Additional Review Problems

Imagine you are given the task of creating a database for a hospital. The hospital wants to keep track of their doctors and staff (including their specialties), their patients, their rooms (when are they occupied and by whom), and appointments.

1. The following is one possible ER diagram for this domain:

   ![ER Diagram](image)

   a. Why does Time have to be an entity set instead of an attribute of Appointment?
   b. What constraints are specified for the Chairs relationship set?
   c. What constraints are specified for the Has Speciality relationship set?
   d. Why does it make sense in this case to separate the two types of people (patients and staff) instead of collapsing them into one entity set?

2. Translate the diagram from (1) into a relational schema.

3. Write SQL commands to create the tables for the relations devised in (2). Make sure to specify the primary keys of each relation and any foreign-key constraints that make sense.
4. Write SQL commands to do the following:
   a. Find the schedule and number of appointments for each doctor on April 1, 2013. Specifically, display the name of the doctor followed by the earliest appointment of the day for that doctor, the latest appointment time of the day, and the total number of appointments that day for each doctor who has at least one appointment on April 1, 2013.
   b. Find the patient and doctor name for all appointments with a surgeon between March 25, 2013 and April 1, 2013.
   c. Find the names of all patients who have had more than 3 appointments in the last week.
   d. Find the names of all staff who make the maximum or minimum salary. Print their name followed by Max or Min, and their specialty (or NULL if they have no specialty).
   e. Find the names of all staff members who do not have an appointment on the afternoon of March 6, 2013.

5. Write relational algebra queries to do the following:
   a. Find the patient and doctor name for all appointments with a surgeon between March 25, 2013 and April 1, 2013.
   b. Find the names of all staff members who do not have an appointment on the afternoon of March 6, 2013.

6. Assume that we’re using a record beginning with a header of field offsets to store the Staff relation:
   - id CHAR(5)
   - name VARCHAR(30)
   - status VARCHAR(10)
   - salary DOUBLE
   - specialty_type VARCHAR(20)

   Assuming 1 byte characters, 4 byte integers, and an 8 byte double for the salary (MySQL’s default). What would the record look like for the following tuple?
   ('12345', 'Doogie Hawser', 'MD', null, 'GP')

7. Assume you were storing variable-length records using the page format discussed in lecture in which each page begins with a header of record offsets. Each record offset is 4 bytes (an integer). By a coincidence, each record you add to this page, page 4, takes up 92 bytes. The page has a total of 960 bytes of memory. How many records could you add before the page would be completely full?
   - What would be the RIDs of these records (list from the first added to the last added)?
   - The record with RID (4, 2) grew and had to be moved: it now is the first record on page 5. What is its RID now and what is its offset value on page 4?

8. Back in the hospital domain, would you choose a B-Tree or a B+Tree to store the Staff records by their primary key? Why?
9. You have two relations, R and S:

<table>
<thead>
<tr>
<th>Relation R</th>
<th>Relation S</th>
</tr>
</thead>
<tbody>
<tr>
<td>a  b  c</td>
<td>c  d  e</td>
</tr>
<tr>
<td>1  2  3</td>
<td>3  4  5</td>
</tr>
<tr>
<td>4  5  6</td>
<td>7  8  9</td>
</tr>
<tr>
<td>7  8  9</td>
<td>9  0  1</td>
</tr>
<tr>
<td>5  5  5</td>
<td></td>
</tr>
</tbody>
</table>

How many rows would the following relation instances contain?

a. the cartesian product of R and S.
b. the natural join of R and S?
c. the left outer join of R and S?
d. the right outer join of R and S?
e. the full outer join of R and S?

Note: Please make sure to go back and study the section notes and lecture notes. These review problems are not comprehensive.