

CSci 4707  
Homework 2  
Fall 2016

Chapter 4, 5, 8 and 9  
Due Thursday, 10/27/2016 13:00

**Submission Guidelines:**

- All homeworks **must be typed (not scanned or photographed from a handwriting).**
- All homeworks due at the **beginning of the class in paper.**

**Any submissions that do not follow the guideline will receive 0 point.**

## A. (40 Points) Chapter 4

1. (10 Points, 5 Points each) Consider the following schema:

- Customer (cid: Integer, cname: String, age: Integer)
- Product (pid: Integer, pname: String, type: String)
- Buy (cid: Integer, pid: Integer, price: Integer)

Write the following queries in **Relational Algebra**:

- Find the cnames of customers who buy products with name 'iPhone' and 'iPad'.
- For every product that is bought by a customer named "Bob", what is the id of the most expensive product that he bought?

2. (20 Points, 10 Points each) Consider the following schema:

- Enroll (sid: integer, cid: integer)
- Onestop (id: integer, sid: integer, cid: integer, method: integer)

The first table keeps track of which students take which courses. The method in the second table specifies whether the enrollment was in classroom, online or outside the university and many more.

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

- Find all the pairs of sids for students that enroll the **exact same courses**. For example, student 1 enrolled in class 10 and 20. Student 2 enrolled in class 10. Student 3 enrolled in class 10 and 20. The pair of Student 1 and 2 must not be in the answer since they do not take the same exact courses. However, Student 1 and 3 must appear in the answer. Each pair of sid should appear only once in the answer.
- Find sids of all students who takes classes using **exactly** two different methods.

**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 whose all their frequent bars 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}.

## B. (30 Points) Chapter 5

**Note:** For this section, all your SQL statement must be a **single query**. (You **cannot** store a temporary result in a VIEW, you **cannot** use UNION / INTERSECTION / MINUS / EXCEPT. However, you may use nested queries)

1. **(10 Points, 5 Points each)** Consider Problem B2 above. For each query, write the SQL statement without using EXCEPT. All your SQL statement must be a single query.
2. **(10 Points)** Consider the following schema (keys are underlined):
  - Customer(cId: integer, cName: varchar(20))
  - Buys(tid: integer, cId: integer, pld: integer)
  - Product(pld: 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: “Find the names of all the customers who have only bought products bought by every other customer.”

3. **(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).

**C. (30 Points) Chapter 8 and 9**

- 1. (10 Points, 5 Points each)** Consider a file that is stored on disk and the size of the file is much larger than the size of the buffer. Three different buffer management policies are available: FIFO, LRU, and MRU. How does each buffer manager replacement policy perform on each condition below? Which buffer management policy is the best (if any)? Clearly state the reason for your choice.
  - a. (5 Points)** The entire file is sequentially scanned many times.
  - b. (5 Points)** File pages are accessed according to a Zipfian distribution, i.e., popular items are requested very frequently, and unpopular items are requested rarely.
- 2. (10 Points)** Consider the following operation: a delete operation specified with an equality condition. For each of the following file organizations: Heap File, Sorted File, Clustered B+tree, Unclustered B+tree, and Unclustered Hash Index, what is the cost of the deletion if no record qualifies? What is the cost if the condition is not on a key?
- 3. (10 Points, 5 Points each)** Answer the following questions:
  - a.** Will equality search be better using Unclustered B+Tree rather than Sorted?
  - b.** Is it possible to have an Unclustered B+Tree index with Alternative 1? If yes, what is the advantage of using Alternative 1 compared to Alternative 2 or 3? If no, why not?