ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»»

Проект на тему:

«Разработка программы или системы, взаимодействующей с реляционной базой данных»

**Выполнил:**

Моисеев Эрчимэн Александрович

**Преподаватель:**

          Брейман Александр Давидович

Москва, 2023

Пояснительная записка

Для одной фирмы разрабатывается специальная программа, которая будет взаимодействовать с реляционной базой данных. Программа и база данных нужны для регулирования и учёта складских запасов одной крупной фирмы по хранению, продаже и перепродаже самых разнообразных материалов (data warehouse about warehouse to a warehouse company). Система будет фиксировать поступление и убыль материалов. Программа будет использоваться для отслеживания этих параметров, а также для контроля остатков на складе и формирования отчетности для руководства компании.

Функциональные требования

1. Страница регистрации сотрудника фирмы
2. Страница входа в систему
3. Страница, на которой написано, как работать с программой (FAQ) и вести учет поступаемого и отпускаемого товара
4. Страница профиля сотрудника:
5. кнопка «выбрать товар»
6. функция, которая создает и редактирует страницу с материалами
7. Страница с поставщиками компании и их контактными данными
8. Страница с информацией о складах и хранимыми там товарами
9. Страница «Материал» со следующими частями:
10. ID товара
11. Название
12. Описание
13. Местоположение товара на складе
14. Размеры материалов
15. Дата поступления материала
16. Дата отбытия материалов (если материал был выкуплен у фирмы)
17. Цена, по которой был куплен материал
18. Цена, по которой материал может быть продан
19. Кнопка «Отпустить товар» (и там же поле, в котором указывается, количество отгружаемого товара и цена)

Нефункциональные требования

1. Программа должна обеспечивать безопасность данных и иметь защиту от попыток несанкционированного доступа

2. Интерфейс программы должен быть интуитивно понятным и удобным для сотрудников компании

3. Программа должна обеспечивать быструю обработку данных и эффективное выполнение запросов

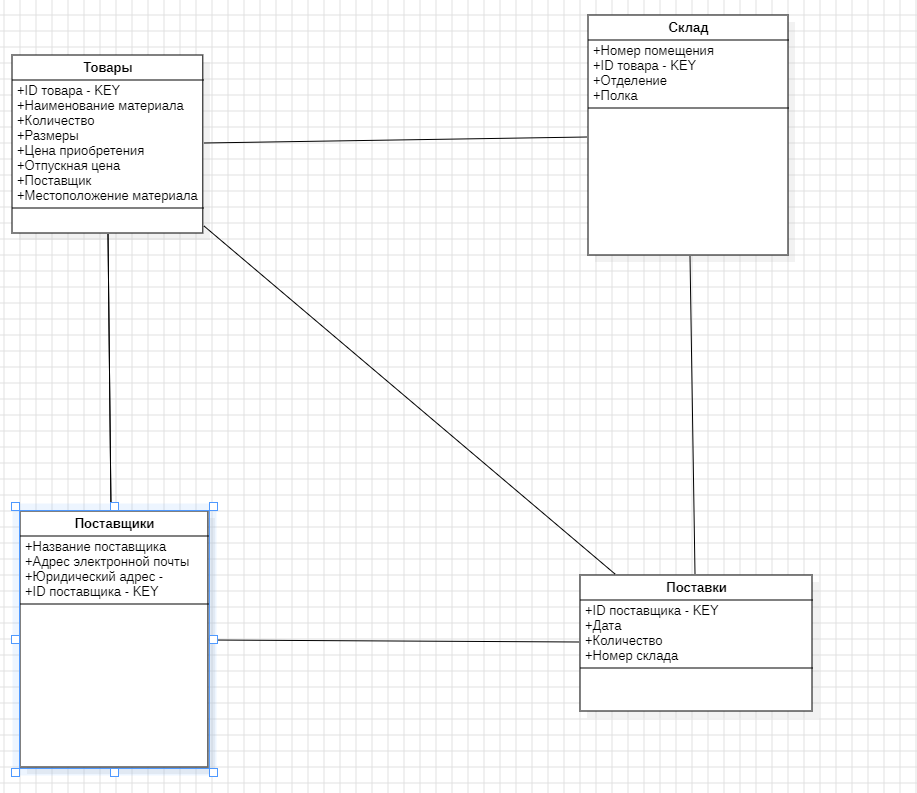
4. Система должна работать на разных платформах (сотрудники могут пользоваться программой и на своем ноутбуке, и на своем телефоне)

5. Желательно, чтобы программа поддерживала определенные иностранные языки (к примеру, китайский язык, так как планируется открытие складов в Китае)

Сущности и атрибуты

1. Goods(GID, GNAME, QUANTITY, SIZE, BUYINGPRICE, SELLINGPRICE, PROVIDER)
2. PROVIDERS(PID, PNAME, EMAIL, ADDRESS)
3. SUPPLIES(PID, DATE, QUANTITY, WARENUMBER)
4. WAREHOUSE(ROOMNUM, GID, BRANCH, SHELF)

Предварительная схема базы данных



Текстовые ограничения на данные

Имя сотрудника: ограничение от 2 до 50 символов

Пароль: ограничение от 8 до 16 символов

Логин сотрудника: ограничение от 6 до 20 символов

ID товара: уникальность, целые числа

ID поставщика: уникальность, целые числа

Название товара: ограничение от 2 до 100 символов

Описание товара: ограничение от 2 до 255 символов

Местоположение товара на складе: ограничение от 2 до 100 символов

Размеры материала: допустимые значения в формате: длина, ширина, высота (например, 10x5x3)

Дата поступления материала: формат даты (день-месяц-год)

Дата отбытия материалов: формат даты, может быть пустым, если материал еще не ушел со склада

Цена, по которой был куплен материал: числовой формат, включая дробные значения (например, 12.50)

Цена, по которой материал может быть продан: числовой формат, включая дробные значения (например, 12.50)

Кнопка "Отпустить товар": отгружаемое количество - целое число, цена - числовой формат, включая дробные значения (например, 12.50)

Страница регистрации сотрудника фирмы:

- Требование: Логин и пароль должны быть уникальными для каждого сотрудника

- Ограничение: Логин должен содержать от 6 до 20 символов, пароль - от 8 до 16 символов

2. Страница входа в систему:

- Требование: Доступ должен быть только для зарегистрированных сотрудников

- Ограничение: Проверка соответствия введенного логина и пароля с данными из базы

3. Страница FAQ:

- Требование: Актуальная информация о работе с программой и учете товаров

- Ограничение: Страница должна быть доступна только авторизованным пользователям

4. Страница профиля сотрудника:

- Требование: Возможность выбора товара и редактирования страницы с материалами

- Ограничение: Разрешение на редактирование только собственных данных

5. Страница с поставщиками компании и их контактными данными:

- Требование: Подробная информация о поставщиках

- Ограничение: Необходимость хранения уникального ID для каждого поставщика, контроль корректности вводимых данных

6. Страница с информацией о складах и хранимыми товарами:

- Требование: Полная информация о каждом складе и товарах, которые хранятся на нем

- Ограничение: Точное указание местоположения товаров на складе

7. Страница «Материал»:

- Требование: Вся информация о каждом материале и его движении

- Ограничение: Проверка уникальности ID товара, ограничение ввода корректных дат и цен, контроль заполняемых полей "Отпустить товар".

Функциональные зависимости:

1. Страница регистрации сотрудника фирмы ➡ Уникальность логина и пароля для каждого сотрудника

2. Страница входа в систему ➡ Доступ только для зарегистрированных сотрудников

3. Страница FAQ ➡ Доступ для авторизованных пользователей

4. Страница профиля сотрудника ➡ Возможность выбора товара и редактирования страницы

5. Страница с поставщиками ➡ Уникальный ID для каждого поставщика

6. Страница с информацией о складах ➡ Точное местоположение товаров на складе

8. Страница "Материал" ➡ Уникальность ID товара, корректные даты и цены

Многозначные зависимости:

1. Регистрация сотрудника (логин) ➡ Требование уникальности

2. Регистрация сотрудника (логин) ➡ Ограничение по количеству символов

3. Вход в систему ➡ Проверка соответствия логина и пароля

4. Профиль сотрудника ➡ Разрешение на редактирование только собственных данных

5. Страница с поставщиками ➡ Обязательное хранение уникального ID и контроль корректности вводимых данных

6. Страница с информацией о складах ➡ Точное местоположение товаров

7. Страница "Материал" ➡ Уникальность ID, корректные даты и цены, контроль полей "Отпустить товар"

Нормализация

Функциональные зависимости:

1. GID -> GNAME, QUANTITY, SIZE, BUYINGPRICE, SELLINGPRICE, PROVIDER

2. PID -> PNAME, EMAIL, ADDRESS

3. (PID, DATE) -> QUANTITY, WARENUMBER

4. GID -> ROOMNUM, BRANCH, SHELF

На основании этих функциональных зависимостей мы можем выполнить нормализацию.

Представленные функциональные зависимости показывают, что некоторые данные могут быть выделены в отдельные таблицы для достижения нормализации.

1. Goods(GID, GNAME, QUANTITY, SIZE, BUYINGPRICE, SELLINGPRICE, PROVIDER) - здесь GID является первичным ключом

2. Providers(PID, PNAME, EMAIL, ADDRESS) - здесь PID является первичным ключом

3. Supplies(PID, DATE, QUANTITY, WARENUMBER) - полагаю что PID в данной таблице также связан с поставщиками и обозначает поставщика, поэтому Supplies(PID, DATE) может быть выделено в отдельную таблицу SuppliesDetails(Composite key (PID, DATE), QUANTITY, WARENUMBER) и оба атрибута в этой таблице становятся внешними ключами, ссылающимися на поставщиков.

4. Warehouse(ROOMNUM, GID, BRANCH, SHELF) - в данной таблице GID также может быть внешним ключом, ссылается на товары из таблицы Goods.

Таким образом, нормализованные таблицы выглядели следующим образом:

1. Goods(GID, GNAME, QUANTITY, SIZE, BUYINGPRICE, SELLINGPRICE, PROVIDER) - GID - первичный ключ

2. Providers(PID, PNAME, EMAIL, ADDRESS) - PID - первичный ключ

3. Supplies(PID, DATE) - внешние ключи (PID, DATE) ссылается на Providers, далее

4. SuppliesDetails(Composite key (PID, DATE), QUANTITY, WARENUMBER) - ключ (PID, DATE)

5. Warehouse(ROOMNUM, GID, BRANCH, SHELF) - GID - внешний ключ, ссылается на Goods.

SQL DDL:

CREATE TABLE Goods(

GID INT PRIMARY KEY,

GNAME VARCHAR(255),

BUYING PRICE DECIMAL,

SELLIN PRICE DECIMAL

QUANITY INT

SIZE INT

PROVIDER VARCHAR(255),

);

CREATE TABLE PROVIDERS(

PID INT PRIMARY KEY,

PNAME VARCHAR(255),

EMAIL VARCHAR(255)

ADDRESS VARCHAR(255

);

CREATE TABLE SUPPLIES(

PID INT PRIMARY KEY,

DATE DATE);

CREATE TABLE SuppliesDetails(

(PID, DATE) COMPOSITE KEY,

QUANTITY INT,

WARENUMBER INT);

CREATE TABLE Warehouse(

ROOMNUM,

GID INT PRIMARY KEY,

BRANCH INT,

SHELF INT);

SQL DML

SELECT \* FROM GOODS;

INSERT INTO SUPPLIES(PID, DATE, GNAME, QUANTITY, PROVIDER) VALUES (1, '2022-01-01', 1, 100, 1);

UPDATE GOODS SET QUANTITY = QUANTITY - 50 WHERE ID = 1;

DELETE FROM SUPPLIES WHERE PID = 1;

Группировка запросов в транзакции:

BEGIN TRANSACTION;

INSERT INTO SUPPLIES(PID, DATE, GNAME, QUANTITY, PROVIDER) VALUES (2, '2022-02-01', 2, 200, 2);

UPDATE GOODS SET QUANTITY = QUANTITY + 200 WHERE ID = 2;

COMMIT;