

## #17 Разработка триггерных решений

### Зачем делать эту лабораторную работу?

1. Чтобы научиться выполнять комплексное обеспечение целостности данных для разработанной базы данных.
2. Чтобы освоить разработку сложных триггеров.
3. Чтобы научиться тестировать разработанные триггерные решения.
4. Чтобы научиться оптимальным образом проектировать триггеры, не ухудшая производительность базы данных.

### Что нужно делать?

1. Создать базу данных.

Создайте таблицы с атрибутами согласно варианту задания. Обеспечьте базовые ограничения целостности данных.

Для примера можно взять базу данных для планирования «Ночи кино» в сети кинотеатров следующей структуры Кинозалы (номер кинозала, название, количество мест), Сеансы (номер сеанса, номер кинозала, название фильма, начало сеанса, конец сеанса), Билеты (номер билета, номер сеанса, место), Предпочтения зрителей (код зрителя, название фильма), Планировщик посещений (код зрителя, дата и время посещения, номер сеанса, номер порядка посещения). Код создания таблиц будет выглядеть так:

```
CREATE TABLE Кинозалы (  
    номер_кинозала INT PRIMARY KEY,  
    название NVARCHAR(100) NOT NULL,  
    количество_мест INT NOT NULL CHECK (количество_мест > 0)  
);  
  
CREATE TABLE Сеансы (  
    номер_сеанса INT PRIMARY KEY,  
    номер_кинозала INT NOT NULL,  
    название_фильма NVARCHAR(100) NOT NULL,  
    начало_сеанса DATETIME NOT NULL,  
    конец_сеанса DATETIME NOT NULL,  
    FOREIGN KEY (номер_кинозала) REFERENCES Кинозалы (номер_кинозала)  
);  
  
CREATE TABLE Билеты (  
    номер_билета INT PRIMARY KEY,  
    номер_сеанса INT NOT NULL,  
    место INT NOT NULL CHECK (место > 0),  
    FOREIGN KEY (номер_сеанса) REFERENCES Сеансы (номер_сеанса)  
);  
  
CREATE TABLE Предпочтения_зрителей (  
    код_зрителя INT NOT NULL,  
    название_фильма NVARCHAR(100) NOT NULL,  
    PRIMARY KEY (код_зрителя, название_фильма)  
);  
GO
```

```

CREATE TABLE Планировщик_посещений (
    код_зрителя INT,
    дата_и_время_посещения DATETIME NOT NULL,
    номер_сеанса INT NOT NULL,
    номер_порядка_посещения INT NOT NULL,
    PRIMARY KEY (код_зрителя, дата_и_время_посещения, номер_порядка_посещения),
    FOREIGN KEY (номер_сеанса) REFERENCES Сеансы (номер_сеанса)
)

```

## 2. Разработать триггер.

Реализуйте триггер согласно варианту задания. В качестве примера для разработанной базы данных условие ограничений целостности может выглядеть так: разработать триггер, который автоматически формирует план посещения сеансов для зрителей, учитывая их предпочтения. План должен формироваться следующим образом: в план первыми включаются сеансы с фильмами из списка предпочтений зрителя, между сеансами должен быть предусмотрен минимум 30-минутный перерыв. Если на выбранный сеанс уже проданы все билеты, сеанс следует исключить из плана. Сеансы не должны временно пересекаться и повторяться в плане одного зрителя. Каждый зритель задает список предпочтений из 3 фильмов.

Пример кода реализации триггера по заданию выше, следующий:

```

CREATE TRIGGER FormVisitPlan
ON Предпочтения_зрителей
AFTER INSERT AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @зритель INT;
    DECLARE @counter INT = 0;
    DECLARE @maxFilms INT = 3;

    -- Определение кода добавленного зрителя
    SELECT @зритель = i.код_зрителя FROM inserted i;

    -- Определение списка предпочитаемых фильмов зрителя
    DECLARE @предпочитаемые_фильмы TABLE (название_фильма NVARCHAR(100));
    INSERT INTO @предпочитаемые_фильмы
    SELECT название_фильма FROM Предпочтения_зрителей WHERE код_зрителя = @зритель;

    -- Определение возможных сеансов, согласно предпочтениям
    DECLARE @возможные_сеансы TABLE (номер_сеанса INT, начало_сеанса DATETIME,
    конец_сеанса DATETIME);
    INSERT INTO @возможные_сеансы
    SELECT номер_сеанса, начало_сеанса, конец_сеанса
    FROM Сеансы
    WHERE название_фильма IN (SELECT название_фильма FROM @предпочитаемые_фильмы)
    AND номер_сеанса NOT IN (SELECT номер_сеанса FROM Планировщик_посещений WHERE
    код_зрителя = @зритель)
    AND NOT EXISTS (
        SELECT * FROM Билеты b
        WHERE b.номер_сеанса = Сеансы.номер_сеанса
        GROUP BY b.номер_сеанса
        HAVING COUNT(*) >= (SELECT количество_мест FROM Кинозалы WHERE номер_кинотеатра =
    Сеансы.номер_кинотеатра)
    );

    -- Сортировка сеансов по началу времени
    DECLARE @отсортированные_сеансы TABLE (номер INT IDENTITY(1,1), номер_сеанса INT);

```

```

INSERT INTO @отсортированные_сеансы (номер_сеанса)
SELECT номер_сеанса FROM @возможные_сеансы
ORDER BY начало_сеанса;

-- Вставка в планировщик посещений
WHILE (@counter < @maxFilms)
BEGIN
    DECLARE @сеанс INT, @начало DATETIME, @конец DATETIME, @последнийКонец DATETIME;

    -- Получить время окончания последнего сеанса в плане
    SELECT TOP 1 @последнийКонец = конец_сеанса FROM Планировщик_посещений p
inner join сеансы s on p.номер_сеанса=s.номер_сеанса
    WHERE код_зрителя = @зритель
    ORDER BY дата_и_время_посещения DESC;

    IF @последнийКонец IS NULL SET @последнийКонец = GETDATE();

    -- Выбрать следующий сеанс для посещения
    SELECT TOP 1 @сеанс = номер_сеанса, @начало = начало_сеанса
    FROM @возможные_сеансы
    WHERE начало_сеанса > DATEADD(MINUTE, 30, @последнийКонец)
    ORDER BY начало_сеанса;

    IF @сеанс IS NOT NULL
    BEGIN
        INSERT INTO Планировщик_посещений(код_зрителя, дата_и_время_посещения,
номер_сеанса, номер_порядка_посещения)
        VALUES (@зритель, @начало, @сеанс, @counter + 1);
        SET @counter = @counter + 1;
    END
    ELSE
    BEGIN
        -- Если не удалось найти подходящий сеанс, выход из цикла
        BREAK;
    END
END
END
END

```

### Как узнать, что все выполнено?

Проверьте пункты в этом чек-листе:

- ☐ Создана база данных со всеми таблицами
- ☐ В базе данных установлены базовые ограничения целостности
- ☐ Разработан код триггера согласно условиям задания
- ☐ Триггер учитывает все варианты возможной модификации данных
- ☐ Триггере генерирует сообщения, которые возможно обработать из внешнего приложения
- ☐ Триггеры протестирован на наборе запросов модификации данных
- ☐ Триггер корректно осуществляет заполнение данными таблицы, указанной в варианте задания

## **Варианты заданий:**

### 1 вариант

Даны таблицы: афиша (код концерта, название, жанр, дата и время проведения, зал, стоимость), Исполнители (код исполнителя, имя), Исполнители концертов (код концерта, код исполнителя), Слушатели (код слушателя, имя, адрес электронной почты, телефон), история посещений (код слушателя, код концерта), Сообщения (код сообщения, дата, текст).

Разработать триггер, формирующий предложения для слушателей таким образом, чтобы при добавлении нового концерта для тех, слушателей, которые посещают концерты в среднем больше одного раза в месяц – составлялось sms-сообщение, а для тех, кто меньше – сообщение на электронную почту. Сообщение на электронную почту должно содержать случайный концерт, анонсируемый в следующем месяце, предпочтительный для слушателя по жанру, а sms-сообщение – предпочтительный по исполнителям. В месяц должно быть не больше одного сообщения для клиента.

### 2 вариант

Даны таблицы: склады (номер склада, адрес, вместимость), материалы (код материала, наименование, единица измерения, размер, количество в упаковке), размещение материалов (код материала, номер склада, количество целых единиц, остатки).

Разработать триггер, оптимизирующий распределение остатков материалов таким образом, чтобы при изменении количества материалов на складах осуществлялась проверка возможности дополнения остатков до целых упаковок или штук. В случае такой возможности остатки следует группировать вместе на том складе, где их больше.

### 3 вариант

Даны таблицы: клиенты (код клиента, фамилия, имя, отчество, контактный телефон), скидочные карты (номер карты, код клиента, тип скидочной карты, процент скидки), товары (код товара, название, стоимость, тип скидочной карты, начало акции, окончание акции), предложения для клиентов (код клиента, код товара, стоимость, дата предложения)

Разработать триггер, формирующий предложения для клиентов на товары по скидке таким образом, что при добавлении нового товара, если акция начнётся не раньше, чем через неделю и не позже, чем через месяц, осуществляется добавление товара и расчет новой стоимости. Предложения должны получить все клиенты, имеющие скидочные карты подходящего типа. Каждый клиент в месяц может получать не больше одного предложения. При наличии у клиента нескольких скидочных карт, ему должны формироваться наиболее выгодные предложения.

### 4 вариант

Даны таблицы: рейсы (номер рейса, пункт отправления, пункт назначения, дата и время отправления), авиабилеты (номер билета, номер рейса, код клиента, дата покупки, стоимость), клиенты (код клиента, фамилия, имя, отчество, адрес электронной почты), рассылки (код клиента, номер рейса, стоимость, дата рассылки)

Разработать триггер, формирующий рекламные рассылки на актуальные рейсы для клиентов таким образом, что при добавлении нового рейса составляется список клиентов, которые еще не летали по этому направлению. Каждому из таких клиентов необходимо включить в рассылку новый рейс, а стоимость снизить пропорционально частоте совершаемых перелётов. Помимо этого, клиентам, совершающим в среднем более двух перелётов в месяц, необходимо добавить ещё два актуальных рейса из того же пункта отправления случайным образом. Рейсы, совершаемые с пересадками (в диапазоне 1-2 суток, пункт назначения и пункт отправления совпадают) считаются за один рейс.

#### 5 вариант

Даны таблицы: кемпинги (номер кемпинга, название, локация), места стоянок (номер кемпинга, номер места, вид стоянки, стоимость аренды в сутки), бронирование мест (номер кемпинга, номер места, дата и время начала аренды, дата и время окончания аренды), предложения для клиентов (номер кемпинга, номер места, дата и время начала аренды, продолжительность).

Разработать триггер, формирующий предложения для туристов по бронированию мест кемпингов таким образом, чтобы при изменении времени бронирования осуществлялась проверка отсутствия занятости мест на какое-либо полное количество дней. В случае такой возможности, если свободные дни выпадают на выходные, необходимо сформировать предложение. Также в качестве предложений необходимо указывать свободные недели и месяцы, если они начинаются с понедельника или начала месяца.

#### 6 вариант

Даны таблицы: станции (номер станции, название, количество путей, является остановочным пунктом), маршруты (номер маршрута, номер по порядку, номер станции, расстояние, время следования по перегону), расписание поездов (номер поезда, номер маршрута, начальная станция, конечная станция, время отправления с начальной станции), остановки (номер поезда, номер станции, время остановки).

Разработать триггер, анализирующий расписание поездов таким образом, чтобы при назначении новой остановки какому-либо поезду выполнялась корректировка расписания всех остальных поездов. По перегону могут следовать два поезда одновременно с интервалом не менее 5 минут. Корректировка происходит для всех поездов, отправляющихся позже, если суммарное время отставания последнего поезда не превышает время более раннего отправления самого первого поезда при корректировке поездов, отправляющихся раньше. В противном случае корректировка происходит для поездов, отправляющихся раньше. Время остановки на остановочных пунктах увеличивать нельзя.

#### 7 вариант

Даны таблицы: Остановки (код остановки, название остановки, адрес остановки), маршруты (номер маршрута, код остановки, порядковый номер, время в пути), кратчайшие маршруты (начальная остановка, конечная остановка, номер маршрута).

Разработать триггер, анализирующий кратчайшие маршруты таким образом, чтобы при добавлении новой остановки к какому-либо маршруту выполнялась проверка расстояний до этой

остановки. Если окажется, что после добавления можно каким-либо образом доехать быстрее, соответствующие данные добавляются или изменяются в таблице Кратчайшие маршруты.

#### 8 вариант

Даны таблицы: Маршруты трамваев (номер маршрута, код остановки, номер остановки), Остановки (Код остановки, название остановки), Пассажиропоток (Код остановки, дата, время начала исследования, время окончания исследования, количество пассажиров), график ремонтов (код остановки, дата начала ремонта, дата окончания ремонта).

Разработать триггер, формирующий предложения по ремонтам таким образом, что при новом исследовании пассажиропотока выполняется анализ загруженности участка движения. Если окажется, что можно организовать движение всех маршрутов на каком-либо участке с меньшим средним количеством пассажиров на одну остановку, то включить остановки изначального участка таких маршрутов в график ремонта.

#### 9 вариант

Даны таблицы: Залы (номер зала, название, вместимость), Переходы между залами (номер первого зала, номер второго зала), экспозиции (номер зала, экспонат), Предпочтения посетителей (код посетителя, экспонат), Маршрут (код посетителя, дата и время посещения, номер зала, номер в порядке посещения)

Разработать триггер, автоматически формирующий маршруты по залам для посетителей, исходя из его предпочтений таким образом, что сначала в маршрут должны попасть залы с экспонатами из списка предпочтений. На посещение одного зала выделяется 10 минут. Если в зале в это время уже присутствует максимально возможное количество посетителей, его необходимо передвинуть в маршрутном списке. Залы в маршруте не должны повторяться. Каждый посетитель задает список предпочтений из 5 экспонатов.

#### 10 вариант

Даны таблицы: Тренеры (код тренера, фамилия, имя, отчество), Тренажерные залы (номер зала, название), Время работы (номер зала, день недели, время начала работы, время окончания работы), Расписание (код тренера, номер зала, время начала тренировки, время окончания тренировки)

Разработать триггер, автоматически формирующий расписание работы тренеров исходя из загруженности залов. В каждом зале может работать не больше 3 тренеров. Время одной тренировки составляет 2 часа. Время работы каждого тренера не должно превышать 40 часов в неделю. Планировать расписание необходимо оптимальным образом, не допуская длительных перерывов между тренировками. Приоритетным является расписание, при котором тренеру как можно дольше планируются тренировки в одном зале.