CSci 4707
Homework 2
Spring 2015

Chapter 3, 4 and 5
Due Thursday, 02/26/2015 14:30

A. (5 Points) Submission Guidelines

- All homeworks **must be typed**
- All homeworks due at the **beginning of the class in paper**
- **No partial credit awarded for this section**

B. (25 Points) Chapter 3

1. **(10 Points, 2.5 Points each)** Imagine you are designing a Database for a Supermarket. The Database will contain information about its Customers (identified uniquely by customer id/custid) and the Products they bought (identified uniquely by product id/pid). For simplicity, each transaction will be timestamped each day (identified by date id/did) and each customer can only buy a specific product once a day. For each situation below, state the SQL command in creating the table. **NOTE: These are different from Homework 1**
    a. We want to keep track only the last time each customer buy a product. For example, Customer A bought Product B on day 1 and buy the same thing on day 2. We will only keep track the last transaction which is on day 2.
    b. We want to keep the history of all transactions information made by each customer. According to the previous example (Part a), we will keep track both transactions.
    c. Every customer must buy at least one product and we want to keep track the last time the customer buy a specific product.
    d. Every customer must buy exactly one product and we want to keep track the last time the customer buy the product.

2. **(10 Points)** Translate Homework 1 C2 Problem into a correct SQL Tables. Write the SQL command for each table you created (You don't need to draw the table).

3. **(5 Points)** Translate Homework 1 C3 Problem into a correct SQL Tables. Write the 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:
   a. Find the snames of students who took courses with name RDBMS and NoSQL.
   b. Find the snames of students who have taken all courses in the Computer Science department.
   c. Find all course id which is taken by at least two different students.
   d. Find pairs of student ids where the first student is more senior (higher year) than the second student.
   e. 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:
   - SuppInfo (suppid: Integer, prodid: Integer)
   - Purchases (purchaseid: Integer, custid: Integer, prodid: Integer, purchaseMethod: Integer)

   This schema comes from a simple retail setting. The first table keeps track of which suppliers supply which products, and each (suppid, prodid) pair

states that supplier number suppid supplies product number prodid. 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:

a. Find all the pairs of suppids for suppliers that supply the exact same products. Each pair of suppids should appear only once in the answer.
b. Find custids 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, both online, their record is not included.

3. **(10 Points)** Consider the following relational schema:
   - Bars(barId, location)
   - Beers(beerId, name)
   - Drinkers(did, age)
   - 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

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."

3. **(10 Points)** Consider the following schema (keys are underlined):
   - Customer(<u>cid</u>, cname, age, gender)
   - Buys(<u>tid</u>, cid, pid)
   - Product(<u>pid</u>, pname, type, mfgr, price)

   This schema carries information about customers who purchase products in a transaction (captured by the transaction identified by tid in the Buys relation). Note that a customer can purchase the same product more than once.

   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(<u>aid</u>, name)
   - Directors(<u>did</u>, name)
   - Movies(<u>mid</u>, name, year, did)

- Cast(aid, mid)

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).