Лабораторная работа №1

“Построение модели данных”

Выполнил студент группы Б20-505
Сорочан Илья

Московский Инженерно-Физический Интститут
Москва 2023

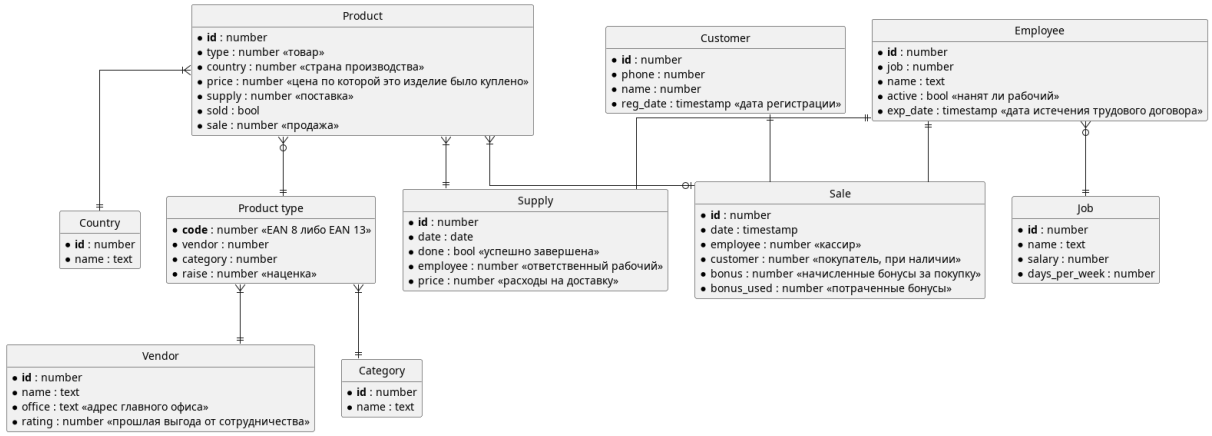
1 Введение

Рассматривается небольшой канцелярский магазин. Задача бд: хранить следующую информацию:

- краткая сводка по товарам (цены и общая информация);
- список работающих сотрудников;
- поставки товаров;
- программа лояльности для покупателей.

2 Архитектура БД

Спроектированная БД имеет следующий вид:



2.1 Товар

Товар в бд представляется двумя сущностями. ”Product”является конкретной товарной единицой. Он так же содержит базовую информацию о товаре, такую как цена закупки, айди поставки, продан ли он и если да, то ссылка на объект продажи и тд. ”Product type тип товара, определяемый штрих-кодом.

Здесь следует немного отступить и объяснить зачем такое разделение и почему оно выполняется именно таким образом. Очевидно, что у каждого производителя есть как тип товара, так и конкретные единицы, которые поставляются. Тип товара регулируется на законодательном уровне при помощи выдачи государством штрих-кодов. Часть штрих-кода хранит информацию о производителе, часть о номере товара среди ассортимента производителя.

В пример можно привести производителя ”каждый день”. У него есть много канцелярских товаров, в том числе и ручек с различным форм-фактором. Даже если цвет у этих ручек будет один их штрих-коды будут отличатся, так как на законодательном уровне это разные товары. Это и есть тип товара. А конкретные единицы - конкретные ручки.

2.2 Поставки и рабочие

Для отслеживания поставок используется сущность ”Supply”. Она хранит все планируемые и проведенные поставки. Дополнительно о каждой из них хранится дата, расходы и ответственный сотрудник (человек от магазина, отвечающий за качество и количество прибывшего товара).

Информация о рабочих хранится в таблице ”Employee”. После увольнения рабочих из базы данных не удаляют. Это необходимо для того, что бы у старых поставок (и покупок), оставался ответственный сотрудник, даже если позже он будет уволен. Дополнительно хранится дата истечения трудового договора. Она полезна как в целом для понимания когда следует обновить контракт с текущим рабочим, так и для удаления слишком ”старых”ответственных за поставки.

Таблица ”Job”хранит информацию о должности.

2.3 Продажи и покупатели

Все покупки хранятся в таблице ”Sale”. Здесь присутсвует ссылка на рабочего (кассира), дата, покупатель. Важно отметить, что баллы, начисляемые за покупку по программе лояльности хранятся здесь же. Это спорное решение, оно облегчает добавление новых записей (не нужно модифицировать ”Customer”), но осложняет списание.

”Customer”хранит информацию о покупателе, которая может быть полезна при массовых рассылках.

2.4 Связи и NF

В первую очередь стоит отметить, что на рисунке **для всех сущностей** жирным обозначер основной ключ, который уникален для каждой записи и однозначно соответствует всем остальным записям.

У каждого ”Product type”есть ”Vendor”и ”Category так как все типы товаров имеют производителей и категорию, согласно которой они могут располагатся в магазине. При этом одна производитель/категория могут соответствать нескольким типам товара.

Товар (”Product”), обязательно имеет тип, страну производства (что можно вынести в тип товара, но не так важно) и поставку, в которой он был привезен в магазин. Так же, если товар продан, то он будет находиться в Sale.

Из соотношений один ко многим еще присутсвует отношение ”Employee”и ”Job но здесь все просто. На одной вакансии могут работать несколько работников.

3 Заключение

В результате данной лабораторной работы:

- была выбрана рабочая область;
- создана архитектура бд и приведена к нормальным формам;
- разработан скрипт, создающий бд.

Приложение А

Использованные программные коды

SQL скрит для создания БД:

```
CREATE TABLE vendor (
    id INTEGER PRIMARY KEY,
    name TEXT,
    office TEXT,
    rating INTEGER
);

CREATE TABLE category (
    id INTEGER PRIMARY KEY,
    name TEXT
);

CREATE TABLE product_type (
    code INTEGER PRIMARY KEY,
    vendor INTEGER,
    category INTEGER,
    raise INTEGER,
    FOREIGN KEY(vendor) REFERENCES vendor(id),
    FOREIGN KEY(category) REFERENCES category(id)
);

CREATE TABLE country (
    id INTEGER PRIMARY KEY,
    name TEXT
);

CREATE TABLE product (
    id INTEGER PRIMARY KEY,
    type INTEGER,
    country INTEGER,
    price INTEGER,
    supply INTEGER,
    sold BOOLEAN,
    sale INTEGER,
    FOREIGN KEY(type) REFERENCES product_type(code),
    FOREIGN KEY(country) REFERENCES country(id),
    FOREIGN KEY(supply) REFERENCES supply(id),
    FOREIGN KEY(sale) REFERENCES sale(id)
);

CREATE TABLE supply (
    id INTEGER PRIMARY KEY,
    date DATE,
    done BOOLEAN,
    employee INTEGER,
    price INTEGER,
```

```
    FOREIGN KEY(employee) REFERENCES employee(id)
);

CREATE TABLE employee (
    id INTEGER PRIMARY KEY,
    job INTEGER,
    name TEXT,
    active BOOLEAN,
    exp_date TIMESTAMP,
    FOREIGN KEY(job) REFERENCES job(id)
);

CREATE TABLE job (
    id INTEGER PRIMARY KEY,
    name TEXT,
    salary INTEGER,
    days_per_week INTEGER
);

CREATE TABLE sale (
    id INTEGER PRIMARY KEY,
    date TIMESTAMP,
    employee INTEGER,
   customer INTEGER,
    bonus INTEGER,
    bonus_used INTEGER,
    FOREIGN KEY(employee) REFERENCES employee(id),
    FOREIGN KEY(customer) REFERENCES customer(id)
);

CREATE TABLE customer (
    id INTEGER PRIMARY KEY,
    phone INTEGER,
    name TEXT,
    reg_date TIMESTAMP
);
```