

CSC343 A1

Haider Sajjad, Muskan Patpatia

June 5, 2019

Relations

- Restaurant(name, owner, capacity, country)
- Patron(PID, fname, lname, birthday)
- Dish(DID, name, dietary)
- Reservation(RID, PID, rname, date)
- Order(RID, DID, number)
- Rating(PID, rname, rating)

Part 1

1. Question: Report the name of the Patron that has given the highest rating to a restaurant. If there are ties, report all of them.

Let $r1$ and $r2$ be two Rating relations

$$r1 := \rho \text{ r1 Rating} \quad (1)$$

$$r2 := \rho \text{ r2 Rating} \quad (2)$$

Let $nottop$ be the relation of all PID's that are not the maximum

$$nottop(PID) := \Pi_{(r1.PID)} (\sigma_{r2.rating > r1.rating \wedge r2.PID \neq r1.PID} (r1 \times r2)) \quad (3)$$

Let $temp$ be the relation of all maximums, pid's that are in Rating but not in $nottop$. Natural Join $temp$ with $patron$ to get the names of the Patrons with pids in $temp$.

$$temp(PID) := (\Pi_{PID} Rating) - nottop \quad (4)$$

$$\Pi_{fname, lname} (temp \bowtie Patron) \quad (5)$$

2. Question: Report the name of the restaurant for which the highest number of reservations were made. If there are ties report all of them.

Can't be done with relational algebra operations.

3. Question: Report the PID(s) of the Patrons(s) who reserved a spot at a restaurant, but did not order anything.

Make an *pidsOrdered* relation that holds the pid of every patron who has made an order by natural joining *Order* with *Reservation*. Then subtract *pidsOrdered* from the set of all *Reservation* pids, this will give all the patron pids who have a reservation but have not ordered anything.

$$pidsOrdered(RID) := \Pi_{RID} (Order \bowtie Reservation) \quad (6)$$

$$(\Pi_{PID} Reservation) - pidsOrdered \quad (7)$$

4. Question: Report the name(s) of the Patrons(s) who have made a reservation to the restaurant named 'Boston Pizza' and ordered 3 of a dish called 'Margherita Pizza'.

Make a relation called *pBP* that holds the pid and rid of every patron who has made a reservation at 'Boston Pizza'. And a relation called *mp* that holds the did of a dish called 'Margherita Pizza'.

$$pBP(PID, RID) := \Pi_{PID, RID} (\sigma_{rname='Boston Pizza'}(Reservation)) \quad (8)$$

$$mp(DID) := \Pi_{DID} (\sigma_{name='Margherita Pizza'}(Dish)) \quad (9)$$

Let *temp* be the relation holding every order from Boston pizza that has 3 of any dish. Then, natural join *temp* and *mp* to get all the orders where 'Margherita Pizza' was ordered 3 times. And natural join again with *Patron* to get the names of the patrons.

$$temp := \sigma_{number=3}(pBP \bowtie Order) \quad (10)$$

$$\Pi_{fname, lname}[(\Pi_{PID}(mp \bowtie temp)) \bowtie Patron] \quad (11)$$

5. Question: Report the owner of the restaurant with the highest average rating. If there are ties, report all of them.

We can't find average in relational algebra.

Let *r1* and *r2* be two instances of *Rating* relation.

$$r1 := \rho \text{ r1 Rating} \quad (12)$$

$$r2 := \rho \text{ r2 Rating} \quad (13)$$

...

6. Question: Report the capacities of the restaurants from which patrons have so far only ordered foods with a 'gluten-free' dietary restriction.

Make a relation called *glutenfree* that holds the did, of every dish that is gluten free. And make a relation called *gfOrders* that holds the rid of every patron who has ordered a gluten free patron.

$$glutenfree(DID) := \Pi_{DID}(\sigma_{directory='gluten-free'}(Dish)) \quad (14)$$

$$temp(RID) := \Pi_{RID}(glutenfree \bowtie Order) \quad (15)$$

Make another relation called *restnames* that holds the restaurant name of every restaurant that serves 'gluten-free food'. Do this by natural joining *temp* with *Reservation*. Finally cross *restnames* and *Restaurant* to get the capacity of every resteraunt serving 'gluten-free'.

$$restnames(rname) := \Pi_{rname}(temp \bowtie Reservation) \quad (16)$$

$$\Pi_{capacity}(\sigma_{restnames.rname=Restaurant.name}(restnames \times Restaurant)) \quad (17)$$

7. Question: Report the restaurant owner for which the very earliest reservation out of all the reservations in the database was made. Report any ties.