Aalto University
School of Science

Aalto University
Computer Science Department

CS-A1155: Databases for Data Science 2024

COURSE PROJECT

Designing a Volunteer Matching System (VMS) with the Finnish Red Cross (FRC)

Responsible Teacher
Nitin Sawhney
Professor of Practice, Department of Computer Science, Aalto University

Teaching Assistants
Amina Chahla
Duy To
Fathima Afrooz
Krzysztof Modrzyński
Minh Ha Le
Rasmus Innanen

Table of Contents

1. Introduction

Databases for Data Science aims to equip the students with the know-how to design and implement relational databases that effectively model real-world scenarios and in accordance with normalization rules. Moreover, the course content focuses on teaching students how to use Unified Modeling Language (UML) to design relational schema and use SQL to conduct queries so as to efficiently capture and extract the desired data for modeling and analyzing crucial aspects of the problem domain.

A course project conducted in teams is essential as it prepares students to apply the principles of relational databases to projects in data science. In spring 2024, the course project is being devised in collaboration with the Finnish Red Cross (FRC) and this project definition is based in part on our deliberations with their experts.

This document provides a description of the topic and problem domain, a brief introduction to the work of the Finnish Red Cross, and detailed information on the mandatory and optional parts of the project.

1.1 Project Description

As per the FRC proposal, the topic for the project is a Volunteer Matching System (VMS). The student team’s role is to create a database and define usage to support the matching of Red Cross Volunteer Capacity (supply) with “Local Multidimensional Vulnerabilities and Crises” (demand).

1.2 Grading Requirements

The project contains two mandatory parts and a final package deliverable that includes an optional aspect (for bonus points). The former consist of creating a UML diagram, designing a relational model, implementing SQL, and a simple data cleaning and analysis. The optional part covers more advanced topics, such as predictive analytics using machine learning. These topics are targeted towards students with advanced background or motivation to extend the project further.

Project submission is done in three phases, including parts 1 and 2, and a final deliverable package that should include revised versions of part 1 and 2 (based on feedback) and any optional contributions to the project.

2. Finnish Red Cross

This section was extracted from the lecture slides provided by the Finnish Red Cross (FRC) during the lecture of 23.04.2024 and includes information presented by Aleksi Aalto and Matias Andersson from FRC.

"The Finnish Red Cross is one of the largest nongovernmental organisations in Finland. The operations of the Red Cross and Red Crescent Movement are steered by seven fundamental principles: humanity, impartiality, neutrality, independence, voluntary service, universality and unity. These principles also steer the cooperation between the Finnish Red Cross and other organizations."

A link to the presentation slides and further information on the work of the organization can be found here.

3. Matching Volunteers: Supply

This section covers the supply side of the volunteer matching system. It includes a detailed description, based on which the students are required to complete part 1 of the project. To comply with the learning goals of the course and taking into consideration prior knowledge of the students, part 1 was devised such that students are provided with the necessary information to create a simple relational data model. Thus, this part took inspiration from the information provided by Finnish Red Cross and simplified it to a manageable database schema for students.

3.1 Part 1: UML and Relational Model

3.1.1 Description

The two main actors of the system are volunteers and beneficiaries:

Beneficiaries have unique IDs, names, and addresses. Any beneficiary can make as many volunteering requests as they need. Requests should include a unique ID, the ID of the beneficiary it was sent by, the number of volunteers needed, a priority value to indicate how urgent the request is (from 0 to 5, with 5 being highest priority), the area of interest where the request lies, a start date, an end date, and a register by date. Requests also have criteria for acceptance and application validity. These criteria comprise of request skills and request locations. Each request skill has a request ID and a skill name and each request location has a request ID and a city ID.

Volunteers on the other hand have unique ID, name, birthdate, email, address, readiness to travel (minutes). A volunteer can choose any combination of areas of interests from the following: CollectDonations, FirstAid, FoodHelp, GuideAndTeach, HelpInCrisis, ImmigrantSupport, OrganiseActivities, PromoteWellbeing, WorkInATeam, WorkInMulticulturalEnvironment, WorkWithElderly, WorkWithYoung. Volunteers can also choose any combination of skills from the following: CommunicationAndMarketing, CookingAndBaking, DigitalCompetence, EventHosting, EventOrganizaton, FinanceAndAccounting, HealthCareOrFirstAid, MeetingPeople, Organizational, PhotographyAndVideo, PublicPerformances, Rescue, TeamGuide, TrainPeople.

Each of these skills have a unique name and a description. Beneficiaries can appraise skills by assigning the required skills for each request: a minimum need (minimum number of volunteers who possess this skill), and a value to indicate importance (from 0 to 5, with 5 being highest priority).

Volunteers can sign up to the system, browse through the volunteering requests, and send up to 20 applications where they apply to the requests. Applications should include a unique ID, the ID of the request it was made to, the ID of the volunteer it was sent by, the time it was modified, and they should indicate whether they are valid or not.

Volunteers operate in ranges. Each volunteer range has a volunteer ID and a city ID. Each city has an ID, a name, and a geolocation.

3.1.2 Requirements:

Requirements for part 1 are the same as in previous years and the student is asked to:

- Draw a UML diagram for the Volunteer Matching System (VMS) database based on the information defined in this document using the notations taught in the course.
- Convert the UML diagram to the relational data model, present the schemas of the relations, and underline the attributes which form the key for each relation.

- Provide answers to the following questions: What are the non-trivial functional dependencies of the database? Are there any forms of redundancy or other anomalies in the database structure? Is the database in the Boyce-Codd Normal Form? If it is not, use the decomposition algorithm (submit both original and decomposed version)

3.2 Part 2: SQL Implementation

3.2.1 Synthetic Data and Queries

In part 2, the students will be provided with synthetic data. The students will also be tasked with creating queries and answering questions. There are a total of 22 questions for each project. For "free choice" questions, grading will depend on correctness, complexity, and relevancy.

A. **BASIC [8 + 4 free choices] = 2 + 3 x 11 = 35 (+ 10 if all are correct) = 45 points**
    1. (2p) For each request, include the starting date and the end date in the title.
    2. (3p) For each request, find volunteers whose skill assignments match the requesting skills. List these volunteers from those with the most matching skills to those with the least (even 0 matching skills). Only consider volunteers who applied to the request and have a valid application.
    3. (3p) For each request, show the missing number of volunteers needed per skill (minimum needed of that skill). Assume a volunteer fulfills the need for all the skills they possess.
    4. (3p) Sort requests and the beneficiaries who made them by the highest number of priority (request's priority value) and the closest 'register by date'.
    5. (3p) For each volunteer, list requests that are within their volunteer range and match at least 2 of their skills (also include requests that don't require any skills).
    6. (3p) For each volunteer, list all the requests where the title matches their area of interest and are still available to register.
    7. (3p) List the request ID and the volunteers who applied to them (name and email) but are not within the location range of the request. Order volunteers by readiness to travel.
    8. (3p) Order the skills overall (from all requests) in the most prioritized to least prioritized (average the importance value).

9-12. [free choice]: (3p/each) Construct 4 queries of your choice and explain why these are important for the VMS.

B. ADVANCED = 10 + 10 + 20 + 25 = 65 (+ 10 if all are correct) = 75 points

a) Views: [1 + 1 free choice] = 5 + 5 = 10 points

1. (5p) Create a view that lists next to each beneficiary the average number of volunteers that applied, the average age that applied, and the average number of volunteers they need across all of their requests.
2. [free choice] (5p) Create a view of your own choice and provide a reasoning of your choice.

b) Trigger and Functions: [2] = 5 + 5 = 10 points

1. (5p) "Finnish personal identity codes are issued by the Population Register Centre (DVV). They consist of a string of numbers that indicates the individual's date of birth, an individualized string, and a control character.

   Examples: 150600A905P

   - 150600 = Date of birth
   - A = the character in the middle
   - 905 = the individualized string
   - P = Control character

   The control character is either a number or a letter. The calculation formula is to divide the value of the nine-digit string made up by the date of birth and the individualized string by 31. Then the value of the division's remainder determines the control character: You must compare the control character you get with the control character of the personal ID you need to check."

   Source: How To Calculate The Control Character For Verifying The Authenticity Of Finnish Business Ids And Personal Ids

Create a check constraint for the volunteer table with a function that validates a volunteer ID when a new volunteer is inserted. The ID is valid if they satisfies:

- Length = 11 characters
- The 7th character (separator) is one of the following: +, -, A, B, C, D, E, F, X, Y, W, V, U
- The correct control character is used

2. (5p) Create a trigger that updates the number of volunteers for a request whenever the minimum need for any of its skill requirements is changed. The total number of volunteers needed for each request is calculated as the sum of unskilled volunteers (those without any skill requirements) and the minimum need for each required skill.

c) Transactions: [1 + 1 free choice] = 15 + 5 = 20 points

1. (15p) Create a transaction that will read valid applications for a request. Then assigns the applicants as such:
    - Prioritize the skills by their value of importance.
    - Assign volunteers with valid applications and who have these skills until the minimum number of volunteers needed for the skills is met (assigning here means is_accepted gets TRUE, you may also create a separate table volunteer_assignment that tracks request_id and volunteer_id who got assigned to the request) (you may use your scoring system for this)
    - Assign the rest of applied volunteers.
    - If the register by date is not past and the minimum number of volunteers is not met (skill based or general), roll back.
    - If the register by date is not past or the minimum number of volunteers is met, commit the assignment.
    - If the register by date is past and the minimum number of volunteers is not met, either add more time to the register by date or accept the volunteers.
2. [free choice] (5p) Create a transaction of your own choice and provide a reasoning of your choice.

d) Analysis [3 + 1 free choice] = 5 + 10 + 5 + 5 = 25 points

1. (5p) Visualize the number of volunteers available by city (according to their volunteer range, note: a volunteer can be available in more than 1 city) compared to the number of volunteers that applied for a request in that city. What's the city with the most (top 2) volunteers and the least (bottom 2)? Make sure your visualization gives a good overview of the current situation and quickly shows the most information.
2. (10p) Create your own scoring system to calculate the matching percentage from all the attributes of a volunteer that you find relevant: e.g: interest, travel readiness, volunteer range, number of skill matches, etc. Make a compelling case for your scoring scheme and suggest a top 5 candidates for each request according to this system. Does it match the candidates you have found in past questions?
3. (5p) For each month, what are the number of valid volunteer applications compared to the number of valid requests? What months have the most and least for each, how about the difference between the requests and volunteers for each month? Is there a general/seasonal trend? Is there any correlation between the time of the year and number of requests and volunteers?
4. [free choice] (5p) Choose a question/topic/problem/aspect you would like to analyze and visualize.

Total points: 45 + 75 = 120 points

3.2.2 Requirements

Requirements for part 2 are the same as in previous years and the student is asked to:

- Create a relational database in SQL based on the design from Project Part I
- Populate the database with the given data
- Create queries to answer the requested questions

The submission needs to contain:

- SQL file used to create tables in the database
- SQL file containing the required queries
- Python file used to create the database
- Python file used to do data cleaning, data analysis and data visualization (if applicable)

- Documentation about the design choices and the results for the required queries
- Presentation slides

4. Matching Volunteers: Demand

Although matching the volunteers on the supply side fits more with the scope of the course, the demand side can present intriguing and additional scope for development (as an optional part of this project). Examples of these include dynamic matching and predictive analytics. More advanced or highly motivated students can pick one of these proposals to discuss in order to get bonus points. Students are allowed to add their own synthetic data and adjust their database design as needed so long as they provide the datasets and all scripts used for grading purposes. Students are not provided with specific requirements, allowing them for more freedom, with the aim to create a diverse pool of project designs and implementations.

4.1 Proposal 1: Dynamic Matching

Students can focus on the case where there is an oversupply or undersupply of volunteers to a specific request or crisis. They are encouraged to come up with solutions on how to dynamically reassign the volunteers such that the matching of supply to demand will operate efficiently. Students are not limited to any tool and can use their expertise outside of the course to do so. A full working implementation is not necessary as long as solutions are discussed and the reasoning and process of problem solving is provided.

4.2 Proposal 2: Predictive Analytics

Students can focus on predictive analytics using machine learning. It is important to understand where vulnerabilities lie and to be able to predict demand. This will allow for a swift response from the FRC to the affected areas. Students are thus encouraged to discuss ways to utilize data and build models that can predict future trends. Students are not required to create machine learning models but if they wish to do so, they are allowed to. Students can provide a discussion on how to gather further data, such as social media scraping, that can help in creating accurate models to detect patterns and predict crises and how to change the database to accommodate for that.

5. Other Optional Additions

Students can also work on other optional aspects if they wish to. These may include but are not limited to visualization, user interfaces, application design, and web software design. Students are encouraged to discuss their plans with the teaching assistants, especially if they want to work on aspects not mentioned in this document.

6. Final Package Deliverable

Students are required to submit a final package deliverable, consisting of revised versions of part 1 and 2 (based on feedback) and any optional additions. All design choices and assumptions need to be explained. Students may add a discussion on the performance and the scalability of the database. They should also include their division of roles and responsibilities in the project, and reflect on how the group work was conducted, challenges encountered, and what you learned from this project along the way.

7. Acknowledgments

We would like to pay special thanks to Aleksi Aalto and Matias Andersson at the Finnish Red Cross (FRC) for their detailed guidance, their informative presentation, and their continuous help towards the development of the project brief.