Created By: Christopher Singleton

```
--=======================================================--
--============ Useful Common Statements ===============--
--=======================================================--
--Note: These are some of the most useful T-SQL statements that I use every day.

--================== IF The Database Exists Then Select It ========================--

IF EXISTS (SELECT name FROM sys.databases WHERE name = N'DatabaseName')
   BEGIN
     ALTER DATABASE [DatabaseName] SET SINGLE_USER WITH ROLLBACK IMMEDIATE

--======================= Drop This Database and End Function ======================--

     DROP DATABASE [DatabaseName] --If it already exists so we can start fresh.
              /*Print out that the table was dropped */
     /*Convert to sysdatetime and then cast to varchar. */
     PRINT 'DatabaseName Database: Dropped Database Successfully.'
     + CAST(CONVERT(varchar, SYSDATETIME(), 121) AS varchar (20))
   END

--=================== Create the Database with Settings ======================--

 --Its always a good idea to plan the size of the database for growth. --Plenty of size...
 CREATE DATABASE [CustomerTransact] ON PRIMARY
    (NAME = N'DatabaseName'
     , FILENAME = N'D:\Location\DatabaseName.mdf'
     --Store Database Here
     , SIZE = 10MB
     , MAXSIZE = 1GB
     , FILEGROWTH = 10MB)
     LOG ON
     (NAME = N'DatabaseName_log'
     --Store Log File Here
     , FILENAME = N'D:\Location\DatabaseName_log.LDF'
     , SIZE = 1MB
     , MAXSIZE = 1GB
     , FILEGROWTH = 10MB)
 GO

 /*Log In Owner Database Name = SA
   Note: SA Means "System Administrator"*/
 EXEC [DatabaseName].dbo.sp_changedbowner @loginame = N'SA', @map=false
GO
```

```
--================Set's the Recovery Record Log Settings=================--

   /*Note: Only use this mode "BULK_LOGGED" when there are no other users, otherwise
     data loss can happen. */

 ALTER DATABASE [DatabaseName] SET RECOVERY BULK_LOGGED
   -- Below: Prints out the system Date Time with a message.
   --Print out that the database was created with Date/Time, CAST to varchar.
 PRINT 'DatabaseName Database: Database was Created Successfully.'
    + CAST(CONVERT(varchar, SYSDATETIME(), 121) AS varchar (20))

--=============== Stored Procedures =================--

IF EXISTS (SELECT * FROM sys.objects WHERE object_id =
   OBJECT_ID(N'[dbo].[ProcedureName]') AND type in (N'P', N'PC'))
   DROP PROCEDURE [dbo].[ProcedureName]

--=================== Triggers =====================--

IF EXISTS (SELECT name FROM sysobjects
   WHERE name = 'TriggerName' AND type = 'TR')
   DROP TRIGGER TriggerName

--=================== Functions ====================--

IF EXISTS (
 SELECT * FROM sysobjects WHERE id = object_id(N'FunctionName')
 AND xtype IN (N'FN', N'IF',N'TF')
)
 DROP FUNCTION FunctionName

--==================== Tables ======================--

--Demonstrats Drop a Permanent Table:
IF OBJECT_ID('dbo.Table', 'U') IS NOT NULL
   DROP TABLE [dbo].[Table];
GO
--Demonstrates Drop a Temp Table:
IF OBJECT_ID('tempdb.dbo.#TempTableName', 'U') IS NOT NULL
   DROP TABLE #TempTableName;

      -- Give info on the table.
   EXEC sp_help TableName

   --If you need to drop a user from a table.
   IF [dbo].[TableName]('User') = 1 Drop table [User]
GO
```

```
--This is a test Function that returns 1 if the table exists or 0 if not.
CREATE FUNCTION [dbo].[Table_Exists]
(
   @TableName VARCHAR(200)
)
   RETURNS BIT
AS
BEGIN
   If Exists(select * from INFORMATION_SCHEMA.TABLES where TABLE_NAME = @TableName)
      RETURN 1;

   RETURN 0;
END
GO

--===================== Rows =========================--

SELECT * FROM [dbo].[SalesOrder];
GO

--Check to see if the Stored Procedure Exists:
IF EXISTS (SELECT * FROM sys.objects WHERE object_id =
  OBJECT_ID(N'[dbo].[uspUpdateCustomerTransact]') AND type in (N'P', N'PC'))
  DROP PROCEDURE [dbo].[uspUpdateCustomerTransact]
GO

CREATE PROCEDURE uspUpdateCustomerTransact
@SaleAmount decimal(8,2) NULL
AS
/*
Created By: Chris Singleton
Date: 02/26/2017
About: Updates one row only when Stored Procedure is called.
Adds 10 cents on the SaleAmount of 26.00 for customerID 1.
*/
BEGIN
    UPDATE TOP (1) CustomerTransact.dbo.SalesOrder
    SET SaleAmount = @SaleAmount + .1
    WHERE CustomerID = 1 AND SaleAmount = 26.00

END;
DECLARE @Ret int
EXEC @Ret = uspUpdateCustomerTransact 26.00;
IF @ret = 0
    PRINT 'Error!';
ELSE
    PRINT 'OrderId entered: ' + cast(@ret as varchar);
GO
DROP PROCEDURE uspUpdateCustomerTransact
```