-- Создание базы данных "Туристическое агентство"

CREATE DATABASE ТуристическоеАгентство;

GO

-- Использование базы данных "Туристическое агентство"

USE ТуристическоеАгентство;

GO

-- Создание таблицы "Сотрудники"

CREATE TABLE Сотрудники (

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

Имя VARCHAR(100) NOT NULL,

Фамилия VARCHAR(100) NOT NULL,

Должность VARCHAR(50) NOT NULL,

Зарплата INT

);

GO

-- Создание таблицы "Туры"

CREATE TABLE Туры (

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

Название VARCHAR(100) NOT NULL,

Описание VARCHAR(500),

Страна VARCHAR(100),

Длительность INT,

Стоимость INT,

Дата\_начала DATE,

Дата\_окончания DATE,

id\_сотрудника INT,

FOREIGN KEY (id\_сотрудника) REFERENCES Сотрудники(id\_сотрудника)

);

GO

-- Создание таблицы "Клиенты"

CREATE TABLE Клиенты (

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

Имя VARCHAR(100) NOT NULL,

Фамилия VARCHAR(100) NOT NULL,

Адрес VARCHAR(100),

Телефон VARCHAR(15),

Email VARCHAR(50)

);

GO

-- Создание таблицы "Бронирования"

CREATE TABLE Бронирования (

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

id\_тура INT,

id\_клиента INT,

Дата\_бронирования DATE,

FOREIGN KEY (id\_тура) REFERENCES Туры(id\_тура),

FOREIGN KEY (id\_клиента) REFERENCES Клиенты(id\_клиента)

);

GO

-- Создание таблицы "Отзывы"

CREATE TABLE Отзывы (

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

id\_тура INT,

id\_клиента INT,

Текст VARCHAR(500),

Рейтинг INT,

FOREIGN KEY (id\_тура) REFERENCES Туры(id\_тура),

FOREIGN KEY (id\_клиента) REFERENCES Клиенты(id\_клиента)

);

GO

-- Заполнение таблицы "Туры"

INSERT INTO Туры (Название, Описание, Страна, Длительность, Стоимость, Дата\_начала, Дата\_окончания)

VALUES

('Путешествие в Таиланд', 'Отдых на пляжах Таиланда', 'Таиланд', 10, 2000, '2023-07-01', '2023-07-10'),

('Экскурсия в Париж', 'Знакомство с достопримечательностями Парижа', 'Франция', 5, 1500, '2023-08-15', '2023-08-20'),

('Горнолыжный отдых в Швейцарии', 'Катание на горных лыжах в Швейцарии', 'Швейцария', 7, 2500, '2023-12-05', '2023-12-12'),

('Поход в горы', 'Активный отдых на природе', 'Россия', 3, 800, '2023-09-10', '2023-09-13'),

('Круиз по Средиземному морю', 'Путешествие на лайнере по курортам Средиземноморья', 'Испания', 14, 5000, '2023-10-20', '2023-11-03');

-- Заполнение таблицы "Клиенты"

INSERT INTO Клиенты (Имя, Фамилия, Адрес, Телефон, Email)

VALUES

('Иван', 'Иванов', 'Москва', '1234567890', 'ivan@example.com'),

('Анна', 'Петрова', 'Санкт-Петербург', '9876543210', 'anna@example.com'),

('Петр', 'Сидоров', 'Екатеринбург', '5555555555', 'petr@example.com'),

('Елена', 'Смирнова', 'Новосибирск', '9998887776', 'elena@example.com'),

('Дмитрий', 'Козлов', 'Казань', '4443332221', 'dmitry@example.com');

-- Заполнение таблицы "Бронирования"

INSERT INTO Бронирования (id\_тура, id\_клиента, Дата\_бронирования)

VALUES

(1, 2, '2023-06-15'),

(3, 1, '2023-07-20'),

(2, 4, '2023-08-01'),

(5, 3, '2023-09-05'),

(4, 5, '2023-10-10');

-- Заполнение таблицы "Сотрудники"

INSERT INTO Сотрудники (Имя, Фамилия, Должность, Зарплата)

VALUES

('Алексей', 'Смирнов', 'Менеджер', 30000),

('Екатерина', 'Иванова', 'Агент по бронированию', 25000),

('Михаил', 'Петров', 'Гид', 20000),

('Ольга', 'Сидорова', 'Администратор', 28000),

('Иван', 'Козлов', 'Директор', 50000);

-- Заполнение таблицы "Отзывы"

INSERT INTO Отзывы (id\_тура, id\_клиента, Текст, Рейтинг)

VALUES

(1, 3, 'Отличный отдых! Все понравилось.', 5),

(2, 5, 'Париж - прекрасный город! Рекомендую!', 4),

(3, 2, 'Незабываемые впечатления от горнолыжного отдыха.', 4),

(4, 1, 'Поход в горы был интересным и активным.', 4),

(5, 4, 'Круиз превзошел все ожидания. Спасибо!', 5);

CREATE VIEW ТурыПоСтранам AS

SELECT Страна, COUNT(\*) AS Количество\_туров

FROM Туры

GROUP BY Страна;

CREATE VIEW БронированияКлиентов AS

SELECT Бронирования.id\_бронирования, Клиенты.Имя, Клиенты.Фамилия, Туры.Название, Бронирования.Дата\_бронирования

FROM Бронирования

JOIN Клиенты ON Бронирования.id\_клиента = Клиенты.id\_клиента

JOIN Туры ON Бронирования.id\_тура = Туры.id\_тура;

CREATE VIEW ОтзывыПоТурам AS

SELECT Туры.id\_тура, Туры.Название, COUNT(Отзывы.id\_отзыва) AS Количество\_отзывов, AVG(Отзывы.Рейтинг) AS Средний\_рейтинг

FROM Туры

LEFT JOIN Отзывы ON Туры.id\_тура = Отзывы.id\_тура

GROUP BY Туры.id\_тура, Туры.Название;

CREATE PROCEDURE ДобавитьКлиента

@Имя VARCHAR(100),

@Фамилия VARCHAR(100),

@Адрес VARCHAR(100),

@Телефон VARCHAR(15),

@Email VARCHAR(50)

AS

BEGIN

INSERT INTO Клиенты (Имя, Фамилия, Адрес, Телефон, Email)

VALUES (@Имя, @Фамилия, @Адрес, @Телефон, @Email);

END;

CREATE PROCEDURE УдалитьТур

@id\_тура INT

AS

BEGIN

DELETE FROM Туры

WHERE id\_тура = @id\_тура;

END;

CREATE PROCEDURE ОбновитьЗарплатуСотрудника

@id\_сотрудника INT,

@Зарплата INT

AS

BEGIN

UPDATE Сотрудники

SET Зарплата = @Зарплата

WHERE id\_сотрудника = @id\_сотрудника;

END;

CREATE TRIGGER ПередВставкойТура

ON Туры

INSTEAD OF INSERT

AS

BEGIN

IF (SELECT COUNT(\*) FROM inserted WHERE Стоимость > 10000) > 0

BEGIN

RAISERROR ('Стоимость тура не может превышать 10000.', 16, 1);

ROLLBACK TRANSACTION;

END;

ELSE

BEGIN

INSERT INTO Туры (Название, Описание, Страна, Длительность, Стоимость, Дата\_начала, Дата\_окончания)

SELECT Название, Описание, Страна, Длительность, Стоимость, Дата\_начала, Дата\_окончания

FROM inserted;

END;

END;

CREATE TRIGGER ПослеУдаленияКлиента

ON Клиенты

AFTER DELETE

AS

BEGIN

DELETE FROM Бронирования

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

END;

CREATE TRIGGER ПослеВставкиТура

ON Туры

AFTER INSERT

AS

BEGIN

-- Вывод сообщения после вставки тура

PRINT 'Новый тур был успешно добавлен.';

END;

CREATE FUNCTION ПолучитьТурыПоСтране(@страна VARCHAR(100))

RETURNS TABLE

AS

RETURN (

SELECT Туры.\*

FROM Туры

WHERE Туры.Страна = @страна

);

CREATE FUNCTION ПолучитьБронированияКлиента(@id\_клиента INT)

RETURNS TABLE

AS

RETURN (

SELECT Бронирования.\*

FROM Бронирования

WHERE Бронирования.id\_клиента = @id\_клиента

);

CREATE FUNCTION ПолучитьОтзывыПоТуру(@id\_тура INT)

RETURNS TABLE

AS

RETURN (

SELECT Отзывы.\*

FROM Отзывы

WHERE Отзывы.id\_тура = @id\_тура

);

1. **Мониторинг работы сервера**

Мониторинг работы сервера является важной задачей, которая позволяет отслеживать его состояние и производительность. Вот некоторые основные аспекты мониторинга работы сервера:

1. Доступность сервера: Мониторинг должен включать проверку доступности сервера. Это может включать проверку доступности сетевого соединения, служб, портов и других ресурсов сервера.

2. Использование ресурсов: Мониторинг должен отслеживать использование ресурсов сервера, таких как процессор, память, дисковое пространство и сетевой трафик.

3. Производительность: Мониторинг должен измерять производительность сервера, включая задержку ответа, скорость передачи данных и другие метрики производительности.

4. Журналы и события: Мониторинг должен отслеживать журналы событий сервера и регистрировать ошибки, предупреждения и другие события, которые могут указывать на проблемы или потенциальные угрозы безопасности. Это помогает операторам мониторинга реагировать на проблемы и предотвращать серьезные нарушения работы сервера.

5. Уведомления: Мониторинг должен предоставлять возможность настройки уведомлений, которые информируют операторов мониторинга о проблемах и событиях, требующих их внимания. Уведомления могут быть отправлены по электронной почте, SMS или другими средствами связи.

6. История и анализ: Мониторинг должен сохранять исторические данные о работе сервера, чтобы можно было проанализировать его производительность и выявить тренды или повторяющиеся проблемы. Это помогает в определении оптимальных настроек сервера и прогнозировании будущих потребностей.