



T.C. İSTANBUL KÜLTÜR ÜNİVERSİTESİ

COM5041 - DATABASE DESIGN AND DEVELOPMENT

LAB 05 – Basic SQL

After completing this Lab, you will be able to

- Understand the SQL Data Definition and Data Types
- Create database and tables
- Specify constraints in SQL
- Write INSERT, DELETE, and UPDATE Statements in SQL
- Write Basic Retrieval Queries in SQL

PROCEDURE 1 - SQL Data Definition and Data Types

Step 1 – CREATE statement is main SQL command for data definition. The CREATE DATABASE statement is used to create a new SQL database.

```
SQLQuery1.sql - DE...M75C8G\hatic (74))* X S
CREATE DATABASE TheFirstDatabase;
```

Step 2- The CREATE TABLE statement is used to create a new table in a database. Create a table named Address using the information provided in the following table.

Name	Data Type	Length	Allow Nulls	Identity
AddressID	int	NA	No	Yes (start at 1 increment by 1)
StreetAddress	varchar	125	No	NA
StreetAddress2	varchar	75	Yes	NA
City	varchar	100	No	NA
State	char	2	No	NA
EmployeeID	int	NA	No	NA

```
SQLQuery1.sql - DE...M75C8G\hatic (74))* X
USE [TheFirstDatabase];
CREATE TABLE Address
(
    AddressID int NOT NULL IDENTITY(1,1),
    StreetAddress varchar(125) NOT NULL,
    StreetAddress2 varchar(75) NULL,
    City varchar(100) NOT NULL,
    State char(2) NOT NULL,
    EmployeeID int NOT NULL
)
```

Step 3 - Create the second table named Employee for TheFirstDatabase.

```
USE [TheFirstDatabase];
CREATE TABLE Employee
(
    EmployeeID int NOT NULL IDENTITY(1,1),
    FirstName varchar(50) NOT NULL,
    MiddleName varchar(50) NULL,
    LastName varchar(50) NOT NULL
)
```

Step 4 - If you want to change one of your tables, you can add columns, change columns, and drop columns using T-SQL.

```
USE [TheFirstDatabase];  
ALTER TABLE Employee  
ADD Gender char(1) NOT NULL;
```

Step 5 - You can you insert data directly into columns or you can also derive columns from other columns. These columns are known as computed columns. Typically, computed columns will extend or enhance the data that is stored in traditional columns.

```
USE [TheFirstDatabase];  
ALTER TABLE Employee  
ADD FullName AS LastName+', '+FirstName;
```

PROCEDURE 2 - Adding constraints to a table

Step 1 - SQL Server allows you to add several constraints to a table The primary goal of most constraints is data integrity. There are five constraints: primary key, default, unique, check, and foreign key.

```
USE [TheFirstDatabase];  
ALTER TABLE Employee  
ADD Active bit NOT NULL;  
  
ALTER TABLE Employee  
ADD SocialSecurityNumber varchar(10) NOT NULL;
```

Step 2 – We add the constraints: primary key, default and unique.

- A primary key is a column that contains a unique list of values Often an integer column is added to a table with the identity property and is used as the primary key.
- If you add an Active column to the Employee table that specifies whether an employee is currently working for the company, the default value will probably be true or 1. Therefore, you should set the default value for that column accordingly.
- For Social Security numbers to the Employee table. Since Social Security numbers are truly unique values, you should add a unique constraint to ensure that a given Social Security number is entered only once.

```

USE [TheFirstDatabase];
ALTER TABLE Employee
ADD CONSTRAINT PK_Employee_EmployeeID
PRIMARY KEY (EmployeeID);

ALTER TABLE Address
ADD CONSTRAINT PK_Address_AddressID
PRIMARY KEY (AddressID);

ALTER TABLE Employee
ADD CONSTRAINT DF_Employee_Active_True DEFAULT(1) FOR Active;

ALTER TABLE Employee
ADD CONSTRAINT UQ_Employee_SocialSecurityNumber
UNIQUE (SocialSecurityNumber);

```

Step 3 - To ensure that only employee IDs that exist in the Employee table are inserted into the Address table, you need to create a foreign key constraint.

```

USE [TheFirstDatabase];
ALTER TABLE Address
ADD CONSTRAINT FK_Employee_To_Address_On_EmployeeID
FOREIGN KEY (EmployeeID)
REFERENCES Employee(EmployeeID);

```

Step 4- Let's create a new table called 'Services', which is about the services and the prices of them given in the company to the customers.

```

USE TheFirstDatabase;
CREATE TABLE Servicess(
ServiceID int NOT NULL,
ServiceName varchar(125) NOT NULL,
Price money NOT NULL,
PRIMARY KEY (ServiceID)
)

```

Step 5- Now, we want to show the relationship between Services and Employees. So we create another table which is Services_by_Employees and we take the EmployeeID and ServiceID from certain tables, then declare them as primary keys of the 'Services_by_Employees' table.

```

USE [TheFirstDatabase]
CREATE TABLE Services_by_Employees1(
EmployeeID int NOT NULL,
ServiceID int NOT NULL,
PRIMARY KEY(EmployeeID, ServiceID),
FOREIGN KEY (EmployeeID) REFERENCES Employee,
FOREIGN KEY (ServiceID) REFERENCES Servicess
)

```

PROCEDURE 3- INSERT, DELETE, and UPDATE Statements in SQL

Step 1- Insert into statement is used to add one row or multiple rows to the table. Using this method, you can insert data into all columns, specific columns, identity columns, and several other variations.

- We can add data to Employee table using T-SQL.

```
USE TheFirstDatabase;  
INSERT INTO Employee(FirstName,LastName,Gender,Active,SocialSecurityNumber)  
VALUES  
( 'Hande', 'Koç', 'K',1, 'S100'),  
( 'Ahmet', 'Efe', 'M',0, 'S101'),  
( 'Ayşe', 'Gülen', 'K',0, 'S102'),  
( 'Merve', 'Yılmaz', 'K',1, 'S103'),  
( 'Murat', 'Gürol', 'M',1, 'S104')
```

- We can add data to Address table using T-SQL.

```
USE TheFirstDatabase;  
INSERT INTO Address(StreetAddress,City,State,EmployeeID)  
VALUES  
( 'Sağlık Mah. Mithat Paşa Cad. No:3 P.k. 06430', 'Ankara', 'TR',4),  
( 'Halide Edip Adivar Mahallesi - Darülaceze Caddesi No:3A BOMONTI', 'İstanbul', 'TR',5),  
( 'Atatürk Mah. 884.sk.No:6/A', 'İzmir', 'TR',6),  
( 'Bayındır Mh. Gazi Bulv. No: 2 Pk: 07030', 'Antalya', 'TR',7),  
( 'Zafer Mh. Ankara Yolu Cd. C Blok No:1 PK:16080', 'Bursa', 'TR',8)
```

Note that; The column list is optional in the INSERT INTO statement, but for purposes of clarity, it's always recommended. If it is not included, the values are inserted into the table based on the order of the columns. The identity columns are not included in the order.

Step 2- When you want to modify one row or several rows in table, you use UPDATE statement. You should be cautious when issuing an UPDATE statement, as it is highly unlikely that every row in a table needs updating. Therefore, always consider including a WHERE clause with every UPDATE statement.

- For the data in the Employee table where the social security number is S104, we change the value in the Active Column from 1 to 0.

```
USE TheFirstDatabase;  
UPDATE Employee  
SET Active = 0  
WHERE SocialSecurityNumber = 'S104'
```

Step 3 - When you want to delete one row or several rows in table, you use DELETE statement. As with the UPDATE statement, always consider including a WHERE clause when executing a DELETE statement. If you do not include WHERE as part of the DELETE statement, all the data will be removed after the query is executed.

- We delete the data in the Address table where the Employee ID is 8.

```
USE TheFirstDatabase;  
DELETE Address  
WHERE EmployeeID=8
```

PROCEDURE 4 - Basic Retrieval Queries in SQL

Step 1- SELECT statement is one basic statement for retrieving information from a database.

- Retrieve the Full Name and Social Security Number of the Employee(s) whose name is 'Hande'.

```
SELECT FullName, SocialSecurityNumber
FROM Employee
WHERE FirstName='Hande'
```

- Retrieve the Employee ID of all employees who live in TR.

```
SELECT EmployeeID
FROM Address
WHERE State='TR'
```

- Retrieve the First Name, Last Name, Gender and Street Address for all employees who live Antalya.

```
SELECT E.FirstName, E.LastName, E.Gender, A.StreetAddress
FROM Employee as E, Address as A
WHERE E.EmployeeID= A.EmployeeID AND A.City='Antalya'
```

Step 2- CROSS PRODUCT returns all combinations of Employee and Address tables in the database.

```
SELECT *
FROM Employee, Address
```

Step 3 – By placing DISTINCT immediately following the SELECT keyword, you remove any duplicates from the list. The following SQL statement selects only the DISTINCT values from the "Gender" column in the "Employee" table:

```
SELECT DISTINCT Gender
FROM Employee
```

Step 4- The UNION ALL command combines the result set of two or more SELECT statements (allows duplicate values). The following SQL statement returns the Employee ID (duplicate values also) from both the "Employee" and the "Address" table:

```
SELECT EmployeeID FROM Employee
UNION ALL
SELECT EmployeeID FROM Address
```

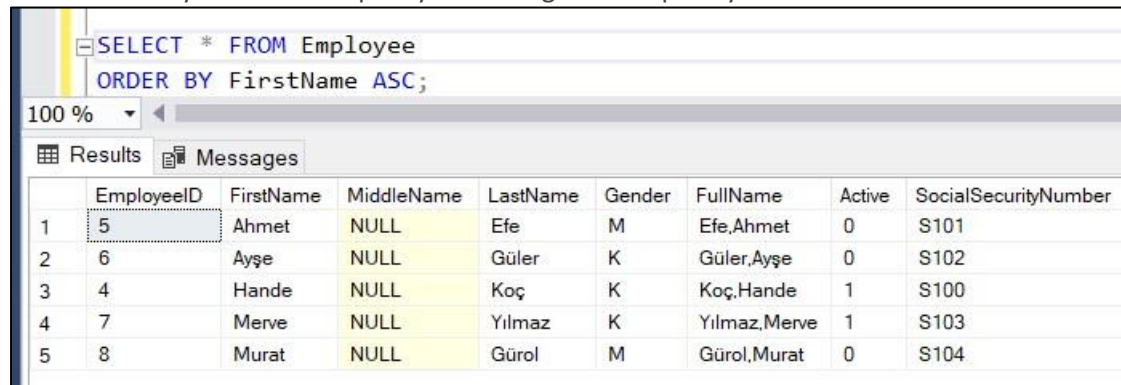
Step 5 - The UNION command combines the result set of two or more SELECT statements (only distinct values). The following SQL statement returns the Employee ID (only distinct values) from both the "Employee" and the "Address" table:

```
SELECT EmployeeID FROM Employee
UNION
SELECT EmployeeID FROM Address
```


Note that; When writing a query with UNION and UNION ALL, both SELECT statements must contain the same number of columns, and the data types must match for each column.

Step 5 - We use ORDER BY clause for ordering of query results.

- We add keyword ASC to specify ascending order explicitly.



The screenshot shows a SQL query window with the following text:

```
SELECT * FROM Employee
ORDER BY FirstName ASC;
```

Below the query window, the 'Results' tab is active, displaying a table with 9 columns: EmployeeID, FirstName, MiddleName, LastName, Gender, FullName, Active, and SocialSecurityNumber. The results are ordered by FirstName in ascending order.

	EmployeeID	FirstName	MiddleName	LastName	Gender	FullName	Active	SocialSecurityNumber
1	5	Ahmet	NULL	Efe	M	Efe,Ahmet	0	S101
2	6	Ayşe	NULL	Güler	K	Güler,Ayşe	0	S102
3	4	Hande	NULL	Koç	K	Koç,Hande	1	S100
4	7	Merve	NULL	Yılmaz	K	Yılmaz,Merve	1	S103
5	8	Murat	NULL	Gürol	M	Gürol,Murat	0	S104

- We add keyword DESC to see result in a descending order of values.



The screenshot shows a SQL query window with the following text:

```
SELECT * FROM Employee
ORDER BY FirstName DESC;
```

Below the query window, the 'Results' tab is active, displaying a table with 9 columns: EmployeeID, FirstName, MiddleName, LastName, Gender, FullName, Active, and SocialSecurityNumber. The results are ordered by FirstName in descending order.

	EmployeeID	FirstName	MiddleName	LastName	Gender	FullName	Active	SocialSecurityNumber
1	8	Murat	NULL	Gürol	M	Gürol,Murat	0	S104
2	7	Merve	NULL	Yılmaz	K	Yılmaz,Merve	1	S103
3	4	Hande	NULL	Koç	K	Koç,Hande	1	S100
4	6	Ayşe	NULL	Güler	K	Güler,Ayşe	0	S102
5	5	Ahmet	NULL	Efe	M	Efe,Ahmet	0	S101

SUMMARY

In this Lab, you learned to create tables that contain columns of varying data types. You learned how to add constraints to the tables and columns that assist in ensuring the consistency and validity of the inserted or modified data. In addition, you learned to write the simple retrieval queries and insert, update, delete commands for the tables.

PROCEDURE 5 - ASSIGNMENT (Upload the solution to the CATs by your ID)

You can access your assignment at the time your section starts. Then, you can download the assignment from the Assignments section in CATS.