МИНОБРНАУКИ РФ

ФГБОУ ВПО Тверской государственный технический университет

Кафедра “Программное обеспечение”

Курсовая работа

дисциплина “Базы данных”

Тема: “ Тема: “ Разработка системы автоматизации с использованием СУБД: online – магазин ”

Выполнил:

студент группы

ПИН 17.05

Завгороднев Егор

Проверил:

Артемов И.Ю

Тверь 2018

Оглавление

[Постановка задачи 3](#_Toc7257267)

[Теоретический раздел 3](#_Toc7257268)

[Предметная область - Онлайн магазин 3](#_Toc7257269)

[Реляционная БД 3](#_Toc7257270)

[Выбранная СУБД, основания выбора СУБД 3](#_Toc7257271)

[Практический раздел 3](#_Toc7257272)

[Схема БД 3](#_Toc7257273)

[Функционал 3](#_Toc7257274)

[Алгоритмы программы 3](#_Toc7257275)

[Выводы 3](#_Toc7257276)

[Краткое резюме 3](#_Toc7257277)

[Преимущества 3](#_Toc7257278)

[Недостатки 3](#_Toc7257279)

[Список литературы 3](#_Toc7257280)

# 

# 

# Постановка задачи

Разработка системы автоматизации с использованием СУБД (MySQL/Firebird/PosgreSQL/MSSQL): online - магазин.

В результате выполнения курсовой работы должно быть разработано программное обеспечение  (в виде исходных кодов) и оформлена пояснительная записка (20-25 стр.) (в печатном и электронном виде).

Исходный код программного обеспечения должен:

* Разрабатывать на языке SQL;
* Содержать полную (корректную) обработку ошибок;
* Выполнять все поставленные задачи

# Теоретический раздел

## Предметная область - Онлайн магазин

Online магазин осуществляет продажу различных категорий товаров. Товары могут быть различных брендов и производителей. У магазина есть список поставщиков, каждый из которых поставляет определённую категорию товаров. Также  магазин хранит список покупателей и номера их банковских карт.

В базе данных должны храниться:

1. Каталог товаров (описание товара, характеристики товара, размещение их по категориям). По каждому товару в БД должны храниться следующие характеристики:

* Название товара
* Описание товара
* Цена
* ID Категории
* ID Бренда
* ID Производителя

2. Список поставщиков (описание поставщиков). По каждому поставщику в БД должны храниться следующие характеристики:

* Наименование поставщика
* Email
* Телефон
* Категория поставляемых товаров

3. Список покупателей. По каждому покупателю в БД хранятся следующие характеристики:

* Логин
* Пароль
* Email
* Фамилия
* Имя
* Отчество

4. Кредитная карта. По каждой кредитной карте в БД хранятся следующие характеристики:

* Номер кредитной карты
* Имя владельца
* Проверочный код
* Срок истечения
* ID покупателя

Для осуществления процесса купли-продажи товара и отслеживания статуса заказа в БД должна храниться информация о заказах.

5. Список заказов. По каждому заказу в БД хранятся следующие характеристики:

* № заказа
* Дата и время заказа
* Стоимость заказа
* ID покупателя
* Товары в заказе

6. Категория товара

* Название категории

7. Бренд товара

* Название бренда

8. Производитель товара

* Название производителя

## Реляционная БД

База данных **–** совокупность данных о некоторых объектах предметной области.

Система управления базами данных - это совокупность языковых и программных средств, которая осуществляет доступ к данным, позволяет их создавать, менять и удалять, обеспечивает безопасность данных и т.д.

Базы данных

* являются средством для рационального и эффективного хранения информации
* обеспечивают надежную защиту информации от потери или порчи.
* экономно использует ресурсы и снабжены механизмами поиска информации, удовлетворяющими к разумным требованиям производительности.

История возникновения и развития технологий баз данных

**4000 г. до н. э.** - средства учёта царской казны и налогов в древнем Шумере, первое упоминание о хранении и обработке данных

**1955 год** - программируемое оборудование обработки записей. Программное обеспечение этого времени поддерживало модель обработки записей на основе файлов. Для хранения данных использовались перфокарты

**Начало 1960- х** – появление термина *база данных*

**Середина 1960 –x** - оперативные сетевые базы данных, в сообществе баз данных Кобол была проработана концепция схем баз данных и концепция независимости данных.

**Начало 1970-х** – реляционная модель данных

Виды баз данных

Классификация по модели данных:

* Иерархическая
* Объектная и объектно-ориентированная
* Объектно-реляционная
* Реляционная
* Сетевая
* Функциональная.

В данной курсовой работе я использую реляционную модель данных.

Реляционная база данных – это набор данных с предопределенными связями между ними. Эти данные организованны в виде набора таблиц, состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных.

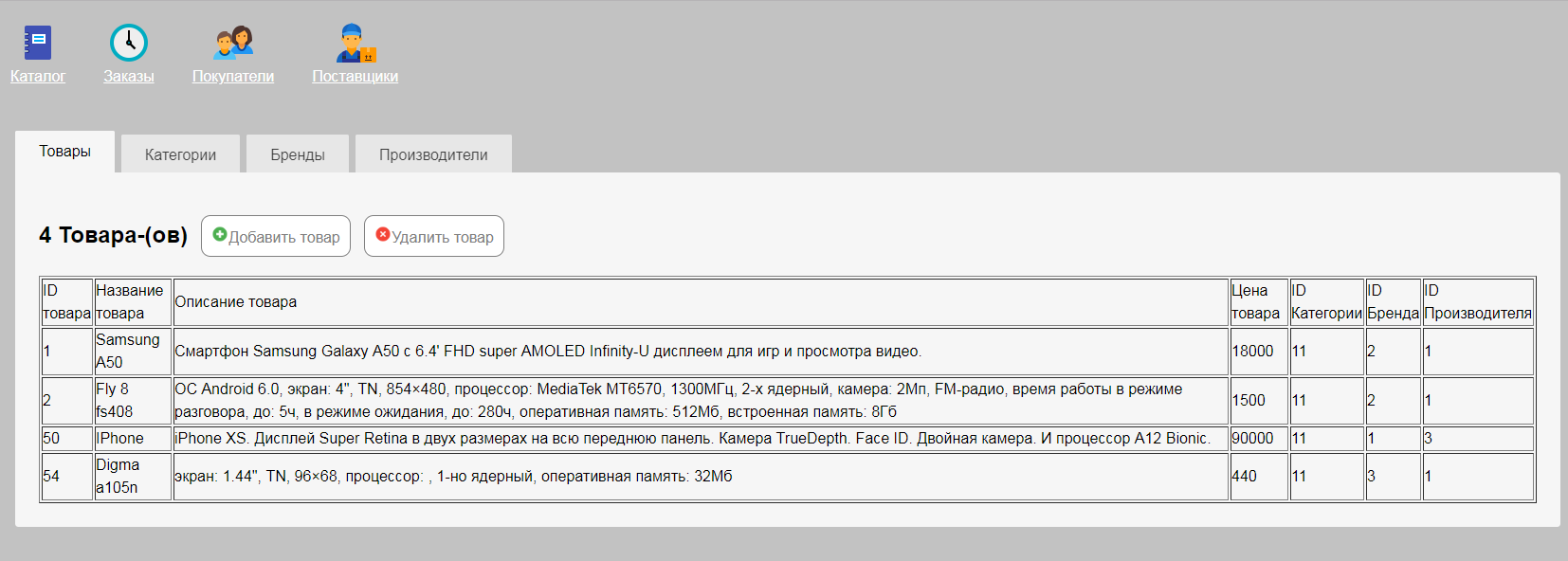
В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке – значение атрибута. Каждая стока таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешних ключей.

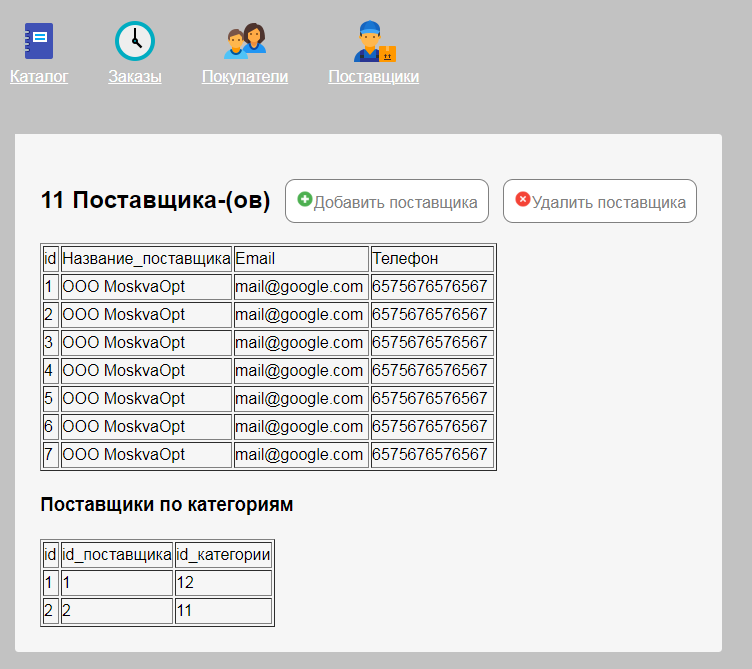
## Выбранная СУБД, основания выбора СУБД

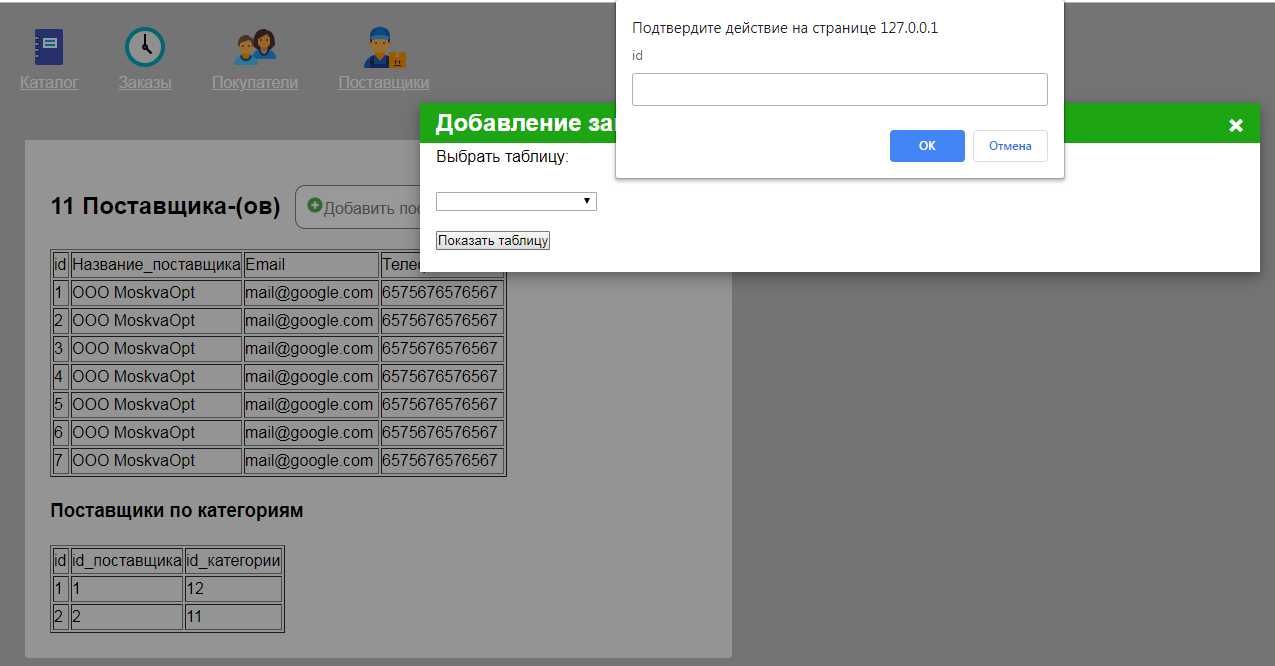
Так как система автоматизации разрабатывалась на языке PHP, наиболее удобным вариантом для меня стала СУБД MySQL. Веб интерфейс PhpMyAdmin позволяет легко взаимодействовать PHP приложениям с MySQL.

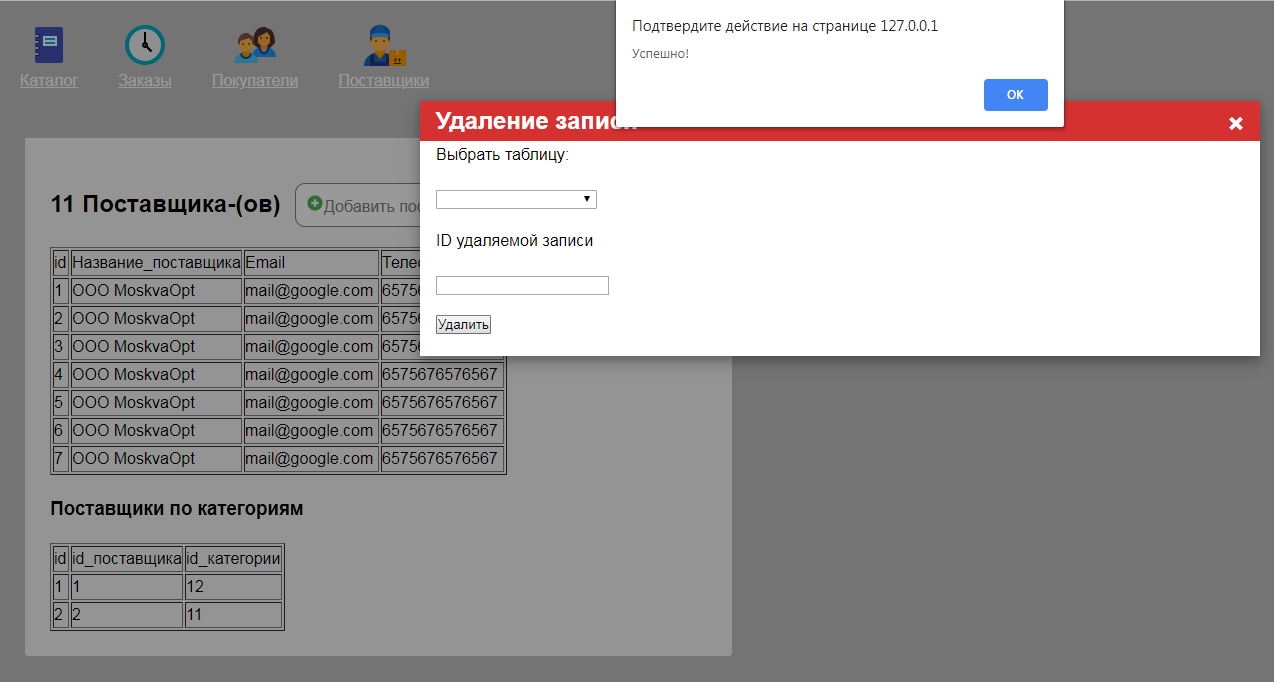
# Практический раздел

## Интерфейс приложения









При проектировании и разработке любой базы данных интерфейс играет важную роль. Он представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными компонентами базы данных.

Пользователь попадает в главное меню, которое предназначено для предоставления пользователю удобной навигации по всем формам базы данных.

Главное меню содержит три раздела:

* Каталог
* Заказы
* Покупатели
* Поставщики

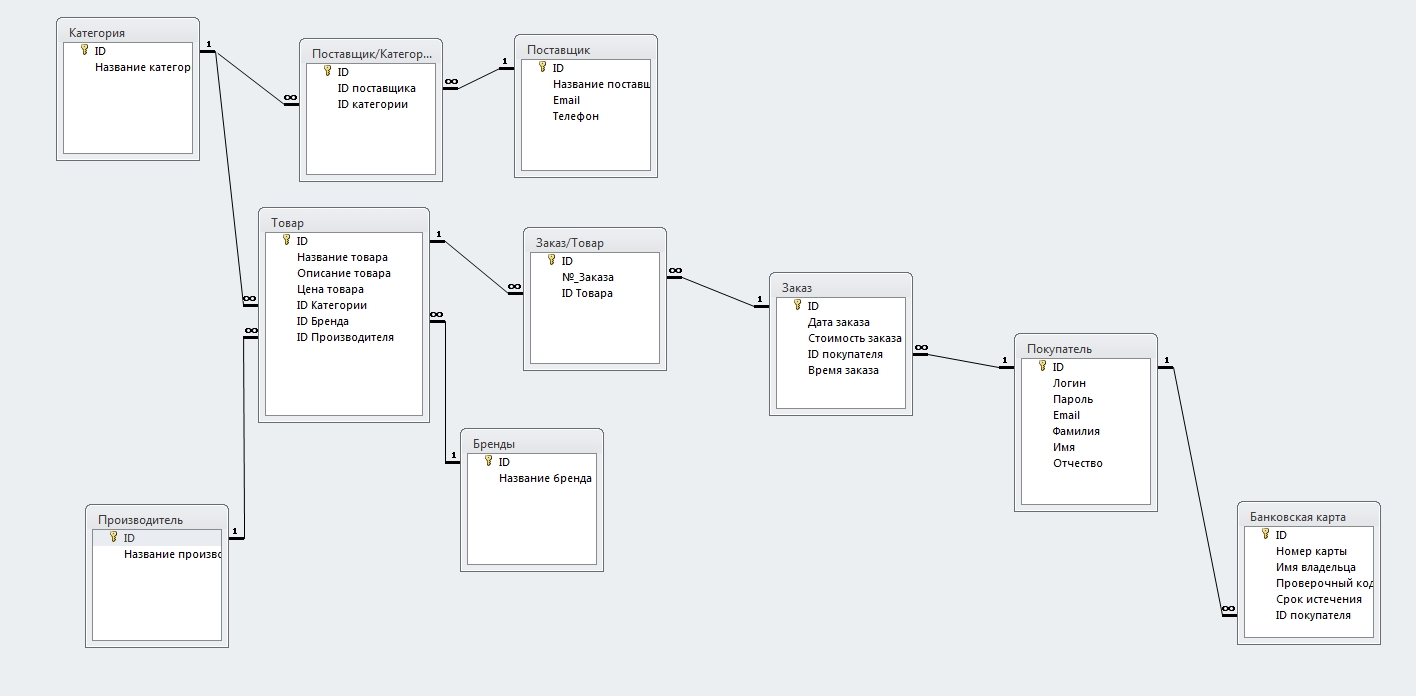
У каталога есть четыре подменю:

* Товары
* Категории
* Бренды
* Производители

Также есть два модальных окна:

* Для добавления записей
* Для удаления записей

## Схема БД



ID таблиц – первичные ключи.

Внешние ключи:

* ID поставщика
* ID категории
* ID бренда
* ID производителя
* ID товара
* ID покупателя

Поля таблиц:

* Банковские карты

|  |  |
| --- | --- |
| `id` | int(11) |
| `Номер\_карты` | bigint(11) |
| `Имя\_владельца` | tinytext |
| `Проверочный\_код` | smallint(4) |
| `Срок\_истечения` | date |
| `Id\_покупателя` | int(11) |

* Категория

|  |  |
| --- | --- |
| `id` | int(11) |
| `Название\_категории` | tinytext |

* Покупатель

|  |  |
| --- | --- |
| `id` | int(11) |
| `Логин` | tinytext |
| `Пароль` | tinytext |
| `EMail` | tinytext |
| `Фамилия` | tinytext |
| `Имя` | tinytext |
| `Отчество` | tinytext |

* Поставщик

|  |  |
| --- | --- |
| `id` | int(11) |
| `Название\_поставщика` | tinytext |
| `Email` | tinytext |
| `Телефон` | bigint(20) |

* Поставщик/Категория

|  |  |
| --- | --- |
| `id` | int(11) |
| `id\_поставщика` | int(11) |
| `id\_категории` | int(11) |

* Производитель

|  |  |
| --- | --- |
| `id` | int(11) |
| `Название\_производителя` | tinytext |

* Бренды

|  |  |
| --- | --- |
| `id` | int(11) |
| `Название\_бренда` | tinytext |

* Заказ

|  |  |
| --- | --- |
| `id` | int(11) |
| `Дата\_заказа` | date |
| `Стоимость\_заказа` | int(11) |
| `ID\_покупателя` | int(11) |
| `Время\_заказа` | time |

* Заказ/Товар

|  |  |
| --- | --- |
| `id` | int(11) |
| `ID\_Заказа` | int(11) |
| `ID\_Товара` | int(11) |

* Товар

|  |  |
| --- | --- |
| `id` | int(11) |
| `Название\_товара` | tinytext |
| `Описание\_товара` | text |
| `Цена\_товара` | double |
| `ID\_Категории` | int(11) |
| `ID\_Бренда` | int(11) |
| `ID\_Производителя` | int(11) |

## Функционал

Приложение обладает следующим функционалом:

* Вывод таблиц из БД
* Добавление в БД
* Удаление из БД (по идентификатору)

## Алгоритмы программы

Клиентская часть программы написана на ЯП Javascript, за стили отвечают HTML и CSS.

Она взаимодействует с базой данных с помощью ЯП PHP. Запросы формируются на языке SQL. Используемая СУБД – MySQL.

Вывод таблиц из БД осуществляется с помощью SQL запросов:

* Формируется запрос к таблице. Например: SELECT \* FROM Товар;
* Циклом выводятся данные из нее.
* Результат выводится в виде HTML таблицы.

Удаление из БД

* В модальном окне пользователь выбирает название и id записи удаляемой таблицы
* К БД отправляется запрос вида: DELETE FROM “Название таблицы” WHERE id= “ID удаляемой записи”.
* Данная запись удаляется из БД и HTML таблицы.

Добавление в БД

* В модальном окне пользователь выбирает название записи таблицы для добавления.
* К серверу отправляется запрос, который отправляет заголовки полей данной таблицы.
* Формируются модальные окна с заголовком поля и полем ввода.
* При правильном заполнении данных, на сервер отправляется запрос вида

INSERT INTO “Название таблицы” (“Заголовки полей через запятую”) VALUES (“Значения полей”).

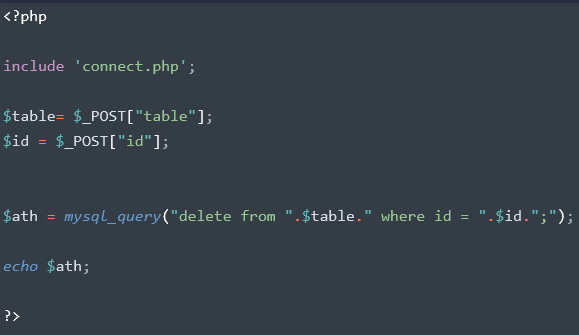
* Запись добавляется в БД и HTML таблицу.

Примеры кодов:

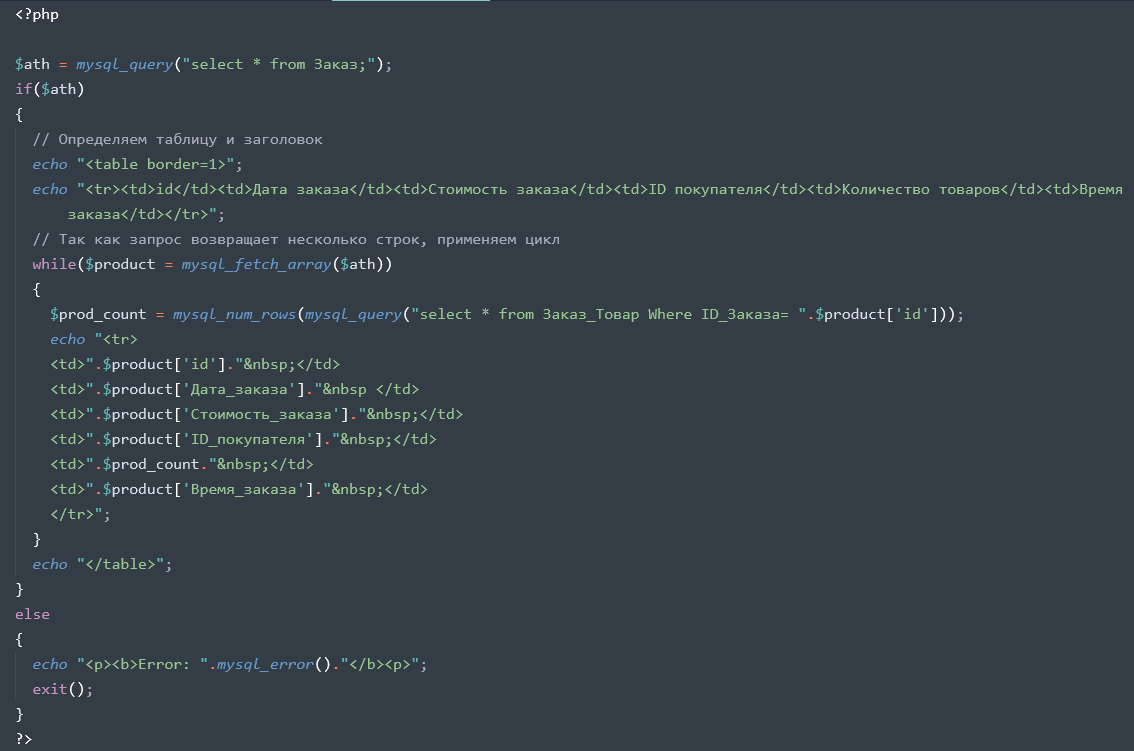
* Отправка запроса на добавление записи в БД



* Отправка запроса на удаление записи в БД



* Вывод HTML таблицы для таблицы БД “Записи”



# Выводы

## Краткое резюме

Программа имеет небольшой функционал и выполняет базовые задачи для данной предметной области.

## Преимущества

* Программа работает(!)
* Она выполняет базовые задачи для данной предметной области.
* Содержит простую обработку ошибок

## Недостатки

* Минимальный функционал
* Не очень удобна для пользователя

# Список литературы

1. [https://www.php.net](https://www.php.net/) – справочник для языка программирования PHP
2. [http://www.softtime.ru](http://www.softtime.ru/) - Взаимодействие PHP и MySQL