|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| федеральное государственное бюджетное образовательное учреждение высшего образования |
| «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ» |
| (КНИТУ-КАИ) |

Институт (факультет) / филиал / отделение СПО \_в\_\_ИКТЗИ, КИТ\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование института (факультета) филиала / отделения СПО)

**ОТЧЕТ**

**по** **прохождению** **\_\_\_\_\_\_\_\_\_учебной\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** **практики**

(наименование практики (вид практики))

Направление подготовки / специальность:

09.02.07 «Информационные системы и программирование»

(наименование направления подготовки / специальности)

Выполнил обучающийся группы \_4238\_

\_\_\_\_\_\_\_\_\_\_\_   \_\_\_\_\_\_\_\_\_\_\_\_

(подпись)            (расшифровка подписи)

Руководитель практики от университета /

отделения СПО в ИКТЗИ, КИТ

\_преподаватель   \_\_Милютина В.Ю.\_\_

(должность)                     (подпись)            (расшифровка подписи)

Отчет защищен с оценкой: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Казань 2024

**АННОТАЦИЯ**

Отчет по учебной практике содержит 94 страниц, 33 рисунка, 6 таблиц, 9 источников литературы.

В данной практической работе будет спроектирована база данных для персонала компании по учету поставок товаров.

Для создания базы данных была использована стандартная среда администрирования баз данных выбранной СУБД MS SQL Server – Среда администрирования MS SQL Server Management.

База данных «PRAKTIKA» включает таблицы, представления, триггеры и хранимые процедуры.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 5](#_Toc170201555)

[ГЛАВА 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 8](#_Toc170201556)

[ГЛАВА 2 РАБОТА С БАЗОЙ ДАННЫХ 15](#_Toc170201557)

[ГЛАВА 3 СОЗДАНИЕ Android ПРИЛОЖЕНИЯ И ПОДКЛЮЧЕНИЕ БАЗЫ ДАННЫХ 33](#_Toc170201558)

[ЗАКЛЮЧЕНИЕ 38](#_Toc170201559)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 39](#_Toc170201560)

[ПРИЛОЖЕНИЕ А 40](#_Toc170201561)

[ПРИЛОЖЕНИЕ Б 65](#_Toc170201562)

|  |  |
| --- | --- |
| **Компетенции, формируемые в результате прохождения практики** | |
| ОК 01 | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |
| ОК 02 | Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности |
| ОК 03 | Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях |
| ОК 04 | Эффективно взаимодействовать и работать в коллективе и команде |
| ОК 05 | Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста |
| ОК 09 | Пользоваться профессиональной документацией на государственном и иностранном языках |
| ПК 11.1 | Осуществлять сбор, обработку и анализ информации для проектирования баз данных |
| ПК 11.2 | Проектировать базу данных на основе анализа предметной области |
| ПК 11.3 | Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области |
| ПК 11.4 | Реализовывать базу данных в конкретной системе управления базами данных |
| ПК 11.5 | Администрирование базы данных |
| ПК 11.6 | Защищать информацию в базе данных с использованием технологии защиты информации |

Дневник прохождения практики

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Срок выполнения этапов прохождения практики | Наименование этапов прохождения практики | Краткое содержание выполненных работ |
| 1 | 15.06.2024 | Проведение инструктажей, получение задания | Ознакомление с инструктажами по правилам проведения противопожарных тренировок, по безопасности трудам. Ознакомление и получения задания на выполнение практики. |
| 1 | 17.06.2024 - 18.06.2024 | Проектирование базы данных | Определение сущностей с их атрибутами и связей между ними. |
| 2 | 19.06.2024 - 20.06.2024 | Физическая реализация БД | Определяется каким образом представляются данные и детали необходимые для СУБД. |
| 3 | 21.06.2024 - 22.06.2024 | Импорт данных | Импортируются данные из excel файла в БД. |
| 4 | 23.06.2024 - 26.06.2024 | Поддержка целостности данных.  Реализация доступа пользователей к БД | Осуществляется поддержка целостности данных с помощью триггеров, представлений и хранимых процедур.  Создаются имена входа, учетные записи и роли БД |
| 5 | 27.06.2024 | Формирование и оформление отчета. | Оформление отчет по прохождению практики. Формирование необходимой документации. |
| 6 | 27.06.2024 | Сдача отчета о прохождение практики. | Пояснение работы созданный БД, сдача отчета о прохождение практики. |

# ВВЕДЕНИЕ

Курсовая работа посвящена разработке базы данных для системы аэропорта. Система аэропорта играет важную роль в обеспечении удобства и эффективности процесса приобретения авиабилетов пассажирами.

При разработке базы данных для системы необходимо учитывать особенности работы авиакомпаний, требования безопасности и конфиденциальности данных пассажиров, а также обеспечить возможность эффективного управления информацией о рейсах, билетах и пассажирах.

**Объектом исследования** является процесс разработки базы данных, а предметом курсовой работы является разработка БД «аэропорт».

**Цель курсовой работы** – создание БД «аэропорт».

Для достижения поставленной цели необходимо выполнить следующие **задачи**:

1. Ознакомиться с литературой по данной теме.
2. Проанализировать предметную область.
3. Построить ER-модель.
4. Создать объекты БД (таблицы, запросы).
5. Определить условия целостности БД, разработать триггеры и хранимые процедуры.
6. Разработать запросы к базе данных.
7. Создать WPF приложение с возможность редактирования таблиц базы данных

# Глава 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## **Теоретические сведения**

Система аэропорта представляет собой сложную организацию, объединяющую различные элементы и процессы для обеспечения безопасного и эффективного функционирования авиационной деятельности.

Важными теоретическими аспектами системы аэропорта являются:

* **Структура системы аэропорта**: аэропорт состоит из различных компонентов, включая инфраструктуру (терминалы, взлетно-посадочные полосы, стоянки), обслуживающие службы (управление рейсами, безопасность, обслуживание пассажиров), техническое обслуживание, управление данными и другие элементы.
* **Функции системы аэропорта**: основные функции аэропорта включают в себя обеспечение безопасности полетов, обслуживание пассажиров и грузов, управление воздушным движением, техническое обслуживание самолетов, обеспечение комфортного пребывания пассажиров и другие задачи.
* **Взаимодействие компонентов системы**: эффективное функционирование аэропорта зависит от взаимодействия различных компонентов системы, таких как службы безопасности, обслуживание пассажиров, управление рейсами и другие. Каждый компонент выполняет свою роль в рамках общей системы.
* **Управление системой аэропорта**: управление аэропортом требует координации различных процессов и ресурсов для достижения поставленных целей. Это включает в себя планирование рейсов, управление ресурсами, контроль качества обслуживания, обеспечение безопасности и другие аспекты управления.
* **Информационные технологии в системе аэропорта**: современные аэропорты активно используют информационные технологии для автоматизации процессов, улучшения обслуживания пассажиров, управления данными о рейсах и пассажирах, контроля безопасности и других задач.

## **Функциональный анализ выбранной темы**

1. Регистрация и аутентификация пользователей:

- Пользователи должны иметь возможность зарегистрироваться в системе для доступа к функционалу бронирования билетов.

- Система должна обеспечивать безопасную аутентификацию пользователей.

2. Поиск и выбор рейсов:

- Пользователи должны иметь возможность искать доступные рейсы по различным критериям (например, дата, направление, время вылета и прилета).

- Система должна предоставлять информацию о доступных рейсах, ценах и классах обслуживания.

3. Бронирование билетов:

- Пользователи должны иметь возможность забронировать билет на выбранный рейс.

- Система должна обеспечивать возможность выбора места в самолете, оплаты билета и получения подтверждения бронирования.

4. Управление бронированием:

- Пользователи должны иметь возможность просматривать и изменять информацию о своем бронировании (например, изменение даты вылета или времени).

- Система должна предоставлять возможность отмены бронирования и возврата средств.

5. Управление данными пассажиров:

- Система должна хранить информацию о пассажирах (например, ФИО, дата рождения, паспортные данные) для оформления билетов.

- Пользователи должны иметь возможность добавлять, изменять или удалять информацию о пассажирах.

6. Управление данными рейсов:

- Система должна хранить информацию о рейсах (например, номер рейса, города вылета и прилета, время вылета и прилета) для предоставления пользователю актуальной информации.

- Администраторы системы должны иметь возможность добавлять новые рейсы, изменять существующие и управлять расписанием.

## **1.3 Выбор программного обеспечения**

SQL предоставляет удобство и эффективность в работе с данными, масштабируемость и гибкость при выполнении запросов. Он обеспечивает безопасность данных, поддержку транзакций и высокую производительность при работе с базой данных. SQL также обладает легкостью в создании отчетов и аналитики, что делает его неотъемлемой частью разработки приложений.

C# является языком программирования высокого уровня с обширной библиотекой классов и функций. Он позволяет использовать LINQ для запросов к данным, что упрощает работу с коллекциями и базами данных. Кроме того, C# обладает возможностью разработки кроссплатформенных приложений с использованием Xamarin, поддержкой асинхронного программирования и параллельных вычислений, а также широкими возможностями по работе с событиями и делегатами.

В целом, сочетание SQL, WPF и C# позволяет создавать мощные, гибкие и безопасные приложения с красивым и интерактивным пользовательским интерфейсом, эффективно работать с данными и обеспечивать высокую производительность. Каждая из этих технологий вносит свой вклад в разработку приложений, обеспечивая разработчикам широкие возможности для реализации различных задач.

# Глава 2 РАБОТА С БАЗОЙ ДАННЫХ

## **2.1 Проектирование базы данных**

Первая диаграмма: Концептуальное проектирование базы данных.

Этапы построения:

1. Определение сущностей
2. Определение связей между сущностями
3. Создание ER-диаграммы

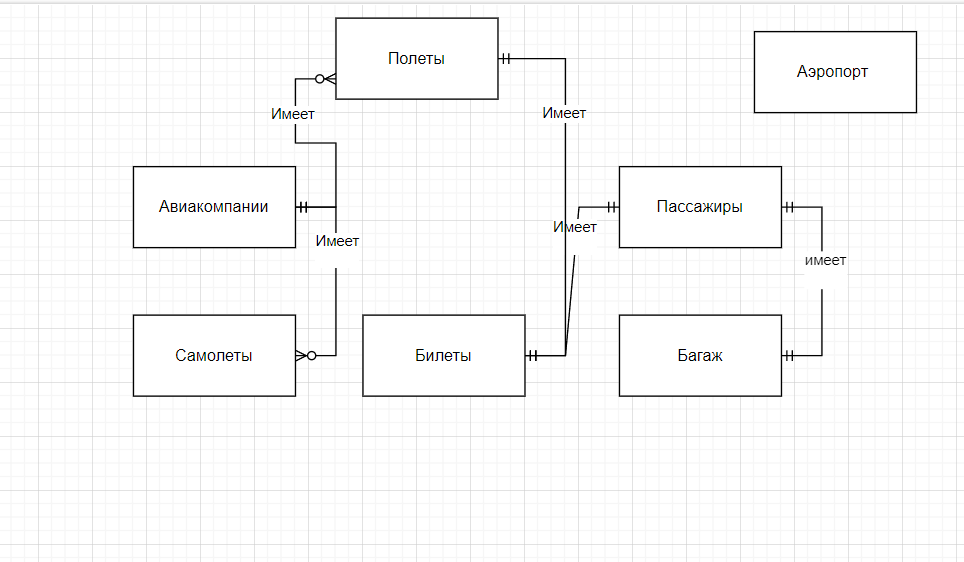


Рисунок 1 – Диаграмма 1

Вторая диаграмма: Логическое проектирование базы данных.

Этапы построения:

1. Выбор модели данных
2. Определение набора таблиц исходя из ER-диаграммы
3. Определение связей
4. Определение атрибутов
5. Приведение таблиц к 3НФ

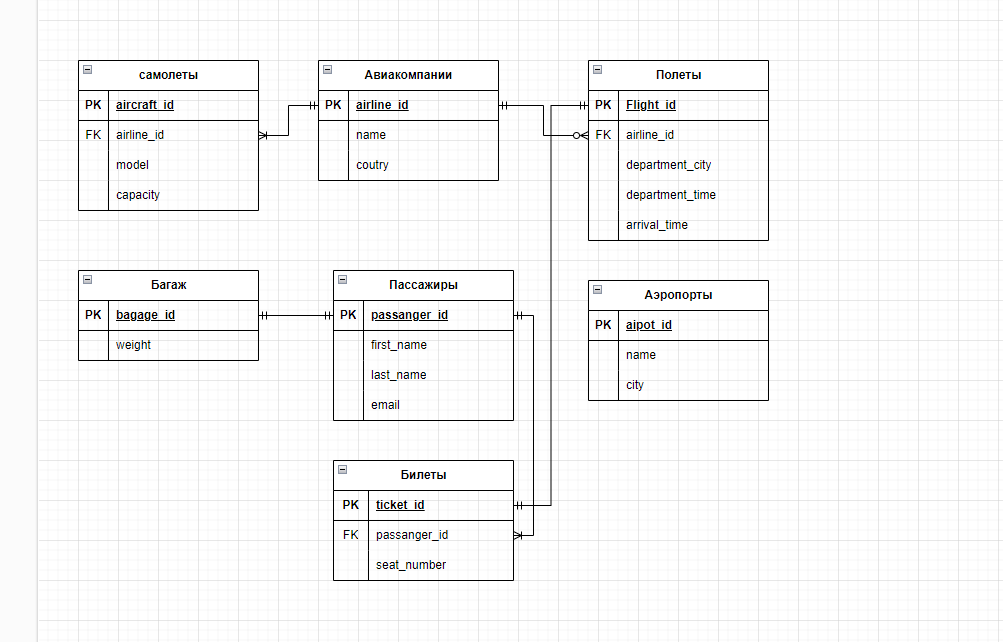


Рисунок 2 – Диаграмма 2

Третья диаграмма: Физическое проектирование базы данных.

Этапы построения:

1. Проектирование таблиц базы данных средствами выбранной СУБД
2. Определение набора таблиц исходя из ER-диаграммы
3. Определение связей
4. Определение атрибутов и их типов данных
5. Определение первичных и внешних ключей

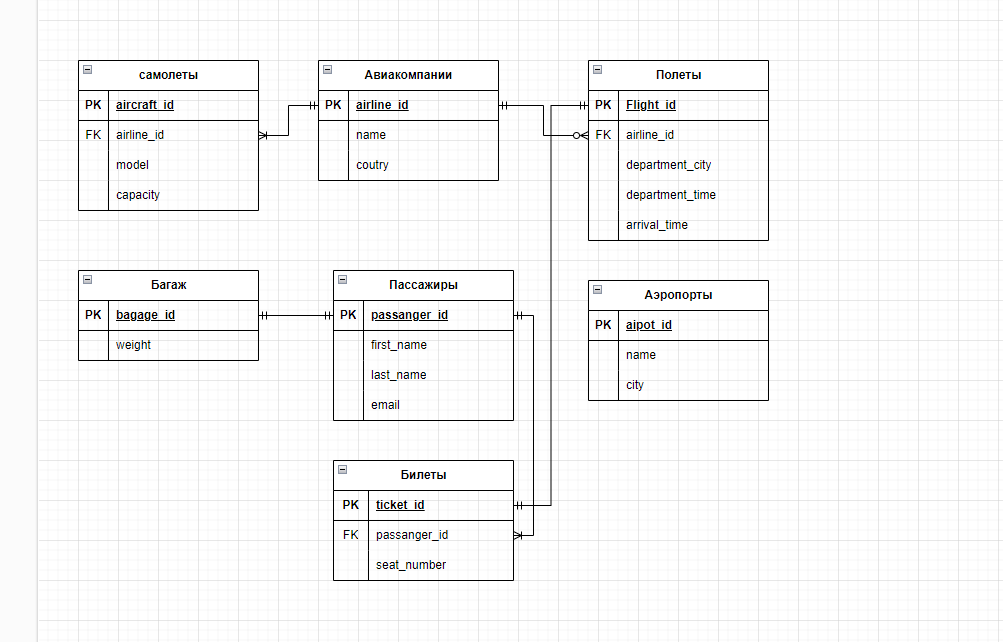


Рисунок 3 – Диаграмма 3

Листинг запросов представлен в таблице 1:

|  |
| --- |
| CREATE TABLE Airlines (  airline\_id INT PRIMARY KEY,  name VARCHAR(255),  country VARCHAR(255)  );  CREATE TABLE Flights (  flight\_id INT PRIMARY KEY,  airline\_id INT,  departure\_city VARCHAR(255),  arrival\_city VARCHAR(255),  departure\_time DATETIME,  arrival\_time DATETIME,  FOREIGN KEY (airline\_id) REFERENCES Airlines(airline\_id)  );  CREATE TABLE Aircraft (  aircraft\_id INT PRIMARY KEY,  airline\_id INT,  model VARCHAR(255),  capacity INT,  FOREIGN KEY (airline\_id) REFERENCES Airlines(airline\_id)  );  CREATE TABLE Passengers (  passenger\_id INT PRIMARY KEY,  first\_name VARCHAR(255),  last\_name VARCHAR(255),  email VARCHAR(255)  );  CREATE TABLE Tickets (  ticket\_id INT PRIMARY KEY,  passenger\_id INT,  flight\_id INT,  seat\_number VARCHAR(10), |

Продолжение таблицы 1:

|  |
| --- |
| FOREIGN KEY (passenger\_id) REFERENCES Passengers(passenger\_id),  FOREIGN KEY (flight\_id) REFERENCES Flights(flight\_id)  );  CREATE TABLE Airports (  airport\_id INT PRIMARY KEY,  name VARCHAR(255),  city VARCHAR(255)  );  CREATE TABLE Baggage (  baggage\_id INT PRIMARY KEY,  passenger\_id INT,  weight DECIMAL(10, 2),  FOREIGN KEY (passenger\_id) REFERENCES Passengers(passenger\_id)  ); |

## **2.2 Физическая реализация базы данных**

Описание сущностей и атрибутов нашей базы данных:

1. Таблица Airlines:

- airline\_id: целочисленное поле, являющееся первичным ключом таблицы.

- name: строковое поле с названием авиакомпании.

- country: строковое поле с названием страны, к которой принадлежит авиакомпания.

1. Таблица Flights:

- flight\_id: целочисленное поле, являющееся первичным ключом таблицы.

- airline\_id: внешний ключ, связывающий рейс с соответствующей авиакомпанией.

- departure\_city: строковое поле с названием города отправления.

- arrival\_city: строковое поле с названием города прибытия.

- departure\_time: поле типа DATETIME, указывающее время отправления.

- arrival\_time: поле типа DATETIME, указывающее время прибытия.

1. Таблица Aircraft:

- aircraft\_id: целочисленное поле, являющееся первичным ключом таблицы.

- airline\_id: внешний ключ, связывающий самолет с соответствующей авиакомпанией.

- model: строковое поле с моделью самолета.

- capacity: целочисленное поле, указывающее вместимость самолета.

1. Таблица Passengers:

- passenger\_id: целочисленное поле, являющееся первичным ключом таблицы.

- first\_name: строковое поле с именем пассажира.

- last\_name: строковое поле с фамилией пассажира.

- email: строковое поле с адресом электронной почты пассажира.

1. Таблица Tickets:

- ticket\_id: целочисленное поле, являющееся первичным ключом таблицы.

- passenger\_id: внешний ключ, связывающий билет с соответствующим пассажиром.

- flight\_id: внешний ключ, связывающий билет с соответствующим рейсом.

- seat\_number: строковое поле с номером места на рейсе.

1. Таблица Airports:

- airport\_id: целочисленное поле, являющееся первичным ключом таблицы.

- name: строковое поле с названием аэропорта.

- city: строковое поле с названием города, в котором расположен аэропорт.

1. Таблица Baggage:

- baggage\_id: целочисленное поле, являющееся первичным ключом таблицы.

- passenger\_id: внешний ключ, связывающий багаж с соответствующим пассажиром.

- weight: числовое поле типа DECIMAL(10, 2), указывающее вес багажа пассажира.

Таким образом, у нас есть база данных, которая позволяет хранить информацию о авиакомпаниях, рейсах, самолетах, пассажирах, билетах, аэропортах и багаже, а также устанавливать связи между этими сущностями.

Связи в нашей базе данных:

1. Airlines и Flights:

- Связь между таблицами Airlines и Flights устанавливается через внешний ключ airline\_id в таблице Flights. Этот ключ связывает каждый рейс с соответствующей авиакомпанией.

1. Airlines и Aircraft:

- Таблицы Airlines и Aircraft связаны через внешний ключ airline\_id в таблице Aircraft. Каждый самолет привязан к конкретной авиакомпании.

1. Flights и Aircraft:

- Связь между таблицами Flights и Aircraft также устанавливается через внешний ключ airline\_id в таблице Aircraft. Это позволяет определить, какой самолет используется для конкретного рейса.

1. Passengers и Tickets:

- Таблицы Passengers и Tickets связаны через внешний ключ passenger\_id в таблице Tickets. Каждый билет привязан к определенному пассажиру.

1. Tickets и Flights:

- Связь между таблицами Tickets и Flights устанавливается через внешний ключ flight\_id в таблице Tickets. Это позволяет определить, на какой рейс куплен конкретный билет.

1. Passengers и Baggage:

- Таблицы Passengers и Baggage связаны через внешний ключ passenger\_id в таблице Baggage. Каждый элемент багажа привязан к определенному пассажиру.

## **2.3 Импорт данных**

На рисунке 5 мы нажимаем ПКМ на нашу базу данных и переходим в «задачи» там мы выбираем «Импорт данных».

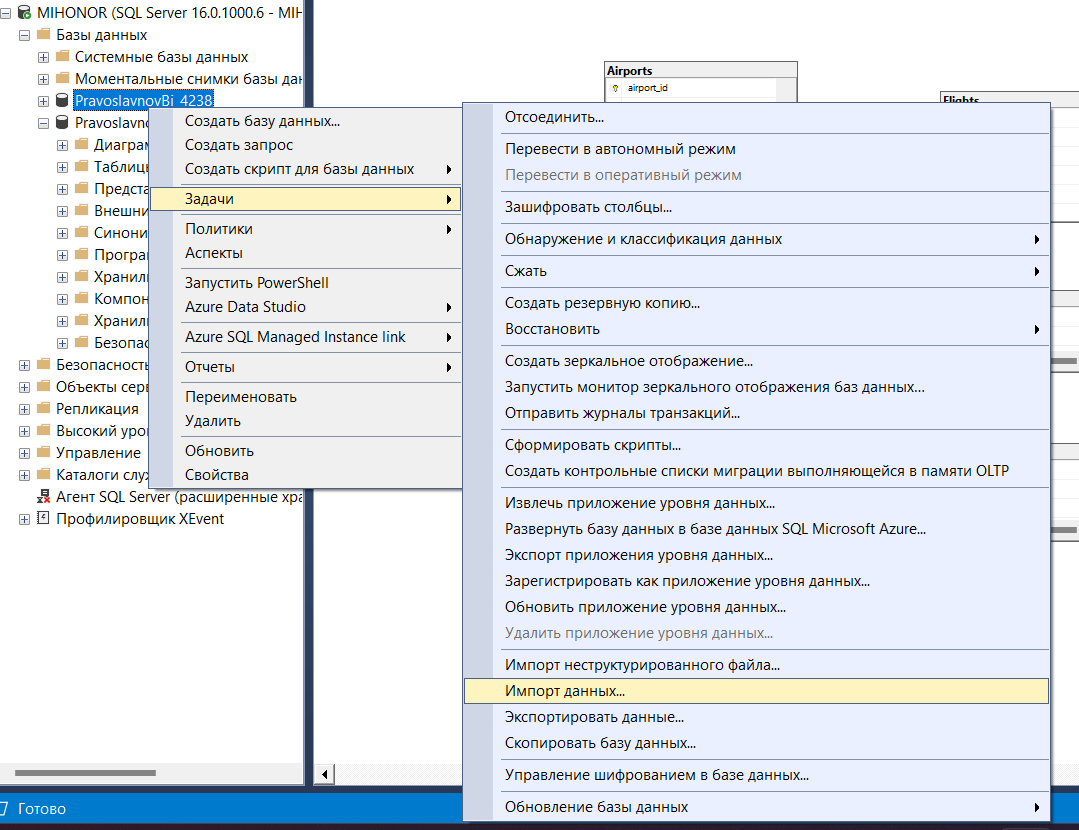


Рисунок 5 – Импорт данных

На рисунке 6 мы выбираем источник данных и путь к файлу в нашем случаи источник данных это «Microsoft Excel" и версия Excel: «Microsoft Excel 97-2003».

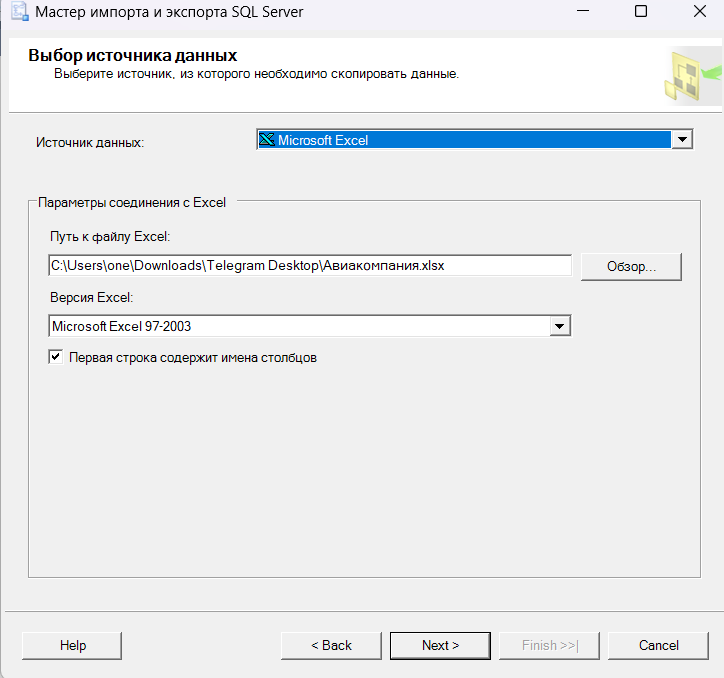


Рисунок 6 – Выбор источника данных

На рисунке 7 мы выбираем назначение и куда мы хотим скопировать данные в нашем случаи это назначение: «Microsoft OLE DB Provider for SQL Server» имя сервера: «MIHONOR» и база данных: «PRAKTIKA».

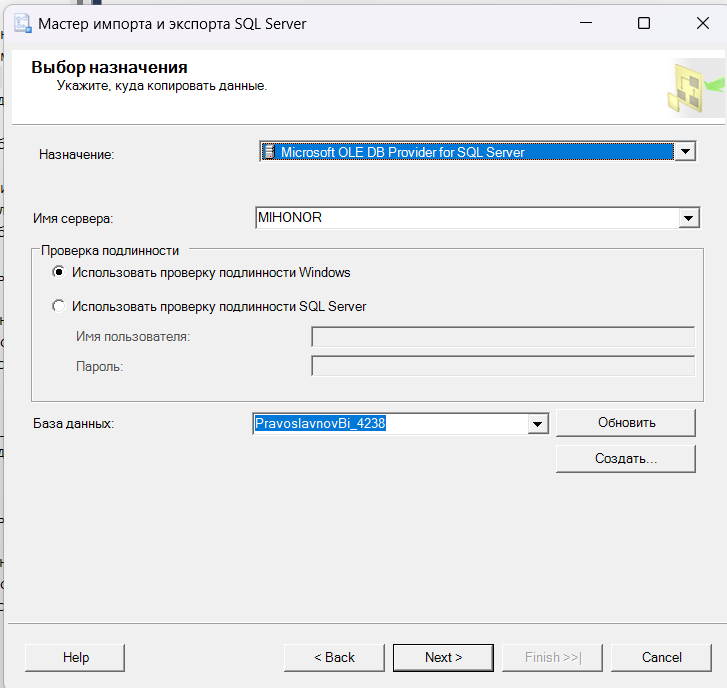


Рисунок 7 – Выбор назначения

На рисунке 8 мы выбираем таблицы, которые мы хотим скопировать, здесь мы нажимаем на «Скопировать данные из одной или нескольких таблиц или представлений».

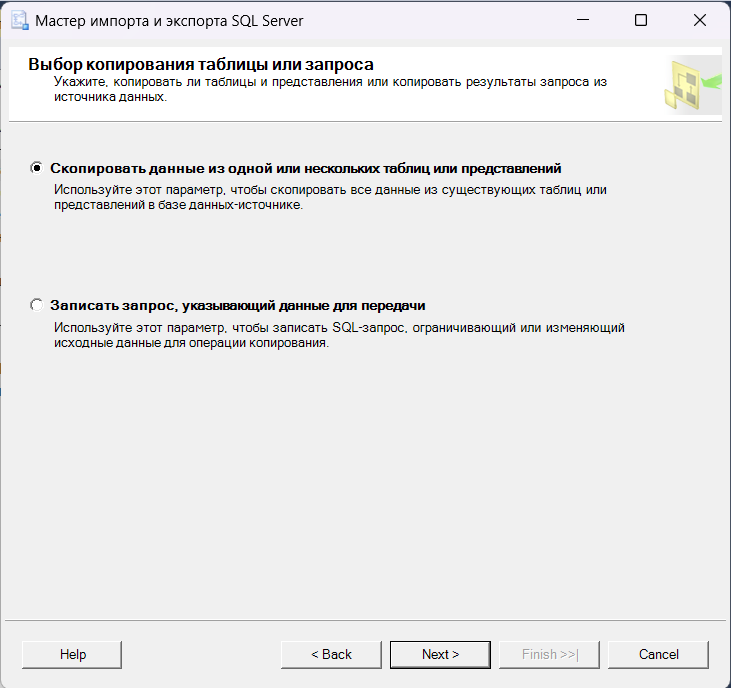


Рисунок 8 – Выбор копирования таблицы или запросов

На рисунке 9 мы выбираем таблицы для копирования из Excel которые значение которых потом будет в базе данных.

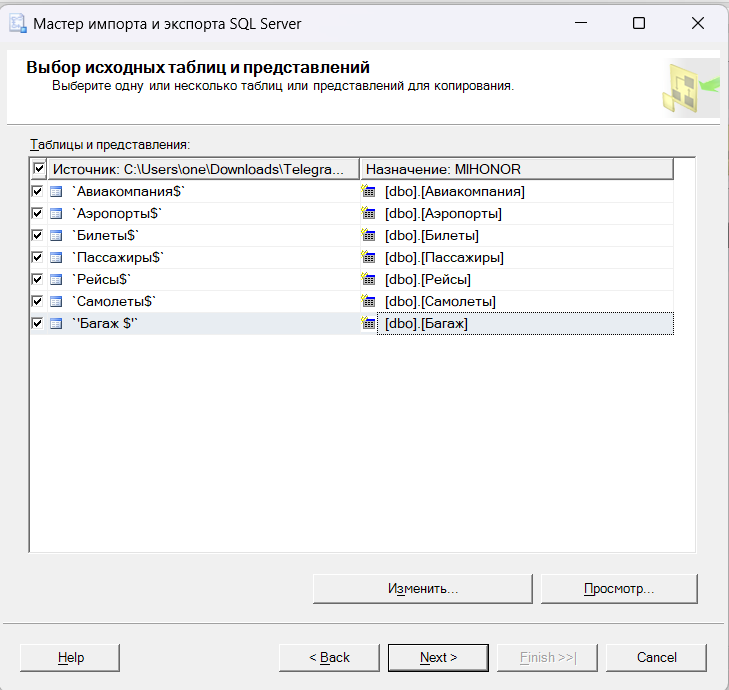


Рисунок 9 – Выбор копирования таблицы или запросов

На рисунке 10 мы выбираем «запустить немедленно» без сохранения пакета служб SSIS.

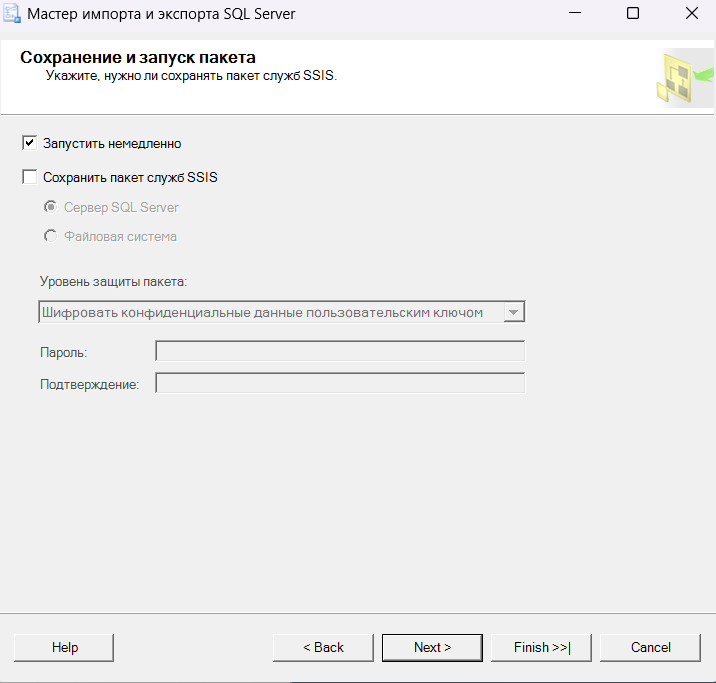


Рисунок 10 – Сохранение и запуск пакета

На рисунке 11 выведены значения таблицы «Авиакомпании».



Рисунок 11 – Таблица «Авиакомпании»

На рисунке 12 выведены значения таблицы «Аэропорты».

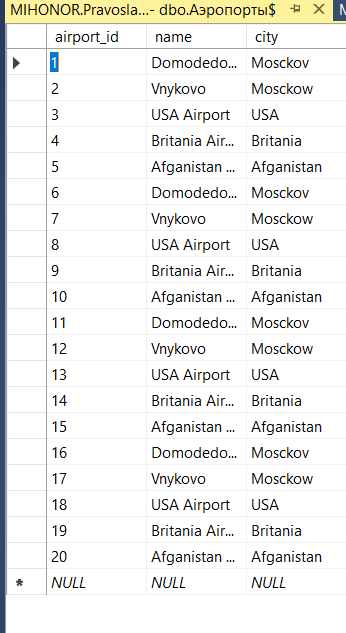
****

Рисунок 12 – Таблица «Аэропорты»

На рисунке 13 выведены значения таблицы «Багаж».

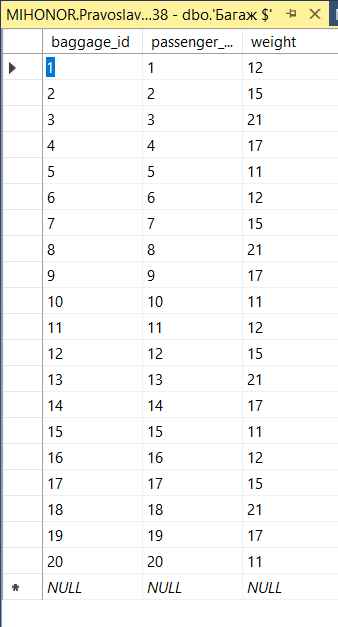


Рисунок 13 – Таблица «Багаж»

На рисунке 14 выведены значения таблицы «Билеты».

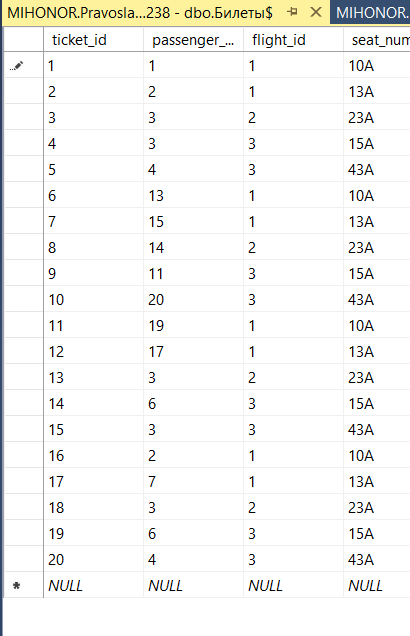


Рисунок 14 – Таблица «Билеты»

На рисунке 15 выведены значения таблицы «Пассажиры».

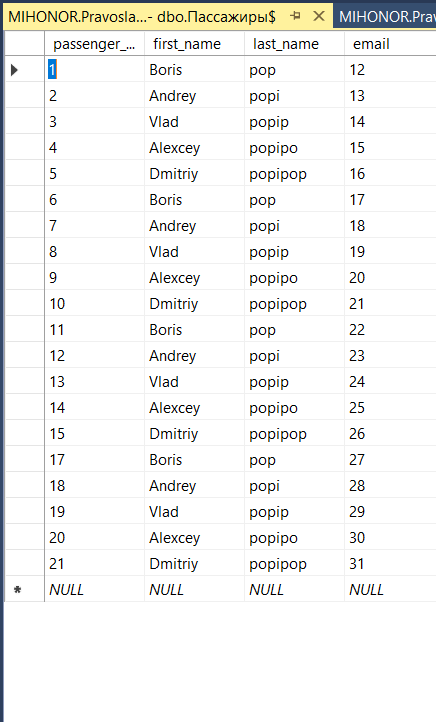


Рисунок 15 – Таблица «Пассажиры»

На рисунке 16 выведены значения таблицы «Рейсы».

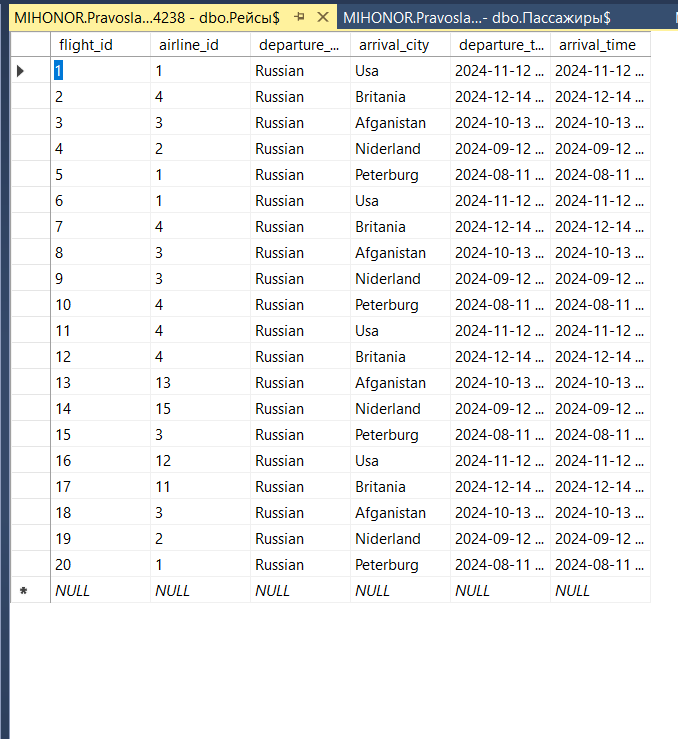


Рисунок 16 – Таблица «Рейсы»

На рисунке 17 выведены значения таблицы «Самолеты».

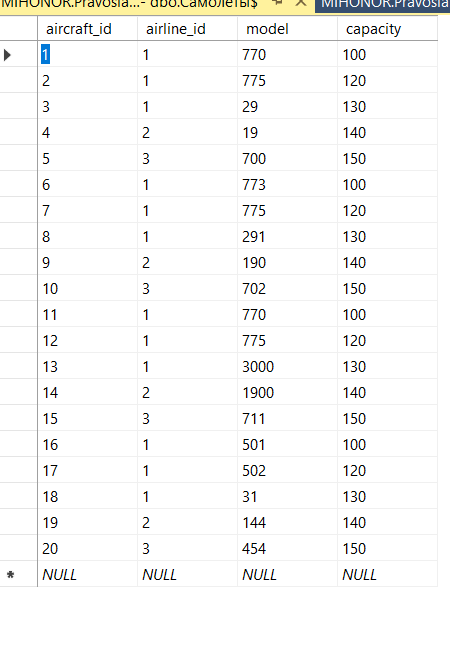


Рисунок 17 – Таблица «Самолеты»

## **2.4 Поддержка целостности данных**

### **2.4.1 Триггеры**

1. tr\_AirlineAdded\_1

|  |
| --- |
| CREATE TRIGGER tr\_AirlineAdded\_1  ON Airlines  AFTER INSERT  AS  BEGIN  PRINT 'Новая запись добавлена в таблицу "Авиакомпании".';  END; |

На рисунке 18 показан триггер, который активируется при вставке новой записи в таблицу «Airline».

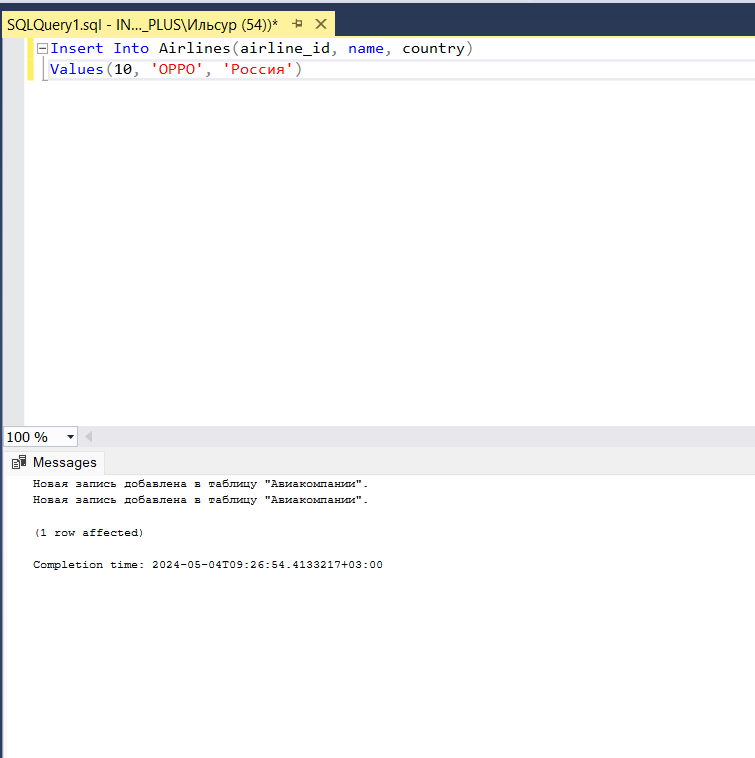


Рисунок 18 – Триггер №1

1. tr\_FlightAdded\_1

|  |
| --- |
| CREATE TRIGGER tr\_FlightAdded\_1  ON Flights  AFTER INSERT  AS  BEGIN  PRINT 'Новая запись добавлена в таблицу "Рейсы".';  END; |

На рисунке 19 показан триггер, который активируется при вставке новой записи в таблицу «Flight».

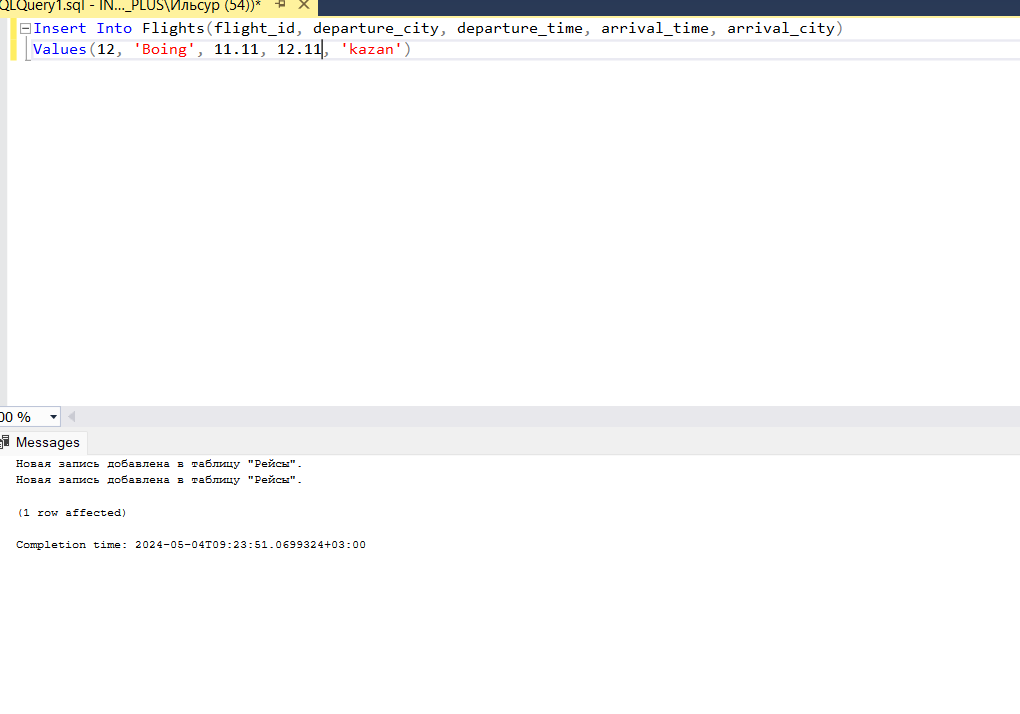


Рисунок 19 – Триггер №2

1. tr\_AircraftAdded\_1

|  |
| --- |
| CREATE TRIGGER tr\_AircraftAdded\_1  ON Aircraft  AFTER INSERT  AS  BEGIN  PRINT 'Новая запись добавлена в таблицу "Самолеты".';  END; |

На рисунке 20 показан триггер, который активируется при вставке новой записи в таблицу «Aircraft».

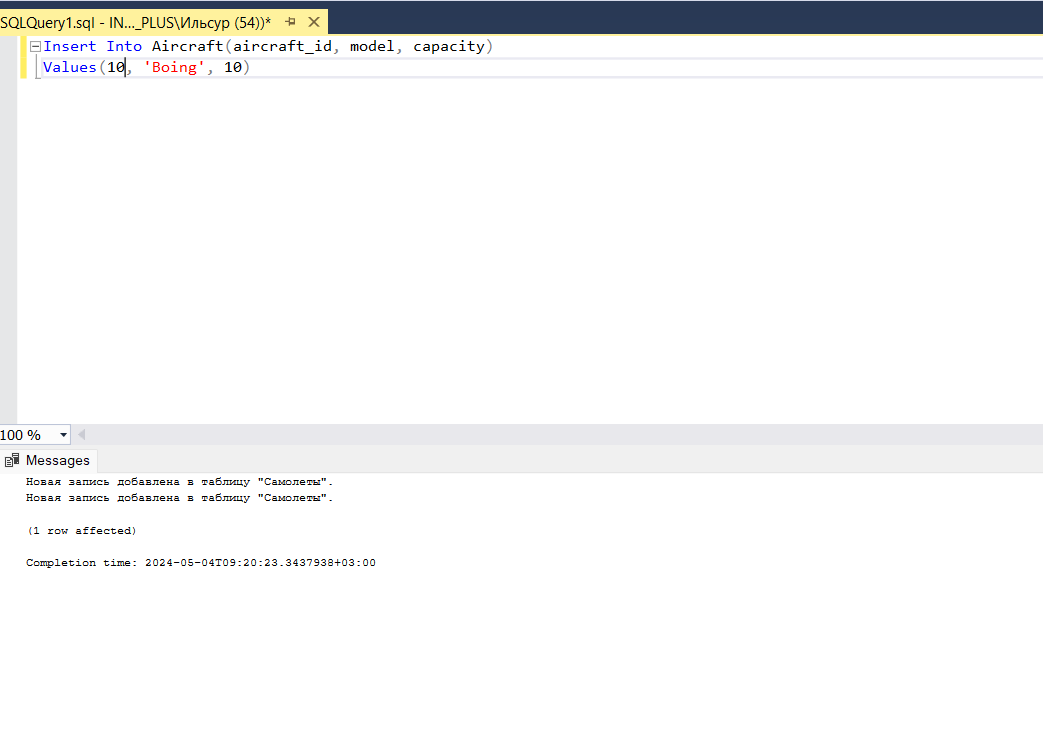


Рисунок 20 – Триггер №3

1. tr\_PassengerAdded­\_1

|  |
| --- |
| CREATE TRIGGER tr\_PassengerAdded\_1  ON Passengers  AFTER INSERT  AS  BEGIN  PRINT 'Новая запись добавлена в таблицу "Пассажиры".';  END; |

На рисунке 21 показан триггер, который активируется при вставке новой записи в таблицу «Passenger».

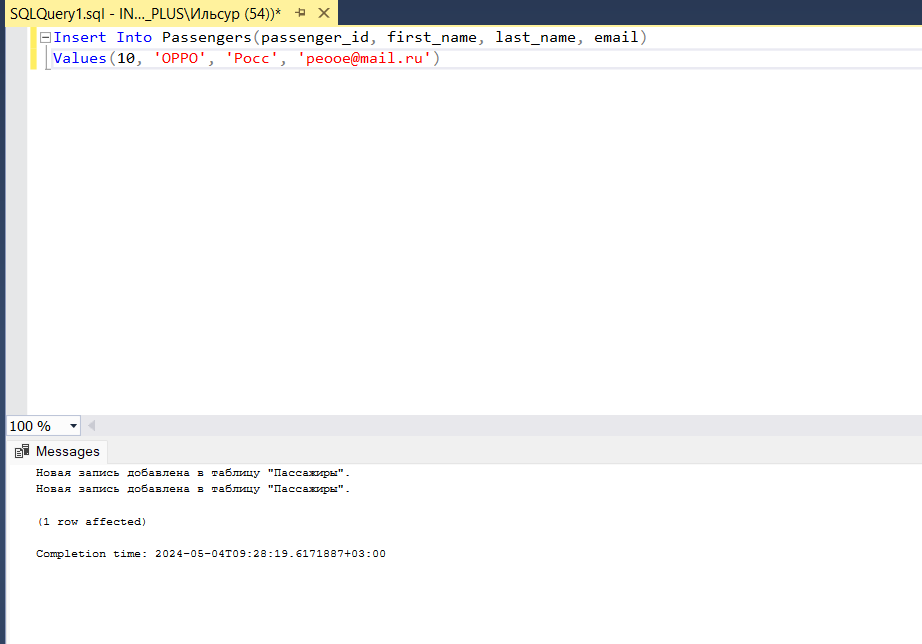


Рисунок 21 – Триггер №4

1. tr\_BaggageAdded\_1

|  |
| --- |
| CREATE TRIGGER tr\_BaggageAdded\_1  ON Baggage  AFTER INSERT  AS  BEGIN  PRINT 'Новая запись добавлена в таблицу "Багаж".';  END; |

На рисунке 22 показан триггер который активируется при вставке новой записи в таблицу «Baggage». Результат:

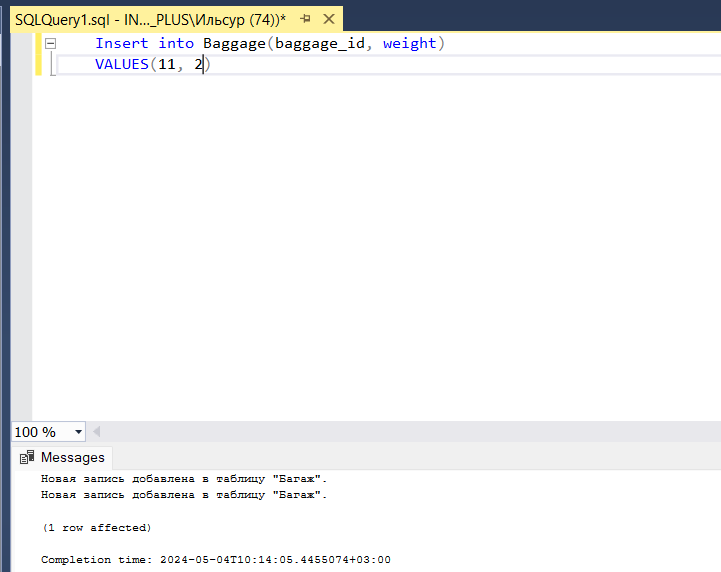


Рисунок 22 – Триггер №5

### **2.4.2 Представления**

1. vw\_Airlines

|  |
| --- |
| CREATE VIEW vw\_Airlines AS  SELECT airline\_id, name, country  FROM Airlines; |

На рисунке 23 показано представление, которое выбирает «airline\_id» «company» «name» загружая это из таблицы «Airlines».

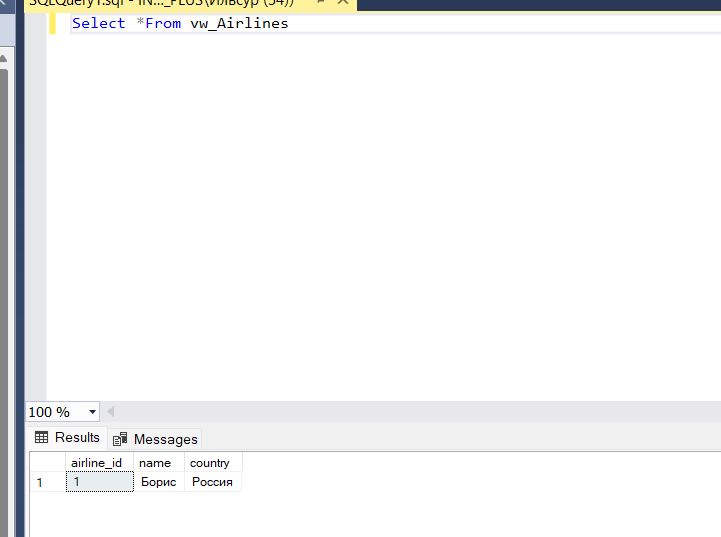


Рисунок 23 – Представление №1

1. vw\_Flights

|  |
| --- |
| CREATE VIEW vw\_Flights AS  SELECT flight\_id, airline\_id, departure\_city, arrival\_city, departure\_time, arrival\_time  FROM Flights; |

На рисунке 24 показано представление которое выбирает «Flights\_id» «airline\_id» «department\_city» и «arrival\_time» и «department\_time» загружая это из таблицы «Flights».

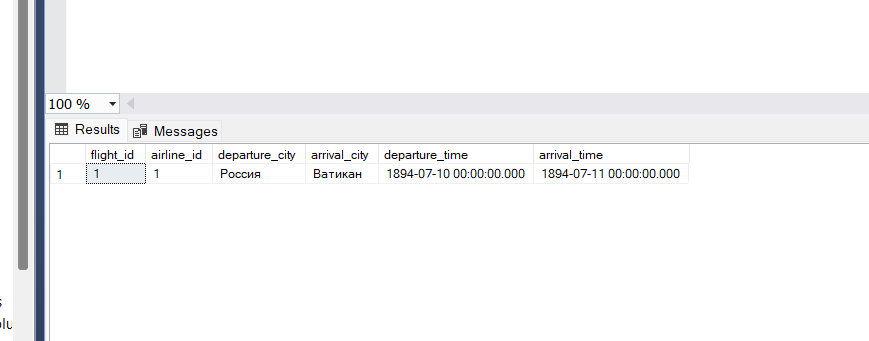


Рисунок 24 – Представление №2

1. vw\_Aircraft

|  |
| --- |
| CREATE VIEW vw\_Aircraft AS  SELECT aircraft\_id, airline\_id, model, capacity  FROM Aircraft; |

 На рисунке 25 показано представление, которое выбирает «aircraft\_id» «airline\_id» «model» и «capacity» загружая это из таблицы Aircraft.

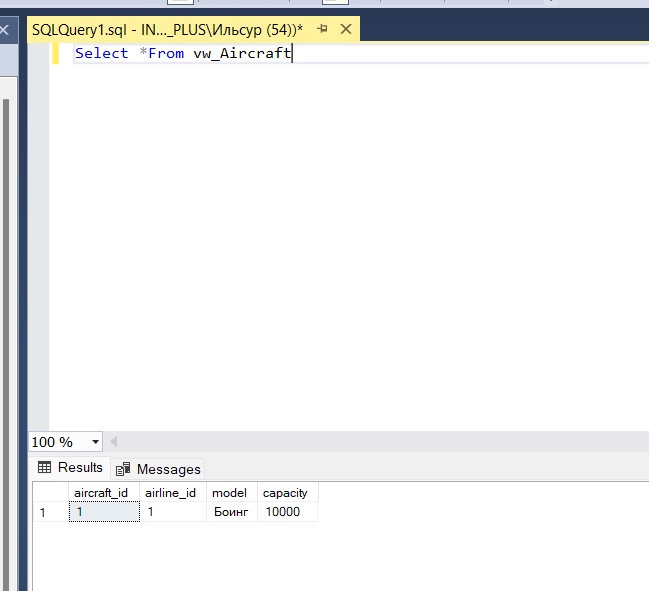


Рисунок 25 – Представление №3

1. vw\_Passengers

|  |
| --- |
| CREATE VIEW vw\_Passengers AS  SELECT passenger\_id, first\_name, last\_name, email  FROM Passengers; |

На рисунке 26 показано представление которое выбирает «passanger\_id» и «first\_name» и «last\_name» и «mail» из таблицы «Passengers».

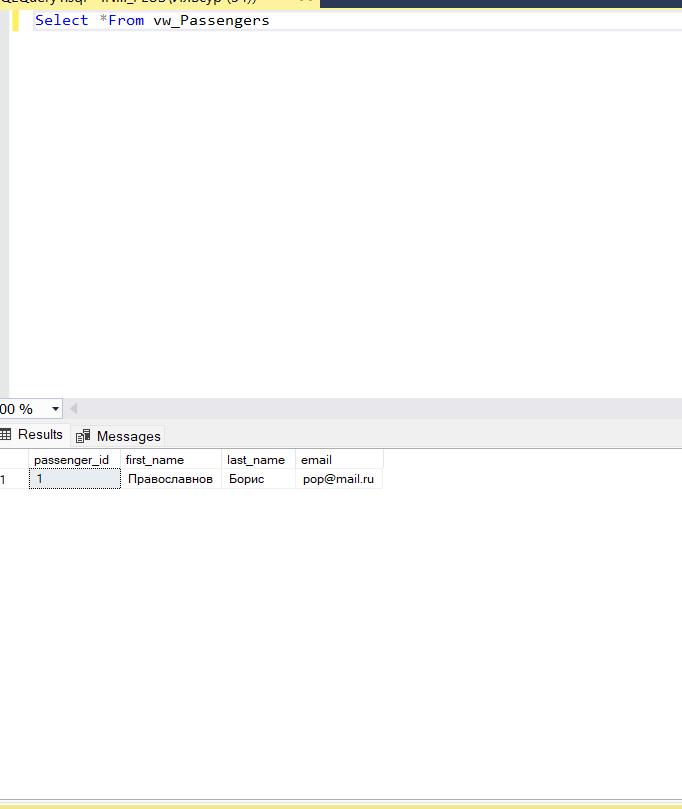


Рисунок 26 – Представление №4

1. vw\_Baggage

|  |
| --- |
| CREATE VIEW vw\_Baggage AS  SELECT baggage\_id, passenger\_id, weight  FROM Baggage; |

На рисунке 27 показано представление, которое выбирает «baggage\_id» и «passenger\_id» и «weight» из таблицы «Baggage».

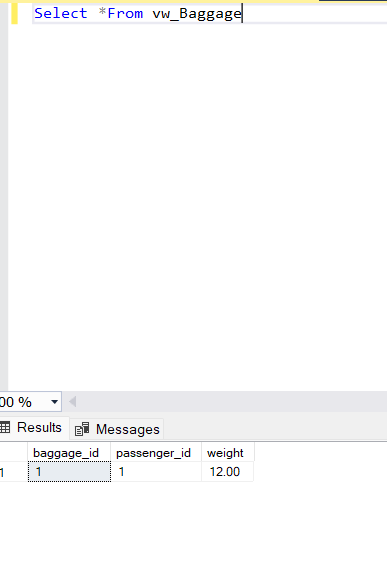


Рисунок 27 – Представление №5

### **2.4.3 Хранимые процедуры**

1. sp\_Update\_Airline

|  |
| --- |
| CREATE PRocedure sp\_Update\_Airline @Airline\_id int  AS  Begin  Update dbo.Airlines  SET name = 'Pobeda'  where @Airline\_id = 1;  End; |

На рисунке 28 изображена процедура которая заменяет значение столбца «name» компании на новое значение но при условии что этот компания с «ID» 1.

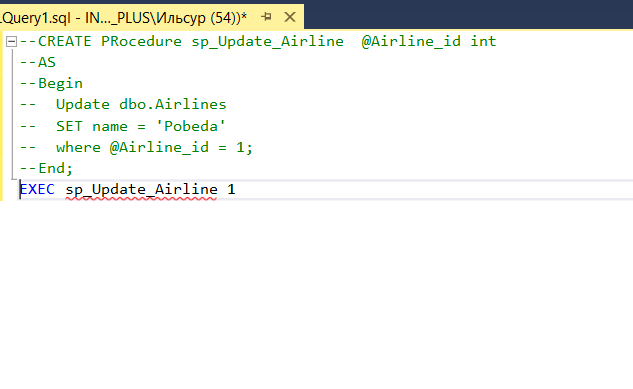


Рисунок 28 – Хранимая процедура №1

1. sp\_Update\_Flights

|  |
| --- |
| CREATE PRocedure sp\_Update\_Flights @Flight\_id int  AS  Begin  Update dbo.Flights  SET arrival\_city = 'kazan'  where @Flight\_id = 1;  End; |

На рисунке 29 изображена процедура, которая заменяет значение столбца «arrival\_city» самолета на новое значение, но при условии, что этот рейс с «ID» 1

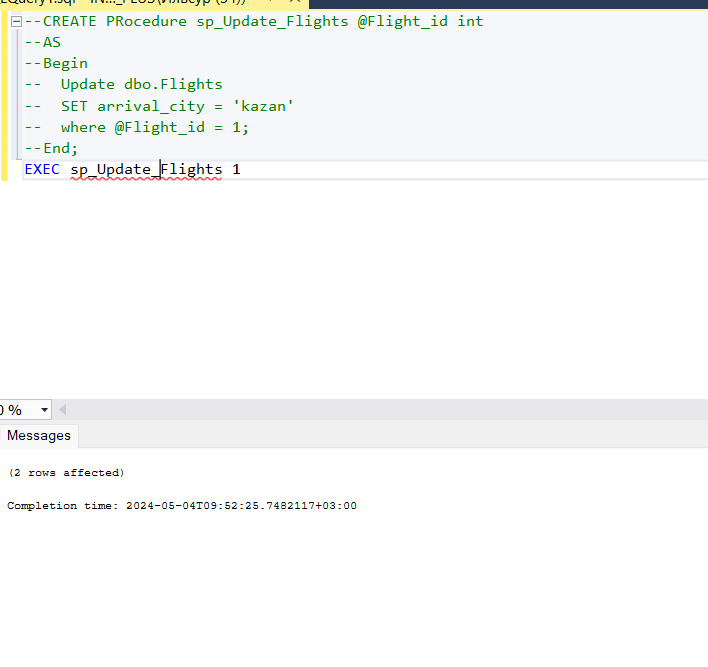


Рисунок 29 – Хранимая процедура №2

1. sp\_Update\_Aircraft

|  |
| --- |
| CREATE PRocedure sp\_Update\_Aircraft @aircraft\_id int  AS  Begin  Update dbo.Aircraft  SET capacity = 111234  where airline\_id = 1;  End; |

На рисунке 30 изображена процедура, которая заменяет значение столбца «capacity» самолета на новое значение, но при условии, что этот самолет принадлежит компании с «ID» 1.

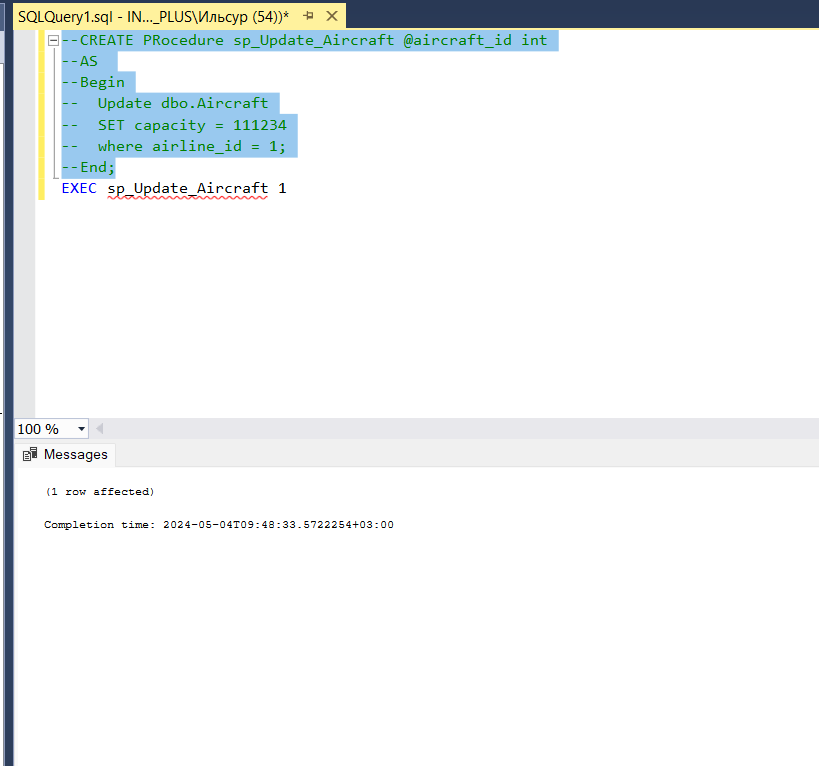


Рисунок 30 – Хранимая процедура №3

1. sp\_Update\_Passanger

|  |
| --- |
| CREATE PRocedure sp\_Update\_Passanger @Passanger\_id int  AS  Begin  Update dbo.Passengers  SET first\_name = 'Boris'  where @Passanger\_id = 1;  End; |

На рисунке 31 изображена процедура, которая заменяет значение столбца «first\_name» пассажира самолета на новое значение, но при условии, что этот пассажир с «ID» 1.

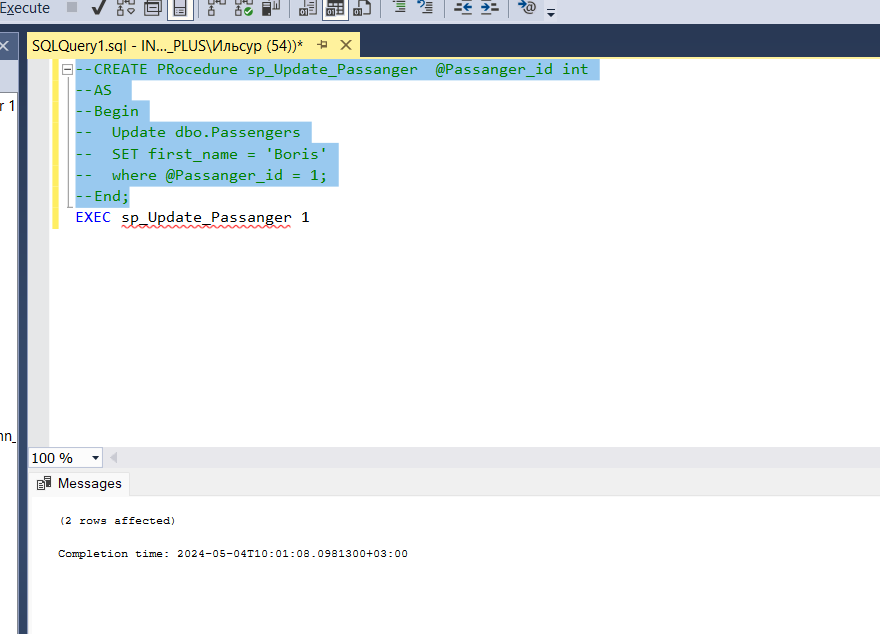


Рисунок 31 – Хранимая процедура №4

1. sp\_Update\_Bagage

|  |
| --- |
| CREATE PRocedure sp\_Update\_Bagage @Bagage\_id int  AS  Begin  Update dbo.Baggage  SET weight = 1000  where @Bagage\_id = 1;  End; |

На рисунке 32 изображена процедура, которая заменяет значение столбца «weight» багажа самолета на новое значение, но при условии, что этот багаж с «ID» 1.

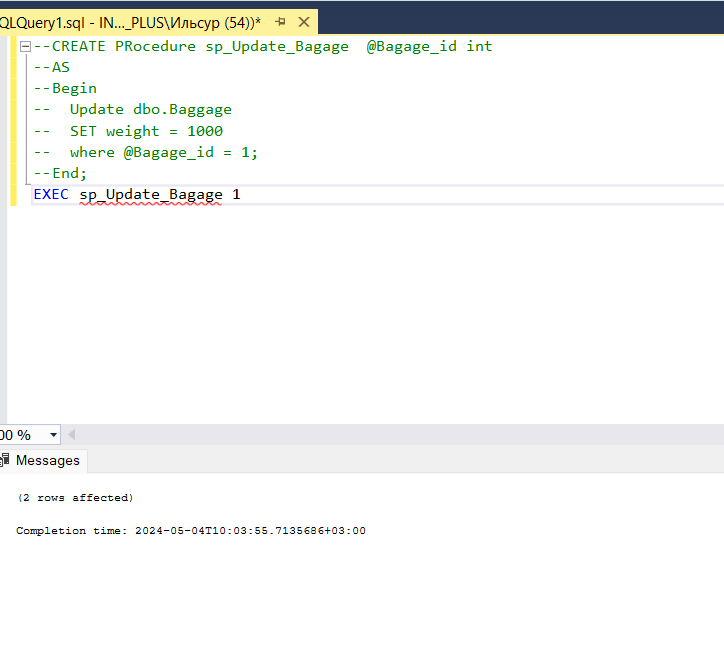


Рисунок 32 – Хранимая процедура №5