# SQL Server

<http://www.analyticsinaction.com/tsql/>

<http://www.dofactory.com/sql/tutorial>

## Special Operators

IS NULL / IS NOT NULL

LIKE ‘D%’ -- All names starting from D

LIKE ‘%D’

LIKE ‘\_D\_’ -- 1 or more at start/end

LIKE ‘\_\_D\_\_’ -- 2 or more at start/end

LIKE ‘\_%D%\_’ -- 1 or more at start/end

LIKE ‘%D%’ -- 0 or more at start/end – get names starting/ending with D as well

WHERE FirstName >= ‘D’ -- All names starting from D to Z

WHERE FirstName >= ‘D’ AND FirstName < ‘E’ -- All names starting from D

WHERE NOT EXISTS (SUBQUERY)

## Date

SELECT CONVERT(DATE, OrderDate) "Order Date" -- To convert datetime (can be any datatype) to date

OrderDate > DATEADD(MONTH, -12, GETDATE()) -- orders in previous 12 months

OrderDate > DATEADD(MONTH, -12, ‘2017-09-17’)

## Alias

"Order Date" OR [Order Date]

## Group by

WHERE [CONDITIONS]

GROUP BY [SELECTED COLUMNS]

HAVING [AGG COLUMNS]

ORDER BY [SELECTED COLUMNS]

## Joins

### Multiple tables

SELECT C.FirstName, O.Id, COUNT(OI.Id), O.OrderNumber, O.OrderDate, SUM(OI.Quantity) Quantity, SUM(OI.UnitPrice) UnitPrice

FROM Customer C

LEFT JOIN "Order" O ON O.CustomerId = C.Id

LEFT JOIN OrderItem OI ON O.Id = OI.OrderId

GROUP BY C.FirstName, O.Id, O.OrderNumber, O.OrderDate

## ESCAPING

Use ‘

IKE ‘Men’’s sports%’

## INDEX

CREATE INDEX IX\_tblCustomer\_FirstName ON Customer (FirstName)

## Pivoted values

Use >, <=, >=

### IMP

1 SELECT C.FirstName, O.Id, COUNT(OI.Id), O.OrderNumber, O.OrderDate, SUM(OI.Quantity) Quantity, SUM(OI.UnitPrice) UnitPrice

FROM Customer C

LEFT JOIN ("Order" O

INNER JOIN OrderItem OI ON O.Id = OI.OrderId)

ON O.CustomerId = C.Id

GROUP BY C.FirstName, O.Id, O.OrderNumber, O.OrderDate

2 SELECT C.FirstName, O.Id, COUNT(OI.Id), O.OrderNumber, O.OrderDate, SUM(OI.Quantity) Quantity, SUM(OI.UnitPrice) UnitPrice

FROM "Order" O

INNER JOIN OrderItem OI ON O.Id = OI.OrderId

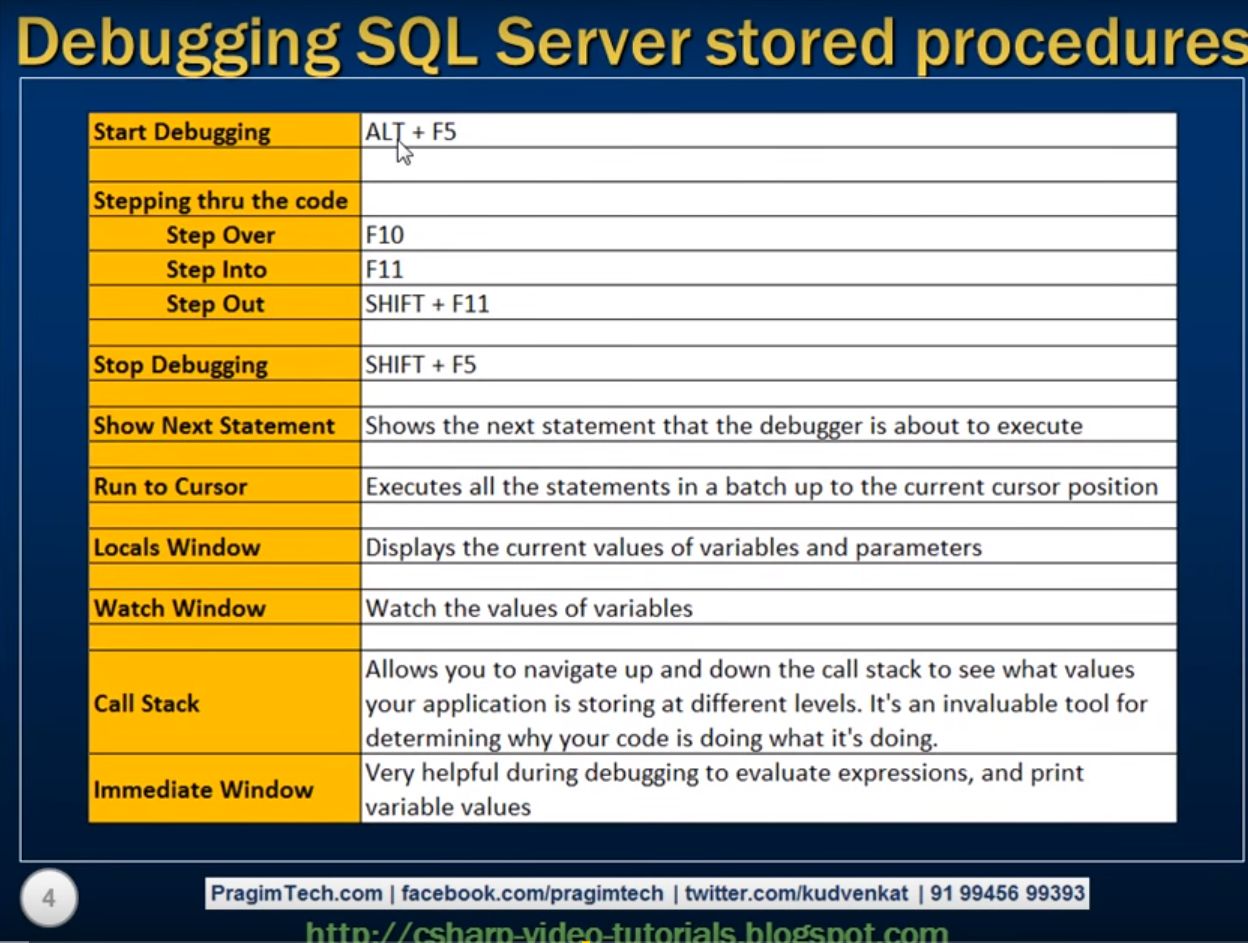
RIGHT JOIN Customer C ON O.CustomerId = C.Id – mention last

GROUP BY C.FirstName, O.Id, O.OrderNumber, O.OrderDate

Both have same result

Preferably use RIGHT / LEFT joins at last

## Debugging



## Assigning to variables using select

BEGIN

DECLARE @Id INT

DECLARE @OrderDate DATE

DECLARE @OrderNumber VARCHAR(100)

DECLARE @CustomerId1 INT

DECLARE @TotalAmount INT

BEGIN TRY

--Insert into "Order" (TotalAmount) values ('abc')

SELECT

@Id = Id,

@OrderDate = CONVERT(DATE, OrderDate),

@OrderNumber = OrderNumber,

@CustomerId1 = CustomerId,

@TotalAmount = TotalAmount

FROM "Order"

WHERE CustomerId = 1

END TRY

BEGIN CATCH

INSERT INTO ErrorLog (ErrorNumber, ErrorProcedure, ErrorMessage, ErrorLine, CreatedDate)

VALUES ( ERROR\_NUMBER(), ERROR\_PROCEDURE(), ERROR\_MESSAGE(), ERROR\_LINE(), SYSDATETIME())

END CATCH

END

# C#/ ASP

## DAL : steps for connecting to database

1. Add Connection String
2. Add configuration manager from assemblies

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

namespace DataAccessLayer

{

    public class DAL

    {

        public DAL()

        {

        }

        public static string ConnectionString

        {

            get

            {

                return ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString.ToString();

            }

        }

        public SqlParameter AddParameter(string parameterName, object value, SqlDbType DbType, int size)

        {

            SqlParameter param = new SqlParameter();

            param.ParameterName = parameterName;

            param.Value = value;

            param.SqlDbType = DbType;

            param.Size = size;

            param.Direction = ParameterDirection.Input;

            return param;

        }

        public DataTable ExecuteDTByProcedure(string ProcedureName, SqlParameter[] Params)

        {

            SqlConnection conn = new SqlConnection(ConnectionString);

            SqlCommand cmd = new SqlCommand();

            cmd.Connection = conn;

            cmd.CommandText = ProcedureName;

            cmd.Parameters.AddRange(Params);

            cmd.CommandType = CommandType.StoredProcedure;

            SqlDataAdapter adapter = new SqlDataAdapter(cmd);

            DataTable dTable = new DataTable();

            try

            {

                adapter.Fill(dTable);

            }

            catch (Exception ex)

            {

            }

            finally

            {

                // Dispose objects

                adapter.Dispose();

                cmd.Parameters.Clear();

                cmd.Dispose();

                conn.Dispose();

            }

            return dTable;

        }

    }

}

## Imp points

try

{

connection.Open();

// ExecuteNonQuery() is Used to execute the insert command.

// This inserts the data into the DB.

output = command.ExecuteNonQuery();

}

catch (Exception ex)

{

throw ex;

}

finally

{

connection.Close();

}

## DataTable

### Assign values of DataTable to controls

BO\_CRM objCRM = new BO\_CRM();

            BLL objBLL = new BLL();

            objCRM.CustomerId = 0;

            DataTable dt = objBLL.GetOrders(objCRM); ;

            GridView2.DataSource = dt;

            GridView2.DataBind();

            DropDownList1.DataValueField = "Order Id"; //"Id";

            DropDownList1.DataTextField = "Order Number";  //"OrderNumber";

            DropDownList1.DataSource = dt;

            DropDownList1.DataBind();

            txtOrderNumber.Text = dt.Rows[4][2].ToString();