

## REPORT

The object of the project is to design, implement and test a relational database for a given scenario and to use queries. The second requirement is to demonstrate Codd's rules.

### 1. Relational schema:

The file "RelationalSchema.pdf" contains database schema with tables and relations.

The file "RelationalSchema.sql" contains SQL script to create the schema.

In order to achieve a relational database the following steps were taken, recursively:

- Database planning: the fact-finding techniques were used.
  - ✓ mission statement: The purpose of the dental clinic database is to maintain appointment system, patients data, payments and cooperation between other dental clinic.
  - ✓ mission objectives: to maintain data on staff, patients, appointments, payments; to perform search on appointments, payments, available treatments and appointments.
- System definition:
  - ✓ Scope and boundaries of database system:
    - The following main entity are include: Branch, Employee, Patient, Appointment, Bill, Payment, Treatment and other
    - The following entity are out of scope: HRM, Marketing, Revenue, Accounts.
  - ✓ The main user views:
    - Dentist view: Patient, Appointment
    - Secretary: Patient, Appointment, Employee, Payment
  - ✓ Terminology: specialized terminology for treatments, categories.
- Requirement collection and analysis:
  - Requirements for view: view of appointments, payments, available treatments, changing treatment price, ect
  - The significant overlap was detected between both views. Therefore the centralized approach used to specify requirements.
- Database design:
  - ✓ Conceptual design:
  - ✓ Logical design:
  - ✓ Physical design: (MySQL DBMS)

ER modelling, a top-down approach, was used in a process of database design. That process was followed by Normalization.

- Implementation: the file 'RelationalSchema.sql' includes SQL definition statement for MySQL DBMS.
- Data conversion and loading: 'TestData.sql' includes SQL statements to INSERT data
- Testing: the file 'Queries.sql' consists SQL statements to test database

### Explanations to the schema:

The database is designed to be used as a shared database between dental clinics, however it may be used for a single dental clinic. The data about branches are stored in 'Branch' relation.

Relations 'Appointment' and 'WorkSchema' use one hour slots that are stored in 'appointmentDateTime' and 'workDateTime' respectively as TIMESTAMP data type. It enables to manage the timeslots.

Table 'WorkSchema' stores working hours of all employees, including secretary. The foreign key has option ON DELETE CASCADE that enables to remove all data from the parent and child table, ie. for secretary. However, a data of employee that have appointment (dentist and nurse) will not be removed as they need to be stored in the 'Appointment' table.

The other foreign keys have default RESTRICT setup as they should be stored in a database.

Table 'Bill' was created on the assumption that there is one bill for a single appointment. The column 'appointmentNumber' is a primary key for that table and a foreign key referencing to 'Appointment' table.

There may be a single or few payments for the same bill.

Clinic will use speciality application for a teeth examination. Those files are stored in the field 'teethDescription' in the 'Patient' table. The health history files are stored in 'historyHealthForm' in the 'Patient' table.

Table 'TreatmentOffered' includes treatments offered by each of an employee and 'price' hence employees may offered different price and every employee updates only their price.

The table 'AppointmentDetail' has a relation with table 'TreatmentOffered' and includes the attributes 'employeeNumber' and 'treatmentNumber' that are foreign keys referenced to the table 'treatmentOffered' ('AppointmentDetail.employeeNumber' relates to the 'TreatmentOffered.employeeNumber', not to the 'Appointment.employeeNumber').

Table 'Cancellation' stores data about cancellations. This data are required as late cancellation may result in a cancellation payment. In case of appointment cancellation the appropriate data are copied from 'Appointment.appointmentDateTime' to 'Cancellation.appointmentDateTime'. After this the data in 'Appointment.appointmentDateTime' are updated to NULL which results in making the appointment slot available.

## **2. Test data.**

All INSERT statements and data are in the 'TestData.sql' file. They cover the hypothetical one week of dental clinic operation.

## **3. Queries.**

All queries are in the 'Queries.sql' file. They cover all the hypothetical queries to the database based on the given scenario.

Some of the queries were run on database named 'g00364553\_zamorski'.

Most of them may be executed separately, however few of them relies on the previous queries.

## **4. Codd's Rules.**

Codd's Rules with the SQL code demonstration and narrative explaining are in the 'Codd'sRules.pdf' file.

## **5. Project Report**