**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**

**Государственное профессиональное образовательное учреждение**

**«Воркутинский арктический горно-политехнический колледж»**

**КУРСОВОЙ ПРОЕКТ**

По дисциплине МДК.07.01 Управление и автоматизация баз данных

**Разработка базы данных «Продажа авиабилетов»**

Выполнил студент гр. ИСП-21 /\_\_\_\_\_\_\_\_\_\_\_\_/Олийнык Юрий Дмитриевич/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОВЕРИЛ**

Научный руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_/ Егоров Данил Павлович/

(подпись) (Ф.И.О.)

Воркута

2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc188260327)

[ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ 4](#_Toc188260328)

[1.1 СУБД SQL Server 4](#_Toc188260329)

[1.2 Microsoft SQL Server Management Studio 5](#_Toc188260330)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 6](#_Toc188260331)

[2.1 Разработка диаграммы ERD 6](#_Toc188260332)

[2.2 Словарь данных 10](#_Toc188260333)

[ГЛАВА 3. Разработка базы данных 16](#_Toc188260334)

[3.1 Разработка базы данных 16](#_Toc188260335)

[3.2 Внедрение в безопасность SQL Server 24](#_Toc188260336)

[3.3 Управление безопасностью уровня сервера 25](#_Toc188260337)

[3.4 Управление участниками уровня базы данных 26](#_Toc188260338)

[3.5 Управление разрешениями уровня базы данных 27](#_Toc188260339)

[3.6 Резервное копирование баз данных 28](#_Toc188260340)

[3.7 Шифрование данных баз данных 29](#_Toc188260341)

[ЗАКЛЮЧЕНИЕ 32](#_Toc188260342)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 33](#_Toc188260343)

# **ВВЕДЕНИЕ**

Современный мир характеризуется стремительным развитием онлайн-сервисов, и рынок продажи авиабилетов не исключение. Эффективная система управления данными о билетах, рейсах и пассажирах критически важна для успешной работы любого онлайн-агентства или авиакомпании.

Так, разработка базы данных для продажи авиабилетов является актуальной задачей, поскольку она решает важную практическую проблему и способствует развитию знаний и навыков в области проектирования и администрирования баз данных, востребованных в современных IT-компаниях.

**Объект:** база данных «Продажа авиабилетов».

**Предмет:** автоматизация базы данных «Продажа авиабилетов».

**Цель работы:** разработать базу данных «Продажи авиабилетов».

**Задачи:**

* выбор инструментария;
* проектирование базы данных;
* разработать базу данных.

# **ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ**

# **СУБД SQL Server**

SQL Server — это система управления базами данных (СУБД) от Microsoft, которая позволяет создавать, модифицировать и управлять базами данных, а также контролировать доступ к ним. Она используется для хранения и обработки данных в приложениях и веб-сайтах, поддерживая различные языки программирования и интерфейсы.

Центральным аспектом в MS SQL Server, как и в любой СУБД, является база данных, представляющая собой организованное хранилище данных. Физически база данных может быть представлена файлом на жестком диске, хотя это не обязательно. Для администрирования баз данных применяются СУБД (DBMS).

SQL Server характеризуется следующими особенностями:

* **Производительность**: работает быстро.
* **Надежность и безопасность**: предоставляет шифрование данных.
* **Простота**: относительно легко в использовании и администрировании.

Для организации баз данных MS SQL Server использует реляционную модель, разработанную Эдгаром Коддом в 1970 году, которая на сегодняшний день является стандартом. Реляционная модель предполагает хранение данных в таблицах, состоящих из строк и столбцов, где каждая строка представляет отдельный объект, а столбцы содержат его атрибуты.

Дополнительно, SQL Server поддерживает:

* **Транзакции**: обеспечивает целостность данных с помощью механизма управления транзакциями, что позволяет выполнять группы операций как единое целое.
* **Индексы**: для ускорения поиска и сортировки данных, что значительно повышает производительность запросов.
* **Хранимые процедуры и триггеры**: для автоматизации задач и обеспечения бизнес-логики на уровне базы данных.
* **Интеграцию с облачными сервисами**: что позволяет использовать SQL Server в гибридных и облачных архитектурах, обеспечивая масштабируемость и доступность.
* Эти особенности делают SQL Server мощным инструментом для работы с данными в различных приложениях и системах.

# **Microsoft SQL Server Management Studio**

SQL Server Management Studio (SSMS) - это интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до базы данных SQL Azure. SSMS предоставляет инструменты для настройки, мониторинга и администрирования экземпляров SQL Server и баз данных. Используйте SSMS для развертывания, мониторинга и обновления компонентов уровня данных, используемых вашими приложениями, а также для создания запросов и сценариев.

Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью ими управлять. Object Explorer предоставляет иерархический вид всех объектов, таких как базы данных, таблицы, представления и процедуры, что упрощает навигацию и управление.

Также есть SQL Server Management Studio Express для Express версии сервера, которая является бесплатной. Однако в ней нет поддержки ряда компонентов (Analysis Services, Integration Services, Notification Services, Reporting Services) и SQL Server 2005 Mobile Edition. SSMS Express идеально подходит для небольших проектов и разработчиков, которые только начинают работать с SQL Server, но для более сложных задач рекомендуется использовать полную версию SSMS, которая предлагает расширенные функции и возможности.

# **ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

# **2.1 Разработка диаграммы ERD**

**Описание:**

Схема «сущность-связь» (также ERD или ER-диаграмма) — это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы, ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ERдиаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные лини для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи - глаголов.

B ER-моделях и моделях данных обычно выделяют до трех уровней детализации:

* **Концептуальная модель данных** - схема наивысшего уровня с минимальным количеством подробностей. Достоинство этого подхода заключается в возможности отобразить общую структуру модели и всю архитектуру системы. Менее масштабные системы могут обойтись и без этой модели. В этом случае можно сразу переходить к логической модели;
* **Логическая модель** **данных** содержит более подробную информацию, нежели концептуальная модель. На этом уровне определяются более подробные операционные и транзакционные сущности. Логическая модель не зависит от технологии, в которой она будет применяться;
* **Физическая модель данных**: на основе каждой логической модели данных можно составить одну или две физических модели. В последних должно присутствовать достаточно технических подробностей для составления и внедрения самой базы данных.

**Элементы:**

Для того чтобы построить ER-диаграмму, можно использовать разные нотации:

* Нотация IDEF1X. Ее относят к фундаментальным, но на практике давно не используют, потому что есть более удобные варианты;
* Нотация Чена. Классическая нотация, которая состоит из простых символов – прямоугольников, овалов и линий. Из-за этого нотацию часто используют для концептуальных моделей, которые презентуют заказчику;
* Нотация Мартина. Ее еще называют «воронья лапка» (от англ. Crow's Foot). Она компактнее нотации Чена, поэтому ее используют для построения ER-моделей логического уровня, когда нужно описать в модели все атрибуты сущностей.

В нотациях Чена и Мартина есть одинаковые элементы: сущности, атрибуты и связи. Но эти элементы диаграмм обозначают разными символами.

Элементы ER-диаграммы в нотации Чена соединяют линиями. Если линия соединяет две сущности, сверху обозначают тип связи:

● 1:1 — «один-к-одному»;

● 1:N — «один-ко-многим»;

● M:N — «многие-ко-многим».

В нотации Мартина сущность также вписывают в прямоугольник, а атрибуты и связи обозначают по-другому:

● атрибуты перечисляют прямо под сущностью;

● связи рисуют разными соединительными линиями (рис. 2.1).

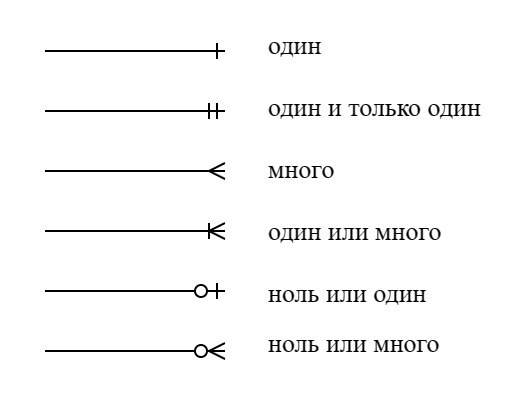


Рисунок 2.1 Виды соединительных линий нотации Мартина

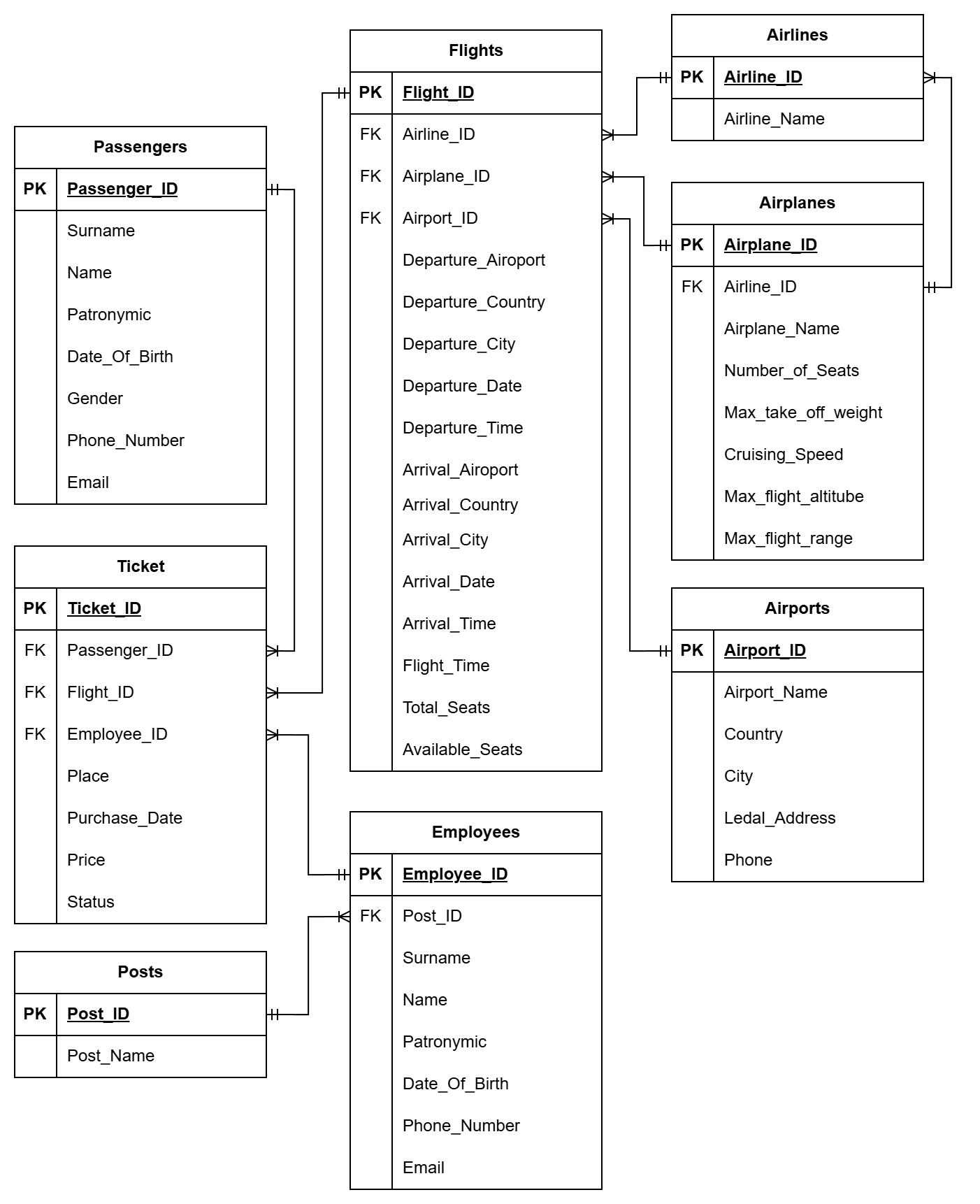


Рисунок 2.2 Диаграмма «Сущность-связь» (Entity-Relationship diagram)

На рис. 2.2 показана ER-диаграмма информационной системы «Продажа авиабилетов». Она показывает сущности и как они связаны между собой. Сущностями являются: Пассажиры (Passengers), Сотрудники (Employees), Рейсы (Flights), Билеты (Tickets), Должности (Posts), Аэропорты (Airports), Авиакомпании (Airlines). Атрибутами сущности Passengers являются: Passenger\_ID, Surname, Name, Patronymic, Date\_Of\_Birth, Gender, Phone\_Number, Email. Атрибутами сущности Employees являются: Employee\_ID, Post\_ID, Surname, Name, Patronymic, Date\_Of\_Birth, Phone\_Number, Email. Атрибутами сущности Posts являются: Post\_ID, Post\_Name. Атрибутами сущности Airports являются: Airport\_ID, Airport\_Name, Country, City, Ledal\_Address, Phone. Атрибутами сущности Airlines являются: Airline\_ID, Airline\_Name. Атрибутами сущности Airplanes являются: Airplane\_ID, Airline\_ID, Airplane\_Name, Number\_of\_Seats, Max\_take\_off\_weight, Cruising \_Speed, Max\_Flight\_altitube, Max\_Flight\_range.

# **2.2 Словарь данных**

Словарь данных (англ. — data dictionary) справочник или централизованное описание метаданных, дающее представление о структуре и содержании данных. У этого термина есть и другое значение, в рамках данной статьи не используемое: словарём данных называют технику моделирования, дополняющую требования при проектировании информационных систем, в культуре Systems Analysis and Design.

Основные компоненты словаря данных:

1. **Сущности**: Основные объекты, о которых идет речь в системе (например, Пользователь, Заказ, Продукт).
2. **Атрибуты**: Характеристики сущностей (например, для сущности Пользователь это может быть Имя, Email, Дата рождения).
3. **Связи**: Отношения между сущностями (например, Пользователь может делать Заказы).
4. **Типы данных**: Указывают, какой тип данных используется для каждого атрибута (например, строка, целое число, дата).
5. **Ограничения**: Правила, которые определяют допустимые значения для атрибутов (например, уникальность, обязательность).
6. **Идентификаторы**: Уникальные ключи, которые используются для идентификации каждой сущности (например, ID Пользователя).

Таблица 1. Пассажиры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| Passengers | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Passenger\_ID** | INT | Y | Уникальный номер (ID) пассажира |
|  | **Surname** | VARCHAR (100) | Y | Фамилия пассажира |
|  | **Name** | VARCHAR (50) | Y | Имя пассажира |
|  | **Patronymic** | VARCHAR (100) | Y | Отчество пассажира |
|  | **Date\_Of\_Birth** | DATE | Y | Дата рождения пассажира |
|  | **Gender** | VARCHAR (20) | Y | Пол пассажира |
|  | **Phone\_Number** | BIGINT | Y | Номер телефона пассажира |
|  | **Email** | VARCHAR (50) | Y | Почта пассажира |

Таблица 2. Билеты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| Tickets | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Ticket\_ID** | INT | Y | Уникальный номер (ID) билета |
| FK | **Passenger\_ID** | INT | Y | Уникальный номер (ID) пассажира (внешний ключ) |
| FK | **Flight\_ID** | INT | Y | Уникальный номер (ID) рейса (внешний ключ) |
| FK | **Employee\_ID** | INT | Y | Уникальный номер (ID) сотрудника (Внешний ключ) |
|  | **Place** | VARCHAR (5) | Y | Место в самолёте |
|  | **Purchase\_Date** | DATE | Y | Дата приобретения билета |
|  | **Price** | DECIMAL (20,2) | Y | Стоимость |
|  | **Status** | VARCHAR (50) | Y | Статус билета |

Таблица 3. Должности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| Posts | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Post\_ID** | INT | Y | Уникальный номер (ID) должности |
|  | **Post\_Name** | VARCHAR (80) | Y | Название должности |

Таблица 4. Сотрудники

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| Employees | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Employee\_ID** | INT | Y | Уникальный номер (ID) сотрудника |
| FK | **Post\_ID** | INT | Y | Уникальный номер (ID) должности сотрудника (внешний ключ) |
|  | **Surname** | VARCHAR (100) | Y | Фамилия сотрудника |
|  | **Name** | VARCHAR (50) | Y | Имя сотрудника |
|  | **Patronymic** | VARCHAR (100) | Y | Отчество сотрудника |
|  | **Date\_Of\_Birth** | DATE | Y | Дата рождения сотрудника |
|  | **Phone\_Number** | BIGINT | Y | Номер телефона сотрудника |
|  | **Email** | VARCHAR (50) | Y | Почта сотрудника |
|  |  |  |  |  |
|  | |  |  | Таблица 5. Рейсы |
| Data Dictionary | | | | |
| Flights | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Flight\_ID** | INT | Y | Уникальный номер (ID) билета |
| FK | **Airline\_ID** | INT | Y | Уникальный номер (ID) авиакомпании (внешний ключ) |
| FK | **Airplane\_ID** | INT | Y | Уникальный номер (ID) самолёта (внешний ключ) |
| FK | **Airport\_ID** | INT | Y | Уникальный номер (ID) аэропорта (внешний ключ) |
|  | **Departure\_Airport** | VARCHAR (100) | Y | Аэропорт отправления |
|  | **Departure\_Country** | VARCHAR (50) | Y | Страна отправления |
|  | **Departure\_Сity** | VARCHAR (60) | Y | Город отправления |
|  | **Departure\_Date** | DATE | Y | Дата отправления |
|  | **Departure\_Time** | TIME | Y | Время отправления |
|  | **Arrival\_Airoport** | VARCHAR (100) | Y | Аэропорт прибытия |
|  | **Arrival\_Country** | VARCHAR (50) | Y | Страна прибытия |
|  | **Arrival\_City** | VARCHAR (60) | Y | Город прибытия |
|  | **Arrival\_Date** | DATE | Y | Дата прибытия |
|  | **Arrival\_Time** | TIME | Y | Время прибытия |
|  | **Flight\_Time** | TIME | Y | Время полёта |
|  | **Total\_Seats** | INT | Y | Общее количество мест |
|  | **Aviable\_Seats** | INT | Y | Количество свободных мест |

Таблица 6. Авиакомпании

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| Airlines | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Airline\_ID** | INT | Y | Уникальный номер (ID) авиакомпании |
|  | **Airline\_Name** | VARCHAR (100) | Y | Название авиакомпании |

Таблица 7. Аэропорты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| Airports | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Airport\_ID** | INT | Y | Уникальный номер (ID) аэропорта |
|  | **Airport\_Name** | VARCHAR (100) | Y | Название аэропорта |
|  | **Country** | VARCHAR (50) | Y | Название страны, в котором находится аэропорт |
|  | **City** | VARCHAR (60) | Y | Название города, в котором находится аэропорт |
|  | **Legal\_Address** | VARCHAR (100) | Y | Юридический адрес аэропорта |
|  | **Phone** | BIGINT | Y | Номер телефона аэропорта |

Таблица 8. Самолёты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| Airplanes | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Airplane\_ID** | INT | Y | Уникальный номер (ID) самолёта |
| FK | **Airline\_ID** | INT | Y | Уникальный номер (ID) авиакомпании (внешний ключ) |
|  | **Airplane\_Name** | VARCHAR (100) | Y | Название самолёта |
|  | **Number\_of\_Seats** | INT | Y | Название страны, в котором находится аэропорт |
|  | **Max\_take\_off\_weight** | INT | Y | Максимальный взлётный вес, кг |
|  | **Cruising\_Speed** | INT | Y | Крейсерская скорость, км/ч |
|  | **Max\_flight\_altitube** | INT | Y | Максимальная высота полета, м |
|  | **Max\_flight\_range** | INT | Y | Максимальная дальность полета, км |

# **ГЛАВА 3. Разработка базы данных**

# **3.1 Разработка базы данных**

Используя SQL Server Management Studio, разработаем базу данных, согласно техническому заданию:

CREATE DATABASE Sale\_of\_air\_tickets;

USE Sale\_of\_air\_tickets;

CREATE TABLE Passengers(

Passenger\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Surname VARCHAR(100) NOT NULL,

Name VARCHAR(50) NOT NULL,

Patronymic VARCHAR(100) NOT NULL,

Date\_Of\_Birth DATE NOT NULL,

Gender VARCHAR(20) NOT NULL,

Phone\_Number BIGINT NOT NULL,

Email VARCHAR(50) NOT NULL

);

CREATE TABLE Posts(

Post\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Post\_Name VARCHAR(80) NOT NULL

);

CREATE TABLE Employees(

Employee\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Post\_ID INT NOT NULL,

Surname VARCHAR(100) NOT NULL,

Name VARCHAR(50) NOT NULL,

Patronymic VARCHAR(100) NOT NULL,

Date\_Of\_Birth DATE NOT NULL,

Phone\_Number BIGINT NOT NULL,

Email VARCHAR(50) NOT NULL,

FOREIGN KEY (Post\_ID) REFERENCES Posts (Post\_ID)

);

CREATE TABLE Airports(

Airport\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Airport\_Name VARCHAR(100) NOT NULL,

Country VARCHAR(50) NOT NULL,

City VARCHAR(60) NOT NULL,

Legal\_Address VARCHAR(300) NOT NULL,

Phone BIGINT NOT NULL

);

CREATE TABLE Airplanes(

Airplane\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Airline\_ID INT NOT NULL,

Airplane\_Name VARCHAR (100) NOT NULL,

Number\_of\_Seats INT NOT NULL,

Max\_take\_off\_weught INT NOT NULL,

Cruising\_Speed INT NOT NULL,

Max\_flight\_altitube INT NOT NULL,

Max\_flight\_range INT NOT NULL,

FOREIGN KEY (Airline\_ID) REFERENCES Airlines (Airline\_ID)

);

CREATE TABLE Airlines(

Airline\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Airline\_Name VARCHAR(100) NOT NULL

);

CREATE TABLE Flights(

Flight\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Airline\_ID INT NOT NULL,

Airplane\_ID INT NOT NULL,

Airport\_ID INT NOT NULL,

Departure\_Airport VARCHAR(60) NOT NULL,

Departure\_Country VARCHAR(50) NOT NULL,

Departure\_Сity VARCHAR(60) NOT NULL,

Departure\_Date DATE NOT NULL,

Departure\_Time TIME NOT NULL,

Arrival\_Airoport VARCHAR(100) NOT NULL,

Arrival\_Country VARCHAR(50) NOT NULL,

Arrival\_City VARCHAR(60) NOT NULL,

Arrival\_Date DATE NOT NULL,

Arrival\_Time TIME NOT NULL,

Flight\_Time TIME NOT NULL,

Total\_Seats INT NOT NULL,

Available\_Seats INT NOT NULL,

FOREIGN KEY (Airline\_ID) REFERENCES Airlines (Airline\_ID),

FOREIGN KEY (Airplane\_ID) REFERENCES Airplanes (Airplane\_ID),

FOREIGN KEY (Airport\_ID) REFERENCES Airports (Airport\_ID)

);

CREATE TABLE Tickets(

Ticket\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Passenger\_ID INT NOT NULL,

Flight\_ID INT NOT NULL,

Employee\_ID INT NOT NULL,

Place VARCHAR(5) NOT NULL,

Purchase\_Date DATE NOT NULL,

Price DECIMAL(20,2) NOT NULL,

Status VARCHAR(50) NOT NULL,

FOREIGN KEY (Passenger\_ID) REFERENCES Passengers (Passenger\_ID),

FOREIGN KEY (Flight\_ID) REFERENCES Flights (Flight\_ID),

FOREIGN KEY (Employee\_ID) REFERENCES Employees (Employee\_ID)

);

Создаем отдельный скрипт и вносим данные в базу данных, согласно техническому заданию:

USE Sale\_of\_air\_tickets;

INSERT INTO Passengers (Surname, Name, Patronymic, Date\_Of\_Birth, Gender, Phone\_Number, Email) VALUES

('Иванов', 'Иван', 'Иванович', '1985-05-15', 'Мужской', 89001234567, 'ivanov@example.com'),

('Петров', 'Петр', 'Петрович', '1990-08-20', 'Мужской', 89007654321, 'petrov@example.com'),

('Сидорова', 'Анна', 'Сидоровна', '1995-12-30', 'Женский', 89003456789, 'sidorova@example.com'),

('Кузнецов', 'Алексей', 'Алексеевич', '1988-03-10', 'Мужской', 89009876543, 'kuznetsov@example.com'),

('Кузнецова', 'Мария', 'Алексеевна', '2010-06-09', 'Мужской', 89009876543, 'kuznetsov@example.com'),

('Кузнецов', 'Данил', 'Алексеевич', '2009-01-12', 'Мужской', 89009876543, 'kuznetsov@example.com'),

('Смирнова', 'Мария', 'Смирновна', '1992-07-25', 'Женский', 89004567890, 'smirnova@example.com'),

('Попов', 'Дмитрий', 'Дмитриевич', '1980-11-05', 'Мужской', 89001239876, 'popov@example.com'),

('Попов', 'Александр', 'Дмитриевич', '2012-03-27', 'Мужской', 89001239876, 'popov@example.com'),

('Васильева', 'Елена', 'Васильевна', '1993-09-15', 'Женский', 89007654322, 'vasilieva@example.com'),

('Морозов', 'Сергей', 'Сергеевич', '1987-04-12', 'Мужской', 89003456788, 'morozov@example.com'),

('Морозов', 'Иван', 'Сергеевич', '2011-09-21', 'Мужской', 89003456788, 'morozov@example.com'),

('Федорова', 'Ольга', 'Федоровна', '1991-01-22', 'Женский', 89009876544, 'fedorova@example.com'),

('Соловьев', 'Андрей', 'Андреевич', '1983-06-18', 'Мужской', 89004567891, 'solovyev@example.com'),

('Соловьева', 'Кристина', 'Андреевна', '2009-02-13', 'Женский', 89004567891, 'solovyev@example.com');

INSERT INTO Posts (Post\_Name) VALUES

("Главный инженер"),

("Инженер"),

("Начальник отдела по работе с персоналом"),

("Диспетчер"),

("Бухгалтер"),

("Начальник отдела по пассажирской работе"),

("Заведующий билетными кассами"),

("Заведующий справочным бюро"),

("Кассир"),

("Начальник отдела по багажевым перевозкам");

INSERT INTO Employees (Post\_ID, Surname, Name, Patronymic, Date\_Of\_Birth, Phone\_Number, Email) VALUES

(1, 'Дорошев', 'Иван', 'Иванович', '1985-05-15', 89001234567, 'ivanov@example.com'),

(2, 'Лимончук', 'Петр', 'Иванович', '1990-06-20', 89007654321, 'petrov@example.com'),

(3, 'Сидоров', 'Сидор', 'Сидорович', '1988-07-25', 89003456789, 'sidorov@example.com'),

(4, 'Кузнецов', 'Алексей', 'Алексеевич', '1992-08-30', 89009876543, 'kuznetsov@example.com'),

(5, 'Смирнов', 'Дмитрий', 'Дмитриевич', '1980-09-10', 89004567890, 'smirnov@example.com'),

(6, 'Федоров', 'Андрей', 'Андреевич', '1983-10-05', 89001239876, 'fedorov@example.com'),

(7, 'Морозов', 'Сергей', 'Сергеевич', '1995-11-15', 89007654321, 'morozov@example.com'),

(8, 'Васильев', 'Николай', 'Николаевич', '1987-12-20', 89003456712, 'vasiliev@example.com'),

(9, 'Лебедев', 'Олег', 'Олегович', '1991-01-25', 89009876512, 'lebedyev@example.com'),

(9, 'Григорьев', 'Анатолий', 'Анатольевич', '1984-02-28', 89004567812, 'grigoryev@example.com'),

(10, 'Мухамедьяров', 'Вячеслав', 'Редисович', '2006-02-27', 89004567812, 'vycheslav@example.com');

INSERT INTO Airports (Airport\_Name, Country, City, Legal\_Address, Phone) VALUES

('Адлер', 'Россия', 'Адлер', "354340, Краснодарский край, г. Сочи, тер. Аэропорт", 78003011991),

('Шереметьево', 'Россия', 'Москва', "141400, Московская область, город Химки, территория аэропорта Шереметьево", 88002000122),

('Хитроу', 'Великобритания', 'Лондон', "The Compass Centre, Nelson Road, Hounslow, Middlesex, TW6 2GW", 448443351801),

('Джон Кеннеди', 'США', 'Нью-Йорк', "The Port, Authority of NY & NJ, Building 14, Jamaica, New York 11430, United States", 17182444444),

('Токио Ханеда', 'Япония', 'Токио', "Terminal 1, 3-3-2 Haneda Airport, Ota-ku, Tokyo 144-0041, Japan" , 81357578000),

('Сидней', 'Австралия', 'Сидней', "Locked Bag 5000 Sydney International Terminal NSW 2020", 61296679111),

('Внуково', 'Россия', 'Москва', "119027, г. Москва, ул. 2-я Рейсовая, д. 2, к. 3.", 74959375555),

('Пулково', 'Россия', 'Санкт-Петербург', "196210, г. Санкт-Петербург, вн.тер.г. Муниципальный Округ Пулковский Меридиан, ул Стартовая, д. 17, стр. 1, помещ. 548" , 78123373822),

('Домодедово', 'Россия', 'Москва', "обл. Московская, г.о. Домодедово, к. 9", 74959336666),

('Красноярск', 'Россия', 'Красноярск', "663013, Красноярский край, Емельяновский р-н, пгт Емельяново, тер. Аэропорт Красноярск, стр. 2" ,73912266222);

INSERT INTO Airlines (Airline\_Name) VALUES

('Аэрофлот'),

('S7 Airlines'),

('Победа'),

('Уральские авиалинии');

INSERT INTO Airplanes (Airline\_ID, Airplane\_Name, Number\_of\_Seats, Max\_take\_off\_weught, Cruising\_Speed, Max\_flight\_altitube, Max\_flight\_range) VALUES

(1, 'Boeing 737', 168, 80000, 852, 12500, 5765),

(2, 'Boeing 737', 168, 80000, 852, 12500, 5765),

(1, 'Airbus A320', 168, 77000, 840, 11800, 6150),

(2, 'Airbus A319', 128, 75500, 840, 11800, 6850),

(3, 'Boeing 737', 168, 80000, 852, 12500, 5765),

(4, 'Airbus A319', 128, 75500, 840, 11800, 6850),

(3, 'Airbus A319', 128, 75500, 840, 11800, 6850),

(3, 'Boeing 747-400', 522, 396890, 913, 13100, 13450),

(1, 'Boeing 747-400', 522, 396890, 913, 13100, 13450);

INSERT INTO Flights (Airline\_ID, Airplane\_ID, Airport\_ID, Departure\_Airport, Departure\_Country, Departure\_Сity, Departure\_Date, Departure\_Time, Arrival\_Airoport, Arrival\_Country, Arrival\_City, Arrival\_Date, Arrival\_Time, Flight\_Time, Total\_Seats, Available\_Seats) VALUES

(3, 25, 2, 'Шереметьево', 'Россия', 'Москва', '2024-10-01', '18:50:00', 'Адлер', 'Россия', 'Москва', '2024-10-01', '22:30:00', '03:50:00', 128, 24),

(3, 25, 7, 'Внуково', 'Россия', 'Москва', '2024-10-01', '19:00:00', 'Пулково', 'Россия', 'Санкт-петербург', '2024-10-01', '21:15:00', '01:30:00', 128, 40),

(1, 19, 7, 'Внуково', 'Россия', 'Москва', '2024-10-01', '09:30:00', 'Пулково', 'Россия', 'Санкт-петербург', '2024-10-01', '11:05:00', '01:35:00', 168, 67),

(3, 25, 2, 'Шереметьево', 'Россия', 'Москва', '2024-10-01', '19:00:00', 'Адлер', 'Россия', 'Санкт-петербург', '2024-10-01', '21:15:00', '01:30:00', 128, 40),

(2, 20, 7, 'Внуково', 'Россия', 'Москва', '2024-10-01', '09:50:00', 'Пулково', 'Россия', 'Санкт-петербург', '2024-10-01', '11:30:00', '01:40:00', 128, 98),

(1, 19, 2, 'Шереметьево', 'Россия', 'Москва', '2024-10-01', '01:05:00', 'Красноярск', 'Россия', 'Красноярск', '2024-10-01', '09:55:00', '04:50:00', 168, 114);

INSERT INTO Tickets (Passenger\_ID, Flight\_ID, Employee\_ID, Place, Purchase\_Date, Price, Status) VALUES

(1, 26, 9, '1A', '2024-09-24', 9000, 'Paid'),

(3, 25, 10, '8C', '2024-09-12', 8600, 'Paid'),

(2, 26, 9, '4B', '2024-09-18', 9000, 'Paid'),

(4, 28, 10, '7A', '2024-09-29', 7800, 'Paid'),

(5, 28, 10, '7B', '2024-09-29', 7800, 'Paid'),

(6, 28, 10, '7C', '2024-09-29', 7800, 'Paid'),

(15, 29, 9, '3C', '2024-09-21', 8300, 'Cancelled'),

(13, 29, 9, '12B', '2024-09-27', 7900, 'Paid'),

(8, 29, 9, '4A', '2024-09-24', 8300, 'Paid');

Создаем отдельный скрипт и выполнили запросы, согласно техническому заданию:

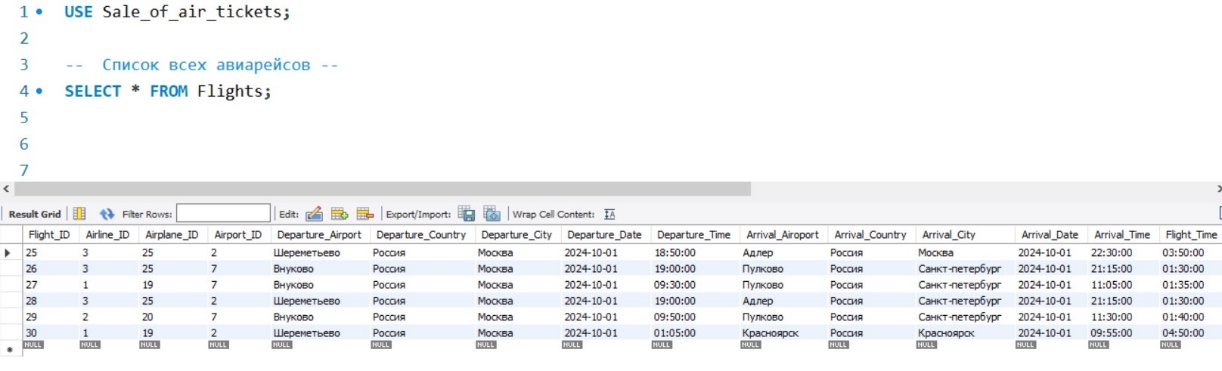


Рисунок 3.1.1 Запрос на список всех авиарейсов

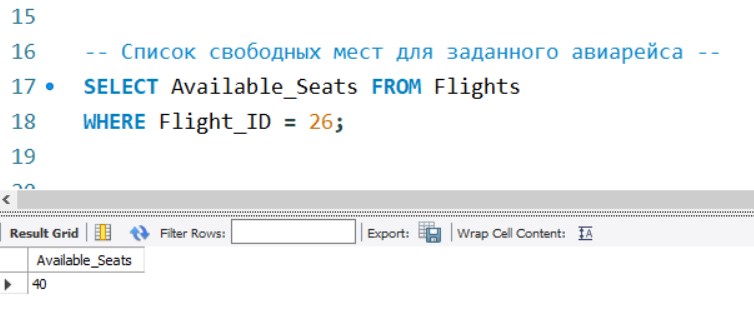


Рисунок 3.1.2 Запрос на список свободных мест для заданного авиарейса

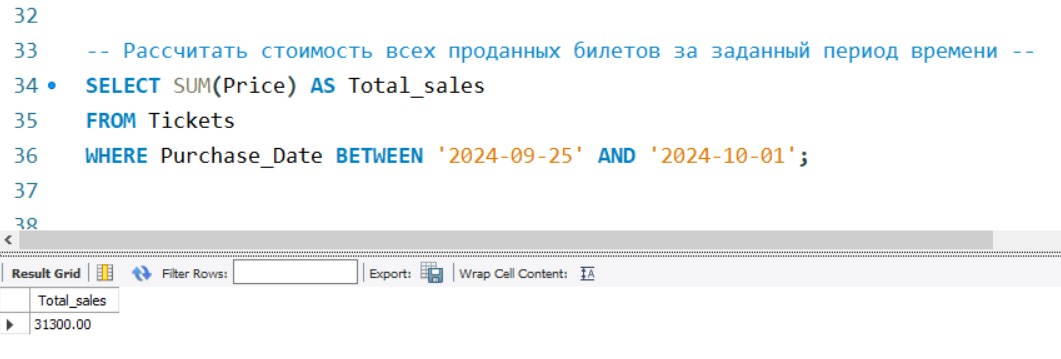


Рисунок 3.1.3 Запрос на расчёт стоимости всех проданных билетов за заданный период времени

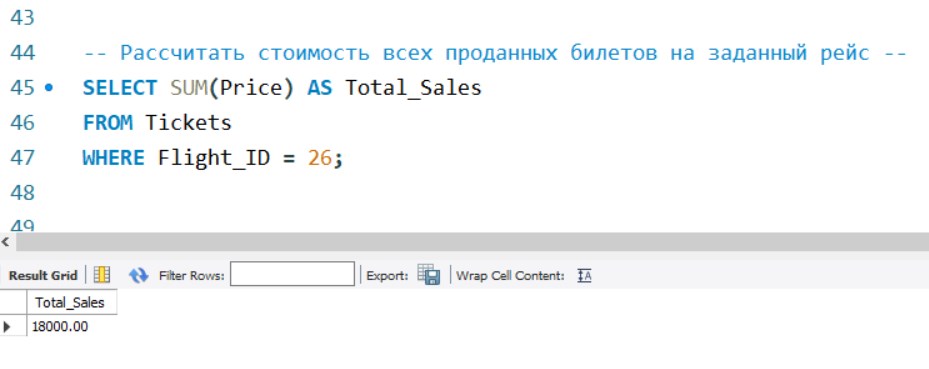


Рисунок 3.1.4 Запрос на расчёт стоимости всех проданных билетов на заданный рейс

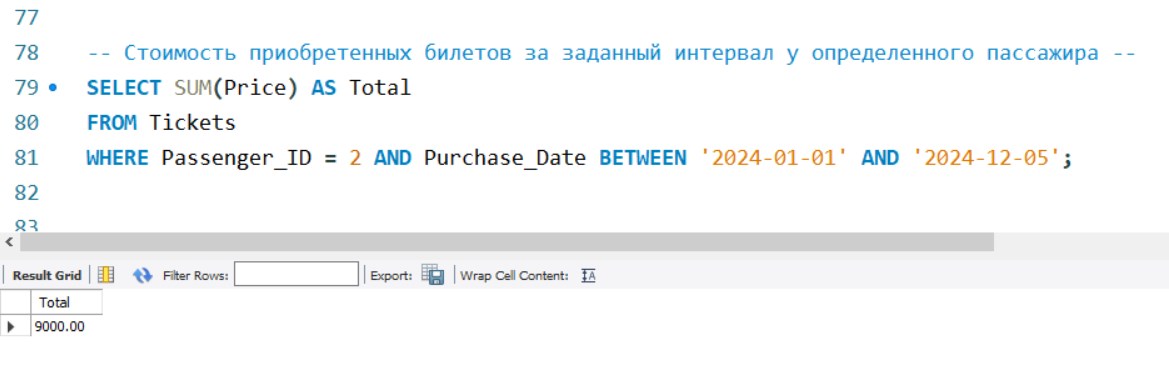


Рисунок 3.1.5 Запрос на стоимость приобретенных билетов за заданный интервал у определенного пассажира

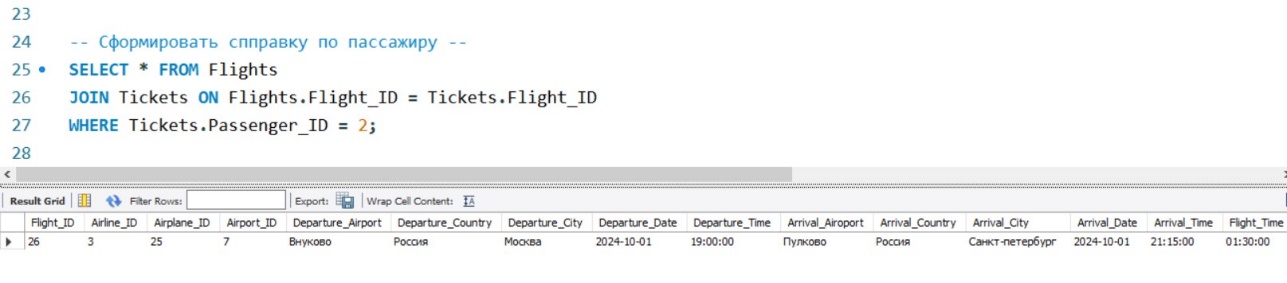


Рисунок 3.1.6 Запрос на сформирование справки по пассажиру



Рисунок 3.1.7 Запрос на отображения списка всех пассажиров заданного рейса

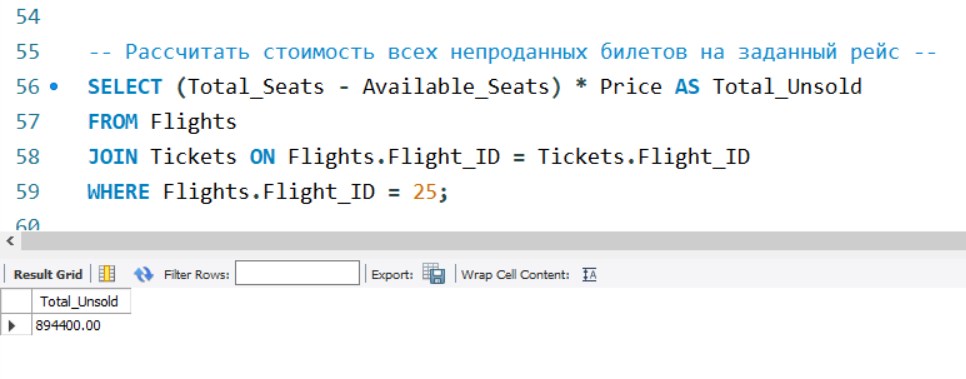


Рисунок 3.1.8 Запрос на расчёт стоимости всех непроданных билетов за заданный рейс

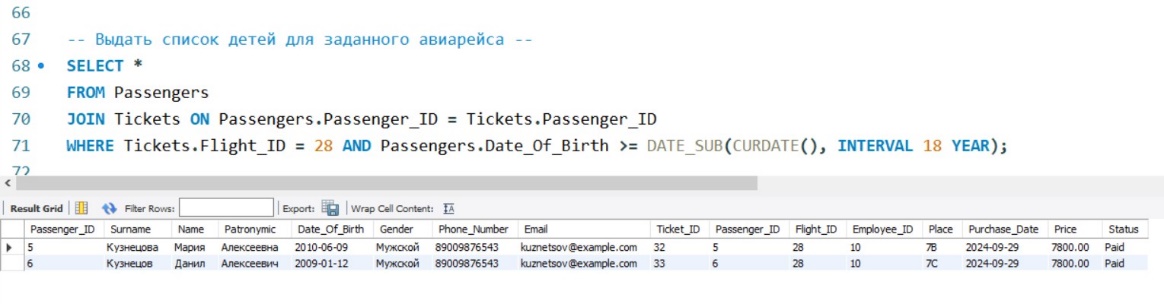


Рисунок 3.1.9 Запрос на список детей для заданного авиарейса

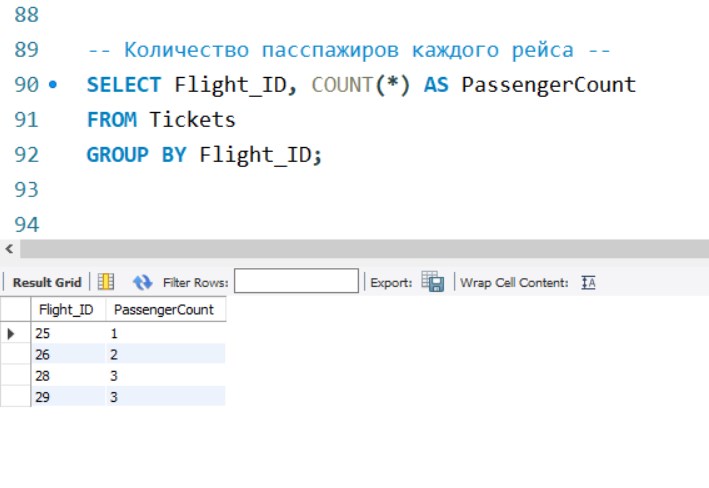


Рисунок 3.1.10 Запрос на количество пассажиров каждого авиарейса

# **3.2 Внедрение в безопасность SQL Server**

Внедрение в безопасность SQL Server — это процесс, направленный на защиту данных и ресурсов, хранящихся в базе данных SQL Server, от несанкционированного доступа, утечек и других угроз. Это включает в себя набор практик, технологий и политик, которые помогают обеспечить целостность, конфиденциальность и доступность данных. Вот основные аспекты внедрения безопасности в SQL Server:

* управление безопасностью уровня сервера,
* управление участниками уровня базы данных,
* управление разрешениями уровня базы данных,
* резервное копирование баз данных,
* шифрование данных баз данных.

# **3.3 Управление безопасностью уровня сервера**

Управление безопасностью — это процесс, который включает в себя создание, реализацию, мониторинг и улучшение систем и мер, направленных на защиту от угроз и рисков в различных областях.

Основные составляющие управления безопасностью включают:

1. **Оценка рисков**: Выявление и анализ потенциальных угроз, уязвимостей и потенциальных последствий для организации или системы.
2. **Разработка политик и процедур**: Создание четких инструкций и норм поведения, которые определяют, как действовать в различных ситуациях.
3. **Обучение и осведомленность**: Проведение регулярных тренингов для сотрудников по вопросам безопасности, чтобы они знали, как действовать в случае инцидента.
4. **Технические меры безопасности**: Использование технологий и инструментов, таких как системы контроля доступа, антивирусное программное обеспечение и другие средства защиты.
5. **Мониторинг и аудит**: Постоянное отслеживание системы безопасности и проведение регулярных проверок для оценки эффективности применения мер безопасности.
6. **Управление инцидентами и реагирование на чрезвычайные ситуации**: Разработка планов действий на случай инцидентов, а также обучение сотрудников действиям в экстренных ситуациях.

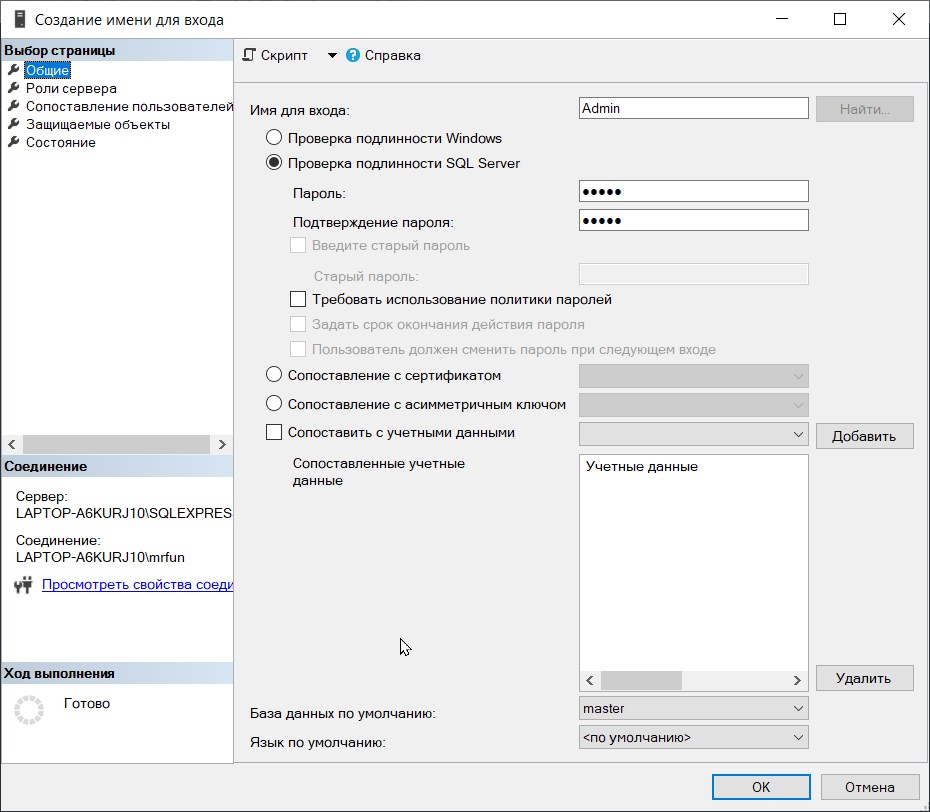


Рисунок 3.3.1 Создаём пользователя и устанавливаем пароль

# **3.4 Управление участниками уровня базы данных**

Управление участниками (или управление пользователями) — это процесс, связанный с контролем и администрированием пользователей в системе, приложении или организации. Это может включать в себя:

1. **Регистрация и аутентификация**: Процессы, позволяющие пользователям создавать учетные записи и входить в систему.
2. **Управление ролями и правами доступа**: Определение, какие действия могут выполнять пользователи в зависимости от их ролей (например, администратор, пользователь, гость).
3. **Мониторинг активности**: Отслеживание действий пользователей для обеспечения безопасности и соблюдения правил.
4. **Управление данными пользователей**: Хранение и обработка информации о пользователях, включая их профили, настройки и предпочтения.
5. **Поддержка и обслуживание**: Обеспечение помощи пользователям, решение их проблем и управление запросами.

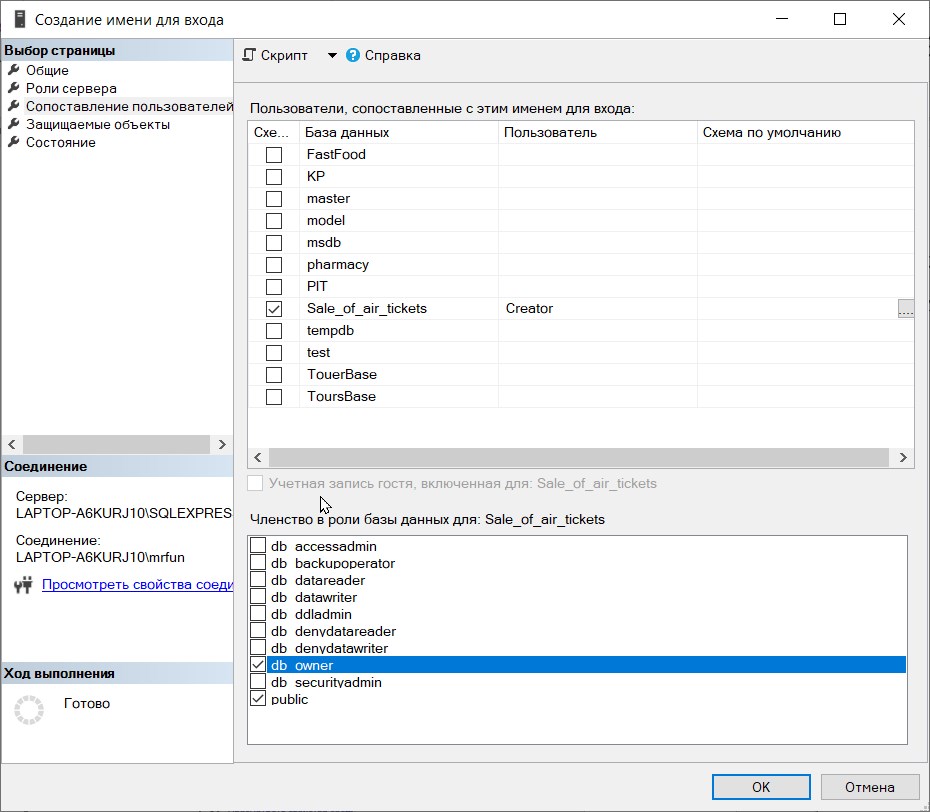


Рисунок 3.4.1 Выдаём пользователю членство в роли базы данных

# **3.5 Управление разрешениями уровня базы данных**

Управление разрешениями — это процесс контроля доступа к ресурсам и функциям в системе или приложении. Оно позволяет определять, кто может выполнять определенные действия, такие как чтение, запись, изменение или удаление данных.

Основные аспекты управления разрешениями включают:

1. **Аутентификация**: Проверка личности пользователя (например, через логин и пароль).
2. **Авторизация**: Определение прав доступа пользователя после его аутентификации.
3. **Роли и группы**: Создание ролей (например, администратор, пользователь, гость) и групп для упрощения управления разрешениями.
4. **Политики доступа**: Установление правил, которые определяют, какие действия разрешены или запрещены для различных пользователей или групп.
5. **Аудит и мониторинг**: Отслеживание действий пользователей для обеспечения безопасности и соответствия политике.

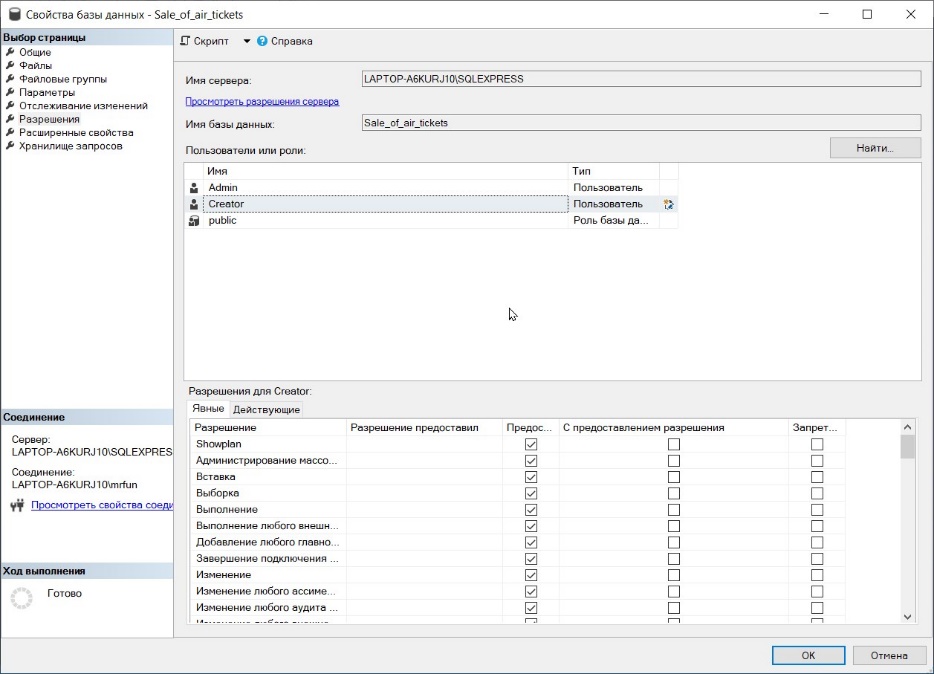


Рисунок 3.5.1 Выдаем доступ для определенного пользователя

# **3.6 Резервное копирование баз данных**

Резервное копирование (BackUp) — это процесс создания копии данных, которые хранятся на компьютере, сервере или любом другом устройстве хранения информации. Цель резервного копирования заключается в защите данных от потери, повреждения, случайного удаления, кибератак или других непредвиденных ситуаций.

Создание BackUp для базы данных Sale\_of\_air\_tickets в файл Sale\_of\_air\_tickets.bak.

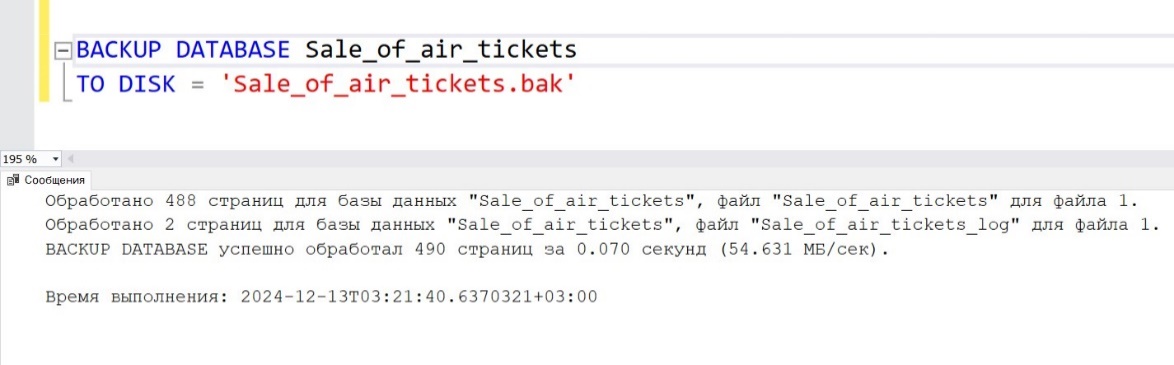


Рисунок 3.6.1 Код для создания файла резервного копирования

Созданный файл сохраняется в папке BackUp, установленной программы SQL Server Management Studio.

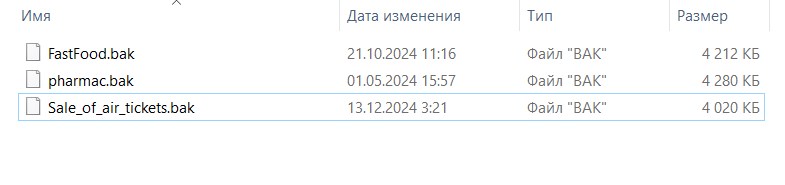


Рисунок 3.6.2 Файл резервного копирования

При помощи “RESTORE DATABASE” можем восстановить базу данных.

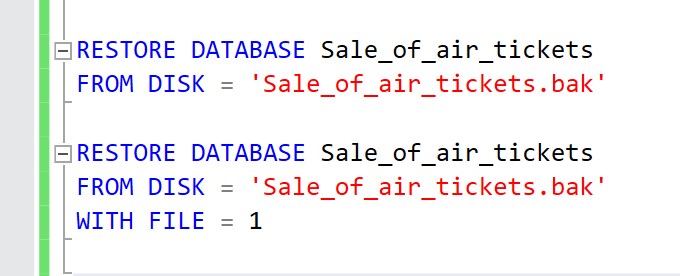
****

Рисунок 3.6.3 Код для восстановления базы данных

# **3.7 Шифрование данных баз данных**

Шифрование данных в баз данных — это процесс преобразования данных в неразборчивый формат, чтобы защитить их от несанкционированного доступа. Это важный аспект безопасности данных, особенно в условиях, когда данные могут быть украдены или неправомерно использованы.

Основные аспекты шифрования данных в базе данных:

1. **Типы шифрования**:
   * **Симметричное шифрование**: Использует один и тот же ключ для шифрования и расшифровки данных. Примеры алгоритмов: AES, DES.
   * **Асимметричное шифрование**: Использует пару ключей — открытый и закрытый. Примеры алгоритмов: RSA, ECC.
2. **Шифрование на уровне приложения**: Данные шифруются перед отправкой в базу данных. Это позволяет контролировать, какие данные шифруются и как.
3. **Шифрование на уровне базы данных**: База данных сама управляет шифрованием данных. Это может быть реализовано через встроенные функции шифрования, доступные в некоторых СУБД (например, Transparent Data Encryption в SQL Server и Oracle).
4. **Ключи шифрования**: Важно правильно управлять ключами шифрования, чтобы предотвратить их компрометацию. Это может включать использование аппаратных модулей безопасности (HSM) для хранения ключей.
5. **Производительность**: Шифрование может влиять на производительность базы данных, поэтому важно оценить компромисс между безопасностью и производительностью.
6. **Соответствие требованиям**: Шифрование данных может быть обязательным для соблюдения различных стандартов и регуляций, таких как GDPR, HIPAA и PCI DSS.

Напишем код для шифрования:

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'SecuryP5';

CREATE CERTIFICATE SecuryCertificate

WITH SUBJECT = 'SecurityCertificate';

CREATE SYMMETRIC KEY DataEncryptionKey

WITH ALGORITHM = AES\_256

ENCRYPTION BY CERTIFICATE SecuryCertificate;

OPEN SYMMETRIC KEY DataEncryptionKey

DECRYPTION BY CERTIFICATE SecuryCertificate;

CREATE TABLE UsersSys (

UserID INT IDENTITY(1,1) PRIMARY KEY,

Login VARCHAR(50),

EncryptedPassword VARBINARY(MAX)

);

OPEN SYMMETRIC KEY DataEncryptionKey

DECRYPTION BY CERTIFICATE SecuryCertificate;

INSERT INTO UsersSys (Login, EncryptedPassword)

VALUES

('Admin', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'AdP5')),

('Client', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'ClP5')),

('Manager', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'MgP5')),

('Creator', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'MgP5'));

CLOSE SYMMETRIC KEY DataEncryptionKey;

SELECT Login, EncryptedPassword

FROM dbo.UsersSys;

SELECT \* FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME = 'UsersSys';

INSERT INTO dbo.UsersSys (Login, EncryptedPassword)

VALUES

('Admin', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'AdP5')),

('Client', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'ClP5')),

('Manager', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'MgP5')),

('Creator', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'MgP5'));

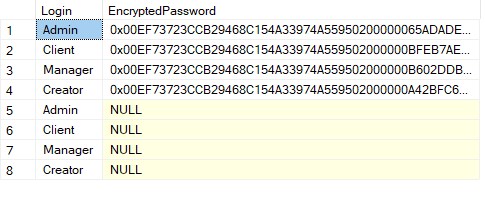


Рисунок 3.7.1 Результат шифрования

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсового проекта была разработана база данных для системы продажи авиабилетов, которая отвечает современным требованиям и обеспечивает эффективное управление данными. Основное внимание было уделено проектированию структуры базы данных, которая включает в себя ключевые сущности, такие как билеты, рейсы, самолёты, авиакомпании, пассажиры, аэропорты, сотрудники, должности.

В результате проведенной работы была достигнута цель проекта — создание функциональной и надежной базы данных, которая может быть использована в реальных условиях для автоматизации процессов продажи авиабилетов. Дальнейшие шаги могут включать интеграцию с веб-интерфейсом и мобильными приложениями, что позволит расширить функциональность системы и улучшить пользовательский опыт.

Полученная база данных соблюдает все требования технического задания, поставленные задачи были выполнены и цель достигнута.

Ссылка на репозиторий: [Yuras67/Course-project-DB](https://github.com/Yuras67/Course-project-DB)

# **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Microsoft SQL Server [Электронный ресурс] // Microsoft – Режим доступа: [Microsoft SQL Server](https://www.microsoft.com/en-us/sql-server) (дата обращения: 23.12.2024)
2. MS SQL [Электронный ресурс]// Skillbox – Режим доступа: [Введение в MS SQL Server: что это за СУБД и как она работает / Skillbox Media](https://skillbox.ru/media/code/baza-dannykh-ms-sql-server-chto-eto-zachem-nuzhna-kak-poyavilas-i-chem-khorosha/?ysclid=m64o7iph16882860437) (дата обращения: 14.12.2024)
3. SQL Server [Электронный ресурс] // Metanit – Режим доступа: [MS SQL Server 2022 и T-SQL (metanit.com)](https://metanit.com/sql/sqlserver/) (дата обращения: 18.12.2024)
4. Диаграмма ER-модель [Электронный ресурс] // WorldSkills – Режим доступа: <https://nationalteam.worldskills.ru/skills/proektirovanie-er-diagrammy/> (дата обращения: 20.12.2024)
5. Интерактивный тренажер по SQL [Электронный ресурс] // Stepik – Режим доступа: [Интерактивный тренажер по SQL – Stepik](https://stepik.org/course/63054/info) (дата обращения: 21.12.2024)
6. Работа с Базой Данных [Видео ресурс] // WorldSkills – Режим доступа: [Работа с базой данных в приложении: чтение, добавление, редактирование, удаление данных (часть 1) — Национальная сборная Worldskills Россия](https://nationalteam.worldskills.ru/skills/rabota-s-bazoy-dannykh-v-prilozhenii-chtenie-dobavlenie-redaktirovanie-udalenie-dannykh-chast-1/?ysclid=lw4p42n3g9933379553) (дата обращения: 26.12.2024)
7. Проектирование Entity Relationship-диаграмм [Электронный ресурс] // Habr – Режим доступа: [Учимся проектированию Entity Relationship-диаграмм / Хабр](https://habr.com/ru/articles/440556/) (дата обращения: 14.12.2024)