

Входной контроль

дисциплины «Технологии разработки распределённых вычислений»

Используемые технологии:

- ОС: Fedora 30;
- ЯП: Java 11;
- Настольная СУБД: SQLite 3;
- Корпоративная СУБД: PostgreSQL 12;
- Табличный процессор: LibreOffice 7 Calc;

Все необходимые зависимости можно посмотреть в */pom.xml*.

1. Проектирование базы данных

Выбранная предметная область – регистратура Городской поликлиники №7 г.Перми. В базе данных будет храниться информация о:

- врачах:
 - ФИО;
 - дата рождения;
 - пол;
 - образование;
 - категория;
 - должность;
 - департамент и специальный департамент, в котором работает врач;
- пациентах:
 - ФИО;
 - дата рождения;
 - пол;
- приемах:
 - время приема;
 - статус (пациент пришел/не пришел);
 - комментарий.

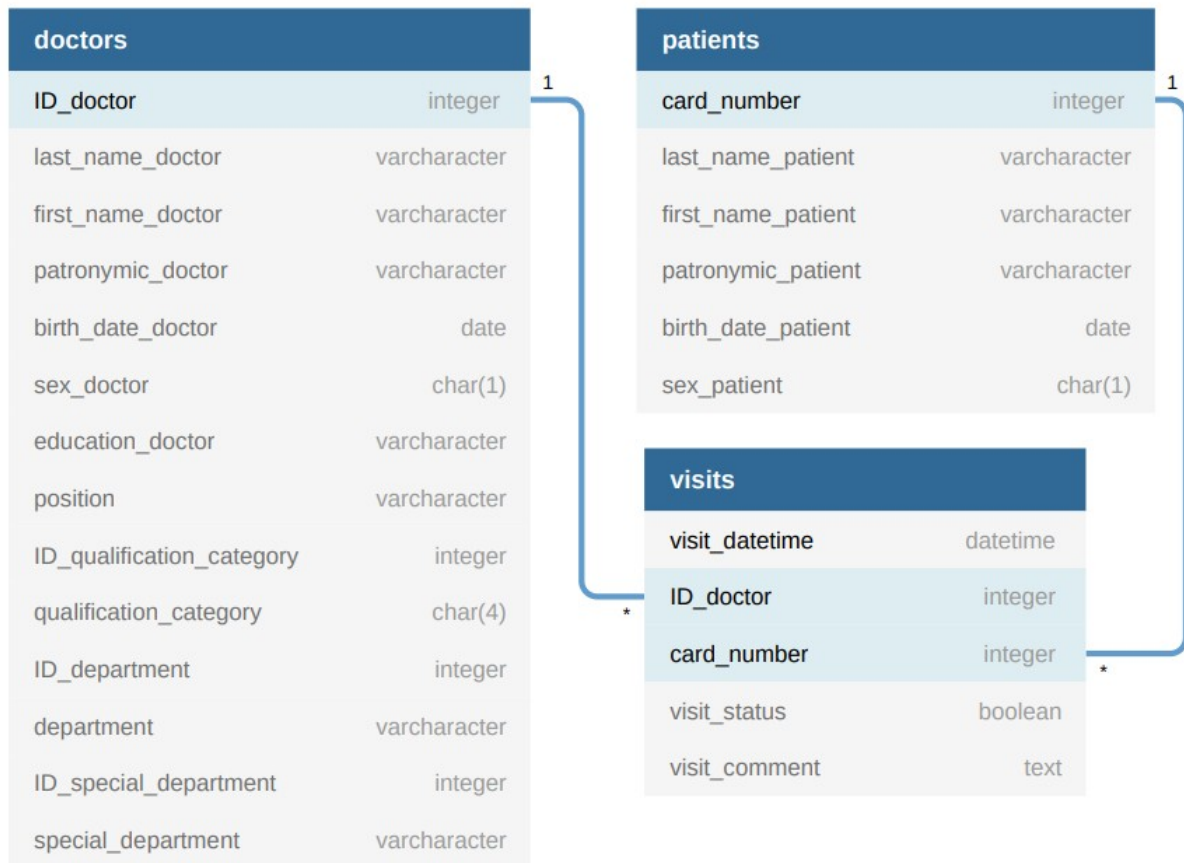
Ненормализованная структура БД выглядит следующим образом:

Perm_city_polyclinic_7_registry	
ID_doctor	int
name_doctor	varcharacter
birth_date_doctor	date
sex_doctor	char(1)
education_doctor	varcharacter
position	varcharacter
qualification_category	char(4)
department	varcharacter
special_department	varcharacter
name_patient	varcharacter
birth_date_patient	date
sex_patient	char(1)
card_number	integer
visit_datetime	datetime
visit_status	boolean
visit_comment	text

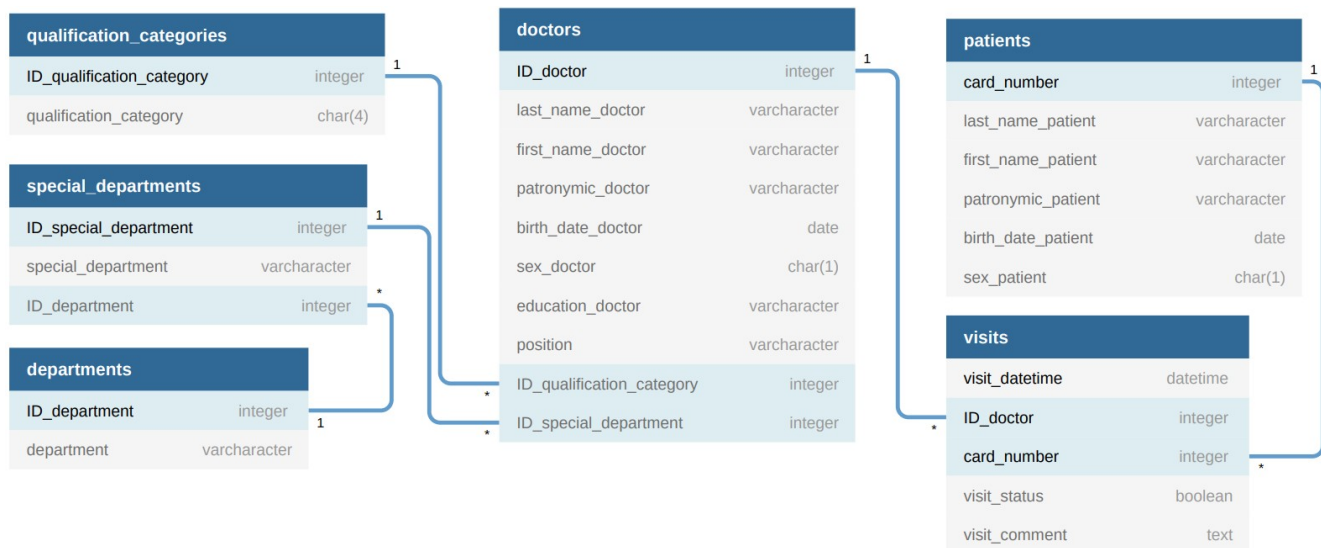
1NF: поля атомарны, введен первичный ключ

Perm_city_polyclinic_7_registry	
ID_doctor	integer
last_name_doctor	varcharacter
first_name_doctor	varcharacter
patronymic_doctor	varcharacter
birth_date_doctor	date
sex_doctor	char(1)
education_doctor	varcharacter
position	varcharacter
ID_qualification_category	integer
qualification_category	char(4)
ID_department	integer
department	varcharacter
ID_special_department	integer
special_department	varcharacter
last_name_patient	varcharacter
first_name_patient	varcharacter
patronymic_patient	varcharacter
birth_date_patient	date
sex_patient	char(1)
card_number	integer
visit_datetime	datetime
visit_status	boolean
visit_comment	text

2NF: все независимые от первичного ключа поля вынесены в отдельные таблицы. Для новых таблиц введены первичные ключи



3NF: все независимые от первичного ключа поля, содержимое которых могло относиться к нескольким записям, вынесены в отдельные таблицы. Для новых таблиц введены первичные ключи



2. Алгоритм работы

Ненормализованная бд хранится в SQLite. Структура БД создается скриптом `/scripts/structureSQLite.sql`, вызываемом в приложении. Структура нормализованной БД также создается скриптом `/scripts/structurePostgreSQL.sql`, вызываемом в приложении.

Данные для ненормализованной БД формируются в приложении. Для заполнения таблицы используются файлы с описанием докторов (`/csv/doctors.csv`) и пациентов (`/csv/patients.csv`). Список возможных часов и дат прием пациентов, статус приема и комментариев к приему задаются в приложении.

Все манипуляции с данными выполняются с помощью библиотеки для объектно-реляционного отображения `joor`.

После заполнения данными БД в третьей нормальной форме PostgreSQL происходит выгрузка таблиц в формате csv в определенную директорию. Все таблицы можно открыть в LibreOffice Calc с помощью макроса `/calc_macro`.

Классы приложения:

- *Main* – основной класс, в котором находятся вызовы загрузки данных в SQLite, создания структуры в PostgreSQL, миграции данных из SQLite в PostgreSQL и выгрузки данных в CSV из PostgreSQL;
- *Migration* – класс, содержащий методы для миграции данных из SQLite в PostgreSQL;
- *PostgreSQL* – класс, содержащий методы для создания структуры БД PostgreSQL, выгрузки данных в CSV;
- *SQLite* – класс, содержащий методы для создания структуры БД SQLite, наполнения ее данными;

3. Сборка

Сборка приложения производится с помощью *maven*:

```
mvn clean install
```

Запуск:

```
java -jar target/db_import_export-1.0-SNAPSHOT.jar
```