

# World Cup Soccer Database

Rebecca Thomas, Dmytry Berkout

May 11, 2016

## Contents

<b>1 Purpose</b>	<b>3</b>
1.1 Purpose of the Document . . . . .	3
1.2 Purpose of the Project . . . . .	3
1.3 Purpose of Phase 1 . . . . .	3
<b>2 Problems and Solutions</b>	<b>3</b>
<b>3 Assumptions</b>	<b>4</b>
 <b>I Phase 1 Documentation</b>	 <b>4</b>
<b>4 Environment and Requirements Analysis</b>	<b>4</b>
4.1 Using MondialDB . . . . .	4
4.2 Extract Transform Load Tool . . . . .	5
4.3 Top-Level Information Flow Diagram . . . . .	5
<b>5 List of Tasks and Task Flow Diagram</b>	<b>7</b>
5.1 Extract, Transform, and Load Task . . . . .	8
5.2 Web Query Processor . . . . .	10
5.3 Webpage SQL Query Processor Task . . . . .	11
5.4 Output Results Task . . . . .	12
<b>6 List of Documents</b>	<b>13</b>
6.1 Queries . . . . .	14
 <b>II Phase 2 Documentation</b>	 <b>14</b>
<b>7 ER Model</b>	<b>15</b>
<b>8 Relational Schema</b>	<b>15</b>
<b>9 Extract, Transform, and Load Task</b>	<b>15</b>

10 Queries

15

# 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 are to 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.

# 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.

### 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.

## 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 6 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.

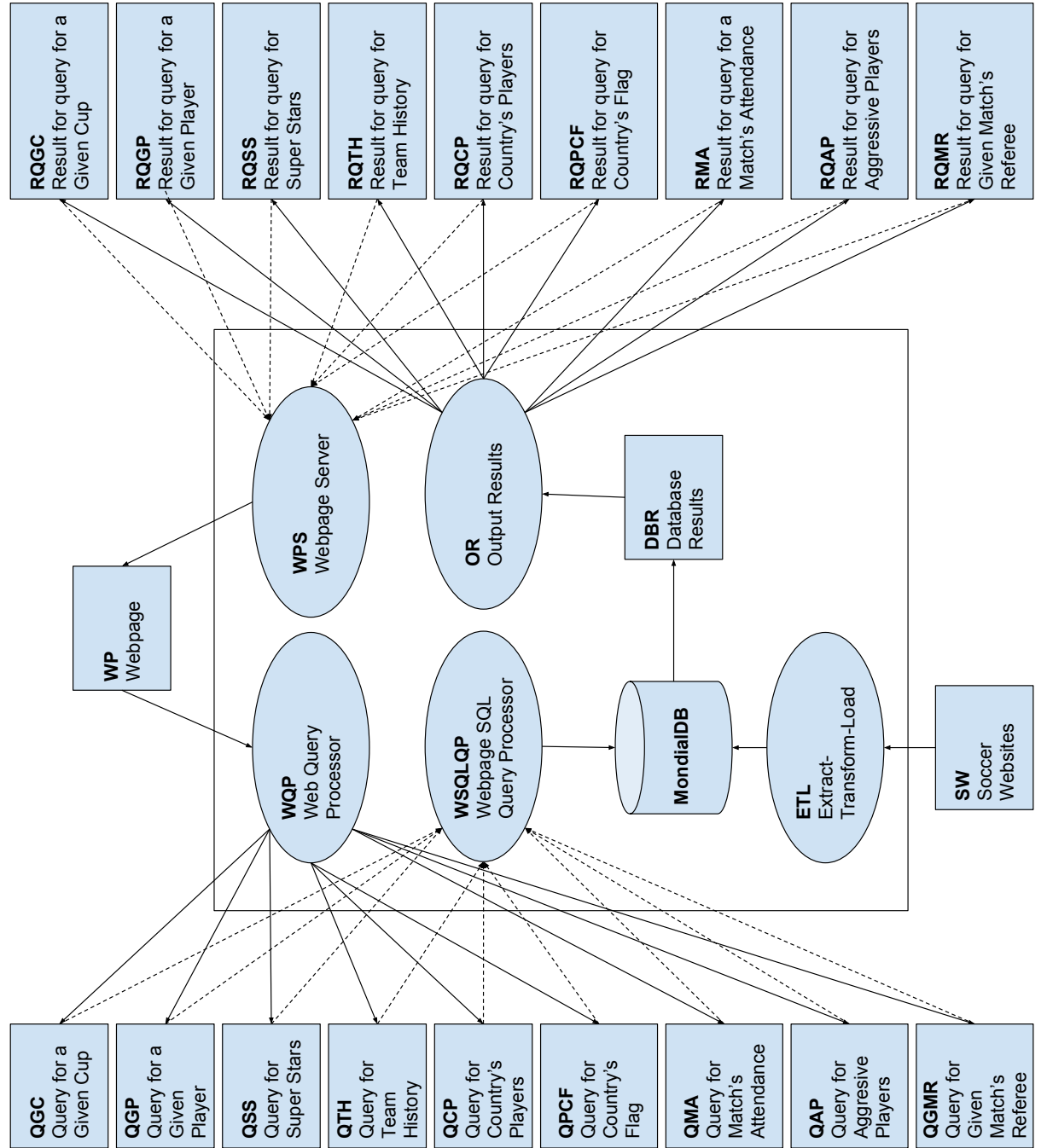


Figure 1: Information Flow Diagram

## 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 7 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

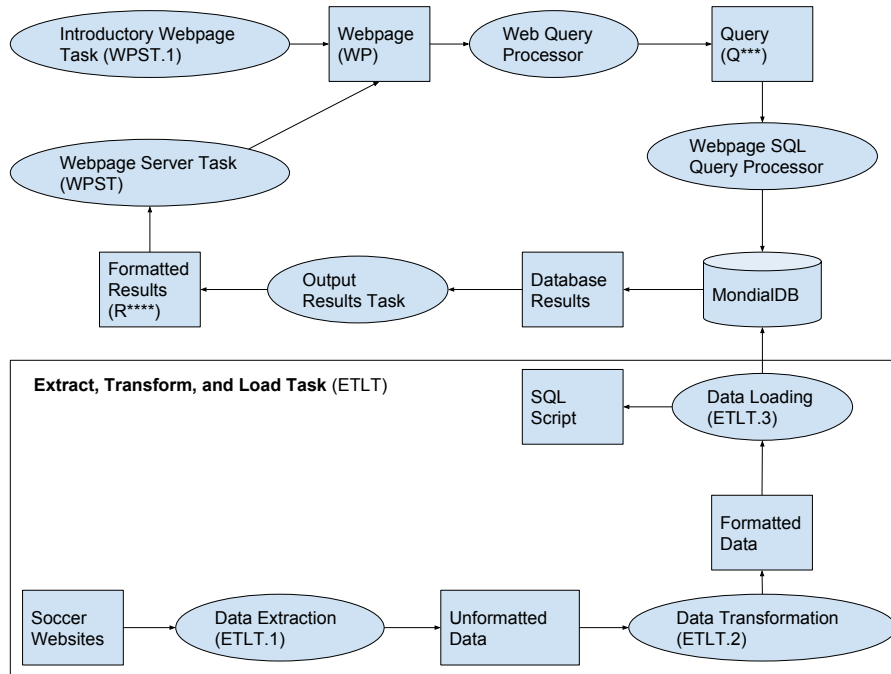


Figure 2: Task Flow Diagram

## 5.1 Extract, Transform, and Load Task

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

### 5.1.1 Data Extraction

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



### 5.1.2 Data Transformation

<b>Task Label</b>	ETLT.2
<b>Task Name</b>	Data Transformation
<b>Performer</b>	Python script
<b>Purpose</b>	To transform data from Soccer Websites into a standardized format
<b>Enabling Condition</b>	On database data insertion
<b>Description</b>	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.
<b>Frequency</b>	On database update
<b>Duration</b>	It depends on how quickly the data goes from being unformatted to being formatted.
<b>Importance</b>	Important
<b>Maximum Delay</b>	It depends on how badly the original data was formatted.
<b>Input</b>	Unformatted Data from Data Extraction
<b>Output</b>	Formatted Data
<b>Document Use</b>	Unformatted Data -> Formatted Data
<b>Operations Performed</b>	Data transformation
<b>Subtasks</b>	None
<b>Error Conditions</b>	The data are formatted badly.

### 5.1.3 Data Loading

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

## 5.2 Web Query Processor

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

### 5.3 Webpage SQL Query Processor Task

<b>Task Label</b>	WSQLQPT
<b>Task Name</b>	Webpage SQL Query Processor Task
<b>Performer</b>	Webpage server
<b>Purpose</b>	Sends the query from the user-specified query task to MondialDB
<b>Enabling Condition</b>	User submits valid query
<b>Description</b>	The WSQLQPT is the final step before the query reaches MondialDB. It will send a perform a raw SQL query in MondialDB
<b>Frequency</b>	Triggers on every received valid user query
<b>Duration</b>	The SQL shouldn't take longer than 15 seconds to run.
<b>Importance</b>	Important
<b>Maximum Delay</b>	The SQL shouldn't take at most longer than 45 seconds to run.
<b>Input</b>	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)
<b>Output</b>	Sends SQL query to MondialDB
<b>Document Use</b>	Query
<b>Operations Performed</b>	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.
<b>Subtasks</b>	None
<b>Error Conditions</b>	The user-specified query is not valid.

## 5.4 Output Results Task

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

## 6 List of Documents

See page 13 for repeated Task Flow Diagram. The documents are the following:

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

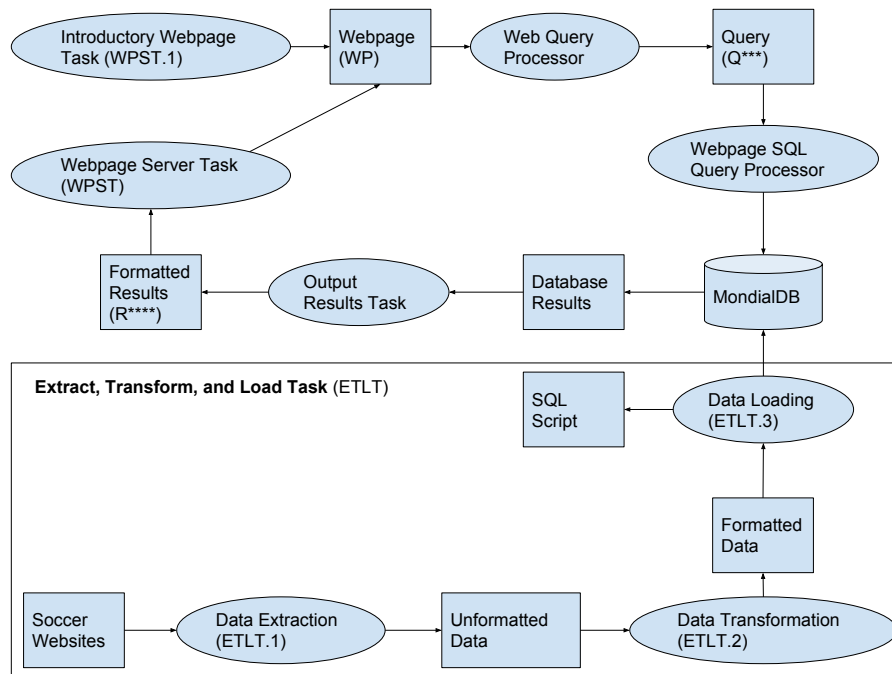


Figure 3: Task Flow Diagram

## 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

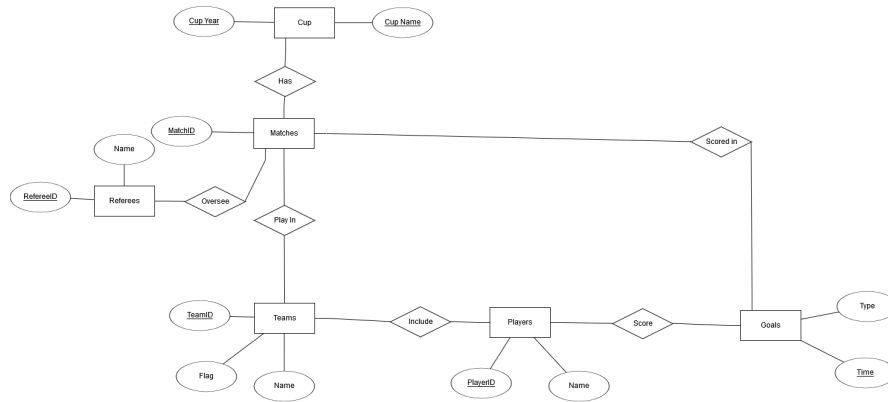


Figure 4: ER Model

## 8 Relational Schema

See Figure 5 on page 16.

## 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`.

## 10 Queries

The code for the SQL queries will be attached separately.

Figure 5: Relational Schema

Cup	<u>CupYear</u>	CupName
-----	----------------	---------

CupMember	<u>CupYear (FK to Cup)</u>	<u>TeamID (FK to Team)</u>	Rank
-----------	----------------------------	----------------------------	------

Game	<u>GameID</u>	CupYear(FK to Cup)	TeamID1 (FK to Team)	TeamID2 (FK to Team)
	Team1Score	Team2Score	Venue	Stadium
	Month	Day		

Goal	<u>Time</u>	GameID(FK to Game)	PlayerID (FK to Player)	TeamID (FK to Team)
	Type			

Penalty	<u>Time</u>	<u>GameID(FK to Game)</u>	<u>PlayerID (FK to Player)</u>	Type
---------	-------------	---------------------------	--------------------------------	------

MatchRefs	<u>RefereeID (FK to Referee)</u>	<u>GameID(FK to Game)</u>
-----------	----------------------------------	---------------------------

Player	<u>PlayerID (FK to Player)</u>	Name	birthday
--------	--------------------------------	------	----------

Team	<u>TeamID</u>	Name	Flag
------	---------------	------	------

TeamMember	<u>Year (FK to Cup)</u>	<u>TeamID(FK to Team)</u>	<u>PlayerID(FK to Player)</u>
------------	-------------------------	---------------------------	-------------------------------

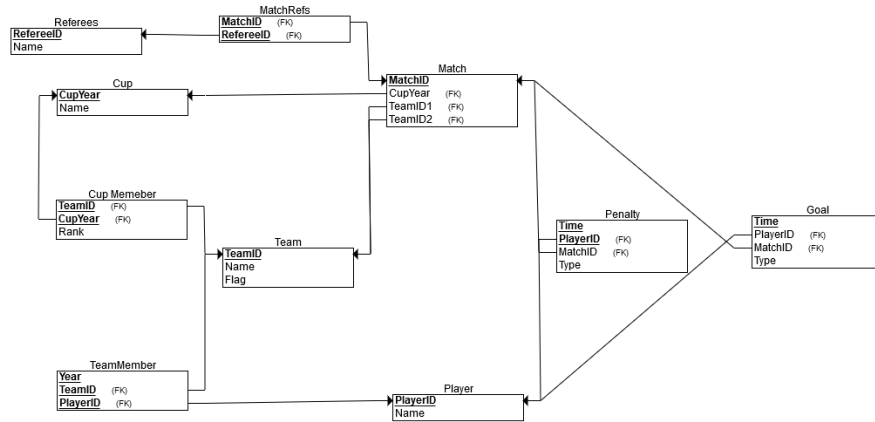


Figure 6: Relational Schema