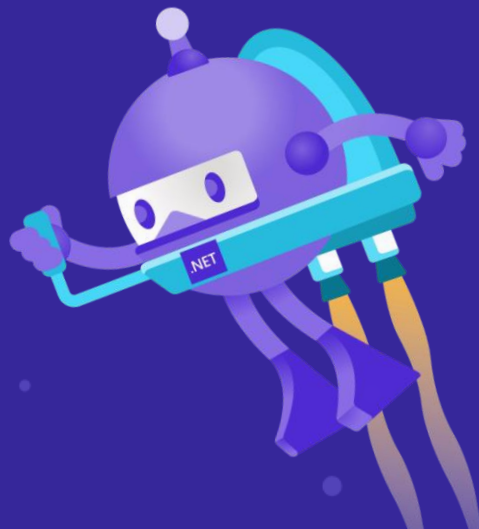
# Day 11 : Working with Data



# Working With Databases

- ADO.NET is a set of classes that expose data access services
- Provides a rich set of components for creating distributed, data-sharing applications
- Can use ADO.NET to connect to these data sources and retrieve, handle, and update the data that they contain
- Includes .NET Framework data providers for connecting to a database, executing commands, and retrieving results
- ADO.NET classes are found in System.Data.dll

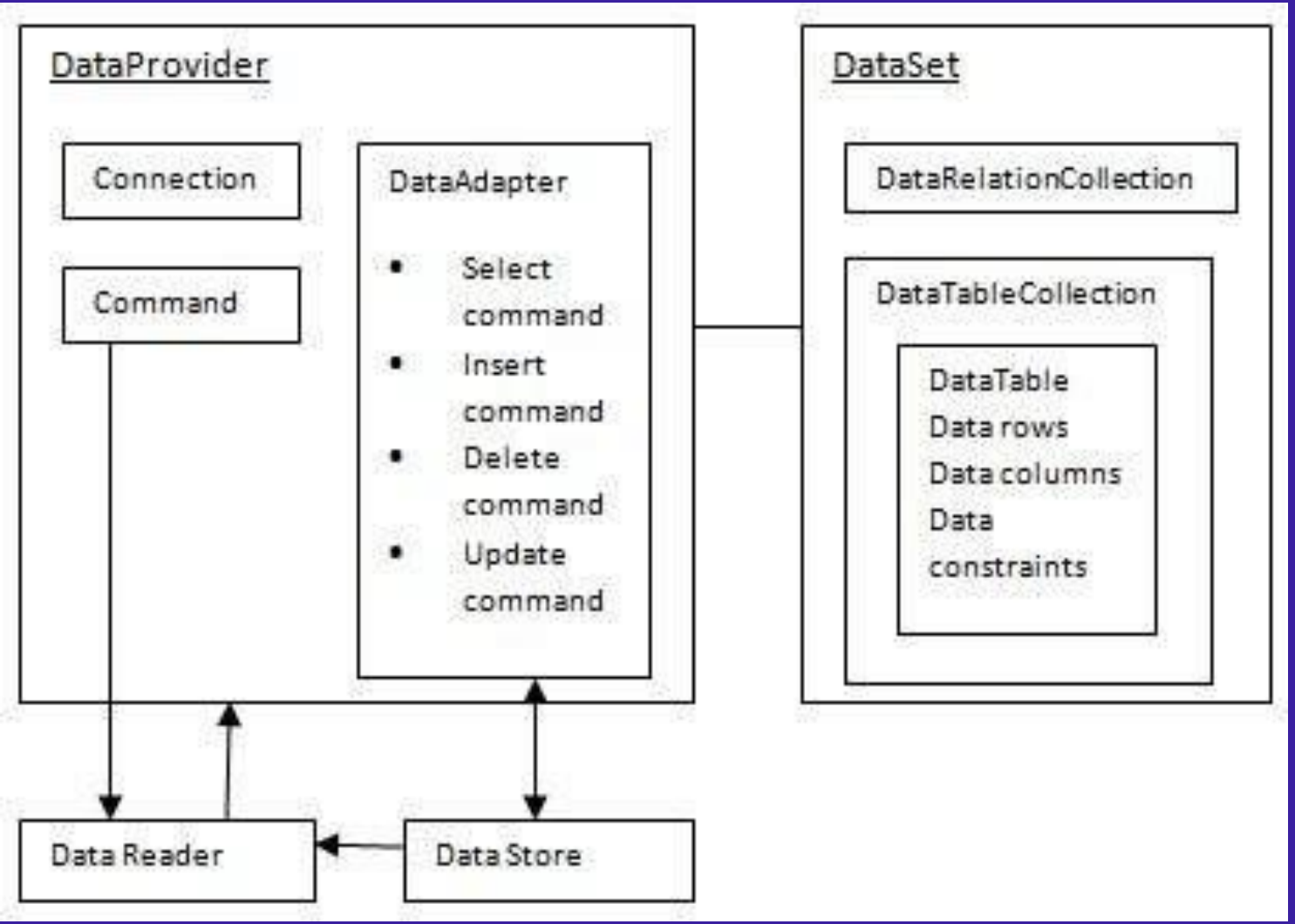
# ADO.NET

- Mainly comprised of two components
  - .NET Framework Data Providers

    are components that have been explicitly designed for data manipulation and fast, forward-only, read-only access to data
  - DataSet

    Is explicitly designed for data access independent of any data source. As a result, it can be used with multiple and differing data sources, used with XML data, or used to manage data local to the application

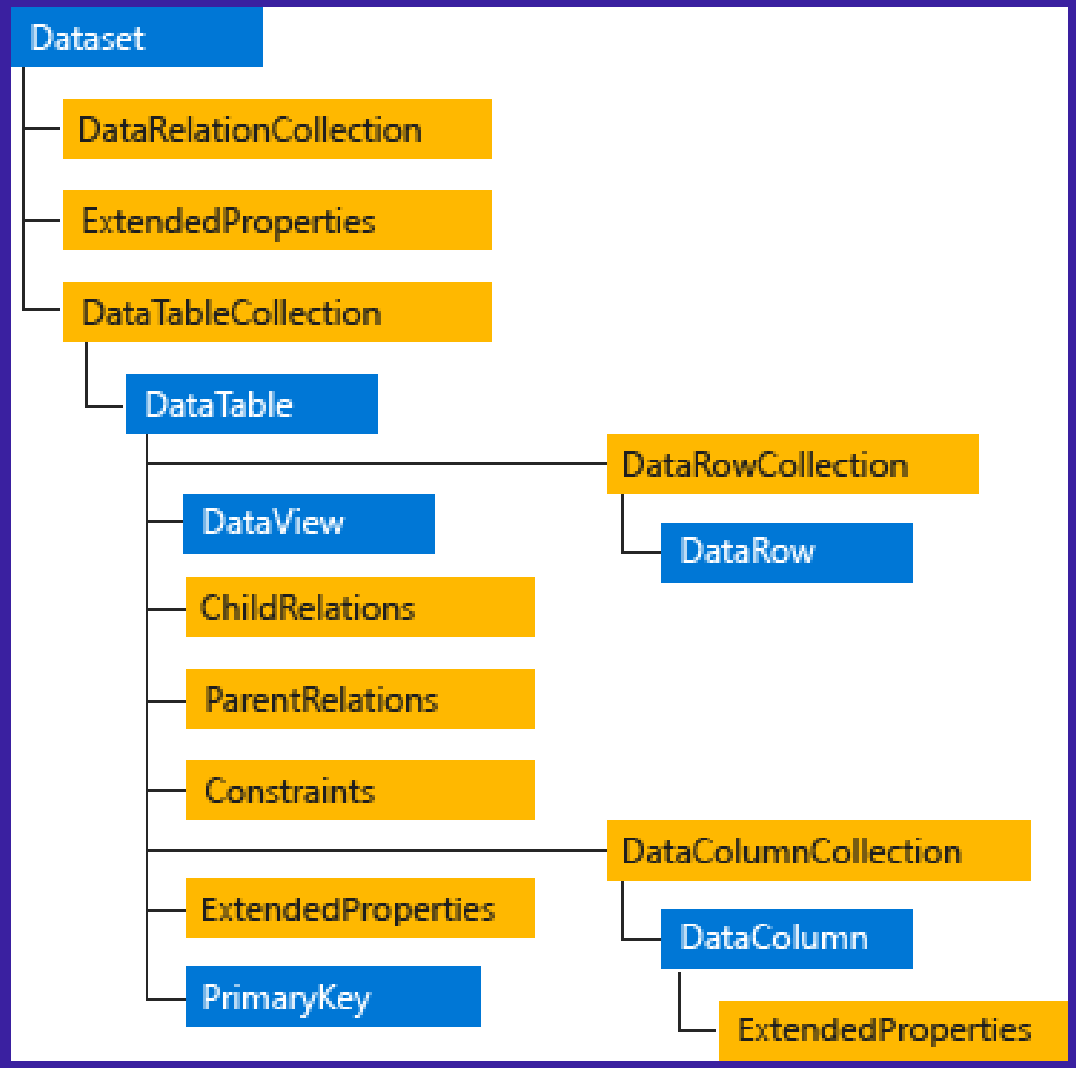
# ADO.NET


DataProvider
Connection
Command
DataAdapter
Select command
Insert command
Delete command
Update command
DataSet
DataRelationCollection
DataTableCollection
DataTable
Data rows
Data columns
Data constraints
Data Reader
Data Store

# .NET Framework Data Providers

| .NET Framework data provider | Description |
|---|---|
| .NET Framework Data Provider for SQL Server | Provides data access for Microsoft SQL Server. Uses the System.Data.SqlClient namespace. |
| .NET Framework Data Provider for OLE DB | For data sources exposed by using OLE DB. Uses the System.Data.OleDb namespace. |
| .NET Framework Data Provider for ODBC | For data sources exposed by using ODBC. Uses the System.Data.Odbc namespace. |
| .NET Framework Data Provider for Oracle | For Oracle data sources. The .NET Framework Data Provider for Oracle supports Oracle client software version 8.1.7 and later, and uses the System.Data.OracleClient namespace. |
| EntityClient Provider | Provides data access for Entity Data Model (EDM) applications. Uses the System.Data.EntityClient namespace. |
| .NET Framework Data Provider for SQL Server Compact 4.0. | Provides data access for Microsoft SQL Server Compact 4.0. Uses the System.Data.SqlServerCe namespace. |

# DataSet


Dataset
DataRelationCollection
ExtendedProperties
DataTableCollection
DataTable
DataRowCollection
DataRow
DataView
ChildRelations
ParentRelations
Constraints
DataColumnCollection
ExtendedProperties
DataColumn
PrimaryKey
ExtendedProperties

# .NET Framework Data Providers – Core Objects

| Object | Description |
| --- | --- |
| Connection | Establishes a connection to a specific data source. The base class for all Connection objects is the DbConnection class. |
| Command | Executes a command against a data source. Exposes Parameters and can execute in the scope of a Transaction from a Connection. The base class for all Command objects is the DbCommand class. |
| DataReader | Reads a forward-only, read-only stream of data from a data source. The base class for all DataReader objects is the DbDataReader class. |
| DataAdapter | Populates a DataSet and resolves updates with the data source. The base class for all DataAdapter objects is the DbDataAdapter class. |

# .NET Framework Data Provider for SQL Server

| Class | Description |
| --- | --- |
| SqlConnection | It is used to create SQL Server connection. This class cannot be inherited. |
| SqlCommand | It is used to execute database queries. This class cannot be inherited. |
| SqlDataAdapter | It represents a set of data commands and a database connection that are used to fill the DataSet. This class cannot be inherited. |
| SqlDataReader | It is used to read rows from a SQL Server database. This class cannot be inherited. |
| SqlException | This class is used to throw SQL exceptions. It throws an exception when an error is occurred. This class cannot be inherited. |

# Connecting to a SQL Database

- SqlConnection class is used to manage connection
- Needs to provide information such as DB Server name, DB name and DB user credentials

```
using (SqlConnection connection = new SqlConnection(“Server=myServerAddress;Database=myDataBase;User Id=myUsername;Password=myPassword”))
{
  connection.Open();
  Console.WriteLine("Connection Established Successfully");
  connection.Close();
}
```

# Executing Commands & Reading Data

- SqlCommand class is used to store and execute SQL statements
- SqlDataReader class is used to read data

```
SqlConnection con = new SqlConnection(“Server=myServerAddress;Database=myDataBase;User Id=myUsername;Password=myPassword”);
// writing sql query
SqlCommand cm = new SqlCommand("select * from student", con);
// Opening Connection
con.Open();
// Executing the SQL query
SqlDataReader sdr = cm.ExecuteReader();
while (sdr.Read())
{
   Console.WriteLine(sdr["name"]+" "+ sdr["email"]);
}
```

# DataSet

- Collection of data tables that contain the data
- It is an in-memory data store that can hold more than one table at the same time

```
SqlConnection con = new SqlConnection(“Server=myServerAddress;Database=myDataBase;User Id=myUsername;Password=myPassword”);

// Opening Connection
con.Open();
// Executing the SQL query
 SqlDataAdapter sde = new SqlDataAdapter("Select * from student", con);
//populating data
DataSet ds = new DataSet();
sde.Fill(ds);
```

# DataTable

- represents relational data into tabular form
- It is an in-memory data store that can hold more than one table at the same time

```
DataTable table = new DataTable();
table.Columns.Add("ID");
table.Columns.Add("Name");
table.Columns.Add("Email");
table.Rows.Add("101", “Amal",“amal@dev.com");
table.Rows.Add("102", "Sam", "sam@test.com");
table.Rows.Add("103", “Dev", “dev@test.com");
table.Rows.Add("104", "Ankur", "ankur@test.com");
```

# Thanks for joining!