CSc 460 — Database Design
Spring 2017 (McCann)

http://www.cs.arizona.edu/classes/cs460/spring17/

Homework #2: Relational Algebra Queries

Due Date: March 8th, 2017, at the beginning of class

Name: _____

Directions: See the next two pages for the homework instructions.

On or before the due date, by the start of class, hand-in a printout of your solutions with this cover page stapled to the front **and** submit your electronically–formatted version of your solutions (the turnin folder is `cs460h2`). If you need to submit your solutions within the 24-hour late window, place your printout in the CS 460 mailbox (bottom row) in CS 713 as soon as you are able to do so.

	— Points —	
Question #	Possible	Deductions
1	5	
2	5	
3	5	
4	5	
5	5	
6	10	
7	10	
8	10	
9	10	
10	10	
11	10	
12	15	
Totals:	100	

Score:

(Continued . . .)

Overview: Becoming proficient in formulating useful database queries takes practice. Knowing how to use pure Relational Algebra is useful background for learning SQL. To use SQL, you may not need to call the Relational Algebra operators directly, but you do need to specify the critical parts of them. It helps to know how those parts fit in to the query.

Software: Richard Leyton, then a student at Oxford Brookes University, wrote a simple DBMS called LEAP as a project. The current version of LEAP is 1.2.6 and runs reasonably well under the LINUX operating system. The syntax of its relational algebra commands differs a bit from what we use in class, but converting between the notations is not hard. There does exist a version of LEAP for Windows, but I have never used it and, given its age, cannot recommend it.

To install LEAP into your lab account, here's what you need to do:

1. From your home directory, type this: `/home/cs460/spring17/leap/scripts/users/leapinstall`
2. When asked to supply the LEAP source directory, respond with this: `/home/cs460/spring17/leap`
3. When asked to supply the target directory, just press Enter. It will default to a subdirectory named `leap` in your account.

To run leap, here's what you need to do:

1. Change directory to your local LEAP bin subdirectory: `cd leap/bin`
2. Run leap: `./leap`
3. Give leap the command `use company` to select our sample database.

To execute the sample query I've provided (in `leap/database/company/source/sample.src` in your account), run leap (see the three steps above) and give this LEAP command: `@ sample` Please note that for this command to work, your source files must be in that directory (`leap/database/company/source`).

Here's a brief list of useful LEAP commands and their syntax:

Operation	**General Format**	**Example of Use**
Use	use *database*	use rental
Select	select (*relation*) (*condition*)	r1=select (borrow) (amount='1000')
Project	project (*relation*) (*attr. list*)	r2=project (spj) (sno,qty)
Join	join (*rel1*) (*rel2*) (*condition*)	j=join (spj) (p) (spj.pno=p.pno)
Union	(*relation*) union (*relation*)	u=(employee) union (manager)
Intersection	(*relation*) intersect (*relation*)	int=(employee) intersect (manager)
Difference	(*relation*) difference (*relation*)	m=(employee) difference (manager)
Cartesian Product	(*relation*) product (*relation*)	prod=(s) product (spj)
Display a Relation	display *relation*	display prod
Copy a Relation	duplicate (*relation*)	copyofs = duplicate (s)
Execute a Source File	@ *filename*	@ sample
Quit LEAP	quit	quit

(Unfortunately (for you!), LEAP does not have the division operator.) Other LEAP commands can be learned by reading the on–line help (type: `help` from within LEAP to get started) or the documentation files in the doc and help subdirectories. Be aware that every attribute is of type STRING or INTEGER, and all constants have to be specified in single quotes (yes, including integers!).

Assignment: The database already contains some relations for a company database in LEAP format. Here are their schemas.

```
Employee (fname, minit, lname, ssn, bdate, address, sex, salary, superssn, dno)
Dept_Locations (d#, dlocation)
Department (dname, dnumber, mgrssn, mgrstartdate)
Works_On (empssn, pno, hours)
Proposal (pname, pnumber, plocation, dnum)
Dependent (essn, dependent_name, sex, bdate, relationship)
```

I've underlined the primary key fields. Foreign keys should be easy to identify, as they have names similar to the corresponding primary keys.

Your task is to create relational algebra queries that correctly answer the following questions. If a query is impossible to answer using LEAP, explain why. (You will want to check with the class staff or your fellow students before concluding that a query cannot be answered.)

1. List all attributes of the employees from department 5.
2. What are the SSNs of the employees working on proposal #10?
3. What is the Cartesian Product of the employees' first names and the dependents' names?
4. (You must **not** use JOIN for this query) At which locations is the research department located?
5. (You must use JOIN for this query) At which locations is the research department located?
6. What are the names of the departments with employees who have a dependent named Alice?
7. Retrieve the SSNs of the employees who work in department 5 or who directly supervise an employee in department 5.
8. What are the birthdates of the managers of the proposals from Stafford?
9. What are the names of the departments managed by people who have dependent female spouses?
10. What are the names of the male dependents who are also dependent on a male employee? (Use $\cap$)
11. What are the salaries of the employees from department 5 who are NOT working on ProductY?
12. What are the names of the departments which employ all genders? (In other words, the departments that have at least one woman and at least one man.) (Yup, this is the $\div$ query.)

Hand In: (1) Turn in a printout that shows your Relational Algebra queries and the answers LEAP produces when it runs them. Please produce the answers in the same order as the questions are listed, and clearly number each of your answers (you can do that by hand). Staple this handout to the front, being sure to write your name on the first page. (2) Submit your LEAP queries (as a `tar` file) using turnin. The submission folder is `cs460h2`.

Want to Learn More About LEAP? Visit `http://leap.sourceforge.net/` (on–line help files!)

Other Requirements and Hints:

- You can easily capture LEAP's output to a file by running LEAP within the script command. First type `script`, then run LEAP, then type `exit`. Everything you saw on the screen is saved in a file named `typescript`. Helpful hint: **Never** run a text editor from within `script`!
- In LEAP, you can create a `.src` file (named `query1.src`, for example) in the `leap/database/company/source` directory in your account, and type into it the sequence of operations needed to answer question #1, above. To execute a file of LEAP commands, type @ followed by the name of the .src file you want to execute. (For example: `@ query1`) Do this while running LEAP, of course. This is a convenient mechanism for storing your queries and for easily creating your final output for submission on the due date.
- I have attempted to write solutions to all of the assigned queries, and believe them to be possible to answer using LEAP. Feel free to help each other out with workarounds, etc., to LEAP's quirks, but write your own queries. (If you pattern your script file(s) after the `sample.src` file I've provided, things should work fine.)
- LEAP may have problems dealing with temporary relations of high degree; if you have problems, remove extra attributes before performing joins.
- When LEAP has a core dump, it doesn't clean up after itself, and as a result it can fail to restart. A simple solution: Save copies of your `.src` files, and reinstall LEAP.
- And finally: Please remember that a correct answer is a query that produces the correct result *in a logically correct way*! Write queries that will work even if the relations' content changes.