**Ado Net**

Database ilə connection qurub data əlavə etmək və ya data üzərində əməliyyat həyata keçirmək üçün istifadə olunan texnologiyalardan biri də Ado Net texnologiyasıdır. Ado Net texonologiyasından istifadə etmək üçün ilk olaraq Database ilə Connection qurmaq lazımdır. Bunun üçün bizə string type ConnectionString lazım olacaq.

**SqlConnection**  
Mssql üçün bu ConnectionString istifadə edilir.

string connectionString = @”Server=serverName; Database=databaseName;Trusted\_Connection=True;”;

Daha sonra Connectionu qurmaq üçün Disposable olan System.Data.SqlClient.SqlConnection type object yaratmaq lazımdır. Open() method’u ilə SqlConnection object’i vasitəsi ilə connection açılmalı və bu connection ilə iş bitdikdən sonra Close() olunmalıdır. Həmçinin Disposable type olduğu üçün sonda Dispose() method’u işə salınmalıdır. Manual olaraq Dispose() və ya Close() method’ları işə salmamaq üçün Using keyword’ü köməyimizə çatır.

using SqlConnection connection3 = new SqlConnection(connectionString);

connection3.Open();

**SqlCommand**

SqlCommand obyekti vasitəsi ilə də biz yazacağımız query’ləri icra edə bilərik. SqlCommand obyekti string type query və SqlConnection type connection qəbul edəcək.

string query = “”;

SqlCommand command = new Command(query,connection);

**ExecuteNonQuery();**

Create,Update və Delete əməliyyatlarını həyata keçirmək üçün command obyektinə ExecuteNonQuery() method’u tətbiq edilir və geriyə dəyişiklik edilən data sətrlərinin sayı olan int type rows qayıdır.

**ExecuteReader();**

Ado Net vasitəsi ilə Select əməliyyatları ExecuReader(); method’u vasitəsi ilə həyata keçirilir. Geriyə SqlDataReader type object qaytarır. Gələn dataları istifadə edə bilmək üçün reader object istifadə edilir. while loop vasitəsi ilə geriyə data qayıdıbsa əgər indexer istifadə edərək gələn sətrdən dataları object type olaraq əldə edirik.

while(reader.HasRows){

int id = (int) reader[0]; və ya (int) reader[“id”];

}

**ExecuteScalar();**

**SQL Injection**

Ado Net istifadəsinin bir çətin tərəfi isə Security hissəsinin də bizim əlimizdə olmasıdır. Əgər düzgün şəkildə query’lər yazmazsaq hər hansısa SQL Injection baş verə bilər. Bu Injection’ın qarşısını almaq üçün parametr’lərdən istifadə edəcəyik. Yəni biz hər hansısa dynamic bir dəyər daxil edib query halına salmaq istəyiriksə, string query = $"SELECT \* FROM Posts WHERE ID = {id}"; əvəzinə string query = "SELECT \* FROM Posts WHERE ID = @id"; yazmalıyıq.  
Bu halda command.Parameters.AddWithValue(parameterName,value); method’u işə salınaraq SQL Injection’ın qarşısı alınır.



Using keyword’ü vasitəsi ilə manual olaraq SqlConnection objecti üçün Close() və Dispose(), SqlCommand objecti üçün Close() və sonda DataReader objecti üçün Close və Dispose() methodlarını çalıştırmağa ehtiyyac qalmır özü bizim üçün bunları həyata keçirirş

string connectionString = “”;

SqlConnection sqlConnection = new(connectionString); və ya

SqlConnection sqlConnection = new();

sqlConnection.ConnectionString = connectionString;

string commandText = “”;

SqlCommand sqlCommand = new(commandText,sqlConnection); və ya

SqlCommand sqlCommand = new();

sqlCommand.CommandType = CommandType.Text - by default SqlCommand object’i üçün CommandType text olur.

sqlCommand.CommandText = commandText;

sqlCommand.Connection = sqlConnection

Pluralize Service