

Лабораторная Работа №1

Вариант 716

Кива Глеб Владимирович, Р3108

Преподаватель: Афанасьев Дмитрий Борисович

Базы данных

Задание.....3

Описание предметной области..... 3

Список сущностей..... 3

Инфологическая модель..... 4

Даталогическая модель.....5

Реализация даталогической модели на SQL..... 6

Вывод..... 9

Задание

1. На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
2. Составить инфологическую модель.
3. Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
4. Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
5. Заполнить созданные таблицы тестовыми данными.

Описание предметной области

Введите вариант:

Описание предметной области, по которой должна быть построена доменная модель:

Тем не менее это так. Все мы сдали, и не только психически. Совсем недавно были здоровы до неприличия. Теперь почти у каждого либо простуда, либо расстройство желудка, либо незаживающая царапина. Усилия Катерины тщетны, порошки и пилюли не помогают. Она махнула на нас рукой и лишь изредка чертыхается.

Пациенты больницы страдают от различных заболеваний. Каждому доктору ставят диагноз, выписывая один/несколько мед. препаратов по болезни, которой старается пациент. Пациент в дальнейшем принимает препараты сам, периодически врач описывает его состояние.

Список сущностей

Стержневые:

- Диагноз (Diagnosis)
- Пациент (Patient)
- Доктор (Doctor)

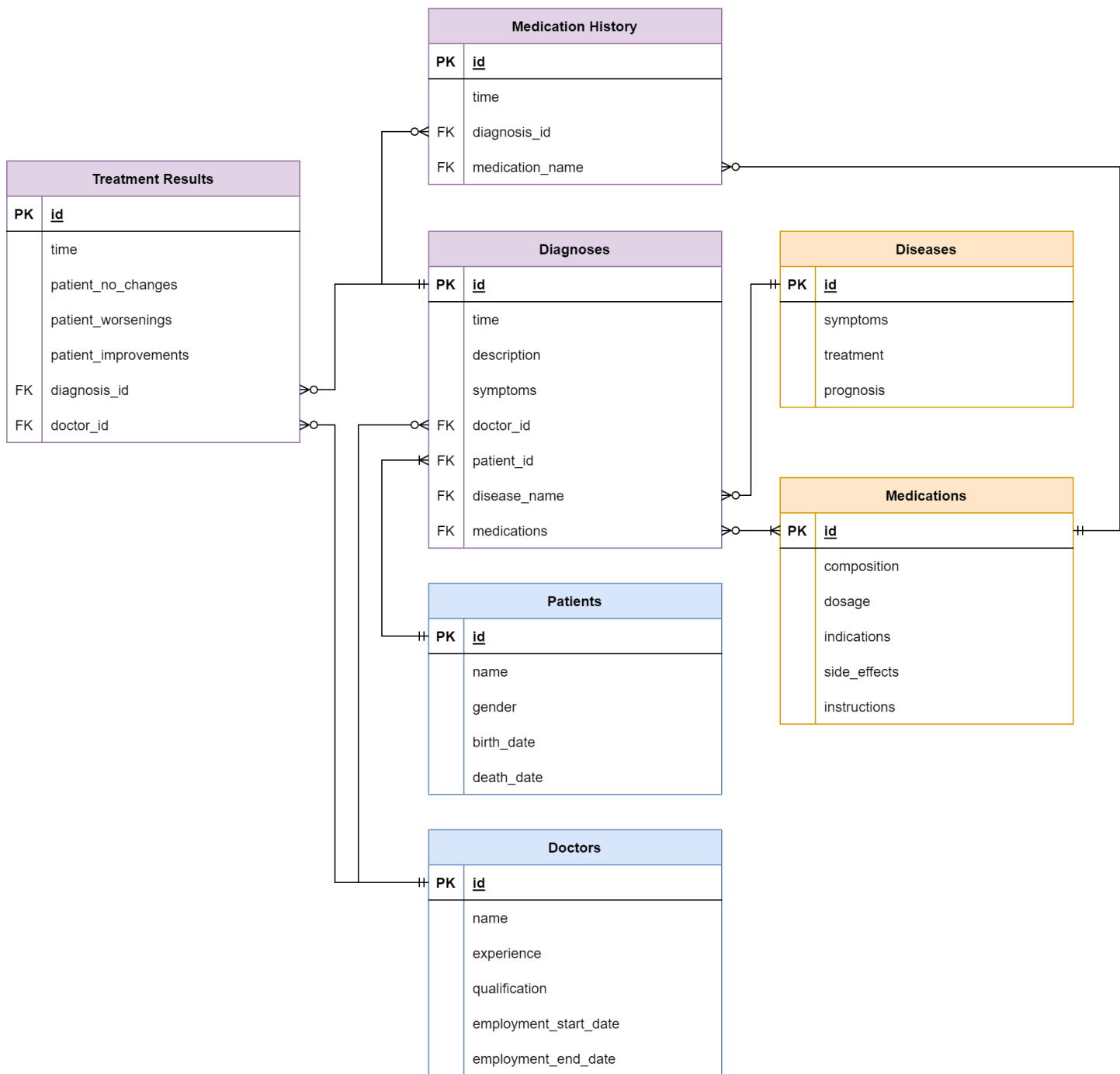
Ассоциативные:

- Назначенные препараты (Assigned Medications)
- История применения препаратов (Medication History)
- Результаты лечения (Treatment Results)

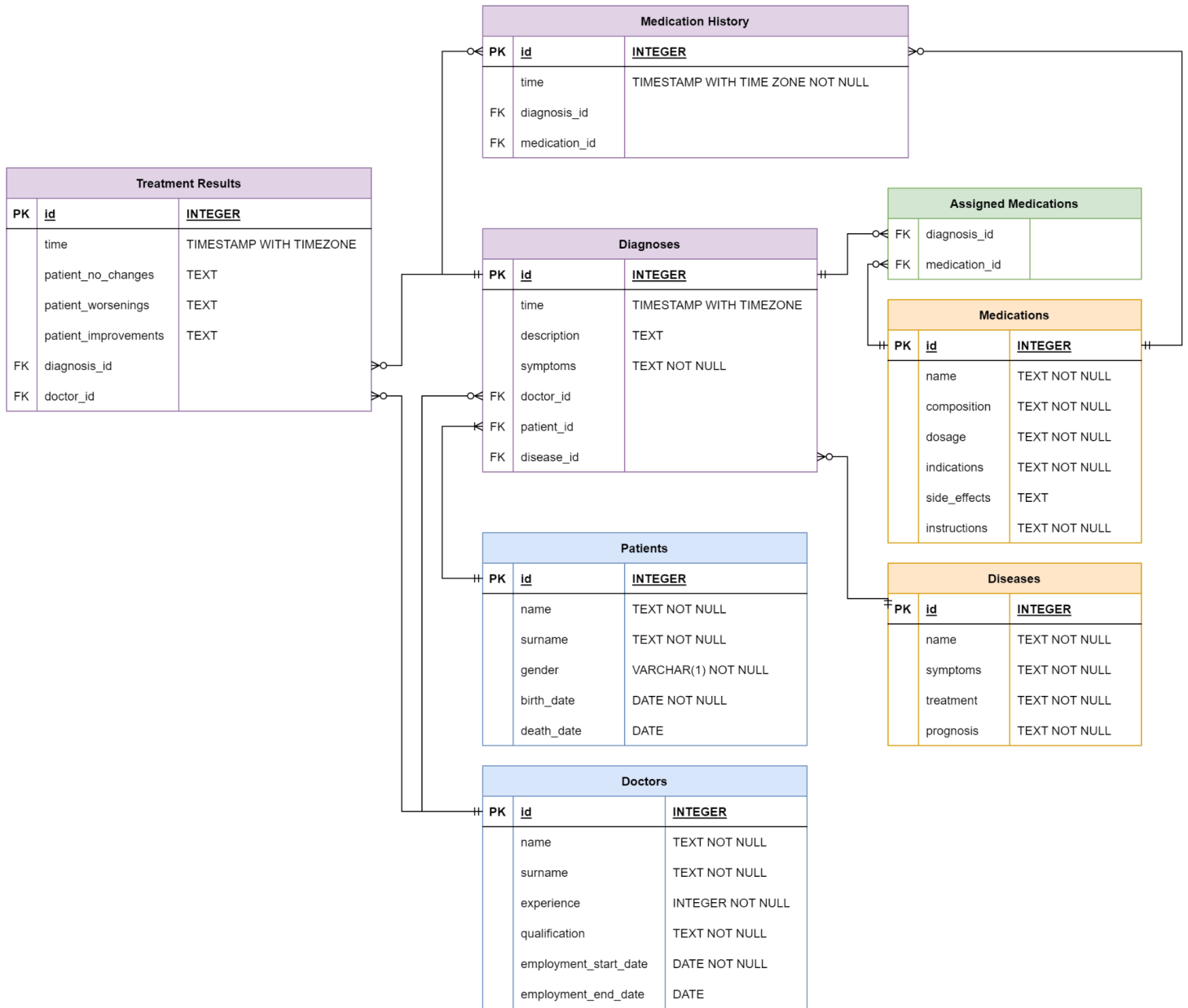
Характеристические:

- Болезни (Diseases)
- Препараты (Medications)

Инфологическая модель



Даталогическая модель



Реализация даталогической модели на SQL

```
BEGIN;
DROP TABLE IF EXISTS MedicationHistory CASCADE;
DROP TABLE IF EXISTS TreatmentResults CASCADE;
DROP TABLE IF EXISTS Diagnoses CASCADE;
DROP TABLE IF EXISTS Medications CASCADE;
DROP TABLE IF EXISTS Diseases CASCADE;
DROP TABLE IF EXISTS Patients CASCADE;
DROP TABLE IF EXISTS Doctors CASCADE;
DROP TABLE IF EXISTS AssignedMedications CASCADE;

CREATE TABLE Patients (
    Id SERIAL PRIMARY KEY,
    Name TEXT NOT NULL,
    Surname TEXT NOT NULL,
    Gender VARCHAR(1) NOT NULL CHECK (Gender IN ('M', 'W')),
    BirthDate DATE NOT NULL,
    DeathDate DATE
);
CREATE TABLE Doctors (
    Id SERIAL PRIMARY KEY,
    Name TEXT NOT NULL,
    Surname TEXT NOT NULL,
    Experience INTEGER NOT NULL,
    Qualification TEXT NOT NULL,
    EmploymentStartDate DATE NOT NULL,
    EmploymentEndDate DATE
);
CREATE TABLE Diseases (
    Id SERIAL PRIMARY KEY,
    Name TEXT NOT NULL,
    Symptoms TEXT NOT NULL,
    Treatment TEXT NOT NULL,
    Prognosis TEXT NOT NULL
);
CREATE TABLE Medications (
    Id SERIAL PRIMARY KEY,
    Name TEXT NOT NULL,
    Composition TEXT NOT NULL,
    Dosage TEXT NOT NULL,
    Indications TEXT NOT NULL,
    SideEffects TEXT,
    Instructions TEXT NOT NULL
);
CREATE TABLE Diagnoses (
    Id SERIAL PRIMARY KEY,
    Time TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    Description TEXT,
    Symptoms TEXT NOT NULL,
    DoctorId INTEGER REFERENCES Doctors(Id),
    PatientId INTEGER REFERENCES Patients(Id),
    DiseaseId INTEGER REFERENCES Diseases(Id)
);
CREATE TABLE MedicationHistory (
```

```

    Id SERIAL PRIMARY KEY,
    Time TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    DiagnosisId INTEGER REFERENCES Diagnoses(Id),
    MedicationId INTEGER REFERENCES Medications(Id)
);
CREATE TABLE TreatmentResults (
    Id SERIAL PRIMARY KEY,
    Time TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    PatientNoChanges TEXT,
    PatientWorsenings TEXT,
    PatientImprovements TEXT,
    DiagnosisId INTEGER REFERENCES Diagnoses(Id),
    DoctorId INTEGER REFERENCES Doctors(Id)
);
CREATE TABLE AssignedMedications (
    DiagnosisId INTEGER REFERENCES Diagnoses(Id),
    MedicationId INTEGER REFERENCES Medications(Id),
    PRIMARY KEY(DiagnosisId, MedicationId)
);

INSERT INTO Doctors (
    Name,
    Surname,
    Experience,
    Qualification,
    EmploymentStartDate
) VALUES (
    'Алёша',
    'Ливси',
    10,
    'Специалист по рому',
    '1999-04-15'
);
INSERT INTO Patients (
    Name,
    Surname,
    Gender,
    BirthDate
) VALUES (
    'Андрей',
    'Смолов',
    'М',
    '2006-10-15'
);
INSERT INTO Diseases (
    Name,
    Symptoms,
    Treatment,
    Prognosis
) VALUES (
    'Корь',
    'Горячая голова и жар',
    'Пить много воды',
    'Развивается в рак'

```

```

);
INSERT INTO Diagnoses (
    Description,
    Symptoms,
    DoctorId,
    PatientId,
    DiseaseId
) VALUES (
    'Ему очень плохо',
    'У него жар',
    1,
    1,
    1
);
INSERT INTO Medications (
    Name,
    Composition,
    Dosage ,
    Indications,
    SideEffects,
    Instructions
) VALUES (
    'Антибиотик АНТИРОМ',
    'Водка',
    '1 ложка',
    'Жар',
    'Трезвость',
    'Пить каждый день до конца жизни'
);
INSERT INTO AssignedMedications (
    DiagnosisId,
    MedicationId
) VALUES (1, 1);
END;

```


Вывод

Во время выполнения лабораторной работы я продумал структуру БД и представил ее в виде инфологической и даталогической моделей, а также познал синтаксис и основные команды для работы с таблицами SQL.