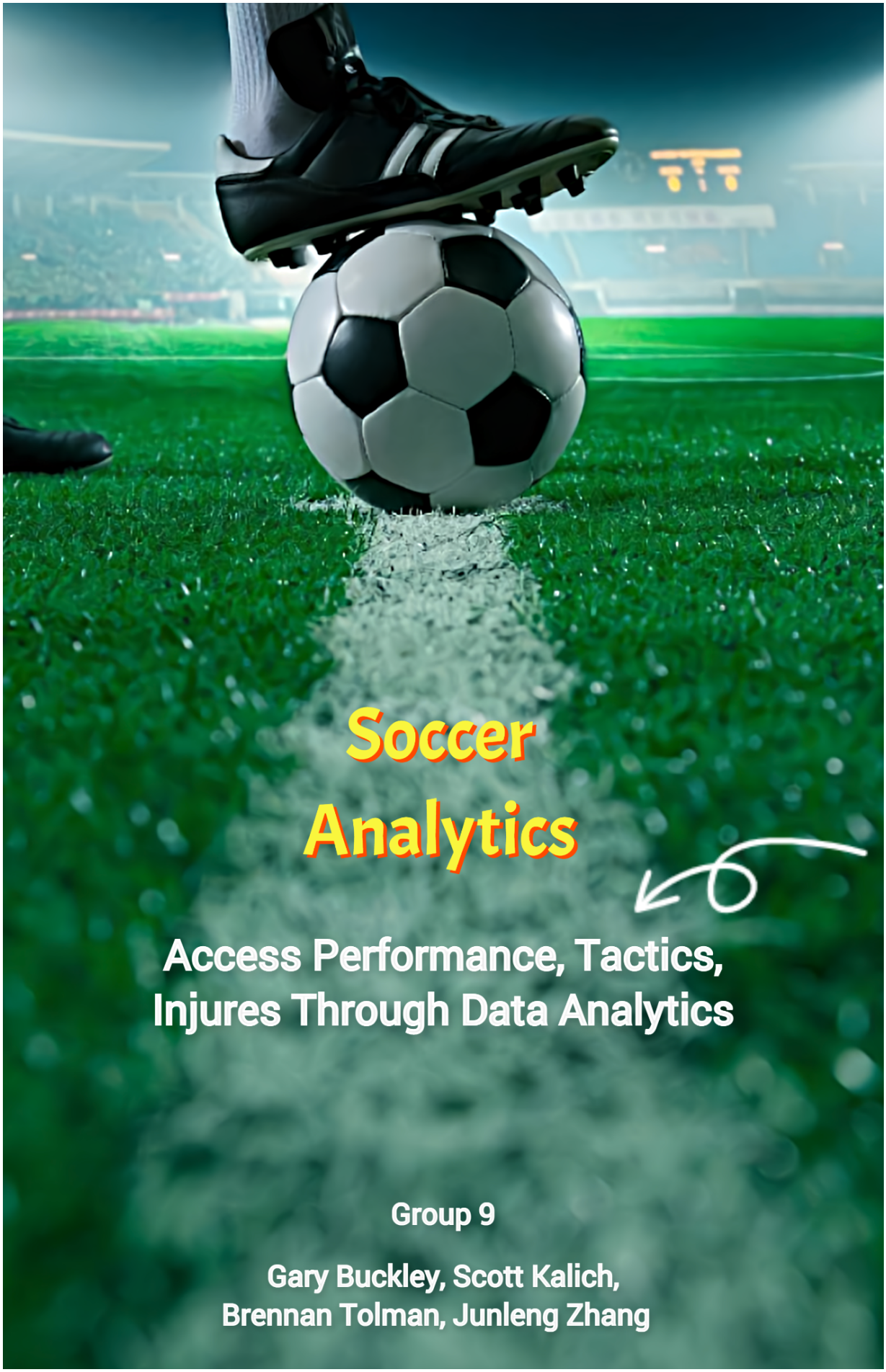
Soccer
Analytics
Access Performance, Tactics,
Injures Through Data Analytics
Group 9
Gary Buckley, Scott Kalich,
Brennan Tolman, Junleng Zhang

EXECUTIVE SUMMARY

There are 352 million people in the world participating in soccer, and the number of spectators in soccer has reached 3.6 billion. Almost one out of every two people in the world is a soccer fan. Judging from the sports events held around the globe, the FIFA World Cup is also the sport that garners the most attention worldwide. The Professional Soccer Organization serves as a preeminent association for soccer team tactics and training. In order to promote and develop soccer culture and create greater economic value, our team provides scientific, evidence-based pictures of how soccer players generate shots in relation to their location on the field. We also consider passing (pass length, pass type, etc.) and set pieces (out-swing, in-swing, short corners) and integrate these events with location analysis. The results of this integrated data analysis are highly intuitive and have great reference value for executive team members in designing and implementing competitive tactics based on questions that these reports address.

Our raw data comes from two flat file sources. They feature information on the player events (start, pass, touch, clearance, etc.) and player location indicated by using an X, Y axis coordinate system along with game dates and team indicators. Our research team ingested this data and established a well-organized data warehouse to better analyze the data relations and highlight the results for executive members.

This report gives a general idea about the physical locations from which players generate shots. At the same time, the report also derives more questions about changing tactics and game safety. The research team used Talend Open Studio and PostgreSQL to build the data warehouse and Metabase to display the research results. The attributes that can be exploited are extracted and transformed from the source data through the ETL process, and added to the created dimension and fact tables for analysis.

This report contains future reports as well as potential analysis of future data. With the introduction of new data, the data can be aggregated to a higher granularity. Through the metabase dashboard, you can analyze the trend of more granular as well as aggregated attributes. At the same time, the data warehouse retains its plasticity, and more dimension tables and fact tables can be added to track the performance of specific teams based on specific problems.

BACKGROUND

Vision & Objectives

It's a joy to watch your favorite team score a goal up close. Goals are one of the biggest things to watch in soccer, but the difficulty of scoring cannot be underestimated. Soccer players maintain high-intensity and highly targeted training all the time to win the cheers of the audience when they score a goal. For the entire club organization, a more effective team strategy means a more robust way to win games. Building a data warehouse by integrating valid matches and shot positions allows us to analyze and classify the zone where effective shots are generated and systematically determine the relationship between shot positions and goal rates. The data warehouse in this report is designed to explore the relationship between shots, zones, and passes.

The main questions we intend to answer are as follows:

- Shots from which zone on the field most often lead to goals?
- Where are most shots generated from?
- When do the most shots happen?

Answers to these questions will provide reference for the soccer organization in modifying team tactics and making constructive suggestions for the organization.

The data warehouse can also bring derived business benefits to the organization. From the perspective of an area with a relatively large number of goals, spectators in this area can get a better viewing experience. This conclusion puts forward a more reasonable distribution method of the ticket price from a scientific point of view. At the same time, using the same stadium auditorium space to rearrange the spatial position of the seats can maximize the benefits. For the participating players, the coach can strengthen the regional pertinence of their training while conducting comprehensive training to achieve a multiplier effect. From a financial point of view, more targeted training can reduce costs.

Products & Services

- postgreSQL 12
- pgAdmin 6.4
- Windows 10 Pro 64-bit Operating System
- NordLayer business VPN
- Talend Open Studio for Data Integration 8.0.1
- Metabase BI Server

- Soccer Organization Data Set
- Microsoft 365 Business Premium
- R Version 4.1.3
- RStudio 2022.02.1+461

DW & Analytics

To serve the organization's vision and objectives, a data warehouse is made to have better access to the desired information. Since a large number of shots come from different locations, the organization needs to classify these events in a timely manner, and obtain various characteristics from them to provide support for strategic, data-driven decision-making. Establishing a proper data warehouse can facilitate analytic scenarios such as location-event association and the aggregation of data within an organization. In order to achieve organizational goals, it is necessary to access the original database tables while also classifying and analyzing its data. However, when the form attributes are complex and transaction updates are required at any time, it is difficult to lock multiple tables at the same time, which will increase the delay of complex queries. On the other hand, if multiple tables are locked, the transaction of database form update will be blocked, resulting in business delay or even interruption. Building a data warehouse supports the collection of data from multiple operating systems into a data warehouse. This way, the data can be better correlated and analyzed, resulting in greater value for the team.

Relevant Context

Soccer is about aesthetics and passion. Our organization is committed to developing and promoting soccer culture. The intense rivalry and variety of games are one of the reasons why soccer continues to attract people to the game. Our organization constantly develops players and creates and improves team tactics, so as to present a more attractive game for the audience. At the same time, we also attach great importance to creating a safer environment for all players. Through data analysis, we hope to systematically solve security problems from a scientific point of view and derive more spectator-friendly and competitive tactics, making teams' efforts more effective at igniting the enthusiasm of spectators, creating better quality entertainment, and providing economic benefits for the society and people within.

DIMENSIONAL MODEL REQUIREMENTS

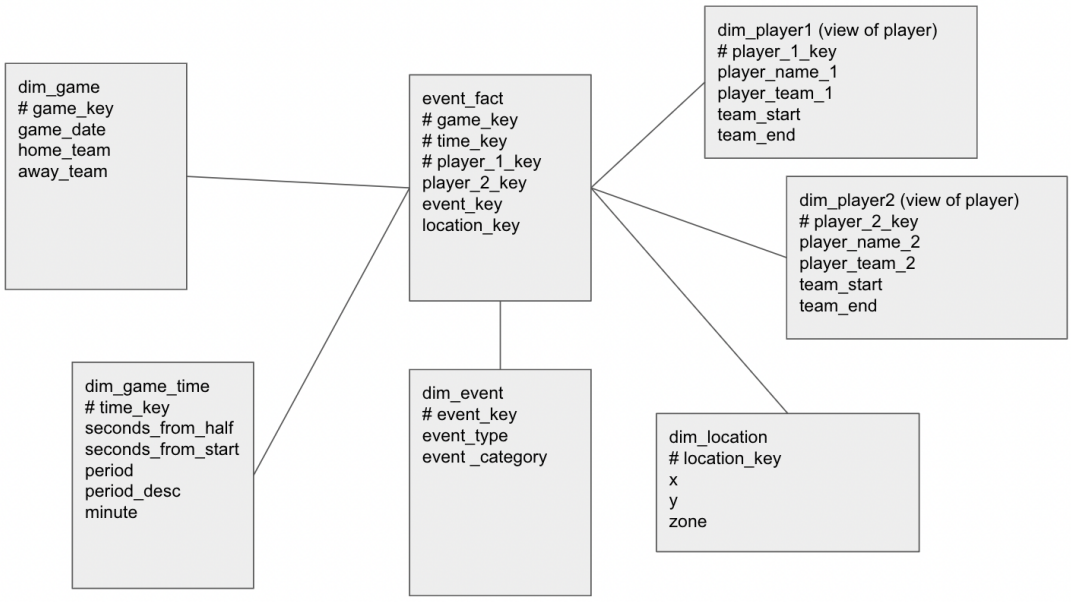

Figure 1

Summary

The iteration of the dimensional model shown above was chosen because it is the most basic design that fulfills the requirements. The game dimension was created in order to store the date in which the game was played along with the home team and away team. With these three attributes, we are able to uniquely identify games that are played. In order to track the time that things happen within the game, we have a dimension called game time which will store the second, minute, and period when events occur. The event dimension captures what type of event happens and categorizes them for seamless aggregation within queries that will not slow down response time. This design features a role-playing dimension for the players tracked. This was done in order to capture things such as assists within the model where two players will be attached to a single event. Finally, we have the location dimension which captures where events take place on the field. This gets aggregated into discrete zones before going into the dimensional model for ease of access in queries. Note that there is no dimension specified currently for date. While we believe that this might be an area of interest later, our current requirements do not necessitate any date based dimensions. All the dimensions are attached through warehouse keys to the event_fact table which allows users to answer the questions listed in the requirements using relatively simple queries. Because we are identifying the occurrence of events in this scenario rather than measuring something like sales, the event_fact

table itself is a factless fact table, and only contains the warehouse keys from the dimension tables. It should be noted that while the event_fact table does contain warehouse keys from each dimension table, only game_key, time_key, and player_key make up the primary key of the table since those three keys alone ensure uniqueness across all records in the fact table. With that in mind, all the dimensions that do not contribute to the primary key (dim_event, dim_location, and dim_player2) are technically analysis dimensions.

Requirements Status

Currently, our requirements are fairly straightforward. We are attempting to help coaching staff understand where shots on goal are generated from as well as identify the best locations to shoot from. Gaining a better understanding of these events will educate coaching staff on how to position valuable players, generate more offensive pressure, and win more games. In the future, there could be compelling reasons to add dimensions related to teams or other factors so that coaches can more easily query how their specific team is performing and compare their statistics against other teams, but for the present our focus is strictly on analyzing shots and goals in terms of location and time.

As it stands, the questions covered by the requirements include:

- Where are most shots generated from?
- Where are most goals generated from?
- When do the most shots happen?

Future requirements that may be implemented are:

- What events lead to injuries?
- How do substitutes change offensive generation?
- What length of pass sequences lead to goals?
- Are out-swinging, in-swinging, or short corners most effective?
- How do we limit shots in high-conversion zones on defense?

Bus Matrix

	Dimensions						
Business Process / Match Strategies	*Location on field*	*Time in Game*	*Player*	*Game*	*Event Type*	*Date*	Notes
Shooting	cartesian product of coordinates						**Team will be captured in player dimension. Type 2 changes**
Taking Shots	x	x	x	x	x		Where should we try to take shots from?
Defending Shots	x	x	x	x	x		How do we limit shots in high-conversion zones?
Scoring Goals	x	x	x	x	x		Where are most goals scored from?
Assisting Goals	x	x	x	x	x		How does the pass location influence goal percentage?
Set Pieces							
Corners	x	x	x	x	x		Are out-swinging, in-swinging, or short corners most effective?
Free Kicks	x	x	x	x	x		When should we shoot vs. cross a free kick?
Player Management							**Include date to see if injuries happen more often on short rest**
Injuries	x	x	x	x	x	x	What conditions lead to injuries more often?
Subbing		x	x	x	x		How and when should we use our substitutes?
Passing			Role-playing for 2-player events				**use views of role-playing player dimension for passer/receiver**
Offense	x	x	x	x	x		What length of pass sequences lead to goals?

DATA WAREHOUSE ARCHITECTURE

Data Sources

The data being used to develop our data warehouse comes from two csv files, both of which contain different but related data. The first file is composed of data related to events during a soccer game which, in this case, is any major event that takes place during a soccer match, including various shots, passes, and penalties. Some examples of events are goalkeeper kicks, offsides penalties, passes, red cards, deflections, and so on. The csv file itself contains the following event related fields:

- game_date_gen (date): the date the match took place in the format YYYY-MM-DD (e.g., '2014-06-14')
- v_team_gen (string): designated code for the visiting team (e.g., 'team18')
- h_team_gen (string): the name of the home team (e.g., 'team18')
- period_desc (string): textual indicator of which half the event took place in; only takes on two discrete values ("First Half" or "Second Half")
- time_from_zero: the time in seconds from the start of the current half rounded to the nearest hundredth (e.g., 31.40)
- event_type (string): type descriptor of the event that took place (e.g., 'Goalkeeper Kick')
- player_name_1_gen (string): designated code for the primary player that initiated the event (e.g., 'player58')
- player_name_2_gen (string): designated code fo the secondary player that participated in the event (e.g., 'player58')
- player_1_team_gen: designated code for the team that the primary player (player_name_1_gen) belongs to
- player_2_team_gen: designated code for the team that the secondary player (player_2_name_gen) belongs to

Each row of the event dataset describes one event involving at most two players that took place at a specified time during a game. Below is a sample of the first five rows of the event data:

game_date_gen	v_team_gen	h_team_gen	period_desc	time_from_zero	event_type	player_name_1_gen	player_name_2_gen	player_1_team_gen	player_2_team_gen
2014-06-14	team18	team2	First Half	0	Start Of Half	player58	NULL	team18	NULL
2014-06-14	team18	team2	First Half	0.5	Pass	player240	NULL	team18	NULL
2014-06-14	team18	team2	First Half	2.1	Touch	player246	NULL	team18	NULL
2014-06-14	team18	team2	First Half	3	Pass	player246	NULL	team18	NULL
2014-06-14	team18	team2	First Half	4.9	Touch	player46	NULL	team18	NULL

Figure 2

As you'll notice, all the player_name_2_gen values in this sample are NULL. Since many events only involve one player, most of the player_name_2_gen values throughout the dataset are NULL.

The second file serving as a data source for the data warehouse is the location dataset. This file contains information about the location of each player on the field for every second throughout a given soccer game. In this dataset, the soccer field is represented by a 2 dimensional coordinate plane and the location of a player at a given time during a game is identified by an x/y value pair that corresponds to a point on that 2 dimensional coordinate plane. A description of the fields in this dataset and a sample are as follows:

- game_date_gen (date): the date the match took place in the format YYYY-MM-DD (e.g., '2014-06-14')
- v_team_gen (string): designated code for the visiting team (e.g., 'team18')
- h_team_gen (string): designated code for the home team (e.g., 'team18')
- team_gen (string): designated code for the team the player belongs to (e.g., 'team18')
- player_name_gen (string): designated code for the player (e.g., 'player58')
- period (int): number identifying which half is associated with given row (e.g., 1 or 2)
- time_from_zero (int): the time in seconds from the start of the current half rounded to the nearest whole number (e.g., 5)
- x (numeric): the x coordinate on the 2 dimensional coordinate plane
- y (numeric): the y coordinate on the 2 dimensional coordinate plane

game_date_gen	v_team_gen	h_team_gen	team_gen	player_name_gen	period	time_from_zero	x	y
2014-06-14	team18	team2	team2	player34	1	980	-33.13999939	40.70999908
2014-06-14	team18	team2	team2	player34	1	981	-31.14999962	40.68999863
2014-06-14	team18	team2	team2	player34	1	982	-29.42000008	40.66999817
2014-06-14	team18	team2	team2	player34	1	983	-27.51000023	40.70000076
2014-06-14	team18	team2	team2	player34	1	984	-25.7000007	40.93999863

Figure 3

In order to utilize the location dataset, it must be joined with the event dataset. The two datasets can be joined via game_date gen, v_team_gen, h_team_gen, player_name_1_gen or player_name_gen, period, and time_from_zero (after rounding the time_from_zero in the event dataset).

Data Pipelines/ETL Processes

As mentioned previously, the raw soccer data used for our data warehouse initially resided in two flat, delimited files (event and location). The end goal of our ETL process was to transform the data from the flat files into a format that could then be inserted into various data warehouse tables in a Postgres database (see section “Dimensional Model” for table names/details). In order to accomplish this, we had to break out our ETL process into three overarching steps:

1. Initial data transformation and loading into the staging area
2. Create and load the presentation area
3. Create and load the fact table

Step one was completed using an R script and the PGAdmin interface; steps two and three were completed exclusively using Talend. This section will describe how these steps were carried out while subsequent sections will go into more detail on the outcomes of the ETL process.

The first step in our ETL process is the initial data cleansing and transformation. As mentioned previously, this was done using an R script (see ‘Appendix B’ for script) that performed various transformations on the raw data from the two flat files. For both files, certain columns were given more understandable names that would make it easier for developers and end users to work with the data (e.g., ‘time_from_zero’ to ‘time_from_half,’ ‘player_name_gen’ to ‘player_name,’ etc.). Some additional columns were also derived from existing columns and are described as follows:

- minute (int): derived from ‘time_from_zero’ column and gives the rounded game minute instead of second
- time_from_start (int): derived from ‘time_from_zero’ in conjunction with ‘period’/’period_desc’ columns and gives the rounded total time from the beginning of the game instead of from the beginning of the current half
- zone (int): derived from the x/y coordinates in the location dataset and identifies which of the 18 zones of a soccer pitch a given x/y coordinate pair fall into (see ‘Appendix A’ for a map of the zones).
- event_category (string): developed by our own soccer SME to group, classify, and give a hierarchical structure to the many different event types that take place in a soccer match

Note that the event_category column was added by joining the event dataset to a mapping of event types to event categories (developed by our team SME).

Lastly, the R script performed some relatively simple transformations on the following columns that included rounding the ‘time_from_zero’ column, ensuring that the ‘period_desc’ values were changed from string descriptors to integers (‘First Half’ to 1, ‘Second Half’ to 2), and dropping erroneous rows in the location dataset where the x/y coordinates were not possible on a regulation-size soccer pitch. Finally, the two datasets were exported as csv files and

subsequently uploaded into the ‘staging’ schema of our Postgres database using the PGAdmin user interface (see section ‘Staging Area’ for more details).

After uploading the transformed data to the staging area, we shift to using Talend for the remainder of our ETL process. The first task to be completed using Talend is the creation of the dimension and fact tables that will constitute our data warehouse. The following image displays this process.

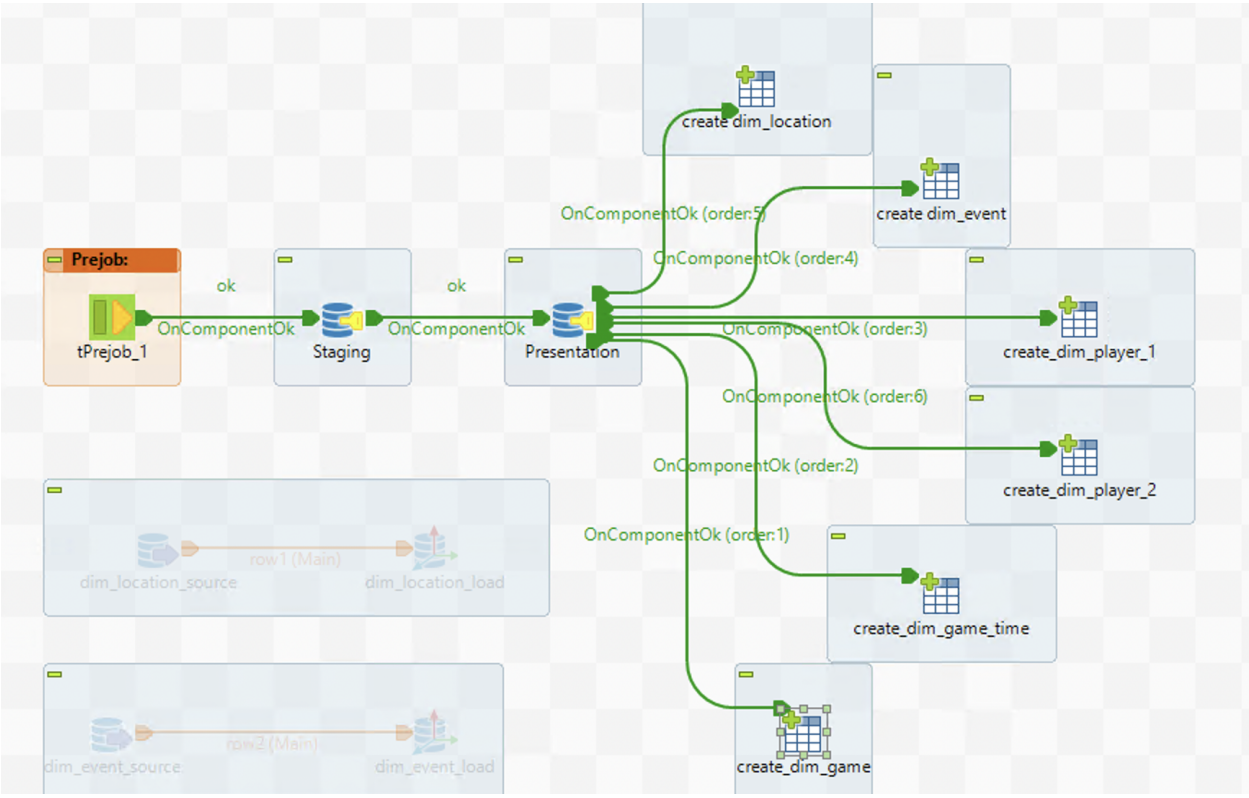

Figure 4

In Talend, we first connect to the relevant database and the ‘staging’ and ‘presentation’ schemas. After connecting, we move on to create each of the dimension tables. Each ‘create’ node shown above creates the pertinent dimension table in the presentation area, implementing the same schema/columns described in the dimensional model (see section ‘Dimensional Model Requirements’).

Following the creation of the dimension tables in the presentation area, we load the tables with the available data.

Figure 5

The loading is performed using DBInput nodes along with DBSCD nodes. Each input node queries either the event or location table in the staging area, extracting the data needed for the dimension table, and then passes the data to the connected DBSCD node. The DBSCD nodes perform two main functions: insert the data into the dimension table and manage any slowly changing dimensions (SCD). The insertion of the data is straightforward and really only requires the correct target table to be selected and verification that the schema is set up properly. When it comes to managing slowly changing dimensions, the DBSCD node allows us to set up surrogate keys, manage different versions, and identify which SCD type each column falls under (Type 0, Type 1, Type 2, or Type 3). And example of the DBSCD node's ability to manage slowly changing dimensions is shown below:

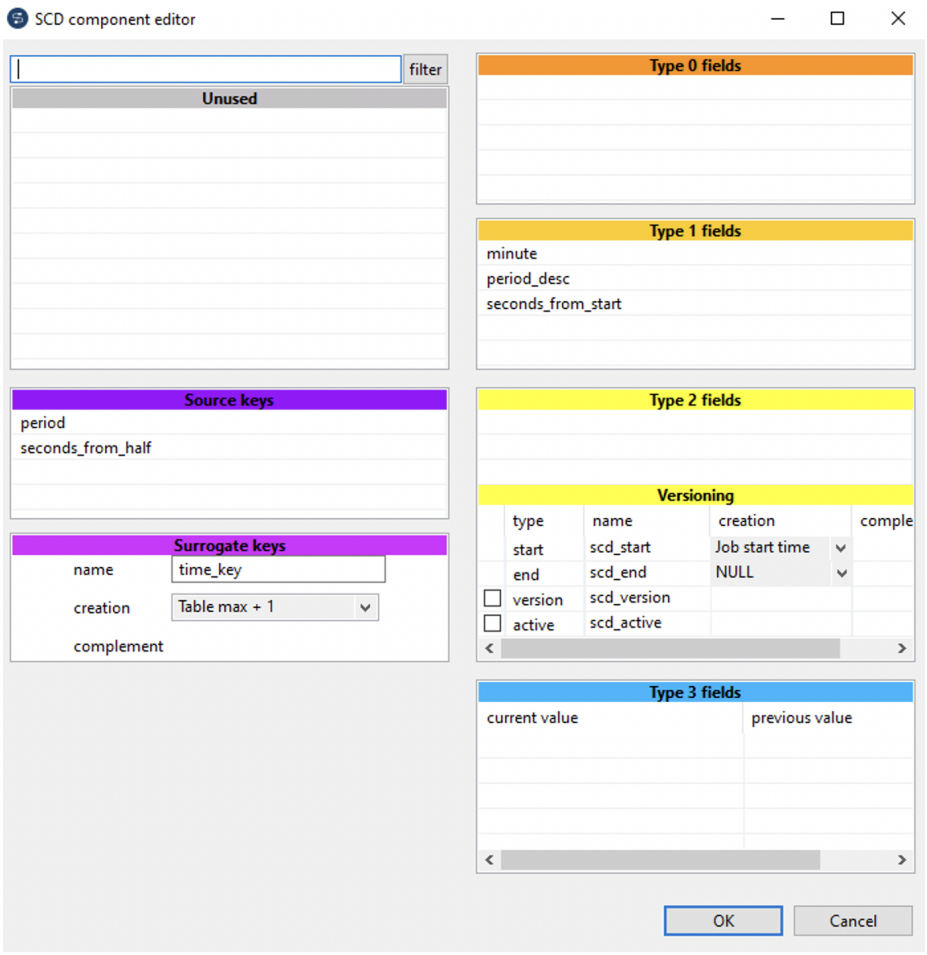

Figure 6

The correct usage of the DBSCD node is an important step in the process because it ensures that any future changes to the dimension tables are handled properly and that queries involving those same tables will still return correct results (see section ‘Presentation Area’).

The final step in the ETL process is to create and load the fact table, which is the core of our dimensional model and data warehouse. As mentioned previously, Talend is used to complete this step.

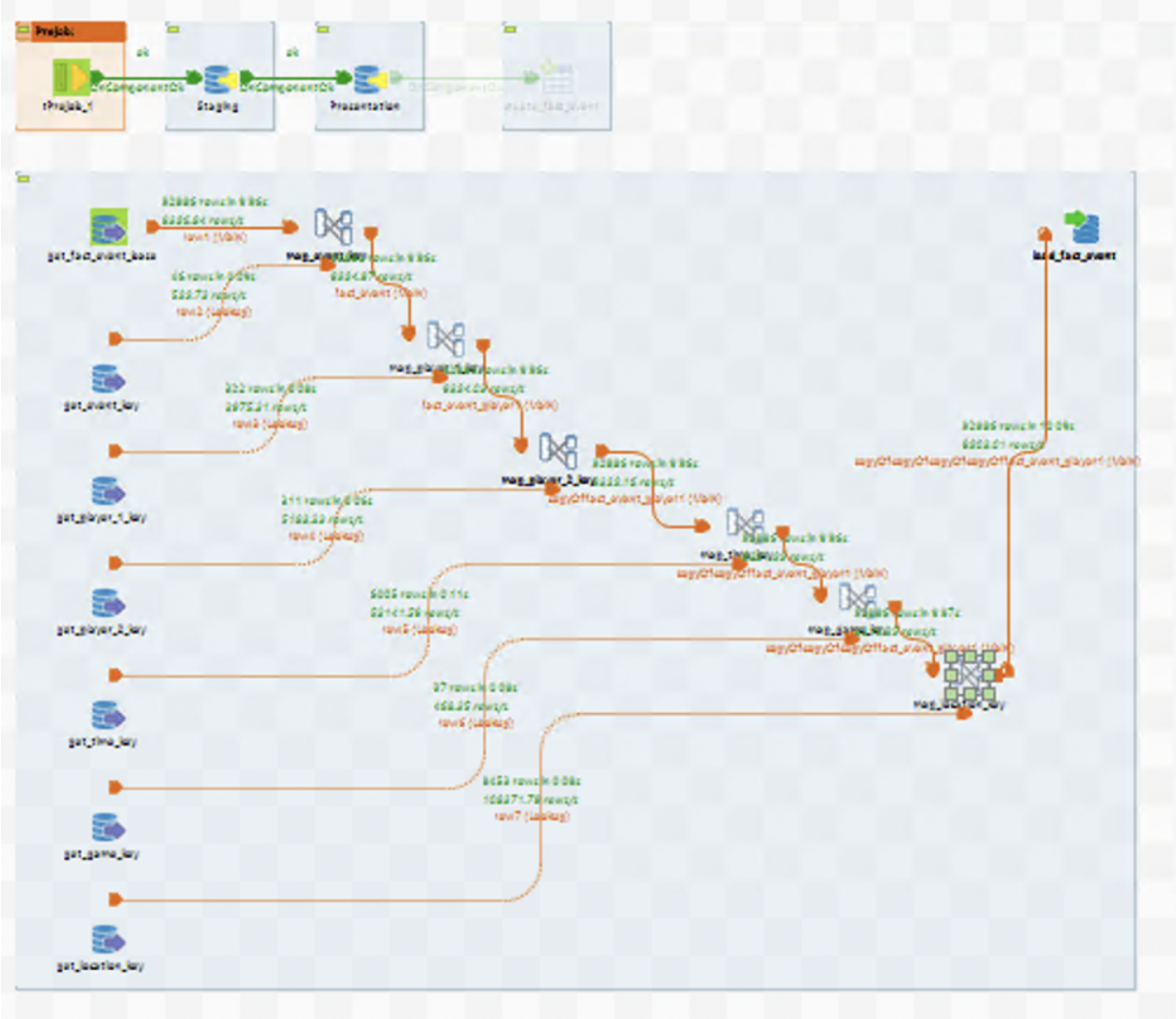

Figure 7

The creation of the fact table is straightforward and only requires a connection to the database and schemas and a ‘create’ node where we specify the schema of the fact table in accordance with our dimensional model.

Loading the fact table in Talend is more involved. In order to correctly populate the fact table, we need to collect the correct surrogate keys from each dimension table for each record in the fact table. As described in the section ‘Dimensional Model Requirements,’ our fact table is a factless fact table and does not have any attributes of its own. Therefore, we first query our staging area along with each of the dimension tables using DBInput nodes. Then in a stepwise fashion the staging area data is joined/mapped to each of the dimension table input nodes, the surrogate keys from the dimension tables are kept, and the columns from the staging table are dropped. The end result is a dataset solely composed of the warehouse/surrogate keys from each of the dimension tables. At this point, the data is loaded into the fact table using a DBOutput node and the ETL process is complete.

Staging Area

In the previous section, the transformation process and the loading of the resulting datasets into the staging area were described. In this section, we will provide more detail on the tables found in the staging area of the database. Recall that we have two primary data sources in the form of flat files; one contains information regarding events during soccer matches while the other contains information regarding the location of each player for each second of those same soccer matches. After both datasets have undergone the appropriate transformations, the resulting two datasets are loaded into the staging area. Therefore, in the staging area we have two tables that are the result of the transformation process: event and location. The tables below give a sample of each of the tables' data and provide more detail on the columns and data types.

Event Table

game_date	a_team	h_team	period_desc	time_from_half	event_type	player_name_1	player_name_2	player_team_1	player_team_2
2014-06-14	team18	team2	First Half	0	Start of Half	player58	NULL	team18	NULL
2014-06-14	team18	team2	First Half	0	Pass	player240	NULL	team18	NULL
2014-06-14	team18	team2	First Half	2	Touch	player246	NULL	team18	NULL
2014-06-14	team18	team2	First Half	3	Pass	player246	NULL	team18	NULL
2014-06-14	team18	team2	First Half	5	Touch	player46	NULL	team18	NULL
2014-06-14	team18	team2	First Half	6	Pass	player46	NULL	team18	NULL
2014-06-14	team18	team2	First Half	9	Pass	player60	NULL	team18	NULL
2014-06-14	team18	team2	First Half	11	Clearance	player283	NULL	team18	NULL
2014-06-14	team18	team2	First Half	14	Touch	player158	NULL	team2	NULL
2014-06-14	team18	team2	First Half	15	Tackle	player158	player58	team2	team18

Figure 8

Location Table

game_date	a_team	h_team	team	player_name	period	time_from_half	x	y	zone	time_from_start	period_desc
2014-06-14	team18	team2	team2	player322	2	1478	0	1	10	4178	Second Half
2014-06-14	team18	team2	team2	player322	2	1479	0	3	7	4179	Second Half
2014-06-14	team18	team2	team2	player322	2	1480	-1	8	7	4180	Second Half
2014-06-14	team18	team2	team2	player322	2	1481	-2	13	7	4181	Second Half

2014-06-14	team18	team2	team2	player322	2	1482	-3	18	7	4182	Second Half
2014-06-14	team18	team2	team2	player322	2	1483	-4	23	7	4183	Second Half
2014-06-14	team18	team2	team2	player322	2	1484	-5	27	8	4184	Second Half
2014-06-14	team18	team2	team2	player322	2	1485	-7	32	8	4185	Second Half
2014-06-14	team18	team2	team2	player322	2	1486	-8	35	8	4186	Second Half
2014-06-14	team18	team2	team2	player322	2	1487	-10	39	8	4187	Second Half

Figure 9

If we compare these data samples to those found in the section ‘Data Sources,’ it becomes apparent why the various transformations were made. Some changes, such as changing column names made the columns more understandable and other changes such as rounding the time fields made it possible to join the two dataset via game_date, a_team, h_team, player_name, period, and time_from_half.

Presentation Area

Similar to the ‘Staging Area’ section, the purpose of this section is to provide more detail on the presentation area of the data warehouse. The Talend jobs described in the section ‘Data Pipelines/ETL Processes’ result in the creation and population of each of the tables and columns described in the section ‘Dimensional Model Requirements.’ To avoid redundancy, please refer to that section on details such as table and column names found in the presentation area.

The following table describes other aspects of the presentation area such as the resultant data type and SCD types:

Table	Column	Data Type	SCD Type	Sample Value
dim_game	game_date	DATE	Source Key	2014-06-14
	home_team	VARCHAR	Source Key	‘team2’
	away_team	VARCHAR	Source Key	‘team18’
dim_game_time	seconds_from_half	INTEGER	Source Key	75
	seconds_from_start	INTEGER	Type 1	1301
	period	INTEGER	Source Key	1
	period_desc	VARCHAR	Type 1	‘First Half’

	minute	INTEGER	Type 1	25
dim_event	event_type	VARCHAR	Source Key	‘Touch’
	event_category	VARCHAR	Type 1	‘Dribble’
dim_location	x	INTEGER	Source Key	43
	y	INTEGER	Source Key	38
	zone	TEXT	Type 1	17
dim_player	player_name	VARCHAR	Source Key	‘player140’
	player_team	VARCHAR	Type 2	‘team16’
	team_start	TIMESTAMP	NA	2022-04-20 03:34:55:869
	team_end	TIMESTAMP	NA	NULL

Figure 10

You’ll notice that the majority of the SCD are of Type 1. This is because nearly all of our dimension attributes do not need to be tracked over time. For example, if a value in event_category were to change, that change would apply to both past and future records. The only exception to this is the player_team attribute on dim_player; if a player changes teams at any point in time, we want to be able to make that change to the dimension values without affecting any teams’ past or future reporting accuracy.

The event_fact table has been omitted from this visual because it simply consists of the warehouse keys from each of the dimension tables. Also, recall that dim_player_1 and dim_player_2 are both role playing dimensions based on the dimension dim_player, which is displayed in the table above.

Security Configuration

The goal of our security configurations is to provide as much data protection as possible while still giving users all the permissions necessary to analyze game events. The potential roles involved in this scenario include an admin, coach/executives, and players. The admin will have all privileges, so we will not go into details on that role. Since there seems to be little reason to believe that soccer players are proficient in SQL, we made the assumption that players would not need access to the data warehouse in Postgres. However, we did assume that at least some of the coaching/executive staff would have some basic, working knowledge of SQL, and therefore would need access to the presentation area in the data warehouse. The following query creates the ‘admin’ and ‘coach’ roles and grants the appropriate permissions in Postgres:

```
CREATE ROLE admin;
GRANT ALL PRIVILEGES ON DATABASE soccer TO admin;
GRANT ALL PRIVILEGES ON SCHEMA soccer_staging, soccer_presentation TO admin;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA soccer_staging, soccer_presentation TO admin;

CREATE ROLE coach;
GRANT SELECT, INSERT, UPDATE ON ALL TABLES IN SCHEMA soccer_presentation TO coach;
```

Figure 11

In Metabase, the goal of our security settings will be to allow all players to see team dashboards, but not any dashboards where metrics are broken out by player. This is best accomplished using collections and permissions. A collection where coaches and admin have the ‘Curate’ permission and players have the ‘View’ permission will be dedicated to team analytics. The following pictures display the permissions associated with the general team ‘Soccer Collection’ and demonstrate a sample user, ‘Bob,’ who has been given the role ‘Player’:

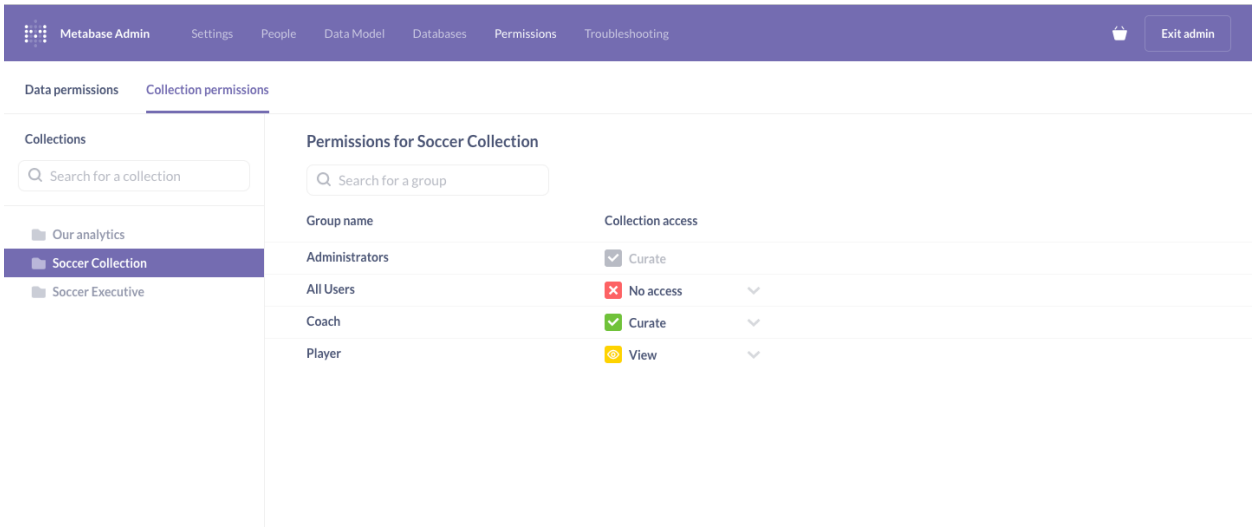

Figure 12

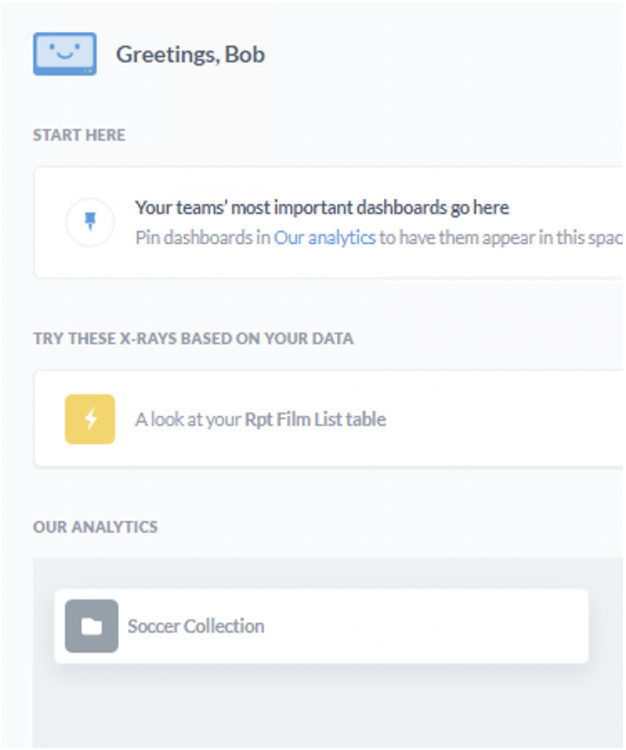

Figure 13

The next images display the permissions for the ‘Soccer Executive’ collection and demonstrate the access of a sample user, ‘Mary,’ who has been given the role ‘Coach’:

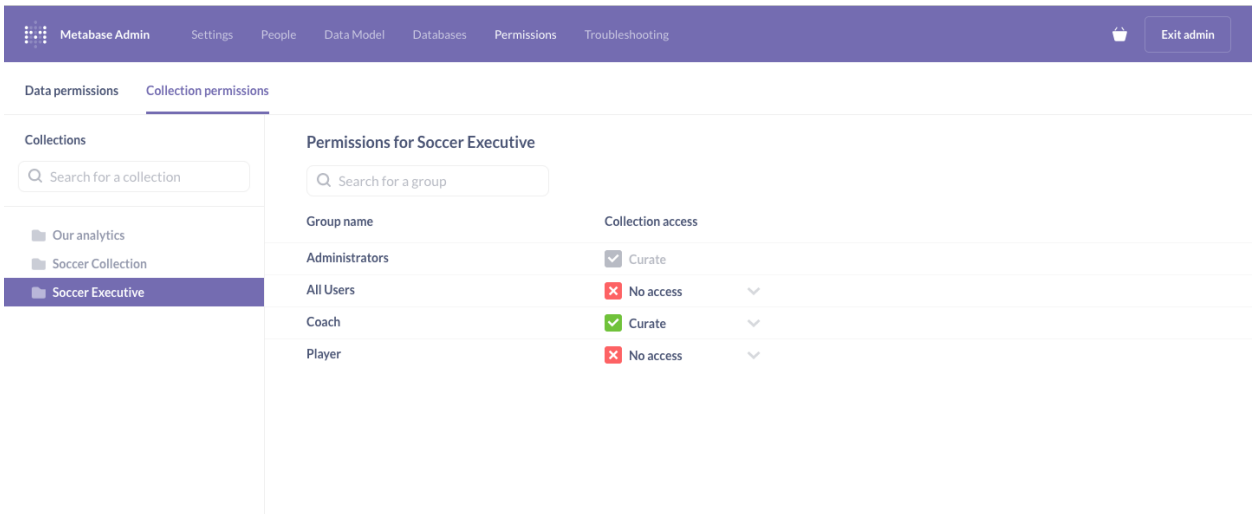

Figure 14

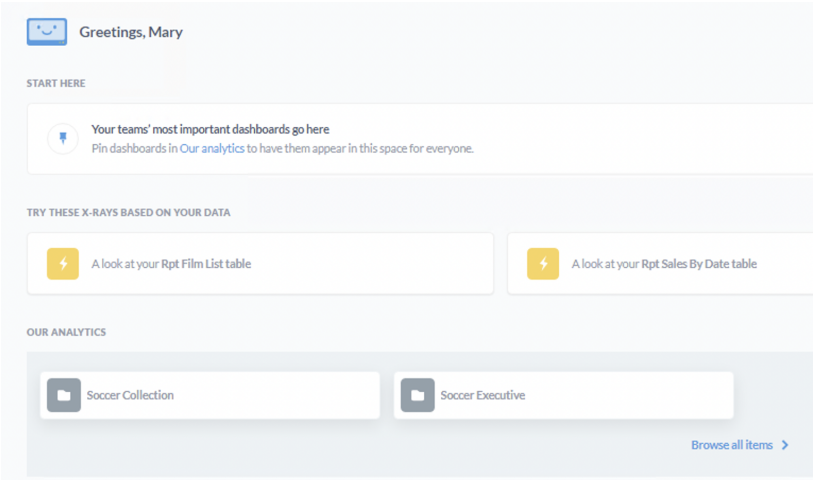

Figure 15

Bob is able to view the general 'Soccer Collection,' but has no access to 'Soccer Executive,' which is why only 'Soccer Collection' appears when he logs in. Mary is a coach so has 'Curate' access to both 'Soccer Collection' and 'Soccer Executive'; when she logs in she is able to see both of these collections. Overall, these security configurations follow the principle of least-privilege in giving the coaches/executives access to data regarding all their players while restricting the players' access to only team level analysis.

Reports/Dashboards

The BI tool used for our reporting and visualizations is Metabase, where we have two collections and two dashboards: one for general team analysis and another specifically for coaches/executives. The following image displays the general team dashboard in the 'Soccer Collection':

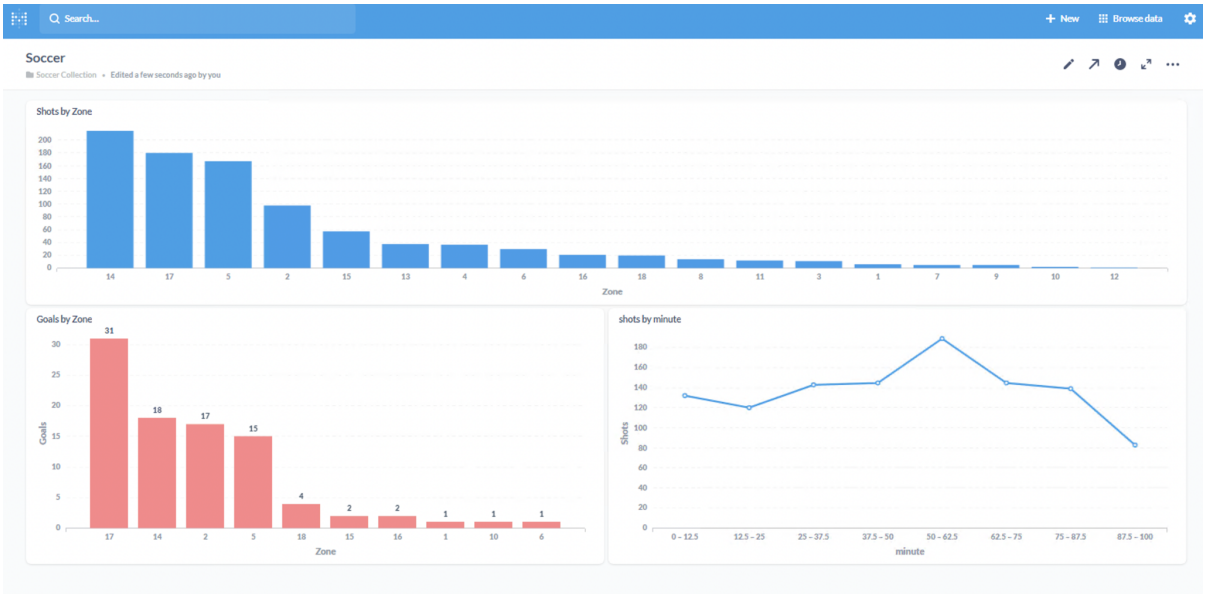

Figure 16

In this dashboard, you'll find three separate visualizations: Shots by Zone, Goals by Zone, and Shots by Minute. Notice that each of these visualizations helps to answer one of the questions originally asked as part of the project's requirements:

- Where are most shots generated from?
- Where are most goals generated from?
- When do the most shots happen?

The first question is answered by the graph titled "Shots by Zone." This underlying graph for this graph is the result of a join between the event_fact table, dim_event, and dim_location and is filtered to return only the events where the event_category is 'Shot.' The data is then grouped by zone and counted. The resulting bar chart makes it clear that most shots are taken from zone 14, but zones 17 and 5 are also popular areas to take shots from.

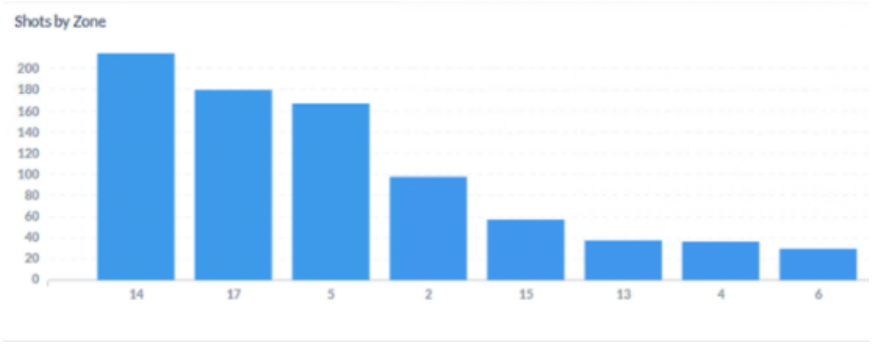

Figure 17

Secondly, the "Goals by Zone" visual helps us identify where the most goals come from. The data for this bar chart again consists of a join between event_fact, dim_event, and dim_location,

but this time only records where the event_category is ‘Goal’ are kept. Similar to the first question, the data is grouped by zone and counts for each zone are collected. From the bar chart, we can see that zone 17 has, by far, the most goals and that zones 14, 2, and 5 all have a similar number of goals.

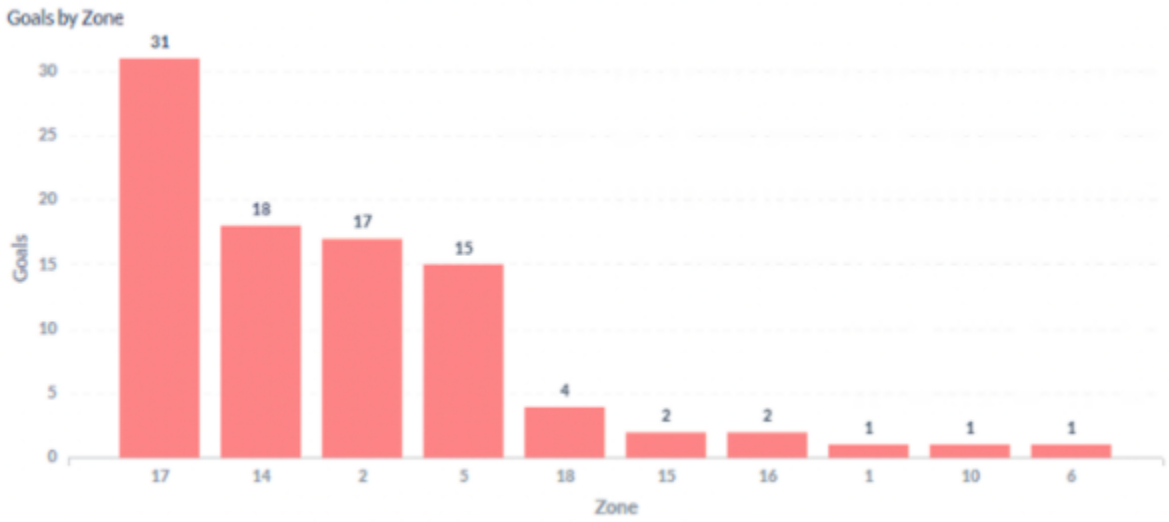

Figure 18

These results become more interesting when we combine them with the results from the first question. While a similar number of shots were taken from zones 14, 17, and 5, zone 17 has far more goals than any other zone, meaning that the effectiveness of shots taken from zone 17 are greater than either zone 14 or zone 5. Similarly, zone 2 came in fourth in terms of shots taken, but is only one goal away from being the zone with the second highest number of goals. While more analysis is required to attempt to establish any causality, this may suggest that shots taken from zones 17 and 2 are more likely to result in a goal than shots from other zones.

Lastly, the ‘Shots by Minute’ graph helps us identify when the most shots were taken. The data used to create this graph comes from a join between event_fact, dim_event, and dim_game_time and is filtered to only those events where the event category is ‘Shot.’ Metabase automatically bins the continuous column ‘minute’ from dim_game_time so that we can then count the number of shots, grouped by the binned minute. Keep in mind that a soccer game has two 45 minute halves, so this visual tells us that the number of shots stays relatively constant during the first half, peaks at the beginning of the second half (bin ‘50 - 62.5’) and then actually drops off towards the end of the game. This is interesting and may provide coaches with some information regarding player stamina during the second half.

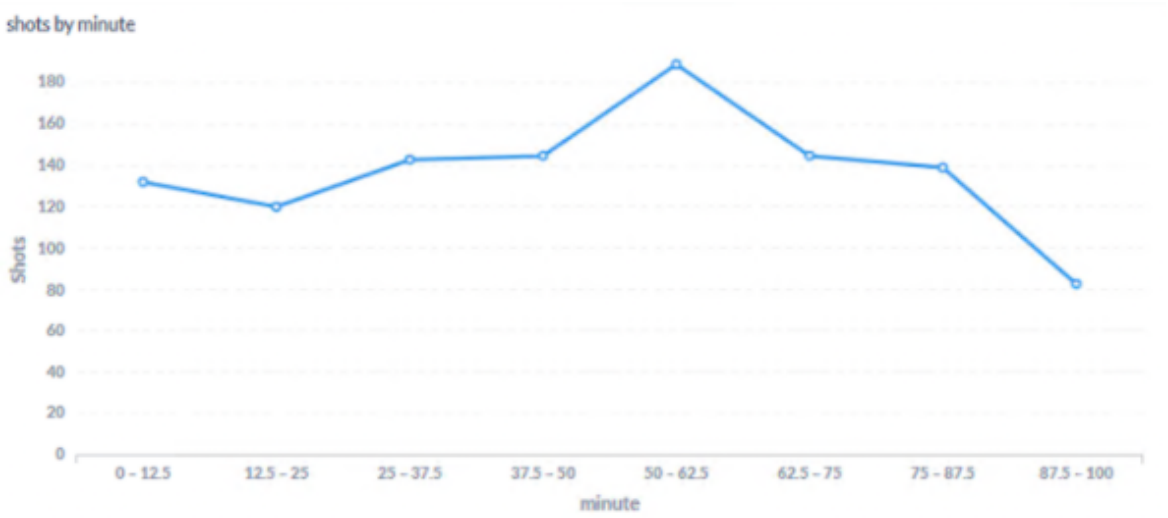

Figure 19

In the ‘Soccer Executive’ collection, we have one dashboard that helps us analyze shot and goal performance by player. Below, you’ll see total shots and goals broken out by player for all players across the league:

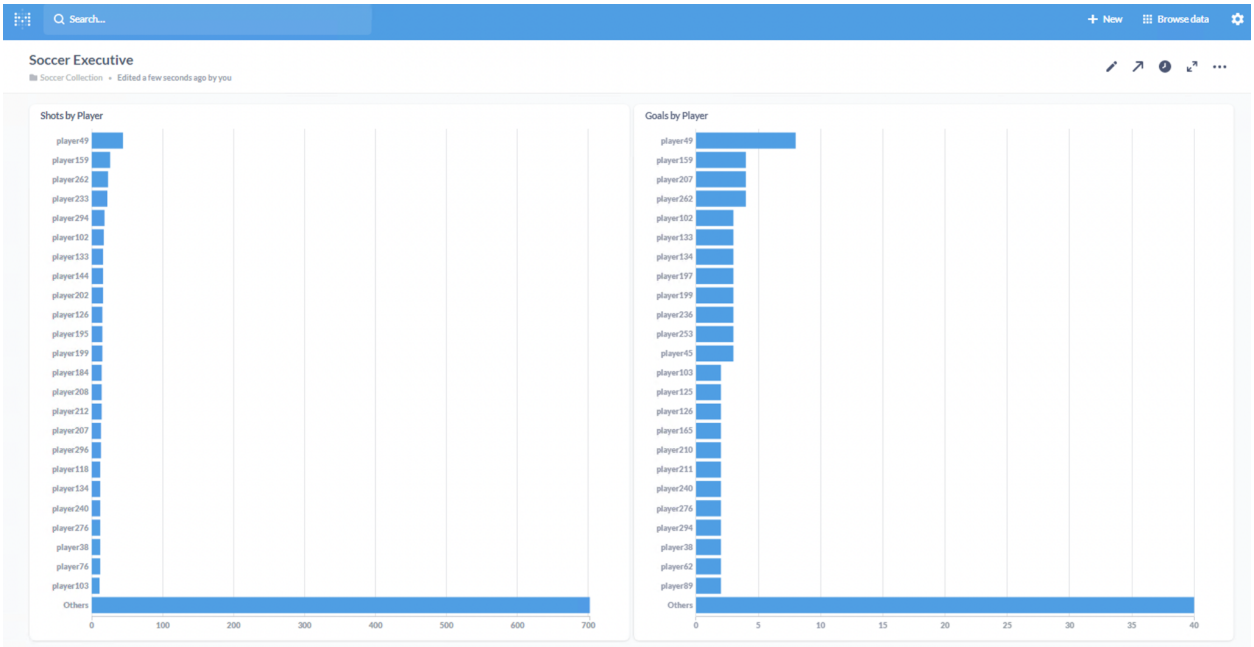

Figure 20

As mentioned previously, this dashboard is only available to those granted the ‘Coach’ role. This information would be highly valuable to any coach wanting to analyze players’ shot effectiveness. For example, it is clear to see that player49 takes the most shots and by far has the most goals and that much of the rank-ordering on the shots visual is the same on the goals visual. However, there are some players where this is not the case. Player 233 has the fourth

highest number of shots, but isn't even mentioned among the top scorers in the goals visual. On the other hand, player207 has the third highest number of goals, but takes a relatively small number of shots as compared to his peers, suggesting that giving player207 more shot opportunities may be worthwhile. This and all the previous visualizations provide the coach and players with valuable information to be used in making any decisions related to shots and goals. Overall, while there still may be many additional tables and analyses that would be highly beneficial (see section 'Future Reports, Dashboards, & Analyses'), the dashboard and visuals created from the current setup provide users with a thorough way to investigate aspects of any event during a game in terms of time and location.

FUTURE REPORTS, DASHBOARDS, & ANALYSES

While the current dimensional model used in this project is limited to queries based on the event_fact table, this model can easily be expanded to allow for a host of other features. For example, the addition of a date dimension would allow analysts to analyze events, such as shots or goals, for a particular section of the season or specifically evaluate events during the playoffs. Additionally, a role-playing dimension for the teams would allow users to easily track how specific teams play against each other over time or how teams perform at home versus away games. Adding these two dimensions opens the way for the creation of an aggregated fact table that groups data by team instead of by player, permitting team performance analysis over time. While querying a specific team's statistics in the current model is feasible, adding this aggregated fact table would drastically simplify the process.

Dashboards that expound on the current data warehouse as well as expansions, such as the added dimension and aggregated fact tables, would also play a critical role in any future reporting and analysis. Dashboards with aggregated statistics could be implemented to display general trends in both team and player performance, and others could be created to provide a more detailed view of where shots and goals come from on the field. New reports could also take advantage of the aggregated fact table, enabling teams to compare their offensive strengths and weaknesses to that of other teams.

Future analyses could include all the items mentioned above, as well as a variety of others. Again, while some of the above analysis, such as team aggregation, is possible given the current data warehouse, expanding our model in the aforementioned ways would make it easier

for users to perform related analysis and consequently make it more accessible to users who are less tech-savvy. It should be remembered that the usefulness of these suggested additions is dependent on how frequently users need such data, but, even though the availability of new data may change our priorities going forward, we believe that adding things such as date dimensions and team aggregations is a natural step in this data warehouse's progression.

REFERENCES

McKinley, Eliot, Eliot McKinley/, Kieran Doyle-Davis, Kieran Doyle-Davis/, Carl Carpenter, Carl Carpenter/, ASA Staff, and ASA Staff/. “American Soccer Analysis.” American Soccer Analysis, April 12, 2022. https://www.americansocceranalysis.com/.

APPENDIX A

Zone Map

Attacking this goal

Defending this goal

APPENDIX B

R script for data cleaning:

```
#load libraries
library(tidyverse)

#read in the event table
event <- read.table(
  "mls_event.tab",
  sep="\t", header=TRUE)

#read in the location table
location <- read.table(
  "mls_location.tab",
  sep='\t',
  header = TRUE
)

#read in the event map
event_map <- read.csv('event_map.csv')
event_map <- event_map %>% select(-count)

#store raw event and location tables
raw_event <- event
raw_location <- location


#clean the event dataset
event <- raw_event %>%
  mutate(time_from_zero = round(time_from_zero), #round timestamps to nearest second
         period = ifelse(period_desc == 'First Half',1,2), #convert halves
         time_from_start =  (period - 1)*45*60 + time_from_zero, #seconds from beginning of game
         minute = ceiling(time_from_start / 60)) %>% #convert time to minute
  rename(game_date = game_date_gen, #renaming all columns
         a_team = v_team_gen,
         h_team = h_team_gen,
         time_from_half = time_from_zero, #make time more clear
         player_name_1 = player_name_1_gen,
         player_name_2 =  player_name_2_gen,
```

```
        player_team_1 = player_1_team_gen,
        player_team_2 = player_2_team_gen) %>%
  left_join(event_map, by = 'event_type') #join category


#clean the location dataset
location <- raw_location %>%
  filter(x >= -60 & x <= 60) %>% #filter impossible x values
  filter(y >= 0 & y <= 70) %>% #filter impossible y values
  mutate(x_help = ifelse(x==60,59,x), #attacking endline belongs to final zones
        y_help = ifelse(y==70,69,y), #left sideline belongs to outside zones
        zone = 3*(floor(x_help/20)+3) + (floor(y_help/(70/3))+1) #map coordinates to zones
  ) %>%
  select(-x_help,-y_help) %>%  #remove help columns
  mutate(time_from_start = (period - 1)*45*60 + time_from_zero, #seconds from beginning of
game
        minute = ceiling(time_from_start / 60), #convert to minute
        period_desc = ifelse(period == 1,'First Half','Second Half'), #add half description
        x = round(x), #round coordinates so dimension doesn't explode
        y = round(y)) %>%
  rename(game_date = game_date_gen, #renaming all columns
        a_team = v_team_gen,
        h_team = h_team_gen,
        team = team_gen,
        player_name = player_name_gen,
        time_from_half = time_from_zero) #make time more clear

#write clean datasets
#event
write_csv(data.frame(event), 'event.csv')
#location
write_csv(data.frame(location), 'location.csv')
```

APPENDIX C

IS6480 Group 9 Soccer Presentation Video

Link to the presentation deck used in the video listed above.
Soccer Presentation Deck

Time Sheet

Team Member	Date	Hours	Description
Brennan	3/24/2022	0.5	Set up the draft for the main Summary report and then spent some time familiarizing myself with the datasets.
Scott	3/25/2022	2	Familiarizing with the datasets and refining requirements for the dimensional model
Brennan	3/26/2022	3	Started thinking things through with a first draft dimensional model. Spent most of my time doing more exploration on the dataset to look at feasibility of our posed requirements.
Brennan	3/30/2022	0.5	Initial group meeting
Gary	3/30/2022	0.5	Initial group meeting
Scott	3/30/2022	0.5	Initial group meeting
Zach	3/30/2022	0.5	Initial group meeting
Scott	4/1/2022	1.5	Meeting with Brennan to talk about our dimensional model and refining based on requirements
Gary	4/1/2022	0.5	Making first draft of BUS Matrix
Zach	4/1/2022	0.5	Formatting table of content
Gary	4/4/2022	0.5	EDA and making zone and location maps
Scott	4/7/2022	0.75	Meeting with group to talk about dimensional model and work to be done for next week's meeting before physical implementation
Gary	4/7/2022	0.75	Meeting with group to talk about dimensional model and work to be done for next week's meeting before physical implementation

Zach	4/7/2022	0.75	Meeting with group to talk about dimensional model and work to be done for next week's meeting before physical implementation
Gary	4/10/2022	0.5	Second draft of BUS matrix
Gary	4/12/2022	1.5	Cleaning datasets
Zach	4/13/2022	3	Background
Scott	4/13/2022	1	Writing up the dimensional model summary
Scott	4/14/2022	1	Working on the Bus matrix and write up
Zach	4/14/2022	1	Continuing of backgound writting
Gary	4/14/2022	1	data import
Gary	4/16/2022	2.5	building dim_location
Scott	4/18/2022	2	Meeting with Brennan and creating presentation deck
Brennan	4/18/2022	2	Meeting with Scott and working on the Data Sources portion of the Summary Report
Gary	4/18/2022	0.5	Watching video to build fact table
Scott	4/19/2022	1.5	Meeting and working on summary report
Gary	4/19/2022	2.5	Meet and load dimensions
Scott	4/19/2022	0.5	Adding to presentation deck
Zach	4/19/2022	0.5	Meeting with group
Zach	4/20/2022	2.5	Executive summary
Gary	4/20/2022	2.5	Building fact table. Making Metabase dashboard
Brennan	4/20/2022	1.5	Worked on the Data Pipeline/ETL Processes portion of the Summary Report
Scott	4/21/2022	1	Finishing presentation deck and making things look nice
Brennan	4/21/2022	2	Completed the Staging and Presentation portion of the Summary Report and am nearly done with Reports and Dashboards.
Gary	4/21/2022	2	Recording
Brennan	4/21/2022	2	Recording
Zach	4/21/2022	2	Recording
Scott	4/21/2022	2	Recording
Gary	4/21/2022	1	Build permissions in Metabase. Review Summary report
Zach	4/21/2022	0.5	make cover page
Brennan	4/21/2022	0.5	More edits on the summary report
Scott	4/21/2022	1	Editing/uploading presentation video and reviewing summary report
Brennan	4/22/2022	2	Made some final edits on the summary report and tried to standardize the writing styles.
Scott	4/22/2022	1.5	Finalizing report, fixing table of contents, and submitting

Report Time Summary			
Brennan	13.5		
Gary	16.25		
Scott	16.25		
Zach	11.25		