

CS/DSA 4513-001 - Fall 2024 - Dr. Le Gruenwald
INDIVIDUAL PROJECT: A PATIENT ASSISTANT NETWORK DATABASE SYSTEM
(Maximum Points: 300 Points = 30% of the course grade)
Assigned: 9/19/2024
Due: 5:00 PM, 11/14/2024, on Canvas

- **Start on this project early so that you will have enough time to consult us if you have questions.**
- **Late submission will be accepted until 11:59 PM, 11/15/2024, with 5% penalty; late submission after this time will not be graded.**
- **ACADEMIC INTEGRITY: 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. Asking generative AI tools (including ChatGPT, Bard, Bing Chat, and other AI writing and coding assistants) to solve any task in this project for you is not allowed. Violations of any of these rules will be considered academic misconduct and will result in action as specified in the Academic Integrity Code at The University of Oklahoma: <http://www.ou.edu/integrity>. Consult also the following web page for a Student's Guide to Academic Integrity at The University of Oklahoma: <http://www.ou.edu/integrity/students>.**

I. DESCRIPTION

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 this section.

There are many categories of people that need to be tracked in the PAN database. Each person can fall into one or more than one of the following categories: clients, volunteers, employees, and donors. PAN tracks the name, social security number, gender, and profession of each person. In addition, PAN stores the contact information for each person consisting of a mailing address, email address, and 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 phone number 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. 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 name, 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.

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 donor and associate him or her with several donations (1/day).
8. Retrieve the name and phone number of the doctor of a particular client (1/week).
9. 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).
10. Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11. Retrieve the names of all teams that were founded after a particular date (1/month).
12. Retrieve the names, social security numbers, contact information, and emergency contact information of all people in the database (1/week).
13. 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).
14. Increase the salary by 10% of all employees to whom more than one team must report. (1/year)
15. 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. (75 points): Design an ER diagram to represent the PAN database defined in Section I.

Task 2. (30 Points): Convert the ER diagram in Task 1 to a Relational Database (i.e. a set of relational schemas).

Task 3. (40 points):

- 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 detailed 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). **Part of this task is for you to find and study the relevant documentation on your own.**

Task 4. (25 points): 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. (Task 5 and Task 6 together = 130 points): Write SQL statements for all queries (1-15) defined in part I. Write a Java application program that uses JDBC and Azure SQL Database to implement all SQL queries (options 1-15), two additional queries for import and export (options 16-17), and the “Quit” option (option 18) as specified in the menu given below. You are free to pick any file format you wish to use for file import and export options. 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

.

.

.

(15) Description of query 15

(16) Import: enter new teams from a data file until the file is empty (the user must be asked to enter the input file name).

(17) 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).

(18) Quit

Task 6. Run the program created for Task 5 to test its correctness. To populate the database, perform 5 queries for each type (1-7) 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 (8-11) and 1 query for each type (12-15). To show the import and export facilities are available, run each option (16-17) once. To show the Quit option is available, run option (18) 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 4 and 5):

Data manipulation and error checking **must be done** by Azure SQL Database. Your Java 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.

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, your 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 program.
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 you choose to use any**) to implement all queries (1-14 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)
8. The project is due on Canvas at 5:00 PM, Thursday, November 14, 2024. Late projects will be accepted until 11:59 PM, Friday, November 15, 2024 with 5% penalty. Late submission after this time will not be graded.
9. The overall project is worth 300 points (30% of your course grade); the late submission penalty is 15 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.

Tasks Performed	Page Number
Task 1. ER Diagram	1
Task 2. Relational Database Schemas	2-3
Task 3.	4-20
3.1. Discussion of storage structures for tables	12-14
3.2. Discussion of storage structures for tables (Azure SQL Database)	15-20
Task 4. SQL statements and screenshots showing the creation of tables in Azure SQL Database	21-50
Task 5.	51-60
5,1 SQL statements (and Transact SQL stored procedures, if any) Implementing all queries (1-15 and error checking)	51-54
5,2 The Java source program and screenshots showing its successful compilation	55-60
Task 6. Java program Execution	61-90
6.1. Screenshots showing the testing of query 1	61-63
6.2. Screenshots showing the testing of query 2	63-65
.	
.	
.	
6.15. Screenshots showing the testing of query 15	80-81
6.16. Screenshots showing the testing of the import and export options	82-84
6.17. Screenshots showing the testing of three types of errors	85-88
6.18. Screenshots showing the testing of the quit option	89-90

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. Asking generative AI tools (including ChatGPT, Bard, Bing Chat, and other AI writing and coding assistants) to solve any tasks in this project for you is not allowed. 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 The University of Oklahoma: <http://www.ou.edu/integrity>. Consult also the following web page for a Student's Guide to Academic Integrity at The University of Oklahoma: <http://www.ou.edu/integrity/students>. Review the Fall 2024 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.