**Stored Procedures and Transactions Notes**

**Rules:**

**Stored Procedures with DML**

Always do the following:

1. Always default parameters to null
   * Check for null (is null)
2. Always check if a DML statement fails
   * Put @@ error
3. Always check for existence of record(s) you are updating/deleting BEFORE the update or delete (DML statement)
   * Exists or not exists
4. \*\*\*Declare variables if data will be reused

**Transactions**

Must put this for every procedure that has **more than one** DML statement:

1. Begin transaction
2. Do DML Statements
3. Rollback transaction
4. Commit transaction

**Syntax**

**SP with Update/Delete DML statement**

Create procedure ProcedureName (@Parameter1 DataType(Value) = null, @Parameter2 DataType(Value) = null, ….)

As

If

@Parameter1 is null or @Parameter2 is null or ….

Begin

RaisError (‘insert error message’,16,1)

End

Else

Begin

--*If DML Statement is an update or delete: check if record exists BEFORE the DML*

*statement*

If [not] exists (insert select statement that needs to be checked)

Begin

Raiserror('insert error message',16,1)

End

Else

--*Do DML statement required*

Begin

--[DML Statement]

--*Check @@ERROR for all DML statements*

If @@ERROR <> 0

Begin

Raiserror('insert error message',16,1)

End

End

End

Return

**SP with Transaction**

Create procedure ProcedureName (@Parameter1 DataType(Value) = null, @Parameter2 DataType(Value) = null, ...)

As

--*Declare variables if needed*

Declare @Variable1 DataType

Declare @Variable2 DataType

...

If

@Parameter1 is null or @Parameter2 is null or ….

Begin

RaisError (‘insert error message’,16,1)

End

Else

Begin

--*If DML Statement is an update or delete: check if record exists BEFORE the DML*

*statement*

If [not] exists (insert select statement that needs to be checked)

Begin

Raiserror('insert error message',16,1)

End

Else

--*Do DML statements required*

Begin

--*Do transactions for multiple DML statements*

Begin Transaction

--[DML Statement 1]

--*Check @@ERROR for all DML statements*

If @@ERROR <> 0

Begin

Raiserror('insert error message',16,1)

--*Do a Rollback if the DML statement fails*

Rollback Transaction

End

Else

Begin

--[DML Statement 2]

--*Check @@ERROR for all DML statements*

If @@ERROR <> 0

Begin

Raiserror('insert error message',16,1)

--*Do a Rollback if the DML statement fails*

Rollback Transaction

End

Else

Begin

--*Commit the transaction at the end if there are no errors*

Commit Transaction

End

End --End DML Statement 2 Code Block

End --End DML Statement 1 Code Block

End --End Exists Code Block

Return