National University of Computer and Emerging Sciences

School of Computing Fall 2022 Islamabad Campus

CS2005: Database Systems (Fall 2022)

Assignment 3

(Deadline: 14th May 2022 11:59 PM)

Assignment groups: This assignment can be done within a group of two (2) students. There is no restriction on the selection of group members. Students are allowed to make groups according to their preferences. The group members must belong to the same sections.

Submission: All submissions MUST be uploaded on Google Classroom. Solutions sent to the emails will not be graded. To avoid last minute problems (unavailability of Google Classroom, load shedding, network down etc.), you are strongly advised to start working on the project from day one.

Combine all your work in one zip file named ROLL_NUM_P_02 (e.g., 20i-1234_P_02.zip). Submit the .zip file in the classroom within a given deadline. Failure to submit according to the above format would result in ZERO marks.

Deadline: Deadline to submit project is **14th May, 2022 11:59 PM.** Correct and timely submission of project is responsibility of every group; hence no relaxation will be given to anyone.

Plagiarism: ZERO marks will be awarded if any significant part of project is found plagiarized..

Healthcare Scheduling

Out task is to develop a healthcare scheduling system for few clinics owned by the same organization on the Eastern coast of USA. The envisioned healthcare system must provide the following high-level functionality.

- Management of clinics, doctors, and patients.
- ***Assignment*** of doctors to the clinics.
- Scheduling patient's ***appointments*** with doctors.
- Management of ***waiting list*** of walk-in patients, i.e., patients without the appointment.
- Management of different type of ***user roles***, i.e., administrator, front desk staff, doctors, and patients.

In this context, your initial analysis of the business requirements are as follows:

1. Each clinic offers a set of specialties/specializations (e.g., orthopedics, cardiology, oncology etc.) and has certain opening hours. Only a doctor with the matching specialty can be assigned to the clinic. Clinics and relevant information are managed by the administrator of the system.
2. Doctors are assigned to the clinics by the administrator of the system. A doctor might be assigned to (i) the same clinic for all working days, (ii) different clinics on different days, (iii) different clinics on the same day, i.e., few hours on different clinics. Moreover, assignments can also be created with recurrence on daily, weekly, or monthly basis and has a range that can be set based on number of occurrences or an end by date. Administrator can also select specific days in weekly and monthly recurrence. For example, if administrator selects Monday and Friday in weekly recurrence

than assignments will be created only on Mondays and Fridays of all the weeks of the recurrence. Administrator can update/change the assignments of doctors at any time. A doctor cannot be assigned to a clinic outside the working hours of that clinic.

3. A patient requests the front desk staff to get the appointment with a doctor (or any doctor with a certain specialty). The front desk staff has visibility to the doctors and their assignments to clinics (however, front desk staff cannot change the assignment). Accordingly front desk staff can schedule appointment of a patient. Like assignments, appointments can also be recurrent. Front desk staff can change/update appointment at any time.
4. Patients can request to book an appointment directly with the doctor or alternatively against a specialty. If an appointment is booked for a specialty, any doctor with the same specialty and assignment on the clinic can be consulted.
5. A patient can get the appointment with the doctor on a clinic only if the doctor is assigned to that clinic for the duration of the timeslot. For example, a patient can get the appointment with a doctor at 1:00 PM on clinic Manhattan, if the doctor is assigned to that clinic from 9:00 AM to 3:00 PM on that day.
6. Patient can login to the healthcare system to see their history of the appointments. However, they cannot see the appointments of other patients.
7. If the assignment of the doctor is cancelled or doctor is on holiday, all the appointments are moved to the waiting list.
8. Front desk staff can view and update the waiting list. If a patient X with an appointment does not show up, the front desk staff can request a patient in the waiting list to consult doctor (in place of X). However, the patient in the waiting list can only consult a doctor with the specialty matches to the specialty for which the appointment is booked.
9. Doctors can see their appointments. Moreover, doctor can see the patients with the same specialty in the waiting list.
10. Admin dashboard can show different statistics related to the assignment of doctors and appointments of patients.

Tasks:

1. Develop an entity-relationship diagram (ERD) for the above requirement specifications. Clearly identify the entities, relationships, multiplicity (i.e., many-one, many-many, etc.), attributes, primary keys, foreign keys, and constraints in your model, using the notations and diagramming rules as described in the textbook and class. NOTE: some basic information is left out of the

above description (e.g., attributes of doctors, patients, clinics etc.), students are supposed to fill the gaps with relevant information.

2. Map the model devised above in the relational data model. You will need to clearly specify the constraints implemented in the relational data model. Your schema should also highlight the user management, i.e., management of administrator, doctor, patient and front desk roles and credentials.
3. Identify different queries that must be supported by the system. **NOTE:** Assume that a calendar view/functionality is available for patients, doctors, and front desk staff to view the assignments and appointments with different filters. This will help you identify the queries. The queries must include creation/updation of clinics, doctors, patients, assignments, appointments etc.
   - The recurrent creation of assignment/appointment must ensure that the clinic timings are observed, and creation does not conflict with an already created assignment/appointment.

   Moreover, identify 15 queries related to the generation of different reports in addition to the queries mentioned below.

   - All patients with appointments on a certain clinic X (weekly, monthly, daily).
   - All patients with appointments related to a certain specialty on all clinics (weekly, monthly, daily).
   - All patient appointments with a certain set of doctors (weekly, monthly, daily)
   - All appointments or assignments on clinic(s) within a certain time.
   - Doctors of same specialty with overlapping assignments on clinic (weekly, monthly, daily)
   - Doctors with least number of appointments on a clinic(s) (weekly, monthly, daily).
   - Doctors with maximum number of appointments related to a specialty. (weekly, monthly, daily).
   - Patient whose recurrent appointment is with the same doctor.
   - Patient whose recurrent appointment is with different doctors.
   - ….

   Finally, queries related to user management, i.e., management of administrator, doctor, patient and front desk roles and credentials.

4. Write relational algebra queries for all view related queries identified in step 3.
5. Insert enough data and demonstrate the full system functionality using some SQL queries.

What to submit

You are required to submit a written report supporting your implementation. This should include your design decisions and rationale on the following issues:

1. ER diagram
2. Relational database schema
3. SQL queries for table, constraints creation.
4. SQL queries for data insertion, fetching etc.
5. SQL and Relational algebra queries for reports.
6. Snapshot of the tuples in tables and results of the SQL queries.

Good Luck!