

Assignment 1

Part 1: Create an instance

CityInstance

cityID	name	country
1	Toronto	Canada
2	Montreal	Canada

TripInstance

tripID	startCity	length
1	1	4

TransportActivityInstance

aID	destCity	mode
4	2	train

TourActivityInstance

aID	name	duration	extra
1	ROM	5	False
2	CN Tower	6	False
3	Toronto Zoo	11	False
5	Old Montreal	5	False
6	Mont Royal	6	True

ItineraryInstance

tripID	day	step	cityID	activityID
1	1	1	1	1
1	1	2	1	2
1	2	1	1	3
1	3	1	1	4
1	4	1	2	5
1	4	2	2	6

StaffInstance

staffID	name	hireDate
1	Me	20240101

TravellerInstance

tID	name	email	citizenship
1	John Doe	a@mail.com	Canada
2	Jane Doe	b@mail.com	Canada
3	Lorem Is	c@mail.com	Canada

BookingInstance

bID	tripID	startDate	traveller	bookedBy	price
1	1	20240131	1	1	500
2	1	20240131	2	1	500
3	1	20240131	3	1	500

ExtraBookingInstance

bID	activity	price
3	6	200

Part 2: Give the output of a query

QueryOutput

duration
5
6
11

Part 3: Violating a constraint

CityInstance

cityID	name	country
1	Toronto	Canada

TripInstance

tripID	startCity	length
1	1	1

StaffRequiredInstance

activityID	role
1	Guide

TourActivityInstance

aID	name	duration	extra
1	CN Tower	6	False

ItineraryInstance

tripID	day	step	cityID	activityID
1	1	1	1	1

StaffInstance

staffID	name	hireDate
1	Me	20240101

TravellerInstance

tID	name	email	citizenship
1	John Doe	a@mail.com	Canada

BookingInstance

bID	tripID	startDate	traveller	bookedBy	price
1	1	20240131	1	1	500

Part 4: Queries

- Every country and tripID tuple where the trip visited the country.

$$Countries(country, tripID) := \Pi_{country, tripID}(City \bowtie Itinerary)$$

- Every tripID that has visited at least three countries.

$$AtLeastThrice(tripID) :=$$

$$\Pi_{T1.tripID} \left[\sigma_{\substack{T1.tripID=T2.tripID=T3.tripID \\ T1.country < T2.country < T3.country}} \left(\rho_{T1} Countries \times \rho_{T2} Countries \times \rho_{T3} Countries \right) \right]$$

- Every tripID that has visited at least four countries.

$$AtLeastFour(tripID) :=$$

$$\Pi_{T1.tripID} \left[\sigma_{\substack{T1.tripID=T2.tripID \\ =T3.tripID=T4.tripID \\ T1.country < T2.country \\ < T3.country < T4.country}} \left(\rho_{T1} Countries \times \rho_{T2} Countries \times \rho_{T3} Countries \times \rho_{T4} Countries \right) \right]$$

- Every tripID that has visited exactly three countries.

$$ExactlyThree(tripID) := AtLeastThrice - AtLeastFour$$

- The tripID and day of every day in any trip.

$$TripDays(tripID, day) := \Pi_{tripID, day} Itinerary$$

- The tripID and day of every day that is not the last day in a trip.

$$NotLastDays(tripID, day) := \Pi_{\substack{T1.tripID, \\ T1.day}} (\sigma_{\substack{T1.tripID = T2.tripID \\ T1.day < T2.day}} (\rho_{T1} TripDays \times \rho_{T2} TripDays))$$

- The tripID and day of every day that is the last day in a trip.

$$LastDay(tripID, day) := TripDays - NotLastDays$$

- The tripID and step of every step on the last day of any trip.

$$LastDaySteps(tripID, day, step) := \Pi_{tripID, day, step} (LastDay \bowtie Itinerary)$$

- The tripID and step of every step that is not the last step on the last day in a trip.

$$NotLastStep(tripID, day, step) := \Pi_{\substack{T1.tripID, \\ T1.day, \\ T1.step}} (\sigma_{\substack{T1.tripID = T2.tripID \\ T1.step < T2.step}} (\rho_{T1} LastDaySteps \times \rho_{T2} LastDaySteps))$$

- The tripID and step of every last step on the last day in a trip.

$$LastStep(tripID, day, step) := LastDaySteps - NotLastStep$$

- The tripID, cityID, and country of every trip's start city.

$$Renamed(tripID, cityID, country) := \Pi_{tripID, cityID, country} (\rho_{T1(tripID, cityID, length)} Trip) \bowtie City$$

- The tripID, cityID, and country of the start city in a trip that visits exactly three countries.

$$Start(tripID, startCityID, startCountry) := \Pi_{tripID, cityID, country} (Itinerary \bowtie ExactlyThree \bowtie Renamed)$$

– The tripID, cityID, and country of the end city in a trip that visits exactly three countries.

$$End(tripID, endCityID, endCountry) := \Pi_{tripID, cityID, country}(Itinerary \bowtie ExactlyThree \bowtie LastStep \bowtie City)$$

$$Start \bowtie End$$

2. – The cityID of every city which is not part of any trip.

$$NoTrip(cityID) := \Pi_{cityID}City - \Pi_{cityID}Itinerary$$

$$\Pi_{name, country}(NoTrip \bowtie City)$$

3. Cannot be expressed.

4. Cannot be expressed.

5. – The bID, activity, and price of every extra booking that is not the most expensive.

$$NotMax(bID, activity, price) :=$$

$$\Pi_{T1.bID, T1.activity, T1.price} \left[\sigma_{\substack{T1.bID \neq T2.bID \\ T1.price < T2.price}} \left(\rho_{T1} ExtraBooking \times \rho_{T2} ExtraBooking \right) \right]$$

– The bID, activity, and price of every extra booking with the maximum price.

$$Max(bID, activity, price) := ExtraBooking - NotMax$$

– The bID, activity, and price of every extra booking that is that is not the second most expensive.

$$NotSecond(bID, activity, price) :=$$

$$\Pi_{T1.bID, T1.activity, T1.price} \left[\sigma_{\substack{T1.bID \neq T2.bID \\ T1.price < T2.price}} \left(\rho_{T1} NotMax \times \rho_{T2} NotMax \right) \right]$$

– The bID, activity, and price of every extra booking with the second greatest price.

$$Second(bID, activity, price) := ExtraBooking - Max - NotSecond$$

$$\Pi_{tripID, activity, price} \left[\left(Max \cup Second \right) \bowtie Booking \right]$$

6. – The staffID of every staff hired before 2020.

$$Before2020(staffID) := \Pi_{staffID}(\sigma_{hireDate < 20200101} Staff)$$

– The staffID and tripID of every 'guide' assignment to a trip where the staff was hired before 2020.

$$Guide(staffID, tripID) := \Pi_{staffID, tripID}(\sigma_{role='guide'} StaffAssignment \bowtie Before2020)$$

– Every possible combination of staffID and tripID where the staff was hired before 2020.

$$EveryPossible(staffID, tripID) := \Pi_{staffID} Before2020 \times \Pi_{tripID} Trip$$

– The staffID of the staff who were not assigned as 'guide' to every trip.

$$Missing(staffID) := \Pi_{staffID}(EveryPossible - Guide)$$

$$Before2020 - Missing$$

7. Note: I solved this before the clarification that returning to a city previously visited at an earlier date counts as visiting a new city.

$$\Pi_{tripID} Itinerary - \Pi_{T1.tripID} \left[\sigma_{\substack{T1.tripID=T2.tripID \\ T1.city=T2.city \\ T1.day < T2.day}} \left(\rho_{T1} Itinerary \times \rho_{T2} Itinerary \right) \right]$$

8. – All bIDs and tripIDs where the booking does not have the earliest start date for the trip.

$$NotEarliestBooking(bID, tripID) := \Pi_{T2.bID, T2.tripID} \left[\sigma_{\substack{T1.tripID=T2.tripID \\ T1.startDate < T2.startDate}} \left(\rho_{T1} Booking \times \rho_{T2} Booking \right) \right]$$

- All bIDs and tripIDs where the booking has the earliest start date for the trip.

$$EarliestBookings(bID, tripID) := \Pi_{bID, tripID} Booking - NotEarliestBooking$$

- The bIDs of all earliest bookings where their trips are booked by at least two other travellers at later start dates.

$$AtLeastThreePeople(bID) := \Pi_{bID} EarliestBookings \cap$$

$$\Pi_{T1.bID} \left[\sigma_{\substack{T1.tripID=T2.tripID=T3.tripID \\ T1.startDate < T2.startDate \\ T1.startDate < T3.startDate \\ T1.traveller < T2.traveller < T3.traveller}} \left(\rho_{T1} Booking \times \rho_{T2} Booking \times \rho_{T3} Booking \right) \right]$$

- All travellers who are not influencers.

$$NotInfluencers(traveller) := \Pi_{traveller} (\Pi_{traveller, bID} Booking - (\Pi_{traveller} Booking \times AtLeastThreePeople))$$

$$\Pi_{name, email} ((\Pi_{traveller} Booking - NotInfluencers) \bowtie Traveller)$$

9. Cannot be expressed.

$$10. \Pi_{T1.staffID, T2.staffID} \left[\sigma_{\substack{T1.staffID < T2.staffID \\ T1.tripID=T2.tripID \\ T1.startDate=T2.startDate}} (\rho_{T1} StaffAssignment \times \rho_{T2} StaffAssignment) \right]$$

Part 5: Additional integrity constraints

1. – The tripID and day of every first day of a trip listed in Itinerary.

$$FirstDay(tripID, day) := \Pi_{tripID, day} Itinerary -$$

$$\Pi_{T2.tripID, T2.day} \left[\sigma_{\substack{T1.tripID=T2.tripID \\ T1.day \hat{=} T2.day}} \left(\rho_{T1} Itinerary \times \rho_{T2} Itinerary \right) \right]$$

$$\Pi_{tripID}(\sigma_{day>1} FirstDay) \cup \Pi_{tripID} \left(\Pi_{tripID, day} Itinerary - \Pi_{tripID, day}(\sigma_{day=1} Itinerary) - \right.$$

$$\left. \Pi_{T2.tripID, T2.day} \left[\sigma_{\substack{T1.tripID=T2.tripID \\ T1.day+1 \hat{=} T2.day}} \left(\rho_{T1} Itinerary \times \rho_{T2} Itinerary \right) \right] \right) = \emptyset$$

2. $\rho_{T1(activityID)}(\Pi_{activity} ExtraBooking) - \Pi_{activityID} Itinerary = \emptyset$

3. $\Pi_{traveller, country}[Booking \bowtie Itinerary \bowtie City] -$

$$\rho_{T2(traveller, country)}(\Pi_{tID, citizenship} Traveller) - \Pi_{traveller, country}(\sigma_{expiry>startDate}(Visa \bowtie Booking)) = \emptyset$$

4. $\Pi_{T1.tripID} \left[\sigma_{\substack{T1.tripID=T2.tripID=T3.tripID \\ T1.day \hat{=} T2.day \hat{=} T3.day \\ T1.country < T2.country < T3.country}} \right.$

$$\left. \left(\rho_{T1}(City \bowtie Itinerary) \times \rho_{T2}(City \bowtie Itinerary) \times \rho_{T3}(City \bowtie Itinerary) \right) \right] = \emptyset$$

5. – The tripID, day, and step of every transport activity in the itinerary.

$$TransportItinerary(tripID, day, step) :=$$

$$\Pi_{tripID, day, step, activityID}(Itinerary \bowtie (\rho_{R1(activityID)} \Pi_{aID} TransportActivity))$$

- The tripID, day, and step of every transport activity that is the last step of a day in a trip.

$$LastStepTransport(tripID, day, step) := TransportItinerary -$$

$$\Pi_{T1.tripID, T1.day, T1.step} \left[\sigma_{\substack{T1.tripID=T2.tripID \\ T1.day \hat{=} T2.day \\ T1.step < T2.step}} \left(\rho_{T1} Itinerary \times \rho_{T2} Itinerary \right) \right]$$

$$\Pi_{T1.tripID} \left[\sigma_{\substack{T1.tripID=T2.tripID \\ T1.day \hat{=} T2.day \\ T1.step+1 \hat{=} T2.step}} \left(\rho_{T1} TransportItinerary \times \rho_{T2} TransportItinerary \right) \right] \cup$$

$$\Pi_{T3.tripID} \left[\sigma_{\substack{T3.tripID=T4.tripID \\ T3.day+1 \hat{=} T4.day}} \left(\rho_{T3}(\Pi_{tripID, day} LastStepTransport) \times \rho_{T4}(\sigma_{step=1} TransportItinerary) \right) \right] \\ = \emptyset$$

6. Cannot be expressed.