mencoba melakukan analisis strategis menggunakan database yang sama yang melakukan pemrosesan transaksi online (OLTP).

Sistem OLTP tipikal memiliki karakteristik seperti sejumlah besar pengguna yang menambah dan mengubah data secara bersamaan. Database mewakili status bisnis tertentu, seperti B. sistem reservasi tiket pesawat, pada titik waktu tertentu.

Tetapi sejumlah besar data yang dikelola dalam beberapa sistem OLTP dapat membuat bisnis kewalahan. Saat basis data tumbuh dengan data yang semakin kompleks, waktu respons dapat berkurang dengan cepat karena persaingan untuk mendapatkan sumber daya yang tersedia..

Sistem OLTP yang umum memiliki banyak user yang menambah data ke dalam database, sementara sejumlah kecil user menghasilkan laporan dari database. Saat volume data semakin besar, laporan akan membutuhkan waktu lama untuk dihasilkan.

Tetapi apabila user perlu menganalisa data mereka, ada banyak masalah yang mungkin timbul:

- Apabila user tidak memahami hubungan yang kompleks antartabel, mereka tidak bisa menghasilkan query adhoc.
- Karena database aplikasi bisa tersegmentasi (terpecah-pecah) di antara beberapa server, user akan sulit menemukan tabel.
- Batasan keamanan mencegah user mengakses data dengan tingkat detail yang mereka butuhkan.
- Administrator database mencegah query adhoc ke sistem OLTP, mencegah user melakukan query yang mungkin dapat mengurangi kinerja database produksi yang sangat penting .

Gudangan data dan pemrosesan analitik online (OLAP) memberikan kunci untuk memecahkan masalah ini.

Gudangan data adalah pendekatan penyimpanan data yang memindahkan sumber data yang heterogen (biasanya tersebar di beberapa basis data OLTP) ke dalam penyimpanan data

yang homogen dan terpisah. Gudang data menawarkan manfaat berikut bagi pengguna yang melakukan analisis:

- Data diatur dengan baik untuk kueri analitik, bukan pemrosesan transaksional.
- Perbedaan antar struktur data pada beberapa database yang heterogen dapat diatasi.
- Aturan transformasi data dapat diterapkan untuk validasi dan konsolidasi data saat mentransfer data dari database OLTP ke gudang data.
- Masalah keamanan dan kinerja dapat diselesaikan tanpa mengubah sistem produksi.

Akhirnya, perusahaan mungkin ingin mempertahankan repositori data yang lebih kecil dan lebih terorganisir berdasarkan topik, yang dikenal sebagai data mart. Tidak seperti gudang data, yang biasanya berisi semua data analitik perusahaan, penyimpanan data biasanya merupakan subkumpulan data perusahaan yang terkait dengan kelompok pengguna atau fungsi bisnis yang lebih kecil.

Apa Itu Data Warehouse?

Gudang data dapat mendukung aplikasi pendukung keputusan dan pemrosesan analitik online (OLAP) karena menyediakan informasi yang:

- Seragam dan konsisten
- Berfokus pada topik tertentu
- Bersifat historis
- Hanya bisa dibaca saja

Data yang Terkonsolidasi dan Konsisten

Gudang data menggabungkan data fungsional dari berbagai sumber dengan konvensi penamaan, dimensi, atribut fisik, dan semantik yang konsisten. Misalnya, banyak aplikasi organisasi menggunakan data serupa dalam format berbeda: tanggal bisa disimpan dalam format Julian atau Gregorian, data true/false bisa direpresentasikan dalam satu/nol, on/off, true/false, atau positif /negatif. Aplikasi yang berbeda juga bisa menggunakan istilah-istilah yang berbeda untuk menjelaskan tipe data yang sama. Sebuah aplikasi bisa menggunakan istilah "balance", sementara aplikasi lain akan menggunakan istilah "total amount" untuk merepresentasikan jumlah uang yang terdapat di dalam rekening bank.

Data harus disimpan di gudang data dalam format berpemilik yang dapat diterima dan

disetujui oleh analis bisnis secara independen dari sumber operasional eksternal. Integrasi ini memungkinkan data perusahaan seperti mainframe lawas, data spreadsheet, atau bahkan data dari web disatukan dalam gudang data dan direferensikan silang secara efisien, memberikan pemahaman bisnis yang lebih baik kepada analis.

Data yang Berorientasi pada Topik Tertentu

Sumber data operasional dari seluruh organisasi cenderung mengandung sejumlah besar data mengenai beberapa fungsi-fungsi yang berhubungan dengan bisnis, seperti misalnya record klien, informasi mengenai produk, dan sebagainya. Tetapi sebagian besar dari informasi ini saling bercampur dengan data yang tidak penting dengan laporan bisnis atau eksekutif, dan diorganisir dalam cara yang bisa membuat proses query data menjadi sulit. Data warehouse mengorganisir informasi bisnis utama saja dari sumber operasional, jadi bisa dengan mudah tersedia bagi para analis bisnis .

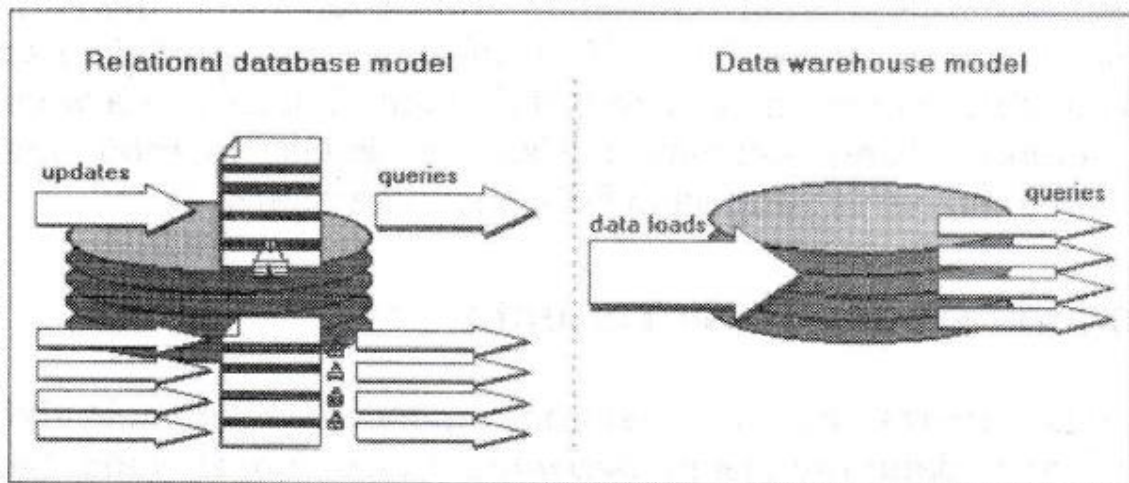
Data yang Bersifat Historis

Data sistem OLTP merepresentasikan nilai saat ini pada saat tertentu. Sistem OLTP seringkali hanya mengandung data saat ini. Sebagai contoh, aplikasi entri pemesanan barang selalu menampilkan nilai inventori saat ini; ia tidak menampilkan inventori kemarin atau dua jam yang lalu. Query kepada inventori yang dilakukan pada masa mendatang bisa memberikan jawaban yang berbeda.

Tetapi, data yang disimpan dalam data warehouse bersifat akurat pada titik waktu tertentu di masa lalu, karena data yang disimpan mewakili informasi historis. Data dalam data warehouse biasanya mewakili data pada periode waktu tertentu, mungkin 10 tahun atau lebih. Sebenarnya, data warehouse menyimpan snapshot data operasional bisnis, yang dihasilkan pada periode waktu tertentu. Data ini akurat untuk waktu tertentu dan biasanya tidak berubah.

Data Read-Only

Karena data yang disimpan di dalam data warehouse mewakili suatu waktu tertentu, operasi penghapusan, penyisipan, dan pembaharuan (kecuali untuk mereka yang terlibat di dalam proses loading data) tidak bisa diterapkan pada data warehouse. Setelah data dipindahkan ke gudang data, data biasanya tidak berubah kecuali datanya salah. Biasanya satu-satunya operasi yang terjadi pada data warehouse setelah dikonfigurasi adalah me-load query ke dalam data.



Karena data tidak berubah setelah di-load, perancangan data warehouse bisa dioptimalkan di dalam query melalui penggunaan indeks, data yang telah dikalkulasi sebelumnya, serta denormalisasi dari database fisik. Apabila data tidak perlu diubah, sarana OLAP bisa digunakan untuk mengelola query yang berhubungan dengan data statis di dalam warehouse dan data dinamis. Sebagai contoh, sebuah tim yang bekerja pada anggaran perusahaan bisa menggunakan data pada data warehouse untuk mendapatkan angka-angka dari tahun sebelumnya bersamaan dengan data dinamis, seperti misalnya anggaran dan prakiraan untuk tahun saat ini.

Data Warehousing dan OLAP

Meskipun istilah "gudang data" dan "pemrosesan analitik online (OLAP)" kadang-kadang digunakan secara bergantian, mereka adalah komponen sistem yang terpisah, sering disebut sebagai komponen pendukung keputusan.

Gudang data adalah database yang berisi informasi yang biasanya mewakili sejarah bisnis suatu organisasi. Data historis di gudang data digunakan dalam aktivitas analitis yang mendukung keputusan bisnis di berbagai tingkatan. Data di gudang data diatur untuk mendukung analitik, bukan pemrosesan transaksi real-time seperti dalam sistem pemrosesan transaksi online (OLTP).

OLAP adalah teknik yang memproses informasi dalam gudang data dalam struktur multidimensi dan memberikan jawaban cepat untuk pertanyaan analitis yang kompleks. Tujuan OLAP adalah mengatur sejumlah besar data sedemikian rupa sehingga dapat dengan cepat dianalisis dan dievaluasi menggunakan alat online seperti Microsoft PivotTables dan

presentasi grafis. Jawaban atas pertanyaan data historis seringkali membutuhkan pertanyaan lanjutan karena peneliti harus menjawab dan mengeksplorasi semua kemungkinan. Sistem OLAP menawarkan kecepatan dan fleksibilitas untuk mendukung analisis waktu nyata.

Memperkenalkan Microsoft Decision Support Services

Microsoft Decision support Services adalah server tingkat menengah baru untuk online analytical processing (OLAP). Sistem Decision Support Services menyediakan server tingkat tinggi yang membangun kubus-kubus data multidimensi untuk menganalisa dan menyediakan akses yang cepat bagi client untuk mengakses informasi pada cube. Microsoft PivotTable Service, client OLE DB 2.0, Microsoft Excel dan aplikasi pihak ketiga digunakan untuk mengambil informasi multidimensi dari server dan menyajikannya kepada pengguna.

Layanan Pendukung Keputusan mengatur data di gudang data ke dalam kubus multidimensi dengan data ringkasan yang telah dihitung sebelumnya untuk memberikan jawaban cepat atas pertanyaan analisis kompleks.

Di antara sumber daya utama dari Decision Support Services adalah sebagai berikut:

- Kemudahan penggunaan yang sangat tinggi, yang disediakan oleh antarmuka user dan wizard.
- Model data yang fleksibel dan tangguh untuk definisi dan penyimpanan kubus.
- Solusi yang terotomatisasi untuk mengatasi "sindrom ledakan data" yang mengalahkan teknologi OLAP tradisional.
- Arsitektur yang memiliki skalabilitas yang tinggi, mampu mendukung beberapa server data warehouse OLAP.
- Cache client/server yang pintar dan untuk memberikan jawaban yang cepat atas query dan mengurangi lalu lintas network
- API didukung dan didokumentasikan dengan luas, dan arsitektur yang terbuka untuk mendukung aplikasi client.

Kemudahan Penggunaan

Konsol Manajemen yang Terintegrasi. Microsoft Decision Support Services memiliki sarana administratif grafis yang disebut OLAP Manager. Ini adalah komponen snap-in dari Microsoft Management Console (MMC). Koneksi ini menyediakan struktur dan antarmuka yang umum untuk mendefinisikan, mengakses, dan mengelola server dan database OLAP.

Anda bisa menggunakan beberapa komponen snap-in dari MMC. Sebagai contoh, Anda bisa menginstal SQL Server Enterprise Manager dan komponen snap-in lainnya dengan OLAP Manager.

Cube Wizard

Anda bisa membuat semua struktur yang diperlukan untuk membuat kubus OLAP dengan wizard yang mudah digunakan. Wizard memandu Anda melalui proses membuat dan mengimplementasikan OLAP, mulai dari memetakan sumber data dan membuat dimensi data, sampai kepada mendefinisikan ukuran dan populasi kubus.

Cube Editor

Anda bisa menyunting struktur kubus yang sudah ada dan membuat struktur baru dengan menggunakan operasi drag-and-drop yang sederhana. Cube Editor adalah pelengkap dari Cube Wizard, di mana Anda bisa memeriksa kubus-kubus yang Anda buat dengan wizard atau untuk membuat kubus baru dengan cepat. Program penyunting ini mengintegrasikan tiga tampilan kubus yang berbeda:

- Panel Tree View menampilkan tampilan dimensi secara hirarkis, tingkat dimensi, ukuran, serta ukuran kubus yang telah dihitung.
- Panel Schema menampilkan tabel-tabel yang sesungguhnya dari data warehouse, tabel dimensi, serta hubungan antartabel yang mendefinisikan skema kubus. Anda bisa mengubah skema kubus pada panel ini.
- Panel Browse menampilkan data di dalam kubus Anda. Anda bisa berpindah ke tampilan data untuk melihat pekerjaan yang sedang berlangsung sambil melakukan perubahan pada struktur kubus. Data contoh akan dihasilkan sementara struktur kubus sedang diproses ; data sesungguhnya akan muncul apabila struktur kubus tidak diubah semenjak pemrosesan kubus yang terakhir.

Dimension Wizard

Anda bisa membuat dimensi bersama yang bisa digunakan oleh semua kubus atau dimensi pribadi yang hanya bisa digunakan oleh sebuah kubus dengan mudah dan cepat dengan Dimension Wizard. Anda bisa memetakan kolom-kolom dari tabel dimensi database ke dalam tingkat dimensi, atau Anda bisa menggunakan dimensi waktu untuk membuat

beberapa dimensi waktu berdasarkan pada satu data dan kolom jam di dalam database. Anda bisa menggunakan Dimension Wizard dengan skema datar (tabel dimensi tunggal) atau skema bintang atau butiran salju (tabel multidimensi).

Menjelajahi Data Kubus

Anda bisa melihat data kubus tanpa meninggalkan OLAP Manager. Anda tidak perlu berpindah ke aplikasi lain untuk melihat proyek kubus Anda.

Usage Analysis Wizard

Usage Analysis Wizard membantu Anda memahami bagaimana kubus digunakan, menampilkan informasi yang terdapat di dalam query, seperti misalnya data, user, waktu jawaban, serta frekuensi di dalam format tabular dan grafis.

Panduan Pengoptimalan Berbasis Penggunaan

Panduan Pengoptimalan Berbasis Penggunaan digunakan untuk menyesuaikan kinerja kubus berdasarkan cara pengguna menggunakan kubus. Anda bisa memandu wizard untuk membuat data koleksi untuk meningkatkan kinerja, berdasarkan kepada kombinasi dari pemakai, banyaknya query dijalankan, waktu jawa ban untuk query, media penyimpanan di mana data terletak, atau kisaran data.

Materi Tutorial dan Umum

Tutorial online dapat membantu Anda menguasai OLAP Manager dalam satu jam. Didesain untuk pengguna OLAP pemula dan lanjutan, tutorial ini akan memandu Anda melalui langkah-langkah membuat kubus sederhana serta operasi yang lebih lanjut seperti membuat partisi dan kubus virtual. Tutorial ini adalah alat yang hebat untuk mempelajari lebih lanjut tentang OLAP, fitur-fiturnya, dan sumber daya OLAP Manager.

Anda juga bisa menemukan informasi mengenai OLAP dan Decision Support Services pada panel kanan dari OLAP Manager.

Data Storage and Aggregation Wizard

Wizard ini membuat koleksi yang sesuai untuk penggunaan kubus Anda dan menentukan media penyimpanan informasi kubus. Pilihan yang tersedia pada wizard ini mengizinkan Anda mengoptimalkan pertukaran antara waktu jawaban dan kebutuhan

penyimpanan, sesuai dengan kebutuhan aplikasi dan user Anda.

Virtual Cube Wizard

Virtual Cube Wizard menghubungkan kubus-kubus serta memilih dimensi dan ukuran dari kubus-kubus tersebut untuk membuat sebuah kubus virtual. Kubus virtual juga mengizinkan sebuah query diarahkan ke beberapa kubus, termasuk kubus-kubus yang sedang dijalankan pada server yang berbeda. Kubus virtual kelihatan bagi user seperti kubus sebenarnya, tetapi tidak membutuhkan ruang penyimpanan tambahan. Kubus ini serupa dengan view dari database relasional yang menghubungkan tabel-tabel.

OLE DB Data Source Locator

Decision Support Services menggunakan komponen Microsoft Data Source Locator untuk membantu di dalam pemilihan sumber data OLE DB atau ODBC.

Model Data yang Fleksibel

Pilihan Penyimpanan Data yang Banyak. Microsoft Decision Support Services mengizinkan Anda menentukan metode penyimpanan yang paling sesuai dengan sistem Anda:

- MOLAP: data dasar kubus disimpan bersama-sama dengan data koleksi di dalam struktur multidimensi. Penyimpanan MOLAP menyediakan kinerja serta kompresi data yang bagus.
- ROLAP: data dasar kubus disimpan bersama-sama dengan data koleksi di dalam database relasional. Penyimpanan ROLAP mengizinkan Anda memaksimalkan investasi Anda pada teknologi database relasional serta sarana untuk mengelola data perusahaan.
- HOLAP: data dasar kubus disimpan di dalam database relasional; data koleksi disimpan di dalam struktur multidimensi dengan kinerja tinggi. Pilihan tambahan untuk HOLAP mencakup kubus serta partisi virtual. Penyimpanan HOLAP menawarkan keuntungan yang ada pada MOLAP untuk data koleksi, tanpa membutuhkan data dasar yang detail.

Penyimpanan Berpartisi

Anda bisa mempartisi sebuah kubus logik dalam bagian-bagian yang dipisahkan secara

fisik. Anda bisa menyimpan setiap partisi dalam tampilan yang berbeda, pada lokasi fisik yang berbeda, dan dengan tingkat koleksi yang sesuai dengan data yang dipartisi. Hasilnya, Anda bisa mengatur kinerja dan karakteristik dari manajemen data di dalam sistem Anda.

Menggabungkan Partisi

Setelah mempartisi kubus, Anda bisa menggabungkan kembali partisi-partisi ke dalam satu kubus fisik. Salah satu penggunaan dari menggabungkan partisi adalah untuk memindahkan bagian dari data kubus ke media penyimpanan yang berbeda, sehingga data bisa lebih sedikit digunakan.

Kubus Virtual

Anda bisa menghubungkan kubus-kubus fisik ke dalam kubus virtual, sama seperti tabel-tabel bisa dihubungkan ke dalam view di dalam database relasional. Kubus virtual menyediakan akses ke dalam data yang terdapat pada kubus-kubus yang digabungkan, tanpa perlu membuat sebuah kubus baru. Cara ini mengizinkan Anda memelihara rancangan yang terbaik untuk setiap kubus.

Skalabilitas

Partisi

Anda bisa menyebarkan kubus pada beberapa server, membaginya ke dalam partisi-partisi. Lalu Microsoft Decision Support Services bisa mengambil data secara paralel untuk mendapatkan query. Partisi ini mengizinkan Anda mengelola strategi penyimpanan Anda, untuk meningkatkan ukuran dengan beberapa server dan meningkatkan kinerjanya.

Update Menaik

Sebuah kubus bisa di-update dengan memproses data yang sudah diubah saja, tidak perlu memproses seluruh tabel. Anda juga bisa melakukan update cepat pada kubus OLAP sementara mereka sedang digunakan.

Dukungan Untuk Oracle 7.3 dan 8.0

Oracle versi 7.3 dan 8.0 bisa digunakan sebagai sumber data warehouse dan tujuan untuk penyimpanan kubus-kubus ROLAP.

Skenario LAN, WAN, Internet, dan Komputer Bergerak

Manajemen cache yang pintar mengintegrasikan server analisa DSS ke dalam client Microsoft PivotTable Service, meminimalkan lalu lintas hubungan LAN dan WAN. PivotTable Service mengandung mesin kalkulasi multidimensi yang efisien untuk lebih meminimalkan lalu lintas network dan mengizinkan analisa data lokal multidimensi apabila client tidak terhubung ke server.

Kontrol Microsoft ActiveX, Active Server Page scripting, serta ADO API menyediakan solusi yang luas untuk query data OLAP pada Internet.

Keamanan Kubus

Akses ke dalam kubus didasarkan kepada keamanan Microsoft Windows NT dan dikelola melalui penugasan aturan-aturan ke dalam kubus .

Dukungan Windows NT untuk Platform DEC Alpha

Selain platform Intel, Windows NT for Digital Equipment Corporation Alpha mendukung semua teknologi client/server dari Decision Support Services.

Dukungan Client Untuk Windows 95 dan Windows 98

PivotTable Services yang dijalankan pada Microsoft Windows 95 dan Windows 98 mendukung aplikasi-aplikasi klien yang tersedia untuk platform ini.

Anggota Yang Dihitung

Anda bisa menggunakan Cube Editor untuk membuat ukuran serta anggota dimensi yang telah dihitung sebelumnya, mengabungkan ekspresi MDX, rumus matematik, serta fungsi buatan user. Calculated Member Builder menyediakan antarmuka untuk menambahkan model-model ekspresi MDS ke dalam ekspresi untuk anggota-anggota yang telah diperhitungkan. Sumber ini mengizinkan Anda mendefinisikan ukuran serta dimensi baru berdasarkan kepada sintaks ekspresi yang kaya yang lebih mudah digunakan.

Arsitektur Server

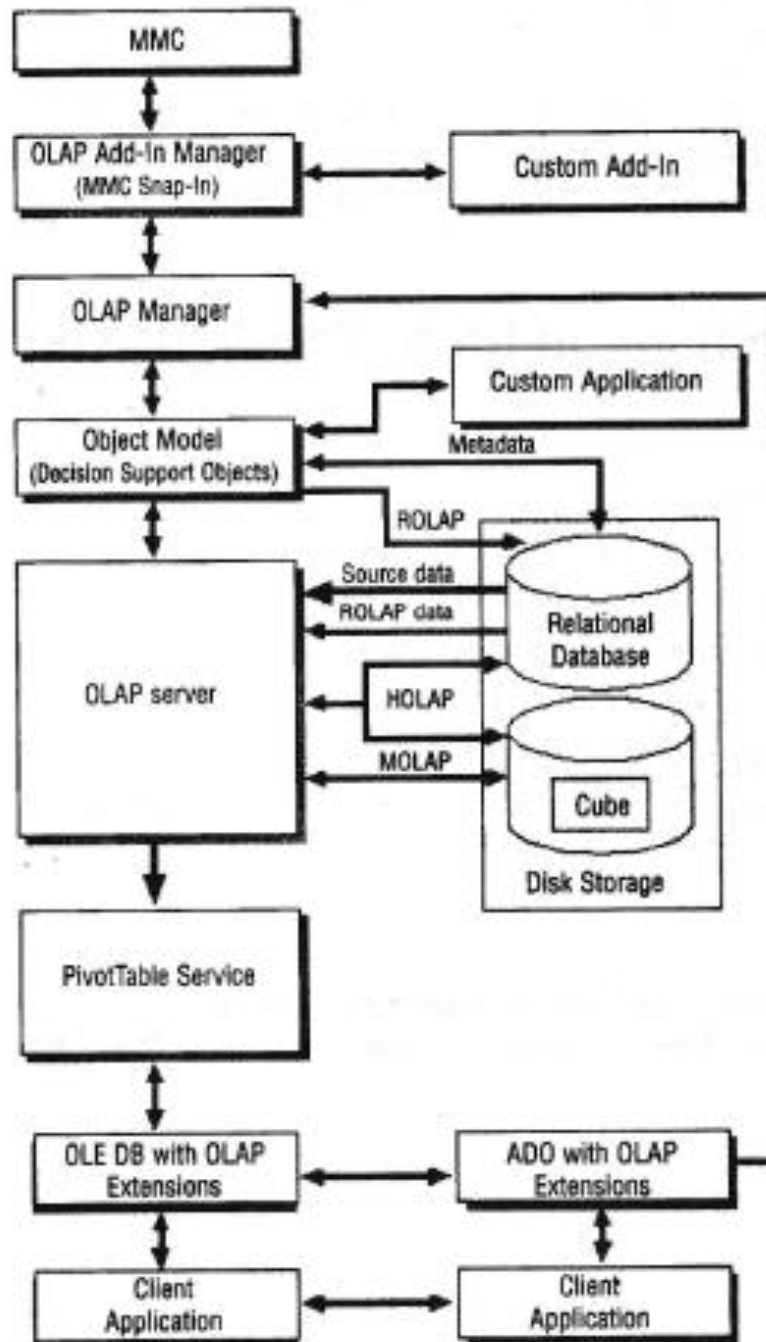
Microsoft Decision Support Services menyediakan kapasitas server untuk pembuatan dan pengelolaan data OLAP multidimensi serta untuk menyediakan data kepada client

melalui Microsoft PivotTable Service. Operasi server mencakup pembuatan kubus data multidimensi dari database relasional yang terdapat pada data warehouse, serta penyimpanan kubus-kubus di dalam struktur multidimensi, pada database relasional, dan gabungan dari kedua nya. Metadata dari struktur multidimensi dari sebuah kubus disimpan di dalam penyimpanan database relasional.

Antarmuka pengguna disediakan oleh Microsoft Management Console (MMC). Antarmuka pengguna ini juga disediakan untuk digunakan dalam aplikasi kustom menggunakan Microsoft Visual Basic 5.0 untuk berinteraksi dengan model objek yang mengontrol server dan dengan plugin antarmuka server.

Arsitektur server dasar ditunjukkan di bawah ini:

OLAP Services System Architecture
(with PivotTable Service)



Arsitektur Client

Microsoft PivotTable Service berkomunikasi dengan server DSS Analisis dan menyediakan antarmuka yang dapat digunakan oleh aplikasi client yang mengakses data OLAP di dalam server. Aplikasi client yang terhubung kepada PivotTable Service yang menggunakan antarmuka OLE DB 2.0 for C++ atau model objek ADO 2.0 untuk bahasa seperti Microsoft Visual Basic. PivotTable Service berkomunikasi dengan server melalui protokol pribadi.

PivotTable Service juga bisa membuat file-file kubus lokal yang mengandung data kubus di dalam server atau dari database relasional OLE DB. Kubus lokal bisa disimpan sebagai file-file multidimensi di dalam komputer client seperti kubus ROLAP pada database relasional, di mana mereka diambil. Kubus lokal bisa digunakan secara offline dengan PivotTable Service untuk analisa yang portabel.

Membuat dan Membangun Data Warehouse dan Sistem OLAP

Langkah-langkah untuk membuat data warehouse mencakup :

- Menentukan kebutuhan bisnis, user, dan teknisi
- Membuat dan membangun database
- Mengambil dan me-load data warehouse
- Membuat dan membangun database OLAP
- Membuat dan memproses data koleksi
- Melakukan query dan memelihara data warehouse serta data base OLAP

Kesimpulan

Bab ini memperkenalkan data warehousing-dan sistem OLAP. Bab berikutnya memberikan contoh praktis di dalam menggunakan OLAP Manager.

BAB 14

Contoh Praktis OLAP Manager

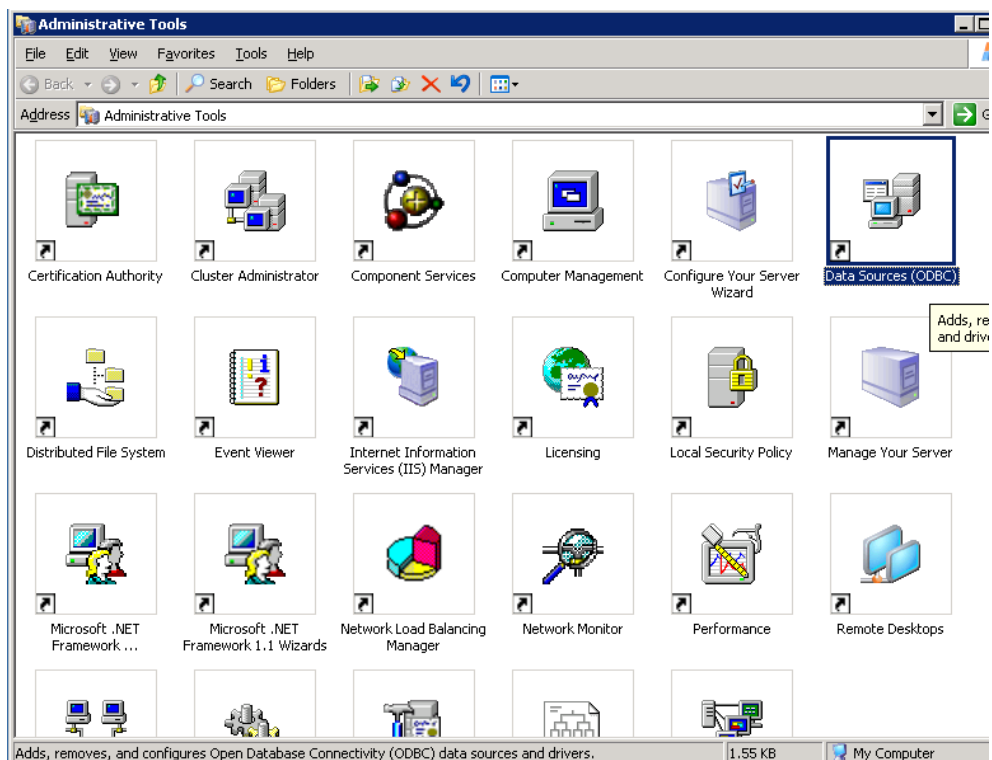
Pada bab ini, kita akan membuat sebuah kubus untuk melakukan query pada database FoodMart. Kubus ini akan dibuat berdasarkan kepada penggunaannya di dalam departemen pemasaran suatu perusahaan, yang ingin mengevaluasi penjualan dari beberapa sudut pandang. Sebelum membuat kubus itu sendiri, kita harus melakukan beberapa persiapan.

Latihan ini menggunakan database FoodMart yang terdapat pada folder FoodMart pada CD pelengkap, yang sudah harus diinstal pada komputer Anda. Ikutilah petunjuk-petunjuk pada Lampiran B untuk menginstal CD sebelum Anda memulai latihan ini.

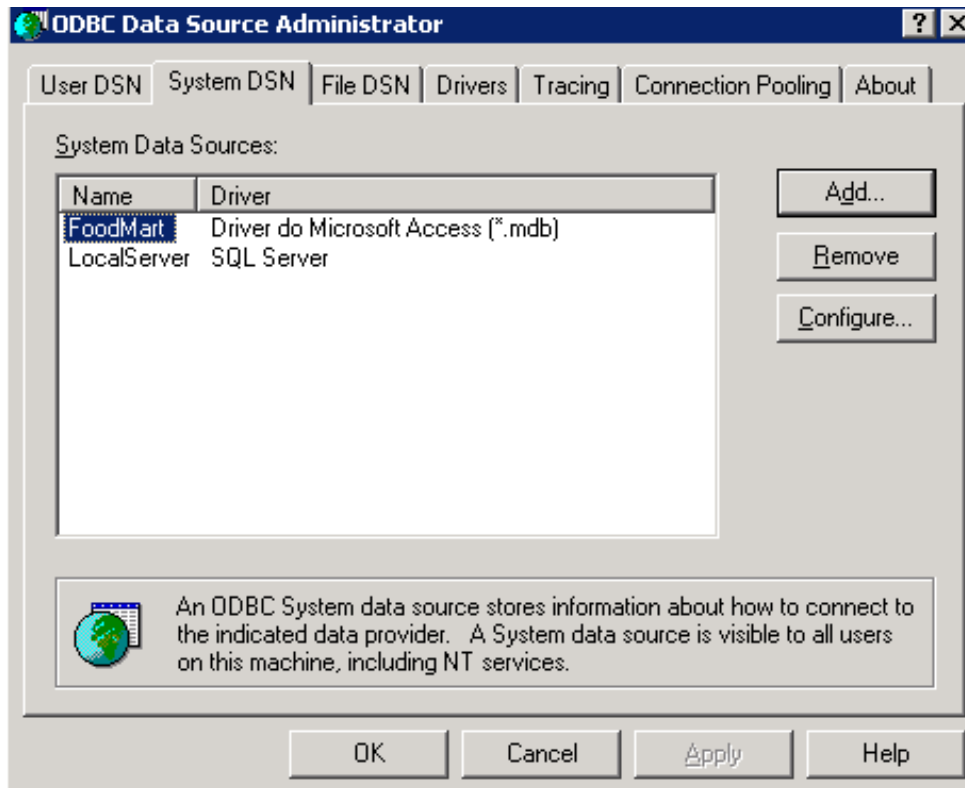
Mengkonfigurasi DSN

Sebelum melanjutkan, kita perlu mengkonfigurasi koneksi dengan sumber data yang akan kita gunakan pada latihan ini. Untuk melakukan itu, ikutilah langkah-langkah berikut ini:

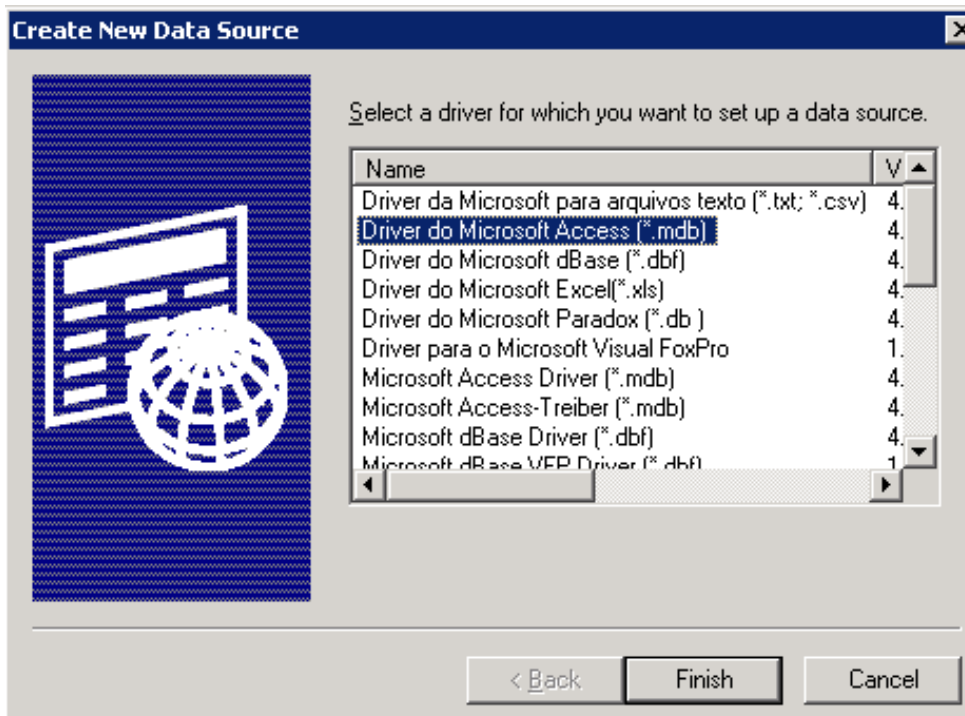
Dari tombol Start, aktifkanlah Control Panel dan klik ganda pada icon ODBC Data Sources seperti pada gambar di bawah ini.



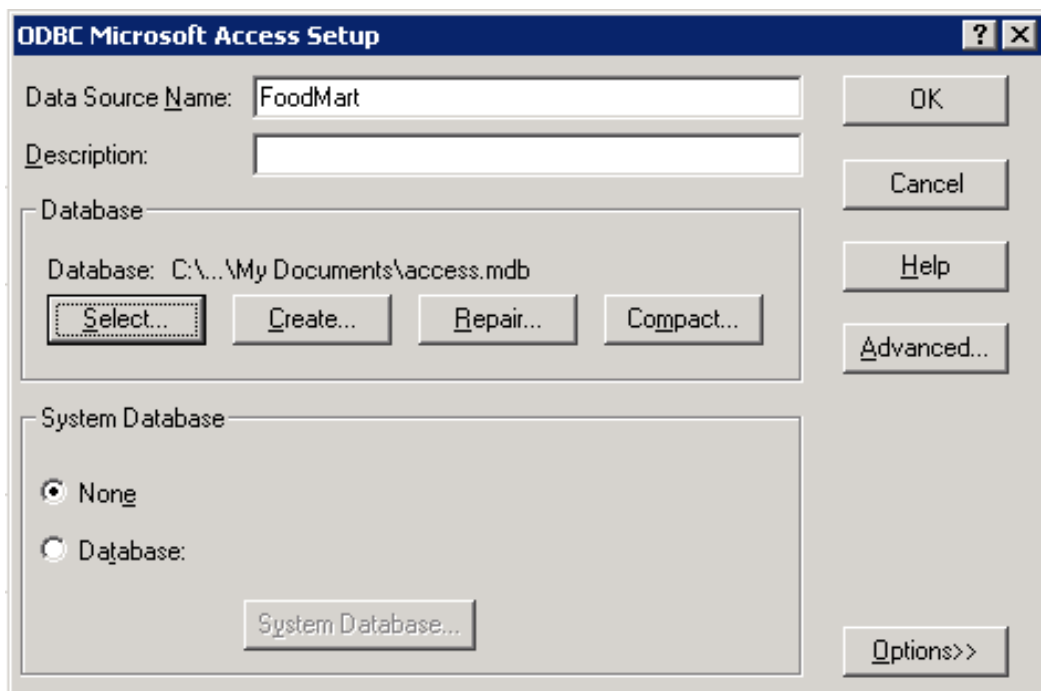
Lalu klik pada tab System DSN, pilihlah FoodMart, dan tekan Add.



Lalu pilihlah driver Access (*.mdb) yang ditampilkan dengan nama FoodMart.



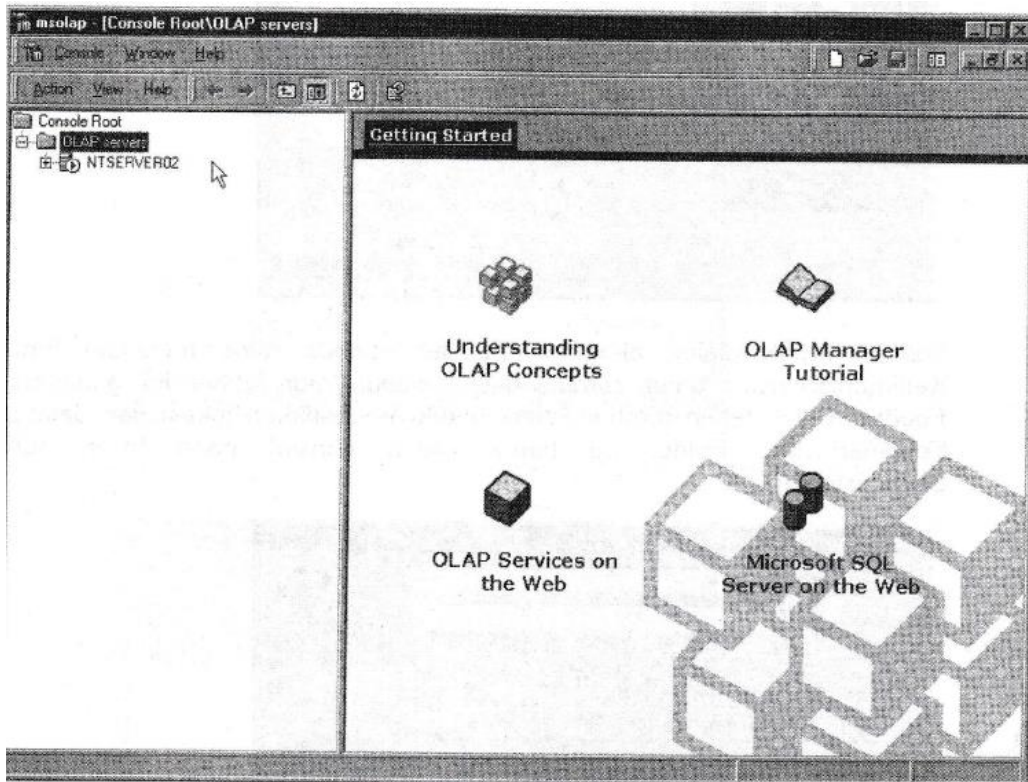
Sebuah kotak dialog akan ditampilkan apabila Anda menekan Finish. Ketikkanlah nama untuk sumber data tersebut. Pada latihan ini, gunakanlah Foodmart dan tekan tombol Select untuk mengetikkan lokasi dari database FoodMart.mdb. Folder ini harus sudah diinstal pada folder \OlapServices\sample.



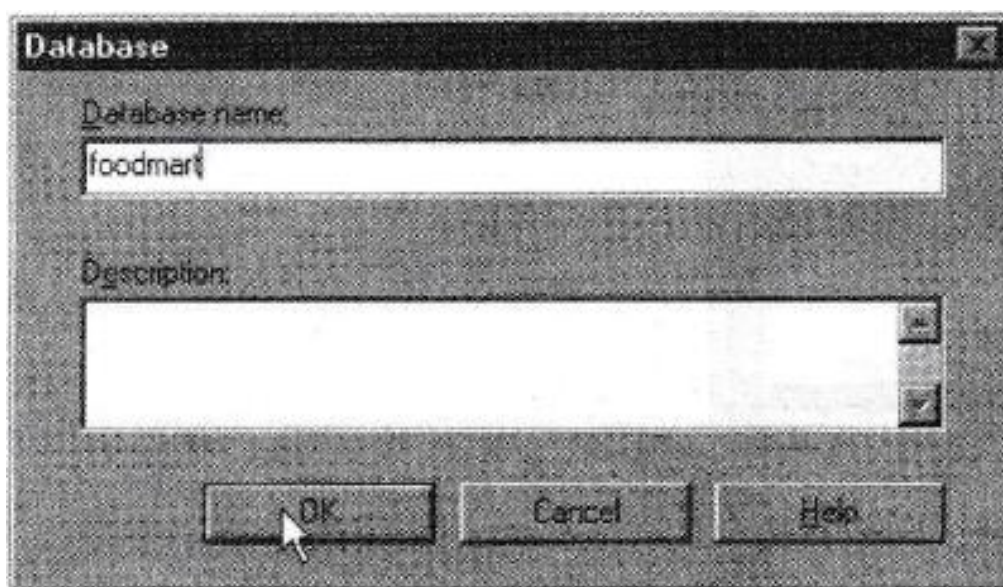
Tekan tombol OK pada kotak dialog ini untuk melengkapi pembuatan sumber data.

Menjalankan OLAP Manager

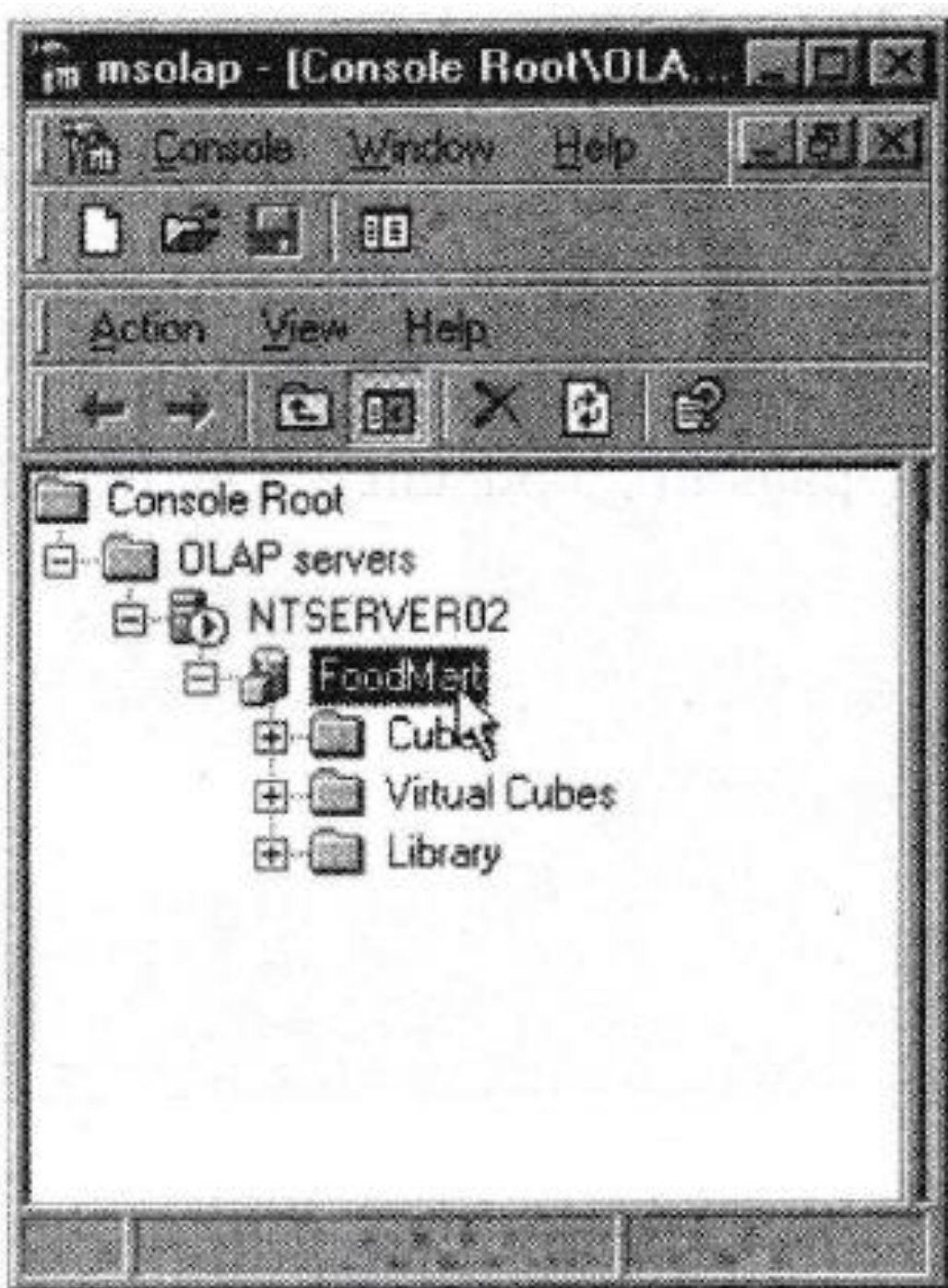
Untuk menjalankan OLAP Manager, pilih Start | Programs | Microsoft SQL Server | Olap Manager. Layar kerja dari OLAP Manager akan ditampilkan.



Sebelum kita memulai proses ini, kita perlu membuat sebuah struktur database baru dan menghubungkannya ke sumber data yang baru kita buat. Klik pada tanda plus (+) di sebelah folder OLAP Servers lalu klik pada server yang digunakan dengan tombol mouse kanan. Pada menu yang muncul, klik Connect. Lalu ketikkan nama Foodmart pada field Database Name pada kotak dialog yang muncul, lalu tekan tombol OK.



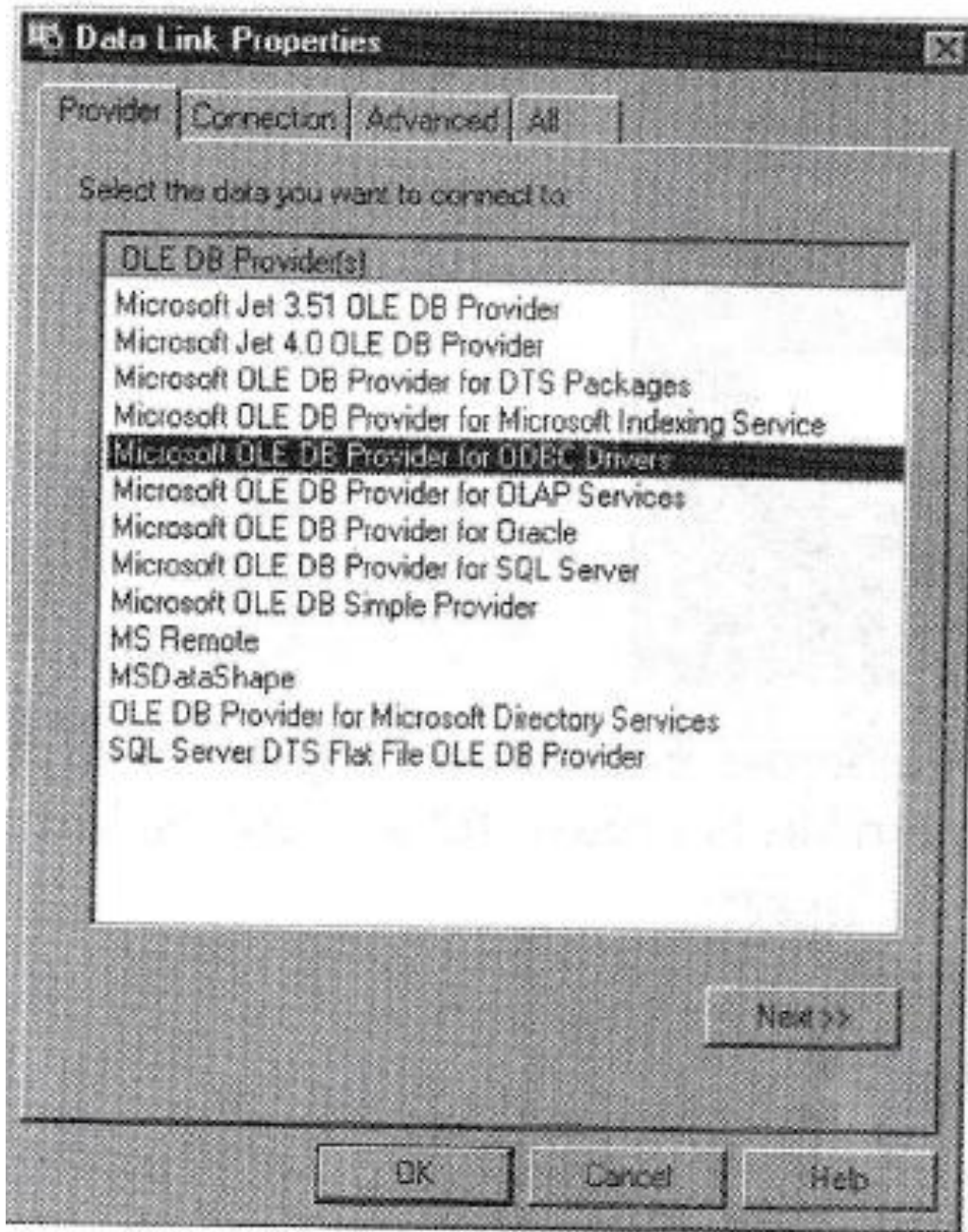
Apabila database FoodMart diperbesar dengan mengklik pada tanda plus, perhatikan bahwa database ini memiliki tiga buah folder: Cubes, Virtual Cubes, dan Library.



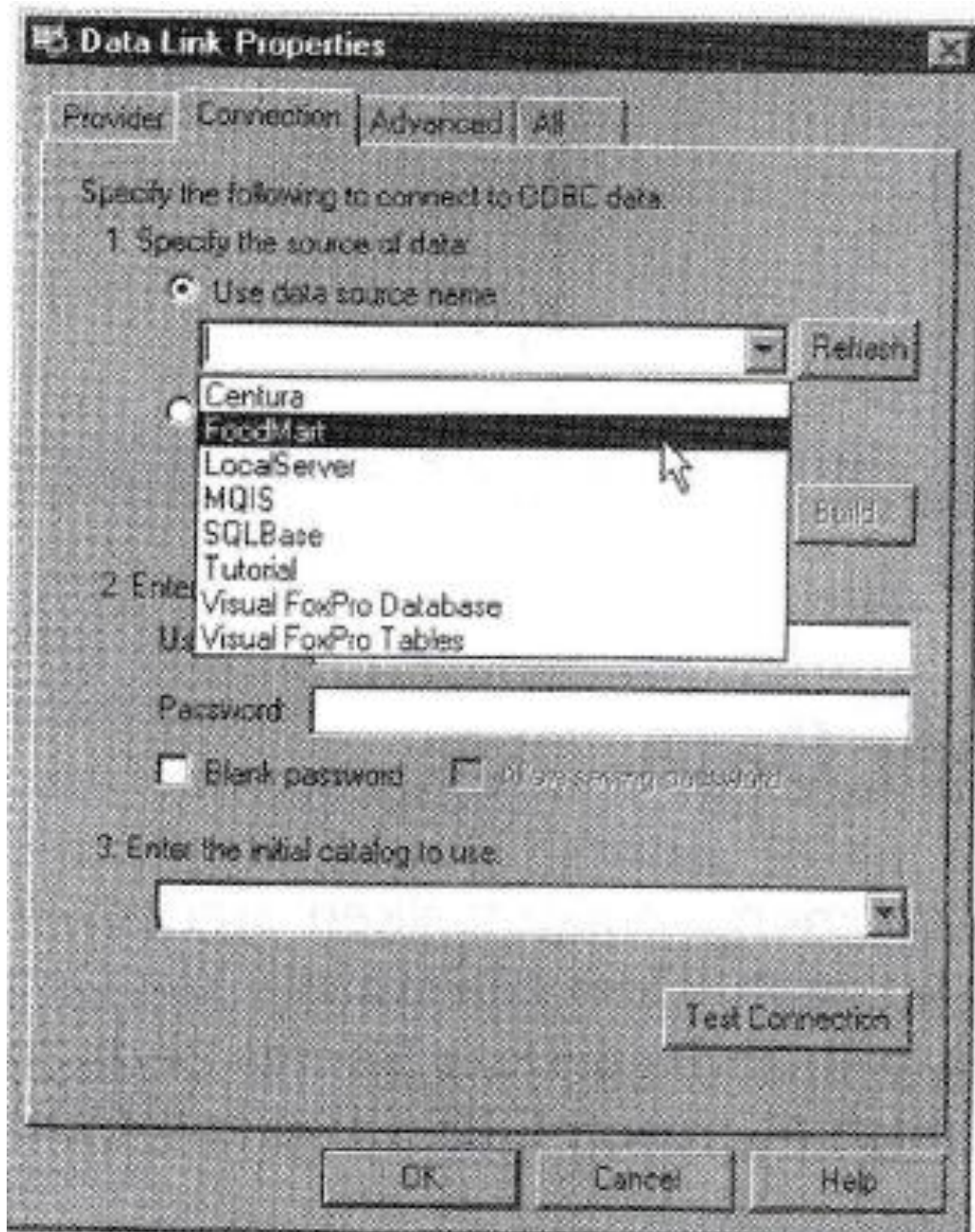
Mengkonfigurasi Sumber Data

Pada langkah berikutnya, kita akan mengkonfigurasi sumber data. Pada tiga tampilan dari OLAP Manager, bukalah folder Library lalu klik Data Sources dengan tombol mouse kanan. Lalu pilihlah New Data Source pada menu yang muncul. Pada layar berikutnya,

pilihlah Microsoft OLE DB untuk ODBC Drivers lalu tekan tombol Next.



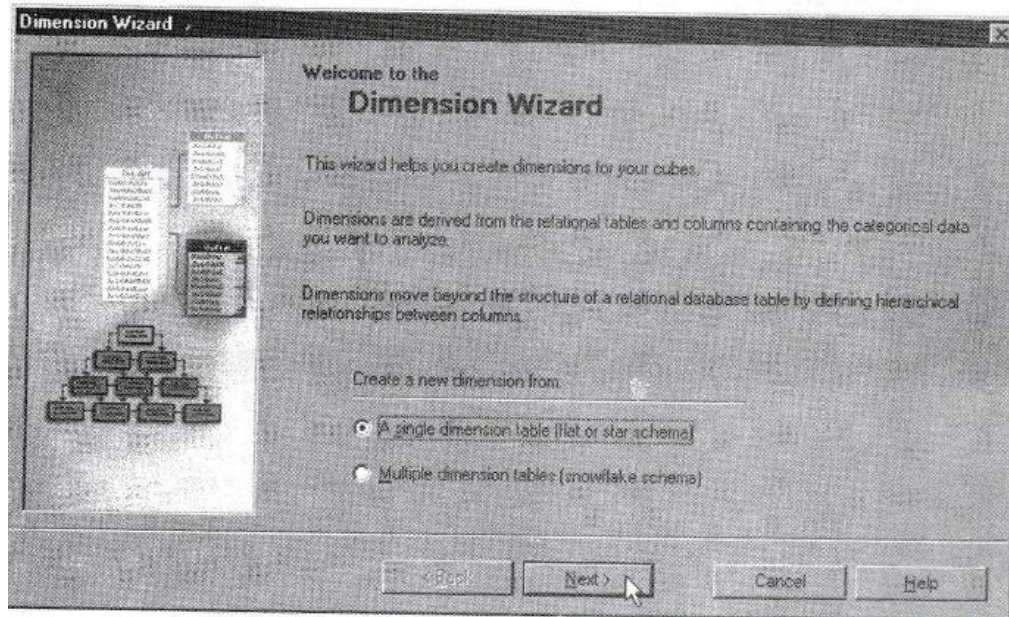
Lalu klik pada tab Connection dan pilihlah FoodMart pada field Use Data Source Name.



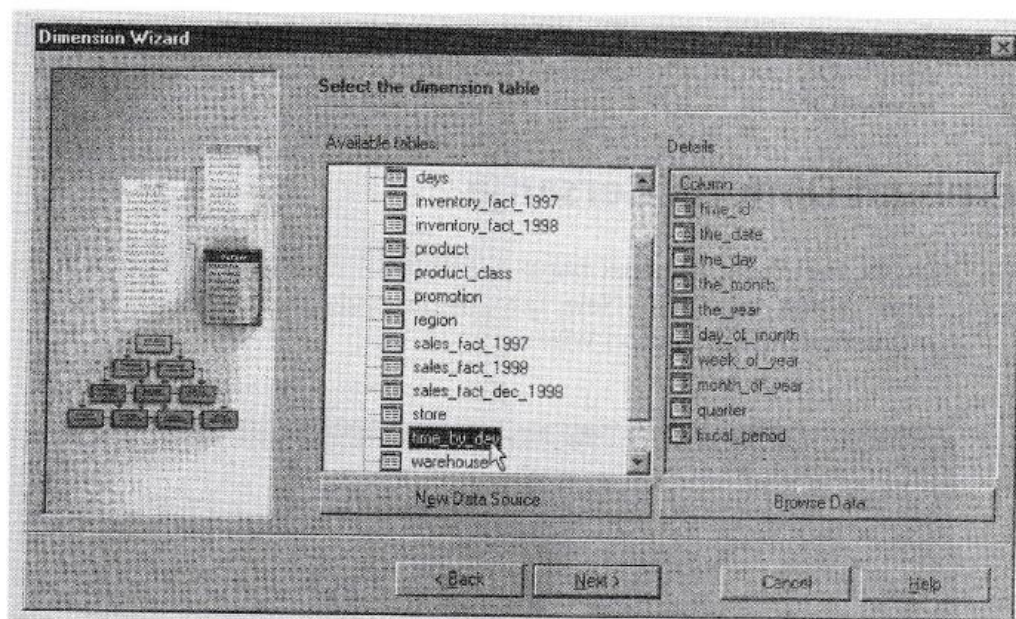
Setelah memilih item FoodMart, tekanlah tombol Test Connection; sebuah pesan OK akan ditampilkan. Lalu tutuplah kotak dialog dengan menekan tombol OK.

Membuat Dimensi Bersama

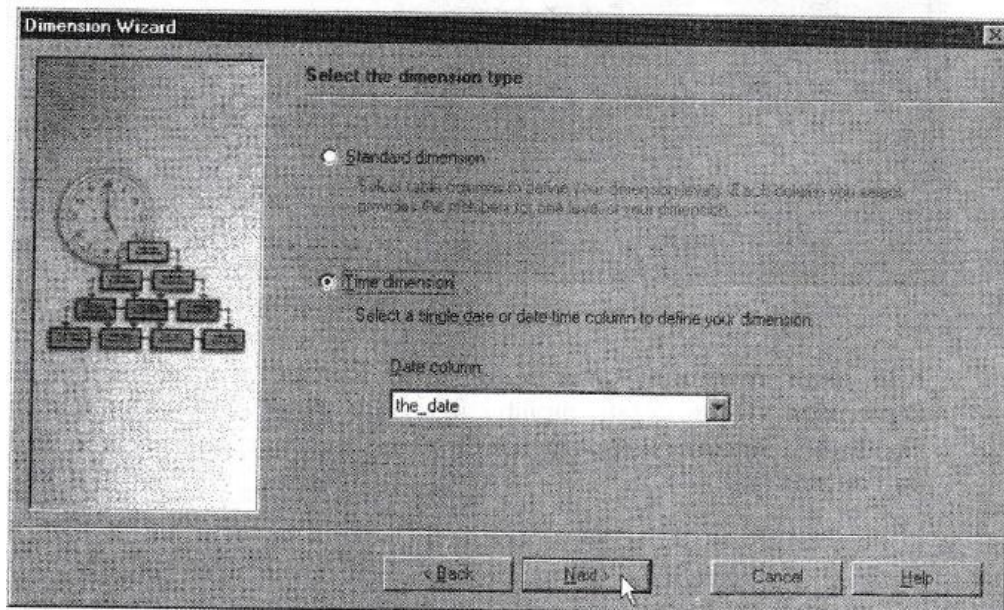
Buat dimensi baru dengan mengklik kanan folder "Common Dimensions" dan pilih "New Dimensions" dari menu yang muncul. Ini akan memunculkan wizard pengukuran.



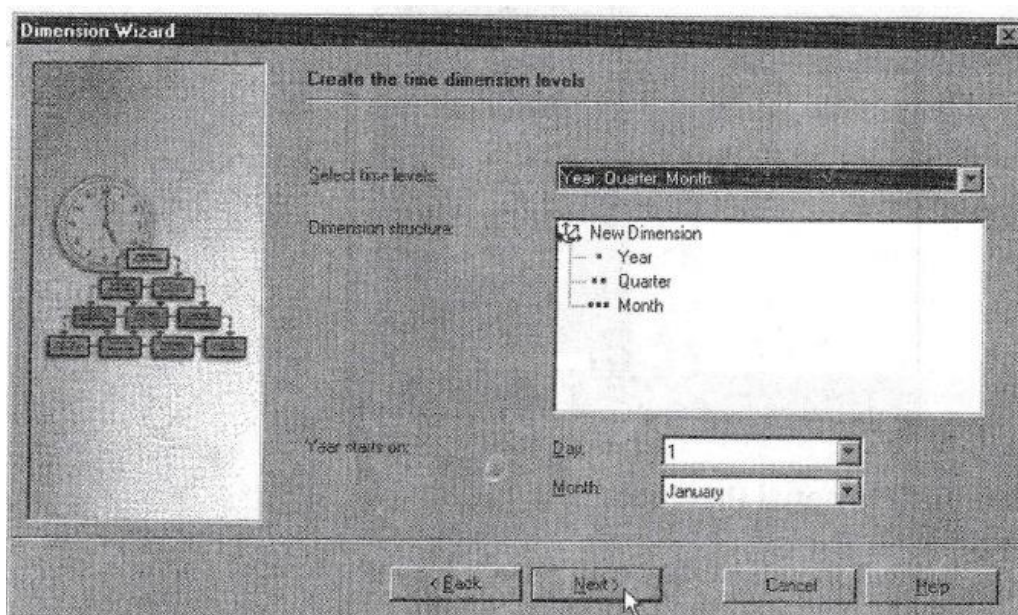
Kita akan membuat dimensi tunggal. Pastikan tombol di sebelah Single Dimension Table telah dipilih, lalu klik Next. Lalu klik pada database Foodmart. Setelah memperbesar database tersebut, pilihlah tabel `time_by_day`, dan tekan Next .



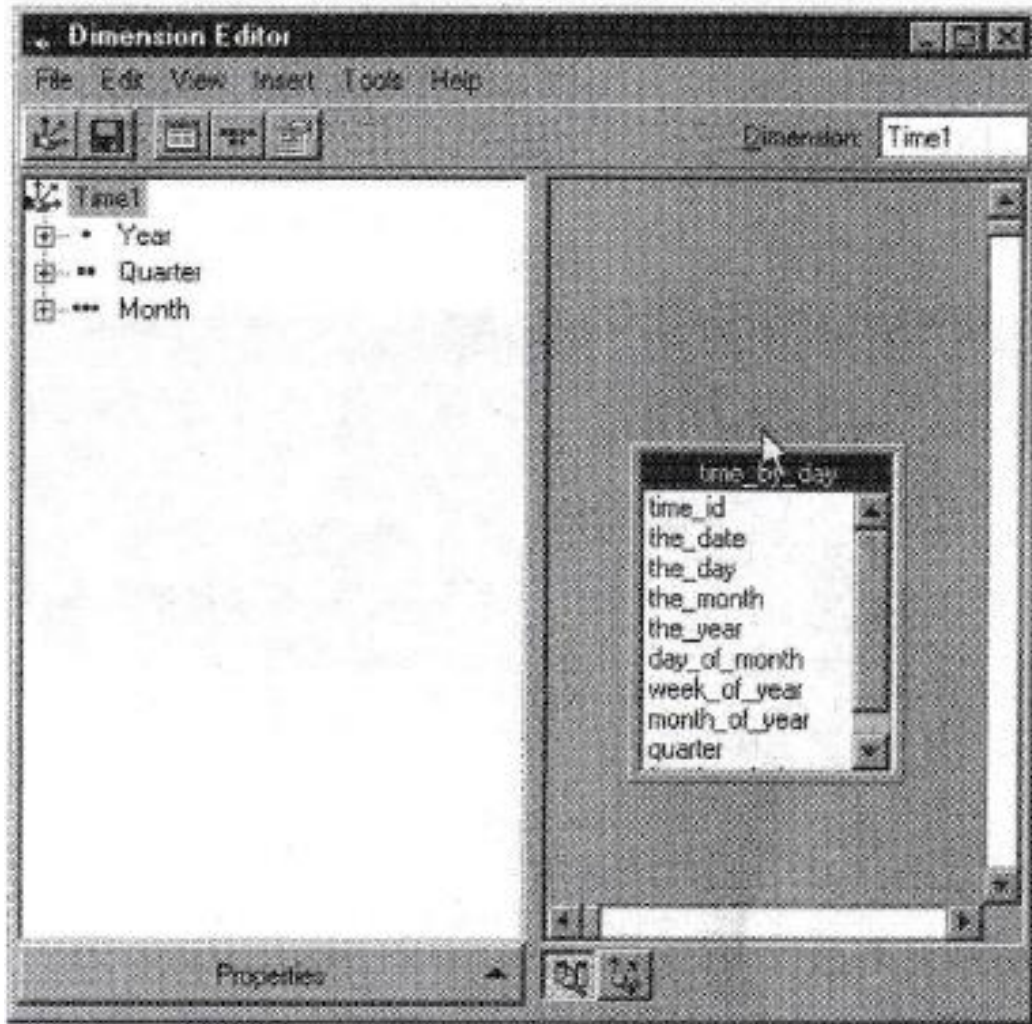
Kini kita akan mendefinisikan tipe dimensi. Pilihlah Time Dimension lalu tekan Next.



Di sini kita mendefinisikan tingkat dimensi waktu. Pada contoh ini, kita memilih Year, Quarter, Month, untuk membuat dimensi dengan subbagian untuk tahun, kuartal tahun, dan bulan, lalu tekan Next.

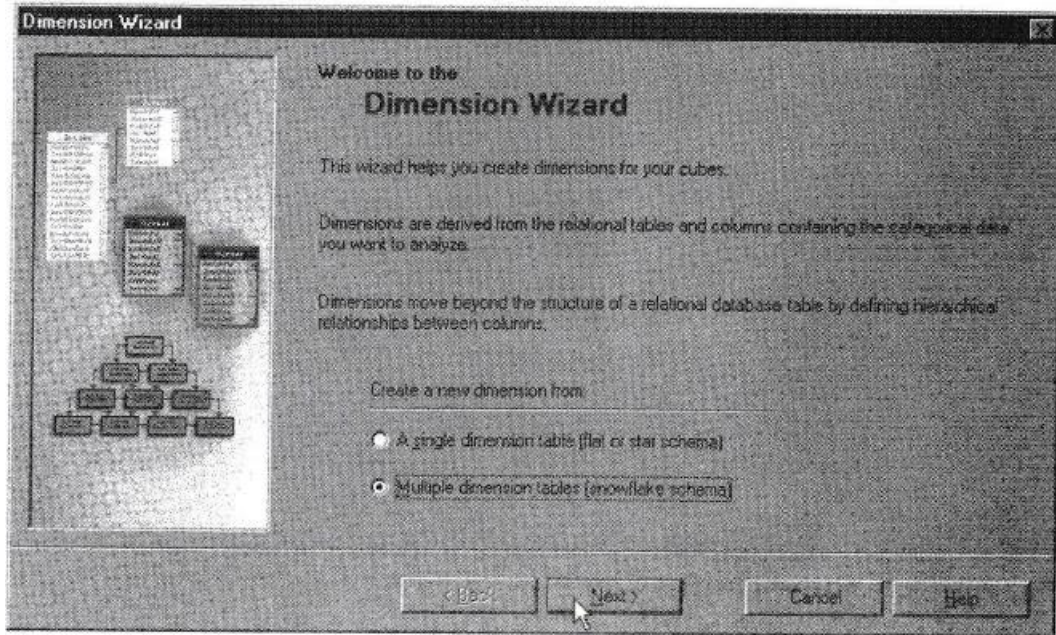


Lalu kita memasukkan nama dari dimensi pada langkah yang terakhir. Ketikkanlah Time1 lalu tekan tombol Finish. Wizard akan membuka Dimension Editor seperti ditunjukkan pada gambar berikut ini:

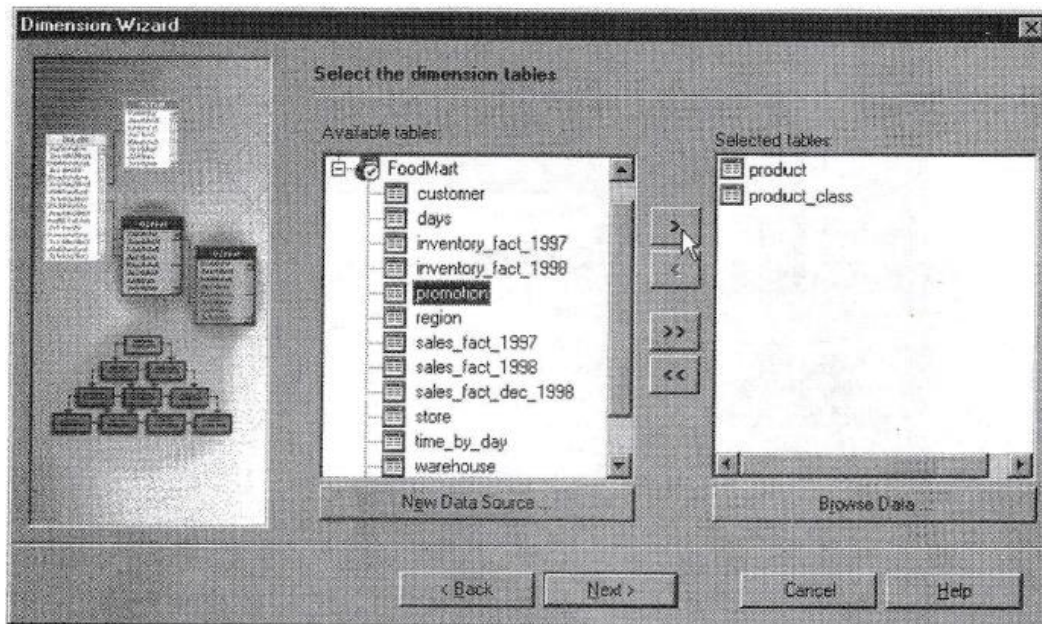


Membuat Dimensi Product

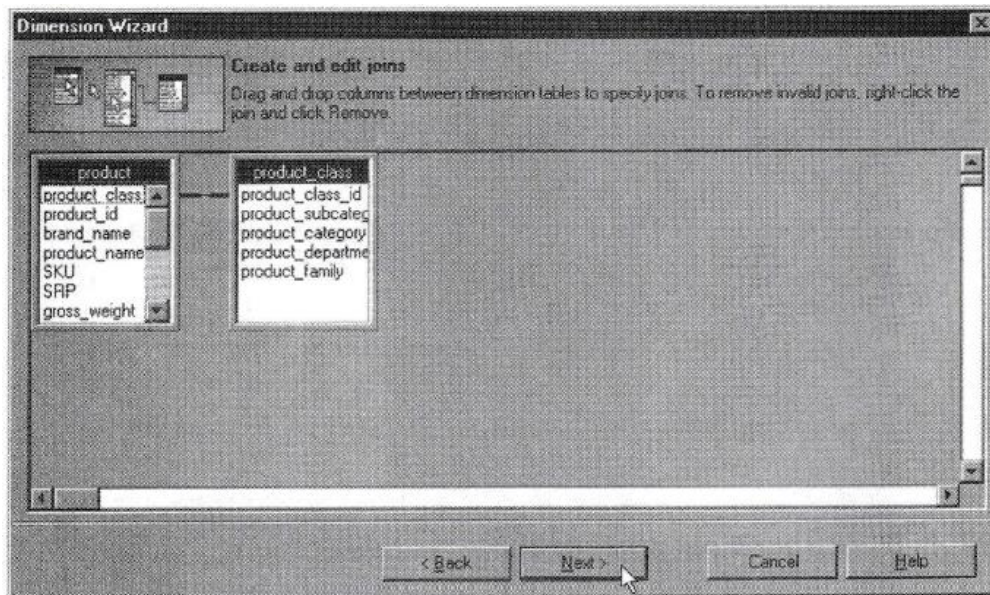
Untuk membuat dimensi Product, tekanlah tombol pertama pada toolbar. Ini akan mengaktifkan Dimension Wizard. Pilihlah Multiple Dimension Tables, lalu tekan tombol Next.



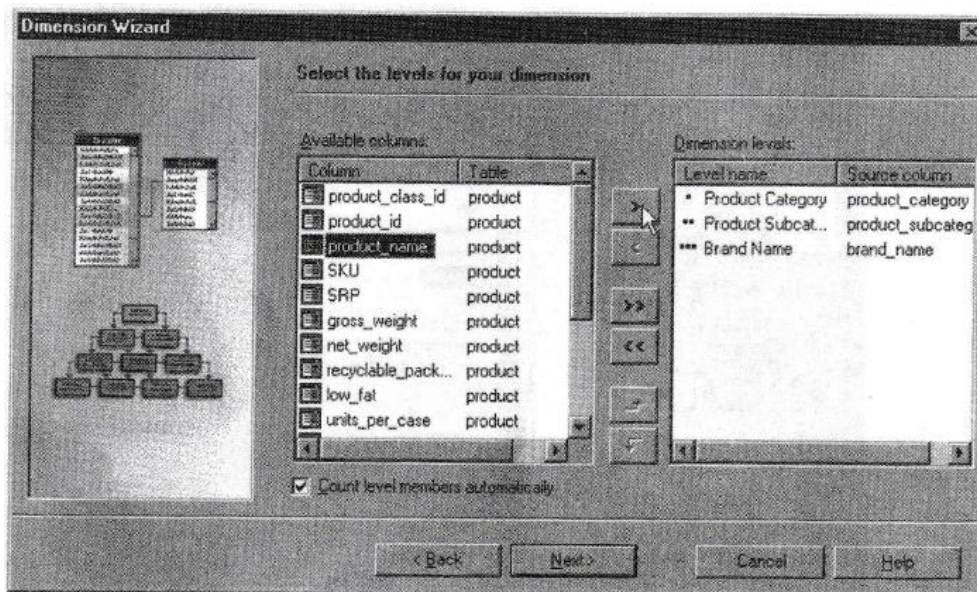
Kemudian pilihlah tabel Product dan Product_class dengan mengklik ganda tabel-tabel tersebut. Tekan Next .



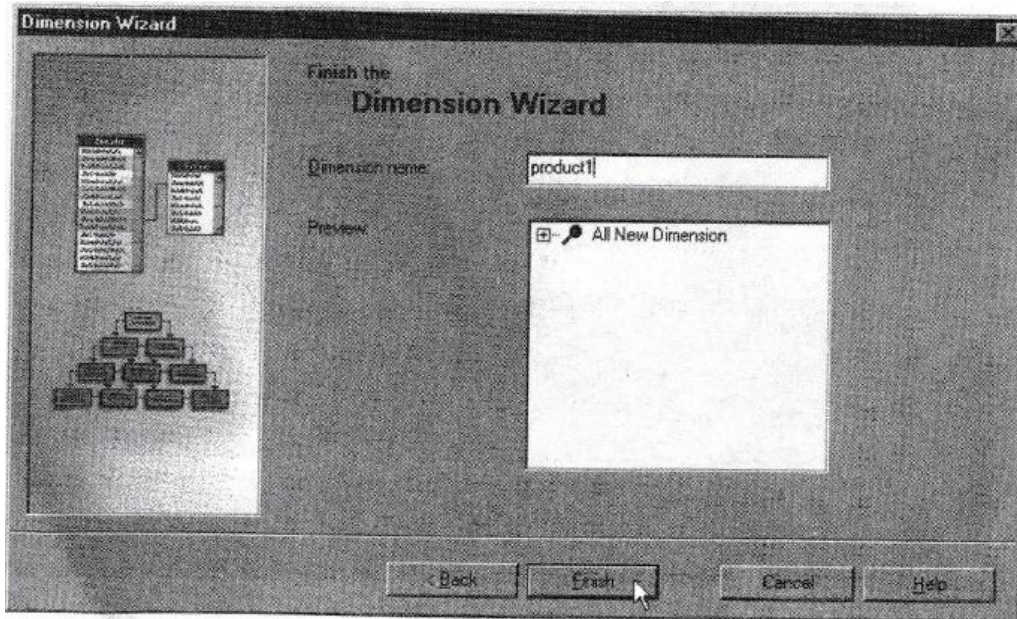
Kedua tabel tersebut ditampilkan dengan link yang ada di antara mereka. Tekan Next untuk melanjutkan.



Pilihlah tingkat dimensi yang akan dibentuk oleh kolom product_category, product_subcategory, dan brand_name, lalu tekan Next.



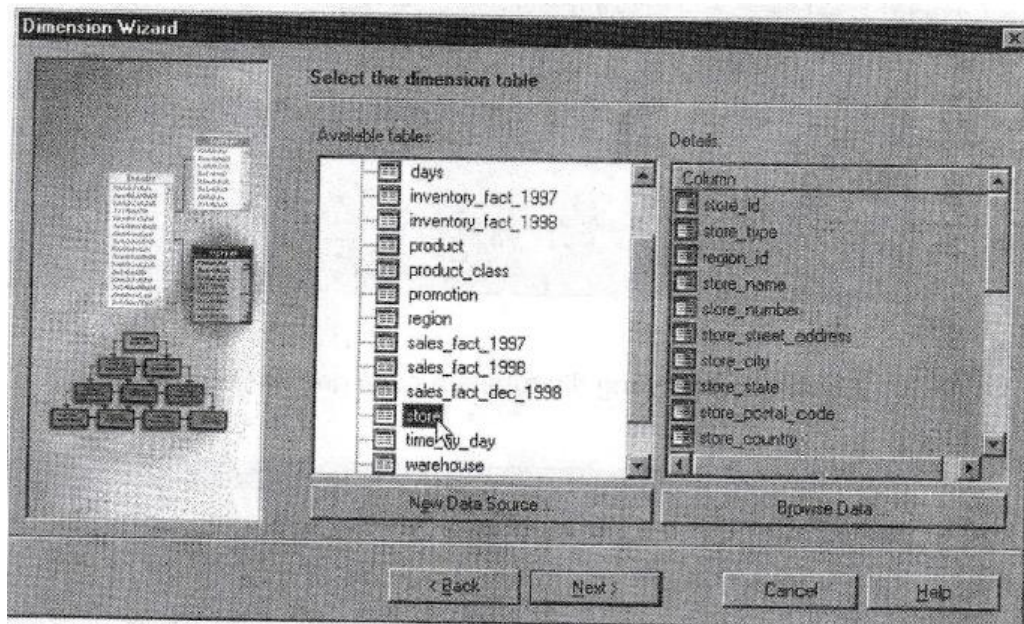
Pada langkah yang terakhir, ketikkanlah nama product1 untuk dimensi ini, lalu tekan tombol Finish.



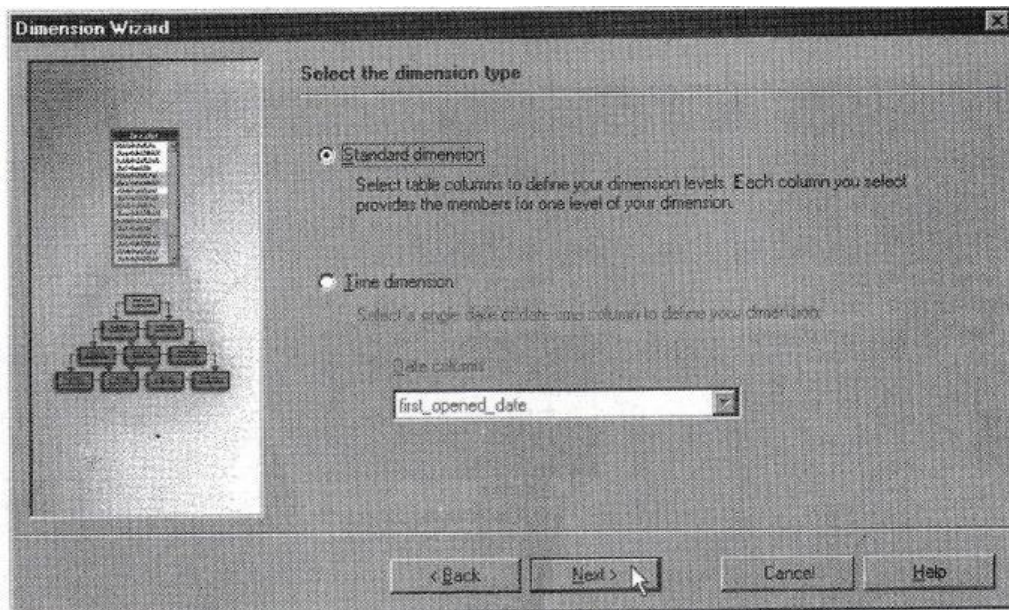
Membuat Dimensi Store

Ada sebuah dimensi lain di dalam kubus ini yang disebut dengan dimensi Store. Ikutilah langkah-langkah berikut ini untuk membuatnya:

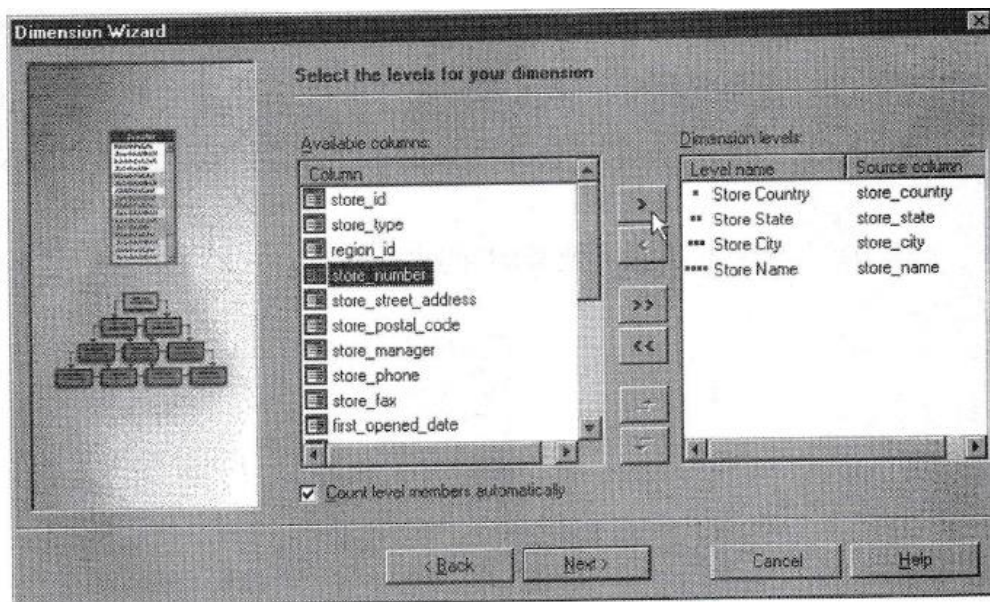
Pada layar pertama dari Dimension Wizard, pilihlah item Single Dimension, lalu tekan Next. Pilihlah database FoodMart dan tabel Store, lalu tekan Next.



Pada layar berikutnya, klik Next untuk tetap memilih Standard Dimension.

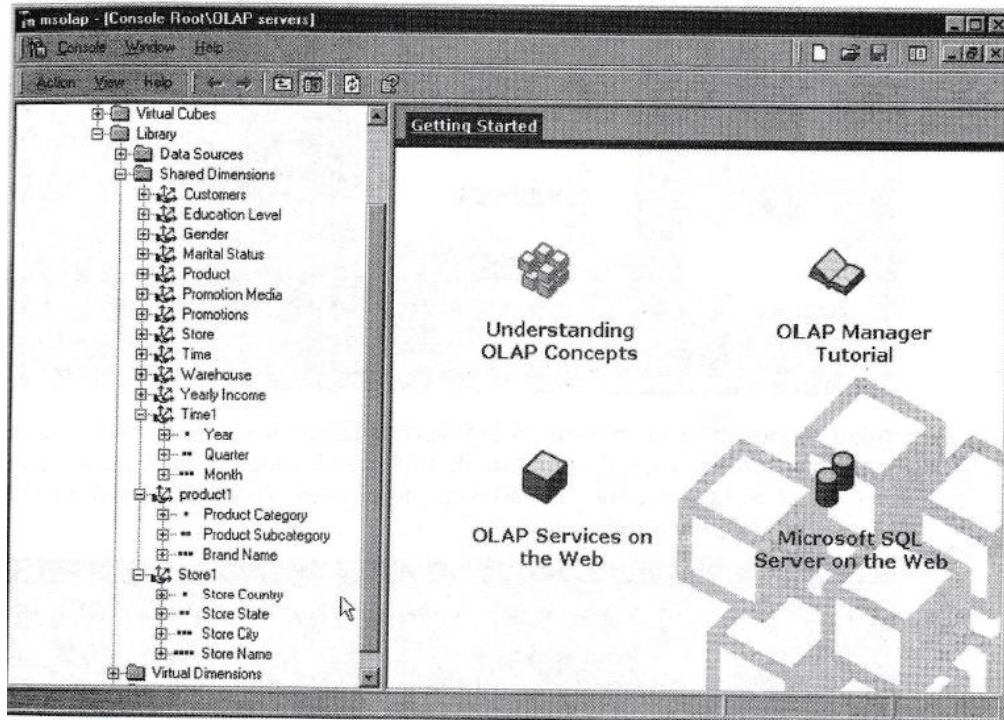


Kini pilihlah tingkat yang ditampilkan pada gambar berikut ini :



Pada langkah yang terakhir, ketikkanlah Store1 sebagai nama dari dimensi tersebut, lalu tekan OK .

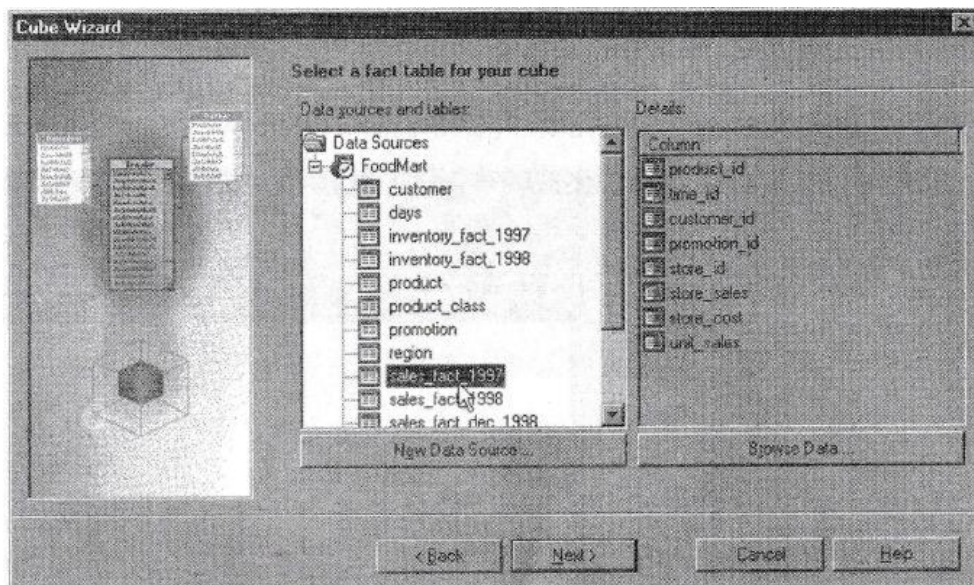
Setelah menutup Dimension Wizard, tekanlah File dan Exit. Periksalah untuk melihat bahwa semuanya telah dibuat di bawah folder Shared Dimensions.



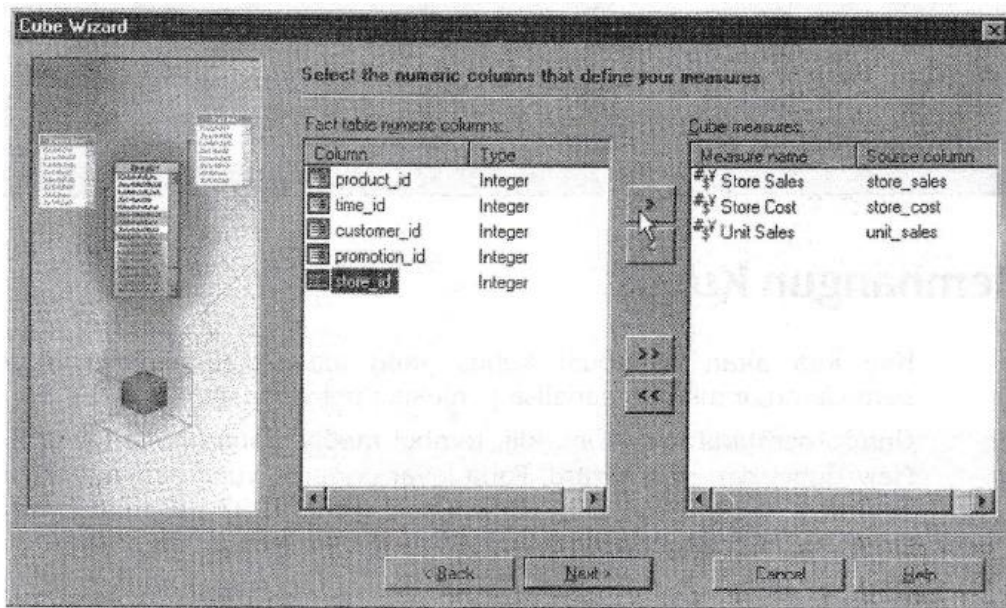
Membangun Kubus

Kini kita akan membuat kubus yang akan digunakan oleh departemen pemasaran untuk menganalisa penjualan perusahaan.

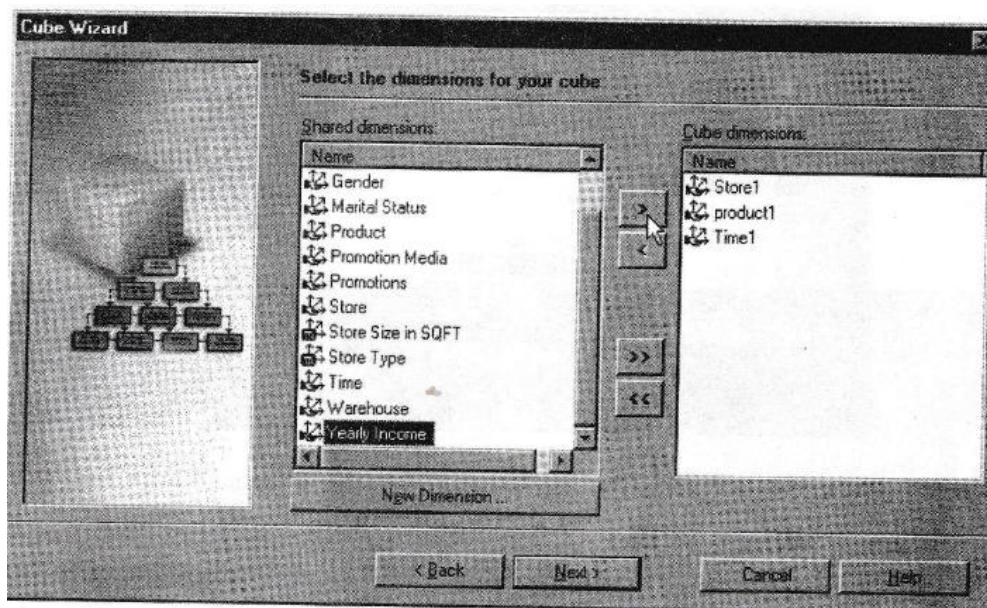
Untuk membuat kubus ini, klik mouse kanan pada folder Cubes, pilih New Cube, dan pilih wizard. Pada layar pertama yang hanya merupakan layar informasi, tekan Next. Lalu pilihlah sales_face_1997 pada database FoodMart. Database ini mengandung informasi penjualan tahun 1997.



Setelah menekan Next, pilihlah kolom-kolom numerik yang akan digunakan pada langkah berikutnya. Gambar berikut ini menunjukkan kolom-kolom yang dipilih. Kolom-kolom tersebut adalah store_sales (penjualan), store_cos (biaya), dan unit_sales (unit penjualan).



Tekan Next dan pilihlah dimensi untuk kubus tersebut. Karena kita ingin menggunakan ketiga dimensi, klik tombol pertama dengan dua panah untuk memilih semuanya.



Membuat Dimensi dengan Cepat

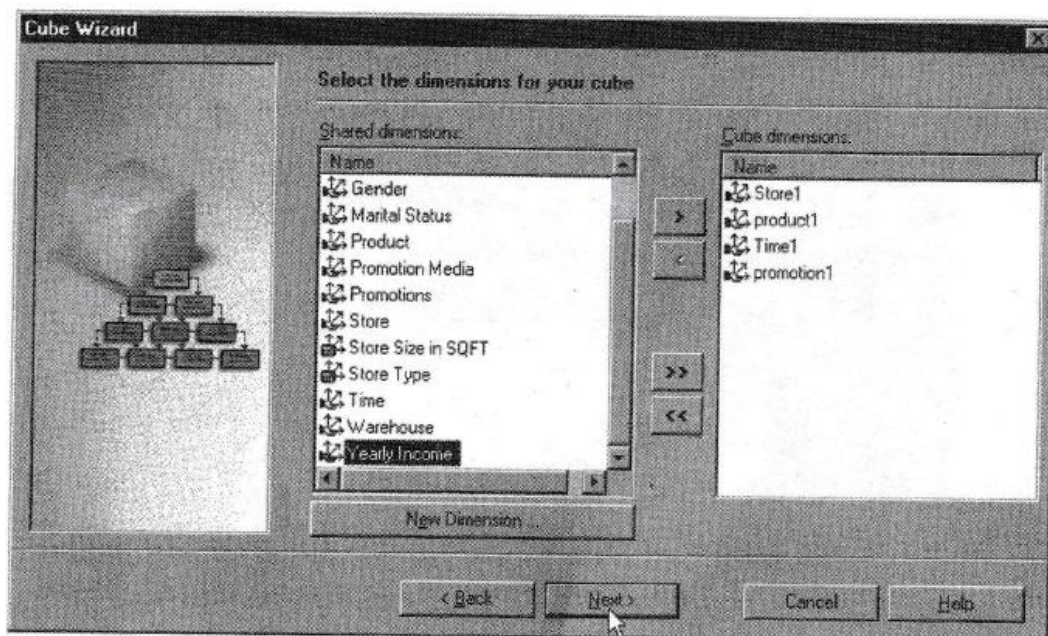
Sebelum menyelesaikan pembuatan kubus, Anda bisa menambahkan dimensi baru kapan saja. Tekan saja tombol New Dimension untuk menjalankan wizard sekali lagi. Pada bagian ini, kita akan membuat sebuah dimensi bernama Promotion, dengan mengikuti langkah-langkah berikut ini:

Pilihlah tabel Single Dimension dan tekan Next.

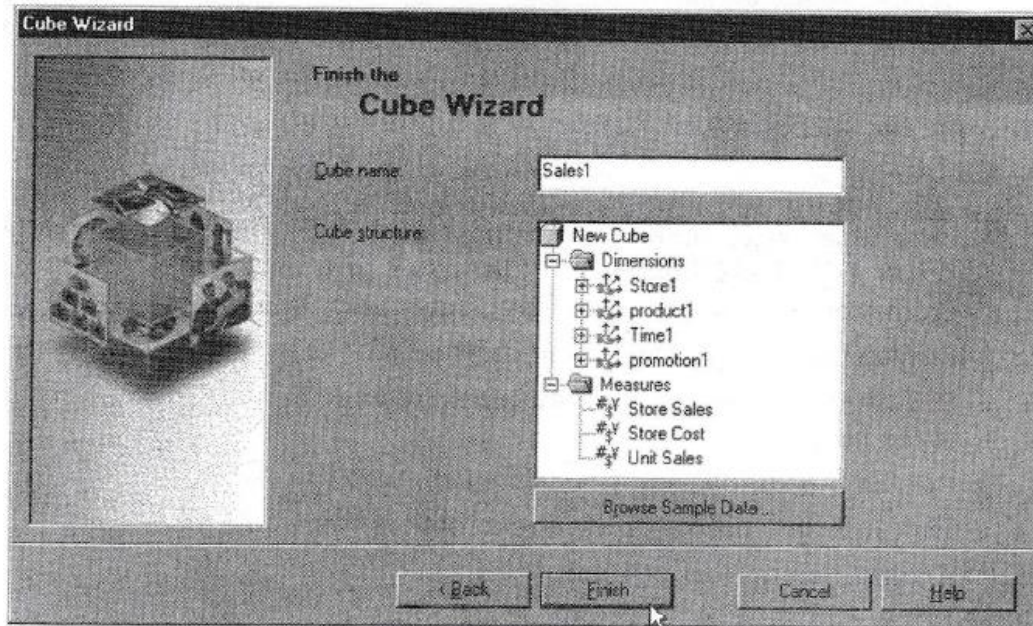
Pilihlah database Tutorial atau tabel Promotion pada database FoodMart. Lalu tekan Next.

Pilihlah Standard Dimension lalu tekan Next.

Pilihlah kolom media_type dan promotion_name, lalu tekan Next. Ketikkanlah promotion1 sebagai nama dari dimensi, lalu tekan Next.



Pada langkah terakhir, ketikkanlah Sales1 sebagai nama kubus lalu tekan tombol Finish.

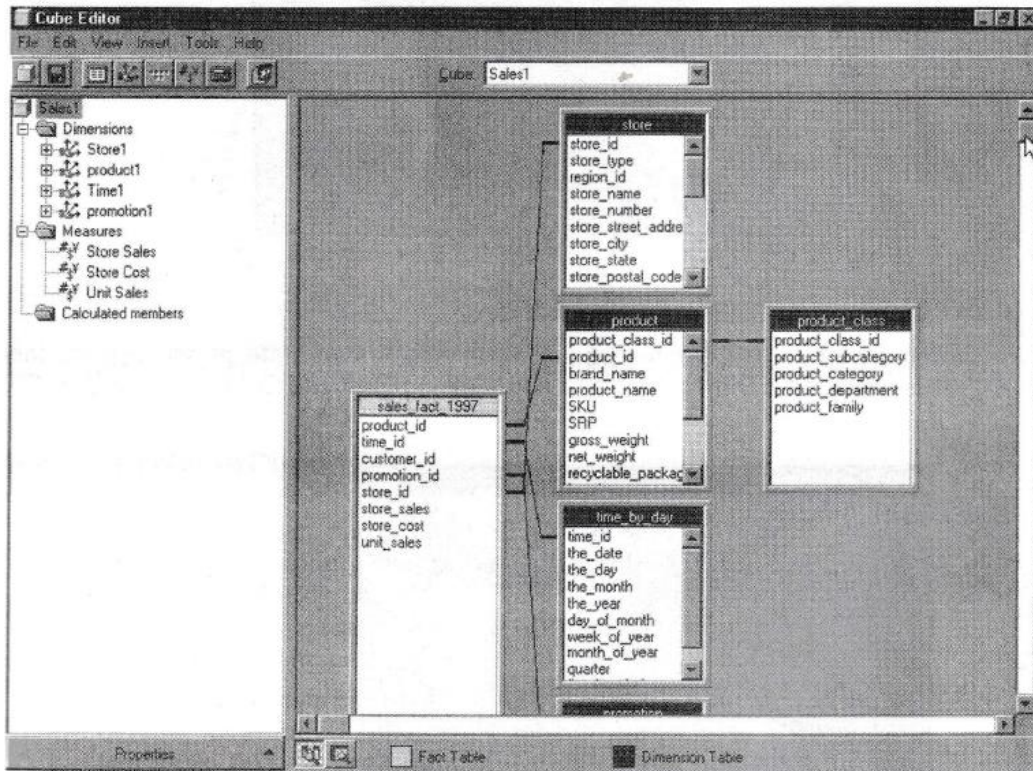


Semua dimensi kini telah tersedia di bawah folder Shared Dimensions.

Menyunting Kubus

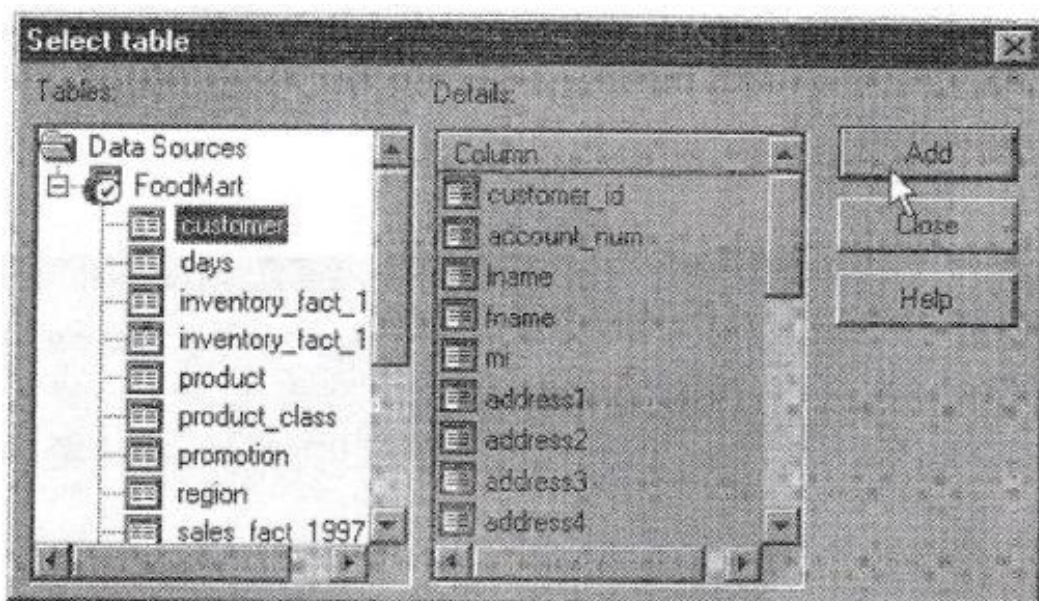
Cube Editor akan dibuka secara otomatis. Jika tidak, bukalah pohon database FoodMart, klik pada kubus Sales dengan tombol mouse kanan, lalu pilih Edit Cube. Cube Editor akan dibuka dan menampilkan antarmuka grafis yang dibentuk oleh dua panel.

Panel kiri dari Cube Editor berupa tampilan pohon kubus. Di sebelah kanan adalah panel kanan dengan tampilan grafis dari struktur kubus. Tabel ditampilkan dengan baris judul berwarna kuning dan tabel dimensi ditampilkan dengan baris judul biru.



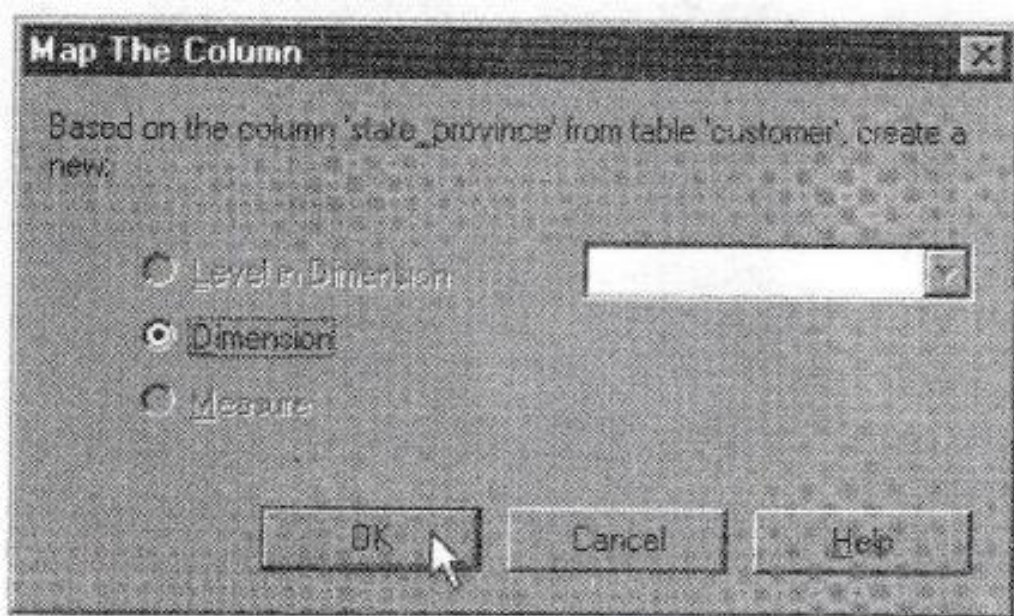
Menambahkan Dimensi Baru

Dengan menggunakan Cube Editor, kita akan menambahkan sebuah dimensi yang mengandung data demografis dari para klien. Untuk menambah dimensi baru, aktifkanlah menu Insert dan pilihan Tables. Pada kotak dialog yang muncul, pilihlah tabel Customer pada database FoodMart.

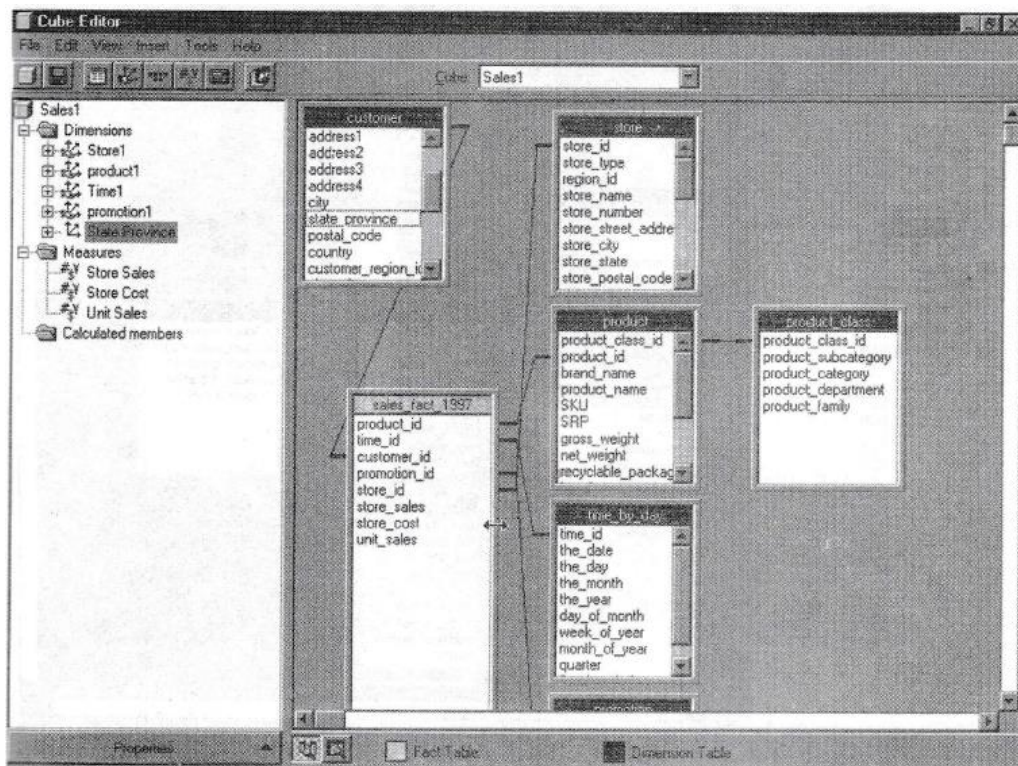


Lalu tekan tombol Add dan tombol Close.

Untuk menambah dimensi baru, klik ganda kolom state-province pada tabel Customer. Pilih Dimension pada kotak dialog Map The Column, lalu tekan OK.



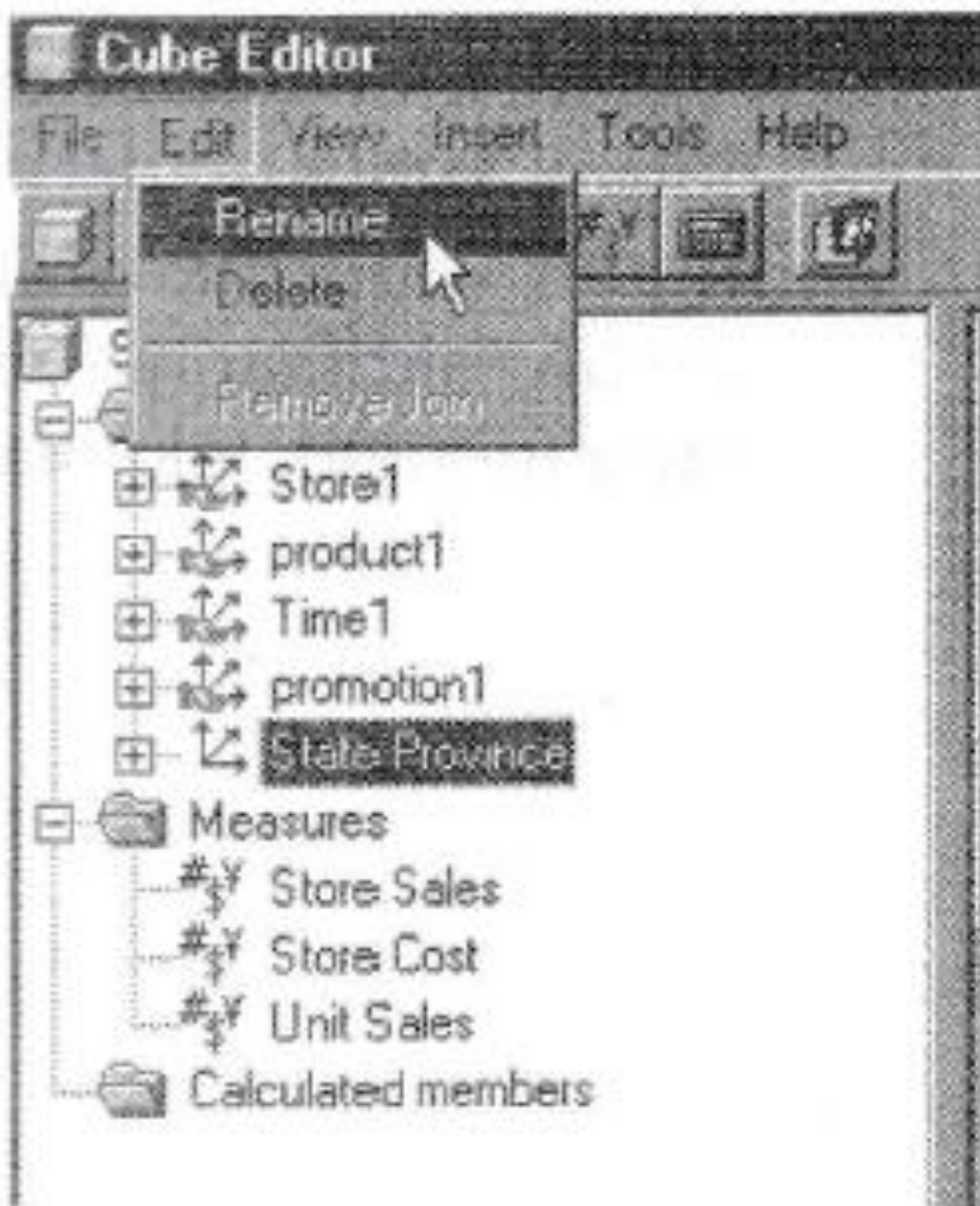
Dimensi yang baru akan ditampilkan pada pohon database dan panel skema.

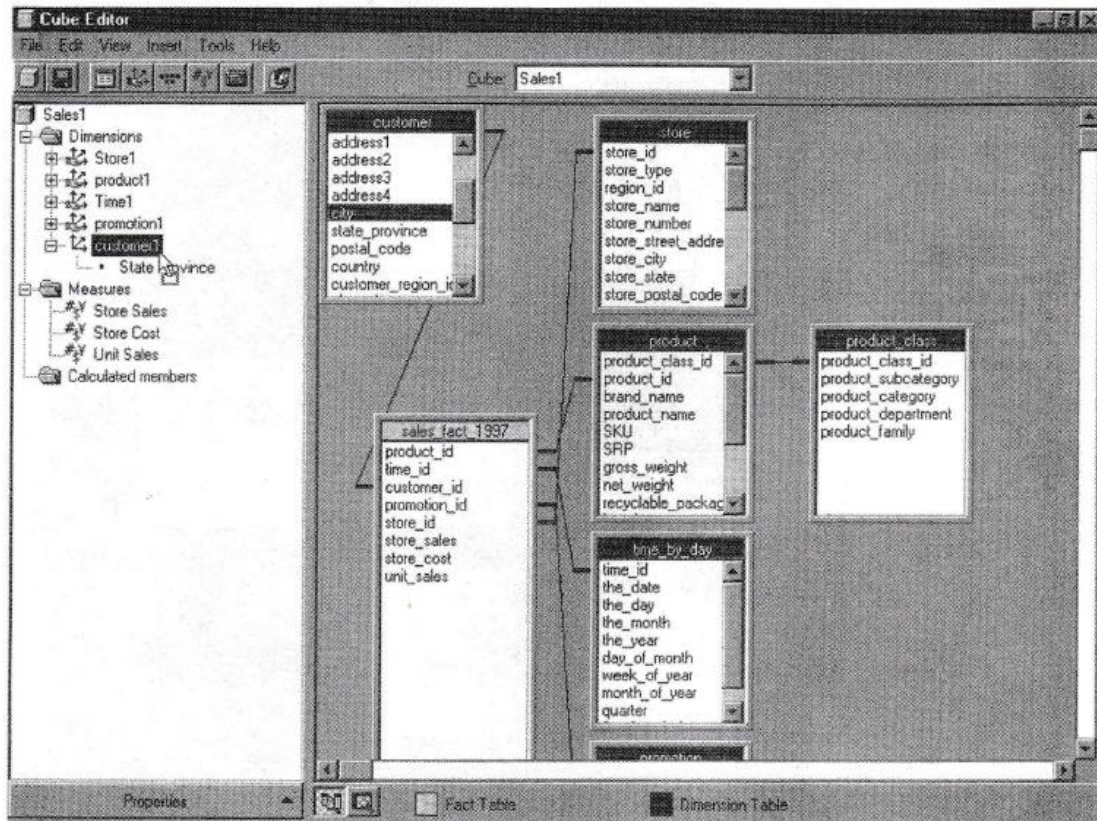


Mengubah Nama Dimensi

Untuk mengubah nama dimensi, klik kanan pada pohon database lalu pilih Rename, atau aktifkan menu Edit dan pilih Rename. Pilihlah state_province pada pohon, dan gunakan pilihan Edit | Rename.

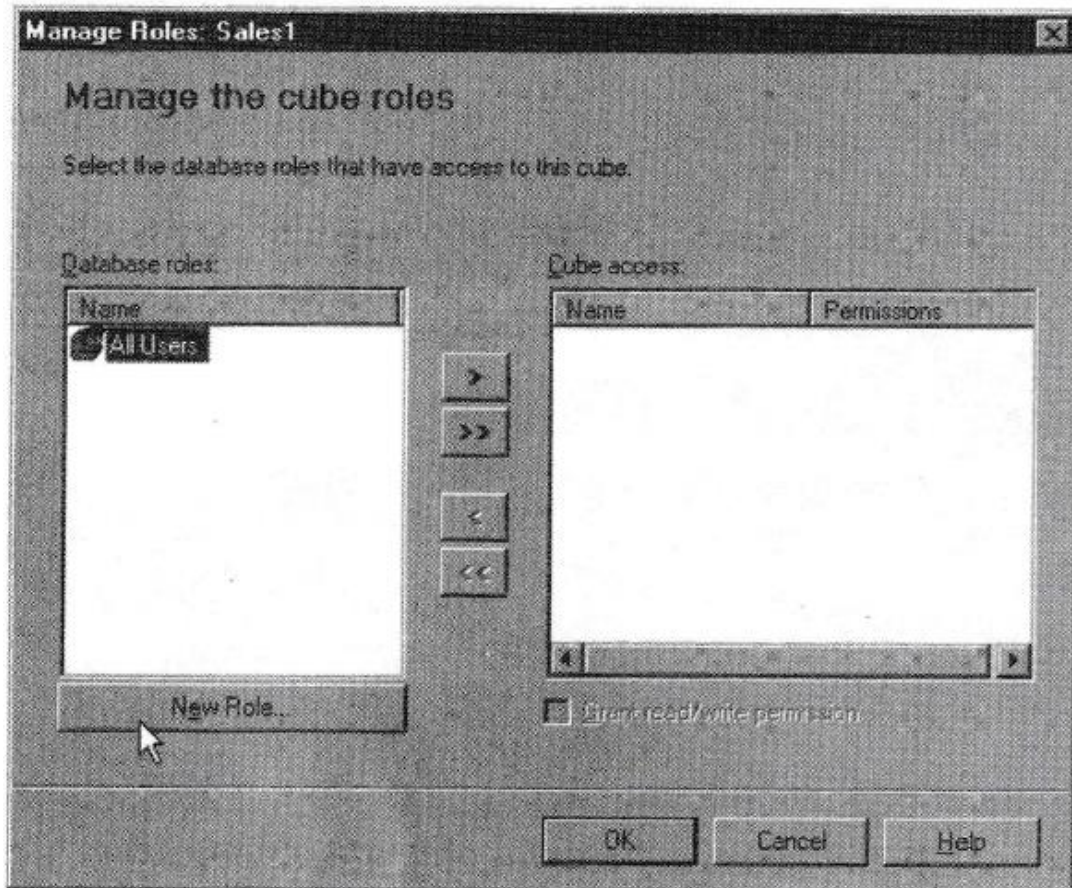
Ketikkan customer l sebagai nama baru untuk dimensi tersebut. Untuk menambahkan item lain, klik lalu geser kolom City dari tabel Customer pada panel skema ke dalam tabel Customer pada pohon tampilan. Dimensi Customer kemudian akan dibentuk dari State Province dan City.



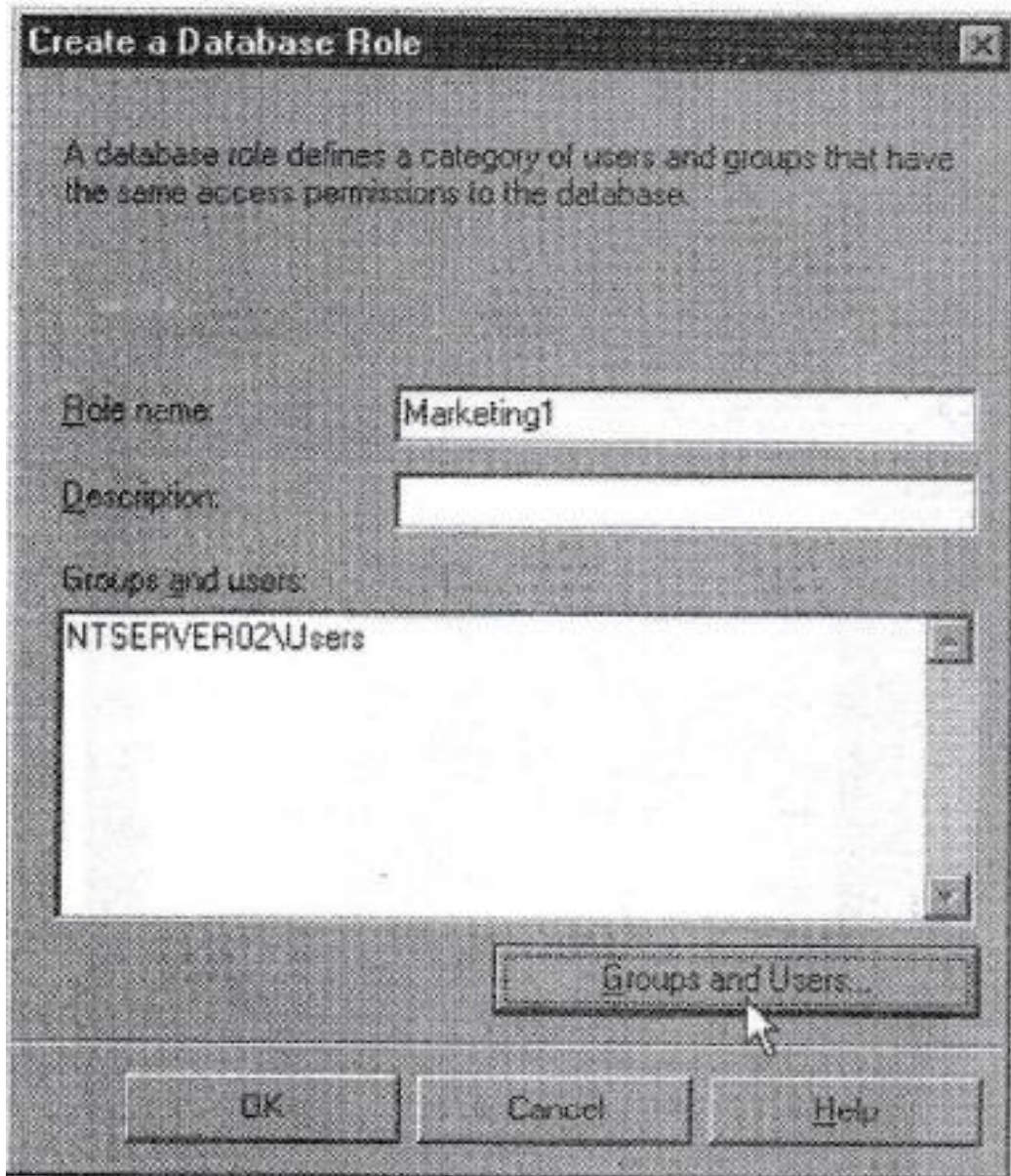


Menambah Aturan

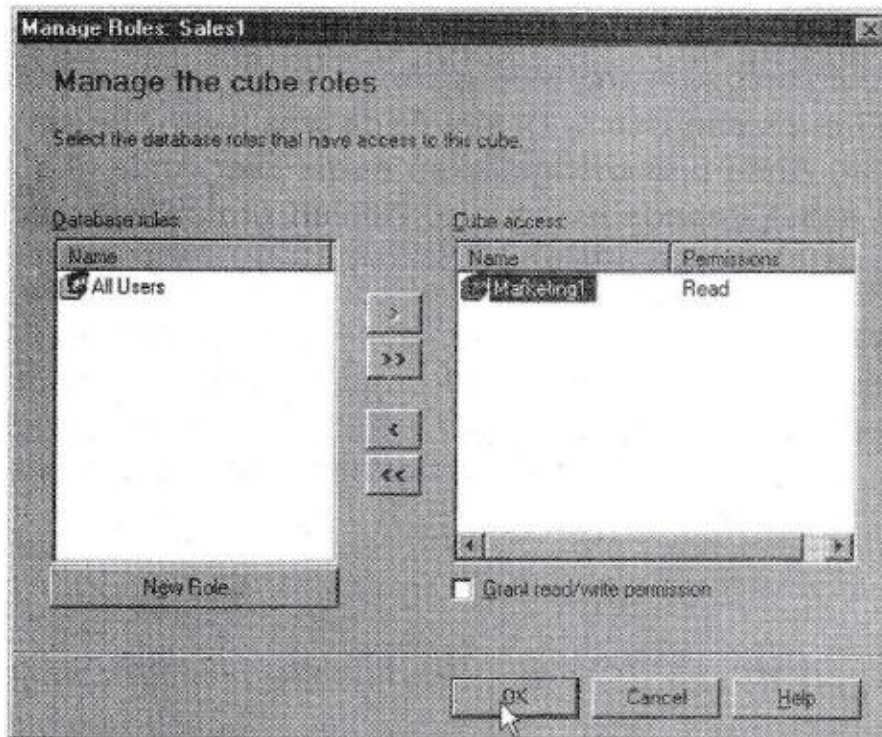
Aturan mendefinisikan user dan kelompok user yang bisa mengakses data di dalam kubus yang di buat. Pada bagian ini, kita akan membuat sebuah aturan yang mengizinkan beberapa user mengakses kubus. Untuk membuat aturan, aktifkanlah menu Tools dan pilihan Manage Roles. Pada kotak dialog yang muncul, tekanlah tombol New Role.



Ketikkan nama aturan pada kotak Role Name; pada contoh ini, gunakanlah Marketing 1. Lalu pilihlah user yang bisa menggunakan kubus tersebut dengan menekan tombol Groups dan Users. Di sini Anda bisa memilih user yang akan memiliki akses, atau Anda bisa menyetikkan nama user Anda jika Anda tidak ingin memberikan akses kepada orang lain. Pilihlah semua user dari data base (dalam contoh ini, data basenya adalah NTSERVER02).



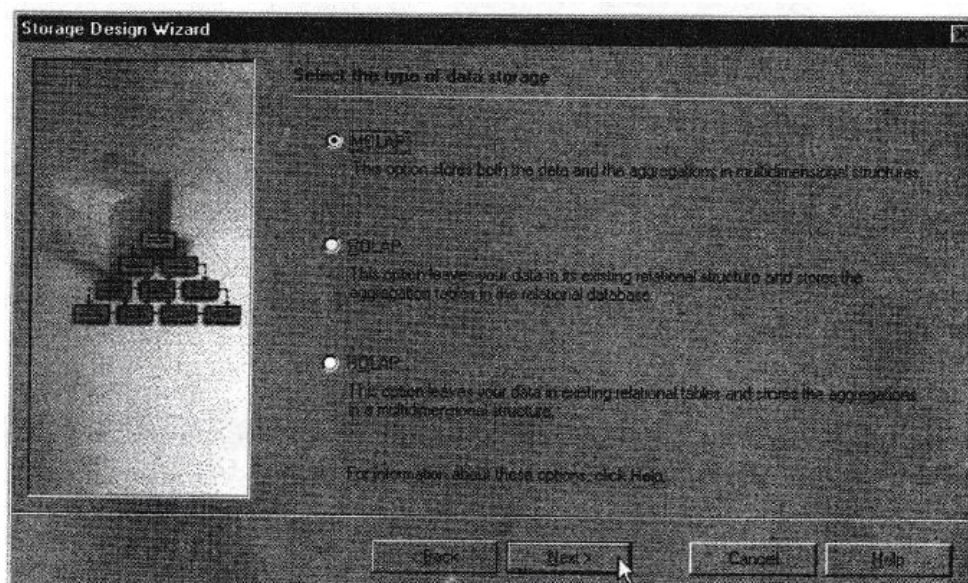
Selesai sudah. Tekan OK untuk melanjutkan ke layar berikutnya seperti ditampilkan di bawah ini.



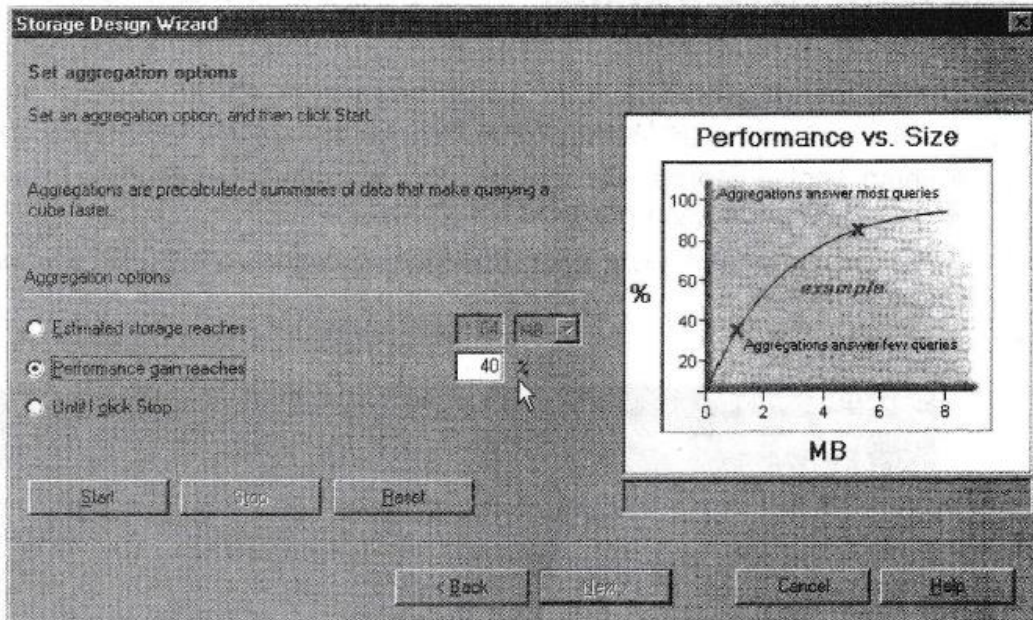
Apabila Anda menekan tombol OK, pembuatan aturan akan selesai.

Merancang Data Koleksi

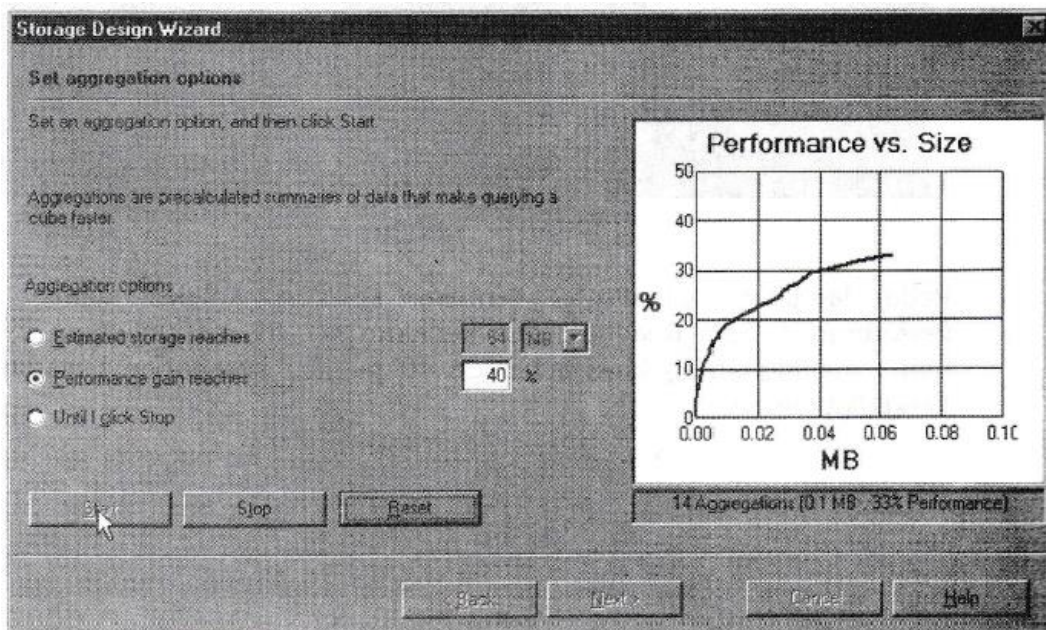
Untuk mendapatkan kinerja yang terbaik di dalam query, Anda harus menjalankan Storage Design Wizard sebelum menjalankan proses query. Pilihlah Tools | Storage Design. Ini akan mengaktifkan Storage Design Wizard. Tekan Next pada layar pertama. Jenis penyimpanan yang akan kita gunakan adalah MOLAP. Pilihlah MOLAP lalu tekan Next.



Pada langkah berikutnya, tentukan pilihan koleksi dengan memilih Performance Gain Reaches, dan ketikkan 40. Ini menunjukkan bahwa DSS harus meningkatkan kinerja hingga 40 persen, berapapun besar ruang disk yang digunakan.



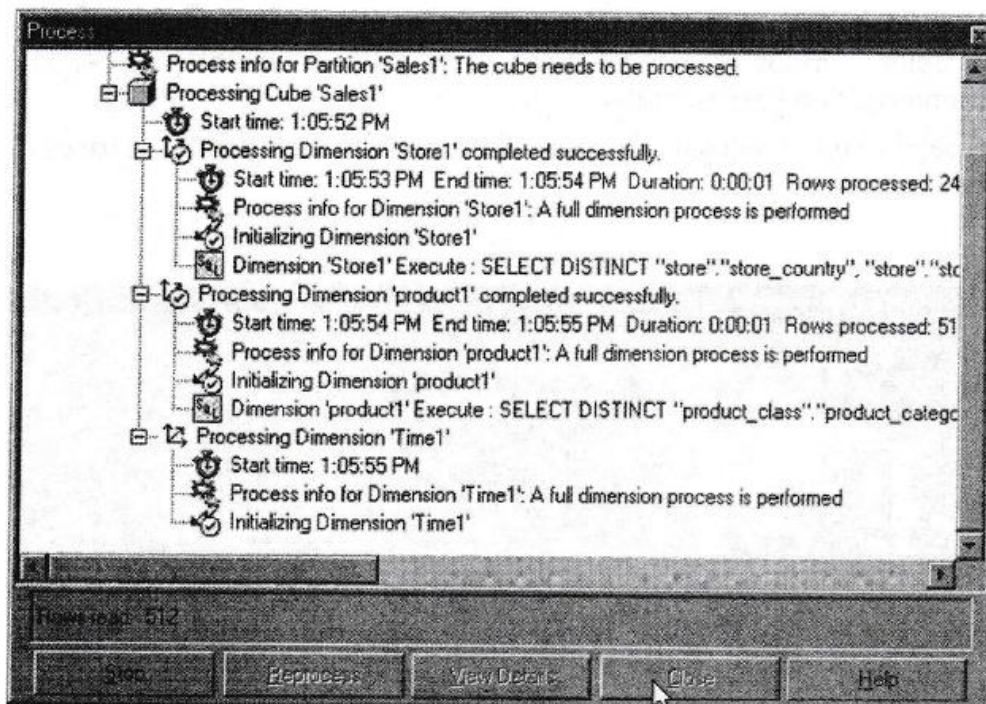
Tekan Start untuk memulai proses. Dalam beberapa saat, sebuah jendela akan menampilkan jumlah koleksi yang dijalankan untuk mendapatkan persentase yang diminta.



Kemudian tekanlah Next, dan pilih pilihan default untuk menyimpan koleksi apabila tombol Finish ditekan. Proses akan dimulai dan sebuah panel menampilkan kemajuannya. Apabila sudah selesai, tekanlah Next, pilih pilihan default (Process Now), lalu tekan Finish.



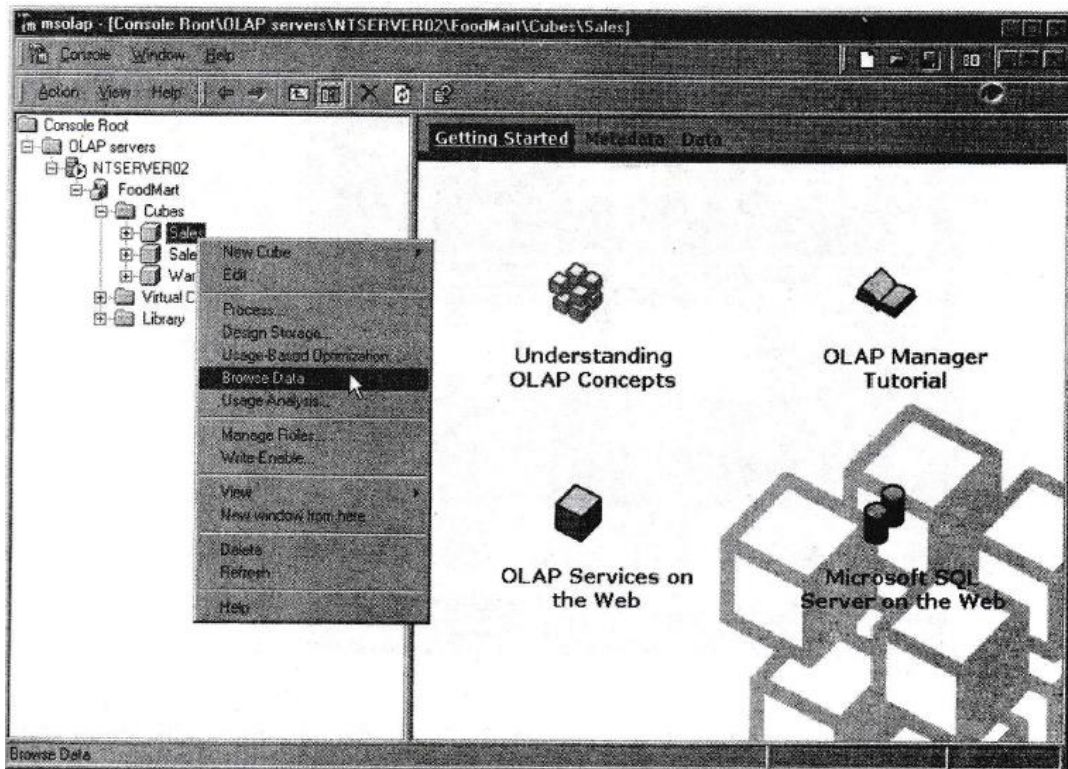
Tugas pemrosesan kubus akan dimulai, dan kemajuannya ditampilkan dalam sebuah panel.



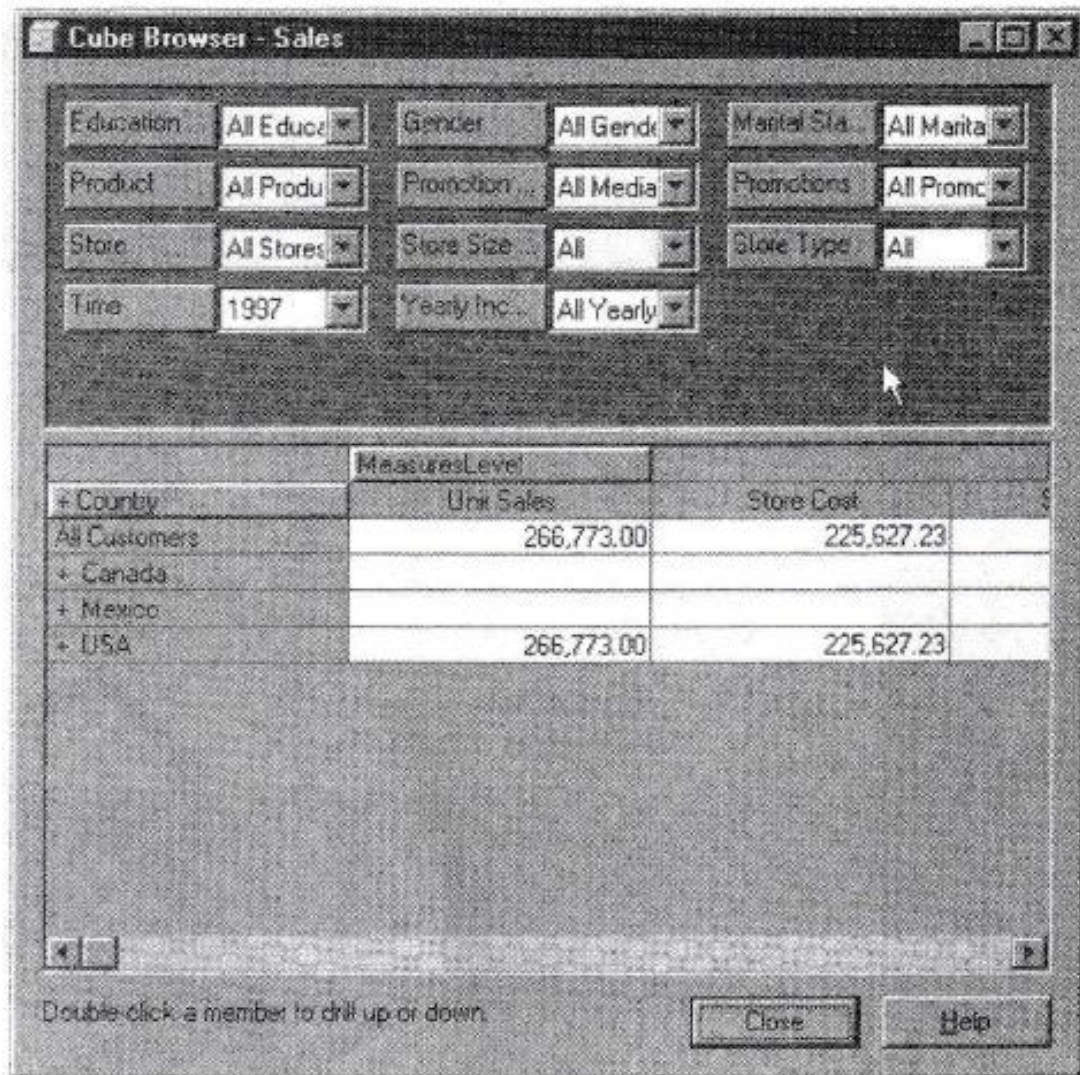
Apabila sebuah pesan muncul dan menandakan bahwa proses telah berakhir tekanlah tombol Close. Jika gagal, tekanlah tombol View Details untuk melihat apa yang terjadi.

Menjelajah dengan Kubus

Akhirnya kita bisa menjelajahi dengan kubus. Lakukanlah tugas ini dengan pilihan Browse Data, yang bisa diaktifkan dari menu shortcut dengan mengklik tombol mouse kanan pada nama kubus, dalam contoh ini adalah kubus Sales.

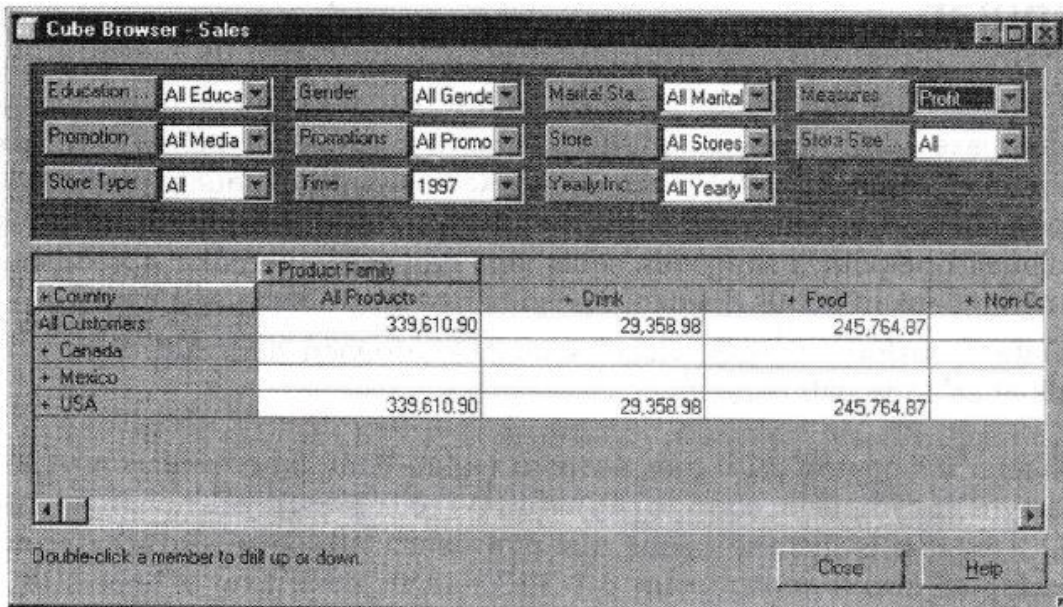


Cube Browser akan muncul. Ia memiliki dua bagian utama: di bagian atas adalah panel dimensi dan di bagian bawah adalah grid.

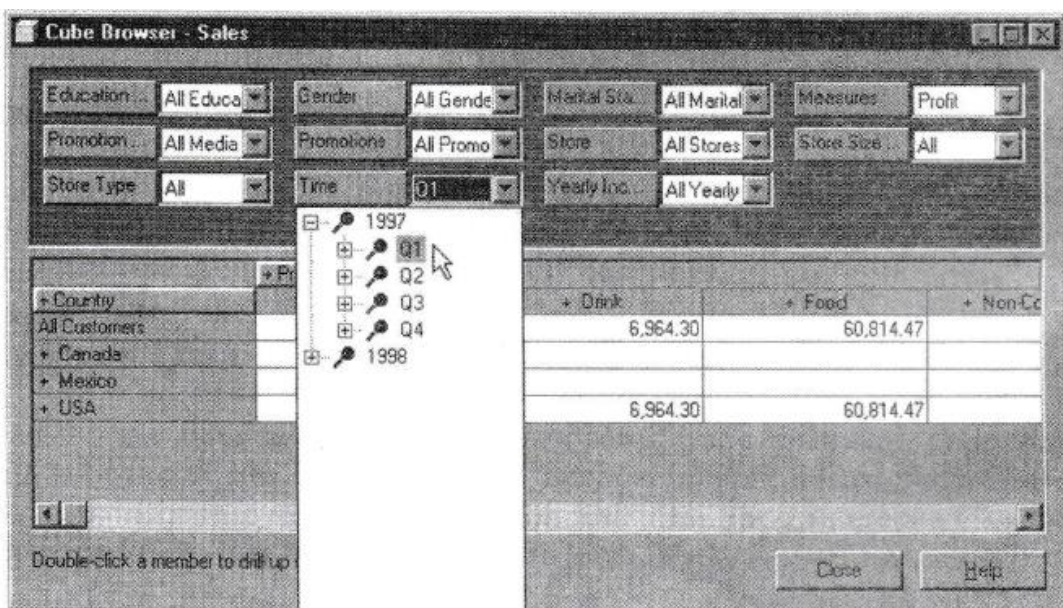


Anda bisa menyaring data yang ditampilkan dengan mengklik pada daftar pilihan yang muncul di sebelah nama dimensi dan memilih item tertentu.

Anda juga bisa mengubah dimensi di dalam grid, menggantikan satu dimensi dengan dimensi lain yang Anda pilih pada panel dimensi. Untuk melakukan ini, klik dan geser dimensi yang diinginkan pada panel dimensi di atas dimensi yang ingin Anda ganti di dalam grid. Sebagai contoh, kita akan menggeser dimensi Product di atas dimensi MeasureLevel. Gambar di bawah ini menampilkan hasilnya.



Jika Anda ingin menam bahkan dimensi ke dalam grid, geserlah dimensi dan letakkan pada daerah yang tidak memiliki dimensi lain. Untuk menyaring data yang ditampilkan, pilihlah tingkat penyaringan pada daftar pilihan. Layar berikutnya menampilkan cara memilih filter untuk kuartal pertama pada tahun 1997.



Untuk memperbesar sebuah kategori, klik ganda pada kategori tersebut.

Kesimpulan

Bekerja dengan kubus adalah tugas yang membutuhkan pengetahuan database yang menyeluruh, terutama mengenai kebutuhan informasi user. Langkah-langkah untuk

mempersiapkan kubus sangatlah penting di dalam perancangan model yang dapat memuaskan user. Tetapi setelah kubus selesai dibuat, operasinya sangatlah sederhana dan setiap pemakai spreadsheet dapat menggunakan kubus dengan mudah hanya dalam beberapa menit.

Data Transformation Services, sebuah komponen dari SQL Server akan dibahas pada bab berikutnya.

BAB 15

Data Transformation Services

Salah satu modul SQL Server yang paling menarik adalah Data Transformation Services (DTS). Dengan DTS Anda dapat mengonversi data dari SQL Server ke format data lain yang kompatibel dengan OLE DB, ODBC, dan file teks. Penggunaan OLE DB mengizinkan akses ke dalam data di dalam penyedia OLE DB, seperti misalnya Access, Excel, dan SQL Server itu sendiri. Anda juga bisa mengakses sumber data seperti Oracle, Sybase, DB2, dan sumber-sumber lain yang memiliki penyedia ODBC, serta mengekspor data untuk membuat tabel pivot pada spreadsheet Excel. DTS adalah elemen dasar untuk pembuatan database untuk penunjang keputusan, serta dapat membantu pengumpulan dan transformasi data.

Konsep

Mengimpor dan Mengekspor Data

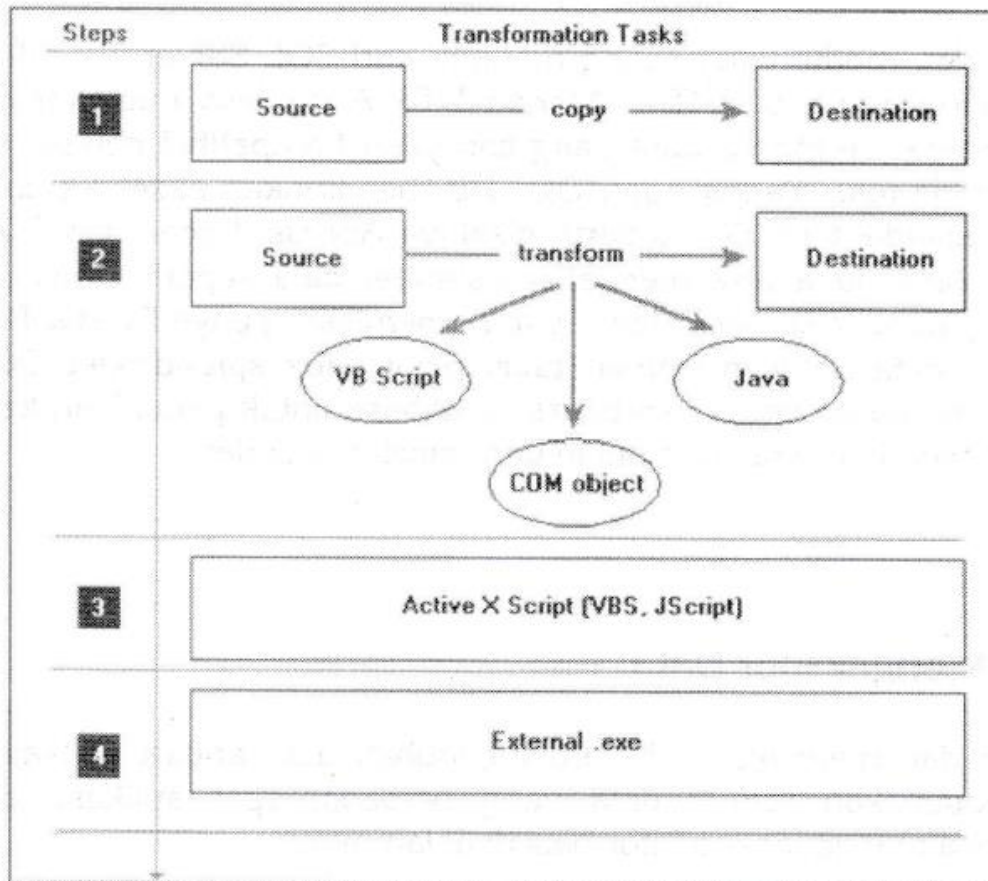
Mengimpor dan mengekspor berarti mengubah data antara aplikasi-aplikasi yang menggunakan format data yang berbeda, agar aplikasi lokal atau eksternal bisa menggunakan informasi di dalamnya.

Mentransformasi Data

Transformasi data adalah serangkaian operasi yang dilakukan pada sumber data sebelum ditulis ke tujuan. Misalnya, kolom tabel dapat merupakan hasil transformasi dan perhitungan yang diterapkan pada kolom dan baris dari satu tabel atau lebih.

Arsitektur

Paket DTS pada dasarnya adalah deskripsi terperinci dari operasi yang harus dilakukan untuk mengimpor, mengekspor, atau mengubah data. Paket DTS mungkin melibatkan penyalinan tabel antar database dan konversi data dari satu format ke format lainnya menggunakan ODBC, ActiveX, perintah SQL, atau bahkan program eksternal yang mungkin merupakan bagian dari paket DTS.



Dengan menggunakan teknologi yang berbeda dan arsitekur COM, Anda bisa dengan mudah membuat struktur dasar dari data warehouse.

Membuat Paket DTS

Paket DTS dapat dibuat dengan dua cara, baik dengan perancang paket DTS atau dengan wizard. Perancang Paket Layanan Transformasi Data adalah alat grafis yang dirancang untuk pengguna tingkat lanjut untuk menggabungkan, mengubah, dan mengintegrasikan data heterogen dari berbagai sumber. Melalui penggunaan DTS Package Designer, Anda bisa mendefinisikan alur kerja dan alur data, serta menyimpannya di dalam penyimpanan SQL Server.

DTS Import dan Export Wizards sama dengan wizard-wizard lain yang berinteraksi dengan user.

Fungsi dari DTS dibatasi oleh kapasitas driver OLE DB atau sumber data. Melalui penggunaan wizard, Anda bisa mengimpor, mengeksport, dan mentransformasi data dari sumber-sumber berikut ini :

- dBase dan Paradox

- Teks ASCII dengan ukuran field yang tetap
- Sumber data ODBC
- Database SQL Server
- Excel spreadsheet
- FileAccess dan FoxPro
- Sumber data OLE DB kustom

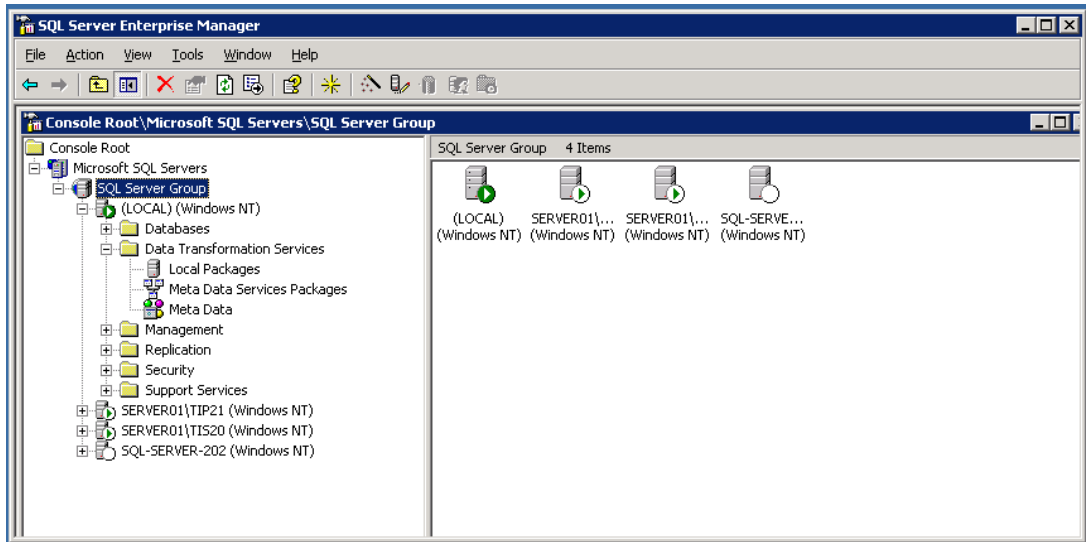
Di antara operasi yang dilakukan oleh wizard adalah:

- Menjalankan skrip ActiveX, Jscript, PerlScript, dan VBScript.
- Meng-copy seluruh tabel atau hasil dari query SQL
- Membuat query di dalam wizard
- Mengubah nama, tipe data, ukuran, ketepatan, skala, dan penerimaan nilai Null dari suatu kolom selama proses copy dari sumber ke tujuan.
- Menyimpan paket DTS di dalam data base SQL Server, dan memprogram eksekusinya

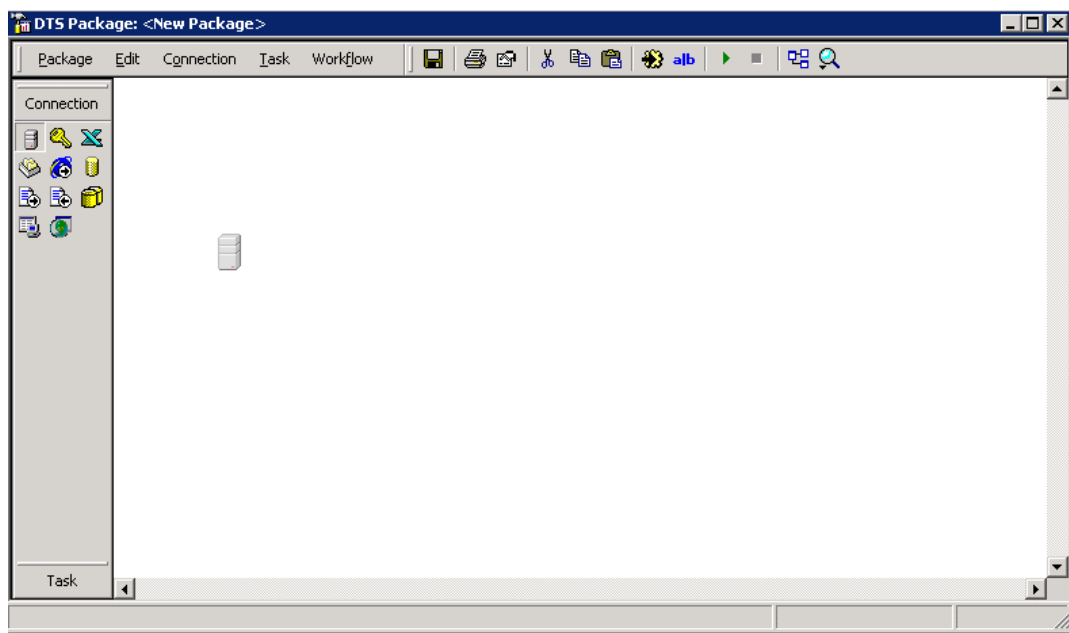
Membuat Paket DTS

Pada bagian ini, kita akan melihat contoh cara membuat paket DTS dengan menggunakan DTS Package Designer. Tugas ini terdiri dari mentransformasi database Northwind ke dalam database bernama DWStar. Anda bisa mengikuti langkah-langkah prosedur ini atau hanya melihat langkah-langkah melalui layar-layar yang ditampilkan pada bagian ini. Langkah pertama adalah membuat database bernama NWStar yang ukurannya sama dengan database NorthWind. Lalu ikutilah langkah-langkah berikut ini:

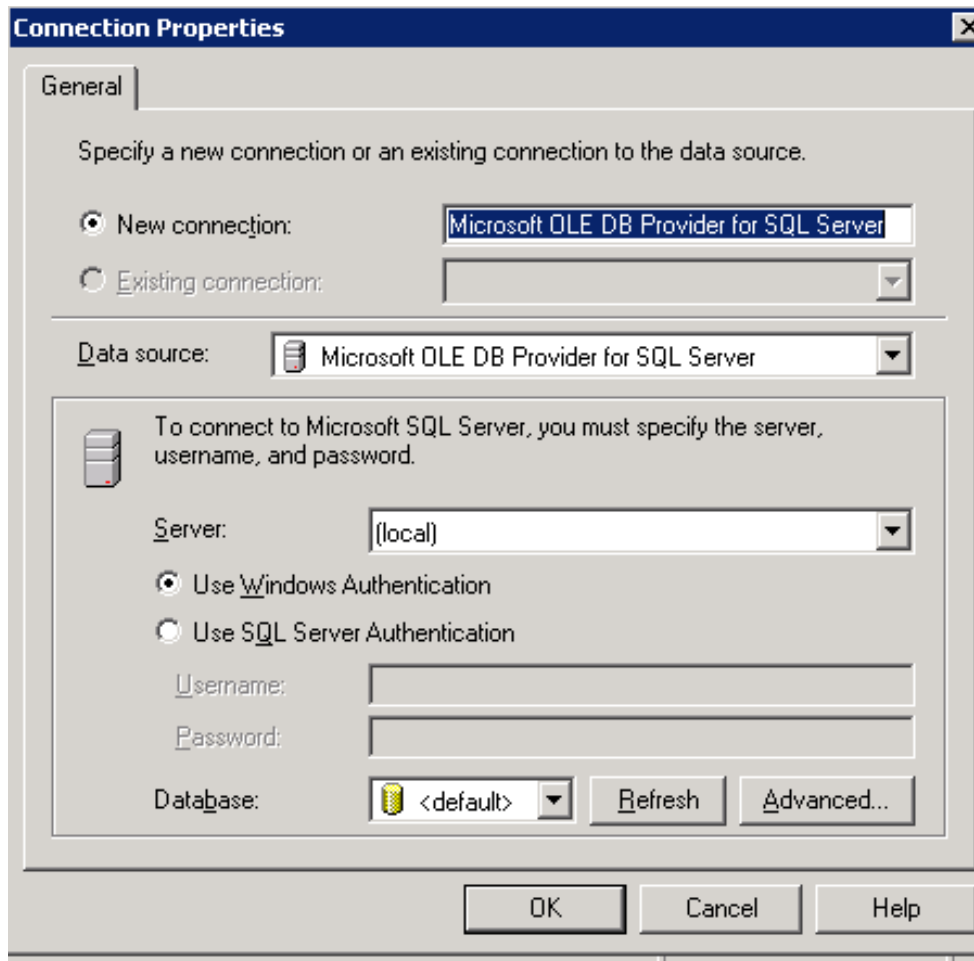
Jalankan Enterprise Manager dan hubungilah server Anda dengan mengklik pada tanda plus di sebelah SQL Server Group, lalu klik pada nama server, dalam contoh ini NTSERVER02.



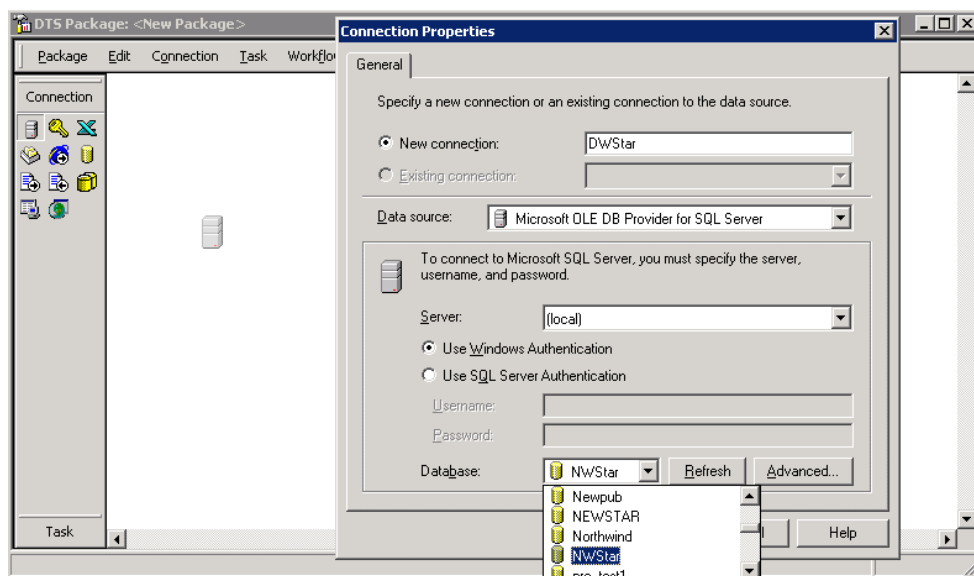
Klik kanan pada folder Data Transformation Services dan pada Local Packages. Pada menu yang muncul, pilihlah New Package. Sarana paket DT akan muncul.



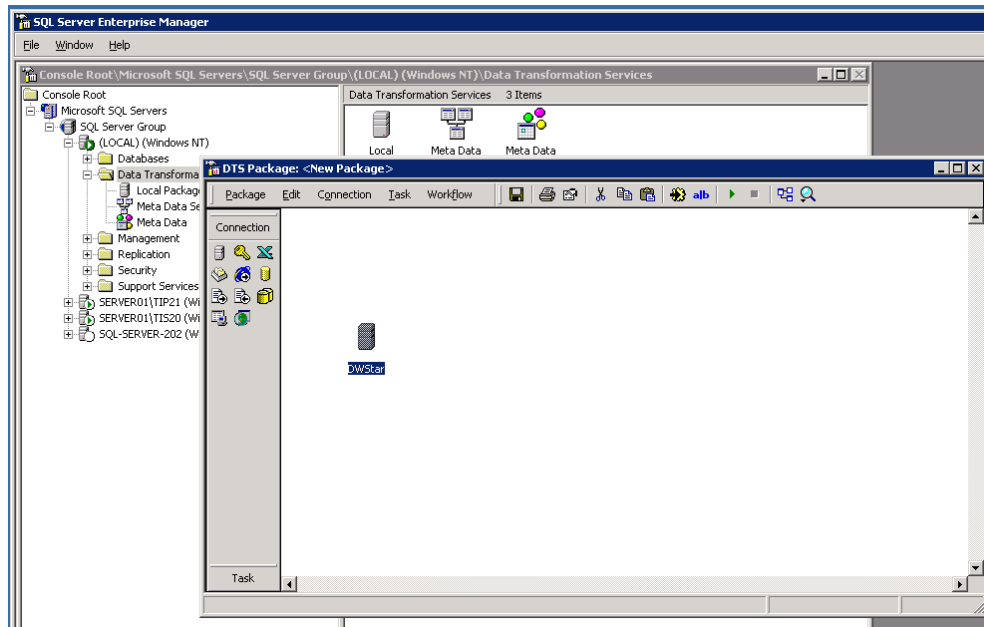
Lalu klik dan geser SQL Server (icon server di sebelah kiri jendela daerah kerja di bawah kata Data pada toolbar) untuk membuat sebuah koneksi. Ini akan membuka kotak dialog Properties.



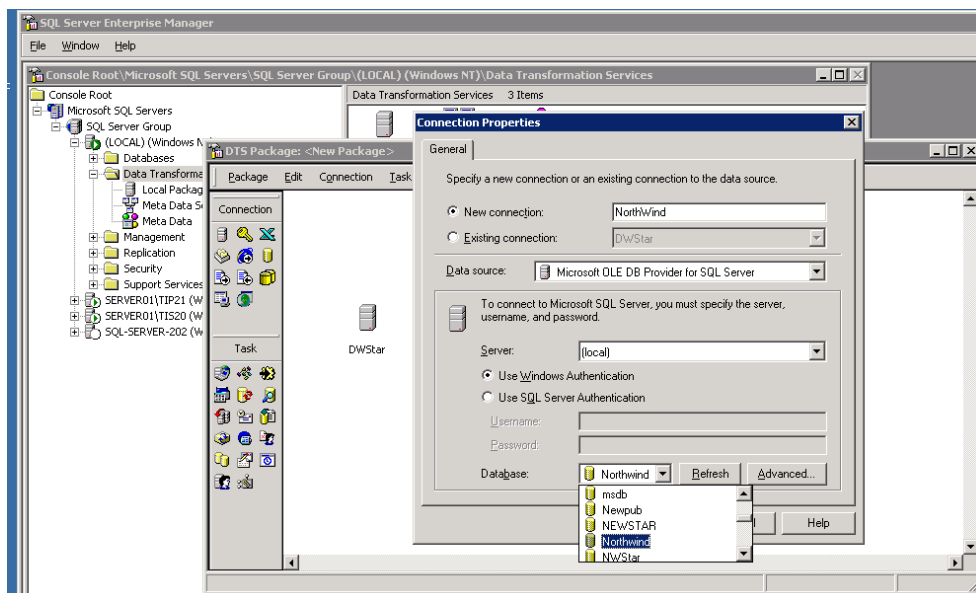
Pada field New Connection, ketikkanlah DWStar seperti ditunjukkan pada layar di bawah ini. Pilihlah Use Windows NT Authentication. Klik tombol Refresh untuk meng-update daftar database dan pilih NWStar pada field Database.



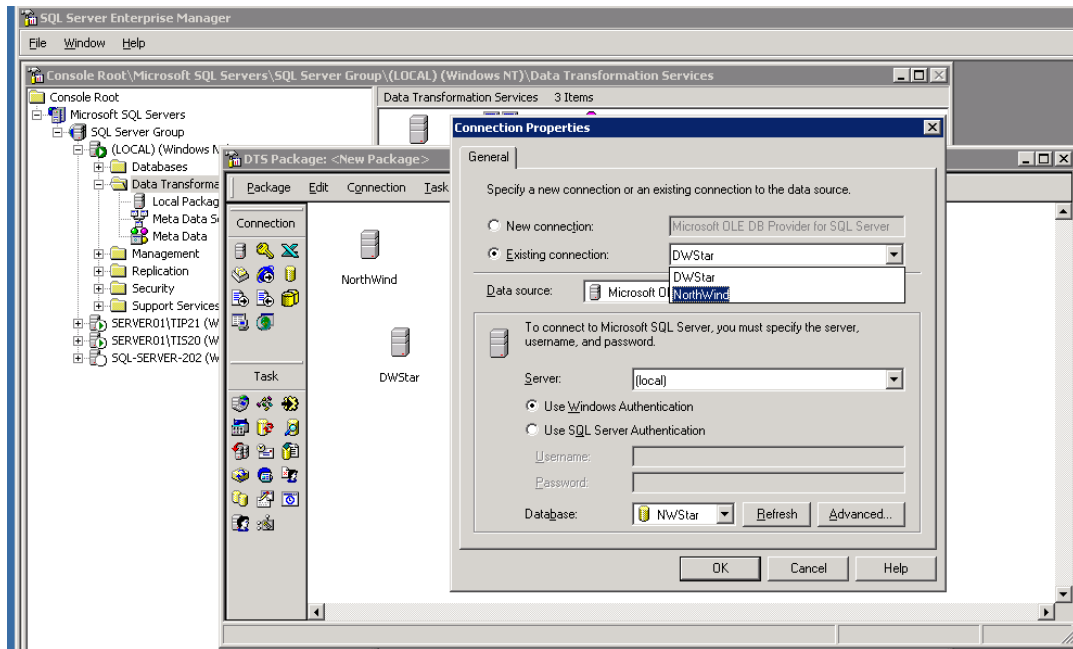
Tekanlah tombol OK. Daerah kerja akan menampilkan bahwa kini Anda telah memiliki koneksi. Setiap dimensi yang ingin diisi atau dipopulasi harus memiliki koneksinya sendiri-sendiri.



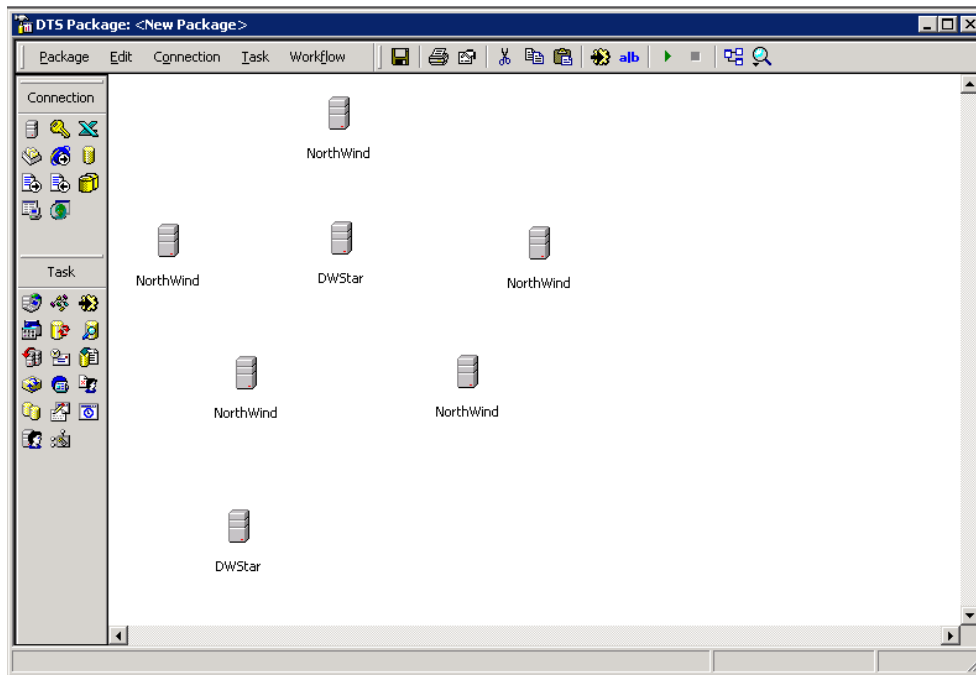
Pada contoh ini, kita akan membuat lima buah koneksi. Untuk membuat koneksi kedua, klik dan geser icon SQL Server ke dalam daerah kerja. Apabila kotak dialog Connection Properties muncul, ketikkanlah NorthWind pada field New Connection, pilih Use Windows NT Authentication, klik Refresh, lalu pilih Northwind dari daftar database.



Tekan tombol OK untuk menyelesaikan pembuatan koneksi. Kita akan menggunakan metode yang berbeda untuk menambahkan koneksi-koneksi lain. Kita tidak akan membuat koneksi baru, tetapi akan menggunakan koneksi yang sudah ada. Untuk melakukan ini, tekan atau geser icon SQL Server ke dalam daerah kerja. Pada kotak dialog Connection Properties, pilihlah item Existing Connection, lalu pilih NorthWind pada daftar.

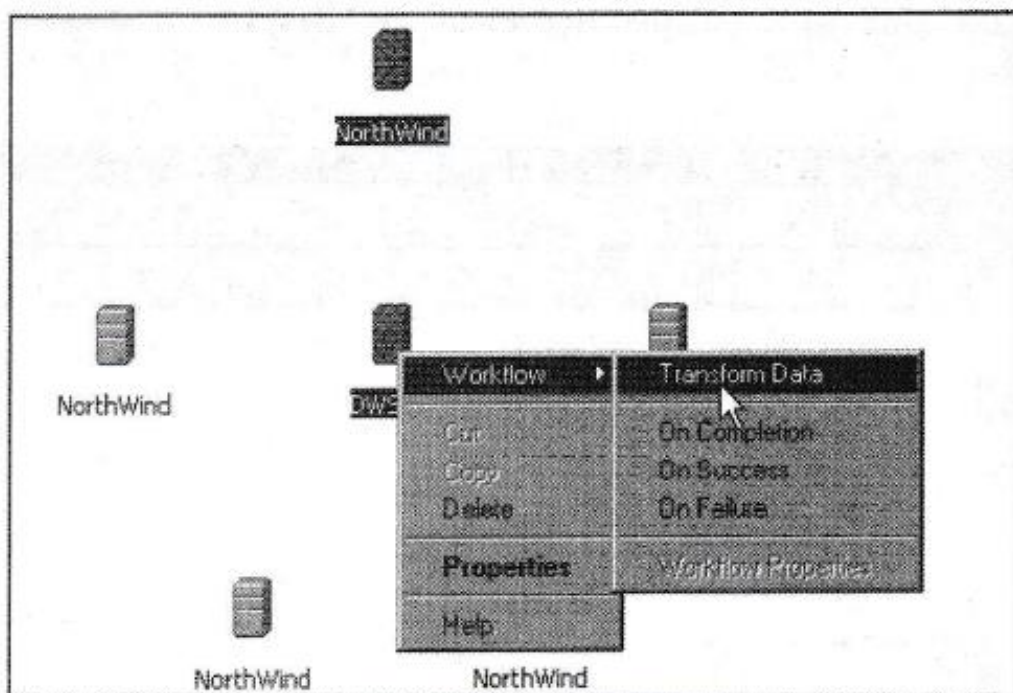


Tekan tombol OK sekali lagi untuk menampilkan koneksi pada daerah kerja. Ulangilah proses ini untuk membuat koneksi-koneksi NorthWind tambahan hingga total sebanyak lima koneksi. Gunakanlah prosedur ini untuk membuat koneksi DWStar kedua. Hasil pekerjaan Anda harus tampak seperti gambar di bawah ini.

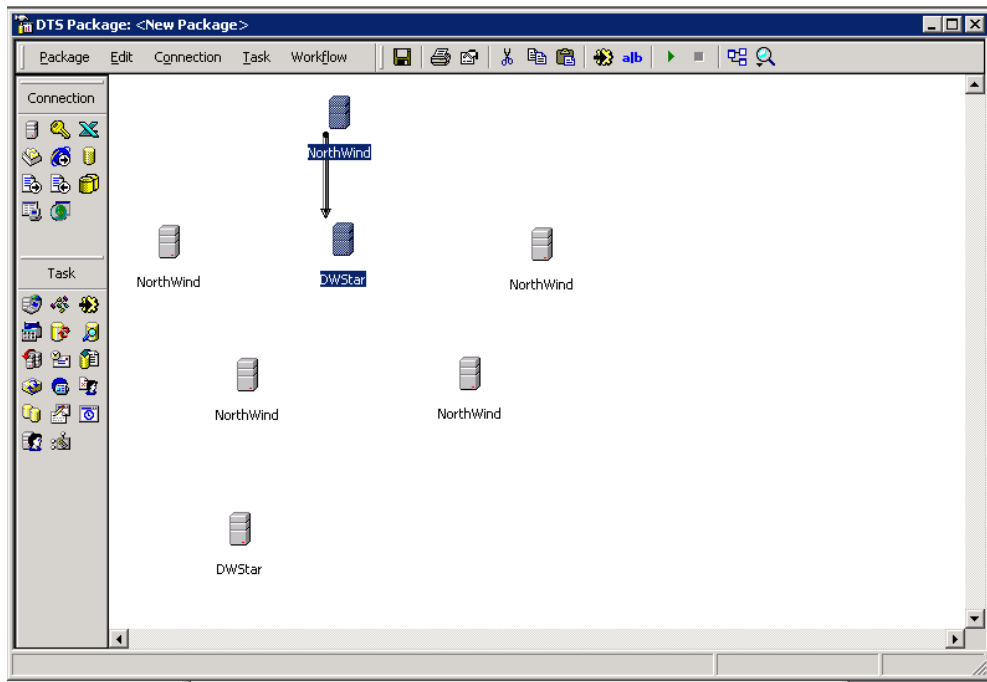


Mentransformasi Data

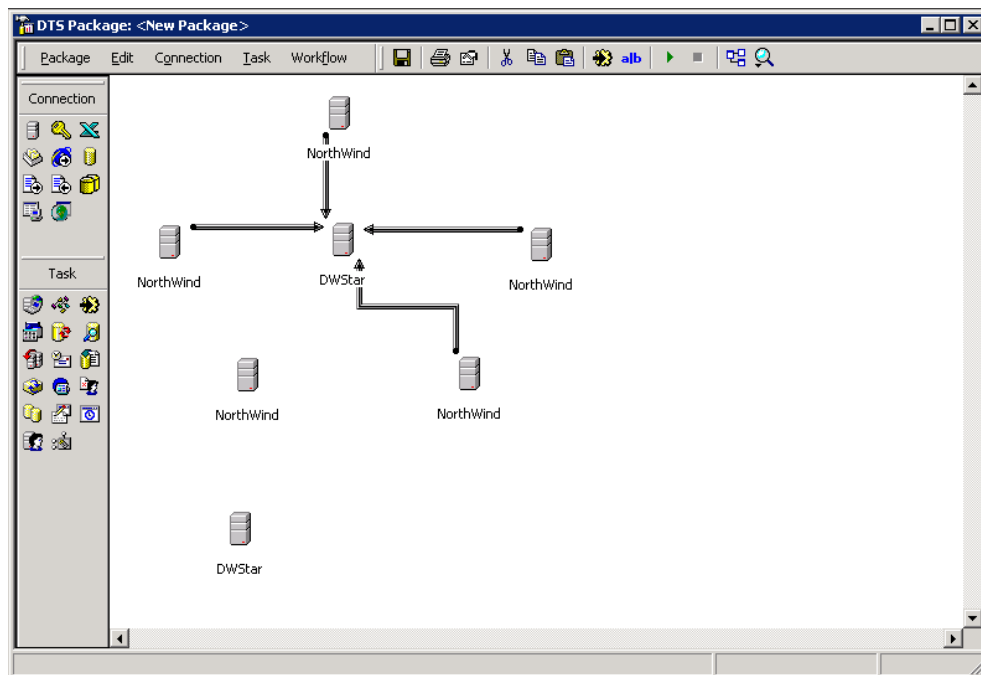
Langkah berikutnya adalah membuat link untuk aliran data. Klik pada icon NorthWind pada bagian atas layar. Lalu tahan tombol Ctrl sambil mengklik icon DWStar, seperti ditampikan pada gambar di bawah ini. Klik dengan tombol mouse kanan pada icon DWStar, lalu pilihlah Workflow | Transform Data.



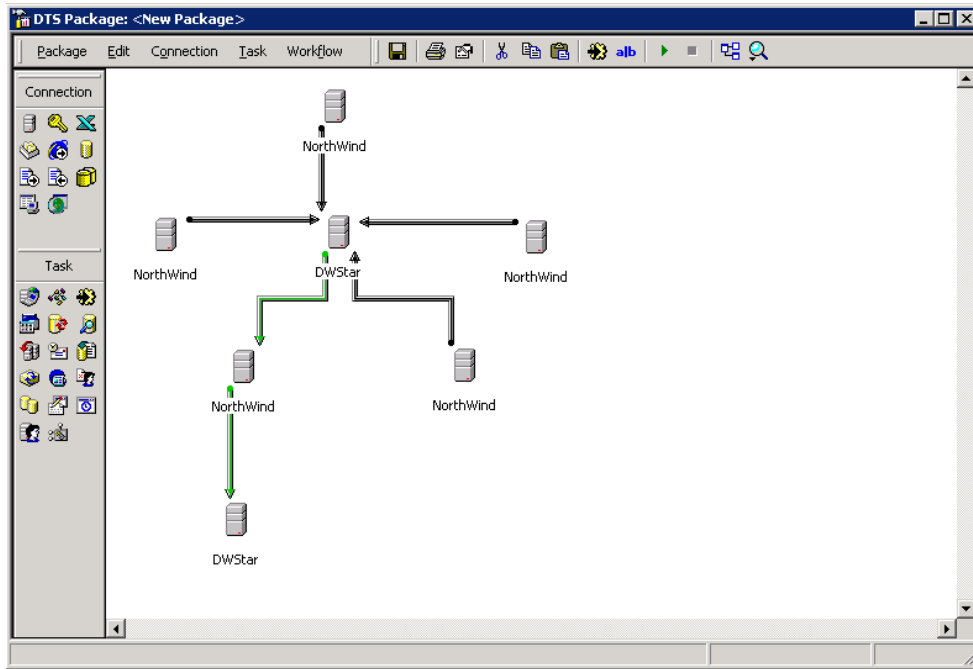
Sebuah panah akan menghubungkan icon-icon.



Ulangilah prosedur ini dengan ketiga koneksi NorthWind yang lain, seperti ditunjukkan di bawah ini.

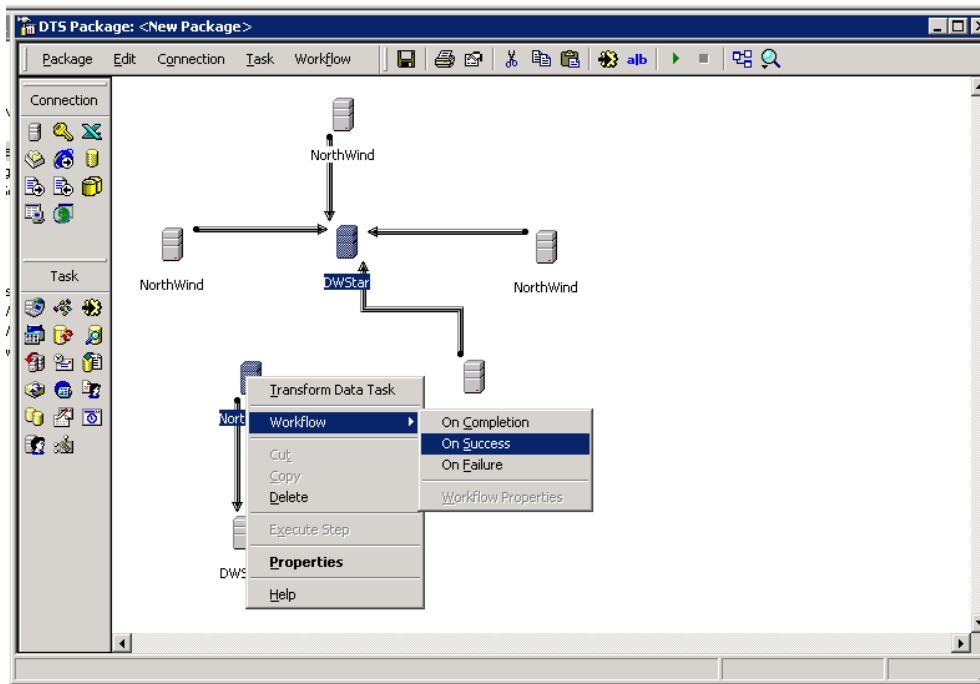


Kini kita akan membuat link lain antara icon NorthWind dan DWStar yang tersisa, seperti ditunjukkan pada gambar di bawah ini.

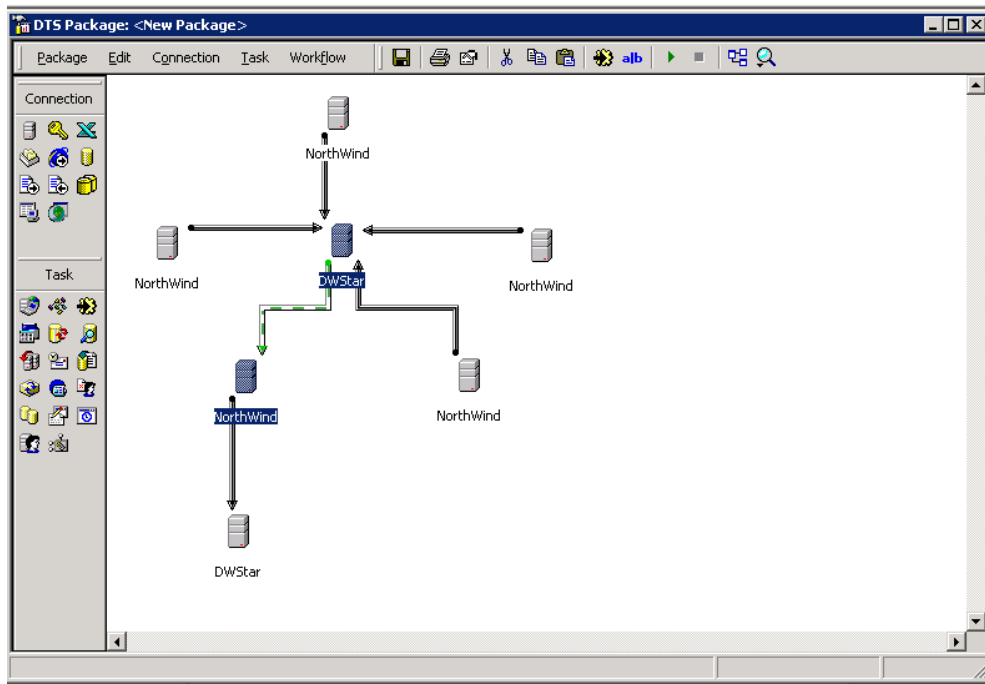


Yang terakhir, kita akan membuat link antara icon DWStar pusat dan icon NorthWind yang belum di-link .

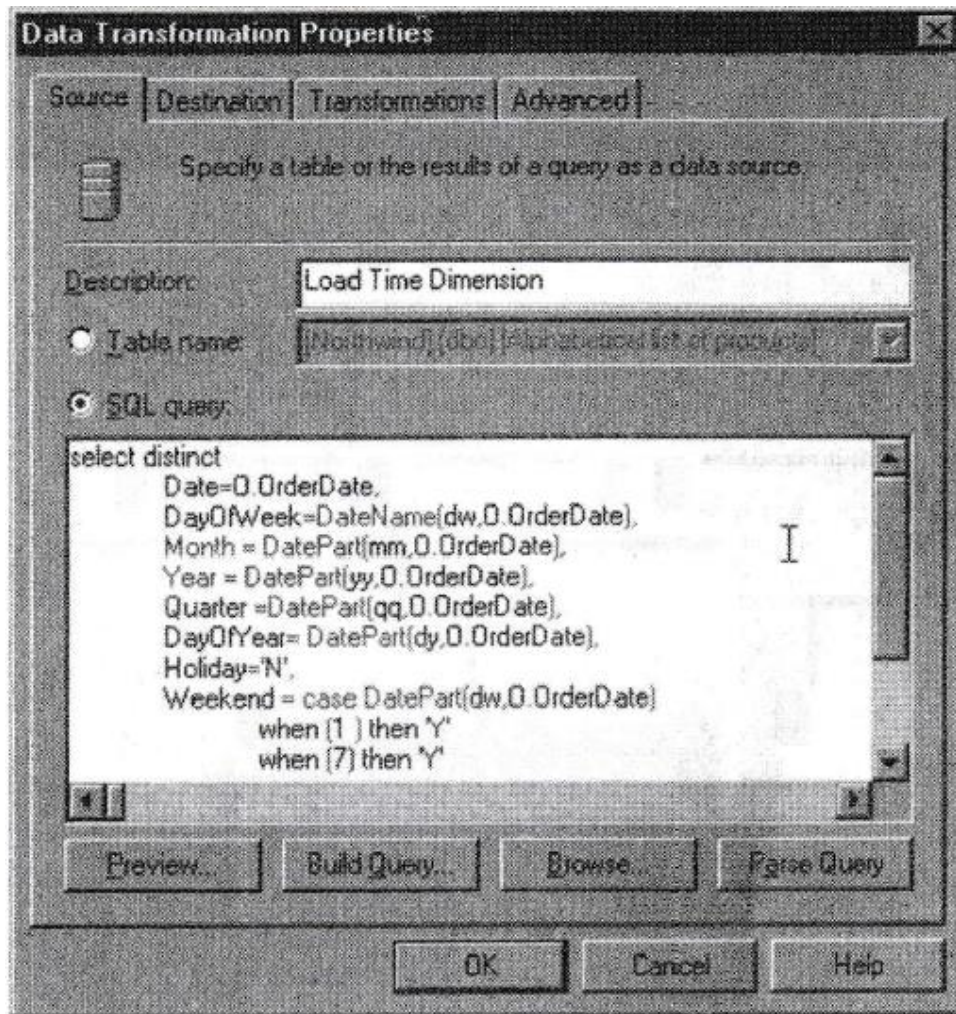
Kali ini, klik icon DWStar terlebih dahulu lalu tetap tahan tombol Ctrl, lalu klik pada NorthWind. Klik kanan pada icon NorthWind, lalu pilih Workflow | On Success.



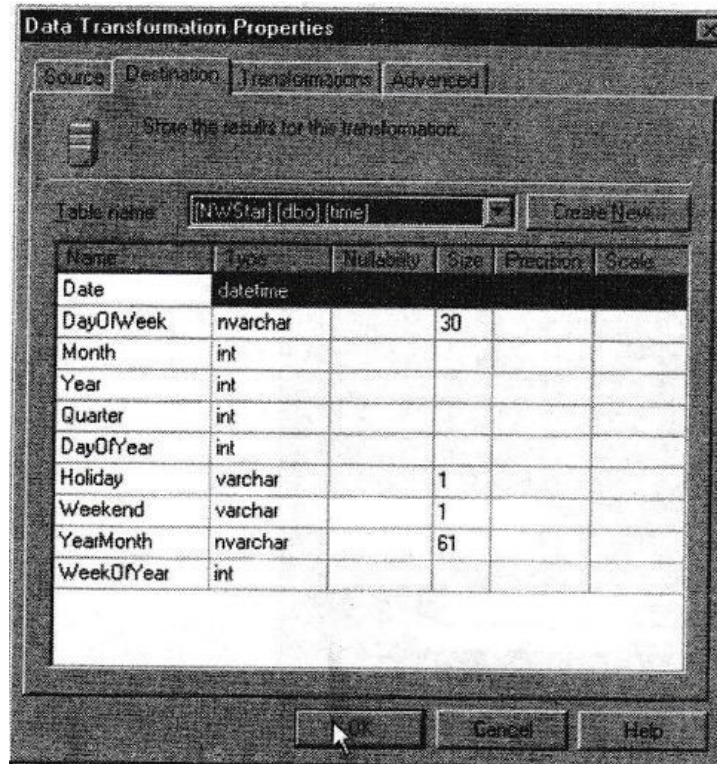
Perhatikan bahwa garis panah memiliki format yang berbeda



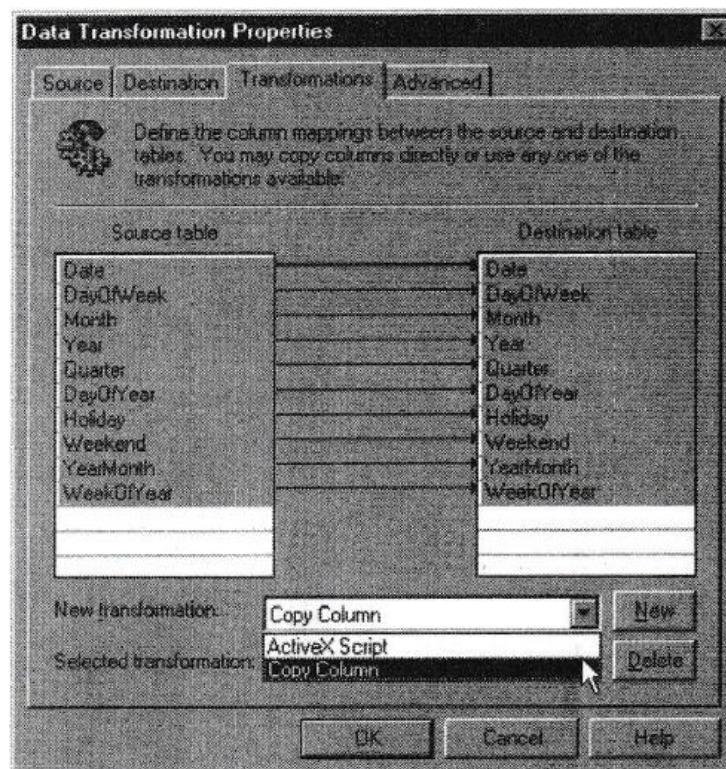
Klik ganda garis yang menghubungkan icon NorthWind di bagian atas layar dengan icon DWStar di tengah-tengah layar. Pada kotak dialog yang muncul, ketikkanlah Load Time Dimension pada kotak Description dan pilihlah item SQL Query. Pada field SQL Query, ketikkanlah query Time Dimension. Query ini bisa ditemukan di akhir dari bab ini, bersama dengan query-query lainnya.



Setelah mengetikkan isi query, klik pada tab Destination. Pada field Table Name, pilihlah NWStar.DBO.Time dari daftar. (Jika tabel ini belum dibuat, bukalah jendela dengan tombol Create New. Gantilah nama tabel dengan Time pada baris pertama perintah, dan tabel akan dibuat).Data Transformation Services



Kemudian klik pada tab Transformation lalu pilih semua kolom yang ditampilkan pada tabel Source dan Destination. Pilihlah Copy Column pada kotak New Transformation, lalu tekan tombol OK. Kolom-kolom akan dipetakan untuk menampilkan korespondensinya.

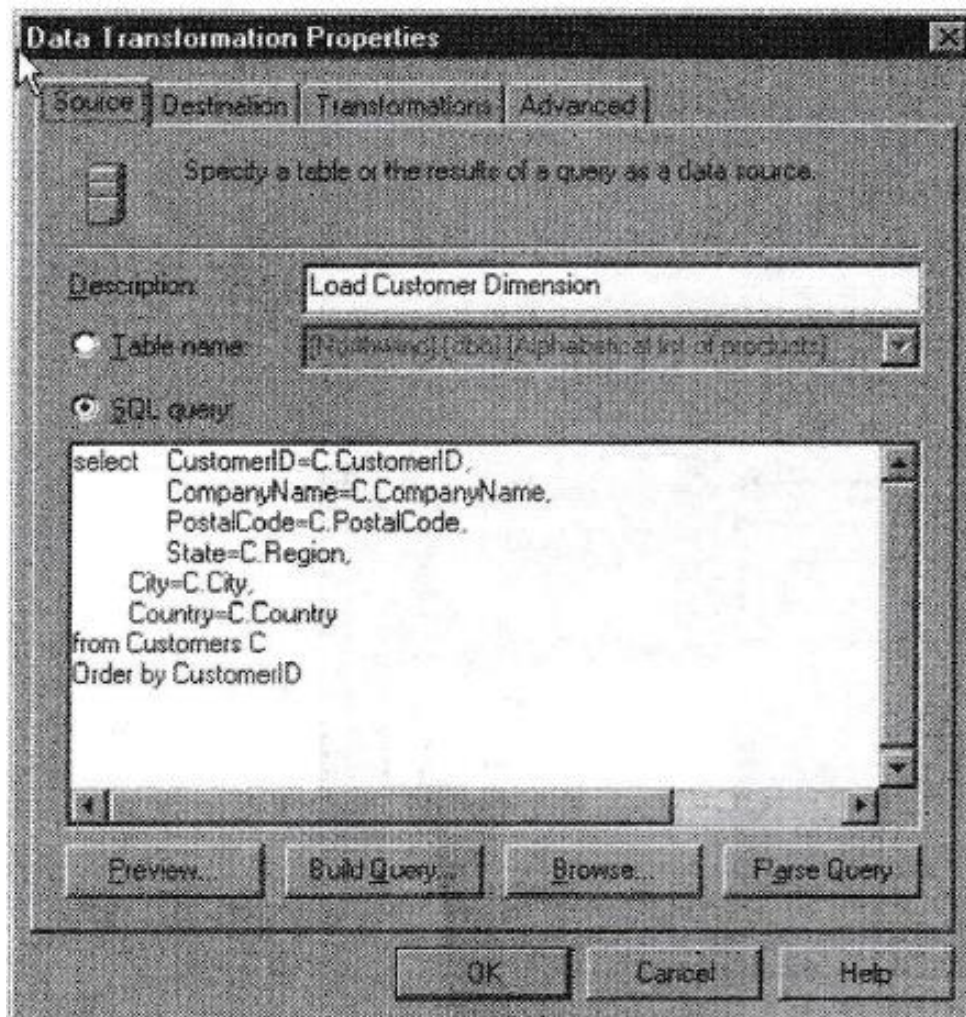


Klik OK untuk menyelesaikan.

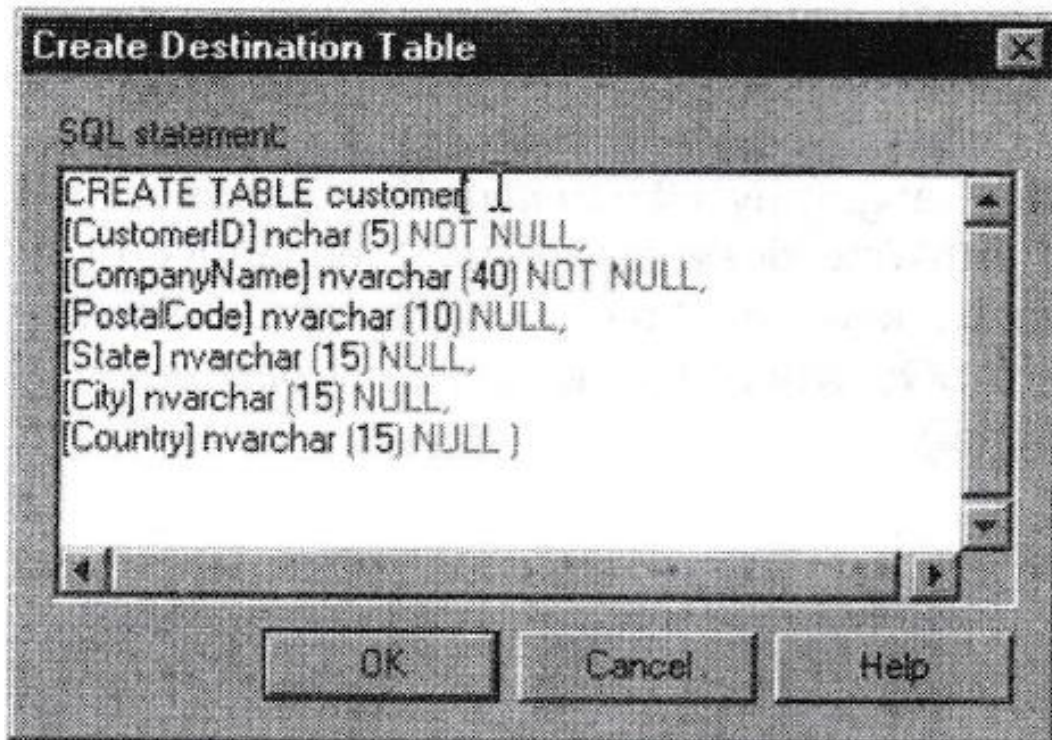
Customer Dimension

Kini kita akan membuat Customer Dimension. Klik ganda pada garis yang menghubungkan icon NorthWind pada sisi kiri daerah kerja dan icon DWStar di tengah-tengah.

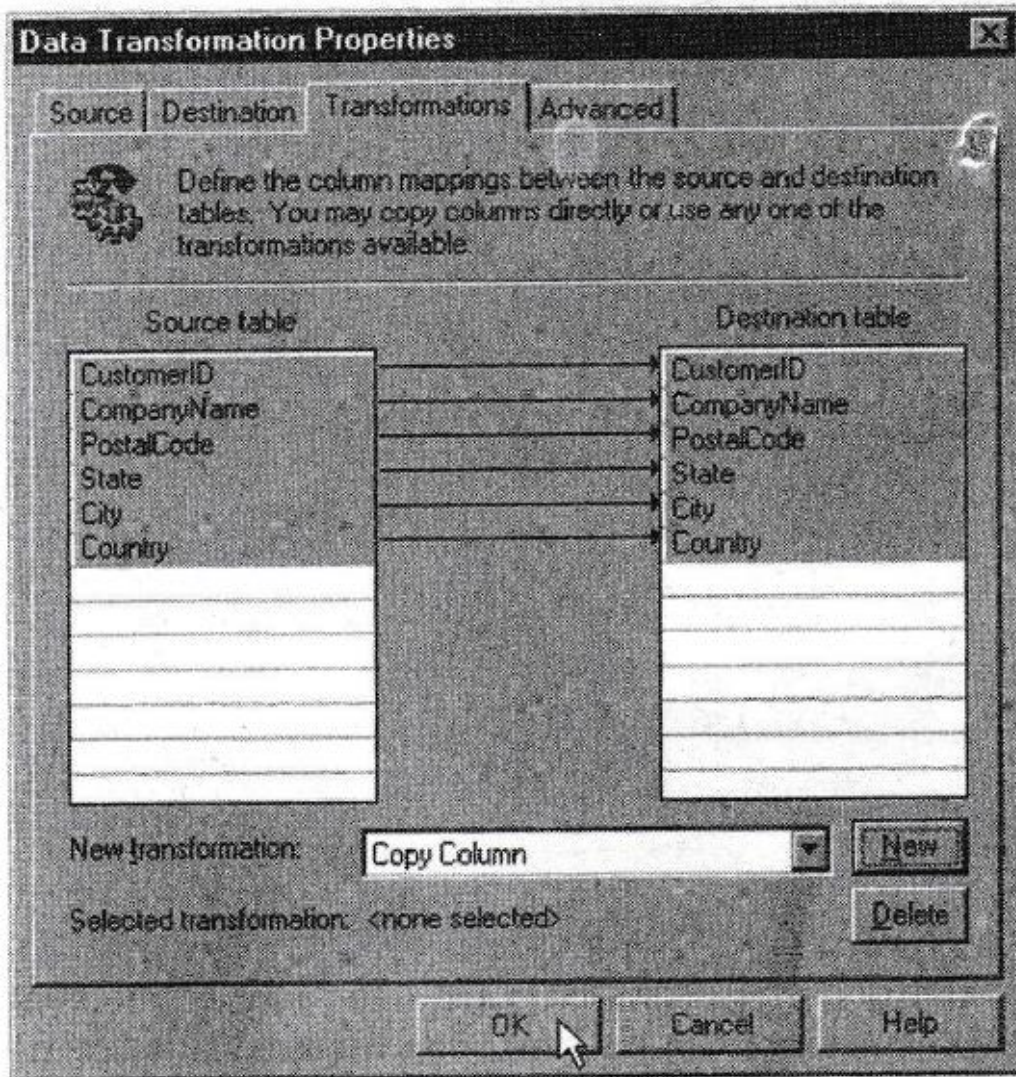
Pada field Description, ketikkanlah Load Customer Dimension. Lalu pilihlah item SQL Query dan ketikkan isi dari dimensi pelanggan yang disediakan pada akhir dari bab ini.



Pilihlah tab Destination dan NWStar.DBO.Customer atau buatlah sebuah tabel baru dengan mengklik tombol Create New, dan menyediakan nama di dalam lokasi yang sesuai pada perintah Create Table.

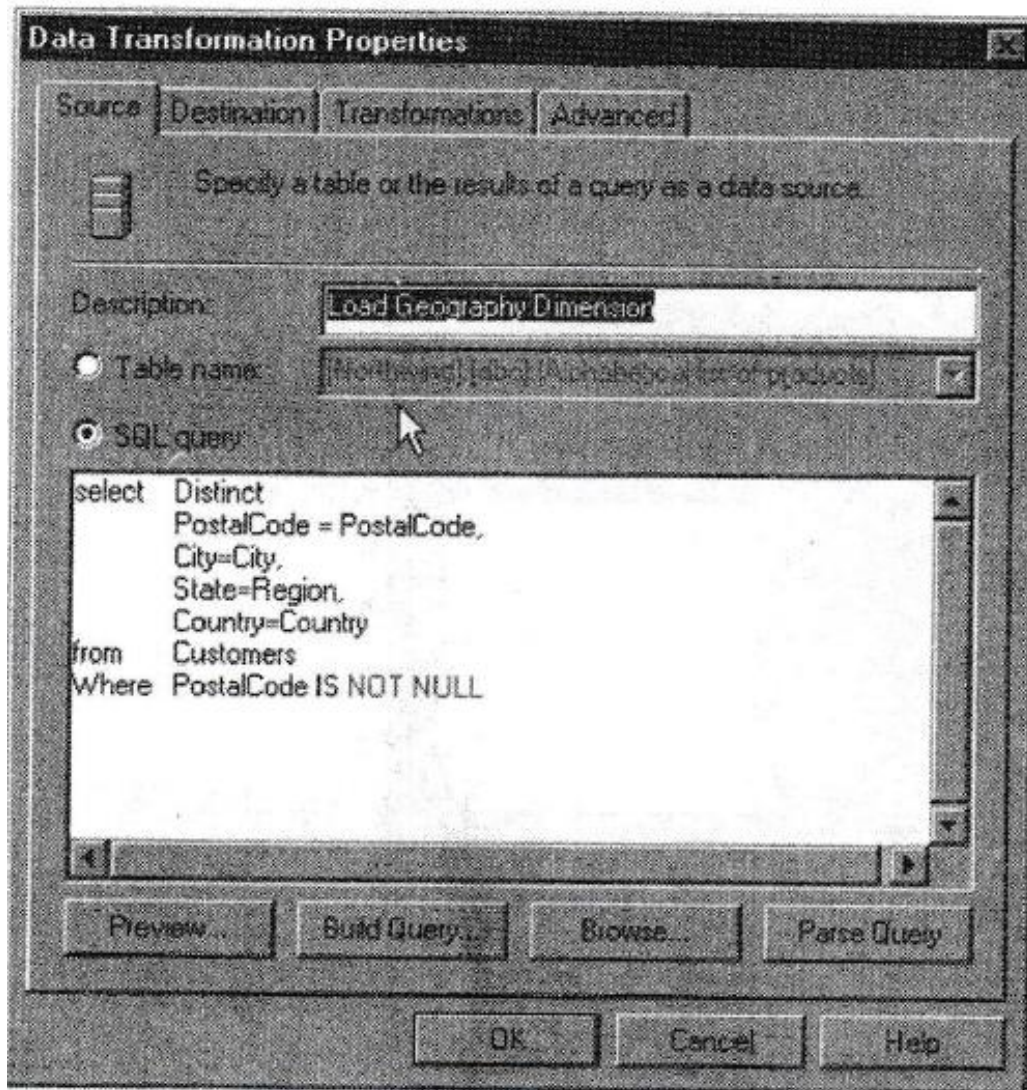


Pada tab Transformations, pilihlah semua field dari tabel Source dan Destinations. Pilih Copy Column pada kotak New Transformation lalu tekan OK untuk memetakan kolom-kolom.

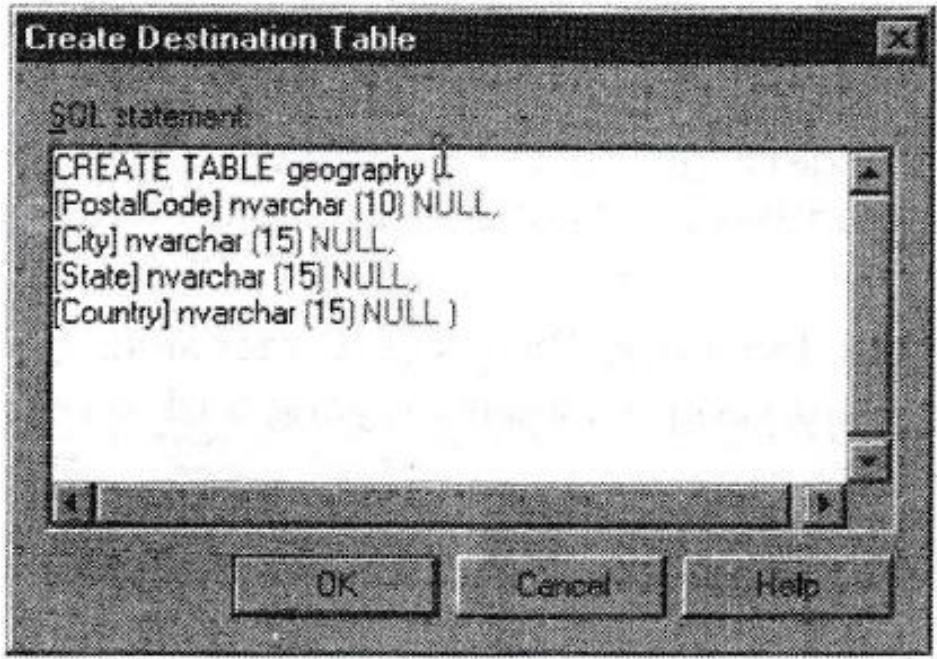


Geography Dimension

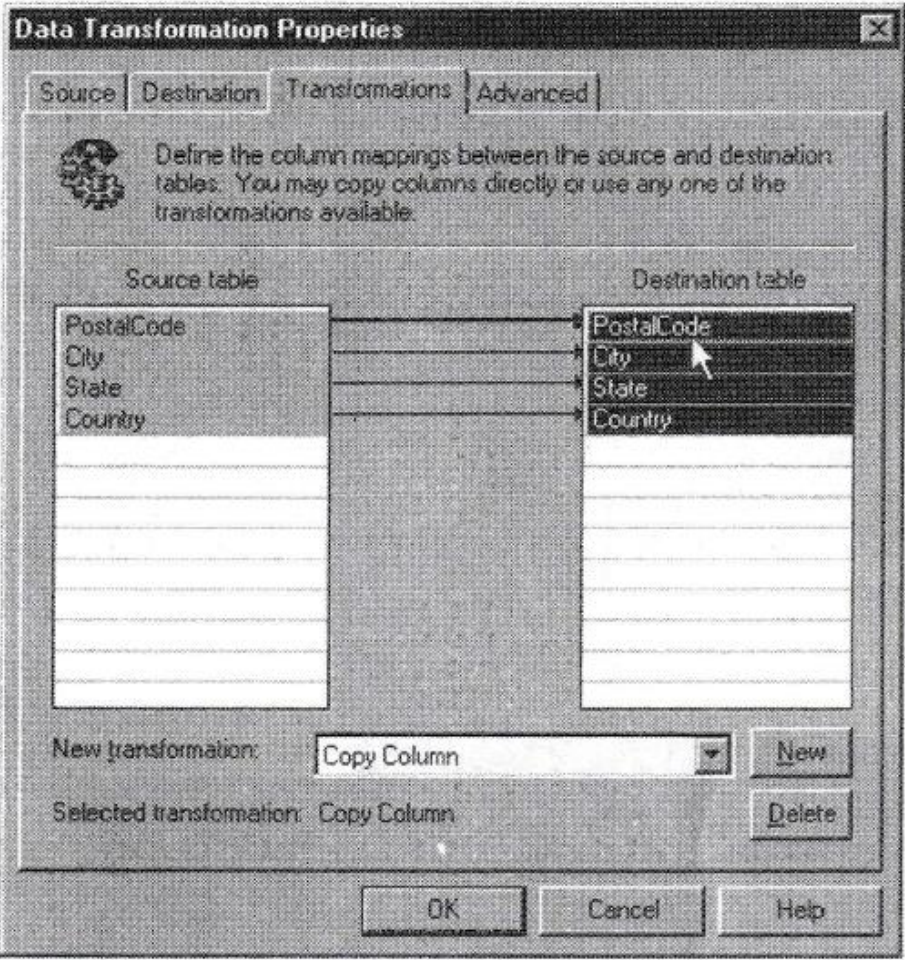
Kini kita akan membuat Geography Dimension. Klik ganda pada garis yang menghubungkan icon NorthWind di sebelah kanan dengan icon DWStar. Pada kotak dialog yang muncul, ketikkan Load Geography Dimension pada field Description. Pilihlah item SQL Query lalu ketikkan isi dari query yang sesuai (terdapat pada akhir bab ini).



Klik pada tab Transformations. (Jika tabel ini belum dibuat, bukalah jendela dengan tombol Create New. Gantilah nama tabel dengan Geography pada baris pertama perintah, dan tabel akan dibuat).



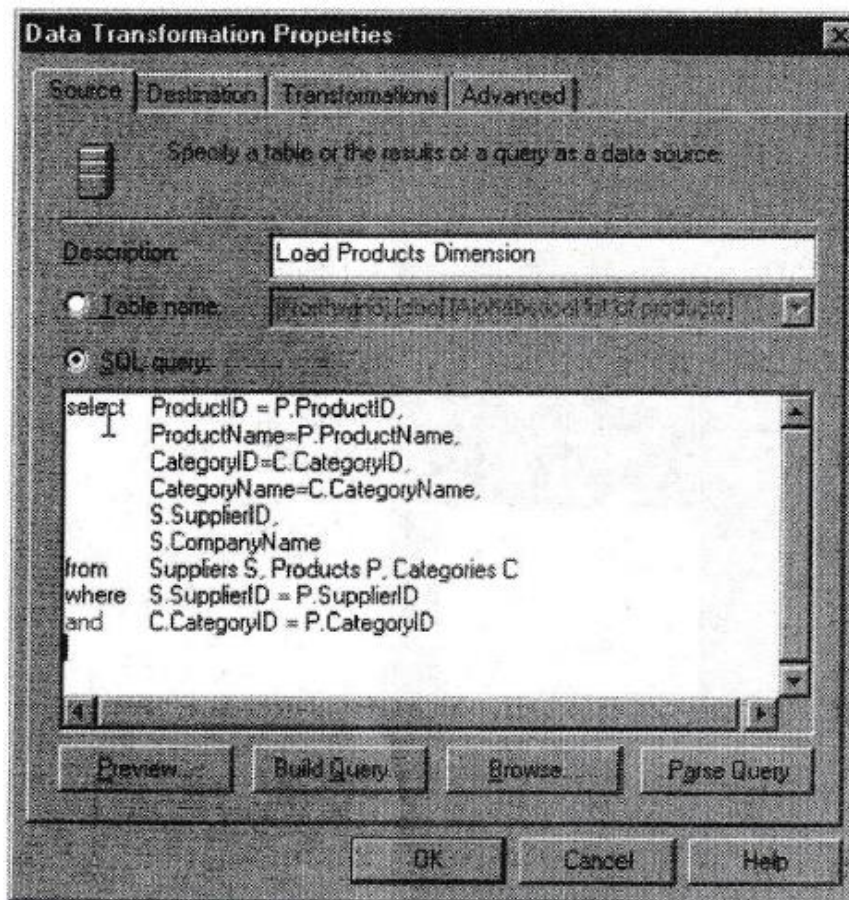
Pilihlah semua field pada tabel Source dan Destinations. Lalu pilihlah Copy Column pada kotak New Transformation lalu tekan OK.



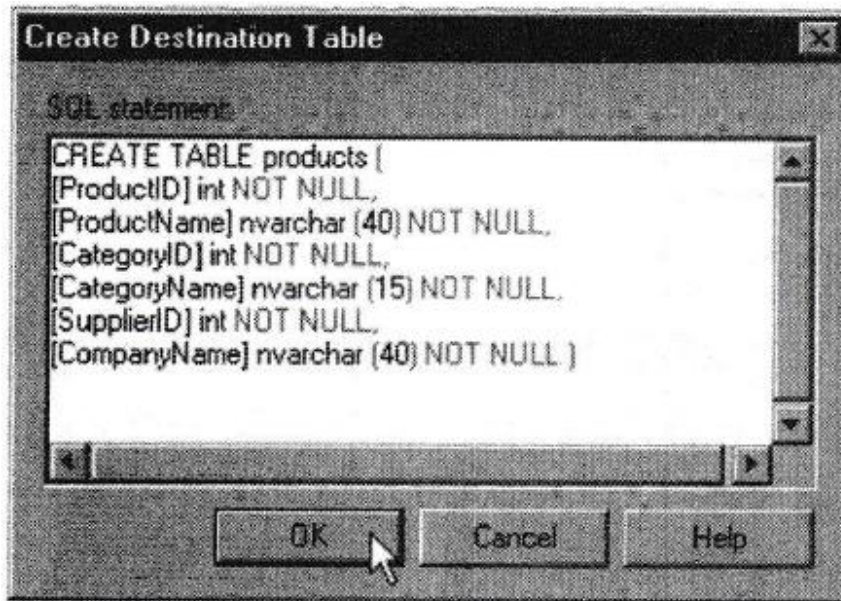
Kolom-kolom dipetakan dengan menampilkan korespondensinya.

Products Dimension

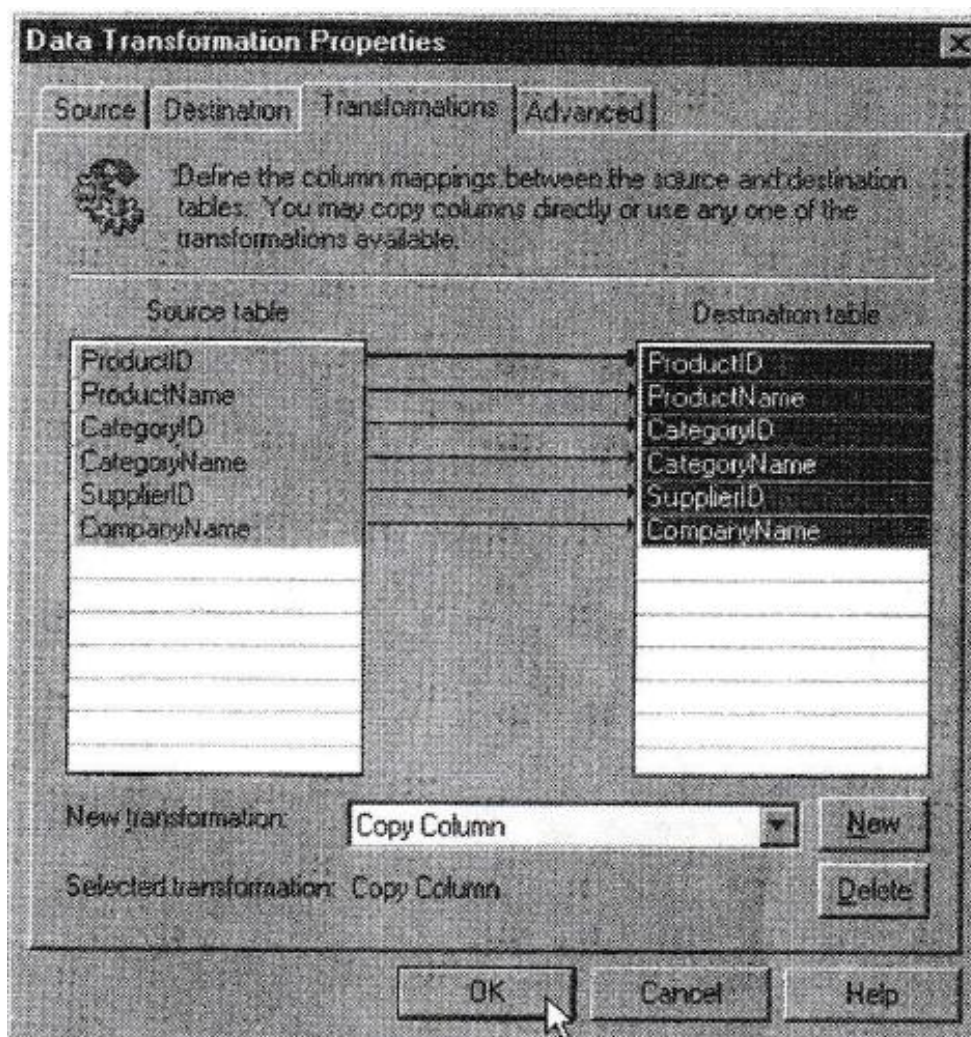
Dimensi ini dibuat dengan mengklik ganda pada garis yang menghubungkan icon NorthWind di sebelah kanan bawah layar dengan icon DWStar di tengah-tengah layar. Pada item Description, ketikkanlah Load Products Dimension. Pilihlah item SQL Query dan ketikkanlah query yang bersesuaian yang terdapat di akhir dari bab ini.



Pada tab Destination, pilihlah NWStar.DBO.Products pada kotak Table Name. (Anda bisa membuat tabel ini jika belum ada, dengan mengklik tombol Create New).



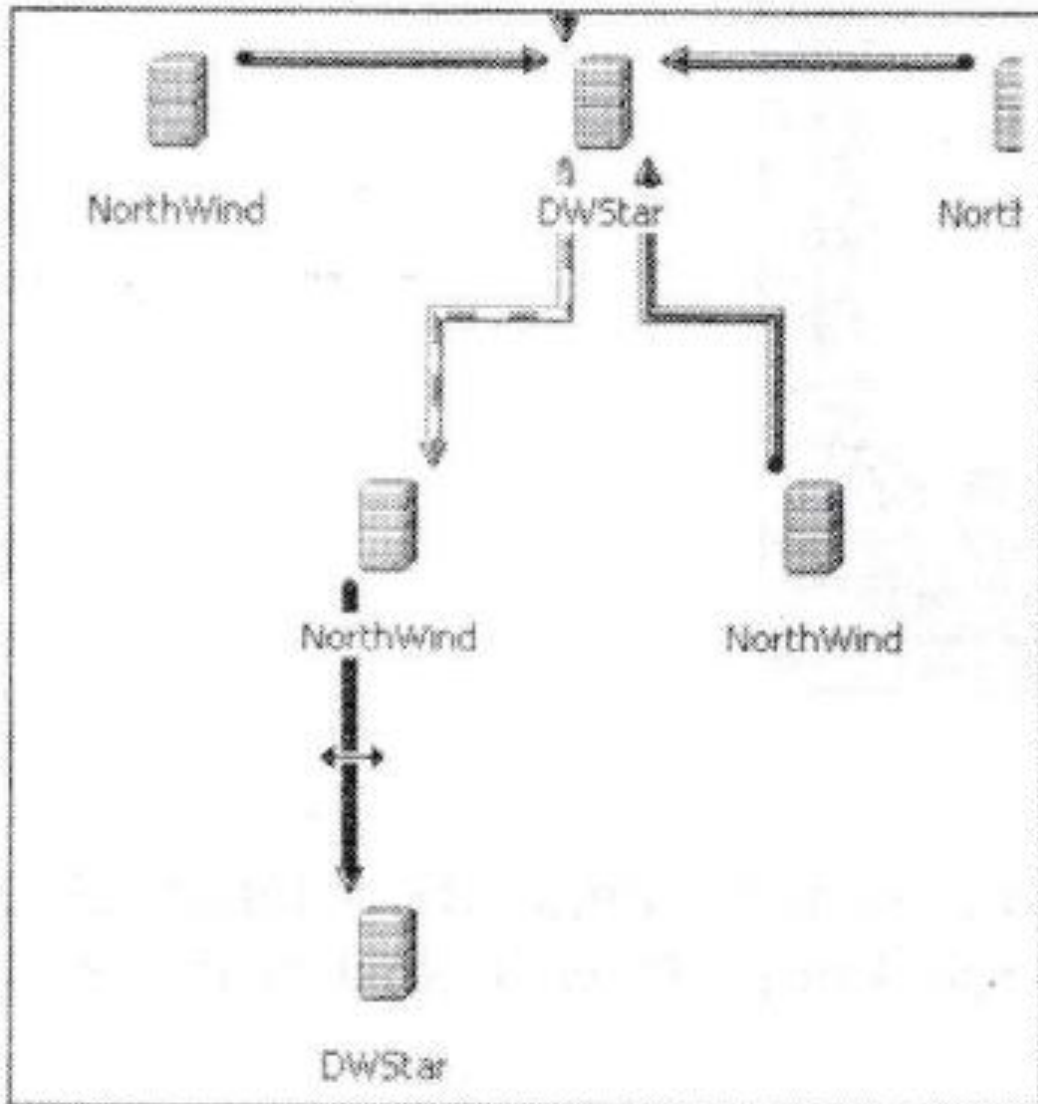
Pada tab Transformations, pilihlah semua field dari tabel Source dan Destination. Pilih Copy Column pada kotak New Transformation. Tekanlah tombol OK.



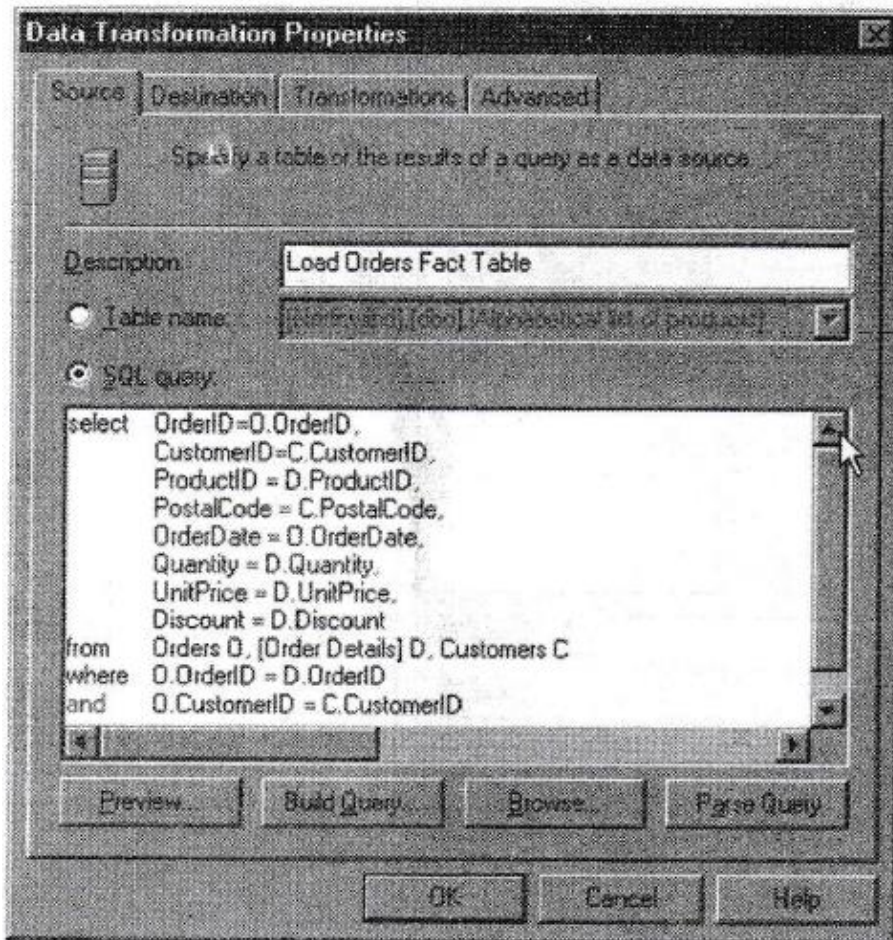
Tekan tombol OK untuk menyelesaikan.

Tabel Orders Fact

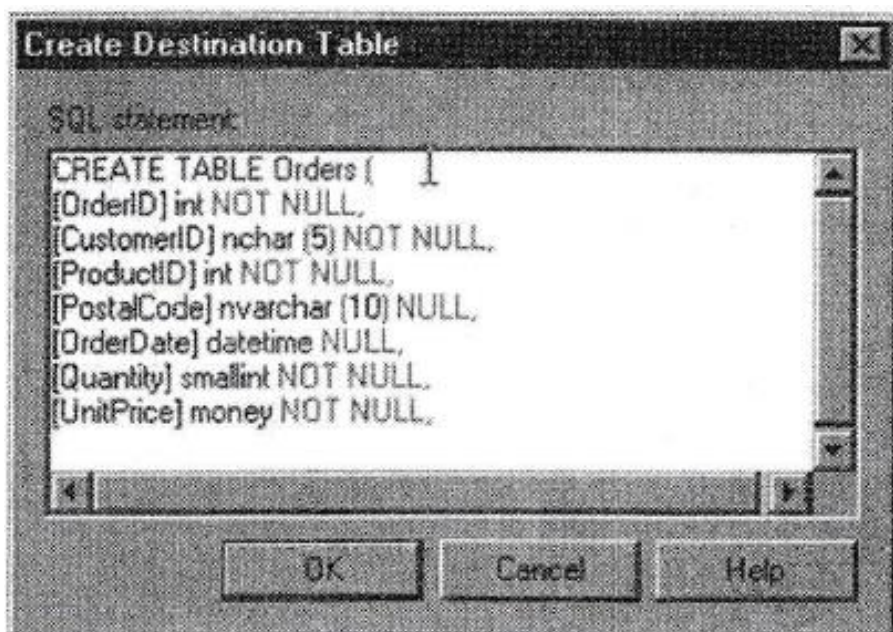
Berikutnya kita akan membuat link antara icon DWStar di bagian bawah layar dengan icon NorthWind di dekatnya.



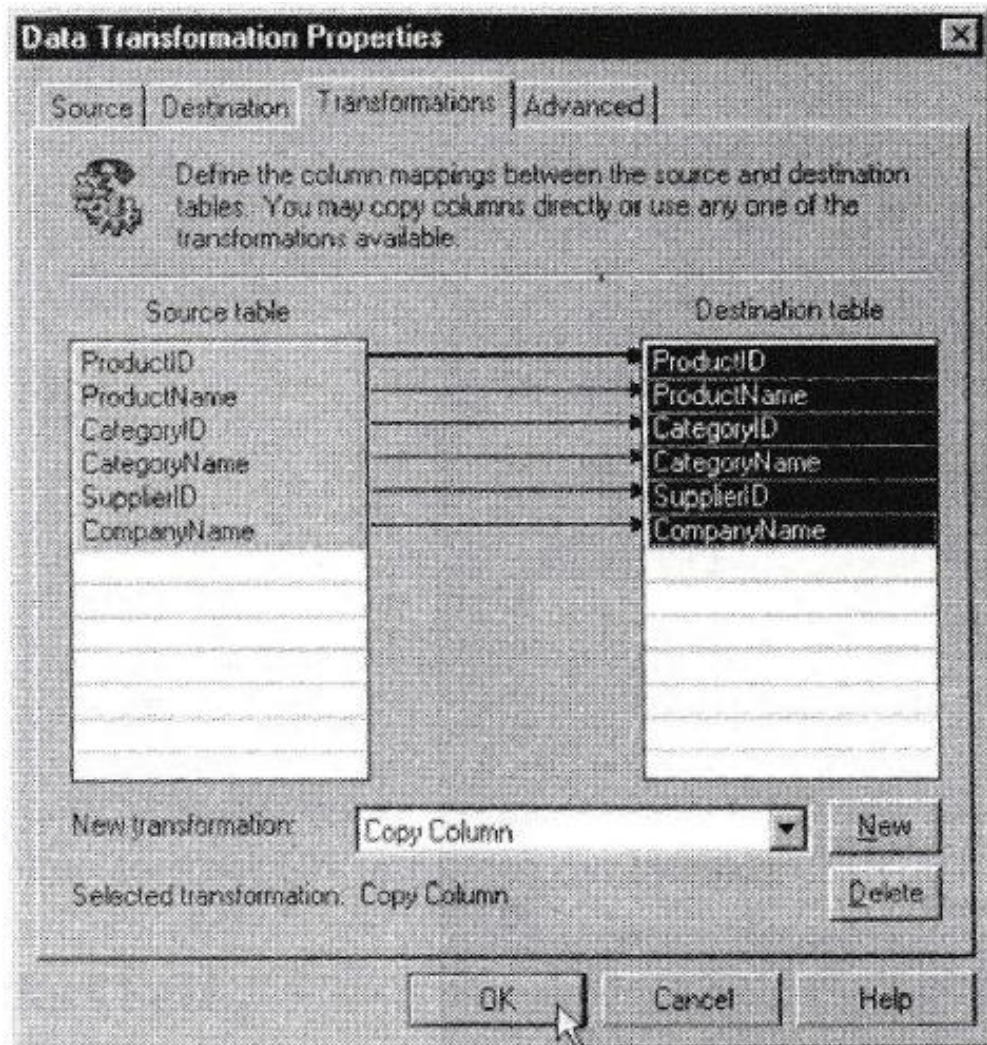
Klik ganda pada garis yang menghubungkan icon-icon tersebut; kotak dialog Data Transformation Properties akan muncul. Pada tab Source, ketikkanlah Load Orders Fact Table pada field Description, pilih SQL Query, lalu ketikkan kode Orders Fact Table yang terdapat di akhir dari bab ini pada jendela SQL Query .



Pada tab Destination, pilih atau buatlah NWStar.DBO.Orders (Anda bisa membuat tabel ini jika belum ada, dengan mengklik tombol Create New).

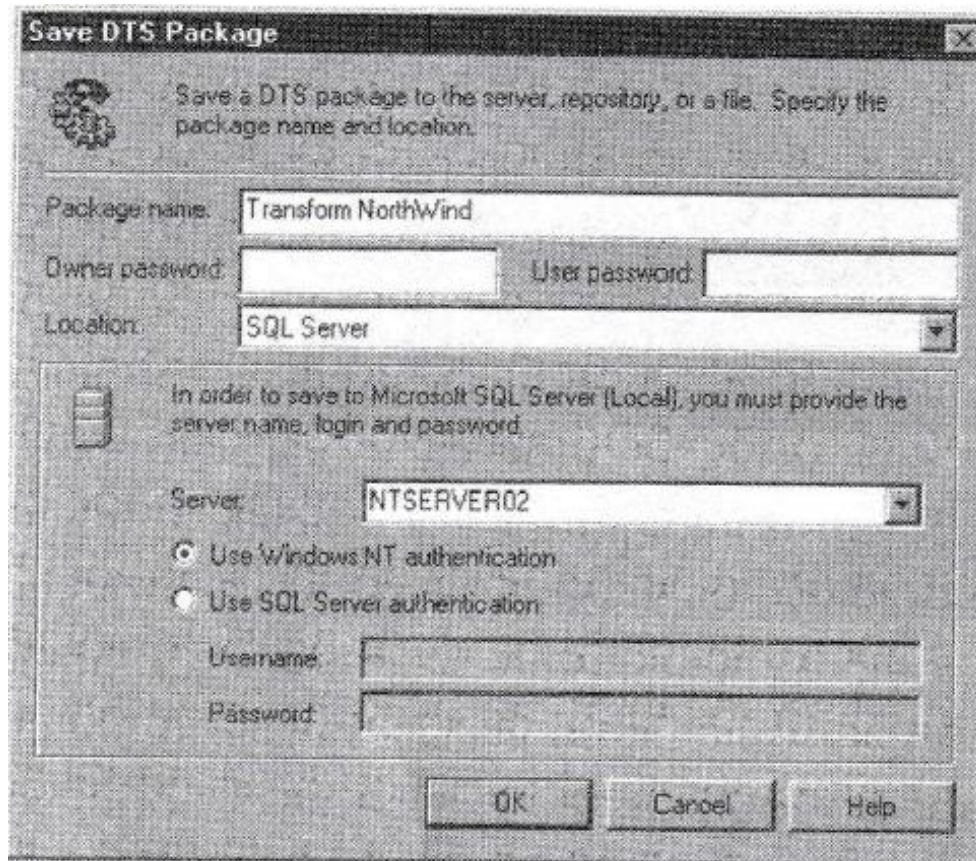


Pada tab Transformations, pilihlah semua field dari tabel Source dan Destination. Pilih Copy Column pada kotak New Transformation. Tekanlah tombol OK.



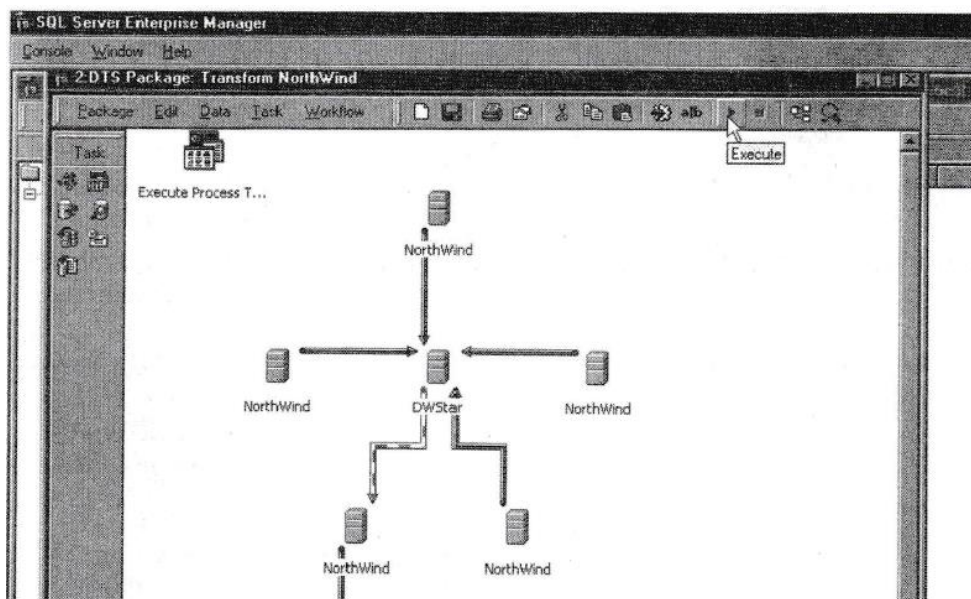
Kita hampir selesai. Yang kita butuhkan hanyalah menyimpan paket DTS. Untuk melakukan ini, klik pada tombol Save Package pada toolbar.

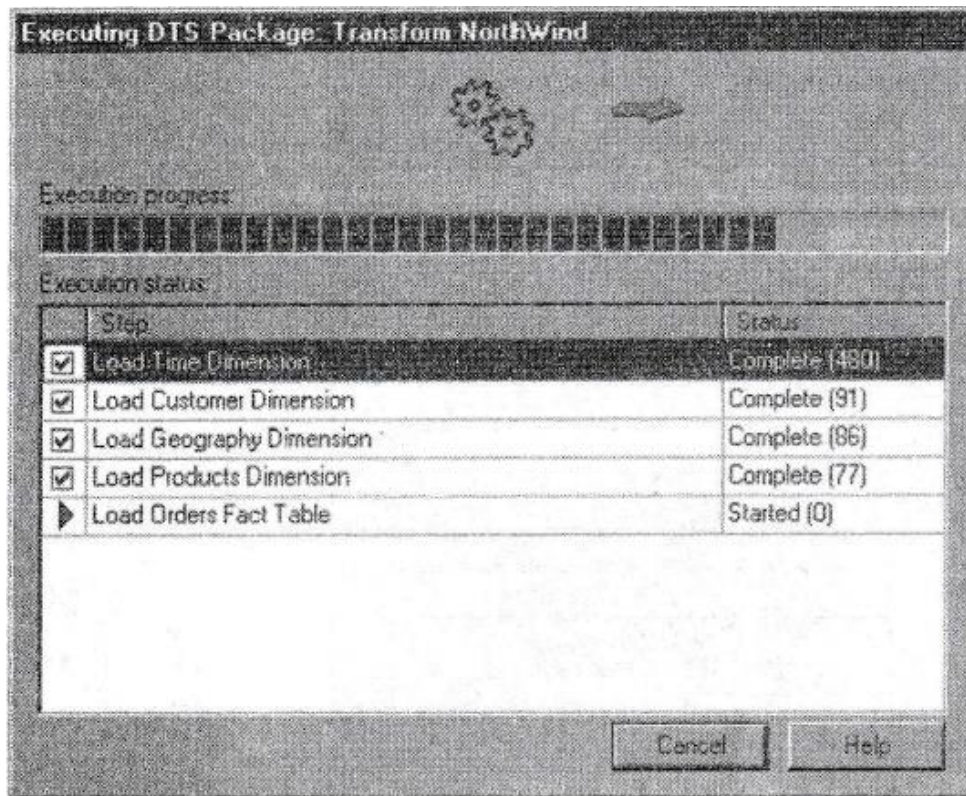
Ketikkanlah Transform NorthWind sebagai nama paket pada kotak dialo Save DTS Package, lalu tekan tombol OK.



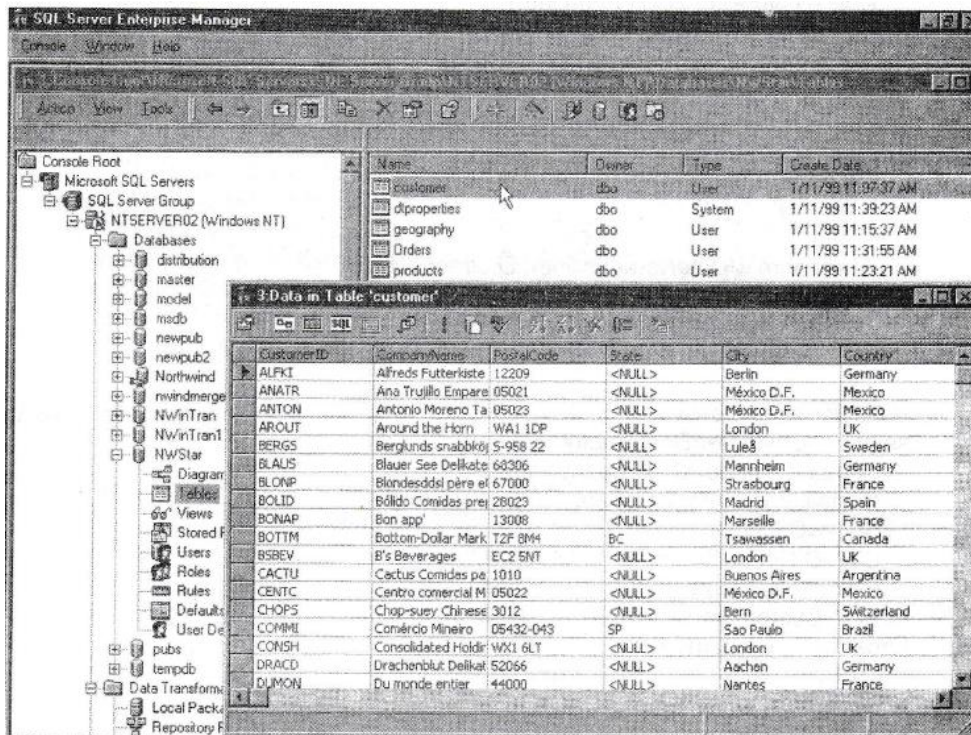
Menjalankan Paket DTS

Tekan tombol Execute pada toolbar.





Jika ada yang salah, status eksekusi akan menandakan error. Kita baru saja mentransformasi sebuah database. Pada contoh ini, kita telah membuat sebuah salinan yang tepat dari database pertama ke dalam database kedua, dengan menggunakan SQL Server itu sendiri sebagai sumbernya. Lihatlah gambar di bawah ini pada halaman berikut ini.



Prosedur ini bisa digunakan untuk mengumpulkan data dari database Oracle, Sybase, atau DB2 dan untuk membuat database SQL baru atau database lainnya dengan format yang didukung oleh produk.

Berikut ini adalah query yang harus diketikkan pada field SQL Query untuk setiap query yang sesuai.

*****Queries to DWStar schema*****

--Time Dimension

select distinct

Date=O.OrderDate,
 DayOfWeek=DateName(dw,0.OrderDate),
 Month = DatePart(mm ,0.OrderDate),
 Year = DatePart(yy,O.OrderDate),
 Quarter =DatePart(qq,0 .OrderDate),
 DayOfYear= DatePart(dy,O.OrderDate),
 Holiday='N',

```

Weekend = case DatePart(dw,O.OrderDate)
    when (1 ) then 'Y'
    when (7) then 'Y'
    else 'N'
    end,
YearMonth = DateName(month, O.OrderDate) + '_' +
    DateName(year, O.OrderDate ),
WeekOfYear=DatePart(wk,O.OrderDate)
from Orders 0

```

-- Customer Dimension

```

Select      CustomerID=C.CustomerID,
            CompanyName=C.CompanyName,
            PostalCode=C.PostalCode,
            State=C.Region,
            City=C.City,
            Country=C.Country
From        Customers C
Order by    CustomerID

```

-- Geography Dimension

```

select      Distinct
            PostalCode = PostalCode,
            City=City,
            State= Region,
            Country=Country
From        Customers
Where       PostalCode IS NOT NULL

```

-- Products Dimension

```

select      ProductID = P.ProductID,
            ProductName=P.ProductName,
            CategoryID=C.CategoryID,
            CategoryName=C.CategoryName,

```

```

S.SupplierID,
S.CompanyName
from Suppliers S,Products P, Categories C
where S.SupplierID = P.SupplierID
and C.CategoryID = P.CategoryID

```

-- Orders Fact Table

```

select OrderID=O.OrderID,
       CustomerID=C.CustomerID,
       ProductID = D.ProductID,
       PostalCode = C.PostalCode,
       OrderDate = O.OrderDate,
       Quantity = D.Quantity,
       UnitPrice = D.UnitPrice,
       Discount = D.Discount
from Orders O, [Order Details] D, Customers C
where O.OrderID = D.OrderID
and O.CustomerID = C.CustomerID
order by OrderID, ProductID

```

Kesimpulan

Bab ini memperkenalkan Data Transformation Services, serta mendemonstrasikan pembuatan dimensi DTS. Bab berikutnya menampilkan cara mengekspor data dengan menggunakan DTS Export Wizard.

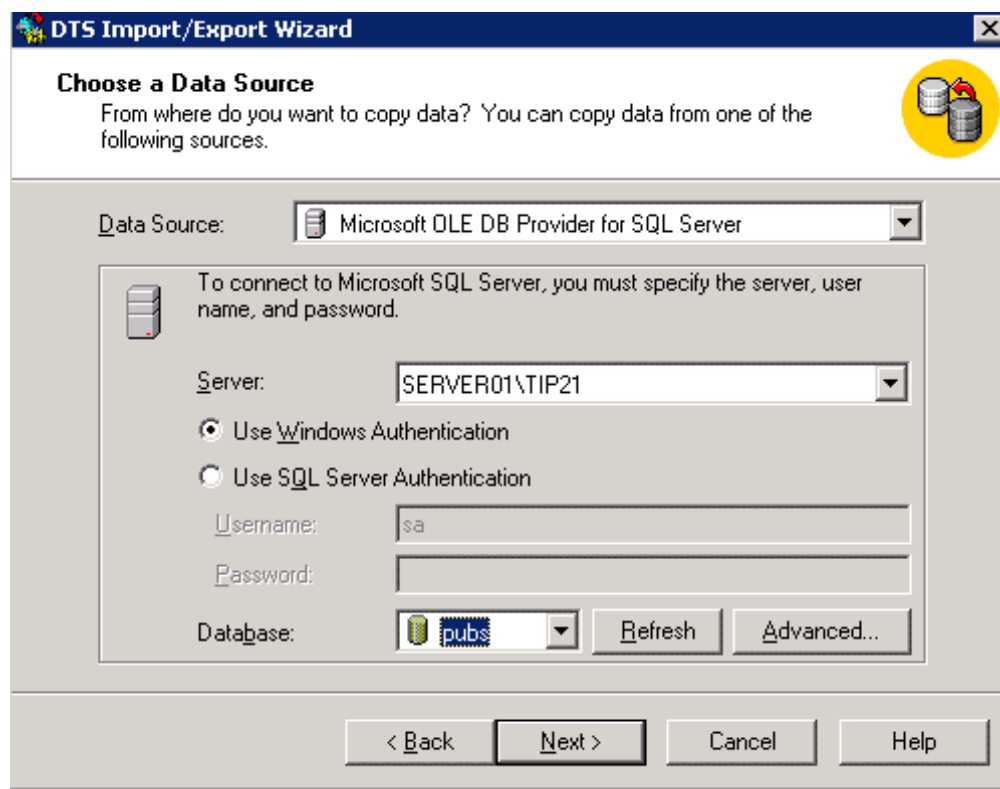
BAB 16

Contoh Praktis DTS

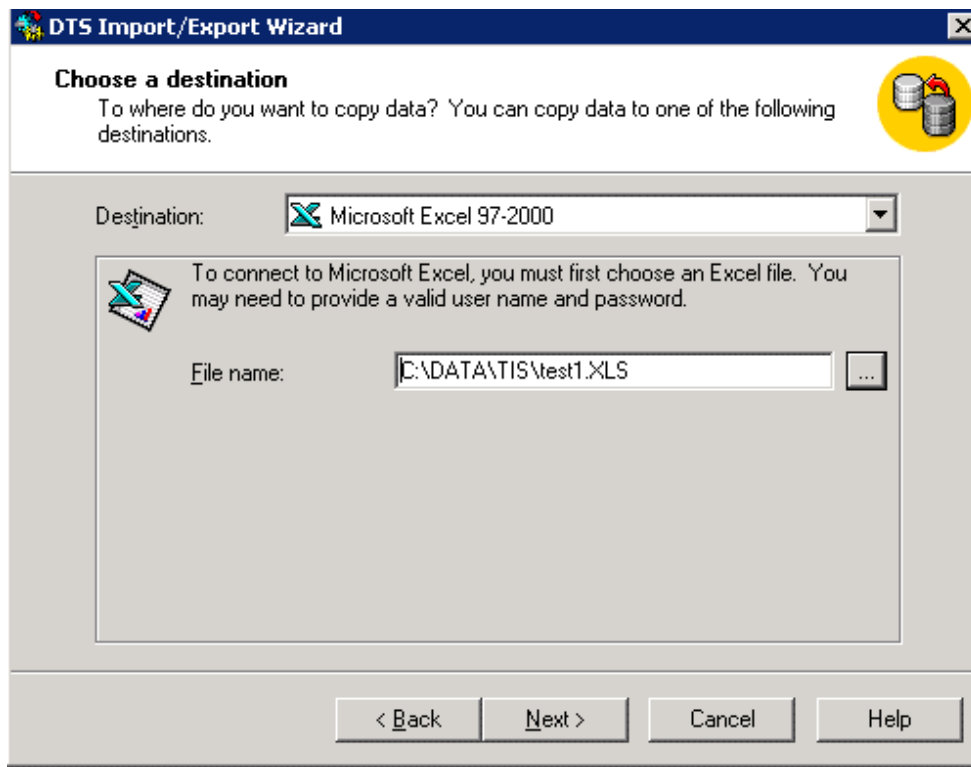
Pada bab ini, kita akan melihat cara mengekspor data dari SQL Server ke file bertipe lain dengan menggunakan Export Wizard .

Menggunakan Export Wizard

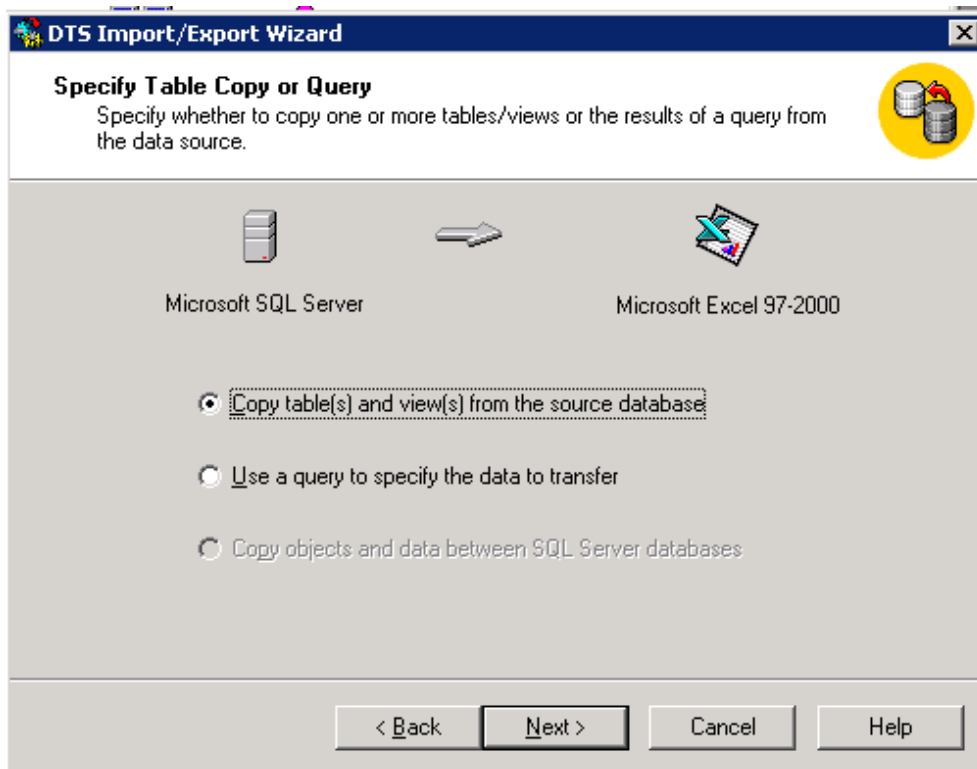
Pertama-tama, perbesarlah pohon database dan klik tombol mouse kanan pada folder Data Transformation Services. Lalu klik All Tasks | Export Data. Tekan tombol Next pada layar pertama wizard. Pada layar kedua, pilihlah Microsoft OLE DB Provider for SQL Server pada field Source. Pilih item Use Windows NT Authentication, lalu pilih database Pubs pada field Database.



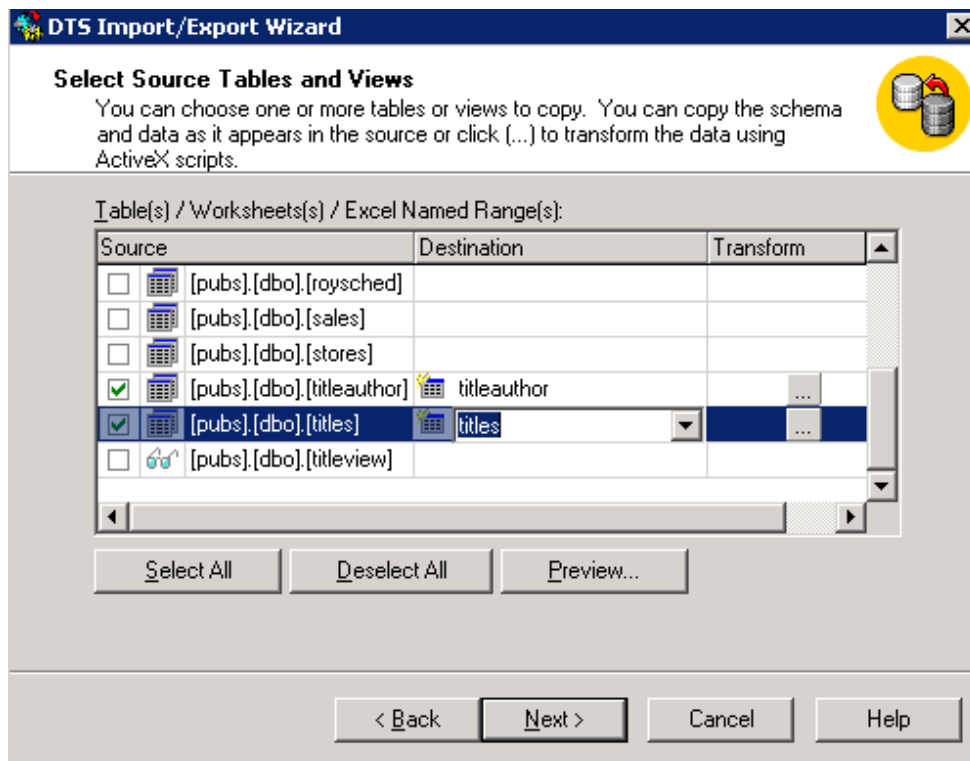
Tekan Next. Pada field Destination, pilihlah Microsoft Excel 8.0 (Excel 97). Pada field File Name, pilihlah nama spreadsheet yang akan menerima data. Pada contoh ini, kita telah membuat sebuah spreadsheet kosong bernama testimport.xls pada direktori MSSQL\Data. Tekan Next.



Pada latihan ini, kita ingin meng-copy semua tabel dari database. Sebagai pilihan, kita bisa membuat sebuah query untuk menyaring tabel dan baris yang akan di-copy. Pilihlah Copy Table(s) dari Source Database lalu tekan Next.



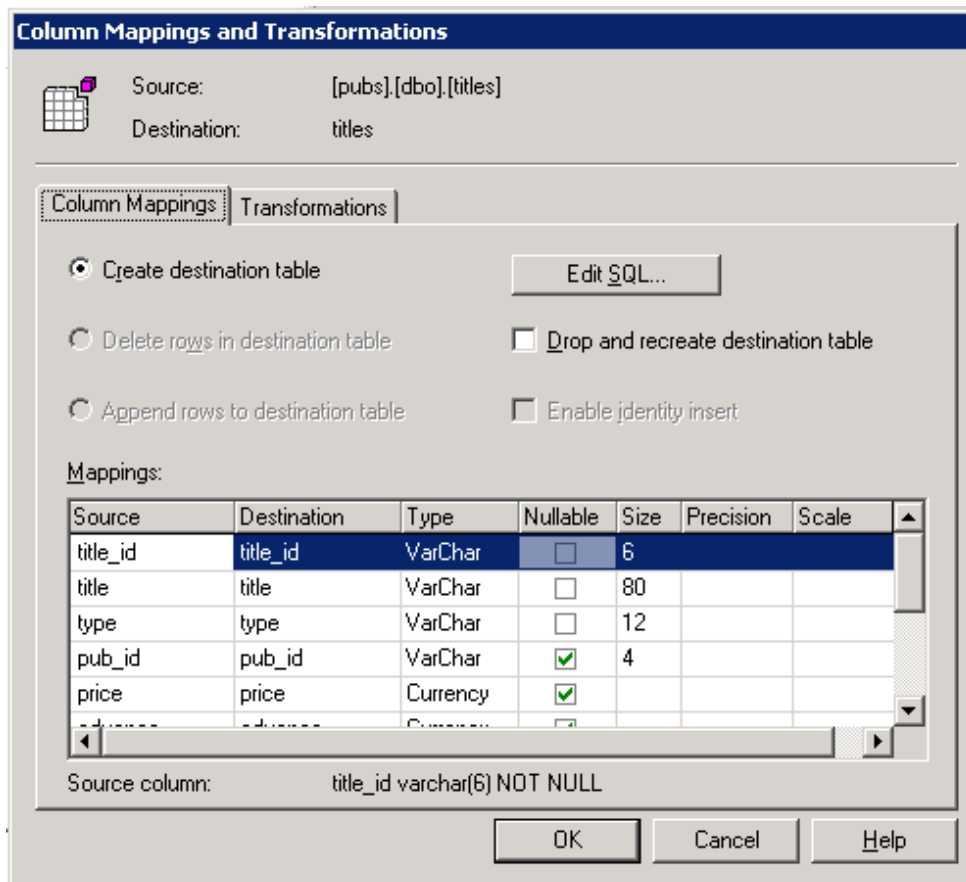
Pada latihan ini, kita akan memilih tabel Authors, Titles, dan Titleauthor.



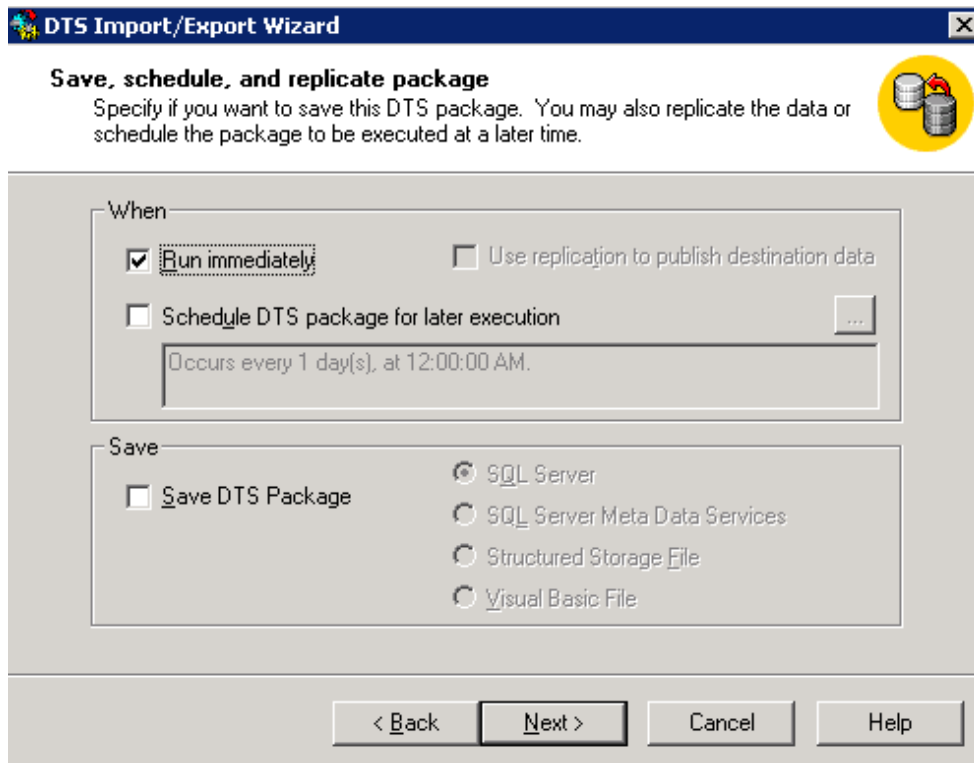
Tekan Next. (Jika Anda ingin mentransformasi sebagian kolom selama proses copy,

tekan tombol pada kolom Transform. Kita tidak akan menampilkan proses tersebut pada latihan ini).

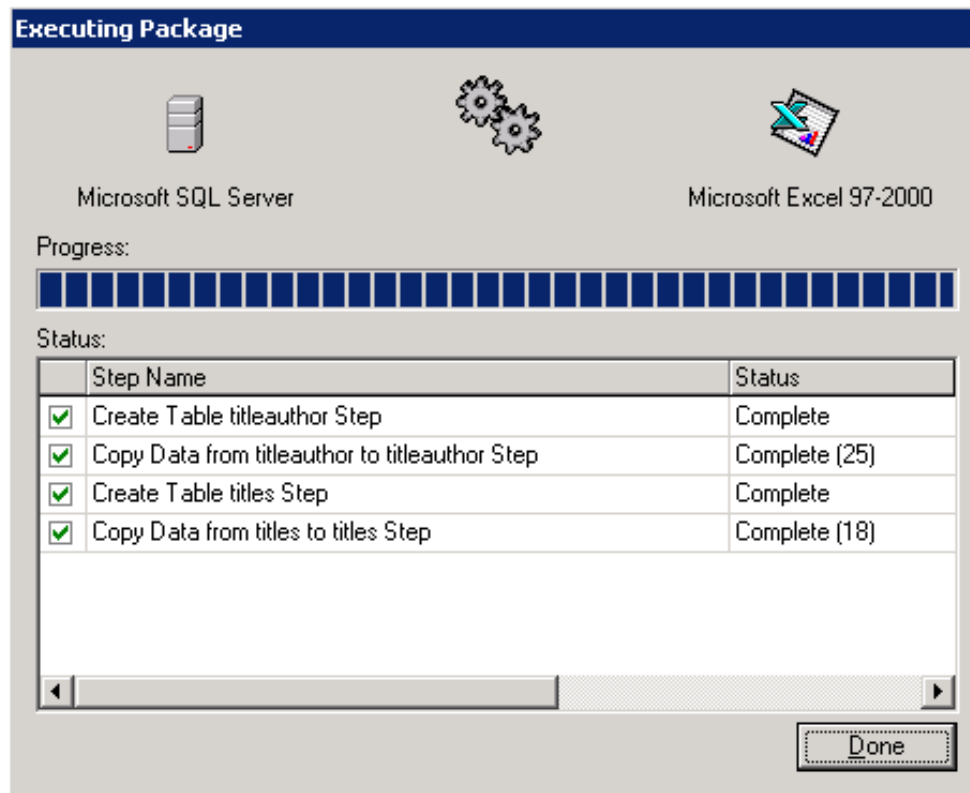
Dalam dialog Column Maps and Transformations yang ditunjukkan pada gambar di bawah ini, Anda dapat mengubah properti tabel yang akan dibuat di posisi target. Pada tab Transformations, Anda bisa menentukan skrip di dalam bahasa VBScript, JavaScript, atau PerlScript untuk mentransformasi atau memproses kolom-kolom pada tabel asli.



Kini mari kita kembali ke proses asli. Setelah menekan Next pada kotak dialog di mana Anda memilih tabel, sebuah layar lain akan muncul dan menanyakan apakah Anda ingin menjalankan operasi ekspor sekarang. Pilihlah Run, lalu tekan Next.



Layar terakhir wizard menampilkan rangkuman mengenai apa yang akan dilakukan. Tekanlah tombol Finish untuk mengakhiri proses. Hasilnya ditampilkan pada gambar di bawah ini.



Perhatikan pada gambar berikut ini bahwa di dalam workbook dari spreadsheet Excel yang ditentukan sebagai lokasi tujuan, ada tiga buah halaman yang masing-masing mengandung satu tabel.

1	A	B	C	D	E	F	G	H	
	title_id	title	type	pub_id	price	advance	royalty	ytd_sales	notes
2	BU1032	The Busy Executive's Database Guide	business	1389	\$19.99	\$5,000.00	10	4095	An overview of available database systems with emphasis on common business appli
3	BU1111	Cooking with Computers: Surreptitious Balance Sheets	business	1389	\$11.95	\$5,000.00	10	3876	Helpful hints on how to use your electronic resources to the best advantage.
4	BU2075	You Can Combat Computer Stress!	business	0736	\$2.99	\$10,125.00	24	16722	The latest medical and psychological techniques for living with the electronic office. E
5	BU7832	Straight Talk About Computers	business	1389	\$19.99	\$5,000.00	10	4095	Annotated analysis of what computers can do for you: a no-hype guide for the critical
6	MC2222	Silicon Valley Gastronomic Treats	mod_cook	0877	\$19.99	\$0.00	12	2032	Favorite recipes for quick, easy, and elegant meals.
7	MC3021	The Gourmet Microwave	mod_cook	0877	\$2.99	\$15,000.00	24	22246	Traditional French gourmet recipes adapted for modern microwave cooking.
8	MC3026	The Psychology of Computer Cooking	UNDECIDED	0877					
9	PC1035	But Is It User Friendly?	popular_comp	1389	\$22.95	\$7,000.00	16	8780	A survey of software for the naive user, focusing on the 'friendliness' of each.
10	PC8888	Secrets of Silicon Valley	popular_comp	1389	\$20.00	\$8,000.00	10	4095	Muckraking reporting on the world's largest computer hardware and software manufact
11	PC9999	Net Etiquette	popular_comp	1389					A must-read for computer conferencing.
12	PS1372	Computer Phobic AND Non-Phobic Individuals: Behavior Variations	psychology	0877	\$21.59	\$7,000.00	10	375	A must for the specialist, this book examines the difference between those who hate c
13	PS2091	Is Anger the Enemy?	psychology	0736	\$10.95	\$2,275.00	12	2045	Carefully researched study of the effects of strong emotions on the body. Metabolic ct
14	PS2106	Life Without Fear	psychology	0736	\$7.00	\$6,000.00	10	111	New exercise, meditation, and nutritional techniques that can reduce the shock of dai
15	PS3333	Prolonged Data Deprivation: Four Case Studies	psychology	0736	\$19.99	\$2,000.00	10	4072	What happens when the data runs dry? Searching evaluations of information-shortage
16	PS7777	Emotional Security: A New Algorithm	psychology	0736	\$7.99	\$4,000.00	10	3336	Protecting yourself and your loved ones from undue emotional stress in the modern w
17	TC3218	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean	trad_cook	0877	\$20.95	\$7,000.00	10	375	Profusely illustrated in color, this makes a wonderful gift book for a cuisine-oriented fri
18	TC4203	Fifty Years in Buckingham Palace Kitchens	trad_cook	0877	\$11.95	\$4,000.00	14	15096	More anecdotes from the Queen's favorite cook describing life among English royalty.
19	TC7777	Sushi, Anyone?	trad_cook	0877	\$14.99	\$8,000.00	10	4095	Detailed instructions on how to make authentic Japanese sushi in your spare time.
20									
21									

Kesimpulan

Bab ini mendemonstrasikan cara mengekspor data dengan menggunakan Export Wizard. Bab berikutnya akan membahas beberapa sarana grafis yang disertakan di dalam SQL Server.

BAB 17

Program Bantu SQL Server 7

Gambaran Singkat.

SQL Server disertai dengan beberapa sarana grafis yang dapat menyederhanakan dan mempercepat eksekusi tugas-tugas administratif. Program-program ini di antaranya adalah:

- Microsoft Management Console
- SQL Server Client Network Utility
- SQL Server Enterprise Manager
- SQL Server Network Utility
- Performance Monitor
- SQL Server Profiler
- SQL Server Query Analyzer
- SQL Server Service Manager
- SQL Server Setup
- Version Upgrade Wizard

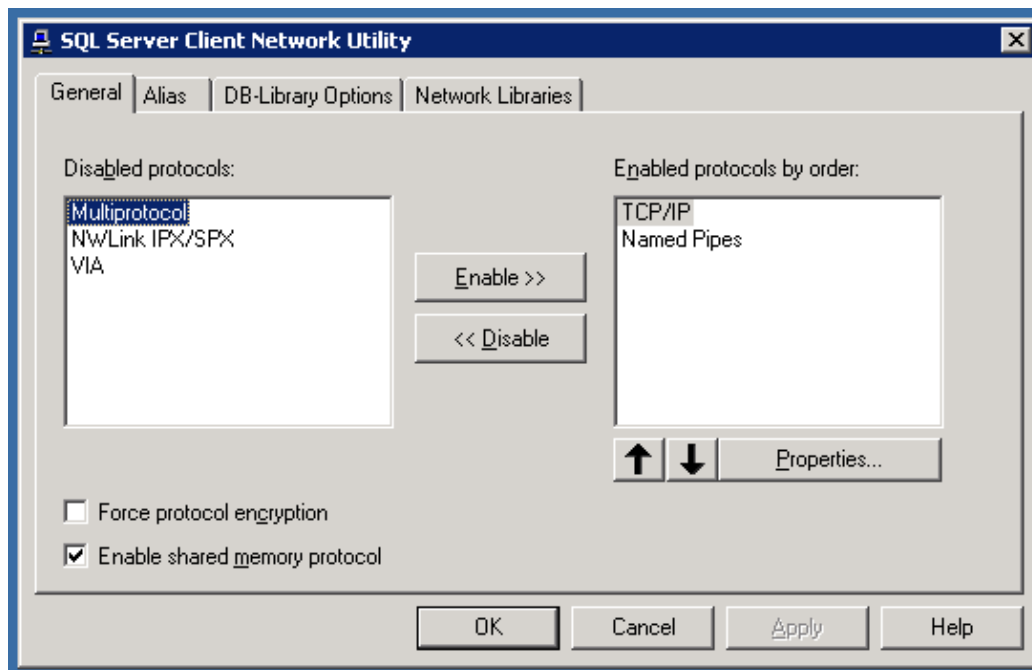
Di bawah ini adalah keterangan singkat tentang masing-masing program tersebut. Beberapa di antaranya telah dibahas pada bab-bab sebelumnya, tetapi yang lainnya akan dibahas dengan rinci pada bab ini.

Microsoft Management Console

Konsol Manajemen Microsoft (MMC) adalah antarmuka pengguna dasar untuk mengelola server BackOffice. Itu ada di belakang beberapa program seperti SQL Server Enterprise Manager atau OLAP Manager.

SQL Server Client Network Utility

Utilitas Jaringan Klien SQL Server adalah program untuk mengelola koneksi jaringan yang ditentukan klien serta pustaka DB dan pustaka jaringan.



Anda bisa mengkonfigurasi item-item berikut ini pada tab General:

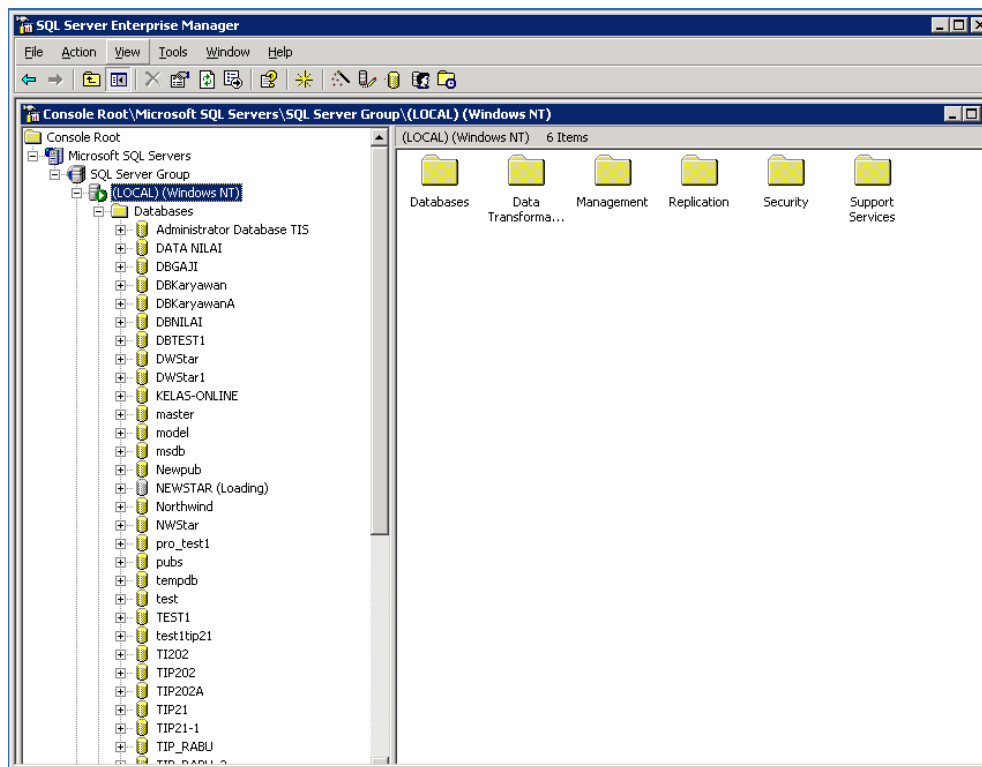
Pilihan	Fungsi
Default Network Library	Menentukan perpustakaan default untuk berkomunikasi dengan SOL Server
Server Alias Configurations	Menunjukkan alias dari komputer client
Network Library	Menunjukkan perpustakaan network yang digunakan
Connection Parameters	Menunjukkan parameter-parameter yang mungkin yang berhubungan dengan alamat koneksi
Add	Mengizinkan penambahan koneksi network baru
Remove	Mengizinkan penghapusan koneksi
Edit	Mengizinkan penyuntingan koneksi

SQL Server Enterprise Manager

Kita sudah pernah membahas mengenai program ini, dan ia merupakan sarana utama untuk mengelola server SQL Server. Di antara tugas-tugas yang dilakukan oleh Enterprise Manager adalah:

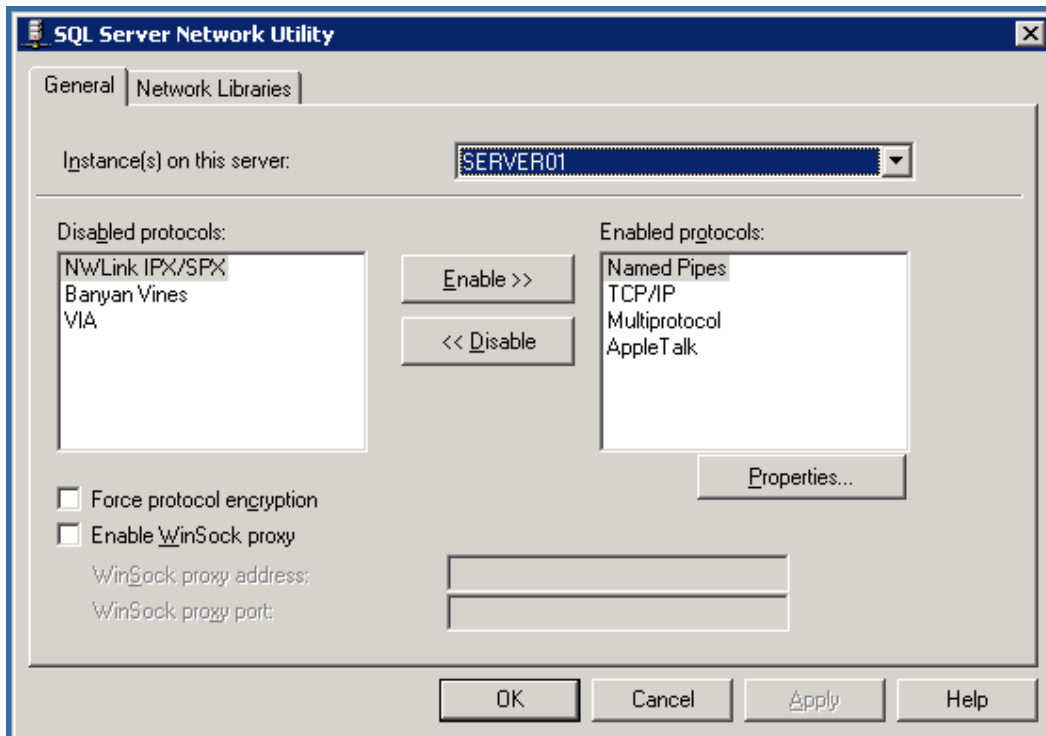
- Membuat, melihat, dan mengelola isi database melalui pembuatan tabel, prosedur, indeks, aturan, dan diagram.
- Mengimpor dan mengekspor data. Mentransformasi data.
- Menjalankan tugas-tugas administratif.

Inilah layar dari Enterprise Manager :



SQL Server Network Utility

Program ini berfungsi mengkonfigurasi koneksi dengan SQL Server apabila protokol komunikasi antara server dan client tidak bekerja dengan baik.



Performance Monitor

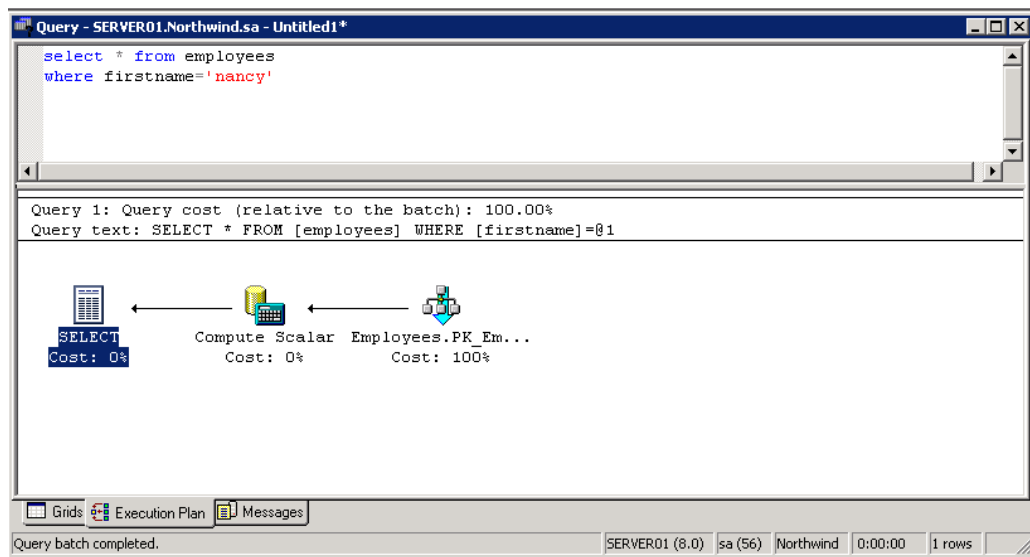
Performance Monitor menawarkan analisa yang cepat tentang aktivitas-aktivitas yang berlangsung di dalam database, dan merupakan program yang bagus untuk mendiagnosa masalah yang mungkin dimiliki oleh sistem.

SQL Server Profiler

SQL Server Profiler terus mencatat aktivitas server. Data dapat dikirim secara grafis di layar atau ke file atau spreadsheet untuk analisis nanti.

SQL Server Query Analyzer

Query Analyzer mengizinkan pengekseskuan perintah-perintah dan skrip SQL dan mengizinkan user melihat langkah-langkah eksekusi dari suatu query secara grafis. Program ini juga menganalisa indeks dan mengusulkan perubahan untuk meningkatkan kinerja. Kita sudah pernah membahasnya secara rinci pada bab sebelumnya.



SQL Server Service Manager

SQL Server Service Manager adalah sebuah program kecil yang digunakan untuk menjalankan atau mengakhiri SQL Server (MSSQLServer), SQL Server Agent, dan Microsoft Distributed Transaction Coordinator (MSDTC). SQL Server Service Manager tampak sebagai icon pada taskbar.

SQL Server Setup

SQL Server Setup digunakan untuk menginstal dan mengkonfigurasi server. Setelah diinstal, ia bisa digunakan untuk beberapa tugas termasuk menyempurnakan pilihan keamanan, merekonstruksi database utama, serta menukar pilihan network.

Version Upgrade Wizard

Upgrade Wizard membantu proses migrasi database dari SQL Server ke versi yang lebih tinggi. Sisa dari bab ini akan membahas secara rinci beberapa program di atas yang belum diulas pada buku ini.

SQL Server Profiler

SQL Server Profiler membantu Anda memantau kejadian-kejadian yang terjadi pada server, seperti misalnya percobaan login, koneksi dan pemutusan hubungan dari server, pengeksekusian skrip batch dari bahasa Transact-SQL, serta deadlock.

Dengan Server Profile, Anda bisa membuat trace (penelusuran kejadian) untuk mengumpulkan informasi mengenai server dan kejadian yang dialaminya. Informasi ini bisa

disimpan di dalam file khusus atau tabel database, atau bisa langsung dilihat pada Server Profile.

Mendaftarkan Server

Server harus didaftarkan terlebih dahulu agar bisa dipantau oleh Server Profiler. Untuk melakukan tugas ini, aktifkanlah pilihan Register SQL Server pada menu Tools, seperti ditunjukkan di bawah ini. Lalu pada kotak dialog Registered SQL Server Properties yang muncul, pilihlah nama server dan tekan OK.

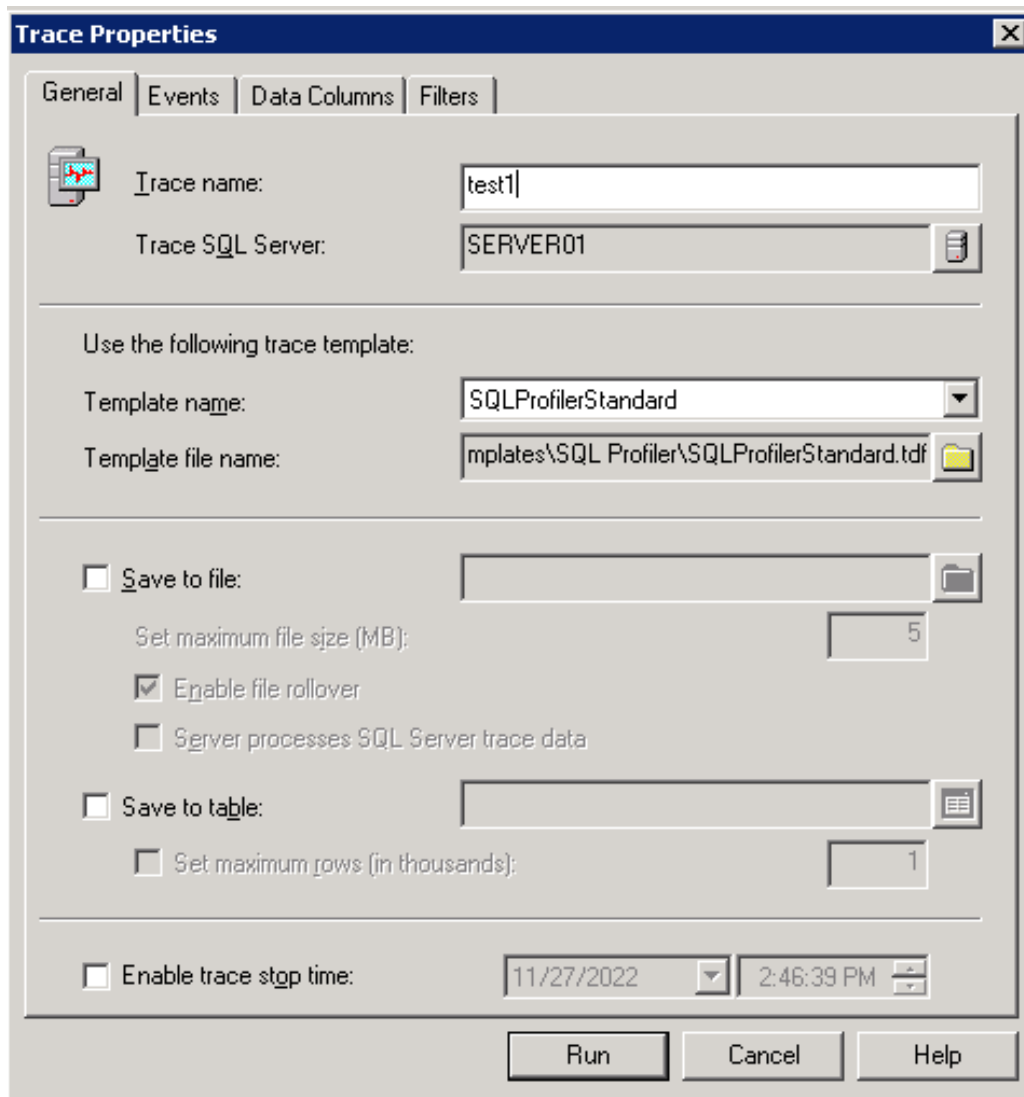
EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID
TraceStart										
ExistingConnection	-- network protocol: LPC set quoted...	SQLAgent - ...	SYSTEM	NT AU...					2164	51
ExistingConnection	-- network protocol: LPC set quoted...	SQLAgent - ...	SYSTEM	NT AU...					2164	52
ExistingConnection	-- network protocol: LPC set quoted...	MS SQLEM		sa					2984	53
ExistingConnection	-- network protocol: LPC set quoted...	SQL Query A...		sa					212	54
ExistingConnection	-- network protocol: LPC set quoted...	MS SQLEM		sa					3692	55
ExistingConnection	-- network protocol: LPC set quoted...	SQL Query A...		sa					288	56
ExistingConnection	-- network protocol: LPC set quoted...	SQL Query A...		sa					308	57
SQL:BatchCompleted	SELECT N'Testing Connection...'	SQLAgent - ...	SYSTEM	NT AU...	0	0	0	0	2164	52
SQL:BatchCompleted	EXECUTE msdb.dbo.sp_sqlagent_get_pe...	SQLAgent - ...	SYSTEM	NT AU...	16	96	0	13	2164	52
SQL:BatchCompleted	SELECT N'Testing Connection...'	SQLAgent - ...	SYSTEM	NT AU...	0	0	0	0	2164	52
SQL:BatchCompleted	EXECUTE msdb.dbo.sp_sqlagent_get_pe...	SQLAgent - ...	SYSTEM	NT AU...	0	96	0	0	2164	52

SELECT N'Testing Connection...'

Ready Ln 1, Col 32 Rows: 1 Connections: 1

Membuat Trace

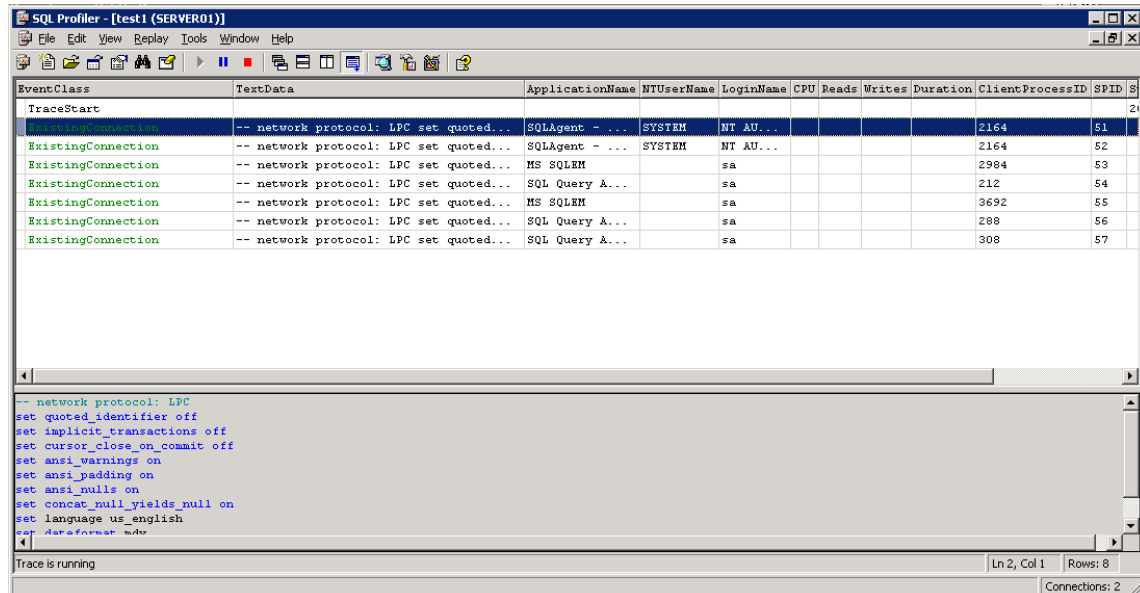
Suatu trace berisikan data yang ditangkap oleh kejadian-kejadian yang dipantau. Untuk membuatnya, aktifkanlah pilihan New Trace dari menu File. Pada kotak dialog Trace Properties, ketikkan Test1 pada field Trace Name.



Tabel berikut ini menunjukkan beberapa pengaturan yang bisa Anda buat pada tab General:

Pilihan	Fungsi
Trace Name	Menentukan nama dari trace
Trace Type Shared	Memberikan akses kepada semua user
Trace Type Private	Memberikan akses hanya kepada user yang membuat trace
SOL Server	Menentukan server yang dipantau
Capture To File	Menangkap hasil trace ke dalam file pada sistem operasi'
Capture To Table	Menangkap hasil trace ke dalam tabel di dalam database

Tekan OK untuk memulai pelacakan. Hasilnya akan ditampilkan seperti gambar di bawah ini.



Filter

Filter adalah kriteria yang menentukan peristiwa yang akan dilacak selama proses pelacakan. Untuk menentukan filter, pilih File | Buka dan klik Lacak Definisi. Di kotak dialog yang muncul, klik tab Filter, lalu pilih kondisi dari daftar Lacak Kriteria Peristiwa.

Menjalankan Trace

Untuk menjalankan pelacakan, pilih File | menjalankan trek Pada dialog yang muncul, karena Anda dapat menjalankan beberapa track sekaligus, klik Selected Tracks dengan satu atau beberapa track tersisa.

Menyimpan Trace

Trace bisa disimpan dengan menggunakan File | Open | Trace Definition. Tentukanlah nama untuk trace ini pada kotak dialog Trace Properties dan pilih Capture to Table. Lalu ketikkan nama untuk database, pemilik, dan tabel.

SQL Server Service Manager

Service Manager harus muncul secara default pada taskbar. Untuk menjalankan SQL

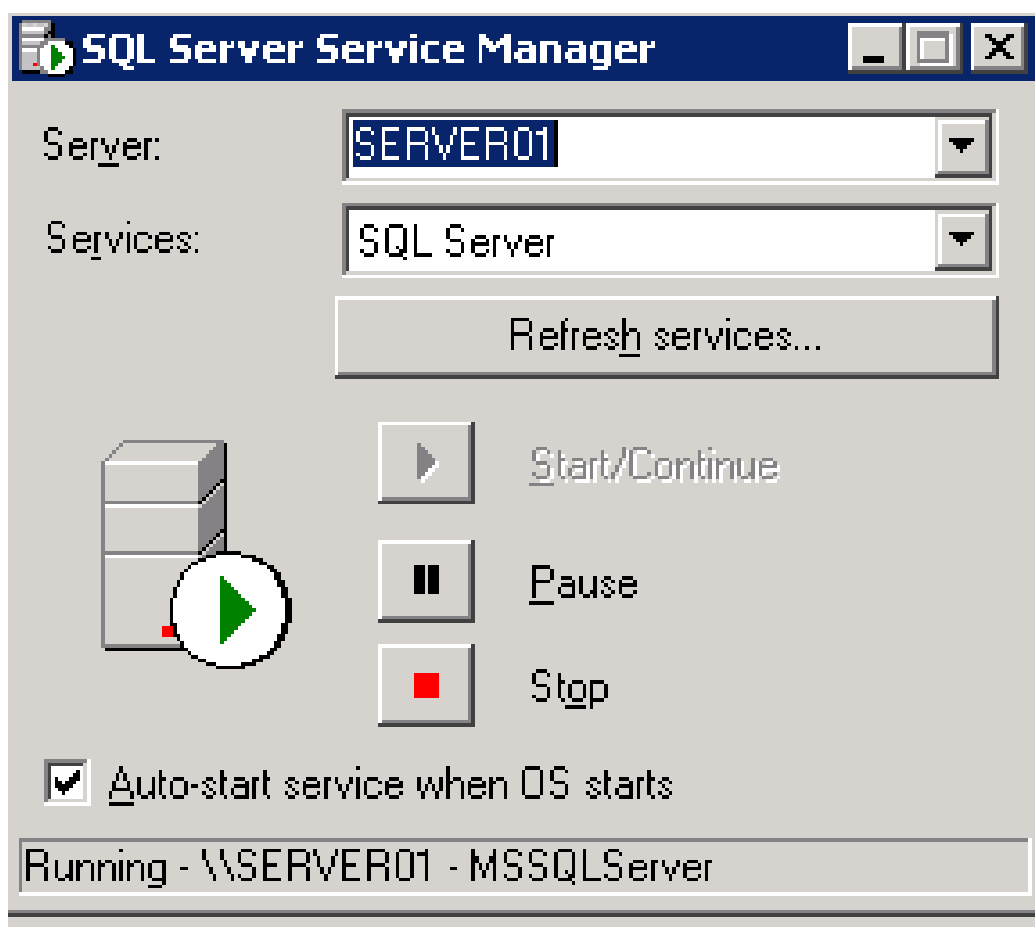
Server, klik ganda icon Service Manager dan jalankan prosedur berikut ini:

Pada field Server, pilihlah nama server yang ingin diinisialisasi atau ketikkan nama lengkap dari remote server.

Pada field Services, pilihlah MSSQLServer.

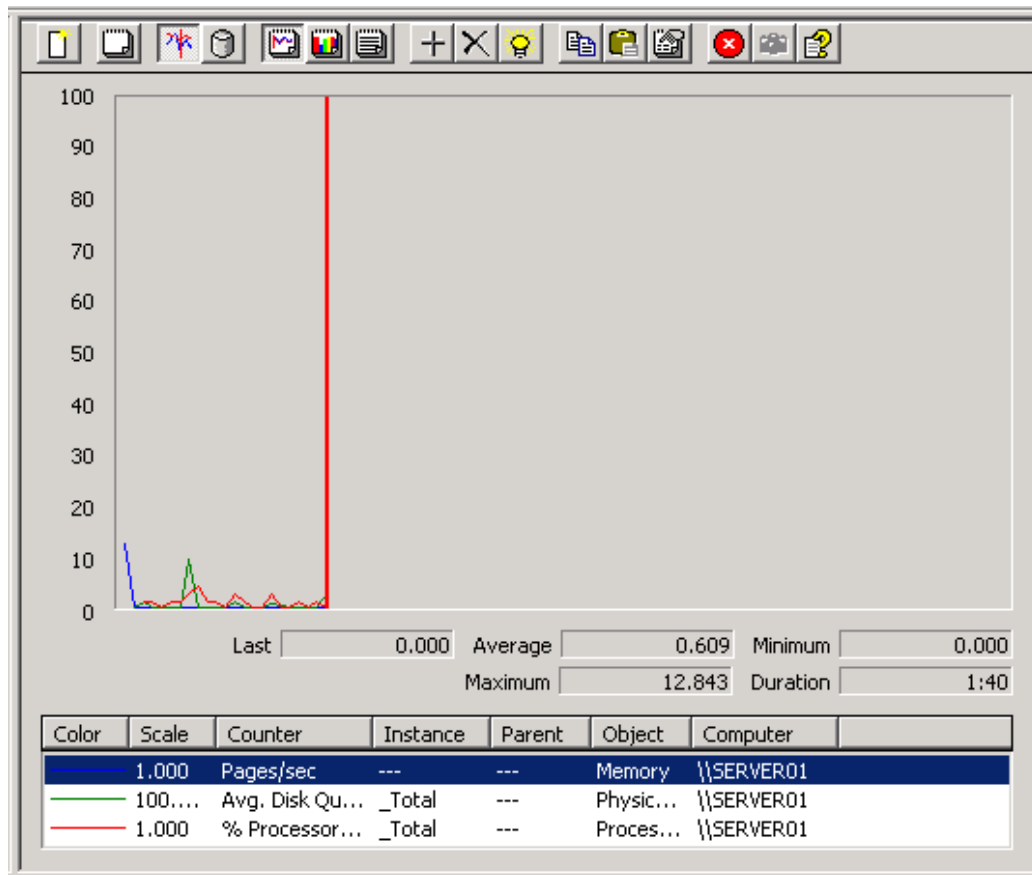
Klik ganda pada Start/Continue.

Dengan tombol Pause, Anda bisa menghentikan SQL Server secara sementara. Untuk mengaktifkan kembali eksekusinya, tekan tombol Start/Continue. Untuk menghentikan eksekusi server, tekanlah tombol Stop.



Performance Monitor

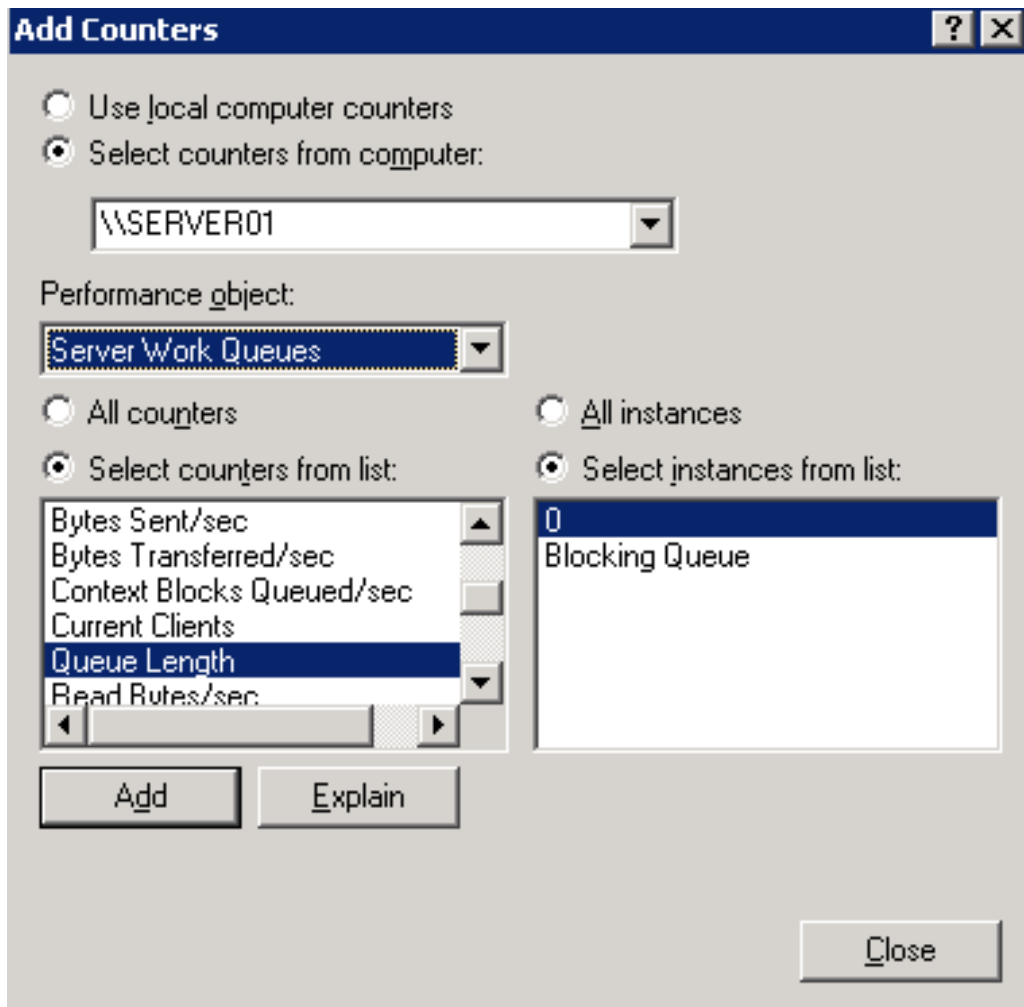
Performance Monitor mirip seperti mesin x-ray yang memantau aktivitas server lokal atau server remote. Dengan Performance Monitor, Anda bisa memeriksa status (atau perilaku) dari objek-objek seperti misalnya prosesor, memory, cache, thread, dan proses. Objek-objek ini bisa dihubungkan dengan counter- counter yang menampilkan data statistik mereka.



Performance Monitor terdiri dari empat buah jendela yang bernama Chart, Alert, Log, dan Report, yang bisa ditampilkan dengan pilihan-pilihan pada menu View.

Data statistik yang dicatat akan hilang jika tidak disimpan sebelum keluar dari program. Anda bisa menyimpan pengaturan yang dilakukan pada jendela dengan mengklik pilihan Save Settings pada menu File. Untuk menyimpan pengaturan yang dilakukan pada semua jendela, pilihlah Save Workspace pada menu File.

Untuk menyertakan analisa objek baru, gunakanlah pilihan Add to Chart pada menu Edit. Pada kotak dialog yang muncul, pilihlah salah satu pilihan pada field Object, seperti ditunjukkan pada gambar di bawah ini.



Kesimpulan

Bab ini memberikan gambaran yang luas mengenai beberapa program bantu SQL Server yang belum dibahas pada buku ini. Program-program ini dapat membantu Anda melakukan tugas-tugas administratif. Berikutnya kita akan membahas mengenai cara mem-backup dan merestorasi data.

BAB 18

Mem-backup dan Merestorasi Data

SQL Server memiliki dua alat untuk membuat cadangan basis data dan komponennya. Alat-alat ini sangat penting untuk menjaga agar database tetap berjalan jika terjadi kegagalan perangkat keras, penghapusan tabel yang tidak disengaja, atau bahkan kehilangan data server.

Membuat cadangan melibatkan menyalin database ke lokasi yang aman. Jika memungkinkan, pencadangan harus dilakukan di komputer lain di media jaringan, pita, atau media magnetik lainnya. Proses pencadangan menyalin semuanya dari database, termasuk catatan transaksi.

Catatan transaksi adalah catatan serial yang berisi semua perubahan yang dilakukan pada database. Log ini digunakan dalam proses pemulihan untuk mencerminkan semua perubahan yang dilakukan pada database sejak pencadangan terakhir.

Proses restorasi database apabila sukses akan mengembalikan database ke status terdahulu pada saat backup terakhir dijalankan. Jika selama proses backup ada transaksi yang belum lengkap, ia akan dibatalkan untuk menjamin konsistensi database.

Jenis-Jenis Backup

SQL Server memungkinkan dua jenis cadangan: sempurna dan berbeda. Cadangan lengkap adalah "snapshot" yang tepat dari database. Cadangan diferensial adalah salinan dari perubahan yang dilakukan pada database sejak pencadangan penuh terakhir. Dengan menggunakan kedua jenis pencadangan ini, Anda dapat membuat mekanisme pencadangan yang aman untuk mengoptimalkan disk dan ruang disk. Misalnya, Anda dapat melakukan pencadangan penuh di pagi hari dan pencadangan diferensial di siang hari dan di penghujung hari.

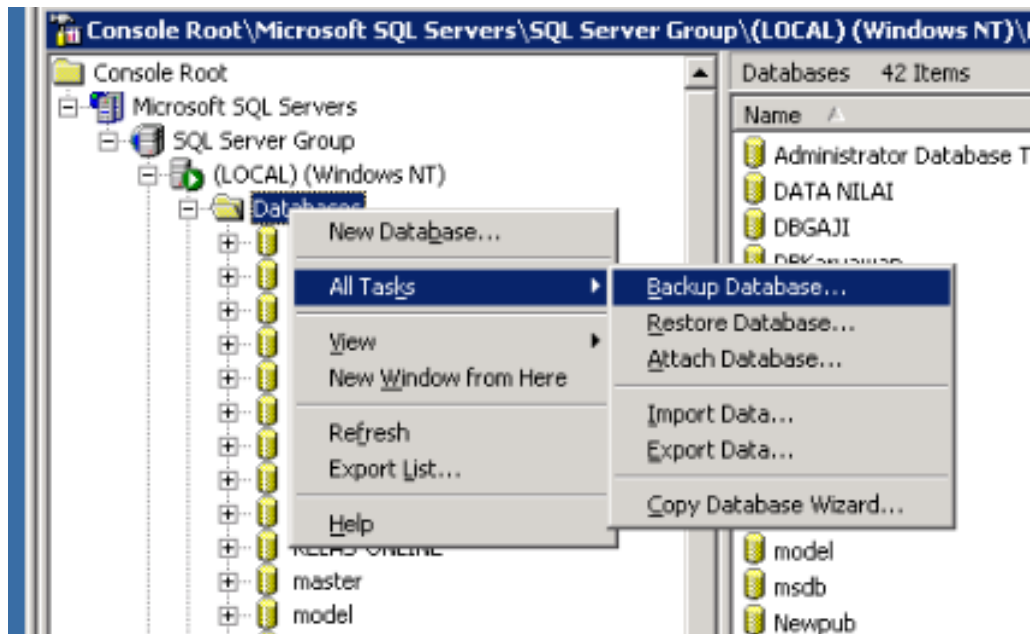
SQL Server menyediakan kemampuan membuat dan memulihkan backup melalui penggunaan perintah-perintah Transact-SQL, seperti `BACKUP DATABASE` atau `RESTORE DATABASE`, atau melalui Enterprise Manager atau Backup Wizard. Pada bab ini kita akan membahas penggunaan Enterprise Manager dan Backup Wizard.

Membuat Backup dengan Enterprise Manager

Untuk mencadangkan dengan Enterprise Manager, klik kanan folder database untuk

mengaktifkan menu konteks.

Pilih Semua Tugas | Pencadangan Basis Data.

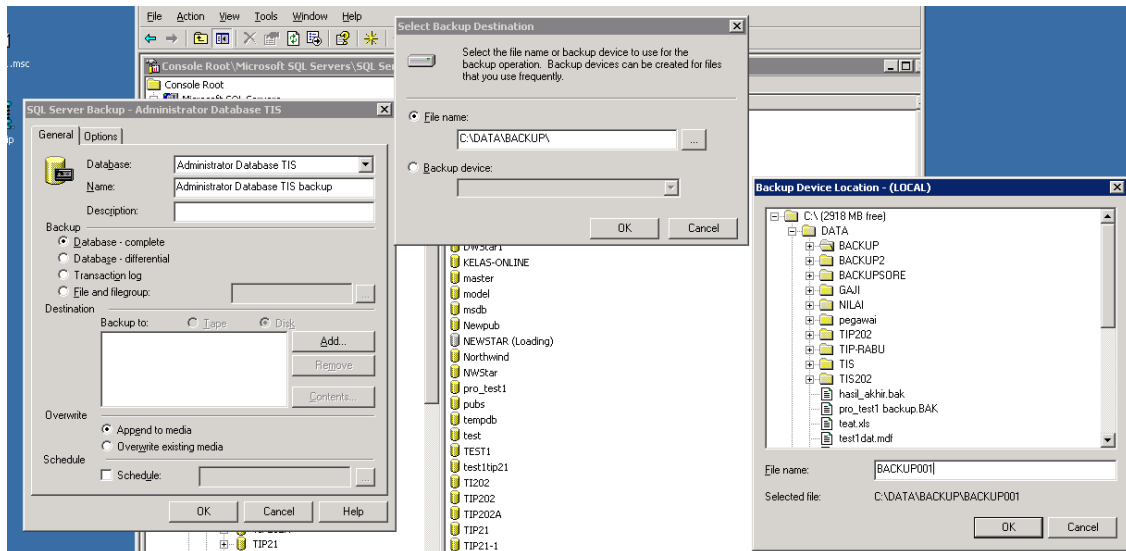


Untuk awalnya, kita akan membuat backup untuk database Newpub2. Langkah pertama adalah memilih database. Pada field Database, pilihlah nama database yang akan di-copy. Database Master akan dipilih secara default. Pilihlah Newpub2 lalu tekan tombol OK. (Jika Anda belum membuat database ini, ikuti petunjuk pada buku ini).

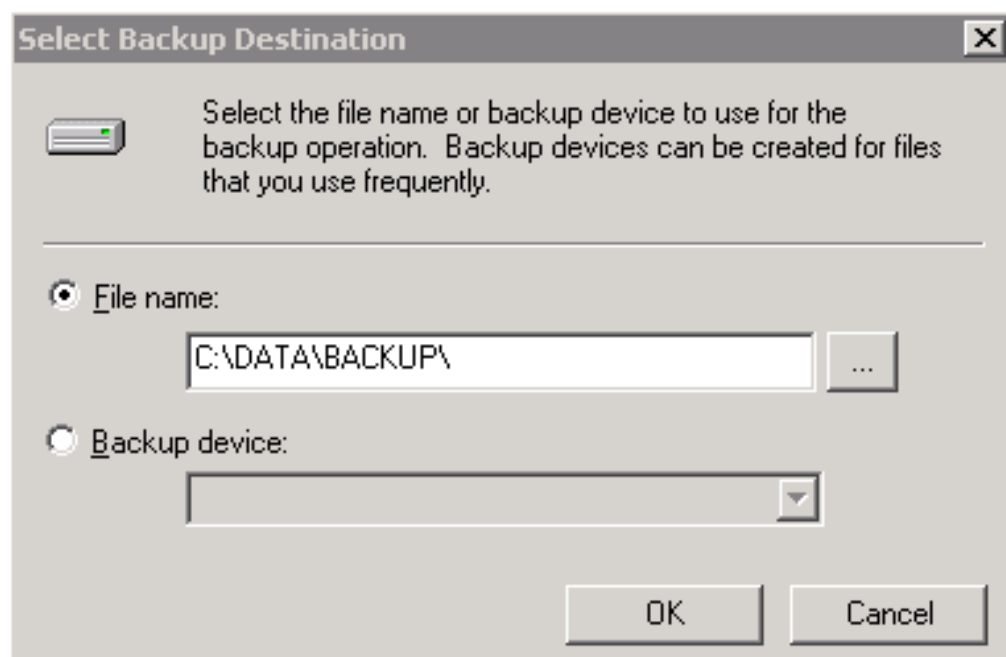
Pada langkah kedua, pilihlah lokasi tujuan untuk backup. Untuk melakukan ini, tekan tombol Add pada panel Destination untuk menampilkan kotak dialog Choose Backup Destination. Defaultnya adalah direktori MSSQL7 \BACKUP. Idealnya, backup harus berada pada komputer network yang berbeda atau drive yang berbeda pada komputer Anda. Untuk memilih lokasi lain, klik tombol Browse (tombol dengan tiga tanda titik), dan pilih folder lain. Pada latihan ini, kita akan menerima direktori default sebagai lokasi tujuan .

Jika komputer Anda memiliki unit tape, alat tersebut akan muncul sebagai pilihan pada kotak dialog lokasi tujuan.

Langkah-langkah ini ditunjukkan pada gambar di bawah ini.



Selain memilih lokasi tujuan, Anda juga harus memberi nama file backup tersebut. Di sini, kita akan menggunakan nama newpub backOl. Lalu tekan tombol OK.

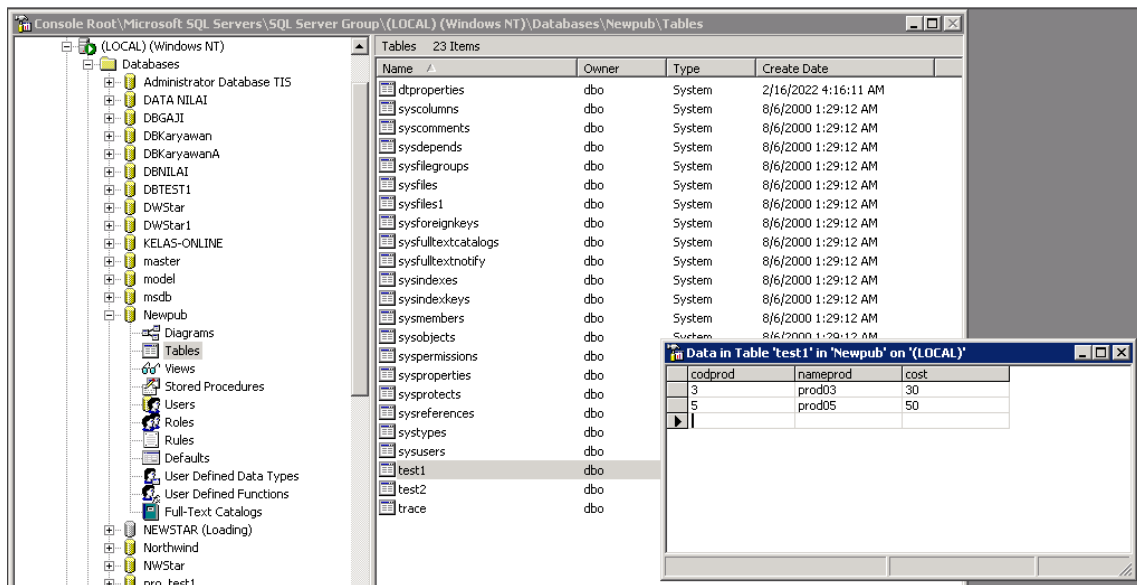


Proses backup akan dijalankan. Proses ini bisa memakan waktu dari beberapa detik hingga beberapa jam, bergantung dari ukuran database Anda.

Mengubah Database

Kini kita akan membuat beberapa perubahan pada database Newpub2. Pertama -tama mari kita buat sebuah tabel dummy yang bernama Test3, yang hanya

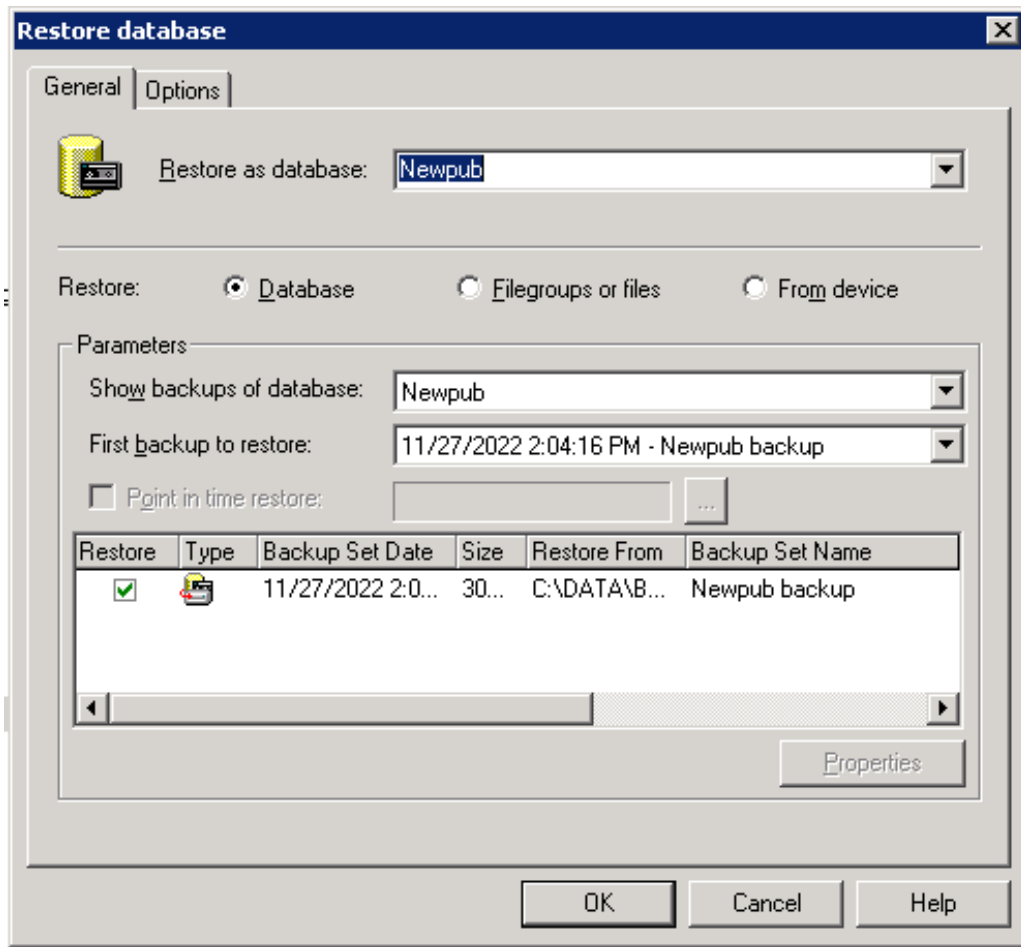
mengandung satu buah kolom bernama codprod. Lalu kita akan menghapus record di mana codprod memiliki nilai 6 pada tabel Test2.



Kini database telah berubah. Karena kita sudah membuat salinan yang lengkap dari database tersebut, kita tidak perlu melakukan backup lengkap.

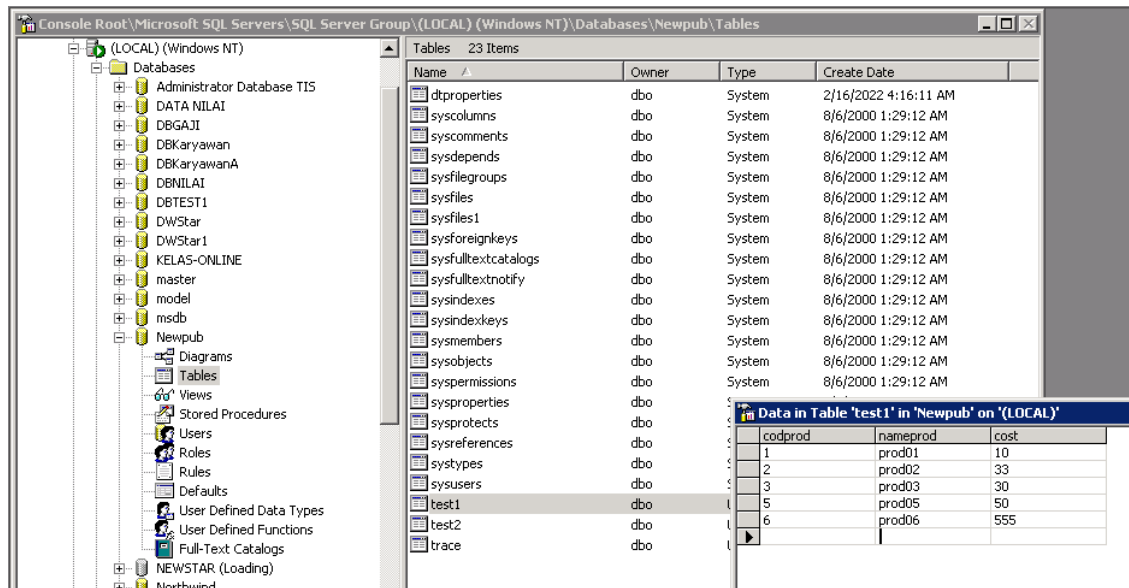
Merestorasi Backup

Untuk memulihkan cadangan penuh, klik kanan folder Database dan pilih Semua Tugas | Pemulihan basis data. Akan muncul kotak dialog seperti pada gambar di bawah ini.



Pada field Restore as Database, pilihlah newpub2. Pada panel Parameters, pilihlah database Newpub2 pada field Show Backups of Database. Tekan tombol OK.

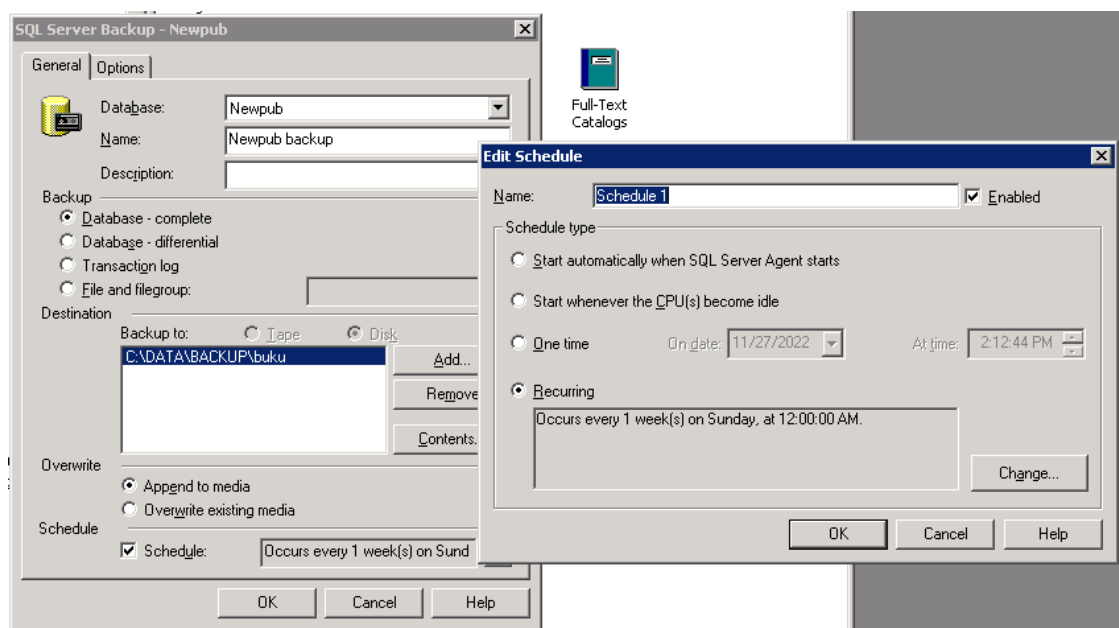
Sebuah kotak dialog akan menampilkan kemajuan dari proses restorasi. Kini kita harus menghadapi ujian terakhir. Kita akan membuka folder Tables dan memeriksa isi dari tabel Test2. Gambar di bawah ini menampilkan hasilnya.



Perhatikan bahwa kini sudah tidak ada lagi tabel Test3, yang dibuat setelah menjalankan backup pertama, dan baris dengan codprod bernilai 6 suda kembali pada tabel Test2.

Menjadwalkan Backup

Backup bisa dilakukan secara langsung atau dijadwalkan agar tidak mempengaruhi pekerjaan. Untuk menjadwalkan backup pilihlah item Schedule pada kotak dialog SQL Server Backup.



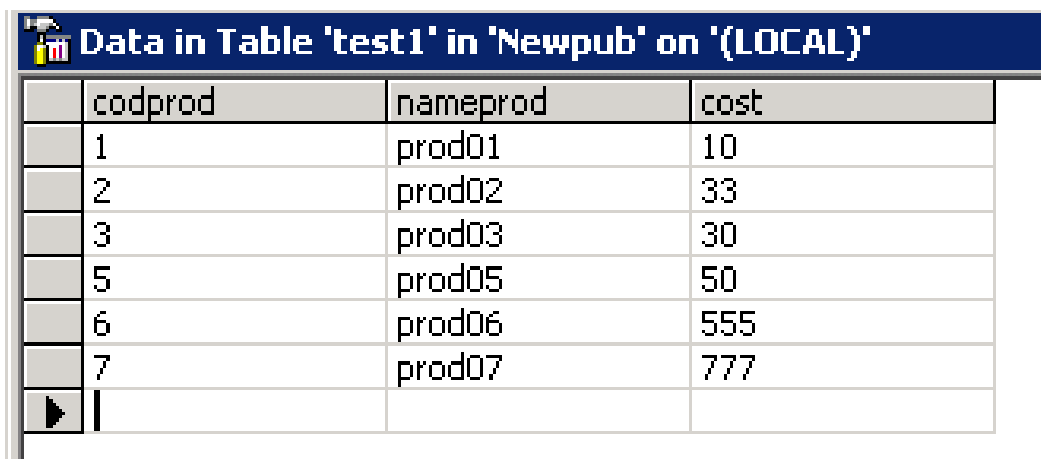
Tekanlah tombol Browse (tombol dengan tiga titik) untuk menampilkan kotak dialog

Edit Schedule, di mana Anda bisa memilih kapan backup akan dijalankan. Anda bisa menjadwalkan sebuah proses backup pada waktu yang ditentukan dengan memilih One Time dan menyetikkan tanggal serta jamnya. Atau Anda bisa menjadwalkan backup periodik dengan pilihan Recurring. Pilihan ini menampilkan frekuensi yang bisa Anda ubah dengan tombol Change. Anda juga bisa menentukan agar proses backup dijalankan saat SQL Server Agent dijalankan atau selama waktu idle CPU.

Membuat Backup Diferensial

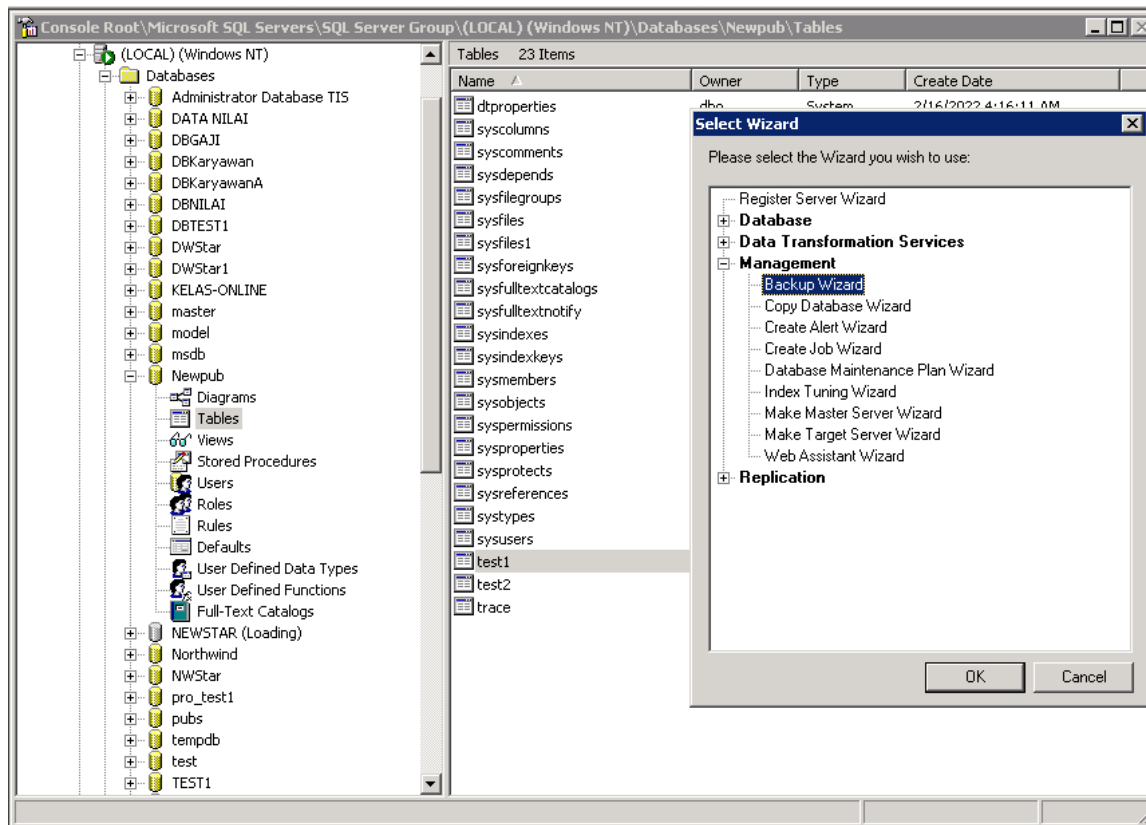
Backup diferensial adalah proses backup di mana kita hanya mem-backup data yang berubah semenjak proses backup terakhir. Cara ini dapat mengurangi informasi yang disimpan di dalam file-file backup, serta menghemat waktu dan ruang. Sebagai contoh, Anda bisa melakukan backup lengkap pada hari Senin dan backup diferensial pada hari-hari lainnya. Jika ada masalah di malam hari, Anda tinggal mengambil backup yang dilakukan pada hari sebelumnya. Kita akan membuat dua buah backup diferensial.

Sebelum membuat backup yang pertama, mari kita masukkan baris lain ke dalam tabel Test1. Ketikkan 7, prod7, 777.

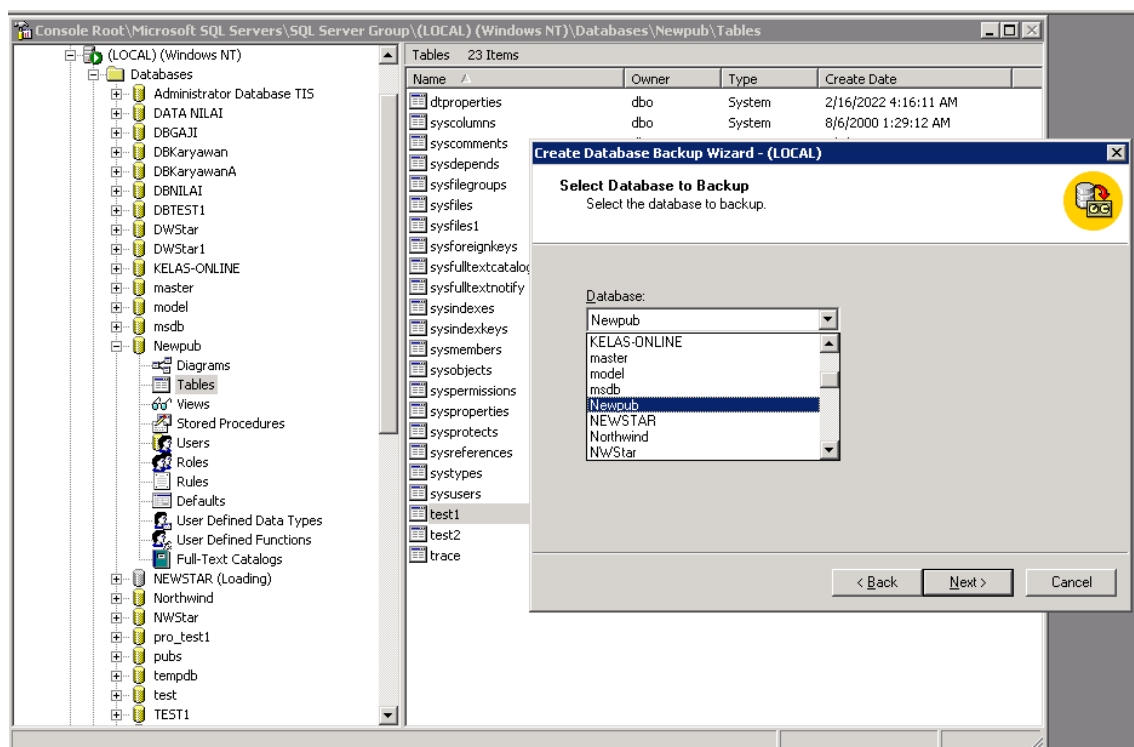


	codprod	nameprod	cost
	1	prod01	10
	2	prod02	33
	3	prod03	30
	5	prod05	50
	6	prod06	555
	7	prod07	777

Apabila Anda menyisipkan baris ini pada tabel Test 1, tabel Test2 secara otomatis akan di-update, karena ada trigger yang telah kita buat pada Bab 10. Kali ini kita akan menggunakan Create Database Backup Wizard. Klik pada menu Tools lalu pilih Wizards. Pilih Management | Backup Wizard, lalu tekan OK.

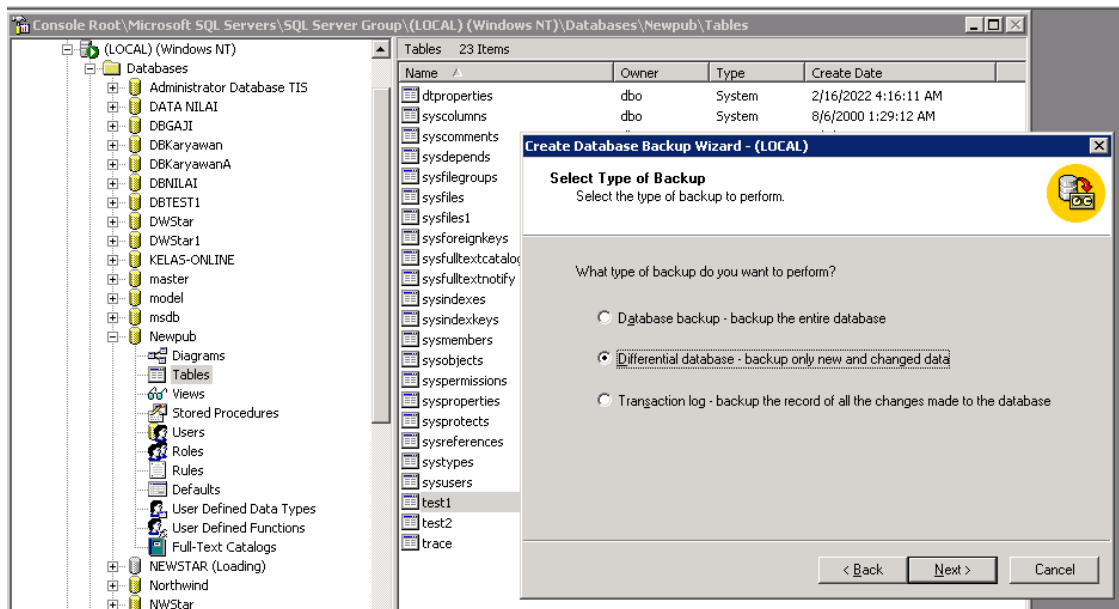


Layar pertama dari wizard menampilkan data yang bersifat informasi. Tekan Next pada layar ini. Pada langkah berikutnya, pilihlah database yang akan di copy dari daftar yang ditampilkan pada field Databases Name.

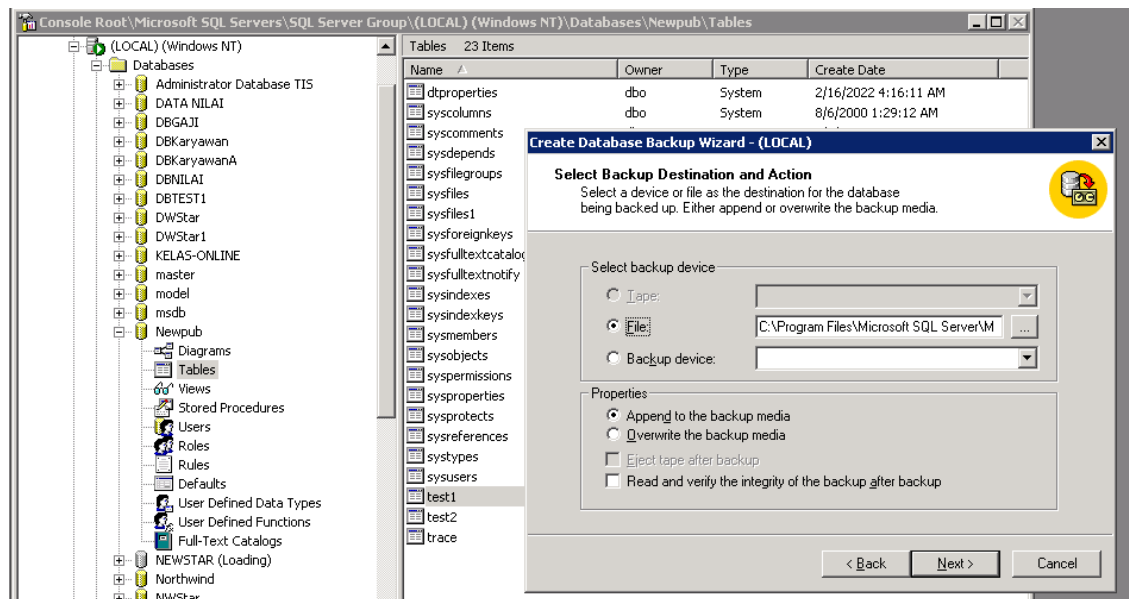


Pada latihan ini, kita akan memilih Newpub2, lalu tekan Next. Langkah berikutnya adalah memastikan atau mengetikkan nama untuk backup serta keterangan tambahan. Ketikkan backup newpub2 iOl sebagai nama backup dan differential backup sebagai keterangannya. Lalu tekan Next. Langkah berikutnya menentukan jenis backup yang akan dilakukan.

Pilihan pertama yaitu Database Backup berfungsi meng-copy seluruh database. Pilihan kedua yaitu Differential Database hanya meng-copy data yang baru atau telah berubah. Pilihan terakhir hanya meng-copy catatan transaksi. Pilihlah Differential Database lalu tekan tombol Next.



Pada langkah ini, pilihlah lokasi tujuan untuk backup. Di sini kita akan menambahkan backup ke dalam file yang telah dibuat dengan pilihan Append.



Setelah menekan tombol Next, kotak dialog yang terakhir akan muncul, yang menampilkan rangkuman dari operasi yang akan dilakukan. Tekan tombol Finish pada kotak dialog ini.

Untuk menguji backup diferensial, mari kita masukkan sebuah record ke dalam tabel Test1 dengan isi sebagai berikut:

```
Codprod=8
Nameprod=prod8
Cost=888
```

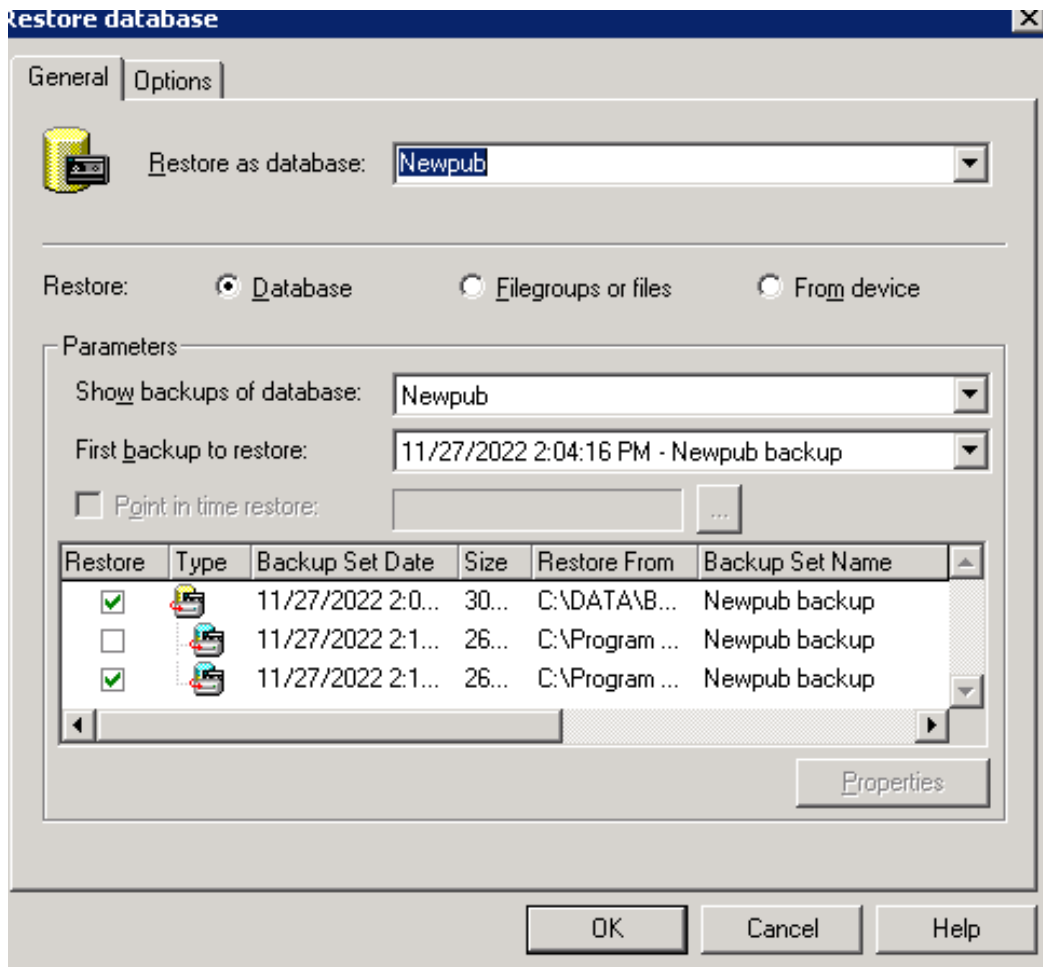
Kemudian kita hapus record dengan codprod 5.

Lalu kita akan membuat backup diferensial lain dengan mengikuti langkah-langkah yang telah dijelaskan di atas dengan menggunakan Backup Wizard dan ketikkanlah namanya newpub2 differential 2.

Merestorasi Backup

Anda bisa menggunakan proses yang sama untuk mengembalikan backup lengkap dan backup diferensial. Pilihlah All Tasks | Restore Data base.

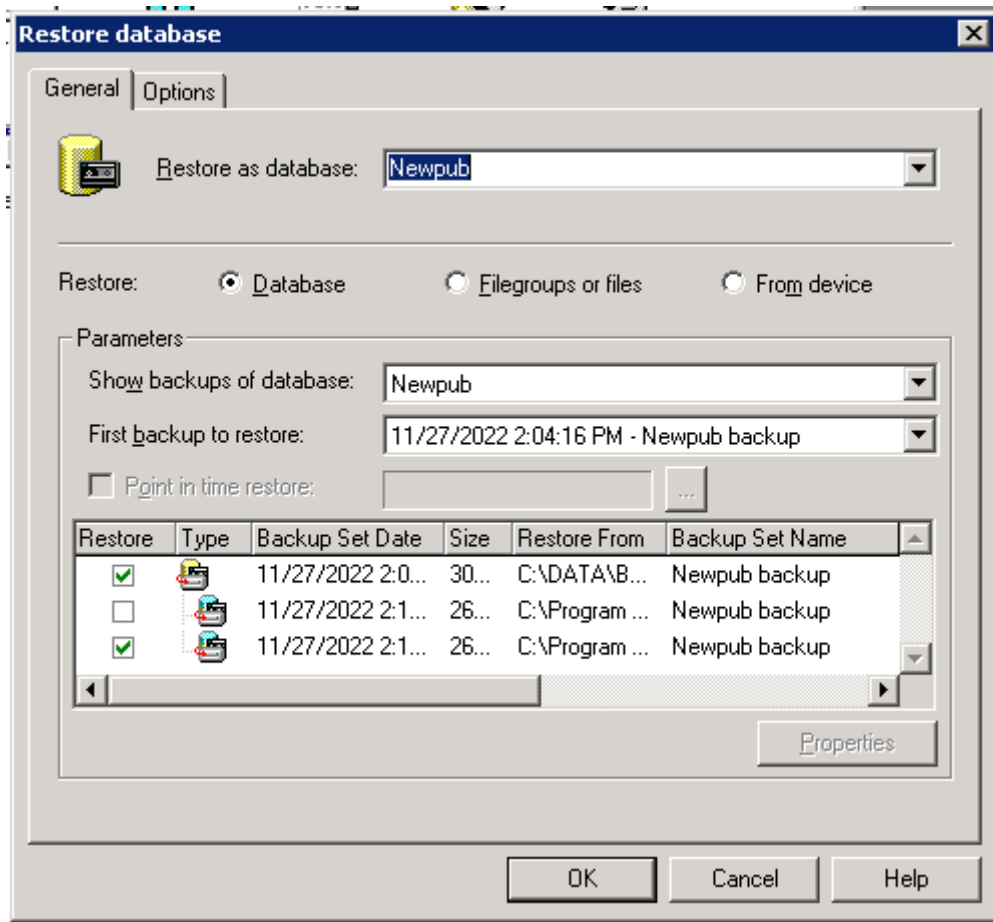
Pada kotak dialog Restore Database, pilihlah database yang akan direstorasi. Di sini kita akan memilih Newpub2.



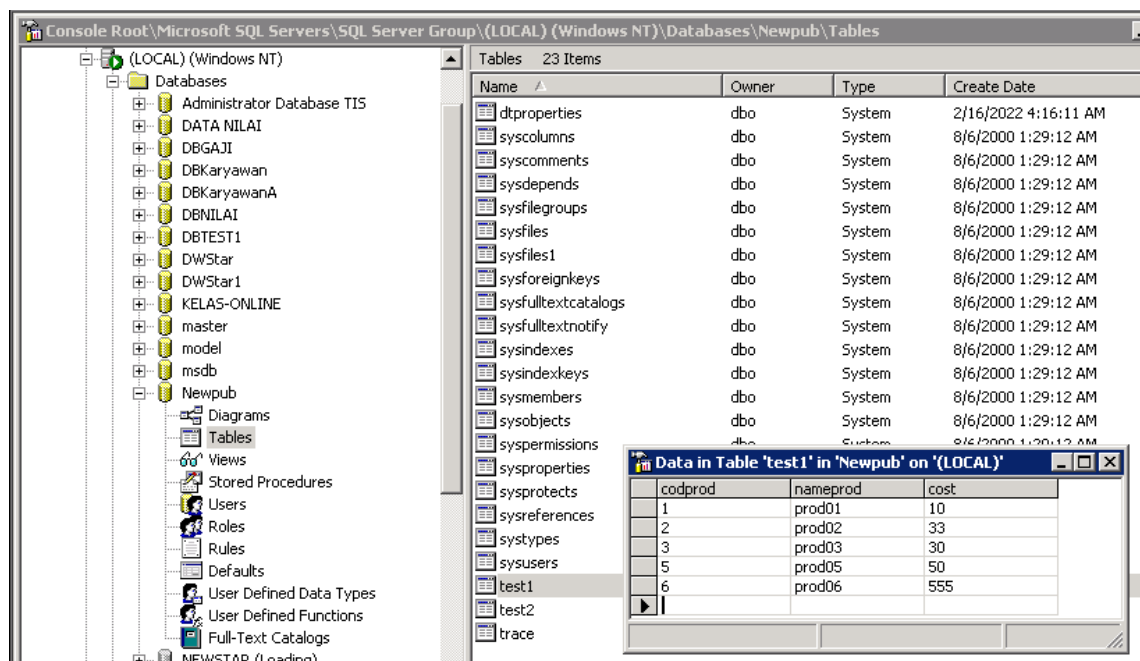
Perhatikanlah pada kolom Backup Set Name bahwa ada tiga proses backup yang telah dilakukan. Yang pertama adalah backup lengkap di awal. Dan dua backup lainnya adalah backup diferensial yang dilakukan kemudian.

Merestorasi Backup Lengkap

Anda bisa merestorasi backup lengkap (yang pertama pada daftar) agar semua perubahan yang telah dilakukan, seperti penyisipan record 7 dan 8 serta penghapusan record 5, dapat dibatalkan. Dalam hal ini, pilihan yang dilakukan ditunjukkan pada gambar di bawah ini:

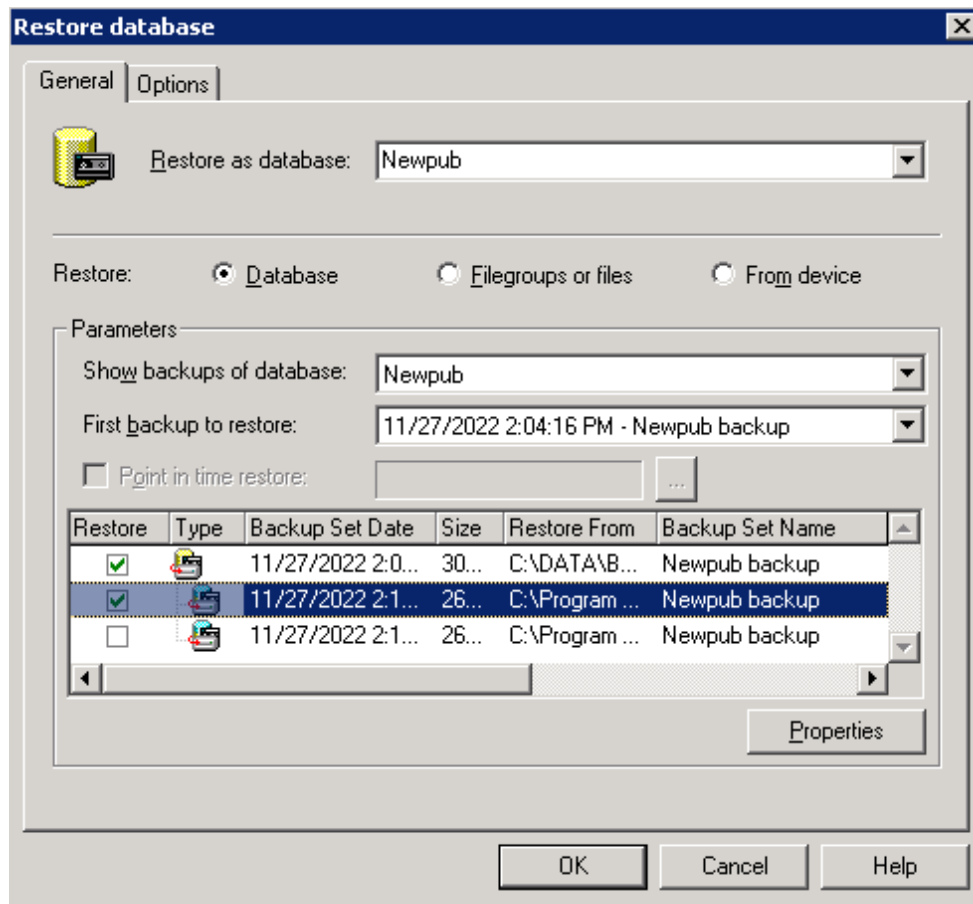


Tekanlah tombol OK yang menyelesaikan proses restorasi. Perhatikanlah isi dari tabel Test1.

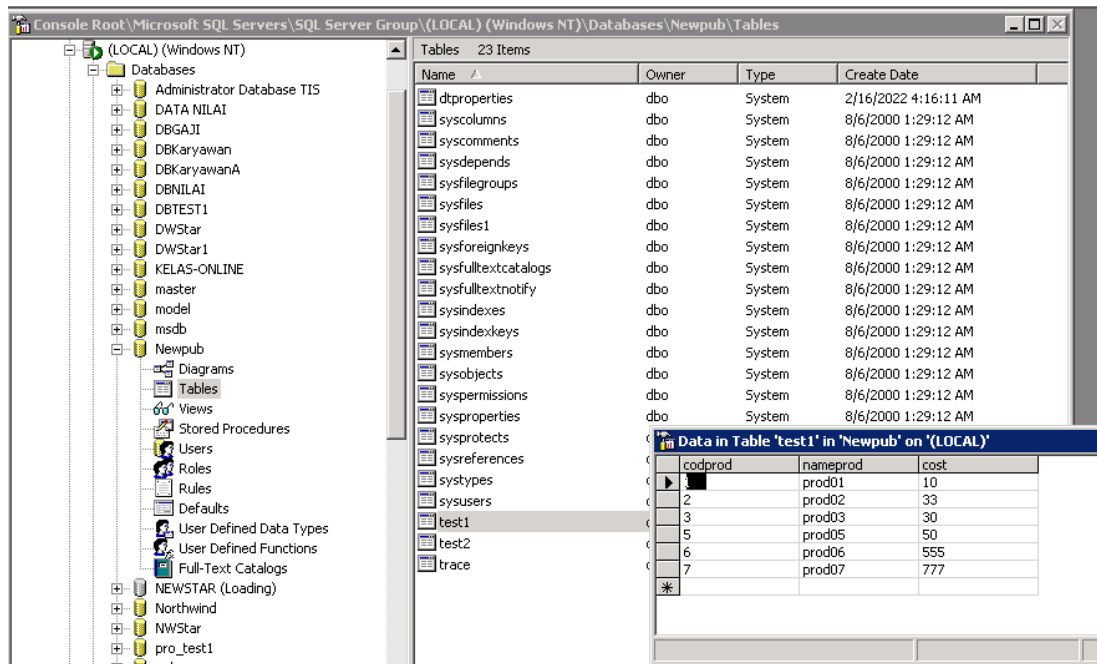


Merestorasi Backup Differensial

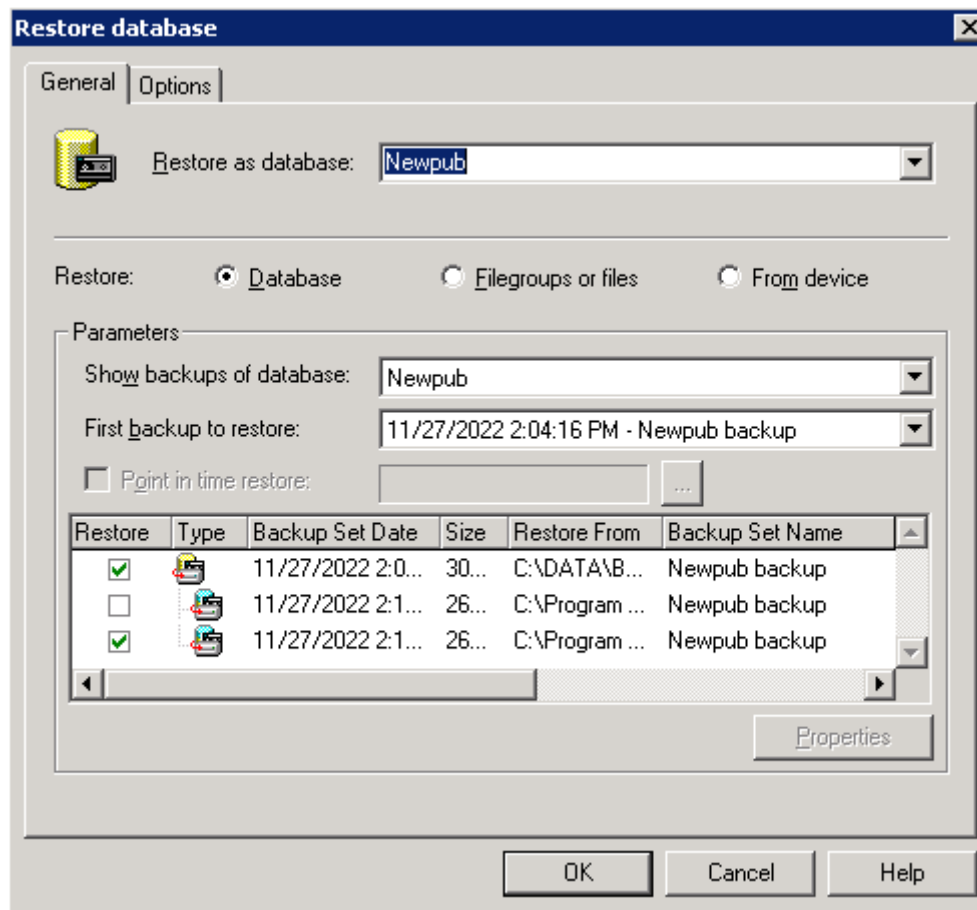
Untuk merestorasi backup differensial, pertama-tama pilihlah backup yang ingin dipulihkan. Memilih kotak pada kolom Restore pada backup differensial secara otomatis akan ikut memilih backup lengkap, karena pada operasi ini program akan merestorasi backup lengkap lalu backup differensial. Setelah memilih backup ini, tekan tombol OK.



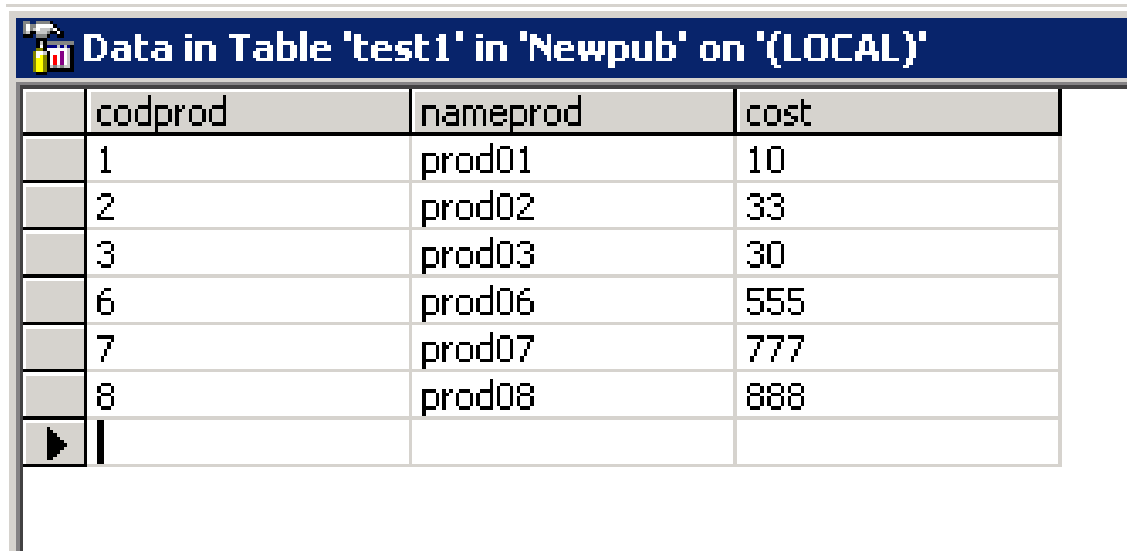
Perhatikanlah isi dari tabel Test 1:



Tabel ini menunjukkan codprod baris 7 yang disisipkan setelah backup lengkap yang terakhir. Kini kita akan melakukan proses restorasi dari backup diferensial yang kedua dengan memilih pada kotak Restore dan menekan tombol OK.



Perhatikan bahwa sekarang codprod record 5 sudah tidak ada lagi dan record 8 telah dikembalikan.



The screenshot shows a table titled "Data in Table 'test1' in 'Newpub' on '(LOCAL)'". The table has three columns: "codprod", "nameprod", and "cost". The data rows are as follows:

	codprod	nameprod	cost
	1	prod01	10
	2	prod02	33
	3	prod03	30
	6	prod06	555
	7	prod07	777
	8	prod08	888
▶			

Kesimpulan

Bab ini menunjukkan betapa mudahnya melakukan operasi backup dan pemulihan data di dalam sebuah database karena adanya antarmuka grafis yang digunakan oleh SQL Server. Dengan berakhirnya bab ini, kita telah menyelesaikan bagian tutorial dari buku ini. Bagian ini menunjukkan aspek-aspek yang paling penting pada SQL Server dan memberikan Anda gambaran singkat dan praktis. Berikutnya Anda akan menemukan panduan referensi tentang bahasa Transact-SQL. Panduan ini memberikan pengenalan mengenai sintaks serta kegunaan dari perintah-perintah dan fungsi yang terdapat di dalamnya.

DAFTAR PUSTAKA

1. Ramalho, Jose. 2011. SQL Server, PT. Gramedia.
2. E. F. Codd. A Relational Model of Data For Large Shared Data Banks. IBM Research Laboratory, San Jose, California. 1970.
<http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>
3. Michael L. Horowitz. An Introduction to Object-Oriented Databases and Database Systems. Information Technology Center Carnegie Mellon University Pittsburgh PA 15213. 1991.
<http://ra.adm.cs.cmu.edu/anon/itc/CMU-ITC-103.pdf>
4. Suk Kyoong Lee. An Extended Relational Database Model For Uncertain And Imprecise Information. Department of Computer Science, University of Iowa, Iowa City, IA. 52242.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.335&rep=rep1&type=pdf>
5. Kotagiri Ramamohanarao, James Harland. An Introduction to Deductive Database Languages and Systems.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.66.68&rep=rep1&type=pdf>
6. Michael J. Cullinan. Data Warehouse Concept. OIRAP. USA.
<http://oirap.rutgers.edu/dwbasics.pdf>
7. Jeffrey A. Hoffer, Mary B. Prescott, Fred R. McFadden. Modern Database Database Management Management 7th Edition Edition. Prentice Prentice Hall. 2005.
<http://www.itu.dk/people/pagh/DBS05/OLAP-MDM-2s.pdf>
8. Tutorial Point. Data Warehouse. Tutorial Point.
http://www.tutorialspoint.com/dwh/dwh_tutorial.pdf
9. Rohan Sharma, Kalpit Shah, Yeshesvini Shirahatti, Smruti Patel. Data Warehouse And OLAP Technology.
<http://www3.cs.stonybrook.edu/~cse634/presentations/Data Warehousing-part-1.pdf>
10. Oracle. Data Warehouse Concept. Oracle. 2001.
https://docs.oracle.com/cd/B13789_01/server.101/b10736/concept.htm
11. TLDP. Linux File System. TLDP.
http://www.tldp.org/LDP/intro-linux/html/sect_03_01.html
12. My Reading Room Site. On Line Analytical Processing And Its Characteristics. India.
http://www.myreadingroom.co.in/images/stories/docs/dbms/on_line%20analytical%20processing%20and%20its%20characteristics.pdf
13. Alberto Abell 'o, Oscar Romero. On Line Analytical Processing. Universitat Politecnica de Catalunya.
<http://www.essi.upc.edu/~aabello/publications/09.encyclopedia.pdf>
14. Robert] Davenport. ETL vs ELT: A Subjective View. Insource IT Consultancy Ltd. USA. June 2008.

- <http://www.dataacademy.com/files/ETL-vs-ELT-White-Paper.pdf>
15. Panos Vassiliadis, Alkis Simitis. Extraction, Transformation, and Loading. IBM Almaden Research Center. USA. 2005.
http://www.cs.uoi.gr/~pvassil/publications/2009_DB_encyclopedia/Extract-Transform-Load.pdf
 16. Oracle. Overview of Extraction, Transformation, and Loading. Oracle. 2015.
http://docs.oracle.com/cd/B19306_01/server.102/b14223/etlover.htm
 17. John Foley. The Top Ten Trends In Data Warehousing. Forbes, Oracle, USA. 2014.
<http://www.forbes.com/sites/oracle/2014/03/10/the-top-10-trends-in-data-warehousing/>
 18. Oracle Corp. Oracle Database 12c For Data Warehousing And Big Data. Oracle Corp. 2014.
<http://www.oracle.com/technetwork/database/bi-datawarehousing/data-warehousing-wp-12c-1896097.pdf>
 19. Suman Bhusan Bhattacharyya. Data Warehousing And Data Mining: An Overview. India. 2001.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.6649&rep=rep1&type=pdf>
 20. Mohit Kumar Gupta, Vishal Verma, Megha Singh Verma. In Memory Database Systems: A Paradigm Shift. Department of Computer Science, M L N College, Yamuna Nagar, India.
<http://arxiv.org/pdf/1402.1258.pdf>
 21. Hector Garcia Molina, Kenneth Salem. Main Memory Database Systems: An Overview. IEEE. 1992.
<http://pages.cs.wisc.edu/~jhuang/qual/main-memory-db-overview.pdf>
 22. SAP Team. Top Ten Questions For Choosing In Memory Databases. SAP Hana Platform.
http://hana.sap.com/content/dam/website/saphana/en_us/abouthana/Top_10_Questions_for_Choosing_In-Memory_Databases.pdf
 23. Stephen Tu, Wenting Zheng, Eddie Kohlert, Barbara Liskov, Samuel Madden. Speedy Transactions In Multicore In Memory Databases. Harvard University. 2013.
<http://db.csall.mit.edu/pubs/silo.pdf>
 24. Oracle Corp. Oracle Times Ten In Memory Database.
<http://www.oracle.com/technetwork/products/timesten/overview/ds-timesten-imdb-129255.pdf>
 25. Flash Memory Technology. Integrated Circuit Technology Corporation.
<http://smithsonianchips.si.edu/ice/cd/MEMORY97/SEC10.PDF>
 26. Deloitte. In Memory Computing Technology The Holy Grail Of Analytics. Deloitte. 2013.
http://www2.deloitte.com/content/dam/Deloitte/de/Documents/technology-media-telecommunications/TMT_Studie_In_MemoryComputing.pdf
 27. Jagan Singh Meena, Simon Min Sze, Umesh Chand, Tseung Yuen Tseng. Overview Of Emerging Non Volatile Memory Technologies. Meena Nanoscale Research Letters.
<http://www.nanoscalereslett.com/content/pdf/1556-276X-9-526.pdf>

28. Hewlett-Packard Development Company. Memory Technology Evolution: An Overview Of System Memory Technologies. Hewlett-Packard Development Company. 2007.
<http://users.ece.gatech.edu/~leehs/ECE6100/papers/DRAM-HP.pdf>
29. Oracle. Implementing Hybrid Columnar Compression On The Oracle ZFS Storage Appliance. Oracle. 2014.
<http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/problemsolver-hcc-52014-2202692.pdf>
30. Oracle. Hybrid Columnar Compression (HCC) On Exadata. Oracle. 2012.
<http://www.oracle.com/technetwork/middleware/bi-foundation/ehcc-twp-131254.pdf>
31. Dew Point. Hybrid Columnar Compression (HCC): A Unique Approach To Oracle Database Compression. Dew Point. 2013.
<http://www.dewpoint.com/sites/default/files/images/Dewpoint%20White%20Paper%20-%20Hybrid%20Columnar%20Compression%20HCC%20Jan2013.pdf>
32. Arun Murthy, Jeffrey Markham, Vinod Vavilapalli, Doug Eadline. Apache Hadoop YARN. Addison Wesley, USA.
http://hortonworks.com/wp-content/uploads/2013/12/Apache.Hadoop.YARN_.Sample.pdf
33. Petrie Wong, Zhian He, Eric Lo. Parallel Analytics As A Service. Department of Computing The Hong Kong Polytechnic University.
<http://www4.comp.polyu.edu.hk/~csc/llo/research/sigmod13.pdf>
34. Marin Dimitrov, Alex Simov, Yavor Petkov. Text Analytics And Linked Data Management As A Service With \$4. Ontotext AD, Bulgaria.
<http://ontotext.com/documents/publications/2015/Wasabi2015-ESWC-S4.pdf>
35. Cap Gemini. Spend Analytics As A Service. Cap Gemini. 2013.
https://www.pl.capgemini.com/resource-file-access/resource/pdf/spend_analytics.pdf
36. Roger Barga, Dennis Gannon, Jared Jackson, Jaliya Ekanayake, Wel Lu, Mohamed Fathalla. Cloud Data Analytics From Excel. eXtreme Computing Group (XCG) Microsoft.
http://research.microsoft.com/en-us/projects/azure/exceldatascope_11_poster1.pdf
37. David Jungwirth. Analytics As A Service Preview. Innovate. 2014.
<http://innovate.automic.com/downloads/Presentations/Tuesday,%20201.7/German%20website/Analytics-as-a-Service.pdf>
38. Oracle. Oracle Big Data Appliance X5-2. Oracle. 2015.
<http://www.oracle.com/technetwork/database/bigdata-appliance/overview/bigdataappliance-datasheet-1883358.pdf>
39. Oracle. Oracle Exadata Database Machine X5-2. Oracle. 2015.
<http://www.oracle.com/technetwork/database/exadata/exadata-x5-2-ds-2406241.pdf>
40. Prof Yike Guo. Current Status And Future Trends In Data Centric Science. Data Science Institute Imperial College London.

http://rois.ac.jp/topics/pdf/yike_guo2.pdf

41. Josh Bersin. The Datafication Of HR. Deloitte Review. 2014.
http://d2mtr37y39tpbu.cloudfront.net/wp-content/uploads/2014/01/DR14_The_Datafication_of_HR.pdf
42. Professor Mark Lycett. Datafication: Making Sense Of Big Data In A Complex World. School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, United Kingdom.
<http://core.ac.uk/download/pdf/19784773.pdf>
43. Dr C.E.A Mulligan. The Impact Of Datafication On Strategic Landscapes. Imperial College, London. Ericsson.
http://www.ericsson.com/industry-transformation/wp-content/uploads/sites/6/2014/04/LME0005_Datafication_IndTrans Teaser_final.pdf
44. Joseph M. Firestone, Ph.D. Data Warehouses and Data Marts: A Dynamic View. Executive Information System. White Paper. 1997.
<http://www.dkms.com/papers/dwdmdv.pdf>
45. Chuck Ballard, Amit Gupta, Vijaya Krishnan, Nelson Pessoa, Olaf Stephan. Data Mart Consolidation : Getting Control of Your Enterprise Information. IBM. 2005.
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246653.pdf>
46. Talus. A Practical Guide To Building Data Marts. Talus.
ftp://ftp.wiley.com/public/computer_books/updates/guide.pdf
47. Angela Bonifati, Fabiano Cattaneo, Alfonso Fuggetta, Stefano Paraboschi. Designing Data Marts For Data Warehouses. ACM. 2001.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.2850&rep=rep1&type=pdf>
48. James Standen. Data Warehouse VS Data Mart. Data Mart IST. 2008.
<http://www.datamartist.com/data-warehouse-vs-data-mart>
49. James Serra. Difference Between ETL And ELT.
<http://www.jamesserra.com/archive/2012/01/difference-between-etl-and-elt/>
50. Andreas de Ruitter. Designing An ETL Process With SSIS : Two Approaches To Extracting And Transforming Data. Microsoft. 2012.
<http://blogs.msdn.com/b/andreasderuitter/archive/2012/12/05 /designing-an-et-process-with-ssis-two-approaches-to-extracting-and-transforming-data.aspx>
51. Marc Bastien. Accessing Multi Dimensional Data Types In Oracle 9i Release 2. Oracle,
<http://ceur-ws.org/Vol-77/09 Bastien.pdf>
52. Keith Laker, Benefits Of A Multi Dimensional Model. Oracle.2006.
<http://www.oracle.com/technetwork/developer-tools/warehouse/benefits.pdf>
53. Oracle, Oracle9i OLAP User's Guide Release 2 (9.2.0.2): Data Structures in Relational and Multidimensional Data Stores. Oracle.

http://docs.oracle.com/cd/B10501_01/olap.920/a95295/designd5.htm

54. Nigel Pendse. What Is OLAP? An Analysis Of What The Often Misused OLAP Term Is Supposed To Mean. The OLAP Report. 2005.

<http://www.olapreport.com/FASML.HTM>

55. Li Niu, et al. Cognition-Driven Decision Support for Business Intelligence: Models, Techniques, Systems and Applications. Springer International Publishing. 2009.

<http://link.springer.com/book/10.1007%2F978-3-642-03208-0>

56. Steve Williams. The Profit Impact Of Business Intelligence. Morgan Kaufmann. 2001.

<https://www.elsevier.com/books/the-profit-impact-of-business-intelligence/williams/978-0-12-372499-1>

ISBN 978-623-8120-01-7 (PDF)

