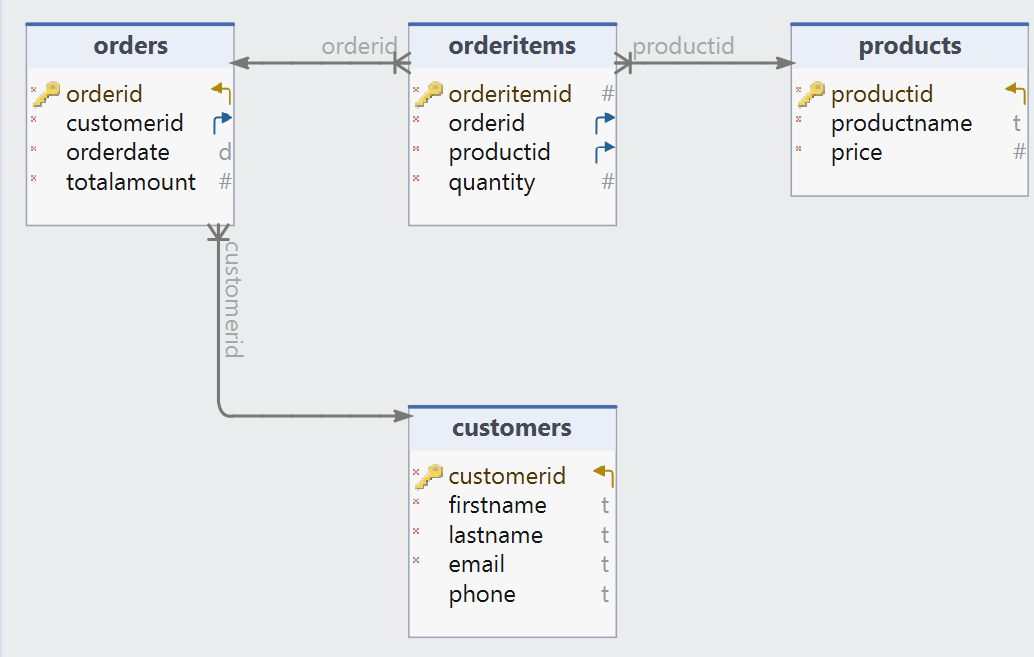
**Лабораторная работа по теме: Реализация структуры Data Vault**

Выполнил: Студент группы 203-321 Хандамов Амирхан Алексеевич

1. Третья нормальная форма



**Таблица Customers**:

* Хранит информацию о клиентах, включая имя, фамилию, email и телефон.
* Уникальный идентификатор клиента (CustomerID) служит первичным ключом.

**Таблица Orders**:

* Содержит информацию о заказах, включая дату заказа и общую сумму.
* Внешний ключ CustomerID связывает заказы с клиентами.

**Таблица Products**:

* Хранит информацию о товарах, включая название и цену.
* Уникальный идентификатор товара (ProductID) служит первичным ключом.

**Таблица OrderItems**:

* Связывает заказы с товарами, указывая, какие товары были заказаны и в каком количестве.
* Внешние ключи OrderID и ProductID связывают записи с таблицами заказов и товаров соответственно.

Код создания:

CREATE TABLE Customers (

CustomerID BIGINT PRIMARY KEY, -- Уникальный идентификатор клиента

FirstName VARCHAR(100) NOT NULL, -- Имя клиента

LastName VARCHAR(100) NOT NULL, -- Фамилия клиента

Email VARCHAR(100) NOT NULL, -- Email клиента

Phone VARCHAR(20) NULL -- Телефон клиента

);

CREATE TABLE Orders (

OrderID BIGINT PRIMARY KEY, -- Уникальный идентификатор заказа

CustomerID BIGINT NOT NULL, -- Внешний ключ на таблицу Customers

OrderDate TIMESTAMP NOT NULL, -- Дата заказа

TotalAmount DECIMAL(18, 2) NOT NULL, -- Общая сумма заказа

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) -- Связь с таблицей клиентов

);

CREATE TABLE Products (

ProductID BIGINT PRIMARY KEY, -- Уникальный идентификатор товара

ProductName VARCHAR(100) NOT NULL, -- Название товара

Price DECIMAL(18, 2) NOT NULL -- Цена товара

);

CREATE TABLE OrderItems (

OrderItemID BIGINT PRIMARY KEY, -- Уникальный идентификатор записи о товаре в заказе

OrderID BIGINT NOT NULL, -- Внешний ключ на таблицу Orders

ProductID BIGINT NOT NULL, -- Внешний ключ на таблицу Products

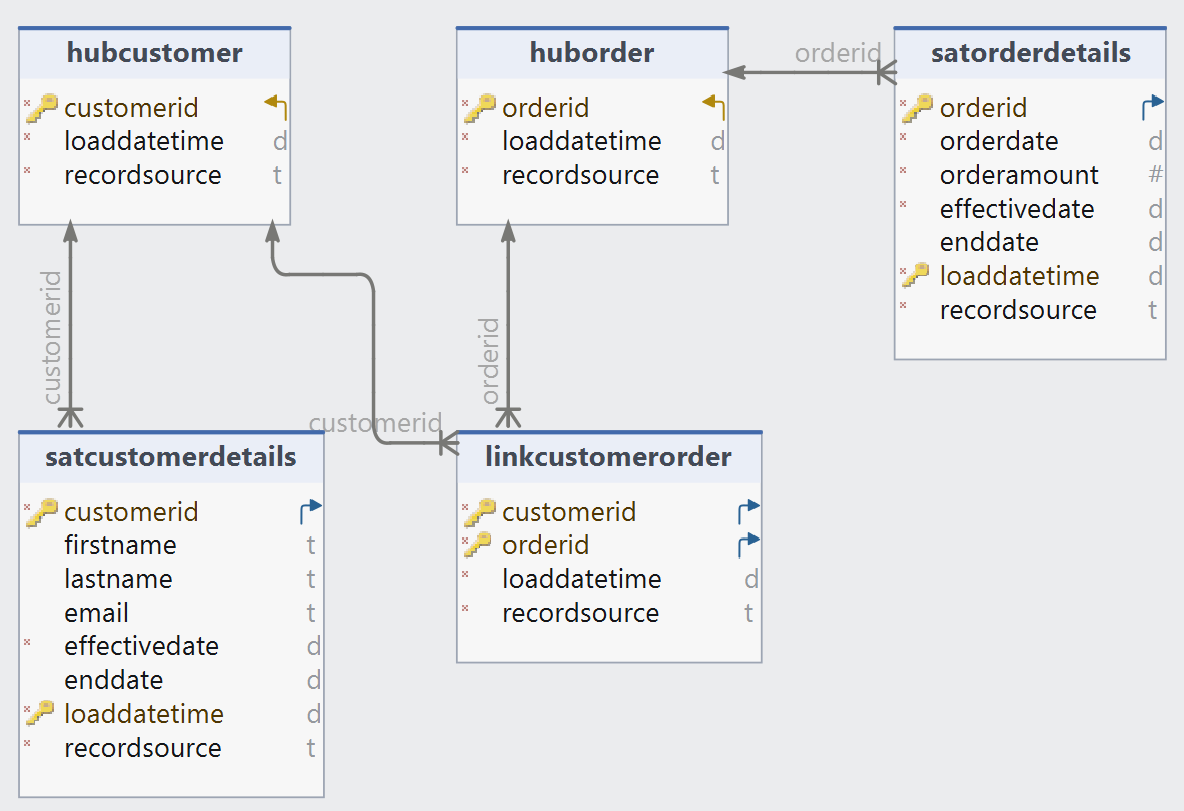
Quantity INT NOT NULL, -- Количество товара в заказе

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID), -- Связь с таблицей заказов

FOREIGN KEY (ProductID) REFERENCES Products(ProductID) -- Связь с таблицей товаров

);

1. Data vault



Моя база данных построена по архитектуре **Data Vault** и предназначена для хранения и обработки информации о клиентах и их заказах. Она состоит из трёх основных компонентов:

1. **Хабы (Hubs)**:
   * В хабах хранятся уникальные бизнес-ключи, которые идентифицируют основные бизнес-сущности, такие как клиенты и заказы.
   * Например, в таблице HubCustomer хранится уникальный идентификатор клиента (CustomerID), а в HubOrder — уникальный идентификатор заказа (OrderID).
2. **Линки (Links)**:
   * Линки связывают сущности между собой. В таблице LinkCustomerOrder отражена связь между клиентами и их заказами, где каждый заказ привязан к конкретному клиенту.
3. **Спутники (Satellites)**:
   * Спутники хранят детальную информацию (атрибуты) о сущностях и обеспечивают историчность данных.
   * В таблице SatCustomerDetails хранятся атрибуты клиентов, такие как имя, фамилия и email. В таблице SatOrderDetails — информация о заказах, включая дату заказа и его сумму. Спутники позволяют отслеживать изменения данных с помощью временных меток и сохранять исторические версии записей.

База данных обеспечивает гибкость, поддерживает историчность данных и легко масштабируется для добавления новых сущностей или атрибутов, что делает её идеальной для интеграции с множеством источников данных, таких как CRM или ERP системы.

Код:

-- Хаб для клиентов

CREATE TABLE HubCustomer (

CustomerID BIGINT PRIMARY KEY, -- Уникальный идентификатор клиента

LoadDateTime TIMESTAMP NOT NULL, -- Время загрузки данных

RecordSource VARCHAR(100) NOT NULL -- Источник данных (например, CRM)

);

-- Хаб для заказов

CREATE TABLE HubOrder (

OrderID BIGINT PRIMARY KEY, -- Уникальный идентификатор заказа

LoadDateTime TIMESTAMP NOT NULL, -- Время загрузки данных

RecordSource VARCHAR(100) NOT NULL -- Источник данных (например, ERP)

);

-- Линк, связывающий клиентов и заказы

CREATE TABLE LinkCustomerOrder (

CustomerID BIGINT NOT NULL, -- Внешний ключ на HubCustomer

OrderID BIGINT NOT NULL, -- Внешний ключ на HubOrder

LoadDateTime TIMESTAMP NOT NULL, -- Время загрузки данных

RecordSource VARCHAR(100) NOT NULL, -- Источник данных (например, CRM или ERP)

PRIMARY KEY (CustomerID, OrderID), -- Композитный первичный ключ

FOREIGN KEY (CustomerID) REFERENCES HubCustomer(CustomerID), -- Связь с хабом клиентов

FOREIGN KEY (OrderID) REFERENCES HubOrder(OrderID) -- Связь с хабом заказов

);

-- Спутник для атрибутов клиентов

CREATE TABLE SatCustomerDetails (

CustomerID BIGINT NOT NULL, -- Внешний ключ на HubCustomer

FirstName VARCHAR(100), -- Имя клиента

LastName VARCHAR(100), -- Фамилия клиента

Email VARCHAR(100), -- Email клиента

EffectiveDate TIMESTAMP NOT NULL, -- Дата начала действия записи

EndDate TIMESTAMP NULL, -- Дата окончания действия записи (NULL, если актуально)

LoadDateTime TIMESTAMP NOT NULL, -- Время загрузки данных

RecordSource VARCHAR(100) NOT NULL, -- Источник данных

PRIMARY KEY (CustomerID, LoadDateTime), -- Композитный ключ (ID клиента + время загрузки)

FOREIGN KEY (CustomerID) REFERENCES HubCustomer(CustomerID) -- Связь с хабом клиентов

);

-- Спутник для атрибутов заказов

CREATE TABLE SatOrderDetails (

OrderID BIGINT NOT NULL, -- Внешний ключ на HubOrder

OrderDate TIMESTAMP NOT NULL, -- Дата заказа

OrderAmount DECIMAL(18,2) NOT NULL, -- Сумма заказа

EffectiveDate TIMESTAMP NOT NULL, -- Дата начала действия записи

EndDate TIMESTAMP NULL, -- Дата окончания действия записи (NULL, если актуально)

LoadDateTime TIMESTAMP NOT NULL, -- Время загрузки данных

RecordSource VARCHAR(100) NOT NULL, -- Источник данных

PRIMARY KEY (OrderID, LoadDateTime), -- Композитный ключ (ID заказа + время загрузки)

FOREIGN KEY (OrderID) REFERENCES HubOrder(OrderID) -- Связь с хабом заказов

);