University of Toronto
CSC343, Winter 2016

Assignment 1: Name: Akhil Gupta SN: 1000357071

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:

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

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. Find all concerts in Toronto in 2016 that have one or more unsold seats costing under $25. Report the event ID.

   – All Concerts in Toronto in 2016.

   $AllConcerts(eID) := \Pi_{eID}\sigma_{type='concert' \wedge city='Toronto' \wedge when.year='2016'}(Event \bowtie Venue)$

   – All tickets available for all concerts under $25.

   $AllTickets(eID, sID) := \Pi_{eID,sID}\sigma_{Ticket.eID=AllConcerts.eID \wedge Ticket.price<25}(Ticket \bowtie AllConcerts)$

   – All purchased tickets for all concerts under $25.

   $AllPurchased(eID) := \Pi_{eID}\sigma_{AllConcerts.eID=Purchase.eID \wedge Purchase.sID=AllTickets.sID \wedge Ticket.price<25}$
   $(AllTickets \bowtie Purchase)$

   $Answer(eID) := \Pi_{eID}(AllTickets - AllPurchased)$

2. Find all users who have paid at least $200 for some ticket, but have never bought a ticket to a musical. You might call these people "big spenders" who hate musicals. For each of them, find all the tickets they've bought for over $200. Report the username, ticket price, event ID, event date and time, and event name.

   – Big spenders who hate musicals.

   $BigSpenders(userName) :=$
   $\Pi_{userName}\sigma_{Ticket.price \geq 200 \wedge Event.eID=Ticket.eID=Purchase.eID \wedge Event.type \neq 'musical'}$
   $(Event \bowtie Ticket \bowtie Purchase)$

   – Big spenders who bought tickets over $200.

   $BigSpendersOver200(userName, Ticket.eID) :=$
   $\Pi_{userName,Ticket.eID}\sigma_{Purchase.username=BigSpenders.username \wedge Purchase.eID=Ticket.eID \wedge Ticket.price>200}$
   $(Ticket \bowtie BigSpenders \bowtie Purchase)$

   – All details about big spenders.

   $Answer(userName, Ticket.price, Event.eID, Event.when, Event.name) :=$
   $\Pi_{userName,Ticket.price,Event.eID,Event.when,Event.name}\sigma_{BigSpendersOver200.Ticket.eID=Event.eID}$
   $(BigSpendersOver200 \bowtie Event)$

3. Find all users who, in two consecutive years, have bought multiple tickets for a single event. Report their user names and email addresses.

   – People who bought tickets before or on the same year of Event.

   $BoughtBeforeOrSameYear(eID, when) :=$
   $\Pi_{eID,when}\sigma_{Purchase.when.year \leq Event.when.year \wedge Purchase.eID=Event.eID}$
   $(Event \bowtie Purchase)$

– At least two tickets bought.

$$AtleastTwo(userName) := \Pi_{userName}\sigma_{B1.eID=B2.eID \wedge B2.when.year=B1.when.year+1 \wedge B1.sID \neq B2.sID} (\rho_{BoughtBeforeOrSameYear}B1 \times \rho_{BoughtBeforeOrSameYear}B2)$$

$$Answer(userName, email) := \Pi_{userName,email}\sigma_{AtLeastTwo.username=User.username} (AtLeastTwo \bowtie User)$$

4. Find all events in 2015 or earlier for which none of the seats at the top price were sold, but every seat at a lower price was sold. Report the event ID and event name.
   – All Events in 2015 or ealier.

$$AllEvents(eID) := \Pi_{eID}\sigma_{when.year \leq 2015}(Event)$$

   – All seats at lower price.

$$LowerPrice(eID, sID) := \Pi_{eID,sID}\sigma_{AllEvents.eID=T1.eID=T2.eID \wedge T1.price<T2.price} (AllEvents \bowtie (\rho_{Ticket}T1 \times \rho_{Ticket}T2))$$

   – All seats at top price.

$$TopPrice(eID, sID) := \Pi_{eID,sID}Ticket - LowerPrice$$

   – All purchased seats at lower price.

$$LowerPricePurchased(eID) := \Pi_{eID}\sigma_{LowerPrice.eID=Purchase.eID \wedge LowerPrice.sID=Purchase.sID} (LowerPrice \bowtie Purchase)$$

   – All purchased seats at top price.

$$TopPricePurchased(eID) := \Pi_{eID}\sigma_{TopPrice.eID \neq Purchase.eID \wedge TopPrice.sID \neq Purchase.sID} (TopPrice \bowtie Purchase)$$

$$Answer(eID, Event.name) := \Pi_{eID,Event.name}\sigma_{Event.name=Event.eID} ((LowerPricePurchased \cap HigherPricePurchased) \bowtie Event)$$

5. For each venue in New York, find the least expensive and the most expensive ticket price for a seat in that venue (for any event) in 2015. Report the venue ID, venue name, lowest price and highest price.
   – All Venues in New York where events happened in 2015.

$$AllVenuesInNY(vID, Venue.name, eID) := \Pi_{vID,venue.name,eID}\sigma_{venue.city='NewYork' \wedge venue.vID=event.venue \wedge when.year=2015} (Venue \bowtie Event)$$

   – All seats available at all venues in NY.

$$AllSeats(vID, Venue.name, eID, sID) := \Pi_{vID,Venue.name,Event.eID,sID}\sigma_{AllVenuesInNY.eID=Ticket.eID} (AllVenuesInNY \bowtie Ticket)$$

– All seats not at highest price.

$NotHighestPrice(vID, Venue.name, eID, sID) :=$
$\Pi_{vID,venue.name,T1.eID,T1.sID}\sigma_{AllVenuesInNY.eID=T1.eID=T2.eID \wedge T1.price<T2.price}$
$(AllVenuesInNY \bowtie (\rho_{Ticket}T1 \times \rho_{Ticket}T2))$

– Most expensive seat.

$TopSeat(vID, Venue.name, sID) := \Pi_{vID,venue.name,sID}(AllSeats - NotHighestPrice)$

– Most expensive ticket.

$HighestPrice(vID, Venue.name, price) :=$
$\Pi_{vID,venue.name,price}\sigma_{TopSeat.sID=Ticket.sID}(TopSeat \bowtie Ticket)$

– All seats not at lower price.

$NotLowerPrice(vID, Venue.name, eID, sID) := \Pi_{vID,venue.name,T1.eID,T1.sID}$
$\sigma_{AllVenuesInNY.eID=T1.eID=T2.eID \wedge T1.price>T2.price}(AllVenuesInNY \bowtie (\rho_{Ticket}T1 \times \rho_{Ticket}T2))$

– Least expensive seat.

$LeastSeat(vID, Venue.name, sID) := \Pi_{vID,venue.name,sID}(AllSeats - NotLowerPrice)$

– Least expensive ticket.

$LeastPrice(vID, Venue.name, price) := \Pi_{vID,venue.name,price}\sigma_{LeastSeat.sID=Ticket.sID}$
$(LeastSeat \bowtie Ticket)$

$Answer(vID, venuename, lowestprice, highestprice) := \Pi_{vID,venue.name,LeastPrice.price,HighestPrice.price}$
$\sigma_{HighestPrice.vID=LeastPrice.vID \wedge HighestPrice.name=LeastPrice.name}(HighestPrice \bowtie LeastPrice)$

6. Find the venue with the greatest number of accessible seats. Report the venue name and city.

   Cannot be expressed.

7. Find every event for which one user bought every ticket for an accessible seat. Report the event name, date and city, and username of the person who bought all the accessible seats.

   – All events with accessible seats.

   $AllAccessible(eID, sID) := \Pi_{eID,sID}\sigma_{Seat.venue=Event.venue \wedge Seat.accessible=true}(Seat \bowtie Event)$

   – All tickets for accessible seats.

   $AllAccessibleTickets(eID, sID) :=$
   $\Pi_{Ticket.eID,Ticket.sID}\sigma_{AllAccessible.eID=Ticket.eID \wedge AllAccessible.sID=Ticket.sID}(AllAccessible \bowtie Ticket)$

   – People who bought accessible seats.

   $AccessiblePurchased(eID, sID, userName) := \Pi_{Purchase.eID,Purchase.sID,userName}$
   $\sigma_{AllAccessible.eID=Purchase.eID \wedge AllAccessible.sID=Purchase.sID}(AllAccessible \bowtie Purchase)$

   – One user who bought all accessible seats.

   $OneUserBoughtAll(eID, userName) := \Pi_{A1.eID,A1.userName}$
   $\sigma_{A1.eID=A2.eID \wedge A1.userName=A2.userName \wedge A1.sID \neq A2.sID}(\rho_{AccessiblePurchased}A1 \times \rho_{AccessiblePurchased}A2)$

$Answer(Event.name, Event.when, Venue.city, userName) :=$
$\Pi_{Event.name,Event.when.date,Venue.city,userName}\sigma_{OneUserBoughtAll.eID=Event.eID \wedge Event.venue=Venue.vID}$
$(OneUserBoughtAll \bowtie Event \bowtie Venue)$

8. Find the events in Toronto in 2015 at which at least half of the seats were unsold. Report the event ID, name and date.

   Cannot be expressed.

9. Find all users who have bought a ticket to at least one event, but have never bought two or more tickets to one event. Report the username, last name and first name.

   – Users who bought at least one ticket.

   $AtLeastOne(eID, sID, userName) := \Pi_{Purchase.eID,Purchase.sID,userName}$
   $\sigma_{Seat.sID=Purchase.sID \wedge Event.eID=Purchase.eID \wedge Seat.venue=Event.venue}(Seat \bowtie Event \bowtie Purchase)$

   – Users who bought at least two tickets.

   $AtLeastTwo(userName) := \Pi_{userName}\sigma_{R1.eID=R2.eID \wedge R1.userName=R2.userName \wedge R1.sID \neq R2.sID}$
   $(\rho_{AtLeastOne}R1 \times \rho_{AtLeastOne}R2)$

   – Users who bought a ticket to atleast one event but never bought two or more tickets.

   $UserIDs(userName) := \Pi_{userName}AtLeastOne - AtLeastTwo$

   $Answer(userName, lastName, firstName) :=$
   $\Pi_{User.userName,lastName,firstName}\sigma_{UserIDs.username=User.username}(UserIDS \bowtie User)$

10. Find all users who have bought a ticket for each concert that the Rolling Stones have played in Toronto in 2000 or since. Report the usernames.

    – All concerts that Rolling Stones has played in Toronto since 2000.

    $Concerts(eID) :=$
    $\Pi_{eID}\sigma_{type='Concert' \wedge Event.name='RollingStones' \wedge Event.venue=Venue.vID \wedge when.year \geq 2000 \wedge city='Toronto'}$
    $(Event \bowtie Venue)$

    – All tickets purchasable for all concerts.

    $AllTickets(eID, sID) := \Pi_{Ticket.eID,Ticket.sID}\sigma_{Concerts.eID=Ticket.eID}(Concerts \bowtie Ticket)$

    – All purchased tickets.

    $Purchased(eID, sID, userName) := \Pi_{Purchase.eID,Purchase.sID,Purchase.userName}$
    $\sigma_{AllTickets.eID=Purchase.eID \wedge AllTickets.sID=Purchase.sID}(AllTickets \bowtie Purchase)$

    $Answer(userName) :=$
    $\Pi_{P1.userName}\sigma_{P!.eID=P2.eID \wedge P1.sID \neq P2.sID \wedge P1.userName=P2.userName}(\rho_{Purchased}P1 \times \rho_{Purchased}P2)$

11. Find all venues at which the Rolling Stones have played a sold out concert (*i.e.* the event is a concert and its name is "Rolling Stones"). For each of these venues, report the name of the owner of the venue.

– All venues and events that Rolling Stones has played.

$$AllVenues(eID, vID) := \Pi_{eID,vID}\sigma_{type='Concert' \wedge Event.name='Rolling\ Stones' \wedge Event.venue=Venue.vID}(Event \bowtie Venue)$$

– All tickets purchasable for Rolling Stone concerts.

$$AllTickets(eID, sID) := \Pi_{Ticket.eID,Ticket.sID}\sigma_{AllVenues.eID=Ticket.eID}(AllVenues \bowtie Ticket)$$

– All purchased Rolling Stone tickets.

$$Purchased(eID, sID) := \Pi_{Purchase.eID,Purchase.sID}$$
$$\sigma_{AllTickets.eID=Purchase.eID \wedge AllTickets.sID=Purchase.sID}(AllTickets \bowtie Purchase)$$

– Not purchased Rolling Stone tickets.

$$NotPurchased(eID, sID) := \Pi_{AllTickets.eID,AllTickets.sID}(AllTickets - AllPurchased)$$

– All events where Rolling Stone concert was sold out.

$$AllEvents(eID) :=$$
$$\Pi_{AllPurchased.eID}\sigma_{NotPurchased.eID \neq AllPurchased.eID}(AllPurchased \bowtie NotPurchased)$$

$$Answer(owner) :=$$
$$\Pi_{Venue.owner}\sigma_{AllEvents.eID=Event.eID \wedge Event.venue=Venue.vID}(AllEvents \bowtie Event \bowtie Venue)$$

12. Find all users who bought a ticket for either the first talk or the second talk in Toronto in 2016. Report their email addresses. Note: There may be only one talk in Toronto in 2016, in which case, the only people in the answer will have bought a ticket to that first (and only) talk. If there is *no* talk in Toronto in 2016, the answer should be an empty relation.

– All talks in Toronto in 2016.

$$AllTalks(eID, when) := \Pi_{eID,when}\sigma_{type='Talk' \wedge when.year=2016 \wedge city='Toronto' \wedge Event.venue=Venue.vID}(Event \bowtie Venue)$$

– Not the first talk in Toronto.

$$NotFirstTalk(eID, when) := \Pi_{T1.eID,T1.when}\sigma_{T1.when>T2.when}(\rho_{AllTalks}T1 \times \rho_{AllTalks}T2)$$

– First talk in Toronto.

$$FirstTalk(eID, when) := \Pi_{eID,when}(AllTalks - NotFirstTalk)$$

– Not the second talk in Toronto.

$$NotSecondTalk(eID, when) := \Pi_{T3.eID,T3.when}\sigma_{T3.when>T4.when}(\rho_{NotFirstTalk}T3 \times \rho_{NotFirstTalk}T4)$$

– Second talk in Toronto.

$$SecondTalk(eID, when) := \Pi_{eID,when}(NotFirstTalk - NotSecondTalk)$$

– Users who bought tickets for the first talk.

$FirstTalkUsers(email) := \Pi_{email}\sigma_{FirstTalk.eID=Purchase.eID \wedge Purchase.userName=User.userName}(FirstTalk \bowtie Purchase \bowtie User)$

– Users who bought tickets for the second talk.

$SecondTalkUsers(email) := \Pi_{email}\sigma_{SecondTalk.eID=Purchase.eID \wedge Purchase.userName=User.userName}(SecondTalk \bowtie Purchase \bowtie User)$

$Answer(email) := (FirstTalkUsers \cup SecondTalkUsers)$

Part 2: Additional Integrity Constraints

1. A ticket for an event must be for a seat in the same venue as the event venue.

   $\sigma_{Seat.sID \neq Ticket.sID \wedge Event.eID \neq Ticket.eID \wedge Seat.venue \neq Event.venue}(Seat \bowtie Event \bowtie Ticket) = \emptyset$

2. A ticket for an event cannot be purchased after the event.

   $\sigma_{Purchase.when > Event.when}(Event \bowtie Purchase) = \emptyset$