CS127 Homework #2

Due: September 27th, 2017 2:59 P.M.

Warmup #1

Suppose you have a relational database of Brown professors containing their Brown ID and their salary (integer). Write a tuple relational calculus formula that returns the ID(s) of the professor(s) with the highest salary.

Professors(id, salary)

$$\{p \mid p \in Professors \wedge \forall p_1 \in Professors(p[salary] \geq p_1[salary])\}$$

Warmup #2

Given the following relational database:

Sellers(sid, sname, address)
Products(pid, pname, color)
Catalog(sid, pid, cost)

1. Find the names of sellers who supply one or more green products

$$\{s \mid \exists s_1 \in Sellers(\exists c_1 \in Catalog(\exists P_1 \in Products(c_1[pid] = p_1[pid]$$
$$\wedge p_1[color] =' green') \wedge c_1[cid] = s_1[sid]) \wedge s[name] = s_1[name])\}$$

2. Find the IDs of sellers who supply every green product

$$\{s|\exists s_1 \in sellers(\exists C_i \in Catalog(\forall p \in Products(p[color] = green \wedge p[pid] = C_i[pid])$$
$$\wedge c_i[sid] = s_1[sid]) \wedge s_1[sid] = s[sid])\}$$

3. Find all pairs of sellers IDs such that the sellers with the first ID charges more for some part than the seller with the second ID.

$$\{t \mid \exists t_1 \in Catalog(\exists t_2 \in Catalog(t2.pid = t1.pid \wedge t2.sid \neq t1.sid \wedge$$
$$t2.cost < t1.cost \wedge t.sid = t2.sid2 \wedge t.sid1 = t1.sid)\}$$

Warmup #3

Using the relational database from problem 2, answer the following questions.

1. Explain why the below query is unsafe.

$$\{s \mid \neg(s \in Sellers)\}$$

   Infinite size output relation

2. State what the following queries compute:

   (a) $\{t \mid \exists t_1 \in Catalog(\exists t_2 \in Catalog(t_2.pid = t_1.pid \wedge t_2.sid = t_1.sid) \wedge t.pid = t_1.pid\}$
   This query gets the IDs of the products that are sold by multiple sellers.

   (b) $\{t \mid \exists t_1 \in Catalog(\exists x \in Sellers(x.sname = \text{“}Stein\text{”} \wedge x.sid = t_1.sid) \wedge \neg(\exists\, s\ Sellers(s.name = \text{‘}Stein' \wedge\ \exists z\ \in\ Catalog(z.sid = z.sid \wedge z.cost > t_1.cost))) \wedge t.pid = t_1.pid)\}$
   This query gets the IDs of the most expensive products sold by Stein.

   (c) $\{t \mid \exists t_1 \in Catalog(\exists x \in Products(x.color = \text{‘}green \wedge x.pid = t_1.pid) \wedge t.sid = t_1.sid) \vee \exists t_2 \in Suppliers(t_2.address = \text{‘}251\ Ives\ Street' \wedge t.sid = t_2.sid)\}$
   This query gets the IDs of the sellers that sell products that are green or are at 251 Ives Street

Problem 4 (To Be Graded)

For the following problem, you are going to come up with an entity relationship model diagram for an enterprise. Suppose you are hired by a cell phone company called Jog Wireless, and they want you to design their database architecture. They have given you a list of the requirements and type of functionality that they want an application to support. Your job is to design an ER diagram and relational schema, and then justify how you could support the functionality they need.

1. Given the following constraints and goals of the database, design an ER diagram for their database.
   - Requirements
     - Manages customer billing information, allowing automated bills to be sent to the home of the customers.
     - Tracks the current type of phones customers have along with each customer's phone number and billing plan type
     - Tracks text messages and phone calls and can determine each individual's phone and text usage per month.
   - Characteristics of Jog Wireless
     - Each customer has an address, a single credit card, a single active phone, and phone number
     - Supports pay-per-minute and pay-per-text, but also supports a fixed rate unlimited plan
     - They do not provide cellular data. They believe WiFi is the future.
   - Desired Application features
     - Calculate a monthly bill for a given customer based on usage
2. Explain your diagram and why you selected the entity and relationship sets you did in five or fewer sentences.
3. Given the ER diagram, write out a relational database schema for your database. In this schema indicate primary and foreign keys.
4. Explain how your database model would allow the application programmer to assemble the monthly bills for the customers.

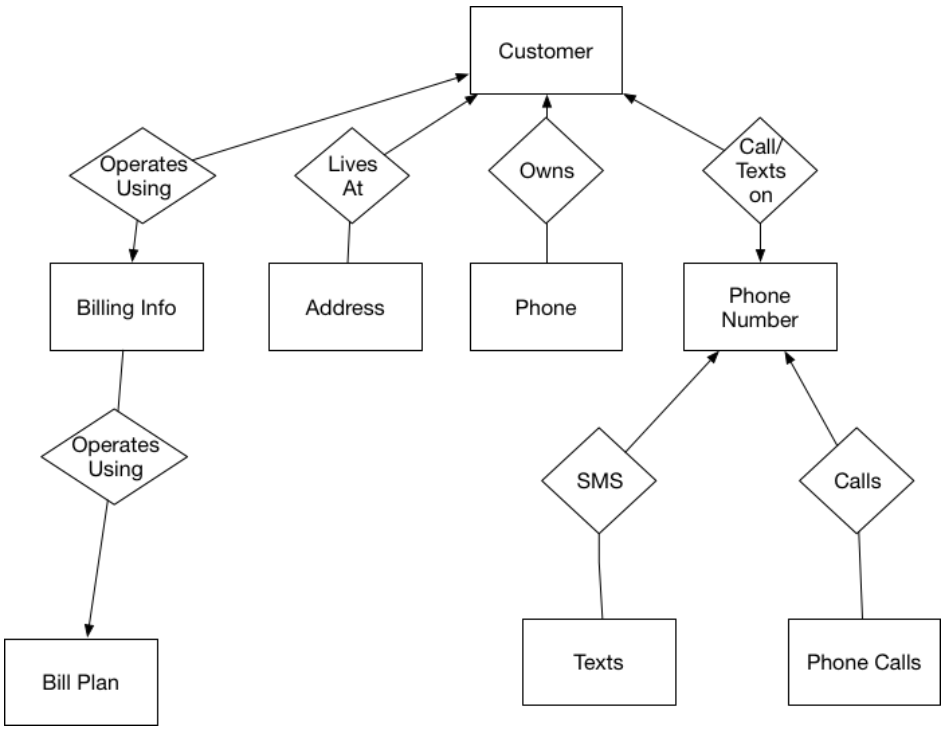
   In the following diagram, the customer table would store all the information about the customer like what is his name, id, email address and other personal information. The billing info, address, phone, and phone numbers table all contain the actual values and link them back to the customer. Then we have the text tables that is a giant table containing all the texts and phone calls for the database. Each mapping in this example is one to one.

```
Customer(ID, name, email_address, add_id)
(ID primary key)
(add_id foreign key address table)

BillingInfo(ID, creditCard, cvv, expiration_date, bill_add_id, billing_type)
(ID is primary key)
(bill_add_id foreign key address table)
(billing_type is a foreign key from billing type)

BillingType(bill_type_id, callRate, textRate, fixed_rate)
bill_type_id is a primary key

Address(add_id, line1, line2, city, state, zip)
(add_id is primary key)
```




```
Phone(cust_id, phone_id)
(cust_id is primary key)
(cust_id is a foreign key from customer)

PhoneNumber(ID, phoneNumber)
(phoneNumber is primary key)
(cust_id is foreign key from customer)

Texts(outgoing, ingoing, time, size)
(outgoing, ingoing, time is primary key)

PhoneCalls(outgoing, ingoing, start_time, end_time)
(outgoing, ingoing, start_time, is primary key)
```

In order to calculate a given monthly bill, we need to make sure that we have all the calls and texts that they have made or sent for the given month. We know that the phone calls are all kept in the phone calls table; thus, we would use that to get their usage. Similarly, we can do the the same with texts. Finally, we also want to use the bill type table to calculate the total.

Note: We recognize that there are several correct answers to this question. We will grade you on the consistency between your explanation and actual design. We will give more credit to those who have spent time considering the different database design, so in presenting material make sure to be thorough in the explanation but also keep the data model simple.

Problem 5 (To Be Graded)

After Bob, aka "Mr. Incredible", turned in your queries, he learned from his boss that the database system actually only speaks tuple relational calculus. Bob does not remember this language. Can you help him translate the following queries to help him keep his job?

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 maker and model of all PCs that are less than $1000 but greater than $800.

$$\{p \mid \exists p_1 \in Product(\exists pc \in PC(pc[price] > 800 \wedge pc[price] < 1000 \wedge pc[model] = p_1[model]) \wedge p[model] = p_1[model] \wedge p_1[make] = p[make])\}$$

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

$$\{pc\_min\_lap \mid \exists m \in product(\exists pcs \in pcs(m[model] = pcs[model]) \wedge \exists m_1 \in product(\forall lap \in laptops(lap[model] \neq m_1[model] \vee m[maker] \neq m_1[maker])) \wedge m[model] = pc\_min\_lap[model])\}$$

3. List the maker that makes the PC with the fastest processor.

$$\{pc \mid \exists pc_1 \in PC(\forall pc_2 \in PC(pc_1[speed] \geq pc_2[speed]) \wedge \exists m \in Product(m[maker] = pc_1[maker] \wedge pc[maker] = pc_1[maker]))\}$$

4. Which maker makes at least 2 differently colored printers but does not make a PC with a 13-inch screen?

$$\{m \mid \exists m_1 \in product(\exists p_1 \in Printers(\exists p_2 \in Printers(\exists m_2 \in product(p_2[color] \neq p_1[color] \wedge m_2[model] = p_2[model] \wedge m_1[model] = p_1[model] \wedge m_1[maker] = m_2[maker])))) \wedge \exists m_3 \in product(m_3[maker] = m_1[maker] \wedge \neg \exists lap \in laptops(m_3[model] = lap[model] \wedge lap[screen] = 13) \wedge m[maker] = m_1[maker]))\}$$

5. What maker makes neither a red printer nor a laptop under $800?

$$\{t \mid \exists m \in product(\forall p \in printer(p[color] \neq' red' \vee p[model] \neq m[model]) \wedge \neg \exists lap \in laptops(lap[price] < 800 \wedge lap[model] = maker[model]) \wedge m[maker] = t[maker])\}$$