Assignment 1: Sample Solutions

Note that there are multiple correct answers to all of these questions.

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

   – Concerts in Toronto in 2016.
   RelevantEvent(eID) :=
   \( \Pi_{eID}^\sigma_{\text{venue}=\text{vID} \land \text{type}='\text{concert}' \land \text{when.year}=2016 \land \text{city}='\text{Toronto}'}(\text{Event} \times \text{Venue}) \)

   – sID is a seat for eID costing under $25.
   CheapSeat(eID,sID) :=
   \( \Pi_{eID,sID}^\sigma_{\text{price}<25}(\text{Ticket}) \)

   – sID is a seat for eID that has not been sold.
   UnsoldSeat(eID,sID) :=
   \( (\Pi_{eID,sID}\text{Ticket}) - (\Pi_{eID,sID}\text{Purchase}) \)

   – sID is a seat for eID that has not been sold.
   Answer(eID) :=
   RelevantEvent \cap \Pi_{eID}(\text{CheapSet} \cap \text{UnsoldSeat})

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.

   – sID is a seat for eID costing over $200.
   PriceySeat(eID,sID) :=
   \( \Pi_{eID,sID}^\sigma_{\text{price}>200}(\text{Ticket}) \)

   – This user bought a pricey seat.
   BoughtPricey(username) :=
   \( \Pi_{\text{username}}(\text{PriceySeat} \bowtie \text{Purchase}) \)

   – This user bought a ticket to a musical.
   BoughtMusical(username) :=
   \( \Pi_{\text{username}}^\sigma_{\text{type}='\text{musical'}}(\text{Purchase} \bowtie \text{Event}) \)

   – The big spenders who hate musicals.
   BSHM(username) :=
   BoughtPricey – BoughtMusical

   – Final answer.
   Answer(username, price, eID, when, name) :=
   \( \Pi_{\text{username, price, eID, when, name}^\sigma_{\text{price}>200}}(\text{Event} \bowtie (\rho_{\text{when} \rightarrow \text{purchaseTime}}\text{Purchase}) \bowtie \text{BSHM} \bowtie \text{Ticket}) \)

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

   – This user bought multiple tickets to an event in year when.year.
   – It doesn’t matter whether we keep the date and time from P1 or P2, since only the year matters,
– and they both have the same year.

\[
\text{Twice}(\text{username}, \text{when}) :=
\Pi_{P1\.username,1\.when} \sigma_{P1\.username = P2\.username} \Pi_{P1\.eID, P1\.sID, P1\.when, P1\.year = P2\.when, P2\.year}
\]

– This user had a Twice year at when1 and when2, and when2 is in the year after when1.

\[
\text{Consecutive}(\text{username}, \text{when1}, \text{when2}) :=
\Pi_{\text{username}, \text{when1}, \text{when2}} \sigma_{\text{when1} = \text{when2} \land \text{when1} + 1 = \text{when2} \land \text{when1} \land \text{when2} \land \text{when1} \land \text{when2}}
\]

– Final answer.

\[
\text{Answer}(\text{username}, \text{email}) :=
\sigma_{\text{username}, \text{email}}(\text{Consecutive} \bowtie User)
\]

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.

– Events in 2015 or earlier.

\[
\text{RelevantEvent}(eID) :=
\Pi_{eID} \sigma_{\text{when.year} \leq 2015}(\text{Event})
\]

– Maximum price for any ticket for eID.

\[
\text{RelevantTicket}(eID, price) :=
\Pi_{eID, sID, price}(\text{RelevantEvent} \bowtie Ticket)
\]

– This is not the maximum price for a ticket to this event.

\[
\text{NotMaxPrice}(eID, price) :=
\Pi_{T1.eID, T1.price} \sigma_{T1.eID = eID} \Pi_{T2.eID, T1.price} \sigma_{T1.price \geq T2.price}(\rho_{T1.Ticket} \times \rho_{T2.Ticket})
\]

– The maximum price for a ticket to this event.

\[
\text{MaxPrice}(eID, price) :=
(\Pi_{eID, price}(\text{RelevantTicket} - \text{NotMaxPrice})
\]

– eID is an event at which none of the seats at its top ticket price were sold.

\[
\text{UnsoldTop}(eID) :=
\Pi_{eID}(\text{MaxPrice} - \Pi_{eID, price}(\text{Purchase} \bowtie Ticket))
\]

– An seat that is available for purchase for this event, and at less than the top price.

\[
\text{CheaperAvailable}(eID, sID) :=
\Pi_{eID, sID} \sigma_{\text{ticket.price} < \text{MaxPrice.price}}(\text{Ticket} \bowtie \text{MaxPrice})
\]

– An unsold seat that is available for purchase for this event, and at less than the top price.

\[
\text{UnsoldCheaper}(eID, sID) :=
\text{CheaperAvailable} - (\Pi_{eID, sID}(\text{Purchase})
\]

– An event at which every seat not at its top price was sold.

– By subtracting from CheaperAvailable, we only get events that actually had some tickets
   – available below the top price. If we wanted to include events that simply had no tickets
   – at a lower price, arguing that it is trivially true that "every seat at a lower price was sold",
   – then we would subtract from every eID in the Event table. Both solutions were accepted
   – for full marks.

\[
\text{SoldOutCheaper}(eID) :=
(\Pi_{eID}(\text{CheaperAvailable}) - (\Pi_{eID}(\text{UnsoldCheaper})) \bowtie \text{Event})
\]

– Final answer.

\[
\text{Answer}(eID, name) :=
\Pi_{eID, name}((\text{SoldOutTop} \cap \text{SoldOutCheaper}) \bowtie \text{Event})
\]
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.

- Venues in New York.
  \[ NYVenue(venue) := \Pi_{vID} \sigma_{\text{city}='NewYork'}(Venue) \]
- A price for some ticket to some event in a New York venue.
  \[ VenuePrice(vID,\text{price}) := \Pi_{\text{venue,price}} \sigma_{\text{when.year}=2015}(\text{Ticket} \bowtie (\text{Event} \bowtie NYVenue)) \]
- Not the maximum price for a ticket to some event in this New York venue.
  \[ NotMaxPrice(vID,\text{price}) := VenuePrice - NotMaxPrice \]
- Maximum price for a ticket to some event at this New York venue.
  \[ MaxPrice(vID,\text{price}) := VenuePrice - NotMaxPrice \]
- Not the minimum price for a ticket to some event at this New York venue.
  \[ NotMinPrice(vID,\text{price}) := VenuePrice - NotMinPrice \]
- The minimum price for a ticket to some event in that New York venue.
  \[ MinPrice(vID,\text{price}) := VenuePrice - NotMinPrice \]
- Min and Max prices for a ticket to some event in that New York venue.
  \[ Spread(vID,\text{lowest},\text{highest}) := \Pi_{vID} \sigma_{MinPrice.price,MaxPrice.price}(MinPrice \times MaxPrice) \]
- Final answer.
  \[ Answer(vID,\text{name},\text{lowest},\text{highest}) := \Pi_{vID} (Spread \bowtie Venue) \]

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

- Can’t be solved with the language we are using.

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.

- An accessible seat for this event.
  \[ Accessible(sID,eID) := \Pi_{eID,sID} \sigma_{\text{accessible}='true'}(Ticket \bowtie Seat) \]
- This user should have bought this ticket to this event if they are to be included in the result.
  \[ ShouldBuy(username,sID,eID) := (\Pi_{username} User \times Accessible) \]
- This user did buy this ticket to this event.
  \[ DidBuy(username,sID,eID) := \Pi_{username,sID,eID}(Accessible \bowtie Purchase) \]
- This user didn’t buy this ticket to this event, and they should have bought.
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.

- Can't be solved with the language we are using.

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.

- User has purchased a ticket to something.
  \[ Purchased(username) := \Pi_{username}(Purchase) \]

- User has purchased two or more tickets to one event.
  \[ Multiples(username) := \Pi_{P1.username,P2.username,P1.eID,P2.eID,P1.sID,P2.sID}(\rho_{P1.Purchase} \times \rho_{P2.Purchase}) \]

- User has not purchased two or more tickets to one event.
  \[ NoMultiples(username) := \Pi_{username}(User) - Multiples \]

- Final answer.
  \[ Answer(username,lastName,firstName) := \Pi_{username,lastName,firstName}(\Pi_{username}(Purchase) - NoMultiples) \]

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.

- Rolling Stones concerts in Toronto in 2000 or since.
  \[ Stones(eID) := \Pi_{eID}(Event.venue=Venue.vID \land \text{city}=\text{Toronto} \land \text{when.year} \geq 2000 \land \text{name}='RollingStones' \land \text{type}=\text{concert}' \land \text{Event} \times \text{Venue} \]

- Straight-forward division.
  \[ ShouldHavePurchased(username, eID) := x \quad (\Pi_{username}(Purchase \times Stones) \]

- Final answer.
  \[ Answer(username) := \Pi_{username}(ShouldHavePurchased - Purchased) \]

\[ Answer(username) := \Pi_{username}(Purchase - Missing) \]
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.

- A Rolling Stones concert.
  \[ \text{Stones}(eID) := \Pi_{eID} \sigma_{\text{Event.name}=\text{RollingStones} \land \text{Event.type}=\text{concert}}(\text{Event}) \]

- A Rolling Stones concert that was not sold out.
  \[ \text{NotSoldout}(eID) := \Pi_{eID} ([\Pi_{eID,sID} \text{Stones} \bowtie \text{Ticket}] - ([\Pi_{eID,sID} \text{Purchase} \bowtie \text{Stones}]) \]

- A sold-out Rolling Stones concert.
  \[ \text{SoldOut}(eID) := \text{Stones} - \text{NotSoldout} \]

- An owner of one of the venues of interest.
  \[ \text{Answer}(\text{owner}) := \Pi_{\text{owner}} \sigma_{\text{Event.venue}=\text{Venue.vID} \land \text{Soldout.eID}=\text{Event.eID}}(\text{Soldout} \times \text{Event} \times \text{Venue}) \]

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.

  \[ \text{TOTalk}(eID, when) := \Pi_{eID, when} \sigma_{\text{city}=\text{Toronto} \land \text{year}=2016 \land \text{type}=\text{talk}}(\text{Event}) \]

- Not the first talk in Toronto in 2016.
  \[ \text{NotFirst}(eID, when) := \Pi_{E1,eID,E1.when} \sigma_{E1.when>E2.when}(\rho_{E1,TOTalk} \times \rho_{E2,TTOTalk}) \]

- The first talk in Toronto in 2016.
  \[ \text{First}(eID, when) := \text{TOTalk} - \text{NotFirst} \]

- Not the second talk in Toronto in 2016.
  \[ \text{NotSecond}(eID, when) := \Pi_{E1,eID,E1.when} \sigma_{E1.when>E2.when}(\rho_{E1,\text{NotFirst}} \times \rho_{E2,\text{NotFirst}}) \]

- The second talk in Toronto in 2016.
  \[ \text{Second}(eID, when) := \text{NotFirst} - \text{NotSecond} \]

- First or second talk in Toronto in 2016.
  \[ \text{Early}(eID) := (\Pi_{eID,\text{First}}) \cup (\Pi_{eID,\text{Second}}) \]

- Bought a ticket to the first or second talk in Toronto in 2016.
  \[ \text{BoughtTicket}(\text{username}) := \Pi_{\text{username}}(\text{Early} \bowtie \text{Purchase}) \]

- Email addresses of those people.
  \[ \text{Answer}(\text{email}) := \Pi_{\text{email}}(\text{BoughtTicket} \bowtie \text{User}) \]
Part 2: Additional 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 ticket for an event must be for a seat in the same venue as the event venue.

\[
s_{\text{Ticket}.sID=\text{Seat}.sID \land \text{Event.venue} \neq \text{Seat.venue}}(\text{Event} \bowtie \text{Ticket} \times \text{Seat}) = \emptyset
\]

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

\[
s_{\text{Purchase.eID}=\text{Event.eID} \land \text{Purchase.when} > \text{Event.when}}(\text{Purchase} \times \text{Event}) = \emptyset
\]