Report on Data Warehouse Creation and Usages in Simple Business Intelligence for Soccer Goal Statistics Using Kimball Methodology

Created By:

Andreas Pangestu Lim
andreas.lim@binus.ac.id

Jonathan
jonathan016@binus.ac.id

Advanced Database Systems

MTI 2019/2020 1st Period

Company/User Profiles

On creating our data warehouse, we obtained the data from Kaggle (https://www.kaggle.com/hugomathien/soccer) with consideration of our data warehouse's end-users. This section will describe our data warehouse's expected user profiles, which are:

1. Soccer Fans
   A soccer fan is a person who usually is bound to a specific team competing in a specific league, though his/her preference of team and league could at times be unrelated. He/she would love to see his/her team win a match, even more satisfied when his/her team beats a popular rival of his/her team. A rival team is a team which is either coming from the same city as the other team, a team that has equal win-lose ratio with the other team, or other causes that made the team identified as rival.
   A fan may just casually follow his/her team's journey from television or internet, or he/she may come to the stadium for watching the match on site, or even manage fans club and organize support for the team as his/her full-time job.
2. Soccer Team Manager
   A soccer team manager is the person responsible for building the team's performance and training the team to have his/her gameplay schema during a match. Due to this reason, losing a match is often associated with the manager's incompetence or fault, though that may not always be the case.
   A manager usually prepares his/her team before facing another team in a match, modifying strategies while considering previous match statistics. After the match, he/she would evaluate the team's performance, one among many key factors being the amount of goal produced by the managed team and the enemy team.
3. Soccer Club Owner
   A soccer club owner is the person who possesses the most share of the team among other shareholders. He/she has the power to fully remove a manager, to change the team, and other benefits as he/she owns the club itself.
   There are many reasons as to why a person would like to have his/her own soccer club, but one that can be assumed is that having a good and reputable club could bring him/her more fame and image, hence the performance of the team is important to him/her as the team not only acts as his/her investment, but also contributes to his/her judgment as a successful person.

Business Processes

Since in the previous section we have described our user profiles, we will describe the business process of the overall data obtained from Kaggle first, then we will continue with business process explanation of each user profile.

An overview of soccer is that a match consists of two teams, a home team and away team. Home team is the team hosting the match, in that the match takes place at the home team's stadium. Away team is the team which is playing against the home team. A match in our database occurs only in a specific season and league of a country, although in real life a match may occur anytime either inside or outside an ongoing season and league of a country. The objective of a match is to score goal against the enemy team, and should the scored goal(s) exceed the enemy's goal(s), then the team is declared as winner. The data in our Kaggle database is match statistics obtained from eight seasons across eleven leagues of eleven countries, with over 288 FIFA-registered teams competing. By FIFA-registered team, we mean the team has FIFA ID in our database.

A soccer fan usually watches and predicts output of a match based on previous match statistics. As stated before, his/her favorite team's win would excite him/her and brings pride against other team fans, especially if the win is over a rival. Note that this is supposed to be used in an entertainment manner, not in negative manner such as betting, though in reality some people misuse the data for negative purposes.

A soccer team manager is responsible for preparing his/her team before a match, and preparation may be done by considering previous match statistics against the enemy team. Seeing the trend of the team's performance in previous season(s) may also help the manager devise a plan for having better trend or maintain previous trend in this season or upcoming season(s). With the right plan and execution, his/her managed team's victory and performance should have good trend.

A soccer club owner is concerned about how his/her team fares against other teams, since he/she invests in the club and the club's performance has impact on his/her image. By evaluating the team's statistics, he/she may be able to make decisions for the future of the club. Either replacing the manager, buying more players, or other actions that he/she would like to take.

Data Warehouse and Database Design

To start, as Kimball Lifecycle methodology steps state, we need to define the project first. We defined our project with the goal of delivering a dashboard of goal statistics for home team. With this project goal, we also specified our project's scope, which is limited to home team goals itself. We only use one project for this data warehouse example, so it can be inferred that our data mart is the data warehouse itself.

Afterwards, since we act as both project team and the end-users, we decided what would be the specific business process we would like to solve. We looked into our data obtained from Kaggle itself which is stored in a single `.sqlite` file. However, we would like to store the OLTP database's data in MySQL using XAMPP as by using XAMPP we do not have to install a database server due to XAMPP's feature to embed database server. We used SQLite feature to extract the data to CSV (Comma-Separated Values) file by exporting the database to CSV files, each for each table in the database. The full schema of the OLTP database can be seen in the Github repository (https://github.com/jonathan016/dwh-pentaho-soccer-kaggle) since it cannot be included in this report due to its massive fields.

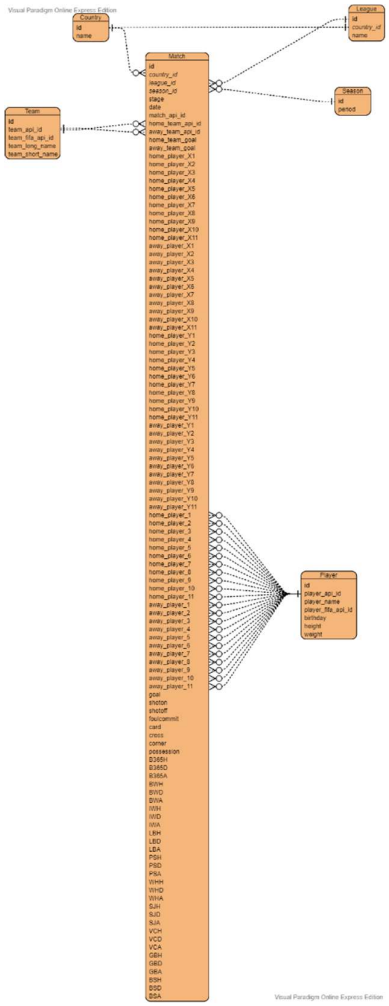


Figure 1 - OLTP Database ERD Schema (http://bit.do/ADS-ERD)

Unfortunately, after looking at the data, we noticed that several columns were nullable, was in XML format, had improper date representation, and many other inconsistencies. We did some fixes to the OLTP database's data, which are changing match date's representation from string to date and extracting season data from match table to a new table. This is to normalize the OLTP database and by doing so adds more dimension to our data warehouse. We cannot use our ETL tool to fix the data during extraction and transformation processes due to the tool's limitation, so we had to fix the data manually.

Our next step is to determine our technical architecture. We set several conventions, such as dimension tables must be named with suffix `_dim`, fact table must be named with suffix `_fact`, and each word in table name must be separated with underscore and is in lowercase format. We also selected our tools for ETL and business intelligence application, which are Pentaho Data Integration (PDI) and Qlik Sense with Pentaho Report Designer (PRD), respectively.

The next step is to identify our facts and dimensions. We investigated the OLTP database's data and figured that our dimensions would include month, season, team, and league dimensions. The league dimension will contain country name as well, since each league is located on a single country, but we would like to keep the opportunity for adding another league on the same country possible. Our main fact on the fact table will be the number of goals scored (`goal_for`), goals conceded (`goal_against`), and we added another fact which is derived from the goals scored subtracted with goals conceded (`total_goal`). We added the latter field to improve our data warehouse's performance as some of our dashboard KPIs will be based on the latter field. Our data warehouse's schema can be seen in the figure below.

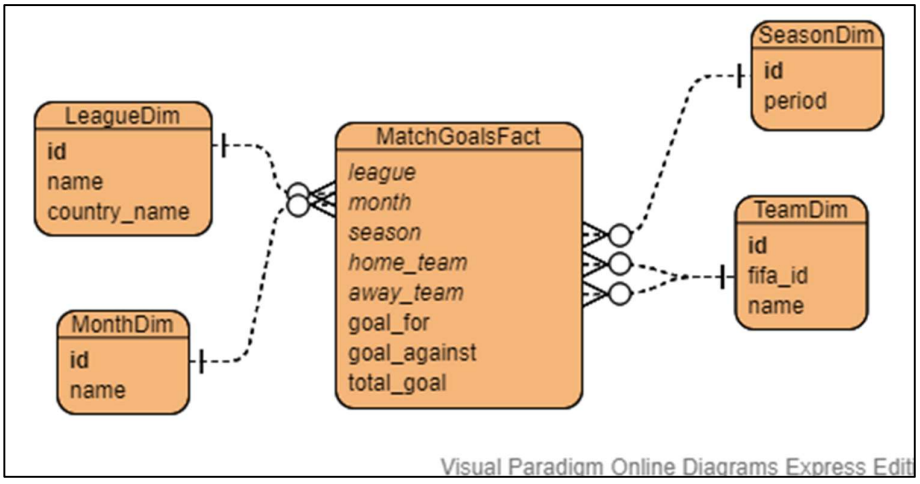


Figure 2 - Data Warehouse Star Schema (http://bit.do/ADS-SS)

Then we continue with the physical database design. We did not have multiple environments settings, nor did we require it due to the scope of the project is small and is simple enough to be done in one environment. Our table naming convention was already setup during our initial steps, and we had also used SCM to manage our progress in the development process. Our only works in this step were implementing our database schema using SQL queries. Some example of our queries as given below; for all scripts containing our queries can be seen in our Github repository (https://github.com/jonathan016/dwh-pentaho-soccer-kaggle).

```
DROP DATABASE IF EXISTS `soccer_dwh`;
CREATE DATABASE `soccer_dwh`;

USE `soccer_dwh`;
DROP TABLE IF EXISTS `league_dim`;

CREATE TABLE `league_dim`(
  `id` INT(11) AUTO_INCREMENT NOT NULL,
  `name` text COLLATE utf8_bin NOT NULL,
  `country_name` text COLLATE utf8_bin NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
---
...
---
USE `soccer_dwh`;

DROP TABLE IF EXISTS `match_goals_fact`;

CREATE TABLE `match_goals_fact`(
  `team` INT(11) NOT NULL,
  `month` INT(11) NOT NULL,
  `season` INT(11) NOT NULL,
  `league` INT(11) NOT NULL,
  `goal_for` INT(11) NOT NULL,
  `goal_against` INT(11) NOT NULL,
  `total_goal` INT(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

Figure 3 - Sample Queries for Data Warehouse Schema Implementation (http://bit.do/ADS-SSIS)

We created five separate transformation files, four for our dimension tables and one for our fact table. By creating separate transformation files, we have a clear view of the transformation process for each dimension and fact. Although this separation requires more effort in running ETL compared to grouping all the ETL flows in one transformation file, this eases our ETL flow development. Moreover, when the OLTP data grows larger – vertically and horizontally – this separation compartmentalizes ETL complexity per transformation file.

Some of our transformations include joining two tables, getting and mapping specific column values of a table, and concatenating values from two columns of a table. This is done to enrich data in our data warehouse's dimension. An example of our transformation is included in the figure below.

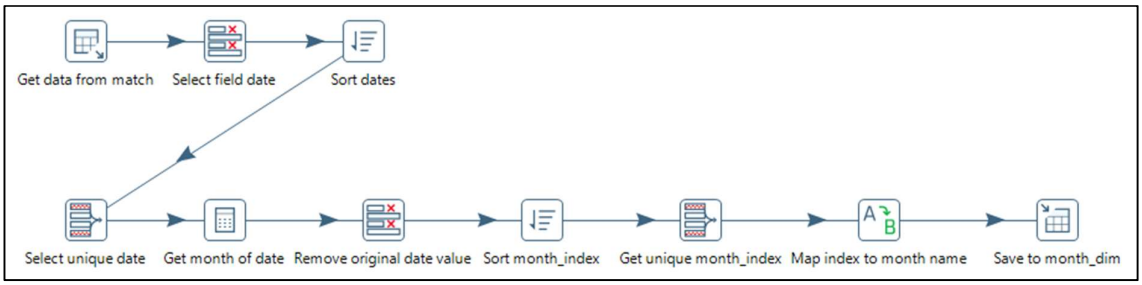


Figure 4 - Month Dimension Transformation (http://bit.do/ADS-TMD)

As can be seen, our transformation is simple, even though many steps are included in the transformation process. These steps ease our understanding of what happens during the ETL process and provides more graphical intuition for each step the data went through in the processing.

The above transformation shows how we extract unique month name only from the match dates in match table in our OLTP database. We will not dive deep into how PDI works, and what each step does, but we will explain how we did the transformation. By fetching and selecting only the date data from match table, we then select unique dates from those matches, as several matches occur in the same date. However, PDI throws a warning to sort the dates first prior to selecting unique values, therefore we fixed our ETL flow by adding sorting every time we would like to select unique values. Then, we get the month of date using PDI's calculator step. Unfortunately, we could not remove the original date in the calculator step, as we only want the month of date itself – which we assigned to a field/column with name `month_index`. Thus, we need to remove the original date value. Note that the month of date is in numerical value, such that October is represented as 10 as of this step. Then we select the unique `month_index`, and we map those indexes to month name by our own mapping definitions in the '*Map index to month name*' step. Finally, we save the mapped values to our output table, which resides in our data warehouse with table name `month_dim`, representing month dimension in our data warehouse's fact table.

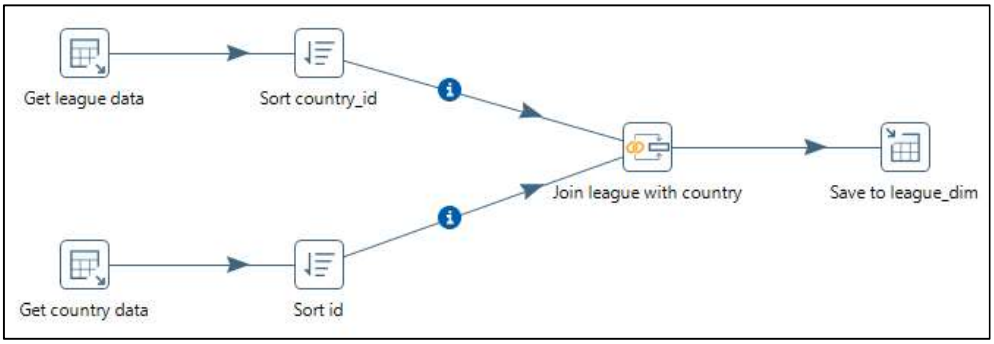


Figure 5 - League Dimension Transformation (http://bit.do/ADS-TLD)

The league dimension transformation is simpler than month dimension as it only gets the league data from OLTP database and join the data with country data to keep the country name in league dimension in our data warehouse. We also need to sort the field of our data prior to joining data results. Finally, after successful join operation, we save to our `league_dim` table in the data warehouse.

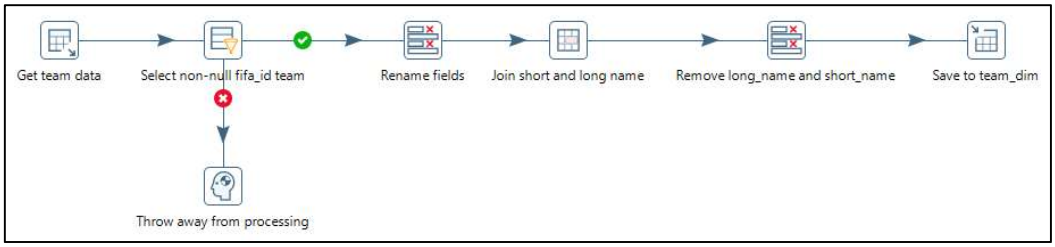


Figure 6 - Team Dimension Transformation (http://bit.do/ADS-TTD)

Our team dimension transformation is even more simple compared to league dimension transformation as it only selects all data, then filters those with null FIFA ID, which we take as unregistered FIFA team. We then rename the fields for our next processing, which is transforming the data by joining the team's long name (example: Arsenal) with its short name (example: ARS) into the following format: `<short_name> - <long_name>`. Lastly, we save the transformed data to our `team_dim` table in the data warehouse.

Figure 7 - Season Dimension Transformation (http://bit.do/ADS-TSD)

Our season dimension transformation is the simplest of all transformation, since it only requires the name of the season, or in the OLTP database referred as `period` in table `season`.

Note that in our dimension transformations, we never save the original identifier used in the OLTP database as we need to use surrogate key for storing dimension data in our data warehouse per Kimball Methodology. We only store `fifa_id` since a team could be identified by its `fifa_id`, hence we believe that `fifa_id` is important to be maintained in our data warehouse data.

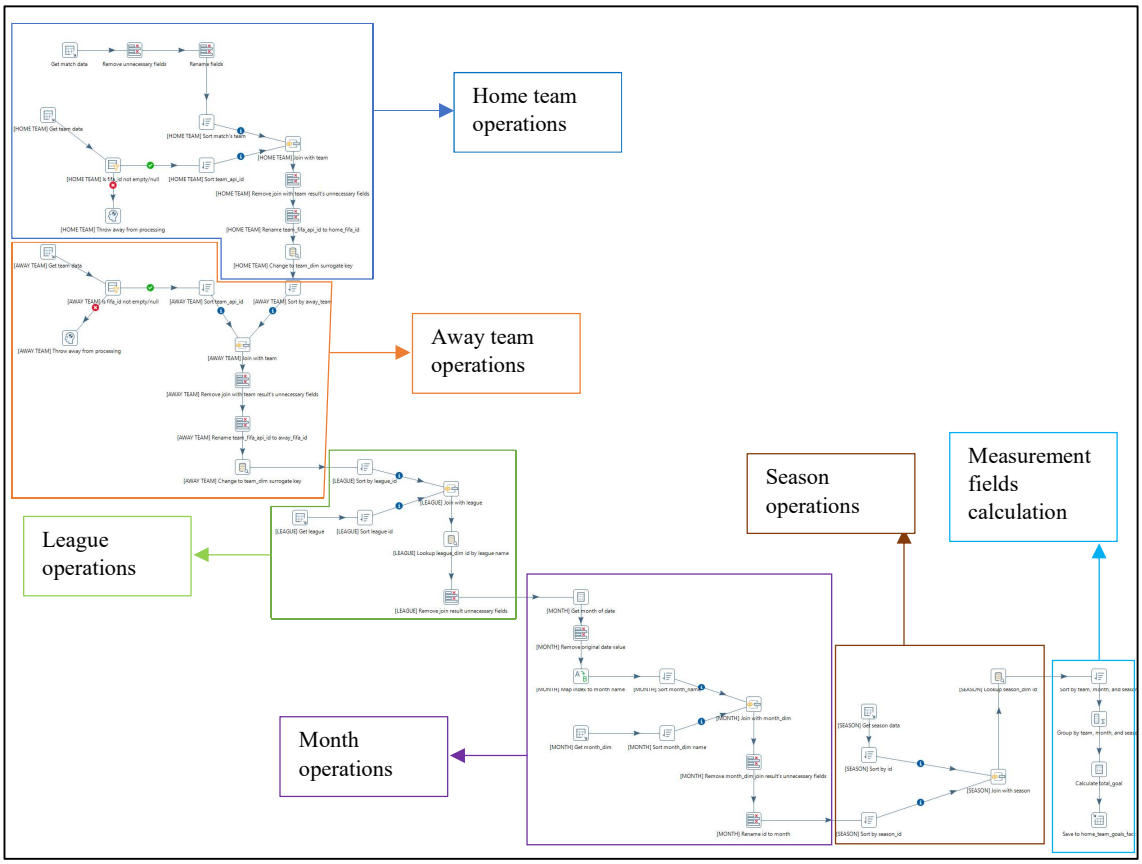


Figure 8 - Fact Table Transformation (http://bit.do/ADS-TMGF)

The most complex transformation is our fact table transformation. Our fact table consists of five foreign keys referencing four dimension tables. We need to join our OLTP database's tables with our data warehouse's tables to obtain our data warehouse's surrogate keys. Our steps include fetching the home team data (dark blue box). We filter those with null `fifa_id` attribute, then we get the surrogate key of the team from our `team_dim` table using merge join operation in PDI. We then fetch and join the processed data with the away team data with the same methodology as how we did the home team (orange box). Afterwards, we fetch the league data and does lookup in our data warehouse database on table `league_dim` and set the currently processed data's league to our `league_dim` id field (green box). Then we continued merge join with month data with prior processing which includes mapping the month of each match data with its month name and perform join operation to obtain the surrogate key of our `month_dim` table in data warehouse (purple box). We also looked up `season_dim` to get surrogate key for season data (brown box). Finally, we calculate our measurement fields using sum aggregation function (light blue box). After all joins and measurement fields calculation, we save the processed data to our data warehouse.

Dashboard Creation Using Qlik Sense

Since all steps toward ETL process have been executed, we began our business intelligence application preparation. Many tools are available for representing data warehouse's data, such as Tableau and Qlik Sense. The representation in Qlik Sense is saved in a project, where each project can have its data, analysis sheets, and stories of those analysis. We use Qlik Sense Desktop for representing the data, though it may be noted that Qlik Sense does provide cloud platform for representing data using Qlik Sense itself.

To load our data warehouse's data, we installed and registered MySQL ODBC connector to our computer. After registering the ODBC, we connect Qlik Sense with our data warehouse and load the data from Qlik Sense's GUI by specifying our targeted database connection. Qlik Sense automatically detects relation between each table, even though some relations may be mapped incorrectly. In our case, only one relation was wrongly created, which is the relation between our fact and month dimension. Fortunately, Qlik Sense offers a mechanism for modifying relation between tables manually, and after specifying what relation both tables have, our data warehouse's data is synchronized perfectly in Qlik Sense. We then load the data to the analysis tab for creating analysis sheets of the data.

The analysis tab provides many graph options for representing our model. Some examples include scatter plot, bar chart, KPI, pie chart, and other visual statistical representations of our data. We created four sheets, one for overview of our data warehouse's data, and three sheets for data representation grouped by time, location, and team, respectively. Each sheet illustrate connection between fact table and dimension table(s); time-grouped sheet includes season and month dimension tables; location-grouped sheet includes league and country dimension tables; and team-grouped sheet includes team dimension table. In each sheet, more than one graph can be used to depict data, and filling data to a graph is done easily using Qlik Sense's GUI. Our data representations are given in the figures below.

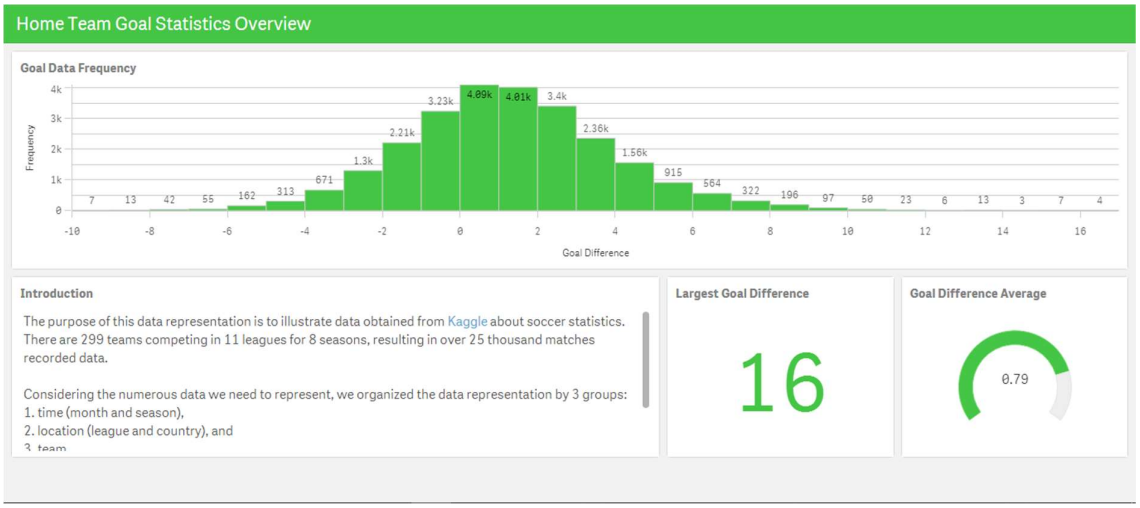


Figure 9 - Dashboard of Overview of Data Warehouse (http://bit.do/ADS-DO)

Figure 9 is an overview of the data warehouse's content, along with goal difference frequency and some notable KPIs that we would like our end-users to see. We also provide a text box containing brief information and disclaimers of our dashboards.

Figure 10 - Dashboard of Time Dimension (Month and Season) of Data Warehouse (http://bit.do/ADS-DT)

In figure 10, we provide a mixed dimension representation of our data warehouse's data, which include month and season dimension, as both dimensions are in the same category, which is time. We provide capabilities for filtering, line chart, bar chart, and pie chart for showing the goal statistics of each month or each season. The graphs are customizable, in that each graph can be used to select views based on either month or season.

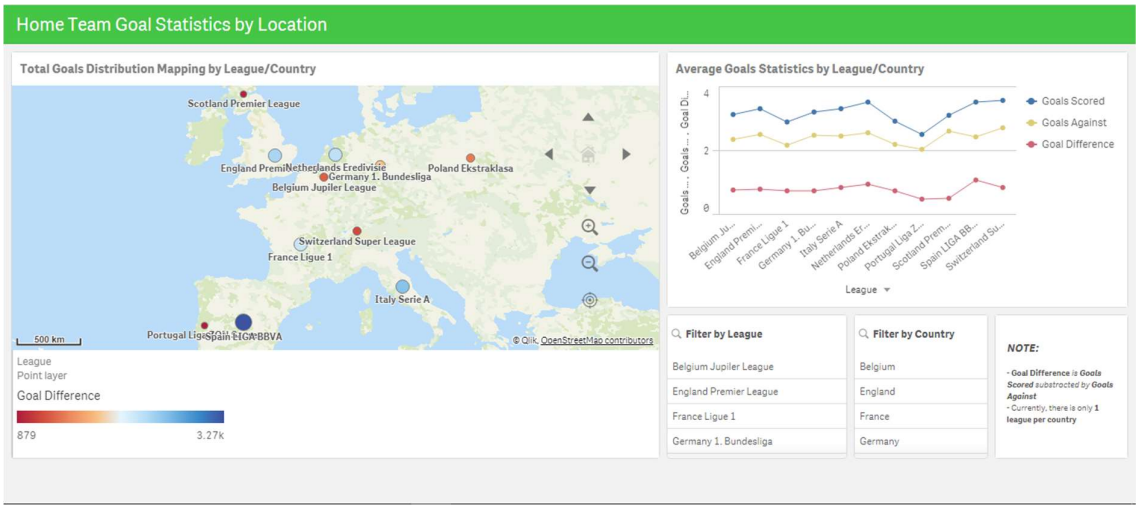


Figure 11 - Dashboard of Location Dimension (League) of Data Warehouse (http://bit.do/ADS-DL)

Figure 11 includes total goals heatmap by the size of the dot over leagues, which could be mapped with the presence of the league's country name, which is used by Qlik Sense to automatically map the data based on location. We also provide filtering ability, either by league or by country, which actually would have the same result. We did that since we would like to scale the data warehouse to support multiple league per country, while our current database does not support that.

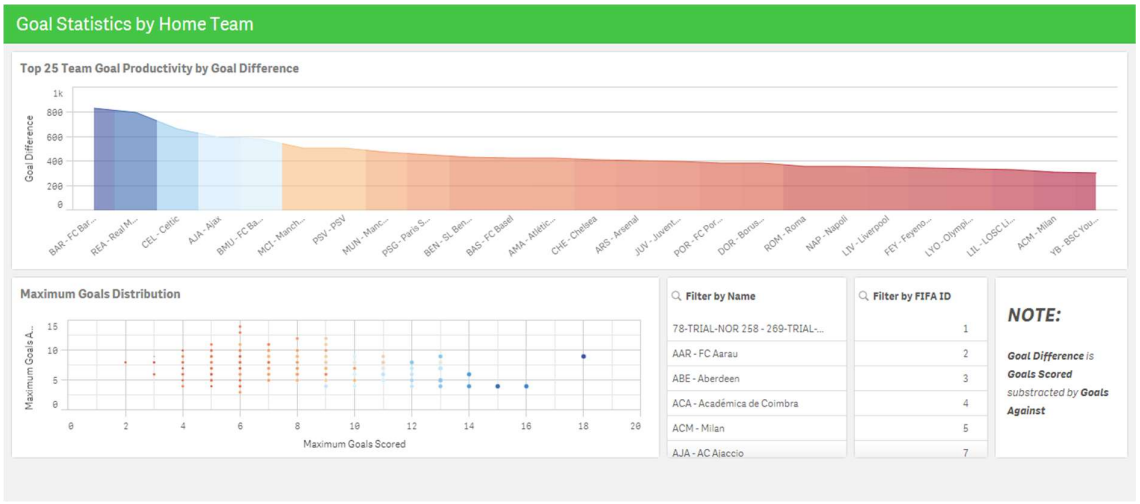
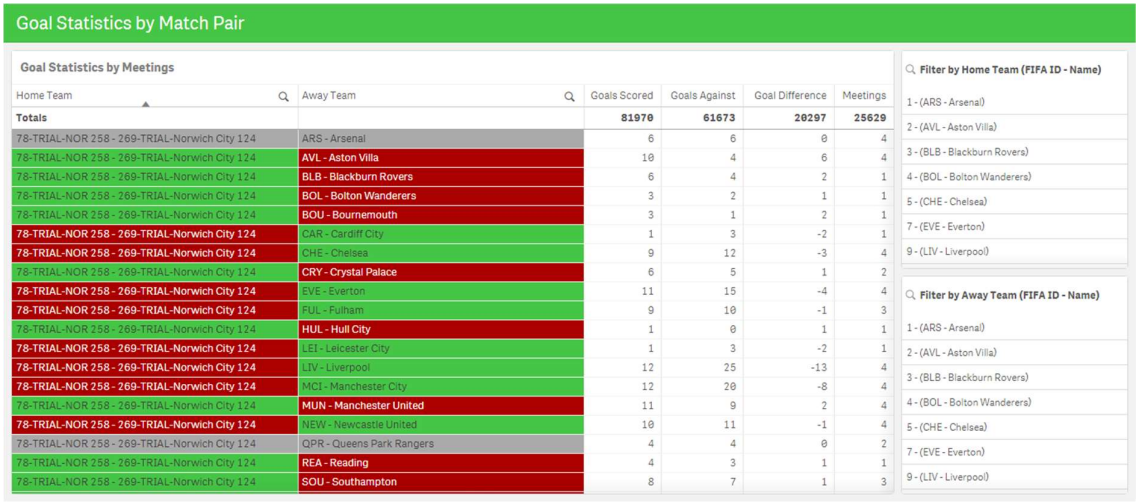


Figure 12 - Dashboard of Home Team Goal Statistics (http://bit.do/ADS-DH)

Figure 12 shows our dashboard for goal statistics by home team. We displayed the goal productivity for top 25 teams, and also distribution graph and filtering abilities. The top 25 teams' data could be obtained just by specifying the limit of data to be retrieved. The filters available are filter by team name and by FIFA ID.



Home Team	Away Team	Goals Scored	Goals Against	Goal Difference	Meetings
Totals		**81970**	**61673**	**20297**	**25629**
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	ARS - Arsenal	6	6	0	4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	AVL - Aston Villa	10	4	6	4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	BLB - Blackburn Rovers	6	4	2	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	BOL - Bolton Wanderers	3	2	1	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	BOU - Bournemouth	3	1	2	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	CAR - Cardiff City	1	3	-2	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	CHE - Chelsea	9	12	-3	4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	CRY - Crystal Palace	6	5	1	2
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	EVE - Everton	11	15	-4	4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	FUL - Fulham	9	10	-1	3
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	HUL - Hull City	1	0	1	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	LEI - Leicester City	1	3	-2	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	LIV - Liverpool	12	25	-13	4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	MCI - Manchester City	12	20	-8	4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	MUN - Manchester United	11	9	2	4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	NEW - Newcastle United	10	11	-1	4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	QPR - Queens Park Rangers	4	4	0	2
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	REA - Reading	4	3	1	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	SOU - Southampton	8	7	1	3

Figure 13 – Dashboard of Match Pair Meetings Goal Statistics (http://bit.do/ADS-DMP)

Finally, in figure 13, our last dashboard, we provide meeting statistics by match pair. This means we aggregate our data warehouse's data by specific home and away team meetings. We calculate the total goals scored, goals against, and goal difference viewing from home team perspective. We also decorated the table with some colors, where grey means tie, green means win, and red means lost. Filtering abilities are also included, and to have this dashboard, we had to load the `team_dim` data twice since in one load, Qlik Sense does not allow multiple associations between the fact table and dimension table.

Report Creation Using Pentaho Report Designer

We also created several reports using Pentaho Report Designer. Pentaho Report Designer allows us to create report by utilizing and making connection to our data warehouse using JDBC connector. By adding the binary for connector to library of Pentaho Report Designer, we were able to connect to the database and query data from Pentaho Report Designer itself. The binaries are obtainable from http://bit.do/ADS-OU.

We created several views to make our report creation easier, and this is to ensure that only specific data can be seen as specified by the view for specific users, though in our case we did not specify which user can use which view since it is out of our scope. The result is that when generating the report, all we need to do is to select all data from a view, hence making our query on Pentaho Report Designer shorter and clearer. Screenshots of the generated reports are given below.

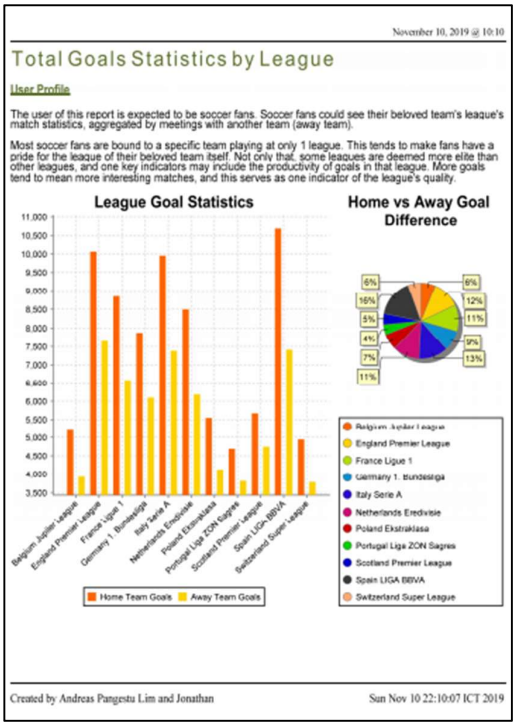


Figure 14 - Report of Goal Statistics by League (http://bit.do/ADS-RL)

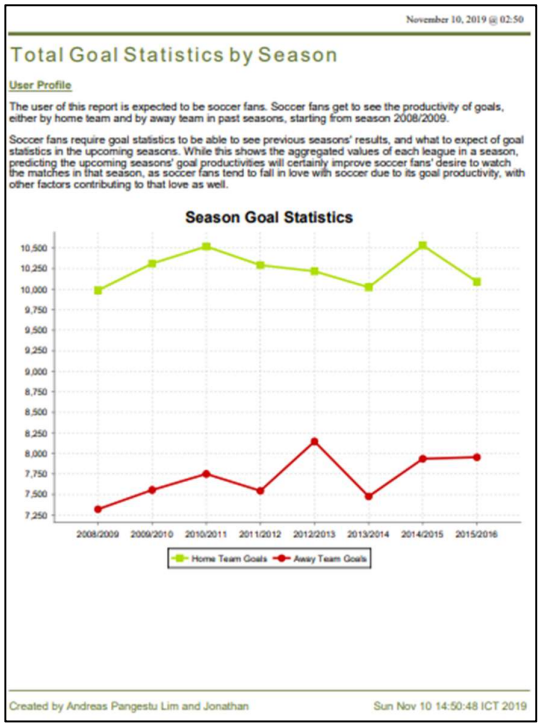


Figure 15 - Report of Total Goal Statistics by Season (http://bit.do/ADS-RS)

The reports generated are total goal statistics by league, season, and table of meeting goal statistics between home team and away team.

In the report grouped by league, we added a pie chart for showing how each league contributes to total goal sum. Not just about how each league contributes to total goal sum, we also provide a bar chart for showing the statistics of home team and away team goal statistics per league.

We also created the report grouped by season, in which we show the line chart of goal statistics (goal difference) by home team and by away team. This is merely a representation of

how productive each season is for home team and for away team and may be useful for determining the trend of upcoming season's for playing as home and playing as away team's goal productivity.

November 10, 2019 @ 10:00

Home Team	Away Team	Meetings	Goals Scored by Home Team	Goals Scored by Away Team	Goal Difference
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	ARS - Arsenal	4	6	6	0
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	AVL - Aston Villa	4	10	4	6
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	BLB - Blackburn Rovers	1	6	4	2
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	BOL - Bolton Wanderers	1	3	2	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	BOU - Bournemouth	1	3	1	2
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	CAR - Cardiff City	1	1	3	-2
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	CHE - Chelsea	4	9	12	-3
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	CRY - Crystal Palace	2	6	5	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	EVE - Everton	4	11	15	-4
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	FUL - Fulham	3	9	10	-1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	HUL - Hull City	1	1	0	1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	LEI - Leicester City	1	1	3	-2
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	LIV - Liverpool	4	12	25	-13
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	MCI - Manchester City	4	12	20	-8
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	MUN - Manchester United	4	11	9	2
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	NEW - Newcastle United	4	10	11	-1
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	QPR - Queens Park Rangers	2	4	4	0
78-TRIAL-NOR 258 - 269-TRIAL-Norwich City 124	REA - Reading	1	4	3	1

Created by Andreas Pangestu Lim and Jonathan

Sun Nov 10 22:00:47 ICT 2019

Figure 16 - Report of Goal Statistics by Team Meetings (http://bit.do/ADS-RTM)

In addition to previous reports, we provide a report for each team meetings statistics across eight seasons. The report is grouped by home team and away team meetings, and without considering each match's result, we only take the sum of goals scored by home team and away team to represent the team's record of goal productivity against another team.

As stated before, we created some views to help with the reporting query, and even though we did not use the views to limit individuals from accessing the data, we prepared the views for future plans of scaling the data warehouse itself. Below is an example of queries we made to create the view.

```
CREATE VIEW `team_total_goal_statistics` AS SELECT
    t1.name AS home_team, t2.name AS away_team,
    SUM(goal_for) AS total_goal_for,
    SUM(goal_against) AS total_goal_against,
    SUM(total_goal) AS total_goal, COUNT(*) AS meetings
FROM `match_goals_fact`
JOIN team_dim AS t1 ON match_goals_fact.home_team = t1.id
JOIN team_dim AS t2 ON match_goals_fact.away_team = t2.id
GROUP BY t1.name, t2.name;
```

Figure 17 - Example of Queries for Creating Report Views (http://bit.do/ADS-RV)

Conclusion

In this report, we have presented data warehouse creation by applying Kimball Methodology. Though some steps were omitted due to our data simplicity and limitation, we have pointed out which parts that do not follow the steps and how it should be done in Kimball Methodology proper implementation.

We have also elaborated our ETL process using Pentaho Data Integration, alongside the transformations themselves. A simple dashboard of five analysis sheets using Qlik Sense is explained as well, with addition reports generated using Pentaho Report Designer by utilizing graph representations and parameters in the generation of the report.

The complete work reported in this report can be obtained from Github (https://github.com/jonathan016/dwh-pentaho-soccer-kaggle). We have made two release versions on the repository, and the release version reported in this report is v2.0.0 and is current master branch. We provided the repository of the project for readers to recheck the validity and have more detailed view of this project, should it be required or necessary.