Business Intelligence I

Fall Semester 2020/2021
Group 4

Prof. Bruno Jardim
Prof. Miguel de Castro Neto

Final Project

Data Warehouse Solution

By

Rodrigo Chaparro Villamizar - m20200040
Sara Nunes - m20201111
Andrea Sánchez Licer - m20200764

INDEX

I. Introduction

A. Presentation of the organization

This report will propose a Data Warehouse solution for IMDb. IMDb stands for Internet Movie Database and it is an online database for movies, television, video games, and additional streaming content.

IMDb began in October 1990 as a very basic film database created and maintained by its founder Col Needham and it was not until 1996 that it was formalized as a company in the United Kingdom. In 1998 IMDb became a subsidiary of Amazon, with the goal of having Amazon use IMDb as a medium to advertise DVD sales. Nowadays, the database is one of the most popular websites for consulting information not only about films, games, and programs, but also information about actors, producers, directors, and even fictional characters.

According to information published on its website, IMDb receives more than 200 million visitors per month and its database contains information on more than 3 million movies and television programs, in addition to information on more than 6 million actors and production staff. Besides getting information about movies and TV series, users can check the billboard, see information about upcoming releases, watch short movies, watch scenes from TV programs and watch TV channel programming (subject to availability in users' city).

An interesting feature is that users can consult a movie, see how it has been rated by people who have gone to see it and, in addition, see how that rating is broken down by age and gender (male or female) so that you can know which sectors of the public preferred the movie.

There is also a paid version called IMDbPro, designed for people working in the entertainment industry. The services provided by this subscription, available since 2002, include complete filmographies, titles in development, full database access including media contacts, company contact information, daily news from the entertainment industry, job opportunities, resume, among others. For people looking for work in the entertainment industry, IMDb Pro allows them to create a page with their pictures and a demo video.

B. Business Needs

IMDb is a world-famous database that users can access for finding simple information about movies. However, in order to provide useful insights and help Amazon make better advertisement decisions, it is missing some Business Intelligence visualizations and a data warehouse for allowing analysts to get information more easily by being able to extract data directly from the source. In this project, we will focus on the Data Warehouse implementation and design. The movie companies need to focus on analyzing and measuring the Movies' profit, budget scores, and the number of votes, for that reason these will be our facts.

When Data Warehouse will be implemented, the company will be able to analyze the gross margin, the budget, the votes, and the score between each movie with different variables that possibly affect the profitability, like the cast, director, duration, country, ratings, genre, runtime, and others. In addition, we will be able to centralize the information from different sources and be able to analyze it all together in a faster way to determine which are the characteristics of a profitable movie.

In other words, streamlines the flow of the information in the company in order to optimize the resources and take better advantage of opportunities to generate a high Return on Investments and determine what makes a movie profitable.

II. Original Data Sources

For this project, we want to get answers about the movie industry, specifically about the movies' success, and we define success by how well the movie was rated and how much the movie profited. For this, we need to look at the factors that intervene in the movies' success, like actors, user votes, etc. to get answers.

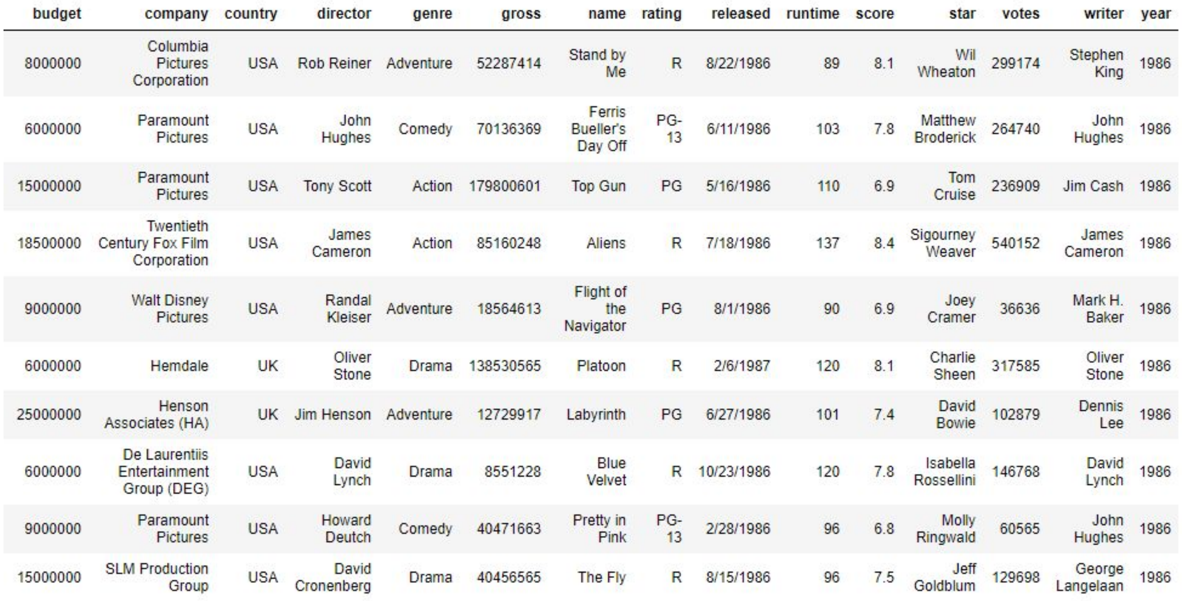
The data chosen for this project was movie data from IMDb which provides information about a list of over five thousand movies (6820). This project uses one data source which was a dataset found on Kaggle, it was updated four years ago, and it contains movies from 1986 until 2016, each year has 220 movies. In total the dataset contains 6820 rows and 15 columns.

This dataset contains 15 variables, approximately half of them are numerical and the rest non-numerical. These variables are described in table 1.

Variable	Description	Data Type
Budget	The budget of a movie in American dollars.	Integer
Company	The production company	String
Country	Country of origin	String
Director	The director	String
Genre	Main genre of the movie	String
Gross	Revenue of the movie in American dollars	Integer
Name	Name of the movie	String
Rating	Rating of the movie (R, PG, etc.)	String
Released	Release date (MM-DD-YYYY)	Date
Runtime	Duration of the movie in minutes	Integer
Score	IMDb average user rating. 1-10	Float
Stars	Number of user votes	String
Votes	Main actor/actress	Integer
Writer	Writer of the movie	String
Year	Year of release	Integer

Table 1: Attributes Description

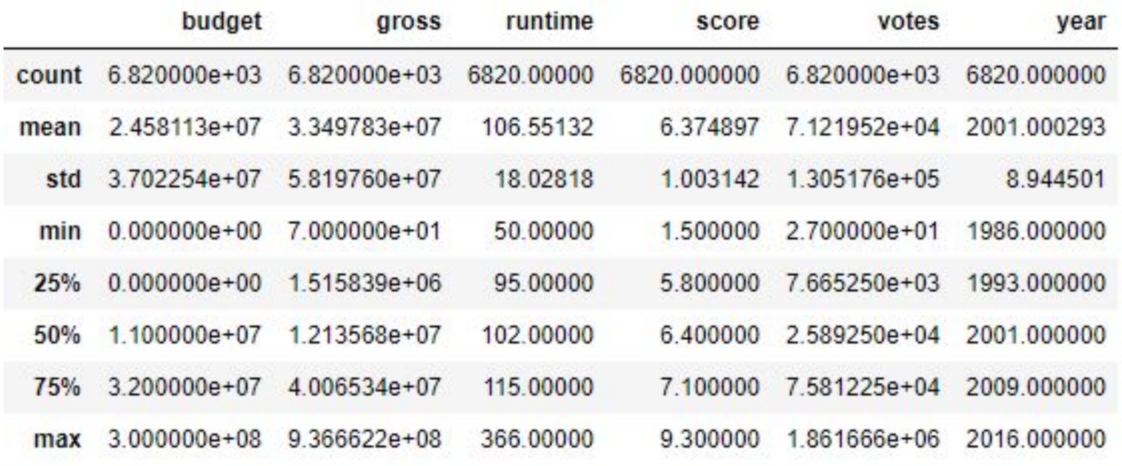
In table 2, we can see what the data (first ten rows) looks like before being transformed.



	budget	company	country	director	genre	gross	name	rating	released	runtime	score	star	votes	writer	year
	8000000	Columbia Pictures Corporation	USA	Rob Reiner	Adventure	52287414	Stand by Me	R	8/22/1986	89	8.1	Wil Wheaton	299174	Stephen King	1986
	6000000	Paramount Pictures	USA	John Hughes	Comedy	70136369	Ferris Bueller's Day Off	PG-13	6/11/1986	103	7.8	Matthew Broderick	264740	John Hughes	1986
	15000000	Paramount Pictures	USA	Tony Scott	Action	179800601	Top Gun	PG	5/16/1986	110	6.9	Tom Cruise	236909	Jim Cash	1986
	18500000	Twentieth Century Fox Film Corporation	USA	James Cameron	Action	85160248	Aliens	R	7/18/1986	137	8.4	Sigourney Weaver	540152	James Cameron	1986
	9000000	Walt Disney Pictures	USA	Randal Kleiser	Adventure	18564613	Flight of the Navigator	PG	8/1/1986	90	6.9	Joey Cramer	36636	Mark H. Baker	1986
	6000000	Hemdale	UK	Oliver Stone	Drama	138530565	Platoon	R	2/6/1987	120	8.1	Charlie Sheen	317585	Oliver Stone	1986
	25000000	Henson Associates (HA)	UK	Jim Henson	Adventure	12729917	Labyrinth	PG	6/27/1986	101	7.4	David Bowie	102879	Dennis Lee	1986
	6000000	De Laurentiis Entertainment Group (DEG)	USA	David Lynch	Drama	8551228	Blue Velvet	R	10/23/1986	120	7.8	Isabella Rossellini	146768	David Lynch	1986
	9000000	Paramount Pictures	USA	Howard Deutch	Comedy	40471663	Pretty in Pink	PG-13	2/28/1986	96	6.8	Molly Ringwald	60565	John Hughes	1986
	15000000	SLM Production Group	USA	David Cronenberg	Drama	40456565	The Fly	R	8/15/1986	96	7.5	Jeff Goldblum	129698	George Langelaan	1986

Table 2: Data Source Sample

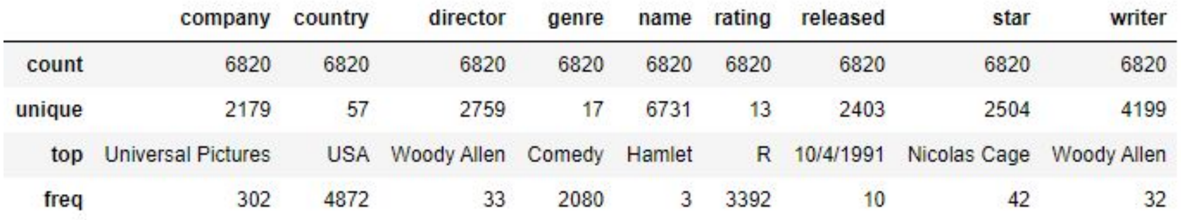
In table 3 is the description of the numerical data. We can see that we have an average budget of 24581129$ and a maximum budget of 300000000$, the standard deviation is 37022536. The average movie gross revenue is 33497828$ and the maximum revenue was 936662225$, the standard deviation is 58197602. The average runtime is 106 minutes, while the minimum is 50 minutes and the maximum 366 minutes, the standard deviation is 18 minutes. The average score of a movie is 6.37, the minimum 1.5 and maximum 9.3 and the standard deviation is 1. The average number of votes is 71219, minimum 27, and maximum 1861666, and the standard deviation is 130517. The average year is 2001 and the oldest movie is from 1986 and the most recent is 2016 and the standard deviation is 8.9.



	budget	gross	runtime	score	votes	year
count	6.820000e+03	6.820000e+03	6820.00000	6820.000000	6.820000e+03	6820.000000
mean	2.458113e+07	3.349783e+07	106.55132	6.374897	7.121952e+04	2001.000293
std	3.702254e+07	5.819760e+07	18.02818	1.003142	1.305176e+05	8.944501
min	0.000000e+00	7.000000e+01	50.00000	1.500000	2.700000e+01	1986.000000
25%	0.000000e+00	1.515839e+06	95.00000	5.800000	7.665250e+03	1993.000000
50%	1.100000e+07	1.213568e+07	102.00000	6.400000	2.589250e+04	2001.000000
75%	3.200000e+07	4.006534e+07	115.00000	7.100000	7.581225e+04	2009.000000
max	3.000000e+08	9.366622e+08	366.00000	9.300000	1.861666e+06	2016.000000

Table 3

In Table 4, there is a description of the non-numerical variables. There are 2179 movie companies and the one that has more movies is Universal Pictures producing 302. The movies were made in 57 different countries and most of the movies (4872) were filmed in the USA. The movies were directed by 2759 directors and the director that produced more movies was Woody Allen. There are 17 different genres, and the most common movie genre is comedy, having 2080 comedy movies, a third of the total movies. There are thirteen different ratings and the most frequent is R, with half of the movies being rated R (3392). There are 2403 unique release dates, and the day that had more movies premiered was 10/04/1991 with 10 movies. There are 2504 different star actors and the one that starred in more movies is Nicolas Cage with 42 movies. There are 4199 different writers and the writer that wrote more movies is Woody Allen with 32 movies.



	company	country	director	genre	name	rating	released	star	writer
count	6820	6820	6820	6820	6820	6820	6820	6820	6820
unique	2179	57	2759	17	6731	13	2403	2504	4199
top	Universal Pictures	USA	Woody Allen	Comedy	Hamlet	R	10/4/1991	Nicolas Cage	Woody Allen
freq	302	4872	33	2080	3	3392	10	42	32

Table 4

III. Staging Area

A. Description and Reasons

There are situations, such as having a high complexity of data, in which at the time of building a data warehouse, it is legitimate to consider the use of a Staging Area (SA). The reason being the advantages that this system provides easing the ETL process.

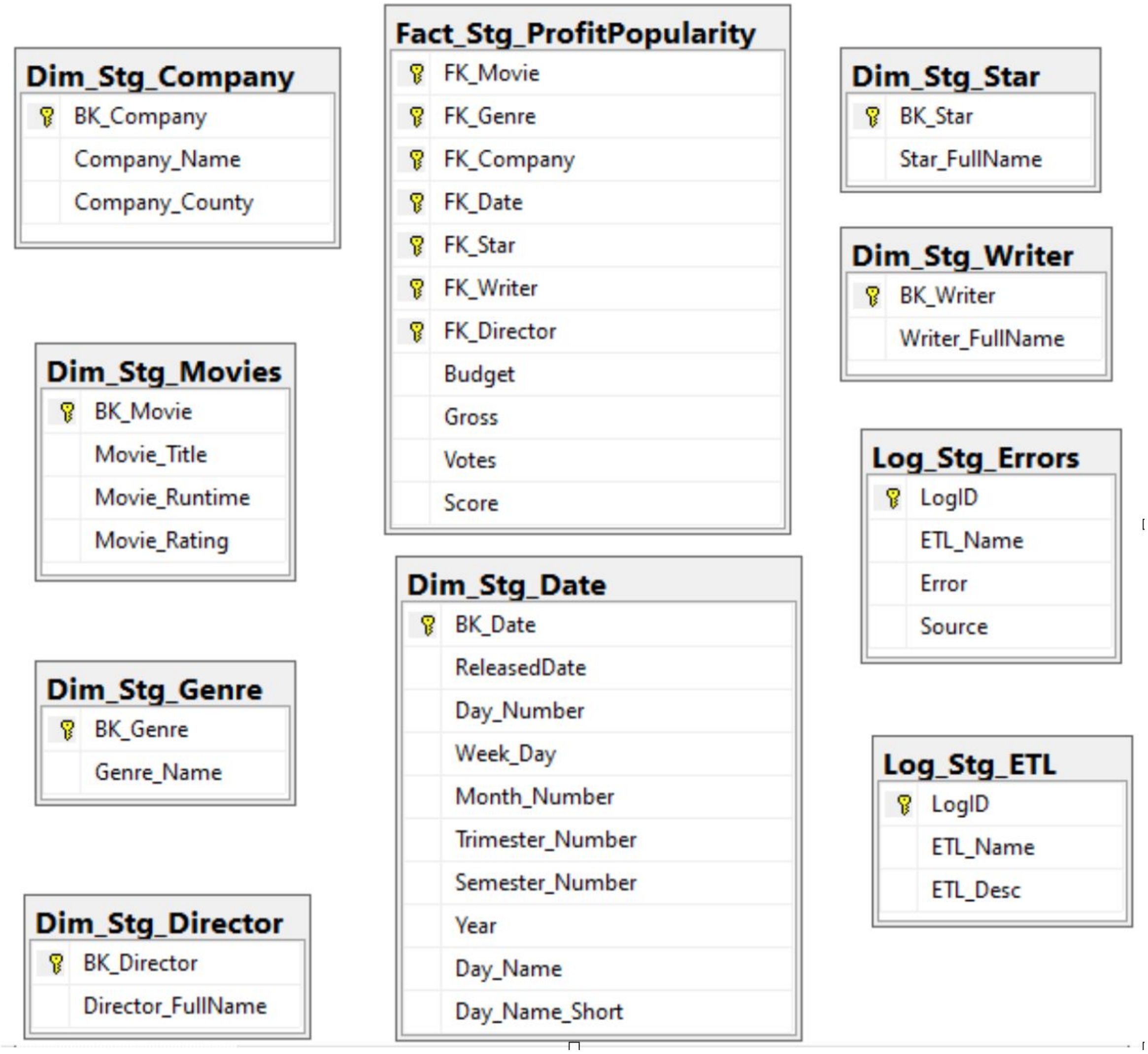
A SA allows the independence of the loading process by blocks or stages. This is very useful and practical when working with lots of data since it avoids having to restart the whole process in case of errors or breakdowns. Moreover, if it is implemented correctly, it makes it possible to restart the different phases of the ETL process independently. This means that if, for example, the transformation process fails, it would be sufficient to repeat this phase, but it would not be necessary to repeat the previous stage: the extraction stage.

Furthermore, the compilation of the different blocks or stages of the staging process can even be adapted to the needs of the clients, although this is always previously contemplated in the general ETL process. Finally, as it is a physically independent disk, in no case does it affect or slow down other processes in the system. The SA allows to do the transformations in a safe environment, without affecting the performance of the datawarehouse.

B. Development of the Staging Area

The data was stored inside Management Studio. Therefore, we accessed the software Visual Studio where SQL Server Integration Services (SSIS) was used to access the data from the database and flat sources, transform it in Visual Studio and loaded into the SA. This is know as ETL process. Please, refer to the ETL section for a more in-depth discussion of the process. Consequently, the SA will contain all the temporary tables created during the extraction process and resulted from the applied transformations.

Its schema is basically a replication of the data warehouse design (p.11), with the same tables and fields, but with the following differences: in the SA there are no Foreign Key (FK) constrains between tables, no relationships between tables, no Surrogate Keys (the table keys are Business Keys (BK)), and no slowly changing dimension fields (SCD). The SA schema can be found below.



Dim_Stg_Company
🔑 BK_Company
Company_Name
Company_County

Fact_Stg_ProfitPopularity
🔑 FK_Movie
🔑 FK_Genre
🔑 FK_Company
🔑 FK_Date
🔑 FK_Star
🔑 FK_Writer
🔑 FK_Director
Budget
Gross
Votes
Score

Dim_Stg_Star
🔑 BK_Star
Star_FullName

Dim_Stg_Writer
🔑 BK_Writer
Writer_FullName

Dim_Stg_Movies
🔑 BK_Movie
Movie_Title
Movie_Runtime
Movie_Rating

Log_Stg_Errors
🔑 LogID
ETL_Name
Error
Source

Dim_Stg_Date
🔑 BK_Date
ReleasedDate
Day_Number
Week_Day
Month_Number
Trimester_Number
Semester_Number
Year
Day_Name
Day_Name_Short

Dim_Stg_Genre
🔑 BK_Genre
Genre_Name

Log_Stg_ETL
🔑 LogID
ETL_Name
ETL_Desc

Dim_Stg_Director
🔑 BK_Director
Director_FullName

Table 5: Staging Area Schema

IV. Data Warehouse

A. Methodology

A data warehouse (DW) according to Inmon (Inmon 02, Imhoff & Galemmo 03), is a data collection oriented to a certain area (company, organization, etc.), integrated, non-volatile, and variable over time, which helps decision-making in the entity in which it is used. On the other hand, Kimball (Kimball 98) defines it as "a copy of the transactional data structured specifically for queries and analysis".

There are many methodologies of design and construction of DW. However, the most used methodologies are Kimball's and Inmon's. In order to understand the major difference between these two methodologies, we must explain in addition to the notion of DW, the Data mart's idea. A Datamart (Kimball et al 98) is a repository of information, similar to a DW, but oriented to an area or department-specific to the organization (e.g., Purchasing, Sales, HR, etc.), as opposed to DW which covers the entire organization, thus, the fundamental difference is its scope.

From the architectural point of view, the biggest difference between the two authors is the sense of the construction of the DW, this is starting with the data marts or ascending (Bottom-up, Kimball), or starting with the whole DW from the beginning or descending (Top-Down, Inmon).

In addition, Inmon's methodology is based on well-known concepts of relational database design (Inmon 02, Imhoff & Galemmo 03); the methodology for building such a system is the usual one for building an information system, using the usual tools, unlike Kimball's, which is based on a dimensional (non-standardized) modeling (Kimball et al 98, 08).

We believe that the methodology most in line with our company in question is that of Kimball, as it provides a minor to major, very versatile approach, and a series of practical tools that help in the implementation of a DW. Therefore, we will detail this methodology in the remainder of this paper.

Kimball’s Methodology

As previously mentioned, Kimball’s philosophy focuses on the fact that, in most organizations, the construction of a data warehouse originates from the interest and effort of a department. That is why in his first version this data warehouse is no more than a departmental data mart.

As other departments need their own data marts, these will be combined with the first one maintaining a standardization methodology through what Kimball calls "conformed dimensions", which will be the common dimensions among the different departments. The key is that these dimensions have to be shared by the different data marts that exist in the organization, thus ensuring the integrity of them and giving rise to the conglomerate of structures that for Kimball make up the data warehouse.

The main advantage of this data warehouse approach is that being made up of small data marts structured in dimensional data models (star or snowflake schemes), specially designed for querying and reporting, the entire data warehouse can be directly exploited by the reporting and data analysis tools without the need for intermediate structures.

Building a DW solution is extremely complex, and Kimball proposes a methodology that helps us to simplify that complexity. The tasks of this methodology (life cycle) are shown in figure 1.

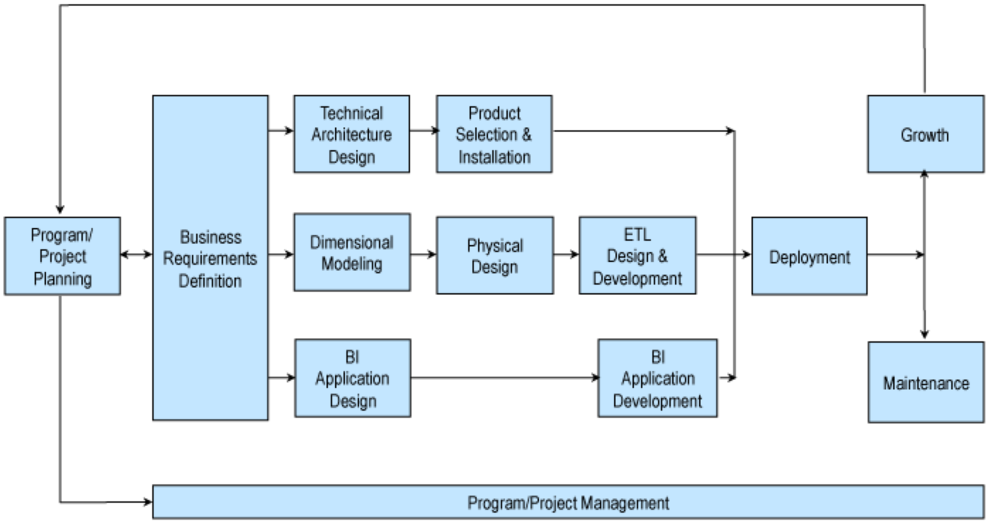

Program/ Project Planning
Business Requirements Definition
Technical Architecture Design
Product Selection & Installation
Dimensional Modeling
Physical Design
ETL Design & Development
BI Application Design
BI Application Development
Deployment
Growth
Maintenance
Program/Project Management

Table 6: Tasks of the Kimball Lifecycle methodology (Kimball et al 98)

There are four key decisions during the design of a dimensional model:

1. Select the business process.

2. Declare the grain.

3. Identify the dimensions.

4. Identify the facts

These steps will be explained and applied in the following sections.

B. Dimensional Model Development

Each data point from a data set can be separated into dimensions and facts. Dimensions are the attributes that answer the question “what?” while facts are the measures that answer “how much?”. The latter are the reporting units to be manipulated, measured, and reported for analysis. The dimensions give extra information regarding the facts and make it possible to drill down into the measured entities.

The technique of restructuring the data set and grouping the data points into logical tables on the basis of facts and dimensions is called dimension modeling. The fact table includes the keys from the dimension tables and the other measurable entities. The dimension tables each contain a surrogate key to identify and link to the fact table, followed by the attributes from the raw data set.

Dimensional Model Diagram

The final diagram with its corresponding dimensions and fact table was achieved selecting a star schema. This schema consists of a central fact table, and it is linked directly to each of the dimension tables via a one-to-one relationship. This schema was selected thinking of the business needs, since it is easily expansible, answering future needs quickly. Moreover, the complexity and the number of physical connections are reduced, being more understandable even for non-technical users. Finally, the dimension “crew”, as it was first created, was in the end divided into three separated dimensions: writer, director, and star. This will allow the reduction of complexity when analyzing the data.

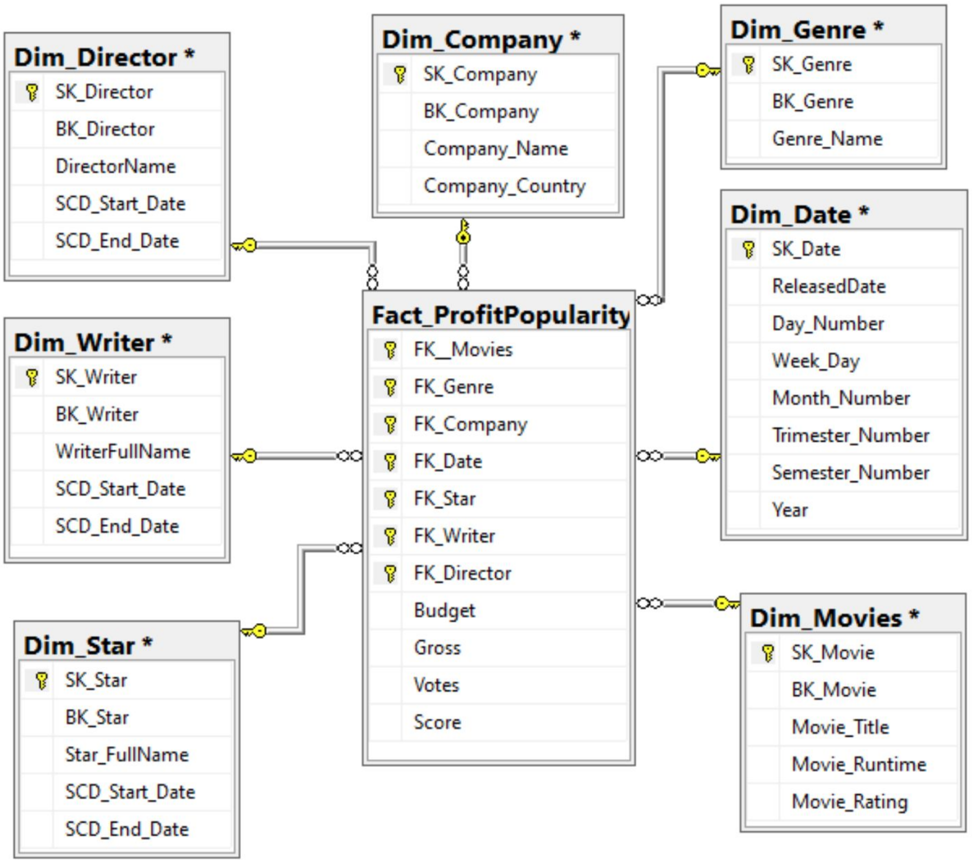


Table 7: Data Warehouse Star Schema

Dimensions

The dimensions arise naturally from the team discussions, and are facilitated by the choice of the level of granularity and the dimensional matrix. Dimension tables have a set of attributes that provide a perspective or form of analysis of a measure in a fact table. Seven dimensions were defined: movies, genre, company, date, star, writer and director. These dimensions were selected because they are considered relevant variables to be associated with the measures that will help finding the solution to the business needs.

Dim_Movies

In this dimension, it is included a surrogate and business key to link it with the fact table, the title of the movies as nvarchar, the time that the movie runs as an integer, and the movie rating as float.

Dim_Genre

In the genre dimension table, a surrogate key, a business key, and the genre are included.

Dim_Company

The company dimension table has a surrogate and business key and also another two values that are the company name and the country of the company.

Dim_Date

In the date dimension, five int data type values are included: number of days, month, trimester, semester, and year. The release date is shown as date type. Finally, we have three nvarchar data types: the day of the week, the name of the day, and the shortened name of the day.

Dim_Star

In the Star Dimension, two int type values can be observed: the surrogate and the business key. This table also contains slowly changing dimensions of the Start Date and End Date dates for recording when the count of movies per star increases.

Dim_Writer

The writer dimension contains the surrogate and business keys as well as Writers' full names. It contains Start Date and End Date dates for recording when the count of movies per writer increases.

Dim_Director

In the Director dimension, there are two int types for each key, there is also one *nvarchar* for the name of the Director. Lastly, there is a Start and End Date for recording when the count of movies per director increases.

Slow-Changing-Dimensions

As slow-changing-dimensions we will include the star, writer, and director dimension, monitoring closely the following attributes respectively: movies participated, written movies, and directed movies. These were the only attributes that can possibly have changed in the future since the data warehouse will expand and have more movies, the actors, writers, and directors can participate in more. For the other dimensions, there should be no changes since those attributes are static.

Hierarchies

Hierarchical dimensions are those dimensions which have a parent-child relationship. In this particular case, a date hierarchy can be identified in the corresponding date dimension table. There are seven levels or hierarchies consisting of release date, year, semester, trimester, month, day, and day of the week.

Fact

This fact table contains all the respective foreign keys of each dimension and includes the measures needed to be analyzed to determine the popularity and the profit, which include the gross profit of the movies, the budget, the votes, and the score.

Granularity

The grain establishes exactly what a single fact table row represents. Granularity means specifying the level of detail. The general suggestion is to start designing the DW at the highest possible level of detail since groupings could then be made at the desired level. In this particular case, the grain declaration consists of:

- Gross margin: the level of detail combines the margin per company, per country, per genre per movie.

- Budget: the level of detail combines the budget per company, per country, per genre per movie.
- Votes: The level of detail combines the votes per company, per country, per genre per movie.
- Score: The level of detail combines the votes per company, per country, per genre per movie.

Since all the facts have the same granularity, we have all in the same fact table which name is Popularity Profit.

V. ETL Processes

The Extraction, Transformation, and Loading (ETL) system is the base on which the data warehouse is nourished. If the ETL system is properly designed, it can extract the data from the data source systems, apply different rules to increase the quality and consistency of the data, consolidate the information from different systems, and finally load the information in the DW in a format suitable for use by the analysis tools.

The design of an ETL process generally consists of six tasks (Trujillo 2003):

- Selecting data for extraction by defining the source data, which usually come from various heterogeneous sources.
- Transforming the sources: Once the data has been extracted from the data sources, it can be transformed or derived. Some of the most common tasks in this step are data filtering, code conversion, calculations of derivative securities, the transformation between different data formats, generation of automatic sequential numbering (derivative keys), etc.
- Join the sources: The different sources can be joined to be loaded into the warehouse as a single source.
- Select the destination for loading: The destination or destinations are selected for loading later the data.
- Join the attributes of the data sources with the attributes of the destination: the attributes (fields) that were obtained from the data sources can be mapped with the corresponding destinations.
- Load the data into the warehouse.

For the design of the ETL process, inside Visual Studio software, the SSIS package was used due to its simplicity as it includes all the tasks related with the process work for the staging area. It also provides drag-and-drop development with components called tasks to build an ETL process.
As previously explained, an intermediate storage area (Staging Area) used for data processing during the ETL process has been used.

A. Extraction Phase

The processes corresponding to the extraction phase focus purely on extracting the source data and store them in tables in the staging area, checking only that you are able to read the data in a suitable format. All developed extraction processes follow exactly the same pattern. A full load approach has been followed, first deleting all the data in each dimension to then load it again making use of two separated, interrelated containers for both steps.

In the SSIS package called “DB to SA”, we start creating a container for deleting each one of the tables in our SA as part of the above-mentioned process. The deleting and loading tasks are organised respectively in two interralated containers.

For the Fact table, instead of a deleting command, a truncate command is used. The difference between this two commands is that, once data is deleted in a dimension and new values are added, the values of the new keys will start from the beginning (counting from 1, generally). On the other hand, when a table is truncated and new values are added, the key values will be generated from the last deleted key value. That means that “Delete” resets the identity (Primary Key) back to the seed (first value), while “Truncate” does not reset the identity (Primary Key).

The data loading was first performed from a flat-file (.csv). Several problems were encountered in this process. At the beginning, certain columns were giving errors due to the use of special characters. Therefore, some data cleaning was carried out in the original file. As loading the data from a flat-file was giving a lot of problems, it was decided to use a database for the majority of the dimensions to load the data into the Staging Area.

Finally, different sources have been used for each dimension:

- For Movies, as the excel file was giving problems, two flat files were created and inserted into the database. From the database, specific data coming from movies and from ratings were selected, merging both flat-files. The specific variables from each source were rating, runtime, movie ID, and title.
- For the Date, the loading was done directly from an excel file and a “conditional split” was used.
- For the Fact table, two flat sources were used.

- The rest of the dimensions were loaded directly from the database using the task OLE DB Source and connecting each task with the database.

Once all the sources tasks are created, it is needed to create the destinations selecting and connecting each of them to the Staging Area.

A few warnings were shown in dimensions Genre, Writer, Star and Date, corresponding to a length warning for writer, star, name day and genre names.

B. Transformation Phase

In the transformation phase, the designed processes take the information saved in the staging tables and apply the necessary transformations to align the structure of the input data with that required by the logical and physical design raised. This is where the data cleaning and checking operations are also carried out of error conditions in order to obtain output data with the specified formats and with the least number of possible inconsistencies. Records with errors are saved in a specific table in the staging area, which will record them to investigate them at any time.

The Movies dimension required the use of "sort" task to select the key and order it to merge the two sources used. Then a data conversion was completed, as it used a flat source from where everything comes in text type.

Moreover, our data set contains information since 2016, so a conditional split has been applied to the Date dimensions in order to use data only from that year onwards.

For the dimensions Director, Writer, and Star the first name and surname are concatenated into Full Name.

In the fact table a "sort" task was also used to merge the two sources. Another function called "multicast" was added in order to count the number of rows that the source is recibing to load them in the log table created in the SA. After this, several data flow tasks are implemented to log the ETL, one to start and another to log the deleting process. Another two tasks are added to "log the table load" and "log the number of data in the popularity/profit fact table load", and a last one to load the errors was created in event handler.

Finally, all the tables are checked and the data is loeaded correctly in the SA tables.

C. Loading Phase

In the loading phase, data from staging tables are processed to be loaded into the star model in the data warehouse. This phase comprises the loading of dimensions and processing of measurement tables, changing natural keys by substitute keys by searching the dimension tables.

The Foreign Keys for the fact table are coming from the dimension tables without existing in the fact table. Therefore, it is need to make them as primary keys in the fact table, making sure all the foreign keys of the connecting dimensions are available.

For the clearing the data sequence, we first implemented the truncation of the fact table as it just contains foreign keys and, only then, the dimensions are deleted. However, this sequence was discarded since the incremental was later implemented, which means that new data will no longer be deleting the old data, but it will get just the new rows from the Staging Area. This incremental load is beneficial because it will speed up the process, as usually the data loads are performed overnight with the expectation of completing them before users can see the data the next day, which may not be enough time for the full load to be completed. In addition, historical data will not be lost.

Then, the loading of the dimensions was performed first, due to the need of having surrogate keys to fill the foreign keys in the fact table, and, afterwards, the fact table was loaded.

For the incremental load, it is needed to ensure that all the new information is going to be new information and not repeated one. This step is done in the SA as wells as in the DW.

In the SA, we must guarantee that the fact table will be loaded incrementally, and, then, in the DW we must guarantee that the dimensions tables are going to be loaded incrementally.

The slowly changing dimensions which include Writer, Star and Director, are already incrementally loaded. In order to define them, it is needed to use the slowly-changing-dimension-wizard task to choose what type of change it is. In the case in question, as the type is 2, an historical attribute name is added since we want to keep track of the old name of the person, i.e someone marries and the lastname changes. The row that was changed will keep the same Business Key as the old row to be able to identify that they are the same records, and it will have a new Surrogate Key as

well as a start and end date to know when this change occurred and to identify if it is the old or new row.
For the non-slowly changing dimensions, slowly changing tasks need to be added and define all the columns as fixed attributes to make sure they are loaded incrementally. By doing this, if we try to insert a repeated value, it will either throw an error or the value will not be changed because it is a fixed attribute, meaning it cannot be changed. If we try to insert new data that does not exist, it will work because it will not violate any constraints.

In the SA, we need to make sure the fact table is loaded incrementally, this is done by creating a variable that stores the date of the most recent loaded row in the DW. This is done by looking up (join) to search what date corresponds to the most recent date, and that value is assign to the variable. This is necessary because if we have the most recent value, we can compare the rows we want to load, and if the rows we want to load are older than the one in the DW, it will mean they are repeated. If the rows we want to load are more recent, it is new data and has to be be loaded into the DW.

To load the fact table in the DW, lookups for all the dimensions need to be performed, where the foreign keys are being retrieved from the surrogate keys values. Looking at the business keys, the surrogate key needs to be retrieved. For the measures, the values from the SA fact table are used. In the dimension, there are surrogate keys and business keys. We need to look at the value of the business keys in the dimensions and look at the value of the business keys as well in the SA fact table. Then, the surrogate key from the dimension that corresponds to that business key is used in the DW fact table.

A few errors were obtained from the dimensions because the surrogate key was missing. This problem was solved adding “identity (1,1)” in the script surrogate key line for each dimension exept Date dimension.

Finally, every step is checked and correctly loaded into the data warehouse solution.

VI. Conclusions and Lessons Learned

The Ralph Kimball methodology was used, since data regarding the benefits and popularity of each film can be extracted effectively and in a simplified way, obtaining successful results at the time of data analysis.

When designing a data warehouse solution, it has been learned the relevance of having a complete understanding of the data source before the complete design of the solution to know what the system is storing and the importance of the data to the business questions. As when we define our data model or when we identify the data origins, it is advisable to have within the ETL system a well-defined nomenclature that allows a better management and understanding of the whole system.

In the beginning of the ETL implementation, a lot of problems occurred and changes were made to solve them. It became clear the importance of modelling the data in the appropriate way and taking the time to design the Data Warehouse. This is going to affect a lot the next stages and the Data Warehouse only serves the purposes if it is what the client/company needs for their business setting. There is no “1 size fits all” solution, each problem is very specific.

A few problems occured in the implementation of the Data Warehouse, most of those problems were because of data types and sizes of the variables of the source files were not matching with the Source table. Additional problems regarding the DW design occurred, for example, missing connections or wrong Foreign Keys.

The result of the project is very satisfactory, all the group members learned a lot about Data Modelling and Data Warehousing as well as regarding the SSIS and the SQL Server Management Studio tools. We managed to work very efficiently and divide the tasks fairly. The project solution is working without errors and the implementation was successful.

References

Imhoff & Galemmo, *Mastering Data Warehouse Design: Relational and Dimensional Techniques*, Wiley Publishing, 2003. [Accessed 14.12.2020]

Inmon, *Building the Data Warehouse*, (Third Edition). John Wiley & Sons, 2002. [Accessed 15.12.2020]

Kimball et al., *The Data Warehouse Lifecycle Toolkit*. New York, Wiley Publishing, 1998. [Accessed 14.12.2020]

Kimball et al., *The Data Warehouse Lifecycle Toolkit*. 2nd Edition. New York, Wiley Publishing, 2008. [Accessed 15.12.2020]

Trujillo, J., Luján-Mora, S., *A UML Based Approach for Modeling ETL Processes in Data Warehouses*. Alicante, Spain. 2003. [Accessed 27.12.2020]

Withee, K., *Microsoft Business Intelligence for Dummies.* Hoboken, Wiley Publishing, 2010. [Accessed 9.11.2020]