Министерство науки и высшего образования Российской федерации

Федерального государственного бюджетного образовательного учреждения высшего образования

Московский государственный технический университет имени Н. Э. Баумана

(Национальный исследовательский университет)

Московский техникум космического приборостроения

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по теме: «БАЗА ДАННЫХ СУДОХОДНОЙ КОМПАНИИ»

Специальность: 09.02.07 Информационные системы и программирование – квалификация «Программист»

Группа ТИП-41

|  |  |
| --- | --- |
| Проверил  Разработал | Филиппов В.В.  Осипова Н.М.  Митрошенкова Е.А.  Симонян П.Р. |

Москва 2021

Содержание

[1 Анализ предметной области 2](#_Toc108147722)

[2 Проектирование базы данных 5](#_Toc108147723)

[3 Проектирование шаблона выходного документа 12](#_Toc108147724)

# 1 Анализ предметной области

Судоходная компания – Организация, владеющая множеством судов различных классов и занимающаяся грузоперевозками.

Каждое судно может перемещаться между разными городами в рамках одного маршрута, отгружать и загружать товары из разных портов. Разрабатываемое ПО должно дать возможность вести учет информации о грузоперевозках и составлять отчет о них в виде текстового документа.

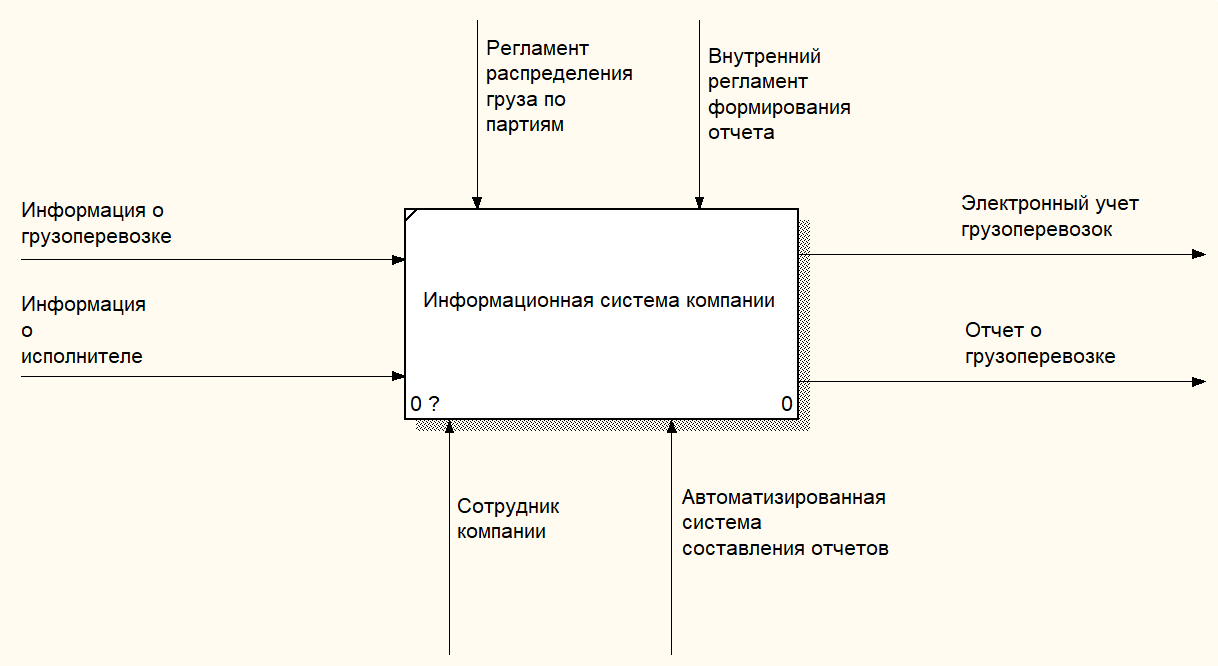
Процесс составления выходного документа и ведения учета деятельности организации представлен на рисунке 1.1. 

Рисунок 1.1 – Контекстная диаграмма

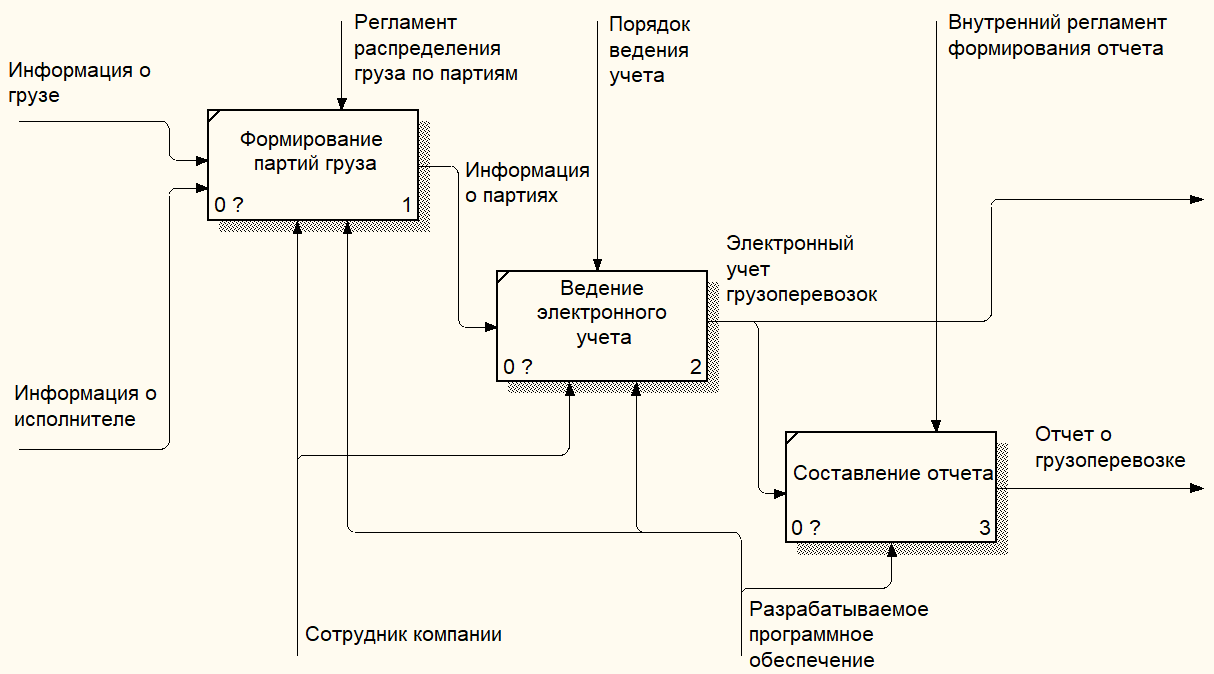
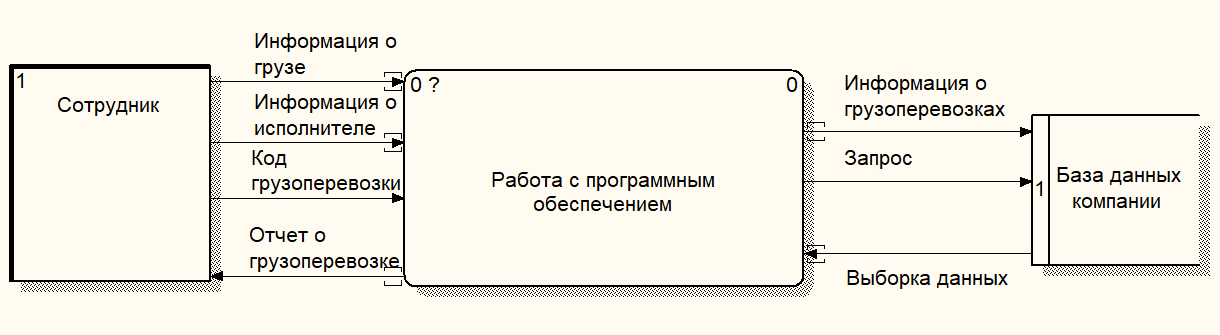
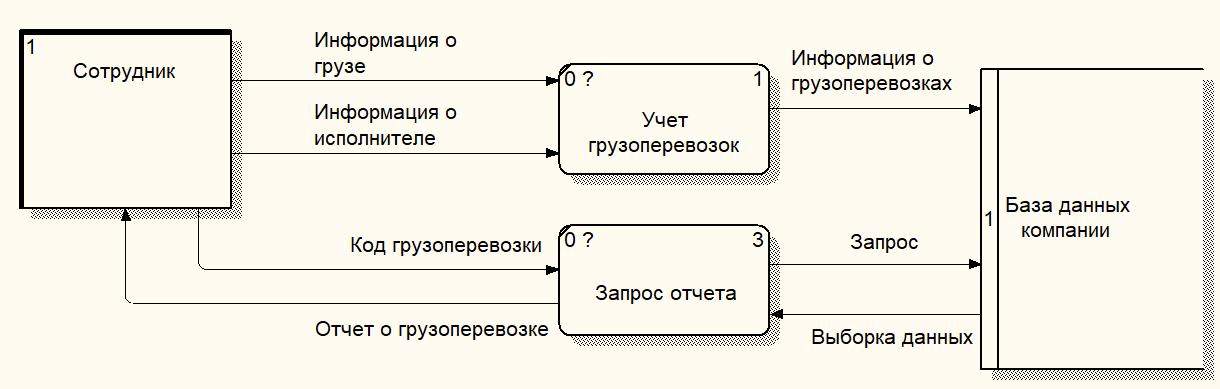
На рисунке 1.2 представлена развертка контекстной диаграммы.

Рисунок 1.2 – Развернутая контекстная диаграмма

На вход системы подается информация о грузе, включая информацию об отправителе, получателе, месте транспортировки груза и т.п. Во входную информацию об исполнителе включаются данные о судне и капитане, совершившем перевозку. В приложении груз объединяется в партии по общим признакам из соображений оптимизации логистики. Вся информация заносится в базу данных. По требованию приложение извлекает эти данные из базы данных и формирует отчет о грузоперевозке.

На рисунках 1.3 и 1.4 изображена диаграмма потоков данных в свернутом и развернутом виде соответственно.

 Рисунок 1.3 – Диаграмма потоков данных

Рисунок 1.4 – Развернутая диаграмма потоков данных

Требования к системе:

1. Программное обеспечение должно обеспечивать безопасный доступ к данным компании, возможность редактировать эти данные, автоматически проверяя их на соответствие разработанным критериям.
2. Приложение должно быть интуитивно простым, редактирование и доступ к данным должно быть наглядным.
3. Приложение должно обладать безопасным входом с паролем и некоторой информацией о пользователе, редактирующем информацию.
4. Система должна составлять отчет, содержащий необходимую информацию о грузоперевозках.

Функции системы:

1. Подключение к базе данных.
2. Вывод требуемой информации о грузоперевозках.
3. Сохранение внесенных изменений в базе данных.
4. Удаление указанных записей из базы данных по требованию.
5. Формирование отчета об указанной поездке, сохранение созданного файла формата Word (.docx) в системе пользователя.

# 2 Проектирование базы данных

Концептуальная схема разработанной базы данных изображена на рисунке 2.1. Она отражает главные сущности базы данных.

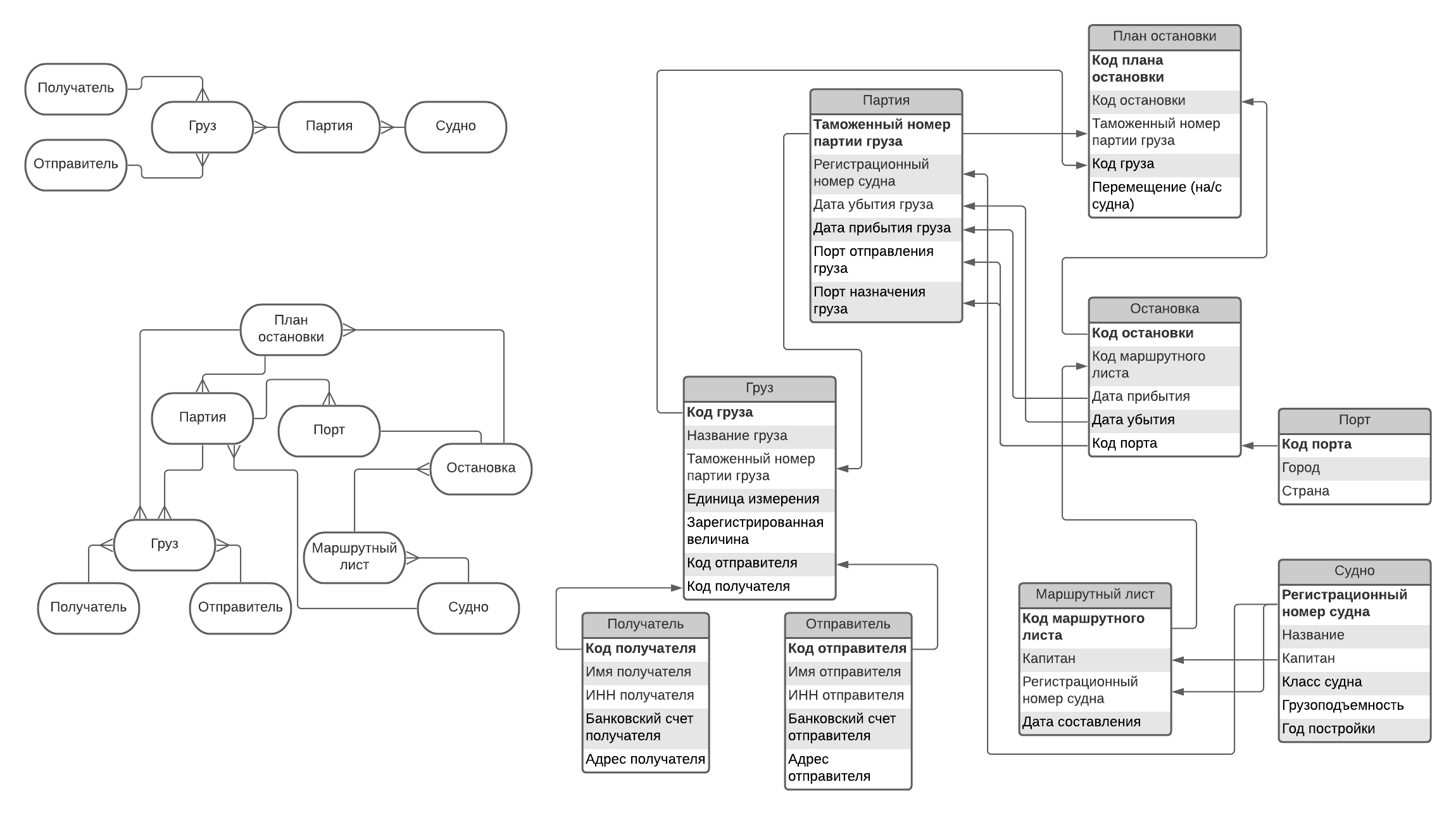


Рисунок 2.1 – Концептуальная схема

На развернутой концептуальной схеме (рис. 2.2) добавлены сущности «Остановка», «Маршрутный лист» и «Порт». Они необходимы для отображения перемещения груза и судна в рамках одного маршрута.

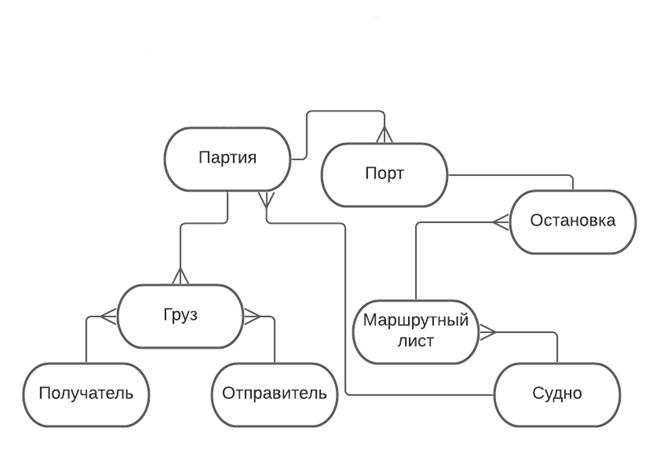


Рисунок 2.2 – Развернутая концептуальная схема

В течение одной поездки судно может совершить несколько остановок с дополнительными загрузками

Далее была разработана логическая схема базы данных. Первичные ключи сущностей изображены жирным шрифтом. Стрелки направлены от главной сущности к зависимой.

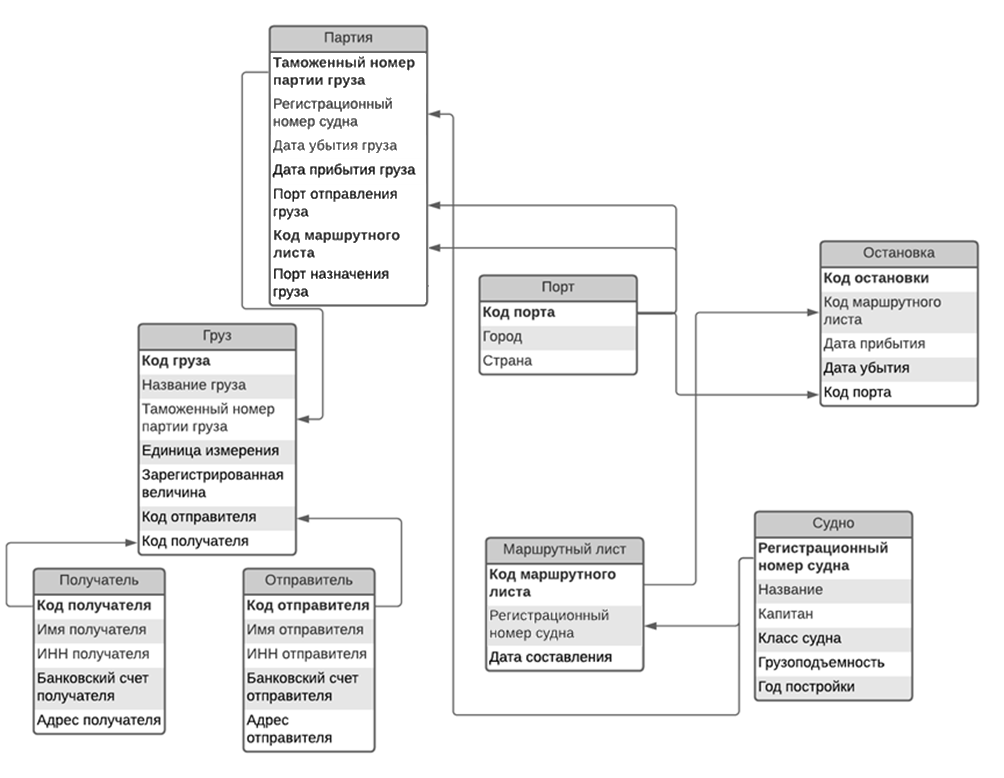


Рисунок 2.3 – Логическая схема

Структура сущностей базы данных представлена следующими таблицами:

Таблица 2.1 – Структура сущности Ship

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Тип данных | Размер | Описание |
| RegNumber | Текстовый | 11 | Регистрационный номер судна |
| Name | Текстовый | 60 | Название |
| Skipper | Текстовый | 60 | ФИО капитана |
| Type | Текстовый | 15 | Тип (танкер, сухогруз) |
| Capacity | Числовой | 10 | Грузоподъемность |
| Year | Числовой | 4 | Год постройки |

Таблица 2.2 – Структура сущности Port

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Тип данных | Размер | Описание |
| ID\_Port | Численный | 7 | Номер пункта |
| City | Текстовый | 35 | Город |
| Country | Текстовый | 60 | Страна |

Таблица 2.3 – Структура сущности Shipment

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Тип данных | Размер | Описание |
| CustomNumber | Числовой | 24 | Таможенный (custom) номер партии груза |
| RegNumber | Числовой | 11 | Регистрационный номер судна |
| DepartureDate | Дата/время | Авто | Дата убытия |
| ArriveDate | Дата/время | Авто | Дата прибытия |
| OriginPort | Текстовый | 35 | Пункт отправления |
| DestinationPort | Текстовый | 35 | Пункт назначения |
| ID\_Route | Численный | 10 | Код маршрутного листа |

Таблица 2.4 – Структура сущности Sender

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Тип данных | Размер | Описание |
| ID\_Sender | Числовой | 9 | Код отправителя |
| SenderName | Текстовый | 30 | Отправитель груза |
| INNSender | Числовой | 10 | ИНН отправителя |
| BankSender | Текстовый | 60 | Банковский счет отправителя |
| AddressSender | Текстовый | 80 | Юр. адрес отправителя |

Таблица 2.5 – Структура сущности Consignee

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Тип данных | Размер | Описание |
| ID\_Consignee | Числовой | 9 | Код получателя |
| ConsigneeName | Текстовый | 30 | Получатель груза |
| INNConsignee | Числовой | 10 | ИНН получателя груза |
| BankConsignee | Текстовый | 60 | Банковский счет получателя |
| AddressConsignee | Текстовый | 80 | Юридический адрес получателя |

Таблица 2.6 – Структура сущности Cargo

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Тип данных | Размер | Описание |
| ID\_Cargo | Числовой | 9 | Код груза |
| CargoName | Текстовый | 30 | Название груза |
| Shipment CustomNumber | Числовой | 24 | Таможенный номер партии груза |
| Unit | Текстовый | 10 | Единица измерения груза |
| InsureValue | Числовой | 8 | Застрахованная величина груза |
| ID\_Sender | Текстовый | 30 | Отправитель груза |
| ID\_Consignee | Текстовый | 30 | Получатель груза |

Таблица 2.7 – Структура сущности RouteList

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Тип данных | Размер | Описание |
| ID\_Route | Численный | 10 | Код маршрутного листа |
| ShipNumber | Численный | 10 | Регистрационный номер судна |

Таблица 2.8 – Структура сущности Stopping

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Тип данных | Размер | Описание |
| ID\_Stopping | Численный | 10 | Код остановки |
| ID\_Route | Численный | 10 | Код маршрутного листа |
| ArriveDate | Дата/время | Авто | Время прибытия |
| DepartureDate | Дата/время | Авто | Время убытия |
| ID\_Port | Численный | 7 | Номер пункта |

Представленная база данных была создана в СУБД MS SQL Server. Запрос на создание и физическая диаграмма представлена ниже.

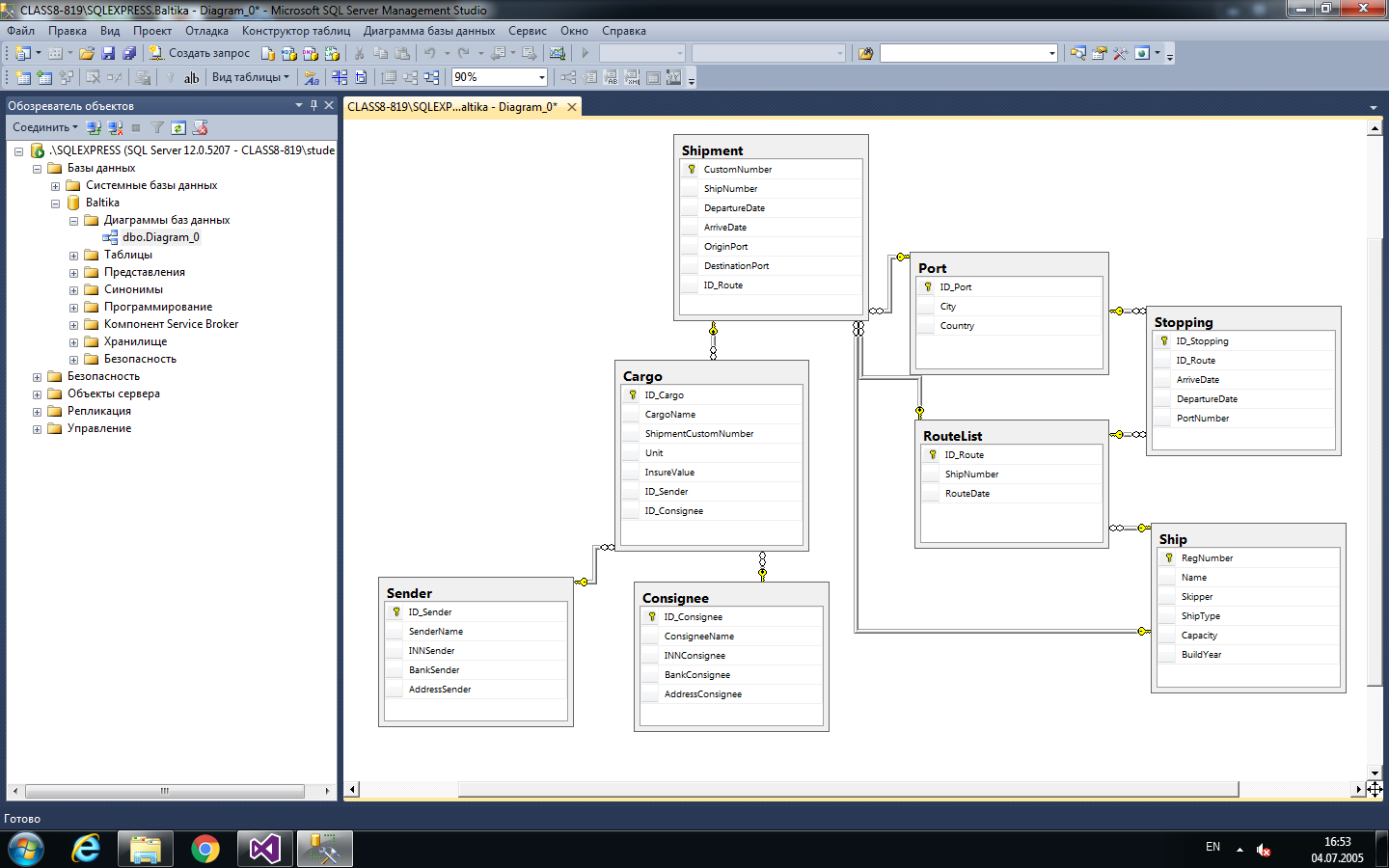


Рисунок 2.4 – Физическая диаграмма базы данных

Листинг 1 запроса на создание:

Create database Baltika;

use Baltika;

create table Ship (

RegNumber Varchar(11) primary key not null,

Name Varchar(60) not null,

Skipper Varchar(60) not null,

ShipType Varchar(15) not null,

Capacity float not null,

BuildYear integer check (BuildYear > 0 and BuildYear < 9999) not null);

create table Port (

ID\_Port Integer Identity(1,1) primary key not null,

City Varchar(35) not null,

Country Varchar(60) not null );

create table Shipment (

CustomNumber Varchar(24) primary key not null,

ShipNumber Varchar(11) not null

foreign key references Ship(RegNumber),

DepartureDate DateTime not null,

ArriveDate DateTime not null,

OriginPort Integer not null

foreign key references Port(ID\_Port),

DestinationPort Integer not null

foreign key references Port(ID\_Port));

create table Sender (

ID\_Sender Integer Identity(1,1) primary key not null,

SenderName Varchar(60) not null,

INNSender Varchar(12) not null,

BankSender Varchar(60) not null,

AddressSender Varchar(80) not null);

create table Consignee (

ID\_Consignee Integer Identity(1,1) primary key not null,

ConsigneeName Varchar(60) not null,

INNConsignee Varchar(12) not null,

BankConsignee Varchar(60) not null,

AddressConsignee Varchar(80) not null);

create table Cargo (

ID\_Cargo Integer Identity(1,1) primary key not null,

CargoName Varchar(30) not null,

ShipmentCustomNumber Varchar(24) not null

foreign key references Shipment(CustomNumber),

Unit Varchar(15) not null,

InsureValue Float not null,

ID\_Sender Integer not null

foreign key references Sender(ID\_Sender),

ID\_Consignee Integer not null

foreign key references Consignee(ID\_Consignee));

create table RouteList (

ID\_Route Integer Identity(1,1) primary key not null,

ShipNumber Varchar(11) not null

foreign key references Ship(RegNumber),

RouteDate Date not null);

create table Stopping (

ID\_Stopping Integer Identity(1,1) primary key not null,

ID\_Route Integer not null

foreign key references RouteList(ID\_Route),

ArriveDate DateTime not null,

DepartureDate DateTime not null,

PortNumber Integer not null

foreign key references Port(ID\_Port));

Ниже приведены запросы на языке SQL, необходимые для вывода данных в приложении:

select \* from Ship; -- Вывод всех транспортных средств

select \* from Port; -- Вывод всех зарегистрированных портов

select \* from Sender; -- Вывод всех зарегистрированных отправителей

select \* from Consignee; -- Вывод всех зарегистрированных получателей

select \* from Cargo; -- Вывод всего зарегистрированного груза

select \* from Shipment where DestinationPort = @port; -- Вывод поставок в определенный порт

select \* from Shipment where OriginPort = @port; -- Вывод поставок из определенного порта

select \* from RouteList where ShipNumber = @ship -- Вывод всех путешествий указанной лодки

select \* from Stopping where ID\_Route = @route order by ArriveDate -- Вывод всех остановок указанного маршрута

# 3 Проектирование шаблона выходного документа

Шаблон выходного документа представляет собой заранее созданный файл, имеющий места, которые в программе будут заменяться необходимыми значениями.

Выходной документ в данной предметной области должен содержать основную информацию о перемещении грузов: дата, время, место убытия и прибытия. Каждый груз идентифицируется кодом и названием, есть информация об отправителе и получателе, величине груза. Обязательно указывается дата составления отчета, его номер. Имя пользователя, зарегистрированного в системе и создавшего отчет, печатается в шапке отчета. Отмечается общая информация о поездке: номер маршрутного листа, регистрационный номер судна, класс судна, его капитан и период путешествия.

На рисунке 3.1 показан пример составленного документа.

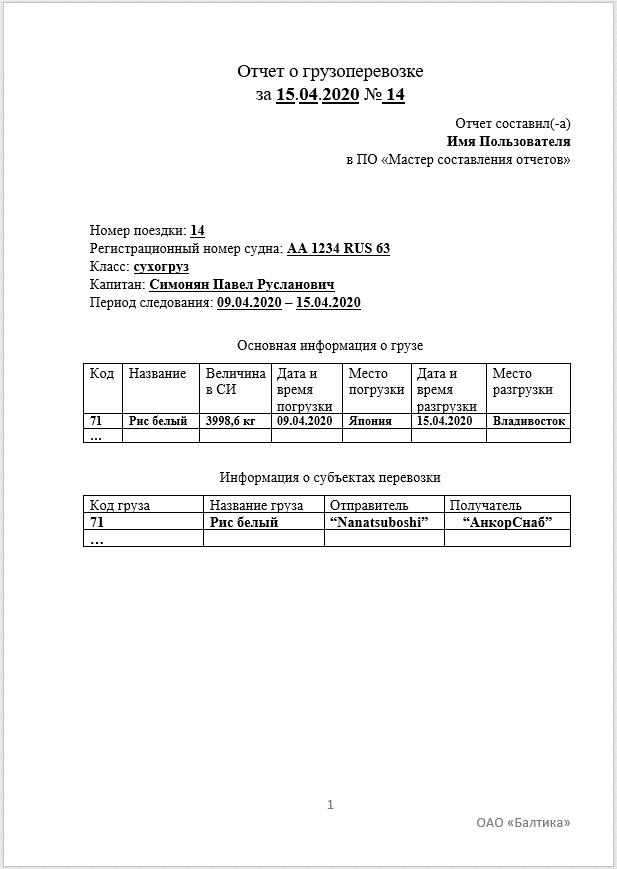
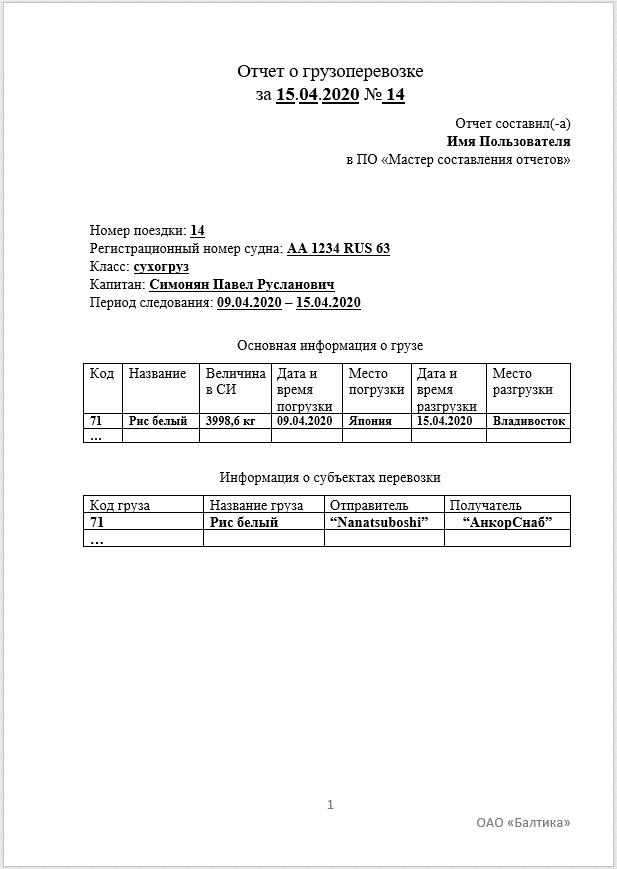


Рисунок 3.1 – Пример выходного документа

На рисунке 3.2 показано техническое описание документа с местами для вставки данных.

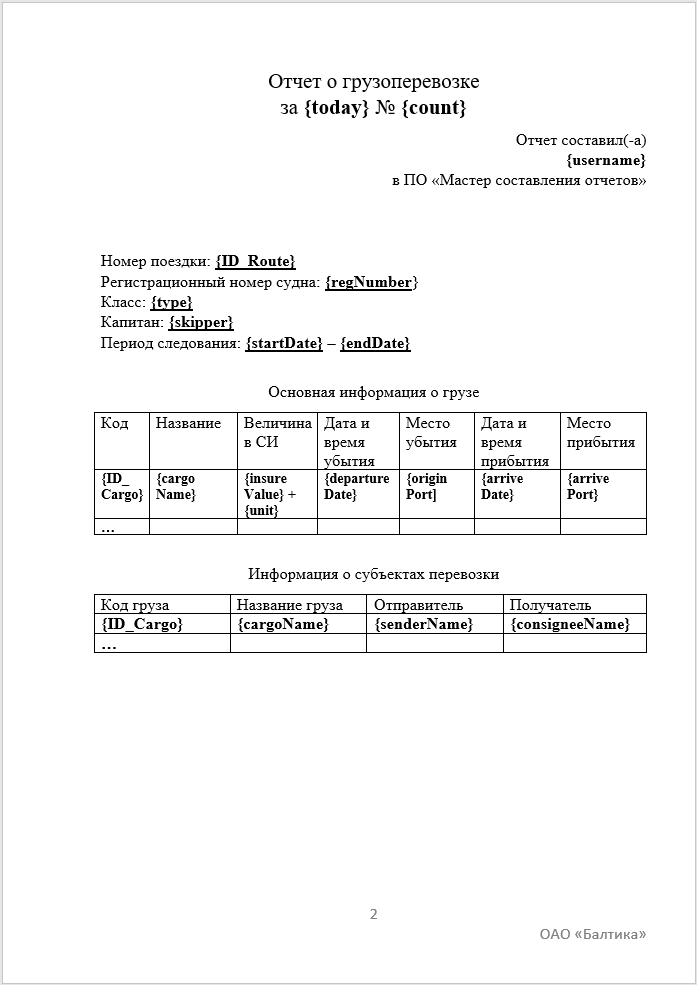


Рисунок 3.2 –Шаблон выходного документа

Они представляют из себя названия переменных в фигурных скобках, напечатанные жирным шрифтом. Названия переменных совпадают с названиями атрибутов, извлекаемых из базы данных и подставляемых в соответствующие места документа. Пользователь может выбрать предмет отчета в приложении, указав необходимый номер поездки.

Запрос на получение этих данных выглядит следующим образом:

use Baltika;

select R.ID\_Route, R.RouteDate, R.ShipNumber,

S.Skipper, S.ShipType,

C.ID\_Cargo, c.CargoName, c.InsureValue, c.Unit,

sh.DepartureDate, sh.OriginPort,

sh.ArriveDate, sh.DestinationPort,

c.ID\_Sender, c.ID\_Consignee

from RouteList R, Ship S, Cargo C, Shipment Sh

where R.ID\_Route = 2 and S.RegNumber = R.ShipNumber and R.ID\_Route = Sh.ID\_Route;