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) | Д.Д. Свинцов |

Москва 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc209477130)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc209477131)

[Задача №1 4](#_Toc209477132)

[Задача №2 6](#_Toc209477133)

[ЗАКЛЮЧЕНИЕ 11](#_Toc209477134)

# ВВЕДЕНИЕ

В этой лабораторной работе выполняется проектирование базы данных для торгового отдела с применением двух различных подходов: модели третьей нормальной формы (3NF) и методологии Data Vault. Основная задача состоит в создании логически выстроенной и производительной структуры хранения, соответствующей принципам обеих моделей. Их использование гарантирует целостность данных, снижает дублирование и облегчает последующий аналитический разбор информации.

Для практической части выбран пятый вариант задания, предполагающий разработку системы, описывающей процесс организации общеинститутского студенческого мероприятия. В рамках работы формируется даталогическая модель с определением ключевых сущностей, их атрибутов и взаимосвязей, отражающих основные процессы торгового отдела.

Третья нормальная форма представляет собой этап нормализации, устраняющий транзитивные зависимости: каждый неключевой атрибут должен быть связан исключительно с первичным ключом и не зависеть от других неключевых данных, что обеспечивает структурную целостность и минимизирует избыточность.

# ОСНОВНАЯ ЧАСТЬ

# Задача №1

На первом этапе лабораторной работы основное внимание уделяется созданию структуры базы данных по принципам третьей нормальной формы (3NF). Эта модель требует, чтобы каждый атрибут таблицы зависел исключительно от первичного ключа и не имел транзитивных связей, что обеспечивает высокую целостность данных и устраняет их избыточность.

Для практической реализации схемы в PostgreSQL планируется разработать несколько взаимосвязанных таблиц, описывающих процесс подготовки к экзамену, полностью соответствующих требованиям 3NF.

Сущности, которые участвуют в процессе:

1. сущность “Мероприятие”;

2. сущность “Организатор”;

3. сущность “Участник”;

4. сущность “Ресурс”;

5. сущность “Задача/Этап подготовки”;

Ниже представлен SQL-скрипт для создания базы данных.

CREATE DATABASE student\_event;

CREATE TABLE Event (

event\_id INT PRIMARY KEY,

event\_name VARCHAR(100),

event\_date DATE,

event\_time TIME,

location VARCHAR(150),

event\_type VARCHAR(50),

description TEXT

);

CREATE TABLE Organizer (

organizer\_id INT PRIMARY KEY,

full\_name VARCHAR(100),

position VARCHAR(100),

email VARCHAR(100),

phone\_number VARCHAR(20),

department VARCHAR(100)

);

CREATE TABLE Participant (

participant\_id INT PRIMARY KEY,

full\_name VARCHAR(100),

role VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20),

faculty VARCHAR(100)

);

CREATE TABLE Resource (

resource\_id INT PRIMARY KEY,

resource\_name VARCHAR(100),

resource\_type VARCHAR(50),

status VARCHAR(50),

responsible\_id INT,

FOREIGN KEY (responsible\_id) REFERENCES Organizer(organizer\_id)

);

CREATE TABLE Task (

task\_id INT PRIMARY KEY,

task\_name VARCHAR(100),

description TEXT,

due\_date DATE,

status VARCHAR(50),

responsible\_id INT,

event\_id INT,

FOREIGN KEY (responsible\_id) REFERENCES Organizer(organizer\_id),

FOREIGN KEY (event\_id) REFERENCES Event(event\_id)

);

Таблица Event хранит сведения о студенческом мероприятии: уникальный идентификатор события, его название, дату и время проведения, место, тип (например, концерт, конференция, спортивное) и краткое описание.

Таблица Organizer содержит информацию об организаторах мероприятия, включая уникальный идентификатор, полное имя, должность, контактные данные (e-mail, телефон) и подразделение или факультет, который они представляют.

Таблица Participant предназначена для хранения данных об участниках события: уникальный идентификатор, полное имя, роль (студент, гость, волонтёр и т. д.), контактную информацию и факультет или учебную группу.

Таблица Resource описывает ресурсы, необходимые для подготовки и проведения мероприятия: уникальный идентификатор ресурса, его название, тип, текущий статус (в наличии, зарезервирован, выдан) и ссылку на ответственного организатора.

Таблица Task фиксирует этапы подготовки и отдельные задачи: уникальный идентификатор задачи, название, подробное описание, срок выполнения, статус (в процессе, завершена и т. п.), ссылку на ответственного организатора и идентификатор мероприятия, к которому относится задача.

# Задача №2

Во второй части лабораторной работы внимание уделяется построению базы данных по методологии Data Vault. Этот подход ориентирован на долговременное хранение и позволяет объединять данные из разных источников, сохраняя их историю и обеспечивая гибкую аналитику.

Архитектура Data Vault опирается на три ключевых элемента. В основе лежат Hubs – центральные таблицы, фиксирующие уникальные бизнес-объекты и содержащие только неизменяемые идентификаторы и минимальный набор стабильных атрибутов. Связи между этими объектами оформляются через Links, которые объединяют хранилища и включают в себя суррогатные ключи, указывающие на соответствующие Hubs. Изменяемые характеристики, такие как стоимость продукции, контактные данные или описания, выносятся в Satellites – таблицы, которые фиксируют полную историю изменений и позволяют анализировать динамику во времени.

В ходе работы для каждого из перечисленных компонентов будут созданы отдельные таблицы, что обеспечит масштабируемость, удобство интеграции новых источников и детальное отслеживание всех корректировок данных.

Для системы, описывающей организацию общеинститутского студенческого мероприятия, можно выделить следующие сущности, атрибуты и элементы:

1. Hubs

Hub\_Event – фиксирует уникальные идентификаторы мероприятий.

Hub\_Organizer – содержит уникальные идентификаторы организаторов.

Hub\_Participant – хранит уникальные идентификаторы участников.

Hub\_Resource – включает уникальные идентификаторы ресурсов (оборудование, помещения и т.д.).

Hub\_Task – хранит уникальные идентификаторы задач или этапов подготовки.

2. Links

Lnk\_EventOrganizer – связывает мероприятия с организаторами.

Lnk\_EventParticipant – устанавливает связь между мероприятием и его участниками.

Lnk\_EventResource – описывает использование конкретных ресурсов для конкретного мероприятия.

Lnk\_EventTask – соединяет мероприятия с задачами подготовки.

Lnk\_TaskResponsible – определяет, какой организатор или участник отвечает за конкретную задачу.

3. Satellites

Sat\_Event – хранит изменяемые характеристики мероприятия (название, место проведения, дата и время, описание, тип).

Sat\_Organizer – содержит атрибуты организаторов (ФИО, должность, контактные данные, факультет/подразделение).

Sat\_Participant – хранит изменяющиеся данные об участниках (ФИО, роль в мероприятии, контактная информация, факультет).

Sat\_Resource – содержит свойства ресурсов (тип, статус, ответственный за использование).

Sat\_Task – фиксирует детали задач (название, описание, срок исполнения, текущий статус).

Sat\_TaskHistory – может дополнительно хранить историю изменения статусов и сроков задач.

SQL-скрипт для создания базы данных.

CREATE DATABASE student\_event\_vault;

CREATE TABLE Hub\_Event (

event\_id INT PRIMARY KEY

);

CREATE TABLE Hub\_Organizer (

organizer\_id INT PRIMARY KEY

);

CREATE TABLE Hub\_Participant (

participant\_id INT PRIMARY KEY

);

CREATE TABLE Hub\_Resource (

resource\_id INT PRIMARY KEY

);

CREATE TABLE Hub\_Task (

task\_id INT PRIMARY KEY

);

CREATE TABLE Lnk\_EventOrganizer (

link\_event\_organizer\_id INT PRIMARY KEY,

event\_id INT NOT NULL REFERENCES Hub\_Event(event\_id),

organizer\_id INT NOT NULL REFERENCES Hub\_Organizer(organizer\_id)

);

CREATE TABLE Lnk\_EventParticipant (

link\_event\_participant\_id INT PRIMARY KEY,

event\_id INT NOT NULL REFERENCES Hub\_Event(event\_id),

participant\_id INT NOT NULL REFERENCES Hub\_Participant(participant\_id)

);

CREATE TABLE Lnk\_EventResource (

link\_event\_resource\_id INT PRIMARY KEY,

event\_id INT NOT NULL REFERENCES Hub\_Event(event\_id),

resource\_id INT NOT NULL REFERENCES Hub\_Resource(resource\_id)

);

CREATE TABLE Lnk\_EventTask (

link\_event\_task\_id INT PRIMARY KEY,

event\_id INT NOT NULL REFERENCES Hub\_Event(event\_id),

task\_id INT NOT NULL REFERENCES Hub\_Task(task\_id)

);

CREATE TABLE Lnk\_TaskResponsible (

link\_task\_responsible\_id INT PRIMARY KEY,

task\_id INT NOT NULL REFERENCES Hub\_Task(task\_id),

organizer\_id INT REFERENCES Hub\_Organizer(organizer\_id),

participant\_id INT REFERENCES Hub\_Participant(participant\_id)

);

CREATE TABLE Sat\_Event (

event\_id INT NOT NULL REFERENCES Hub\_Event(event\_id),

event\_name VARCHAR(100),

event\_date DATE,

event\_time TIME,

location VARCHAR(150),

event\_type VARCHAR(50),

description TEXT

);

CREATE TABLE Sat\_Organizer (

organizer\_id INT NOT NULL REFERENCES Hub\_Organizer(organizer\_id),

full\_name VARCHAR(100),

position VARCHAR(100),

email VARCHAR(100),

phone\_number VARCHAR(20),

department VARCHAR(100)

);

CREATE TABLE Sat\_Participant (

participant\_id INT NOT NULL REFERENCES Hub\_Participant(participant\_id),

full\_name VARCHAR(100),

role VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20),

faculty VARCHAR(100)

);

CREATE TABLE Sat\_Resource (

resource\_id INT NOT NULL REFERENCES Hub\_Resource(resource\_id),

resource\_name VARCHAR(100),

resource\_type VARCHAR(50),

status VARCHAR(50),

notes TEXT

);

CREATE TABLE Sat\_Task (

task\_id INT NOT NULL REFERENCES Hub\_Task(task\_id),

task\_name VARCHAR(100),

description TEXT,

due\_date DATE,

status VARCHAR(50)

);

CREATE TABLE Sat\_TaskHistory (

task\_id INT NOT NULL REFERENCES Hub\_Task(task\_id),

status\_change\_date TIMESTAMP,

old\_status VARCHAR (50),

new\_status VARCHAR (50)

);

Ниже приведена даталогическая модель данной системы (Рисунок 1).

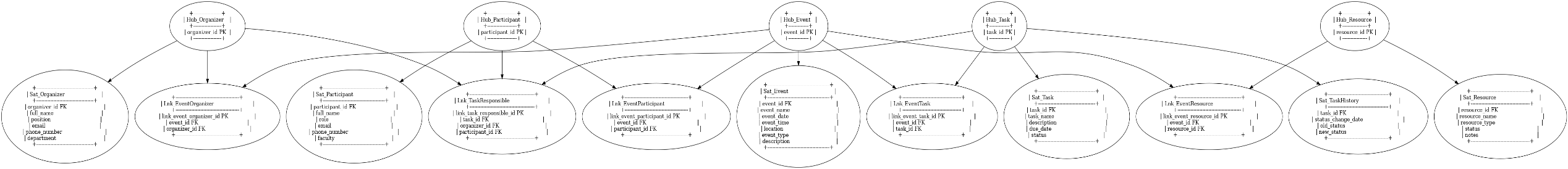


Рисунок 1 - Data Vault

# ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы была разработана структура базы данных по двум моделям: третьей нормальной форме (3NF) и Data Vault, каждая из которых имеет свои особенности и преимущества.

В первой части создавалась база данных по модели 3NF, направленной на минимизацию избыточности и обеспечение целостности данных. Были спроектированы таблицы с необходимыми атрибутами и четкими связями между ними, что предотвращает дублирование и аномалии при обновлении, вставке и удалении данных. SQL-реализация показала, как сущности взаимодействуют и обеспечивают эффективное управление информацией.

Во второй части структура базы данных строилась по модели Data Vault, ориентированной на интеграцию данных из разных источников и гибкость при изменении структуры. Созданные хранилища (hubs), связи (links) и атрибуты (satellites) позволили зафиксировать ключевые сущности, их отношения и исторические данные. Такой подход обеспечивает масштабируемость и легкую адаптацию к изменениям бизнес-требований.

В результате лабораторной работы были реализованы обе модели, что позволило сравнить их возможности и оценить эффективность для управления информацией и работы с большими объемами данных.