World Cup Soccer Database

Rebecca Thomas, Dmytry Berkout

May 11, 2016

Contents

1 Purpose

1.1 Purpose of the Document

This document is an introduction to the MondialDB project. It will provide details on the implementation and purpose of MondialDB. This document will provide a detailed design of MondialDB. It will include an information flow diagram and task forms that go into the inner workings of the project. It will also include the problems we encountered and the solutions we had for them.

1.2 Purpose of the Project

The purpose of the project is to create a database that can access various details about the World Cup. It will contain soccer team names and members and statistics such as player records, team records, penalty information, and the rank obtained. Another major part of this project is creating a tool to extract information about the World Cup from the Internet. It will be able to extract information from various websites and put them into a database readable format. The tool will be able to deal with data from European and American websites and transform it into a standard format. This project will allow for easy and readable queries.

1.3 Purpose of Phase 1

In this phase of the project we must design the database and receive feedback on it. We will attempt to catch any significant errors in design at this stage and prevent a broken or severely flawed implementation from going through. If design errors are caught early it will require considerably less work to fix them.

1.4 Purpose of Phase 2

In this phase of the project we must create the graphical schema, relational schema, and pseudo code for the project.

1.5 Purpose of Phase 3

In this phase of the project we must create a user manual on how to use the completed database system.

2 Problems and Solutions

The following list is composed of problems we encountered with the conceptual design of the project.

- Problem: We lack database design experience. Without experience, it is difficult to design a database.

  Solution: Follow project examples and study how databases are designed.

- Problem: We don't understand the World Cup and we don't know where to find data for it.

  Solution: Find out how the World Cup works. Research various World Cup websites. Find several that are usable for our project.

- Problem: We don't know how to extract data from websites.

  Solution: Research data extraction. Make a very basic test script. See what suites are available for use.

- Problem: We don't know how to create websites.

  Solution: Research the creation of websites. Make a very basic website. See what suites are available for use.

- Problem: We don't live close together.

  Solution: Use text messages, phone calls, and email to communicate. Use Google documents to share data. S

- Problem: The websites are in UTF-8, but when I try printing in UTF-8, it fails in certain cases.

  Solution: Convert it to the latin-1 encoding and then print it.

- Problem: Certain teams don't exist anymore.

  Solution: Give them their own team id anyway.

- Problem: There are three different types of Goals - Regular goal, Penalty goal, and Own goal.

  Solution: Add a Type field to goal that's an enum to describe the type of goal.

3 Assumptions

The assumptions we made are the following:

- Our web server will not be overloaded despite not having restrictions on who can use it.
- Our web server software will not fail.
- The database software will be sufficient for the scope of this project.
- The data pulled from websites is accurate.
- We will not need multilingual support. We will support only English.
- We will not need handicap support for our website.
- We will not need a mobile accessible version for our website.

- Our users are average people who do not have a computer science background but are able to comfortably use the internet.
- Coaches should not be put into the player table.
- Different teams have unique names. For example, there aren't two Brazil teams.
- A player can score more than one goal in a match.
- Teams that haven't qualified for a world cup aren't added to the database.

Part I
Phase 1 Documentation

4 Environment and Requirements Analysis

4.1 Using MondialDB

The user will interact with MondialDB through our website. The user will connect to the website using a web browser and a simple webpage will be displayed. The sole purpose of this website is to run specific queries from the web-server through MondialDB and send the results back to the user. The website will only contain a selection of the predefined queries. On selection, a website form will appear which will contain the necessary information to perform the specified query. Both the website and web-server will check the input for validity. Once the query is processed, a table of results will appear underneath the form.

4.2 Extract Transform Load Tool

For this project we will write a python script to pull data from selected websites. The ideal script will be as simple as possible while robust enough to work with a number of different websites. We will input a list of websites and the tool should automatically convert the data into database format and insert it, or provide a script to insert it into MondialDB.

4.3 Top-Level Information Flow Diagram

See Figure 1 on page 7 for the Top-Level Information Flow Diagram. The flow is generally as follows:

1. Data is input into MondialDB from Soccer Websites through Extract-Transform-Load.
2. The user asks for the website and is provided it through the Webpage Server.

3. The Web Query Processor decides what kind of query the user is asking for.

4. The Webpage SQL Query Processor translates the user query into SQL to be executed on MondialDB.

5. The query is executed and database results are returned from MondialDB

6. The results are outputted to the user with the format depending on the kind of query.

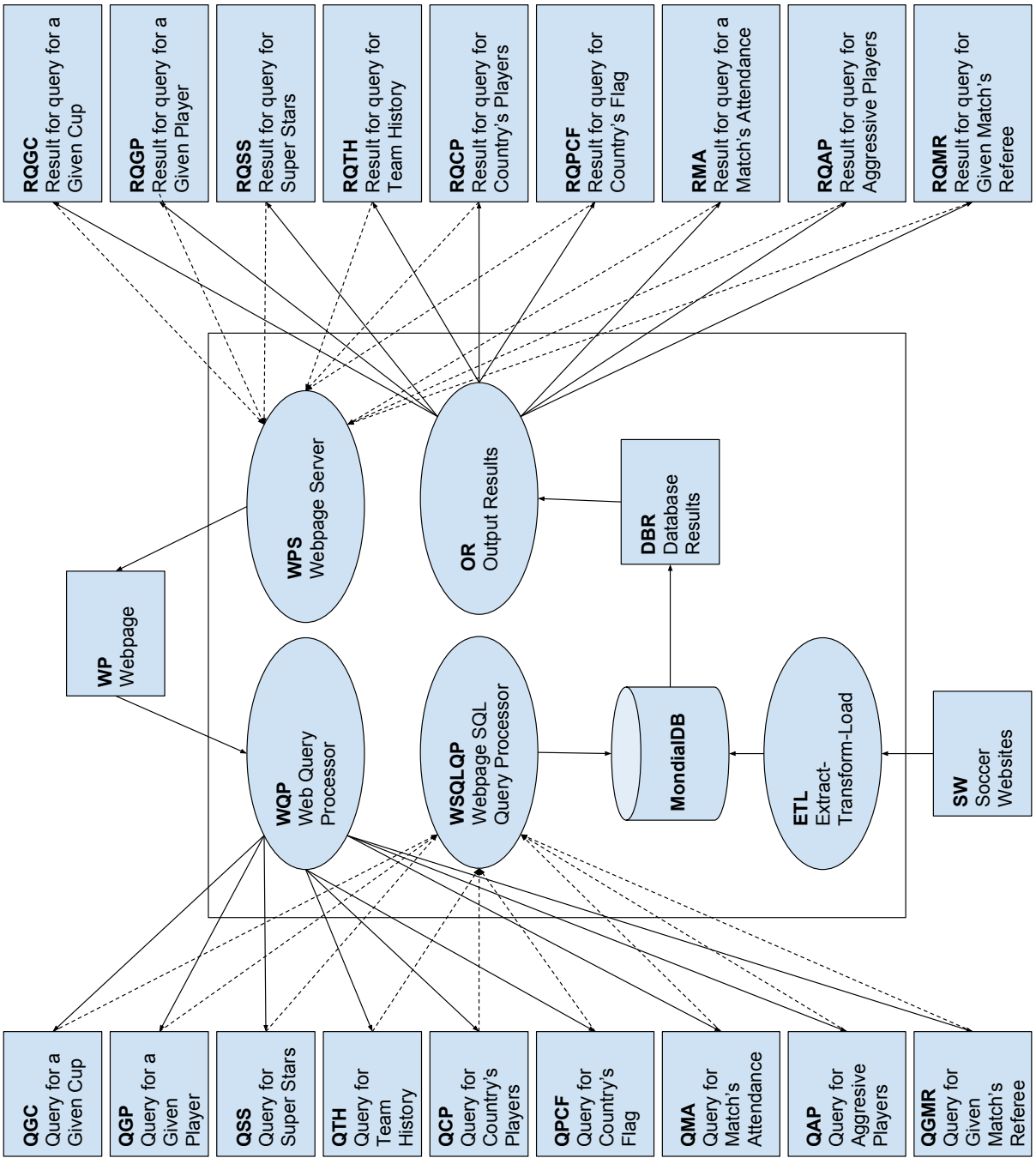

Figure 1: Information Flow Diagram

5 List of Tasks and Task Flow Diagram

We describe the tasks and subtasks necessary to make, populate, and query MondialDB. See Figure 2 on page 8 for the Task Flow Diagram. The tasks are the following:

- Extract, Transform, and Load Task
- Webpage Server Task
- Web Query Processor
- Webpage SQL Query Processor
- Output Results Task

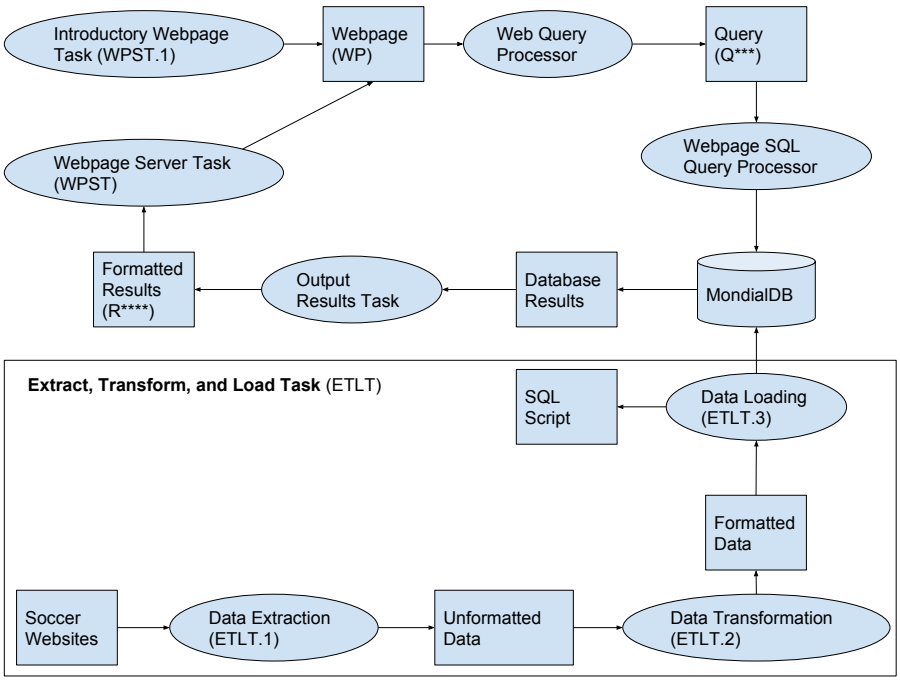

Figure 2: Task Flow Diagram

5.1 Extract, Transform, and Load Task

Task Label ETLT
Task Name Extract, Transform, and Load Task
Performer Python script
Purpose To extract data from Soccer Websites, transform it into a usable format, and send it to MondialDB
Enabling Condition On database creation or database update
Description It takes the information from Soccer Websites and puts the information into MondialDB.
Frequency On database update
Duration It will depend on the extraction, transformation, and load subtasks.
Importance Most important
Maximum Delay It depends on the subtasks.
Input Soccer Websites
Output Copy of the SQL script run by the python script as well as debugging information
Document Use Soccer Websites, Unformatted Data, Formatted Data
Operations Performed Data extraction, data transformation, and data loading
Subtasks Data extraction (ETLT.1), data transformation (ETLT.2), data loading (ETLT.3)
Error Conditions Errors from subtasks

5.1.1 Data Extraction

Task Label ETLT.1
Task Name Data Extraction
Performer Python script
Purpose To extract data from Soccer Websites
Enabling Condition On database data insertion
Description It pulls HTML from the Soccer Websites. It then parses the HTML for data to put into MondialDB.
Frequency On database update
Duration It depends on how quickly websites are scraped and extracted.
Importance Most important
Maximum Delay It depends on how many Soccer Websites are chosen.
Input Soccer Websites
Output Unformatted Data from the Soccer Websites
Document Use Soccer Websites -> Unformatted Data
Operations Performed Data extraction
Subtasks None
Error Conditions Soccer Websites are invalid. The format of the website is invalid or confusing.

5.1.2 Data Transformation

Task Label	ETLT.2
Task Name	Data Transformation
Performer	Python script
Purpose	To transform data from Soccer Websites into a standardized format
Enabling Condition	On database data insertion
Description	It standardizes the data produced by Data Extraction. For example, names that are formatted like "Last Name, First Name" and "First Name Last Name" shall be changed into a standard format.
Frequency	On database update
Duration	It depends on how quickly the data goes from being unformatted to being formatted.
Importance	Important
Maximum Delay	It depends on how badly the original data was formatted.
Input	Unformatted Data from Data Extraction
Output	Formatted Data
Document Use	Unformatted Data -> Formatted Data
Operations Performed	Data transformation
Subtasks	None
Error Conditions	The data are formatted badly.

5.1.3 Data Loading

Task Label	ETLT.3
Task Name	Data Loading
Performer	Python script
Purpose	To load formatted data into MondialDB
Enabling Condition	On database data insertion
Description	Makes an SQL script that inserts the standardized data from Data Transformation into MondialDB.
Frequency	On database update
Duration	It depends on how quickly the data is inserted into MondialDB
Importance	Most important
Maximum Delay	It depends on how the slow the connection between the data collector and the database is.
Input	Formatted Data
Output	Log from inserting into MondialDB. Copy of the SQL script run by the python script.
Document Use	Formatted Data -> SQL Script
Operations Performed	Data Loading
Subtasks	None
Error Conditions	There is a faulty connection with the database.

5.2 Web Query Processor

Task Label	WQPT
Task Name	Web Query Processor
Performer	Web-server
Purpose	Processes queries from the web-server and sends the respective query to the Webpage SQL Query Processor Task.
Enabling Condition	After web-server runs
Description	It determines the user-specified query from the forms on the Webpage. Once this query is validated, it sends the query on to the Webpage SQL Query Processor.
Frequency	Always on
Duration	As long as the database is active
Importance	Important
Maximum Delay	Response delay to user requests should be short and less than 10 seconds.
Input	User-inputted form data from the Webpage
Output	Sends Query type to Webpage SQL Processor Task. See Query Types.
Document Use	Webpage -> Query
Operations Performed	Validates queries and determines query type.
Subtasks	None
Error Conditions	The query isn't valid.

5.3 Webpage SQL Query Processor Task

Task Label WSQLQPT
Task Name Webpage SQL Query Processor Task
Performer Webpage server
Purpose Sends the query from the user-specified query task to MondialDB
Enabling Condition User submits valid query
Description The WSQLQPT is the final step before the query reaches MondialDB. It will send a perform a raw SQL query in MondialDB
Frequency Triggers on every received valid user query
Duration The SQL shouldn't take longer than 15 seconds to run.
Importance Important
Maximum Delay The SQL shouldn't take at most longer than 45 seconds to run.
Input User-specified query from the following list:
QGC (Query for a given cup)
QGP (Query for a given player)
QSS(Query for Super Stars)
QTH(Query for Team History)
QCP(Query for Country's Players)
QPCF(Query for Country's Flag)
QMA (Query for Match's Attendance)
QAP (Query for Aggressive Players)
QGMR (Query for a Given Match's Referee)
Output Sends SQL query to MondialDB
Document Use Query
Operations Performed If the query is a valid query from the aforementioned list, then create the SQL command. If it is not a valid query, send an error message back.
Subtasks None
Error Conditions The user-specified query is not valid.

5.4 Output Results Task

Task Label	OR
Task Name	Output Results Task
Performer	Webpage Server
Purpose	Sends the query results to the user
Enabling Condition	MondialDB receives query and sends output to Output Results Task
Description	The WSQLQPT is the final step before the query reaches MondialDB. It will send a raw SQL query to MondialDB.
Frequency	Triggers on every received request
Duration	In ideal network conditions, it should be short and less than 10 seconds.
Importance	Most Important
Maximum Delay	Response delay to user requests should be short and less than 10 seconds.
Input	For a user-specified query, Database Results
Output	The Formatted Result from the results
Document Use	Database Results -> Formatted Results
Operations Performed	Sends query results to Webpage Server Task
Subtasks	None
Error Conditions	MondialDB gave back error conditions instead of valid results. Connection to user is disrupted. Web server becomes overloaded.

6 List of Documents

See page 14 for repeated Task Flow Diagram. The documents are the following:

- Webpage
- Query
- Database Results
- Formatted Results
- Soccer Websites
- Unformatted Data
- Formatted Data
- SQL Script

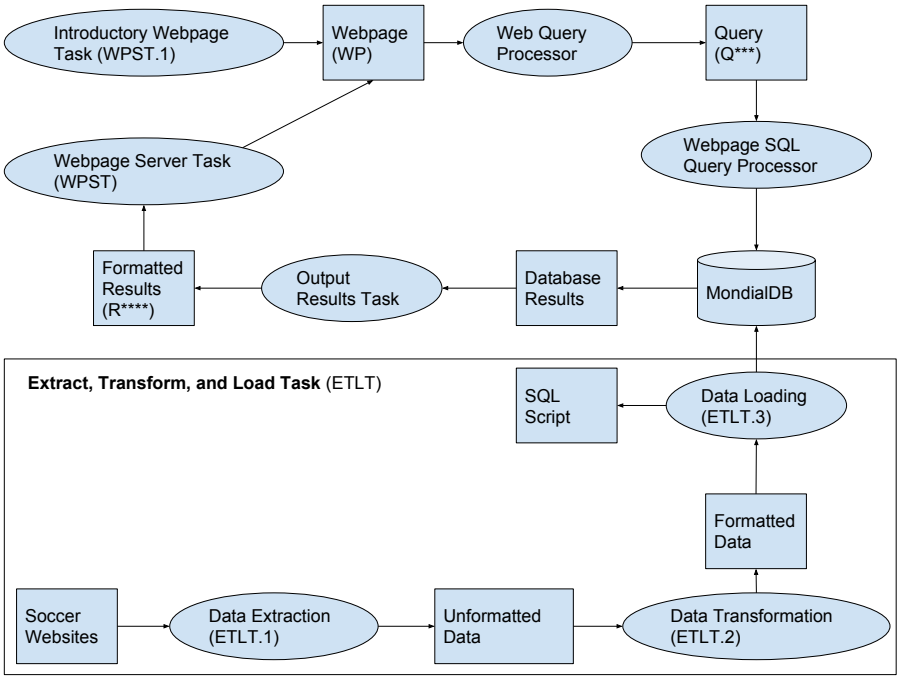

Figure 3: Task Flow Diagram

6.1 Queries

The types of Queries:

- QGC (Query for a given cup)
- QGP (Query for a given player)
- QSS (Query for Super Stars)
- QTH (Query for Team History)
- QCP (Query for Country's Players)
- QPCF(Query for Country's Flag)
- QMA (Query for Match's Attendance)
- QGMS (Query for Aggressive Players)
- QGMR (Query for a Given Match's Referee)

Query for Country's Flag This query returns information about a country's flag. Most notably this includes a link to a picture of that country's flag. The query is interesting because it adds an extra dimension to the database. Users might find it easier to recognize a country by its flag instead of by its name. Others might be curious and be inspired to learn more interesting information about flags. Flags traditionally represent countries. They are symbols for the people of that country to rally around and to represent. They represent the history of the nation. Thus, they are interesting.

Query for Match's Attendance This query returns information about a specific match. It will give back what stadium the match was held in. It will also tell the user how many people attended that particular match. Attendance information is wonderfully informative. The user could try to find out what teams are most popular, what stadiums are most popular, or what host countries are most popular.

Query for Aggressive Players This query tells users who the most aggressive players are. The metric of aggressiveness is determined by how many penalties they accumulate. It is interesting to see the human side of soccer and how vicious it can get.

Query for a Given Match's Referee This query tells users what referee was calling for a particular match. Many people assume referees are biased against the team that they like. By using this query, they could see if the referee really is biased against their favorite team.

Part II
Phase 2 Documentation

7 ER Model

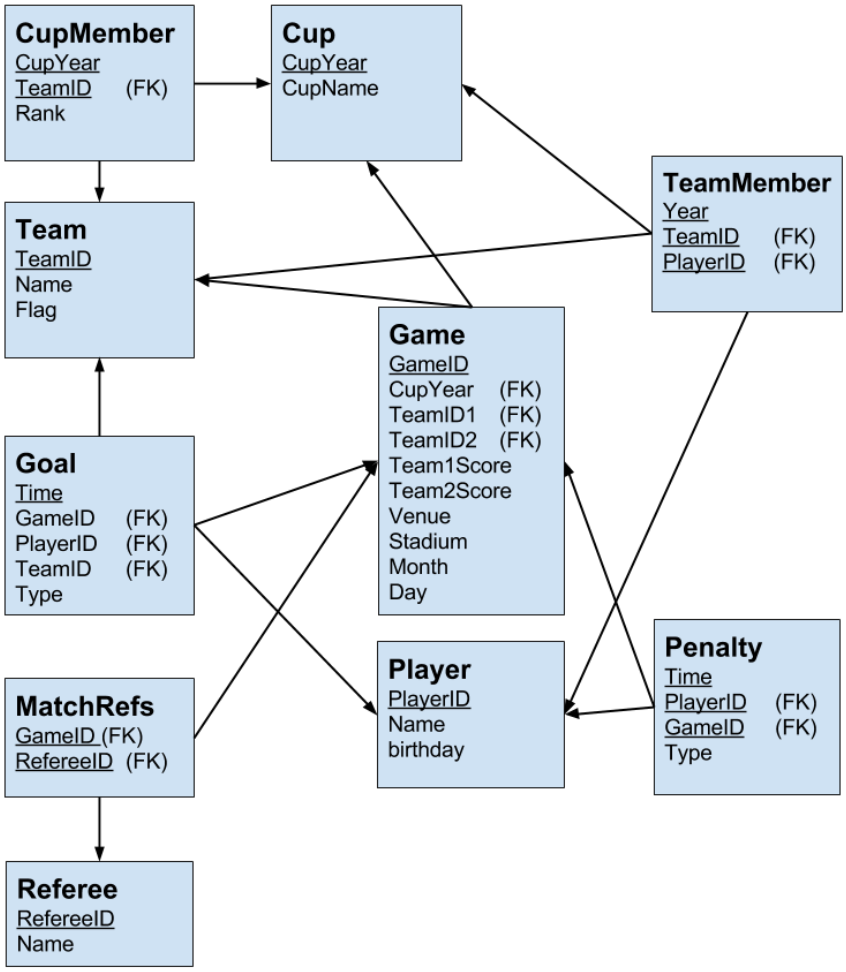

Figure 4: ER Model

Figure 5: Relational Schema

Cup	**CupYear**	CupName

CupMember	**CupYear (FK to Cup)**	**TeamID (FK to Team)**	Rank

Game	**GameID**	CupYear(FK to Cup)	TeamID1 (FK to Team)	TeamID2 (FK to Team)
	Team1Score	Team2Score	Venue	Stadium
	Month	Day		

Goal	**Time**	GameID(FK to Game)	PlayerID (FK to Player)	TeamID (FK to Team)
	Type			

Penalty	**Time**	**GameID(FK to Game)**	**PlayerID (FK to Player)**	Type

MatchRefs	**RefereeID (FK to Referee)**	**GameID(FK to Game)**

Player	**PlayerID (FK to Player)**	Name	birthday

Team	**TeamID**	Name	Flag

TeamMember	**Year (FK to Cup)**	**TeamID(FK to Team)**	**PlayerID(FK to Player)**

8 Relational Schema

See Figure 5 on page 17.

9 Extract, Transform, and Load Task

The basic process is to grab the html of a website, parse it, and then execute a regex on its contents to capture data. After extraction, the program transforms the data into something acceptable. In most cases, no transformation is necessary. However, dates always need to be reformatted into a standard format such as YYYY-MM-DD. After transformation, the program loads the information into the database.

Throughout extraction, transformation, and loading, the program logs important information to files extracting_logYmd-HMS and loading_logYmd-HMS.

All of the ETL code can be found online at `https://github.com/rathomas99/fifa-world-cup-scrape`.

10 Queries

The code for the SQL queries will be attached separately.

Part III
Phase 3 Documentation

11 User Manual

11.1 How to Use the Webpage

Go to the IP Address `http://100.15.105.119/`. You will see the website. On the right hand side are various query types. If you click on the name of the query, a textbox asking for input will show up. Enter in the input that you would like to query on and then click the button "Get Result".

11.2 How to Use the ETL Tool

You must install the prerequisite python libraries first.
```
sudo pip install requests
sudo pip install pymysql
sudo pip install beautifulsoup4
```
You must set up the MySQL database with the aforementioned schema. In `load.py`, make sure that the function openDB() has the correct host, user, password, and database. After installing the python libraries and setting up the database, run `python extract.py`. It will scrape the internet for FIFA World Cup data. The extracting python script will call the loading python script by itself. There will be two log files created. One log file will list warnings, errors, and messages from the extracting script. The other log file will list warnings, errors, and messages from the loading script. If loading goes perfectly well, the log file will essentially look like an SQL file that inserts all the relevant data.

12 Limitations

1. The scraping doesn't always get the data it needs.
2. If you try to insert a goal before the players and the teams, it won't work. A goal needs to fulfill the foreign key constraints of the player existing and the teams existing.
3. Not all players are added.
4. Not all teams are added.
5. If one step in the process fails, the rest of it fails too.
6. It's slow.
7. The extraction does not take into account penalty shootouts.