CS/DSA 4513-001 - Fall 2020 - Dr. Le Gruenwald
INDIVIDUAL PROJECT: A PATIENT ASSISTANT NETWORK DATABASE SYSTEM
Assigned: 10/12/2020; Due: 3:30 PM 11/23/2020 on Canvas
(Late submission will be accepted until 11:59 PM 11/24/2020 with 5% penalty; late submission after this time will NOT be accepted)

I. DATABASE APPLICATION REQUIREMENTS

Overview

The Patient Assistance Network (PAN) is a non-profit organization that provides support and care for patients. PAN needs to implement a database system to keep track of the personnel necessary to support the organization. In this project, your task will be to design and implement this database system. The information that needs to be stored in the database is described in the next section.

System Requirements

There are many categories of people that need to be tracked in the PAN database. Each person may fall into more than one of the following categories: clients, volunteers, employees, and donors. PAN tracks the name, social security number, birth date, race, gender, and profession of each person. In addition, PAN stores the contact information for each person consisting of a mailing address, email address, and home, work, and cell phone number. PAN also sends a monthly newsletter to people on its mailing list, so the database should indicate whether or not each person in the database is on that list. Finally, the system should have the ability to store a list of emergency contacts for the people in the database. This information should record the name and contact information for each of the emergency contacts along with their respective relationship to the person in the database.

PAN tracks its list of clients in the database. For each client, PAN tracks the name and phone number of his or her doctor and attorney. PAN also tracks the date the client was first assigned to the organization. Each client has a list of needs. Examples of these needs include visiting, shopping, housekeeping, transportation, yard work, and food. Each of these needs is also associated with a value indicating its importance to the client (1-10). PAN also tracks the list of insurance policies that each client has. Each insurance policy has a unique policy id, a provider id, provider address, and a type such as life, health, home, or auto.

PAN provides care for each client using teams that contain many volunteers. Each team cares for several clients, and more than one team may care for a client. Each team is identified by its name, and each team also has a type and a date it was formed. A volunteer may serve on multiple teams. For each volunteer, the database should store the date he or she first joined PAN and the date and location of his or her most recent training course. In addition, PAN should record the number of hours a volunteer worked each month for a particular team. Note that the volunteers do not work the same number of hours each month. One of the volunteers on a team serves as the team leader. This information should be tracked in the database as well. In addition, volunteers and clients may switch teams; so, the database system should provide the ability to mark whether or not each volunteer and client is active or inactive on a specific team.

Every team reports periodically to one PAN employee to discuss its current status, and more than one team may report to the same employee. The database should record the date of each report as well as a description of its content. For each employee, the database should store the employee's salary, marital status, and hire date. An employee may charge several expenses each month. The database should track the date of the expense, along with the amount and its description.

PAN depends on support from its donors. The database should track these people as well as record each of their donations. This information should include the date, amount, and type of donation along with the name of the fund raising campaign that generated the donation if it is

applicable. If the donation was made by check, the database should record the check number. If the donation was made by credit card, the database should record the card number, card type, and expiration date. In addition, each donor may wish to remain anonymous, so the database should record that information as well.

Each person in the database may be affiliated with an external organization. The database should track this information as well. Each organization should have a unique name, mailing address, phone number, and contact person. In addition, each organization may sponsor one or more PAN teams, and a team may have more than one sponsor. If the organization is a business, the database should record the business type, size, and company web site. If the organization is a church, the database should record its religious affiliation. Each organization may also make several donations to PAN, and the database should track the same donation information as it does for individual donors. This includes the ability for the organization to make anonymous donations.

Queries and Their Frequencies for the PAN Database System

1. Enter a new team into the database (1/month).
2. Enter a new client into the database and associate him or her with one or more teams (1/week).
3. Enter a new volunteer into the database and associate him or her with one or more teams (2/month).
4. Enter the number of hours a volunteer worked this month for a particular team (30/month).
5. Enter a new employee into the database and associate him or her with one or more teams (1/year).
6. Enter an expense charged by an employee (1/day).
7. Enter a new organization and associate it to one or more PAN teams (2/week).
8. Enter a new donor and associate him or her with several donations (1/day).
9. Enter a new organization and associate it with several donations (1/day).
10. Retrieve the name and phone number of the doctor of a particular client (1/week).
11. Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be sorted by the total amount of expenses (1/month).
12. Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
13. Retrieve the names and contact information of the clients that are supported by teams sponsored by an organization whose name starts with a letter between B and K. The client list should be sorted by name (1/week).
14. Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the total amount of the donations, and indicate if each donor wishes to remain anonymous (1/week).
15. Retrieve the names of all teams that were founded after a particular date (1/month).
16. Increase the salary by 10% of all employees to whom more than one team must report. (1/year)
17. Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5 (4/year).

II. TASKS TO BE PERFORMED

Task 1. Design an ER diagram and a relational database to represent the PAN database defined in part I.
Task 2. Provide a Data Dictionary that lists the names, types, and sizes (in bytes) of all attributes and associated constraints for EACH table.
Task 3.
      3.1. Discuss choices of appropriate storage structures for each relational table assuming that all types of storage structures discussed in class (Lecture Topic 4) are available. For each table, identify the queries (from the list of the given queries) that access the table, the type of each of those queries (insertion, deletion, random search, or range

search), the search keys (if any) involved in each of those queries, the frequency of each of those queries, your choice of the file organization for the table, and your justifications. Use the following format to fill out your answers:

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization	Justifications

3.2. Discuss choices of storage structures for each relational table when implementing it in Azure SQL Database (if different from the previous choices specified in 3.1).

Task 4. Construct SQL statements to create tables and implement them on Azure SQL Database. All Create statements must include appropriate constraints as defined in Task 2. For each table, you must include SQL statements that create the same storage structure as the one you selected for Azure SQL Database implementation in Task 3.2 (e.g., if you have decided that a table X must have an index on attribute Y, then you must include an SQL statement to create an index on attribute Y for table X).

Task 5. Write SQL statements for all queries (1-17) defined in part I. Write a Java application program that uses JDBC and Azure SQL Database to implement all SQL queries (options 1-17), two additional queries for import and export (options 18-19), and the "Quit" option (option 20) as specified in the menu given below. The program will stop execution only when the user chooses the "Quit" option; otherwise all options must be available for the user to choose at all times. Your program must be commented properly.

WELCOME TO THE PATIENT ASSISTANT NETWORK DATABASE SYSTEM

(1) Description of query 1
(2) Description of query 2
.
.
.
(17) Description of query 17
(18) Import: enter new teams from a data file until the file is empty (the user must be asked to enter the input file name).
(19) Export: Retrieve names and mailing addresses of all people on the mailing list and output them to a data file instead of screen (the user must be asked to enter the output file name).
(20) Quit

Task 6. Run the program created for Tasks 5 to test its correctness. To populate the database, perform 5 queries for each type (1-9) and show the contents of the affected tables after those queries are completed. To show database access is possible, perform 2 queries of each type (10-12) and 1 query for each type (13-17). To show the import and export facilities are available, run each option (18-19) once. To show the Quit option is available, run option (20) at least once. To demonstrate that Azure SQL can detect errors, you also need to perform 3 queries of different types that contain some errors.

Important Notes for the Java + JDBC + Azure SQL Database Part (Tasks 5 and 6):
Data manipulation and error checking **must be done** by Azure SQL Database. Your program is only to create the menu, accept choices, form queries, submit them to Azure SQL Database for execution, and display results or error messages.

Task 7. Write a Web database application using Azure SQL Database and JSP (or PHP if you are familiar with it) which provides the Web pages for query 1 and query 15. Since both queries take the input data from the user, there should be two Web pages for each query as follows: for query 1, one Web page to allow the user to enter the input data and one to display a message confirming the successful execution of the insertion; and for query 15, there should be one Web page to allow the user to enter the input data and one to display the retrieval results with appropriate headings. To show that your Web application works correctly, run the Web application so that queries 1 and 15 will be executed in this order: first query 15, then query 1, and then query 15 again, making sure that the results of query 1 will change the results of query 15 that follow query 1.

III. SUBMISSION

1. The project report must be typed (diagrams must be computer-generated).
2. The project report must be paginated.
3. The project report must include a cover page, which contains the following information: course name and number, section number, semester and year, instructor's name, author's name, id, and email address, and title of the project.
4. The project report must include a table of contents using the format specified in Section IV.
5. Appropriate section headings must be included to show your solutions for the corresponding tasks/subtasks.
6. In-line comments must be provided for the Java and JSP/PHP programs.
7. Submit the following files (DO NOT SUBMIT ZIP FILES) on Canvas: 1) one single PDF file containing the entire project report with the contents as specified in Section IV (with the file name as Your Last Name_Your First Name_IP_REPORT); 2) one SQL file (extension .sql) containing the SQL statements to create tables as explained in Task 4 in Section II (with the file name as Your Last Name_Your First Name_IP_Task4); 3) one SQL file (extension .sql) containing the SQL statements (and Transact SQL Stored Procedures, if any) to implement all queries (1-19 and error checking) for Task 5 in Section II ((with the file name as Your Last Name_Your First Name_IP_Task5a); 4) one Java file (extension .java) containing the Java source program for Task 5 in Section II (with the file name as Your Last Name_Your First Name_IP_Task5b); and 5) Java, JSP (extension .jsp) and HTML files (if any, extension .html), or PHP (extension .php) and HTML files (if any) which comprise your Web application for Task 7 in Section II (just the files with the source code you created/edited, do not try to retain their directory structure used during the running of the application) (each of the files in item (4) must have a file name starting with Your Last Name_Your First Name_IP_Task7, for example, Smith_Joe_IP_Task7.jsp)
8. The project is due on Canvas at 3:30 PM Monday, November 23, 2020. Late projects will be accepted until 11:59 PM Tuesday, November 24, 2020 with 5% penalty.
9. The overall project is worth 250 points (25% of your course grade); the late submission penalty is 12.5 points

IV. REQUIRED FORMAT FOR THE TABLE OF CONTENTS OF THE PROJECT REPORT

For each task, indicate the pages in which it appears. For each subtask, indicate its start and end page numbers. Below is an example of the table of contents.

V. ACADEMIC MISCONDUCT

This is an INDIVIDUAL PROJECT. You must do this project by yourself. You must not collaborate with your classmates or anyone else on this project. You must not receive help from any person or provide help to any person on this project. You must not share the project description/requirements or your project answers with anyone. A violation of any of these rules will be considered academic misconduct and will result in action as specified in the Academic Integrity Code at OU: http://integrity.ou.edu/files/Academic_Misconduct_Code.pdf. Review the Fall 2020 CS/DSA 4513-001 course syllabus for more information concerning academic misconduct. If you have questions about this project, see your TAs or Dr. Gruenwald on Zoom during their office hours.