Group 4

ISYS 637-603

Abstract

A concise data warehouse solution to fulfil the business intelligence needs of IMDB

IMDB Data warehouse

Design and Implementation of a Data Warehouse for Online Movie Database

**Contents**

**[1.](#_Toc481589559)****[Executive Summary](#_Toc481589559)** [2](#_Toc481589559)

[**2.** **About IMDB** 3](#_Toc481589560)

[**3.** **Why is Data Warehouse Needed?** 3](#_Toc481589561)

[**4.** **Data Warehouse Architecture** 4](#_Toc481589562)

[**4.1** **Star Schema** 5](#_Toc481589563)

[**5.** **Data Warehouse Matrix** 10](#_Toc481589564)

[**6.** **Meta Data** 10](#_Toc481589565)

[**6.1** **Dimensions** 10](#_Toc481589566)

[**6.2** **Facts** 16](#_Toc481589567)

[**7.** **ETL Plan** 17](#_Toc481589568)

[**7.1** **Data Mappings for Data Warehouse (including sources, staging and target details and transformations)** 18](#_Toc481589569)

[**7.2** **Data Extraction Rules** 25](#_Toc481589570)

[**7.3** **Data transformation and cleaning rules** 26](#_Toc481589571)

[**7.4** **Implementation plan** 26](#_Toc481589572)

[**8.** **Business Intelligence Reporting** 44](#_Toc481589573)

[**8.1** **Data Mart creation using SSAS** 45](#_Toc481589574)

[**8.2** **Report building from individual Data Mart is SSRS** 50](#_Toc481589575)

[**8.2.1** **Movie Performance** 50](#_Toc481589576)

[**i.** **Production House Performance** 61](#_Toc481589577)

[**ii.** **Awards Distribution** 63](#_Toc481589578)

[**iii.** **Theatre Performance** 66](#_Toc481589579)

[**8.3** **User Manual for creating Dynamic Reports** 73](#_Toc481589580)

[**8.3.1** **Browsing cubes via SSRS** 73](#_Toc481589581)

[**8.3.2** **Browsing cubes using pivot** 76](#_Toc481589582)

[**9.** **Glossary of terms** 82](#_Toc481589583)

[**9.** **Bibliography** 85](#_Toc481589584)

[**10.** **Author of the Report** 86](#_Toc481589585)

[**11.** **Date Report Created** 86](#_Toc481589586)

[**12.** **Contribution of each group member to the project** 86](#_Toc481589587)

# **Executive Summary**

The entertainment industry is part of the technological advancements, which has transformed the way business is being conducted. Data driven decisions give great insights into the business viability. Implementation of Data Warehouse can help businesses utilize data to make decisions about future movie productions.

Our Data Warehouse leverages the data residing in the IMDB databases and other transactional data to find out the movie performance. This data can be used for Business Intelligence requirements, for e.g. to find which is the most profitable genre, which Actor- Director pair works best for various genres etc. A data warehouse gets data from various sources and once the collation and analyses of such data has been performed, it can be used by production houses.

This report outlines the process of creating a data warehouse from scratch using SSIS, SSAS and SSRS. Sample reports that satisfy Business Intelligence questions are shown as well as the method for creating customizable reports.

The Business Intelligence reports obtained from implementing this Data Warehouse would ultimately increase revenue by leveraging previous movie data. Using a consolidated Data Warehouse instead of separately housed data sources greatly improve the efficiency of creating Business Intelligence reports and aid the business in delivering top performance.

# **About IMDB**

The Internet Movie Database (abbreviated IMDb) is an online database of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews, operated by IMDb.com, Inc., a subsidiary of Amazon.

IMDb originated with a Usenet posting by British film fan and computer programmer Col Needham entitled "Those Eyes", about actors with beautiful eyes. Others with similar interests soon responded with additions or different lists of their own. Needham subsequently started an "Actors List", while Dave Knight began a "Directors List", and Andy Krieg took over "THE LIST" from Hank Driskill, which would later be renamed the "Actress List". Both lists had been restricted to people who were alive and working, but soon retired people were added, so Needham started what was then (but did not remain) a separate "Dead Actors/Actresses List". The goal of the participants now was to make the lists as inclusive as possible.

By late 1990, the lists included almost 10,000 movies and television series correlated with actors and actresses appearing therein. On October 17, 1990, Needham developed and posted a collection of Unix shell scripts which could be used to search the four lists, and thus the database that would become the IMDb was born. At the time, it was known as the "rec.arts.movies movie database".

# **Why is Data Warehouse Needed?**

A goal of every business is to make better business decisions than their competitors. That is where business intelligence (BI) comes in. BI turns the massive amount of data from operational systems into a format that is easy to understand, current, and correct so decisions can be made on the data.

The idea is to create a permanent storage space for the data needed to support reporting, analysis, and other BI functions. While it may seem wasteful to store data in multiple places (source systems and the data warehouse), the many advantages of doing that more than justify the effort and expense.

Data warehouses reside on servers dedicated to this function running a database management system (DBMS) such as SQL Server and using Extract, Transform, and Load (ETL) software such as SQL Server Integration Services (SSIS) to pull data from the source systems and into the data warehouse.

In respect to IMDB, it is usually considered that there is no universal way to claim the goodness of movies. Many people rely on critics to gauge the quality of a film, while others use their instincts. However, it takes the time to obtain a reasonable amount of critics review after a movie is released. Moreover, human instinct sometimes is unreliable. Thus, a data warehouse solution can be used to derive facts over word of mouth. The data collection reflects lists of movies and associated information. The amount of data stored in the database can be used to answer multiple business intelligence question such as performance of a movie, theatres and the production houses. The data can also be used as a basis to decide on a new project.

# **Data Warehouse Architecture**

END USER

Trend Reports

Directors / Casting agents

Production House

BI LAYER

Power Pivot

SSRS

OLAP CUBES

Date Warehouse

DATA WAREHOUSE

DATA MARTS

Production

Awards

Theatre

Movie

STAGING AREA

Staging Schema

SOURCES

Flat Files (csv, xls)

IMDB source DB

Kaggle DB

## **Star Schema**

Multiple data marts are created to answer the different business questions. This section gives an overview of the data-marts created and the fact, dimension tables used to create them along with their relationship. All the data marts created follow start schema.

Also, note: All attributes of the dimensions are of SCD type 1 (contents are overwritten on change) except for the ones denoted as SCD 2. These attributes need the data to be preserved even after they change. This is handled by use of start\_date and end\_date attributes in the appropriate dimensions. If end\_date is null then no change has happened whereas, if end\_date is populated then it denotes that specific entry as old (not current) and end date also represents the effective date of the change just the way start\_date denotes the day when entry happened.

* Awards

The data mart for analyzing the winners of different awards is shown in the below figure. It contains:

* Awards Fact table
* Actor, Movie and Director Dimension tables

The grain of the fact table is an Award. A transactional entry happens every time an award is announced / awarded.

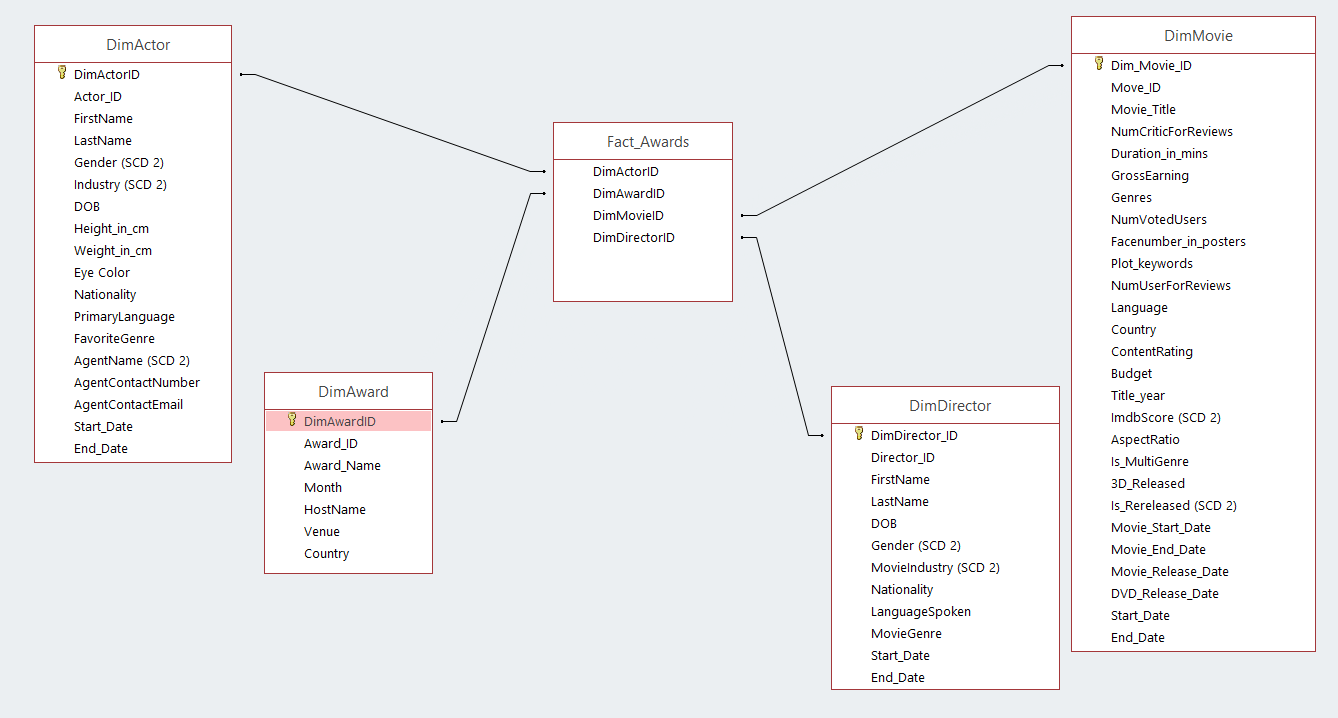


Figure: Awards Data mart

* Movie Performance

The data mart for analyzing a movie’s performance is shown in the below figure. It contains:

* Movie Performance Fact table
* Date, Movie, Actor and Director Dimension tables

The grain of the fact table is a Movie. A transactional entry happens every time a movie is release and at the end of each week.

Here, Role Playing is used to represent actor in different views. Actor dimension plays the roles of LeadActor and SupportingActor in the fact table.

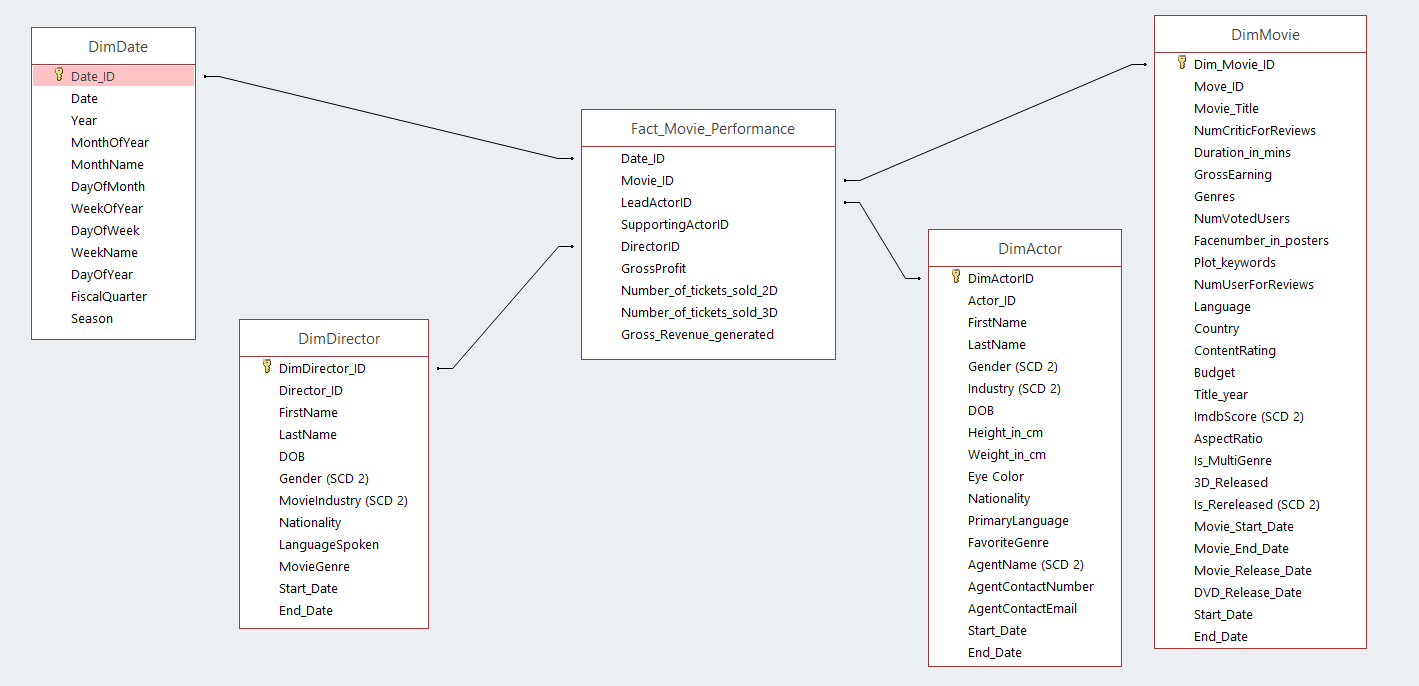


Figure: Movie Performance data mart

* Movie Production

The data mart for analyzing the performance of a movie production-house is shown in the below figure. It contains:

* Movie Production Fact table
* Date, Movie, Actor and Production House Dimension tables

The grain of the fact table is a Movie. A transactional entry happens every time a movie is release and weekly from that point on.

Here, Role Playing is used to represent date and actor in different views. Date dimension (DimDate) plays the roles of Movie start date, end date, release date and DVD release date in the fact table. And Actor dimension plays the roles of LeadActor and SupportingActor in the fact table.

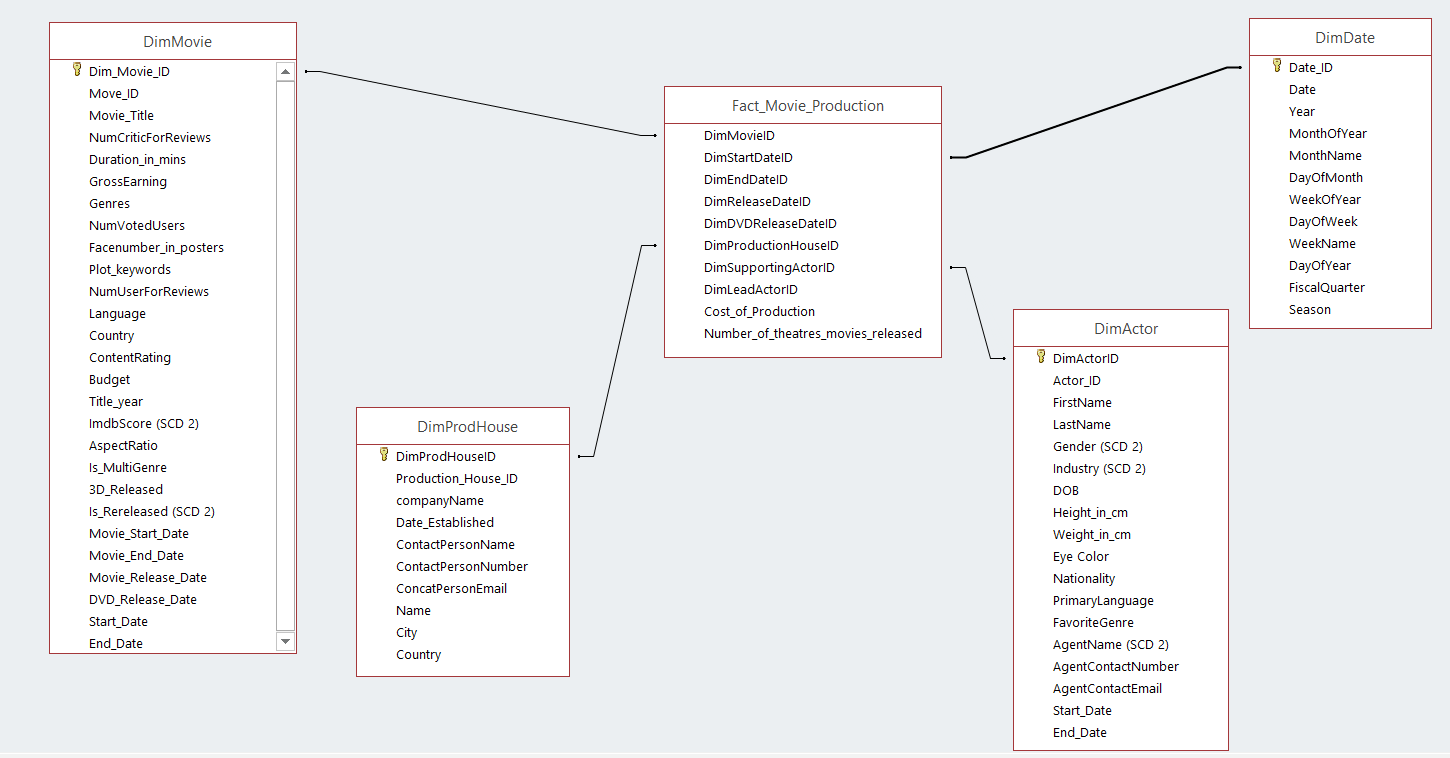


Figure: Movie Production data mart

* Theatre Performance

The data mart for analyzing the performance of a movie theatre is shown in the below figure. It contains:

* Movie Theatre Fact table
* Date, Movie and Theatre Dimension tables

The grain of the fact table is a Movie release in a theatre. A transactional entry happens every time a movie is release and weekly from that point on.

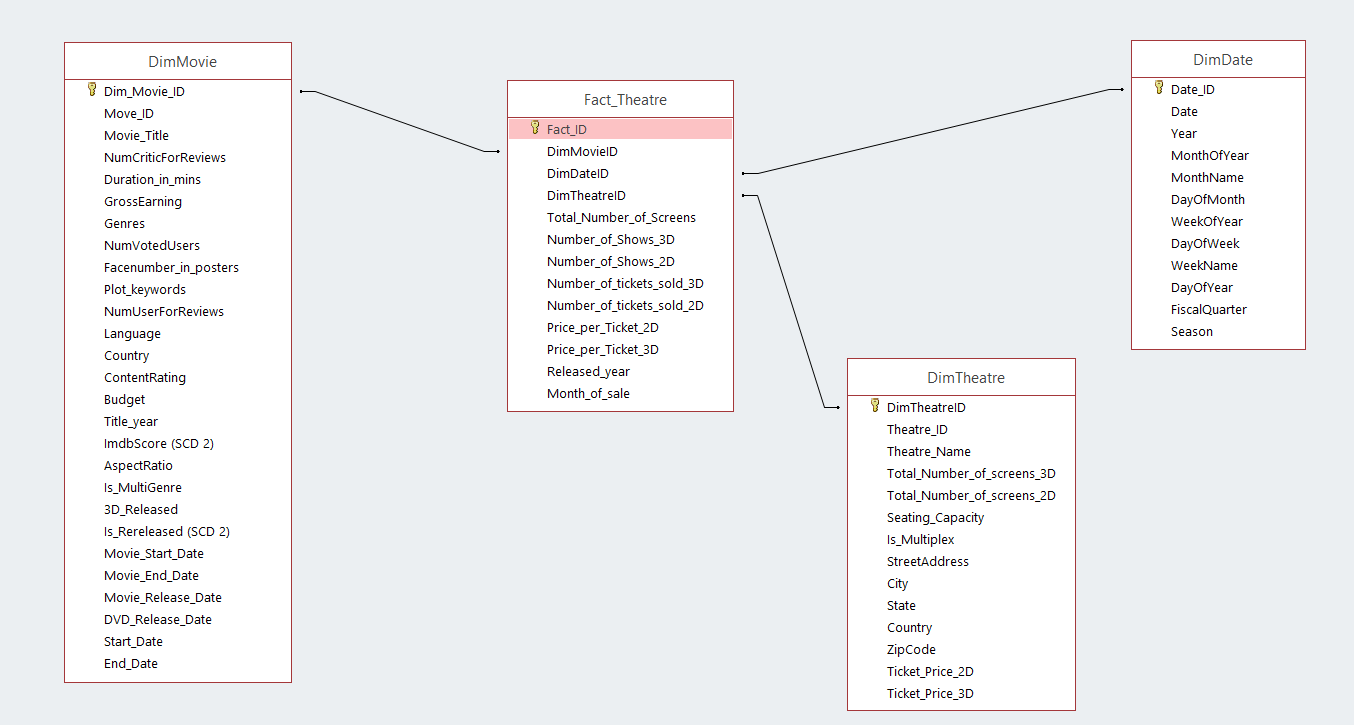


Figure: Theatre performance data mart

# **Data Warehouse Matrix**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Business Process** | ***Date*** | ***Actor*** | ***Theatre*** | ***Movie*** | ***Production House*** | ***Director*** | ***Awards*** |
| **Production House Performance** | **X** | **X** |  | **X** | **X** |  |  |
| **Movie Performance** | **X** | **X** |  | **X** |  | **X** |  |
| **Theatre Performance** | **X** |  | **X** | **X** |  |  |  |
| **Awards Distribution** |  | **X** |  | **X** |  | **X** | **X** |

# **Meta Data**

## **Dimensions**

|  |  |
| --- | --- |
| **Dimension Meta Data** | **Description** |
| Name of Dimension | DimAward |
| Business Definition | This dimension holds the data that contains the various attributes for the awards that could be won by any movie, director, actor etc. |
| Attributes | |  |  | | --- | --- | | Attributes | Format | | DimAwardId | AutoNumber | | Award\_Name | ShortText | | Country | ShortText | | Venue | ShortText | | Host\_name | ShortText | | Award\_id | Number | | Month | ShortText | |
| SCD | All dimensions are of SCD type 1. |
| Hierarchy | No Hierarchy |
| Load Frequency | Weekly |
| Source | CSV |
| Conformed | Yes. The event attributes remains same across all fact tables |
| Role Playing | No Roles |

|  |  |
| --- | --- |
| **Dimension Meta Data** | **Description** |
| Name of Dimension | DimProductionHouse |
| Business Definition | This dimension holds the data that contains the various attributes for the Production house which produces any movie. |
| Attributes | |  |  | | --- | --- | | Attributes | Format | | DimProdHouseId | AutoNumber | | CompanyName | ShortText | | City | ShortText | | Country | ShortText | | Name | ShortText | | ProdHouse\_id | Number | | Date\_Established | Date/Time | | ContactPersonEmail | ShortText | | ContactPersonNumber | ShortText | | ProdHouse\_id | ShortText | | Date\_Established | ShortText | |
| SCD | All dimensions are of SCD type 1. |
| Hierarchy | No Hierarchy |
| Load Frequency | Weekly |
| Source | Relational Database |
| Conformed | Yes. The event attributes remains same across all fact tables |
| Role Playing | No Roles |

|  |  |
| --- | --- |
| **Dimension Meta data** | **Description** |
| Name of Dimension | Dim\_Actor |
| Business Definition | This dimension holds the data that contains the various attributes for the actor (Lead actor and Supporting actors). |
| Attributes Format | |  |  | | --- | --- | | Attributes | Format | | DimActorID | AutoNumber | | Actor\_ID | ShortText | | FirstName | ShortText | | LastName | ShortText | | Gender (SCD 2) | ShortText | | Industry (SCD 2) | ShortText | | DOB | Date/Time | | Height\_in\_cm | Number | | Weight\_in\_cm | Number | | Eye Color | ShortText | | Nationality | ShortText | | PrimaryLanguage | ShortText | | FavoriteGenre | ShortText | | AgentName (SCD 2) | ShortText | | AgentContactNumber | ShortText | | AgentContactEmail | ShortText | | Start\_Date | Date/Time | | End\_Date | Date/Time | |
| SCD | All dimensions are of SCD type 1 except the below:  Gender, Industry and AgentName are SCD type2. These are handled by use of start\_date and end\_date attributes. |
| Hierarchy | No Hierarchy |
| Load Frequency | Weekly |
| Source | Relational Database and CSV |
| Conformed | Yes. The event attributes remains same across all fact tables |
| Role Playing | DimActor plays the role of lead actor and supporting actor as a part of Fact\_Movie\_Performance and Fact\_Movie\_Production |

|  |  |
| --- | --- |
| **Dimension meta data** | **Description** |
| Name of Dimension | DimDirector |
| Business Definition | This dimension holds the data that contains the various attributes for the movie director. |
| Attributes Format | |  |  | | --- | --- | | Attributes | Format | | DimDirectorID | int | | Director\_ID | float | | FirstName | varchar | | LastName | varchar | | DOB | datetime | | Gender | varchar | | MovieIndustry | float | | Nationality | varchar | | Language Spoken | varchar | | MovieGenre | varchar | |
| SCD | All dimensions are of SCD type 1. |
| Hierarchy | No Hierarchy |
| Load Frequency | Weekly |
| Source | Relational Database and CSV |
| Conformed | Yes. The event attributes remains same across all fact tables |
| Role Playing | No Roles |

|  |  |
| --- | --- |
| **Dimension Meta data** | **Description** |
| Name of Dimension | Dim\_Theatre |
| Business Definition | This dimension holds the data that contains the various attributes about a theatre. |
| Attributes Format | |  |  | | --- | --- | | Attributes | Format | | DimTheatreID | AutoNumber | | Theatre\_ID | ShortText | | Theatre\_Name | ShortText | | Total\_Number\_of\_screens\_3D | Number | | Total\_Number\_of\_screens\_2D | Number | | Seating\_Capacity | Number | | Is\_Multiplex | Yes/No | | StreetAddress | ShortText | | City | ShortText | | State | ShortText | | Country | ShortText | | ZipCode | Number | | Ticket\_Price\_2D | Number | | Ticket\_Price\_3D | Number | |
| SCD | All dimensions are of SCD type 1. |
| Hierarchy | StreetAddress < City < State < Country |
| Load Frequency | Weekly |
| Source | Relational Database |
| Conformed | Yes. The event attributes remains same across all fact tables |
| Role Playing | No Roles |

|  |  |
| --- | --- |
| **Dimension Meta data** | **Description** |
| Name of Dimension | Dim\_Movie |
| Business Definition | This dimension holds the data that contains the various attributes about a movie. |
| Attributes Format | |  |  | | --- | --- | | Target table Attributes | Data Types | | DimMovieID | int | | Movie\_ID | int | | Num\_critic\_for\_reviews | float | | sDuration\_in\_minutes | float | | GrossEarning | float | | Genres | varchar | | Movie\_title | varchar | | Num\_voted\_users | float | | Facenumber\_in\_poster | float | | Plot\_keywords | varchar | | Num\_user\_for\_reviews | float | | Language | varchar | | Country | varchar | | Content\_rating | varchar | | Budget | float | | Title\_year | float | | Imdb\_score | float | | Aspect\_ratio | float | | Is\_Multigenre | varchar | | 3D\_Released | varchar | | Is\_Rereleased | varchar | | Movie\_Start\_Date | datetime | | Movie\_End\_Date | datetime | | Movie\_Release\_Date | datetime | | DVD\_Release\_Date | datetime | |
| SCD | All dimensions are of SCD type 1. |
| Hierarchy | No Hierarchy |
| Load Frequency | Weekly |
| Source | Relational Database |
| Conformed | Yes. The event attributes remains same across all fact tables |
| Role Playing | No Roles |

|  |  |
| --- | --- |
| **Dimension Meta data** | **Description** |
| Name of Dimension | DimDate |
| Business Definition | This dimension holds the data about Date and time. |
| Attributes Format | |  |  | | --- | --- | | Attributes | Format | | DimdateID | int | | Date | datetime | | Year | int | | MonthofYear | int | | MonthName | varchar | | DayofMonth | float | | WeekofYear | float | | DayofWeek | float | | WeekName | varchar | | DayofYear | float | | Fiscal Quarter | varchar | | Season | varchar | | DimdateID | int | | Date | datetime | |
| SCD | All dimensions are of SCD type 1. |
| Hierarchy | Date < Week < Month < Year |
| Load Frequency | Weekly |
| Source | Relational Database |
| Conformed | Yes. The event attributes remains same across all fact tables |
| Role Playing | Yes. There are role playing dimensions of Movie\_start\_date, Movie\_end\_date, Movie\_release\_date, DVD\_release\_date |

## **Facts**

|  |  |
| --- | --- |
| **Fact Table Meta Data** | **Description** |
| Name of Fact table | Fact\_Movie\_Production |
| Business definition | This transactional fact table is created to capture the performance of the production houses. |
| Dimensions | DimDate, DimMovie, DimProdHouse, DimActor |
| Grain | Record is created for each movie capturing the production house Of the movie, cost of production and the number of thetares movie was released in. |
| Load Frequency | Weekly |
| Source | Transactional Database |
| Measures / Facts | Cost Of Production  Number\_of\_theatres\_movies\_released |

|  |  |
| --- | --- |
| **Fact Table Meta Data** | **Description** |
| Name of Fact table | Fact\_Theatre |
| Business definition | This transactional fact table is created to capture the performance of the theaters worldwide. |
| Dimensions | Dim\_Date, DimMovie, DimTheatre |
| Grain | Record is created for each movie released in each theatre capturing the all theatre attributes for each movie such as ticket price, no of shows etc. |
| Load Frequency | Weekly |
| Source | Transactional Database |
| Measures / Facts | Total\_Number\_of\_Screens  Number\_of\_Shows\_3D  Number\_of\_Shows\_2D  Number\_of\_tickets\_sold\_3D  Number\_of\_tickets\_sold\_2D  Price\_per\_Ticket\_2D  Price\_per\_Ticket\_3D |

|  |  |
| --- | --- |
| **Fact Table Meta Data** | **Description** |
| Name of Fact table | Fact\_Award |
| Business definition | This transactional fact table is created to capture the winners of the various awards. |
| Dimensions | DimAward, DimMovie, DimDirector, DimActor |
| Grain | Record is created for each award capturing the winner of that year/month’s winner. |
| Load Frequency | Weekly |
| Source | Transactional Database |
| Measures / Facts | Factless Fact table |

|  |  |
| --- | --- |
| **Fact Table Meta Data** | **Description** |
| Name of Fact table | Fact\_Movie\_Performance |
| Business definition | This transactional fact table is created to capture the performance of the movies worldwide. |
| Dimensions | Dim\_Date, DimDirector, DimActor, DimMovie |
| Grain | Record is created for each movie released capturing the movie’s performance such as number of tickets sold, profit price, etc. |
| Load Frequency | Weekly |
| Source | Transactional Database |
| Measures / Facts | Gross Profit,  Number\_of\_tickets\_sold\_2D,  Number\_of\_tickets\_sold\_3D,  Gross\_Revenue\_Generated |

# **ETL Plan**

ETL refers to ‘Extract, Transform and load’. The various steps in the ETL process are as follows:

* Extracting data from databases: Data presented in the .csv files is extracted and loaded into the staging area by using Data flow tasks.
* Transforming the extracted data: For storing the data in relevant formats to enable query execution Data present in the staging area is cleaned and transformed to cater to the Business Intelligence questions.
* Loading the data into a final target database: The final tables that have been created are loaded into the Facts and Dimension tables.

Following is the layout of the **ETL plan** for Data Warehouse implementation:

* + Preparation of Data mappings of the Data from sources in Excel to staging area and from the staging area to the data warehouse.
  + Determine the Data extraction rules
  + Determine the Data transformation and cleansing rules
  + Implementation plan : Plan and execute procedures for extraction and loading

## **Data Mappings for Data Warehouse (including sources, staging and target details and transformations)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Source File | Transformation Rule |
| DimActor | DimActorID | int | Actor\_table.xlsx | Surrogate key of the dimension. Inserted as incremental key while loading data. |
| DimActor | Actor\_Id | float | Actor\_table.xlsx | Primary Key |
| DimActor | First\_name | varchar | Actor\_table.xlsx |  |
| DimActor | Last\_name | varchar | Actor\_table.xlsx |  |
| DimActor | Gender | varchar | Actor\_table.xlsx |  |
| DimActor | Industry | varchar | Actor\_table.xlsx |  |
| DimActor | DOB | datetime | Actor\_table.xlsx |  |
| DimActor | Height\_in\_cm | float | Actor\_table.xlsx |  |
| DimActor | Weight\_in\_lb | float | Actor\_table.xlsx |  |
| DimActor | EyeColor | varchar | Actor\_table.xlsx |  |
| DimActor | Nationality | varchar | Actor\_table.xlsx |  |
| DimActor | PrimaryLanguage | varchar | Actor\_table.xlsx |  |
| DimActor | AgentName | varchar | Actor\_table.xlsx |  |
| DimActor | AgentContactNumber | varchar | Actor\_table.xlsx |  |
| DimActor | AgentContactEmail | varchar | Actor\_table.xlsx |  |
| DimActor | FavoriteGenre | varchar | Actor\_table.xlsx |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Source File | Transformation Rule |
| DimAward | DimAwardID | int | award.csv | Surrogate key of the dimension. Inserted as incremental key while loading data. |
| DimAward | Award\_Id | float | award.csv | Primary key |
| DimAward | Award\_name | varchar | award.csv |  |
| DimAward | Country | varchar | award.csv |  |
| DimAward | Venue | varchar | award.csv |  |
| DimAward | Host\_name | varchar | award.csv |  |
| DimAward | Month | varchar | award.csv |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Source File | Transformation Rule |
| DimDate | DimdateID | int | Date.xlsx | Primary key |
| DimDate | Date | datetime | Date.xlsx |  |
| DimDate | Year | int | Date.xlsx |  |
| DimDate | MonthofYear | int | Date.xlsx |  |
| DimDate | MonthName | varchar | Date.xlsx |  |
| DimDate | DayofMonth | float | Date.xlsx |  |
| DimDate | WeekofYear | float | Date.xlsx |  |
| DimDate | DayofWeek | float | Date.xlsx |  |
| DimDate | WeekName | varchar | Date.xlsx |  |
| DimDate | DayofYear | float | Date.xlsx |  |
| DimDate | Fiscal Quarter | varchar | Date.xlsx |  |
| DimDate | Season | varchar | Date.xlsx |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Source File | Transformation Rule |
| DimDirector | DimDirectorID | int | Director.xlsx | Surrogate key of the dimension. Inserted as incremental key while loading data. |
| DimDirector | Director\_ID | float | Director.xlsx | Primary key |
| DimDirector | FirstName | varchar | Director.xlsx |  |
| DimDirector | LastName | varchar | Director.xlsx |  |
| DimDirector | DOB | datetime | Director.xlsx |  |
| DimDirector | Gender | varchar | Director.xlsx |  |
| DimDirector | MovieIndustry | float | Director.xlsx |  |
| DimDirector | Nationality | varchar | Director.xlsx |  |
| DimDirector | Language Spoken | varchar | Director.xlsx |  |
| DimDirector | MovieGenre | varchar | Director.xlsx |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Source File | Transformation Rule |
| DimMovie | DimMovieID | int | Movie.xlsx | Surrogate key of the dimension. Inserted as incremental key while loading data. |
| DimMovie | Movie\_ID | int | Movie.xlsx | Primary key |
| DimMovie | Num\_critic\_for\_reviews | float | Movie.xlsx |  |
| DimMovie | Duration\_in\_minutes | float | Movie.xlsx |  |
| DimMovie | GrossEarning | float | Movie.xlsx |  |
| DimMovie | Genres | varchar | Movie.xlsx |  |
| DimMovie | Movie\_title | varchar | Movie.xlsx |  |
| DimMovie | Num\_voted\_users | float | Movie.xlsx |  |
| DimMovie | Facenumber\_in\_poster | float | Movie.xlsx |  |
| DimMovie | Plot\_keywords | varchar | Movie.xlsx |  |
| DimMovie | Num\_user\_for\_reviews | float | Movie.xlsx |  |
| DimMovie | Language | varchar | Movie.xlsx |  |
| DimMovie | Country | varchar | Movie.xlsx |  |
| DimMovie | Content\_rating | varchar | Movie.xlsx |  |
| DimMovie | Budget | float | Movie.xlsx |  |
| DimMovie | Title\_year | float | Movie.xlsx |  |
| DimMovie | Imdb\_score | float | Movie.xlsx |  |
| DimMovie | Aspect\_ratio | float | Movie.xlsx |  |
| DimMovie | Is\_Multigenre | varchar | Movie.xlsx |  |
| DimMovie | 3D\_Released | varchar | Movie.xlsx |  |
| DimMovie | Is\_Rereleased | varchar | Movie.xlsx |  |
| DimMovie | Movie\_Start\_Date | datetime | Movie.xlsx |  |
| DimMovie | Movie\_End\_Date | datetime | Movie.xlsx |  |
| DimMovie | Movie\_Release\_Date | datetime | Movie.xlsx |  |
| DimMovie | DVD\_Release\_Date | datetime | Movie.xlsx |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Source File | Transformation Rule |
| DimProdHouse | DimProdHouseID | int | production.xlsx | Surrogate key of the dimension. Inserted as incremental key while loading data. |
| DimProdHouse | ProdHouse\_ID | float | production.xlsx | Primary key |
| DimProdHouse | CompanyName | varchar | production.xlsx |  |
| DimProdHouse | City | varchar | production.xlsx |  |
| DimProdHouse | Country | varchar | production.xlsx |  |
| DimProdHouse | Name | varchar | production.xlsx |  |
| DimProdHouse | ContactPersonEmail | varchar | production.xlsx |  |
| DimProdHouse | DateEstablished | datetime | production.xlsx |  |
| DimProdHouse | ContactPersonNumber | varchar | production.xlsx |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Source File | Transformation Rule |
| DimTheatre | DimTheatreID | int | Theatre.xlsx | Surrogate key of the dimension. Inserted as incremental key while loading data. |
| DimTheatre | theatre\_ID | Int | Theatre.xlsx | Primary key |
| DimTheatre | Theatre\_Name | varchar | Theatre.xlsx |  |
| DimTheatre | Total\_num\_of\_Screens\_2D | float | Theatre.xlsx |  |
| DimTheatre | Total\_num\_of\_Screens\_3D | float | Theatre.xlsx |  |
| DimTheatre | Seating\_Capacity | float | Theatre.xlsx |  |
| DimTheatre | IsMultiplex | bit | Theatre.xlsx |  |
| DimTheatre | Street\_Address | varchar | Theatre.xlsx |  |
| DimTheatre | City | varchar | Theatre.xlsx |  |
| DimTheatre | State | varchar | Theatre.xlsx |  |
| DimTheatre | Country | varchar | Theatre.xlsx |  |
| DimTheatre | ZipCode | float | Theatre.xlsx |  |
| DimTheatre | Ticket\_price\_2D | float | Theatre.xlsx |  |
| DimTheatre | Ticket\_price\_3D | float | Theatre.xlsx |  |

**Fact tables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Staging table attributes | Transformation Rule |
| fact\_Awards | DimActorID | int | Actor\_Id | Foreign key of dimension table corresponding to Actor ID |
| fact\_Awards | DimDirectorID | Int | Director\_ID | Foreign key of dimension table corresponding to Director ID |
| fact\_Awards | DimMovieId | int | Movie\_Id | Foreign key of dimension table corresponding to movie ID |
| fact\_Awards | DimAwardID | int | Award\_Id |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Staging table attributes | Transformation Rule |
| fact\_Movie\_Performance | DimMovieId | int | Movie\_Id | Foreign key of dimension table corresponding to movie ID |
| fact\_Movie\_Performance | DimActorID | int | Actor\_ID | Foreign key of dimension table corresponding to actor ID |
| fact\_Movie\_Performance | DimActorID | int | Actor\_ID | Foreign key of dimension table corresponding to actor ID |
| fact\_Movie\_Performance | DimDirectorID | int | Director\_Id | Foreign key of dimension table corresponding to director ID |
| fact\_Movie\_Performance | DimDateID | int | Date\_Id | Foreign key of dimension table corresponding to date ID |
| fact\_Movie\_Performance | Number\_of\_tickets\_sold\_2D | float |  |  |
| fact\_Movie\_Performance | Number\_of\_tickets\_sold\_3D | float |  |  |
| fact\_Movie\_Performance | Gross\_Revenue\_Generated | float |  |  |
| fact\_Movie\_Performance | GrossProfit | float |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Staging table attributes | Transformation Rule |
| fact\_movie\_production | DimMovieID | int | Movie\_Id | Foreign key of dimension table corresponding to Movie ID |
| fact\_movie\_production | DimStartDateID | Int | Date\_ID | Foreign key of dimension table corresponding to Date ID |
| fact\_movie\_production | DimEndDateId | int | Date\_ID | Foreign key of dimension table corresponding to Date ID |
| fact\_movie\_production | DimReleaseDateId | int | Date\_ID | Foreign key of dimension table corresponding to Date ID |
| fact\_movie\_production | DimDVDReleaseDateId | int | Date\_ID | Foreign key of dimension table corresponding to Date ID |
| fact\_movie\_production | DimProdHouseID | int | ProdHouse\_ID | Foreign key of dimension table corresponding to ProdHouse\_ID |
| fact\_movie\_production | DimLeadActorID | int | Actor\_Id | Foreign key of dimension table corresponding to Actor\_ID |
| fact\_movie\_production | DimSupportingActorID | int | Actor\_id | Foreign key of dimension table corresponding to Actor\_ID |
| fact\_movie\_production | Cost\_of\_production | float |  |  |
| fact\_movie\_production | Number\_of\_Theatres\_Movie\_Released\_In | int |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target table | Target table Attributes | Data Types | Staging table attributes | Transformation Rule |
| fact\_Theatre | DimMovieID | int | Movie\_Id | Foreign key of dimension table corresponding to Movie ID |
| fact\_Theatre | DimTheatreID | Int | Date\_ID | Foreign key of dimension table corresponding to Theatre ID |
| fact\_Theatre | DimDateId | int | Date\_ID | Foreign key of dimension table corresponding to Date ID |
| fact\_Theatre | Number\_of\_tickets\_sold\_2D | int |  |  |
| fact\_Theatre | Number\_of\_tickets\_sold\_3D | int |  |  |
| fact\_Theatre | Number\_of\_shows\_2D | int |  |  |
| fact\_Theatre | Number\_of\_shows\_3D | int |  |  |
| fact\_Theatre | Total\_number\_of\_Screens | int |  |  |
| fact\_Theatre | Released\_year | float |  |  |
| fact\_Theatre | Month\_of\_sale | int |  |  |
| fact\_Theatre | Price\_per\_ticket\_2D |  |  |  |
| fact\_Theatre | Price\_per\_ticket\_3D |  |  |  |

## **Data Extraction Rules**

The process of retrieving data out from data sources for processing or storage is known as Data Extraction. Data extraction is the initial step of data transforming, loading and then designing the data warehouse. Data present in data files is often poorly structured. The import to the staging system of such data is usually followed by data transformation before moving ahead.

To achieve data extraction, we employed the following steps:

* Source data that is present in the Comma Separated Value (.csv) and other formats is extracted and imported into Microsoft SQL server as tables.
* This data is used for Data transformation and further loading. Once data has been put extracted into the staging area, it is cleaned and transformed to create Dimension and fact tables. Once these Dimension and fact tables have been verified with respect to the Business Intelligence needs, it is loaded into the Data Warehouse area.

## **Data transformation and cleaning rules**

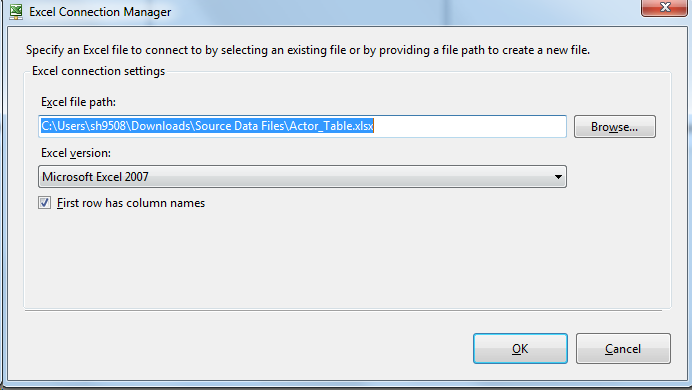
The next step is to clean the data that has been extracted from the source files. In order to maintain the consistency of the data throughout, it is essential to clean and transform the data. It ensures that all dirty data is removed and does not lead to any false results. The clean data is then loaded into individual data marts for further analysis. The following actions were performed to clean the data:

* **Removal of Dirty Data:** Attributes in data sources which were irrelevant to the business questions asked, were ignored while extracting data. Records having strange values were deleted. For example, records having gibberish special character values “%%^$#%” were deleted.
* **Removal of Null Values**: All the null values present in various tables are deleted.
* **Surrogate Key Creation**: All dimension and fact tables have surrogate keys created before the data is loaded in the data warehouse.
* **Derived Attributes:** The derived attributes in the Dimensional Table and Fact Tables are as follows:
* In the Movie\_Performance fact table derived column is added with name Gross Profit.

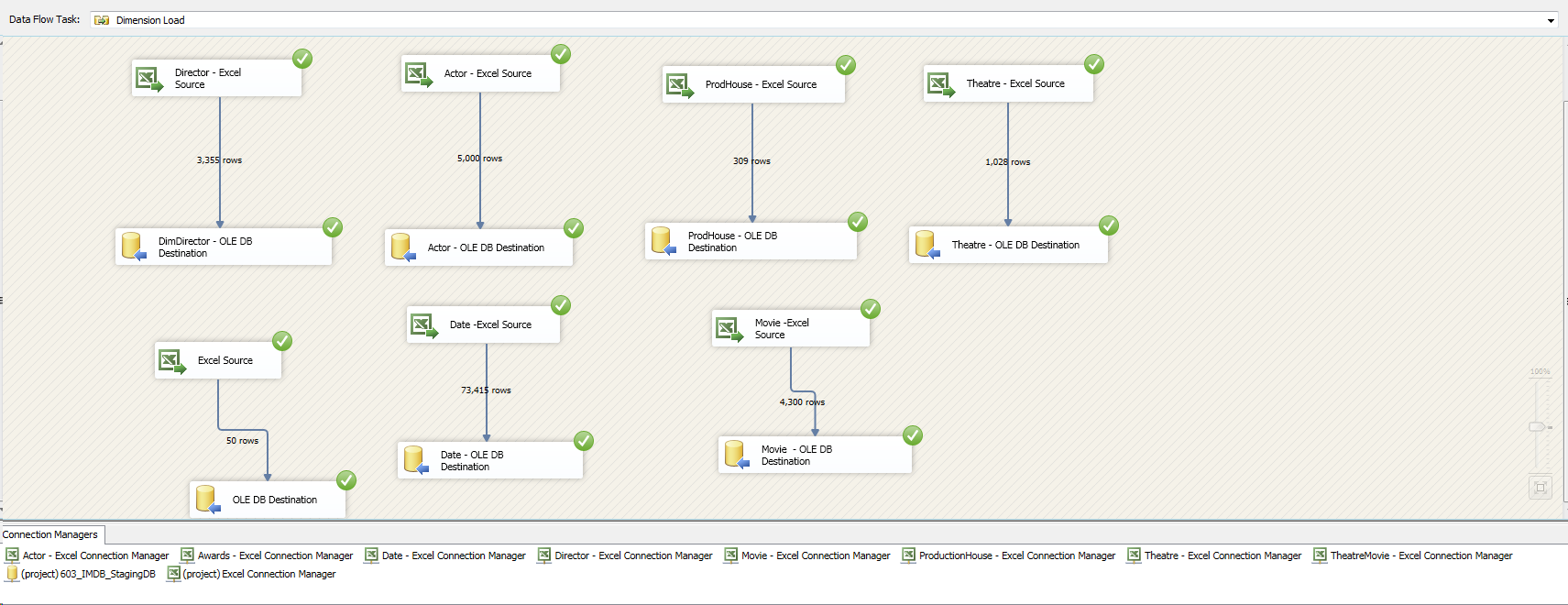
## **Implementation plan**

**Dimension Creation:**

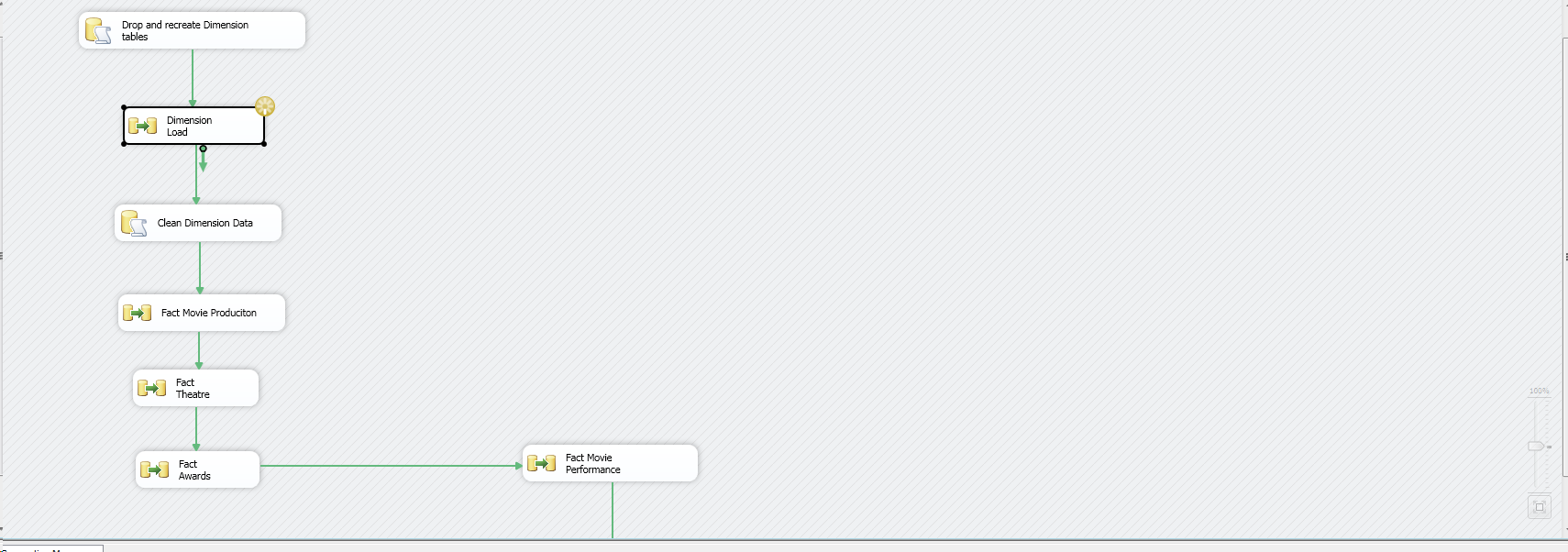
Step1: Establishing the Connection Manager



Step 2: Establish Connection Manager for all Dimensions:



Step 3: Executing Dimension Load Control Flow:





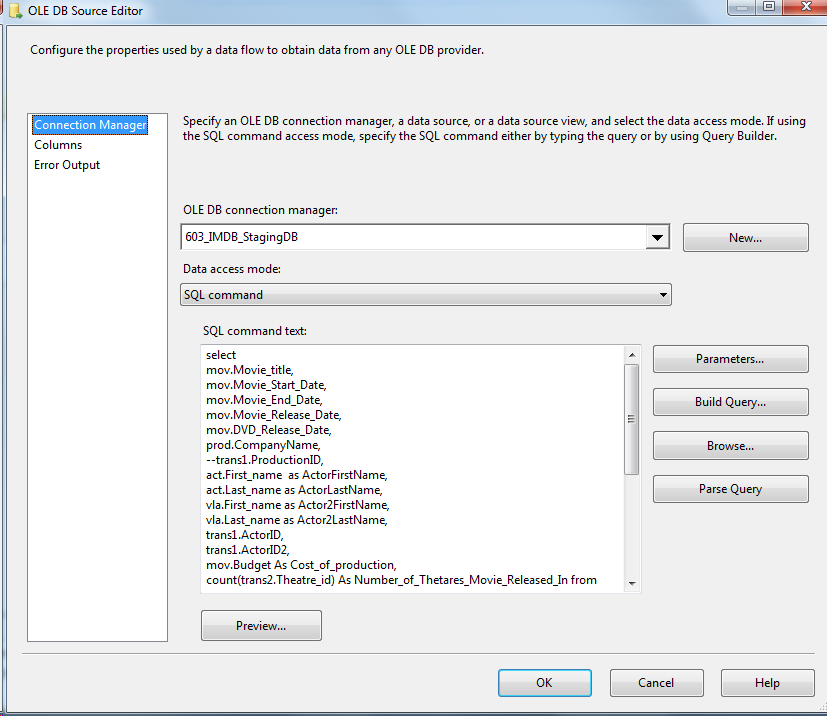
**Facts Creation:**

1. Fact Movie Production:

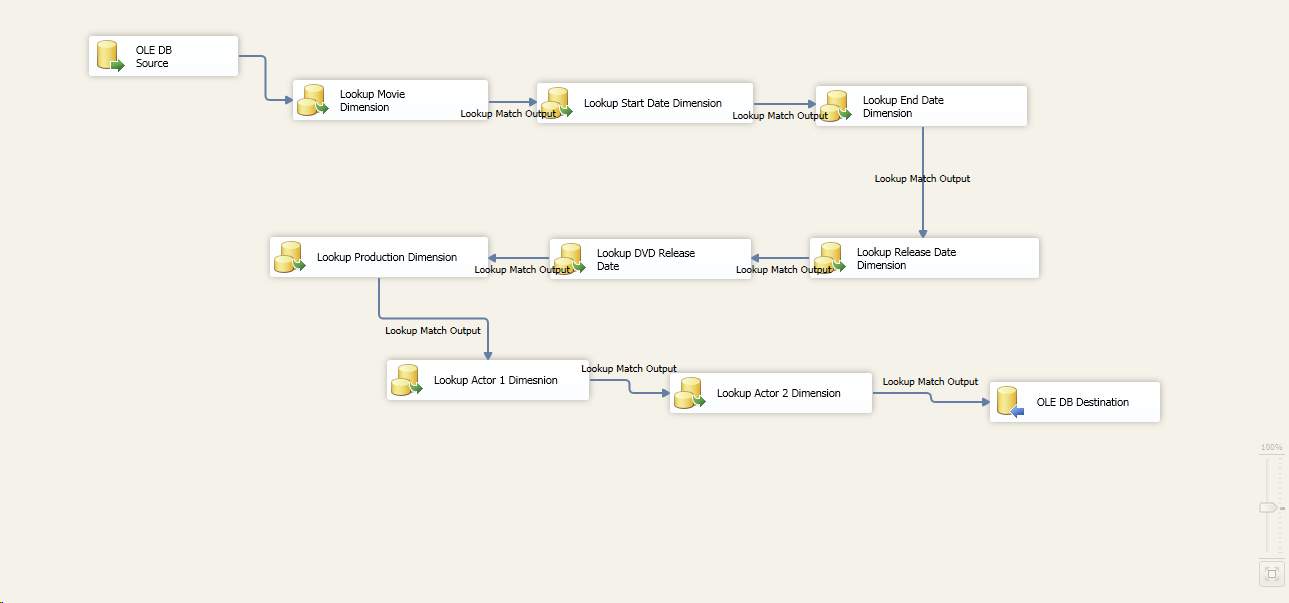
Step 1: Establish Fact Movie Production Control Flow



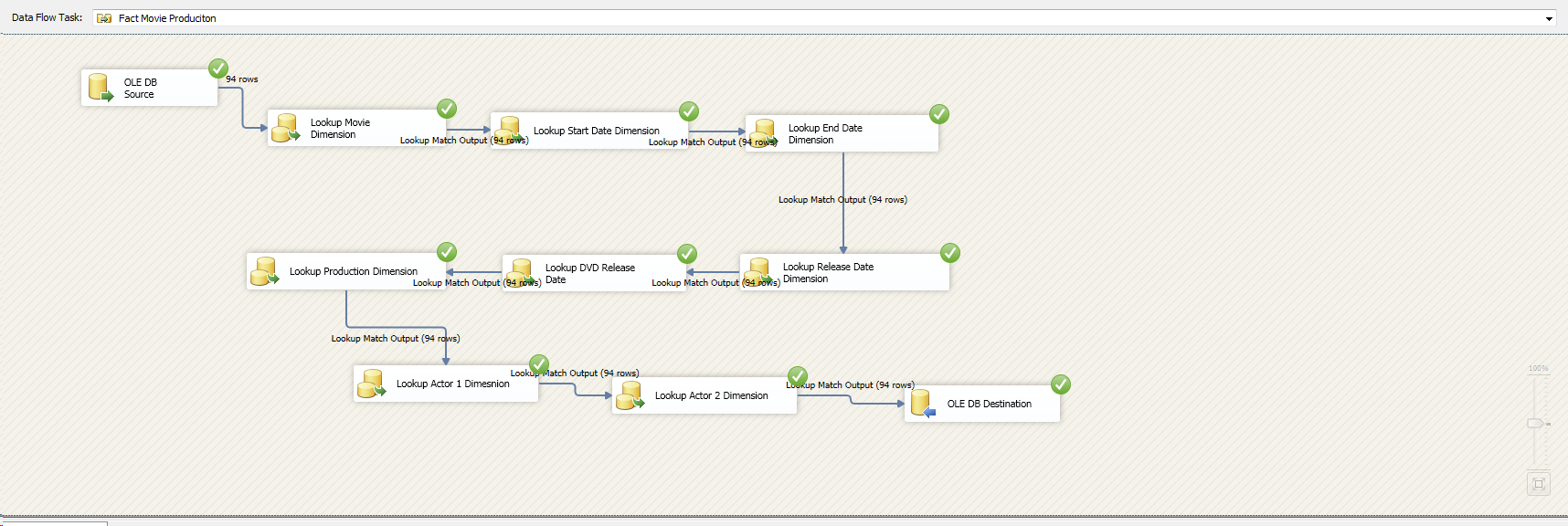
Step 2: Establishing Pre fact query and providing it as OLE DB data source:



Step 3: Making Lookups on all the Dimension Tables which are required:



Step 4: Successful execution of all Lookups:



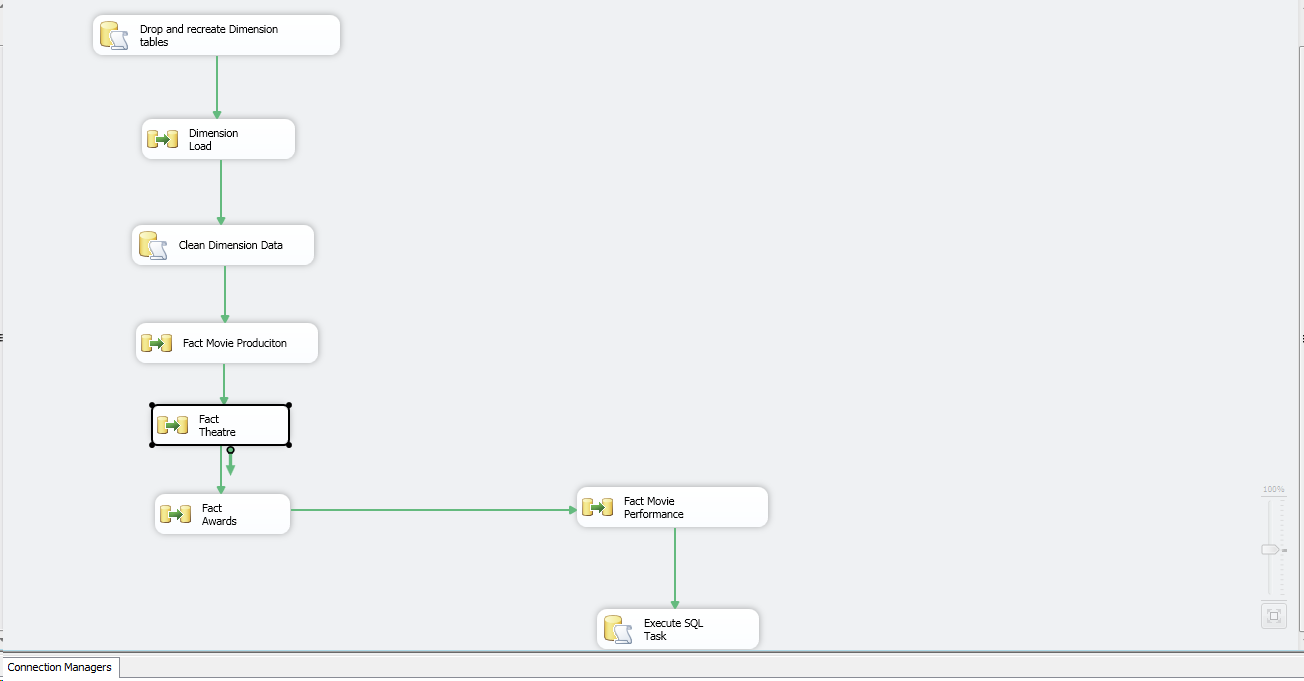
Step 5: Executing fact- Movie Production Control Flow:



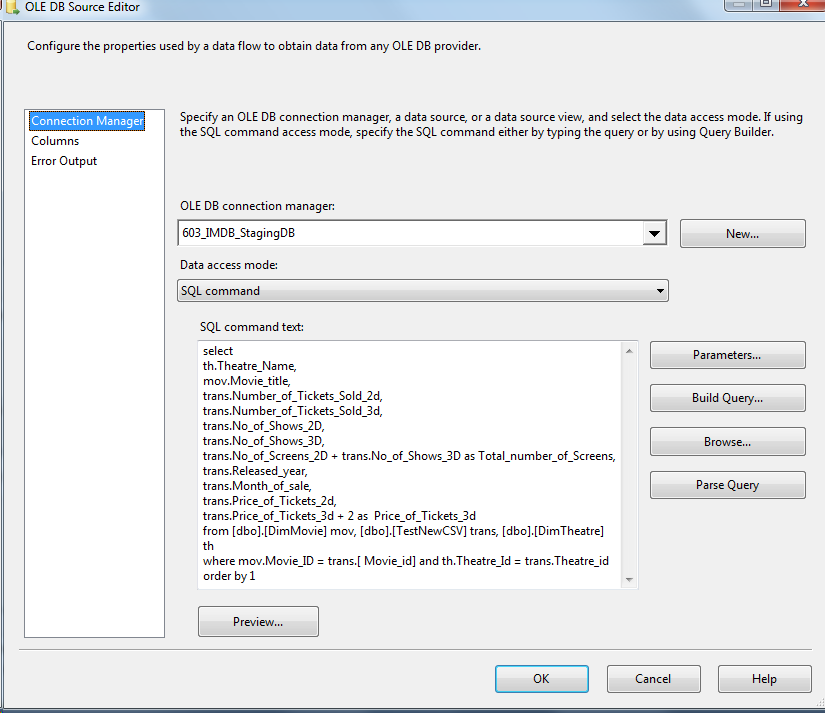


1. Fact Theatre

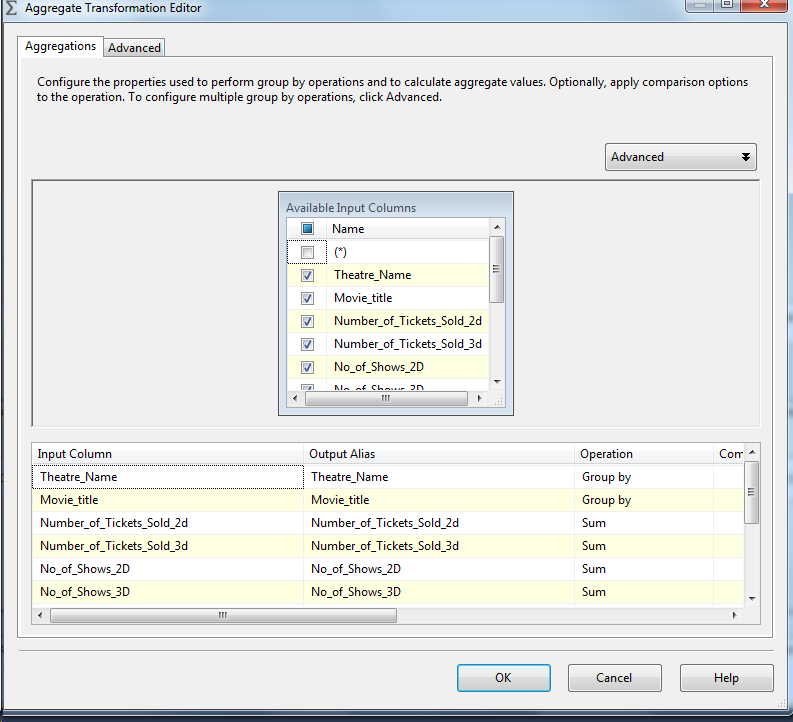
Step 1: Establish Fact Theatre Control Flow



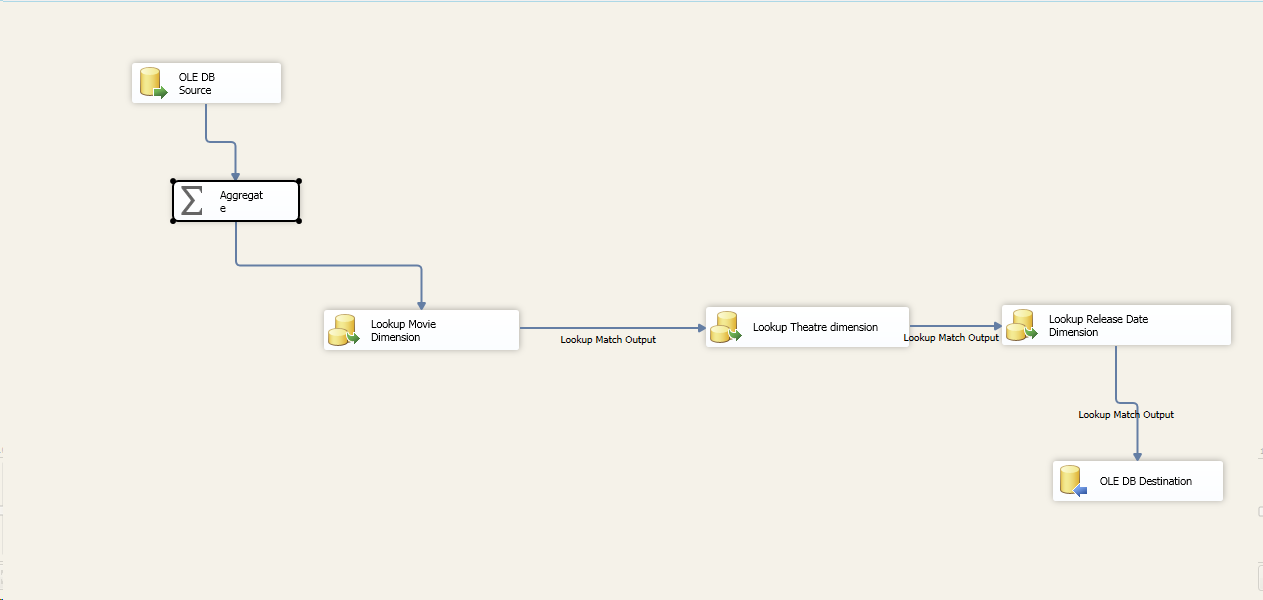
Step 2: Establishing Pre fact query and providing it as OLE DB data source:



Step 3: Performing aggregation on the pre Fact query:



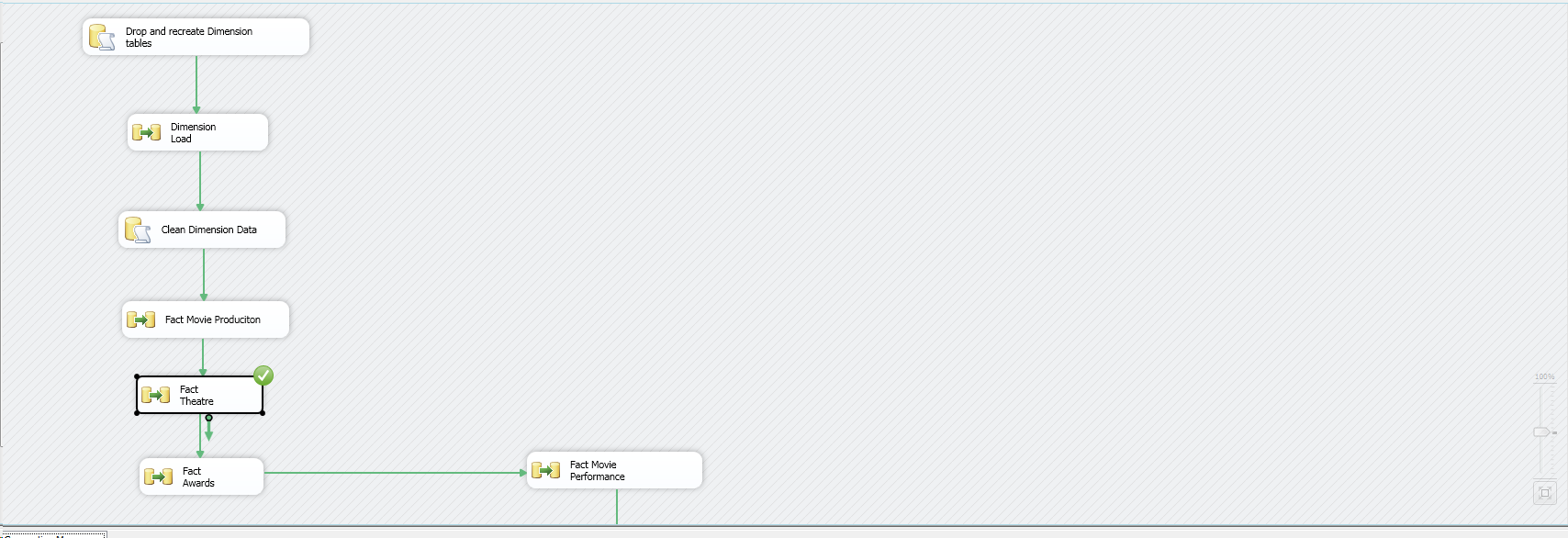
Step 4: Making Lookups on all the Dimension Tables which are required:



Step 5: Successful execution of all Lookups:

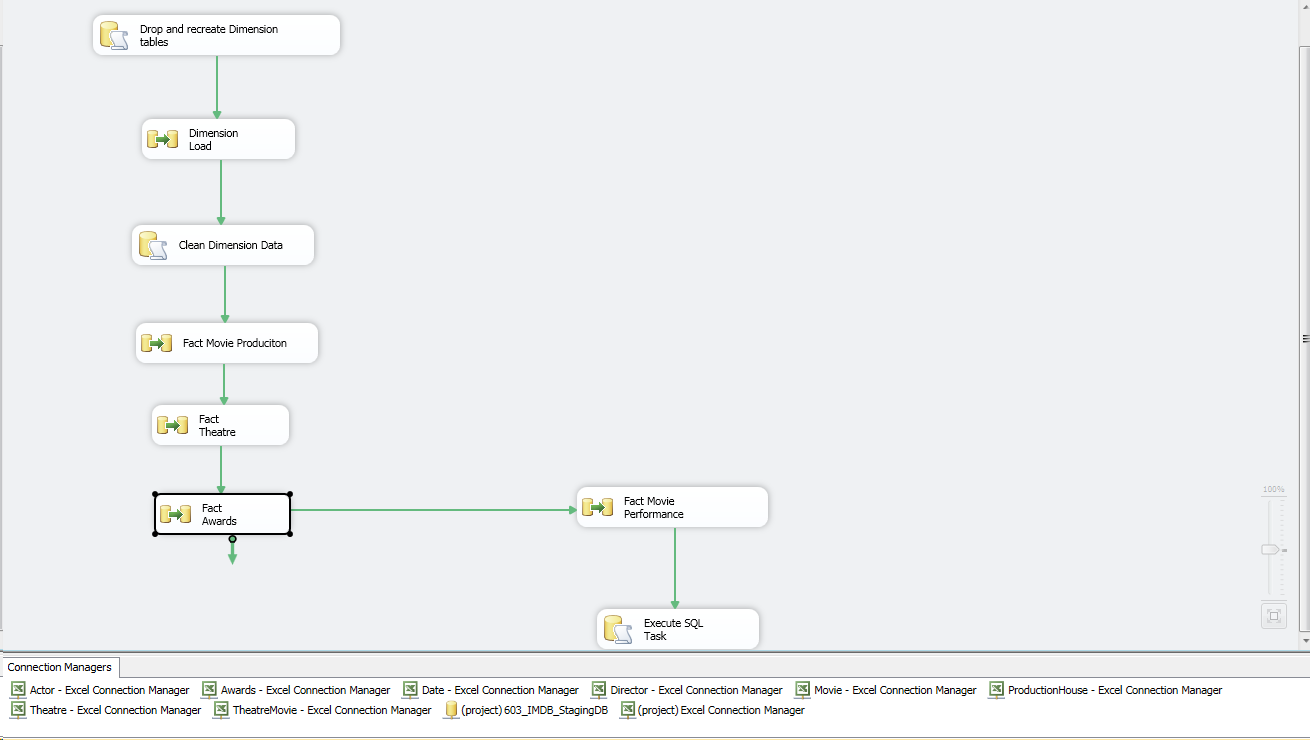


Step 6: Executing fact- theatre Control Flow:

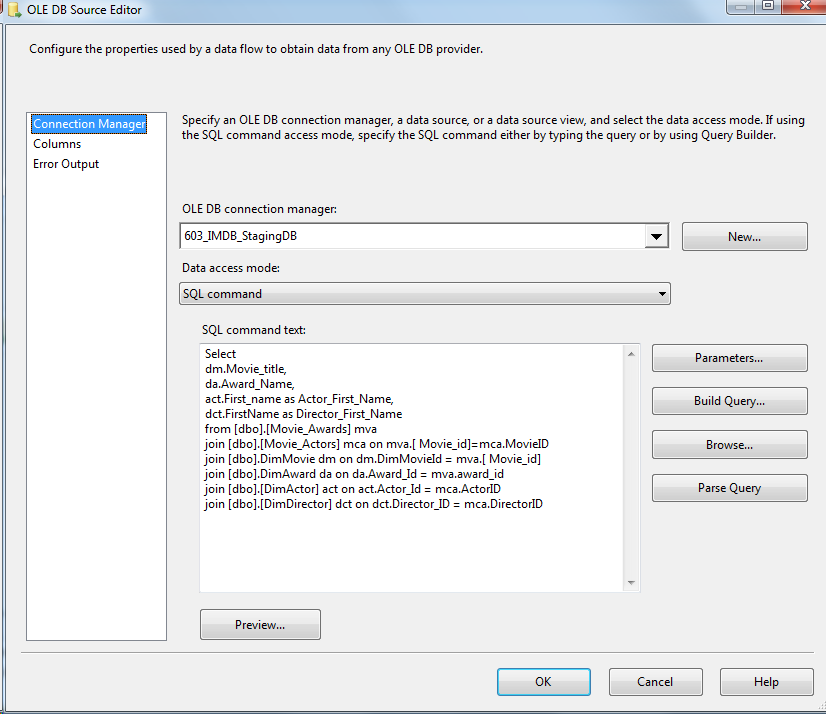


1. Fact Awards

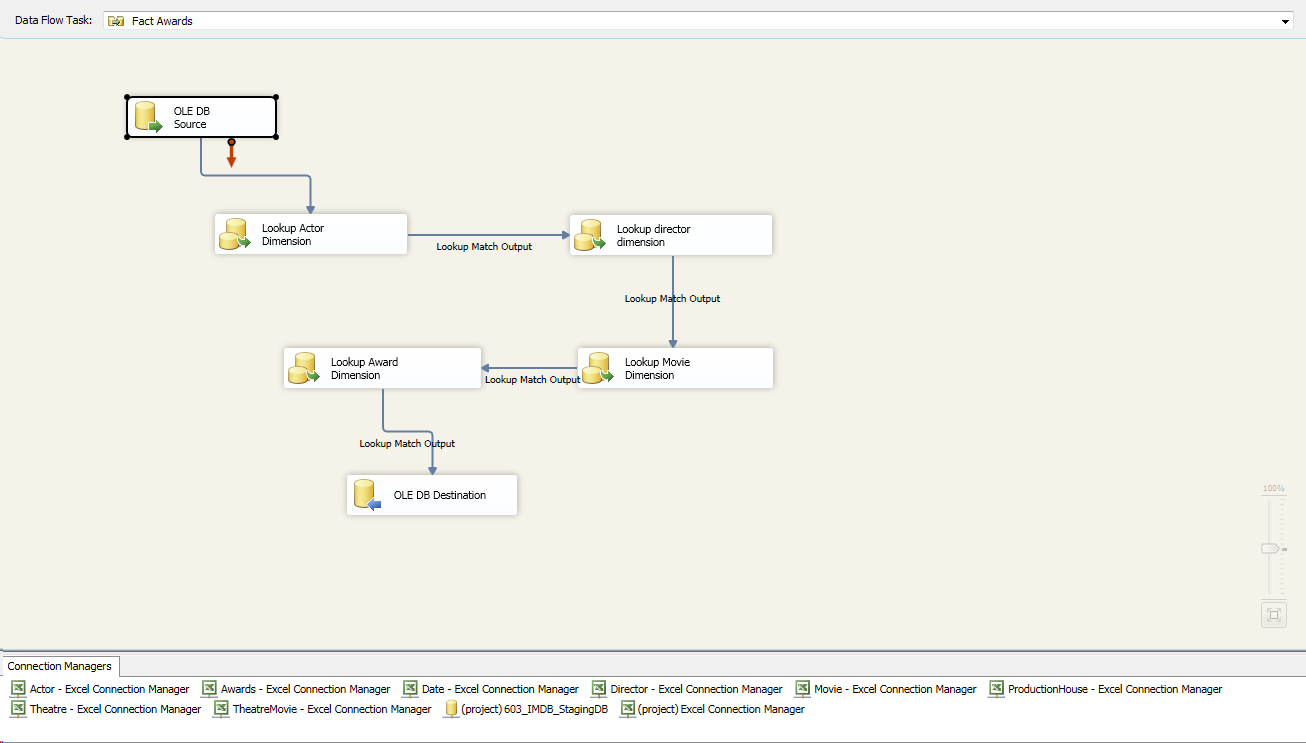
Step 1: Establish Fact Awards Control Flow



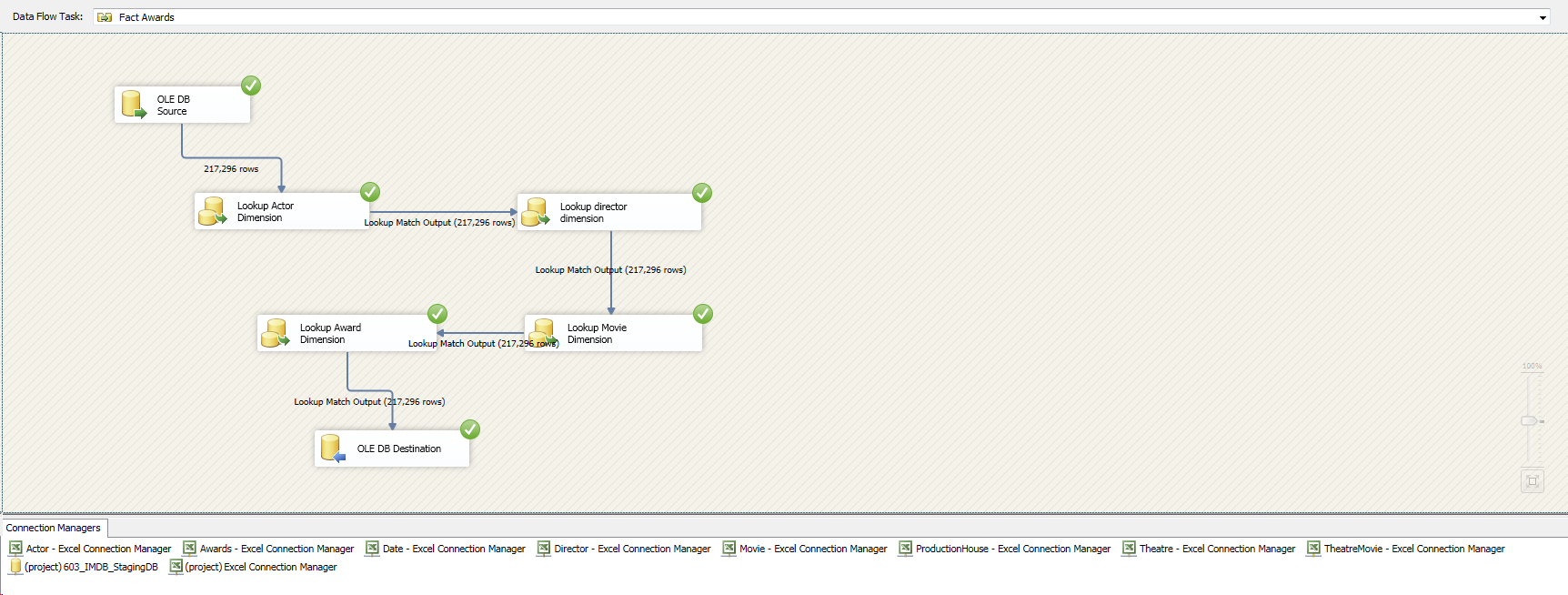
Step 2: Establishing Pre fact query and providing it as OLE DB data source:



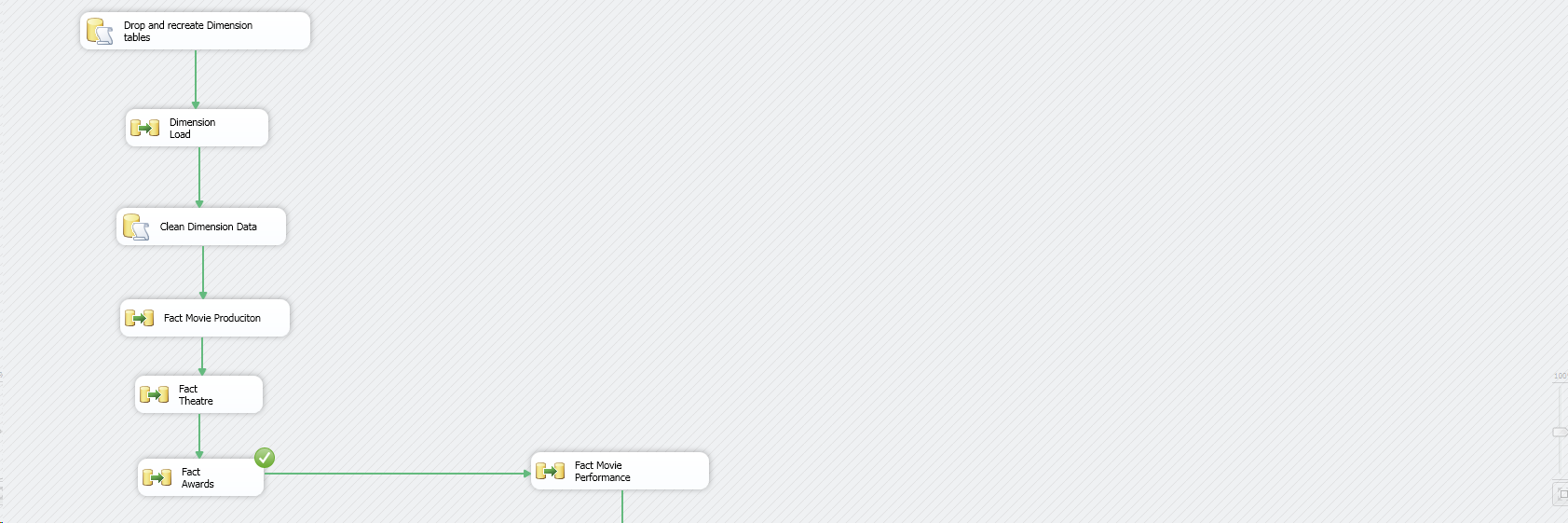
Step 3: Making Lookups on all the Dimension Tables which are required:



Step 4: Successful execution of all Lookups:

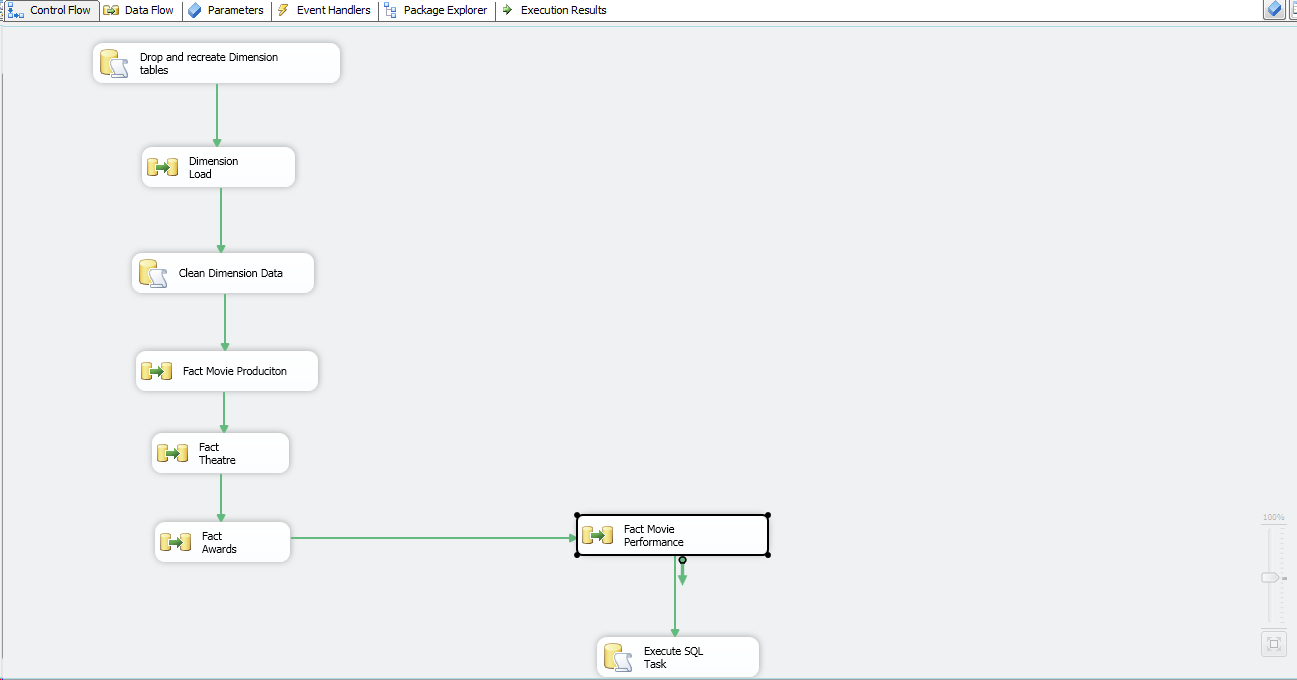


Step 5: Executing fact- Awards Control Flow:

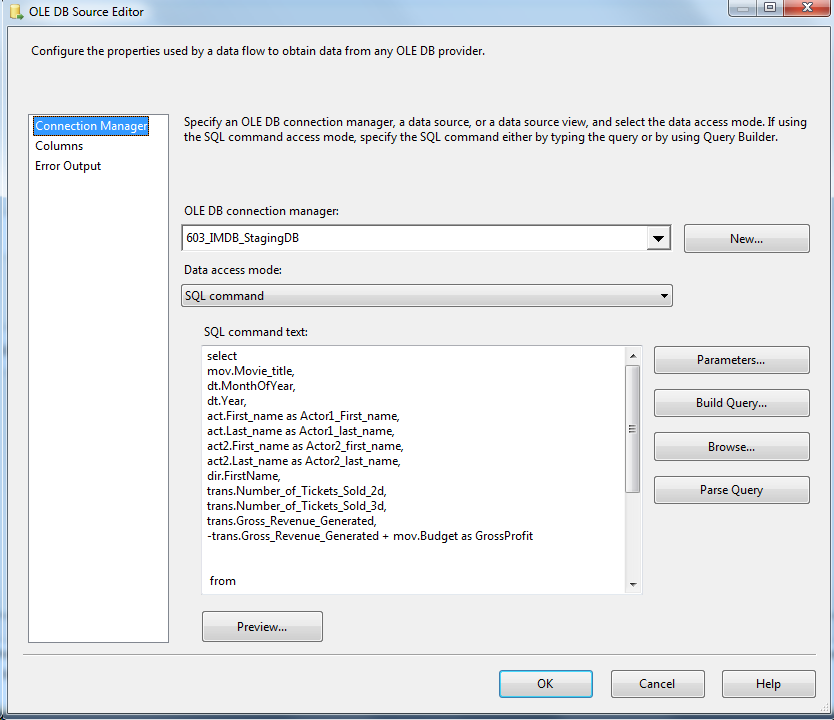


1. Fact Movie Performance

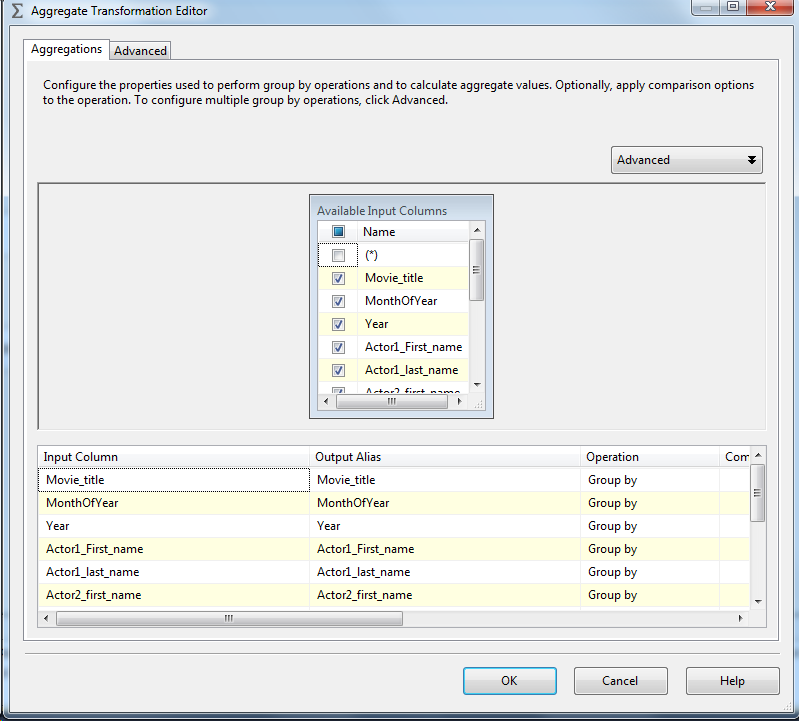
Step 1: Establish Fact Awards Control Flow



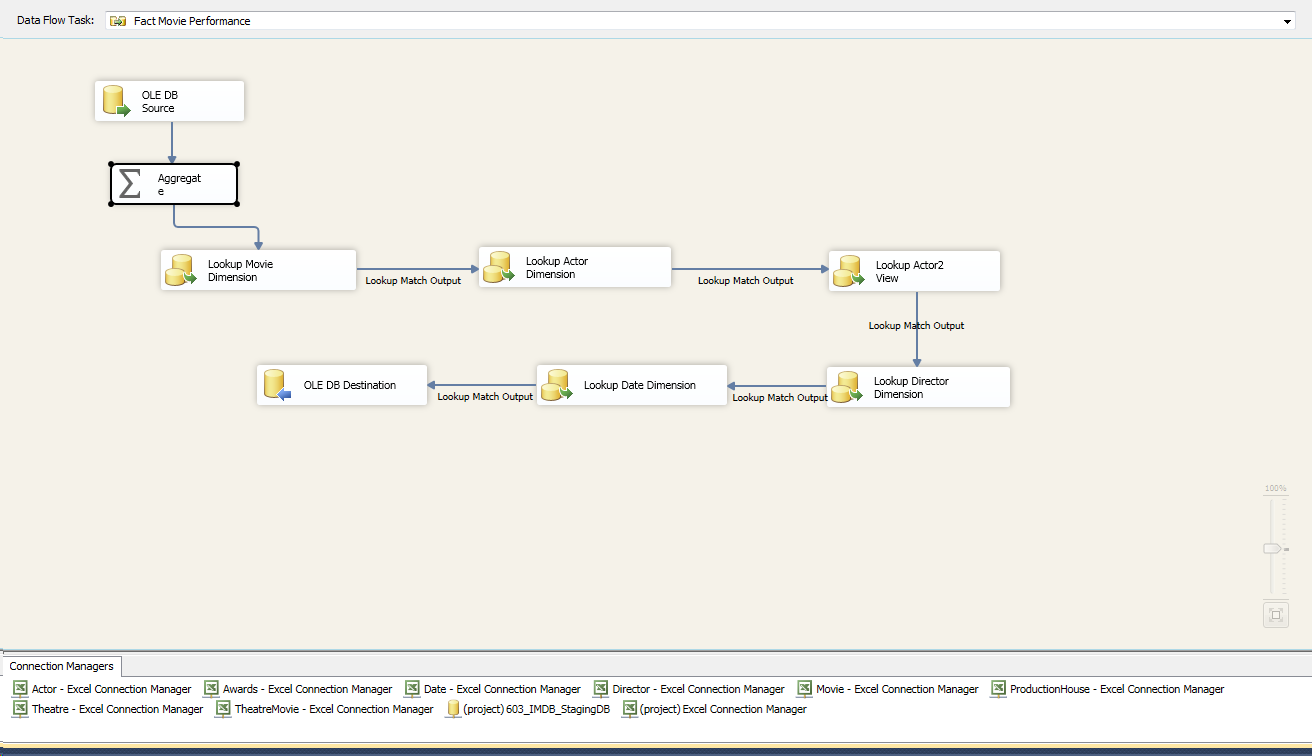
Step 2: Establishing Pre fact query and providing it as OLE DB data source:



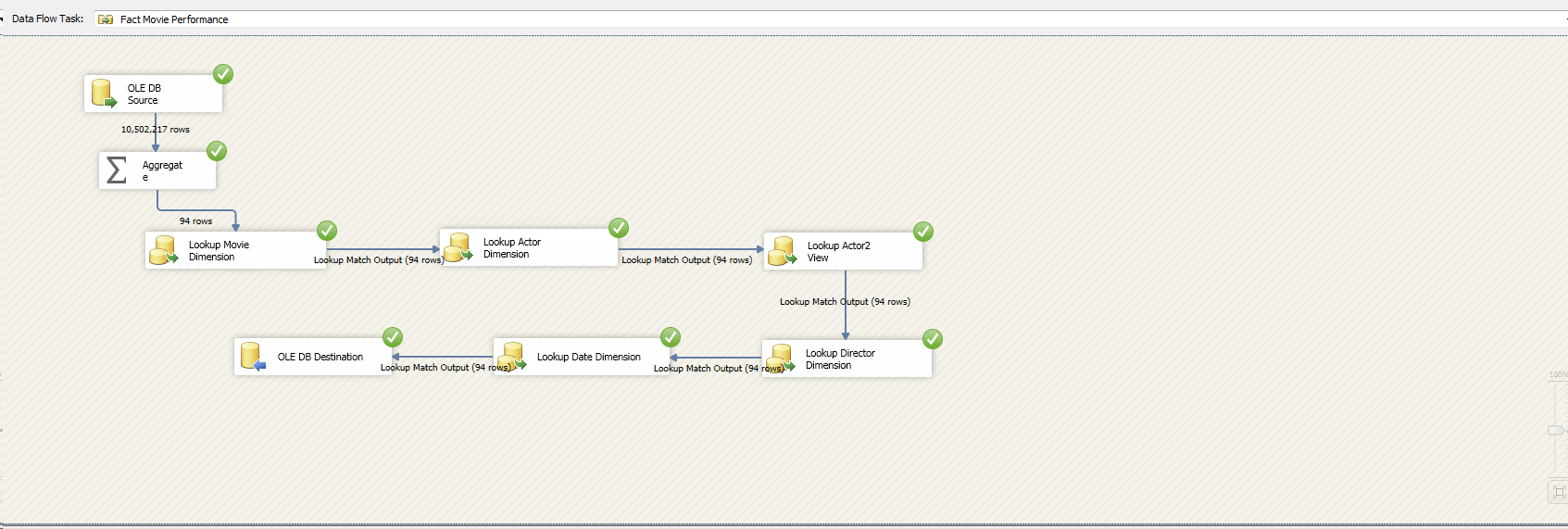
Step 3: Performing aggregation on the pre Fact query:



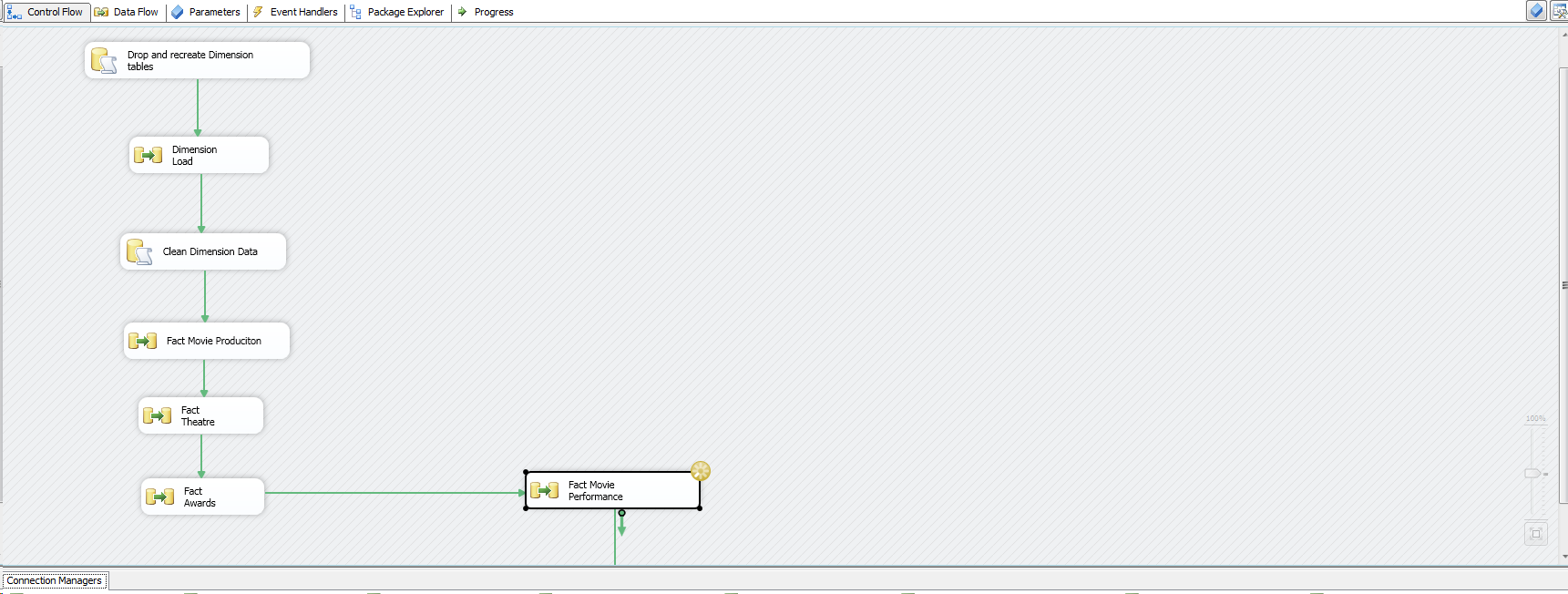
Step 4: Making Lookups on all the Dimension Tables which are required:

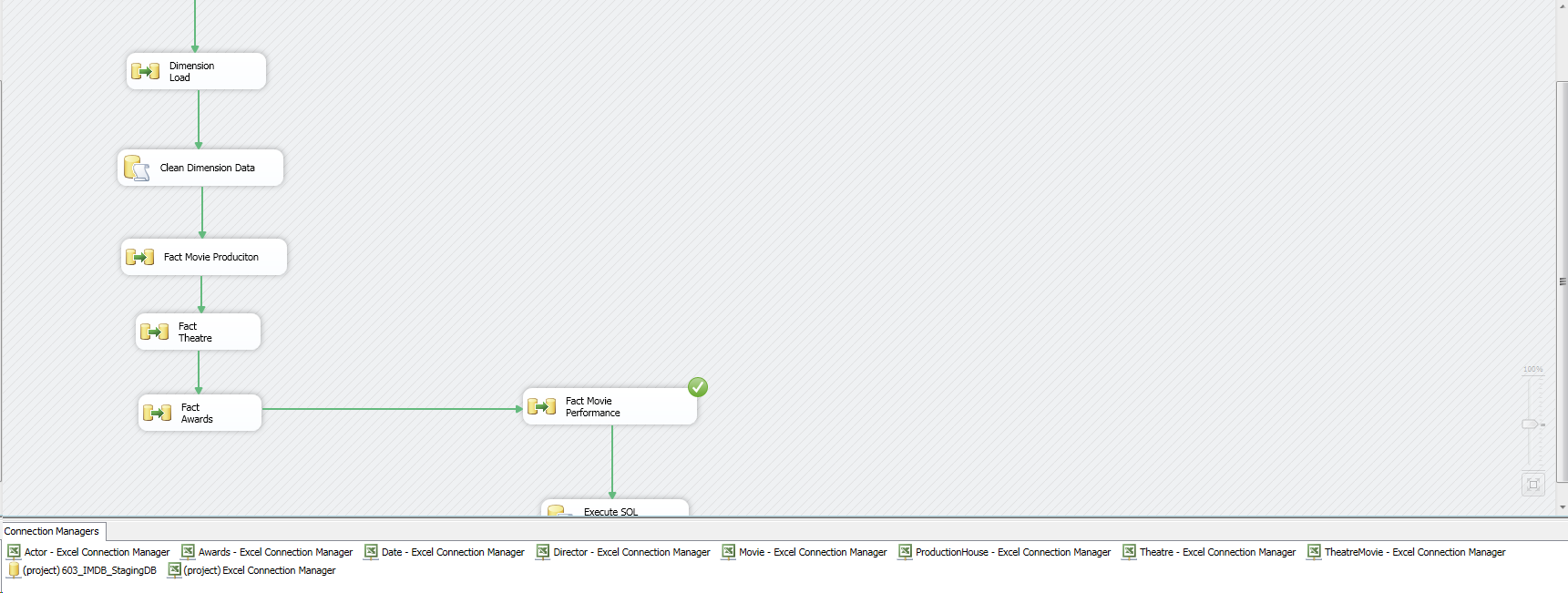


Step 5: Successful execution of all Lookups:



Step 6: Executing fact- Awards Control Flow:





# **Business Intelligence Reporting**

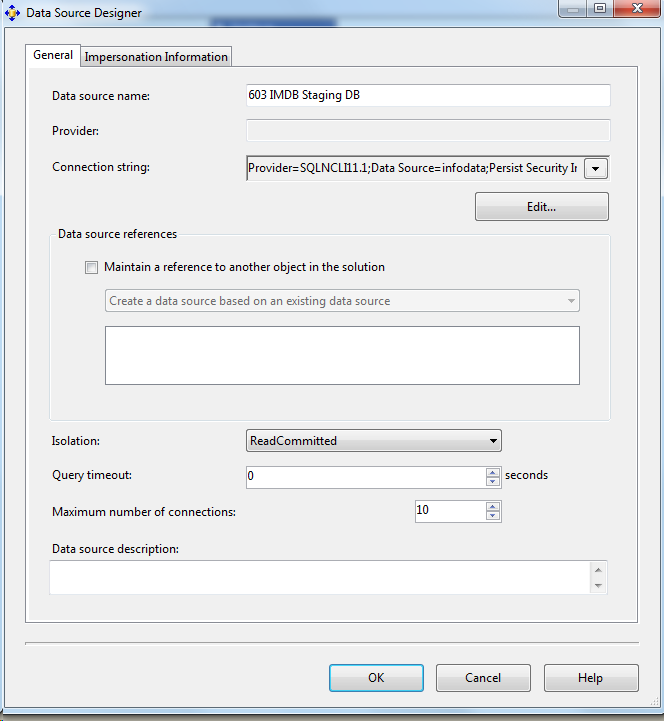
“Business intelligence (BI) is a technology-driven process for analyzing data and presenting actionable information to help corporate executives, business managers and other end users make more informed business decisions. BI encompasses a variety of tools, applications and methodologies that enable organizations to collect data from internal systems and external sources, prepare it for analysis, develop and run queries against the data, and create reports, dashboards and data visualizations to make the analytical results available to corporate decision makers as well as operational workers.”[4]

|  |  |  |
| --- | --- | --- |
| **Reporting Tool** | **Data Mart** | **Questions Answered** |
| SSRS over SSAS | Movie Performance | 4 |
| SSRS over SSAS | Production Performance | 1 |
| SSRS over SSAS | Awards Distributions | 1 |
| SSRS over SSAS | Theatre Performance | 2 |

## **Data Mart creation using SSAS**

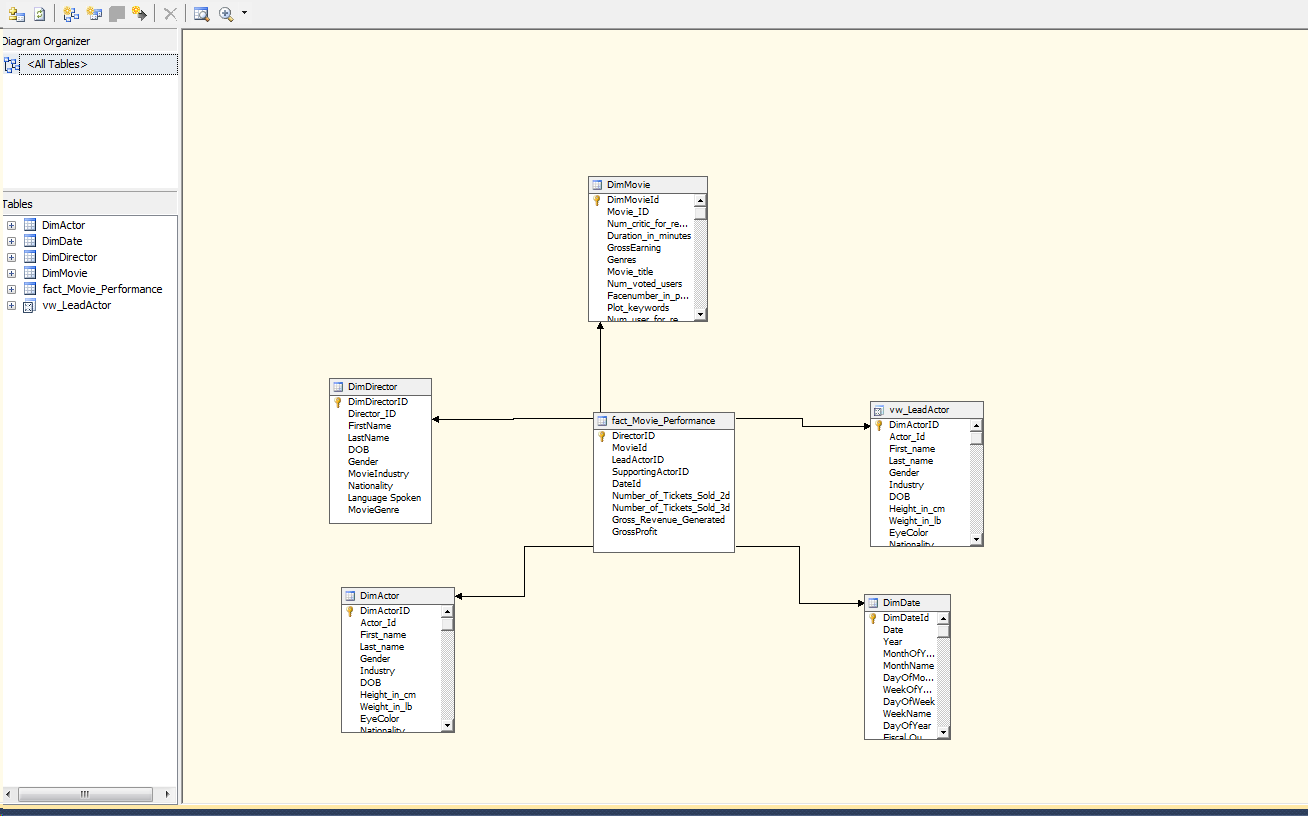
Following are the screenshots for systematic creation of a data mart.

Establishing Data Source for making cubes

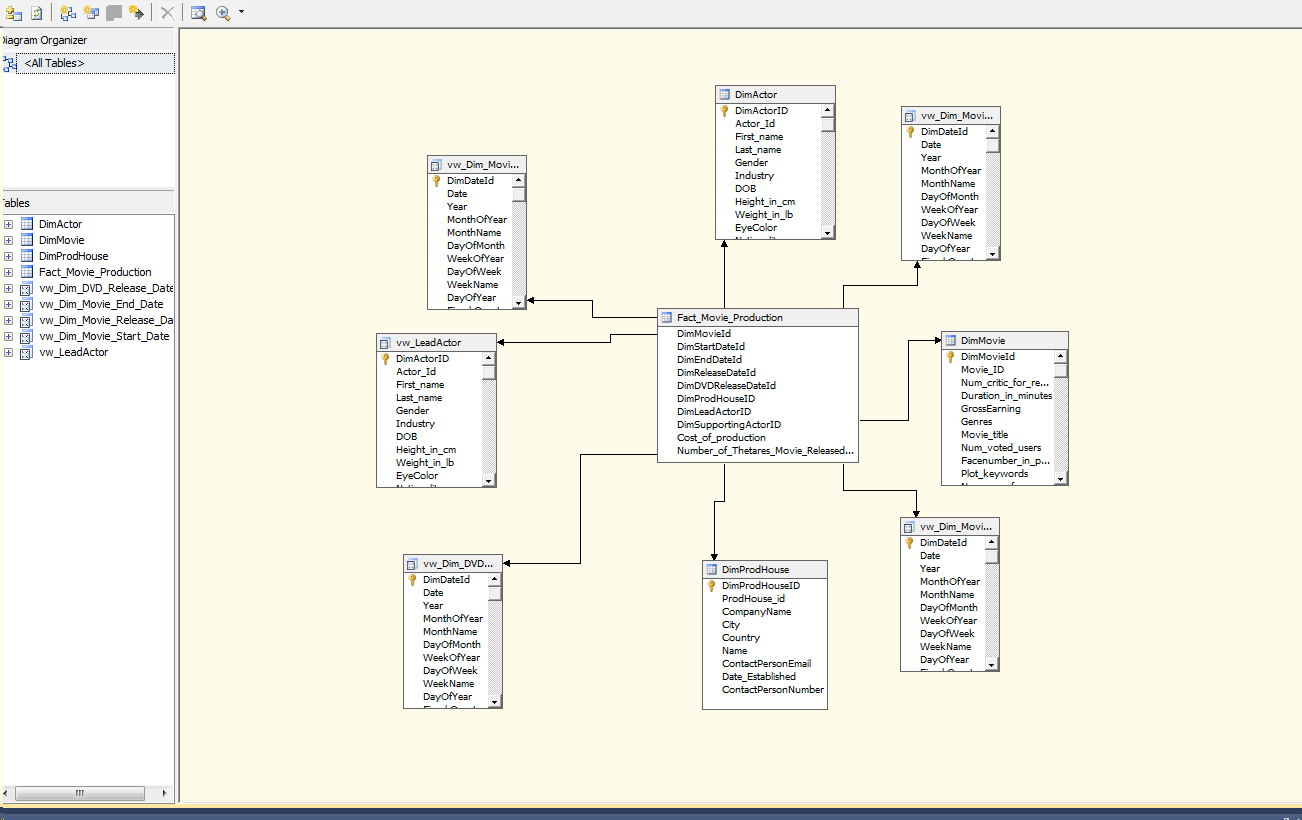


Creating Data Source views for 4 Data Marts

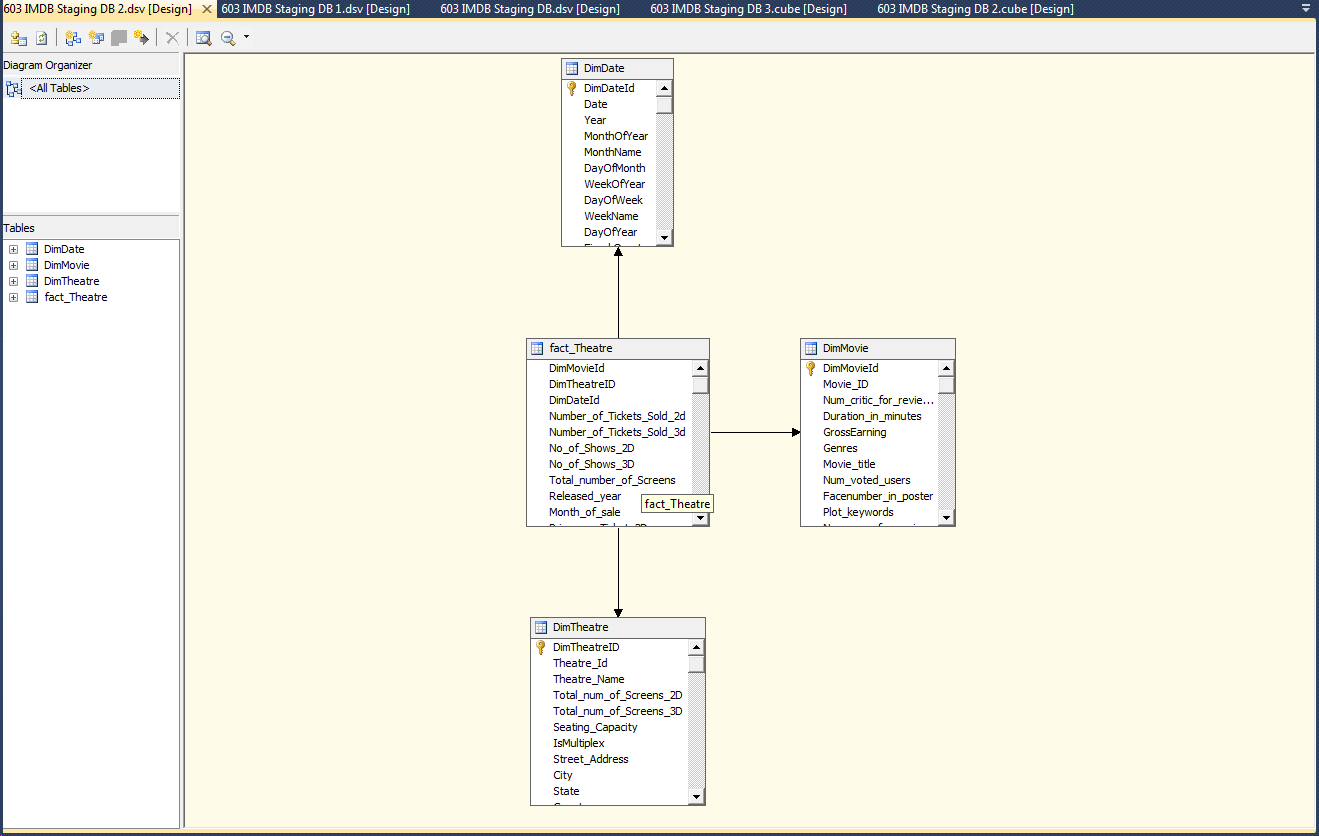
**Movie Performance**



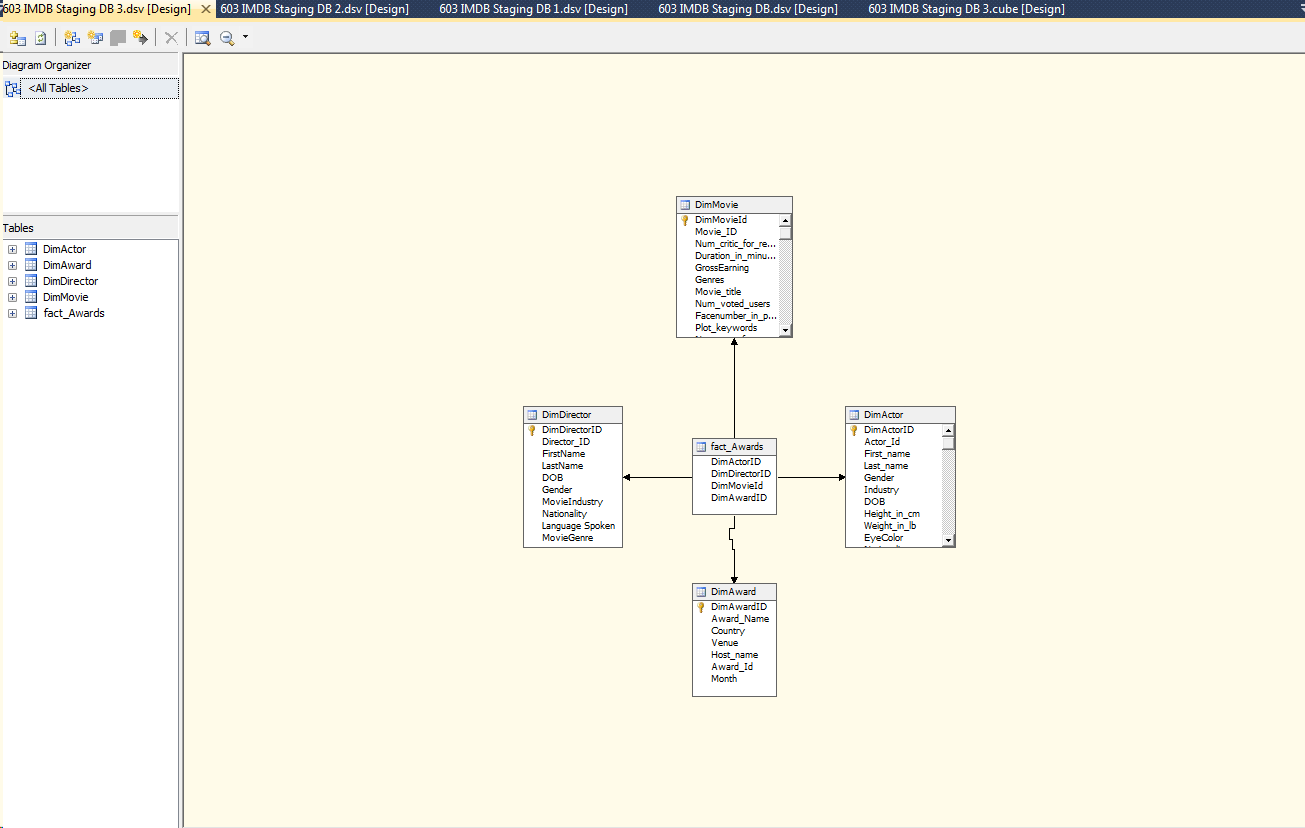
**Production House Performance**



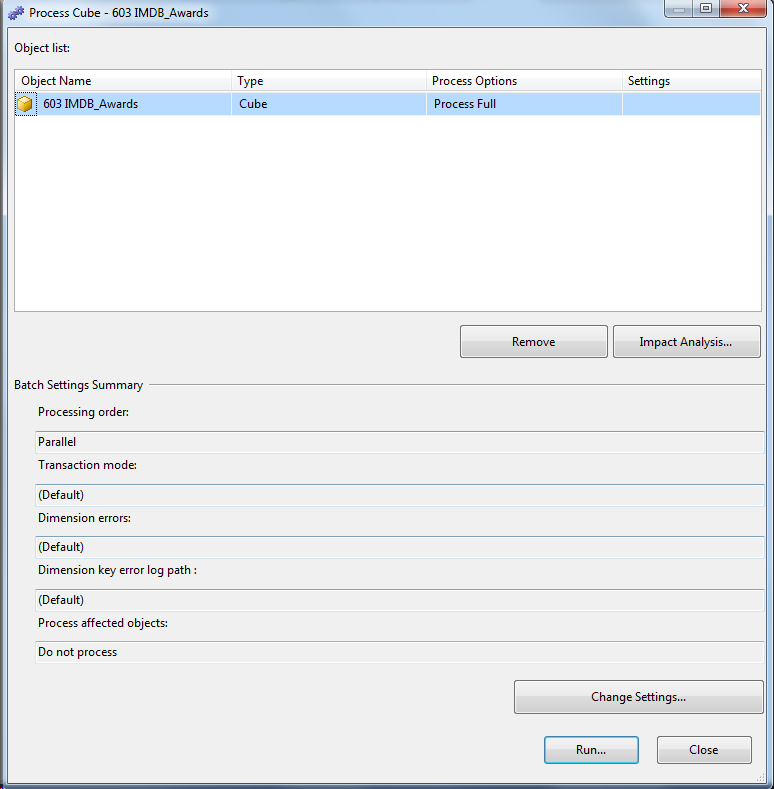
**Theatre Performance**

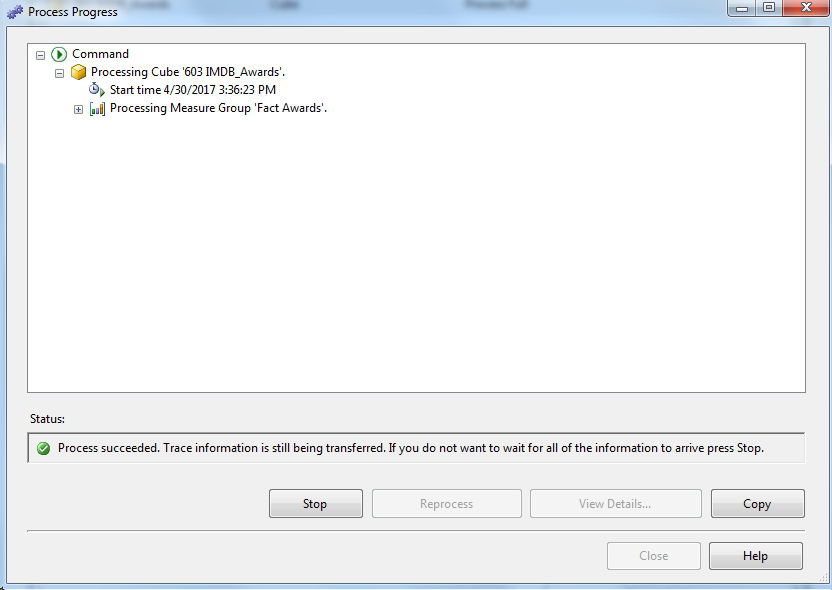


**Awards Distribution**



Following steps are followed for the processing of the cubes:

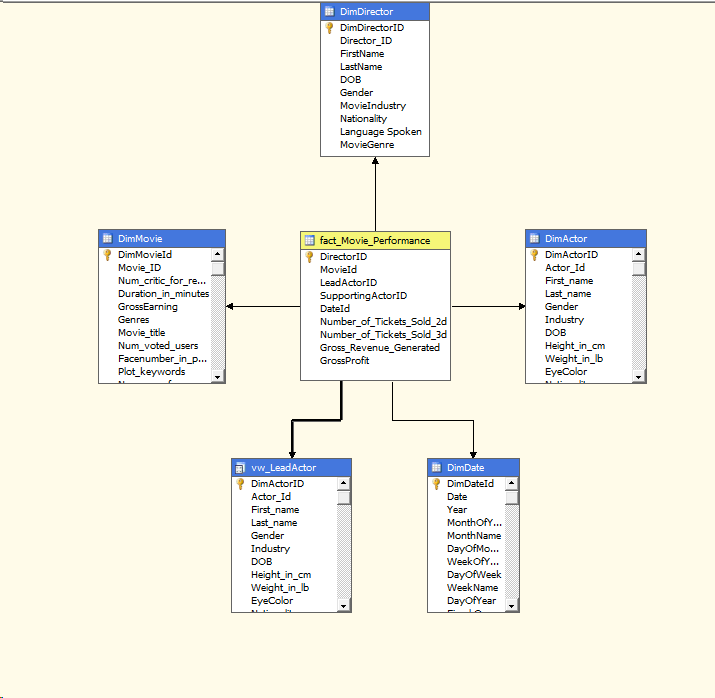




## **Report building from individual Data Mart is SSRS**

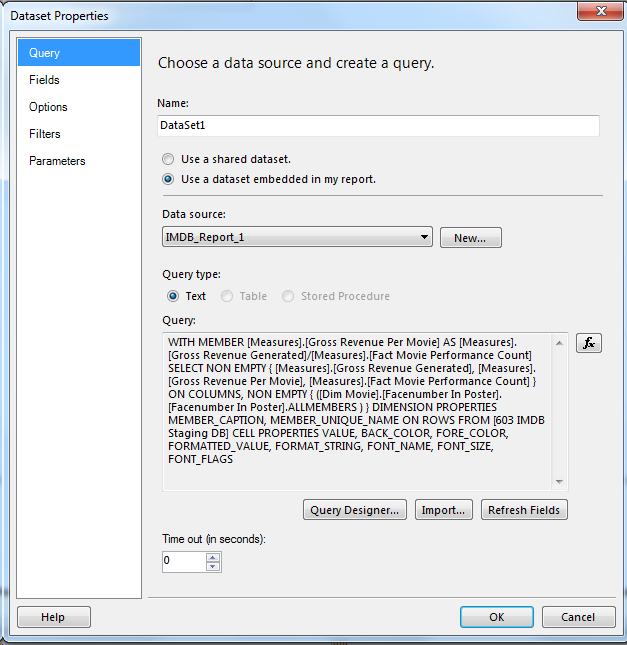
## **Movie Performance**

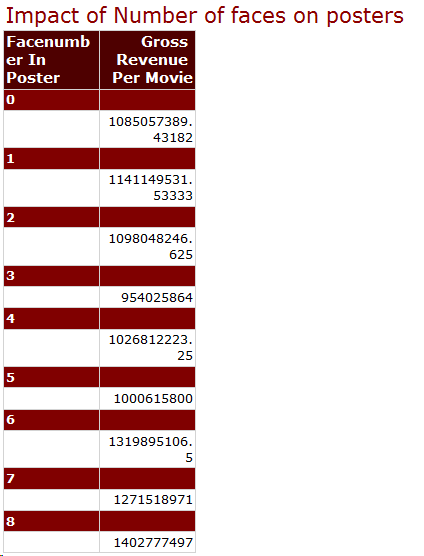
**Cube Name – IMDB\_Movie\_Performance.cube**

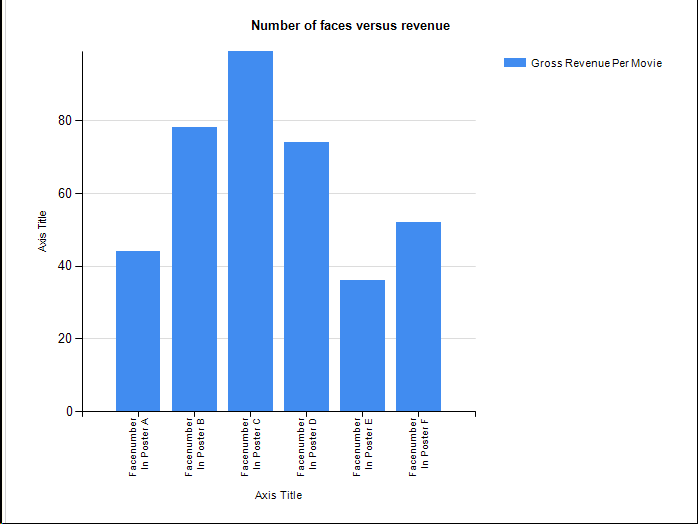


**Business Questions Catered**

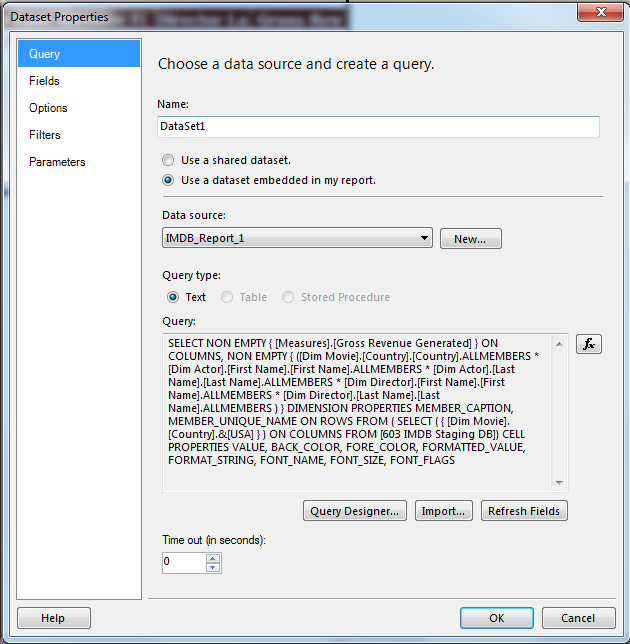
1. **What is the impact of number of faces in a poster on generated revenue generated per movie?**





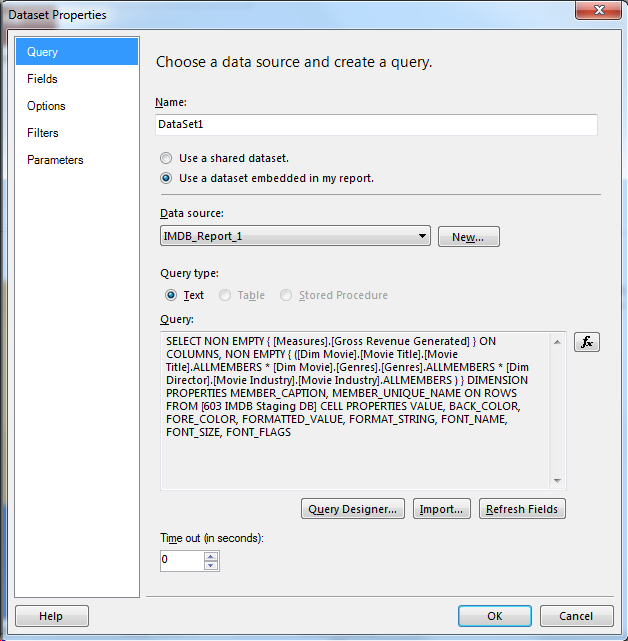


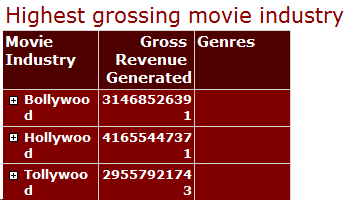
1. **What are the best actor director combination for a movie generating highest revenue based on each country?**

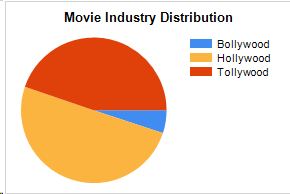


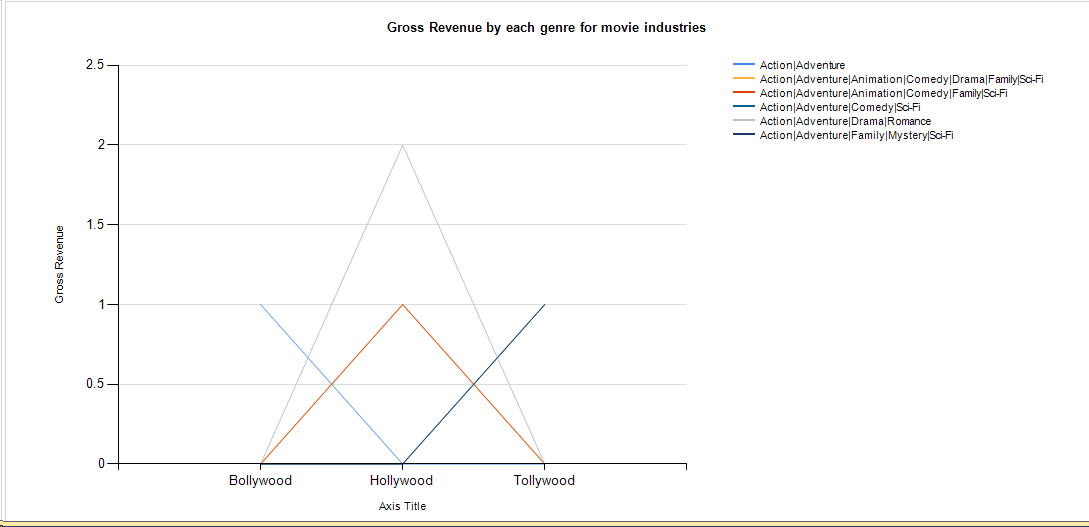


1. **What are the highest grossing movie industries and what genres are the highest grossers in each industry?**

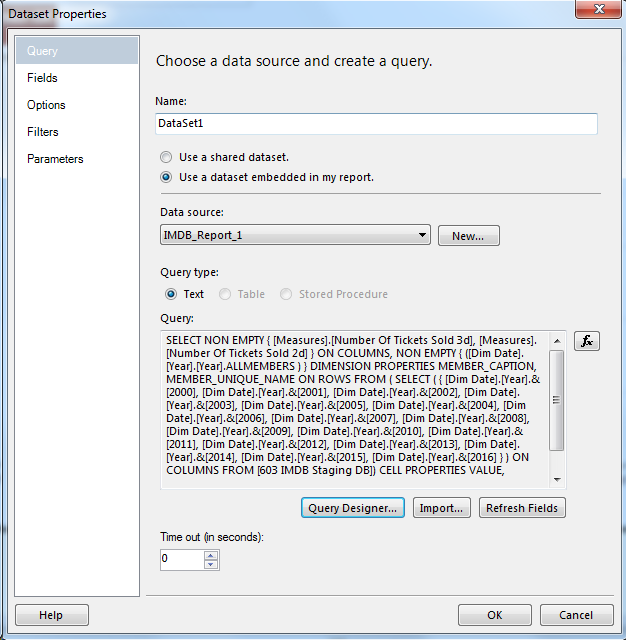


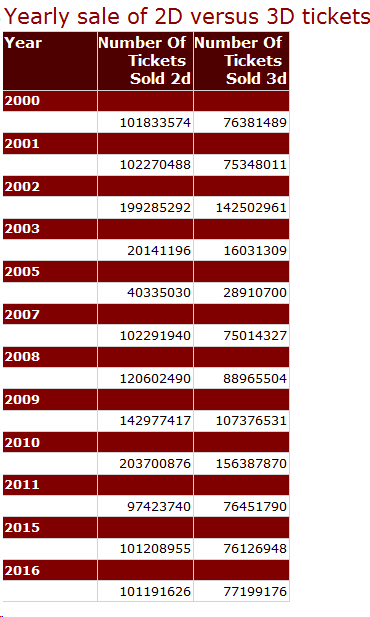






1. **What is the popularity trend for 2D movies vs 3D movies in last one decade?**

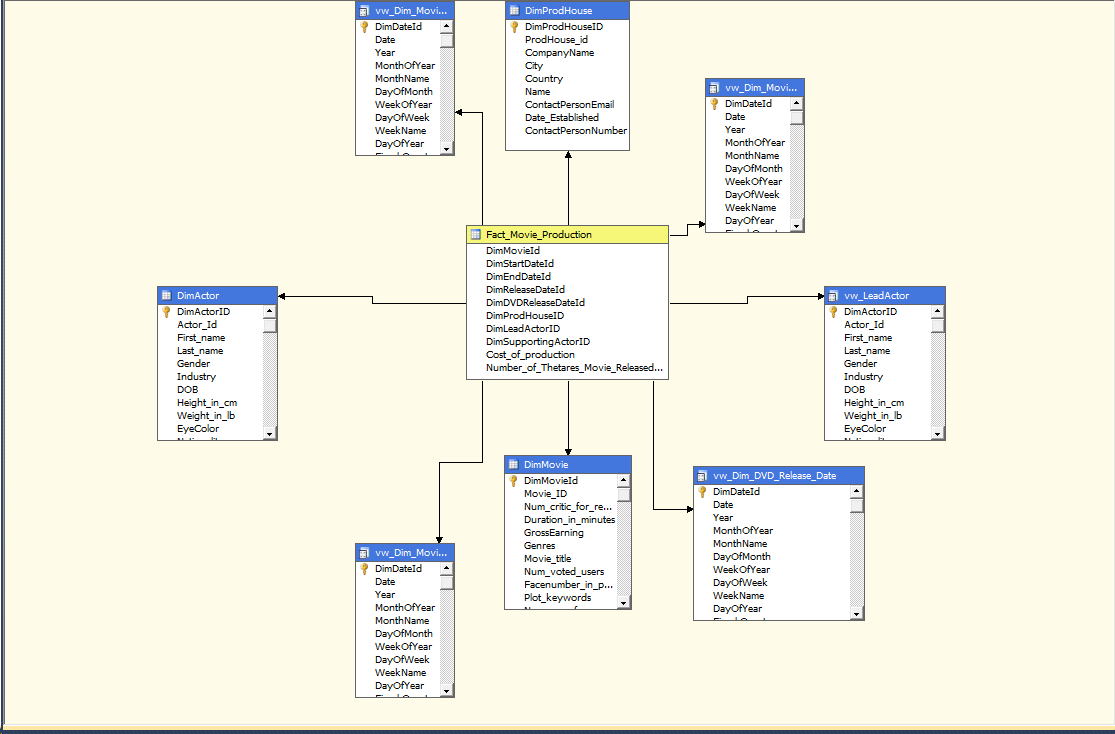






### **Production House Performance**

**Cube Name – IMDB\_Production\_House\_Performance.cube**



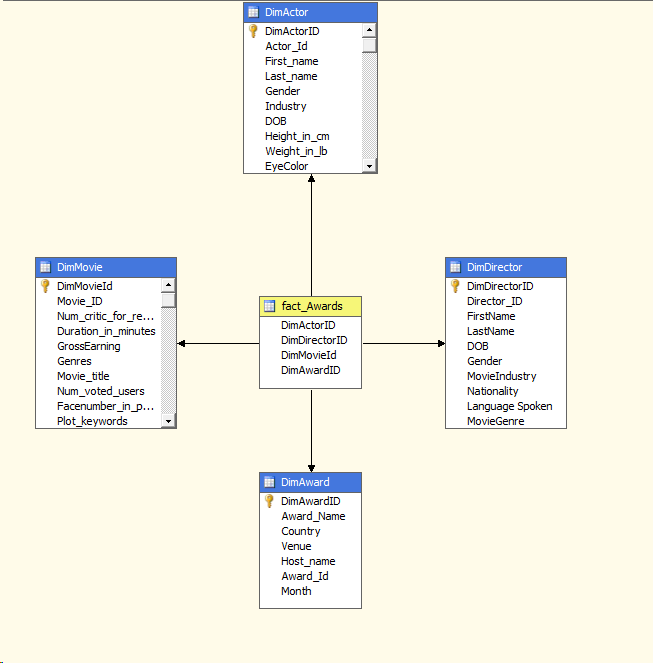
**Business Questions Catered**

1. **What pair of actors casted in a movie result in higher cost of production?**



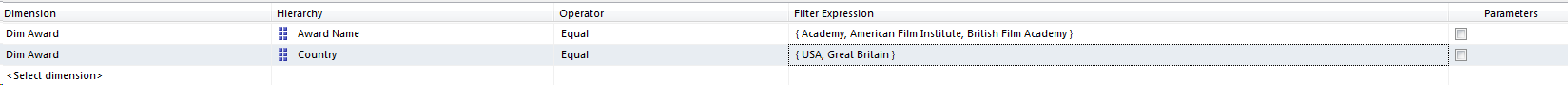
### **Awards Distribution**

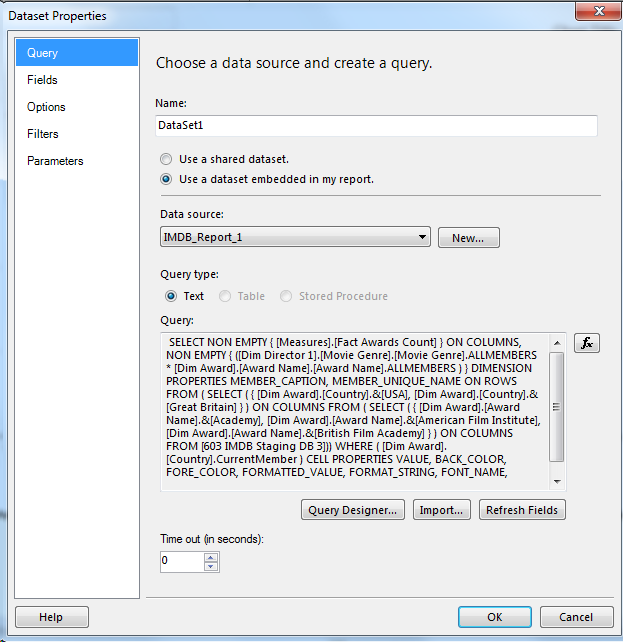
**Cube Name – IMDB\_Awards\_distribution.cube**

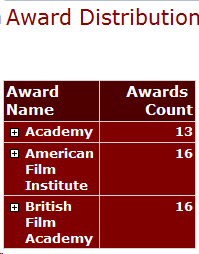


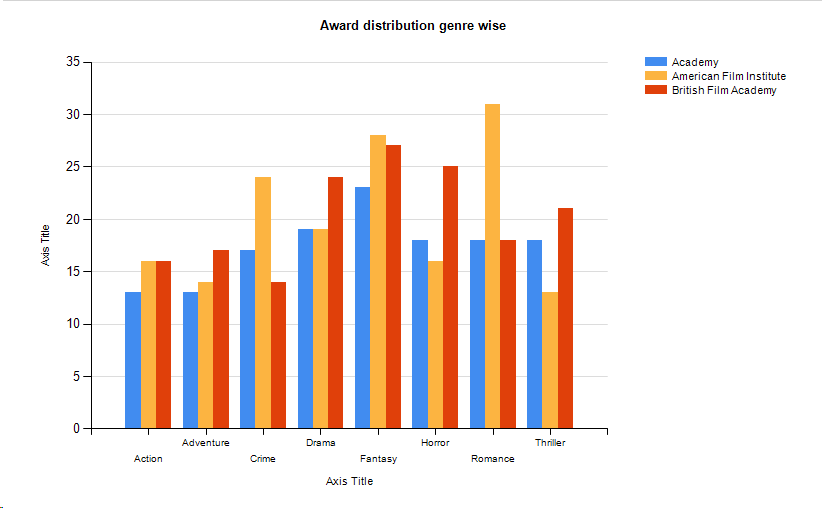
**Business Questions Catered**

1. **What are the most popular genre, which won for major awards in a particular country?**



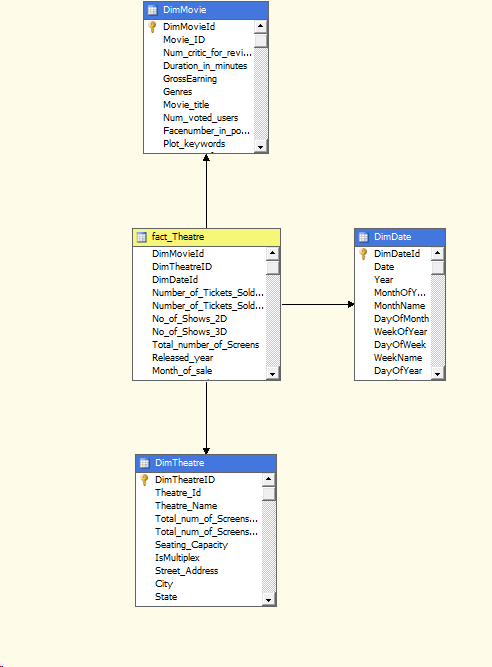






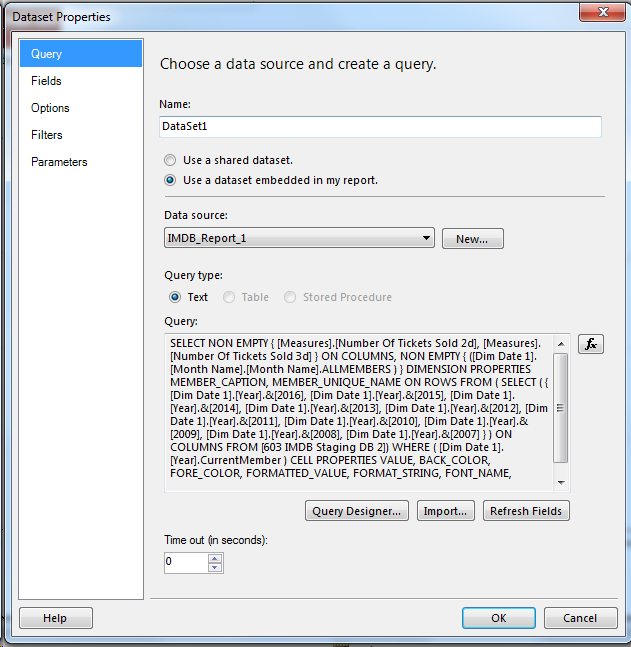
### **Theatre Performance**

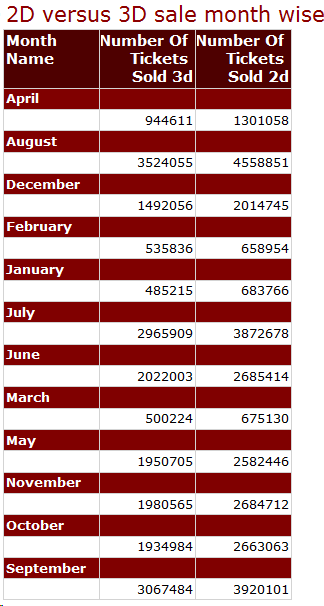
**Cube Name – IMDB\_Theatre\_Performance.cube**

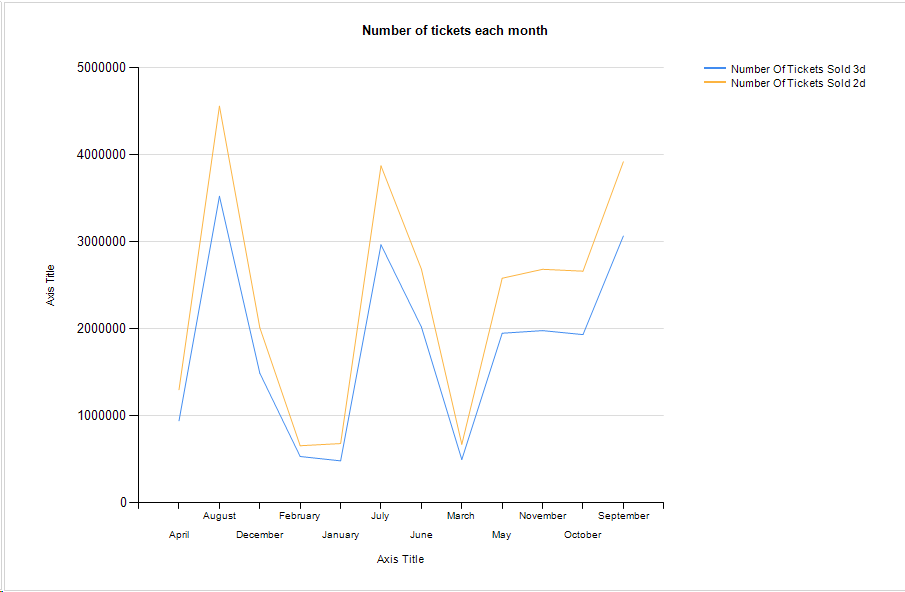


**Business Questions Catered**

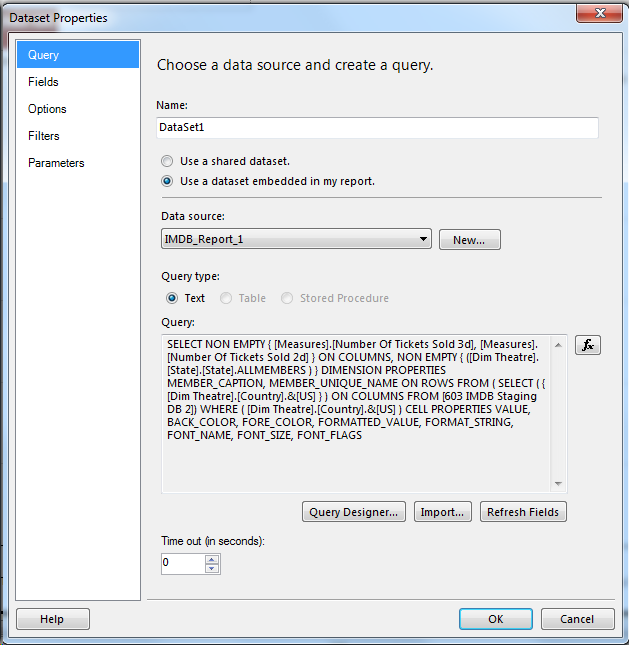
1. **What is the month wise trend of tickets sold across theatres? Which month is the most popular in terms of 2D vs 3D tickets?**

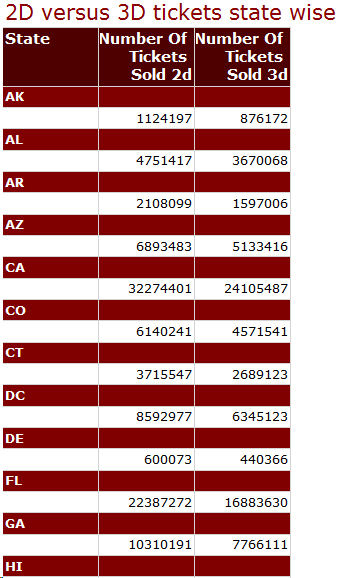


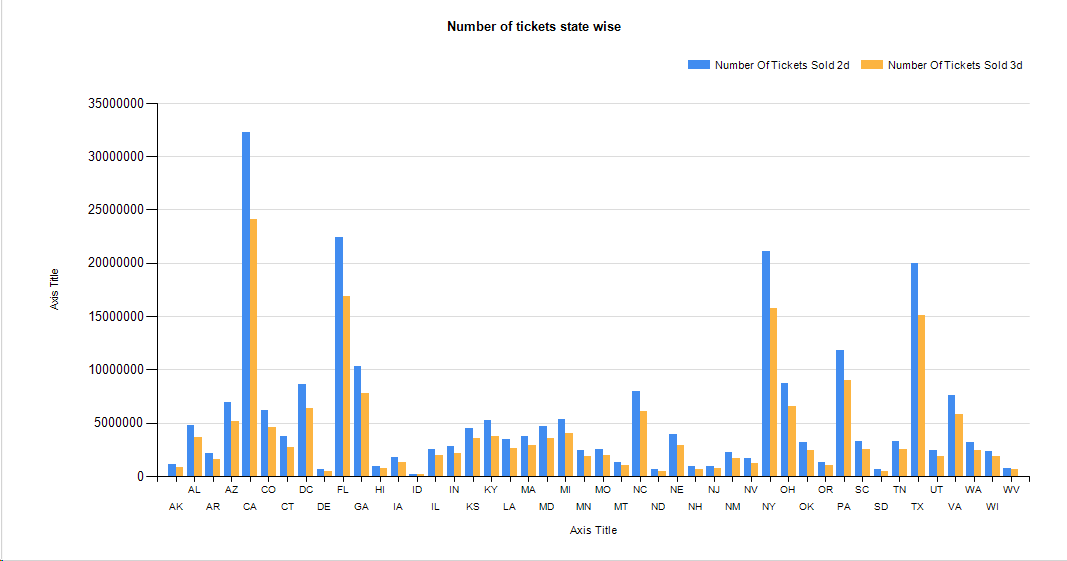




1. **What are the popularity trends of 2D tickets sold versus 3D tickets sold across different states of a country?**



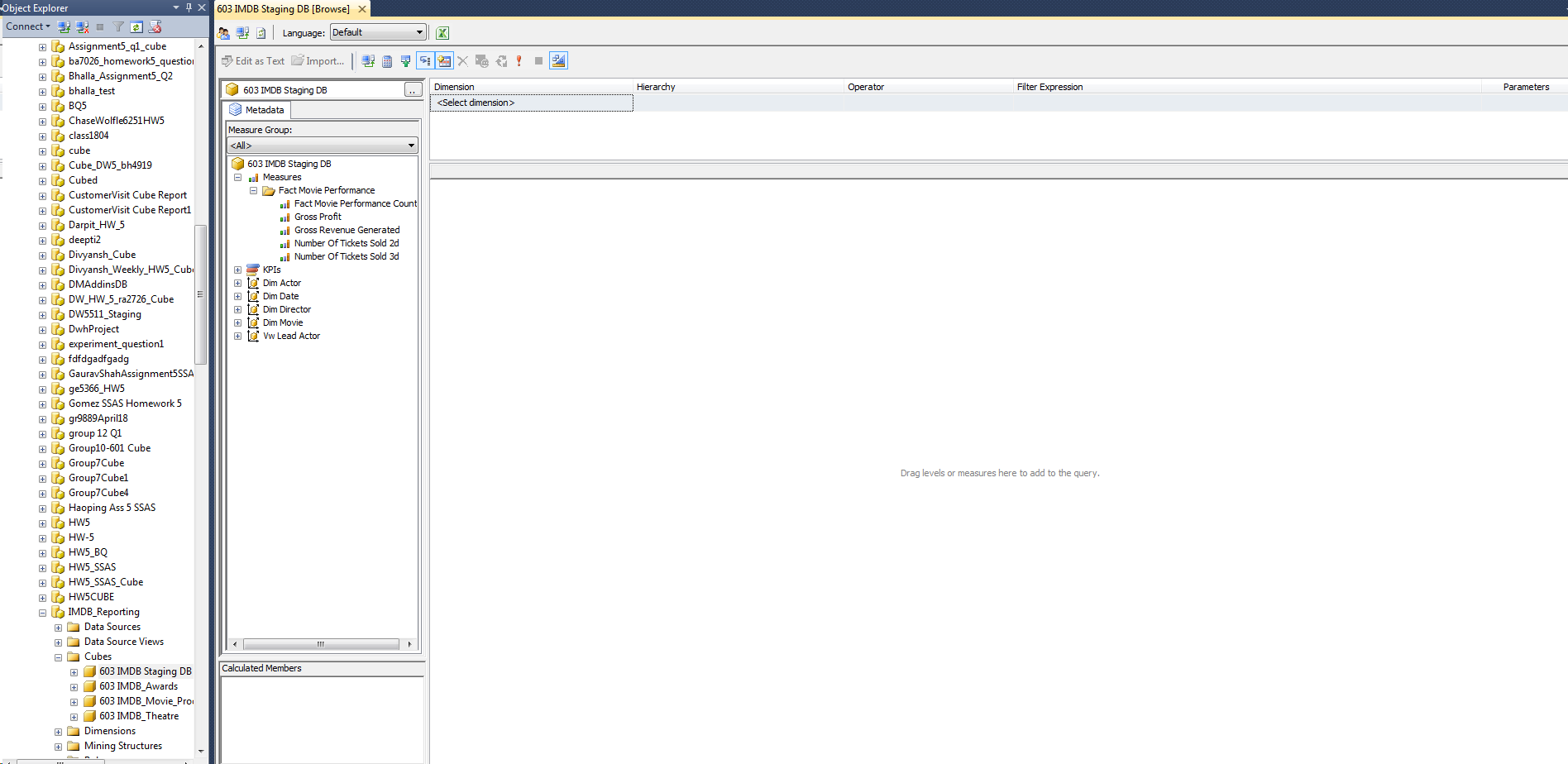




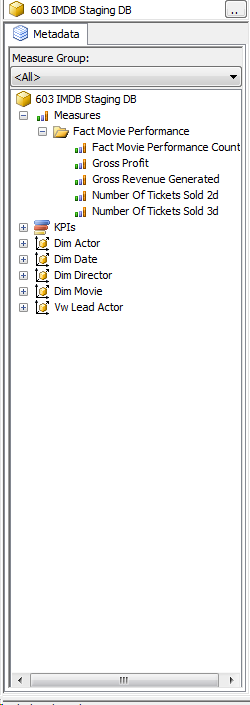
## **User Manual for creating Dynamic Reports**

## **Browsing cubes via SSRS**

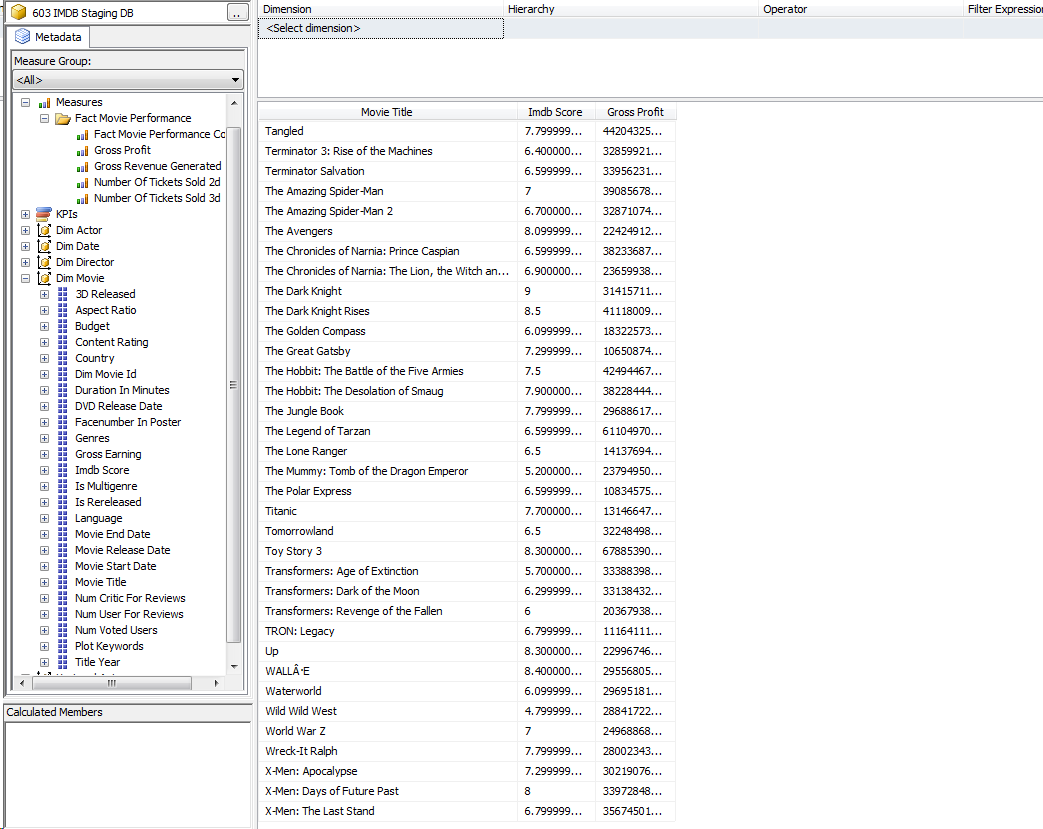
Browse the desired cube. The cubes have been deployed on server and can be browsed easily.



On the left pane, user will see all the measures and dimensions for a particular data mart.



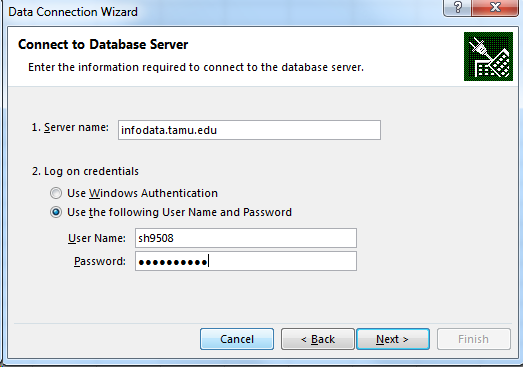
User can drag and drop the measures and corresponding fields from the dimensions which the user requires.

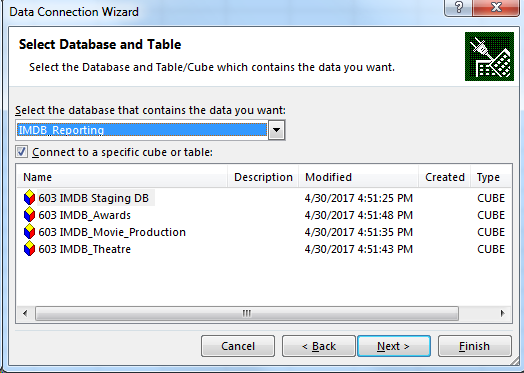


Similarly, user can browse any of the 4 cubes to determine trends and generate reports he/she needs.

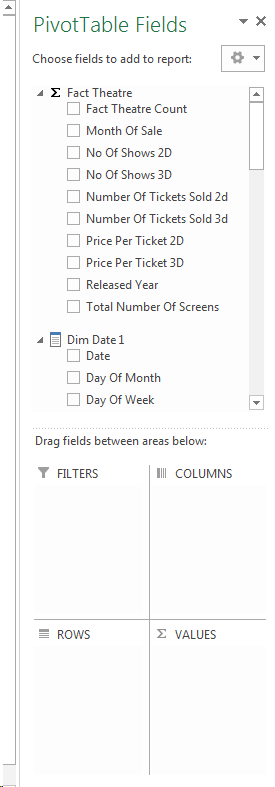
### **Browsing cubes using pivot**

Connecting to the cubes

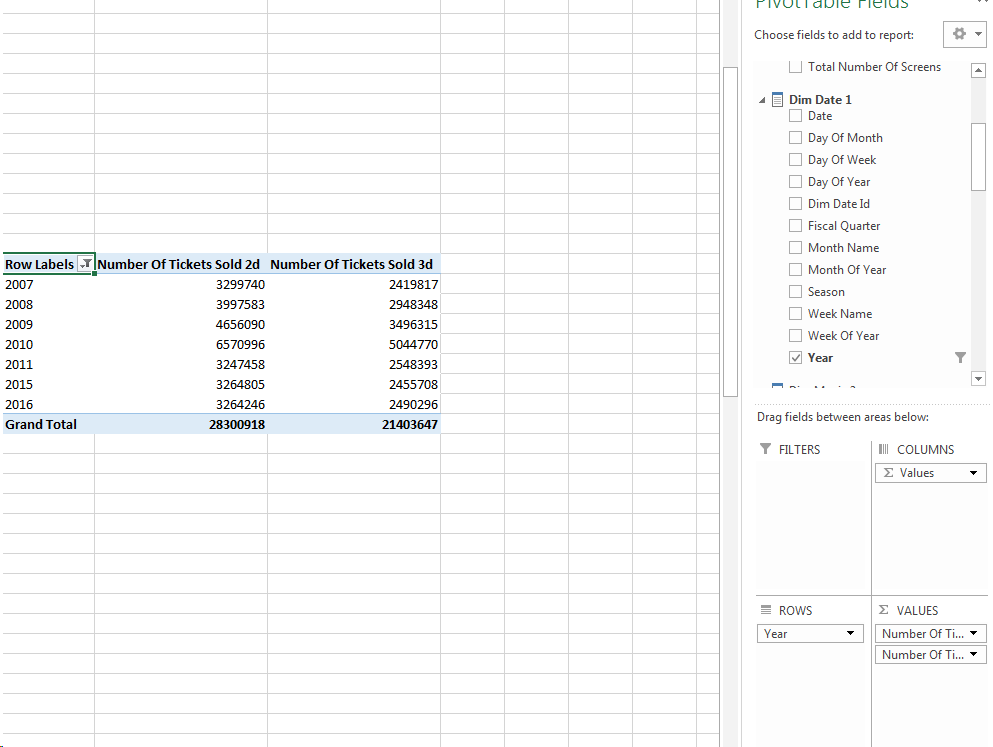


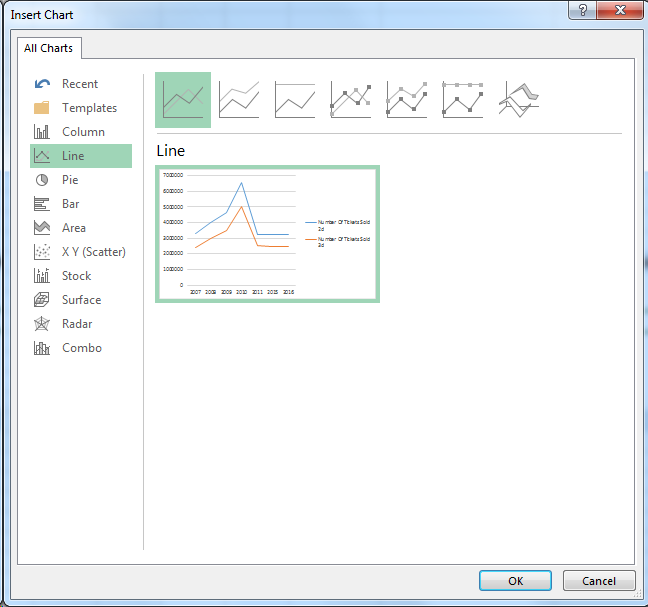


User can drag and drop the measures and corresponding fields from the dimensions which the user requires.

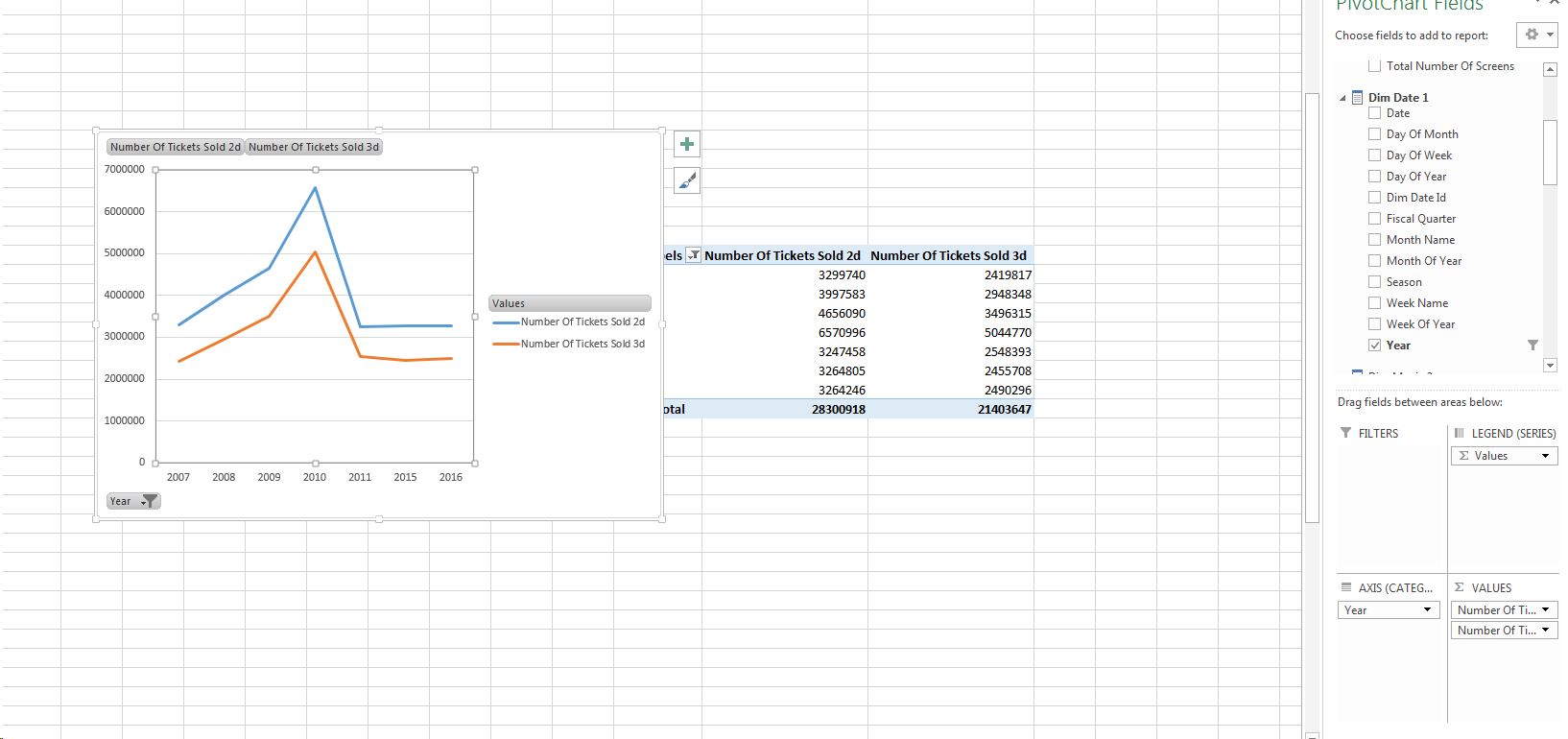


Sample report generated after user selection:

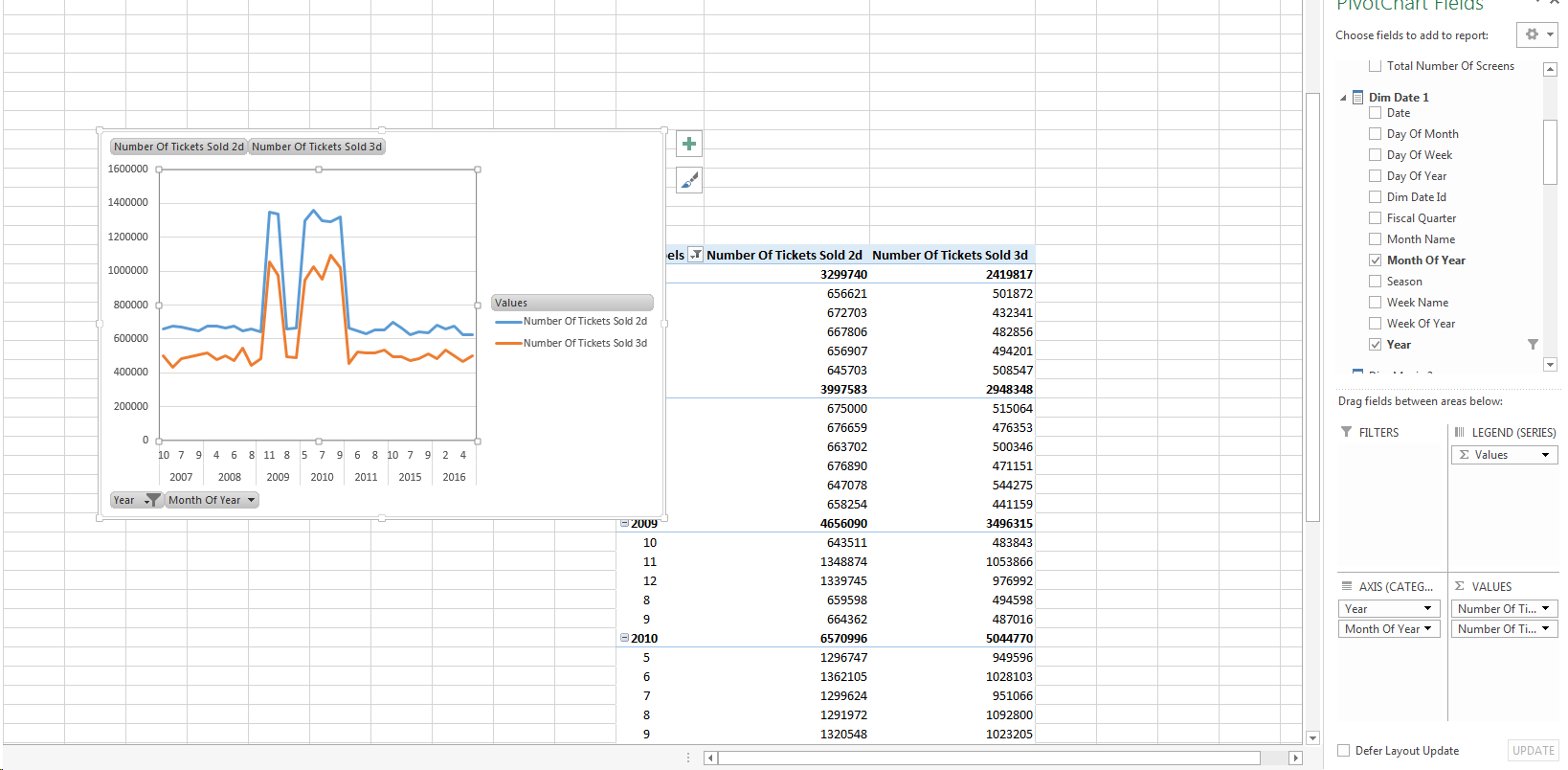




Trends are generated as per the figure below after user selection



User can now change the dimensions or measures and corresponding changes will be reflected in the graph.



This way either by deploying cubes directly to the server or implementing the cubes using pivots would enable the user to generate reports and trends from a particular data mart. This will help users who do not have prior knowledge of SQL servers to go ahead and generate the reports they desire.

# **Glossary of terms**

**A**

**Aggregation**: One way of speeding up query performance. Facts are summed up for selected dimensions from the original fact table. The resulting aggregate table will have fewer rows, thus making queries that can use them go faster.

**Attribute**: Attributes represent a single type of information in a dimension. For example, year is an attribute in the Time dimension.

**C**

**Conformed Dimension**: A dimension that has exactly the same meaning and content when being referred to from different fact tables.

**D**

**Data Cleansing**: The transformation of data in its current state to a pre-defined, standardized format using packaged software or program modules.

**Data Extraction**: The process of pulling data from operational and external data sources in order to prepare the source data for the data warehouse environment.

**Data Integration**: The movement of data between two co-existing systems. The Interfacing of this data may occur once every hour, once a day, etc.

**Data Integrity**: The quality of the data residing in the database objects. The measurement which users consider when analyzing the value and reliability of the data.

**Data Mart**: A data warehouse data class organized for a business functional area or department. The database contains data summarized at multiple levels of granularity and may be designed using relational or multidimensional database structures.

**Data Mart Data Model**: The logical representation of the specific information requirements organized around a department of functional area.

**Data Migration**: The movement of data from one database to another database -- but not necessarily to a working application or subsystem tables.

**Data Model**: A representation of the specific information requirements of a business area.

**Derived Attribute**: A value that is derived by some algorithm from the values of other attributes; for example, profit, which is the difference between revenue and expense.

**Data Mart**: Data marts have the same definition as the data warehouse (see below), but data marts have a more limited audience and/or data content.

**Data Warehouse**: A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process (as defined by Bill Inmon).

**Data Warehousing**: The process of designing, building, and maintaining a data warehouse system.

**Dimension**: A multidimensional structure, which represents a side of a multidimensional cube. Each dimension represents a different category, such as region, time, product type etc.

**Dimensional Model**: A type of data modeling suited for data warehousing. In a dimensional model, there are two types of tables: dimensional tables and fact tables. Dimensional table records information on each dimension, and fact table records all the "fact", or measures.

**Dimension Table**: A table that contains discrete values (usually a countable text field like school or degree). Also, see fact table. Imagine viewing a spreadsheet. The row and column names would be the dimensions and the numeric data within would be the facts.

**Drill Across**: Data analysis across dimensions.

**Drill Down**: Data analysis to a child attribute.

**Drill Through:** Data analysis that goes from an OLAP cube into the relational database.

**Drill Up:** Data analysis to a parent attribute.

**E**

**ETL**: Stands for Extraction, Transformation, and Loading. The movement of data from one area to another.

**Extraction, Transformation and Loading (ETL) Tool**: Software that is used to extract data from a data source like an operational system or data warehouse, modify the data and then load it into a data mart, data warehouse or multi-dimensional data cube.

**F**

**Fact Table**: A type of table in the dimensional model. A fact table typically includes two types of columns: fact columns and foreign keys to the dimensions.

**G**

**Grain**: A term used to describe how finally broken down a fact is in a table. For example, we might have wages individually recorded per employee in one table but we might have another table with wages aggregated by department.

**H**

**Hierarchy**: A hierarchy defines the navigating path for drilling up and drilling down. All attributes in a hierarchy belong to the same dimension.

**M**

**Metadata**: Data about data. For example, the number of tables in the database is a type of metadata.

**Measure**: A quantifiable variable or value stored in a multi-dimensional OLAP cube. It is a value in the cell at the intersection of two or more dimensions.

**Metric**: A measured value. For example, "Total movie ticket sales" is a metric.

**MOLAP**: Multidimensional OLAP. MOLAP systems store data in the multidimensional cubes.

**O**

**OLAP**: On-Line Analytical Processing. OLAP should be designed to provide end users a quick way of slicing and dicing the data.

**S**

**SSIS**: SQL Server Integration Services is a platform for building enterprise-level data integration and data transformations solutions. You use Integration Services to solve complex business problems by copying or downloading files, sending e-mail messages in response to events, updating data warehouses, cleaning and mining data, and managing SQL Server objects and data

**SSAS**: SQL Server Analysis Services is an analytical data engine used in decision support and business analytics, providing the analytical data for business reports and client applications such as Power BI, Excel, Reporting Services reports, and other data visualization tools.

**SSRS**: SQL Server Reporting Services is a solution that customers deploy on their own premises for creating, publishing, and managing reports, then delivering them to the right users in different ways, whether that is viewing them in web browser, on their mobile device, or as an email in their in-box.

**Star Schema**: A common form of dimensional model. In a star schema, a single dimension table represents each dimension.

# **Bibliography**

[1] Business analysis framework for the data warehouse design and architecture of a data warehouse: <http://www.tutorialspoint.com/dwh/dwh_architecture.htm>

[2] Kimball Technical DW/BI System Architecture: <http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/technical-dw-bi-system-architecture/>

[3] Data Warehousing and Knowledge Discovery: 12th International Conference, DaWaK 2010, Bilbao, Spain, August 30 - September 2, 2010, Proceedings:<https://books.google.com/books?id=IBZtCQAAQBAJ&printsec=frontcover#v=onepage&q&f=false>

[4] Data Warehouse Architecture: <http://www.1keydata.com/datawarehousing/data-warehouse-architecture.html>

[5] Concepts of dimensional data modeling: <https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.whse.doc/ids_ddi_350.htm>

[6] The Data Warehouse Toolkit, Third Edition: The Definitive Guide to Dimensional Modeling by Ralph Kimball, Margy Ross. John Wiley & Sons, Jul 1, 2013

[7] The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data by Ralph Kimball, Joe Caserta. John Wiley & Sons, Apr 27, 2011

[8] Business Intelligence: Practices, Technologies, and Management by Rajiv Sabherwal, Irma Becerra-Fernandez. John Wiley & Sons, 2011

[9] SQL Server Technical Documentation -SQL Server Integration Services: <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation>

[10] SQL Server Analysis Services: <https://docs.microsoft.com/en-us/sql/analysis-services/analysis-services>

# **Author of the Report**

|  |  |
| --- | --- |
| **Name** | **UIN** |
| Aditya Dakur Rudraiah | 126003208 |
| Ajay Thomas | 825008104 |
| Apurva Shrivastava | 925009508 |
| Isha Arora | 825008103 |
| Poonam Tare | 825008406 |

# **Date Report Created**

**3rd May, 2017**

# **Contribution of each group member to the project**

**Collective work as a group:**

* Collecting the Movie data and preparing transactional data.
* Finalizing BI questions that the data warehouse will address.
* Drafting and finalizing the dimensional model for the data set.
* Documentation for the respective tasks performed.

**Individual Contributions:**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Staging | Data warehouse | Reporting |
| Aditya Dakur | Creating the data sets and  loading the data from source CSV tables to staging | * ETLTransfomations and loading the dimensions from Staging to data warehouse. * Loading the Facts from staging to data warehouse. | Generated Movie performance reports using SSRS and SSAS |
| Ajay Thomas | Creating the data sets and loading the data from source CSV tables to staging | * ETL Transformations and loading the dimensions from Staging to data warehouse. * Loading the Facts from staging to data warehouse. | Generated Theatre performance reports using SSRS and SSAS. |
| Apurva Shrivastava | Deformalizing the data provided and creating source tables and staging databases | * Loading the dimensions from Staging to data warehouse. * Loading the Facts from staging to data warehouse. | Implementation of SSAS and SSRS reporting. Also created power pivots. |
| Isha Arora | Creating the data sets and  loading the data from source CSV tables to staging | * Loading the dimensions from Staging to data warehouse. * Loading the Facts from staging to data warehouse. | Generated Awards performance reports using SSRS and SSAS |
| Poonam Tare | Creating the data sets and  loading the data from source CSV tables to staging | * Loading the dimensions from Staging to data warehouse. * Loading the Facts from staging to data warehouse. | Generated Production house performance reports using SSRS and SSAS |