

תכנות באינטרנט

ADO.Net

גלעד מרקמן

קריית החינוך פארק המדע,

נס ציונה



קריית החינוך
השש שנתית
פארק המדע

עמיד של חדשנות



GM

Internet Programming
Courses

עבודה בשיטה המקושרת

Command.ExecuteScalar
Command.ExecuteNonQuery

• התחברות למסד הנתונים

```
SqlConnection con = new SqlConnection(connectionString);
```

• בניית פקודת SQL

```
SqlCommand cmd = new SqlCommand("SQL String", con);
```

• ביצוע הפקודה (שאילתה)

- con.Open();
- object scalar = cmd.ExecuteScalar(); // מחזיר שדה אחד מהרשומה הראשונה
- int n = cmd.ExecuteNonQuery(); // מחזיר את מספר הרשומות שעודכנו
- con.Close();

- בניגוד לעבודה בשיטה המקושרת בשיטה הלא מקושרת חובה לפתוח ולסגור את החיבור:

- `connection.Open()`
- `connection.Close()`

דוגמה

טעינת ערך בודד

Login

Command.ExecuteScalar

[main](#)[Register](#)[Users](#)[Login](#)[More](#)

Using ExecuteScalar

user name: Password:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack)
    {
        string SQL = $"SELECT admin FROM tblUsers " +
            $"WHERE userName='{Request.Form["userName']}' AND password = '{Request.Form["password']}'";

        object scalar = GetScalar(SQL); // מחזיר רק את הערך של Admin

        if (scalar != null)
        {
            Session["userName"] = Request.Form["userName"];
            Session["Login"] = true;
            Session["Admin"] = (bool)scalar;
            message.InnerHtml = "Admin: " + Session["Admin"].ToString();
        }
        else
        {
            Session["userName"] = "Visitor";
            Session["Login"] = false;
            Session["Admin"] = false;
            message.InnerHtml = "Unknown user name or password";
        }
    }
}
```

הפעולה GetScalar

1 reference

```
public object GetScalar(string SQL)
{
    // התחברות למסד הנתונים
    string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=
    SqlConnection con = new SqlConnection(connectionString);

    // בניית פקודת SQL
    SqlCommand cmd = new SqlCommand(SQL, con);

    // ביצוע השאילתא
    con.Open();
    object scalar = cmd.ExecuteScalar();
    con.Close();

    return scalar;
}
```


command.ExecuteScalar()

- הפעולה מבצעת שאילתת SQL
- מחזירה ערך של שדה אחד הנמצא בעמודה הראשונה של השורה הראשונה.
- הערך המוחזר הוא Object שיכול להכיל את כל סוגי הערכים בטבלה.
- כדי לעשות שימוש בערך המוחזר יש לבצע המרה Casting

דוגמה

הוספת רשומה

Register

Command.ExecuteNonQuery

[Users](#)[Login](#)[More](#)

Register - Using ExecuteNonQuery AND ExecuteScalar

User Name:	<input type="text"/>
Password	<input type="password"/>
Confirm Password	<input type="password"/>
first Name:	<input type="text"/>
Last Name:	<input type="text"/>
תאריך לידה	<input type="text" value="mm/dd/yyyy"/>
City:	<input type="text"/>
E-Mail	<input type="text"/>
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
<input type="button" value="Submit"/>	<input type="button" value="Reset"/>

בדיקה אם שם משתמש קיים והוספת רשומה



```
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack)
    {
        // בדיקה אם קיים שם משתמש
        string SQL = $"SELECT COUNT (admin) FROM tblUsers " +
            $"WHERE userName='{Request.Form["userName']}' AND password = '{Request.Form["password"]}''";

        int count = (int) GetScalar(SQL);
        if (count > 0)
        {
            userNameAlert.InnerHtml = "User Name is taken";
        }
        else
        {
            // בניית השורה להוספה
            SQL = $"INSERT INTO tblUsers (firstName, lastName, userName, password, birthday, city) " +
                $"VALUES ('{Request.Form["firstName"]}', '{Request.Form["lastName"]}', " +
                $"'{Request.Form["userName"]}', '{Request.Form["password"]}', " +
                $"'{Request.Form["date"]}', '{Request.Form["city"]}''";

            ExecuteNonQuery(SQL);
            Response.Redirect("http://localhost:59467/Pages/Main.aspx");
        }
    }
}
```

GetScalar + Select Count



```
public object GetScalar(string SQL)
{
    // התחברות למסד הנתונים
    string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\
SqlConnection con = new SqlConnection(connectionString);

    // בניית פקודת SQL
    SqlCommand cmd = new SqlCommand(SQL, con);

    // ביצוע השאילתא
    con.Open();
    object scalar = cmd.ExecuteScalar();
    con.Close();

    return scalar;
}
```

```
string SQL = $"SELECT COUNT (admin) FROM tblUsers " +
    $"WHERE userName='{Request.Form["userName"]}' AND password = '{Request.Form["password"]}';
```

הוספת רשומה



```
SQL = $"INSERT INTO tblUsers (firstName, lastName, userName, password, birthday, city) " +  
    $"VALUES ('{Request.Form["firstName"]}', '{Request.Form["lastName"]}', " +  
    $"'{Request.Form["userName"]}', '{Request.Form["password"]}', " +  
    $"'{Request.Form["date"]}', '{Request.Form["city"]}');";  
ExecuteNonQuery(SQL);
```

```
public int ExecuteNonQuery(string SQL)  
{  
    // התחברות למסד הנתונים  
    string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\  
    SqlConnection con = new SqlConnection(connectionString);  
  
    // בניית פקודת SQL  
    SqlCommand cmd = new SqlCommand(SQL, con);  
  
    // ביצוע השאילתא  
    con.Open();  
    int n = cmd.ExecuteNonQuery();  
    con.Close();  
  
    return n;  
}
```

ExecuteNonQuery()

- פעולה זו מבצעת שאילתא של SQL שאינה כרוכה בקבלת נתונים.
- הפקודה יכולה לבצע כל פקודת SQL, למעט SELECT.
- השיטה מחזירה את מספר השורות ששוננו בבסיס הנתונים.

עבודה בשיטה המקושרת עם DataReader

• התחברות למסד הנתונים

```
SqlConnection con = new SqlConnection(connectionString);
```

• בניית פקודת SQL - SELECT

```
SqlCommand cmd = new SqlCommand("SQL String", con);
```

• ביצוע הפקודה (שאלתה)

- con.Open();
- SqlDataReader reader = cmd.ExecuteReader();
 - עיבוד הנתונים שהתקבלו //
- reader.Close();
- con.Close();

- `SqlDataReader reader = cmd.ExecuteReader();`
 - `reader.HasRows` // אמת אם הוחזרו שורות
 - `reader.Read();` // קרא את השורה הבאה מבסיס הנתונים. גם לפני קריאה ראשונה
 - `Reader.GetValue(0);` // מחזיר אובייקט של העמודה הראשונה – מחייב המרה
 - `reader.GetString(1);` // מחזיר מחרוזת בעמודה 1 – יש לוודא שסוג הנתונים תואם
 - `reader.GetBoolean(5);` // מחזיר ערך בוליאני
 - ...
- `reader.Close();`
- `con.Close();`

- חובה לפתוח ולסגור את החיבור:
 - `connection.Open()`
 - `connection.Close()`
- יש לסגור בסיום גם את ה-`Reader`.
 - `Reader.Close()`
- `DataReader`
 - הנתונים בתוך ה `DataReader` נשמרים כל עוד החיבור פתוח.
 - הקידום של הרשומות ב `DataReader` הוא קדימה בלבד. לא ניתן לחזור לרשומה אחורה.

דוגמה

שליפת רשומה

Login

DataReader

[main](#)[Register](#)[Users](#)[Login](#)[More](#)

Using ExecuteReader AND SqlDataReader

user name: Password:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack)
    {
        // התחברות למסד הנתונים
        string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gilad\source\
SqlConnection con = new SqlConnection(connectionString);

        // בניית פקודת SQL
        string SQL = $"SELECT firstName, admin FROM tblUsers " +
            $"WHERE userName='{Request.Form["userName"]}' AND password = '{Request.Form["password"]}';
        SqlCommand cmd = new SqlCommand(SQL, con);

        // ביצוע השאילתא
        con.Open();
        SqlDataReader reader = cmd.ExecuteReader();

        // שימוש בנתונים שהתקבלו
```

עיבוד נתונים



```
// שימוש בנתונים שהתקבלו
if (reader.HasRows)
{
    reader.Read();
    Session["userName"] = reader.GetString(0);
    Session["Login"] = true;
    Session["Admin"] = reader.GetBoolean(1);
    message.InnerHtml = "Admin: " + Session["Admin"].ToString();
    //Response.Redirect("/Pages/Main.aspx");
}
else
{
    Session["userName"] = "Visitor";
    Session["Login"] = false;
    Session["Admin"] = false;
    message.InnerHtml = "Unknown user name or password";
}
reader.Close();
con.Close();
}
```

- ה- DataReader הוא למעשה אובייקט מסוג Iterator.
- אובייקט זה מאפשר לנו לעבור על כל השורות שנשלפו בשאילתה בלולאה:

```
while (reader.HasRows)
{
    reader.Read();
    user.userName = reader.GetValue(0).ToString();
    user.Admin = (bool)reader.GetValue(1);
    object obj = reader.GetValue(2);
    // Do something with Data
}
```




סיכום תכנות בסיס נתונים

• התחברות למסד הנתונים

```
SqlConnection con = new SqlConnection(connectionString);
```

• בניית מחרוזת על שאילתה של SQL

```
String sql = "SELECT * FROM users";
```

• בניית Command:

```
SqlCommand Command = new SqlCommand("SQL String", con);
```

• בניית DataAdapter

```
SqlDataAdapter adapter = new SqlDataAdapter(Command);
```

• בניית DataSet לאיכסון נתונים מקומי

```
DataSet ds = new DataSet();
```

• טעינת הנתונים ממסד הנתונים ל DataSet

```
adapter.Fill(ds, "users");
```

• עדכון בסיס נתונים בשיטה הלא מקושרת באמצעות:

```
SqlCommandBuilder builder = new SqlCommandBuilder(adapter);
```

```
adapter.UpdateCommand= builder.GetDeleteCommand();
```

```
adapter.UpdateCommand = builder.GetUpdateCommand();
```

```
adapter.UpdateCommand = builder.GetInsertCommand();
```

```
adapter.Update(ds, "users");
```

• הפקודות בשיטה המקושרת:

- `Command.ExecuteScalar`
- `Command.ExecuteNonQuery`
- `Command.ExecuteReader`

• שימוש באובייקט `DataReader`

- `reader.HasRows;`
- `reader.Read();`
- `Reader.GetValue(0);`
- `reader.GetString(1);`
- `reader.GetBoolean(5);`

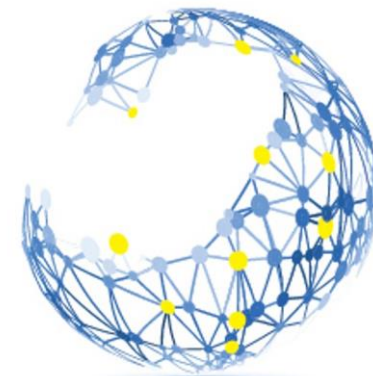
פקודה	הסבר
<code>DataRow dr = Table.NewRow();</code>	יצירת שורה על פי סכימה של הטבלה
<code>DataRow dr = DataSet.Tables["tblUsers"].NewRow();</code>	כנ"ל – על פי טבלה ב DataSet
<code>dr["columnName"] = Value;</code>	הכנסת ערכים לשורה
<code>Table.Rows.Add(dr);</code>	עדכון שורה בטבלה
<code>DataRow [] dr = ds.Table["tblUsers"].Select("Condition");</code>	יצירת מערך של מצביעים לשורות בטבלה על פי פקודת Select
<code>Dr[0].Delete();</code>	מחיקת שורה מהמערך וגם מהטבלה

- עליכם להכין אתר אינטרנט הכולל גישה לבסיס נתונים.
- אפיון האתר מצורף במסמך "מטלה מסכמת".
- תיעוד הפרוייקט: יש לצרף בנוסף קובץ וורד שיכלול שם האתר ודם המגיש בנוסף לגבי כל אחד מהדפים שלכם:
 - צילום מסך של הרצת הדף.
 - קוד html.
 - קוד CSS.
 - קוד גאווה סקריפט.
 - קוד צד שרת.

• איפיון פרויקט מסכם

סיימנו
תודה רבה
ומקווה שהיה לכם
לעזר

גלעד מרקמן



קריית החינוך
השש שנתית
פארק המדע
עתיד של חדשנות

