**CS 425 – Database Organization**

**Fall 2023**

Homework 1.2

Group Members:

Shriya Prasanna (A20521733)

Girish Rajani-Bathija (A20503736)

Due Date: 9/10/23

Contributions:  
Both members completed all questions together, compared/discussed answers, and then took 1 file as a submission.

**Database Schema Implementation:**

CREATE DATABASE assignment2;

USE assignment2;

CREATE TABLE IF NOT EXISTS Team (

TmID CHAR(3) PRIMARY KEY,

ConfID CHAR(2),

Ranking INT,

Playoff CHAR(2),

Name VARCHAR(50),

Won INT,

Lost INT,

Games INT);

CREATE TABLE IF NOT EXISTS Coach(

BioID VARCHAR(25) PRIMARY KEY,

TmID CHAR(3),

Won INT,

Lost INT,

Games INT,

Stint INT,

Constraint FK\_Coach\_TmID FOREIGN KEY (TmID) REFERENCES Team (TmID));

CREATE TABLE IF NOT EXISTS Person(

BioID VARCHAR(25) PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

BirthDate VARCHAR(50),

BirthCity VARCHAR(50),

BirthCountry VARCHAR(50));

CREATE TABLE IF NOT EXISTS Player(

BioID VARCHAR(25) PRIMARY KEY,

TmID CHAR(3),

Played INT,

Started INT,

Minutes INT,

Points INT,

Attempts INT,

Made INT,

Constraint FK\_Player\_TmID FOREIGN KEY (TmID) REFERENCES Team (TmID));

**Loaded the data into MySQL Database using the csv files:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**SQL Query Solutions:**

1. Output the number of rows in each of your 4 relations (using 4 SELECT statements) in this order: Coach, Person, Player, Team. Call the result column LOADED each time.

**Coach Table**

SELECT COUNT(\*) AS LOADED FROM Coach;

A screenshot of a computer

Description automatically generated

**Person Table**

SELECT COUNT(\*) AS LOADED FROM Table;

**A screenshot of a computer

Description automatically generated**

**Player Table**

SELECT COUNT(\*) AS LOADED FROM Player;

**A screenshot of a test results

Description automatically generated**

**Team Table**

SELECT COUNT(\*) AS LOADED FROM Team;

A screenshot of a computer screen

Description automatically generated

1. Output everything in the Team relation, in ascending (increasing) order of TmID.

SELECT \* FROM Team ORDER BY TmID ASC;

A screenshot of a computer

Description automatically generated

1. Show the TmID of teams with Milwaukee in their name, but your query must cater for the fact that people spell this incorrectly – everyone starts with “MIL”, then somewhere later they have a “W” and even later a “K”. Example misspellings are Millwaukee, Milwakee, Milwuakee, Milwaukey, etc.

SELECT TmID FROM team WHERE Name like 'Mil%w%k%';

A screenshot of a computer

Description automatically generated

1. What are the lowest and the highest number of Games (played) in the coach data? Call the first result column LOWEST and the second result column HIGHEST.

SELECT MIN(Games) AS LOWEST, MAX(Games) AS HIGHEST FROM Coach;

A screenshot of a computer

Description automatically generated

1. Show the BioID, TmID, Points and Attempts for each player that scored more than 2000 points, in decreasing order of Points. Players with the same number of Points should be shown in alphabetical order of BioID.

SELECT BioID, TmID, Points, Attempts FROM Player WHERE Points > 2000 ORDER BY POINTS DESC, BioID

A screenshot of a computer

Description automatically generated

1. Show BioID of persons born in Gabon (GAB) and BioID of persons born in Egypt (EGY), if any.

SELECT BioID FROM Person where BirthCountry = 'GAB' OR BirthCountry = 'EGY';

A screenshot of a computer

Description automatically generated

1. Show the following for Jamesle01: his Points, the full Name of his team, & the BioID of his team’s coach.

SELECT Player.BioID AS Player\_Name, Points, Name AS Team\_Name, Coach.BioID AS Coach FROM Player,Team,Coach WHERE Player.BioID = 'Jamesle01' AND player.TmID = team.TmID AND coach.TmID = team.TmID;

**Second Approach Using Join Statement:**

select

player.BioID as Player\_Name

, player.points as Points

, team.Name as Team\_Name

, coach.BioID as Coach from player

join team on team.TmID=player.TmID and player.BioID='Jamesle01'

join coach on coach.TmID=team.TmID

A screenshot of a computer

Description automatically generated

1. Show the BioID of players whose BirthCountry is in the data, but their BirthCity is not in the data.

SELECT Player.BioID FROM Player,Person WHERE Player.BioID = Person.BioID AND BirthCountry <> '' AND BirthCity = '';

A screenshot of a computer

Description automatically generated

1. Which coach(es) Won the most games? Give BioID and the number Won.

SELECT Bioid, Won

FROM Coach

WHERE Won = (SELECT max(Won) FROM Coach );

A screenshot of a computer

Description automatically generated

1. What percentage of players have Points scored as zero? Call the result column NONSCORERS.

WITH perc AS (

SELECT count(points) AS noscore FROM player WHERE points = 0)

SELECT (noscore/count(BioID))\*100 AS NONSCORERS FROM player, perc GROUP BY noscore;

A screenshot of a computer

Description automatically generated

1. How many teams have Lost more games than they have Won? Call the result column LOSERS.

SELECT COUNT(Name) AS LOSERS FROM Team WHERE Lost > Won;

A screenshot of a computer

Description automatically generated

1. How many teams belong to each ConfID? Call the 2nd column CONFSIZE.

SELECT ConfID, COUNT(TmID) AS CONFSIZE FROM Team GROUP BY ConfID;

A screenshot of a computer

Description automatically generated

1. How many countries do the persons in this data come from? Call the result column NUMLANDS.

SELECT DISTINCT COUNT(BirthCountry) AS NUMLANDS FROM Person

A screenshot of a computer

Description automatically generated

1. Which pairs of teams have the exact same record (meaning the same values for Won and the same values for Lost)? Show their Won value and then their Lost value and then the 2 Names, making sure that the 3rd column Name is alphabetically before the 4th column Name so information is not repeated. Call the 3rd column TEAM1 and the 4th column TEAM2.

select t1.won,t1.lost,t1.name as TEAM1,t2.name as TEAM2 from team t1, team t2

where t1.won=t2.won and t1.lost=t2.lost and t1.name <> t2.name

order by Team1 asc

A screenshot of a sports team

Description automatically generated

1. For each of the 5 Rankings in ConfID “EC”, show the average number of games Lost by those teams. Call the 2nd column AVLOSSES.

SELECT Ranking, AVG(Lost) AS AVLOSSES FROM Team WHERE ConfID = "EC" GROUP BY Ranking;

A screenshot of a computer

Description automatically generated

1. Give a SQL statement to output “INVALID” if any information in any row (tuple) of any relation has invalid data (e.g., the Games value is not equal to Won value plus Lost value). If all the data is valid then it should output an empty table. Call the result column ANYPROBLEMS.

select

CASE

WHEN team.games = team.won + team.lost THEN

CASE

WHEN team.ranking<=5 and team.ranking > 0 THEN

CASE

WHEN player.points>=0 and player.Attempts >=0 THEN

CASE

WHEN coach.games = coach.won + coach.lost THEN ""

ELSE "INVALID" END END END

END AS ANYPROBLEMS

from team

inner join player on player.TmID=team.TmID

inner join coach on team.TmID = coach.TmID

inner join person as p on coach.BioID=p.BioID

inner join person as p1 on player.BioID=p1.BioID;

A screenshot of a computer

Description automatically generated