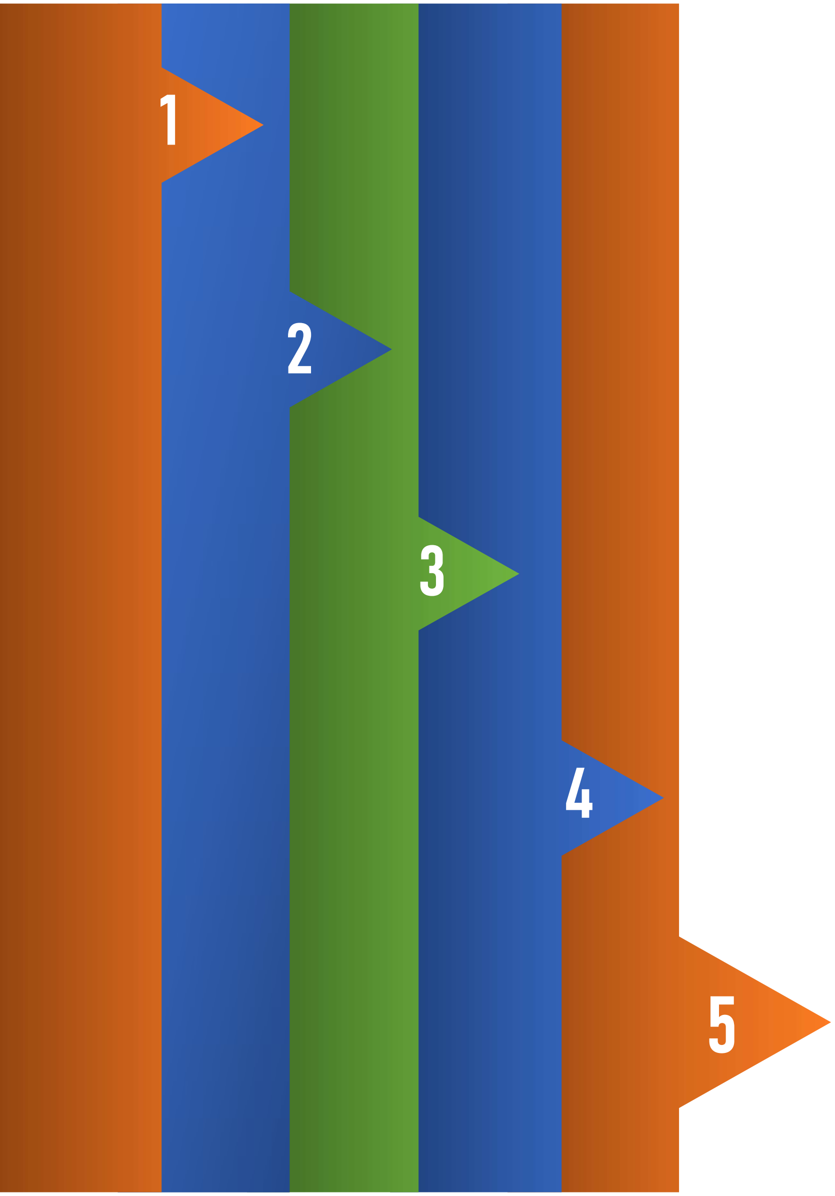

ADO.Net

1. **Connect to the Database**
2. **Prepare an SQL Command /Stored Procedure**
3. **Execute the SQL Command**
4. **Executing stored procedure**
5. **Retrieve the results and display them in the application**

Connect to Database

- Database connection can be established by invoking class SqlConnection
- Make object for class SqlConnection and use methods like Open() to open the connection and Close() to close the connection
- For accessing Database pass database credentials as arguments while SqlConnection object creation

```
SqlConnection con = null; try {
con = new SqlConnection("data source=MC1JUNB2145; database=student;
user id = sa; password = pass@word1");
//code for sql query
con.Open();
//code for query execution
con.Close(); }catch(Exception e){ WriteLine(e.Message);}
```

DatabasePrepare an SQL Command /Stored Procedure

- To access the database's data after establishing a connection, a sqlcommand object is generated.
- Give the command object the SQL command text and the connection object as arguments.
- Whenever procedures are involved, the SqlCommand object should be passed the procedure command and the connection object as parameters.

```
SqlCommand cm = new SqlCommand("select * from student_table", con);

SqlCommand cm = new SqlCommand("insert into student_table values (105,
'Ramesh', 'Ramesh@dotnettutorial.net', '1122334455')", con);

SqlCommand cm = new SqlCommand("Exec [dbo].[UpdateRestdate]
@bookid,@bookdate ", con);
```

Execute the SQL Command

- After making command object calling ExecuteReader() or ExecuteNonQuery method using the command object to execute the sql script.
- If the sql command text like insert or create or update or delete which returns single value as output parameter after exection then ExecuteNonQuery is called and store the value in a variable.
- If the sql command text returns a sequence of data like select statement then ExecuteReader method is called and stored the return value in SqlDataReader Class.

```
SqlCommand cm = new SqlCommand("select * from student_table", con);
SqlDataReader sdr = cm.ExecuteReader();

SqlCommand cmd = new SqlCommand("insert into student_table values (105, 'Ramesh',
'Ramesh@dotnettutorial.net', '1122334455')", con);
int rowsAffected = cmd.ExecuteNonQuery();
```

Executing stored procedure

- As like sql in while executing stored procedures if the procedure returns single value then ExecuteNonQuery method is called.
- Else if the procedure returns a sequence of data, then ExecuteReader method is called and the value stored in SqlDataReader class.

```
 SqlCommand cmobj = new SqlCommand("Exec [dbo].[UpdateRestdate] 
@bookid,@bookdate ", conobj);
                cmobj.Parameters.AddWithValue("@bookid", 
SqlDbType.Int).Value = e.bookid;
                cmobj.Parameters.AddWithValue("@bookdate", 
SqlDbType.DateTime).Value = e.bookdate;
//int numberofrow updated = cmobj.ExecuteNonQuery();
                  SqlDataReader sd = cmobj.ExecuteReader();
```

Retrieve the results and display them in the application

- After gertting sequence of data from sql query or stored procedure a reference object of Sql data reader is used to store the data.
- This object is used to call Read method to retrieve all stored datain the datareader object and can print data using the respective field name.
- This read method will return the values until there is no data to retrieve or it return as false.

```
SqlDataReader sd = cmobj.ExecuteReader();
            if (sd.HasRows)
            {                    while (sd.Read())
           {
Console.WriteLine("bookid : " + sd["BookingIds"] + "\n" + "id : " + sd["UserId"] + "\n" +
"RestaurantId = " + sd["RestaurantId"] + "\n" + "BookingDate = " + sd["BookingDate"] + "\n" +
"MealType = " + sd["MealType"] + "\n" + "NoOfPeople = " + sd["NoOfPeople"]);
                }
```

Ok guys,Please wake up

Thank you for your time.