Contents

Overview of the project to RECREATE THE BICLASS DATABASE START SCHEMA

You will re-create the BIClass Database Star Schema using the FileUpload.OriginallyLoadedData table[1].

You will create stored procedures to load the individual table of the star schema.

They will be executed within one stored procedure which will pass one parameter to that may truncate all of the data except from the FileUpload.OriginallyLoadedData table.

You add two new tables:

1. [CH01-01-Dimension].[DimProductCategory]
2. [CH01-01-Dimension].[DimProductSubcategory]

The table will be related to the product table using the grandparent to parent to child relationship below:

1. [CH01-01-Dimension].[DimProductCategory]
    a. [CH01-01-Dimension].[DimProductSubcategory]
        i. [CH01-01-Dimension].[DimProduct]

The stored procedures are stubs where you fill in the appropriate SQL. Please document the each of the procedures.

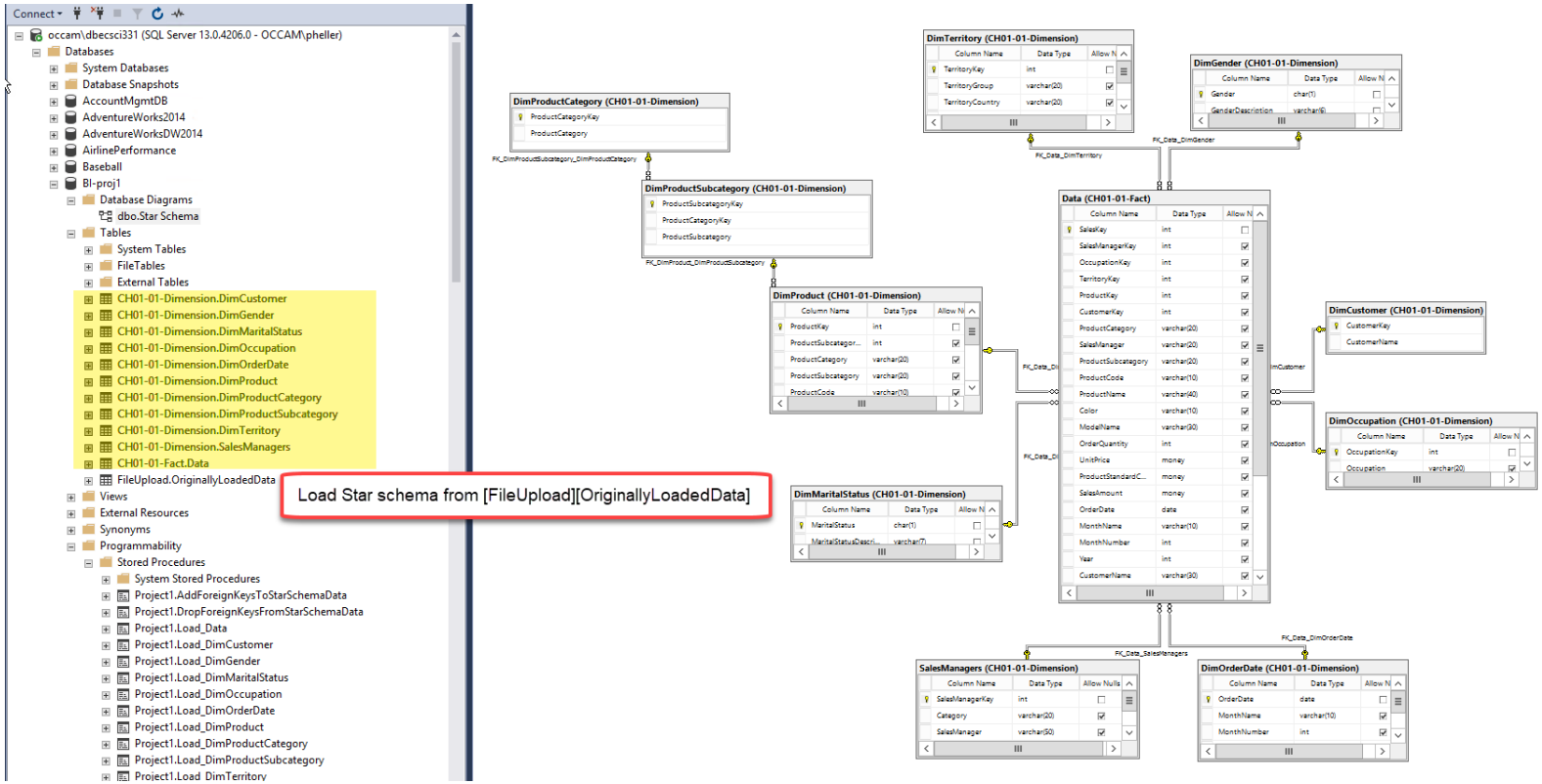
Please be aware of referential integrity[2] issues when deleting/ inserting. The assignment is customizable to the way that you envision the design. This will be an individual and group project.

[1] The FileUpload.OriginallyLoadedData will have to be joined with 3 dimensional tables to get their surrogate key to populate the fact table/

[2] Hint, the have to drop and recreate the foreign keys as part of the load process.

Create a group PowerPoint presentation that describes the efforts of the team with voice annotation[3] as well as text. Choice the best design of the team with contributions by each individual.

Database Information



[3] https://youtu.be/wlha2MaoJEk

Sample Stored Procedure

```
USE [BIClass]
GO
/****** Object:  StoredProcedure [Project1].[Load_Data]    Script Date: 9/3/2017 4:46:38 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      YourName
-- Create date:
-- Description:
-- =============================================
ALTER PROCEDURE [Project1].[Load_Data]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    print 'insert statements for procedure here'
END
```

Sample Join to load the fact table

```
INSERT INTO [CH01-01-Fact].Data
    (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)
SELECT
    old.SalesManagerKey, old.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
FROM
    FileUpload.OriginallyLoadedData AS old INNER JOIN
    [CH01-01-Dimension].DimProduct AS dp
            ON dp.ProductName = old.ProductName INNER 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
END;
```

Truncate Tables Example

```
-- =============================================
ALTER PROCEDURE [Project1].[TruncateStarSchemaData]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
        truncate table [CH01-01-Fact].data;
        truncate table [CH01-01-Dimension].SalesManagers;
        truncate table [CH01-01-Dimension].DimProductSubcategory;
        truncate table [CH01-01-Dimension].DimProductCategory;
        truncate table [CH01-01-Dimension].DimGender;
        truncate table [CH01-01-Dimension].DimMaritalStatus;
        truncate table [CH01-01-Dimension].DimOccupation;
        truncate table [CH01-01-Dimension].DimOrderDate;
        truncate table [CH01-01-Dimension].DimTerritory;
        truncate table [CH01-01-Dimension].DimProduct;
        truncate table [CH01-01-Dimension].DimCustomer;

end
```

[Project1].[LoadStarSchemaData]

```
ALTER PROCEDURE [Project1].[LoadStarSchemaData]
    -- Add the parameters for the stored procedure here
    @YesNo CHAR(1) = 'Y'
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @return_value INT;
    --
    --  Drop All of the foreign keys prior to truncating tables in the star schema
    --
    --IF (@YesNo = 'Y') EXEC @return_value = [Project1].[TruncateStarSchemaData];
    --
    EXEC  [Project1].[DropForeignKeysFromStarSchemaData];
    --
    --  Always truncate the Star Schema Data
    --
    EXEC @return_value = [Project1].[TruncateStarSchemaData];
    --
    --  Load the star schema
    --
    EXEC @return_value = [Project1].[Load_DimProductCategory];
    EXEC @return_value = [Project1].[Load_DimProductSubcategory];
    EXEC @return_value = [Project1].[Load_DimProduct];
    EXEC @return_value = [Project1].[Load_SalesManagers];
    EXEC @return_value = [Project1].[Load_DimGender];
    EXEC @return_value = [Project1].[Load_DimMaritalStatus];
    EXEC @return_value = [Project1].[Load_DimOccupation];
    EXEC @return_value = [Project1].[Load_DimOrderDate];
    EXEC @return_value = [Project1].[Load_DimTerritory];
    EXEC @return_value = [Project1].[Load_DimCustomer];
    EXEC @return_value = [Project1].[Load_Data];
  --
    --  Recreate all of the foreign keys prior after loading the star schema
    --
    EXEC [Project1].[AddForeignKeysToStarSchemaData];
    --

END;
```