

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Информационных систем**

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Управление данными»**  
**Тема: «Разработка базы данных для автоматизации складского учёта»**

Студентка гр. 0361

\_\_\_\_\_

Солонухина А.Л.

Преподаватель

\_\_\_\_\_

Татарникова Т.М.

Санкт-Петербург

2022

## ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студентка Солонухина А.Л.

Группа 0361

Тема работы: разработка базы данных для автоматизации складского учёта.

Исходные данные:

*Вариант 18.*

Спроектировать базу данных, построить программу, обеспечивающую взаимодействие с ней в режиме диалога, для автоматизации складского учета.

В базе данных должны содержаться следующие данные:

- информация об единицах хранения, номер ордера, дата, код поставщика, балансный счет, код сопроводительного документа по справочнику документов, номер сопроводительного документа, код материала по справочнику материалов, счет материала, код единицы измерения, количество пришедшего материала, цена единицы измерения);
- информация о хранящихся на складе материалах (справочник материалов, код класса материала, код группы материала, наименование материала);
- информация об единицах измерения конкретных видов материалов, код материала, единица измерения (метры, килограммы, литры и т.д.);
- информация о поставщиках материалов, код поставщика, его наименование, ИНН, юридический адрес (индекс, город, улица, дом), адрес банка (индекс, город, улица, дом), номер банковского счета.

Помимо SQL запросов для создания таблиц базы данных, разработать пакет, состоящий из процедур и функций, позволяющий:

- посчитать количество поставщиков данного материала;
- предоставить возможность добавления единицы хранения с указанием всех реквизитов;
- вывести список поставщиков с указанием всех реквизитов данного материала на склад;
- для указанного адреса банка посчитать количество поставщиков склада, пользующихся услугами этого банка.

Содержание пояснительной записки: введение, анализ предметной области, обоснование модели данных, обоснование выбора СУБД, описание функций групп пользователей, описание функций управления данными, организация защиты БД, заключение, список использованных источников, приложение А – руководство пользователя БД, приложение Б – листинг программного кода.

Предполагаемый объем пояснительной записки:

Не менее 80 страниц.

Дата выдачи задания: 01.09.2022

Дата сдачи реферата: 21.12.2022

Дата защиты реферата: 23.12.2022

Студент \_\_\_\_\_

Солонухина А.Л.

Преподаватель \_\_\_\_\_

Татарникова Т.М.

## СОДЕРЖАНИЕ

	Введение	5
1.	Анализ предметной области	7
1.1.	Описание объектов предметной области	7
1.2.	Формулировка задач, решаемых БД	8
1.3.	Краткое описание алгоритмов решения задач	8
1.4.	Определение групп пользователей БД	9
1.5.	Описание выходных документов, генерируемых системой	10
1.6.	Описание входных документов для заполнения данными БД	10
2.	Обоснование модели данных	11
3.	Обоснование выбора СУБД	12
4.	Описание функций групп пользователей	13
4.1.	Группы пользователей	13
4.2.	Назначение прав доступа	13
5.	Описание функций управления данными	14
5.1.	Хранение	14
5.2.	Манипулирование	16
5.3.	Доступ к данным	18
5.4.	Предоставление запрашиваемых данных пользователю	19
6.	Организация защиты БД	22
6.1.	Описание ограничений целостности для каждого информационного объекта	22
6.2.	Рекомендуемые средства физической защиты	24
6.3.	Описание процедуры подтверждения подлинности	24
	Заключение	25
	Список использованных источников	26
	Приложение А. Руководство пользователя БД	27
	Приложение Б. Листинг программного кода	41

## **ВВЕДЕНИЕ**

Автоматизация складского учёта является необходимым мероприятием для повышения эффективности работы торговых и производящих предприятий, в которых так или иначе происходит хранение большого объема материалов. Это, в свою очередь, позволяет увеличить доход предприятий, так как уменьшает количество ошибок при ведении учёта и экономит время, которое потребовалось бы на выполнение задач учета вручную, с использованием бумажных носителей информации или построчно заполняемых текстовых журналов.

В области автоматизации складского учёта важное место занимает упорядоченное хранение большого количества данных – это данные о хранимых материалах, договорах об их поставке, поставщиках и прочие данные, косвенно относящиеся к хранимым материалам.

Так как склад сам по себе является структурированной системой, необходимо автоматизировать создание и сортировку записей и ордеров по материалам, ведение справочников поставщиков и документов. Для материалов важно чтобы работник склада мог быстро определить его количество в наличии, а также местоположение, основываясь на координатах в каталоге. Для изменяющихся во времени данных нужна возможность изменения информации о них в системе без необходимости создания новых записей.

Также при условии большого количества данных складского учёта необходима автоматизация получения конкретных выборок информации, относящейся к хранимым материалам, в соответствии с запросами пользователей (работников склада).

Таким образом, автоматизация складского учёта в виде создания базы данных с возможностями хранения, управления и получения ответов по заданным критериям необходима и полезна.

Для создания базы данных предстоит решить следующие задачи:

- 1) проанализировать предметную область и обрабатываемые в ней данные;
- 2) построить и обосновать модель данных;
- 3) выбрать СУБД;
- 4) разработать несколько групп пользователей и определить их права доступа к БД;
- 5) описать функции управления БД;
- 6) разработать и описать защиту данных в БД;
- 7) построить приложение для взаимодействия пользователей с БД.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Описание объектов предметной области

В соответствии с заданием и особенностями предметной области выберем объекты и их характеристики, которые впоследствии будут использоваться при построении базы данных:

### *1) Поставщик*

- Идентификатор поставщика
- Наименование поставщика
- ИНН поставщика
- Юридический адрес поставщика
- Адрес банка
- Номер банковского счёта

### *2) Сопровождающий документ*

- Идентификатор документа
- Номер документа
- Дата заключения документа
- Наименование документа
- Поставщик, с которым заключён документ
- Материал, о поставке которого заключён документ

### *3) Единица измерения*

- Идентификатор единицы измерения
- Наименование единицы измерения

### *4) Материал*

- Идентификатор материала
- Идентификатор класса материала
- Идентификатор группы материала
- Наименование материала

- Единица измерения материала

## **1.2. Формулировка задач, решаемых БД**

База данных, предназначенная для автоматизации складского учёта, должна автоматизировать следующие задачи:

- 1) создание и редактирование справочника поставщиков, которые поставляют складу материалы;
- 2) создание и редактирование справочника документов, сопровождающих поставку материалов;
- 3) создание и редактирование справочника единиц измерения, в которых измеряются хранимые материалы;
- 4) создание и редактирование справочника материалов с указанием их координат (класс/группа) в каталоге склада;
- 5) создание и редактирование ордеров об единицах хранения (коды из вышеперечисленных справочников, описывающие хранящиеся на складе материалы);
- 6) осуществление подсчёта числа поставщиков, поставляющих на склад указанный материал (из справочника);
- 7) осуществление подсчёта числа поставщиков, пользующихся услугами указанного банка (по адресу банка);
- 8) получение информации по указанным материалам, поставляемым на склад.

## **1.3. Краткое описание алгоритмов решения задач**

Для решения перечисленных в предыдущем пункте задач необходимо чтобы база данных была структурирована следующим образом: каждой таблице соответствует отдельный объект (пункт 1.1.) и для удобства сбора



информации по материалам нужен производный объект – *ордер по единицам хранения* со следующими характеристиками:

- Идентификатор ордера
- Дата
- Идентификатор поставщика
- Балансный счёт
- Идентификатор сопровождающего документа
- Номер сопровождающего документа
- Идентификатор материала
- Счёт материала
- Идентификатор единицы измерения
- Количество пришедшего материал
- Цена единицы измерения

Для создания и редактирования справочников и ордеров необходимо разработать набор пользовательских форм, создающих при помощи SQL-запросов соответствующие объекты.

Для получения информации по критериям, указанным в задачах, необходимо составить пакет соответствующих SQL-запросов, учитывая структуру перечисленных таблиц.

#### **1.4. Определение групп пользователей БД**

С учётом особенностей организации процесса складского учёта можно выделить две группы пользователей:

*Директор.* Эта роль предназначена для сотрудника склада, предположительно являющегося его владельцем или другим должностным лицом с полным доступом к ведению складского учёта.

**Наблюдатель.** Эта роль предназначена для клиентов склада (предприятия, которому принадлежит склад) – людей, не имеющих доступ к ведению складского учёта, но имеющих право просмотра и поиска хранимой там информации.

### 1.5. Описание выходных документов, генерируемых системой

В соответствии с заданием на построение БД генерирование выходных документов (справок и отчётов) не предусмотрено.

Выходными данными в данном случае являются результаты поиска по отдельным критериям, указанным в пункте 1.2.

### 1.6. Описание входных документов для заполнения данными БД

В качестве входных данных в создаваемой БД используются:

- информация об *единицах хранения* (номер ордера, дата, код поставщика, балансный счет, код сопроводительного документа, номер сопроводительного документа, код материала, счет материала, код единицы измерения, количество пришедшего материала, цена единицы измерения);
- информация о *сопроводительных документах* (код документа, номер документа, дата, наименование документа, поставщик, поставляемый материал);
- информация о хранящихся на складе *материалах* (код материала, код класса материала, код группы материала, наименование материала, единица измерения материала);
- информация о *единицах измерения* (код единицы измерения, единица измерения);
- информация о *поставщиках* материалов (код поставщика, его наименование, ИНН, юридический адрес (индекс, город, улица, дом), адрес банка (индекс, город, улица, дом), номер банковского счета).

## 2. ОБОСНОВАНИЕ МОДЕЛИ ДАННЫХ

Так как складской учёт представляет собой изначально область структурирования информации (то есть упорядоченность – основная его цель), принято решение о выборе реляционной модели БД.

В соответствии с перечисленными в пункте 1 объектами (*поставщик, сопроводительный документ, единица измерения, материал*) и производными объектами (*единица хранения*) построена схема БД организации складского учёта (рисунок 1).

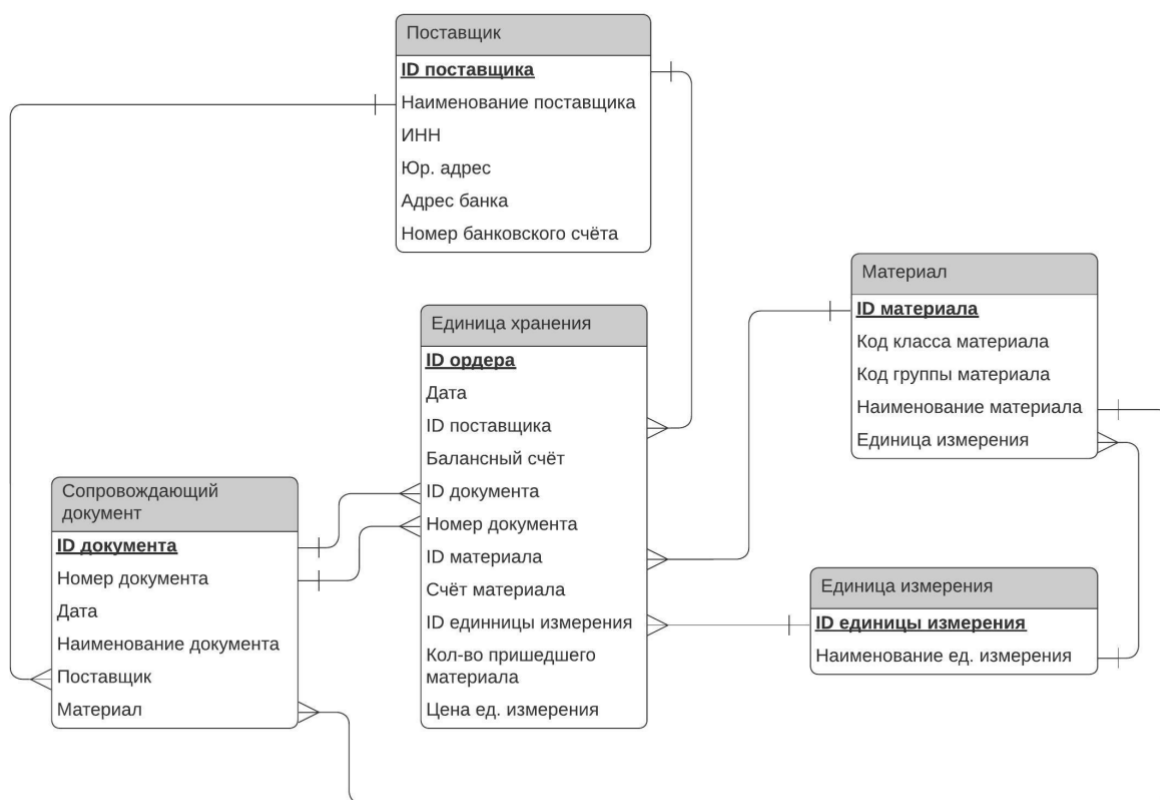


Рисунок 1 – Схема БД для автоматизации складского учёта

### 3. ОБОСНОВАНИЕ ВЫБОРА СУБД

В качестве СУБД выбрана MySQL, так как она:

- универсальна для использования в веб-приложениях (программа для взаимодействия пользователей с БД построена в виде веб-приложения);
- свободно распространяемая;
- простая в использовании через приложение Терминал;
- имеет удобные вспомогательные приложения (использовалось приложение MySQL Workbench);
- изначально имеет встроенные инструменты безопасности данных.

## 4. ОПИСАНИЕ ФУНКЦИЙ ГРУПП ПОЛЬЗОВАТЕЛЕЙ

### 4.1. Группы пользователей

Для использования БД определены две группы пользователей:

**Директор** – создаёт и редактирует справочники и ордера по единицам хранения, осуществляет поиск, предусмотренный указанным ранее функционалом.

**Наблюдатель** – может просматривать справочники и ордера по единицам хранения, осуществляет поиск, предусмотренный указанным ранее функционалом.

### 4.2. Назначение прав доступа

Права доступа для групп пользователей Директор и Наблюдатель представлены в таблице 1, где: S – SELECT (чтение данных), I – INSERT (добавление данных), U – UPDATE (модификация данных), D – DELETE (удаление данных).

Таблица 1 – Права доступа групп пользователей БД

Объект	Директор	Наблюдатель
Поставщик	SIUD	S
Сопроводительный документ	SIUD	S
Единица измерения	SID	S
Материал	SID	S
Единица хранения	SID	S

## 5. ОПИСАНИЕ ФУНКЦИЙ УПРАВЛЕНИЯ ДАННЫМИ

### 5.1. Хранение

Данные базы данных «sklad» хранятся в пяти таблицах: Vendor, Document, Measure, Material и Material\_Order. Таблицы создаются при помощи указанных далее команд (SQL-запросов):

```
CREATE TABLE `Vendor` (  
    `VendorID` int NOT NULL AUTO_INCREMENT,  
    `VendorName` varchar(50) NOT NULL,  
    `VendorINN` varchar(12) NOT NULL,  
    `VendorAdress` varchar(100) NOT NULL,  
    `VendorBankAdress` varchar(100) NOT NULL,  
    `VendorBankNumber` varchar(16) NOT NULL,  
    PRIMARY KEY (`VendorID`),  
    UNIQUE KEY `VendorName` (`VendorName`)  
)
```

```
CREATE TABLE `Measure` (  
    `MeasureID` int NOT NULL AUTO_INCREMENT,  
    `MeasureName` varchar(50) NOT NULL,  
    PRIMARY KEY (`MeasureID`),  
    UNIQUE KEY `MeasureName` (`MeasureName`)  
)
```

```
CREATE TABLE `Material` (  
    `MaterialID` int NOT NULL AUTO_INCREMENT,  
    `MaterialKlass` int NOT NULL,  
    `MaterialGroup` int NOT NULL,  
    `MaterialName` varchar(50) NOT NULL,  
    `MaterialMeasure` varchar(50) NOT NULL,  
    PRIMARY KEY (`MaterialID`),  
    UNIQUE KEY `MaterialName` (`MaterialName`),
```

```

        KEY `MeasureName_idx` (`MaterialMeasure`),
        CONSTRAINT `MeasureName` FOREIGN KEY (`MaterialMeasure`)
REFERENCES `Measure` (`MeasureName`) ON DELETE CASCADE ON UPDATE
CASCADE
    )

```

```

CREATE TABLE `Document` (
    `DocumentID` int NOT NULL AUTO_INCREMENT,
    `DocumentNum` int NOT NULL,
    `DocumentDate` varchar(50) NOT NULL,
    `DocumentName` varchar(50) NOT NULL,
    `DocumentVendor` varchar(50) NOT NULL,
    `DocumentMaterial` varchar(50) DEFAULT NULL,
    PRIMARY KEY (`DocumentID`),
    KEY `DocumentNum` (`DocumentNum`),
    KEY `VendorName_idx` (`DocumentVendor`),
    KEY `MaterialName_idx` (`DocumentMaterial`),
    CONSTRAINT `MaterialName` FOREIGN KEY (`DocumentMaterial`)
REFERENCES `Material` (`MaterialName`) ON DELETE CASCADE ON UPDATE
CASCADE,
    CONSTRAINT `VendorName` FOREIGN KEY (`DocumentVendor`)
REFERENCES `Vendor` (`VendorName`) ON DELETE CASCADE ON UPDATE
CASCADE
)

```

```

CREATE TABLE `Material_Order` (
    `OrderID` int NOT NULL AUTO_INCREMENT,
    `OrderDate` varchar(50) NOT NULL,
    `OrderVendorID` int NOT NULL,
    `OrderNum` varchar(10) NOT NULL,
    `OrderDocumentID` int NOT NULL,
    `OrderDocumentNum` int NOT NULL,
    `OrderMaterialID` int NOT NULL,
    `OrderMaterialNum` int NOT NULL,
    `OrderMeasureID` int NOT NULL,

```

```

        `OrderMaterialSum` int NOT NULL,
        `OrderMeasurePrice` int NOT NULL,
        PRIMARY KEY (`OrderID`),
        KEY `VendorID_idx` (`OrderVendorID`),
        KEY `DocumentID_idx` (`OrderDocumentID`),
        KEY `DocumentNum_idx` (`OrderDocumentNum`),
        KEY `MaterialID_idx` (`OrderMaterialID`),
        KEY `MeasureID_idx` (`OrderMeasureID`),
        CONSTRAINT `DocumentID` FOREIGN KEY (`OrderDocumentID`)
REFERENCES `Document` (`DocumentID`) ON DELETE CASCADE ON UPDATE
CASCADE,
        CONSTRAINT `DocumentNum` FOREIGN KEY (`OrderDocumentNum`)
REFERENCES `Document` (`DocumentNum`) ON DELETE CASCADE ON UPDATE
CASCADE,
        CONSTRAINT `MaterialID` FOREIGN KEY (`OrderMaterialID`)
REFERENCES `Material` (`MaterialID`) ON DELETE CASCADE ON UPDATE
CASCADE,
        CONSTRAINT `MeasureID` FOREIGN KEY (`OrderMeasureID`)
REFERENCES `Measure` (`MeasureID`) ON DELETE CASCADE ON UPDATE
CASCADE,
        CONSTRAINT `VendorID` FOREIGN KEY (`OrderVendorID`)
REFERENCES `Vendor` (`VendorID`) ON DELETE CASCADE ON UPDATE
CASCADE
    )

```

## 5.2. Манипулирование

Запрос INSERT INTO Vendor (VendorName, VendorINN, VendorAdress, VendorBankAdress, VendorBankNumber) VALUES (name, inn, adress, bank\_adress, bank\_number) используется для создания записи о поставщике.

Для редактирования записи о поставщике используются запросы: SELECT \* FROM Vendor WHERE VendorID=id и UPDATE Vendor SET



VendorName=name, VendorINN=inn, VendorAdress=adress, VendorBankAdress=bank\_adress, VendorBankNumber=bank\_number WHERE VendorID=id, где id – идентификатор редактируемого поставщика.

Удаление записи о поставщике происходит при помощи запроса DELETE FROM Vendor WHERE VendorID=id, где id – идентификатор удаляемого поставщика.

Запрос INSERT INTO Document (DocumentNum, DocumentDate, DocumentName, DocumentVendor) VALUES (num, date, name, vendor) используется для создания записи о документе.

Добавление в информации о документе с идентификатором id поставляемого материала из справочника материалов происходит при помощи запроса UPDATE Document SET DocumentMaterial=material WHERE DocumentID=id.

Удаление записи о документе происходит при помощи запроса DELETE FROM Document WHERE DocumentID=id, где id – идентификатор удаляемого документа.

Запрос INSERT INTO Measure (MeasureName) VALUES (name) используется для создания записи об единице измерения.

Удаление записи о документе происходит при помощи запроса DELETE FROM Measure WHERE MeasureID=id, где id – идентификатор удаляемой единицы измерения.

Запрос INSERT INTO Material (MaterialKlass, MaterialGroup, MaterialName, MaterialMeasure) VALUES (klass, group, name, measure) используется для создания записи о материале.

Удаление записи о материале происходит при помощи запроса DELETE FROM Material WHERE MaterialID=id, где id – идентификатор удаляемого материала.

Запрос INSERT INTO Material\_Order (OrderDate, OrderVendorID, OrderNum, OrderDocumentID, OrderDocumentNum, OrderMaterialID, OrderMaterialNum, OrderMeasureID, OrderMaterialSum, OrderMeasurePrice) VALUES (date, document.split(' - ')[4], num, document.split(' - ')[0], document.split(' - ')[1], document.split(' - ')[6], material\_num, document.split(' - ')[8], material\_sum, measure\_price] используется для создания записи об единице хранения. Элементы в нём берутся из передаваемого массива с разделителями « - », а material\_num = measure\_price \* material\_sum.

Удаление записи об единице хранения происходит при помощи запроса DELETE FROM Material\_Order WHERE OrderID=id, где id – идентификатор удаляемой единицы хранения.

### 5.3. Доступ к данным

Пользователю Director (Директор) были выданы наборы прав доступа к таблицам БД следующими запросами:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON sklad.Vendor  
TO 'Director'@'localhost';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON  
sklad.Document TO 'Director'@'localhost';
```

```
GRANT SELECT, INSERT, DELETE ON sklad.Measure TO  
'Director'@'localhost';
```

```
GRANT SELECT, INSERT, DELETE ON sklad.Material TO  
'Director'@'localhost';
```

```
GRANT SELECT, INSERT, DELETE ON sklad.Material_Order  
TO 'Director'@'localhost';
```

Итоговая таблица, выводимая по запросу прав доступа этого пользователя представлена на рисунке 2.

```
[mysql> SHOW GRANTS FOR 'Director'@'localhost';  
+-----+  
| Grants for Director@localhost |  
+-----+  
| GRANT USAGE ON *.* TO `Director`@`localhost` |  
| GRANT SELECT, INSERT, UPDATE, DELETE ON `sklad`.`document` TO `Director`@`localhost` |  
| GRANT SELECT, INSERT, DELETE ON `sklad`.`material_order` TO `Director`@`localhost` |  
| GRANT SELECT, INSERT, DELETE ON `sklad`.`material` TO `Director`@`localhost` |  
| GRANT SELECT, INSERT, DELETE ON `sklad`.`measure` TO `Director`@`localhost` |  
| GRANT SELECT, INSERT, UPDATE, DELETE ON `sklad`.`vendor` TO `Director`@`localhost` |  
+-----+
```

Рисунок 2 – Права доступа Директора

Пользователю Viewer (Наблюдатель) были выданы наборы прав доступа к таблицам БД следующим запросом:

```
GRANT SELECT ON `sklad`.* TO 'Viewer'@'localhost';
```

Итоговая таблица, выводимая по запросу прав доступа этого пользователя представлена на рисунке 3.

```
[mysql> SHOW GRANTS FOR 'Viewer'@'localhost';  
+-----+  
| Grants for Viewer@localhost |  
+-----+  
| GRANT USAGE ON *.* TO `Viewer`@`localhost` |  
| GRANT SELECT ON `sklad`.* TO `Viewer`@`localhost` |  
+-----+
```

Рисунок 3 – Права доступа Наблюдателя

## 5.4. Предоставление запрашиваемых данных пользователю

При помощи запроса `SELECT * FROM Vendor` страница получает данные о Поставщиках для их отображения, `SELECT * FROM Document` – о Документах, `SELECT * FROM Measure` – об Единицах измерения, `SELECT * FROM Material ORDER BY MaterialKlass, MaterialGroup` – о Материалах, сортируя их по возрастанию сперва кода класса, затем кода группы для их отображения.

При помощи запроса `SELECT * FROM Material_Order` страница получает данные об Единицах хранения для их отображения.

Запрос `SELECT Vendor.VendorID, Vendor.VendorName, Document.DocumentID, Document.DocumentNum, Document.DocumentName, Document.DocumentVendor, Document.DocumentMaterial, Material.MaterialID, Material.MaterialMeasure, Measure.MeasureID, Measure.MeasureName FROM Vendor, Document, Material, Measure WHERE Vendor.VendorName=Document.DocumentVendor AND Material.MaterialName=Document.DocumentMaterial AND Measure.MeasureName=Material.MaterialMeasure` используется для получения данных по предыдущим объектам, которые используются пользователем для создания записи об единице хранения.

Список уникальных поставляемых материалов для выбора перед поиском предоставляется пользователю запросом `SELECT DISTINCT DocumentMaterial FROM Document;`

Количество поставщиков, поставляющих выбранный товар, находится с использованием запроса `SELECT COUNT(DISTINCT DocumentVendor) FROM Document WHERE DocumentMaterial=material;`

Список уникальных адресов банков, которыми пользуются поставщики склада предоставляется пользователю запросом `SELECT DISTINCT VendorBankAdress FROM Vendor;`

Количество поставщиков, пользующихся услугами выбранного банка, находится с использованием запроса `SELECT COUNT(DISTINCT VendorName) FROM Vendor WHERE VendorBankAdress=bank;`

Список поставленных материалов предоставляется пользователю запросом `SELECT DISTINCT DocumentMaterial FROM Document;`

Список всех поставщиков выбранного материала (с указанием всех его реквизитов) предоставляется пользователю с помощью запроса `SELECT DISTINCT Material.MaterialName, Material.MaterialKlass, Material.MaterialGroup, Material.MaterialMeasure, Document.DocumentVendor FROM Material, Document WHERE Document.DocumentMaterial=material and Material.MaterialName=Document.DocumentMaterial;`

## 6. ОРГАНИЗАЦИЯ ЗАЩИТЫ БД

### 6.1. Описание ограничений целостности для каждого информационного объекта

Ограничения целостности для объектов представлены на рисунках 4, 5, 6, 7 и 8.

Для каждой характеристики объекта в таблицах представлены описания следующих параметров:

varchar (число) – **максимальная длина строки**;

Null NO/ YES – характеристика **не может/может быть пустой**;

Key PRI – **ключ объекта**;

Key UNI – характеристика должна быть **уникальной**;

Default NULL – изначально характеристика имеет **пустое значение**;

EXTRA auto\_increment – при создании нового объекта к каждому последующему идентификатору **автоматически прибавляется единица**.

```
[mysql> describe Vendor;
```

Field	Type	Null	Key	Default	Extra
VendorID	int	NO	PRI	NULL	auto_increment
VendorName	varchar(50)	NO	UNI	NULL	
VendorINN	varchar(12)	NO		NULL	
VendorAdress	varchar(100)	NO		NULL	
VendorBankAdress	varchar(100)	NO		NULL	
VendorBankNumber	varchar(16)	NO		NULL	

Рисунок 4 – Ограничения целостности для объекта «Поставщик»

```
[mysql> describe Document;
```

Field	Type	Null	Key	Default	Extra
DocumentID	int	NO	PRI	NULL	auto_increment
DocumentNum	int	NO	MUL	NULL	
DocumentDate	varchar(50)	NO		NULL	
DocumentName	varchar(50)	NO		NULL	
DocumentVendor	varchar(50)	NO	MUL	NULL	
DocumentMaterial	varchar(50)	YES	MUL	NULL	

Рисунок 5 – Ограничения целостности для объекта «Сопровождающий документ»

```
mysql> describe Measure;
```

Field	Type	Null	Key	Default	Extra
MeasureID	int	NO	PRI	NULL	auto_increment
MeasureName	varchar(50)	NO	UNI	NULL	

Рисунок 6 – Ограничения целостности для объекта «Единица измерения»

```
[mysql> describe Material;
```

Field	Type	Null	Key	Default	Extra
MaterialID	int	NO	PRI	NULL	auto_increment
MaterialKlass	int	NO		NULL	
MaterialGroup	int	NO		NULL	
MaterialName	varchar(50)	NO	UNI	NULL	
MaterialMeasure	varchar(50)	NO	MUL	NULL	

Рисунок 7 – Ограничения целостности для объекта «Материал»

```
[mysql> describe Material_Order;
```

Field	Type	Null	Key	Default	Extra
OrderID	int	NO	PRI	NULL	auto_increment
OrderDate	varchar(50)	NO		NULL	
OrderVendotID	int	NO	MUL	NULL	
OrderNum	varchar(10)	NO		NULL	
OrderDocumentID	int	NO	MUL	NULL	
OrderDocumentNum	int	NO	MUL	NULL	
OrderMaterialID	int	NO	MUL	NULL	
OrderMaterialNum	int	NO		NULL	
OrderMeasureID	int	NO	MUL	NULL	
OrderMaterialSum	int	NO		NULL	
OrderMeasurePrice	int	NO		NULL	

Рисунок 8 – Ограничения целостности для объекта «Единица хранения»

Другие

## 6.2. Рекомендуемые средства физической защиты

Для поддержания целостности и сохранности данных рекомендуется регулярное *резервное копирование структуры и данных БД*. Рекомендуемая частота – каждые 24 часа. Резервное копирование можно проводить с помощью инструмента *Data Export* приложения MySQL Workbench.

Эта мера поможет защитить БД от:

- сбоев в носителях информации;
- сбоев оборудования;
- ошибок пользователей.

## 6.3. Описание процедуры подтверждения подлинности

Подтверждение личности для доступа к БД происходит по логину (username) и паролю пользователя автоматически при запуске соответствующей (именной) сборки пользовательского приложения.



## **ЗАКЛЮЧЕНИЕ**

По итогам выполнения курсовой работы была спроектирована и построена база данных для автоматизации складского учёта – технологии, широко применяемой в сфере производства и торговли.

Для взаимодействия пользователей с БД написано пользовательское веб-приложение с использованием Node.js, Express и HTML. С учётом формата приложения была выбрана СУБД MySQL, достоинства которой были описаны в работе.

Задачи, поставленные в задании на курсовую работу, были выполнены, а используемые SQL-запросы задокументированы в отчёте.

По итогам тестирования взаимодействия пользователей двух существующих групп (Директор и Наблюдатель) с БД с помощью веб-приложения было выявлено соответствие заявленным требованиям к объектам БД и её структуре.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Курс Управление данными. ЛЭТИ. URL: [https://open.eltech.ru/courses/course-v1:kafedra-is+CS1+fall\\_2022/course/](https://open.eltech.ru/courses/course-v1:kafedra-is+CS1+fall_2022/course/) (дата обращения – 12.10.2022);
2. Руководство по Node.js. MySQL. URL: <https://metanit.com/web/nodejs/> (дата обращения – 12.10.2022);
3. MySQL Workbench. Руководство. URL: <https://simple-scada.com/help/manual/mysql-workbench.html> (дата обращения – 18.11.2022);

## ПРИЛОЖЕНИЕ А. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Меню веб-приложения представлено страницей с набором ссылок-заголовков (рисунок 9).

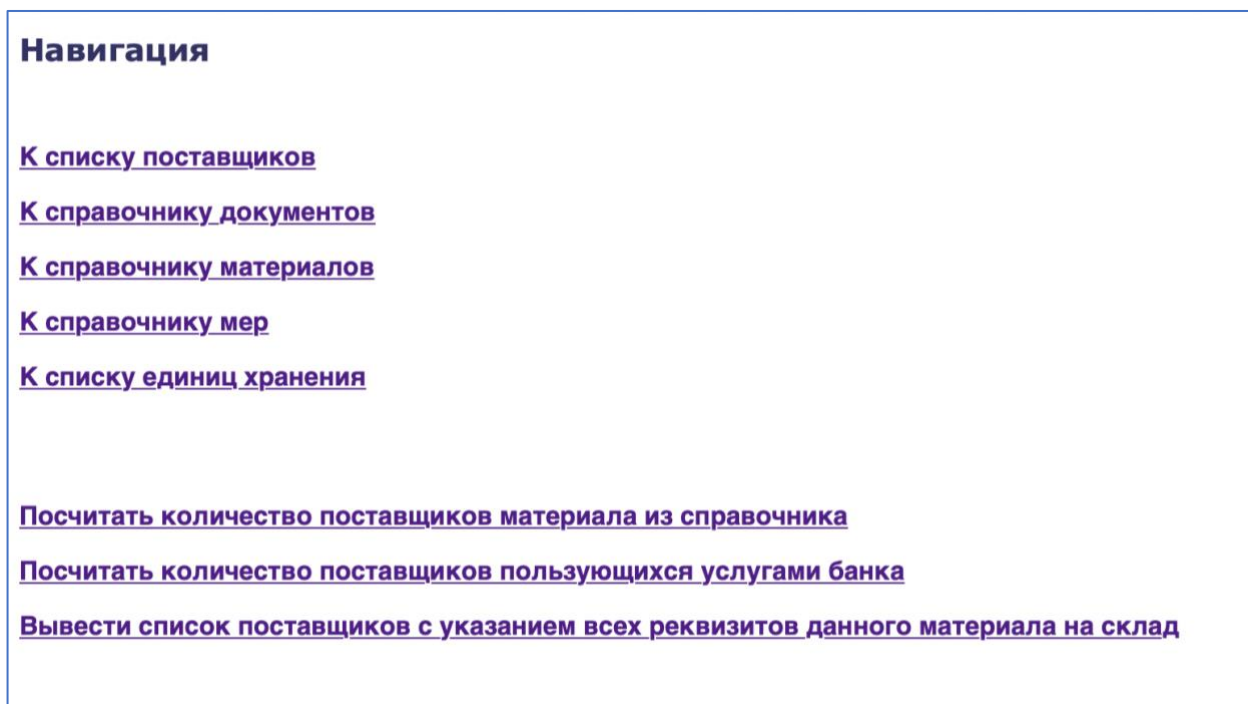


Рисунок 9 – Меню навигации

На рисунке 9 видно, что пользователю предлагается выбор пяти страниц справочников и трёх пунктов по составлению выборок и подсчётов.

При выборе пункта «К списку поставщиков» нажатием на него открывается страница справочника поставщиков склада. Пример представлен на рисунке 10.

На странице находится таблица с поставщиками и их характеристиками, над таблицей расположена ссылка на страницу добавления нового поставщика в справочник, под таблицей ссылка на возврат к меню навигации. В таблице помимо данных справа в крайних столбцах расположены кнопка удаления поставщика и ссылка на форму редактирования поставщика.

Список поставщиков

[Добавить поставщика](#)

Код поставщика	Наименование	ИНН	Юридический адрес	Адрес банка	Номер счёта		
1	ИП "Верёвочкин"	1204505434	124256 г. Саратов, ул. Мира, д. 25	912758 г. Саратов, ул. Нечаева, д. 10	6235075071652087	<a href="#">Edit</a>	<a href="#">Delete</a>
2	ООО "Бетон"	967584376234	235523 г. Псков, ул. Юнarmии, д. 12	706859 г. Саратов, ул. Нечаева, д. 10	8897086985745865	<a href="#">Edit</a>	<a href="#">Delete</a>
3	ООО "Вышиванка"	695000069956	235645 г. Псков, ул. Цветочная, д. 27	123099 г. Саратов, ул. Нечаева, д. 10	0982530713057057	<a href="#">Edit</a>	<a href="#">Delete</a>
4	ИП "Пчёлка"	2194869142	123000 г. Москва, ул. Сатиров, д. 11	567893 г. Северодвинск, ул. Славы Бестужева, д. 18	2476624961924898	<a href="#">Edit</a>	<a href="#">Delete</a>
5	ОАО "Металл"	165245768977	213551 г. Тобольск, ул. Титова, д. 32	512523 г. Тамбов, ул. Волковская, д. 29	6547435613461613	<a href="#">Edit</a>	<a href="#">Delete</a>

[В главное меню](#)

Рисунок 10 – Список поставщиков

При выборе пункта «Добавить поставщика» открывается форма (рисунок 11) где требуется ввести информацию о поставщике в текстовые формы.

Добавление поставщика

Наименование

ИНН

Юридический дрес

Почтовый индекс

Город

Улица

Дом

Адрес банка

Почтовый индекс

Город

Улица

Дом

Номер счёта

Отправить

[К списку поставщиков](#)

Рисунок 11 – Добавление поставщика

На рисунке 12 представлена ситуация, когда одно или несколько полей не заполнены – при нажатии кнопки «Отправить» появляется соответствующее уведомление и запись не создаётся, дожидаясь заполнения всех обязательных полей. Эта особенность предусмотрена во избежание ошибок при заполнении элементов БД со свойством NOT NULL.

**Добавление поставщика**

Заполните это поле

Наименование

ИНН

815767556916

Юридический адрес

Почтовый индекс

Город

Улица

Дом

Рисунок 12 – Незаполненное поле

В каждом поле предусмотрено пороговое допустимое количество символов для ввода (в соответствии со свойствами БД), далее которого запись символов не идёт.

После правильного заполнения всех полей формы и нажатия кнопки «Отправить» новая запись добавляется в справочник поставщиков. Примеры заполнения формы и добавления записи представлены на рисунках 13 и 14.

### Добавление поставщика

Наименование

ИНН

Юридический адрес

Почтовый индекс  
 Город

Улица  
 Дом

Адрес банка

Почтовый индекс  
 Город

Улица  
 Дом

Номер счёта

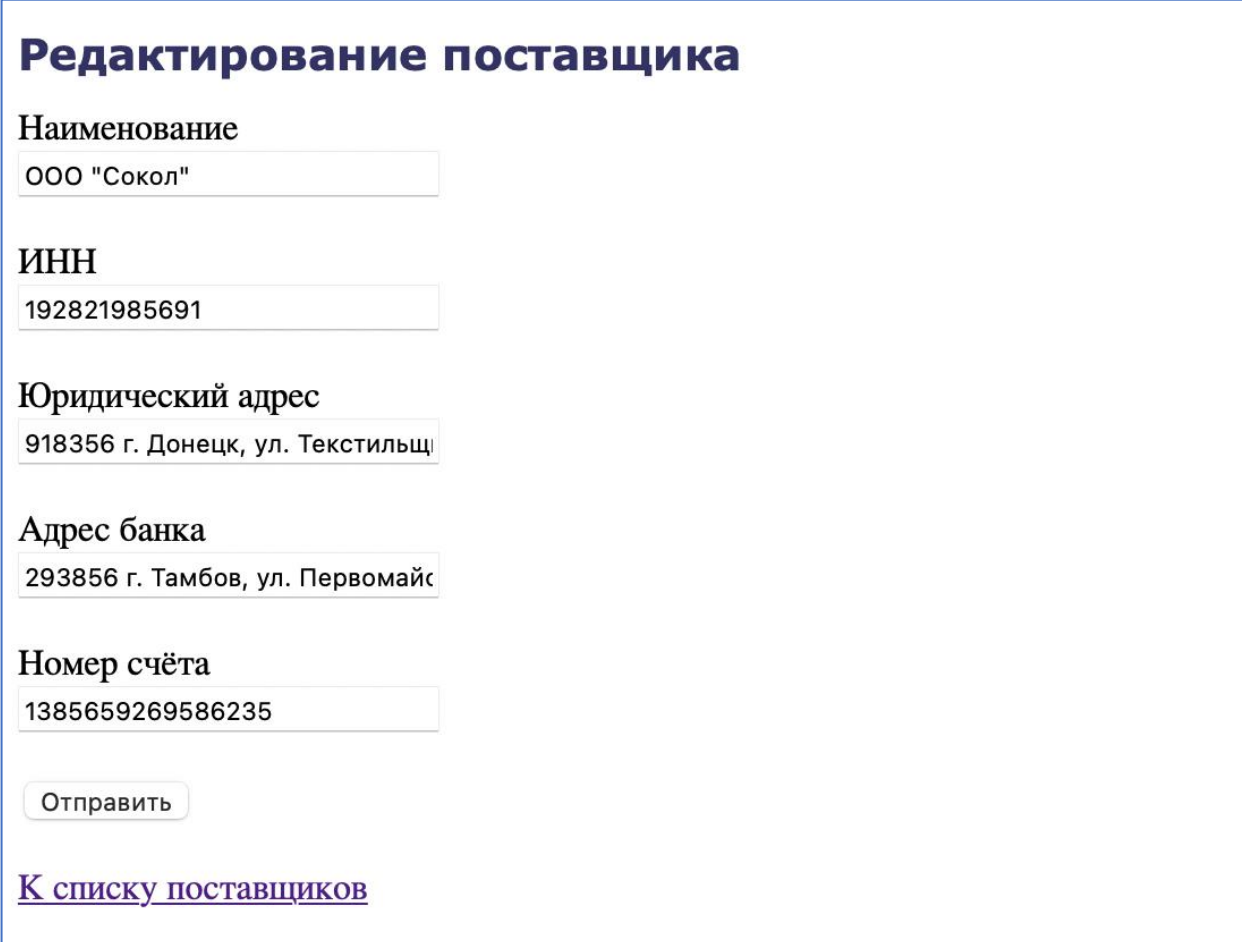
[К списку поставщиков](#)

Рисунок 13 – Пример заполнения формы

Список поставщиков							
<a href="#">Добавить поставщика</a>							
Код поставщика	Наименование	ИНН	Юридический адрес	Адрес банка	Номер счёта		
1	ИП "Верёвочкин"	1204505434	124256 г. Саратов, ул. Мира, д. 25	912758 г. Саратов, ул. Нечаева, д. 10	6235075071652087	<a href="#">Edit</a>	<a href="#">Delete</a>
2	ООО "Бетон"	967584376234	235523 г. Псков, ул. Юнarmии, д. 12	706859 г. Саратов, ул. Нечаева, д. 10	8897086985745865	<a href="#">Edit</a>	<a href="#">Delete</a>
3	ООО "Вышиванка"	695000069956	235645 г. Псков, ул. Цветочная, д. 27	123099 г. Саратов, ул. Нечаева, д. 10	0982530713057057	<a href="#">Edit</a>	<a href="#">Delete</a>
4	ИП "Пчёлка"	2194869142	123000 г. Москва, ул. Сатиров, д. 11	567893 г. Северодвинск, ул. Славы Бестужева, д. 18	2476624961924898	<a href="#">Edit</a>	<a href="#">Delete</a>
5	ОАО "Металл"	165245768977	213551 г. Тобольск, ул. Титова, д. 32	512523 г. Тамбов, ул. Волковская, д. 29	6547435613461613	<a href="#">Edit</a>	<a href="#">Delete</a>
8	ООО "Сокол"	192821985691	918356 г. Донецк, ул. Текстильщиков, д. 16	293856 г. Тамбов, ул. Первомайская, д. 25	1385659269586235	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">В главное меню</a>							

Рисунок 14 – Поставщик добавлен в справочник

При выборе функции «Edit» открывается форма редактирования поставщика, заполненная имеющимися данными (рисунок 15).



**Редактирование поставщика**

Наименование  
ООО "Сокол"

ИНН  
192821985691

Юридический адрес  
918356 г. Донецк, ул. Текстильщи

Адрес банка  
293856 г. Тамбов, ул. Первомайс

Номер счёта  
1385659269586235

[К списку поставщиков](#)

Рисунок 15 – Форма редактирования поставщика

Все формы при редактировании также проверяются на заполненность и имеют ограничение по числу символов. После редактирования сохранение изменений подтверждается кнопкой «Отправить».

Нажатие кнопки «Удалить» в таблице удаляет соответствующую строку.

Аналогично Справочнику поставщиков выполняется просмотр и создание/удаление элементов Справочника документов (рисунки 16, 17 и 18).

### Добавление документа

Номер  
45

Дата (гггг-мм-дд)  
21.12.2022

дек. 2022 г. ◀ ● ▶

Пн	Вт	Ср	Чт	Пт	Сб	Вс
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

ОАО "Бетон"

ОАО "Вышиванка"

Отправить

[К списку документов](#)

Рисунок 16 – Форма добавления документа (заполнение поля даты)

### Добавление документа

Номер  
45

Дата (гггг-мм-дд)  
06.12.2022

Наименование  
Договор о поставке верёв

Поставщик

Поставщики из справочника

ИП "Верёвочкин"

ОАО "Бетон"

ОАО "Вышиванка"

Отправить

[К списку документов](#)

Рисунок 17 – Форма добавления документа



Справочник документов							
<a href="#">Добавить документ</a>							
Код документа	Номер документа	Дата	Наименование	Поставщик	Материал		
1	7453	2022-12-03	Договор о поставке ниток	ООО "Вышиванка"	Нитки	<a href="#">Add</a>	<a href="#">Delete</a>
2	12	2022-12-01	Договор о поставке бетона	ООО "Бетон"	Бетон	<a href="#">Add</a>	<a href="#">Delete</a>
3	666	2022-10-07	Соглашение о поставке воска	ИП "Пчёлка"	Воск	<a href="#">Add</a>	<a href="#">Delete</a>
4	45	2022-12-06	Договор о поставке верёвки	ИП "Верёвочкин"		<a href="#">Add</a>	<a href="#">Delete</a>
<a href="#">В главное меню</a>							

Рисунок 18 – Отображение справочника документов

Для добавления к документу поставляемого материала (или обновления уже имеющегося) нужно нажать кнопку «Add» и выбрать нужный материал из справочника (рисунок 19).

## Добавление поставляемого материала

Материал

Мёд  
Бетон  
**Верёвка**  
Нитки

Отправить

[К списку документов](#)

Рисунок 19 – Добавление поставляемого материала к документу

Пример добавления и отображения материалов (аналогично предыдущим справочникам) представлен на рисунках 20 и 21. При отображении справочника материалов они сортируются по возрастанию кодов классов и для каждого класса по возрастанию кодов групп.

## Добавление материала

Класс

Группа

Наименование

Мера

Мера из справочника  
Блок  
Бочка  
**Грамм**

[К списку материалов](#)

Рисунок 20 – Добавление материала

### Справочник материалов

[Добавить материал](#)

Код материала	Класс	Группа	Наименование	Мера	
1	1	1	Воск	Блок	<input type="button" value="Delete"/>
2	1	2	Мёд	Бочка	<input type="button" value="Delete"/>
4	2	1	Верёвка	Километр	<input type="button" value="Delete"/>
6	2	13	Порох	Грамм	<input type="button" value="Delete"/>
5	2	15	Нитки	Метр	<input type="button" value="Delete"/>
3	3	2	Бетон	Грамм	<input type="button" value="Delete"/>

[В главное меню](#)

Рисунок 21 – Отображение справочника материалов

Добавление и отображение единиц измерения (аналогично предыдущим справочникам) представлено на рисунках 22 и 23.

## Добавление единицы измерения

Единица измерения

[К справочнику единиц измерения](#)

Рисунок 22 – Добавление единицы измерения

## Справочник единиц измерения

[Добавить единицу измерения](#)

Код е.и.	Единица измерения	
5	Блок	<input type="button" value="Delete"/>
4	Бочка	<input type="button" value="Delete"/>
3	Грамм	<input type="button" value="Delete"/>
1	Километр	<input type="button" value="Delete"/>
2	Метр	<input type="button" value="Delete"/>

[В главное меню](#)

Рисунок 23 – Отображение справочника единиц измерения

Добавление и отображение единиц хранения представлено на рисунках 24 и 25. Характеристика «Счёт материала» заполняется автоматически при добавлении записи (произведение количества на цену единицы измерения).

### Добавление единицы хранения

Дата

02.12.2022

Балансный счёт

0117568876

Сочетания из справочника

Сочетания из справочника

1 - 7453 - Договор о поставке ниток - ООО "Вышиванка" - 3 - Нитки - 5 - Метр - 2  
2 - 12 - Договор о поставке бетона - ООО "Бетон" - 2 - Бетон - 3 - Грамм - 3  
3 - 666 - Соглашение о поставке воска - ИП "Пчёлка" - 4 - Воск - 1 - Блок - 5  
4 - 45 - Договор о поставке верёвки - ИП "Верёвочкин" - 1 - Верёвка - 4 - Километр - 1

Цена единицы измерения, руб.

1200

Количество пришедшего материала

2

Отправить

[К списку документов](#)

Рисунок 24 – Добавление единицы хранения

### Единицы хранения

[Добавить единицу хранения](#)

Код ордера	Дата	Код поставщика	Балансный счёт	Код сопр. документа	Номер сопр. документа	Код материала	Счёт материала	Код ед. изм-я	Кол-во пришедшего материала	Цена ед. изм-я	
4	2022-12-02	3	1362473546	1	7453	5	30	2	6	5	Delete
5	2022-06-18	2	2460000240	2	12	3	300	3	1	300	Delete
6	2022-12-04	1	0896758647	4	45	4	2500	1	5	500	Delete
7	2022-12-02	4	0117568876	3	666	1	2400	5	2	1200	Delete

[В главное меню](#)

Рисунок 25 – Отображение единиц хранения

Для того чтобы подсчитать количество поставщиков материала нужно на соответствующей странице нажать кнопку «Count» напротив нужного материала. Материалы приведены в соответствии с документами о поставке. Пример подсчёта количества поставщиков материала представлен на рисунках 26 и 27.

**Посчитать количество поставщиков материала из справочника**  
Поставляемые материалы  
1. Бетон   
2. Верёвка   
3. Воск   
4. Нитки   
[В главное меню](#)

Рисунок 26 – Меню подсчёта (материалы)

Бетон поставляют: 1 поставщика(ов).  
[Назад](#)

Рисунок 27 – Результат подсчёта

Для того чтобы подсчитать количество поставщиков, пользующихся услугами одного банка, нужно на соответствующей странице нажать кнопку «Count» напротив адреса нужного банка. Адреса банков приведены в соответствии с информацией о поставщиках из справочника. Пример подсчёта количества поставщиков представлен на рисунках 28 и 29.

## Посчитать количество поставщиков пользующихся услугами банка

Адреса банков из справочника поставщиков

1. 912758 г. Саратов, ул. Нечаева, д. 10
2. 567893 г. Северодвинск, ул. Славы Бестужева, д. 18
3. 512523 г. Тамбов, ул. Волковская, д. 29
4. 293856 г. Тамбов, ул. Первомайская, д. 25

[В главное меню](#)

Рисунок 28 – Меню подсчёта (адреса банков)

Услугами банка по адресу 912758 г. Саратов, ул. Нечаева, д. 10 пользуются: 3 поставщика(ов).

[Назад](#)

Рисунок 29 – Результат подсчёта

Для того чтобы узнать поставщиков конкретного материала (с просмотром характеристик материала), нужно на соответствующей странице нажать кнопку «Ok» напротив адреса нужного материала. Материалы приведены в соответствии с документами о поставке. Пример результата запроса представлен на рисунках 30 и 31.

## Вывести список поставщиков с указанием всех реквизитов данного материала на склад

Поставляемые материалы

1. Бетон
2. Верёвка
3. Воск
4. Нитки

[В главное меню](#)

30 – Меню материалов

## **Список поставщиков материала**

Наименование: Бетон,  
Класс: 3, группа: 2  
Единица измерения: Грамм

### **Поставщики:**

1. ООО "Бетон"

[Назад](#)

31 – Результат запроса

Для пользователя с ограниченными правами доступа (Наблюдатель) при попытке добавления/редактирования/удаления элементов таблиц выводится запрещающее сообщение (пример на рисунке 32).

**У вас нет прав на изменение состояния таблиц**

[К справочнику поставщиков](#)

Рисунок 32 – Сообщение для Наблюдателя

В случае, если пользователь попыбует добавить объект, нарушив при этом уникальность характеристик, требующих обязательного выполнения этого свойства, объект не добавится, а приложение выведет соответствующее запрещающее сообщение (пример на рисунках 33 и 34).

## Добавление поставщика

Наименование

ООО "Бетон"

ИНН

345364758697

Юридический адрес

Почтовый индекс

375268

Город

Пенза

Улица

Мира

Дом

47

Адрес банка

Почтовый индекс

293856

Город

Пенза

Улица

Галкиной

Дом

12

Номер счёта

0238562395629356

Отправить

[К списку поставщиков](#)

Рисунок 33 – Добавление поставщика с неуникальным наименованием

Поставщик с таким наименованием уже находится в справочнике поставщиков!

[К списку поставщиков](#)

Рисунок 34 – Запрещающее сообщение



## ПРИЛОЖЕНИЕ Б. ЛИСТИНГ ПРОГРАММНОГО КОДА

### Главный файл app.js:

```
const mysql = require("mysql2");
const express = require("express");

const app = express();
const urlencodedParser = express.urlencoded({extended:
false});

const pool = mysql.createPool({
  connectionLimit: 5,
  host: "localhost",
  user: "Director",
  database: "sklad",
  password: "germanchef111"
});
const user = 'Director'

// const pool = mysql.createPool({
//   connectionLimit: 5,
//   host: "localhost",
//   user: "Viewer",
//   database: "sklad",
//   password: "germanchef222"
// });
// const user = 'Viewer'

app.set("view engine", "hbs");

// получение меню
app.get("/", function(req, res){
  res.render("index.hbs");
});
```

```

//поставщики
{
// получение списка поставщиков
app.get("/index_vendor", function(req, res){
    pool.query("SELECT * FROM Vendor", function(err, data) {
        if(err) return console.log(err);
        res.render("vendor/index_vendor.hbs", {
            Vendor: data
        });
    });
});

// возвращаем форму для добавления данных
app.get("/create_vendor", function(req, res){
    if(user=="Viewer"){
        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a href="/index_vendor">К справочнику
поставщиков</a>');
    } else {
        res.render("vendor/create_vendor.hbs");
    }
});

// получаем отправленные данные и добавляем их в БД
app.post("/create_vendor", urlencodedParser, function (req,
res) {

    if(!req.body) return res.sendStatus(400);
    const name = req.body.name;
    const inn = req.body.inn;
    const adress = req.body.adress_1 + ' г. ' +
req.body.adress_2 + ', ул. ' + req.body.adress_3 + ', д. ' +
req.body.adress_4;

```

```

        const bank_address = req.body.bank_address_1 + ' г. ' +
req.body.bank_address_2 + ', ул. ' + req.body.bank_address_3 + ', д.
' + req.body.bank_address_4;

        const bank_number = req.body.bank_number;

        pool.query("INSERT INTO Vendor (VendorName, VendorINN,
VendorAddress, VendorBankAddress, VendorBankNumber) VALUES
(?,?,?,?,?)", [name, inn, adress, bank_address, bank_number],
function(err, data) {
            if(err) {
                res.send('Поставщик с таким наименованием уже
находится в справочнике поставщиков! <br/><br/><a
href="/index_vendor">К списку поставщиков</a>');
                return console.log(err);
            }
            else{
                res.redirect("/index_vendor");
            }
        });
    });

    // получем id редактируемого пользователя, получаем его из бд
и отправляю с формой редактирования
    app.get("/edit_vendor/:id", function(req, res){
        const id = req.params.id;
        if(user=="Viewer"){
            res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a href="/index_vendor">К справочнику
поставщиков</a>');
        } else {
            pool.query("SELECT * FROM Vendor WHERE VendorID=?", [id],
function(err, data) {
                if(err) return console.log(err);
                res.render("vendor/edit_vendor.hbs", {
                    Vendor: data[0]
                });
            });
        }
    });

```

```

    });
  }
});

// получаем отредактированные данные и отправляем их в БД
app.post("/edit_vendor", urlencodedParser, function (req,
res) {
  if(!req.body) return res.sendStatus(400);
  const id = req.body.id;
  const name = req.body.name;
  const inn = req.body.inn;
  const adress = req.body.adress;
  const bank_adress = req.body.bank_adress;
  const bank_number = req.body.bank_number;
  pool.query("UPDATE Vendor SET VendorName=?, VendorINN=?,
VendorAdress=?, VendorBankAdress=?, VendorBankNumber=? WHERE
VendorID=?", [name, inn, adress, bank_adress, bank_number, id],
function(err, data) {
  if(err) {
    res.send('Поставщик с таким наименованием уже
находится в справочнике поставщиков! <br/><br/><a
href="/index_vendor">К списку поставщиков</a>');
    return console.log(err);
  }
  res.redirect("/index_vendor");
});
});

// получаем id удаляемого пользователя и удаляем его из бд
app.post("/delete_vendor/:id", function(req, res){

  const id = req.params.id;
  if(user=="Viewer"){

```

```

        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a          href="/index_vendor">К          справочнику
поставщиков</a>');
    } else {
        pool.query("DELETE FROM Vendor WHERE VendorID=?", [id],
function(err, data) {
            if(err) return console.log(err);
            res.redirect("/index_vendor");
        });
    }
});
}

//документы
{
// получение списка документов
app.get("/index_document", function(req, res){
    pool.query("SELECT * FROM Document", function(err, data)
{
        if(err) return console.log(err);
        res.render("document/index_document.hbs", {
            Document: data
        });
    });
});

// возвращаем форму для добавления данных
app.get("/create_document", function(req, res){
    if(user=="Viewer"){
        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a          href="/index_document">К          справочнику
документов</a>');
    } else {
        pool.query("SELECT * FROM Vendor", function(err, data) {
            if(err) return console.log(err);

```

```

        res.render("document/create_document.hbs", {
            Vendor: data
        });
    });
}
});

// получаем отправленные данные и добавляем их в БД
app.post("/create_document",    urlencodedParser,    function
(req, res) {

    if(!req.body) return res.sendStatus(400);
    const num = req.body.num;
    const date = req.body.date;
    const name = req.body.name;
    const vendor = req.body.vendor;
    pool.query("INSERT INTO Document (DocumentNum,
DocumentDate, DocumentName, DocumentVendor) VALUES (?, ?, ?, ?)",
[num, date, name, vendor], function(err, data) {
        if(err) {
            res.send('Данные некорректны <br/><br/><a
href="/index_document">К списку документов</a>');
            return console.log(err);
        }
        res.redirect("/index_document");
    });
});

//добавление материала к документу
app.get("/add_material/:id", function(req, res){
    const id = req.params.id;
    if(user=="Viewer"){
        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a href="/index_document">К справочнику
документов</a>');
    }
}

```

```

    } else {
        pool.query("SELECT * FROM Material", function(err, data)
{
            if(err) return console.log(err);
            res.render("document/add_material.hbs", {
                Material: data, id
            });
        });
    }
});

// получаем данные и отправляем их в БД
app.post("/add_material", urlencodedParser, function (req,
res) {
    if(!req.body) return res.sendStatus(400);
    const id = req.body.id;
    const material = req.body.material;
    pool.query("UPDATE Document SET DocumentMaterial=? WHERE
DocumentID=?", [material, id], function(err, data) {
        if(err) {
            res.send('чет-не то, дружище <br/><br/><a
href="/index_document">К списку документов</a>');
            return console.log(err);
        }
        res.redirect("/index_document");
    });
});

// получаем id удаляемого и удаляем его из бд
app.post("/delete_document/:id", function(req, res){

    const id = req.params.id;
    if(user=="Viewer"){

```

```

        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a          href="/index_document">К          справочнику
документов</a>');
    } else {
        pool.query("DELETE FROM Document WHERE DocumentID=?",
[id], function(err, data) {
            if(err) return console.log(err);
            res.redirect("/index_document");
        });
    }
});
}

//меры
{
// получение списка мер
app.get("/index_measure", function(req, res){
    pool.query("SELECT * FROM Measure", function(err, data) {
        if(err) return console.log(err);
        res.render("measure/index_measure.hbs", {
            Measure: data
        });
    });
});

// возвращаем форму для добавления данных
app.get("/create_measure", function(req, res){
    if(user=="Viewer"){
        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a          href="/index_measure">К          справочнику          единиц
измерения</a>');
    } else {
        res.render("measure/create_measure.hbs");
    }
});
}

```



```

// получаем отправленные данные и добавляем их в БД
app.post("/create_measure", urlencodedParser, function (req,
res) {

    if(!req.body) return res.sendStatus(400);
    const name = req.body.name;
    pool.query("INSERT INTO Measure (MeasureName) VALUES (?)",
[name], function(err, data) {
        if(err) {
            res.send('Такая единица измерения уже есть в
справочнике <br/><br/><a href="/index_measure">К справочнику
единиц измерения</a>');
            return console.log(err);
        }
        res.redirect("/index_measure");
    });
});

// получаем id удаляемого и удаляем его из бд
app.post("/delete_measure/:id", function(req, res){

    const id = req.params.id;
    if(user=="Viewer"){
        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a href="/index_measure">К справочнику единиц
измерения</a>');
    } else {
        pool.query("DELETE FROM Measure WHERE MeasureID=?", [id],
function(err, data) {
            if(err) return console.log(err);
            res.redirect("/index_measure");
        });
    }
});

```

```

    }

    //материалы
    {
    // получение списка материалов
    app.get("/index_material", function(req, res){
        pool.query("SELECT * FROM Material ORDER BY MaterialKlass,
MaterialGroup", function(err, data) {
            if(err) return console.log(err);
            res.render("material/index_material.hbs", {
                Material: data
            });
        });
    });

    // возвращаем форму для добавления данных
    app.get("/create_material", function(req, res){
        if(user=="Viewer"){
            res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a href="/index_material">К справочнику
материалов</a>');
        } else {
            pool.query("SELECT * FROM Measure", function(err, data)
            {
                if(err) return console.log(err);
                res.render("material/create_material.hbs", {
                    Measure: data
                });
            });
        }
    });

    // получаем отправленные данные и добавляем их в БД
    app.post("/create_material", urlencodedParser, function
    (req, res) {

```

```

    if(!req.body) return res.sendStatus(400);
    const klass = req.body.klass;
    const group = req.body.group;
    const name = req.body.name;
    const measure = req.body.measure;
    pool.query("INSERT INTO Material (MaterialKlass,
MaterialGroup, MaterialName, MaterialMeasure) VALUES (?, ?, ?,
?)", [klass, group, name, measure], function(err, data) {
        if(err) {
            res.send('что-то не так <br/><br/><a
href="/index_material">К списку материалов</a>');
            return console.log(err);
        }
        res.redirect("/index_material");
    });
});

// получаем id удаляемого и удаляем его из бд
app.post("/delete_material/:id", function(req, res){

    const id = req.params.id;
    if(user=="Viewer"){
        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a href="/index_material">К справочнику
материалов</a>');
    } else {
        pool.query("DELETE FROM Material WHERE MaterialID=?",
[id], function(err, data) {
            if(err) return console.log(err);
            res.redirect("/index_material");
        });
    }
});
}

```

```

//единицы хранения
{
    // получение списка
    app.get("/index_order", function(req, res){
        pool.query("SELECT * FROM Material_Order", function(err,
data) {
            if(err) return console.log(err);
            res.render("order/index_order.hbs", {
                Material_Order: data
            });
        });
    });

    // возвращаем форму для добавления данных
    app.get("/create_order", function(req, res){
        if(user=="Viewer"){
            res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a href="/index_order">Назад</a>');
        } else {
            pool.query("SELECT Vendor.VendorID, Vendor.VendorName,
Document.DocumentID, \
                Document.DocumentNum,                Document.DocumentName,
Document.DocumentVendor, \
                Document.DocumentMaterial,                Material.MaterialID,
Material.MaterialMeasure, Measure.MeasureID, Measure.MeasureName
\
                FROM Vendor, Document, Material, Measure \
                WHERE Vendor.VendorName = Document.DocumentVendor AND
Material.MaterialName      =      Document.DocumentMaterial      AND
Measure.MeasureName = Material.MaterialMeasure",
            function(err, data) {
                if(err) return console.log(err);
                res.render("order/create_order.hbs", {
                    Some: data

```

```

        });
    });
}
});

// получаем отправленные данные и добавляем их в БД
app.post("/create_order", urlencodedParser, function (req,
res) {

    if(!req.body) return res.sendStatus(400);
    const date = req.body.date;
    const num = req.body.num;
    const document = req.body.document;
    const measure_price = req.body.measure_price;
    const material_sum = req.body.material_sum;
    const material_num = measure_price * material_sum;
    pool.query("INSERT INTO Material_Order (OrderDate,
OrderVendorID, OrderNum,\
    OrderDocumentID, OrderDocumentNum, OrderMaterialID,
OrderMaterialNum, OrderMeasureID, OrderMaterialSum,
OrderMeasurePrice) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)",
    [date, document.split(' - ')[4], num, document.split('
- ')[0], document.split(' - ')[1], document.split(' - ')[6],
material_num, document.split(' - ')[8], material_sum,
measure_price], function(err, data) {
        if(err) {
            res.send('что-то не так <br/><br/><a
href="/index_order">К списку единиц хранения</a>');
            return console.log(err);
        }
        res.redirect("/index_order");
    });
});

// получаем id удаляемого и удаляем его из бд

```

```

app.post("/delete_order/:id", function(req, res){

    const id = req.params.id;
    if(user=="Viewer"){
        res.send('У вас нет прав на изменение состояния таблиц
<br/><br/><a href="/index_order">Назад</a>');
    } else {
        pool.query("DELETE      FROM      Material_Order      WHERE
OrderID=?", [id], function(err, data) {
            if(err) return console.log(err);
            res.redirect("/index_order");
        });
    }
});

//количество поставщиков материала
{
    app.get("/index_count_material", function(req, res){
        pool.query("SELECT  DISTINCT  DocumentMaterial  FROM
Document", function(err, data) {
            if(err) return console.log(err);

res.render("count_material/index_count_material.hbs", {
            Document: data
        });
    });
});

app.post("/count_material/:material",      urlencodedParser,
function(req, res){

    const material = req.params.material;

```

```

        pool.query("SELECT COUNT(DISTINCT DocumentVendor) FROM
Document WHERE DocumentMaterial=?", [material], function(err,
data) {

    if(err) return console.log(err);
    var data1 = JSON.stringify(data);
    var data2 = parseInt(data1.match(/\d+/));
    res.send(material + ' поставляют: ' + data2.toString()
+
    ' поставщика(ов). <br/><br/><a
href="/index_count_material">Назад</a>');
    });
});
}

//количество поставщиков по адресу банка
{
    app.get("/index_count_bank", function(req, res){
        pool.query("SELECT DISTINCT VendorBankAdress FROM
Vendor", function(err, data) {
            if(err) return console.log(err);
            res.render("count_bank/index_count_bank.hbs", {
                Vendor: data
            });
        });
    });

    app.post("/count_bank/:bank", urlencodedParser,
function(req, res){

        const bank = req.params.bank;
        pool.query("SELECT COUNT(DISTINCT VendorName) FROM Vendor
WHERE VendorBankAdress=?", [bank], function(err, data) {
            if(err) return console.log(err);
            var data1 = JSON.stringify(data);
            var data2 = parseInt(data1.match(/\d+/));

```

```

        res.send('Услугами банка по адресу ' + bank + '
пользуются: ' + data2.toString() + ' поставщика(ов). <br/><br/><a
href="/index_count_bank">Назад</a>');
    });
});
}

//список поставщиков с реквизитами материала
{
    app.get("/index_count_vendor", function(req, res){
        pool.query("SELECT DISTINCT DocumentMaterial FROM
Document", function(err, data) {
            if(err) return console.log(err);
            res.render("count_vendor/index_count_vendor.hbs", {
                Document: data
            });
        });
    });

    app.post("/count_vendor/:material",      urlencodedParser,
function(req, res){

        const material = req.params.material;
        pool.query("SELECT DISTINCT Material.MaterialName,
Material.MaterialKlass, Material.MaterialGroup,
Material.MaterialMeasure, \
Document.DocumentVendor FROM Material, Document WHERE
Document.DocumentMaterial=? and
Material.MaterialName=Document.DocumentMaterial", [material],
function(err, data) {
            if(err) return console.log(err);
            res.render("count_vendor/page_count_vendor.hbs", {
                Some: data, Smth: data[0]
            });
        });
    });
}

```



```

    });
}

app.listen(3000, function(){
    console.log("Сервер ожидает подключения...");
});

```

### **Файл страницы навигации index.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Навигация</title>
    <meta charset="utf-8" />
    <style type="text/css">
        H1 {
            font-size: 120%;
            font-family: Verdana, Arial, Helvetica, sans-serif;
            color: #336;
        }
        a {
            font-family: 'Open Sans', sans-serif;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1>Навигация</h1><br>
    <p><a href="/index_vendor">К списку поставщиков</a></p>
    <p><a href="/index_document">К справочнику
документов</a></p>
    <p><a href="/index_material">К справочнику
материалов</a></p>
    <p><a href="/index_measure">К справочнику мер</a></p>

```

```

        <p><a href="/index_order">К списку единиц
хранения</a></p>
        <br><br>
        <p><a href="/index_count_material">Посчитать количество
поставщиков материала из справочника</a></p>
        <p><a href="/index_count_bank">Посчитать количество
поставщиков пользующихся услугами банка</a></p>
        <p><a href="/index_count_vendor">Вывести список
поставщиков с указанием всех реквизитов данного материала на
склад</a></p>
    </body>
</html>

```

### **Файл справочника поставщиков index\_vendor.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Список поставщиков</title>
    <meta charset="utf-8" />
    <style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}
TABLE {
    border-spacing: 0 10px;
    font-family: 'Open Sans', sans-serif;
    font-weight: bold;
}
td {
    vertical-align: middle;
    padding: 10px;
    font-size: 14px;

```

```

        text-align: center;
        border-top: 2px solid #56433D;
        border-bottom: 2px solid #56433D;
        border-right: 2px solid #56433D;
    }
    th {
        padding: 10px 20px;
        background: #56433D;
        color: #F9C941;
        border-right: 2px solid;
        font-size: 0.9em;
    }
    a {
        font-family: 'Open Sans', sans-serif;
        font-weight: bold;
    }
</style>
</head>
<body>
    <h1>Список поставщиков</h1>
    <p><a href="/create_vendor">Добавить поставщика</a></p>
    <table cellpadding="5" border="1">
        <tr><th>Код поставщика</th>
        <th>Наименование</th>
        <th>ИНН</th><th>Юридический адрес</th>
        <th>Адрес банка</th>
        <th>Номер счёта</th>
        <th></th>
        <th></th></tr>
        {{#each Vendor}}
            <tr>
                <td>{{this.VendorID}}</td>
                <td>{{this.VendorName}}</td>
                <td>{{this.VendorINN}}</td>
                <td>{{this.VendorAdress}}</td>

```

```

        <td>{{this.VendorBankAdress}}</td>
        <td>{{this.VendorBankNumber}}</td>
        <td>
            <a
href="/edit_vendor/{{this.VendorID}}">Edit</a>
            </td>
        <td>
            <form
action="delete_vendor/{{this.VendorID}}"                method="POST"
style="display:inline;"
            <input                                type="submit"
value="Delete"/>
            </form>
        </td>
    </tr>
    {{/each}}
</table>
<p><a href="/">В главное меню</a></p>
</body>
<html>

```

### **Файл создания поставщиков create\_vendor.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Добавление поставщика</title>
    <meta charset="utf-8" />
    <style>.field{width:180px;}</style>
    <style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}

```

```

        </style>
</head>
<body>
    <h1>Добавление поставщика</h1><br>
    <form method="POST">
        <label>Наименование</label><br>
        <input name="name" maxlength="50" required><br><br>
        <label>ИНН</label><br>
        <input name="inn" maxlength="12" required><br><br>
        <label>Юридический адрес</label><br><br>
        Почтовый индекс
        <br><input name="adress_1" maxlength="6" required>
        Город
        <input name="adress_2" maxlength="25" required><br>
        Улица
        <br><input name="adress_3" maxlength="25" required>
        Дом
        <input name="adress_4" maxlength="25" required>
        <br><br><br>
        <label>Адрес банка</label><br><br>
        Почтовый индекс
        <br><input      name="bank_adress_1"      maxlength="6"
required>
        Город
        <input      name="bank_adress_2"      maxlength="25"
required><br>
        Улица
        <br><input      name="bank_adress_3"      maxlength="25"
required>
        Дом
        <input name="bank_adress_4" maxlength="25" required>
        <br><br><br>
        <label>Номер счёта</label><br>
        <input      name="bank_number"      maxlength="16"
required><br><br>

```

```

        <input type="submit" value="Отправить" />
    </form>
    <br><a href="/index_vendor">К списку поставщиков</a>
</body>
<html>

```

### **Файл редактирования поставщиков edit\_vendor.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Редактирование поставщика</title>
    <meta charset="utf-8" />
    <style>.field{width:180px;}</style>
    <style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}
    </style>
</head>
<body>
    <h1>Редактирование поставщика</h1>
    <form action="/edit_vendor" method="POST">
        <input
            type="hidden"
            name="id"
value="{{Vendor.VendorID}}" />
        <label>Наименование</label><br>
        <input
            name="name"
            value="{{Vendor.VendorName}}"
class="field"/><br><br>
        <label>ИНН</label><br>
        <input
            name="inn"
            value="{{Vendor.VendorINN}}"
class="field"/><br><br>
        <label>Юридический адрес</label><br>

```

```

        <input name="adress" value="{{Vendor.VendorAdress}}"
class="field"/><br><br>
        <label>Адрес банка</label><br>
        <input
                                name="bank_adress"
value="{{Vendor.VendorBankAdress}}" class="field"/><br><br>
        <label>Номер счёта</label><br>
        <input
                                name="bank_number"
value="{{Vendor.VendorBankNumber}}" class="field"/><br><br>

        <input type="submit" value="Отправить"/>
    </form>
    <br><a href="/index_vendor">К списку поставщиков</a>
</body>
<html>

```

### **Файл справочника документов index\_document.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Справочник документов</title>
    <meta charset="utf-8" />
    <style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}
TABLE {
    border-spacing: 0 10px;
    font-family: 'Open Sans', sans-serif;
    font-weight: bold;
}
td {
    vertical-align: middle;

```

```

padding: 10px;
font-size: 14px;
text-align: center;
border-top: 2px solid #56433D;
border-bottom: 2px solid #56433D;
border-right: 2px solid #56433D;
}
th {
padding: 10px 20px;
background: #56433D;
color: #F9C941;
border-right: 2px solid;
font-size: 0.9em;
}
a {
font-family: 'Open Sans', sans-serif;
font-weight: bold;
}
</style>
</head>
<body>
<h1>Справочник документов</h1>
<p><a href="/create_document">Добавить документ</a></p>
<table cellpadding="5" border="1">
<tr><th>Код<th>Номер
документа</th><th>Дата</th><th>Наименование</th><th>Поставщик</t
h><th>Материал</th><th></th><th></th></tr>
{{#each Document}}
<tr>
<td>{{this.DocumentID}}</td>
<td>{{this.DocumentNum}}</td>
<td>{{this.DocumentDate}}</td>
<td>{{this.DocumentName}}</td>
<td>{{this.DocumentVendor}}</td>
<td>{{this.DocumentMaterial}}</td>

```



```

        <td>
            <a
href="/add_material/{{this.DocumentID}}">Add</a>
        </td>
        <td>
            <form
action="delete_document/{{this.DocumentID}}"                method="POST"
style="display:inline;"
            <input                                type="submit"
value="Delete"/>
            </form>
        </td>
    </tr>
    {{/each}}
</table>
    <p><a href="/">В главное меню</a></p>
</body>
<html>

```

### **Файл создания документов create\_document.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Добавление документа</title>
    <meta charset="utf-8" />
    <style>.field{width:180px;}</style>
    <style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}
    </style>
</head>

```

```

<body>
  <h1>Добавление документа</h1>
  <form method="POST">
    <label>Номер</label><br>
    <input name="num" required><br><br>
    <label>Дата (гггг-мм-дд)</label><br>
    <input type="date" name="date" required><br><br>
    <label>Наименование</label><br>
    <input name="name" required><br><br>
    <label>Поставщик</label><br>
    <p><select size="3" multiple name="vendor" required>
      <option disabled>Поставщики из
справочника</option>
      {{#each Vendor}}
        <option>{{this.VendorName}}</option>
      {{/each}}
    </select></p>
    <input type="submit" value="Отправить" />

  </form>
  <a href="/index_document">К списку документов</a>
</body>
<html>

```

### **Файл добавления материалов add\_material.hbs:**

```

<!DOCTYPE html>
<html>
<head>
  <title>Добавление поставляемого материала</title>
  <meta charset="utf-8" />
  <style>.field{width:180px;}</style>
  <style type="text/css">
H1 {
  font-size: 120%;

```

```

        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: #336;
    }
</style>
</head>
<body>
    <h1>Добавление поставляемого материала</h1>
    <form action="/add_material" method="POST">
        <input type="hidden" name="id" value="{{id}}"/>
        <label>Материал</label><br>
        <p><select      size="3"      multiple      name="material"
required>
                <option          disabled>Поставщики          из
справочника</option>
                {{#each Material}}
                    <option>{{this.MaterialName}}</option>
                {{/each}}
            </select></p>
            <input type="submit" value="Отправить" />

        </form>
        <a href="/index_document">К списку документов</a>
</body>
<html>

```

### **Файл справочника единиц измерения index\_measure.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Справочник единиц измерения</title>
    <meta charset="utf-8" />
    <style type="text/css">
        H1 {
            font-size: 120%;

```

```

        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: #336;
    }
    TABLE {
        border-spacing: 0 10px;
        font-family: 'Open Sans', sans-serif;
        font-weight: bold;
    }
    td {
        vertical-align: middle;
        padding: 10px;
        font-size: 14px;
        text-align: center;
        border-top: 2px solid #56433D;
        border-bottom: 2px solid #56433D;
        border-right: 2px solid #56433D;
    }
    th {
        padding: 10px 20px;
        background: #56433D;
        color: #F9C941;
        border-right: 2px solid;
        font-size: 0.9em;
    }
    a {
        font-family: 'Open Sans', sans-serif;
        font-weight: bold;
    }
</style>
</head>
<body>
    <h1>Справочник единиц измерения</h1>
    <p><a href="/create_measure">Добавить единицу
измерения</a></p>
    <table cellpadding="5" border="1">

```

```

        <tr><th>Код                                     е.и.</th><th>Единица
измерения</th><th></th></tr>
        {{#each Measure}}
            <tr>
                <td>{{this.MeasureID}}</td>
                <td>{{this.MeasureName}}</td>
                <td>
                    <form
action="delete_measure/{{this.MeasureID}}"                method="POST"
style="display:inline;"
                    <input                                type="submit"
value="Delete"/>
                </form>
            </td>
        </tr>
    {{/each}}
</table>
<p><a href="/">В главное меню</a></p>
</body>
<html>

```

### **Файл создания единиц измерения create\_measure.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Добавление единицы измерения</title>
    <meta charset="utf-8" />
    <style>.field{width:180px;}</style>
    <style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}

```

```

        </style>
</head>
<body>
    <h1>Добавление единицы измерения</h1>
    <form method="POST">
        <label>Единица измерения</label><br>
        <input name="name" required><br><br>
        <input type="submit" value="Отправить" />
    </form>
    <a href="/index_measure">К справочнику единиц
измерения</a>
</body>
<html>

```

### **Файл справочника материалов index\_material.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Справочник материалов</title>
    <meta charset="utf-8" />
    <style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}
TABLE {
    border-spacing: 0 10px;
    font-family: 'Open Sans', sans-serif;
    font-weight: bold;
}
td {
    vertical-align: middle;
    padding: 10px;
}

```

```

        font-size: 14px;
        text-align: center;
        border-top: 2px solid #56433D;
        border-bottom: 2px solid #56433D;
        border-right: 2px solid #56433D;
    }
    th {
        padding: 10px 20px;
        background: #56433D;
        color: #F9C941;
        border-right: 2px solid;
        font-size: 0.9em;
    }
    a {
        font-family: 'Open Sans', sans-serif;
        font-weight: bold;
    }
</style>
</head>
<body>
    <h1>Справочник материалов</h1>
    <p><a href="/create_material">Добавить материал</a></p>
    <table cellpadding="5" border="1">
        <tr><th>Код
материала</th><th>Класс</th><th>Группа</th><th>Наименование</th>
<th>Мера</th><th></th></tr>
        {{#each Material}}
            <tr>
                <td>{{this.MaterialID}}</td>
                <td>{{this.MaterialKlass}}</td>
                <td>{{this.MaterialGroup}}</td>
                <td>{{this.MaterialName}}</td>
                <td>{{this.MaterialMeasure}}</td>
                <td>

```

```

                                <form
action="delete_material/{{this.MaterialID}}"                method="POST"
style="display:inline;"
                                <input                        type="submit"
value="Delete"/>
                                </form>
                                </td>
                                </tr>
                                {{/each}}
                                </table>
                                <p><a href="/">В главное меню</a></p>
                                </body>
                                <html>

```

### **Файл создания материалов create\_material.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Добавление материала</title>
    <meta charset="utf-8" />
    <style>.field{width:180px;}</style>
    <style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}
    </style>
</head>
<body>
    <h1>Добавление материала</h1>
    <form method="POST">
        <label>Класс</label><br>
        <input name="klass" required><br><br>

```



```

<label>Группа</label><br>
<input name="group" required><br><br>
<label>Наименование</label><br>
<input name="name" required><br><br>
<label>Мера</label><br>
<p><select size="3" multiple name="measure" required>
  <option disabled>Мера из справочника</option>
  {{#each Measure}}
    <option>{{this.MeasureName}}</option>
  {{/each}}
</select></p>
<input type="submit" value="Отправить" />

</form>
<a href="/index_material">К списку материалов</a>
</body>
<html>

```

### **Файл справочника единиц измерения index\_order.hbs:**

```

<!DOCTYPE html>
<html>
<head>
  <title>Единицы хранения</title>
  <meta charset="utf-8" />
  <style type="text/css">
    H1 {
      font-size: 120%;
      font-family: Verdana, Arial, Helvetica, sans-serif;
      color: #336;
    }
    TABLE {
      border-spacing: 0 10px;
      font-family: 'Open Sans', sans-serif;
      font-weight: bold;
    }
  </style>

```

```

    }
    td {
        vertical-align: middle;
        padding: 10px;
        font-size: 14px;
        text-align: center;
        border-top: 2px solid #56433D;
        border-bottom: 2px solid #56433D;
        border-right: 2px solid #56433D;
    }
    th {
        padding: 10px 20px;
        background: #56433D;
        color: #F9C941;
        border-right: 2px solid;
        font-size: 0.9em;
    }
    a {
        font-family: 'Open Sans', sans-serif;
        font-weight: bold;
    }
</style>
</head>
<body>
    <h1>Единицы хранения</h1>
    <p><a href="/create_order">Добавить единицу
хранения</a></p>
    <div><table cellpadding="5" border="1">
        <tr><th>Код ордера</th><th>Дата</th><th>Код
поставщика</th><th>Балансный счёт</th><th>Код сопр.
документа</th><th>Номер сопр. документа</th><th>Код
материала</th><th>Счёт материала</th><th>Код ед. изм-
я</th><th>Кол-во пришедшего материала</th><th>Цена ед. изм-
я</th><th></th></tr>
        {{#each Material_Order}}
```

```

        <tr>
            <td>{{this.OrderID}}</td>
            <td>{{this.OrderDate}}</td>
            <td>{{this.OrderVendorID}}</td>
            <td>{{this.OrderNum}}</td>
            <td>{{this.OrderDocumentID}}</td>
            <td>{{this.OrderDocumentNum}}</td>
            <td>{{this.OrderMaterialID}}</td>
            <td>{{this.OrderMaterialNum}}</td>
            <td>{{this.OrderMeasureID}}</td>
            <td>{{this.OrderMaterialSum}}</td>
            <td>{{this.OrderMeasurePrice}}</td>
            <td>
                <form
                    action="delete_order/{{this.OrderID}}"                      method="POST"
                    style="display:inline;"
                                <input                                  type="submit"
                                value="Delete"/>
                </form>
            </td>
        </tr>
    {{/each}}
</table></div>
<p><a href="/">В главное меню</a></p>
</body>
<html>

```

### **Файл создания единиц измерения create\_order.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Добавление единицы хранения</title>
    <meta charset="utf-8" />
    <style>.field{width:180px;}</style>

```

```

<style type="text/css">
H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #336;
}
</style>
</head>
<body>
<h1>Добавление единицы хранения</h1>
<form method="POST">
    <label>Дата</label><br>
    <input type="date" name="date" required><br><br>
    <label>Балансный счёт</label><br>
    <input name="num" maxlength="10" required><br><br>

    <label>Сочетания из справочника</label><br>
    <p><select      size="6"      multiple      name="document"
required>
                                <option          disabled>Сочетания          из
справочника</option>
                                {{#each Some}}
                                    <option>{{this.DocumentID}}          -
{{this.DocumentNum}}          -          {{this.DocumentName}}          -
{{this.DocumentVendor}}          -          {{this.VendorID}}          -
{{this.DocumentMaterial}}          -          {{this.MaterialID}}          -
{{this.MaterialMeasure}} - {{this.MeasureID}}</option>
                                {{/each}}
                                </select></p>

    <label>Цена единицы измерения, руб.</label><br>
    <p><input      type="number"      name="measure_price"
value="0" min="0" max="100000" step="1" required></p><br>
    <label>Количество пришедшего материала</label><br>

```

```

        <p><input      type="number"      name="material_sum"
value="0" min="0" max="10000" step="1" required></p><br>

        <input type="submit" value="Отправить" />

    </form>
    <br><a href="/index_order">К списку документов</a>
</body>
<html>

```

### **Файл подсчета поставщиков по материалу count\_material.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Счет</title>
    <meta charset="utf-8" />
    <style type="text/css">
        H2 {
            font-size: 120%;
            font-family: Verdana, Arial, Helvetica, sans-serif;
            color: #336;
        }
    </style>
</head>
<body>
    <h2>Посчитать количество поставщиков материала из
справочника</h2>
    <label>Поставляемые материалы</label><br>
    <ol>
        {{#each Document}}
        <li>{{this.DocumentMaterial}}
            <form
action="count_material/{{this.DocumentMaterial}}" method="POST"
style="display:inline;">

```

```

        <input type="submit" value="Count"/>
    </form>
</li>
    {{/each}}
</ol>
<p><a href="/">В главное меню</a></p>
</body>
<html>

```

### **Файл подсчёта поставщиков по адресу банка count\_bank.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Счёт</title>
    <meta charset="utf-8" />
    <style type="text/css">
        H2 {
            font-size: 120%;
            font-family: Verdana, Arial, Helvetica, sans-serif;
            color: #336;
        }
    </style>
</head>
<body>
    <h2>Посчитать количество поставщиков пользующихся
услугами банка</h2>
    <label>Адреса банков из справочника
поставщиков</label><br>
    <ol>
        {{#each Vendor}}
        <li>{{this.VendorBankAdress}}
            <form
action="count_bank/{{this.VendorBankAdress}}" method="POST"
style="display:inline;">

```

```

        <input type="submit" value="Count"/>
    </form>
</li>
    {{/each}}
</ol>
    <p><a href="/">В главное меню</a></p>
</body>
<html>

```

### **Файл поиска поставщиков для материала index\_count\_vendor.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Инфо</title>
    <meta charset="utf-8" />
    <style type="text/css">
        H2 {
            font-size: 120%;
            font-family: Verdana, Arial, Helvetica, sans-serif;
            color: #336;
        }
    </style>
</head>
<body>
    <h2>Вывести список поставщиков с указанием всех
реквизитов данного материала на склад</h2>
    <label>Поставляемые материалы</label><br>
    <ol>
        {{#each Document}}
        <li>{{this.DocumentMaterial}}
            <form
action="count_vendor/{{this.DocumentMaterial}}"      method="POST"
style="display:inline;">
                <input type="submit" value="Ok"/>

```

```

        </form>
    </li>
    {{/each}}
</ol>
<p><a href="/">В главное меню</a></p>
</body>
<html>

```

## **Файл ответа по поиску поставщиков для материала page count vendor.hbs:**

```

<!DOCTYPE html>
<html>
<head>
    <title>Инфо</title>
    <meta charset="utf-8" />
    <style type="text/css">
        H2 {
            font-size: 120%;
            font-family: Verdana, Arial, Helvetica, sans-serif;
            color: #336;
        }
    </style>
</head>
<body>
    <h2>Список поставщиков материала</h2>
    Наименование:      {{Smth.MaterialName}},      <br>Класс:
    {{Smth.MaterialKlass}}, группа: {{Smth.MaterialGroup}}
    <br>Единица измерения: {{Smth.MaterialMeasure}}
    <br><h3>Поставщики:</h3>
    <ol>
        {{#each Some}}
        <li>{{this.DocumentVendor}}</li>
        {{/each}}
    </ol>

```



```
<p><a href="/index_count_vendor">Назад</a></p>
</body>
<html>
```