МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ Федеральное государственное бюджетное образовательное учреждение высшего образования

«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «КубГУ»)

**Факультет компьютерных технологий и прикладной математики Кафедра информационных технологий**

**ОТЧЕТ**

**по дисциплине**

**«Технологии проектирования программного обеспечения»**

**Лабораторная работа №6**

Работу выполнил: В.Л.Дурнев

Работу принял преподаватель: А.Н. Полетайкин, доцент кафедры ИТ

Тема: Проектирование базы данных программной системы.

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.

Индивидуальная тема: мессенджер социальной сети.

1. На основе модели классов UML, разработанной при выполнении

лабораторной работы №5, произвести идентификацию сущностей

информационной базы ПС и связей между ними:

- изучить массив нормативно-справочной (условно-постоянной)

информации, определить состав соответствующих справочников.

- изучить массив входной (текущей) информации, структурировать его

по ключевым сущностям с указанием всех атрибутов.

Как правило, документ с обычным "бумажными" таблицами

разбивается по принципу: одна "бумажная" таблица – одна сущность.

Соответственно, каждому входному документу, выделенному при

выполнении лабораторной работы №1, поставить в соответствие одну

сущность или (в результате нормализации) структуру связанных сущностей.

Количество сущностей в модели зависит от предметной области.

2. При помощи CASE-средства ERWin разработать ER-диаграмму

логической модели данных.

3. Выполнить сравнительный анализ полученной логической модели с

моделью классов, разработанной при выполнении лабораторной работы №5.

При необходимости скорректировать диаграмму классов.

4. Провести нормализацию сущностей логической модели данных и

разработать ER-диаграмму физической модели данных. Имена, атрибуты и

назначение сущностей физической модели данных привести в формате табл.

8. Сущности разделить на оперативные и справочные. При разработке

модели определить сущности, их первичные и внешние ключи и атрибуты, а

также связи между сущностями. Цель физического моделирования – это

таблицы в нормальных формах высшего, минимум, третьего (НФ3) порядка.

5. Средствами ERWin на основе физической модели данных выполнить

генерацию SQL-кода для создания реляционной базы данных ПС.

6. В соответствии с требованиями технического задания,

разработанного при выполнении лабораторной работы №3, провести

обоснованный выбор СУБД.

7. В выбранной СУБД развернуть БД, доработать её структуру с учетом

возможной нормализации отношений, а также доработать структуру таблиц с

учетом ограничений на значения полей. Сформировать ER-диаграмму БД.

8. Выполнить описание таблиц БД в формате табл. 9 с указанием для

всех полей типов данных выбранной СУБД. Столбцы «Условие на значение»,

«Значение по умолчанию» и «Примечание» заполнять при необходимости.

9. Выполнить краткое описание таблиц в формате табл. 10. Описание

связей в БД и условия целостности данных привести в виде табл. 11.

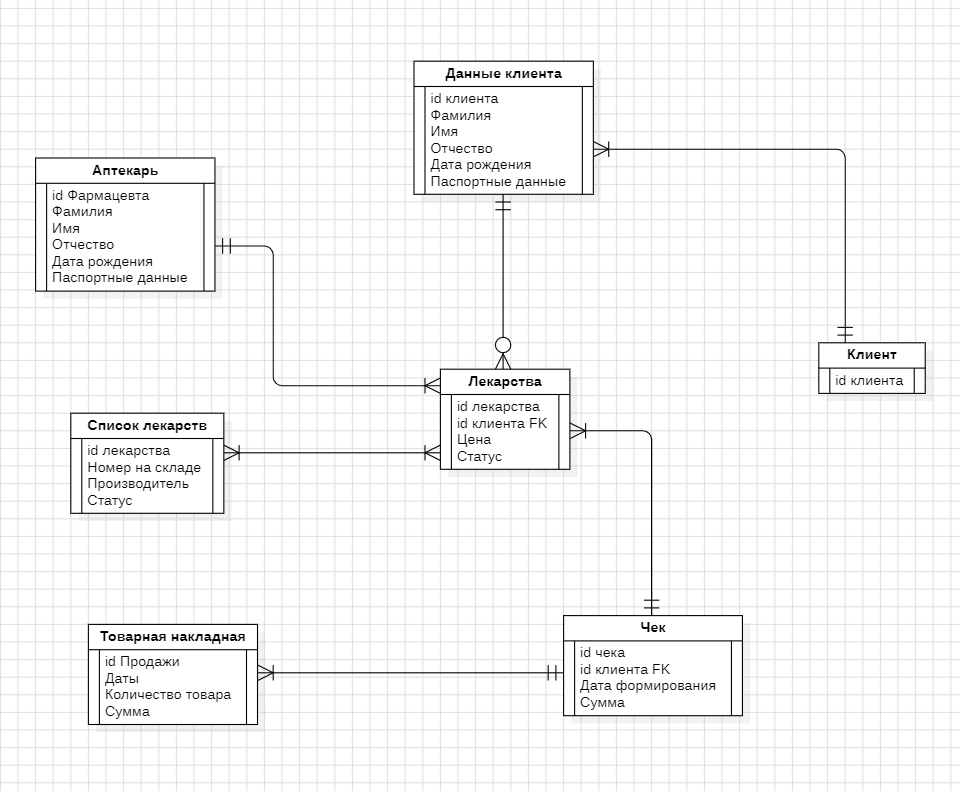
Допустимые типы связей «один-к-одному» и «один-ко-многим»

**Задача 1:**

Идентификация сущностей информационной базы ПС

|  |  |
| --- | --- |
| Документ | Сущность |
| Электронные лекарства | Лекарства |
| Фармацевт | Аптекарь |
| Клиенты | Клиент |
| Данные клиентов | Данные клиента |
| Чеки | Чек |
| Заказы | Заказы |
| Учет продажи | Учет продажи |

**Задача 2**



**Задача 3:**

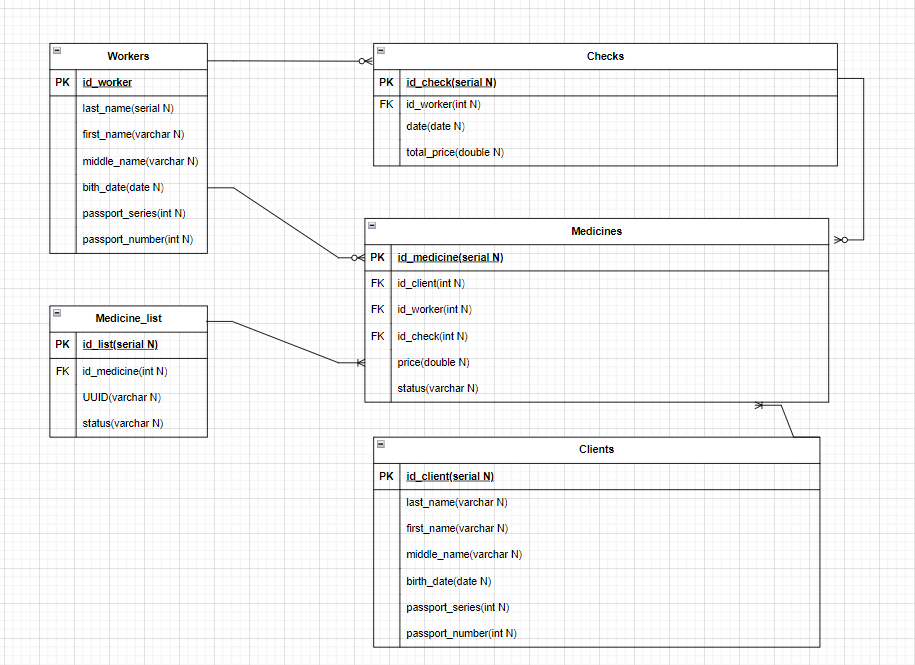
Исходя из построенной диаграммы классов и логической модели данных

можно сделать вывод, что была выполнена правильная проектировка

будущей базы данных. Была проведена нормализация данных. Сравнительный анализ с диаграммой классов показал на верную проектировку будущей БД. Была проведена нормализация данных.

**Задача 4, 7:**

Er-diagram





Сущности физической модели данных

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущность | Атрибуты | Описание |
| Справочные | | | |
| 1 | Фармацевт | ID фармацевта, фамилия, имя, отчество, дата рождения, паспортные данные | Информация о фармацевте |
| 2 | Лекарства | ID лекарства,цена, статус | Информация проданных лекарствах |
| 3 | Клиенты | ID клиента, фамилия, имя, отчество, дата рождения, паспортные данные | Информация о клиентах |
| Оперативные | | | |
| 4 | Список Лекарств | ID списка, ID лекарства, UUID,статус | Информация о лекартсвах |
| 5 | Чеки | ID чека, ID аптекаря, дата оформления, сумма | Информация о продажах |

**Задача 5:**

CREATE TABLE IF NOT EXISTS Workers(

id\_worker serial primary key,

last\_name varchar(100) NOT NULL,

first\_name varchar(100) NOT NULL,

middle\_name varchar(100),

birth\_date date NOT NULL,

passport\_series int NOT NULL,

passport\_number int NOT NULL

);

CREATE TABLE IF NOT EXISTS Clients(

id\_client serial primary key,

last\_name varchar(100) NOT NULL,

first\_name varchar(100) NOT NULL,

middle\_name varchar(100),

birth\_date date NOT NULL,

passport\_series int NOT NULL,

passport\_number int NOT NULL

);

CREATE TABLE IF NOT EXISTS Medicine\_list(

id\_list serial primary key,

id\_medicine int NOT NULL,

UUID varchar(100) NOT NULL,

status varchar(255) NOT NULL

);

CREATE TABLE IF NOT EXISTS Checks(

id\_check serial primary key,

id\_worker int NOT NULL references Workers(id\_worker) ON DELETE CASCADE,

date\_of\_formation date NOT NULL,

total\_price double precision default 0

);

CREATE TABLE IF NOT EXISTS Medicines(

id\_medicine serial primary key,

id\_client int NOT NULL references Clients(id\_client) ON DELETE CASCADE,

id\_worker int NOT NULL references Clients(id\_client) ON DELETE CASCADE,

id\_check int NOT NULL references Clients(id\_client) ON DELETE CASCADE,

price double precision NOT NULL,

status varchar(255) NOT NULL

);

**Задача 6:**

Для разработки была выбрана СУБД SQLite по следующим причинам:

1) Наличие удобного интерфейса при работе с БД

2) Бесплатная объектно-реляционная СУБД с открытым исходным кодом

3) Высокая мощность и широкая функциональность

4) Поддержка сложных запросов

**Задача 8:**

Таблица Clients

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Условие назначение | Значение по умолчанию | Примечание |
| Id\_client | serial | 4 |  |  |  |
| Last\_name | varchar | 6 |  |  |  |
| First\_name | varchar | 6 |  |  |  |
| Middle\_name | varchar | 6 |  |  |  |
| Passport\_series | integer | 4 |  |  |  |
| Passport\_number | integer | 4 |  |  |  |

Таблица Medicine\_list

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Условие назначение | Значение по умолчанию | Примечание |
| Id\_list | serial | 4 |  |  |  |
| Id\_medicine | integer | 4 |  |  |  |
| UUID | varchar | 6 |  |  |  |
| status | varchar | 6 |  |  |  |

Таблица Checks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Условие назначение | Значение по умолчанию | Примечание |
| Id\_check | serial | 4 |  |  |  |
| Id\_worker | integer | 4 |  |  |  |
| Date\_of\_formation | date | 4 |  |  |  |
| Total\_price | Double precision | 8 |  |  | 0 |

Таблица Workers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Условие назначение | Значение по умолчанию | Примечание |
| Id\_worker | serial | 4 |  |  |  |
| Last\_name | varchar | 6 |  |  |  |
| First\_name | varchar | 6 |  |  |  |
| Middle\_name | varchar | 6 |  |  |  |
| Passport\_series | integer | 4 |  |  |  |
| Passport\_number | integer | 4 |  |  |  |

Таблица Medicines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Условие назначение | Значение по умолчанию | Примечание |
| Id\_medicine | serial | 4 |  |  |  |
| Id\_client | integer | 4 |  |  |  |
| Id\_worker | integer | 4 |  |  |  |
| price | Double precision | 8 |  |  |  |
| status | varchar | 6 |  |  |  |

**Задача 9:**

Список разработанных таблиц

|  |  |  |
| --- | --- | --- |
| № | Имя таблицы | Описание |
| 1 | Clients | Описывает сущность клиента |
| 2 | Workers | Описывает сущность аптекаря |
| 3 | Medicines | Описывает сущность лекарств |
| 4 | Medicine\_list | Описывает сущность списка лекарств |
| 5 | Checks | Описывает сущность чеков |

Связи между таблицами БД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Родительская таблица | | Дочерняя таблица | | Тип связи |
| Название | Атрибут | Название | Атрибут |
| Medicine\_list | id\_medicine | Medicine | id\_medicine | Один-ко-многим |
| Workers | id\_worker | Medicines | id\_worker | Один-ко-многим |
| Checks | id\_check | Medicines | id\_check | Один-ко-многим |
| Clients | id\_client | Medicines | id\_client | Один-ко-многим |
| Workers | id\_worker | Checks | id\_worker | Один-ко-многим |