****

FACULTY OF ENGINEERING

SCHOOL OF COMPUTING

SEMESTER 1/20202021

**SCSP3213: Business Intelligence**

**LECTURER:** Dr. Noorfa Hazlinna Mustafa

**SECTION:** 02

**TITLE:** AA Project (PlayStore Applications)

**Name:** Aiman Shaquan Bin Mohd Yusof

**Matric No:** A18CS0021

Table of Contents

[Introduction 1](#_Toc63192444)

[Introduction 1](#_Toc63192445)

[Background of Problem 1](#_Toc63192446)

[Objective 2](#_Toc63192447)

[Scope 2](#_Toc63192448)

[Information of Data source 3](#_Toc63192449)

[Methodology and BI tool 4](#_Toc63192450)

[ETL Process - Talend 4](#_Toc63192451)

[Data Warehouse 4](#_Toc63192452)

[Data Visualization – PowerBI 5](#_Toc63192453)

[Data Warehouse 6](#_Toc63192454)

[Transformation table 6](#_Toc63192455)

[Components 7](#_Toc63192456)

[Data Visualization 16](#_Toc63192457)

[Reports 16](#_Toc63192458)

[Charts 18](#_Toc63192459)

[Dashboards 25](#_Toc63192460)

[Conclusion 26](#_Toc63192461)

[References 27](#_Toc63192462)

# Introduction

## I**ntroduction**

Nowadays, the usage of applications through mobile phone has been the new normal in our life. For companies such as Google that provides PlayStore or Apple that provides AppStore as a platform for all types of applications, it is quite important for them to monitor the applications that had been brought in the platform. Most of the platforms are using the user’s search history and interactions on the platform as significance for their recommendation system. But it is also quite beneficial for them to further review and study a better recommendation algorithm using data such as type, genre, or category by ratings and total installs of the applications. Thus, further preprocessing and visualizations are needed to enhance this goal.

## Background of Problem

It is quite challenging to advocate for a platform that respond and collect every single data without quality data and evidence. Precise, reliable and integrity data would not only help in the analysis, but also in planning, preparations and prospective problems. Without a quality data, it is impossible for any case studies to stimulate any analysis and insights to the top managements. They would have a harder time interpreting improtant data such as ratings, reviews and total installs to lend them insights to their own platform. This is also because the data could be uncleaned, inconsistent, inaccurate and incomplete.  
 For this project, we used Google Play Store datasets containing data about the applications in the platform and ratings of the applications. This data consists of thousands of user’s ratings in different applications including their sentiments based on their comment. The data set is clearly are not at the high quality and cleaned because of the inconsistency, duplicates and noises in the data. ETL architecture would help in dealing with the data transformation including the preprocessing in data cleaning using Talend for further analysis and visualisation using PowerBI.

## Objective

The objectives of this case study are:

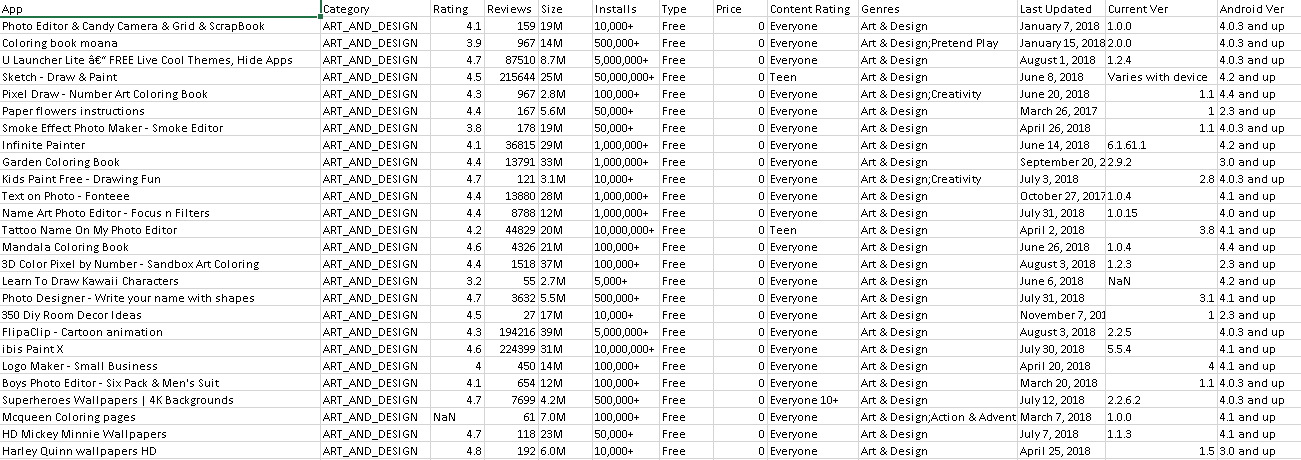
1. To propose and implement a fitting Extract, Transform, and Load (ETL) architecture, and Data Warehousing process for the datasets
2. To implement a suitable architecture of data warehouse design
3. To do further analysis on the applications and its ratings and details in the Google PlayStore

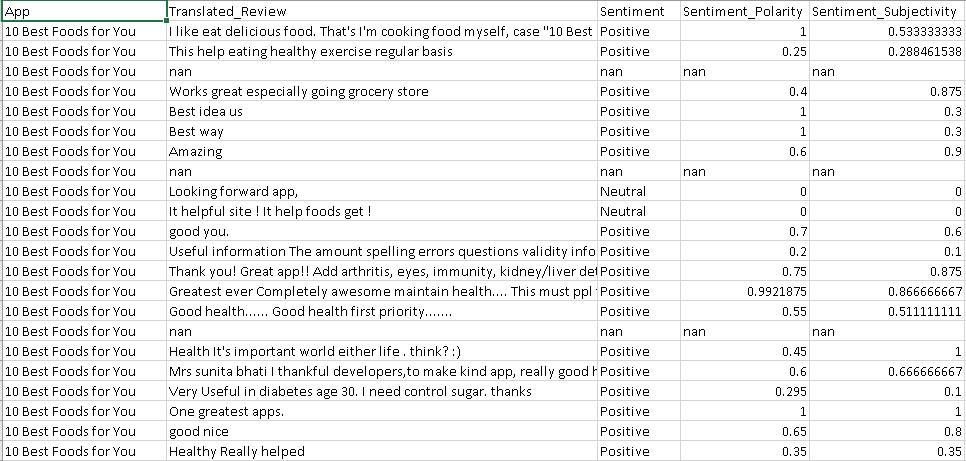
## Scope

During revising this case study, Google PlayStore applications is chosen for the purpose in implementing our knowledge of ETL architecture and designing the processes for the data warehousing also doing an analysis on the based on the case study. Appropriate data warehouse schema and ETL methods using Talend, and analysis and visualizations using PowerBI would help in ensuring smoothness of the work and in enhancing the product developed of the case study.

## Information of Data source

The datasets that are used for this case study contains two .csv files: googleplaystore.csv that contains the applications data and googleplaystore\_user\_reviews.csv that contains the reviews of every users on the applications.

  
Figure 1.0: googleplaystore.csv Data

  
Figure 1.1: googleplaystore\_user\_review.csv Data

As we can see there’s a lot of data cleaning that’s needed to do, from inconsistency of the data, reliability of the data, to data duplicates and null values. Thus, showing the importance of implementing a proper data preprocessing or ETL architecture.

# Methodology and BI tool

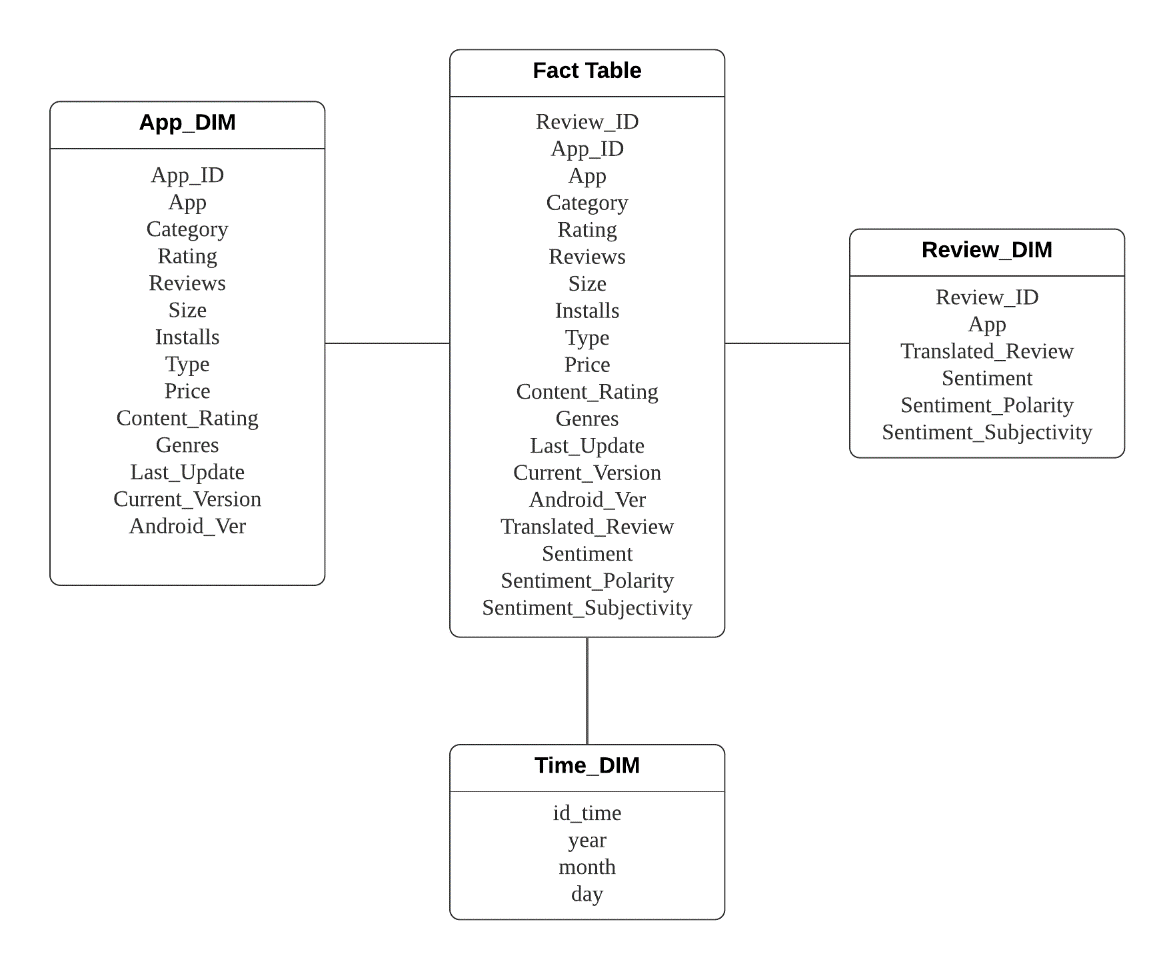
## ETL Process - Talend

Talend is used for the Google PlayStore apps and ratings datasets that are going through the process of cleaning and filtering. Many of the columns and rows are transformed in order to make a reliable and consistent data. The process in ETL architectures are as follows:

* Filtering is used to filter out unneeded rows and columns
* Removing the incomplete rows
* Removing or replacing null values
* Parsing values into real reliable values

## Data Warehouse

Talend is also used for the joining of the data warehouse schema. It would take several columns from application’s dimension, review’s dimension and time dimension as a star scheme.

  
Figure 2.0: Star scheme

Talend is used in joining the the transformed data using tMap component. This is also used to filter out unneeded columns, and parsing data. The final fact table is connected to the database mySQL as database connection.

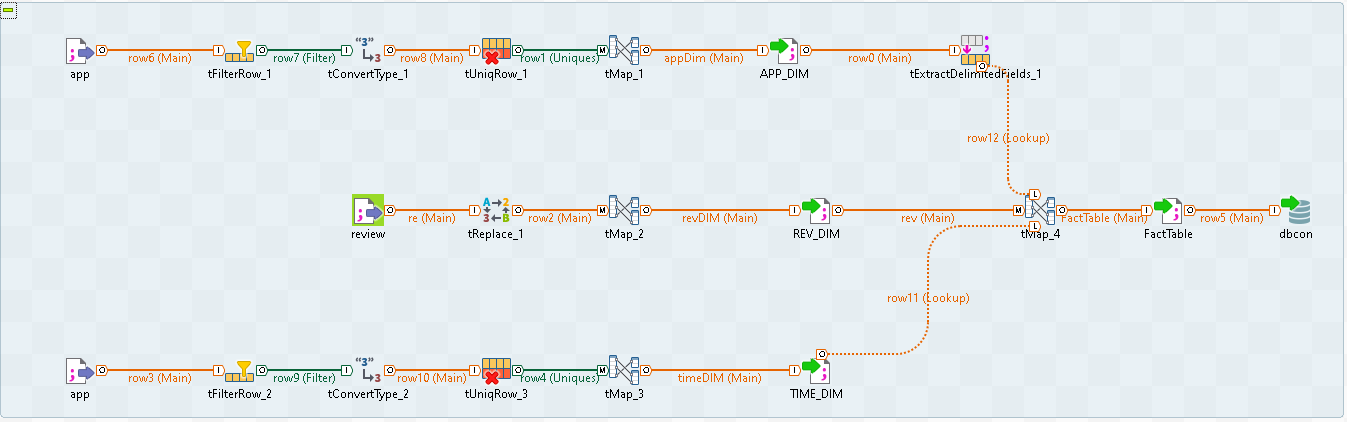
## Data Visualization – PowerBI

PowerBI is used to generate appropriate visualizations for the data from the data warehouse after the ETL process. The variations of charts and the ability to handle thousands of rows simultaneously would smoothen the case studies and helps in giving more insights. Interactable charts and dashboard representing the data warehouse for further analysis going through the process as follows:

* Loading the data warehouse or fact table that has been transformed from Talend before hand
* Design multiple charts based on ratings of the users on the applications by different attributes
* Design multiple charts based on total installs on the applications by different attributes
* Create connections between the charts in the dashboard for more insights
* Use filters and slicers as interactives for the dashboard
* Publish the project into the PowerBI desktop workspace

# Data Warehouse

In creating the data warehouse design, googleplaystore.csv and googleplaystore\_user\_review.csv source file is imported as metadata as app and review respectively. This is to ensure the implementation of the components for transformation to be done easily.

  
Figure 3.0: Data Warehouse Design implementation

## Transformation table

|  |  |  |
| --- | --- | --- |
| **Data source** | **Columns** | **Transformations** |
| googleplaystore.csv | Android\_Ver | Removing rows that are null |
| Reviews | Change the value into integers |
| App | Generate app id as a new column and remove every other symbol in its names |
| Installs | Remove “+” signs from the string and parse it into integer value |
| Price | Remove the dollar signs from the price |
| Last\_Updated | Parse it into a date format and pass to time dimension |
| googleplaystore\_user\_reviews.csv | Translated\_review | Replace all “nan” comments as real null and remove all the symbols |
| App | Remove every other symbol in its names |

## Components

In order to achieve the data warehouse design, various types of components are needed to implement the transformation that’s needed for the columns and rows in each data source.

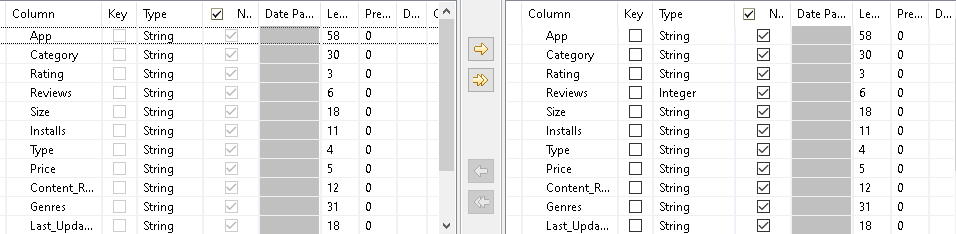
For the APP\_DIM a lot of the cleaning and filtering took place. The components that are used in the making of the APP\_DIM as below:

* tFilterRow component is used to filter out the null value that’s found in the data app file input.

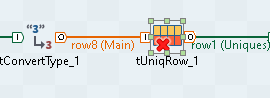
  
Figure 3.1.1: tFilterRow implementation on app metadata  
  
Figure 3.1.2: tFilterRow component

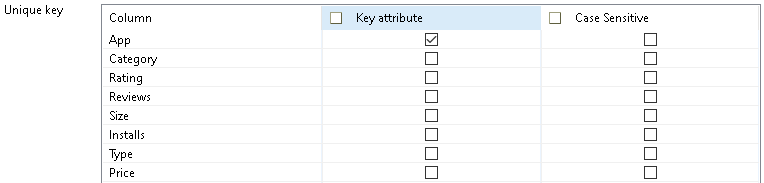
* tConvertType is used to convert the String data type of column Review in App metadata to real integer data type.

  
Figure 3.2.1: tConvertType implementation

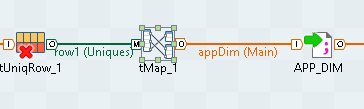
  
Figure 3.2.2: tConvertType schema

* tUniqRow is used to filter out duplicates for the column App.

  
Figure 3.3.1: tUniqRow implementation

  
Figure 3.3.2: tUniqRow component

* tMap is used in making the complete transformation in the making of App dimension implementing the data warehouse schema.

  
Figure 3.4.1: tMap implementation for APP\_DIM

* + Generating sequence id for each unique app using expression Numeric.sequence into a new App\_ID column

  
Figure 3.4.2: tMap component generating sequence as id for column App\_ID

* + Removing all symbols from applications’ name using expression replaceAll and pattern

  
Figure 3.4.3: tMap component and expression in removing all symbols from column App

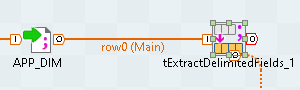
* + Removing “+” symbol from column Installs to make it available to be converted into integer using expression substring

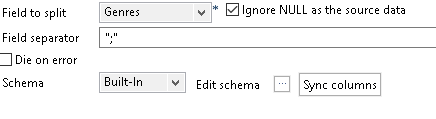
  
Figure 3.4.4: tMap component and expression removing the “+” symbols from column Installs

* + Removing “$” sign from column Price to make it available to be converted into double using expression replaceAll and pattern

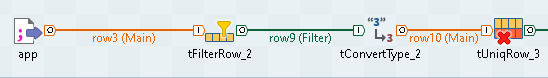
  
Figure 3.4.5: tMap component and expression removing the “$” sign from column Price

* tExtractDelimitedFields is used to extract extra delimited “;” from column genre to ensure the consitency of the data.

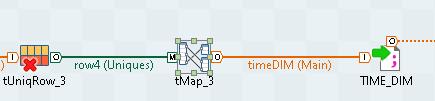
  
Figure 3.5.1: tExtractDelimitedFields implementation

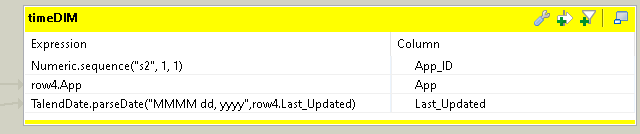
  
Figure 3.5.2: tExtractDelimitedFields components

For TIME\_DIM following the data warehouse schema, the components that is used is the same as APP\_DIM as it uses the same app metadata input. The component that are used in the making of TIME\_DIM as below:

  
Figure 3.6.1: Filtering components

* tMap is used to generating app id the same as App\_ID in APP\_DIM and parsing real date data type from string in column Last\_Updated by using expression parseDate and patterns.

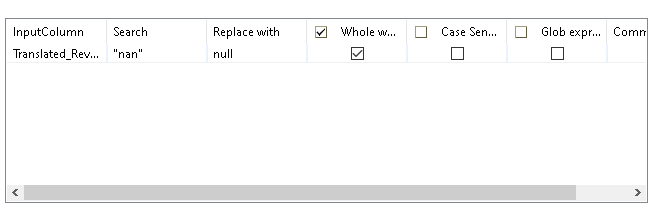
  
Figure 3.6.2: tMap implementation for TIME\_DIM

  
Figure 3.6.3: tMap component for generating id and parsing dates

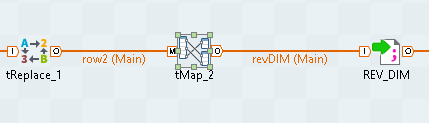
For REV\_DIM, it uses googleplaystore\_user\_review.csv as an input metadata called review. Different way of cleaning the data needed different components for the data source. The components that are used as below:

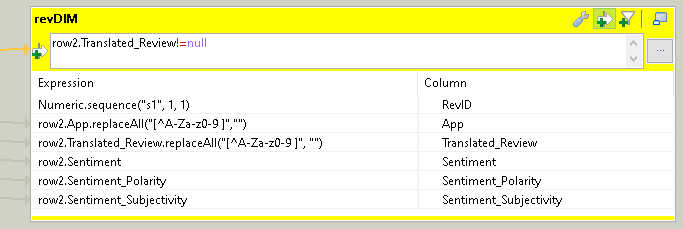
* tReplace is used to remove string “nan” from column Translated\_Review and replace it with real null value.

  
Figure 3.7.1: tReplace implementation

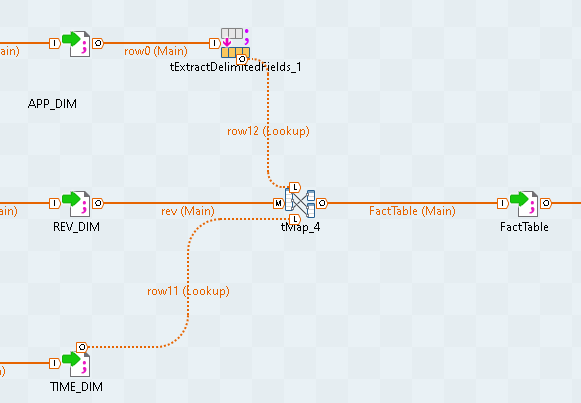
  
Figure 3.7.2: tReplace component

* tMap is implemented to filter out rows with null value of column Translated\_Review, generate id for each Reviews into a new column RevID, removing symbols from column App, and removing all symbols from column Translated\_Reviews

  
Figure 3.8.1: tMap implementation for REV\_DIM

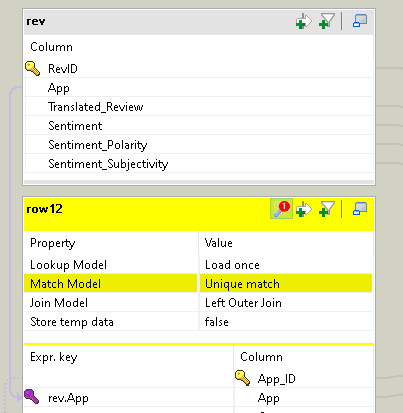
  
Figure 3.8.2: tMap component in completing the REV\_DIM

In the making of FactTable or completing the Data Warehouse design, component tMap is also used in joining each Dimensions, parsing data type from column Sentiment\_Polarity, Sentiment\_Subjectivity, Rating, Price, and Installs into doubles and integers. It uses REV\_DIM as the main and APP\_DIM and TIME\_DIM as a look up.

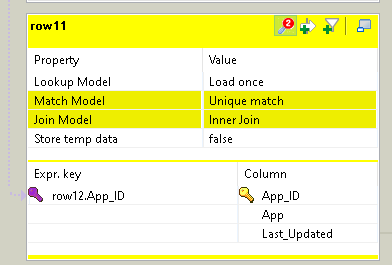
  
Figure 3.9: tMap implementation on joining and cleaning in making the complete Data Warehouse

The process that took place in making a full joining, cleaning and reliable values from the data is as below:

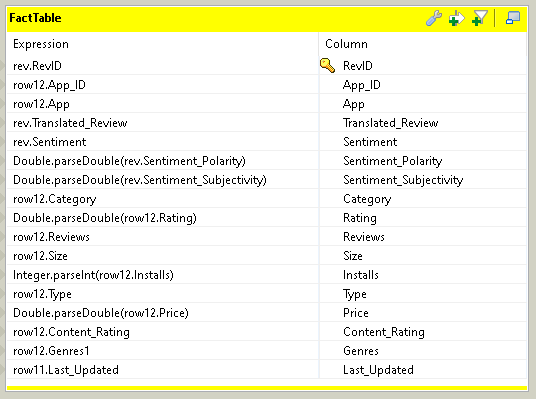
* + Left outer join for REV\_DIM and APP\_DIM using column App

  
Figure 3.9.1: Left outer join implementation on REV\_DIM and APP\_DIM

* + Inner join for APP\_DIM and TIME\_DIM using column App\_ID

  
Figure 3.9.2: Inner join implementation on APP\_DIM and TIME\_DIM

* + Implementation of parsing real data type into column Sentiment\_Polarity, Sentiment\_Subjectivity, Rating, Price, and Installs into doubles and integers and merging every Dimension tables completing the Data Warehouse schema.

  
Figure 3.9.3: Implementation of parsing the columns into real data type and merging every tables

* tDbOutput is used as Database connection from the data warehouse completing the ETL architecture process.

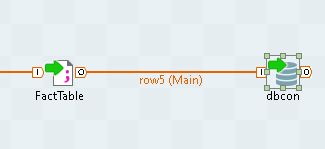
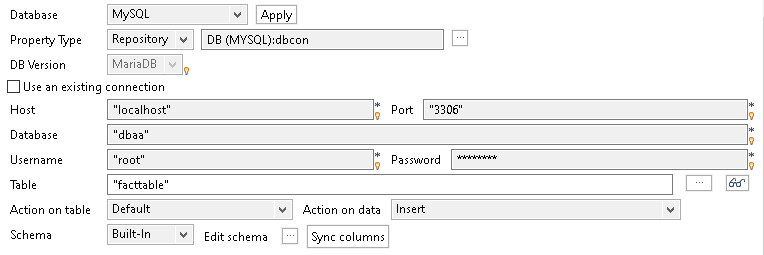
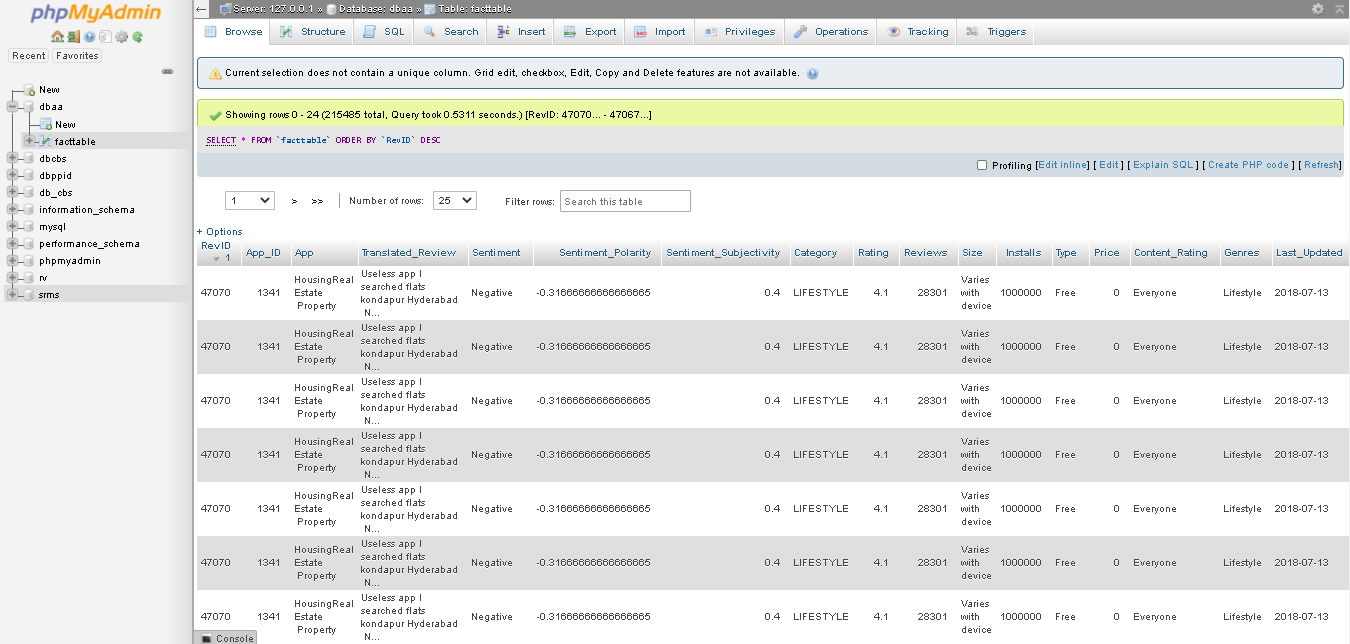


Figure 4.0: tDbOutput implementation

  
Figure 4.1: tDbOutput component

  
Figure 4.2: Data loaded into the database component phpMyAdmin and mySQL

using the dbOutput

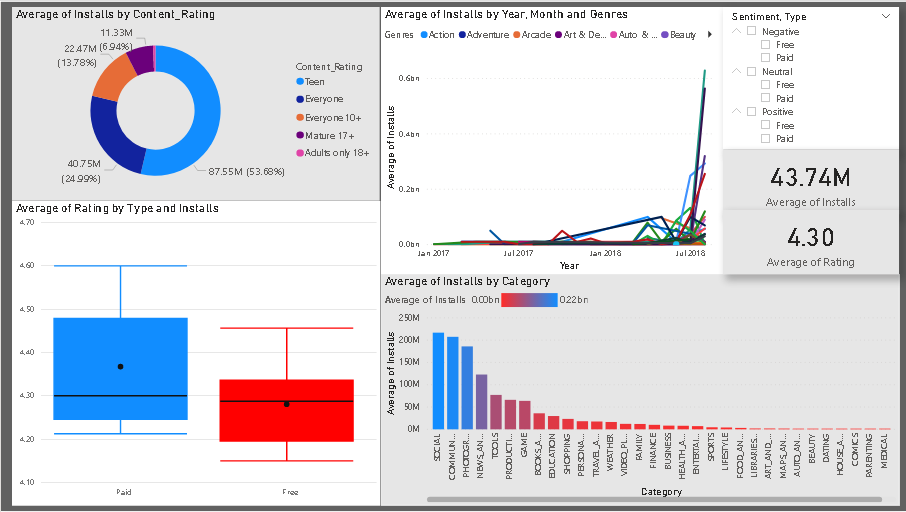
# Data Visualization

PowerBI is used to generate appropriate visualizations for the data from the data warehouse after the ETL process. In order to give meaningful insights, it is important to find relations between the data and choose a suitable presentation of the data.

## Reports

**Report 1:** Average Installs

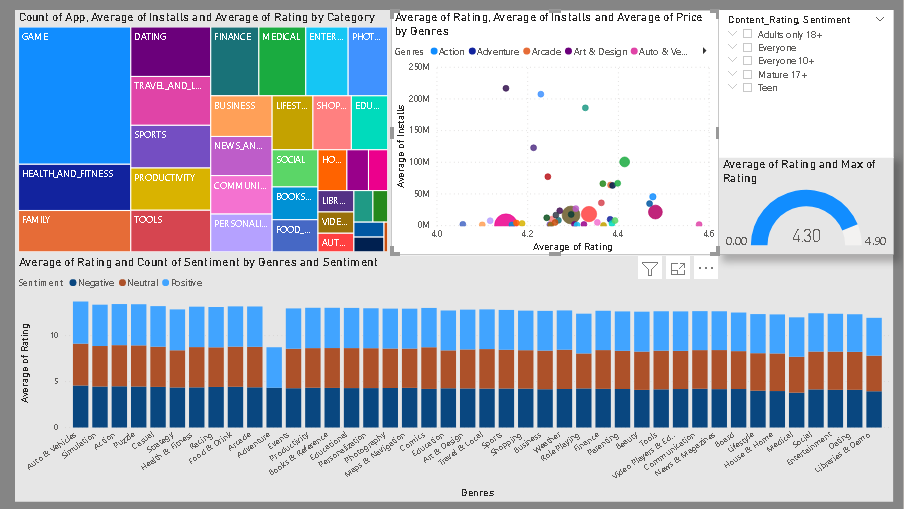
Report1 consist of four different charts such as Donut chart, Box plot, line chart, and bar graph. It also includes a slicer on Sentiment and Type for more interactivity.

  
Figure 5.0.1: Report 1

The goal of this report is to study the type, category, ratings, and sentiment based on the Average installs on the applications. It could give the top managements insights on which type of apps are most likely to be installed, what are the most favorible categories, compare it to average ratings, and see what are the trends on installed apps by genre.

**Report 2:** Average Rating

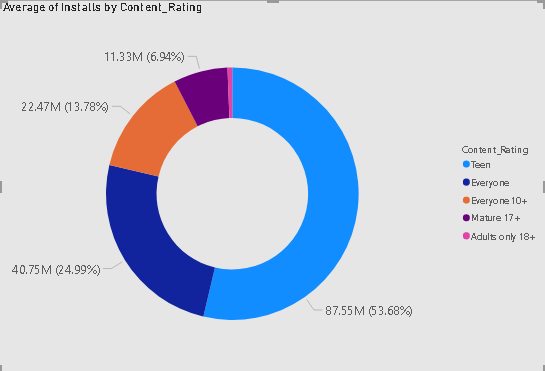
Report2 consist of three different charts such as Treemap, scattered plot, and stacked bar chart. It also includes a slicer on Content rating and Sentiment for more interactivity.

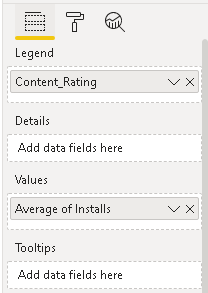
  
Figure 5.0.2: Report 2

The goal of this report is to study the type, category, ratings, and sentiment based on the Average ratings on the applications. It could give the top managements insights on which type of apps are most likely to be recommended, what are the most favorible categories, compare it to average installs on the scatter plot, and could helps in determining which app is reliable or legitimate based on the rating and sentiments.

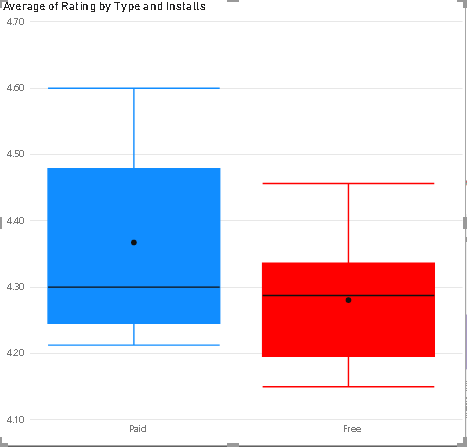
## Charts

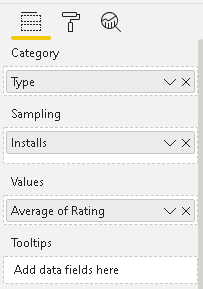
**Chart #1**: Average installs by Content Rating  
 This chart uses Donut Chart to give percentage of the average installs by content Rating. It gives insights on what content rating of the applications that are more favorable to be installed. This chart shows that the highest applications installed are for Teen, and the least would be for Adults Only applications.

  
Figure 5.1.1: Average of applications installed by Content rating

  
Figure 5.1.2: The columns and values used for the donut chart

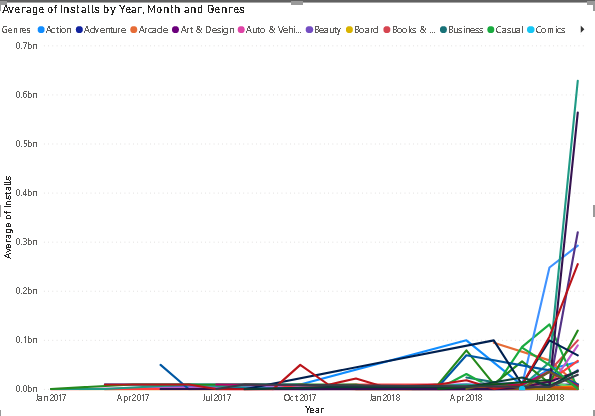
**Chart #2:** Average Rating by type and installs  
 This chart uses Box Plot to give insights on the mean, max, min on the ratings, by using total installs as samples and comparing between the Paid apps and Free apps. We can see that the median of ratings between both types are not that different from each other.

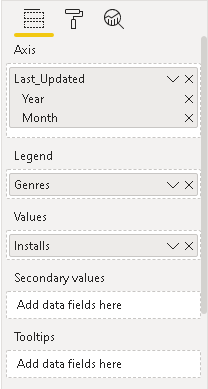
  
Figure 5.2.1: Average rating by type and installs

****Figure 5.2.2: The columns and values used for the box plot

**Chart #3:** Average installs by Date and Genres

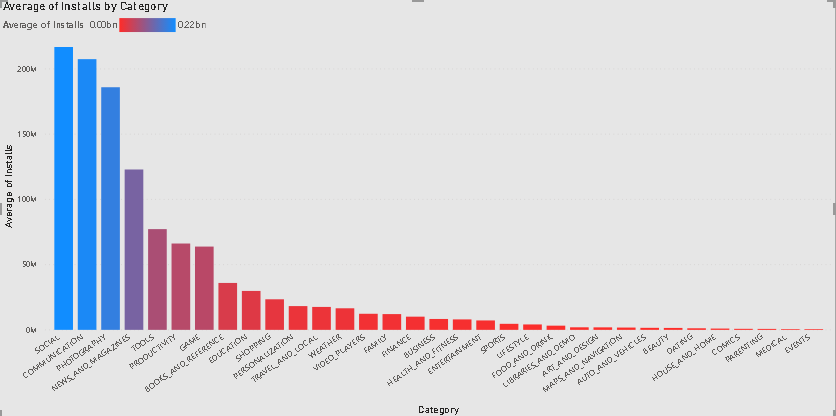
This chart uses Line chart to give insights on what genres are trending and most likely to be growing or installed by the users. Over the course of 2017 until 2018, we can see that the most installed photography genre apps followed by social media and news magazines.

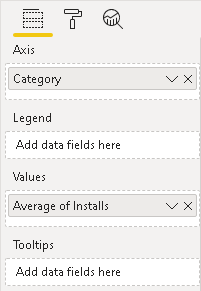
  
Figure 5.3.1: Average installs by Date and Genres

****Figure 5.3.2:The columns, values, and legends used for the line chart

**Chart #4:** Average installs by category

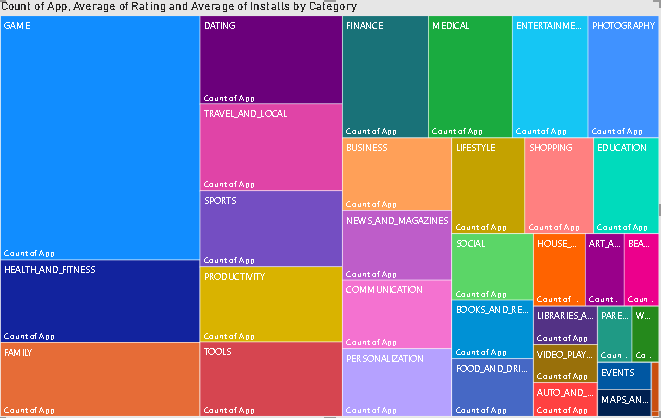
This chart uses bar graph to give insights on what categories are the top and most likely to be installed by the users. We could see that there’s a huge gap in applications installed by categories between social, communication, photography and other categories.

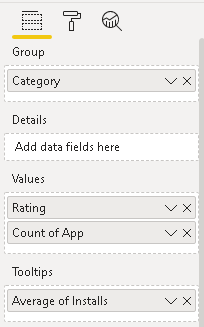
****Figure 5.4.1: Average installs by category

****Figure 5.4.2:The columns and values used for the bar graph

**Chart #5:** Count of App, with average installs and rating by Category

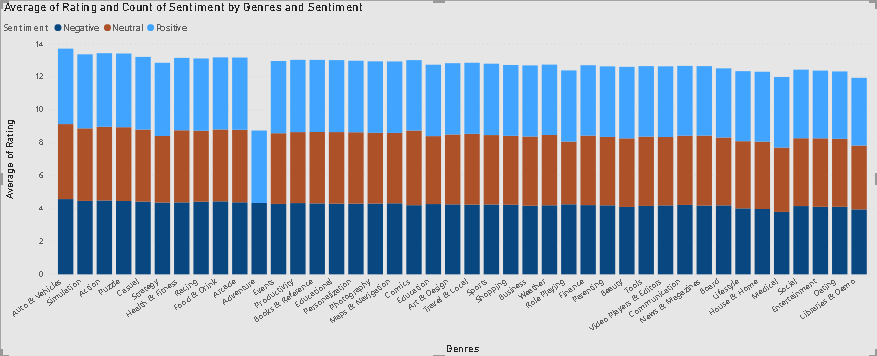
This chart uses Tree map to give insights on which categories are the most installed. Tooltips is used to show average installs and rating of the category when the cursor hovers over it. This also shows that category game has the highest count in the platform followed by health and fitness, and family.

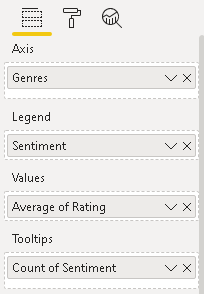
****Figure 5.5.1: Count of App, with average installs and rating by Category

   
Figure 5.5.2:The group, values and tooltips used for the treemap

**Chart #6:** Average Rating and sentiment count by Genres and Sentiment

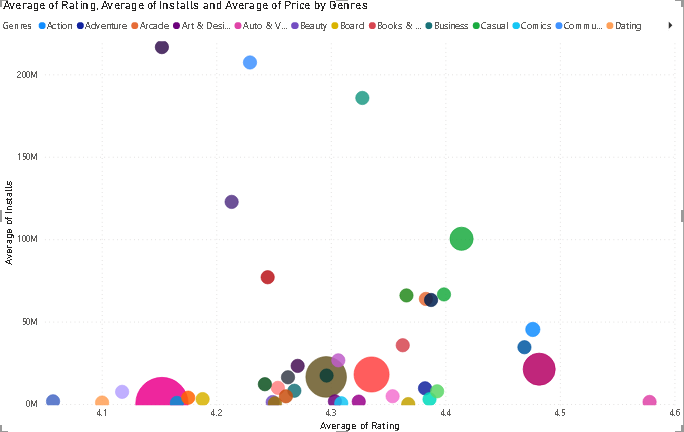
This chart uses stacked bar chart to give insights on ratings on genre and the comments’ sentiment. Tooltips also used to show the count of the sentiments when the cursor hovers over it. After sorting, this also shows that the genre Auto&Vehicle has the highest rating.

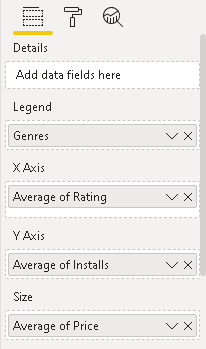
  
Figure 5.6.1: Average Rating and sentiment count by genres and sentiment type

  
Figure 5.6.2:The columns, legend, values and tooltips used for the stacked bar graph

**Chart #7:** Average Rating and average installs by average price and genres

This chart uses scatter plot to give insights on the relationship between average ratings and average installs by genre. Size of the plot determines the average price in the category. This chart shows that medications category has the highest price, but have a lower than average in rating and installs.

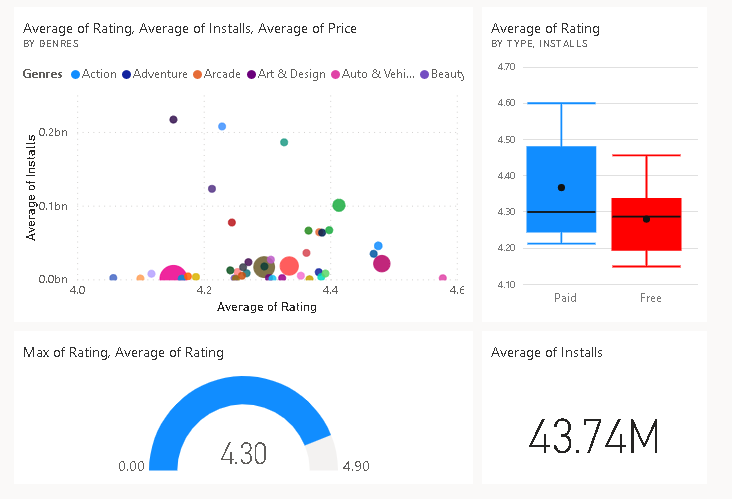
  
Figure 5.7.1: Average Rating and average installs by price and genres

  
Figure 5.7.2:The Legends, axis and size used for the scatter plot

## Dashboards

**Dashboard:** Relationship between Ratings and Installs

This dashboard consists of two charts and two visualizations from both of the reports, that will help the top managements to see the relations between ratings and installs. The charts chosen for this dashboard could give little more insights on the types, categories, and price of the apps in their platforms.

  
Figure 6.1: Dashboard from PowerBI Service

The dashboard shows the average installs, and average of rating by type using total installs as samples could help the top management see the mean or median rating and average applications installed from their platform easily. They could also see the average rating and the max rating of the applications and further see what categories are leading in rating and installations. All of the components in the dashboard could be clicked to see the reports that would give further and more focused insights on every topic.

# Conclusion

The further the year progressed, the more of these applications are being installed from the platform. By knowing the relationship between user’s rating and installations with the application’s genre, category and type, the top management could gain more insights on the interactions on their platform. With a proper ETL architecture, data warehouse design and data visualizations, the datasets could grant more informations on the apps and its reviews or ratings. Thus, further helps their recommendation algorithm for the platform, and not only by using the search history and clicks or interactions.  
 By using this analysis, it could offer more comprehension on what category or type of applications the users from the platform desire, what applications are reliable or credible, and what recommendations are suitable to ensure the dependability and maximize the time usage of their platform.

# References

Frie K, Hartmann-Boyce J, Jebb S, Albury C, Nourse R, Aveyard P, Insights from Google Play Store User Reviews for the Development of Weight Loss Apps: Mixed-Method Analysis, JMIR Mhealth Uhealth 2017;5(12): e203. Retrieved 30 January 2021, from <https://mhealth.jmir.org/2017/12/e203>

Islam, M. R. (2014, April). Numeric rating of Apps on Google Play Store by sentiment analysis on user reviews. In 2014 International Conference on Electrical Engineering and Information & Communication Technology (pp. 1-4). IEEE. Retrieved 31 January 2021, from <https://ieeexplore.ieee.org/abstract/document/6919058/>

Martin, W. (2016, May). Causal impact for app store analysis. In Proceedings of the 38th International Conference on Software Engineering Companion (pp. 659-661). Retrieved 31 January 2021, from [https://dl.acm.org/doi/abs/10.1145/2889160.2891033](https://dl.acm.org/doi/abs/10.1145/2889160.2891033?casa_token=5udM41oeOCYAAAAA:yOEKl4l4fcFvI0dBf154XBGucD6qt_3Cudn6gKfpufqu2z8oxJT1hmdHUPs4t2bHirZ_BuuY1T07pg)