MLDS-413 Introduction to Databases and Information Retrieval

Homework 4: INNER JOINs

Name 1: _____

NetID 1: _____

Name 2: _____

NetID 2: _____

Instructions

You should submit this homework assignment via Canvas. Acceptable formats are word files, text files, and pdf files. Paper submissions are not allowed and they will receive an automatic zero.

As explained during lecture and in the syllabus, assignments are done in groups. The groups have been created and assigned. Each group needs to submit only one assignment (i.e., there is no need for both partners to submit individually the same homework assignment).

Each group can submit solutions multiple times (for example, you may discover an error in your earlier submission and choose to submit a new solution set). We will grade only the last submission and ignore earlier ones.

Make sure you submit your solutions before the deadline. The policies governing academic integrity, tardiness and penalties are detailed in the syllabus.

Homework Instructions

For this assignment, you will use the program "DB Browser for SQLite" (available at http://sqlitebrowser.org/). This is the same software we have worked with in class. I posted several sample database files on Canvas in the Lecture Materials page. These database files can be opened with the DB Browser for SQLite. The database files we will use for this homework are:

- EntertainmentAgency.sqlite
- BowlingLeague.sqlite
- SchoolScheduling.sqlite

For every question, we expect to see both your SQL code and the resulting data. Copy and paste both the SQL code and the results into a document and submit it following the submission instructions.

Here is an example question that applies to the SalesOrders.sqlite database:

Question: What bikes cost more than $1000?

Answer:

```
SELECT ProductName, RetailPrice
FROM Products JOIN Categories
     ON Products.CategoryID = Categories.CategoryID
WHERE CategoryDescription = "Bikes"
      AND RetailPrice > 1000;
```

Output:

```
"Trek 9000 Mountain Bike"  "1200"
"Eagle FS-3 Mountain Bike" "1800"
"GT RTS-2 Mountain Bike"   "1650"
```

You must answer each question with a single query.

You may find it helpful to use the "Basic SQL Cheat Sheet" posted on Canvas.

Each one of the questions below is worth **10 points**.

SchoolScheduling.sqlite

1) How many students are majoring in English or Mathematics? To receive credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

```
SELECT COUNT(studentID)
FROM    Students
   JOIN Majors ON Students.StudMajor = Majors.MajorID
WHERE Major = "English" OR Major = "Mathematics";

Output:
6
```

2) What is the full name of the instructor of the class that has the highest average students' grade? Your output should list the full name of the instructor, the Class ID, and the average students' grade of that class. The full name of the instructor can be formed by concatenating the last name, a comma, a space, and the first name. For example, the full name of the professor in this MLDS-413 class is the string "Hardavellas, Nikos". To receive credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

```
Full credit solution:
SELECT Staff.StfLastname || ", " || Staff.StfFirstName AS InstructorFullName,
       Classes.ClassID,
       AVG(Student_Schedules.Grade) AS AvgGrade
FROM    Student_Schedules
   JOIN Classes ON Student_Schedules.ClassID = Classes.ClassID
   JOIN Student_Class_Status
     ON (    Student_Schedules.ClassStatus = Student_Class_Status.ClassStatus
         AND Student_Class_Status.ClassStatusDescription = "Completed")
   JOIN Faculty_Classes ON Faculty_Classes.ClassID = Classes.ClassID
   JOIN Staff ON Staff.StaffID = Faculty_Classes.StaffID
GROUP BY Student_Schedules.ClassID
ORDER BY AvgGrade DESC LIMIT 1;

Output of full credit solution:
"Viescas, Caleb"  "2223"      "93.795"

Partial credit solution (9 points):
The query below calculates the average grade including students that completed the
class, withdrew, or are currently taking it. Students that withdrew or are currently
taking a class have a grade of 0, which skews the true grade average of the class.
Hence, this solution is awarded partial credit.

SELECT Staff.StfLastname || ", " || Staff.StfFirstName AS InstructorFullName,
 Classes.ClassID,
 AVG(Student_Schedules.Grade) AS AvgGrade
FROM    Student_Schedules
   JOIN Classes ON Student_Schedules.ClassID = Classes.ClassID
   JOIN Faculty_Classes ON Faculty_Classes.ClassID = Classes.ClassID
   JOIN Staff ON Staff.StaffID = Faculty_Classes.StaffID
GROUP BY Student_Schedules.ClassID
ORDER BY AvgGrade DESC LIMIT 1

Output of partial credit solution:
"Waldal, Deb"     "2410"      "93.6433333333333"
```

3) What is the percentage of students with majors in English or Mathematics? To receive full credit for this question you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all). To receive partial credit you **must** use the JOIN operator and you **must not** use subqueries in the WHERE clause (subqueries elsewhere are fine).

```
Full credit solution 1:
SELECT 100.0 * COUNT(DISTINCT S1.StudentID) / COUNT(DISTINCT S2.StudentID)
FROM    Majors
   JOIN Students AS S1 ON Majors.MajorID = S1.StudMajor
   JOIN Students AS S2
WHERE Majors.Major = "English" OR Majors.Major = "Mathematics";


Full credit solution 2:
SELECT 100.0 * SUM(StudMajor = MajorID) / COUNT(DISTINCT StudentID)
FROM students JOIN majors
WHERE Majors.Major = "English" OR Majors.Major = "Mathematics";


Partial credit solution (7 points):
This solution uses a subquery to calculate the percentage.

SELECT 100.0 * SUM(StuEngMath) / (SELECT COUNT(StudentID) FROM Students)
FROM (SELECT COUNT(StudentID) AS StuEngMath
      FROM    Students
         JOIN Majors ON Students.StudMajor = Majors.MajorID
      WHERE Major = "English" OR Major = "Mathematics");


Output:
33.333
```

EntertainmentAgency.sqlite

4) What percentage of all entertainer members are male entertainer members whose musical style is Jazz, and what percentage of all entertainer members are female entertainer members whose musical style is Jazz? You should provide a single query that outputs the percentages of each gender separately and indicates which is which. To receive full credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all). To receive partial credit you **must** use the JOIN operator and you **must not** use subqueries in the FROM and WHERE clauses (subqueries elsewhere are fine).

```
Full credit solution:
SELECT Gender,
      100.0 * COUNT(DISTINCT Members.MemberID) / COUNT(DISTINCT EM2.MemberID)
FROM    Entertainer_Members AS EM1
   JOIN Entertainer_Styles
     ON EM1.EntertainerID = Entertainer_Styles.EntertainerID
   JOIN Musical_Styles ON Musical_Styles.StyleID = Entertainer_Styles.StyleID
   JOIN Members ON Members.MemberID = EM1.MemberID
   JOIN Entertainer_Members AS EM2
WHERE Musical_Styles.StyleName = "Jazz"
GROUP BY Gender;
```

Partial credit solution:
This solution uses a subquery to calculate the percentage.

```
SELECT Gender,
      100.0 * COUNT(*) / (SELECT COUNT(DISTINCT MemberID) FROM
Entertainer_Members)
FROM    Entertainer_Members AS EM1
   JOIN Entertainer_Styles
     ON EM1.EntertainerID = Entertainer_Styles.EntertainerID
   JOIN Musical_Styles ON Musical_Styles.StyleID = Entertainer_Styles.StyleID
   JOIN Members ON Members.MemberID = EM1.MemberID
WHERE Musical_Styles.StyleName = "Jazz"
GROUP BY Gender;
```

Output:

```
"F"   "12"
"M"   "16"
```

5) What is the full name (in the form "LastName, FirstName") of the top 3 agents who have the highest average commission per engagement? The commission can be calculated by multiplying the contract price with the agent's commission rate. To receive credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

```
SELECT Agents.AgtLastName || ", " || Agents.AgtFirstName,
       AVG(Agents.CommissionRate * Engagements.ContractPrice) AS AvgCommision
FROM    Agents
   JOIN Engagements ON Agents.AgentID = Engagements.AgentID
GROUP BY Agents.AgentID
ORDER BY AvgCommision DESC LIMIT 3;
```

Output:
Note that we print out the average commission below just to provide some additional information. You only need to print out the names for full credit.

```
"Kennedy John"    "122.175"
"Viescas Carol"   "65.2631578947368"
"Smith Karen"     "60.1602941176471"
```

6) What is the total income of the Jazz entertainers (i.e., the sum of all Jazz entertainers' income across all of their engagements) and the total income of the Salsa entertainers? The income of each entertainer for each engagement is the ContractPrice of the engagement minus the agent's commission. To receive credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

```
SELECT StyleName,
       SUM(ContractPrice - ContractPrice*CommissionRate)
FROM    Engagements
   JOIN Agents ON Engagements.AgentID = Agents.AgentID
   JOIN Entertainer_Styles
     ON Entertainer_Styles.EntertainerID = Engagements.EntertainerID
   JOIN Musical_Styles ON Musical_Styles.StyleID = Entertainer_Styles.StyleID
WHERE Musical_Styles.StyleName in ("Jazz", "Salsa")
GROUP BY Musical_Styles.StyleID;
```

Output:

```
"Jazz"      "19623.3"
"Salsa"     "19115.7"
```

7) What are the top 5 musical styles that have the highest number of unique customers, and how many customers each of these styles has? To receive credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

```
SELECT Musical_Styles.StyleName,
       COUNT(DISTINCT Customers.CustomerID) AS NumCustomers
FROM    Engagements
   JOIN Customers ON Engagements.CustomerID = Customers.CustomerID
   JOIN Entertainer_Styles
     ON Engagements.EntertainerID = Entertainer_Styles.EntertainerID
   JOIN Musical_Styles ON Musical_Styles.StyleID = Entertainer_Styles.StyleID
GROUP BY Musical_Styles.StyleID
ORDER BY NumCustomers DESC
LIMIT 5;
```

Output of solution 2:

```
"Country"         "13"
"Show Tunes"      "12"
"60's Music"      "11"
"Top 40 Hits"     "11"
"Classical"       "10"
```

BowlingLeague.sqlite

8) Which teams have captains with the same last name? Each such team must be listed exactly once, along with the team captain's full name (in the form "LastName, FirstName"). To receive full credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

```
SELECT T1.TeamName AS Team1,
       B1.BowlerLastName || ", " || B1.BowlerFirstName AS Team1_CaptainName,
       T2.TeamName as Team2,
       B2.BowlerLastName || ", " || B2.BowlerFirstName AS Team2_CaptainName
FROM    Bowlers AS B1
   JOIN Teams AS T1 ON T1.CaptainID = B1.BowlerID
   JOIN Bowlers AS B2
   JOIN Teams AS T2 ON T2.CaptainID = B2.BowlerID
WHERE B1.BowlerLastName = B2.BowlerLastName
      AND T1.TeamID < T2.TeamID;
```

Output:

```
"Dolphins"  "Viescas, Suzanne"     "Manatees"  "Viescas, Michael"
```

9) In question 8 you identified the bowling teams that have captains with the same last name. List all the matches in which any of these teams participates. The output should provide the TourneyDate, TourneyLocation, odd and even Team Names, and Lanes. You can use the team names identified in question #8 here to make the query easier. To receive credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

Full credit solution:

```
SELECT TourneyDate, TourneyLocation, T1.TeamName, T2.TeamName, Lanes
FROM    Tourney_Matches
   JOIN Teams AS T1 ON T1.TeamID = Tourney_Matches.OddLaneTeamID
   JOIN Teams AS T2 ON T2.TeamID = Tourney_Matches.EvenLaneTeamID
   JOIN Tournaments ON Tournaments.TourneyID = Tourney_Matches.TourneyID
WHERE T1.TeamName IN ("Dolphins", "Manatees")
      OR T2.TeamName IN ("Dolphins", "Manatees");
```

Output of full credit solution (26 rows):

```
"1999-06-05"  "Red Rooster Lanes"   "Dolphins"    "Orcas"       "05-06"
"1999-06-05"  "Red Rooster Lanes"   "Manatees"    "Swordfish"   "07-08"
"1999-06-12"  "Thunderbird Lanes"   "Dolphins"    "Manatees"    "25-26"
"1999-06-19"  "Bolero Lanes"        "Manatees"    "Orcas"       "19-20"
"1999-06-19"  "Bolero Lanes"        "Dolphins"    "Swordfish"   "21-22"
"1999-06-26"  "Imperial Lanes"      "Marlins"     "Dolphins"    "09-10"
"1999-06-26"  "Imperial Lanes"      "Terrapins"   "Manatees"    "13-14"
"1999-07-03"  "Sports World Lanes"  "Dolphins"    "Sharks"      "13-14"
"1999-07-03"  "Sports World Lanes"  "Manatees"    "Barracudas"  "17-18"
"1999-07-10"  "Totem Lanes"         "Marlins"     "Manatees"    "05-06"
"1999-07-10"  "Totem Lanes"         "Terrapins"   "Dolphins"    "07-08"
"1999-07-17"  "Acapulco Lanes"      "Manatees"    "Sharks"      "15-16"
"1999-07-17"  "Acapulco Lanes"      "Dolphins"    "Barracudas"  "19-20"
"1999-07-24"  "Red Rooster Lanes"   "Manatees"    "Dolphins"    "25-26"
"1999-07-31"  "Thunderbird Lanes"   "Orcas"       "Manatees"    "19-20"
"1999-07-31"  "Thunderbird Lanes"   "Swordfish"   "Dolphins"    "21-22"
"1999-08-07"  "Bolero Lanes"        "Dolphins"    "Marlins"     "09-10"
"1999-08-07"  "Bolero Lanes"        "Manatees"    "Terrapins"   "13-14"
"1999-08-14"  "Imperial Lanes"      "Sharks"      "Dolphins"    "13-14"
"1999-08-14"  "Imperial Lanes"      "Barracudas"  "Manatees"    "17-18"
"1999-08-21"  "Sports World Lanes"  "Manatees"    "Marlins"     "05-06"
"1999-08-21"  "Sports World Lanes"  "Dolphins"    "Terrapins"   "07-08"
"1999-08-28"  "Totem Lanes"         "Sharks"      "Manatees"    "15-16"
"1999-08-28"  "Totem Lanes"         "Barracudas"  "Dolphins"    "19-20"
"1999-09-04"  "Acapulco Lanes"      "Orcas"       "Dolphins"    "05-06"
"1999-09-04"  "Acapulco Lanes"      "Swordfish"   "Manatees"    "07-08"
```

Partial credit solution (8 points):

Answers that show the matches only between Dolphins and Manatees will receive partial credit.

```
SELECT TourneyDate, TourneyLocation, T1.TeamName, T2.TeamName, Lanes
FROM    Tourney_Matches
   JOIN Teams AS T1 ON T1.TeamID = Tourney_Matches.OddLaneTeamID
   JOIN Teams AS T2 ON T2.TeamID = Tourney_Matches.EvenLaneTeamID
   JOIN Tournaments ON Tournaments.TourneyID = Tourney_Matches.TourneyID
WHERE T1.TeamName IN ("Dolphins", "Manatees")
      AND T2.TeamName IN ("Dolphins", "Manatees");
```

Output of partial credit solution:

```
"1999-06-12"  "Thunderbird Lanes"   "Dolphins"    "Manatees"    "25-26"
"1999-07-24"  "Red Rooster Lanes"   "Manatees"    "Dolphins"    "25-26"
```

10) How many teams have different players in the same team with the same last name? To receive credit you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

```
SELECT COUNT(DISTINCT B1.TeamID)
FROM    Bowlers AS B1
   JOIN Bowlers AS B2 ON B1.BowlerLastName = B2.BowlerLastName
WHERE B1.BowlerID != B2.BowlerID
      AND B1.TeamID = B2.TeamID;
```

Output

```
5
```