**Project 2:**

**CSE 5331:** Database Models and Implementation.

**Intermediate Report:** Documentation related to the Data Structures, sample Pseudo Code and Mongo DB installation along with MySQL import procedures.

**By,**

**Pramod Kemisetty**

**1001570333.**

**Files:**

1. **TeamScores.py :** This python file contains the logic and the data structures needed to map the Relational Data retrieved from the MySQL into Mongo DB. We need to use a python mongo Db connector – called “pymongo” to connect the python code to the Mongo DB as well as a “mysql.connector” to get the rows from the MySQL. Using a “Select” statement we will retrieve the data from MySQL. Now, we will use arrays to store each of the attributes mentioned in the problem statement like Team (i.e. the Team Name), and a collection of the team match (GAME) scores where in turn each match scores the date of the match, the name of the city and stadium where the match was played, repeat of the team’s name, the team score in the match, the name of the opposing team, and the score of the opposing team.

*This is achieved by writing a query which retrieves all the necessary fields by using joins from the MySQL.*

*Also, by using “json.dumps” we will convert the rows into JSON type for insertion into the Mongo DB.*

1. **PlayerData.py:** This python file contains the logic and the data structures needed to map the Relational Data retrieved from the MySQL into Mongo DB. We need to use a python mongo Db connector – called “pymongo” to connect the python code to the Mongo DB as well as a “mysql.connector” to get the rows from the MySQL. Using a “Select” statement we will retrieve the data from MySQL. Now, we will use arrays to store each of the attributes mentioned in the problem statement like the player name (Pname), the player’s team name (Team), the player number (PNo) and position (Position), and a collection of games that the player has started – for each game (match) the player has started, include the MatchDate, City, Stadium Name, and opposing team name. Also include for each player a collection of goals that the player has scored (if any) – for each goal scored, include the GoalType, Time, MatchDate, City, Stadium Name, and opposing team name.

*This is achieved by writing a query which retrieves all the necessary fields by using joins from the MySQL.*

*Also, by using “json.dumps” we will convert the rows into JSON type for insertion into the Mongo DB.*

**Data Structures:**

1. **TeamScores.py.**

* *TeamName []*

*Collection of attributes under Team Match scores.*

* *MatchDate []*
* *CityName []*
* *StadiumName []*
* *TeamName [] – (Second mention of the team name).*
* *TeamScore []*
* *OpposingTeamName []*
* *OpposingScore []*

1. **PlayerData.py.**

* *PlayerName []*
* *PlayerTeamName []*
* *PlayerNumber []*
* *Position []*

*If the player has started the match --*

* *PlayerStartMatch []*
* *MatchDate []*
* *City []*
* *StadiumName []*
* *OpposingTeamName []*

*If the Player has scored a goal --*

* *GoalType []*
* *Time []*
* *MatchDate []*
* *City []*
* *StadiumName []*
* *OpposingTeamName []*

**Pseudo code:**

1. *TeamScores.py:*

* *Import mysql.connector.*
* *Import json*
* *Import pymongo*
* *Import MongoClient*
* *Client = MongoClient(‘mongodb://localhost:27017’)*
* *Db = client.MySQL.*
* *First we will set the connection parameters like user = “root”, password = “password”, host = “127.0.0.1”, database = “MySQL”.*
* *Create an object called Cursor for the connection established.*
* *Write a select query that will fetch all the required attributes for TeamScores.*
* *Using a For Loop, iterate all the records retrieved from the select query and store them in the arrays mentioned in the Data structures section like TeamName [], MatchDate [], CityName [], StadiumName [], TeamName [] – (Second mention of the team name).*
* *TeamScore [], OpposingTeamName [], OpposingScore [].*
* *Now using json.dumps (obj\_arr) where obj\_arr is the array where the result set from the MySQL select query has been stored is passed.*
* *My\_json\_String is another variable where all the converted json objects are stored.*
* *Now, cursor. Close ()*
* *Connection. Close ()*
* *Result = db. teamscore. insertmany(obj\_arr).*
* *Print (result. inserted\_ids) to check if the documents have been inserted correctly into the Database.*

1. *PlayerData.py:*

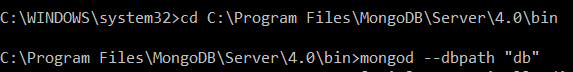
* *Import mysql.connector.*
* *Import json*
* *Import pymongo*
* *Import MongoClient*
* *Client = MongoClient(‘mongodb://localhost:27017’)*
* *Db = client.MySQL.*
* *First we will set the connection parameters like user = “root”, password = “password”, host = “127.0.0.1”, database = “MySQL”.*
* *Create an object called Cursor for the connection established.*
* *Write a select query that will fetch all the required attributes for TeamScores.*
* *Using a For Loop, iterate all the records retrieved from the select query and store them in the arrays mentioned in the Data structures section like PlayerName [], PlayerTeamName [], PlayerNumber [], Position [] etc.*
* *Now using json. dumps (obj\_arr) where obj\_arr is the array where the result set from the MySQL select query has been stored is passed.*
* *My\_json\_String is another variable where all the converted json objects are stored.*
* *Now, cursor. Close ()*
* *Connection. Close ()*
* *Result = db. teamscore. insertmany(obj\_arr).*
* *Print (result. inserted\_ids) to check if the documents have been inserted correctly into the Database.*

***Setup and Installation of MongoDB:***

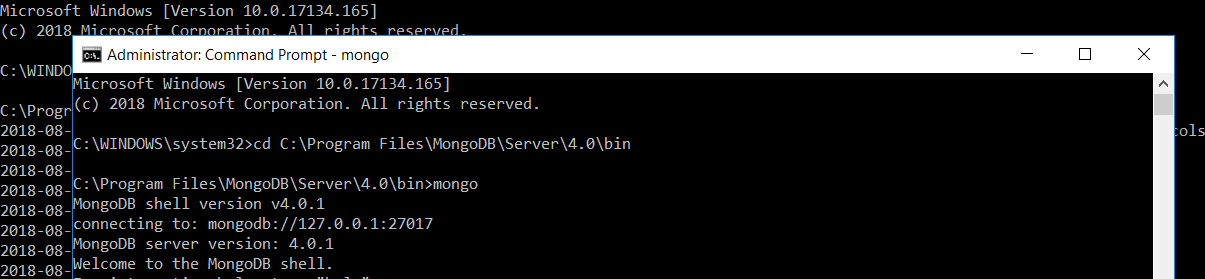
* *After downloading the setup file, we have to run the installation wizard to run the MongoDB.*



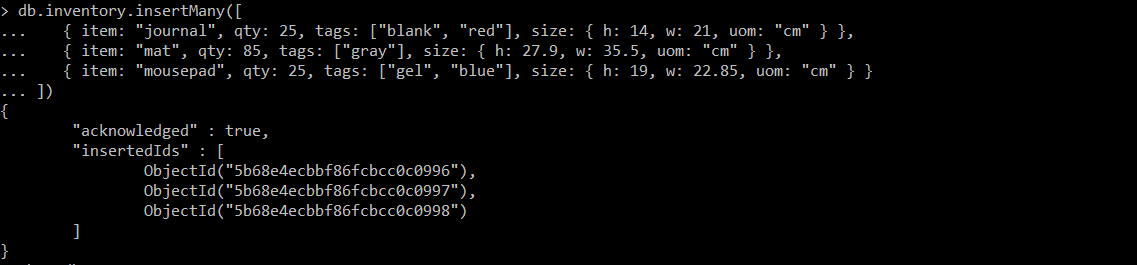
* *After that, we have to run the following commands, after creating a new folder called db.*

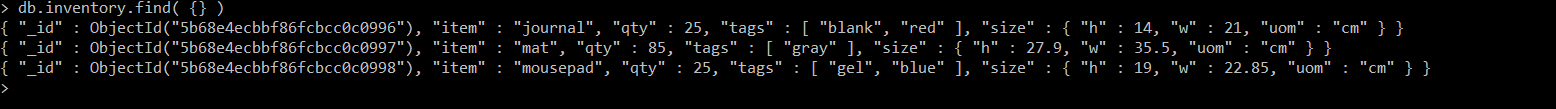


* *Now open a new Cmd prompt (As admin) and run the following commands to start the Mongo Server.*



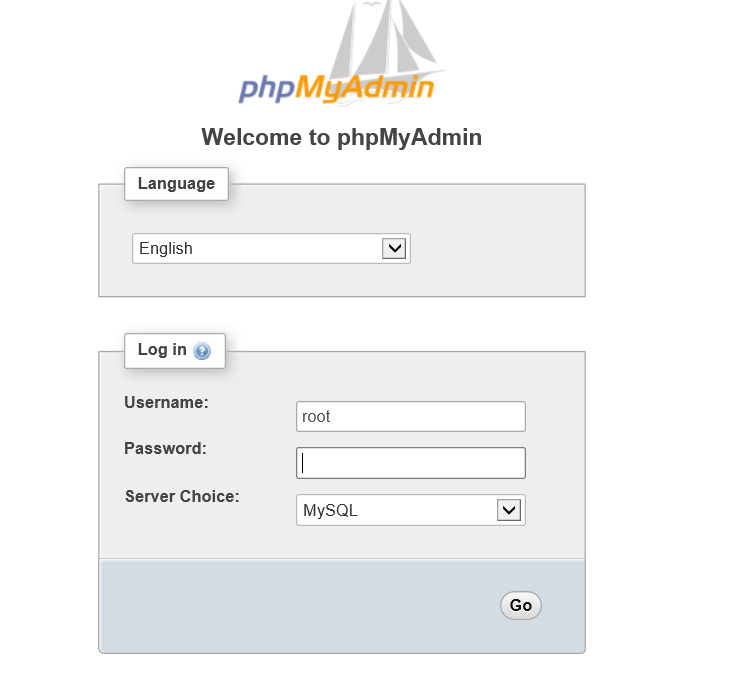
* *Inserting multiple records into the Mongo Db.*



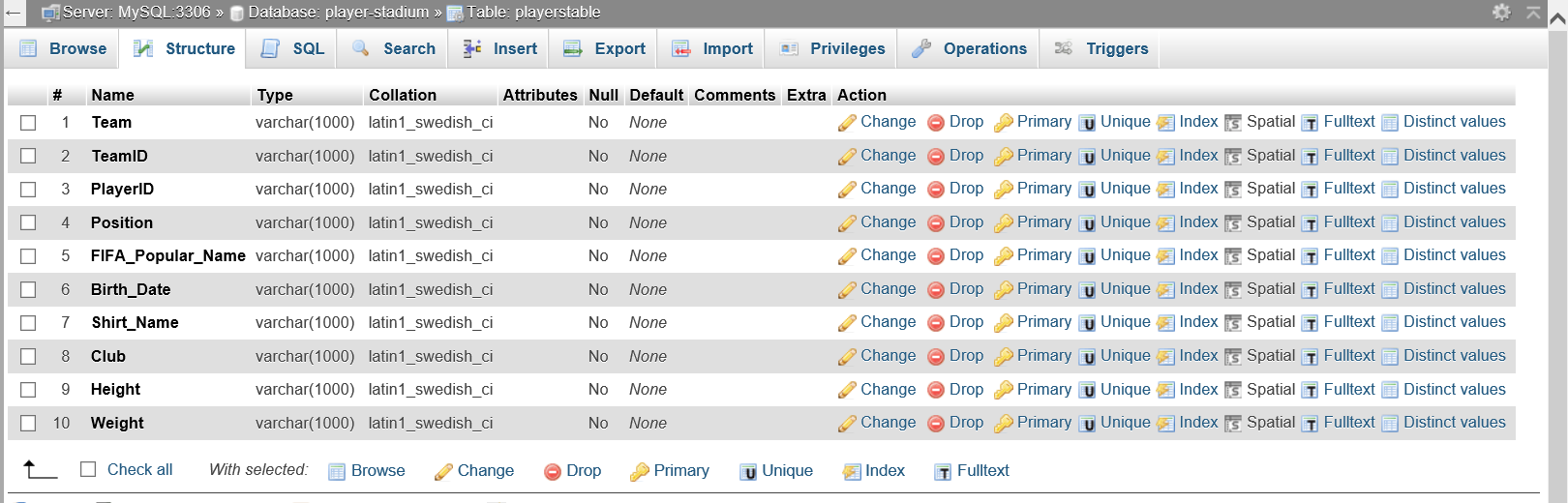
* *Retrieving the inserted records to cross verify.*

***Inserting Records from the .txt file into MySQL:***

* Open phpMyAdmin in the Wamp Server and the home page will look like this.



* After login, we must create a db and a table like below with the following column names.

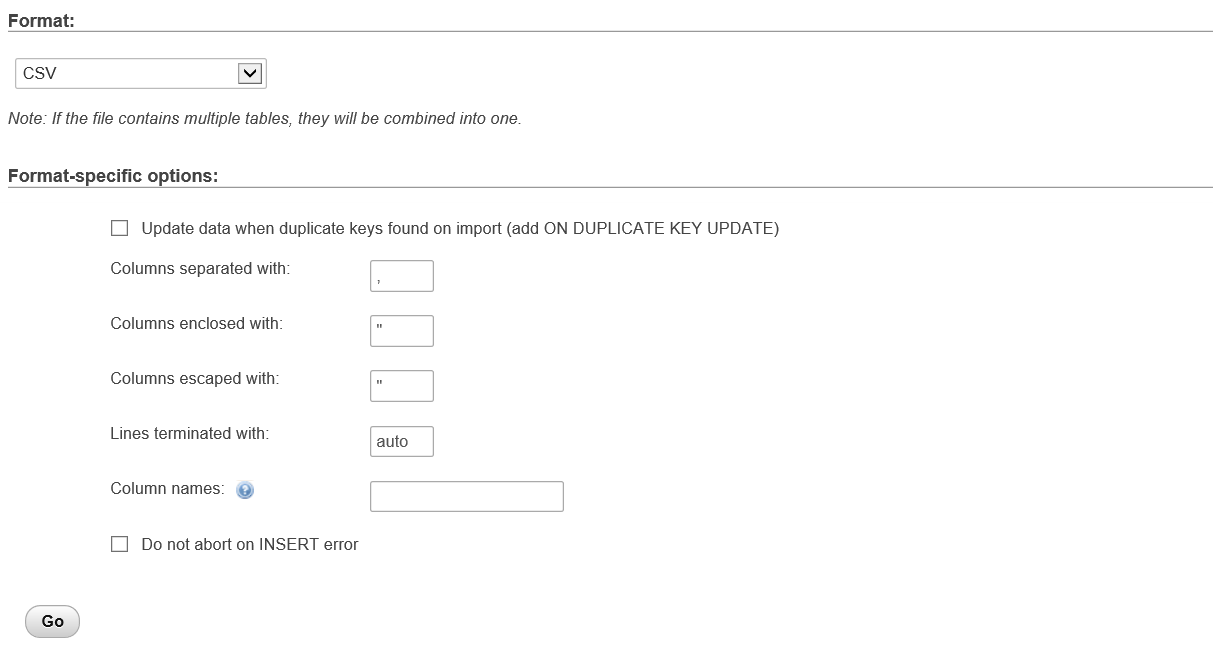


I have used the Players-text.txt file to create a table as PlayersTable and insert the data in bulk.

* Then navigate to the Import tab and browse for the file to be imported as below.



Note that the .txt file is converted into .csv file for bulk import of the file and the attributes are left unchanged.



These rules are necessary for the data to be mapped to the respective columns. So, they should be kept intact.

* After successful insertion of the data, the populated table looks like this,

