



IEGULDĪJUMS TAVĀ NĀKOTNĒ!

Eiropas Reģionālās attīstības fonds

Prioritāte: 2.1. Zinātne un inovācijas

Pasākums: 2.1.1. Zinātne, pētniecība un attīstība

Aktivitāte: 2.1.1.1. Atbalsts zinātnei un pētniecībai

Projekts: "Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem"

Projekta sākuma datums: 2014.gada 1.janvāris.

Projekta beigu datums: 2015.gada 30.jūnijs.

Līguma Nr. 2013/0031/2DP/2.1.1.1.0/13/APIA/VIAA/010

ESF finansējuma saņēmējs: SIA, SWH SETS

Sadarbības partneris: Elektronikas un datorzinātņu institūts (EDI)

Projekta aktivitātes Nr.3.6 "Savienošanas (join) meta modelis" progresa pārskats

Pārskats Nr.25. par periodu no 2014.gada 1.jūlija līdz 2014.gada 31.decembrim.

SATURS

1.	Kopsavilkums	3
2.	Ievads	4
3.	Tabulu kombinēšana	5
3.1.	SQL valodā	5
3.1.1.	Inner join	5
3.1.2.	Cross join	5
3.1.3.	Outer joins	5
3.2.	Tabulu kombinēšana Entity Framework	6
3.3.	Prasības Join metamodelim	7
4.	Join metamodelis tehnoloģiskajam modelim	8
4.1.	Join	8
4.2.	Jpart	9
4.3.	JCondition	9
4.4.	JoiningPair	9
5.	Rezultāti	10
6.	Literatūras saraksts	11

1. Kopsavilkums

Pārskata periodā (2014-07-01 – 2014-12-31.) projekta „Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem” aktivitātes Nr.3.6 "Savienošanas (join) meta modelis" ietvaros veikti šādi darbi:

1. Tabulu kombinēšanas SQL valodā izpēte.
2. Tabulu kombinēšana Entity Framework izpēte.
3. Prasību analīze Join metamodelim.
4. Tehnoloģiskā modeļa Join metamodelim izstrāde.
5. Join metamodeļa paraugģeneratora izstrāde.
6. Aktivitātes pētnieciskā darbība apspriesta ik nedēļas projekta semināros.

2. Ievads

Šis pārskats ir veltīts projekta apakšaktivitātes Nr.3.6 "Savienošanas (join) meta modelis" ietvaros paveiktajam.

Pārskata trešā nodaļā ir aprakstīta tabulu kombinēšana SQL valodā un Entity Framework, kā arī ir izklāstītas Join metamodeļa prasības. Savukārt, pārskata ceturtā nodaļa veltīta tehnoloģiskajam modelim Join metamodelim.

3. Tabulu kombinēšana

3.1. SQL valodā

Datubāzu vadību sistēmas ir veidotas tā, lai varētu darboties ar ierakstu kopām un lai varētu veikt ierakstu kombinēšanu (join) no dažādām tabulām. Ierakstu kombinēšanas rezultātā rodas rakstu kopa, kuru var izmantot kā tabulu vai arī noglabāt kā tabulu. Šāda pieeja ir efektīva, jo tabulu kombinēšana notiek uz datubāzes servera datubāzes līmenī.

SQL valodā tabulu kombinēšana ir viena no valodas pamatkonstrukcijām. Divas tabulas var kombinēt, izmantojot kombinēšanas nosacījumu. SQL ANSI standarts paredz 5 tabulu kombinēšanas veidus: inner join, cross join, left outer join, right outer join un full outer join [1].

3.1.1. Inner join

Inner join atlasa tos un tikai tos ierakstus no tabulām, kur abu tabulu ieraksti atbilst kombinēšanas nosacījumiem.

SQL ANSI standartā ir sekojoša Inner join sintakse:

```
SELECT <kolonu saraksts>  
FROM <kreisās puses tabula>  
[INNER] JOIN <labās puses tabula>  
ON <kombinēšanas nosacījums>
```

To pašu rezultātu iegūst abas tabulas rakstot FROM nosacījumā un nosacījumu pie **WHERE**:

```
SELECT <kolonu saraksts>  
FROM <kreisās puses tabula>,<labās puses tabula>  
WHERE <kombinēšanas nosacījums>
```

3.1.2. Cross join

Cross join rezultāts satur visas iespējamās tabulu rindiņu kombinācijas t.i. tabulu Dekarta reizinājumu. SQL ANSI standartā ir sekojoša cross join sintakse:

```
SELECT <kolonu saraksts>  
FROM <kreisās puses tabula>  
JOIN <labās puses tabula>
```

To pašu rezultātu iegūst abas tabulas rakstot FROM nosacījumā.

```
SELECT <kolonu saraksts>  
FROM <kreisās puses tabula>,<labās puses tabula>
```

3.1.3. Outer joins

Outer join atlasa ierakstus no abām tabulām, kas apmierina nosacījumu un no vienas vai abām tabulām atlasa arī tos ierakstus, kuriem kombinēšanas nosacījumā otras tabulas kolonas vērtība ir NULL. SQL ANSI standartā ir sekojoša outer join sintakse:

Left outer join:

```
SELECT <kolonu saraksts>
FROM <kreisās puses tabula>
LEFT [OUTER] JOIN <labās puses tabula>
ON <kombinēšanas nosacījums>
```

Right outer join:

```
SELECT <kolonu saraksts>
FROM <kreisās puses tabula>
RIGHT [OUTER] JOIN <labās puses tabula>
ON <kombinēšanas nosacījums>
```

Full outer join:

```
SELECT <kolonu saraksts>
FROM <kreisās puses tabula>
FULL [OUTER] JOIN <labās puses tabula>
ON <kombinēšanas nosacījums>
```

Ir DBVS, piemēram, ORACLE, kurās pastāv alternatīva Outer join sintakse.

3.2. Tabulu kombinēšana Entity Framework

Tabulu kombinēšanai Entity Framework izmanto LINQ to Entities. [2]

Join sintakse LINQ ir līdzīga kā SQL [3], bet LINQ ir vairāki ierobežojumi. LINQ nav iespējas veidot right outer join un full outer join. Vispārīgā gadījumā join nosacījumā var būt patvaļīgi nosacījumi (vienāds, lielāks, mazāks, lielāks vai vienāds, mazāks vai vienāds), taču LINQ join nosacījumos pieļauj tikai vienādības (equal join), kas ir tipiskais tabulu kombinēšanas paņēmiens.

Tā izskatās inner join piemērs:

```
var innerJoinQuery =
    from category in categories
    join prod in products on category.ID equals prod.CategoryID
    select new { ProductName = prod.Name, Category = category.Name };
```

Left outer join rakstīšanai iznanto funkciju DefaultIfEmpty:

```
var leftOuterJoinQuery =
    from category in categories
    join prod in products on category.ID equals prod.CategoryID into
    prodGroup
    from item in prodGroup.DefaultIfEmpty(new Product { Name =
    String.Empty, CategoryID = 0 })
    select new { CatName = category.Name, ProdName = item.Name };
```

Gadījumā ja tabulai ir saliktās atslēgas (composite keys), tabulu kombinēšana var notikt arī salīdzinot vairākus lauku pārus no abām tabulām. [4]

Join piemērs ar ko

```
var query = from o in db.Orders
            from p in db.Products
            join d in db.OrderDetails
            on new {o.OrderID, p.ProductID} equals new {d.OrderID,
            d.ProductID} into details
```

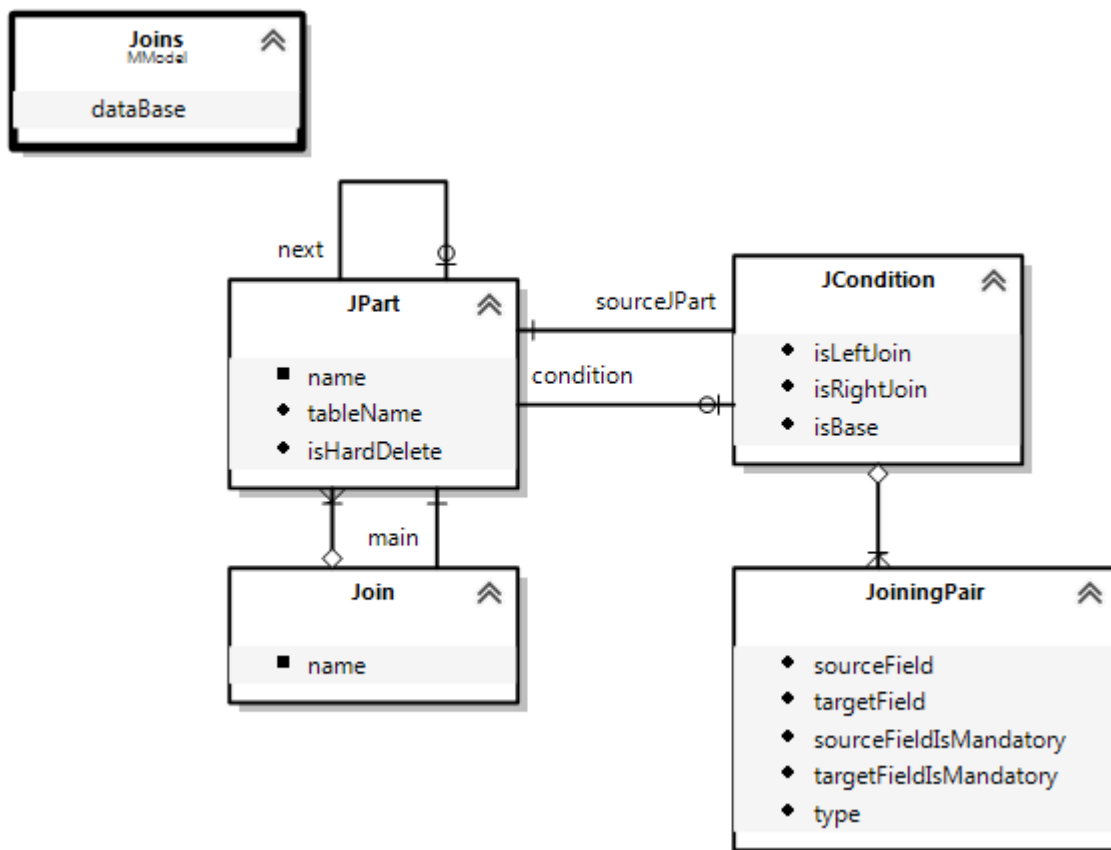
```
from d in details
select new {o.OrderID, p.ProductID, d.UnitPrice};
```

3.3. Prasības Join metamodelim

Lai arī konkrētajā datubāzes pieejas realizācijā (Entity Framework) [5] netiek atbalstīts right outer join un full outer join, Join metamodelim būtu jābūt tādām, lai ar tā palīdzību varētu aprakstīt visas SQL ANSI standarta kombinēšanas pamatkonstrukcijas, izmantojot tabulu kombinēšanu ar vienādībām, lai izvēloties citu realizācijas vidi, būtu iespējams izmantot minētās konstrukcijas. Metamodelim vajadzētu atbalstīt join nosacījumus ar vienādības nosacījumu, tajā skaitā nosacījumus, kas izmanto saliktās atslēgas.

4. Join metamodelis tehnoloģiskajam modelim

Zīmējumā redzams join metamodelis tehnoloģiskajam modelim ar vārdu Joins. Izmantojot join metamodeli, modelī var aprakstīt datu bāzu tabulu kombinēšanu.



Zīmējums 1

Metamodelim ir metaatribūts:

Vārds	Datu tips	Apraksts
dataBase	string	Datubāzes vārds.

Tālāk tiks aprakstīta katra metaentīcijas instance un tai piederošo metaatribūtu un metasaišu instances.

4.1. Join

Join reprezentē datu bāzes tabulu kombinēšanu.

Vārds	Datu tips	Apraksts
name	string	Join vārds .
main		Norāde uz galveno JPart - saturīgi norāde uz tabulu, ar kuru tiek sākta tabulu kombinēšana.

4.2. Jpart

JPart reprezentē datu bāzes tabulu, kura tiek izmantota datu bāzes tabulu kombinēšanā.

Vārds	Datu tips	Apraksts
name	string	JPart vārds.
tableName	string	Tabulas vārds datubāzē.
isHardDelete	bool	Ja „true”, JPart atbilstošajai tabulai tiek veikta fiziska ierakstu dzēšana, pretējā gadījumā dzēšot ierakstu tabulā, tas netiek fiziski dzēsts, bet gan atzīmēts kā dzēsts.
next		Norāde uz nākošo JPart daļu – faktiski norāde uz tabulu, kura tiks pievienota kā nākošā.
condition		Norāde uz JCondition (kombinēšanas nosacījumu)

4.3. JCondition

JCondition reprezentē nosacījumu, kurš tiek izmantota datu bāzes tabulu kombinēšanā.

Vārds	Datu tips	Apraksts
isLeftJoin	bool	Ja vērtība ir “true”, kombinē kā Left outer join.
isRightJoin	bool	Ja vērtība ir “true”, kombinē kā Right outer join.
isBase	bool	Ja vērtība ir “true”, kombinēšanas nosacījums kombinē ar bāzes tabulu. [6]
sourceJPart		Norāde uz kreisās puses Jpart (Tabulu, kurai pievieno).

Piezīme. Ja isLeftJoin = “true” un Ja isRightJoin = “true”, tas nozīmē, ka kombinēšanas nosacījums kombinē kā full outer join.

4.4. JoiningPair

JoiningPair reprezentē nosacījuma daļu kas atbilst vienāds operatoram. Tā kā viens JCondition var saturēt vairākus JoiningPair, metamodelis atbalsta kombinēšanu, izmantojot saliktās atslēgas.

Vārds	Datu tips	Apraksts
sourceField	String	Kreisās puses tabulas lauka vārds
targetField	String	Labās puses tabulas lauka vārds
sourceFieldIsMandatory	Bool	“true”, ja kreisās puses tabulas lauks tabulā ir obligāts
targetFieldIsMandatory	Bool	“false”, ja kreisās puses tabulas lauks tabulā ir obligāts
type	String	Nosacījumā esošo lauku datu tips

Izmantojot Join metamodeli, tiks ģenerētas klases objektu kombinēšanas atbalstam.

5. Rezultāti

Aktivitātes ietvaros tika izpētīts tabulu kombinēšana valodā SQL un Entity Framework un izstrādāts Join metamodeļa paraugģenerators.

6. Literatūras saraksts

- [1] Join (SQL) http://en.wikipedia.org/wiki/Join_%28SQL%29
- [2] Query Expression Syntax Examples: Join Operators - <http://msdn.microsoft.com/en-us/library/vstudio/bb896266%28v=vs.110%29.aspx>
- [3] join clause (C# Reference) <http://msdn.microsoft.com/en-us/library/bb311040.aspx>
- [4] How to: Join by Using Composite Keys (C# Programming Guide) <http://msdn.microsoft.com/en-us/library/bb907099.aspx>
- [5] Nr. 3.4 "MS Entity Framework integrācija"
- [6] Nr. 3.3 "Datu bāzes meta modeļa izstrāde un pētniecība, kas ietver dažādu datu bāzu vadības sistēmu atbalsta izpēti"