Gyubeom Kim

INFO 498

Randal Root

2018/08/18

**Final Project**

**Introduction:**

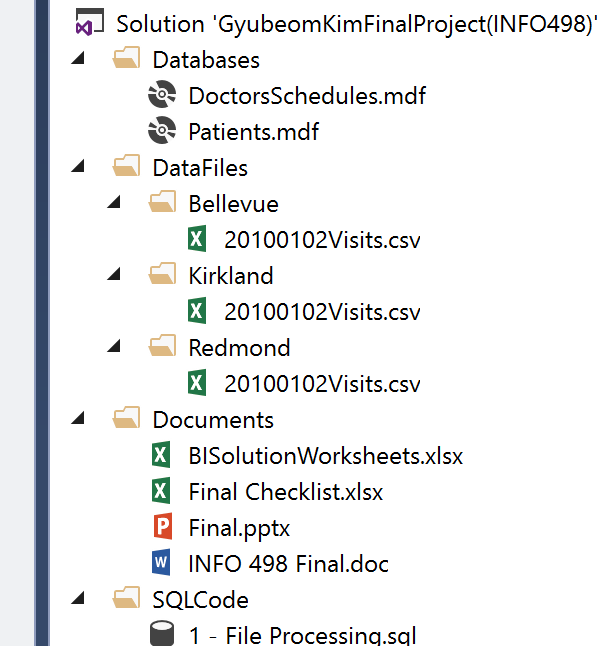
For the Final Project of one of my academic courses, INFO 498, I created an entire ETL solution. Before the process of making it, I reviewed the whole data structures by reviewing all data diagrams on the documents. I also organized data that I need for the project because organizing is the key for the complex database design. Then, I started to utilize SQL, SSIS, and SQL Agent Job for the project.

**Summary:**

For conducting the project for creating an entire ETL process using various tasks. Firstly, I reviewed whole data and created OLTP database. Then, I put data from some files to the OLTP. As putting the data in OLTP database, I was able to insert it into the newly created data ware house. In data ware house, I created whole processes as stored procedures as a process of creating ETL; therefore, I could successfully execute my ETL processes in the SSIS package. Then, I made a SQL Agent job for back up three databases on the assigned destination. Through all that steps, I could end with the final project.

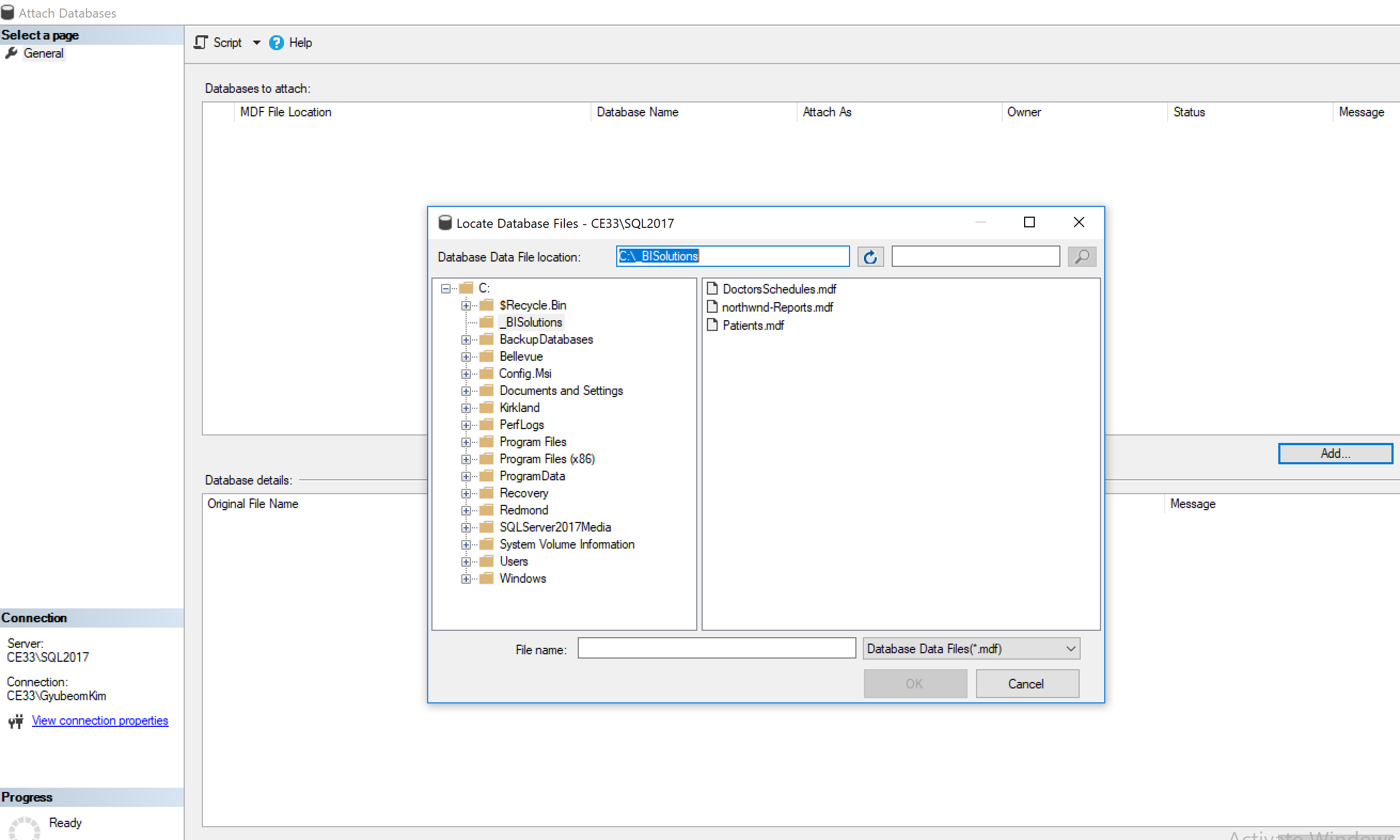
**Technical Details:**

**Creating a Blank Visual Studio Solution**

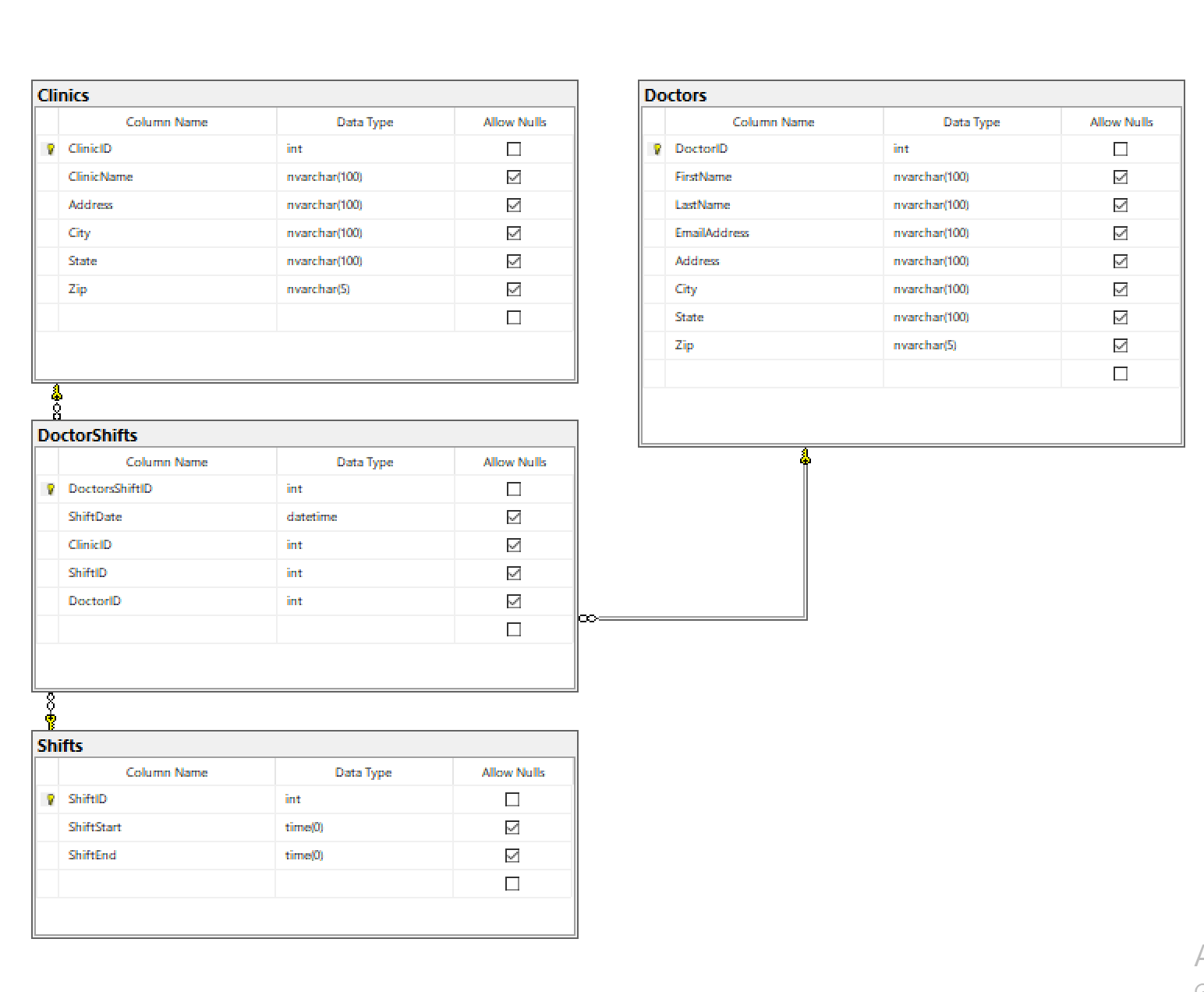


I simply created a blank visual studio solution for organizing the code and documents. I created some new folders and classified each code and documents as putting into the assigned folder as the picture shown above. Organizing is the key for the complex database design.

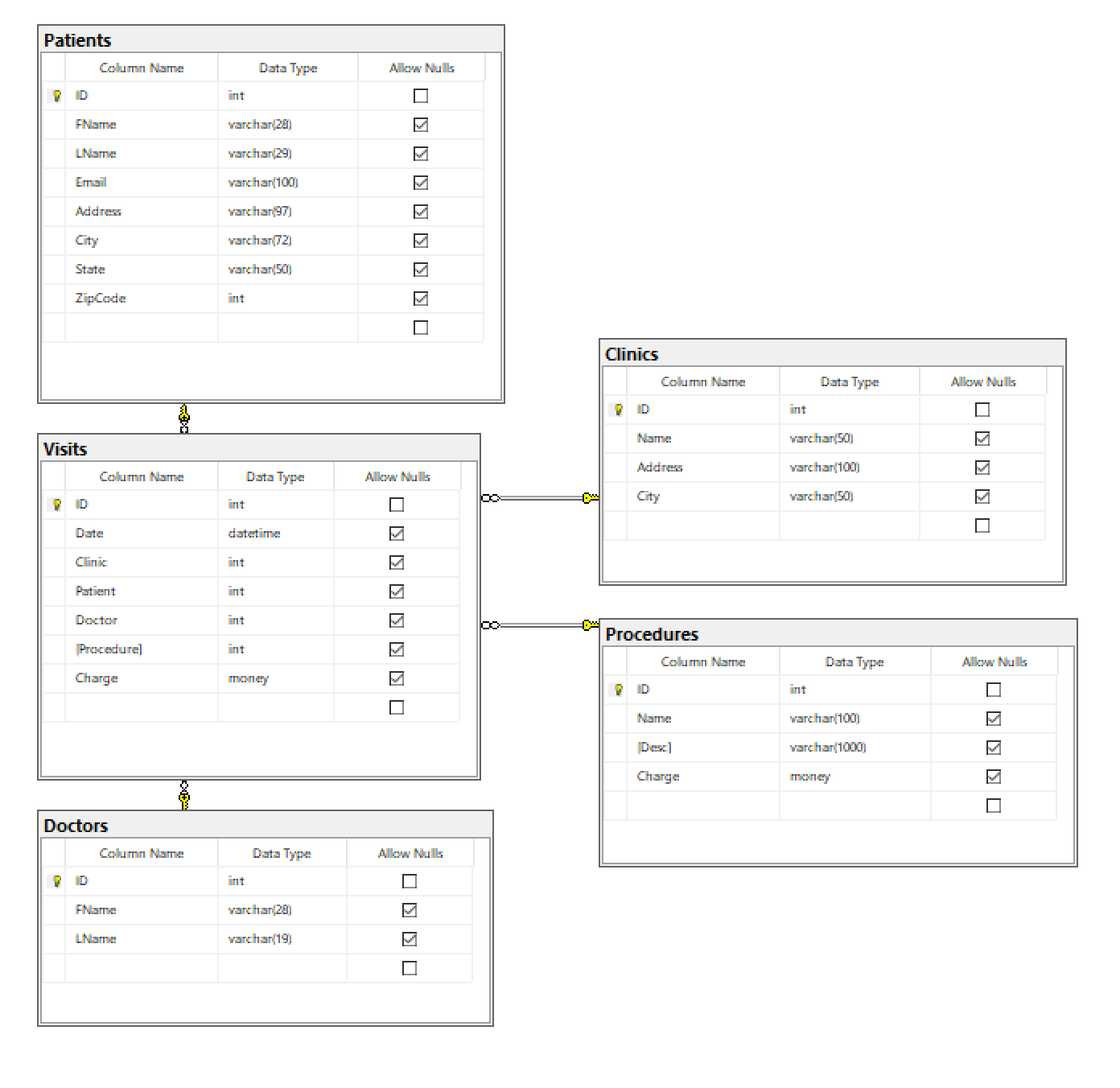
**Creating the OLTP Database and Data Warehouse**



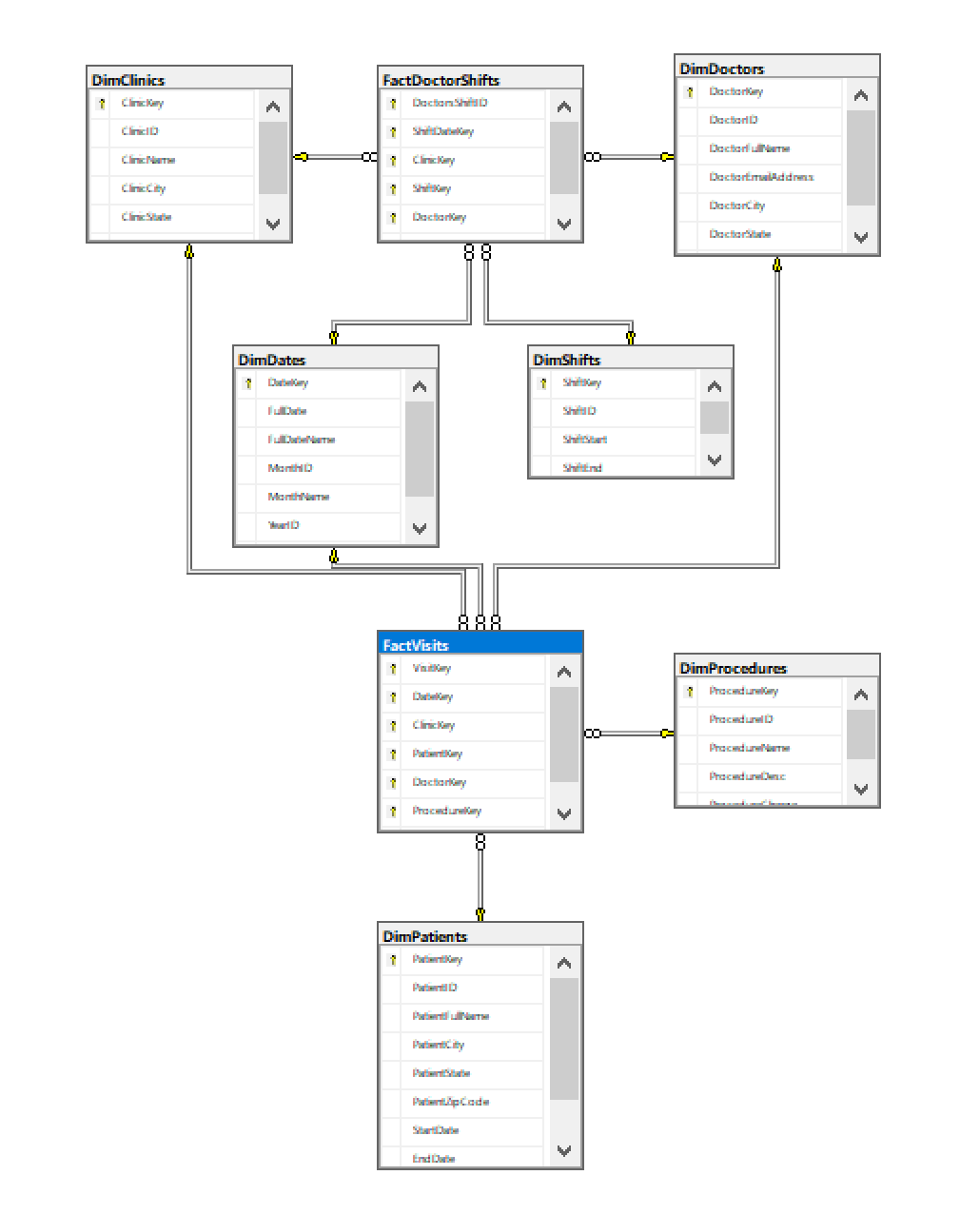
As you can see in the picture shown above, I attached OLTP databases, DoctorSchedulres.mdf and Patients.md. Also, I created Date Warehourse with a given SQL script. The structure of databases can be captured on the picture shown below. As you see the diagrams, it make you easy to undersantd the relationship.



DoctorSchedules diagram

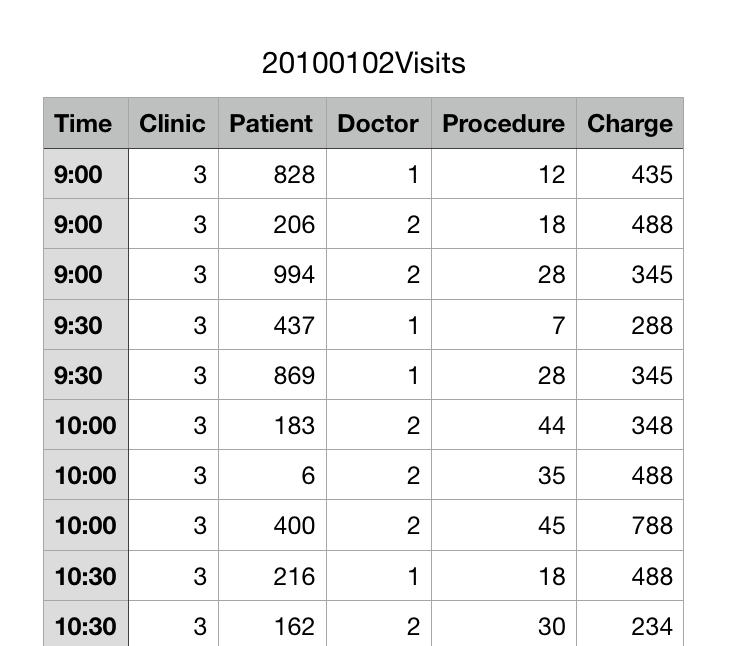
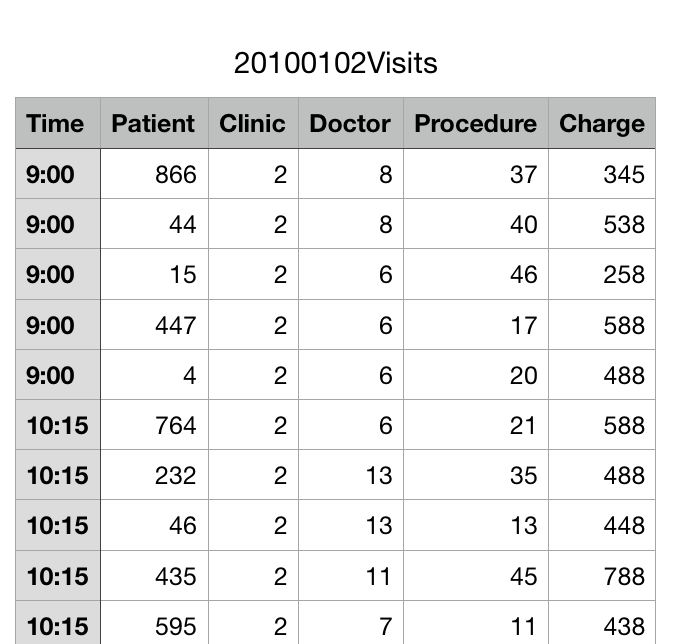
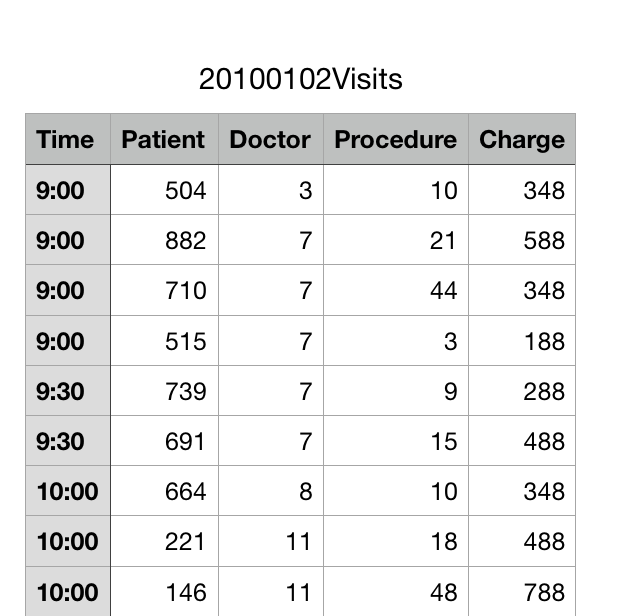


Patients Database



DAta WareHouse

**Review CSV File Data and Insert Into The OLTP database**

****

[Bellevue] [Kirkland] [Redmond]

Before creating a code task for inserting .csv file data into the OLTP database, I firstly reviewed the .csv files. As you can see the picture shown above, I was able to find difference amongst three data files. There is no clinic in Bellevue and no dates in all three tables. After finding such errors, I strarted to create SQL task, inserting such data into the OLTP database, Patients. Firstly, I created a stroed procedure for truncating and creating three tables.

For exmaple,

If(Select Object\_ID('Table Name')) Is Not Null

Truncate Table [dbo].[Table Name];

Else

CREATE TABLE [dbo].[Table Name](

Value / Type

)

Then, I created one more stored procedure for insertig data into the tables.

For example,

BULK INSERT [dbo].[Table Name]

FROM ‘File Path\File Name’

WITH

(FIRSTROW = 2,

FIELDTERMINATOR = ',',

ROWTERMINATOR = '\n'

);

After inserting data into the tables, I created a view for combining all data from tables and editing some missing values (adding ClinicID = 1 to Bellevue and adding date [2010-10-02] for all tables in Time Column).

For example,

Create View vETLNewVisitData

/\* Author: <GyubeomKim>

\*\* Desc: creating view for combining data

\*\* Change Log: When,Who,What

\*\* 2018-08-14,<GyubeomKim>,Created Sproc.

\*/

As

SELECT [Time] AS [Date],[Clinic] = 1, [Patient], [Doctor], [Procedure], [Charge]

From [tempdb].[dbo].[StagingBellevue]

Union All

SELECT [Time] AS [Date],[Clinic], [Patient], [Doctor], [Procedure], [Charge]

From [tempdb].[dbo].[StagingKirkland]

Union All

SELECT [Time] AS [Date],[Clinic] = 1, [Patient], [Doctor], [Procedure], [Charge]

From [tempdb].[dbo].[StagingRedmond]

GO

This is an important process for the next step, merging with visits table in Patients database. If there is difference in columns or type, Merge cannot be happened. As making a view for combining three table data, I could easily insert such data into the visits table.

For example,

With NewRows

As (

Select

[Date] = Cast('2010-01-02' + ' ' + [Date] as datetime)

,[Clinic] = Cast([Clinic] as int) \* 100

,[Patient] = Cast([Patient] as int)

,[Doctor] = Cast([Doctor] as int)

,[Procedure] = Cast([Procedure] as int)

,[Charge] = Cast([Charge] as money)

From [VETLNewVisitData]

Except

Select [Date], [Clinic], [Patient],[Doctor], [Procedure], [Charge]

From [Patients].[dbo].Visits

)

Insert Into [Patients].[dbo].Visits

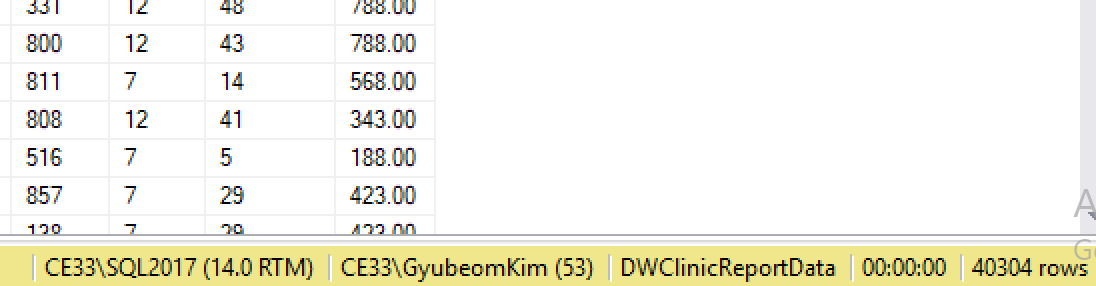
([Date], [Clinic], [Patient],[Doctor], [Procedure], [Charge])

Select

[Date], [Clinic], [Patient],[Doctor], [Procedure], [Charge]

From NewRows

Order By 1,2,3,4,5,6



Select \* From Patients.dbo.Visits;

Through the merging step, 154 data from three table add into the table as the picture shown above.

**Flush and Fill the data warehouse with OLTP database data:**

1. Drop Foreign Keys:

In this task, there few steps for the process. I firstly dropped the Foreign Keys that connected with the Fact tables in Data Warehouse, DWClinicReportData.

For example,

Alter Table [dbo].[FactVisits]

Drop CONSTRAINT [fkFactVisitsToDimClinics];

1. Truncate Table

Then, I truncated table for the process of flush.

For example,

Truncate Table [DWClinicReportData].dbo.DimClinics;

1. Create Views for each table and Fill the data into Tables

**DimClinics, DimDoctors, DimProcedures, DimShifts**

I simply created a view for it. Then, I used type 1 of SCD.

For example,

Create View vETLDimClinics

/\* Author: <GyubeomKim>

\*\* Desc: Extracts and transforms data for DimClinics

\*\* Change Log: When,Who,What

\*\* 2018-08-14,<GyubeomKim>,Created Sproc.

\*/

As

SELECT

[ClinicID] = Cast(DC.ClinicID as int)

,[ClinicName] = Cast(IsNull(DC.ClinicName, 'Missing Data') as nvarchar(100))

,[ClinicCity] = Cast(IsNull(DC.City, 'Missing Data') as nvarchar(100))

,[ClinicState] = Cast(IsNull(DC.[State], 'Missing Data') as nvarchar(100))

,[ClinicZip] = Cast(IsNull(DC.Zip, 'Missing Data') as nvarchar(5))

FROM [DoctorsSchedules].dbo.Clinics as DC

Go

Create Procedure pETLFillDimClinics

/\* Author: <GyubeomKim>

\*\* Desc: Inserts data into DimClinics using the vETLDimClinics

\*\* Change Log: When,Who,What

\*\* 2018-08-15,<GyubeomKim>,Created Sproc.

\*/

AS

Begin

Declare @RC int = 0;

Begin Try

-- ETL Processing Code --

IF ((Select Count(\*) From DimClinics) = 0)

Begin

INSERT INTO [DWClinicReportData].dbo.DimClinics

([ClinicID],[ClinicName],[ClinicCity],[ClinicState],[ClinicZip])

SELECT

[ClinicID]

,[ClinicName]

,[ClinicCity]

,[ClinicState]

,[ClinicZip]

FROM vETLDimClinics

End

Set @RC = +1

End Try

Begin Catch

Print Error\_Message()

Set @RC = -1

End Catch

Return @RC;

End

Go

For the DimShift, I changed some time with using case.

For example,

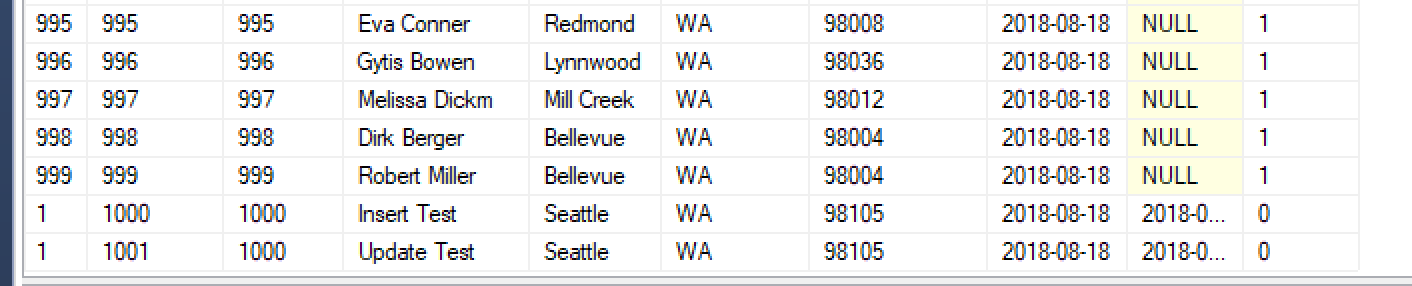
Case When [] Then [] End;

**DimDates**

For DimDates, there is nothing much changed here; therefore I just filled the data.

**DimPatients**

For DimPatients, I used type 2 of SCD. As using type 2, you can track the history of it as the picture shown below.



‘1’ means there is data and ‘0’ means deleted.

**FactDoctorShifts, FactVisits**

As same as I did on other dimensional tables, I created views and procedures. However, I brought all surrogate keys from other tables by using join in the views.

For example,

Create View vETLFactVisits

/\* Author: <GyubeomKim>

\*\* Desc: Extracts and transforms data for FactVisits

\*\* Change Log: When,Who,What

\*\* 2018-08-15,<GyubeomKim>,Created Sproc.

\*/

As

SELECT

[VisitKey] = Cast(PV.ID as int)

,[DateKey] = Cast(DDA.DateKey as int)

,[ClinicKey] = Cast(DC.ClinicKey as int)

,[PatientKey] = Cast(DP.PatientKey as int)

,[DoctorKey] = Cast(DD.DoctorKey as int)

,[ProcedureKey] = Cast(DPR.ProcedureKey as int)

,[ProcedureVistCharge] = Cast(PV.Charge as money)

FROM [Patients].dbo.Visits as PV

JOIN [DWClinicReportData].dbo.DimClinics as DC

ON PV.Clinic = Cast(DC.ClinicID as int) \* 100

JOIN [DWClinicReportData].dbo.DimPatients as DP

On PV.Patient = DP.PatientKey

JOIN [DWClinicReportData].dbo.DimDoctors as DD

On PV.Doctor = DD.DoctorKey

JOIN [DWClinicReportData].dbo.DimProcedures as DPR

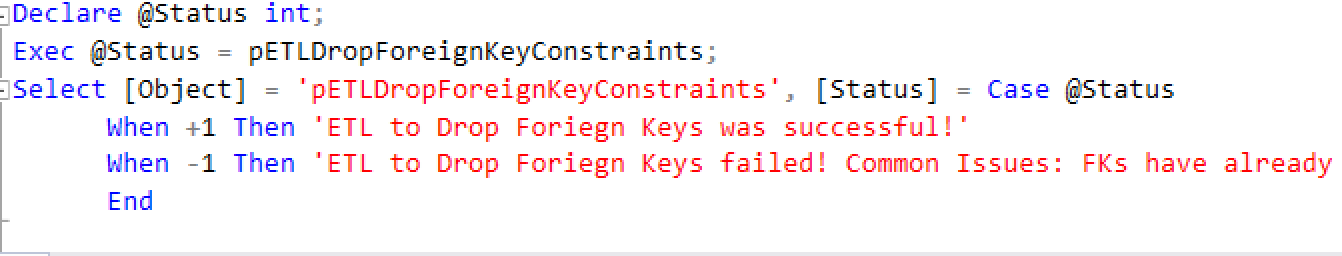
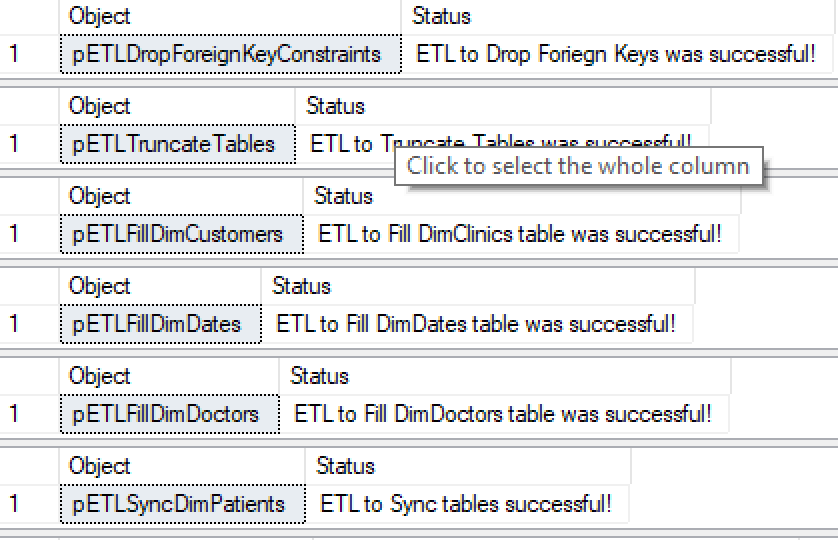
On PV.[Procedure] = DPR.ProcedureKey

JOIN [DWClinicReportData].dbo.DimDates as DDA

On Cast(Convert(nVarchar(50), PV.[Date], 112) as int) = DDA.DateKey;

Go

1. Review

****

Through using such code, I could easily check the result. All procedures worked well.

1. Add Foreign Key

Since I dropped the Foreign Keys as the first step of the process, I re-created for the process.

For example,

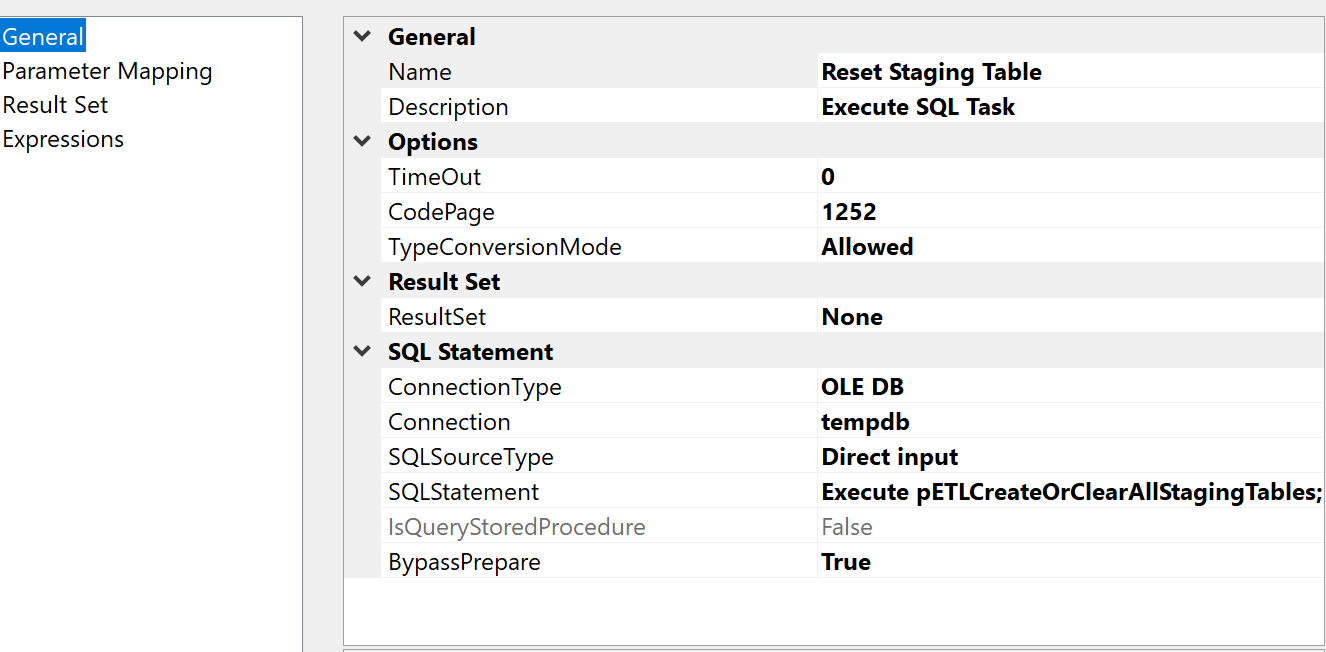
Alter Table [dbo].[FactVisits]

Add Constraint [fkFactVisitsToDimClinics]

FOREIGN KEY (ClinicKey) REFERENCES DimClinics(ClinicKey);

**Creating a SSIS ETL Project:**

I simply created a SSIS project inside of my final solution. Then, I added all the procedures in my SQL file for running this. The picture is shown below.





**Creating a SQL Agent Job for backup all three databases:**

Before creating the actual job for the backup, I created scripts for it.

For example,

BACKUP DATABASE Patients

TO DISK = N'C:\BackDatabases\Patients.bak'

WITH INIT;

Then, I statred to create a job and set the schedule up as once a day as the pictures shown below.

