**DB2 PROJECT 2 PHASE 2**

**Team Members for Group 1:**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Elsy Fernandes | **1001602253** |
| Maria Lancy Devadoss | **1001639262** |

**Project:**

To implement a program that helps to learn usage of MongoDB as an example of a document oriented NOSQL system and see how data is stored and queried in such a system.

Two documents (complex object) -TEAM\_SCORES and PLAYER\_DATA need to be designed.

**Data Files**: We will have data files for the World cup database - Team, Players, Game, Stadium, Starting\_Lineups and Goals tables in relational (flat file) format.

**Language:**

Python

Used python panda’s framework to read data from files and store data as dataframe.

Used Pymongo to connect to local mongo DB client

**PREREQUISITES:**

* Make sure python is installed
* Download the project, extract folder from zip and open command prompt from the project location
* Install Pandas using pip install pandas
* Install pymongo using pip install pymongo
* Download the data set files (CSV files) in the folder named DB2 attached along with project and store it in Desktop. (ex:C:/Users/Maria/Desktop/DB2/Team.csv).
* Make sure mongo.exe is started before running the python file
* Once all done, Run the code by typing python finalcode.py (You can also run code using Pycharm IDE)

**PLAYER\_DATA :**

This collection will hold information like player name (FIFAPopularName), team name (Team), player number (PlayerID), position (Position),Player Number(PlayerID), collection of Games which player started and collection of Goals that player scored.

This collection uses data from Team, Players, Game, Starting\_Lineups, Stadium, Goals

**Implementation:**

1) Initiate an iterator, i =0 to distinct player name i.e.; FIFAPopularName

2) Read the Player.csv file and get FIFAPopularName, PlayerID, TeamID, Position

3) Read Starting\_Lineups.csv file and based on PlayerID and TeamID select the GameID for which the player has started the Game. Read Game.csv and retrieve the Game details which match the acquired GameID

4)Use the Game details obtained to retrieve Match Date, SID based on TeamID of Game and TeamID of Player. Read Stadium.csv to retrieve SCity, SName based on SID. Read Team.csv to retrieve opposing Team name based on TeamID. Read Goals.csv to retrieve Time based on PlayerID and TeamID. Store all the acquired information in the document

5) Increment the iterator I, so that the next Player’s details can be fetched. Perform the steps 2 to 4 until the iterator value reached the last row value.

6)Once all the Player details have been fetched store the document in to the PLAYER\_DATA document

**TEAM\_SCORES:**

This collection will hold information like team name (Team), collection of matches (Game Scores)-which contains MatchDate, Stadium city and Name, Team Name, Team score, Opposing Team Name and score.

This collection uses data from Team, Stadium, Game

**Implementation:**

1) Initiate an iterator, from i = 0 to number of TeamID (ie number of teams in Team file).

2) Read the Team.csv file and get the Team Name for the Team for the particular TeamID

3) Read the Game.csv file and get the data based on the TeamID. Get the game details such as Match Date, Team Name, Team Score, Opposing Team Name, Opposing Team score. Read Team.csv to retrieve Opposing Team Name ,Team Name based on their respective TeamID. From Stadium.csv based on SID retrieve Stadium name and City. Store all this information in a document

4) Increment the iterator I, so that the next Team’s details can be fetched. Perform the steps 2 to 4 until the iterator value reached the last row value.

5) Once all team details have been fetched, insert the document into the TEAM\_SCORES document