Pset 2: Schema design and Query Plans

Assigned: 9/17/2018
Due: 9/24/2018 11:59 PM


The purpose of this problem set is to give you some practice with concepts related to schema design and query planning.

Part 1

Suppose you are creating a database for a hospital. you must keep track of patients, rooms, and doctors.

Specifically you will need to keep track of:

1. The rooms in the hospital. Each room has a unique room number, a building it is in and a size (in ft^2). Rooms can have zero or more patients in them.
2. Patients, their room number, 0 or more doctors assigned to them, their name, ssn, and a status (critical, stable, etc).
3. Doctors, their names, ssn’s and patients they are seeing, and whether or not they are on duty.
4. Buildings, which have names, and rooms.

Problems

1. [4 Points] Draw an ER diagram representing your data. Include a few sentences of justification for why you drew it the way you did.
2. [3 Points] Transform your ER diagram into relations and write out a list of functional dependencies for this schema.
3. [2 Points] In what normal form is the schema from question 2? (you can give the highest normal form). Is your schema redundancy and anomaly free? Justify your answer.
4. [2 points] Suppose you want to ensure that each patient has no more than 2 doctors assigned to them? Is it possible to model it as an ER diagram? If it is, how would the ER diagram look like? How do the relations from question 2 change? Justify your answers and, if applicable, also discuss alternative model designs.
5. [2 points] Assume the ER diagram from Question 1 (every patient has 0 or more doctors assigned to them). If possible, create an ER diagram that ensures that for every patient at least one of their assigned doctors is on duty. Justify your answer.
6. [3 points] Assume the schema from question 2, but we now decide patients can have multiple rooms (maybe for different types of procedures etc). How would your schema change? Suggest a view that will allow legacy programs written across the old database to continue to function. Does your view provide full backwards compatibility? Can you think of any cases that it does not handle properly?

(Views allow us to have logical independence in relational models. You may wish to reference this article discussing how Views are used: [https://en.wikipedia.org/wiki/View_(SQL)](https://en.wikipedia.org/wiki/View_(SQL))

7. [4 Points] We have the following table which keeps track of club information and room reservations.

<table>
<thead>
<tr>
<th>Club Name</th>
<th># Members</th>
<th>Bldng/Room#</th>
<th>StartTime</th>
<th>EndTime</th>
<th>Room Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go Club</td>
<td>8</td>
<td>Student Center/4-140</td>
<td>Sep 13 2018 2:30pm</td>
<td>Sep 13 2018 5:30pm</td>
<td>ClassRoom</td>
</tr>
<tr>
<td>HackMit</td>
<td>800</td>
<td>Stata/32-123</td>
<td>Sep 20 2018 10:30 am</td>
<td>Sep 21 2018 10:30 am</td>
<td>Amphitheater</td>
</tr>
<tr>
<td>Taekwondo</td>
<td>25</td>
<td>Zesiger Center/3-301</td>
<td>Oct 12 2018 1:00 pm</td>
<td>Oct 12 2018 3:00pm</td>
<td>Gym</td>
</tr>
<tr>
<td>HackMit</td>
<td>800</td>
<td>Stata/32-123</td>
<td>Sep 22 2018 10:30 am</td>
<td>Sep 22 2018 11:30 am</td>
<td>Amphitheater</td>
</tr>
<tr>
<td>Taekwondo</td>
<td>25</td>
<td>Zesiger Center/3-301</td>
<td>Oct 12 2018 1:00 pm</td>
<td>Oct 12 2018 3:00pm</td>
<td>Gym</td>
</tr>
<tr>
<td>Taekwondo</td>
<td>25</td>
<td>Zesiger Center/3-301</td>
<td>Oct 19 2018 1:00 pm</td>
<td>Oct 19 3:00pm</td>
<td>Gym</td>
</tr>
<tr>
<td>Go Club</td>
<td>8</td>
<td>Stata/32-123</td>
<td>Dec 8 2018 1:30 pm</td>
<td>Dec 8 2018 4:30 pm</td>
<td>Amphitheater</td>
</tr>
</tbody>
</table>

Here we assume that the club name uniquely identifies a club. We also assume bldng/room# uniquely identifies a specific bldng/room. Only one club can have a room reserved for a particular time period (and there will be only one reservation per room, per time period). Convert this
schema to the BCNF. Include a few sentences of justification for why you chose the tables you did.

**Part 2**

We define the following schemas (from ps1)

Listings(id, name, host_id, price, city, accommodates, reviews_rating, number_of_reviews)

Reviews(id, listing_id, reviewer_id, reviewer_id, date)

For the following queries, what is the equivalent relational algebra and query plan? (assume we use sequential scans and nested loops joins)

1. [1 Point] SELECT id, reviews_rating FROM listings WHERE number_of_reviews>=600
2. [1 Point] SELECT reviewer_id FROM reviews JOIN listings ON reviews.listing_id=listings.id
3. [1 Point] SELECT listing.name, listing.price FROM listings JOIN reviews ON reviews.listing_id=listings.id WHERE reviewer_id = 123 and reviews_rating > 4
4. [2 Points] SELECT reviewer_id FROM reviews JOIN listings ON reviews.listing_id=listings.id GROUP BY reviews.reviewer_id HAVING COUNT(reviews.listing_id)>10 ORDER BY AVG(price) DESC LIMIT 5