Database Systems, CSCI 4380-01
Homework # 1(a+b) Answers

Database Description. Suppose you are given the following database for AirBnB for a single city (shortened from the actual public Airbnb database):

```
hosts(host_id, host_name, host_url, host_since, host_location, host_about,
  host_response_time, host_acceptance_rate, host_is_superhost, host_identity_verified)

neighborhoods(neighbourhood_group,neighbourhood)

listings(listing_id,name,host_id,neighbourhood,latitude,longitude,
  room_type, price, minimum_nights,number_of_reviews, scores_rating, scores_accuracy,
  scores_cleanliness, scores_checkin, scores_communication,scores_location)

calendar(listing_id,date, available, price)

reviews(review_id, listing_id, date, reviewer_id, reviewer_name, comments)
```

`Hosts` are individuals who rent out their apartments.

`Neighbourhoods` contains the names of specific areas of the city (e.g. `Manhattan` (neighbourhood group) and `SoHo` (neighbourhood.

`Listings` are for specific properties being rented. A listing can be for an entire home, a room in a home or another property type (given by `room_type`). Each listing contains information about where the property is, where there is a requirement of minimum number of nights to book and the average scores (out of 5) for different aspects of the property: rating, accuracy of description, cleanliness, checkin process, location and the communication with the host. The price attribute in listings is the average price for this property.

`Calendar` contains the price of the property for each day of the year and whether it is available or booked (`available` is true or false) on that day.

Note: All date fields are formatted as `mon-day-year`, e.g. `01-31-2019`. You can assume that you can check if a date value `X` comes after another value `Y` by checking whether `X > Y`.

Write the following queries using relational algebra (pay attention to the attributes required in the output!):

Question 1. The following queries only need a single SELECT (σ), followed by a PROJECT (π) and RENAMING (ρ) as necessary:

(a) Return the id and name of all listings in the `Murray Hill` neighbourhood for a whole house (`room_type`) with 4.5 or higher scores for accuracy, cleanliness and communication and is for less than $80.

Answer.

$$\Pi_{listing\_id,name}(\sigma_C Listings)$$

where $C =$ `neighbourhood='Murray Hill' and room_type = 'whole house' and scores_accuracy` $\geq$ `4.5 and scores_cleanliness` $\geq$ `4.5 and scores_communication` $\geq$ `4.5 and price` $<$ `80`.

(b) Return the id, name and URL of all superhosts hosts who have been a host since 2016 and have an acceptance rate of 100%.

Answer.

$$\Pi_{host\_id,host\_name,host\_URL}(\sigma_C Hosts)$$

where $C =$ `host_since` $\geq$ `'1/1/2016' and host_since` $\leq$ `'12/31/2016' and host_acceptance_rate = 100`.

Question 2. The following queries combine SELECT (σ), SET operations ($\cap, \cup, -$), PROJECTION (π) and RENAMING (ρ) as necessary:

(a) Return id of listings that are either available for at least one day in the month of November 2019 or are hosted by host with id 2845.

Answer.

$$\begin{aligned} R1 &= \Pi_{listing\_id}(\sigma_{host\_id=2845}\, Listings) \\ R2 &= \Pi_{listing\_id}(\sigma_{available=True \text{ and } date<='11/31/2019' \text{ and } date>='11/1/2019'}\, Calendar) \\ Result &= R1 \cup R2 \end{aligned}$$

(b) Return the id of listings that are not available on any day in November 2019 and have at least 100 reviews.

Answer.

$$\begin{aligned} R1 &= \Pi_{listing\_id}(\sigma_{number\_of\_reviews>=100}\, Listings) \\ R2 &= \Pi_{listing\_id}(\sigma_{available=True \text{ and } date<='11/31/2019' \text{ and } date>='11/1/2019'}\, Calendar) \\ Result &= R1 - R2 \end{aligned}$$

Question 3. The following queries combine SELECT (σ) statements with any number of JOINS as neede ($\bowtie$, theta or natural) (or CARTESIAN PRODUCT), followed by a PROJECT (π) and RENAMING (ρ) as necessary:

(a) Return the id and name of all hosts who have at least one listing with score 3 or lower for cleanliness for a property that is available on date `'9/10/2019'`.

Answer.

$$
\begin{aligned}
R1 &= \sigma_{scores\_cleanliness<=3}\ Listings \\
R2 &= \Pi_{listing\_id}(\sigma_{available=True \text{ and } date='9/10/2019'}\ Calendar) \\
Result &= \Pi_{host\_id,host\_name}(R1 \bowtie R2 \bowtie Hosts)
\end{aligned}
$$

Note: The projection for listing_id for Calendar is NECESSARY if we use natural join. Otherwise, the natural join would also require that the price listed in Calendar and Listing be the same. This is not necessary.

(b) Return the id, name of all listings that has a review by a reviewer (`reviews.reviewer_name`) that has the same name as the host of the listing (`hosts.host_name`).

Answer.

$$
\begin{aligned}
R1(lid1, name, hname) &= \Pi_{listing\_id,name,host\_name}(Listings \bowtie Hosts) \\
R2 &= R1 \bowtie_{lid1=listing\_id \text{ and } hname=reviewer\_name}\ Reviews \\
Result &= \Pi_{listing\_id,name}(R2)
\end{aligned}
$$

Note: If you did a natural join between R1 and Reviews, you still need to use selection to make sure that the name of the reviewer is the same as the host. You can achieve the same by renaming the attributes. Just be careful to use each one properly. If you are using theta-join, then you must rename all attributes so that there are no attributes in commmon between the relations.

Question 4. Write the following queries using relational algebra using any combination of operators (pay attention to the attributes required in the output!):

(a) Return id, name, latitude and longitude of listings that are available in two consecutive days in November 2019. Return their name and ID. (Note: you can compare two days as follows: `day1 = day2+1` to check that they are consecutive.)

Answer.

$$
\begin{aligned}
R0 &= \sigma_{available=True \text{ and } date<='11/31/2019' \text{ and } date>='11/1/2019'} Calendar \\
R1 &= \Pi_{listing\_id,date}(R1) \\
R2(listing\_id, date2) &= R1 \\
R3 &= \sigma_{date=date2+1}\ (R1 \bowtie R2) \\
Result &= \Pi_{listing\_id,name,latitude,longitude}\ (R3 \bowtie Listings)
\end{aligned}
$$

(b) Return the name of neighborhood groups where the listings for `'Entire Home'` rooms (`room_type`) start from $400 despite some listings of this room type having scores lower than 4 (`scores_rating`).

In short, some listings of this type has scores lower than 4 and none of the listings have prices lower than $400.

Answer.

$$
\begin{aligned}
R1 &= Neighborhoods \bowtie (\sigma_{room\_type='Entire\ Home'}(Listings)) \\
R2 &= \Pi_{neigbourhood\_group}(\sigma_{scores\_rating<4}\, R1) \\
R3 &= \Pi_{neigbourhood\_group}(\sigma_{price<400}\, R1) \\
R4 &= \Pi_{neigbourhood\_group}(R1) \\
Result &= R2 - R3
\end{aligned}
$$

Given the room type constraint:

R2: all neigbourhood_groups (NGs) with a score less than 4.

R3: all NGs with price less than 400.

(c) We are searching for potential fake reviewers in this final one.

Return the reviewer id of reviewers who have written a review for at least one listing in all of the `neighbourhood_groups` in the database.

Note that this is a challenging query, so we will for sure give partial credit even if your solution is not perfect! Do your best.

Answer.

$$
\begin{aligned}
R1 &= \Pi_{reviewer\_id} Reviews \;\; \#allreviewers \\
R2 &= \Pi_{neigbourhood\_group} Neighborhoods \;\; \#allneighbourhoodgroups, NGs \\
R3 &= R1 \times R2 \\
R4 &= \Pi_{neigbourhood\_group, reviewer\_id}(Listings \bowtie Reviews \bowtie Neighborhoods) \\
R5 &= \Pi_{reviewer\_id}(R3 - R4) \\
Result &= R1 - R5
\end{aligned}
$$

R3: all possible NG and reviewer pairs possible, R4 is all NG, reviewer pairs that are actually in the database.

R5: will have a reviewer if there is at least one NG that the reviewer did not write a review for

For the result, we simply exlude all those reviewers in R5. These should be the reviewers we want, who have written a review for ALL NGs.

Question 5. You are given the following relations with associated set of functional dependencies. For each relation, find the keys. List all.

(a) $R1(A, B, C, D, E, F, G)$, $\mathcal{F} = \{A \to BC, A \to DEFG, F \to B\}$

Answer. Key: A

(b) $R2(A, B, C, D, E, F, G)$, $\mathcal{F} = \{DE \to AF, AC \to G\}$

Answer. Key: BCDE

(c) $R3(A, B, C, D, E, F, G)$, $\mathcal{F} = \{AB \to CDEF, F \to A\}$

Answer. Keys: ABG, BFG

(d) $R4(A, B, C, D, E, F, G)$, $\mathcal{F} = \{AC \to DE, CDF \to G, BG \to A\}$

Answer. Keys: ABCF, BCFG