CS2102 Database Systems

Semester 1 2019/2020

Tutorial 01 (*Selected Answers*)

Quiz

For the quiz, we will use the following relation instances below.

r

A	B	C	D
1	0	0	1
2	3	2	3
1	1	2	3
0	3	1	2

s

E	F	G	H
2	1	1	2
1	0	0	1
0	0	0	0
3	3	3	3

1. What is the resulting relation instance from the expression $Q = \pi_{A,B}(\boldsymbol{r})$?
2. What is the resulting relation instance from the expression $Q = \pi_{C,D}(\boldsymbol{r})$?
3. What is the resulting relation instance from the expression $Q = \sigma_{A=1}(\boldsymbol{r})$?
4. What is the resulting relation instance from the expression $Q = \sigma_{A\neq 1}(\boldsymbol{s})$?
5. What is the resulting relation instance from the expression $Q = \sigma_{A=E}(\boldsymbol{r} \times \boldsymbol{s})$?
6. What is the resulting relation instance from the expression $Q = \pi_E\left(\rho_{(E,F,G,H)}(r)\right)$?
7. What is the resulting relation instance from the expression $Q = \boldsymbol{r} \cup \rho_{(A,B,C,D)}(\boldsymbol{s})$?
8. What is the resulting relation instance from the expression $Q = \rho_{(E,F,G,H)}(\boldsymbol{r}) \cap \boldsymbol{s}$?
9. What is the resulting relation instance from the expression $Q = \boldsymbol{r} - \rho_{(A,B,C,D)}(\boldsymbol{s})$?
10. Which of the following property is correct about binary operators?
    a) Union $\cup$ is associative
    b) Union $\cup$ is commutative
    c) Intersection $\cap$ is associative
    d) Intersection $\cap$ is commutative
    e) Set difference $-$ is associative
    f) Set difference $-$ is commutative
11. Which of the following property is correct about unary operators?
    a) Projection π may remove column
    b) Projection π may add column
    c) Projection π may remove rows
    d) Projection π may add rows
    e) Projection π may reorder columns
    f) Projection π may rename columns
    g) Selection σ may remove column
    h) Selection σ may add column
    i) Selection σ may remove rows
    j) Selection σ may add rows
    k) Selection σ may reorder columns
    l) Selection σ may rename columns
    m) Renaming ρ may remove column
    n) Renaming ρ may add column
    o) Renaming ρ may remove rows
    p) Renaming ρ may add rows
    q) Renaming ρ may reorder columns
    r) Renaming ρ may rename columns

Tutorial Questions *[Discussion: 12, 13, 14(a), 14(b), 14(c)]*

12. Consider the following relation instance *r* of the relational schema *R(A,B,C,D)*.

r

A	B	C	D
3	0	0	1
2	1	2	0
1	1	2	0
0	0	1	2

a) Based on *r*, write down all the possible superkeys of *R*.
b) In addition to *r*, suppose that it is also known that *{A,C}* is a superkey of *R*. Based on the given information, write down all the possible candidate keys of *R*.

Solution:
In this question (and also Q11), we are mainly *inferring* properties of the relation schema from an instance of the schema. As such, the superkeys/candidate keys/foreign keys identified here are generally only *possibilities* unless we have further information to derive more definite answers.

a) Possible superkeys of *R* are *{A}*, *{A,B}*, *{A,C}*, *{A,D}*, *{A,B,C}*, *{A,B,D}*, *{A,C,D}*, and *{A,B,C,D}*.
b) Even if *{A,C}* is indeed superkey of *R*, *{A,C}* may not be a candidate key of *R* since *{A}* is a possible superkey of *R* based on *r*. Since we only confirm *{A,C}* as superkey, any superkey which is a superset of *{A,C}* cannot be a candidate key of *R*. Out of the remaining --*{A}*, *{A,B}*, *{A,D}*, and *{A,B,D}*-- all are possible but if *{A}* is a candidate key then the rest cannot. If *{A,B}* and/or *{A,D}* is a candidate key then *{A,B,D}* cannot.

13. Consider a relational database consisting of two relations with schema *R(<u>A</u>,B)* and *S(<u>W</u>,X,Y,Z)*, where the primary keys of *R* and *S* are *{A}* and *{W}* respectively. Let *r* and *s* be the current instance of *R* and *S* respectively as shown below.

r

<u>A</u>	B
3	0
2	1
1	1
0	0

s

<u>W</u>	X	Y	Z
0	4	0	null
1	null	2	null
2	1	2	null
3	0	1	null

Based on the current database instance, write down all the possible foreign keys in *S* that refer to attribute *A* in *R*.

Solution:
The *possible* foreign keys are *W*, *Y*, and *Z*.

14. Two queries Q_1 and Q_2 on a relational database with schema *D* are defined to be **equivalent queries** (denoted $Q_1 \equiv Q_2$) if for every legal instance *d* of *D*, both Q_1 and Q_2 compute the same results on *d*.
Consider a database with the following relational schema: *R(<u>A</u>,C)*, *S(<u>A</u>,D)*, and *T(<u>X</u>,Y)*, with the primary key attributes underlined. Assume that 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 and Q_2 are equivalent queries.

a) $Q_1 = \pi_A(\sigma_{A<10}(R))$ and $Q_2 = \sigma_{A<10}(\pi_A(R))$
b) $Q_1 = \pi_A(\sigma_{C<10}(R))$ and $Q_2 = \sigma_{C<10}(\pi_A(R))$
c) $Q_1 = \pi_{D,Y}(S \times T)$ and $Q_2 = \pi_D(S) \times \pi_Y(T)$

d) $Q_1 = \pi_{D,Y}(S \times T)$ and $Q_2 = \pi_{D,Y}(T \times S)$
e) $Q_1 = \left(R \times \pi_D(S)\right) \times T$ and $Q_2 = R \times (\pi_D(S) \times T)$
f) $Q_1 = \pi_A(R \cup S)$ and $Q_2 = \pi_A(R) \cup \pi_A(S)$
g) $Q_1 = \pi_A(R - S)$ and $Q_2 = \pi_A(R) - \pi_A(S)$

Solution:
a) Equivalent
b) Not equivalent Q_2 is an *invalid* relational algebra expression as the selection condition refers to a *non-existent* attribute
c) Equivalent
d) Equivalent
e) Equivalent
f) Equivalent
g) Not equivalent Consider the following database instance *d*: *r* = { (10, 10) } and *s* = { (10, 20) }. The result of Q_1 on *d* is { (10) } while the result of Q_2 on *d* is ∅.

15. Consider the following relational database schema discussed in class, where the primary key of each relation is underlined.

    Pizzas (<u>pizza</u>)
    Customers (<u>cname</u>, area)
    Restaurants (<u>rname</u>, area)
    Contains (<u>pizza, ingredient</u>)
    Sells (<u>rname, pizza</u>, price)
    Likes (<u>cname, pizza</u>)

    `Pizzas` indicates all the pizzas of interest. **`Customers`** indicates the name and location of each customer. **`Restaurants`** indicates the name and location of each restaurant. **`Contains`** indicates the ingredients used in each pizza. **`Sells`** indicates the pizzas sold by restaurants and their prices. **`Likes`** indicates the pizzas that customers like.

    The following are all the foreign key constraints on the database schema:
    - `Contains.pizza` is a foreign key that refers to `Pizzas.pizza`
    - `Sells.rname` is a foreign key that refers to `Restaurants.name`
    - `Sells.pizza` is a foreign key that refers to `Pizzas.pizza`
    - `Likes.cname` is a foreign key that refers to `Customers.pizza`
    - `Likes.pizza` is a foreign key that refers to `Pizzas.pizza`

    Answer each of the following queries using relational algebra.
    a) Find pizzas that Alice likes but Bob does not like.
    b) Find all customer-restaurant pairs (C,R) where C and R are both located in the same area, and C likes some pizza that is sold by R.
    c) Suppose that the database contains an additional relation `Dislikes(`<u>`cname,pizza`</u>`)` which indicates the pizzas that customers do not like. The database also satisfies the following constraint: for every customer $c \in \pi_{\text{cname}}(\text{Customers})$ and for every pizza $p \in \pi_{\text{pizza}}(\text{Contains})$, either $(c,p) \in \text{Likes}$ or $(c,p) \in \text{Dislikes}$ (*in other words, you know the likes and dislikes of every customers with respect to all pizzas, and they cannot both like and dislike a pizza*). Given this database, find all customer pairs (C_1, C_2) such that C_1 likes some pizza that C_2 does not like.
    d) Consider the original database schema without the `Dislikes` relation. Write a query to compute the `Dislikes` relation.
    e) Find all customer pairs (C_1, C_2) such that $C_1 < C_2$ and they like exactly the same pizzas.
    f) For each restaurant, find the price of the most expensive pizzas sold by that restaurant. Excludes restaurants that do not sell any pizza.

Solution:

a) $Q_{ans} = \pi_{\text{pizza}}(\sigma_{\text{cname}='Alice'}(\text{Likes})) - \pi_{\text{pizza}}(\sigma_{\text{cname}='Bob'}(\text{Likes}))$

b) $Q_1 = \text{Customers} \times \rho_{R_2(\text{rname}_2,\text{area}_2)}(\text{Restaurants})$

$Q_2 = Q_1 \times \rho_{S_3(\text{rname}_3,\text{pizza}_3,\text{price}_3)}(\text{Sells})$

$Q_3 = Q_2 \times \rho_{L_4(\text{cname}_4,\text{pizza}_4)}(\text{Likes})$

$Q_4 = \sigma_{(\text{area}=\text{area}_2)\wedge(\text{rname}_2=\text{rname}_3)\wedge(\text{cname}=\text{cname}_4)\wedge(\text{pizza}_3=\text{pizza}_4)}(Q_3)$

$Q_{ans} = \pi_{\text{cname},\text{rname}}(Q_4)$

c) $Q_{ans} = \pi_{\text{cname},\text{cname}_2}\left(\sigma_{\text{pizza}=\text{pizza}_2}\left(\text{Likes} \times \rho_{\text{Dislikes}_2(\text{cname}_2,\text{pizza}_2)}(\text{Dislikes})\right)\right)$

d) $Q_{ans} = \left((\pi_{\text{cname}}(\text{Customers})) \times \text{Pizzas}\right) - \text{Likes}$

e) Let `LikeDislike(cname,cname2)` denotes the output computed by part (c).

$Q_1 = \pi_{\text{cname},\text{cname}_2}\left(\sigma_{\text{cname}<\text{cname}_2}\left(\text{Likes} \times \rho_{\text{Likes}_2(\text{cname}_2,\text{pizza}_2)}(\text{Likes})\right)\right)$

$Q_2 = \text{LikeDislike} \cup \left(\pi_{\text{cname}_2,\text{cname}}(\text{LikeDislike})\right)$

$Q_{ans} = Q_1 - Q_2$

Note that the solution above will *exclude* pairs of customers who don't like any pizza. To include these pairs as well, the `Likes` table should be replaced by `Customers` table as follows.

$Q_1 = \pi_{\text{cname},\text{cname}_2}\left(\sigma_{\text{cname}<\text{cname}_2}\left(\text{Customers} \times \rho_{\text{Cust}_2(\text{cname}_2,\text{area}_2)}(\text{Customers})\right)\right)$

$Q_2 = \text{LikeDislike} \cup \left(\pi_{\text{cname}_2,\text{cname}}(\text{LikeDislike})\right)$

$Q_{ans} = Q_1 - Q_2$

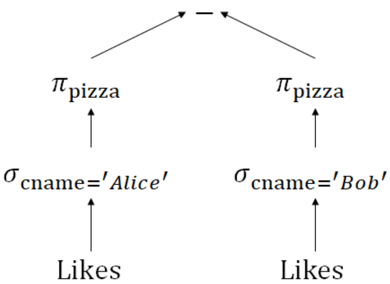
f) $Q_1 = \pi_{\text{rname},\text{price}}(\text{Sells})$

$Q_2 = \text{Sells} \times \rho_{\text{Sells}_2(\text{rname}_2,\text{pizza}_2,\text{price}_2)}(\text{Sells})$

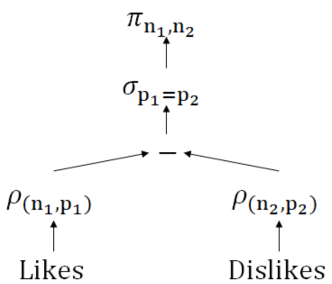
$Q_3 = \pi_{\text{rname},\text{price}}\left(\sigma_{(\text{rname}=\text{rname}_2)\wedge(\text{price}<\text{price}_2)}(Q_2)\right)$

$Q_{ans} = Q_1 - Q_3$

Some diagrams:



a)



c)

d)