

ЗАДАЧА 1

Структура данных

```
SET NOCOUNT ON
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--/*
IF OBJECT_ID('[TestDoc].[Accounts]') IS NOT NULL
    DROP TABLE [TestDoc].[Accounts]

IF OBJECT_ID('[TestDoc].[Contracts]') IS NOT NULL
    DROP TABLE [TestDoc].[Contracts]
--*/

-----
-- Структура данных                                     --
-----

IF SCHEMA_ID('TestDoc') IS NULL
    EXEC('
        CREATE SCHEMA [TestDoc]
    ')
GO

IF OBJECT_ID('[TestDoc].[Contracts]') IS NULL
    -- Договора
    CREATE TABLE [TestDoc].[Contracts]
    (
        [Id]          Int          NOT NULL IDENTITY(1,1),
        [DocNo]       NVarChar(50) NOT NULL,
        [DateFrom]    Date          NOT NULL, -- Дата, когда договор начал действовать
        [DateTo]      Date          NULL,   -- Дата, когда договор прекращает действовать
        (последний день действия договора); NULL = бесконечность
        -- ... И еще какие-то поля
        PRIMARY KEY CLUSTERED([Id])
    )
GO

IF OBJECT_ID('[TestDoc].[Accounts]') IS NULL
    -- Счета
    CREATE TABLE [TestDoc].[Accounts]
    (
        [Id]          Int          NOT NULL IDENTITY(1,1),
        [Contract_Id] Int          NOT NULL, -- Договор, в рамках которого счет заключен
        [Number]      NVarChar(50) NOT NULL, -- Номер счета
        [DateTimeFrom] DateTime    NOT NULL, -- Момент времени (дата+время!), когда счет начал
        действовать
        [DateTimeTo]  DateTime      NULL,   -- Момент времени (дата+время!), когда счет
        прекратил действовать
        -- ... И еще какие-то поля
        PRIMARY KEY CLUSTERED([Id]),
        FOREIGN KEY ([Contract_Id]) REFERENCES [TestDoc].[Contracts] ([Id])
    )
GO
```

Есть таблица контрактов **[TestDoc].[Contracts]**. У контракта есть период действия. Поле **[DateFrom]** – дата начала действия договора. Поле **[DateTo]** – последний день действия договора. **NULL** в поле **[DateTo]** – бесконечность.

Есть таблица счетов. Счета открываются в рамках договора. Поле **[DateTimeFrom]** – момент времени, когда счет открыли. **[DateTimeTo]** – момент времени, когда счета закрыли.

Задание

Напишите скрипт, который вернет список «ошибок» в системе. Под «ошибкой» подразумевается ситуация, когда счет действует (действовал), а договор, в рамках которого заключен счет, не действует (не действовал).

Задача 2

Структура

```
set nocount on
set ansi_nulls on
set quoted_identifier on
go

-----
-- Создание структуры
-----

if schema_id('Test') is null
    exec('create schema [Test]')
go

if object_id('[Test].[Contracts]') is null
    create table [Test].[Contracts]
    (
        [Id]          int      not null identity(1,1),
        [Type_Id]     int      not null,
        [Client_Id]   int      not null,
        [DateFrom]    date     not null,
        [DateTo]      date     null,
        primary key clustered ([Id])
    )
go
```

Есть таблица договоров (**[Test].[Contracts]**). У договора есть период действия. Также есть тип (**[Type_Id]**) и указание клиента (**[Client_Id]**).

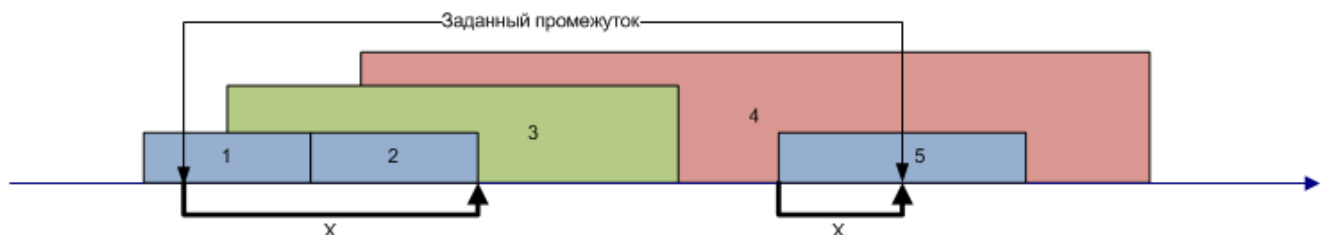
Напишите скрипт, который вернет список всех непересекающихся и несмежных периодов (т.е., между двумя периодами должен быть минимум один «пустой» день) между **@DateBegin** и **@DateEnd** включительно, когда у клиента был хотя бы один активный договор типа **@Type_Id**. Если, можете

Пример параметров:

```
declare
    @Type_Id      int      = 1,
    @DateBegin    date     = '20000601',
    @DateEnd      date     = '20010131'
```

Учтите, что у клиента может быть несколько действующих договоров одного типа одновременно.

Пояснение:



На данной картинке каждый прямоугольник – договор, расположенный на временной шкале. Цветом обозначен тип договора (т.е. например, синие – с типом 1, зеленые с типом 2, красные с типом 3).

В заданном временном промежутке нужно найти все диапазоны времени, когда действовал хотя бы один договор «синего типа». На рисунке то, что нужно вернуть обозначено как «X».

ЗАДАЧА 3

Структура

```
if schema_id('TestCars') is null
    exec('create schema [TestCars]')
go

if object_id('[TestCars].[Points]') is null
    -- Точки
    create table [TestCars].[Points]
    (
        [Id]          int      not null identity(1,1),
        [Type_Id]     char(1) not null,          -- тип точки: D = Склад; S = Магазин
        primary key clustered ([Id]),
        check ([Type_Id] in ('D', 'S'))
    )
go

if object_id('[TestCars].[Cars]') is null
    -- Машины
    create table [TestCars].[Cars]
    (
        [Id]          int      not null identity(1,1),
        [Capacity]    int      not null,          -- Грузоподъемность
        primary key clustered ([Id])
    )
go

if object_id('[TestCars].[Routes]') is null
    -- Машины
    create table [TestCars].[Routes]
    (
        [Id]          int      not null identity(1,1),
        [Point_Id]    int      not null,
        [Car_Id]      int      not null,
        [Load]        int      not null,          -- Изменение загрузки авто при посещении
                                                -- данной точки. +N = в авто дозагрузили N кг; -N = из машины выгрузили N кг
                                                -- При этом в магазине точка
                                                -- разгружается, а при посещении склада загружается/дозагружается
        [ArrivalTime] datetime not null,
        [DepartureTime] datetime not null,
        primary key clustered ([Id]),
        foreign key ([Point_Id]) references [TestCars].[Points] ([Id]),
        foreign key ([Car_Id]) references [TestCars].[Cars] ([Id])
    )
go
```

Есть таблица точек [TestCars].[Points]. Точками являются склады или магазины.

Есть таблица машин [TestCars].[Cars]. У каждой машины может быть своя грузоподъемность.

Есть точки маршрута машин [TestCars].[Routes], которые посещены машинами.

В течение дня машина совершает несколько поездок. Каждая поездка начинается с загрузки машины на складе. Далее машина посещает несколько магазинов, в каждом из которых частично разгружается. Затем машина может прибыть на один из складов,

дозагрузиться и начать следующий маршрут. Каждый маршрут всегда проезжается в течение одного календарного дня (нет маршрутов с началом и концом в разных днях).

Машина движется всегда с одной и той же скоростью, скорости разных машин могут отличаться. По одному и тому же маршруту может проехать несколько различных машин.

Задание

1. найти топ-3 неэффективных маршрутов и топ-3 неэффективных машин по каждому из критериев:

1.1. Недозагруженность машины, т.е. отношение «загрузка машины/грузоподъемность» после выезда со склада в начале маршрута

1.2. Размер остатка в машине после посещения всех магазинов на маршруте

2. Найти самую быструю машину или доказать, что по представленным данным это невозможно сделать.

**РЕШЕНИЕ ВЫЛОЖИТЬ НА ЛЮБОЙ ФАЙЛООБМЕНИК И ПРИСЛАТЬ ССЫЛКУ НА E-MAIL
NICK@KESCOM.COM**