**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

**МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ**

**им. А. Н. ТИХОНОВА**

Проектирование реляционной базы данных «Магазин Высокой моды»

по дисциплине:

«Базы Данных»

Колодин Максим

Тернов Михаил

студентов 2 курса образовательной программы «Информатика и вычислительная техника»

группы БИВ205

**Москва**

**2022**

Оглавление

[Инфологическое проектирование 3](#_Toc102915684)

[Анализ информационных задач и круга пользователей системы 4](#_Toc102915685)

[Определение требований к операционной обстановке 5](#_Toc102915686)

[Выбор СУБД и других программных средств](#_Toc102915687) 6

[Логическое проектирование реляционной БД](#_Toc102915688) 6

[Составление реляционных выражений](#_Toc102915689) 7

[Нормализация полученных отношений](#_Toc102915690) 9

[Определение дополнительных ограничений целостности](#_Toc102915691) 11

[Описание групп пользователей и прав доступа](#_Toc102915691) 13

Реализация проекта базы данных14

Начальный состав данных для справочных таблиц16

Создание представлений (готовых запросов) 17

Назначение прав доступа 18

Создание триггеров 18

Разработка стратегии резервного копирования 20

Инфологическое проектирование

База данных создается для обслуживания сотрудников магазина и его клиентов. База Данных должна содержать данные о товарах, их разнообразии и количестве.

В соответствии с предметной областью система строится с учётом следующих особенностей:

* Один клиент может совершить несколько актов покупки. Один акт покупки относится к одному клиенту.
* Один акт покупки может включать много товаров. При этом один товар может быть в одном акте покупки.
* Один товар относится к одной категории. Одна категория может быть у многих товаров.
* Один товар относится к одному бренду. Один бренд может относиться к многим товарам.
* Конкретный товар находится в наличии. В наличии может находиться много товаров.

Для создания ER-модели необходимо выделить сущности области и указать их атрибуты. **Идентифицирующие** атрибуты мы выделим полужирным шрифтом, *многозначные* – курсивом, составные подчеркнем:

1. **Товары.** Атрибуты: Название товара, **id товара (артикул),** *цена*, *материал, категория****.***
2. **Клиент**. Атрибуты: ФИО, *телефон,* адрес.
3. **Акт покупки.** Атрибуты: **номер,** дата, количество, цена.
4. **Бренд.** Атрибуты: **Название,** *страна*, *дата основания*.
5. **Наличие.** Атрибуты: **id позиции**, *количество, размер, цвет*.

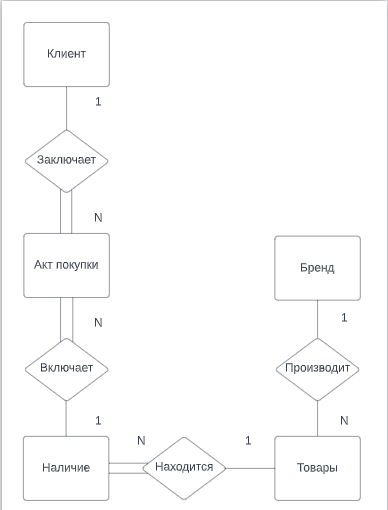


Схема 1 ER-диаграмма предметной области «Магазин высокой моды»

Анализ информационных задач и круга пользователей системы

Определим группы пользователей, их основные задачи и запросы к БД:

1. Владелец
   1. Проверка наличия товаров для закупки
   2. Просмотр цены для соответствия курсу
   3. Добавление, удаление, изменение товаров
   4. Изменение характеристик товаров
   5. Удаление товаров
   6. Добавление, изменение, удаление брендов
   7. Изменение цены
   8. Добавление, изменение, удаление данных о клиентах
2. Старший менеджер
   1. Добавление разнообразия товаров
   2. Добавление количества товаров
   3. Получение информации по актам покупок
   4. Получение информации о наличии товара при работе с клиентом
   5. Добавление, изменение, удаление данных о клиентах
3. Менеджер
   1. Получение информации о наличии товара при работе с клиентом
   2. Получение информации о клиенте для сообщения о наличии товара
   3. Составление, изменение акта покупки
4. Клиент
   1. Получение информации о наличии товара
   2. Получение информации о необходимых брендах
   3. Получение информации о цене

Определение требований к операционной обстановке

На основе результатов анализа Предметной области можно приблизительно оценить объём памяти, требуемой для хранения данных. Примем ориентировочно, что:

1. Всего в магазине может храниться 3000 различных товаров (1К на каждый)
2. Количество клиентов в год будет составлять примерно 15000 (1К на клиента)
3. Количество актов покупки в год будет составлять примерно 30000 (0.5К на каждый)
4. Количество брендов в магазине будет составлять 200 (0.5К на каждый)
5. На общее наличие товаров будет отводиться примерно 18000 (1К на каждый)

Тогда объем памяти для хранения данных на первый год составляет:

Коэффициент 2 необходим для того, чтобы учесть необходимость выделения памяти под дополнительные структуры (например, индексы).

Как видим, база данных занимает небольшое количество дискового пространства. Из года в год будет увеличивается количество клиентов в базе, могут добавляться новые бренды, товары и прочее. Так что возьмем стандартный 1G чтобы покрыть потребности на несколько лет с излишком.

Требуемый объем оперативной памяти определяет на основании анализа интенсивности запросов и объема результирующих данных. Для нашей БД требуемый объем памяти не велик, поэтому ни каких специальных требований к объему внешней и оперативной памяти компьютера не предъявляется.

Выбор СУБД и других программных средств

Анализ информационных задач показывает, что для реализации требуемых функций подходят почти все СУБД для ПЭВМ (PostgreSQL, MS Access, Firebird, MySQL и др.). Все они поддерживают реляционную модель данных и предоставляют разнообразные возможности для работы с данными.

Объём внешней и оперативной памяти, требующийся для функционирования СУБД, обычно указывается в сопроводительной документации.

Для реализации данной задачи будем использовать СУБД PostgreSQL версии 13.5.

Логическое проектирование реляционной БД

База данных создаётся на основании схемы базы данных. Преобразование ER–диаграммы(Схема 1) в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы) БД. Связь типа 1:N (один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится для дочернего отношения. Внешнему ключу должен соответствовать первичный или уникальный ключ основного (родительского) отношения.

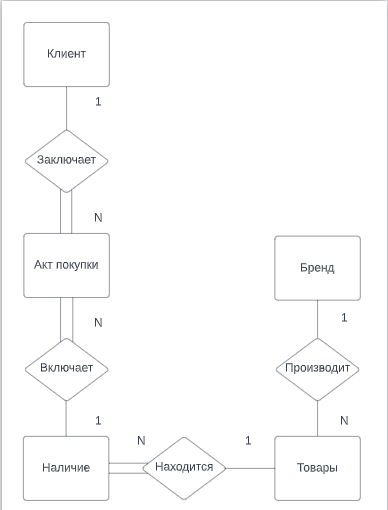


Схема 2 ER-диаграмма предметной области «Магазин высокой моды»

Для схемы БД будем использовать обозначения, представленные на Рисунок 1.

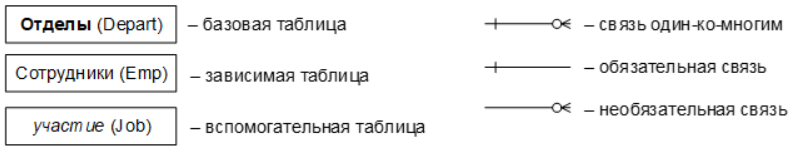


Рисунок 1 Обозначения, используемые в схеме базы данных

Полученная схема реляционной базы данных (РБД) приведена на Схема 3.

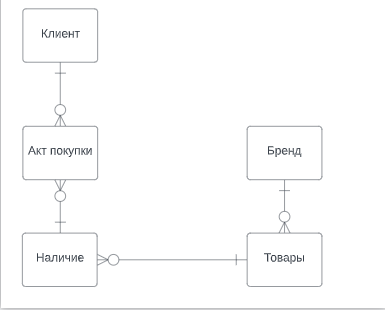


Схема 3 Схема РБД без нереализуемых связей

Составление реляционных выражений

Каждое реляционное отношение соответствует одной сущности (объекту ПрО) и в него вносятся все атрибуты этой сущности. Для каждого отношения определяются первичный ключ и внешние ключи (в соответствии со схемой БД). В том случае, если базовое отношение не имеет потенциальных ключей, вводится **суррогатный первичный ключ**, который не несёт смысловой нагрузки и служит только для идентификации записей.

Отношения приведены в Таблицах.Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так: N – числовой, C – символьный тип фиксированной длины, V – символьный тип переменной длины, B – логический тип, D – дата (этот тип имеет стандартную длину, зависящую от СУБД, поэтому она не указывается).

Для отношения Товары, стоит ввести первичный ключ id товара (артикул) (**product\_id**), который будет являться ID данного товара.

Таблица 1. Схема отношения Товары (products)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Id товара(артикул) | product\_id | V(8) | **Первичный ключ** |
| Название | product\_name | V(20) | Обязательное поле |
| Цена 1 единицы | price | N(6) | Обязательное поле |
| Материал | material | V(12) | Многозначное поле |
| Категория | category | V(13) | **Обязательное поле** |
| Id бренда | Brand\_id | N(6) | **Внешний ключ к brand** |

Потенциальным ключом отношения Клиент является поле Телефон. Но он занимает много места, поэтому стоит ввести суррогатный первичный ключ id клиента(**client\_id**).

Таблица 2 Схема отношения Клиенты (clients)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Id клиента | client\_id | N(6) | **Первичный ключ** |
| ФИО | name | V(80) | Обязательное поле |
| Телефон | phone | N(11) | Обязательное многозначное поле |
| Адрес | address | V(64) | Необязательное поле |

Потенциальным ключом отношения *Акт покупки* является поле Номер. Можно не вводить первичный ключ, так как на него никто не ссылается. Но нужно связать номер заказов (актов покупки) с клиентом, чтобы знать, кто что приобретал. Для этого добавим внешний ключ Id клиента.

Таблица 3 Схема отношения Акт покупки (purchase)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Номер акта | act\_num | N(6) | **Первичный ключ** |
| Дата | act\_date | D | Обязательное поле |
| Id клиента | client\_id | N(6) | **Внешний ключ к client** |
| Общая цена | price | N(6) | Обязательное поле |
| Количество | quantity | N(3) | Обязательное поле |
| Id товара(артикул) | product\_id | V(8) | **Внешний ключ к Товары** |
| Id наличия | availability\_id | N(6) | **Внешний ключ к Наличие** |

Потенциальным ключом отношения Бренд является поле Id бренда. Сделаем его первичным ключом, чтобы ссылаться на него с таблицы Товары.

Таблица 4 Схема отношения Бренд (brand)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Название | brand\_name | V(25) | Обязательное поле |
| Страна | country | V(10) | Обязательное поле |
| Дата основания | Brand\_date | D(6) | Обязательное поле |
| Id бренда | Brand\_id | N(6) | **Первичный ключ** |

Потенциальным ключом отношения Наличие является поле Id позиции. первичный ключ, так как на него никто не ссылается. Но нужно связать эту таблицу с таблицей Товары внешним ключом, которым и будет поле Id позиции.

Таблица 5 Схема отношения Наличие (availability)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Id наличия | availability\_id | N(6) | **Первичный ключ** |
| Id товара(артикул) | product\_id | V(8) | **Внешний ключ к products** |
| Количество | quantity | N(3) | Обязательное поле |
| Размер | size | V(8) | Обязательное многозначное поле |
| Цвет | color | V(10) | Обязательное поле |

Нормализация полученных отношений

Для приведения таблиц к **1НФ** разобьем сложные (составные) атрибуты на простые и вынесем многозначные атрибуты в отдельные отношения.

**Товары.** Многозначный атрибут Материал. Вынесем отдельное отношение Материалы. На отношение Материалы ссылаются другие таблицы, поэтому вводим первичный ключ. Также создадим таблицу с полями Номер материала и Id товара, чтобы показать, что у одного товара может быть много материалов. После нормализации это превращается в 3 отношения 6-8.

*Таблица 6. Схема отношения Материалы (materials)*

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Номер материала | material\_id | N(4) | **Первичный ключ** |
| Название материала | material\_name | V(32) | Обязательное поле |

*Таблица 7. Схема отношения Товары (products)*

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Id товара(артикул) | product\_id | V(8) | **Первичный ключ** |
| Название | product\_name | V(20) | Обязательное поле |
| Цена 1 единицы | price | N(6) | Обязательное поле |
| Категория | category | V(13) | **Обязательное поле, внешний ключ к category name** |
| Id бренда | Brand\_id | N(6) | **Внешний ключ к brand** |

*Таблица 8. Схема отношения Товары и материалы (products and materials)*

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Номер материала | material\_id | N(4) | **Внешний ключ к materials** |
| Id товара | product\_id | V(8) | **Внешний ключ к products** |
| Процент наличия материала | mat\_percent | N(3) | Обязательное поле |

Создадим справочную таблицу Название категории.

Таблица 9. Схема отношения Название категории (category name)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Название категории | Category\_name | V(13) | **Первичный ключ** |

**Клиенты.** Многозначный атрибут Номер. Вынесем отдельное отношение Номера. На отношение Материалы ссылаются другие таблицы, поэтому вводим первичный ключ. После нормализации это превращается в 2 отношения 10-11.

*Таблица 10. Схема отношения Номера телефонов (phone numbers)*

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Id клиента | client\_id | N(6) | Обязательное поле, **внешний ключ к client** |
| Телефон | phone | N(11) | Обязательное поле |

*Таблица 11. Схема отношения Клиент (client)*

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Id клиента | client\_id | N(6) | **Первичный ключ** |
| Фамилия | 2nd\_name | V(20) | Обязательное поле |
| Имя | name | V(20) | Обязательное поле |
| Отчество | patronym | V(20) | Необязательное поле |
| Адрес | address | V(64) | Необязательное поле |

**Наличие.** Многозначный атрибут Размер. Вынесем отдельное отношение Размеры, чтобы показать, что у определённой позиции может быть несколько различных размеров. После нормализации это превращается в 2 отношения 11-12.

*Таблица 12 Схема отношения Размеры (sizes)*

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Размер | size | V(8) | **Первичный ключ** |

*Таблица 13 Схема отношения Цвета (colors)*

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Цвет | color | V(10) | **Первичный ключ** |

*Таблица 14 Схема отношения Наличие (availability)*

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Размер | size | V(8) | **Внешний ключ к Размеры** |
| Id позиции | pos\_id | N(5) | **Первичный ключ** |
| Id товара | product\_id | V(8) | **Внешний ключ к Товары (products)** |
| Количество | quantity | N(3) | Обязательное поле |
| Цвет | color | V(10) | **Внешний ключ к Цвета** |

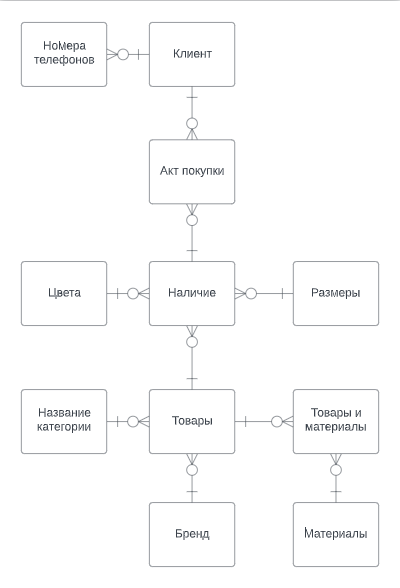
****

Схема 4 Окончательная схема БД проектной организации

Определение дополнительных ограничений целостности

Перечислим ограничения целостности, которые не указаны в табл. 1-12.

1. При покупке определённого товара определённого размера и цвета общее количество должно уменьшаться на количество, указанное в акте.
2. Нельзя купить товар, которого нет в наличии, то есть то, у которого наличие равно 0.

Ограничение 1 нельзя реализовать в схеме отношения. В БД подобное ограничение целостности будет реализован в программе (через специальную процедуру контроля данных – триггер).

Описание групп пользователей и прав доступа

Опишем для каждой группы пользователей права доступа к каждой таблице (Таблица 15). Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта. Используются следующие сокращения:

Таблица 15. Обозначения прав доступа

|  |  |  |
| --- | --- | --- |
| Обозначение | Расшифровка | Альтернативное обозначение |
| I (insert) | добавление данных | C (create) |
| S (select) | чтение данных | R (read) |
| U (update) | модификация данных | U (update) |
| D (delete) | удаление данных | D (delete) |

Таблица 16. Права доступа к таблицам для групп пользователей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблицы | Группы пользователей (роли) | | | |
| Владелец | Старший менеджер | Менеджер | Клиент |
| Материалы | SUID | SUI | S | S |
| Товары | SUID | SUID | S | S |
| Товары и материалы | SUID | SUID | S | S |
| Название категории | SUID | SUID | S | S |
| Номера телефонов | SUID | SUID | SUID | S\* |
| Клиент | SUID | SUID | SUID | S\* |
| Размеры | SUID | SUID | S | S |
| Цвета | SUID | SUID | S | S |
| Наличие | SUID | SUID | S | S |
| Акт покупки | SUID | SI | SI |  |
| Бренд | SUID | S | S | S |

\*(только свои

Реализация проекта базы данных

Создание таблиц

Отношение materials (Материалы):

CREATE TABLE materials (

material\_id numeric(6) PRIMARY KEY,

material\_name varchar(32) NOT NULL

);

Отношение category\_name (Название категории):

CREATE TABLE category\_name (

category\_name varchar(13) PRIMARY KEY

);

Отношение client (Клиент):

CREATE TABLE client (

client\_id numeric(6) PRIMARY KEY,

snd\_name varchar(20) NOT NULL,

Name varchar(20) NOT NULL,

Patronym varchar(20),

address varchar(64)

);

Отношение sizes (Размеры):

CREATE TABLE sizes (

size varchar(8) PRIMARY KEY

);

Отношение colors (Цвета):

CREATE TABLE colors (

color varchar(10) PRIMARY KEY

);

Отношение phone numbers (Номера телефонов):

CREATE TABLE phone\_numbers (

client\_id numeric(6),

phone numeric(11) NOT NULL,

FOREIGN KEY(client\_id) REFERENCES client(client\_id)

);

Отношение brand (Бренд):

CREATE TABLE brand (

brand\_name varchar(25) NOT NULL,

country varchar(10) NOT NULL,

brand\_date date NOT NULL,

brand\_id numeric(6) PRIMARY KEY

);

Отношение products (Товары):

CREATE TABLE products (

product\_id varchar(8) PRIMARY KEY,

product\_name varchar(20) NOT NULL,

price numeric(6) NOT NULL,

category varchar(13) NOT NULL,

brand\_id numeric(6) NOT NULL,

FOREIGN KEY(category) REFERENCES category\_name(category\_name),

FOREIGN KEY(category) REFERENCES category\_name(category\_name),

FOREIGN KEY(brand\_id) REFERENCES brand(brand\_id)

);

Отношение products and materials (Товары и материалы):

CREATE TABLE products\_and\_materials(

material\_id numeric(4) NOT NULL,

product\_id varchar(8) NOT NULL,

mat\_percent numeric(3) NOT NULL,

FOREIGN KEY(material\_id) REFERENCES materials(material\_id),

FOREIGN KEY(product\_id) REFERENCES products(product\_id)

);

Отношение availability (Наличие):

CREATE TABLE availability(

size varchar(8) NOT NULL,

pos\_id numeric(5) PRIMARY KEY,

product\_id varchar(8) NOT NULL,

quantity numeric(3) NOT NULL,

color varchar(10) NOT NULL,

FOREIGN KEY(size) REFERENCES sizes(size),

FOREIGN KEY(product\_id) REFERENCES products(product\_id),

FOREIGN KEY(color) REFERENCES colors(color)

);

Отношение purchase (Акт покупки):

CREATE TABLE purchase(

act\_num numeric(6) PRIMARY KEY,

act\_date date NOT NULL,

client\_id numeric(6) NOT NULL,

price numeric(6) NOT NULL,

quantity numeric(3) NOT NULL,

product\_id varchar(8) NOT NULL,

availability\_id numeric(6) NOT NULL,

FOREIGN KEY(client\_id) REFERENCES client(client\_id),

FOREIGN KEY(product\_id) REFERENCES products(product\_id),

FOREIGN KEY(availability\_id) REFERENCES availability(pos\_id)

);

Начальный состав данных для справочных таблиц

1. Таблица Название категории:

*insert into category\_name values (’Джинсы’);*

*insert into category\_name values (’Футболка’);*

*insert into category\_name values (’Худи’);*

*insert into category\_name values (’Свитшот’);*

*insert into category\_name values (’Носки’);*

*insert into category\_name values (’Бейсболка’);*

*insert into category\_name values (’Штаны’);*

*insert into category\_name values (’Брюки’);*

*insert into category\_name values (’Кардиган’);*

*insert into category\_name values (’Толстовка’);*

*insert into category\_name values (’Куртка’);*

*insert into category\_name values (’Ветровка’);*

*insert into category\_name values (’Бомбер’);*

*insert into category\_name values (’Жилетка’);*

*insert into category\_name values (’Пуховик’);*

*insert into category\_name values (’Тренч’);*

*insert into category\_name values (’Пончо’);*

Создание представлений (готовых запросов)

1. Товары, которые находятся в наличии

create view avail\_products as

select product\_name from products

join availability on abailability.product\_id = products.product\_id

where availability.quantity>0

1. Клиенты и их акты покупки

create view clients\_and\_purchases as

select 2nd\_name, name, patronym from client

join purchase on purchase.client\_id = client.client\_id

1. Бренды в наличии и их страна происхождения

create view brands as

select brand\_name, country from brand

1. Товары и материалы, из которых они сделаны

create view products\_materials as

select product\_name, material\_name from products\_and\_materials pam

join products on pam.product\_id = products.product\_id

join materials on materials.material\_id = pam.material\_id

Назначение прав доступа

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Представления | Группы пользователей | | | |
| Владелец | Старший менеджер | Менеджер | Клиент |
| Товары, которые находятся в наличии | SUID | SUI | SU | S |
| Клиенты и их акты покупки | SUID | SUID | SUI | S\* |
| Бренды в наличии и их страна происхождения | SUID | SUID | SUI | S |
| Товары и материалы | SUID | SUID | SUI | S |

Создание триггеров

create or replace function decrease\_product() returns trigger as $$

declare

cnt integer;

begin

SELECT price into cnt FROM products WHERE NEW.product\_id = product\_id;

if cnt is NULL then

raise exception 'Такого нет на складе!';

else

UPDATE availability SET quantity=quantity-1 WHERE NEW.product\_id = product\_id;

end if;

return NEW;

end;

$$ LANGUAGE plpgsql;

create trigger tr\_insert\_purchase

before insert on purchase

for each row

execute procedure decrease\_product();

Разработка стратегии резервного копирования

Интенсивность обновления разработанной базы данных низкая, поэтому

для обеспечения сохранности вполне достаточно проводить полное резервное копирование БД раз в день (перед окончанием рабочего дня).

Для разработанной БД нет необходимости держать сервер включенным

круглосуточно, поэтому можно создать соответствующее задание

операционной системы, которое будет автоматически запускаться перед выключением сервера.