

Veri Tabanı

6. Hafta Dersi



Dersin Hedefleri

- SQL – Yapısal Sorgulama Dili
 - Veri Tanımlama Dili (DDL)
 - Create
 - Alert
 - Drop

SQL – Yapısal Sorgulama Dili

- i. SQL: Structured Query Language
- ii. IBM tarafından sql standartları tanımlanmış ardından ISO ve ANSI tarafından da bir standart olarak kabule edilmiştir.
- iii. SQL bir programlama dili değildir.
- iv. SQL komutları kullanarak veritabanı ve tablo oluşturma, kayıt ekleme, kayıt güncelleme, silme ve listeleme işlemleri yapılır.
- v. Program geliştirme aşamasında sql tek başına yeteli olmadığı için ayrıca programlama dillerinden faydalanılır.

SQL – Yapısal Sorgulama Dili

vi. Standart sql ifadelerinde fonksiyon döngü ve karşılaştırma gibi programlamaya yönelik ifadeler kullanılamamaktadır. Bu nedenle PL/SQL ve T-SQL sorgulama dilleri geliştirilmiştir.

PL/SQL: Oracle tarafından geliştirilen ve oracle veritabanı sistemlerine özel dildir. Temel sql ifadelerinin yanında akış kontrolleri ve değişken kullanımına imkan sağlar.

T-SQL: Microsoft ve Sybase tarafından geliştirilmiştir. Temel sql ifadelerinin yanında akış kontrolleri ve değişken kullanımına imkan sağlar.

SQL – Yapısal Sorgulama Dili

Sql ifadeleri yapısal olara 3 gruba ayrılır.

- 1) **DDL** (Data Definition Language – Veri tanımlama dili)
- 2) **DML** (Data Manipulation language – Veri işleme dili)
- 3) **DCL** (Data Control Language – Veri kontrol dili)

1- Veri Tanımlama Dili (DDL)

- **Create - Alter – Drop** komutları bu gruptadır.
- Veri tanımlama dili verinin ne olduğundan çok verinin tipi ile ilgilenir.
- Veri tabanı oluşturma, tablo oluşturma, tablo içindeki alanların türleri, anahtar alanların belirlenmesi gibi işlemleri gerçekleştirmek için kullanılır.

1- Veri Tanımlama Dili (DDL)

VERİ TABANI OLUŞTURMA

- Create Database komutu ile gerçekleştirilir.
- Yeni bir veri tabanı oluşturulmasını veya daha önce oluşturulan veri tabanına ait dosyaların kullanılmasını sağlar.
- **CREATE DATABASE** veritabanı_ismi

1- Veri Tanımlama Dili (DDL)

- SQL Server Transact-SQL
- Create Database komutu farklı parametreler alabilmektedir.
- <https://msdn.microsoft.com/tr-tr/library/ms176061.aspx>

1- Veri Tanımlama Dili (DDL)

- Örnek:

```
CREATE DATABASE db_ismi
ON (
    NAME = dosya_ismi,
    FILENAME = 'fiziki_dosya_ismi',
    SIZE = dosya_boyutu,
    MAXSIZE = maksimum_boyut,
    FILEGROWTH = dosya_artiş_miktari )
LOG ON (
    NAME = ' = dosya_ismi',
    FILENAME = ' fiziki_dosya_ismi',
    SIZE = dosya_boyutu,
    MAXSIZE = maksimum_boyut,
    FILEGROWTH = dosya_artiş_miktari )
```

1- Veri Tanımlama Dili (DDL)

- **TABLO OLUŞTURMA**
- Create Table komutu ile gerçekleştirilir.
- Kullanımı:

CREATE TABLE tablo_ismi
(sutun_ismi1 veri_tipi, ek_ozellikler
sutun_ismi2 veri_tipi)
Şeklindedir.

- http://www.w3schools.com/sql/sql_create_table.asp

1- Veri Tanımlama Dili (DDL)

- **CONSTRAINT (KISITLAMA)**
- Veri tabanı sistemlerinde veri bütünlüğünü sağlamak için CONSTRAINT olarak isimlendirilen bazı zorlayıcı ifadeler de kullanılabilir.
- **NULL/NOT NULL, PRIMARY KEY, FOREIGN KEY, CHECK, UNIQUE, DEFAULT**

1- Veri Tanımlama Dili (DDL)

- **NULL / NOT NULL**
- Tablo alanının boş değer içerip içermeyeceğini tanımlar.
- Örnek

```
Create table ogrenciler  
(ogr_no int NOT NULL,  
Ad varchar(25)  
)
```

1- Veri Tanımlama Dili (DDL)

- **UNIQUE**
- Tablo alanının benzersiz olmasını sağlar. UNIQUE olarak belirlenen alan içerisinde bir değer iki sefer kullanılamaz.

1- Veri Tanımlama Dili (DDL)

- **UNIQUE**
- **SQL Server / Oracle / MS Access:**
- CREATE TABLE Persons
(
P_Id int NOT NULL UNIQUE,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255)
)

1- Veri Tanımlama Dili (DDL)

- **UNIQUE**
- **MySQL:**
- CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
UNIQUE (P_Id)
)

1- Veri Tanımlama Dili (DDL)

- Birden fazla kolonda **UNIQUE** tanımlama
- **MySQL / SQL Server / Oracle / MS Access:**
- **CREATE TABLE Persons**
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CONSTRAINT uc_PersonID UNIQUE (P_Id,LastName)
)

1- Veri Tanımlama Dili (DDL)

- **PRIMARY KEY tanımlama:**
- Primary key olarak tanımlanacak alan oluşturulurken ek özellik olarak PRIMARY KEY yazılması veya constraint olarak belirtilidir.

1- Veri Tanımlama Dili (DDL)

- **PRIMARY KEY** tanımlama:
- **SQL Server / Oracle / MS Access:**
- CREATE TABLE Persons
(
P_Id int NOT NULL PRIMARY KEY,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255)
)

1- Veri Tanımlama Dili (DDL)

- **PRIMARY KEY** tanımlama:
- **MySQL:**
- CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
PRIMARY KEY (P_Id)
)

1- Veri Tanımlama Dili (DDL)

- **Birden fazla kolonu birleştirerek PRIMARY KEY tanımlama:**
- CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CONSTRAINT pk_PersonID PRIMARY KEY (P_Id,LastName)
)

1- Veri Tanımlama Dili (DDL)

- **FOREIGN KEY TANIMLAMA**

Persons

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

Orders

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	2
4	24562	1

- P_Id kolonu Persons tablosunda Primary Key dir.
- Orders tablosunda ise Foreign Key olarak kullanılacaktır.

1- Veri Tanımlama Dili (DDL)

- **FOREIGN KEY TANIMLAMA**

- **MySQL:**

- CREATE TABLE Orders

(

O_Id int NOT NULL,

OrderNo int NOT NULL,

P_Id int,

PRIMARY KEY (O_Id),

FOREIGN KEY (P_Id) REFERENCES Persons(P_Id)

)

1- Veri Tanımlama Dili (DDL)

- **FOREIGN KEY TANIMLAMA**

- **SQL Server / Oracle / MS Access:**

- CREATE TABLE Orders

```
(  
O_Id int NOT NULL PRIMARY KEY,  
OrderNo int NOT NULL,  
P_Id int FOREIGN KEY REFERENCES Persons(P_Id)  
)
```

1- Veri Tanımlama Dili (DDL)

- **Constraint kullanımı ile Foreign Key tanımlama**
- **MySQL / SQL Server / Oracle / MS Access:**
- **CREATE TABLE Orders**
(
O_Id int NOT NULL,
OrderNo int NOT NULL,
P_Id int,
PRIMARY KEY (O_Id),
CONSTRAINT fk_PerOrders FOREIGN KEY (P_Id)
REFERENCES Persons(P_Id)
)

1- Veri Tanımlama Dili (DDL)

- **Foreign Key**
- `CONSTRAINT fk_mkod FOREIGN KEY(mkod)
REFERENCES musteri(mkod) ON UPDATE CASCADE
ON DELETE CASCADE`
- `CONSTRAINT fk_aracno FOREIGN KEY(aracno)
REFERENCES musteri(aracno) ON UPDATE CASCADE
ON DELETE CASCADE`

1- Veri Tanımlama Dili (DDL)

- **CHECK Kullanımı**
- Check kısıtlaması bir sütunun alabileceği değer aralığını sınırlamak için kullanılır.
- Örneğin notlar tablosunda vize ve final sütunları olduğunu düşünülduğünde bu kolonların 0 dan küçük değer almaması gerekir. Aynı şekilde 100 den büyük değer alması da önlenmelidir. Check kısıtlaması ile bu işlemler yapılabilmektedir.

1- Veri Tanımlama Dili (DDL)

- **CHECK Kullanımı**
- **MySQL:**
- CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CHECK (P_Id>0)
)

1- Veri Tanımlama Dili (DDL)

- **CHECK Kullanımı**
- **SQL Server / Oracle / MS Access:**
- **CREATE TABLE Persons**
(
P_Id int NOT NULL CHECK (P_Id>0),
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255)
)

1- Veri Tanımlama Dili (DDL)

- **Constraint ile CHECK Kullanımı ve birden fazla sütunda koşul oluşturma**
- **MySQL / SQL Server / Oracle / MS Access:**
- **CREATE TABLE Persons**
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CONSTRAINT chk_Person CHECK (P_Id>0 AND City='Sandnes')
)

1- Veri Tanımlama Dili (DDL)

- **DEFAULT Kullanımı**
- Default kısıtlaması kolona varsayılan bir değer eklemek için kullanılır.
- Varsayılan değer, kolan için değer belirtilmemiş tüm kayıtlara uygulanır.

1- Veri Tanımlama Dili (DDL)

- **DEFAULT Kullanımı**
- **My SQL / SQL Server / Oracle / MS Access:**
- **CREATE TABLE Persons**
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255) DEFAULT 'Sandnes'
)

1- Veri Tanımlama Dili (DDL)

- **DEFAULT** kısıtlaması sistem değerlerini kullanan fonksiyonlar ile birlikte kullanılabilir.
- CREATE TABLE Orders
(
O_Id int NOT NULL,
OrderNo int NOT NULL,
P_Id int,
OrderDate date DEFAULT **GETDATE()**
)

1- Veri Tanımlama Dili (DDL)

- **OTOMATİK ARTTIRMA**
- MS SQL Server'da otomatik artış için
- IDENTITY(1,1) ifadesi kullanılır.
- 1 den başlayıp birer birer artsın anlamına gelmektedir.

1- Veri Tanımlama Dili (DDL)

- **Örnek**

- Create table ogrenciler

(ogr_no int IDENTITY(1,1) NOT NULL,

Ad nchar(25)

)

1- Veri Tanımlama Dili (DDL)

- **Sütun içindeki değerleri otomatik arttırma:**
- MySQL de ise `AUTO_INCREMENT` ifadesi kullanılır.

1- Veri Tanımlama Dili (DDL)

- **Örnek**

- Create table ogrenciler

(ogr_no int NOT NULL AUTO_INCREMENT,

Ad varchar(25)

)AUTO_INCREMENT = 4;

1- Veri Tanımlama Dili (DDL)

VERİ TABANINDA DÜZENLEME YAPMA (ALTER)

- Daha önce oluşturulmuş veri tabanı nesnesinin özelliğini değiştirmek için kullanılır.
- Yapılmak istenen değişiklikler parametre olarak verilir.

1- Veri Tanımlama Dili (DDL)

VERİ TABANINDA DÜZENLEME YAPMA (ALTER)

- Kullanımı:

ALTER TABLE tablo_ismi İŞLEM sütun_adi

İŞLEM: ADD, DROP, ALTER

1- Veri Tanımlama Dili (DDL)

- **ALTER Komutunun VTYS lerde kullanımı**
- **SQL Server / MS Access:**
 - ALTER TABLE table_name
ALTER COLUMN column_name datatype
- **My SQL / Oracle (prior version 10G):**
 - ALTER TABLE table_name
MODIFY COLUMN column_name datatype
- **Oracle 10G and later:**
 - ALTER TABLE table_name
MODIFY column_name datatype

1- Veri Tanımlama Dili (DDL)

• Örnek:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

P_Id	LastName	FirstName	Address	City	DateOfBirth
1	Hansen	Ola	Timoteivn 10	Sandnes	
2	Svendson	Tove	Borgvn 23	Sandnes	
3	Pettersen	Kari	Storgt 20	Stavanger	

ALTER TABLE Persons
ADD DateOfBirth date



1- Veri Tanımlama Dili (DDL)

- Örnek:

P_Id	LastName	FirstName	Address	City	DateOfBirth
1	Hansen	Ola	Timoteivn 10	Sandnes	
2	Svendson	Tove	Borgvn 23	Sandnes	
3	Pettersen	Kari	Storgt 20	Stavanger	

ALTER TABLE Persons
ALTER COLUMN
DateOfBirth year

1- Veri Tanımlama Dili (DDL)

• Örnek:

P_Id	LastName	FirstName	Address	City	DateOfBirth
1	Hansen	Ola	Timoteivn 10	Sandnes	
2	Svendson	Tove	Borgvn 23	Sandnes	
3	Pettersen	Kari	Storgt 20	Stavanger	

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

**ALTER TABLE Persons
DROP COLUMN
DateOfBirth**

1- Veri Tanımlama Dili (DDL)

ALTER-UNIQUE KOMUTU KULLANIMI

- **MySQL / SQL Server / Oracle / MS Access:**
- ***ALTER TABLE Persons ADD UNIQUE (P_Id)***

Komutu Persons tablosunda P_Id sütununu Unique olarak tanımlayarak tabloda düzenleme işlemi gerçekleştirmiş olur.

1- Veri Tanımlama Dili (DDL)

ALTER-UNIQUE KOMUTU KULLANIMI

Constraint ile birlikte kullanımı:

```
ALTER TABLE Persons  
ADD CONSTRAINT uc_PersonID UNIQUE(P_Id,LastName)
```

Persons tablosuna yeni bir constraint oluşturur ve P_Id, LastName sütunlarını Unique yapar.

1- Veri Tanımlama Dili (DDL)

ALTER-UNIQUE KOMUTU KULLANIMI

Constraint ile birlikte kullanımı:

MySQL:

- ALTER TABLE Persons
DROP INDEX uc_PersonID

SQL Server / Oracle / MS Access:

- ALTER TABLE Persons
DROP CONSTRAINT uc_PersonID

Daha önce oluşturulmuş olan uc_PersonID isimli Constraint i siler.

1- Veri Tanımlama Dili (DDL)

ALTER-PRIMARY KEY KOMUTU KULLANIMI

- **MySQL / SQL Server / Oracle / MS Access:**
- **ALTER TABLE Persons ADD PRIMARY KEY (P_Id)**
- Daha önce oluşturulmuş olan Persons tablosundaki P_Id sütununu Primary Key olarak belirler.

1- Veri Tanımlama Dili (DDL)

ALTER-PRIMARY KEY KOMUTU KULLANIMI

- **MySQL / SQL Server / Oracle / MS Access:**
- **ALTER TABLE Persons
ADD CONSTRAINT pk_PersonID PRIMARY KEY
(P_Id,LastName)**

1- Veri Tanımlama Dili (DDL)

ALTER-PRIMARY KEY KOMUTU KULLANIMI

- **MySQL:**
- **ALTER TABLE Persons
DROP PRIMARY KEY**
- **SQL Server / Oracle / MS Access:**
- **ALTER TABLE Persons
DROP CONSTRAINT pk_PersonID**
- Persons tablosundaki birincil anahtar olan kolonun birincil anahtar özelliğini kaldırır.

1- Veri Tanımlama Dili (DDL)

ALTER-FOREIGN KEY KOMUTU KULLANIMI

- **MySQL / SQL Server / Oracle / MS Access:**
- **ALTER TABLE Orders
ADD FOREIGN KEY (P_Id)
REFERENCES Persons(P_Id)**

Orders tablosundaki P_Id sütununu Persons tablosundaki P_Id sütununu referans alan yabancı anahtar oluşmasını sağlar.

1- Veri Tanımlama Dili (DDL)

ALTER-FOREIGN KEY KOMUTU KULLANIMI

Constraint ile birlikte kullanımı

- **MySQL / SQL Server / Oracle / MS Access:**
- **ALTER TABLE Orders
ADD CONSTRAINT fk_PerOrders
FOREIGN KEY (P_Id)
REFERENCES Persons(P_Id)**

1- Veri Tanımlama Dili (DDL)

ALTER-FOREIGN KEY KOMUTU KULLANIMI

- **MySQL:**
 - ALTER TABLE Orders
DROP FOREIGN KEY fk_PerOrders
- **SQL Server / Oracle / MS Access:**
 - ALTER TABLE Orders
DROP CONSTRAINT fk_PerOrders
- *Orders tablosunda var olan sütunun yabancı anahtar özelliğini siler.*

1- Veri Tanımlama Dili (DDL)

ALTER-CHECK KOMUTU KULLANIMI

- **MySQL / SQL Server / Oracle / MS Access:**
- **ALTER TABLE Persons ADD CHECK (P_Id>0)**
- Persons tablosunun P_Id sütununa geçerlilik kuralı oluşturur.

1- Veri Tanımlama Dili (DDL)

ALTER-CHECK KOMUTU KULLANIMI

- Constraint ile birlikte kullanımı
- MySQL / SQL Server / Oracle / MS Access:
- ALTER TABLE Persons
ADD CONSTRAINT chk_Person CHECK (P_Id>0 AND
City='Sandnes')

1- Veri Tanımlama Dili (DDL)

ALTER-CHECK KOMUTU KULLANIMI

- **MySQL:**
- ALTER TABLE Persons
DROP CHECK chk_Person
- **SQL Server / Oracle / MS Access:**
- ALTER TABLE Persons
DROP CONSTRAINT chk_Person
- Oluşturulan geçerlilik kuralını kaldırır.

1- Veri Tanımlama Dili (DDL)

ALTER-DEFAULT KOMUTU KULLANIMI

- **MySQL:**
 - ALTER TABLE Persons
ALTER City SET DEFAULT 'SANDNES'
- **SQL Server / MS Access:**
 - ALTER TABLE Persons
ALTER COLUMN City SET DEFAULT 'SANDNES'
- **Oracle:**
 - ALTER TABLE Persons
MODIFY City DEFAULT 'SANDNES'
- Persons tablosunun City kolonuna varsayılan değer atar.

1- Veri Tanımlama Dili (DDL)

ALTER-DEFAULT KOMUTU KULLANIMI

- **MySQL:**
 - ALTER TABLE Persons
ALTER City DROP DEFAULT
- **SQL Server / Oracle / MS Access:**
 - ALTER TABLE Persons
ALTER COLUMN City DROP DEFAULT
- Persons tablosunun City kolonundaki varsayılan değeri kaldırır.

1- Veri Tanımlama Dili (DDL)

DROP ifadesi

Veri Tabanı nesnelerini silmeye yarar.

DROP TABLE table_name

DROP DATABASE database_name

1- Veri Tanımlama Dili (DDL)

NOT: TRUNCATE TABLE table_name

- Veri Tanımlama diline ait bir komut değildir.
- Tablo içindeki tüm kayıtları siler otomatik artan sütunu sıfırlar.
- Kayıtları fiziksel olarak DB den siler ve tabloyu boşaltır.

TEŞEKKÜRLER...

www.aliosmangokcan.com
mail@aliosmangokcan.com

KAYNAK: Turgut ÖZSEVEN- VTYS1

www.w3schools.com/sql