CREATE DATABASE Аэропорт;

GO

USE Аэропорт;

GO

-- Создание таблицы "Авиакомпании"

CREATE TABLE Авиакомпании (

id\_авиакомпании INT PRIMARY KEY IDENTITY(1, 1),

Название VARCHAR(100) NOT NULL,

Адрес VARCHAR(200) NOT NULL,

Телефон VARCHAR(20) NOT NULL,

Email VARCHAR(100)

);

GO

-- Создание таблицы "Аэропорты"

CREATE TABLE Аэропорты (

id\_аэропорта INT PRIMARY KEY IDENTITY(1, 1),

Название VARCHAR(100) NOT NULL,

Адрес VARCHAR(200) NOT NULL,

Город VARCHAR(100) NOT NULL,

Страна VARCHAR(100) NOT NULL

);

GO

-- Создание таблицы "Рейсы"

CREATE TABLE Рейсы (

id\_рейса INT PRIMARY KEY IDENTITY(1, 1),

id\_авиакомпании INT,

id\_аэропорта INT,

Время\_отправления DATETIME NOT NULL,

Время\_прибытия DATETIME NOT NULL,

FOREIGN KEY (id\_авиакомпании) REFERENCES Авиакомпании(id\_авиакомпании),

FOREIGN KEY (id\_аэропорта) REFERENCES Аэропорты(id\_аэропорта)

);

GO

-- Создание таблицы "Пассажиры"

CREATE TABLE Пассажиры (

id\_пассажира INT PRIMARY KEY IDENTITY(1, 1),

Имя VARCHAR(100) NOT NULL,

Фамилия VARCHAR(100) NOT NULL,

Паспорт VARCHAR(20) NOT NULL,

Телефон VARCHAR(20) NOT NULL,

Email VARCHAR(100)

);

GO

-- Создание таблицы "Билеты"

CREATE TABLE Билеты (

id\_билета INT PRIMARY KEY IDENTITY(1, 1),

id\_пассажира INT,

id\_рейса INT,

Дата\_покупки DATE NOT NULL,

FOREIGN KEY (id\_пассажира) REFERENCES Пассажиры(id\_пассажира),

FOREIGN KEY (id\_рейса) REFERENCES Рейсы(id\_рейса)

);

GO

USE Аэропорт;

GO

INSERT INTO Авиакомпании (Название, Адрес, Телефон, Email)

VALUES ('Аэрофлот', 'Москва, Россия', '+7 495 223-55-55', 'info@aeroflot.ru'),

('Lufthansa', 'Франкфурт, Германия', '+49 69 696-0', 'info@lufthansa.com'),

('Emirates', 'Дубай, ОАЭ', '+971 4 708 1111', 'info@emirates.com');

INSERT INTO Аэропорты (Название, Адрес, Город, Страна)

VALUES ('Шереметьево', 'Москва, Россия', 'Москва', 'Россия'),

('Хитроу', 'Лондон, Великобритания', 'Лондон', 'Великобритания'),

('Джона Ф. Кеннеди', 'Нью-Йорк, США', 'Нью-Йорк', 'США');

INSERT INTO Рейсы (id\_авиакомпании, id\_аэропорта, Время\_отправления, Время\_прибытия)

VALUES (1, 1, '2023-06-01 10:00:00', '2023-06-01 12:30:00'),

(2, 2, '2023-06-01 15:45:00', '2023-06-01 18:30:00'),

(3, 3, '2023-06-01 20:15:00', '2023-06-02 02:45:00');

INSERT INTO Пассажиры (Имя, Фамилия, Паспорт, Телефон, Email)

VALUES ('Иван', 'Иванов', 'AB123456', '+7 999 123-45-67', 'ivanov@example.com'),

('Анна', 'Смирнова', 'CD654321', '+7 999 987-65-43', 'smirnova@example.com'),

('Петр', 'Петров', 'EF987654', '+7 999 567-89-01', 'petrov@example.com');

INSERT INTO Билеты (id\_пассажира, id\_рейса, Дата\_покупки)

VALUES (1, 1, '2023-05-30'),

(2, 2, '2023-05-31'),

(3, 3, '2023-06-01');

USE Аэропорт;

GO

CREATE PROCEDURE Добавить\_Авиакомпанию

@Название VARCHAR(100),

@Адрес VARCHAR(200),

@Телефон VARCHAR(20),

@Email VARCHAR(100)

AS

BEGIN

INSERT INTO Авиакомпании (Название, Адрес, Телефон, Email)

VALUES (@Название, @Адрес, @Телефон, @Email);

END;

go

CREATE PROCEDURE Получить\_Рейсы\_По\_Дата

@Дата DATE

AS

BEGIN

SELECT \*

FROM Рейсы

WHERE CONVERT(DATE, Время\_отправления) = @Дата;

END;

go

CREATE PROCEDURE Удалить\_Пассажира\_С\_Билетами

@id\_пассажира INT

AS

BEGIN

BEGIN TRANSACTION;

DELETE FROM Билеты

WHERE id\_пассажира = @id\_пассажира;

DELETE FROM Пассажиры

WHERE id\_пассажира = @id\_пассажира;

COMMIT;

END;

Go

USE Аэропорт;

GO

CREATE VIEW Пассажиры\_с\_билетами AS

SELECT П.id\_пассажира, П.Имя, П.Фамилия, П.Паспорт, П.Телефон, П.Email,

Б.id\_билета, Б.Дата\_покупки, Б.id\_рейса

FROM Пассажиры П

INNER JOIN Билеты Б ON П.id\_пассажира = Б.id\_пассажира;

go

CREATE VIEW Пассажиры\_с\_рейсами AS

SELECT П.id\_пассажира, П.Имя, П.Фамилия, П.Паспорт, П.Телефон, П.Email,

Р.id\_рейса, Р.Время\_отправления, Р.Время\_прибытия

FROM Пассажиры П

INNER JOIN Билеты Б ON П.id\_пассажира = Б.id\_пассажира

INNER JOIN Рейсы Р ON Б.id\_рейса = Р.id\_рейса;

go

CREATE VIEW Авиакомпании\_с\_рейсами AS

SELECT А.Название AS Авиакомпания, А.Адрес, А.Телефон, А.Email,

Р.id\_рейса, Р.Время\_отправления, Р.Время\_прибытия

FROM Авиакомпании А

INNER JOIN Рейсы Р ON А.id\_авиакомпании = Р.id\_авиакомпании;

Go

USE Аэропорт;

GO

CREATE TRIGGER Удаление\_Пассажира

ON Пассажиры

AFTER DELETE

AS

BEGIN

DELETE FROM Билеты

WHERE id\_пассажира IN (SELECT id\_пассажира FROM deleted);

END;

go

CREATE TRIGGER Проверка\_Даты\_Рейса

ON Рейсы

INSTEAD OF INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT \*

FROM inserted

WHERE Время\_отправления > Время\_прибытия

)

BEGIN

RAISERROR('Время отправления должно быть раньше времени прибытия.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END;

-- Если проверка прошла успешно, выполните операцию INSERT или UPDATE

INSERT INTO Рейсы (id\_авиакомпании, id\_аэропорта, Время\_отправления, Время\_прибытия)

SELECT id\_авиакомпании, id\_аэропорта, Время\_отправления, Время\_прибытия

FROM inserted;

END;

go

CREATE TRIGGER Обновление\_Адреса\_Аэропорта

ON Аэропорты

AFTER UPDATE

AS

BEGIN

UPDATE Аэропорты

SET Город = inserted.Город

FROM Аэропорты

INNER JOIN inserted ON Аэропорты.id\_аэропорта = inserted.id\_аэропорта;

END;

Go

USE Аэропорт;

GO

CREATE FUNCTION Получить\_Количество\_Рейсов\_ПоАвиакомпании

(@id\_авиакомпании INT)

RETURNS INT

AS

BEGIN

DECLARE @количество\_рейсов INT;

SELECT @количество\_рейсов = COUNT(\*)

FROM Рейсы

WHERE id\_авиакомпании = @id\_авиакомпании;

RETURN @количество\_рейсов;

END;

go

CREATE FUNCTION Получить\_Полное\_Имя\_Пассажира

(@id\_пассажира INT)

RETURNS VARCHAR(200)

AS

BEGIN

DECLARE @полное\_имя VARCHAR(200);

SELECT @полное\_имя = Имя + ' ' + Фамилия

FROM Пассажиры

WHERE id\_пассажира = @id\_пассажира;

RETURN @полное\_имя;

END;

go

CREATE FUNCTION Получить\_Список\_Аэропортов\_ПоСтране

(@Страна VARCHAR(100))

RETURNS TABLE

AS

RETURN

SELECT Название, Адрес, Город

FROM Аэропорты

WHERE Страна = @Страна;

go

1. **Какие современные инструментальные средства проектирования схемы базы данных используют?**

Существует несколько современных инструментальных средств для проектирования схемы базы данных. Вот некоторые из них:

1. ER-диаграммы: ER-диаграммы (диаграммы сущность-связь) являются одним из наиболее распространенных инструментов для моделирования баз данных. Они позволяют визуально представить сущности (таблицы), их атрибуты и связи между ними.

2. UML-диаграммы: Unified Modeling Language (UML) - это стандартный язык моделирования, который также может быть использован для проектирования баз данных. UML-диаграммы классов и диаграммы объектов могут быть применены для описания сущностей и их связей в базе данных.

3. CASE-средства: CASE (Computer-Aided Software Engineering) - это программные средства, которые предоставляют набор инструментов для анализа, проектирования и разработки систем, включая базы данных. Они обычно предлагают функциональности ER-моделирования, автоматической генерации кода, документирования и других задач, связанных с проектированием баз данных.

4. Программное обеспечение для моделирования данных: Существуют специализированные программные инструменты для моделирования данных, такие как MySQL Workbench, Microsoft Visio, Oracle SQL Developer Data Modeler и другие. Они предоставляют возможности для создания ER-диаграмм, генерации сценариев создания таблиц, управления связями и других аспектов проектирования баз данных.

5. Онлайн-сервисы: В последнее время стали популярны онлайн-сервисы, которые позволяют создавать и моделировать базы данных через веб-интерфейс. Примеры таких сервисов включают dbdiagram.io, draw.io и Lucidchart.