



Group Project #2

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

To-Do List

To-do list

To be completed by: 11/20/2020

Deadline: 11/20/2020

Name: Jamil Kocacal

Date: 11/5/2020

Project 2

| % done | Phase | Start By | Original Due By | Revised Due By | Number Of Days | Revision Notes |
|--------|--|------------|-----------------|----------------|----------------|---|
| 100% | Planning | 11/5/2020 | 11/5/2020 | | One | |
| 100% | Create tasks within the to-do list | 11/5/2020 | 11/19/2020 | | Fifteen | |
| 100% | Setup bak file in ssms | 11/5/2020 | 11/7/2020 | | Two | |
| 100% | Prepare questions for professor Heller | 11/7/2020 | 11/10/2020 | | Three | |
| 100% | Create dimproduct and data stored procedure | 11/10/2020 | 11/15/2020 | 16-Nov-20 | Five | Extended due to misunderstanding on load data procedure |
| 100% | Create power point slides | 11/15/2020 | 11/16/2020 | | One | |
| 100% | Power point recordings about To-do list and procedures | 11/16/2020 | 11/19/2020 | | Three | |
| 100% | JDBC Recording | 11/16/2020 | 11/19/2020 | | Three | |
| 0% | | | | | | |
| | | | | | | |

















Project Planner

Project 2 Group-4 Planner

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1

 Plan Duration
  Actual Start
  % Complete
  Actual (beyond plan)
  % Complete (beyond plan)

| ACTIVITY | PLAN START | PLAN DURATION | ACTUAL START | Actual Duration | PERCENT COMPLETE | DAYS |
|--|------------|---------------|--------------|-----------------|------------------|---|
| | | | | | | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 3 |
| *Assign tasks and roles to each member | 1 | 1 | 1 | 1 | 100% |  |
| *Gather all files and resources needed | 1 | 1 | 1 | 1 | 100% |  |
| *Create needed tables. | 2 | 3 | 2 | 2 | 100% |  |
| *Ask Questions for better understanding. | 3 | 1 | 3 | 3 | 100% |  |
| *Distribute procedures to all group members. | 5 | 1 | 5 | 1 | 100% |  |
| *Each member complete their stored procedures. | 5 | 2 | 5 | 4 | 100% |  |
| *Update stored procedures to meet project specs. | 9 | 2 | 9 | 2 | 100% |  |
| *Test to assure all stored procedures work. | 10 | 2 | 10 | 3 | 100% |  |
| *Discuss the JDBC implementation | 11 | 1 | 11 | 1 | 100% |  |
| *Complete the JDBC portion | 11 | 4 | 11 | 4 | 100% |  |
| *Discuss the layout and outline of the PowerPoints | 13 | 1 | 13 | 1 | 100% |  |
| *Complete SQLDoc portion to add into presentation | 13 | 1 | 13 | 1 | 100% |  |
| *Create powerpoints and have all members add in their slides | 13 | 3 | 13 | 3 | 100% |  |
| *Add audio to slides and complete both Power Points | 16 | 1 | 16 | 1 | 100% |  |

Project Planner



Meeting Notes

CS331 10:45 Group 4 Project 2 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: October 31st, 2020

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

Absences: None

Team Roles:

Group leader: Erik Kim

Agenda/meeting notes taker: Harjit Liyal

To-Do List: Jamil Kocacal

Project Planner: Marlon Louis

PowerPoint: Haibo Liu & Danny Kong

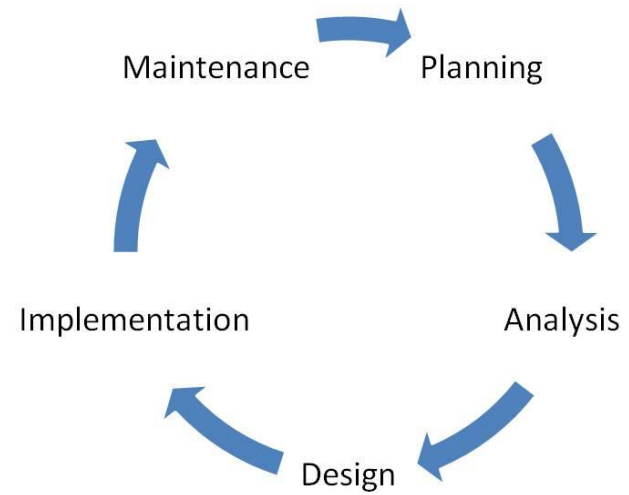
Video editing: Jonathan Eng

Link: https://docs.google.com/document/d/1uny_hR6ldEE1km_OOq34-6r-FzU_v34CY6YpWvbz5UQ/edit?usp=sharing

Note: Entire word document and PDF are also included in the project

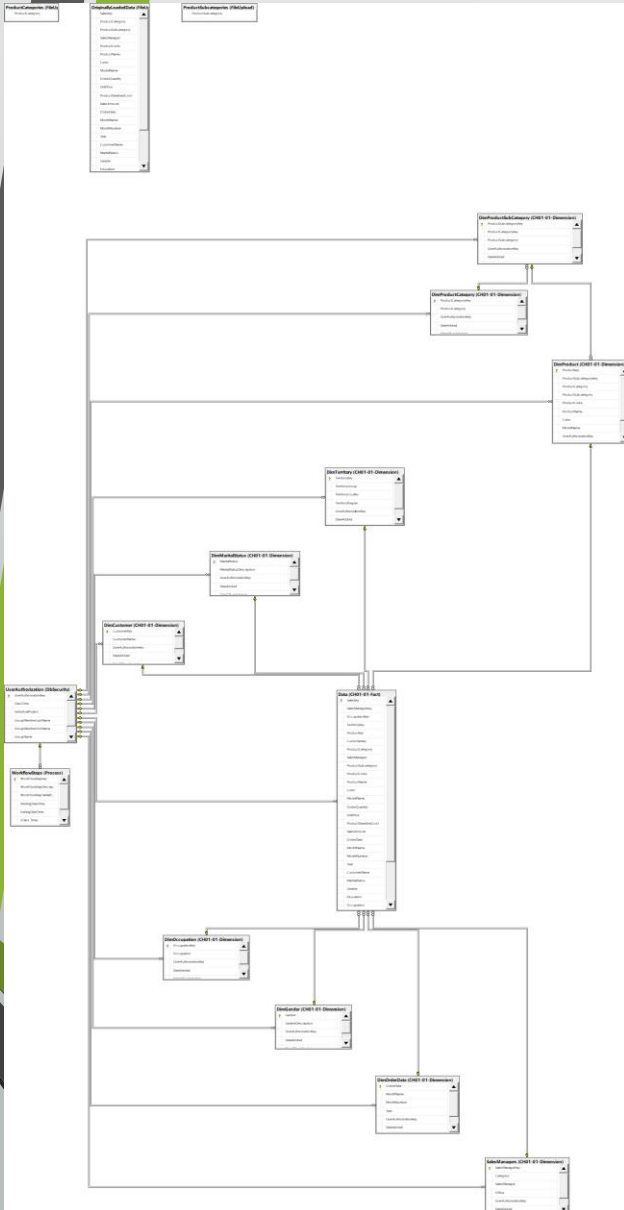


SSMS Lifecycle



Planning

- Read the project specifications
- Drafted up our project management items (To Do List, Project Planner, Meeting Notes)
- Discussed and drafted a plan of action, divided work amongst the project members
- Had the project leader create the foundation so that other members could integrate their work seamlessly (recreating tables, stored procedures to be used in other stored procedures, etc.)
- Created a template as a group to create individual stored procedures efficiently
- Discussed project specifications after each meeting to address questions that would be asked in class later



Analysis

- Looked over the project specifications and made sure our code followed the set guidelines.
- Made questions as a group and asked Professor Heller and each other for clarification.
- Created and ran the JDBC to meet the project guidelines.



Design

- Developed the individual stored procedures
- Set up the variables, parameters, and attributes needed for data insertion
- Creating a view of newly inputted data to ensure procedure worked as expected



Implementation

- Tested the created Procedures in SSMS by executing each individually
- Used Truncate to restart entries for re-testing when required
- Observed the output of the program, taking note of if the output has the same entries as the table we wish to copy
- Observed the newly created View, to see if the new information was saved with the correct entries
- Observed the WorkFlowSteps table to ensure the tested outputs are being documented, allowing for easily obtainable and information reference points if needed



Maintenance

- Fixed several times on execute each procedure
- Double check UserAuthorizationKey and every parameters' name are same from OriginallyLoadedData table.
- Deleted truncate/exec checks in stored procedured after end
- Checked the view table does work (change the select at the bottom to actual view)





Stored Procedures

Erik's Stored Procedures

- Process.usp_ShowWorkflowSteps
- Process.usp_TrackWorkflow
- Project2.AddForeignKeysToStarSchemaData
- Project2.DropForeignKeysFromStarSchemaData
- Project2.TruncateStarSchemaData
- Project2.LoadStarSchema
- Utils.DropProcsInCSCI331FinalProject



Process.usp_ShowWorkflowSteps

```
1 USE [BIClass]
2 GO
3 /***** Object: StoredProcedure [Process].[usp_ShowWorkflowSteps]    Script Date: 11/17/2020 5:50:32 PM *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8 --=====
9 -- Author:      Erik Kim
10 -- Create date: 11/17/2020
11 -- Description: Show workflow steps
12 --=====
13 ALTER PROCEDURE [Process].[usp_ShowWorkflowSteps]
14 AS
15 BEGIN
16     -- SET NOCOUNT ON added to prevent extra result sets from
17     -- interfering with SELECT statements.
18     SET NOCOUNT ON;
19     SELECT *
20     FROM [Process].[WorkflowSteps];
21 END
22
```



Process.usp_TrackWorkFlow

```
1 USE [BIClass]
2 GO
3 /***** Object: StoredProcedure [Process].[usp_TrackWorkFlow]    Script Date: 11/17/2020 5:50:37 PM *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8 -- *****
9 -- Author:      Erik Kim
10 -- Create date: 11/13/2020
11 -- Description: Track work flow
12 -- *****
13 ALTER PROCEDURE [Process].[usp_TrackWorkFlow]
14 -- Add the parameters for the stored procedure here
15     @WorkflowDescription NVARCHAR(100),
16     @WorkflowStepTableRowCount INT,
17     @StartingDateTime DATETIME2,
18     @EndingDateTime DATETIME2,
19     @UserAuthorizationKey INT
20 AS
21 BEGIN
22     -- SET NOCOUNT ON added to prevent extra result sets from
23     -- interfering with SELECT statements.
24     SET NOCOUNT ON;
25
26     -- Insert statements for procedure here
27     INSERT INTO [Process].[WorkflowSteps]
28     (
29         WorkflowStepDescription,
30         WorkflowStepTableRowCount,
31         StartingDateTime,
32         EndingDateTime,
33         [Class Time],
34         UserAuthorizationKey
35     )
36     VALUES
37     (@WorkflowDescription, @WorkflowStepTableRowCount, @StartingDateTime, @EndingDateTime, '10:45',
38     @UserAuthorizationKey);
39
40 END;
41
```



Project2.AddForeignKeysToStarSchemaData

```
1 1 [TSQL]
2 -- Project: StarSchema (Project) [StarSchema]
3 -- Date: 11/11/2023
4 -- Author: [User]
5 -- Description: Add foreign keys to the Star Schema
6
7 -- Create the Star Schema
8
9 -- Create the Fact Table
10 CREATE TABLE FactSales
11 (
12     SalesID INT PRIMARY KEY,
13     ProductID INT,
14     CustomerID INT,
15     SalesDate DATE,
16     SalesAmount DECIMAL(10,2)
17 )
18
19 -- Create the Dimension Tables
20
21 -- Create the Product Dimension
22 CREATE TABLE ProductDim
23 (
24     ProductID INT PRIMARY KEY,
25     ProductName VARCHAR(100),
26     ProductCategory VARCHAR(50)
27 )
28
29 -- Create the Customer Dimension
30 CREATE TABLE CustomerDim
31 (
32     CustomerID INT PRIMARY KEY,
33     CustomerName VARCHAR(100),
34     CustomerAddress VARCHAR(200)
35 )
36
37 -- Create the Sales Date Dimension
38 CREATE TABLE SalesDateDim
39 (
40     SalesDate DATE PRIMARY KEY,
41     SalesDateLabel VARCHAR(100)
42 )
43
44 -- Add foreign keys to the Fact Table
45
46 -- Add foreign key to ProductDim
47 ALTER TABLE FactSales
48 ADD CONSTRAINT FK_FactSales_ProductDim
49 FOREIGN KEY (ProductID)
50 REFERENCES ProductDim (ProductID)
51
52 -- Add foreign key to CustomerDim
53 ALTER TABLE FactSales
54 ADD CONSTRAINT FK_FactSales_CustomerDim
55 FOREIGN KEY (CustomerID)
56 REFERENCES CustomerDim (CustomerID)
57
58 -- Add foreign key to SalesDateDim
59 ALTER TABLE FactSales
60 ADD CONSTRAINT FK_FactSales_SalesDateDim
61 FOREIGN KEY (SalesDate)
62 REFERENCES SalesDateDim (SalesDate)
63
64 -- Create the FactSalesHistory Table
65 CREATE TABLE FactSalesHistory
66 (
67     SalesID INT PRIMARY KEY,
68     ProductID INT,
69     CustomerID INT,
70     SalesDate DATE,
71     SalesAmount DECIMAL(10,2)
72 )
73
74 -- Add foreign keys to the FactSalesHistory Table
75
76 -- Add foreign key to ProductDim
77 ALTER TABLE FactSalesHistory
78 ADD CONSTRAINT FK_FactSalesHistory_ProductDim
79 FOREIGN KEY (ProductID)
80 REFERENCES ProductDim (ProductID)
81
82 -- Add foreign key to CustomerDim
83 ALTER TABLE FactSalesHistory
84 ADD CONSTRAINT FK_FactSalesHistory_CustomerDim
85 FOREIGN KEY (CustomerID)
86 REFERENCES CustomerDim (CustomerID)
87
88 -- Add foreign key to SalesDateDim
89 ALTER TABLE FactSalesHistory
90 ADD CONSTRAINT FK_FactSalesHistory_SalesDateDim
91 FOREIGN KEY (SalesDate)
92 REFERENCES SalesDateDim (SalesDate)
93
94 -- Create the FactSalesSummary Table
95 CREATE TABLE FactSalesSummary
96 (
97     SalesDate DATE PRIMARY KEY,
98     ProductID INT,
99     SalesAmount DECIMAL(10,2)
100 )
101
102 -- Add foreign keys to the FactSalesSummary Table
103
104 -- Add foreign key to SalesDateDim
105 ALTER TABLE FactSalesSummary
106 ADD CONSTRAINT FK_FactSalesSummary_SalesDateDim
107 FOREIGN KEY (SalesDate)
108 REFERENCES SalesDateDim (SalesDate)
109
110 -- Add foreign key to ProductDim
111 ALTER TABLE FactSalesSummary
112 ADD CONSTRAINT FK_FactSalesSummary_ProductDim
113 FOREIGN KEY (ProductID)
114 REFERENCES ProductDim (ProductID)
115
116 -- Create the FactSalesSummaryHistory Table
117 CREATE TABLE FactSalesSummaryHistory
118 (
119     SalesDate DATE PRIMARY KEY,
120     ProductID INT,
121     SalesAmount DECIMAL(10,2)
122 )
123
124 -- Add foreign keys to the FactSalesSummaryHistory Table
125
126 -- Add foreign key to SalesDateDim
127 ALTER TABLE FactSalesSummaryHistory
128 ADD CONSTRAINT FK_FactSalesSummaryHistory_SalesDateDim
129 FOREIGN KEY (SalesDate)
130 REFERENCES SalesDateDim (SalesDate)
131
132 -- Add foreign key to ProductDim
133 ALTER TABLE FactSalesSummaryHistory
134 ADD CONSTRAINT FK_FactSalesSummaryHistory_ProductDim
135 FOREIGN KEY (ProductID)
136 REFERENCES ProductDim (ProductID)
137
138 -- Create the FactSalesSummarySummary Table
139 CREATE TABLE FactSalesSummarySummary
140 (
141     SalesDate DATE PRIMARY KEY,
142     SalesAmount DECIMAL(10,2)
143 )
144
145 -- Add foreign keys to the FactSalesSummarySummary Table
146
147 -- Add foreign key to SalesDateDim
148 ALTER TABLE FactSalesSummarySummary
149 ADD CONSTRAINT FK_FactSalesSummarySummary_SalesDateDim
150 FOREIGN KEY (SalesDate)
151 REFERENCES SalesDateDim (SalesDate)
152
153 -- Add foreign key to ProductDim
154 ALTER TABLE FactSalesSummarySummary
155 ADD CONSTRAINT FK_FactSalesSummarySummary_ProductDim
156 FOREIGN KEY (ProductID)
157 REFERENCES ProductDim (ProductID)
158
159 -- Create the FactSalesSummarySummaryHistory Table
160 CREATE TABLE FactSalesSummarySummaryHistory
161 (
162     SalesDate DATE PRIMARY KEY,
163     SalesAmount DECIMAL(10,2)
164 )
165
166 -- Add foreign keys to the FactSalesSummarySummaryHistory Table
167
168 -- Add foreign key to SalesDateDim
169 ALTER TABLE FactSalesSummarySummaryHistory
170 ADD CONSTRAINT FK_FactSalesSummarySummaryHistory_SalesDateDim
171 FOREIGN KEY (SalesDate)
172 REFERENCES SalesDateDim (SalesDate)
173
174 -- Add foreign key to ProductDim
175 ALTER TABLE FactSalesSummarySummaryHistory
176 ADD CONSTRAINT FK_FactSalesSummarySummaryHistory_ProductDim
177 FOREIGN KEY (ProductID)
178 REFERENCES ProductDim (ProductID)
179
180 -- Create the FactSalesSummarySummarySummary Table
181 CREATE TABLE FactSalesSummarySummarySummary
182 (
183     SalesDate DATE PRIMARY KEY,
184     SalesAmount DECIMAL(10,2)
185 )
186
187 -- Add foreign keys to the FactSalesSummarySummarySummary Table
188
189 -- Add foreign key to SalesDateDim
190 ALTER TABLE FactSalesSummarySummarySummary
191 ADD CONSTRAINT FK_FactSalesSummarySummarySummary_SalesDateDim
192 FOREIGN KEY (SalesDate)
193 REFERENCES SalesDateDim (SalesDate)
194
195 -- Add foreign key to ProductDim
196 ALTER TABLE FactSalesSummarySummarySummary
197 ADD CONSTRAINT FK_FactSalesSummarySummarySummary_ProductDim
198 FOREIGN KEY (ProductID)
199 REFERENCES ProductDim (ProductID)
200
201 -- Create the FactSalesSummarySummarySummaryHistory Table
202 CREATE TABLE FactSalesSummarySummarySummaryHistory
203 (
204     SalesDate DATE PRIMARY KEY,
205     SalesAmount DECIMAL(10,2)
206 )
207
208 -- Add foreign keys to the FactSalesSummarySummarySummaryHistory Table
209
210 -- Add foreign key to SalesDateDim
211 ALTER TABLE FactSalesSummarySummarySummaryHistory
212 ADD CONSTRAINT FK_FactSalesSummarySummarySummaryHistory_SalesDateDim
213 FOREIGN KEY (SalesDate)
214 REFERENCES SalesDateDim (SalesDate)
215
216 -- Add foreign key to ProductDim
217 ALTER TABLE FactSalesSummarySummarySummaryHistory
218 ADD CONSTRAINT FK_FactSalesSummarySummarySummaryHistory_ProductDim
219 FOREIGN KEY (ProductID)
220 REFERENCES ProductDim (ProductID)
221
222 -- Create the FactSalesSummarySummarySummarySummary Table
223 CREATE TABLE FactSalesSummarySummarySummarySummary
224 (
225     SalesDate DATE PRIMARY KEY,
226     SalesAmount DECIMAL(10,2)
227 )
228
229 -- Add foreign keys to the FactSalesSummarySummarySummarySummary Table
230
231 -- Add foreign key to SalesDateDim
232 ALTER TABLE FactSalesSummarySummarySummarySummary
233 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummary_SalesDateDim
234 FOREIGN KEY (SalesDate)
235 REFERENCES SalesDateDim (SalesDate)
236
237 -- Add foreign key to ProductDim
238 ALTER TABLE FactSalesSummarySummarySummarySummary
239 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummary_ProductDim
240 FOREIGN KEY (ProductID)
241 REFERENCES ProductDim (ProductID)
242
243 -- Create the FactSalesSummarySummarySummarySummaryHistory Table
244 CREATE TABLE FactSalesSummarySummarySummarySummaryHistory
245 (
246     SalesDate DATE PRIMARY KEY,
247     SalesAmount DECIMAL(10,2)
248 )
249
250 -- Add foreign keys to the FactSalesSummarySummarySummarySummaryHistory Table
251
252 -- Add foreign key to SalesDateDim
253 ALTER TABLE FactSalesSummarySummarySummarySummaryHistory
254 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummaryHistory_SalesDateDim
255 FOREIGN KEY (SalesDate)
256 REFERENCES SalesDateDim (SalesDate)
257
258 -- Add foreign key to ProductDim
259 ALTER TABLE FactSalesSummarySummarySummarySummaryHistory
260 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummaryHistory_ProductDim
261 FOREIGN KEY (ProductID)
262 REFERENCES ProductDim (ProductID)
263
264 -- Create the FactSalesSummarySummarySummarySummarySummary Table
265 CREATE TABLE FactSalesSummarySummarySummarySummarySummary
266 (
267     SalesDate DATE PRIMARY KEY,
268     SalesAmount DECIMAL(10,2)
269 )
270
271 -- Add foreign keys to the FactSalesSummarySummarySummarySummarySummary Table
272
273 -- Add foreign key to SalesDateDim
274 ALTER TABLE FactSalesSummarySummarySummarySummarySummary
275 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummarySummary_SalesDateDim
276 FOREIGN KEY (SalesDate)
277 REFERENCES SalesDateDim (SalesDate)
278
279 -- Add foreign key to ProductDim
280 ALTER TABLE FactSalesSummarySummarySummarySummarySummary
281 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummarySummary_ProductDim
282 FOREIGN KEY (ProductID)
283 REFERENCES ProductDim (ProductID)
284
285 -- Create the FactSalesSummarySummarySummarySummarySummaryHistory Table
286 CREATE TABLE FactSalesSummarySummarySummarySummarySummaryHistory
287 (
288     SalesDate DATE PRIMARY KEY,
289     SalesAmount DECIMAL(10,2)
290 )
291
292 -- Add foreign keys to the FactSalesSummarySummarySummarySummarySummaryHistory Table
293
294 -- Add foreign key to SalesDateDim
295 ALTER TABLE FactSalesSummarySummarySummarySummarySummaryHistory
296 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummarySummaryHistory_SalesDateDim
297 FOREIGN KEY (SalesDate)
298 REFERENCES SalesDateDim (SalesDate)
299
300 -- Add foreign key to ProductDim
301 ALTER TABLE FactSalesSummarySummarySummarySummarySummaryHistory
302 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummarySummaryHistory_ProductDim
303 FOREIGN KEY (ProductID)
304 REFERENCES ProductDim (ProductID)
305
306 -- Create the FactSalesSummarySummarySummarySummarySummarySummary Table
307 CREATE TABLE FactSalesSummarySummarySummarySummarySummarySummary
308 (
309     SalesDate DATE PRIMARY KEY,
310     SalesAmount DECIMAL(10,2)
311 )
312
313 -- Add foreign keys to the FactSalesSummarySummarySummarySummarySummarySummary Table
314
315 -- Add foreign key to SalesDateDim
316 ALTER TABLE FactSalesSummarySummarySummarySummarySummarySummary
317 ADD CONSTRAINT FK_FactSalesSummarySummarySummarySummarySummarySummary_SalesDateDim
318 FOREIGN KEY (SalesDate)
319 REFERENCES SalesDateDim (SalesDate)
320
318
```



Project2.DropForeignKeysFromStarSchemaData

```
1 USE [BIClass]
2 GO
3 /***** Object: StoredProcedure [Project2].[DropForeignKeysFromStarSchemaData]    Script Date: 11/17/2020 5:14:48 AM *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8
9 -- Author:      Erik Kim
10 -- Create date: 11/17/2020
11 -- Description: Drop the Foreign Keys from the Star Schema
12 --
13
14 ALTER PROCEDURE [Project2].[DropForeignKeysFromStarSchemaData] @UserAuthorizationKey INT
15 AS
16 BEGIN
17     SET NOCOUNT ON;
18     DECLARE @StartingDateTime DATETIME2 = SYSUTETIME();
19
20     ALTER TABLE [CWS-01-Fact].[Data] DROP CONSTRAINT FK_Data_DisCustomer;
21     ALTER TABLE [CWS-01-Fact].[Data] DROP CONSTRAINT FK_Data_DisVendor;
22     DROP CONSTRAINT FK_Data_DisMaritalStatus;
23     ALTER TABLE [CWS-01-Fact].[Data] DROP CONSTRAINT FK_Data_DisOccupation;
24     ALTER TABLE [CWS-01-Fact].[Data] DROP CONSTRAINT FK_Data_DisOrderDate;
25     ALTER TABLE [CWS-01-Fact].[Data] DROP CONSTRAINT FK_Data_DisProduct;
26     ALTER TABLE [CWS-01-Fact].[Data] DROP CONSTRAINT FK_Data_DisTerritory;
27     ALTER TABLE [CWS-01-Fact].[Data] DROP CONSTRAINT FK_Data_SalesManagers;
28     ALTER TABLE [CWS-01-Dimension].[DisProduct]
29     DROP CONSTRAINT FK_DisProduct_DisProductSubCategory;
30     ALTER TABLE [CWS-01-Dimension].[DisProductSubCategory]
31     DROP CONSTRAINT FK_DisProductSubCategory_DisProductCategory;
32     ALTER TABLE [CWS-01-Dimension].[DisCustomer]
33     DROP CONSTRAINT FK_DisCustomer_UserAuthorization;
34     ALTER TABLE [CWS-01-Dimension].[DisVendor]
35     DROP CONSTRAINT FK_DisVendor_UserAuthorization;
36     ALTER TABLE [CWS-01-Dimension].[DisMaritalStatus]
37     DROP CONSTRAINT FK_DisMaritalStatus_UserAuthorization;
38     ALTER TABLE [CWS-01-Dimension].[DisOccupation]
39     DROP CONSTRAINT FK_DisOccupation_UserAuthorization;
40     ALTER TABLE [CWS-01-Dimension].[DisOrderDate]
41     DROP CONSTRAINT FK_DisOrderDate_UserAuthorization;
42     ALTER TABLE [CWS-01-Dimension].[DisProduct]
43     DROP CONSTRAINT FK_DisProduct_UserAuthorization;
44     ALTER TABLE [CWS-01-Dimension].[DisProductCategory]
45     DROP CONSTRAINT FK_DisProductCategory_UserAuthorization;
46     ALTER TABLE [CWS-01-Dimension].[DisProductSubCategory]
47     DROP CONSTRAINT FK_DisProductSubCategory_UserAuthorization;
48     ALTER TABLE [CWS-01-Dimension].[DisTerritory]
49     DROP CONSTRAINT FK_DisTerritory_UserAuthorization;
50     ALTER TABLE [CWS-01-Dimension].[SalesManagers]
51     DROP CONSTRAINT FK_SalesManagers_UserAuthorization;
52     ALTER TABLE [CWS-01-Fact].[Data] DROP CONSTRAINT FK_Data_UserAuthorization;
53     ALTER TABLE [Process].[WorkFlowSteps]
54     DROP CONSTRAINT FK_WorkFlowSteps_UserAuthorization;
55
56     DECLARE @WorkFlowStepsTableCount INT;
57     SET @WorkFlowStepsTableCount = 0;
58     DECLARE @EndingDateTime DATETIME2 = SYSUTETIME();
59     EXEC [Process].[Log_TrackWorkFlow] 'Drop Foreign Keys',
60         @WorkFlowStepsTableCount,
61         @StartingDateTime,
62         @EndingDateTime,
63         @UserAuthorizationKey;
64 END;
```



Project2.TruncateStarSchemaData

```
1 USE [BIClass]
2 GO
3 /***** Object: StoredProcedure [Project2].[TruncateStarSchemaData]    Script Date: 11/17/2020 5:14:56 AM *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8 --
9 -- Author:      Erik Kim
10 -- Create date: 11/13/2020
11 -- Description: Truncate StarSchemaData
12 --
13 ALTER PROCEDURE [Project2].[TruncateStarSchemaData]
14     @UserAuthorizationKey int
15 AS
16 BEGIN
17     -- SET NOCOUNT ON added to prevent extra result sets from
18     -- interfering with SELECT statements.
19     SET NOCOUNT ON;
20     DECLARE @StartingDateTime DATETIME2 = SYSDATETIME();
21
22     TRUNCATE TABLE [CH01-01-Dimension].DimCustomer;
23     ALTER SEQUENCE PkSequence.DimCustomerSequenceObject RESTART WITH 1;
24     TRUNCATE TABLE [CH01-01-Dimension].DimGender;
25     TRUNCATE TABLE [CH01-01-Dimension].DimMaritalStatus;
26     TRUNCATE TABLE [CH01-01-Dimension].DimOccupation;
27     ALTER SEQUENCE PkSequence.OccupationSequenceObject RESTART WITH 1;
28     TRUNCATE TABLE [CH01-01-Dimension].DimOrderDate;
29     TRUNCATE TABLE [CH01-01-Dimension].DimProduct;
30     ALTER SEQUENCE PkSequence.DimProductSequenceObject RESTART WITH 1;
31     TRUNCATE TABLE [CH01-01-Dimension].DimProductCategory;
32     ALTER SEQUENCE PkSequence.DimProductCategorySequenceObject RESTART WITH 1;
33     TRUNCATE TABLE [CH01-01-Dimension].DimProductSubCategory;
34     ALTER SEQUENCE PkSequence.DimProductSubCategorySequenceObject RESTART WITH 1;
35     TRUNCATE TABLE [CH01-01-Dimension].DimTerritory;
36     ALTER SEQUENCE PkSequence.DimTerritorySequenceObject RESTART WITH 1;
37     TRUNCATE TABLE [CH01-01-Dimension].SalesManagers;
38     ALTER SEQUENCE PkSequence.SalesManagersSequenceObject RESTART WITH 1;
39     TRUNCATE TABLE [CH01-01-Fact].Data;
40     ALTER SEQUENCE PkSequence.DataSequenceObject RESTART WITH 1;
41
42     DECLARE @WorkflowStepTableRowCount INT;
43     SET @WorkflowStepTableRowCount = 0;
44     DECLARE @EndingDateTime DATETIME2 = SYSDATETIME();
45     EXEC ([Process1].Usp_TrackWorkflow) @WorkflowStepTableRowCount,
46     @StartingDateTime,
47     @EndingDateTime,
48     @UserAuthorizationKey;
49
50 end
51
52
```



Project2.LoadStarSchema

```
1 USE [BIClass]
2 GO
3 /***** Object: StoredProcedure [Project2].[LoadStarSchemaData]    Script Date: 11/17/2020 5:15:04 AM *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8 --=====
9 -- Author:      YourName
10 -- Create date:
11 -- Description:
12 --=====
13 ALTER PROCEDURE [Project2].[LoadStarSchemaData] @UserAuthorizationKey INT
14 AS
15 BEGIN
16     SET NOCOUNT ON;
17     DECLARE @StartingDateTime DATETIME2 = SYSDATETIME();
18
19     --
20     -- Drop All of the foreign keys prior to truncating tables in the star schema
21     --
22     EXEC [Project2].[DropForeignKeysFromStarSchemaData] @UserAuthorizationKey = 1;
23     --
24     -- Check row count before truncation
25     EXEC [Project2].[ShowTableStatusRowCount]
26         @UserAuthorizationKey = 2, -- Change -1 to the appropriate UserAuthorizationKey
27         @TableStatus = N''Pre-truncate of tables''
28     --
29     -- Always truncate the Star Schema Data
30     --
31     EXEC [Project2].[TruncateStarSchemaData] @UserAuthorizationKey = 1;
32     --
33     -- Load the star schema
34     --
35     EXEC [Project2].[Load_DimProductCategory] @UserAuthorizationKey = 4; -- Change -1 to the appropriate UserAuthorizationKey
36     EXEC [Project2].[Load_DimProductSubcategory] @UserAuthorizationKey = 4; -- Change -1 to the appropriate UserAuthorizationKey
37     EXEC [Project2].[Load_DimProduct] @UserAuthorizationKey = 4; -- Change -1 to the appropriate UserAuthorizationKey
38     EXEC [Project2].[Load_SalesManagers] @UserAuthorizationKey = 3; -- Change -1 to the appropriate UserAuthorizationKey
39     EXEC [Project2].[Load_DimGender] @UserAuthorizationKey = 5; -- Change -1 to the appropriate UserAuthorizationKey
40     EXEC [Project2].[Load_DimMaritalStatus] @UserAuthorizationKey = 2; -- Change -1 to the appropriate UserAuthorizationKey
41     EXEC [Project2].[Load_DimOccupation] @UserAuthorizationKey = 7; -- Change -1 to the appropriate UserAuthorizationKey
42     EXEC [Project2].[Load_DimUnderlying] @UserAuthorizationKey = 7; -- Change -1 to the appropriate UserAuthorizationKey
43     EXEC [Project2].[Load_DimTerritory] @UserAuthorizationKey = 3; -- Change -1 to the appropriate UserAuthorizationKey
44     EXEC [Project2].[Load_DimCustomer] @UserAuthorizationKey = 3; -- Change -1 to the appropriate UserAuthorizationKey
45     EXEC [Project2].[Load_Fact] @UserAuthorizationKey = 6; -- Change -1 to the appropriate UserAuthorizationKey
46     --
47     -- Recreate all of the foreign keys prior after loading the star schema
48     --
49     -- Check row count before truncation
50     EXEC [Project2].[ShowTableStatusRowCount]
51         @UserAuthorizationKey = 2, -- Change -1 to the appropriate UserAuthorizationKey
52         @TableStatus = N''Row Count after loading the star schema''
53     --
54     --
55     EXEC [Project2].[AddForeignKeysToStarSchemaData] @UserAuthorizationKey = 1; -- Change -1 to the appropriate UserAuthorizationKey
56     --
57     DECLARE @WorkFlowStepTableRowCount INT;
58     SET @WorkFlowStepTableRowCount = 0;
59     SET @EndingDateTime DATETIME2 = SYSDATETIME();
60     DECLARE @EndingDateTime DATETIME2 = SYSDATETIME();
61     EXEC [Process].[Log_TrackWorkflow] 'Load Star Schema Data',
62         @WorkFlowStepTableRowCount,
63         @StartingDateTime,
64         @EndingDateTime,
65         @UserAuthorizationKey;
66 END;
```



Utils.DropProcsInCSCI331FinalProject

```
1 USE [BIClass]
2 GO
3 /***** Object: StoredProcedure [Utils].[DropProcsInCSCI331FinalProject]   Script Date: 11/17/2020 6:19:45 PM *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8 -- *****
9 -- Author:      Erik Kim
10 -- Create date: 11/15/2020
11 -- Description: Drop procs
12 -- *****
13 ALTER procedure [Utils].[DropProcsInCSCI331FinalProject] @UserAuthorizationKey INT
14 as
15 begin
16     set nocount on;
17     DECLARE @StartingDateTime DATETIME2 = SYSDATETIME();
18     --select concat('drop procdure if exists ', schema_name(o.schema_id), '.', name)
19     --from sys.objects as o
20     --where o.type = 'P'
21     -- and o.schema_id = 9;
22
23     drop proc if exists Project2.Load_SalesManagers;
24     drop proc if exists Project2.Load_DimProductSubcategory;
25     drop proc if exists Project2.Load_DimProductCategory;
26     drop proc if exists Project2.Load_DimGender;
27     drop proc if exists Project2.Load_DimMaritalStatus;
28     drop proc if exists Project2.Load_DimOccupation;
29     drop proc if exists Project2.Load_DimOrderDate;
30     drop proc if exists Project2.Load_DimTerritory;
31     drop proc if exists Project2.Load_DimProduct;
32     drop proc if exists Project2.Load_DimCustomer;
33     drop proc if exists Project2.Load_Data;
34     drop proc if exists Project2.TruncateStarSchemaData;
35     drop proc if exists Project2.LoadStarSchemaData;
36     DROP PROC IF EXISTS Project2.AddForeignKeysToStarSchemaData;
37     DROP PROC IF EXISTS Project2.DropForeignKeysFromStarSchemaData;
38     DROP PROC IF EXISTS Project2.ShowTableStatusRowCount;
39
40     DECLARE @WorkFlowStepTableRowCount INT;
41     SET @WorkFlowStepTableRowCount = 0;
42     DECLARE @EndingDateTime DATETIME2 = SYSDATETIME();
43     EXEC [Process].[usp_TrackWorkFlow] 'Drop_Procedures',
44         @WorkFlowStepTableRowCount,
45         @StartingDateTime,
46         @EndingDateTime,
47         @UserAuthorizationKey;
48
49     DROP PROC IF EXISTS Process.usp_TrackWorkFlow;
50 end;
```



Harjit's Stored Procedures

- Project2.[Load_DimMaritalStatus]
- Project2.[ShowTableStatusRowCount]



Project2.[Load_DimMaritalStatus] Code

```
USE [BIClass];
GO
/***** Object:  StoredProcedure [Project2].[Load_DimMaritalStatus]
Script Date: 11/14/2020 4:19:20 PM *****/
SET ANSI_NULLS ON;
GO
SET QUOTED_IDENTIFIER ON;
GO

-- =====
-- Author:      Harjit Liyal
-- Create date: 11/14/2020
-- Description: [Project2].[Load_DimMaritalStatus]
-- =====

ALTER PROCEDURE [Project2].[Load_DimMaritalStatus] @UserAuthorizationKey INT
AS
BEGIN
    SET NOCOUNT ON;

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

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

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

    INSERT INTO [CH01-01-Dimension].[DimMaritalStatus]
    (
        MaritalStatus,
        MaritalStatusDescription,
        UserAuthorizationKey,
        DateAdded,
        DateOfLastUpdate
    )
```

```
SELECT DISTINCT
    MaritalStatus,
    CASE
        WHEN OLD.MaritalStatus = 'M' THEN
            'Married'
        ELSE
            'Single'
    END AS MaritalStatusDescription,
    @UserAuthorizationKey,
    @DateAdded,
    @DateOfLastUpdate
FROM FileUpload.OriginallyLoadedData AS OLD;

EXEC (
    DROP VIEW IF EXISTS G10_4.uvw_DimMaritalStatus');
EXEC (
    CREATE VIEW G10_4.uvw_DimMaritalStatus AS
    SELECT MaritalStatus, MaritalStatusDescription, UserAuthorizationKey, DateAdded, DateOfLastUpdate
    FROM [CH01-01-Dimension].[DimMaritalStatus] ');
---VIEW for NEW Table---
DECLARE @EndingDateTime DATETIME2;
SET @EndingDateTime = SYSDATETIME();

DECLARE @WorkflowStepTableRowCount INT;
SET @WorkflowStepTableRowCount = (SELECT COUNT(*) FROM [CH01-01-Dimension].[DimMaritalStatus]);

EXEC [Process].[usp_TrackWorkflow] 'Procedure: [Project2].[Load_MaritalStatus] loads data into [CH01-01-Dimension].[DimMaritalStatus]',
    @WorkflowStepTableRowCount,
    @StartingDateTime,
    @EndingDateTime,
    @UserAuthorizationKey;

SELECT * FROM G10_4.uvw_DimMaritalStatus;
END;
```

Note: Entire code is supplied in this slides' "Notes" tab



Project2.[ShowTableStatusRowCount] Code

```
USE [BIClass];
GO
/***** Object: StoredProcedure [Project2].[ShowTableStatusRowCount]    Script Date: 11/14/2020 11:21:23 PM *****/
SET ANSI_NULLS ON;
GO
SET QUOTED_IDENTIFIER ON;
GO
--
-- Author: Harjit Loyal
-- Create date: 11/14/2020
-- Description: [Project2].[ShowTableStatusRowCount]
--
ALTER PROCEDURE [Project2].[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 = 'CH01-01-Dimension.DimCustomer',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimCustomer
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.DimGender',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimGender
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.DimMaritalStatus',
```

```
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimMaritalStatus
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.DimOccupation',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimOccupation
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.DimOrderDate',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimOrderDate
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.DimProduct',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimProduct
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.DimProductCategory',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimProductCategory
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.DimProductSubcategory',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimProductSubcategory
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.DimTerritory',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].DimTerritory
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Dimension.SalesManagers',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Dimension].SalesManagers
    UNION ALL
    SELECT TableStatus = @TableStatus,
           TableName = 'CH01-01-Fact.Data',
           [Row Count] = COUNT(*)
    FROM [CH01-01-Fact].[Data]
    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: [Project2].[ShowStatusRowCount] loads data into [Project2].[ShowTableStatusRowCount]',
                                       @WorkflowStepTableRowCount,
                                       @StartingDateTime,
                                       @EndingDateTime,
                                       @UserAuthorizationKey;

END;
```

Note: Entire code is supplied in this slides' "Notes" tab



Danny's Stored Procedures

- Project2.Load_DimTerritory
- Project2.Load_SalesManagers



Project2.Load_DimTerritory

```
CREATE PROCEDURE [Project2].[Load_DimTerritory]
    @UserAuthorizationKey int
AS
BEGIN
    SET NOCOUNT ON

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

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

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

    INSERT INTO [CH01-01-Dimension].DimTerritory
        ([TerritoryRegion], [TerritoryCountry], [TerritoryGroup], UserAuthorizationKey, DateAdded, DateOfLastUpdate)

    SELECT DISTINCT FUp.[TerritoryRegion], FUp.[TerritoryCountry], FUp.[TerritoryGroup], @UserAuthorizationKey, @DateAdded, @DateOfLastUpdate
    FROM FileUpload.OriginallyLoadedData AS FUp

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

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = (SELECT COUNT(*) FROM [CH01-01-Dimension].DimTerritory);

    EXEC('
    DROP VIEW IF EXISTS G10_4.uvw_DimTerritory')
    EXEC('
    CREATE VIEW G10_4.uvw_DimTerritory AS
    SELECT TerritoryKey, TerritoryGroup, TerritoryCountry, TerritoryRegion, UserAuthorizationKey, DateAdded, DateOfLastUpdate
    FROM [CH01-01-Dimension].[DimTerritory] ')

    EXEC [Process].[usp_TrackWorkFlow]
        'Procedure: [Project2].[Load_DimTerritory] loads data into [CH01-01-Dimension].[DimTerritory]',
        @WorkFlowStepTableRowCount,
        @StartingDateTime,
        @EndingDateTime,
        @UserAuthorizationKey

    SELECT *
    FROM G10_4.uvw_DimTerritory
END
```

*Full code for
stored procedure
is in notes



Project2.Load_SalesManagers

*Full Code for
stored procedure is
in notes

```
CREATE PROCEDURE [Project2].[Load_SalesManagers]
    @UserAuthorizationKey int
AS
BEGIN
    SET NOCOUNT ON;

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

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

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

    INSERT INTO [CH01-01-Dimension].[SalesManagers]
        (SalesManager, Category, UserAuthorizationKey, DateAdded, DateOfLastUpdate)

    SELECT DISTINCT FileUpload_OriginallyLoadedData.[SalesManager], FileUpload_OriginallyLoadedData.[ProductSubcategory], @UserAuthorizationKey, @DateAdded, @DateOfLastUpdate
    FROM FileUpload_OriginallyLoadedData

    UPDATE [CH01-01-Dimension].[SalesManagers]
    SET Office = 'Redmond'
    WHERE SalesManager = 'Maurizio Macagno' OR SalesManager = 'Marco Russo';

    UPDATE [CH01-01-Dimension].[SalesManagers]
    SET Office = 'Seattle'
    WHERE SalesManager = 'Alberto Ferrari' OR SalesManager = 'Luis Bonifaz';

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

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = (SELECT COUNT(*) FROM [CH01-01-Dimension].[SalesManagers]);

    ----New Table View----

    EXEC('
    DROP VIEW IF EXISTS G10_4.uvw_SalesManagers')
    EXEC('
    CREATE VIEW G10_4.uvw_SalesManagers AS
    SELECT SalesManagerKey, SalesManager, Category, Office, UserAuthorizationKey, DateAdded, DateOfLastUpdate
    FROM [CH01-01-Dimension].[SalesManagers] ')

    ----New Table View----
```

```
EXEC [Process].[usp_TrackWorkFlow]
    'Procedure: [Project2].[Load_SalesManagers] loads data into [CH01-01-Dimension].[SalesManagers]',
    @WorkFlowStepTableRowCount,
    @StartingDateTime,
    @EndingDateTime,
    @UserAuthorizationKey

SELECT *
FROM G10_4.uvw_SalesManagers
END;
```



Marlon's Stored Procedures

- Project2.Load_DimGender
- Project2.Load_DimCustomer



Project2.Load_DimGender

```
-- =====
ALTER PROCEDURE [Project2].[Load_DimGender] @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();

    INSERT INTO [CH01-01-Dimension].[DimGender] (
        Gender, GenderDescription, UserAuthorizationKey, DateAdded, DateOfLastUpdate)
    SELECT DISTINCT Gender, CASE Gender WHEN 'M' THEN 'Male' ELSE 'Female' END AS GenderDescription, @UserAuthorizationKey, @DateAdded, @DateOfLastUpdate
    FROM FileUpload.OriginallyLoadedData

    ---VIEW for NEW Table---
    EXEC(
        DROP VIEW IF EXISTS G10_4.uvw_DimGender')
    EXEC(
        CREATE VIEW G10_4.uvw_DimGender AS
        SELECT Gender, GenderDescription, UserAuthorizationKey, DateAdded, DateOfLastUpdate
        FROM [CH01-01-Dimension].[DimGender] ')
    ---VIEW for NEW Table--

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

    DECLARE @WorkflowStepTableRowCount INT;
    SET @WorkflowStepTableRowCount = (SELECT COUNT(*) FROM [CH01-01-Dimension].[DimGender]);

    EXEC [Process].[usp_TrackWorkflow]
        'Procedure: [Project2].[Load_DimGender] loads data into [CH01-01-Dimension].[DimGender]',
        @WorkflowStepTableRowCount,
        @StartingDateTime,
        @EndingDateTime,
        @UserAuthorizationKey

    EXEC('SELECT * FROM G10_4.uvw_DimGender')
END;
```

Code in Notes section



Project2.Load_DimCustomer

```
-- =====
ALTER PROCEDURE [Project2].[Load_DimCustomer] @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();

    INSERT INTO [CH01-01-Dimension].[DimCustomer] (
        CustomerName, UserAuthorizationKey, DateAdded, DateOfLastUpdate)
    SELECT DISTINCT CustomerName, @UserAuthorizationKey, @DateAdded, @DateOfLastUpdate
    FROM FileUpload.OriginallyLoadedData

    ---VIEW for NEW Table---
    EXEC('
    DROP VIEW IF EXISTS G10_4.uvw_DimCustomer')
    EXEC('
    CREATE VIEW G10_4.uvw_DimCustomer AS
    SELECT CustomerKey, CustomerName, UserAuthorizationKey, DateAdded, DateOfLastUpdate
    FROM [CH01-01-Dimension].[DimCustomer] ')
    ---VIEW for NEW Table--

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

    DECLARE @WorkFlowStepTableRowCount INT;
    SET @WorkFlowStepTableRowCount = (SELECT COUNT(*) FROM [CH01-01-Dimension].DimCustomer);

    EXEC [Process].[usp_TrackWorkFlow]
        'Procedure: [Project2].[Load_DimCustomer] loads data into [CH01-01-Dimension].[DimCustomer]',
        @WorkFlowStepTableRowCount,
        @StartingDateTime,
        @EndingDateTime,
        @UserAuthorizationKey

    EXEC('SELECT * FROM G10_4.uvw_DimCustomer')
```

Code in Notes section



Haibo's Stored Procedures



[Project2].[Load_DimOccupation]



[Project2].[Load_DimOrderDate]



```

GO
--DROP VIEW IF EXISTS Loaderr
GO
--CREATE VIEW Loaderr AS
--SELECT * FROM [CH01-01-Dimension].[DimOccupation]
GO
/***** Object: StoredProcedure [Project2].[Load_DimOccupation]    Script Date: 11/12/2020 3:10:41 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- Author:      Haibo Liu
-- Create date: 11/12/2020
-- Description: [Project2].[Load_DimOccupation]
-- *****
DROP PROCEDURE IF EXISTS [Project2].[Load_DimOccupation];
go

create PROCEDURE [Project2].[Load_DimOccupation]
    @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();

    INSERT INTO [CH01-01-Dimension].[DimOccupation]
    (Occupation,UserAuthorizationKey,DateAdded,DateOfLastUpdate)

```

```

SELECT DISTINCT FileUpload.OriginallyLoadedData.[Occupation],@User,
FROM FileUpload.OriginallyLoadedData

---VIEW for NEW Table---
EXEC('
DROP VIEW IF EXISTS G10_4.uvw_DimOccupation')
EXEC('
CREATE VIEW G10_4.uvw_DimOccupation AS
SELECT OccupationKey,Occupation,UserAuthorizationKey,DateAdded,DateOfLastUpdate
FROM [CH01-01-Dimension].[DimOccupation] ')
---VIEW for NEW Table---

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

DECLARE @WorkflowStepTableRowCount INT;
SET @WorkflowStepTableRowCount = @@ROWCOUNT;

EXEC [Process].[usp_TrackWorkflow]
'Procedure: [Project2].[Load_DimOccupation] loads data into [
@WorkflowStepTableRowCount,
@StartingDateTime,
@EndingDateTime,
@UserAuthorizationKey

SELECT *
FROM G10_4.uvw_DimOrderDate
END

TRUNCATE TABLE Process.WorkflowSteps;
EXEC [Project2].[Load_DimOccupation] 7;

```

[Project2].[Load_DimOccupation]



[Project2].[Load_DimOrder Date]

```
USE [BIClass]
GO

/***** Object: StoredProcedure [Project2].[Load_DimOrderDate]    Script Date: 11/14/2020 12:15:49 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- Author:      Maibo Liu
-- Create date: 11/14/2020
-- Description: [Project2].[Load_DimOrderDate]
-- *****/
ALTER PROCEDURE [Project2].[Load_DimOrderDate]
    @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 = SYSUTETIME();

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

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

    Insert into [CH01-01-Dimension].[DimOrderDate]
    ((OrderDate),(MonthName),MonthNumber,[Year],UserAuthorizationKey,@DateAdded,@DateOfLastUpdate)

    select DISTINCT A.[OrderDate],A.[MonthName],A.MonthNumber,A.[Year],@UserAuthorizationKey,@DateAdded,@DateOfLastUpdate
    from FileUpload.OriginallyLoadedData as A

    ---VIEW for NEW Table---
    EXEC('
    DROP VIEW IF EXISTS G10_4.uvw_DimOrderDate')
    EXEC('
    CREATE VIEW G10_4.uvw_DimOrderDate AS
    SELECT OrderDate, MonthName, MonthNumber, UserAuthorizationKey, DateAdded, DateOfLastUpdate
    FROM [CH01-01-Dimension].[DimOrderDate] ')
    ---VIEW for NEW Table---

    DECLARE @EndingDateTime DATETIME2;
    set @EndingDateTime = SYSUTETIME();

    DECLARE @WorkflowStepTableRowCount INT;
    SET @WorkflowStepTableRowCount = @@ROWCOUNT;

    EXEC [Process].[usp_TrackWorkflow]
        'Procedure: [Project2].[Load_DimOrderDate] loads data into [CH01-01-Dimension].[DimOrderDate]',
        @WorkflowStepTableRowCount,
        @StartingDateTime,
        @EndingDateTime,
        @UserAuthorizationKey

    SELECT *
    FROM G10_4.uvw_DimOccupation

END

--Truncate
TRUNCATE TABLE [CH01-01-Dimension].[DimOrderDate]
--Execute
EXEC [Project2].[Load_DimOrderDate] 7;
```



Jonathan's Stored Procedures

- [Project2].[Load_DimProductCategory]
- [Project2].[Load_DimProductSubCategory]



Jonathan's Stored Procedures

[Project2].[Load_DimProductCategory]

```
USE BIClass;
GO

/***** Object:  StoredProcedure [Project2].[Load_DimProductCategory]
Script Date: 11/14/2020 9:37:05 PM *****/

SET ANSI_NULLS ON;
GO
SET QUOTED_IDENTIFIER ON;
GO
=====
-- Author:      Jonathan Eng
-- Create date: 11/14/2020
-- Description: [Project2].[Load_DimProductCategory]
=====
ALTER PROCEDURE [Project2].[Load_DimProductCategory] @UserAuthorizationKey INT
AS
BEGIN
    SET NOCOUNT ON;

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

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

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

    INSERT INTO [CH01-01-Dimension].[DimProductCategory]
    (
        ProductCategory,
        UserAuthorizationKey,
        DateAdded,
        DateOfLastUpdate
    )
    SELECT DISTINCT
        FileUpload.OriginallyLoadedData.[ProductCategory],
        @UserAuthorizationKey,
        @DateAdded,
        @DateOfLastUpdate
    FROM FileUpload.OriginallyLoadedData;
```

```
---VIEW for NEW Table---
EXEC ('
    DROP VIEW IF EXISTS G10_4.uvw_DimProductCategory');
EXEC ('
    CREATE VIEW G10_4.uvw_DimProductCategory AS
    SELECT ProductCategoryKey, ProductCategory, UserAuthorizationKey, DateAdded,
    DateOfLastUpdate
    FROM [CH01-01-Dimension].[DimProductCategory] ');
=====
DECLARE @EndingDateTime DATETIME2;
SET @EndingDateTime = SYSDATETIME();

DECLARE @WorkFlowStepTableRowCount INT;
SET @WorkFlowStepTableRowCount =
(
    SELECT COUNT(*) FROM [CH01-01-Dimension].DimProductCategory
);

EXEC [Process].[usp_TrackWorkFlow] 'Procedure: [Project2].[Load_DimProductCategory]
loads data into [CH01-01-Dimension].[DimProductCategory]',
    @WorkFlowStepTableRowCount,
    @StartingDateTime,
    @EndingDateTime,
    @UserAuthorizationKey;

SELECT *
FROM G10_4.uvw_DimProductCategory;
END;
```



Jonathan's Stored Procedures

[Project2].[Load_DimProductSubCategory]

```
USE BIClass;
GO

/***** Object: StoredProcedure [Project2].[Load_DimProductSubcategory]
Script Date: 11/14/2020 9:37:05 PM *****/
SET ANSI_NULLS ON;
GO
SET QUOTED_IDENTIFIER ON;
GO
-- =====
-- Author: Jonathan Eng
-- Create date: 11/14/2020
-- Description: [Project2].[Load_DimProductSubcategory]
-- =====
ALTER PROCEDURE [Project2].[Load_DimProductSubcategory] @UserAuthorizationKey INT
AS
BEGIN
    SET NOCOUNT ON;

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

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

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

    INSERT INTO [CH01-01-Dimension].[DimProductSubCategory]
    (
        ProductCategoryKey,
        ProductSubcategory,
        UserAuthorizationKey,
        DateAdded,
        DateOfLastUpdate
    )
    SELECT DISTINCT
        DPC.ProductCategoryKey,
        OLD.ProductSubcategory,
        @UserAuthorizationKey,
        @DateAdded,
        @DateOfLastUpdate
    FROM FileUpload_OriginallyLoadedData AS OLD
    FULL JOIN [CH01-01-Dimension].[DimProductCategory] AS DPC
    ON OLD.[ProductCategory] = DPC.[ProductCategory];
```

```
---VIEW for NEW Table---
EXEC ('
DROP VIEW IF EXISTS G10_4.uvw_DimProductSubCategory');
EXEC ('
CREATE VIEW G10_4.uvw_DimProductSubCategory AS
SELECT ProductSubCategoryKey, ProductCategoryKey, ProductSubcategory,
UserAuthorizationKey, DateAdded, DateOfLastUpdate
FROM [CH01-01-Dimension].[DimProductSubCategory] ');
---VIEW for NEW Table---
DECLARE @EndingDateTime DATETIME2;
SET @EndingDateTime = SYSDATETIME();

DECLARE @WorkFlowStepTableRowCount INT;
SET @WorkFlowStepTableRowCount =
(
    SELECT COUNT(*) FROM [CH01-01-Dimension].[DimProductSubCategory]
);

EXEC [Process].[usp_TrackWorkFlow] 'Procedure:
[Project2].[Load_DimProductSubCategory] loads data into [CH01-01-
Dimension].[DimProductSubCategory]',
@WorkFlowStepTableRowCount,
@StartingDateTime,
@EndingDateTime,
@UserAuthorizationKey;

SELECT *
FROM G10_4.uvw_DimProductSubCategory;

END;
```



Jamil's Stored Procedures

- [Project2].[Load_DimProduct]
- [Project2].[Load_Data]



[Project2].[Load_DimProduct]

```
USE [BIClass]
GO
/***** Object: StoredProcedure [Project2].[Load_DimProduct]    Script Date: 11/17/2020 8:04:14 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Jamil Kocacal
-- Create date: 11/14/2020
-- Description: Populate the Data table
-- =====
ALTER PROCEDURE [Project2].[Load_DimProduct] @UserAuthorizationKey INT
AS
BEGIN
    SET NOCOUNT ON;

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

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

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

    INSERT INTO [CH01-01-Dimension].[DimProduct] (
        ProductSubcategoryKey,
        ProductCategory,
        ProductSubcategory,
        ProductCode,
        ProductName,
        Color,
        ModelName,
        UserAuthorizationKey,
        DateAdded,
        DateOfLastUpdate
    )
    SELECT DISTINCT
        DPSC.ProductSubcategoryKey,
        OLD.ProductCategory,
        DPSC.ProductSubcategory,
        OLD.ProductCode,
        OLD.ProductName,
        OLD.Color,
        OLD.ModelName,
        @UserAuthorizationKey,
        @DateAdded,
        @DateOfLastUpdate
    FROM FileUpload.OriginallyLoadedData AS OLD
```

```
    OLD.ModelName,
    @UserAuthorizationKey,
    @DateAdded,
    @DateOfLastUpdate
FROM FileUpload.OriginallyLoadedData AS OLD
FULL JOIN [CH01-01-Dimension].[DimProductSubCategory] AS DPSC
    ON OLD.[ProductSubcategory] = DPSC.[ProductSubcategory];

---VIEW for NEW Table---
EXEC ('
DROP VIEW IF EXISTS G10_4.uvw_DimProduct');
EXEC ('
CREATE VIEW G10_4.uvw_DimProduct AS
SELECT
    ProductKey,
    ProductSubcategoryKey,
    ProductCategory,
    ProductSubcategory,
    ProductCode,
    ProductName,
    Color,
    ModelName,
    UserAuthorizationKey,
    DateAdded,
    DateOfLastUpdate
FROM [CH01-01-Dimension].[DimProduct] ');
---VIEW for NEW Table--

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

DECLARE @WorkflowStepTableRowCount INT;
SET @WorkflowStepTableRowCount = (SELECT COUNT(*) FROM [CH01-01-Dimension].[DimProduct]);

EXEC [Process].[usp_TrackWorkflow] 'Procedure: [Project2].[Load_DimProduct] loads data into [CH01-01-Dimension].[DimProduct]',
    @WorkflowStepTableRowCount,
    @StartingDateTime,
    @EndingDateTime,
    @UserAuthorizationKey;

SELECT *
FROM G10_4.uvw_DimProduct;

END;
```



[Project2].[Load_Data]

```
USE [BIClass]
GO
/***** Object: StoredProcedure [Project2].[Load_Data]    Script Date: 11/17/2020 7:54:13 PM
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Jamil Kocacal
-- Create date: 11/14/2020
-- Description: Populate the Data table
-- =====
ALTER PROCEDURE [Project2].[Load_Data] @UserAuthorizationKey INT
AS
BEGIN
    SET NOCOUNT ON;

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

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

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

    INSERT INTO [CH01-01-Fact].[Data] (
        SalesKey,
        SalesManagerKey,
        OccupationKey,
        TerritoryKey,
        ProductKey,
        CustomerKey,
        ProductCategory,
        SalesManager,
        ProductSubcategory,
        ProductCode,
        ProductName,
        Color,
        ModelName,
        OrderQuantity,
        UnitPrice,
        ProductStandardCost,
        SalesAmount,
        OrderDate,
        [MonthName],
        MonthNumber,
        [Year],
        CustomerName,
        MaritalStatus,
        Gender,
        Education,
        Occupation,
        TerritoryRegion,
        TerritoryCountry,
        TerritoryGroup,
        UserAuthorizationKey,
        DateAdded,
        DateOfLastUpdate
    )
    SELECT DISTINCT
```

```
old.SalesKey,
sm.SalesManagerKey,
do.OccupationKey,
dt.TerritoryKey,
dp.ProductKey,
dc.CustomerKey,
old.ProductCategory,
old.SalesManager,
old.ProductSubcategory,
old.ProductCode,
old.ProductName,
old.Color,
old.ModelName,
old.OrderQuantity,
old.UnitPrice,
old.ProductStandardCost,
old.SalesAmount,
old.OrderDate,
old.[MonthName],
old.MonthNumber,
old.[Year],
old.CustomerName,
old.MaritalStatus,
old.Gender,
old.Education,
old.Occupation,
old.TerritoryRegion,
old.TerritoryCountry,
old.TerritoryGroup,
@UserAuthorizationKey,
@DateAdded,
@DateOfLastUpdate
FROM
Fileupload.OriginallyLoadedData AS old LEFT JOIN
[CH01-01-Dimension].DimProduct AS dp
    on dp.ProductName = old.ProductName AND
    dp.ProductCode = old.ProductCode LEFT JOIN
[CH01-01-Dimension].DimTerritory AS dt
    on dt.TerritoryCountry = old.TerritoryCountry AND
    dt.TerritoryGroup = old.TerritoryGroup AND
    dt.TerritoryRegion = old.TerritoryRegion INNER JOIN
[CH01-01-Dimension].DimCustomer AS dc
    on dc.CustomerName = old.CustomerName LEFT JOIN
[CH01-01-Dimension].SalesManagers AS sm
    on sm.SalesManager = old.SalesManager and
    sm.Category = old.ProductSubcategory LEFT JOIN
[CH01-01-Dimension].DimOccupation AS do
    on do.Occupation = old.Occupation

---VIEW for NEW Table---
EXEC ('
DROP VIEW IF EXISTS G10_4.uvw_FactData');
EXEC ('
CREATE VIEW G10_4.uvw_FactData AS
SELECT
    Saleskey,
    SalesManagerKey,
    OccupationKey,
    TerritoryKey,
    ProductKey,
    CustomerKey,
```

```
---VIEW for NEW Table---
EXEC ('
DROP VIEW IF EXISTS G10_4.uvw_FactData');
EXEC ('
CREATE VIEW G10_4.uvw_FactData AS
SELECT
    Saleskey,
    SalesManagerKey,
    OccupationKey,
    TerritoryKey,
    ProductKey,
    CustomerKey,
    ProductCategory,
    SalesManager,
    ProductSubcategory,
    ProductCode,
    ProductName,
    Color,
    ModelName,
    OrderQuantity,
    UnitPrice,
    ProductStandardCost,
    SalesAmount,
    OrderDate,
    [MonthName],
    MonthNumber,
    [Year],
    CustomerName,
    MaritalStatus,
    Gender,
    Education,
    Occupation,
    TerritoryRegion,
    TerritoryCountry,
    TerritoryGroup,
    UserAuthorizationKey,
    DateAdded,
    DateOfLastUpdate
FROM [CH01-01-Fact].[Data] ');
---VIEW for NEW Table--

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

DECLARE @workFlowStepTableRowCount INT;
SET @workFlowStepTableRowCount = (SELECT COUNT(*) FROM [CH01-01-Fact].[Data]);

EXEC [Process].[usp_TrackWorkflow] 'Procedure: [Project2].[Load_DimProductsSubCategory] loads data into [CH01-01-Dimension].[DimProductsSubCategory]',
    @workFlowStepTableRowCount,
    @StartingDateTime,
    @EndingDateTime,
    @UserAuthorizationKey;

SELECT *
FROM G10_4.uvw_FactData;
END;
```

