Middle East Technical University Department of Computer Engineering

CENG 351

Data Management & File Structures

Fall 2021–2022

Programming Assignment 1

f

Due date: 7 Jan 2022, Friday, 23:59

1 Introduction

In this assignment, you are asked to create a system by designing queries and implementing predefined functions to operate on a database for a vaccination tracking service called CENGVAC. You will provide certain tasks and a well-defined interface.

All necessary data files, classes and the interface you will implement are provided to you in the source files. Do not confuse interface with graphical user interface (GUI). Interface is an abstract type in Java used to specify a behavior that classes must implement. (If you remember from the Programming Language Concepts course, a Java interface corresponds to a C++ class that all of its methods are pure virtual.)

The first thing you should do is implementing functions which create the necessary tables corresponding to the schema given in Section 3. Then, you should design queries to accomplish the given tasks. Lastly, you should implement the interface using the queries you have designed as they give the desired results when defined parameters are given. You will not implement the Evaluation class. It will be implemented by us to manipulate the database through the predefined interface and evaluate your implementations. Your task is to build up class(es) which implement(s) the provided interface.

2 Objectives

This assignment aims to help you get familiar with

- Connecting and querying to MySQL Server using Java Database Connectivity (JDBC)

3 Schema

You will use (strictly) the schema given below in the scope of this assignment.

User (userID:int, userName:varchar(30), age:int, address:varchar(150), password:varchar(30), status:varchar(15))
Vaccine (code:int, vaccinename:varchar(30), type:varchar(30))

Vaccination (code:int, userID:int, dose:int, vacdate:date) References Vaccine (code), User (userID)
AllergicSideEffect (effectcode:int, effectname:varchar(50))
Seen (effectcode:int, code:int, userID:int, date:date, degree:varchar(30)) References AllergicSideEffect (effectcode), Vaccination (code), User (userID)

Your task is to generate a class named CENGVACDB (it should belong to package ceng.ceng351.cengvacdb) which implements ICengVacDB interface. You can create any additional classes if necessary. CENGVACDB class should be able to accomplish the following tasks:

- Creating the database tables
- Inserting data into tables
- Find vaccines that have not been applied to any user
- List users who have been vaccinated for two doses since the given date
- List of the two recent vaccines applied to any user that do not contain 'vac' in their name
- List users who have had at least two doses of vaccine and have experienced at most one side effect
- List users who have been vaccinated with all vaccines that can cause the given side effect
- List users who have been vaccinated with at least two different types of vaccines within the given time interval
- List side effects experienced by users who have two doses of vaccine with less than 20 days between doses
- Calculate the average number of doses of vaccinated users over the age of 65
- Update his/her status to "eligible" after 120 days have passed since the user was vaccinated last time
- Given vaccine name, delete the vaccine(s) from the database
- Dropping the database tables

Tasks are explained in more detail below. For each task, there is a corresponding method in ICENGVACDB interface. You need to implement them in CENGVACDB class. Necessary data files (for populating the tables) to accomplish these tasks are provided.

In **data** folder there are 5 txt files that correspond to each table. You will use these tables when you are inserting data. Data files, interfaces and classes for fulfilling these tasks will be provided as source files. You can assume all information will be complete and consistent, i.e. all necessary data will be inserted before executing a query. You can find detailed description about the usage of the functions in provided source files. Your results should not include any repetition. Therefore, please do not forget to use DISTINCT keyword when appropriate in your queries.

Make sure you are using Java (version13) before starting.

3.1 Creating the database tables (10 pts)

You will create all the tables according to the schema described above.
You can assume that tables will be created before executing any other database operation.
Do not forget to define **foreign keys** while you are creating tables.
Also, you need to consider that whenever a vaccine is deleted, allergic side effect related to it should also be deleted.
Output: the number of tables that are created successfully.

3.2 Inserting data into tables (5 pts)

You will insert data into appropriate tables.
Output: the number of rows inserted successfully.

3.3 Find vaccines that have not been applied to any user (5 pts)

Output: code, vaccinename, type
Order the results by code in ascending order

3.4 List users who have been vaccinated for two doses since the given date (5 pts)

Input: vacdate
Output: userID, userName, address
Order the results by userID in ascending order

3.5 List of the two recent vaccines applied to any user that do not contain 'vac' in their name (10 pts)

Output: code, vaccinename, type
Order the results by code in ascending order

3.6 List users who have had at least two doses of vaccine and have experienced at most one side effect (10 pts)

Output: userID, userName, address
Order the results by userID in ascending order

3.7 List users who have been vaccinated with all vaccines that can cause the given side effect (10 pts)

Input: effectname
Output: userID, userName, address
Order the results by userID in ascending order

3.8 List users who have been vaccinated with at least two different types of vaccines within the given time interval (10 pts)

You should include dateStart and dateEnd in the result, it is a CLOSED interval.
Input: startDate, endDate
Output: userID, userName, address
Order the results by userID in ascending order

3.9 List side effects experienced by users who have two doses of vaccine with less than 20 days between doses (10 pts)

Output: effectcode, effectname
Order the results by effectcode in ascending order

3.10 Calculate the average number of doses of vaccinated users over the age of 65 (5 pts)

Output: the average result of applied doses for the given condition

3.11 Update his/her status to "eligible" after 120 days have passed since the user was vaccinated last time (10 pts)

Input: given date
Output: number of rows affected.

3.12 Given vaccine name, delete the vaccine(s) from the database (5 pts)

Note that related allergic side effects of the vaccine(s) should be deleted as well. You should handle that by design of your schema.
Input: vaccinename
Output: number of rows in the vaccine table after delete operation.

3.13 Dropping the database tables (5 pts)

You will drop all the tables (if they exist).
Output: number of tables that are dropped successfully.

4 Regulations

1. Programming Language: Java(Version13).

2. Database: An account on the MySQL server on a remote machine will be created for each of you and an e-mail including credentials and connection configuration will be sent to your metumail. You must use JDBC driver to connect to the database. Your final submission must connect to the MySQL server on the remote machine. So, make sure that the connection information is correct before submitting your homework.

3. Attachments: Necessary source files and JDBC driver is provided.
4. Input: All strings will be case-sensitive and they will not include any non-English characters. Note that they may include punctuation.
5. Cheating: We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.
6. Evaluation: It is GUARANTEED that input files are correctly formatted and sample data will be given to you. There will be no surprises about the data, similar (and larger) data will be used while evaluating homeworks. Your program will be evaluated automatically using "black-box" technique so make sure to obey the specifications. Please, be noticed that you have to accomplish tasks only within your sql queries not with any other Java programming facilities.

5 Submission

Submission will be done via ODTUClass. Create a compressed file named ceng.tar.gz that contains CENGVACDB.java and all other classes created by yourself. You will not submit interface and class files provided by us. So, be sure you do not modify them during implementation. The compressed file should contain a directory tree same as the package tree. That is, you should compress the directory named 'ceng' which contains a directory named 'ceng351' which contains a directory named 'cengvacdb' which contains your source files.

```
ceng
└─ ceng351
   └─ cengvacdb
      ├─ CENGVACDB.java
      ├─ AnotherClassIfYouNeed1.java
      ├─ AnotherClassIfYouNeed2.java
      ├─ ..
      ├─ ...
      └─ AnotherClassIfYouNeedN.java
```