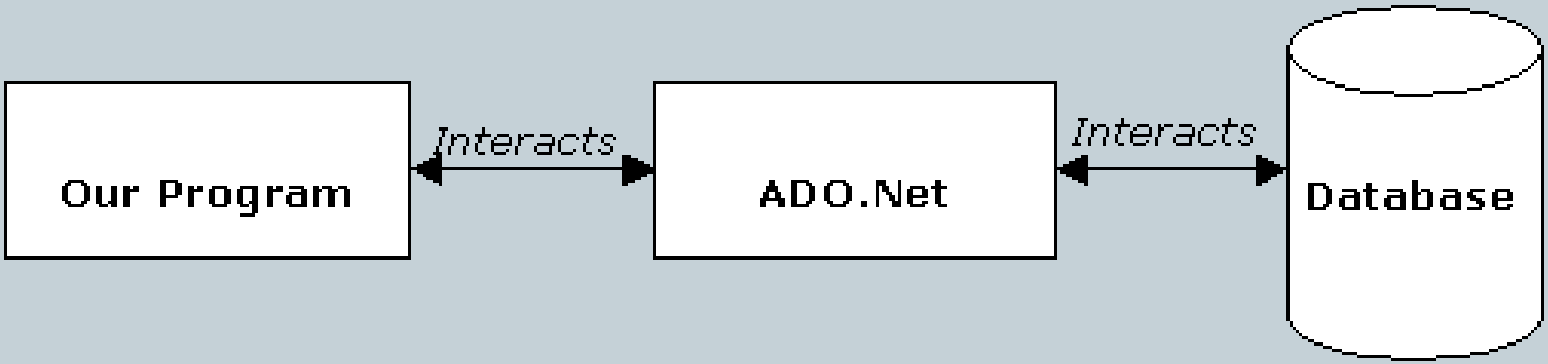
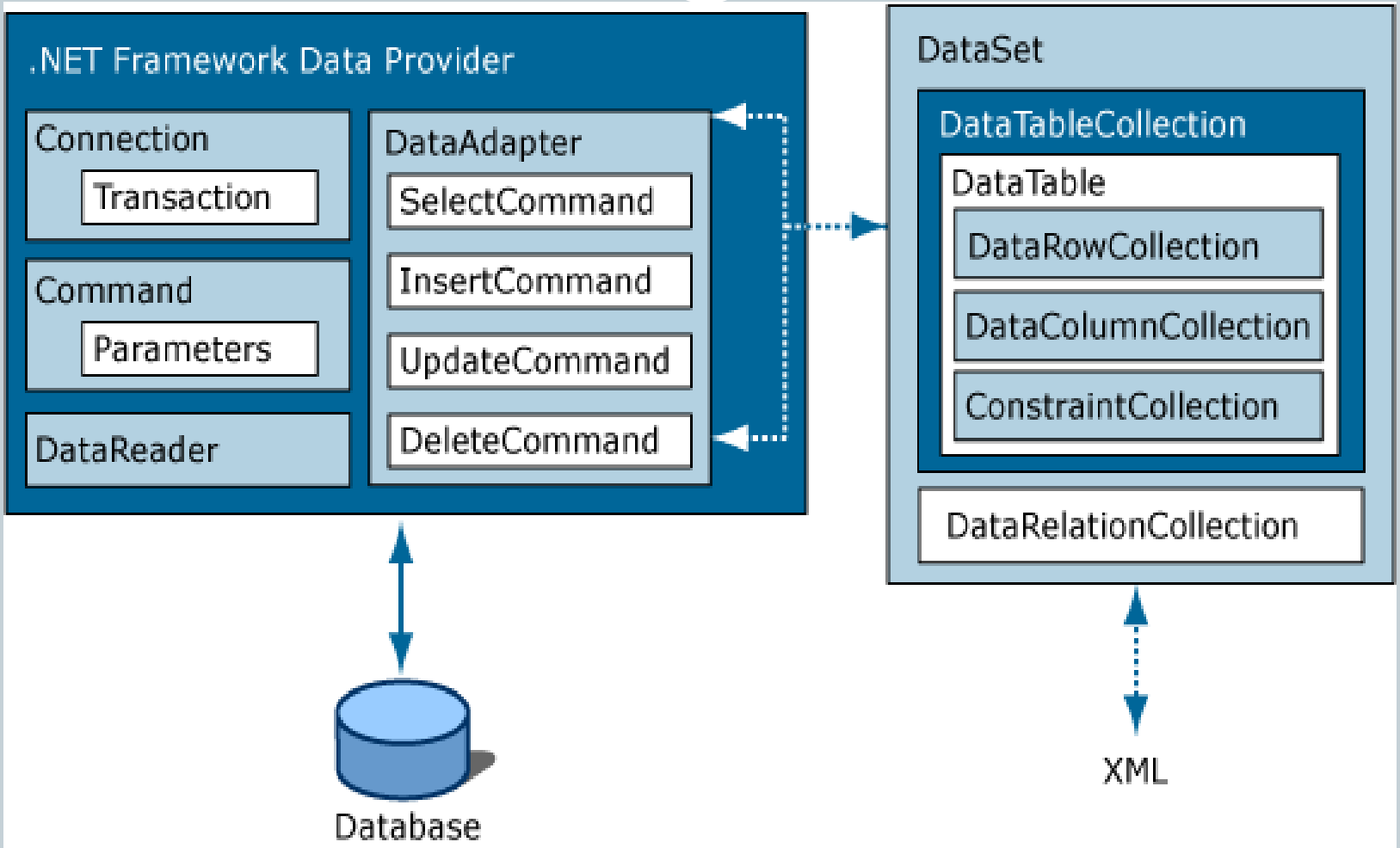
Unit -6

DATABASE PROGRAMMING WITH ADO.NET

- A data-access technology that enables applications to connect to data stores and manipulate data contained in them in various ways .
- An object oriented framework that allows you to interact with database systems



ADO.NET Architecture



ADO.NET Core Objects

A .NET Framework data provider is used for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, placed in a DataSet in order to be exposed to the user as needed.

Data Provider	Namespace
SQL Server	`System.Data.SqlClient`
OLE DB	`System.Data.OleDb`
ODBC	`System.Data.Odbc`
Oracle	`System.Data.OracleClient`

Connection Class

- You need to establish a connection class object for inserting, updating, deleting and retrieving data from a database.
- The Connection class allows you to establish a connection to the data source.
- The Connection class object needs the necessary information to discover the data source and this information is provided by a connection string.

- **Connection Strings:**
- You need to supply a connection string in the Connection class object. The connection string is a series of name/value settings separated by semicolons (;). A connection string requires a few peices of information such as the location of the database, the database name and the database authentication mechanism.

Command

- The Command Class allows performing any data definition tasks such as creating and altering tables and databases, retrieving, updating and deleting of records. The Command object used to execute SQL queries can be inline text or a Stored Procedure.

- The three important methods exposed by the SqlCommand object is shown below:
- ExecuteScalar
- ExecuteNonQuery
- ExecuteReader

- **ExecuteScalar** is useful for returning a single value from the database. For example, using this method we can retrieve a sum of sales made by a specific product, total number of records in the employee table, unique id by supplying filtering conditions and so on. Since this method performs faster we do not need to go for the Reader method just to retrieve a single scalar value.

  ExecuteNonQuery is useful for performing data manipulation on the database. Simply, the ExecuteNonQuery is for executing the DML statements. The return value of the ExecuteNonQuery is an integral value that represents the number of rows affected by the Operation.

  ExecuteReader is used when we need to retrieve rows and columns of data using the SQL select statements. As the data retrieved is a table of data, ExecuteReader returns SqlDataReader. We should iterate through this object to get the required values.

DataReader

- DataReader is used to read the data from database and it is a read and forward only connection oriented architecture during fetch the data from database. DataReader will fetch the data very fast when compared with dataset.
- Generally we will use ExecuteReader object to bind data to datareader.

DataSet

- DataSet is a disconnected orient architecture that means there is no need of active connections during work with datasets and it is a collection of DataTables and relations between tables. It is used to hold multiple tables with data. You can select data form tables,

DataAdapter

- DataAdapter will acts as a Bridge between DataSet and database. This dataadapter object is used to read the data from database and bind that data to dataset. Dataadapter is a disconnected oriented architecture.

DataTable

- DataTable represents a single table in the database. It has rows and columns. There is no much difference between dataset and datatable, dataset is simply the collection of datatables.

Differences

DataReader	Dataset
Used in a connected architecture	Used in a disconnected architecture.
Provides better performance	Provides lower performance.
DataReader object has read-only access	A DataSet object has read/write access
DataReader object supports a single table based on a single SQL query of one database	A DataSet object supports multiple tables from various databases.
A DataReader object has faster access to data.	A DataSet object has slower access to data.
We can't create a relation in a data reader.	We can create relations in a dataset.
Whereas a DataReader doesn't support data reader communicates with the command object.	A Dataset supports integration with XML Dataset communicates with the Data Adapter only.
DataReader cannot modify data.	DataSet can modify data.