

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЗАПОЛЯРНЫЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.  
М. ФЕДОРОВСКОГО**

**ОТЧЕТ**

**по лабораторной работе  
по дисциплине “Базы данных”**

*Тема: проектирование реляционной базы данных  
PostgreSQL*

**Группа: ИС-21**

**Студент: 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(пациенты)** – содержит информацию о пациентах
- **Doctors(доктора)** – содержит данные о врачах
- **Record(записи)** – содержит информацию о записях в регистратуре
- **Specialties(специальности)** – содержит данные о специальностях врачей
- **medical records(мед.записи)** – содержит информацию о результатах осмотра пациента (например чем он болел)
- **offices(кабинеты)** – содержит данные о кабинетах
- **schedule(расписание)** – содержит данные о расписании врачей

## Используемые ограничения:

- Primary Key
- Not NULL
- Foreign Key

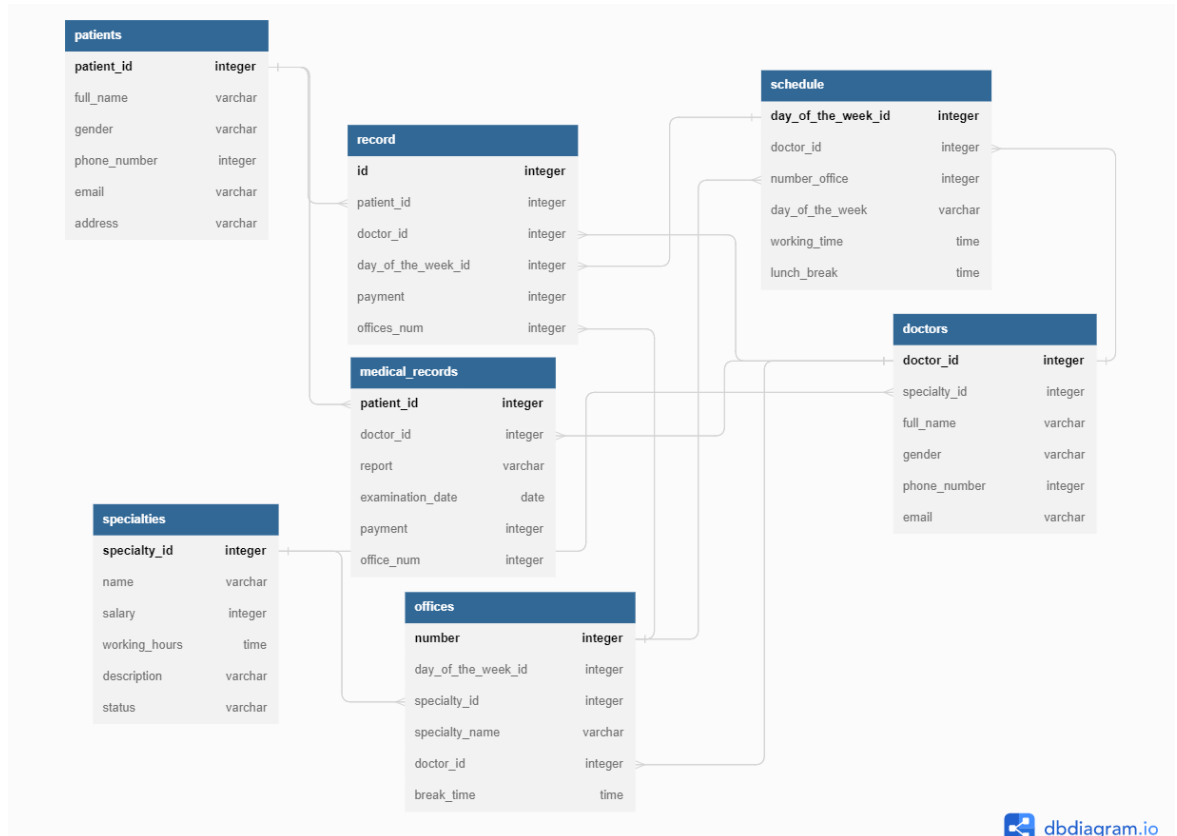


Рис. 1 – физическая модель БД

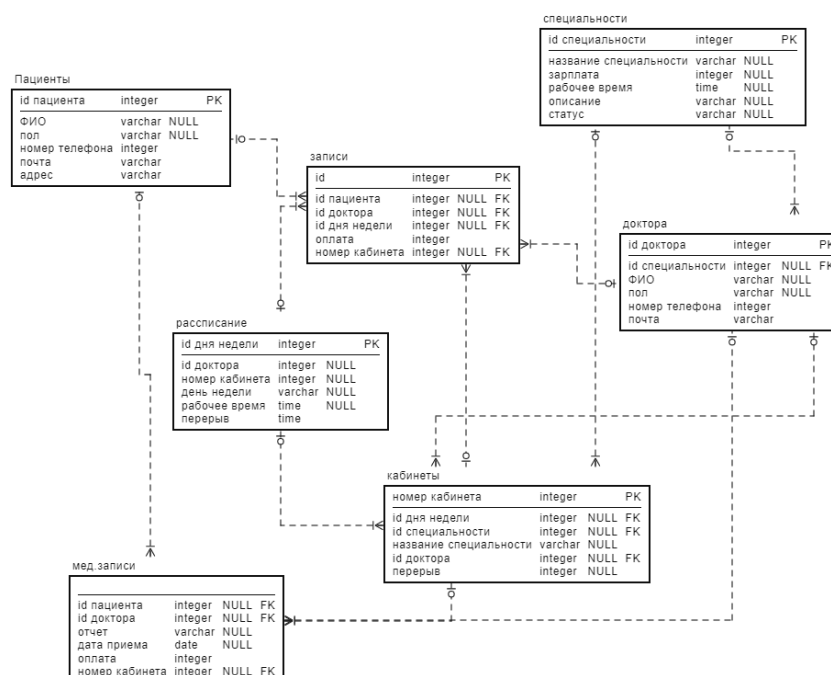


Рис. 2 – логическая модель базы данных

Код создания БД на языке целевой СУБД:

```
CREATE TABLE "patients" (  
  "patient_id" integer PRIMARY KEY,  
  "full_name" varchar NOT NULL,  
  "gender" varchar NOT NULL,  
  "phone_number" integer,  
  "email" varchar,  
  "address" varchar  
);  
CREATE TABLE "doctors" (  
  "doctor_id" integer PRIMARY KEY,  
  "specialty_id" integer NOT NULL,  
  "full_name" varchar NOT NULL,  
  "gender" varchar NOT NULL,  
  "phone_number" integer,  
  "email" varchar  
);  
CREATE TABLE "record" (  
  "id" integer PRIMARY KEY,  
  "patient_id" integer NOT NULL,  
  "doctor_id" integer NOT NULL,  
  "day_of_the_week_id" integer NOT  
NULL,  
  "payment" integer,  
  "offices_num" integer NOT NULL  
);  
CREATE TABLE "specialties" (  
  "specialty_id" integer PRIMARY KEY,  
  "name" varchar NOT NULL,  
  "salary" integer,  
  "working_hours" time,  
  "description" varchar,  
  "status" varchar NOT NULL  
);  
CREATE TABLE "medical_records" (  
  "patient_id" integer PRIMARY KEY,  
  "doctor_id" integer NOT NULL,  
  "report" varchar NOT NULL,  
  "examination_date" date NOT NULL,  
  "payment" integer,  
  "office_num" integer NOT NULL  
);  
CREATE TABLE "offices" (  
  "number" integer PRIMARY KEY,  
  "day_of_the_week_id" integer NOT  
NULL,  
  "specialty_id" integer NOT NULL,  
  "specialty_name" varchar NOT NULL,  
  "doctor_id" integer NOT NULL,  
  "break_time" time 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,  
  "working_time" time,  
  "lunch_break" time  
);  
  
ALTER TABLE "record" ADD FOREIGN  
KEY ("patient_id") REFERENCES  
"patients" ("patient_id");  
  
ALTER TABLE "record" ADD FOREIGN  
KEY ("doctor_id") REFERENCES  
"doctors" ("doctor_id");  
  
ALTER TABLE "record" ADD FOREIGN  
KEY ("day_of_the_week_id")  
REFERENCES "schedule"  
("day_of_the_week_id");  
  
ALTER TABLE "medical_records" ADD  
FOREIGN KEY ("patient_id")  
REFERENCES "patients" ("patient_id");  
  
ALTER TABLE "medical_records" 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 "offices" ADD FOREIGN  
KEY ("doctor_id") REFERENCES  
"doctors" ("doctor_id");  
  
ALTER TABLE "schedule" ADD  
FOREIGN KEY ("doctor_id")  
REFERENCES "doctors" ("doctor_id");  
  
ALTER TABLE "schedule" ADD  
FOREIGN KEY ("number_office")  
REFERENCES "offices" ("number");  
  
ALTER TABLE "record" ADD FOREIGN  
KEY ("offices_num") REFERENCES  
"offices" ("number");
```