**CCO\_eCoaching\_V2\_DB\_Implementation\_Plan**

Contents

[Summary 2](#_Toc386805558)

[1. Create Database – Will be done by DBA 2](#_Toc386805559)

[2. Create Schema [EC] and Users 2](#_Toc386805560)

[3. Create Tables 2](#_Toc386805561)

[4. Create Stored Procedures 2](#_Toc386805562)

[5. Create Functions 2](#_Toc386805563)

[6. Import Employee\_Hierarchy table Data from production Coaching DB on VDENSSDBP07\SCORP01 2](#_Toc386805564)

[7. Import DIM Table(s) and Supporting Table(s) Data from eCoachingDev DB on VRIVFSSDBT02\SCORD01,1437 2](#_Toc386805565)

[8. Initial Populate of Employee\_ID\_TO\_Lan\_ID Table 2](#_Toc386805566)

[9. Initial Populate for CSR Hierarchy 2](#_Toc386805567)

[10. Stage SSIS Packages and Config Files 2](#_Toc386805568)

[11. Transfer Historical Data 2](#_Toc386805569)

[12. Setup and Configure SQL agent Jobs – May not be needed. 2](#_Toc386805570)

[Details 2](#_Toc386805571)

[1. Create Database eCoaching 2](#_Toc386805572)

[2. Create Schema EC and Users 2](#_Toc386805573)

[3. Create Tables 5](#_Toc386805574)

[4. Create Procedures 23](#_Toc386805575)

[5. Create Functions 149](#_Toc386805576)

[6. Import Employee Hierarchy Data from Old Coaching DB 168](#_Toc386805577)

[7. Import DIM and Supporting Table Data from eCoaching Dev 169](#_Toc386805578)

[8. Initial Populate of Employee\_ID\_TO\_Lan\_ID Table 170](#_Toc386805579)

[9. Initial Populate for CSR Hierarchy 173](#_Toc386805580)

[10. Stage SSIS Packages 175](#_Toc386805581)

[11. Transfer Historical Data 175](#_Toc386805582)

# Summary

## Create Database – Will be done by DBA

## Create Schema [EC] and Users– Will be done by DBA

## Create Tables

## Create Stored Procedures

## Create Functions

## Import Employee\_Hierarchy table Data from production Coaching DB on VDENSSDBP07\SCORP01

## Import DIM Table(s) and Supporting Table(s) Data from eCoachingDev DB on VRIVFSSDBT02\SCORD01,1437

## Initial Populate of Employee\_ID\_TO\_Lan\_ID Table

## Initial Populate for CSR Hierarchy

## Stage SSIS Packages and Config Files

## Transfer Historical Data

## Setup and Configure SQL agent Jobs – May not be needed.

**Note:** Reminder: The updated IQS/verint feed to ecl with the additional Program value has to be implemented to coincide with this release.

# Details

For steps 1 and 2 please submit a ticket for DBA to complete.

Could get this done a day or 2 in advance.

### Create Database eCoaching

See note above. Will be done by DBA - Database eCoaching will be created.

(So they can set data files, log files, indexes, temp file on appropriate disks and configure backups).

Database name eCoaching

### Create Schema EC and Users

See note above. Will be done by DBA - Database eCoaching will be created.

/\*\*\*\*\*\* Object: Schema [EC] Script Date: 05/02/2014 09:54:42 \*\*\*\*\*\*/

CREATE SCHEMA [EC] AUTHORIZATION [dbo]

GO

/\*\*\*\*\*\* Object: DatabaseRole [coaching\_admin] Script Date: 05/02/2014 09:55:18 \*\*\*\*\*\*/

CREATE ROLE [ecoaching\_admin] AUTHORIZATION [dbo]

GO

/\*\*\*\*\*\* Object: DatabaseRole [coaching\_admin] Script Date: 05/02/2014 09:55:18 \*\*\*\*\*\*/

CREATE ROLE [ecoaching\_user] AUTHORIZATION [dbo]

GO

/\*\*\*\*\*\* Object: User [AD\brian.coughlin] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [AD\brian.coughlin] FOR LOGIN [AD\brian.coughlin] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [AD\Doug.Stearns] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [AD\Doug.Stearns] FOR LOGIN [AD\Doug.Stearns] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [AD\jourdain.augustin] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [AD\jourdain.augustin] FOR LOGIN [AD\jourdain.augustin] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [AD\Susmitha.Palacherla] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [AD\Susmitha.Palacherla] FOR LOGIN [AD\Susmitha.Palacherla] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\211geilde] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\211geilde] FOR LOGIN [VNGT\211geilde] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [vngt\211hinmda] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [vngt\211hinmda] FOR LOGIN [VNGT\211hinmda] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [vngt\211mishla] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [vngt\211mishla] FOR LOGIN [VNGT\211mishla] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\211queeti] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\211queeti] FOR LOGIN [VNGT\211queeti] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [vngt\augujo] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [vngt\augujo] FOR LOGIN [VNGT\augujo] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\cmsprod] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\cmsprod] FOR LOGIN [VNGT\cmsprod] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\coaching\_users] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\coaching\_users] FOR LOGIN [VNGT\coaching\_users]

GO

/\*\*\*\*\*\* Object: User [VNGT\cougbr] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\cougbr] FOR LOGIN [VNGT\cougbr] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\curtja] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\curtja] FOR LOGIN [VNGT\curtja] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\lefebi] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\lefebi] FOR LOGIN [VNGT\lefebi] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\palasu] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\palasu] FOR LOGIN [VNGT\palasu] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\VACMSMPMD01$] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\VACMSMPMD01$] FOR LOGIN [VNGT\VACMSMPMD01$] WITH DEFAULT\_SCHEMA=[EC]

GO

/\*\*\*\*\*\* Object: User [VNGT\VADENMWBP11$] Script Date: 05/02/2014 10:00:17 \*\*\*\*\*\*/

CREATE USER [VNGT\VADENMWBP11$] FOR LOGIN [VNGT\VADENMWBP11$] WITH DEFAULT\_SCHEMA=[EC]

GO

### Create Tables

Implementation by Engineering team begins here.

--1. Create Table [EC].[Coaching\_Log]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 12/23/2013

-- Description: Fact table to store coaching log records.

-- =============================================

CREATE TABLE [EC].[Coaching\_Log](

[CoachingID] [bigint] IDENTITY(1,1) NOT NULL,

[FormName] [nvarchar](50) NOT NULL,

[ProgramName] [nvarchar](50) NULL,

[SourceID] [int] NOT NULL,

[StatusID] [int] NOT NULL,

[SiteID] [int] NOT NULL,

[CSR] [nvarchar](50) NOT NULL,

[CSRID] [nvarchar](10) NOT NULL,

[SubmitterID] [nvarchar](10) NULL,

[EventDate] [datetime] NULL,

[CoachingDate] [datetime] NULL,

[isAvokeID] [bit] NULL,

[AvokeID] [nvarchar](40) NULL,

[isNGDActivityID] [bit] NULL,

[NGDActivityID] [nvarchar](40) NULL,

[isUCID] [bit] NULL,

[UCID] [nvarchar](40) NULL,

[isVerintID] [bit] NULL,

[VerintID] [nvarchar](40) NULL,

[VerintEvalID] [nvarchar](20) NULL,

[Description] [nvarchar](max) NULL,

[CoachingNotes] [nvarchar](4000) NULL,

[isVerified] [bit] NULL,

[SubmittedDate] [datetime] NULL,

[StartDate] [datetime] NULL,

[SupReviewedAutoDate] [datetime] NULL,

[isCSE] [bit] NULL,

[MgrReviewManualDate] [datetime] NULL,

[MgrReviewAutoDate] [datetime] NULL,

[MgrNotes] [nvarchar](3000) NULL,

[isCSRAcknowledged] [bit] NULL,

[CSRReviewAutoDate] [datetime] NULL,

[CSRComments] [nvarchar](3000) NULL,

[EmailSent] [bit] NULL,

[numReportID] [int] NULL,

[strReportCode] [nvarchar](30) NULL,

[isCoachingRequired] [bit] NULL,

[strReasonNotCoachable] [nvarchar](30) NULL,

[txtReasonNotCoachable] [nvarchar](3000) NULL,

CONSTRAINT [PK\_Coaching\_Log] PRIMARY KEY CLUSTERED

(

[CoachingID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [EC].[Coaching\_Log] ADD DEFAULT ((0)) FOR [EmailSent]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--2. Create Table [EC].[Coaching\_Log\_Reason]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 12/23/2013

-- Description: Fact table to store coaching reason attributes for the Coaching logs

-- stored in the main Coaching\_log table.

-- =============================================

CREATE TABLE [EC].[Coaching\_Log\_Reason](

[CoachingID] [bigint] NOT NULL,

[CoachingReasonID] [bigint] NOT NULL,

[SubCoachingReasonID] [bigint] NOT NULL,

[Value] [nvarchar](30) NULL,

CONSTRAINT [PK\_Coaching\_Log\_Reason] PRIMARY KEY CLUSTERED

(

[CoachingID] ASC,

[CoachingReasonID] ASC,

[SubCoachingReasonID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [EC].[Coaching\_Log\_Reason] WITH NOCHECK ADD CONSTRAINT [fkCoachingID] FOREIGN KEY([CoachingID])

REFERENCES [EC].[Coaching\_Log] ([CoachingID])

GO

ALTER TABLE [EC].[Coaching\_Log\_Reason] CHECK CONSTRAINT [fkCoachingID]

GO

-- 3. Table [EC].[Employee\_Hierarchy\_Stage]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 12/23/2013

-- Description: Used to stage Employee data from PeopleSoft.

-- =============================================

CREATE TABLE [EC].[Employee\_Hierarchy\_Stage](

[Emp\_ID] [nvarchar](10) NOT NULL,

[Emp\_Name] [nvarchar](70) NULL,

[Emp\_Email] [nvarchar](50) NULL,

[Emp\_Site] [nvarchar](50) NULL,

[Emp\_Job\_Code] [nvarchar](20) NULL,

[Emp\_Job\_Description] [nvarchar](50) NULL,

[Emp\_LanID] [nvarchar](30) NULL,

[Emp\_Program] [nvarchar](20) NULL,

[Sup\_Emp\_ID] [nvarchar](10) NULL,

[Sup\_Name] [nvarchar](70) NULL,

[Sup\_Email] [nvarchar](50) NULL,

[Sup\_Job\_Code] [nvarchar](50) NULL,

[Sup\_Job\_Description] [nvarchar](50) NULL,

[Sup\_LanID] [nvarchar](30) NULL,

[Mgr\_Emp\_ID] [nvarchar](10) NULL,

[Mgr\_Name] [nvarchar](70) NULL,

[Mgr\_Email] [nvarchar](50) NULL,

[Mgr\_Job\_Code] [nvarchar](50) NULL,

[Mgr\_Job\_Description] [nvarchar](50) NULL,

[Mgr\_LanID] [nvarchar](30) NULL,

[Start\_Date] [datetime] NULL,

[Active] [nvarchar](1) NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- 4. Table [EC].[Employee\_Hierarchy]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 1223/2013

-- Description: Used to store the Employee Information to be used by the Coaching application.

-- =============================================

CREATE TABLE [EC].[Employee\_Hierarchy](

[Emp\_ID] [nvarchar](10) NOT NULL,

[Emp\_Name] [nvarchar](70) NULL,

[Emp\_Email] [nvarchar](50) NULL,

[Emp\_LanID] [nvarchar](30) NULL,

[Emp\_Site] [nvarchar](50) NULL,

[Emp\_Job\_Code] [nvarchar](50) NULL,

[Emp\_Job\_Description] [nvarchar](50) NULL,

[Emp\_Program] [nvarchar](20) NULL,

[Sup\_ID] [nvarchar](10) NULL,

[Sup\_Name] [nvarchar](70) NULL,

[Sup\_Email] [nvarchar](50) NULL,

[Sup\_LanID] [nvarchar](20) NULL,

[Sup\_Job\_Code] [nvarchar](50) NULL,

[Sup\_Job\_Description] [nvarchar](50) NULL,

[Mgr\_ID] [nvarchar](10) NULL,

[Mgr\_Name] [nvarchar](70) NULL,

[Mgr\_Email] [nvarchar](50) NULL,

[Mgr\_LanID] [nvarchar](20) NULL,

[Mgr\_Job\_Code] [nvarchar](50) NULL,

[Mgr\_Job\_Description] [nvarchar](50) NULL,

[Start\_Date] [nvarchar](10) NULL,

[End\_Date] [nvarchar](10) NULL,

[Active] [nvarchar](1) NULL,

CONSTRAINT [PK\_Emp\_ID] PRIMARY KEY CLUSTERED

(

[Emp\_ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [EC].[Employee\_Hierarchy] ADD CONSTRAINT [DF\_\_Employee\_\_\_End\_D\_\_03317E3D] DEFAULT ((99991231)) FOR [End\_Date]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- 5. Table [EC].[EmpID\_To\_SupID\_Stage]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 01/02/2014

-- Description: Used to stage the Employee Sup Information from WFM.

-- =============================================

CREATE TABLE [EC].[EmpID\_To\_SupID\_Stage](

[Emp\_ID] [nvarchar](20) NOT NULL,

[Emp\_Name] [nvarchar](50) NULL,

[Emp\_Job\_Code] [nvarchar](5) NULL,

[Emp\_Site\_Code] [nvarchar](20) NULL,

[Sup\_ID] [nvarchar](20) NULL,

[Emp\_Program] [nvarchar](20) NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- 6. Table [EC].[EmployeeID\_To\_LanID]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Employee ID to lan ID links

-- =============================================

CREATE TABLE [EC].[EmployeeID\_To\_LanID](

[EmpID] [nvarchar](20) NOT NULL,

[StartDate] [int] NOT NULL,

[EndDate] [int] NOT NULL,

[LanID] [nvarchar](30) NOT NULL,

[DatetimeInserted] [datetime] NOT NULL,

[DatetimeLastUpdated] [datetime] NOT NULL,

CONSTRAINT [PK\_EmployeeID\_To\_LanID] PRIMARY KEY CLUSTERED

(

[LanID] ASC,

[StartDate] ASC,

[EndDate] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- 7. Table [EC].[CSR\_Hierarchy]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store CSR-SUP-MGR Hierarchy

-- =============================================

CREATE TABLE [EC].[CSR\_Hierarchy](

[EmpID] [nvarchar](10) NOT NULL,

[SupID] [nvarchar](10) NULL,

[MgrID] [nvarchar](10) NULL,

[StartDate] [datetime] NOT NULL,

[EndDate] [datetime] NULL,

CONSTRAINT [PK\_CSR\_ID] PRIMARY KEY CLUSTERED

(

[EmpID] ASC,

[StartDate] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--8. Create Table [EC].[DIM\_Date]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Date Dimension Data

-- =============================================

CREATE TABLE [EC].[DIM\_Date](

[DateKey] [int] NOT NULL,

[FullDate] [datetime] NULL,

[DateName] [nvarchar](11) NULL,

[DayOfWeek] [int] NULL,

[DayNameOfWeek] [nvarchar](10) NULL,

[DayOfMonth] [int] NULL,

[WeekOfYear] [int] NULL,

[MonthName] [nvarchar](10) NULL,

[MonthOfYear] [int] NULL,

[CalendarQuarter] [int] NULL,

[CalendarYear] [int] NULL,

[CalendarYearMonth] [nvarchar](7) NULL,

[CalendarYYYYQQ] [nvarchar](7) NULL,

CONSTRAINT [PK\_Dim\_Date] PRIMARY KEY CLUSTERED

(

[DateKey] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--9. Create Table [EC].[DIM\_Site]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Site Dimension Data

-- =============================================

CREATE TABLE [EC].[DIM\_Site](

[SiteID] [int] IDENTITY(1,1) NOT NULL,

[City] [nvarchar](20) NOT NULL,

[State] [nvarchar](20) NULL,

[StateCity] [nvarchar](30) NOT NULL,

CONSTRAINT [PK\_Site\_ID] PRIMARY KEY CLUSTERED

(

[SiteID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [EC].[Coaching\_Log] WITH NOCHECK ADD CONSTRAINT [fkCoachingSiteID] FOREIGN KEY([SiteID])

REFERENCES [EC].[Dim\_Site] ([SiteID])

GO

ALTER TABLE [EC].[Coaching\_Log] CHECK CONSTRAINT [fkCoachingSiteID]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--10. Create Table [EC].[DIM\_Source]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Source Dimension Data

-- =============================================

CREATE TABLE [EC].[DIM\_Source](

[SourceID] [int] NOT NULL,

[CoachingSource] [nvarchar](100) NOT NULL,

[SubCoachingSource] [nvarchar](100) NOT NULL,

CONSTRAINT [Source\_ID] PRIMARY KEY CLUSTERED

(

[SourceID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [EC].[Coaching\_Log] WITH NOCHECK ADD CONSTRAINT [fkCoachingSourceID] FOREIGN KEY([SourceID])

REFERENCES [EC].[Dim\_Source] ([SourceID])

GO

ALTER TABLE [EC].[Coaching\_Log] CHECK CONSTRAINT [fkCoachingSourceID]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--11. Create Table [DIM\_Status]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Status Dimension Data

-- =============================================

CREATE TABLE [EC].[DIM\_Status](

[StatusID] [int] IDENTITY(1,1) NOT NULL,

[Status] [nvarchar](100) NOT NULL,

CONSTRAINT [Status\_ID] PRIMARY KEY CLUSTERED

(

[StatusID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [EC].[Coaching\_Log] WITH NOCHECK ADD CONSTRAINT [fkCoachingStatusID] FOREIGN KEY([StatusID])

REFERENCES [EC].[Dim\_Status] ([StatusID])

GO

ALTER TABLE [EC].[Coaching\_Log] CHECK CONSTRAINT [fkCoachingStatusID]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--12. Create Table [DIM\_Coaching\_Reason]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Coaching Reason Dimension Data

-- =============================================

CREATE TABLE [EC].[DIM\_Coaching\_Reason](

[CoachingReasonID] [int] IDENTITY(1,1) NOT NULL,

[CoachingReason] [nvarchar](100) NOT NULL,

CONSTRAINT [Coaching\_Reason\_ID] PRIMARY KEY CLUSTERED

(

[CoachingReasonID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--13. Create Table [DIM\_Sub\_Coaching\_Reason]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Sub Coaching Reason Dimension Data

-- =============================================

CREATE TABLE [EC].[DIM\_Sub\_Coaching\_Reason](

[SubCoachingReasonID] [int] IDENTITY(1,1) NOT NULL,

[SubCoachingReason] [nvarchar](200) NOT NULL,

CONSTRAINT [Sub\_Coaching\_Reason\_ID] PRIMARY KEY CLUSTERED

(

[SubCoachingReasonID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

--14. Create Table [EC].[Migrated\_Employees]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Migrated Employee Data

-- =============================================

CREATE TABLE [EC].[Migrated\_Employees](

[Emp\_ID] [nvarchar](20) NOT NULL,

[Emp\_VNGT\_LanID] [nvarchar](30) NULL,

[Emp\_AD\_LanID] [nvarchar](30) NULL,

[Emp\_Name] [nvarchar](70) NULL,

[Emp\_Email] [nvarchar](50) NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--15. Create Table [EC].[Quality\_Coaching\_Stage]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to stage quality data

-- =============================================

CREATE TABLE [EC].[Quality\_Coaching\_Stage](

[Eval\_ID] [nvarchar](20) NULL,

[Eval\_Date] [datetime] NULL,

[Eval\_Site\_ID] [int] NULL,

[User\_ID] [nvarchar](20) NULL,

[User\_EMPID] [nvarchar](20) NULL,

[User\_LANID] [nvarchar](30) NULL,

[SUP\_ID] [nvarchar](20) NULL,

[SUP\_EMPID] [nvarchar](20) NULL,

[MGR\_ID] [nvarchar](20) NULL,

[MGR\_EMPID] [nvarchar](20) NULL,

[Journal\_ID] [nvarchar](30) NULL,

[Call\_Date] [datetime] NULL,

[Summary\_CallerIssues] [nvarchar](max) NULL,

[Coaching\_Goal\_Discussion] [nvarchar](4000) NULL,

[CSE] [nvarchar](2) NULL,

[Evaluator\_ID] [nvarchar](20) NULL,

[Program] [nvarchar](20) NULL,

[Source] [nvarchar](30) NULL,

[Oppor\_Rein] [nvarchar](20) NULL,

[Date\_Inserted] [datetime] NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--16. Create Table [EC].[Quality\_Coaching\_Fact]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Quality Fact data

-- =============================================

CREATE TABLE [EC].[Quality\_Coaching\_Fact](

[Eval\_ID] [nvarchar](20) NULL,

[Eval\_Date] [datetime] NULL,

[Eval\_Site\_ID] [int] NULL,

[User\_ID] [nvarchar](20) NULL,

[User\_EMPID] [nvarchar](20) NULL,

[User\_LANID] [nvarchar](30) NULL,

[SUP\_ID] [nvarchar](20) NULL,

[SUP\_EMPID] [nvarchar](20) NULL,

[MGR\_ID] [nvarchar](20) NULL,

[MGR\_EMPID] [nvarchar](20) NULL,

[Journal\_ID] [nvarchar](30) NULL,

[Call\_Date] [datetime] NULL,

[Summary\_CallerIssues] [nvarchar](max) NULL,

[Coaching\_Goal\_Discussion] [nvarchar](4000) NULL,

[CSE] [nvarchar](2) NULL,

[Evaluator\_ID] [nvarchar](20) NULL,

[Program] [nvarchar](20) NULL,

[Source] [nvarchar](30) NULL,

[Oppor\_Rein] [nvarchar](20) NULL,

[Date\_Inserted] [datetime] NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--17. Create Table [EC].[Quality\_Coaching\_Rejected]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store rejected Quality data

-- =============================================

CREATE TABLE [EC].[Quality\_Coaching\_Rejected](

[Eval\_ID] [nvarchar](20) NULL,

[Eval\_Date] [datetime] NULL,

[Eval\_Site\_ID] [int] NULL,

[User\_ID] [nvarchar](20) NULL,

[User\_EMPID] [nvarchar](20) NULL,

[User\_LANID] [nvarchar](30) NULL,

[SUP\_ID] [nvarchar](20) NULL,

[SUP\_EMPID] [nvarchar](20) NULL,

[MGR\_ID] [nvarchar](20) NULL,

[MGR\_EMPID] [nvarchar](20) NULL,

[Journal\_ID] [nvarchar](30) NULL,

[Call\_Date] [datetime] NULL,

[Summary\_CallerIssues] [nvarchar](max) NULL,

[Coaching\_Goal\_Discussion] [nvarchar](4000) NULL,

[CSE] [nvarchar](2) NULL,

[Evaluator\_ID] [nvarchar](20) NULL,

[Program] [nvarchar](20) NULL,

[Source] [nvarchar](30) NULL,

[Oppor\_Rein] [nvarchar](20) NULL,

[Reject\_reason] [nvarchar](40) NULL,

[Date\_Rejected] [datetime] NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--18. Create Table [EC].[Quality\_FileList]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Quality file information

-- =============================================

CREATE TABLE [EC].[Quality\_FileList](

[File\_SeqNum] [int] IDENTITY(1,1) NOT NULL,

[File\_Name] [nvarchar](260) NULL,

[File\_LoadDate] [datetime] NULL,

[Count\_Staged] [int] NULL,

[Count\_Loaded] [int] NULL,

[Count\_Rejected] [int] NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--19. Create Table [EC].[Outlier\_Coaching\_Stage]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to stage Outliers data

-- =============================================

CREATE TABLE [EC].[Outlier\_Coaching\_Stage](

[Report\_ID] [int] NULL,

[Report\_Code] [nvarchar](20) NULL,

[Form\_Type] [nvarchar](20) NULL,

[Source] [nvarchar](60) NULL,

[Form\_Status] [nvarchar](30) NULL,

[Event\_Date] [datetime] NULL,

[Submitted\_Date] [datetime] NULL,

[Start\_Date] [datetime] NULL,

[Submitter\_LANID] [nvarchar](30) NULL,

[Submitter\_Name] [nvarchar](30) NULL,

[Submitter\_Email] [nvarchar](50) NULL,

[CSR\_LANID] [nvarchar](30) NULL,

[CSR\_EMPID] [nvarchar](20) NULL,

[CSR\_Site] [nvarchar](20) NULL,

[Program] [nvarchar](30) NULL,

[CoachReason\_Current\_Coaching\_Initiatives] [nvarchar](20) NULL,

[TextDescription] [nvarchar](3000) NULL,

[FileName] [nvarchar](260) NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--20. Create Table [EC].[OutLier\_FileList]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Outliers file information

-- =============================================

CREATE TABLE [EC].[OutLier\_FileList](

[File\_SeqNum] [int] IDENTITY(1,1) NOT NULL,

[File\_Name] [nvarchar](260) NULL,

[File\_LoadDate] [datetime] NULL,

[Count\_Staged] [int] NULL,

[Count\_Loaded] [int] NULL,

[Count\_Rejected] [int] NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--21. Create Table [EC].[Outlier\_Coaching\_Rejected]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store rejected outliers records

-- =============================================

CREATE TABLE [EC].[Outlier\_Coaching\_Rejected](

[Report\_ID] [int] NULL,

[Report\_Code] [nvarchar](20) NULL,

[Form\_Type] [nvarchar](20) NULL,

[Source] [nvarchar](60) NULL,

[Form\_Status] [nvarchar](30) NULL,

[Event\_Date] [datetime] NULL,

[Submitted\_Date] [datetime] NULL,

[Start\_Date] [datetime] NULL,

[Submitter\_LANID] [nvarchar](30) NULL,

[Submitter\_Name] [nvarchar](30) NULL,

[Submitter\_Email] [nvarchar](50) NULL,

[CSR\_LANID] [nvarchar](30) NULL,

[CSR\_Site] [nvarchar](20) NULL,

[Program] [nvarchar](30) NULL,

[CoachReason\_Current\_Coaching\_Initiatives] [nvarchar](20) NULL,

[TextDescription] [nvarchar](3000) NULL,

[FileName] [nvarchar](260) NULL,

[Rejected\_Reason] [nvarchar](200) NULL,

[Rejected\_Date] [datetime] NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--22. Create Table [EC].[Outlier\_Coaching\_Fact]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/02/2014

-- Description: Used to store Outliers Fact data

-- =============================================

CREATE TABLE [EC].[Outlier\_Coaching\_Fact](

[Report\_ID] [int] NULL,

[Report\_Code] [nvarchar](20) NULL,

[Form\_Type] [nvarchar](20) NULL,

[Source] [nvarchar](60) NULL,

[Form\_Status] [nvarchar](30) NULL,

[Event\_Date] [datetime] NULL,

[Submitted\_Date] [datetime] NULL,

[Start\_Date] [datetime] NULL,

[Submitter\_LANID] [nvarchar](30) NULL,

[Submitter\_Name] [nvarchar](30) NULL,

[Submitter\_Email] [nvarchar](50) NULL,

[CSR\_LANID] [nvarchar](30) NULL,

[CSR\_Site] [nvarchar](20) NULL,

[Program] [nvarchar](30) NULL,

[CoachReason\_Current\_Coaching\_Initiatives] [nvarchar](20) NULL,

[TextDescription] [nvarchar](3000) NULL,

[FileName] [nvarchar](260) NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--23. Create Table [EC].[Historical\_Dashboard\_ACL]

-- =============================================

-- Author:Susmitha Palacherla

-- Create Date: 01/02/2014

-- Description: Used to store the list of users with access to the Historical Dashboard and ARC CSRs

-- =============================================

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [EC].[Historical\_Dashboard\_ACL](

[Row\_ID] [int] IDENTITY(1,1) NOT NULL,

[User\_LanID] [nvarchar](30) NOT NULL,

[User\_Name] [nvarchar](30) NOT NULL,

[Role] [nvarchar](30) NOT NULL,

[End\_Date] [nvarchar](10) NOT NULL,

[Updated\_By] [nvarchar](30) NULL,

[IsAdmin] [nvarchar](1) NULL,

CONSTRAINT [PK\_Historical\_Dashboard\_ACL] PRIMARY KEY CLUSTERED

(

[Row\_ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY],

UNIQUE NONCLUSTERED

(

[User\_LanID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [EC].[Historical\_Dashboard\_ACL] ADD DEFAULT (N'ECL') FOR [Role]

GO

ALTER TABLE [EC].[Historical\_Dashboard\_ACL] ADD DEFAULT (N'99991231') FOR [End\_Date]

GO

ALTER TABLE [EC].[Historical\_Dashboard\_ACL] ADD DEFAULT (N'N') FOR [IsAdmin]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--24. Create Table [EC].[Historical\_Source]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [EC].[Historical\_Source](

[strFormType] [varchar](50) NULL,

[strSource] [varchar](50) NULL,

[SourceID] [varchar](50) NULL

) ON [PRIMARY]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--25. Create Table [EC].[HistoricalMultiCodesForProgram]

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [EC].[HistoricalMultiCodesForProgram](

[EMPLID] [varchar](50) NULL,

[MULTI] [varchar](50) NULL,

[PSTARTDATE] [varchar](50) NULL,

[PENDDATE] [varchar](50) NULL,

[Program] [nvarchar](50) NULL

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Create Procedures

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 1: Coaching\_Log\_Create Procedures

--1. Create SP [EC].[sp\_InsertInto\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_InsertInto\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 02/03/2014

-- Description: This procedure inserts the e-Coaching records into the Coaching\_Log table.

-- The main attributes of the eCL are written to the Coaching\_Log table.

-- The Coaching Reasons are written to the Coaching\_Reasons Table.

--

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log]

( @nvcFormName Nvarchar(50),

@nvcCSR Nvarchar(40),

@nvcProgramName Nvarchar(50),

@intSourceID INT,

@intStatusID INT,

@SiteID INT,

@nvcSubmitter Nvarchar(40),

@dtmEventDate datetime,

@dtmCoachingDate datetime,

@bitisAvokeID bit ,

@nvcAvokeID Nvarchar(40)= NULL ,

@bitisNGDActivityID bit,

@nvcNGDActivityID Nvarchar(40)= NULL ,

@bitisUCID bit,

@nvcUCID Nvarchar(40)= NULL,

@bitisVerintID bit,

@nvcVerintID Nvarchar(255)= NULL,

@intCoachReasonID1 INT,

@intSubCoachReasonID1 INT,

@nvcValue1 Nvarchar(30),

@intCoachReasonID2 INT = NULL,

@intSubCoachReasonID2 INT = NULL,

@nvcValue2 Nvarchar(30) = NULL,

@intCoachReasonID3 INT = NULL,

@intSubCoachReasonID3 INT = NULL,

@nvcValue3 Nvarchar(30) = NULL,

@intCoachReasonID4 INT = NULL,

@intSubCoachReasonID4 INT = NULL,

@nvcValue4 Nvarchar(30) = NULL,

@intCoachReasonID5 INT,

@intSubCoachReasonID5 INT,

@nvcValue5 Nvarchar(30),

@intCoachReasonID6 INT,

@intSubCoachReasonID6 INT,

@nvcValue6 Nvarchar(30),

@intCoachReasonID7 INT,

@intSubCoachReasonID7 INT,

@nvcValue7 Nvarchar(30),

@intCoachReasonID8 INT,

@intSubCoachReasonID8 INT,

@nvcValue8 Nvarchar(30),

@intCoachReasonID9 INT,

@intSubCoachReasonID9 INT,

@nvcValue9 Nvarchar(30),

@intCoachReasonID10 INT,

@intSubCoachReasonID10 INT,

@nvcValue10 Nvarchar(30),

@intCoachReasonID11 INT,

@intSubCoachReasonID11 INT,

@nvcValue11 Nvarchar(30),

@intCoachReasonID12 INT,

@intSubCoachReasonID12 INT,

@nvcValue12 Nvarchar(30),

@nvcDescription Nvarchar(3000) ,

@nvcCoachingNotes Nvarchar(3000) ,

@bitisVerified bit ,

@dtmSubmittedDate datetime ,

@dtmStartDate datetime ,

@dtmSupReviewedAutoDate datetime ,

@bitisCSE bit ,

@dtmMgrReviewManualDate datetime ,

@dtmMgrReviewAutoDate datetime ,

@nvcMgrNotes Nvarchar(3000) ,

@bitisCSRAcknowledged bit ,

@dtmCSRReviewAutoDate datetime ,

@nvcCSRComments Nvarchar(3000),

@bitEmailSent bit

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

-- Fetch the Employee ID of the current User (@nvcCSR) and Employee ID of the Submitter (@nvcSubmitter).

DECLARE @nvcEmpID Nvarchar(10),

@nvcSubmitterID Nvarchar(10),

@dtmDate datetime

SET @dtmDate = GETDATE()

SET @nvcEmpID = EC.fn\_nvcGetEmpIdFromLanID(@nvcCSR,@dtmDate)

SET @nvcSubmitterID = EC.fn\_nvcGetEmpIdFromLanID(@nvcSubmitter,@dtmDate)

INSERT INTO [EC].[Coaching\_Log]

([FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[CSR]

,[CSRID]

,[SubmitterID]

,[EventDate]

,[CoachingDate]

,[isAvokeID]

,[AvokeID]

,[isNGDActivityID]

,[NGDActivityID]

,[isUCID]

,[UCID]

,[isVerintID]

,[VerintID]

,[Description]

,[CoachingNotes]

,[isVerified]

,[SubmittedDate]

,[StartDate]

,[SupReviewedAutoDate]

,[isCSE]

,[MgrReviewManualDate]

,[MgrReviewAutoDate]

,[MgrNotes]

,[isCSRAcknowledged]

,[CSRReviewAutoDate]

,[CSRComments]

,[EmailSent])

VALUES

(@nvcFormName

,@nvcProgramName

,@intSourceID

,@intStatusID

,@SiteID

,@nvcCSR

,@nvcEmpID

,@nvcSubmitterID

,@dtmEventDate

,@dtmCoachingDate

,@bitisAvokeID

,@nvcAvokeID

,@bitisNGDActivityID

,@nvcNGDActivityID

,@bitisUCID

,@nvcUCID

,@bitisVerintID

,@nvcVerintID

,@nvcDescription

,@nvcCoachingNotes

,@bitisVerified

,@dtmSubmittedDate

,@dtmStartDate

,@dtmSupReviewedAutoDate

,@bitisCSE

,@dtmMgrReviewManualDate

,@dtmMgrReviewAutoDate

,@nvcMgrNotes

,@bitisCSRAcknowledged

,@dtmCSRReviewAutoDate

,@nvcCSRComments

,@bitEmailSent)

SELECT @@IDENTITY AS 'Identity';

DECLARE @I BIGINT = @@IDENTITY

IF NOT @intCoachReasonID1 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID1,@intSubCoachReasonID1,

CASE WHEN @intCoachReasonID1 = 6 THEN 'Opportunity' ELSE @nvcValue1 END)

END

IF NOT @intCoachReasonID2 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID2,@intSubCoachReasonID2,@nvcValue2)

END

IF NOT @intCoachReasonID3 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID3, @intSubCoachReasonID3,@nvcValue3)

END

IF NOT @intCoachReasonID4 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID4, @intSubCoachReasonID4,@nvcValue4)

END

IF NOT @intCoachReasonID5 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID5, @intSubCoachReasonID5,@nvcValue5)

END

IF NOT @intCoachReasonID6 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID6, @intSubCoachReasonID6,@nvcValue6)

END

IF NOT @intCoachReasonID7 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID7, @intSubCoachReasonID7,@nvcValue7)

END

IF NOT @intCoachReasonID8 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID8, @intSubCoachReasonID8,@nvcValue8)

END

IF NOT @intCoachReasonID9 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID9, @intSubCoachReasonID9,@nvcValue9)

END

IF NOT @intCoachReasonID10 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID10, @intSubCoachReasonID10,@nvcValue10)

END

IF NOT @intCoachReasonID11 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID11, @intSubCoachReasonID11,@nvcValue11)

END

IF NOT @intCoachReasonID12 IS NULL

BEGIN

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID12, @intSubCoachReasonID12,@nvcValue12)

END

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

SELECT @ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE()

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

IF ERROR\_NUMBER() IS NULL

RETURN 1

ELSE IF ERROR\_NUMBER() <> 0

RETURN ERROR\_NUMBER()

ELSE

RETURN 1

END

END CATCH

END -- sp\_InsertInto\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--2. Create SP [EC].[sp\_SelectRecordStatus]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectRecordStatus'

)

DROP PROCEDURE [EC].[sp\_SelectRecordStatus]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 12/13/13

-- Description: This procedure selects the status of a record from the Coaching\_Log table

-- =====================================================================CREATE PROCEDURE [EC].[sp\_SelectRecordStatus] @strFormID nvarchar(50)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max)

SET @nvcSQL = 'SELECT s.Status strFormStatus

FROM [EC].[Coaching\_Log] cl WITH (NOLOCK),

[EC].[DIM\_Status] s

Where cl.StatusID = s.StatusID

And cl.FormName = '''+@strFormID+''''

EXEC (@nvcSQL)

END --sp\_SelectRecordStatus

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--3. Create SP [EC].[sp\_SelectCSRsbyLocation]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectCSRsbyLocation'

)

DROP PROCEDURE [EC].[sp\_SelectCSRsbyLocation]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: 12/13/13

-- Description: This procedure selects the CSRs from a table by location

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectCSRsbyLocation]

@strCSRSitein nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strEDate nvarchar(8),

@strRole1 nvarchar(30),

@strRole2 nvarchar(30),

@strRole3 nvarchar(30),

@strRole4 nvarchar(30)

Set @strEDate = '99991231'

Set @strRole1 = 'WACS01'

Set @strRole2 = 'WACS02'

Set @strRole3 = 'WACS03'

Set @strRole4 = '%Engineer%'

SET @nvcSQL = 'SELECT [Emp\_Name] + '' ('' + [Emp\_LanID] + '') '' + [Emp\_Job\_Description] as FrontRow1

,[Emp\_Name] + ''$'' + [Emp\_Email] + ''$'' + [Emp\_LanID] + ''$'' + [Sup\_Name] + ''$'' + [Sup\_Email] + ''$'' + [Sup\_LanID] + ''$'' + [Sup\_Job\_Description] + ''$'' + [Mgr\_Name] + ''$'' + [Mgr\_Email] + ''$'' + [Mgr\_LanID] + ''$'' + [Mgr\_Job\_Description] as BackRow1, [Emp\_Site]

FROM [EC].[Employee\_Hierarchy] WITH (NOLOCK)

where (([Emp\_Job\_Code] Like '''+@strRole1+''') OR ([Emp\_Job\_Code] Like '''+@strRole2+''') OR ([Emp\_Job\_Code] Like '''+@strRole3+''') OR ([Emp\_Job\_Description] Like '''+@strRole4+'''))

and [End\_Date] = ''99991231''

and [Emp\_Site] = '''+@strCSRSitein+'''

and [Emp\_LanID]is not NULL and [Sup\_LanID] is not NULL and [Mgr\_LanID]is not NULL

Order By [Emp\_Site] ASC, [Emp\_Name] ASC'

EXEC (@nvcSQL)

END -- sp\_SelectCSRsbyLocation

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--4. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_CSRCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_CSRCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_CSRCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 12/13/13

-- Description: This procedure selects the CSR e-Coaching records from the Coaching\_Log table

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_CSRCompleted] @strCSRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[eh].[Emp\_Name] strCSRName,

[eh].[Sup\_Name] strCSRSupName,

[eh].[Mgr\_Name] strCSRMgrName,

[S].[Status] strFormStatus,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and eh.[Emp\_LanID] = '''+@strCSRin+'''

and [S].[Status] = '''+@strFormStatus+'''

Order By [cl].[SubmittedDate] DESC'

EXEC (@nvcSQL)

END -- sp\_SelectFrom\_Coaching\_Log\_CSRCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--5. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_CSRPending]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_CSRPending'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_CSRPending]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 3/13/2014

-- Description: This procedure selects the CSR e-Coaching records from the Coaching\_Log table

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_CSRPending] @strCSRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30)

Set @strFormStatus = 'Pending CSR Review'

Set @strFormStatus2 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[eh].[Emp\_Name] strCSRName,

[S].[Status] strFormStatus,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and eh.[Emp\_LanID] = '''+@strCSRin+'''

and ([S].[Status] = '''+@strFormStatus+''' or [S].[Status] = '''+@strFormStatus2+''')

Order By [cl].[SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_CSRPending

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--6. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_HistoricalSUP]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_HistoricalSUP'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_HistoricalSUP]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: 4/30/12

-- Description: \* This procedure selects the CSR e-Coaching completed records to display on SUP historical page

-- Last Update: 03/07/2014 - Updated for ECoachingDev

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/28/2014> - Adapted for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_HistoricalSUP]

@strSourcein nvarchar(100),

@strCSRSitein nvarchar(30),

@strCSRin nvarchar(30),

@strSUPin nvarchar(30),

@strMGRin nvarchar(30),

@strSDatein datetime,

@strEDatein datetime,

@strIsOpp nvarchar(8),

@strStatusin nvarchar(30),

@strIsForce nvarchar(8)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

--@strFormStatus2 nvarchar(30),

--@strFormStatus3 nvarchar(30),

--@strFormStatus4 nvarchar(30),

@strSDate nvarchar(8),

@strEDate nvarchar(8)

Set @strFormStatus = 'Inactive'

--Set @strFormStatus2 = 'Pending CSR Review'

--Set @strFormStatus3 = 'Pending Supervisor Review'

--Set @strFormStatus4 = 'Pending Manager Review'

Set @strSDate = convert(varchar(8),@strSDatein,112)

Set @strEDate = convert(varchar(8),@strEDatein,112)

SET @nvcSQL = 'select x.strFormID

,x.strCSRName

,x.strCSRSupName

,x.strCSRMgrName

,x.strFormStatus

,x.strSource

,x.SubmittedDate

,x.strSubmitterName

,x.numOpportunity

,x.numReinforcement

from (

SELECT [cl].[FormName] strFormID

,[eh].[Emp\_Name] strCSRName

,[eh].[Sup\_Name] strCSRSupName

,[eh].[Mgr\_Name] strCSRMgrName

,[s].[Status] strFormStatus

,[so].[SubCoachingSource] strSource

,[cl].[SubmittedDate] SubmittedDate

,[sh].[Emp\_Name] strSubmitterName

,case when [clr].[Value] = ''Opportunity'' THEN 1 ELSE 0 END numOpportunity

,case when [clr].[Value] = ''Reinforcement'' THEN 1 ELSE 0 END numReinforcement

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK),

[EC].[DIM\_Source] so,

[EC].[Coaching\_Log\_Reason] clr,

[EC].[DIM\_Site] si

WHERE cl.CSRID = eh.Emp\_ID

AND cl.StatusID = s.StatusID

AND cl.SubmitterID = sh.EMP\_ID

AND cl.SourceID = so.SourceID

AND cl.CoachingID = clr.CoachingID

AND cl.SiteID = si.SiteID

AND [so].[SubCoachingSource] Like '''+@strSourcein+'''

and [s].[Status] Like '''+@strStatusin+'''

AND ISNULL([eh].[Emp\_Name], '' '') LIKE '''+@strCSRin+'''

AND ISNULL([eh].[Sup\_Name], '' '') LIKE '''+@strSUPin+'''

AND ISNULL([eh].[Mgr\_Name], '' '') LIKE '''+@strMGRin+'''

and ISNULL([si].[City], '' '') LIKE '''+@strCSRSitein+'''

and convert(varchar(8),[cl].[SubmittedDate],112) >= '''+@strSDate+'''

and convert(varchar(8),[cl].[SubmittedDate],112) <= '''+@strEDate+'''

and [s].[Status] <> '''+@strFormStatus+'''

) x

where ISNULL(x.numOpportunity, '' '') LIKE '''+@strIsOpp+'''

and ISNULL(x.numReinforcement, '' '') LIKE '''+@strIsForce+'''

Order By x.SubmittedDate DESC'

EXEC (@nvcSQL)

END -- sp\_SelectFrom\_Coaching\_Log\_HistoricalSUP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--7. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRCSRCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_MGRCSRCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRCSRCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the CSR e-Coaching records from the Coaching\_Log table

-- Where the status is Prnding Review.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/17/2014> - Removed Submitter name from query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRCSRCompleted]

@strSourcein nvarchar(100),

@strCSRMGRin nvarchar(30),

@strCSRSUPin nvarchar(30),

@strCSRin nvarchar(30),

@strSDatein datetime,

@strEDatein datetime

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strSDate nvarchar(8),

@strEDate nvarchar(8)

Set @strFormStatus = 'Completed'

Set @strSDate = convert(varchar(8),@strSDatein,112)

Set @strEDate = convert(varchar(8),@strEDatein,112)

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[eh].[Emp\_Name] strCSRName,

[eh].[Sup\_Name] strCSRSupName,

[eh].[Mgr\_Name] strCSRMgrName,

[s].[Status] strFormStatus,

[sc].[SubCoachingSource] strSource,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[DIM\_Source] sc,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and [cl].[SourceID] = [sc].[SourceID]

and [eh].[Mgr\_LanID] = '''+@strCSRMGRin+'''

and [S].[Status] = '''+@strFormStatus+'''

and [sc].[SubCoachingSource] Like '''+@strSourcein+'''

and [eh].[Emp\_Name] Like '''+@strCSRin+'''

and [eh].[Sup\_Name] Like '''+@strCSRSUPin+'''

and convert(varchar(8),[cl].[SubmittedDate],112) >= '''+@strSDate+'''

and convert(varchar(8),[cl].[SubmittedDate],112) <= '''+@strEDate+'''

Order By [cl].[SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_MGRCSRCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--8. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRCSRPending]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_MGRCSRPending'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRCSRPending]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the CSR e-Coaching records from the Coaching\_Log table

-- Where the status is Prnding Review.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/17/2014> - Removed Submitter name from query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRCSRPending]

@strCSRMGRin nvarchar(30),

@strCSRSUPin nvarchar(30),

@strSourcein nvarchar(100),

@strCSRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30),

@strFormStatus4 nvarchar(30)

Set @strFormStatus = 'Pending CSR Review'

Set @strFormStatus2 = 'Pending Supervisor Review'

Set @strFormStatus3 = 'Pending Manager Review'

Set @strFormStatus4 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[eh].[Emp\_Name] strCSRName,

[eh].[Sup\_Name] strCSRSupName,

[eh].[Mgr\_Name] strCSRMgrName,

[s].[Status] strFormStatus,

[sc].[SubCoachingSource] strSource,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[DIM\_Source] sc,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and [cl].[SourceID] = [sc].[SourceID]

and [eh].[Mgr\_LanID] = '''+@strCSRMGRin+'''

and (([S].[Status] = '''+@strFormStatus+''') or ([S].[Status] = '''+@strFormStatus2+''') or ([S].[Status] = '''+@strFormStatus3+''') or ([S].[Status] = '''+@strFormStatus4+'''))

and [eh].[Emp\_Name] Like '''+@strCSRin+'''

and [eh].[Sup\_Name] Like '''+@strCSRSUPin+'''

Order By [SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_MGRCSRPending

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--9. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRPending]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_MGRPending'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRPending]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the CSR e-Coaching records from the Coaching\_Log table

-- Where the status is Pending Review.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/14/2014> - Per Jourdain Augustin modified the CSR and SUP filters to use the name instead of the LanID

-- and changed the manager and status filters

-- Last Update: <03/17/2014> - Removed Submitter name from query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MGRPending]

@strCSRMGRin nvarchar(30),

@strCSRin nvarchar(30),

@strCSRSUPin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30)

Set @strFormStatus = 'Pending Manager Review'

Set @strFormStatus2 = 'Pending Supervisor Review'

Set @strFormStatus3 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[eh].[Emp\_LanID] strCSR,

[eh].[Emp\_Name] strCSRName,

[eh].[Sup\_Name] strCSRSupName,

[s].[Status] strFormStatus,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and (([eh].[Mgr\_LanID] = '''+@strCSRMGRin+''' and [S].[Status] = '''+@strFormStatus+''') OR ([eh].[Sup\_LanID] = '''+@strCSRMGRin+''' and ([S].[Status] = '''+@strFormStatus2+''' or [S].[Status] = '''+@strFormStatus3+''')))

and [eh].[Emp\_Name] Like '''+@strCSRin+'''

and [eh].[Sup\_Name] Like '''+@strCSRSUPin+'''

Order By [SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_MGRPending

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--10. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_MyCompSubmitted\_DashboardStaff]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_MyCompSubmitted\_DashboardStaff'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MyCompSubmitted\_DashboardStaff]

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the support staff's submitted comopleted records from the Coaching\_Log table and displayed on dashboard

-- Where the user's LAN is strSubmitter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/14> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MyCompSubmitted\_DashboardStaff]

@strUserin nvarchar(30),

@strCSRin nvarchar(30),

@strCSRSupin nvarchar(30),

@strCSRMgrin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID

,[s].[Status] strFormStatus

,[eh].[Emp\_Name] strCSRName

,[eh].[Sup\_Name] strCSRSupName

,[eh].[Mgr\_Name] strCSRMgrName

,[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

WHERE cl.CSRID = eh.Emp\_ID

AND cl.StatusID = s.StatusID

AND cl.SubmitterID = sh.EMP\_ID

AND sh.Emp\_LanID = '''+@strUserin+'''

AND [eh].[Emp\_Name]= '''+@strCSRin+'''

AND [eh].[Sup\_Name]= '''+@strCSRSupin+'''

AND [eh].[Mgr\_Name]= '''+@strCSRMgrin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order By [cl].[SubmittedDate] DESC'

EXEC (@nvcSQL)

END--sp\_SelectFrom\_Coaching\_Log\_MyCompSubmitted\_DashboardStaff

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--11. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_MyPenSubmitted\_DashboardStaff]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_MyPenSubmitted\_DashboardStaff'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MyPenSubmitted\_DashboardStaff]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the support staff's submitted pending records from the Coaching\_Log table and displayed on dashboard

-- Where the user's LAN is strSubmitter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/14> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MyPenSubmitted\_DashboardStaff]

@strUserin nvarchar(30),

@strCSRin nvarchar(30),

@strCSRSupin nvarchar(30),

@strCSRMgrin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30)

Set @strFormStatus = 'Pending CSR Review'

Set @strFormStatus2 = 'Pending Manager Review'

Set @strFormStatus3 = 'Pending Supervisor Review'

SET @nvcSQL = 'SELECT

cl.FormName strFormID

,S.Status strFormStatus

,eh.Emp\_Name strCSRName

,eh.Sup\_Name strCSRSupName

,eh.Mgr\_Name strCSRMgrName

,cl.SubmittedDate SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.SubmitterID = sh.Emp\_ID

and sh.Emp\_LanID = '''+@strUserin+'''

and eh.Emp\_Name Like '''+@strCSRin+'%''

and eh.Sup\_Name Like '''+@strCSRSupin+'%''

and eh.Mgr\_Name Like '''+@strCSRMgrin+'%''

and ((S.Status = '''+@strFormStatus+''') or (S.Status = '''+@strFormStatus2+''') or (S.Status = '''+@strFormStatus3+'''))

Order By cl.SubmittedDate DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_MyPenSubmitted\_DashboardStaff

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--12. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_Dashboard]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_Dashboard'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_Dashboard]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the user's recent submitted records from the Coaching\_Log table and displayed on dashboard (includes completed)

-- Where the user's LAN is strSubmitter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/14> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_Dashboard]

@strUserin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[s].[Status] strFormStatus,

[eh].[Emp\_Name] strCSRName,

[eh].[Sup\_Name] strCSRSupName,

[eh].[Mgr\_Name] strCSRMgrName,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strUserin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order By [cl].[SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_Dashboard

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--13. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardMGR]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardMGR'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardMGR]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the Supervisor user's submitted records from the Coaching\_Log table and displayed on dashboard (includes completed)

-- Where the user's LAN is strSubmitter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/12/14> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardMGR]

@strUserin nvarchar(30),

@strCSRin nvarchar(30),

@strCSRSupin nvarchar(30),

@strCSRMgrin nvarchar(30),

@strStatusin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT cl.[FormName] strFormID,

s.[Status] strFormStatus,

eh.[Emp\_Name] strCSRName,

eh.[Sup\_Name] strCSRSupName,

eh.[Mgr\_Name] strCSRMgrName,

cl.[SubmittedDate] SubmittedDate

from EC.Coaching\_Log cl WITH(NOLOCK),

EC.Employee\_Hierarchy eh,

EC.DIM\_Status s

where cl.StatusID = s.StatusID

and cl.CSRID = eh.Emp\_ID

and cl.submitterID = (

select sh.emp\_ID

from EC.Employee\_Hierarchy sh

where sh.emp\_LanID = '''+@strUserin+''')

and eh.[Emp\_Name] LIKE '''+@strCSRin+'''

and eh.[Sup\_Name] LIKE '''+@strCSRSupin+'''

and eh.[Mgr\_Name] LIKE '''+@strCSRMgrin+'''

and s.[Status] LIKE '''+@strStatusin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order by cl.[SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardMGR

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--14. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardSUP]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardSUP'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardSUP]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the Supervisor user's submitted records from the Coaching\_Log table and displayed on dashboard (includes completed)

-- Where the user's LAN is strSubmitter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/12/14> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardSUP]

@strUserin nvarchar(30),

@strCSRin nvarchar(30),

@strCSRSupin nvarchar(30),

@strCSRMgrin nvarchar(30),

@strStatusin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT cl.[FormName] strFormID,

s.[Status] strFormStatus,

eh.[Emp\_Name] strCSRName,

eh.[Sup\_Name] strCSRSupName,

eh.[Mgr\_Name] strCSRMgrName,

cl.[SubmittedDate] SubmittedDate

from EC.Coaching\_Log cl WITH(NOLOCK),

EC.Employee\_Hierarchy eh,

EC.DIM\_Status s

where cl.StatusID = s.StatusID

and cl.CSRID = eh.Emp\_ID

and cl.submitterID = (

select sh.emp\_ID

from EC.Employee\_Hierarchy sh

where sh.emp\_LanID = '''+@strUserin+''')

and eh.[Emp\_Name] LIKE '''+@strCSRin+'''

and eh.[Sup\_Name] LIKE '''+@strCSRSupin+'''

and eh.[Mgr\_Name] LIKE '''+@strCSRMgrin+'''

and s.[Status] LIKE '''+@strStatusin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order by cl.[SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_MySubmitted\_DashboardSUP

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--15. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPCSRCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_SUPCSRCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPCSRCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the SUP e-Coaching records from the Coaching\_Log table

-- Where the status is Completed.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/17/2014> - Removed Submitter name from query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPCSRCompleted]

@strSourcein nvarchar(100),

@strCSRSUPin nvarchar(30),

@strCSRin nvarchar(30),

@strCSRMGRin nvarchar(30),

@strSDatein datetime,

@strEDatein datetime

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strSDate nvarchar(8),

@strEDate nvarchar(8)

Set @strFormStatus = 'Completed'

Set @strSDate = convert(varchar(8),@strSDatein,112)

Set @strEDate = convert(varchar(8),@strEDatein,112)

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[eh].[Emp\_Name] strCSRName,

[eh].[Sup\_Name] strCSRSupName,

[eh].[Mgr\_Name] strCSRMgrName,

[s].[Status] strFormStatus,

[sc].[SubCoachingSource] strSource,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[DIM\_Source] sc,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and [cl].[SourceID] = [sc].[SourceID]

and [eh].[Sup\_LanID] = '''+@strCSRSUPin+'''

and [eh].[Mgr\_Name] Like '''+@strCSRMGRin+'''

and [S].[Status] = '''+@strFormStatus+'''

and [eh].[Emp\_Name] Like '''+@strCSRin+'''

and [sc].[SubCoachingSource] Like '''+@strSourcein+'''

and convert(varchar(8),[cl].[SubmittedDate],112) >= '''+@strSDate+'''

and convert(varchar(8),[cl].[SubmittedDate],112) <= '''+@strEDate+'''

Order By [cl].[SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_SUPCSRCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--16. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPCSRPending]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_SUPCSRPending'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPCSRPending]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <07/28/2013>

-- Description: \* This procedure selects the CSR e-Coaching records from the Coaching\_Log table

-- Where the status is Prnding Review.

-- Last Update: <3/14/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/17/2014> - Removed Submitter name from query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPCSRPending]

@strCSRSUPin nvarchar(30),

@strCSRin nvarchar(30),

@strSourcein nvarchar(100)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30)

Set @strFormStatus = 'Pending CSR Review'

Set @strFormStatus2 = 'Pending Manager Review'

Set @strFormStatus3 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[eh].[Emp\_Name] strCSRName,

[eh].[Sup\_Name] strCSRSupName,

[eh].[Mgr\_Name] strCSRMgrName,

[s].[Status] strFormStatus,

[sc].[SubCoachingSource] strSource,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[DIM\_Source] sc,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and [cl].[SourceID] = [sc].[SourceID]

and [eh].[Sup\_LanID] = '''+@strCSRSUPin+'''

and (([S].[Status] = '''+@strFormStatus+''') or ([S].[Status] = '''+@strFormStatus2+''') or ([S].[Status] = '''+@strFormStatus3+'''))

and [eh].[Emp\_Name] Like '''+@strCSRin+'''

and [sc].[SubCoachingSource] Like '''+@strSourcein+'''

Order By [eh].[Sup\_LanID],[cl].[SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_SUPCSRPending

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--17. Create SP [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPPending]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_Log\_SUPPending'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPPending]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <07/20/2013>

-- Description: \* This procedure selects the CSR e-Coaching records from the Coaching\_Log table

-- Where the status is Prnding Review.

-- Last Update: <03/12/2014> - Modified for eCoachingDev DB

-- Last Update: <03/17/2014> - Removed Submitter name from query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_Log\_SUPPending] @strCSRSUPin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30)

Set @strFormStatus = 'Pending Supervisor Review'

Set @strFormStatus2 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT [cl].[FormName] strFormID,

[eh].[Emp\_LanID] strCSR,

[eh].[Emp\_Name] strCSRName,

[eh].[Sup\_Name] strCSRSupName,

[S].[Status] strFormStatus,

[cl].[SubmittedDate] SubmittedDate

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and eh.[Sup\_LanID] = '''+@strCSRSUPin+'''

and ([S].[Status] = '''+@strFormStatus+''' OR [S].[Status] = '''+@strFormStatus2+''')

Order By [cl].[SubmittedDate] DESC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_Log\_SUPPending

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--18. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <4/30/12>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from completed e-Coaching records to display on Historical dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: 03/27/2014 - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted]

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.Emp\_Name CSR,

s.City strCSRSite

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Site] s,

[EC].[DIM\_Status] st,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.SiteID = s.SiteID

and cl.StatusID = st.StatusID

and st.Status <> '''+@strFormStatus+'''

Order By eh.Emp\_Name ASC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--19. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted2]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted2'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted2]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <4/30/12>

-- Last Update: 03/07/2014

-- Updated per SCR 12414 to add NOLOCK Hint

-- Last Update: 03/27/2014 - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted2]

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.Emp\_Name CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] st,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = st.StatusID

and st.Status <> '''+@strFormStatus+'''

Order By eh.Emp\_Name ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogDistinctCSRCompleted2

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--20. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <7/12/12>

-- Description: \* This procedure selects the distinct MGRs from completed e-Coaching records to display on Historical dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: 03/27/2014 - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted]

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.Mgr\_Name MGR,

s.City strCSRSite

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Site] s,

[EC].[DIM\_Status] st,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.SiteID = s.SiteID

and cl.StatusID = st.StatusID

and st.Status <> '''+@strFormStatus+'''

Order By eh.Mgr\_Name ASC'

EXEC (@nvcSQL)

END --sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--21. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted2]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted2'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted2]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <7/12/12>

-- Description: \* This procedure selects the distinct MGRs from completed e-Coaching records to display on Historical dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: 03/27/2014 - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted2]

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.Mgr\_Name MGR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] st,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = st.StatusID

and st.Status <> '''+@strFormStatus+'''

Order By eh.Mgr\_Name ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogDistinctMGRCompleted2

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--22. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <7/12/12>

-- Last Update: <>

-- Description: \* This procedure selects the distinct SUPs from completed e-Coaching records to display on Historical dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: 03/27/2014 - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted]

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.Sup\_Name SUP,

s.City strCSRSite

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Site] s,

[EC].[DIM\_Status] st,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.SiteID = s.SiteID

and cl.StatusID = st.StatusID

and st.Status <> '''+@strFormStatus+'''

Order By eh.Sup\_Name ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--23. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted2]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted2'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted2]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <7/12/12>

-- Last Update: <>

-- Description: \* This procedure selects the distinct SUPs from completed e-Coaching records to display on Historical dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: 03/27/2014 - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted2]

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.Sup\_Name SUP

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] st,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = st.StatusID

and st.Status <> '''+@strFormStatus+'''

Order By eh.Sup\_Name ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogDistinctSUPCompleted2

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--24. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSR]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctCSR'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSR]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/14/2014> - Changed filters, added additional Manager filter and 2 more status filters per Jourdain Augustin

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSR] @strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30)

Set @strFormStatus = 'Pending Manager Review'

Set @strFormStatus2 = 'Pending Supervisor Review'

Set @strFormStatus3 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Emp\_Name] CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and (([eh].[Mgr\_LanID] = '''+@strCSRMGRin+''' and [S].[Status] = '''+@strFormStatus+''') OR ([eh].[Sup\_LanID] = '''+@strCSRMGRin+''' and ([S].[Status] = '''+@strFormStatus2+''' or [S].[Status] = '''+@strFormStatus3+''')))

Order By [eh].[Emp\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctCSR

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--25. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/12/14> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT distinct [eh].[Emp\_Name] CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order By [eh].[Emp\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRSubmitted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--26. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeam]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeam'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeam]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/12/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeam]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30),

@strFormStatus4 nvarchar(30)

Set @strFormStatus = 'Pending CSR Review'

Set @strFormStatus2 = 'Pending Supervisor Review'

Set @strFormStatus3 = 'Pending Manager Review'

Set @strFormStatus4 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Emp\_Name] CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and eh.[Mgr\_LanID] = '''+@strCSRMGRin+'''

and (([S].[Status] = '''+@strFormStatus+''') or ([S].[Status] = '''+@strFormStatus2+''') or ([S].[Status] = '''+@strFormStatus3+''') or ([S].[Status] = '''+@strFormStatus4+'''))

Order By [eh].[Emp\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeam

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--27. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeamCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeamCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeamCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeamCompleted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Emp\_Name] CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and eh.[Mgr\_LanID] = '''+@strCSRMGRin+'''

and [S].[Status] = '''+@strFormStatus+'''

Order By [eh].[Emp\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctCSRTeamCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--28. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctMGRSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctMGRSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctMGRSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct supervisors from e-Coaching records to display on dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/12/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctMGRSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT distinct [eh].[Mgr\_Name] MGR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order By [eh].[Mgr\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctMGRSubmitted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-29. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUP]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctSUP'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUP]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/14/2014> - Changed filters, added additional Manager filter and 2 more status filters per Jourdain Augustin

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUP] @strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30)

Set @strFormStatus = 'Pending Manager Review'

Set @strFormStatus2 = 'Pending Supervisor Review'

Set @strFormStatus3 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Sup\_Name] SUP

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and (([eh].[Mgr\_LanID] = '''+@strCSRMGRin+''' and [S].[Status] = '''+@strFormStatus+''') OR ([eh].[Sup\_LanID] = '''+@strCSRMGRin+''' and ([S].[Status] = '''+@strFormStatus2+''' or [S].[Status] = '''+@strFormStatus3+''')))

Order By [eh].[Sup\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctSUP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--30. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct supervisors from e-Coaching records to display on dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/12/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT distinct [eh].[Sup\_Name] SUP

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order By [eh].[Sup\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPSubmitted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--31. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeam]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeam'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeam]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <07/30/2013>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeam]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30),

@strFormStatus4 nvarchar(30)

Set @strFormStatus = 'Pending CSR Review'

Set @strFormStatus2 = 'Pending Supervisor Review'

Set @strFormStatus3 = 'Pending Manager Review'

Set @strFormStatus4 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Sup\_Name] SUP

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and eh.[Mgr\_LanID] = '''+@strCSRMGRin+'''

and (([S].[Status] = '''+@strFormStatus+''') or ([S].[Status] = '''+@strFormStatus2+''') or ([S].[Status] = '''+@strFormStatus3+''') or ([S].[Status] = '''+@strFormStatus4+'''))

Order By [eh].[Sup\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeam

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--32. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeamCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeamCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeamCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/13/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeamCompleted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Sup\_Name] SUP

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and eh.[Mgr\_LanID] = '''+@strCSRMGRin+'''

and [S].[Status] = '''+@strFormStatus+'''

Order By [eh].[Sup\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogMgrDistinctSUPTeamCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--33. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedCSRSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedCSRSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedCSRSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on staff dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedCSRSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT distinct [eh].[Emp\_Name] CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order By [eh].[Emp\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedCSRSubmitted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--34. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedMGRSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedMGRSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedMGRSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct managers from e-Coaching records to display on staff dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedMGRSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT distinct [eh].[Mgr\_Name] MGR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order By [eh].[Mgr\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedMGRSubmitted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--35. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedSUPSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedSUPSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedSUPSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct Supervisors from e-Coaching records to display on staff dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <03/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedSUPSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT distinct [eh].[Sup\_Name] SUP

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

and s.[Status] <> '''+@strFormStatus+'''

Order By [eh].[Sup\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogStaffDistinctCompletedSUPSubmitted

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--36. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingCSRSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingCSRSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingCSRSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on staff dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingCSRSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30)

Set @strFormStatus = 'Completed'

Set @strFormStatus2 = 'Inactive'

SET @nvcSQL = 'SELECT distinct [eh].[Emp\_Name] CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

and s.[Status] <> '''+@strFormStatus+'''

AND S.Status <> '''+@strFormStatus2+'''

Order By [eh].[Emp\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingCSRSubmitted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--37. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingMGRSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingMGRSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingMGRSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct managers from e-Coaching records to display on staff dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingMGRSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30)

Set @strFormStatus = 'Completed'

Set @strFormStatus2 = 'Inactive'

SET @nvcSQL = 'SELECT distinct [eh].[Mgr\_Name] MGR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

AND S.Status <> '''+@strFormStatus+'''

AND S.Status <> '''+@strFormStatus2+'''

Order By [eh].[Mgr\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingMGRSubmitted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--38. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingSUPSubmitted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingSUPSubmitted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingSUPSubmitted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct Supervisors from e-Coaching records to display on staff dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingSUPSubmitted]

@strCSRMGRin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30)

Set @strFormStatus = 'Completed'

Set @strFormStatus2 = 'Inactive'

SET @nvcSQL = 'SELECT distinct [eh].[Sup\_Name] SUP

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.[SubmitterID] = sh.[Emp\_ID]

and sh.[Emp\_LanID] = '''+@strCSRMGRin+'''

AND S.Status <> '''+@strFormStatus+'''

AND S.Status <> '''+@strFormStatus2+'''

Order By [eh].[Sup\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogStaffDistinctPendingSUPSubmitted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--39. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSR]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogSupDistinctCSR'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSR]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSR] @strCSRSUPin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.[Emp\_Name] AS CSR

FROM [EC].[Coaching\_Log] cl WITH(NOLOCK),

[EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.SubmitterID = sh.emp\_ID

and sh.Emp\_LanID = '''+@strCSRSUPin+'''

and s.Status <> '''+@strFormStatus+'''

Order By CSR ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogSupDistinctCSR

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--40. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeam]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeam'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeam]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/12/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeam]

@strCSRSUPin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30),

@strFormStatus2 nvarchar(30),

@strFormStatus3 nvarchar(30)

Set @strFormStatus = 'Pending CSR Review'

Set @strFormStatus2 = 'Pending Manager Review'

Set @strFormStatus3 = 'Pending Acknowledgement'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Emp\_Name] CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and eh.[Sup\_LanID] = '''+@strCSRSUPin+'''

and (([S].[Status] = '''+@strFormStatus+''') or ([S].[Status] = '''+@strFormStatus2+''') or ([S].[Status] = '''+@strFormStatus3+'''))

Order By [eh].[Emp\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeam

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--41. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeamCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeamCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeamCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct CSRs from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/12/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeamCompleted]

@strCSRSUPin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Emp\_Name] CSR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and [eh].[Sup\_LanID] = '''+@strCSRSUPin+'''

and [S].[Status] = '''+@strFormStatus+'''

Order By [eh].[Emp\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogSupDistinctCSRTeamCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--42. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctMGR]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogSupDistinctMGR'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctMGR]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct supervisors from e-Coaching records to display on dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctMGR] @strCSRSUPin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.[Mgr\_Name] AS MGR

FROM [EC].[Coaching\_Log] cl WITH(NOLOCK),

[EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.SubmitterID = sh.Emp\_ID

and sh.Emp\_LanID = '''+@strCSRSUPin+'''

and s.Status <> '''+@strFormStatus+'''

Order By MGR ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogSupDistinctMGR

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--43. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctMGRTeamCompleted]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogSupDistinctMGRTeamCompleted'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctMGRTeamCompleted]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct managers for supervisors from e-Coaching records to display on dashboard for filter.

-- Last Update: <3/12/2014> - Modified for eCoachingDev DB and Updated per SCR 12369 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctMGRTeamCompleted]

@strCSRSUPin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Completed'

SET @nvcSQL = 'SELECT DISTINCT [eh].[Mgr\_Name] MGR

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and [eh].[Sup\_LanID] = '''+@strCSRSUPin+'''

and [S].[Status] = '''+@strFormStatus+'''

Order By [eh].[Mgr\_Name] ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogSupDistinctMGRTeamCompleted

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--44. Create SP [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctSUP]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Coaching\_LogSupDistinctSUP'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctSUP]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure selects the distinct supervisors from e-Coaching records to display on dashboard for filter.

-- Last Update: 03/07/2014

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/11/2014> - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Coaching\_LogSupDistinctSUP] @strCSRSUPin nvarchar(30)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus nvarchar(30)

Set @strFormStatus = 'Inactive'

SET @nvcSQL = 'SELECT DISTINCT eh.[Sup\_Name] AS SUP

FROM [EC].[Coaching\_Log] cl WITH(NOLOCK),

[EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s

where cl.CSRID = eh.Emp\_ID

and cl.StatusID = s.StatusID

and cl.SubmitterID = sh.Emp\_ID

and sh.Emp\_LanID = '''+@strCSRSUPin+'''

and s.Status <> '''+@strFormStatus+'''

Order By SUP ASC'

EXEC (@nvcSQL)

End --sp\_SelectFrom\_Coaching\_LogSupDistinctSUP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--45. Create SP [EC].[sp\_SelectReviewFrom\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectReviewFrom\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_SelectReviewFrom\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 11/16/11

-- Description: This procedure selects an e-Coaching record from the Coaching\_Log table for review.

-- Last Modified by: Susmitha Palacherla

-- Last Update: 03/05/2014

-- Updated per SCR 12359 to add NOLOCK Hint

-- Last Update: 03/24/2014 - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectReviewFrom\_Coaching\_Log] @strFormIDin nvarchar(50)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max)

SET @nvcSQL = 'SELECT cl.CoachingID numID,

cl.FormName strFormID,

sc.CoachingSource strFormType,

s.Status strFormStatus,

cl.EventDate EventDate,

cl.CoachingDate CoachingDate,

sh.Emp\_LanID strSubmitter,

sh.Emp\_Name strSubmitterName,

sh.Emp\_Email strSubmitterEmail,

cl.CSR strCSR,

eh.Emp\_Name strCSRName,

eh.Emp\_Email strCSREmail,

st.City strCSRSite,

eh.Sup\_LanID strCSRSup,

eh.Sup\_Name strCSRSupName,

eh.Sup\_Email strCSRSupEmail,

eh.Mgr\_LanID strCSRMgr,

eh.Mgr\_Name strCSRMgrName,

eh.Mgr\_Email strCSRMgrEmail,

sc.SubCoachingSource strSource,

CL.UCID strCID,

CL.isVerintID isVerintMonitor,

CL.VerintID strVerintID,

CL.isAvokeID isBehaviorAnalyticsMonitor,

CL.AvokeID strBehaviorAnalyticsID,

CL.isNGDActivityID isNGDActivityID,

CL.NGDActivityID strNGDActivityID,

CC.AHT,

CC.ARC "ARC Issue",

CC.Attendance,

CC.BCCPPI "BCC Process Procedure Issues",

CC.CCI "Current Coaching Initiative",

CC.CSE "Customer Service Escalation",

CC.Feedback,

CC.HR "HR Guideline Issues",

CC.OMR "OMR / Exceptions",

CC.Quality,

CC.Recognition,

CC.Schedule "Schedule Adherence",

CC.SecFloor "Secure Floor Violations",

CC.Other,

CL.Description txtDescription,

CL.CoachingNotes txtCoachingNotes,

CL.isVerified,

CL.SubmittedDate,

CL.StartDate,

CL.SupReviewedAutoDate,

CL.isCSE,

CL.MgrReviewManualDate,

CL.MgrReviewAutoDate,

CL.MgrNotes txtMgrNotes,

CL.isCSRAcknowledged,

CL.isCoachingRequired,

CL.CSRReviewAutoDate,

CL.CSRComments txtCSRComments

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[DIM\_Source] sc,

[EC].[DIM\_Site] st,

(SELECT ccl.FormName,

MAX(CASE WHEN [cr].[CoachingReason] = ''AHT'' THEN [clr].[Value] ELSE NULL END) AHT,

MAX(CASE WHEN [cr].[CoachingReason] = ''ARC Issue'' THEN [clr].[Value] ELSE NULL END) ARC,

MAX(CASE WHEN [cr].[CoachingReason] = ''Attendance'' THEN [clr].[Value] ELSE NULL END) Attendance,

MAX(CASE WHEN [cr].[CoachingReason] = ''BCC Process Procedure Issues'' THEN [clr].[Value] ELSE NULL END) BCCPPI,

MAX(CASE WHEN [cr].[CoachingReason] = ''Current Coaching Initiative'' THEN [clr].[Value] ELSE NULL END) CCI,

MAX(CASE WHEN [cr].[CoachingReason] = ''Customer Service Escalation'' THEN [clr].[Value] ELSE NULL END) CSE,

MAX(CASE WHEN [cr].[CoachingReason] = ''Feedback'' THEN [clr].[Value] ELSE NULL END) Feedback,

MAX(CASE WHEN [cr].[CoachingReason] = ''HR Guideline Issues'' THEN [clr].[Value] ELSE NULL END) HR,

MAX(CASE WHEN [cr].[CoachingReason] = ''OMR / Exceptions'' THEN [clr].[Value] ELSE NULL END) OMR,

MAX(CASE WHEN [cr].[CoachingReason] = ''Quality'' THEN [clr].[Value] ELSE NULL END) Quality,

MAX(CASE WHEN [cr].[CoachingReason] = ''Recognition'' THEN [clr].[Value] ELSE NULL END) Recognition,

MAX(CASE WHEN [cr].[CoachingReason] = ''Schedule Adherence'' THEN [clr].[Value] ELSE NULL END) Schedule,

MAX(CASE WHEN [cr].[CoachingReason] = ''Secure Floor Violations'' THEN [clr].[Value] ELSE NULL END) SecFloor,

MAX(CASE WHEN [cr].[CoachingReason] = ''Other'' THEN [clr].[Value] ELSE NULL END) Other

FROM [EC].[Coaching\_Log\_Reason] clr,

[EC].[DIM\_Coaching\_Reason] cr,

[EC].[Coaching\_Log] ccl WITH(NOLOCK)

WHERE [ccl].[FormName] = '''+@strFormIDin+'''

AND [clr].[CoachingReasonID] = [cr].[CoachingReasonID]

AND [ccl].[CoachingID] = [clr].[CoachingID]

GROUP BY ccl.FormName ) CC,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and [cl].[SourceID] = [sc].[SourceID]

and [cl].[SiteID] = [st].[SiteID]

and [cl].[SubmitterID] = [sh].[Emp\_ID]

and [cl].[FormName] = [CC].[FormName]

Order By [cl].[FormName]'

EXEC (@nvcSQL)

END --sp\_SelectReviewFrom\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--46. Create SP [EC].[sp\_Update1Review\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update1Review\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_Update1Review\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 11/16/12

-- Description: \* This procedure allows supervisors to update the e-Coaching records from review page.

-- Last Update: 03/04/2014

-- Updated per SCR 12359 to handle deadlocks with retries.

-- Last Update: 03/17/2014 - Modified for eCoachingDev DB

-- Last Update: 03/25/2014 - Modified Update query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Update1Review\_Coaching\_Log]

(

@nvcFormID Nvarchar(50),

@nvcFormStatus Nvarchar(30),

@dtmSupReviewedAutoDate datetime,

@dtmCoachingDate datetime,

@nvctxtCoachingNotes Nvarchar(max)

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

UPDATE [EC].[Coaching\_Log]

SET StatusID = (select StatusID from EC.DIM\_Status where status = @nvcFormStatus),

SupReviewedAutoDate = @dtmSupReviewedAutoDate,

CoachingDate = @dtmCoachingDate,

CoachingNotes = @nvctxtCoachingNotes

from EC.Coaching\_Log

WHERE FormName = @nvcFormID

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

END

END CATCH

END --sp\_Update1Review\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--47. Create SP [EC].[sp\_Update2Review\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update2Review\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_Update2Review\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 11/16/11

-- Description: \* This procedure allows managers to update the e-Coaching records from the review page with Yes, this is a confirmed Customer Service Escalation.

-- Last Update: 12/13/13

-- Updated per SCR xxxxx to handle deadlocks with retries.

-- Last Update: 03/17/2014 - Modified for eCoachingDev DB

-- Last Update: 03/25/2014 - Modified Update query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Update2Review\_Coaching\_Log]

(

@nvcFormID Nvarchar(50),

@nvcFormStatus Nvarchar(30),

@dtmSupReviewedAutoDate datetime,

@dtmCoachingDate datetime,

@bitisCSE bit,

@nvctxtCoachingNotes Nvarchar(max)

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

UPDATE [EC].[Coaching\_Log]

SET StatusID = (select StatusID from EC.DIM\_Status where status = @nvcFormStatus),

SupReviewedAutoDate = @dtmSupReviewedAutoDate,

CoachingDate = @dtmCoachingDate,

isCSE = @bitisCSE,

CoachingNotes = @nvctxtCoachingNotes

from EC.Coaching\_Log

WHERE FormName = @nvcFormID

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

END

END CATCH

END --sp\_Update2Review\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--48. Create SP [EC].[sp\_Update3Review\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update3Review\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_Update3Review\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 11/16/12

-- Description: This procedure allows managers to update the e-Coaching records from the review page with No, this is not a confirmed Customer Service Escalation.

-- Last Update: 03/04/2014

-- Updated per SCR 12359 to handle deadlocks with retries.

-- Last Update: 03/17/2014 - Modified for eCoachingDev DB

-- Last Update: 03/25/2014 - Modified Update query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Update3Review\_Coaching\_Log]

(

@nvcFormID Nvarchar(50),

@nvcFormStatus Nvarchar(30),

@dtmMgrReviewAutoDate datetime,

@dtmMgrReviewManualDate datetime,

@bitisCSE bit,

@nvcMgrNotes Nvarchar(max)

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

UPDATE [EC].[Coaching\_Log]

SET StatusID = (select StatusID from EC.DIM\_Status where status = @nvcFormStatus),

isCSE = @bitisCSE,

MgrReviewAutoDate = @dtmMgrReviewAutoDate,

MgrReviewManualDate = @dtmMgrReviewManualDate,

MgrNotes = @nvcMgrNotes

from EC.Coaching\_Log

WHERE FormName = @nvcFormID

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

END

END CATCH

END --sp\_Update3Review\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--49. Create SP [EC].[sp\_Update4Review\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update4Review\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_Update4Review\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 11/16/12

-- Description: This procedure allows csrs to update the e-Coaching records from the review page.

-- Last Update: 03/04/2014

-- Updated per SCR 12359 to handle deadlocks with retries.

-- Last Update: 03/17/2014 - Modified for eCoachingDev DB

-- Last Update: 03/25/2014 - Modified Update query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Update4Review\_Coaching\_Log]

(

@nvcFormID Nvarchar(50),

@nvcFormStatus Nvarchar(30),

@bitisCSRAcknowledged bit,

@nvcCSRComments Nvarchar(max),

@dtmCSRReviewAutoDate datetime

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

UPDATE [EC].[Coaching\_Log]

SET StatusID = (select StatusID from EC.DIM\_Status where status = @nvcFormStatus),

isCSRAcknowledged = @bitisCSRAcknowledged,

CSRReviewAutoDate = @dtmCSRReviewAutoDate,

CSRComments = @nvcCSRComments

from EC.Coaching\_Log

WHERE FormName = @nvcFormID

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

END

END CATCH

END --sp\_Update4Review\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--50. Create SP [EC].[sp\_Update5Review\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update5Review\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_Update5Review\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 11/16/12

-- Description: This procedure allows managers to update the e-Coaching records from the review page for Outlier records.

-- Last Update: 03/04/2014

-- Updated per SCR 12359 to handle deadlocks with retries.

-- Last Update: 03/27/2014 - Modified for eCoachingDev DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Update5Review\_Coaching\_Log]

(

@nvcFormID Nvarchar(50),

@nvcFormStatus Nvarchar(30),

@nvcstrReasonNotCoachable Nvarchar(30),

@dtmMgrReviewAutoDate datetime,

@dtmMgrReviewManualDate datetime,

@bitisCoachingRequired bit,

-- @nvcstrCoachReason\_Current\_Coaching\_Initiatives Nvarchar(30),

@nvcMgrNotes Nvarchar(max),

@nvctxtReasonNotCoachable Nvarchar(max)

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

UPDATE EC.Coaching\_Log

SET StatusID = (select StatusID from EC.DIM\_Status where status = @nvcFormStatus),

strReasonNotCoachable = @nvcstrReasonNotCoachable,

isCoachingRequired = @bitisCoachingRequired,

MgrReviewAutoDate = @dtmMgrReviewAutoDate,

MgrReviewManualDate = @dtmMgrReviewManualDate,

MgrNotes = @nvcMgrNotes,

txtReasonNotCoachable = @nvctxtReasonNotCoachable

WHERE FormName = @nvcFormID

OPTION (MAXDOP 1)

UPDATE EC.Coaching\_Log\_Reason

SET Value = (CASE WHEN @bitisCoachingRequired = 'True' then 'Opportunity' ELSE 'Not Coachable' END)

FROM EC.Coaching\_Log cl INNER JOIN EC.Coaching\_Log\_Reason clr

ON cl.CoachingID = clr.CoachingID

INNER JOIN EC.DIM\_Coaching\_Reason cr ON cr.CoachingReasonID = clr.CoachingReasonID

WHERE cl.FormName = @nvcFormID

and cr.CoachingReason = 'Current Coaching Initiative'

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

END

END CATCH

END --sp\_Update5Review\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--51. Create SP [EC].[sp\_Update6Review\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update6Review\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_Update6Review\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: 7/31/13

-- Description: \* This procedure allows csrs to update the e-Coaching records from the review page for Pending Acknowledgment records.

-- Last Update: 03/04/2014

-- Updated per SCR 12359 to handle deadlocks with retries.

-- Last Update: 03/17/2014 - Modified for eCoachingDev DB

-- Last Update: 03/25/2014 - Modified Update query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Update6Review\_Coaching\_Log]

(

@nvcFormID Nvarchar(50),

@nvcFormStatus Nvarchar(30),

@bitisCSRAcknowledged bit,

@dtmCSRReviewAutoDate datetime

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

UPDATE [EC].[Coaching\_Log]

SET StatusID = (select StatusID from EC.DIM\_Status where Status = @nvcFormStatus),

isCSRAcknowledged = @bitisCSRAcknowledged,

CSRReviewAutoDate = @dtmCSRReviewAutoDate

from EC.Coaching\_Log

WHERE FormName = @nvcFormID

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

END

END CATCH

END --sp\_Update6Review\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--52. Create SP [EC].[sp\_Update7Review\_Coaching\_Log]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update7Review\_Coaching\_Log'

)

DROP PROCEDURE [EC].[sp\_Update7Review\_Coaching\_Log]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: 7/31/13

-- Description: \* This procedure allows Sups to update the e-Coaching records from the review page for Pending Acknowledgment records.

-- Last Update: 03/04/2014

-- Updated per SCR 12359 to handle deadlocks with retries.

-- Last Update: 03/17/2014 - Modified for eCoachingDev DB

-- Last Update: 03/25/2014 - Modified Update query

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Update7Review\_Coaching\_Log]

(

@nvcFormID Nvarchar(50),

@nvcFormStatus Nvarchar(30),

@dtmSUPReviewAutoDate datetime

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

UPDATE [EC].[Coaching\_Log]

SET StatusID = (select StatusID from EC.DIM\_Status where status = @nvcFormStatus),

SUPReviewedAutoDate = @dtmSUPReviewAutoDate

from EC.Coaching\_Log

WHERE FormName = @nvcFormID

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

END

END CATCH

END --sp\_Update7Review\_Coaching\_Log

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--53. Create SP [EC].[sp\_Whoami]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Whoami'

)

DROP PROCEDURE [EC].[sp\_Whoami]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <7/22/13>

-- Last Update: <>

-- Description: \* This procedure selects the user's info from a table

-- Updated per SCR 12369 to add NOLOCK Hint

-- Last Update: <3/12/2014> - Updated per SCR 12369 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Whoami]

(

@strUserin Nvarchar(30)

)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max)

SET @nvcSQL = 'SELECT [Emp\_Job\_Code] + ''$'' + [Emp\_Email] + ''$'' + [Emp\_Name] as Submitter

FROM [EC].[Employee\_Hierarchy]WITH(NOLOCK)

WHERE [Emp\_LanID] = '''+@strUserin+''''

EXEC (@nvcSQL)

END --sp\_Whoami

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--54. Create SP [EC].[sp\_Whoisthis]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Whoisthis'

)

DROP PROCEDURE [EC].[sp\_Whoisthis]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <7/23/13>

-- Last Update: <>

-- Description: \* This procedure selects the CSR's hierarchy information from a table

--

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Whoisthis]

(

@strUserin Nvarchar(30)

)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max)

SET @nvcSQL = 'SELECT [Sup\_LanID] + ''$'' + [Mgr\_LanID] Flow

FROM [EC].[Employee\_Hierarchy]

WHERE [Emp\_LanID] = '''+@strUserin+''''

EXEC (@nvcSQL)

END --sp\_Whoisthis

GO

--Part 2: Employee Hierarchy RelatedStored Procedures

--1. PROCEDURE [EC].[sp\_Update\_Employee\_Hierarchy\_Stage]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update\_Employee\_Hierarchy\_Stage'

)

DROP PROCEDURE [EC].[sp\_Update\_Employee\_Hierarchy\_Stage]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_Update\_Employee\_Hierarchy\_Stage] Script Date: 03/12/2014 18:09:23 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 12/2/2013

-- Description: Performs the following actions.

-- Deletes records with missing Employee IDs

-- Removes Alpha characters from first 2 positions of Emp\_ID, Sup\_EMP\_ID, Mgr\_Emp\_ID

-- Upadtes CSR Sup ID values with the SUP from WFM

-- Deletes records with Missing SUP IDs

-- Populates Supervisor attributes

-- Populates Manager attributes

-- Last update: 03/05/2014

-- Updated per redesign

-- =============================================

CREATE PROCEDURE [EC].[sp\_Update\_Employee\_Hierarchy\_Stage]

AS

BEGIN

BEGIN

DELETE FROM [EC].[Employee\_Hierarchy\_Stage]

WHERE EMP\_ID = ' ' or EMP\_ID is NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Removes Alpha characters from first 2 positions of Emp\_ID, Sup\_EMP\_ID, Mgr\_Emp\_ID

BEGIN

UPDATE [EC].[Employee\_Hierarchy\_Stage]

SET [Emp\_ID]=[EC].[RemoveAlphaCharacters]([Emp\_ID]),

[Sup\_EMP\_ID]=[EC].[RemoveAlphaCharacters]([Sup\_EMP\_ID]),

[Mgr\_Emp\_ID]=[EC].[RemoveAlphaCharacters]([Mgr\_Emp\_ID]),

[Emp\_LanID]= REPLACE([Emp\_LanID], '#',''),

[Emp\_Email]= REPLACE([Emp\_Email], '#','')

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Set Sup\_Emp\_ID and Program for CSrs from WFM

BEGIN

UPDATE [EC].[Employee\_Hierarchy\_Stage]

SET [Sup\_Emp\_ID] = [Sup\_ID],

[Emp\_Program]=

CASE WHEN WFMSUP.[Emp\_Program]like 'FFM%'

THEN 'Marketplace' ELSE 'Medicare' END

FROM [EC].[EmpID\_To\_SupID\_Stage] WFMSUP JOIN [EC].[Employee\_Hierarchy\_Stage]INFO

ON LTRIM(WFMSUP.Emp\_ID) = LTRIM(INFO.Emp\_ID)

WHERE INFO.[Emp\_Job\_Code]in ('WACS01','WACS02','WACS03')

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Update Mgr\_Emp\_ID to be Supervisor's supervisor

BEGIN

UPDATE [EC].[Employee\_Hierarchy\_Stage]

SET Mgr\_Emp\_ID =[EC].[fn\_strMgrEmpIDFromEmpID] (Emp\_ID)

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- This will ensure that all users with missing sup id and mgr id will not be in the Employee\_Hierarchy table.

BEGIN

DELETE FROM [EC].[Employee\_Hierarchy\_Stage]

WHERE [Sup\_Emp\_ID] = ' ' or [Sup\_Emp\_ID] is NULL or [Mgr\_Emp\_ID] = ' ' or [Mgr\_Emp\_ID] is NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Populates Supervisor attributes

BEGIN

UPDATE Emp

SET [Sup\_Name]= Sup.[Emp\_Name],

[Sup\_Email]= Sup.[Emp\_Email],

[Sup\_Job\_Code]= Sup.[Emp\_Job\_Code],

[Sup\_Job\_Description]= Sup.[Emp\_Job\_Description],

[Sup\_LanID]= Sup.[Emp\_LanID]

FROM [EC].[Employee\_Hierarchy\_Stage] as Emp Join [EC].[Employee\_Hierarchy\_Stage]as Sup

ON LTRIM(Emp.[Sup\_Emp\_ID])= LTRIM(Sup.[EMP\_ID])

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Populates Manager attributes

BEGIN

UPDATE Emp

SET [Mgr\_Name]= Mgr.[Emp\_Name],

[Mgr\_Email]= Mgr.[Emp\_Email],

[Mgr\_Job\_Code]= Mgr.[Emp\_Job\_Code],

[Mgr\_Job\_Description]= Mgr.[Emp\_Job\_Description],

[Mgr\_LanID]= Mgr.[Emp\_LanID]

FROM [EC].[Employee\_Hierarchy\_Stage] as Emp Join [EC].[Employee\_Hierarchy\_Stage]as Mgr

ON LTRIM(Emp.[Mgr\_Emp\_ID])= LTRIM(Mgr.[EMP\_ID])

OPTION (MAXDOP 1)

END

END -- [EC].[sp\_Update\_Employee\_Hierarchy\_Stage]

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--2. PROCEDURE [EC].[sp\_Populate\_Employee\_Hierarchy]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Populate\_Employee\_Hierarchy'

)

DROP PROCEDURE [EC].[sp\_Populate\_Employee\_Hierarchy]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_Populate\_Employee\_Hierarchy] Script Date: 12/03/2013 10:33:11 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 07/25/2013

-- Description: Performs the following actions.

-- Updates existing records and Inserts New records from the Staging table.

-- Last Modified By: Susmitha Palacherla

-- Last Modified Date: 02/28/2014

-- updated to add start date for redesign project

-- =============================================

CREATE PROCEDURE [EC].[sp\_Populate\_Employee\_Hierarchy]

AS

BEGIN

-- Assigns End\_Date to Inactive Records

BEGIN

UPDATE [EC].[Employee\_Hierarchy]

SET [END\_DATE] = CONVERT(nvarchar(10),getdate(),112)

FROM [EC].[Employee\_Hierarchy\_Stage] S JOIN [EC].[Employee\_Hierarchy]H

ON H.Emp\_ID = S.Emp\_ID

AND S.Active in ('T', 'D')

AND H.END\_DATE= '99991231'

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Assigns Open Ended End\_Date for Rehire records

BEGIN

UPDATE [EC].[Employee\_Hierarchy]

SET [END\_DATE] = '99991231'

FROM [EC].[Employee\_Hierarchy\_Stage] S JOIN [EC].[Employee\_Hierarchy]H

ON H.Emp\_ID = S.Emp\_ID

AND S.Active not in ('T', 'D')

AND H.END\_DATE <> '99991231'

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Updates Existing Records

BEGIN

UPDATE [EC].[Employee\_Hierarchy]

SET [Emp\_Email] = S.Emp\_Email

,[Emp\_LanID] = S.Emp\_LanID

,[Emp\_Site] = [EC].[fn\_strSiteNameFromSiteLocation](S.Emp\_Site)

,[Emp\_Job\_Code] = S.Emp\_Job\_Code

,[Emp\_Job\_Description] = S.Emp\_Job\_Description

,[Emp\_Program] = S.Emp\_Program

,[Sup\_ID] = S.Sup\_EMP\_ID

,[Sup\_Name] = S.Sup\_Name

,[Sup\_Email] = S.Sup\_Email

,[Sup\_LanID] = S.Sup\_LanID

,[Sup\_Job\_Code] = S.Sup\_Job\_Code

,[Sup\_Job\_Description] = S.Sup\_Job\_Description

,[Mgr\_ID] = S.Mgr\_EMP\_ID

,[Mgr\_Name] = S.Mgr\_Name

,[Mgr\_Email] = S.Mgr\_Email

,[Mgr\_LanID] = S.Mgr\_LanID

,[Mgr\_Job\_Code] = S.Mgr\_Job\_Code

,[Mgr\_Job\_Description] = S.Mgr\_Job\_Description

,[Start\_Date] = CONVERT(nvarchar(8),S.[Start\_Date],112)

,[Active] = S.Active

FROM [EC].[Employee\_Hierarchy]H JOIN [EC].[Employee\_Hierarchy\_Stage]S

ON H.[Emp\_ID] = S.[EMP\_ID]

WHERE H.[Emp\_ID] is NOT NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Inserts New Records

BEGIN

INSERT INTO [EC].[Employee\_Hierarchy]

([Emp\_ID]

,[Emp\_Name]

,[Emp\_Email]

,[Emp\_LanID]

,[Emp\_Site]

,[Emp\_Job\_Code]

,[Emp\_Job\_Description]

,[Emp\_program]

,[Sup\_ID]

,[Sup\_Name]

,[Sup\_Email]

,[Sup\_LanID]

,[Sup\_Job\_Code]

,[Sup\_Job\_Description]

,[Mgr\_ID]

,[Mgr\_Name]

,[Mgr\_Email]

,[Mgr\_LanID]

,[Mgr\_Job\_Code]

,[Mgr\_Job\_Description]

,[Start\_Date]

,[Active]

)

SELECT S.[Emp\_ID]

,S.[Emp\_Name]

,S.[Emp\_Email]

,S.[Emp\_LanID]

,[EC].[fn\_strSiteNameFromSiteLocation](S.[Emp\_Site])

,S.[Emp\_Job\_Code]

,S.[Emp\_Job\_Description]

,S.[Emp\_Program]

,S.[Sup\_Emp\_ID]

,S.[Sup\_Name]

,S.[Sup\_Email]

,S.[Sup\_LanID]

,S.[Sup\_Job\_Code]

,S.[Sup\_Job\_Description]

,S.[Mgr\_Emp\_ID]

,S.[Mgr\_Name]

,S.[Mgr\_Email]

,S.[Mgr\_LanID]

,S.[Mgr\_Job\_Code]

,S.[Mgr\_Job\_Description]

,CONVERT(nvarchar(8),S.[Start\_Date],112)

,S.[Active]

FROM [EC].[Employee\_Hierarchy\_Stage]S Left outer Join [EC].[Employee\_Hierarchy]H

ON S.Emp\_ID = H.Emp\_ID

WHERE (H.EMP\_ID IS NULL and S.Emp\_ID <> '')

OPTION (MAXDOP 1)

END

END --sp\_Populate\_Employee\_Hierarchy

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--3. PROCEDURE [EC].[sp\_Update\_EmployeeID\_To\_LanID]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update\_EmployeeID\_To\_LanID'

)

DROP PROCEDURE [EC].[sp\_Update\_EmployeeID\_To\_LanID]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_Update\_EmployeeID\_To\_LanID] Script Date: 03/12/2014 18:14:38 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/03/2014

-- Description: Performs the following actions.

-- Adds an End Date to an Employee ID to lan ID combination that is different from the existing record.

-- Inserts new records for the changed and new combinations.

-- Last Modified By:

-- Last Modified Date:

-- =============================================

CREATE PROCEDURE [EC].[sp\_Update\_EmployeeID\_To\_LanID]

AS

BEGIN

DECLARE @dtNow DATETIME

SET @dtNow = GETDATE()

-- Assigns End\_Date to an Employee ID to Lan ID link that should no longer exist

BEGIN

UPDATE [EC].[EmployeeID\_To\_LanID]

SET [EndDate] = CONVERT(nvarchar(10),@dtNow,112),

[DatetimeLastUpdated]= @dtNow

FROM [EC].[Employee\_Hierarchy]EH JOIN [EC].[EmployeeID\_To\_LanID]LAN

ON EH.[Emp\_LanID]= LAN.[LanID]

AND EH.[Emp\_ID]= LAN.[EmpID]

WHERE LAN.[EndDate] = '99991231'

AND EH.[Active]in ('T', 'D')

OPTION (MAXDOP 1)

END

PRINT N'STEP1'

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Inserts links for new Employee ID to lan ID pair

BEGIN

INSERT INTO [EC].[EmployeeID\_To\_LanID]

([EmpID]

,[StartDate]

,[EndDate]

,[LanID]

,[DatetimeInserted]

,[DatetimeLastUpdated])

(SELECT

Emp\_ID,

Start\_Date,

End\_Date,

Emp\_LanID,

@dtNow ,

@dtNow

FROM [EC].[Employee\_Hierarchy]EH LEFT OUTER JOIN [EC].[EmployeeID\_To\_LanID]LAN

ON EH.[Emp\_LanID]= LAN.[LanID]

AND EH.[Emp\_ID]= LAN.[EmpID]

WHERE LAN.[EmpID]IS NULL AND LAN.[LanID]IS NULL

AND EH.[Emp\_LanID] IS NOT NULL

AND EH.[ACTIVE] NOT IN ('T','D'))

OPTION (MAXDOP 1)

END

PRINT N'STEP2'

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Inserts a new link for changed Employee ID to lan ID pair

BEGIN

--DECLARE @dtNow DATETIME

--SET @dtNow = GETDATE()

INSERT INTO [EC].[EmployeeID\_To\_LanID]

([EmpID]

,[StartDate]

,[EndDate]

,[LanID]

,[DatetimeInserted]

,[DatetimeLastUpdated])

(SELECT

Emp\_ID,

--CONVERT(nvarchar(10),@dtNow,112),

Start\_Date,

End\_Date,

Emp\_LanID,

@dtNow ,

@dtNow

FROM [EC].[Employee\_Hierarchy]EH LEFT OUTER JOIN [EC].[EmployeeID\_To\_LanID]LAN

ON EH.[Emp\_LanID]= LAN.[LanID]

AND EH.[Emp\_ID]= LAN.[EmpID]

WHERE LAN.[EmpID]IS NULL

AND EH.[Emp\_LanID] IS NOT NULL

AND EH.[ACTIVE] NOT IN ('T','D'))

OPTION (MAXDOP 1)

END

PRINT N'STEP3'

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Inserts a new link for a rehired Employee using the same lanid

BEGIN

--DECLARE @dtNow DATETIME

--SET @dtNow = GETDATE()

INSERT INTO [EC].[EmployeeID\_To\_LanID]

([EmpID]

,[StartDate]

,[EndDate]

,[LanID]

,[DatetimeInserted]

,[DatetimeLastUpdated])

(SELECT

Emp\_ID,

--CONVERT(nvarchar(10),@dtNow,112),

Start\_Date,

End\_Date,

Emp\_LanID,

@dtNow ,

@dtNow

FROM [EC].[Employee\_Hierarchy]EH JOIN [EC].[EmployeeID\_To\_LanID]LAN

ON EH.[Emp\_LanID]= LAN.[LanID]

AND EH.[Emp\_ID]= LAN.[EmpID]

WHERE LAN.[EndDate]<>99991231

AND EH.[Emp\_LanID] IS NOT NULL

AND EH.[ACTIVE] NOT IN ('T','D'))

OPTION (MAXDOP 1)

END

PRINT N'STEP4'

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

BEGIN

;With Overlaps (OverlaplanId, BaselanId)

As

(

Select Overlap.lanId, Base.lanId

From [EC].[EmployeeID\_To\_LanID] As Base

Inner Join [EC].[EmployeeID\_To\_LanID] As Overlap On Overlap.LanID = Base.lanid

Where (Overlap.StartDate > Base.StartDate)

And (Overlap.StartDate < Base.EndDate)

)

--select lan.[EmpID]

-- ,lan.[StartDate]

-- ,lan.[EndDate]

-- ,lan.[LanID]

-- from [eCoachingDev].[EC].[EmployeeID\_To\_LanID]lan join Overlaps o

--on lan.LanID = o.BaselanId

--This below sql will help us update the overlap rows to assign startdate of next row -1 to previous row

Update Base

set EndDate = [EC].[fn\_intDatetime\_to\_YYYYMMDD](DATEADD(DAY,-1,CONVERT(datetime,convert(char(8),Overlap.StartDate))))

From [EC].[EmployeeID\_To\_LanID] As Base

Inner Join [EC].[EmployeeID\_To\_LanID] As Overlap On Overlap.LanID = Base.lanid

Where (Overlap.StartDate > Base.StartDate)

And (Overlap.StartDate < Base.EndDate)

END

PRINT N'STEP5'

END --sp\_Update\_EmployeeID\_To\_LanID

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--4. PROCEDURE [EC].[sp\_Update\_CSR\_Hierarchy]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update\_CSR\_Hierarchy'

)

DROP PROCEDURE [EC].[sp\_Update\_CSR\_Hierarchy]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_Update\_CSR\_Hierarchy] Script Date: 03/12/2014 18:17:41 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 01/20/2014

-- Description: Performs the following actions.

-- Adds an End Date to an Employee record with a Hierarchy change.

-- Inserts a new row for the Upadted Hierarchy.

-- Last Modified By:

-- Last Modified Date:

-- =============================================

CREATE PROCEDURE [EC].[sp\_Update\_CSR\_Hierarchy]

AS

BEGIN

-- Assigns End\_Date to CSR records with changed Hierarchy.

BEGIN

UPDATE [EC].[CSR\_Hierarchy]

SET [EndDate] = DATEADD(dd, DATEDIFF(dd, 0, GETDATE()), 0)

FROM [EC].[Employee\_Hierarchy]EH JOIN [EC].[CSR\_Hierarchy]CH

ON EH.[Emp\_ID]= CH.[EmpID]

WHERE (EH.[Sup\_ID]<> CH.[SupID]OR EH.[Mgr\_ID]<> CH.[MgrID])

AND (EH.[Sup\_ID]IS NOT NULL AND EH.[Mgr\_ID] IS NOT NULL)

AND CH.[EndDate] = '9999-12-31 00:00:00.000'

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Inserts new rows for CSRs with changed Hierarchy.

BEGIN

;

With LatestRecord as

(Select [EmpID], max([EndDate])as LEnd\_Date from [EC].[CSR\_Hierarchy]

GROUP BY [EmpID])

INSERT INTO [EC].[CSR\_Hierarchy]

([EmpID]

,[SupID]

,[MgrID]

,[StartDate]

,[EndDate]

)

SELECT distinct EH.[Emp\_ID]

,EH.[Sup\_ID]

,EH.[Mgr\_ID]

, DATEADD(dd, DATEDIFF(dd, 0, GETDATE()), 0)

,'9999-12-31 00:00:00.000'

FROM [EC].[Employee\_Hierarchy]EH JOIN

(SELECT C.\* FROM [EC].[CSR\_Hierarchy] C JOIN LatestRecord L

ON C. EMPID =L.EMPID WHERE L.LEnd\_Date <> '9999-12-31 00:00:00.000') AS CH

on EH.Emp\_ID = CH.EmpID

where EH.[Emp\_Job\_Code]in ('WACS01','WACS02','WACS03')

AND (EH.[Sup\_ID]<> CH.[SupID]OR EH.[Mgr\_ID]<> CH.[MgrID])

AND (EH.[Sup\_ID]IS NOT NULL AND EH.[Mgr\_ID] IS NOT NULL)

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

-- Inserts New CSR records

BEGIN

INSERT INTO [EC].[CSR\_Hierarchy]

([EmpID]

,[SupID]

,[MgrID]

,[StartDate]

,[EndDate] )

SELECT EH.[Emp\_ID]

,EH.[Sup\_ID]

,EH.[Mgr\_ID]

, DATEADD(dd, DATEDIFF(dd, 0, GETDATE()), 0)

,'9999-12-31 00:00:00.000'

FROM [EC].[Employee\_Hierarchy]EH LEFT OUTER JOIN [EC].[CSR\_Hierarchy]CH

ON EH.[Emp\_ID]= CH.[EmpID]

WHERE CH.[EmpID]IS NULL

AND EH.[Emp\_Job\_Code]in ('WACS01','WACS02','WACS03')

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

END --sp\_Update\_CSR\_Hierarchy

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--5. PROCEDURE [EC].[sp\_InactivateCoachingLogsForTerms]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_InactivateCoachingLogsForTerms'

)

DROP PROCEDURE [EC].[sp\_InactivateCoachingLogsForTerms]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_InactivateCoachingLogsForTerms] Script Date: 03/12/2014 18:17:41 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 04/09/2014

-- Description: Inactivate Coaching logs for Termed Employees

-- Last Modified Date:

-- Last Updated By:

-- =============================================

CREATE PROCEDURE [EC].[sp\_InactivateCoachingLogsForTerms]

AS

BEGIN

-- Inactivate Coaching logs for Termed Employees

BEGIN

UPDATE [EC].[Coaching\_Log]

SET [StatusID] = 2

FROM [EC].[Coaching\_Log] C JOIN [EC].[Employee\_Hierarchy]H

ON C.[CSR] = H.[Emp\_LanID]

AND C.[CSRID] = H.[Emp\_ID]

WHERE CAST(H.[End\_Date] AS DATETIME)< GetDate()

AND H.[Active] in ('T','D')

AND H.[End\_Date]<> '99991231'

AND C.[StatusID] not in (1,2)

OPTION (MAXDOP 1)

END

END -- [EC].[sp\_InactivateCoachingLogsForTerms]

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--Part 3: Maintenance Stored Procedures

--1. Create SP [EC].[sp\_Update\_Migrated\_User\_Logs]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update\_Migrated\_User\_Logs'

)

DROP PROCEDURE [EC].[sp\_Update\_Migrated\_User\_Logs]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_Update\_Migrated\_User\_Logs] Script Date: 01/15/2014 20:15:39 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 04/05/2014

-- Description: Updates historical Coaching logs for Migrated users

-- =============================================

CREATE PROCEDURE [EC].[sp\_Update\_Migrated\_User\_Logs]

AS

BEGIN

-- Update CSR value in Coaching logs for migrated users

BEGIN

UPDATE [EC].[Coaching\_Log]

SET [CSR] = H.[Emp\_LanID]

FROM [EC].[Coaching\_Log]F JOIN [EC].[Employee\_Hierarchy]H

ON F.[CSRID] = H.[Emp\_ID]

WHERE F.[CSR] <> H.[Emp\_LanID]

OPTION (MAXDOP 1)

END

-- Update Lan ID value in Employee ID To Lan ID table for migrated users

BEGIN

UPDATE [EC].[EmployeeID\_To\_LanID]

SET [LANID] = H.[Emp\_LanID]

FROM [EC].[EmployeeID\_To\_LanID]L JOIN [EC].[Employee\_Hierarchy]H

ON L.[EMPID] = H.[Emp\_ID]

WHERE L.[LANID] <> H.[Emp\_LanID]

OPTION (MAXDOP 1)

END

END -- [EC].[sp\_Update\_Migrated\_User\_Logs]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-2. Create SP [EC].[sp\_SelectCoaching4Contact]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectCoaching4Contact'

)

DROP PROCEDURE [EC].[sp\_SelectCoaching4Contact]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: 6/10/13

-- Description: This procedure queries db for feed records to send out mail

-- Last Modified by: Susmitha Palacherla

-- Last Modified Date:9/19/2013

-- SCR 11051 to add numid to select list and numid length check to where clause.

-- Last Modified Date: 3/28/2013 - Adapted for new eCoaching DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectCoaching4Contact]

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max),

@strFormStatus1 nvarchar(30),

@strFormStatus2 nvarchar(30),

@strSource nvarchar(30),

@strSource2 nvarchar(30),

@strFormType nvarchar(30),

@strFormMail nvarchar (30)

Set @strFormStatus1 = 'Completed'

Set @strFormStatus2 = 'Inactive'

Set @strSource = 'OMR'

Set @strSource2 = 'IQS'

Set @strFormType = 'Indirect'

--Set @strFormMail = 'jourdain.augustin@gdit.com'

SET @nvcSQL = 'SELECT cl.CoachingID numID

,cl.FormName strFormID

,s.Status strFormStatus

,eh.Emp\_Email strCSREmail

,eh.Sup\_Email strCSRSupEmail

,eh.Mgr\_Email strCSRMgrEmail

,so.SubCoachingSource strSource

,eh.Emp\_Name strCSRName

,so.CoachingSource strFormType

,cl.SubmittedDate SubmittedDate

,cl.CoachingDate CoachingDate

,cl.EmailSent EmailSent

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[DIM\_Status] s,

[EC].[Coaching\_Log] cl,

[EC].[DIM\_Source] so,

[EC].[Coaching\_Log\_Reason] clr

WHERE cl.CSRID = eh.Emp\_ID

AND cl.StatusID = s.StatusID

AND cl.SourceID = so.SourceID

AND cl.CoachingID = clr.CoachingID

AND S.Status <> '''+@strFormStatus1+'''

AND S.Status <> '''+@strFormStatus2+'''

AND ([so].[SubCoachingSource] Like '''+@strSource+'%'' OR [so].[SubCoachingSource] LIKE '''+@strSource2+'%'')

AND cl.EmailSent = ''False''

AND ((so.CoachingSource =''Pending Acknowledgement'' and eh.Emp\_Email is NOT NULL and eh.Sup\_Email is NOT NULL)

OR (s.Status =''Pending Supervisor Review'' and eh.Sup\_Email is NOT NULL)

OR (s.Status =''Pending Manager Review'' and eh.Mgr\_Email is NOT NULL)

OR (s.Status =''Pending CSR Review'' and eh.Emp\_Email is NOT NULL))

AND LEN(cl.FormName) > 10

Order By cl.SubmittedDate DESC'

--and [strCSREmail] = '''+@strFormMail+'''

EXEC (@nvcSQL)

END -- sp\_SelectCoaching4Contact

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-3. Create SP [EC].[sp\_UpdateFeedMailSent]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_UpdateFeedMailSent'

)

DROP PROCEDURE [EC].[sp\_UpdateFeedMailSent]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: 08/23/13

-- Description: This procedure updates emailsent column to "True" for records from mail script.

-- Last Update: <>

-- Last Update: 03/27/2014 - Modified for redesigned DB

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_UpdateFeedMailSent]

(

@nvcNumID nvarchar(30)

)

AS

BEGIN

DECLARE

@sentValue nvarchar(30),

@intNumID int

SET @sentValue = 0 --1

SET @intNumID = CAST(@nvcNumID as INT)

UPDATE [EC].[Coaching\_Log]

SET EmailSent = @sentValue

WHERE CoachingID = @intNumID

END --sp\_UpdateFeedMailSent

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--4. Create SP [EC].[sp\_Check\_AgentRole]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Check\_AgentRole'

)

DROP PROCEDURE [EC].[sp\_Check\_AgentRole]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: <11/16/11>

-- Last Update: <>

-- Description: \* This procedure returns the Row\_ID from the ACl table if agent belongs to the role being checked.

-- Last Update: 03/12/2014 - Updated per SCR 12359 to add NOLOCK Hint

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Check\_AgentRole]

(

@nvcLANID Nvarchar(30),

@nvcRole Nvarchar(30)

)

AS

Declare

@ROWID INT

BEGIN

SELECT @ROWID = [Row\_ID]

FROM [EC].[Historical\_Dashboard\_ACL] WITH(NOLOCK)

WHERE [User\_LanID] = @nvcLANID

AND [Role]= @nvcRole

AND [End\_Date]='99991231'

IF @ROWID IS NULL RETURN 0

ELSE

RETURN @ROWID

END

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 4: DIM tables Stored Procedures

--1. Create SP [EC].[sp\_Dim\_Date\_Add\_Date\_Range]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Dim\_Date\_Add\_Date\_Range'

)

DROP PROCEDURE [EC].[sp\_Dim\_Date\_Add\_Date\_Range]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- CREATE date: 01/22/2014

-- Modified by:

-- Last modified date:

-- Description:

-- Given a begin and end date, populate table

-- AC.Dim\_Date for the given date range.

-- This is adapted from code found on 01/03/2012 at the

-- following URL:

-- http://arcanecode.com/2009/11/18/populating-a-kimball-date-dimension/

-- The following comments are from the original post at that

-- URL:

-- A few notes, this code does nothing to the existing table, no deletes

-- are triggered before hand. Because the DateKey is uniquely indexed,

-- it will simply produce errors if you attempt to insert duplicates.

-- You can however adjust the Begin/End dates and rerun to safely add

-- new dates to the table every year.

--

-- If the begin date is after the end date, no errors occur but nothing

-- happens as the while loop never executes.

-- =============================================

CREATE PROCEDURE [EC].[sp\_Dim\_Date\_Add\_Date\_Range]

@intBeginDate INT,

@intEndDate INT

AS

BEGIN

SET NOCOUNT ON -- turn off all the 1 row inserted messages

-- Hold our dates

DECLARE @BeginDate DATETIME

DECLARE @EndDate DATETIME

SET @BeginDate = EC.fn\_dtYYYYMMDD\_to\_Datetime(@intBeginDate)

SET @EndDate = EC.fn\_dtYYYYMMDD\_to\_Datetime(@intEndDate)

-- Holds a flag so we can determine if the date is the last day of month

DECLARE @LastDayOfMon CHAR(1)

-- Number of months to add to the date to get the current Fiscal date

DECLARE @FiscalYearMonthsOffset INT

-- These two counters are used in our loop.

DECLARE @DateCounter DATETIME --Current date in loop

DECLARE @FiscalCounter DATETIME --Fiscal Year Date in loop

-- Set this to the number of months to add to the current date to get

-- the beginning of the Fiscal year. For example, if the Fiscal year

-- begins July 1, put a 6 there.

-- Negative values are also allowed, thus if your 2010 Fiscal year

-- begins in July of 2009, put a -6.

SET @FiscalYearMonthsOffset = 6

-- Start the counter at the begin date

SET @DateCounter = @BeginDate

WHILE @DateCounter <= @EndDate

BEGIN

-- Calculate the current Fiscal date as an offset of

-- the current date in the loop

SET @FiscalCounter = DATEADD(m, @FiscalYearMonthsOffset, @DateCounter)

-- Set value for IsLastMonthDay

IF MONTH(@DateCounter) = MONTH(DATEADD(d, 1, @DateCounter))

SET @LastDayOfMon = 'N'

ELSE

SET @LastDayOfMon = 'Y'

-- add a record into the date dimension table for this date

INSERT INTO EC.DIM\_Date

(

[DateKey]

, [FullDate]

, [DateName]

, [DayOfWeek]

, [DayNameOfWeek]

, [DayOfMonth]

, [WeekOfYear]

, [MonthName]

, [MonthOfYear]

, [CalendarQuarter]

, [CalendarYear]

, [CalendarYearMonth]

, [CalendarYYYYQQ]

)

VALUES (

( YEAR(@DateCounter) \* 10000 ) + ( MONTH(@DateCounter) \* 100 )

+ DAY(@DateCounter) --DateKey

, @DateCounter -- FullDate

, CAST(DATEPART(mm, @DateCounter) AS NVARCHAR(2)) + '/'

+ CAST(DATEPART(dd, @DateCounter) AS NVARCHAR(2)) + '/'

+ CAST(YEAR(@DateCounter) AS NVARCHAR(4)) -- Modified DateName to match table

-- Below is [DateName] as downloaded:

-- , CAST(YEAR(@DateCounter) AS CHAR(4)) + '/'

-- + RIGHT('00' + RTRIM(CAST(DATEPART(mm, @DateCounter) AS CHAR(2))), 2) + '/'

-- + RIGHT('00' + RTRIM(CAST(DATEPART(dd, @DateCounter) AS CHAR(2))), 2) --DateName

, DATEPART(dw, @DateCounter) --DayOfWeek

, DATENAME(dw, @DateCounter) --DayNameOfWeek

, DATENAME(dd, @DateCounter) --DayOfMonth

, DATENAME(ww, @DateCounter) --WeekOfYear

, DATENAME(mm, @DateCounter) --MonthName

, MONTH(@DateCounter) --MonthOfYear

, DATENAME(qq, @DateCounter) --CalendarQuarter

, YEAR(@DateCounter) --CalendarYear

, CAST(YEAR(@DateCounter) AS CHAR(4)) + '-'

+ RIGHT('00' + RTRIM(CAST(DATEPART(mm, @DateCounter) AS CHAR(2))), 2) --CalendarYearMonth

, CAST(YEAR(@DateCounter) AS CHAR(4)) + 'Q' + DATENAME(qq, @DateCounter) --CalendarYYYYQQ

)

-- Increment the date counter for next pass thru the loop

SET @DateCounter = DATEADD(d, 1, @DateCounter)

END

END -- sp\_Dim\_Date\_Add\_Date\_Range

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--2. Create SP [EC].[sp\_Dim\_Date\_Add\_Unknown\_Row]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Dim\_Date\_Add\_Unknown\_Row'

)

DROP PROCEDURE [EC].[sp\_Dim\_Date\_Add\_Unknown\_Row]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- ALTER date: 01/22/2014

-- Modified by:

-- Last modified date:

-- Description:

-- Populate a row for an unknown value in dimension table

-- AC.Dim\_Date.

-- This is adapted from AJ Adams' code file named

-- "ACSchemaTableDDL.sql", dated 03/28/2011, and archived in

-- Version Manager at "cms\BCC Performance Mgmt\Alternate

-- Channels Scorecard\03\_Load\SQL", VM revision # 1.0.

-- =============================================

CREATE PROCEDURE [EC].[sp\_Dim\_Date\_Add\_Unknown\_Row]

AS

BEGIN

DELETE FROM EC.DIM\_Date

WHERE DateKey = -1

INSERT INTO EC.Dim\_Date (

[DateKey],

[FullDate],

[DateName],

[DayOfWeek],

[DayNameOfWeek],

[DayOfMonth],

[WeekOfYear],

[MonthName],

[MonthOfYear],

[CalendarQuarter],

[CalendarYear],

[CalendarYearMonth],

[CalendarYYYYQQ]

)

VALUES(-1, '1900-01-01 00:00:00.000', '1/1/1900', 1, 'Monday', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);

END -- sp\_Dim\_Date\_Add\_Unknown\_Row

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 5: Quality Load Stored Procedures

-- 1. Create Procedure [EC].[sp\_InsertInto\_Coaching\_Log\_Quality]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_InsertInto\_Coaching\_Log\_Quality'

)

DROP PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log\_Quality]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_InsertInto\_Coaching\_Log\_Quality] Script Date: 11/14/2013 14:59:42 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 02/23/2014

-- Description: This procedure inserts the Quality scorecards into the Coaching\_Log table.

-- The main attributes of the eCL are written to the Coaching\_Log table.

-- The Coaching Reasons are written to the Coaching\_Reasons Table.

--

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log\_Quality]

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

DECLARE @maxnumID INT

-- Fetches the maximum CoachingID before the insert.

SET @maxnumID = (SELECT IsNUll(MAX([CoachingID]), 0) FROM [EC].[Coaching\_Log])

-- Inserts records from the Quality\_Coaching\_Stage table to the Coaching\_Log Table

INSERT INTO [EC].[Coaching\_Log]

([FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[CSR]

,[CSRID]

,[SubmitterID]

,[EventDate]

,[isAvokeID]

,[isNGDActivityID]

,[isUCID]

,[isVerintID]

,[VerintID]

,[VerintEvalID]

,[Description]

,[SubmittedDate]

,[StartDate]

,[isCSE]

,[isCSRAcknowledged]

)

SELECT DISTINCT

lower(csr.Emp\_LanID) [FormName],

CASE qs.Program

WHEN NULL THEN csr.Emp\_Program

WHEN '' THEN csr.Emp\_Program

ELSE qs.Program END [ProgramName],

211 [SourceID],

[EC].[fn\_strStatusIDFromIQSEvalID](qs.CSE, qs.Oppor\_Rein )[StatusID],

[EC].[fn\_intSiteIDFromEmpID](LTRIM(qs.User\_EMPID))[SiteID],

lower(csr.Emp\_LanID) [CSR],

qs.User\_EMPID [CSRID],

qs.Evaluator\_ID [SubmitterID],

qs.Call\_Date [EventDate],

0 [isAvokeID],

0 [isNGDActivityID],

0 [isUCID],

1 [isVerintID],

qs.Journal\_ID [VerintID],

qs.Eval\_ID [VerintEvalID],

EC.fn\_nvcHtmlEncode(qs.Summary\_CallerIssues)[Description],

GetDate() [SubmittedDate],

qs.Eval\_Date [StartDate],

CASE WHEN qs.CSE = '' THEN 0

ELSE 1 END [isCSE],

0 [isCSRAcknowledged]

FROM [EC].[Quality\_Coaching\_Stage] qs

join EC.Employee\_Hierarchy csr on qs.User\_EMPID = csr.Emp\_ID

left outer join EC.Coaching\_Log cf on qs.[Eval\_ID] = cf.[VerintEvalID]

where cf.[VerintEvalID] is null

OPTION (MAXDOP 1)

WAITFOR DELAY '00:00:00:05' -- Wait for 5 ms

UPDATE [EC].[Coaching\_Log]

SET [FormName] = 'eCL-'+[FormName] +'-'+ convert(varchar,CoachingID)

where [FormName] not like 'eCL%'

OPTION (MAXDOP 1)

WAITFOR DELAY '00:00:00:05' -- Wait for 5 ms

-- Inserts records into Coaching\_Log\_reason table for each record inserted into Coaching\_log table.

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID]

,[CoachingReasonID]

,[SubCoachingReasonID]

,[Value])

SELECT cf.[CoachingID],

10,

42,

qs.[Oppor\_Rein]

FROM [EC].[Quality\_Coaching\_Stage] qs JOIN [EC].[Coaching\_Log] cf

ON qs.[Eval\_ID] = cf.[VerintEvalID] LEFT OUTER JOIN [EC].[Coaching\_Log\_Reason] cr

ON cf.[CoachingID] = cr.[CoachingID]

WHERE cr.[CoachingID] IS NULL

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SELECT @ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE()

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

IF ERROR\_NUMBER() IS NULL

RETURN 1

ELSE IF ERROR\_NUMBER() <> 0

RETURN ERROR\_NUMBER()

ELSE

RETURN 1

END CATCH

END -- sp\_InsertInto\_Coaching\_Log\_Quality

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- 2. Create Procedure [EC].[ sp\_Update\_Coaching\_Log\_Quality]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update\_Coaching\_Log\_Quality'

)

DROP PROCEDURE [EC].[sp\_Update\_Coaching\_Log\_Quality]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 04/23/2014

-- Description: This procedure updates the Quality scorecards into the Coaching\_Log table.

-- The txtdescription is updated in the Coaching\_Log table.

-- The updated Oppor/Re-In value is updated in the Coaching\_Reasons Table.

--

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_Update\_Coaching\_Log\_Quality]

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

-- Update txtDescription for existing records

UPDATE [EC].[Coaching\_Log]

SET [Description] = EC.fn\_nvcHtmlEncode(S.[Summary\_CallerIssues])

FROM [EC].[Quality\_Coaching\_Stage]S INNER JOIN [EC].[Coaching\_Log]F

ON S.[Eval\_ID] = F.[VerintEvalID]

AND S.[Journal\_ID] = F.[VerintID]

WHERE F.[VerintEvalID] is NOT NULL

OPTION (MAXDOP 1)

WAITFOR DELAY '00:00:00:05' -- Wait for 5 ms

-- Update Oppor/Re-In value in Coaching\_Log\_reason table for each record updated in Coaching\_log table.

UPDATE [EC].[Coaching\_Log\_reason]

SET [Value]= S.[Oppor\_Rein]

FROM [EC].[Quality\_Coaching\_Stage] S JOIN [EC].[Coaching\_Log] F

ON S.[Eval\_ID] = F.[VerintEvalID] JOIN [EC].[Coaching\_Log\_Reason] R

ON F.[CoachingID] = R.[CoachingID]

WHERE S.[Oppor\_Rein] <> [Value]

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SELECT @ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE()

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

IF ERROR\_NUMBER() IS NULL

RETURN 1

ELSE IF ERROR\_NUMBER() <> 0

RETURN ERROR\_NUMBER()

ELSE

RETURN 1

END CATCH

END -- sp\_Update\_Coaching\_Log\_Quality

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- 3. Create Procedure [EC].[ sp\_Update\_Quality\_Fact]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update\_Quality\_Fact'

)

DROP PROCEDURE [EC].[sp\_ Update\_Quality\_Fact]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 05/14/2014

-- Description: This procedure updates the existing records in the Quality Fact table

-- and inserts new records. =====================================================================

CREATE PROCEDURE [EC].[sp\_Update\_Quality\_Fact]

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

-- Update existing records in quality Fact Table

UPDATE [EC].[Quality\_Coaching\_Fact]

SET [Summary\_CallerIssues] = S.[Summary\_CallerIssues],

[Date\_Inserted]=S.[Date\_Inserted],

[Oppor\_Rein]= S.[Oppor\_Rein]

FROM [EC].[Quality\_Coaching\_Stage]S INNER JOIN [EC].[Quality\_Coaching\_Fact]F

ON S.[Eval\_ID] = F.[Eval\_ID]

AND S.[Journal\_ID] = F.[Journal\_ID]

WHERE F.[Eval\_ID] is NOT NULL

OPTION (MAXDOP 1)

WAITFOR DELAY '00:00:00:05' -- Wait for 5 ms

-- Append new records to Quality Fact Table

INSERT INTO [EC].[Quality\_Coaching\_Fact]

([Eval\_ID]

,[Eval\_Date]

,[Eval\_Site\_ID]

,[User\_ID]

,[User\_EMPID]

,[User\_LANID]

,[SUP\_ID]

,[SUP\_EMPID]

,[MGR\_ID]

,[MGR\_EMPID]

,[Journal\_ID]

,[Call\_Date]

,[Summary\_CallerIssues]

,[Coaching\_Goal\_Discussion]

,[CSE]

,[Evaluator\_ID]

,[Program]

,[Source]

,[Oppor\_Rein]

,[Date\_Inserted])

SELECT

S.[Eval\_ID]

,S.[Eval\_Date]

,S.[Eval\_Site\_ID]

,S.[User\_ID]

,S.[User\_EMPID]

,S.[User\_LANID]

,S.[SUP\_ID]

,S.[SUP\_EMPID]

,S.[MGR\_ID]

,S.[MGR\_EMPID]

,S.[Journal\_ID]

,S.[Call\_Date]

,S.[Summary\_CallerIssues]

,S.[Coaching\_Goal\_Discussion]

,S.[CSE]

,S.[Evaluator\_ID]

,S.[Program]

,S.[Source]

,S.[Oppor\_Rein]

,S.[Date\_Inserted]

FROM

[EC].[Quality\_Coaching\_Stage] S LEFT OUTER JOIN

[EC].[Quality\_Coaching\_Fact] F ON

S.[Eval\_ID] = F.[Eval\_ID]

WHERE(F.[Eval\_ID] IS NULL)

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SELECT @ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE()

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

IF ERROR\_NUMBER() IS NULL

RETURN 1

ELSE IF ERROR\_NUMBER() <> 0

RETURN ERROR\_NUMBER()

ELSE

RETURN 1

END CATCH

END -- sp\_Update\_Quality\_Fact

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 6: OMR Load Stored Procedures

--1. Create SP [EC].[sp\_InsertInto\_Coaching\_Log\_Outlier]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_InsertInto\_Coaching\_Log\_Outlier'

)

DROP PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log\_Outlier]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 03/10/2014

-- Last modified by: Susmitha Palacherla

-- Loads all records from [EC].[Outlier\_Coaching\_Stage]

-- =============================================

CREATE PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log\_Outlier]

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

DECLARE @maxnumID INT,

@dtmDate DATETIME

-- Fetches the maximum CoachingID before the insert.

SET @maxnumID = (SELECT IsNUll(MAX([CoachingID]), 0) FROM [EC].[Coaching\_Log])

-- Fetches the Date of the Insert

SET @dtmDate = GETDATE()

-- Inserts records from the Outlier\_Coaching\_Stage table to the Coaching\_Log Table

INSERT INTO [EC].[Coaching\_Log]

([FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[CSR]

,[CSRID]

,[SubmitterID]

,[EventDate]

,[isAvokeID]

,[isNGDActivityID]

,[isUCID]

,[isVerintID]

,[Description]

,[SubmittedDate]

,[StartDate]

,[isCSRAcknowledged]

,[numReportID]

,[strReportCode]

)

select Distinct LOWER(cs.CSR\_LANID) [FormName],

CASE cs.Program

WHEN NULL THEN csr.Emp\_Program

WHEN '' THEN csr.Emp\_Program

ELSE cs.Program END [ProgramName],

212 [SourceID],

[EC].[fn\_strStatusIDFromStatus](cs.Form\_Status)[StatusID],

[EC].[fn\_intGetSiteIDFromLanID](cs.CSR\_LANID,@dtmDate)[SiteID],

LOWER(cs.CSR\_LANID) [CSR],

cs.CSR\_EMPID [CSRID],

[EC].[fn\_nvcGetEmpIdFromLanId](LOWER(cs.Submitter\_LANID),@dtmDate)[SubmitterID],

cs.Event\_Date [EventDate],

0 [isAvokeID],

0 [isNGDActivityID],

0 [isUCID],

0 [isVerintID],

EC.fn\_nvcHtmlEncode(cs.TextDescription) [Description],

cs.Submitted\_Date SubmittedDate,

cs.Start\_Date [StartDate],

0 [isCSRAcknowledged],

cs.Report\_ID [numReportID],

cs.Report\_Code [strReportCode]

from [EC].[Outlier\_Coaching\_Stage] cs join EC.Employee\_Hierarchy csr on cs.CSR\_EMPID = csr.Emp\_ID

left outer join EC.Coaching\_Log cf on cs.Report\_ID = cf.numReportID and cs.Report\_Code = cf.strReportCode

where cf.numReportID is Null and cf.strReportCode is null

-- Updates the strFormID value

WAITFOR DELAY '00:00:00:05' -- Wait for 5 ms

UPDATE [EC].[Coaching\_Log]

SET [FormName] = 'eCL-'+[FormName] +'-'+ convert(varchar,CoachingID)

where [FormName] not like 'eCL%'

OPTION (MAXDOP 1)

WAITFOR DELAY '00:00:00:05' -- Wait for 5 ms

-- Inserts records into Coaching\_Log\_reason table for each record inserted into Coaching\_log table.

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID]

,[CoachingReasonID]

,[SubCoachingReasonID]

,[Value])

SELECT cf.[CoachingID],

5,

[EC].[fn\_intSubCoachReasonIDFromRptCode](SUBSTRING(cf.strReportCode,1,3)),

os.[CoachReason\_Current\_Coaching\_Initiatives]

FROM [EC].[Outlier\_Coaching\_Stage] os JOIN [EC].[Coaching\_Log] cf

ON os.[Report\_ID] = cf.[numReportID] AND os.[Report\_Code] = cf.[strReportCode]

LEFT OUTER JOIN [EC].[Coaching\_Log\_Reason] cr

ON cf.[CoachingID] = cr.[CoachingID]

WHERE cr.[CoachingID] IS NULL

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SELECT @ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE()

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

IF ERROR\_NUMBER() IS NULL

RETURN 1

ELSE IF ERROR\_NUMBER() <> 0

RETURN ERROR\_NUMBER()

ELSE

RETURN 1

END CATCH

END -- sp\_InsertInto\_Coaching\_Log\_Outlier

GO

--2. Create SP [EC].[sp\_InsertInto\_Outlier\_Rejected]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_InsertInto\_Outlier\_Rejected'

)

DROP PROCEDURE [EC].[sp\_InsertInto\_Outlier\_Rejected]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_InsertInto\_Outlier\_Rejected] Script Date: 01/18/2014 13:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 01/15/2014

-- Last modified by: Susmitha Palacherla

-- Description:

-- Loads all records from [EC].[Outlier\_Coaching\_Stage]Table

-- that dont have a corresponding CSR Lanid in the Employee\_Hierarchy table

-- Into tjhe Outliers rejected Table.

-- =============================================

CREATE PROCEDURE [EC].[sp\_InsertInto\_Outlier\_Rejected]

AS

BEGIN

-- Inserts records from the Outlier\_Coaching\_Stage table to the Outlier\_Rejected table.

INSERT INTO [EC].[Outlier\_Coaching\_Rejected]

([Report\_ID]

,[Report\_Code]

,[Form\_Type]

,[Source]

,[Form\_Status]

,[Event\_Date]

,[Submitted\_Date]

,[Start\_Date]

,[Submitter\_LANID]

,[Submitter\_Name]

,[Submitter\_Email]

,[CSR\_LANID]

,[CSR\_Site]

,[CoachReason\_Current\_Coaching\_Initiatives]

,[TextDescription]

,[FileName]

,[Rejected\_Reason]

,[Rejected\_Date])

SELECT S.[Report\_ID]

,S.[Report\_Code]

,S.[Form\_Type]

,S.[Source]

,S.[Form\_Status]

,S.[Event\_Date]

,S.[Submitted\_Date]

,S.[Start\_Date]

,S.[Submitter\_LANID]

,S.[Submitter\_Name]

,S.[Submitter\_Email]

,S.[CSR\_LANID]

,S.[CSR\_Site]

,S.[CoachReason\_Current\_Coaching\_Initiatives]

,S.[TextDescription]

,S.[FileName]

,'CSR\_LANID not found in Hierarchy Table'

,GETDATE()

FROM [EC].[Outlier\_Coaching\_Stage]S left outer join [EC].[Outlier\_Coaching\_Rejected] R

ON S.Report\_ID = R.Report\_ID and S.Report\_Code = R.Report\_Code and S.CSR\_LANID = R.CSR\_LANID

LEFT OUTER JOIN EC.[Employee\_Hierarchy]H

ON S.CSR\_LANID = H.Emp\_LanID

WHERE H.Emp\_LanID is NULL

AND R.Report\_ID is NULL and R.Report\_Code is NULL and R.CSR\_LANID is NULL

END -- sp\_InsertInto\_Outlier\_Rejected

GO

Part 7: Historical Dashboard Procedures

--1. PROCEDURE [EC].[sp\_InsertInto\_Historical\_Dashboard\_ACL]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_InsertInto\_Historical\_Dashboard\_ACL'

)

DROP PROCEDURE [EC].[sp\_InsertInto\_Historical\_Dashboard\_ACL]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_InsertInto\_Historical\_Dashboard\_ACL] Script Date: 11/07/2013 10:59:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 09/06/2012

-- Description: This procedure inserts user records into the Historical\_Dashboard\_ACL table.

-- Last Update: 10/18/2013

-- last Modified by: Susmitha Palacherla

-- Modified per SCR 10617 to removed hard coded authorized users and look at the Isadmin flag in the ACL Table.

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_InsertInto\_Historical\_Dashboard\_ACL]

(

@nvcACTION Nvarchar(10),

@nvcLANID Nvarchar(30),

@nvcUserLANID Nvarchar(30),

@nvcRole Nvarchar(30),

@nvcErrorMsgForEndUser Nvarchar(180) OUT

)

AS

BEGIN

DECLARE @nvcHierarchyLevel Nvarchar(20),

@nvcSQL Nvarchar(max),

@ROWID int,

@ENDDATE nvarchar(10),

@nvcIsAdmin Nvarchar(1)

SET @nvcErrorMsgForEndUser = N''

-- Removing the domain name from the Lanid.

SET @nvcLANID = SUBSTRING(@nvcLANID, CHARINDEX('\', @nvcLANID) + 1, LEN(@nvcLANID))

-- Checking the App Role of the User

SET @nvcIsAdmin = (SELECT CASE WHEN End\_Date = '99991231' THEN [ISADMIN] ELSE 'N' END

FROM [EC].[Historical\_Dashboard\_ACL]WHERE [User\_LanID] = @nvcLANID)

-- Checking if the Inserter is authorized to insert.

IF @nvcIsAdmin = 'Y'

BEGIN

IF @nvcACTION = 'ADD'

BEGIN

IF EXISTS (SELECT Emp\_LANID From [EC].[Employee\_Hierarchy] WHERE Emp\_LANID = @nvcUserLANID)

BEGIN

SELECT @ROWID = ROW\_ID from [EC].[Historical\_Dashboard\_ACL]

WHERE [User\_LanID]=@nvcUserLANID

SELECT @ENDDATE = End\_Date from [EC].[Historical\_Dashboard\_ACL]

WHERE [User\_LanID]=@nvcUserLANID

IF @ROWID IS NULL

BEGIN

INSERT INTO [EC].[Historical\_Dashboard\_ACL]

([User\_LanID]

,[User\_Name]

,[Role]

,[Updated\_By])

VALUES

(@nvcUserLanID ,EC.fn\_strUserName(@nvcUserLanID),

@nvcRole, @nvcLANID

)

SET @nvcErrorMsgForEndUser = N'Requested user ' + EC.fn\_strUserName(@nvcUserLanID) + N' successfully added.'

END --@ROWID IS NULL

ELSE

IF @ENDDATE = '99991231'

BEGIN

SET @nvcErrorMsgForEndUser = N'Requested user ' + EC.fn\_strUserName(@nvcUserLanID) + N' already exists in the system. You may select the existing record and update the Role.'

END --@ENDDATE = '99991231'

ELSE

IF @ENDDATE <> '99991231'

BEGIN

UPDATE [EC].[Historical\_Dashboard\_ACL]

SET End\_Date = '99991231' ,

[Role]= @nvcRole

WHERE [User\_LanID]=@nvcUserLANID

SET @nvcErrorMsgForEndUser = N'Requested user ' + EC.fn\_strUserName(@nvcUserLanID) + N' has been Re-activated as an ' + @nvcRole + N' user.'

END --@ENDDATE <> '99991231'

END

ELSE

BEGIN

SET @nvcErrorMsgForEndUser = N'Requested user ' + @nvcUserLANID + N' is not a valid user'

END --EXISTS @nvcUserLANID

END --@nvcACTION = 'ADD'

ELSE

IF @nvcACTION = 'REMOVE'

UPDATE [EC].[Historical\_Dashboard\_ACL]

SET [END\_DATE] = CONVERT(nvarchar(10),getdate(),112),

[Updated\_By] = @nvcLANID

Where User\_LanID = @nvcUserLANID

END --@nvcACTION = 'REMOVE'

ELSE

BEGIN

SET @nvcErrorMsgForEndUser = N'Requester ' + @nvclanid + N' is not authorized to ADD/REMOVE Records.'

END --@nvcIsAdmin = 'Y'

END --sp\_InsertInto\_Historical\_Dashboard\_ACL

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--2. PROCEDURE [EC].[sp\_SelectFrom\_Historical\_Dashboard\_ACL]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_SelectFrom\_Historical\_Dashboard\_ACL'

)

DROP PROCEDURE [EC].[sp\_SelectFrom\_Historical\_Dashboard\_ACL]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_SelectFrom\_Historical\_Dashboard\_ACL] Script Date: 10/10/2012 11:18:50 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 09/2012

-- Last Update: <>

-- Description: \* This procedure selects the user records from the Historical\_Dashboard\_ACL table

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_SelectFrom\_Historical\_Dashboard\_ACL]

(

@nvcRole Nvarchar(30)

)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max)

SET @nvcSQL = 'SELECT [Row\_ID],[User\_LanID],[User\_Name],[Role]

FROM [EC].[Historical\_Dashboard\_ACL]

where (Role = '''+@nvcRole+''' AND End\_Date > getdate())

Order by [User\_LanID]'

EXEC (@nvcSQL)

END -- [sp\_SelectFrom\_Historical\_Dashboard\_ACL]

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--3. PROCEDURE [EC].[sp\_UpdateHistorical\_Dashboard\_ACL]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_UpdateHistorical\_Dashboard\_ACL'

)

DROP PROCEDURE [EC].[sp\_UpdateHistorical\_Dashboard\_ACL]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_UpdateHistorical\_Dashboard\_ACL] Script Date: 10/10/2012 11:21:34 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: <09/20/12>

-- Last Update: <>

-- Description: \* This procedure allows supervisors to update the e-Coaching records from review page.

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_UpdateHistorical\_Dashboard\_ACL]

(

@nvcRowID Nvarchar(30),

@nvcLANID Nvarchar(30),

@nvcUserLANID Nvarchar(30),

@nvcRole Nvarchar(30)

--UNUSED PARAMETERS

-- @nvcRole VARCHAR(30) = NULL

)

AS

BEGIN

UPDATE [EC].[Historical\_Dashboard\_ACL]

SET [User\_LanID] = @nvcUserLANID, [Role] = @nvcRole, [User\_Name] = EC.fn\_strUserName(@nvcUserLANID),

[Updated\_By] = @nvcLANID

WHERE Row\_ID = @nvcRowID

END -- [sp\_UpdateHistorical\_Dashboard\_ACL]

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--4. PROCEDURE [EC].[sp\_DeleteFromHistoricalDashboardACL]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_DeleteFromHistoricalDashboardACL'

)

DROP PROCEDURE [EC].[sp\_DeleteFromHistoricalDashboardACL]

GO

/\*\*\*\*\*\* Object: StoredProcedure [EC].[sp\_DeleteFromHistoricalDashboardACL] Script Date: 11/01/2013 12:20:22 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Jourdain Augustin

-- Create Date: 09/19/2012

-- Description: Delete record from Historical dashboard ACL table

-- Last Update: 10/18/2013

-- Last Modified by: Susmitha Palacherla

-- Modified per SCR 10617 to removed hard coded authorized users and look at the IsAdmin flag in the ACL Table.

-- =====================================================================

CREATE PROCEDURE [EC].[sp\_DeleteFromHistoricalDashboardACL]

(

@nvcACTION Nvarchar(10),

@nvcLANID Nvarchar(30),

@nvcUserLANID Nvarchar(30),

@nvcRole Nvarchar(20) = NULL,

@nvcErrorMsgForEndUser Nvarchar(180) OUT

)

AS

BEGIN

DECLARE @nvcHierarchyLevel Nvarchar(20),

@nvcSQL Nvarchar(max),

@ROWID int,

@nvcIsAdmin Nvarchar(1)

SET @nvcErrorMsgForEndUser = N''

-- Removing the domain name from the Lanid.

SET @nvcLANID = SUBSTRING(@nvcLANID, CHARINDEX('\', @nvcLANID) + 1, LEN(@nvcLANID))

-- Checking the App Role of the User

SET @nvcIsAdmin = (SELECT CASE WHEN End\_Date = '99991231' THEN [ISADMIN] ELSE 'N' END

FROM [EC].[Historical\_Dashboard\_ACL]WHERE [User\_LanID] = @nvcLANID)

-- Checking if the User is authorized to Remove

IF @nvcIsAdmin = 'Y'

BEGIN

IF @nvcACTION = 'REMOVE'

UPDATE [EC].[Historical\_Dashboard\_ACL]

SET [END\_DATE] = CONVERT(nvarchar(10),getdate(),112),

[Updated\_By] = @nvcLANID

Where User\_LanID = @nvcUserLANID

ELSE

SET @nvcErrorMsgForEndUser = N'Action ' + @nvcACTION + N' is not an acceptable action.'

END

ELSE

BEGIN

SET @nvcErrorMsgForEndUser = N'Requester ' + @nvclanid + N' is not authorized to ADD/REMOVE Records.'

END

END --sp\_DeleteFromHistoricalDashboardACL

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Create Functions

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--1. FUNCTION [EC].[fn\_dtYYYYMMDD\_to\_Datetime]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_dtYYYYMMDD\_to\_Datetime'

)

DROP FUNCTION [EC].[fn\_dtYYYYMMDD\_to\_Datetime]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_dtYYYYMMDD\_to\_Datetime] Script Date: 03/13/2014 15:21:30 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 01/22/2014

-- Description: Given an integer representing a date in

-- format YYYYMMDD, return the corresponding datetime value.

-- =============================================

CREATE FUNCTION [EC].[fn\_dtYYYYMMDD\_to\_Datetime]

(

@intYYYYMMDD int

)

RETURNS datetime

AS

BEGIN

DECLARE

@dtYYYYMMDD datetime,

@strYYYYMMDD nchar(8)

SET @strYYYYMMDD = cast(@intYYYYMMDD as nchar(8))

SET @dtYYYYMMDD = cast(@strYYYYMMDD as datetime)

RETURN @dtYYYYMMDD

END -- fn\_dtYYYYMMDD\_to\_Datetime()

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--2. FUNCTION [EC].[fn\_intDatetime\_to\_YYYYMMDD]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_intDatetime\_to\_YYYYMMDD'

)

DROP FUNCTION [EC].[fn\_intDatetime\_to\_YYYYMMDD]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_intDatetime\_to\_YYYYMMDD] Script Date: 03/13/2014 15:24:50 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 02/05/2014

-- Description: Given a datetime value, return an integer

-- in format YYYYMMDD representing the date.

-- =============================================

CREATE FUNCTION [EC].[fn\_intDatetime\_to\_YYYYMMDD]

(

@dtDatetime datetime

)

RETURNS int

AS

BEGIN

DECLARE

@strYYYYMMDD nchar(8),

@intYYYYMMDD int

SET @strYYYYMMDD = convert(nchar(8), @dtDatetime, 112)

SET @intYYYYMMDD = cast(@strYYYYMMDD as int)

RETURN @intYYYYMMDD

END

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--3. DROP/RECREATE FUNCTION [EC].[fn\_intGetSiteIDFromLanID]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_intGetSiteIDFromLanID'

)

DROP FUNCTION [EC].[fn\_intGetSiteIDFromLanID]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_intGetSiteIDFromLanID] Script Date: 03/13/2014 15:26:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 03/05/2014

-- Description:

-- \* Given an LAN ID and a date, return the Employee ID of the

-- person who had that LAN ID on that date.

-- The Employee ID is then used to look up the site and site ID of the user.

-- =============================================

CREATE FUNCTION [EC].[fn\_intGetSiteIDFromLanID]

(

@nvcLanID NVARCHAR(20),

@dtmDate DATETIME

)

RETURNS INT

AS

BEGIN

DECLARE @nvcEmpID NVARCHAR(10),

@intDate INT,

@strSiteName NVARCHAR(30),

@intSiteID INT

SET @intDate = EC.fn\_intDatetime\_to\_YYYYMMDD (@dtmDate)

SET @nvcEmpID =

(

SELECT EmpID

FROM EC.EmployeeID\_To\_LanID

WHERE

LanID = @nvclanID AND

@intDate BETWEEN StartDate AND EndDate

)

IF @nvcEmpID is NULL

SET @nvcEmpID = '999999'

SELECT @strSiteName = [Emp\_Site]

FROM [EC].[Employee\_Hierarchy]

WHERE [Emp\_ID] = @nvcEmpID

IF @strSiteName is NULL OR @strSiteName ='OTHER'

SET @strSiteName= N'Unknown'

SELECT @intSiteID = [SiteID]

FROM [EC].[DIM\_Site]

WHERE [City] = @strSiteName

RETURN @intSiteID

END --[EC].[fn\_intGetSiteIDFromLanID]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--4. Drop/Recreate Function [EC].[fn\_intSiteIDFromEmpID]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_intSiteIDFromEmpID'

)

DROP FUNCTION [EC].[fn\_intSiteIDFromEmpID]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_intSiteIDFromEmpID] Script Date: 03/13/2014 21:17:42 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 02/23/2014

-- Description: Given an Employee ID returns the Employee site ID.

-- Last Modified by: Susmitha Palacherla

-- Last update:

-- =============================================

CREATE FUNCTION [EC].[fn\_intSiteIDFromEmpID]

(

@strEmpID nvarchar(10)

)

RETURNS INT

AS

BEGIN

DECLARE

@strSiteName nvarchar(30),

@intSITEID INT

SELECT @strSiteName = [Emp\_Site]

FROM [EC].[Employee\_Hierarchy]

WHERE [Emp\_ID] = @strEmpID

IF @strSiteName is NULL OR @strSiteName ='OTHER'

SET @strSiteName= N'Unknown'

SELECT @intSiteID = [SiteID]

FROM [EC].[DIM\_Site]

WHERE [City] = @strSiteName

RETURN @intSiteID

END --fn\_intSiteIDFromEmpID

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--5. Drop/Recreate Function [EC].[fn\_intSubCoachReasonIDFromRptCode]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_intSubCoachReasonIDFromRptCode'

)

DROP FUNCTION [EC].[fn\_intSubCoachReasonIDFromRptCode]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_intSubCoachReasonIDFromRptCode] Script Date: 03/13/2014 21:19:31 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 03/05/2014

-- Description: Given the 3 letter Report code returns the Sub coaching reason for the OMR log.

-- =============================================

CREATE FUNCTION [EC].[fn\_intSubCoachReasonIDFromRptCode] (

@strRptCode NVARCHAR(10)

)

RETURNS INT

AS

BEGIN

DECLARE @intSubCoachReasonID INT

IF @strRptCode IS NOT NULL

SET @intSubCoachReasonID =

CASE @strRptCode

WHEN N'CAN' THEN 20

WHEN N'DFQ' THEN 21

WHEN N'OPN' THEN 22

WHEN N'ISQ' THEN 23

WHEN N'OSC' THEN 24

WHEN N'ACW' THEN 25

WHEN N'AHT' THEN 26

WHEN N'SPI' THEN 27

WHEN N'ACO' THEN 28

WHEN N'IAE' THEN 29

WHEN N'IDE' THEN 30

WHEN N'IEE' THEN 31

WHEN N'INF' THEN 32

WHEN N'ISG' THEN 33

WHEN N'LCS' THEN 34

WHEN N'NIT' THEN 35

WHEN N'RME' THEN 36

WHEN N'SLG' THEN 37

WHEN N'TRN' THEN 38

ELSE -1

END

ELSE

SET @intSubCoachReasonID = -1

RETURN @intSubCoachReasonID

END -- fn\_intSubCoachReasonIDFromRptCode()

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--6. Drop/Recreate Function [EC].[fn\_nvcGetEmpIdFromLanId]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_nvcGetEmpIdFromLanId'

)

DROP FUNCTION [EC].[fn\_nvcGetEmpIdFromLanId]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create Date: 02/05/2014

-- Description:

-- \* Given an LAN ID and a date, return the Employee ID of the

-- person who had that LAN ID on that date.

-- =============================================

CREATE FUNCTION [EC].[fn\_nvcGetEmpIdFromLanId]

(

@nvcLanID Nvarchar(20),

@dtmDate Datetime

)

RETURNS nvarchar(10)

AS

BEGIN

DECLARE @nvcEmpID Nvarchar(10),

@intDate Int

SET @intDate = EC.fn\_intDatetime\_to\_YYYYMMDD (@dtmDate)

SET @nvcEmpID = (

SELECT EmpID

FROM EC.EmployeeID\_To\_LanID

WHERE

LanID = @nvclanID AND

@intDate BETWEEN StartDate AND EndDate

)

IF @nvcEmpID IS NULL

SET @nvcEmpID =

(SELECT DISTINCT Emp\_ID

FROM EC.Employee\_Hierarchy

WHERE Emp\_LanID = @nvclanID

AND @intDate BETWEEN Start\_Date AND End\_Date

)

IF @nvcEmpID IS NULL SET @nvcEmpID = '999999'

RETURN @nvcEmpID

END --[EC].[fn\_nvcGetEmpIdFromLanId]

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--7. Drop/Recreate Function [EC].[fn\_nvcHtmlEncode]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_nvcHtmlEncode'

)

DROP FUNCTION [EC].[fn\_nvcHtmlEncode]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 06/08/2012

-- Last update by: Susmitha Palacherla

-- Encodes special characters that can pose a security risk when sent to front end.

-- Last update: 09/22/2013

-- Description: Modified per SCR 11115 to accept Summary of Caller issues upto 7000 chars. had to use nvarchar(max)

-- =============================================CREATE FUNCTION [EC].[fn\_nvcHtmlEncode]

(

@UnEncoded as nvarchar(max)

)

RETURNS nvarchar(max)

AS

BEGIN

-- Declare the return variable here

DECLARE @Encoded as nvarchar(max)

-- Add the T-SQL statements to compute the return value here

SELECT @Encoded =

Replace(

Replace(

Replace(

Replace(

Replace(

Replace(

Replace(

Replace(

Replace(

Replace(

Replace(

Replace(@UnEncoded,'&','&amp;'),

'<', '&lt;'),

'>', '&gt;'),

'`', '&#39;'),

'"', '&quot;'),

'‘', '&lsquo;'),

'’', '&rsquo;'),

'′', '&prime;'),

'″', '&Prime;'),

'“', '&ldquo;'),

'”', '&rdquo;'),

'-', '&ndash;')

-- Return the result of the function

RETURN @Encoded

END

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--8. Drop/Recreate Function [EC].[fn\_strAddSpaceToName]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_strAddSpaceToName'

)

DROP FUNCTION [EC].[fn\_strAddSpaceToName]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_strAddSpaceToName] Script Date: 03/13/2014 21:23:39 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 07/30/2013

-- Last modified by:

-- Last modified date:

-- Description:

-- \* Given an Employee Name Adds a space after the comma.

-- =============================================

CREATE Function [EC].[fn\_strAddSpaceToName](@Name VarChar(50))

RETURNS VarChar(50)

AS

BEGIN

--SET @NAME = REPLACE(@NAME, ',' ,', ')

SET @NAME=

SUBSTRING(@NAME,1,(CHARINDEX(',',@NAME))) + ' ' + SUBSTRING(@NAME,(CHARINDEX(',',@NAME )+1),40)

RETURN @NAME

END -- fn\_strAddSpaceToName

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--9. Drop/Recreate Function [EC].[fn\_strMgrEmpIDFromEmpID]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_strMgrEmpIDFromEmpID'

)

DROP FUNCTION [EC].[fn\_strMgrEmpIDFromEmpID]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_strMgrEmpIDFromEmpID] Script Date: 03/13/2014 21:25:41 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 07/23/2013

-- Description: Given an Employee ID returns the Manager Employee ID.

-- Last Modified by: Susmitha Palacherla

-- Last update: 12/19/2013

-- Updated per SCR 11855 to use supervisors supervisor as Manager employee id.

-- =============================================

CREATE FUNCTION [EC].[fn\_strMgrEmpIDFromEmpID]

(

@strEmpID nvarchar(10)

)

RETURNS NVARCHAR(10)

AS

BEGIN

DECLARE

@strSupEmpID nvarchar(10),

@strMgrEmpID nvarchar(10)

SELECT @strSupEmpID = [Sup\_Emp\_ID]

FROM [EC].[Employee\_Hierarchy\_Stage]

WHERE [Emp\_ID] = @strEmpID

SELECT @strMgrEmpID =[Sup\_Emp\_ID]

FROM [EC].[Employee\_Hierarchy\_Stage]

WHERE [Emp\_ID] = @strSupEmpID

IF @strMgrEmpID IS NULL

SET @strMgrEmpID= N'999999'

RETURN @strMgrEmpID

END --fn\_strMgrEmpIDFromEmpID

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--10. Drop/Recreate Function [EC].[fn\_strSiteNameFromSiteLocation]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_strSiteNameFromSiteLocation'

)

DROP FUNCTION [EC].[fn\_strSiteNameFromSiteLocation]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_strSiteNameFromSiteLocation] Script Date: 03/13/2014 21:27:32 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 07/25/2013

-- Description: Given a site location returns the site name

-- Last modified by: Susmitha Palacherla

-- Last modified date: 12/09/2013

-- Added mapping for new sites per SCR 11629 to sue automated PS file. all non CCO call center locations set to OTHER.

-- =============================================

CREATE FUNCTION [EC].[fn\_strSiteNameFromSiteLocation] (

@strSiteLocation NVARCHAR(50)

)

RETURNS NVARCHAR(20)

AS

BEGIN

DECLARE @strSiteName NVARCHAR(20)

IF @strSiteLocation IS NOT NULL

SET @strSiteName =

CASE @strSiteLocation

WHEN N'AZ-Phoenix-8900 N 22nd Avenue' THEN N'Phoenix'

WHEN N'FL-Lynn Haven-1002 Arthur Dr' THEN N'Lynn Haven'

WHEN N'FL-Riverview-3020 US Hwy 301 S' THEN N'Riverview'

WHEN N'IA-Coralville-2400 Oakdale Blv' THEN N'Coralville'

WHEN N'IA-Coralville-2450 Oakdale Blv' THEN N'Coralville'

WHEN N'KS-Lawrence-3833 Greenway Dr' THEN N'Lawrence'

WHEN N'KY-Corbin-14892 N USHighway25E' THEN N'Corbin'

WHEN N'KY-London-4550 Old Whitley Rd' THEN N'London'

WHEN N'KY-Winchester-1025 Bypass Rd' THEN N'Winchester'

WHEN N'LA-Bogalusa-411 IndustrialPkwy' THEN N'Bogalusa'

WHEN N'MS-Hattiesburg-5912 Highway 49' THEN N'Hattiesburg'

WHEN N'TX-Houston-5959 Corporate Dr' THEN N'Houston'

WHEN N'TX-Waco-1205 N Loop 340' THEN N'Waco'

WHEN N'UT-Layton-2195 N Univ Pk Blvd' THEN 'Layton'

WHEN N'UT-Sandy-8475 S Sandy Parkway' THEN N'Sandy'

WHEN N'VA-Chester-701 Liberty Way' THEN N'Chester'

WHEN N'VA-Arlington-3330 N Washington' THEN N'Arlington'

ELSE 'OTHER'

END

ELSE

SET @strSiteName = N'Unknown'

--IF @strSiteName like '%HOME%'

-- SET @strSiteName = 'Other'

RETURN @strSiteName

END -- fn\_strSiteNameFromSiteLocation()

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--11. Drop/Recreate Function [EC].[fn\_strStatusIDFromIQSEvalID]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_strStatusIDFromIQSEvalID'

)

DROP FUNCTION [EC].[fn\_strStatusIDFromIQSEvalID]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_strStatusIDFromIQSEvalID] Script Date: 03/13/2014 21:29:47 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 02/23/2014

-- Last modified by:

-- Last modified date:

-- Description:

-- \* Given an IQS Eval ID determines the Status for the Coaching Log.

-- The Status is then used to look up the Status ID.

-- =============================================

CREATE FUNCTION [EC].[fn\_strStatusIDFromIQSEvalID] (

@strCSE NVARCHAR(20),@strOppor\_Rein NVARCHAR(20)

)

RETURNS NVARCHAR(50)

AS

BEGIN

DECLARE @strStatus NVARCHAR(50),

@strStatusID INT

IF @strCSE ='' and @strOppor\_Rein in ('Opportunity','Opportunity-PWC','Did Not Meet Goal')

SET @strStatus = 'Pending Supervisor Review'

ELSE

IF @strCSE in ('1','2','3','4','5','6','7','8') and @strOppor\_Rein in ('Opportunity','Opportunity-PWC','Did Not Meet Goal')

SET @strStatus = 'Pending Manager Review'

ELSE

IF @strOppor\_Rein in ('Reinforcement','Met Goal')

SET @strStatus= 'Pending Acknowledgement'

IF @strStatus is NULL

SET @strStatus = 'Unknown'

SELECT @strStatusID = [StatusID] FROM [EC].[DIM\_Status]

WHERE [Status]= @strStatus

RETURN @strStatusID

END -- fn\_strStatusIDFromIQSEvalID()

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--12. Drop/Recreate Function [EC].[fn\_strStatusIDFromStatus]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_strStatusIDFromStatus'

)

DROP FUNCTION [EC].[fn\_strStatusIDFromStatus]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_strStatusIDFromStatus] Script Date: 03/13/2014 21:31:00 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 03/06/2014

-- Last modified by:

-- Last modified date:

-- Description: Given a Status returns the Status id from Status table.

--

-- =============================================

CREATE FUNCTION [EC].[fn\_strStatusIDFromStatus]

(

@strStatus NVARCHAR(30)

)

RETURNS INT

AS

BEGIN

DECLARE @strStatusID INT

SELECT @strStatusID = [StatusID] FROM [EC].[DIM\_Status]

WHERE [Status]= @strStatus

RETURN @strStatusID

END -- fn\_strStatusIDFromStatus()

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--13. Drop/Recreate Function [EC].[fn\_strUserName]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_strUserName'

)

DROP FUNCTION [EC].[fn\_strUserName]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_strUserName] Script Date: 03/13/2014 21:32:19 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 09/21/2012

-- Description: Given a LAN ID, fetches the User Name from the Employee Hierarchy table.

-- If no match is found returns 'Unknown'

-- Last Modified date: 09/18/2013

-- Modified By: Susmitha Palacherla

-- Modified to use Employee\_Hierarchy Table

-- =============================================

CREATE FUNCTION [EC].[fn\_strUserName]

(

@strUserLanId nvarchar(30) -- LAN ID of person requesting CSR scorecard

)

RETURNS NVARCHAR(30)

AS

BEGIN

DECLARE

@strUserName nvarchar(30)

-- Strip domain off of the @strRequesterLanId parameter.

--SET @strUserLanId = RTRIM(SUBSTRING(@strUserLanId, CHARINDEX(N'\', @strUserLanId) + 1, 100))

SET @strUserLanId = SUBSTRING(@strUserLanId, CHARINDEX('\', @strUserLanId) + 1, LEN(@strUserLanId))

SELECT @strUserName = Emp\_Name

FROM [EC].[Employee\_Hierarchy]

WHERE Emp\_LanID = @strUserLanId

IF @strUserName IS NULL

SET @strUserName = N'UnKnown'

RETURN @strUserName

END

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--14. FUNCTION [EC].[RemoveAlphaCharacters]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'RemoveAlphaCharacters'

)

DROP FUNCTION [EC].[RemoveAlphaCharacters]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[RemoveAlphaCharacters] Script Date: 12/03/2013 11:04:08 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 07/20/2013

-- Last modified by:

-- Last modified date:

-- Description:

-- \* Given an Employee ID or Other String removes the alpha characters

-- If they exist in the first 2 Positions.

-- =============================================

CREATE Function [EC].[RemoveAlphaCharacters](@Temp VarChar(10))

Returns VarChar(10)

AS

Begin

Declare @RemoveValues as varchar(10) = '%[A-Za-z]%'

While PatIndex(@RemoveValues, @Temp) > 0 and PatIndex(@RemoveValues, @Temp) <= 2

Set @Temp = Stuff(@Temp, PatIndex(@RemoveValues, @Temp), 1, '')

Return @Temp

End -- RemoveAlphaCharacters

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--15. FUNCTION [EC].[fn\_intSiteIDFromSite]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_intSiteIDFromSite'

)

DROP FUNCTION [EC].[fn\_intSiteIDFromSite]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_intSiteIDFromSite] Script Date: 04/09/2014 09:29:35 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 03/05/2014

-- Description: Given the Site Name returns the Site ID.

-- =============================================

CREATE FUNCTION [EC].[fn\_intSiteIDFromSite] (

@strSite NVARCHAR(20)

)

RETURNS INT

AS

BEGIN

DECLARE @intSiteID INT

IF @strSite IS NOT NULL

SET @intSiteID =

CASE @strSite

WHEN N'Bogalusa' THEN 1

WHEN N'Boise' THEN 2

WHEN N'Brownsville' THEN 3

WHEN N'Chester' THEN 4

WHEN N'Coralville' THEN 5

WHEN N'Corbin' THEN 6

WHEN N'Hattiesburg' THEN 7

WHEN N'Houston' THEN 8

WHEN N'London' THEN 9

WHEN N'Lawrence' THEN 10

WHEN N'Layton' THEN 11

WHEN N'Lynn Haven' THEN 12

WHEN N'Pearl' THEN 13

WHEN N'Phoenix' THEN 14

WHEN N'Riverview' THEN 15

WHEN N'Sandy' THEN 16

WHEN N'Waco' THEN 17

WHEN N'Winchester' THEN 18

WHEN N'Arlington' THEN 19

ELSE -1 END

ELSE

SET @intSiteID = -1

RETURN @intSiteID

END -- fn\_intSiteIDFromSite()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--16. FUNCTION [EC].[fn\_intSourceIDFromOldSource]

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'fn\_intSourceIDFromOldSource'

)

DROP FUNCTION [EC].[fn\_intSourceIDFromOldSource]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [EC].[fn\_intSourceIDFromOldSource] Script Date: 04/09/2014 09:31:18 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 03/05/2014

-- Description: Given the Old Form Type and Source value returns the Source ID.

-- =============================================

CREATE FUNCTION [EC].[fn\_intSourceIDFromOldSource] (

@strFormType NVARCHAR(20), @strSource NVARCHAR(60)

)

RETURNS INT

AS

BEGIN

DECLARE @intSourceID INT

SET @intSourceID = ( SELECT [SourceID]

FROM [EC].[Historical\_Source]

WHERE [strFormType]=@strFormType

AND[strSource]= @strSource)

IF @intSourceID IS NULL SET @intSourceID =213

RETURN @intSourceID

END -- fn\_intSourceIDFromOldSource()

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Import Employee Hierarchy Data from Old Production Coaching DB

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This step will be used to copy Employee Hierarchy and ACL table data from the current production coaching database to the new eCoaching Database.

Step1: We will be using the Import Data task wizard. Data will be imported for the following tables.

1. [EC].[Employee\_Hierarchy]
2. [EC].[Historical\_Dashboard\_ACL]

Login to Instance VDENSSDBP07\SCOPRP01

Right Click on Database eCoaching 🡪Tasks🡪Import Data

Source is VDENSSDBP07\SCORP01 and Source Database is Coaching

Destination is VDENSSDBP07\SCORP01 and Destination DB is **eCoaching**

Copy Data from One or More Tables

Select [EC].[Employee\_Hierarchy] in both Source and Destination panes

Select [EC].[ Historical\_Dashboard\_ACL] in both Source and Destination panes

For [EC].[ Historical\_Dashboard\_ACL] select Edit properties and Check for Enabling Identity Insert.

Run Immeditely and Finish.

Step2: Insert row for Unknown Employee in eCoaching DB in production.

INSERT INTO [EC].[Employee\_Hierarchy]

([Emp\_ID]

,[Emp\_Name]

,[Emp\_Email]

,[Emp\_LanID]

,[Emp\_Site]

,[Emp\_Job\_Code]

,[Emp\_Job\_Description]

,[Sup\_ID]

,[Sup\_Name]

,[Sup\_Email]

,[Sup\_LanID]

,[Sup\_Job\_Code]

,[Sup\_Job\_Description]

,[Mgr\_ID]

,[Mgr\_Name]

,[Mgr\_Email]

,[Mgr\_LanID]

,[Mgr\_Job\_Code]

,[Mgr\_Job\_Description]

,[Start\_Date]

,[End\_Date]

,[Active])

VALUES

('999999'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,'Unknown'

,20000101

,99991231

,'A')

GO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Import DIM and Supporting Table Data from eCoaching on Dev

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In this step we will be importing data from all the DIM and other supporting tables from the newly redesigned dev database.

Data will be imported for the following tables.

1. [EC].[DIM\_Coaching\_Reason]
2. [EC].[DIM\_Date]
3. [EC].[DIM\_Site]
4. [EC].[DIM\_Source]
5. [EC].[DIM\_Status]
6. [EC].[DIM\_Sub\_Coaching\_Reason]
7. [EC].[Historical\_Source]
8. [EC].[HistoricalMultiCodesForprogram]

Login to Instance VDENSSDBP07\SCOPRP01

Right Click on Database eCoaching 🡪Tasks🡪Import Data

Source is VRIVFSSDBT02\SCORD01,1437 and Source Database is eCoachingDev on dev

Destination is VDENSSDBP07\SCORP01 and Destination DB is **eCoaching on prod**

Copy Data from One or More Tables

Select the fllowing 8 tables in both the left and right panes.

1. [EC].[DIM\_Coaching\_Reason]
2. [EC].[DIM\_Date]
3. [EC].[DIM\_Site]
4. [EC].[DIM\_Source]
5. [EC].[DIM\_Status]
6. [EC].[DIM\_Sub\_Coaching\_Reason]
7. [EC].[Historical\_Source]
8. [EC].[HistoricalMultiCodesForprogram]

For the following tables select Edit properties and Check for Enabling Identity Insert.

1. [EC].[DIM\_Coaching\_Reason]
2. [EC].[DIM\_Site]
3. [EC].[DIM\_Status]
4. [EC].[DIM\_Sub\_Coaching\_Reason]

Run Immeditely and Finish.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Initial Populate of Employee\_ID\_TO\_Lan\_ID Table

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Fill the gaps for Start\_Date in employee\_Hierarchy Table from Scorecards DB
2. Fill the gaps for Start\_Date in employee\_Hierarchy Table from Verint
3. Assign default Start\_date for remaining gaps
4. Initial Insert into Employee\_ID\_To\_Lan\_ID Table
5. Handle overlaps of lan ID usage
6. Fill Gaps from Scorecard Table

\*Run each step at a time in the sql query window pointing to the specified database. Note that in some queries you might have to modify the association for the database accordingly.

NEXT 2 QUERIES ARE RUN AGAINST THE SCORECARDS DATABASE IN APPROPRIATE ENVIRONMENT

COACHING DB NAME IN QUERY REFERS TO PRODUCTION . UPDATE TO TO APPROPRIATE ENVIRONMENT IF RUNNING ELSEWHERE.

--1. Fill the gaps for Start\_Date in employee\_Hierarchy Table from Scorecards DB

;With StartDt as

(SELECT

[CsrEmployeeId]

,[CsrLanId]

,[Start\_Date]

,ROW\_NUMBER() OVER( PARTITION BY [CsrEmployeeId]

ORDER BY [Start\_Date]) AS Rn

FROM [bcc\_scrd].[HIERARCHY\_V02])

--SELECT \* FROM StartDt

--WHERE Rn=1

UPDATE [eCoaching].[EC].[Employee\_Hierarchy]

SET Start\_Date= sd.START\_DATE

FROM [eCoaching].[EC].[Employee\_Hierarchy]ec join StartDt sd

ON ec.Emp\_ID = sd.CsrEmployeeId

AND ec.Start\_Date is NULL

AND sd.Rn=1

--2. Fill the gaps for Start\_Date in employee\_Hierarchy Table from Verint

;With StartDt as

(SELECT

[Employee\_ID]

,CONVERT(nvarchar(10),[User\_Hire\_Date],112)as [User\_Hire\_Date]

,ROW\_NUMBER() OVER( PARTITION BY [Employee\_ID]

ORDER BY [User\_Hire\_Date]) AS Rn

FROM [bcc\_scrd].[Verint\_User\_Stage])

--SELECT \* FROM StartDt

--WHERE Rn=1

UPDATE[eCoaching].[EC].[Employee\_Hierarchy]

SET Start\_Date= sd.[User\_Hire\_Date]

FROM [eCoaching].[EC].[Employee\_Hierarchy]ec join StartDt sd

ON ec.Emp\_ID = sd.[Employee\_ID]

AND ec.Start\_Date is NULL

AND sd.Rn=1

NEXT 4 QUERIES (3 thru 6) RUN IN A WINDOW POINTING TO NEW COACHING DB eCoaching on prod

--3. Assign default Start\_date for remaining gaps

UPDATE [EC].[Employee\_Hierarchy]

SET Start\_Date= '20130101'

WHERE Start\_Date is NULL and end\_date >'20130101'

--4. Initial Insert into Employee\_ID\_To\_Lan\_ID Table

DECLARE @dtNow DATETIME

SET @dtNow = GETDATE()

INSERT INTO [EC].[EmployeeID\_To\_LanID]

([EmpID]

,[StartDate]

,[EndDate]

,[LanID]

,[DatetimeInserted]

,[DatetimeLastUpdated])

(SELECT

Emp\_ID,

--CONVERT(nvarchar(10),@dtNow,112),

Start\_Date,

End\_Date,

Emp\_LanID,

@dtNow ,

@dtNow

FROM [EC].[Employee\_Hierarchy]EH LEFT OUTER JOIN [EC].[EmployeeID\_To\_LanID]LAN

ON EH.[Emp\_LanID]= LAN.[LanID]

AND EH.[Emp\_ID]= LAN.[EmpID]

WHERE LAN.[EmpID]IS NULL AND LAN.[LanID]IS NULL

AND EH.[Emp\_LanID] IS NOT NULL)

--5. Handle overlaps of Lan ID usage

;With Overlaps (OverlaplanId, BaselanId)

As

(

Select Overlap.lanId, Base.lanId

From [EC].[EmployeeID\_To\_LanID] As Base

Inner Join [EC].[EmployeeID\_To\_LanID] As Overlap On Overlap.LanID = Base.lanid

Where (Overlap.StartDate > Base.StartDate)

And (Overlap.StartDate < Base.EndDate)

)

/\*

select lan.[EmpID]

,lan.[StartDate]

,lan.[EndDate]

,lan.[LanID]

from [eCoaching].[EC].[EmployeeID\_To\_LanID]lan join Overlaps o

on lan.LanID = o.BaselanId

\*/

--This below sql will be used to overcome the lan id overlaps by updating the overlap rows to assign startdate of next row -1 to previous row

Update Base

set EndDate = [EC].[fn\_intDatetime\_to\_YYYYMMDD](DATEADD(DAY,-1,CONVERT(datetime,convert(char(8),Overlap.StartDate))))

From [EC].[EmployeeID\_To\_LanID] As Base

Inner Join [EC].[EmployeeID\_To\_LanID] As Overlap On Overlap.LanID = Base.lanid

Where (Overlap.StartDate > Base.StartDate)

And (Overlap.StartDate < Base.EndDate)

SCORECARD DB in NEXT QUERY (6) BELOW IS FOR PRODUCTION.

IF RUNNING in DIFF ENV UPDATE ACCORDINGLY.

--DEV - [BCC\_Scorecards\_Dev]

--TEST- [BCC\_Scorecards\_Test]

--6. Add missing records from Scorecards database

DECLARE @dtNow DATETIME

SET @dtNow = GETDATE()

;With Firstrecord AS

(SELECT

[EMPLOYEE\_ID]

,[START\_DATE]

,[END\_DATE]

,[LAN\_ID]

,ROW\_NUMBER() OVER( PARTITION BY [EMPLOYEE\_ID]

ORDER BY [START\_DATE]) AS Rn

FROM

[BCC\_Scorecards].[bcc\_scrd].[EMPLOYEE\_ID\_TO\_LAN\_ID]

--Where [EMPLOYEE\_ID]='232767'

)

--Select \* FROM Firstrecord

--WHERE Rn = 1

INSERT INTO [EC].[EmployeeID\_To\_LanID]

([EmpID]

,[StartDate]

,[EndDate]

,[LanID]

,[DatetimeInserted]

,[DatetimeLastUpdated])

Select F.[EMPLOYEE\_ID],F.[START\_DATE], F.[END\_DATE], F.[LAN\_ID], @dtNow,@dtNow

FROM Firstrecord AS F LEFT OUTER JOIN [EC].[EmployeeID\_To\_LanID]LAN

ON F.EMPLOYEE\_ID = LAN.[EmpID]

WHERE LAN.[EmpID]IS NULL

AND F.Rn =1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Initial Populate for CSR Hierarchy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Run this in a window pointed to BCC\_Scorecards DB

COACHING DB NAME IN QUERY REFERS TO PRODUCTION . UPDATE TO TO APPROPRIATE ENVIRONMENT IF RUNNING ELSEWHERE.

--<Query Begins Here>

-- Record set with rownum based on start date

; WITH StartD AS

( SELECT

[CsrEmployeeId]

,[SupervisorEmployeeId]

,[ManagerEmployeeId]

,[Start\_Date]

,[End\_Date]

,ROW\_NUMBER() OVER( PARTITION BY [CsrEmployeeId]

ORDER BY [Start\_Date]) AS Rn

FROM

[bcc\_scrd].[HIERARCHY\_V02]

where SupervisorEmployeeId <> 'Unknown' and ManagerEmployeeId <> 'Unknown'

),

-- Record set with (rownum +1) based on end date

EndD AS

( SELECT

[CsrEmployeeId]

,[SupervisorEmployeeId]

,[ManagerEmployeeId]

,[Start\_Date]

, End\_Date

,1+ROW\_NUMBER() OVER( PARTITION BY [CsrEmployeeId]

ORDER BY End\_Date )

AS Rn

FROM

[bcc\_scrd].[HIERARCHY\_V02]

where SupervisorEmployeeId <> 'Unknown' and ManagerEmployeeId <> 'Unknown'

)

-- starting insert into CSR\_Hierarchy table

insert into[eCoaching].[EC].[CSR\_Hierarchy]

([EmpID]

,[SupID]

,[MgrID]

,[StartDate], [Enddate]

)

-- Query first row based on start date

SELECT distinct

s.[CsrEmployeeId]

,s.[SupervisorEmployeeId]

, s.[ManagerEmployeeId]

, [bcc\_scrd].[fn\_dtYYYYMMDD\_to\_Datetime](s.[Start\_Date])as StartDate

,case when s.End\_date <> 99991231 Then [bcc\_scrd].[fn\_dtYYYYMMDD\_to\_Datetime](s.[End\_Date])

Else '9999-12-31 00:00:00.000' END as EndDate

FROM

StartD AS s

where Rn=1

-- Union that with

UNION ALL

-- The rest of the records using the end date of the previous record as start date for subsequent record.

SELECT distinct

s.[CsrEmployeeId]

,s.[SupervisorEmployeeId]

, s.[ManagerEmployeeId]

, DATEADD(day, 1, [bcc\_scrd].[fn\_dtYYYYMMDD\_to\_Datetime](e.[End\_Date]))as StartDate

,case when s.End\_date <> 99991231 Then [bcc\_scrd].[fn\_dtYYYYMMDD\_to\_Datetime](s.[End\_Date])

Else '9999-12-31 00:00:00.000' END as EndDate

FROM

StartD AS s

JOIN

EndD AS e

ON e.[CsrEmployeeId] = s.[CsrEmployeeId]

AND e.Rn = s.Rn;

--</Query Ends Here>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Stage SSIS Packages

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Download the following 7 files from version manager (3 .dtsx files, 3 config files, 1 VBS file)

Location:- \eCoaching\_V2\code\

* + Prod\_Outlier\_Coaching.dtsConfig
  + Prod\_IQS\_Coaching.dtsConfig
  + Prod\_Employee\_Hierarchy.dtsConfig
  + IQS\_Coaching.dtsx
  + Outlier\_Coaching.dtsx
  + Employee\_Hierarchy.dtsx
  + Notifications\_prod.vbs

1. Make a backup copy of the existing Coaching directory on database share at [\\vdenssdbp07\scorecard-ssis\](file:///\\vdenssdbp07\scorecard-ssis\)

and name it Olddb\_Coaching\_mmddyyy or something similar to indicate it is for the old database.

1. The files downloaded from version manager will be used to replace the files in the Coaching directory.
2. Copy the 3 .dtsx and 3 .config files into the Coaching directory overwriting the existing files.
3. Copy the Notifications\_prod.vbs to the \Coaching\Notifications\ directory.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Transfer Historical Data

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Open a new SQL window pointing to eCoaching DB on VDENSSDBP07\SCORP01

Run below queries in order one at a time.

Note the step number at the top of the query to make sure you are running in the right order.

--Step1: Stage the data from old Coaching database to a staging table in the new database

SELECT \*

INTO EC.HT\_Data

FROM Coaching.EC.Coaching\_log

STEPS 2 and 3 will need to be run in same window one after the other. The enable identity insert in step 2 will apply for step 3 only in that window.

Note: Before running step 2 make sure there is no data in the eCoaching.Coaching\_Log table by doing a select.

If there is data reset the identity by running the following command.

--DBCC CheckIdent("[eCoaching].[EC].[Coaching\_Log]", RESEED,0)

-- This is to ensure that we are starting at identity insert at 1.

--Step2: This statement enabled identity insert on the Coaching\_Log table

SET IDENTITY\_INSERT [EC].[Coaching\_Log] ON

--Step3: This statement populates the historical data into Coaching\_Log table

INSERT INTO [EC].[Coaching\_Log]

([CoachingID]

,[FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[CSR]

,[CSRID]

,[SubmitterID]

,[EventDate]

,[CoachingDate]

,[isAvokeID]

,[AvokeID]

,[isNGDActivityID]

,[NGDActivityID]

,[isUCID]

,[UCID]

,[isVerintID]

,[VerintID]

,[VerintEvalID]

,[Description]

,[CoachingNotes]

,[isVerified]

,[SubmittedDate]

,[StartDate]

,[SupReviewedAutoDate]

,[isCSE]

,[MgrReviewManualDate]

,[MgrReviewAutoDate]

,[MgrNotes]

,[isCSRAcknowledged]

,[CSRReviewAutoDate]

,[CSRComments]

,[EmailSent]

,[numReportID]

,[strReportCode]

)

SELECT D.[numID]

, D.[strFormID]

,CASE WHEN D.[SubmittedDate]< '2013-06-01 00:00:00.000'

THEN 'Medicare' Else NULL END -- Program

,[EC].[fn\_intSourceIDFromOldSource](D.[strFormType],D.[strsource])

,[EC].[fn\_strStatusIDFromStatus]( D.[strFormStatus])

,[EC].[fn\_intSiteIDFromSite](D.[strCSRSite])

,D.[strCSR]

,1111--csrid

,[EC].[fn\_nvcGetEmpIdFromLanId](LOWER(D.[strSubmitter]),D.[SubmittedDate])

,D.[EventDate]

,D.[CoachingDate]

,D.[isBehaviorAnalyticsMonitor]

,CASE WHEN LEN(D.[strBehaviorAnalyticsID]) < 40 THEN D.[strBehaviorAnalyticsID]

ELSE NULL END --AvokeID

,D.[isNGDActivityID]

,CASE WHEN LEN(D.[strNGDActivityID]) < 40 THEN D.[strNGDActivityID]

ELSE NULL END --NGDActivityID

,0

,0

,D.[isVerintMonitor]

,CASE WHEN LEN(D.[strVerintID]) < 40 THEN D.[strVerintID]

ELSE NULL END -- VerintID

,D.[strVerint\_EvalID]--

,D.[txtDescription]

,D.[txtCoachingNotes]

,D.[isVerified]

,D.[SubmittedDate]

,D.[StartDate]

,D.[SupReviewedAutoDate]

,D.[isCSE]

,D.[MgrReviewManualDate]

,D.[MgrReviewAutoDate]

,D.[txtMgrNotes]

,D.[isCSRAcknowledged]

,D.[CSRReviewAutoDate]

,D.[txtCSRComments]

,D.[EmailSent]

,D.[numReportID]

,D.[strReportCode]

FROM [EC].[HT\_Data]D LEFT outer join [EC].[Coaching\_Log]C

ON D.[numID]=C.[CoachingID]

AND D.[strFormID]= C.[FormName]

WHERE C.[CoachingID] IS NULL AND C.[FormName]IS NULL

and d.strformid is not null

GO

--Step4: Populate CSRID

UPDATE [EC].[Coaching\_Log]

SET [CSRID]= [EC].[fn\_nvcGetEmpIdFromLanId](LOWER([CSR]),[Submitteddate])

--Step5: Populate Program for Missing records from Employee table

Update [EC].[Coaching\_Log]

SET [ProgramName]= EH.Emp\_program

FROM [EC].[Coaching\_Log]CL JOIN [EC].[Employee\_Hierarchy]EH

ON CL.[CSRID] = [EH].[EMP\_ID]

WHERE CL.[ProgramName] is NULL

--Step6: Populate Program for Missing records from Historical values

Update [EC].[Coaching\_Log]

SET [ProgramName]= HM.Program

FROM [EC].[Coaching\_Log]CL JOIN [EC].[HistoricalMultiCodesForProgram]HM

ON CL.CSRID = HM.EMPLID

WHERE CL.[ProgramName] is NULL

--Step7:. Populate Coaching\_Log\_Reason table

;WITH HCR AS

(SELECT DISTINCT numID,

CASE Coach\_Reason

WHEN N'strCoachReason\_AHT' THEN 1

WHEN N'strCoachReason\_ Attendance' THEN 3

WHEN N'strCoachReason\_BCCSOP' THEN 4

WHEN N'strCoachReason\_ Behavioral\_Analytics' THEN 14

WHEN N'strCoachReason\_ Current\_Coaching\_Initiatives' THEN 5

WHEN N'strCoachReason\_CSE' THEN 6

WHEN N'strCoachReason\_Feedback' THEN 7

WHEN N'strCoachReason\_HR\_Guideline' THEN 8

WHEN N'strCoachReason\_Quality' THEN 10

WHEN N'strCoachReason\_Recognition' THEN 11

WHEN N'strCoachReason\_Schedule\_Adherence' THEN 12

WHEN N'strCoachReason\_SecureFloor' THEN 13

WHEN N'strCoachReason\_ShotCalls' THEN 9

WHEN N'strCoachReason\_TrainingDevelopment' THEN 14

WHEN N'strCoachReason\_Other' THEN 14

END AS Coach\_Reason,

Coach\_Reason\_Value

FROM

(SELECT numID

,[strCoachReason\_AHT]

,[strCoachReason\_ Attendance]

,[strCoachReason\_BCCSOP]

,[strCoachReason\_ Behavioral\_Analytics]

,[strCoachReason\_ Current\_Coaching\_Initiatives]

,[strCoachReason\_CSE]

,[strCoachReason\_Feedback]

,[strCoachReason\_HR\_Guideline]

,[strCoachReason\_Quality]

,[strCoachReason\_Recognition]

,[strCoachReason\_Schedule\_Adherence]

,[strCoachReason\_SecureFloor]

,[strCoachReason\_ShotCalls]

,[strCoachReason\_TrainingDevelopment]

,[strCoachReason\_Other]

FROM [EC].[HT\_Data] where strformid is not null)p

UNPIVOT

(Coach\_Reason\_Value FOR Coach\_Reason IN

([strCoachReason\_AHT]

,[strCoachReason\_ Attendance]

,[strCoachReason\_BCCSOP]

,[strCoachReason\_ Behavioral\_Analytics]

,[strCoachReason\_ Current\_Coaching\_Initiatives]

,[strCoachReason\_CSE]

,[strCoachReason\_Feedback]

,[strCoachReason\_HR\_Guideline]

,[strCoachReason\_Quality]

,[strCoachReason\_Recognition]

,[strCoachReason\_Schedule\_Adherence]

,[strCoachReason\_SecureFloor]

,[strCoachReason\_ShotCalls]

,[strCoachReason\_TrainingDevelopment]

,[strCoachReason\_Other] ))

AS unpvt

where Coach\_Reason\_Value != '')

INSERT INTO [EC].[Coaching\_Log\_Reason]

([CoachingID]

,[CoachingReasonID]

,[SubCoachingReasonID]

,[Value])

(

SELECT distinct numID, Coach\_Reason, 42 as Sub\_Coach\_Reason, MAX(Coach\_Reason\_Value)AS Coach\_Reason\_Value

FROM HCR

where Coach\_Reason is NOT NULL

GROUP BY numID, Coach\_Reason)