МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)   
  
  
Институт информатики и кибернетики  
Кафедра программных систем  
  
Дисциплина  
**Практикум на ЭВМ  
  
  
  
ОТЧЕТ**по лабораторному практикуму  
  
Вариант №15

Студенты: Атякшев С. А., Матвеев В. Д.  
Группа: 6301-020302D  
  
Преподаватель: Повова-Коварцева Д. А.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc153808304)

[ХОД РАБОТЫ 5](#_Toc153808305)

[Часть 1. Проектирование базы данных 5](#_Toc153808306)

[Часть 2. Конструирование запросов 8](#_Toc153808307)

[Часть 3. Разработка интерфейса пользователя для работы с данными в БД. Создание форм. 12](#_Toc153808308)

[Часть 4. Конструирование отчета 16](#_Toc153808309)

[ПРИЛОЖЕНИЕ А 18](#_Toc153808310)

[ПРИЛОЖЕНИЕ B 21](#_Toc153808311)

[ПРИЛОЖЕНИЕ C 23](#_Toc153808312)

# ВВЕДЕНИЕ

Проект «Интернет-продажи»

Интернет-магазины реализуют потребителям бытовую технику (утюги, электрочайники, кухонные комбайны и др.) разных моделей известных фирмпроизводителей (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 определенные 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 определенным интернет-магазином;
* заказ не обязательно может быть доставлен. Каждая доставка товара обязательно связана с заказом.

# ХОД РАБОТЫ

## Часть 1. Проектирование базы данных

1. Разработать ER-модель предметной области, описанной в проекте в соответствии с вариантом задания. Каждую сущность охарактеризовать набором атрибутов, при необходимости добавить атрибуты в сущности. Установить связи между сущностями.
2. Создать в выбранной СУБД разработанную БД. Наполнить таблицы записями (не менее 15 записей для каждой таблицы).

При выполнении работы была разработана физическая модель. На рисунке 1.1 изображена физическая модель базы данных.

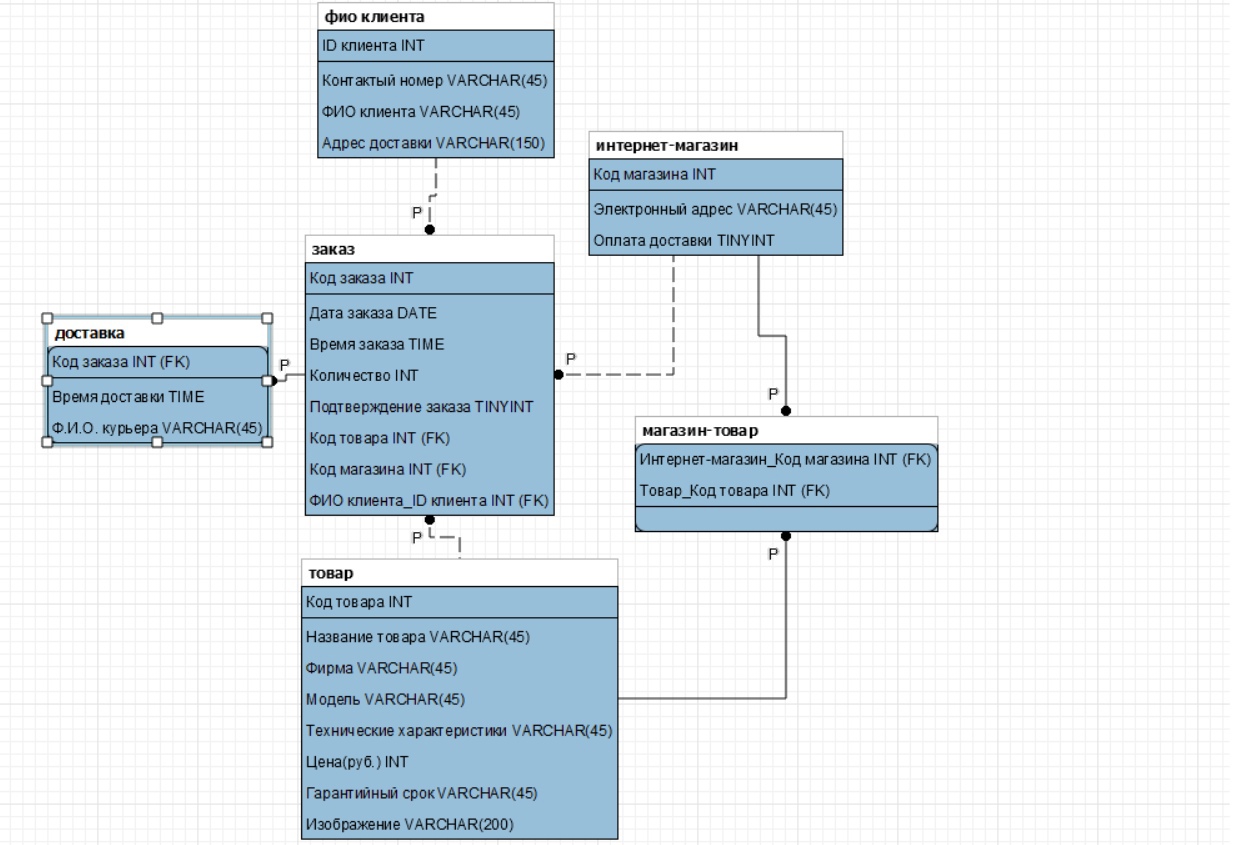


Рисунок 1.1 – Физическая модель

Домены атрибутов сущностей:

1. Сущность заказ:

* код заказа – int(pk);
* дата заказа – date;
* время заказа – time;
* количество – int;
* подтверждение заказа – tinyint;
* код товара – int(fk);
* код магазина – int(fk);
* ФИО клиента\_ID клиента – int(fk).

1. Сущность товар:

* код товара – int(pk);
* название товара – varchar(45);
* фирма – varchar(45);
* модель – varchar(45);
* технические характеристики – varchar(45);
* цена(руб.) – int;
* гарантийный срок – varchar(45);
* изображение – varchar(200).

1. Сущность фио клиента:

* ID клиента – int(pk);
* контактый номер – varchar(45);
* ФИО клиента – varchar(45);
* адрес доставки – varchar(150).

1. Сущность доставка:

* код заказа – int(pk, fk);
* время доставки – time;
* Ф.И.О. курьера – varchar(45).

1. Сущность интернет-магазин:

* код магазина – int(pk);
* электронный адрес – varchar(45);
* оплата доставки – tinyint.

1. Сущность магазин-товар:

* интернет-магазин\_Код магазина – int(pk, fk);
* товар\_Код товара – int(pk, fk).

Скрипт создания и наполнения БД в СУБД приведен в приложении А.

## Часть 2. Конструирование запросов

1. Сформулировать и сконструировать в выбранной СУБД запросы к БД, которые будут необходимы для предметной области (в соответствии с вариантом задания):

* запрос на выборку избранных полей таблицы, с использованием синонима (алиаса) и сортировкой записей (ORDER BY);
* запрос с использованием сортировки (ORDER BY) и группировки (GROUP BY);
* запрос с использованием предложения DISTINCT;
* запрос с использованием операций сравнения;
* запросы для предикатов: IN, BETWEEN, LIKE, IS NULL;
* запросы с использованием агрегатных функций (COUNT, SUM, AVG, MAX, MIN), производящие обобщенную групповую обработку значений полей (используя ключевые фразы GROUP BY и HAVING);
* запрос на выборку данных из двух связанных таблиц. Выбрать несколько полей, по которым сортируется вывод;
* многотабличный запрос с использованием внутреннего и внешнего соединения;
* многотабличный запрос с использованием оператора UNION.

1. Создайте SQL команды для модификации данных (INSERT, UPDATE, DELETE).

Результат запроса вывода информации, о том в каком магазине товар есть в наличии, отображён на рисунке 2.1. Код запроса: SELECT `Название товара`, `Интернет-магазин\_Код магазина` FROM товар INNER JOIN `магазин-товар ON товар.`Код товара`=`магазин-товар`.`Товар\_Код товара`.



Рисунок 2.1 – Результат запроса вывода информации, о том в каком магазине товар есть в наличии

Результат запроса вывода информации, о том какой курьер доставит тот или иной товар, отображён на рисунке 2.2. Код запроса: SELECT `Ф.И.О. курьера`, `Код товара` FROM доставка LEFT JOIN заказ ON доставка.`Код заказа` = заказ.`Код заказа`.



Рисунок 2.2 – Результат запроса вывода информации, о том какой курьер доставит тот или иной товар

Результат запроса вывода информации о названии товара, его количестве, дате и времени заказа, отображён на рисунке 2.3. Код запроса: SELECT `Название товара`, Количество, `Дата заказа`, `Время заказа` FROM товар JOIN заказ ON товар.`Код товара`=заказ.`Код товара` ORDER BY заказ.Количество AND товар.`Название товара`.

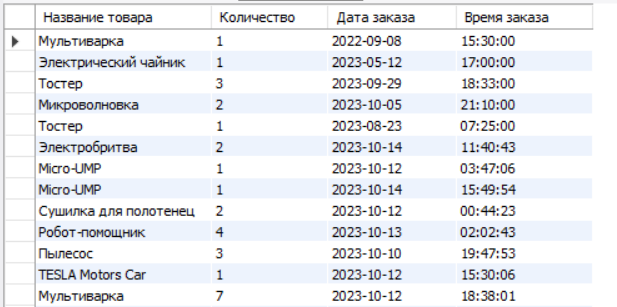


Рисунок 2.3 – Результат запроса вывода информации о названии товара, его количестве, дате и времени заказа

Результат запроса вывода информации о клиентах с их телефоном в алфавитном порядке их фамилий отображён на рисунке 2.4. Код запроса: SELECT `ФИО клиента` AS Клиент, `Контактый номер` AS Телефон FROM `фио клиента` ORDER BY `ФИО клиента`.

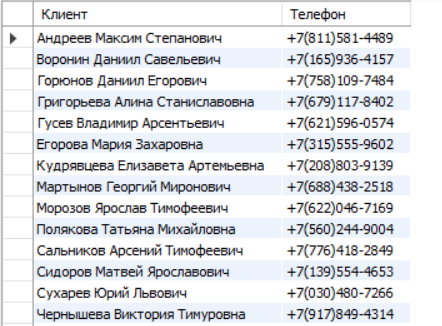


Рисунок 2.4 – Результат запроса вывода информации о клиентах с их телефоном в алфавитном порядке их фамилий

Результат запроса вывода информации о товарах и их цене стоимостью меньше 10000 рублей отображён на рисунке 2.5. Код запроса: SELECT `Название товара`, `Цена(руб.)` FROM товар WHERE `Цена(руб.)`<10000.

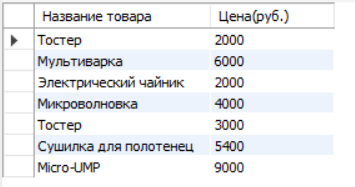


Рисунок 2.5 – Результат запроса вывода информации о товарах и их цене стоимостью меньше 10000 рублей

Результат запроса вывода всех доставщиков с фамилией Щербаков отображён на рисунке 2.6. Код запроса: SELECT `Ф.И.О. курьера` FROM доставка WHERE`Ф.И.О. курьера` LIKE 'Щербаков%'.



Рисунок 2.6 – Результат запроса вывода всех доставщиков с фамилией Щербаков

## Часть 3. Разработка интерфейса пользователя для работы с данными в БД. Создание форм.

Изучите теоретические основы разработки форм, рассмотренные в методических указаниях или других источниках.

1. создайте необходимые формы для ввода информации в базу данных (созданной на предыдущих лабораторных работах), согласно предметной области своего варианта;
2. проверьте работу форм (введите, измените и удалите около 10 записей в каждой форме);
3. проверьте правильность работы обеспечения целостности данных;
4. обдумайте и создайте формы, которые, возможно, будут полезными для будущих пользователей вашей БД.

Для разработки интерфейса пользователя для работы с базой данных использовались: html, css, php. На backend части были реализованы все http запросы, необходимые для взаимодействия с базой данных MySql. На frontend части с помощью скриптов php поступают get запросы на сервер, после чего происходит отображение данных на странице. Для добавления, обновления или удаления данных реализуются post, put и delete запросы соответственно.

На формах вывода таблиц выводятся все записи выбранной таблицы с возможностью обновления или удаления данных в них. На рисунке 3.1 представлен вывод данных таблицы, содержащую информацию о товарах.

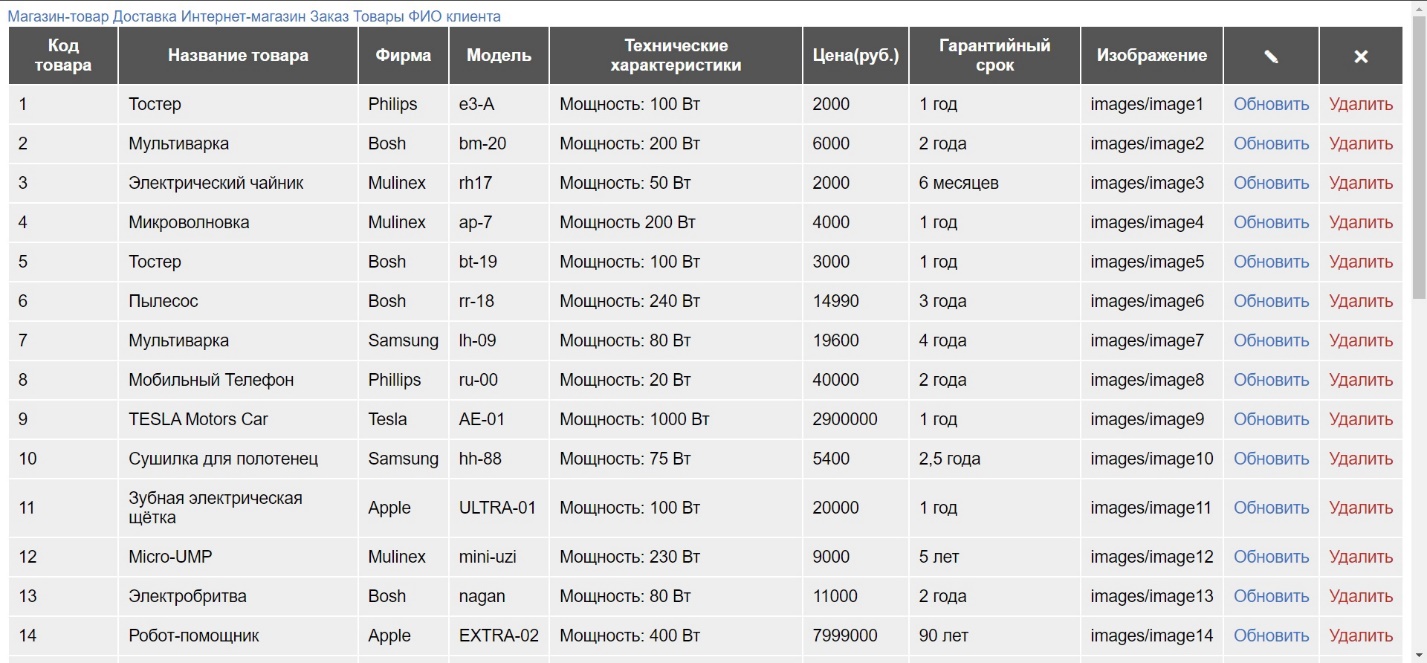


Рисунок 3.1 – Вывод всех товаров

На рисунке 3.2 представлен вывод данных таблицы, содержащую информацию о заказах.



Рисунок 3.2 – Вывод всех заказов

Также разработаны формы добавления записей в таблицы, в которых пользователь по отдельности вводит все поля. На рисунке 3.3 представлена форма добавления новой записи в таблицу, содержащую информацию о товарах. 

Рисунок 3.3 – Форма добавления товара

На рисунке 3.4 представлена форма добавления новой записи в таблицу, содержащую информацию о заказах.

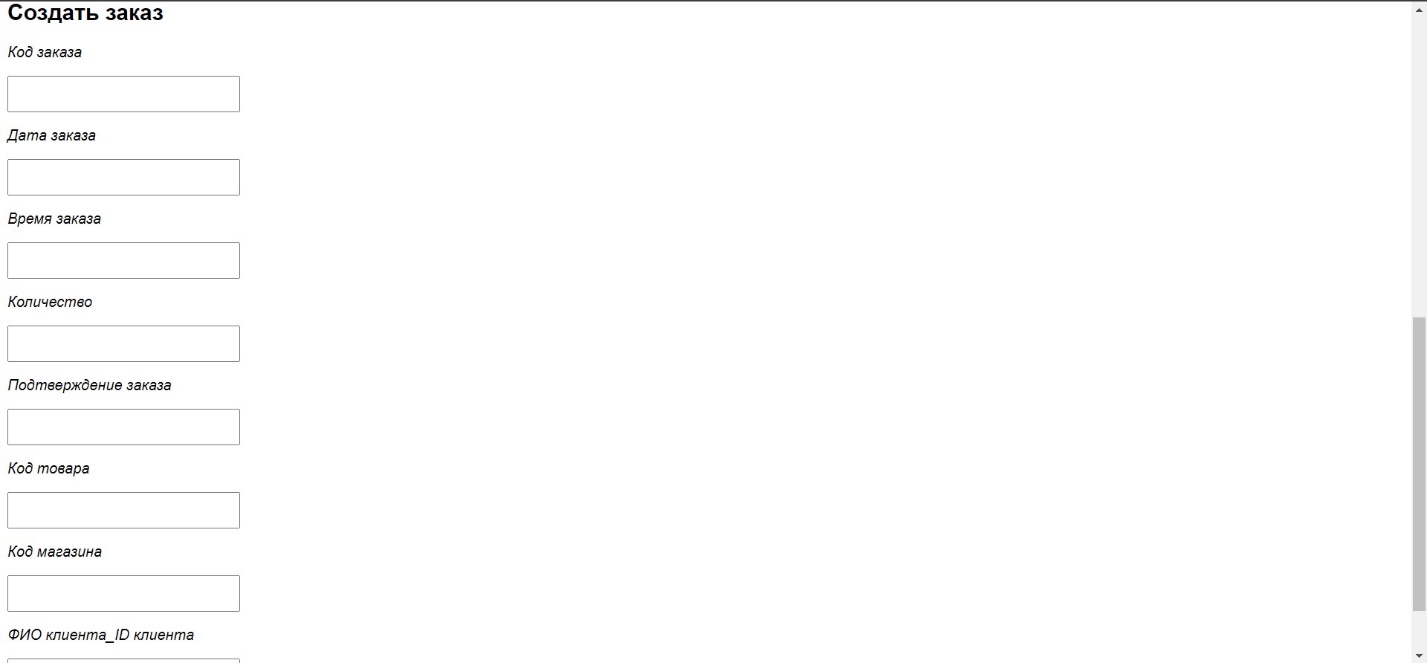


Рисунок 3.4 - Форма добавления заказа

Формы обновления записей в таблицах содержат поля записи таблицы, которую пользователь захотел обновить. Пользователь может отредактировать эти поля и таким образом обновить запись в таблице. На рисунке 3.5 представлена форма обновления товара.



Рисунок 3.5 – Форма обновления товара

На рисунке 3.6 представлена форма обновления заказа.

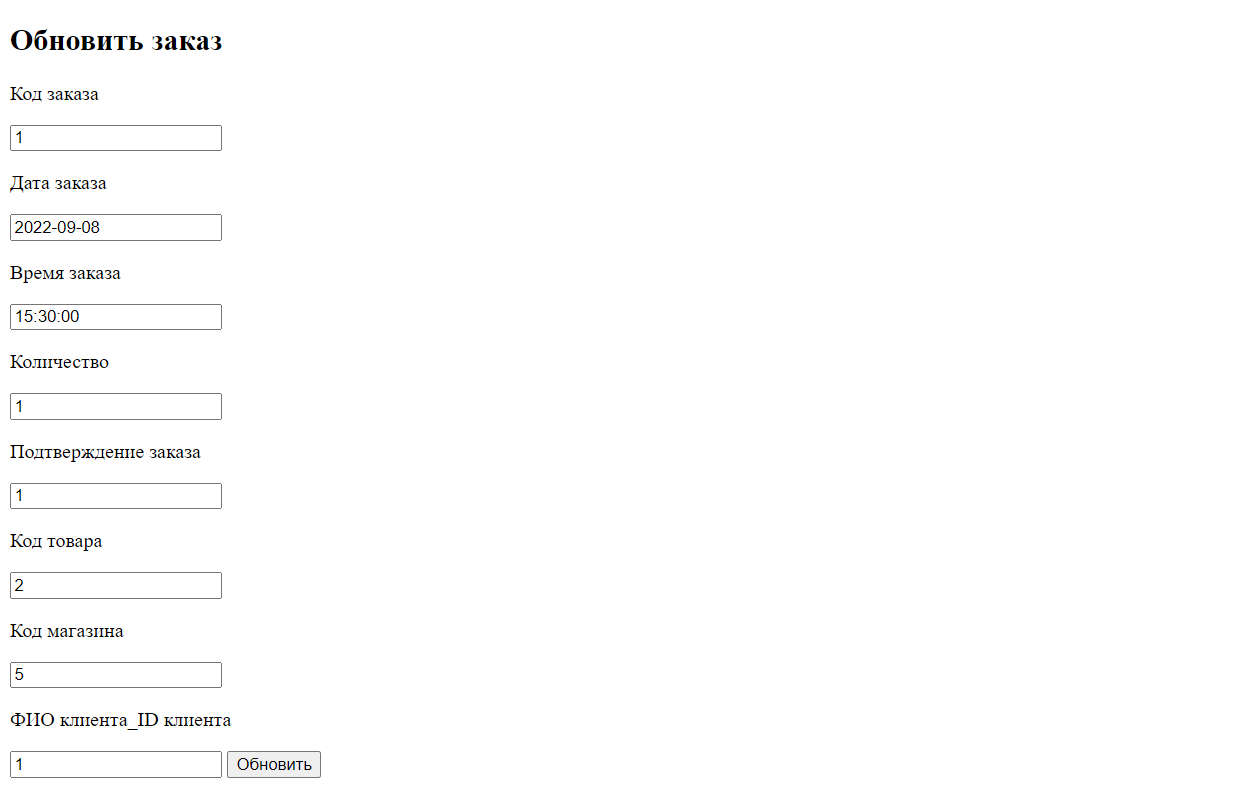


Рисунок 3.6 – Форма обновления заказа

Код страниц приведен в приложении B.

## Часть 4. Конструирование отчета

Сконструировать отчет (объект базы данных), вид которого следует выбрать из раздела "Отчеты, выводимые на основе базы данных"(находится в конце сборника заданий). В отчете произвести вычисления в строках и подвести частные и общие итоги (если они указаны).

Вид отчета:

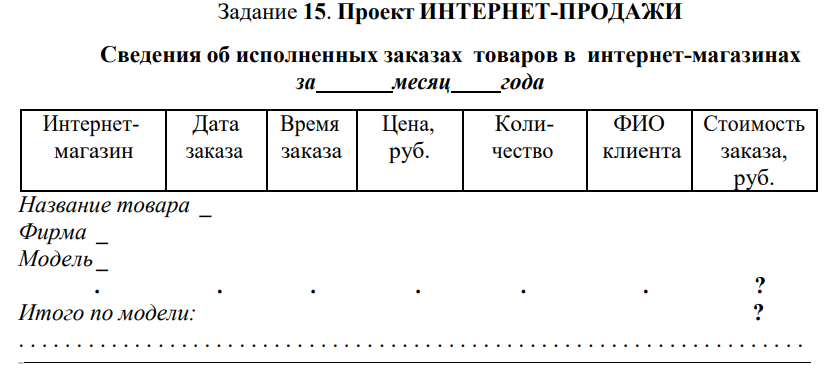


Рисунок 4.1 – Вид отчёта

На рисунке 4.2 представлен результат вывода отчета.



Рисунок 4.2 - Результат вывода отчета

На рисунке 4.2 представлен результат вывода отчета в файл pdf.

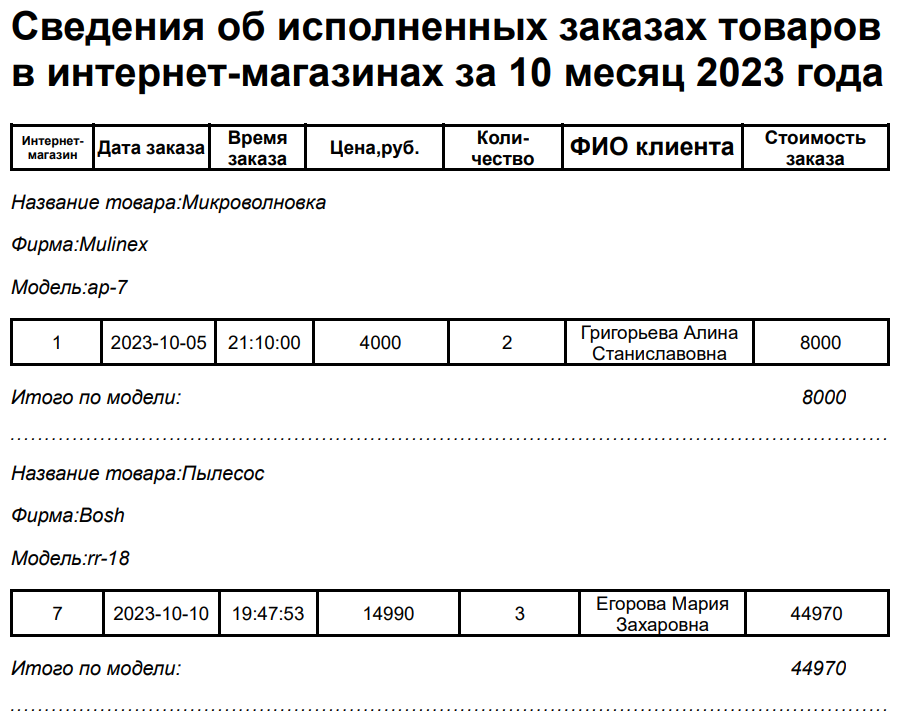


Рисунок 4.2 - Результат вывода отчета в файл pdf

Код страницы выполнения операции приведён в приложении C.

# ПРИЛОЖЕНИЕ А

CREATE SCHEMA IF NOT EXISTS `salesdb` DEFAULT CHARACTER SET utf8mb3 ;

USE `salesdb` ;

-- -----------------------------------------------------

-- Table `salesdb`.`интернет-магазин`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `salesdb`.`интернет-магазин` (

`Код магазина` INT NOT NULL,

`Электронный адрес` VARCHAR(45) NOT NULL,

`Оплата доставки` TINYINT NULL DEFAULT NULL,

PRIMARY KEY (`Код магазина`),

UNIQUE INDEX `Код магазина\_UNIQUE` (`Код магазина` ASC) VISIBLE,

UNIQUE INDEX `Электронный адрес\_UNIQUE` (`Электронный адрес` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `salesdb`.`товар`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `salesdb`.`товар` (

`Код товара` INT NOT NULL,

`Название товара` VARCHAR(45) NOT NULL,

`Фирма` VARCHAR(45) NOT NULL,

`Модель` VARCHAR(45) NULL DEFAULT NULL,

`Технические характеристики` VARCHAR(45) NOT NULL,

`Цена(руб.)` INT NOT NULL,

`Гарантийный срок` VARCHAR(45) NULL DEFAULT NULL,

`Изображение` VARCHAR(200) NULL DEFAULT NULL,

PRIMARY KEY (`Код товара`),

UNIQUE INDEX `Код товара\_UNIQUE` (`Код товара` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `salesdb`.`фио клиента`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `salesdb`.`фио клиента` (

`ID клиента` INT NOT NULL,

`Контактый номер` VARCHAR(45) NULL DEFAULT NULL,

`ФИО клиента` VARCHAR(45) NULL DEFAULT NULL,

`Адрес доставки` VARCHAR(150) NULL DEFAULT NULL,

PRIMARY KEY (`ID клиента`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `salesdb`.`заказ`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `salesdb`.`заказ` (

`Код заказа` INT NOT NULL,

`Дата заказа` DATE NOT NULL,

`Время заказа` TIME NOT NULL,

`Количество` INT NOT NULL,

`Подтверждение заказа` TINYINT NULL DEFAULT NULL,

`Код товара` INT NOT NULL,

`Код магазина` INT NOT NULL,

`ФИО клиента\_ID клиента` INT NOT NULL,

PRIMARY KEY (`Код заказа`),

UNIQUE INDEX `Код заказа\_UNIQUE` (`Код заказа` ASC) VISIBLE,

INDEX `fk\_Заказ\_Товар\_idx` (`Код товара` ASC) VISIBLE,

INDEX `fk\_Заказ\_Интернет-магазин1\_idx` (`Код магазина` ASC) VISIBLE,

INDEX `fk\_Заказ\_ФИО клиента1\_idx` (`ФИО клиента\_ID клиента` ASC) VISIBLE,

CONSTRAINT `fk\_Заказ\_Интернет-магазин1`

FOREIGN KEY (`Код магазина`)

REFERENCES `salesdb`.`интернет-магазин` (`Код магазина`),

CONSTRAINT `fk\_Заказ\_Товар`

FOREIGN KEY (`Код товара`)

REFERENCES `salesdb`.`товар` (`Код товара`),

CONSTRAINT `fk\_Заказ\_ФИО клиента1`

FOREIGN KEY (`ФИО клиента\_ID клиента`)

REFERENCES `salesdb`.`фио клиента` (`ID клиента`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `salesdb`.`доставка`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `salesdb`.`доставка` (

`Время доставки` TIME NOT NULL,

`Ф.И.О. курьера` VARCHAR(45) NOT NULL,

`Код заказа` INT NOT NULL,

PRIMARY KEY (`Код заказа`),

CONSTRAINT `fk\_Доставка\_Заказ1`

FOREIGN KEY (`Код заказа`)

REFERENCES `salesdb`.`заказ` (`Код заказа`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `salesdb`.`магазин-товар`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `salesdb`.`магазин-товар` (

`Интернет-магазин\_Код магазина` INT NOT NULL,

`Товар\_Код товара` INT NOT NULL,

PRIMARY KEY (`Интернет-магазин\_Код магазина`, `Товар\_Код товара`),

INDEX `fk\_магазин-товар\_Товар1\_idx` (`Товар\_Код товара` ASC) VISIBLE,

CONSTRAINT `fk\_магазин-товар\_Интернет-магазин1`

FOREIGN KEY (`Интернет-магазин\_Код магазина`)

REFERENCES `salesdb`.`интернет-магазин` (`Код магазина`),

CONSTRAINT `fk\_магазин-товар\_Товар1`

FOREIGN KEY (`Товар\_Код товара`)

REFERENCES `salesdb`.`товар` (`Код товара`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

INSERT INTO `salesdb`.`интернет-магазин` (`Код магазина`, `Электронный адрес`, `Оплата доставки`) VALUES ('1', 'janessa76@yahoo.com', '1');

INSERT INTO `salesdb`.`интернет-магазин` (`Код магазина`, `Электронный адрес`, `Оплата доставки`) VALUES ('2', 'harber.aubree@harris.com', '1');

INSERT INTO `salesdb`.`интернет-магазин` (`Код магазина`, `Электронный адрес`, `Оплата доставки`) VALUES ('3', 'udietrich@gmail.com', '0');

INSERT INTO `salesdb`.`интернет-магазин` (`Код магазина`, `Электронный адрес`, `Оплата доставки`) VALUES ('4', 'lilliana.okuneva@yahoo.com', '1');

INSERT INTO `salesdb`.`интернет-магазин` (`Код магазина`, `Электронный адрес`, `Оплата доставки`) VALUES ('5', 'sierra.brakus@blick.com', '0');

INSERT INTO `salesdb`.`товар` (`Код товара`, `Название товара`, `Фирма`, `Модель`, `Технические характеристики`, `Цена(руб.)`, `Гарантийный срок`) VALUES ('1', 'Тостер', 'Philips', 'e3-A', 'Мощность: 100 Вт', '2000', '1 год');

INSERT INTO `salesdb`.`товар` (`Код товара`, `Название товара`, `Фирма`, `Модель`, `Технические характеристики`, `Цена(руб.)`, `Гарантийный срок`) VALUES ('2', 'Мультиварка', 'Bosh', 'bm-20', 'Мощность: 200 Вт', '6000', '2 года');

INSERT INTO `salesdb`.`товар` (`Код товара`, `Название товара`, `Фирма`, `Модель`, `Технические характеристики`, `Цена(руб.)`, `Гарантийный срок`) VALUES ('3', 'Электрический чайник', 'Mulinex', 'rh17', 'Мощность: 50 Вт', '2000', '6 месяцев');

INSERT INTO `salesdb`.`товар` (`Код товара`, `Название товара`, `Фирма`, `Модель`, `Технические характеристики`, `Цена(руб.)`, `Гарантийный срок`) VALUES ('4', 'Микроволновка', 'Mulinex', 'ap-7', 'Мощность 200 Вт', '4000', '1 год');

INSERT INTO `salesdb`.`товар` (`Код товара`, `Название товара`, `Фирма`, `Модель`, `Технические характеристики`, `Цена(руб.)`, `Гарантийный срок`) VALUES ('5', 'Тостер', 'Bosh', 'bt-19', 'Мощность: 100 Вт', '3000', '1 год');

INSERT INTO `salesdb`.`заказ` (`Код заказа`, `Дата заказа`, `Время заказа`, `Количество`, `Ф.И.О. клиента`, `Контактный телефон`, `Подтверждение заказа`, `Код товара`, `Код магазина`) VALUES ('1', '2022-09-8', '15:30', '1', 'Лобанов Григорий Георгьевич', '+7642657403578', '1', '2', '5');

INSERT INTO `salesdb`.`заказ` (`Код заказа`, `Дата заказа`, `Время заказа`, `Количество`, `Ф.И.О. клиента`, `Контактный телефон`, `Подтверждение заказа`, `Код товара`, `Код магазина`) VALUES ('2', '2023-05-12', '17:00', '1', 'Григорьев Григорий Петрович', '+729921953168', '1', '3', '4');

INSERT INTO `salesdb`.`заказ` (`Код заказа`, `Дата заказа`, `Время заказа`, `Количество`, `Ф.И.О. клиента`, `Контактный телефон`, `Подтверждение заказа`, `Код товара`, `Код магазина`) VALUES ('3', '2023-09-29', '18:33', '3', 'Орехова Агата Ильяовна', '+73906822079', '1', '1', '1');

INSERT INTO `salesdb`.`заказ` (`Код заказа`, `Дата заказа`, `Время заказа`, `Количество`, `Ф.И.О. клиента`, `Контактный телефон`, `Подтверждение заказа`, `Код товара`, `Код магазина`) VALUES ('4', '2023-10-05', '21:10', '2', 'Петров Владлен Вадимович', '+782014804126', '0', '4', '1');

INSERT INTO `salesdb`.`заказ` (`Код заказа`, `Дата заказа`, `Время заказа`, `Количество`, `Ф.И.О. клиента`, `Контактный телефон`, `Подтверждение заказа`, `Код товара`, `Код магазина`) VALUES ('5', '2023-08-23', '7:25', '1', 'Громова Пелагея Иосифовна', '+77938478774', '1', '1', '1');

INSERT INTO `salesdb`.`доставка` (`Дата доставки`, `Время доставки`, `Адрес доставки`, `Ф.И.О. клиента`, `Ф.И.О. курьера`, `Код заказа`) VALUES ('2022-09-10', '14:00', '591856, Волгоградская область, город Домодедово, бульвар Гоголя, 75', 'Лобанов Григорий Георгьевич', 'Пахомов Алексей Кириллович', '1');

INSERT INTO `salesdb`.`доставка` (`Дата доставки`, `Время доставки`, `Адрес доставки`, `Ф.И.О. клиента`, `Ф.И.О. курьера`, `Код заказа`) VALUES ('2023-05-14', '12:00', '215112, Ярославская область, город Балашиха, ул. Будапештсткая, 29', 'Григорьев Григорий Петрович', 'Фёдоров Георгий Кириллович', '2');

INSERT INTO `salesdb`.`доставка` (`Дата доставки`, `Время доставки`, `Адрес доставки`, `Ф.И.О. клиента`, `Ф.И.О. курьера`, `Код заказа`) VALUES ('2023-10-01', '15:00', '196003, Калининградская область, город Клин, бульвар Гагарина, 50', 'Орехова Агата Ильяовна', 'Ершов Иван Геннадьевич', '3');

INSERT INTO `salesdb`.`доставка` (`Дата доставки`, `Время доставки`, `Адрес доставки`, `Ф.И.О. клиента`, `Ф.И.О. курьера`, `Код заказа`) VALUES ('2023-10-07', '17:00', '248005, Белгородская область, город Серпухов, въезд Гагарина, 72', 'Петров Владлен Вадимович', 'Лыткин Арсений Рубенович', '4');

INSERT INTO `salesdb`.`доставка` (`Дата доставки`, `Время доставки`, `Адрес доставки`, `Ф.И.О. клиента`, `Ф.И.О. курьера`, `Код заказа`) VALUES ('2023-08-26', '16:00', '943870, Ивановская область, город Люберцы, пл. Балканская, 93', 'Громова Пелагея Иосифовна', 'Зайцев Николай Геннадиевич', '5');

# ПРИЛОЖЕНИЕ B

Страница вывода информации о товарах

<?php

require\_once 'assets/config/connect.php';

$goods = mysqli\_query($connect, "SELECT \* FROM `товар`");

$goods\_dump = mysqli\_fetch\_all($goods);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="assets\css\styles.css"/>

<title>Товар</title>

</head>

<body>

<header>

<div>

<nav>

<a href="shop-good.php">Магазин-товар</a>

<a href="delivery.php">Доставка</a>

<a href="shop.php">Интернет-магазин</a>

<a href="orders.php">Заказ</a>

<a href="goods.php">Товары</a>

<a href="names.php">ФИО клиента</a>

<a href="report.php">Отчёт</a>

</nav>

</div>

</header>

<table>

<tr>

<th>Код товара</th>

<th>Название товара</th>

<th>Фирма</th>

<th>Модель</th>

<th>Технические характеристики</th>

<th>Цена(руб.)</th>

<th>Гарантийный срок</th>

<th>Изображение</th>

<th>&#9998;</th>

<th>&#10006;</th>

</tr>

<?php

foreach ($goods\_dump as $item) {

?>

<tr>

<td><?= $item[0]?></td>

<td><?= $item[1]?></td>

<td><?= $item[2]?></td>

<td><?= $item[3]?></td>

<td><?= $item[4]?></td>

<td><?= $item[5]?></td>

<td><?= $item[6]?></td>

<td><?= $item[7]?></td>

<td><a href="goods-update.php?id=<?= $item[0]?>">Обновить</a></td>

<td><a href="vendor/goods-delete.php?id=<?= $item[0]?>">Удалить</a></td>

</tr>

<?php

};

?>

</table>

<h2>Создать товар</h2>

<form action="vendor/goods-create.php" method="post">

<p>Код товара</p>

<input type="text" name="goods-code"/>

<p>Название товара</p>

<input type="text" name="goods-title"/>

<p>Фирма</p>

<input type="text" name="goods-firm"/>

<p>Модель</p>

<input type="text" name="goods-model"/>

<p>Технические характеристики</p>

<input type="text" name="goods-specifications"/>

<p>Цена(руб.)</p>

<input type="text" name="goods-price"/>

<p>Гарантийный срок</p>

<input type="text" name="goods-guarantee\_period"/>

<p>Изображение</p>

<input type="text" name="goods-image"/>

<button type="submit">Добавить</button>

</form>

</body>

</html>

# ПРИЛОЖЕНИЕ C

<?php

require\_once 'assets/config/connect.php';

$report\_month = $\_POST['report-month'];

$report\_year = $\_POST['report-year'];

$report\_month1 = $report\_month+1;

$report = mysqli\_query($connect, "SELECT `Название товара`, `Фирма`, `Модель`, `Код магазина`, `Дата заказа`, `Время заказа`, `Цена(руб.)`, `Количество`, `ФИО клиента`, `Цена(руб.)`\*`Количество`FROM заказ

JOIN товар USING(`Код товара`)

JOIN `фио клиента` ON `заказ`.`ФИО клиента\_ID клиента` = `фио клиента`.`ID клиента`

WHERE `Дата заказа` BETWEEN '$report\_year-$report\_month-01' AND '$report\_year-$report\_month1-01'

ORDER BY `Код товара`");

$report\_dump = mysqli\_fetch\_all($report);

$counts = mysqli\_query($connect, "SELECT COUNT(`Код товара`) FROM заказ

WHERE `Дата заказа` BETWEEN '$report\_year-$report\_month-01' AND '$report\_year-$report\_month1-01'

GROUP BY `Код товара`

ORDER BY `Код товара`");

$counts\_dump = mysqli\_fetch\_all($counts);

$totals = mysqli\_query($connect, "SELECT SUM(`Цена(руб.)`\*`Количество`)FROM заказ

JOIN товар USING(`Код товара`)

WHERE `Дата заказа` BETWEEN '$report\_year-$report\_month-01' AND '$report\_year-$report\_month1-01'

GROUP BY `Код товара`

ORDER BY `Код товара`");

$totals\_dump = mysqli\_fetch\_all($totals);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" media ="screen" href="assets\css\report-style.css"/>

<link rel="stylesheet" media = "print" href="assets\css\print.css">

<title>Отчёт</title>

</head>

<body>

<header>

<div>

<nav>

<a href="shop-good.php">Магазин-товар</a>

<a href="delivery.php">Доставка</a>

<a href="shop.php">Интернет-магазин</a>

<a href="orders.php">Заказ</a>

<a href="goods.php">Товары</a>

<a href="names.php">ФИО клиента</a>

<a href="report.php">Отчёт</a>

</nav>

</div>

</header>

</table>

<form action="report.php" method="post" class = "formm">

<p class = "date">Месяц</p>

<input type="number" name="report-month" min = "1" max = "12"/>

<p class = "date">Год</p>

<input type="text" name="report-year"/>

<button type="submit">Сконструировать отчёт</button>

<a href="#" onclick="window.print()" class = "prbut">Распечатать</a>

</form>

<h1>Сведения об исполненных заказах товаров в интернет-магазинах за <?=$report\_month?> месяц <?=$report\_year?> года</h1>

<table>

<tr>

<th class = "shop">Интернет-магазин</th>

<th class = "order-date">Дата заказа</th>

<th class="time">Время заказа</th>

<th class="price">Цена,руб.</th>

<th class = "kolvo">Коли-<br>

чество</th>

<th class= "fio">ФИО клиента</th>

<th>Стоимость заказа</th>

</tr>

</table>

<?php

$n = 0;

for ($i=0; $i < count($report\_dump); $i++) {

?>

<p>

Название товара:<?=$report\_dump[$i][0]?>

</p>

<p>

Фирма:<?=$report\_dump[$i][1]?>

</p>

<p>

Модель:<?=$report\_dump[$i][2]?>

</p>

<table>

<?php

for($j=0; $j < $counts\_dump[$n][0]; $j++){

?>

<tr>

<td class = "shop"><?=$report\_dump[$i][3]?></td>

<td class = "order-date"><?=$report\_dump[$i][4]?></td>

<td class="time"><?=$report\_dump[$i][5]?></td>

<td class="price"><?=$report\_dump[$i][6]?></td>

<td class = "kolvo"><?=$report\_dump[$i][7]?></td>

<td class= "fio"><?=$report\_dump[$i][8]?></td>

<td><?=$report\_dump[$i][9]?></td>

</tr>

<?php

$i++;};

?>

</table>

<div class = "itogdiv">

<p>Итого по модели:</p>

<p class = "total"><?=$totals\_dump[$n][0]?></p>

</div>

<p class = "dots"><b>. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

</b></p>

<?php

$i+=-1;

$n++;

};

?>

</body>

</html>