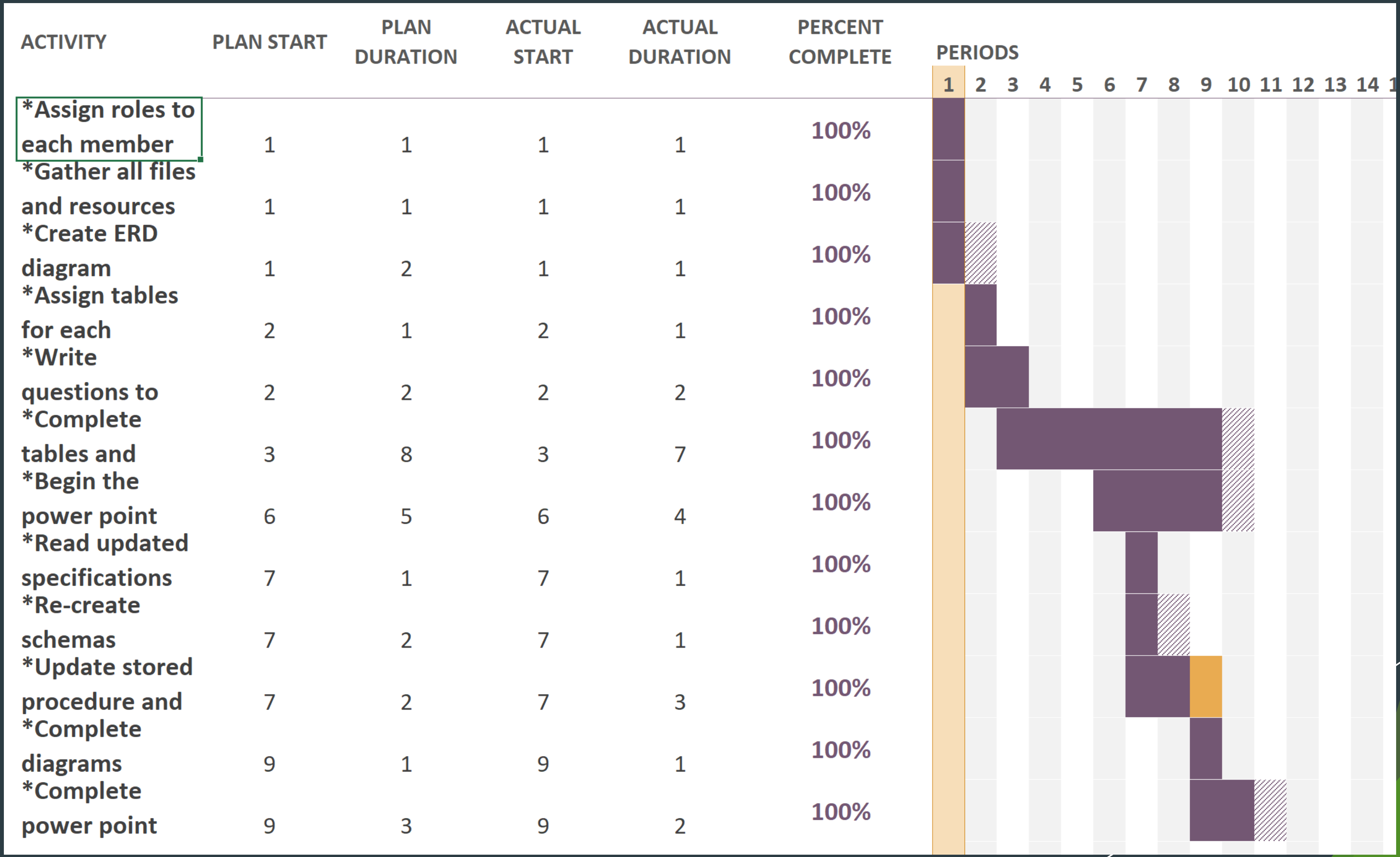
CS331 Final Project

Members: Erik Kim, Jonathan Eng, Harjit Liyal, Haibo Liu, Danny Kong, Jamil Kocacal, Marlon Louis

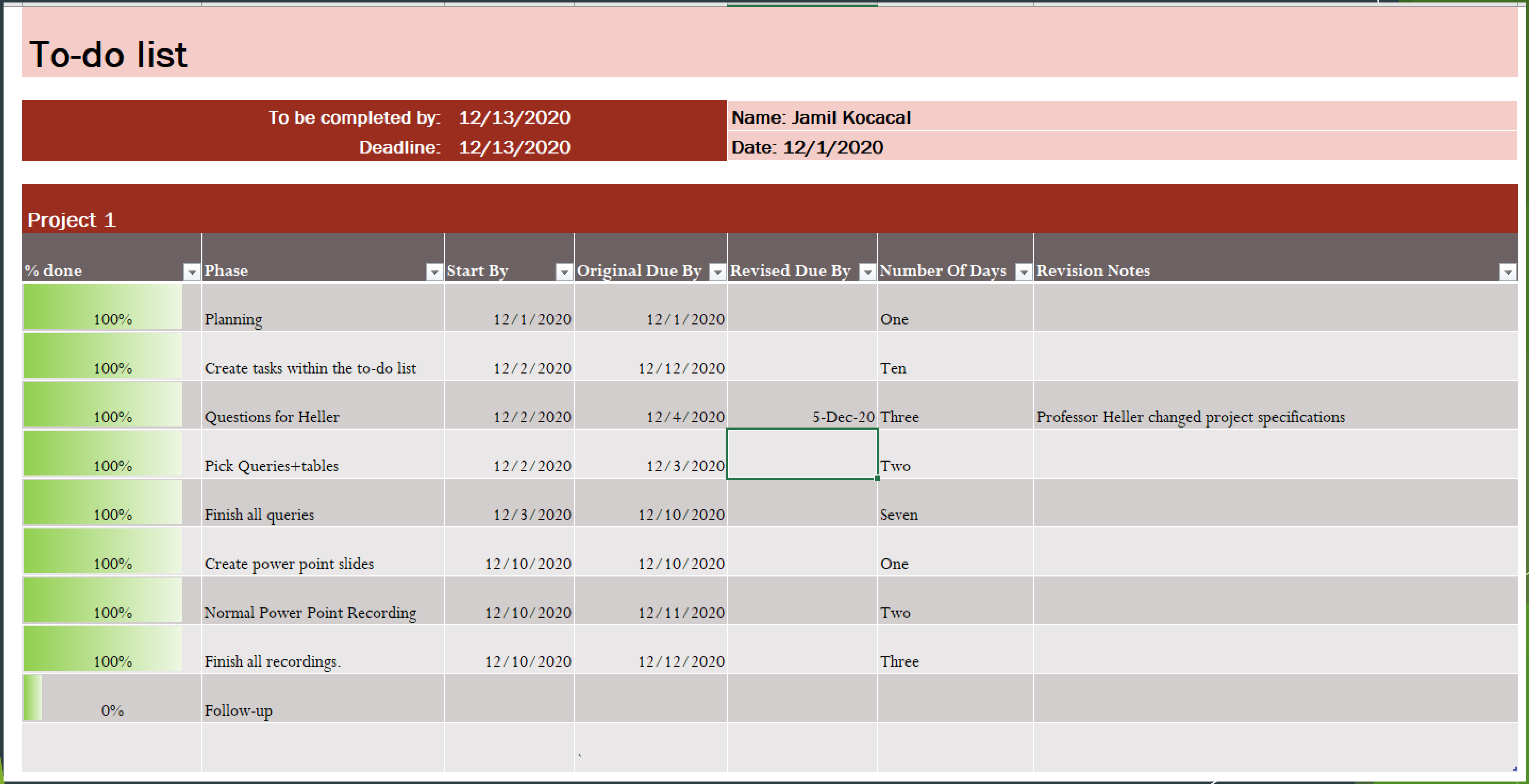
Fall 2020

Project Planner



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
*Assign roles to each member	1	1	1	1	100%
*Gather all files and resources	1	1	1	1	100%
*Create ERD diagram	1	2	1	1	100%
*Assign tables for each	2	1	2	1	100%
*Write questions to	2	2	2	2	100%
*Complete tables and	3	8	3	7	100%
*Begin the power point	6	5	6	4	100%
*Read updated specifications	7	1	7	1	100%
*Re-create schemas	7	2	7	1	100%
*Update stored procedure and	7	2	7	3	100%
*Complete diagrams	9	1	9	1	100%
*Complete power point	9	3	9	2	100%

To-Do List



To-do list

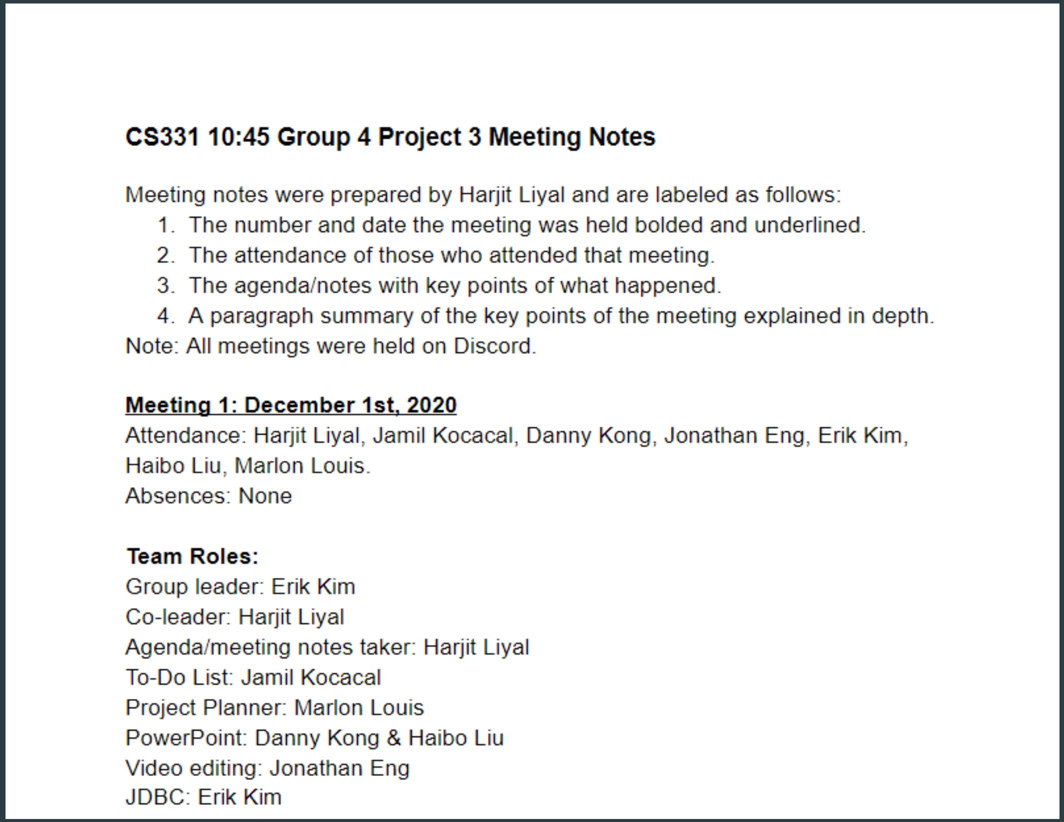
To be completed by: 12/13/2020
Deadline: 12/13/2020

Name: Jamil Kocacal
Date: 12/1/2020

Project 1

% done	Phase	Start By	Original Due By	Revised Due By	Number Of Days	Revision Notes
100%	Planning	12/1/2020	12/1/2020		One	
100%	Create tasks within the to-do list	12/2/2020	12/12/2020		Ten	
100%	Questions for Heller	12/2/2020	12/4/2020	5-Dec-20	Three	Professor Heller changed project specifications
100%	Pick Queries+tables	12/2/2020	12/3/2020		Two	
100%	Finish all queries	12/3/2020	12/10/2020		Seven	
100%	Create power point slides	12/10/2020	12/10/2020		One	
100%	Normal Power Point Recording	12/10/2020	12/11/2020		Two	
100%	Finish all recordings.	12/10/2020	12/12/2020		Three	
0%	Follow-up					

Meeting Notes



CS331 10:45 Group 4 Project 3 Meeting Notes

Meeting notes were prepared by Harjit Liyal and are labeled as follows:

1. The number and date the meeting was held bolded and underlined.
2. The attendance of those who attended that meeting.
3. The agenda/notes with key points of what happened.
4. A paragraph summary of the key points of the meeting explained in depth.

Note: All meetings were held on Discord.

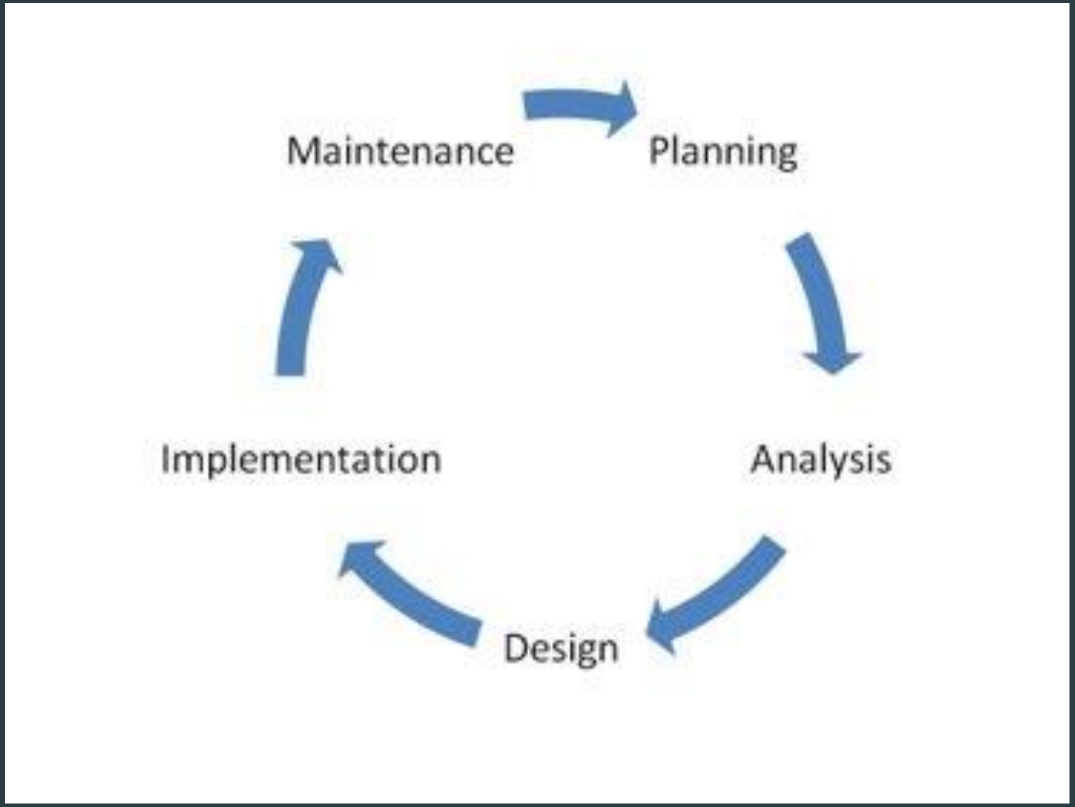
Meeting 1: December 1st, 2020
Attendance: Harjit Liyal, Jamil Kocacal, Danny Kong, Jonathan Eng, Erik Kim, Haibo Liu, Marlon Louis.
Absences: None

Team Roles:
Group leader: Erik Kim
Co-leader: Harjit Liyal
Agenda/meeting notes taker: Harjit Liyal
To-Do List: Jamil Kocacal
Project Planner: Marlon Louis
PowerPoint: Danny Kong & Haibo Liu
Video editing: Jonathan Eng
JDBC: Erik Kim

Link: https://docs.google.com/document/d/1mhdwZ6LVNgfP_QdhOUQDr_1c3-Q7fyE2li-nNdEt948/edit?usp=sharing

Note: Meeting notes are included with the project

SSMS Lifecycle



Planning

- Assigned roles to every group member.
- Tracked workflow, meetings, and deadlines using meeting notes, planner, and to-do list.
- Divided work amongst all members.
- Created ERD Diagram to show relationships among all assigned tables.
- Created a template for all tables and stored procedures.

Analysis

- We looked over the project specifications when they were updated by Professor Heller.
- As a group, we created questions about what was expected and asked Professor Heller for clarification.
- Reviewed Professor Heller's feedback as a group and implemented it into our project.
- Created & Designed database and JDBC to meet guidelines.

Design Decision

- Identified a conflict where columns that would be referenced as foreign keys would not always be unique or primary keys (Instructors with the same name, rooms with the same number)
- Created a solution where we created columns with the primary key and identity constraints to guarantee uniqueness so that they could be referenced as foreign keys
- This method would mean that we use primary keys to guarantee constraints would work correctly
- Our database parses information from the data and stores it into separate schemas for organization

Implementation

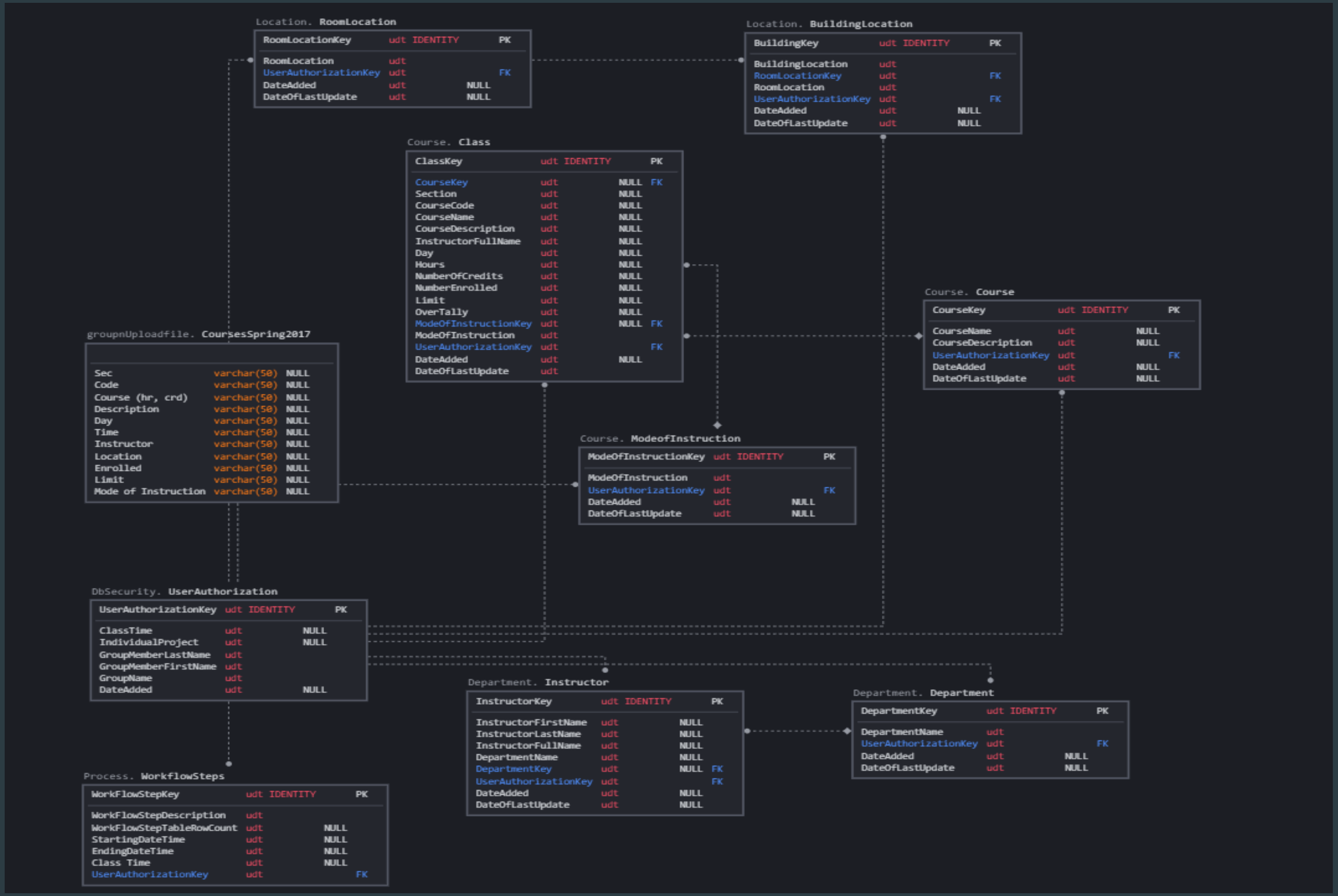
- Tested the created Procedures in SSMS by executing each one individually
- Used Truncate to restart entries for re-testing if required
- Observed the contents of the newly created View, taking note of if it contained the information desired
- Observed the contents of the newly created Table, taking note of if it contained the contents of the view, in addition to other columns we want to add
- Observed the WorkFlowSteps table to ensure the tested outputs are documented, allowing for easily obtainable and informative reference points if needed

Maintenance

- Distribute the tables to everyone, and all columns data types fixed to be user defined
- Fixed the store procedure and views that can not get data from the table, and tested it in own db
- Fixed some errors for create free queries.

Database Walkthrough

ERD Diagram



```
Location. RoomLocation
RoomLocationKey         udt IDENTITY        PK
RoomLocation            udt
UserAuthorizationKey    udt                 FK
DateAdded               udt           NULL
DateOfLastUpdate        udt           NULL
```

```
Location. BuildingLocation
BuildingKey             udt IDENTITY        PK
BuildingLocation        udt
RoomLocationKey         udt                 FK
RoomLocation            udt
UserAuthorizationKey    udt                 FK
DateAdded               udt           NULL
DateOfLastUpdate        udt           NULL
```

```
Course. Class
ClassKey                udt IDENTITY        PK
CourseKey               udt           NULL  FK
Section                 udt           NULL
CourseCode              udt           NULL
CourseName              udt           NULL
CourseDescription       udt           NULL
InstructorFullName      udt           NULL
Day                     udt           NULL
Hours                   udt           NULL
NumberOfCredits         udt           NULL
NumberEnrolled          udt           NULL
Limit                   udt           NULL
OverTally               udt           NULL
ModeOfInstructionKey    udt           NULL  FK
ModeOfInstruction       udt
UserAuthorizationKey    udt                 FK
DateAdded               udt           NULL
DateOfLastUpdate        udt
```

```
Course. Course
CourseKey               udt IDENTITY        PK
CourseName              udt           NULL
CourseDescription       udt           NULL
UserAuthorizationKey    udt                 FK
DateAdded               udt           NULL
DateOfLastUpdate        udt           NULL
```

```
groupnUploadfile. CoursesSpring2017
Sec                  varchar(50) NULL
Code                 varchar(50) NULL
Course (hr, crd)     varchar(50) NULL
Description          varchar(50) NULL
Day                  varchar(50) NULL
Time                 varchar(50) NULL
Instructor           varchar(50) NULL
Location             varchar(50) NULL
Enrolled             varchar(50) NULL
Limit                varchar(50) NULL
Mode of Instruction  varchar(50) NULL
```

```
Course. ModeofInstruction
ModeOfInstructionKey udt IDENTITY        PK
ModeOfInstruction    udt
UserAuthorizationKey udt                 FK
DateAdded            udt           NULL
DateOfLastUpdate     udt           NULL
```

```
DbSecurity. UserAuthorization
UserAuthorizationKey   udt IDENTITY      PK
ClassTime              udt         NULL
IndividualProject      udt         NULL
GroupMemberLastName    udt
GroupMemberFirstName   udt
GroupName              udt
DateAdded              udt         NULL
```

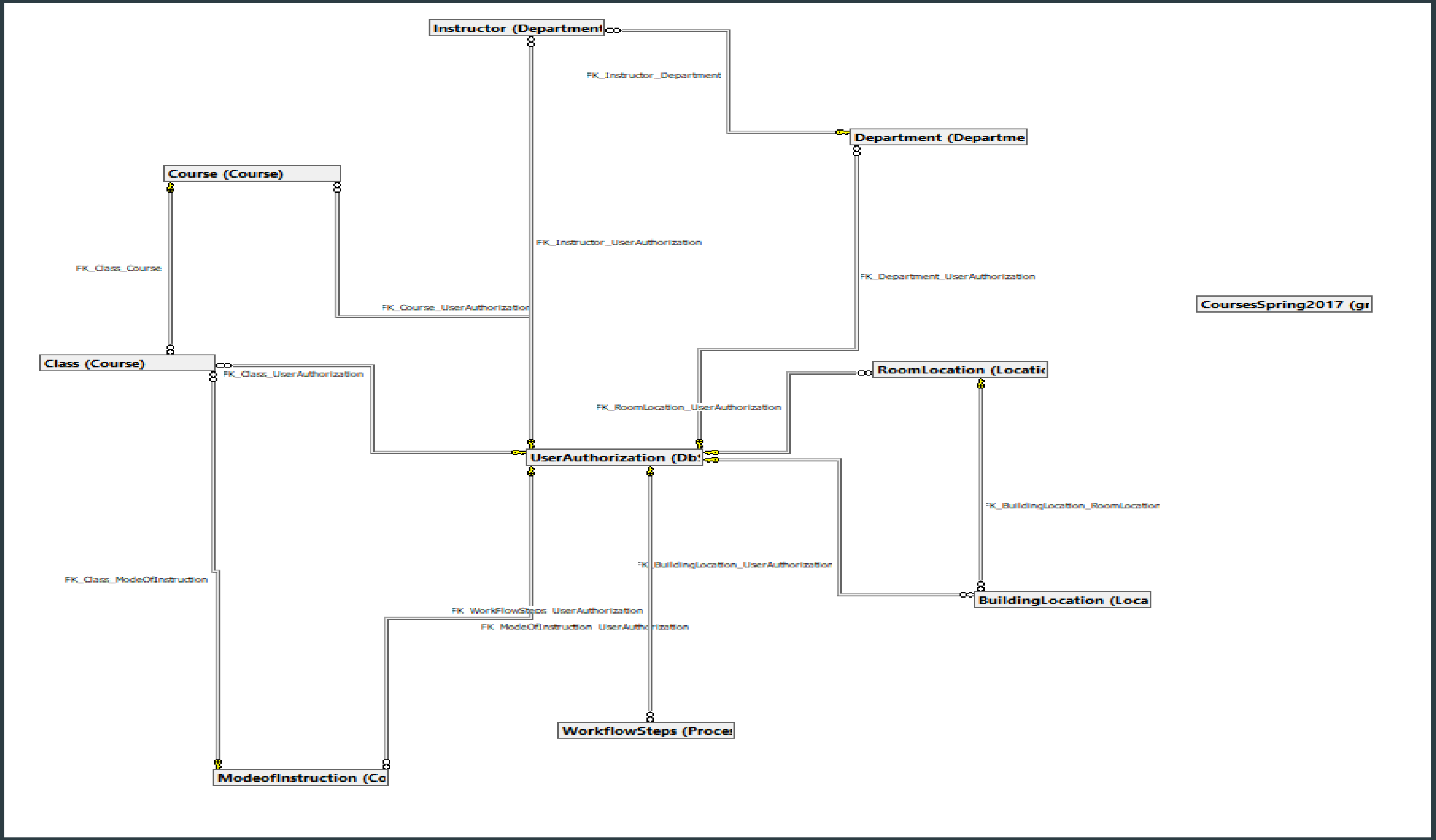
```
Department. Instructor
InstructorKey          udt IDENTITY        PK
InstructorFirstName    udt           NULL
InstructorLastName     udt           NULL
InstructorFullName     udt           NULL
DepartmentName         udt           NULL
DepartmentKey          udt           NULL  FK
UserAuthorizationKey   udt                 FK
DateAdded              udt           NULL
DateOfLastUpdate       udt           NULL
```

```
Department. Department
DepartmentKey          udt IDENTITY        PK
DepartmentName         udt
UserAuthorizationKey   udt                 FK
DateAdded              udt           NULL
DateOfLastUpdate       udt           NULL
```

```
Process. WorkflowSteps
WorkFlowStepKey            udt IDENTITY        PK
WorkFlowStepDescription    udt
WorkFlowStepTableRowCount  udt           NULL
StartingDateTime           udt           NULL
EndingDateTime             udt           NULL
Class Time                 udt           NULL
UserAuthorizationKey       udt                 FK
```

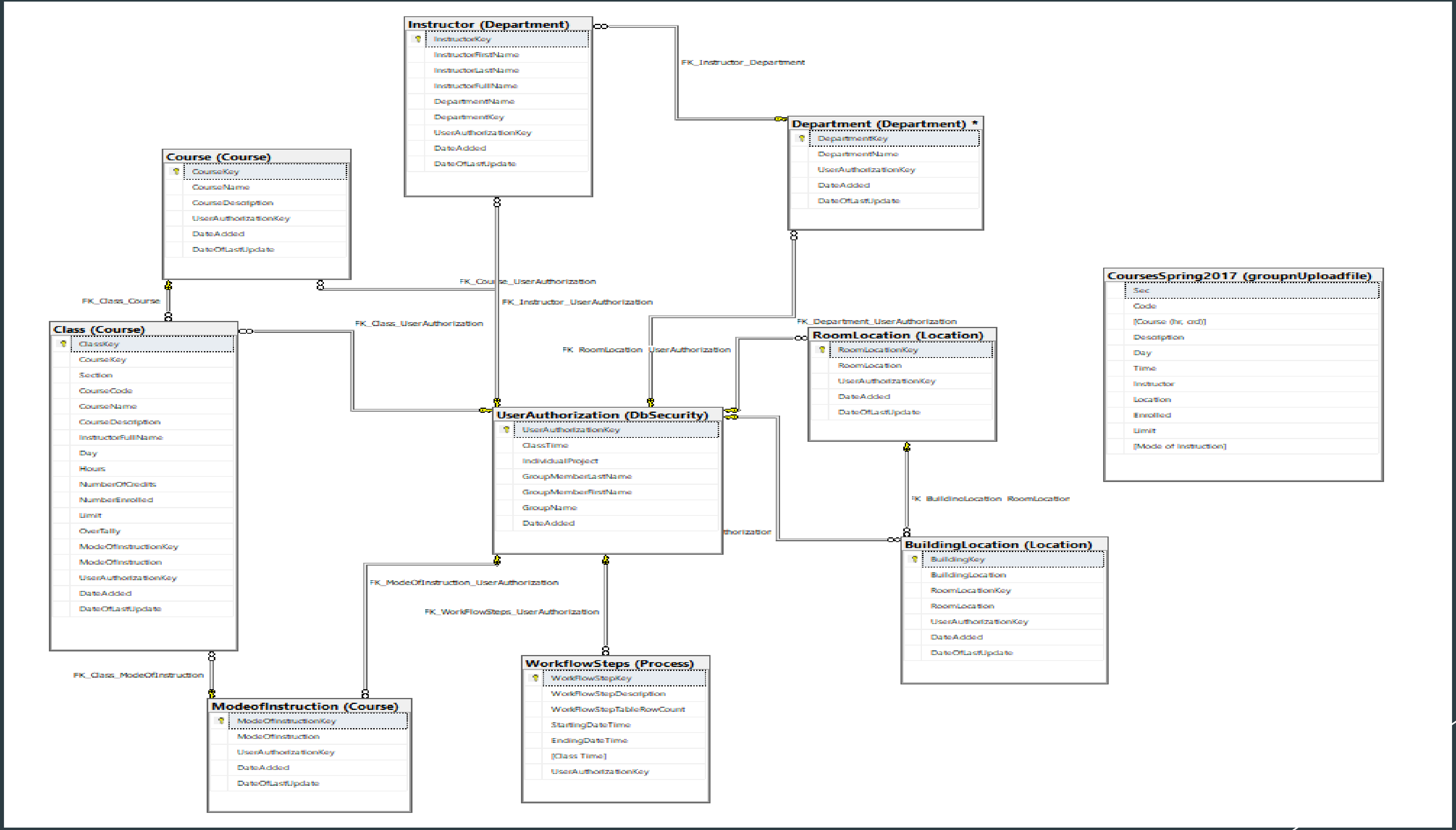
Note: Diagram is included with the project

Conceptual Data Model


Instructor (Department
FK_Instructor_Department
Department (Departme
Course (Course)
FK_Class_Course
FK_Course_UserAuthorization
FK_Instructor_UserAuthorization
FK_Department_UserAuthorization
CoursesSpring2017 (gr
Class (Course)
FK_Class_UserAuthorization
RoomLocation (Locatic
FK_RoomLocation_UserAuthorization
UserAuthorization (Db
FK_BuildingLocation_RoomLocation
FK_BuildingLocation_UserAuthorization
FK_Class_ModeOfInstruction
BuildingLocation (Loca
FK_WorkFlowSteps_UserAuthorization
FK_ModeOfInstruction_UserAuthorization
WorkflowSteps (Proce
ModeofInstruction (Co

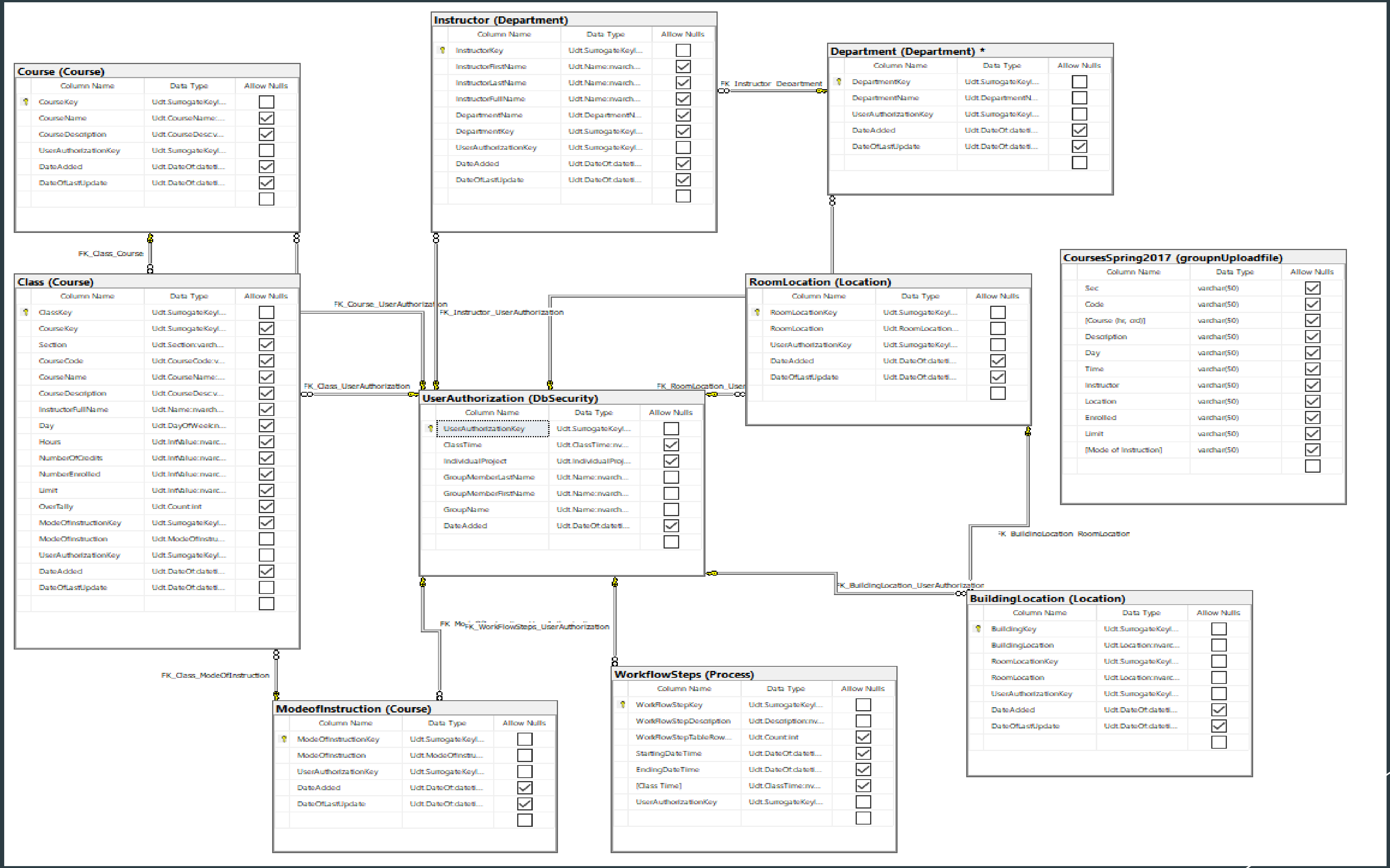
Note: Diagram is included with the project

Logical Data Model



Note: Diagram is included with the project

Physical Data Model


FK_Class_Course
FK_Course_UserAuthorization
FK_Instructor_UserAuthorization
FK_Instructor_Department
FK_Class_UserAuthorization
FK_RoomLocation_User
FK_BuildingLocation_RoomLocation
FK_BuildingLocation_UserAuthorization
FK_Mo
FK_WorkFlowSteps_UserAuthorization
FK_Class_ModeOfInstruction

Course (Course)

	Column Name	Data Type	Allow Nulls
🔑	CourseKey	Udt.SurrogateKeyI...	☐
	CourseName	Udt.CourseName:...	✓
	CourseDescription	Udt.CourseDesc:v...	✓
	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
	DateAdded	Udt.DateOf:dateti...	✓
	DateOfLastUpdate	Udt.DateOf:dateti...	✓
			☐

Instructor (Department)

	Column Name	Data Type	Allow Nulls
🔑	InstructorKey	Udt.SurrogateKeyI...	☐
	InstructorFirstName	Udt.Name:nvarch...	✓
	InstructorLastName	Udt.Name:nvarch...	✓
	InstructorFullName	Udt.Name:nvarch...	✓
	DepartmentName	Udt.DepartmentN...	✓
	DepartmentKey	Udt.SurrogateKeyI...	✓
	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
	DateAdded	Udt.DateOf:dateti...	✓
	DateOfLastUpdate	Udt.DateOf:dateti...	✓
			☐

Department (Department) *

	Column Name	Data Type	Allow Nulls
🔑	DepartmentKey	Udt.SurrogateKeyI...	☐
	DepartmentName	Udt.DepartmentN...	☐
	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
	DateAdded	Udt.DateOf:dateti...	✓
	DateOfLastUpdate	Udt.DateOf:dateti...	✓
			☐

Class (Course)

	Column Name	Data Type	Allow Nulls
🔑	ClassKey	Udt.SurrogateKeyI...	☐
	CourseKey	Udt.SurrogateKeyI...	✓
	Section	Udt.Section:varch...	✓
	CourseCode	Udt.CourseCode:v...	✓
	CourseName	Udt.CourseName:...	✓
	CourseDescription	Udt.CourseDesc:v...	✓
	InstructorFullName	Udt.Name:nvarch...	✓
	Day	Udt.DayOfWeek:n...	✓
	Hours	Udt.IntValue:nvarc...	✓
	NumberOfCredits	Udt.IntValue:nvarc...	✓
	NumberEnrolled	Udt.IntValue:nvarc...	✓
	Limit	Udt.IntValue:nvarc...	✓
	OverTally	Udt.Count:int	✓
	ModeOfInstructionKey	Udt.SurrogateKeyI...	✓
	ModeOfInstruction	Udt.ModeOfInstru...	☐
	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
	DateAdded	Udt.DateOf:dateti...	✓
	DateOfLastUpdate	Udt.DateOf:dateti...	☐
			☐

RoomLocation (Location)

	Column Name	Data Type	Allow Nulls
🔑	RoomLocationKey	Udt.SurrogateKeyI...	☐
	RoomLocation	Udt.RoomLocation...	☐
	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
	DateAdded	Udt.DateOf:dateti...	✓
	DateOfLastUpdate	Udt.DateOf:dateti...	✓
			☐

CoursesSpring2017 (groupnUploadfile)

	Column Name	Data Type	Allow Nulls
	Sec	varchar(50)	✓
	Code	varchar(50)	✓
	[Course (hr, crd)]	varchar(50)	✓
	Description	varchar(50)	✓
	Day	varchar(50)	✓
	Time	varchar(50)	✓
	Instructor	varchar(50)	✓
	Location	varchar(50)	✓
	Enrolled	varchar(50)	✓
	Limit	varchar(50)	✓
	[Mode of Instruction]	varchar(50)	✓
			☐

UserAuthorization (DbSecurity)

	Column Name	Data Type	Allow Nulls
🔑	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
	ClassTime	Udt.ClassTime:nv...	✓
	IndividualProject	Udt.IndividualProj...	✓
	GroupMemberLastName	Udt.Name:nvarch...	☐
	GroupMemberFirstName	Udt.Name:nvarch...	☐
	GroupName	Udt.Name:nvarch...	☐
	DateAdded	Udt.DateOf:dateti...	✓
			☐

BuildingLocation (Location)

	Column Name	Data Type	Allow Nulls
🔑	BuildingKey	Udt.SurrogateKeyI...	☐
	BuildingLocation	Udt.Location:nvarc...	☐
	RoomLocationKey	Udt.SurrogateKeyI...	☐
	RoomLocation	Udt.Location:nvarc...	☐
	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
	DateAdded	Udt.DateOf:dateti...	✓
	DateOfLastUpdate	Udt.DateOf:dateti...	✓
			☐

WorkflowSteps (Process)

	Column Name	Data Type	Allow Nulls
🔑	WorkFlowStepKey	Udt.SurrogateKeyI...	☐
	WorkFlowStepDescription	Udt.Description:nv...	☐
	WorkFlowStepTableRow...	Udt.Count:int	✓
	StartingDateTime	Udt.DateOf:dateti...	✓
	EndingDateTime	Udt.DateOf:dateti...	✓
	[Class Time]	Udt.ClassTime:nv...	✓
	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
			☐

ModeofInstruction (Course)

	Column Name	Data Type	Allow Nulls
🔑	ModeOfInstructionKey	Udt.SurrogateKeyI...	☐
	ModeOfInstruction	Udt.ModeOfInstru...	☐
	UserAuthorizationKey	Udt.SurrogateKeyI...	☐
	DateAdded	Udt.DateOf:dateti...	✓
	DateOfLastUpdate	Udt.DateOf:dateti...	✓
			☐

Note: Diagram is included with the project

Tables & Stored Procedures

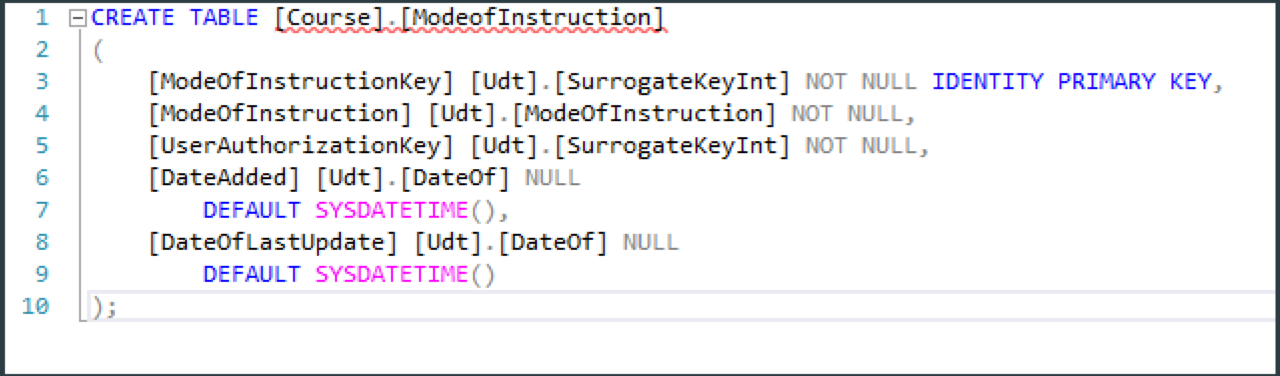
Erik’s Tables & Stored Procedures

- ModeOfInstruction table
- ModeOfInstruction Stored procedure
- DBSecurity table
- DBSecurity Stored procedure
- ProcessWorkFlow table
- ProcessWorkFlow Stored procedure
- AddForeignKeys Stored procedure
- DropForeignKeys Stored procedure
- TruncateData Stored procedure
- ShowTableStatusRowCount Stored procedure
- LoadQueensCourseSchedule Stored Procedure

Erik’s Data Anomalies and Fix

- ModeOfInstruction had a data anomaly where there were blanks at the end
- To fix them, I coalesced them to TBA so that there would not only be no blank spaces, but a placeholder to default to when modes of instruction were not yet declared for certain classes

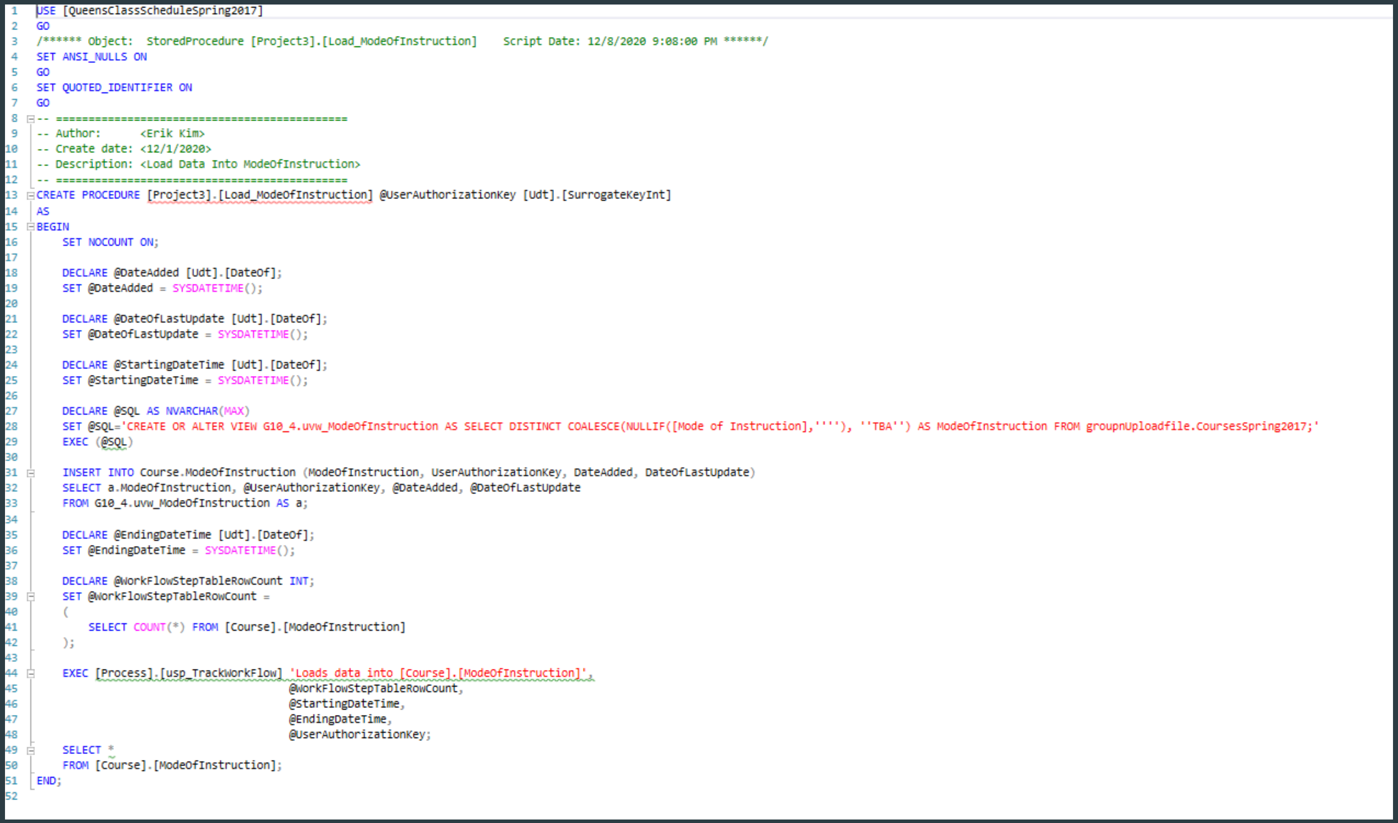
ModeOfInstruction Table Code



```
CREATE TABLE [Course].[ModeofInstruction]
(
    [ModeOfInstructionKey] [Udt].[SurrogateKeyInt] NOT NULL IDENTITY PRIMARY KEY,
    [ModeOfInstruction] [Udt].[ModeOfInstruction] NOT NULL,
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] NOT NULL,
    [DateAdded] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME(),
    [DateOfLastUpdate] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME()
);
```

Note: Code is provided in the notes tab on this slide

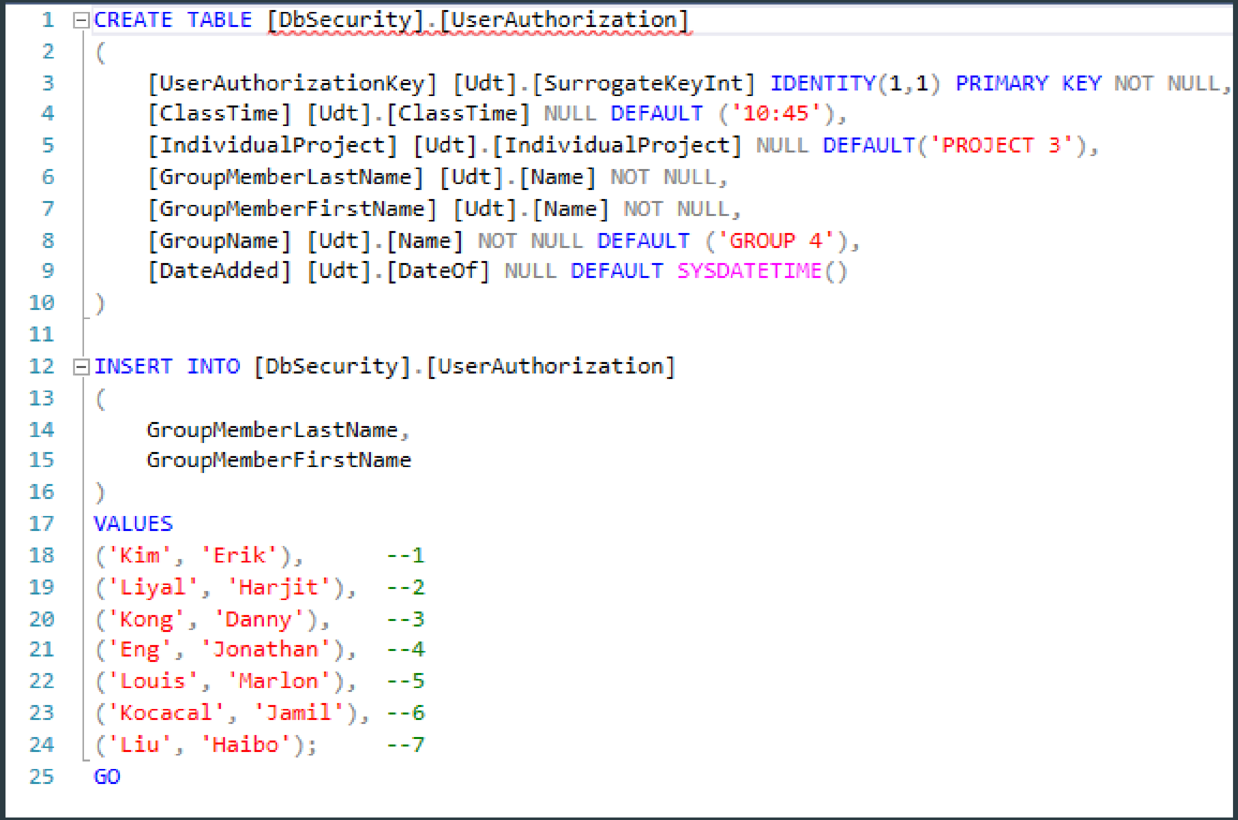
ModeOfInstruction Stored Procedure Code



```sql
USE [QueensClassScheduleSpring2017]
GO
/****** Object:  StoredProcedure [Project3].[Load_ModeOfInstruction]    Script Date: 12/8/2020 9:08:00 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Erik Kim>
-- Create date: <12/1/2020>
-- Description: <Load Data Into ModeOfInstruction>
-- =============================================
CREATE PROCEDURE [Project3].[Load_ModeOfInstruction] @UserAuthorizationKey [Udt].[SurrogateKeyInt]
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @DateAdded [Udt].[DateOf];
    SET @DateAdded = SYSDATETIME();

    DECLARE @DateOfLastUpdate [Udt].[DateOf];
    SET @DateOfLastUpdate = SYSDATETIME();

    DECLARE @StartingDateTime [Udt].[DateOf];
    SET @StartingDateTime = SYSDATETIME();

    DECLARE @SQL AS NVARCHAR(MAX)
    SET @SQL='CREATE OR ALTER VIEW G10_4.uvw_ModeOfInstruction AS SELECT DISTINCT COALESCE(NULLIF([Mode of Instruction],''''), ''TBA'') AS ModeOfInstruction FROM groupnUploadfile.CoursesSpring2017;'
    EXEC (@SQL)

    INSERT INTO Course.ModeOfInstruction (ModeOfInstruction, UserAuthorizationKey, DateAdded, DateOfLastUpdate)
    SELECT a.ModeOfInstruction, @UserAuthorizationKey, @DateAdded, @DateOfLastUpdate
    FROM G10_4.uvw_ModeOfInstruction AS a;

    DECLARE @EndingDateTime [Udt].[DateOf];
    SET @EndingDateTime = SYSDATETIME();

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount =
    (
        SELECT COUNT(*) FROM [Course].[ModeOfInstruction]
    );

    EXEC [Process].[usp_TrackWorkFlow] 'Loads data into [Course].[ModeOfInstruction]',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;

    SELECT *
    FROM [Course].[ModeOfInstruction];
END;
```

Note: Code is provided in the notes tab on this slide

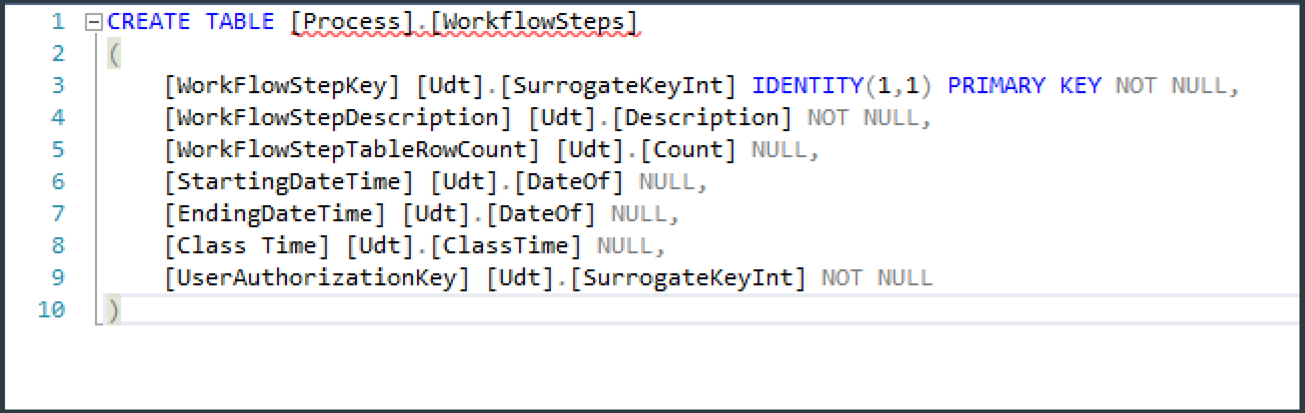
DBSecurity Table Code



```sql
CREATE TABLE [DbSecurity].[UserAuthorization]
(
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] IDENTITY(1,1) PRIMARY KEY NOT NULL,
    [ClassTime] [Udt].[ClassTime] NULL DEFAULT ('10:45'),
    [IndividualProject] [Udt].[IndividualProject] NULL DEFAULT('PROJECT 3'),
    [GroupMemberLastName] [Udt].[Name] NOT NULL,
    [GroupMemberFirstName] [Udt].[Name] NOT NULL,
    [GroupName] [Udt].[Name] NOT NULL DEFAULT ('GROUP 4'),
    [DateAdded] [Udt].[DateOf] NULL DEFAULT SYSDATETIME()
)

INSERT INTO [DbSecurity].[UserAuthorization]
(
    GroupMemberLastName,
    GroupMemberFirstName
)
VALUES
('Kim', 'Erik'),       --1
('Liyal', 'Harjit'),   --2
('Kong', 'Danny'),     --3
('Eng', 'Jonathan'),   --4
('Louis', 'Marlon'),   --5
('Kocacal', 'Jamil'), --6
('Liu', 'Haibo');      --7
GO
```

Note: Code is provided in the notes tab on this slide

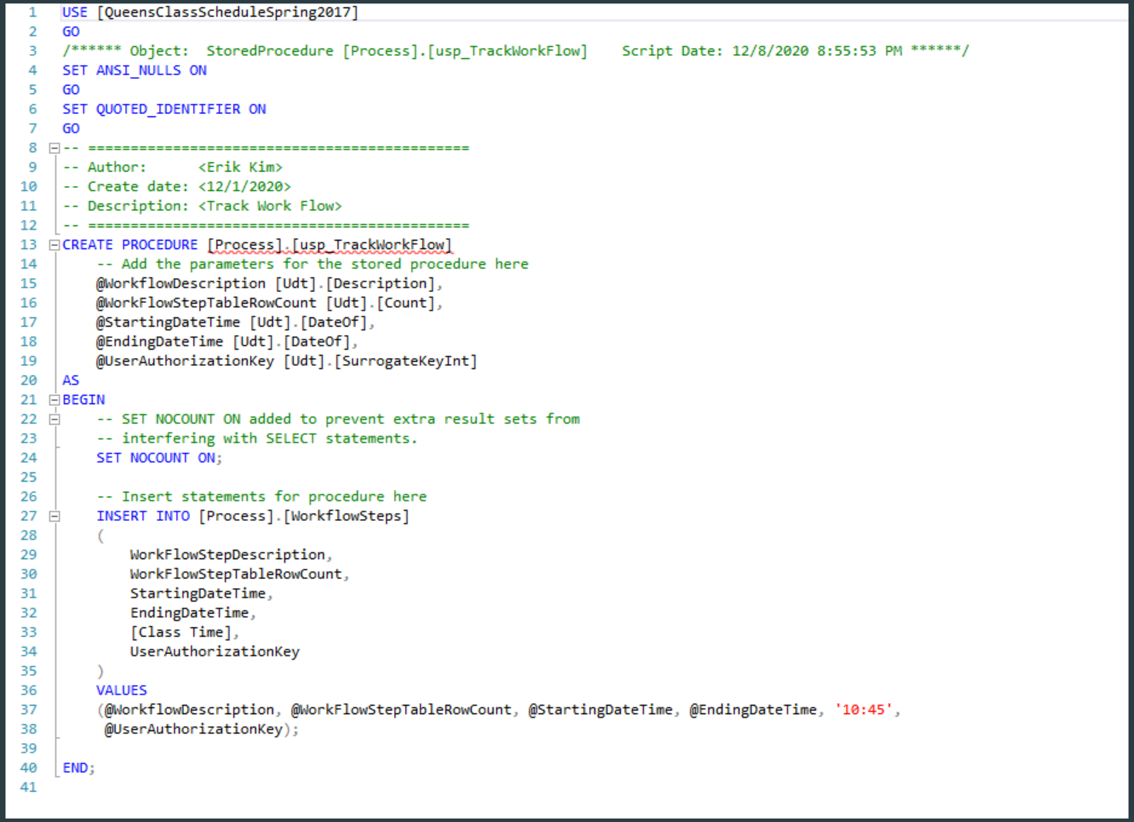
WorkFlow Table Code



```
CREATE TABLE [Process].[WorkflowSteps]
(
    [WorkFlowStepKey] [Udt].[SurrogateKeyInt] IDENTITY(1,1) PRIMARY KEY NOT NULL,
    [WorkFlowStepDescription] [Udt].[Description] NOT NULL,
    [WorkFlowStepTableRowCount] [Udt].[Count] NULL,
    [StartingDateTime] [Udt].[DateOf] NULL,
    [EndingDateTime] [Udt].[DateOf] NULL,
    [Class Time] [Udt].[ClassTime] NULL,
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] NOT NULL
)
```

Note: Code is provided in the notes tab on this slide

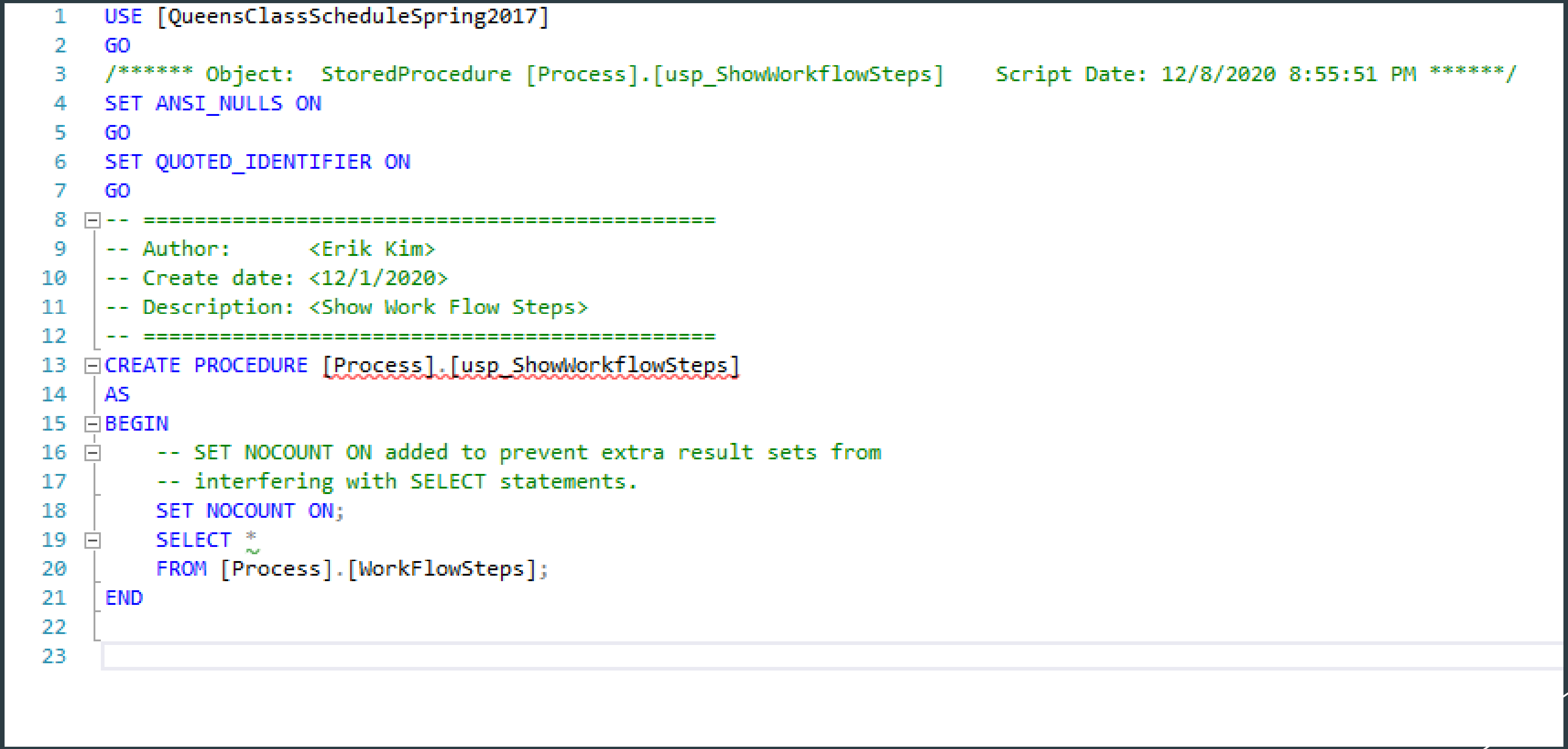
TrackWorkFlow Stored Procedure Code



```sql
USE [QueensClassScheduleSpring2017]
GO
/****** Object:  StoredProcedure [Process].[usp_TrackWorkFlow]    Script Date: 12/8/2020 8:55:53 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Erik Kim>
-- Create date: <12/1/2020>
-- Description: <Track Work Flow>
-- =============================================
CREATE PROCEDURE [Process].[usp_TrackWorkFlow]
    -- Add the parameters for the stored procedure here
    @WorkflowDescription [Udt].[Description],
    @WorkFlowStepTableRowCount [Udt].[Count],
    @StartingDateTime [Udt].[DateOf],
    @EndingDateTime [Udt].[DateOf],
    @UserAuthorizationKey [Udt].[SurrogateKeyInt]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    INSERT INTO [Process].[WorkflowSteps]
    (
        WorkFlowStepDescription,
        WorkFlowStepTableRowCount,
        StartingDateTime,
        EndingDateTime,
        [Class Time],
        UserAuthorizationKey
    )
    VALUES
    (@WorkflowDescription, @WorkFlowStepTableRowCount, @StartingDateTime, @EndingDateTime, '10:45',
     @UserAuthorizationKey);

END;
```

Note: Code is provided in the notes tab on this slide

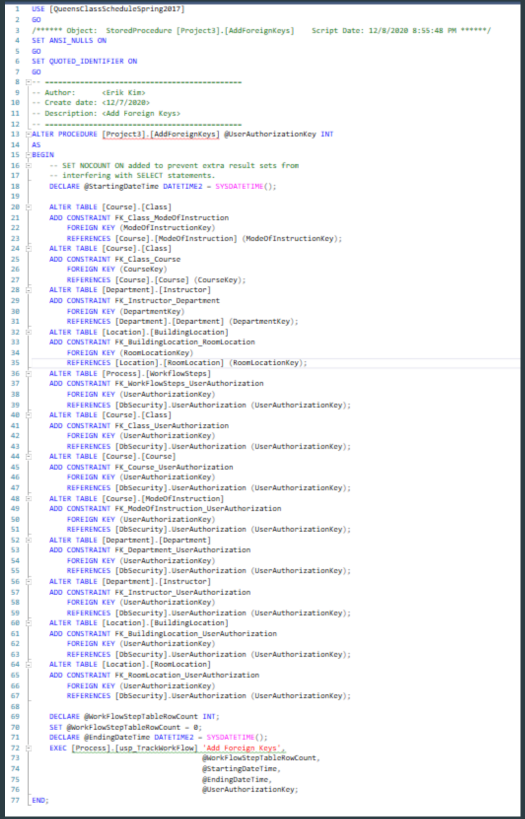
ShowWorkFlowSteps Stored Procedure Code



```
USE [QueensClassScheduleSpring2017]
GO
/****** Object:  StoredProcedure [Process].[usp_ShowWorkflowSteps]    Script Date: 12/8/2020 8:55:51 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Erik Kim>
-- Create date: <12/1/2020>
-- Description: <Show Work Flow Steps>
-- =============================================
CREATE PROCEDURE [Process].[usp_ShowWorkflowSteps]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    SELECT *
    FROM [Process].[WorkFlowSteps];
END


```

Note: Code is provided in the notes tab on this slide

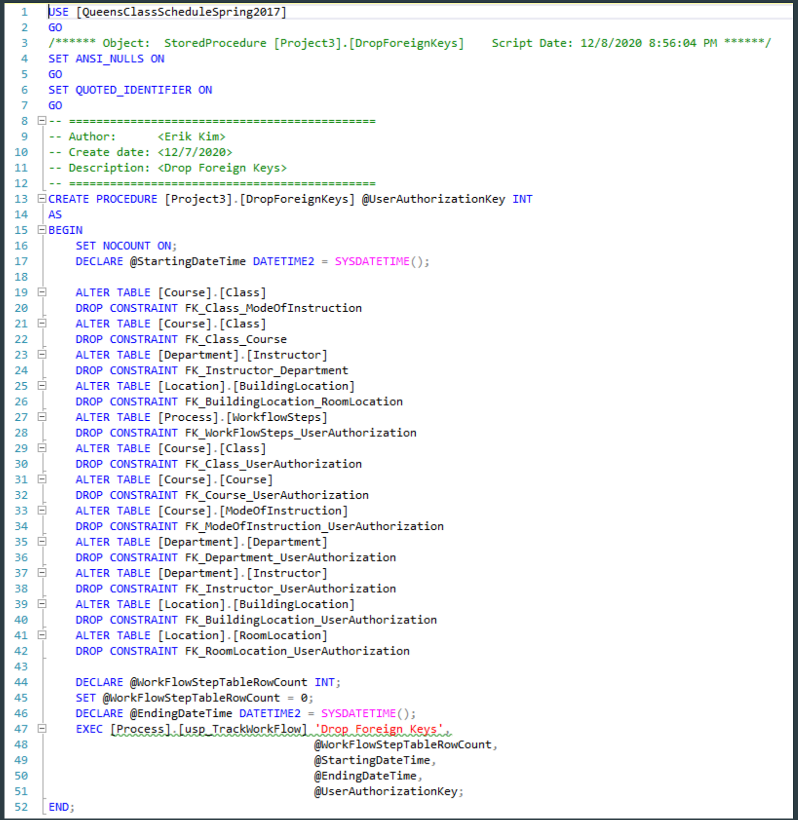
AddForeignKeys Stored Procedure Code



```sql
USE [QueensClassScheduleSpring2017]
GO
/****** Object:  StoredProcedure [Project3].[AddForeignKeys]    Script Date: 12/8/2020 8:55:48 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Erik Kim>
-- Create date: <12/7/2020>
-- Description: <Add Foreign Keys>
-- =============================================
ALTER PROCEDURE [Project3].[AddForeignKeys] @UserAuthorizationKey INT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    DECLARE @StartingDateTime DATETIME2 = SYSDATETIME();

    ALTER TABLE [Course].[Class]
    ADD CONSTRAINT FK_Class_ModeOfInstruction
        FOREIGN KEY (ModeOfInstructionKey)
        REFERENCES [Course].[ModeOfInstruction] (ModeOfInstructionKey);
    ALTER TABLE [Course].[Class]
    ADD CONSTRAINT FK_Class_Course
        FOREIGN KEY (CourseKey)
        REFERENCES [Course].[Course] (CourseKey);
    ALTER TABLE [Department].[Instructor]
    ADD CONSTRAINT FK_Instructor_Department
        FOREIGN KEY (DepartmentKey)
        REFERENCES [Department].[Department] (DepartmentKey);
    ALTER TABLE [Location].[BuildingLocation]
    ADD CONSTRAINT FK_BuildingLocation_RoomLocation
        FOREIGN KEY (RoomLocationKey)
        REFERENCES [Location].[RoomLocation] (RoomLocationKey);
    ALTER TABLE [Process].[WorkflowSteps]
    ADD CONSTRAINT FK_WorkFlowSteps_UserAuthorization
        FOREIGN KEY (UserAuthorizationKey)
        REFERENCES [DbSecurity].UserAuthorization (UserAuthorizationKey);
    ALTER TABLE [Course].[Class]
    ADD CONSTRAINT FK_Class_UserAuthorization
        FOREIGN KEY (UserAuthorizationKey)
        REFERENCES [DbSecurity].UserAuthorization (UserAuthorizationKey);
    ALTER TABLE [Course].[Course]
    ADD CONSTRAINT FK_Course_UserAuthorization
        FOREIGN KEY (UserAuthorizationKey)
        REFERENCES [DbSecurity].UserAuthorization (UserAuthorizationKey);
    ALTER TABLE [Course].[ModeOfInstruction]
    ADD CONSTRAINT FK_ModeOfInstruction_UserAuthorization
        FOREIGN KEY (UserAuthorizationKey)
        REFERENCES [DbSecurity].UserAuthorization (UserAuthorizationKey);
    ALTER TABLE [Department].[Department]
    ADD CONSTRAINT FK_Department_UserAuthorization
        FOREIGN KEY (UserAuthorizationKey)
        REFERENCES [DbSecurity].UserAuthorization (UserAuthorizationKey);
    ALTER TABLE [Department].[Instructor]
    ADD CONSTRAINT FK_Instructor_UserAuthorization
        FOREIGN KEY (UserAuthorizationKey)
        REFERENCES [DbSecurity].UserAuthorization (UserAuthorizationKey);
    ALTER TABLE [Location].[BuildingLocation]
    ADD CONSTRAINT FK_BuildingLocation_UserAuthorization
        FOREIGN KEY (UserAuthorizationKey)
        REFERENCES [DbSecurity].UserAuthorization (UserAuthorizationKey);
    ALTER TABLE [Location].[RoomLocation]
    ADD CONSTRAINT FK_RoomLocation_UserAuthorization
        FOREIGN KEY (UserAuthorizationKey)
        REFERENCES [DbSecurity].UserAuthorization (UserAuthorizationKey);

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = 0;
    DECLARE @EndingDateTime DATETIME2 = SYSDATETIME();
    EXEC [Process].[usp_TrackWorkFlow] 'Add Foreign Keys',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;
END;
```

Note: Code is provided in the notes tab on this slide

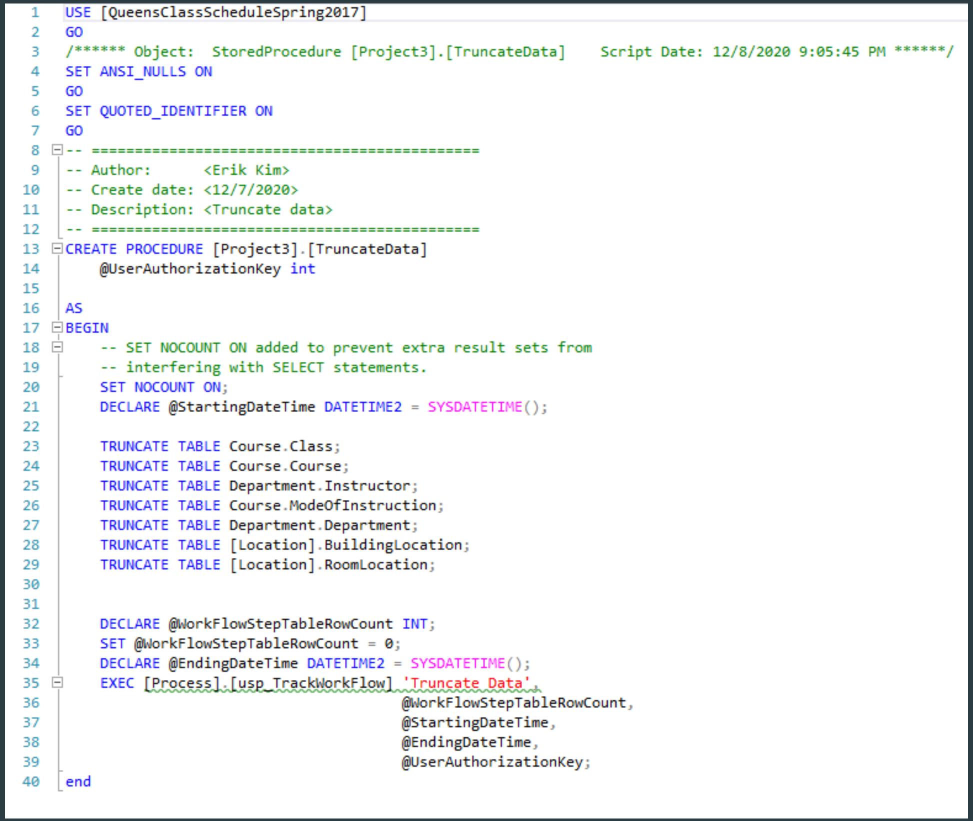
DropForeignKeys Stored Procedure Code



```sql
USE [QueensClassScheduleSpring2017]
GO
/****** Object:  StoredProcedure [Project3].[DropForeignKeys]    Script Date: 12/8/2020 8:56:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Erik Kim>
-- Create date: <12/7/2020>
-- Description: <Drop Foreign Keys>
-- =============================================
CREATE PROCEDURE [Project3].[DropForeignKeys] @UserAuthorizationKey INT
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @StartingDateTime DATETIME2 = SYSDATETIME();

    ALTER TABLE [Course].[Class]
    DROP CONSTRAINT FK_Class_ModeOfInstruction
    ALTER TABLE [Course].[Class]
    DROP CONSTRAINT FK_Class_Course
    ALTER TABLE [Department].[Instructor]
    DROP CONSTRAINT FK_Instructor_Department
    ALTER TABLE [Location].[BuildingLocation]
    DROP CONSTRAINT FK_BuildingLocation_RoomLocation
    ALTER TABLE [Process].[WorkflowSteps]
    DROP CONSTRAINT FK_WorkFlowSteps_UserAuthorization
    ALTER TABLE [Course].[Class]
    DROP CONSTRAINT FK_Class_UserAuthorization
    ALTER TABLE [Course].[Course]
    DROP CONSTRAINT FK_Course_UserAuthorization
    ALTER TABLE [Course].[ModeOfInstruction]
    DROP CONSTRAINT FK_ModeOfInstruction_UserAuthorization
    ALTER TABLE [Department].[Department]
    DROP CONSTRAINT FK_Department_UserAuthorization
    ALTER TABLE [Department].[Instructor]
    DROP CONSTRAINT FK_Instructor_UserAuthorization
    ALTER TABLE [Location].[BuildingLocation]
    DROP CONSTRAINT FK_BuildingLocation_UserAuthorization
    ALTER TABLE [Location].[RoomLocation]
    DROP CONSTRAINT FK_RoomLocation_UserAuthorization

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = 0;
    DECLARE @EndingDateTime DATETIME2 = SYSDATETIME();
    EXEC [Process].[usp_TrackWorkFlow] 'Drop Foreign Keys',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;
END;
```

Note: Code is provided in the notes tab on this slide

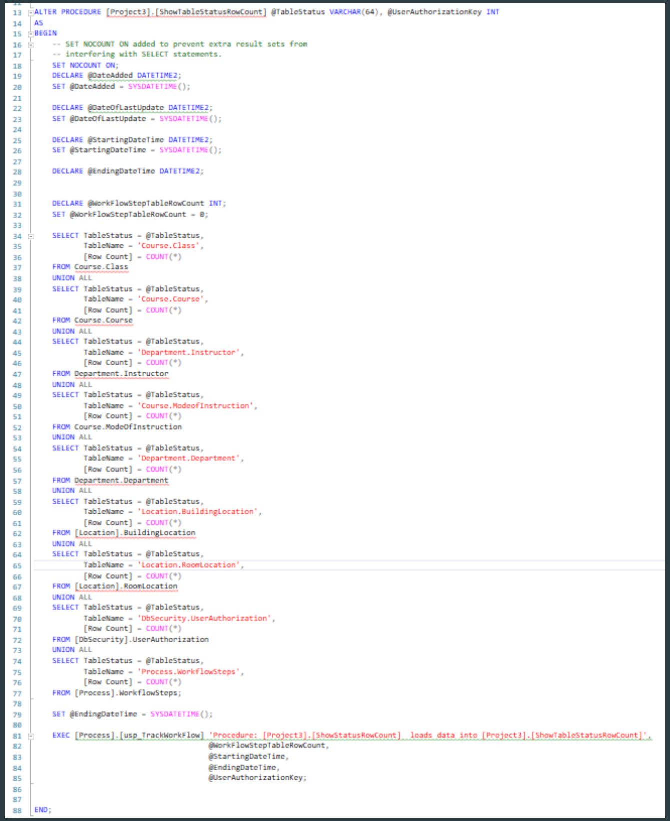
TruncateData Stored Procedure Code



```sql
USE [QueensClassScheduleSpring2017]
GO
/****** Object:  StoredProcedure [Project3].[TruncateData]    Script Date: 12/8/2020 9:05:45 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Erik Kim>
-- Create date: <12/7/2020>
-- Description: <Truncate data>
-- =============================================
CREATE PROCEDURE [Project3].[TruncateData]
    @UserAuthorizationKey int

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    DECLARE @StartingDateTime DATETIME2 = SYSDATETIME();

    TRUNCATE TABLE Course.Class;
    TRUNCATE TABLE Course.Course;
    TRUNCATE TABLE Department.Instructor;
    TRUNCATE TABLE Course.ModeOfInstruction;
    TRUNCATE TABLE Department.Department;
    TRUNCATE TABLE [Location].BuildingLocation;
    TRUNCATE TABLE [Location].RoomLocation;


    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = 0;
    DECLARE @EndingDateTime DATETIME2 = SYSDATETIME();
    EXEC [Process].[usp_TrackWorkFlow] 'Truncate Data',
                                    @WorkFlowStepTableRowCount,
                                    @StartingDateTime,
                                    @EndingDateTime,
                                    @UserAuthorizationKey;
end
```

Note: Code is provided in the notes tab on this slide

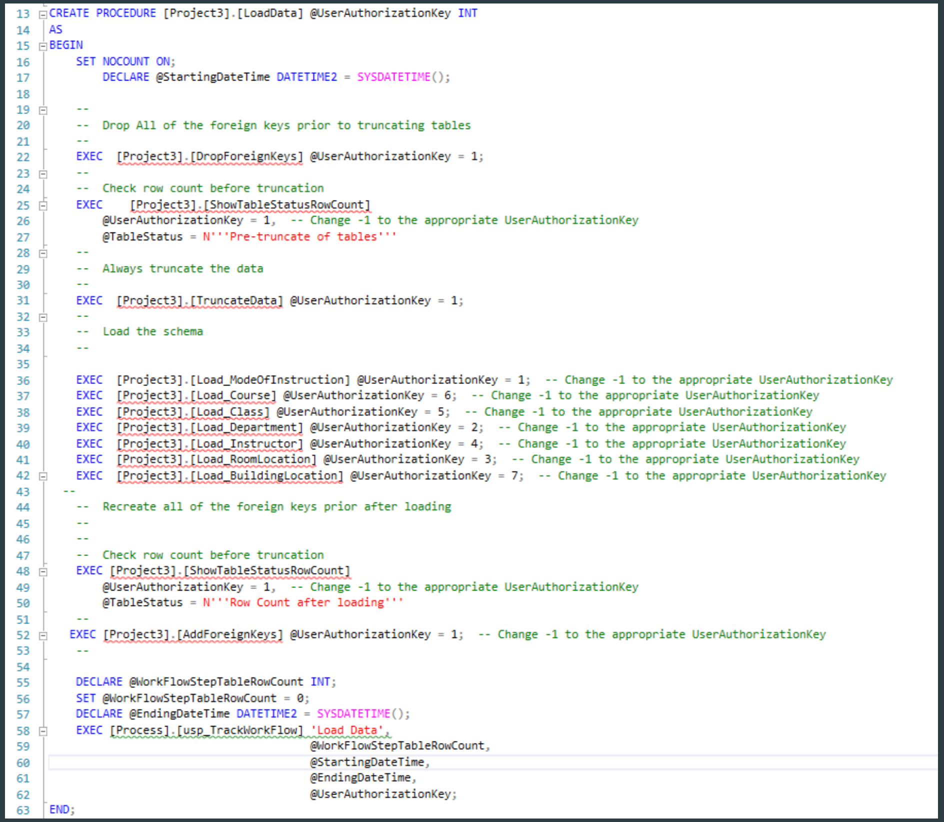
ShowTableStatusRowCount Stored Procedure Code



```sql
ALTER PROCEDURE [Project3].[ShowTableStatusRowCount] @TableStatus VARCHAR(64), @UserAuthorizationKey INT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    DECLARE @DateAdded DATETIME2;
    SET @DateAdded = SYSDATETIME();

    DECLARE @DateOfLastUpdate DATETIME2;
    SET @DateOfLastUpdate = SYSDATETIME();

    DECLARE @StartingDateTime DATETIME2;
    SET @StartingDateTime = SYSDATETIME();

    DECLARE @EndingDateTime DATETIME2;


    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = 0;

    SELECT TableStatus = @TableStatus,
           TableName = 'Course.Class',
           [Row Count] = COUNT(*)
    FROM Course.Class
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'Course.Course',
           [Row Count] = COUNT(*)
    FROM Course.Course
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'Department.Instructor',
           [Row Count] = COUNT(*)
    FROM Department.Instructor
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'Course.ModeofInstruction',
           [Row Count] = COUNT(*)
    FROM Course.ModeOfInstruction
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'Department.Department',
           [Row Count] = COUNT(*)
    FROM Department.Department
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'Location.BuildingLocation',
           [Row Count] = COUNT(*)
    FROM [Location].BuildingLocation
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'Location.RoomLocation',
           [Row Count] = COUNT(*)
    FROM [Location].RoomLocation
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'DbSecurity.UserAuthorization',
           [Row Count] = COUNT(*)
    FROM [DbSecurity].UserAuthorization
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'Process.WorkflowSteps',
           [Row Count] = COUNT(*)
    FROM [Process].WorkflowSteps;

    SET @EndingDateTime = SYSDATETIME();

    EXEC [Process].[usp_TrackWorkFlow] 'Procedure: [Project3].[ShowStatusRowCount]  loads data into [Project3].[ShowTableStatusRowCount]',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;


END;
```

Note: Code is provided in the notes tab on this slide

LoadQueensCourseSchedule Stored Procedure Code



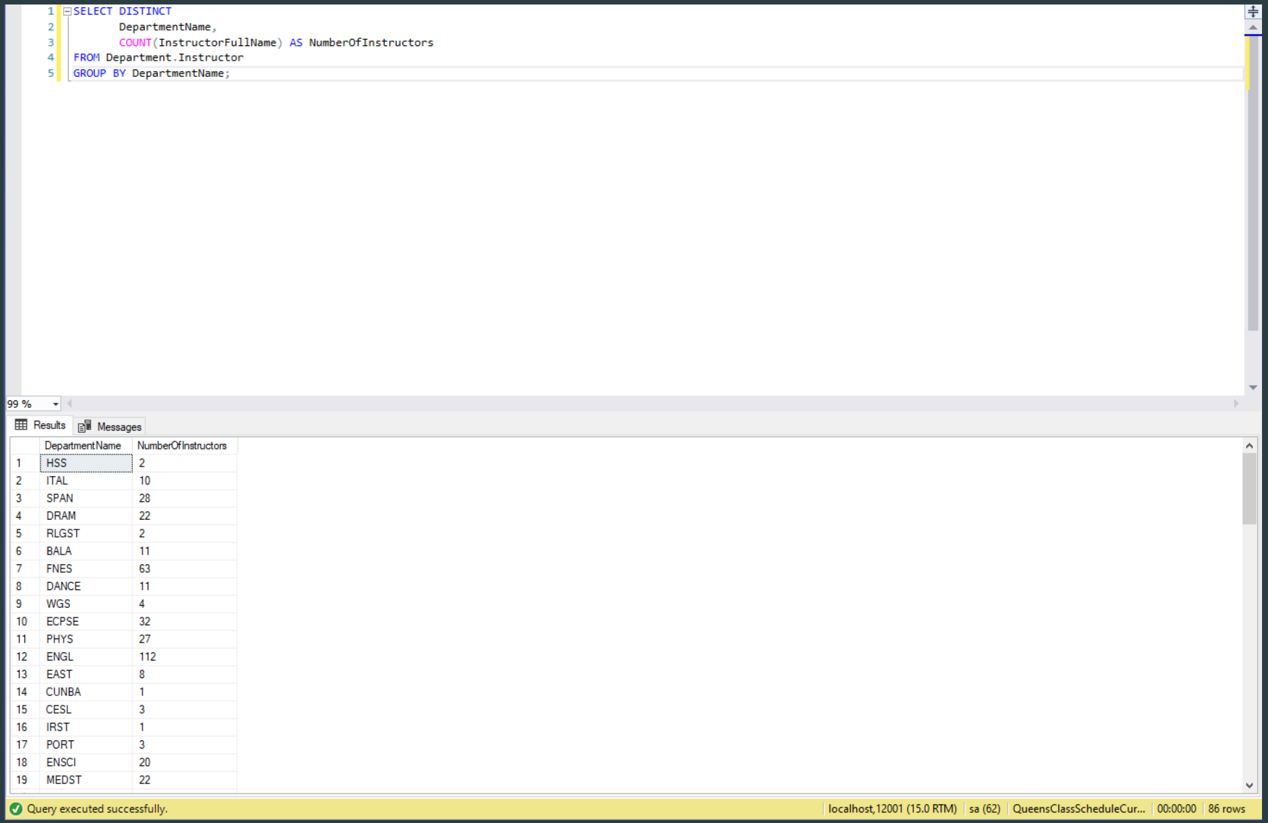
```sql
CREATE PROCEDURE [Project3].[LoadData] @UserAuthorizationKey INT
AS
BEGIN
    SET NOCOUNT ON;
        DECLARE @StartingDateTime DATETIME2 = SYSDATETIME();

    --
    --  Drop All of the foreign keys prior to truncating tables
    --
    EXEC  [Project3].[DropForeignKeys] @UserAuthorizationKey = 1;
    --
    --  Check row count before truncation
    EXEC    [Project3].[ShowTableStatusRowCount]
        @UserAuthorizationKey = 1,  -- Change -1 to the appropriate UserAuthorizationKey
        @TableStatus = N'''Pre-truncate of tables'''
    --
    --  Always truncate the data
    --
    EXEC  [Project3].[TruncateData] @UserAuthorizationKey = 1;
    --
    --  Load the schema
    --

    EXEC  [Project3].[Load_ModeOfInstruction] @UserAuthorizationKey = 1;  -- Change -1 to the appropriate UserAuthorizationKey
    EXEC  [Project3].[Load_Course] @UserAuthorizationKey = 6;  -- Change -1 to the appropriate UserAuthorizationKey
    EXEC  [Project3].[Load_Class] @UserAuthorizationKey = 5;  -- Change -1 to the appropriate UserAuthorizationKey
    EXEC  [Project3].[Load_Department] @UserAuthorizationKey = 2;  -- Change -1 to the appropriate UserAuthorizationKey
    EXEC  [Project3].[Load_Instructor] @UserAuthorizationKey = 4;  -- Change -1 to the appropriate UserAuthorizationKey
    EXEC  [Project3].[Load_RoomLocation] @UserAuthorizationKey = 3;  -- Change -1 to the appropriate UserAuthorizationKey
    EXEC  [Project3].[Load_BuildingLocation] @UserAuthorizationKey = 7;  -- Change -1 to the appropriate UserAuthorizationKey
  --
    --  Recreate all of the foreign keys prior after loading
    --
    --
    --  Check row count before truncation
    EXEC [Project3].[ShowTableStatusRowCount]
        @UserAuthorizationKey = 1,  -- Change -1 to the appropriate UserAuthorizationKey
        @TableStatus = N'''Row Count after loading'''
    --
   EXEC [Project3].[AddForeignKeys] @UserAuthorizationKey = 1;  -- Change -1 to the appropriate UserAuthorizationKey
    --

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = 0;
    DECLARE @EndingDateTime DATETIME2 = SYSDATETIME();
    EXEC [Process].[usp_TrackWorkFlow] 'Load Data',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;
END;
```

Note: Code is provided in the notes tab on this slide

Erik’s Queries

- Query 2 Proposition: How many instructors are in each department?
- Free Query Proposition: How many changes has each user made to the database?

NumberOfInstructorsPerDepartment



```sql
SELECT DISTINCT
        DepartmentName,
        COUNT(InstructorFullName) AS NumberOfInstructors
FROM Department.Instructor
GROUP BY DepartmentName;
```

99 %

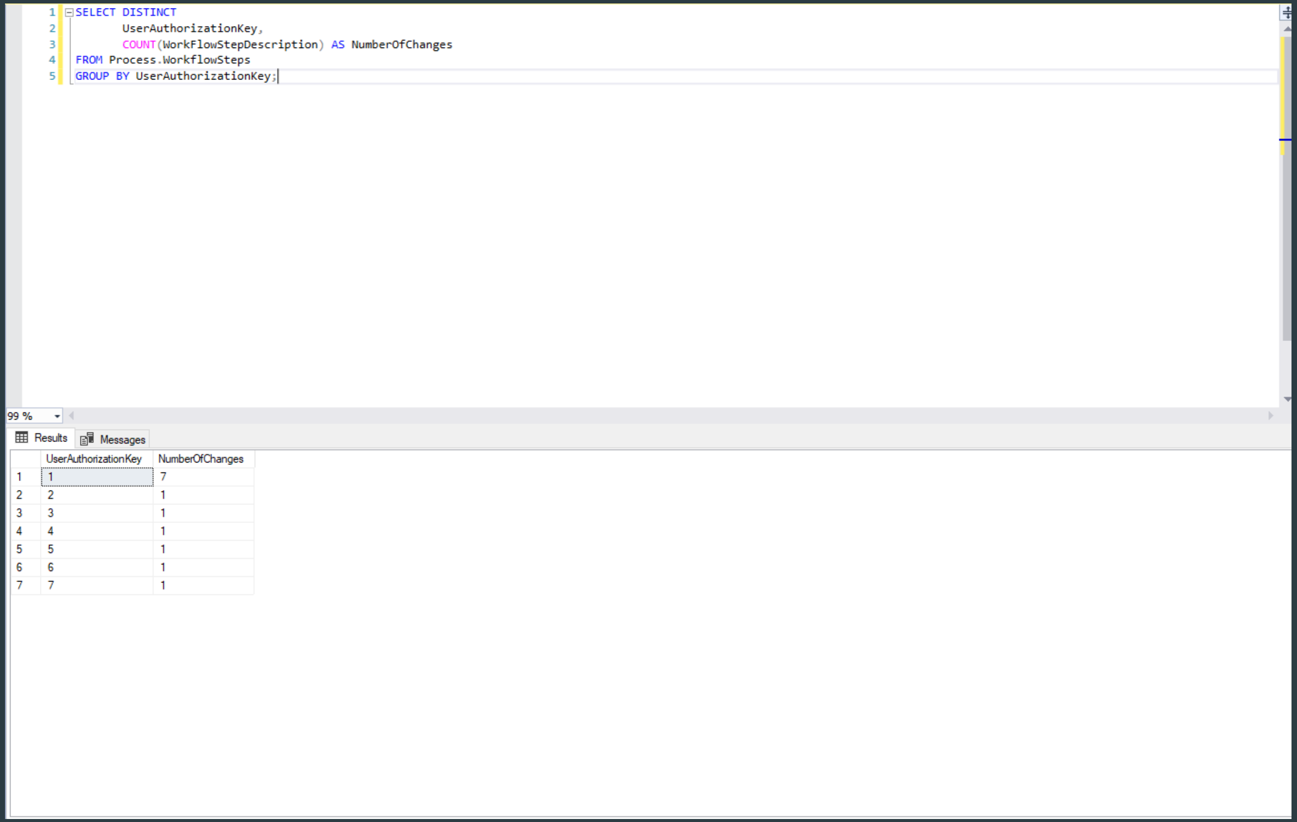
Results | Messages

	DepartmentName	NumberOfInstructors
1	HSS	2
2	ITAL	10
3	SPAN	28
4	DRAM	22
5	RLGST	2
6	BALA	11
7	FNES	63
8	DANCE	11
9	WGS	4
10	ECPSE	32
11	PHYS	27
12	ENGL	112
13	EAST	8
14	CUNBA	1
15	CESL	3
16	IRST	1
17	PORT	3
18	ENSCI	20
19	MEDST	22

Query executed successfully. | localhost,12001 (15.0 RTM) | sa (62) | QueensClassScheduleCur... | 00:00:00 | 86 rows

Note: Code is provided in the notes tab on this slide

NumberOfChangesMadePerUser



```sql
SELECT DISTINCT
       UserAuthorizationKey,
       COUNT(WorkFlowStepDescription) AS NumberOfChanges
FROM Process.WorkflowSteps
GROUP BY UserAuthorizationKey;
```

99 %

Results | Messages

	UserAuthorizationKey	NumberOfChanges
1	1	7
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1

Note: Code is provided in the notes tab on this slide

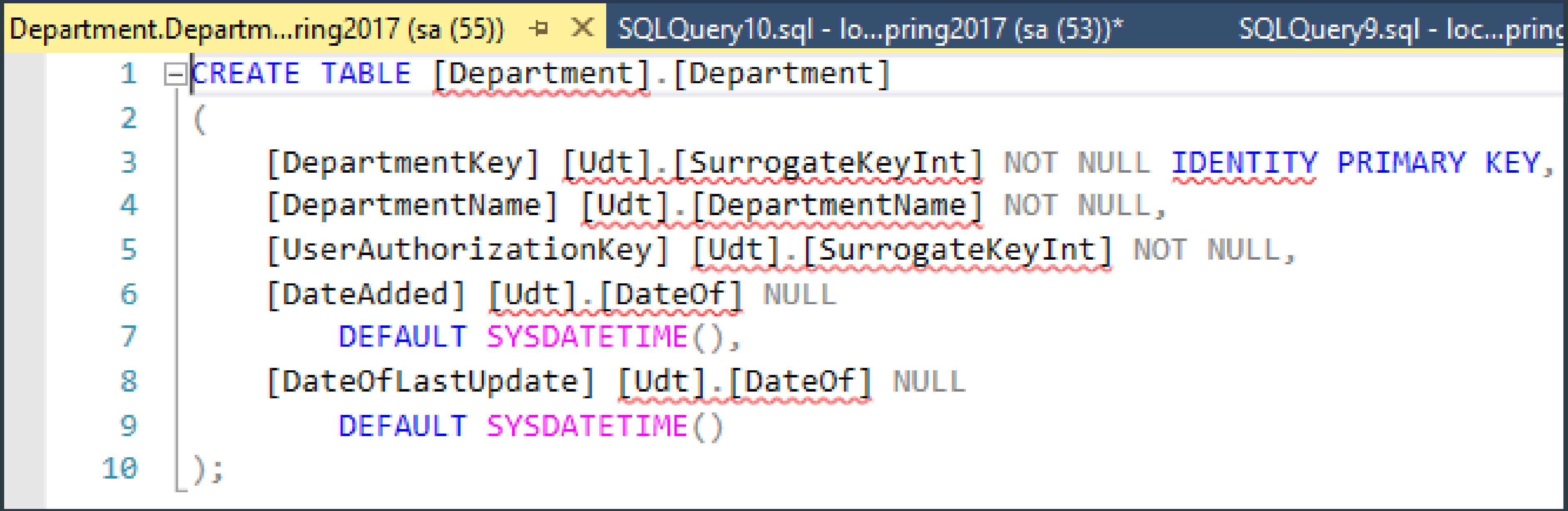
Harjit’s Tables & Stored Procedures

- Department Table
- Department stored procedure

Harjit’s Data Anomalies and Fixes

- Some department names were blank; department names are not allowed to be blank.
- To fix I coalesced them if they were null with “TBA” as a placeholder.

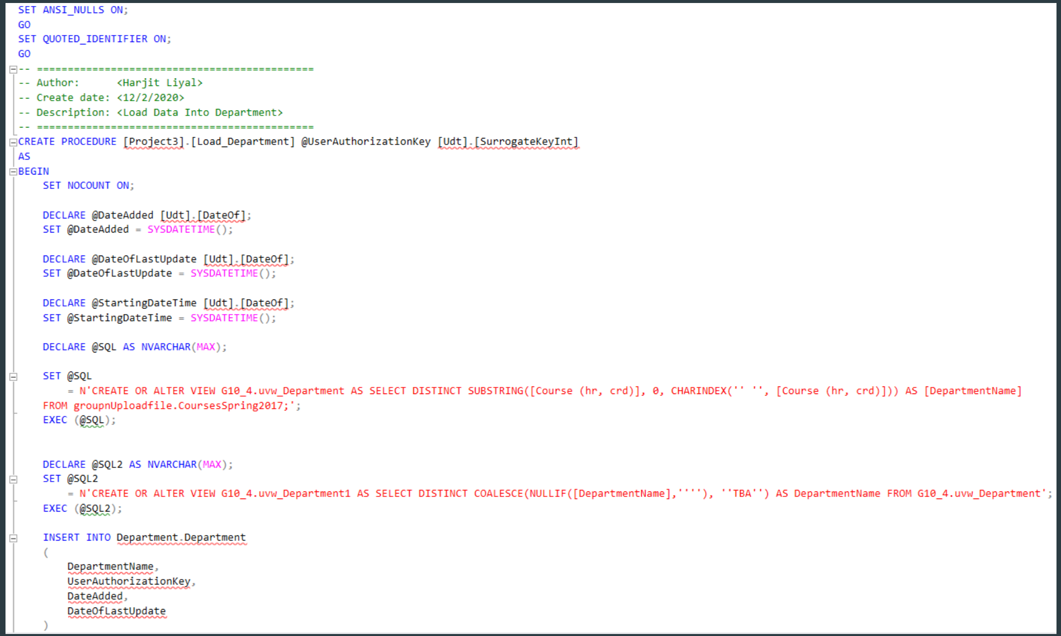
Department Table Code


Department.Departm...ring2017 (sa (55))
SQLQuery10.sql - lo...pring2017 (sa (53))*
SQLQuery9.sql - loc...pring

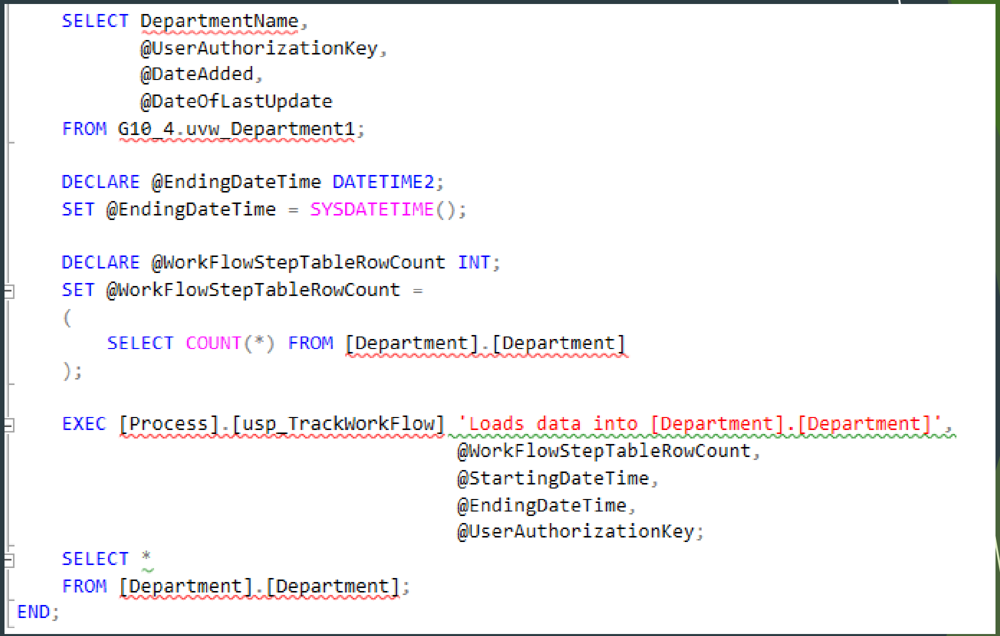
```sql
CREATE TABLE [Department].[Department]
(
    [DepartmentKey] [Udt].[SurrogateKeyInt] NOT NULL IDENTITY PRIMARY KEY,
    [DepartmentName] [Udt].[DepartmentName] NOT NULL,
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] NOT NULL,
    [DateAdded] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME(),
    [DateOfLastUpdate] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME()
);
```

Note: Code is provided in the notes tab on this slide

Department Stored Procedure Code



```sql
SET ANSI_NULLS ON;
GO
SET QUOTED_IDENTIFIER ON;
GO
-- =============================================
-- Author:      <Harjit Liyal>
-- Create date: <12/2/2020>
-- Description: <Load Data Into Department>
-- =============================================
CREATE PROCEDURE [Project3].[Load_Department] @UserAuthorizationKey [Udt].[SurrogateKeyInt]
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @DateAdded [Udt].[DateOf];
    SET @DateAdded = SYSDATETIME();

    DECLARE @DateOfLastUpdate [Udt].[DateOf];
    SET @DateOfLastUpdate = SYSDATETIME();

    DECLARE @StartingDateTime [Udt].[DateOf];
    SET @StartingDateTime = SYSDATETIME();

    DECLARE @SQL AS NVARCHAR(MAX);

    SET @SQL
        = N'CREATE OR ALTER VIEW G10_4.uvw_Department AS SELECT DISTINCT SUBSTRING([Course (hr, crd)], 0, CHARINDEX('' '', [Course (hr, crd)])) AS [DepartmentName]
    FROM groupnUploadfile.CoursesSpring2017;';
    EXEC (@SQL);


    DECLARE @SQL2 AS NVARCHAR(MAX);
    SET @SQL2
        = N'CREATE OR ALTER VIEW G10_4.uvw_Department1 AS SELECT DISTINCT COALESCE(NULLIF([DepartmentName],''''), ''TBA'') AS DepartmentName FROM G10_4.uvw_Department';
    EXEC (@SQL2);

    INSERT INTO Department.Department
    (
        DepartmentName,
        UserAuthorizationKey,
        DateAdded,
        DateOfLastUpdate
    )
```



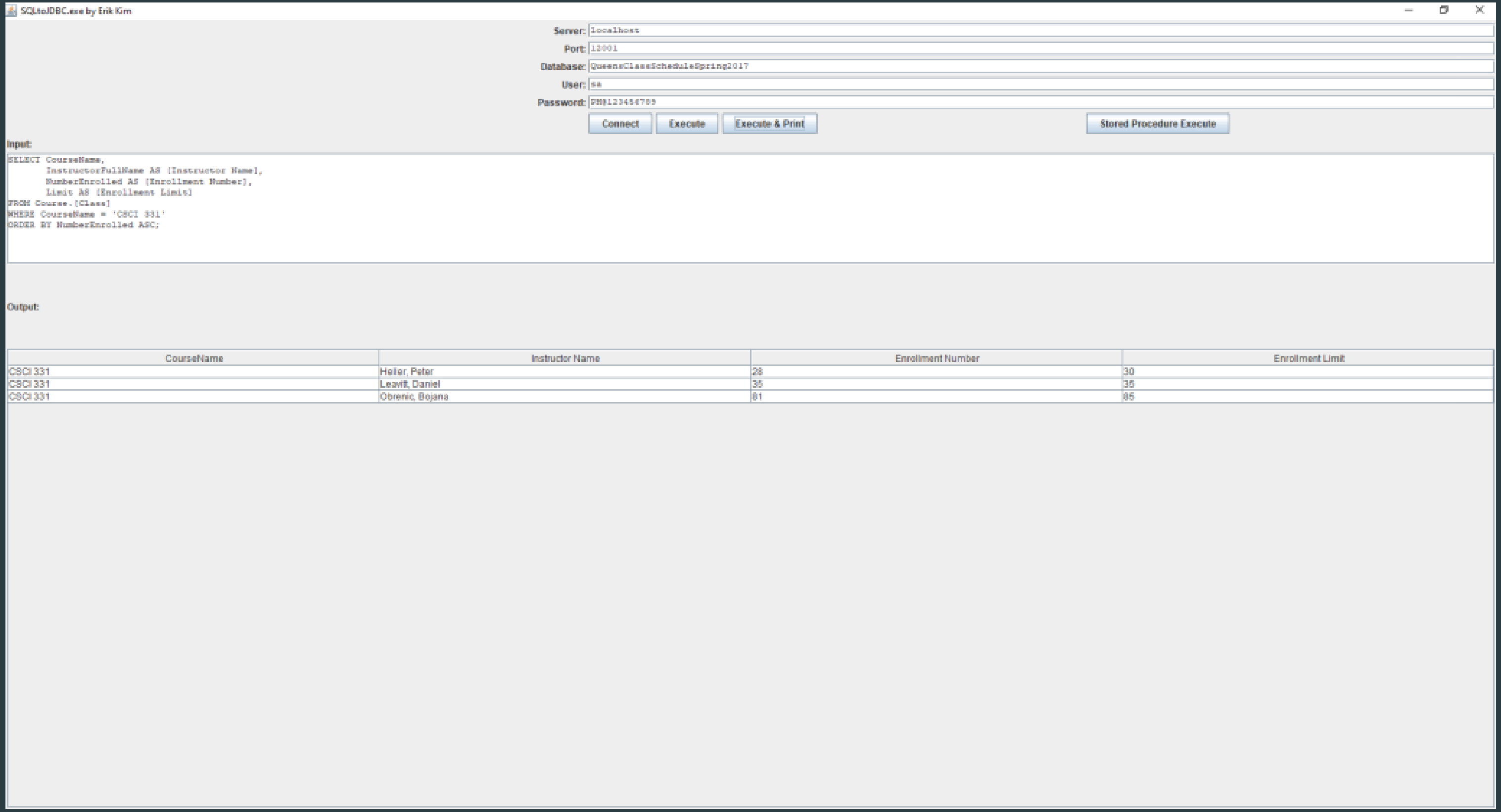
```sql
    SELECT DepartmentName,
           @UserAuthorizationKey,
           @DateAdded,
           @DateOfLastUpdate
    FROM G10_4.uvw_Department1;

    DECLARE @EndingDateTime DATETIME2;
    SET @EndingDateTime = SYSDATETIME();

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount =
    (
        SELECT COUNT(*) FROM [Department].[Department]
    );

    EXEC [Process].[usp_TrackWorkFlow] 'Loads data into [Department].[Department]',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;

    SELECT *
    FROM [Department].[Department];
END;
```

Note: Code is provided in the notes tab on this slide

Harjit’s Queries

- FreeQuery.331Roster: How many CSCI 331 courses are being taught by which professor and what’s their enrollment number and enrollment limit?
- FreeQuery.OnlineClasses: Show all courses that are being taught online?

FreeQuery.CS331Roster



Server: localhost
Port: 12001
Database: QueensClassScheduleSpring2017
User: sa
Password: PH@123456789

Input:

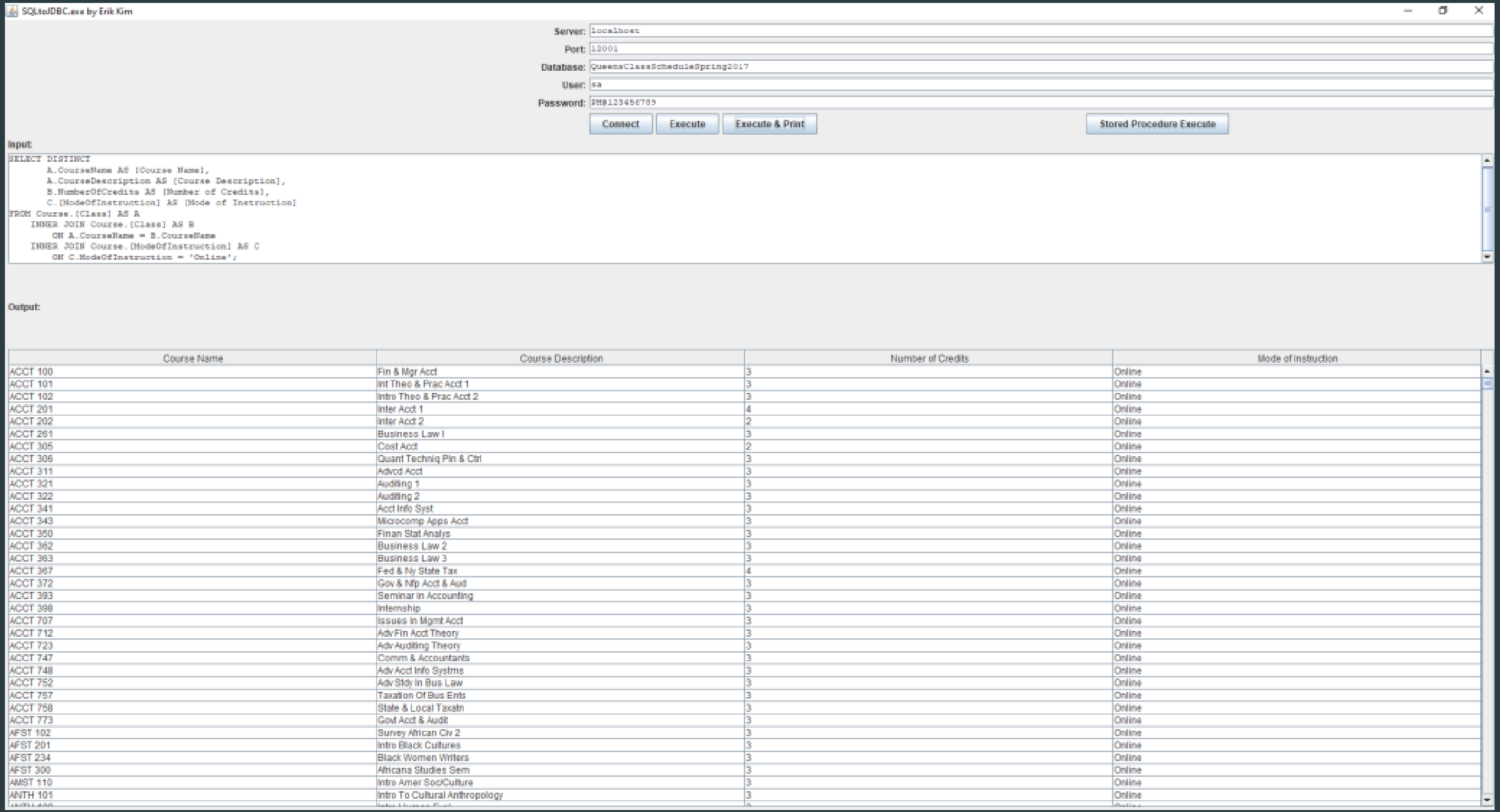
```
SELECT CourseName,
       InstructorFullName AS [Instructor Name],
       NumberEnrolled AS [Enrollment Number],
       Limit AS [Enrollment Limit]
FROM Course.[Class]
WHERE CourseName = 'CSCI 331'
ORDER BY NumberEnrolled ASC;
```

Output:

CourseName	Instructor Name	Enrollment Number	Enrollment Limit
CSCI 331	Heller, Peter	28	30
CSCI 331	Leavitt, Daniel	35	35
CSCI 331	Obrenic, Bojana	81	85

Note: Code is provided in the notes tab on this slide

FreeQuery.OnlineClasses



Server: localhost
Port: 12001
Database: QueensClassScheduleSpring2017
User: sa
Password: PH@123456789

Input:

```
SELECT DISTINCT
        A.CourseName AS [Course Name],
        A.CourseDescription AS [Course Description],
        B.NumberOfCredits AS [Number of Credits],
        C.[ModeOfInstruction] AS [Mode of Instruction]
FROM Course.[Class] AS A
    INNER JOIN Course.[Class] AS B
        ON A.CourseName = B.CourseName
    INNER JOIN Course.[ModeOfInstruction] AS C
        ON C.ModeOfInstruction = 'Online';
```

Output:

Course Name	Course Description	Number of Credits	Mode of Instruction
ACCT 100	Fin & Mgr Acct	3	Online
ACCT 101	Int Theo & Prac Acct 1	3	Online
ACCT 102	Intro Theo & Prac Acct 2	3	Online
ACCT 201	Inter Acct 1	4	Online
ACCT 202	Inter Acct 2	2	Online
ACCT 261	Business Law I	3	Online
ACCT 305	Cost Acct	2	Online
ACCT 306	Quant Techniq Pln & Ctrl	3	Online
ACCT 311	Advcd Acct	3	Online
ACCT 321	Auditing 1	3	Online
ACCT 322	Auditing 2	3	Online
ACCT 341	Acct Info Syst	3	Online
ACCT 343	Microcomp Apps Acct	3	Online
ACCT 350	Finan Stat Analys	3	Online
ACCT 362	Business Law 2	3	Online
ACCT 363	Business Law 3	3	Online
ACCT 367	Fed & Ny State Tax	4	Online
ACCT 372	Gov & Nfp Acct & Aud	3	Online
ACCT 393	Seminar in Accounting	3	Online
ACCT 398	Internship	3	Online
ACCT 707	Issues In Mgmt Acct	3	Online
ACCT 712	Adv Fin Acct Theory	3	Online
ACCT 723	Adv Auditing Theory	3	Online
ACCT 747	Comm & Accountants	3	Online
ACCT 748	Adv Acct Info Systms	3	Online
ACCT 752	Adv Stdy In Bus Law	3	Online
ACCT 757	Taxation Of Bus Ents	3	Online
ACCT 758	State & Local Taxatn	3	Online
ACCT 773	Govt Acct & Audit	3	Online
AFST 102	Survey African Civ 2	3	Online
AFST 201	Intro Black Cultures	3	Online
AFST 234	Black Women Writers	3	Online
AFST 300	Africana Studies Sem	3	Online
AMST 110	Intro Amer Soc/Culture	3	Online
ANTH 101	Intro To Cultural Anthropology	3	Online

Note: Code is provided in the notes tab on this slide

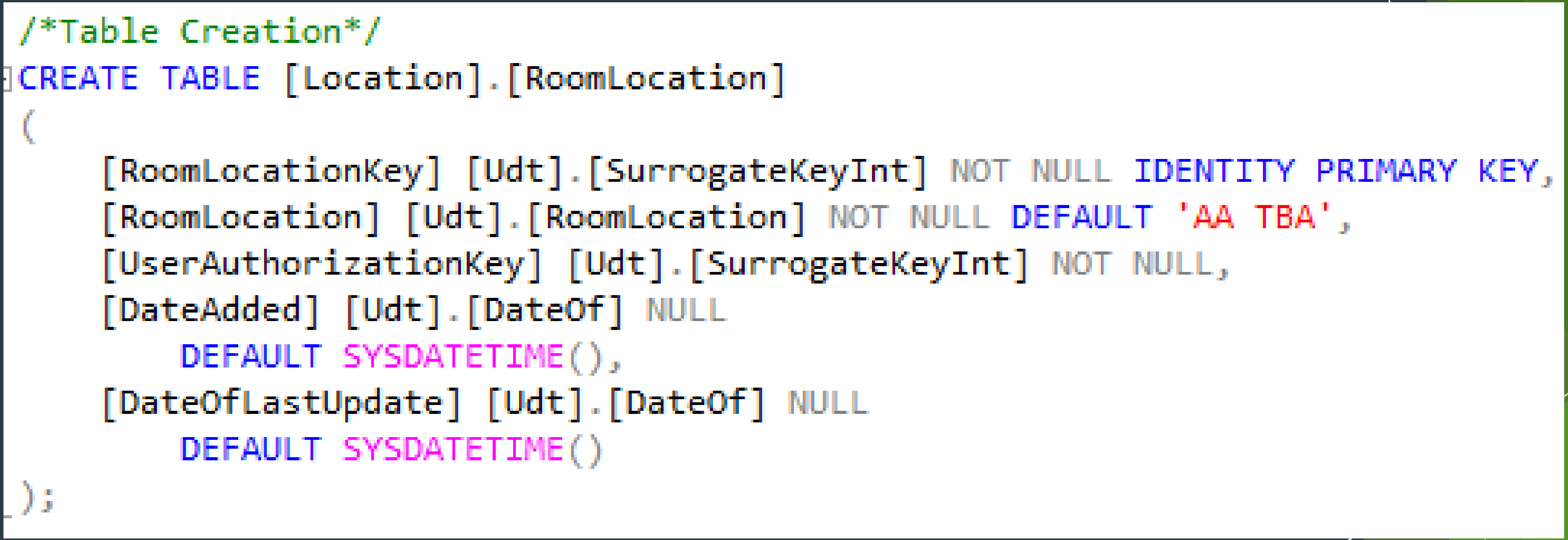
Danny’s Tables & Stored Procedures

- RoomLocation Table
- RoomLocation Stored Procedure

Danny’s Data Anomalies and Fixes

- Some room numbers were blank, which should not be allowed
- Instead of blanks, the room number would be listed as TBA

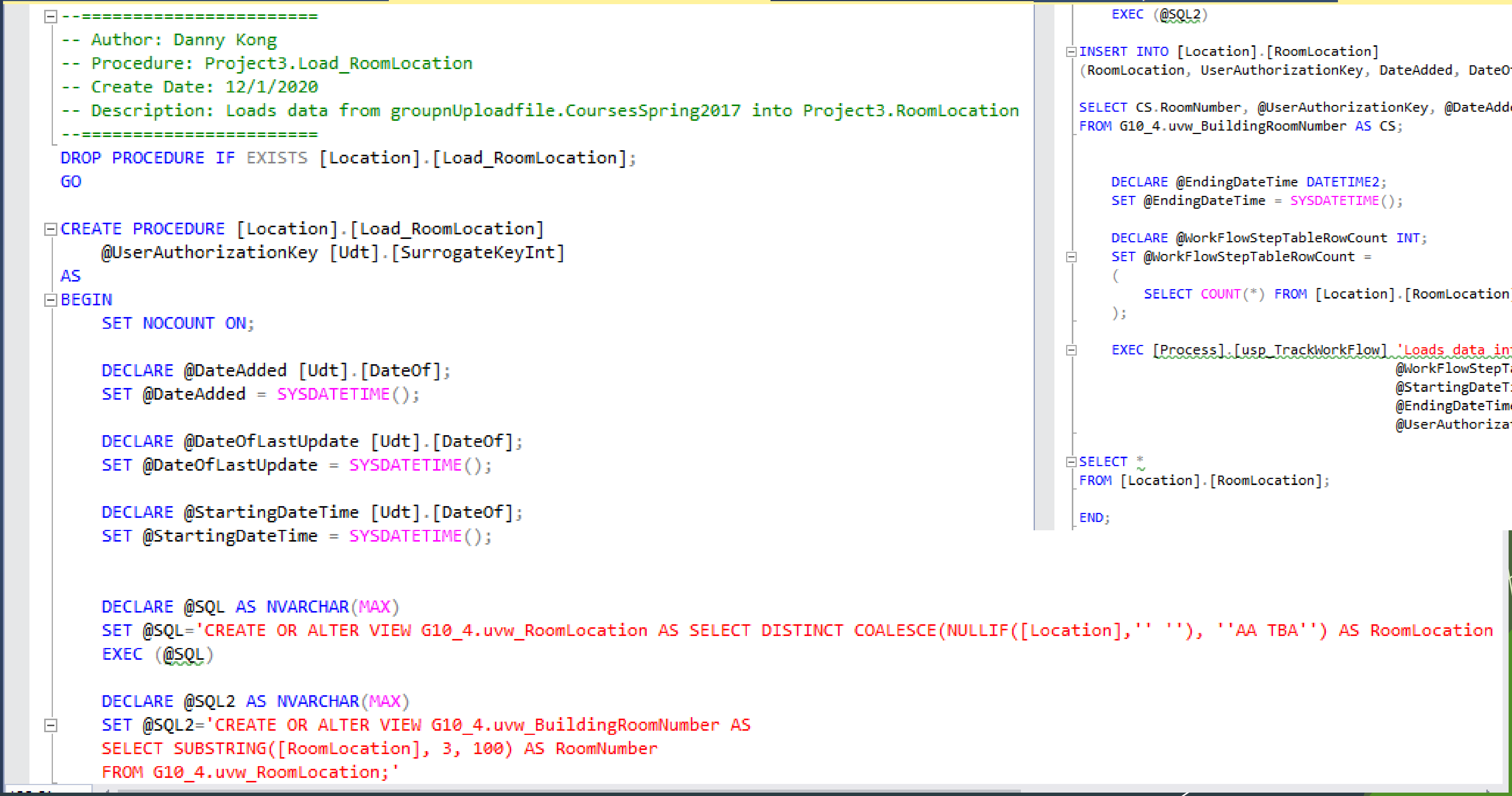
RoomLocation Table Code



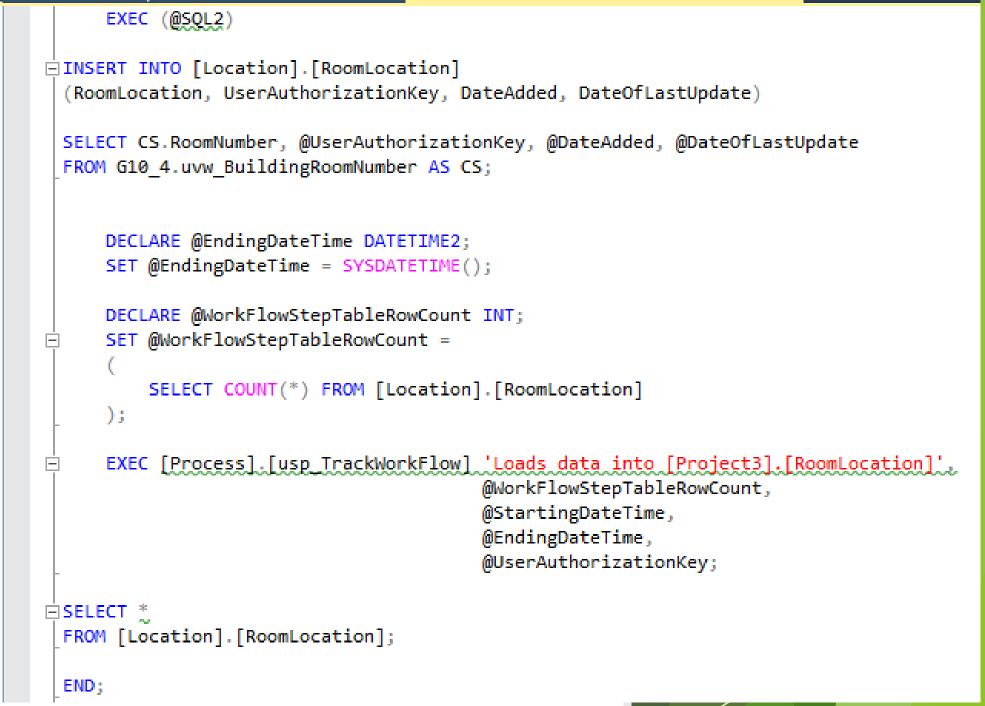
```
/*Table Creation*/
CREATE TABLE [Location].[RoomLocation]
(
    [RoomLocationKey] [Udt].[SurrogateKeyInt] NOT NULL IDENTITY PRIMARY KEY,
    [RoomLocation] [Udt].[RoomLocation] NOT NULL DEFAULT 'AA TBA',
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] NOT NULL,
    [DateAdded] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME(),
    [DateOfLastUpdate] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME()
);
```

Note: Code is provided in the notes tab on this slide

RoomLocation Stored Procedure Code



```sql
--======================
-- Author: Danny Kong
-- Procedure: Project3.Load_RoomLocation
-- Create Date: 12/1/2020
-- Description: Loads data from groupnUploadfile.CoursesSpring2017 into Project3.RoomLocation
--======================
DROP PROCEDURE IF EXISTS [Location].[Load_RoomLocation];
GO

CREATE PROCEDURE [Location].[Load_RoomLocation]
    @UserAuthorizationKey [Udt].[SurrogateKeyInt]
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @DateAdded [Udt].[DateOf];
    SET @DateAdded = SYSDATETIME();

    DECLARE @DateOfLastUpdate [Udt].[DateOf];
    SET @DateOfLastUpdate = SYSDATETIME();

    DECLARE @StartingDateTime [Udt].[DateOf];
    SET @StartingDateTime = SYSDATETIME();


    DECLARE @SQL AS NVARCHAR(MAX)
    SET @SQL='CREATE OR ALTER VIEW G10_4.uvw_RoomLocation AS SELECT DISTINCT COALESCE(NULLIF([Location],'' ''), ''AA TBA'') AS RoomLocation
    EXEC (@SQL)

    DECLARE @SQL2 AS NVARCHAR(MAX)
    SET @SQL2='CREATE OR ALTER VIEW G10_4.uvw_BuildingRoomNumber AS
    SELECT SUBSTRING([RoomLocation], 3, 100) AS RoomNumber
    FROM G10_4.uvw_RoomLocation;'
```



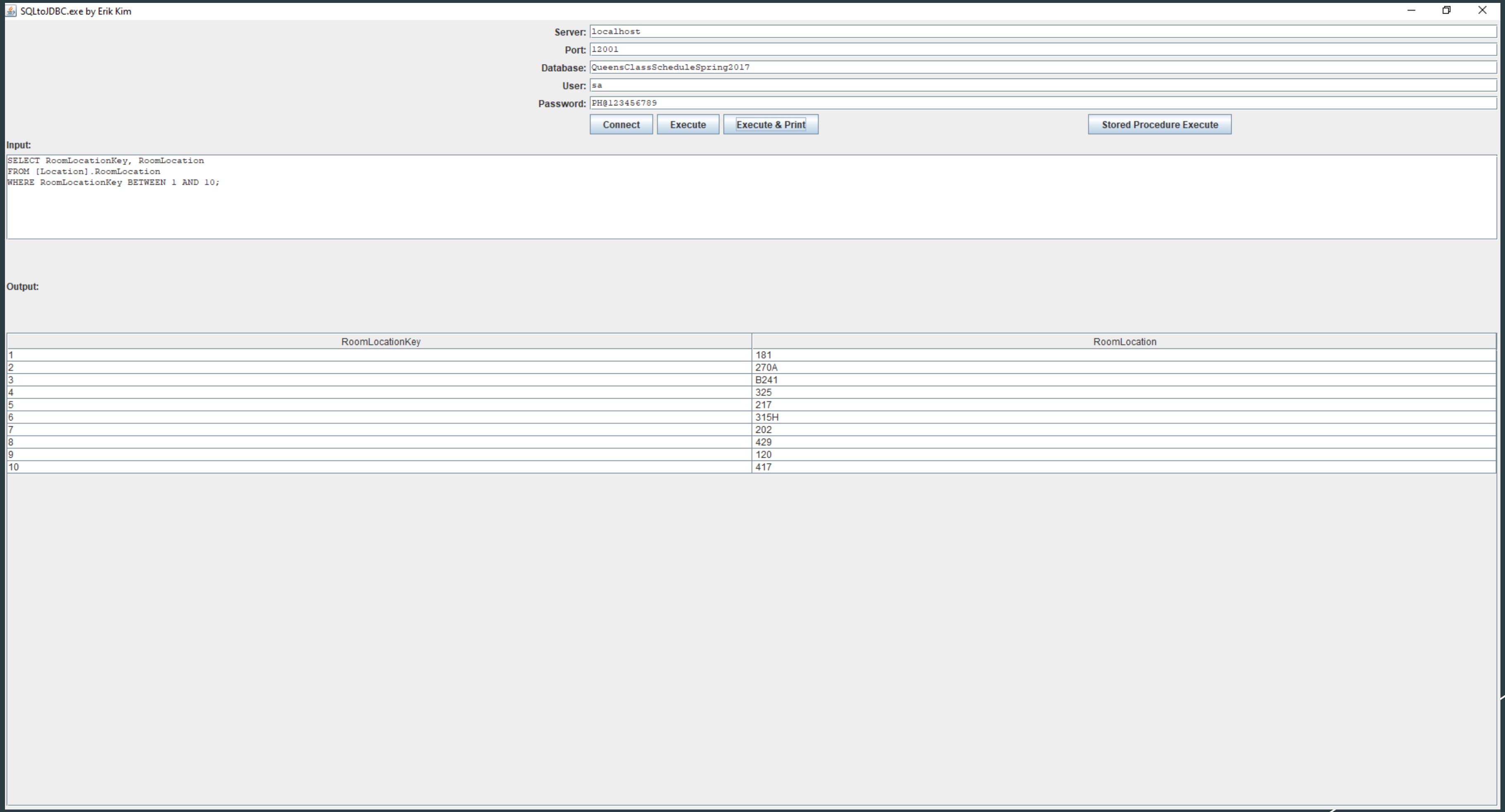
```sql
    EXEC (@SQL2)

INSERT INTO [Location].[RoomLocation]
(RoomLocation, UserAuthorizationKey, DateAdded, DateOfLastUpdate)

SELECT CS.RoomNumber, @UserAuthorizationKey, @DateAdded, @DateOfLastUpdate
FROM G10_4.uvw_BuildingRoomNumber AS CS;


    DECLARE @EndingDateTime DATETIME2;
    SET @EndingDateTime = SYSDATETIME();

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount =
    (
        SELECT COUNT(*) FROM [Location].[RoomLocation]
    );

    EXEC [Process].[usp_TrackWorkFlow] 'Loads data into [Project3].[RoomLocation]',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;

SELECT *
FROM [Location].[RoomLocation];

END;
```

Note: Code is provided in the notes tab on this slide

Danny’s Queries

- Free Query 1: Display all rooms that have a RoomLocationKey between 1 and 10
- Free Query 2: Display all rooms that have a letter at the end of the room number

FreeQuery.RoomKey1-10



Server: localhost
Port: 12001
Database: QueensClassScheduleSpring2017
User: sa
Password: PH@123456789

Input:

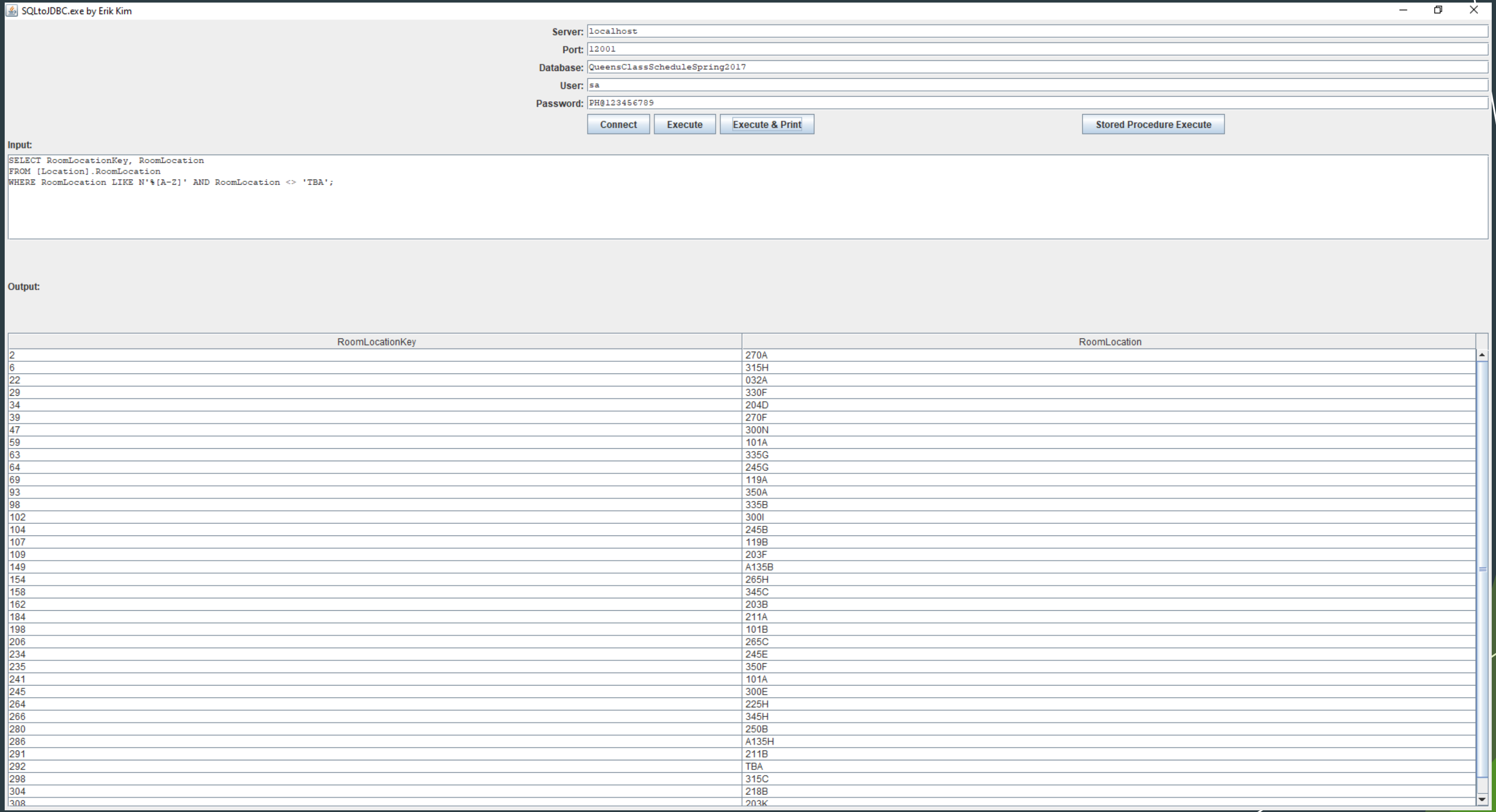
```
SELECT RoomLocationKey, RoomLocation
FROM [Location].RoomLocation
WHERE RoomLocationKey BETWEEN 1 AND 10;
```

Output:

RoomLocationKey	RoomLocation
1	181
2	270A
3	B241
4	325
5	217
6	315H
7	202
8	429
9	120
10	417

Note: Code is provided in the notes tab on this slide

FreeQuery.RoomEndwLetter



Server: localhost
Port: 12001
Database: QueensClassScheduleSpring2017
User: sa
Password: PH@123456789

Input:

```
SELECT RoomLocationKey, RoomLocation
FROM [Location].RoomLocation
WHERE RoomLocation LIKE N'%[A-Z]' AND RoomLocation <> 'TBA';
```

Output:

RoomLocationKey	RoomLocation
2	270A
6	315H
22	032A
29	330F
34	204D
39	270F
47	300N
59	101A
63	335G
64	245G
69	119A
93	350A
98	335B
102	300I
104	245B
107	119B
109	203F
149	A135B
154	265H
158	345C
162	203B
184	211A
198	101B
206	265C
234	245E
235	350F
241	101A
245	300E
264	225H
266	345H
280	250B
286	A135H
291	211B
292	TBA
298	315C
304	218B
308	203K

Note: Code is provided in the notes tab on this slide

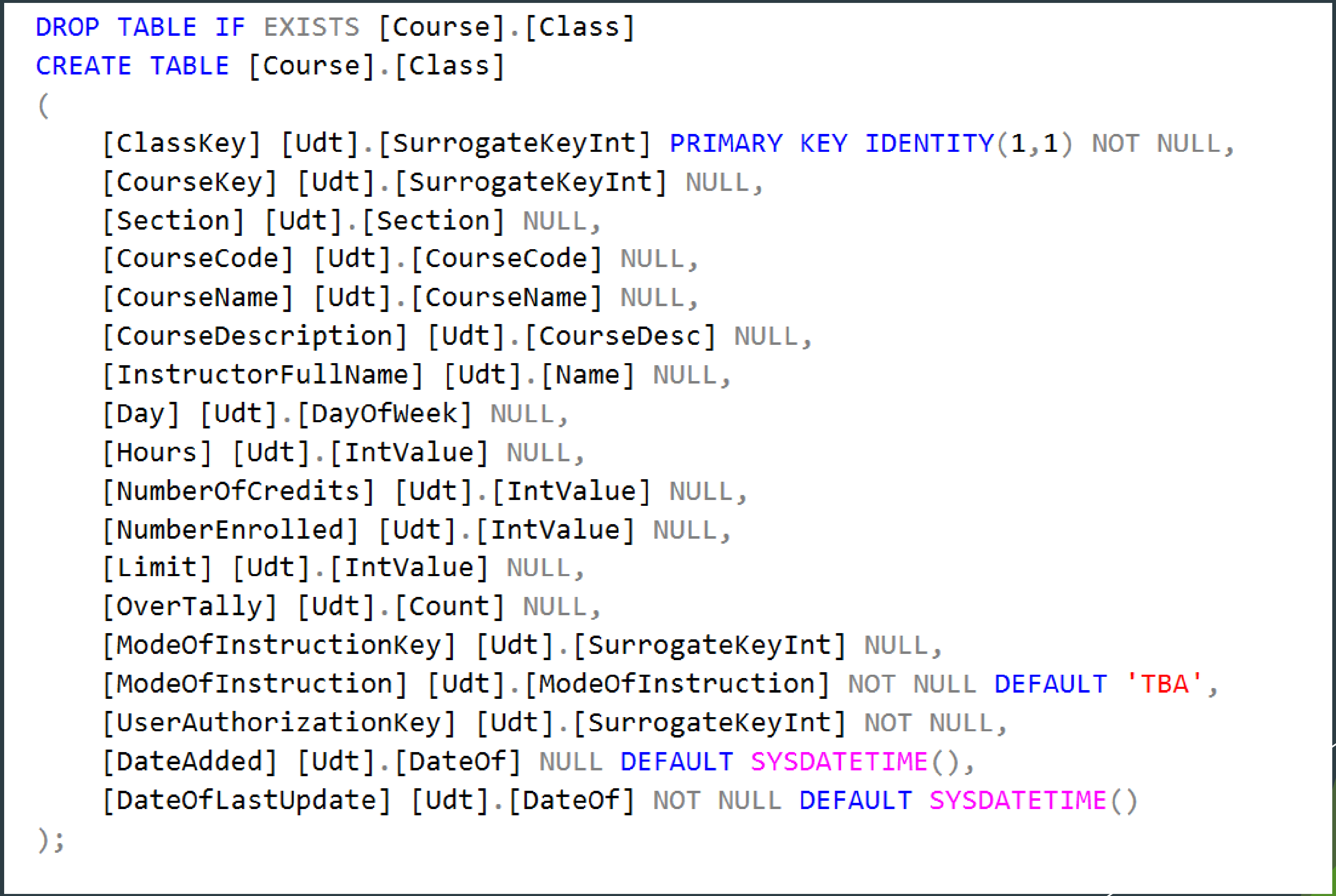
Marlon’s Tables & Stored Procedures

- Class Table
- Class Stored Procedure

Marlon’s Data Anomalies and Fixes

- ANOMALY: Days were left blank for some classes
- ANOMALY: Class times were not specified for all courses
- ANOMALY: Instructors were not specified for some classes
- ANOMALY: Location was left blank for some classes
- SOLUTION: To resolve these issues, TBA was put in place of the blanks

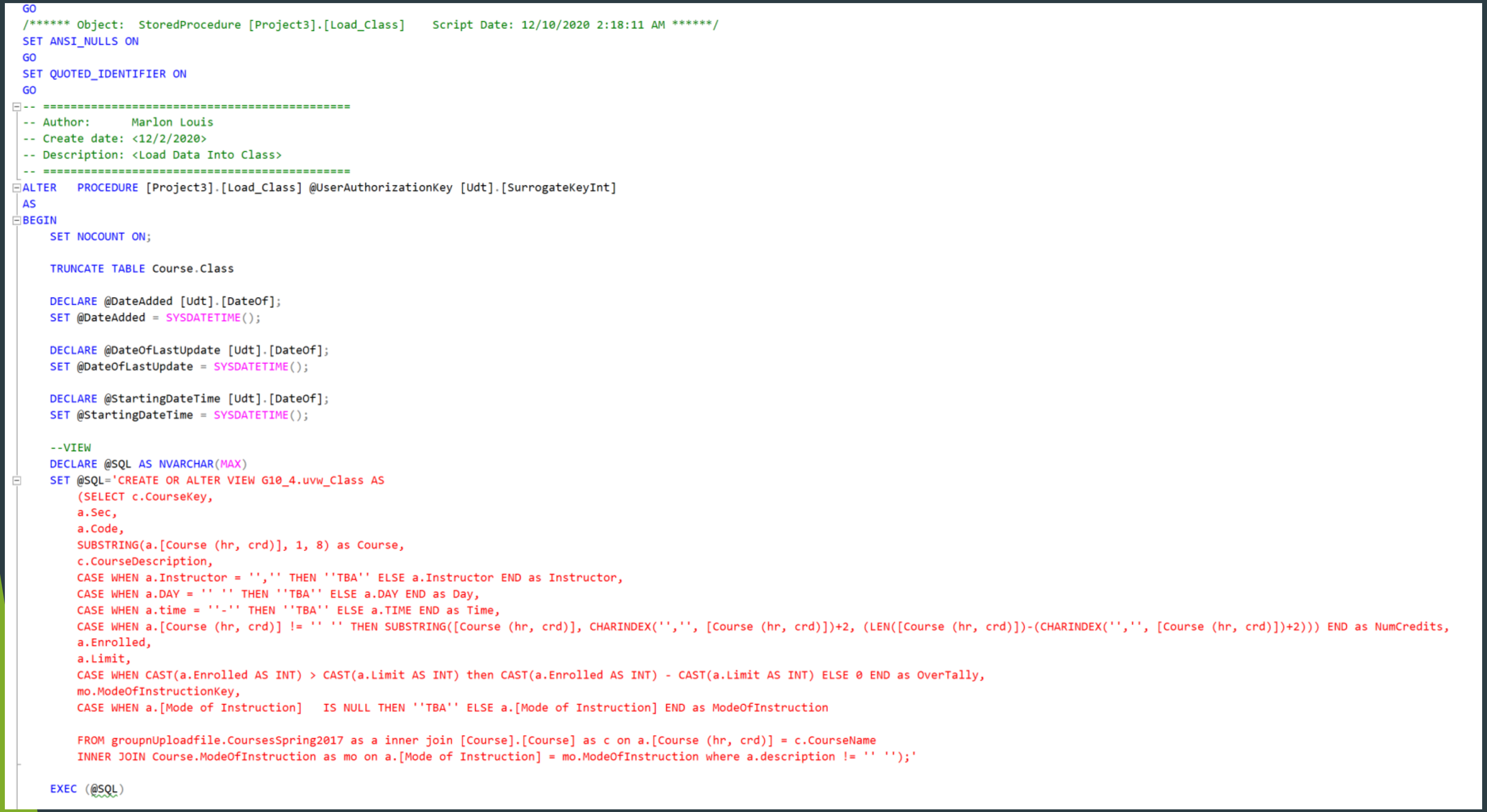
Class Table Code



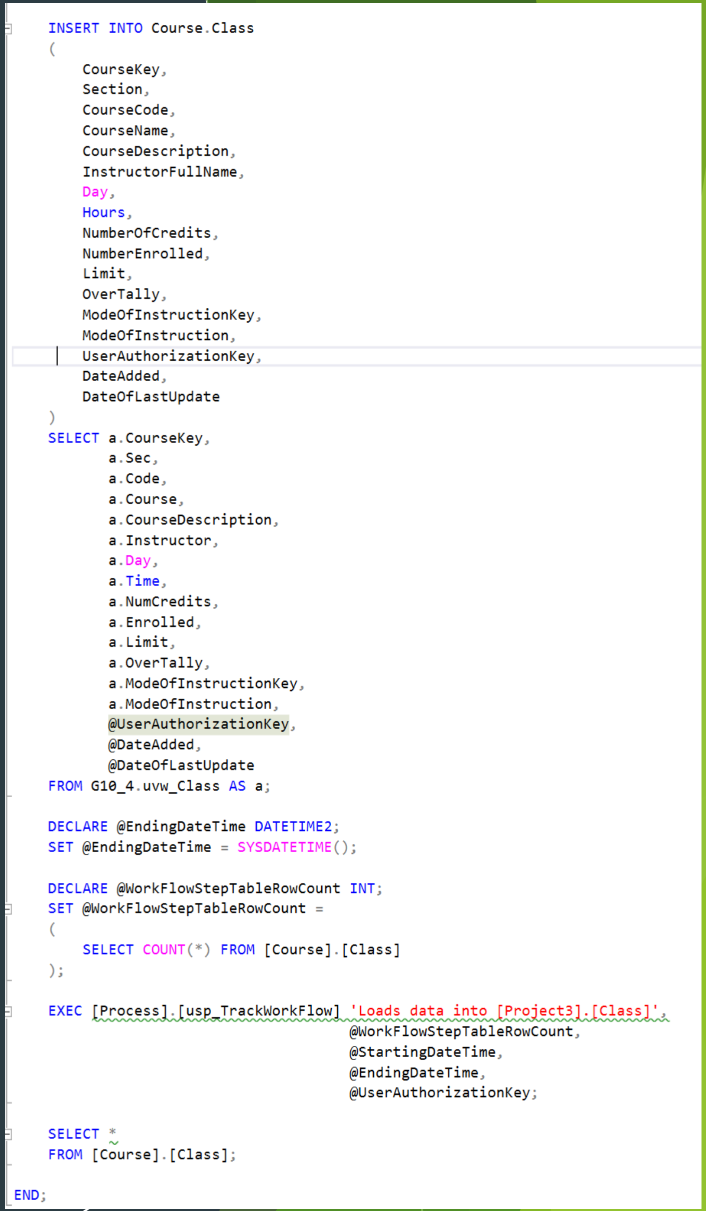
```
DROP TABLE IF EXISTS [Course].[Class]
CREATE TABLE [Course].[Class]
(
    [ClassKey] [Udt].[SurrogateKeyInt] PRIMARY KEY IDENTITY(1,1) NOT NULL,
    [CourseKey] [Udt].[SurrogateKeyInt] NULL,
    [Section] [Udt].[Section] NULL,
    [CourseCode] [Udt].[CourseCode] NULL,
    [CourseName] [Udt].[CourseName] NULL,
    [CourseDescription] [Udt].[CourseDesc] NULL,
    [InstructorFullName] [Udt].[Name] NULL,
    [Day] [Udt].[DayOfWeek] NULL,
    [Hours] [Udt].[IntValue] NULL,
    [NumberOfCredits] [Udt].[IntValue] NULL,
    [NumberEnrolled] [Udt].[IntValue] NULL,
    [Limit] [Udt].[IntValue] NULL,
    [OverTally] [Udt].[Count] NULL,
    [ModeOfInstructionKey] [Udt].[SurrogateKeyInt] NULL,
    [ModeOfInstruction] [Udt].[ModeOfInstruction] NOT NULL DEFAULT 'TBA',
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] NOT NULL,
    [DateAdded] [Udt].[DateOf] NULL DEFAULT SYSDATETIME(),
    [DateOfLastUpdate] [Udt].[DateOf] NOT NULL DEFAULT SYSDATETIME()
);
```

Note: Code is provided in the notes tab on this slide

Class Stored Procedure Code



```sql
GO
/****** Object:  StoredProcedure [Project3].[Load_Class]    Script Date: 12/10/2020 2:18:11 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Marlon Louis
-- Create date: <12/2/2020>
-- Description: <Load Data Into Class>
-- =============================================
ALTER   PROCEDURE [Project3].[Load_Class] @UserAuthorizationKey [Udt].[SurrogateKeyInt]
AS
BEGIN
    SET NOCOUNT ON;

    TRUNCATE TABLE Course.Class

    DECLARE @DateAdded [Udt].[DateOf];
    SET @DateAdded = SYSDATETIME();

    DECLARE @DateOfLastUpdate [Udt].[DateOf];
    SET @DateOfLastUpdate = SYSDATETIME();

    DECLARE @StartingDateTime [Udt].[DateOf];
    SET @StartingDateTime = SYSDATETIME();

    --VIEW
    DECLARE @SQL AS NVARCHAR(MAX)
    SET @SQL='CREATE OR ALTER VIEW G10_4.uvw_Class AS
        (SELECT c.CourseKey,
        a.Sec,
        a.Code,
        SUBSTRING(a.[Course (hr, crd)], 1, 8) as Course,
        c.CourseDescription,
        CASE WHEN a.Instructor = '','' THEN ''TBA'' ELSE a.Instructor END as Instructor,
        CASE WHEN a.DAY = '' '' THEN ''TBA'' ELSE a.DAY END as Day,
        CASE WHEN a.time = ''-'' THEN ''TBA'' ELSE a.TIME END as Time,
        CASE WHEN a.[Course (hr, crd)] != '' '' THEN SUBSTRING([Course (hr, crd)], CHARINDEX('','', [Course (hr, crd)])+2, (LEN([Course (hr, crd)])-(CHARINDEX('','', [Course (hr, crd)])+2))) END as NumCredits,
        a.Enrolled,
        a.Limit,
        CASE WHEN CAST(a.Enrolled AS INT) > CAST(a.Limit AS INT) then CAST(a.Enrolled AS INT) - CAST(a.Limit AS INT) ELSE 0 END as OverTally,
        mo.ModeOfInstructionKey,
        CASE WHEN a.[Mode of Instruction]   IS NULL THEN ''TBA'' ELSE a.[Mode of Instruction] END as ModeOfInstruction

        FROM groupnUploadfile.CoursesSpring2017 as a inner join [Course].[Course] as c on a.[Course (hr, crd)] = c.CourseName
        INNER JOIN Course.ModeOfInstruction as mo on a.[Mode of Instruction] = mo.ModeOfInstruction where a.description != '' '');'

    EXEC (@SQL)
```



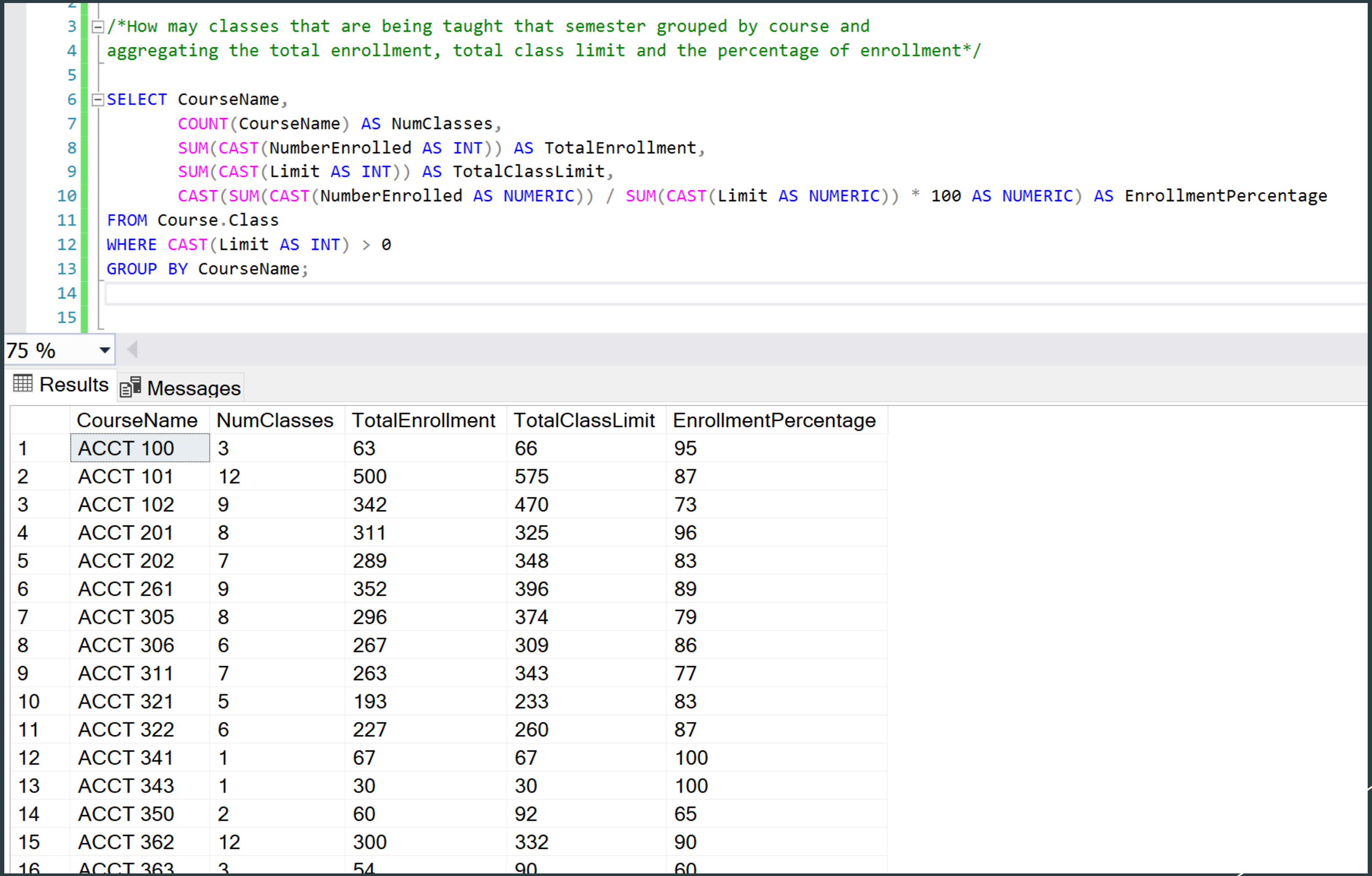
```sql
    INSERT INTO Course.Class
    (
        CourseKey,
        Section,
        CourseCode,
        CourseName,
        CourseDescription,
        InstructorFullName,
        Day,
        Hours,
        NumberOfCredits,
        NumberEnrolled,
        Limit,
        OverTally,
        ModeOfInstructionKey,
        ModeOfInstruction,
        UserAuthorizationKey,
        DateAdded,
        DateOfLastUpdate
    )
    SELECT a.CourseKey,
           a.Sec,
           a.Code,
           a.Course,
           a.CourseDescription,
           a.Instructor,
           a.Day,
           a.Time,
           a.NumCredits,
           a.Enrolled,
           a.Limit,
           a.OverTally,
           a.ModeOfInstructionKey,
           a.ModeOfInstruction,
           @UserAuthorizationKey,
           @DateAdded,
           @DateOfLastUpdate
    FROM G10_4.uvw_Class AS a;

    DECLARE @EndingDateTime DATETIME2;
    SET @EndingDateTime = SYSDATETIME();

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount =
    (
        SELECT COUNT(*) FROM [Course].[Class]
    );

    EXEC [Process].[usp_TrackWorkFlow] 'Loads data into [Project3].[Class]',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;

    SELECT *
    FROM [Course].[Class];

END;
```

Note: Code is provided in the notes tab on this slide

Marlon’s Queries

- Query 3: How may classes that are being taught that semester grouped by course and aggregating the total enrollment, total class limit and the percentage of enrollment
- Free query: Find all courses that have less than half enrollment

Project3.Query3

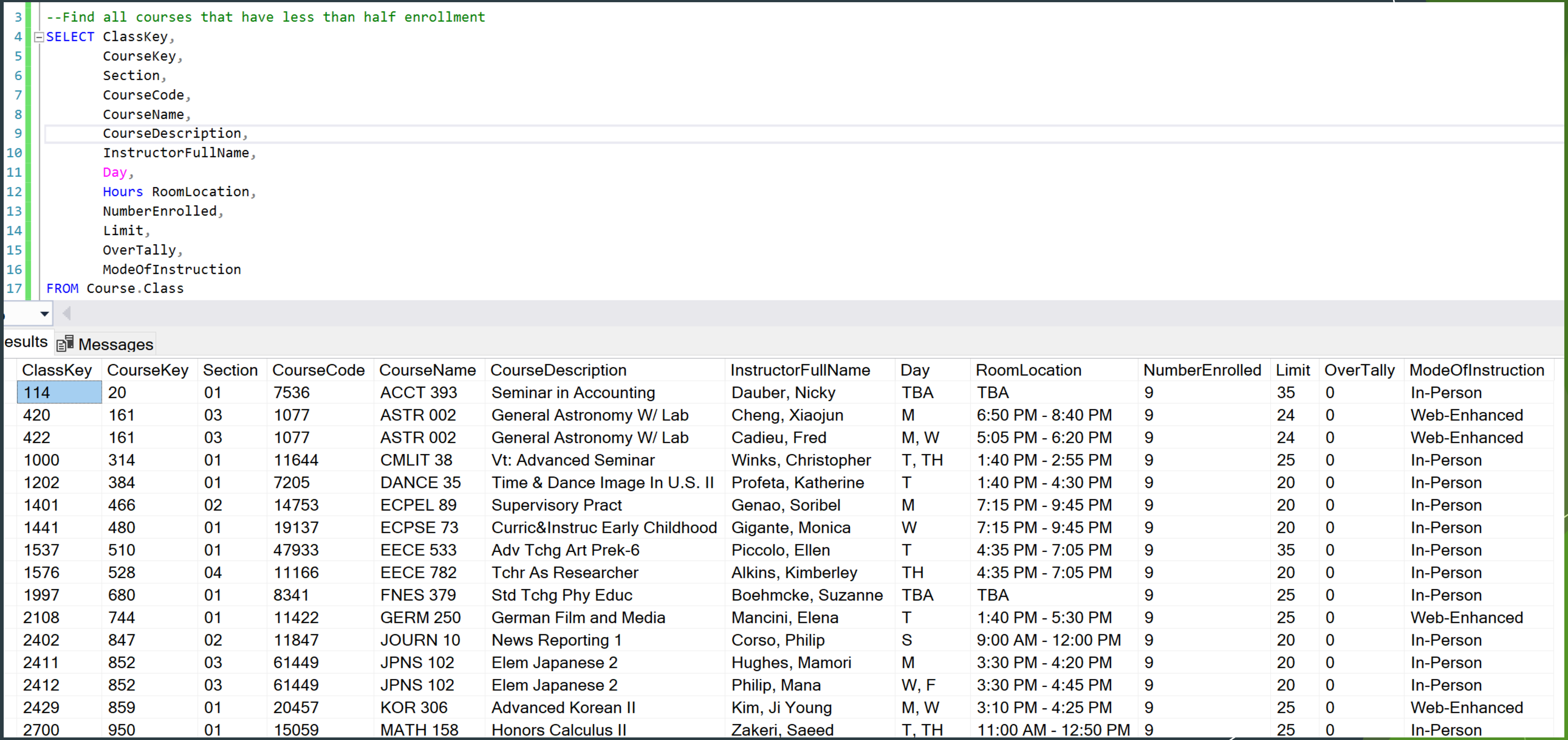


```
/*How may classes that are being taught that semester grouped by course and
aggregating the total enrollment, total class limit and the percentage of enrollment*/

SELECT CourseName,
        COUNT(CourseName) AS NumClasses,
        SUM(CAST(NumberEnrolled AS INT)) AS TotalEnrollment,
        SUM(CAST(Limit AS INT)) AS TotalClassLimit,
        CAST(SUM(CAST(NumberEnrolled AS NUMERIC)) / SUM(CAST(Limit AS NUMERIC)) * 100 AS NUMERIC) AS EnrollmentPercentage
FROM Course.Class
WHERE CAST(Limit AS INT) > 0
GROUP BY CourseName;
```

	CourseName	NumClasses	TotalEnrollment	TotalClassLimit	EnrollmentPercentage
1	ACCT 100	3	63	66	95
2	ACCT 101	12	500	575	87
3	ACCT 102	9	342	470	73
4	ACCT 201	8	311	325	96
5	ACCT 202	7	289	348	83
6	ACCT 261	9	352	396	89
7	ACCT 305	8	296	374	79
8	ACCT 306	6	267	309	86
9	ACCT 311	7	263	343	77
10	ACCT 321	5	193	233	83
11	ACCT 322	6	227	260	87
12	ACCT 341	1	67	67	100
13	ACCT 343	1	30	30	100
14	ACCT 350	2	60	92	65
15	ACCT 362	12	300	332	90
16	ACCT 363	3	54	90	60

Note: Code is provided in the notes tab on this slide

FreeQuery.LowEnrollment



```
--Find all courses that have less than half enrollment
SELECT ClassKey,
       CourseKey,
       Section,
       CourseCode,
       CourseName,
       CourseDescription,
       InstructorFullName,
       Day,
       Hours RoomLocation,
       NumberEnrolled,
       Limit,
       OverTally,
       ModeOfInstruction
FROM Course.Class
```

ClassKey	CourseKey	Section	CourseCode	CourseName	CourseDescription	InstructorFullName	Day	RoomLocation	NumberEnrolled	Limit	OverTally	ModeOfInstruction
114	20	01	7536	ACCT 393	Seminar in Accounting	Dauber, Nicky	TBA	TBA	9	35	0	In-Person
420	161	03	1077	ASTR 002	General Astronomy W/ Lab	Cheng, Xiaojun	M	6:50 PM - 8:40 PM	9	24	0	Web-Enhanced
422	161	03	1077	ASTR 002	General Astronomy W/ Lab	Cadieu, Fred	M, W	5:05 PM - 6:20 PM	9	24	0	Web-Enhanced
1000	314	01	11644	CMLIT 38	Vt: Advanced Seminar	Winks, Christopher	T, TH	1:40 PM - 2:55 PM	9	25	0	In-Person
1202	384	01	7205	DANCE 35	Time & Dance Image In U.S. II	Profeta, Katherine	T	1:40 PM - 4:30 PM	9	20	0	In-Person
1401	466	02	14753	ECPEL 89	Supervisory Pract	Genao, Soribel	M	7:15 PM - 9:45 PM	9	20	0	In-Person
1441	480	01	19137	ECPSE 73	Curric&Instruc Early Childhood	Gigante, Monica	W	7:15 PM - 9:45 PM	9	20	0	In-Person
1537	510	01	47933	EECE 533	Adv Tchg Art Prek-6	Piccolo, Ellen	T	4:35 PM - 7:05 PM	9	35	0	In-Person
1576	528	04	11166	EECE 782	Tchr As Researcher	Alkins, Kimberley	TH	4:35 PM - 7:05 PM	9	20	0	In-Person
1997	680	01	8341	FNES 379	Std Tchg Phy Educ	Boehmcke, Suzanne	TBA	TBA	9	25	0	In-Person
2108	744	01	11422	GERM 250	German Film and Media	Mancini, Elena	T	1:40 PM - 5:30 PM	9	25	0	Web-Enhanced
2402	847	02	11847	JOURN 10	News Reporting 1	Corso, Philip	S	9:00 AM - 12:00 PM	9	20	0	In-Person
2411	852	03	61449	JPNS 102	Elem Japanese 2	Hughes, Mamori	M	3:30 PM - 4:20 PM	9	20	0	In-Person
2412	852	03	61449	JPNS 102	Elem Japanese 2	Philip, Mana	W, F	3:30 PM - 4:45 PM	9	20	0	In-Person
2429	859	01	20457	KOR 306	Advanced Korean II	Kim, Ji Young	M, W	3:10 PM - 4:25 PM	9	25	0	Web-Enhanced
2700	950	01	15059	MATH 158	Honors Calculus II	Zakeri, Saeed	T, TH	11:00 AM - 12:50 PM	9	25	0	In-Person

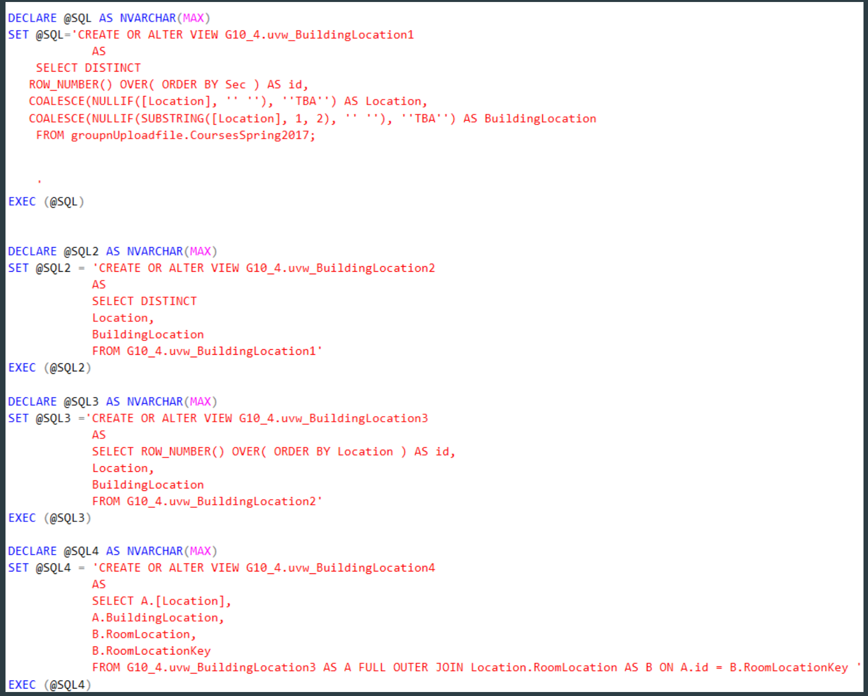
Note: Code is provided in the notes tab on this slide

Haibo’s Tables & Stored Procedures

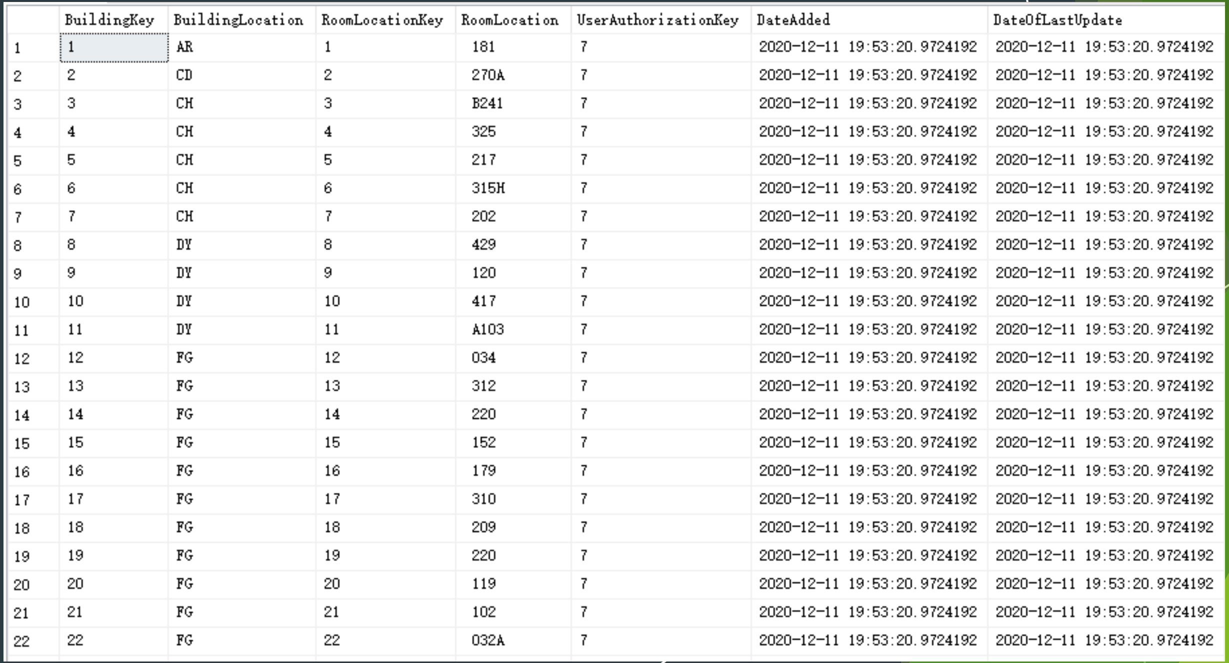
- BuildingLocationTable
- BuildingLocation Stored Procedure

Haibo’s Data Anomalies and Fixes

- The Store procedure show nothing
- My table relate to Danny's table, when he was changed his table, I have to fix my view for reference it, I added 3 more views to fix it.

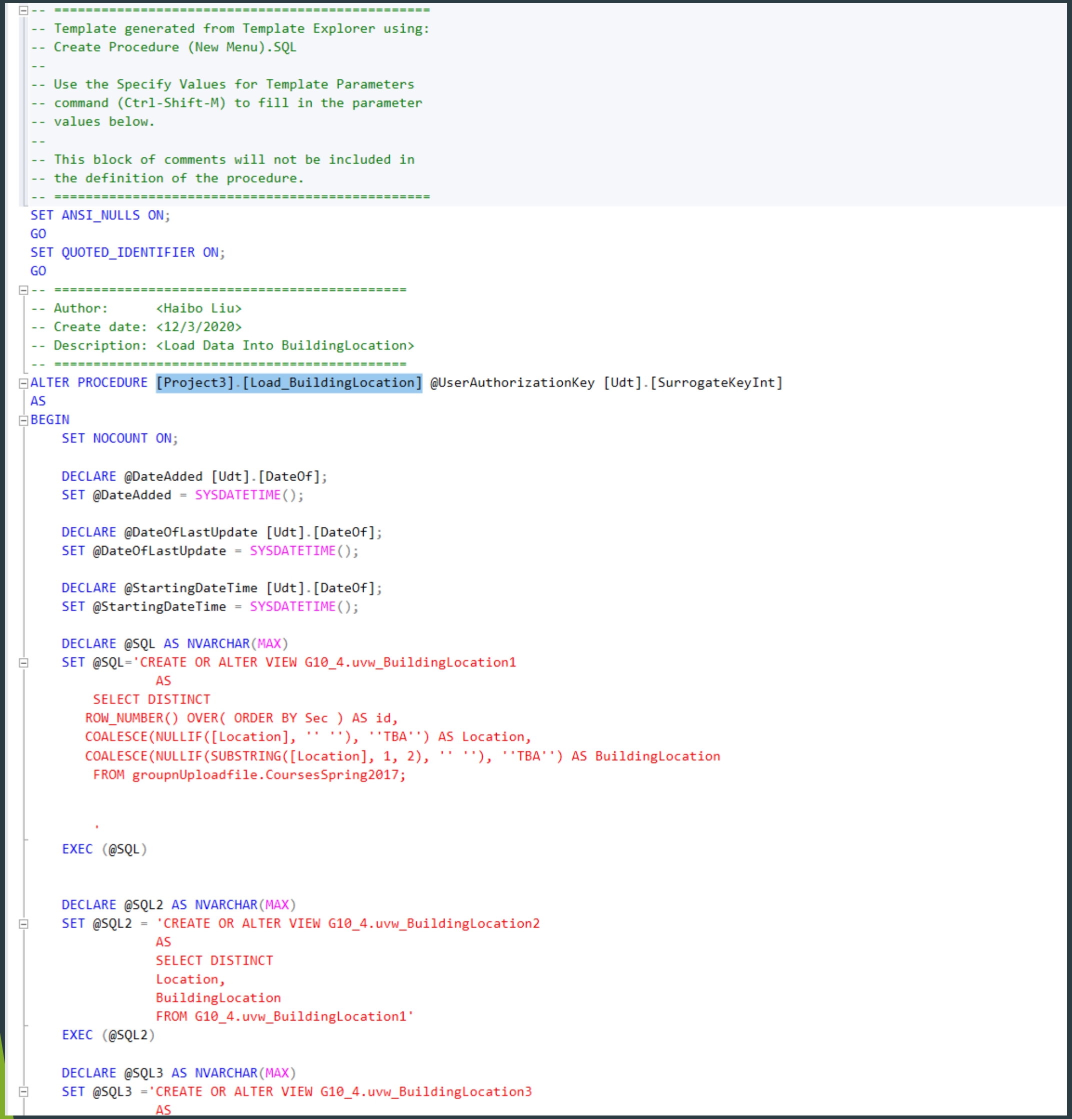


```sql
DECLARE @SQL AS NVARCHAR(MAX)
SET @SQL='CREATE OR ALTER VIEW G10_4.uvw_BuildingLocation1
            AS
    SELECT DISTINCT
   ROW_NUMBER() OVER( ORDER BY Sec ) AS id,
   COALESCE(NULLIF([Location], '' ''), ''TBA'') AS Location,
   COALESCE(NULLIF(SUBSTRING([Location], 1, 2), '' ''), ''TBA'') AS BuildingLocation
    FROM groupnUploadfile.CoursesSpring2017;


    '
EXEC (@SQL)


DECLARE @SQL2 AS NVARCHAR(MAX)
SET @SQL2 = 'CREATE OR ALTER VIEW G10_4.uvw_BuildingLocation2
            AS
            SELECT DISTINCT
            Location,
            BuildingLocation
            FROM G10_4.uvw_BuildingLocation1'
EXEC (@SQL2)

DECLARE @SQL3 AS NVARCHAR(MAX)
SET @SQL3 ='CREATE OR ALTER VIEW G10_4.uvw_BuildingLocation3
            AS
            SELECT ROW_NUMBER() OVER( ORDER BY Location ) AS id,
            Location,
            BuildingLocation
            FROM G10_4.uvw_BuildingLocation2'
EXEC (@SQL3)

DECLARE @SQL4 AS NVARCHAR(MAX)
SET @SQL4 = 'CREATE OR ALTER VIEW G10_4.uvw_BuildingLocation4
            AS
            SELECT A.[Location],
            A.BuildingLocation,
            B.RoomLocation,
            B.RoomLocationKey
            FROM G10_4.uvw_BuildingLocation3 AS A FULL OUTER JOIN Location.RoomLocation AS B ON A.id = B.RoomLocationKey '
EXEC (@SQL4)
```

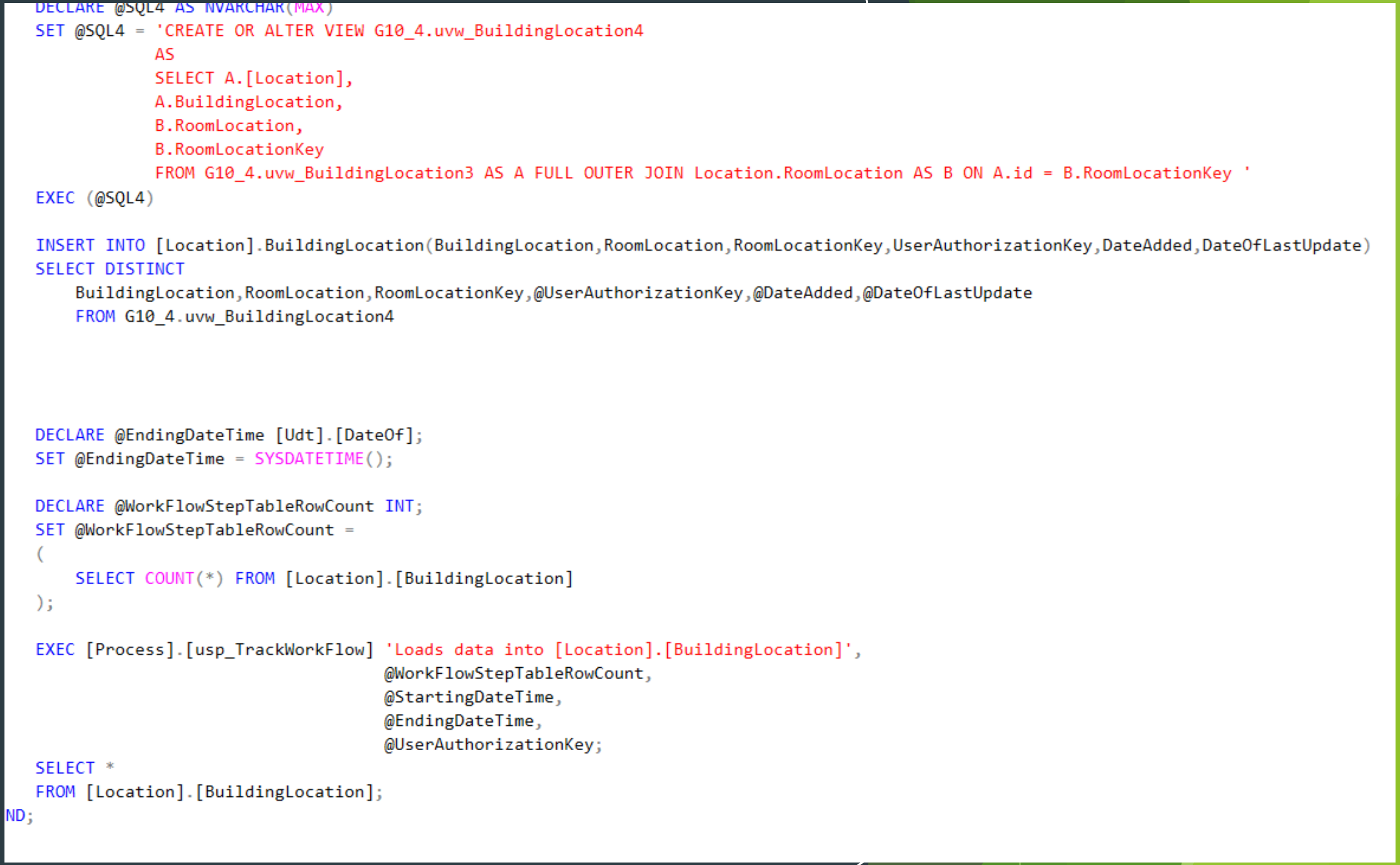


	BuildingKey	BuildingLocation	RoomLocationKey	RoomLocation	UserAuthorizationKey	DateAdded	DateOfLastUpdate
1	1	AR	1	181	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
2	2	CD	2	270A	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
3	3	CH	3	B241	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
4	4	CH	4	325	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
5	5	CH	5	217	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
6	6	CH	6	315H	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
7	7	CH	7	202	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
8	8	DY	8	429	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
9	9	DY	9	120	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
10	10	DY	10	417	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
11	11	DY	11	A103	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
12	12	FG	12	034	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
13	13	FG	13	312	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
14	14	FG	14	220	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
15	15	FG	15	152	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
16	16	FG	16	179	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
17	17	FG	17	310	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
18	18	FG	18	209	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
19	19	FG	19	220	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
20	20	FG	20	119	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
21	21	FG	21	102	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192
22	22	FG	22	032A	7	2020-12-11 19:53:20.9724192	2020-12-11 19:53:20.9724192

BuildingLocation Store Procedure Code



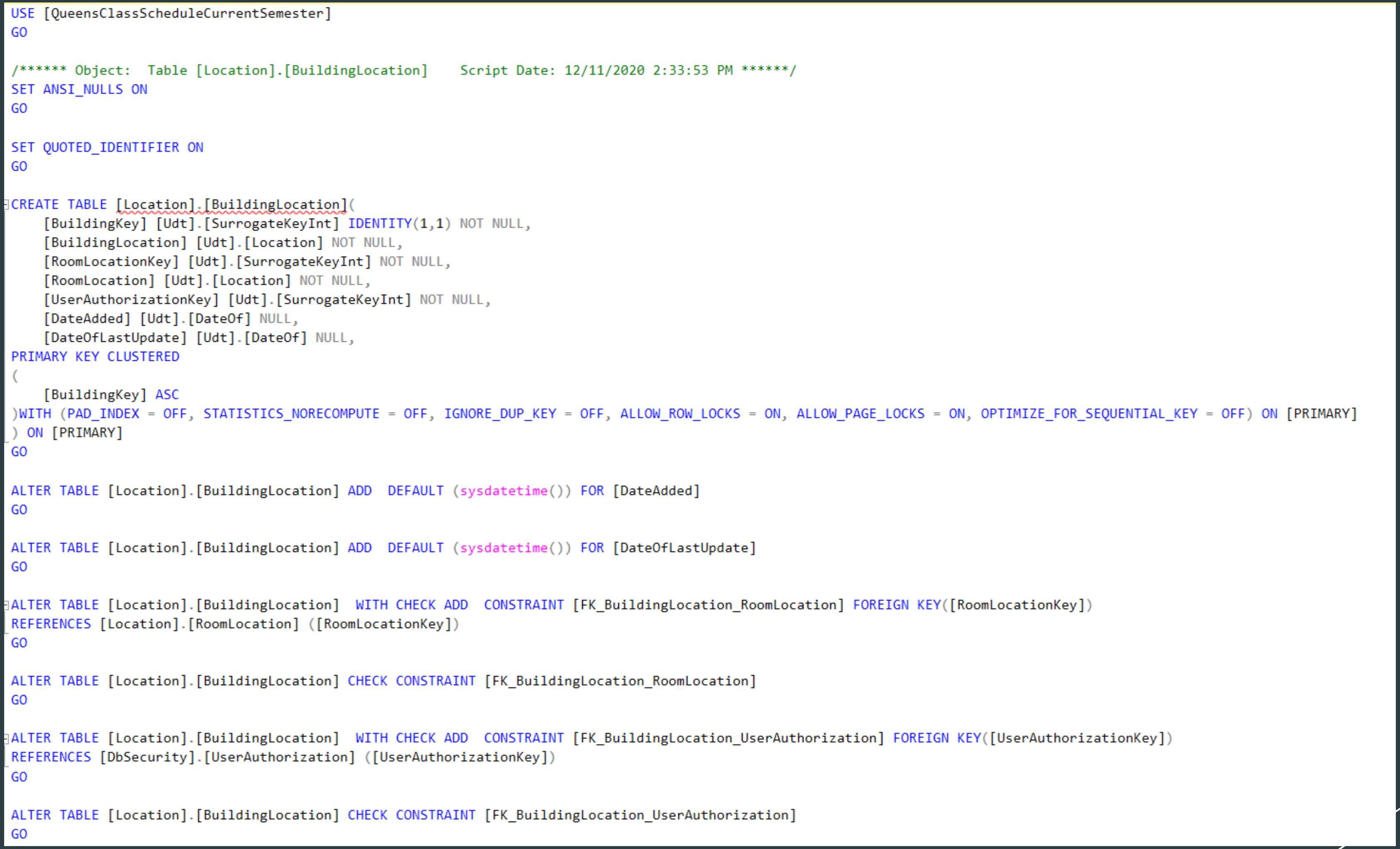
```sql
-- ================================================
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the procedure.
-- ================================================
SET ANSI_NULLS ON;
GO
SET QUOTED_IDENTIFIER ON;
GO
-- =============================================
-- Author:      <Haibo Liu>
-- Create date: <12/3/2020>
-- Description: <Load Data Into BuildingLocation>
-- =============================================
ALTER PROCEDURE [Project3].[Load_BuildingLocation] @UserAuthorizationKey [Udt].[SurrogateKeyInt]
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @DateAdded [Udt].[DateOf];
    SET @DateAdded = SYSDATETIME();

    DECLARE @DateOfLastUpdate [Udt].[DateOf];
    SET @DateOfLastUpdate = SYSDATETIME();

    DECLARE @StartingDateTime [Udt].[DateOf];
    SET @StartingDateTime = SYSDATETIME();

    DECLARE @SQL AS NVARCHAR(MAX)
    SET @SQL='CREATE OR ALTER VIEW G10_4.uvw_BuildingLocation1
            AS
        SELECT DISTINCT
       ROW_NUMBER() OVER( ORDER BY Sec ) AS id,
       COALESCE(NULLIF([Location], '' ''), ''TBA'') AS Location,
       COALESCE(NULLIF(SUBSTRING([Location], 1, 2), '' ''), ''TBA'') AS BuildingLocation
        FROM groupnUploadfile.CoursesSpring2017;


        '
    EXEC (@SQL)


    DECLARE @SQL2 AS NVARCHAR(MAX)
    SET @SQL2 = 'CREATE OR ALTER VIEW G10_4.uvw_BuildingLocation2
                AS
                SELECT DISTINCT
                Location,
                BuildingLocation
                FROM G10_4.uvw_BuildingLocation1'
    EXEC (@SQL2)

    DECLARE @SQL3 AS NVARCHAR(MAX)
    SET @SQL3 ='CREATE OR ALTER VIEW G10_4.uvw_BuildingLocation3
                AS
```



```sql
    DECLARE @SQL4 AS NVARCHAR(MAX)
    SET @SQL4 = 'CREATE OR ALTER VIEW G10_4.uvw_BuildingLocation4
                AS
                SELECT A.[Location],
                A.BuildingLocation,
                B.RoomLocation,
                B.RoomLocationKey
                FROM G10_4.uvw_BuildingLocation3 AS A FULL OUTER JOIN Location.RoomLocation AS B ON A.id = B.RoomLocationKey '
    EXEC (@SQL4)

    INSERT INTO [Location].BuildingLocation(BuildingLocation,RoomLocation,RoomLocationKey,UserAuthorizationKey,DateAdded,DateOfLastUpdate)
    SELECT DISTINCT
        BuildingLocation,RoomLocation,RoomLocationKey,@UserAuthorizationKey,@DateAdded,@DateOfLastUpdate
        FROM G10_4.uvw_BuildingLocation4




    DECLARE @EndingDateTime [Udt].[DateOf];
    SET @EndingDateTime = SYSDATETIME();

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount =
    (
        SELECT COUNT(*) FROM [Location].[BuildingLocation]
    );

    EXEC [Process].[usp_TrackWorkFlow] 'Loads data into [Location].[BuildingLocation]',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;

    SELECT *
    FROM [Location].[BuildingLocation];
ND;
```

Note: Code is provided in the notes tab on this slide

BuildingLocation Table Code



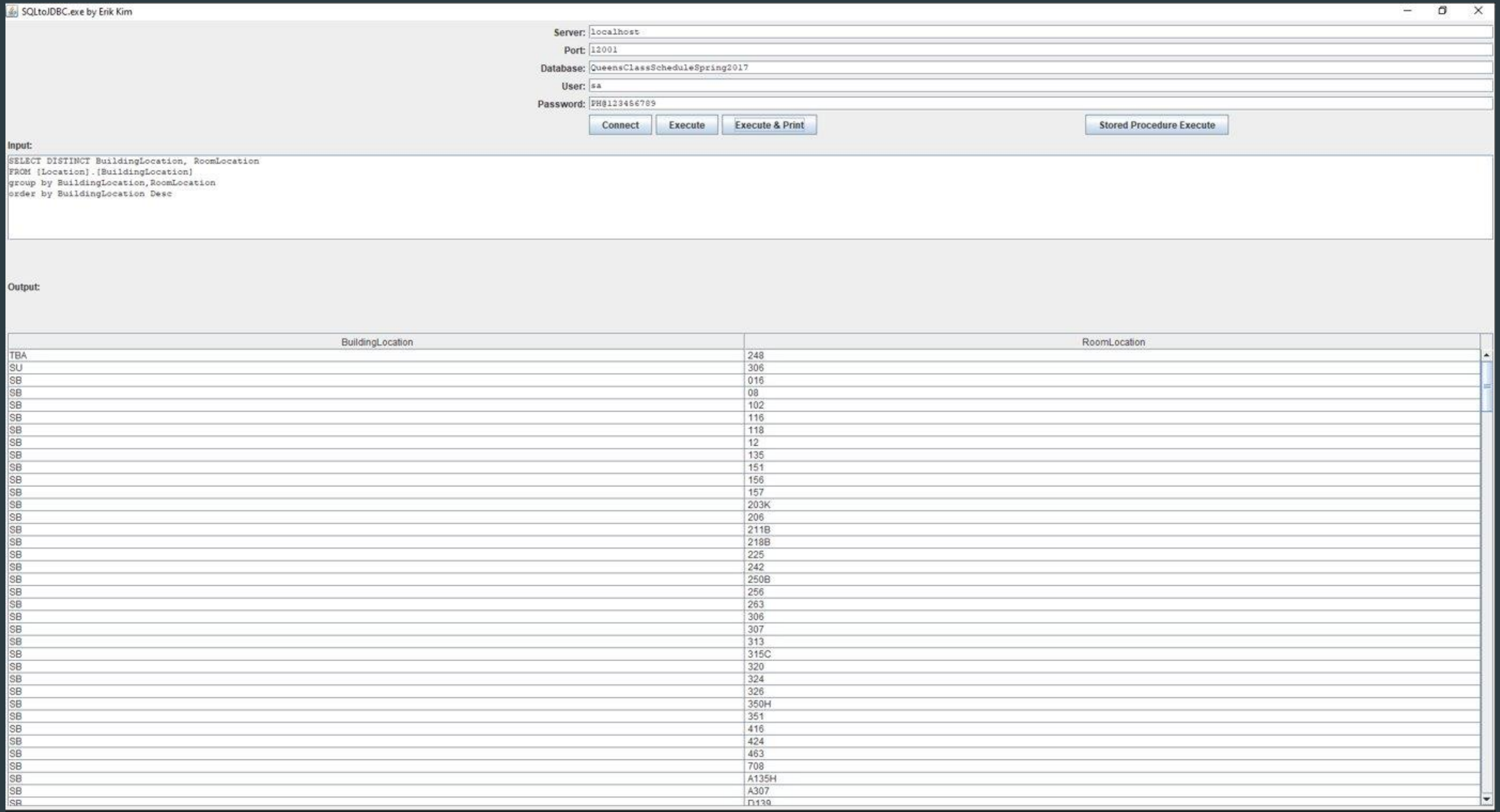
```sql
USE [QueensClassScheduleCurrentSemester]
GO

/****** Object:  Table [Location].[BuildingLocation]    Script Date: 12/11/2020 2:33:53 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [Location].[BuildingLocation](
    [BuildingKey] [Udt].[SurrogateKeyInt] IDENTITY(1,1) NOT NULL,
    [BuildingLocation] [Udt].[Location] NOT NULL,
    [RoomLocationKey] [Udt].[SurrogateKeyInt] NOT NULL,
    [RoomLocation] [Udt].[Location] NOT NULL,
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] NOT NULL,
    [DateAdded] [Udt].[DateOf] NULL,
    [DateOfLastUpdate] [Udt].[DateOf] NULL,
PRIMARY KEY CLUSTERED
(
    [BuildingKey] 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

ALTER TABLE [Location].[BuildingLocation] ADD  DEFAULT (sysdatetime()) FOR [DateAdded]
GO

ALTER TABLE [Location].[BuildingLocation] ADD  DEFAULT (sysdatetime()) FOR [DateOfLastUpdate]
GO

ALTER TABLE [Location].[BuildingLocation]  WITH CHECK ADD  CONSTRAINT [FK_BuildingLocation_RoomLocation] FOREIGN KEY([RoomLocationKey])
REFERENCES [Location].[RoomLocation] ([RoomLocationKey])
GO

ALTER TABLE [Location].[BuildingLocation] CHECK CONSTRAINT [FK_BuildingLocation_RoomLocation]
GO

ALTER TABLE [Location].[BuildingLocation]  WITH CHECK ADD  CONSTRAINT [FK_BuildingLocation_UserAuthorization] FOREIGN KEY([UserAuthorizationKey])
REFERENCES [DbSecurity].[UserAuthorization] ([UserAuthorizationKey])
GO

ALTER TABLE [Location].[BuildingLocation] CHECK CONSTRAINT [FK_BuildingLocation_UserAuthorization]
GO
```

Note: Code is provided in the notes tab on this slide

Haibo’s Queries

- Free query_ALL BUILDING: show all Buildinglocation that match up the RoomLocation
- Free query_KY BUILDING: find all building that begin with letter KY

Free query 1



Server: localhost
Port: 12001
Database: QueensClassScheduleSpring2017
User: sa
Password: PH@123456789

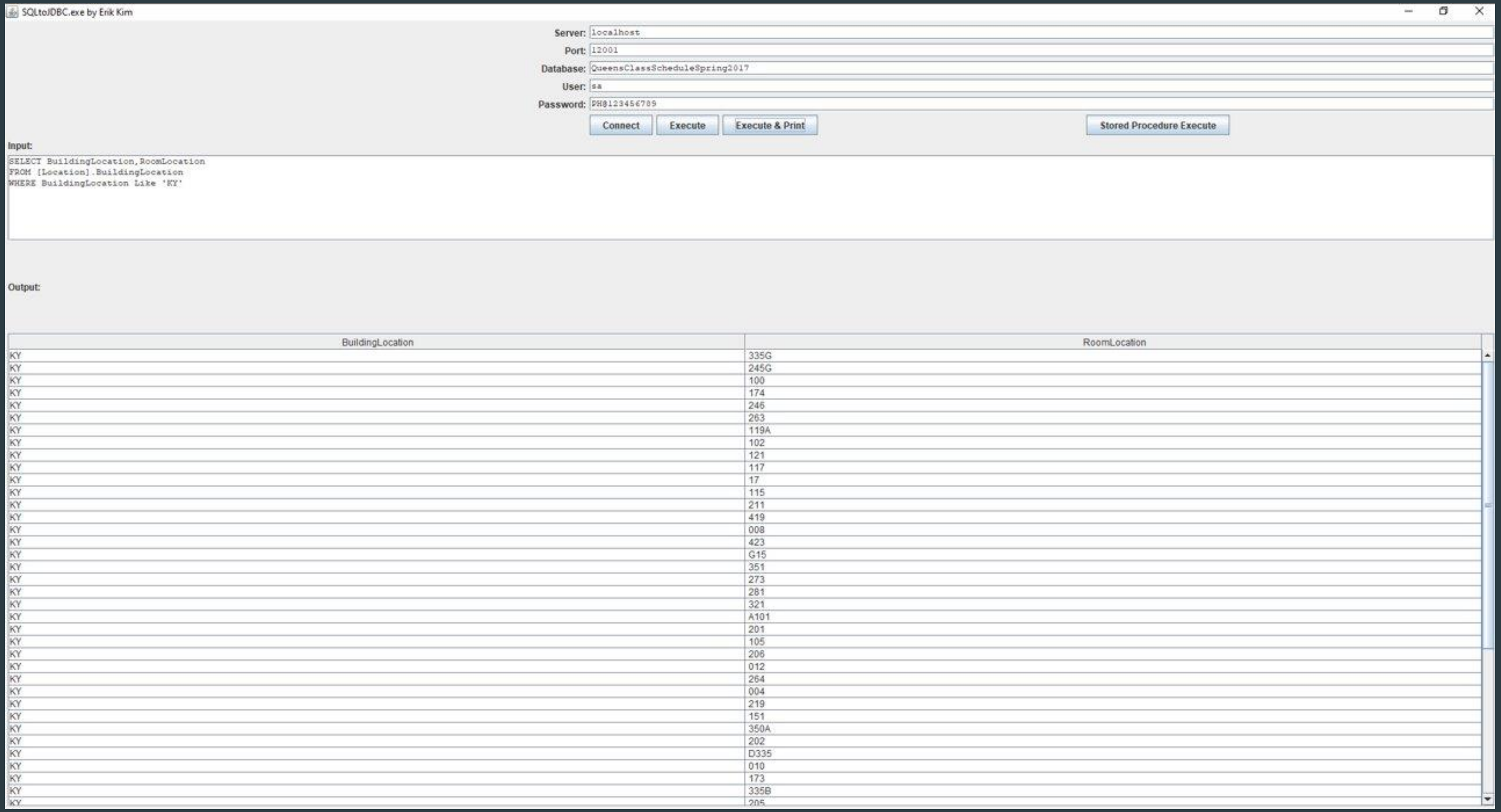
Input:

```
SELECT DISTINCT BuildingLocation, RoomLocation
FROM [Location].[BuildingLocation]
group by BuildingLocation,RoomLocation
order by BuildingLocation Desc
```

Output:

BuildingLocation	RoomLocation
TBA	248
SU	306
SB	016
SB	08
SB	102
SB	116
SB	118
SB	12
SB	135
SB	151
SB	156
SB	157
SB	203K
SB	206
SB	211B
SB	218B
SB	225
SB	242
SB	250B
SB	256
SB	263
SB	306
SB	307
SB	313
SB	315C
SB	320
SB	324
SB	326
SB	350H
SB	351
SB	416
SB	424
SB	463
SB	708
SB	A135H
SB	A307
SB	D139

Free query 2



Server: localhost
Port: 12001
Database: QueensClassScheduleSpring2017
User: sa
Password: PH@123456789

Input:

```
SELECT BuildingLocation,RoomLocation
FROM [Location].BuildingLocation
WHERE BuildingLocation Like 'KY'
```

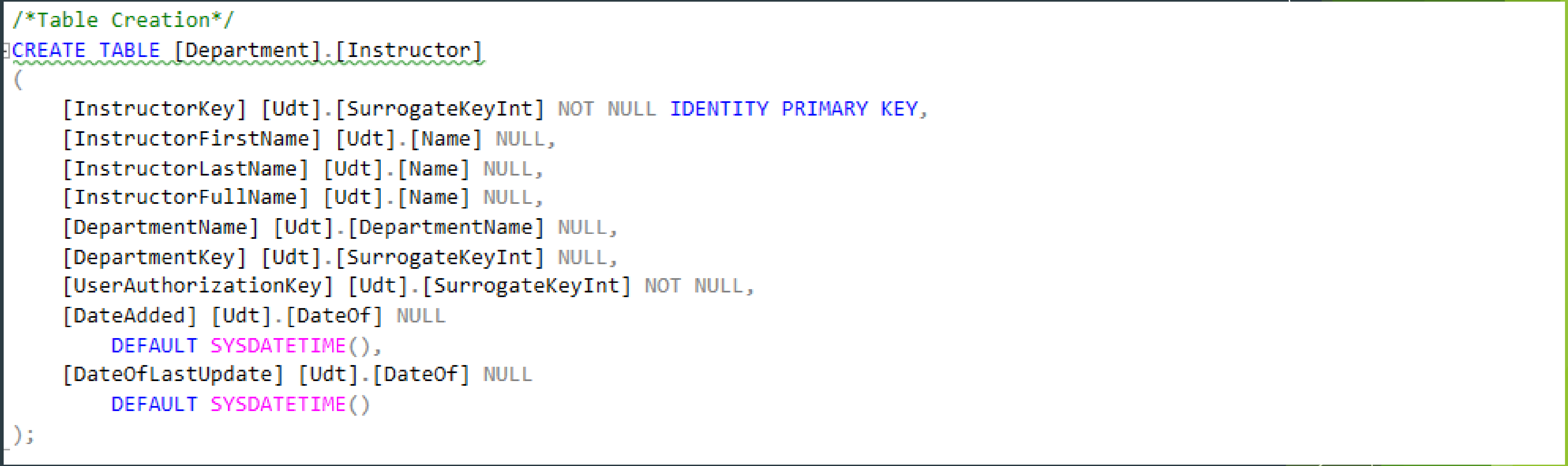
Output:

BuildingLocation	RoomLocation
KY	335G
KY	245G
KY	100
KY	174
KY	246
KY	263
KY	119A
KY	102
KY	121
KY	117
KY	17
KY	115
KY	211
KY	419
KY	008
KY	423
KY	G15
KY	351
KY	273
KY	281
KY	321
KY	A101
KY	201
KY	105
KY	206
KY	012
KY	264
KY	004
KY	219
KY	151
KY	350A
KY	202
KY	D335
KY	010
KY	173
KY	335B
KY	205

Jonathan’s Tables & Stored Procedures

- Instructor Table
- Instructor Stored Procedure

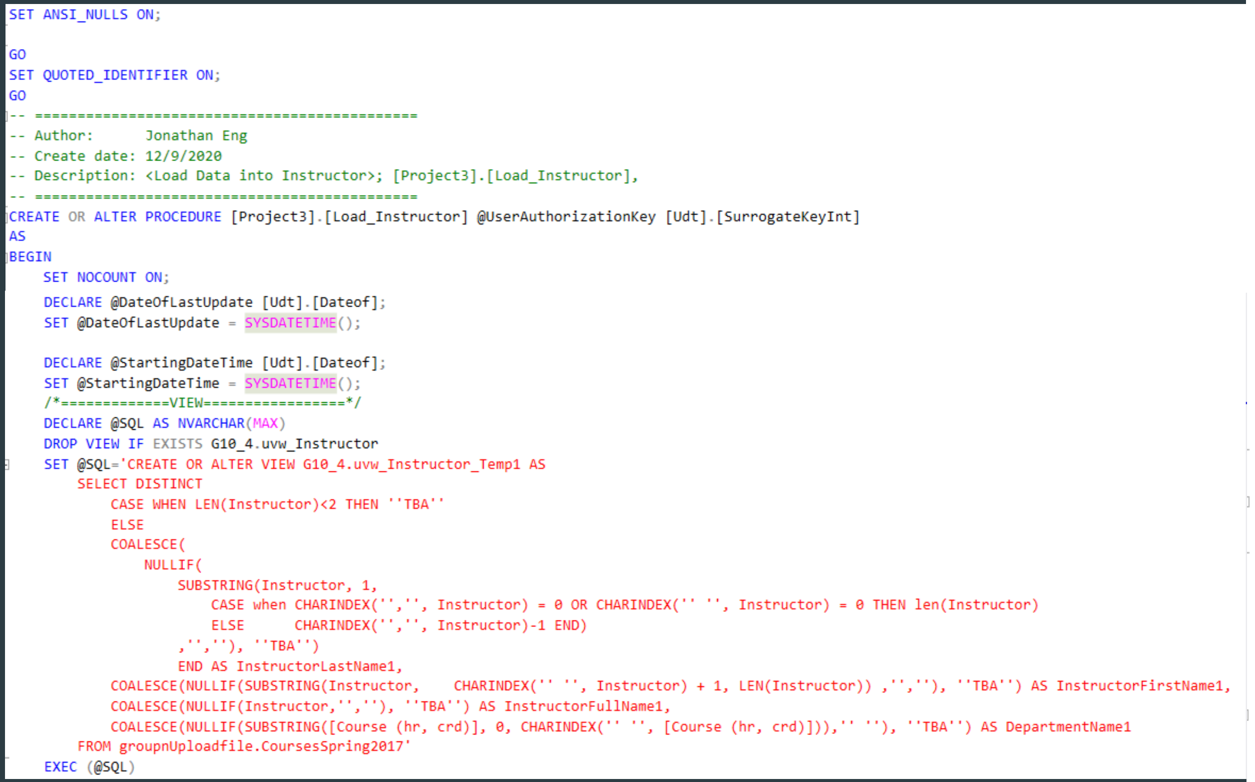
Instructor Table Code



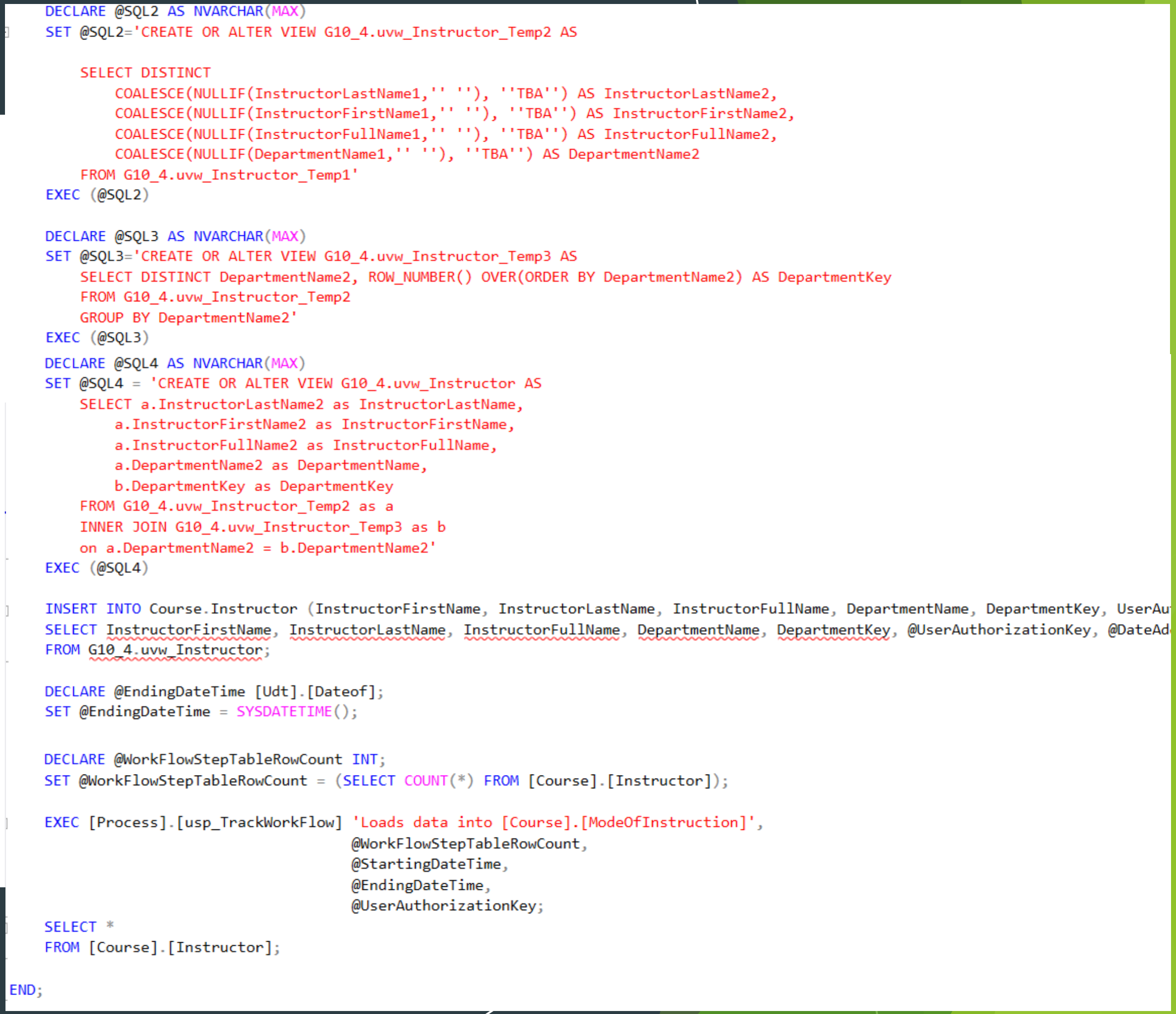
```
/*Table Creation*/
CREATE TABLE [Department].[Instructor]
(
    [InstructorKey] [Udt].[SurrogateKeyInt] NOT NULL IDENTITY PRIMARY KEY,
    [InstructorFirstName] [Udt].[Name] NULL,
    [InstructorLastName] [Udt].[Name] NULL,
    [InstructorFullName] [Udt].[Name] NULL,
    [DepartmentName] [Udt].[DepartmentName] NULL,
    [DepartmentKey] [Udt].[SurrogateKeyInt] NULL,
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] NOT NULL,
    [DateAdded] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME(),
    [DateOfLastUpdate] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME()
);
```

Note: Code is provided in the notes tab on this slide

Instructor Stored Procedure



```sql
SET ANSI_NULLS ON;

GO
SET QUOTED_IDENTIFIER ON;
GO
-- ============================================
-- Author:      Jonathan Eng
-- Create date: 12/9/2020
-- Description: <Load Data into Instructor>; [Project3].[Load_Instructor],
-- ============================================
CREATE OR ALTER PROCEDURE [Project3].[Load_Instructor] @UserAuthorizationKey [Udt].[SurrogateKeyInt]
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @DateOfLastUpdate [Udt].[Dateof];
    SET @DateOfLastUpdate = SYSDATETIME();

    DECLARE @StartingDateTime [Udt].[Dateof];
    SET @StartingDateTime = SYSDATETIME();
    /*============VIEW================*/
    DECLARE @SQL AS NVARCHAR(MAX)
    DROP VIEW IF EXISTS G10_4.uvw_Instructor
    SET @SQL='CREATE OR ALTER VIEW G10_4.uvw_Instructor_Temp1 AS
        SELECT DISTINCT
            CASE WHEN LEN(Instructor)<2 THEN ''TBA''
            ELSE
            COALESCE(
                NULLIF(
                    SUBSTRING(Instructor, 1,
                        CASE when CHARINDEX('','', Instructor) = 0 OR CHARINDEX('' '', Instructor) = 0 THEN len(Instructor)
                        ELSE      CHARINDEX('','', Instructor)-1 END)
                    ,'','''), ''TBA'')
                    END AS InstructorLastName1,
            COALESCE(NULLIF(SUBSTRING(Instructor,    CHARINDEX('' '', Instructor) + 1, LEN(Instructor)) ,'','''), ''TBA'') AS InstructorFirstName1,
            COALESCE(NULLIF(Instructor,'','''), ''TBA'') AS InstructorFullName1,
            COALESCE(NULLIF(SUBSTRING([Course (hr, crd)], 0, CHARINDEX('' '', [Course (hr, crd)])),'' ''), ''TBA'') AS DepartmentName1
        FROM groupnUploadfile.CoursesSpring2017'
    EXEC (@SQL)
```

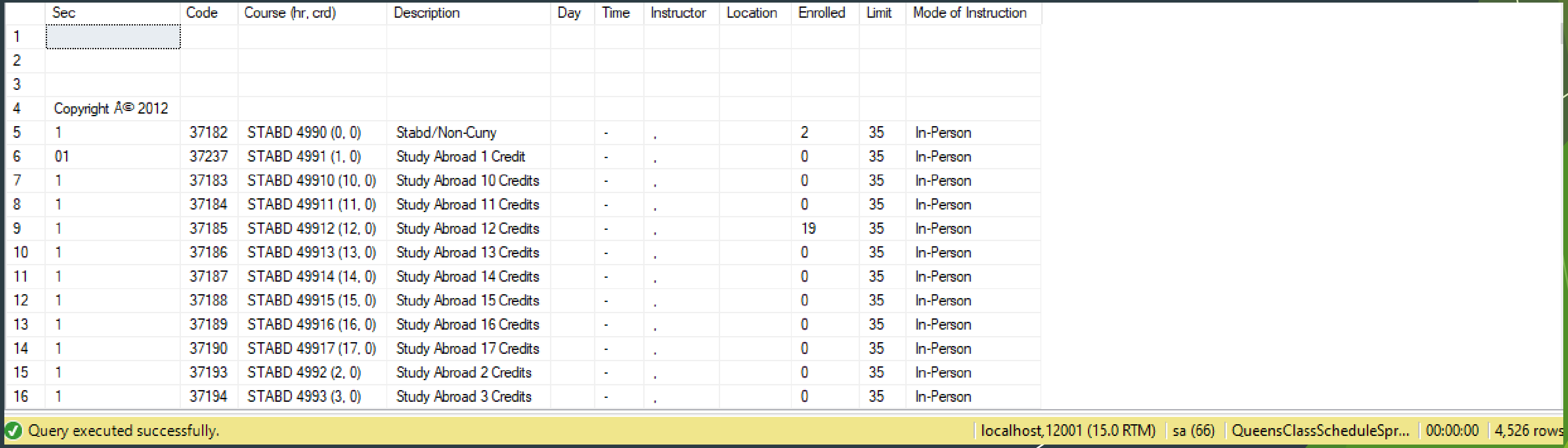


```sql
    DECLARE @SQL2 AS NVARCHAR(MAX)
    SET @SQL2='CREATE OR ALTER VIEW G10_4.uvw_Instructor_Temp2 AS

        SELECT DISTINCT
            COALESCE(NULLIF(InstructorLastName1,'' ''), ''TBA'') AS InstructorLastName2,
            COALESCE(NULLIF(InstructorFirstName1,'' ''), ''TBA'') AS InstructorFirstName2,
            COALESCE(NULLIF(InstructorFullName1,'' ''), ''TBA'') AS InstructorFullName2,
            COALESCE(NULLIF(DepartmentName1,'' ''), ''TBA'') AS DepartmentName2
        FROM G10_4.uvw_Instructor_Temp1'
    EXEC (@SQL2)

    DECLARE @SQL3 AS NVARCHAR(MAX)
    SET @SQL3='CREATE OR ALTER VIEW G10_4.uvw_Instructor_Temp3 AS
        SELECT DISTINCT DepartmentName2, ROW_NUMBER() OVER(ORDER BY DepartmentName2) AS DepartmentKey
        FROM G10_4.uvw_Instructor_Temp2
        GROUP BY DepartmentName2'
    EXEC (@SQL3)
    DECLARE @SQL4 AS NVARCHAR(MAX)
    SET @SQL4 = 'CREATE OR ALTER VIEW G10_4.uvw_Instructor AS
        SELECT a.InstructorLastName2 as InstructorLastName,
            a.InstructorFirstName2 as InstructorFirstName,
            a.InstructorFullName2 as InstructorFullName,
            a.DepartmentName2 as DepartmentName,
            b.DepartmentKey as DepartmentKey
        FROM G10_4.uvw_Instructor_Temp2 as a
        INNER JOIN G10_4.uvw_Instructor_Temp3 as b
        on a.DepartmentName2 = b.DepartmentName2'
    EXEC (@SQL4)

    INSERT INTO Course.Instructor (InstructorFirstName, InstructorLastName, InstructorFullName, DepartmentName, DepartmentKey, UserAu
    SELECT InstructorFirstName, InstructorLastName, InstructorFullName, DepartmentName, DepartmentKey, @UserAuthorizationKey, @DateAd
    FROM G10_4.uvw_Instructor;

    DECLARE @EndingDateTime [Udt].[Dateof];
    SET @EndingDateTime = SYSDATETIME();

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = (SELECT COUNT(*) FROM [Course].[Instructor]);

    EXEC [Process].[usp_TrackWorkFlow] 'Loads data into [Course].[ModeOfInstruction]',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;

    SELECT *
    FROM [Course].[Instructor];

END;
```

Note: Code is provided in the notes tab on this slide

Jonathan’s Data Anomalies and Fixes

- Instructor and Course Column contained *Blanks (), spaces ( ), and Dashes(-)*
  - Caused Issues with the *Substring Function* when attempting to grab only the instructors first and last name and Department Name separately due to the *Substring* looking for a *CharAt space* or *comma*
  - This was accounted for by using CASE WHEN's for when the Length is less than 2, the value changes to 'TBA'



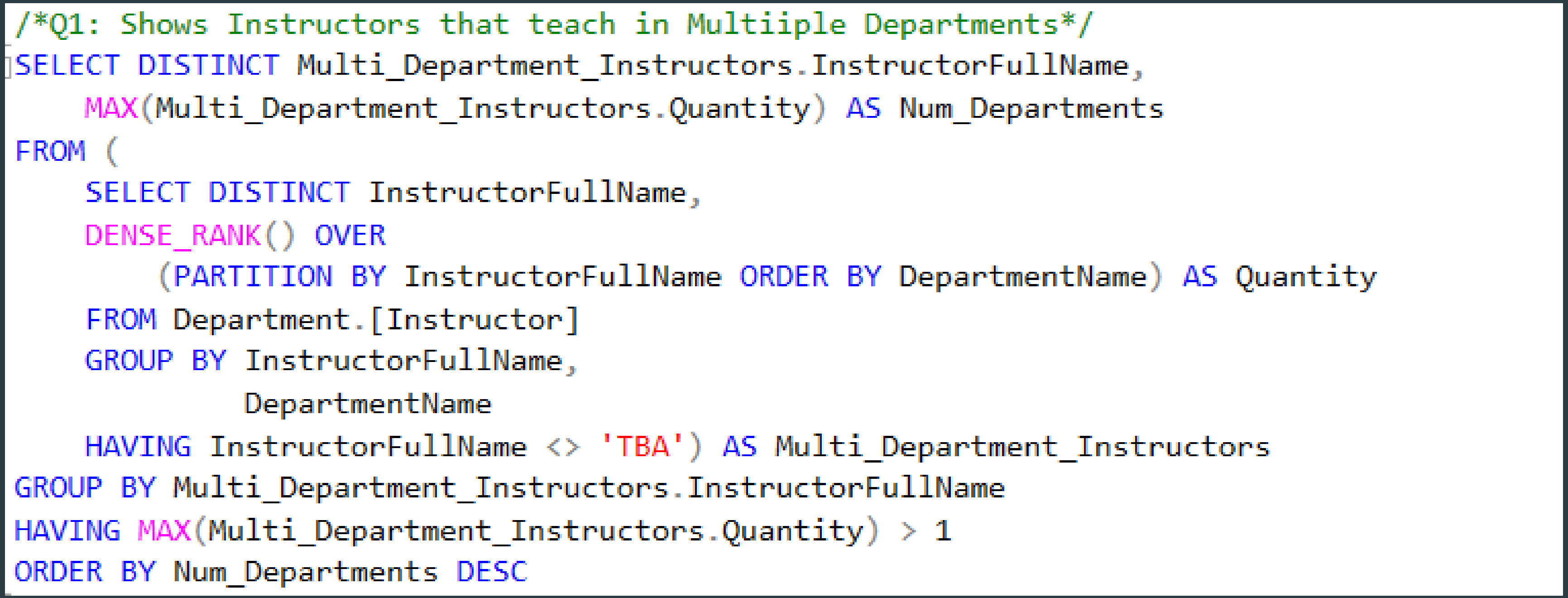
	Sec	Code	Course (hr, crd)	Description	Day	Time	Instructor	Location	Enrolled	Limit	Mode of Instruction
1											
2											
3											
4	Copyright Â© 2012										
5	1	37182	STABD 4990 (0, 0)	Stabd/Non-Cuny		-	,		2	35	In-Person
6	01	37237	STABD 4991 (1, 0)	Study Abroad 1 Credit		-	,		0	35	In-Person
7	1	37183	STABD 49910 (10, 0)	Study Abroad 10 Credits		-	,		0	35	In-Person
8	1	37184	STABD 49911 (11, 0)	Study Abroad 11 Credits		-	,		0	35	In-Person
9	1	37185	STABD 49912 (12, 0)	Study Abroad 12 Credits		-	,		19	35	In-Person
10	1	37186	STABD 49913 (13, 0)	Study Abroad 13 Credits		-	,		0	35	In-Person
11	1	37187	STABD 49914 (14, 0)	Study Abroad 14 Credits		-	,		0	35	In-Person
12	1	37188	STABD 49915 (15, 0)	Study Abroad 15 Credits		-	,		0	35	In-Person
13	1	37189	STABD 49916 (16, 0)	Study Abroad 16 Credits		-	,		0	35	In-Person
14	1	37190	STABD 49917 (17, 0)	Study Abroad 17 Credits		-	,		0	35	In-Person
15	1	37193	STABD 4992 (2, 0)	Study Abroad 2 Credits		-	,		0	35	In-Person
16	1	37194	STABD 4993 (3, 0)	Study Abroad 3 Credits		-	,		0	35	In-Person

```
Query executed successfully.    localhost,12001 (15.0 RTM) | sa (66) | QueensClassScheduleSpr... | 00:00:00 | 4,526 rows
```

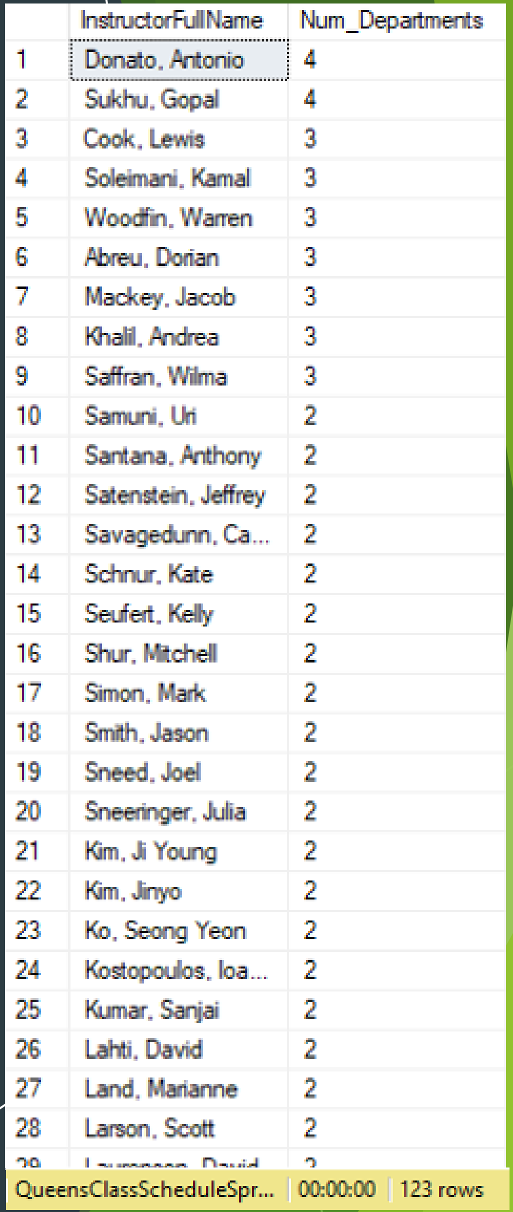
Jonathan’s Queries

- Query 1:
  Shows all Instructors teaching in Multiple Departments
- Free query:
  Shows the Amount & Percentage of Instructors in each Department

Query 1



```
/*Q1: Shows Instructors that teach in Multiple Departments*/
SELECT DISTINCT Multi_Department_Instructors.InstructorFullName,
    MAX(Multi_Department_Instructors.Quantity) AS Num_Departments
FROM (
    SELECT DISTINCT InstructorFullName,
    DENSE_RANK() OVER
        (PARTITION BY InstructorFullName ORDER BY DepartmentName) AS Quantity
    FROM Department.[Instructor]
    GROUP BY InstructorFullName,
             DepartmentName
    HAVING InstructorFullName <> 'TBA') AS Multi_Department_Instructors
GROUP BY Multi_Department_Instructors.InstructorFullName
HAVING MAX(Multi_Department_Instructors.Quantity) > 1
ORDER BY Num_Departments DESC
```

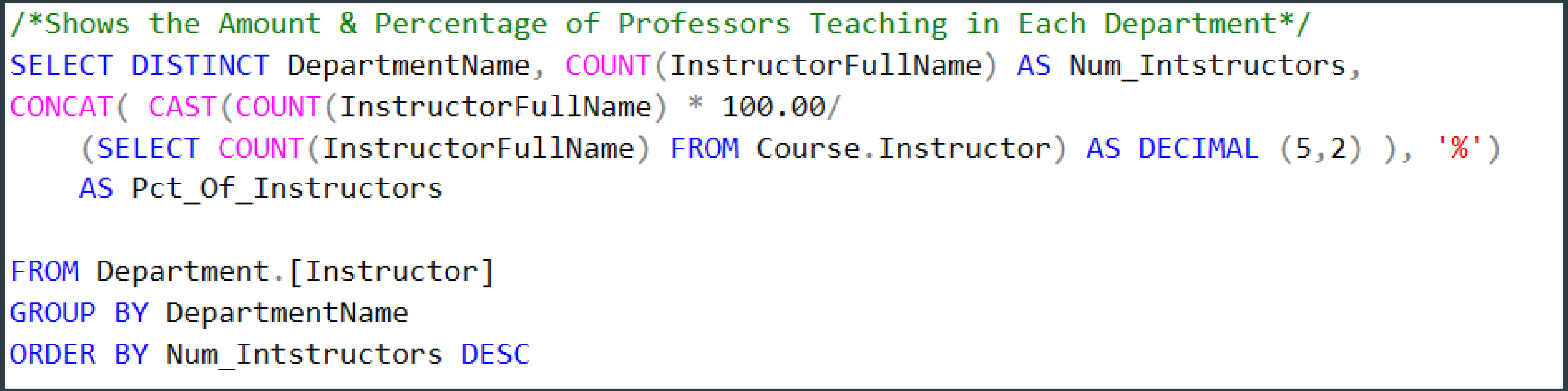


	InstructorFullName	Num_Departments
1	Donato, Antonio	4
2	Sukhu, Gopal	4
3	Cook, Lewis	3
4	Soleimani, Kamal	3
5	Woodfin, Warren	3
6	Abreu, Dorian	3
7	Mackey, Jacob	3
8	Khalil, Andrea	3
9	Saffran, Wilma	3
10	Samuni, Uri	2
11	Santana, Anthony	2
12	Satenstein, Jeffrey	2
13	Savagedunn, Ca...	2
14	Schnur, Kate	2
15	Seufert, Kelly	2
16	Shur, Mitchell	2
17	Simon, Mark	2
18	Smith, Jason	2
19	Sneed, Joel	2
20	Sneeringer, Julia	2
21	Kim, Ji Young	2
22	Kim, Jinyo	2
23	Ko, Seong Yeon	2
24	Kostopoulos, Ioa...	2
25	Kumar, Sanjai	2
26	Lahti, David	2
27	Land, Marianne	2
28	Larson, Scott	2
29	[illegible]	[illegible]

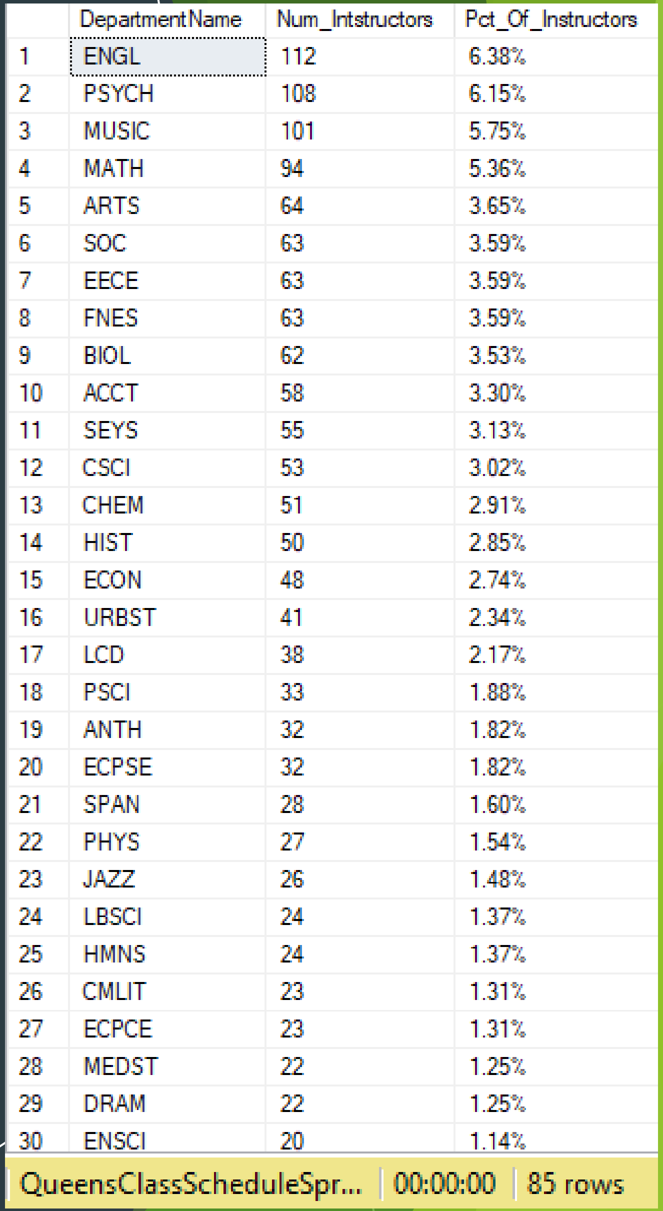
QueensClassScheduleSpr... | 00:00:00 | 123 rows

Note: Code is provided in the notes tab on this slide

Free query



```
/*Shows the Amount & Percentage of Professors Teaching in Each Department*/
SELECT DISTINCT DepartmentName, COUNT(InstructorFullName) AS Num_Intstructors,
CONCAT( CAST(COUNT(InstructorFullName) * 100.00/
    (SELECT COUNT(InstructorFullName) FROM Course.Instructor) AS DECIMAL (5,2) ), '%')
    AS Pct_Of_Instructors

FROM Department.[Instructor]
GROUP BY DepartmentName
ORDER BY Num_Intstructors DESC
```



	DepartmentName	Num_Intstructors	Pct_Of_Instructors
1	ENGL	112	6.38%
2	PSYCH	108	6.15%
3	MUSIC	101	5.75%
4	MATH	94	5.36%
5	ARTS	64	3.65%
6	SOC	63	3.59%
7	EECE	63	3.59%
8	FNES	63	3.59%
9	BIOL	62	3.53%
10	ACCT	58	3.30%
11	SEYS	55	3.13%
12	CSCI	53	3.02%
13	CHEM	51	2.91%
14	HIST	50	2.85%
15	ECON	48	2.74%
16	URBST	41	2.34%
17	LCD	38	2.17%
18	PSCI	33	1.88%
19	ANTH	32	1.82%
20	ECPSE	32	1.82%
21	SPAN	28	1.60%
22	PHYS	27	1.54%
23	JAZZ	26	1.48%
24	LBSCI	24	1.37%
25	HMNS	24	1.37%
26	CMLIT	23	1.31%
27	ECPCE	23	1.31%
28	MEDST	22	1.25%
29	DRAM	22	1.25%
30	ENSCI	20	1.14%

QueensClassScheduleSpr... | 00:00:00 | 85 rows

Note: Code is provided in the notes tab on this slide

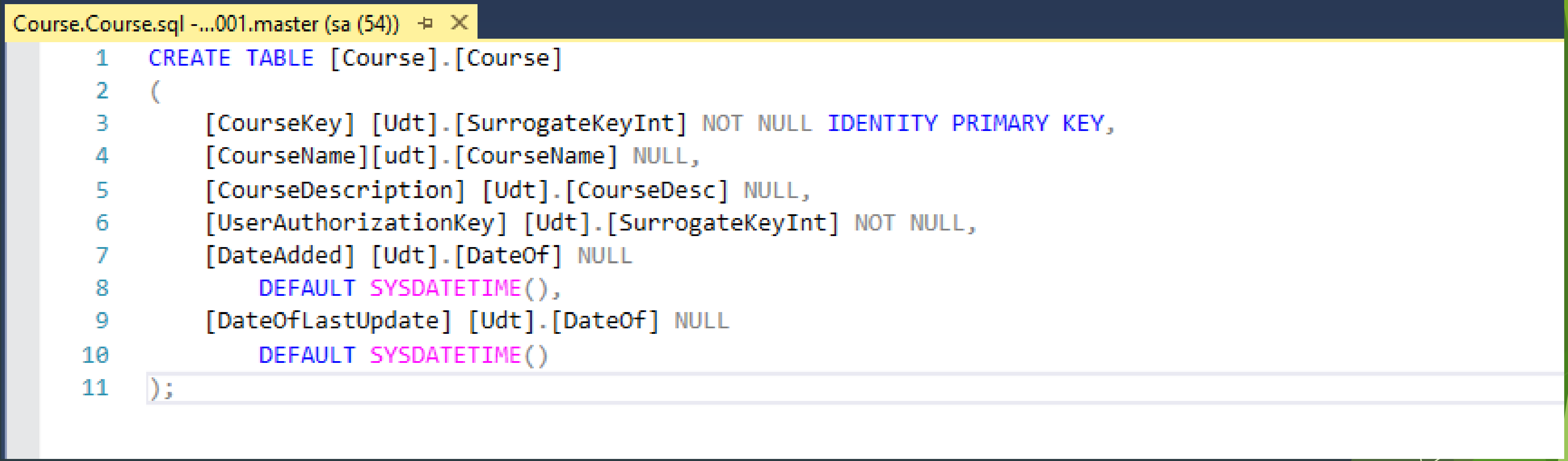
Jamil’s Tables & Stored Procedures

- Course Table
- Course Stored Procedure

Jamil’s Data Anomalies and Fixes

- There were very little Data Anomalies in my table. There was one row that didn't show anything so to fix this I coalesced the blank row with 'TBA'

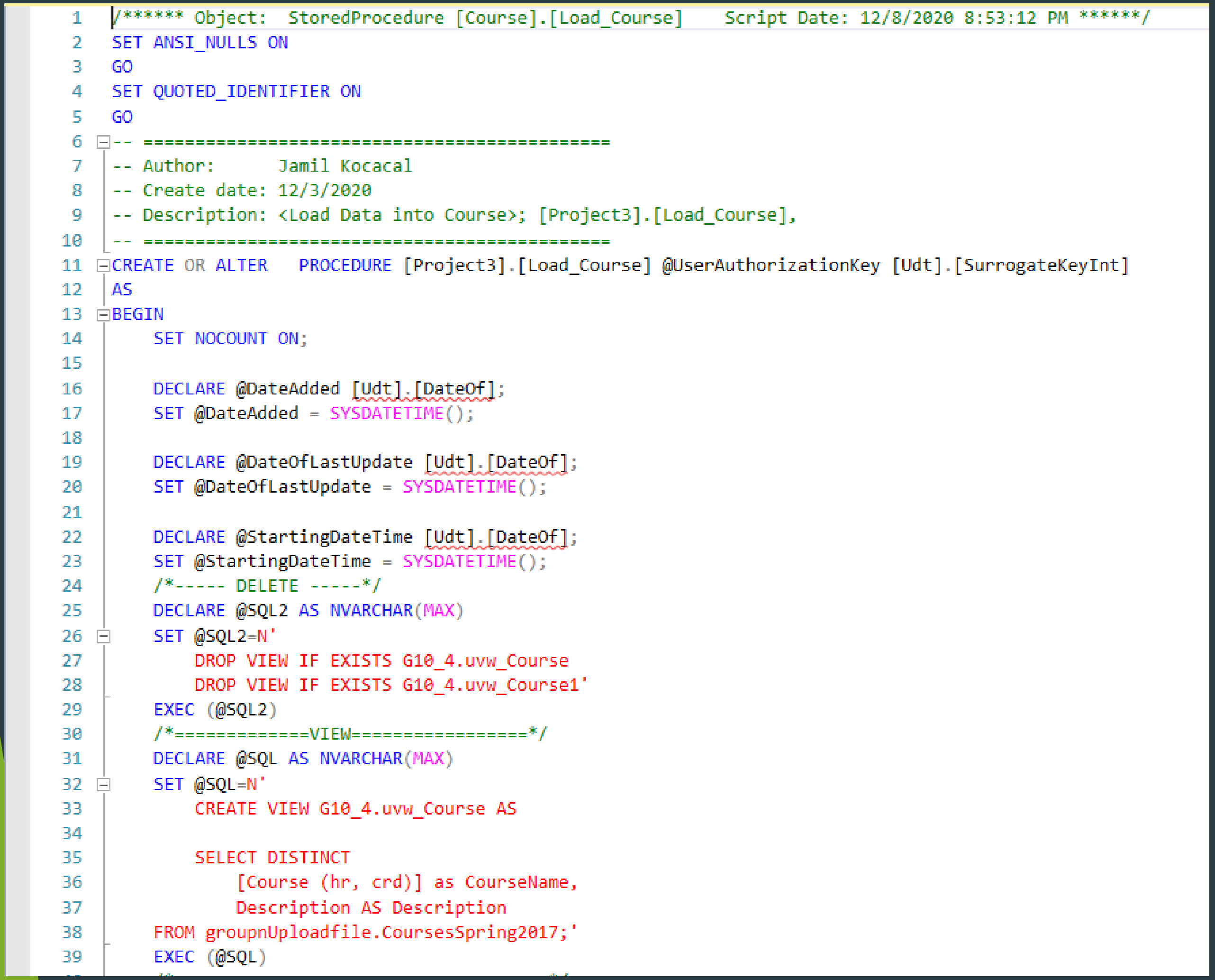
Course Table Code


Course.Course.sql -...001.master (sa (54))

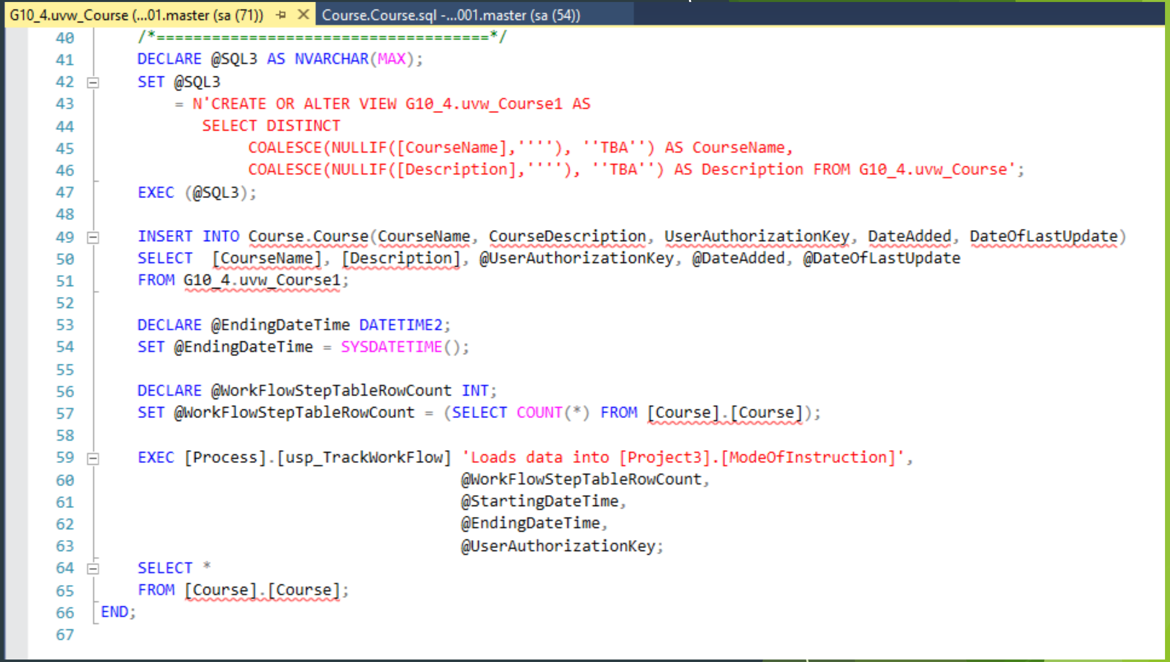
```
CREATE TABLE [Course].[Course]
(
    [CourseKey] [Udt].[SurrogateKeyInt] NOT NULL IDENTITY PRIMARY KEY,
    [CourseName][udt].[CourseName] NULL,
    [CourseDescription] [Udt].[CourseDesc] NULL,
    [UserAuthorizationKey] [Udt].[SurrogateKeyInt] NOT NULL,
    [DateAdded] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME(),
    [DateOfLastUpdate] [Udt].[DateOf] NULL
        DEFAULT SYSDATETIME()
);
```

Note: Code is provided in the notes tab on this slide

Course Stored Procedure Code



```sql
/****** Object:  StoredProcedure [Course].[Load_Course]    Script Date: 12/8/2020 8:53:12 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Jamil Kocacal
-- Create date: 12/3/2020
-- Description: <Load Data into Course>; [Project3].[Load_Course],
-- =============================================
CREATE OR ALTER   PROCEDURE [Project3].[Load_Course] @UserAuthorizationKey [Udt].[SurrogateKeyInt]
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @DateAdded [Udt].[DateOf];
    SET @DateAdded = SYSDATETIME();

    DECLARE @DateOfLastUpdate [Udt].[DateOf];
    SET @DateOfLastUpdate = SYSDATETIME();

    DECLARE @StartingDateTime [Udt].[DateOf];
    SET @StartingDateTime = SYSDATETIME();
    /*----- DELETE -----*/
    DECLARE @SQL2 AS NVARCHAR(MAX)
    SET @SQL2=N'
        DROP VIEW IF EXISTS G10_4.uvw_Course
        DROP VIEW IF EXISTS G10_4.uvw_Course1'
    EXEC (@SQL2)
    /*============VIEW================*/
    DECLARE @SQL AS NVARCHAR(MAX)
    SET @SQL=N'
        CREATE VIEW G10_4.uvw_Course AS

        SELECT DISTINCT
            [Course (hr, crd)] as CourseName,
            Description AS Description
    FROM groupnUploadfile.CoursesSpring2017;'
    EXEC (@SQL)
```


G10_4.uvw_Course (...01.master (sa (71))
Course.Course.sql -...001.master (sa (54))

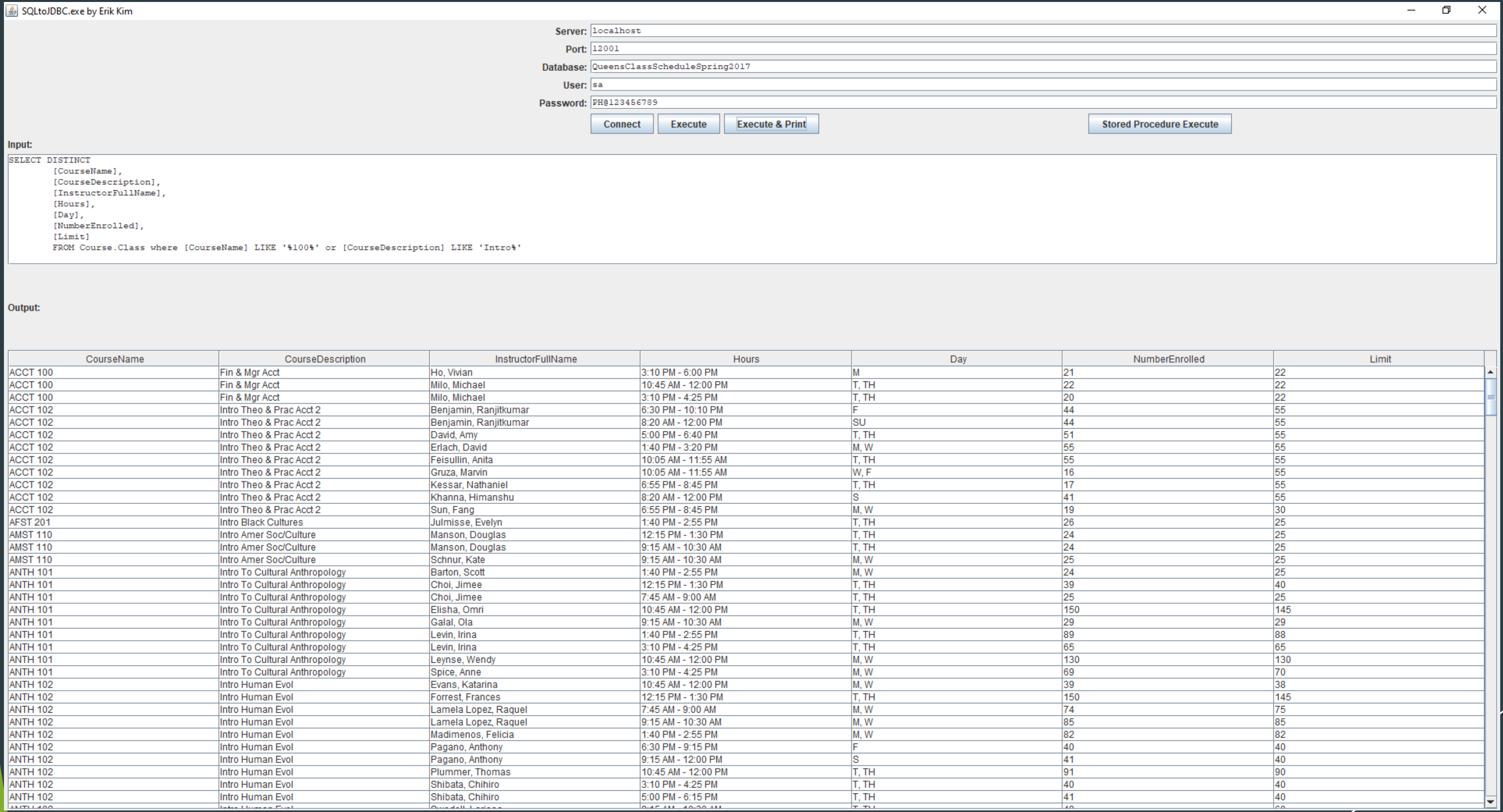
```sql
    /*==================================*/
    DECLARE @SQL3 AS NVARCHAR(MAX);
    SET @SQL3
        = N'CREATE OR ALTER VIEW G10_4.uvw_Course1 AS
          SELECT DISTINCT
               COALESCE(NULLIF([CourseName],''''), ''TBA'') AS CourseName,
               COALESCE(NULLIF([Description],''''), ''TBA'') AS Description FROM G10_4.uvw_Course';
    EXEC (@SQL3);

    INSERT INTO Course.Course(CourseName, CourseDescription, UserAuthorizationKey, DateAdded, DateOfLastUpdate)
    SELECT  [CourseName], [Description], @UserAuthorizationKey, @DateAdded, @DateOfLastUpdate
    FROM G10_4.uvw_Course1;

    DECLARE @EndingDateTime DATETIME2;
    SET @EndingDateTime = SYSDATETIME();

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = (SELECT COUNT(*) FROM [Course].[Course]);

    EXEC [Process].[usp_TrackWorkFlow] 'Loads data into [Project3].[ModeOfInstruction]',
                                       @WorkFlowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;

    SELECT *
    FROM [Course].[Course];
END;
```

Note: Code is provided in the notes tab on this slide

Jamil’s Queries

- FreeQuery.CSCI_Search: What are the available CSCI courses right now?
- FreeQuery.EasySearch: What are the available intro or 100 level courses?

FreeQuery.EasySearch:



Server: localhost
Port: 12001
Database: QueensClassScheduleSpring2017
User: sa
Password: PH@123456789

Input:

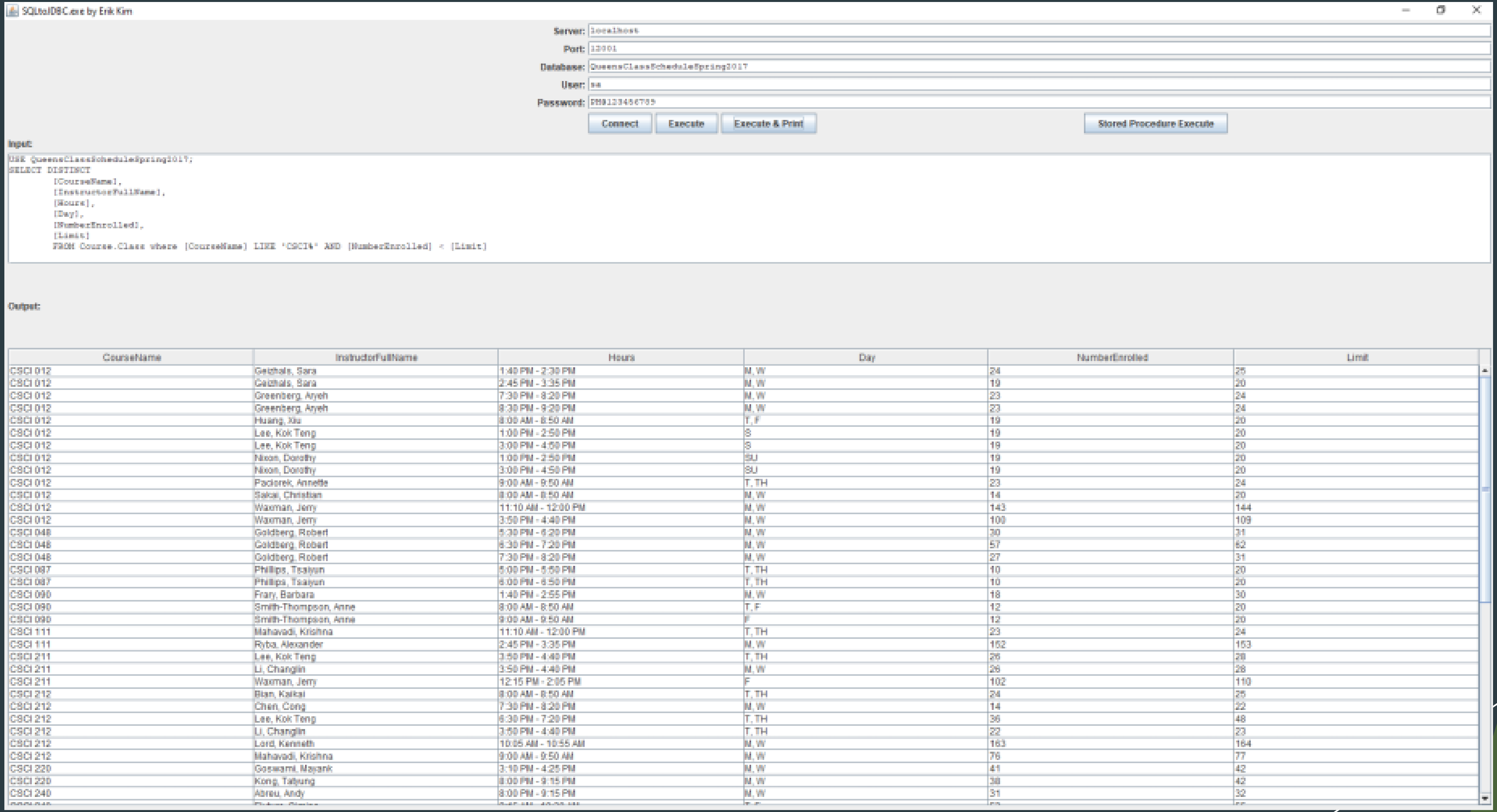
```
SELECT DISTINCT
        [CourseName],
        [CourseDescription],
        [InstructorFullName],
        [Hours],
        [Day],
        [NumberEnrolled],
        [Limit]
        FROM Course.Class where [CourseName] LIKE '%100%' or [CourseDescription] LIKE 'Intro%'
```

Output:

CourseName	CourseDescription	InstructorFullName	Hours	Day	NumberEnrolled	Limit
ACCT 100	Fin & Mgr Acct	Ho, Vivian	3:10 PM - 6:00 PM	M	21	22
ACCT 100	Fin & Mgr Acct	Milo, Michael	10:45 AM - 12:00 PM	T, TH	22	22
ACCT 100	Fin & Mgr Acct	Milo, Michael	3:10 PM - 4:25 PM	T, TH	20	22
ACCT 102	Intro Theo & Prac Acct 2	Benjamin, Ranjitkumar	6:30 PM - 10:10 PM	F	44	55
ACCT 102	Intro Theo & Prac Acct 2	Benjamin, Ranjitkumar	8:20 AM - 12:00 PM	SU	44	55
ACCT 102	Intro Theo & Prac Acct 2	David, Amy	5:00 PM - 6:40 PM	T, TH	51	55
ACCT 102	Intro Theo & Prac Acct 2	Erlach, David	1:40 PM - 3:20 PM	M, W	55	55
ACCT 102	Intro Theo & Prac Acct 2	Feisullin, Anita	10:05 AM - 11:55 AM	T, TH	55	55
ACCT 102	Intro Theo & Prac Acct 2	Gruza, Marvin	10:05 AM - 11:55 AM	W, F	16	55
ACCT 102	Intro Theo & Prac Acct 2	Kessar, Nathaniel	6:55 PM - 8:45 PM	T, TH	17	55
ACCT 102	Intro Theo & Prac Acct 2	Khanna, Himanshu	8:20 AM - 12:00 PM	S	41	55
ACCT 102	Intro Theo & Prac Acct 2	Sun, Fang	6:55 PM - 8:45 PM	M, W	19	30
AFST 201	Intro Black Cultures	Julmisse, Evelyn	1:40 PM - 2:55 PM	T, TH	26	25
AMST 110	Intro Amer Soc/Culture	Manson, Douglas	12:15 PM - 1:30 PM	T, TH	24	25
AMST 110	Intro Amer Soc/Culture	Manson, Douglas	9:15 AM - 10:30 AM	T, TH	24	25
AMST 110	Intro Amer Soc/Culture	Schnur, Kate	9:15 AM - 10:30 AM	M, W	25	25
ANTH 101	Intro To Cultural Anthropology	Barton, Scott	1:40 PM - 2:55 PM	M, W	24	25
ANTH 101	Intro To Cultural Anthropology	Choi, Jimee	12:15 PM - 1:30 PM	T, TH	39	40
ANTH 101	Intro To Cultural Anthropology	Choi, Jimee	7:45 AM - 9:00 AM	T, TH	25	25
ANTH 101	Intro To Cultural Anthropology	Elisha, Omri	10:45 AM - 12:00 PM	T, TH	150	145
ANTH 101	Intro To Cultural Anthropology	Galal, Ola	9:15 AM - 10:30 AM	M, W	29	29
ANTH 101	Intro To Cultural Anthropology	Levin, Irina	1:40 PM - 2:55 PM	T, TH	89	88
ANTH 101	Intro To Cultural Anthropology	Levin, Irina	3:10 PM - 4:25 PM	T, TH	65	65
ANTH 101	Intro To Cultural Anthropology	Leynse, Wendy	10:45 AM - 12:00 PM	M, W	130	130
ANTH 101	Intro To Cultural Anthropology	Spice, Anne	3:10 PM - 4:25 PM	M, W	69	70
ANTH 102	Intro Human Evol	Evans, Katarina	10:45 AM - 12:00 PM	M, W	39	38
ANTH 102	Intro Human Evol	Forrest, Frances	12:15 PM - 1:30 PM	T, TH	150	145
ANTH 102	Intro Human Evol	Lamela Lopez, Raquel	7:45 AM - 9:00 AM	M, W	74	75
ANTH 102	Intro Human Evol	Lamela Lopez, Raquel	9:15 AM - 10:30 AM	M, W	85	85
ANTH 102	Intro Human Evol	Madimenos, Felicia	1:40 PM - 2:55 PM	M, W	82	82
ANTH 102	Intro Human Evol	Pagano, Anthony	6:30 PM - 9:15 PM	F	40	40
ANTH 102	Intro Human Evol	Pagano, Anthony	9:15 AM - 12:00 PM	S	41	40
ANTH 102	Intro Human Evol	Plummer, Thomas	10:45 AM - 12:00 PM	T, TH	91	90
ANTH 102	Intro Human Evol	Shibata, Chihiro	3:10 PM - 4:25 PM	T, TH	40	40
ANTH 102	Intro Human Evol	Shibata, Chihiro	5:00 PM - 6:15 PM	T, TH	41	40

Note: Code is provided in the notes tab on this slide

FreeQuery.CSCI_Search:



SQLtoJDBC.exe by Erik Kim

Server: localhost

Port: 12001

Database: QueensClassScheduleSpring2017

User: sa

Password: PM0123456789

Input:

```
USE QueensClassScheduleSpring2017;
SELECT DISTINCT
        [CourseName],
        [InstructorFullName],
        [Hours],
        [Day],
        [NumberEnrolled],
        [Limit]
        FROM Course.Class where [CourseName] LIKE 'CSCI%' AND [NumberEnrolled] < [Limit]
```

Output:

CourseName	InstructorFullName	Hours	Day	NumberEnrolled	Limit
CSCI 012	Gelzhals, Sara	1:40 PM - 2:30 PM	M, W	24	25
CSCI 012	Gelzhals, Sara	2:45 PM - 3:35 PM	M, W	19	20
CSCI 012	Greenberg, Aryeh	7:30 PM - 8:20 PM	M, W	23	24
CSCI 012	Greenberg, Aryeh	8:30 PM - 9:20 PM	M, W	23	24
CSCI 012	Huang, Xiu	8:00 AM - 8:50 AM	T, F	19	20
CSCI 012	Lee, Kok Teng	1:00 PM - 2:50 PM	S	19	20
CSCI 012	Lee, Kok Teng	3:00 PM - 4:50 PM	S	19	20
CSCI 012	Nixon, Dorothy	1:00 PM - 2:50 PM	SU	19	20
CSCI 012	Nixon, Dorothy	3:00 PM - 4:50 PM	SU	19	20
CSCI 012	Paciorek, Annette	9:00 AM - 9:50 AM	T, TH	23	24
CSCI 012	Sakai, Christian	8:00 AM - 8:50 AM	M, W	14	20
CSCI 012	Waxman, Jerry	11:10 AM - 12:00 PM	M, W	143	144
CSCI 012	Waxman, Jerry	3:50 PM - 4:40 PM	M, W	100	109
CSCI 048	Goldberg, Robert	5:30 PM - 6:20 PM	M, W	30	31
CSCI 048	Goldberg, Robert	6:30 PM - 7:20 PM	M, W	57	62
CSCI 048	Goldberg, Robert	7:30 PM - 8:20 PM	M, W	27	31
CSCI 087	Phillips, Tsaiyun	5:00 PM - 5:50 PM	T, TH	10	20
CSCI 087	Phillips, Tsaiyun	6:00 PM - 6:50 PM	T, TH	10	20
CSCI 090	Frary, Barbara	1:40 PM - 2:55 PM	M, W	18	30
CSCI 090	Smith-Thompson, Anne	8:00 AM - 8:50 AM	T, F	12	20
CSCI 090	Smith-Thompson, Anne	9:00 AM - 9:50 AM	F	12	20
CSCI 111	Mahavadi, Krishna	11:10 AM - 12:00 PM	T, TH	23	24
CSCI 111	Ryba, Alexander	2:45 PM - 3:35 PM	M, W	152	153
CSCI 211	Lee, Kok Teng	3:50 PM - 4:40 PM	T, TH	26	28
CSCI 211	Li, Changlin	3:50 PM - 4:40 PM	M, W	26	28
CSCI 211	Waxman, Jerry	12:15 PM - 2:05 PM	F	102	110
CSCI 212	Bian, Kaikai	8:00 AM - 8:50 AM	T, TH	24	25
CSCI 212	Chen, Cong	7:30 PM - 8:20 PM	M, W	14	22
CSCI 212	Lee, Kok Teng	6:30 PM - 7:20 PM	T, TH	36	48
CSCI 212	Li, Changlin	3:50 PM - 4:40 PM	T, TH	22	23
CSCI 212	Lord, Kenneth	10:05 AM - 10:55 AM	M, W	163	164
CSCI 212	Mahavadi, Krishna	9:00 AM - 9:50 AM	M, W	76	77
CSCI 220	Goswami, Mayank	3:10 PM - 4:25 PM	M, W	41	42
CSCI 220	Kong, Taiyung	8:00 PM - 9:15 PM	M, W	38	42
CSCI 240	Abreu, Andy	8:00 PM - 9:15 PM	M, W	31	32

Note: Code is provided in the notes tab on this slide