**Top Scorer: A Soccer Analytics Navigator**

*By*

**Kashish Shah (USC ID: 1842286535)**

**Saniya Kirkire (USC ID: 6354401648)**

**Shreyansh Khandelwal (USC ID: 6973534682)**

**MID-TERM REPORT**

According to our timeline we have been working on the HDFS commands and created PMR queries to perform search and analytics on the dataset.

We have implemented a total of 7 commands that will be working as bridge between the web app and mySQL in order to emulate HDFS.

We also created a table called namenode to store the meta data of the directories created and files uploaded. This table is an emulation of namenode in HDFS.

The table has 4 columns:

1. Id – This field assigns an id to all the meta data. It is set as AUTO INCREMENT and PRIMARY KEY for the table.
2. Path – This field stores the path of all the Directories and Files created in the EDFS.
3. Name – This field stores the name of the file added to the EDFS.
4. Type – This field stores the type of the newly created file/directory.

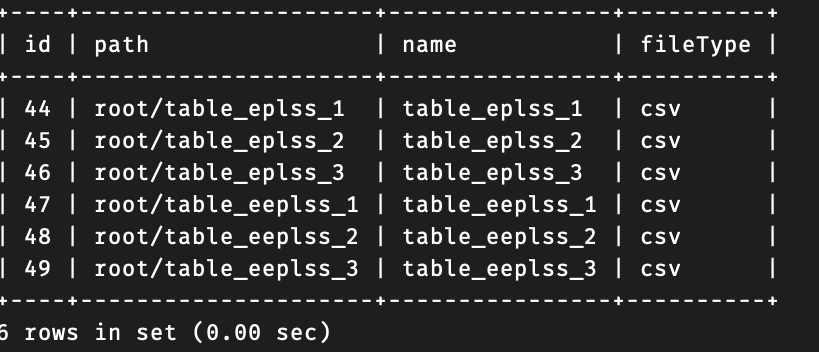


Fig. 1- Namenode

Along with Namenode we have also create datanode which is basically the table which holds the data. For example, if a user uploads a csv and decides to divide it in 3 partitions, we will create three datanodes with name ‘table\_filename\_1, table\_filename\_2, and table\_filename\_3’

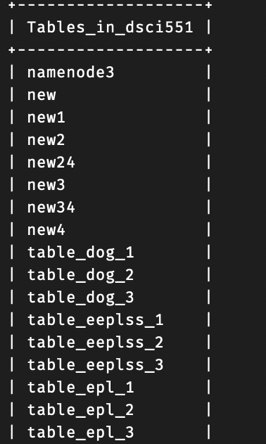


Fig. 2 – csv being stored as Datanode

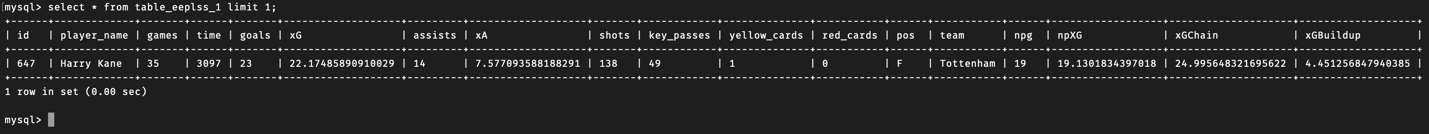
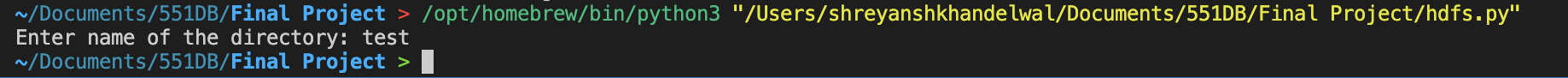


Fig. 3 – csv data as Datanode

The EDFS commands are:

1. **makeDir** – This function takes name as an argument from the user, and it does 2 tasks. Firstly, it creates a new folder or directory. Subsequently, it will add meta data created in the namenode.



Text

Description automatically generated

1. **ls** – This function simply just shows all the directories and files present. It takes no argument from user and outputs a list of paths to showcase the list of present files/directories.

Graphical user interface, text

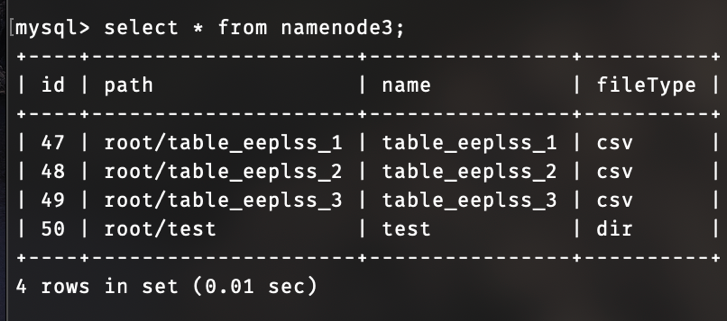
Description automatically generated

1. **cat** – This function takes name of the file as input from the user and outputs the content of the file. If the type of the file is a csv it will show a data frame in the output whereas if the type is txt, it will simply show all the content like a nano would. For example, if the user gives input as ‘epl-goalScorer(20-21).csv’ as input it will show the data in a pandas data frame.

Graphical user interface

Description automatically generated

1. **remove** – This function simply just deletes the directory or file specified by the user. It takes name of the file as an argument from user and deletes it from both the namenode and datanode table.



Graphical user interface, text

Description automatically generated

Text

Description automatically generated with low confidence

1. **Upload** – This function will ask the user to upload the csv files. It will then update 2 tables. Firstly, it will create an entry in the namenode and then it will store the data in the datanode table.

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated

1. **getPartitionLoc** – This function will allow users to get path of their file partitions that is created by the EDFS.

Graphical user interface, application

Description automatically generated

1. **readPartition** – This function will read a particular partition of the uploaded csv and return the data frame of that partition.

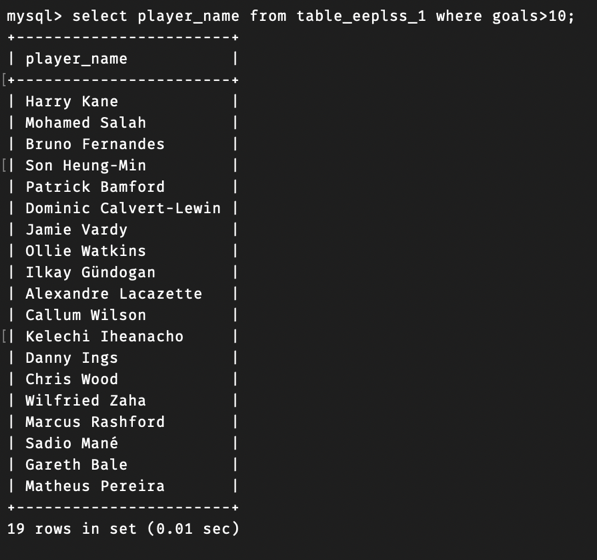
Text

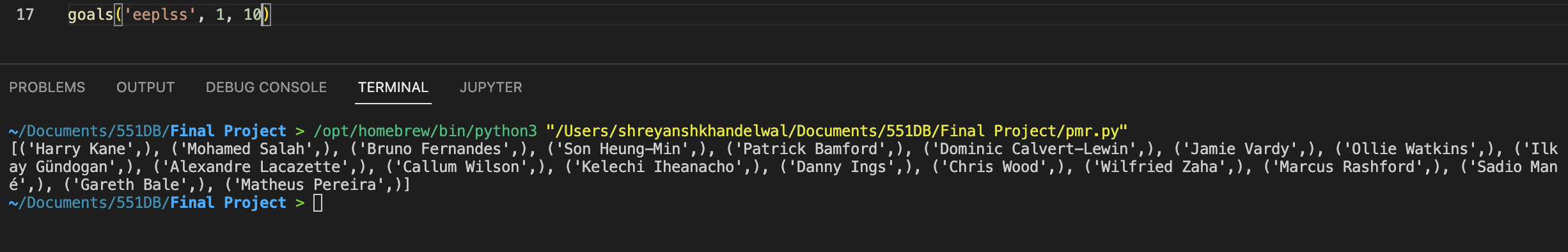
Description automatically generated

**PMR**

Below are the SQL queries that will be used to perform PMR. All the queries can run and perform on the full dataset or partition of the dataset. The user will specify if he/she wants to perform the task on the whole dataset or a partition.

* Searching and returning a list of names of players who have scored goals greater than user input





* Searching and returning a list of names of players who have collected xG between the range provided by the user

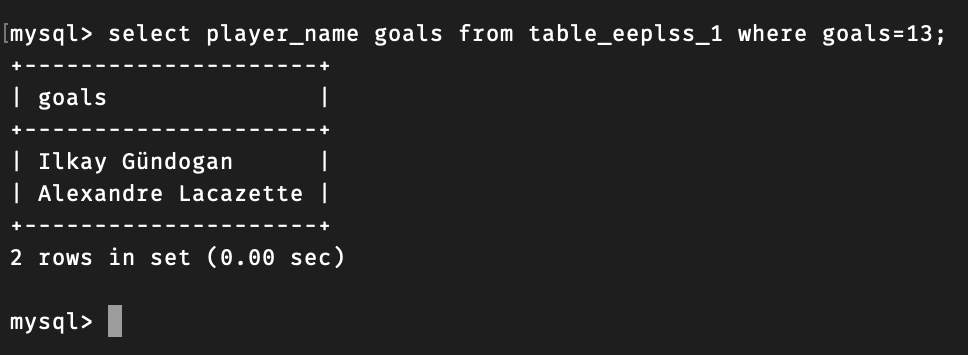
Text

Description automatically generated

Text

Description automatically generated

* Returning a list of players who have scored the number of goals given by the user.



Graphical user interface, text

Description automatically generated