

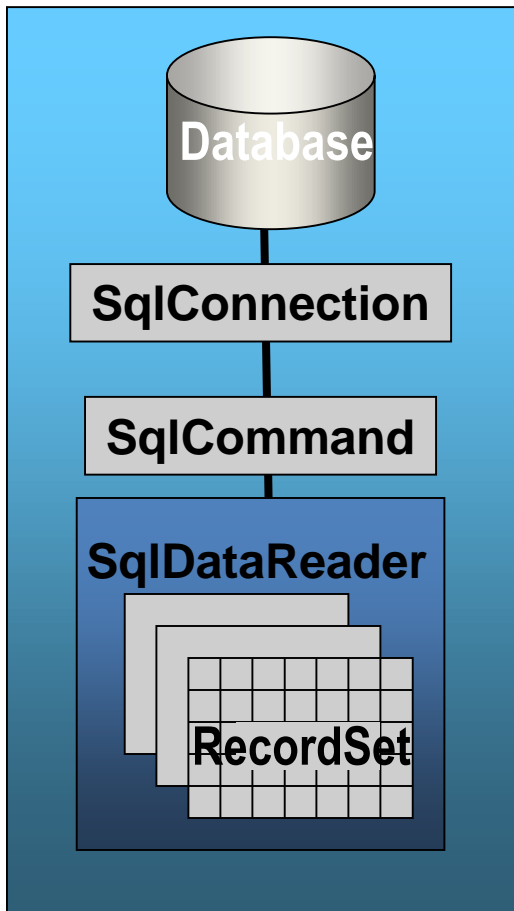
מבוא לתכנות II

הרצאה 10 – II ADO - DataSet ,GridView ,

פרוצדורות

סמסטר 2

סקירה



• **SqlConnection** – מתחבר למסד הנתונים.

• **SqlDataAdapter** – מפעיל פקודות sql.

• **DataSet** – מסד נתונים בזיכרון מטמון.

מחלקת DataSet

- מחלקת *DataSet* של .NET. הינה סט של נתונים, מנותקים מכל מקור מידע כלשהו.
- ניתן לראות את DataSet בתור ערכת רשומות מנותקת אשר לא יודעת כלום לגבי מקור המידע שלה או היעד אליו המידע מגיע.

מחלקת DataSet

- DataSet עובד עם כל צורות אחסון הנתונים:
– רגיל, יחסי והירארכי (מכיוון שאין לו שום ידע לגבי מקור המידע)
- כדי להשתמש ב DataSets יש לייבא את ספריית
System.Data

מחלקת DataSet

- יוצר בזיכרון מטמון נתונים אשר נלקחים ממקור מידע.
- דרך נפוצה לייצג ולשלוט בנתונים
 - מיכל נתונים אוניברסלי.
 - לא מיועד רק למסדי נתונים.
- נועד להיות מנותק ממקור המידע
 - התחבר, בצע שאילתה, התנתק.
- יכול להשתמש בXML
 - כדי לקרוא ולכתוב מידע בפורמט XML
 - כדי לקרוא ולכתוב XMLSchema.

DataAdapter

- **DataAdapter** הוא האובייקט אשר מתחבר למקור המידע כדי למלא את DataSet במידע.
- ולכן הוא מקושר ספציפית למקור המידע המקורי:
 - SqlDataAdapter, OleDbDataAdapter.
- אובייקט **DataAdapter** מתפקד בתור גשר בין DataSet למקור המידע.
- שימוש כממלא הנתונים של DataSets (**Fill**) ומעדכן (**Update**) מסדי נתונים.



מילוי DataSet ועדכון

- המחזוריות:

- `DataAdapter.Fill(DataSet,"TABLENAME")`

- המשתמש משנה את המידע בDataSet

- `DataAdapter.Update(DataSet, "TABLENAME")`

- כאשר אנו מעדכנים (באמצעות `Update()`) הDataAdapter לוקח (רק) את השורות שבהן המידע שונה:

- אם נוספו שורות, עושה פעולות הכנסה

- אם שונו שורות, עושה פעולות עדכון

- אם נמחקו שורות, עושה פעולות מחיקה

```
DataSet ds = new DataSet();  
SqlDataAdapter adptr = new SqlDataAdapter(command, con);  
adptr.Fill(ds,"t1"); //opens and closes the DB!
```

- זה ימלא את הDataSet עם השאילתה המבוקשת ויקרא לטבלה t1.

פקודות הDataAdapter

- ניתן להשתמש בבנאי של DataAdapter עם פקודה ואובייקט חיבור.
- או שניתן להשתמש בפקודות DataAdapter (כדי לשנות את מסד הנתונים ולא הDataSet, אבל רק על מידע שקיים).
- מכיל 4 אובייקטי פקודות
 - SelectCommand
 - InsertCommand
 - UpdateCommand
 - DeleteCommand
- **הפקודה הזו באופן אוטומטי. נדרש מפתח ראשי בטבלה** **SqlCommandBuilder(DataAdapter) - יבנה את**

עדכון

```
SqlDataAdapter adptr = new SqlDataAdapter(cmd, con);  
SqlCommandBuilder comb = new SqlCommandBuilder(adptr);  
adptr.Update(t); //opens and closes the DB!
```

- זה יעדכן את השורות הרלוונטיות במסד הנתונים, בהתאם לשינויים שנעשו ב DataSet\DataTable.

עבודה עם מסד נתונים - סיכום

- כדי לעבוד עם מסדי נתונים יש להשתמש:

- DataReader

- מחובר, לקריאה בלבד, לא שמור בזיכרון.

- או

- DataSet

- מנותק, בר שינוי, שמור בזיכרון.

- DataAdapter

- לוגיקה שמאפשרת לקחת מידע ממקור מידע, למלא DataSet, ולהחזיר שינויים למקור המידע.

- פקודת הכנסה תיראה כך:

```
sql="Insert into Authors (au_id,au_lname,au_fname,phone,  
address,city,state,zip,contract) Values (@Au_id, @Au_lname,  
@Au_fname, @Phone, @Address, @City, @State, @Zip,  
@Contract) “
```

- והפרמטרים יוספו לפקודה:

```
cmd.Parameters.Add(New SqlParameter("@Au_lname",  
frmLname.text)) ;
```

```
cmd.Parameters.Add(New SqlParameter("@Au_fname",  
frmFname.text)) ;
```

```
cmd.Parameters.Add(New SqlParameter("@Address",  
frmAddress.text)) ;
```

DataGridView

- פקד ***DataGridView*** הוא הפקד המורכב והחזק ביותר להצגת ושינוי מידע.
- יש לו כמעט 100 תכונות, מתודות ואירועים.
- ה***DataGridView*** מציג טבלת נתונים (אשר נלקחו ממקור מידע, כמו מ***SQL***)

מקור מידע

- כדי לשייך מקור מידע לDataGridView, נצטרך לבצע את הפעולות הבאות:

```
MyDataGrid.DataSource=ds.Tables["Authors"];
```

- תחביר חלופי יהיה להגדיר גם מקור מידע וגם DataMember מהDataSet :

```
MyDataGrid.DataSource=ds;  
MyDataGrid.DataMember="Authors";
```

דוגמה

Form1

☒ sort by ID
☐ sort by Name
☐ sort by Family

Show Table

Select By Param

Update DB

Edit Name to Capital

Column1	ID	Name	Family
<input type="checkbox"/>	3	er3	er3
<input type="checkbox"/>	4	a444	44
<input type="checkbox"/>	5	a557	555
<input type="checkbox"/>	7	NAME7	777
<input type="checkbox"/>	8	888	88
<input type="checkbox"/>	9	9977777777	99
<input type="checkbox"/>	11	WHAT	thef
<input type="checkbox"/>	12	NIR7	chen7
<input type="checkbox"/>	13	QWE	qwe
<input type="checkbox"/>	15	BAR	refaely
<input type="checkbox"/>	17	n17	f17

- CREATE PROCEDURE ProcName @ParameterName
ParameterType [= default] [OUTPUT],... AS

SQL Statement

- @@ERROR - השגיאה שsql זורק.

- CREATE PROCEDURE SearchUser @MyID int ,
@FamilyName varchar(20) output AS

Select @FamilyName= Family

From tbUser

where ID=@MyID

return @@ERROR

GO

דוגמה לפרוצדורה 2 – מחזיר טבלה שלמה

```
CREATE PROCEDURE [dbo].[SearchUserTable] @MyID  
int AS  
  
    Select *  
  
    From tbUser  
  
    where ID=@MyID
```

קריאה לפרוצדורות

- `myCon.Open()`
- `SqlCommand MySPCommand = new SqlCommand("SearchUser", myCon)`
- `MySPCommand.CommandType = CommandType.StoredProcedure`
- `SqlParameter parName = new SqlParameter("@MyPar1", SqlDbType, [Size])`
- `parName.Value = Value (for input parameter)`
- `parName.Direction = ParameterDirection.Input/output/ReturnValue`

קריאה לפרוצדורות

MySPCommand.Parameters.Add(parName)

MySPCommand.ExecuteNonQuery()

myCon.Close()

parName.Value...

דוגמה

The screenshot shows a Windows application titled "DataReader". The interface includes several buttons: "LOAD", "INSERT", "DELETE", "UPDATE", and "UPDATE USING PARAMETER". Below these buttons are three input fields labeled "ID", "NAME", and "FAMILY". The "ID" field contains the value "7". Below the input fields is a button labeled "UPDATE DS TO DB". Further down is a text input field labeled "STORED PROCEDURE" and a button labeled "SEARCH".

A modal dialog box is displayed in the foreground, indicating a successful operation. The dialog contains the text "Succeeded! the family name is : f7" and an "OK" button.

At the bottom of the application window is a table with the following data:

ID	Name	Family
3	er3	er
4	444	44
5	55	55
7	n7	f7
8	n8	f8
9	ddd	ddd