

# Database Systems

Spring 2019

Lab Manual 4

---

**Instructor:** Muhammad Junaid Zaffar

**Purpose:**

- Joins
- Connecting SQL Server database with C# application

**Joins:**

- Cross Join
- Natural Join
- Equi Join
- Inner Join
- Left Outer Join
- Right Outer Join
- Full Outer Join

**C# Connection with database:**

There are eight (8) basic steps that must be followed in order to successfully communicate with a database. Let's take a detail overview of all these one by one.

**1. Import Required Package**

Import the package `System.Data.SqlClient` that contains useful classes and interfaces to access & work with database. `using System.Data.SqlClient;`

**2. Define Connection URL**

To get a connection, we need to specify the URL of a database (Actually we need to specify the address of the database which is in the form of URL)

```
String conURL =  
"Data Source=(local);Initial Catalog=MedicalEncyclopedia;Integrated  
Security=True;MultipleActiveResultSets=True";
```

```
String conURL =  
"Data Source=ServerName;Initial Catalog=DbName;User  
ID=userId;Password=password;MultipleActiveResultSets=True";
```

**3. Establish Connection With DataBase**

- Use `SqlConnection` to get the connection object.
- The URL of the database is passed to the `SqlConnection` constructor.

```
SqlConnection conn = new SqlConnection(conURL);
```

**4. Write statement**

```
String cmd = "SELECT * FROM TableName WHERE FirstColumn = @@";
```

**5. Create Statement**

Create sql command object

```
SqlCommand command = new SqlCommand(command, conn);  
// Add the parameters if required  
command.Parameters.Add(new SqlParameter("0", 1));
```

## 6. Execute a Query

The next step is to execute statements. Two methods are generally used for executing SQL queries. These are:

- **ExecuteReader(sql) method**

Used for SQL SELECT queries.

Returns the ResultSet object that contains the results of the query and can be used to access the query results.

```
SqlDataReader reader = command.ExecuteReader();
```

- **command.ExecuteNonQuery() method**

This method is used for executing an update statement like INSERT, UPDATE or DELETE. Returns an Integer value representing the number of rows updated.

```
int count = command.ExecuteNonQuery();
```

## 7. Process Results of the Query

- The ResultSet provides array index that takes a column index or name and returns the data as object which can then be casted into different types
- The ResultSet maintains the data in the form tables (rows & columns)
- First row has index 1, not 0.
- The read method of ResultSet returns true or false depending upon whether the next row is available (exist) or not and moves the cursor
- Always remember to call read() method at-least once
- To retrieve the data of the column of the current row you need to use the various getters provided by the ResultSet.
- For example, the following code snippet will iterate over the whole ResultSet

```
while (reader.Read())  
{  
    Console.WriteLine(reader[0]+" "+ reader[1]);  
}
```

## 8. Close the Connection

An opening connection is expensive, postpone this step if additional database operations are expected

```
con.Close();
```