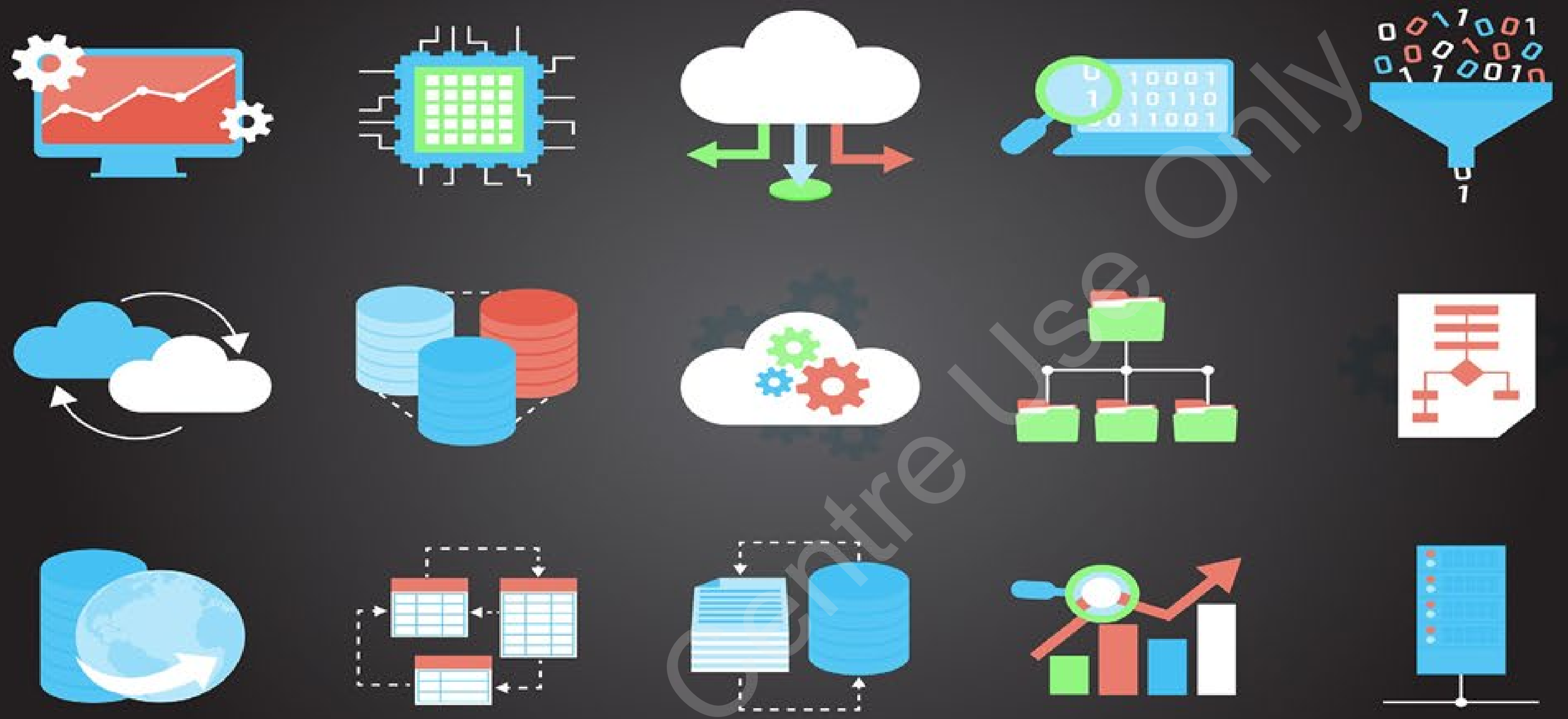
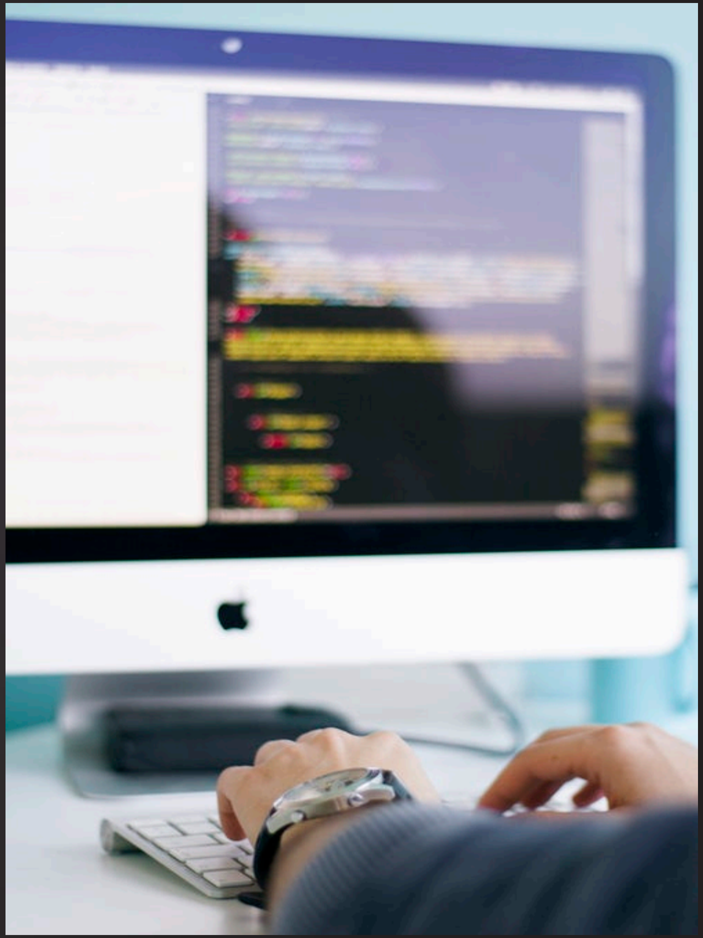


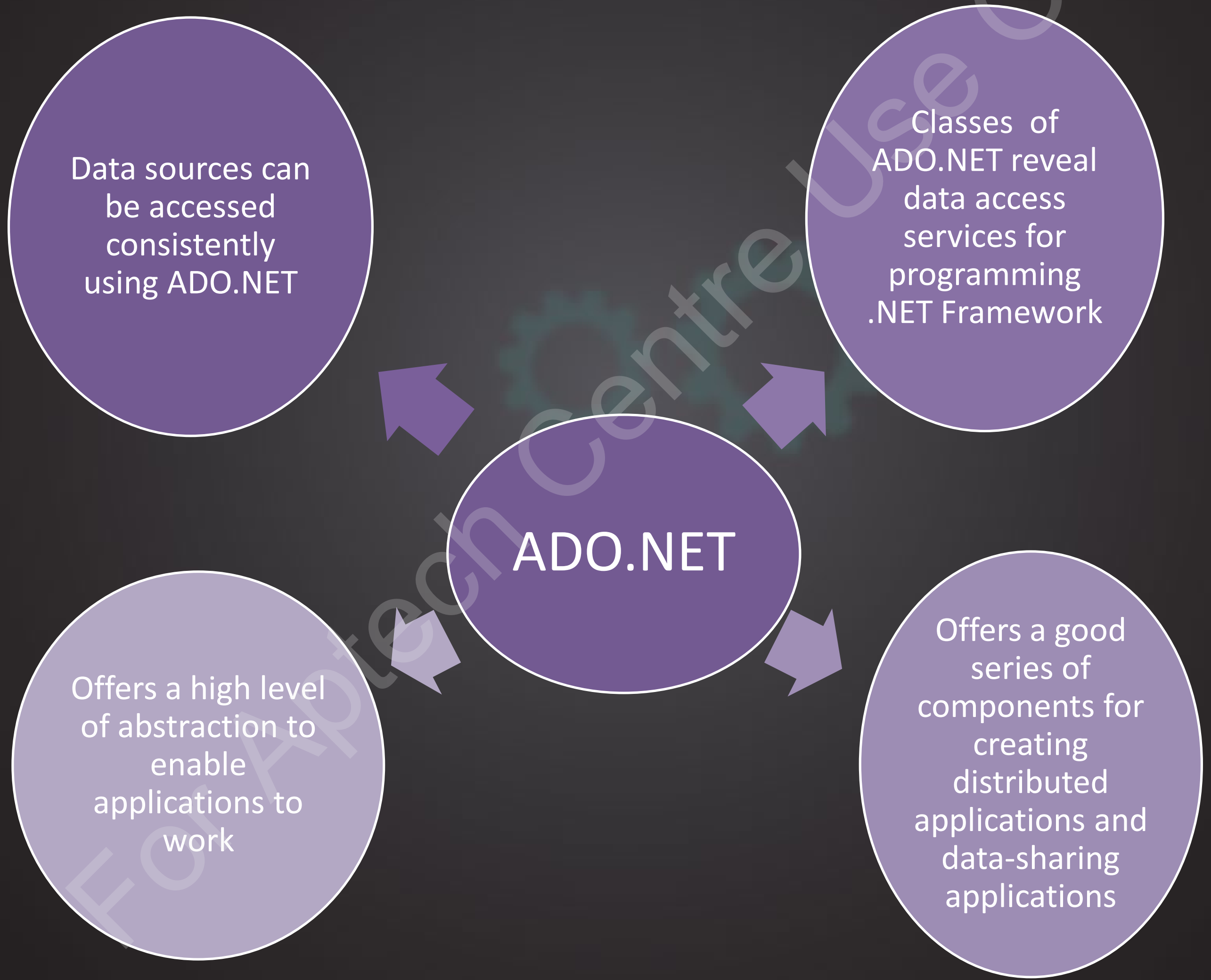
Session 04:
Data Access Using ADO.NET

Objectives

- Identify database connection
- Explain how to use the database command object
- Explain how to use a DataReader
- Explain how to manage database transactions



Introduction to Database Connection



ADO.NET

Data sources can be accessed consistently using ADO.NET

Classes of ADO.NET reveal data access services for programming .NET Framework

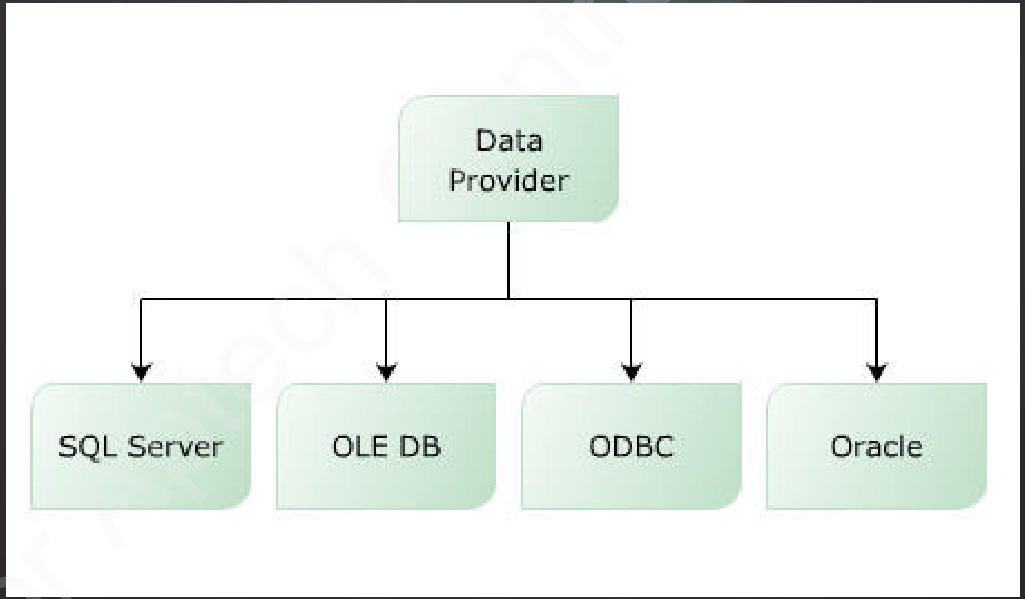
Offers a high level of abstraction to enable applications to work

Offers a good series of components for creating distributed applications and data-sharing applications

Connecting to a Data Source in ADO.NET

A `DbConnection` object exists for each .NET Framework data provider included within .NET Framework

Data Providers

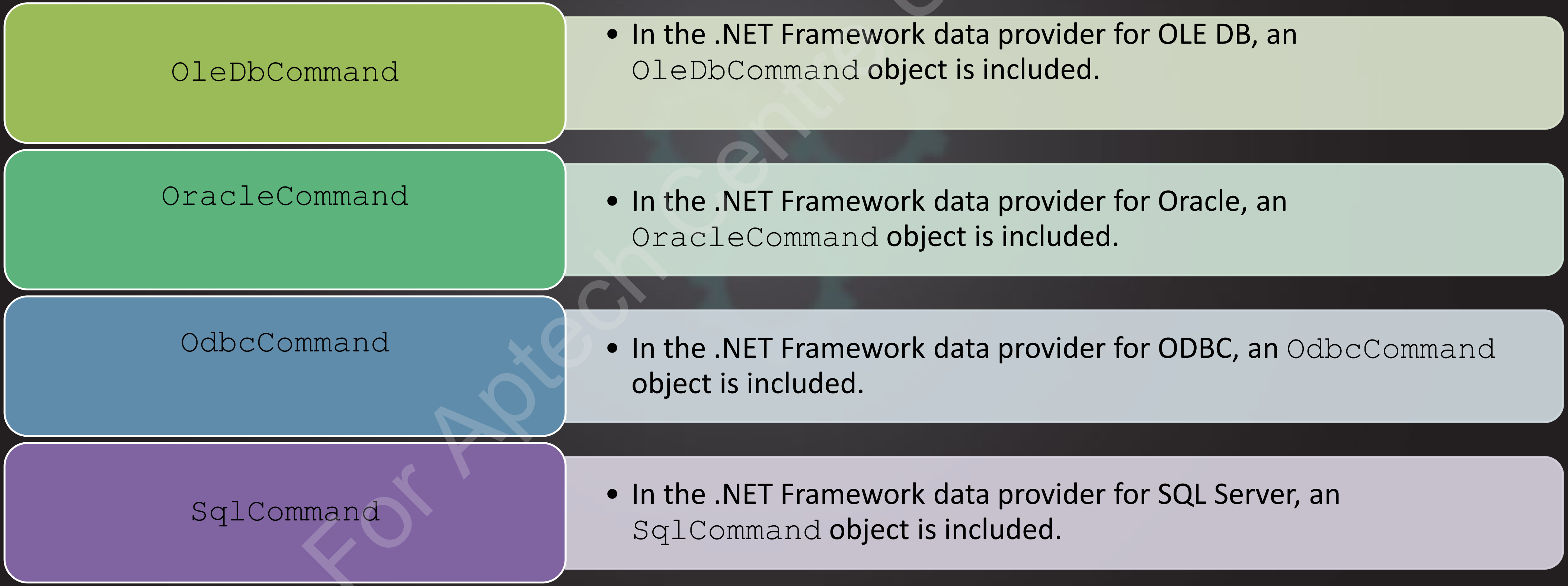


Connection Strings in ADO.NET

Initialization information passed as a parameter to a data source from a data provider is stored in a connection string

Using the Database Command object

There is a command object for each .NET Framework data provider within the .NET Framework. Some of them are listed as follows:



`OleDbCommand`

- In the .NET Framework data provider for OLE DB, an `OleDbCommand` object is included.

`OracleCommand`

- In the .NET Framework data provider for Oracle, an `OracleCommand` object is included.

`OdbcCommand`

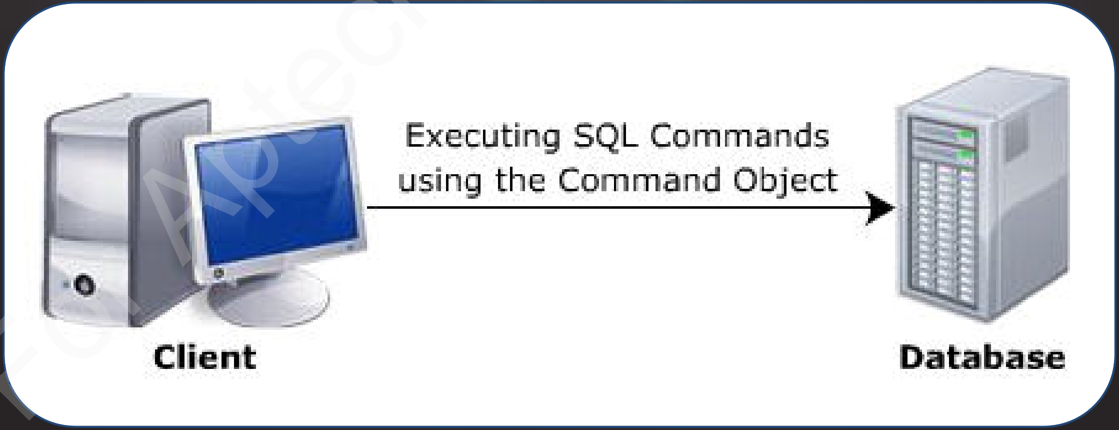
- In the .NET Framework data provider for ODBC, an `OdbcCommand` object is included.

`SqlCommand`

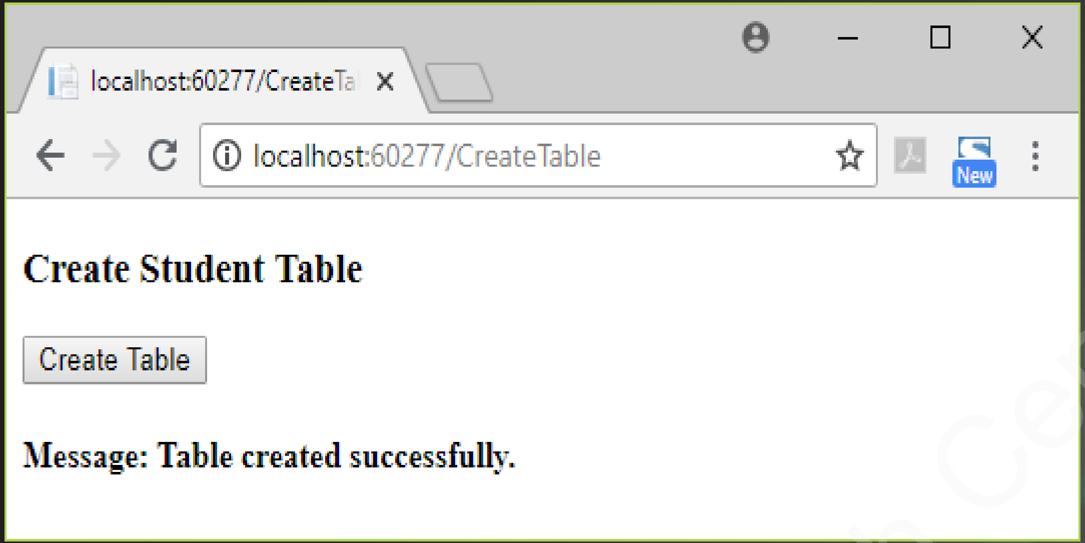
- In the .NET Framework data provider for SQL Server, an `SqlCommand` object is included.

Executing a Command

Method Name	Description
`ExecuteReader()`	Used for returning a `DataReader` object.
`ExecuteScalar()`	Used for returning a single scalar value.
`ExecuteNonQuery()`	Used for executing a command, which cannot return a row.
`ExecuteXMLReader()`	Used for returning an `XmlReader`. This is available only for an `SqlCommand` object.
`ExecuteReader()`	Used for returning a `DataReader` object.



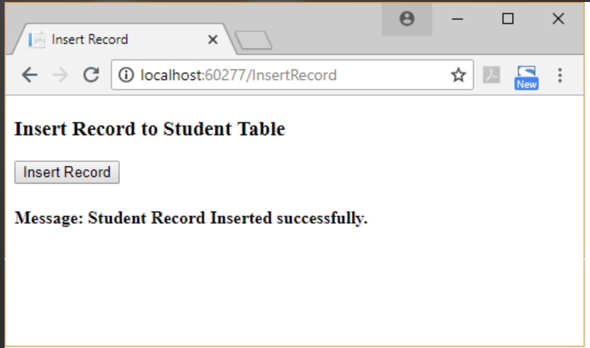
Connect to a Database and Execute a Command



Create Student Table

Create Table

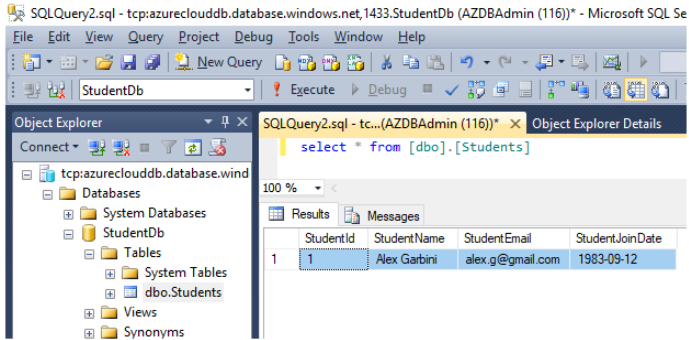
Message: Table created successfully.



Insert Record to Student Table

Insert Record

Message: Student Record Inserted successfully.



SQLQuery2.sql - tcp:azureclouddb.database.windows.net,1433.StudentDb (AZDBAdmin (116))* - Microsoft SQL Se

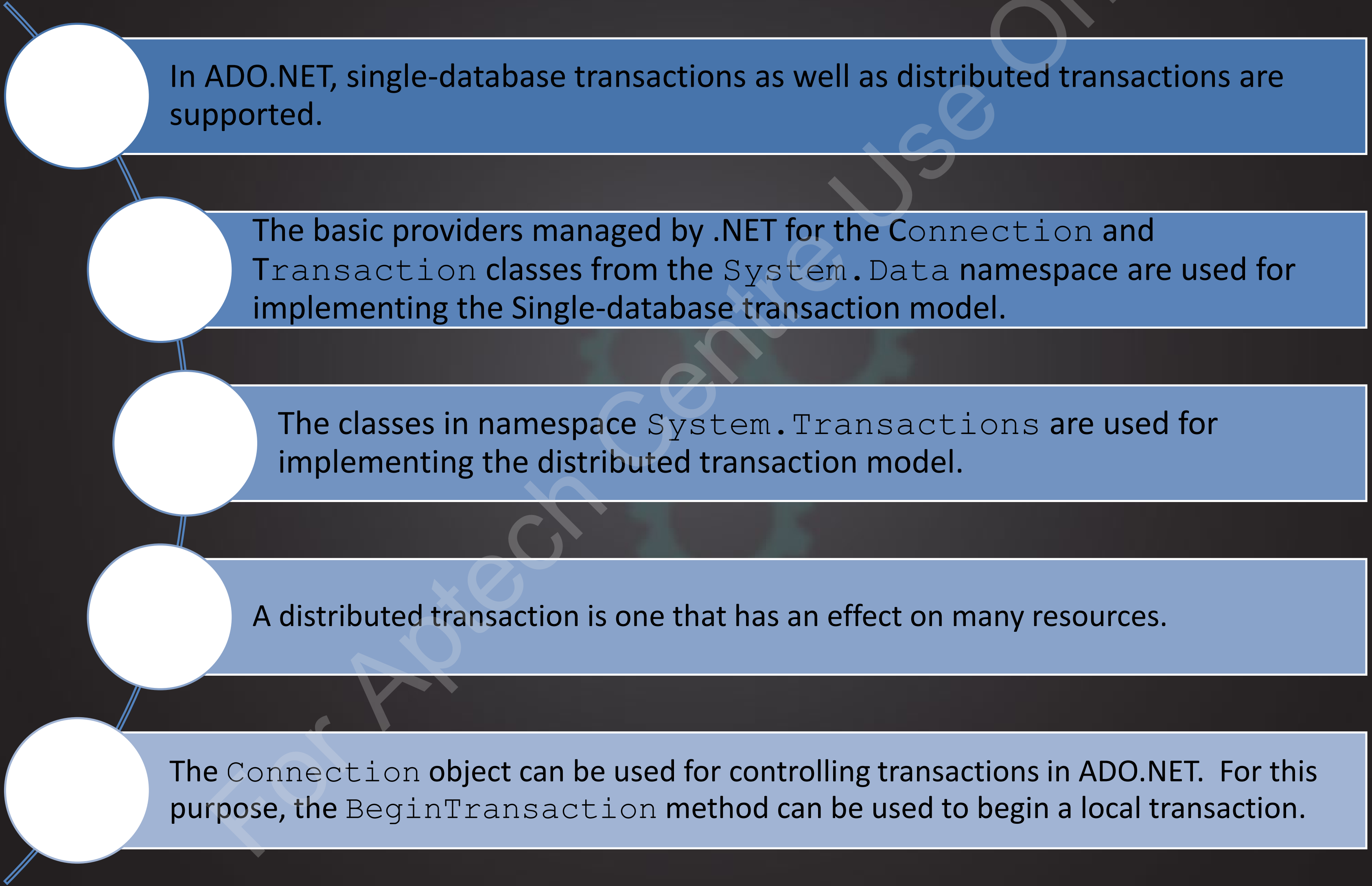
```
select * from [dbo].[Students]
```

	StudentId	StudentName	StudentEmail	StudentJoinDate
1	1	Alex Garbini	alex.g@gmail.com	1983-09-12

ADO.NET DataReader

Use ADO.NET DataReader to obtain a read-only stream of data, which is also forward-only, from a database. When the query is executed, results are returned. These results are kept in the client's network buffer. They can be accessed by raising a request using the DataReader's `Read` method. Using the DataReader helps in improving the performance of an application.

Managing Database Transactions



In ADO.NET, single-database transactions as well as distributed transactions are supported.

The basic providers managed by .NET for the `Connection` and `Transaction` classes from the `System.Data` namespace are used for implementing the Single-database transaction model.

The classes in namespace `System.Transactions` are used for implementing the distributed transaction model.

A distributed transaction is one that has an effect on many resources.

The `Connection` object can be used for controlling transactions in ADO.NET. For this purpose, the `BeginTransaction` method can be used to begin a local transaction.

Summary

- ADO.NET comprises a set of classes that offer data access services for .NET Framework.
- In ADO.NET, data providers of the .NET Framework act as a link between a data source and an application.
- SQL Server .NET Framework Data Provider supports a connection string format.
- After a successful connection is made with a data source, a user can run commands to manipulate or retrieve data.
- A DataReader is a forward-only and read-only cursor type and can be used to display data.
- In ADO.NET, single-database transactions as well as distributed transactions are supported.