

Sri Lanka Institute of Information Technology

Data warehousing and Business Intelligence

Assignment 1



Student Registration No: IT19083742

Student Name: Vithana K.C.D

Step 1: Data Set Selection

This data set contains Health care analytics data of American hospital health camps. And These camps are organized in several cities with low work-life balance. They reach out to working people and ask them to register for these health camps. For those who attend, health camp provides them the facility to undergo health checks and tests.

These camps have been conducted 150 time over a period of 4 years and the Number of people taking tests at the camps is over 150,000. In the last 4 years, they have stored data of ~ 250,000 tests they have done. Patients need to register beforehand to participate in the health camps.

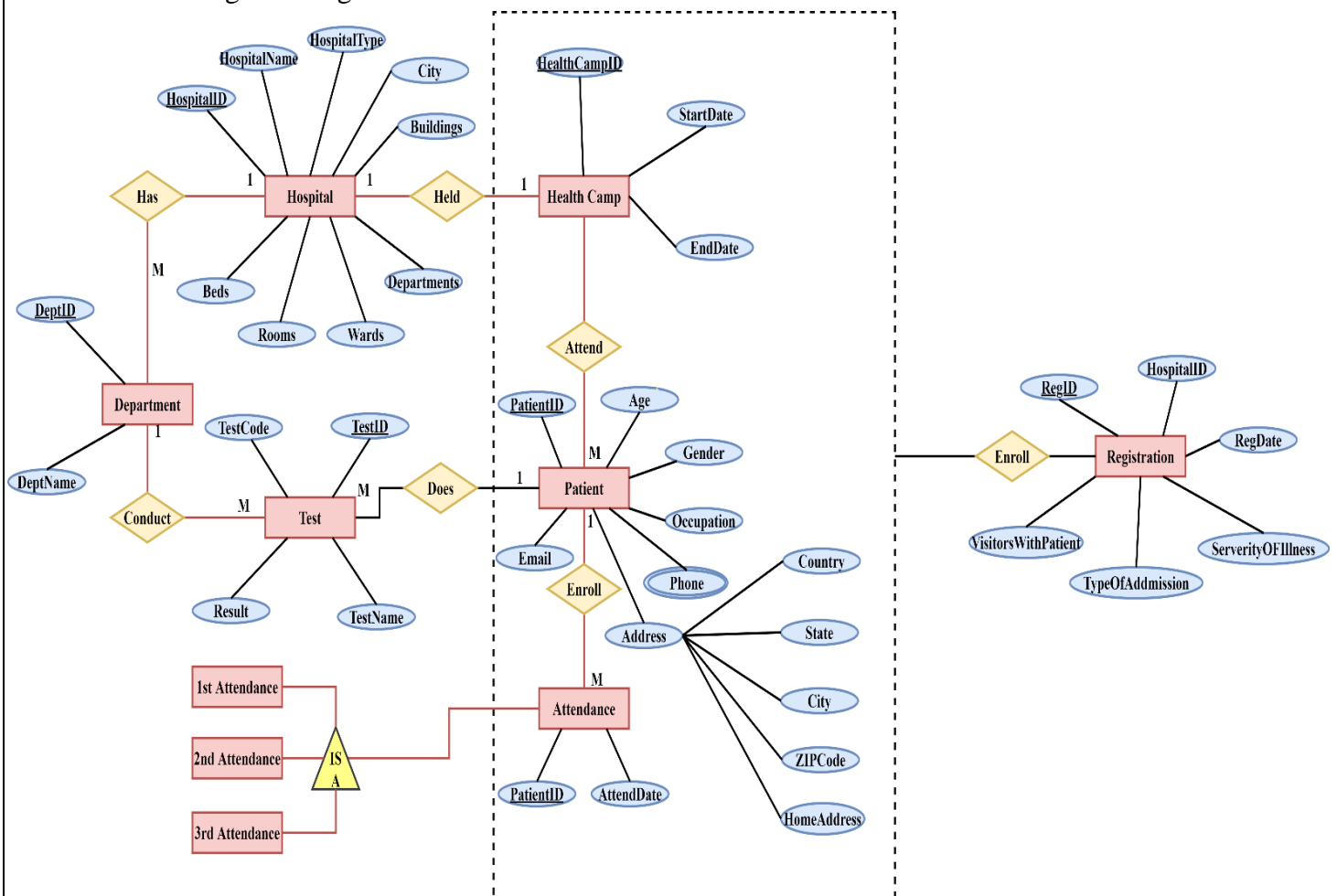
There are three stages of the health camp that each participant must attend. The patients' attendance is methodically maintained. The first and second stage provides people with advice and drugs for any illness they are diagnosed with. The third stage provides participants with direction to do the tests for their diagnosed ailment.

This dataset contains Health camp details,

- Hospital Details
- Patient Details
- Patient Addresses
- Registration Details
- Tests Details
- Health Camp Details
- Department Details
- Attendance Details
- Attendance Fee Details
- Test Prices 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, Excel and csv. And they were used to create the following,

1.Database(.bak)

Patient.txt, RegistrationFee.txt and Attendance.txt file was imported to the Patient Database. Test.txt and TestPrices.txt was imported to the Test Database. And both of those data were used as the DB source data.

2.Text(.txt)

PatientAddress.txt was used directly.

3.Excel(.xls)

HealthCamp.xls and Department.xls was used.

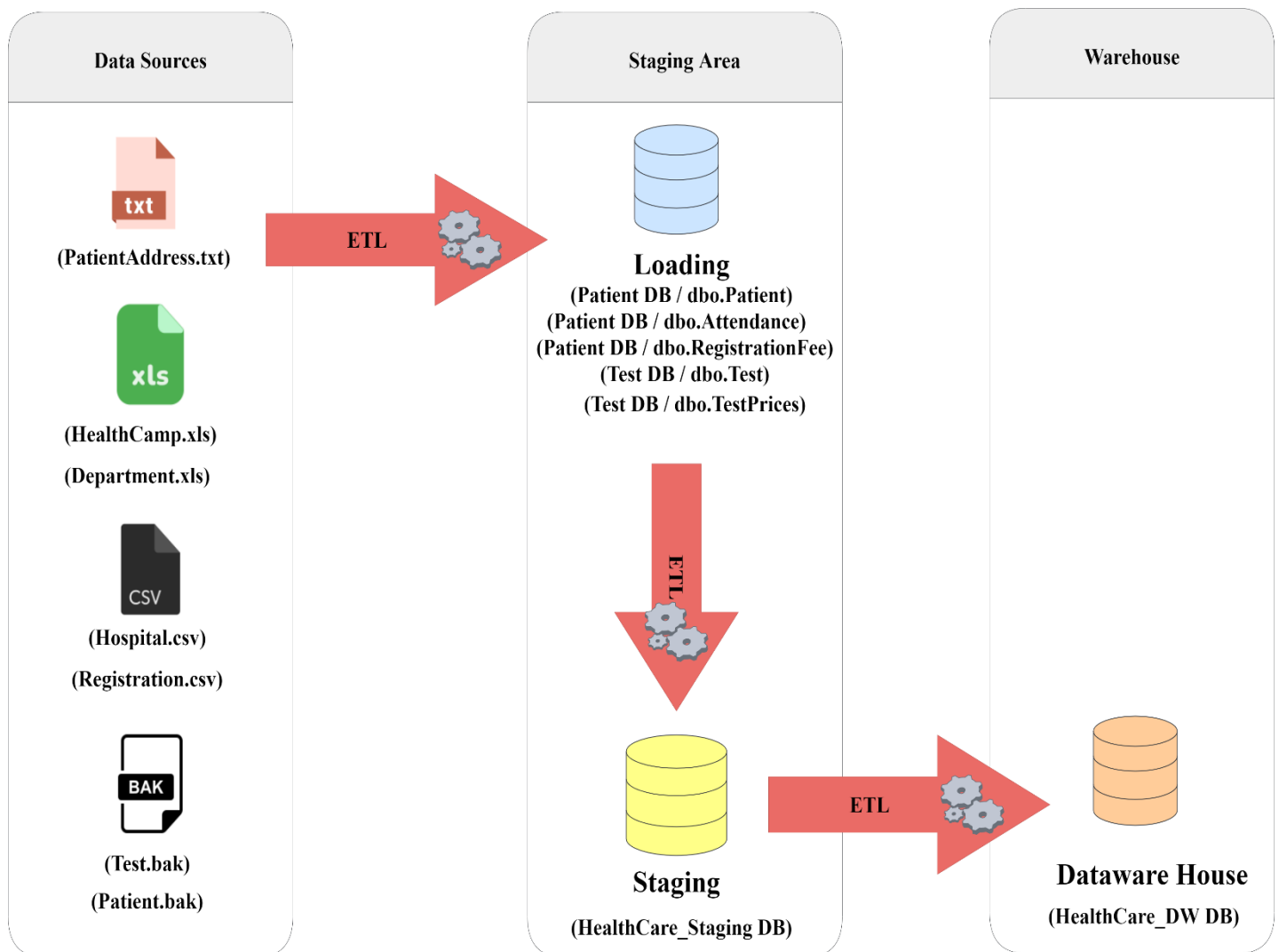
4.Comma Separated Values (.csv)

Hospital.csv and Regisgration.csv was used.

Data Source Type	Source Name	Column Name	Data Type	Description
Database File (.bak)	Dbo.Patient	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
	dbo.Attendance	PatientID	int	PatientID
		1stCampID	int	1 st Health Camp ID
		2ndCampID	int	2 nd Health Camp ID
		3rdCampID	int	3 rd Health Camp ID
		RegDate	datetime	Registration Date
		RegID	int	Registration ID
	dbo.RegistrationFees	PatientID	int	PatientID
		HospitalID	int	HospitalID where patient registered
		RegistrationFee	nvarchar(255)	Registration Fee
	dbo.Test	TestID	int	Unique ID
		PatientID	int	Patient's Unique ID
		TestName	nvarchar(255)	Test Name
		DepartmentID	int	ID of the Department which the test was done
		TestCode	nvarchar(255)	Test Code
		Result	nvarchar(255)	Test Result
	dbo.TestPrices	PatientID	int	PatientID
		TestID	int	Test ID
		TestPrice	nvarchar(255)	Price of Test
Excel File	HealthCamp.xls	HealthCampID		Unique ID
		StartDate		Health Camp Start Date
		EndDate		Health Camp End Date
		HospitalID		Hospital ID which the Health Camp was held
	Department.xls	DepartmentID		Unique ID
		DepartmentName		Department Name
CSV File	Registration.csv	RegisterID	int	Unique ID
		PatientID	int	Patient ID which was used to register in the health camp
		HospitalID	int	Hospital ID which the Health Camp was held
		RegDate	datetime	Registered Date
		TypeOfAddmission	nvarchar(255)	Type of Admission
		SeverityOfIllness	nvarchar(255)	Severity of Illness
		VisitorsWithPatient	int	Visitors with Patient
	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
		Buildings	nvarchar(255)	No of Buildings in the hospital
		Departments	nvarchar(255)	No of departments in the buildings
		Wards	nvarchar(255)	No of wards in the department
		Rooms	nvarchar(255)	No of rooms in the ward
		Beds	nvarchar(255)	No of beds in the room
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, ‘.xls’ component is used to represent Excel 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. Test, Patient, TestPrices, AdmissionFees and Attendance text files was imported to the database and was used to create the tables. And these tables were used as the DB source data.

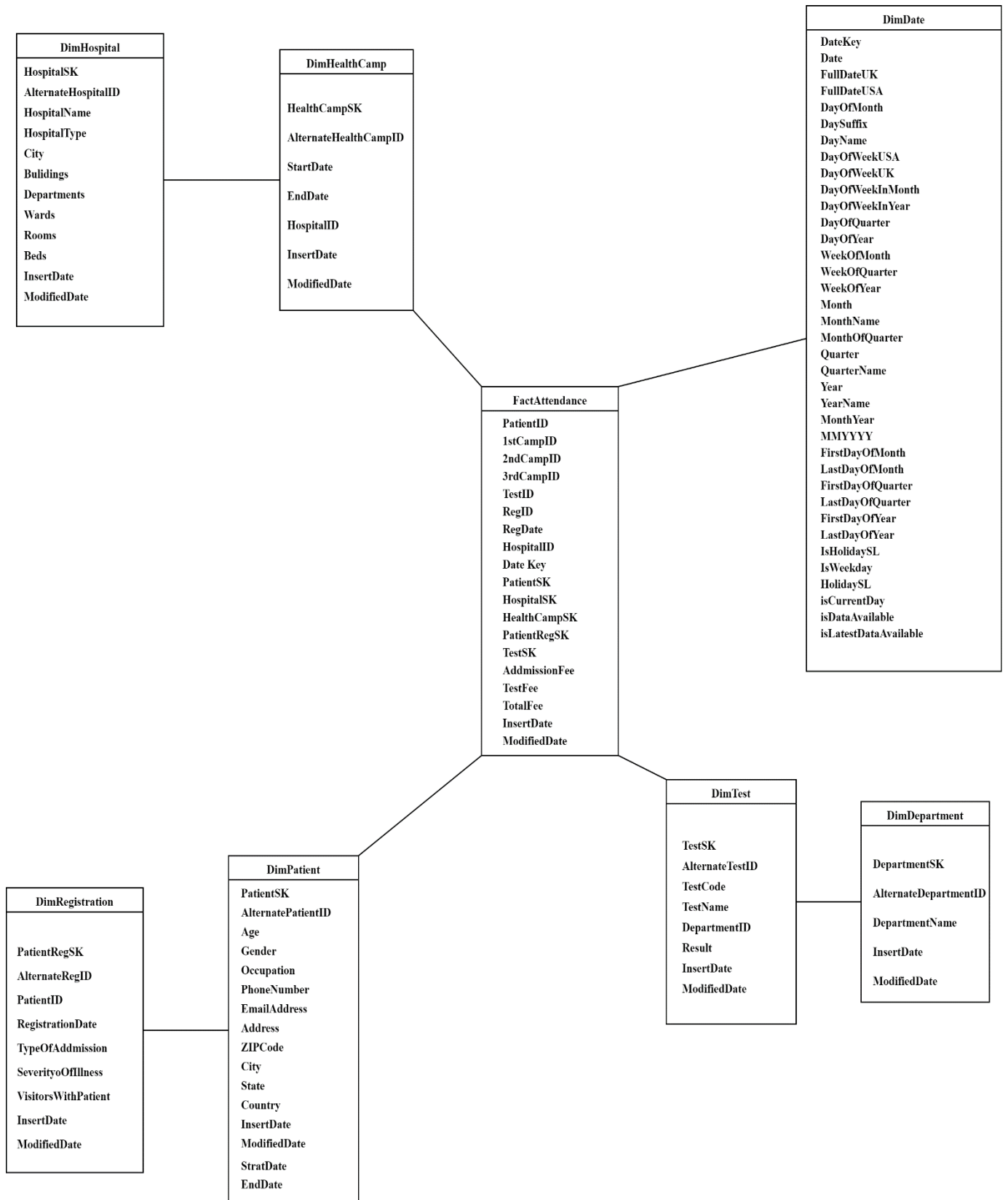
Staging DB component represents creating staging level tables through the ‘Extract’.

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
 - Hospital – buildings → departments → ward → rooms → beds
 - Patient – country → state → city → ZIP code → address
- Slowly Changing Dimension
 - 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
 - FK - Patient ID, Health Camp ID, Test ID, Hospital ID, Date Key, Registration ID, Department ID

Assumptions

- Patient dimension was considered as a slowly changing dimension.

Step 5: ETL development

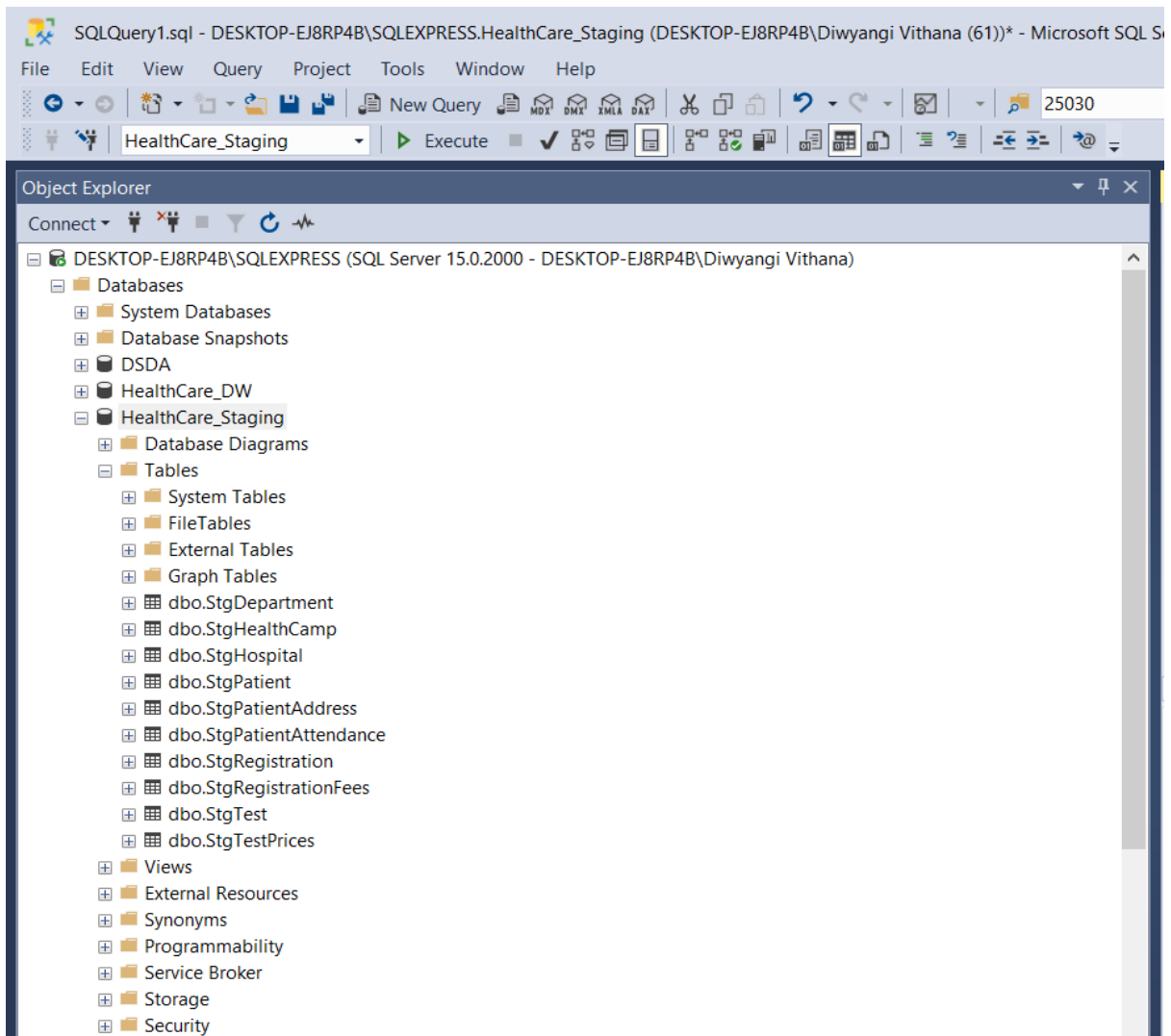
1.Extract

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

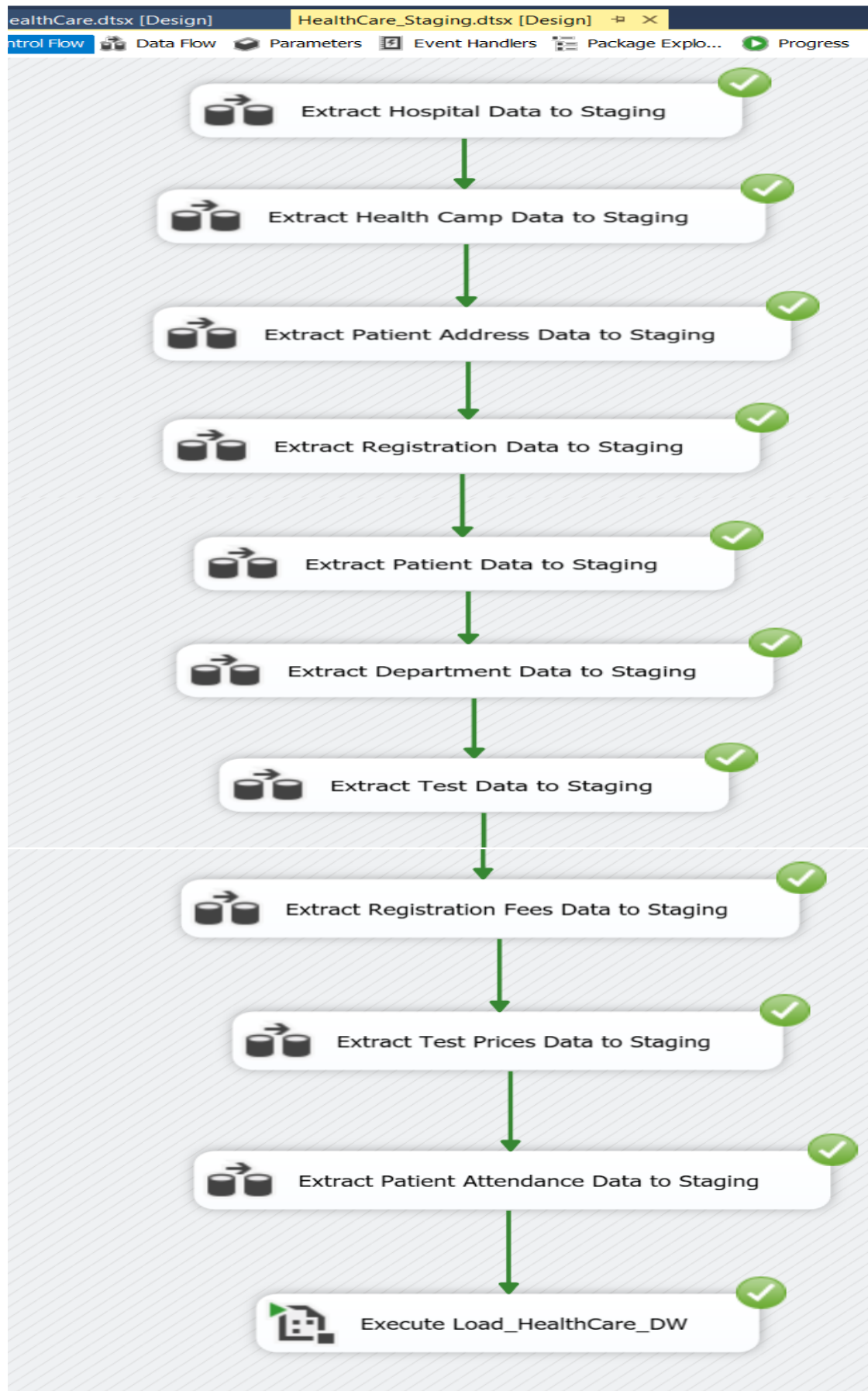
Flat file connection was used for text files and csv files, Excel file connections for excel file, DB source connection for DB file. All those tables were imported to the HealthCare_Staging DB, which contains the below tables,

1. SgtHospital
2. SgtHealthCamp
3. SgtPatientAddress
4. SgtPatient
5. SgtRegistration
6. SgtDepartment
7. SgtTest
8. SgtAttendance
9. SgtTestPrices
10. SgtAddmissionFee

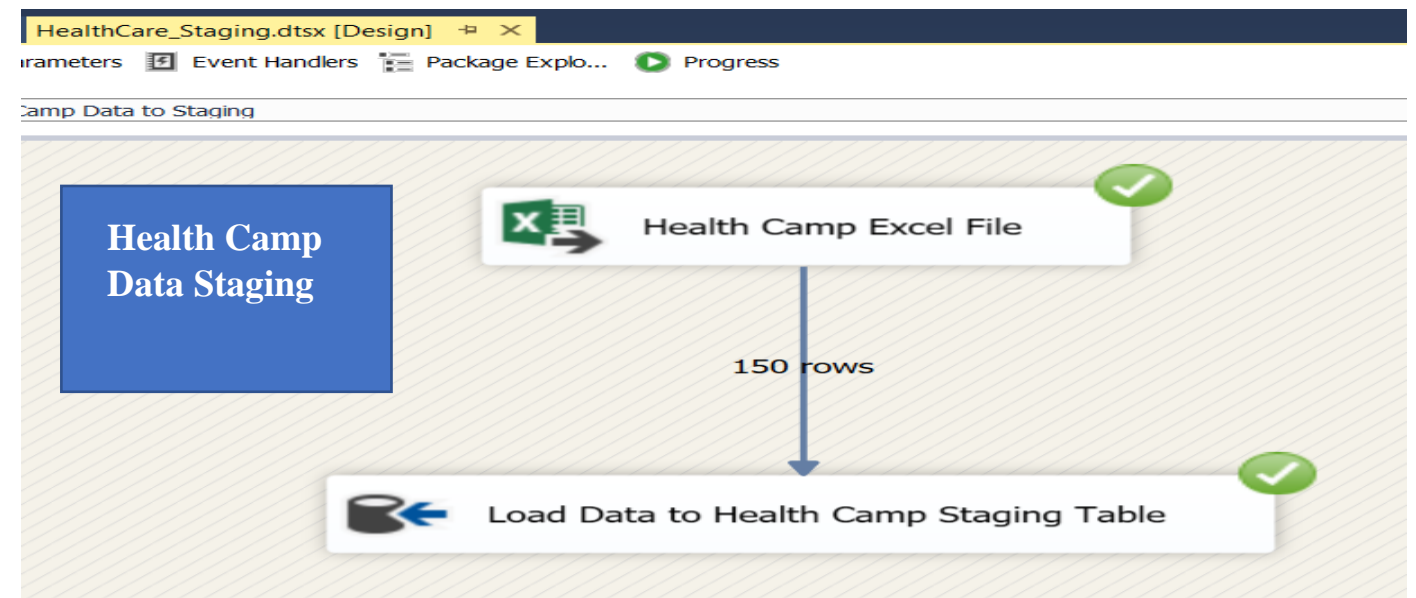
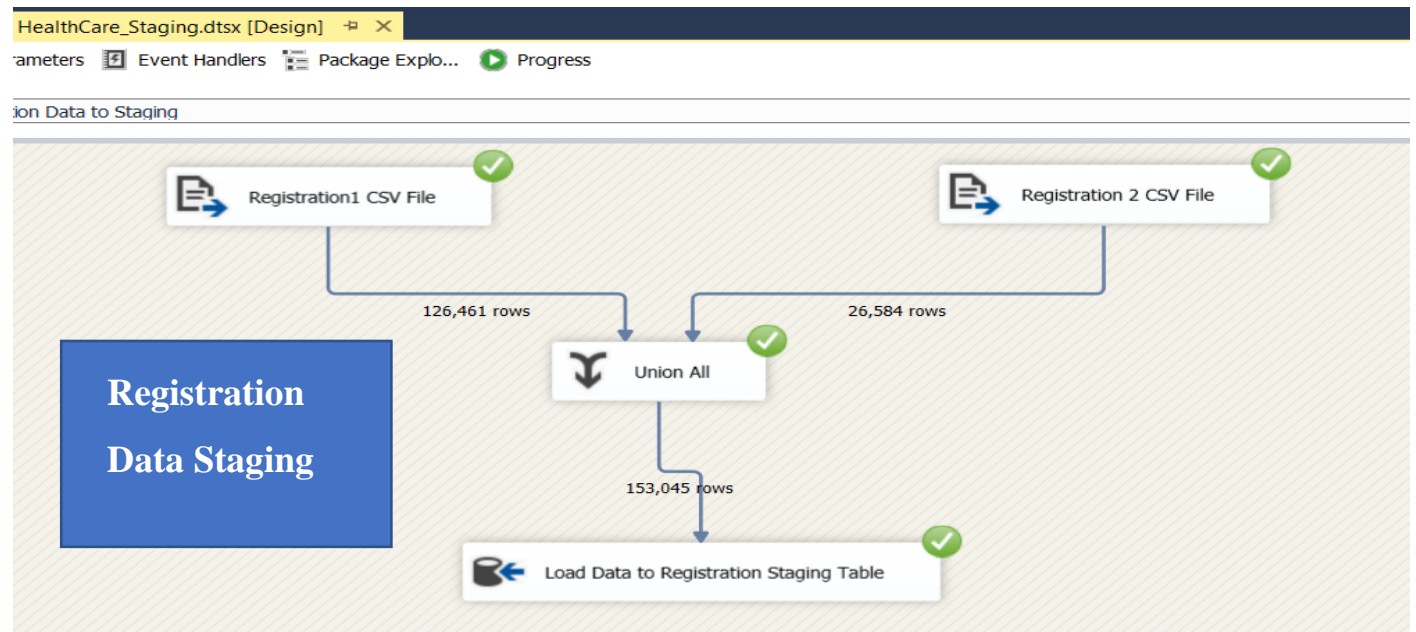
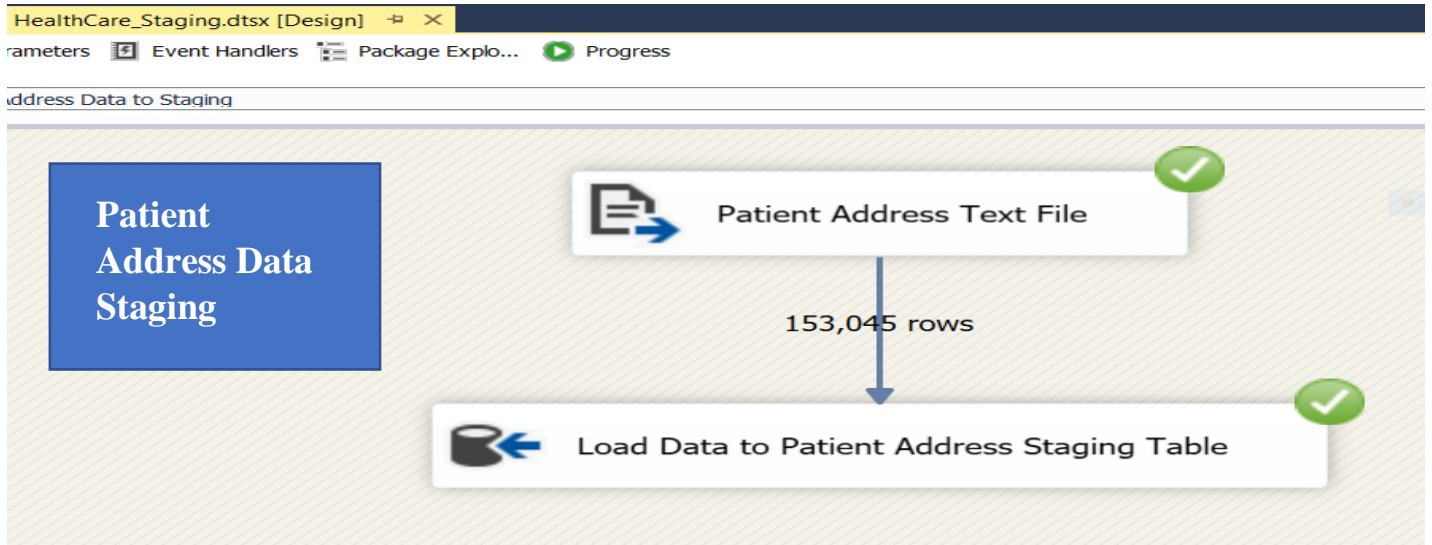
- **Snapshot of SSMS Staging Database**



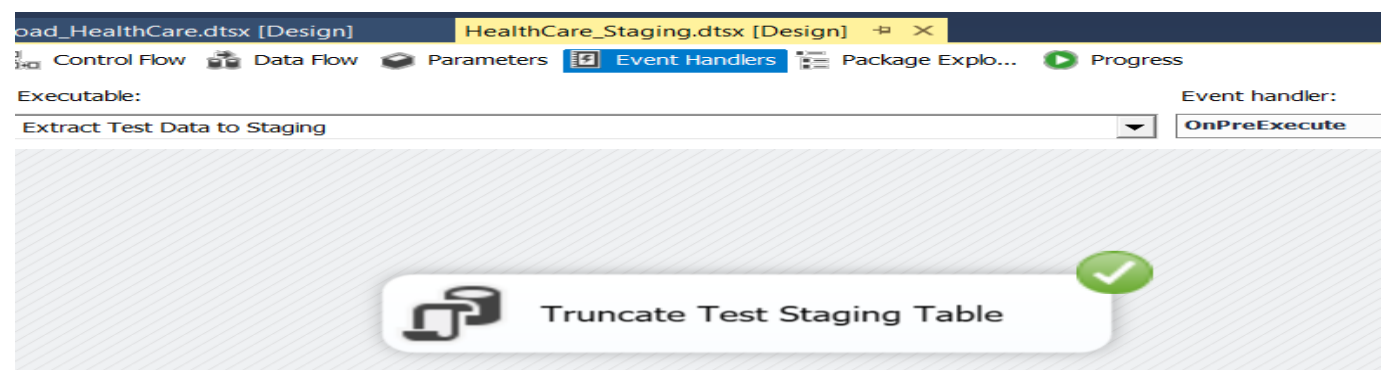
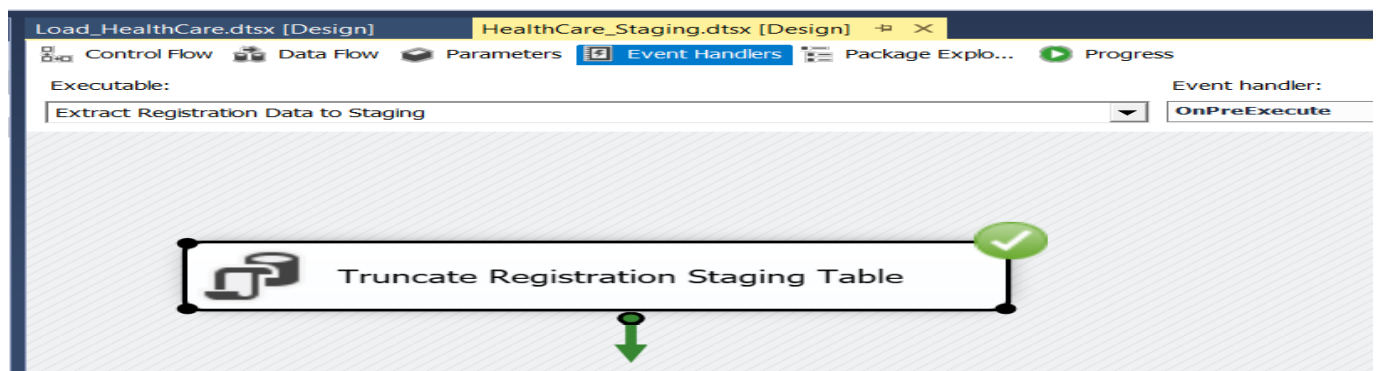
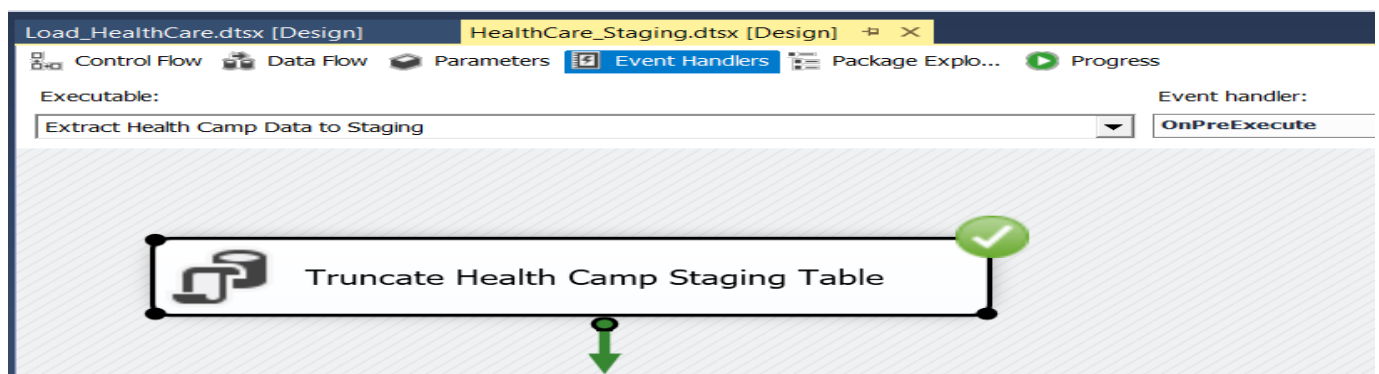
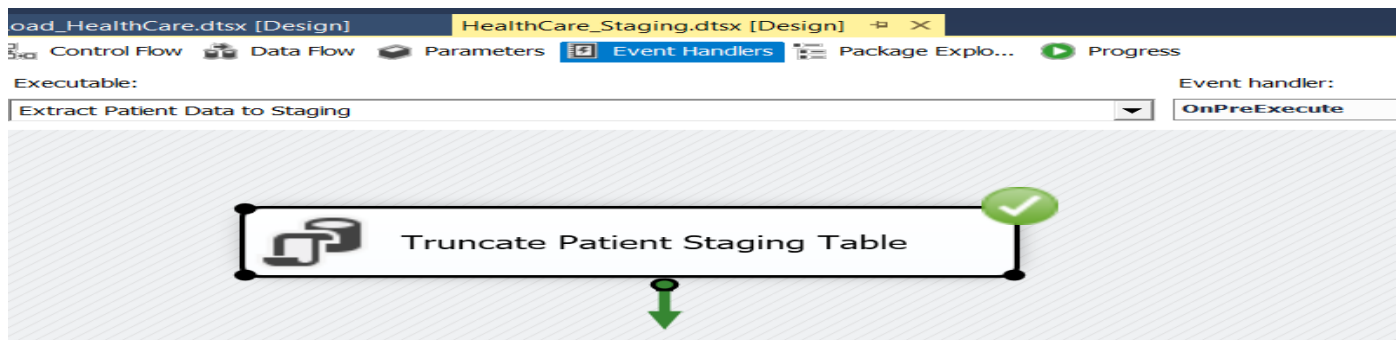
- **Snapshot of Visual Studio Control Flow of Extract**



- Snapshots of several data types of Data Flows

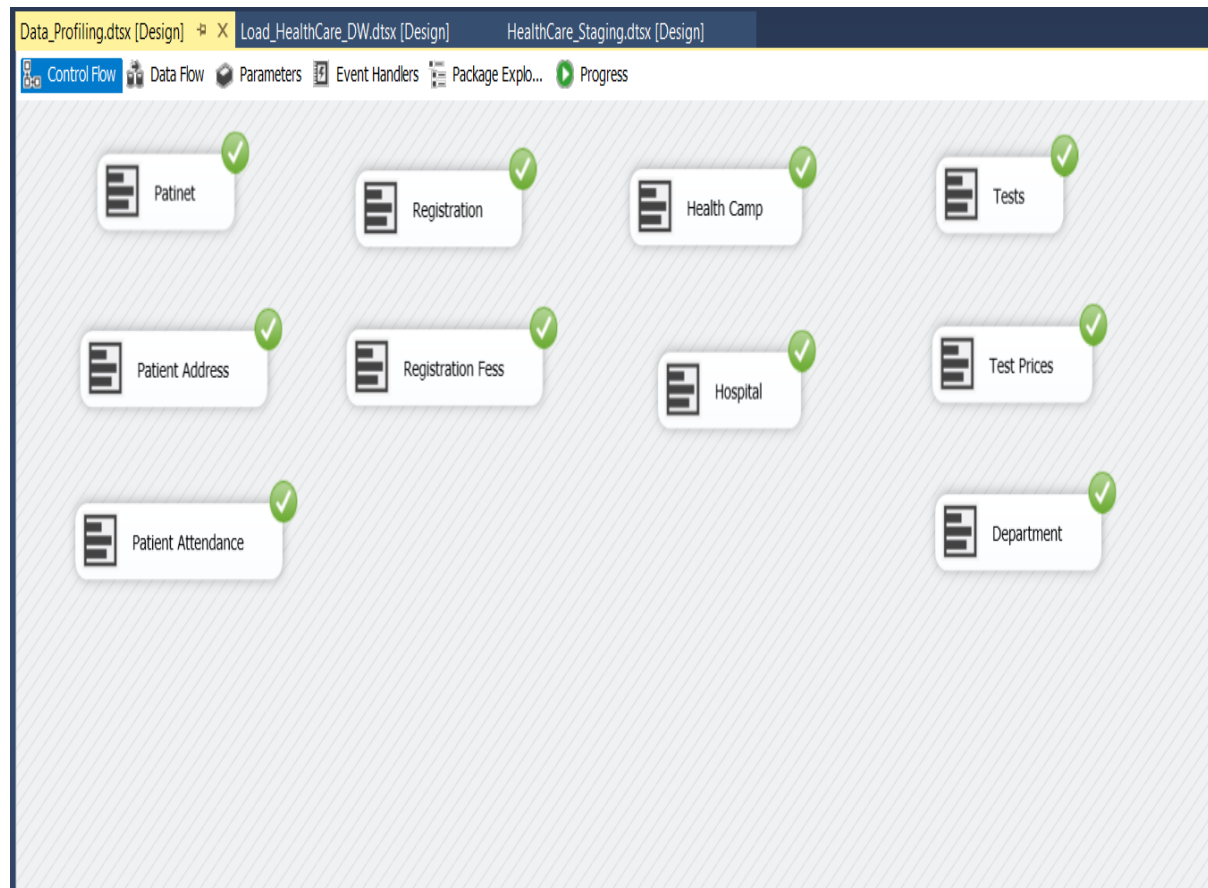


- **Event Handling (Truncate Staging Data)**



- **Data profiling**

Used Data_Profiling package to profiling the staging tables



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, HealthCare_DW, which contains the below tables,

1. DimHospital
2. DimHealthCamp
3. DimPatient
4. DimRegistration
5. DimDepartment
6. DimTest
7. DimDate
8. AttendanceFact

Used Transformation Tasks

1. Lookups

Health Camps’ Hospital ID is looked when loading to the in-Health Camp DimTable using Hospital Table

Tests’ Department ID is looked when loading to the to the in Test DimTable using Department Table.

2. Derived Columns

Replace Null Hospital Types in Hospital DimTable

Replace Null Gender Values in Patient DimTable

3. Union

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

4. Sort and Merge

‘Sort’ is used sort out the Patient and Registration data and they are merged

‘Merge’ using PatientID.



Update Functions

- Patient

SQLQuery9.sql - DE...yangi Vithana (56))*

```
USE [HealthCare_DW]
GO
/***** Object: StoredProcedure [dbo].[UpdateDimPatient]    Script Date: 5/14/2021 6:53:46 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[UpdateDimPatient]
    @PatientID int,
    @Age int,
    @Gender nvarchar(50),
    @Occupation nvarchar(255),
    @PhoneNumber nvarchar(50),
    @EmailAddress nvarchar(100),
    @Address nvarchar(255),
    @ZIPCode int,
    @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, PhoneNumber, EmailAddress, Address, ZIPCode, City, State, Country, InsertDate, ModifiedDate)
        values
        (@PatientID, @Age, @Gender, @Occupation, @PhoneNumber, @EmailAddress, @Address, @ZIPCode, @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,
        PhoneNumber = @PhoneNumber,
        EmailAddress = @EmailAddress,
        Address = Address,
        ZIPCode = ZIPCode,
        City = City,
        State = State,
        Country = Country
        where AlternatePatientID = @PatientID
    END;
END;
GO
```

- Registration

SQLQuery10.sql - D...yangi Vithana (55)  

```
USE [HealthCare_DW]
GO

/***** Object: StoredProcedure [dbo].[UpdateDimRegistration]    Script Date: 5/14/2021 6:54:15 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[UpdateDimRegistration]
    @RegID int ,
    @PatientID int,
    @HospitalID int,
    @RegistrationDate datetime,
    @TypeOfAdmission nvarchar(50),
    @SeverityOfIllness nvarchar(50),
    @VisitorsWithPatient int
AS
BEGIN
    if not exists (select PatientRegSK
    from dbo.DimRegistration
    where AlternateRegID = @RegID)
    BEGIN
        insert into dbo.DimRegistration
        (AlternateRegID, PatientID, HospitalID , RegistrationDate, TypeOfAdmission, SeverityOfIllness, VisitorsWithPatient,
        InsertDate, ModifiedDate)
        values
        (@RegID, @PatientID, @HospitalID , @RegistrationDate, @TypeOfAdmission, @SeverityOfIllness, @VisitorsWithPatient,
        GETDATE(), GETDATE())
        END;
    if exists (select PatientRegSK
    from dbo.DimRegistration
    where AlternateRegID = @RegID)
    BEGIN
        update dbo.DimRegistration
        set AlternateRegID = @RegID ,
        PatientID = @PatientID ,
        HospitalID = @HospitalID,
        RegistrationDate = @RegistrationDate,
        TypeOfAdmission = @TypeOfAdmission ,
        SeverityOfIllness = @SeverityOfIllness ,
        VisitorsWithPatient = @VisitorsWithPatient
        where AlternateRegID = @RegID
        END;
    END;
GO
```


- Hospital

SQLQuery8.sql - DE...yangi Vithana (55) ✖

```
USE [HealthCare_DW]
GO

/***** Object: StoredProcedure [dbo].[UpdateDimHospital]    Script Date: 5/14/2021 6:53:24 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[UpdateDimHospital]
    @HospitalID int,
    @HospitalName varchar(200),
    @HospitalType varchar(100),
    @City varchar(50),
    @Bulidings varchar(50),
    @Departments varchar(50),
    @Wards varchar(50),
    @Rooms varchar(50),
    @Beds varchar(50)
AS
BEGIN
    if not exists (select HospitalSK
        from dbo.DimHospital
        where AlternateHospitalID = @HospitalID)
    BEGIN
        insert into dbo.DimHospital
        (AlternateHospitalID, HospitalName, HospitalType, City, Bulidings, Departments, Wards, Rooms, Beds ,
        InsertDate, ModifiedDate)
        values
        (@HospitalID, @HospitalName, @HospitalType, @City, @Bulidings, @Departments, @Wards, @Rooms, @Beds,
        GETDATE(), GETDATE())
        END;
    if exists (select HospitalSK
        from dbo.DimHospital
        where AlternateHospitalID = @HospitalID)
    BEGIN
        update dbo.DimHospital
        set HospitalName = @HospitalName,
        HospitalType = @HospitalType,
        City = @City,
        Bulidings = @Bulidings,
        Departments = @Departments,
        Wards = @Wards,
        Rooms = @Rooms,
        Beds = @Beds,
        ModifiedDate = GETDATE()
        where AlternateHospitalID = @HospitalID
        END;
    END;
```

- Health Camp

SQLQuery7.sql - DE...yangi Vithana (55) ✕

```
USE [HealthCare_DW]
GO
```

/****** Object: StoredProcedure [dbo].[UpdateDimHealthCamp] Script Date: 5/14/2021 6:52:53 AM *****/

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [dbo].[UpdateDimHealthCamp]
    @HealthCampID int,
    @HospitalID int,
    @StartDate datetime,
    @EndDate datetime
AS
BEGIN
    if not exists (select HealthCampSK
        from dbo.DimHealthCamp
        where AlternateHealthCampID = @HealthCampID)
    BEGIN
        insert into dbo.DimHealthCamp
            (AlternateHealthCampID, StartDate, EndDate, HospitalID, InsertDate, ModifiedDate)
        values
            (@HealthCampID, @StartDate, @EndDate, @HospitalID, GETDATE(), GETDATE())
        END;
    if exists (select HealthCampSK
        from dbo.DimHealthCamp
        where AlternateHealthCampID = @HealthCampID)
    BEGIN
        update dbo.DimHealthCamp
        set HospitalID = @HospitalID,
            StartDate = @StartDate,
            EndDate = @EndDate,
            ModifiedDate = GETDATE()
        where AlternateHealthCampID = @HealthCampID
        END;
    END;
GO
```

- Department

SQLQuery6.sql - DE...yangi Vithana (55) X

```
USE [HealthCare_DW]
GO

/***** Object: StoredProcedure [dbo].[UpdateDimDepartment]    Script Date: 5/14/2021 6:51:50 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[UpdateDimDepartment]
    @DepartmentID int,
    @DepartmentName nvarchar(255)
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
        where AlternateDepartmentID = @DepartmentID
    END;
END;
GO
```

- Test

SQLQuery11.sql - D...yangi Vithana (54) ✖

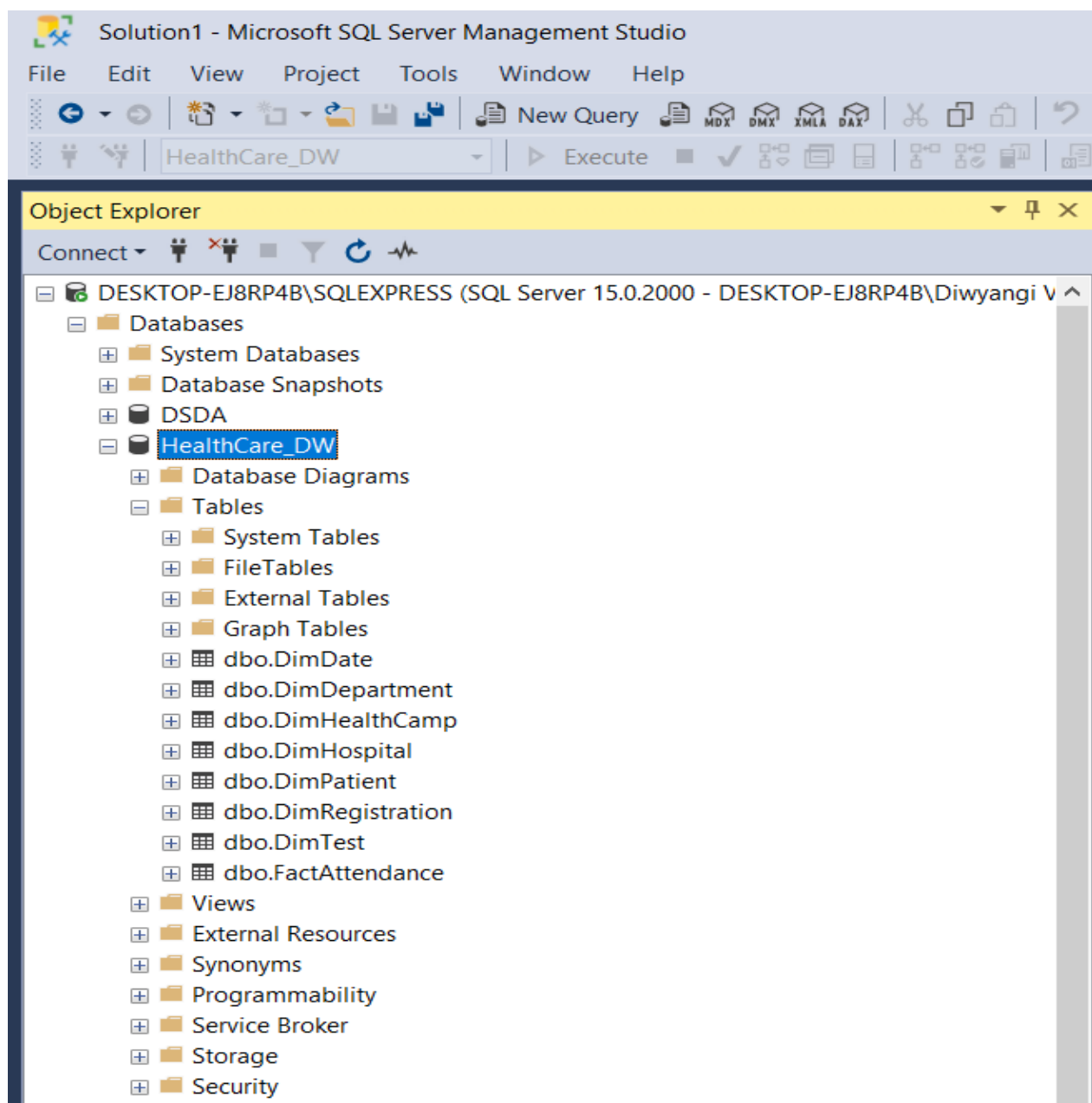
```
USE [HealthCare_DW]
GO

/***** Object: StoredProcedure [dbo].[UpdateDimTest]    Script Date: 5/14/2021 6:54:45 AM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[UpdateDimTest]
    @TestID int,
    @TestCode int,
    @TestName nvarchar(200),
    @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, PatientID, TestCode, TestName, DepartmentID, Result, InsertDate, ModifiedDate)
        values
        (@TestID, @PatientID, @TestCode, @TestName, @DepartmentID, @Result, GETDATE(), GETDATE())
        END;
    if exists (select TestSK
    from dbo.DimTest
    where AlternateTestID = @TestID)
    BEGIN
        update dbo.DimTest
        set AlternateTestID = @TestID,
            TestCode = @TestCode,
            TestName = @TestName,
            DepartmentID = @DepartmentID,
            Result = @Result,
            PatientID = @PatientID
        where AlternateTestID = @TestID
        END;
    END;
GO
```

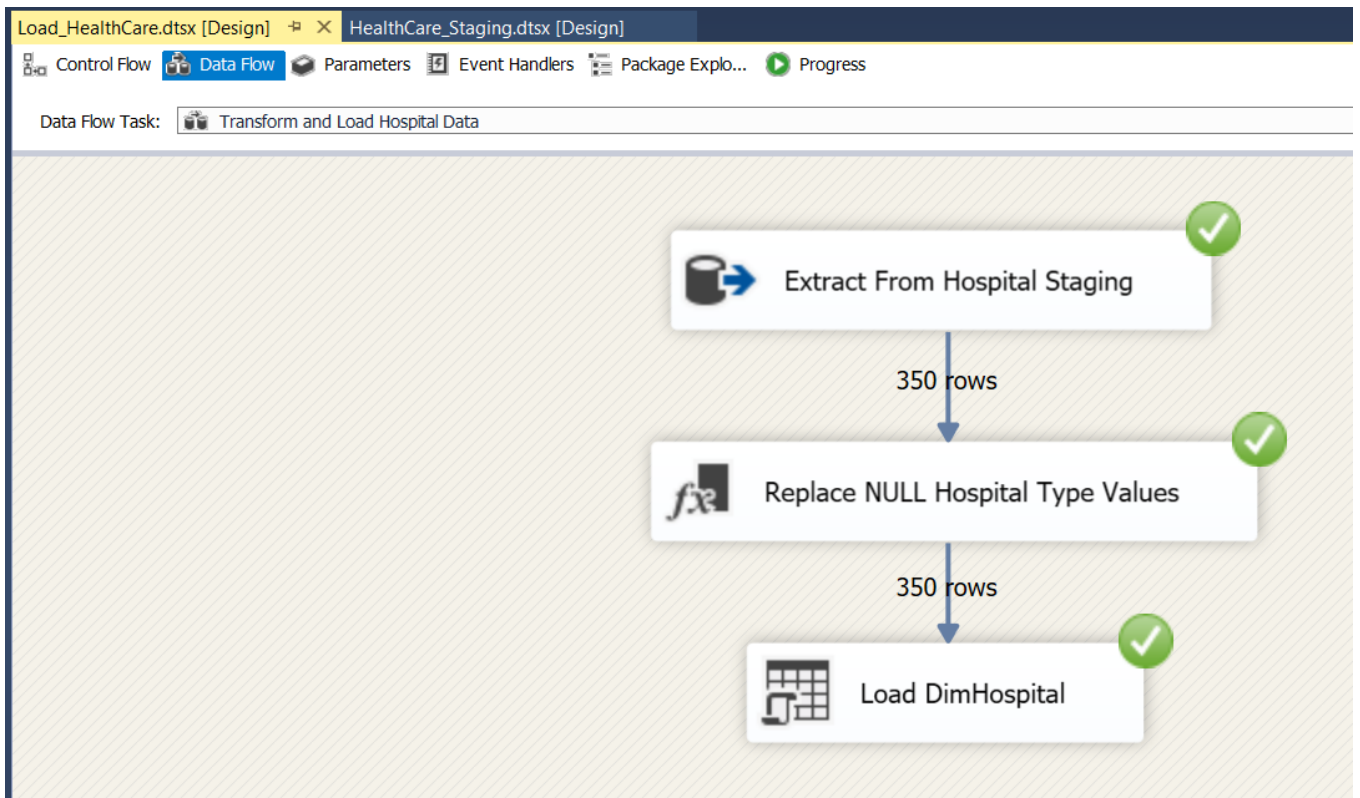
- **Snapshot of SQL server Data warehouse Database**



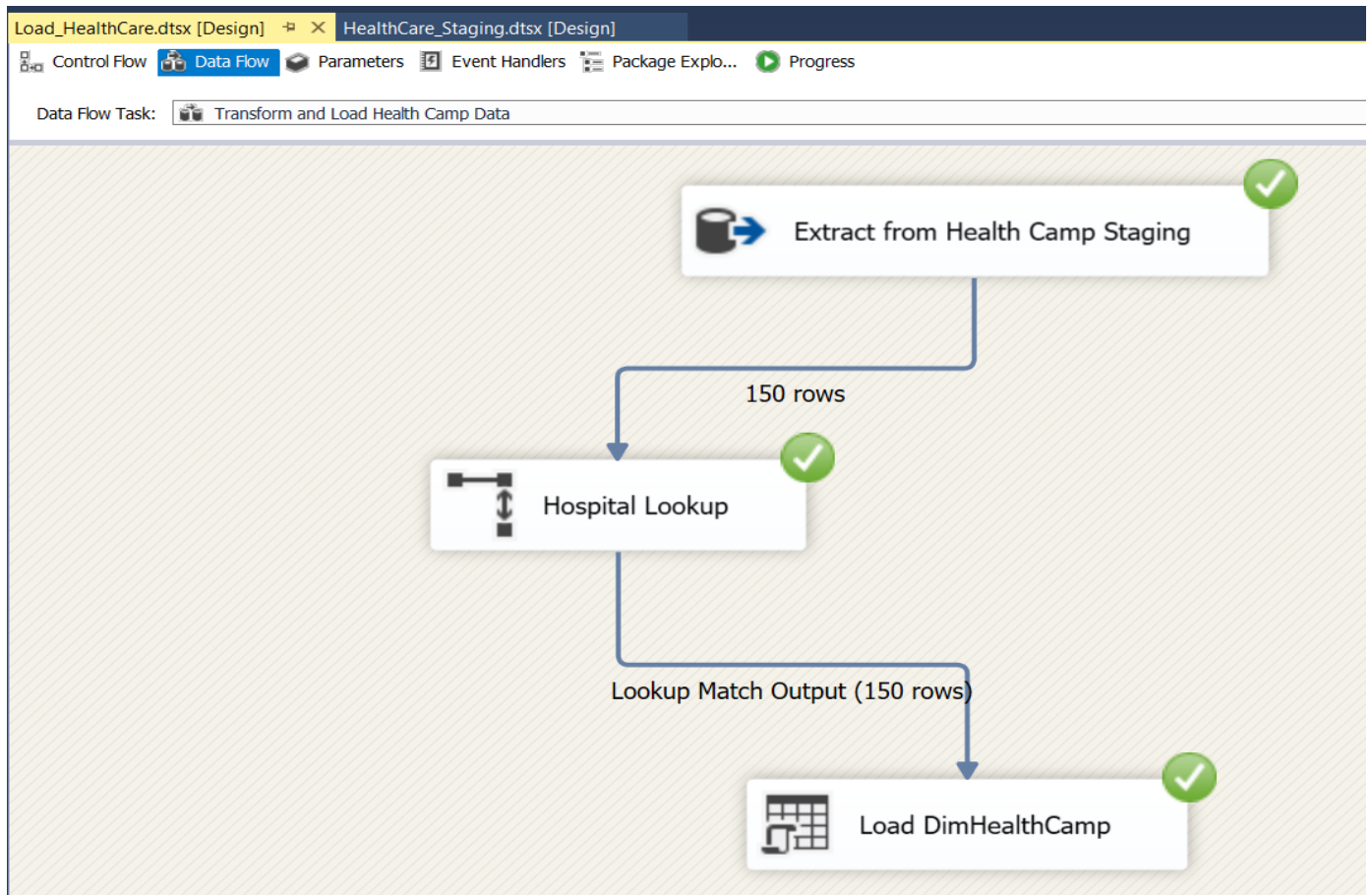
- **Snapshot of Visual Studio Control Flow of Extraction**



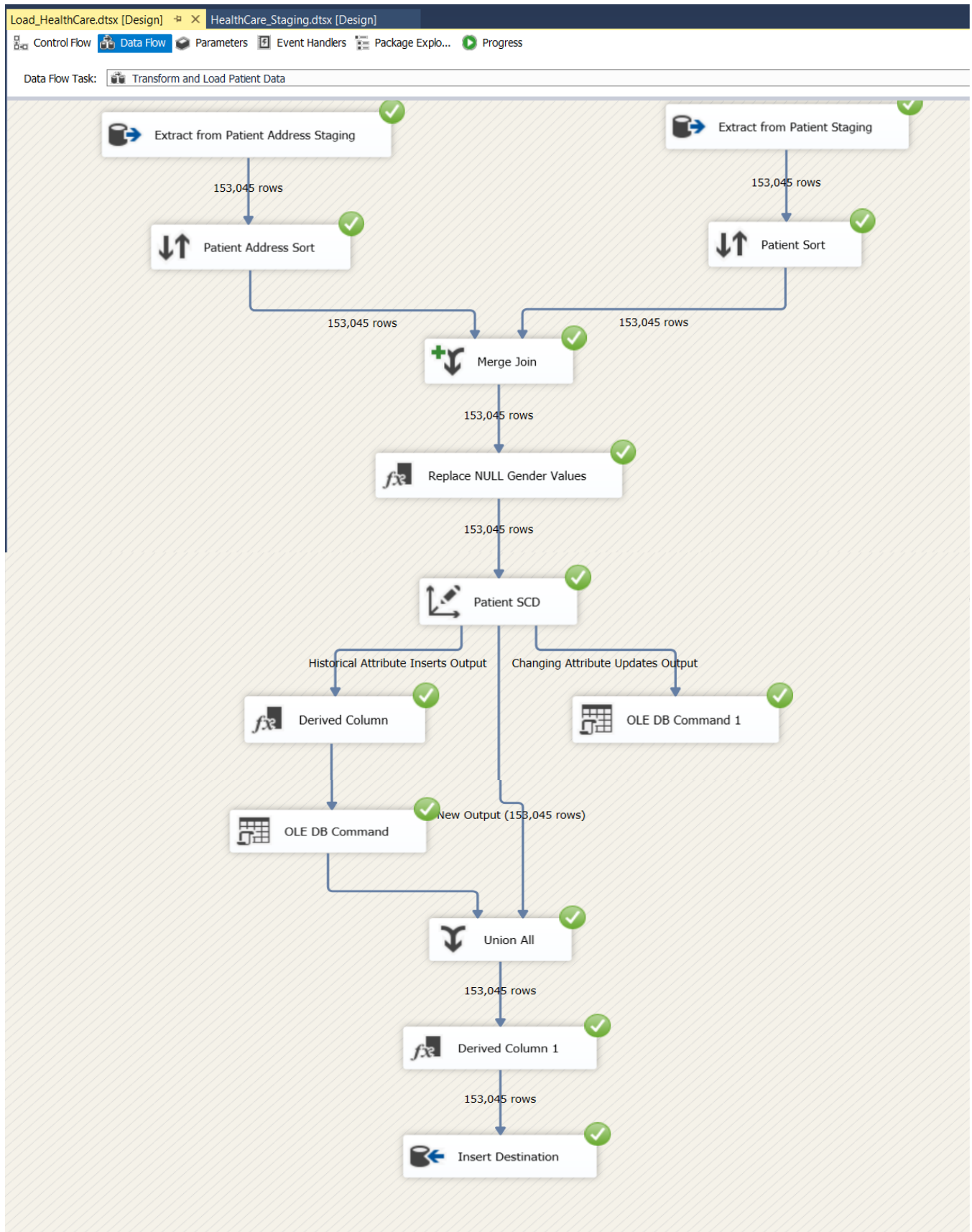
- **Hospital Data Transform and Load**



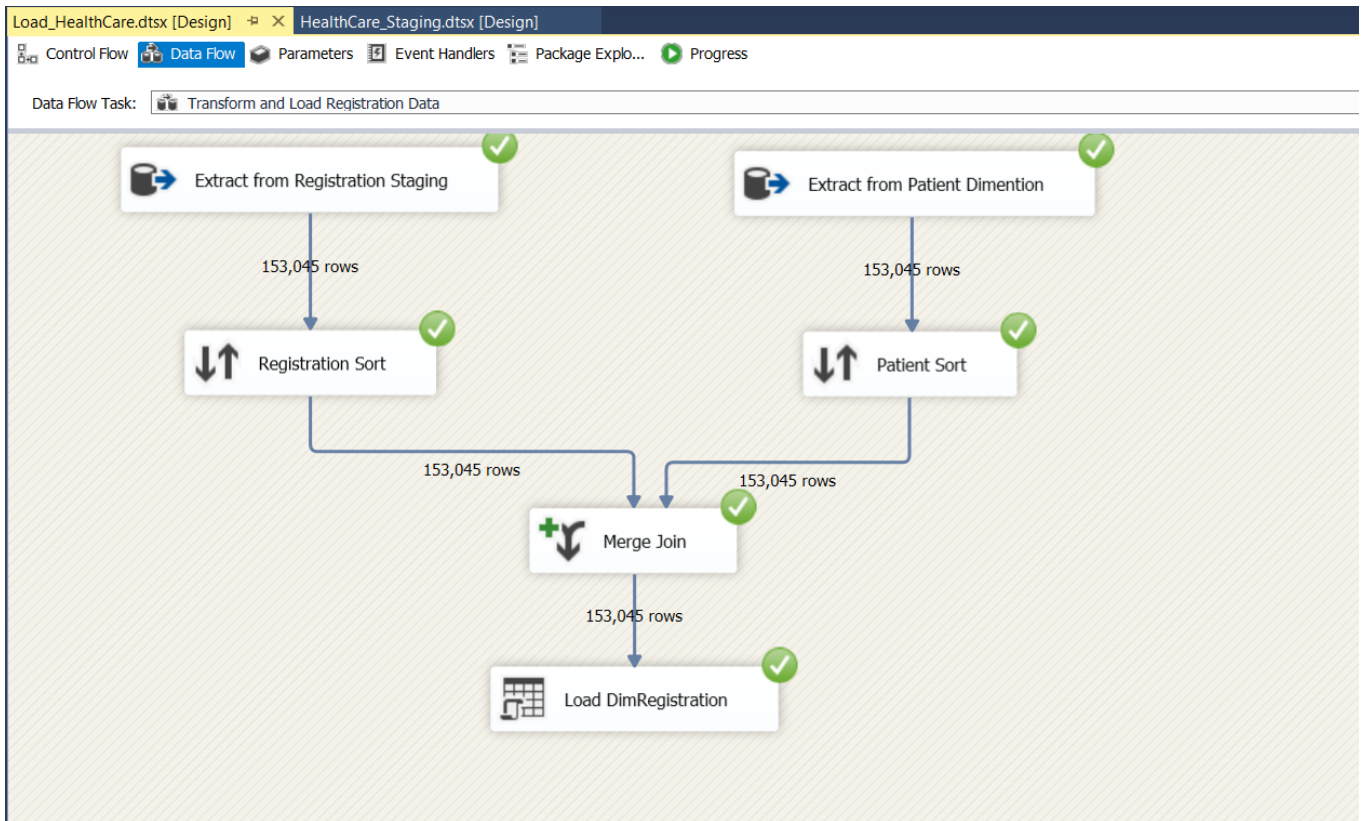
- **Health Camp Data Transform and Load**



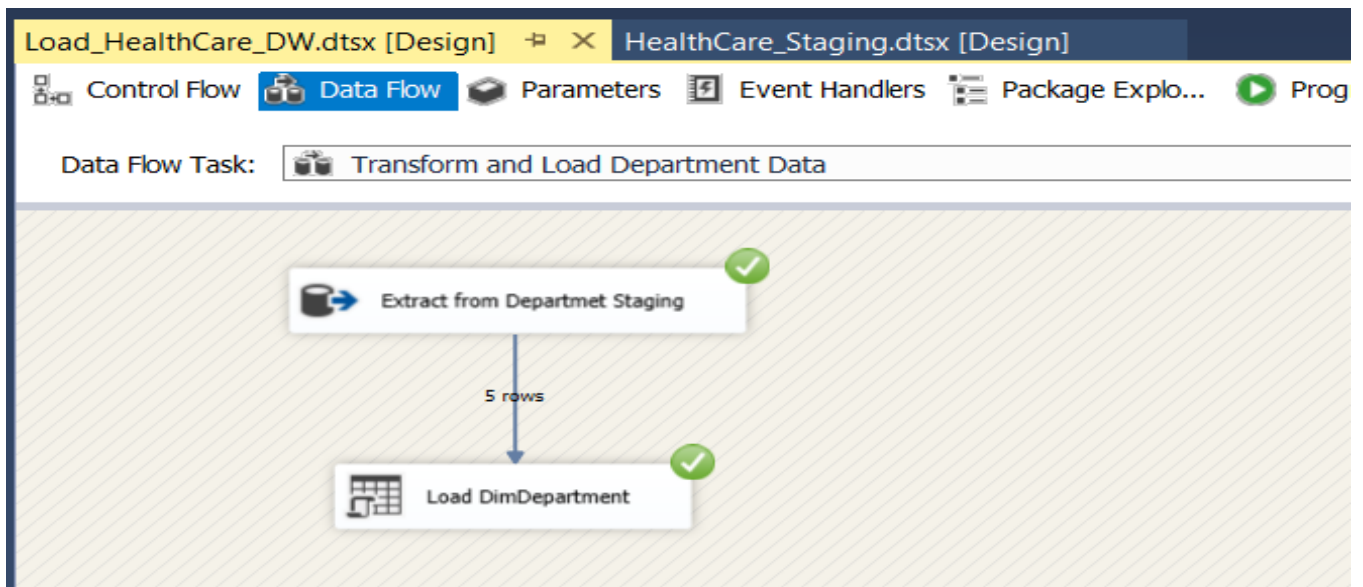
- Patient data Transform and Load

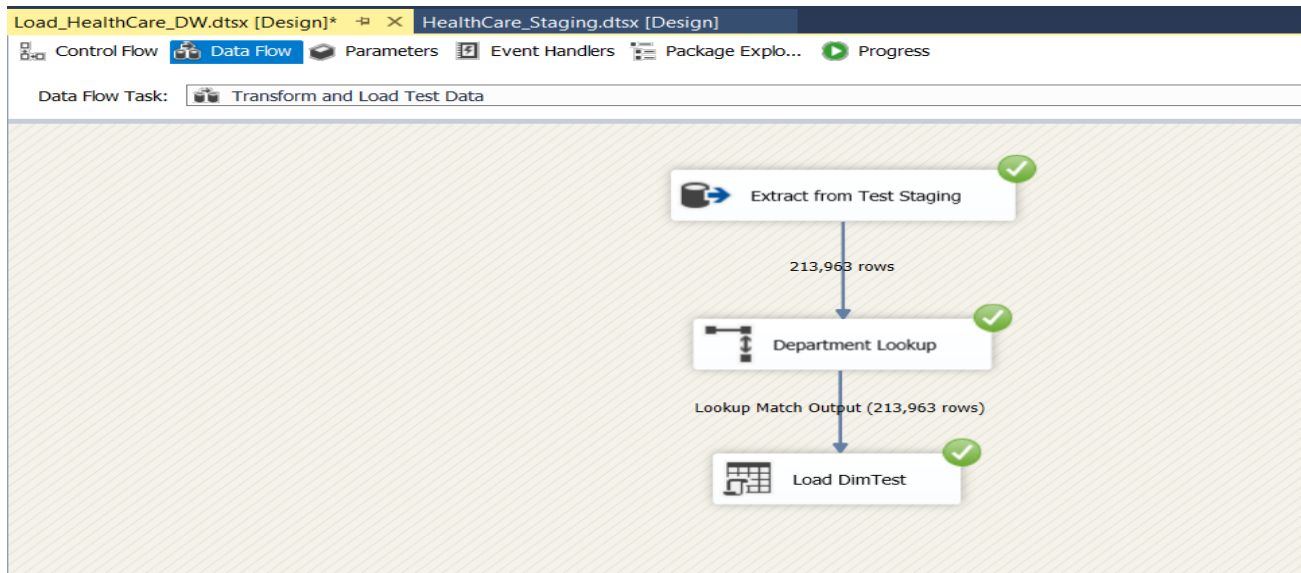


- **Registration Data Transform and Load**



- **Test and Department Data Transform and Load**





- **Attendance Fact Table Transform and Load**

