Sri Lanka Institute of Information Technology

IT3021

Data Warehousing and Business Intelligence

3 rd Year 1st Semester

Assignment 1

Student NO: IT19160344

Name: Liyanage K.L.O.G Y3S1.15(DS)

Table of Contents

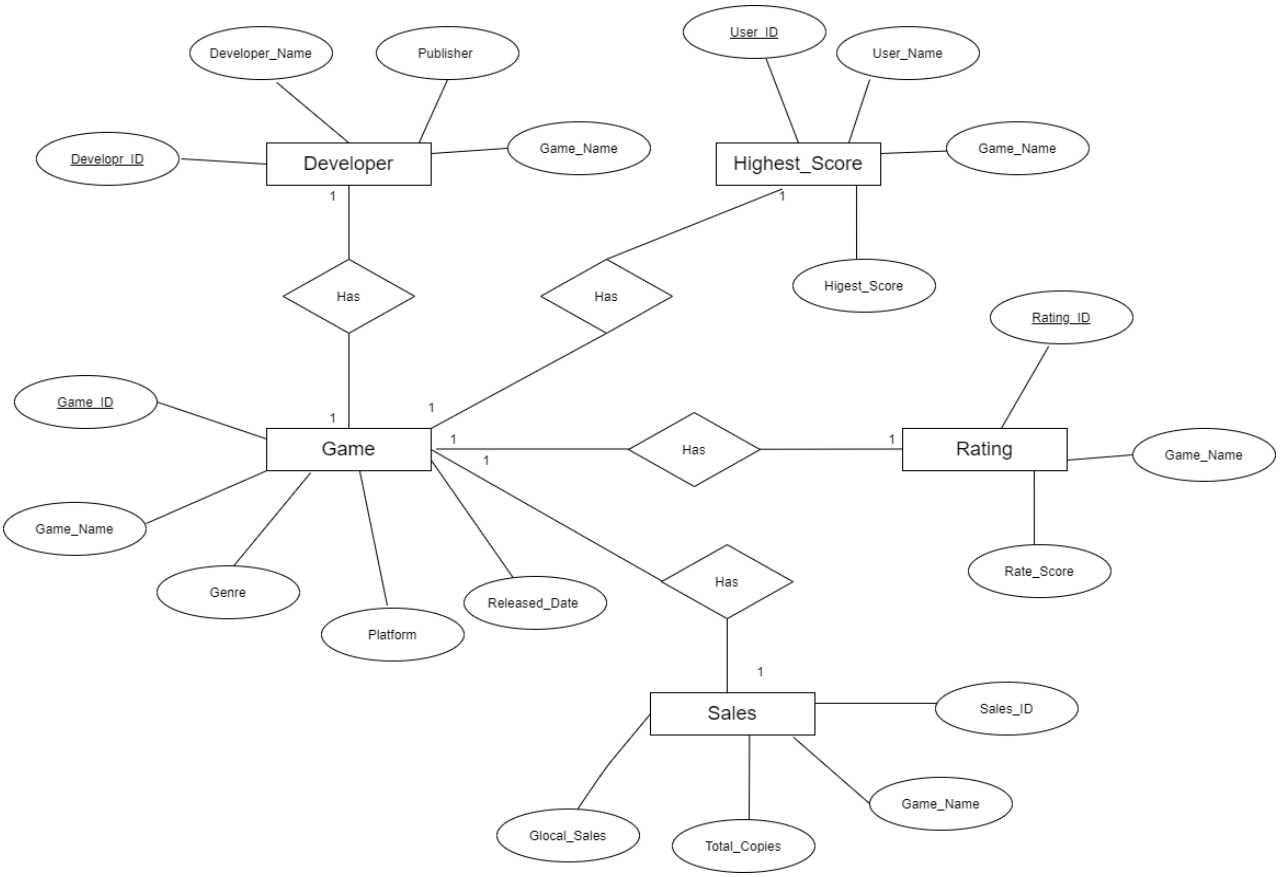
1. Dataset Selection

Link mention here show the source dataset:
https://www.kaggle.com/ashaheedq/video-games-sales-2019

Description Of the Dataset

This dataset describes sale details of each video games in different genres. And also this data set describes developer details in each game ,Rating details in each games and highest score user details in each games .

Entity Relationship Diagram



2. Preparation of Dataset

When I was selecting the dataset, all the files consist only with .csv extension. But the assignment asks us to do with couple of different file sources. Because of that I changed some files into,

- **Text files (.txt)**
- **CSV files**

Text file

- **Developer Details –** Developer.txt contains developer details in each game and including developer id ,developer name, game name and publisher details

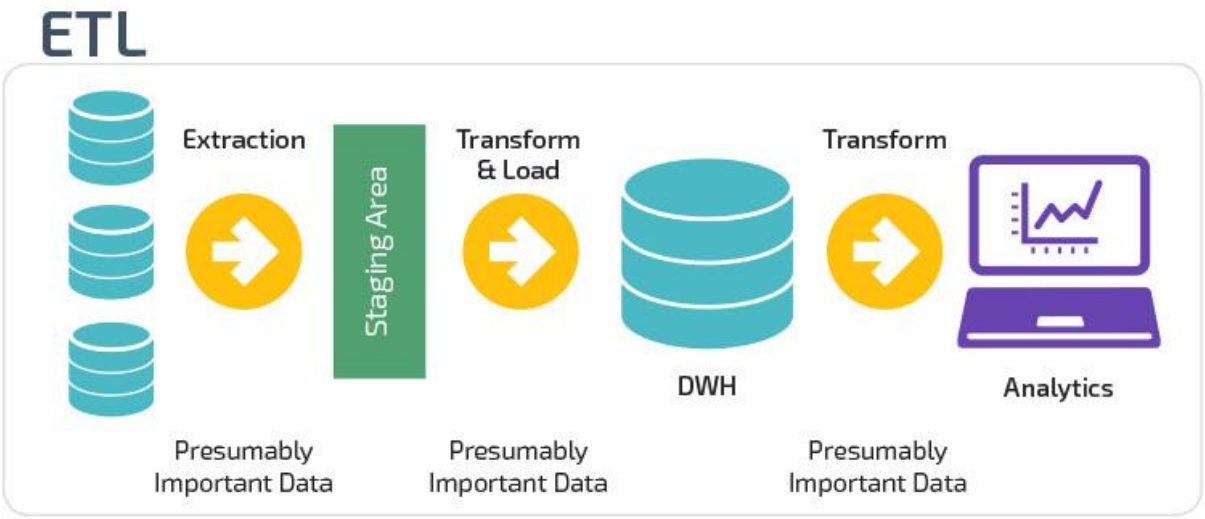
CSV files

- **Game Details –** Game.csv contains game details in each game including in game id , game name ,platform , genre and publish date
- **Rating Details -** Rating.csv contains rating in each game including rating id , game name , rating score.
- **Highest User Score details –** HighestUserScore_Details.csv contains highest score user details in each game including user id , user name , game name and user score .
- **Sales details –** Sales.csv contains sales details in each game including sales id, Game name ,total copies ,global sales ,sales in japan ,sales in north America and etc .

3. Solution Architecture

Architectural Diagram

This is followed by a particular implementation of OLAP Architecture, which specifies the aggregation of data from different sources, such as Organizational Flat Files (TEXT files), Flat Files (CSV), which is processed through the ETL method, which involves extracting data from sources that reach the Staging layer (intermediate layer) and then transforming the staging data using the memcached algorithm.

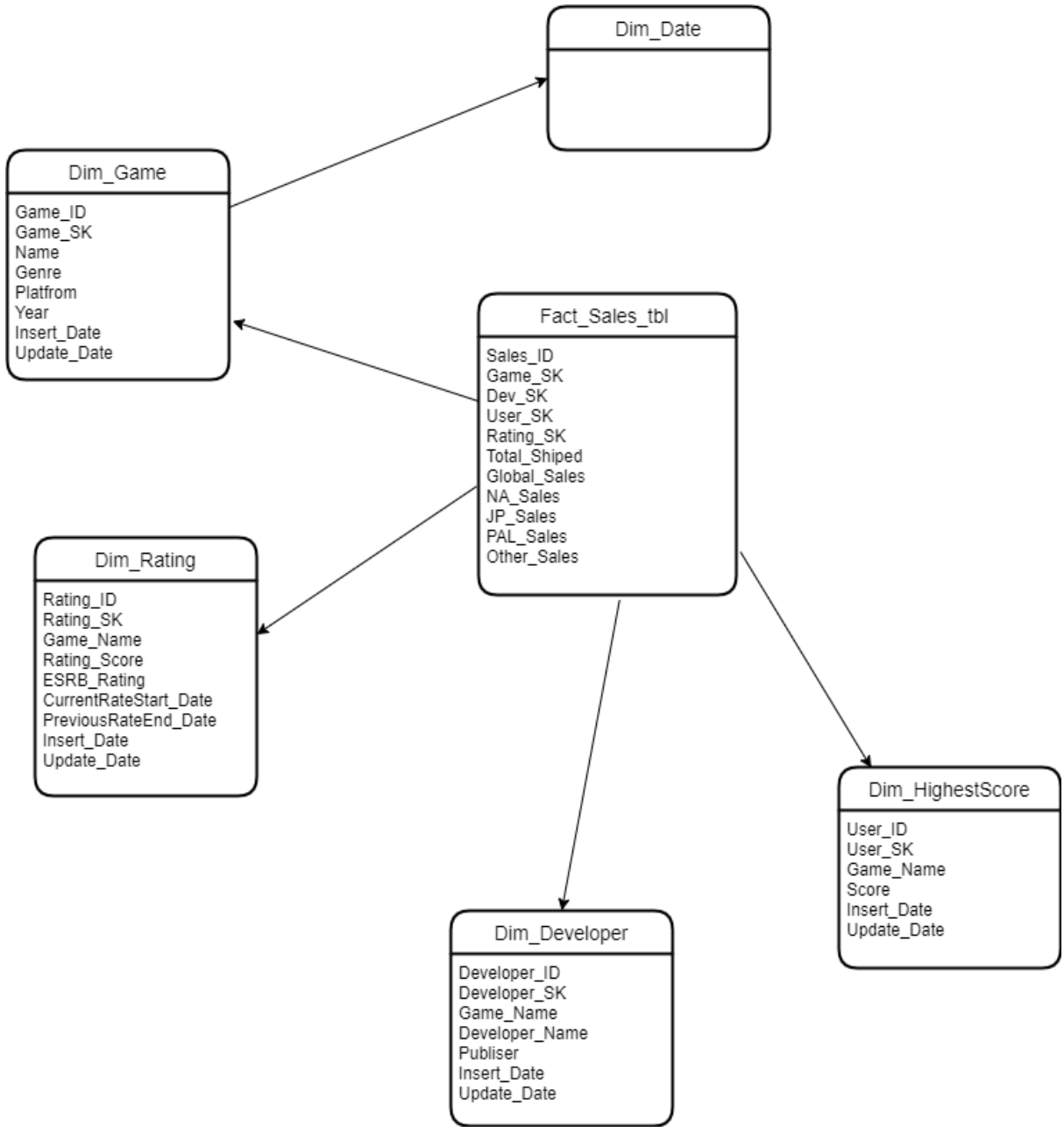


All the data in the sources should be bringing into the staging database. Those sources are consisting with 2 different file types. After bringing them into staging layer

a) SalesStaging
b) RatingStaging
c) DeveloperStaging
d) HighestScoreStaging
e) GameStaging

4. Data Warehouse Design and Development

A Star Schema is used to design the Data Warehouse as shown in the diagram below. 4 Dimension tables are designed with another Fact table. Fact_Sales_tbl table was create by merging the Dim_Developer table , Dim_Game table Dim_HighestScore and Dim_Rating



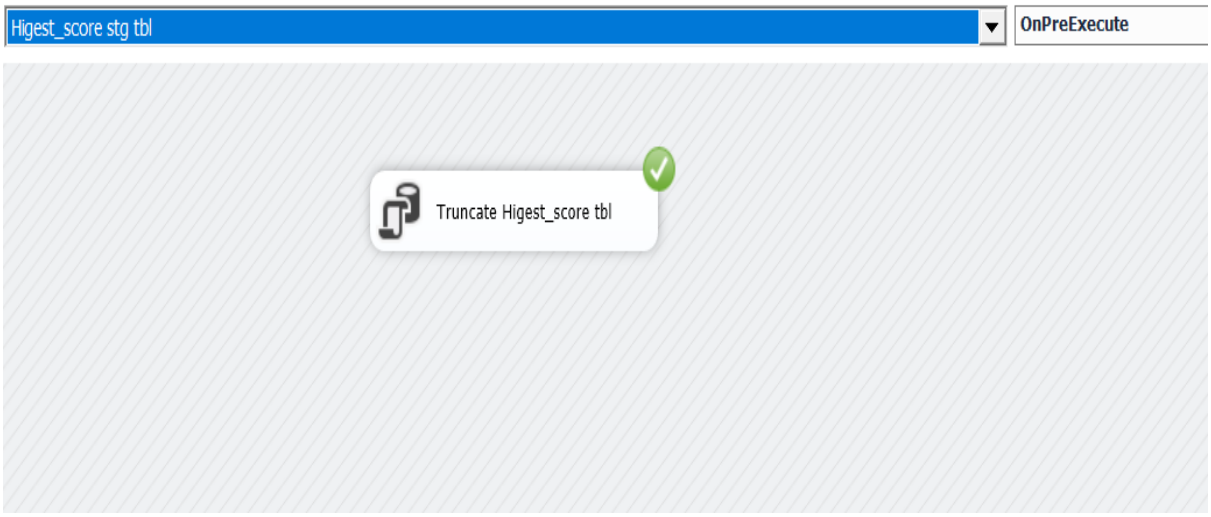
Assumptions

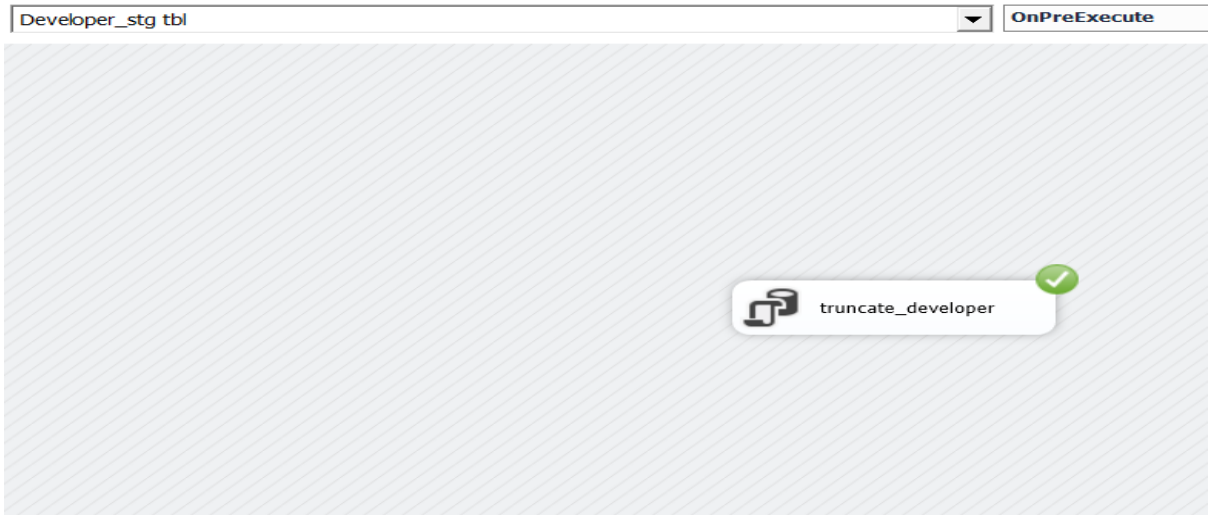
- Dim_Rating considered as slowly changing dimension .

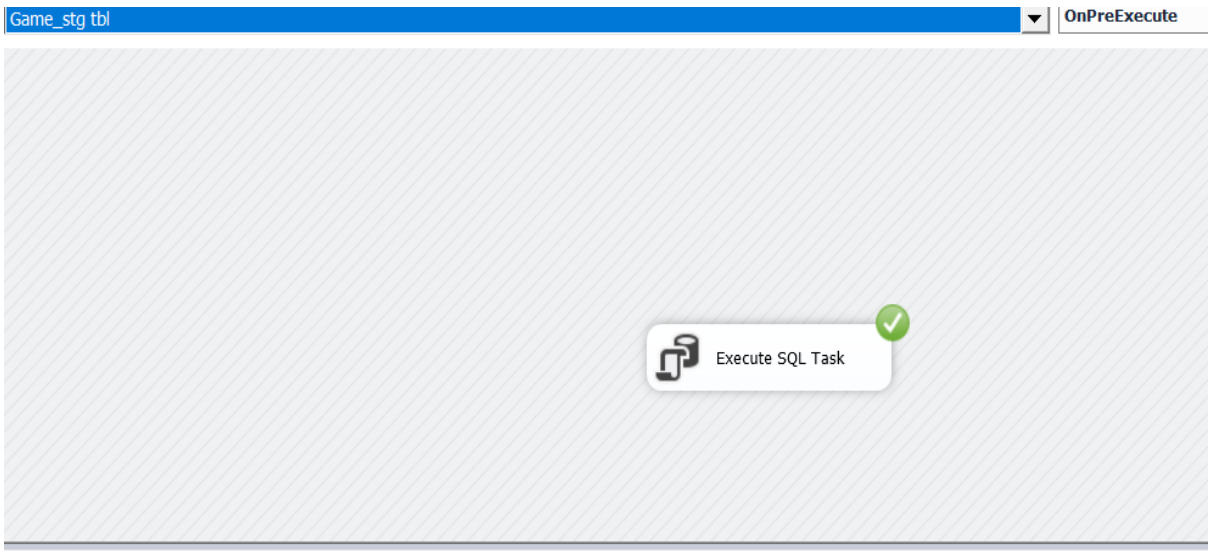
5. ETL Development

Extraction

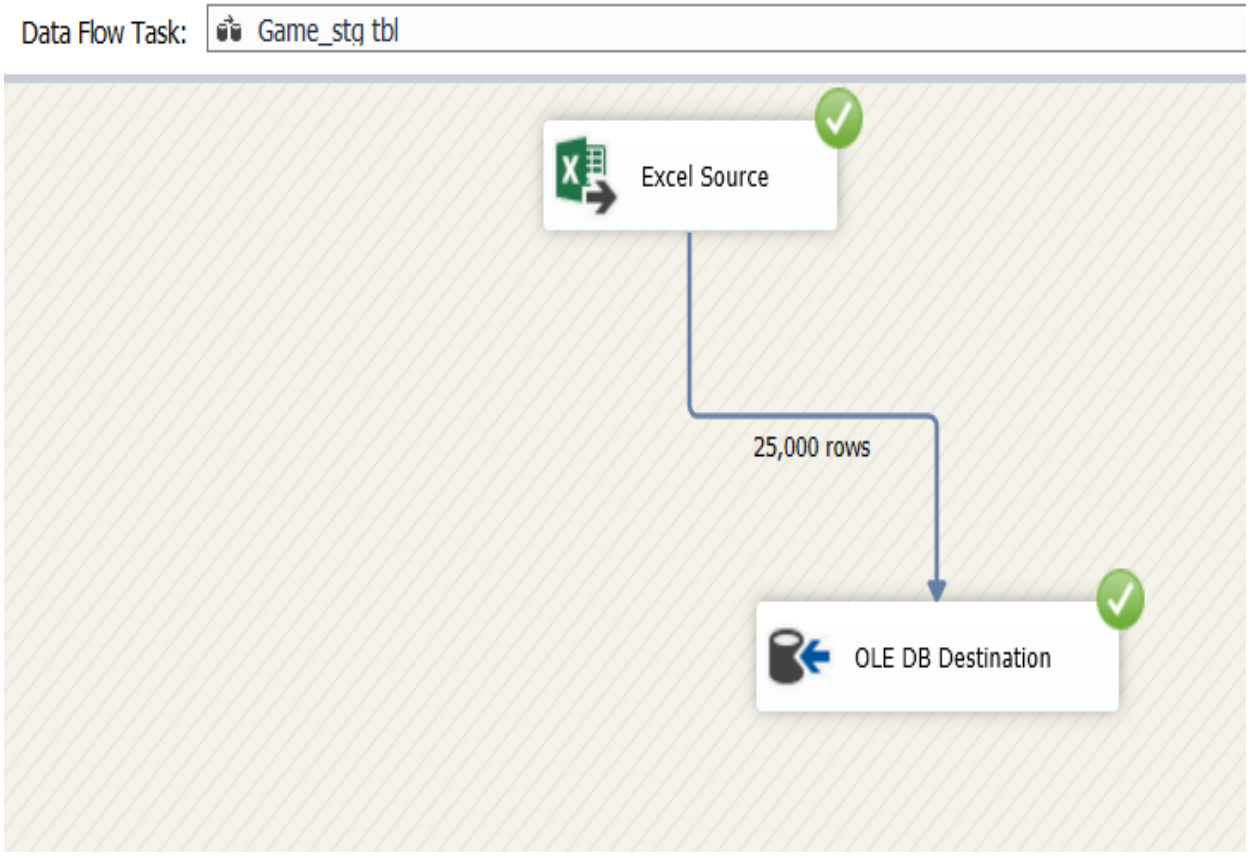
Data was obtained from sources using SQL Server Data Tools in the first step. The tables in the staging database were truncated before each extraction in the data extraction process and it was added to the current database.

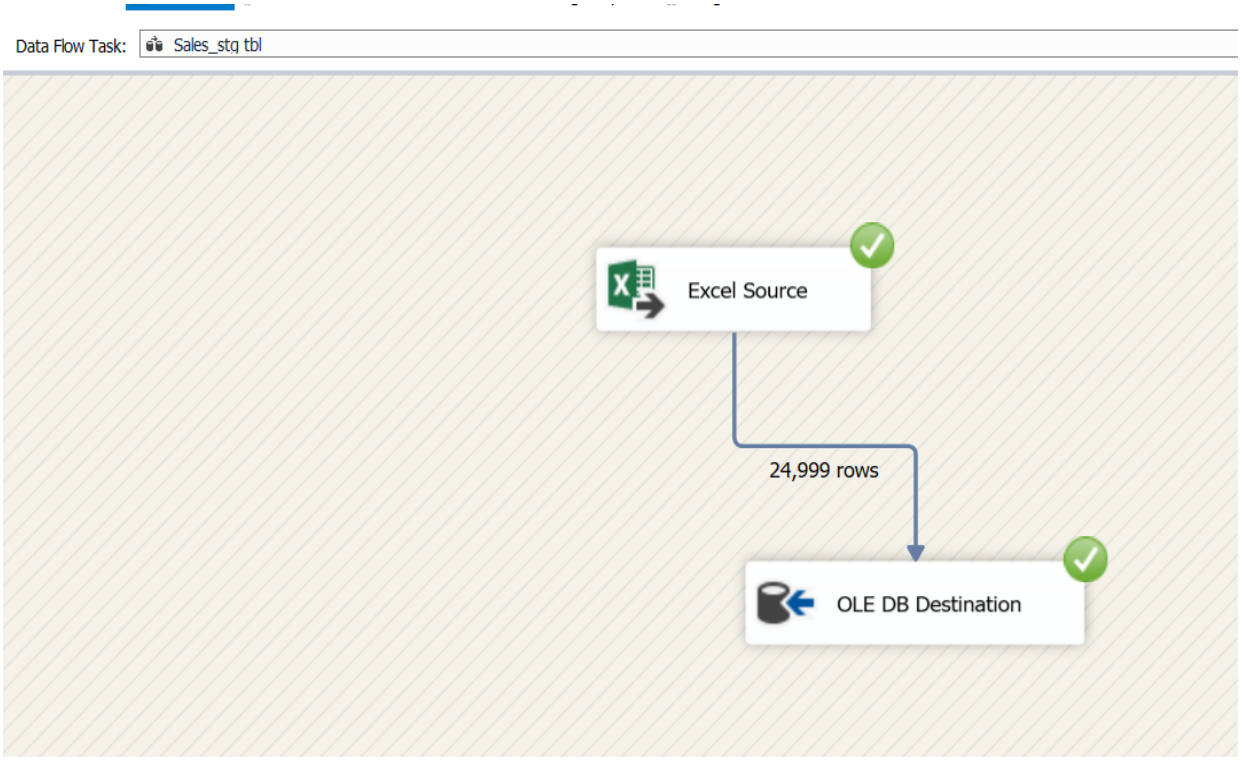


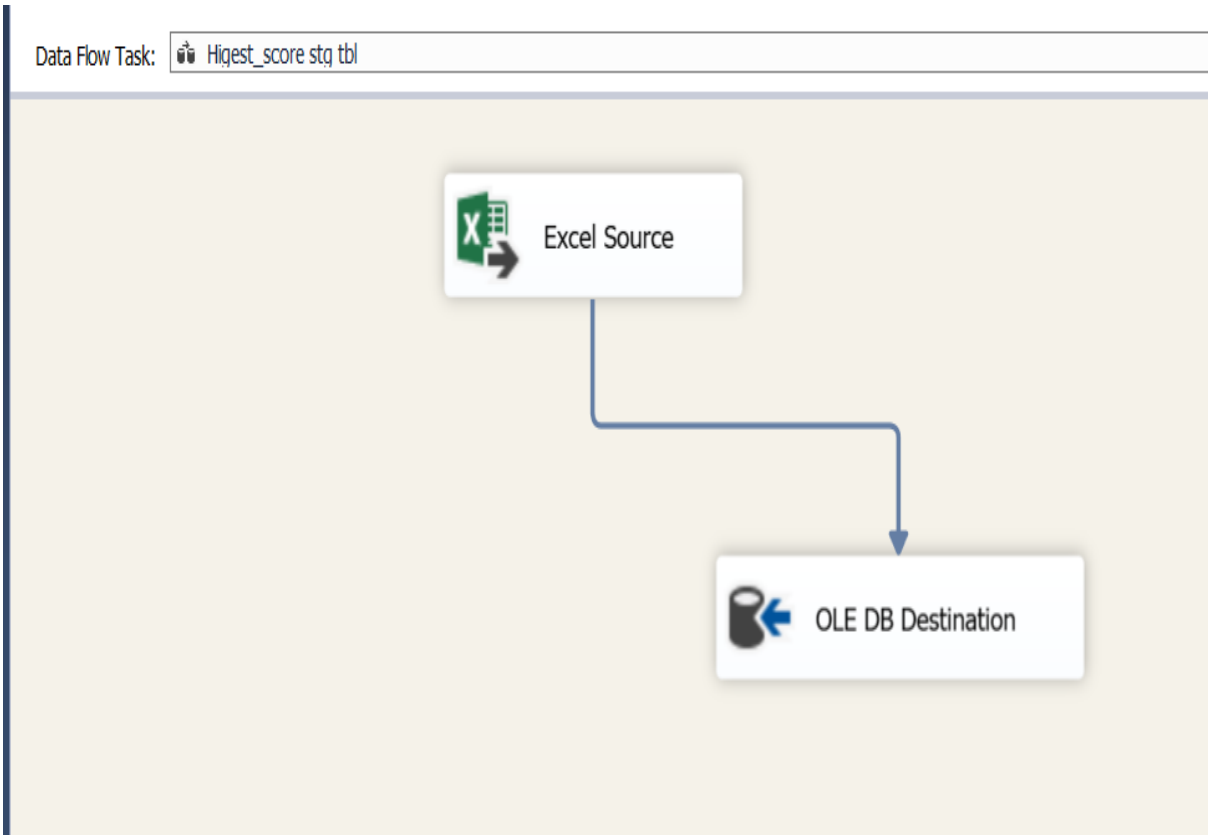




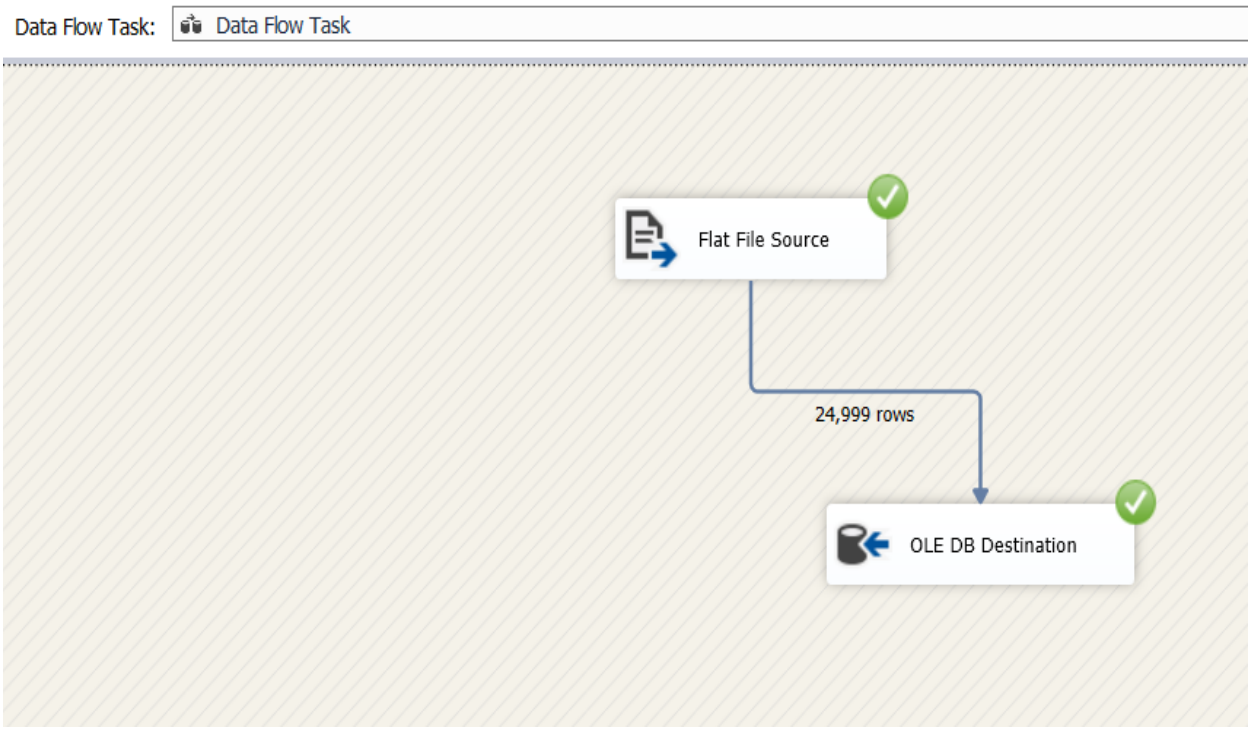
Extracting data from .csv file



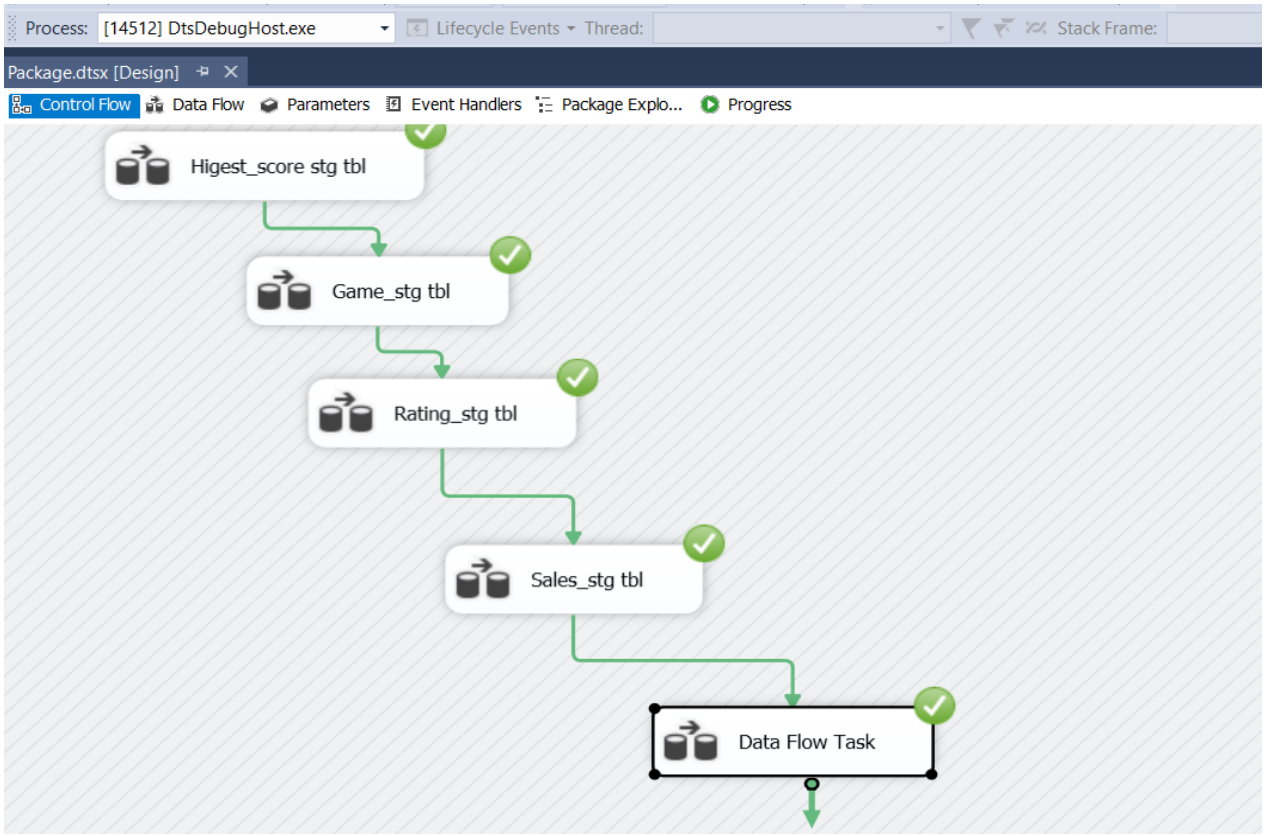

Data Flow Task:
Sales_stg tbl
Excel Source
24,999 rows
OLE DB Destination


Data Flow Task:
Higest_score stg tbl
Excel Source
OLE DB Destination

Extracting data from .txt file



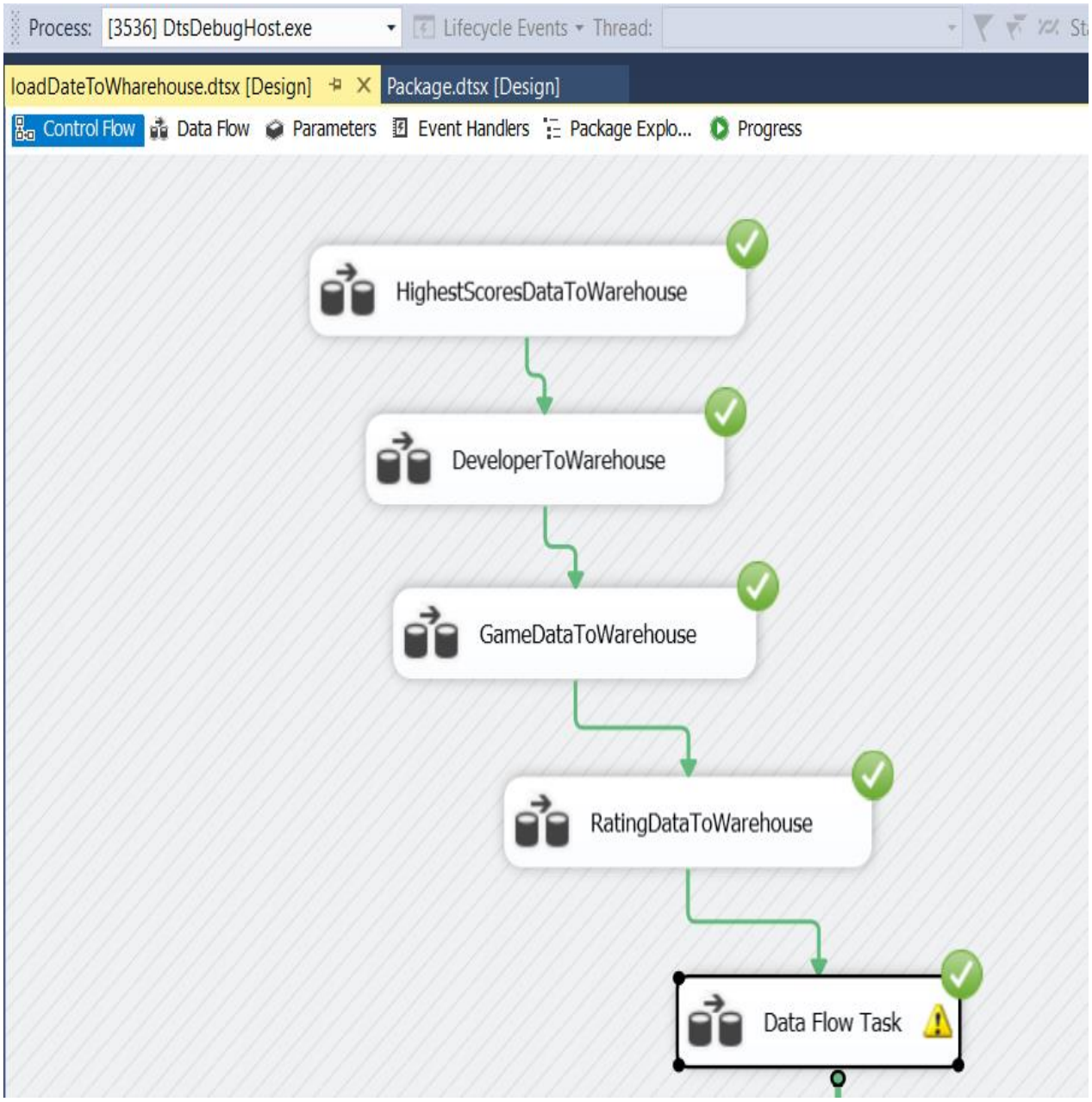
Entire Data Staging Map



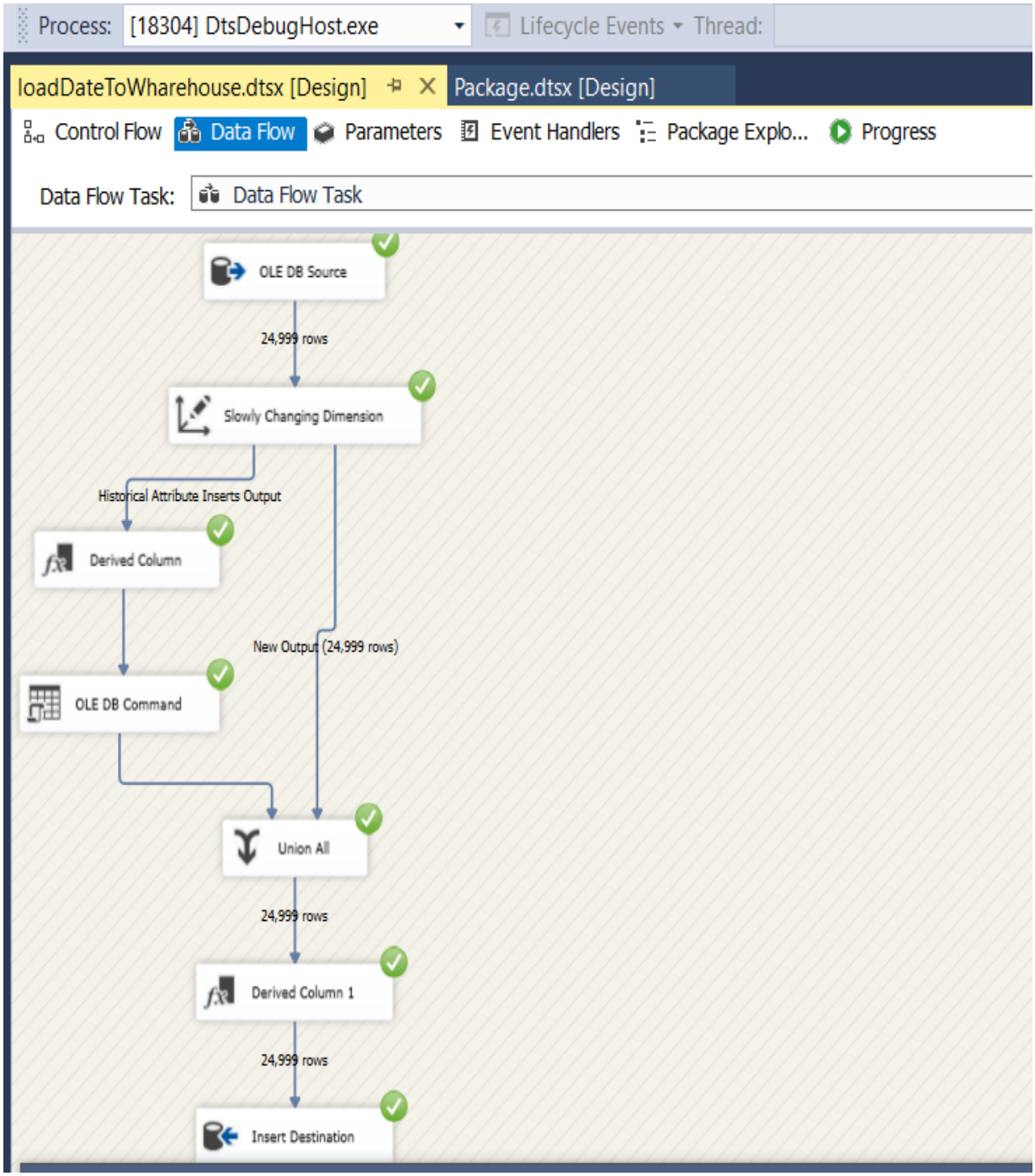
Transformation and Loading Stages

The data in the staging database was used to start the transformation layer. The database's data is washed, checked, converted, and then incorporated into the warehouse's data base.

Entire Data Transformation and Loading map

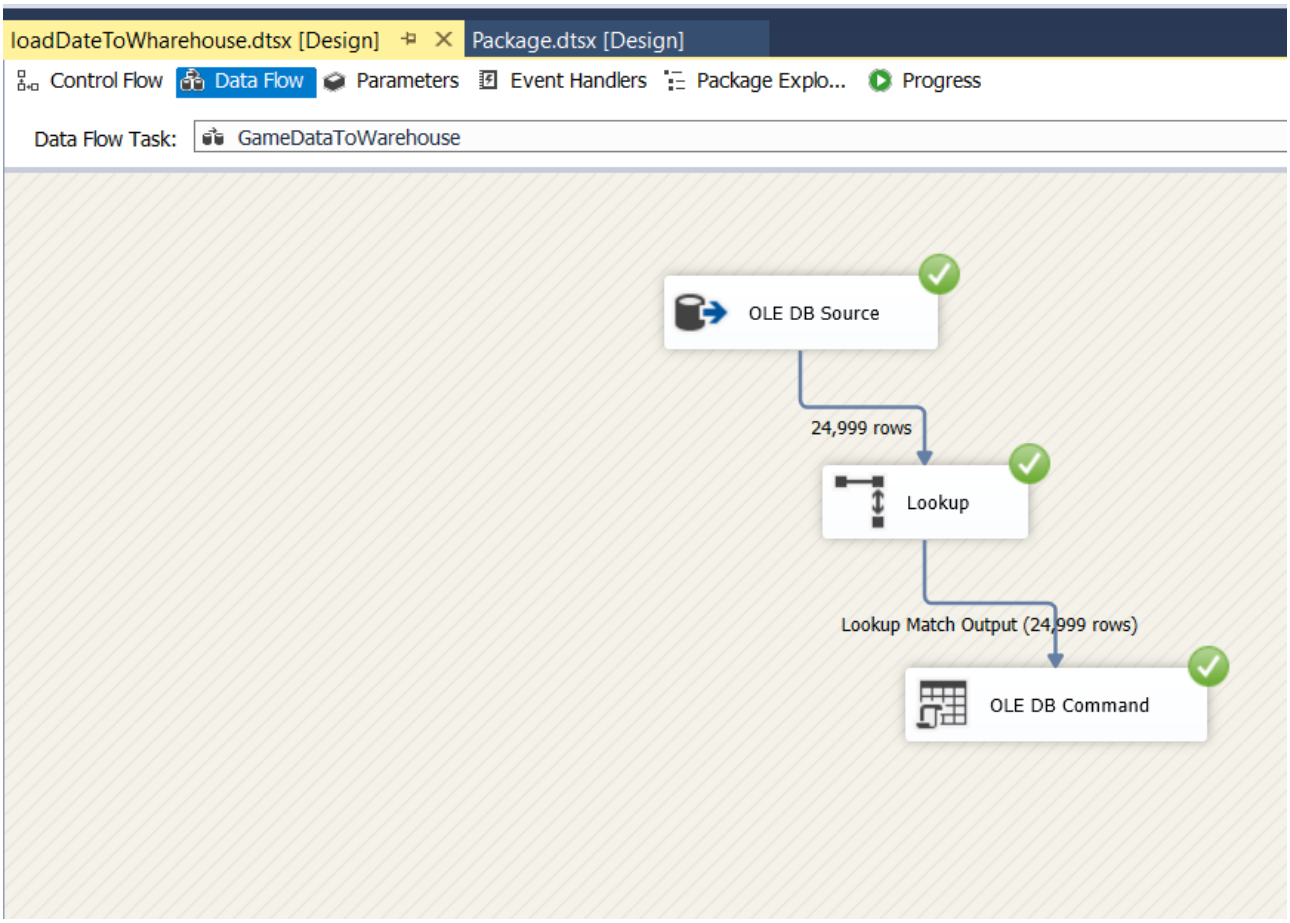


Initiating the Dim_Rating dimensional table



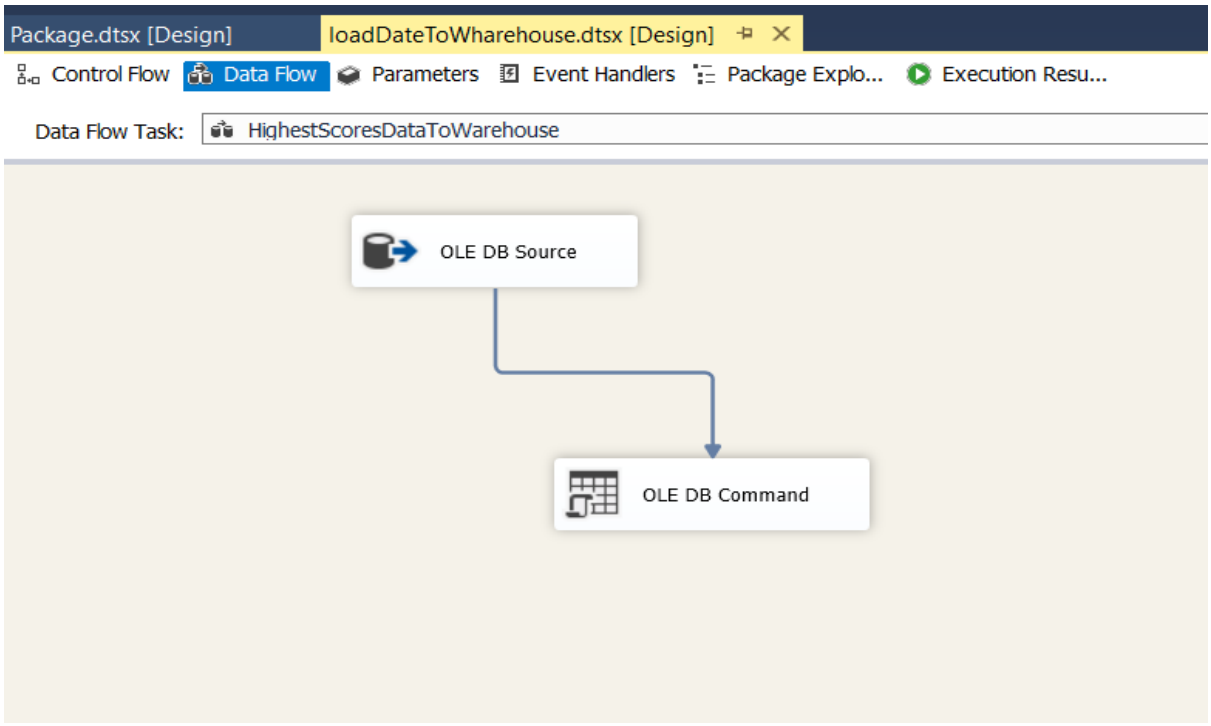
When I am loading data from RatingStg to Dim_Rating I took this as a slowly changing dimension table . . As the result of that the program can identify the changes happen and it can update the Data warehouse with placing historical data as CurrentRank_startData and CurrentRank_endDate.

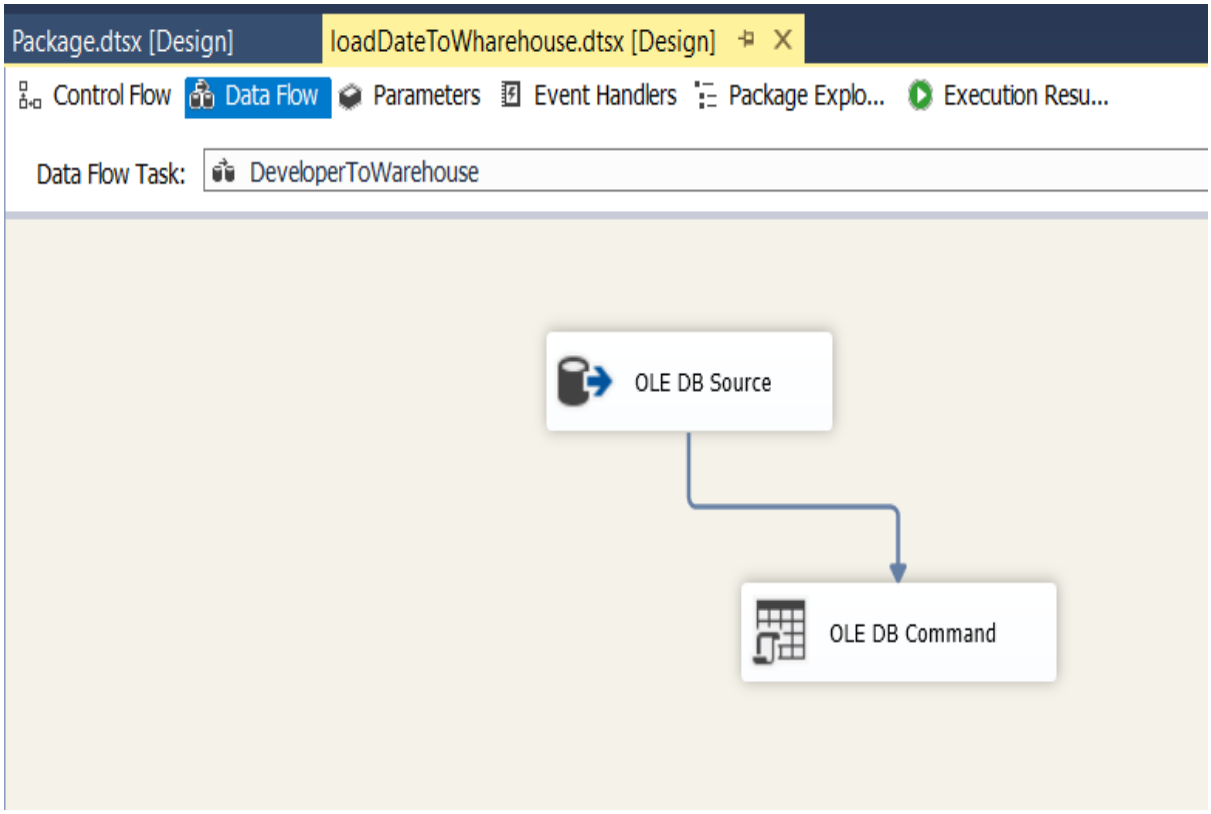
Initiating the Dim_Game dimensional table



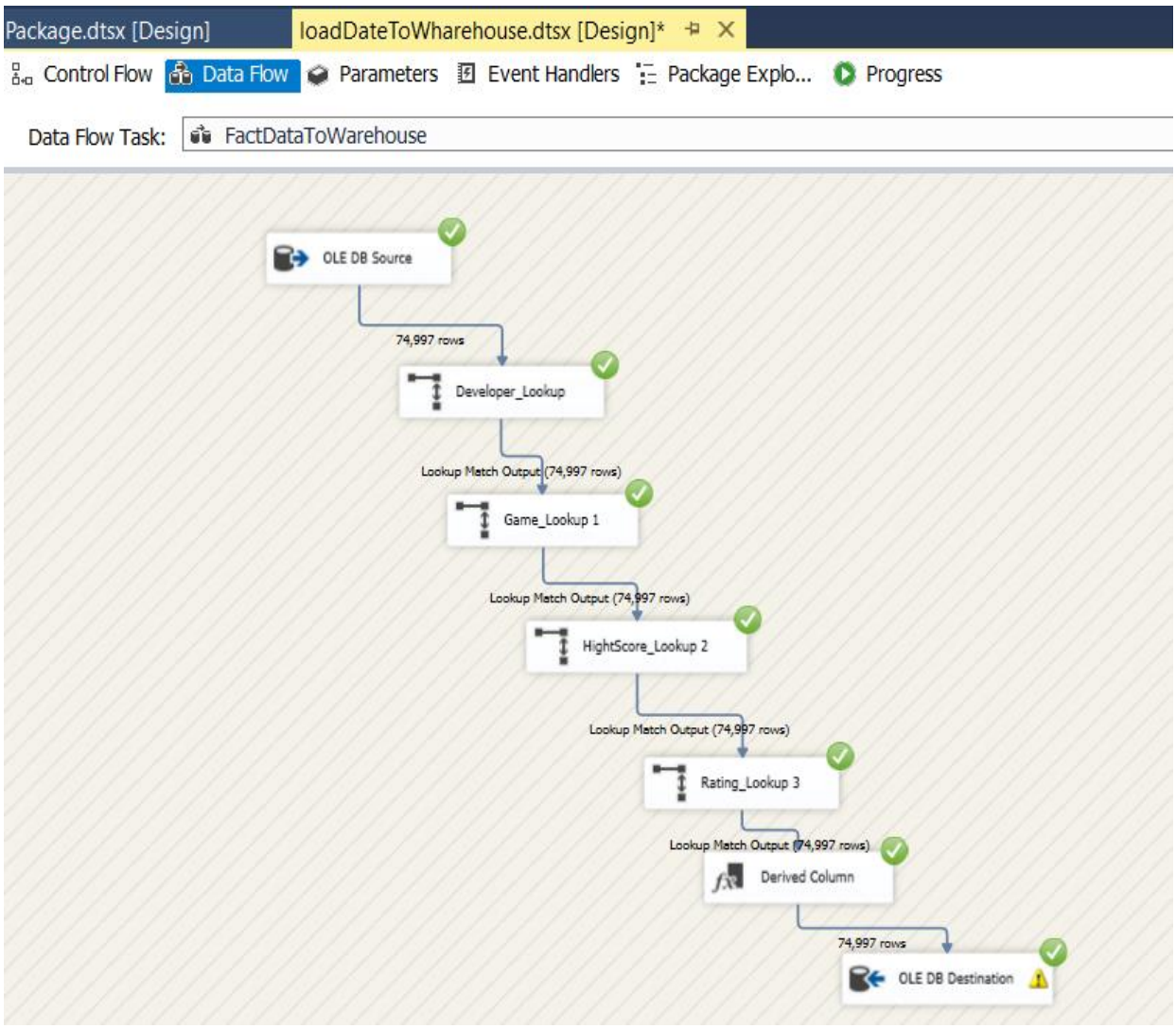
When I am loading Data from GameNew_stg to Dim_Game table I used lookup to connect Published date column with Dim_Date table .

Initiating the Dim_Developer and Dim_HighestScore dimensional table





Initiating the Fact_Sales_tbl FactTable



For merging all dimensions tables with fact table I used lookups , and some null values in my SalesStg table so I used derived column and replace ‘0’ for all null values in my fact table .

Creating procedures to ensure that recursion does not occur

- For developer Dimension Table

SQLQuery2.sql - not connected | SQLQuery1.sql - DE...-8THTS28\Dell (56))*

```
USE [vedioGames_WH]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDeveloper]    Script Date: 08/05/2021 15:36:10 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDeveloper]
@Developer_ID nvarchar(255),
@Game_Name nvarchar(255),
@D_Name nvarchar(255),
@Publisher_Name nvarchar(255)
AS
BEGIN
if not exists (select Dev_SK
from dbo.Dim_Developer
where Developer_ID= @Developer_ID)
BEGIN
insert into dbo.Dim_Developer
(Developer_ID ,Game_Name, D_Name,Publisher_Name, InsertDate, UpdateDate)
values
(@Developer_ID, @Game_Name,@D_Name,@Publisher_Name ,GETDATE(), GETDATE())
END;
if exists (select Dev_SK
from dbo.Dim_Developer
where Developer_ID = @Developer_ID)
BEGIN
update dbo.Dim_Developer
set Game_Name = @Game_Name,
D_Name = @D_Name,
Publisher_Name = @Publisher_Name,
UpdateDate = GETDATE()
where Developer_ID = @Developer_ID
END;
END;
```

- For Game Dimension Table

SQLQuery4.sql | SQLQuery3.sql - DE...-8THTS28\Dell (58)) | SQLQuery1.sql - DE...-8THTS28\Dell (56))*

```
USE [vedioGames_WH]
GO
/****** Object:  StoredProcedure [dbo].[UpdateGame]    Script Date: 08/05/2021 19:49:23 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateGame]
@Game_ID nvarchar(255),
@Name nvarchar(255),
@Genre nvarchar(255),
@Platform nvarchar(255),
@Year datetime
AS
BEGIN
if not exists (select Game_SK
from dbo.Dim_Game
where Game_ID = @Game_ID )
BEGIN
insert into dbo.Dim_Game
(Game_ID,Name,Genre,Platform,Year,InsertDate,UpdateDate)
values
(@Game_ID, @Name,@Genre,@Platform,@Year,GETDATE(), GETDATE())
END;
if exists (select Game_ID
from dbo.Dim_Game
where Game_ID = @Game_ID )
BEGIN
update dbo.Dim_Game
set Name = @Name,
Genre= @Genre,
Platform = @Platform,
Year = @Year,
UpdateDate = GETDATE()
where Game_ID  = @Game_ID
END;
END;
```

- For HighestScore Dimension Table