University of Toronto
CSC343, Summer 2019

Assignment 1:
Mengning Yang (1002437552)
Keyu Wang (1003540625)

Unary operators on relations:

- $\Pi_{x,y,z}(R)$
- $\sigma_{condition}(R)$
- $\rho_{New}(R)$
- $\rho_{New(a,b,c)}(R)$

Binary operators on relations:

- $R \times S$
- $R \bowtie S$
- $R \bowtie_{condition} S$
- $R \cup S$
- $R \cap S$
- $R - S$

Logical operators:

- $\vee$
- $\wedge$
- $\neg$

Assignment:

- $New(a, b, c) := R$

Stacked subscripts:

- $\sigma_{this.something > that.something \wedge \\ this.otherthing \leq that.otherthing}$

Below is the text of the assignment questions; we suggest you include it in your solution. We have also included a nonsense example of how a query might look in LaTeX. We used `\var` in a couple of places to show what that looks like. If you leave it out, most of the time the algebra looks okay, but certain words, *e.g.*, "Offer" look horrific without it.

The characters "\\" create a line break and "[5pt]" puts in five points of extra vertical space. The algebra is easier to read with extra vertical space. We chose "–" to indicate comments, and added less vertical space between comments and the algebra they pertain to than between steps in the algebra. This helps the comments visually stick to the algebra.

Part 1: Queries

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

   – All patrons who have given ratings to a restaurant.

   $GivenRatings(fname, lname, rating) := \Pi_{fname,lname,rating}(Rating \bowtie Patron)$

   – Name of patrons who's rating is not the highest.

   $NotTop(fname, lname, rating) := \Pi_{P1.fname,P1.lname,P1.rating}\sigma_{P1.rating<P2.rating}$
   $(\rho_{P1}GivenRatings) \times (\rho_{P2}GivenRatings)$

   – Name of patrons who has given the highest rating to a restaurant.

   $Answer(fname, lname) := \Pi_{fname,lname}(GivenRatings - NotTop)$

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

   Cannot be expressed.

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

   – All patrons who have made a reservation at a restaurant.

   $Reserved(PID, RID, rname) := \Pi_{PID,RID}(Patron \bowtie Reservation)$

   – Patrons(s) who reserved a spot at a restaurant and ordered something.

   $Ordered(PID, RID, rname) := \Pi_{PID,RID}\sigma_{Reserved.rname=Order.rname}(Order \bowtie Reserved)$

   – PID of Patrons(s) who reserved a spot at a restaurant but did not order anything.

   $Answer(PID) := \Pi_{PID}(Reserved - Ordered)$

4. 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'.

– Name of patrons who have made a reservation to Boston Pizza

$Reserved(fname, lname, RID) := \Pi_{fname,lname,RID}\sigma_{Reservation.rname='BostonPizza'}(Patron \bowtie Reservation)$

– RID of orders with 3 "Margherita Pizza"

$3Pizza(RID) := \Pi_{RID}\sigma_{(Order.number=3)\wedge(Dish.name='MargheritaPizza')}(Order \bowtie Dish)$

– 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'

$Consumer(fname, lname) := \Pi_{fname,lname}(Reserved \bowtie 3Pizza)$

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

   Cannot be expressed.

6. Report the capacities of the restaurants from which patrons have so far only ordered foods with a 'gluten-free' dietary restriction.
   – All Restaurants that have reservations.
   $Reserved(rname, RID, capacity) := \Pi_{rname,RID,capacity}\sigma_{Restaurant.name=Reservation.rname}(Restaurant \times Reservation)$

   – RID of Orders with food other than 'gluten-free'

   $WithGluten(RID) := \Pi_{RID}\sigma_{Dish.dietary\neq'gluten-free'}(Order \bowtie Dish)$

   – Restaurants that have orders other than 'gluten-free'

   $NotOnly(rname, RID, capacity) := \Pi_{rname,RID,capacity}(Reserved \bowtie WithGluten)$
   – Capacity of restaurants that has only gluten-free orders so far.

   $Answer(capacity) := \Pi_{capacity}(Reserved - NotOnly)$

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

   – All restaurants that have reservations.

   $Reserved(name, owner, date) := \Pi_{name,owner,date}\sigma_{Restaurant.name=Reservation.rname}(Restaurant \times Reservation)$

   – Find all the reservations that are not the earliest.

   $NotEarliest(name, owner, date) := \Pi_{R1.name,R1.owner,R1.date}\sigma_{R1.date<R2.date}[(\rho_{R1}Reserved) \times (\rho_{R2}Reserved)]$

   – Owner of restaurants that has the earliest reservation.

   $Answer(owner) := \Pi_{owner}(Reserved - NotEarliest)$

8. Report the PID(s) of the Patrons who have made reservations to the restaurant named 'Red Lobster' on their birthday.
   – PIDs of Patrons who went to restaurants to celebrate their birthdays.
   $Celebration(PID, rname) := \Pi_{PID,rname}\sigma_{Patron.birthday=Reservation.date}(Reservation \bowtie Patron)$

– Answer
$Answer(PID) := \Pi_{PID}\sigma_{rname="RedLobster"}(Celebration)$

9. Consider all patrons that have made reservations to at least two different restaurants. For each of those patrons, report their name, and the names and ratings of all of the restaurants they went to (not ones they rated without actually going to).
– Patrons who have made reservations
$ReservedPatron(PID, fname, lname, RID, rname) := \Pi_{PID,fname,lname,RID,rname}(Patron \bowtie Reservation)$

– Patrons that have made reservations to at least two different restaurants
$QualifiedPatron(PID, fname, lname, RID) := \Pi_{PID,fname,lname,RID}$
$\sigma_{(R1.fname=R2.fname)\wedge(R1.lname=R2.lname)\wedge(R1.rname\neq R2.rname)}[(\rho_{R1}ReservedPatron)\times(\rho_{R2}ReservedPatron)]$

– Restaurants that have reservations
$ReservedRestaurant(PID, RID, rname) := \Pi_{PID,RID,rname}\sigma_{Restaurant.name=Reservation.rname}(Restaurant\times Reservation)$

– RID of reservation of a restaurant that has orders
$QualifiedRestaurant(PID, RID, rname) := \Pi_{PID,RID,rname}(ReservedRestaurant \bowtie Order)$

– All restaurants that qualified patrons went to (They made a reservation and ordered something)
$RestaurantWentTo(PID, fname, lname, rname) := \Pi_{PID,fname,lname,rname}(QualifiedRestaurant \bowtie QualifiedPatron)$

– Ratings added of each of RestaurantWentTo
$AddedRating(fname, lname, rname, rating) := \Pi_{fname,lname,rname,rating}(RestaurantWentTo \bowtie Rating)$

– Report those patron's name, the names and ratings of all of the restaurants they went to
$Answer(fname, lname, rname, rating) := \Pi_{fname,lname,rname,rating}(AddedRating)$

10. Report the name of all Restaurants that had reservations made on every day that someone made a reservation at the restaurant named 'Pickle Barrel'.
– Dates of someone made a reservation at the restaurant named 'Pickle Barrel'
$PickleDates(rname, date) := \Pi_{rname,date}\sigma_{rname="PicKleBarrel"}(Reservation)$
– All reservation dates
$AllDates(rname, date) := \Pi_{rname,date}(Reservation)$

$ShouldHaveBeen(rname, date) := \Pi_{rname,date}PickleDates \times AllDates$
$WereNotAlways(rname, date) := ShouldHaveBeen - PickleDates$ –Answer
$Answer(rname) := (\Pi_{rname}AllDates) - (\Pi_{rname}WereNotAlways)$

Part 2: Integrity Constraints

Express the following integrity constraints with the notation $R = \emptyset$, where R is an expression of relational algebra. You are welcome to define intermediate results with assignment and then use them in an integrity constraint.

1. A restaurant owner can only own one restaurant.

Answer:
$\sigma_{(R1.name \neq R2.name) \wedge (R1.onwer = R2.owner)}[(\rho_{R1} Restaurant) \times (\rho_{R2} Restaurant)] = \emptyset$

2. Patrons who did not make a reservation for a restaurant cannot review it.
Answer:
– patrons who made a reservation for a restaurant
$Reserved(PID, rname) := \Pi_{PID,rname} \sigma_{Patron.PID = Reservation.PID}(Patron \times Reservation)$

– all patrons who did not make a reservation for a restaurant.
$NotReserved(PID, rname) := (\Pi_{PID,rname} Reservation) - Reserved$

– all patrons who did not make a reservation but wrote a review.
$\sigma_{(NotReserved.PID = Rating.PID) \wedge (Rating.rname = NotReserved.rname)}[(NotReserved \times Rating)] = \emptyset$

3. A Patron cannot make multiple reservations in one day for a restaurant that has a capacity less than 100.
Answer:
– all restaurants that has a capacity less than 100
$validRestaurant(name) := \Pi_{name} \sigma_{capacuty < 100}(Restaurant)$

– all reservations of restaurants that has a capacity less than 100.
$Reserved(PID, date) := \Pi_{PID,date} \sigma_{Reservation.rname = validRestaurant.name}(Reservation \times validRestaurant)$

– PID of patrons who made more than one reservations in one day.
$\Pi_{R1.PID} \sigma_{(R1.PID = R2.PID) \wedge (R1.date = R2.date)}[(\rho_{R1} Reserved) \times (\rho_{R2} Reserved)] = \emptyset$