**Assignment : Relational Database SQL**

28th JANUARY 2018

Team Members:

Prashant Wankhede (PRW19@pitt.edu)

Abhishek Shankarnarayanan (ABS100@pitt.edu)

**DDL Statements**

-- MySQL Script generated by MySQL Workbench

-- Sat Jan 27 22:58:19 2018

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema baseball

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema baseball

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `baseball` DEFAULT CHARACTER SET utf8 ;

USE `baseball` ;

-- -----------------------------------------------------

-- Table `baseball`.`players`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `baseball`.`players` (

`playerID` VARCHAR(50) NOT NULL,

`namefirst` VARCHAR(45) NULL,

`namelast` VARCHAR(45) NULL,

PRIMARY KEY (`playerID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `baseball`.`batting`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `baseball`.`batting` (

`playerID` VARCHAR(50) NOT NULL,

`yearID` INT NULL,

`teamID` VARCHAR(45) NULL,

`Games` INT NULL,

`Atbats` INT NULL,

`RunsScored` INT NULL,

`Hits` INT NULL,

`Homeruns` INT NULL,

`RunsBattedIn` INT NULL,

`Walks` INT NULL,

`strikeOuts` INT NULL,

`players\_playerID` VARCHAR(50) NOT NULL,

PRIMARY KEY (`playerID`),

INDEX `fk\_batting\_players\_idx` (`players\_playerID` ASC),

CONSTRAINT `fk\_batting\_players`

FOREIGN KEY (`players\_playerID`)

REFERENCES `baseball`.`players` (`playerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `baseball`.`halloffame`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `baseball`.`halloffame` (

`playerID` VARCHAR(50) NOT NULL,

`yearid` INT NULL,

`votedBy` VARCHAR(45) NULL,

`ballots` INT NULL,

`needed` INT NULL,

`votes` INT NULL,

`inducted` VARCHAR(45) NULL,

`category` VARCHAR(45) NULL,

`players\_playerID` VARCHAR(50) NOT NULL,

PRIMARY KEY (`playerID`),

INDEX `fk\_halloffame\_players1\_idx` (`players\_playerID` ASC),

CONSTRAINT `fk\_halloffame\_players1`

FOREIGN KEY (`players\_playerID`)

REFERENCES `baseball`.`players` (`playerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

USE `baseball` ;

-- -----------------------------------------------------

-- Placeholder table for view `baseball`.`view1`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `baseball`.`view1` (`id` INT);

-- -----------------------------------------------------

-- View `baseball`.`view1`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `baseball`.`view1`;

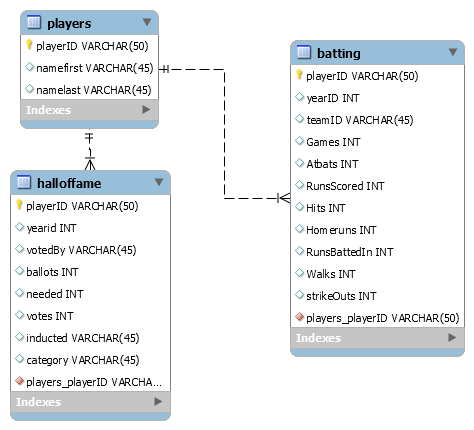
USE `baseball`;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**ER Diagram**



**Data Import Strategy**

**Python Codes**

**Batting​ ​Table:**

import​​ ​​​​csv

import​​ ​​​​MySQLdb

mydb=​ ​MySQLdb.connect(​"localhost"​​,​"root"​​,​"aaaaaa"​​,​"baseball "​​)

rowcount=​0

print​​ ​​​​'connected​​ ​​to​​ ​​data​​ ​​base'

cursor​ ​=​ ​mydb.cursor()

csv\_data=csv.reader(​file​(​'Batting.csv'​​),​quoting​=csv.QUOTE\_NONE)

print​​ ​​​​'file​​ ​​opened'

for​​ ​​​​row​ ​​in​​ ​​​​csv\_data:

​ ​​ ​​ ​​print​​ ​​​​row

​ ​​ ​​ ​​if​​ ​​​​rowcount>​0​:

​ ​​ ​​ ​​ ​​ ​​ ​​ ​cursor.execute(​'INSERT​​ ​​INTO

batting(playerID, yearID, teamID, Games, Atbats, RunsScored, Hits, Homeruns, RunsBattedIn, Walks, strikeOuts)'

​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​'VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'​​,row)

​ ​​ ​​ ​rowcount=rowcount+​1

mydb.commit()

cursor.close()

print​​ ​​​​'Done'

**Hall​ ​Of​ ​Fame​ ​Table:**

import​​ ​​​​csv

import​​ ​​​​MySQLdb

mydb=​ ​MySQLdb.connect(​"localhost"​​,​"root"​​,​"aaaaaa"​​,​"baseball "​​)

rowcount=​0

print​​ ​​​​'connected​​ ​​to​​ ​​data​​ ​​base'

cursor​ ​=​ ​mydb.cursor()

csv\_data=csv.reader(​file​(​HallOfFame.csv'​​),​quoting​=csv.QUOTE\_NONE)

print​​ ​​​​'file​​ ​​opened'

for​​ ​​​​row​ ​​in​​ ​​​​csv\_data:

​ ​​ ​​ ​​print​​ ​​​​row

​ ​​ ​​ ​​if​​ ​​​​rowcount>​0​:

​ ​​ ​​ ​​ ​​ ​​ ​​ ​cursor.execute(​'INSERT​​ ​​INTO halloffame (playerID, yearid, votedBy, ballots, needed, votes, inducted, category)'

​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​'VALUES(%s,%s,%s,%s,%s,%s,%s,%s)'​​,row)

​ ​​ ​​ ​rowcount=rowcount+​1

mydb.commit()

cursor.close()

print​​ ​​​​'Done'

**Player​ ​table:-**

import​​ ​​​​csv

import​​ ​​​​MySQLdb

mydb=​ ​MySQLdb.connect(​"localhost"​​,​"root"​​,​"aaaaaa"​​,​"baseball "​​)

rowcount=​0

print​​ ​​​​'connected​​ ​​to​​ ​​data​​ ​​base'

cursor​ ​=​ ​mydb.cursor()

csv\_data=csv.reader(​file​(​'Players.csv'​​),​quoting​=csv.QUOTE\_NONE)

print​​ ​​​​'file​​ ​​opened'

for​​ ​​​​row​ ​​in​​ ​​​​csv\_data:

​ ​​ ​​ ​​print​​ ​​​​row

​ ​​ ​​ ​​if​​ ​​​​rowcount>​0​:

​ ​​ ​​ ​​ ​​ ​​ ​​ ​cursor.execute(​'INSERT​​ ​​INTO

player(playerID,nameFirst,nameLast)'

​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​​​ ​​'VALUES(%s,%s,%s)'​​,row)

​ ​​ ​​ ​rowcount=rowcount+​1

mydb.commit()

cursor.close()

print​​ ​​​​'Done'

**Questions**

1.

a)how many players are inducted into the hall of fame?

**Answer:**

312

Select count(\*) from halloffame where inducted = ‘Y’;

b)how many distinct players in the batting data set are in the hall of fame?

1240

Select count (\*) from newtable (select Distinct playerID from batting, halloffame where batting.playerID=halloffame.playerID) as newtable;

2 a)what player has exactly 3000 hits total for his career?

**Answer:**

|  |  |
| --- | --- |
| Roberto | Clemente |

SELECT players.namefirst, players.namelast

FROM

(

SELECT batting.playerID,

SUM(batting.H) as Hits

FROM batting group by batting.playerID)

AS N1 WHERE newtable.Hits=3000) AS N2, players WHERE N2.playerID=players.playerID;

b)what year was he inducted into the hall of fame?

**Answer:**

1973

SELECT hall\_of\_fame.yearID

FROM hall\_of\_fame

WHERE hall\_of\_fame.PlayerID=

(

SELECT newtable.playerID

FROM

(

SELECT batting.playerID,

SUM(batting.H) as Hits

FROM batting group by batting.playerID)

AS newtable WHERE newtable.Hits=3000);

3 what player had the highest number of homeruns in 1919?

**Answer:**

|  |  |
| --- | --- |
| Babe | Ruth |

Select namefirst,namelast from players where players.playerID= (Select playerID from batting where SUM(Homeruns) as Sums Group by playerID Order By sums Limit 1);

4 what team has had the most hits after the year 2000, how many?

**Answer:**

Boston Red Sox, 23029

QUERY:

SELECT team.franchName, table2.hits FROM

(

SELECT table1.teamID,table1.HITS FROM

(

SELECT batting.teamID,SUM(batting.H) as HITS

FROM batting

WHERE batting.yearID >2000

GROUP BY batting.teamID) AS table1

ORDER BY table1.HITS DESC ) AS table2

INNER JOIN team ON table2.teamID = team.franchID

ORDER BY table2.hits DESC LIMIT 1;

a) What player has the most cumulative hits?

**ANSWER**: Pete Rose

b) How many does he have?

**ANSWER**: 4256

c) Is he in the hall of fame?

**ANSWER**: YES

d) How many votes has he received total?

**ANSWER**: 74

SELECT table2.nameFirst,table2.nameLast,table2.total as total\_hits,hall\_of\_fame.inducted,sum(hall\_of\_fame.votes) as Totalvotes FROM

(

SELECT player.playerID,player.nameFirst,player.nameLast,table1.total FROM

(

SELECT batting.playerID, SUM(batting.H) AS TOTAL

FROM batting GROUP BY batting.playerID) AS table1

INNER JOIN player on table1.playerID = player.playerID

ORDER BY table1.total desc LIMIT 1)

AS table2

INNER JOIN hall\_of\_fame ON table2.playerID= hall\_of\_fame.PlayerID

GROUP BY table2.nameFirst,table2.nameLast,table2.total,hall\_of\_fame.inducted;

**Query for next questions:**

CREATE VIEW view1 as

SELECT

table2.playerID,

table1.nameFirst, table1.nameLast,

table1.total\_games,table1.total\_atbats,table1.total\_runsScored,table1.total\_hits,

Table1.total\_hrs,table1.total\_rbis,table1.total\_walks,

table1.total\_sos,table2.inducted

FROM

(

SELECT batting.playerID, players.namefirst,players.namelast,

SUM(batting.Games) as total\_games, SUM(batting.Atbats) as total\_atbats,

SUM(batting.RunsScored) as total\_runsScored, SUM(batting.Hits) as total\_hits,

SUM(batting.Homeruns) as total\_hrs, SUM(batting.RunsBattedIn) as total\_rbis,

SUM(batting.Walks) as total\_walks, SUM(batting.strikeOuts)as total\_sos

FROM batting

INNER JOIN players on batting.playerID=players.playerID

GROUP BY batting.playerID

)

as table1 RIGHT JOIN

(

SELECT halloffame.PlayerID, halloffame.inducted FROM halloffame

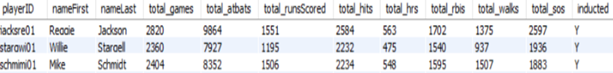
where inducted='Y'

)

AS table2 ON table1.playerID = table2.PlayerID;

6 a)what 3 players have the highest number of cumulative strikeouts over their career that have been inducted into the hall of fame?b)how many total strikeouts do they each have?

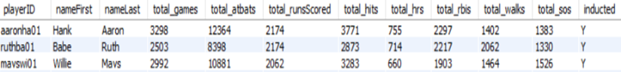
**Answer:**



SELECT \* FROM view1 ORDER BY view1.total\_sos desc LIMIT 3;

7 a)what 3 players have the highest cumulative total number of homeruns and have been inducted into the hall of fame?

**Answer:**



SELECT \* FROM view1 ORDER BY view1.total\_hrs desc LIMIT 3;

b)what 3 players have the highest cumulative total number of homeruns and have NOT been inducted into the hall of fame?

**Answer:**

|  |  |  |
| --- | --- | --- |
| Barry | Bonds | |
| Mark | McGwire | |
| Sammy | Sosa |

CREATE VIEW view1 as

SELECT

table2.playerID,

table1.nameFirst, table1.nameLast,

table1.total\_games,table1.total\_atbats,table1.total\_runsScored,table1.total\_hits,

Table1.total\_hrs,table1.total\_rbis,table1.total\_walks,

table1.total\_sos,table2.inducted

FROM

(

SELECT batting.playerID, players.namefirst,players.namelast,

SUM(batting.Games) as total\_games, SUM(batting.Atbats) as total\_atbats,

SUM(batting.RunsScored) as total\_runsScored, SUM(batting.Hits) as total\_hits,

SUM(batting.Homeruns) as total\_hrs, SUM(batting.RunsBattedIn) as total\_rbis,

SUM(batting.Walks) as total\_walks, SUM(batting.strikeOuts)as total\_sos

FROM batting

INNER JOIN players on batting.playerID=players.playerID

GROUP BY batting.playerID

)

as table1 RIGHT JOIN

(

SELECT halloffame.PlayerID, halloffame.inducted FROM halloffame

where inducted='N'

)

AS table2 ON table1.playerID = table2.PlayerID;

SELECT \* FROM view1 ORDER BY view1.total\_hrs desc LIMIT 3;

8 what player had the highest batting average from 1930 to 1940 (inclusively) with a cumulative total of hits over 1800 and has NOT been elected into the hall of fame? batting average = hits/atbats

**Answer:**

|  |  |
| --- | --- |
| Bill | Terry |

Select namefirst, namelast from players where players.playerID

IN

(

(Select playerID, (batting.Hits)/(batting.Atbats) as T from batting Order by T desc Limit 1)

Inner Join

( Select playerID from halloffame where yearID >= 1930 AND yearID <=1940 AND inducted = ‘N’ Limit 1));

9 what player averaged the highest number of walks from 1900-1920 (inclusively) with over 500 atbats?

**Answer:**

|  |  |
| --- | --- |
| Jimmy | Sheckard |
|  |  |

Select namefirst, namelast from players where players.playerID IN

(Select playerID, Walks from batting where

yearID >= 1900 AND yearID <= 1920 AND Atbats > 500

Order by Walks desc Limit 1);

10 write a query that uses a non equi join and explain what you are doing.

**Answer:**

As in the Question 8, following subquery was used.

Select playerID, (batting.Hits)/(batting.Atbats) as T from batting Order by T desc Limit 1

Inner Join

Select playerID from halloffame where yearID >= 1930 AND yearID <=1940 AND inducted = ‘N’;

The query will return all the playerID’s in batting table that matches with playerID’s in halloffame table with the condition that yearID is between 1930 and 1940 inclusive which satisfies Non Equijoin criteria.

11 write a query that uses an OUTER Join and explain what you are doing.

**Answer:**

Query that uses an OUTER Join

Select players.namefirst, players.namelast from players

Full Outer Join batting

On players.playerID=batting.playerID

Where players.playerID Is Null

Or batting.playerID is Null

Order by players.namefirst;

Consider a join of the batting table and the players table on their playerID columns. The results show only the players that are present in both the tables. The ISO FULL OUTER JOIN operator indicates that all rows from both tables are to be included in the results, regardless of whether there is matching data in the tables.

The Where clause is included with full outer join so that the query returns the rows such that there is no match between the tables.