**Yapısal Sorgulama Dili (SQL, Structured Query Language)**

1970’lerde ortaya çıkan SQL, kullanım amaçlarına bağlı olark bazı alt sınıflara ayrılmaktadır:

* Veri tanımla dili (DDL, data definition language)
* Veri işleme dili (DML, data manipulation language)
* Veri kontrol dili (DCL, data control language)

**Veri Tanımlama Dili Komutları**

*Veritabanı Nesnelerinin Oluşturulması:* CREATE komutu ile veritabanı nesneleri oluşturulur.

* CREATE DATABASE: Yeni veritabanı oluşturmak için kullanılır.

Genel Form

|  |
| --- |
| CREATE DATABASE <veritabanı\_adı>  CONTAINMENT = NONE|PARTIAL  ON PRIMARY  (  NAME = <veri\_dosyasının\_mantıksal\_adı>,  FILENAME = <veri\_dosyasının\_fiziksel\_adı>,  SIZE = <veri\_dosyasının\_başlangıç\_boyutu>,  MAXSIZE = <veri\_dosyasının\_maksimum\_boyutu>,  FILEGROWTH = <veri\_dosyasının\_büyüme\_miktarı>  )  LOG ON  (  NAME = <log\_dosyasının\_mantıksal\_adı>,  FILENAME = <log\_dosyasının\_fiziksel\_adı>,  SIZE = <log\_dosyasının\_başlangıç\_boyutu>,  MAXSIZE = <log\_dosyasının\_maksimum\_boyutu>,  FILEGROWTH = <log\_dosyasının\_büyüme\_miktarı>  ) |

CONTAINMENT = NONE ise veritabanı diğer veritabanlarından ve VTYS ayarlarından izole (tek başına çalışan) bir yapıdadır, CONTAINMENT = PARTIAL ise diğer veritabanları ve VTYS ayarlarının bir parçasıdır.

ON PRIMARY bölümüham veri dosyasına ilişkin ayarlamaların yapılmasını, LOG ON log dosyasına ilişkin ayarlamaların yapılmasını sağlar.

SQL Komutu

|  |
| --- |
| CREATE DATABASE [ogrenciDB2]  CONTAINMENT = NONE  ON PRIMARY  (  NAME = N'ogrenciDB2',  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\ogrenciDB2.mdf' ,  SIZE = 5120KB , MAXSIZE = UNLIMITED,  FILEGROWTH = 1024KB  )  LOG ON  (  NAME = N'ogrenciDB2\_log',  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\ogrenciDB2\_log.ldf' ,  SIZE = 2048KB ,  MAXSIZE = 2048GB ,  FILEGROWTH = 10%) |

* CREATE TABLE: Mevcut veritabanında yeni tablo oluşturmak için kullanılır.

Genel Form

|  |
| --- |
| CREATE TABLE <tablo\_adı>  (  <alan\_adı1> veri\_tipi()  NULL | NOT NULL  PRIMARY KEY  UNIQUE  DEFAULT <değer>  REFERENCES <tablo\_adı>(<alan\_adı>)  CHECK (<mantıksal\_ifade>)  IDENTITY(<başlangıç>,<artış\_miktarı>),  <alan\_adı2> veri\_tipi()  ...,  <alan\_adı3> veri\_tipi()  ...,  ...  <alan\_adın> veri\_tipi()  ...  ); |

*Kısıtlamalar*

NULL: Alana boş değer girileceğine izin verir.

NOT NULL: Alana boş değer girilmeyeceğini belirtir.

UNIQUE: Bu alanın değeri bazında tablo satırları tekrar edemez. Bu alan, tablo için bir anahtır.

DEFAULT: Alan için varsayılan değer tanımlar.

PRIMARY KEY: Tablodaki her bir kayda özel, o kaydı belirleyen anahtardır. Bu alan, aynı zamanda UNIQUE özelliğine de sahiptir. Bu alan, NULL değerine sahip olamaz. Herhangi bir tabloda birçok anahtar alan (UNIQUE) olabilir, fakat sadece bir adet PRIMARY KEY vardır.

REFERENCES: Tabloların alanları arasında yabancıl anahtar tanımlaması oluşturmak için kullanılır.

CHECK: Alana kaydedilecek değerler üzerinde sınırlama oluşturmak için kullanılır.

IDENTITY: Tablodaki bir alanın otomatik olarak artan bir değere sahip olması için kullanılır.

SQL Serverda sık kullanılan veri tipleri ve açıklamalarına bakınız.

SQL Komutu

|  |
| --- |
| CREATE TABLE ogrenci  (  OgrenciNo INT  NOT NULL  PRIMARY KEY  IDENTITY(1,1),  TCKimlikNo INT  NOT NULL  UNIQUE,  Adi VARCHAR(20)  NOT NULL,  Soyadi VARCHAR(30)  NOT NULL,  Bolumu VARCHAR(10)  NOT NULL  REFERENCES Bolum (Bolum\_Kodu),  Kayit\_Tarihi DATE  NOT NULL  DEFAULT (GETDATE()),  Yas INT  NULL  CHECK((Yas>6) AND (Yas<100)),  Ortalama INT  NULL  ) |

SQL Komutu-CREATE TABLE için genişletilmiş form örneği

|  |
| --- |
| CREATE TABLE ogrenci  (  ogrenciNo INT NOT NULL,  Adi VARCHAR(20) NOT NULL,  Soyadi VARCHAR(30) NOT NULL,  Bolumu VARCHAR(10) NULL,  CONSTRAINT cn\_check\_bolum CHECK (Bolumu='Endüstri' OR Bolumu='Makina' OR Bolumu='Bilgisayar'),  Kayit\_Tarihi DATE NOT NULL DEFAULT (getdate()),  Yas INT NULL,  Ortalama INT NULL,  CONSTRAINT cn\_primarykey\_ogrenci PRIMARY KEY CLUSTERED  (  OGRENCİnO asc  ),  Ders\_Kodu VARCHAR(10),  CONSTRAINT cn\_foreignkey\_derskodu FOREIGN KEY (Ders\_Kodu) REFERENCES Ders(Ders\_Kodu)  ) |

SQL Komutu-Hesaplanan alanlar ile ilgili örnek

|  |
| --- |
| CREATE TABLE OgrenciDers  (  Ogrenci\_No INT NOT NULL,  Ders\_Kodu VARCHAR(10) NOT NULL,  Kac\_Kez\_Aldi INT NULL,  Vize\_Notu INT NULL,  Final\_Notu INT NULL,  Sonuc1 AS  (  CONVERT(VARCHAR(10),  CASE  WHEN ( (0.40)\*Vize\_Notu+  (0.60)\*Final\_Notu) >=  (44.5) THEN 'Geçti'  ELSE 'Kaldı'  END  )  ),  CONSTRAINT PK\_OgrenciDers\_01 PRIMARY KEY CLUSTERED  (  Ogrenci\_No ASC,  Ders\_Kodu ASC  )  ); |

Hesaplanan alanlarda, alan tipinin belirtilmesine gerek yoktur. Çünkü, bu alanlar veritabanında tutulmazlar ve ihtiyaç halinde verilen fonksiyona uygun şekilde hesaplanırlar.

* CREATE VIEW: VIEW veritabanında tutulmayıp veritabanındaki mevcut verilerden dinamik olarak oluşturulan yapıdır.

VIEW’lerin veritabanında kullanılmasının temel nedenleri

* Programlama ortamındaki sorguları basitleştirmesi
* Veri bağımsızlığını sağlaması
* Yetkilendirme işlemlerinde basitliğin oluşturulması
* Ham veri tablolarının bazı kullanıcı erişimlerine kapatılması

Genel Form

|  |
| --- |
| CREATE VIEW <görüntü\_adı>  AS  SELECT ifadesi; |

SQL Komutu-Hesaplanan alanlar ile ilgili örnek

|  |
| --- |
| CREATE VIEW Gecen\_Ogr\_List  AS  SELECT \*  FROM OgrenciDers  WHERE OgrenciDers.Sonuc1='Geçti'; |

SQL Komutu-Oluşturulan VIEW’in kullanımı

|  |
| --- |
| SELECT \*  FROM Gecen\_Ogr\_List  WHERE Ders\_Kodu='MUH01'; |

* CREATE INDEX: Indeksler, sık kullanılan sorguların performansının artırılması için tablo veya görüntüler için oluşturulan yapılar olup bir veya birkaç alan üzerinde tanımlanır. VTYS, sadece seçilen alanların değerlerini tablodan ayrı bir yapıda sıralayıp alan değerleriyle tablodaki gerçek verileri ilişkilendirir. Böylece, gerçek verilerin bulunduğu adreslere hızlı bir şekilde erişilir.

Genel Form

|  |
| --- |
| CREATE [UNIQUE][CLUSTERED | NONCLUSTERED] INDEX <indeks\_adı>  ON <tablo\_veya\_görüntü\_adı>  (  <alan\_adı1> [ASC | DESC],  <alan\_adı2> [ASC | DESC],  ...  <alan\_adın> [ASC | DESC]  ); |

SQL Komutu

|  |
| --- |
| CREATE NONCLUSTERED INDEX idx\_ogrenci\_01  ON ogrenci  (  Adi ASC,  Soyadi ASC  ); |

*Veritabanı Nesnelerinin silinmesi:* DROP komutu ile veritabanı nesneleri kalıcı olarak silinir.

DROP DATABASE <veritabani\_adı>;

DROP TABLE <tablo\_adı>;

DROP VIEW <görüntü\_adı>;

DROP INDEX <tablo\_adı>.<indeks\_adı>;

*Tablo Yapısı ve Kısıtlamaların Düzenlenmesi:* ALTER komutu ile veritabanı nesneleri düzenlenir.

Genel Form

|  |
| --- |
| --Yeni alan ekleme  ALTER TABLE <tablo\_adi>  ADD <alan\_adı1> veri\_tipi()  NULL|NOT NULL  PRIMARY KEY  UNIQUE  DEFAULT <değer>  REFERENCES <tablo\_adı>(<alan\_adı>)  CHECK (<mantıksal\_ifade>)  IDENTITY(<başlangıç>,<artış\_miktarı>);  --Mevcut alan üzerine yeni sınırlama ekleme  ALTER TABLE <tablo\_adı> [WITH NOCHECK]  ADD CONSTRAINT <kısıtlama\_adı>  CHECK(<mantıksal\_ifade>);  --Mevcut alan için varsayılan değer tanımlama  ALTER TABLE <tablo\_adı> [WITH NOCHECK]  ADD CONSTRAINT <kısıtlama\_adı>  DEFAULT <değer/fonksiyon> FOR <alan\_adı>;  --Mevcut alanı birincil anahtar yapma  ALTER TABLE <tablo\_adı> [WITH NOCHECK]  ADD CONSTRAINT <kısıtlama\_adı>  PRIMARY KEY CLUSTERED (<alan\_adı>);  --Mevcut alan için yabancıl anahtar tanımlama  ALTER TABLE <tablo\_adı> [WITH NOCHECK]  ADD CONSTRAINT <kısıtlama\_adı>  FOREIGN KEY (<alan\_adı>)  REFERENCES <hedef\_tablo\_adı> (<hedef\_alan\_adı>);  --Mevcut bir kısıtlamanın silinmesi  ALTER TABLE <tablo\_adı>  DROP CONSTRAINT <kısıtlama\_adı>; |

**WITH NOCHECK** ile kısıtlama oluşturulurken mevcut verilerin yeni kısıtlamayı sağlayıp sağlamadığının kontrolünün yapılmamasını ifade eder.

SQL Komutu

|  |
| --- |
| ALTER TABLE ogrenci  ADD column\_b VARCHAR(20) NULL  CONSTRAINT exb\_unique UNIQUE;  ALTER TABLE ogrenci  ADD CONSTRAINT exd\_check CHECK (Yas>1);  ALTER TABLE ogrenci  ADD CONSTRAINT col\_b\_def DEFAULT 20 FOR Yas;  ALTER TABLE ogrenci WITH NOCHECK  ADD CONSTRAINT idx\_pk\_ogrenciDEL22  PRIMARY KEY CLUSTERED (ogrenciNo);  ALTER TABLE ogrenci WITH NOCHECK  ADD CONSTRAINT FK\_ContactBacup\_Contact  FOREIGN KEY (Bolumu)  REFERENCES Bolum(Bolum\_Kodu);  ALTER TABLE ogrenci  DROP CONSTRAINT [cn\_primarykey\_ogrenci]; |

**Veri İşleme Dili Komutları**

*Veritabanı Sorgulama:* SELECT komutu ile veritabanında bulunan mevcut veriler getirilir.

* CREATE DATABASE
* CREATE TABLE
* CREATE VIEW
* CREATE INDEX