CPSC 332: File Structures and Database Systems

Department of Computer Science

CALIFORNIA STATE UNIVERSITY
FULLERTON

Spring 2020
Semester Project
Due Date:

Part 1: Due April 15 (11:59pm)
Part 2: Due May 7 (11:59pm)

Project submission should be made via Titanium. Each student should work on the project in groups of 3, and submit a final report, a demo and presentation for the project. Each student of the group will be asked to submit a review of their group members, where each member will assign the percentage of contribution towards the project for their group members. Logically, 33 % contribution from each member is expected. Any student who receives a review lower than 25 % from their group members may receive a lower grade than other members.

Note: Project submissions made through email will not be accepted. Only submissions made on ***Titanium*** shall receive grades.

Maximum Marks: 100

The project comprises 2 main parts, including extra credit options -- for additional functionality or extra credit part.

Design a database for a doctor's office. The database is used for tracking the patients and doctor data. Normalize the database up to the 3rd normal form.

Business Rules –

1. One patient can see many doctors in the same office on the same day.
2. One doctor can also see many patients on the same day.
3. If one doctor sees the same patient multiple times in one day, then it's only recorded once in the database.
4. For each patient, we are recording only primary phone numbers.
5. A doctor can also be a patient.
6. A doctor cannot be his/her own patient.
7. Each patient can be given many prescriptions(medicine).
8. Each patient can be given many tests (medical test).
9. A patient cannot be given a new prescription, test without first visiting the doctor.
10. Make sure that the Doctor ID is the first 2 letters of his first name followed by number. Eg. If the Doctor's name is Steve Gates, then his DoctorID can be ST2048.

Tools to use: MySQL Workbench, Xampp

THINGS TO TURN IN FOR THIS PROJECT -

1. One SCRIPT to create this database (call it DocOffice) with MySQL server.
2. Doctor Robert Stevens is retiring. We need to inform all his patients, and ask them to select a new doctor. For this purpose, Create a VIEW that finds the names and Phone numbers of all of Robert's patients.
3. Create a view which has First Names, Last Names of all doctors who gave out prescriptions for Vicodin.
4. Create a view which shows the First Name and Last name of all doctors and their specialty's.
5. Modify the view created in Q4 to show the First Name and Last name of all doctors and their specialties ALSO include doctors who DO NOT have any specialty.

6. Create trigger on the DoctorSpeciality so that every time a doctor specialty is updated or added, a new entry is made in the audit table. The audit table will have the following (Hint-The trigger will be on DoctorSpecialty table).

    a. Doctor's FirstName
    b. Action(indicate update or added)
    c. Specialty
    d. Date of modification

7. Create a script to do the following (Write the script for this)
    a. If first time backup take backup of all the tables
    b. If not the first time remove the previous backup tables and take new backups.

Extra credit:

c. Create a stored procedure that gives Prescription name and the number of patients from the city of Fullerton with that prescription.

Example

| 20 | Aspirin |
| 2 | Ciprofloxacin |

Part 1 Submission: ER Diagram: April 15 (11:59pm) **[20 points]**
INDIVIDUAL Submission
No group submission allowed for Part 1

Show all required:
- Tables
- Primary keys
- Foreign keys: Specifically write Foreign keys in a separate table or show through schema diagram]
- Relationships

Part 2 Submission: Everything else: May 7 (11:59pm) **[80 points]**

Included but not limited to:
- Create a database and populate it.
- Follow standard documentation and programming ethics.
- One demo and presentation per group
- Final Project report with description and screenshots of all the different functionality of the application.
    - Final report: One per group
    - Report should be a pdf format

- Provide your .sql files for database creation
- Provide all your script files and all other files associated with the application
- One review form per member - for providing other group members feedback.