A syringe and a bottle of liquid

Description automatically generated with medium confidence

Project report

Group 17

Table of Contents

[Introduction 2](#_Toc137819201)

[Design and use case 2](#_Toc137819202)

[Purpose of the database and use case 2](#_Toc137819203)

[UML diagram and relational schema 3](#_Toc137819204)

[First UML diagram 3](#_Toc137819205)

[Revised UML diagram 4](#_Toc137819206)

[First relation schema 4](#_Toc137819207)

[Revised relation schema 5](#_Toc137819208)

[SQL files and usage 5](#_Toc137819209)

[Assumptions 5](#_Toc137819210)

[Design approach 5](#_Toc137819211)

[Performance Analysis and improvements 5](#_Toc137819212)

[Possible improvement and changes 5](#_Toc137819213)

[Improvements effect on database efficiency and usability 5](#_Toc137819214)

[Common query time 5](#_Toc137819215)

[Working as a team 5](#_Toc137819216)

[Task division and roles 5](#_Toc137819217)

[Estimated time schedule for group project 5](#_Toc137819218)

[Problem solving as a group 5](#_Toc137819219)

[Conclusion 6](#_Toc137819220)

[Advantages and disadvantages in database 6](#_Toc137819221)

[What to be done in the future 6](#_Toc137819222)

[Overall evaluation 6](#_Toc137819223)

# Introduction

The project report covers 5 sections: Introduction, Design and use case, Performance analysis and improvements, Working as a team, Conclusion. In the Introduction part, we will introduce the context of the project and explain briefly about other sections.

In this project, we designed and created a database to keep track of different vaccine types, transportation of vaccine batches, treatment plans, and patient data.

PostgreSQL relational database management and Python language are used to create tables, manipulate data, interact with database, and perform data analysis.

# Design and use case

## Purpose of the database and use case

The purpose of designing the database is to provide an organized and efficient structure for storing, managing, and retrieving data. Databases are designed to handle large volumes of structured or unstructured data in a consistent and secure manner. They provide a centralized repository for storing data that can be accessed by multiple users or applications concurrently.

In our project, the purpose is to efficiently store data such as covid-19 vaccine, vaccine manufactures, transporting of vaccines in Finnish hospitals and clinics, vaccination events in Finland, patients and their symptoms, the staff of each hospital/clinic in Finland and their working schedules.

There are various use cases, including data storage and retrieval, data management, data sharing and collaboration, and data analytics. By storing data in database in a structured format, we enable efficient searching, filtering, and data sorting. We can also manage data such as adding, modifying, deleting while keeping data integrity and consistency. The use of database also facilitates data sharing and collaboration among multiple users or systems.

## UML diagram and relational schema

### First UML diagram

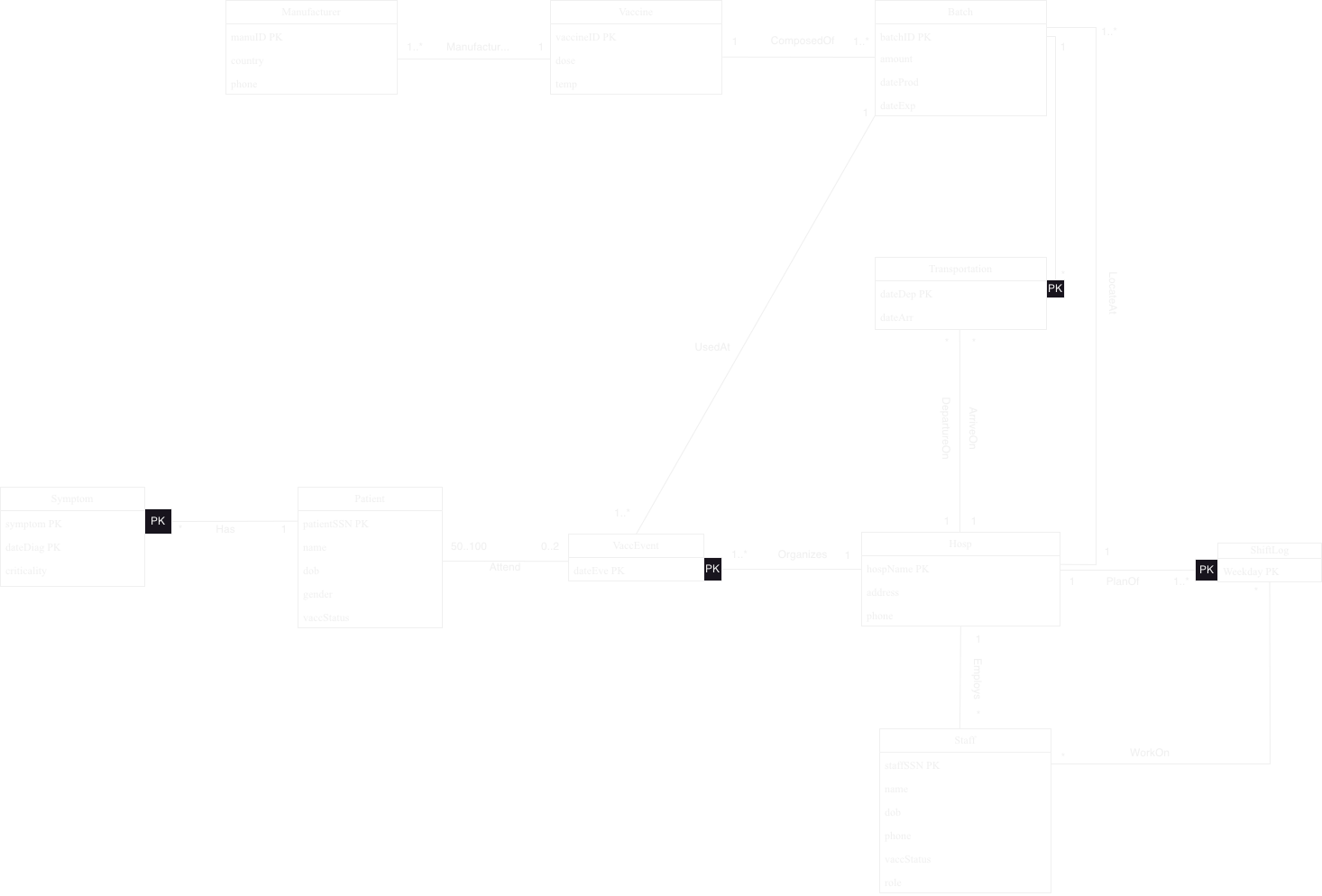


Figure 1first UML diagram

### Revised UML diagram

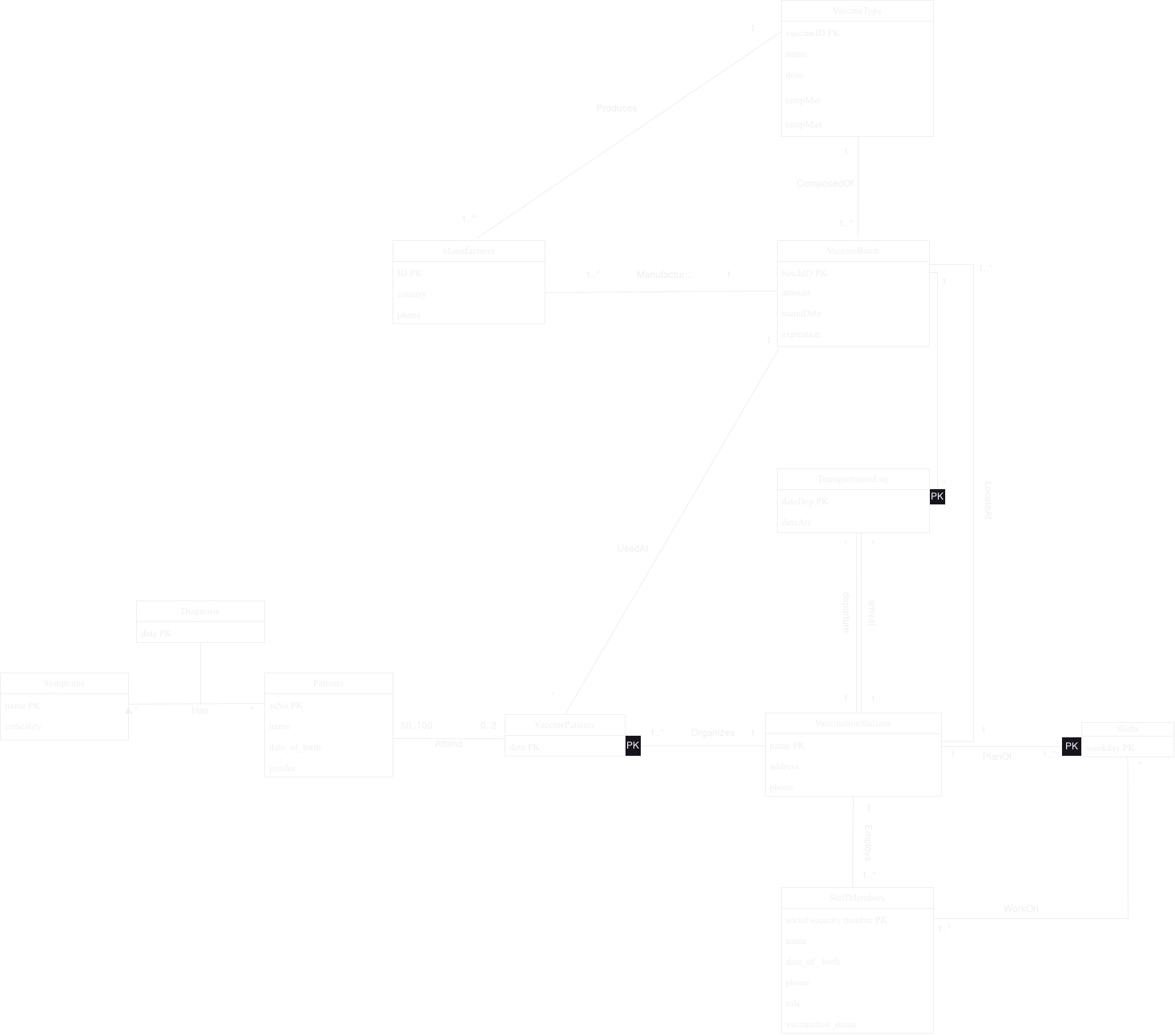


Figure revised UML diagram

### First relation schema

Manufacturer(manuID, country, phone, vaccineID)

Vaccine(vaccineID, dose, temperature)

Batch(batchID, amount, dateProd, dateExp, hospName , vaccineID)

Hosp(hospName, address, phone)

Transportation(batchID, dateDep, dateArr, hospDep, hospArr)

Staff(staffSSN, name, dob, phone, vaccStatus, role, hospName)

ShiftLog(hospName, weekday)

WorkOn(staffSSN, weekday, hospName)

VaccEvent(dateEve, hospName, batchID)

Patient(patientSSN, name, dob, gender, vaccStatus)

Attends(patientSSN, dateEve, hospName)

Symptom(symptom, patientSSN, dateDiag, criticality)

### Revised relation schema

Manufacturer(ID, country, phone, batchID)

VaccineType(vaccineID, name, dose, tempMin, tempMax)

VaccineBatch(batchID, amount, manuDate, expiration, name , vaccineID)

VaccinationStations(name, address, phone)

TransportationLog(batchID, dateDep, dateArr, nameDep, nameArr)

StaffMembers(social\_security\_number, name, date\_of\_birth, phone, role, vaccStatus, name)

Shifts(name, weekday)

WorkOn(social\_security\_number, weekday, name)

VaccinePatients(date, name, batchID)

Patients(ssNo, name, dob, gender)

Attends(ssNo, date, name)

Symptom(name, criticality)

Has( ssNo, name, date)

## SQL files and usage

In the project, PostgreSQL relational database management system is used to create, manage, and manipulate the database. SQL queries are performed in creating tables and retrieving data to perform data analysis.

All SQL files are stored in database folder under ‘cs-a-1155-2023-vaccine-distribution-group-17’ folder. There are all together 11 sql files. The sql file ‘create\_and\_file\_db\_psql’ is used to create tables, and the file ‘query1’ to ‘query7’ are the queries answering project part 2 questions. SQL file ‘vacc\_patient’, ‘freq1’, ‘ferq2’ are queries creating views. These 3 views should be created before running ‘query7’ as these views are the foundation of the analysis.

## 

## Assumptions

* Each manufacturer only works from 1 country.
* Each manufacturer can only make one type of vaccine.
* All phone numbers are unique, and no two individuals/manufacturers share a number.
* SSNs are unique.
* Phone numbers and SSNs cannot be NULL.
* Any transportation of vaccines must take at least 1 day.
* Vaccine batches can’t be split
  + every vaccine in a vaccine batch must be at the same hospital.
* All staffs must be working at exactly 1 establishment.
* All hospitals and clinics must organise at least 1 vaccination event.
* Hospitals and clinics must have at least 1 staff.
* Staff can work multiple days, and there must be at least one staff member working when

a vaccination shift is scheduled.

* Staff members are not required to work vaccination shifts (some staff members can work

on no vaccination shift at all).

* No patient attends more than 2 vaccination events.
* Each vaccine type must be used to create at least 1 batch.
* Each vaccine type must have at least 1 manufacturer.
* Each batch must contain only 1 type of vaccine.
* Each hospital and clinic must have at least 1 vaccinations shift plan.
* A patient can be diagnosed with the same illness multiple times on different dates.
* A vaccination event must be hosted by exactly 1 hospital/clinic.
* Multiple patients can have the same symptoms on the same date.
* A patient can be diagnosed with multiple symptoms on the same date.
* There are no entries in ShiftLog when a vaccination event is not taking place.
  + For example, if Central Hospital holds vaccination events every Monday and Thursday, the primary key (Central Hospital, Tuesday) does not exist.

## Design approach

When designing the database, we followed the following steps to ensure a well-structured and efficient database design.

We start by identifying the main entities presented in the database, such as vaccine batch, vaccine type, manufacturer, hospital and clinics, staff, patients etc.

Then we define attributes within each entity according to description. For example, for ‘Manufacturer’ entity, we defined attributes ‘ID’, ‘Country’, and ‘Phone’. We also identify primary key for each class, such as ‘ID’ as primary key for ‘Manufacturer’ class, marked as ‘PK’.

Then, we define the relations between entities. For example, one type of vaccine can be produced by many manufactures and one manufacturer should only manufactures one type of vaccine. We present these relations using lines with appropriate cardinality indicators, such as ‘1 to many’ or ‘many to many’.

# Performance Analysis and improvements

## Possible improvement and changes

## Improvements effect on database efficiency and usability

## Common query time

# Working as a team

## Task division and roles

## Estimated time schedule for group project

## Problem solving as a group

# Conclusion

## Advantages and disadvantages in database

## What to be done in the future

## Overall evaluation