Sri Lanka Institute of Information Technology

Data warehousing and Business Intelligence

Assignment 1

Student Registration No: IT20178840

Student Name: Weerasinghe T.R

Step 1: Data Set Selection

This data set contains MediTech_Analytics data of American hospital and a related clinic system. These clinic are conducted a different time period of the year. As this is done affiliated to the hospital so, many facilities are provided to the patients .

An appointment is required before visiting these clinics and there is an AttendanceID associated with the AppointmentID So,Patients can attend clinics for several days with the same AppoinmentID.

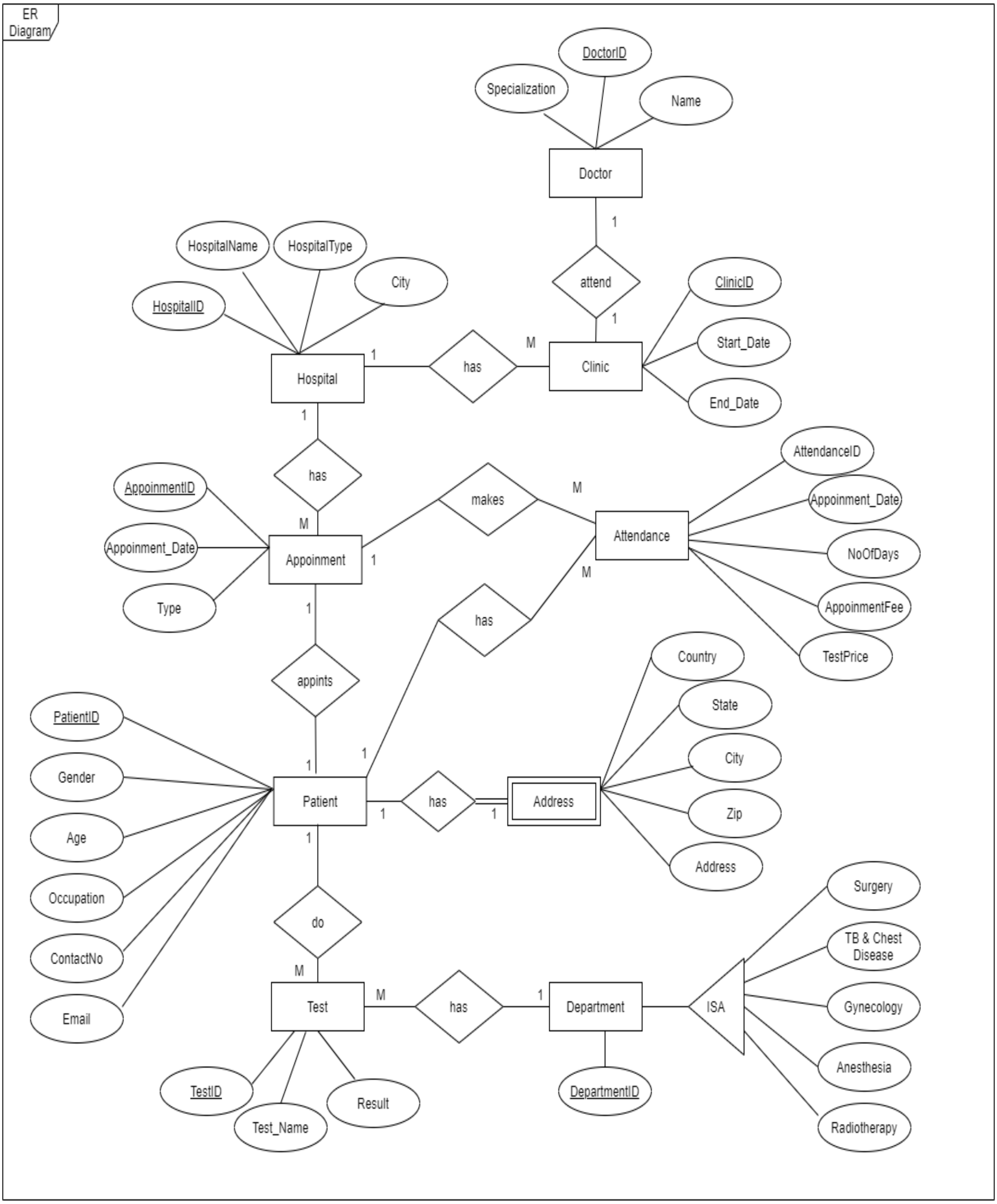
This data set contains data on over 10,000 tests performed by hospitals and affiliated clinics on more than 5,000 patients in two years.

This dataset contains Meditech analytics details,

- Hospital Details
- Patient Details
- Patient Addresses
- Appointment Details
- Tests Details
- Clinic Details
- Department Details
- Attendance Details
- Doctor Details

Also, there are some added details to this database.

Following ER- diagram will describe the scenario of the selected dataset.



Step 2: Preparation of Data Sources

The whole of data was in ‘csv’ file type and they were separated into the following data sources, Database, Text and csv. And they were used to create the following,

1.Database(.bak)

Patient.txt, Attendance.txt, Appoinment.txt, Test.txt, Clinic.txt, Department.txt and Doctor.txt files were imported to the MediTech_analytics Database.

2.Text(.txt)

PatientAddress.txt was used directly.

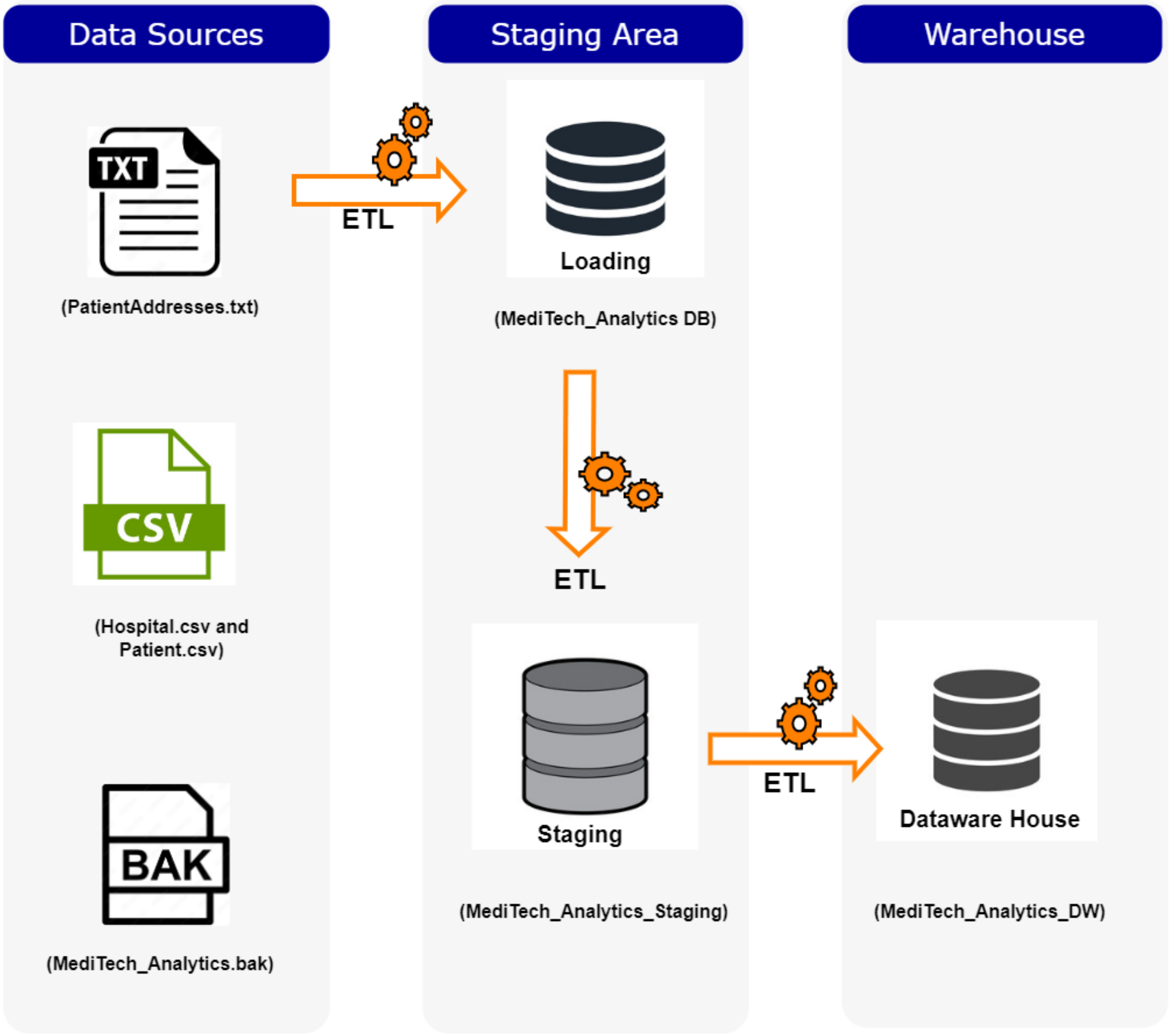
3.Comma Separated Values (.csv)

Patient.csv and hospital.csv was used

Data Source Type	Source Name	Column Name	Data Type	Description
Database File (.bak)s	dbo.Doctor	DoctorID	int	Unique ID
		Name	varchar(100)	Doctor’s name
		Specialization	varchar(100)	Doctor’s specialization
	dbo.Appointment	AppointmentID	int	Unique ID
		PatientID	int	Patient ID
		HospitalID	int	Hospital ID which the Clinic was held
		AppointmentDate	datetime	Appointment Date
		TypeOfAddmission	varchar(100)	Type of Admission
	dbo.Attendance	AttendanceID	int	Unique ID
		PatientID	int	PatientID
		ClinicID	int	ClinicID
		AppointmentID	int	AppointmentID
		AppointmentDate	datetime	Appointment Date
		NoOfDays	int	Number of days patient attend to the clinic
		TestPrice	money	Price of Test
		AppoinmentFee	money	AppoinmentFee
	dbo.Department	DepartmentID	int	Unique ID
		DepartmentName	varchar(100)	Department Name
	dbo.Test	TestID	Int	Unique ID
		PatientID	Int	Patient’s Unique ID
		TestName	varchar(300)	Test Name
		DepartmentID	int	ID of the Department which the test was done

		Result	varchar(50)	Test Result
	dbo.Clinic	ClinicID	int	Unique ID
		StartDate	datetime	Clinic Start Date
		EndDate	datetime	Clinic End Date
		HospitalID	int	Hospital ID which the Clinic was held
		DoctorID	int	DoctorID
CSV File	Patient.csv	PatientID	int	Unique ID
		Gender	nvarchar(255)	Gender (Male/Female)
		Age	int	Age
		Occupation	nvarchar(255)	Patient’s Job
		Phone	nvarchar(255)	Phone Number
		Email	nvarchar(255)	Email Address
	Hospital.csv	HospitalID	int	Unique ID
		HospitalName	nvarchar(255)	Hospital Name
		HospitalType	nvarchar(255)	Hospital Type
		City	nvarchar(255)	City where the hospital is located
Text File	PatientAddress.txt	PatientID	int	Unique ID
		Country	nvarchar(255)	Patient’s Country
		State	nvarchar(255)	Patient’s State
		City	nvarchar(255)	Patient’s City
		ZIP	nvarchar(255)	ZIP code of the Patient
		Address	nvarchar(255)	Patient’s Address

Step 3: Solution Architecture



Above architecture shows the high-level BI solution to the warehouse design.

Data Sources

‘.txt’ component represents Text files,‘.csv’ component is used to display Comma Separated files and ‘.bak’ component represents database files.

Staging Area

Loading DB component represents the process of the creating database tables. Appointment, Test, Attendance, Department, Clinic and Doctor text files was imported to the database and was used to create the tables. And these tables were used as the DB source data.

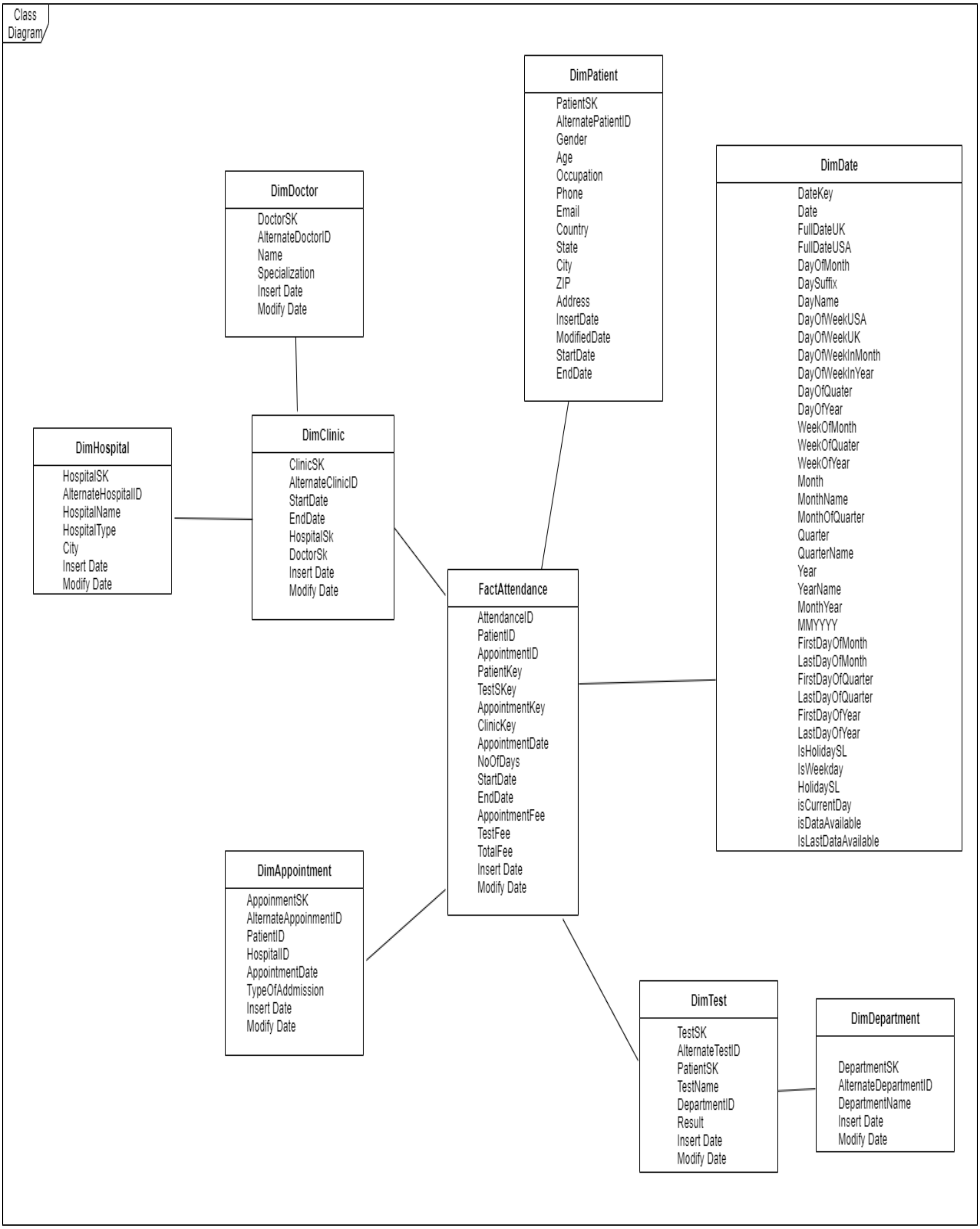
Staging DB component represents creating staging level tables through the ‘Extract’.

Data Warehouse

Data warehouse DB component is used display the cratering dimension tables in the warehouse using ‘Transform’ and ‘Load.’

Step 4: Data Warehouse Design & Development

Following figure will show how the fact table and dimension tables was combined in a rational manner.



Schema Type

For this scenario, snowflake schema type was used.

Dimension Types

- Hierarchical Dimension
  - Date – all the hierarchies in date
  - Patient – country → state → city → ZIP code → address
- Slowly Changing Dimension
  - Attendance – used type 2
    - NoOfDays column set as changing attributes
  - Patient – used type 2
  - Following columns were set as changing attributes.
    - Address
    - Phone Number
    - Country
    - City
    - State
    - ZIP code

- Fact Table
  - Numbers – Test Price, Attendance Fee, Total Amount, NoOfDays
  - FK - Patient ID, Clinic ID, Test ID, Hospital ID, Date Key, Appointment ID, Department ID

Assumptions

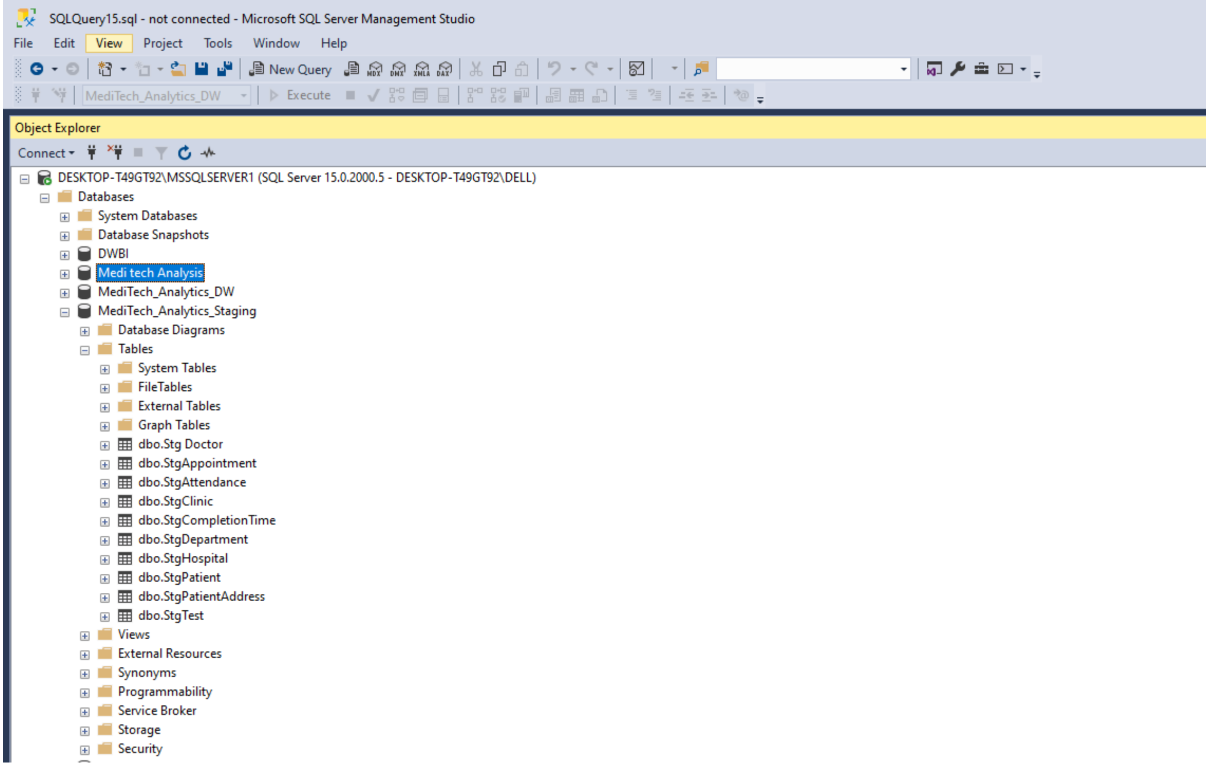
- Patient dimension was considered as a slowly changing dimension.

Step 5: ETL development

1.Extract

In this step, All the data sources were imported to the staging tables by using the relevant Data connection.

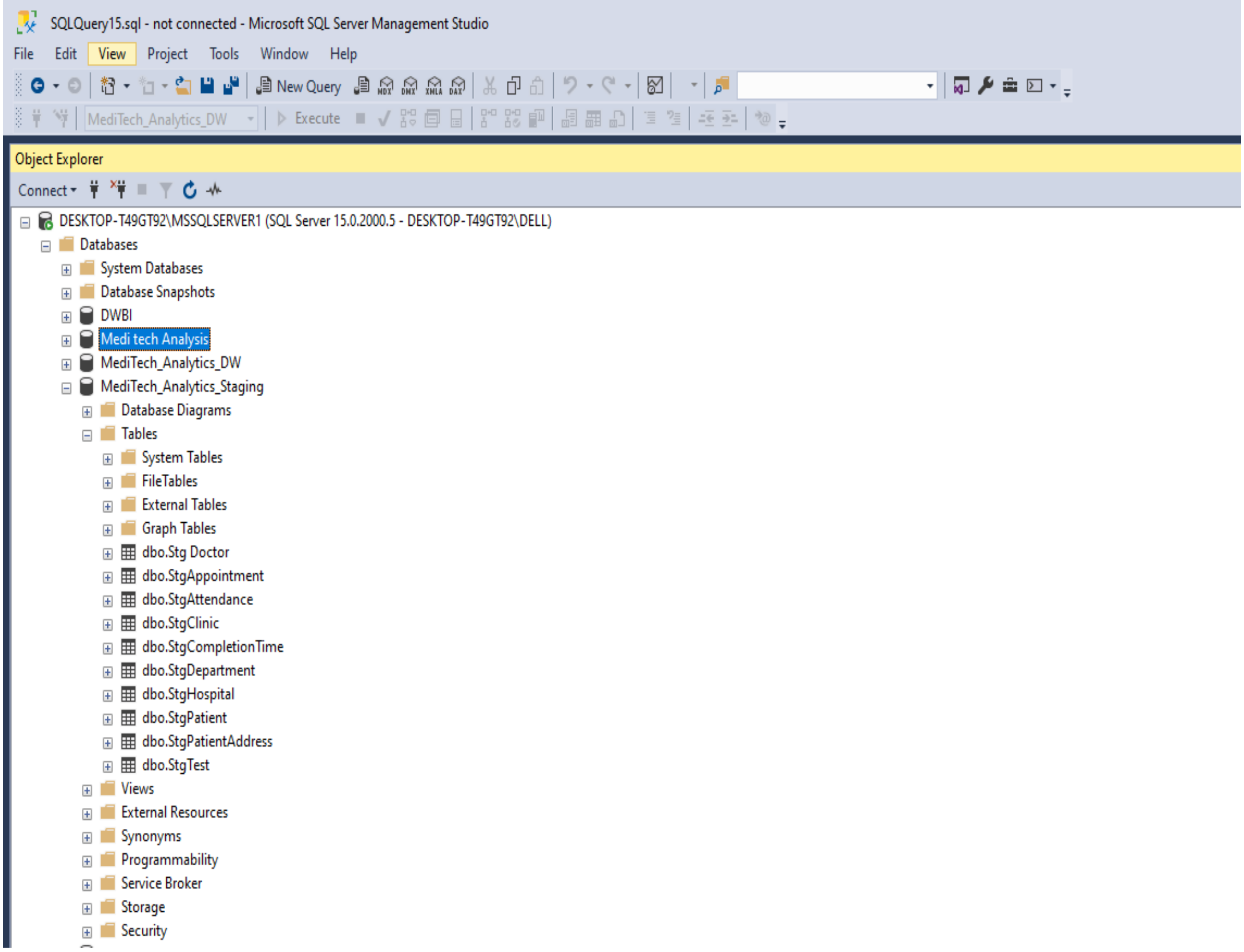
Flat file connection was used for text files and csv files,



DB source connection for DB file. All those tables were imported to the MediTech_Analytics_Staging DB, which contains the below tables,

1. StgHospital
2. StgClinic
3. StgPatientAddress
4. StgPatient
5. StgAppointment
6. StgDepartment
7. SgtTest
8. StgAttendance
9. StgDoctor
10. StgCompletionTime

- **Snapshot of SSMS Staging Database**

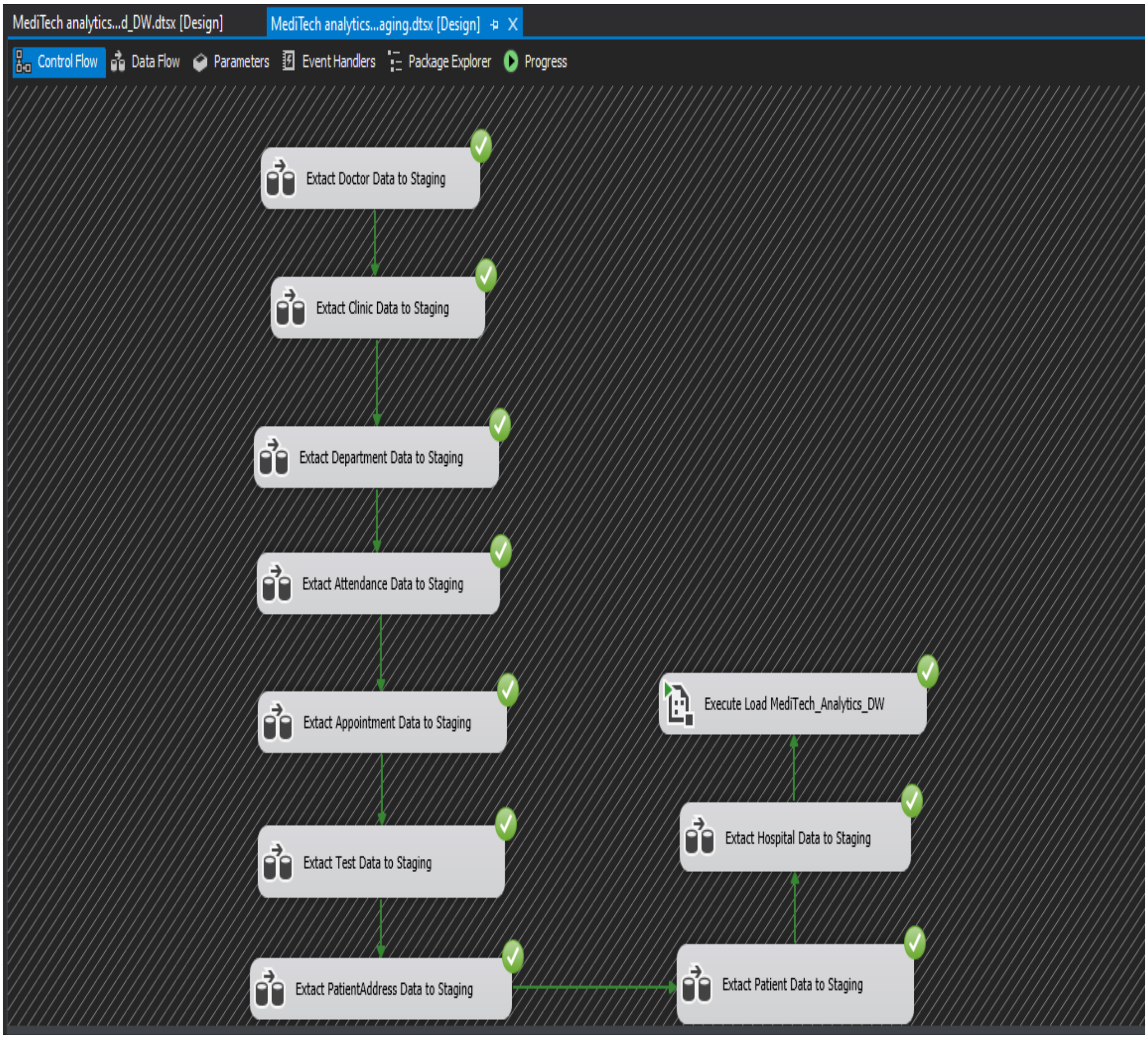


SQLQuery15.sql - not connected - Microsoft SQL Server Management Studio

Object Explorer

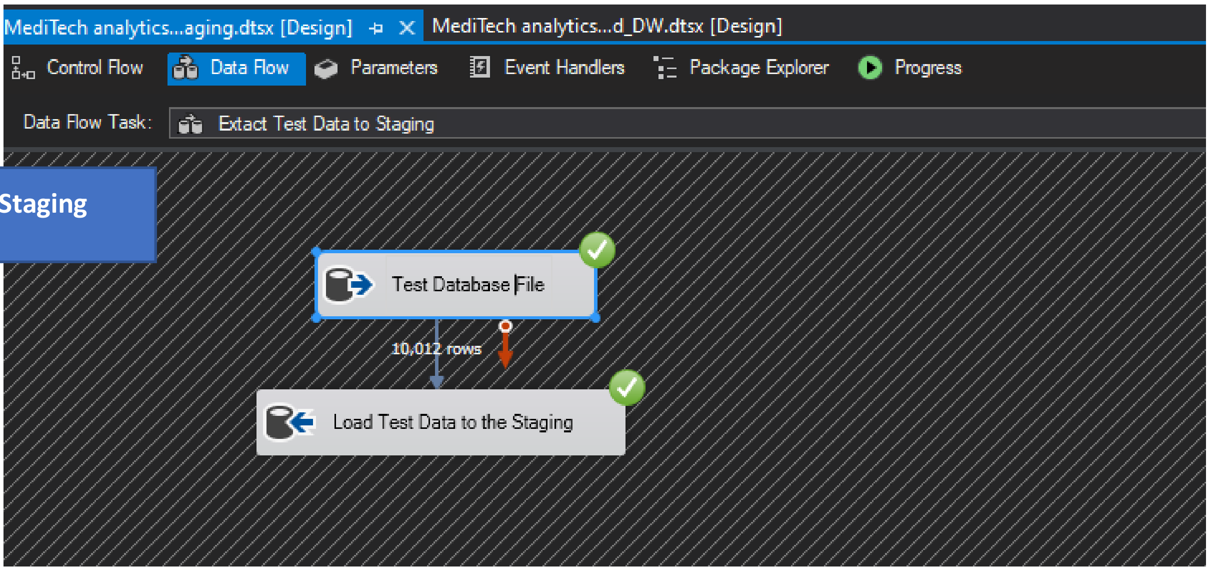
```
DESKTOP-T49GT92\MSSQLSERVER1 (SQL Server 15.0.2000.5 - DESKTOP-T49GT92\DELL)
  Databases
    System Databases
    Database Snapshots
    DWBI
    Medi tech Analysis
    MediTech_Analytics_DW
    MediTech_Analytics_Staging
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.Stg Doctor
        dbo.StgAppointment
        dbo.StgAttendance
        dbo.StgClinic
        dbo.StgCompletionTime
        dbo.StgDepartment
        dbo.StgHospital
        dbo.StgPatient
        dbo.StgPatientAddress
        dbo.StgTest
      Views
      External Resources
      Synonyms
      Programmability
      Service Broker
      Storage
      Security
```

- **Snapshot of Visual Studio Control Flow of Extract**

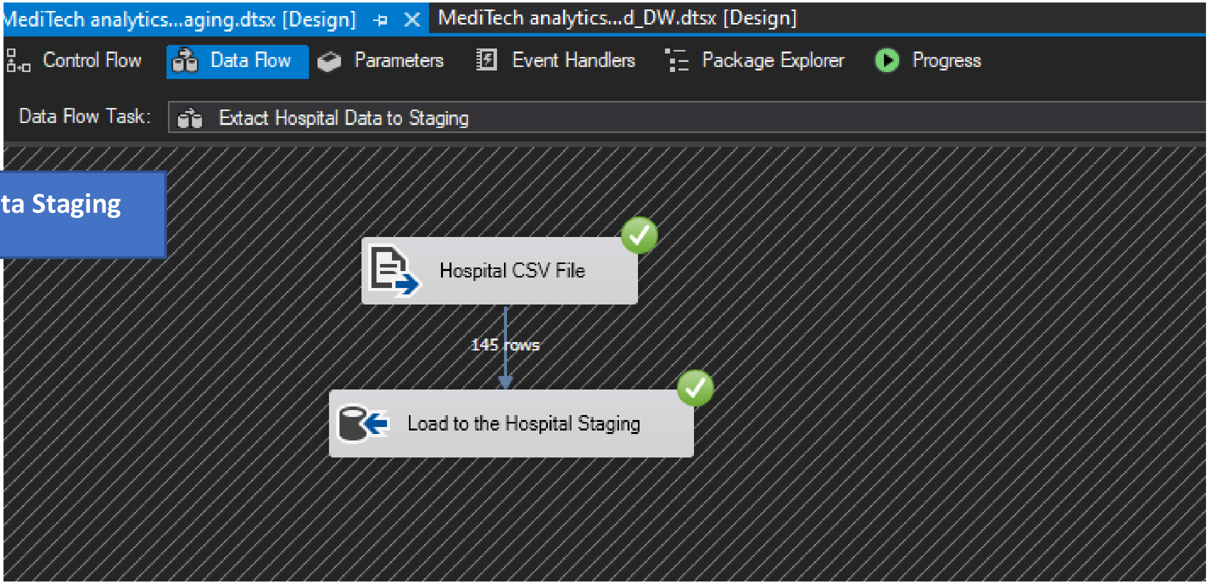

MediTech analytics...d_DW.dtsx [Design]
MediTech analytics...aging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Extact Doctor Data to Staging
Extact Clinic Data to Staging
Extact Department Data to Staging
Extact Attendance Data to Staging
Extact Appointment Data to Staging
Extact Test Data to Staging
Extact PatientAddress Data to Staging
Extact Patient Data to Staging
Extact Hospital Data to Staging
Execute Load MediTech_Analytics_DW

- **Snapshots of several data types of Data Flows**

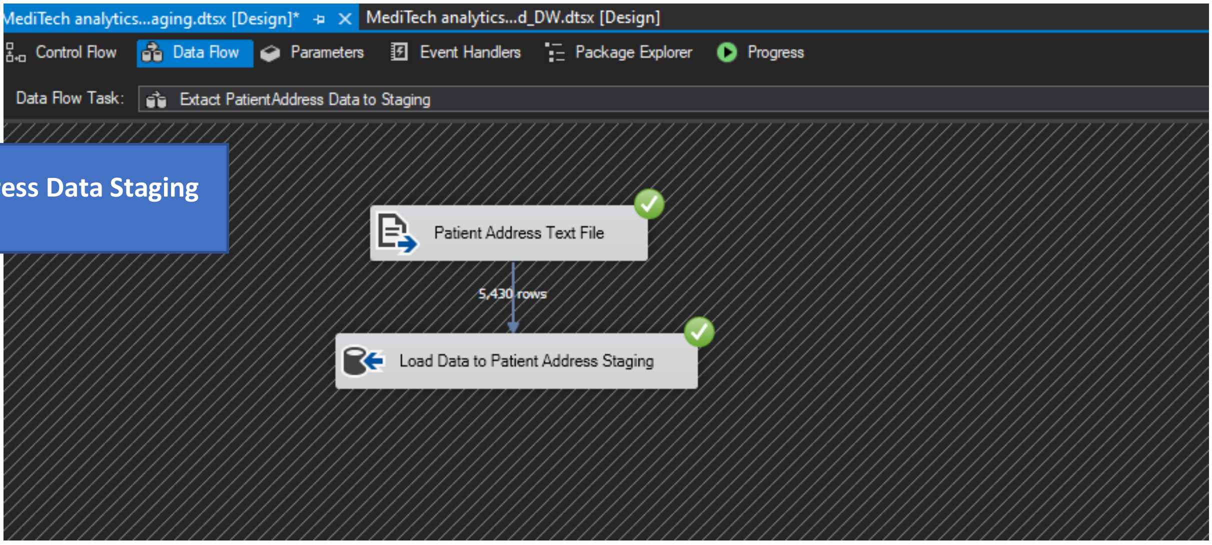
Test Data Staging


MediTech analytics...aging.dtsx [Design]
MediTech analytics...d_DW.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Extact Test Data to Staging
Test Database File
10,012 rows
Load Test Data to the Staging

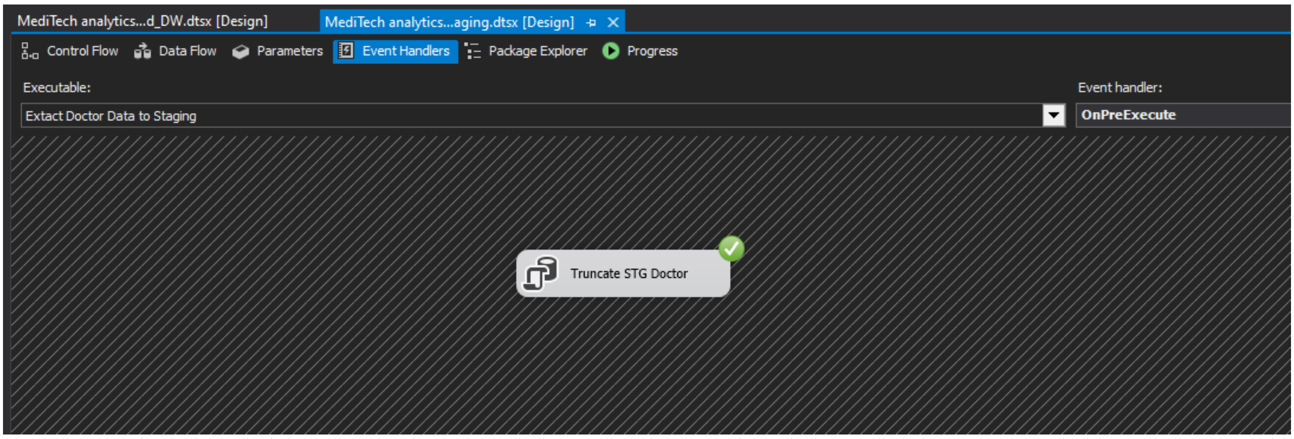
Hospital Data Staging


MediTech analytics...aging.dtsx [Design]
MediTech analytics...d_DW.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Extact Hospital Data to Staging
Hospital CSV File
145 rows
Load to the Hospital Staging

Patient Address Data Staging

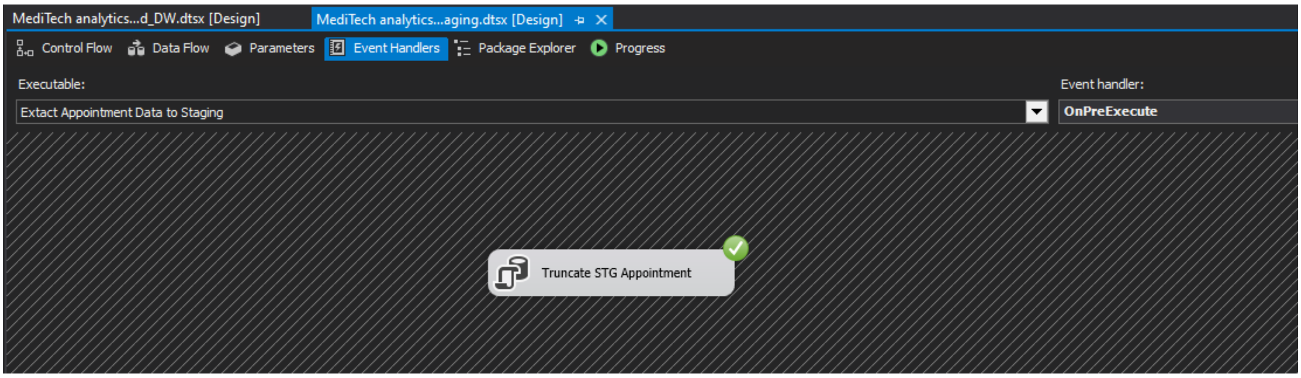

MediTech analytics...aging.dtsx [Design]*
MediTech analytics...d_DW.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Extact PatientAddress Data to Staging
Patient Address Text File
5,430 rows
Load Data to Patient Address Staging

- **Event Handling (Truncate Staging Data)**


MediTech analytics...d_DW.dtsx [Design]
MediTech analytics...aging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress

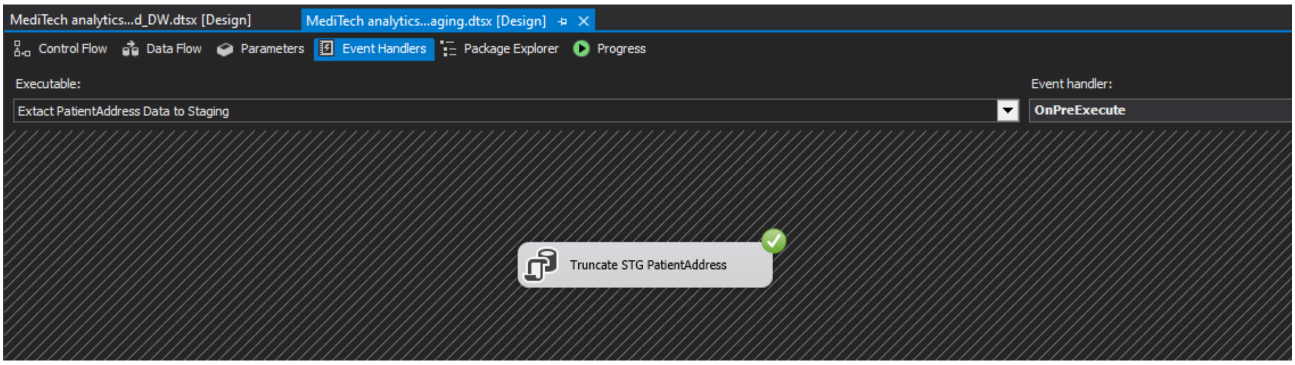
Executable: Extact Doctor Data to Staging

Event handler: OnPreExecute


MediTech analytics...d_DW.dtsx [Design]
MediTech analytics...aging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress

Executable: Extact Appointment Data to Staging

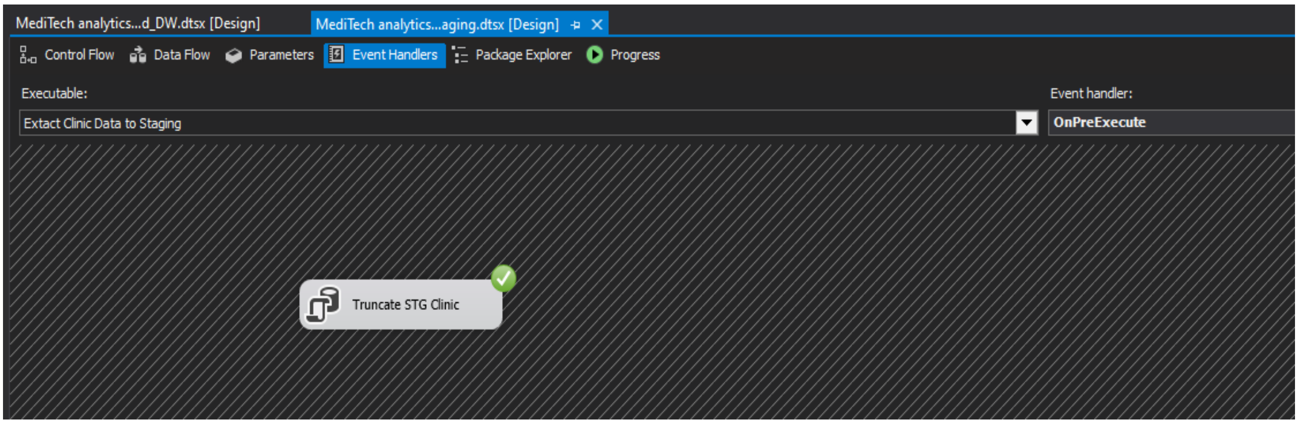
Event handler: OnPreExecute


MediTech analytics...d_DW.dtsx [Design]
MediTech analytics...aging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress

Executable: Extact PatientAddress Data to Staging

Event handler: OnPreExecute

Truncate STG PatientAddress

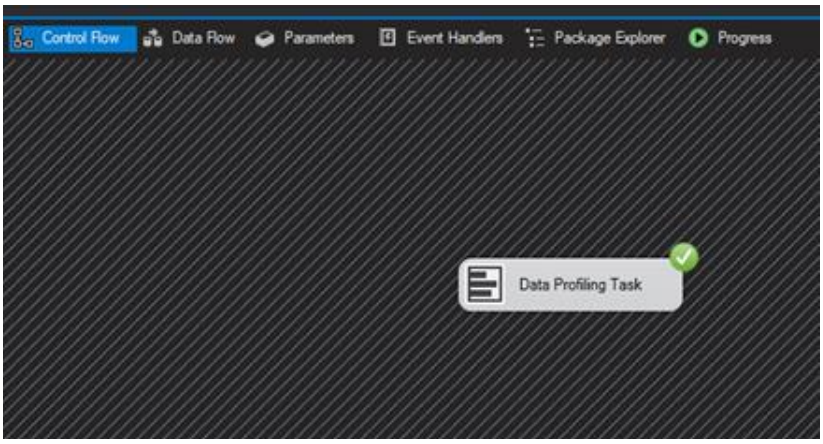

MediTech analytics...d_DW.dtsx [Design]
MediTech analytics...aging.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress

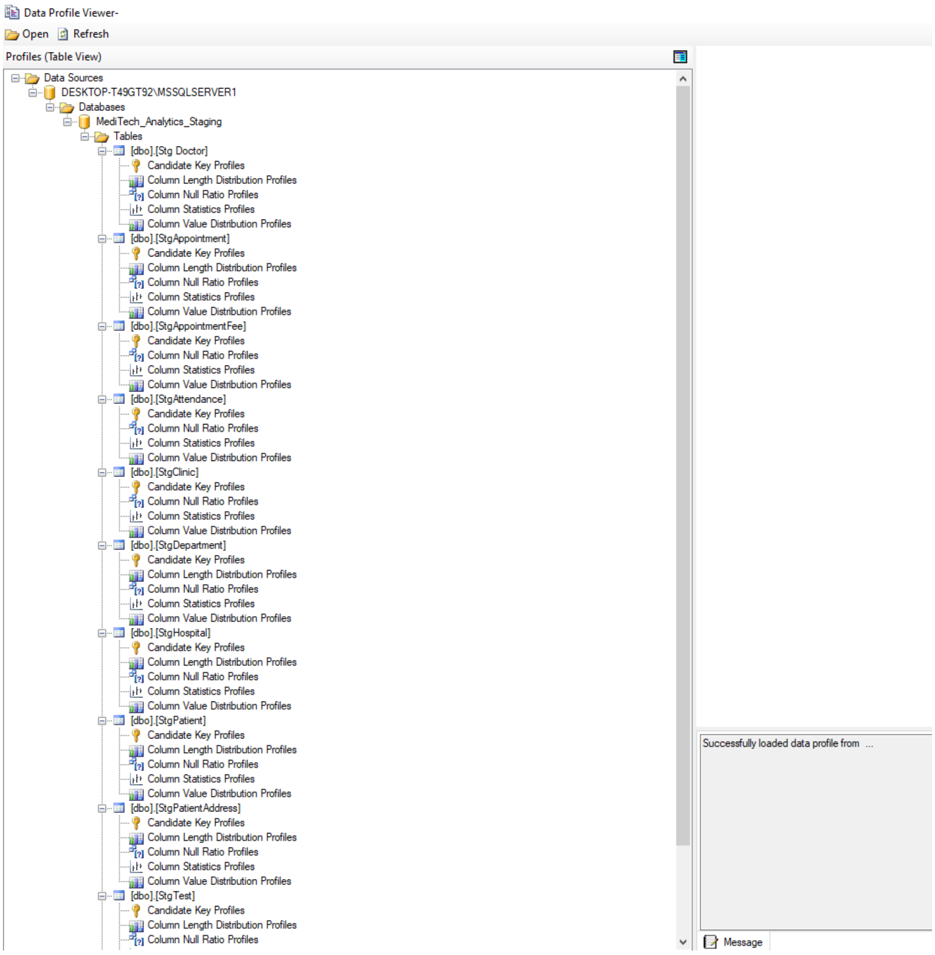
Executable: Extact Clinic Data to Staging

Event handler: OnPreExecute

- **Data profiling**

Used Data_Profiling package to profiling the staging tables





```
Data Sources
  DESKTOP-T49GT92\MSSQLSERVER1
    Databases
      MediTech_Analytics_Staging
        Tables
          [dbo].[Stg Doctor]
            Candidate Key Profiles
            Column Length Distribution Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgAppointment]
            Candidate Key Profiles
            Column Length Distribution Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgAppointmentFee]
            Candidate Key Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgAttendance]
            Candidate Key Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgClinic]
            Candidate Key Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgDepartment]
            Candidate Key Profiles
            Column Length Distribution Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgHospital]
            Candidate Key Profiles
            Column Length Distribution Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgPatient]
            Candidate Key Profiles
            Column Length Distribution Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgPatientAddress]
            Candidate Key Profiles
            Column Length Distribution Profiles
            Column Null Ratio Profiles
            Column Statistics Profiles
            Column Value Distribution Profiles
          [dbo].[StgTest]
            Candidate Key Profiles
            Column Length Distribution Profiles
            Column Null Ratio Profiles
```

```
Successfully loaded data profile from  ...
```

3.Transform & Load

In this step, both the ‘Transform’ and ‘Load’ are done. Firstly, The Dimension tables in the Datawarehouse DB data were created. Then, using the relevant components, data from the staging tables was loaded into the warehouse tables, MediTech_Analytics, which contains the below tables,

1. DimHospital
2. DimClinic
3. DimPatient
4. DimAppointment
5. DimDepartment
6. DimTest
7. DimDoctor
8. AttendanceFact

Used Transformation Tasks

1. Lookups

   Tests’ Department ID is looked when loading to the DimTest table using Department table.
   DimPatient’s PatientID is looked when loading using DimAttendance table.

   .

2. Derived Columns

   Derived column is used in FactAttendance to derive both StartDate and EndDate by using GETDATE() expression and to derive the Total Amount too.

3. Union

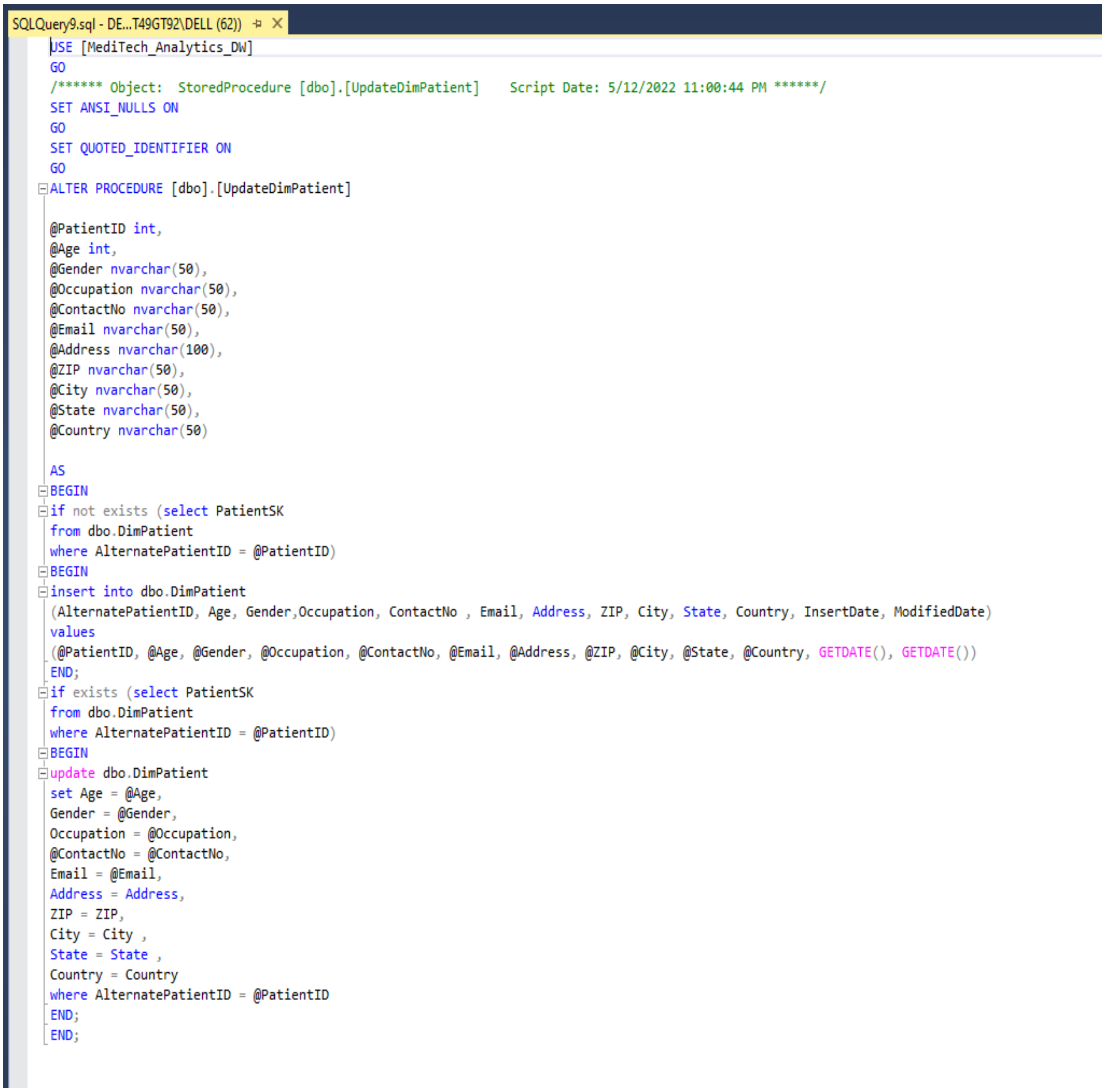
   Union is used in the Extract step to combine and get all the data from data files.

4. Sort and Merge

   ‘Sort’ is used sort out the Patient and Appointment data and they are merged ‘Merge’ using PatientID.

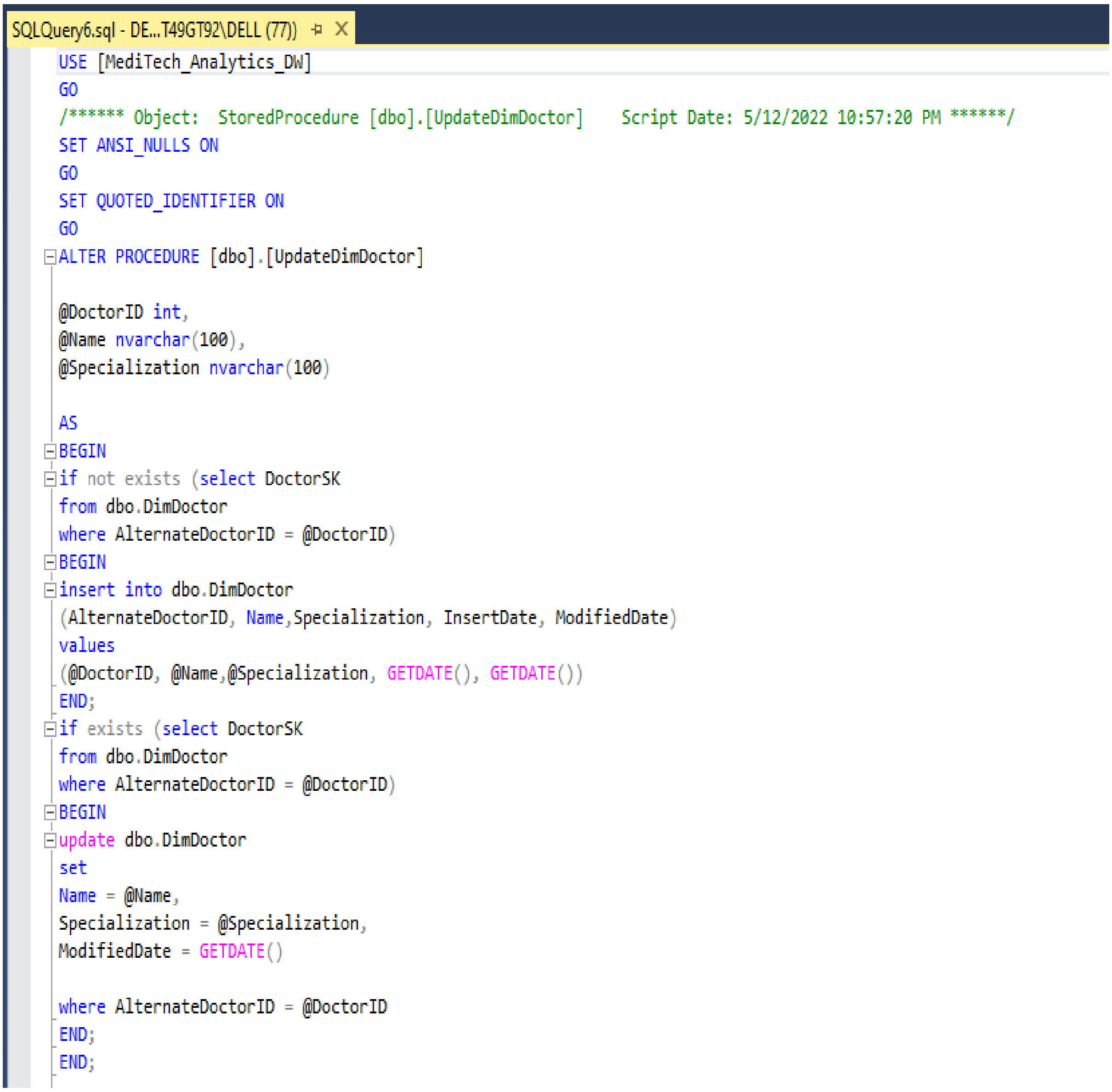
Update Functions

- DimPatient



```sql
USE [MediTech_Analytics_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDimPatient]    Script Date: 5/12/2022 11:00:44 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimPatient]

@PatientID int,
@Age int,
@Gender nvarchar(50),
@Occupation nvarchar(50),
@ContactNo nvarchar(50),
@Email nvarchar(50),
@Address nvarchar(100),
@ZIP nvarchar(50),
@City nvarchar(50),
@State nvarchar(50),
@Country nvarchar(50)

AS
BEGIN
if not exists (select PatientSK
from dbo.DimPatient
where AlternatePatientID = @PatientID)
BEGIN
insert into dbo.DimPatient
(AlternatePatientID, Age, Gender,Occupation, ContactNo , Email, Address, ZIP, City, State, Country, InsertDate, ModifiedDate)
values
(@PatientID, @Age, @Gender, @Occupation, @ContactNo, @Email, @Address, @ZIP, @City, @State, @Country, GETDATE(), GETDATE())
END;
if exists (select PatientSK
from dbo.DimPatient
where AlternatePatientID = @PatientID)
BEGIN
update dbo.DimPatient
set Age = @Age,
Gender = @Gender,
Occupation = @Occupation,
@ContactNo = @ContactNo,
Email = @Email,
Address = Address,
ZIP = ZIP,
City = City ,
State = State ,
Country = Country
where AlternatePatientID = @PatientID
END;
END;
```

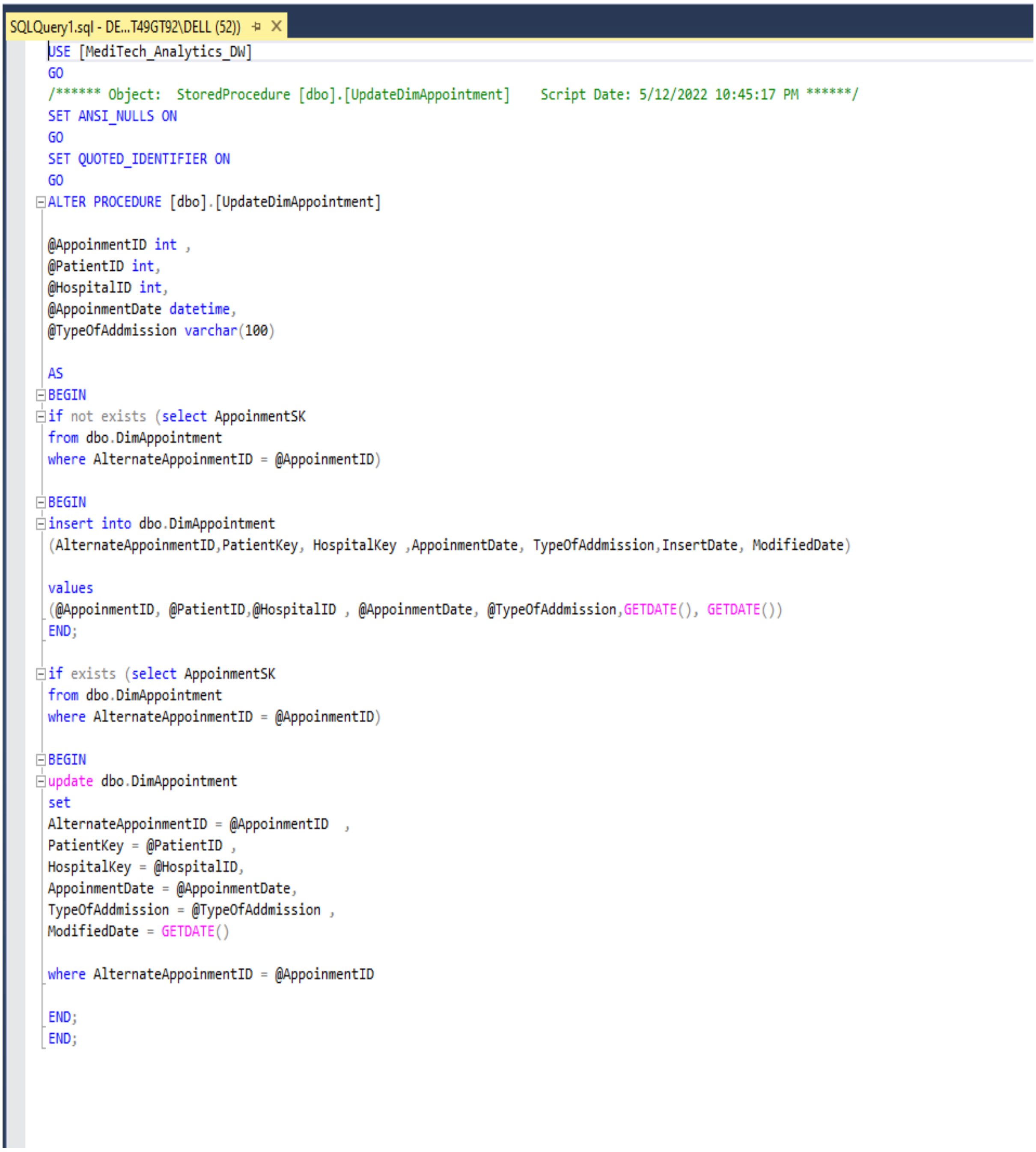
- DimDoctor



SQLQuery6.sql - DE...T49GT92\DELL (77))

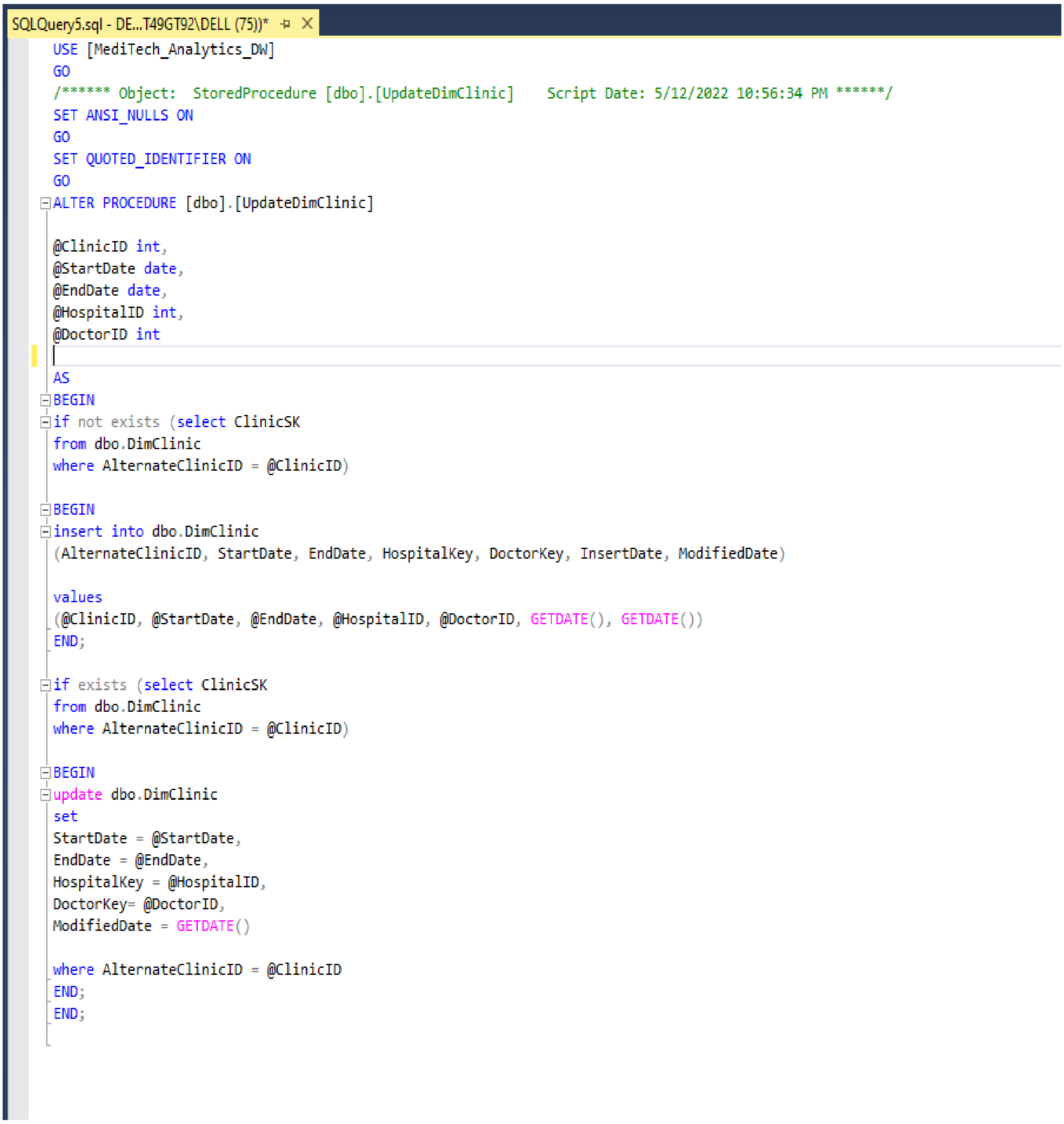
```
USE [MediTech_Analytics_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDimDoctor]    Script Date: 5/12/2022 10:57:20 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimDoctor]

@DoctorID int,
@Name nvarchar(100),
@Specialization nvarchar(100)

AS
BEGIN
if not exists (select DoctorSK
from dbo.DimDoctor
where AlternateDoctorID = @DoctorID)
BEGIN
insert into dbo.DimDoctor
(AlternateDoctorID, Name,Specialization, InsertDate, ModifiedDate)
values
(@DoctorID, @Name,@Specialization, GETDATE(), GETDATE())
END;
if exists (select DoctorSK
from dbo.DimDoctor
where AlternateDoctorID = @DoctorID)
BEGIN
update dbo.DimDoctor
set
Name = @Name,
Specialization = @Specialization,
ModifiedDate = GETDATE()

where AlternateDoctorID = @DoctorID
END;
END;
```

- DimAppointment



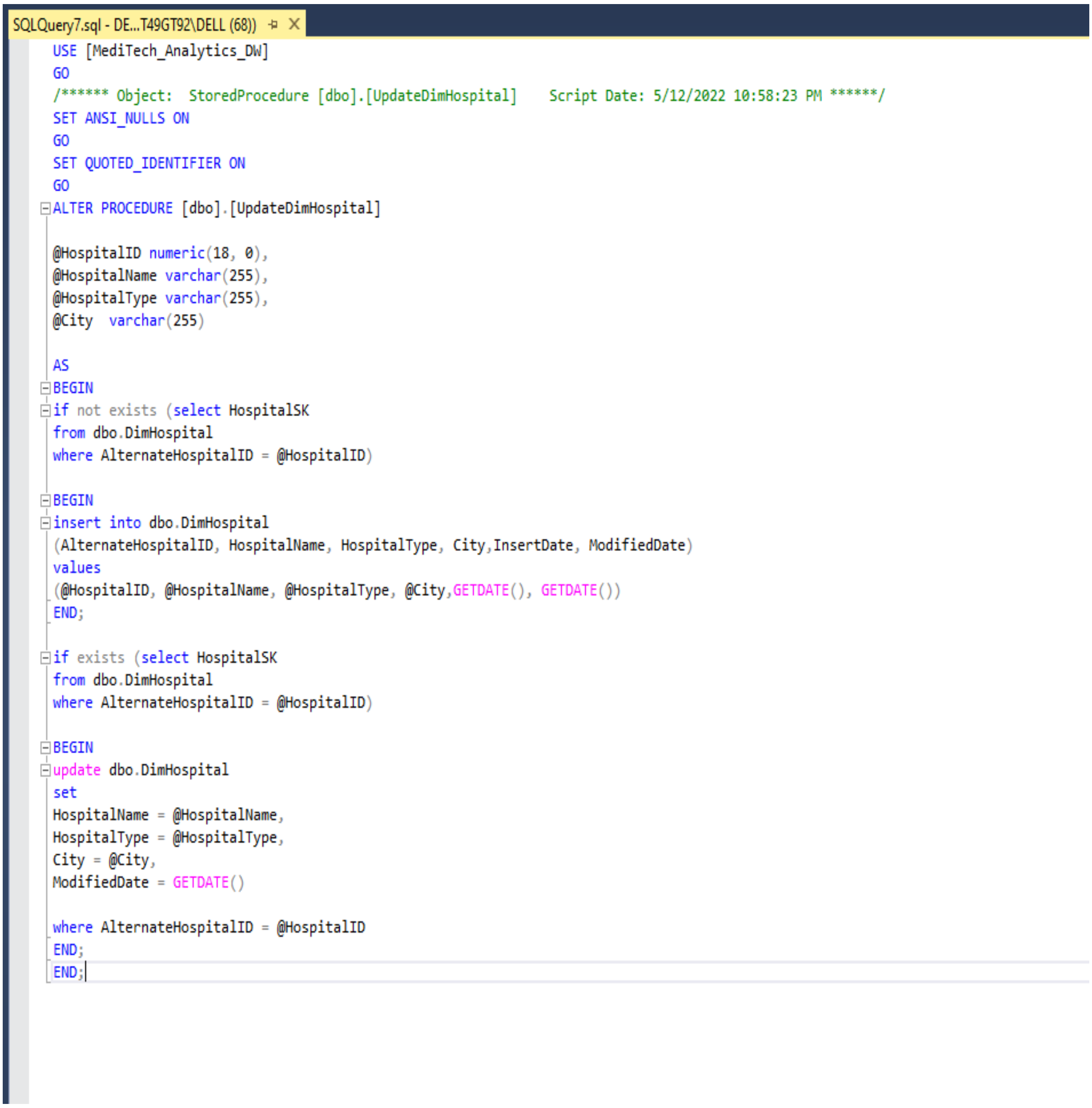
```
USE [MediTech_Analytics_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDimAppointment]    Script Date: 5/12/2022 10:45:17 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimAppointment]

@AppoinmentID int ,
@PatientID int,
@HospitalID int,
@AppoinmentDate datetime,
@TypeOfAddmission varchar(100)

AS
BEGIN
if not exists (select AppoinmentSK
from dbo.DimAppointment
where AlternateAppoinmentID = @AppoinmentID)

BEGIN
insert into dbo.DimAppointment
(AlternateAppoinmentID,PatientKey, HospitalKey ,AppoinmentDate, TypeOfAddmission,InsertDate, ModifiedDate)

values
(@AppoinmentID, @PatientID,@HospitalID , @AppoinmentDate, @TypeOfAddmission,GETDATE(), GETDATE())
END;

if exists (select AppoinmentSK
from dbo.DimAppointment
where AlternateAppoinmentID = @AppoinmentID)

BEGIN
update dbo.DimAppointment
set
AlternateAppoinmentID = @AppoinmentID  ,
PatientKey = @PatientID ,
HospitalKey = @HospitalID,
AppoinmentDate = @AppoinmentDate,
TypeOfAddmission = @TypeOfAddmission ,
ModifiedDate = GETDATE()

where AlternateAppoinmentID = @AppoinmentID

END;
END;
```

- DimClinic



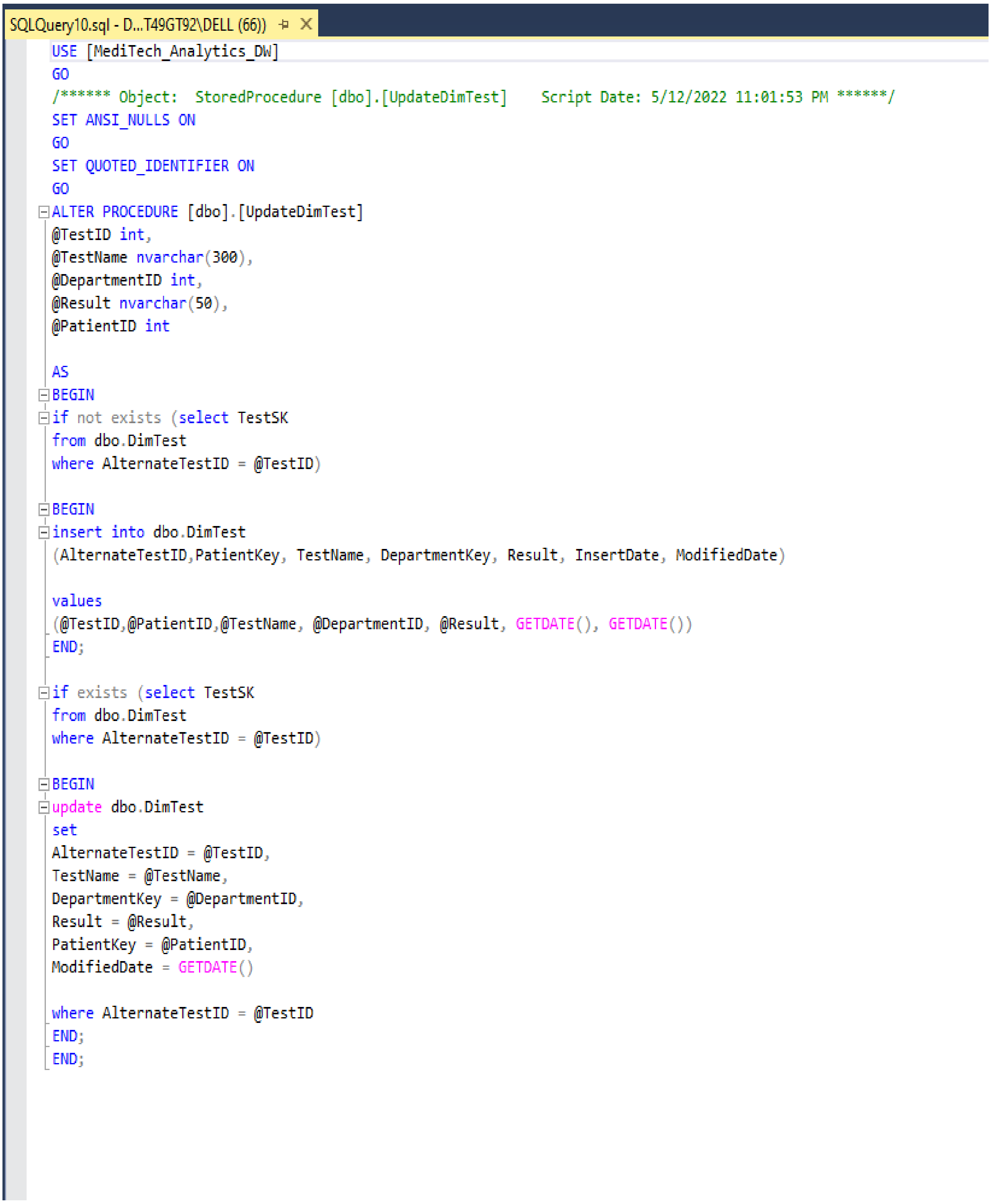
```
USE [MediTech_Analytics_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDimClinic]    Script Date: 5/12/2022 10:56:34 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimClinic]

@ClinicID int,
@StartDate date,
@EndDate date,
@HospitalID int,
@DoctorID int

AS
BEGIN
if not exists (select ClinicSK
from dbo.DimClinic
where AlternateClinicID = @ClinicID)

BEGIN
insert into dbo.DimClinic
(AlternateClinicID, StartDate, EndDate, HospitalKey, DoctorKey, InsertDate, ModifiedDate)

values
(@ClinicID, @StartDate, @EndDate, @HospitalID, @DoctorID, GETDATE(), GETDATE())
END;

if exists (select ClinicSK
from dbo.DimClinic
where AlternateClinicID = @ClinicID)

BEGIN
update dbo.DimClinic
set
StartDate = @StartDate,
EndDate = @EndDate,
HospitalKey = @HospitalID,
DoctorKey= @DoctorID,
ModifiedDate = GETDATE()

where AlternateClinicID = @ClinicID
END;
END;
```

- DimHospital



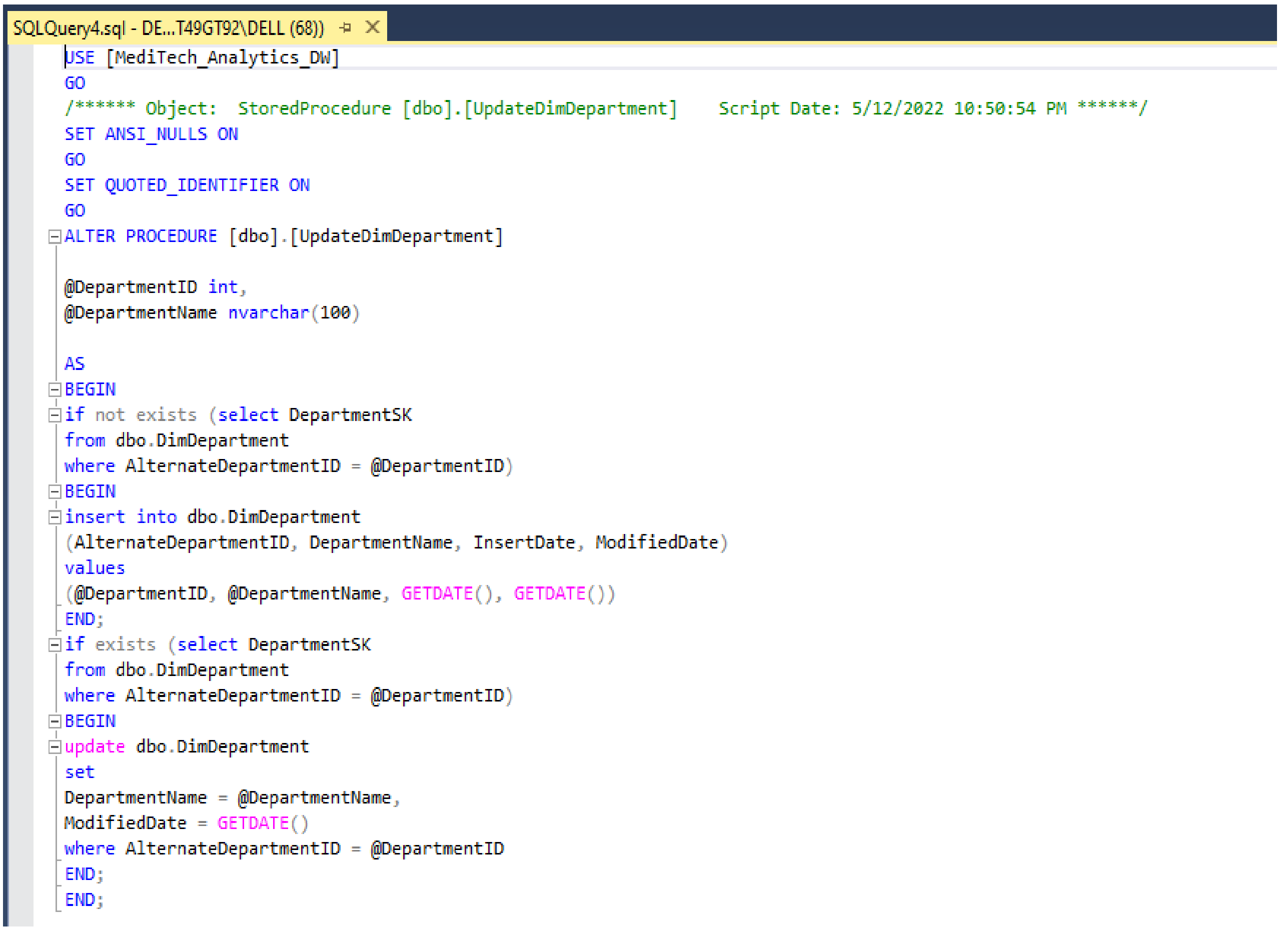
```
USE [MediTech_Analytics_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDimHospital]    Script Date: 5/12/2022 10:58:23 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimHospital]

@HospitalID numeric(18, 0),
@HospitalName varchar(255),
@HospitalType varchar(255),
@City  varchar(255)

AS
BEGIN
if not exists (select HospitalSK
from dbo.DimHospital
where AlternateHospitalID = @HospitalID)

BEGIN
insert into dbo.DimHospital
(AlternateHospitalID, HospitalName, HospitalType, City,InsertDate, ModifiedDate)
values
(@HospitalID, @HospitalName, @HospitalType, @City,GETDATE(), GETDATE())
END;

if exists (select HospitalSK
from dbo.DimHospital
where AlternateHospitalID = @HospitalID)

BEGIN
update dbo.DimHospital
set
HospitalName = @HospitalName,
HospitalType = @HospitalType,
City = @City,
ModifiedDate = GETDATE()

where AlternateHospitalID = @HospitalID
END;
END;
```

- DimTest



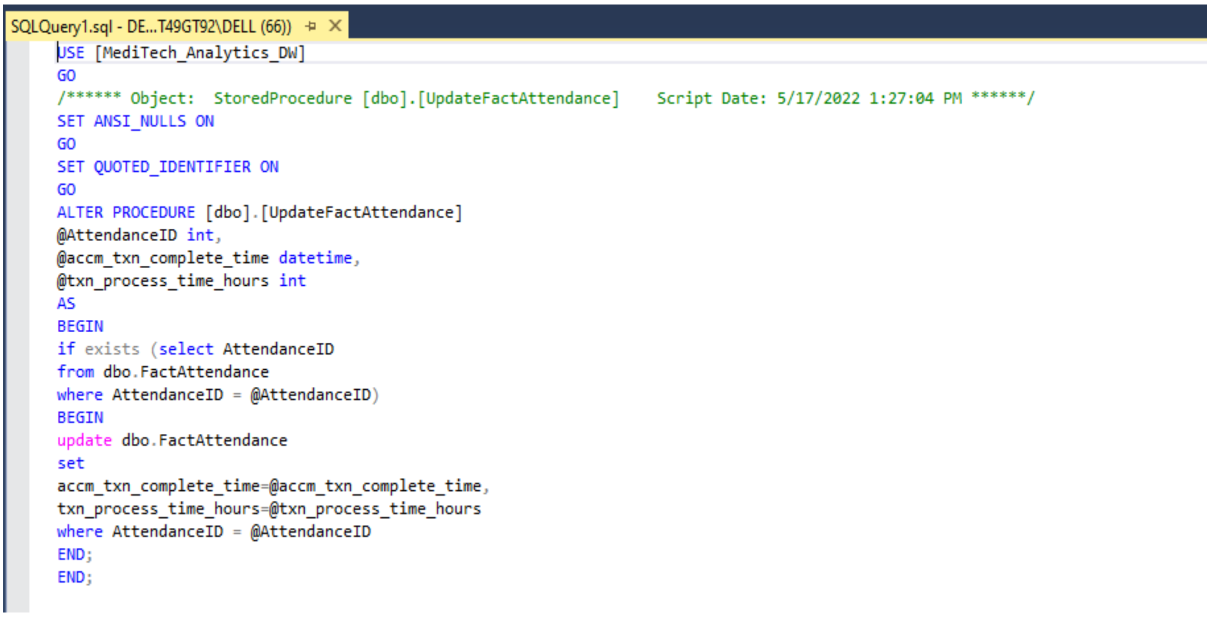
```sql
USE [MediTech_Analytics_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDimTest]    Script Date: 5/12/2022 11:01:53 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimTest]
@TestID int,
@TestName nvarchar(300),
@DepartmentID int,
@Result nvarchar(50),
@PatientID int

AS
BEGIN
if not exists (select TestSK
from dbo.DimTest
where AlternateTestID = @TestID)

BEGIN
insert into dbo.DimTest
(AlternateTestID,PatientKey, TestName, DepartmentKey, Result, InsertDate, ModifiedDate)

values
(@TestID,@PatientID,@TestName, @DepartmentID, @Result, GETDATE(), GETDATE())
END;

if exists (select TestSK
from dbo.DimTest
where AlternateTestID = @TestID)

BEGIN
update dbo.DimTest
set
AlternateTestID = @TestID,
TestName = @TestName,
DepartmentKey = @DepartmentID,
Result = @Result,
PatientKey = @PatientID,
ModifiedDate = GETDATE()

where AlternateTestID = @TestID
END;
END;
```

- DimDepartment



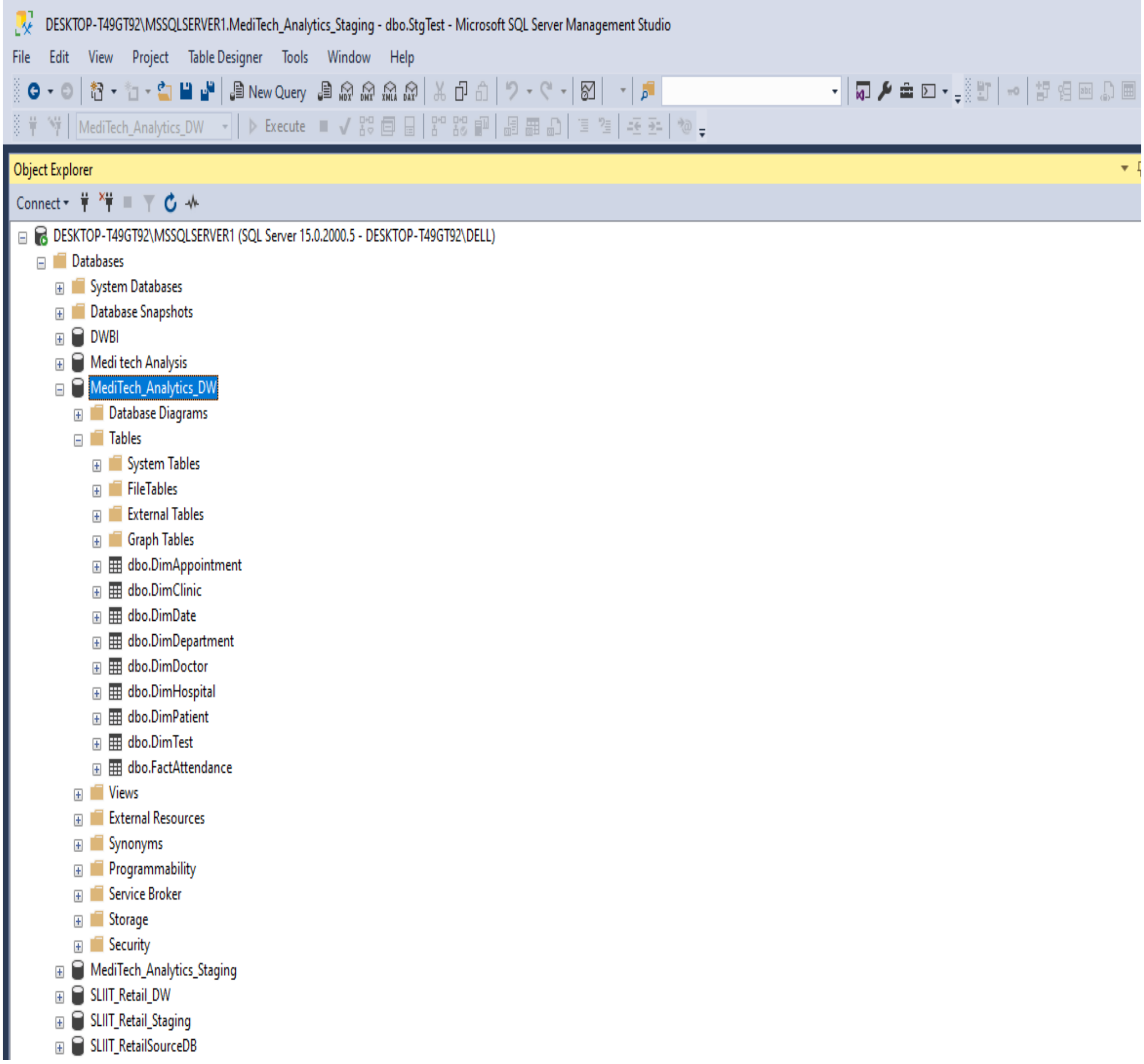
```
USE [MediTech_Analytics_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDimDepartment]    Script Date: 5/12/2022 10:50:54 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimDepartment]

@DepartmentID int,
@DepartmentName nvarchar(100)

AS
BEGIN
if not exists (select DepartmentSK
from dbo.DimDepartment
where AlternateDepartmentID = @DepartmentID)
BEGIN
insert into dbo.DimDepartment
(AlternateDepartmentID, DepartmentName, InsertDate, ModifiedDate)
values
(@DepartmentID, @DepartmentName, GETDATE(), GETDATE())
END;
if exists (select DepartmentSK
from dbo.DimDepartment
where AlternateDepartmentID = @DepartmentID)
BEGIN
update dbo.DimDepartment
set
DepartmentName = @DepartmentName,
ModifiedDate = GETDATE()
where AlternateDepartmentID = @DepartmentID
END;
END;
```

- DimFactAttendance



```
USE [MediTech_Analytics_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateFactAttendance]    Script Date: 5/17/2022 1:27:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateFactAttendance]
@AttendanceID int,
@accm_txn_complete_time datetime,
@txn_process_time_hours int
AS
BEGIN
if exists (select AttendanceID
from dbo.FactAttendance
where AttendanceID = @AttendanceID)
BEGIN
update dbo.FactAttendance
set
accm_txn_complete_time=@accm_txn_complete_time,
txn_process_time_hours=@txn_process_time_hours
where AttendanceID = @AttendanceID
END;
END;
```

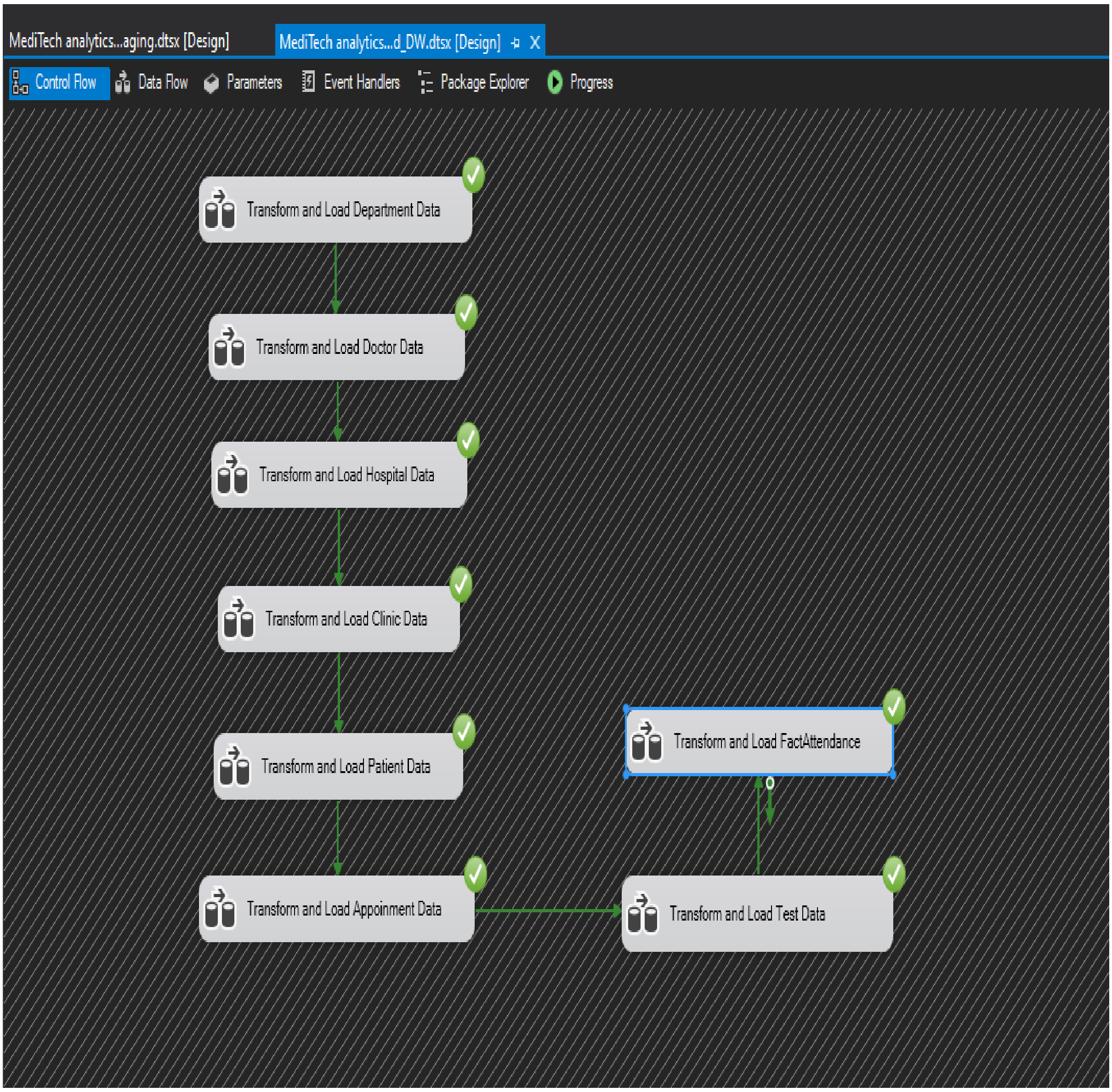
- **Snapshot of SQL server Data warehouse Database**



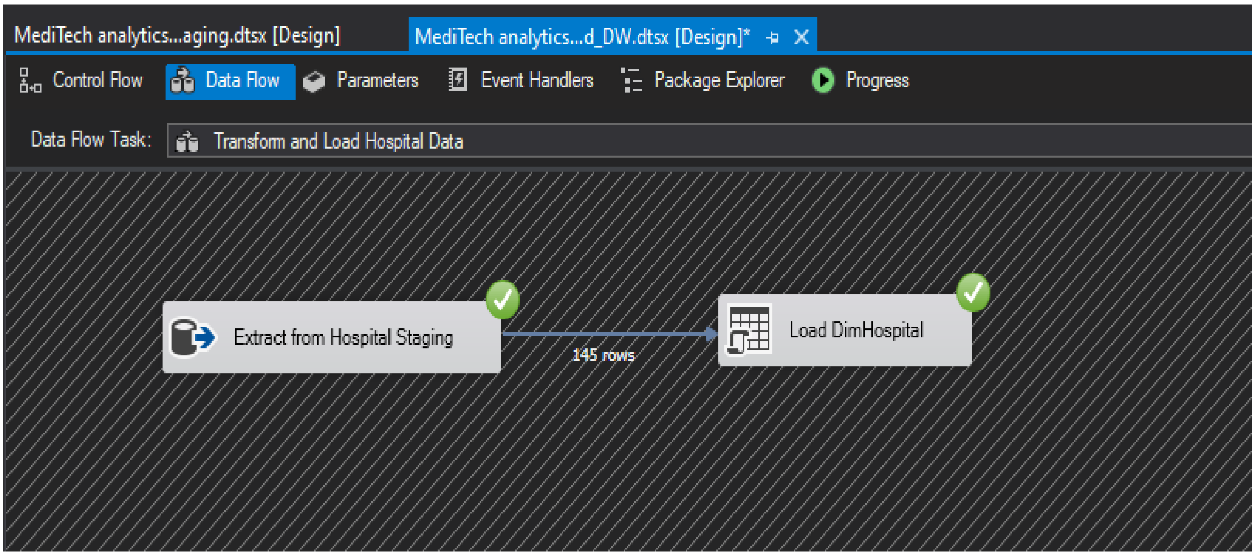
DESKTOP-T49GT92\MSSQLSERVER1.MediTech_Analytics_Staging - dbo.StgTest - Microsoft SQL Server Management Studio

```
DESKTOP-T49GT92\MSSQLSERVER1 (SQL Server 15.0.2000.5 - DESKTOP-T49GT92\DELL)
  Databases
    System Databases
    Database Snapshots
    DWBI
    Medi tech Analysis
    MediTech_Analytics_DW
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.DimAppointment
        dbo.DimClinic
        dbo.DimDate
        dbo.DimDepartment
        dbo.DimDoctor
        dbo.DimHospital
        dbo.DimPatient
        dbo.DimTest
        dbo.FactAttendance
      Views
      External Resources
      Synonyms
      Programmability
      Service Broker
      Storage
      Security
    MediTech_Analytics_Staging
    SLIIT_Retail_DW
    SLIIT_Retail_Staging
    SLIIT_RetailSourceDB
```

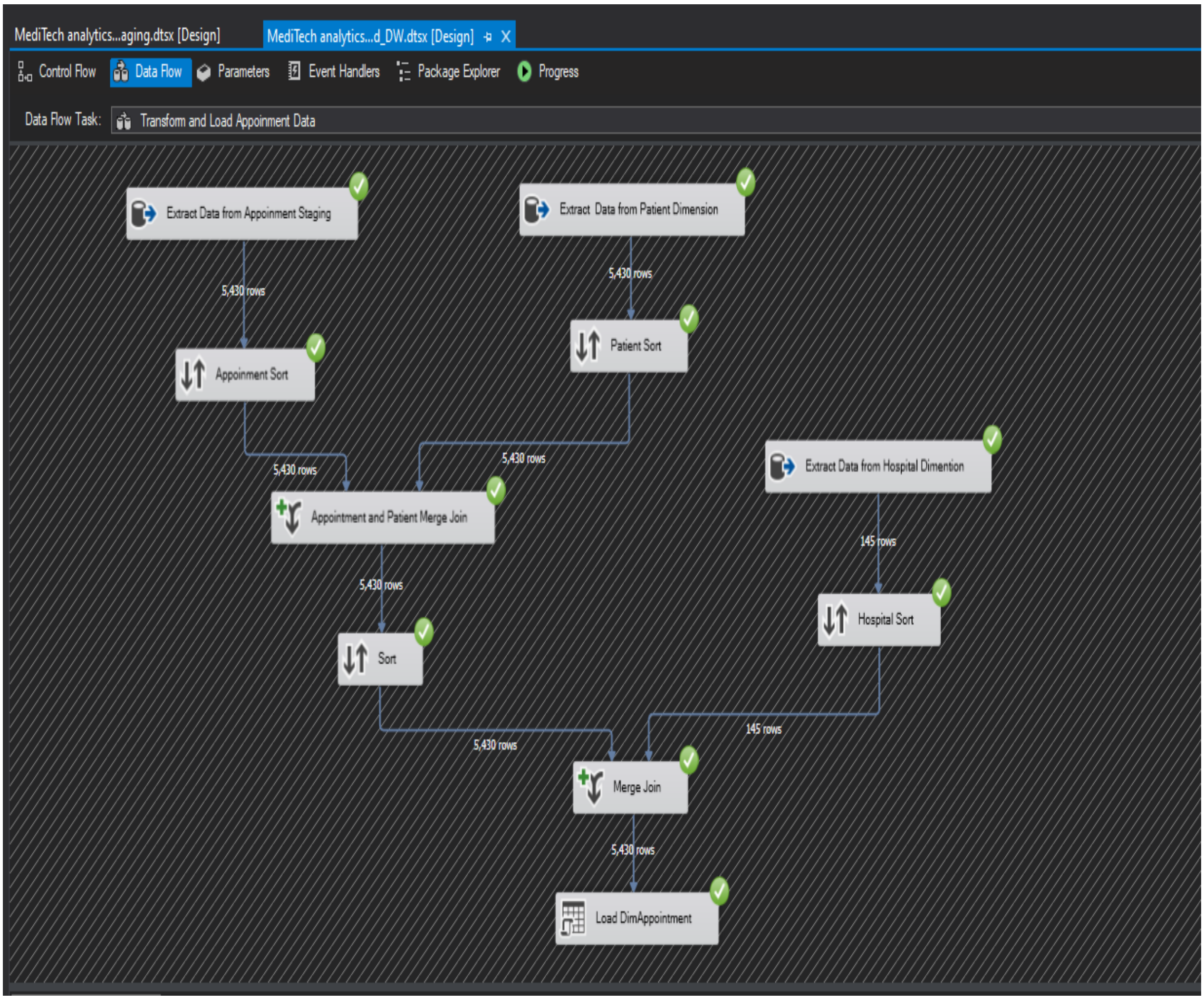
Snapshot of Visual Studio Control Flow of Extraction


MediTech analytics...aging.dtsx [Design]
MediTech analytics...d_DW.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Transform and Load Department Data
Transform and Load Doctor Data
Transform and Load Hospital Data
Transform and Load Clinic Data
Transform and Load Patient Data
Transform and Load Appoinment Data
Transform and Load FactAttendance
Transform and Load Test Data

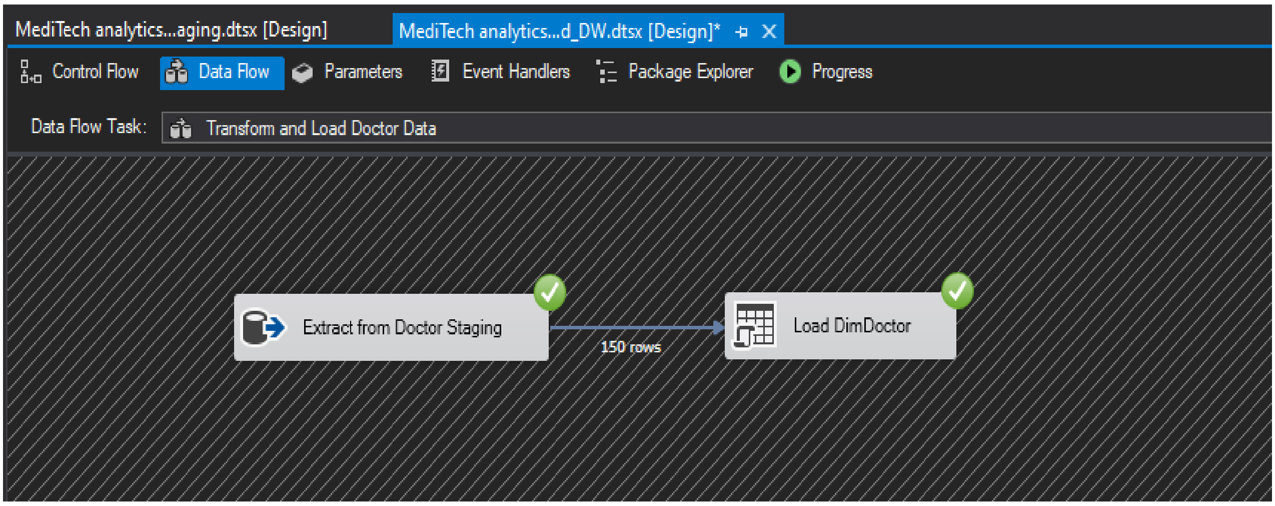
- **Hospital Data Transform and Load**


MediTech analytics...aging.dtsx [Design]
MediTech analytics...d_DW.dtsx [Design]*
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Transform and Load Hospital Data
Extract from Hospital Staging
145 rows
Load DimHospital

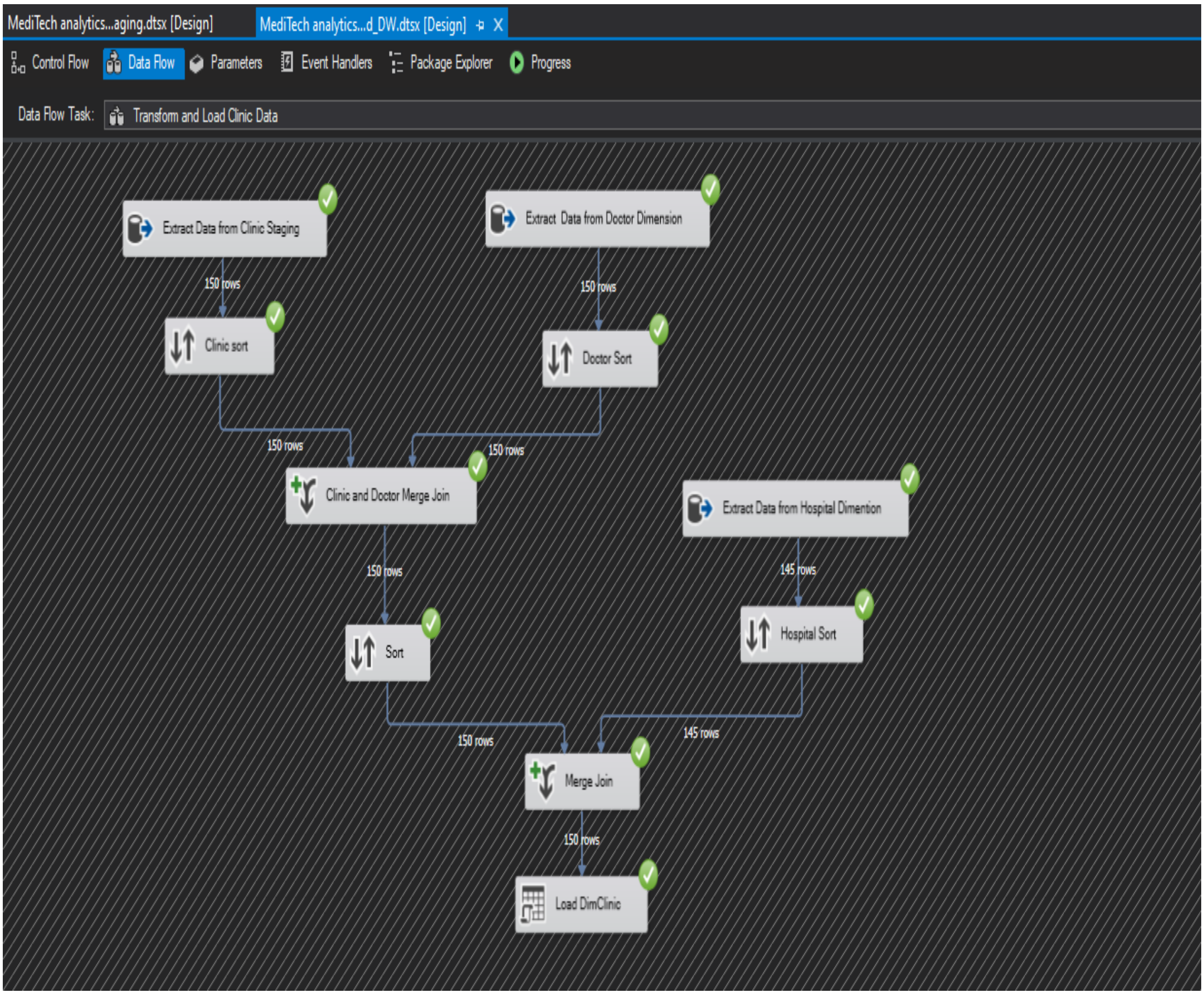
- **Appointment Data Transform and Load**


MediTech analytics...aging.dtsx [Design]
MediTech analytics...d_DW.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Transform and Load Appoinment Data
Extract Data from Appoinment Staging
Extract Data from Patient Dimension
5,430 rows
5,430 rows
Appoinment Sort
Patient Sort
5,430 rows
5,430 rows
Extract Data from Hospital Dimention
Appointment and Patient Merge Join
145 rows
5,430 rows
Hospital Sort
Sort
145 rows
5,430 rows
Merge Join
5,430 rows
Load DimAppointment

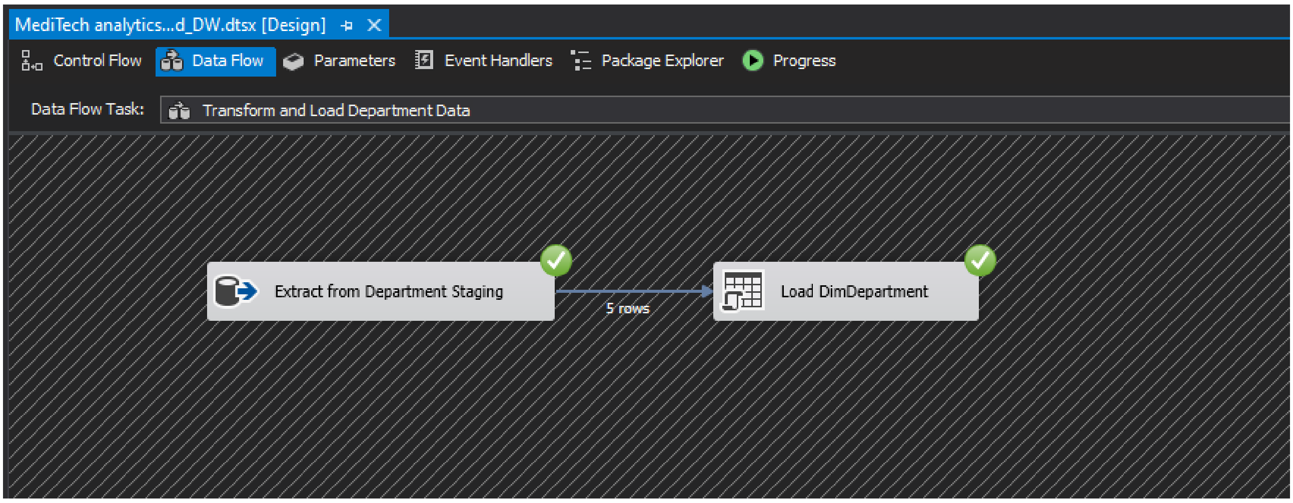
- **Doctor Data Transform and Load**


MediTech analytics...aging.dtsx [Design]
MediTech analytics...d_DW.dtsx [Design]*
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Transform and Load Doctor Data
Extract from Doctor Staging
150 rows
Load DimDoctor

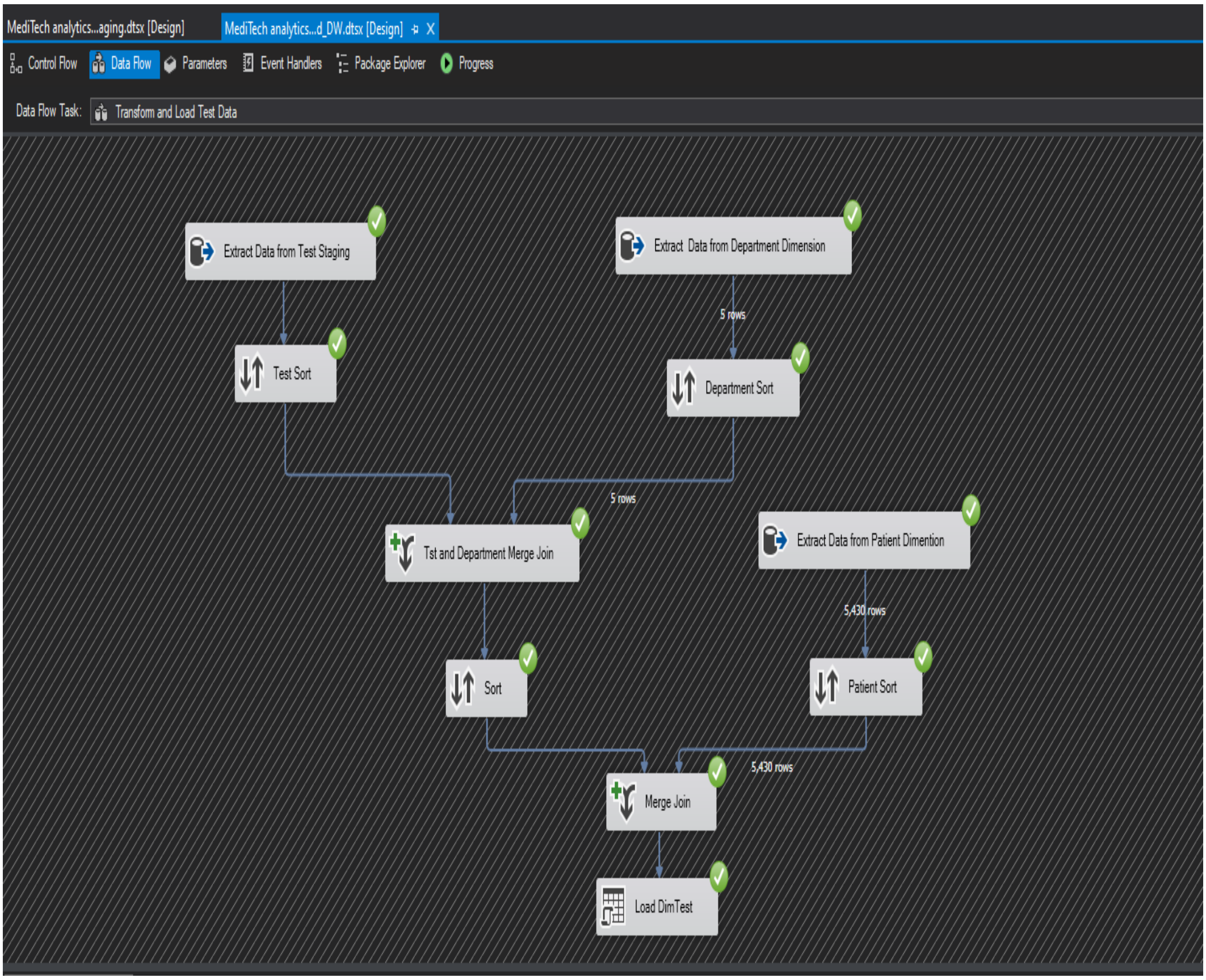
- **Clinic Data Transform and Load**


MediTech analytics...aging.dtsx [Design]
MediTech analytics...d_DW.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Transform and Load Clinic Data
Extract Data from Clinic Staging
150 rows
Clinic sort
Extract Data from Doctor Dimension
150 rows
Doctor Sort
150 rows
150 rows
Clinic and Doctor Merge Join
150 rows
Sort
Extract Data from Hospital Dimention
145 rows
Hospital Sort
150 rows
145 rows
Merge Join
150 rows
Load DimClinic

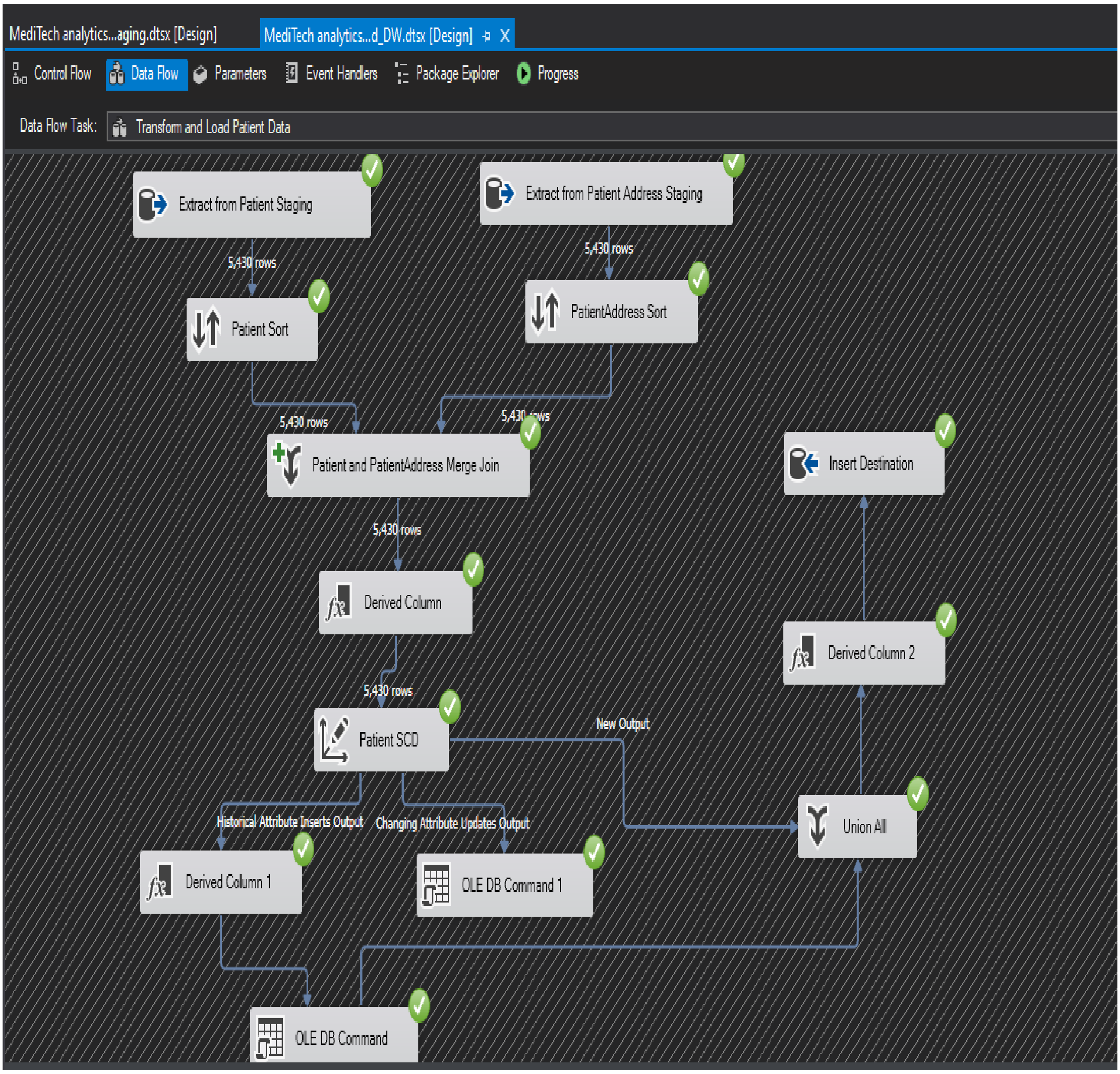
- **Department Data Transform and Load**



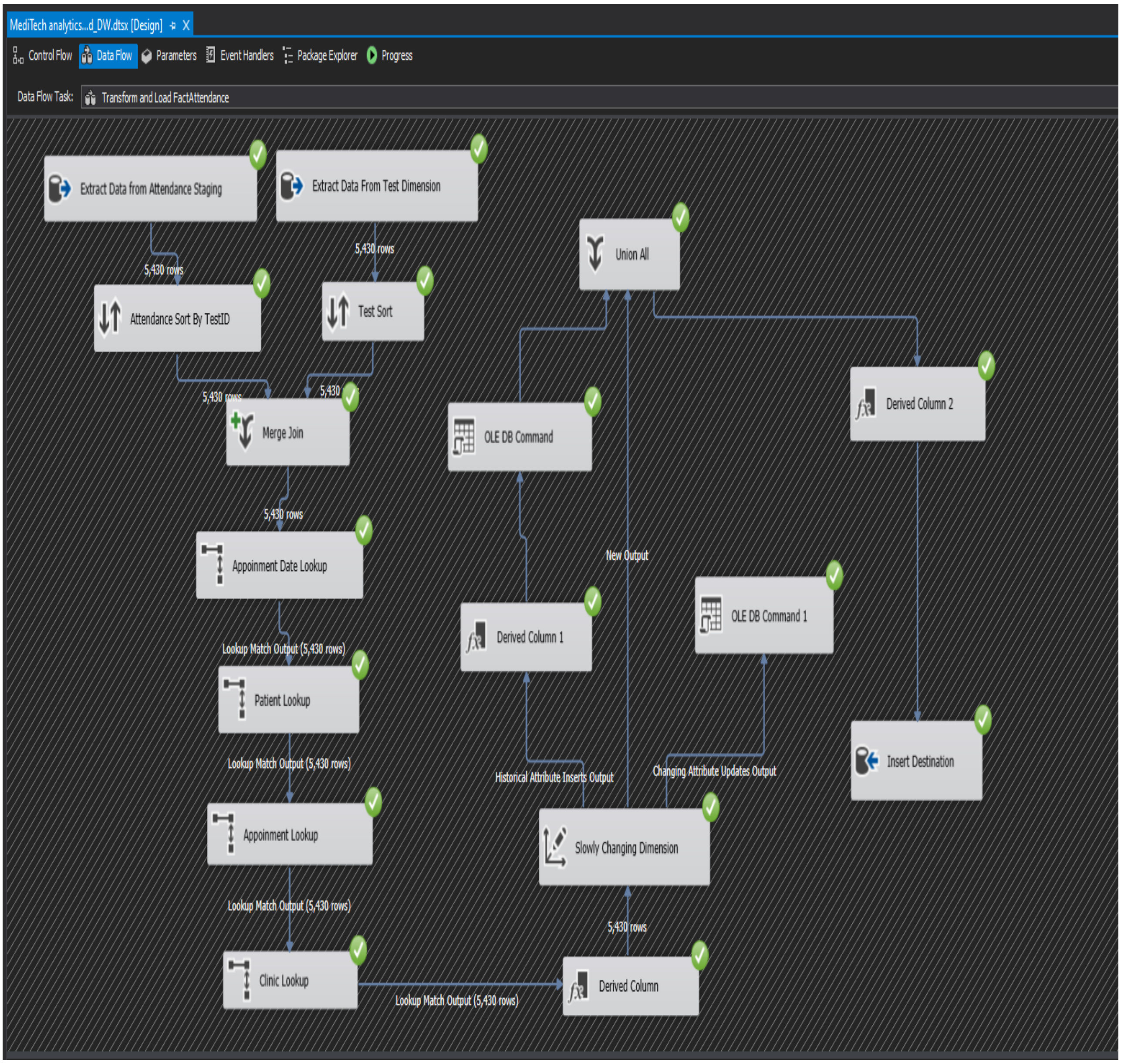
- **Test Data Transform and Load**


MediTech analytics...aging.dtsx [Design]
MediTech analytics...d_DW.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Transform and Load Test Data
Extract Data from Test Staging
Test Sort
Extract Data from Department Dimension
5 rows
Department Sort
5 rows
Tst and Department Merge Join
Extract Data from Patient Dimention
5,430 rows
Sort
Patient Sort
5,430 rows
Merge Join
Load DimTest

- **Patient Data Transform and Load**

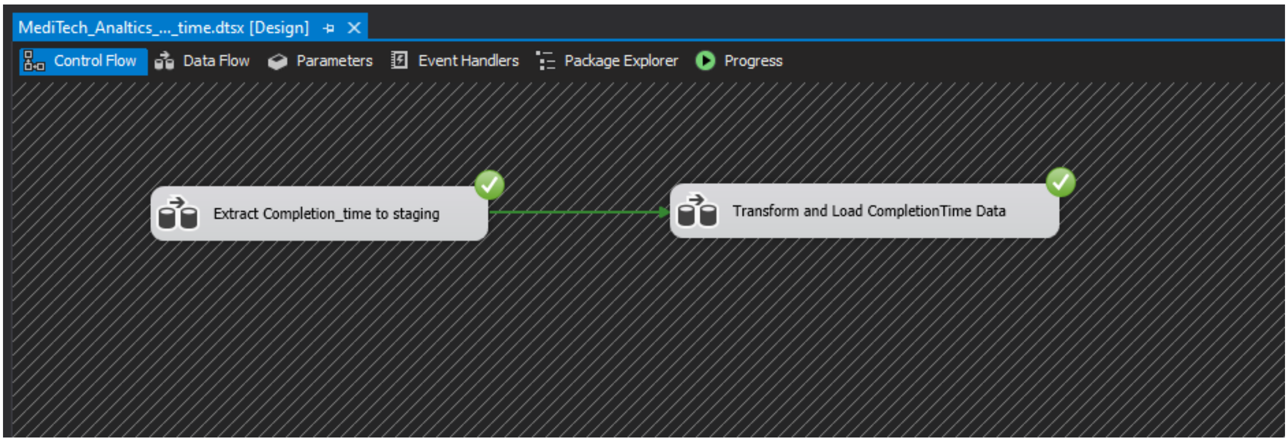


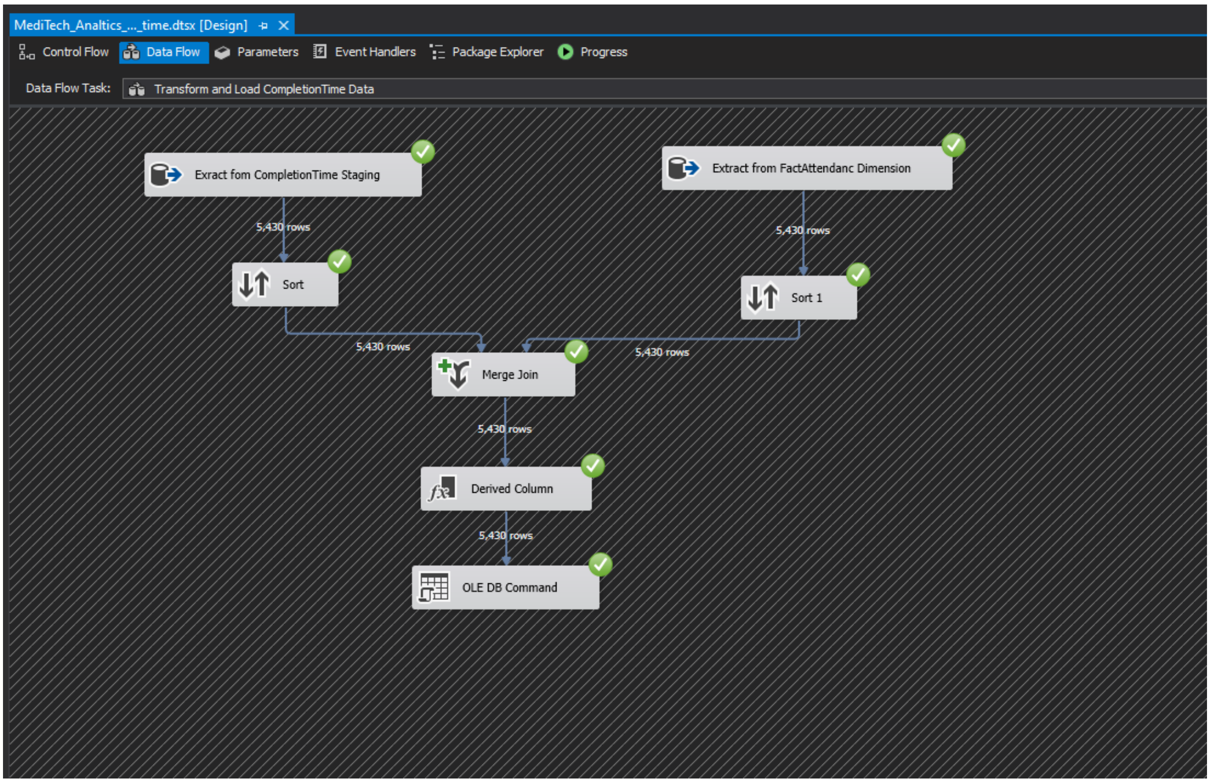
- **FactAttendance Data Transform and Load**

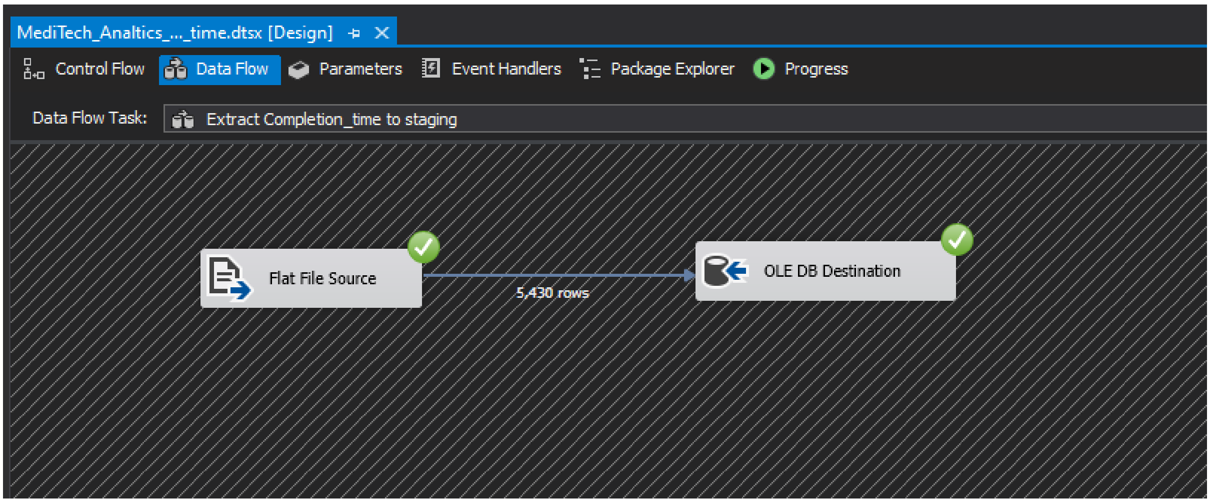

MediTech analytics...d_DW.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Data Flow Task:
Transform and Load FactAttendance
Extract Data from Attendance Staging
Extract Data From Test Dimension
5,430 rows
5,430 rows
Attendance Sort By TestID
Test Sort
5,430 rows
5,430
Merge Join
5,430 rows
Appoinment Date Lookup
Lookup Match Output (5,430 rows)
Patient Lookup
Lookup Match Output (5,430 rows)
Appoinment Lookup
Lookup Match Output (5,430 rows)
Clinic Lookup
Lookup Match Output (5,430 rows)
Derived Column
5,430 rows
Slowly Changing Dimension
Historical Attribute Inserts Output
Changing Attribute Updates Output
New Output
Derived Column 1
OLE DB Command
OLE DB Command 1
Union All
Derived Column 2
Insert Destination

Step 6:

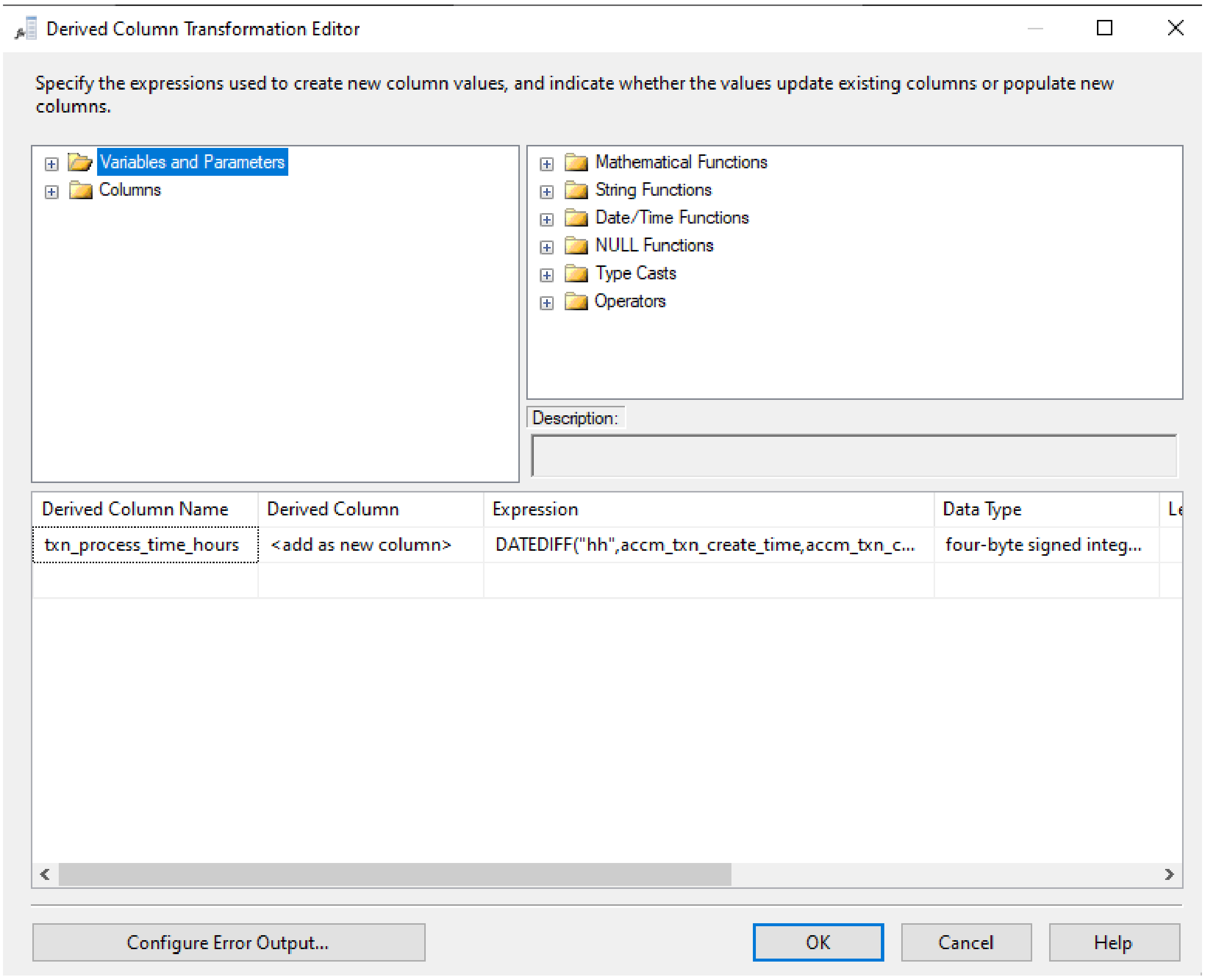
- **Accumulating Fact Table**


MediTech_Analtics_..._time.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Extract Completion_time to staging
Transform and Load CompletionTime Data





Data Flow Task: Extract Completion_time to staging


Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Derived Column Name	Derived Column	Expression	Data Type	Le
txn_process_time_hours	<add as new column>	DATEDIFF("hh",accm_txn_create_time,accm_txn_c...	four-byte signed integ...	