Цель: изучение программных средств для разработки моделей информационной базы ПС, проработка методов нормализации отношений в БД, приобретение навыков применения CASE-средста ERwin для моделирования базы данных ПС.

Изображение выглядит как диаграмма, схематичный

Автоматически созданное описание

Рисунок 1 – модель классов UML

Таблица 1 – идентификация сущностей информационной базы ПС

|  |  |
| --- | --- |
| Документ | Сущность |
| Перечень препаратов и их описание | Препарат |
| Перечень препаратов в заказе | Рецепт |
| Данные о работнике поликлиники | Работник поликлиники |
| Запись о пациенте в журнал | Пациент |
| Заявка на приобретение препарата | Заявка пациента |
| Данные о нахождении заказа пациента | Статус заказа |
| Данные о работнике, пациенте и заказе | Заказ |
| Данные о враче, выписывающем рецепт | Врач |

Таблица 2 – Сущности физической модели данных

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущность | Атрибуты | Описание |
| Справочные | | | |
| 1 | Работник | ID работника, Фамилия, Имя, Отчество, дата рождения | Информация о работниках |
| 2 | Препарат | ID препарата, название, состав, срок годности, наличие на складе, стоимость, скидка | Информация о препарате |
| 3 | Врач | ID врача, ФИО, дата рождения, специальность | Информация о враче |
| Оперативные | | | |
| 4 | Рецепт | ID рецепта, ФИО врача, диагноз, необходимый препарат | Информация о рецепте пациента |
| 5 | Пациент | ИНН, Фамилия, Имя, Отчество, Дата рождения, ID рецепта, наличие льгот | Информация о пациенте |
| 6 | Заявка пациента | ID заявки, ИНН пациента, время подачи заявки | Информация о заявках на покупку |
| 7 | Статус заказа | ID статуса, ID заявки, ID работника, статус, время изменения | Информация о статусе заказа |
| 8 | Заказ | ID препарата, ИНН пациента, ID заявки, ID работника | Информация о заказе |

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 2 – ER-диаграмма логической модели данных

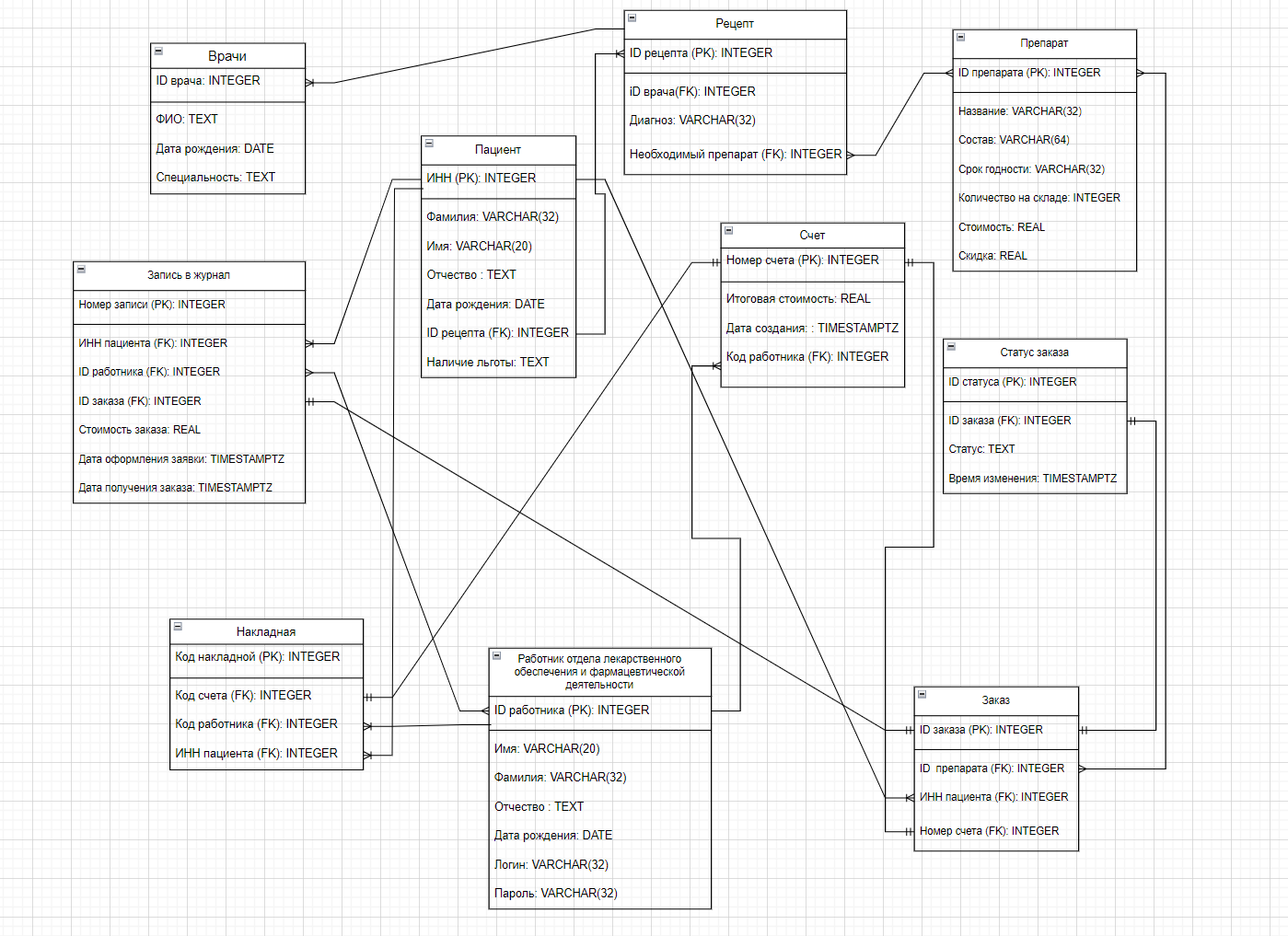


Рисунок 3 – ER-диаграмма физической модели данных

Сравнительный анализ с диаграммой классов, изображенный на Рисунке 1, показал на верную проектировку будущей БД. Существует вероятность дублирования информации, которую можно избежать, проведя нормализацию. Также существуют атрибуты, которые могут быть постоянно заполнены Null.

Для разработки ПС была выбрана СУБД PostgreSQL по следующим причинам:

* Бесплатная объектно-реляционная СУБД с открытым исходным кодом
* Гибкий доступ к базам данных, их организация и хранение
* Поддержка множества типов данных
* Работа с большими объемами
* Поддержка сложных запросов
* Одновременная модификация базы
* Высокая мощность и широкая функциональность
* Наличие удобного интерфейса при работе с БД

Далее представлена генерация SQL-кода для создания реляционной базы данных для ПС:

CREATE TABLE doctor (

id\_doctor BIGSERIAL NOT NULL PRIMARY KEY,

FIO VARCHAR(80) NOT NULL,

date\_of\_b DATE,

specialization TEXT NOT NULL

);

CREATE TABLE recipe (

id\_recipe BIGSERIAL NOT NULL PRIMARY KEY,

id\_doctor INTEGER REFERENCES doctor (id\_doctor),

diagnos VARCHAR(32) NOT NULL,

id\_prep INTEGER REFERENCES preparat (id\_prep)

);

CREATE TABLE preparat (

id\_prep BIGSERIAL NOT NULL PRIMARY KEY,

name VARCHAR(32) NOT NULL,

sostav VARCHAR(64) NOT NULL,

srok\_god VARCHAR(32) NOT NULL,

kol INTEGER DEFAULT 0,

price REAL NOT NULL,

discount REAL DEFAULT 0

);

CREATE TABLE patient (

id\_patient BIGSERIAL NOT NULL PRIMARY KEY,

surname VARCHAR(32) NOT NULL,

name VARCHAR(20) NOT NULL,

otchestvo TEXT,

date\_of\_bir DATE NOT NULL,

id\_recipe INTEGER REFERENCES recipe (id\_recipe),

plyclinic(# lgot TEXT DEFAULT 0

);

CREATE TABLE worker (

id\_worker BIGSERIAL NOT NULL PRIMARY KEY,

surname VARCHAR(32) NOT NULL,

name VARCHAR(20) NOT NULL,

otchestvo TEXT,

date\_of\_bir DATE NOT NULL,

login VARCHAR(32) NOT NULL,

parol VARCHAR(32) NOT NULL

);

CREATE TABLE chet (

id\_chet BIGSERIAL NOT NULL PRIMARY KEY,

price REAL NOT NULL,

date TIMESTAMPTZ NOT NULL,

id\_worker INTEGER REFERENCES worker (id\_worker)

);

CREATE TABLE zakaz (

id\_zakaz BIGSERIAL NOT NULL PRIMARY KEY,

id\_prep INTEGER REFERENCES preparat (id\_prep),

id\_patient INTEGER REFERENCES patient (id\_patient),

id\_chet INTEGER REFERENCES chet (id\_chet)

);

# CREATE TABLE state\_zakaz (

id\_state\_zakaz BIGSERIAL NOT NULL PRIMARY KEY,

id\_zakaz INTEGER REFERENCES zakaz (id\_zakaz),

state TEXT NOT NULL,

time TIMESTAMPTZ NOT NULL

);

CREATE TABLE naklad (

id\_naklad BIGSERIAL NOT NULL PRIMARY KEY,

id\_chet INTEGER REFERENCES chet (id\_chet),

id\_worker INTEGER REFERENCES worker (id\_worker),

id\_patient INTEGER REFERENCES patient (id\_patient)

);

CREATE TABLE write\_mag (

id\_write\_mag BIGSERIAL NOT NULL PRIMARY KEY,

id\_patient INTEGER REFERENCES patient (id\_patient),

id\_worker INTEGER REFERENCES worker (id\_worker),

id\_zakaz INTEGER REFERENCES zakaz (id\_zakaz),

price REAL NOT NULL,

date\_oform TIMESTAMPTZ NOT NULL,

date\_poluch TIMESTAMPTZ NOT NULL

);

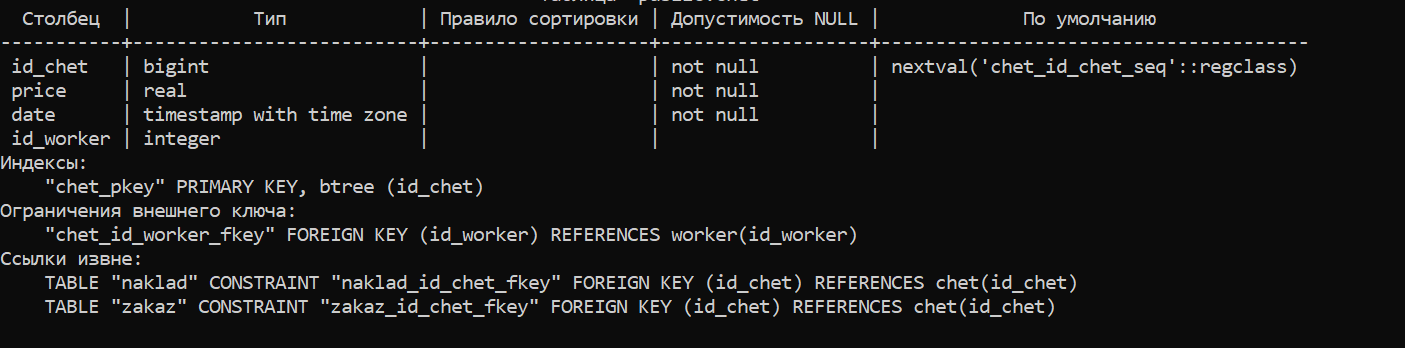
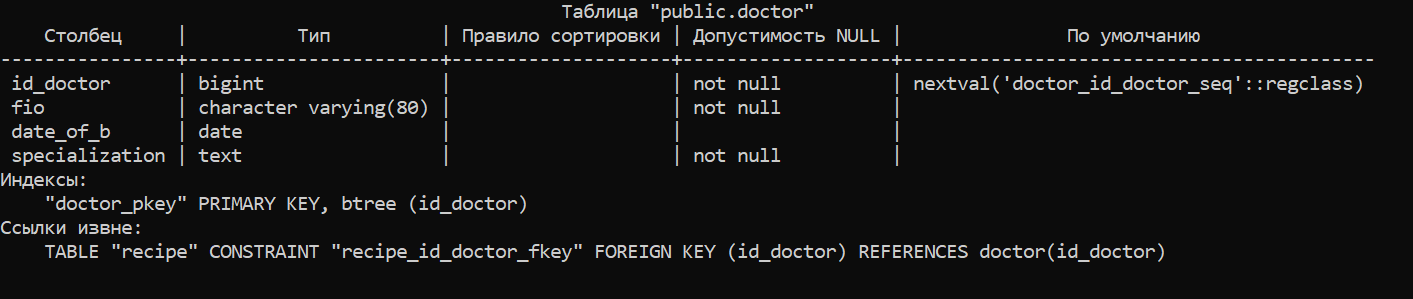


Рисунок 4.1 – Структура таблицы chet

Рисунок 4.2 – Структура таблицы doctor

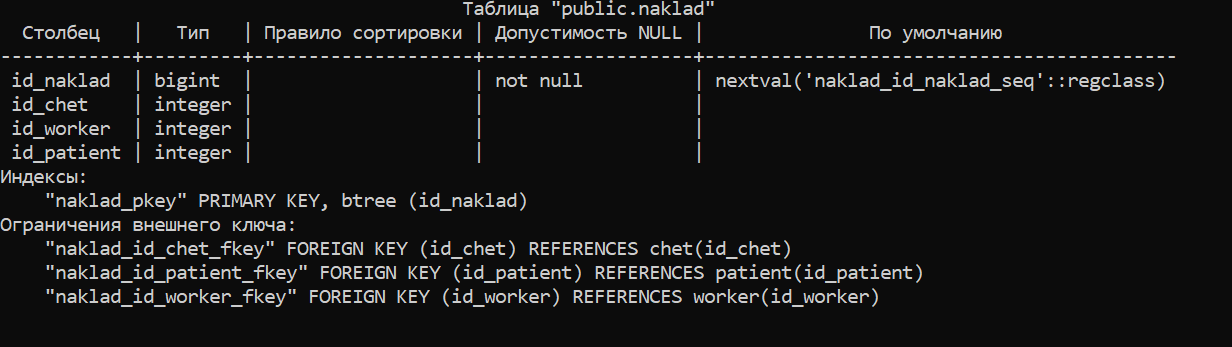


Рисунок 4.3 – Структура таблицы naklad

Изображение выглядит как текст

Автоматически созданное описание Рисунок 4.4 – Структура таблицы patient

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4.5 – Структура таблицы preparat

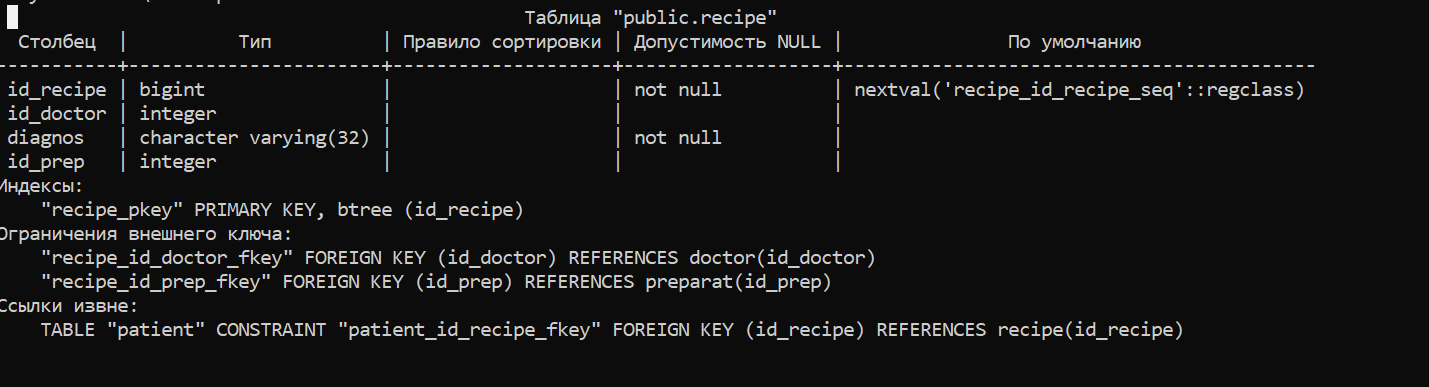


Рисунок 4.6 – Структура таблицы recipe

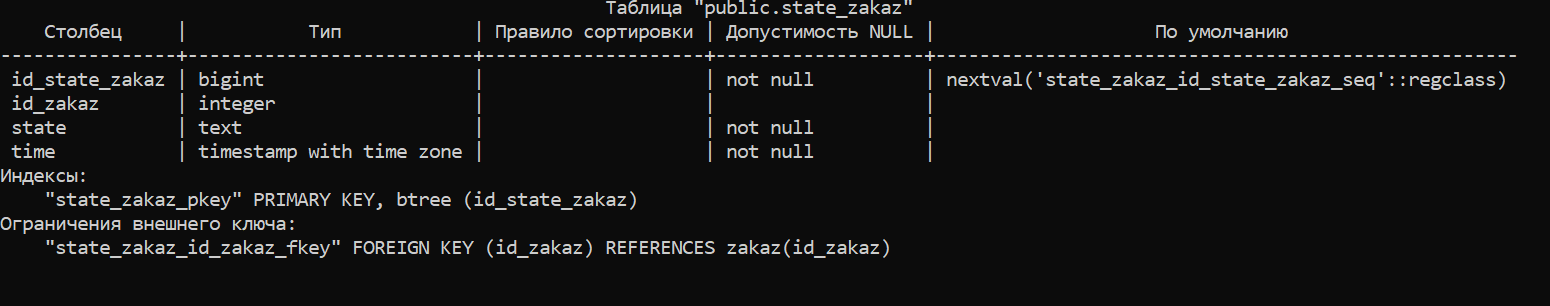


Рисунок 4.7 – Структура таблицы state\_zakaz

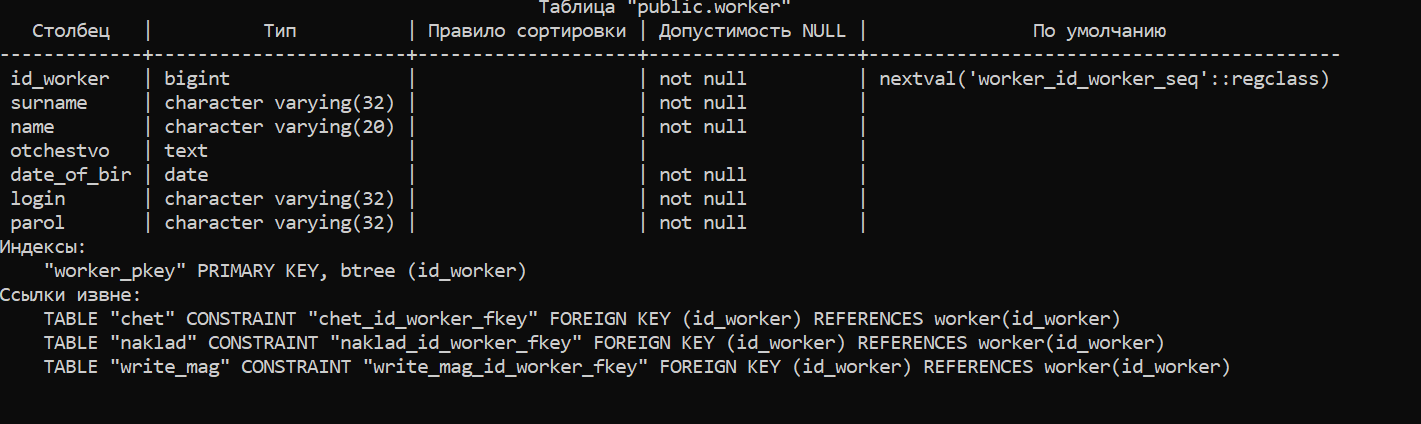


Рисунок 4.8 – Структура таблицы worker

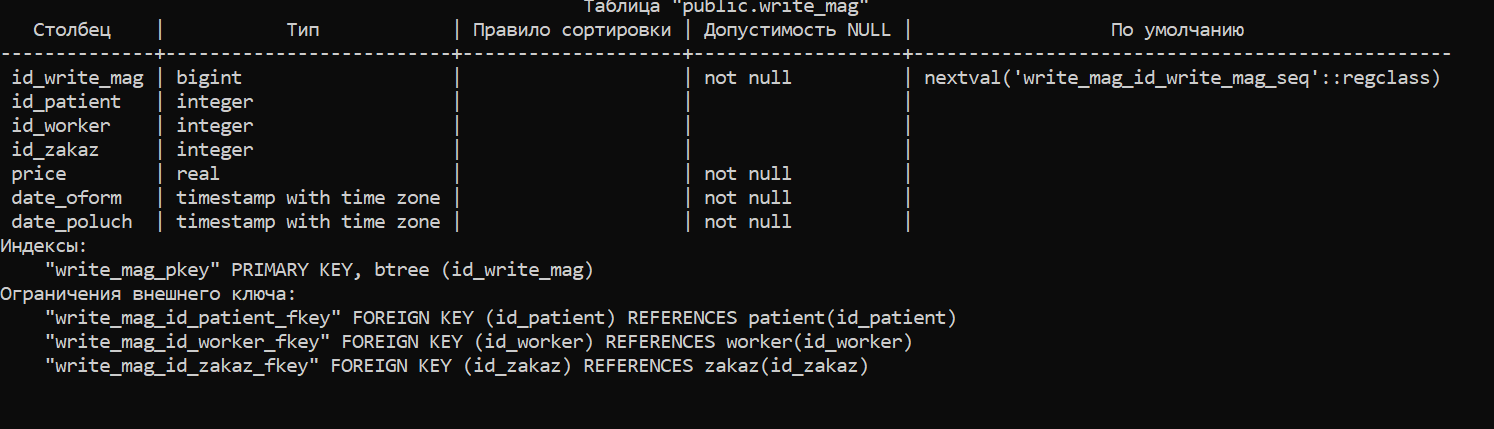


Рисунок 4.9 – Структура таблицы write\_mag

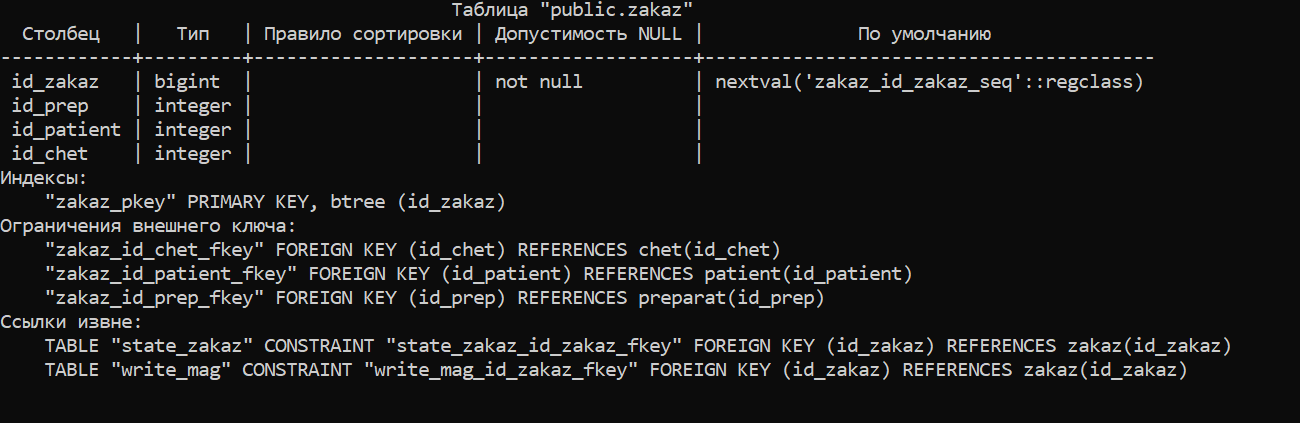


Рисунок 4.10 – Структура таблицы zakaz

Таблица 3 – Список разработанных таблиц

|  |  |  |
| --- | --- | --- |
| № | Имя таблицы | Описание |
| 1 | chet | Описывает сущность счета |
| 2 | doctor | Описывает сущность врача |
| 3 | naklad | Описывает сущность накладной |
| 4 | patient | Описывает сущность пациента |
| 5 | preparate | Описывает сущность препарата |
| 6 | recipe | Описывает сущность рецепта |
| 7 | state\_zakaz | Описывает сущность статуса заказа |
| 8 | worker | Описывает сущность работника поликлиники |
| 9 | write\_mag | Описывает сущность записи в журнал |
| 10 | zakaz | Описывает сущность заказа |

Таблица 4 – Связи между таблицами БД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Родительская таблица | | Дочерняя таблица | | Тип связи |
| Название | Атрибут | Название | Атрибут |
| recipe | id\_doctor | doctor | id\_doctor | Один-ко-многим |
| patient | id\_recipe | recipe | id\_recipe | Один-ко-многим |
| recipe | id\_preparate | preparate | id\_preparate | Многие-ко-многим |
| naklad | id\_worker | worker | id\_worker | Один-ко-многим |
| write\_mag | id\_zakaz | zakaz | id\_zakaz | Один-к- одному |
| write\_mag | id\_worker | worker | id\_worker | Многие-ко-многим |
| naklad | id\_patient | patient | id\_patient | Один-ко-многим |
| naklad | id\_chet | chet | id\_chet | Один-к- одному |
| zakaz | id\_preparate | preparate | id\_preparate | Многие-ко-многим |
| chet | id\_worker | worker | id\_worker | Один-ко-многим |
| zakaz | id\_chet | chet | id\_chet | Один-к- одному |
| stat\_zak | id\_zakaz | zakaz | id\_zakaz | Один-к- одному |
| zakaz | id\_patient | patient | id\_patient | Один-ко-многим |
| write\_mag | id\_patient | patient | id\_patient | Один-ко-многим |

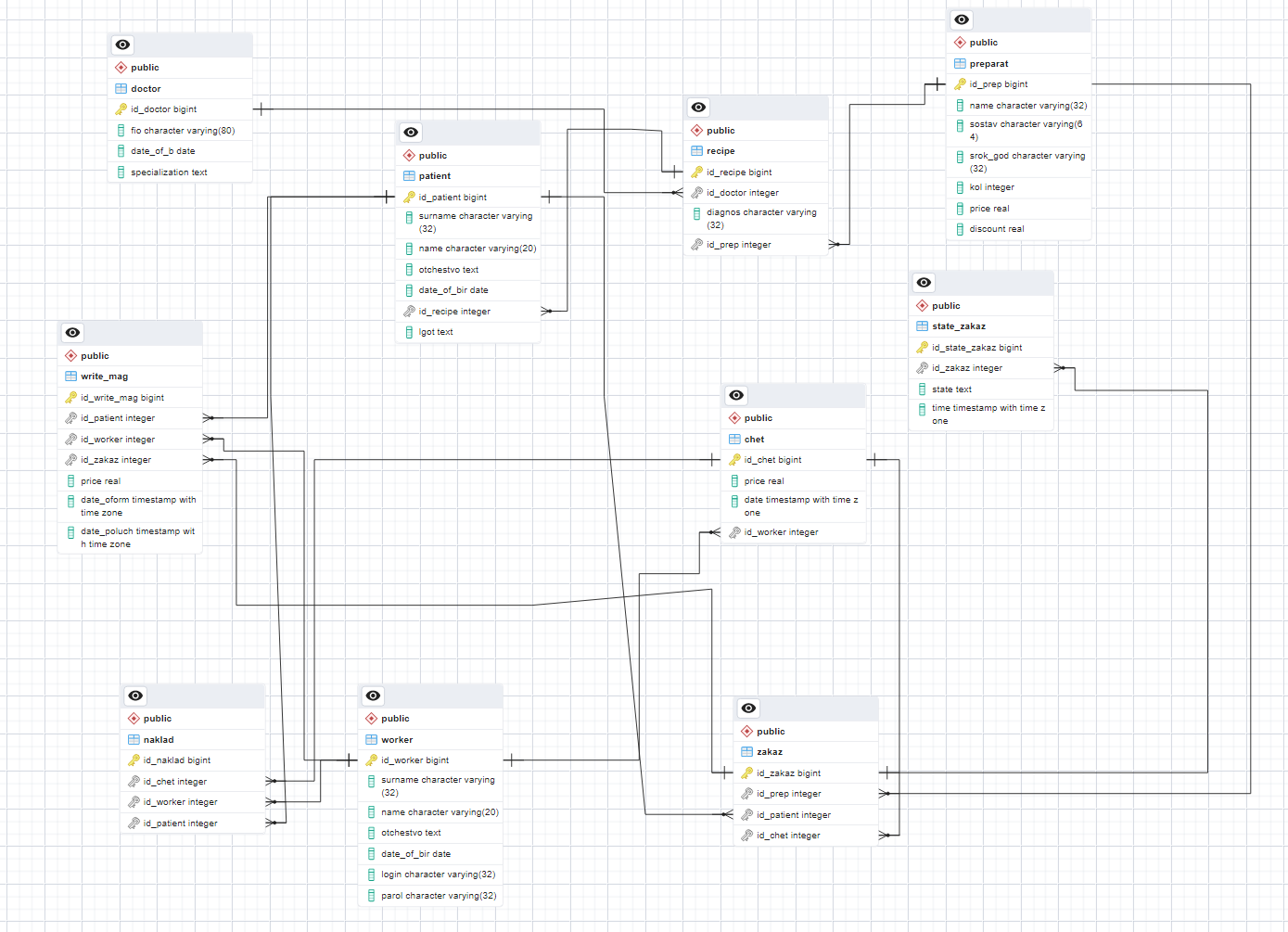


Рисунок 5 – Структура БД в PostgreSQL

Вывод: изучили программные средства для разработки моделей информационной базы ПС, проработали методы нормализации отношений в БД, приобрели навыки применения CASE-средств ERwin для моделирования базы данных ПС.