

SQL : Structred Query Language. Yapısal sorgulama dilidir. Veritabandaki verileri yönetmek için bir çeşit kodlama dilidir.

Veritabanı : İçerisinde veri barındıran yazılımlardır. Verileri listeler halinde tablo ve satırlarda tutan yapıdır. Veritabanları, tablolar, kolonlar satırlar ve index'lerden oluşur temel anlamda. Index'ler, verinin daha hızlı çekilebilmesi adına veriye bir index vererek tutar.

Veritabanı Sunucu : Veritabanı sunucusu network üzerinden belirlenen bir porttan sistemi dinler, kendisine gelen komutlara göre istenilen veriyi istemciye gönderir. Gelen komutlarla işlem gerçekleştirir, bu işlemleri gerçekleştirirken tamamen kendi cpu, ram vs kendi kaynaklarını kullanır. SQL Server, MYSQL, PostreSQL, Oracle gibi sistemler veritabanı sunucusudur. Excel veritabanı sunucusu değildir çünkü bir excel dosyası açtığımızda kendi bilgisayaramızın kaynaklarını kullanırız.

Client & Veritabanı Sunucusu : Client bilgisayar, veritabanından mesela faturalar listesini çekip ekranda göstermek istesin, bunun için veritabanı sunucusuna bağlanması gerekiyor öncelikle, fiziksel bağlantı şart, bu fiziksel bağlantı üzerinde portların açık olmalı mesela firewall varsa fiziksel bağlantı gerçekleşemez. Fiziksel bağlantı sağlanırsa, bir sonraki aşama fiziksel bağlantıdan sonraki herkes bu veritabanı sunucusuna girememesi için kullanıcı adı ve şifreyle korunmuş olması gerekir. İstemci bilgisayar belli protokollerle bilgi gönderiyor, kullanıcı adı şifresiyle birlikte. Veritabanı sunucusu bilgileri kontrol ediyor ve mesela kullanıcı adı şifre yanlışsa girişe izin vermiyor. Eğer doğru kullanıcı adı ve şifreyle istemci bilgisayar bağlanırsa, bağlantı kabul edildi bilgisiyle giriş sağlanır. Artık istemci ve sunucu birbirine bağlı, istemcinin gönderdiği komutları sunucu çalıştırabilir. Mesela fatura listesini istiyordu, sql komutu gönderir. SQL komutu çünkü bir veritabanı sunucusu sql diliyle hareket edebilir. Bu yüzden sql dili bütün veritabanları için ortak bir dildir. Bazı veritabanları sql dilini kendine has geliştirmiştir fakat temel sql cümleleri her veritabanında aynıdır. Client, veritabanı sunucusuna select * from fatura diye komut gönderir, sunucu bu komutu kontrol eder herhangi bir yanlışlık yok ise client' a istediği veriyi binary şekilde gönderir. Ve client'te binary gelen data herhangi bir kontrol üzerinde gösterilir. Veritabanından sadece veri çekilmez veri ekleme, silme ve güncelleme işlemleride yapılabilir. Mesela client komut gönderir veri eklemek için insert komutuyla, veritabanı sunucusu herhangi bir yanlışlık bulamazsa veriyi kaydeder sunucuya. Client'ada bilgi verir eklendiğine dair.

FOREIGN KEY : Tablolar arası ilişkilendirmeyi sağlar. PK ile FK bağlanır, PK'da olmayan bir ID bağlandığı FK id'de yazılırsa hata verir. Böylece veri tutarlılığı korunmuş olur.

Foreign Keyler, primary keyler bunların hepsi nasıl ki validation amacı kullanıcıdan doğru bilgiyi almak ise foreign key, primary key'de doğru bilgi tutmak içindir. Mesela x tablo id'si ile y tablodaki bir kolon bağlandı. Artık y tablosundaki o kolona ancak x tablosundaki ID'ler gelebilir.

Primary Key : Tekrar etmeyen alan kısaca. Mesela ID primary key olsun, ve otomatik artan olsun, otomatik artacağından otomatikman tekrar etmeyen alan olacağından primary key olarak işaretlenebilir. Mesela müşteri tel tutulucaksa, tekrar etmeyeceği için onuda primary key olarak işaretleyebilirdik fakat otomatik artan alanın yani ID alanının primary key olarak işaretlenmesi performans açısından daha iyidir. Bir tabloda bir tane primary key olabilir.

Constraint : Verileri eklemeyi kısıtlamak.

Null Constraint : Null alan geçişini engellemek için.

Check Constraint : Mesela şu gelen datalar öğrenci tablosundansa, admin tabanında YETKİİD kısmına alma gibi kısıtlamalardır.

Default Constraint : Sütun için default değer.

Select : Görüntülemek, listelemek.

Primary key default'ta otomatikman index'i 0'dır.

İndex'ler 4'e ayrılır, Clustered Index (Satır Bazlı), Non-Clustered Index (Satır Bazlı), Clustered Column Store Index (Sütun Bazlı), Non-Clustered Column Store Index (Sütun Bazlı).

<https://medium.com/@dilanygt95/mssql-index-yapisi-2471d821c0cb>

Roll userı rolluserid üzerinden değil mesela veriliş tairihly arama yaparsak tablonun bir indexeda ihtiyacı olur. Mesela rol user dolmaya başlarsa, 1000-2000 veri olursa

Rol yönetimi defaultta istenir.

Tablolarda constraint belirtilmelidir. Önce tablo açılır, sonra tüm constraintler belirlenir sonra foreign key varsa belirlenir.

Constraintlerin hepsi transaction'ı bitirmeye yöneliktir. En önemli konu transaction, 2. Olarak datayla iş yapıldığından düzgün alabilmek için constraint'tir mssqlde.

MSSQL'de kullanılan özel işlem dilidir T-SQL.

Transact sql (T-SQL) dili 3

--DDL

--DML

--DCL olmak üzere 3 e ayrılır.

DDL (VERİ KONTROL DİLİ) -> create ,alter ,drop -> db açmak, db objeleri açmak, tabloları mesela güncellemek değiştirmek gibi. Yeni açılan şeyler ddl komutlarıyla yapılır mesela sağ tık create table arkada ddl komutları çalışır. Alter değiştirmek, drop silmek.

Örn -> Create Database example

Örn2 : Alter Table Class Alter Column OgrenciAdi Nvarchar(30) Not Null

Class tablosunda OgrenciAdi kolonu nvarchar ve not null değiştirildi.

Drop Database Okul – okul db silindi.

Veritabanı Manipülasyon Komutları

CREATE DATABASE : Yeni veritabanı açılır. Veritabanı objesi.

ALTER DATABASE : Mevcut veritabanının özelliklerini değiştirmek

CREATE TABLE : Tablo açmak

ALTER TABLE: Mevcut tablonun özelliklerini değiştirmek

DROP TABLE: Tabloyu tamamen siler.

CREATE INDEX : Bir index açılır.

DROP INDEX: Index silinir.

DML (VERİ İŞLEME DİLİ)

Manipüle etmek dilidir. Insert, update, delete, select.

Data Manipülasyon Komutları

Select: Kullanıcının herhangi bir tablodan herhangi kolonlara ulaşmasını sağlar. Veritabanındaki kayıtları çeker.

INSERT: Kullanıcını belirli bir tablodaki belirli alanlara veri yerleştirmesini sağlar.

UPDATE: Kullanıcının belirli bir tablodaki eski değerlerini yeni değerleri ile değiştirir.

DELETE: Kullanıcının belirli bir tablodaki belirli alanları silmesini sağlar.

TRUNCATE TABLE : Tablonun içeriğini tamamen siler. İlk oluşturulduğu haline döndürür. Delete'te ise belirli şartlara göre silmeler gerçekleştirilebilir.

Select Name, Surname FROM EMPLOYEES Where Kat = 2;

Insert INTO EMPLOYEES (adi,soyadi) VALUES (ferid, akşahin);

Update EMPLOYEES SET (name='ferid' AND surname = 'Akşahin') WHERE Tel=3254;

Teli 3254 olanın ad soyad bilgileri güncelleme

Delete EMPLOYEES where Tel= 3254;

Db objesi -> triggerlar, functionlar, stored procedures, view oluşturmak, table'lar vs.

Full backup -> Databasein tüm backupı.

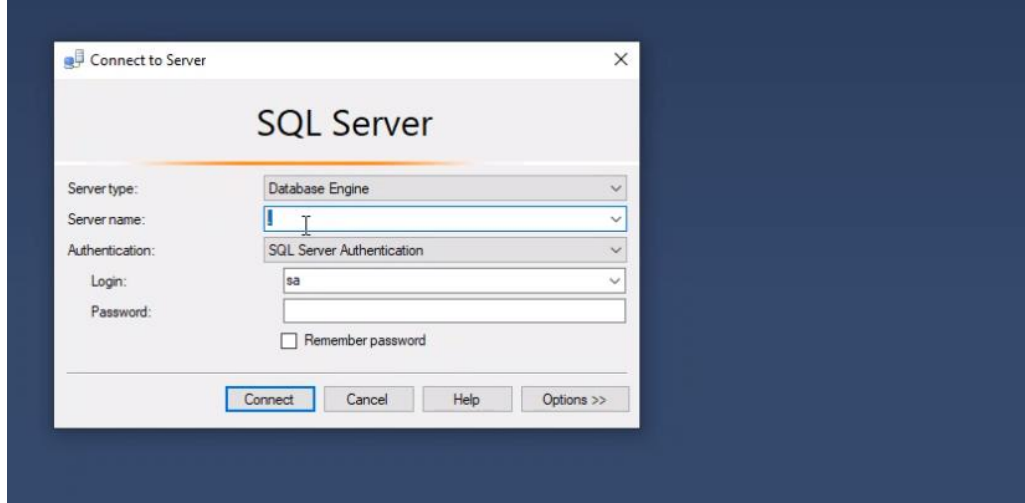
Differential backup -> Yedeklemek için seçtiğiniz dosyaların sadece Archive Bit değeri ON olanlar ve son Full backup'dan itibaren değişen ya da yeni dosyaları yedekler. Yedekleme işleminden sonra Archive Bit değeri OFF haline çekilmez.

ConnectionString : Bağlantının yapılabilmesi için hangi ana makine'ye bağlantı yapılacağını, o ana makinedeki hangi veritabanına bağlanacağımızı, o veritabanına bağlanmak için gerekli olan kullanıcı adı ve şifresi gibi bilgilerin tutulduğu bir kod.

Örn : Data Source = localhost; Initial Catalog = Northwind; Integrated Security = SSPI

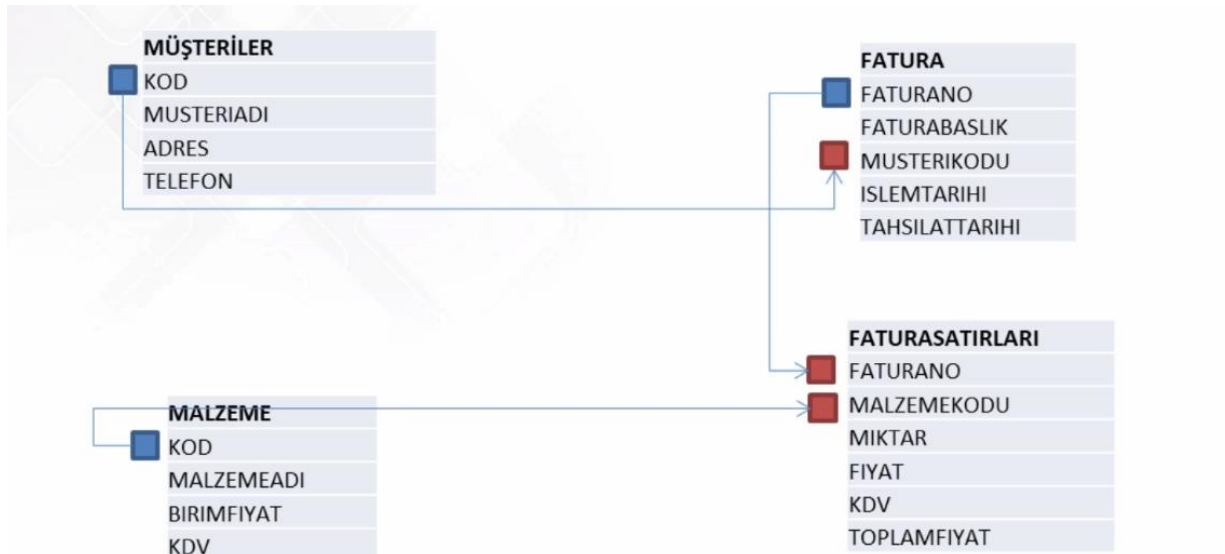
"Integrated Security = SSPI" -> bağlantı güvenliğinin ne şekilde yapılacağı. SSPI -> Windows Authentication seçeneğini kullanarak bağlanmış oluruz. Yani kullanıcı adı ve şifre girilmez.

Connection string arayüz



sa : system administrator

RDMS (Relational Database Management System) (İlişkisel Veritabanı Yönetim Sistemleri) : Tekrar eden verileri tekilleştirmek amacıyla yapılan veritabanı yönetim sistemleridir. Kısaca birden fazla tablo olan veritabanında, tabloların primary key'ini fk olarak diğer tabloya aktarmaktır. Böylece diğer tablo için her veriyi tekrar yazmak gerekmez, id ile o tablodaki o id'nin verilerine erişilebilir.



İlişkisel veritabanı yönetim sistemine bir örnektir. Müşterinin tüm bilgilerini faturada yazmak yerine, görüldüğü üzere fatura tablosunda, müşteri kodu ile müşteri tablosunda o faturaya ait müşterinin bilgilerine erişilebilir, aynı şekilde fatura no ile o faturaya ait bilgilere fatura satırları tablosundan erişilebilir, fature satırları tablosunda ise malzeme kodu ile malzeme tablosunda o fatura no için malzeme bilgilerine erişilebilir. RDMS uygulanmasaydı, veriler tablolarda ihtiyaç olduğundan her tablo için aynı veriler tekrar yazılacaktı.