

CSci 4707
Homework 2
Fall 2015

Chapter 3, 4 and 5
Due Tuesday, 10/27/2015 13:00

A. (5 Points) Submission Guidelines

- All homeworks **must be typed**
- All homeworks due at the **BEGINNING** of the class in paper
- Late homework will get these points if the homework is typed
- **For Part D, in addition to the hard copy submission, please submit an SQL file via Moodle with the following format:**
<lastname>_<x500>_homework2.sql
- **No partial credit awarded for this section**

B. (25 Points) Chapter 3

Update: for this section, please check the correct ER diagram from the solution of Homework 1

- 1. (10 Points, 2.5 Points each)** Translate Homework 1 B1 Problem (a-d) into a correct SQL Tables. Write the correct CREATE TABLE SQL command for each table you created (You don't need to draw the table).
- 2. (10 Points)** Translate Homework 1 B2 Problem into a correct SQL Tables. Write the correct CREATE SQL command for each table you created (You don't need to draw the table).
- 3. (5 Points)** Translate Homework 1 B3 Problem into a correct SQL Tables. Write the correct CREATE SQL command for each table you created (You don't need to draw the table).

C. (30 Points) Chapter 4

1. (10 Points, 2 Points each) Consider the following schema:

- Students (sid: Integer, sname: String, year: Integer)
- Courses (cid: Integer, cname: String, department: String)
- OneStop (sid: Integer, cid: Integer, credits: Integer)

Write the following queries in Relational Algebra:

- Find the snames of students who took courses with name RDBMS and NoSQL.
- Find the snames of students who have taken all courses in the Computer Science department.
- Find all course id which is taken by at least two different students.
- Find pairs of student ids where the first student is more senior (higher year) than the second student.
- Find the student ids of students who have taken all Computer Science or all Electrical Engineering courses.

2. (10 Points, 5 Points each) Consider the following schema:

- SupplInfo (suppld: integer, prodl: integer)
- Purchases (purchaseld: integer, custId: integer, prodl: integer, purchaseMethod: integer)

This schema comes from a simple retail setting. The first table keeps track of which suppliers supply which products, and each (suppld, prodl) pair states that supplier number suppld supplies product number prodl. The second table keeps track of customer purchases – a purchase has an ID, a customer ID and a flag to specify whether the purchase was in person, online or over the phone.

Write the following queries in Relational Algebra over the above schema:

- Find all the pairs of supplids for suppliers that supply the exact same products. For example, supplier 1 supplies item A and B. Supplier 2 supplies item A. and Supplier 3 supplies item A and B. The pair of

Supplier 1 and 2 must not be in the answer since they do not supply the same exact product. However, Supplier 1 and 3 must appear in the answer. Each pair of suppls should appear only once in the answer.

- b. Find custlds of all customers who made purchases using exactly two different methods. E.g. if a customer made three purchases – one over the phone and two in person, their record is included in the answer. But if they made two purchases and both of them are online, their record is not included.

3. (10 Points) Consider the following relational schema:

- Bars(barId:integer, location:varchar(20))
- Beers(beerId:integer, name:varchar(20))
- Drinkers(dId:integer, age:integer)
- Serves(barId, beerId)
- Frequents(dId, barId)
- Likes(dId, beerId)

It contains information about bars, beers and drinkers. A tuple (bar1, beer1) belongs to the Serves relation iff the bar bar1 serves the beer beer1. A tuple (drinker2, bar2) belongs to the Frequents relation iff the drinker drinker2 frequents the bar bar2. A tuple (drinker3, beer3) belongs to the Likes relation if the drinker drinker3 likes the beer beer3. You can assume that each bar serves at least one beer. You can also assume that each drinker likes at least one beer and frequents at least one bar.

Write the following queries in Relational Algebra: “Find the ids of the drinkers who only frequent bars that serve at least one beer that they like”. In other words, you should compute the set {dId | dId is a drinker and for each bar barId that dId frequents, barId serves some beer beerId that dId likes}.

D. (40 Points) Chapter 5

Note:

In addition to the hard copy submission, please submit an SQL file via Moodle with the following format: <lastname>_<x500>_homework2.sql

Your answer must be able to be executed on Oracle in the CSE machine (please test this). Unrunnable script will receive 0 point for this section. Please check the naming of your answer (table, attributes, etc.) so it matches with the problem.

1. **(10 Points, 5 Points each)** Consider Problem C2 above. For each query, write the SQL statement. All your SQL statement must be a single query.
2. **(10 Points)** Consider Problem C3 above. Write the following query in SQL, **without** using UNION or EXCEPT: “Find all the pairs (barId1, barId2) such that barId1 and barId2 are the ids of two different bars that serve exactly the same set of beers.” Each pair of barIds should appear only once in the answer.
3. **(10 Points)** Consider the following schema (keys are underlined):
 - Customer(cId: integer, cName: varchar(20))
 - Buys(tid: integer, cId: integer, pId: integer)
 - Product(pId: integer, pName:varchar(20))

This schema carries information about customers who purchase products in a transaction (captured by the transaction identified by tid in the Buys relation).

Write the following query in SQL **without** using EXCEPT: “Find the names of all the customers who have only bought products bought by every other customer.”

4. **(10 Points)** Consider the following relational schema (keys are underlined):
 - Actors(aId: integer, name: varchar(20))

- Directors(dId: integer, name: varchar(20))
- Movies(mId: integer, name: varchar(20), dId: integer)
- Casts(aId: integer, mId: integer)

Write the following query in SQL: “List the actor names of actors cast only in movies directed by Spielberg” (Note: They do not have to be cast in all movies by Spielberg, but they cannot be cast in any movie that is not by Spielberg).