Cairo University
Faculty of Computers and Artificial Intelligence
Information Systems Department
Database Systems -1

Database Systems-1 Project

Project Instructions:

1. This is a team project of at least 5 members and 6 members at most.
2. The members should be from the same group/lab or with the same lab TA (Note: if you from “ALL” group, then all team members should be from “ALL”).
3. Each team should select one idea from the proposed systems shown below to work on it.
4. Fill the project template with the following information.
    - Team members names and IDs.
    - Your section number.
    - Your lab TA name.
    - Title of the project that you have selected from the proposed below.
5. You are required to implement the proposed functionalities in the assigned project or propose your own.
6. All team members must participate and contribute to the project.
7. The final project deliverables will be uploaded on the classroom and discussed with your TA.

1. Project Deliverables

Phase 1: The system entity relationship diagram
• Students are required to construct the system’s ERD (conceptual model) • ERD should include at least 5 entities including at least 1 many-to-many relationship. Including Weak Entity (1 bonus) • Primary keys, foreign keys, and relationships should be clearly defined in the ERD. • Review the system's entity relationship diagram with your TA (make updates if needed)

Phase 2: Software Application + Conceptual ERD + Corresponding Physical ERD +Implemented database on MS SQL Server

- Convert the conceptual ERD into a physical ERD.
- Submit both conceptual and physical ERD.
- Deliver the SQL Server database generated from the physical ERD with populated data.
- Implement an application program using C# (or other permitted languages) that includes at least:
  - 2 insert statements on 2 different tables.
  - 2 delete statements on 2 different tables (with conditions).
  - 2 update statements on 2 different tables (with condition).
  - Select data from any table(s) of the database.
  - Select data that involves more than one table of the database (using joins).
  - Generate 1 meaningful report (bonus).
  - Implement GUI (bonus).

2. Proposed systems (choose only one system to implement)

1- University Library Management (Ex. http://www-sul.stanford.edu).

- **Proposed functionalities:**
  - Signing up a new user (e.g. admin, student)
  - Updating a user's details.
  - Adding a book (by admin)
  - Updating a book details (by admin)
  - Browsing books (by admin and students)
  - Showing a list of books that satisfy certain criteria (e.g., ISBN, Publication year, author…)

2- Flight reservation system (Ex. http://www.egyptair.com)

- **Proposed functionalities:**
  - Signing up a new user (e.g. admin, customer)
  - Updating a user's details.
  - Adding an aircraft (by admin)
  - Updating an aircraft details (by admin)
  - Adding a flight (by admin)
  - Updating a flight details (by admin)
  - Showing a list of available flights that satisfy certain criteria (e.g. date, source, destination, required number of seats…)
  - Performing operations on flights: booking, cancelling, changing flight class.)

3- Train booking (Ex. http://www.amtrak.com)

- **Proposed functionalities:**
  - Signing up a new user (e.g. admin, customer)
  - Updating a user's details.
  - Adding a train (by admin)
  - Updating a train details (by admin)
  - Adding a trip (by admin)
  - Updating a trip details (by admin)
  - Showing a list of available seats that satisfy certain criteria (e.g. date, time, source, destination, required number of seats…)
  - Performing operations on trips: Booking, and canceling.

4- Bank System

- There are multiple banks and each bank has many branches. Each bank has a name, code and address. Each branch has an address and a branch number.
- Each branch has multiple customers. Each customer has an SSN, name, phone, and address.

  Some customers may take different types of loans from these bank branches. Each branch offers multiple loans. Loan number, loan type, and loan amountmust be kept for each loan.
- One customer can have multiple accounts. For each account; account number, balance and type must be kept for it.
- **Proposed functionalities:**
  - Signing up a new user (e.g. customer, employee)
  - Updating a user details
  - Add bank (by admin)
  - Add bank branch (by Admin)
  - Add a customer (by employee)
  - Showing a list of loan (e.g. industry loan, commercial loan, Personal loan ...)
  - Showing a list of customers

- Showing a list of loans with customer name and employee name
- Performing operations on loans: request and start operation (by customer)
- Performing operations on loans: accept (reject, pay loan (by employee)

5- Hospital Management System

- **Proposed Functionalities:**
  - Signing up a new user (e.g., admin, doctor, nurse, patient)
  - Updating a user's details.
  - Adding a patient (by admin)
  - Add medicine details and assign a medicine for patients (by doctor)
  - Adding new appointment for a patient (by admin or patient)
  - Cancel patient appointment.
  - Showing a list of all appointments for a specific doctor
  - Showing a list of appointments for a specific doctor that satisfy certain criteria (e.g. date, specific patient, …)
  - Showing all medical history for a specific patient

6- Restaurant Ordering and Management

- **Proposed Functionalities:**
  - Signing up a new user (e.g., admin, manager, waiter/waitress, customer)
  - Updating a user's details.
  - Adding and updating the menu item (by admin/manager)
  - Table reservation system (by customer or staff)
    - Add table reservation
    - Cancel table reservation
    - Update table reservation (number of people, reservation date/time, …)
  - Order placement with items from menu (by customer or waiter/waitress)
  - Display order details
  - Showing a list of all orders and total payments (for specific day / week)
  - Showing a list of all reservation filtered by (table, date/time, or both)
  - Performing operations on table reservation: (accept, reject table reservation)

7- Faculty Management System

- **Proposed Functionalities:**
  - Signing up a new user (e.g., admin, staff, student)
  - Updating a user's details.
  - Creating courses (by admin)
  - Enrolling students into a course (by admin)
  - Add departments (by admin)
  - Assign students to departments (major)
  - Showing all students that satisfy one or more criteria (e.g. graduation year, major, course, GPA, …)
  - Showing total number of students per course/department
  - Showing all courses
  - Showing a student details.

8- Inventory Management System

- **Proposed Functionalities:**
  - Signing up a new user (e.g., admin, warehouse manager, sales representative, customer)
  - Updating a user's details.
  - Adding new products (by admin/warehouse manager)
  - Updating product details (by admin/warehouse manager)
  - Adding/ updating a category details
  - Assign product to a specific category
  - Placing or cancel an order (by customer)
  - Showing an order detail (e.g. order items, date/time, delivery date)

9- Event Ticketing System

- **Proposed Functionalities:**
  - Signing up a new user (e.g., admin, event organizer, customer)
  - Creating new events (by admin/event organizer)
  - Updating a user's details.
  - Add event sponsor.
  - Show all sponsors for specific event.
  - Setting ticket types and pricing (by admin/event organizer)
  - Searching for events by criteria (date, location, type)
  - Purchasing or return a ticket (by customer)