# EECS-317 Data Management and Information Processing

Homework 4: INNER JOINs

Name 1: Alicia Burris

NetID 1: anb0847

Name 2: Naomi Kaduwela

NetID 2:Nak133

# 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 (randomly) by the instructors. 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.

**Due Date: Thursday November 8, 11:59 pm**

# 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 Slides 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).

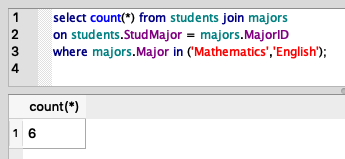
select count(\*) from students join majors

on students.StudMajor = majors.MajorID

where majors.Major in ('Mathematics','English');

Answer:

6



1. 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 our professor in this EECS 317 class is “Hardavellas, Nikos”. To receive credit you **must** **not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

SELECT Student\_Schedules.ClassID,

AVG(Grade),

StfFirstname|| " ," || StfLastname as TeacherName

FROM Student\_Schedules JOIN Faculty\_Classes

ON Student\_Schedules.ClassID=Faculty\_Classes.ClassID

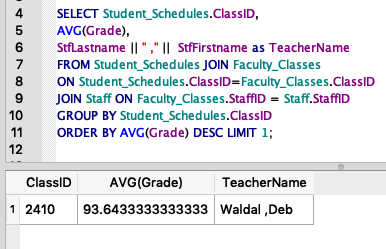
JOIN Staff ON Faculty\_Classes.StaffID = Staff.StaffID

GROUP BY Student\_Schedules.ClassID

ORDER BY AVG(Grade) DESC LIMIT 1;

Answer:

"2410" "93.6433333333333" "Deb ,Waldal"



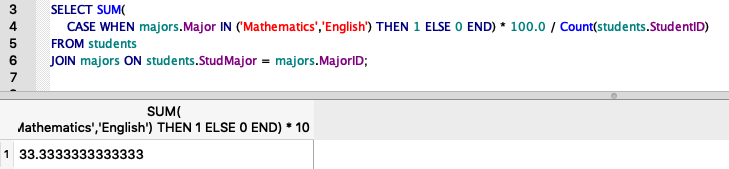
1. What is the percentage of students with majors in English or Mathematics? To receive full credit for this question you **must not** use subqueries anywhere (i.e., no nested SELECT clauses at all). To receive partial credit you **must** use the JOIN operator and you **must not** use subqueries in the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

**SELECT SUM(**

**CASE WHEN majors.Major IN ('Mathematics','English') THEN 1 ELSE 0 END) \* 100.0 / Count(students.StudentID)**

**FROM students**

**JOIN majors ON students.StudMajor = majors.MajorID;**

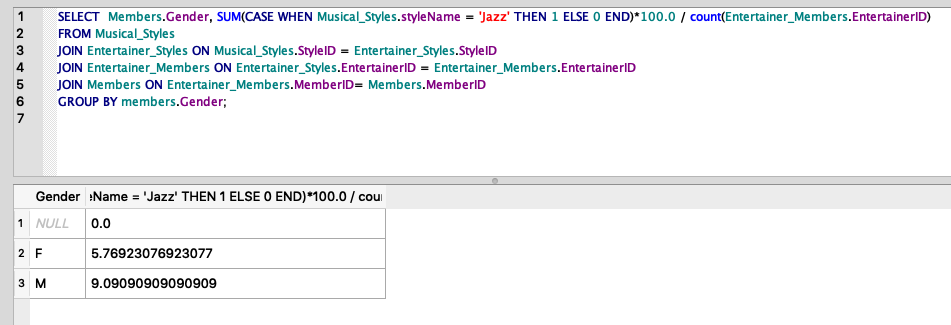
****

Answer:

33.3333333333333

# EntertainmentAgency.sqlite

1. What is the percentage of male and female entertainer members whose musical style is Jazz? The query should output the percentages of each gender separately and indicate 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).

****

select members.gender, sum(case when musical\_styles.styleName = 'Jazz' then 1 else 0 end) \* 100.0 / count(Entertainer\_Members.EntertainerID)

from Musical\_Styles

join Entertainer\_Styles on Musical\_Styles.StyleID = Entertainer\_Styles.StyleID

join Entertainer\_Members on Entertainer\_Styles.EntertainerID = Entertainer\_Members.EntertainerID

join Members on Entertainer\_Members.MemberID = Members.MemberID

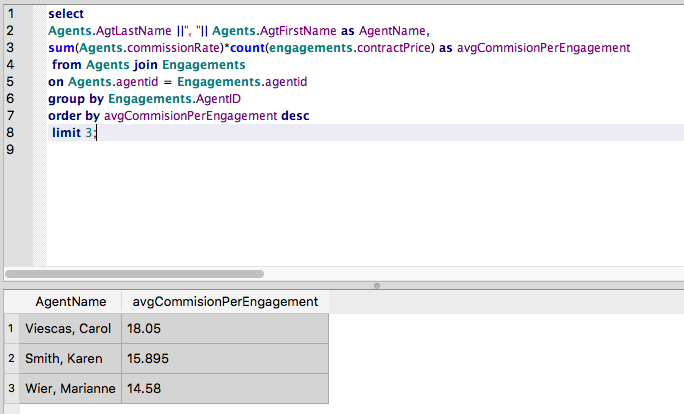
group by members.Gender;

Answer:

"F" "5.76923076923077"

"M" "9.09090909090909"

1. 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 as AgentName,

sum(Agents.commissionRate)\*count(engagements.contractPrice) as avgCommisionPerEngagement

from Agents join Engagements

on Agents.agentid = Engagements.agentid

group by Engagements.AgentID

order by avgCommisionPerEngagement desc

limit 3;

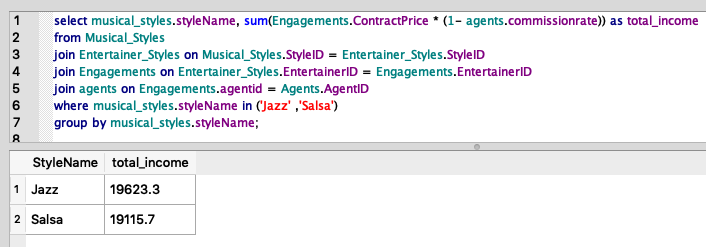
Answer:

"Viescas, Carol" "18.05"

"Smith, Karen" "15.895"

"Wier, Marianne" "14.58"

1. 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 musical\_styles.styleName, sum(Engagements.ContractPrice \* (1- agents.commissionrate)) as total\_income

from Musical\_Styles

join Entertainer\_Styles on Musical\_Styles.StyleID = Entertainer\_Styles.StyleID

join Engagements on Entertainer\_Styles.EntertainerID = Engagements.EntertainerID

join agents on Engagements.agentid = Agents.AgentID

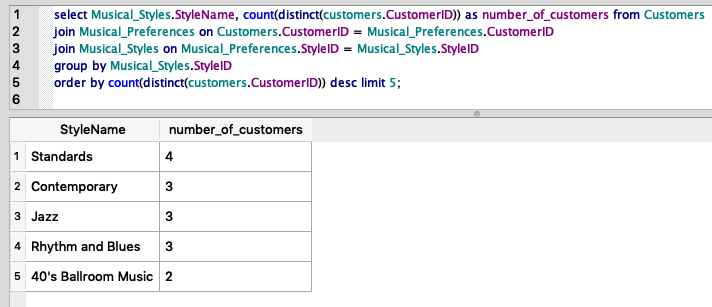
where musical\_styles.styleName in ('Jazz' ,'Salsa')

group by musical\_styles.styleName;

"Jazz" "19623.3"

"Salsa" "19115.7"

1. 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 number\_of\_customers from Customers

join Musical\_Preferences on Customers.CustomerID = Musical\_Preferences.CustomerID

join Musical\_Styles on Musical\_Preferences.StyleID = Musical\_Styles.StyleID

group by Musical\_Styles.StyleID

order by count(distinct(customers.CustomerID)) desc limit 5;

"Standards" "4"

"Contemporary" "3"

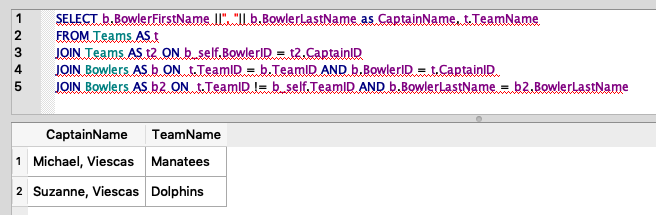
"Jazz" "3"

"Rhythm and Blues" "3"

"40's Ballroom Music" "2"

# BowlingLeague.sqlite

1. 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 b.BowlerFirstName ||", "|| b.BowlerLastName as CaptainName, t.TeamName

FROM Teams AS t

JOIN Teams AS t2 ON b\_self.BowlerID = t2.CaptainID

JOIN Bowlers AS b ON t.TeamID = b.TeamID AND b.BowlerID = t.CaptainID

JOIN Bowlers AS b2 ON t.TeamID != b\_self.TeamID AND b.BowlerLastName = b2.BowlerLastName

Output:

"Michael, Viescas" "Manatees"

"Suzanne, Viescas" "Dolphins"

1. List the TourneyDate, TourneyLocation, Team Names and Lanes of the teams that have captains with the same last name. You can use only the team name from the results of the previous question. To receive credit you **must** **not** use subqueries anywhere (i.e., no nested SELECT clauses at all).

select distinct Tournaments.TourneyDate, Tournaments.TourneyLocation, t1.teamname as OddTeam, t2.teamname as EvenTeam, Tourney\_Matches.Lanes

from Tournaments join Tourney\_Matches on Tournaments.TourneyID = Tourney\_Matches.TourneyID

join teams as t1 on t1.teamid = Tourney\_Matches.OddLaneTeamID

join teams as t2 on t2.teamid = Tourney\_Matches.EvenLaneTeamID

join Teams as t3 on t3.TeamID = Tourney\_Matches.OddLaneTeamID or t3.TeamID = Tourney\_Matches.EvenLaneTeamID

join Bowlers on t3.CaptainID = Bowlers.BowlerID

where Bowlers.bowlerlastname = 'Viescas';

"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"

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

