CSC423 Database Systems

Project: Design, development and implementation of a relational database

Due: Deadline date @ 11:59 PM EST

A relational database should be developed following the subsequent steps for the case study described in this PDF:

1. Develop a conceptual data model reflecting the following requirements: (11/02/21)
    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: (11/19/21)
    a. Derive relations from the conceptual model.
    b. Validate the logical model using normalization to 3NF.
    c. Validate the logical model against user transactions.
    d. Define integrity constraints:
        i. Primary key constraints.
        ii. Referential integrity/Foreign key constraints.
        iii. Alternate key constraints (if any).
        iv. 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. (12/09/21)
    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 5 SQL queries using embedded SQL (see Python tutorial).
    d. Upload all the code and documentation to GitHub.

a, b, d : 1.4
c, e : 1.85
f: 2.1

Reports: A report will be created for *each deadline* including detailed documentation of each of the steps, the results obtained at each stage, test data, sample output and conclusion. For example, the ER diagrams for the conceptual and logical models must be included in

each of the reports. All assumptions made in the design must be clearly stated. Screenshots of the contents of the database created in step 3 must be included in the report.
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: Redwood University

A university has requested the design and implementation of a database to store its data. The university encompasses multiple departments, each of which has a chair. The university does not want to store particular information regarding the chair, rather information pertaining to the department name and chair name, as well as the number of faculty members the department has. Department names must always start with *Department*.

The university has numerous students and each of them has declared at least one major. Additionally, the name and initials of a student are stored. Initials must be more than one character long. For each major, the university wants to store the major name, the department it is associated with, and a code. For example, 'Biology' is associated with department 3 (i.e., the Department of Biology) and has the code 'BIO'. Major codes must be three characters. Majors can be declared by one or more students. A major references one department, however a department offers one or more majors.

Each department has the possibility of hosting events, and an event can be (collaboratively) hosted by one or more departments. In addition to the event name, the university would like to store the start and end dates of the event. As it is logical, an event cannot end before the start date. Information pertaining to events are stored ahead of time, therefore at the time of insertion an event cannot be a past date or the current date. Students must attend one or more events, and each event will comprise one or more students.