Virginia Tech. **CS 4604 – Introduction to DBMS**
Computer Science **Spring 2016, Prakash**

Homework 1: Relational Algebra and SQL

(due February 10th, 2016, 4:00pm, in class—hard-copy please)

Reminders:

a. Out of 100 points. Contains 6 pages.
b. Rough time-estimates: 3~6 hours.
c. Each HW is supposed to be done individually.
d. Please type your answers. Illegible handwriting may get no points, at the discretion of the grader. Only drawings may be hand-drawn, as long as they are neat and legible.
e. There could be more than one correct answer. We shall accept them all.
f. Whenever you are making an assumption, please state it clearly.
g. Unless otherwise mentioned, you may use any SQL/RA operator seen in class/in textbook.
h. Unless otherwise specified, assume set-semantics for RA and bag-semantics for SQL.
i. Feel free to use the linear notation for RA and create intermediate views for SQL.
j. Lead TA for this HW: Sorour Amiri.

Q1. RA: Warming up [12 points]

Consider the following three tables, T1, T2 and T3:

T1

A	B	C	D
10	a	5	10
15	b	8	9
25	a	6	0

T2

E	B
1	b
5	c
6	b

T3

A	N
10	b
25	c
10	b

Show the results of the following relational algebra queries (3 points each):

Q1.1. $\sigma_{D<5}(T1)$

Q1.2. $\sigma_{E>2 \wedge B=b}(T2)$

Q1.3. $\sigma_{D<4}(T1 \bowtie T2 \bowtie T3)$

Q1.4. $T1 \bowtie_{T1.C=T2.E} T2$

Q2. RA: Monotone Operators [15 points]

An operator on relations is said to be *monotone* if whenever we add a tuple to one of its arguments, the result contains all the tuples that it contained before adding the tuple, plus perhaps more tuples. Which of the 5 fundamental RA operators we saw in class (see Lecture 2) are monotone? For each either explain why it is monotone, or give a small example showing it is not.

Q3. RA: Products and Manufacturers [22 points]

Consider the following relational database that stores information about computer hardware products (keys are underlined, field types are omitted):

Product (maker, model, type)
PC (model, speed, ram, hd, price)
Laptop (model, speed, ram, hd, screen, price)
Printer (model, color, type, price)

The Product relation gives the manufacturer, model number and type (PC/laptop/printer) of various products. The PC relation gives for each model number that is a PC the speed (of the processor, in gigahertz), the amount of RAM (in GB), the size of hard disk (in GB), and the price. The Laptop relation is similar, with screen size included. The Printer relation records whether it is a color printer or not (Yes/No), and the type (laser/ink-jet) and the price.

Write the following queries in relational algebra:

Q3.1. (2 points) Find the model numbers of all color laser printers.

Q3.2. (3 points) Find the model number and price of all products (of any type) made by manufacturer 'HP'.

Q3.3. (5 points) Find those manufacturers of at least two different computers (PCs or laptops) with speeds of at least 2.80.
Hint: You may need a 'self' Cartesian product and a join.

Q3.4. (6 points) Find the manufacturers who sell exactly three different models of PC.
Hint: Similar in spirit to Q3.3, but you may have to use the set difference as well.

Q3.5. (6 points) Find the manufacturers who sell PCs with at least all the RAM sizes seen in laptops.

Q4. SQL: The Restaurant DB [21 points]

The following relational database schema stores information about a restaurant. The primary keys are underlined and field types are omitted. Study the schema and its corresponding description to answer the questions.

Customer (cid, cname, cage, caddress, cphone)
Table(tno, capacity, availability)
Item(iid, iname, price)
Order(oid, cid, iid, tno, quantity)
Employee(eid, ename, eaddress, eage, salary)
Reservation(rid, eid, cid, date, time)

The customer is someone who eats in the restaurant. The Customer relation provides customer id, the name, age, address and phone number information. The Table relation gives the restaurant's occupancy information that includes the table number, the number of people who can sit on the table, and the table's availability information. The restaurant serves several items. Each of the Items has an identification number (e.g., 82285), name (e.g., salad), and unit price (e.g., $15). Soon after arrival in the restaurant a customer places an order. Order relation contains order id, customer id, ordered items id, each item's quantity. Each order contains one or more items, and the quantity of each item (e.g., order 10 contains one soup and two salads). Here we are assuming that a customer can visit the restaurant at most once in a day and can't place multiple orders at a time. Restaurant employees serve customers. The Employee relation contains employee id, name, address, age, and salary information. The customer can make a reservation in the restaurant beforehand. The Reservation relation records reservation, employee, and customer ids, and reservation date and time.

In this assignment, you will only deal with querying part of SQL. You are NOT allowed to tamper with (change the contents of) the database, i.e., CREATE, INSERT, DELETE, ALTER, UPDATE etc.

Write SQL queries that answer the questions below (one query per question). The query answers must not contain duplicates, but you should use the SQL keyword distinct only when necessary. For this question, creation of temporary tables is NOT allowed, i.e., for each question you have to write exactly one SQL statement (possible using nested SQL).

Q4.1. (3 points) Find the customer names that ordered "Steak".

Q4.2. (3 points) Find the name of the employee who made the reservation for both customers "James Schlitt" and "Nicole Diamond".

Q4.3. (4 points) Find the name of the highest paid employee(s) whose age is more than 30.

Q4.4. (5 points) How many customers ordered more than three items?

Q4.5. (6 points) Find those table numbers where people ordered more than $30 worth of food.

Q5. SQLite: Storing Employee Information [30 points]

This question is on a consumer complaint database that stores information about the complaints government has received about financial products and services. Download and install SQLite3 from http://www.sqlite.org. Feel free to use SQLite3 for testing and practicing SQL queries in general.

Warm-up

Follow the documentation and load the small sample database at: http://courses.cs.vt.edu/~cs4604/Spring16/homeworks/hw1/cs4604-hw1.db

It has a table "complaints" which includes Complaint ID (Primary Key), Date received, Product, Issue, Consumer complaint narrative, Company public response, Company, State, ZIP code, Date sent to company, Company response to consumer, Timely response? (Yes-No), Consumer disputed? (Yes-No).

As a sanity check that you have the correct database, running the following command at a Unix/Linux/Cygwin prompt:

```
your-machine% sqlite3 cs4604-hw1.db 'select count(*) from complaints;'
```

should return 50

We want to write SQL queries to do the following:

- **Query1:** Return distinct Issues about "Credit Karma Inc." company.
  Hint: you have to look at the Company column.
- **Query2**: Count the number of Products in the database.

Larger CSV file

A bigger raw comma separated value (csv) file is given here: http://courses.cs.vt.edu/~cs4604/Spring16/homeworks/hw1/complaints.csv

It is a subset from consumer complaint dataset (if you are curious, the official dataset is at http://catalog.data.gov/dataset/consumer-complaint-database). We want to write queries to do the following:

- **Query3**: List companies which have at least one product in common with "Equifax" company.
- **Query4**: Return all states and total number of complaints reported from them**.**

Life without SQL

Finally, imagine you do not have access to a DBMS. In your favorite language (Python/Perl/Ruby/Java/C++ etc.) write code to do both queries above (Query3 and Query4) on the csv data file directly.

Deliverables

Q5.1. (2 points) The SQL query for Query1.

Q5.2. (2 points) The SQL query for Query2.

Q5.3. (2 points) The output of running Query1 in SQLite on the small sample database.

Q5.4. (2 points) The output of running Query2 in SQLite on the small sample database.

Q5.5. (4 points) The SQL query for Query3.

Q5.6. (4 points) The SQL query for Query4.

Q5.7. (3 points) The output of running Query3 on the csv file after loading it in SQLite.

Q5.8. (3 points) The output of running Query4 on the csv file after loading it in SQLite.

Q5.9. (4 points) Hard copy of your python/perl/etc code for doing Query3 on the raw csv file directly.

Q5.10. (4 points) Hard copy of your python/perl/etc code for doing Query4 on the raw csv file directly.

Hints:

For loading the csv file,

- The end-of-line convention follows the DOS format (CRLF).
- Use the .import and .mode csv commands of sqlite3 or check the link: https://www.sqlite.org/cli.html
- A cheat sheet for sqlite3 commands http://www.cheatsheets.org/own/sqlite/Syntax.Diagrams.For.SQLite.html
- As a sanity check, the command

```
your-machine% wc -l complaints.csv
```

should return

```
2081 complaints.csv
```