# Vytvoreni a pouziti #temp tabulky na Update s joinem dalsich tabulek

Update s joinem pomuze zrychlit vkladani a updaty velkych mnozstvi dat na server. Pouzito ve zmenovadle, vytvorime tabulku #tempMotivlet. Problem byl, ze jsme potrebovali dostat na server velke mnozstvi motivletu, museli jsme ale kontrolovat jestli nove vytvorene data jeste neexistuji a resit dalsi veci jako PrimaryId na MotiveVersion atd. Filip to vyresil tim, ze najoinoval nekolik tabulek a pak resil situaci pomoci toho, ze nekde vzniklo null. :

# Filipuv script

BEGIN TRAN

--SELECT \* FROM Media.Role

--SELECT \* FROM media.Platform

**Tady vytvori #tempMotivlet**

CREATE TABLE #tempMotivlet

(

             [Id] [int] NOT NULL,

             [OwnerId] [int] NOT NULL,

             [ProductBrandId] [int] NOT NULL,

             [CompanyBrandId] [int] NOT NULL,

             [CategoryId] [smallint] NOT NULL,

             [TelcoId] [tinyint] NOT NULL,

             [MarketId] [tinyint] NOT NULL,

             [GenderId] [tinyint] NOT NULL,

             [PlatformId] [tinyint] NOT NULL,

             [RoleId] [tinyint] NOT NULL,

             [ProductDetailId] [smallint] NOT NULL,

             [Created] [datetime] NOT NULL ,

             [CreatedBy] [tinyint] NOT NULL,

             [Modified] [datetime] NULL,

             [ModifiedBy] [tinyint] NULL

)

**Vlozi do nej ze skutecne tabulky Motivlet jen zaznamy ktere odpovidaji Filtrum.**

INSERT INTO #tempMotivlet

        ( Id ,

                    OwnerId,

          ProductBrandId ,

          CompanyBrandId ,

          CategoryId ,

          TelcoId ,

          MarketId ,

          GenderId ,

          PlatformId ,

          RoleId ,

          ProductDetailId ,

          Created ,

          CreatedBy ,

          Modified ,

          ModifiedBy         )

SELECT Id, OwnerId, ProductBrandId, CompanyBrandId, CategoryId, TelcoId, MarketId, GenderId, PlatformId, RoleId, ProductDetailId, Created, CreatedBy, Modified, ModifiedBy FROM Media.Motivlet

WHERE PlatformId = 3 AND RoleId = 2

**A zmeni jim nejakou propertu podle filtru NewValues (zmenovadlo)**

UPDATE #tempMotivlet SET RoleId = 3

--select \* FROM #tempMotivlet tmp

--     JOIN Media.Motivlet m

--           ON m.OwnerId = tmp.OwnerId AND

--           m.CompanyBrandId = tmp.CompanyBrandId AND m.ProductBrandId = tmp.ProductBrandId AND m.ProductDetailId = tmp.ProductDetailId AND m.CategoryId = tmp.CategoryId

--           AND m.RoleId = tmp.RoleId AND m.MarketId = tmp.MarketId AND m.GenderId = tmp.GenderId AND m.PlatformId = tmp.PlatformId

--     JOIN Media.MotiveVersionToMotivlet mvm

--           ON mvm.MotivletId = tmp.Id

--     LEFT JOIN Media.MotiveVersionToMotivlet mvm2

--           ON mvm2.MotivletId = m.Id

--     WHERE mvm2.MotivletId IS NULL

--select \* FROM #tempMotivlet tmp

--     JOIN Media.Motivlet m

--           ON m.OwnerId = tmp.OwnerId AND

--           m.CompanyBrandId = tmp.CompanyBrandId AND m.ProductBrandId = tmp.ProductBrandId AND m.ProductDetailId = tmp.ProductDetailId AND m.CategoryId = tmp.CategoryId

--           AND m.RoleId = tmp.RoleId AND m.MarketId = tmp.MarketId AND m.GenderId = tmp.GenderId AND m.PlatformId = tmp.PlatformId

--     JOIN Media.MotiveVersionToMotivlet mvm

--           ON mvm.MotivletId = tmp.Id

--     LEFT JOIN

--     (

--           select tmp.Id FROM #tempMotivlet tmp

--     JOIN Media.Motivlet m

--           ON m.OwnerId = tmp.OwnerId AND

--           m.CompanyBrandId = tmp.CompanyBrandId AND m.ProductBrandId = tmp.ProductBrandId AND m.ProductDetailId = tmp.ProductDetailId AND m.CategoryId = tmp.CategoryId

--           AND m.RoleId = tmp.RoleId AND m.MarketId = tmp.MarketId AND m.GenderId = tmp.GenderId AND m.PlatformId = tmp.PlatformId

--     JOIN Media.MotiveVersionToMotivlet mvm

--           ON mvm.MotivletId = tmp.Id

--     LEFT JOIN Media.MotiveVersionToMotivlet mvm2

--           ON mvm2.MotivletId = m.Id

--     WHERE mvm2.MotivletId IS NULL

--     ) intr

--     ON intr.Id = tmp.Id

--     WHERE intr.Id IS NULL

**Updatne vazebni tabulku. Vysvetleni jak zjisti ktere MotivletId ma dat ktere versne je pod scriptem**

--UPDATUJU VAYEBNI TABULKU NA NOVE MOTIVLETY

UPDATE mvm SET mvm.MotivletId = m.Id FROM #tempMotivlet tmp

       JOIN Media.Motivlet m

             ON m.OwnerId = tmp.OwnerId AND

             m.CompanyBrandId = tmp.CompanyBrandId AND m.ProductBrandId = tmp.ProductBrandId AND m.ProductDetailId = tmp.ProductDetailId AND m.CategoryId = tmp.CategoryId

             AND m.RoleId = tmp.RoleId AND m.MarketId = tmp.MarketId AND m.GenderId = tmp.GenderId AND m.PlatformId = tmp.PlatformId

       JOIN Media.MotiveVersionToMotivlet mvm

             ON mvm.MotivletId = tmp.Id

       LEFT JOIN Media.MotiveVersionToMotivlet mvm2

             ON mvm2.MotivletId = m.Id

       WHERE mvm2.MotivletId IS NULL

**A smaze vazby**

--VAZBY, KTERE NEBYLO MOZNE VYTVORIT, PROTOZE JIZ EXISTUJI, MAZU

DELETE mvm FROM #tempMotivlet tmp

       JOIN Media.Motivlet m

             ON m.OwnerId = tmp.OwnerId AND

             m.CompanyBrandId = tmp.CompanyBrandId AND m.ProductBrandId = tmp.ProductBrandId AND m.ProductDetailId = tmp.ProductDetailId AND m.CategoryId = tmp.CategoryId

             AND m.RoleId = tmp.RoleId AND m.MarketId = tmp.MarketId AND m.GenderId = tmp.GenderId AND m.PlatformId = tmp.PlatformId

       JOIN Media.MotiveVersionToMotivlet mvm

             ON mvm.MotivletId = tmp.Id

       LEFT JOIN

       (

             select tmp.Id FROM #tempMotivlet tmp

       JOIN Media.Motivlet m

             ON m.OwnerId = tmp.OwnerId AND

             m.CompanyBrandId = tmp.CompanyBrandId AND m.ProductBrandId = tmp.ProductBrandId AND m.ProductDetailId = tmp.ProductDetailId AND m.CategoryId = tmp.CategoryId

             AND m.RoleId = tmp.RoleId AND m.MarketId = tmp.MarketId AND m.GenderId = tmp.GenderId AND m.PlatformId = tmp.PlatformId

       JOIN Media.MotiveVersionToMotivlet mvm

             ON mvm.MotivletId = tmp.Id

       LEFT JOIN Media.MotiveVersionToMotivlet mvm2

             ON mvm2.MotivletId = m.Id

       WHERE mvm2.MotivletId IS NULL

       ) intr

       ON intr.Id = tmp.Id

       WHERE intr.Id IS NULL

DROP TABLE #tempMotivlet

rollback

# Postup

1. Vytvorimet tabulku. Tato #temp tabulka je na serveru vytvorena jenom v ramci jedne sesny (session) . Neni videt mezi ostatnimi tabulkami a po skonceni se same smaze, neni teda nutne se starat o jeji mazani.
2. Abychom takovou tabulku mohli vubec pouzit potrebujeme si vytvorit **mapovaci tridu pro tuto fiktivni entitu**. Normalne takova tabulka nema Id ale tady ho potrebuju potrebuju:

[TableName(Name = "#tempMotivlet")]

public class TempMotivlet

{

public int Id { get; set; }

[Nullable]

public int? OwnerId { get; set; }

... atd

1. Tuto tabulku musim pridat na Model :

public Table<TempMotivlet> TempMotivlet => GetTable<TempMotivlet>();

1. Vytvorime tempTabulku na MotivletDao:

public void CreateTempMotivletTable()

{

using (var model = CreateDbContext())

{

model.SetCommand(@"CREATE TABLE #tempMotivlet

(

[Id][int] NOT NULL,

[OwnerId][int] NOT NULL,

[ProductBrandId][int] NOT NULL,

[CompanyBrandId][int] NOT NULL,

[CategoryId][smallint] NOT NULL,

[TelcoId][tinyint] NOT NULL,

[MarketId][tinyint] NOT NULL,

[GenderId][tinyint] NOT NULL,

[PlatformId][tinyint] NOT NULL,

[RoleId][tinyint] NOT NULL,

[ProductDetailId][smallint] NOT NULL,

[Created][datetime] NOT NULL,

[CreatedBy][tinyint] NOT NULL,

[Modified][datetime] NULL,

[ModifiedBy][tinyint] NULL

)

");

model.ExecuteNonQuery();

}

}

1. Motivlety ktere do no vlozime jiz maji zmenene property. Pouzijeme InsertBatch, ten insertuje rychleji pomoci streamu.

UpdateFirstValuePair(dictionary);

private void UpdateFirstValuePair(Dictionary<Motivlet, List<TempMotivlet>> list)

{

**m\_motivletDao.TempMotivletInsertBatch(list);**

}

public void TempMotivletInsertBatch(IEnumerable<TempMotivlet> motivlets)

{

using (var model = CreateDbContext())

{

model.InsertBatch(motivlets);

}

}

1. A rovnou v predchozi metode updatujeme:

m\_motivletDao.UpdateMotivlets();

Tady se da pomoci Linq pomerne snadno napsatu Update s Joinem:

public void UpdateMotivlets()

{

using (var model = CreateDbContext())

{

var query = from mot in model.Motivlet Z tabulky Motivlet

join temp in model.TempMotivlet on mot.Id equals temp.Id vyber jen ty ktere maji Id jako v #temp (uz tady mame mnohem mene polozek)

select new {Mot = mot, Temp = temp}; a udelej anonymni typ.

query.Set(d => d.Mot.OwnerId, x => x.Temp.OwnerId) Updatni hodnoty v tabulce Motivlet podle hodnot z #temp tabulky.

.Set(d => d.Mot.CompanyBrandId, x => x.Temp.CompanyBrandId)

.Set(d => d.Mot.ProductBrandId, x => x.Temp.ProductBrandId)

.Set(d => d.Mot.ProductDetailId, x => x.Temp.ProductDetailId)

.Set(d => d.Mot.CategoryId, x => x.Temp.CategoryId)

.Set(d => d.Mot.RoleId, x => x.Temp.RoleId)

.Set(d => d.Mot.MarketId, x => x.Temp.MarketId)

.Set(d => d.Mot.GenderId, x => x.Temp.GenderId)

.Set(d => d.Mot.PlatformId, x => x.Temp.PlatformId)

.Update();

}

}

**Vysvetleni co to tam ten Filip provadi :D :**

Pripoji se **mvm tabulka**, zase jen pro hodnoty z temp.

A pripoji se znovu **mvm tabulka** tentokrat ale pro hodnoty z tabulky Motivlet.

To je velky rozdil protože tabulky Motivlet a temp muzou vypadat takto (pokud je pouzity alespon jeden LeftJoin): Uplne stejne motivlety mají v těchto tabulkach ruzne Id protože ty které jsou v tabulce temp jsme menili.

|  |  |
| --- | --- |
| Mot | Temp |
| 6 | 2 |
| 7 | 3 |
| 8 | 4 |

V Temp jsou puvodni Id cka (protoze v kodu jsme motivletum menili jen property, nikoli Id ) a v Mot jsou Id-cka motivletu ktere jsme vyhledali na zaklade zmenenych vlastnosti.

Pripojime skutecnou tabulku mvm zalozena na temp.Id = motivletId. To je cislo verze a puvodniho motivletId. (Protoze ve skutecne vazebni tabulce jeste neni nic zmeneno a porad ukazuje ma vazby podle puvodnich motivletu) Tech MVM.VersionId které mají motivlet c.3 muze byt klidne vic to není nic proti nicemu :D

|  |  |  |  |
| --- | --- | --- | --- |
| Mot | Temp | MVM.VersionId | MVM.MotivletId |
| 6 | 2 | 15 | 2 |
| 7 | 3 | 17 | 3 |
| 8 | 4 | 19 | 4 |

Mame motivlet který splnuje vlastnosti na které chceme updatnout , je oznaceny cislem 3 ve sloupci temp( tam jsou uz zmenene motivlety) (Id ma porad puvodni, jen vlastnosti ma nove)

Takova vazba uz urcite existuje, protoze ve zmenovadle jsme hledali duplicity a motivlety kterym jsme zmenili vlastnosti a nenasli jsme takovy jiz existujici motivlet jsme jen updatli.

Co se stane když pripojime ještě jednu tabulku MVM zalozenou ale na motivlet.Id ? (To je Id motivletu ktere chceme zmenit v puvodni vazbe)4. May 20184. May 20184. May 2018

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mot | Temp | MVM.VersionId | MVM.MotivletId | MVM.VersionId | MVM.MotivletId |
| 6 | 2 | 15 | 2 |  |  |
| 7 | 3 | 17 | 3 | 51 | 37 |
| 8 | 4 | 19 | 4 |  |  |

MVM.VersionId a MVM.MotivletId jsou id na puvodnim motivletu => To znamena ze mame skutecne Id verze (17).

Ve Filipovo scriptu se na říká

UPDATE mvm SET mvm.MotivletId = m.Id FROM #tempMotivlet tmp

       WHERE mvm2.MotivletId IS NULL

Ve Filipovo pripade je to pravda protože on udelal script kde vybral všechny polozky z Motivlet a zmenil jim natvrdo nejakou propertu takze vznikali motivlety které ve vazebnich tabulkach nebyly. Takze v jeho pripade mohla nastat situace kde nebyla vazba pro Mot 22. Takze vlastne říká: Zmen MVM.MotivletId v radku, kde je MVM.MotivletId == null. na 22.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mot | Temp | MVM.VersionId | MVM.MotivletId | MVM.VersionId | MVM.MotivletId |
| 6 | 2 | 15 | 2 |  |  |
| 22 | 3 | 17 | 3 | Null | Null |
| 8 | 4 | 19 | 4 |  |  |

Ja ale nikdy mit null nebudu takze reknu: zmen MVM.MotivletId v radku, kde MVM.MotivletId != MVM.MotivletId na hodnotu Mot z tohoto radku. Jinymi slovy :

var query = from mot in model.Motivlet

join temp in model.TempMotivlet on mot.Id equals temp.Id

select new {Mot = mot, Temp = temp};

query.Set(d => d.Mot.OwnerId, x => x.Temp.OwnerId) . . .

**A je to .**

**Ted se pokusim o to same s vazebni tabulkou. Mame kolekci MotiveVersiontToMotivlet entit (uz zmenene). Vazby na ktere chceme menit uz existuji.**