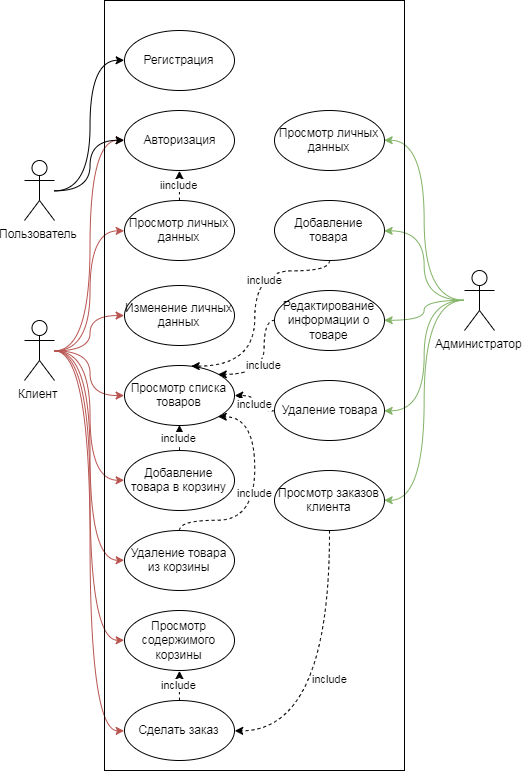
Выполним проектирование базы данных информационной системы для цветочного магазина.

Клиента при входе в электронный журнал встречает окно авторизации. Авторизовавшись при помощи логина и пароля, клиент перенаправляется на главную страницу магазина. При нажатии кнопки выхода из учетной записи клиент возвращается на страницу авторизации. Выбрав один из пунктов, таких как просмотр *ассортимента* или просмотр корзины, на стороне сервера произойдет загрузка информации в соответствии с выбранной операцией. Далее клиент может просматривать и изменять личные данные в соответствующем разделе личного кабинета.

Пользователь в роли администратора при входе также попадает на окно авторизации. Авторизовавшись при помощи логина и пароля, администратор перенаправляется на главную страницу, предназначенную для данного пользователя. Перед администратором появляется окно со списком ассортимента с функциями добавления нового цветка, удаления существующего цветка или же редактирования информации о существующем цветке. Заполнив соответствующие ячейки, администратор добавляет запись в базу данных о новом растении.

Из этого описания понятны основные функции системы, изображенные на рисунке с помощью нотации диаграммы прецедентов UML (смотрите рисунок 1).



*Рисунок 1- диаграмма вариантов использования*

На основе проведенного анализа предметной области были выделены следующие сущности:

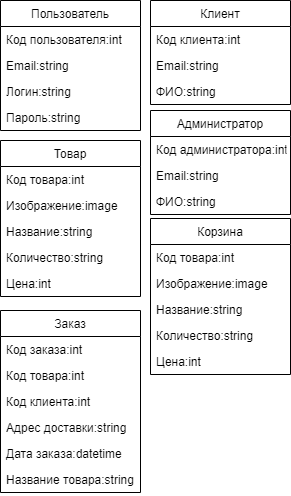
1. Пользователь – хранение сведений о логине и пароле;
2. Клиент – хранение данных покупателя;
3. Администратор – хранение данных администратора, ответственного за размещение, редактирование и удаление ассортимента;
4. Товар – ассортимент цветочного магазина;
5. Корзина – хранение понравившихся клиенту товаров;

В таблице 1 представлен список атрибутов каждой выделенной сущности:

Таблица 1 – Список атрибутов сущностей

|  |  |  |
| --- | --- | --- |
| **№** | **Сущность** | **Атрибут** |
| 1 | Пользователь | Код пользователя |
| Логин пользователя |
| Пароль пользователя |
| Email пользователя |
| 2 | Клиент | Код клиента |
| Email клиента |
| ФИО клиента |
| 3 | Администратор | Код администратора |
| Email администратора |
| ФИО администратора |
| 4 | Товар | Код товара |
| Изображение товара |
| Название товара |
| Количество товара |
| Цена товара |
| 5 | Корзина | Код товара |
| Название товара |
| Изображение товара |
| Количество товара |
| Цена товара |
| 6 | Заказ | Код заказа |
| Адрес доставки |
| Дата заказа |
| Код товара |
| Код клиента |
| Название товара |

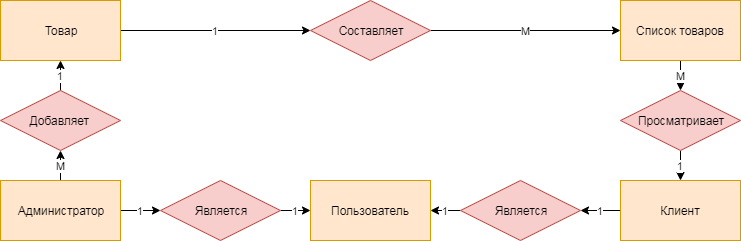
Основные сущности, информация о которых потребуются во время работы, показаны на рисунке, при этом используется нотация диаграммы классов UML. Каждый прямоугольник соответствует одной сущности, внутри записаны поля и типы данных.



*Рисунок 2 – основные сущности*

Каждая сущность, содержит поле id, которое идентифицирует объект.

Выше были отображены основные сущности, но не отображены роли пользователей, хотя их тоже должна хранить система. Они показаны ниже на ER-диаграмме в нотации Чена.

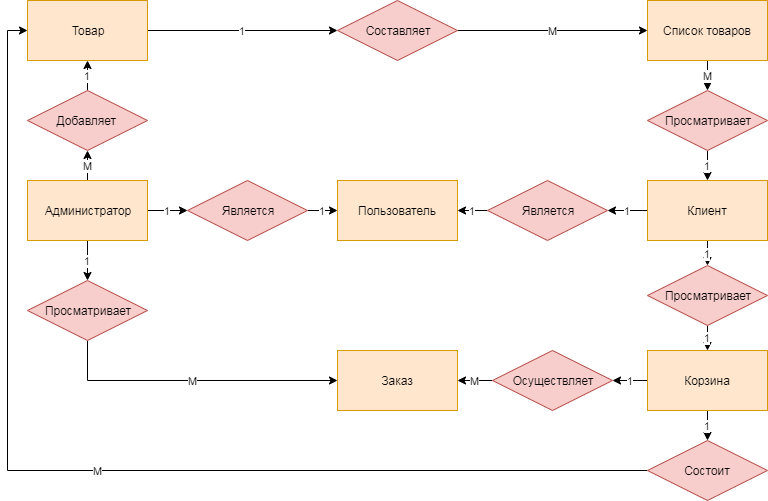


*Рисунок 3 – ER-диаграмма в нотации Чена*

На диаграмме выделены роли клиента и администратора, а также основные отношения между сущностями.

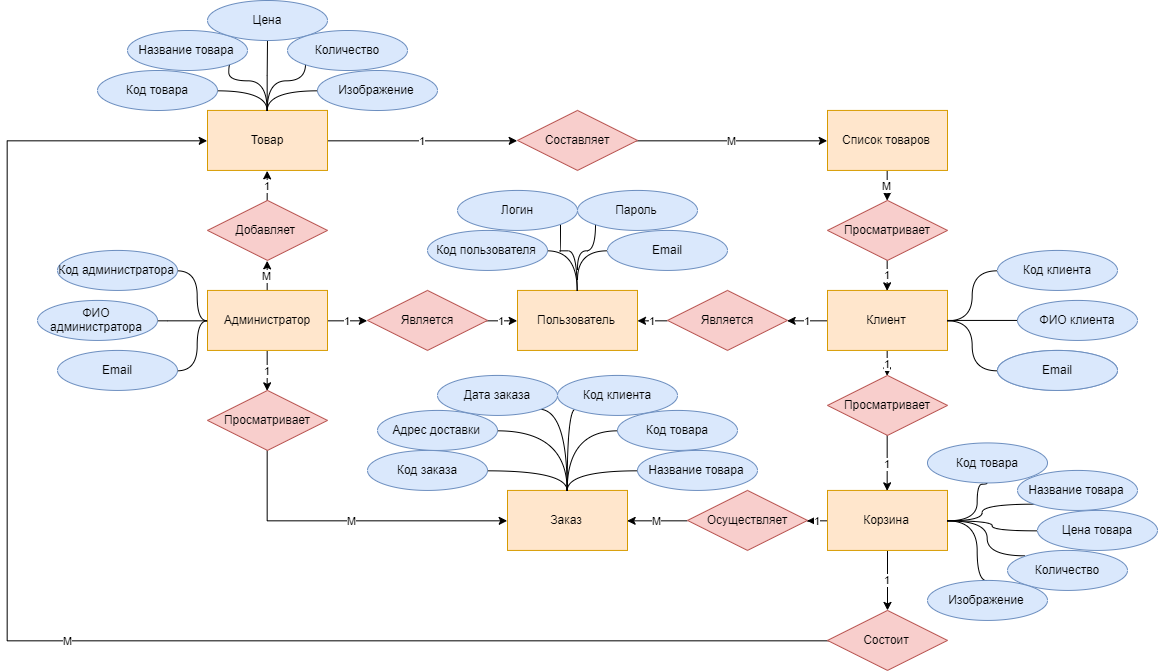
На диаграмме проставлены кратности связей.

Модифицированная ER-диаграмма представлена на рисунке 4.



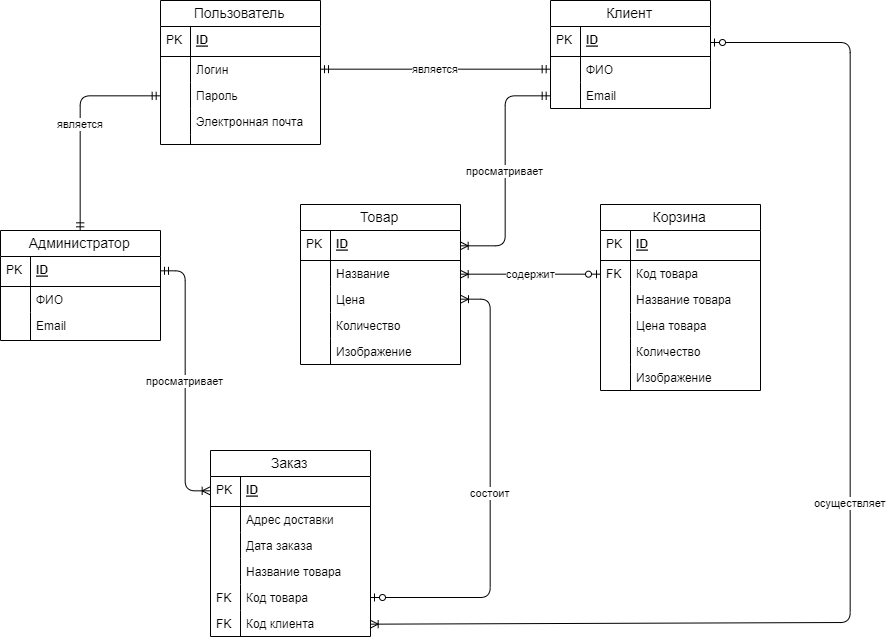
*Рисунок 4 – Модифицированная ER-диаграмма*

Для формирования схемы данных необходимо сначала дополнить ER-диаграмму реквизитами сущностей (уточнить ее) — результат приведен на рисунке 5.



*Рисунок 5 – ER-диаграмма с атрибутами в нотации Чена*

Перестроим ранее спроектированную диаграмму под нотацию Мартина. Результат представлен на рисунке 6.



*Рисунок 6 – ER-диаграмма с атрибутами в нотации Мартина*

Выполним нормализацию представленных в концептуальной диаграмме сущностей.

Выясним, находится ли наша схема данных в первой нормальной форме.

Первая нормальная форма предполагает, что таблица не должна содержать повторяющихся столбцов или таких столбцов, которые содержат наборы значений. Сформированная во втором задании модель данных соответствует данному условию.

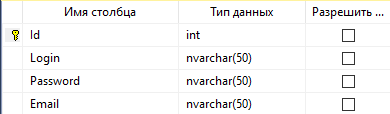
Далее выясним, находится ли наша модель во второй НФ. Во второй нормальной форме каждый столбец в таблице, который не является ключом, должен зависеть от ключа. Так как ни одна из таблиц не имеет составного первичного ключа можно сделать вывод, что модель базы данных приведена ко второй нормальной форме.

В таблицах также отсутствуют транзитивные зависимости, что говорит о соответствии модели данных третьей нормальной форме.

Выполним физическое проектирование базы данных и реализуем ее в СУБД Microsoft SQL Server.

Определим характеристику для каждого из атрибутов.

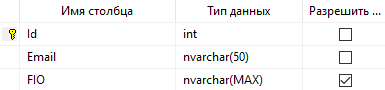
На рисунке 7 – 12 представлены таблицы сущностей с указанием типа данных каждого атрибута.



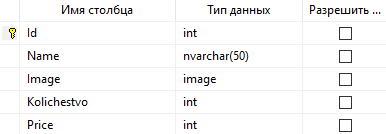
*Рисунок 7 – Сущность «Пользователь»*



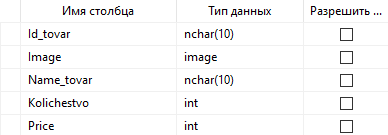
*Рисунок 8 – Сущность «Клиент»*



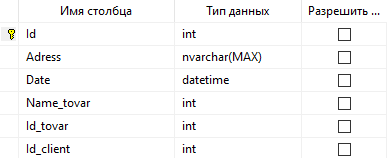
*Рисунок 9 – Сущность «Администратор»*



*Рисунок 10 – Сущность «Товар»*

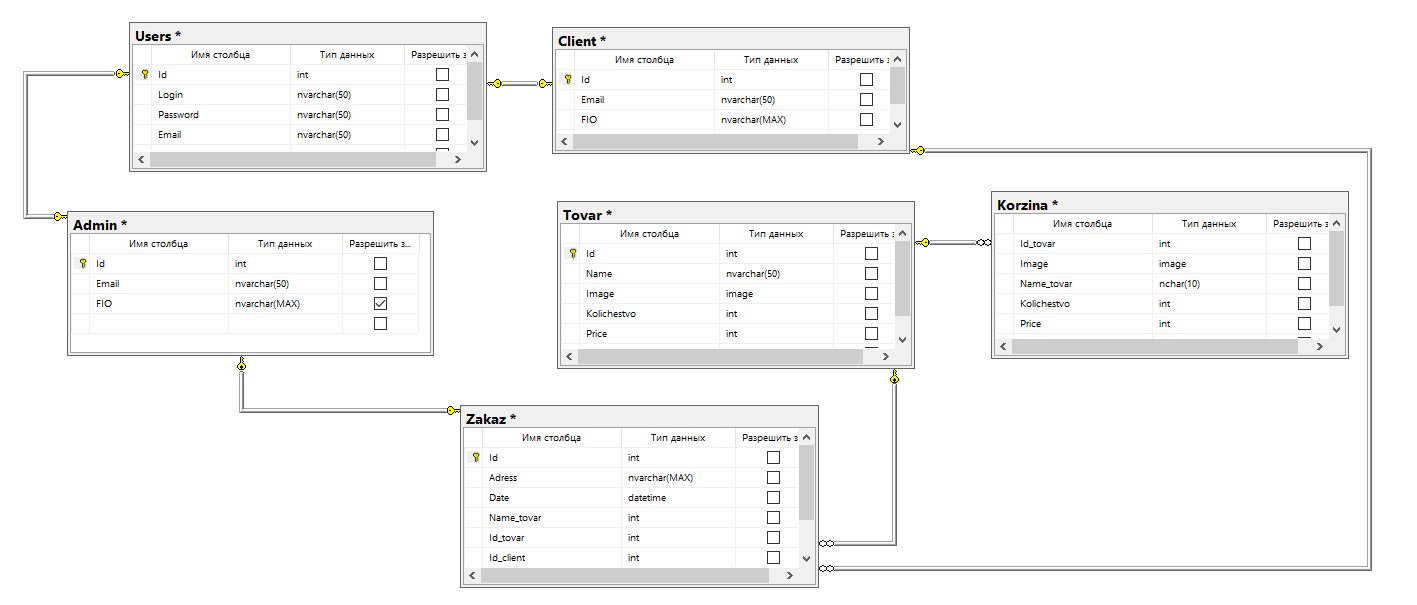


*Рисунок 11 – Сущность «Корзина»*



*Рисунок 12 – Сущность «Заказ»*

Схема созданной базы данных с установленными связями представлена на рисунке 13.



*Рисунок 13 – Разработанная схема данных*