string cnnString =WebConfigurationManager.ConnectionStrings["MyString"].ConnectionString;

SqlConnection cnn = new SqlConnection(cnnString);

SqlTransaction transaction;

cnn.Open();

transaction = cnn.BeginTransaction();

try

{

SqlCommand cmd1 = new SqlCommand("sproc1", cnn);

cmd1.CommandType = CommandType.StoredProcedure;

cmd1.Parameters.Add(new SqlParameter("@Param1", SqlDbType.NVarChar, 50));

cmd1.Parameters["@Param1"].Value = paramValue1;

**cmd1.ExecuteNonQuery();**

}

catch (SqlException sqlEx)

{

transaction.Rollback();

}

finally

{

cnn.Close();

cnn.Dispose();

}

transaction.Commit();

|  |  |
| --- | --- |
| SCALAR |  |

<connectionStrings>

<add name="SqlConn" connectionString="Server=ORLDWV004;Database=ATT\_UID;Trusted\_Connection=True;" providerName="System.Data.SqlClient"/>

public static string ExecScalar(string EmplID)

{

try

{

SqlConnection sqlcon = new

SqlConnection(ConfigurationManager.ConnectionStrings["SqlConn"].ConnectionString);

sqlcon.Open();

SqlCommand sqlCommand = new SqlCommand();

sqlCommand.Connection = sqlcon;

sqlCommand.CommandText = "CheckDuplicate";

sqlCommand.CommandType = System.Data.CommandType.StoredProcedure;

sqlCommand.Parameters.AddWithValue("@EmplID", EmplID);

var attuid = sqlCommand.ExecuteScalar();

if (attuid == null) attuid = "";

sqlcon.Close();

return attuid.ToString();

}

catch (Exception es)

{

log.Error("When searching ATT\_Duplicate" + es.ToString());

return "-";

}

}