

Database Systems Project Documentation

Project 01 Blood Bank Management System

Team Members:

- 1) HARSH PRIYADARSHI - 2020A7PS0110P
- 2) SHASHANK SHREEDHAR BHATT - 2020A7PS0078P

INTRODUCTION

This document is going to introduce you to our project. It is meant to give you a rough gist of what we have done, the exact implementation of which will be explained in the video.

The project assigned to us is the Blood Bank Management System. In this project, we have made a Blood Bank Management System where we match people wishing to give blood (donors) and people needing the blood (receivers) using their blood types and location.

When designing the Management System, we took heavy inspiration from the NSS Blood Donation Camp that was recently organised.

The system is a menu-driven, selective query and update-based program that uses Python for its backend & frontend and MySQL for the actual database. It models a traditional blood bank considering all the different operations one may perform. It consists of a GUI with which the user can interface.

It consists of the Main Menu and 12 options to manipulate the given database. All the options here are on the ADMIN side, as it would finally be the Blood Bank inputting all this information into a database. The 12 options are insertion, update and deletion operations on various database tables.

REQUIREMENTS

- MySQL (preferably version 8.0.28) and Python 3 have been used.
- The program requires that the 2 “.ico” images and the “.jpg” image labelled ‘bb.jpg’, ‘bb.ico’ and ‘images.ico’ be placed in the same folder as the python file named ‘try.py’.
- Now, within the python file itself, the user must enter their MySQL account password in the required places (indicated clearly by a comment) so that the software will be able to connect to the database.
- ❖ Other than the above formalities, the user must also make sure that they have downloaded the following python libraries:

➤ mysql.connector

- `pip install mysql.connector` in case you need to install the package

➤ tkinter

- `pip install tk` in case you need to install the library

➤ PIL

- `pip install Pillow` in case you need to install the library

➤ email.mime

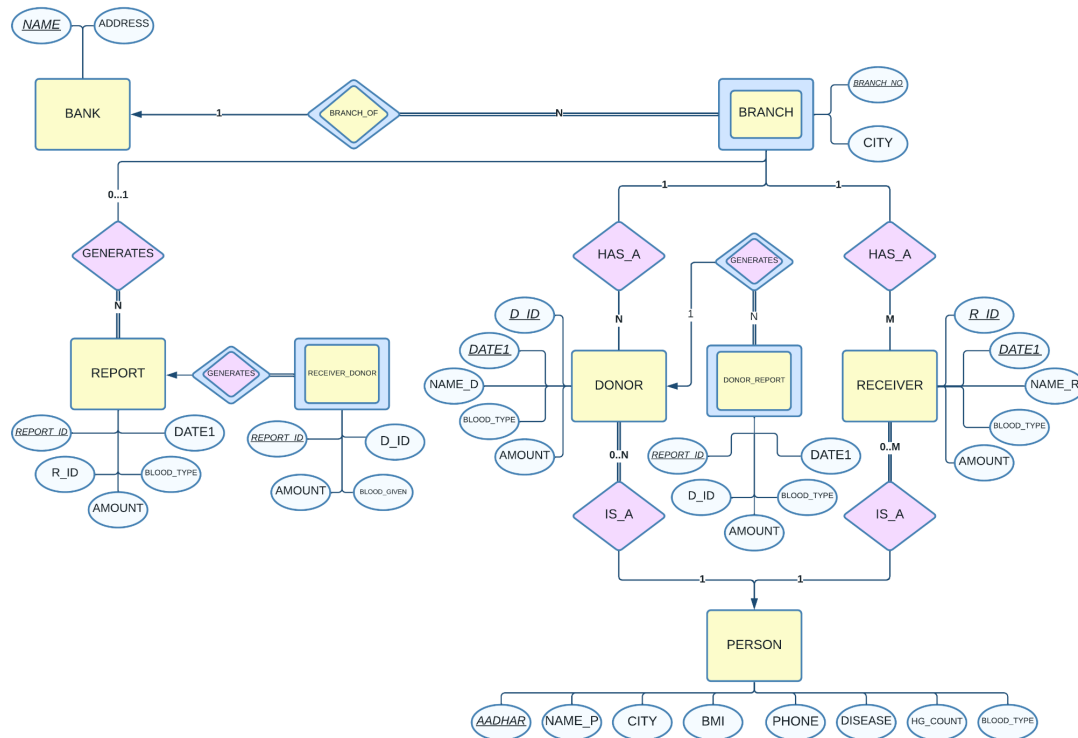
- `pip install email` in case you need to install the library

➤ turtle

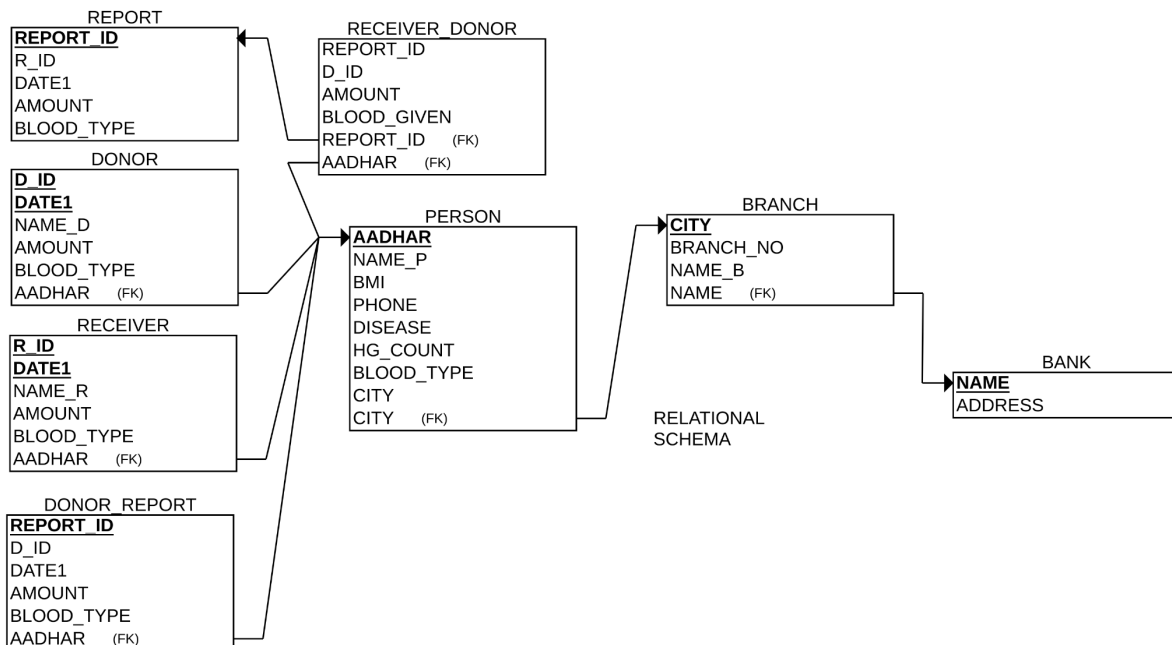
- `pip install Pillow` in case you need to install the library

System Modelling

- The Entity-Relationship diagram of the system:



- The Relational Schema of the system:



Data Normalisation

All tables were chosen such that the tables are already in at least BCNF. Therefore, there was no need for Normalization other than the REPORT table in the 2NF form.

We normalised the REPORT table into BCNF. We split the REPORT table into REPORT & RECEIVER_DONOR tables. We did the split to remove the transitive dependencies and make the LHS of all functional dependencies a superkey, thereby reducing the possible data anomalies.

All the tables are in BCNF as in all the tables. Only the superkeys determine all of the attributes.