Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МОСКОВСКИЙ ПОЛИТЕХ)

Факультет информационных технологий

Кафедра «Прикладная информатика» Форма обучения: очная

Лабораторная работа №2

Тема: «Реализация структуры Data Vault»

по дисциплине

«Хранилища данных»

|  |  |  |
| --- | --- | --- |
| Студент | (личная подпись) | А.Р. Халитова |

Москва 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc187603310)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc187603311)

[Задача №1 4](#_Toc187603312)

[Задача №2 6](#_Toc187603313)

[ЗАКЛЮЧЕНИЕ 10](#_Toc187603314)

# ВВЕДЕНИЕ

В данной лабораторной работе рассматривается проектирование базы данных для торгового отдела с использованием двух различных моделей данных: третьей нормальной формы (3NF) и модели Data Vault. Основная цель работы заключается в создании эффективной и логически структурированной базы данных, которая будет соответствовать требованиям 3NF и модели Data Vault. Эти модели обеспечивают целостность данных, минимизируют избыточность и упрощают процесс анализа информации.

Работа выполняется в соответствии с вариантом 13 задания, который подразумевает создание системы, описывающей процесс работы торгового отдела. Для этого будет разработана даталогическая модель, включающая сущности, атрибуты и связи между ними, отражающие основные процессы торгового отдела.

Третья нормальная форма (3NF) – это вид нормализации баз данных, направленный на устранение транзитивных зависимостей. В соответствии с 3NF, все неключевые атрибуты должны зависеть только от первичного ключа и не зависеть друг от друга, что обеспечивает целостность и уменьшает избыточность данных.

# ОСНОВНАЯ ЧАСТЬ

# Задача №1

Первая часть лабораторной работы будет сосредоточена на реализации структуры базы данных в соответствии с моделью третьей нормальной формы (3NF). Модель 3NF предполагает, что все атрибуты в таблицах зависят только от первичного ключа и не содержат транзитивных зависимостей. Это позволяет избежать избыточности данных и улучшить их целостность.

Для реализации структуры базы данных в соответствии с третьей нормальной формой (3NF) в PostgreSQL необходимо создать несколько таблиц, которые будут описывать процесс подготовки к экзамену.

Сущности, которые участвуют в процессе:

1. сущность «Работники»;

2. сущность «Покупатели»;

3. сущность «Товары»;

4. сущность «Заказы»;

5. сущность «Товары в заказах».

Ниже представлен SQL-скрипт для создания базы данных.

CREATE DATABASE sales;

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20)

);

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20),

address VARCHAR(200)

);

CREATE TABLE Products (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(100),

description TEXT,

price DECIMAL(10, 2)

);

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id INT NOT NULL REFERENCES Customers(customer\_id),

employee\_id INT NOT NULL REFERENCES Employees(employee\_id),

order\_date DATE,

total\_amount DECIMAL(10, 2)

);

CREATE TABLE OrderItems (

order\_item\_id INT PRIMARY KEY,

order\_id INT NOT NULL REFERENCES Orders(order\_id),

product\_id INT NOT NULL REFERENCES Products(product\_id),

quantity INT,

unit\_price DECIMAL(10, 2)

);

Таблица Employees: хранит информацию о работниках, включая уникальный идентификатор, имя, фамилию, электронную почту и номер телефона.

Таблица Customers содержит информацию о покупателях, включая уникальный идентификатор, имя, фамилию, электронную почту, номер телефона и адрес.

Таблица Products содержит информацию о товарах, включая уникальный идентификатор, название продукта, описание и цену.

Таблица Orders содержит информацию о заказах, включая уникальный идентификатор, идентификатор покупателя (ссылается на таблицу Customers), идентификатор работника (ссылается на таблицу Employees), дату заказа и стоимость заказа.

Таблица OrderItems хранит информацию об товарах в заказах, включая уникальный идентификатор, идентификатор заказа (ссылается на таблицу Orders), идентификатор товара (ссылается на таблицу Products), количество и стоимость.

# Задача №2

Вторая часть лабораторной работы посвящена реализации структуры базы данных в соответствии с моделью Data Vault, которая предназначена для хранения исторических данных и обеспечивает гибкость в анализе и интеграции данных из различных источников. Модель Data Vault состоит из трех основных компонентов:

Хранилища (Hubs): Это таблицы, представляющие ключевые бизнес-сущности, такие как товары и покупатели. Они содержат уникальные идентификаторы (суррогатные ключи) и минимальный набор атрибутов, не подверженных изменениям.

Связи (Links): Это таблицы, устанавливающие отношения между различными бизнес-сущностями, например, связь между покупателями и заказами. Они содержат суррогатные ключи из соответствующих хранилищ.

Атрибуты (Satellites): Это таблицы, содержащие атрибуты, которые могут изменяться со временем, такие как цены товаров, адреса клиентов или описания продуктов. Они хранят историческую информацию, что позволяет отслеживать изменения и проводить анализ данных во времени.

В рамках данной модели будут созданы отдельные таблицы для каждого из этих компонентов, чтобы обеспечить гибкость, масштабируемость и возможность отслеживания изменений в данных.

Для системы, описывающей порядок подготовки к экзамену, можно выделить следующие сущности, атрибуты и элементы:

1. Hubs:

• Hub\_Employees: хранит уникальные идентификаторы работников.

• Hub\_Customers: хранит уникальные идентификаторы покупателей.

• Hub\_Products: хранит уникальные идентификаторы товаров.

2. Links:

• Lnk\_Order: связывает покупателей и работников в заказах.

• Lnk\_OrderItems: связывает продукты с заказами.

3. Satellites:

• Sat\_Employees: хранит атрибуты работников (имя, фамилия, email, номер телефона).

• Sat\_Customers: хранит атрибуты предметов (имя, фамилия, email, номер телефона, адрес).

• Sat\_Products: хранит атрибуты товаров (дата и цена).

• Sat\_Orders: хранит дату и стоимость заказа.

• Sat\_OrderItems: хранит количество и стоимость продуктов в заказе.

SQL-скрипт для создания базы данных.

CREATE DATABASE sales;

CREATE TABLE Hub\_Employees (

employee\_id INT PRIMARY KEY,

);

CREATE TABLE Hub\_Customers (

customer\_id INT PRIMARY KEY,

);

CREATE TABLE Hub\_Products (

product\_id INT PRIMARY KEY,

);

CREATE TABLE Lnk\_Orders (

order\_id INT PRIMARY KEY,

customer\_id INT NOT NULL REFERENCES Hub\_Customers(customer\_id),

employee\_id INT NOT NULL REFERENCES Hub\_Employees(employee\_id),

order\_date DATE

);

CREATE TABLE Lnk\_OrderItems (

order\_item\_id INT PRIMARY KEY,

order\_id INT NOT NULL REFERENCES Lnk\_Orders(order\_id),

product\_id INT NOT NULL REFERENCES Hub\_Products(product\_id)

);

CREATE TABLE Sat\_Employees (

employee\_id INT NOT NULL REFERENCES Hub\_Employees(employee\_id),

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20)

);

CREATE TABLE Sat\_Customers (

customer\_id INT NOT NULL REFERENCES Hub\_Customers(customer\_id),

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20),

address VARCHAR(200)

);

CREATE TABLE Sat\_Products (

product\_id INT NOT NULL REFERENCES Hub\_Products(product\_id),

product\_name VARCHAR(100),

description TEXT,

price DECIMAL(10, 2)

);

CREATE TABLE Sat\_Orders (

order\_id INT NOT NULL REFERENCES Lnk\_Orders(order\_id),

total\_amount DECIMAL(10, 2)

);

CREATE TABLE Sat\_OrderItems (

order\_item\_id INT NOT NULL REFERENCES Lnk\_OrderItems(order\_item\_id),

quantity INT,

unit\_price DECIMAL(10, 2)

);

Ниже приведена даталогическая модель данной системы (Рисунок 1).

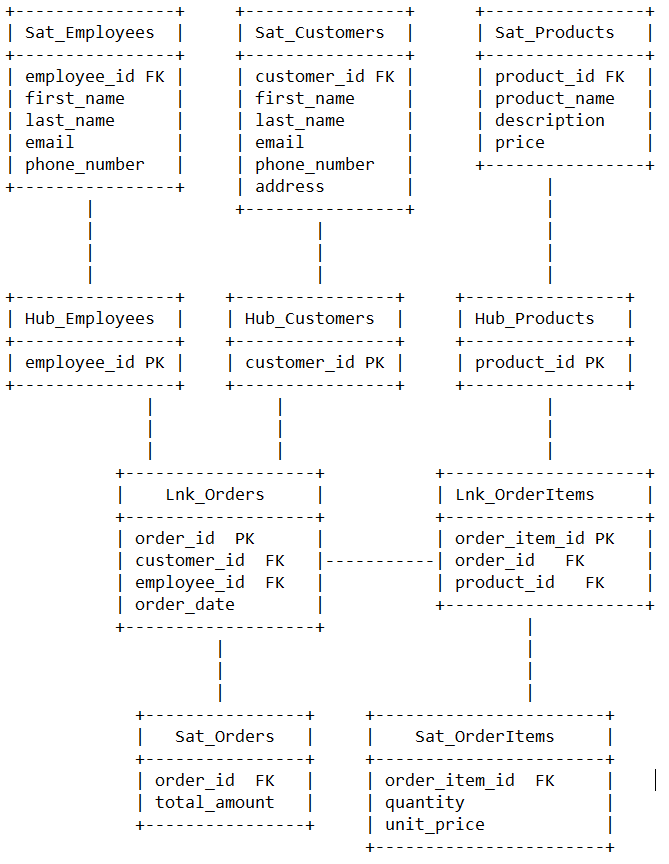


Рисунок 1 - Data Vault

# ЗАКЛЮЧЕНИЕ

Данная лабораторная работа была направлена на разработку структуры базы данных, соответствующей двум различным моделям: третьей нормальной форме (3NF) и модели Data Vault. Эти модели представляют собой разные подходы к проектированию баз данных, каждый из которых имеет свои особенности и преимущества.

Первая часть работы была посвящена созданию базы данных в соответствии с моделью 3NF. Эта модель нацелена на минимизацию избыточности данных и обеспечение целостности информации. В процессе разработки была построена даталогическая модель, включающая таблицы, отражающие сущности. Каждая таблица была спроектирована таким образом, чтобы содержать только необходимые атрибуты, а связи между ними были четко определены. Такой подход позволяет избежать дублирования данных и упрощает их обработку, а также предотвращает аномалии при обновлении, вставке и удалении данных, что делает структуру базы данных более эффективной и надежной. Реализованный SQL-код создания базы данных демонстрирует, как эти сущности связаны друг с другом и как они могут быть использованы для эффективного управления информацией.

Вторая часть лабораторной работы была посвящена реализации структуры базы данных по модели Data Vault. Эта модель ориентирована на интеграцию данных из различных источников и обеспечивает гибкость в изменении структуры данных. В процессе разработки была создана даталогическая модель, включающая хранилища (hubs), связи (links) и атрибуты (satellites). Хранилища представляют собой ключевые бизнес-сущности, связи фиксируют отношения между ними, а атрибуты содержат историческую информацию. Такой подход позволяет легко адаптироваться к изменениям в бизнес-требованиях и обеспечивает высокую степень масштабируемости системы.

Система хранилищ данных Data Vault обеспечивает масштабируемость и гибкость хранения данных, что позволяет использовать ее для работы с большими массивами данных. В рамках данной лабораторной работы были успешно созданы схемы баз данных, соответствующие как третьей нормальной форме, так и модели Data Vault, что позволяет нам оценить преимущества и особенности каждой из этих моделей в контексте решения задач по управлению информацией.