Федеральное государственное бюджетное образовательное учреждение  
высшего образования

РЫБИНСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ П.А. СОЛОВЬЁВА

Факультет радиоэлектроники и информатики

Кафедра математического и программного обеспечения электронных  
вычислительных средств

**КУРСОВОЙ ПРОЕКТ**

по дисциплине

«Базы данных»

на тему:

«Торговая фирма»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

|  |  |
| --- | --- |
| Студенты группы ИПБ-20 | Батурин К.А., Никанов П.А. |
| Руководитель | Шаров В.Г., Задорина Н.А. |

Рыбинск 2023

Оглавление

[1. Введение 4](#_Toc138931483)

[1.1. Актуальность разработки 4](#_Toc138931484)

[1.2. Цель работы и постановка задачи 4](#_Toc138931485)

[1.3. Технические и программные средства реализации 4](#_Toc138931486)

[2. Постановка задачи 6](#_Toc138931487)

[3. Описание предметной области 7](#_Toc138931488)

[3.1. Характеристики ООО «Торговая фирма» 7](#_Toc138931489)

[3.2. Общая характеристика объекта управления 8](#_Toc138931490)

[4. Инфологическая модель 12](#_Toc138931491)

[4.1. Сущности и их ключевые атрибуты 12](#_Toc138931492)

[4.2. Связи и бизнес-правила 12](#_Toc138931493)

[4.2.1. Связи 12](#_Toc138931494)

[4.2.2. Бизнес-правила 12](#_Toc138931495)

[4.3. Диаграмма ER-типа 12](#_Toc138931496)

[4.4. CASE-диаграмма 12](#_Toc138931497)

[5. Даталогическая модель 14](#_Toc138931498)

[5.1. Отношения базы данных 14](#_Toc138931499)

[5.2. Предварительные отношения базы данных 14](#_Toc138931500)

[6. Реализация базы данных 19](#_Toc138931501)

[6.1. Создание таблиц в базе данных 19](#_Toc138931502)

[Приложение А 24](#_Toc138931503)

[Приложение Б 25](#_Toc138931504)

[Приложение В 26](#_Toc138931505)

[Приложение Г 27](#_Toc138931506)

# 1. Введение

## 1.1. Актуальность разработки

Актуальность разработки системы «Моя торговля» заключается в автоматизации процессов складского учёта. Использование данной программы облегчит такие процессы, как:

- приёмка товаров на склад;

- заказ товаров у поставщиков;

- отпуск товаров со склада;

- получение справок о наличии товаров на складах;

- получение справок о активных и завершенных заказах.

## 1.2. Цель работы и постановка задачи

Создание системы, обеспечивающей предоставление информации о складах, товарах, в них содержащихся, о поставщиках и ими поставляемых товарах. Система должна поддерживать историю совершенных операций отгрузки и приёмки товаров по каждому складу для ведения учёта и выдачи справок.

## 1.3. Технические и программные средства реализации

Система «Моя торговля» представляет собой приложение на ЯП Java, использующим «динамические» запросы MySQL. Связь будет осуществляться благодаря JDBC.

JDBC – это платформенно независимый промышленный стандарт взаимодействия Java-приложений с реляционными базами данных.

Основная часть системы реализована на Java с помощью платформы JavaFX, позволяющей создавать приложения с насыщенным графическим интерфейсом. С его помощью реализована интерфейсная и логическая часть программы, обеспечивающая визуальное представление для работы с программой, и содержащую в себе основные инструменты для работы с базой данных. Клиент вызывает определенные функции API (Application Programming Interface) с параметрами в виде запросов (в нашем случае запросы, сформированные на языке SQL), движок MySQL обрабатывает и выполняет запросы и возвращает необходимые данные клиенту.

Приведем преимущества MySQL (а также соответствующего ему типа SQL) над другими СУБД:

* Открытый исходный код – распространяется бесплатно для домашнего применения;
* Простота – MySQL легко устанавливается, имеет понятный интерфейс, а разнообразие плагинов и дополнительных приложений упрощает работу с БД;
* Функционал – включает в себя практически весь необходимый набор инструментов, который может пригодиться при разработке любого проекта;
* Безопасность – многие системы безопасности уже встроены и работают по умолчанию;
* Масштабируемость – может использоваться в работе как с малым, так и с большим объемом данных;
* Скорость – является одной из самых быстрых среди имеющихся на современном рынке.

# 2. Постановка задачи

**Торговая фирма**

Торговая фирма занимается оптовой торговлей. Фирма имеет сеть поставщиков товаров. Поставщики находятся в разных городах. Каждый поставщик специализируется на поставке одного товара, склады фирмы позволяют разместить ограниченное количество каждого из товаров.

Товар на фирму может поступать по ее заявкам или по предложению поставщика без предварительного заказа. Отпуск товаров производится по заявкам.

Система должна обеспечивать выполнение следующих операций:

* прием предложения поставщика о возможности поставить товар в виде накладной;
* прием товара на склад или отказ в приеме;
* формирование заказа поставщику на поставку товара;
* отпуск товара или отказ;
* получение справок о сделанных заказах поставщикам, о предложениях поставщиков, о заявках на отпуск товаров;
* получение других справок.

# 3. Описание предметной области

## 3.1. Характеристики ООО «Торговая фирма»

ООО «Торговая фирма» – небольшое предприятие с численностью персонала около 60 человек. Предприятие занимается оптовой торговлей.

Для обеспечения производственной программы соответствующими материальными ресурсами на предприятии ООО «Торговая фирма» имеется несколько складов в разных городах.

На складах товарно-материальные ценности размещают по секциям, а внутри них – по группам на стеллажах, что обеспечивает их быструю приёмку, отпуск и контроль соответствия фактического наличия.

На каждый номенклатурный номер товаров кладовщик заполняет Товарный ярлык (форма № ТОРГ-11, Приложение А) и прикрепляет его к месту хранения товаров. В ярлыке указывается наименование товара, его код, описание единицы измерения товара (код по ОКЕИ и полное название), артикул, количество товаров в ячейке и учётная стоимость одной единицы.

Учёт движения и остатка товаров осуществляют в Карточках учёта материалов. На каждый номенклатурный номер открывают отдельную карточку. В карточках указывают наименование товара, номенклатурный номер, размер, место хранения, единицу измерения, цену закупки одного товара, балансовый счёт и другие данные.

Запись в карточках кладовщик делает на основе первичных документов в день совершения операций. После каждой записи по поступлению или отгрузке товара в его карточку вносится изменённый остаток.

Товары поступают в фирму от поставщиков путём приобретения за плату. На поступающие от поставщиков товары, на основании сопроводительных документов (накладная, счёт-фактура), на складе выписывают Приходный ордер (форма № М-4, Приложение Б). Если в течение дня от одного и того же поставщика приходит несколько поставок, каждая заносится в отдельный документ. Сведения из приходного ордера кладовщики заносят в Карточку учёта материалов.

Для учёта отпуска товаров сторонним организациям применяется Накладная на отпуск материалов на сторону (форма № М-15, Приложение В). Она оформляется в двух экземплярах, один из которых отдаётся получателю, а второй остаётся на складе.

Списание товаров может осуществляться в следующих случаях:

* в связи с продажей, осуществлённой на возмездном основании контрагенту по ранее заключённому договору;
* в связи с привлечением посредника;
* в связи с форс-мажорными обстоятельствами.

Обобщение и группировка информации о движении товаров осуществляется в Отчётах о движении товарно-материальных ценностей в местах хранения (форма № МХ-20, Приложение Г) каждый квартал, отдельно по каждому складу. Отчёты составляются отдельно по приходу и расходу материалов, в них подсчитываются итоги оборотов за квартал (отчётный период) в разрезе групп товаров по синтетическим счетам и субсчетам, по каждому складу.

Ежеквартально составляется сводная ведомость движения товаров, в которую переносятся итоги из отчётов движения товаров по складам. В сводных ведомостях движения товаров приводятся также данные об остатках товарно-материальных ценностей на начало и конец квартала (отчётного периода) в разрезе групп товаров по соответствующим синтетическим счетам и субсчетам бухгалтерского учёта.

Данные сводной ведомости движения товаров и накопительных отчётов ежеквартально сверяются с данными аналитического учёта.

## 3.2. Общая характеристика объекта управления

При приёмке товаров на склад от поставщиков, входными документами являются накладная, или счёт-фактура. Если фактические данные соответствуют данным в документах, то на складе выписывают Приходный ордер (форма № М-4, Приложение Б) и делают запись в Карточку учёта материалов.

При отпуске товаров со склада сторонним организациям, выходным документом является Накладная на отпуск материалов на сторону (форма № М-15, Приложение В).

Основными входными документами являются:

* накладная;
* счёт-фактура.

Выходными документами являются:

* товарный ярлык;
* приходный ордер;
* накладная на отпуск товаров на сторону;
* отчёт о движении товарно-материальных ценностей в местах хранения.

Образцы бланков выходных документов приведены в приложениях.

Информация может представляться как в виде документов в письменной форме, так и в электронном виде (на магнитных носителях или по каналам связи) при наличии гарантий их достоверности и защиты от несанкционированного доступа и искажений.

Ввод данных может быть разрешен только после проверки наличия прав доступа лица к изменению базы данных. Только в этом случае может быть обеспечена защита информации. Поскольку ввод информации могут осуществлять различные категории пользователей, у каждой категории должен быть доступ только к части базы данных, и соответственно внесение изменений может быть разрешено только для определенной части базы данных.

Ввод информации о движении материалов выполняется кладовщиком на складе. Здесь выполняется работа с документами. Если документы предоставляются в бумажном виде, то выполняется ручной ввод информации оператором. При ручном переводе данных на машинный носитель информации возможны ошибки вследствие разнообразных причин:

* ошибки при записи результатов в промежуточные документы;
* ошибки при считывании данных из промежуточных документов при вводе с клавиатуры;
* преднамеренное искажение данных.

Любая ошибка может привести либо к ошибке в расчетах, либо к более серьезным последствиям.

**Список ограничений:**

* название продукта не должно превышать 128 символов;
* описание продукта не должно превышать 512 символов;
* номенклатурный номер и артикул товара не должен превышать 30 символов;
* название склада не должно превышать 128 символов;
* название города, в котором расположен склад, и улицы не должно превышать 64 символов;
* наименование фирмы поставщика не должно превышать 128 символов;
* название города, в котором расположен офис поставщика, не должно превышать 32 символов;
* полный адрес поставщика не должен превышать 128 символов;
* данные платёжного счёта поставщика не могут превышать 20 символов;
* телефонный номер поставщика не может превышать 20 символов;
* дополнительные сведения о банке поставщика не могут превышать 64 символов;
* поставщик обязан поставлять какой-либо товар;
* наименование типа упаковки хранимого на складе товара не должно превышать 30 символов;
* каждое из перечисленных: фамилия, имя и отчество пользователя системы - не могут превышать 45 символов;
* логин пользователя системы не может превышать 64 символов.

Программа разрабатывается для следующих типов пользователей:

* кладовщик;
* поставщик-заказчик;
* администратор.

Программа для кладовщика будет обеспечивать выполнение следующих функций:

* приёмка и отгрузка товаров со склада;
* заказ товаров от поставщиков на склад;
* формирование и просмотр отчётов по продажам товаров на складе, общей выручке и общих затратах за выбранный период времени;
* формирование и запись отчёта о движении товарно-материальных ценностей на данном складе;
* формирование и запись документов о приёмке и отгрузке товаров;
* формирование и запись товарного ярлыка;
* удовлетворение и отклонение запроса на поставку товара со склада.

Программа для поставщика-заказчика будет обеспечивать выполнение следующих функций:

* регистрация новой фирмы поставщика;
* регистрация или выбор существующего товара для поставок;
* регистрация предложения на поставку своего товара определённому складу;
* регистрация заказов имеющихся на складах товаров;
* удовлетворение и отклонение запроса на поставку товара.

Программа для администратора будет обеспечивать выполнение следующих функций:

- регистрация нового поставщики-заказчика или кладовщика;

- просмотр списка всех пользователей системы.

# 4. Инфологическая модель

## 4.1. Сущности и их ключевые атрибуты

* Товар(ID товара, Название товара);
* Поставщик(ID поставщика, ID товара, Название поставщика, Город поставщика);
* Склад(ID склада, Название склада).

## 4.2. Связи и бизнес-правила

### 4.2.1. Связи

* Поставщик поставляет товар
* Склад хранит товары

### 4.2.2. Бизнес-правила

* Товар может храниться на нескольких складах.
* Склад может хранить несколько типов товара.
* Товар может не храниться ни на одном складе.
* Склад может не хранить ни один тип товара.
* Товар может поставляться несколькими поставщиками.
* Поставщик может поставлять только один тип товара.
* Товар может не поставляться поставщиком.
* Поставщик обязан поставлять товар.

## 4.3. Диаграмма ER-типа

ER-диаграмма представлена в нотации Чена на рисунке 1.

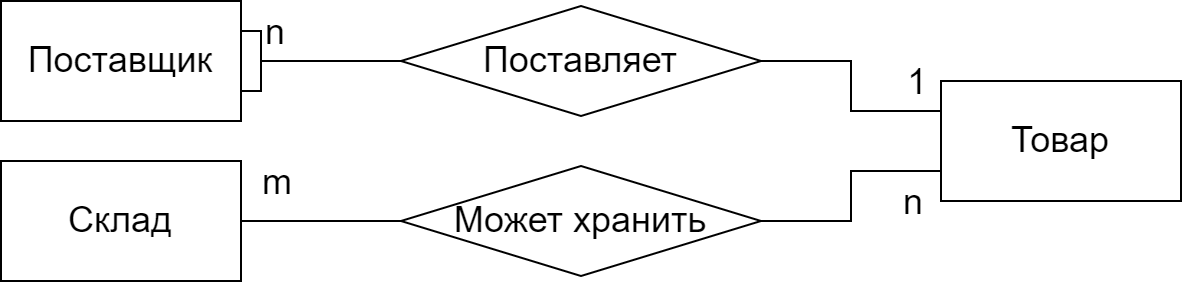


Рисунок 1 – ER-диаграмма в нотации Чена

## 4.4. CASE-диаграмма

CASE-диаграмма, полученная в результате обратного инжиниринга в MySQL Workbench представлена на рисунке 2.

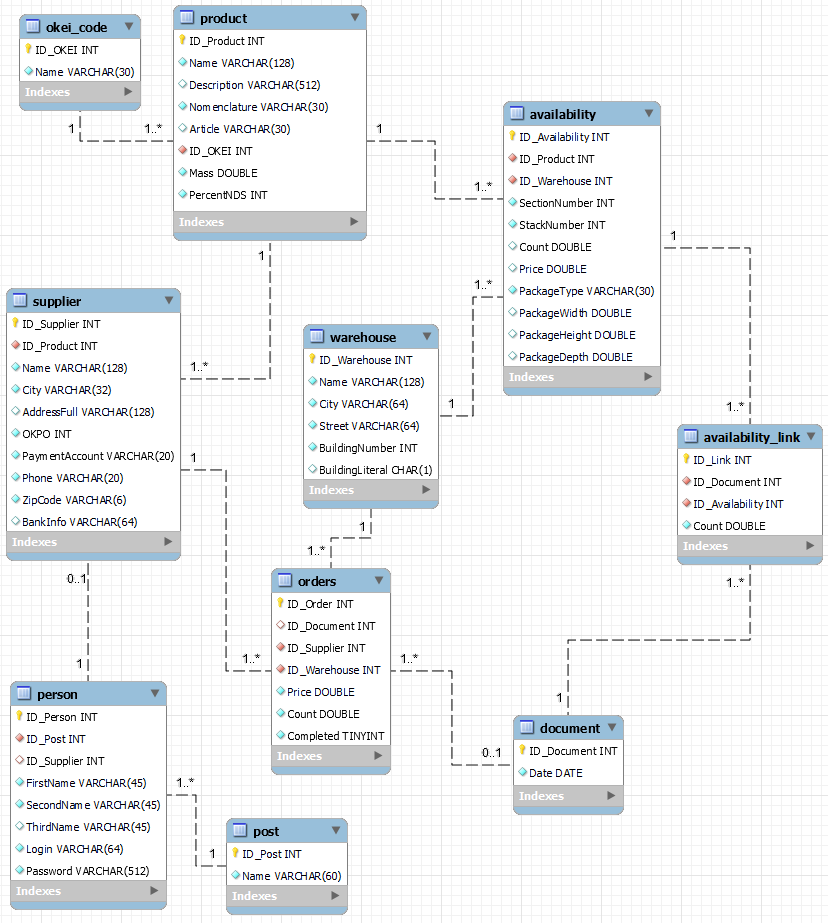


Рисунок 2 – CASE-диаграмма базы данных

# 5. Даталогическая модель

## 5.1. Отношения базы данных

* ОКЕИ\_Код(Ид, Наименование);
* Товар(Ид, Наименование, Описание, Номенклатура, Артикул, ИдОКЕИ, Масса, ПроцентНДС);
* ТоварНаСкладе(Ид, ИдТовара, ИдСклада, Секция, Стеллаж, Количество, Цена, ТипУпаковки, Ширина, Высота, Глубина);
* Склад(Ид, Название, Город, Улица, НомерЗдания, ЛитераЗдания);
* Поставщик(Ид, ИдТовара, Название, Город, ПолныйАдрес, КодОКПО, ПлатёжныйСчёт, Телефон, ПочтовыйИндекс, БанковскаяИнформация);
* Должность(Ид, Название);
* Пользователь(Ид, ИдДолжности, ИдКомпании, Имя, Фамилия, Отчество, Логин, ХешПароля);
* Заказы(Ид, ИдДокумента, ИдПоставщика, ИдСклада, Цена, Количество, СтатусЗаказа);
* Документ(Ид, ДатаСоставления);
* СвязьДокументаСТоваромНаСкладе(Ид, ИдДокумента, ИдТовараНаСкладе, Количество);

## 5.2. Предварительные отношения базы данных

Таблица – описание отношений в БД

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование информационного объекта** | **Наименование атрибута** | **Имя атрибута в ИЛМ** | **Тип** | **Длина** | **Ключ** |
| ОКЕИ\_Код | Ид | ID\_OKEI | INT | - | + |
| Наименование | Name | VARCHAR | 30 | - |
| Товар | Ид | ID\_Product | INT | - | + |
| Наименование | Name | VARCHAR | 128 | - |
| Описание | Description | VARCHAR | 512 | - |
| Номенклатура | Nomenclature | VARCHAR | 30 | - |
| Артикул | Article | VARCHAR | 30 | - |
| ИдОКЕИ | ID\_OKEI | INT | - | - |
| Масса | Mass | DOUBLE | - | - |
| ПроцентНДС | PercentNDS | INT | - | - |
| ТоварНаСкладе | Ид | ID\_Availability | INT | - | + |
| ИдТовара | ID\_Product | INT | - | - |
| ИдСклада | ID\_Warehouse | INT | - | - |
| Секция | SectionNumber | INT | - | - |
| Стеллаж | StackNumber | INT | - | - |
| Количество | Count | DOUBLE | - | - |
| Цена | Price | DOUBLE | - | - |
| ТипУпаковки | PackageType | VARCHAR | 30 | - |
| Ширина | PackageWidth | DOUBLE | - | - |
| Высота | PackageHeight | DOUBLE | - | - |
| Глубина | PackageDepth | DOUBLE | - | - |
| Склад | Ид | ID\_Warehouse | INT | - | + |
| Название | Name | VARCHAR | 128 | - |
| Город | City | VARCHAR | 64 | - |
| Улица | Street | VARCHAR | 64 | - |
| НомерЗдания | BuildingNumber | INT | - | - |
| ЛитераЗдания | BuildingLiteral | CHAR | 1 | - |
| Поставщик | Ид | ID\_Supplier | INT | - | + |
| ИдТовара | ID\_Product | INT | - | - |
| Название | Name | VARCHAR | 128 | - |
| Город | City | VARCHAR | 32 | - |
| ПолныйАдрес | AddressFull | VARCHAR | 128 | - |
| КодОКПО | OKPO | INT | - | - |
| ПлатёжныйСчёт | PaymentAccount | VARCHAR | 20 | - |
| Телефон | Phone | VARCHAR | 20 | - |
| ПочтовыйИндекс | ZipCode | VARCHAR | 6 | - |
| БанковскаяИнформация | BankInfo | VARCHAR | 64 | - |
| Должность | Ид | ID\_Post | INT | - | + |
| Название | Name | VARCHAR | 60 | - |
| Пользователь | Ид | ID\_Person | INT | - | + |
| ИдДолжности | ID\_Post | INT | - | - |
| ИдКомпании | ID\_Supplier | INT | - | - |
| Имя | FirstName | VARCHAR | 45 | - |
| Фамилия | SecondName | VARCHAR | 45 | - |
| Отчество | ThirdName | VARCHAR | 45 | - |
| Логин | Login | VARCHAR | 64 | - |
| ХешПароля | Password | VARCHAR | 512 | - |
| Заказы | Ид | ID\_Order | INT | - | + |
| ИдДокумента | ID\_Document | INT | - | - |
| ИдПоставщика | ID\_Supplier | INT | - | - |
| ИдСклада | ID\_Warehouse | INT | - | - |
| Цена | Price | DOUBLE | - | - |
| Количество | Count | DOUBLE | - | - |
| СтатусЗаказа | Completed | TINYINT | - | - |
| Документ | Ид | ID\_Document | INT | - | + |
| ДатаСоставления | Date | DATE | - | - |
| СвязьДокументаСТоваромНаСкладе | Ид | ID\_Link | INT | - | + |
| ИдДокумента | ID\_Document | INT | - | - |
| ИдТовараНаСкладе | ID\_Availability | INT | - | - |
| Количество | Count | DOUBLE | - | - |

Сводная таблица отношений представлена в таблице 2.

|  |  |
| --- | --- |
| **Имя таблицы** | **Описание** |
| ОКЕИ\_Код | Выдержка из общероссийского классификатора единиц измерения |
| Товар | Описание товара |
| ТоварНаСкладе | Таблица хранения товара на складе |
| Склад | Описание склада |
| Поставщик | Описание поставщика |
| Должность | Описание должностей |
| Пользователь | Описание пользователей |
| Заказы | Таблица хранения заказа и его статуса |
| Документ | Таблица хранения даты совершения заказа |
| СвязьДокументаСТоваромНаСкладе | Таблица, связывающая заказ с изменениями хранения товара на складе |

# 6. Реализация базы данных

Работа с базой данных осуществляется с помощью СУБД MySQL совместно с используемым в ней видом SQL от Oracle. В качестве протокола обмена используются вызовы функций (API). Данные обработчики всегда вызывают функции, которые формируют SQL запросы к СУБД и возвращают ответы в виде списка объектов конкретной структуры данных.

СУБД использует триггеры и хранимые процедуры. Все запросы формируются динамически в приложении.

Используемые триггеры:

Триггер удаления продукта

**delimiter** //  
**CREATE** **TRIGGER** DelProduct **BEFORE** **DELETE** **ON** product  
**FOR** **EACH** **ROW  
BEGIN** **DELETE** **FROM** availability  
 **WHERE** availability.ID\_Product = **OLD**.ID\_Product;  
 **DELETE** **from** supplier  
 **WHERE** supplier.ID\_Product = **OLD**.ID\_Product;  
**END**;//  
**delimiter**;

Триггер удаления поставщика

**delimiter** //  
**CREATE** **TRIGGER** DelSupplier **BEFORE** **DELETE** **ON** supplier  
**FOR** **EACH** **ROW  
BEGIN** **DELETE** **FROM** orders  
 **WHERE** orders.ID\_Supplier = **OLD**.ID\_Supplier;  
 **DELETE** **FROM** person  
 **WHERE** person.ID\_Supplier = **OLD**.ID\_Supplier;  
**END**;//  
**delimiter**;

Триггер удаления складской записи

**delimiter** //  
**CREATE** **TRIGGER** DelAvailability **BEFORE** **DELETE** **ON** availability  
**FOR** **EACH** **ROW  
BEGIN** **DELETE** **FROM** availability\_link  
 **WHERE** availability\_link.ID\_Availability = **OLD**.ID\_Availability;  
**END**;//  
**delimiter**;

## 6.1. Создание таблиц в базе данных

Таблицы в базе данных создавались при помощи графического интерфейса приложения MySQL Workbench. Ниже вы можете увидеть результат прямого инжиниринга базы данных.

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

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

-- Schema course\_work

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

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

-- Schema course\_work

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

CREATE SCHEMA IF NOT EXISTS `course\_work` DEFAULT CHARACTER SET cp1251 ;

USE `course\_work` ;

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

-- Table `course\_work`.`okei\_code`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`okei\_code` (

`ID\_OKEI` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(30) NOT NULL,

PRIMARY KEY (`ID\_OKEI`),

UNIQUE INDEX `ID\_OKEI\_UNIQUE` (`ID\_OKEI` ASC) VISIBLE)

ENGINE = InnoDB

AUTO\_INCREMENT = 9992

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`product`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`product` (

`ID\_Product` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(128) NOT NULL,

`Description` VARCHAR(512) NULL DEFAULT NULL,

`Nomenclature` VARCHAR(30) NOT NULL,

`Article` VARCHAR(30) NULL DEFAULT NULL,

`ID\_OKEI` INT UNSIGNED NOT NULL,

`Mass` DOUBLE NOT NULL,

`PercentNDS` INT UNSIGNED NOT NULL,

`StorageConditions` VARCHAR(1024) NULL DEFAULT NULL,

`Deleted` TINYINT NOT NULL DEFAULT '0',

PRIMARY KEY (`ID\_Product`),

UNIQUE INDEX `ID\_Product\_UNIQUE` (`ID\_Product` ASC) VISIBLE,

INDEX `ID\_OKEI\_idx` (`ID\_OKEI` ASC) VISIBLE,

CONSTRAINT `ID\_OkeiP`

FOREIGN KEY (`ID\_OKEI`)

REFERENCES `course\_work`.`okei\_code` (`ID\_OKEI`))

ENGINE = InnoDB

AUTO\_INCREMENT = 6

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`warehouse`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`warehouse` (

`ID\_Warehouse` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(128) NOT NULL,

`City` VARCHAR(64) NOT NULL,

`Street` VARCHAR(64) NOT NULL,

`BuildingNumber` INT UNSIGNED NOT NULL,

`BuildingLiteral` CHAR(1) NULL DEFAULT NULL,

PRIMARY KEY (`ID\_Warehouse`),

UNIQUE INDEX `ID\_Warehouse\_UNIQUE` (`ID\_Warehouse` ASC) VISIBLE)

ENGINE = InnoDB

AUTO\_INCREMENT = 5

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`availability`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`availability` (

`ID\_Availability` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`ID\_Product` INT UNSIGNED NOT NULL,

`ID\_Warehouse` INT UNSIGNED NOT NULL,

`SectionNumber` INT UNSIGNED NOT NULL,

`StackNumber` INT UNSIGNED NOT NULL,

`Count` DOUBLE NULL DEFAULT '0',

`Price` DOUBLE NULL DEFAULT '0',

`PackageType` VARCHAR(30) NOT NULL,

`PackageWidth` DOUBLE NULL DEFAULT NULL,

`PackageHeight` DOUBLE NULL DEFAULT NULL,

`PackageDepth` DOUBLE NULL DEFAULT NULL,

PRIMARY KEY (`ID\_Availability`),

UNIQUE INDEX `ID\_Availability\_UNIQUE` (`ID\_Availability` ASC) VISIBLE,

INDEX `ID\_Product\_idx` (`ID\_Product` ASC) VISIBLE,

INDEX `ID\_Warehouse\_idx` (`ID\_Warehouse` ASC) VISIBLE,

CONSTRAINT `ID\_ProductA`

FOREIGN KEY (`ID\_Product`)

REFERENCES `course\_work`.`product` (`ID\_Product`),

CONSTRAINT `ID\_WarehouseA`

FOREIGN KEY (`ID\_Warehouse`)

REFERENCES `course\_work`.`warehouse` (`ID\_Warehouse`))

ENGINE = InnoDB

AUTO\_INCREMENT = 25

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`document`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`document` (

`ID\_Document` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`Date` DATE NOT NULL,

PRIMARY KEY (`ID\_Document`),

UNIQUE INDEX `ID\_Document\_UNIQUE` (`ID\_Document` ASC) VISIBLE)

ENGINE = InnoDB

AUTO\_INCREMENT = 29

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`availability\_link`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`availability\_link` (

`ID\_Link` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`ID\_Document` INT UNSIGNED NOT NULL,

`ID\_Availability` INT UNSIGNED NOT NULL,

`Count` DOUBLE NOT NULL,

PRIMARY KEY (`ID\_Link`),

UNIQUE INDEX `ID\_Link\_UNIQUE` (`ID\_Link` ASC) VISIBLE,

INDEX `fk\_document\_has\_availability\_availability1\_idx` (`ID\_Availability` ASC) VISIBLE,

INDEX `fk\_document\_has\_availability\_document1\_idx` (`ID\_Document` ASC) VISIBLE,

CONSTRAINT `ID\_AvailabilityDHA`

FOREIGN KEY (`ID\_Availability`)

REFERENCES `course\_work`.`availability` (`ID\_Availability`),

CONSTRAINT `ID\_DocumentDHA`

FOREIGN KEY (`ID\_Document`)

REFERENCES `course\_work`.`document` (`ID\_Document`))

ENGINE = InnoDB

AUTO\_INCREMENT = 29

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`supplier`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`supplier` (

`ID\_Supplier` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`ID\_Product` INT UNSIGNED NOT NULL,

`Name` VARCHAR(128) NOT NULL,

`City` VARCHAR(32) NOT NULL,

`AddressFull` VARCHAR(128) NULL DEFAULT NULL,

`OKPO` INT UNSIGNED NOT NULL,

`PaymentAccount` VARCHAR(20) NOT NULL,

`Phone` VARCHAR(20) NOT NULL,

`ZipCode` VARCHAR(6) NOT NULL,

`BankInfo` VARCHAR(64) NULL DEFAULT NULL,

PRIMARY KEY (`ID\_Supplier`),

UNIQUE INDEX `ID\_Supplier\_UNIQUE` (`ID\_Supplier` ASC) VISIBLE,

INDEX `ID\_Product\_idx` (`ID\_Product` ASC) VISIBLE,

CONSTRAINT `ID\_ProductS`

FOREIGN KEY (`ID\_Product`)

REFERENCES `course\_work`.`product` (`ID\_Product`))

ENGINE = InnoDB

AUTO\_INCREMENT = 6

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`orders`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`orders` (

`ID\_Order` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`ID\_Document` INT UNSIGNED NULL DEFAULT NULL,

`ID\_Supplier` INT UNSIGNED NOT NULL,

`ID\_Warehouse` INT UNSIGNED NOT NULL,

`Price` DOUBLE NOT NULL,

`Count` DOUBLE NOT NULL,

`Completed` TINYINT NOT NULL DEFAULT '0',

PRIMARY KEY (`ID\_Order`),

UNIQUE INDEX `ID\_Order\_UNIQUE` (`ID\_Order` ASC) VISIBLE,

INDEX `fk\_document\_has\_supplier\_supplier1\_idx` (`ID\_Supplier` ASC) VISIBLE,

INDEX `fk\_document\_has\_supplier\_document1\_idx` (`ID\_Document` ASC) VISIBLE,

INDEX `fk\_order\_from\_warehouse\_warehouse1\_idx` (`ID\_Warehouse` ASC) VISIBLE,

CONSTRAINT `fk\_order\_from\_warehouse\_warehouse1`

FOREIGN KEY (`ID\_Warehouse`)

REFERENCES `course\_work`.`warehouse` (`ID\_Warehouse`),

CONSTRAINT `ID\_DocumentDHS`

FOREIGN KEY (`ID\_Document`)

REFERENCES `course\_work`.`document` (`ID\_Document`),

CONSTRAINT `ID\_SupplierDHS`

FOREIGN KEY (`ID\_Supplier`)

REFERENCES `course\_work`.`supplier` (`ID\_Supplier`))

ENGINE = InnoDB

AUTO\_INCREMENT = 39

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`post`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`post` (

`ID\_Post` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(60) NOT NULL,

PRIMARY KEY (`ID\_Post`),

UNIQUE INDEX `idroles\_UNIQUE` (`ID\_Post` ASC) VISIBLE)

ENGINE = InnoDB

AUTO\_INCREMENT = 4

DEFAULT CHARACTER SET = cp1251;

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

-- Table `course\_work`.`person`

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

CREATE TABLE IF NOT EXISTS `course\_work`.`person` (

`ID\_Person` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`ID\_Post` INT UNSIGNED NOT NULL,

`ID\_Supplier` INT UNSIGNED NULL DEFAULT NULL,

`FirstName` VARCHAR(45) NOT NULL,

`SecondName` VARCHAR(45) NOT NULL,

`ThirdName` VARCHAR(45) NULL DEFAULT NULL,

`Login` VARCHAR(64) NOT NULL,

`Password` VARCHAR(512) NOT NULL,

PRIMARY KEY (`ID\_Person`),

UNIQUE INDEX `ID\_Person\_UNIQUE` (`ID\_Person` ASC) VISIBLE,

INDEX `fk\_person\_roles1\_idx` (`ID\_Post` ASC) VISIBLE,

INDEX `fk\_person\_supplier1\_idx` (`ID\_Supplier` ASC) VISIBLE,

CONSTRAINT `fk\_person\_supplier1`

FOREIGN KEY (`ID\_Supplier`)

REFERENCES `course\_work`.`supplier` (`ID\_Supplier`),

CONSTRAINT `ID\_PostP`

FOREIGN KEY (`ID\_Post`)

REFERENCES `course\_work`.`post` (`ID\_Post`))

ENGINE = InnoDB

AUTO\_INCREMENT = 10

DEFAULT CHARACTER SET = cp1251;

USE `course\_work`;

DELIMITER $$

USE `course\_work`$$

CREATE

DEFINER=`root`@`localhost`

TRIGGER `course\_work`.`DelProduct`

BEFORE DELETE ON `course\_work`.`product`

FOR EACH ROW

BEGIN

Delete from availability

where availability.ID\_Product = old.ID\_Product;

delete from supplier

where supplier.ID\_Product=old.ID\_Product;

END$$

USE `course\_work`$$

CREATE

DEFINER=`root`@`localhost`

TRIGGER `course\_work`.`DelAvailability`

BEFORE DELETE ON `course\_work`.`availability`

FOR EACH ROW

BEGIN

delete from availability\_link

where availability\_link.ID\_Availability = old.ID\_Availability;

END$$

USE `course\_work`$$

CREATE

DEFINER=`root`@`localhost`

TRIGGER `course\_work`.`DelSupplier`

BEFORE DELETE ON `course\_work`.`supplier`

FOR EACH ROW

BEGIN

delete from orders

where orders.ID\_Supplier = old.ID\_Supplier;

delete from person

where person.ID\_Supplier=old.ID\_Supplier;

END$$

DELIMITER ;

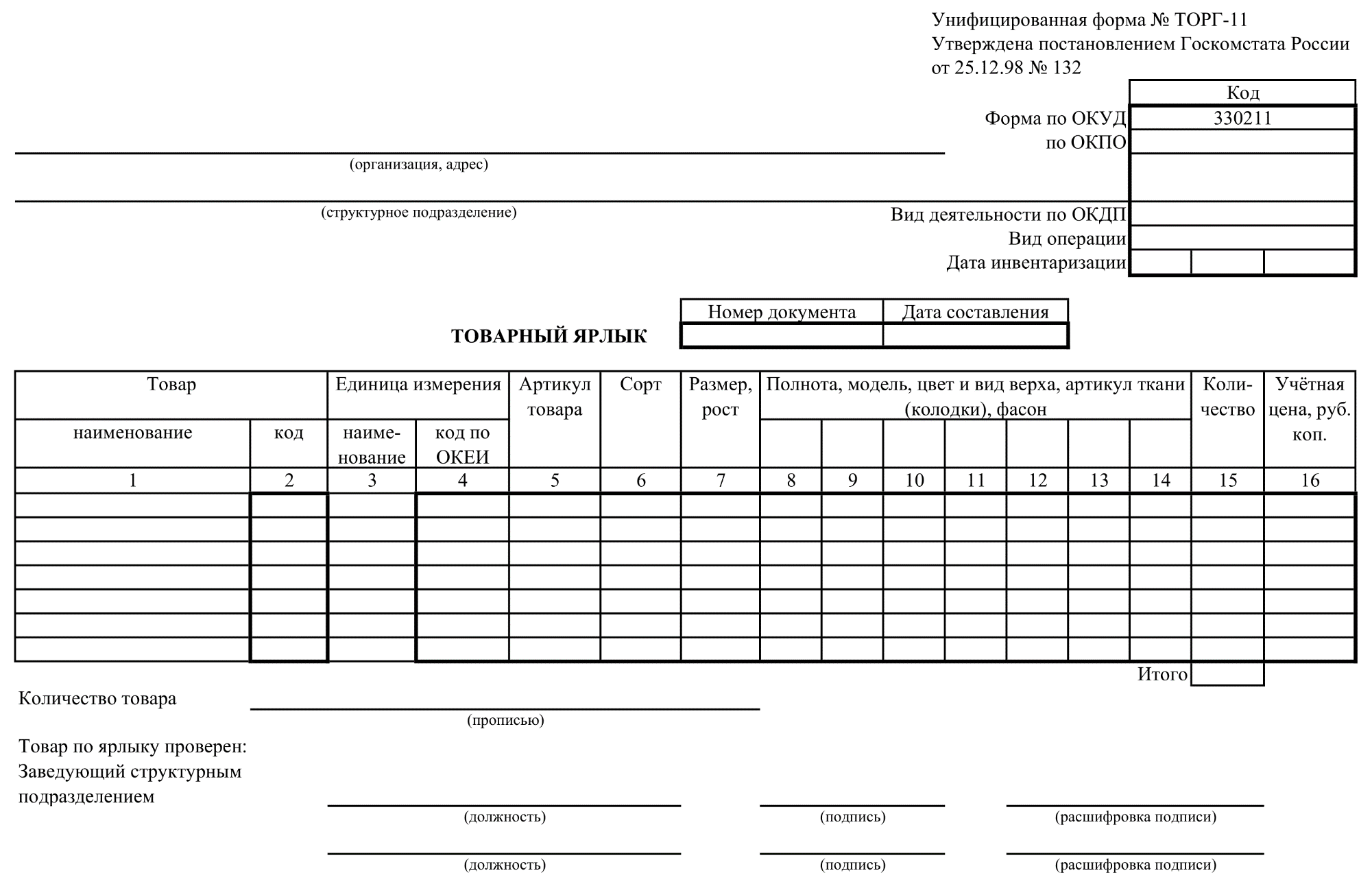
SET SQL\_MODE=@OLD\_SQL\_MODE;

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

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

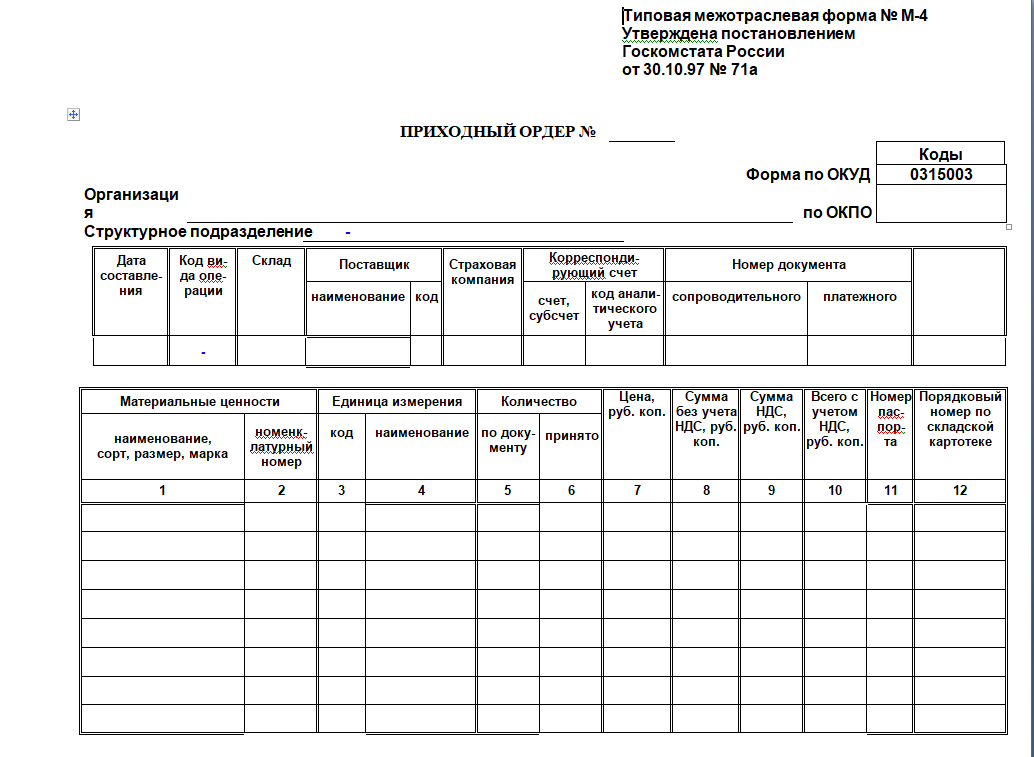
# Приложение А

Товарный ярлык, форма № ТОРГ-11



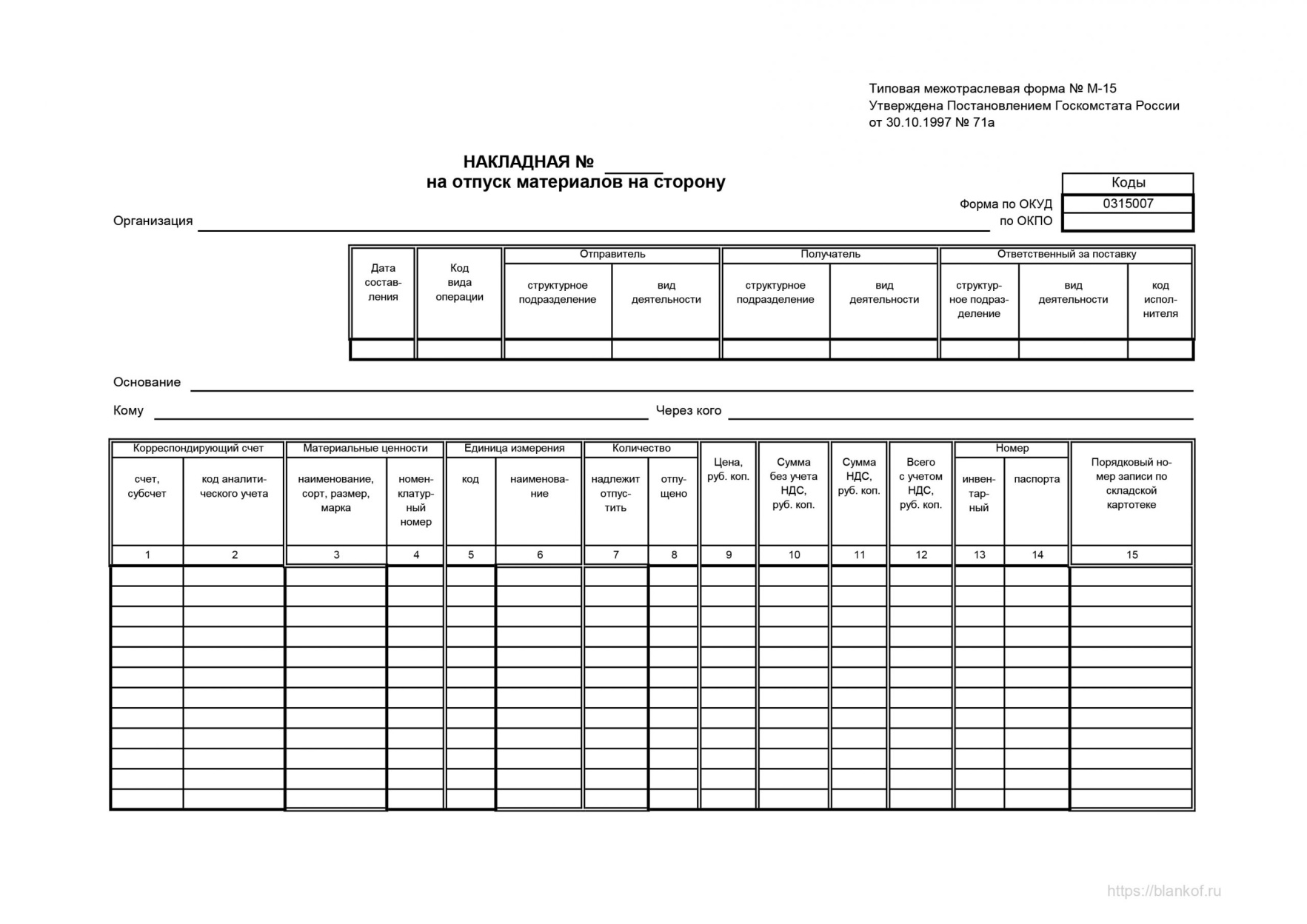
# Приложение Б

Приходной ордер, форма № М-4



# Приложение В

Накладная на отпуск товаров на сторону, форма № М-15



# Приложение Г

Отчёт о движении товарно-материальных ценностей в местах хранения, форма № МХ-20

