



BIL3203 – DATABASE MANAGEMENT



Alper VAHAPLAR

2021 – 2022 ©

Other SQL Operations

Structured Query Language - SQL



- **DDL (Data Definition Language)**
 - Database and table creation, update and delete operations,
- **DML (Data Manipulation Language)**
 - Data Entry, query, update and delete operations,
- **DCL (Data Control Language)**
 - Authorization and authentication operations.

Table Creation



- CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
column_name3 data_type,
....
)



Table Creation

```
■ CREATE TABLE Persons  
(  
  P_Id int,  
  LastName varchar(255),  
  FirstName varchar(255),  
  Address varchar(255),  
  City varchar(255)  
)
```



Table Creation – Key Definition

- CREATE TABLE Persons
(
P_Id AUTOINCREMENT NOT NULL PRIMARY KEY,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255)
)



Table Creation – Key Definition

- 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)
)

SQL – Data Manipulation

- sınıf (ogrno, adi, soyadi, dtarihi, sinifi)
- CREATE TABLE sınıf
(ogrno VARCHAR(10) PRIMARY KEY,
adi VARCHAR(15),
soyadi VARCHAR(15),
dtarihi INTEGER(4),
sinifi INTEGER(1))

ogrno	adi	soyadi	dtarihi	sinifi
2005285001	Alper	Vahaplar	1980	1

SQL – Data Insertion



- INSERT INTO table_name

VALUES (val₁, ..., val_N);

SQL – Data Insertion

- INSERT INTO table_name
VALUES (val₁, ..., val_N);
- Ex : sınıf (ogrno,_adi, soyadi, dtarihi,sinifi)
INSERT INTO sınıf
VALUES ('20062850','Hasan','Hüseyin',1980,1)

SQL – Data Insertion



- INSERT INTO table_name
 (field1, field2, ..., fieldN)
 VALUES (val1, val2, ..., valN)

SQL – Data Insertion

- `INSERT INTO table_name
(field1, field2, ..., fieldN)
VALUES (val1, val2, ..., valN)`
- Ex:
`INSERT INTO sinif (adi, ogrno)
VALUES ('Ayşe', '2005285012')`

SQL – Data Update



- UPDATE table_name
SET field1=value1, ..., fieldN=valueN;
- Ex: UPDATE **sinif** SET dtarihi=1988



SQL – Data Update

- UPDATE table_name
 SET field1=value1, ..., fieldN=valueN;
- Ex: Increment the “sinifi” value of everyone by 1
 UPDATE sinif
 SET sinifi=sinifi+1

SQL – Data Update

- UPDATE table_name
SET field1=value1, ..., fieldN=valueN;
WHERE condition
- Ex: UPDATE sinif
SET soyadi='Vahaplar'
WHERE ogrno='2005285001'



SQL – Data Update

- UPDATE table_name
 SET field1=value1, ..., fieldN=valueN;
 WHERE condition
- Ex: Decrement the year of student by 2 whose year is greater than 1980.
- UPDATE sinif
 SET dtarihi=dtarihi – 2
 WHERE dtarihi>1980

SQL – Exercises



- Ex: Set the year of student to 1999 whose sinifi is 3.
sinif (ogrno, adi, soyadi, dtarihi, sinifi)
- UPDATE sınıf SET dtarihi=1999
WHERE sinifi=3

SQL – Data Deletion



- DELETE FROM table_name

- Ex:

DELETE FROM sinif

SQL – Data Deletion

- DELETE FROM table_name
WHERE condition
- Ex: Delete the students with name 'Hasan'.
DELETE FROM sinif
WHERE adi='Hasan'

SQL – Data Deletion

- Ex: Delete the student with number 2008285001.
- `sinif (ogrno, adi, soyadi, dtarihi, sinifi)`
- `DELETE FROM sınıf`
`WHERE ogrno='2008285001'`



Removing Tables from Database

- DROP TABLE table_name
- Ex:

DROP TABLE Persons



Adding – Removing a column of a table

- Adding a new column
- ALTER TABLE table_name
ADD column_name datatype
- Ex:
ALTER TABLE Persons ADD DateOfBirth date
- Removing a column
- ALTER TABLE table_name
DROP column_name
- Ex:
ALTER TABLE Persons DROP DateOfBirth



Changing Table Data Type

- ALTER TABLE Persons
MODIFY column_name new_datatype
- Ex:
ALTER TABLE Persons
MODIFY City int

- A virtual table which contains columns from one or more tables.
- It does not contain any data directly.
- Imaginary tables or stored query results.
- Can be used as a table to *select*.
- No INSERT, UPDATE or DELETE queries work on.
- Why to use views?
 - To hide some rows or columns,
 - Joining multiple columns in different tables and use as a single table,
 - Organizing data,
 - Keep the results of some operations.

- **CREATE VIEW film AS**

```
SELECT title, year, name FROM movie,actor,cast
WHERE movie.id=cast.movieid
AND actor.id=cast.actorid
AND ord=1
```

title	year	name
Star Wars	1977	Mark Hamill
Star Wars: Episode V - The Empire Strikes Back	1980	Liam Neeson
Star Wars: Episode I - The Phantom Menace	1979	Mark Hamill
Back to the Future	1985	Michael J. Fox
Back to the Future Part II	1989	Michael J. Fox
Back to the Future Part III	1990	Michael J. Fox
Police Academy	1984	Steve Guttenberg
Police Academy 4: Citizens on Patrol	1987	Steve Guttenberg

Views



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the server structure for 'localhost\sqlexpress (SQL Serv)', including 'Databases', 'deneme', 'Tables', 'Views', and 'Security'. The 'deneme' database is expanded, showing 'dbo.kategori' and 'dbo.urun'. The main query window shows the following SQL query:

```
SELECT * FROM kategori
```

The Results pane displays the following data:

	kat_no	kat_adi	aciklama
1	1	Meyveler	Taze mevsim meyveleri
2	2	Sebzeler	Tarladan sofranıza sağlık kaynağı
3	3	Salata Malzemeleri	Şifalı otlar, yeşillikler

The status bar at the bottom indicates: Query executed succes... | localhost\sqlexpress (9.0 SP2) | manav (52) | deneme | 00:00:00 | 3 rows

Views



The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the 'deneme' database structure, including tables 'dbo.kategori' and 'dbo.urun'. The central query editor contains the SQL statement: `SELECT * FROM urun`. The Results pane at the bottom shows the output of this query as a table with 5 rows and 7 columns.

	urun_no	urun_adi	kat_no	birimi	fiyati	aciklama
1	1	Elma	1	kilo	3,50	Yeşil ve Sarı
2	2	Amut	1	kilo	4,75	Deveci
3	3	İspanak	2	kilo	1,00	Kuzu
4	4	Ananas	1	adet	5,25	Tropik
5	5	Taze Fasulye	2	kilo	4,00	Ayşe Kadın

The status bar at the bottom indicates: Query executed successfully | localhost\\sqlexpress (9.0 SP2) | manav (52) | deneme | 00:00:00 | 12 rows

Views



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'deneme', including tables 'dbo.kategori' and 'dbo.urun', and a view 'dbo.urunler'. The main query editor window contains the following SQL code:

```
--View oluşturmak
--Her ürünün adını, kategori adını, birimini ve fiyatını
--içeren bir view oluşturalım.
CREATE VIEW urunler AS
SELECT urun_adi, kat_adi, birimi, fiyatı
FROM urun, kategori
WHERE urun.kat_no=kategori.kat_no
```

The Messages pane at the bottom shows the execution result: "Command(s) completed successfully." The status bar at the bottom indicates the query was executed successfully on the 'deneme' database.

Views



Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query | [Icons]

deneme | Execute [Icons]

Object Explorer

Connect | [Icons]

- deneme
 - Database Diagrams
 - Tables
 - System Tables
 - dbo.kategori
 - dbo.urun
 - Views
 - System Views
 - dbo.urunler
 - Synonyms
 - Programmability
 - Service Broker
 - Security
- Security
- Server Objects
 - Backup Devices

SQLQuery10.sql - ...eme (manav (58))*

```
--View oluşturmak  
--Her ürünün adını, kategori adını, birimini ve fiyatını  
--içeren bir view oluşturalım.  
SELECT * FROM urunler
```

Results Messages

	urun_adi	kat_adi	birimi	fiyatı
1	Elma	Meyveler	kilo	3,85
2	Amut	Meyveler	kilo	4,75
3	İspanak	Sebzeler	kilo	1,00
4	Ananas	Meyveler	adet	5,25
5	Taze Fasulye	Sebzeler	kilo	4,00

Registered Servers | Object Explorer | Query executed succe... | localhost\\sqlexpress (9.0 SP2) | manav (58) | deneme | 00:00:00 | 12 rows

Ready | Ln 4 | Col 22 | Ch 22 | INS

Triggers



- A trigger is a code object that executes when a table is modified by an INSERT, UPDATE, or DELETE statement.
- Each trigger is associated with one table.
- FOR/AFTER , INSTEAD OF types.

Triggers



The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the 'deneme' database structure, including tables 'dbo.kategori', 'dbo.silinenurun', and 'dbo.urun', along with a newly created trigger 'sil'. The main query window shows the following SQL code:

```
--Trigger yaratma
--Ürün tablosundan bir ürün silindiğinde o kaydı
--silinenurun tablosuna kopyalanayan trigger.

CREATE trigger sil ON urun FOR DELETE
AS
INSERT into silinenurun SELECT * FROM deleted
```

The Messages pane at the bottom indicates that the command was completed successfully. The status bar at the bottom shows the query was executed successfully on 'localhost\\sqlexpress (9.0 SP2)' in the 'deneme' database, with 0 rows affected.

Triggers



Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query | [Icons]

deneme | Execute [Icons]

Object Explorer

Connect | [Icons]

deneme

- Database Diagrams
- Tables
 - System Tables
 - dbo.kategori
 - dbo.silinenurun
 - dbo.urun
 - Columns
 - Keys
 - Constraints
 - Triggers
 - sil
 - Indexes
 - Statistics
 - Views
 - Synonyms

SQLQuery13.sql - ...eme (manav (54))* SQLQuery10.sql - ...eme (manav (58))*

```
--Trigger yaratma
--Ürün tablosundan bir ürün silindiğinde o kaydı
--silinenurun tablosuna kopyalanayan trigger.

CREATE trigger sil ON urun INSTEAD OF DELETE
AS
INSERT into silinenurun SELECT * FROM deleted
```

Messages

Command(s) completed successfully.

Registered Servers | Object Explorer | Query executed succes... | localhost\sqlexpress (9.0 SP2) | manav (58) | deneme | 00:00:00 | 0 rows

Ready | Ln 5 | Col 38 | Ch 38 | INS