-- Создание базы данных "Расписание занятий"

CREATE DATABASE Расписание\_занятий;

GO

-- Использование базы данных "Расписание занятий"

USE Расписание\_занятий;

GO

-- Создание таблицы "Преподаватели"

CREATE TABLE Преподаватели (

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

Имя VARCHAR(100) NOT NULL,

Фамилия VARCHAR(100) NOT NULL,

Должность VARCHAR(50) NOT NULL

);

-- Создание таблицы "Предметы"

CREATE TABLE Предметы (

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

Название VARCHAR(100) NOT NULL,

Описание TEXT

);

-- Создание таблицы "Аудитории"

CREATE TABLE Аудитории (

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

Название VARCHAR(100) NOT NULL,

Корпус VARCHAR(50) NOT NULL

);

-- Создание таблицы "Группы"

CREATE TABLE Группы (

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

Название VARCHAR(100) NOT NULL,

Курс INT NOT NULL

);

-- Создание таблицы "Расписание"

CREATE TABLE Расписание (

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

id\_предмета INT,

id\_преподавателя INT,

id\_аудитории INT,

id\_группы INT,

Дата\_занятия DATE NOT NULL,

Время\_начала TIME NOT NULL,

Время\_окончания TIME NOT NULL,

FOREIGN KEY (id\_предмета) REFERENCES Предметы(id\_предмета),

FOREIGN KEY (id\_преподавателя) REFERENCES Преподаватели(id\_преподавателя),

FOREIGN KEY (id\_аудитории) REFERENCES Аудитории(id\_аудитории),

FOREIGN KEY (id\_группы) REFERENCES Группы(id\_группы)

);

-- Заполнение таблицы "Преподаватели"

INSERT INTO Преподаватели (Имя, Фамилия, Должность)

VALUES ('Иван', 'Иванов', 'Профессор'),

('Петр', 'Петров', 'Доцент'),

('Анна', 'Сидорова', 'Ассистент');

-- Заполнение таблицы "Предметы"

INSERT INTO Предметы (Название, Описание)

VALUES ('Математика', 'Основы алгебры и геометрии'),

('Физика', 'Классическая механика и электродинамика'),

('История', 'История России');

-- Заполнение таблицы "Аудитории"

INSERT INTO Аудитории (Название, Корпус)

VALUES ('Аудитория 101', 'Корпус А'),

('Аудитория 202', 'Корпус Б'),

('Аудитория 303', 'Корпус В');

-- Заполнение таблицы "Группы"

INSERT INTO Группы (Название, Курс)

VALUES ('Группа А', 1),

('Группа Б', 2),

('Группа В', 3);

-- Заполнение таблицы "Расписание"

INSERT INTO Расписание (id\_предмета, id\_преподавателя, id\_аудитории, id\_группы, Дата\_занятия, Время\_начала, Время\_окончания)

VALUES (1, 1, 1, 1, '2023-06-01', '08:00:00', '10:00:00'),

(2, 2, 2, 2, '2023-06-02', '10:30:00', '12:30:00'),

(3, 3, 3, 3, '2023-06-03', '13:00:00', '15:00:00');

CREATE VIEW РасписаниеПоAудиториям AS

SELECT Р.id\_расписания, Р.Дата\_занятия, Р.Время\_начала, Р.Время\_окончания, А.Название AS Аудитория, П.Название AS Предмет, Пр.Фамилия + ' ' + Пр.Имя AS Преподаватель, Г.Название AS Группа

FROM Расписание Р

JOIN Аудитории А ON Р.id\_аудитории = А.id\_аудитории

JOIN Предметы П ON Р.id\_предмета = П.id\_предмета

JOIN Преподаватели Пр ON Р.id\_преподавателя = Пр.id\_преподавателя

JOIN Группы Г ON Р.id\_группы = Г.id\_группы;

CREATE VIEW ПреподавателиПоПредметам AS

SELECT Пр.Имя AS Имя\_преподавателя, Пр.Фамилия AS Фамилия\_преподавателя, П.Название AS Предмет

FROM Преподаватели Пр

JOIN Расписание Р ON Пр.id\_преподавателя = Р.id\_преподавателя

JOIN Предметы П ON Р.id\_предмета = П.id\_предмета;

CREATE VIEW ЗанятияПоДатам AS

SELECT Р.Дата\_занятия, П.Название AS Предмет, Г.Название AS Группа, Пр.Имя AS Имя\_преподавателя, Пр.Фамилия AS Фамилия\_преподавателя, А.Название AS Аудитория

FROM Расписание Р

JOIN Предметы П ON Р.id\_предмета = П.id\_предмета

JOIN Группы Г ON Р.id\_группы = Г.id\_группы

JOIN Преподаватели Пр ON Р.id\_преподавателя = Пр.id\_преподавателя

JOIN Аудитории А ON Р.id\_аудитории = А.id\_аудитории;

CREATE PROCEDURE ДобавитьПреподавателя

@Имя VARCHAR(100),

@Фамилия VARCHAR(100),

@Должность VARCHAR(50)

AS

BEGIN

INSERT INTO Преподаватели (Имя, Фамилия, Должность)

VALUES (@Имя, @Фамилия, @Должность)

END;

CREATE PROCEDURE УдалитьГруппу

@id\_группы INT

AS

BEGIN

DELETE FROM Группы WHERE id\_группы = @id\_группы

END;

CREATE PROCEDURE ОбновитьВремяЗанятия

@id\_расписания INT,

@Время\_начала TIME,

@Время\_окончания TIME

AS

BEGIN

UPDATE Расписание

SET Время\_начала = @Время\_начала,

Время\_окончания = @Время\_окончания

WHERE id\_расписания = @id\_расписания

END;

CREATE TRIGGER ПроверкаЗаписиПреподавателя

ON Преподаватели

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT \* FROM Расписание WHERE id\_преподавателя IN (SELECT id\_преподавателя FROM inserted))

BEGIN

RAISERROR('Преподаватель имеет связанные записи в расписании. Удаление или изменение невозможно.', 16, 1)

ROLLBACK TRANSACTION

END

END;

CREATE TRIGGER АвтоВыставлениеДатыЗанятия

ON Расписание

INSTEAD OF INSERT

AS

BEGIN

INSERT INTO Расписание (id\_предмета, id\_преподавателя, id\_аудитории, id\_группы, Дата\_занятия, Время\_начала, Время\_окончания)

SELECT

i.id\_предмета,

i.id\_преподавателя,

i.id\_аудитории,

i.id\_группы,

GETDATE(),

i.Время\_начала,

i.Время\_окончания

FROM inserted AS i

END;

CREATE TRIGGER АвтоУдалениеЗанятий

ON Группы

AFTER DELETE

AS

BEGIN

DELETE FROM Расписание

WHERE id\_группы IN (SELECT id\_группы FROM deleted)

END;

CREATE FUNCTION ПолучитьЗанятияПоДате(@дата DATE)

RETURNS TABLE

AS

RETURN (

SELECT Расписание.\*, Преподаватели.Имя AS ИмяПреподавателя, Преподаватели.Фамилия AS ФамилияПреподавателя

FROM Расписание

INNER JOIN Преподаватели ON Расписание.id\_преподавателя = Преподаватели.id\_преподавателя

WHERE Расписание.Дата\_занятия = @дата

);

CREATE FUNCTION ПолучитьЗанятияПоПредмету(@id\_предмета INT)

RETURNS TABLE

AS

RETURN (

SELECT Расписание.\*, Предметы.Название AS НазваниеПредмета, Предметы.Описание

FROM Расписание

INNER JOIN Предметы ON Расписание.id\_предмета = Предметы.id\_предмета

WHERE Расписание.id\_предмета = @id\_предмета

);

CREATE FUNCTION ПолучитьЗанятияПоГруппе(@id\_группы INT)

RETURNS TABLE

AS

RETURN (

SELECT Расписание.\*, Группы.Название AS НазваниеГруппы, Группы.Курс

FROM Расписание

INNER JOIN Группы ON Расписание.id\_группы = Группы.id\_группы

WHERE Расписание.id\_группы = @id\_группы

);

1. **Резервное копирование баз данных. Восстановление баз данных**

Резервное копирование баз данных является важной практикой для обеспечения безопасности данных. Вот основные шаги процедуры резервного копирования баз данных:

1. Определение стратегии резервного копирования: необходимо определить, какие базы данных требуют резервного копирования и с какой периодичностью.

2. Выбор типа резервного копирования: существуют различные типы резервного копирования, такие как полное, дифференциальное или инкрементальное.

3. Планирование расписания резервного копирования: Установите расписание для выполнения резервного копирования.

4. Выполнение резервного копирования: запустите процесс резервного копирования для выбранных баз данных.

5. Проверка целостности резервной копии: после завершения резервного копирования необходимо проверить целостность и доступность копии.

Восстановление баз данных осуществляется в случае сбоев, потери данных или других непредвиденных ситуаций. Основные шаги процедуры восстановления баз данных включают:

1. Определение источника восстановления: Укажите, откуда будет производиться восстановление данных.

2. Выбор метода восстановления: В зависимости от доступных данных выберите метод восстановления.

3. Выполнение восстановления: запустите процесс восстановления баз данных в соответствии с выбранным методом.

4. Проверка восстановленных данных: после завершения восстановления проверьте целостность и доступность восстановленных данных. Убедитесь, что данные восстановлены корректно и система готова к использованию.