**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра веб-технологий и компьютерного моделирования

ДАНИЛОВА Ксения Денисовна РЕДЬКИН Максим Витальевич

**Проектирование базы данных интернет-магазин**

студентов II курса 2 группы специальности   
1-31 03 08 Математика и информационные технологии (по направлениям)

Руководитель

Кремень Юрий Алексеевич

Минск, 2023

1. **ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ИНТЕРНЕТ-МАГАЗИН**

**1.1.1. ПОСТАНОВКА ЗАДАЧИ**

Предположим, необходимо разработать систему для отслеживания финансовой составляющей работы компании в соответствии со следующим описанием.

* На Интернет-сайте компании представлены (выставлены на продажу) некоторые товары.
* Каждый из товаров имеет некоторое название, цену и единицу измерения (штуки, килограммы, литры).
* Определяющее значение имеют стандартные анкетные данные, а также телефон и адрес электронной почты для связи.
* В случае приобретения товаров на сумму свыше 5000 р. клиент переходит в категорию «постоянных клиентов» и получает скидку на каждую покупку в размере 2%.
* По каждому факту продажи автоматически фиксируется клиент, товары, количество, дата продажи, дата доставки.
  + 1. **ИНИЦИИРОВАНИЕ ПРОЕКТА (ФАЗА 0)**

Допустим, что на этой фазе были определены все участники проекта, определены их функции. Уточним цели проектирования.

В результате предварительного изучения поставленной задачи определены требования к разрабатываемой системе:

* Необходимо отслеживать какие товары пользуются спросом и популярностью и исходя из этих данных делать соответствующую подборку рекомендованных товаров новым клиентом.
* Все товары разбиты на группы, для более удобного поиска.
* Для удобства клиентов каждый товар должен иметь соответствующее описание, содержащее цену, единицу измерения и краткую информацию о товаре.
* Необходимо отслеживать какими товарами интересуется конкретный клиент и исходя из этих данных делать персональную подборку рекомендованных товаров.
* Клиент может добавлять товары в «корзину» на некоторое время, если сомневается в осуществлении покупки.
* Необходимо напоминать клиентам о программе перехода в категорию «постоянных клиентов».
* Клиент должен иметь возможность отслеживать общую сумму выкупа.

Сформулированные требования определяют исходные данные для всех остальных фаз проектирования системы.

* + 1. **ОПРЕДЕЛЕНИЕ МНОЖЕСТВ СУЩНОСТЕЙ (ФАЗА 1)**
       1. **Выделение множеств сущностей**

В соответствии с описанием предметной области можно выделить следующие множества сущностей:

*Клиенты*

*Постоянные клиенты*

*Товары*

*Единицы измерения*

*Категории*

*Заказы*

*Товары в заказе*

*Оплаты*

*Продажа*

* + - 1. **Описание множеств сущностей**

Описание выделенных множеств сущностей приведено в табл. 1.

Таблица 1. Пул сущностей

|  |  |  |
| --- | --- | --- |
| **Имя множества сущностей** | **Определение множества сущностей** | **Описание множества сущностей** |
| Клиенты | Физические лица, намеревающиеся совершать покупку. | Клиенты заходят в интернет-магазин, изучают список товаров, осуществляют заказы. |
| Постоянные клиенты | Клиенты, которые приобрели товаров на сумму свыше 5000 р., получающие скидку на каждую покупку в размере 2%. | Чтобы иметь возможность перейти в категорию «постоянные клиенты» нужно иметь статус клиента. |
| Товары | Объекты, которые представлены в списке товаров, имеют соответствующее описание и могут быть приобретены клиентами. | Товары представлены в списке товаров, разбиты на категории и имеют единицы измерения. |
| Единицы измерения | То, в чем измеряются товары. | Единицы измерения: штуки, килограммы, литры. |
| Категории | Группы товаров, которые воспринимаются клиентами как взаимосвязанные и взаимозаменяемые. | Категории помогают клиентам быстрее найти нужные товары. |
| Заказы | Запросы, отправляемые клиентами на покупку товара, выбранного в интернет-магазине, и его доставку, по указанному в запросе адресу. | Информация о заказе:  данные клиента, заказанные товары, их количество, дата оформления заказа, ожидаемая дата доставки. |
| Товары в заказе | Товар, находящийся в заказе. | Товар, находящийся в заказе. |
| Оплаты | Осуществление платежа за заказ. | Оплаты осуществляются банковской платежной картой. |
| Продажа | Процесс покупки товара клиентом (продажи продавцом). | По каждому факту продажи автоматически фиксируется клиент, товары, количество, дата продажи, дата доставки. |

* + 1. **ПОСТРОЕНИЕ МОДЕЛИ УРОВНЯ СУЩНОСТЕЙ (ФАЗА 2)**
       1. **Матрица связей**

На основании анализа постановки задачи устанавливаются связи между множествами сущностей. Матрица связей приведена в табл. 2.

Таблица 2. Матрица связей

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Клиенты** | **Постоянные клиенты** | **Товары** | **Единицы измерения** | **Категории** | **Заказы** | **Товары в заказе** | **Оплаты** | **Продажа** |
| **Клиенты** |  | x |  |  |  | x |  |  | x |
| **Постоянные клиенты** | x |  |  |  |  |  |  |  |  |
| **Товары** |  |  |  | x | x |  | x |  | x |
| **Единицы измерения** |  |  | x |  |  |  |  |  |  |
| **Категории** |  |  |  |  |  |  |  |  |  |
| **Заказы** | x |  |  |  |  |  | x | x |  |
| **Товары в заказе** |  |  | x |  |  | x |  |  |  |
| **Оплаты** |  |  |  |  |  | x |  |  |  |
| **Продажа** | x |  | x |  |  |  |  |  |  |

* + - 1. **Описание связей**

Описание выделенных связей приводится в табл. 3.

Таблица 3. Описание связей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1-я сущность** | **2-я сущность** | **Имя связи** | **Тип связи** | **Мощность связи** | **Описание связи** |
| Клиенты | Постоянные клиенты | Могут находить в категории | идент. опред. | Z | В случае приобретения товаров на сумму свыше 5000 р. клиент переходит в категорию «постоянных клиентов» и получает скидку на каждую покупку в размере 2%. |
| Клиенты | Заказы | Совершают | обязат. неидент. опред. | P | При покупке товаров клиенты осуществляют заказы. *Один* клиент может осуществить *один и более* заказов. |
| Клиенты | Продажа | Участвуют в | обязат. неидент. опред. | P | Товары фиксируются по факту продажи. *Один* клиент может учувствовать в *одной и более* продажах. |
| Товары | Единицы измерения | Имеют | обязат. неидент. опред. | P | Каждый товар имеет единицу измерения (штуки, килограммы, литры). *Один и более* товаров могут иметь *одну* единицу измерения. |
| Товары | Категории | Содержатся в | обязат. неидент. неопред. | P, B | *Один и более* товаров могут находиться в *одной и более* категориях. |
| Товары | Товары в заказе | Содержатся в | необязат. неидент. опред. | P | Товар может содержаться только в *одном* заказе. В *одном* заказе может содержаться *один и более* товаров. |
| Товары | Продажа | Осуществляют | необязат. неидент. опред. | P | Товары фиксируются по факту продажи товары. *Один* товар может учувствовать в *одной и более* продажах. |
| Товары в заказе | Заказы | Составляют | обязат. неидент. опред. | P | Заказы состоят из товаров. В *одном* заказе могут быть *один и более* товаров. |
| Заказы | Оплаты | Способствуют | обязат. неидент. опред. | P | Заказы оплачиваются банковской платежной картой. |

* + - 1. **Концептуальная схема уровня сущностей**

В соответствии с определенными множествами сущностей и связями между ними строится концептуальная схема уровня сущностей (рис 1):



* + 1. **ПОСТРОЕНИЕ МОДЕЛИ УРОВНЯ КЛЮЧЕЙ (ФАЗА 3)**
       1. **Разрешение неопределенных связей**

Прежде всего, выполним разрешение неопределенных связей (типа "многие ко многим"). В рассматриваемом примере одна такая связь. В этом случае вводится дополнительная сущность(таблица) Товары-Категории, описание которой представлено в дополнении (табл. 4) к таблице пул сущностей̆ (табл. 1). Соответственно, вносятся необходимые изменения в определение связей: добавляются новые связи, уточняются типы связей (табл. 5).

Таблица 4. Пул сущностей

|  |  |  |
| --- | --- | --- |
| **Имя множества сущностей** | **Определение множества сущностей** | **Описание множества сущностей** |
| Товары-Категории | Информация об id товара и id категории товаров. | Каждый товар имеет свой уникальный id, Каждая категория товаров имеет свой уникальный id. Id определяют какие товары относятся к каким категориям товаров, эти связи представлены в новой введенной сущности. |

Таблица 5. Определение связей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1-я сущность** | **2-я сущность** | **Имя связи** | **Тип связи** | **Мощность связи** | **Описание связи** |
| Клиенты | Постоянные клиенты | Могут находить в категории | идент. опред. | Z | В случае приобретения товаров на сумму свыше 5000 р. клиент переходит в категорию «постоянных клиентов» и получает скидку на каждую покупку в размере 2%. |
| Клиенты | Заказы | Совершают | обязат. неидент. опред. | P | При покупке товаров клиенты осуществляют заказы. *Один* клиент может осуществить *один и более* заказов. |
| Клиенты | Продажа | Участвуют в | обязат. неидент. опред. | P | Товары фиксируются по факту продажи. *Один* клиент может учувствовать в *одной и более* продажах. |
| Товары | Единицы измерения | Имеют | обязат. неидент. опред. | P | Каждый товар имеет единицу измерения (штуки, килограммы, литры). *Один и более* товаров могут иметь *одну* единицу измерения. |
| Товары | Товары-Категории | Содержатся в | идент. неопред. | B | *Каждый* товар может находиться в *одной и более* категориях или *ни в одной* категории. |
| Товары | Товары в заказе | Содержатся в | необязат. неидент. опред. | P | Товар может содержаться только в *одном* заказе. В *одном* заказе может содержаться *один и более* товаров. |
| Товары | Продажа | Осуществляют | необязат. неидент. опред. | P | Товары фиксируются по факту продажи товары. *Один* товар может учувствовать в *одной и более* продажах. |
| Категории | Товары-Категории | Составляют | идент. опред. | P | Товары делятся на категории. В *одной* категории может быть *один и более* товаров. |
| Товары в заказе | Заказы | Составляют | обязат. неидент. опред. | P | Заказы состоят из товаров. В *одном* заказе могут быть *один и более* товаров. |
| Заказы | Оплаты | Способствуют | обязат. неидент. опред. | P | Заказы оплачиваются банковской платежной картой. |

* + - 1. **Определение ключевых атрибутов и доменов**

Определение ключевых атрибутов осуществляется одновременно с определением доменов, на которых определяются эти атрибуты. Сначала из анализа задачи для каждого множества сущностей̆ выделяются ключи (простые или составные), позволяющие идентифицировать экземпляры сущностей̆. В случае если ключей̆ несколько, один выбирается в качестве первичного ключа, а остальные определяются как альтернативные. Связи между множествами сущностей̆ представляются с помощью внешних ключей̆. Таким образом, определяются все ключи. Для каждого атрибута определяется домен.

В рассматриваемом примере для выделенных сущностей̆ можно предложить следующие ключи и домены:

**КЛИЕНТЫ**

Первичный ключ: в качестве первичного ключа удобно выделить id клиента, который будем называть customer\_id. Все клиенты различны и каждый имеет свой собственный id. Атрибут определим на типе данных int.

**ПОСТОЯННЫЕ КЛИЕНТЫ**

Первичный ключ: в качестве первичного ключа удобно выделить id постоянного клиента, который будем называть regular\_id. Все постоянные клиенты различны и каждый имеет свой собственный id. Атрибут определим на типе данных int.

Внешний ключ: так как данное множество сущностей связано с множеством сущностей КЛИЕНТЫ, эту связь можно реализовать при помощи атрибута – customer\_id, определенного на типе int. Т.е. атрибут customer\_id будет являться внешним ключом к таблице КЛИЕНТЫ.

**ТОВАРЫ**

Первичный ключ: в качестве первичного ключа удобно выделить id товара, который будем называть product\_id. Все товары различны и каждый имеет свой собственный id. Атрибут определим на типе данных int.

Внешний ключ: так как данное множество сущностей связано с множеством сущностей ЕДИНИЦЫ ИЗМЕРЕНИЯ, эту связь можно реализовать при помощи атрибута – unit\_of\_measurement\_id, определенного на типе int. Т.е. атрибут unit\_of\_measurement\_id будет являться внешним ключом к таблице ЕДИНИЦЫ ИЗМЕРЕНИЯ.

**ЕДИНИЦЫ ИЗМЕРЕНИЯ**

Первичный ключ: в качестве первичного ключа удобно выделить id единицы измерения, который будем называть unit\_of\_measurement\_id. Все единицы измерения различны, и каждая имеет свой собственный id. Атрибут определим на типе данных int.

**КАТЕГОРИИ**

Первичный ключ: в качестве первичного ключа удобно выделить id категории товаров, который будем называть category\_id. Все категории различны, и каждая имеет свой собственный id. Атрибут определим на типе данных int.

**ТОВАРЫ-КАТЕГОРИИ**

Первичный ключ: целесообразно определить составной первичный ключ из двух атрибутов. В качестве первичных ключей используются id товара и id категории товаров, которые называются product\_id и category\_id. Все товары различны, и каждый имеет свой собственный id. Все категории различны, и каждая имеет свой собственный id Атрибуты определены на типе данных int.

Внешний ключ: так как данное множество сущностей связано с множеством сущностей ТОВАРЫ, эту связь можно реализовать при помощи атрибута – product\_id, определенного на типе int. Т.е. атрибут product\_id будет входить как в состав первичного ключа, так и являться внешним ключом к таблице ТОВАРЫ.

**ЗАКАЗЫ**

Первичный ключ: в качестве первичного ключа удобно выделить id заказа, который будем называть order\_id. Все заказы различны и каждый имеет свой собственный id. Атрибут определим на типе данных int.

Внешний ключ: так как данное множество сущностей связано с множеством сущностей КЛИЕНТЫ, эту связь можно реализовать при помощи атрибута – customer\_id, определенного на типе int. Т.е. атрибут customer\_id будет являться внешним ключом к таблице КЛИЕНТЫ.

**ТОВАРЫ В ЗАКАЗЕ**

Первичный ключ: в качестве первичного ключа выделяется id товара в заказе, который будем называть order\_item\_id. Атрибут определим на типе данных int.

Внешний ключ: так как данное множество сущностей связано с множествами сущностей ЗАКАЗЫ и ТОВАРЫ, эти связи можно реализовать при помощи атрибутов – order\_id и product\_id, определенных на типе int. Т.е. атрибут order\_id будет являться внешним ключом к таблице ЗАКАЗЫ, а product\_id будет являться внешним ключом к таблице ТОВАРЫ.

**ОПЛАТЫ**

Первичный ключ: в качестве первичного ключа удобно выделить id оплаты, который будем называть payment\_id. Все оплаты различны, и каждая имеет свой собственный id. Атрибут определим на типе данных int.

Внешний ключ: так как данное множество сущностей связано с множеством сущностей ЗАКАЗЫ, эту связь можно реализовать при помощи атрибута – order\_id, определенного на типе int. Т.е. атрибут order\_id будет являться внешним ключом к таблице ЗАКАЗЫ.

**ПРОДАЖА**

Первичный ключ: в качестве первичного ключа удобно выделить id продажи, который будем называть sale\_id. Все продажи различны, и каждая имеет свой собственный id. Атрибут определим на типе данных int.

Внешний ключ: так как данное множество сущностей связано с множествами сущностей КЛИЕНТЫ и ТОВАРЫ, эти связи можно реализовать при помощи атрибутов – customer\_id и product\_id, определенных на типе int. Т.е. атрибут customer\_id будет являться внешним ключом к таблице КЛИЕНТЫ, а product\_id будет являться внешним ключом к таблице ТОВАРЫ.

Описание ключевых атрибутов приведено в табл. 6.

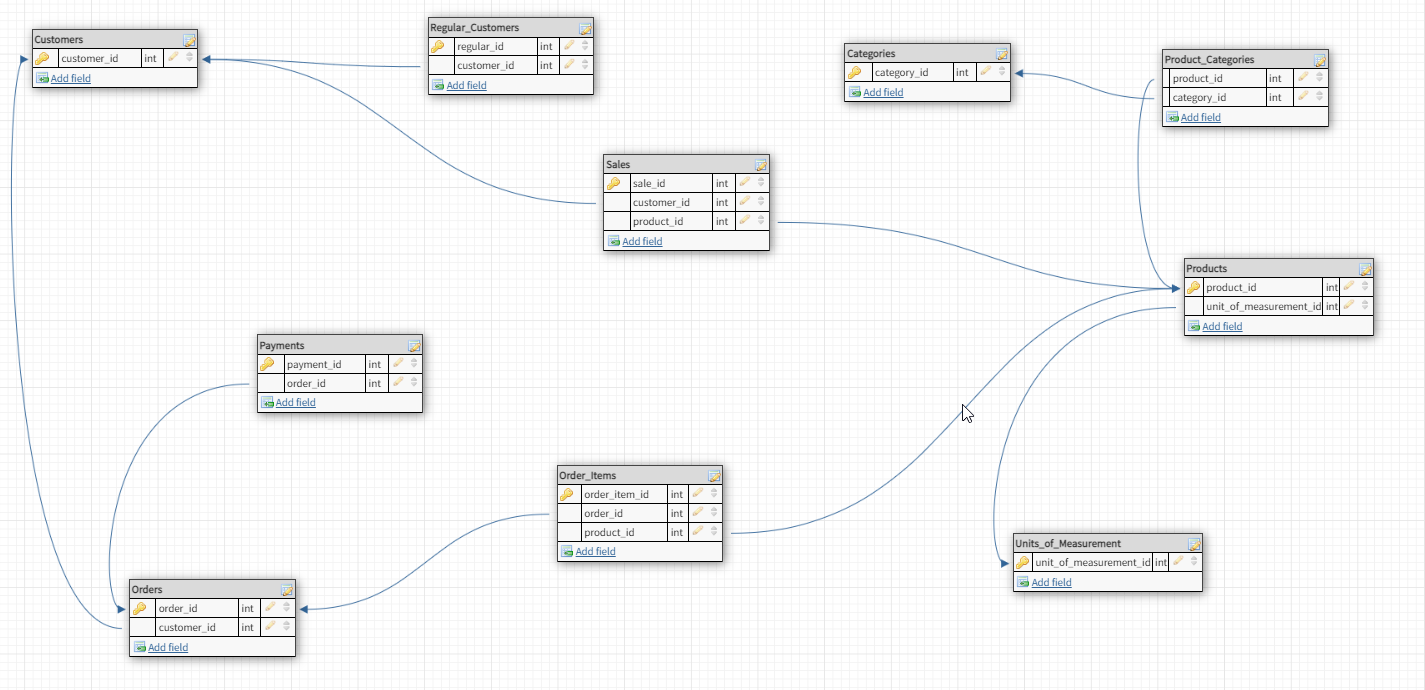
Таблица 6. Описание атрибутов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Множество сущностей** | **Имя атрибута** | **Тип данных** | **Признак обязательности** | **Примечание** |
| Клиенты | customer\_id | int | Not null | PK |
| Постоянные клиенты | regular\_id | int | Not null | PK |
| customer\_id | int | Not null | FK |
| Товары | product\_id | int | Not null | PK |
| unit\_of\_measurement\_id | int | Not null | FK |
| Единицы измерения | unit\_of\_measurement\_id | int | Not null | PK |
| Категории | category\_id | int | Not null | PK |
| Товары-Категории | product\_id | int | Not null | PK1.1, FK |
| category\_id | int | Not null | PK1.2 |
| Заказы | order\_id | int | Not null | PK |
| customer\_id | int | Not null | FK |
| Товары в заказе | order\_item\_id | int | Not null | PK |
| order\_id | int | Not null | FK1.1 |
| product\_id | int | Not null | FK1.2 |
| Оплаты | payment\_id | int | Not null | PK |
| order\_id | int | Not null | FK |
| Продажа | sale\_id | int | Not null | PK |
| customer\_id | int | Not null | FK1.1 |
| product\_id | int | Not null | FK1.2 |

* + - 1. **Концептуальная схема**

По результатам определения первичных ключей строится концептуальная схема уровня ключей (рис. 2).

Рис. 2. Концептуальная схема уровня ключей



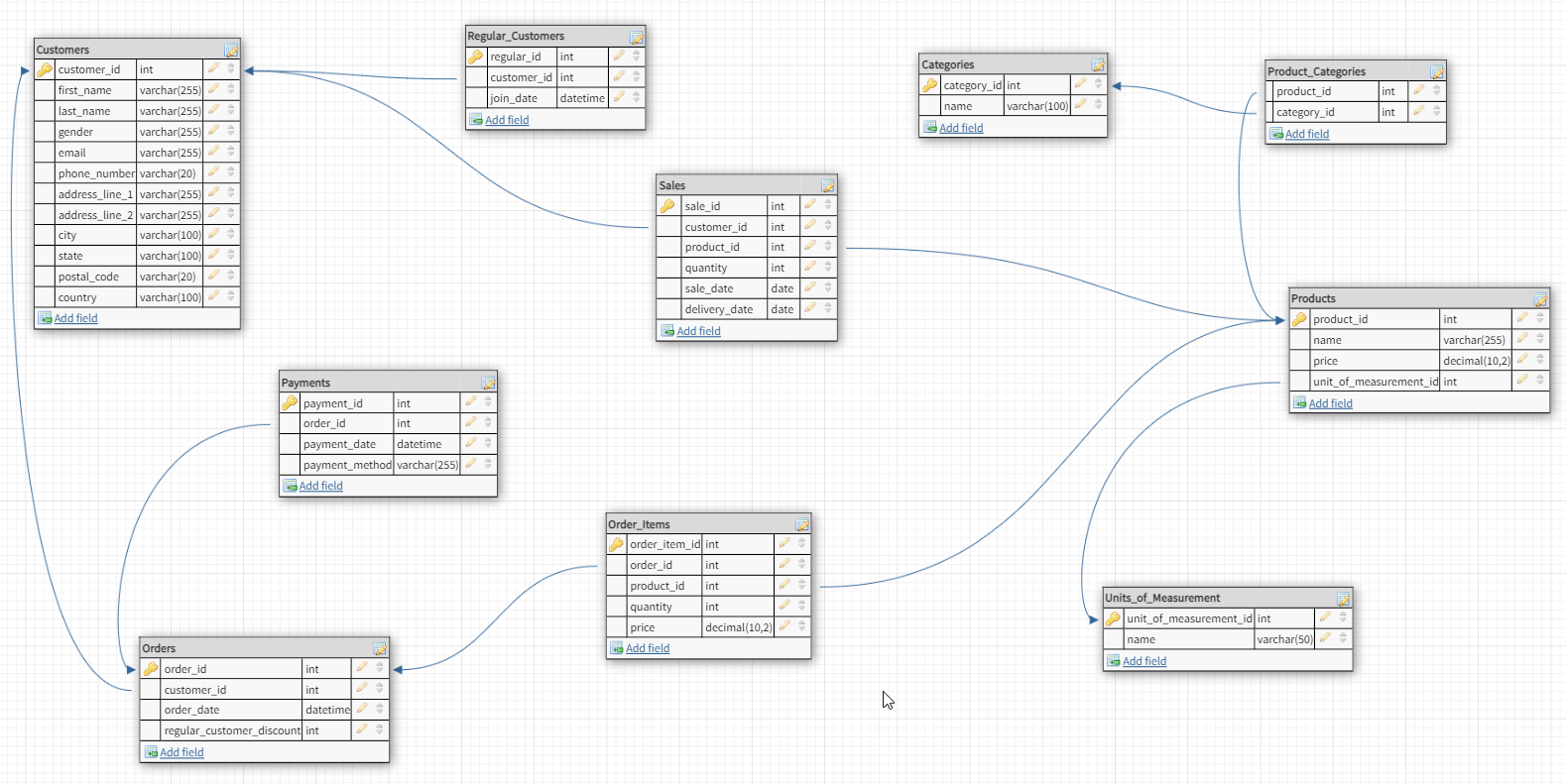
* + 1. **ПОСТРОЕНИЕ ПОЛНОАТРИБУТНОЙ МОДЕЛИ (ФАЗА 4)**

На этом этапе в соответствии с проведенным анализом предметной области определяются все необходимые атрибуты и домены, на которых эти атрибуты определены. Соответствующим образом модифицируются таблица описания доменов и атрибутов (табл. 7) и строится полноатрибутная концептуальная схема (рис. 3).

Таблица 7. Описание атрибутов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Множество сущностей** | **Имя атрибута** | **Тип данных** | **Признак обязательности** | **Примечание** |
| Клиенты | customer\_id | int | Not null | PK |
| first\_name | varchar(255) | Not null |  |
| last\_name | varchar(255) | Not null |  |
| gender | varchar(255) | Null |  |
| email | varchar(255) | Not null |  |
| phone\_number | varchar(20) | Null |  |
| address\_line\_1 | varchar(255) | Null |  |
| address\_line\_2 | varchar(255) | Null |  |
| city | varchar(100) | Not null |  |
| state | varchar(100) | Not null |  |
| postal\_code | varchar(20) | Not null |  |
| country | varchar(100) | Not null |  |
| Постоянные клиенты | regular\_id | int | Not null | PK |
| customer\_id | int | Not null | FK |
| join\_date | datetime | Not null |  |
| Товары | product\_id | int | Not null | PK |
| unit\_of\_measurement\_id | int | Not null | FK |
| name | varchar(255) | Not null |  |
| price | decimal(10, 2) | Not null |  |
| Единицы измерения | unit\_of\_measurement\_id | int | Not null | PK |
| name | varchar(50) | Not null |  |
| Категории | category\_id | int | Not null | PK |
| name | varchar(50) | Not null |  |
| Товары-Категории | product\_id | int | Not null | PK1.1, FK |
| category\_id | int | Not null | PK1.2 |
| Заказы | order\_id | int | Not null | PK |
| customer\_id | int | Not null | FK |
| order\_date | datetime | Not null |  |
| regular\_customer\_discount | int | Not null |  |
| Товары в заказе | order\_item\_id | int | Not null | PK |
| order\_id | int | Not null | FK1.1 |
| product\_id | int | Not null | FK1.2 |
| quantity | int | Not null |  |
| price | decimal(10, 2) | Not null |  |
| Оплаты | payment\_id | int | Not null | PK |
| order\_id | int | Not null | FK |
| payment\_date | datetime | Not null |  |
| payment\_method | varchar(255) | Not null |  |
| Продажа | sale\_id | int | Not null | PK |
| customer\_id | int | Not null | FK1.1 |
| product\_id | int | Not null | FK1.2 |
|  | quantity | int | Not null |  |
|  | sale\_date | date | Not null |  |
|  | delivery\_date | date | Not null |  |

Рис. 4. Полноатрибутная концептуальная схема



**1.1.7. ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ**

Создадим базу данных Онлайн Магазин / Online\_Store.

USE master;

GO

DROP DATABASE IF EXISTS Online\_Store;

GO

CREATE DATABASE Online\_Store;

GO

USE Online\_Store;

GO

В соответствии с приведенной структурой, внутренняя схема разрабатываемой базы данных представлена в табл. 8 – 18.

Таблица 8. Клиенты / Customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код клиента | customer\_id | int | Not null | PK |
| Имя | first\_name | varchar(255) | Not null |  |
| Фамилия | last\_name | varchar(255) | Not null |  |
| Пол | gender | varchar(255) | Null |  |
| Электронная почта | email | varchar(255) | Not null |  |
| Номер телефона | phone\_number | varchar(20) | Null |  |
| Первая адресная строка | address\_line\_1 | varchar(255) | Null |  |
| Вторая адресная строка | address\_line\_2 | varchar(255) | Null |  |
| Город | city | varchar(100) | Not null |  |
| Область | state | varchar(100) | Not null |  |
| Почтовый код | postal\_code | varchar(20) | Not null |  |
| Страна | country | varchar(100) | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Customers] (

customer\_id int NOT NULL,

first\_name varchar(255) NOT NULL,

last\_name varchar(255) NOT NULL,

gender varchar(255),

email varchar(255) NOT NULL UNIQUE,

phone\_number varchar(20),

address\_line\_1 varchar(255),

address\_line\_2 varchar(255),

city varchar(100) NOT NULL,

state varchar(100) NOT NULL,

postal\_code varchar(20) NOT NULL,

country varchar(100) NOT NULL,

CONSTRAINT [PK\_CUSTOMERS] PRIMARY KEY CLUSTERED

(

[customer\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

Таблица 9. Единицы измерения / Units\_of\_Measurement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код единицы измерения | unit\_of\_measurement\_id | int | Not null | PK |
| Название | name | varchar(50) | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Units\_of\_Measurement] (

unit\_of\_measurement\_id int NOT NULL UNIQUE,

name varchar(50) NOT NULL UNIQUE,

CONSTRAINT [PK\_UNITS\_OF\_MEASUREMENT] PRIMARY KEY CLUSTERED

(

[unit\_of\_measurement\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

Таблица 10. Товары / Products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код товара | product\_id | int | Not null | PK |
| Название | name | varchar(255) | Not null |  |
| Цена | price | decimal(10, 2) | Not null |  |
| Код единицы измерения | unit\_of\_measurement\_id | int | Not null | FK |

Описание таблицы на SQL:

CREATE TABLE [Products] (

product\_id int NOT NULL UNIQUE,

name varchar(255) NOT NULL,

price decimal(10,2) NOT NULL CHECK (price > 0),

unit\_of\_measurement\_id int NOT NULL,

CONSTRAINT [PK\_PRODUCTS] PRIMARY KEY CLUSTERED

(

[product\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

ALTER TABLE [Products] WITH CHECK ADD CONSTRAINT [Products\_fk0] FOREIGN KEY ([unit\_of\_measurement\_id]) REFERENCES [Units\_of\_Measurement]([unit\_of\_measurement\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Products] CHECK CONSTRAINT [Products\_fk0]

GO

Таблица 11. Постоянные клиенты / Regular\_Customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код постоянного клиента | regular\_id | int | Not null | PK |
| Код клиента | customer\_id | int | Not null | FK |
| Дата получения статуса | join\_date | datetime | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Regular\_Customers] (

regular\_id int NOT NULL UNIQUE,

customer\_id int NOT NULL,

join\_date datetime NOT NULL,

CONSTRAINT [PK\_REGULAR\_CUSTOMERS] PRIMARY KEY CLUSTERED

(

[regular\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

ALTER TABLE [Regular\_Customers] WITH CHECK ADD CONSTRAINT [Regular\_Customers\_fk0] FOREIGN KEY ([customer\_id]) REFERENCES [Customers]([customer\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Regular\_Customers] CHECK CONSTRAINT [Regular\_Customers\_fk0]

GO

Таблица 12. Категории / Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код категории | category\_id | int | Not null | PK |
| Название | name | varchar(50) | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Categories] (

category\_id int NOT NULL UNIQUE,

name varchar(100) NOT NULL UNIQUE,

CONSTRAINT [PK\_CATEGORIES] PRIMARY KEY CLUSTERED

(

[category\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

Таблица 13. Товары-Категории / Product\_Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код товара | product\_id | int | Not null | PK1.1, FK |
| Код категории | category\_id | int | Not null | PK1.2 |

Описание таблицы на SQL:

CREATE TABLE [Product\_Categories] (

product\_id int NOT NULL,

category\_id int NOT NULL

)

GO

ALTER TABLE [Product\_Categories] WITH CHECK ADD CONSTRAINT [Product\_Categories\_fk0] FOREIGN KEY ([product\_id]) REFERENCES [Products]([product\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Product\_Categories] CHECK CONSTRAINT [Product\_Categories\_fk0]

GO

ALTER TABLE [Product\_Categories] WITH CHECK ADD CONSTRAINT [Product\_Categories\_fk1] FOREIGN KEY ([category\_id]) REFERENCES [Categories]([category\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Product\_Categories] CHECK CONSTRAINT [Product\_Categories\_fk1]

GO

Таблица 14. Заказы / Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код заказа | order\_id | int | Not null | PK |
| Код клиента | customer\_id | int | Not null | FK |
| Дата заказа | order\_date | datetime | Not null |  |
| Скидка постоянному клиенту | regular\_customer\_discount | int | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Orders] (

order\_id int NOT NULL UNIQUE,

customer\_id int NOT NULL,

order\_date datetime NOT NULL,

regular\_customer\_discount INT CHECK (regular\_customer\_discount IN (0,2)) NOT NULL DEFAULT 0,

CONSTRAINT [PK\_ORDERS] PRIMARY KEY CLUSTERED

(

[order\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

ALTER TABLE [Orders] WITH CHECK ADD CONSTRAINT [Orders\_fk0] FOREIGN KEY ([customer\_id]) REFERENCES [Customers]([customer\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Orders] CHECK CONSTRAINT [Orders\_fk0]

GO

Таблица 15. Товары в заказе / Order\_Items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код товара в заказе | order\_item\_id | int | Not null | PK |
| Код заказа | order\_id | int | Not null | FK1.1 |
| Код товара | product\_id | int | Not null | FK1.2 |
| Количество товаров | quantity | int | Not null |  |
| Цена | price | decimal(10, 2) | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Order\_Items] (

order\_item\_id int NOT NULL UNIQUE,

order\_id int NOT NULL,

product\_id int NOT NULL,

quantity int NOT NULL,

price decimal(10,2) NOT NULL CHECK (price > 0),

CONSTRAINT [PK\_ORDER\_ITEMS] PRIMARY KEY CLUSTERED

(

[order\_item\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

ALTER TABLE [Order\_Items] WITH CHECK ADD CONSTRAINT [Order\_Items\_fk0] FOREIGN KEY ([order\_id]) REFERENCES [Orders]([order\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Order\_Items] CHECK CONSTRAINT [Order\_Items\_fk0]

GO

ALTER TABLE [Order\_Items] WITH CHECK ADD CONSTRAINT [Order\_Items\_fk1] FOREIGN KEY ([product\_id]) REFERENCES [Products]([product\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Order\_Items] CHECK CONSTRAINT [Order\_Items\_fk1]

GO

Таблица 16. Оплаты / Payments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код оплаты | payment\_id | int | Not null | PK |
| Код заказа | order\_id | int | Not null | FK |
| Дата оплаты | payment\_date | datetime | Not null |  |
| Способ оплаты | payment\_method | varchar(255) | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Payments] (

payment\_id int NOT NULL UNIQUE,

order\_id int NOT NULL,

payment\_date datetime NOT NULL,

payment\_method varchar(255) CHECK (payment\_method IN ('cash', 'card', 'coupon')) NOT NULL DEFAULT 'cash',

CONSTRAINT [PK\_PAYMENTS] PRIMARY KEY CLUSTERED

(

[payment\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

ALTER TABLE [Payments] WITH CHECK ADD CONSTRAINT [Payments\_fk0] FOREIGN KEY ([order\_id]) REFERENCES [Orders]([order\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Payments] CHECK CONSTRAINT [Payments\_fk0]

GO

Таблица 17. Продажа / Sales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код продажи | sale\_id | int | Not null | PK |
| Код клиента | customer\_id | int | Not null | FK1.1 |
| Код товара | product\_id | int | Not null | FK1.2 |
| Количество | quantity | int | Not null |  |
| Дата продажи | sale\_date | date | Not null |  |
| Дата доставки | delivery\_date | date | Not null |  |

Описание таблицы на SQL:

CREATE TABLE [Sales] (

sale\_id int NOT NULL UNIQUE,

customer\_id int NOT NULL,

product\_id int NOT NULL,

quantity int NOT NULL,

sale\_date date NOT NULL,

delivery\_date date NOT NULL,

CONSTRAINT [PK\_SALES] PRIMARY KEY CLUSTERED

(

[sale\_id] ASC

) WITH (IGNORE\_DUP\_KEY = OFF)

)

GO

ALTER TABLE [Sales] WITH CHECK ADD CONSTRAINT [Sales\_fk0] FOREIGN KEY ([customer\_id]) REFERENCES [Customers]([customer\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Sales] CHECK CONSTRAINT [Sales\_fk0]

GO

ALTER TABLE [Sales] WITH CHECK ADD CONSTRAINT [Sales\_fk1] FOREIGN KEY ([product\_id]) REFERENCES [Products]([product\_id])

ON UPDATE CASCADE

GO

ALTER TABLE [Sales] CHECK CONSTRAINT [Sales\_fk1]

GO

Заполняем таблицу клиентов (Customers):

INSERT INTO [Customers] (customer\_id, first\_name, last\_name, gender, email, phone\_number, address\_line\_1, address\_line\_2, city, state, postal\_code, country)

VALUES (1, 'John', 'Doe', 'Male', 'john.doe@gmail.com', '555-1234', '123 Main St', 'Apt 2B', 'New York', 'NY', '10001', 'USA'),

(2, 'Jane', 'Doe', 'Female', 'jane.doe@mail.ru', '555-5678', '456 Elm St', NULL, 'Los Angeles', 'CA', '90001', 'USA'),

(3, 'Bob', 'Smith', 'Male', 'bob.smith@gmail.com', '555-2468', '789 Oak St', NULL, 'San Francisco', 'CA', '94101', 'USA'),

(4, 'Alice', 'Johnson', 'Female', 'alice.johnson@gmail.com', '555-3691', '321 Pine St', NULL, 'Chicago', 'IL', '60601', 'USA'),

(5, 'Tom', 'Brown', 'Male', 'tom.brown@mail.ru', '162-74-73', '55 Lenina St', NULL, 'Homel', 'HR', '331051', 'Belarus'),

(6, 'Valery', 'Brown', 'Chair', 'valery.brown@mail.ru', '771-81-75', '55 Lenina St', NULL, 'Homel', 'HR', '331051', 'Belarus'),

(7, 'Borya', 'Bandit', 'Female', 'borya2007.bn@gmail.com', '583-3213', '89 National St', NULL, 'Gdynia', 'GV', '781031', 'Poland'),

(8, 'John', 'Rogus', 'Male', 'john.rod@gmail.com', '545-1898', '789 Valenca St', NULL, 'San Francisco', 'CA', '42323', 'USA'),

(9, 'David', 'Johnson', 'Male', 'david.johnson@yahoo.com', '555-9876', '789 Oak St', NULL, 'San Francisco', 'CA', '94102', 'USA'),

(10, 'Sarah', 'Garcia', 'Female', 'sarah.garcia@hotmail.com', '555-4321', '1010 Pine St', 'Unit 5D', 'Seattle', 'WA', '98101', 'USA'),

(11, 'Michael', 'Brown', 'Male', 'michael.brown@gmail.com', '555-5555', '555 Elm St', NULL, 'Boston', 'MA', '02110', 'USA'),

(12, 'Emily', 'Davis', 'Female', 'emily.davis@gmail.com', '555-6789', '222 Maple St', 'Apt 10', 'Los Angeles', 'CA', '90001', 'USA'),

(13, 'William', 'Wilson', 'Male', 'william.wilson@yahoo.com', '555-2468', '333 Oak St', NULL, 'Atlanta', 'GA', '30303', 'USA'),

(14, 'Elizabeth', 'Anderson', 'Female', 'elizabeth.anderson@hotmail.com', '555-3698', '444 Park Ave', 'Suite 2B', 'Houston', 'TX', '77002', 'USA'),

(15, 'James', 'Jackson', 'Male', 'james.jackson@gmail.com', '555-7539', '555 Main St', NULL, 'Philadelphia', 'PA', '19102', 'USA'),

(16, 'Emma', 'Lee', 'Female', 'emma.lee@gmail.com', '555-8642', '666 Maple St', 'Apt 12', 'Dallas', 'TX', '75201', 'USA'),

(17, 'Christopher', 'Perez', 'Male', 'christopher.perez@yahoo.com', '555-3210', '777 Oak St', NULL, 'Miami', 'FL', '33101', 'USA'),

(18, 'Ava', 'Taylor', 'Female', 'ava.taylor@gmail.com', '555-9876', '888 Pine St', 'Unit 3C', 'Phoenix', 'AZ', '85001', 'USA'),

(19, 'Daniel', 'Thomas-More', 'Male', 'daniel.thomasM@hotmail.com', '555-5555', '999 Elm St', NULL, 'Denver', 'CO', '80202', 'USA'),

(20, 'Sophia', 'Moore', 'Female', 'sophia.moore@gmail.com', '555-6789', '111 Maple St', 'Apt 7', 'Portland', 'OR', '97201', 'USA'),

(21, 'Daniel', 'Thomas', 'Male', 'daniel.thomas@hotmail.com', '555-5555', '999 Elm St', NULL, 'Denver', 'CO', '80202', 'USA'),

(22, 'Sophia', 'Moon', 'Female', 'sophia.moon@gmail.com', '555-6789', '111 Maple St', 'Apt 7', 'Portland', 'OR', '97201', 'USA'),

(23, 'Benjamin', 'Martin', 'Male', 'benjamin.martin@yahoo.com', '555-2468', '222 Oak St', NULL, 'New Orleans', 'LA', '70112', 'USA'),

(24, 'Isabella', 'Hernandez', 'Female', 'isabella.hernandez@gmail.com', '555-3698', '333 Park Ave', 'Suite 10A', 'San Diego', 'CA', '92101', 'USA'),

(25, 'Mason', 'Lopez', 'Male', 'mason.lopez@hotmail.com', '555-7539', '444 Elm St', NULL, 'Las Vegas', 'NV', '89101', 'USA'),

(26, 'Mia', 'Gonzalez', 'Female', 'mia.gonzalez@gmail.com', '555-8642', '555 Maple St', 'Apt 9', 'Austin', 'TX', '78701', 'USA'),

(27, 'Ethan', 'Rodriguez', 'Male', 'ethan.rodriguez@yahoo.com', '555-3210', '666 Oak St', NULL, 'Charlotte', 'NC', '28202', 'USA'),

(28, 'Chloe', 'Parker', 'Female', 'chloe.parker@hotmail.com', '555-9876', '777 Pine St', 'Unit 4D', 'Nashville', 'TN', '37201', 'USA');

Заполняем таблицу единиц измерения (Units\_Of\_Measurement):

INSERT INTO [Units\_of\_Measurement] (unit\_of\_measurement\_id, name)

VALUES (1, 'Each'),

(2, 'Pound'),

(3, 'Ounce'),

(4, 'Gallon'),

(5, 'Quart'),

(6, 'Liter'),

(7, 'Dozen');

Заполняем таблицу категорий (Categories):

INSERT INTO [Categories] (category\_id, name)

VALUES (1, 'Fruit'),

(2, 'Vegetables'),

(3, 'Meat'),

(4, 'Dairy'),

(5, 'Bakery'),

(6, 'Snacks'),

(7, 'Beverages'),

(8, 'Frozen Foods'),

(9, 'Canned Goods'),

(10, 'Seafood'),

(11, 'Pasta'),

(12, 'Condiments'),

(13, 'Candy'),

(14, 'Nuts'),

(15, 'Produce'),

(16, 'Spices');

Заполняем таблицу товаров (Products):

INSERT INTO [Products] (product\_id, name, price, unit\_of\_measurement\_id)

VALUES (1, 'Apple', 0.50, 1),

(2, 'Banana', 0.25, 1),

(3, 'Carrot', 1.00, 2),

(4, 'Milk', 2.50, 4),

(5, 'Bread', 1.50, 1),

(6, 'Chips', 3.00, 1),

(7, 'Soda', 1.00, 6),

(8, 'Potato', 5.00, 2),

(9, 'Yogurt', 1.20, 3),

(10, 'Cookie', 9.00, 7),

(11, 'Frozen Pizza', 4.99, 7),

(12, 'Canned Tuna', 1.75, 2),

(13, 'Salmon Fillet', 10.99, 7),

(14, 'Spaghetti', 2.49, 5),

(15, 'Ketchup', 1.29, 3),

(16, 'Bagels', 3.50, 1),

(17, 'Almonds', 6.99, 4),

(18, 'Oranges', 0.75, 1),

(19, 'Steak', 12.99, 2),

(20, 'Water', 0.50, 6);

Заполняем промежуточную таблицу категорий и товаров (Products\_Categories):

INSERT INTO [Product\_Categories] (product\_id, category\_id)

VALUES (1, 1),

(2, 1),

(3, 2),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 2),

(9, 4),

(10, 5),

(11, 10),

(12, 16),

(13, 4),

(14, 11),

(15, 2),

(16, 5),

(17, 13),

(18, 1),

(19, 3),

(20, 14);

Заполняем таблицу постоянных клиентов (Regular\_Customers):

INSERT INTO [Regular\_Customers] (regular\_id, customer\_id, join\_date)

VALUES (1, 1, '2021-01-01'),

(2, 6, '2021-01-01'),

(3, 4, '2021-01-01'),

(4, 12, '2021-01-01'),

(5, 7, '2021-01-01'),

(6, 9, '2021-01-01');

Заполняем таблицу заказов (Orders):

INSERT INTO [Orders] (order\_id, customer\_id, order\_date, regular\_customer\_discount)

VALUES (1, 1, '2022-01-01', (SELECT CASE WHEN EXISTS(SELECT 1 FROM [Regular\_Customers] WHERE customer\_id = 1) THEN 2 ELSE 0 END)),

(2, 6, '2023-02-01', (SELECT CASE WHEN EXISTS(SELECT 1 FROM [Regular\_Customers] WHERE customer\_id = 6) THEN 2 ELSE 0 END)),

(3, 3, '2022-03-01', (SELECT CASE WHEN EXISTS(SELECT 1 FROM [Regular\_Customers] WHERE customer\_id = 3) THEN 2 ELSE 0 END)),

(4, 4, '2022-04-01', (SELECT CASE WHEN EXISTS(SELECT 1 FROM [Regular\_Customers] WHERE customer\_id = 4) THEN 2 ELSE 0 END));

Заполняем таблицу товаров заказов (Order\_Items):

INSERT INTO [Order\_Items] (order\_item\_id, order\_id, product\_id, quantity, price)

VALUES (1, 2, 1, 500, (SELECT price FROM [Products] WHERE product\_id = 1) \* 500),

(2, 1, 7, 6, (SELECT price FROM [Products] WHERE product\_id = 7) \* 6),

(3, 1, 3, 1, (SELECT price FROM [Products] WHERE product\_id = 3) \* 1),

(4, 1, 4, 5, (SELECT price FROM [Products] WHERE product\_id = 4) \* 5),

(5, 3, 1, 1, (SELECT price FROM [Products] WHERE product\_id = 1) \* 1),

(6, 3, 2, 8, (SELECT price FROM [Products] WHERE product\_id = 2) \* 8),

(7, 4, 6, 3, (SELECT price FROM [Products] WHERE product\_id = 6) \* 3),

(8, 4, 5, 10, (SELECT price FROM [Products] WHERE product\_id = 5) \* 10),

(9, 4, 9, 56, (SELECT price FROM [Products] WHERE product\_id = 9) \* 56);

Заполняем таблицу оплат заказов (Payments):

INSERT INTO [Payments] (payment\_id, order\_id, payment\_date, payment\_method)

VALUES (1, 1, (SELECT order\_date FROM [Orders] WHERE order\_id = 1), 'cash'),

(2, 2, (SELECT order\_date FROM [Orders] WHERE order\_id = 2), 'card'),

(3, 3, (SELECT order\_date FROM [Orders] WHERE order\_id = 3), 'coupon'),

(4, 4, (SELECT order\_date FROM [Orders] WHERE order\_id = 4), 'cash');

Заполняем таблицу продаж (Sales):

INSERT INTO [Sales] (sale\_id, customer\_id, product\_id, quantity, sale\_date, delivery\_date)

VALUES (1, 6, 1, 500, (SELECT order\_date FROM [Orders] WHERE customer\_id = 6), '2023-02-01'),

(2, 1, 7, 6, (SELECT order\_date FROM [Orders] WHERE customer\_id = 1), '2022-01-02'),

(3, 1, 3, 1, (SELECT order\_date FROM [Orders] WHERE customer\_id = 1), '2022-01-02'),

(4, 1, 4, 5, (SELECT order\_date FROM [Orders] WHERE customer\_id = 1), '2022-01-02'),

(5, 3, 1, 1, (SELECT order\_date FROM [Orders] WHERE customer\_id = 3), '2022-03-06'),

(6, 3, 2, 8, (SELECT order\_date FROM [Orders] WHERE customer\_id = 3), '2022-03-06'),

(7, 4, 6, 3, (SELECT order\_date FROM [Orders] WHERE customer\_id = 4), '2022-04-01'),

(8, 4, 5, 10, (SELECT order\_date FROM [Orders] WHERE customer\_id = 4), '2022-04-01'),

(9, 4, 9, 56, (SELECT order\_date FROM [Orders] WHERE customer\_id = 4), '2022-04-01');

Продемонстрируем три примера возникновения ошибок при вставке в таблицу Orders:

1. Конфликт инструкции INSERT с ограничением PRIMARY\_KEY

INSERT INTO [Orders] (order\_id, customer\_id, order\_date, regular\_customer\_discount)

VALUES (1, 1, '2022-01-01', (SELECT CASE WHEN EXISTS(SELECT 1 FROM [Regular\_Customers] WHERE customer\_id = 1) THEN 2 ELSE 0 END));

(1, 2, '2022-01-01', (SELECT CASE WHEN EXISTS(SELECT 1 FROM [Regular\_Customers] WHERE customer\_id = 2) THEN 2 ELSE 0 END));

-- Конфликт с ограничениями PRIMARY KEY, поскольку заказы с таким id уже существуют

2. Конфликт инструкции INSERT с ограничениями FOREIGN KEY “Orders\_fk0”

INSERT INTO [Orders] (order\_id, customer\_id, order\_date, regular\_customer\_discount)

VALUES (13, 45, '2022-01-01', (SELECT CASE WHEN EXISTS(SELECT 1 FROM [Regular\_Customers] WHERE customer\_id = 45) THEN 2 ELSE 0 END));

-- Конфликт инструкции INSERT с ограничением FOREIGN KEY "Orders\_fk0". Конфликт произошел в базе данных "OnlineStore10", таблица "dbo.Customers", column 'customer\_id', поскольку клиента с таким id не существует.

3. Конфликт инструкции INSERT с ограничениями CHECK ”CK\_\_Orders\_\_regular\_\_\_37A5467C”

INSERT INTO [Orders] (order\_id, customer\_id, order\_date, regular\_customer\_discount)

VALUES (13, 2, '2022-01-01', 45);

-- Конфликт инструкции INSERT с ограничением CHECK "CK\_\_Orders\_\_regular\_\_\_37A5467C". Конфликт произошел в базе данных "OnlineStore10", таблица "dbo.Orders", column 'regular\_customer\_discount', поскольку допустимые значения этого поля – значения 0 и 2.

Пример успешной вставки строки в таблицу Orders:

INSERT INTO [Orders] (order\_id, customer\_id, order\_date, regular\_customer\_discount)

VALUES (5, 8, '2022-01-01', (SELECT CASE WHEN EXISTS(SELECT 1 FROM [Regular\_Customers] WHERE customer\_id = 8) THEN 2 ELSE 0 END));