

Институт ИТАСУ

Кафедра инженерной кибернетики

Направление подготовки: 01.03.04 «прикладная математика»

Квалификация (степень): бакалавр

КУРСОВАЯ РАБОТА

учебная дисциплина

«Специальные главы по базам данных»

VII семестр 2019 – 2020 у. г.

Студент: Новицкий Дмитрий

Группа: БПМ-16-2

Преподаватель: доц., к.т.н. А. С. Кожаринов

Оценка:

Дата:

Оглавление

Список используемых основных сокращений.....	3
Введение	4
Постановка задачи	5
Цель работы	5
Описание работы приложения	5
Использованные средства разработки и системные требования	7
Краткое описание языковых и инструментальных средств, использованных для создания приложения.....	7
Модель данных.....	8
Описание таблиц базы данных.....	9
Описание структуры таблиц.....	10
Описание клиентского приложения.....	17
Общие сведения.....	17
Список классов	17
Организация ролевого доступа к данным.....	23
Тематические запросы к данным	25
Выводы.....	27
Список используемых источников.....	28

Список используемых основных сокращений

1. ИС – информационная система.
2. АИС – автоматизированная информационная система.
3. БД – база данных.
4. СУБД – система управления базами данных.
5. ВПП – взлётно-посадочная полоса.
6. SQL – structured query language («язык структурированных запросов»)

Введение

В настоящее время главное место в экономике организаций занимает развитие информационных технологий. Успех практически любого предприятия состоит в том, чтобы организация работала как слаженный механизм.

С появлением информационных технологий все большее количество процессов прошло автоматизацию. С каждым годом сильнее ощущается потребность в них – без хорошей, отлаженной техники, программного обеспечения, обученного к работе с ней персонала невозможно конкурировать на рынке. Одной из наиболее важных характеристик эффективности деятельности организации является наличие хорошей информационной технологии, позволяющей работать всем подразделениям предприятия достаточно быстро, правильно и без сбоев.

Инструментом, позволяющим получить информацию необходимого объема и качества, являются АИС различного назначения.

Постановка задачи

Разрабатываемое приложение предназначено для предметной области «Аэропорт».

Рассматриваемая предметная область представляет собой ИС аэропорта, направленную на сбор и обработку информации для предоставления услуг авиаперевозок. ИС аэропорта связывает поставщика услуг – аэропорт и их потребителей – пассажиров. Для оптимального взаимодействия этих сторон необходимо автоматизировать информационные процессы авиаоператора, это приведет к улучшению качества и повышению быстродействия работы аэропорта. БД должна решать довольно узкий круг задач, таких как:

- Обслуживание клиентов, то есть:
 - Возможность покупки билетов на рейс
 - Возможность бронирования мест на рейс
 - Возможность отслеживания билетов и броней
 - Разработка комфортного для пользователя интерфейса
- Доступ к заданным данным и возможность их редактирования сотрудниками и администратором аэропорта.
- Разработка функционала для возможности регулирования деятельности аэропорта в зависимости от внешних факторов (погодных условий, человеческих факторов и другие).
- Изменение необходимой информации по мере её поступления в систему аэропорта.

Цель работы

Цель работы – создать оптимальную модель предметной области, в качестве которой выступает система отношений между аэропортом, сотрудниками и клиентами аэропорта.

Описание работы приложения

При запуске приложения запускается стартовое окно, которое позволит пользователю либо зарегистрироваться, либо войти в уже созданный аккаунт. После выполнения аутентификации пользователя запускается окно пользователя, в котором клиент сможет провести покупку/бронь билета на рейс, сможет отследить уже купленные

билеты, узнать основную информацию о приложении и, при необходимости, изменить интерфейс клиентского приложения по своему желанию.

Использованные средства разработки и системные требования

Исходя из того, что большинство сотрудников аэропорта работает за персональным компьютером или ноутбуком, было принято решение о создании клиентского приложения на персональный компьютер.

Были проанализированы данные, необходимые для грамотного функционирования приложения такие как, информация о рейсах, информация о пассажирах, информация о билетах, возможности взаимосвязи данных, в результате чего был сделан вывод о необходимости и удобстве систематизации данных, что привело к выбору реляционной СУБД.

Краткое описание языковых и инструментальных средств, использованных для создания приложения

В качестве языка программирования базы данных был выбран объектно-ориентированный язык программирования C#. Среда программирования – Visual Studio 2019. База данных реализована с помощью использования элементов Visual C# Windows Forms.

Как было замечено ранее, в качестве СУБД была выбрана реляционная модель данных. По степени распределённости данная СУБД относится к классу локальной, по способу доступа к БД данная СУБД относится к классу клиент-серверная.

Модель данных

Первым этапом реализации базы данных – это создание ER-модели базы данных. На рис. 1 изображена данная модель. Рассмотрим её более подробно. Для начала введём следующие обозначения:

- Множества сущностей изображаются в виде прямоугольников.
- Множества отношений изображаются в виде ромбов.
- Если сущности участвуют в отношениях, они будут связаны линией.
- Атрибуты изображаются в виде овалов и связываются линией с одной сущностью.

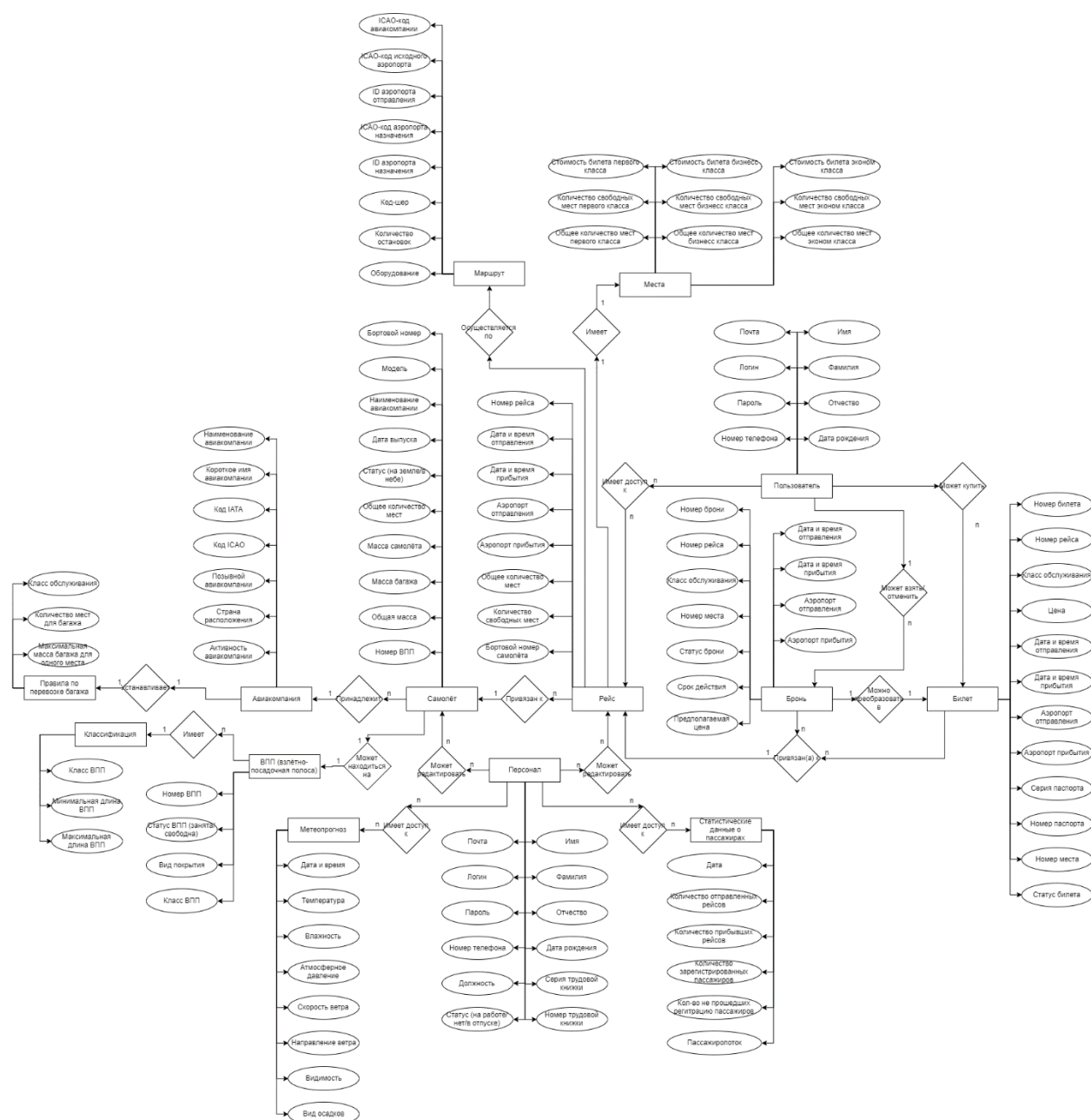


Рис. 1. ER-модель базы данных.

Всего на схеме изображено:

- 14 сущностей
- 16 отношений
- 109 атрибутов

Ключевыми сущностями в данной схеме являются «Пользователь», «Билет» и «Бронь», поскольку с помощью этих сущностей будет осуществляться взаимодействие клиента с системой.

Описание таблиц базы данных

Каждая из представленных на рис. 1 сущностей – это отдельная таблица в базе данных. Рассмотрим каждую из них поподробнее в виде таблицы.

№	Название таблицы	Назначение
1	Авиакомпания	Необходима для хранения данных об авиакомпании.
2	Билет	Одна из ключевых таблиц в БД. Необходима для хранения данных о билетах, купленных пользователями
3	Бронь	Одна из ключевых таблиц в БД. Необходима для хранения данных о бронях, созданных пользователями
4	ВПП	Необходима для хранения информации о ВПП
5	Классификация ВПП	Необходима для хранения данных о классах ВПП
6	Маршрут	Необходима для хранения данных о маршрутах.
7	Места	Необходима для просмотра пользователем информации о наличии мест определённого класса на выбранный рейс и о стоимости мест
8	Метеопрогноз	Необходима для сохранения информации о погодных условиях. Данная таблица будет необходима для принятия решений об изменении функционирования работы аэропорта

9	Персонал	Необходима для хранения данных о персонале. В дальнейшем таблицы «Персонал» и «Пользователь» будут объединены
10	Пользователь	Одна из ключевых таблиц в БД. Необходима для хранения данных о пользователе
11	Правила по перевозке багажа	Необходима для хранения дополнительных данных о возможностях перевозки багажа пассажиром
12	Рейс	Необходима для хранения данных о рейсах
13	Самолёт	Необходима для хранения данных о технических характеристиках самолёта
14	Статистические данные о пассажирах	Необходима для сбора и систематизации данных о пассажирах для дальнейшего принятия решения системой об изменениях в системе работы аэропорта

Таблица 1. Общий список таблиц в БД.

Описание структуры таблиц

Рассмотрим краткое описание структуры каждой таблицы.

№	Название поля	Тип данных	Ключ (PK/FK)
1	ID	INT	PK
2	Наименование	NVARCHAR	—
3	Короткое имя	NVARCHAR	—
4	Код IATA	NVARCHAR	—
5	Код ICAO	NVARCHAR	—
6	Позывной авиакомпании	NVARCHAR	—
7	Страна расположения	NVARCHAR	—
8	Активность авиакомпании	INT	—

9	ID правила по перевозке багажа	INT	FK
---	--------------------------------	-----	----

Таблица 2. Структура таблицы «Авиакомпания»

№	Название поля	Тип данных	Ключ (PK/FK)
1	Номер билета	INT	PK
2	Номер рейса	INT	—
3	Класс обслуживания	NVARCHAR	—
4	Цена	FLOAT	—
5	Дата и время отправления	DATETIME	—
6	Дата и время прибытия	DATETIME	—
7	Аэропорт отправления	NVARCHAR	—
8	Аэропорт прибытия	NVARCHAR	—
9	Серия паспорта	INT	—
10	Номер паспорта	INT	—
11	Номер места	INT	—
12	Статус билета	INT	FK

Таблица 3. Структура таблицы «Билет»

№	Название поля	Тип данных	Ключ (PK/FK)
1	Номер брони	INT	PK
2	Номер рейса	INT	—
3	Класс обслуживания	NVARCHAR	—
4	Цена	FLOAT	—
5	Дата и время отправления	DATETIME	—
6	Дата и время прибытия	DATETIME	—

7	Аэропорт отправления	NVARCHAR	–
8	Аэропорт прибытия	NVARCHAR	–
9	Номер места	INT	–
10	Статус брони	INT	FK
11	Срок действия	DATETIME	–

Таблица 4. Структура таблицы «Бронь»

№	Название поля	Тип данных	Ключ (PK/FK)
1	Номер	INT	PK
2	Статус	INT	–
3	Вид покрытия	NVARCHAR	–
4	Класс	INT	–

Таблица 5. Структура таблицы «ВПП»

№	Название поля	Тип данных	Ключ (PK/FK)
1	Класс	INT	PK
2	Минимальная длина	INT	–
3	Максимальная длина	INT	–

Таблица 6. Структура таблицы «Классификация ВПП»

№	Название поля	Тип данных	Ключ (PK/FK)
1	ID	INT	PK
2	ICAO-код авиакомпании	NVARCHAR	–
3	ICAO-код исходного аэропорта	NVARCHAR	–
4	ID аэропорта отправления	INT	–
5	ICAO-код аэропорта назначения	NVARCHAR	–

6	ID аэропорта назначения	INT	—
7	Код-шер	NVARCHAR	—
8	Количество остановок	INT	—
9	Оборудование	NVARCHAR	—

Таблица 7. Структура таблицы «Маршрут»

№	Название поля	Тип данных	Ключ (PK/FK)
1	ID	INT	PK
2	Общее количество мест первого класса	INT	—
3	Количество свободных мест первого класса	INT	—
4	Стоимость билета первого класса	FLOAT	—
5	Общее количество мест бизнес класса	INT	—
6	Количество свободных мест бизнес класса	INT	—
7	Стоимость билета бизнес класса	FLOAT	—
8	Общее количество мест эконом класса	INT	—
9	Количество свободных мест эконом класса	INT	—
10	Стоимость билета эконом класса	FLOAT	—

Таблица 8. Структура таблицы «Места»

№	Название поля	Тип данных	Ключ (PK/FK)
1	ID	INT	PK

2	Дата и время	DATETIME	—
3	Температура	FLOAT	—
4	Влажность	INT	—
5	Атмосферное давление	INT	—
6	Скорость ветра	FLOAT	—
7	Направление ветра	NVARCHAR	—
8	Видимость	INT	—
9	Вид осадков	NVARCHAR	—

Таблица 9. Структура таблицы «Метеопрогноз»

№	Название поля	Тип данных	Ключ (PK/FK)
1	ID	INT	PK
2	Почта	NVARCHAR	—
3	Логин	NVARCHAR	—
4	Пароль	NVARCHAR	—
5	Имя	NVARCHAR	—
6	Фамилия	NVARCHAR	—
7	Отчество	NVARCHAR	—
8	Дата рождения	DATE	—
9	Номер телефона	NVARCHAR	—
10	Роль	NVARCHAR	—

Таблица 10. Структура таблицы «Пользователь»

№	Название поля	Тип данных	Ключ (PK/FK)
1	ID	INT	PK

2	Класс обслуживания	NVARCHAR	—
3	Количество мест для багажа	INT	—
4	Максимальная масса багажа для одного места	INT	—

Таблица 11. Структура таблицы «Правила по перевозке багажа»

№	Название поля	Тип данных	Ключ (PK/FK)
1	Номер рейса	INT	PK
2	Дата и время отправления	DATETIME	—
3	Дата и время прибытия	DATETIME	—
4	Аэропорт отправления	NVARCHAR	—
5	Аэропорт прибытия	NVARCHAR	—
6	Общее количество мест	INT	—
7	Количество свободных мест	INT	—
8	ID самолёта	INT	FK
9	ID маршрута	INT	FK
10	ID таблицы мест	INT	FK

Таблица 12. Структура таблицы «Рейс»

№	Название поля	Тип данных	Ключ (PK/FK)
1	ID	INT	PK
2	Бортовой номер	NVARCHAR	—
3	Модель	NVARCHAR	—
4	ID авиакомпании	INT	—
5	Дата выпуска	DATE	—
6	Статус самолёта	INT	—

7	Общее количество мест	INT	—
8	Масса самолёта	FLOAT	—
9	Масса багажа	FLOAT	—
10	Общая масса	FLOAT	—
11	Номер ВПП	INT	—

Таблица 13. Структура таблицы «Самолёт»

№	Название поля	Тип данных	Ключ (PK/FK)
1	ID	INT	PK
2	Дата	DATE	—
3	Количество отправленных рейсов	INT	—
4	Количество прибывших рейсов	INT	—
5	Количество зарегистрированных пассажиров	INT	—
6	Количество не прошедших регистрацию пассажиров	INT	—
7	Пассажиропоток	INT	—

Таблица 14. Структура таблицы «Статистические данные о пассажирах»

Описание клиентского приложения

Общие сведения

Данное клиентское приложение направлено на клиентов (пользователей), сотрудников и администраторов аэропорта.

Со стороны клиента приложение реализует минимальный функционал для продажи билетов на рейсы, а также предоставляет возможность бронирования мест на выбранный рейс. Сотрудники аэропорта имеют ограниченный доступ к данным, которые могут непосредственно влиять на работу аэропорта. Данные предоставляются для сотрудников, чтобы последние имели возможность своевременно реагировать на непредвиденные обстоятельства, либо на обстоятельства, которые могут негативным образом отразиться на работе аэропорта. Администраторы имеют возможность вносить изменения в любые таблицы, а также имеют возможность добавлять в базу данных пользователей с любыми правами («пользователь», «сотрудник», «администратор»).

Список классов

Каждый из представленных ниже классов представляет отдельную форму (окно) со своим функционалом и предназначением. Рассмотрим подробнее каждый из классов.

- **Admin form** (форма администратора). Данная форма активируется, если зайти в клиентское приложение от роли администратора. Данная форма предоставляет возможности для работы с каждой из таблиц в БД как с помощью SQL запросов, так и с помощью элемента управления DataGridView, позволяющего отображать данные в табличном формате, в настраиваемой сетке.

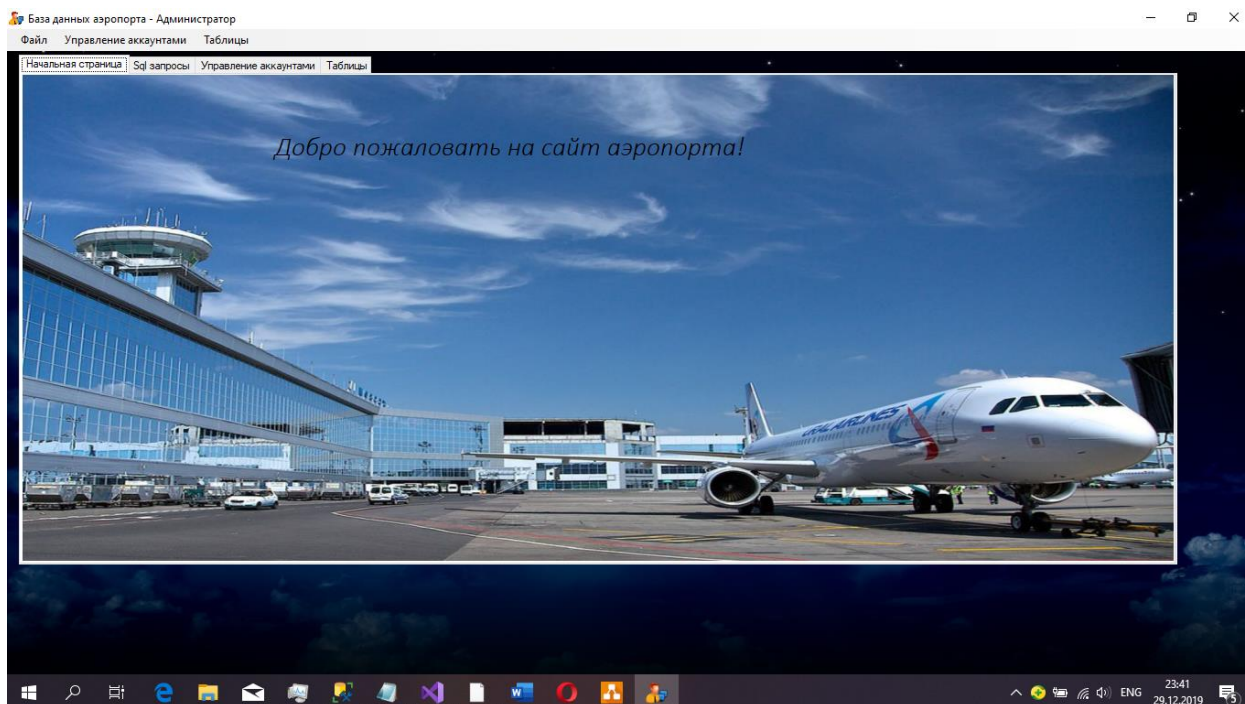


Рис 2. Форма администратора.

- **Authorization form** (форма авторизации). Данная форма активируется, если пользователь выберет пункт «Войти» на стартовой форме. Данная форма предоставляет необходимый функционал для авторизации пользователя в системе.

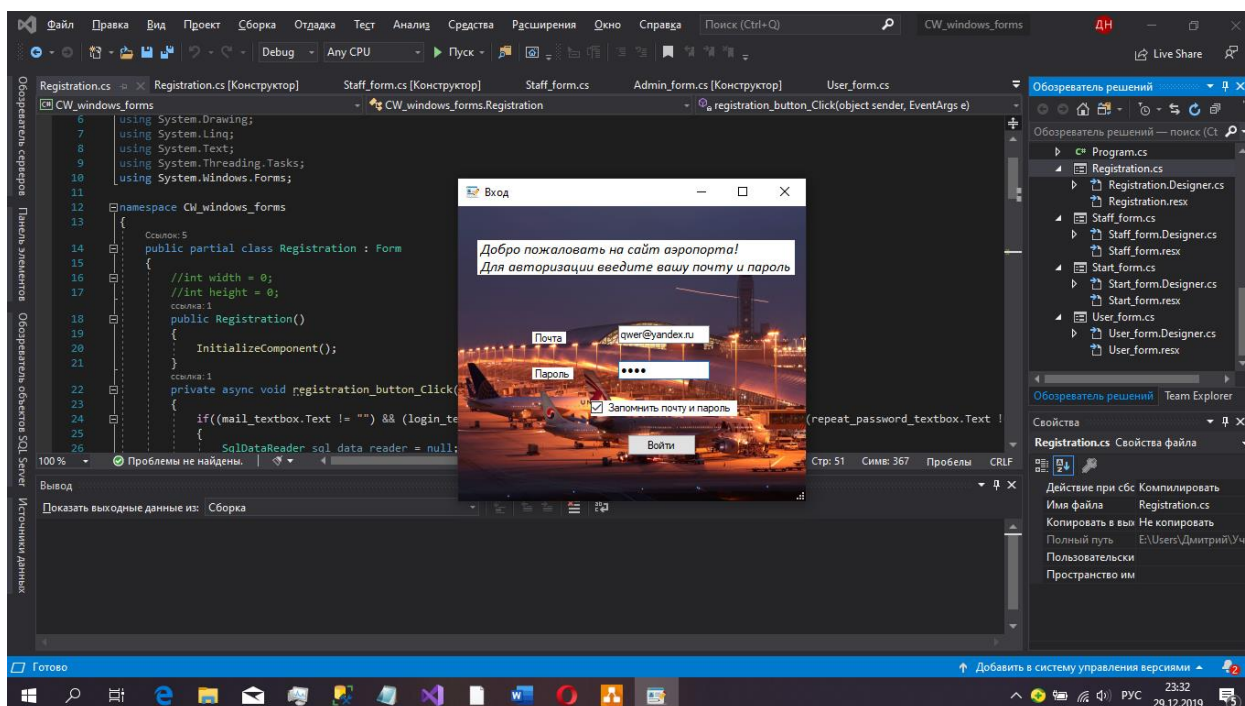


Рис 3. Форма авторизации.

- **Passport data form** (форма паспортных данных). Данная форма активируется, если пользователь в пользовательской форме выберет опцию покупки билета. В таком случае пользователю будет предложено ввести паспортные данные. Данная форма предоставляет пользователю возможность ввода паспортных данных для завершения процедуры покупки билета.

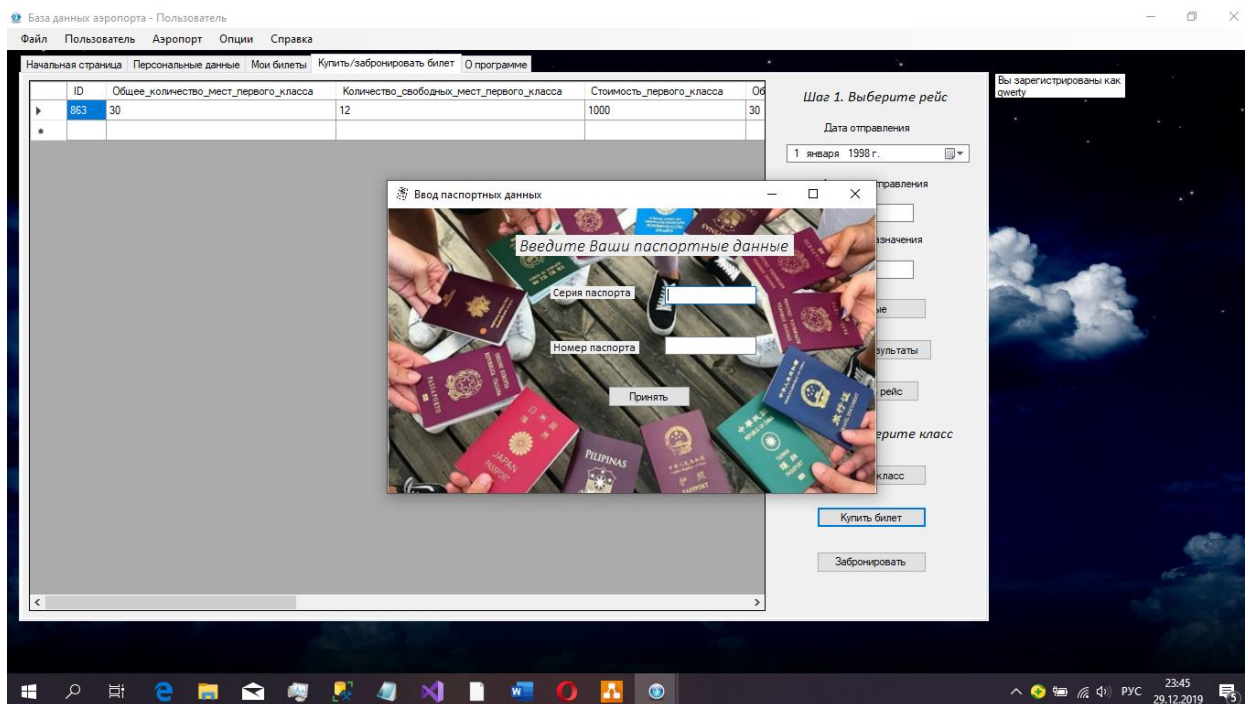


Рис 4. Форма паспортных данных.

- **Registration form** (форма регистрации). Данная форма активируется, если пользователь выберет пункт «Зарегистрироваться» на стартовой форме. Данная форма предоставляет необходимый функционал для регистрации пользователя. В данной форме можно зарегистрировать только пользователя с ролью «Пользователь».

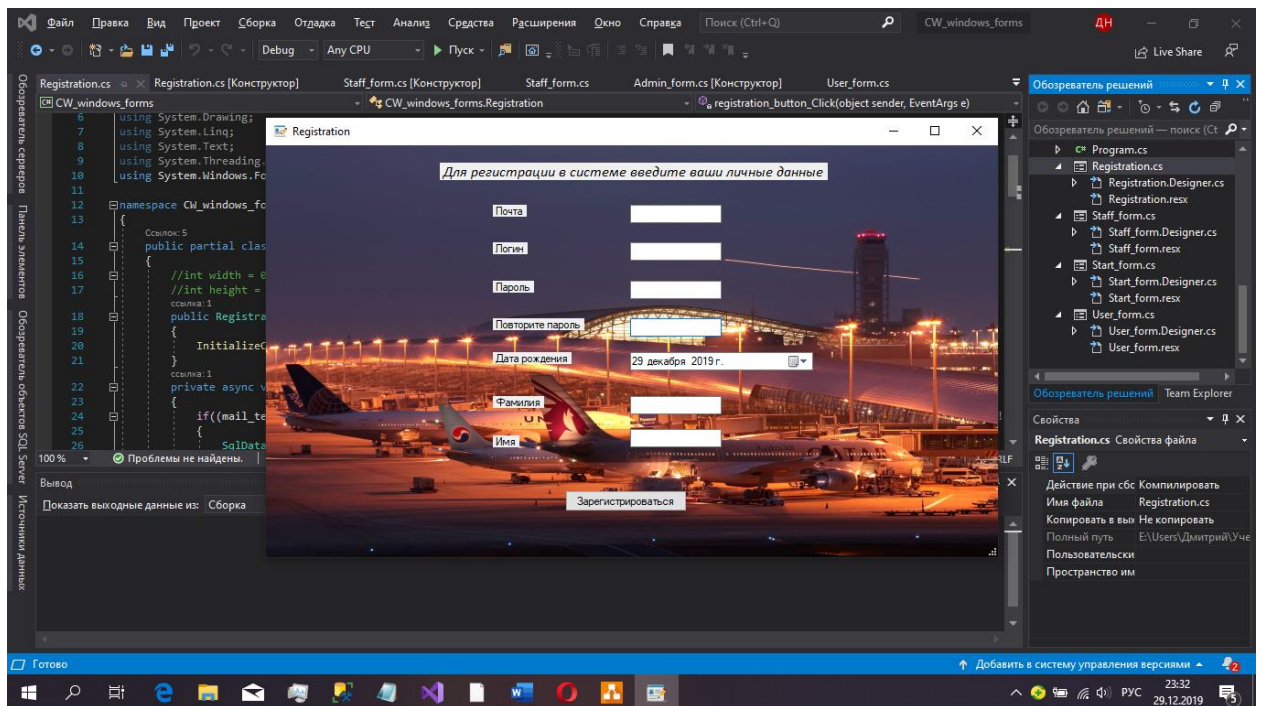


Рис. 5. Форма регистрации.

- **Staff form** (форма для персонала). Данная форма активируется, если зайти в клиентское приложение от роли персонала. Данная форма предоставляет возможности для работы персонала с ограниченным количеством таблиц для поддержания работы аэропорта в стабильном состоянии.

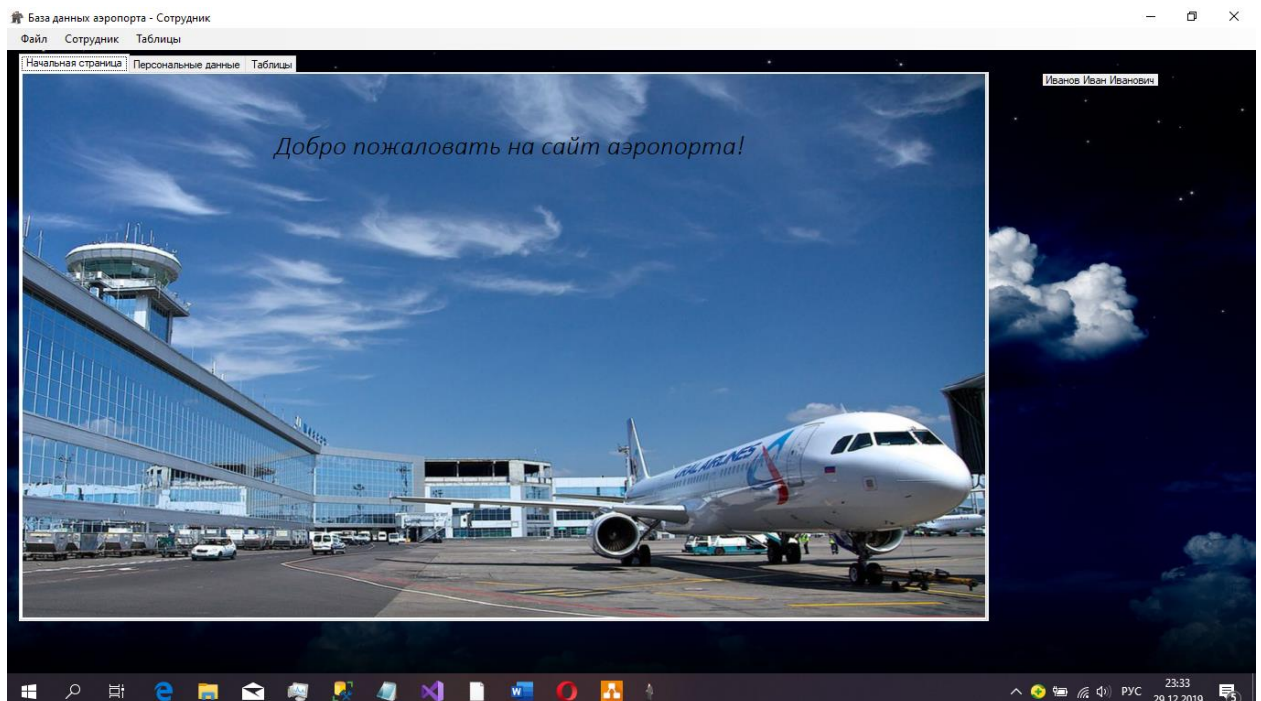


Рис. 6. Форма для персонала.

- **Start form** (начальная форма). Данная форма активируется при запуске приложения. Данная форма предоставляет пользователю возможность либо пройти авторизацию, либо пройти процесс регистрации.

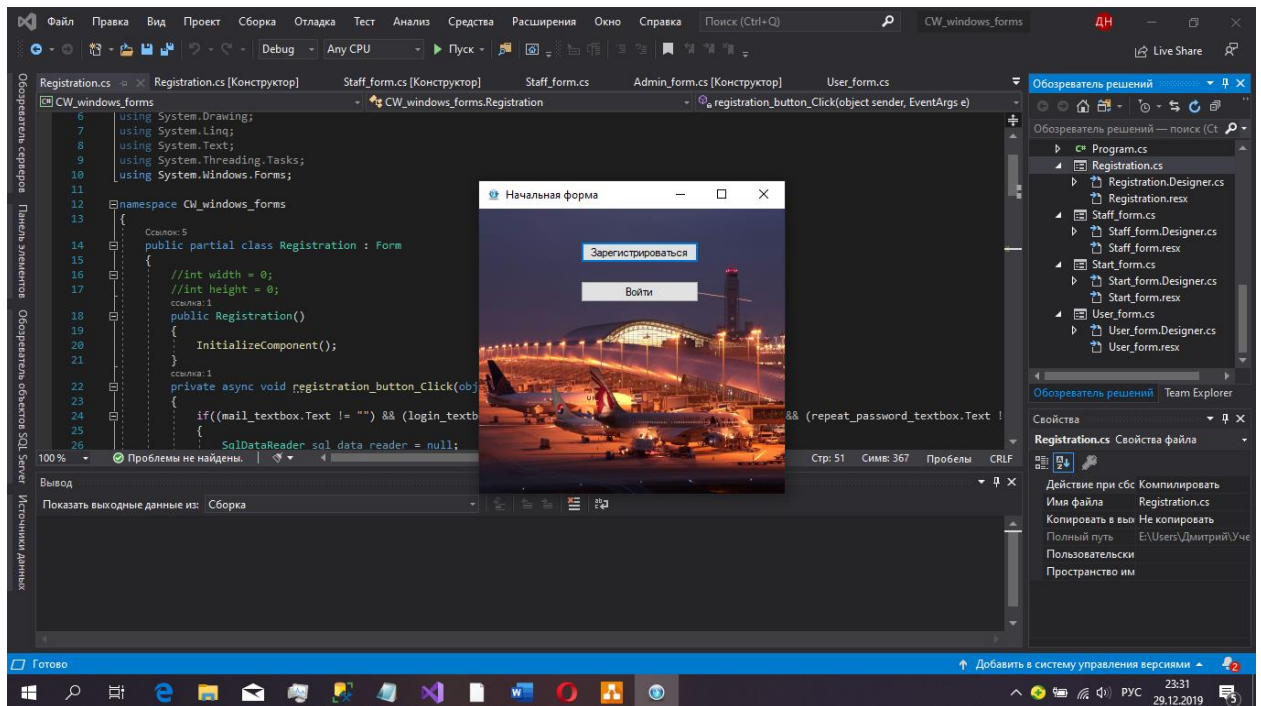


Рис. 7. Начальная форма.

- **User form** (форма пользователя). Данная форма активируется, если зайти в клиентское приложение от роли пользователя. Данная форма предоставляет пользователю возможности для покупки и бронирования билета, просмотра уже имеющихся билетов и броней, а также возможности для изменения интерфейса приложения в соответствии с потребностями пользователя.

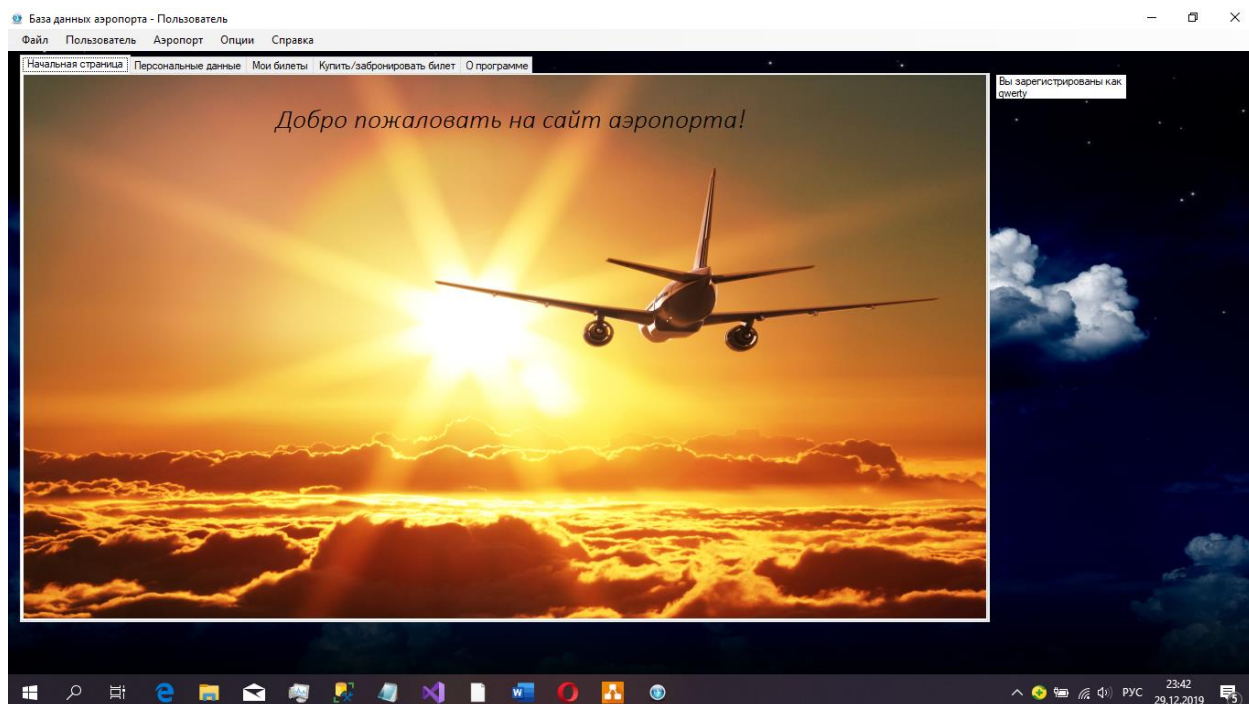


Рис. 8. Форма пользователя.

Организация ролевого доступа к данным

В клиентском приложении предусмотрено выделение следующих трёх ролей:

- Пользователь
- Сотрудник
- Администратор

Рассмотрим назначение и систему привилегий, закреплённой за ролью более подробно.

Пользователь, после входа в свой аккаунт имеет возможность просматривать только свои билеты и брони из соответствующих таблиц. Кроме того, пользователь имеет возможность изменить некоторые данные своего аккаунта (фамилия, имя, отчество, логин, пароль, номер телефона), однако, для изменения пароля будет необходимо ввести старый пароль, чтобы увеличить уровень безопасности данных клиента. Также пользователь имеет возможность купить билет на рейс или забронировать место в соответствии с наличием свободных мест, то есть, если свободные места на рейс отсутствуют, то пользователь не сможет забронировать место или купить билет.

Сотрудник, после входа в свой аккаунт имеет возможность просматривать (но не изменять) следующие таблицы:

- Авиакомпания
- Активность авиакомпании
- ВПП
- Классификация ВПП
- Маршрут
- Места
- Метеопрогноз
- Правила по перевозке багажа
- Рейс
- Самолёт
- Статистические данные о пассажирах
- Статус билета
- Статус брони
- Статус самолёта

Отсутствие возможности редактировать данные обусловлено тем предположением, что база данных должна автоматически собирать необходимые данные (например, автоматически обновляться данные в таблице «Метеопрогноз» через подключение через интернет к одной из крупных баз данных погоды), а сотрудник должен анализировать данные и, только в случае необходимости направить администратору просьбу о необходимости отредактировать те или иные данные.

Администратор, после входа в свой аккаунт имеет возможность просматривать и редактировать данные в любой из таблиц БД. Администратор может работать с табличными данными как с помощью sql команд, так и с помощью упрощённого графического интерфейса для работы с данными.

Тематические запросы к данным

Рассмотрим основные тематические запросы к данным, которые в большей степени отражают процесс функционирования БД.

Данный запрос необходим для вывода на экран билетов, которые закреплены за пользователем:

```
SELECT Билет.Номер_билета, Билет.Номер_рейса, Билет.Класс_обслуживания,  
Билет.Цена, Билет.Дата_время_отправления, Билет.Дата_время_прибытия,  
Билет.Аэропорт_отправления, Билет.Аэропорт_прибытия, Билет.Номер_места,  
Билет.Статус_билета  
FROM Билет, Билет_Пользователь  
WHERE Билет_Пользователь.ID_пользователя = 1 and Билет.Номер_билета =  
Билет_Пользователь.Номер_билета
```

Следующий sql запрос реализуется для выбора подходящих для пользователя рейсов:

```
SELECT Номер_рейса, Дата_время_отправления, Дата_время_прибытия,  
Аэропорт_отправления, Аэропорт_прибытия, Общее_количество_мест,  
Количество_свободных_мест  
FROM Рейс  
WHERE (Рейс.Аэропорт_отправления = 'Москва') and (Рейс.Аэропорт_прибытия =  
'Париж') and  
(DAY(Рейс.Дата_время_отправления) = 1) and  
(MONTH(Рейс.Дата_время_отправления) = 1) and  
(YEAR(Рейс.Дата_время_отправления) = 1998)
```

Нижеприведённая транзакция реализуется при покупке пользователем билета:

```
BEGIN TRANSACTION  
INSERT INTO Билет (Номер_рейса, Класс_обслуживания, Цена,  
Дата_время_отправления, Дата_время_прибытия, Аэропорт_отправления,  
Аэропорт_прибытия, Серия_паспорта, Номер_паспорта, Номер_места,  
Статус_билета)  
VALUES (1, 1, '', '', '', '', '', '', '', '', '')  
  
-- Уменьшаем количество свободных мест в таблице с рейсами  
UPDATE Рейс
```

```

SET Количество_свободных_мест = Количество_свободных_мест - 1
WHERE Номер_рейса = 1

-- Уменьшаем количество свободных мест по классу в таблице с местами
UPDATE Места
SET Количество_свободных_мест_первого_класса =
Количество_свободных_мест_первого_класса - 1
WHERE ID = (SELECT ID_таблицы_мест FROM Рейс WHERE Номер_рейса = 1)

DECLARE @my_ticket INT
SET @my_ticket = (SELECT Билет.Номер_билета FROM Билет WHERE
(Номер_рейса = 1 and Класс_обслуживания = 'FIRST' and Номер_места = 1))

INSERT INTO Билет_Пользователь (Номер_билета, ID_пользователя)
VALUES (@my_ticket, '')
COMMIT

```

Выводы

В рамках данной курсовой работы был выполнен системный анализ предметной области и её изучение. Была разработана даталогическая и ER-модель базы данных. Исходя из данных моделей была разработана база данных, отражающая наиболее значимые факторы предметной области. Полученная база данных полностью удовлетворяет поставленным целям данной курсовой работы. Работа по реализации базы данных полностью выполнена.

Список используемых источников

1. https://otherreferats.allbest.ru/transport/00335587_0.html – описание предметной области аэропорта.
2. <https://www.draw.io> – удобное приложение для разработки ER-диаграммы.
3. <https://icon-icons.com> – сайт, на котором можно найти тысячи разных иконок.

Приложения

Скрипт базы данных

```
USE [master]

GO

/***** Object: Database [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF]  Script Date: 30.12.2019 2:11:53
*****/

CREATE DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'AirportDB_2', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\Copy_DB\AirportDB_2.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED,
FILEGROWTH = 65536KB )

LOG ON

( NAME = N'AirportDB_2_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\Copy_DB\AirportDB_2_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB ,
FILEGROWTH = 65536KB )

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET COMPATIBILITY_LEVEL = 140

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF].[dbo].[sp_fulltext_database]
@action = 'enable'
end

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET ANSI_NULL_DEFAULT OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET ANSI_NULLS OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET ANSI_PADDING OFF

GO
```

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET ANSI_WARNINGS OFF
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET ARITHABORT OFF
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET AUTO_CLOSE ON
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET AUTO_SHRINK OFF
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET AUTO_UPDATE_STATISTICS ON
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET CURSOR_CLOSE_ON_COMMIT  
OFF
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET CURSOR_DEFAULT GLOBAL
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET CONCAT_NULL_YIELDS_NULL  
OFF
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET NUMERIC_ROUNDABORT OFF
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET QUOTED_IDENTIFIER OFF
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET RECURSIVE_TRIGGERS OFF
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET DISABLE_BROKER
```

GO

```

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET
AUTO_UPDATE_STATISTICS_ASYNC OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET
DATE_CORRELATION_OPTIMIZATION OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET ALLOW_SNAPSHOT_ISOLATION
OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET READ_COMMITTED_SNAPSHOT
OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET HONOR_BROKER_PRIORITY OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET RECOVERY SIMPLE

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET MULTI_USER

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET PAGE_VERIFY CHECKSUM

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET DB_CHAINING OFF

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET FILESTREAM(
NON_TRANSACTED_ACCESS = OFF )

```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET TARGET_RECOVERY_TIME = 60  
SECONDS
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET DELAYED_DURABILITY =  
DISABLED
```

GO

```
ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET QUERY_STORE = OFF
```

GO

```
USE [C:\PROGRAM FILES\MICROSOFT SQL  
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF]
```

GO

/***** Object: Table [dbo].[Авиакомпания] Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[Авиакомпания](  
    [ID] [int] NOT NULL,  
    [Наименование] [nvarchar](max) NOT NULL,  
    [Короткое_имя] [nvarchar](max) NOT NULL,  
    [Код_IATA] [nvarchar](max) NOT NULL,  
    [Код_ICAO] [nvarchar](max) NOT NULL,  
    [Позывной] [nvarchar](max) NOT NULL,  
    [Страна_расположения] [nvarchar](max) NOT NULL,  
    [Активность] [int] NOT NULL,  
    [ID_правила_по_перевозке_багажа] [int] NOT NULL,  
    CONSTRAINT [PK_Авиакомпания] PRIMARY KEY CLUSTERED  
(  
    [ID] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =  
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```


GO

/***** Object: Table [dbo].[Активность_авиакомпаний] Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Активность_авиакомпаний](

[Активность] [int] NOT NULL,

[Значение] [nvarchar](max) NOT NULL

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Билет] Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Билет](

[Номер_билета] [int] IDENTITY(1,1) NOT NULL,

[Номер_рейса] [int] NOT NULL,

[Класс_обслуживания] [nvarchar](max) NOT NULL,

[Цена] [float] NOT NULL,

[Дата_время_отправления] [datetime] NOT NULL,

[Дата_время_прибытия] [datetime] NOT NULL,

[Аэропорт_отправления] [nvarchar](max) NOT NULL,

[Аэропорт_прибытия] [nvarchar](max) NOT NULL,

[Серия_паспорта] [int] NOT NULL,

[Номер_паспорта] [int] NOT NULL,

[Номер_места] [int] NOT NULL,

[Статус_билета] [int] NOT NULL,

CONSTRAINT [PK_Билет] PRIMARY KEY CLUSTERED

(

[Номер_билета] ASC

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Билет_Пользователь]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Билет_Пользователь](
    [Номер_билета] [int] NOT NULL,
    [ID_пользователя] [int] NOT NULL,
    CONSTRAINT [PK_Билет_Пользователь] PRIMARY KEY CLUSTERED
(
    [Номер_билета] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[Бронь]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Бронь](
    [Номер_брони] [int] IDENTITY(1,1) NOT NULL,
    [Номер_рейса] [int] NOT NULL,
    [Класс_обслуживания] [nvarchar](max) NOT NULL,
    [Цена] [float] NOT NULL,
    [Дата_время_отправления] [datetime] NOT NULL,
    [Дата_время_прибытия] [datetime] NOT NULL,
    [Аэропорт_отправления] [nvarchar](max) NOT NULL,
    [Аэропорт_прибытия] [nvarchar](max) NOT NULL,

```

```

        [Номер_места] [int] NOT NULL,

        [Статус_брони] [int] NOT NULL,

        [Срок_действия] [datetime] NULL,

CONSTRAINT [PK_Бронь] PRIMARY KEY CLUSTERED
(
        [Номер_брони] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Бронь_Пользователь]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Бронь_Пользователь](
        [Номер_брони] [int] NOT NULL,

        [ID_пользователя] [int] NOT NULL,

CONSTRAINT [PK_Бронь_Пользователь] PRIMARY KEY CLUSTERED
(
        [Номер_брони] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[ВПП]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ВПП](
        [Номер] [int] NOT NULL,

        [Статус] [int] NOT NULL,

```

```

        [Вид_покрытия] [nvarchar](max) NOT NULL,

        [Класс] [int] NOT NULL,

CONSTRAINT [PK_ВПП] PRIMARY KEY CLUSTERED

(

        [Номер] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Классификация_ВПП]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Классификация_ВПП](

        [Класс] [int] NOT NULL,

        [Минимальная_длина] [int] NOT NULL,

        [Максимальная_длина] [int] NOT NULL,

CONSTRAINT [PK_Классификация] PRIMARY KEY CLUSTERED

(

        [Класс] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[Маршрут]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Маршрут](

        [ID] [int] NOT NULL,

        [ICAO-код_авиакомпаний] [nvarchar](max) NOT NULL,

```

```

[ICAO-код_исходного_аэропорта] [nvarchar](max) NOT NULL,
[ID_аэропорта_отправления] [int] NOT NULL,
[ICAO-код_аэропорта_назначения] [nvarchar](max) NOT NULL,
[ID_аэропорта_назначения] [int] NOT NULL,
[Код-шер] [nvarchar](max) NOT NULL,
[Количество_остановок] [int] NOT NULL,
[Оборудование] [nvarchar](max) NOT NULL,
CONSTRAINT [PK_Маршрут] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

/***** Object: Table [dbo].[Места]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Места](
    [ID] [int] NOT NULL,
    [Общее_количество_мест_первого_класса] [int] NOT NULL,
    [Количество_свободных_мест_первого_класса] [int] NOT NULL,
    [Стоимость_первого_класса] [float] NOT NULL,
    [Общее_количество_мест_бизнес_класса] [int] NOT NULL,
    [Количество_свободных_мест_бизнес_класса] [int] NOT NULL,
    [Стоимость_бизнес_класса] [float] NOT NULL,
    [Общее_количество_мест_эконом_класса] [int] NOT NULL,
    [Количество_свободных_мест_эконом_класса] [int] NOT NULL,
    [Стоимость_эконом_класса] [float] NOT NULL,
    CONSTRAINT [PK_Места] PRIMARY KEY CLUSTERED
(
    [ID] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[Метеопрогноз]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Метеопрогноз](
    [ID] [int] NOT NULL,
    [Дата_время] [datetime] NOT NULL,
    [Температура] [float] NOT NULL,
    [Влажность] [int] NOT NULL,
    [Атмосферное_давление] [int] NOT NULL,
    [Скорость_ветра] [float] NOT NULL,
    [Направление_ветра] [nvarchar](max) NOT NULL,
    [Видимость] [int] NOT NULL,
    [Вид_осадков] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Метеопрогноз] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Пользователь]  Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Пользователь](
    [ID] [int] IDENTITY(1,1) NOT NULL,

```

```

        [Почта] [nvarchar](max) NOT NULL,
        [Логин] [nvarchar](max) NOT NULL,
        [Пароль] [nvarchar](max) NOT NULL,
        [Имя] [nvarchar](max) NOT NULL,
        [Фамилия] [nvarchar](max) NOT NULL,
        [Отчество] [nvarchar](max) NULL,
        [Дата_рождения] [date] NOT NULL,
        [Номер_телефона] [nvarchar](max) NULL,
        [Роль] [nvarchar](max) NOT NULL,
CONSTRAINT [PK_Пользователь] PRIMARY KEY CLUSTERED
(
        [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

/***** Object: Table [dbo].[Правила_по_перевозке_багажа]  Script Date: 30.12.2019 2:11:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Правила_по_перевозке_багажа](
        [ID] [int] NOT NULL,
        [Класс_обслуживания] [nvarchar](max) NOT NULL,
        [Количество_мест_для_багажа] [int] NOT NULL,
        [Максимальная_масса_багажа_для_одного_места] [int] NOT NULL,
CONSTRAINT [PK_Правила_по_перевозке_багажа] PRIMARY KEY CLUSTERED
(
        [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

```

/***** Object: Table [dbo].[Рейс] Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Рейс](

[Номер_рейса] [int] NOT NULL,

[Дата_время_отправления] [datetime] NOT NULL,

[Дата_время_прибытия] [datetime] NOT NULL,

[Аэропорт_отправления] [nvarchar](max) NOT NULL,

[Аэропорт_прибытия] [nvarchar](max) NOT NULL,

[Общее_количество_мест] [int] NOT NULL,

[Количество_свободных_мест] [int] NOT NULL,

[ID_самолёта] [int] NOT NULL,

[ID_маршрута] [int] NOT NULL,

[ID_таблицы_мест] [int] NOT NULL,

CONSTRAINT [PK_Рейс] PRIMARY KEY CLUSTERED

(

[Номер_рейса] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Самолёт] Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Самолёт](

[ID] [int] NOT NULL,

[Бортовой_номер] [nvarchar](max) NOT NULL,

[Модель] [nvarchar](max) NOT NULL,

[ID_авиакомпании] [int] NOT NULL,


```

[Дата_выпуска] [date] NOT NULL,

[Статус_самолёта] [int] NOT NULL,

[Общее_количество_мест] [int] NOT NULL,

[Масса_самолёта] [float] NOT NULL,

[Масса_багажа] [float] NOT NULL,

[Общая_масса] [float] NOT NULL,

[Номер_ВПП] [int] NOT NULL,

CONSTRAINT [PK_Самолёт] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

/***** Object: Table [dbo].[Статистические_данные_о_пассажирах]  Script Date: 30.12.2019 2:11:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Статистические_данные_о_пассажирах](
    [ID] [int] NOT NULL,
    [Дата] [date] NOT NULL,
    [Количество_отправленных_рейсов] [int] NOT NULL,
    [Количество_прибывших_рейсов] [int] NOT NULL,
    [Количество_зарегистрированных_пассажиров] [int] NOT NULL,
    [Количество_не_прошедших_регистрацию_пассажиров] [int] NOT NULL,
    [Пассажиропоток] [int] NOT NULL,
    CONSTRAINT [PK_Статистические_данные_о_пассажирах] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

/***** Object: Table [dbo].[Статус_билета] Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Статус_билета](

[Статус_билета] [int] NOT NULL,

[Значение] [nvarchar](max) NOT NULL,

CONSTRAINT [PK_Статус_билета] PRIMARY KEY CLUSTERED

(

[Статус_билета] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Статус_брони] Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Статус_брони](

[Статус_брони] [int] NOT NULL,

[Значение] [nvarchar](max) NOT NULL,

CONSTRAINT [PK_Статус_брони] PRIMARY KEY CLUSTERED

(

[Статус_брони] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Статус_самолёта] Script Date: 30.12.2019 2:11:54 *****/

SET ANSI_NULLS ON

```

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Статус_самолёта](
    [Статус_самолёта] [int] NOT NULL,
    [Значение] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Статус_самолёта] PRIMARY KEY CLUSTERED
(
    [Статус_самолёта] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

ALTER TABLE [dbo].[Билет] WITH CHECK ADD CONSTRAINT [FK_Билет_Билет_Пользователь] FOREIGN
KEY([Номер_билета])
REFERENCES [dbo].[Билет_Пользователь] ([Номер_билета])

GO

ALTER TABLE [dbo].[Билет] CHECK CONSTRAINT [FK_Билет_Билет_Пользователь]

GO

ALTER TABLE [dbo].[Билет] WITH CHECK ADD CONSTRAINT [FK_Билет_Рейс] FOREIGN KEY([Номер_рейса])
REFERENCES [dbo].[Рейс] ([Номер_рейса])

GO

ALTER TABLE [dbo].[Билет] CHECK CONSTRAINT [FK_Билет_Рейс]

GO

ALTER TABLE [dbo].[Билет] WITH CHECK ADD CONSTRAINT [FK_Билет_Статус_билета] FOREIGN
KEY([Статус_билета])
REFERENCES [dbo].[Статус_билета] ([Статус_билета])

GO

ALTER TABLE [dbo].[Билет] CHECK CONSTRAINT [FK_Билет_Статус_билета]

GO

ALTER TABLE [dbo].[Билет_Пользователь] WITH CHECK ADD CONSTRAINT
[FK_Билет_Пользователь_Пользователь] FOREIGN KEY([ID_пользователя])
REFERENCES [dbo].[Пользователь] ([ID])

GO

```

```

ALTER TABLE [dbo].[Билет_Пользователь] CHECK CONSTRAINT [FK_Билет_Пользователь_Пользователь]

GO

ALTER TABLE [dbo].[Бронь] WITH CHECK ADD CONSTRAINT [FK_Бронь_Бронь_Пользователь] FOREIGN
KEY([Номер_брони])

REFERENCES [dbo].[Бронь_Пользователь] ([Номер_брони])

GO

ALTER TABLE [dbo].[Бронь] CHECK CONSTRAINT [FK_Бронь_Бронь_Пользователь]

GO

ALTER TABLE [dbo].[Бронь] WITH CHECK ADD CONSTRAINT [FK_Бронь_Рейс] FOREIGN KEY([Номер_рейса])

REFERENCES [dbo].[Рейс] ([Номер_рейса])

GO

ALTER TABLE [dbo].[Бронь] CHECK CONSTRAINT [FK_Бронь_Рейс]

GO

ALTER TABLE [dbo].[Бронь] WITH CHECK ADD CONSTRAINT [FK_Бронь_Статус_брони] FOREIGN
KEY([Статус_брони])

REFERENCES [dbo].[Статус_брони] ([Статус_брони])

GO

ALTER TABLE [dbo].[Бронь] CHECK CONSTRAINT [FK_Бронь_Статус_брони]

GO

ALTER TABLE [dbo].[Бронь_Пользователь] WITH CHECK ADD CONSTRAINT
[FK_Бронь_Пользователь_Пользователь] FOREIGN KEY([ID_пользователя])

REFERENCES [dbo].[Пользователь] ([ID])

GO

ALTER TABLE [dbo].[Бронь_Пользователь] CHECK CONSTRAINT [FK_Бронь_Пользователь_Пользователь]

GO

USE [master]

GO

ALTER DATABASE [C:\PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL14.SQLEXPRESS\MSSQL\DATA\COPY_DB\AIRPORTDB_2.MDF] SET READ_WRITE

GO

```

Пример программного кода клиентского приложения

В качестве примера программного кода клиентского приложения приведена функция, реализующая первую стадию покупки билета на рейс пользователем.

```
private void buy_ticket_button_Click(object sender, EventArgs e)
```

```

{
    try
    {
        if (value_table != "Места")
        {
            MessageBox.Show("Сначала выберите рейс", "Уведомление",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        else if (choose_class_button.Text == "Выберите класс")
        {
            MessageBox.Show("Сначала выберите класс", "Предупреждение",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        else
        {
            int selected_row_index = airport_dataGridView.CurrentRow.Index;
            if (((choose_class_button.Text == "Первый класс") &&
                (Convert.ToInt32(airport_dataGridView[2, selected_row_index].Value) == 0)) ||
                ((choose_class_button.Text == "Бизнес класс") &&
                (Convert.ToInt32(airport_dataGridView[5, selected_row_index].Value) == 0)) ||
                ((choose_class_button.Text == "Эконом класс") &&
                (Convert.ToInt32(airport_dataGridView[8, selected_row_index].Value) == 0)))
            {
                MessageBox.Show("Свободных мест нет", "Уведомление",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            else
            {
                My_values.passport_form = new Passport_data_form();
                My_values.passport_form.Show();
                My_values.passport_form.Activate();
                // Поставить дальнейшее выполнение программы на ожидание
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(),
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```