**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической Кибернетики и Информационных Технологий

****

**Отчет по курсовой работе**

по предмету «Технологии баз данных»

на тему:

«Разработка БД для АСУ «Автовокзал»»

Выполнил:

студент группы БВТ1802

Руководитель:

Вовик А. Г.

Москва 2021

Содержание

[**Введение** 3](#_Toc72322448)

[**Глава 1. Анализ предметной области** 3](#_Toc72322449)

[**1. Анализ проблемы учета продаж транспортных билетов** 3](#_Toc72322450)

[**1.1. Проблемы учета продаж транспортных билетов** 4](#_Toc72322451)

[**1.2. Формирование цели и задач проекта, определение стратегии достижения поставленной цели** 4](#_Toc72322452)

[**Глава 2. Проектирование базы данных «Автовокзал»** 5](#_Toc72322453)

[**2. Разработка концепции автоматизации учета продаж топливной и технической продукции в сети АЗС «Русь»** 5](#_Toc72322454)

[**2.1. Разработка структуры данных** 6](#_Toc72322455)

[**2.2. Структура таблиц** 7](#_Toc72322456)

[**Глава 3. Программная реализация** 10](#_Toc72322457)

[**3.1. Разработка контрольного примера** 10](#_Toc72322458)

[**3.2. Разработка алгоритмов и текстов программ** 12](#_Toc72322459)

[**3.3 Разработка оконных форм взаимодействия с конечным пользователем** 19](#_Toc72322460)

[**4. Выводы и результаты** 21](#_Toc72322461)

[**Список литературы** 22](#_Toc72322462)

[**Приложение 1 Описание объекта автоматизации, договор покупки транспортного билета** 25](#_Toc72322463)

# **Введение**

В последнее время из-за роста туризма и популяризации общественного транспорта увеличился спрос на междугородние перевозки. Люди предпочитают больше пользоваться услугами общественного транспорта, чем собственным автомобилем. Это удобно, быстро и менее затратно. Пассажиру не нужно тратить время и средства для обслуживания автомобиля.

Сотрудникам трудно выполнять бумажную работу с множеством документов. Необходимо улучшить и ускорить работу в компаниях, занимающихся продажей транспортных билетов.

Современному человеку трудно представить работу компаний без компьютера, управление экономикой на всех ее уровнях, научные исследования. Целью данной проектной работы будет автоматизация системы управления и создания базы данных для автовокзала.

В итоге данной научно-исследовательской работы была спроектирована база данных для автоматизированной системы управления продажами транспортных билетов. Построена логическая и физическая структура данных и база данных. Внедрена автоматизированная система по учету продаж билетов, что существенно ускорило работу по продажам билетов в меньшие сроки.

# **Глава 1. Анализ предметной области**

## **1. Анализ проблемы учета продаж транспортных билетов**

Автовокзалы должны постоянно контролировать все процессы движения общественного транспорта и эффективно обновлять списки своих клиентов, сотрудников и транспортного парка.

Классический способ мониторинга с помощью бумажных форм и отчётов неэффективен и небезопасен для автовокзала. Могут возникать различные проблемы, из-за которых автовокзал может нести убытки.

Было принято решение разработать АСУ БД для автовокзала, чтобы эффективно управлять финансовыми, материальными и информационными потоками.

Продажа билетов достаточно сложный процесс, с которым связано много документов и лиц, разработка АСУ БД для данной отрасли необходима. Разработанная система позволит фирме снизить свои риски и нарастить объём продаж билетов, многие процессы станут автоматизированы, что позитивно отразится на работе автовокзала.

## **1.1. Проблемы учета продаж транспортных билетов**

Анализ учета продаж транспортных билетов выявил следующие проблемы:

– отсутствуют требований к ведению истории о продажах. При продаже билета отсутствует полная история продажа билета, что небезопасно для автовокзала. Нет контроля за финансовым потоков, что может приводить к убыткам;

– большое количество бумажных документов в процессе продажи транспортного билета. В процессе работы возникает большое количество бумажных документов, что усложняет поиск нужных бумажных документов и увеличивает время продажи транспортного билета. Такое нельзя допускать, автовокзал должен быть готов всегда предоставить обслуживание клиенту, иначе он рискует уменьшить свою клиентскую базу, что нежелательно;

– сложность контроля за ходом выполнения продаж билетов сотрудниками. Руководители должны следить за ходом выполнения работ сотрудниками. Управленцам должна быть доступна информация о всех финансовых, материальных и информационных потоках, иначе качество предоставляемых автовокзалом услуг может снизиться, что клиенты не оценят. Это приводит к финансовым убыткам;

– ручная работа. Много необоснованных трудозатрат с бумажными документами. Сотрудники тратят много времени на необоснованную работу, которую может проделывать АСУ. Если мы избавим сотрудников от ненужной работы, мы получим прирост в скорости работы, что приведёт к повышению дохода.

## **1.2. Формирование цели и задач проекта, определение стратегии достижения поставленной цели**

Целью проекта является разработка БД для автовокзала для автоматизации продаж транспортных билетов, что позволит автовокзалу увеличить количество продаж билетов и прибыль.

Для достижения поставленной цели определим следующие задачи проекта:

– провести системный анализ предметной области автовокзала, изучить полностью процесс работы данного автовокзала и разработать соответствующую концепцию будущей АСУ;

– разработать логическую и физическую модель БД, модель, которая призвана облегчить работу по созданию БД, что позволит уменьшить время разработки АСУ БД;

– осуществить реализацию БД на выбранной СУБД, реализация будет осуществляться с помощью СУБД Microsoft Server SQL и языка программирования C#, а также будет представлен код необходимых объектов на языке SQL, что поможет портировать АСУ на другие СУБД.

Разрабатываемая автоматизированная система управления продажами транспортных билетов для автовокзала является актуальной в связи с высокой потребностью граждан в общественном транспорте.

# **Глава 2. Проектирование базы данных «Автовокзал»**

## **2. Разработка концепции автоматизации учета продаж топливной и технической продукции в сети АЗС «Русь»**

Разработаем следующую концепцию АСУ автовокзала.

Опираясь на проблемы учёта продаж транспортных билетов и на конкретный анализ процесса работы данной области, представим в АСУ автовокзала основные таблицы:

– продажи билетов, хранит информацию о автобусе, пассажире, цене и месте прибытия.

– пассажиры, хранит информацию о клиенте.

– автобусы, хранит информацию о автобусах и их производителях.

– производители, хранит информацию о производителях и их реквизиты.

– классы, хранит информацию о классе данного билета.

– города, хранит информацию о месте прибытия.

– страны, хранит информацию о стране места прибытия.

– сотрудники, хранит информацию о сотрудниках, которые работают на автовокзале.

– должности, хранит информацию о должностях сотрудников.

– терминалы, хранит информацию о терминалах автовокзала, с которых осуществляются

## **2.1. Разработка структуры данных**

Разработаем структуру хранения информации. На основании предложенной структуры будет проектироваться база данных. С этой целью выделим объекты и связи между ними. Анализ показывает, что в разрабатываемой системе выделяются такие объекты как: сотрудники, должности, автобусы, поставщики, пассажиры, города, страны, классы, терминалы, продажи билетов. Отобразим на Рисунке 1 объекты.

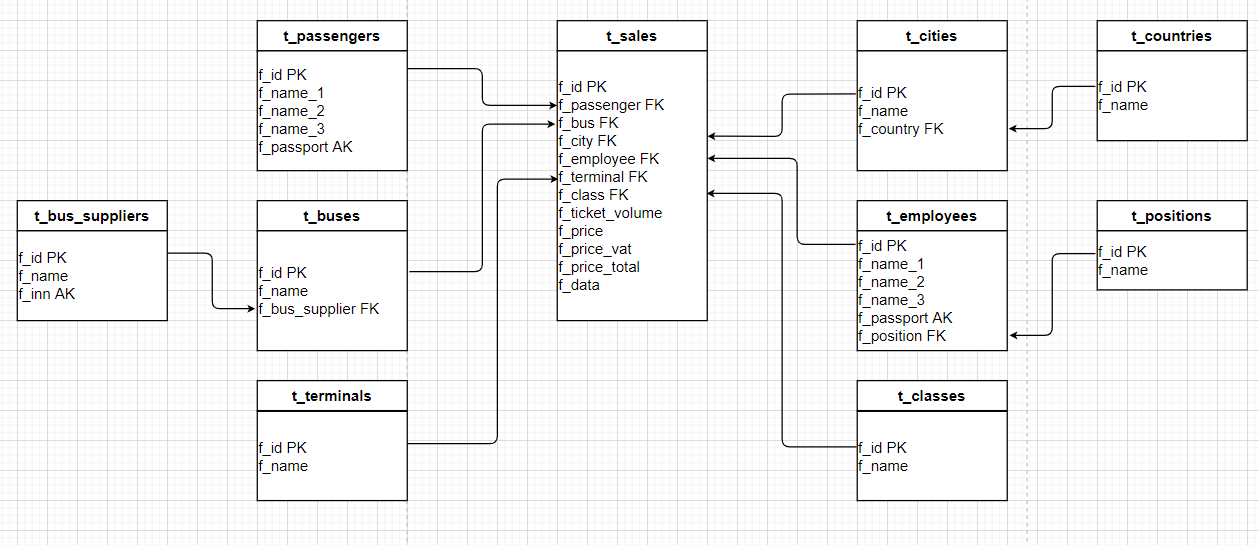


Рисунок 1 – Структура БД

На данном рисунке показаны основные объекты, без которых невозможны продажи транспортных билетов.

В данной главе была разработана логическая модель данных. В следующей главе будет рассмотрено, на основании логической модели данных, разработка физической модели.

На данном рисунке показаны таблицы и взаимосвязи (один ко многим) между ними. Например, таблица «Продажи», т.е. договор продажи может включать много значений таблицы «Автобусы», т.е. типы транспортного средства. Это означает, что акты продаж могут составляться для нескольких видов автобусов. Аналогично этому примеру построены другие таблицы и связи.

## **2.2. Структура таблиц**

Структура таблицы включает, что каждый объект должен содержать определенный тип данных. Опишем структуру таблиц.

Сотрудники

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Фамилия | NVARCHAR(50) |  |  |
| Имя | NVARCHAR(50) |  |  |
| Отчество | NVARCHAR(50) |  |  |
| Паспорт | NVARCHAR(10) | Уникальный ключ | Unique key |
| Должность | INT | Внешний ключ | Foreign key |

Должности

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Название | NVARCHAR(50) |  |  |

Автобусы

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Название | NVARCHAR(50) |  |  |
| Поставщик | INT | Внешний ключ | Foreign key |

Поставщики

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Название | NVARCHAR(50) |  |  |
| ИНН | NVARCHAR(10) |  |  |

Пассажиры

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Фамилия | NVARCHAR(50) |  |  |
| Имя | NVARCHAR(50) |  |  |
| Отчество | NVARCHAR(50) |  |  |
| Паспорт | NVARCHAR(10) | Уникальный ключ | Unique key |

Города

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Название | NVARCHAR(50) |  |  |
| Страна | INT | Внешний ключ | Foreign key |

Страны

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Название | NVARCHAR(50) |  |  |

Классы

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Название | NVARCHAR(50) |  |  |

Терминалы

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Название | NVARCHAR(50) |  |  |

Продажи

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Пояснение | Ключи |
| Ключ | INT | Первичный ключ | Primary key |
| Пассажир | INT | Внешний ключ | Foreign key |
| Автобус | INT | Внешний ключ | Foreign key |
| Город | INT | Внешний ключ | Foreign key |
| Сотрудник | INT | Внешний ключ | Foreign key |
| Терминал | INT | Внешний ключ | Foreign key |
| Класс | INT | Внешний ключ | Foreign key |
| Количество | INT |  |  |
| Цена | MONEY |  |  |
| Цена с НДС | MONEY |  |  |
| Цена сумма | MONEY |  |  |
| Дата | DATE |  |  |

Были разработаны все основные и дополнительных таблицы, которые используется в разрабатываемой АСУ БД автовокзала. Далее представлен контрольный пример проекта и пользовательский интерфейс, с которым взаимодействуют сотрудники.

# **Глава 3. Программная реализация**

## **3.1. Разработка контрольного примера**

Для проверки возможности представлять информацию о продажах транспортных билетов в спроектированной структуре данных, разработаем контрольный пример.

Данный пример содержит: информацию о всех основных таблицах.

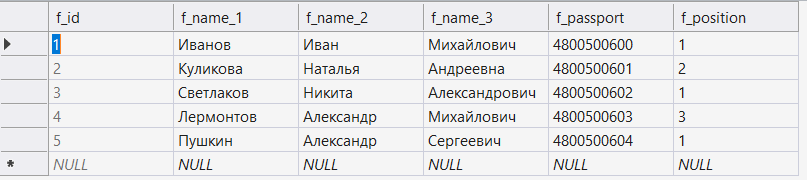


Рисунок 1 – Таблица «Сотрудники»

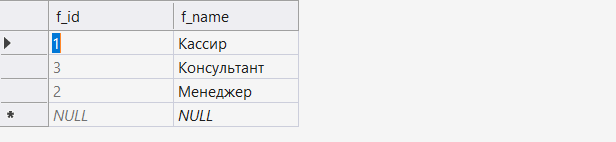


Рисунок 2 – Таблица «Должности»

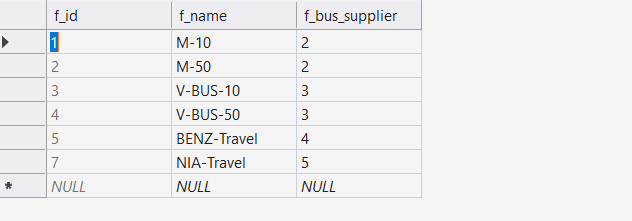


Рисунок 3 – Таблица «Автобусы»

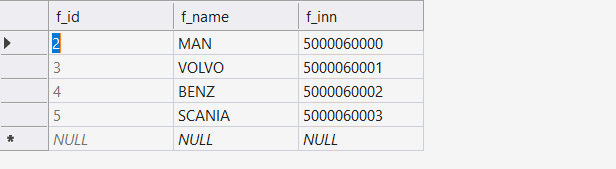


Рисунок 4 – Таблица «Поставщики»



Рисунок 5 – Таблица «Города»

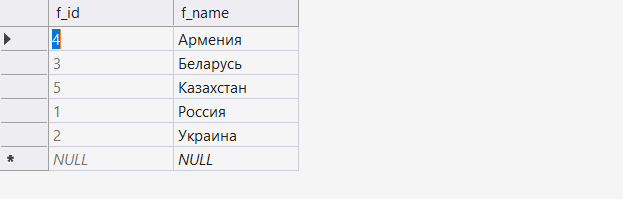


Рисунок 6 – Таблица «Страны»

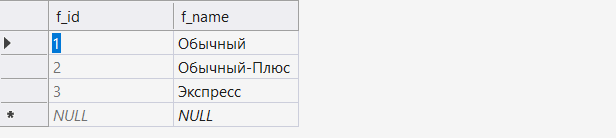


Рисунок 7 – Таблица «Классы»

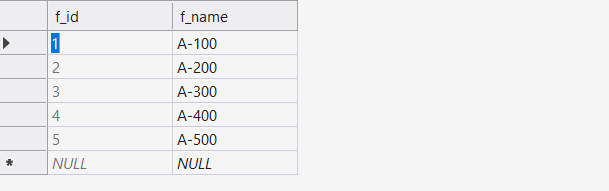


Рисунок 8 – Таблица «Терминалы»

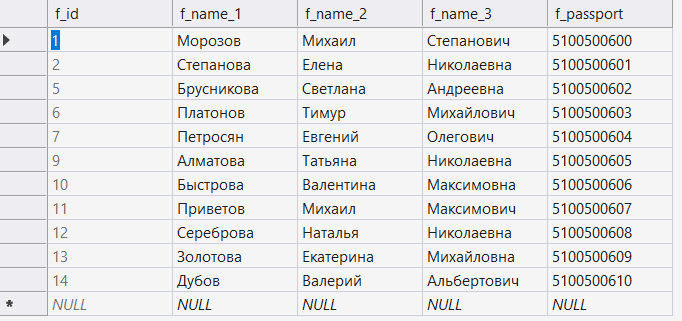


Рисунок 9 – Таблица «Пассажиры»

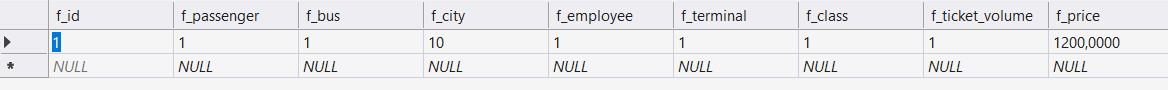


Рисунок 10 – Таблица «Продажи билетов»

Этот контрольный пример был заполнен в программе Visual Studio 2019 c помощью Microsoft Server SQL.

## **3.2. Разработка алгоритмов и текстов программ**

Разработка базы данных должна начинаться с построения структуры таблиц и определения первичных, внешних и альтернативных ключей.

Все ключи в таблицах именуются следующим образом: Первичный ключ = Primary key, Внешний ключ = Foreign key, уникальный ключ = Unique key.

Создадим БД «Database\_BUS»

**Создание таблиц**

Создадим таблицу «Сотрудники».

CREATE TABLE [dbo].[t\_employees] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name\_1] NVARCHAR (50) NOT NULL,

[f\_name\_2] NVARCHAR (50) NOT NULL,

[f\_name\_3] NVARCHAR (50) NULL,

[f\_passport] NVARCHAR (10) NOT NULL,

[f\_position] INT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC),

CONSTRAINT [c\_fk\_t\_employees\_f\_position] FOREIGN KEY ([f\_position]) REFERENCES [dbo].[t\_positions] ([f\_id])

);

Создадим таблицу «Должности».

CREATE TABLE [dbo].[t\_positions] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name] NVARCHAR (50) NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC)

);

GO

CREATE UNIQUE NONCLUSTERED INDEX [idx\_positions\_name]

ON [dbo].[t\_positions]([f\_name] ASC);

Создадим таблицу «Автобусы».

CREATE TABLE [dbo].[t\_buses] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name] NVARCHAR (50) NOT NULL,

[f\_bus\_supplier] INT NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC),

CONSTRAINT [c\_fk\_t\_buses\_f\_bus\_supplier] FOREIGN KEY ([f\_bus\_supplier]) REFERENCES [dbo].[t\_bus\_suppliers] ([f\_id])

);

Создадим таблицу «Поставщики».

CREATE TABLE [dbo].[t\_bus\_suppliers] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name] NVARCHAR (50) NOT NULL,

[f\_inn] NVARCHAR (10) NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC)

);

GO

CREATE UNIQUE NONCLUSTERED INDEX [idx\_bus\_suppliers\_inn]

ON [dbo].[t\_bus\_suppliers]([f\_inn] ASC);

Создадим таблицу «Города».

CREATE TABLE [dbo].[t\_cities] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name] NVARCHAR (50) NOT NULL,

[f\_country] INT NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC),

CONSTRAINT [c\_fk\_t\_cities\_f\_country] FOREIGN KEY ([f\_country]) REFERENCES [dbo].[t\_countries] ([f\_id])

);

Создадим таблицу «Страны».

CREATE TABLE [dbo].[t\_countries] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name] NVARCHAR (50) NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC)

);

GO

CREATE UNIQUE NONCLUSTERED INDEX [idx\_countries\_name]

ON [dbo].[t\_countries]([f\_name] ASC);

Создадим таблицу «Классы».

CREATE TABLE [dbo].[t\_classes] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name] NVARCHAR (50) NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC)

);

GO

CREATE UNIQUE NONCLUSTERED INDEX [idx\_classes\_name]

ON [dbo].[t\_classes]([f\_name] ASC);

Создадим таблицу «Терминалы».

CREATE TABLE [dbo].[t\_terminals] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name] NVARCHAR (50) NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC)

);

GO

CREATE UNIQUE NONCLUSTERED INDEX [idx\_terminals\_name]

ON [dbo].[t\_terminals]([f\_name] ASC);

Создадим таблицу «Пассажиры».

CREATE TABLE [dbo].[t\_passengers] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_name\_1] NVARCHAR (50) NOT NULL,

[f\_name\_2] NVARCHAR (50) NOT NULL,

[f\_name\_3] NVARCHAR (50) NULL,

[f\_passport] NVARCHAR (10) NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC)

);

GO

CREATE UNIQUE NONCLUSTERED INDEX [idx\_passengers\_passport]

ON [dbo].[t\_passengers]([f\_passport] ASC);

Создадим таблицу «Продажи билетов» и зададим три триггера для автоматического подсчёта цены.

CREATE TABLE [dbo].[t\_sales] (

[f\_id] INT IDENTITY (1, 1) NOT NULL,

[f\_passenger] INT NULL,

[f\_bus] INT NOT NULL,

[f\_city] INT NOT NULL,

[f\_employee] INT NOT NULL,

[f\_terminal] INT NOT NULL,

[f\_class] INT NOT NULL,

[f\_ticket\_volume] INT NOT NULL,

[f\_price] MONEY NOT NULL,

[f\_price\_vat] MONEY NULL,

[f\_price\_total] MONEY NULL,

[f\_data] DATE NOT NULL,

PRIMARY KEY CLUSTERED ([f\_id] ASC),

CONSTRAINT [c\_fk\_t\_sales\_f\_passenger] FOREIGN KEY ([f\_passenger]) REFERENCES [dbo].[t\_passengers] ([f\_id]),

CONSTRAINT [c\_fk\_t\_sales\_f\_bus] FOREIGN KEY ([f\_bus]) REFERENCES [dbo].[t\_buses] ([f\_id]),

CONSTRAINT [c\_fk\_t\_sales\_f\_city] FOREIGN KEY ([f\_city]) REFERENCES [dbo].[t\_cities] ([f\_id]),

CONSTRAINT [c\_fk\_t\_sales\_f\_employee] FOREIGN KEY ([f\_employee]) REFERENCES [dbo].[t\_employees] ([f\_id]),

CONSTRAINT [c\_fk\_t\_sales\_f\_terminal] FOREIGN KEY ([f\_terminal]) REFERENCES [dbo].[t\_terminals] ([f\_id]),

CONSTRAINT [c\_fk\_t\_sales\_f\_class] FOREIGN KEY ([f\_class]) REFERENCES [dbo].[t\_classes] ([f\_id])

);

GO

CREATE TRIGGER tr\_ticket\_price\_1

ON t\_sales

FOR INSERT

AS

BEGIN

UPDATE

t\_sales

SET

f\_price\_vat = f\_price / 100 \* 120

WHERE

t\_sales.f\_id = (select f\_id from inserted);

END

GO

CREATE TRIGGER tr\_ticket\_price\_2

ON t\_sales

FOR INSERT

AS

BEGIN

UPDATE

t\_sales

SET

f\_price\_total = f\_price\_vat \* f\_ticket\_volume

WHERE

t\_sales.f\_id = (select f\_id from inserted);

END

GO

CREATE TRIGGER tr\_ticket\_price\_3

ON t\_sales

FOR INSERT

AS

BEGIN

UPDATE

t\_sales

SET

f\_price\_total = f\_price\_total + (f\_class \* 1000)

WHERE

t\_sales.f\_id = (select f\_id from inserted);

END

**Создание представлений**

Создадим представление «Сотрудники»

CREATE VIEW [dbo].[v\_employees]

AS SELECT

t\_employees.f\_id ID,

t\_employees.f\_name\_1 Фамилия,

t\_employees.f\_name\_2 Имя,

t\_employees.f\_name\_3 Отчество,

t\_employees.f\_passport Паспорт,

t\_positions.f\_name Должность

FROM [t\_employees],[t\_positions]

WHERE t\_employees.f\_position = t\_positions.f\_id;

Создадим представление «Автобусы»

CREATE VIEW [dbo].[v\_buses]

AS SELECT

t\_buses.f\_id ID,

t\_buses.f\_name Название,

t\_bus\_suppliers.f\_name Поставщик

FROM [t\_buses],[t\_bus\_suppliers]

WHERE t\_buses.f\_bus\_supplier = t\_bus\_suppliers.f\_id;

Создадим представление «Города»

CREATE VIEW [dbo].[v\_cities]

AS SELECT

t\_cities.f\_id ID,

t\_cities.f\_name Название,

t\_countries.f\_name Страна

FROM [t\_cities],[t\_countries]

WHERE t\_cities.f\_country = t\_countries.f\_id;

Создадим представление «Классы»

CREATE VIEW [dbo].[v\_classes]

AS SELECT

f\_id ID,

f\_name Название

FROM [t\_classes]

Создадим представление «Терминалы»

CREATE VIEW [dbo].[v\_terminals]

AS SELECT

f\_id ID,

f\_name Название

FROM [t\_terminals]

Создадим представление «Пассажиры»

CREATE VIEW [dbo].[v\_passengers]

AS SELECT

f\_id ID,

f\_name\_1 Фамилия,

f\_name\_2 Имя,

f\_name\_3 Отчество,

f\_passport Паспорт

FROM [t\_passengers]

Создадим представление «Продажи билетов»

CREATE VIEW [dbo].[v\_sales]

AS SELECT

t\_sales.f\_id ID,

t\_passengers.f\_name\_1 Фамилия,

t\_passengers.f\_name\_2 Имя,

t\_passengers.f\_name\_3 Отчество,

t\_buses.f\_name Автобус,

t\_cities.f\_name Город,

t\_employees.f\_name\_1 Работник,

t\_terminals.f\_name Терминал,

t\_classes.f\_name Класс,

t\_sales.f\_ticket\_volume Количество,

t\_sales.f\_price Цена,

t\_sales.f\_price\_vat Цена\_НДС,

t\_sales.f\_price\_total Цена\_Сумма,

t\_sales.f\_data Дата

FROM [t\_sales],[t\_passengers],[t\_buses],[t\_cities],[t\_employees],[t\_terminals],[t\_classes]

WHERE t\_sales.f\_passenger = t\_passengers.f\_id and

t\_sales.f\_bus = t\_buses.f\_id and

t\_sales.f\_city = t\_cities.f\_id and

t\_sales.f\_employee = t\_employees.f\_id and

t\_sales.f\_terminal = t\_terminals.f\_id and

t\_sales.f\_class = t\_classes.f\_id;

**Основной код интерфейса приложения**

using System;

using System.Windows.Forms;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

namespace TBD\_BUS

{

public partial class Form1 : Form

{

private SqlConnection sqlConnection = null;

private SqlDataAdapter dataAdapter = null;

private DataSet dataSet = null;

private SqlCommand command = null;

private void view(string str)

{

dataAdapter = new SqlDataAdapter(str, sqlConnection);

dataSet = new DataSet();

dataAdapter.Fill(dataSet);

dataGridView1.DataSource = dataSet.Tables[0];

dataAdapter.Dispose();

dataSet.Dispose();

}

private void procedure(string str)

{

command = new SqlCommand(str, sqlConnection);

if (command.ExecuteNonQuery().ToString() == "-1")

{

MessageBox.Show("Ошибка!");

}

else

{

MessageBox.Show("Успешно!");

}

command.Dispose();

}

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

sqlConnection = new SqlConnection(ConfigurationManager.ConnectionStrings["BUSDB"].ConnectionString);

sqlConnection.Open();

if (sqlConnection.State == ConnectionState.Open)

{

MessageBox.Show("Подключение к БД установлено");

}

}

private void button1\_Click(object sender, EventArgs e)

{

view("select \* from v\_passengers");

}

private void button2\_Click(object sender, EventArgs e)

{

view("select \* from v\_buses");

}

private void button3\_Click(object sender, EventArgs e)

{

view("select \* from v\_cities");

}

private void button4\_Click(object sender, EventArgs e)

{

view("select \* from v\_employees");

}

private void button5\_Click(object sender, EventArgs e)

{

view("select \* from v\_terminals");

}

private void button6\_Click(object sender, EventArgs e)

{

view("select \* from v\_classes");

}

private void button7\_Click(object sender, EventArgs e)

{

view("select \* from v\_sales");

}

private void buttonSale\_Click(object sender, EventArgs e)

{

procedure($"INSERT INTO t\_sales(f\_passenger, f\_bus, f\_city, f\_employee, f\_terminal, f\_class, f\_ticket\_volume, f\_price, f\_data)VALUES(N'{textBox1.Text}', N'{textBox2.Text}', N'{textBox3.Text}', N'{textBox4.Text}', N'{textBox5.Text}', N'{textBox6.Text}',N'{textBox7.Text}', N'{textBox8.Text}', SYSDATETIME());");

}

private void buttonAdd\_Click(object sender, EventArgs e)

{

procedure($"INSERT INTO t\_passengers(f\_name\_1, f\_name\_2, f\_name\_3, f\_passport) VALUES(N'{textBox9.Text}', N'{textBox10.Text}', N'{textBox11.Text}', N'{textBox12.Text}');");

}

}

}

## **3.3 Разработка оконных форм взаимодействия с конечным пользователем**

Ведение учёта продаж билетов описывается при помощи оконных форм взаимодействия с конечным пользователем.

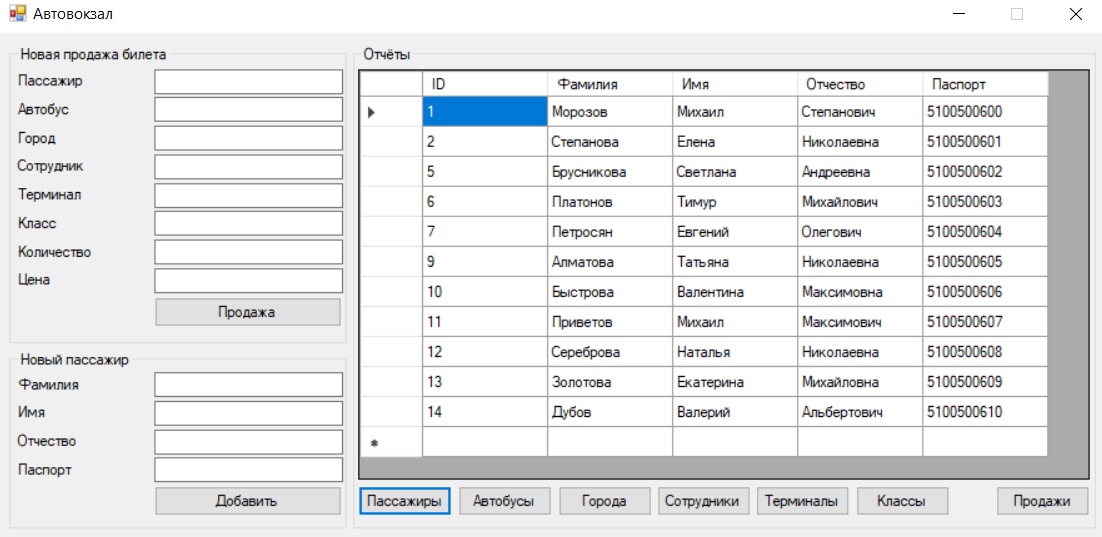


Рисунок 1 – Форма «Автовокзал»

Для работы с формой или формирования отчётов, следует нажать на одну из кнопок формы автовокзала.

Заполнение данных о продажах билетов или добавлении новых пассажиров происходит при помощи главной формы, представленная на Рисунке 2.

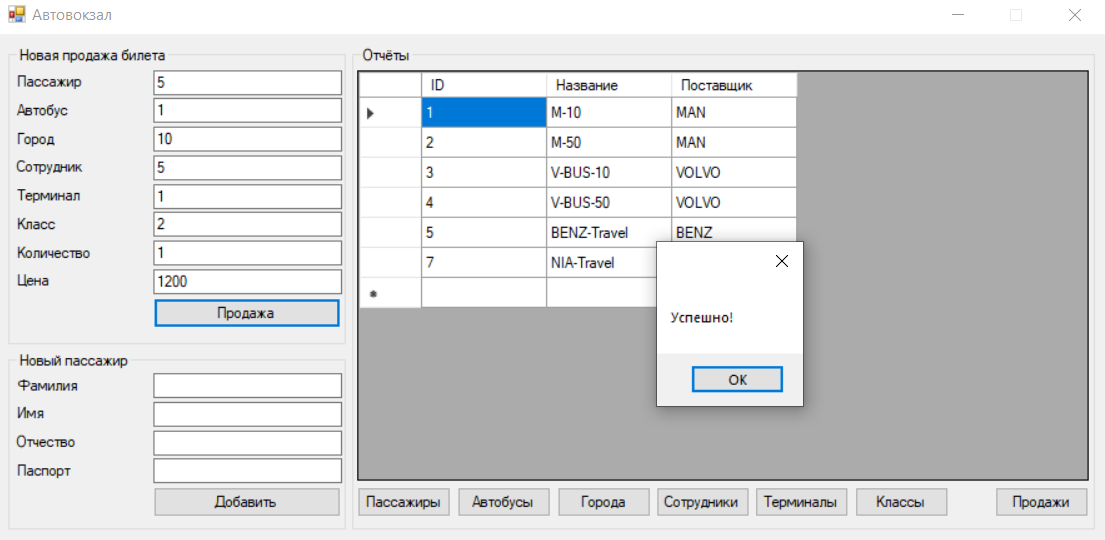
****

Рисунок 2 – Формы «Продажа билета или добавление пассажира»

Вывода данных БД представлен на Рисунке 3.

При нажатии на одну из кнопок в отчёт будут выведены данные из выбранного представления.

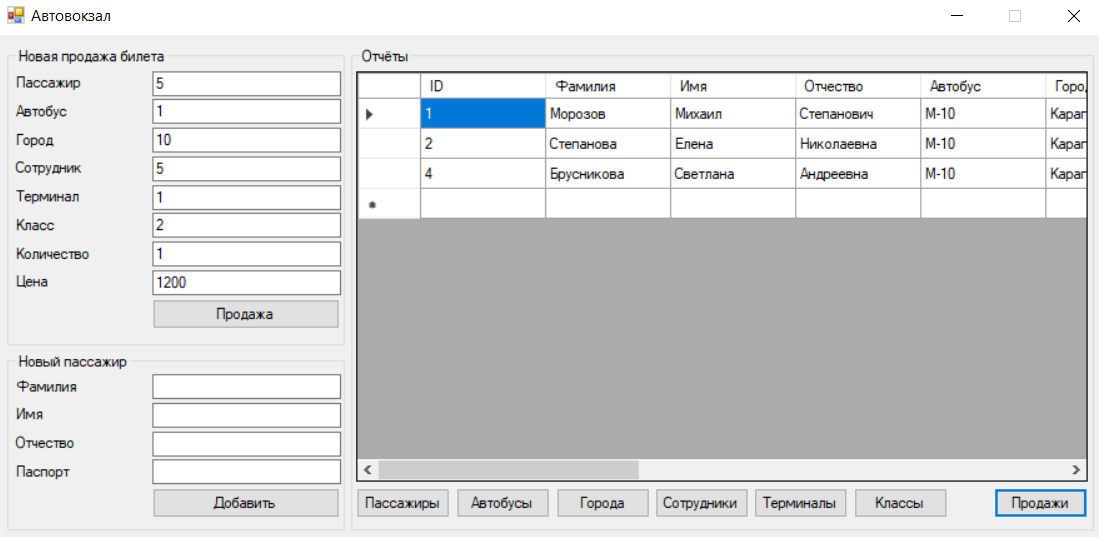
****

Рисунок 3 – Формы «Вывод информации»

Эти оконные формы помогают усовершенствовать процесс ввода, передачи и хранения информации автовокзала.

# **4. Выводы и результаты**

При выполнении проектной работы по проектированию и разработке автоматизированной системы учета продаж транспортных билетов выявлены следующие результаты:

– проанализирован порядок системы учета продаж билетов. Выявлены слабые места в данном учете продаж билетов. Предложен вариант наиболее рационального решения по автоматизации учета продаж билетов;

– проанализировано и выбрано, соответствующее программное обеспечение, СУБД. Построена структура данных по автоматизации учета продаж билетов. Спроектирована и разработана база данных по учету продаж билетов.

Выводы:

– оконные формы лучше разрабатывать в средствах Visual Studio 2019 с использованием языка программирования C#, которые следует создать на основании учёта продаж товаров;

– при анализе и выборе программного обеспечения следует учитывать плюсы и минусы современных антивирусных программ, операционных систем и баз данных;

– представлять программный код построения таблиц и запросов следует в виде отчёта.

– разработанная АСУ БД позволит повысить объём продаж автомобилей и прибыль для данного автовокзала;

– взаимодействие с ФНС целесообразно осуществлять в электронном формате.

# **Список литературы**

1. Давиденко К.Я. Технология программирования АСУ ТП. - М.: Энергоатомиздат, 1986.-184 с.: ил.

2. Олейник С.У. и др. Автоматизированные системы управления машиностроительными предприятиями: Учеб. для вузов./ - М.: Высш. шк., 1991. - 222 с.: ил. ISBN 5-06-001859-8

3. Баранюк В.А. и др. Основы создания больших АСУ. М.: Сов. радио. 1979.

4. Мамиконов А.Г. Проектирование АСУ: Учебник для специальности «АСУ» вузов. - М.: Высш. шк., 1987. -303 с.: ил.

5. Грекул М.И. Проектирование информационных систем. [www.intuit.ru](http://www.intuit.ru).

6. Черемных С.В. и др. Моделирование и анализ систем. IDEF-Технологии : практикум.,- М.: «ФиС», 2002 г.-188 с.

7. Черемных С.В. и др.Структурный анализ систем. IDEF-Технологии.- М.:«ФиС»,2002 г.-207 с.

8. Мейер Д. Теория реляционных баз данных: Пер. с англ. - М.: Мир, 1987. - 608 с., ил.

9. Мартин Дж. Организация баз данных в вычислительных системах. Издательство "Мир" 1980.

10. Кодд Е.Ф. «Реляционная модель данных для больших совместно используемых банков данных» с. 145-160 в журнале «СУБД» N1/1995. Москва. ул.Авиамоторная, 55.

11. Гилула М.М. Множественная модель данных в информационных системах. - М.: Наука, 1992. - 208с. ISBN 5-02-006938-8

12. Ковтун И.И. Кузьмин И.В. «Матрично-реляционная модель данных в автоматизированных системах управления потоками процессуальных документов и органами предварительного следствия» в материалах научно-практического семинара ВНИИ МВД «Персональный компьютер на службе криминальной милиции и следствия. Возможности и перспективы». 1997 год. Стр. 54-61.

13. Ковтун И.И. «Матрично-реляционная модель данных в автоматизированных производственных системах». В журнале «Информатика-машиностроение». N3 1997 год. Стр. 2-11. Москва. «ВиражЦентр».

14. Ковтун И.И. «Матрично-реляционный подход к проектированию автоматизированных систем управления предприятием». В журнале «Информатика-машиностроение». N4 1997 год. Стр. 5-15. Москва. «ВиражЦентр».

15. I.I.Kovtun. «A Matrix-Relational Data Model for Information Representation and Processing in Computer-Aided Monitoring and Management Systems». In journal «Programming and Computer Software», Vol. 23, No. 6, 1997, pp. 331- 341. Moskow. «Nauka». 69

16. Oracle 8. Энциклопедия пользователя.: Пер. с англ./ Компания Advannced Information Systems и др. – К.: Издательство «ДиаСофт», 1998. – 864 с.

17. Мартин Грабер. Введение в SQL. Перевод В.А. Ястребов. Издательство «ЛОРИ». 1996. (Martin Gruber. Understunding SQL. Copyright 1990. SYBEX Inc., 2021. Challenger Drive. Alameda, CA 94501).

ГОСТ 34.003-90 Основные понятия.

ГОСТ 34.201-89. Виды, комплектность и обозначение документов при создании автоматизированных систем.

ГОСТ 34.601-90. Автоматизированные системы. Стадии создания.

ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы.

ГОСТ 34.603-92. Виды испытаний автоматизированных систем.

ГОСТ 24.103-84. Основные положения.

ГОСТ 24.104-85. Общие требования к разработке АСУ.

ГОСТ 24.602-86. Состав и содержание работ по стадиям создания.

ГОСТ 19.101-77. Виды программ и программных документов.

ГОСТ 24.202-80. Требования к содержанию документа «Техникоэкономическое обоснование создания АСУ».

ГОСТ 24.203-80. Требования к содержанию общесистемных документов.

ГОСТ 24.204-80. Требования к содержанию документа «Описание постановки задачи».

ГОСТ 24.205-80. Требования к содержанию документов по информационному обеспечению.

ГОСТ 24.206-80. Требования к содержанию документов по техническому обеспечению.

ГОСТ 24.207-80. Требования к содержанию документов по программному обеспечению.

ГОСТ 24.208-80. Требования к содержанию документов стадии «Ввод в эксплуатацию».

ГОСТ 24.209-80. Требования к содержанию документов по организационному обеспечению.

ГОСТ 24.210-82. Требования к содержанию документов по функциональной части.

ГОСТ 24.211-82. Требования к содержанию документа «Описание алгоритма».

ГОСТ 24.301-80. Общие требования к выполнению текстовых документов.

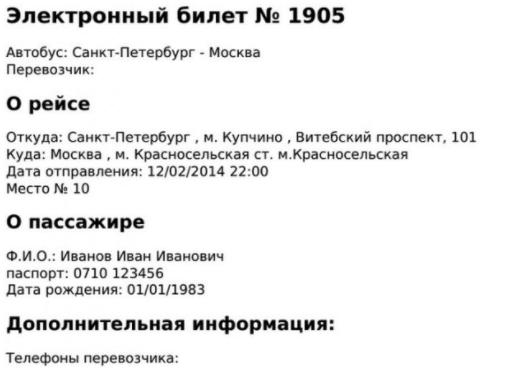
ГОСТ 24.302-80. Общие требования к выполнению схем.

ГОСТ 24.304-82. Требования к выполнению чертежей.

ГОСТ 24.401-80. Внесение изменений в техническую документацию.

ГОСТ 24.402-80. Учет, хранение и обращение документации.

# **Приложение 1 Описание объекта автоматизации, договор покупки транспортного билета**

****