

#18 Обеспечение целостности данных в рекурсивных структурах

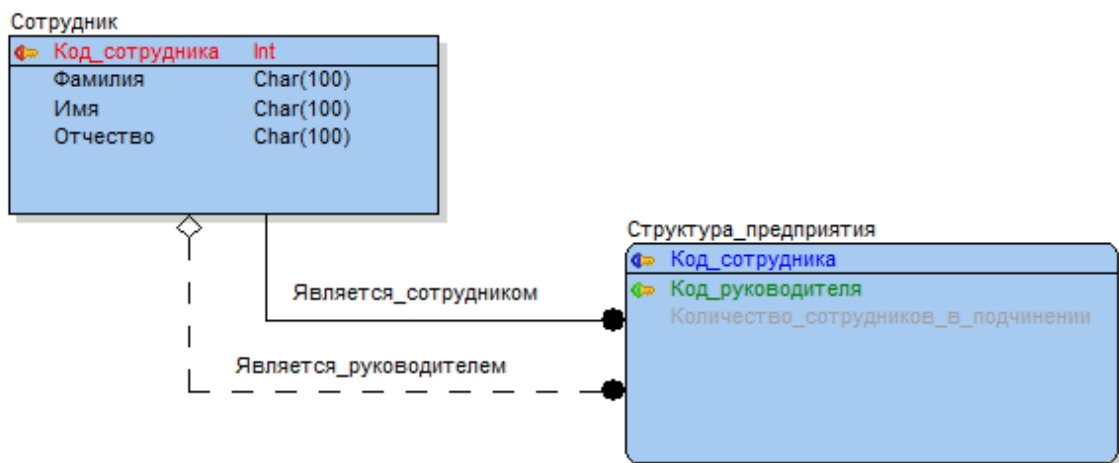
Зачем делать эту лабораторную работу?

1. Чтобы научиться реализовывать иерархические и сетевые структуры данных средствами реляционных баз данных.
2. Чтобы научиться задавать ограничения целостности для рекурсивных структур.
3. Чтобы освоить механизмы обеспечения целостности данных в рекурсивных структурах с помощью триггеров.
4. Чтобы научиться использовать оператор With для организации рекурсивных запросов.

Что нужно делать?

1. Создать базу данных.

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



Для приведённой выше схемы данных код будет выглядеть так:

```
CREATE TABLE [Сотрудник]
(
    [Код_сотрудника] Int NOT NULL,
    [Фамилия] Char(100) NOT NULL,
    [Имя] Char(100) NOT NULL,
    [Отчество] Char(100) NOT NULL
)

ALTER TABLE [Сотрудник] ADD CONSTRAINT [PK_Сотрудник] PRIMARY KEY ([Код_сотрудника])

CREATE TABLE [Структура_предприятия]
(
    [Код_сотрудника] Int NOT NULL,
    [Код_руководителя] Int NULL,
    [Количество_сотрудников_в_подчинении] Int NULL
)
```

```
ALTER TABLE [Структура_предприятия] ADD CONSTRAINT [PK_Структура_предприятия] PRIMARY KEY ([Код_сотрудника])
```

```
ALTER TABLE [Структура_предприятия] ADD CONSTRAINT [Является_сотрудником] FOREIGN KEY ([Код_сотрудника]) REFERENCES [Сотрудник] ([Код_сотрудника]) ON UPDATE NO ACTION ON DELETE NO ACTION
```

```
ALTER TABLE [Структура_предприятия] ADD CONSTRAINT [Является_руководителем] FOREIGN KEY ([Код_руководителя]) REFERENCES [Сотрудник] ([Код_сотрудника]) ON UPDATE NO ACTION ON DELETE NO ACTION
```

2. Разработать триггер.

Реализуйте триггер согласно варианту задания. Триггер должен проверять все условия задания. В данном случае для корректного заполнения таблицы «Структура предприятия» необходимо выполнение следующих условий:

- 1) Каждый сотрудник имеет только одного руководителя;
- 2) Каждый сотрудник не является сам себе руководителем;
- 3) Каждый руководитель в первую очередь сотрудник;
- 4) Есть только один сотрудник, генеральный директор, который никому не подчиняется;
- 5) Каждый сотрудник не должен находиться в роли собственного руководителя не только непосредственно, но и опосредствованно, через других сотрудников.

В данном случае правила 1 и 3 уже выполняются за счёт базовых ограничений целостности данных. Все остальные правила требуют написание триггера.

Пример кода реализации триггера для условий выше, следующий:

```
CREATE TRIGGER Проверка_структуры
ON [Структура_предприятия]
AFTER INSERT, UPDATE
AS
BEGIN
    -- Условие 2
    IF EXISTS(SELECT 1 FROM inserted i WHERE i.[Код_сотрудника] = i.[Код_руководителя])
    BEGIN
        RAISERROR('Сотрудник не может быть своим собственным руководителем', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    -- Условие 4
    IF (SELECT COUNT(*) FROM [Структура_предприятия] WHERE [Код_руководителя] IS NULL) >
1
    BEGIN
        RAISERROR('В системе может быть только один генеральный директор, который никому
не подчиняется', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END;

    -- Условие 5
    WITH RecursiveHierarchy (Код_сотрудника, Код_руководителя) AS (
        -- Инициализация рекурсии с новой вставленной или обновлённой записи
        SELECT i.[Код_сотрудника], i.[Код_руководителя]
        FROM inserted i
```

```

UNION ALL

-- Рекурсивное соединение для построения полной иерархии
SELECT sp.[Код_сотрудника], sp.[Код_руководителя]
FROM [Структура_предприятия] sp
INNER JOIN RecursiveHierarchy rh ON sp.[Код_сотрудника] = rh.[Код_руководителя]
)
-- Проверка на наличие самоподчинения посредством проверки рекурсивной иерархии

SELECT 1
FROM RecursiveHierarchy
WHERE [Код_сотрудника] = [Код_руководителя] AND [Код_руководителя] IS NOT NULL
GROUP BY [Код_сотрудника], [Код_руководителя]
HAVING COUNT(*) > 1;
IF @@ROWCOUNT>0
BEGIN

    RAISERROR('Недопустимо, чтобы сотрудник был опосредованно подчинен самому себе.',
16, 1);
    ROLLBACK TRANSACTION;
    RETURN;
END
END

```

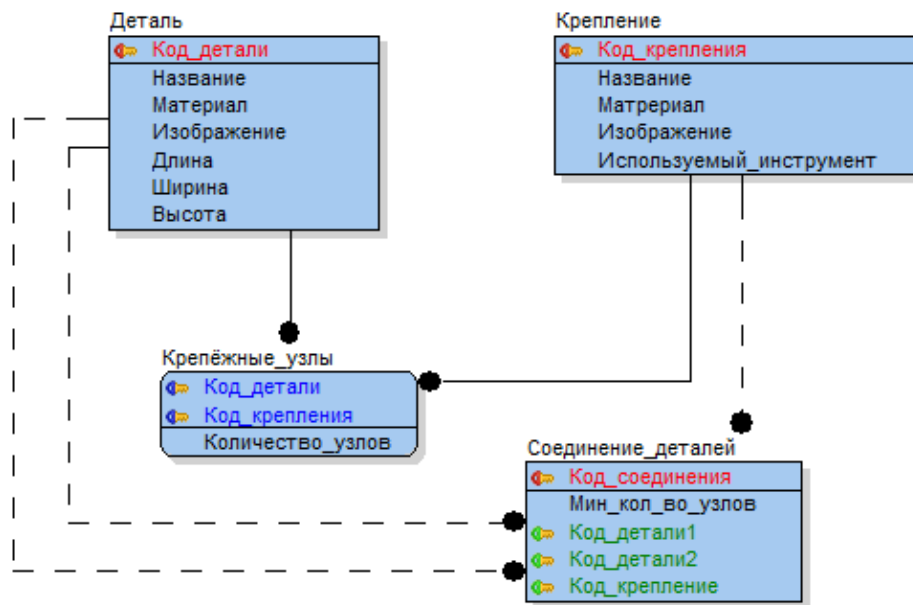
Как узнать, что все выполнено?

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 обеспечивает корректное заполнение рекурсивной структуры данных
- ☐ Триггер генерирует сообщения, которые возможно обработать из внешнего приложения
- ☐ Триггеры протестирован на наборе запросов модификации данных

Варианты заданий:

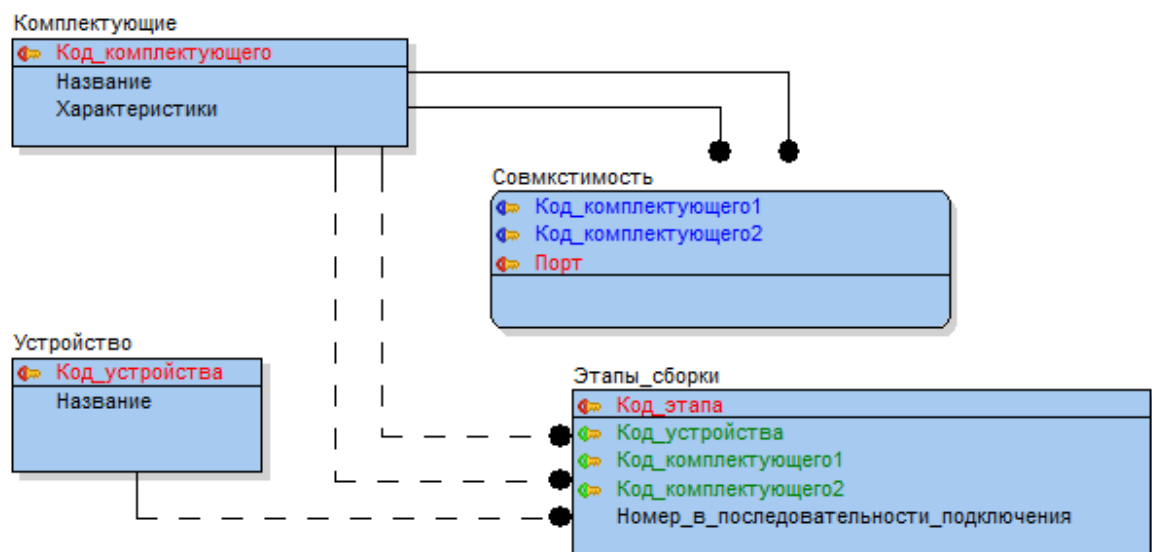
1 вариант



Для таблицы Соединение деталей должны обеспечиваться следующие ограничения целостности данных:

1. Одинаковые детали имеют только один способ крепления конкретным элементом крепления;
2. Можно скреплять детали, только имеющие необходимое минимальное количество узлов для крепления;
3. Можно скреплять детали только одинаковыми крепежными элементами;
4. Каждая деталь может соединяться с такой же деталью не более пяти раз;
5. Правило 4 также распространяется на соединение через другие детали;
6. Минимальное количество узлов для соединения не должно превышать количества узлов в деталях;
7. При соединении первая деталь должна иметь большее или равное количество узлов для соединения.
8. Нельзя допускать последовательностей соединения более пяти деталей с постоянно уменьшающимся минимальным количеством узлов крепления.

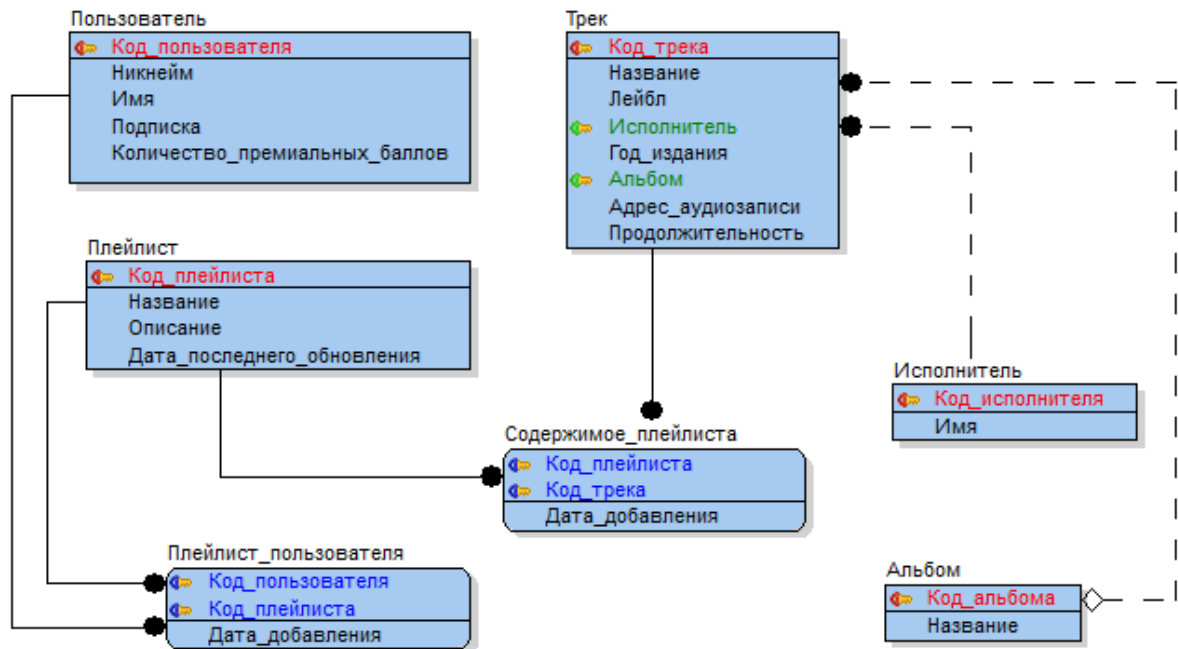
2 вариант



Для таблицы Этапы сборки должны обеспечиваться следующие ограничения целостности данных:

1. Одинаковые комплектующие не могут подключаться друг к другу;
2. Можно подключать только совместимые комплектующие;
3. Комплектующее 2 всегда подключается к уже смонтированному комплектующему 1;
4. Все одинаковые комплектующие подключаются одновременно;
5. При подключении должен соблюдаться количественный принцип от меньшего к большему: например, к одной материнской плате подключается много планок оперативной памяти;
6. Нельзя допускать подключение на следующих этапах в одной последовательности уже ранее подключенных комплектующих;
7. При подключении максимальное количество этапов не должно превышать 10;
8. Нельзя допускать последовательностей подключения комплектующих с чередующимися портами подключения.

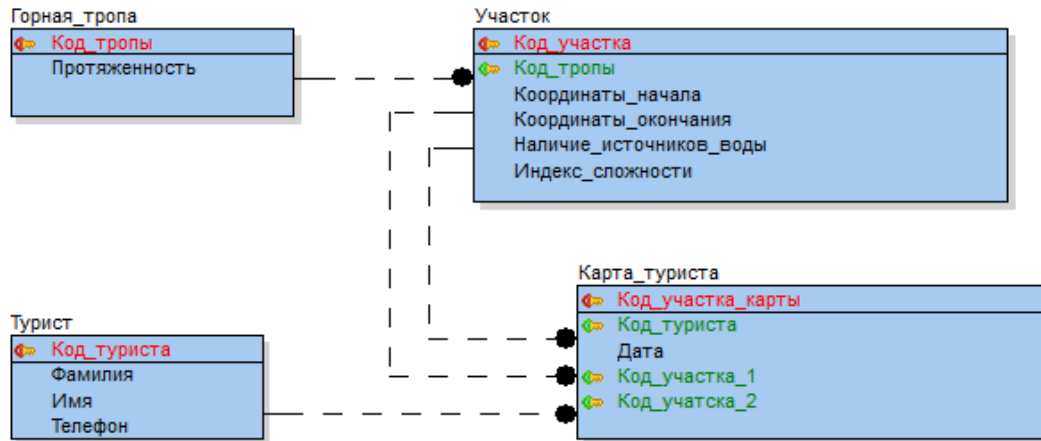
3 вариант



Для таблицы Последовательность воспроизведения должны обеспечиваться следующие ограничения целостности данных:

1. Одинаковые треки не могут воспроизводиться последовательно;
2. Сначала воспроизводится трек 1, затем трек 2;
3. Для каждого пользователя в пределах одной даты последовательность воспроизведения должна быть непрерывной;
4. Треки в последовательности должны быть из предпочитаемых жанров пользователя;
5. Треки, для которых все жанры совпадают, не должны повторяться или чередоваться в последовательности;
6. Не допускается включение треков с постоянно увеличивающимся количеством жанров в последовательностях больше трёх подряд;
7. Последовательности треков не должны быть закольцованы;
8. Для каждого пользователя разнообразие последовательностей каждой последующей даты должно быть более 70% по отношению к предыдущей дате.

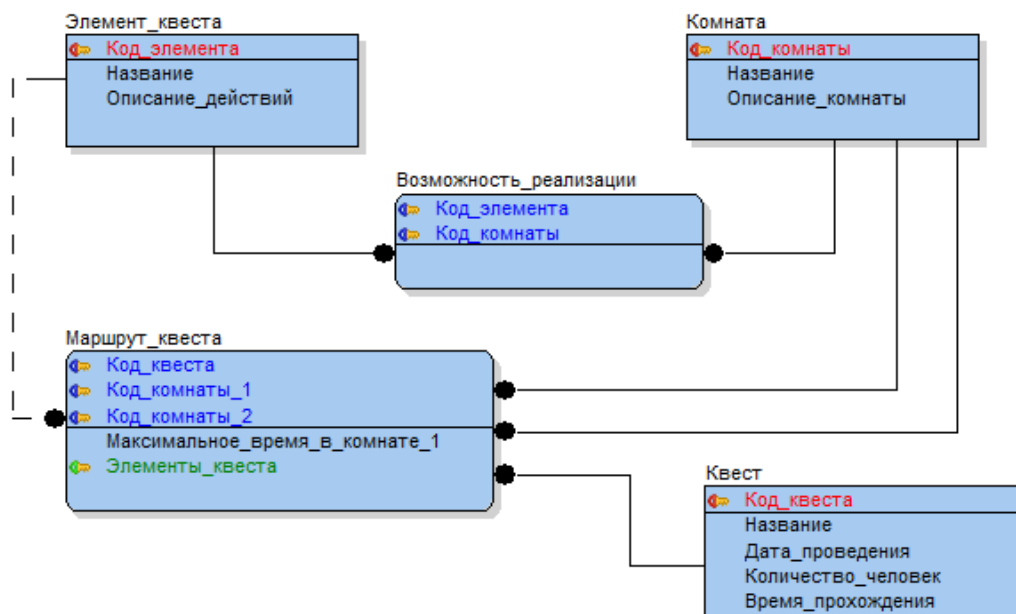
4 вариант



Для таблицы Карта туриста должны обеспечиваться следующие ограничения целостности данных:

1. В одну дату посещения для одного туриста можно включать только участки из одной тропы;
2. В карте для туриста маршрут может только разветвляться, но не сходиться в итоге в одну тропу;
3. При планировании участков каждый второй участок должен быть с наличием воды;
4. Недопустимо включать больше трёх участков подряд с повышением или понижением индекса сложности;
5. В одной карте может быть пересечение троп не более, чем в двух участках подряд;
6. Суммарный индекс сложности участков каждого нового ответвления не должен превышать средний суммарный индекс сложности уже сформированных ранее ответвлений карты более чем на 25%;
7. Повторение пройденных участков ни для одной ветки карты не допускается;
8. Для каждой последующей даты варианты реализации карты для одного и того же туриста должны различаться более чем на 50%.

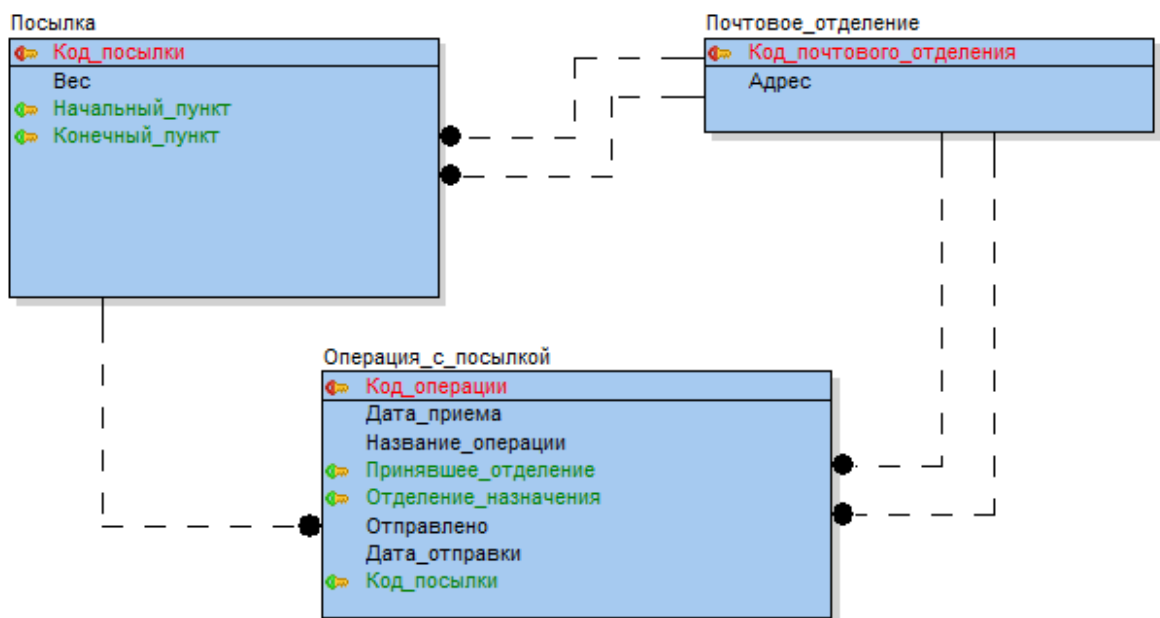
5 вариант



Для таблицы Маршруты квестов должны обеспечиваться следующие ограничения целостности данных:

1. Для каждой даты маршруты квестов должны быть непрерывные;
2. Каждый маршрут должен начинаться в одной комнате, маршруты могут расходиться только по принципу ветвления;
3. Элементы квестов в каждой из веток не должны повторяться;
4. Общее время нахождения во всех комнатах каждой из веток не должно превышать времени квеста;
5. Комнаты должны быть предназначены для реализации заданных элементов квеста;
6. В одну дату маршруты квестов не должны накладываться по времени при прохождении одних и тех же комнат;
7. Количество веток маршрута для одного квеста не должно превышать количества человек;
8. Время нахождения в каждой следующей комнате должно уменьшаться пропорционально.

6 вариант



Для таблицы Операции с посылками должны обеспечиваться следующие ограничения целостности данных:

1. Первой операцией для каждой посылки должно быть отправление посылки из начального пункта;
2. Пересылка должна осуществляться последовательно от одного отделения к другому;
3. Следующее отделение не может принять посылку, если предыдущее не поставило опцию «отправлено»;
4. Запрещается многократная пересылка между одними и теми же почтовыми отделениями;
5. Недопустима пересылка между пятью и более отделениями с постоянно увеличивающимися интервалами ожидания;
6. В случае «возврата» посылка должна пересылаться тем же путём;
7. Повторное отправление можно осуществлять тем же путём, но с заменой каждого второго почтового отделения;
8. Посылки не должны повторять путь недоставленных предыдущих посылок, если их начальный и конечный пункты совпадают.