«Большинство задач имеет несколько решений, просьба предоставить оптимальное на Ваш взгляд. Если в условиях задачи Вы видите инвариантность, то просьба предоставить несколько решений, удовлетворяющих на Ваш взгляд, разным трактовкам данной задачи»

**Задача 1.**

Дана структура А.

1. Таблица сотрудников.

CREATE TABLE dbo.Persons

(

[Id] Int NOT NULL,

[Name] VarChar(**250**) NOT NULL, -- Имя сотрудника

[Depart\_Id] Int NULL, -- Ссылка на Departs

CONSTRAINT [PK\_Persons] PRIMARY KEY CLUSTERED ([Id]),

CONSTRAINT [FK\_Persons\_Departs]

FOREIGN KEY ([Depart\_Id]) REFERENCES dbo.Departs ([Id])

)

|  |  |  |
| --- | --- | --- |
| Id | Name | Depart\_Id |
| 1 | Иванов | 1 |
| 2 | Петров | 1 |
| 4 | Сидоров | 2 |
| 5 | Антонов | 2 |
| 7 | Бубонов | 2 |

1. Таблица отделов.

CREATE TABLE dbo.Departs

(

[Id] Int NOT NULL,

[Name] VarChar(**250**) NOT NULL, -- Название отдела

CONSTRAINT [PK\_Departs] PRIMARY KEY CLUSTERED ([Id])

)

|  |  |
| --- | --- |
| Id | Name |
| 1 | ААА |
| 2 | БББ |
| 3 | ВВВ |

Написать запрос, который будет возвращать список сотрудников с указанием для каждого сотрудника отдела, в котором данный сотрудник работает. Т.е. результат должен содержать два поля: PersonName, DepartName.

**Задача 2.**

*По структуре А.*

Написать запрос, который будет возвращать список отделов с указанием для каждого отдела кол-ва сотрудников в данном отделе. Т.е. результат должен содержать два поля: DepartName, PersonsCount.

**Задача 3.**

Дана структура Б.

Таблица документов. С огромным кол-вом записей.

CREATE TABLE dbo.Docs

(

[Id] Int NOT NULL,

[DateTime] DateTime NOT NULL, -- Дата и время документа

[Number] VarChar(**100**) NOT NULL, -- Номер документа

-- ...

[Client\_Id] Int NOT NULL, -- Ссылка на клиента

[Value] Money NULL, -- Изменение баланса взаиморасчетов с клиентом ("+" получены деньги от клиента, "-" отгружен товар клиенту)

CONSTRAINT [PK\_Docs] PRIMARY KEY CLUSTERED ([Id]),

-- ...

)

Предложите индекс, который позволит оптимально отвечать на вопрос: «Каков баланс (сумма всех Value по клиенту с начала истории до заданного момента времени) по клиенту К на дату Д?». Напишите такой запрос с использованием переменных:

DECLARE

@Client\_Id Int,

@BalanceDateTime DateTime

**Задача 4.**

-- Таблица инструментов

CREATE TABLE [Instruments]

(

[Id] Int NOT NULL PRIMARY KEY CLUSTERED,

[Name] VarChar(**255**) NOT NULL

)

-- Таблица курсов

CREATE TABLE [Rates]

(

[Instrument\_Id] Int NOT NULL REFERENCES [Instruments] ([Id]),

[Date] Date NOT NULL,

[Value] Numeric(**32**,**8**) NULL,

PRIMARY KEY CLUSTERED([Instrument\_Id], [Date])

)

Для понимания, представьте себе таблицу курсов доллара. А таблице Rates не на все даты есть значения (при этом для одного инструмента на некоторую дату может быть определено значение, а для другого может не определено значение).

В качестве фильтра даны дата, на которую надо сделать выборку данных, и набор идентификаторов инструментов, для которых надо вывести курсы.

DECLARE

@Date Date = '20130401'

DECLARE @Instruments Table

(

[Id] Int NOT NULL,

PRIMARY KEY CLUSTERED([Id])

)

INSERT INTO @Instruments

VALUES (**1**), (**10**), (**101**), /\* ... \*/ (**10101**)

Напишите запрос, который вернет список инструментов ([Instrument\_Id], [Instrument\_Name]), с указанием для каждого инструмента ближайшего последнего [Value] не позднее даты @Date, а также дату, на которую данное значение [Value] было установлено в таблице [Rates].

SELECT

[Instrument\_Id] = ...,

[Instrument\_Name] = ...,

[Value] = ...,

[Date] = ...

...

|  |  |  |
| --- | --- | --- |
| Instrument\_Id | Date | Value |
| 1 | D1 | X11 |
| 2 | D1 | X21 |
| 1 | D2 | X12 |
| 3 | D2 | X32 |
| 1 | D3 | X13 |

На дату D3 ответом будет:

|  |  |  |  |
| --- | --- | --- | --- |
| Instrument\_Id | Instrument\_Name | Date | Value |
| 1 | Name\_1 | D3 | X13 |
| 2 | Name\_2 | D1 | X21 |
| 3 | Name\_3 | D2 | X32 |

**Задача 5.**

Дана структура В.

-- Таблица счетов (1000 записей)

CREATE TABLE [Accounts]

(

[Account\_Id] Int NOT NULL PRIMARY KEY CLUSTERED,

[Name] VarChar(**255**) NOT NULL

)

-- Таблица проводок (10 000 000 записей)

CREATE TABLE [Transactions]

(

[Transaction\_Id] Int NOT NULL PRIMARY KEY NONCLUSTERED,

[Account\_Id] Int NOT NULL REFERENCES [Accounts]([Account\_Id]),

[Date] DateTime NOT NULL,

[Value] Numeric(**18**,**2**) NOT NULL

)

CREATE CLUSTERED INDEX [IX\_Transactions] ON [Transactions] ([Account\_Id], [Date])

Оптимизировать следующий запрос, не изменяя структуры таблиц, не изменяя существующие и не добавляя новых индексов.

SELECT

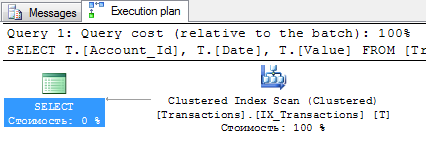
T.[Account\_Id],

T.[Date],

T.[Value]

FROM [Transactions] T

WHERE T.[Date] >= '20100101' AND T.[Date] < '20100201'



Приветствуются развернутые ответы с приведением Execution Plan-а, анализом read-ов и с комментариями.

**Задача 6.**

Дана таблица TestDocs. В этой таблице есть поля [Type\_Id], [DateTime], [Number]…

Поле [Number] заполняется автоматически. На поле [Number] не определено UNIQUE умышлено (для целей задачи, предлагать не надо).

SET NOCOUNT ON

SET ANSI\_NULLS ON

SET QUOTED\_IDENTIFIER ON

GO

IF OBJECT\_ID('[dbo].[TestDocs]') IS NULL

CREATE TABLE [dbo].[TestDocs]

(

[Id] Int NOT NULL IDENTITY(**1**,**1**),

[Type\_Id] Char(**1**) COLLATE Cyrillic\_General\_BIN NOT NULL,

[DateTime] DateTime NOT NULL,

[Number] VarChar(**50**) COLLATE Cyrillic\_General\_BIN NOT NULL,

[spid] Int DEFAULT(@@SPID) NOT NULL,

/\* ... \*/

PRIMARY KEY CLUSTERED([Id])

)

GO

IF OBJECT\_ID('[dbo].[TestDocs (After Insert)]') IS NULL

EXEC('

CREATE TRIGGER [dbo].[TestDocs (After Insert)]

ON [dbo].[TestDocs]

AFTER INSERT

AS

RETURN

')

GO

ALTER TRIGGER [dbo].[TestDocs (After Insert)]

ON [dbo].[TestDocs]

AFTER INSERT

AS

WAITFOR DELAY '00:00:00.100'

RETURN

GO

1. Создайте необходимые индексы, если считаете нужным.
2. Напишите хранимую процедуру вставки документа.

IF OBJECT\_ID('[dbo].[TestDocs\_Insert]') IS NULL

EXEC ('CREATE PROCEDURE [dbo].[TestDocs\_Insert] AS RETURN (-1)')

GO

ALTER PROCEDURE [dbo].[TestDocs\_Insert]

@Type\_Id Char(**1**),

@DateTime DateTime,

@NewNumber VarChar(50) = NULL OUT – Исключительно OUTPUT-параметр!

AS

-- Нужно вставить данные в таблицу [dbo].[TestDocs]

-- Поле Number заполнить по правилу:

-- YYYYMMDD/Type-Номер

-- где YYYY - год DateTime

-- MM - месяц DateTime

-- DD - день DateTime

-- Type - тип документа

-- Номер - уникальный номер по порядку в РАМКАХ даты и типа!

-- При этом номера должны расти внутри дня, НЕ ОБРАЩАЯ ВНИМАНИЕ НА ВРЕМЯ, в порядке поступления документов.

-- Номера для потока:

-- 1) <Type\_Id="A" DateTime="2013-01-01 09:15:00"> Номер будет: 20130101/A-1

-- 2) <Type\_Id="B" DateTime="2013-01-01 08:15:00"> Номер будет: 20130101/B-1

-- 3) <Type\_Id="C" DateTime="2013-01-01 03:15:00"> Номер будет: 20130101/C-1

-- 4) <Type\_Id="A" DateTime="2013-01-01 02:10:00"> Номер будет: 20130101/A-2

-- 5) <Type\_Id="B" DateTime="2013-01-01 04:15:00"> Номер будет: 20130101/B-2

-- 6) <Type\_Id="B" DateTime="2013-01-01 04:20:00"> Номер будет: 20130101/B-3

-- 7) <Type\_Id="A" DateTime="2013-01-01 07:25:00"> Номер будет: 20130101/A-3

-- 8) <Type\_Id="A" DateTime="2013-01-02 02:15:00"> Номер будет: 20130102/A-1

-- 9) <Type\_Id="B" DateTime="2013-01-02 04:15:00"> Номер будет: 20130102/B-1

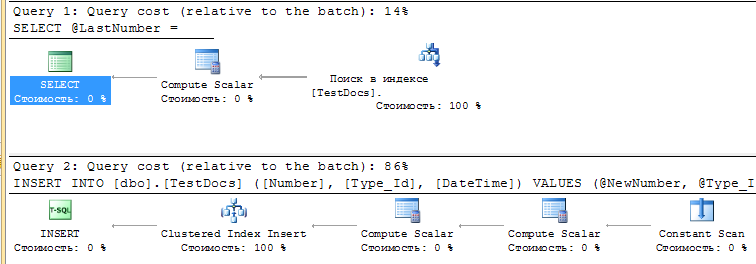
--10) <Type\_Id="C" DateTime="2013-01-03 03:15:00"> Номер будет: 20130103/C-1

/\* ... \*/

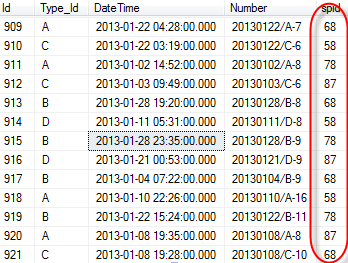
RETURN (**1**)

GO

1. При тестировании используйте не менее 4-х параллельных сессий.
2. Убедитесь в том, что Вы используете минимально необходимое кол-во ресурсов; что параллельные потоки не занимаются ожиданием друг друга (для 1000 документов в каждом потоке все потоки должны отработать примерно за 2 минуты).



1. Убедитесь в отсутствии задвоений номеров.
2. Убедитесь, что вставка в таблицу будет осуществляться из разных потоков параллельно, а не один после полностью отработавшего другого (в таблице поле [spid] будет заполнено вразнобой).



Скрипт тестирования (DELAY не менять):

------------------------------------------------------

-- ТЕСТ!

-- Для тестирования запустить параллельно в нескольких окнах!

------------------------------------------------------

SET NOCOUNT ON

SET ANSI\_NULLS ON

SET QUOTED\_IDENTIFIER ON

SELECT [@@TRANCOUNT.Start] = @@TRANCOUNT

IF @@TRANCOUNT > **0** ROLLBACK

GO

DECLARE

@Type\_Id Char(**1**),

@DateTime DateTime,

@Count Int = **1000**

WHILE @Count > **0** BEGIN

SELECT

@Type\_Id = CHAR(ASCII('A') + CAST(RAND()\***4** AS Int)),

@DateTime = DATEADD(

day,

FLOOR(RAND()\***30**),

DATEADD

(

hour,

FLOOR(RAND()\***24**),

DATEADD

(

minute,

FLOOR(RAND()\***60**),

CAST('20130101' AS DateTime)

)

)

)

-- SELECT [@Type\_Id] = @Type\_Id, [@DateTime] = @DateTime

EXEC [dbo].[TestDocs\_Insert]

@Type\_Id = @Type\_Id,

@DateTime = @DateTime

WAITFOR DELAY '00:00:00.050'

SET @Count -= **1**

END

GO

SELECT [@@TRANCOUNT.End] = @@TRANCOUNT

IF @@TRANCOUNT > **0** ROLLBACK

GO

**Задача 7.**

Дана таблица подразделений с древовидной связью Id <- Parent\_Id.

SET NOCOUNT ON

SET ANSI\_NULLS ON

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [Departs]

(

[Id] Int NOT NULL IDENTITY(**1**,**1**),

[Parent\_Id] Int NULL REFERENCES [Departs] ([Id]),

[Name] NVarChar(**100**) NOT NULL,

PRIMARY KEY CLUSTERED([Id])

)

GO

-- Предложите индексы, если считаете нужным

Предложите необходимые индексы, если считаете нужным.

Дан параметр – идентификатор некоторого подразделения. Надо выдать все подразделения, исключив из выборки указанное в параметре, а также исключив всех его подчиненных элементов.

DECLARE

@WithOutDepart\_Id Int = ...

-- Выдать все подразделения, кроме подразделения @WithOutDepart\_Id и всех его потомков

...

SELECT D.\*

FROM [Departs] D

...