Basics of database systems

Project – Database design

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2022

TABLE OF CONTENTS

1 DEFINITION

The database in this project is developed for a railway company to combine the information of available trains, routes and tickets with customers and employees of the railways company for easy access. Because of the large number of expected users, the database will contain a lot of data so good optimization is a must. The database allows the administration of train routes and sold tickets, but also enables shift managers to assign employees to a specific train to work on. The database should contain important data of trains such as departure and destination locations as well as departure and arrival times. Trains can travel to multiple stations through other stations and every trip between two stations is stored in the database, regardless if it happened in the past or will happen in the future. So in other words, this database can also be used to schedule and sell tickets to future journeys.

Possible users of the database are administrators, staff, customers and for example information boards. The administrators have total access to the database. Staff members of the company can in addition to their own data, read information of trains and customer's tickets and also change data regarding to these. This is enabled due to possible humane errors in the purchase of tickets or for example if the train is late for some reason and the information has to be updated. Customers can read and change their own data and read data regarding to trains, their trips and stations. Information boards can read all the data regarding to trains, their trips and stations.

2 MODELING

2.1 Concept model

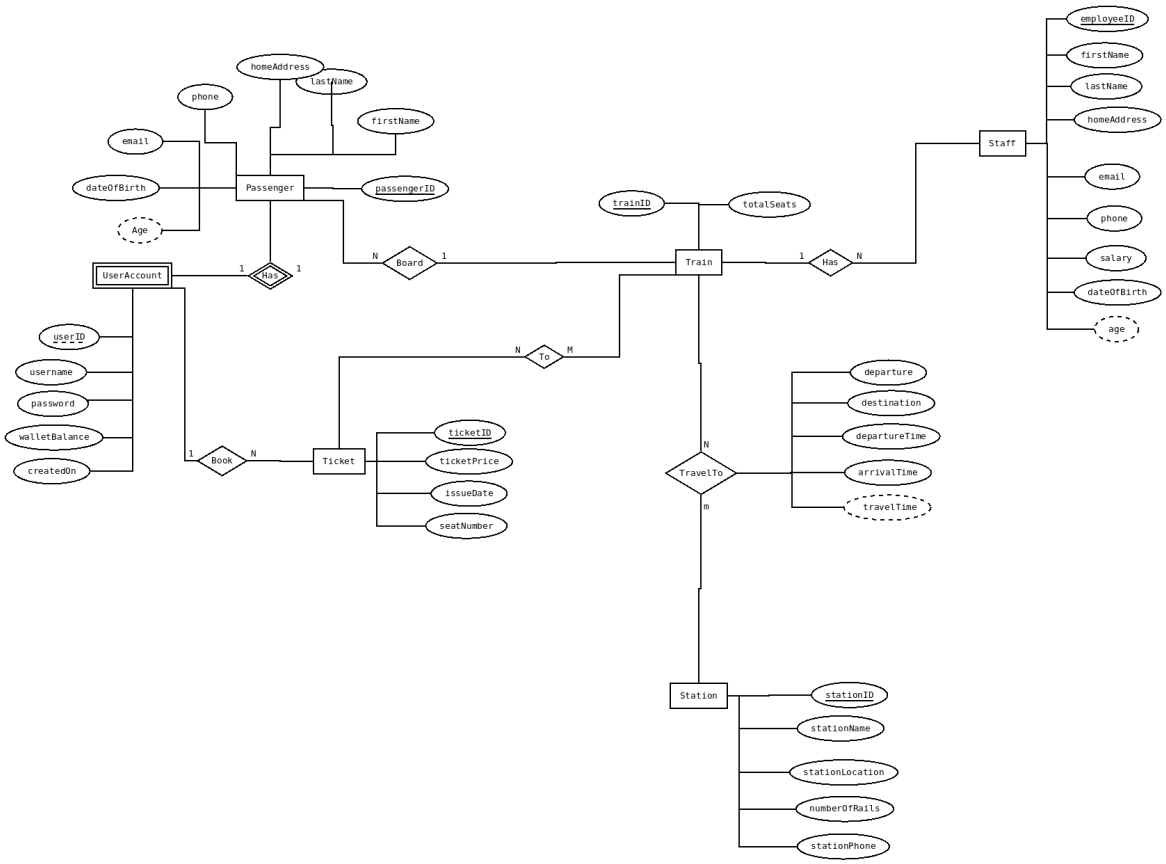

Photo 1: ER-model

In the conceptual model of the database seen in Photo 1, there are six entities: passenger, user account, ticket, train, station, staff. these are used to represent the needed items to form a database which handles tickets to trains and their timetables. Between these entities there are relationships which are presented with diamond shaped objects.

The model has one 1:1 relationship between passengers and their user accounts. This is made to represent the fact that one passenger can only create one user account to book a ticket. There are two 1:N relatioships and these are: user accounts and tickets and train and staff. This means that one user account can buy multiple tickets but the same ticket can be bought only by user account and multiple staff members can work on one train but a staff member can work only one train at a time. Finally, there are two N:M relatioships that are between tickets and trains together with trains and stations. The first relationship models

the real world case of regional tickets, which means that a ticket can be used on multiple trains and a train can be boarded with a variety of ticket types. The other relationship models the case where one train can travel to multiple stations and one station can have multiple trains travelling to it.

Attributes of the entities shown in the picture are the data that will be stored in the database. Underlined attributes are key attributes and they identify each data set from one another in the entity.

2.2 Relational model

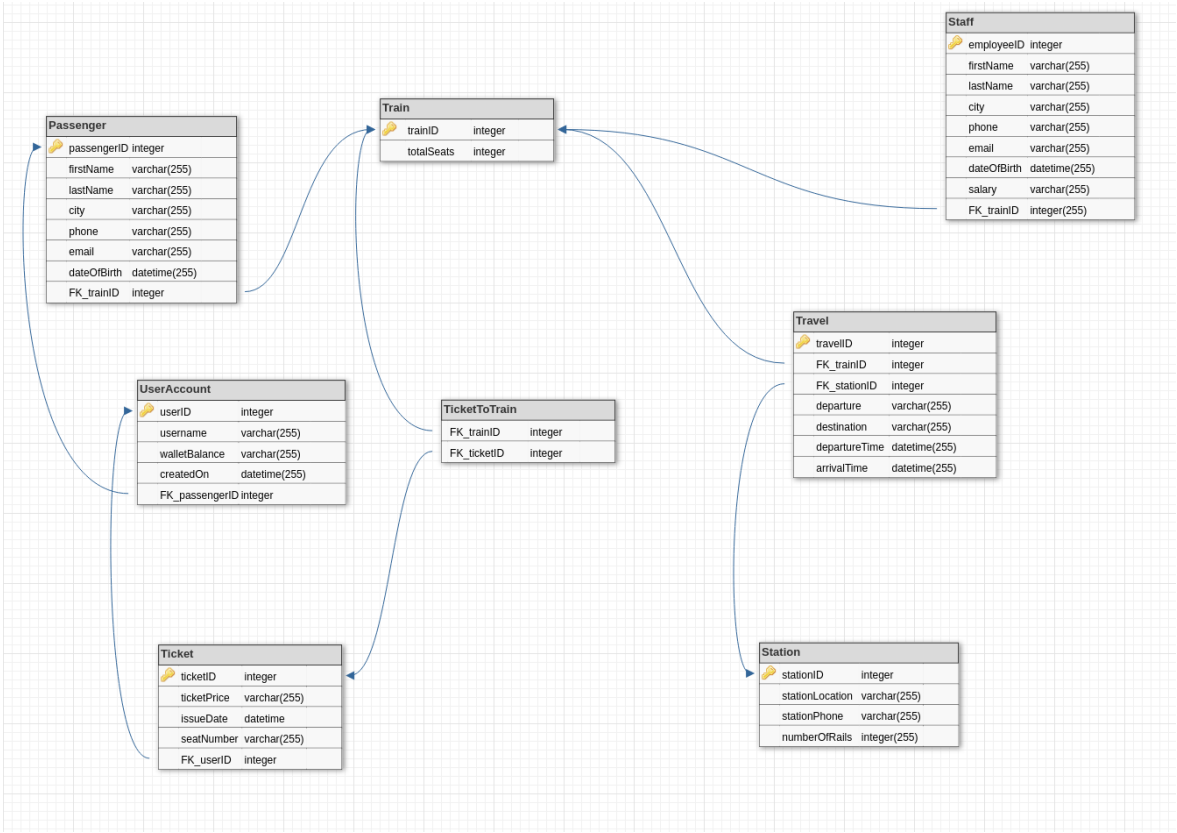

Photo 2: Relational model

The tranformation from ER-model to a relational model seen in picture 2 was done using the transformation rules. First each strong entity was made into a relation and key attributes were used to form primary keys for the entities. Primary keys are presented in the model with a yellow key next to the data type of the primary key. The ER-model had one

weak entity which was made into a relation just like the strong entity however the primary key was formed using the key attributes from both the weak and strong entities.
The transformation of attributes was simple because the ER-model only had a few derived attributes in addition to key and normal attributes. Derived attributes had to be discarded. Finally, the relationships between entites had to be transformed. The 1:1 relationships meant that one of the relations had to have a foreign key in the relational model. The foreign keys were placed in the relation that made most sense to the function of the database. In 1:N relationships the foreign key was placed on the “many” side of the relationship. In the N:M relationships a linking relation had to be formed, which would store the attributes that were directly related to the relationship and the foreign keys to both of the relations in the relationship.

3 DATABASE IMPLEMENTATION

Transformation of ER-model to relational model was pretty straightforward. There were a few comprehension problems about how to transform N:M relationships but that was due to my knowledge and not the design of the model. So I had to double check a few things.
During the conversion I also noticed that the information of end users like customers would not be useful information for the user of the database. Because of that the database and it’s user interface focuses on the trains and their schedules.
To speed up the use of the database I implemented indices for stations which cover a big part of the databases daily usage. Another subject for indices could be the times of trains departuring and arriving. However these values are updated frequently, so could be that indices would just slow things down.

For the implementation of the Python program, I queried the database in DB Browser for SQLite with traditional SQL queries and the transported these queries to the Python program. The Python program prints simple queries regarding to station and train information, can update train departure time and insert a new station. It also demonstrates number of trains departuring from each station using the Bokeh library.