Оглавление

[**Введение** 2](#_Toc85150529)

[**Описание предметной области** 3](#_Toc85150530)

[**Схема БД** 5](#_Toc85150531)

[**Описание таблиц** 5](#_Toc85150532)

[**Код создания таблиц** 6](#_Toc85150533)

[**Запросы** 7](#_Toc85150534)

[**Представления** 7](#_Toc85150535)

[**Вывод** 8](#_Toc85150536)

[**Добавление** 8](#_Toc85150537)

[**Изменение** 9](#_Toc85150538)

[**Удаление** 10](#_Toc85150539)

[**Сложные запросы** 10](#_Toc85150540)

[**ХП** 12](#_Toc85150541)

[**Триггеры** 13](#_Toc85150542)

[**Клиентская часть** 15](#_Toc85150543)

[**Окно входа** 16](#_Toc85150544)

[**Окно кассира** 17](#_Toc85150545)

[**Окно сотрудника** 18](#_Toc85150546)

[**Окно пилота** 21](#_Toc85150547)

[**Окно администратора** 21](#_Toc85150548)

[**Заключение** 25](#_Toc85150549)

[**Список литературы** 26](#_Toc85150550)

# **Введение**

В настоящий момент нет ни одного большого предприятия, которое не использовало бы программы(софты) для облегчения работы и ее автоматизации.

В данной курсовой работе мне нужно было разработать многопользовательскую автоматизированную систему управления организацией. В моем случае объект автоматизации – служба аэропорта.

Было разработано приложение на языке программирования C#, использующее информацию, хранящуюся в SQL-базе.

# **Описание предметной области**

Группы пользователей

|  |  |
| --- | --- |
| № пп | Наименование пользователя |
| 1 | Кассир |
| 2 | Сотрудник |
| 3 | Пилот |
| 4 | Администратор |

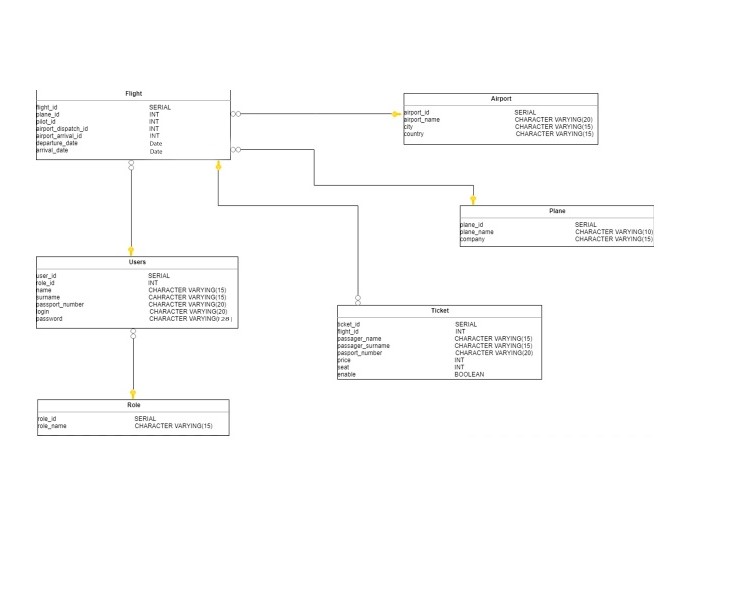
Функции групп пользователей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № пп | Выполняемая функция | Входные данные | Выходные данные | Функции, которые должны быть реализованы в ИС |
| **Кассир** | | | | |
| 1 | Просмотр количества свободных билетов на рейс | Id рейса | Число свободных билетов | Вывод числа свободных билетов |
| 2 | Продать билет | Id рейса |  | Уменьшение числа свободных билетов |
| 3 | Просмотр стоимости билета | Id рейса | Стоимость билета | Вывод стоимости билета |
| 4 | Принять возврат билета | Данные билета |  | Добавить этот билет снова в продажу |
| **Сотрудник** | | | | |
| 1 | Просмотр всех рейсов |  | Список рейсов | Вывод списка рейсов |
| 2 | Просмотр списка самолетов |  | Список самолетов | Вывод списка самолетов |
| 3 | Изменение данных самолета | Id Самолета |  | Изменение данных самолета |
| 4 | Установить стоимость билетов/количество возможных билетов | Id рейса |  | Добавление стоимости и количества мест в самолета |
| 5 | Изменить данные рейса | Id рейса |  | Внести изменения в рейс |
| 6 | Добавить новый рейс | Данные о рейсе | Обновленный список рейсов | Добавление рейса в базу |
| 7 | Добавить самолет | Данные нового самолета |  | Добавление в базу самолетов новый самолет |
| **Пилот** | | | | |
| 1 | Просмотр своих рейсов |  | Список рейсов | Вывод списков рейсов указанного пилота |
| 2 | Просмотр данных своего самолета |  | Данные самолета | Вывод данных самолета указанного пилота |
| 3 | Просмотр всего списка самолетов |  | Список самолетов | Вывод списка самолетов |
| **Администратор** | | | | |
| 1 | Просмотр всех рейсов и самолетов |  | Список рейсов/самолетов | Вывод списков рейсов и самолетов |
| 2 | Просмотр всех работников аэропорта |  | Список работников и их должность | Вывод списка работников с их должностью |
| 3 | Добавить нового работника | Персональные данные работника и должность | Обновленные список работников | Добавить работника в базу |

Хранимые данные

|  |  |  |  |
| --- | --- | --- | --- |
| № пп | Хранимые данные | Пользователи, которым разрешен доступ | Ограничения по типу и значению |
| 1 | Сотрудники | Администратор |  |
| 2 | Кассир | Администратор |  |
| 2 | Пилоты | Администратор |  |
| 3 | Администраторы | Администратор |  |
| 4 | Рейсы | Все |  |
| 5 | Самолеты | Все |  |
| 6 | Билеты | Кассир, администратор |  |

# **Схема БД**



## **Описание таблиц**

В таблице Role описаны должности работников аэропорта.

В таблице Users собрана вся информация о работниках, т.е. их id, должность, имя, фамилия, паспортные данные, логин и пароль.

В таблице Flights описана информация о рейсах: id рейса, id самолета, id пилота, аэропорты вылета и посадки, даты вылета и дата посадки.

В таблице Airport – информация о аэропорте, где он находится: id аэропорта, название, город и страна аэропорта.

В таблице Plane: id самолета, название самолета и компания, владеющая самолетом.

В таблице Ticket: id билета, id рейса, имя и фамилия пассажира, его паспортные данные, стоимость билета, место в самолете и доступен ли билет.

## **Код создания таблиц**

Create Table Role

(role\_id int IDENTITY(1,1) PRIMARY KEY Check(role\_id Between 1 And 4) NOT NULL,

role\_name varchar(15) Not Null

)

Go

Create Table Users

(user\_id int IDENTITY(1,1) Primary KEY Not Null,

role\_id int FOREIGN KEY REFERENCES Role Not Null,

name varchar(15) Not Null,

surname varchar(15) Not Null,

passport\_number varchar(20) Not Null,

login varchar(20) Not Null,

password varchar(128) Not Null

)

Go

Create Table Plane

(plane\_id int Identity(1,1) Primary Key Not Null,

plane\_name varchar(10) Not Null,

company varchar(15) Not Null

)

Go

Create Table Airport

(airport\_id int Identity(1,1) Primary Key Not Null,

airport\_name varchar(20) Not Null,

city varchar(15) Not Null,

country varchar(15) Not Null

)

Go

Create Table Flight

(flight\_id int Identity(1,1) Primary Key Not Null,

plane\_id int Foreign Key References Plane Not Null,

pilot\_id int Foreign Key References Users Not Null,

airport\_dispatch\_id int Foreign Key References Airport Not Null,

airport\_arrival\_id int Foreign Key References Airport Not Null,

departure\_date date Not Null,

arrival\_date date Not Null

)

Go

Create Table Ticket

(ticket\_id int Identity(1,1) Primary Key Not Null,

flight\_id int Foreign Key References Flight Not Null,

passager\_name varchar(15) Not Null,

passager\_surname varchar(15) Not Null,

passport\_number varchar(20) Not Null,

price int Not Null,

seat int Not Null,

enable int Check(enable Between 0 And 1) Not Null

)

# **Запросы**

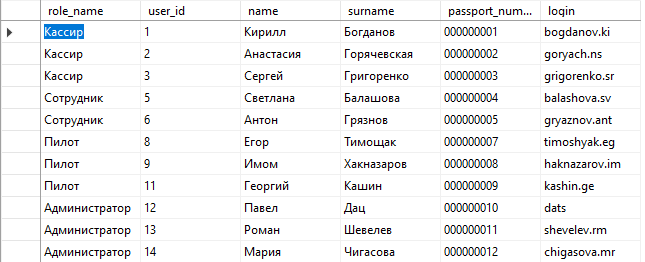
## **Представления**

* Создать представление объединения таблиц Users и Role по полю role\_id

CREATE VIEW test1 As

Select role\_name, user\_id,name,surname,passport\_number,login From Role r,Users u

Where r.role\_id=u.role\_id



* Создать представление, которое бы показывало помесячную выручку за год, с сортировкой по месяцам

Create View test2 As

Select Sum(t.price) as price, MONTH(f.departure\_date) as month From Ticket t Join Flight f On

t.flight\_id=f.flight\_id

Group By MONTH(f.departure\_date)



* Создать представление, которое выводит имя пилота и его кол-во рейсов за месяц

Create View test3 As

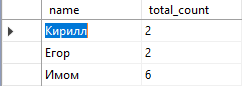
Select u.name, a.total\_count From Users u Join

(Select u1.name, COUNT(\*) as total\_count From Users u1 Join Flight f On

u1.user\_id=f.pilot\_id

Group By name) a On

u.name=a.name



## **Вывод**

* Вывести стоимость(price) из таблицы Ticket по определенному рейсу(flight\_id)

SELECT DISTINCT price FROM Ticket

WHERE flight\_id=4



* Вывести все рейсы(Flight)

Select \* From Flight

* Вывести все самолеты(Plane)

Select \* From Plane

* Вывод стоимости билета

Select Distinct price From Ticket

* Вывод списка рейсов(Flight), где определенный pilot\_id

Select airport\_dispatch\_id, airport\_arrival\_id, departure\_date From Flight, Users

Where role\_id=3 And user\_id=pilot\_id And pilot\_id=8

* Вывод данных самолета(Plane), где работает пилот pilot\_id

Select plane\_name, company From Plane Inner Join Flight On

Plane.plane\_id=Flight.plane\_id

Where Flight.pilot\_id=8



* Вывод всех самолетов(Plane)

Select \* From Plane

* Вывести всех работников аэропорта (users)

Select name, surname, role\_id From Users

## **Добавление**

* Добавить новые рейсы, т.е. ввести все данные нового рейса и добавить в таблицу Flight

Insert Into Flight(flight\_id,plane\_id,pilot\_id,airport\_dispatch\_id,airport\_arrival\_id,departure\_date,arrival\_date)

Values('4','3','9','3','13','2020-10-15','2020-10-16')

* Добавить нового работника(user’а), заполнить все его данные и добавить в таблицу Users

Insert Into Users(role\_id,name,surname,passport\_number,login,password)

Values('4', 'Дима1','Руснак', '000000013', 'dima\_rs', 'test')



* Добавить самолет(Plane), т.е. ввести все данные и добавить с таблицу Plane

Insert Into Plane(plane\_name,company)

Values('test3', 'test3')

* Добавить билет (указать данные пассажира)

Insert Into Ticket(flight\_id,passager\_name,passager\_surname,passport\_number,price,seat, enable)

Values('4','test4', 'test4', '000000154', '125000', '4', '1')

* Добавить новый аэропорт, указав необходимые параметры

Insert Into Airport(airport\_name,city,country)

Values('test5', 'test5', 'test5')

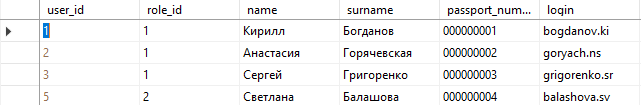
## **Изменение**

* Повысить сотрудника/кассира: изменить их role\_id в таблице Users

Update Users

Set role\_id=4

Where passport\_number='000000004'



* Изменить какой-либо параметр рейса(Flight)

Update Flight

Set pilot\_id=11

Where flight\_id=1

* Изменить какой-либо параметр самолета(Plane)

UPDATE Plane

SET plane\_name='Юг-тест'

WHERE plane\_id=5;

* Изменить место пассажира в самолете по его паспортным данным

Update Ticket

Set seat=10

Where passport\_number='164432183'

* Изменить данные пассажира в билете

Update Ticket

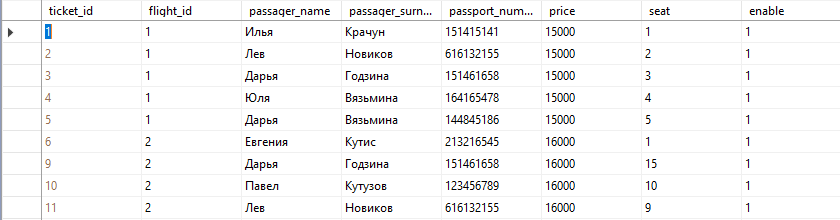
Set passager\_name='test6', passager\_surname='test6'

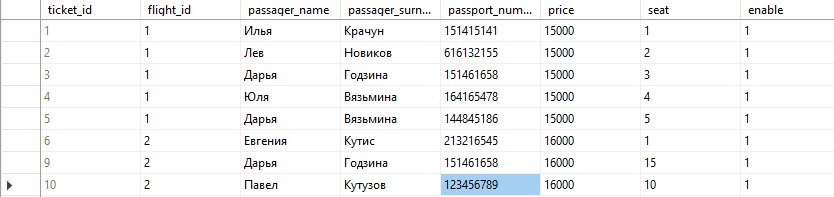
Where passport\_number='164432183'

## **Удаление**

* Удалить пассажира по номеру паспорта

Delete From Ticket Where passport\_number=' 616132155'





* Удалить рейс из базы

DELETE FROM Flight WHERE flight\_id=6;

* Удалить учетную запись пилота/кассира

Delete From Users Where role\_id=2 And passport\_number='000000006'

* Удалить самолет из базы

Delete From Plane Where plane\_id=7

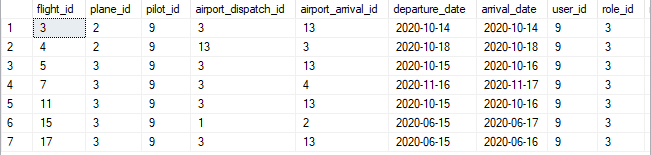
## **Сложные запросы**

* Вывести рейсы, которые проводил пилот, указав его паспорт

Select \* From Flight Join Users On

Flight.pilot\_id=Users.user\_id

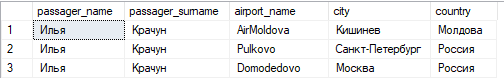
Where passport\_number='000000008'



* Вывести список пассажиров рейсов и аэропорт прибытия по определенной дате

Select passager\_name, passager\_surname, airport\_name, city,country From Ticket,Airport,Flight

Where Flight.departure\_date='2020-10-18'



* Вывести выручку за месяц

Select Sum(price) From Ticket Join Flight On

Ticket.flight\_id=Flight.flight\_id

Where Flight.departure\_date>'2020-10-1' And Flight.departure\_date<'2020-10-31'

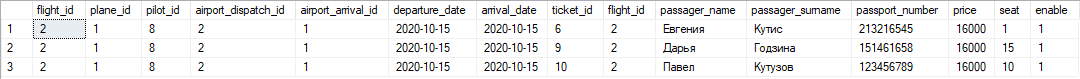


* Вывести рейс, у которого самая дорогая цена билетов

Select \* From Flight Join Ticket On

Flight.flight\_id=Ticket.flight\_id

Where price = (Select Max(price) From Ticket)



* Вывести страну, куда чаще всего совершаются рейсы

Select Distinct country From Airport Join Flight On

Airport.airport\_id=Flight.airport\_arrival\_id

Where Flight.airport\_arrival\_id=

(SELECT Top 1 airport\_arrival\_id as cnt FROM Flight

GROUP BY airport\_arrival\_id

ORDER BY cnt

Desc)



* Вывести все рейсы, указав аэропорт вылета

Select \* From Flight Join Airport On

Flight.airport\_dispatch\_id=Airport.airport\_id

Where Airport.airport\_name='AirMoldova'



* Изменить пароль юзеру, указав фамилию, имя и паспорт

Update Users

Set password='test\_kurs'

Where name='Павел' And surname='Дац' And passport\_number='000000010'

* Изменить название самолета, указав паспортные данные пилота, который на нем работает

Update Plane

Set plane\_name='test10'

From Plane p Join Flight f On

p.plane\_id=f.plane\_id

Join Users u On

u.role\_id=3 And f.pilot\_id=u.user\_id

Where u.passport\_number='000000007'

## **ХП**

* Вывод пассажиров определенного рейса, указав id рейса

Go

Create Procedure Passager(@flight\_id As Int) As

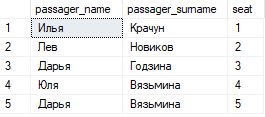
Begin

Select t.passager\_name, t.passager\_surname, t.seat From Ticket t, Flight f

Where t.flight\_id=f.flight\_id And @flight\_id=f.flight\_id

End;

Exec Passager '1'



* Вывод всех работников, указав их должности

Go

Create Procedure Workers As

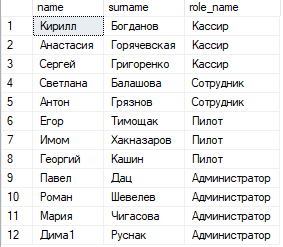
Begin

Select u.name, u.surname, r.role\_name From Users u, Role r

Where u.role\_id=r.role\_id

End;

Exec Workers



* Вывести id рейса, названия самолетов, даты отправки/посадки, имя пилотов всех возможных рейсов

Go

Create Procedure InfoFlights As

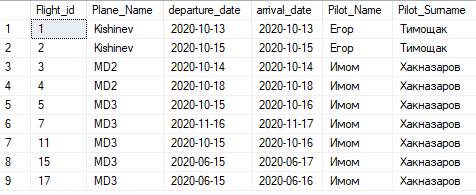
Begin

Select f.flight\_id As Flight\_id, p.plane\_name As Plane\_Name, f.departure\_date, f.arrival\_date, u.name As Pilot\_Name, u.surname As Pilot\_Surname From Flight f, Plane p, Users u

Where f.plane\_id=p.plane\_id And f.pilot\_id=u.user\_id And u.role\_id=3

End;

Exec InfoFlights



## **Триггеры**

* Проверка даты вылета и посадки (чтобы дата посадки не была раньше даты вылета)

Create Trigger checkData On Flight

After Insert, Update

As

Begin

Set Nocount On;

declare @data1 date

set @data1 = (Select departure\_date From inserted)

declare @date2 date

set @date2 = (Select arrival\_date From inserted)

if @data1>@date2

begin

rollback tran

raiserror('Data error',16,10)

end

end

Insert Into Flight(plane\_id,pilot\_id,airport\_dispatch\_id,airport\_arrival\_id,departure\_date,arrival\_date)

Values('3','9','1','2','15-06-2020','14-06-2020')



* Цена не может быть отрицательна или равной 0

Create Trigger CostCheck On Ticket

After Insert, Update

As

Begin

Set Nocount On;

declare @cost int

set @cost = (Select price From inserted)

if @cost<=0

begin

rollback tran

raiserror('Cost error',16,10)

end

end

Insert Into Ticket(flight\_id,passager\_name,passager\_surname,passport\_number,price,seat,enable)

Values(1, 'triggerCost', 'triggerCost', '1411654321', -105, 14, 1)



* На повторение паспорта

CREATE TRIGGER IsPassport

ON Users

FOR INSERT, UPDATE

AS

BEGIN

IF EXISTS(SELECT passport\_number FROM Users WHERE passport\_number IN (SELECT passport\_number FROM Inserted) GROUP BY passport\_number HAVING count(\*) > 1)

BEGIN

RAISERROR('Есть такой паспорт!', 16, 1)

ROLLBACK TRAN

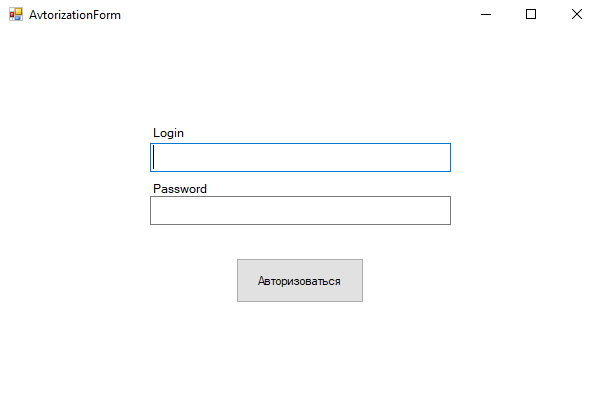
END

END

# **Клиентская часть**

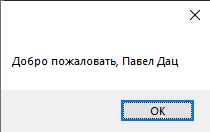
Для разработки клиентского приложения был использован язык программирования C#. Для всех видов пользователей реализованы отдельные интерфейсы.

## **Окно входа**

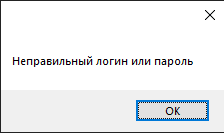


Вход в систему осуществляется через логин и пароль. Логин уникален для каждого пользователя. Пароли зашифрованы SHA512 шифрованием.

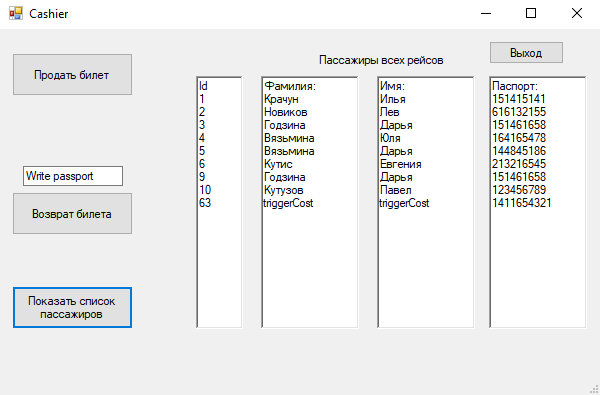
В случае успешного входа, появляется окно приветствия пользователя, с указание его имени и фамилии, которые хранятся в БД.



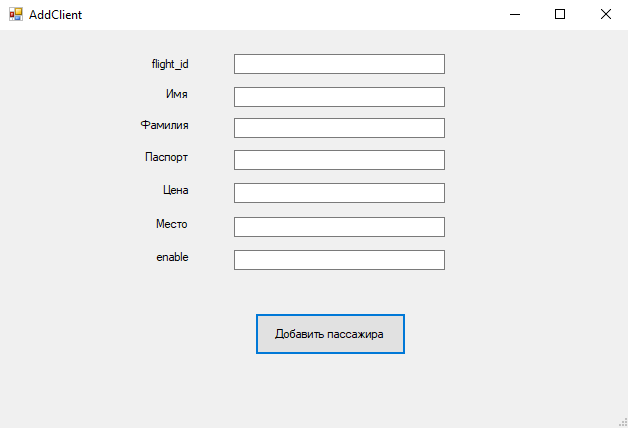
Если в логине или пароле допущена ошибка, то приложение уведомляет об этом.



## **Окно кассира**

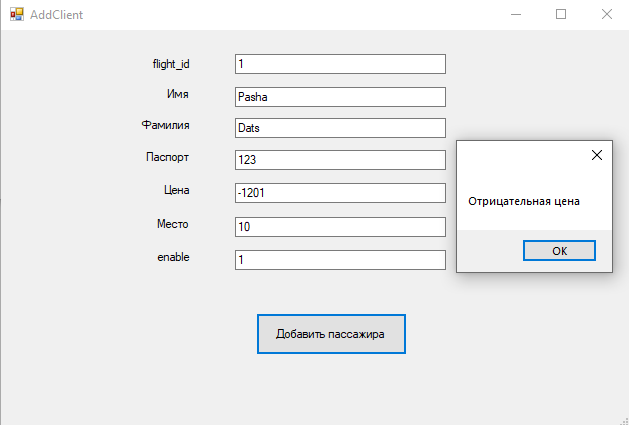


Кассир может также продать билет, для этого при нажатии на кнопку открывается новое окно с местами для заполнения.



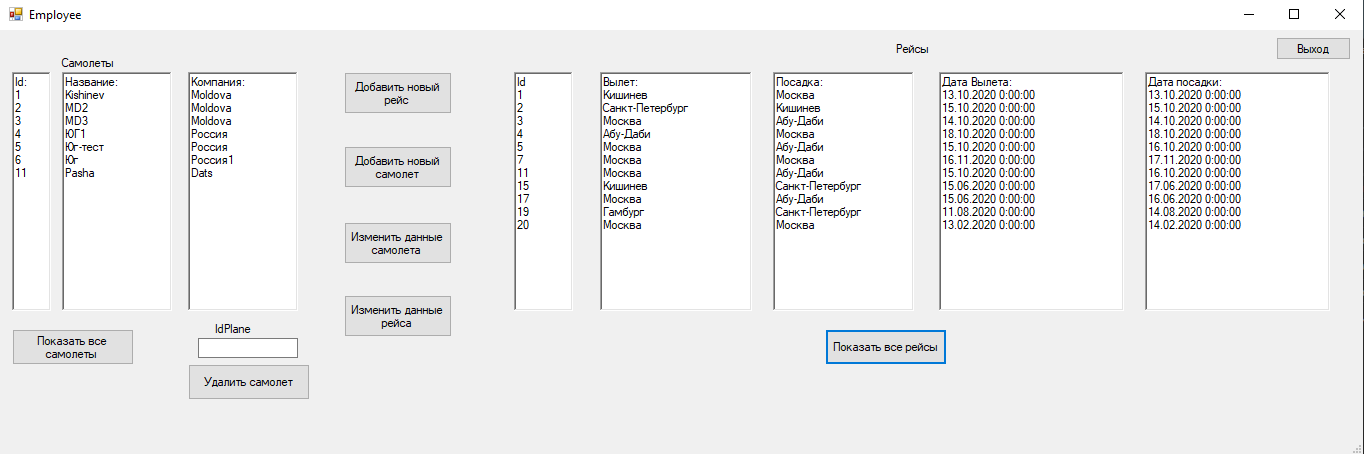
Тут идет проверка на то, чтобы все поля были заполнены, цена не была отрицательной и не равнялась 0. Если паспорт уже существует в базе, то идет проверка имени и фамилии пассажира, если они совпадают, то пассажир вносится в БД, иначе информация об ошибке. Также тут проверяется занятость места.

Приведем пример отрицательно цены



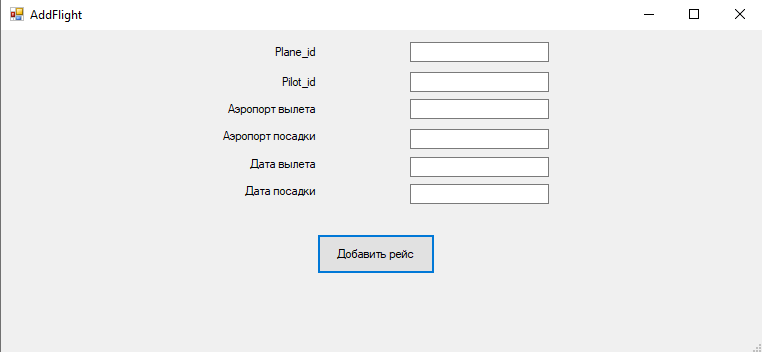
Все остальные проверки работают аналогично.

## **Окно сотрудника**



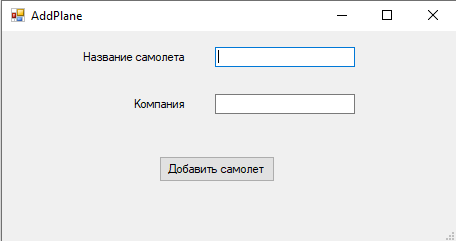
Сотрудник может просматривать все доступные самолеты и рейсы. Добавлять/изменять данные рейса, добавлять/изменять данные самолета, удалять самолеты.

При нажатии кнопки «Добавить новый рейс» открывается новое окно для заполнения формы.

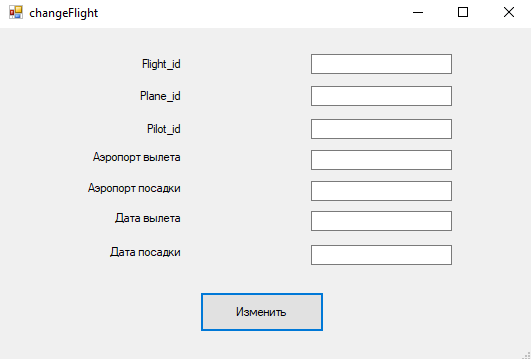


Единственная проверка в данной форме – дата вылета не может быть позже, чем дата посадки.

При нажатии кнопки «Добавить новый самолет»:

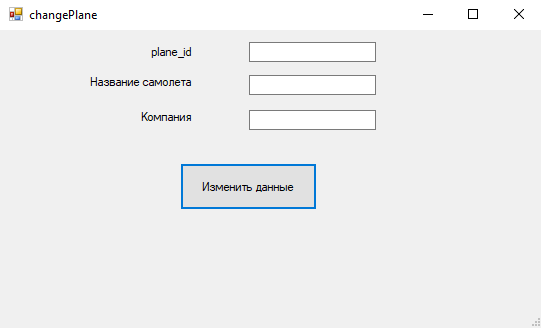


При нажатии «Изменить данные рейса»:



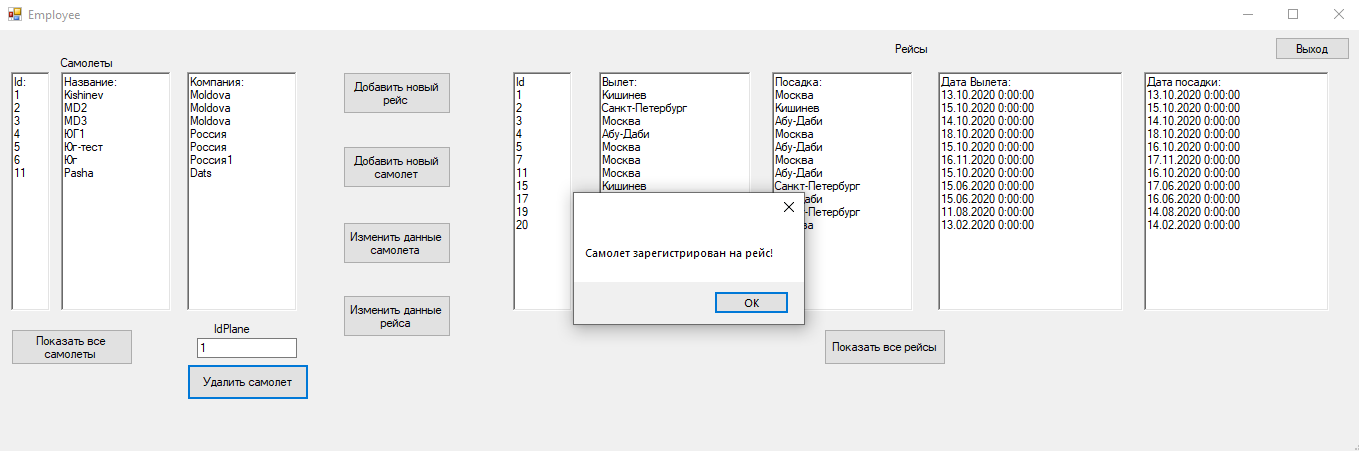
Тут не все поля могут быть заполнены, а только некоторые.

При нажатии «Изменить данные самолета»:

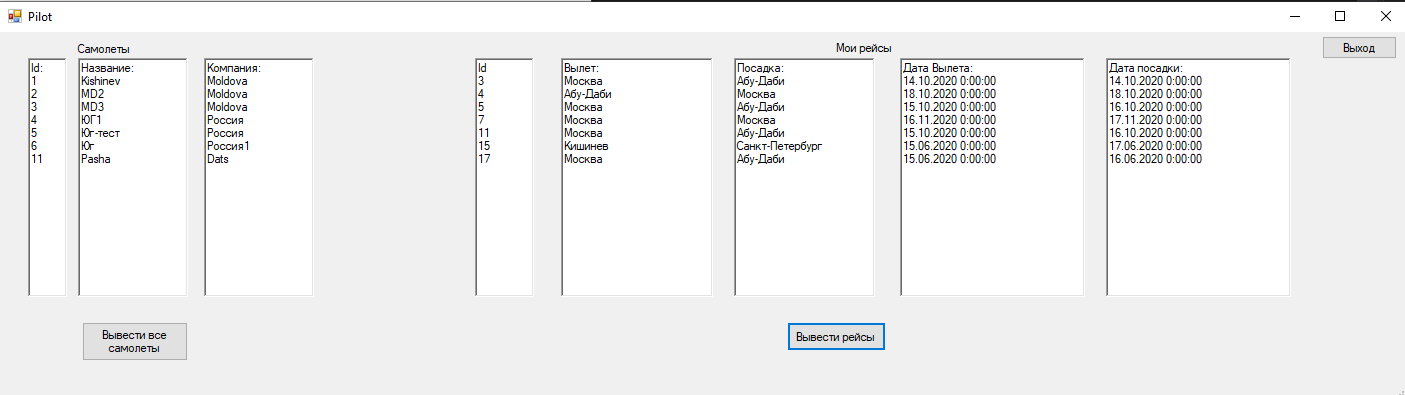


Также не все поля должны быть заполнены.

При попытке удалить самолет идет проверка, если самолет зарегистрирован на какой-либо рейс, если да – то окно с ошибкой.



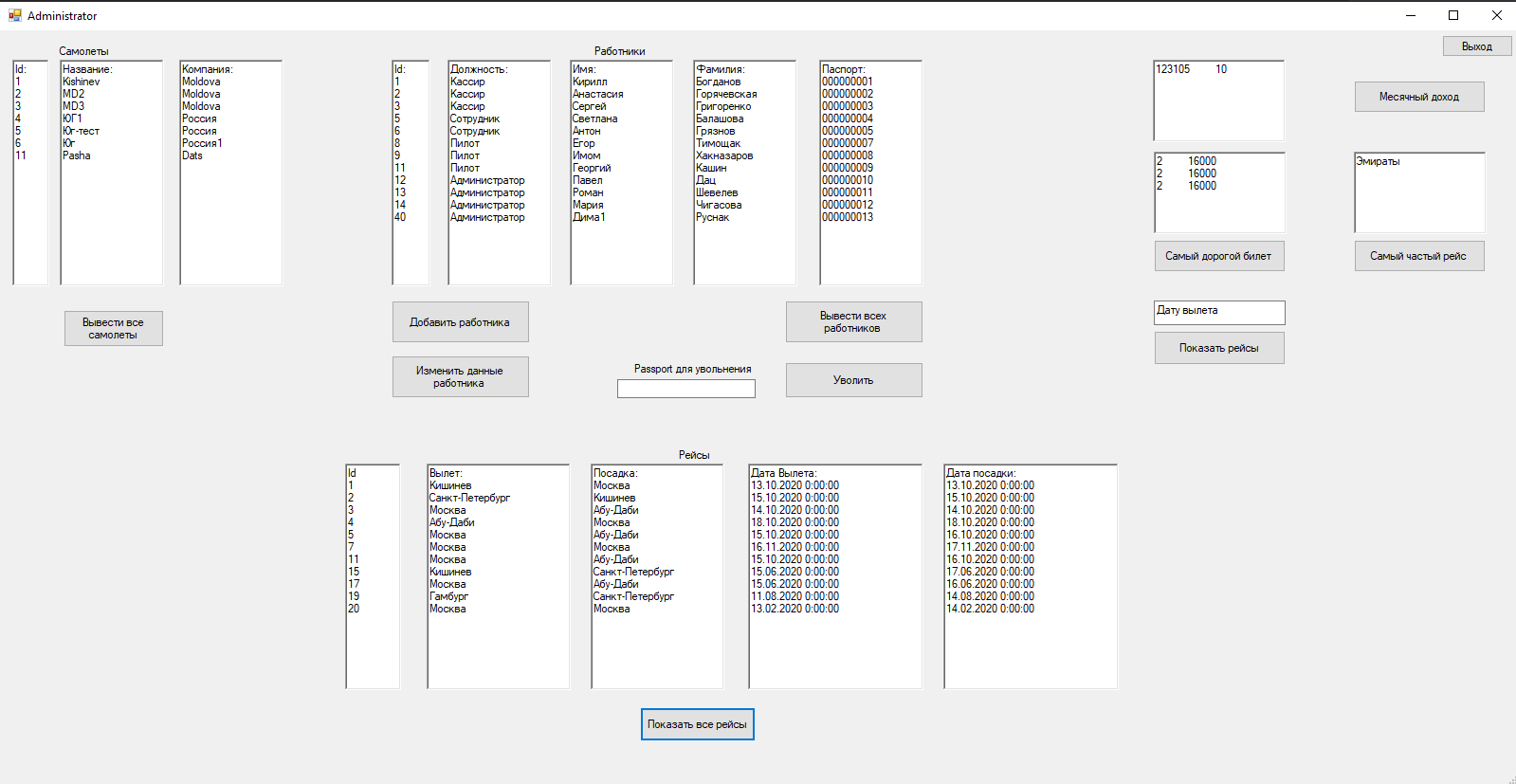
## **Окно пилота**



Пилот может видеть список всех самолетов и свои рейсы.

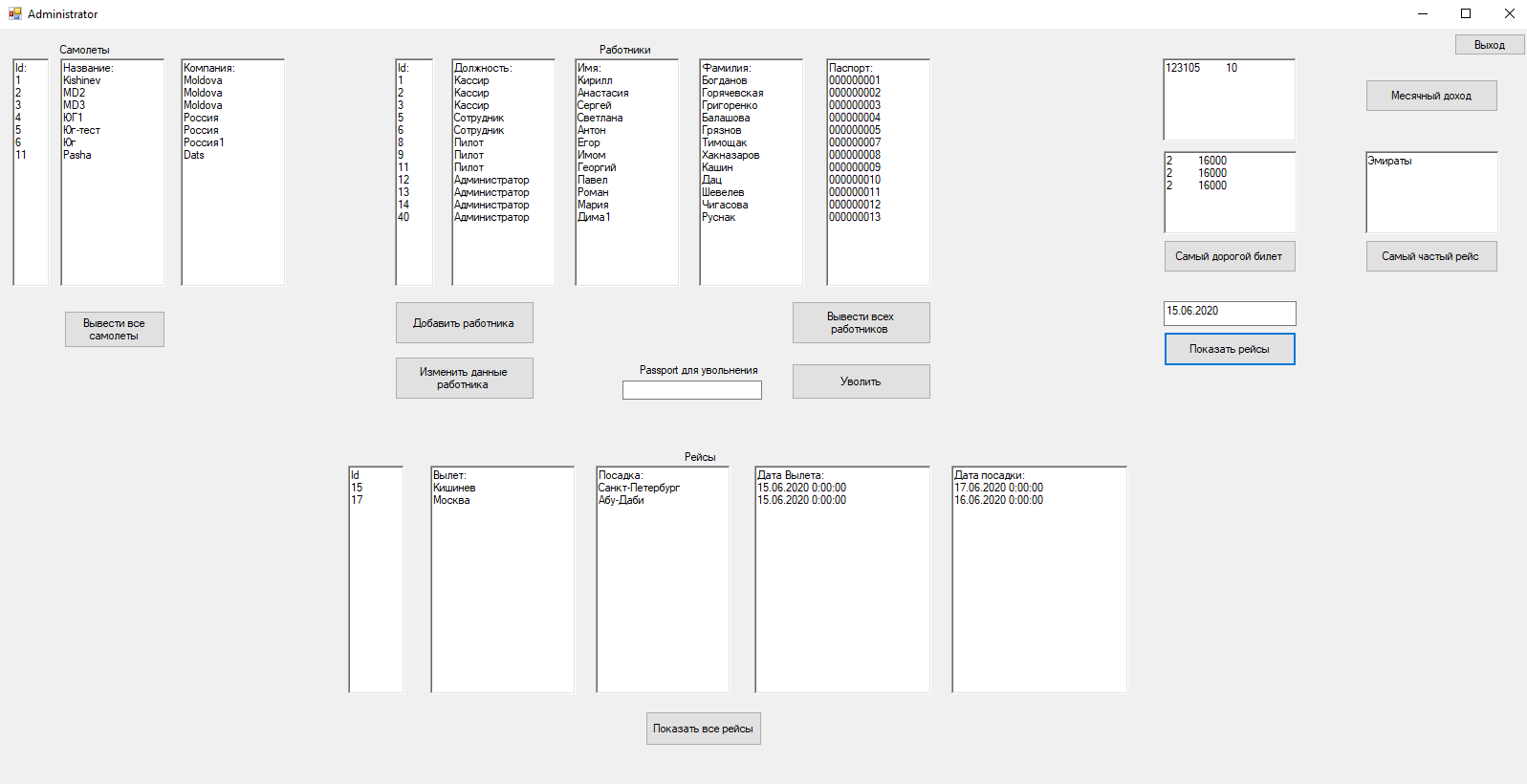
При авторизации идет перехват user\_id пилота, благодаря чему я могу вывести только его рейсы.

## **Окно администратора**

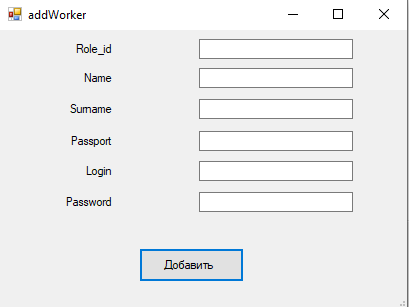


Администратор может просматривать все самолеты, список всех работников, добавлять/изменять данные работников или увольнять их, просмотр всех рейсов. Вывод помесячного дохода с указанием суммы прибыли и номер месяца, самого дорогого билета, самого частого рейса.

При указании даты вылета, можно выводить рейсы, которые имеют дату вылета ту, что мы указали.

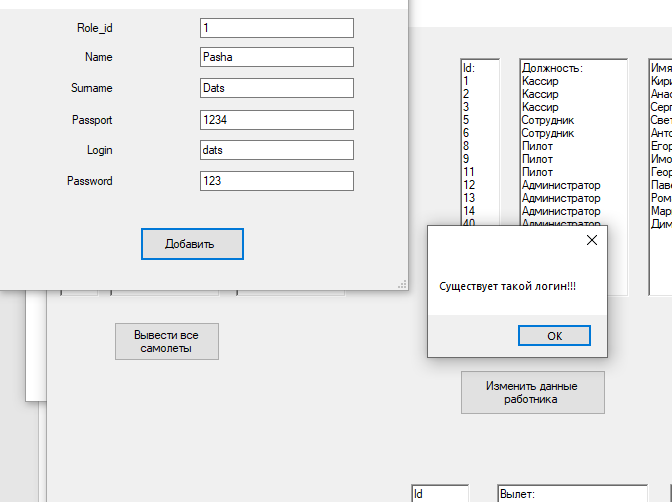


При нажатии кнопки «Добавить работника»:



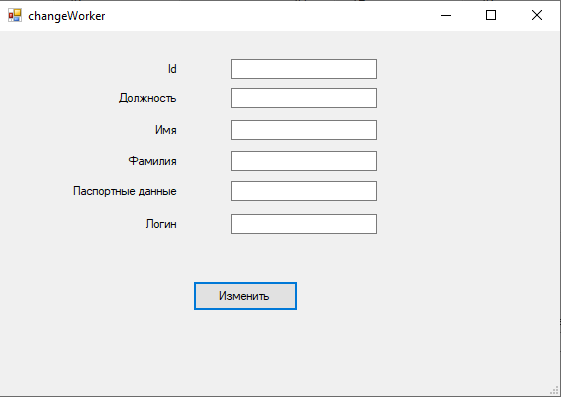
Тут есть ряд проверок: все поля должны быть заполнены, паспорт не может повторяться, логин не может повторяться.

Попробуем добавить нового работника с логином, который уже есть:



Все остальные ошибки выглядят аналогично.

При нажатии «Изменить данные работника»:



Тут могут быть заполнены не все поля. Но также проверяем паспорт и логин на уникальность.

При увольнении работника, он удаляется из БД и из таблицы в приложении.

# **Заключение**

В качестве СУБД я использовал Microsoft SQL Server. Данная СУБД используется для работы с БД размером от персональных до крупных БД масштаба предприятия.

Я выбрал данную СУБД потому, что с ней я уже работал ранее. А также есть очень много документаций про ее синтаксис.

В качестве среды разработки использовал C#. Выбрал эту среду потому, что также в ней работал.

Трудность возникла при шифровании паролей. С таким никогда не сталкивался и не знал, как работать, но, прочитав документации, понял что к чему. По итогу справился с этим заданием.

# **Список литературы**

1. Биллиг, В. А. Основы программирования на С# / В.А. Биллиг. - М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, **2015**. - 488 c.
2. Джеймс, Р. Грофф SQL. Полное руководство / Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. - М.: Вильямс, 2014. - 960 c.
3. Диго, С.М. Базы данных / С.М. Диго. - М.: Финансы и статистика, **2013**. - 592 c.
4. Кузнецов, С.Д. Основы баз данных / С.Д. Кузнецов. - М.: Бином, **2016**. - 484 c.
5. Уотсон, Карли Visual C# 2010. Полный курс / Карли Уотсон и др. - М.: Вильямс, **2014**. - 960 c.