

Project 3

By: Nahiyan Ahmed, Joseph Abdel-Monem,
Qamar Din, Arun Kuniel

Create Schemas

```
--SCHEMAS-----
DROP SCHEMA IF EXISTS DbSecurity;
GO
CREATE SCHEMA DbSecurity;
GO
DROP SCHEMA IF EXISTS PkSequence
GO
CREATE SCHEMA PkSequence;
GO
DROP SCHEMA IF EXISTS Process;
GO
CREATE SCHEMA Process;
GO
CREATE SCHEMA [Udt]
GO
CREATE SCHEMA [Location];
GO
CREATE SCHEMA [Education];
GO
CREATE SCHEMA [Faculty];
GO
CREATE SCHEMA [Time];
GO
CREATE SCHEMA [Project3]
GO
```

UDT's

```
--TYPES-----  
[***** Object: UserDefinedDataType [Udt].[SurrogateKeyInt] Script Date: 11/27/2021*****]  
CREATE TYPE [Udt].[SurrogateKeyInt] FROM [int] NULL  
GO  
CREATE TYPE [Udt].[Name] FROM [NVARCHAR](30) NULL  
GO  
CREATE TYPE [Udt].[Numbers] FROM INT NULL  
GO  
  
--Original table cleanup (tba for most nulls)  
--Decide what to do for students Enrolled>Limit  
--uncompleted
```

UPDATE TABLES

```
--Give online classes, blackboard location  
UPDATE [Uploadfile].[CurrentSemesterCourseOfferings] SET [Location]='BlackBoard' WHERE [Mode of Instruction] ='Online'  
GO  
--TBAs for nulls  
UPDATE [Uploadfile].[CurrentSemesterCourseOfferings] SET [Day]='TBA' WHERE [Day] =''  
GO  
UPDATE [Uploadfile].[CurrentSemesterCourseOfferings] SET [Time]='TBA' WHERE [Time] ='-'  
GO  
UPDATE [Uploadfile].[CurrentSemesterCourseOfferings] SET [Instructor]='TBA' WHERE [Instructor] =','  
GO  
UPDATE [Uploadfile].[CurrentSemesterCourseOfferings] SET [Location]='TBA' WHERE [Location] =''  
GO  
UPDATE [Uploadfile].[CurrentSemesterCourseOfferings] SET [Instructor]='TBA' WHERE [Instructor] =','  
GO  
UPDATE [Uploadfile].[CurrentSemesterCourseOfferings] SET [Location]='TBA' WHERE [Location] =''  
GO
```

User Authorization Table

```
-- =====
-- Author: Joseph Abdel-Monem
-- Procedure: create DbSecurity.UserAuthorization table
-- Creation:
-- Description: Table containing users
-- =====
:DROP TABLE IF EXISTS DbSecurity.UserAuthorization

:CREATE SEQUENCE PkSequence.UserAuthorizationSequenceObject AS INT MINVALUE 1;
:CREATE TABLE DbSecurity.UserAuthorization(
    UserAuthorizationKey INT NOT NULL DEFAULT(NEXT VALUE FOR PkSequence.UserAuthorizationSequenceObject) PRIMARY KEY,
    ClassTime nchar(5) NULL DEFAULT (N'10:45'),
    [Individual Project] nvarchar(60) NULL DEFAULT('PROJECT 3'),
    GroupMemberLastName nvarchar(35) NOT NULL,
    GroupMemberFirstName nvarchar(25) NOT NULL,
   GroupName nvarchar(20) DEFAULT (N'G1045_3'),
    DateAdded datetime2 null DEFAULT (SYSDATETIME())
);

:INSERT INTO DbSecurity.UserAuthorization(
    GroupMemberLastName,
    GroupMemberFirstName
)
VALUES
    ('Abdel-Monem', 'Joseph'),
    ('Din', 'Qamar'),
    ('Ahmed', 'Nahiyana'),
    ('Kuniel', 'Arun');
:GO
```

Workflow Steps

```
-- =====
-- Author:      Joseph
-- Procedure:   Process.WorkflowSteps
-- Creation:    11/14/2021
-- Description: Keeps track of each workstep
-- =====

CREATE SEQUENCE PkSequence.WorkflowStepsSequenceObject AS INT MINVALUE 1;

DROP TABLE IF EXISTS Process.WorkflowSteps
CREATE TABLE Process.WorkflowSteps(
    WorkFlowStepKey INT NOT NULL DEFAULT(NEXT VALUE FOR PkSequence.WorkflowStepsSequenceObject),
    WorkFlowStepDescription NVARCHAR(100) NOT NULL,
    WorkFlowStepTableRowCount INT NULL DEFAULT (0),
    StartingDateTime DATETIME2(7) NULL DEFAULT SYSDATETIME(),
    EndingDateTime DATETIME2(7) NULL DEFAULT SYSDATETIME(),
    ClassTime CHAR(5) NULL DEFAULT ('10:45'),
    UserAuthorizationKey INT NOT NULL,
    DateAdded DATETIME2 NULL DEFAULT SYSDATETIME(),
    DateOfLastUpdate DATETIME2 NULL DEFAULT SYSDATETIME(),
    PRIMARY KEY(WorkFlowStepKey)
);
GO
```

TrackWorkFlow

```
-- =====
-- Author:      Joseph
-- Procedure:   Process.usp_TrackWorkflow
-- Creation:    11/13/2021
-- Description: Stored procedure to keep track of workflows
-- =====

CREATE PROCEDURE Process.usp_TrackWorkflow
    @StartTime DATETIME2(7),
    @WorkFlowDescription NVARCHAR(100),
    @WorkFlowStepTableRowCount INT,
    @UserAuthorizationKey INT
AS
BEGIN
    INSERT INTO Process.WorkFlowSteps (StartingDateTime, WorkFlowStepDescription, WorkFlowStepTableRowCount, UserAuthorizationKey)
    VALUES (
        @StartTime,
        @WorkFlowDescription,
        @WorkFlowStepTableRowCount,
        @UserAuthorizationKey
    )
END;
GO
```

Create Department Table

```
-- =====
-- Author: Joseph Abdel-Monem
-- Procedure: CREATE [Faculty].[Department]
-- Creation:
-- Description: Table containing Department info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Faculty].[Department] (
    [DepartmentID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [DepartmentName] [Udt].[Name] NOT NULL

CONSTRAINT [PK_Department] PRIMARY KEY CLUSTERED ([DepartmentID] ASC)
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
    ON [PRIMARY]

GO
```

Create Time Day Table

```
-- =====
-- Author: Joseph Abdel-Monem
-- Procedure: CREATE [Time].[Day]
-- Creation:
-- Description: Table containing Day info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Time].[Day] (
    [DayID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [Day] VARCHAR (10) NOT NULL

CONSTRAINT [PK_Day] PRIMARY KEY CLUSTERED ([DayID] ASC)
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
    ON [PRIMARY]

GO
```

Create Time ClassDay Table

```
-- =====
-- Author: Joseph Abdel-Monem
-- Procedure: CREATE [Time].[ClassDay]
-- Creation:
-- Description: Table containing ClassDay info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Time].[ClassDay] (
    [ClassDayID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [ClassID] [Udt].[SurrogateKeyInt] NOT NULL,
    [DayID] [Udt].[SurrogateKeyInt] NOT NULL

CONSTRAINT [PK_ClassDay] PRIMARY KEY CLUSTERED ([ClassDayID] ASC)
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
CONSTRAINT [FK_Time_ClassID] FOREIGN KEY([ClassID]) REFERENCES [Education].[Class] ([ClassID]),
CONSTRAINT [FK_Time_DayID] FOREIGN KEY([DayID]) REFERENCES [Time].[Day] ([DayID]) )

GO
```

Create Education.ModeOfInstruction

```
-- =====
-- Author: Nahyan Ahmed
-- Procedure: CREATE [Education].[ModeOfInstruction]
-- Creation: 12/13/2021
-- Description: Table containing Mode Of Instruction info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Education].[ModeOfInstruction] (
    [ModeOfInstructionID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [ModeOfInstruction] [Udt].[Name] NOT NULL

CONSTRAINT [PK_ModeOfInstruction] PRIMARY KEY CLUSTERED ([ModeOfInstructionID] ASC)
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
```

Create Location Building Table

```
-- =====
-- Author: Joseph Abdel-Monem
-- Procedure: CREATE [Location].[Building]
-- Creation:
-- Description: Table containing Mode Of Building info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Location].[Building] (
    [BuildingID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [BuildingName] [Udt].[Name] NOT NULL

CONSTRAINT [PK_Building] PRIMARY KEY CLUSTERED ([BuildingID] ASC)
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
ON [PRIMARY]
GO
```

Create Location Room Table

```
-- =====
-- Author: Nahyan Ahmed
-- Procedure: CREATE [Location].[Room]
-- Creation: 12/13/2021
-- Description: Table containing Room info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Location].[Room] (
    [RoomID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [BuildingID] [Udt].[SurrogateKeyInt],
    [RoomNumber] [Udt].[Name] NOT NULL

CONSTRAINT [PK_Room] PRIMARY KEY CLUSTERED ([RoomID] ASC)
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
CONSTRAINT [FK_Location_Room] FOREIGN KEY([BuildingID])
    REFERENCES [Location].[Building] ([BuildingID])
GO
```

Create Faculty Instructor Table

```
-- =====
-- Author: Joseph Abdel-Monem
-- Procedure: CREATE [Faculty].[Instructor]
-- Creation:
-- Description: Table containing Mode Of Instruction info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Faculty].[Instructor] (
    [InstructorID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [Instructor] NVARCHAR(60) NOT NULL,
    [InstructorFirstName] [Udt].[Name] NOT NULL,
    [InstructorLastName] [Udt].[Name] NOT NULL

CONSTRAINT [PK_Instructor] PRIMARY KEY CLUSTERED ([InstructorID] ASC)
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
    ON [PRIMARY]
GO
```

Create Faculty Instructor Department Table

```
-- =====
-- Author: Joseph Abdel-Monem
-- Procedure: CREATE [Faculty].[InstructorDepartment]
-- Creation:
-- Description: Table containing Instructor Department info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Faculty].[InstructorDepartment] (
    [InstructorDepartmentID] [Udt].[SurrogateKeyInt] IDENTITY (1,1) NOT NULL,
    [InstructorID] [Udt].[SurrogateKeyInt] NOT NULL,
    [DepartmentID] [Udt].[SurrogateKeyInt] NOT NULL,
    CONSTRAINT [PK_InstructorDepartment] PRIMARY KEY CLUSTERED ([InstructorDepartmentID] ASC)
        WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
    CONSTRAINT [FK_Faculty_Instructor] FOREIGN KEY([InstructorID])
        REFERENCES [Faculty].[Instructor] ([InstructorID]),
    CONSTRAINT [FK_Faculty_Department] FOREIGN KEY([DepartmentID])
        REFERENCES [Faculty].[Department] ([DepartmentID])
CREATE NONCLUSTERED INDEX [idx_nc_departmentkey] ON [Faculty].[InstructorDepartment]
(
    [DepartmentID] ASC
)
GO

CREATE NONCLUSTERED INDEX [idx_nc_instructorkey] ON [Faculty].[InstructorDepartment]
(
    [InstructorID] ASC
)
GO
```

Create Table Education Course Table

```
-- =====
-- Author: Joseph Abdel-Moneim
-- Procedure: CREATE Table [Education].[Course]
-- Creation:
-- Description: Table containing course info

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Education].[Course](
    [CourseID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [DepartmentID] [Udt].[SurrogateKeyInt] NOT NULL,
    [CourseDescription] [Udt].[Name] NOT NULL,
    [CourseCode] [Udt].[Numbers] NOT NULL,
    [CourseHours] NVARCHAR (2) NOT NULL,
    [CourseCredits] NVARCHAR (2) NOT NULL,
    [DateAdded] [datetime2](7) NOT NULL,
    [DateOfLastUpdated] [datetime2](7) NOT NULL,
    [AuthorizedUserId] [Udt].[Numbers] NOT NULL,
    CONSTRAINT [PK_Course] PRIMARY KEY CLUSTERED ([CourseID] ASC)
        WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
              ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
    CONSTRAINT [FK_Course_Department] FOREIGN KEY([DepartmentID])
        REFERENCES [Faculty].[Department] ([DepartmentID]))

ALTER TABLE [Education].[Class] ADD CONSTRAINT [DF_Course_DateAdded] DEFAULT (sysdatetime()) FOR [DateAdded]
GO

ALTER TABLE [Education].[Class] ADD CONSTRAINT [DF_Course_DateOfLastUpdated] DEFAULT (sysdatetime()) FOR [DateOfLastUpdated]
GO
```

Create Education Class Table

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

:CREATE TABLE [Education].[Class](
    [ClassID] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [InstructorID] [Udt].[SurrogateKeyInt] NOT NULL,
    [RoomID] [Udt].[SurrogateKeyInt] NOT NULL,
    [ModeID] [Udt].[SurrogateKeyInt] NOT NULL,
    [CourseID] [Udt].[SurrogateKeyInt] NOT NULL,

    [ClassSection] [Udt].[Numbers] NOT NULL ,
    [ClassCode] [Udt].[Numbers] NOT NULL,
    [StartTime] VARCHAR(10) NOT NULL,
    [EndTime] VARCHAR(10) NOT NULL,
    [Enrolled] [Udt].[Numbers] NOT NULL,
    [Limit] [Udt].[Numbers] NOT NULL,
    [Semester] VARCHAR(25) NULL,
    [DateAdded] [datetime2](7) NOT NULL,
    [DateOfLastUpdated] [datetime2](7) NOT NULL,
    [AuthorizedUserId] [int] NOT NULL,
    CONSTRAINT [PK_Class] PRIMARY KEY CLUSTERED ([ClassID] ASC)
        WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
    ON [PRIMARY]

ALTER TABLE [Education].[Class] ADD CONSTRAINT [DF_Class_DateAdded] DEFAULT (sysdatetime()) FOR [DateAdded]
GO

ALTER TABLE [Education].[Class] ADD CONSTRAINT [DF_Class_DateOfLastUpdated] DEFAULT (sysdatetime()) FOR [DateOfLastUpdated]
GO

ALTER TABLE [Education].[Class] ADD CONSTRAINT [FK_Class_Instructor] FOREIGN KEY (InstructorID) REFERENCES [Faculty].[Instructor] ([InstructorID]);
GO

ALTER TABLE [Education].[Class] ADD CONSTRAINT [FK_Class_Room] FOREIGN KEY (RoomID) REFERENCES [Location].[Room] (RoomID);
GO

ALTER TABLE [Education].[Class] ADD CONSTRAINT [FK_Class_ModeOfInstruction] FOREIGN KEY ([ModeID]) REFERENCES [Education].[ModeOfInstruction] ([ModeOfInstructionID]);
GO

ALTER TABLE [Education].[Class] ADD CONSTRAINT [FK_Class_Course] FOREIGN KEY ([CourseID]) REFERENCES [Education].[Course] ([CourseID]);
GO
```

Load Faculty Department

```
-- =====
-- Author:      Joseph Abdel-Monem
-- Procedure:   LoadFaculty.Department
-- Creation:    11/14/2021
-- Description: Loads data into Fact Department table

CREATE PROCEDURE Project3.LoadFacultyDepartment
    @UserKey INT
AS
BEGIN
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @RowCount as int;
    INSERT INTO [Faculty].[Department] (DepartmentName)
    SELECT DISTINCT SUBSTRING ([Course (hr, crd)], 1,CHARINDEX(' ',[Course (hr, crd)])-1)
    FROM [Uploadfile].[CurrentSemesterCourseOfferings]

    SET @RowCount = (SELECT COUNT(*)FROM [Faculty].[Department] );

    EXEC Process.usp_TrackWorkflow
        @StartTime = @CURRENTTIME,
        @WorkflowDescription = 'Loads data into Department table',
        @WorkflowStepTableRowCount = @RowCount,
        @UserAuthorizationKey = @UserKey;
END
GO
```

Load Location Building

```
-- =====
-- Author:      Joseph Abdel-Monem
-- Procedure:   LoadLocation.Building
-- Creation:    11/14/2021
-- Description: Loads data into Building table

CREATE PROCEDURE Project3.LoadLocationBuilding
    @UserKey INT
AS
BEGIN
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @RowCount as int;
    INSERT INTO [Location].[Building] (BuildingName)
    SELECT DISTINCT
        CASE [Location]
            WHEN N'BlackBoard' THEN N'BLACKBOARD'
            WHEN N'TBA' THEN N'TBA'
            ELSE SUBSTRING ([Location], 1,CHARINDEX(' ',[Location]))
        END
    FROM [Uploadfile].[CurrentSemesterCourseOfferings]
    SET @RowCount = (SELECT COUNT(*)FROM [Location].[Building] );

    EXEC Process.usp_TrackWorkFlow
        @StartTime = @CURRENTTIME,
        @WorkflowDescription = 'Loads data into Building table',
        @WorkflowStepTableRowCount = @RowCount,
        @UserAuthorizationKey = @UserKey;
END
GO
```

Load Faculty Instructor

```
-- =====
-- Author:      Joseph Abdel-Monem
-- Procedure:   LoadFaculty.Instructor
-- Creation:    11/14/2021
-- Description: Loads data into Instructor table

CREATE PROCEDURE Project3.LoadFacultyInstructor
    @UserKey INT
AS
BEGIN
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @RowCount as int;
    INSERT INTO [Faculty].[Instructor] ([Instructor],[InstructorFirstName],[InstructorLastName])
    SELECT DISTINCT [Instructor],
    CASE [Instructor]
        WHEN N'TBA' THEN N'TBA'
        ELSE SUBSTRING ([Instructor], 1,CHARINDEX(',',[Instructor])-1)
    END,
    CASE [Instructor]
        WHEN N'TBA' THEN N'TBA'
        ELSE RIGHT(Instructor, len(Instructor)-charindex(',', Instructor))
    END
    FROM [Uploadfile].[CurrentSemesterCourseOfferings]
    SET @RowCount = (SELECT COUNT(*)FROM [Faculty].[Instructor]);

    EXEC Process.usp_TrackWorkflow
    @StartTime = @CURRENTTIME,
    @WorkflowDescription = 'Loads data into Instructor table',
    @WorkflowStepTableRowCount = @RowCount,
    @UserAuthorizationKey = @UserKey;
END
GO
```

Load Time Day

```
-- =====
-- Author:      Joseph Abdel-Monem
-- Procedure:   LoadFaculty.Instructor
-- Creation:    11/14/2021
-- Description: Loads data into Instructor table

CREATE PROCEDURE Project3.LoadTimeDay
    @UserKey INT
AS
BEGIN
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @RowCount as int;
    INSERT INTO [Time].[Day] ([Day])
    SELECT DISTINCT
        CASE
            WHEN [Day] LIKE N'SU' THEN N'Sunday'
            WHEN [Day] LIKE N'S' THEN N'Saturday'
            WHEN [Day] LIKE N'M' THEN N'Munday'
            WHEN [Day] LIKE N'T' THEN N'Tuesday'
            WHEN [Day] LIKE N'W' THEN N'Wednesday'
            WHEN [Day] LIKE N'TH' THEN N'Thursday'
            WHEN [Day] LIKE N'F' THEN N'Friday'
            ELSE N'TBA'
        END
    FROM [Uploadfile].[CurrentSemesterCourseOfferings]

    SET @RowCount = (SELECT COUNT(*)FROM [Time].[Day]);

    EXEC Process.usp_TrackWorkflow
        @StartTime = @CURRENTTIME,
        @WorkflowDescription = 'Loads data into Instructor table',
        @WorkflowStepTableRowCount = @RowCount,
        @UserAuthorizationKey = @UserKey;
END
GO
/*EXEC sys.sp_columns
    @table_name = N'Instructor',
    @table_owner = N'Faculty'
*/ --testing
```

Load Class Data

```
-- =====
-- Author:      Joseph Abdel-Monem
-- Procedure:   LoadEducation.Class
-- Creation:    11/14/2021
-- Description: Loads data into Fact Class table

CREATE PROCEDURE Project3.LoadClass_Data
    @UserKey INT
AS
BEGIN
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @RowCount as int;

    -- NEED TO FINISH OTHER TABLES FIRST
    INSERT INTO [Education].[Class](ClassSection,ClassCode,StartTime,EndTime,Enrolled, Limit,Semester,[AuthorizedUserId] )
    SELECT Sec,Code,
        CASE Time
            WHEN N'T%' THEN N'TBA'
            ELSE SUBSTRING (Time, 1,CHARINDEX('M',Time))
        END AS StartTime,
        CASE Time
            WHEN N'T%TBA' THEN N'TBA'
            ELSE SUBSTRING (Time,CHARINDEX('-',Time)+2,LEN(Time))
        END AS EndTime,
        Enrolled, Limit, Semester, @Userkey AS [AuthorizedUserId]
    FROM [Uploadfile].[CurrentSemesterCourseOfferings]

END
GO
```

Load Room

```
-- =====
-- Author:      Nahyan Ahmed
-- Procedure:   LoadRoom.Class
-- Creation:    12/13/2021
-- Description: Loads data into Room table
-- =====
CREATE PROCEDURE Project3.LoadRoom
    @UserKey int
AS
BEGIN
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @RowCount as int;
    INSERT INTO [Location].[Room] (RoomNumber)

        SELECT DISTINCT
        CASE [Location]
            WHEN N'BlackBoard' THEN N'BLACKBOARD'
            WHEN N'TBA' THEN N'TBA'
            ELSE RIGHT ([Location], 4)
        END
        FROM Uploadfile.CurrentSemesterCourseOfferings

        SET @RowCount = (SELECT COUNT(*)FROM [Location].[Room] );

        EXEC Process.usp_TrackWorkFlow
        @StartTime = @CURRENTTIME,
        @WorkFlowDescription = 'Loads data into Room table',
        @WorkFlowStepTableRowCount = @RowCount,
        @UserAuthorizationKey = @UserKey;
END
GO
```

Load Mode Of Instruction

```
-- =====
-- Author:      Nahiyan Ahmed
-- Procedure:   LoadModeOfInstruction
-- Creation:    12/13/2021
-- Description: Loads data into Mode Of Instruction table
-- =====
CREATE PROCEDURE Project3.LoadModeOfInstruction
    @UserKey int
AS
BEGIN
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @RowCount as int;
    INSERT INTO [Education].[ModeOfInstruction] (ModeOfInstruction)

        SELECT DISTINCT [Mode of Instruction]
        FROM Uploadfile.CurrentSemesterCourseOfferings

        SET @RowCount = (SELECT COUNT(*)FROM [Education].[ModeOfInstruction] );

    EXEC Process.usp_TrackWorkFlow
        @StartTime = @CURRENTTIME,
        @WorkFlowDescription = 'Loads data into Mode Of Instruction table',
        @WorkFlowStepTableRowCount = @RowCount,
        @UserAuthorizationKey = @UserKey;
END
GO
```

Load Education Course

```
-- =====
-- Author:      Qamar Din
-- Procedure:   LoadEducation.Course
-- Creation:    12/14/2021
-- Description: Loads data into Course table
-- =====
CREATE PROCEDURE Project3.LoadEducationCourse
    @UserKey INT
AS
BEGIN
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @RowCount int;

    INSERT INTO [Education].[Course] (CourseCode, CourseDescription, CourseHours, CourseCredits)
    SELECT
        Code,
        Description,
        CONVERT(INT, SUBSTRING([Course (hr, crd)], CHARINDEX(',', [Course (hr, crd)]) - 1, 1)) AS Hours,
        CONVERT(INT, SUBSTRING([Course (hr, crd)], CHARINDEX(',', [Course (hr, crd)]) + 2, 1)) AS Credits

    FROM [Uploadfile].[CurrentSemesterCourseOfferings]

    SET @RowCount = (SELECT COUNT(*)FROM [Education].[Course] );
    EXEC Process.usp_TrackWorkFlow
        @StartTime = @CURRENTTIME,
        @WorkFlowDescription = 'Loads data into Course table',
        @WorkFlowStepTableRowCount = @RowCount,
        @UserAuthorizationKey = @UserKey;
END
GO
```

Load ERD Data

```
ALTER PROCEDURE [Project3].[LoadERDData]
    @UserKey INT
AS
BEGIN

    SET NOCOUNT ON;
    DECLARE @CURRENTTIME DATETIME2(7) = SYSDATETIME();
    DECLARE @return_value INT;

    -- Drop foreign keys prior to truncating Star Schema tables
    EXEC [Project3].[DropForeignKeysFromSchemaData];

    -- Check row count before truncation
    EXEC[Project3].[ShowTableStatusRowCount]
        @TableStatus = N'Before truncating tables';

    -- Truncate Star Schema Data
    EXEC [Project3].[TruncateStarData];

    -- Load the Whole Schema
    EXEC [Project3].[LoadFacultyDepartment] @UserKey=1;
    EXEC [Project3].[LoadFacultyInstructor] @UserKey = 1;
    EXEC [Project3].[LoadTimeDay] @UserKey = 1;
    EXEC [Project3].[LoadModeOfInstruction] @UserKey = 1;

    EXEC [Project3].[LoadLocationBuilding] @UserKey = 1;
    EXEC [Project3].[LoadLocationRoom] @UserKey = 1;
    EXEC [Project3].[LoadTimeClassDay] @UserKey = 1;
    EXEC [Project3].[LoadInstructorDepartment] @UserKey = 1;
    EXEC [Project3].[LoadEducationCourse] @UserKey = 1;

    EXEC [Project3].[LoadEducationClass] @UserKey = 1;

    -- Check row after before truncation
    EXEC[Project3].[ShowTableStatusRowCount]
        @TableStatus = N'After truncating tables';

    EXEC Process.usp_TrackWorkFlow
        @StartTime = @CURRENTTIME,
        @WorkFlowDescription = 'Loads data into individual table',
        @WorkFlowStepTableRowCount = @RowCount,
        @UserAuthorizationKey = @UserKey;
END ;
```

Truncate the tables

```
-- =====
-- Author: Joseph Abdel-Monem
-- Create date: 11/14/21
-- Description: Truncates the tables
-- =====
***** Object: StoredProcedure [Project3].[TruncateSchemaData] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [Project3].[TruncateSchemaData]
AS

BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    truncate table [Education].[Class];
    truncate table [Education].[Course];
    truncate table [Education].[ModeOfInstruction];
    truncate table [Faculty].[Department];
    truncate table [Faculty].[Instructor];
    truncate table [Faculty].[InstructorDepartment];
    truncate table [Location].[Building];
    truncate table [Location].[Room];
    truncate table [Time].[ClassDay];
    truncate table [Time].[Day];

end
GO
```

Table Row Status

```
-- =====
-- Author:      Nahiyah Ahmed
-- Procedure:   table row status
-- Creation:    12/13/2021
-- Description: Show table row status count
-- =====
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [Project3].[TableRowStatus]
    @GroupMemberUserAuthorizationKey int,
    @TableStatus NVARCHAR(30)

AS
BEGIN
    SET NOCOUNT ON;
    select TableStatus = @TableStatus, TableName =[Location].Building, COUNT(*) as numRows
    FROM [Location].Building
    select TableStatus = @TableStatus, TableName ='Education.Class', COUNT(*) as numRows
    FROM Education.Class
    select TableStatus = @TableStatus, TableName ='Education.Course', COUNT(*) as numRows
    FROM Education.Course
    select TableStatus = @TableStatus, TableName ='Faculty.Department', COUNT(*) as numRows
    FROM Faculty.Department
    select TableStatus = @TableStatus, TableName ='Faculty.Instructor', COUNT(*) as numRows
    FROM Faculty.Instructor
    select TableStatus = @TableStatus, TableName ='Faculty.InstructorDepartment', COUNT(*) as numRows
    FROM Faculty.InstructorDepartment
    select TableStatus = @TableStatus, TableName ='Education.ModeOfInstruction', COUNT(*) as numRows
    FROM Education.ModeOfInstruction
    select TableStatus = @TableStatus, TableName =[Location].Room, COUNT(*) as numRows
    FROM [Location].Room
    select TableStatus = @TableStatus, TableName ='Uploadfile.CurrentSemesterCourseOfferings', COUNT(*) as numRows
    FROM Uploadfile.CurrentSemesterCourseOfferings

END;
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Add Foreign Keys To Schema Data

```
-- =====
-- Author:      Joseph Abdel-Monem
-- Procedure:   Process.AddForeignKeysToSchemaData
-- Create date: 11/14/21
-- Description: Add foreign keys to the Tables
-- =====
ALTER PROCEDURE [Project3].[AddForeignKeysToSchemaData]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    -- Insert statements for procedure here
    ALTER TABLE [CH01-01-Fact].[Data] WITH CHECK ADD CONSTRAINT [FK_Data_DimCustomer] FOREIGN KEY([CustomerKey])
    REFERENCES [CH01-01-Dimension].[DimCustomer] ([CustomerKey])
    --
    ALTER TABLE [CH01-01-Fact].[Data] CHECK CONSTRAINT [FK_Data_DimCustomer]
    --
    ALTER TABLE [CH01-01-Fact].[Data] WITH CHECK ADD CONSTRAINT [FK_Data_DimGender] FOREIGN KEY([Gender])
    REFERENCES [CH01-01-Dimension].[DimGender] ([Gender])
    --
END;
```

Drop Foreign Keys

```
-- =====
-- Author: Joseph abdel-Moneim
-- Procedure: [Project3].[DropForeignKeysFromStarSchemaData]
-- Create date: 12/16/21
-- Description: Drop Foreign Keys
-- =====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [Project3].[DropForeignKeysFromSchemaData]
    @GroupMemberUserAuthorizationKey INT
AS
BEGIN
    -- Dropping All foreign keys as part of the load process to for referential integrity purposes.

    --EXEC sys.sp_helpconstraint @objname = N'', -- nvarchar(776)

    ALTER TABLE [Education].[Class]
    DROP CONSTRAINT IF EXISTS [FK_Class_Instructor]
    ALTER TABLE [Education].[Class]
    DROP CONSTRAINT IF EXISTS [FK_Class_Room]
    ALTER TABLE [Education].[Class]
    DROP CONSTRAINT IF EXISTS [FK_Class_ModeOfInstruction]
    ALTER TABLE [Education].[Class]
    DROP CONSTRAINT IF EXISTS [FK_Class_Course]

    ALTER TABLE [Education].[Course]
    DROP CONSTRAINT IF EXISTS [FK_Course_Department]

    ALTER TABLE [Faculty].[InstructorDepartment]
    DROP CONSTRAINT IF EXISTS [FK_Faculty_Instructor]
    ALTER TABLE [Faculty].[InstructorDepartment]
    DROP CONSTRAINT IF EXISTS [FK_Faculty_Department]

    ALTER TABLE [Time].[ClassDay]
    DROP CONSTRAINT IF EXISTS [FK_Time_ClassID]
    ALTER TABLE [Time].[ClassDay]
    DROP CONSTRAINT IF EXISTS [FK_Time_DayID]

    ALTER TABLE [Location].[Room]
    DROP CONSTRAINT IF EXISTS [FK_Location_Room]

END;
```

Add foreign keys

```
-- Author:      Joseph Abdel-Moneim
-- Procedure:   Process.AddForeignKeysToSchemaData
-- Create date: 11/14/21
-- Description: Add foreign keys to the Tables
-- =====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [Project3].[AddForeignKeysToSchemaData]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    -- Insert statements for procedure here

    ALTER TABLE [Education].[Class]
    ADD CONSTRAINT [FK_Class_Instructor] FOREIGN KEY ([InstructorID]) REFERENCES Faculty.Instructor ([InstructorID])
    ALTER TABLE [Education].[Class]
    ADD CONSTRAINT [FK_Class_Room] FOREIGN KEY ([RoomID]) REFERENCES [Location].[Room] ([RoomID])
    ALTER TABLE [Education].[Class]
    ADD CONSTRAINT [FK_Class_ModeOfInstruction] FOREIGN KEY ([ModeID]) REFERENCES [Education].[ModeOfInstruction] ([ModeOfInstructionID])
    ALTER TABLE [Education].[Class]
    ADD CONSTRAINT [FK_Class_Course] FOREIGN KEY ([CourseID]) REFERENCES [Education].[Course] ([CourseID])

    ALTER TABLE [Education].[Course]
    ADD CONSTRAINT [FK_Course_Department] FOREIGN KEY ([DepartmentID]) REFERENCES [Faculty].[Department] ([DepartmentID])

    ALTER TABLE [Faculty].[InstructorDepartment]
    ADD CONSTRAINT [FK_Faculty_Instructor] FOREIGN KEY ([InstructorID]) REFERENCES [Faculty].[Instructor] ([Instructor])
    ALTER TABLE [Faculty].[InstructorDepartment]
    ADD CONSTRAINT [FK_Faculty_Department] FOREIGN KEY ([DepartmentID]) REFERENCES [Faculty].[Department] ([DepartmentID])

    ALTER TABLE [Time].[ClassDay]
    ADD CONSTRAINT [FK_Time_ClassID] FOREIGN KEY ([ClassID]) REFERENCES [Education].[Class] ([ClassID])
    ALTER TABLE [Time].[ClassDay]
    ADD CONSTRAINT [FK_Time_DayID] FOREIGN KEY ([DayID]) REFERENCES [Time].[Day] ([DayID])

    ALTER TABLE [Location].[Room]
    ADD CONSTRAINT [FK_Location_Room] FOREIGN KEY ([BuildingID]) REFERENCES [Location].[Building] ([BuildingID])

END;
```