

Заявка

```
CREATE DATABASE SchoolCascadeDemo;
GO
USE SchoolCascadeDemo;
GO

-- Таблица: Ученици
CREATE TABLE Students (
    student_id INT PRIMARY KEY IDENTITY(1,1),
    full_name NVARCHAR(100) NOT NULL,
    class NVARCHAR(20)
);

-- Таблица: Курсове
CREATE TABLE Courses (
    course_id INT PRIMARY KEY IDENTITY(1,1),
    course_name NVARCHAR(50) NOT NULL
);

-- Таблица: Записвания на ученици (M:N)
CREATE TABLE StudentsCourses (
    student_id INT NOT NULL,
    course_id INT NOT NULL,
    date_enrolled DATE DEFAULT GETDATE(),
    CONSTRAINT PK_StuCou PRIMARY KEY (student_id, course_id),
    CONSTRAINT FK_StudentsCourses_Students FOREIGN KEY (student_id)
        REFERENCES Students(student_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT FK_StudentsCourses_Courses FOREIGN KEY (course_id)
        REFERENCES Courses(course_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- Таблица: Оценки
CREATE TABLE Grades (
    grade_id INT PRIMARY KEY IDENTITY(1,1),
    student_id INT NULL,
    course_id INT NULL,
    grade DECIMAL(3,2),
    CONSTRAINT FK_Grades_Students FOREIGN KEY (student_id)
        REFERENCES Students(student_id)
        ON DELETE SET NULL,
    CONSTRAINT FK_Grades_Courses FOREIGN KEY (course_id)
        REFERENCES Courses(course_id)
        ON DELETE NO ACTION
);
GO

INSERT INTO Students (full_name, class) VALUES
('Ivan Ivanov', '11A'),
('Maria Petrova', '11B'),
('Georgi Kolev', '12A');

INSERT INTO Courses (course_name) VALUES
```

```
('Databases'),
('Programming'),
('Mathematics');

INSERT INTO StudentsCourses (student_id, course_id) VALUES
(1,1), (1,2), (2,1), (2,3), (3,2);

INSERT INTO Grades (student_id, course_id, grade) VALUES
(1,1,5.20),
(1,2,4.80),
(2,1,5.50),
(2,3,5.00),
(3,2,5.80);
```

Задача 1. Изтрий ученика *Ivan Ivanov* от таблицата **Students**.

Наблюдавай ефекта върху **StudentsCourses** (трябва да се изтрият) и **Grades** (student_id става NULL).

Задача 2. Опитай да изтриеш курса *Databases*

SQL Server трябва да откаже изтриването, защото има NO ACTION връзка с таблица **Grades**.

Задача 3. Изтрий всички записи от **StudentsCourses** и виж какво се случва с останалите таблици.

Проверка на каскадата и целостта.

Задача 4. Добави нов ученик *Petar Hristov* и запиши го в два курса. След това изтрий курса *Programming*. Наблюдавай кои записи се изтриват каскадно.

Задача 5. Опитай да изтриеш *Mathematics*.

Увери се, че връзката в **StudentsCourses** също се актуализира.

Задача 6. Промени student_id на *Maria Petrova* от 2 на 22 с ON UPDATE CASCADE

Увери се, че връзката в **StudentsCourses** също се актуализира.

Задача 7. Промени ON DELETE NO ACTION в **Grades** за курсовете на ON DELETE SET NULL.

Изтрий курс и наблюдавай разликата.

Задача 8. Вмъкни нов ученик и курс, направи запис в StudentsCourses и Grades, после изтрий ученика. Демонстрирай как се прилага CASCADE и SET NULL едновременно.

Задача 9. Изпълните следната задачка :

```
SELECT  
    Students.full_name,  
    Grades.grade  
FROM Students  
JOIN Grades  
    ON Students.student_id = Grades.student_id;
```

```
SELECT  
    Students.full_name,  
    Grades.grade
```

Това казва на SQL Server **кои колони искаме да се покажат в резултата.**

Избираме три полета:

- Students.full_name → името на ученика
- Grades.subject → предметът
- Grades.grade → оценката

Забележи, че пишем Students. и Grades. пред имената на колоните, защото и двете таблици имат student_id и SQL трябва да знае от коя таблица взема данни.

```
FROM Students
```

Оказваме, че **основната таблица** (първата, от която ще тръгнем) е Students.

SQL ще започне с нея и ще опита да намери свързани данни в другите таблици.

```
JOIN Grades  
    ON Students.student_id = Grades.student_id;
```

Това е **INNER JOIN** (по подразбиране).

Означава:

→ Вземи **всички ученици, за които има съвпадение** в таблица Grades, и са свързани чрез колоната student_id.

ON Students.student_id = Grades.student_id

е условието, по което SQL съединява редовете от двете таблици.

```
SELECT  
    Students.full_name,  
    Courses.course_name,  
    Grades.grade  
FROM Students  
JOIN Grades  
    ON Students.student_id = Grades.student_id  
JOIN Courses  
    ON Grades.course_id = Courses.course_id;
```

Задача 10. Обясни разликата между CASCADE, SET NULL и NO ACTION на база резултатите, които видя.

Обобщение на поведението.