|  |
| --- |
| 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: заочная

Лабораторная работа №2

«Тема: Реализация структуры DataVault.»

по дисциплине

«Хранилища данных»

А. Р. Пыльцын

|  |  |  |  |
| --- | --- | --- | --- |
| Студент | (личная подпись) |  | (И.О. Фамилия) |

Москва 2024

**Выполнение работы**

Для выполнения работы был использован сервис <https://dbdiagram.io/d>. В рамках работы выбран Вариант 7: система должна описывать порядок поставок товара в систему розничных киосков. Диаграмма была построена в соответствии с темой.

**Структура базы данных описывается кодом:**

// Таблицы-хабы

Table hub\_kiosks {

kiosk\_key **int** [pk] // уникальный бизнес-ключ киоска

created\_at datetime

}

Table hub\_users {

user\_key **int** [pk] // уникальный бизнес-ключ пользователя

created\_at datetime

}

Table hub\_suppliers {

supplier\_key **int** [pk] // уникальный бизнес-ключ поставщика

created\_at datetime

}

Table hub\_products {

product\_key **int** [pk] // уникальный бизнес-ключ продукта

created\_at datetime

}

Table hub\_orders {

order\_key **int** [pk] // уникальный бизнес-ключ заказа

created\_at datetime

}

// Таблицы-связи (Link)

Table link\_kiosk\_orders {

kiosk\_order\_key **int** [pk] // уникальный бизнес-ключ связи между киосками и заказами

kiosk\_key **int** [**not** null]

order\_key **int** [**not** null]

created\_at datetime

}

Table link\_supplier\_products {

supplier\_product\_key **int** [pk] // уникальный бизнес-ключ связи между поставщиками и продуктами

supplier\_key **int** [**not** null]

product\_key **int** [**not** null]

created\_at datetime

}

// Таблицы-спутники (Satellite)

Table sat\_kiosks {

kiosk\_key **int** [pk] // ссылается на hub\_kiosks.kiosk\_key

name varchar

location varchar

created\_at datetime

}

Table sat\_users {

user\_key **int** [pk] // ссылается на hub\_users.user\_key

full\_name varchar

country\_code **int**

created\_at datetime

}

Table sat\_suppliers {

supplier\_key **int** [pk] // ссылается на hub\_suppliers.supplier\_key

name varchar

contact\_info varchar

country\_code **int**

created\_at datetime

}

Table sat\_products {

product\_key **int** [pk] // ссылается на hub\_products.product\_key

name varchar

description varchar

price decimal

status varchar // статус товара

created\_at datetime

}

Table sat\_orders {

order\_key **int** [pk] // ссылается на hub\_orders.order\_key

order\_date datetime

total\_amount decimal

status varchar // статус заказа

created\_at datetime

}

Table sat\_order\_items {

order\_key **int** [pk] // ссылается на hub\_orders.order\_key

product\_key **int** [**not** null] // ссылается на hub\_products.product\_key

quantity **int**

price decimal

created\_at datetime

}

Table sat\_delivery\_schedule {

order\_key **int** [pk] // ссылается на hub\_orders.order\_key

delivery\_date datetime

delivery\_status varchar

created\_at datetime

}

Table sat\_inventory {

kiosk\_key **int** [pk] // ссылается на hub\_kiosks.kiosk\_key

product\_key **int** [**not** null] // ссылается на hub\_products.product\_key

quantity **int**

restock\_date datetime

created\_at datetime

}

// Создание связей

Ref: sat\_kiosks.kiosk\_key > hub\_kiosks.kiosk\_key

Ref: sat\_users.user\_key > hub\_users.user\_key

Ref: sat\_suppliers.supplier\_key > hub\_suppliers.supplier\_key

Ref: sat\_products.product\_key > hub\_products.product\_key

Ref: sat\_orders.order\_key > hub\_orders.order\_key

Ref: link\_kiosk\_orders.kiosk\_key > hub\_kiosks.kiosk\_key

Ref: link\_kiosk\_orders.order\_key > hub\_orders.order\_key

Ref: link\_supplier\_products.supplier\_key > hub\_suppliers.supplier\_key

Ref: link\_supplier\_products.product\_key > hub\_products.product\_key

Ref: sat\_order\_items.order\_key > hub\_orders.order\_key

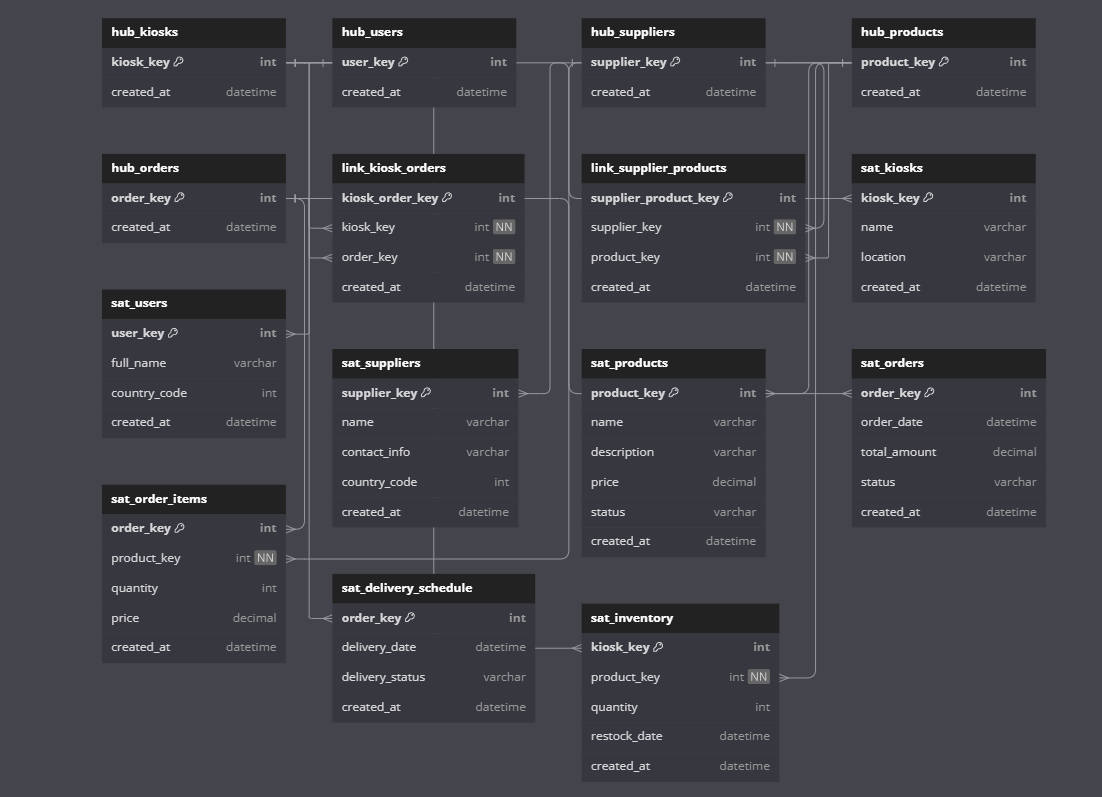
Ref: sat\_order\_items.product\_key > hub\_products.product\_key

Ref: sat\_delivery\_schedule.order\_key > hub\_orders.order\_key

Ref: sat\_inventory.kiosk\_key > hub\_kiosks.kiosk\_key

Ref: sat\_inventory.product\_key > hub\_products.product\_key

**Сформирована следующая схема:**



В формате PostgreSQL запрос на создание базы данных будет выглядеть следующим образом:

CREATE TABLE "hub\_kiosks" (

"kiosk\_key" **int** PRIMARY KEY,

"created\_at" datetime

);

CREATE TABLE "hub\_users" (

"user\_key" **int** PRIMARY KEY,

"created\_at" datetime

);

CREATE TABLE "hub\_suppliers" (

"supplier\_key" **int** PRIMARY KEY,

"created\_at" datetime

);

CREATE TABLE "hub\_products" (

"product\_key" **int** PRIMARY KEY,

"created\_at" datetime

);

CREATE TABLE "hub\_orders" (

"order\_key" **int** PRIMARY KEY,

"created\_at" datetime

);

CREATE TABLE "link\_kiosk\_orders" (

"kiosk\_order\_key" **int** PRIMARY KEY,

"kiosk\_key" **int** NOT NULL,

"order\_key" **int** NOT NULL,

"created\_at" datetime

);

CREATE TABLE "link\_supplier\_products" (

"supplier\_product\_key" **int** PRIMARY KEY,

"supplier\_key" **int** NOT NULL,

"product\_key" **int** NOT NULL,

"created\_at" datetime

);

CREATE TABLE "sat\_kiosks" (

"kiosk\_key" **int** PRIMARY KEY,

"name" varchar,

"location" varchar,

"created\_at" datetime

);

CREATE TABLE "sat\_users" (

"user\_key" **int** PRIMARY KEY,

"full\_name" varchar,

"country\_code" **int**,

"created\_at" datetime

);

CREATE TABLE "sat\_suppliers" (

"supplier\_key" **int** PRIMARY KEY,

"name" varchar,

"contact\_info" varchar,

"country\_code" **int**,

"created\_at" datetime

);

CREATE TABLE "sat\_products" (

"product\_key" **int** PRIMARY KEY,

"name" varchar,

"description" varchar,

"price" decimal,

"status" varchar,

"created\_at" datetime

);

CREATE TABLE "sat\_orders" (

"order\_key" **int** PRIMARY KEY,

"order\_date" datetime,

"total\_amount" decimal,

"status" varchar,

"created\_at" datetime

);

CREATE TABLE "sat\_order\_items" (

"order\_key" **int** PRIMARY KEY,

"product\_key" **int** NOT NULL,

"quantity" **int**,

"price" decimal,

"created\_at" datetime

);

CREATE TABLE "sat\_delivery\_schedule" (

"order\_key" **int** PRIMARY KEY,

"delivery\_date" datetime,

"delivery\_status" varchar,

"created\_at" datetime

);

CREATE TABLE "sat\_inventory" (

"kiosk\_key" **int** PRIMARY KEY,

"product\_key" **int** NOT NULL,

"quantity" **int**,

"restock\_date" datetime,

"created\_at" datetime

);

ALTER TABLE "sat\_kiosks" ADD FOREIGN KEY ("kiosk\_key") REFERENCES "hub\_kiosks" ("kiosk\_key");

ALTER TABLE "sat\_users" ADD FOREIGN KEY ("user\_key") REFERENCES "hub\_users" ("user\_key");

ALTER TABLE "sat\_suppliers" ADD FOREIGN KEY ("supplier\_key") REFERENCES "hub\_suppliers" ("supplier\_key");

ALTER TABLE "sat\_products" ADD FOREIGN KEY ("product\_key") REFERENCES "hub\_products" ("product\_key");

ALTER TABLE "sat\_orders" ADD FOREIGN KEY ("order\_key") REFERENCES "hub\_orders" ("order\_key");

ALTER TABLE "link\_kiosk\_orders" ADD FOREIGN KEY ("kiosk\_key") REFERENCES "hub\_kiosks" ("kiosk\_key");

ALTER TABLE "link\_kiosk\_orders" ADD FOREIGN KEY ("order\_key") REFERENCES "hub\_orders" ("order\_key");

ALTER TABLE "link\_supplier\_products" ADD FOREIGN KEY ("supplier\_key") REFERENCES "hub\_suppliers" ("supplier\_key");

ALTER TABLE "link\_supplier\_products" ADD FOREIGN KEY ("product\_key") REFERENCES "hub\_products" ("product\_key");

ALTER TABLE "sat\_order\_items" ADD FOREIGN KEY ("order\_key") REFERENCES "hub\_orders" ("order\_key");

ALTER TABLE "sat\_order\_items" ADD FOREIGN KEY ("product\_key") REFERENCES "hub\_products" ("product\_key");

ALTER TABLE "sat\_delivery\_schedule" ADD FOREIGN KEY ("order\_key") REFERENCES "hub\_orders" ("order\_key");

ALTER TABLE "sat\_inventory" ADD FOREIGN KEY ("kiosk\_key") REFERENCES "hub\_kiosks" ("kiosk\_key");

ALTER TABLE "sat\_inventory" ADD FOREIGN KEY ("product\_key") REFERENCES "hub\_products" ("product\_key");

**Заключение**

В итоге была разработана база данных, описывающая порядок поставок товара в систему розничных киосков. В работе представлен PostgreSQL и код в соответствии с моделью Data Vault, который формирует схему базы данных.