Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет им. Г.В. Плеханова»

**Московский приборостроительный техникум**

**Курсовой проект**

ПМ 11 «Разработка, администрирование и защита баз данных»

МДК 11.01 «Технология разработки и защиты баз данных»

Специальность 09.02.07 «Информационные системы и программирование»

Тема: «Разработка информационной системы «Багажное отделение аэропорта»

**Пояснительная записка**

Листов: 245



Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ / А.А. Шимбирёв

«19» Декабря 2022 г.

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_ / Д.А. Вишняков

«19» Декабря 2022 г.

2022

Оглавление

[1. ОБЩАЯ ЧАСТЬ. 4](#_Toc122319397)

[1.1. Цель разработки 4](#_Toc122319398)

[1.2. Средства разработки 4](#_Toc122319399)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 5](#_Toc122319400)

[2.1. Постановка задачи 5](#_Toc122319401)

[2.2. Внешняя спецификация 8](#_Toc122319402)

[2.2.1. Описание задачи 8](#_Toc122319403)

[2.3. Проектирование 13](#_Toc122319404)

[2.3.1. Схема архитектуры программы 13](#_Toc122319405)

[3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 24](#_Toc122319406)

[ЗАКЛЮЧЕНИЕ 29](#_Toc122319407)

[СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ 30](#_Toc122319408)

[**ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ** 32](#_Toc122319409)

[**ПРИЛОЖЕНИЕ Б. СЦЕНАРИЙ ТЕСТОВ И РЕЗУЛЬТАТЫ ТЕСТОВЫХ ИСПЫТАНИЙ** 79](#_Toc122319410)

[**ПРИЛОЖЕНИЕ В. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ** 90](#_Toc122319411)

[**ПРИЛОЖЕНИЕ Г. СКРИПТ БАЗЫ ДАННЫХ** 97](#_Toc122319412)

# ВВЕДЕНИЕ

Данный курсовой проект выполнен в соответствии с заданием на курсовое проектирование. Темой проекта является разработка информационной системы «Багажное отделение аэропорта». Реализация подсистем: «сотрудники», «склад», «услуги», «клиенты».

Из данной цели вытекают следующие задачи:

1. Осуществлять сбор, обработку и анализ информации для проектирования баз данных;

2. Проектировать базу данных на основе анализа предметной области;

3. Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области;

4. Реализовывать базу данных в конкретной системе управления базами данных.

5. Администрировать базу данных.

6. Защищать информацию в базе данных с использованием технологии защиты информации.

Данный сервис поможет может аэропотам вести учет багажа, полеты и анализировать информацию о полетах.

1. ОБЩАЯ ЧАСТЬ.
   1. Цель разработки

Предоставить руководителям аэропорта удобный инструмент для администрирования и анализа деятельности аэропорта.

* 1. Средства разработки

В качестве программных средств разработки использовалось ПО, представленное в таблице 1.

Таблица 1 – Программные средства разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип средства | Название средства | Назначение |
| 1 | 2 | 3 | 4 |
| 1 | Операционная система | Windows 10 | Организация взаимодействия программ и пользователя |
| 2 | Инструментальное средство разработки программных решений | Visual Studio 2022 | Разработка веб приложения |
| 3 | Графическая оболочка | PgAdmin 4 | Администрирование базы данных. Разработка схем для проектирования приложения. |

В качестве средств вычислительной техники использовался персональный компьютер и ноутбук Lenovo V12. Их характеристики представлены в таблице 2.

Таблица 2 – Технические средства

|  |  |  |
| --- | --- | --- |
| № | Тип оборудования | Наименование оборудования |
| 1 | 2 | 3 |
| Персональный компьютер | | |
| 1 | Центральный процессор | AMD Ryzen 5 3500-X |
| 2 | Количество ядер | 6 |
| 3 | Видеоадаптер | NVIDIA GeForce GTX 1650 |
| 4 | Оперативная память | 12GB DDR4 |
| 5 | Твердотельный накопитель | 512GB SSD |
| 6 | Разрешение экрана | 1920x1080 |
| Lenovo V12 | | |
| 7 | Центральный процессор | AMD Ryzen 5000 |
| 8 | Количество ядер | 4 |
| 9 | Видеоадаптер | Интегрированная |
| 10 | Оперативная память | 16GB |
| 11 | Внутренняя память | 1TB SSD |
| 12 | Разрешение экрана | 1920x1080 |

1. СПЕЦИАЛЬНАЯ ЧАСТЬ
   1. Постановка задачи

Необходимо разработать веб приложение для контроля информационной системы аэропорта.

* + 1. Входные и выходные данные

Входные данные представлены ниже.

* Сотрудник: представляет сотрудника аэропорта, который работает с багажным отделением. Содержит такие атрибуты, как employee\_id, имя, адрес электронной почты, номер телефона и job\_title.
* Склад: представляет собой склад, где хранится багаж. Содержит такие атрибуты, как warehouse\_id, местоположение и вместимость.
* Услуга: представляет собой услугу, предоставляемую аэропортом по обработке багажа. Содержит такие атрибуты, как service\_id, имя и описание.
* Клиент: представляет клиента, который зарегистрировал багаж в аэропорту. Содержит такие атрибуты, как customer\_id, имя, адрес электронной почты и номер телефона.
* Багаж: представляет собой место багажа, которое было зарегистрировано клиентом. Содержит такие атрибуты, как baggage\_id, вес, размеры, content\_description и customer\_id.
* Багажное отделение: представляет собой отсек в самолете, где хранится багаж. Содержит такие атрибуты, как compartment\_id, airplane\_type, capacity и warehouse\_id.
* Полет: представляет собой полет самолета в город. Содержит такие атрибуты, как compartment\_id, FlightNumber, DepartureTime, ArrivalTime, Destination, DepartureGate, ArrivalGate.
* Должность: представляет собой должность сотрудника аэропорта. Содержить такие атрибуты, как Id, Name, Salary.

Выходные данные представлены в таблице ниже.

* Резервная копия базы данных с данными.
* Файлы CSV выгрузки данных
* Граффик количества полетов в города
* Граффик показывающий число полетов за каждый месяц года.
  + 1. Подробные требования к проекту

Необходимо разработать бэкенд приложение на web framework-е ASP .NET на языке программирования C# и клиент приложение на framework-е Aalonia на языке программирования C#.

Для администратора:

При нажатии на кнопку «База данных» пользователь переходит на окно, где распологается список всех таблиц базы данных. Нажатием по записи списка пользователь переходит на окно редактирования данными. На этой странице располагается таблица, кнопка для добавления, редактирования и удаления данных, поисковая строка и кнопки навигации. При нажатии на кнопку доабвления данных появлсяется форма с кнопками «добавить» и «отмена».

При вводе значения в поисковую строку и нажатия клавишы «Ввод» или кнопки «поиск» происходит поиск с подчеркиванием найденых совпадений.

При нажатии на кнопку назад, пользователь возвращается к списку таблиц базы данных.

При нажатии на кнопку «Диаграммы» пользователь увидит две диаграммы.

При нажатии на кнопку «резервное копирование» пользователь переходит на окно, где предусмотрены функции резервного копирования базы данных. Пользователь видит кнопки «Выгрузить» и «Восстановить». При нажатии на кнопку «Выгрузить» происходит выгрузка базы данных вместе с данными в sql файл. Теперь этот файл появился в списке, если нажать на кнопку «восстановить» база данных вернется к точке, на которой была выгружена.

При нажатии на кнопку «Выгрузка данных» пользователь переходит на окно, где предусмотрены функции выгрузки данных в CSV файл. Пользователь видит кнопки «Экспортировать данные в CSV файлы». При нажатии на кнопку «Экспортировать данные в CSV файлы» будет создан файл csv на каждую таблицу.

* 1. Внешняя спецификация
     1. Описание задачи

Приложение должно включать в себя в набор следующих функций:

* Создавать объекты в базе данных;
* Редактировать объекта в базе данных;
* Удалять объекты из базы данных;
* Выводить информацию об объектах из базы данных;
* Создавать резервную копию базы данных;
* Восстановить базу из резервной копии;
* Выгрузка данных в csv файлы;
* Вывод граффика показываюшего количество полетов в города;
* Вывод граффика показывающего число полетов в месяц на протяжении года.
  + 1. Входные и выходные данные

В таблице ниже представлены входные данные

Таблица 3 – Входные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Поле | Ограничение | Тип | Описание |
| 1 | 2 | 3 | 4 | 5 |
| public."Baggages" | | | | |
| 1 | "Id" | NOT NULL, | integer | Поле для ввода "Id" |
| 2 | "Weight" | NOT NULL, | real | Поле для ввода "Weight" |
| 3 | "Size" | NOT NULL, | text | Поле для ввода "Size" |
| 4 | "FlightId" | NOT NULL, | integer | Поле для ввода "FlightId" |
| 5 | "CustomerId" | NOT NULL, | integer | Поле для ввода "CustomerId" |
| 6 | "ServiceId" | NOT NULL, | integer | Поле для ввода "ServiceId" |
| 7 | "WarehouseId" | NOT NULL, | integer | Поле для ввода "WarehouseId" |
| 8 | "EmployeeId" | NOT NULL | integer | Поле для ввода "EmployeeId" |
| public."Customers" | | | | |
| 1 | "Id" | NOT NULL, | integer | Поле для ввода "Id" |
| 2 | "FirstName" | NOT NULL, | text | Поле для ввода "FirstName" |
| 3 | "LastName" | NOT NULL, | text | Поле для ввода "LastName" |
| 4 | "Phone" | NOT NULL, | text | Поле для ввода "Phone" |
| 5 | "Email" | NOT NULL | text | Поле для ввода "Email" |
| public."Employees" | | | | |
| 1 | "Id" | NOT NULL, | integer | Поле для ввода "Id" |
| 2 | "Name" | NOT NULL, | text | Поле для ввода "Name" |
| 3 | "Phone" | NOT NULL, | text | Поле для ввода "Phone" |
| 4 | "Email" | NOT NULL, | text | Поле для ввода "Email" |
| 5 | "WarehouseId" | NOT NULL, | integer | Поле для ввода "WarehouseId" |
| 6 | "PositionId" | NOT NULL | integer | Поле для ввода "PositionId" |
| public."Flights" | | | | |
| 1 | "Id" | NOT NULL, | integer | Поле для ввода "Id" |
| 2 | "FlightNumber" | NOT NULL, | text | Поле для ввода "FlightNumber" |
| 3 | "DepartureTime" | with time zone NOT NULL, | timestamp | Поле для ввода "DepartureTime" |
| 4 | "ArrivalTime" | with time zone NOT NULL, | timestamp | Поле для ввода "ArrivalTime" |
| 5 | "Destination" | NOT NULL, | text | Поле для ввода "Destination" |
| 6 | "DepartureGate" | NOT NULL, | text | Поле для ввода "DepartureGate" |
| 7 | "ArrivalGate" | NOT NULL | text | Поле для ввода "ArrivalGate" |
| public."Positions" | | | | |
| 1 | "Id" | NOT NULL, | integer | Поле для ввода "Id" |
| 2 | "Name" | NOT NULL, | text | Поле для ввода "Name" |
| 3 | "Salary" | NOT NULL | numeric | Поле для ввода "Salary" |
| public."Services" | | | | |
| 1 | "Id" | NOT NULL, | integer | Поле для ввода "Id" |
| 2 | "Name" | NOT NULL, | text | Поле для ввода "Name" |
| 3 | "Description" | NOT NULL | text | Поле для ввода "Description" |
| public."Users" | | | | |
| 1 | "Id" | NOT NULL, | integer | Поле для ввода "Id" |
| 2 | "Login" | NOT NULL, | text | Поле для ввода "Login" |
| 3 | "Password" | NOT NULL, | text | Поле для ввода "Password" |
| 4 | "Email" | NOT NULL, | text | Поле для ввода "Email" |
| 5 | "Role" | NOT NULL | integer | Поле для ввода "Role" |
| public."Warehouses" | | | | |
| 1 | "Id" | NOT NULL, | integer | Поле для ввода "Id" |
| 2 | "Location" | NOT NULL, | text | Поле для ввода "Location" |
| 3 | "Phone" | NOT NULL, | text | Поле для ввода "Phone" |
| 4 | "Email" | NOT NULL | text | Поле для ввода "Email" |
| 5 | "WarehouseId" | NOT NULL, | integer | Поле для ввода "WarehouseId" |
| 6 | "PositionId" | NOT NULL | integer | Поле для ввода "PositionId" |

В таблице ниже представлены выходные данные

Таблица 4 – Выходные данные

|  |  |  |
| --- | --- | --- |
| № | Имя поля | Описание |
| 1 | 2 | 3 |
| 1 | Резервная копия базы данных с данными | Файл sql |
| 2 | Граффик количества заказанных услуг | Картинка показываюшая количество полетов в города |
| 3 | Файл CSV выгрузки данных | Файл csv |
| 4 | Граффик показывающий количество бронирований за каждый месяц года | Картинка показывающий число полетовза каждый месяц года |

* + 1. Методы

Данная программа должна быть написана с использованием таких парадигм как: наследование, полиморфизм, инкапсуляция и абстракция.

* + 1. Тесты

Сценарий тестирования и результаты тестовых испытаний приведен в Приложении А.

* + 1. Контроль целостности данных

В Таблице ниже представлен контроль целостности данных, описывающий ситуации и реакцию приложения на выполнение задач, связанных с различными действиями программы.

Таблица 5 – Контроль целостности данных

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ситуация | Аномалия | Реакция программы |
| 1 | 2 | 3 | 4 |
| 1 | Попытка добавить данные в таблицу | Не заполнены поля | Вывод сообщения об ошибке заполнения полей |
| 2 | Попытка изменить данные в таблице | Не заполнены поля | Вывод сообщения об ошибке заполнения полей |

* 1. Проектирование
     1. Схема архитектуры программы

На рисунке ниже представлена схема архитектуры проекта

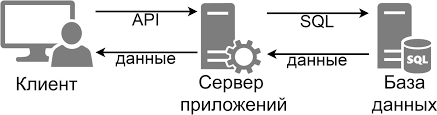


Рисунок 1 – Схема архитектуры программы

* + 1. Логическая схема данных

На рисунке ниже представлена логическая схема базы данных

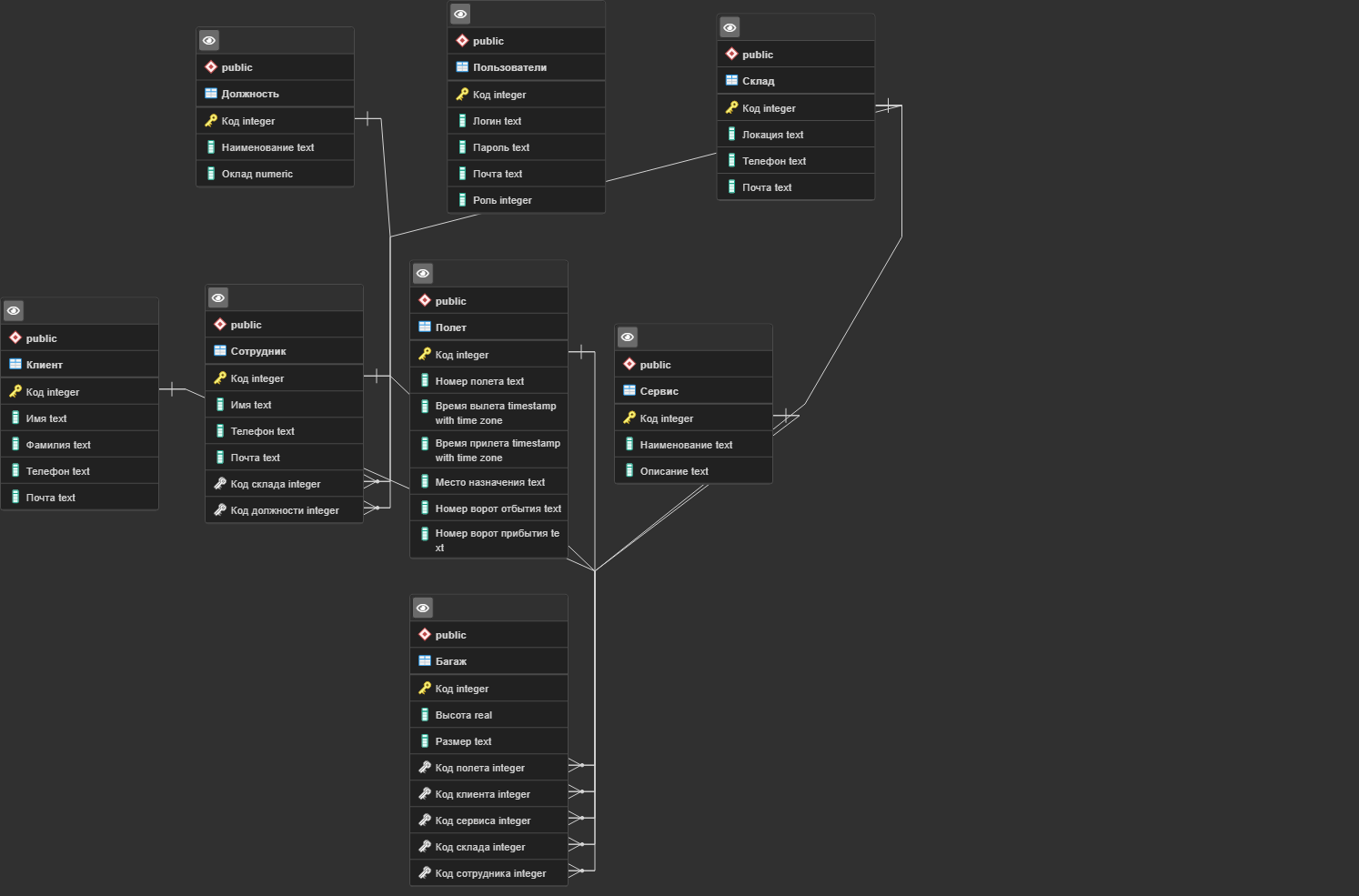


Рисунок 2 – логическая схема базы данных

* + 1. Физическая схема данных

На рисунке ниже представлена физическая схема данных

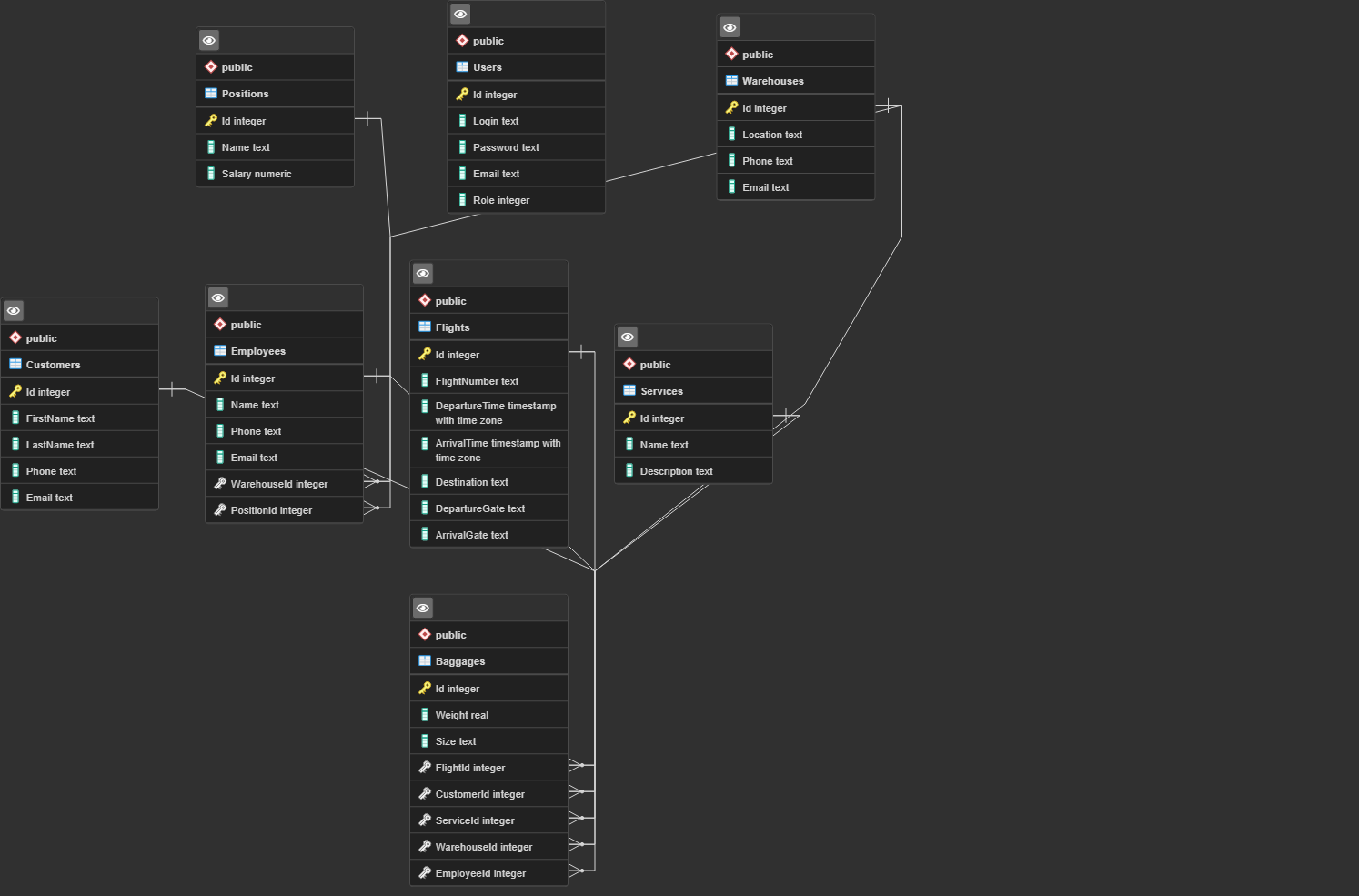


Рисунок 3 – физическая схема базы данных

* + 1. Структурная схема

На рисунке ниже представлена структурная схема программы.

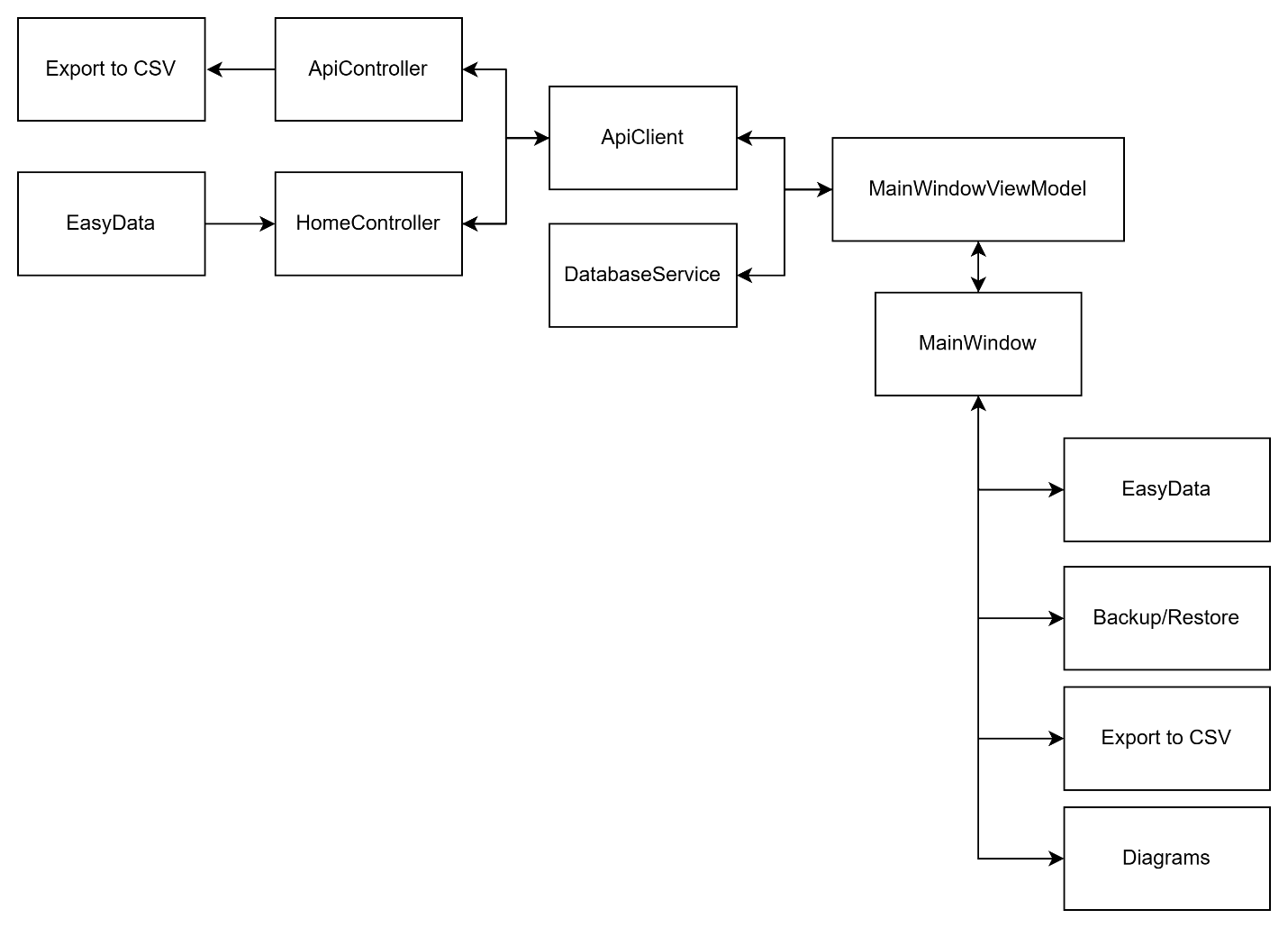


Рисунок 4 – структурная схема

В таблице ниже представлено название и описание всех модулей структурной схемы приложения.

Таблица – описание всех модулей структурной схемы

| № | Модуль | Описание |
| --- | --- | --- |
| 1 | 2 | 3 |
| Client | | |
| 1 | Baggage.cs | Данный модуль представляет собой модель багажа |
| 2 | CharsModel.cs | Данный модуль представляет собой модель графиков |
| 3 | Customer.cs | Данный модуль представляет собой модель клиента |
| 4 | Employee.cs | Данный модуль представляет собой модель сотрудника |
| 5 | Flight.cs | Данный модуль представляет собой модель полета |
| 6 | Position.cs | Данный модуль представляет собой модель должности |
| 7 | Role.cs | Данный модуль представляет собой модель роли |
| 8 | Service.cs | Данный модуль представляет собой модель службы |
| 9 | User.cs | Данный модуль представляет собой модель пользователя |
| 10 | Warehouse.cs | Данный модуль представляет собой модель склада |
| 11 | ApiClient.cs | Данный модуль представляет собой Api клиента |
| 12 | DatabaseService.cs | Данный модуль представляет собой сервис для взаимодействия с БД |
| 13 | MainWindowViewModel.cs | Данный модуль представляет собой viewmodel главного окна |
| 14 | ViewModelBase.cs | Данный модуль представляет собой базовую view модель |
| 15 | MainWindow.axaml | Данный модуль представляет собой главное окно |
| 16 | MainWindow.axaml.cs | Данный модуль представляет собой класс для главного окна |
| 17 | App.axaml | Данный модуль представляет собой view приложения |
| 18 | App.axaml.cs | Данный модуль представляет собой логику приложения |
| 19 | Program.cs | Данный модуль представляет собой базовый класс |
| 20 | ViewLocator.cs | Данный модуль представляет локатор для View |
| Backend | | |
| 1 | ApiController.cs | Программный компонент, отвечающий за управление api запросов |
| 2 | HomeController.cs | Программный компонент, отвечающий за управление базовыми задачами |
| 3 | Baggage.cs | Данный модуль представляет собой модель багажа |
| 4 | ChartModel.cs | Данный модуль представляет собой модель графиков |
| 5 | Customer.cs | Данный модуль представляет собой модель клиента |
| 6 | Employee.cs | Данный модуль представляет собой модель сотрудника |
| 7 | ErrorViewModel.cs | Данный модуль представляет собой модель полета |
| 8 | Flight.cs | Данный модуль представляет собой модель должности |
| 9 | Position.cs | Данный модуль представляет собой модель роли |
| 10 | Role.cs | Данный модуль представляет собой модель службы |
| 11 | Service.cs | Данный модуль представляет собой модель пользователя |
| 12 | User.cs | Данный модуль представляет собой модель склада |
| 13 | Warehouse.cs | Данный модуль представляет собой модель багажа |
| 14 | Charts.cshtml | Данный модуль представляет собой view для диаграмм |
| 15 | EasyData.cshtml | Данный модуль представляет собой view для easyData |
| 16 | Index.cshtml | Данный модуль представляет собой представление главной страницы |
| 17 | Privacy.cshtml | Данный модуль представляет собой представление страницы политика конфиденциальности |
| 18 | Error.cshtml | Данный модуль представляет собой представление страницы ошибок |
| 19 | \_Layout.cshtml | Данный модуль представляет собой представление базовой страницы |
| 20 | \_ValidationScriptsPartial.cshtml | Данный модуль представляет собой представление базовой страницы валидации |
| 21 | \_ViewImports.cshtml | Данный модуль имортирует нужные библиотеки |
| 22 | \_ViewStart.cshtml | Данный модуль представляет собой базовое представление |
| 23 | DatabaseContext.cs | Данный модуль представляет собой сервис для взаимодействия с БД |
| 24 | Program.cs | Данный модуль представляет собой главный класс. Используется для конфигурации приложения и его старта. |

* + 1. Функциональная схема

На рисунке ниже представлена функциональная схема приложения

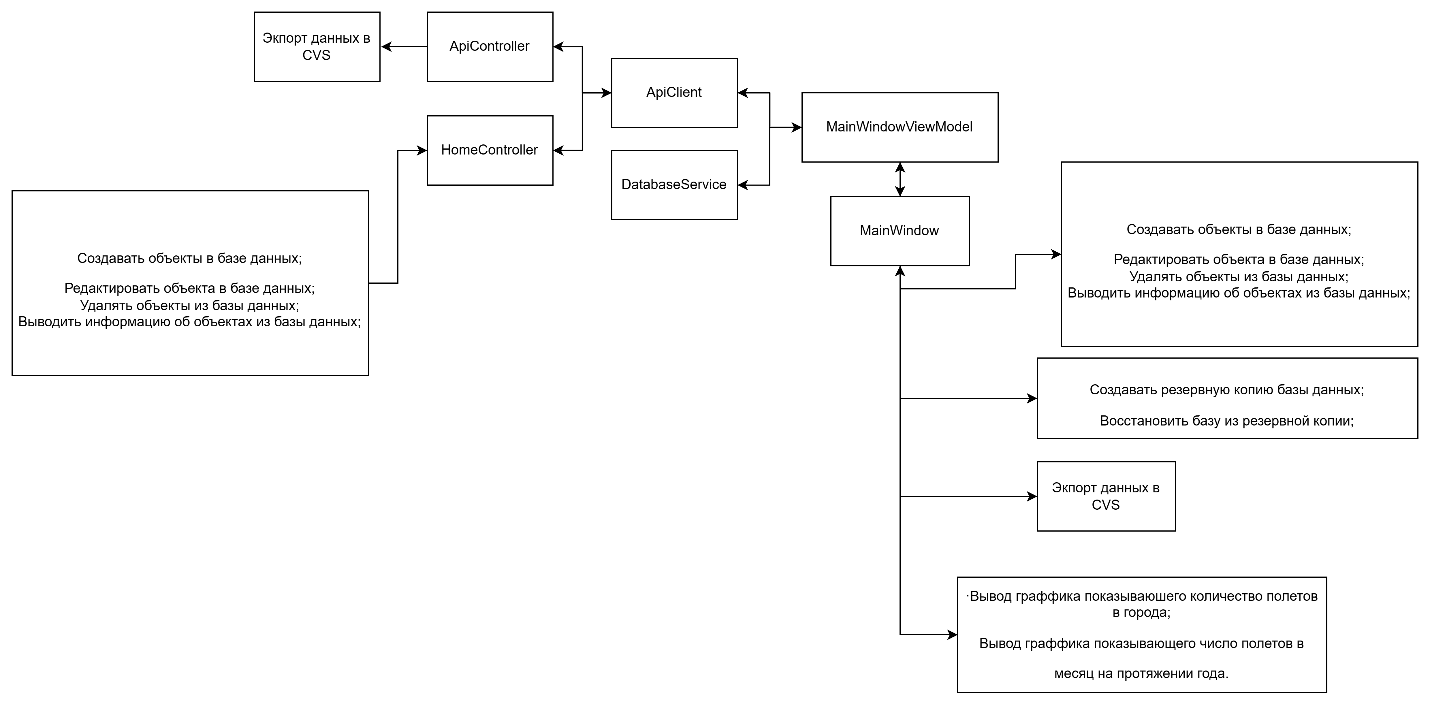


Рисунок 5 – функциональная схема

* + 1. Диаграмма классов

На рисунке ниже представлена диаграмма клиента

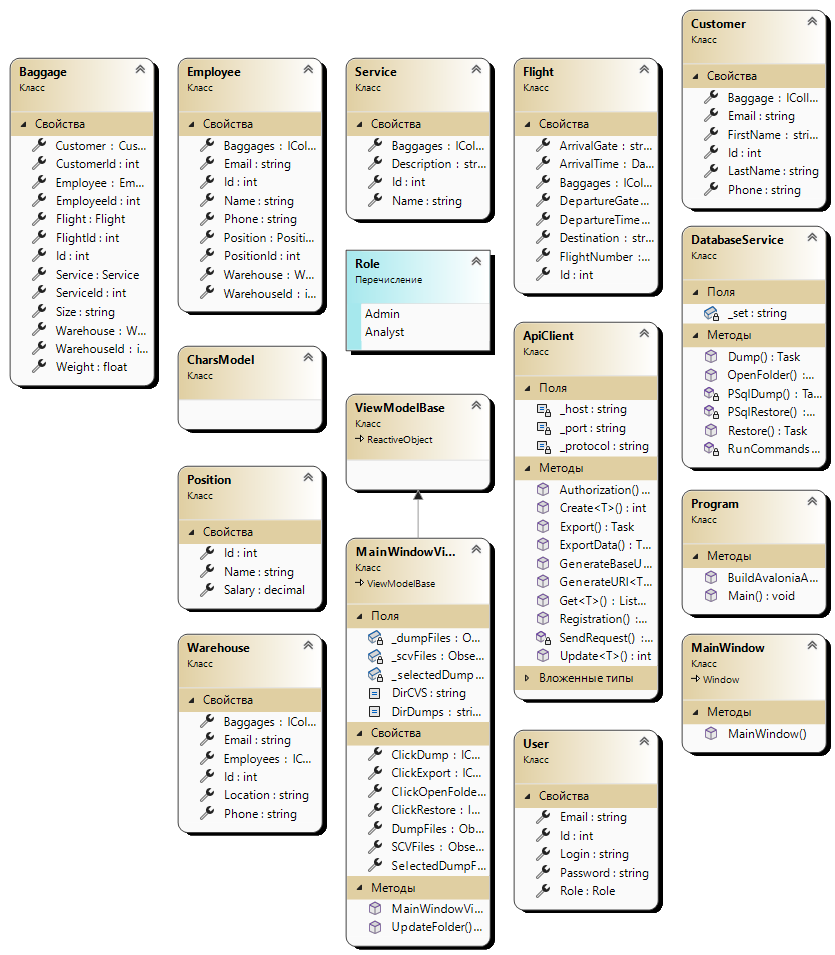


Рисунок 6 – диаграмма классов клиента

На рисунке ниже представлена диаграмма классов бэкенда

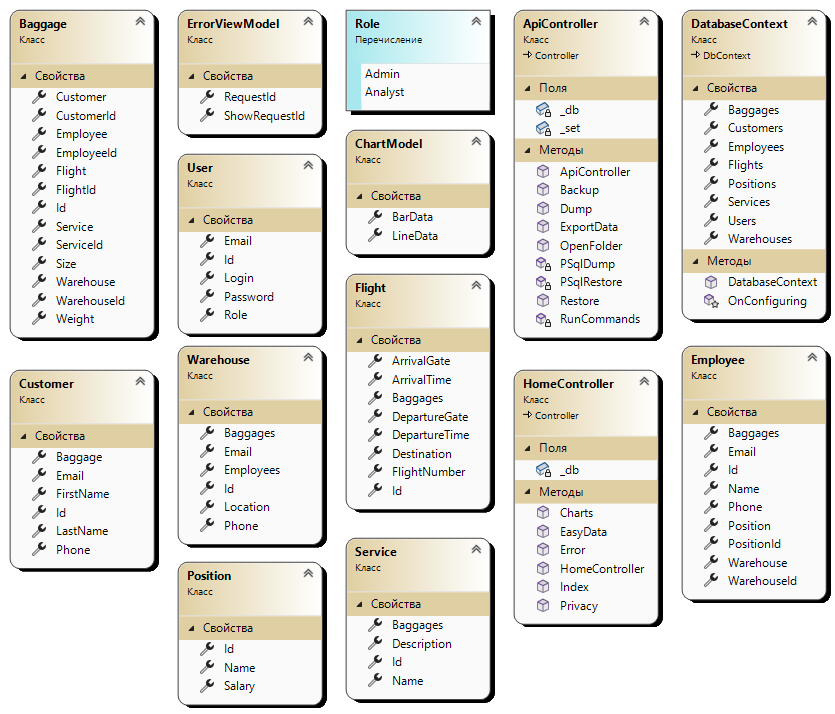


Рисунок – диаграмма классов бэкенда

* + 1. Схема тестирования

На рисунке ниже представлена схема тестирования приложения

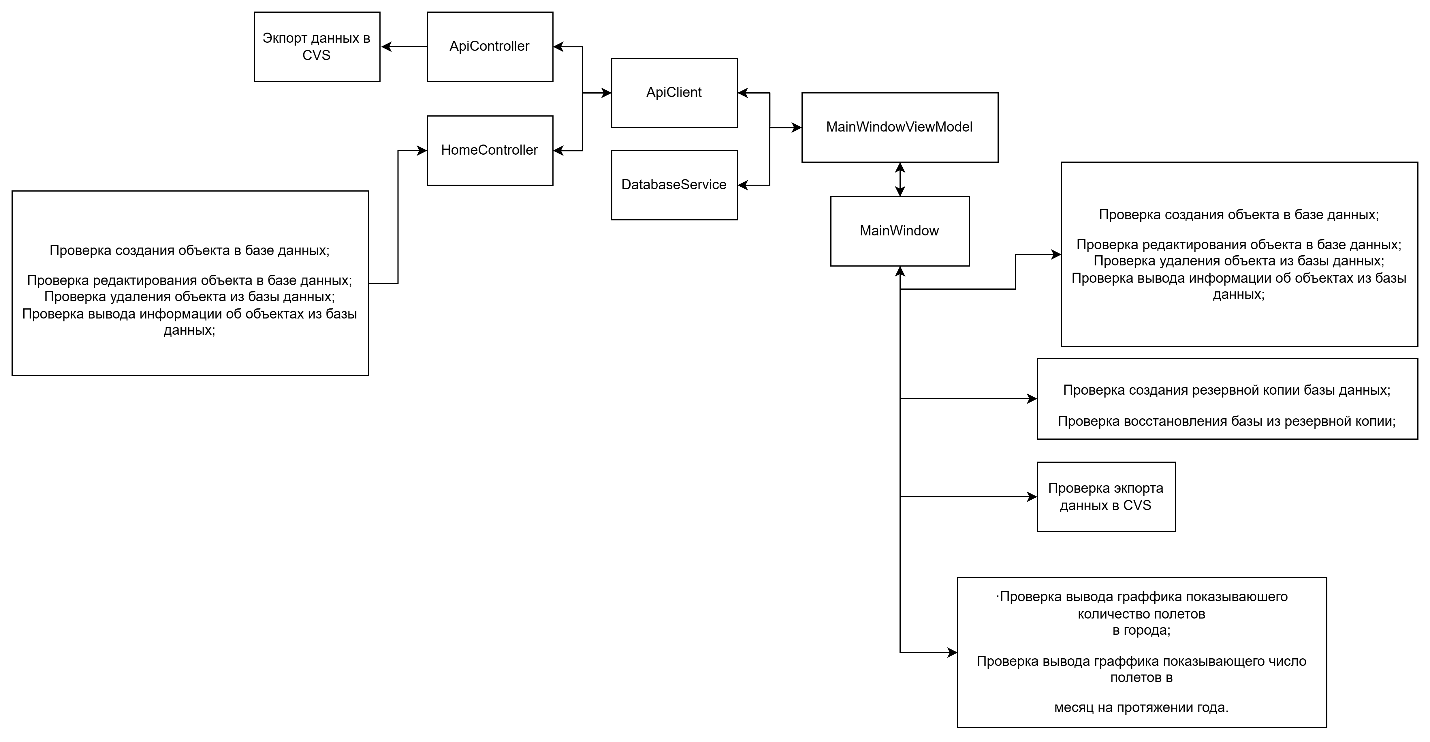


Рисунок 8 – схема тестирования приложения

# ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

* 1. Инструментальные средства разработки

Для разработки программного решения была выбрана быстрая и мощная кроссплатформенная IDE Visual studio 2022. Visual studio предоставляет более 2200 инспекций кода, сотни контекстных действий и рефакторингов, в сочетании с продвинутой функциональностью сред разработки. Именно за обширный набор инструментов была выбрана эта среда разработки.

Для администрирования базы данных PostgreSQL был выбран кроссплатформенный графический инструмент с открытым исходным кодом pgAdmin4. pgAdmin4 позволяет выполнять задачи мониторинга, обслуживания, конфигурирования сервера PostgreSQL, а также создавать и выполнять SQL-запросы.

За основной текстовый редактор был взят Microsoft Word 2019 года. Word позволяет создавать документы распространенного формата, которые каждый пользователь ПК сможет доступно прочитать и распечатать, что также повлияло на выбор.

* 1. Отладка программы

Отладка программы была проведена с помощью встроенных средств откладки в IDE Visual Studio. Отладка работает при включенном приложении и приостанавливается, доходя до точки остановки или в случае критичной ошибки. Тестирование приложения происходит по методике белого ящика.

Наиболее распространенными ошибками было взаимодействие приложения с API.

* 1. Защитное программирование
     1. Защита от ошибок

В коде предусмотрены защиты от ошибок с помощью проверок на пустоту, отсутствие директории и конструкций try catch. Пример защиты от ошибок с помощью проверки на отсутсвие директории представлен на рисунке ниже.

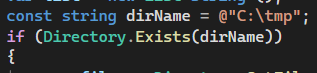


Рисунок – Защита от пустоты

На рисунке ниже представлен пример зашиты от ошибок с помощью конструкции try catch.

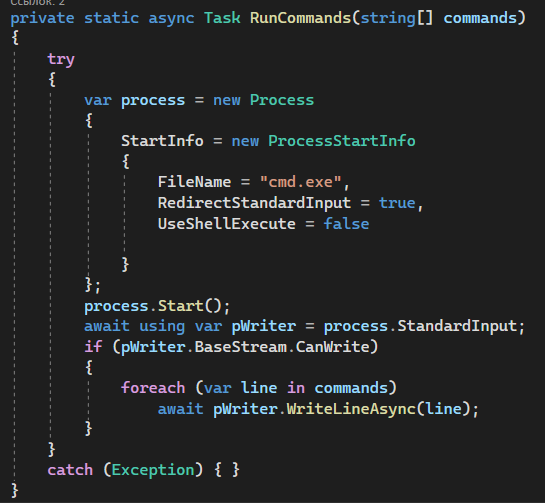


Рисунок – Защита с помощью конструкции try catch

* + 1. Защита приложения

Приложение также обеспечивает защиту от инъекций. На рисунке ниже представлена защита от инъекции.

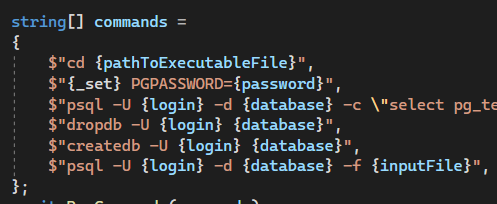


Рисунок 11 – Защита от инъекции

* 1. Характеристики программы

Характеристика программы представлена в таблице ниже.

Таблица – характеристика программы

| № | Название класса | Размер модуля в КБ | Используемые библиотеки | Дополнительные файлы |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Baggage.cs | 1 |  | Нет |
| 2 | CharsModel.cs | 1 |  | Нет |
| 3 | Customer.cs | 1 | using System.Collections.Generic; | Нет |
| 4 | Employee.cs | 1 | using System.Collections.Generic; | Нет |
| 5 | Flight.cs | 1 | using System; using System.Collections.Generic; | Нет |
| 6 | Position.cs | 1 |  | Нет |
| 7 | Role.cs | 1 |  | Нет |
| 8 | Service.cs | 1 | using System.Collections.Generic; | Нет |
| 9 | User.cs | 1 |  | Нет |
| 10 | Warehouse.cs | 1 | using System.Collections.Generic; | Нет |
| 11 | .NETCoreApp,Version=v6.0.AssemblyAttributes.cs | 1 |  | Нет |
| 12 | AirportBaggageCompartment.AssemblyInfo.cs | 1 |  | Нет |
| 13 | ApiClient.cs | 4 | using System; using System.Collections.Generic; using System.IO; using System.Linq; using System.Net; using System.Threading.Tasks; using AirportBaggageCompartment.Models; using Microsoft.AspNetCore.Identity; using Newtonsoft.Json; | Нет |
| 14 | DatabaseService.cs | 2 | using System; using System.Diagnostics; using System.Runtime.InteropServices; using System.Threading.Tasks; | Нет |
| 15 | MainWindowViewModel.cs | 2 | using System.Collections.ObjectModel; using System.Diagnostics; using System.IO; using System.Linq; using System.Threading.Tasks; using System.Windows.Input; using AirportBaggageCompartment.Services; using Avalonia.Threading; using ReactiveUI; | Нет |
| 16 | ViewModelBase.cs | 1 | using ReactiveUI; | Нет |
| 17 | MainWindow.axaml | 3 | - | Нет |
| 18 | MainWindow.axaml.cs | 1 | using Avalonia.Controls; | Нет |
| 19 | App.axaml | 1 |  | Нет |
| 20 | App.axaml.cs | 1 | using Avalonia; using Avalonia.Controls.ApplicationLifetimes; using Avalonia.Markup.Xaml; using AirportBaggageCompartment.ViewModels; using AirportBaggageCompartment.Views; | Нет |
| 21 | Program.cs | 1 | using Avalonia; using Avalonia.ReactiveUI; using System; | Нет |
| 22 | ViewLocator.cs | 1 | using System; using Avalonia.Controls; using Avalonia.Controls.Templates; using AirportBaggageCompartment.ViewModels; | Нет |
| 23 | .NETCoreApp,Version=v6.0.AssemblyAttributes.cs | 1 | - | Нет |
| 24 | ConsoleApp1.AssemblyInfo.cs | 1 | - | Нет |
| 25 | ConsoleApp1.GlobalUsings.g.cs | 1 | - | Нет |
| 26 | Program.cs | 1 | - | Нет |
| 27 | ApiController.cs | 3 | using System.Diagnostics; using System.Runtime.InteropServices; using Microsoft.AspNetCore.Mvc; using Microsoft.EntityFrameworkCore; | Нет |
| 28 | HomeController.cs | 1 | using System.Diagnostics; using Microsoft.AspNetCore.Mvc; using AirportBaggageCompartmentBackend.Models; | Нет |
| 29 | Baggage.cs | 1 | - | Нет |
| 30 | ChartModel.cs | 1 | - | Нет |
| 31 | Customer.cs | 1 | - | Нет |
| 32 | Employee.cs | 1 | - | Нет |
| 33 | ErrorViewModel.cs | 1 | - | Нет |
| 34 | Flight.cs | 1 | - | Нет |
| 35 | Position.cs | 1 | - | Нет |
| 36 | Role.cs | 1 | - | Нет |
| 37 | Service.cs | 1 | - | Нет |
| 38 | User.cs | 1 | - | Нет |
| 39 | Warehouse.cs | 1 | - | Нет |
| 40 | .NETCoreApp,Version=v6.0.AssemblyAttributes.cs | 1 | - | Нет |
| 41 | AirportBaggageCompartmentBackend.AssemblyInfo.cs | 1 | - | Нет |
| 42 | AirportBaggageCompartmentBackend.GlobalUsings.g.cs | 1 | - | Нет |
| 43 | AirportBaggageCompartmentBackend.RazorAssemblyInfo.cs | 1 | - | Нет |
| 44 | Charts.cshtml | 1 | - | Нет |
| 45 | EasyData.cshtml | 1 | - | Нет |
| 46 | Index.cshtml | 1 | - | Нет |
| 47 | Privacy.cshtml | 1 | - | Нет |
| 48 | Error.cshtml | 1 | - | Нет |
| 49 | \_Layout.cshtml | 1 | - | Нет |
| 50 | \_ValidationScriptsPartial.cshtml | 1 | - | Нет |
| 51 | \_ViewImports.cshtml | 1 | @using AirportBaggageCompartmentBackend @using AirportBaggageCompartmentBackend.Models | Нет |
| 52 | \_ViewStart.cshtml | 1 | - | Нет |
| 53 | DatabaseContext.cs | 1 | using AirportBaggageCompartmentBackend.Models; using Microsoft.EntityFrameworkCore; | Нет |
| 54 | Program.cs | 1 | using AirportBaggageCompartmentBackend; using EasyData.Services; using Microsoft.AspNetCore.Authentication.Cookies; | Нет |

# ЗАКЛЮЧЕНИЕ

После проведенной работы в разработке приложения на языке программирования C#, был создан программный комплекс состоящий из бэкенд части «AirportBaggageCompartmentBackend». И клиента: «AirportBaggageCompartment» цель использования которого заключается в предоставление руководителям аэропорта удобный инструмент для администрирования и анализа деятельности аэропорта.

В процессе разработки были достигнуты следующие цели: сбор, обработка и анализ информации, проектировка базы данных, разработка объектов базы данных на основе результатов анализа предметной области, реализация базы данных, администрация базы данных, защита информации в базе данных. Так же была выполнена отладка и тестирование программного продукта.

При разработки использовалась PgAdmin 4, инструментальное средство разработки программных решений Visual studio 2022, браузер для поиска информации.

В ходе использования программного комплекса пользователь сможет увидеть, что основной сценарий работает без сбоев, и готов к эксплуатации.

# СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ

1. Краткий справочник по минимальным API. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/fundamentals/minimal-apis?view=aspnetcore-7.0> Дата обращения: (01.11.2022);
2. Фильтры в приложениях с минимальным API. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/fundamentals/minimal-apis?view=aspnetcore-7.0> Дата обращения: (01.11.2022);
3. Размещение и развертывание ASP.NET Core: <https://learn.microsoft.com/ru-ru/aspnet/core/host-and-deploy/?view=aspnetcore-7.0> Дата обращения: (01.11.2022);
4. Использование Razor Pages с Entity Framework Core в ASP.NET Core: <https://learn.microsoft.com/ru-ru/aspnet/core/data/ef-rp/intro?view=aspnetcore-7.0&tabs=visual-studio>. Дата обращения: (01.11.2022);
5. Часть 2. Razor Страницы с EF Core ASP.NET Core — CRUD. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/data/ef-rp/crud?view=aspnetcore-7.0>. Дата обращения: (01.11.2022);
6. Часть 3. Razor Страницы с EF Core ASP.NET Core — сортировка, фильтрация, разбиение на страницы. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/data/ef-rp/sort-filter-page?view=aspnetcore-7.0>. Дата обращения: (01.11.2022);
7. Часть 4. Razor Страницы с EF Core миграцией в ASP.NET Core: [https://learn.microsoft.com/ru-ru/aspnet/core/data/ef rp/migrations?view=aspnetcore-7.0&tabs=visual-studio](https://learn.microsoft.com/ru-ru/aspnet/core/data/ef%20rp/migrations?view=aspnetcore-7.0&tabs=visual-studio). Дата обращения: (01.11.2022);
8. Часть 5. Razor Страницы с EF Core ASP.NET Core — модель данных. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/data/ef-rp/complex-data-model?view=aspnetcore-7.0&tabs=visual-studio>. Дата обращения: (01.11.2022);
9. Часть 6. Razor Страницы с EF Core ASP.NET Core — чтение связанных данных. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/data/ef-rp/read-related-data?view=aspnetcore-7.0&tabs=visual-studio>. Дата обращения: (01.11.2022);
10. Часть 7. Razor Страницы с EF Core ASP.NET Core — обновление связанных данных. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/data/ef-rp/update-related-data?view=aspnetcore-7.0> Дата обращения: (01.11.2022);
11. Часть 8. Razor Страницы с EF Core ASP.NET Core — параллелизм. URL:<https://learn.microsoft.com/ru-ru/aspnet/core/data/ef-rp/concurrency?view=aspnetcore-7.0&tabs=visual-studio>. Дата обращения: (01.11.2022);
12. Документация по веб-API ASP.NET Core. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-7.0>. Дата обращения: (01.11.2022);
13. Начало работы с NSwag и ASP.NET Core. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/getting-started-with-nswag?view=aspnetcore-7.0&tabs=visual-studio>. Дата обращения: (01.11.2022);
14. .NET справочник по командам средства OpenAPI и установка. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/web-api/microsoft.dotnet-openapi?view=aspnetcore-7.0>. Дата обращения: (01.11.2022);
15. Типы возвращаемых значений действий контроллера в веб-API ASP.NET Core. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/web-api/action-return-types?view=aspnetcore-7.0>. Дата обращения: (01.11.2022);
16. JsonPatch в веб-API ASP.NET Core. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/web-api/jsonpatch?view=aspnetcore-7.0>. Дата обращения: (01.11.2022);
17. Форматирование данных отклика в веб-API ASP.NET Core. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/web-api/advanced/formatting?view=aspnetcore-7.0>. Дата обращения: (01.11.2022);

# **ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ**

АННОТАЦИЯ

В данном программном документе «Текст программы» приведен исходный текст программного изделия для аэропорта

В документе представлен раздел “Текст программы” с подразделами:

1. Наименование программы
2. Область применения программы
3. Модули
4. Код программы

В подразделе “Наименование программы” содержится наименование программы, которая была создана для данной информационной системы.

В подразделе “Область применения программы” содержится описание того, для чего была предназначена программа.

В подразделе “Модули” содержится таблица с названием модуля, его описанием, количеством строк и размерами файла к КБайтах.

В подразделе “Код программы” содержится код каждого модуля, представленных в разделе “Модули”.

1. ТЕКСТ ПРОГРАММЫ
   1. Наименование программ

Наименование – «AirportBaggageCompartment».

* 1. Область применения программы

Программа AirportBaggageCompartment предназначена для контроля информационной системы багажного отделения аэропорта.

* 1. Модули

В Таблице ниже представлены модули для приложения AirportBaggageCompartment.

Таблица - Модули

| Модуль | Описание | Количество строк кода | Размер (КБ) |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Baggage.cs | Данный модуль представляет собой модель багажа | 23 | 1 |
| CharsModel.cs | Данный модуль представляет собой модель графиков | 6 | 1 |
| Customer.cs | Данный модуль представляет собой модель клиента | 14 | 1 |
| Employee.cs | Данный модуль представляет собой модель сотрудника | 19 | 1 |
| Flight.cs | Данный модуль представляет собой модель полета | 17 | 1 |
| Position.cs | Данный модуль представляет собой модель должности | 8 | 1 |
| Role.cs | Данный модуль представляет собой модель роли | 7 | 1 |
| Service.cs | Данный модуль представляет собой модель службы | 12 | 1 |
| User.cs | Данный модуль представляет собой модель пользователя | 10 | 1 |
| Warehouse.cs | Данный модуль представляет собой модель склада | 14 | 1 |
| ApiClient.cs | Данный модуль представляет собой Api клиента | 184 | 4 |
| DatabaseService.cs | Данный модуль представляет собой сервис для взаимодействия с БД | 86 | 2 |
| MainWindowViewModel.cs | Данный модуль представляет собой viewmodel главного окна | 76 | 2 |
| ViewModelBase.cs | Данный модуль представляет собой базовую view модель | 7 | 1 |
| MainWindow.axaml | Данный модуль представляет собой главное окно | 78 | 3 |
| MainWindow.axaml.cs | Данный модуль представляет собой класс для главного окна | 11 | 1 |
| App.axaml | Данный модуль представляет собой view приложения | 13 | 1 |
| App.axaml.cs | Данный модуль представляет собой логику приложения | 28 | 1 |
| Program.cs | Данный модуль представляет собой базовый класс | 22 | 1 |
| ViewLocator.cs | Данный модуль представляет локатор для View | 27 | 1 |

* 1. Код программы

Baggage.cs

namespace AirportBaggageCompartment.Models;

public class Baggage

{

public int Id { get; set; }

public float Weight { get; set; }

public string Size { get; set; }

public int FlightId { get; set; }

public virtual Flight Flight { get; set; }

public int CustomerId { get; set; }

public virtual Customer Customer { get; set; }

public int ServiceId { get; set; }

public virtual Service Service { get; set; }

public int WarehouseId { get; set; }

public virtual Warehouse Warehouse { get; set; }

public int EmployeeId { get; set; }

public virtual Employee Employee { get; set; }

}

CharsModel.cs

namespace AirportBaggageCompartment.Models;

public class CharsModel

{

}

Customer.cs

using System.Collections.Generic;

namespace AirportBaggageCompartment.Models;

public class Customer

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Phone { get; set; }

public string Email { get; set; }

public virtual ICollection<Baggage> Baggage { get; set; }

}

Employee.cs

using System.Collections.Generic;

namespace AirportBaggageCompartment.Models;

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public string Phone { get; set; }

public string Email { get; set; }

public int WarehouseId { get; set; }

public virtual Warehouse Warehouse { get; set; }

public virtual ICollection<Baggage> Baggages { get; set; }

public int PositionId { get; set; }

public virtual Position Position { get; set; }

}

Flight.cs

using System;

using System.Collections.Generic;

namespace AirportBaggageCompartment.Models;

public class Flight

{

public int Id { get; set; }

public int FlightNumber { get; set; }

public DateTime DepartureTime { get; set; }

public DateTime ArrivalTime { get; set; }

public string Destination { get; set; }

public string DepartureGate { get; set; }

public string ArrivalGate { get; set; }

public virtual ICollection<Baggage> Baggages { get; set; }

}

Position.cs

namespace AirportBaggageCompartment.Models;

public class Position

{

public int Id { get; set; }

public string Name { get; set; }

public decimal Salary { get; set; }

}

Role.cs

namespace AirportBaggageCompartment.Models;

public enum Role

{

Admin,

Analyst

}

Service.cs

using System.Collections.Generic;

namespace AirportBaggageCompartment.Models;

public class Service

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public virtual ICollection<Baggage> Baggages { get; set; }

}

User.cs

namespace AirportBaggageCompartment.Models;

public class User

{

public int Id { get; set; }

public string Login { get; set; }

public string Password { get; set; }

public string Email { get; set; }

public Role Role { get; set; }

}

Warehouse.cs

using System.Collections.Generic;

namespace AirportBaggageCompartment.Models;

public class Warehouse

{

public int Id { get; set; }

public string Location { get; set; }

public string Phone { get; set; }

public string Email { get; set; }

public virtual ICollection<Employee> Employees { get; set; }

public virtual ICollection<Baggage> Baggages { get; set; }

}

.NETCoreApp,Version=v6.0.AssemblyAttributes.cs

// <autogenerated />

using System;

using System.Reflection;

[assembly: global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETCoreApp,Version=v6.0", FrameworkDisplayName = ".NET 6.0")]

AirportBaggageCompartment.AssemblyInfo.cs

//------------------------------------------------------------------------------

// <auto-generated>

// Этот код создан программой.

// Исполняемая версия:4.0.30319.42000

//

// Изменения в этом файле могут привести к неправильной работе и будут потеряны в случае

// повторной генерации кода.

// </auto-generated>

//------------------------------------------------------------------------------

using System;

using System.Reflection;

[assembly: System.Reflection.AssemblyCompanyAttribute("AirportBaggageCompartment")]

[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]

[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]

[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0")]

[assembly: System.Reflection.AssemblyProductAttribute("AirportBaggageCompartment")]

[assembly: System.Reflection.AssemblyTitleAttribute("AirportBaggageCompartment")]

[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Создано классом WriteCodeFragment MSBuild.

ApiClient.cs

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Threading.Tasks;

using AirportBaggageCompartment.Models;

using Microsoft.AspNetCore.Identity;

using Newtonsoft.Json;

namespace AirportBaggageCompartment.Services;

public class ApiClient

{

private const string \_port = "44357";

private const string \_host = "localhost";

private const string \_protocol = "https";

public enum CRUD

{

create,

get,

update,

delete

}

public struct Method

{

public static string POST = "POST";

public static string GET = "GET";

public static string DELETE = "DELETE";

public static string PUT = "PUT";

}

public static string GenerateURI<T>(CRUD CRUD)

{

return $"{\_protocol}://{\_host}:{\_port}/api/easydata/models/\_\_default/sources/{typeof(T).Name}/{CRUD}";

}

public static string GenerateBaseURI()

{

return $"{\_protocol}://{\_host}:{\_port}/api";

}

static string SendRequest(string? body, string uri, string method)

{

try

{

var httpWebRequest = (HttpWebRequest)WebRequest.Create(uri);

httpWebRequest.ContentType = "application/json; charset=utf-8";

httpWebRequest.Method = method;

if (body?.Length > 0)

{

using (var streamWriter = new StreamWriter(httpWebRequest.GetRequestStream()))

{

streamWriter.Write(body);

streamWriter.Flush();

}

}

var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();

using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))

{

var result = streamReader.ReadToEnd();

return result;

}

}

catch (Exception ee)

{

return ee.ToString();

}

}

public static async Task Export()

{

await Task.Run(() => {

var url = $"{GenerateBaseURI()}/ExportData";

SendRequest(String.Empty, url, Method.GET);

});

}

public static List<T>? Get<T>()

{

try

{

string uri = GenerateURI<T>(CRUD.get);

string json = SendRequest(null, uri, Method.POST);

return JsonConvert.DeserializeObject<List<T>>(json);

}

catch(Exception e)

{

}

return null;

}

public static int Create<T>(T obj)

{

try

{

string json = JsonConvert.SerializeObject(obj, new JsonSerializerSettings

{

DateTimeZoneHandling = DateTimeZoneHandling.Utc

});

string uri = GenerateURI<T>(CRUD.create);

string jsonRequest = SendRequest(json, uri, Method.POST);

dynamic jsonObj = JsonConvert.DeserializeObject(jsonRequest);

return Convert.ToInt32(jsonObj["record"]["Id"]);

}

catch (Exception ex)

{

}

return 0;

}

public static int Update<T>(T obj)

{

try

{

string json = JsonConvert.SerializeObject(obj, new JsonSerializerSettings

{

DateTimeZoneHandling = DateTimeZoneHandling.Utc

});

string uri = GenerateURI<T>(CRUD.update);

string jsonRequest = SendRequest(json, uri, Method.POST);

dynamic jsonObj = JsonConvert.DeserializeObject(jsonRequest);

return Convert.ToInt32(jsonObj["record"]["Id"]);

}

catch (Exception ex)

{

}

return 0;

}

public async static Task ExportData()

{

await Task.Run(() => {

var url = $"http://{\_host}:{\_port}/home/ExportData";

SendRequest(String.Empty, url, Method.GET);

});

}

public static bool Registration(User user)

{

try

{

var newId = Create(user);

return true;

}

catch(Exception ex)

{

}

return false;

}

public static User? Authorization(string login, string password)

{

var listObjs = Get<User>();

var findUser = listObjs.FirstOrDefault(x => x.Login == login);

if (findUser != null)

{

var hasher = new PasswordHasher<User>();

var s = hasher.VerifyHashedPassword(findUser, findUser.Password, password);

if (s == PasswordVerificationResult.Success) return findUser;

}

return null;

}

}

DatabaseService.cs

using System;

using System.Diagnostics;

using System.Runtime.InteropServices;

using System.Threading.Tasks;

namespace AirportBaggageCompartment.Services;

public class DatabaseService

{

private static readonly string \_set = RuntimeInformation.IsOSPlatform(OSPlatform.Windows) ? "set " : "export ";

public static async Task Dump()

{

var date = DateTime.Now;

var fileName = $"{date.Year}{date.Month}{date.Day}\_{date.Hour}{date.Minute}";

await PSqlDump(

@"C:\Program Files\PostgreSQL\14\bin\",

"333",

"postgres",

"Airport",

$@"C:/tmp/{fileName}");

}

public static async Task Restore(string name)

{

await PSqlRestore(

@"C:\Program Files\PostgreSQL\14\bin\",

"333",

"postgres",

"Airport",

$"/tmp/{name}");

}

public static void OpenFolder()

{

Process.Start("explorer", @"C:\tmp\csv");

}

private static async Task PSqlDump(string pathToExecutableFile, string password, string login, string database, string outputFile)

{

string[] commands =

{

$"cd {pathToExecutableFile}",

$"{\_set} PGPASSWORD={password}",

$"pg\_dump.exe -U {login} {database} > {outputFile}.sql"

};

await RunCommands(commands);

}

private static async Task PSqlRestore(string pathToExecutableFile, string password, string login, string database, string inputFile)

{

string[] commands =

{

$"cd {pathToExecutableFile}",

$"{\_set} PGPASSWORD={password}",

$"psql -U {login} -d {database} -c \"select pg\_terminate\_backend(pid) from pg\_stat\_activity where datname = '{database}'",

$"dropdb -U {login} {database}",

$"createdb -U {login} {database}",

$"psql -U {login} -d {database} -f {inputFile}",

};

await RunCommands(commands);

}

private static async Task RunCommands(string[] commands)

{

var process = new Process

{

StartInfo = new ProcessStartInfo

{

FileName = "cmd.exe",

RedirectStandardInput = true,

UseShellExecute = false

}

};

process.Start();

await using var pWriter = process.StandardInput;

if (pWriter.BaseStream.CanWrite)

{

foreach (var line in commands)

await pWriter.WriteLineAsync(line);

}

}

}

MainWindowViewModel.cs

using System.Collections.ObjectModel;

using System.Diagnostics;

using System.IO;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Input;

using AirportBaggageCompartment.Services;

using Avalonia.Threading;

using ReactiveUI;

namespace AirportBaggageCompartment.ViewModels;

public class MainWindowViewModel : ViewModelBase

{

public const string DirDumps = @"C:\tmp";

public const string DirCVS = @"C:\tmp\csv";

private ObservableCollection<string> \_dumpFiles = new ObservableCollection<string>();

private ObservableCollection<string> \_scvFiles = new ObservableCollection<string>();

public ObservableCollection<string> DumpFiles { get { return \_dumpFiles; } set { \_dumpFiles = value; } }

public ObservableCollection<string> SCVFiles { get { return \_scvFiles; } set { \_scvFiles = value; } }

public ICommand ClickDump { get; private set; }

public ICommand ClickRestore { get; private set; }

public ICommand ClickExport { get; private set; }

public ICommand ClickOpenFolder { get; private set; }

private string \_selectedDumpFile;

public string SelectedDumpFile { get { return \_selectedDumpFile; } set { \_selectedDumpFile = value; } }

public async void UpdateFolder()

{

await Task.Run(async () => {

while (true)

{

var filesDumps = Directory.GetFiles(DirDumps);

var filesCVS = Directory.GetFiles(DirCVS);

DumpFiles.Clear();

SCVFiles.Clear();

var listDumps = filesDumps.Select(Path.GetFileName)!.Where(x => x.Contains(".sql"));

var listCSV = filesCVS.Select(Path.GetFileName)!.Where(x => x.Contains(".csv"));

foreach (var s in listDumps) DumpFiles.Add(s!);

foreach (var s in listCSV) SCVFiles.Add(s!);

await Task.Delay(1000);

}

});

}

public MainWindowViewModel()

{

ClickDump = ReactiveCommand.CreateFromTask(async () =>

{

await DatabaseService.Dump();

});

ClickRestore = ReactiveCommand.CreateFromTask(async () =>

{

await DatabaseService.Restore(SelectedDumpFile);

});

ClickExport = ReactiveCommand.CreateFromTask(async () =>

{

await ApiClient.Export();

});

ClickOpenFolder = ReactiveCommand.CreateFromTask(async () =>

{

Process.Start("explorer", DirCVS);

});

UpdateFolder();

}

}

ViewModelBase.cs

using ReactiveUI;

namespace AirportBaggageCompartment.ViewModels;

public class ViewModelBase : ReactiveObject

{

}

MainWindow.axaml

<Window xmlns="https://github.com/avaloniaui"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:vm="using:AirportBaggageCompartment.ViewModels"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:webview="clr-namespace:WebViewControl;assembly=WebViewControl.Avalonia"

mc:Ignorable="d" d:DesignWidth="800" d:DesignHeight="450"

x:Class="AirportBaggageCompartment.Views.MainWindow"

x:DataType="vm:MainWindowViewModel"

Icon="/Assets/avalonia-logo.ico"

Title="AirportBaggageCompartment">

<Design.DataContext>

<!-- This only sets the DataContext for the previewer in an IDE,

to set the actual DataContext for runtime, set the DataContext property in code (look at App.axaml.cs) -->

<vm:MainWindowViewModel/>

</Design.DataContext>

<Grid>

<TabControl>

<TabItem Header="База данных">

<webview:WebView x:Name="webview" Address="https://localhost:44357/Home/EasyData"/>

</TabItem>

<TabItem Header="Резервное копирование">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="250"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="50"/>

<RowDefinition Height="50"/>

<RowDefinition/>

</Grid.RowDefinitions>

<Button Grid.Row="0" Command="{Binding ClickDump}" Content="Создать резервную копию"/>

<Button Grid.Row="1" Command="{Binding ClickRestore}" Content="Восстановить"/>

</Grid>

<Grid Grid.Column="1">

<ListBox SelectedItem="{Binding SelectedDumpFile}" Items="{Binding DumpFiles}"/>

</Grid>

</Grid>

</TabItem>

<TabItem Header="Выгрузка данных">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="250"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="50"/>

<RowDefinition Height="50"/>

<RowDefinition/>

</Grid.RowDefinitions>

<Button Grid.Row="0" Command="{Binding ClickExport}" Content="Выгрузить данные"/>

<Button Grid.Row="1" Command="{Binding ClickOpenFolder}" Content="Открыть папку"/>

</Grid>

<Grid Grid.Column="1">

<ListBox Items="{Binding SCVFiles}"/>

</Grid>

</Grid>

</TabItem>

<TabItem Header="Диаграммы">

<webview:WebView x:Name="webview2" Address="https://localhost:44357/Home/Charts"/>

</TabItem>

</TabControl>

</Grid>

</Window>

MainWindow.axaml.cs

using Avalonia.Controls;

namespace AirportBaggageCompartment.Views;

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

}

App.axaml

<Application xmlns="https://github.com/avaloniaui"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:AirportBaggageCompartment"

x:Class="AirportBaggageCompartment.App">

<Application.DataTemplates>

<local:ViewLocator/>

</Application.DataTemplates>

<Application.Styles>

<FluentTheme Mode="Light"/>

</Application.Styles>

</Application>

App.axaml.cs

using Avalonia;

using Avalonia.Controls.ApplicationLifetimes;

using Avalonia.Markup.Xaml;

using AirportBaggageCompartment.ViewModels;

using AirportBaggageCompartment.Views;

namespace AirportBaggageCompartment;

public partial class App : Application

{

public override void Initialize()

{

AvaloniaXamlLoader.Load(this);

}

public override void OnFrameworkInitializationCompleted()

{

if (ApplicationLifetime is IClassicDesktopStyleApplicationLifetime desktop)

{

desktop.MainWindow = new MainWindow

{

DataContext = new MainWindowViewModel(),

};

}

base.OnFrameworkInitializationCompleted();

}

}

Program.cs

using Avalonia;

using Avalonia.ReactiveUI;

using System;

namespace AirportBaggageCompartment;

class Program

{

// Initialization code. Don't use any Avalonia, third-party APIs or any

// SynchronizationContext-reliant code before AppMain is called: things aren't initialized

// yet and stuff might break.

[STAThread]

public static void Main(string[] args) => BuildAvaloniaApp()

.StartWithClassicDesktopLifetime(args);

// Avalonia configuration, don't remove; also used by visual designer.

public static AppBuilder BuildAvaloniaApp()

=> AppBuilder.Configure<App>()

.UsePlatformDetect()

.LogToTrace()

.UseReactiveUI();

}

ViewLocator.cs

using System;

using Avalonia.Controls;

using Avalonia.Controls.Templates;

using AirportBaggageCompartment.ViewModels;

namespace AirportBaggageCompartment;

public class ViewLocator : IDataTemplate

{

public IControl Build(object data)

{

var name = data.GetType().FullName!.Replace("ViewModel", "View");

var type = Type.GetType(name);

if (type != null)

{

return (Control)Activator.CreateInstance(type)!;

}

return new TextBlock { Text = "Not Found: " + name };

}

public bool Match(object data)

{

return data is ViewModelBase;

}

}

.NETCoreApp,Version=v6.0.AssemblyAttributes.cs

// <autogenerated />

using System;

using System.Reflection;

[assembly: global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETCoreApp,Version=v6.0", FrameworkDisplayName = ".NET 6.0")]

ConsoleApp1.AssemblyInfo.cs

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

using System;

using System.Reflection;

[assembly: System.Reflection.AssemblyCompanyAttribute("ConsoleApp1")]

[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]

[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]

[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0")]

[assembly: System.Reflection.AssemblyProductAttribute("ConsoleApp1")]

[assembly: System.Reflection.AssemblyTitleAttribute("ConsoleApp1")]

[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Создано классом WriteCodeFragment MSBuild.

ConsoleApp1.GlobalUsings.g.cs

// <auto-generated/>

global using global::System;

global using global::System.Collections.Generic;

global using global::System.IO;

global using global::System.Linq;

global using global::System.Net.Http;

global using global::System.Threading;

global using global::System.Threading.Tasks;

Program.cs

// See https://aka.ms/new-console-template for more information

Console.WriteLine("Hello, World!");

ApiController.cs

using System.Diagnostics;

using System.Runtime.InteropServices;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

namespace AirportBaggageCompartmentBackend.Controllers;

public class ApiController : Controller

{

private readonly DatabaseContext \_db;

private readonly string \_set = RuntimeInformation.IsOSPlatform(OSPlatform.Windows) ? "set " : "export ";

public ApiController(DatabaseContext db)

{

\_db = db;

}

public List<string> Backup()

{

var list = new List<string>();

const string dirName = @"C:\tmp";

if (Directory.Exists(dirName))

{

var files = Directory.GetFiles(dirName);

list.AddRange(files.Select(Path.GetFileName)!.Where(x => x!.Contains("sql"))!);

}

else

Directory.CreateDirectory(@"C:\tmp");

return list;

}

public async Task ExportData()

{

const string dirName = @"C:\tmp\csv";

if (!Directory.Exists(dirName)) Directory.CreateDirectory(@"C:\tmp\csv");

await \_db.Database.ExecuteSqlRawAsync("call BackupCSV();");

}

public async Task Dump()

{

var date = DateTime.Now;

var fileName = $"{date.Year}{date.Month}{date.Day}\_{date.Hour}{date.Minute}";

await PSqlDump(

@"C:\Program Files\PostgreSQL\14\bin\",

"333",

"postgres",

"Airport",

$@"C:/tmp/{fileName}");

}

public async Task Restore(string name)

{

await PSqlRestore(

@"C:\Program Files\PostgreSQL\14\bin\",

"333",

"postgres",

"Airport",

$"/tmp/{name}");

}

public void OpenFolder()

{

Process.Start("explorer", @"C:\tmp\csv");

}

private async Task PSqlDump(string pathToExecutableFile, string password, string login, string database, string outputFile)

{

string[] commands =

{

$"cd {pathToExecutableFile}",

$"{\_set} PGPASSWORD={password}",

$"pg\_dump.exe -U {login} {database} > {outputFile}.sql"

};

await RunCommands(commands);

}

private async Task PSqlRestore(string pathToExecutableFile, string password, string login, string database, string inputFile)

{

string[] commands =

{

$"cd {pathToExecutableFile}",

$"{\_set} PGPASSWORD={password}",

$"psql -U {login} -d {database} -c \"select pg\_terminate\_backend(pid) from pg\_stat\_activity where datname = '{database}'",

$"dropdb -U {login} {database}",

$"createdb -U {login} {database}",

$"psql -U {login} -d {database} -f {inputFile}",

};

await RunCommands(commands);

}

private static async Task RunCommands(string[] commands)

{

var process = new Process

{

StartInfo = new ProcessStartInfo

{

FileName = "cmd.exe",

RedirectStandardInput = true,

UseShellExecute = false

}

};

process.Start();

await using var pWriter = process.StandardInput;

if (pWriter.BaseStream.CanWrite)

{

foreach (var line in commands)

await pWriter.WriteLineAsync(line);

}

}

}

HomeController.cs

using System.Diagnostics;

using Microsoft.AspNetCore.Mvc;

using AirportBaggageCompartmentBackend.Models;

namespace AirportBaggageCompartmentBackend.Controllers;

public class HomeController : Controller

{

private readonly DatabaseContext \_db;

public HomeController(DatabaseContext db)

{

\_db = db;

}

public IActionResult EasyData() => View();

public IActionResult Charts()

{

var model = new ChartModel();

var barData = new Dictionary<string, int>();

var lineData = new Dictionary<string, int>();

var listFlightDestination = \_db.Flights.Select(x => x.Destination).Distinct().ToList();

listFlightDestination.ForEach(destination =>

{

var count = \_db.Flights.Where(x => x.Destination == destination).Count();

barData.Add(destination, count);

});

var date = new DateTime(2023, 1, 1);

while (date.Month < 12)

{

var label = date.ToString("MMMM");

var data = \_db.Flights.Count(x => x.ArrivalTime.Month == date.Month);

lineData.Add(label, data);

date = date.AddMonths(1);

}

model.BarData = barData;

model.LineData = lineData;

return View(model);

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

Baggage.cs

namespace AirportBaggageCompartmentBackend.Models;

public class Baggage

{

public int Id { get; set; }

public float Weight { get; set; }

public string Size { get; set; }

public int FlightId { get; set; }

public virtual Flight Flight { get; set; }

public int CustomerId { get; set; }

public virtual Customer Customer { get; set; }

public int ServiceId { get; set; }

public virtual Service Service { get; set; }

public int WarehouseId { get; set; }

public virtual Warehouse Warehouse { get; set; }

public int EmployeeId { get; set; }

public virtual Employee Employee { get; set; }

}

ChartModel.cs

namespace AirportBaggageCompartmentBackend.Models;

public class ChartModel

{

public Dictionary<string, int> BarData { get; set; }

public Dictionary<string, int> LineData { get; set; }

}

Customer.cs

namespace AirportBaggageCompartmentBackend.Models;

public class Customer

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Phone { get; set; }

public string Email { get; set; }

public virtual ICollection<Baggage> Baggage { get; set; }

}

Employee.cs

namespace AirportBaggageCompartmentBackend.Models;

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public string Phone { get; set; }

public string Email { get; set; }

public int WarehouseId { get; set; }

public virtual Warehouse Warehouse { get; set; }

public virtual ICollection<Baggage> Baggages { get; set; }

public int PositionId { get; set; }

public virtual Position Position { get; set; }

}

ErrorViewModel.cs

namespace AirportBaggageCompartmentBackend.Models;

public class ErrorViewModel

{

public string? RequestId { get; set; }

public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

}

Flight.cs

namespace AirportBaggageCompartmentBackend.Models;

public class Flight

{

public int Id { get; set; }

public string FlightNumber { get; set; }

public DateTime DepartureTime { get; set; }

public DateTime ArrivalTime { get; set; }

public string Destination { get; set; }

public string DepartureGate { get; set; }

public string ArrivalGate { get; set; }

public virtual ICollection<Baggage> Baggages { get; set; }

}

Position.cs

namespace AirportBaggageCompartmentBackend.Models;

public class Position

{

public int Id { get; set; }

public string Name { get; set; }

public decimal Salary { get; set; }

}

Role.cs

namespace AirportBaggageCompartmentBackend.Models;

public enum Role

{

Admin,

Analyst

}

Service.cs

namespace AirportBaggageCompartmentBackend.Models;

public class Service

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public virtual ICollection<Baggage> Baggages { get; set; }

}

User.cs

namespace AirportBaggageCompartmentBackend.Models;

public class User

{

public int Id { get; set; }

public string Login { get; set; }

public string Password { get; set; }

public string Email { get; set; }

public Role Role { get; set; }

}

Warehouse.cs

namespace AirportBaggageCompartmentBackend.Models;

public class Warehouse

{

public int Id { get; set; }

public string Location { get; set; }

public string Phone { get; set; }

public string Email { get; set; }

public virtual ICollection<Employee> Employees { get; set; }

public virtual ICollection<Baggage> Baggages { get; set; }

}

.NETCoreApp,Version=v6.0.AssemblyAttributes.cs

// <autogenerated />

using System;

using System.Reflection;

[assembly: global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETCoreApp,Version=v6.0", FrameworkDisplayName = "")]

AirportBaggageCompartmentBackend.AssemblyInfo.cs

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

using System;

using System.Reflection;

[assembly: System.Reflection.AssemblyCompanyAttribute("AirportBaggageCompartmentBackend")]

[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]

[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]

[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0")]

[assembly: System.Reflection.AssemblyProductAttribute("AirportBaggageCompartmentBackend")]

[assembly: System.Reflection.AssemblyTitleAttribute("AirportBaggageCompartmentBackend")]

[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Создано классом WriteCodeFragment MSBuild.

AirportBaggageCompartmentBackend.GlobalUsings.g.cs

// <auto-generated/>

global using global::Microsoft.AspNetCore.Builder;

global using global::Microsoft.AspNetCore.Hosting;

global using global::Microsoft.AspNetCore.Http;

global using global::Microsoft.AspNetCore.Routing;

global using global::Microsoft.Extensions.Configuration;

global using global::Microsoft.Extensions.DependencyInjection;

global using global::Microsoft.Extensions.Hosting;

global using global::Microsoft.Extensions.Logging;

global using global::System;

global using global::System.Collections.Generic;

global using global::System.IO;

global using global::System.Linq;

global using global::System.Net.Http;

global using global::System.Net.Http.Json;

global using global::System.Threading;

global using global::System.Threading.Tasks;

AirportBaggageCompartmentBackend.RazorAssemblyInfo.cs

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

using System;

using System.Reflection;

[assembly: Microsoft.AspNetCore.Mvc.ApplicationParts.ProvideApplicationPartFactoryAttribute("Microsoft.AspNetCore.Mvc.ApplicationParts.ConsolidatedAssemblyApplicationPartFact" +

"ory, Microsoft.AspNetCore.Mvc.Razor")]

// Создано классом WriteCodeFragment MSBuild.

Charts.cshtml

@model ChartModel

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<div>

<div style="margin: 20px; align-content: center">

<h1>Количество полетов в города</h1>

</div>

<div class="card-body">

<canvas id="BarChart" max-height="100" max-width="100"></canvas>

</div>

<div style="margin: 20px">

<h1>Число полетов за год</h1>

</div>

<div class="card-body">

<canvas id="LineChart" height="100" width="100"></canvas>

</div>

</div>

<script>

const barCTX = document.getElementById('BarChart');

const lineCTX = document.getElementById('LineChart');

var BarLabels = [];

var BarData = [];

var LineLabels = [];

var LineData = [];

@{

foreach (var i in Model.BarData)

{

@:BarLabels.push('@Html.Raw(i.Key)');

@:BarData.push(@i.Value);

}

foreach (var i in Model.LineData)

{

@:LineLabels.push('@Html.Raw(i.Key)');

@:LineData.push(@i.Value);

}

}

new Chart(barCTX, {

type: 'bar',

data: {

labels: BarLabels,

datasets: [{

label: 'Количество полётов',

data: BarData,

borderWidth: 1

}]

},

});

new Chart(lineCTX, {

type: 'line',

data: {

labels: LineLabels,

datasets: [{

label: 'Количество бронирований',

data: LineData,

hoverOffset: 4

}]

},

fill: false,

borderColor: 'rgb(75, 192, 192)',

tension: 0.1

});

</script>

EasyData.cshtml

<link rel="stylesheet" href="https://cdn.korzh.com/ed/1.4.8/easydata.min.css" />

<div id="EasyDataContainer"></div>

@section Scripts {

<script src="https://cdn.korzh.com/ed/1.4.8/easydata.min.js" type="text/javascript"></script>

<script>

window.addEventListener('load', function () {

new easydata.crud.EasyDataViewDispatcher().run()

});

</script>

}

Index.cshtml

@{

Layout = "\_Layout";

}

<link rel="stylesheet" href="https://cdn.korzh.com/ed/1.4.8/easydata.min.css" />

<div id="EasyDataContainer"></div>

@section Scripts {

<script src="https://cdn.korzh.com/ed/1.4.8/easydata.min.js" type="text/javascript"></script>

<script>

window.addEventListener('load', function () {

new easydata.crud.EasyDataViewDispatcher().run()

});

</script>

}

Privacy.cshtml

@{

ViewData["Title"] = "Privacy Policy";

}

<h1>@ViewData["Title"]</h1>

<p>Use this page to detail your site's privacy policy.</p>

Error.cshtml

@model ErrorViewModel

@{

ViewData["Title"] = "Error";

}

<h1 class="text-danger">Error.</h1>

<h2 class="text-danger">An error occurred while processing your request.</h2>

@if (Model.ShowRequestId)

{

<p>

<strong>Request ID:</strong> <code>@Model.RequestId</code>

</p>

}

<h3>Development Mode</h3>

<p>

Swapping to <strong>Development</strong> environment will display more detailed information about the error that occurred.

</p>

<p>

<strong>The Development environment shouldn't be enabled for deployed applications.</strong>

It can result in displaying sensitive information from exceptions to end users.

For local debugging, enable the <strong>Development</strong> environment by setting the <strong>ASPNETCORE\_ENVIRONMENT</strong> environment variable to <strong>Development</strong>

and restarting the app.

</p>

\_Layout.cshtml

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>@ViewData["Title"] - AirportBaggageCompartmentBackend</title>

<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css"/>

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true"/>

<link rel="stylesheet" href="~/AirportBaggageCompartmentBackend.styles.css" asp-append-version="true"/>

</head>

<body>

<div class="container">

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2023 - Аэропорт

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

\_ValidationScriptsPartial.cshtml

<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>

<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>

\_ViewImports.cshtml

@using AirportBaggageCompartmentBackend

@using AirportBaggageCompartmentBackend.Models

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

\_ViewStart.cshtml

@{

Layout = "\_Layout";

}

DatabaseContext.cs

using AirportBaggageCompartmentBackend.Models;

using Microsoft.EntityFrameworkCore;

namespace AirportBaggageCompartmentBackend;

public class DatabaseContext : DbContext

{

public DbSet<Baggage> Baggages { get; set; }

public DbSet<Customer> Customers { get; set; }

public DbSet<Employee> Employees { get; set; }

public DbSet<Flight> Flights { get; set; }

public DbSet<Position> Positions { get; set; }

public DbSet<Service> Services { get; set; }

public DbSet<Warehouse> Warehouses { get; set; }

public DbSet<User> Users { get; set; }

public DatabaseContext()

{

try

{

Database.EnsureCreated();

}

catch (Exception)

{

// ignored

}

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseNpgsql(

"host=localhost;port=5432;database=Airport;username=postgres;password=333");

}

}

Program.cs

using AirportBaggageCompartmentBackend;

using EasyData.Services;

using Microsoft.AspNetCore.Authentication.Cookies;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddDbContext<DatabaseContext>();

// установка конфигурации подключения

builder.Services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)

.AddCookie(options => //CookieAuthenticationOptions

{

options.LoginPath = new PathString("/Account/Authorization");

});

builder.Services.AddDistributedMemoryCache();

builder.Services.AddSession();

// Add services to the container.

builder.Services.AddControllersWithViews();

var app = builder.Build();

app.MapEasyData(options => {

options.UseDbContext<DatabaseContext>();

});

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication(); // аутентификация

app.UseAuthorization(); // авторизация

app.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

# **ПРИЛОЖЕНИЕ Б. СЦЕНАРИЙ ТЕСТОВ И РЕЗУЛЬТАТЫ ТЕСТОВЫХ ИСПЫТАНИЙ**

АННОТАЦИЯ

В данном программном документе приведены сценарии тестовых испытаний сервиса для контроля информационной системы багажного отделения аэропорта.

В разделе «Объекты испытаний» указаны программные продукты, над которыми проводилось тестирование

В разделе «Цель испытаний» указана цель проведения испытаний.

В разделе «Требования к программе» указаны требования, подлежащие проверке во время испытаний и заданные в техническом задании на программу.

В разделе «Средства и порядок испытаний» указаны технические и программные средства, используемые во время испытаний, а также порядок проведения испытаний.

В разделе «Методы испытаний» приведено описание используемых методов испытаний.

В разделе «Тестовые примеры» приведены таблицы с тестовыми данными для проведения тестирования.

В разделе «Результаты тестирования» приведены таблицы результатов тестирования в соответствии раздела «Тестовые примеры».

1. ОБЪЕКТЫ ИСПЫТАНИЙ
   1. Наименование объектов

Наименование – «AirportBaggageCompartment»

* 1. Область AirportBaggageCompartment объектов

Программа Otel предназначена для контроля информационной системы багажного отделения аэропорта.

1. ЦЕЛЬ ИСПЫТАНИЙ

Целью проведения испытаний является проверка соответствия разработанной программы требованиям, изложенным в пункте «Специальная часть».

1. ТРЕБОВАНИЯ К ПРОГРАММЕ
   1. Схема тестирования AirportBaggageCompartment

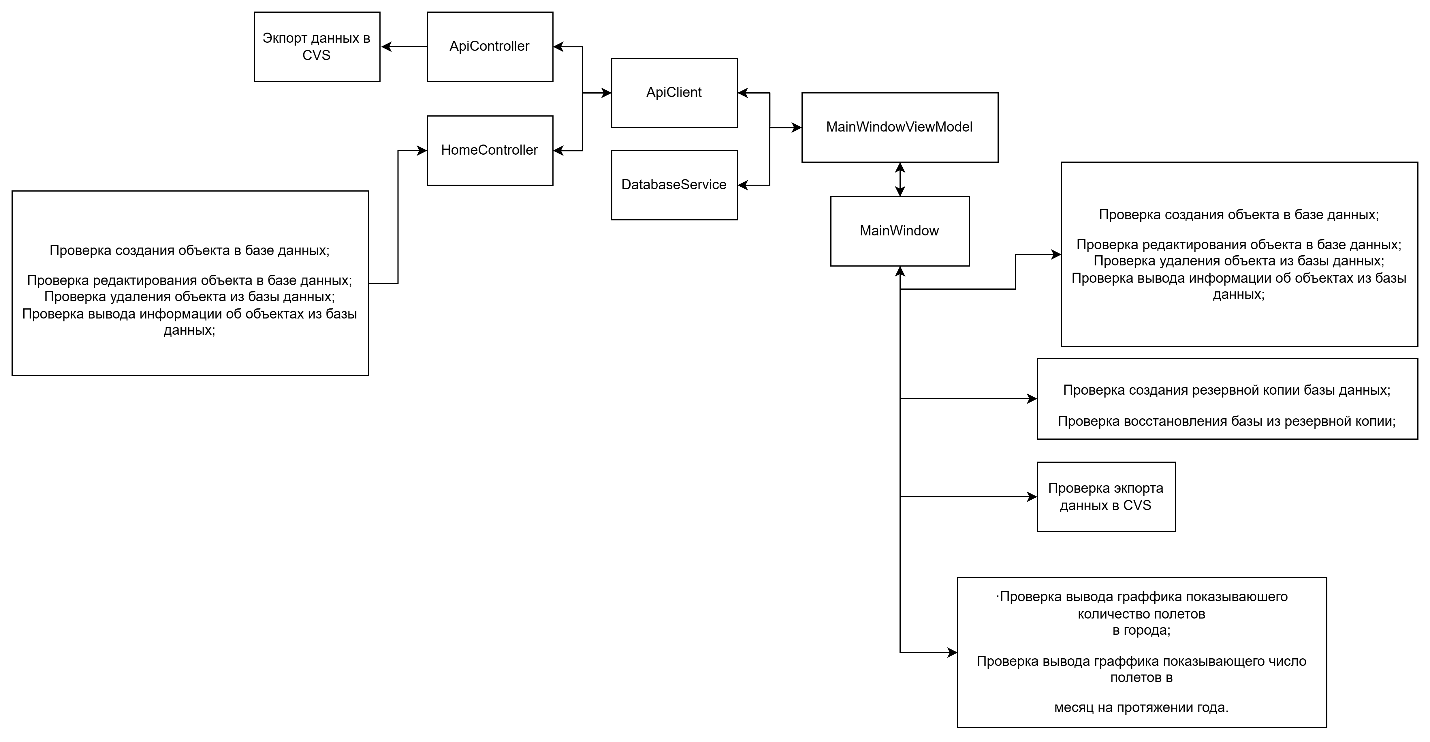


Рисунок 12 - Схема тестирования

1. ТЕСТОВЫЕ ПРИМЕРЫ

Все тестовые данные для проведения тестирования приложения представлены в таблицах 8 - 14.

* 1. Проверка создавания обьектов в базе данных

Для успешного прохожденния данного тест кейса необходимо заполнить форму данными и нажать на кнопку «добавить».

Таблица – Проверка создания обьектов в базе данных

| № п/п | Действие (Входное значение) | Ожидаемый результат |
| --- | --- | --- |
| 1 | Нажата кнопка «+»  Нажата кнопку «добавить» | Добавление не пройдено. Сообщение об ошибке: «Пожалуйста заполните поля». |
| 2 | Нажата кнопка «+»  Заполнены поля без «\*»  Не заполнены поля с «\*»  Нажата кнопку «добавить» | Добавление не пройдено. Сообщение об ошибке: «Пожалйства заполните поля». |
| 3 | Нажата кнопка «+»  Не заполнены поля без «\*»  Заполнены поля с «\*»  Нажата кнопку «добавить» | Добавление пройдено. |
| 4 | Нажата кнопка «+»  Все поля заполнены  Нажата кнопку «добавить» | Добавление пройдено. |

* 1. Проверка редактирования обьектов в базе данных

Для успешного прохожденния данного тест кейса необходимо заполнить форму данными и нажать на кнопку «изменить».

Таблица – Проверка редактирования обьектов в базе данных

| № п/п | Действие (Входное значение) | Ожидаемый результат |
| --- | --- | --- |
| 1 | Нажата кнопка «редактировать»  Нажата кнопку «подтвердить» | Редактирование пройдено. |
| 2 | Нажата кнопка «редактировать»  Заполнены поля без «\*»  Не заполнены поля с «\*»  Нажата кнопку «подтвердить» | Добавление не пройдено. Сообщение об ошибке: «Пожалйства заполните поля». |
| 3 | Нажата кнопка «редактировать»  Не заполнены поля без «\*»  Заполнены поля с «\*»  Нажата кнопку «подтвердить» | Редактирование пройдено. |
| 4 | Нажата кнопка «редактировать»  Все поля заполнены  Нажата кнопку «подтвердить» | Редактирование пройдено. |

* 1. Проверка функции удаления данных из базы данных

Для успешного прохожденния данного тест кейса необходимо нажать на кнопку «Удалить»

Таблица – Проверка функции удаления данных из базы данных.

| № п/п | Действие (Входное значение) | Ожидаемый результат |
| --- | --- | --- |
| 1 | Нажата кнопка «Удалить» | Данные удалились из базы данных. |

* 1. Проверка функции «вывод информации об обьектах из базы данных»

Для успешного прохожденния данного тест кейса необходимо выбрать сущность на странице «Данные»

Таблица – Проверка функции «вывод информации об обьектах из базы данных»

| № п/п | Действие (Входное значение) | Ожидаемый результат |
| --- | --- | --- |
| 1 | Выбрана сущность на странице «Данные» | Вывод таблицы с данными сущностей из базы данных |
| 2 | Написан запрос в браузере “http://localhost:5000/users/get” | Вывод данных о докторов в формате json |

* 1. Проверка функции «Создание резервной копии базы данных»

Для успешного прохожденния данного тест кейса необходимо перейти на страницу «Резервное копирование» и нажать на кнопку «Сгрузить»

Таблица – Проверка функции «Создание резервной копии базы данных»

| № п/п | Действие (Входное значение) | Ожидаемый результат |
| --- | --- | --- |
| 1 | Нажата кнопка «Создать резервную копию» | Создается файл в директории «C/pg\_dumps/» с названием содержащем в себе текущее дату и время. В файле находится скрипт для востановления базы данных |

* 1. Проверка функции «Востановление базы данных»

Для успешного прохожденния данного тест кейса необходимо перейти на страницу «Резервное копирование». Выбрать файл и нажать на кнопку «Восстановить»

Таблица – Проверка функции «Востановление базы данных»

| № п/п | Действие (Входное значение) | Ожидаемый результат |
| --- | --- | --- |
| 1 | Нажата кнопка «Восстановить» | Воспроизводиться выбранный скрипт. База данных востанавливается. |

* 1. Проверка функции «Выгрузка данных в csv файлы»

Для успешного прохожденния данного тест кейса необходимо перейти на страницу «Резервное копирование» и нажать на кнопку «Выгрузить данные в csv файлы»

Таблица – проверка функции «Выгрузка данных в csv файлы»

| № п/п | Действие (Входное значение) | Ожидаемый результат |
| --- | --- | --- |
| 1 | Нажата кнопка «Выгрузить данные». Данные в базе данных отсутсвуют | Создаются csv файлы с заголовками строк таблицы. |
| 2 | Нажата кнопка «Выгрузить данные». Данные в базе данных присутствуют | Создаются csv файлы с заголовками строк таблицы и данными разделенными. |
| 3 | Нажата кнопка «Выгрузить». Данные в базе данных присутствуют. Директория «C/ pg\_dumps/csv» не сущетсвует. | Создается директория «C/ pg\_dumps/csv».  Создаются csv файлы с заголовками строк таблицы и данными разделенными. |

1. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Все тестовые данные для проведения тестирования приложения водителя представлены в таблицах 15 – 21

Тестирование производилось по тестовым данным, представленным в таблицах 8 – 15 в пункте «Тестовые примеры»

Таблица – Проверка создания обьектов в базе данных

| № п/п | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста |
| --- | --- | --- | --- | --- |
| 1 | Нажата кнопка «+»  Нажата кнопку «добавить» | Добавление не пройдено. Сообщение об ошибке: «Пожалуйста заполните поля». | Добавление не пройдено. Сообщение об ошибке: «Пожалуйста заполните поля». | Пройден |
| 2 | Нажата кнопка «+»  Заполнены поля без «\*»  Не заполнены поля с «\*»  Нажата кнопку «добавить» | Добавление не пройдено. Сообщение об ошибке: «Пожалйства заполните поля». | Добавление не пройдено. Сообщение об ошибке: «Пожалйства заполните поля». | Пройден |
| 3 | Нажата кнопка «+»  Не заполнены поля без «\*»  Заполнены поля с «\*»  Нажата кнопку «добавить» | Добавление пройдено. | Добавление пройдено. | Пройден |
| 4 | Нажата кнопка «+»  Все поля заполнены  Нажата кнопку «добавить» | Добавление пройдено. | Добавление пройдено. | Пройден |

* 1. Проверка редактирования обьектов в базе данных

Для успешного прохожденния данного тест кейса необходимо заполнить форму данными и нажать на кнопку «изменить».

Таблица – Проверка редактирования обьектов в базе данных

| № п/п | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста |
| --- | --- | --- | --- | --- |
| 1 | Нажата кнопка «редактировать»  Нажата кнопку «подтвердить» | Редактирование пройдено. | Редактирование пройдено. | Пройден |
| 2 | Нажата кнопка «редактировать»  Заполнены поля без «\*»  Не заполнены поля с «\*»  Нажата кнопку «подтвердить» | Добавление не пройдено. Сообщение об ошибке: «Пожалйства заполните поля». | Добавление не пройдено. Сообщение об ошибке: «Пожалйства заполните поля». | Пройден |
| 3 | Нажата кнопка «редактировать»  Не заполнены поля без «\*»  Заполнены поля с «\*»  Нажата кнопку «подтвердить» | Редактирование пройдено. | Редактирование пройдено. | Пройден |
| 4 | Нажата кнопка «редактировать»  Все поля заполнены  Нажата кнопку «подтвердить» | Редактирование пройдено. | Редактирование пройдено. | Пройден |

* 1. Проверка функции удаления данных из базы данных

Для успешного прохожденния данного тест кейса необходимо нажать на кнопку «Удалить»

Таблица – Проверка функции удаления данных из базы данных.

| № п/п | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста |
| --- | --- | --- | --- | --- |
| 1 | Нажата кнопка «Удалить» | Данные удалились из базы данных | Данные удалились из базы данных | Пройден |

* 1. Проверка функции «вывод информации об обьектах из базы данных»

Для успешного прохожденния данного тест кейса необходимо выбрать сущность на странице «Данные»

Таблица – Проверка функции «вывод информации об обьектах из базы данных»

| № п/п | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста |
| --- | --- | --- | --- | --- |
| 1 | Выбрана сущность на странице «Данные» | Вывод таблицы с данными сущностей из базы данных | Вывод таблицы с данными сущностей из базы данных | Пройден |
| 2 | Написан запрос в браузере “http://localhost:5000/api/doctors/” | Вывод данных о докторов в формате json | Вывод данных о докторов в формате json | Пройден |

* 1. Проверка функции «Создание резервной копии базы данных»

Для успешного прохожденния данного тест кейса необходимо перейти на страницу «Резервное копирование» и нажать на кнопку «Сгрузить»

Таблица – Проверка функции «Создание резервной копии базы данных»

| № п/п | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста |
| --- | --- | --- | --- | --- |
| 1 | Нажата кнопка «Создать резервную копию» | Создается файл в директории «C/ pg\_dumps /» с названием содержащем в себе текущее дату и время. В файле находится скрипт для востановления базы данных | Создается файл в директории «C/ pg\_dumps /» с названием содержащем в себе текущее дату и время. В файле находится скрипт для востановления базы данных | Пройден |

* 1. Проверка функции «Востановление базы данных»

Для успешного прохожденния данного тест кейса необходимо перейти на страницу «Резервное копирование». Выбрать файл и нажать на кнопку «Восстановить»

Таблица – Проверка функции «Востановление базы данных»

| № п/п | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста |
| --- | --- | --- | --- | --- |
| 1 | Нажата кнопка «Восстановить» | Воспроизводиться выбранный скрипт. База данных востанавливается. | Воспроизводиться выбранный скрипт. База данных востанавливается | Пройден |

* 1. Проверка функции «Выгрузка данных в csv файлы»

Для успешного прохожденния данного тест кейса необходимо перейти на страницу «Резервное копирование» и нажать на кнопку «Выгрузить данные в csv файлы»

Таблица – проверка функции «Выгрузка данных в csv файлы»

| № п/п | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста |
| --- | --- | --- | --- | --- |
| 1 | Нажата кнопка «Выгрузить данные». Данные в базе данных отсутсвуют | Создаются csv файлы с заголовками строк таблицы. | Создаются csv файлы с заголовками строк таблицы. | Пройден |
| 2 | Нажата кнопка «Выгрузить данные». Данные в базе данных присутствуют | Создаются csv файлы с заголовками строк таблицы и данными разделенными. | Создаются csv файлы с заголовками строк таблицы и данными разделенными. | Пройден |
| 3 | Нажата кнопка «Выгрузить данные». Данные в базе данных присутствуют. Директория «C/ dumps /csv» не сущетсвует. | Создается директория «C/ pg\_dumps /csv».  Создаются csv файлы с заголовками строк таблицы и данными разделенными. | Создается директория «C/ pg\_dumps /csv».  Создаются csv файлы с заголовками строк таблицы и данными разделенными. | Пройден |

# **ПРИЛОЖЕНИЕ В. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

АННОТАЦИЯ

В данном программном документе приведено руководство пользователя на использование приложения для контроля информационной системы багажного отделения аэропорта.

В разделе “Условия выполнения программы” описаны рекомендуемые и минимальные конфигурация ПК или ноутбука для использования данных программ. Также описано какие программные средства необходимы для пользования программ.

В разделе “Выполнение программы” находятся выполнение программ с описанием функций – описывает возможности программ и результат действий.

В разделе “Сообщение оператору” – описывается в виде таблицы какие сообщение будут выводится пользователя при тех или иных действиях

1. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

В таблице ниже представлены максимальные (или рекомендуемые) и минимальные технические средства для использования программы.

Таблица – Технические средства

|  |  |  |
| --- | --- | --- |
| № | Тип оборудования | Наименование оборудования |
| 1 | 2 | 3 |
| Максимальные или рекомендуемые | | |
| 1 | Размер экрана: | 15.6 " |
| 2 | Разрешение экрана: | 1920x1080 |
| 3 | Линейка процессора: | Intel Core i3 4170 |
| 4 | Количество ядер процессора: | 4 |
| 5 | Оперативная память: | 8 ГБ, DDR3 |
| 6 | Тип видеокарты: | Дискретная |
| 7 | Видеокарта: | nVidia GeForce GTX 1050 Ti |
| 8 | Конфигурация накопителей: | SSD 128 ГБ, HDD 512 ТБ |
| 9 | Операционная система: | Windows 10 Домашняя |
| Минимальные | | |
| 1 | Размер экрана: | 11 " |
| 2 | Разрешение экрана: | 1366 x 768 |
| 3 | Линейка процессора: | Intel Pentium M 753 1.2 ГГц |
| 4 | Количество ядер процессора: | 1 |
| 5 | Оперативная память: | 512 МБ, DDR2 SDRAM |
| 6 | Тип видеокарты: | Интегрированная |
| 7 | Видеокарта: | Intel Graphics Media Accelerator (GMA) 900 |
| 8 | Конфигурация накопителей: | HDD 60 ГБ |
| 9 | Общий объем всех накопителей: | 60 ГБ |
| 10 | Операционная система: | Windows 10 Lite х64 |

1. ВЫПОЛНЕНИЕ ПРОГРАММЫ

При старте приложение пользователь видит страницу с сущностями из базы данных

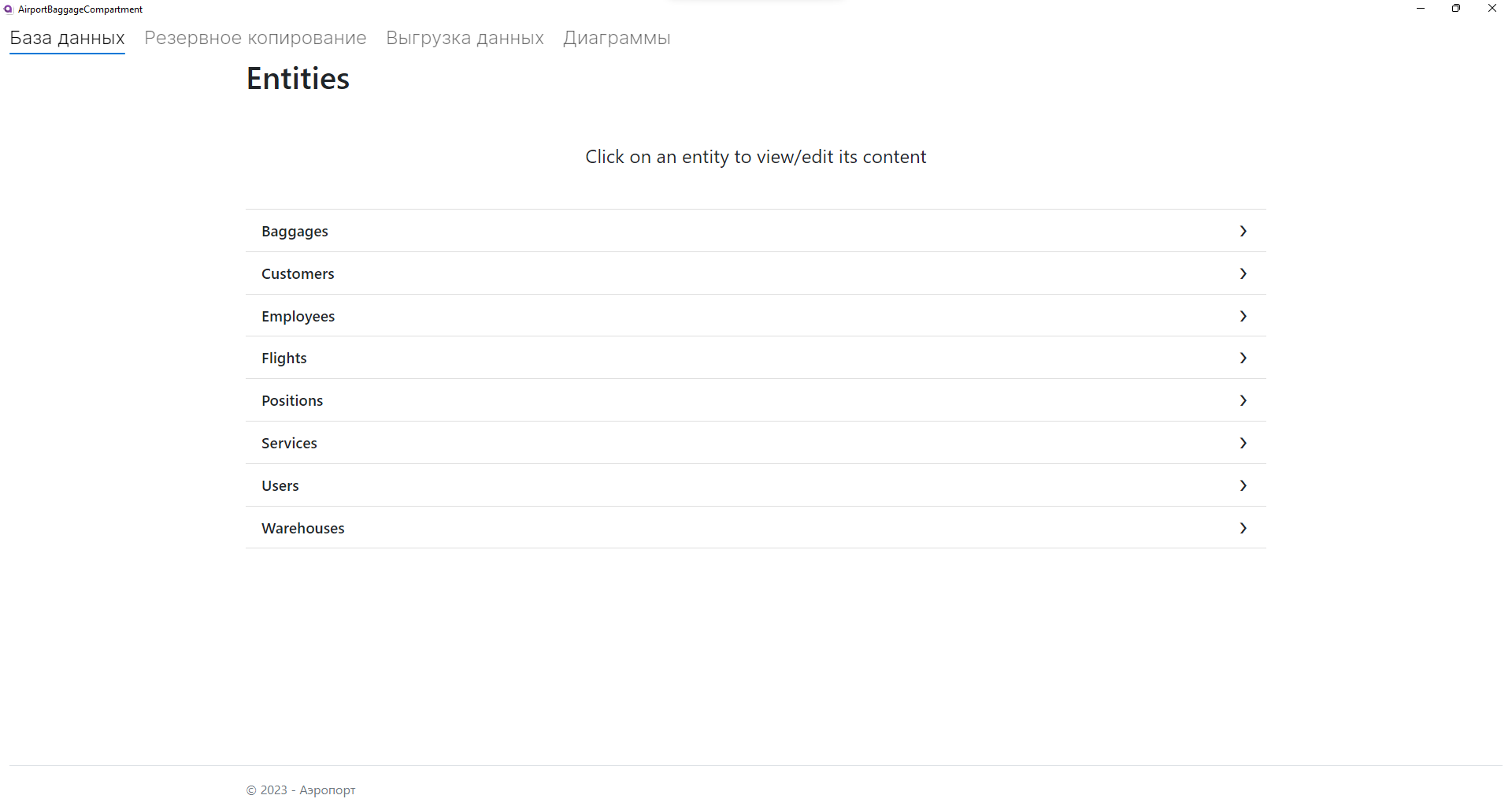


Рисунок 13 – страница с данными

Сверху распологается навигационная панель. С помощью нее пользователь может перейти на страницы «База данных», «Резервное копирование», «Выгрузка данных», «Диаграммы».

На странице «База данных» пользователь увидит список сущностей базы данных.

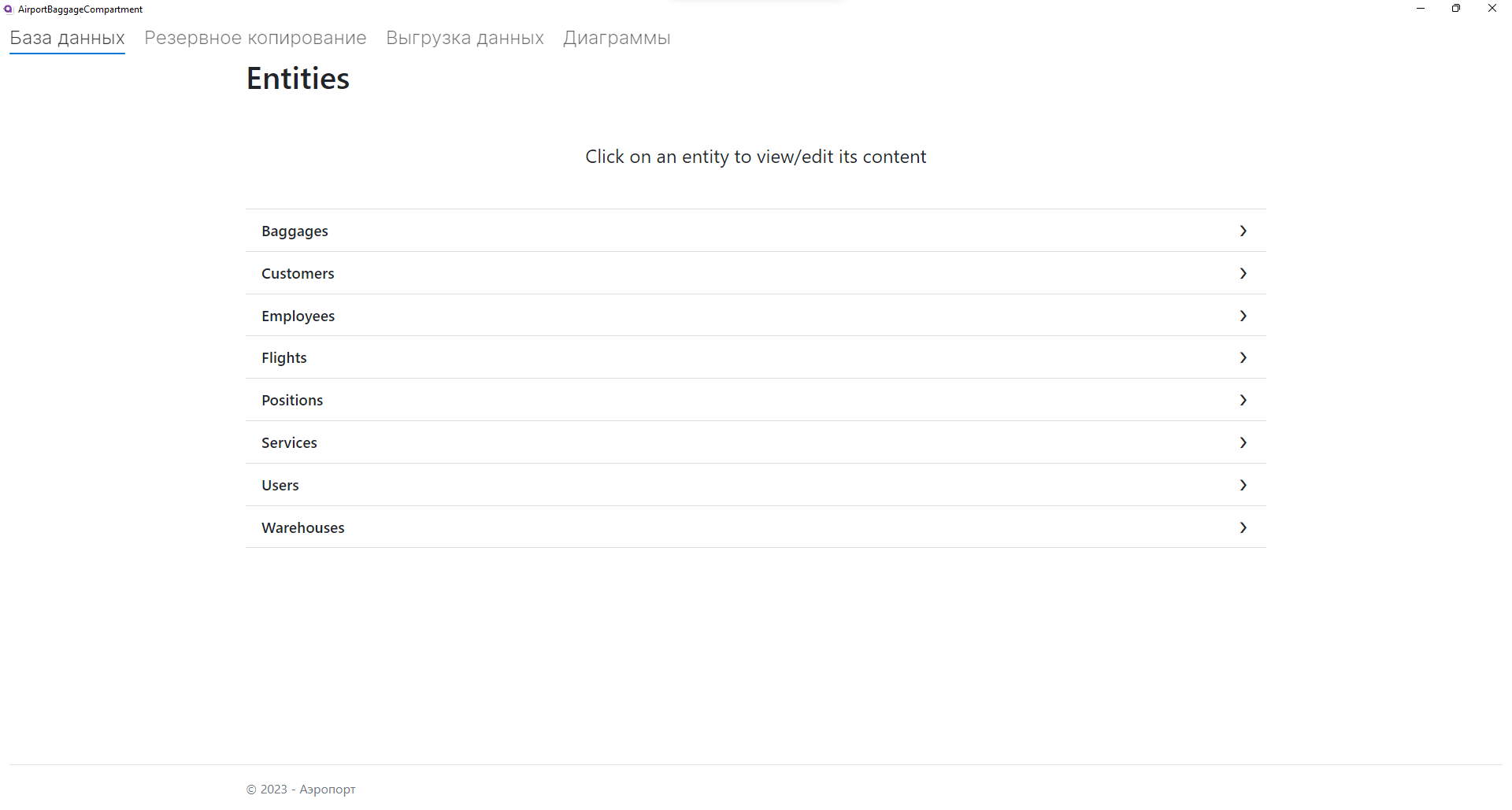


Рисунок 14 – страница «Данные»

Чтобы перейти на страницу редактирования сущности базы данных пользователю необходимо нажать на одну из них. Далее пользователю отобразится следующая страница.

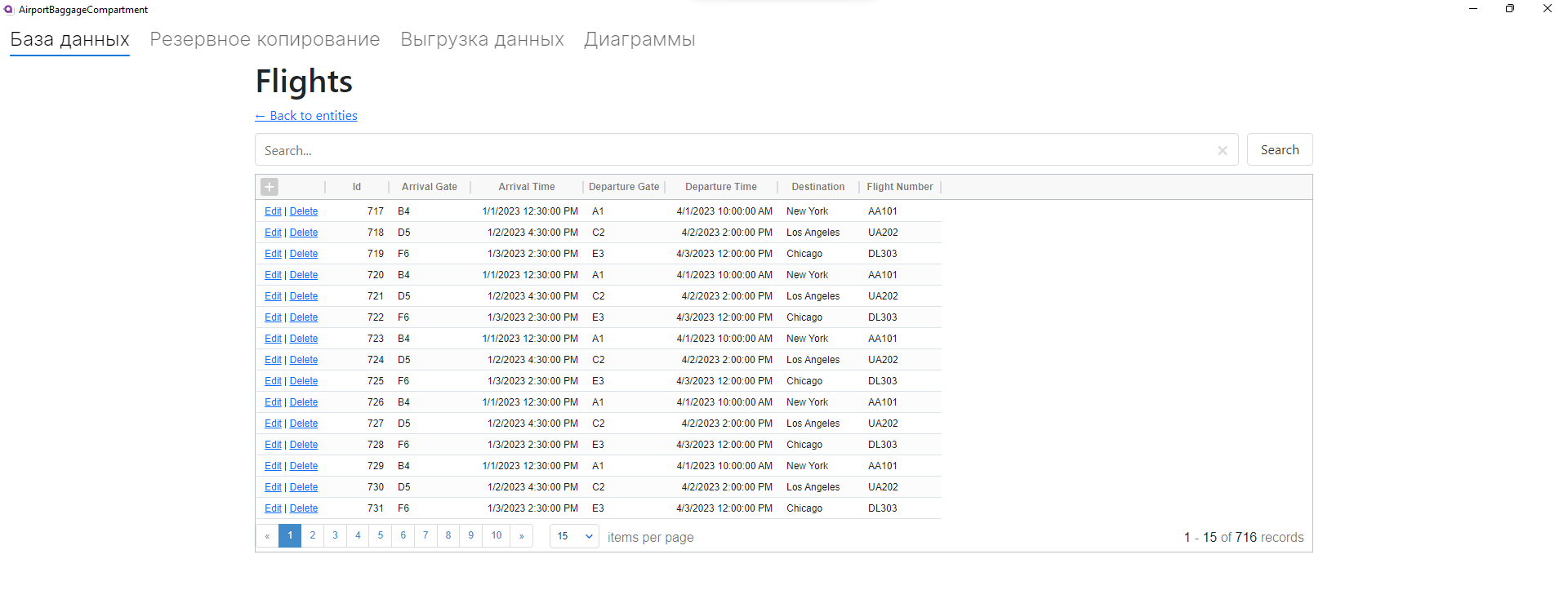


Рисунок 15 – страница редактирования сущности в базе данных

Страница добавления/редактрирования обьекта в базе данных.

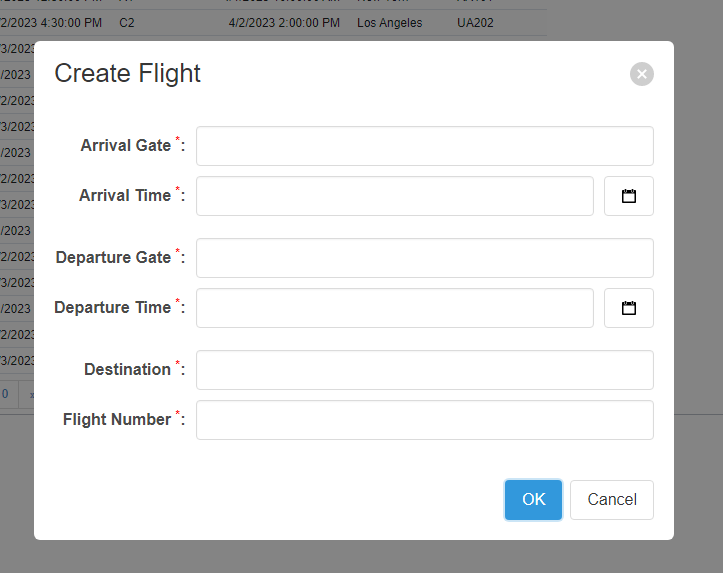


Рисунок 16 – форма добавления/редактирования

Если пользователь перейдет на страницу «Резервное копирование» то увидит две кнопки и список.

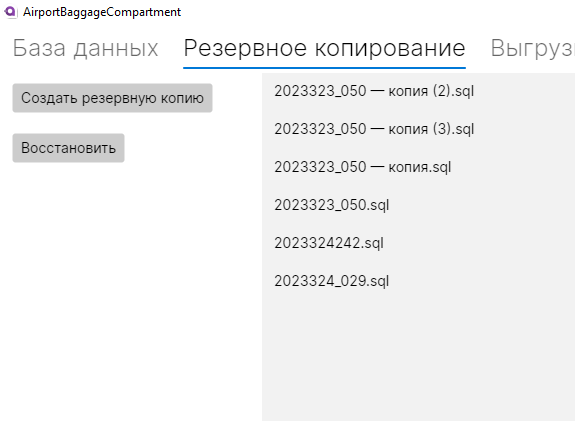
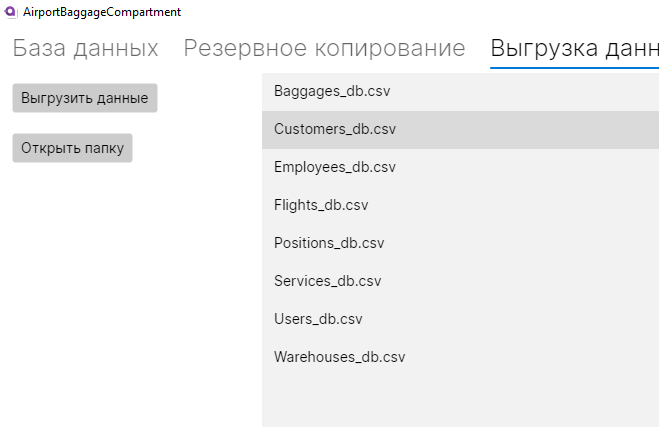


Рисунок 17 – страница «Резервное копирование»

Чтобы создать резервную копию базы данных пользователю необходимо нажать на кнопку «Создать резервную копию».

Для восстановления данных пользователю необходимо нажать на кнопку «Восстановить».

Если пользователь перейдет на страницу «Выгрузка данных» то увидит две кнопки и список.



Для выгрузки данных в cvs файлы пользователю необходимо нажать на кнопку «Выгрузить данные».

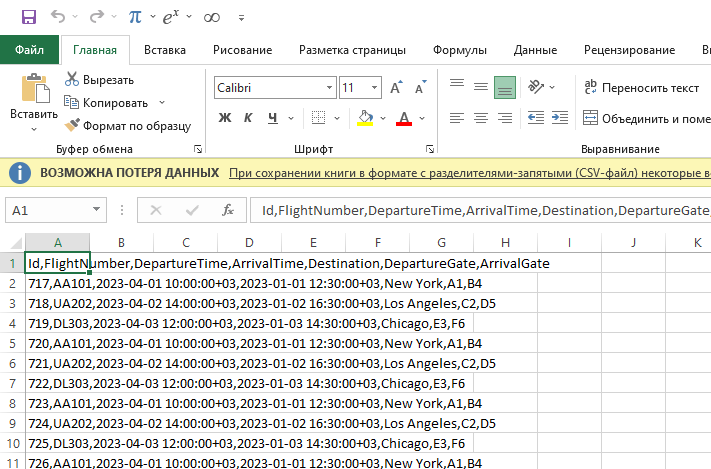


Рисунок 18 – результат выгрузки данных в csv файл

Если пользователь перейдет на страницу «Диаграммы», то увидит граффики количества полетов в города и число полетов за год.

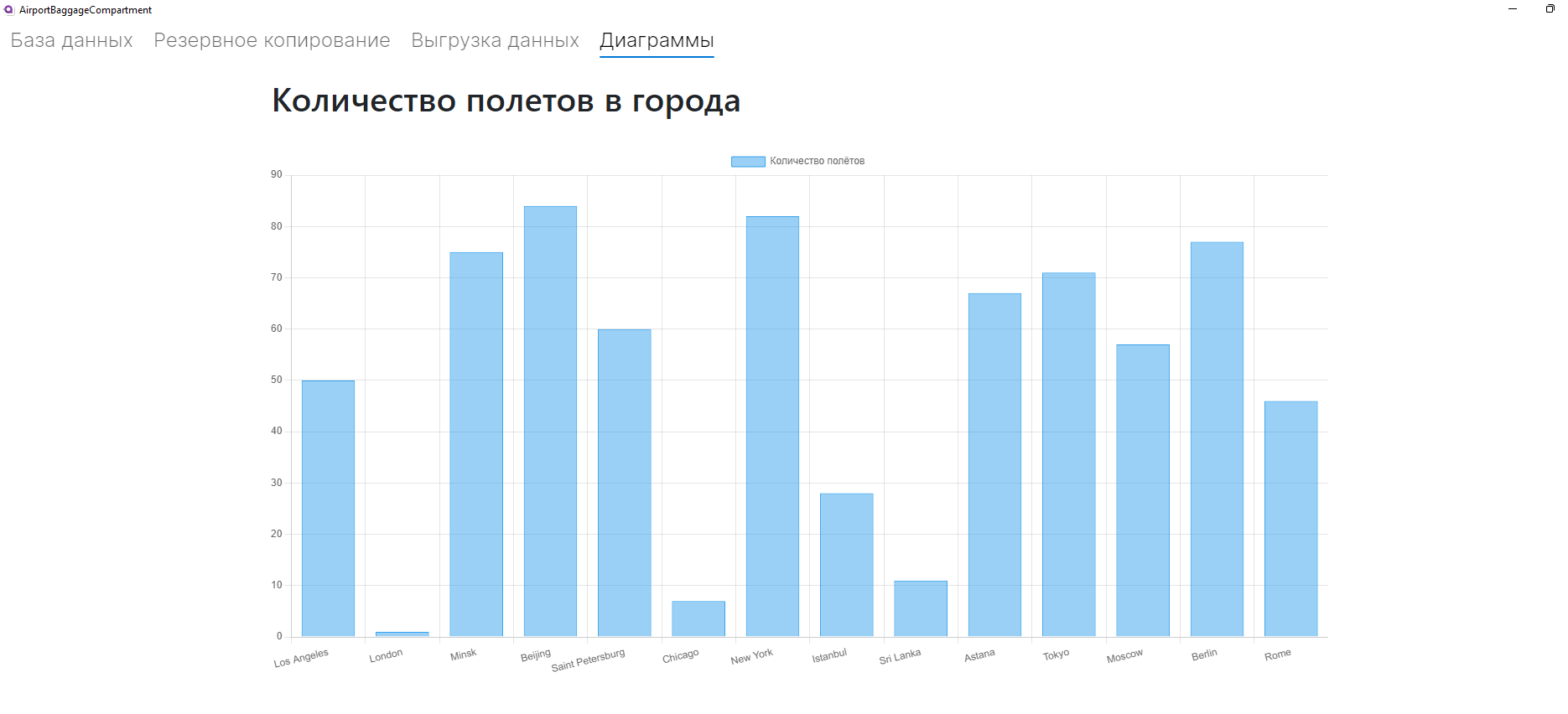


Рисунок – граффик (1)

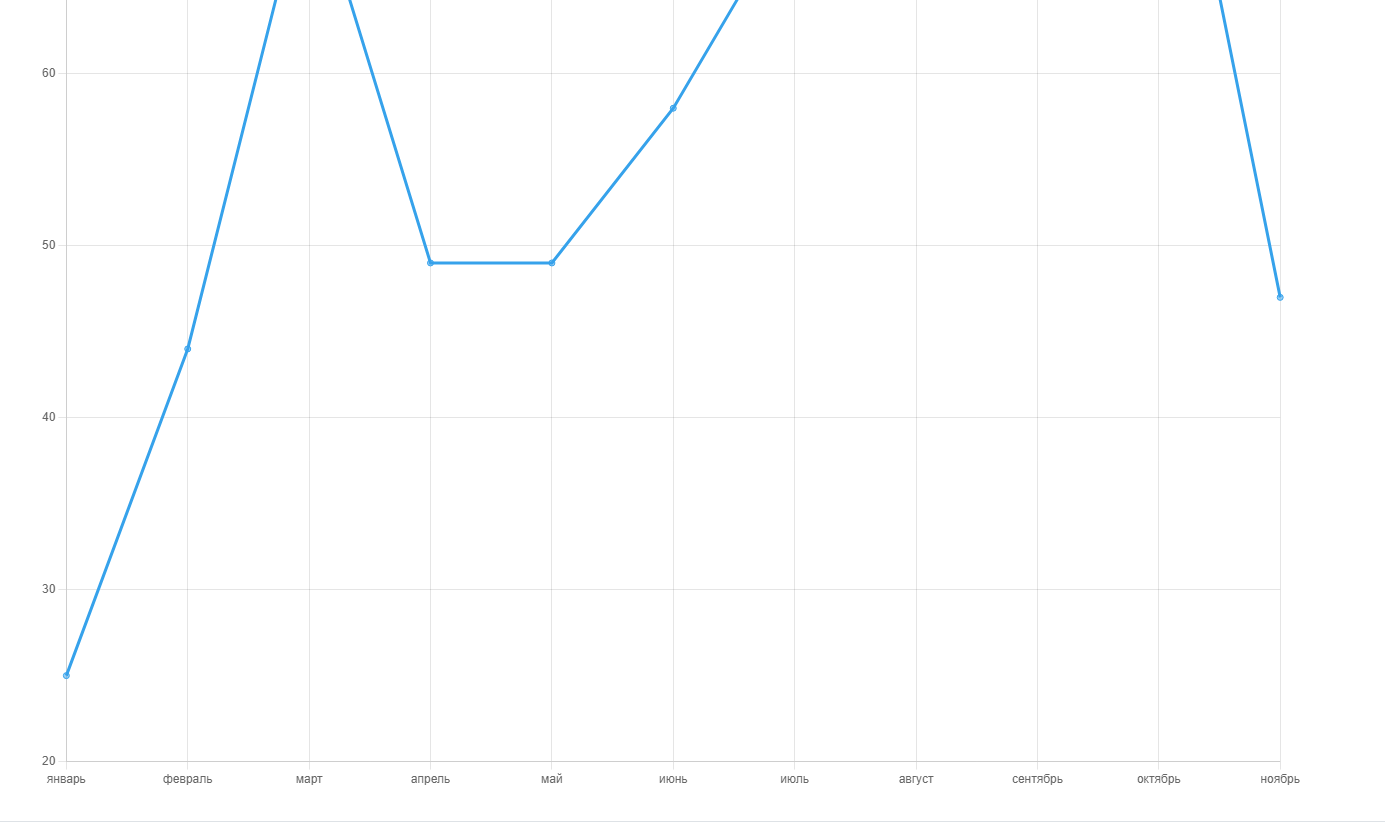


Рисунок – граффик (2)

# **ПРИЛОЖЕНИЕ Г. СКРИПТ БАЗЫ ДАННЫХ**

АННОТАЦИЯ

В данном программном документе приведён базы данных для информационной системы багажного отделения аэропорта.

В данном программном документе, в разделе «Скрипт базы данных» приведён скрипт базы данных программного изделия.

1. СКРИПТ БАЗЫ ДАННЫХ

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

CREATE PROCEDURE public.backupcsv()

LANGUAGE plpgsql

AS $$

DECLARE

mviews RECORD;

myName text;

BEGIN

FOR mviews IN SELECT table\_name FROM information\_schema.tables

WHERE table\_type = 'BASE TABLE' and table\_schema = 'public' LOOP

EXECUTE format(E'COPY %I TO \'C:\\tmp\\csv\\%s\_db.csv\' (FORMAT CSV, HEADER TRUE, DELIMITER \',\', ENCODING \'WIN1251\')', mviews.table\_name, mviews.table\_name);

END LOOP;

end;

$$;

ALTER PROCEDURE public.backupcsv() OWNER TO postgres;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

CREATE TABLE public."Baggages" (

"Id" integer NOT NULL,

"Weight" real NOT NULL,

"Size" text NOT NULL,

"FlightId" integer NOT NULL,

"CustomerId" integer NOT NULL,

"ServiceId" integer NOT NULL,

"WarehouseId" integer NOT NULL,

"EmployeeId" integer NOT NULL

);

ALTER TABLE public."Baggages" OWNER TO postgres;

ALTER TABLE public."Baggages" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Baggages\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Customers" (

"Id" integer NOT NULL,

"FirstName" text NOT NULL,

"LastName" text NOT NULL,

"Phone" text NOT NULL,

"Email" text NOT NULL

);

ALTER TABLE public."Customers" OWNER TO postgres;

ALTER TABLE public."Customers" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Customers\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Employees" (

"Id" integer NOT NULL,

"Name" text NOT NULL,

"Phone" text NOT NULL,

"Email" text NOT NULL,

"WarehouseId" integer NOT NULL,

"PositionId" integer NOT NULL

);

ALTER TABLE public."Employees" OWNER TO postgres;

ALTER TABLE public."Employees" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Employees\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Flights" (

"Id" integer NOT NULL,

"FlightNumber" text NOT NULL,

"DepartureTime" timestamp with time zone NOT NULL,

"ArrivalTime" timestamp with time zone NOT NULL,

"Destination" text NOT NULL,

"DepartureGate" text NOT NULL,

"ArrivalGate" text NOT NULL

);

ALTER TABLE public."Flights" OWNER TO postgres;

ALTER TABLE public."Flights" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Flights\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Positions" (

"Id" integer NOT NULL,

"Name" text NOT NULL,

"Salary" numeric NOT NULL

);

ALTER TABLE public."Positions" OWNER TO postgres;

ALTER TABLE public."Positions" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Positions\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Services" (

"Id" integer NOT NULL,

"Name" text NOT NULL,

"Description" text NOT NULL

);

ALTER TABLE public."Services" OWNER TO postgres;

ALTER TABLE public."Services" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Services\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Users" (

"Id" integer NOT NULL,

"Login" text NOT NULL,

"Password" text NOT NULL,

"Email" text NOT NULL,

"Role" integer NOT NULL

);

ALTER TABLE public."Users" OWNER TO postgres;

ALTER TABLE public."Users" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Users\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Warehouses" (

"Id" integer NOT NULL,

"Location" text NOT NULL,

"Phone" text NOT NULL,

"Email" text NOT NULL

);

ALTER TABLE public."Warehouses" OWNER TO postgres;

ALTER TABLE public."Warehouses" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Warehouses\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

COPY public."Baggages" ("Id", "Weight", "Size", "FlightId", "CustomerId", "ServiceId", "WarehouseId", "EmployeeId") FROM stdin;

\.

COPY public."Customers" ("Id", "FirstName", "LastName", "Phone", "Email") FROM stdin;

1 John Doe 123-456-7890 johndoe@example.com

2 Jane Smith 555-555-1212 janesmith@example.com

3 Bob Johnson 555-123-4567 bob@example.com

4 Samantha Williams 555-987-6543 samantha@example.com

\.

COPY public."Employees" ("Id", "Name", "Phone", "Email", "WarehouseId", "PositionId") FROM stdin;

1 John Doe 123-456-7890 johndoe@example.com 1 1

2 Jane Smith 555-555-1212 janesmith@example.com 1 2

3 Bob Johnson 555-123-4567 bob@example.com 3 3

4 Samantha Williams 555-987-6543 samantha@example.com 4 4

\.

COPY public."Flights" ("Id", "FlightNumber", "DepartureTime", "ArrivalTime", "Destination", "DepartureGate", "ArrivalGate") FROM stdin;

717 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

718 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

719 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

720 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

721 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

722 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

723 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

724 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

725 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

726 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

727 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

728 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

729 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

730 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

731 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

732 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

733 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

734 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

735 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

736 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

737 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

738 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

739 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

740 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 New York E3 F6

741 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

742 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

743 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

744 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

745 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

746 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

747 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

748 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

749 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

750 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

751 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

752 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

753 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

754 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

755 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

756 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

757 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

758 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

759 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

760 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

761 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

762 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

763 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

764 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

765 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

766 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

767 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

768 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

769 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

770 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

771 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

772 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

773 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

774 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

775 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

776 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

777 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

778 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

779 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

780 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

781 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

782 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

783 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

784 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

785 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

786 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

787 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

788 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

789 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

790 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

791 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

792 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

793 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

794 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

795 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

796 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

797 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

798 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

799 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

800 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

801 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

802 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

803 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

804 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

805 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

806 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

807 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

808 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

809 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

810 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

811 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

812 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

813 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

814 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

815 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

816 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

817 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

818 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

819 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

820 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

821 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

822 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

823 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

824 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

825 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

826 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

827 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

828 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

829 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

830 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

831 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

832 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

833 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

834 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

835 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

836 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

837 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

838 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

839 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

840 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

841 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

842 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

843 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

844 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

845 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

846 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

847 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

848 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

849 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

850 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

851 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

852 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

853 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

854 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

855 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

856 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Moscow E3 F6

857 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

858 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

859 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

860 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

861 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

862 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

863 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

864 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

865 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

866 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

867 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

868 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

869 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

870 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

871 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

872 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

873 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

874 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

875 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

876 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

877 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

878 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

879 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

880 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

881 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

882 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

883 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

884 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

885 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

886 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

887 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

888 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

889 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

890 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

891 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

892 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

893 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

894 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

895 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

896 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

897 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

898 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

899 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

900 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

901 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

902 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

903 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

904 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

905 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

906 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

907 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

908 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

909 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

910 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

911 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

912 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

913 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

914 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

915 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

916 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

917 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

918 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

919 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

920 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

921 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

922 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

923 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

924 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

925 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

926 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

927 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

928 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

929 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

930 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

931 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

932 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

933 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

934 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

935 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

936 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

937 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

938 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

939 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

940 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

941 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

942 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

943 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

944 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

945 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

946 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

947 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

948 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

949 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

950 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

951 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

952 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

953 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

954 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

955 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

956 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

957 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

958 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

959 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

960 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

961 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

962 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

963 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

964 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

965 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

966 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

967 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

968 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

969 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

970 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

971 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

972 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

973 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

974 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

975 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

976 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

977 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

978 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

979 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

980 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

981 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

982 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

983 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

984 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

985 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

986 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

987 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

988 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

989 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

990 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

991 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

992 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

993 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

994 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

995 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

996 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

997 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

998 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

999 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1000 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1001 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1002 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1003 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1004 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1005 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1006 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1007 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1008 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1009 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1010 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1011 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1012 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1013 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1014 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1015 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1016 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1017 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1018 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1019 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1020 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1021 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1022 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1023 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1024 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1025 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1026 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1027 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1028 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1029 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1030 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1031 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1032 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1033 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1034 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1035 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1036 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1037 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1038 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1039 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1040 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1041 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1042 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1043 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1044 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1045 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1046 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1047 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1048 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1049 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1050 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1051 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1052 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1053 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1054 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1055 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1056 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1057 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1058 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1059 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1060 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1061 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1062 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1063 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1064 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1065 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1066 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1067 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1068 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1069 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1070 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1071 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1072 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1073 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1074 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1075 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1076 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1077 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1078 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1079 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1080 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1081 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1082 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1083 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1084 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1085 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1086 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1087 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1088 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1089 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1090 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1091 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1092 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1093 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1094 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1095 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1096 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1097 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1098 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1099 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1100 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1101 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1102 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1103 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1104 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1105 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1106 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1107 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1108 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1109 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1110 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1111 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1112 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1113 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1114 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1115 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1116 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1117 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1118 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1119 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1120 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1121 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1122 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1123 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1124 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1125 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1126 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1127 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1128 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1129 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1130 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1131 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1132 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1133 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1134 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1135 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1136 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1137 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1138 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1139 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1140 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1141 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1142 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1143 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1144 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1145 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1146 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1147 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1148 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1149 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1150 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1151 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1152 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1153 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1154 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1155 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1156 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1157 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1158 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1159 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1160 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1161 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1162 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1163 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1164 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1165 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1166 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1167 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1168 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1169 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1170 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1171 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1172 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1173 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1174 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1175 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1176 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1177 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1178 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1179 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1180 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1181 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1182 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1183 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1184 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1185 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1186 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1187 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1188 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1189 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1190 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1191 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1192 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1193 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1194 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1195 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1196 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1197 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1198 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1199 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1200 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1201 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1202 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1203 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1204 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1205 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1206 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1207 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1208 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1209 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1210 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1211 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1212 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1213 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1214 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1215 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1216 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1217 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1218 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1219 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1220 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1221 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1222 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1223 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1224 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1225 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1226 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1227 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1228 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1229 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1230 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1231 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1232 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1233 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1234 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1235 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1236 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1237 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1238 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1239 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1240 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1241 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1242 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1243 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1244 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1245 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1246 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1247 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1248 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1249 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1250 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1251 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1252 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1253 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1254 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1255 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1256 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1257 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1258 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1259 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1260 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1261 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1262 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1263 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1264 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1265 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1266 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1267 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1268 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1269 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1270 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1271 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1272 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1273 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1274 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1275 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1276 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1277 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1278 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1279 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1280 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1281 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1282 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1283 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1284 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1285 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1286 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1287 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1288 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1289 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1290 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1291 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1292 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1293 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1294 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1295 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1296 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1297 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1298 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1299 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1300 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1301 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1302 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1303 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1304 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1305 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1306 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1307 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1308 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1309 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1310 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1311 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1312 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1313 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1314 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1315 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1316 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1317 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1318 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1319 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1320 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1321 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1322 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1323 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1324 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1325 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1326 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1327 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1328 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1329 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1330 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1331 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1332 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1333 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1334 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1335 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1336 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1337 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1338 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1339 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1340 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1341 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1342 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1343 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1344 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1345 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1346 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1347 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1348 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1349 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1350 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1351 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1352 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1353 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1354 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1355 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1356 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1357 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1358 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1359 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1360 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1361 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1362 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1363 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1364 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1365 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1366 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1367 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1368 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1369 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1370 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1371 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1372 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1373 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1374 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1375 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1376 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1377 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1378 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1379 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1380 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1381 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1382 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1383 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1384 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1385 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1386 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1387 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1388 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1389 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1390 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1391 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1392 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1393 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1394 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1395 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1396 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1397 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1398 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1399 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1400 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1401 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1402 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1403 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1404 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1405 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1406 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1407 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1408 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1409 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1410 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1411 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1412 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1413 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1414 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1415 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1416 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1417 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1418 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1419 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1420 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1421 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1422 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1423 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1424 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1425 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1426 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1427 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1428 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1429 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1430 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1431 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1432 BA404 2023-04-04 16:00:00+03 2023-01-04 18:30:00+03 London G4 H7

\.

COPY public."Positions" ("Id", "Name", "Salary") FROM stdin;

1 Обработчики багажа 100000

2 Оператор сканера багажа 100000

3 Техник багажной системы 100000

4 Агент по выдаче багажа 100000

5 Менеджер службы багажа 100000

6 Специалист по отслеживанию багажа 100000

7 Водитель багажного транспорта 100000

\.

COPY public."Services" ("Id", "Name", "Description") FROM stdin;

1 Baggage Tracking Track the movement of passengers’ baggage throughout the airport

2 Baggage Sorting Sort and organize baggage based on the flight and destination

3 Baggage Loading Load baggage onto the appropriate aircraft

4 Baggage Unloading Unload baggage from the aircraft and transport it to the baggage claim area

\.

COPY public."Users" ("Id", "Login", "Password", "Email", "Role") FROM stdin;

\.

COPY public."Warehouses" ("Id", "Location", "Phone", "Email") FROM stdin;

1 New York 123-456-7890 nywarehouse@example.com

2 Los Angeles 555-555-1212 lawarehouse@example.com

3 Chicago 555-123-4567 chiwarehouse@example.com

4 Miami 555-987-6543 miwarehouse@example.com

\.

SELECT pg\_catalog.setval('public."Baggages\_Id\_seq"', 1, false);

SELECT pg\_catalog.setval('public."Customers\_Id\_seq"', 1, false);

SELECT pg\_catalog.setval('public."Employees\_Id\_seq"', 4, true);

SELECT pg\_catalog.setval('public."Flights\_Id\_seq"', 1432, true);

SELECT pg\_catalog.setval('public."Positions\_Id\_seq"', 7, true);

SELECT pg\_catalog.setval('public."Services\_Id\_seq"', 4, true);

SELECT pg\_catalog.setval('public."Users\_Id\_seq"', 1, false);

SELECT pg\_catalog.setval('public."Warehouses\_Id\_seq"', 4, true);

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "PK\_Baggages" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Customers"

ADD CONSTRAINT "PK\_Customers" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Employees"

ADD CONSTRAINT "PK\_Employees" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Flights"

ADD CONSTRAINT "PK\_Flights" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Positions"

ADD CONSTRAINT "PK\_Positions" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Services"

ADD CONSTRAINT "PK\_Services" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Users"

ADD CONSTRAINT "PK\_Users" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Warehouses"

ADD CONSTRAINT "PK\_Warehouses" PRIMARY KEY ("Id");

CREATE INDEX "IX\_Baggages\_CustomerId" ON public."Baggages" USING btree ("CustomerId");

CREATE INDEX "IX\_Baggages\_EmployeeId" ON public."Baggages" USING btree ("EmployeeId");

CREATE INDEX "IX\_Baggages\_FlightId" ON public."Baggages" USING btree ("FlightId");

CREATE INDEX "IX\_Baggages\_ServiceId" ON public."Baggages" USING btree ("ServiceId");

CREATE INDEX "IX\_Baggages\_WarehouseId" ON public."Baggages" USING btree ("WarehouseId");

CREATE INDEX "IX\_Employees\_PositionId" ON public."Employees" USING btree ("PositionId");

CREATE INDEX "IX\_Employees\_WarehouseId" ON public."Employees" USING btree ("WarehouseId");

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Customers\_CustomerId" FOREIGN KEY ("CustomerId") REFERENCES public."Customers"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Employees\_EmployeeId" FOREIGN KEY ("EmployeeId") REFERENCES public."Employees"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Flights\_FlightId" FOREIGN KEY ("FlightId") REFERENCES public."Flights"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Services\_ServiceId" FOREIGN KEY ("ServiceId") REFERENCES public."Services"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Warehouses\_WarehouseId" FOREIGN KEY ("WarehouseId") REFERENCES public."Warehouses"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Employees"

ADD CONSTRAINT "FK\_Employees\_Positions\_PositionId" FOREIGN KEY ("PositionId") REFERENCES public."Positions"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Employees"

ADD CONSTRAINT "FK\_Employees\_Warehouses\_WarehouseId" FOREIGN KEY ("WarehouseId") REFERENCES public."Warehouses"("Id") ON DELETE CASCADE;

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

CREATE PROCEDURE public.backupcsv()

LANGUAGE plpgsql

AS $$

DECLARE

mviews RECORD;

myName text;

BEGIN

FOR mviews IN SELECT table\_name FROM information\_schema.tables

WHERE table\_type = 'BASE TABLE' and table\_schema = 'public' LOOP

EXECUTE format(E'COPY %I TO \'C:\\tmp\\csv\\%s\_db.csv\' (FORMAT CSV, HEADER TRUE, DELIMITER \',\', ENCODING \'WIN1251\')', mviews.table\_name, mviews.table\_name);

END LOOP;

end;

$$;

ALTER PROCEDURE public.backupcsv() OWNER TO postgres;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

CREATE TABLE public."Baggages" (

"Id" integer NOT NULL,

"Weight" real NOT NULL,

"Size" text NOT NULL,

"FlightId" integer NOT NULL,

"CustomerId" integer NOT NULL,

"ServiceId" integer NOT NULL,

"WarehouseId" integer NOT NULL,

"EmployeeId" integer NOT NULL

);

ALTER TABLE public."Baggages" OWNER TO postgres;

ALTER TABLE public."Baggages" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Baggages\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Customers" (

"Id" integer NOT NULL,

"FirstName" text NOT NULL,

"LastName" text NOT NULL,

"Phone" text NOT NULL,

"Email" text NOT NULL

);

ALTER TABLE public."Customers" OWNER TO postgres;

ALTER TABLE public."Customers" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Customers\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Employees" (

"Id" integer NOT NULL,

"Name" text NOT NULL,

"Phone" text NOT NULL,

"Email" text NOT NULL,

"WarehouseId" integer NOT NULL,

"PositionId" integer NOT NULL

);

ALTER TABLE public."Employees" OWNER TO postgres;

ALTER TABLE public."Employees" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Employees\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Flights" (

"Id" integer NOT NULL,

"FlightNumber" text NOT NULL,

"DepartureTime" timestamp with time zone NOT NULL,

"ArrivalTime" timestamp with time zone NOT NULL,

"Destination" text NOT NULL,

"DepartureGate" text NOT NULL,

"ArrivalGate" text NOT NULL

);

ALTER TABLE public."Flights" OWNER TO postgres;

ALTER TABLE public."Flights" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Flights\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Positions" (

"Id" integer NOT NULL,

"Name" text NOT NULL,

"Salary" numeric NOT NULL

);

ALTER TABLE public."Positions" OWNER TO postgres;

ALTER TABLE public."Positions" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Positions\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Services" (

"Id" integer NOT NULL,

"Name" text NOT NULL,

"Description" text NOT NULL

);

ALTER TABLE public."Services" OWNER TO postgres;

ALTER TABLE public."Services" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Services\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Users" (

"Id" integer NOT NULL,

"Login" text NOT NULL,

"Password" text NOT NULL,

"Email" text NOT NULL,

"Role" integer NOT NULL

);

ALTER TABLE public."Users" OWNER TO postgres;

ALTER TABLE public."Users" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Users\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

CREATE TABLE public."Warehouses" (

"Id" integer NOT NULL,

"Location" text NOT NULL,

"Phone" text NOT NULL,

"Email" text NOT NULL

);

ALTER TABLE public."Warehouses" OWNER TO postgres;

ALTER TABLE public."Warehouses" ALTER COLUMN "Id" ADD GENERATED BY DEFAULT AS IDENTITY (

SEQUENCE NAME public."Warehouses\_Id\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

COPY public."Baggages" ("Id", "Weight", "Size", "FlightId", "CustomerId", "ServiceId", "WarehouseId", "EmployeeId") FROM stdin;

\.

COPY public."Customers" ("Id", "FirstName", "LastName", "Phone", "Email") FROM stdin;

1 John Doe 123-456-7890 johndoe@example.com

2 Jane Smith 555-555-1212 janesmith@example.com

3 Bob Johnson 555-123-4567 bob@example.com

4 Samantha Williams 555-987-6543 samantha@example.com

\.

COPY public."Employees" ("Id", "Name", "Phone", "Email", "WarehouseId", "PositionId") FROM stdin;

1 John Doe 123-456-7890 johndoe@example.com 1 1

2 Jane Smith 555-555-1212 janesmith@example.com 1 2

3 Bob Johnson 555-123-4567 bob@example.com 3 3

4 Samantha Williams 555-987-6543 samantha@example.com 4 4

\.

COPY public."Flights" ("Id", "FlightNumber", "DepartureTime", "ArrivalTime", "Destination", "DepartureGate", "ArrivalGate") FROM stdin;

717 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

718 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

719 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

720 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

721 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

722 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

723 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

724 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

725 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

726 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

727 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

728 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

729 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

730 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

731 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

732 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

733 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

734 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

735 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

736 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

737 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 Chicago E3 F6

738 AA101 2023-04-01 10:00:00+03 2023-01-01 12:30:00+03 New York A1 B4

739 UA202 2023-04-02 14:00:00+03 2023-01-02 16:30:00+03 Los Angeles C2 D5

740 DL303 2023-04-03 12:00:00+03 2023-01-03 14:30:00+03 New York E3 F6

741 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

742 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

743 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

744 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

745 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

746 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

747 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

748 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

749 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

750 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

751 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

752 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

753 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

754 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

755 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

756 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

757 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

758 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

759 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

760 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

761 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

762 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

763 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

764 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

765 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

766 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

767 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

768 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

769 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

770 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

771 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

772 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

773 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

774 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

775 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

776 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

777 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

778 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

779 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

780 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

781 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

782 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

783 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

784 DL303 2023-04-03 12:00:00+03 2023-02-03 14:30:00+03 New York E3 F6

785 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

786 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

787 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

788 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

789 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

790 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

791 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

792 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

793 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

794 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

795 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

796 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

797 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

798 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

799 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

800 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

801 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

802 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

803 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

804 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

805 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

806 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

807 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

808 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

809 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

810 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

811 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

812 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

813 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 New York E3 F6

814 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

815 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

816 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

817 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

818 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

819 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

820 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

821 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

822 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

823 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

824 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

825 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

826 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

827 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

828 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

829 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

830 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

831 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

832 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

833 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

834 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

835 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

836 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

837 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

838 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

839 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

840 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

841 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

842 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

843 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

844 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

845 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

846 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

847 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

848 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

849 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

850 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

851 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

852 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

853 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

854 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

855 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Los Angeles E3 F6

856 DL303 2023-04-03 12:00:00+03 2023-03-03 14:30:00+03 Moscow E3 F6

857 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

858 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

859 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

860 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

861 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

862 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

863 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

864 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

865 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

866 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

867 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

868 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

869 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

870 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

871 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

872 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

873 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

874 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

875 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

876 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

877 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

878 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

879 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

880 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

881 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

882 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

883 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

884 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

885 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

886 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

887 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

888 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

889 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

890 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

891 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

892 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

893 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

894 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

895 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

896 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

897 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

898 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

899 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

900 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

901 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

902 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

903 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

904 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

905 DL303 2023-04-03 12:00:00+03 2023-04-03 14:30:00+03 Moscow E3 F6

906 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

907 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

908 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

909 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

910 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

911 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

912 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Moscow E3 F6

913 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

914 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

915 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

916 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

917 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

918 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

919 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

920 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

921 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

922 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

923 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

924 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

925 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

926 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

927 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

928 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

929 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

930 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

931 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

932 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

933 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

934 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

935 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

936 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

937 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

938 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

939 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

940 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

941 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

942 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

943 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

944 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

945 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

946 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

947 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

948 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

949 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

950 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

951 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

952 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

953 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

954 DL303 2023-04-03 12:00:00+03 2023-05-03 14:30:00+03 Saint Petersburg E3 F6

955 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

956 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

957 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

958 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

959 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

960 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

961 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

962 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

963 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

964 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

965 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

966 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

967 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

968 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

969 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

970 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

971 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

972 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Saint Petersburg E3 F6

973 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

974 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

975 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

976 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

977 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

978 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

979 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

980 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

981 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

982 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

983 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

984 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

985 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

986 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

987 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

988 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

989 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

990 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

991 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

992 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

993 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

994 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

995 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

996 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

997 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

998 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

999 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1000 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1001 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1002 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1003 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1004 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1005 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1006 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1007 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1008 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1009 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1010 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1011 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1012 DL303 2023-04-03 12:00:00+03 2023-06-03 14:30:00+03 Rome E3 F6

1013 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1014 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1015 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1016 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1017 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1018 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Rome E3 F6

1019 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1020 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1021 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1022 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1023 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1024 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1025 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1026 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1027 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1028 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1029 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1030 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1031 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1032 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1033 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1034 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1035 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1036 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1037 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1038 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1039 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1040 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1041 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1042 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1043 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1044 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1045 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1046 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1047 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1048 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1049 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1050 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1051 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1052 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1053 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1054 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1055 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1056 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1057 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1058 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1059 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1060 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1061 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1062 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1063 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1064 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1065 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1066 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1067 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1068 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1069 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1070 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1071 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1072 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1073 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1074 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1075 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1076 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1077 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1078 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1079 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1080 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1081 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1082 DL303 2023-04-03 12:00:00+03 2023-07-03 14:30:00+03 Berlin E3 F6

1083 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1084 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1085 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1086 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1087 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1088 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1089 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1090 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1091 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1092 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1093 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1094 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1095 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Berlin E3 F6

1096 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1097 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1098 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1099 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1100 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1101 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1102 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1103 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1104 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1105 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1106 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1107 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1108 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1109 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1110 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1111 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1112 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1113 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1114 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1115 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1116 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1117 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1118 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1119 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1120 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1121 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1122 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1123 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1124 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1125 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1126 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1127 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1128 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1129 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1130 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1131 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1132 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1133 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1134 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1135 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1136 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1137 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1138 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1139 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1140 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1141 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1142 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1143 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1144 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1145 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1146 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1147 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1148 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1149 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1150 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1151 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1152 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1153 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1154 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1155 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1156 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1157 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1158 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1159 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1160 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1161 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1162 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Astana E3 F6

1163 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1164 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1165 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1166 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1167 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1168 DL303 2023-04-03 12:00:00+03 2023-08-03 14:30:00+03 Minsk E3 F6

1169 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1170 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1171 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1172 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1173 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1174 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1175 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1176 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1177 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1178 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1179 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1180 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1181 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1182 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1183 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1184 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1185 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1186 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1187 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1188 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1189 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1190 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1191 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1192 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1193 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1194 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1195 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1196 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1197 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1198 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1199 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1200 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1201 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1202 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1203 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1204 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1205 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1206 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1207 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1208 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1209 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1210 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1211 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1212 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1213 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1214 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1215 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1216 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1217 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1218 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1219 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1220 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1221 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1222 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1223 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1224 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1225 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1226 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1227 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1228 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1229 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1230 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1231 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1232 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1233 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1234 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1235 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1236 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1237 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Minsk E3 F6

1238 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1239 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1240 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1241 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1242 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1243 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1244 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1245 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1246 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1247 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1248 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1249 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1250 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1251 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1252 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1253 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1254 DL303 2023-04-03 12:00:00+03 2023-09-03 14:30:00+03 Beijing E3 F6

1255 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1256 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1257 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1258 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1259 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1260 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1261 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1262 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1263 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1264 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1265 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1266 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1267 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1268 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1269 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1270 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1271 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1272 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1273 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1274 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1275 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1276 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1277 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1278 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1279 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1280 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1281 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1282 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1283 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1284 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1285 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1286 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1287 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1288 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1289 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1290 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1291 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1292 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1293 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1294 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1295 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1296 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1297 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1298 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1299 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1300 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1301 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1302 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1303 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1304 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1305 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1306 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1307 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1308 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1309 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1310 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1311 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1312 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1313 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1314 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1315 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1316 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1317 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1318 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1319 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1320 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1321 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Beijing E3 F6

1322 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1323 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1324 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1325 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1326 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1327 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1328 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1329 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1330 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1331 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1332 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1333 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1334 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1335 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1336 DL303 2023-04-03 12:00:00+03 2023-10-03 14:30:00+03 Tokyo E3 F6

1337 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1338 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1339 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1340 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1341 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1342 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1343 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1344 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1345 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1346 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1347 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1348 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1349 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1350 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1351 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1352 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1353 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1354 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1355 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1356 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1357 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1358 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1359 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1360 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1361 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1362 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1363 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1364 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1365 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1366 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1367 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1368 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1369 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1370 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1371 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1372 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1373 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1374 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1375 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1376 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1377 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1378 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1379 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1380 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1381 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1382 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1383 DL303 2023-04-03 12:00:00+03 2023-11-03 14:30:00+03 Tokyo E3 F6

1384 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1385 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1386 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1387 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1388 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1389 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1390 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1391 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1392 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Tokyo E3 F6

1393 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1394 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1395 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1396 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1397 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1398 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1399 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1400 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1401 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1402 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1403 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Sri Lanka E3 F6

1404 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1405 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1406 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1407 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1408 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1409 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1410 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1411 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1412 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1413 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1414 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1415 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1416 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1417 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1418 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1419 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1420 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1421 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1422 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1423 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1424 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1425 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1426 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1427 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1428 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1429 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1430 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1431 DL303 2023-04-03 12:00:00+03 2023-12-03 14:30:00+03 Istanbul E3 F6

1432 BA404 2023-04-04 16:00:00+03 2023-01-04 18:30:00+03 London G4 H7

\.

COPY public."Positions" ("Id", "Name", "Salary") FROM stdin;

1 Обработчики багажа 100000

2 Оператор сканера багажа 100000

3 Техник багажной системы 100000

4 Агент по выдаче багажа 100000

5 Менеджер службы багажа 100000

6 Специалист по отслеживанию багажа 100000

7 Водитель багажного транспорта 100000

\.

COPY public."Services" ("Id", "Name", "Description") FROM stdin;

1 Baggage Tracking Track the movement of passengers’ baggage throughout the airport

2 Baggage Sorting Sort and organize baggage based on the flight and destination

3 Baggage Loading Load baggage onto the appropriate aircraft

4 Baggage Unloading Unload baggage from the aircraft and transport it to the baggage claim area

\.

COPY public."Users" ("Id", "Login", "Password", "Email", "Role") FROM stdin;

\.

COPY public."Warehouses" ("Id", "Location", "Phone", "Email") FROM stdin;

1 New York 123-456-7890 nywarehouse@example.com

2 Los Angeles 555-555-1212 lawarehouse@example.com

3 Chicago 555-123-4567 chiwarehouse@example.com

4 Miami 555-987-6543 miwarehouse@example.com

\.

SELECT pg\_catalog.setval('public."Baggages\_Id\_seq"', 1, false);

SELECT pg\_catalog.setval('public."Customers\_Id\_seq"', 1, false);

SELECT pg\_catalog.setval('public."Employees\_Id\_seq"', 4, true);

SELECT pg\_catalog.setval('public."Flights\_Id\_seq"', 1432, true);

SELECT pg\_catalog.setval('public."Positions\_Id\_seq"', 7, true);

SELECT pg\_catalog.setval('public."Services\_Id\_seq"', 4, true);

SELECT pg\_catalog.setval('public."Users\_Id\_seq"', 1, false);

SELECT pg\_catalog.setval('public."Warehouses\_Id\_seq"', 4, true);

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "PK\_Baggages" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Customers"

ADD CONSTRAINT "PK\_Customers" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Employees"

ADD CONSTRAINT "PK\_Employees" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Flights"

ADD CONSTRAINT "PK\_Flights" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Positions"

ADD CONSTRAINT "PK\_Positions" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Services"

ADD CONSTRAINT "PK\_Services" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Users"

ADD CONSTRAINT "PK\_Users" PRIMARY KEY ("Id");

ALTER TABLE ONLY public."Warehouses"

ADD CONSTRAINT "PK\_Warehouses" PRIMARY KEY ("Id");

CREATE INDEX "IX\_Baggages\_CustomerId" ON public."Baggages" USING btree ("CustomerId");

CREATE INDEX "IX\_Baggages\_EmployeeId" ON public."Baggages" USING btree ("EmployeeId");

CREATE INDEX "IX\_Baggages\_FlightId" ON public."Baggages" USING btree ("FlightId");

CREATE INDEX "IX\_Baggages\_ServiceId" ON public."Baggages" USING btree ("ServiceId");

CREATE INDEX "IX\_Baggages\_WarehouseId" ON public."Baggages" USING btree ("WarehouseId");

CREATE INDEX "IX\_Employees\_PositionId" ON public."Employees" USING btree ("PositionId");

CREATE INDEX "IX\_Employees\_WarehouseId" ON public."Employees" USING btree ("WarehouseId");

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Customers\_CustomerId" FOREIGN KEY ("CustomerId") REFERENCES public."Customers"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Employees\_EmployeeId" FOREIGN KEY ("EmployeeId") REFERENCES public."Employees"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Flights\_FlightId" FOREIGN KEY ("FlightId") REFERENCES public."Flights"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Services\_ServiceId" FOREIGN KEY ("ServiceId") REFERENCES public."Services"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Baggages"

ADD CONSTRAINT "FK\_Baggages\_Warehouses\_WarehouseId" FOREIGN KEY ("WarehouseId") REFERENCES public."Warehouses"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Employees"

ADD CONSTRAINT "FK\_Employees\_Positions\_PositionId" FOREIGN KEY ("PositionId") REFERENCES public."Positions"("Id") ON DELETE CASCADE;

ALTER TABLE ONLY public."Employees"

ADD CONSTRAINT "FK\_Employees\_Warehouses\_WarehouseId" FOREIGN KEY ("WarehouseId") REFERENCES public."Warehouses"("Id") ON DELETE CASCADE;