|  |  |
| --- | --- |
| Image result for csueb logo | **BAN 622 – Data Warehousing and BI** |
| **Project 3**  **Points: 20** |

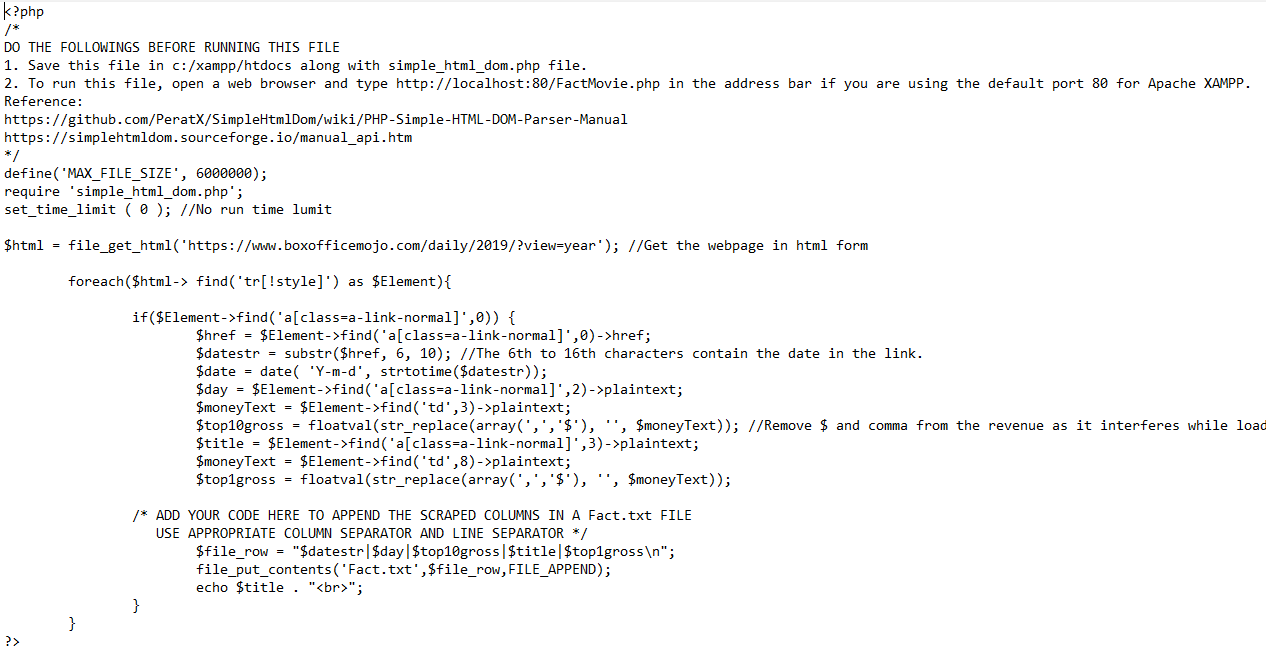
**Note: Below contains responses to the questions and screenshots for the responses. A separate word document will contain just the deliverable screenshots.**

**Data Scraping**

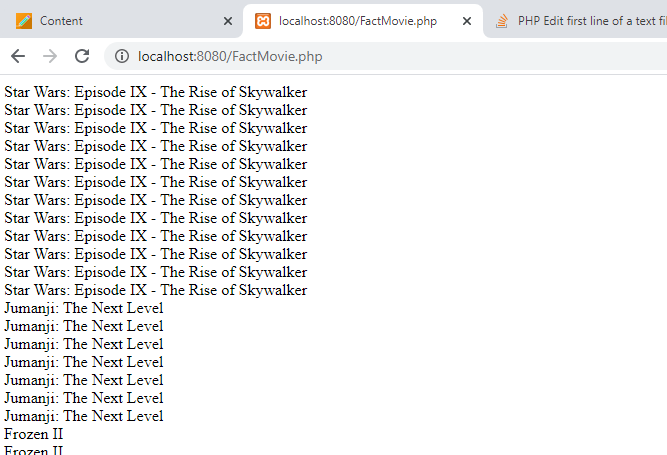
You need to scrape movie data from two separate websites. Use FactMovie.php file to scrape day wise revenue of top grossing movies of 2019. Before running the file, you need to add appropriate code in the FactMovie.php file to store the scraped data in a **Fact.txt** file. Following is the website from where you will scrape the relevant information.

<https://www.boxofficemojo.com/daily/2019/?view=year>

**Editted FactMovie.php file**

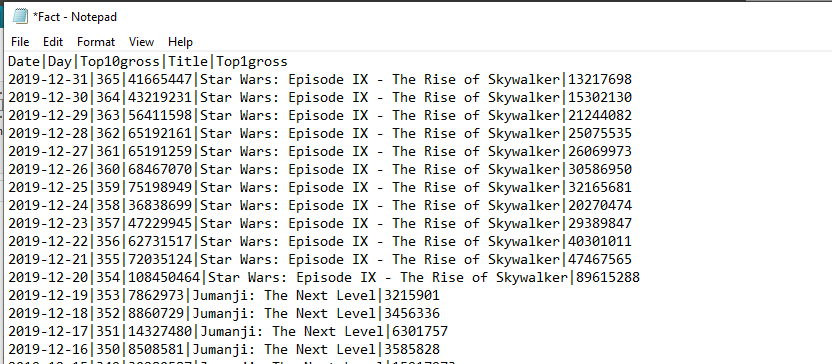


**Execution and Completion of PHP script (simple\_html\_dom.php is also in htdocs)**



After scraping, add the header columns Date, Day, Top10gross, Title, and Top1gross in the first line of the Fact.txt table. Use the same column separator you used while appending each row. Note that the sequence of the header columns should be same as the values appended. The Fact.txt file should contain 365 rows of values.

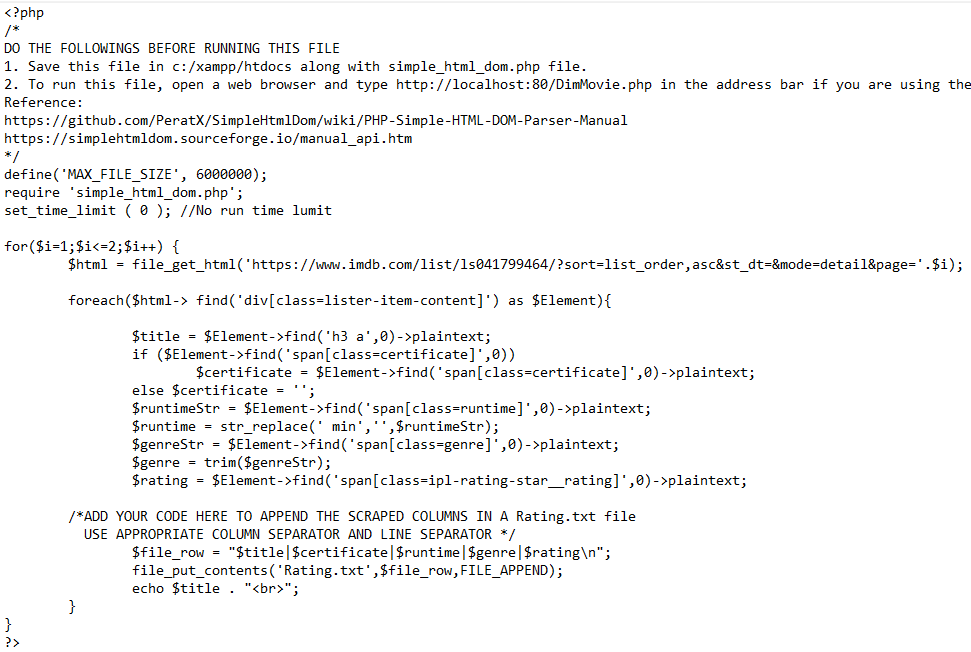
**Adding the header column names to Fact.txt**

****

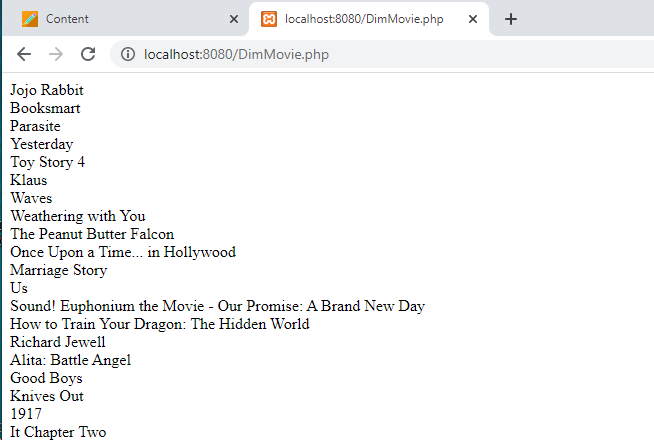
Use DimMovie.php file to scrape movie specific information of year 2019. Following is the website from where you will scrape the data.

<https://www.imdb.com/list/ls041799464/?sort=list_order,asc&st_dt=&mode=detail&page=1>

**Editted DimMovie.php file**

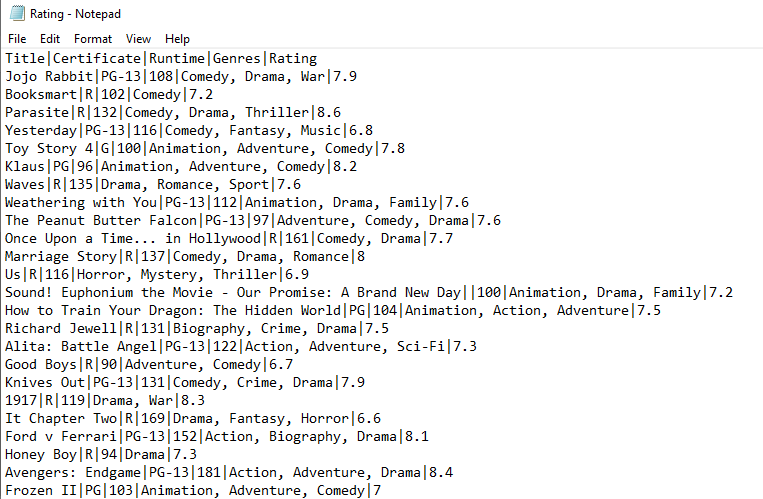
****

**Execution and Completion of PHP script (simple\_html\_dom.php is also in htdocs)**



You will store the scraped data in **Rating.txt** file. Add the header columns Title, Certificate, Runtime, Genres, and Rating in the Rating.txt file. This files should contain 143 rows of values.

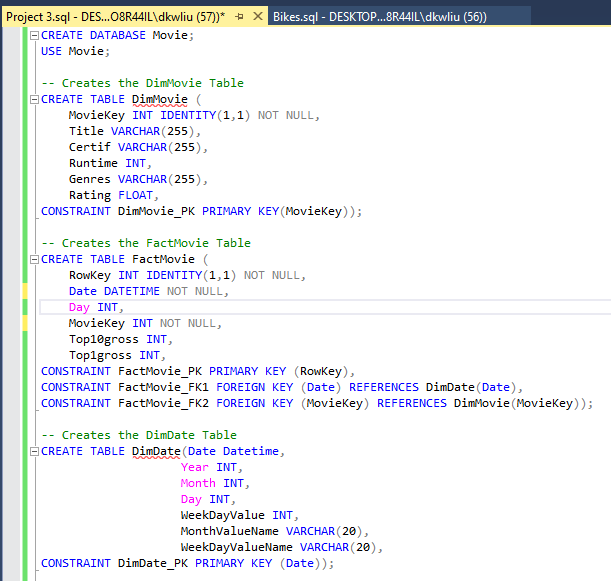
**Adding the header column names to Rating.txt**

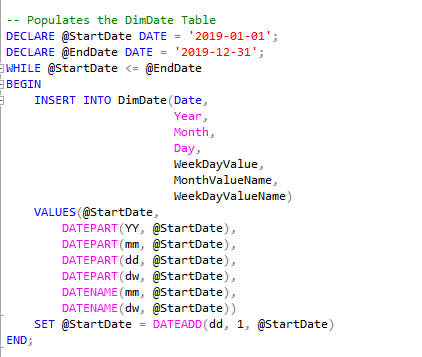


**Movie Database**

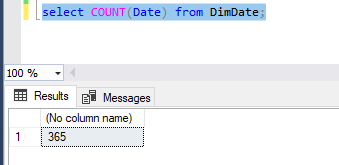
In SQL Management Studio (SSMS), create a **Movie** database containing DimMovie, DimDate, and FactMovie tables. Name this file as **Project 3.sql**. DimMovie should have a MovieKey which is auto generated. It should also contain the title, certificate, runtime, genres, and rating columns. Remember that the size of the title and genres should be large enough to accommodate long values. DimDate should have 365 rows corresponding to each day of the year 2019. You can use the date dimension query used in the classroom exercise to populate this dimension. FactMovie table should contain an auto generated RowKey, Date, Day, Top10Gross, Top1Gross, and appropriate foreign key.

**SQL codes for creating DimMovie, DimDate, and FactMovie**



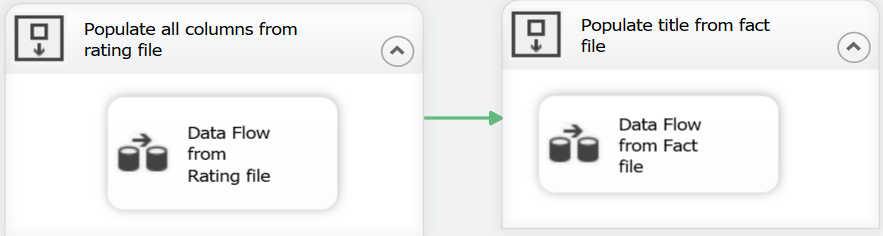


**Screenshot of the number of rows in the DimDate table**

****

**Populating the Database**

In SQL Server Data Tools (SSDT), create a multidimensional SSIS project to populate the DimMovie table. Drag two **Sequence Containers** in the Control Flow window. Then drag a Data Flow task inside each of these containers. Connect the first container to the second by dragging the green arrow as shown below.



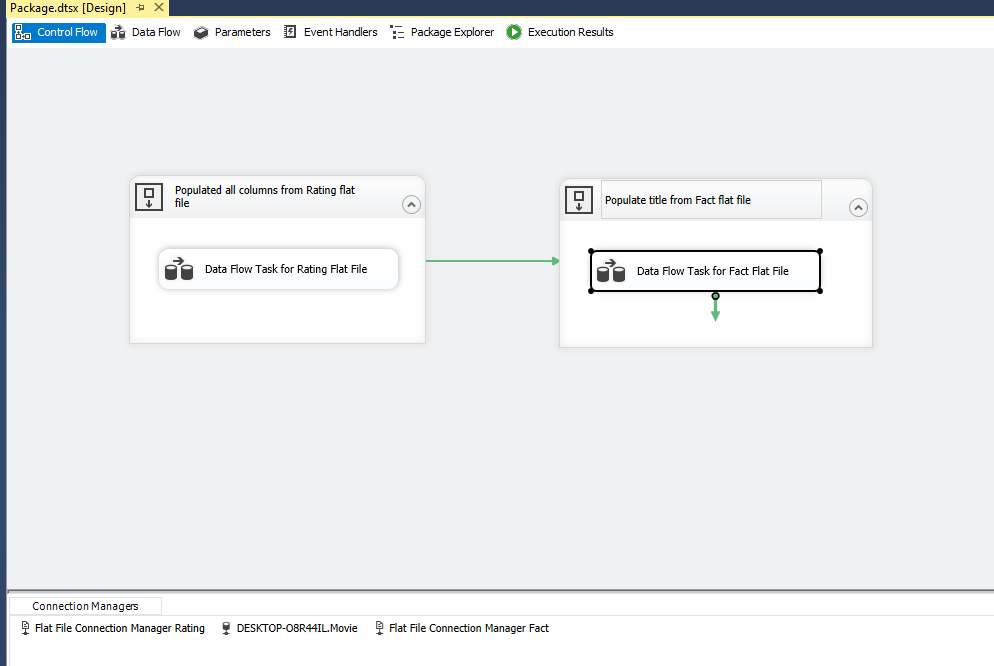
The first data flow task will populate the DimMovie dimension from Rating.txt file. Once the data is added to the dimension, the data flow task in the second container will automatically execute and populate the remaining movie titles from Fact.txt.

Double clicking the first data flow task will open the Data Flow window where you create a model to populate all 143 rows from Rating.txt to the DimMovie dimension. While creating a connection to the flat file in the connection manager editor, change the **Output Column Width** in the **Advance** tab for Title and Genre to the same size as in the DimMovie dimension. Remember to check the box for column names in the first data row. Also retain the null values from the source file while transforming. Use a lookup tool to append new movie titles which are not already been populated in the dimension. Remember to use **partial cache** mode while looking up for the title. Partial cache is case insensitive for lookup value.

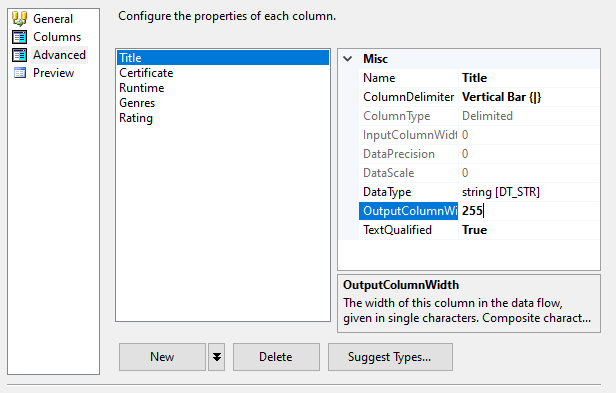
Similarly, use the second data flow task to populate unmatched movie title from Fact.txt to the movie dimension. Remember to use **Sort** tool to remove duplicate movies while reading the source file. Also use partial cache mode for case insensitive lookup.

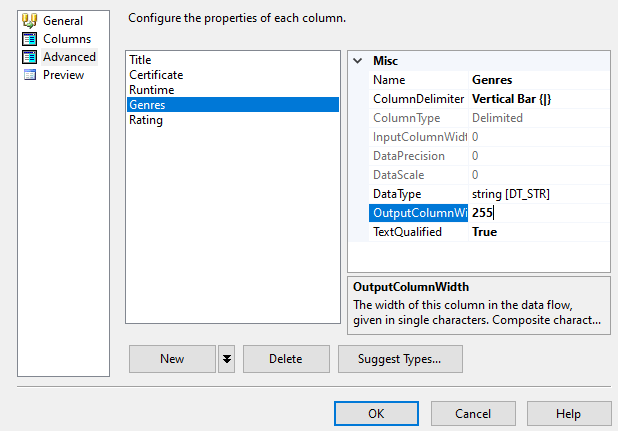
Once all the tables are populated, create a database diagram. **Take screenshots of the database diagram and both data flow models.**

**Created two sequence containers with a data flow in each to fill DimMovie**

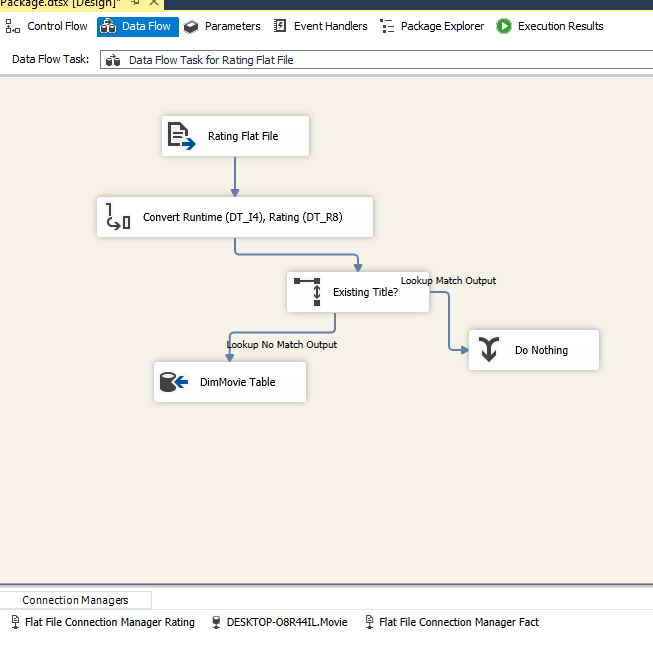


**Changing Title’s and Genres OutputColumnWidth in Rating.txt to 255 (Same size as the DimMovie columns)**

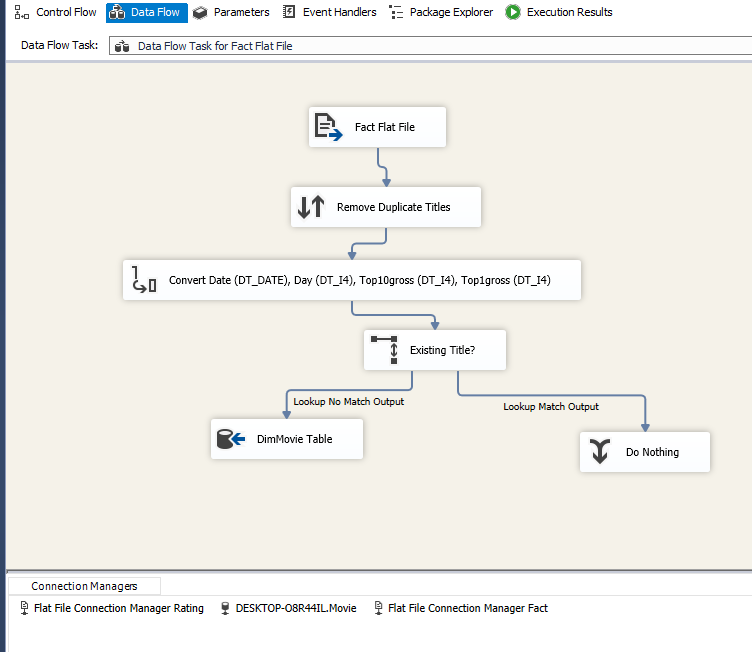
****

****

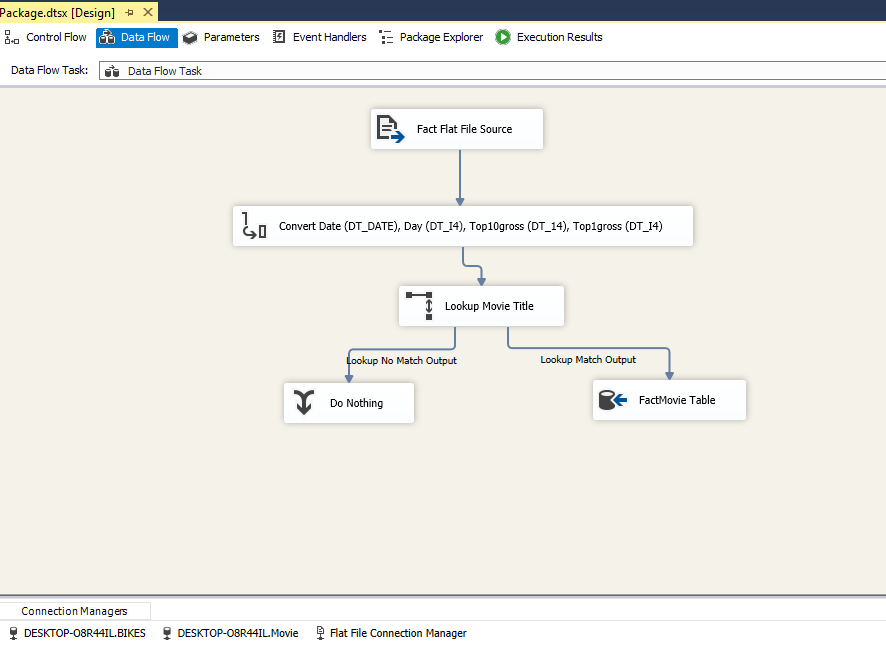
**Data flow 1, which populates DimMovie table with Rating data**

****

**Data flow 2, which populates DimMovie table with Fact data titles**

****

**Data flow for inserting the Fact.txt flat data into the FactMovie table**

****

**Database diagram of the Movie database**

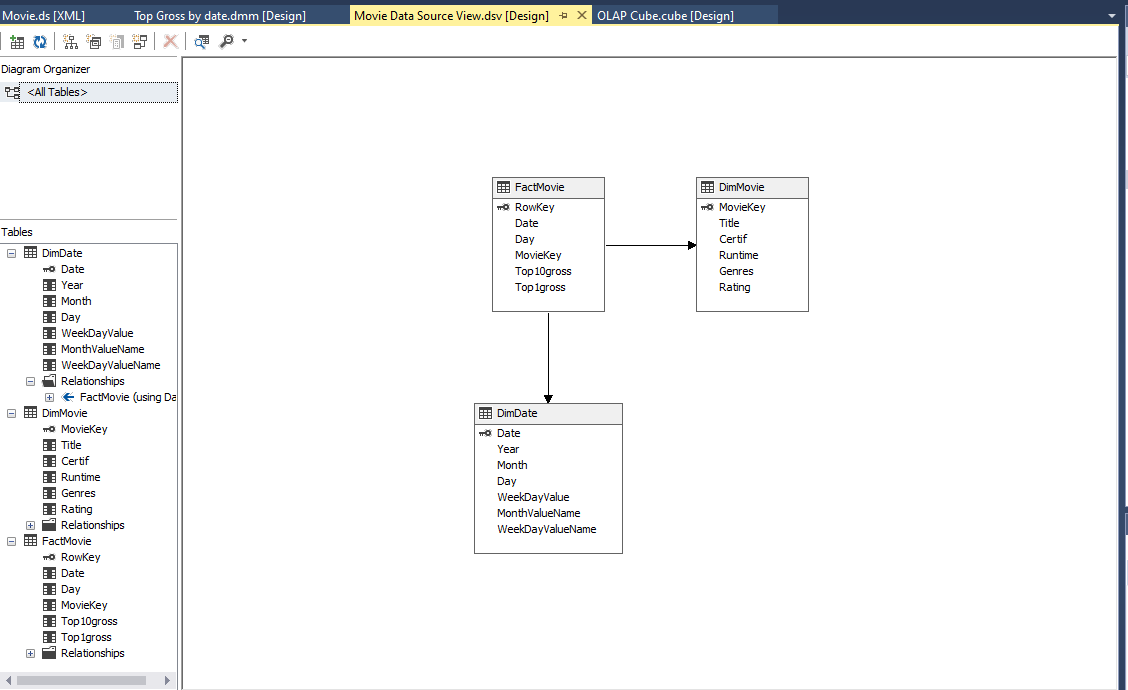
****

**Data Mining**

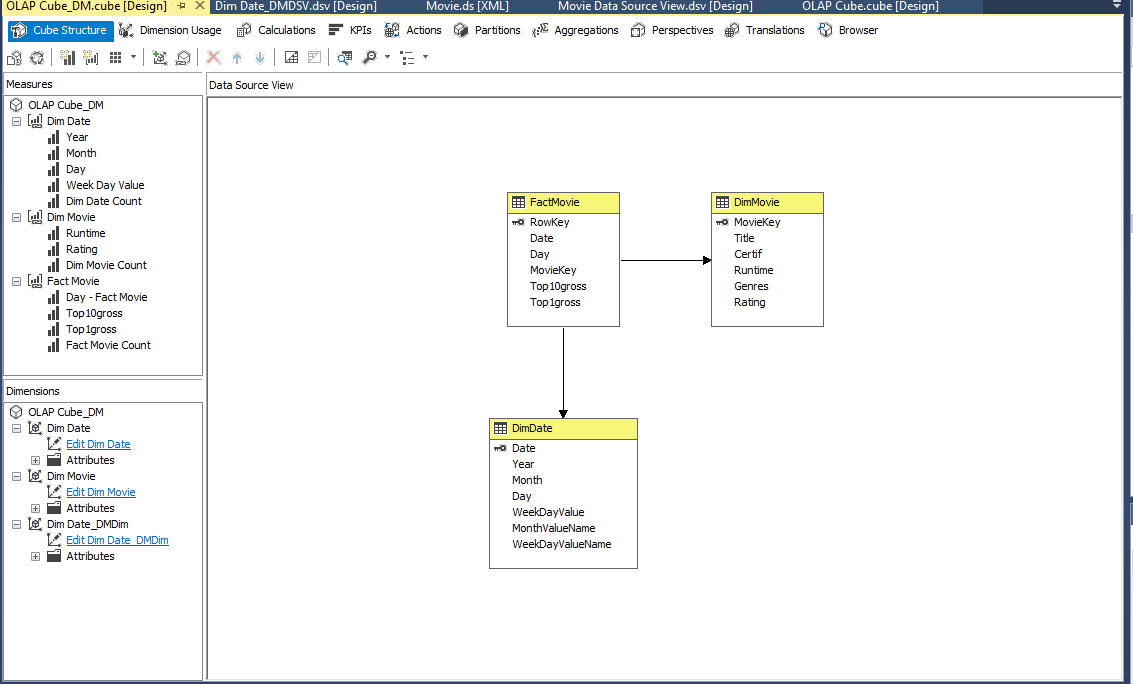
1. **Time Series Analysis**: Following video explains how to create a time series project in SSAS. <https://www.youtube.com/watch?v=UCXDR2VXnQM>.

Connect to the Analysis Services Server in SSMS. Create an Analysis Services Multidimensional and Data Mining project with the name **Time Series**. Create an OLAP cube from the movie database. Create a new mining structure by right-clicking the **Mining Structure** folder in the Solution Explorer. Use the existing cube to do the mining structure. Select Time Series mining model. Select the Date dimension as the Source Cube Dimension. Select Date as the Case Key. Select Top10Gross and Top1Gross as the Case Level Columns. Top10Gross and Top1Gross should also be used as the input and predictable attributes. You can select Date dimension to slice the source cube. Name the Data mining and the model as **Top Gross by date**. Check the box to create cube using mining model. Right-click the ‘Top Gross by date.dmm’ data mining model in the solution explorer and click Process to deploy the model. Click Run to process mining structure after successful deployment.

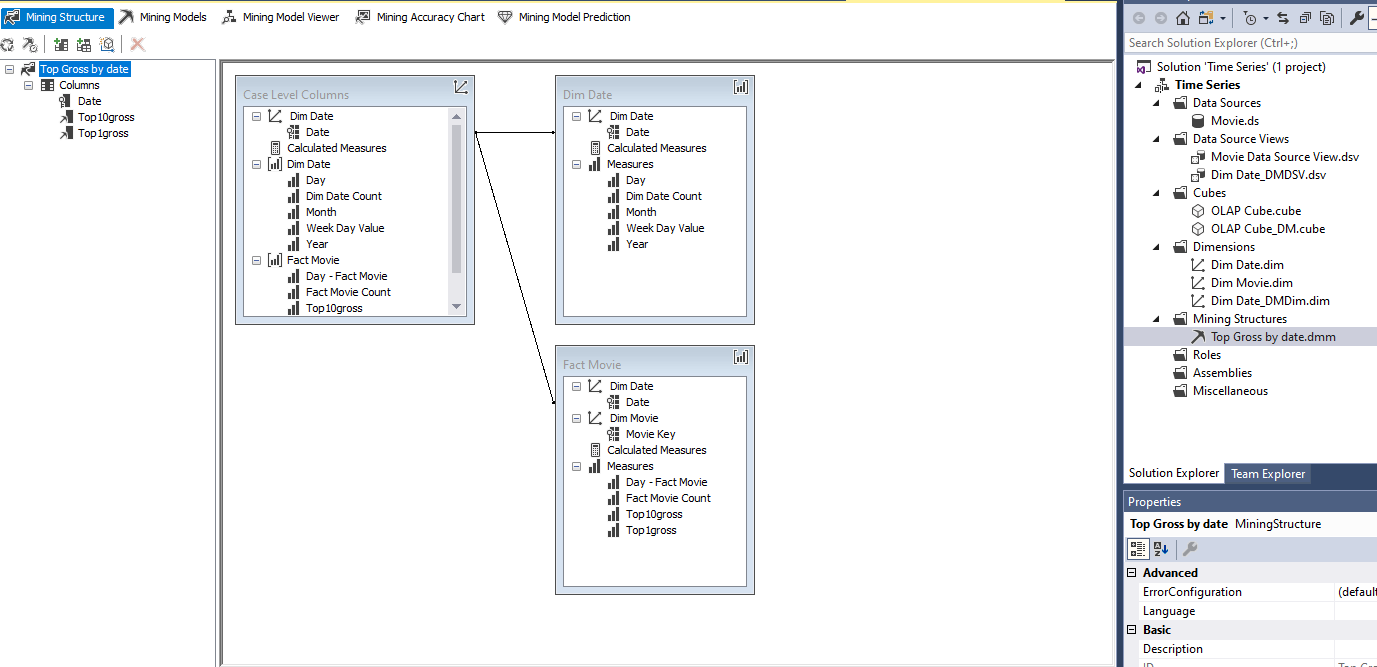
**Data Source/Data Source View**



**OLAP Cube**

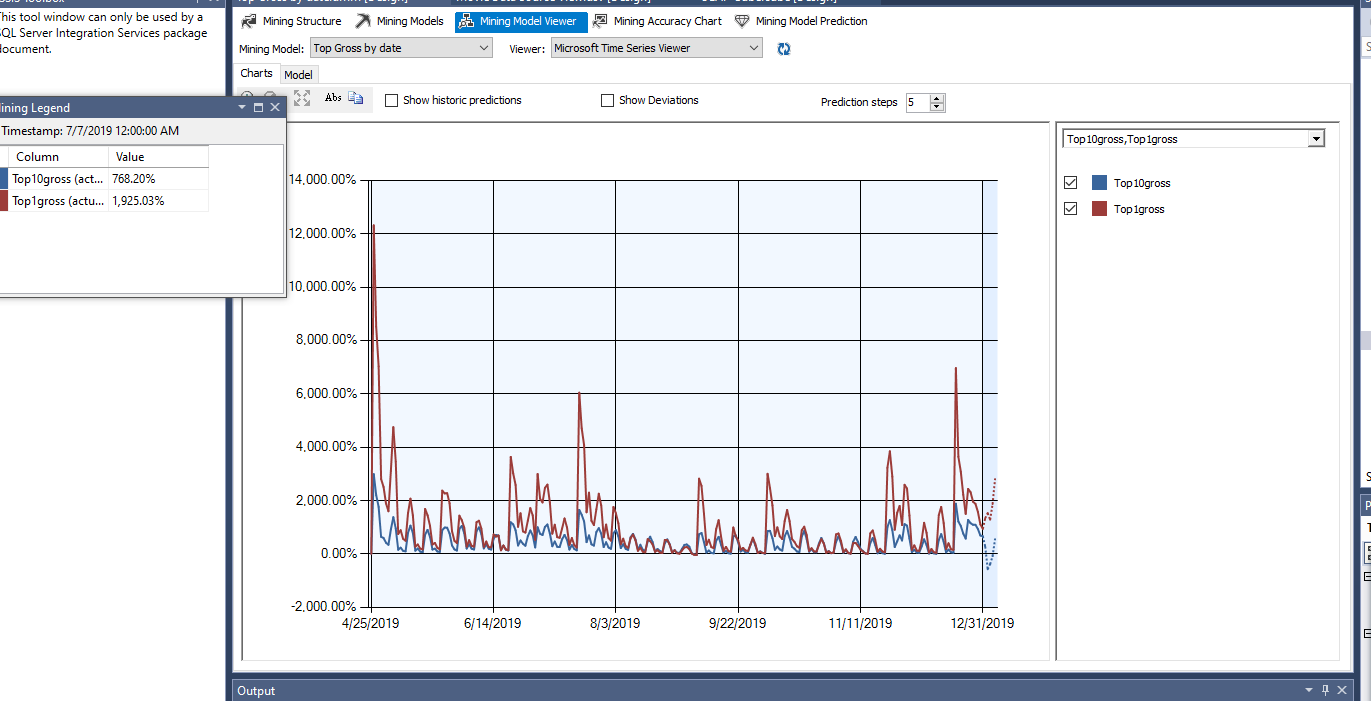
****

**Time Series Mining Structure (Used Existing Cube)**

****

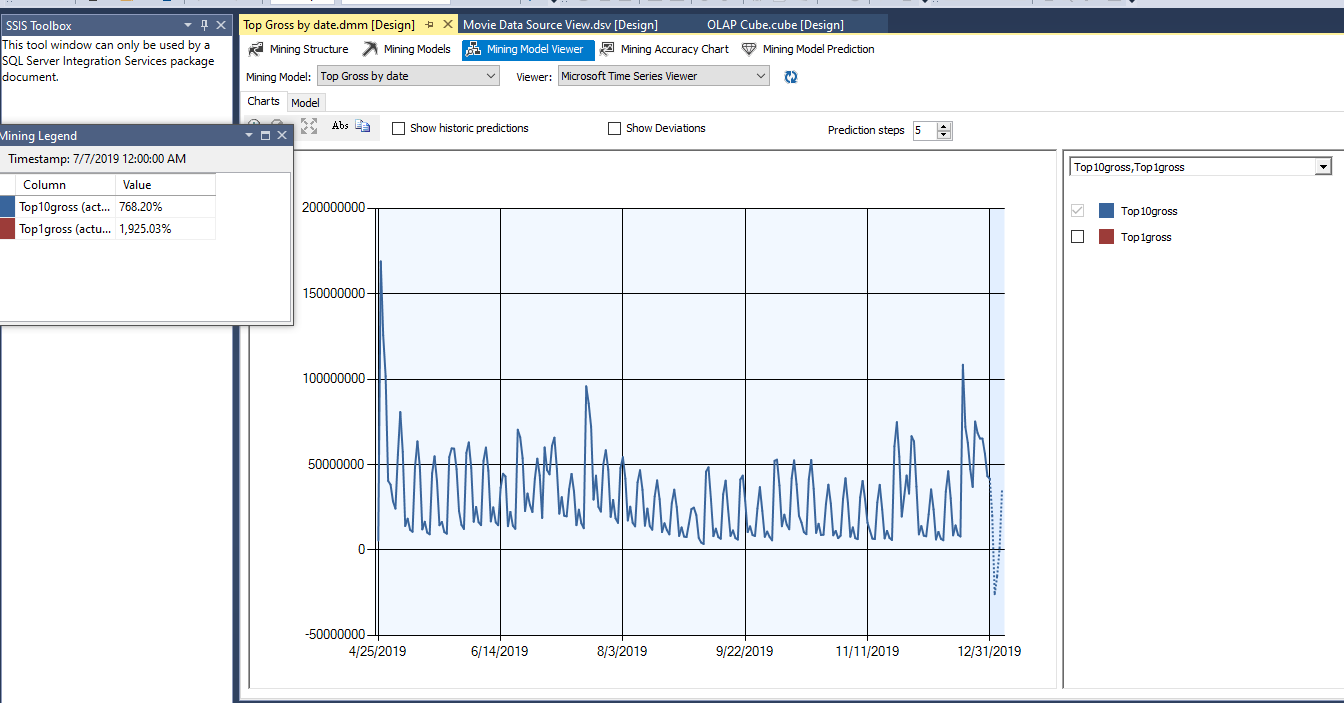
Open the **Mining Model Viewer** tab. **Take a screenshot of the time series chart.**

**Screenshot of the time series chart created from the Movie database OLAP cube**

**

Now uncheck the box of Top1Gross. *What do you infer from the time series chart? Why there are periodic spikes in Top10Gross revenue? Give suitable reason for periodic cycles. What is your guess about the number of days in each cycle?*

**Screenshot of the time series chart created from the Movie database OLAP cube with Top1gross unchecked**

**

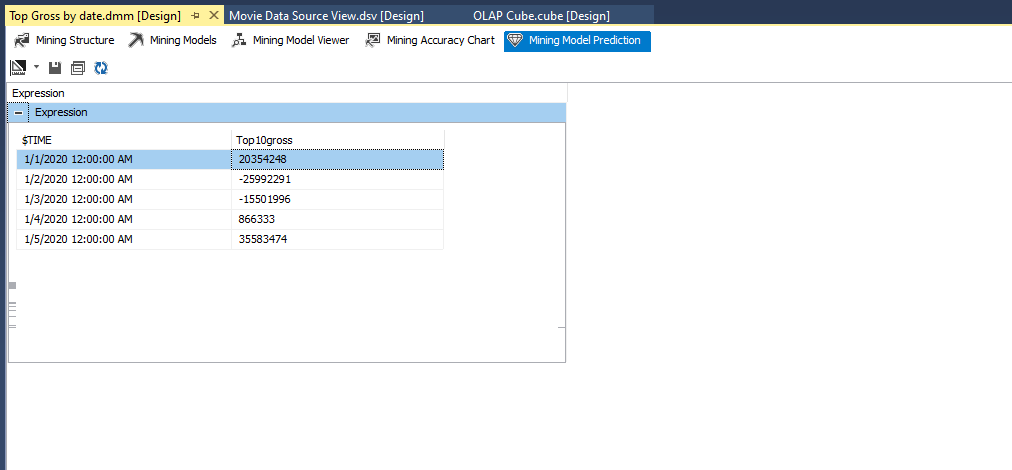
**RESPONSE:**

**From looking at the time series chart, we can see that there are several spikes in top10 box office gross earnings throughout the year. Hence, we can infer that the box office earnings happen in cycles, in which there are events that result in spikes in earnings throughout the year. These periodic cycles may be explained by new movie releases in theater, such that, when a new movie releases, it will attract a large base of customers, which will also dramatically increase the gross earning of the top 10 movies. Over time, the customer base will decrease in size as most of the customers would have already watched the movie, resulting in a lower top 10 gross earning. Since each cycle begins with a spike in earnings, and there seems to be approximately 7 spikes every 50 days, I would assume that each cycle is approximately 7 days.**

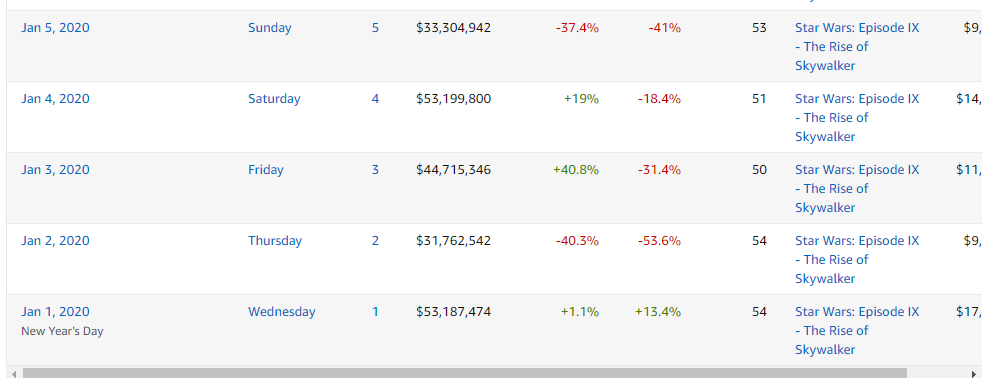
Click on **Mining Model Prediction** tab. Select Prediction Function as Source and Predict as Field. Drag and drop Top10Gross from Mining Model widow to the Criteria/Argument column, then add a comma followed by 5 as shown below. This means that you want to predict Top10Gross for next 5 days.

Click on the drop-down arrow below the Mining Structure tab and expand the expression to see the predicted revenues for next 5 days. **Take a screenshot of the predictions**. *How good is this prediction (look at the website for actual values)?*

**Screenshot of the mining model prediction for the next 5 days**



**Screenshot of the actual top 10 gross earnings**



**RESPONSE:**

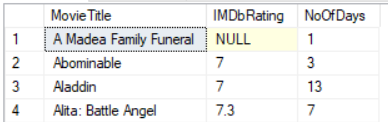
**According to boxofficemojo.com, the top 10 gross box office earnings from January 1,2020 to January 5, 2020 are $53,187,474, $31,762,542, $44,715,346, $53,199,800, and $33,304,942. In the time series prediction, for the same 5 days, we get $20,354,248, -$25,992,291, -$15,501,996, $866,333, and $35,583,474. The predicted values at January 2 and 3 (-$25,992,291 and -$15,501,996) reported losses in gross earnings, which is far off the mark from the actual values ($31,762,542 and $44,715,346). Values predicted at January 1 and 4 ($20,354,248 and $866,333) are all much lower than the actual gross earning values on the same days ($53,187,474** **and $53,199,800). The most accurate prediction would be the one for January 5, 2020, which is $35,583,474. In comparison, the actual value is $33,304,942.**

1. **Linear Regression Analysis**: Following video explains how to create a linear regression project in SSAS: <https://www.youtube.com/watch?v=MJblg9ywfwo>

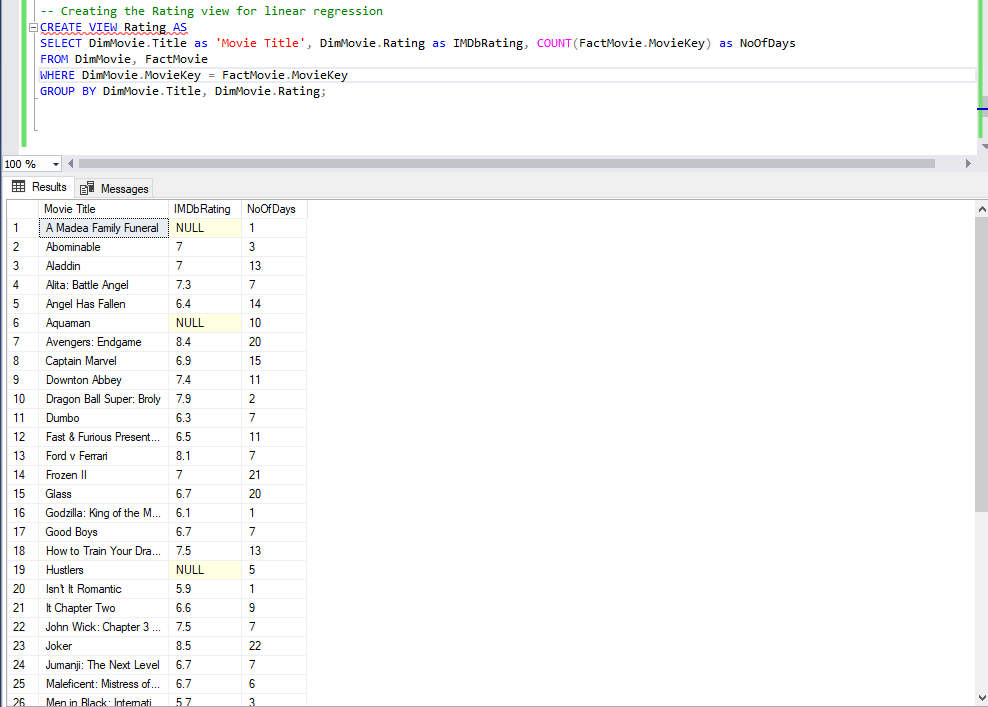
Test the following hypothesis using simple OLS regression.

*Hypothesis: Higher the movie rating, more number of days it will remain as the top grossing movie.*

To test the above hypothesis, you need to create a view named **Rating** using SQL query in the Movie database with columns Movie Title, IMDbRating, and NoOfDays. The attribute NoOfDays will contain the number of days a movie remained as the number one release in terms of gross revenue. Following are few rows of the Rating view.

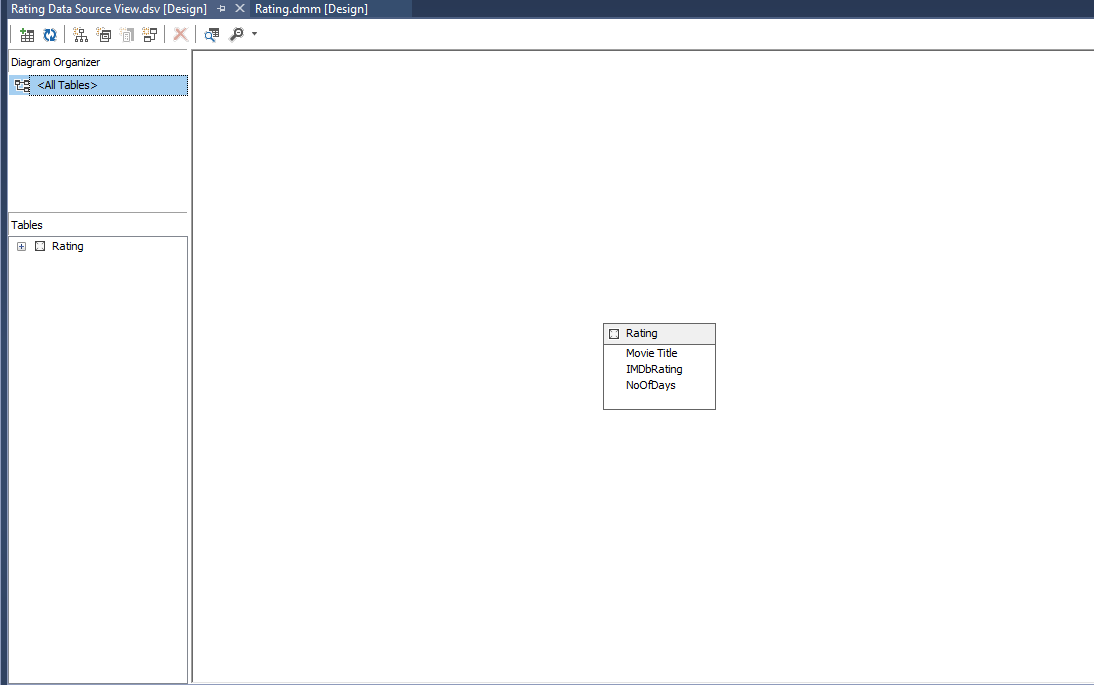


**Rating view created in the Movie database**

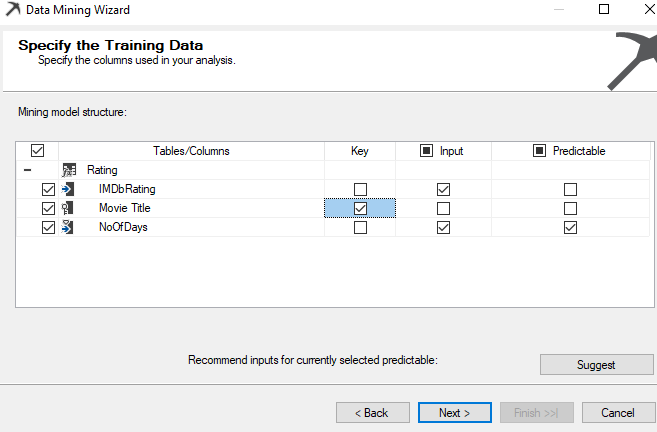


Create another SSAS Multidimensional and Data Mining project with the name **Regression**. Add the Rating view as the source. Create a linear regression data mining model with NoOfDays as the predictable attribute. Both IMDbRating and NoOfDays will be inputs and Movie Title will be the key. Use zero percentage data for testing. Deploy and process the model. Once successfully processed, click on the Mining Model Viewer tab. It will open Mining Legend windows. *What is the regression equation?* *Do the model estimates support the hypothesis (do not consider the significance level as it is not provided)?*

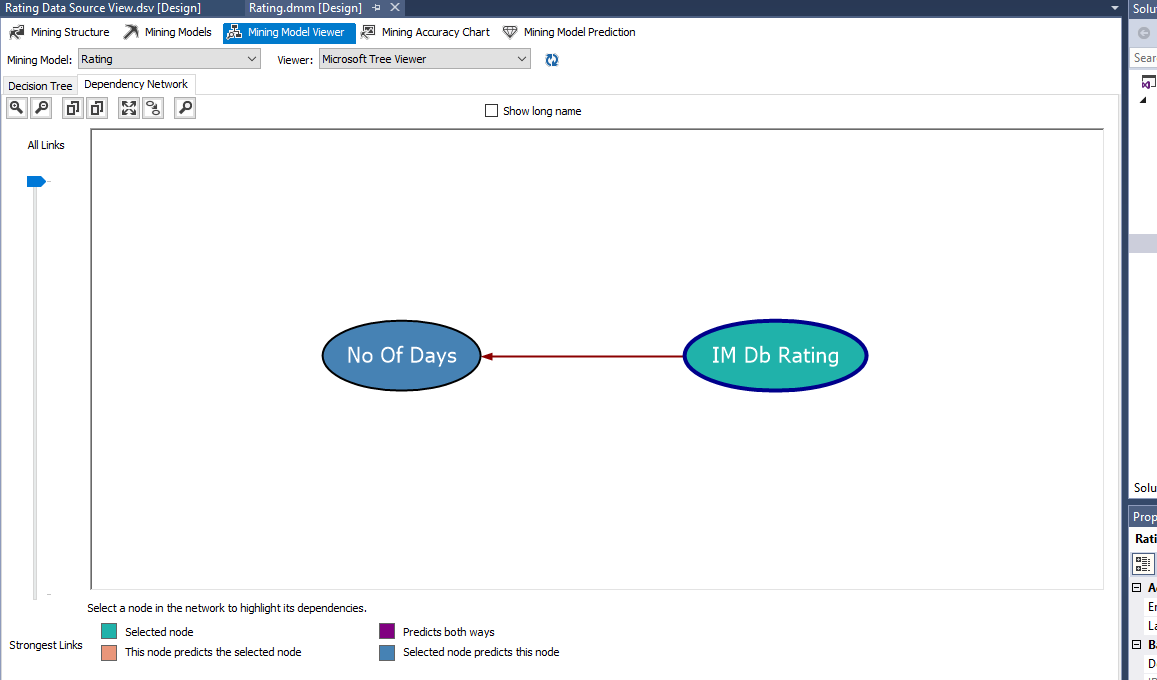
**Data source view of the data source used for regression analysis**

****

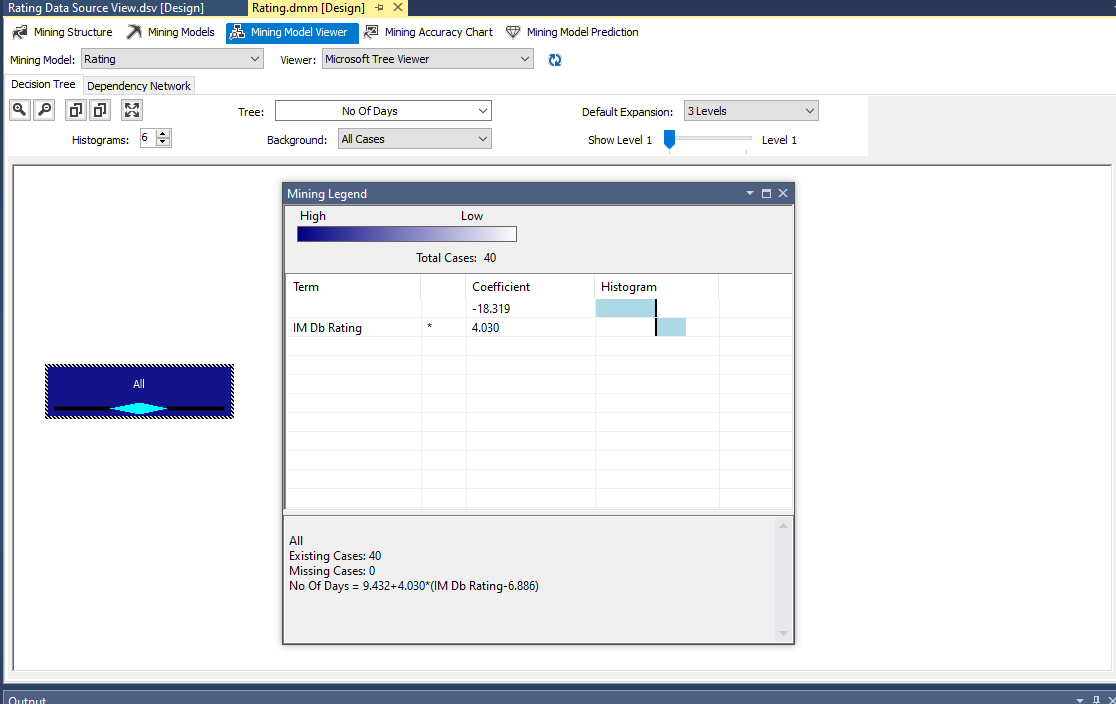
**Setting the Regression Inputs, Keys and Predictables**

****

**Dependency graph of regression model**

****

**Processed regression model**

****

**RESPONSE:**

**Shown above, the regression equation is:**

**No Of Days = 9.432 + 4.030 \* (IMDb Rating-6.886)**

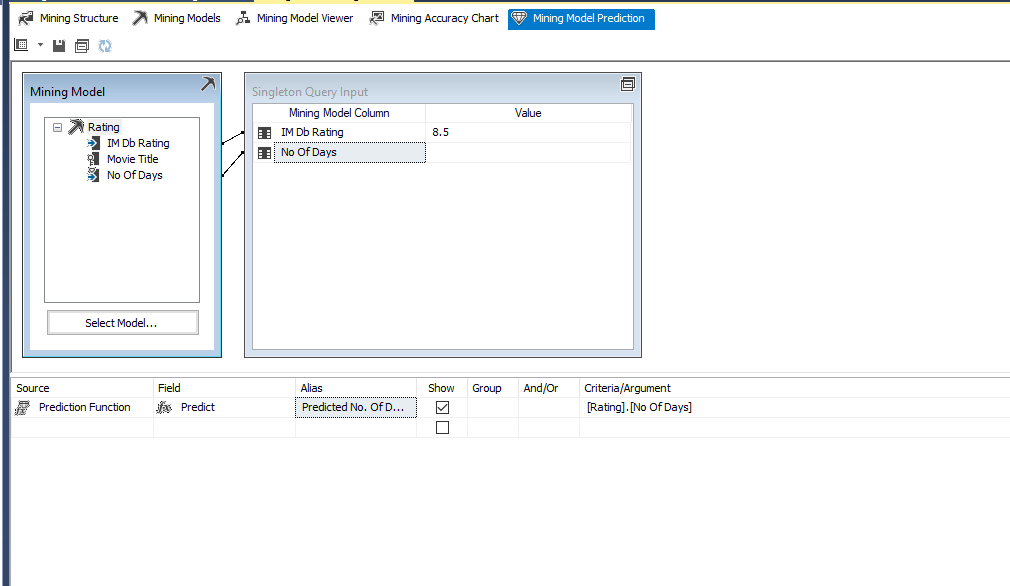
**Using the calculations below, which I hand calculated, it is shown that the number of days that a movie remains no.1 indeed increases with the ratings.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ratings** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Predicted** | **-14** | **-10** | **-6** | **-2** | **2** | **6** | **10** | **14** | **18** | **22** |

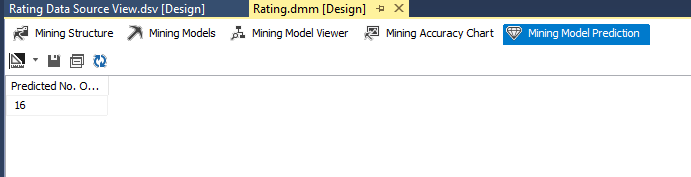
**Predicted values above are rounded to integers. As this is a linear regression function that was created while utilizing the IMDbRating and NoOfDays values, and it is shown that predicted NoOfDays values increases while IMDbRating values increases, this clearly shows that IMDbRating and NoOfDays have a positive correlation. Thus, the model estimates confirm the hypothesis.**

Click on the Mining Model Prediction tab and select Rating as the Case table. Right click to select Singleton query. Predict how long a movie with rating 8.5 will remain as the number one release. **Take a screenshot of the result.**

**Singleton Query Setup**



**Result of Singleton Query**

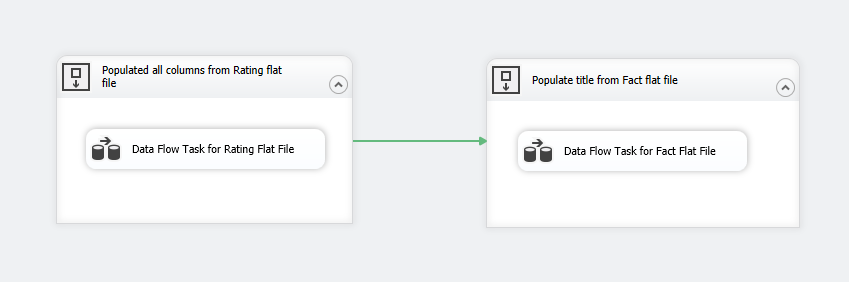
****

**RESPONSE:**

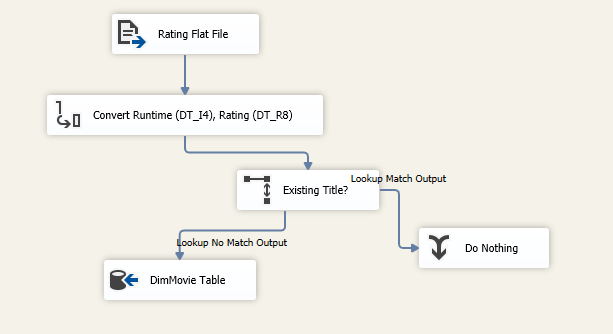
**With a movie rating of 8.5, the movie is predicted to remain number 1 for 16 days.**

**Flow Chart for DimMovie**

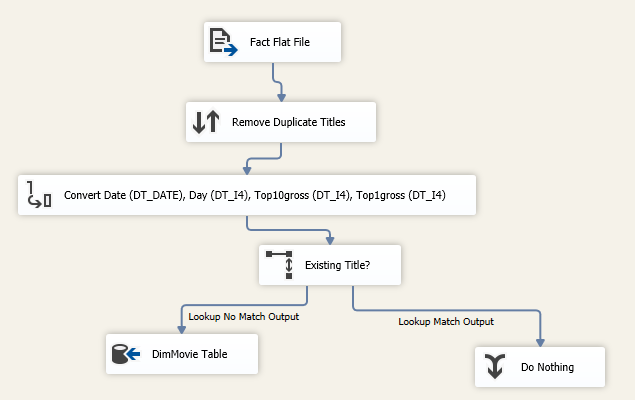
**Overall Control Flow**



**Data Flow 1**

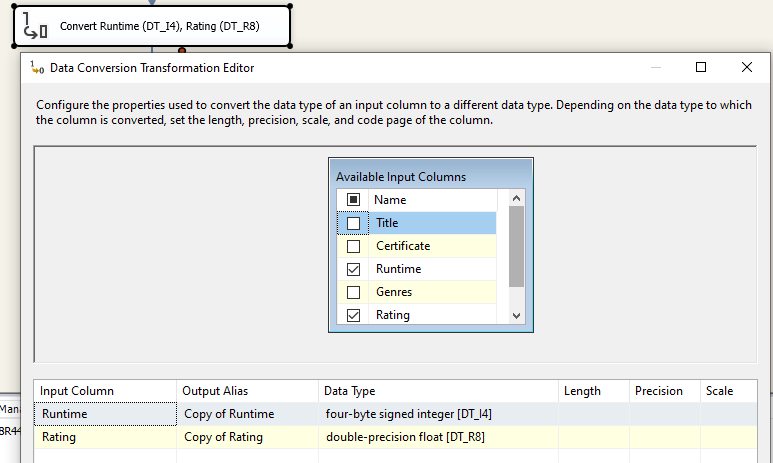
****

**Data Flow 2**

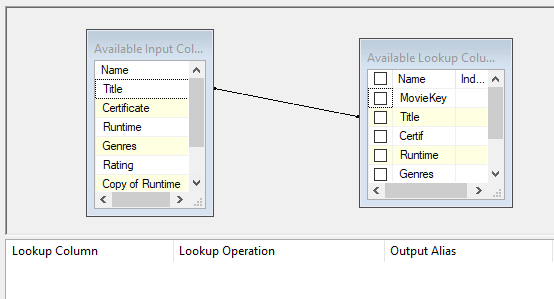
****

**Flow Steps:**

1. **Two sequence container tools (Shown above as “Overall Control Flow”) are placed in the control flow designer**
2. **Data flow 1 (Shown above as “Data Flow 1”) is placed into the first container, and a data flow 2 (Shown above as “Data Flow 2”) is placed into the second container.**
3. **A green arrow connects sequence container 1 to sequence container 2. By doing so, the completion of the processes in data flow 1 in sequence container 1 (left container) will lead into the data flow 2 processes in sequence container 2 (right container).**
4. **Data flow 1 executes**
5. **Rating.txt file data is inserted into the “Flat File Source” tool (renamed “Rating Flat File”). Included data columns are “Title”,”Certificate”,”Runtime”,”Genres”, and “Rating”.**
   1. **When inserting data from Rating.txt, the Genres and Title columns were converted to an OutputColumnWidth of 255.**
6. **The Rating.txt data from the flat file tool is then transferred to the “Data Conversion” tool, (renamed “Convert Runtime (DT\_I4), Rating (DT\_R8)”). Here, the “Runtime” data column is converted to the four-byte signed integer type, and the “Rating” data column is converted to the double-precision float type.**

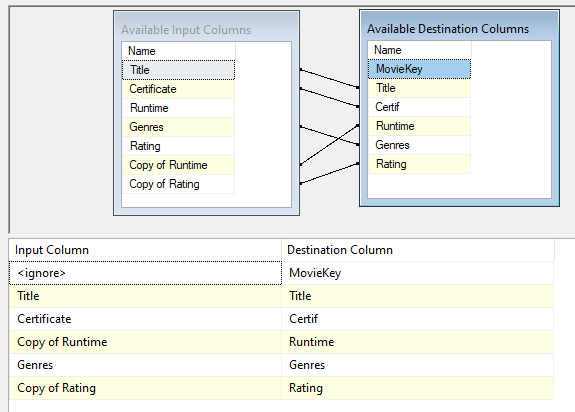
****

1. **Converted data from step 6 is transferred to a “Lookup” tool (renamed “Existing Title?”). Here, the tool checks if the values in the “Title” column of the data from step 6 matches any values in the “Title” column of the DimMovie dimension.**

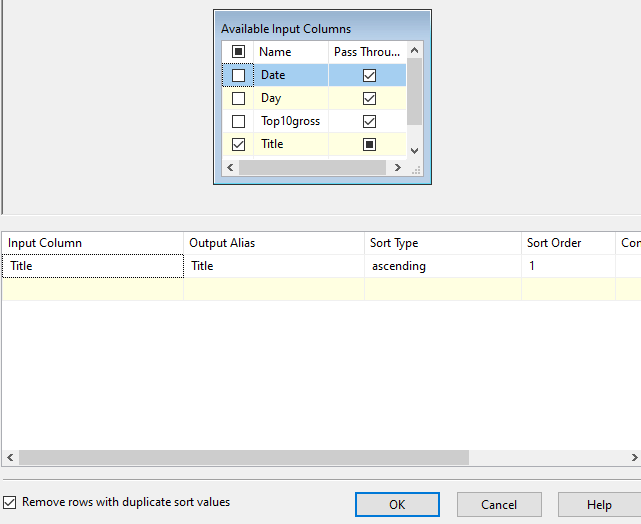
****

* 1. **Rows with a “Title” value that matches a “Title” value in DimMovie dimension will be transferred to the “Union All’ tool (renamed “Do Nothing”) where nothing will happen.**
  2. **Rows with non-matching titles will be sent to an “OLE DB Destination” tool.**

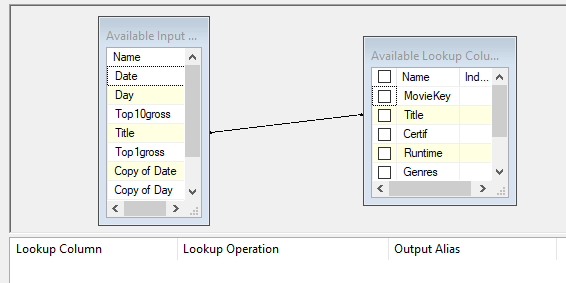
1. **The data from step 7a will be transferred to the “OLE DB Destination” tool (renamed “DimMovie Table”) where the data will be inserted into the DimMovie dimension.**
   1. **Title data will be inserted into the Title column of DimDimension.**
   2. **Certificate data will be inserted into Certif column of DimDimension.**
   3. **Converted Runtime data will be inserted into the RunTime column of DimDimension.**
   4. **Genres data will be inserted into the Genres column of DimDimension.**
   5. **Converted Rating data will be inserted into the Rating column of DimDimension.**

****

1. **Data flow 1 will transition to the data flow 2 in sequence container 2 (data flow and containers shown above).**
2. **Fact.txt data will be inserted into the “Flat File Source” tool (renamed “Fact Flat File”). Included data columns are “Date”, “Day”,”Top10gross”,”Title”, and “Top1gross”.**
3. **Data from step 10 is transferred into a “Sort” tool (renamed “Remove Duplicate Titles”). Here, the data are sorted in ascending order according to the “Title” column values. “Remove rows with duplicate sort values” box was checked, signaling that the tool will exclude rows with duplicate “Title” values. Thus, all rows that goes through the tool will have rows with unique “Title” values.**

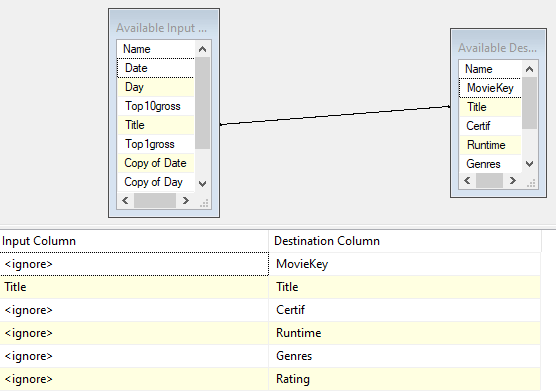
****

1. **Data are transferred into the “Convert Data” tool (renamed “Convert Date (DT\_DATE), Day (DT\_I4), Top10gross (DT\_I4), Top1gross (DT\_I4)”). “Date” column is converted to the date type. Day, Type10gross, and Top1gross columns are converted to four-byte-signed integer types.**
2. **Data from step 12 are transferred to a “Lookup” tool (renamed “Existing Title?”). “Title” values will be matched with the “Title” values in the DimMovie dimension.**

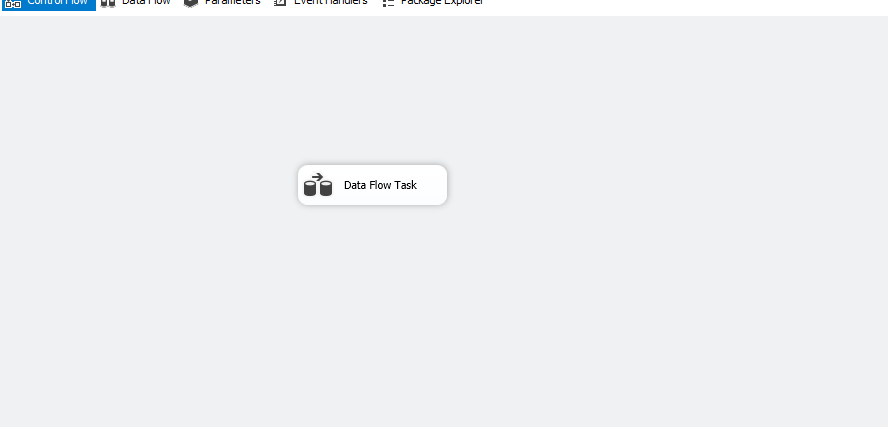
****

* 1. **Rows with “Title” values that matches will be sent to the “Union All” tool (renamed “Do Nothing”) where nothing will happen.**
  2. **Rows with non-matching “Title” values will be sent to the “OLE DB Destination” tool.**

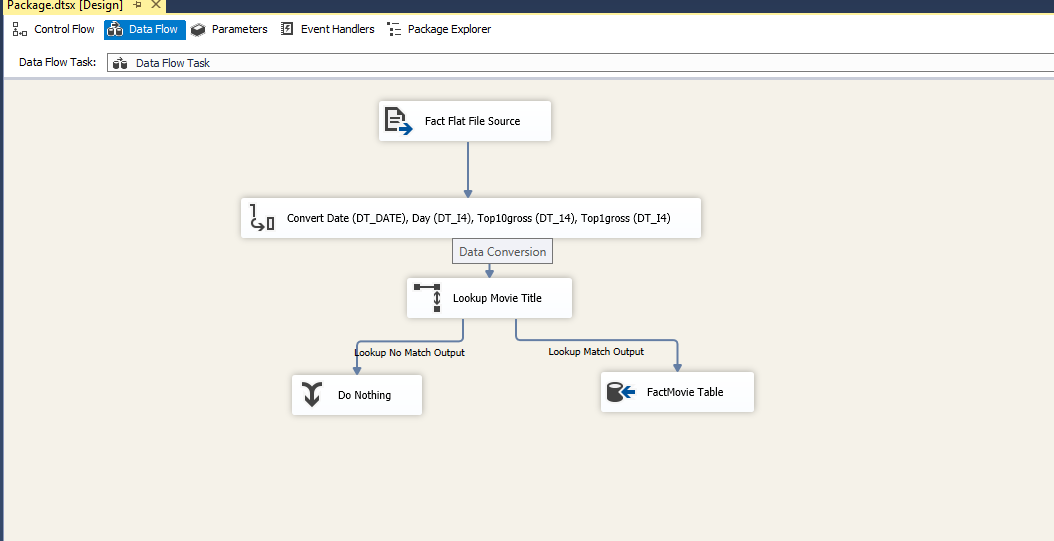
1. **Data from step 13b is transferred to an “OLE DB Destination” tool (renamed “DimMovie Table”). Here, the data will be inserted into the DimMovie dimension. Since Fact.txt file’s only common column with DimMovie is “Title”, only the “Title” value from each row will insert into the DimMovie dimension.**

****

**Flow Chart for FactMovie:**

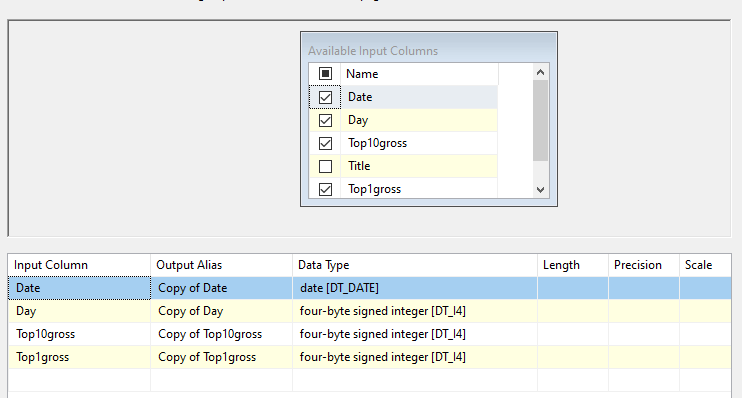
**Control Flow for FactMovie**

**Data Flow of Factmovie**

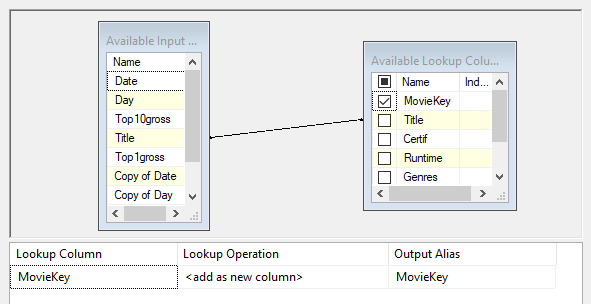
****

**Steps:**

1. **Fact.txt data will be inserted into the “Flat File Source” tool (renamed “Fact Flat File Source”). Included data columns are “Date”, “Day”, “Top10gross”, “Title”, and “Top1gross”.**
2. **Data from step 1 is transferred to a “Data Conversion” tool (renamed “Convert Date (DT\_DATE), Day (DT\_I4), Top10gross (DT\_14), Top1gross (DT\_I4)’). “Date” column is converted to date type. “Day”, “Top10gross”, and “Top1gross” columns are converted to four-byte signed integer type.**

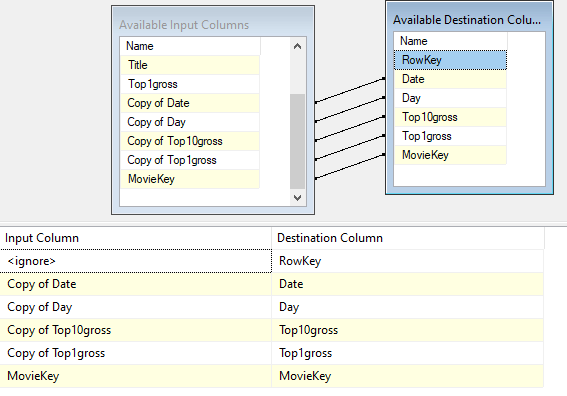
****

1. **Data from step 2 is transferred to a “Lookup” tool (renamed “Lookup Movie Title”). Here, the tool matches “Title” values from step 2 data with the “Title” values in the DimMovie Dimension.**

****

* 1. **If the “Title” value in a row in the step 2’s data matches a “Title” value in the DimMovie dimension, the tool will obtain the “MovieKey” value from the row in DimMovie with the matching “Title” value. This row from step 2’s data will then be transferred to an “OLE DB Destination” tool, along with the “MovieKey” column key value.**
  2. **Non-matching rows in the dataset will be transferred to a “Union All” tool (renamed “Do Nothing”) and nothing will happen.**

1. **The data from step 3a will be transferred to an “OLE DB Destination” tool (renamed: “FactMovie Table”).** 
   1. **Converted “Date” column values will be inserted into the “Date” column of FactMovie table.**
   2. **Converted “Day” column values will be inserted into the “Day” column of the FactMovie table.**
   3. **Converted “Top10gross” column values will be inserted into the “Top10gross” column of the FactMovie table.**
   4. **Converted “Top1gross” column values will be inserted into the “Top1gross” column of the FaceMovie table.**
   5. **“MovieKey” values obtained from step 3a will be inserted into the “MovieKey” column of the FactMovie table.**

****

**Deliverables:**

In a Word document briefly describe the data flow of the Movie dimension and the fact table. Also answer the questions asked. Name the Word document as last names of you and your partner. You need to submit the screenshots of each data flow, database diagram, time series chart, and predictions. Submit the Fact.txt and Rating.txt files. Also submit the SQL file containing all the commands you have used to create the database and views. The query file should have .sql extension so that it can be directly opened in SSMS. The due date of this project will be provided by the instructor.