CS2102: Database System

T05: Relational Algebra — AY2023/24 Sem 2: Week 07

Discussions

1. **(Algebra)** Consider the following schema:

Relation	Description
`Pizzas(pizza: TEXT)`	All the pizzas of interest.
`Customers(cname: TEXT, area: TEXT)`	The name and location of each customer.
`Restaurants(rname: TEXT, area: TEXT)`	The name and location of each restaurant.
`Likes(cname: TEXT, pizza: TEXT)`	Pizzas that customers like.
`Sells(rname: TEXT, pizza: TEXT, price: INTEGER)`	Pizzas sold by restaurants and the prices.

Additionally, we have the following foreign key constraints on the database schema:

- $(Likes.cname) \rightsquigarrow (Customers.cname)$
- $(Likes.pizza) \rightsquigarrow (Pizzas.pizza)$
- $(Sells.rname) \rightsquigarrow (Restaurants.rname)$
- $(Sells.pizza) \rightsquigarrow (Pizzas.pizza)$

Answer each of the following queries using relational algebra. You are encouraged to check your result on our relational algebra tools[1]. The `csv` files for the relations above are available on Canvas.

(a) Find all pizzas that Alice likes but is not liked by Bob[2].

(b) Find all customer-restaurant pairs (C, R) where C and R both located in the same area and C likes some pizza that is sold by R where the price of the pizza is less than \$15.

(c) Suppose the relation *Likes* contains all information about all customers. In other words, if the pair $(cname, pizza)$ is not in the relation *Likes*, it means that the customer *cname* *dislikes* the pizza *pizza*. Write a relational algebra expression to find for all customers, the pizza that they dislike. The result should be of the form $(cname, pizza)$.

[1] https://relational-algebra-evaluator.onrender.com/

[2] The intention is to state it as *"all pizzas that Alice likes but Bob does not like"*, however, this may indicate dislike which we have not discussed the underlying assumption yet.

2. **(RA Operator)** This question considers a binary relational algebra operator called the **division operator** denoted by $\div$[3].

Consider two relations $R(A_1, \cdots, A_m, B_1, \cdots, B_n)$ and $S(B_1, \cdots, B_n)$ where $m \geq 1$ and $n \geq 1$. That is, the set of attributes in S is a *proper* subset of the set of attributes in R.

Assume that the attributes that are in R but not in S are ordered as $(A_1, \cdots, A_m)$ in the schema of R and the schema of S is $(B_1, \cdots, B_n)$. Let L denote the list of attributes in the schema of R.

The division of R by S (denoted by $R \div S$) computes the largest set of tuples $Q \subseteq \pi_{[A1, \cdots, A_m]}(R)$ such that for every tuple $(a_1, \cdots, a_m) \in Q$,

$$\pi_{[L]}(\{(a_1, \cdots, a_m)\} \times S) \subseteq R$$

Q is also referred to as the **quotient** of $R \div S$ and its schema is $(A_1, \cdots, A_m)$. The following example illustrates $R \div S$ given two relations $R(A, B)$ and $S(B)$.

R

A	**B**
a	1
a	2
b	1
c	1
c	2
c	3
d	2
d	3

S

B
1
2

R÷S

A
a
c

(a) Consider the schema above. We now pose the following question.

Find the restaurants that sell all the pizzas that Alice likes and don't sell any pizza that Bob likes.

Write a relational algebra expression for this query that uses the division and natural join operators.

(b) Given relations $R(A, B)$ and $S(B)$, write a relational algebra expression to compute the division of R by S using only the basic relational operators (*i.e.*, σ, π, ρ, $\times$, $\cup$, $\cap$ and $-$).

[3] It is a "division" operation because it kind of reverses the "multiplication" operation $\times$.

3. **(Equivalent)** Recap that two queries Q_1 and Q_2 on a relational database with schema R are defined to be **equivalent** (denoted by $Q_1 \equiv Q_2$) if for *every* valid instance r of R

   - both Q_1 and Q_2 always compute the same results on r, *and*
   - the order of the columns are the same

   Consider a database with the following relational schema: $R(\underline{A}, C)$, $S(\underline{A}, D)$, and $T(\underline{X}, Y)$, with primary key attributes underlined. Assume all the attributes have integer domain. For each of the following pairs of queries Q_1 and Q_2, state whether or not $Q_1 \equiv Q_2$.

	$\mathbf{Q_1}$	$\mathbf{Q_2}$
(i)	$\pi_{[A]}(\sigma_{[A<10]}(R))$	$\sigma_{[A<10]}(\pi_{[A]}(R))$
(ii)	$\pi_{[A]}(\sigma_{[C<10]}(R))$	$\sigma_{[C<10]}(\pi_{[A]}(R))$
(iii)	$\pi_{[D,Y]}(S \times T)$	$\pi_{[D]}(S) \times \pi_{[Y]}(T)$
(iv)	$\pi_{[D,Y]}(S \times T)$	$\pi_{[D,Y]}(T \times S)$

Challenge

The answers to the following questions is given without explanation. Please discuss them on Canvas.

1. **(Equivalent)** Recap that two queries Q_1 and Q_2 on a relational database with schema R are defined to be **equivalent** (denoted by $Q_1 \equiv Q_2$) if for *every* valid instance r of R

   - both Q_1 and Q_2 always compute the same results on r, *and*
   - the order of the columns are the same

   Consider a database with the following relational schema: $R(\underline{A}, C)$, $S(\underline{A}, D)$, and $T(\underline{X}, Y)$, with primary key attributes underlined. Assume all the attributes have integer domain. For each of the following pairs of queries Q_1 and Q_2, state whether or not $Q_1 \equiv Q_2$.

	$\mathbf{Q_1}$	$\mathbf{Q_2}$
(i)	$(R \times \pi_{[D]}(S)) \times T$	$R \times (\pi_{[D]}(S) \times T)$
(ii)	$\pi_{[A]}(R \cup S)$	$\pi_{[A]}(R) \cup \pi_{[A]}(S)$
(iii)	$\pi_{[A]}(R - S)$	$\pi_{[A]}(R) - \pi_{[A]}(S)$

   What about *isomorphism*, can you check if $Q_1 \cong Q_2$.

2. **(Algebra)** Consider the following schema:

Relation	**Description**
`Pizzas(`<u>`pizza`</u>`: `*`TEXT`*`)`	All the pizzas of interest.
`Customers(`<u>`cname`</u>`: `*`TEXT`*`, area: `*`TEXT`*`)`	The name and location of each customer.
`Restaurants(`<u>`rname`</u>`: `*`TEXT`*`, area: `*`TEXT`*`)`	The name and location of each restaurant.
`Likes(`<u>`cname`</u>`: `*`TEXT`*`, `<u>`pizza`</u>`: `*`TEXT`*`)`	Pizzas that customers like.
`Sells(`<u>`rname`</u>`: `*`TEXT`*`, `<u>`pizza`</u>`: `*`TEXT`*`, price: `*`INTEGER`*`)`	Pizzas sold by restaurants and the prices.

   Additionally, we have the following foreign key constraints on the database schema:

   - $(Likes.cname) \rightsquigarrow (Customers.cname)$
   - $(Likes.pizza) \rightsquigarrow (Pizzas.pizza)$
   - $(Sells.rname) \rightsquigarrow (Restaurants.rname)$
   - $(Sells.pizza) \rightsquigarrow (Pizzas.pizza)$

   Answer each of the following queries using relational algebra.

   (a) For each restaurant, find the price of the most expensive pizzas sold by that restaurant. Exclude restaurants that do not sell any pizza.

   (b) Find all customer-pizza pairs (C, P) where the pizza P sold by some restaurant that is located in the same area as that of the customer C. Include customers whose associated set of pizzas is empty.