

# Лабораторная работа №1

*Тема: проектирование реляционной базы данных*

*PostgreSQL*

**Выполнил студент группы:** ИС-21

Игнатьев Данила

**Цель работы:** Получение практических навыков проектирования реляционных баз данных и работы в CASE-средстве.

**Задание:** спроектировать согласно варианту реляционную базу данных, используя методологию IDEF1X, для этого необходимо:

- построить ER-модель (логическую и физическую) с помощью CASEсредства, например dbdiagram.io или Lucidchart (не менее 7 сущностей),
- **описать** ограничения целостности (CONSTRAINT, UNIQUE)
- вывести программный код создания БД на языке целевой СУБД (прямое проектирование),
- перенести код в СУБД (выполнить),
- \*внести изменения в схему БД и построить новую физическую модель измененной базы данных (обратное проектирование), если доступно создание ODBC источника данных.

База данных приемного отделения больницы предназначена для хранения данных о пациентах их результатах осмотров, врачах, их расписании и специальностях, а также записях в регистратуре.

В данной базе данных сущностями являются следующие понятия:

- **Patients** – содержит информацию о пациентах
- **Doctors** – содержит данные о врачах
- **Record** – содержит информацию о записях в регистратуре
- **Specialties** – содержит данные о специальностях врачей
- **medical records** – содержит информацию о результатах осмотра пациента (например чем он болел)
- **offices** – содержит данные о кабинетах
- **schedule** – содержит данные о расписании врачей

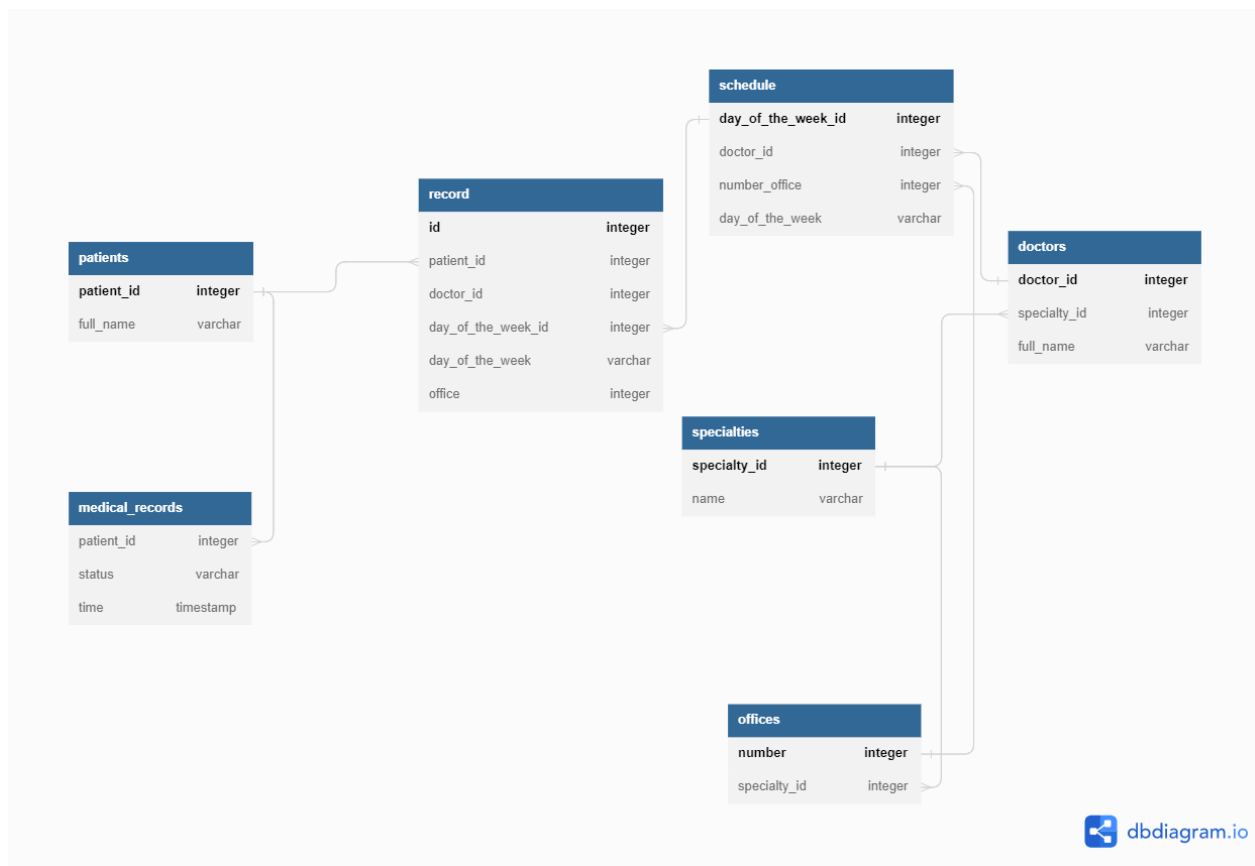


Рис. 1 – модель БД

Код создания БД на языке целевой СУБД:

```
CREATE TABLE "patients" (  
  "patient_id" integer PRIMARY KEY,  
  "full_name" varchar NOT NULL  
);
```

```
CREATE TABLE "doctors" (  
  "doctor_id" integer PRIMARY KEY,  
  "specialty_id" integer,  
  "full_name" varchar NOT NULL  
);
```

```
CREATE TABLE "record" (  
  "id" integer PRIMARY KEY,  
  "patient_id" integer,  
  "doctor_id" integer,  
  "day_of_the_week_id" integer,  
  "day_of_the_week" varchar NOT NULL,  
  "office" integer NOT NULL  
);
```

```
CREATE TABLE "specialties" (  
  "specialty_id" integer PRIMARY KEY,  
  "name" varchar NOT NULL  
);
```

```
CREATE TABLE "medical_records" (  
  "patient_id" integer,  
  "status" varchar NOT NULL,  
  "time" timestamp NOT NULL  
);
```

```
CREATE TABLE "offices" (  
  "number" integer PRIMARY KEY,  
  "specialty_id" integer NOT NULL  
);
```

```
CREATE TABLE "schedule" (  
  "day_of_the_week_id" integer PRIMARY KEY,  
  "doctor_id" integer NOT NULL,  
  "number_office" integer NOT NULL,  
  "day_of_the_week" varchar NOT NULL  
);
```

```
ALTER TABLE "record" ADD FOREIGN KEY ("patient_id") REFERENCES "patients" ("patient_id");
```

```
ALTER TABLE "medical_records" ADD FOREIGN KEY ("patient_id") REFERENCES "patients"  
("patient_id");
```

```
ALTER TABLE "schedule" ADD FOREIGN KEY ("number_office") REFERENCES "offices" ("number");
```

```
ALTER TABLE "schedule" ADD FOREIGN KEY ("doctor_id") REFERENCES "doctors" ("doctor_id");
```

```
ALTER TABLE "doctors" ADD FOREIGN KEY ("specialty_id") REFERENCES "specialties" ("specialty_id");
```

```
ALTER TABLE "offices" ADD FOREIGN KEY ("specialty_id") REFERENCES "specialties" ("specialty_id");
```

```
ALTER TABLE "record" ADD FOREIGN KEY ("day_of_the_week_id") REFERENCES "schedule"  
("day_of_the_week_id");
```