ФМИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ  
  
Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королёва»

(Самарский университет)  
  
Факультет информатики  
Кафедра программных систем  
  
Дисциплина  
**Практикум на ЭВМ  
  
  
  
ОТЧЕТ**  
Разработка базы данных  
Вариант № 15

Студенты: Гижевская В.Д.

Петрова А.А.  
Группа: 6313 – 020302D  
  
Преподаватель: Попова-Коварцева Д. А.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2020

Тема 1. **Проектирование базы данных**

1. *Концептуальное проектирование.* Разработать ER-модель предметной области, описанной в проекте. Каждую сущность охарактеризовать набором атрибутов.
2. *Логическое проектирование.* Преобразовать ER-модель в реляционную модель. С помощью CASE-средства разработать логическую модель БД.
3. *Физическое проектирование.* Создать в выбранной СУБД разработанную БД.
4. Оформите отчет о выполнении лабораторной работы, включающий описание логической модели данных.

Задание

Интернет-магазины реализуют потребителям бытовую технику (утюги, электрочайники, кухонные комбайны и др.) разных моделей известных фирм- производителей (Philips, Bosh, Mulinex и др.). Заказы осуществляются клиентами в интернет-магазинах в любое время суток. После подтверждения заказа клиентом по телефону курьер доставляет ему товар по указанному адресу.

Необходимо спроектировать базу данных **ИНТЕРНЕТ-ПРОДАЖИ**, информация которой будет использоваться для анализа спроса потребителей на конкретные модели товаров разных производителей, динамики реализации товаров в интернет-магазинах за определённые интервалы времени, сравнения условий доставки товаров в разных магазинах и др.

В БД должна храниться информация:

* об ИНТЕРНЕТ-МАГАЗИНАХ: *код магазина, электронный адрес, оплата доставки*

(Да/Нет);

* ТОВАРАХ: *код товара, название товара, фирма, модель, технические характеристики, цена* (руб.)*, гарантийный срок, изображение*;
* ЗАКАЗАХ: *код заказа, код магазина, код товара, дата заказа, время заказа, количество, Ф.И.О. клиента, контактный телефон, подтверждение заказа* (Да/Нет)*;*
* ДОСТАВКЕ: *код заказа, дата доставки, время доставки, адрес доставки, Ф.И.О. клиента, Ф.И.О. курьера.*

При проектировании БД необходимо учитывать следующее:

* товар может продаваться в *нескольких* интернет-магазинах. Интернет- магазин может предлагать к продаже *несколько* товаров;
* товар может быть связан с *несколькими* заказами. Заказ связан с *одним* товаром;
* в интернет-магазине могут осуществляться *несколько* заказов. Заказ связан с

*одним* интернет-магазином;

* заказ подлежит *одной* доставке. Доставка связана с *одним* заказом.

Кроме того следует учесть:

* каждый товар *обязательно реализуется* через интернет-магазины. Каждый интернет-магазин *обязательно* реализует товары;
* товар *не обязательно* может быть заказан. Каждый заказ *обязательно* связан с товаром;
* магазин *не обязательно* может иметь заказы. Каждый заказ *обязательно*

связан с определённым интернет-магазином;

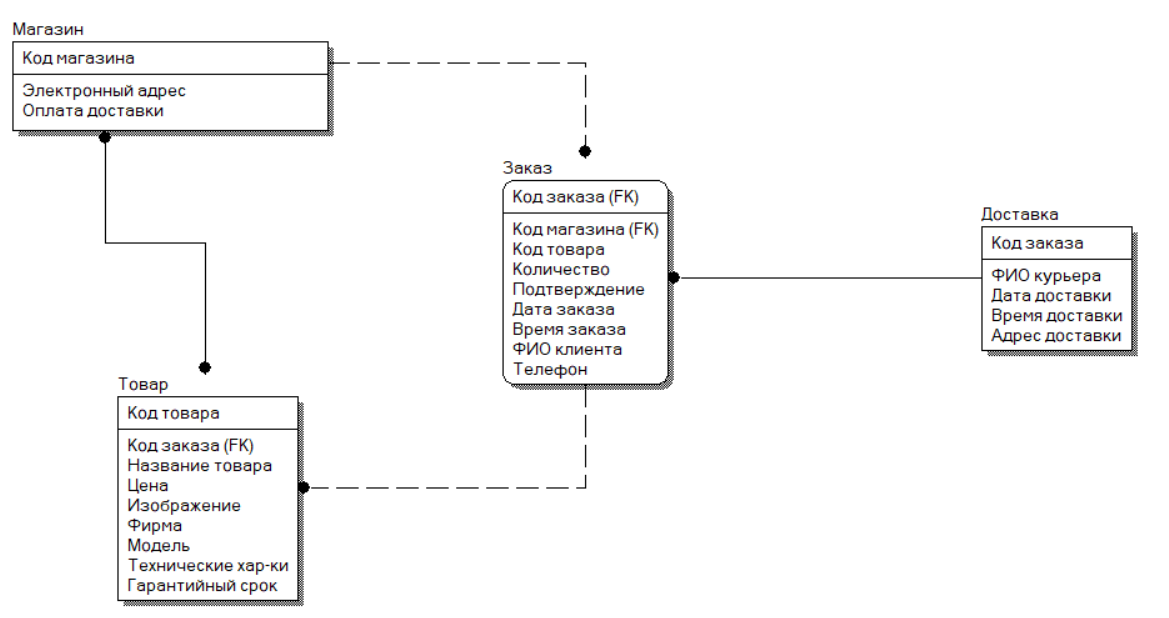
* заказ *не обязательно* может быть доставлен. Каждая доставка товара *обязательно* связана с заказом.

Ход выполнения работы

1. Первичная схема базы данных



1. Логическая модель базы данных



Сущность Магазин. Атрибуты:

* Код магазина (ключ)
* Электронный адрес
* Оплата доставки

Сущность Товар. Атрибуты:

* Код товара (ключ)
* Код заказа (FK)
* Название товара
* Цена
* Изображение
* Фирма
* Модель
* Технические характеристики
* Гарантийный срок

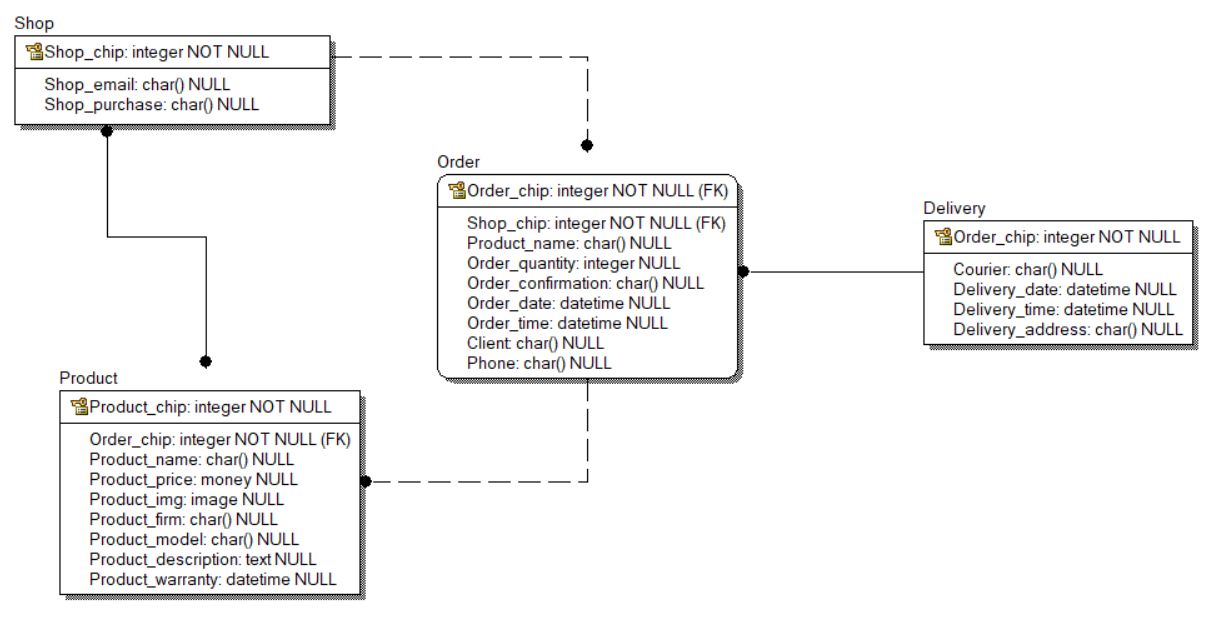
Сущность Заказ. Атрибуты:

* Код заказа (ключ)
* Код магазина (FK)
* Код товара
* Количество
* Подтверждение
* Дата заказа
* Время заказа
* ФИО клиента
* Телефон

Сущность Доставка. Атрибуты:

* Код заказа (ключ)
* ФИО курьера
* Дата доставки
* Время доставки
* Адрес доставки

1. Физическая модель



CREATE TABLE Delivery

(

Order\_chip int NOT NULL ,

Courier varchar(50) NULL ,

Delivery\_date date NULL ,

Delivery\_name varchar(18) NULL ,

Delivery\_adress varchar(50) NULL

)

go

ALTER TABLE Delivery

ADD CONSTRAINT XPKDelivery PRIMARY KEY CLUSTERED (Order\_chip ASC)

go

CREATE TABLE Magazine

(

Shop\_chip int NOT NULL ,

Product\_chip int NOT NULL

)

go

ALTER TABLE Magazine

ADD CONSTRAINT XPKMagazine PRIMARY KEY CLUSTERED (Shop\_chip ASC,Product\_chip ASC)

go

CREATE TABLE Orders

(

Order\_chip int NOT NULL ,

Shop\_chip int NULL ,

Product\_name varchar(50) NULL ,

Order\_quantity int NULL ,

Order\_confirmation varchar(18) NULL ,

Order\_date date NULL ,

Order\_time varchar(18) NULL ,

Client varchar(30) NULL ,

Phone varchar(30) NULL

)

go

ALTER TABLE Orders

ADD CONSTRAINT XPKOrder PRIMARY KEY CLUSTERED (Order\_chip ASC)

go

CREATE TABLE Product

(

Product\_chip int NOT NULL ,

Product\_name varchar(18) NULL ,

Product\_price int NULL ,

Product\_img varchar(18) NULL ,

Product\_firm varchar(18) NULL ,

Product\_model varchar(18) NULL ,

Product\_description varchar(100) NULL ,

Product\_warrantly varchar(18) NULL ,

Order\_chip int NULL

)

go

ALTER TABLE Product

ADD CONSTRAINT XPKProduct PRIMARY KEY CLUSTERED (Product\_chip ASC)

go

CREATE TABLE Shop

(

Shop\_email varchar(50) NULL ,

Shop\_purchase varchar(18) NULL ,

Shop\_chip int NOT NULL

)

go

ALTER TABLE Shop

ADD CONSTRAINT XPKShop PRIMARY KEY CLUSTERED (Shop\_chip ASC)

go

ALTER TABLE Delivery

ADD CONSTRAINT R\_8 FOREIGN KEY (Order\_chip) REFERENCES Orders(Order\_chip)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Magazine

ADD CONSTRAINT R\_6 FOREIGN KEY (Shop\_chip) REFERENCES Shop(Shop\_chip)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Magazine

ADD CONSTRAINT R\_7 FOREIGN KEY (Product\_chip) REFERENCES Product(Product\_chip)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Orders

ADD CONSTRAINT R\_1 FOREIGN KEY (Shop\_chip) REFERENCES Shop(Shop\_chip)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Product

ADD CONSTRAINT R\_3 FOREIGN KEY (Order\_chip) REFERENCES Orders(Order\_chip)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go