Database Systems

Project: Design, development and implementation of a relational database

<u>Due:</u> Deadline date @ 11:59 PM EST

A relational database should be developed following the subsequent steps for the case study described in this PDF. All questions must be answered in the order described here; answers should be preceded by the question number. Answers must be typed-in and diagrams must be created using the appropriate software, **no handwritten** parts will be accepted.

1. Develop a conceptual data model reflecting the following requirements: (11/05/24)
    a. Identify the main entity types.
    b. Identify the main relationship types between the entity types identified in "a".
    c. Determine the multiplicity constraints for each relationship identified in "b".
    d. Identify attributes and associate them with entity or relationship types.
    e. Determine candidate and primary key attributes for each (strong) entity type.
    f. Generate the E-R diagram for the conceptual level (no FKs as attributes).
2. Develop a logical data model based on the following requirements: (11/21/24)
    a. Derive relations from the conceptual model.
    b. Validate the logical model using normalization to 3NF.
    c. Validate the logical model against 5 user transactions. (Note: These will be then implemented in 3c).
    d. Define integrity constraints:
        i. Primary key constraints.
        ii. Referential integrity/Foreign key constraints.
        iii. Alternate key constraints (if any).
        iv. Required data.
        v. Attribute domain constraints.
        vi. General constraints (if any).
    e. Generate the E-R diagram for the logical level (contains FKs as attributes).
3. Translate the logical data model for the Oracle Enterprise DBMS. (12/07/24)
    a. Develop SQL code to create the entire database schema, reflecting the constraints identified in previous steps.
    b. Create at least 5 tuples for each relation in your database.
    c. Develop the 5 SQL queries that correspond to 2c using embedded SQL.
    d. Upload all the code and documentation to GitHub.

Notes: A *report* will be created for *each deadline* including detailed documentation of each of the steps addressing all the required items. ER diagrams for the conceptual and logical models *must* be included in the corresponding reports. All assumptions made in the design must be clearly stated. Screenshots of the contents of the database created for each part of step 3 *must* be included in the report and the corresponding queries to show the table contents *must* be included in the code.

GitHub: The code generated during step 3 (SQL statements + program) must be uploaded to a GitHub repository. The link to the repository must be provided in the last report. The GitHub repository must also include all the documentation (i.e., reports) generated in the three steps.

Case Study: Pawsome Pets

A company called Pawsome Pets runs multiple clinics. The company would like for their data to be stored in a database. The following description was obtained during the analysis phase:

"Each of the Pawsome Pets clinics has several staff members and a member of staff manages at most one clinic (not all staff manage clinics). Each clinic has a unique clinic number (clinicNo) and each member of staff has a unique staff number (staffNo). Additionally, the company would like to store each clinic's name, address and telephone number, as well as the staff's name, address, telephone number, DOB, position and salary.

When a pet owner contacts a clinic, the owner's pet is registered with the clinic. An owner can own one or more pets, but a pet can only be registered at one clinic. Each owner has a unique owner number (ownerNo), a name, an address and a telephone number. Each pet has a unique pet number (petNo), name, DOB, animal species, breed and color.

When the pet comes to the clinic, it undergoes an examination by a member of the consulting staff. The database should store the following information for each examination: chief complaint (i.e., the main cause for the visit), description (i.e., what was done during the examination), date seen and actions taken (e.g., a treatment was prescribed, tests were ordered). A unique examination number (examNo) is assigned to each examination."