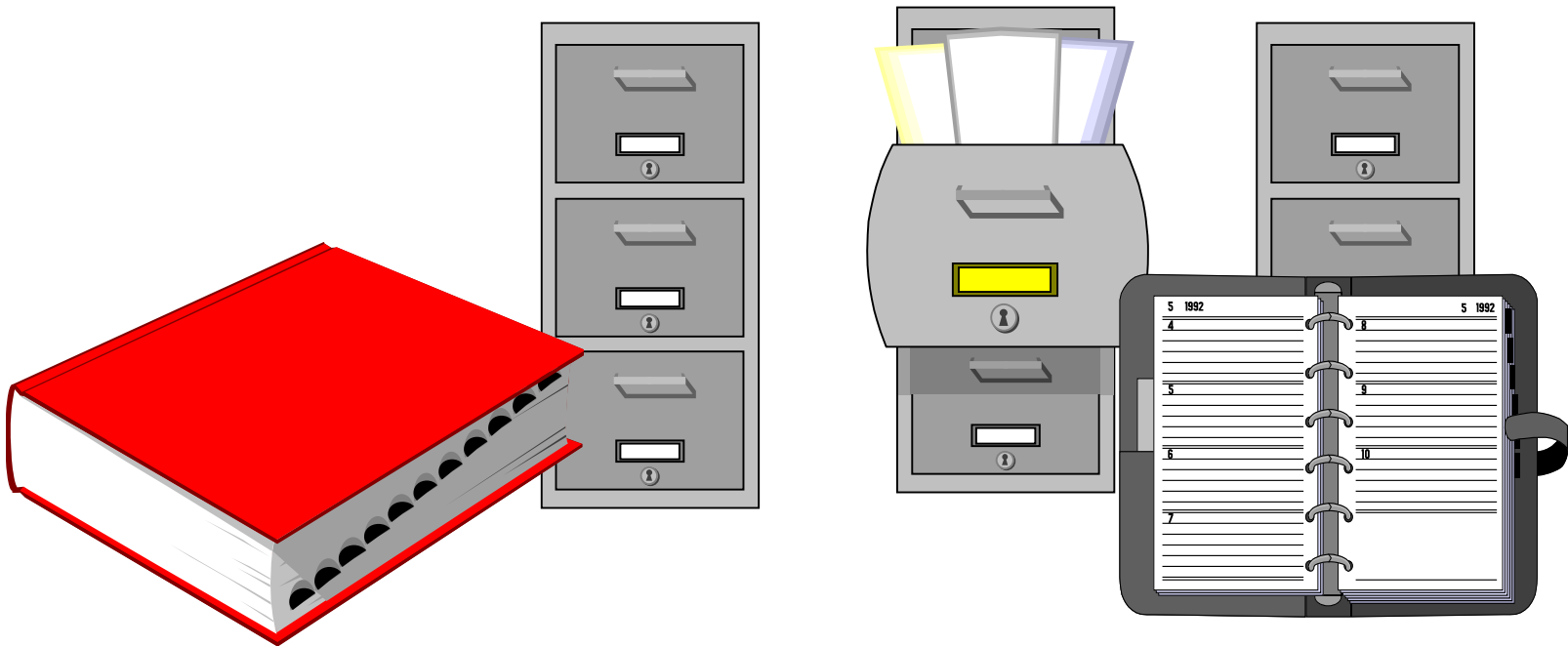


Introduction to Programming

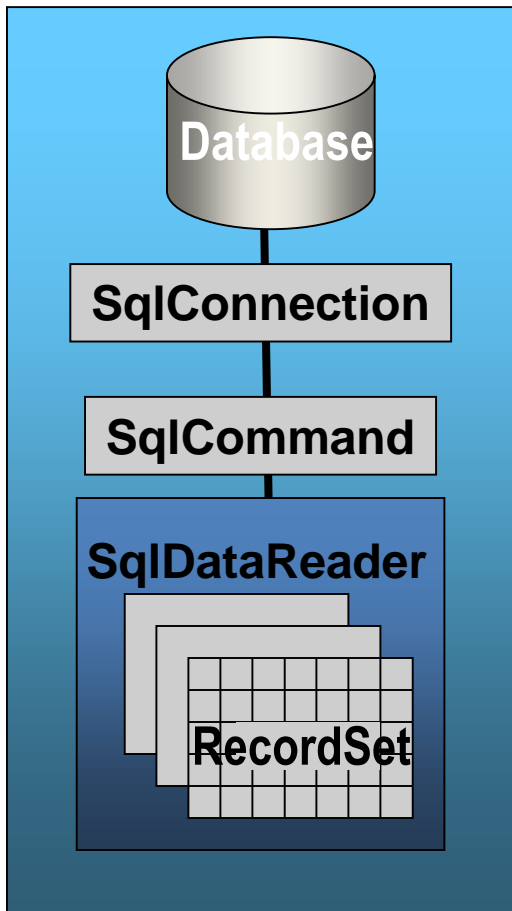
Lecture 9 – ADO I - DataReader

Semester II

ADO.NET



OVERVIEW



- **SqlConnection** - connects to the DB
- **SqlCommand** - executes sql commands
- **SqlDataReader** - reads records from the SqlCommand

SqlConnection(C#)

```
System.Data.SqlClient.SqlConnection sqlConnection1;  
sqlConnection1 = new  
    System.Data.SqlClient.SqlConnection(@"Data Source=HOME-  
PC\SQLEXPRESS;Initial Catalog=asp;Integrated Security=True"  
);
```

- Data Source = 'ServerName'
- Initial Catalog= 'DatabaseName'

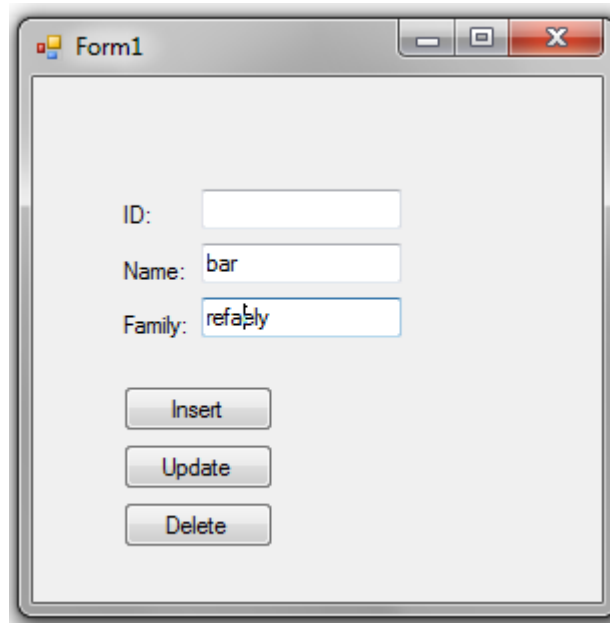
SqlCommand(C#)

```
System.Data.SqlClient.SqlCommand sqlCommand1;  
sqlCommand1 = new SqlCommand("SELECT * FROM tbUser ORDER BY  
ID",sqlConnection1);  
sqlCommand1.Connection.Open();  
sqlCommand1.ExecuteNonQuery(); //(returns the number of rows  
affected)  
sqlCommand1.Connection.Close();
```

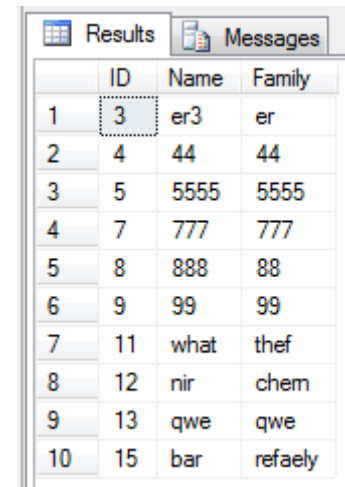
- For none query: Insert, Update and Delete

Example : NoneQuery

- Implement the example together in class to learn the DB connection and the separation of the DB layer from the UI layer



A screenshot of a Windows application window titled "Form1". The window contains three text input fields labeled "ID:", "Name:", and "Family:". The "Name:" field contains the text "bar" and the "Family:" field contains the text "refaely". Below the input fields are three buttons labeled "Insert", "Update", and "Delete".



A screenshot of a database results window. It has two tabs: "Results" and "Messages". The "Results" tab is active, showing a table with three columns: "ID", "Name", and "Family". The table contains 10 rows of data. The first row has ID 1, Name 3, and Family er3. The second row has ID 2, Name 44, and Family 44. The third row has ID 3, Name 5555, and Family 5555. The fourth row has ID 4, Name 777, and Family 777. The fifth row has ID 5, Name 888, and Family 88. The sixth row has ID 6, Name 99, and Family 99. The seventh row has ID 7, Name what, and Family thef. The eighth row has ID 8, Name nir, and Family chem. The ninth row has ID 9, Name qwe, and Family qwe. The tenth row has ID 10, Name bar, and Family refaely.

	ID	Name	Family
1	3	er3	er
2	4	44	44
3	5	5555	5555
4	7	777	777
5	8	888	88
6	9	99	99
7	11	what	thef
8	12	nir	chem
9	13	qwe	qwe
10	15	bar	refaely

SqlDataReader(C#)

- All DataReaders (SqlDataReader, OleDbDataReader) are connected to the database (until the Close()).
 - Reads only one record at a time.
 - Does not remember previous record. Forward-only stream reading.
 - Can't perform sorting, filtering or counting of the resulting set.
 - A minimal memory usage.

SqlDataReader(C#)

```
System.Data.SqlClient.SqlDataReader SqlDataReader1;  
SqlDataReader1 = sqlCommand1.ExecuteReader(); //instead of  
ExecuteNonQuery()  
SqlDataReader1.Read(); // -> returns true if there are records  
to read, false otherwise
```

```
SqlDataReader1["id"]; // get the value for the Field named 'id'.  
SqlDataReader1[2]; // get the value for the Field indexed 2.  
SqlDataReader1.Close();
```


Example : Query

- Implement the example together in class to learn the DB connection and the separation of the DB layer from the UI layer

