CS127 Homework #1

Due: September 20th, 2017 2:59 P.M.

Handing In

Upload your homework to gradescope.
Please write your Banner ID on your submission. Do not write your name on the submission.

Grading Information

Grading for the homeworks works as follows:

- One warmup problem will be graded at random.
- All other problems will also be graded in detail and will be given a score.

Solutions for the warmup problems will be provided along with your graded work.

Warmup #1

Given the following database schema:

owner(owner_name, age)
owns(owner_name, num_cats)

Write relational algebra expressions to accomplish the following tasks:

1. Find the names of all owners above the age of 50.

$$\Pi_{\text{owner\_name}}(\sigma_{\text{age} > 50}(\text{owner}))$$

2. Find the names of all owners who own fewer than 5 cats.

$$\Pi_{\text{owner\_name}}(\sigma_{\text{num\_cats} < 5}(\text{owner} \bowtie \text{owns}))$$

3. Find the names of all owners who are below the age of 25 and own more than 4 cats.

$$\Pi_{\text{owner\_name}}(\sigma_{\text{age}<25 \ \wedge \ \text{num\_cats} > 4}(\text{owner} \bowtie \text{owns}))$$

Warmup #2

Consider the following database schema:

employee(employee_id, name)
company(company_id, company_name)

consulting(e_id, c_id)

This schema represents the scenario wherein an employee (of a consulting business) can only consult for one company at a time. The *consulting* relation shows which company each employee is currently consulting for. The *c_id* attribute corresponds to a company's *company_id* attribute, and the *e_id* attribute corresponds to an employee's *employee_id* attribute.

Now say we want the database to reflect that an employee can consult for multiple companies simultaneously. Should *e_id* remain a primary key of the *consulting* relation? If not, what should be the new primary key?

No. The new primary key should be (*c_id, e_id*).

Warmup #3

Given the following database schema:

person(name, age, account_id)
account(account_id, balance)

A person's *account_id* may or may not correspond to an *account_id* in the *account* relation.

Write relational algebra expressions to accomplish the following tasks:

1. Find the names of all people who have an existing account.

$$\Pi_{\text{name}}(\text{person} \bowtie \text{account})$$

2. Find the names of all people who have an account balance of greater than 500.

$$\Pi_{\text{name}}(\sigma_{\text{balance} > 500}(\text{person} \bowtie \text{account}))$$

3. Suppose every member of the bank will withdraw money via check equal to their age in two days. Find the names and balances for all customers in two days (assuming no other transactions).

$$\Pi_{\text{name, balance} + \text{age}}(\text{person} \bowtie \text{account})$$

Warmup #4

Given the following database schema:

student(student_name, concentration)
TAs(student_name, TA_name)
enrolled(student_name, college_name)

This schema represents students enrolled at various colleges, and reflects which students are TA'd by which TAs (who are themselves students).

Assume each student only has one TA, and that it is possible for a student to not be enrolled in a college.

Write relational algebra expressions to accomplish the following tasks:

1. Find the names of all students who concentrate in the same subject as their TA.

   $\Pi_{\text{student\_name}}(\sigma_{\text{concentration = concentrationTA}}(\text{student}$
   $\bowtie (\rho_{\text{studentTA(TA\_name, concentrationTA, student\_name)}}(\text{student}$
   $\bowtie_{\text{student\_name = TA\_name}} (\rho_{\text{renameTA(pupil\_name, TA\_name)}}(\text{TAs})))))$

2. Find the names of all students who are not enrolled at "Brown University".

   $\Pi_{\text{student\_name}}(\text{student})$
   $-\Pi_{\text{student\_name}}(\sigma_{\text{college\_name = "Brown University"}}(\text{enrolled}))$

Problem 5 (To Be Graded)

Bob Parr, aka, "Mr. Incredible", breaks large amounts of office equipment because of his super strength. His boss is starting to get annoyed and wants him to translate a series of questions to relational algebra so that they can order replacement equipment. The database they are using only speaks relational algebra. Bob skipped that day in his database course (because he was saving the world) thus he needs your help. Can you help Bob translate the relational algebra query to get him the information his boss needs?

Schema:

Product(Maker, model, type)
PC(Model, speed, ram, harddrive, screen, price)
Laptops(model, speed, ram, harddrive, screen, price)
Printer(model, color, type, price)

Questions

1. Find the make and model of all the pcs that are less that $1000 but greater than $800 dollars?

   $$\Pi_{\text{maker, model}}(\sigma_{\text{price } < 1000 \wedge price > 800}(\text{Product} \bowtie \text{PC}))$$

2. What are the models of pcs that are not made by a company that also makes laptops?

   $$\Pi_{model}(pc \bowtie product \bowtie (\Pi_{maker}(PC \bowtie Product) - \Pi_{maker}(Laptop \bowtie Product)))$$

3. Using no aggregate function, list the manufacture that makes the pc with the fastest processor?

   $$\Pi_{maker}((\Pi_{model}(PC) - \Pi_{PC.model}(\sigma_{PC.speed<PC2.speed}(PC \times \rho_{PC2}(PC)))) \bowtie Product)$$

4. Which maker makes at least 3 differently colored printers but does not make a PC with a 13 inch screen?

   $\Pi_{maker}(\sigma_{color\_count>2}({}_{maker}g_{count(color)as color\_count}(Printer \bowtie Product)))$
   $-\Pi_{maker}(\sigma_{screen==13}(PC \bowtie Product))$

5. What maker does not make a red printer or a laptop under $800?

   $\Pi_{maker}(Product) - \Pi_{maker}(\sigma_{color==red}(Printer \bowtie Product))$
   $-\Pi_{maker}(\sigma_{price<800}(Laptop \bowtie Product))$

6. Which manufacture makes the most unique laptop models?

   $$A \leftarrow_{maker\ g_{count(model)\ as\ model\_count}}(Product \bowtie Laptops)$$

   The result that is saved in A is all the counts of all the laptops that each maker makes. We then need to find the maker that has the most unique models, which is exactly the problem of 3 with a different input relation.