Úvod

Funkce v šablonách

Konstanty v šablonách

Promìnné v šablonách

COM rozhraní

Poøadí volání šablon

Rozdíl mezi makrem a šablonou.

Funkce v šablonách a makrech

ForTable Projede všechny tabulky.

ForAlterKey Projede všechny alternativní klíèe, které patøí aktuálni tabulce.

ForCol Projede všechny slupce, které patøí aktuální tabulce.

ForPkCol Projede všechny sloupce tvoøící primární klíè, které patøí aktuální tabulce.

ForPFkCol

ForRelPk

ForIndex Projede všechny indexy patøící aktuální tabulce.

ForIndexCol Projede všechny položky indexu patøící aktuálnímu indexu

ForParent

ForChild

ForDict Projede všechny uživatelské typy v modelu.

ForTrigger Projede všechny triggry v modelu.

ForProcedure Projede všechny uložené procedury v modelu.

ForView Projede všechny pohledy v modelu.

Template (NazevSablony) Vyhodnotí šablonu s názvem NazevSablony

TemplateToFile(NazevSablony,NazevSouboru) Podobné jako Template, ale výsledek uloží do souboru. Používá se hlavnì v HTML
reportech,

Macro (NazevMakra) Vyhodnotí makro s názvem NazevMakra

If(podminka,vyraz1,vyraz2) Vyhodnotí podmínku, pokud je pravdivá vrátí výraz1, pokud není pravdivá, vrátí výraz2.
pozn.Jde o funkci, vyhodnocuje oba výrazy, protože parametry funkce mají vyšší prioritu než funkce samotná.

Iff(podminka,vyraz1,vyraz2). Podobné jako **if**, ale mnohem efektivnìjší, nejedná se totiž o funkci, ale o makro. Vyhodnocuje jen ten výraz, který se opravdu použije. Používejte tehdy, pokud jsou **"vyraz1"** a **"vyraz2"** složitìjší.

Naprostá vìtšina funkcí má 5 parametrù. Zde je jejich popis.

Dále
Konstanty v šablonách a makrech
Promìnné v šablonách a makrech
Rozdíl mezi makrem a šablonou
5 parametrù..

Poøadí volání šablon a maker a jejich viditelnost

Úvod.

Konstanty v šablonách a makrech

Protože nìkteré znaky v makrech a šablonách mají speciální význam, musíte je nahradit pøíslušnou konstantou.

C1 zavináè **@**

C2 procenta **%**

C3 levá složená závorka **{**

C4 pravá složená závorka **}**

C5 úvozovky **"**

C6 apostrof **'** (vìtšinou není nutno nahrazovat apostrof konstantou)

CR nový øádek

TB tabelátor

pøíklad:

Pokud chcete vygenerovat create table s názvem tabulky u uvozovkách:

```
Create table "t1" (…
```

pak musíte musíte napsat šablonu takto:

```
@fortable("" ,"", "create table %C5%%TableName%%C5%, "", "")
```

nebo

```
@fortable("", "", "create table " + C5 + TableName + C5, "", "")
```

Úvod.

Promìnné v šablonách a makrech

DictName
TableName
EntName
ConstraintPkName
AlterKeyname
AlterKeyConstraintName
AlterKeyKeys
PkChildName
PkChildDefaultValue
PkParentName
AtrName
AtrDescription
AtrIsDict
ColName
ExistPk
AtrPkName
TypSQL
Type
Length
Decimal
NotNull
Check, CheckValue
Def, DefValue
IndexName
IndexExpr
IndexFilter
IndexFilterExist
Unique
Clustered
Desc
IndexColDesc
BeforeScript
AfterScript
UserTrigger
Trigger
TriggerName
Procedure
ProcedureName
View
ViewName
TableStorage
IndexStorage
CreatedDate
ModifiedDate
LDropTableGener
LDropTriggerGener
LTriggersGener
LTriggersUserGener
LBeforeScript
LAfterScript
LPkAsConstraint
LFkAsConstraint

LRefIntegGener
LIndexGener
LDropDomainGener
LDomainGener
LTableGener
LAlterKeysGener
LProceduresGener
LViewsGener
ProjectName
ModelName
AuthorName
Version
Company
DatabaseType
IsParent
IsChild
FRelName
RelName
ParentTableName
ChildTableName
ParentKeys
ChildKeys
ConstraintExist
Constraint,ConstraintCheck
ConstraintDefault
ConstraintAtrUnique
CheckExist
DefaultExist
Default, DefaultValue
DefExist
Def2Exist
Def2,Def2Value
LIndexExist
UniqueAtr
LPkGener
HtmlFileName
HtmlDirectory

Úvod.

Rozdíl mezi makrem a šablonou

V CASE Studiu jsou 2 druhy skriptu, **makro** a **šablona**

Makro se používá, pokud máte složitejší textový výraz s mnoha podmínkami (i vnoøenými). Napø. skript **CreateTable** je je napsán makro, protože je dost složitý,

```
cr+
"Create table %tablename% %tablestorage% ("+
forcol("", "", cr+tb+  if(AtrIsDict,macro(CreateAtrib2),macro(CreateAtrib)) ,
",", "")+
if(existpk and lPkGener, ","+cr+if(lPkAsConstraint and (not
Empty(ConstraintPkName)),"Constraint %ConstraintPkName%","")+ " Primary Key
("+forpkcol("", "", ColName, ",", ")"+cr) , cr )+
");"+cr+showmessage("Table %tablename%")
```

pozn. 4. a 5. øádek je v originále jeden øádek.

Je to v podstatì jeden textový výraz. Podobnì byste to psali ve vìtšinì programovacích jazykù. Všimnìte si výrazu `forcol("", "" .....` Vypadá to jako funkce, ale je tu rozdíl. Je v prioritì parametrù, ve funkci mají parametry obvykle vyšší prioritu než funkce samotná, zde je to naopak.. To znamená, že parametry se pøedávají jako kusy zdrojového kodu, a vyhodnocují se až uvnitø funkce. Napø. ve `forcol` se vyhodnotí parametry pro všechny atributy v modelu. O parametrech více zde.

Uvnitø øetìzce mùžete použít promìnné, v tom se makro podobá šablonì.

pø.

```
"Create table %tablename% %tablestorage% ("  ...
```

se chová stejnì jako

"Create table " + tablename + " " + tablestorage + "("

Pozor, nesmíte do sebe zanoøit 2 øetìzce. Øíïte se barvami. Pokud to jinak nejde, rozdìlte makro na více maker.

Šablona se používá, pokud je výraz jednodušší a obsáhlejší. Je to v podstatì normální text, do kterého jsou vloženy výrazy k vyhodnocení. Napø. skript **CreateTriggerUpdate** je napsán takto:

```
{lEntParUpdTrig or lEntChildUpdTrig}
/* Update trigger for %tablename% */

CREATE Trigger tu_%tablename% for %tablename% @showmessage("Trigger for
%tablename% ")
before update as
@if(lEntParUpdRest or lEntChildUpdRest,"declare variable numrows integer;","")
begin
     @forchild( "", "", template(triggerparentupdate), "", "" )
     @forparent("", "", template(triggerchildupdate), "", "" )
end
^
```

Ve složených závorkách je direktiva, uvnitø je logický výraz. Pokud parser na direktivu narazí, vyhodnotí ji. Pokud je výsledek pravdivý (**true**), parser pokraèuje ve vyhodnocování šablony. Pokud je výsledek nepravdivý (**false**), parser zaène pøeskakovat øádky do té doby, dokud nenarazí na další direktivu pøíp. konec šablony. **Direktiva musí být úplnì vlevo u kraje, jinak je ignorována !!!**.

Mezi znaky % jsou promìnné.

Vše mezi znaky **%** se pro zmìnu chová jako makro, mùžete tam používat textové výrazy. **Nezanoøujte øetìzce !!!**.

Funkce se odlišují od ostatního textu tím, že je pøed ní znak @, zavináè. Uvnitø funkce se už zavináè nepoužívá, jednotlivé
parametry funkcí se chovají jako makra.

Jak se volají

Makro se volá

z jiného makra:
```
macro(NazevMakra)
```

ze šablony
```
@macro(NazevMakra)
```

Šablona se volá

z makra
```
template(NazevSablony)
```

z jiné sablony
```
@template(NazevSablony)
```

Úvod**.**

5 parametrù

Naprostá vìtšina funkcí (hlavnì ty, jejichž název zaèíná na FOR) má 5 parametru.

1. parametr, vyhodnotí se jenom pro prvni objekt.
2. parametr, vyhodnotí se pro všechny objekty kromì prvního.
3. parametr, vyhodnotí se pro úplnì všechny objekty.
4. parametr, vyhodnotí se pro všechny objekty krome posledního.
5. parametr, vyhodnotí se jenom pro poslední objekt.

```
pø:
ForCol("Create table %TableName% (", "", "%ColName% %TypSQL%",",%cr%", ")")
```

Zde máte hodnì zjednodušený pøíklad výrazu pro vytvoøení SQL skriptu Create Table.
V prvním parametru máte text "Create .... Ten se vykoná jen jednou pro první sloupec. Konèí levou závorkou.
Druhý parametr je prázdný.
Ve tøetím parametru se vygeneruje název sloupce (atributu) a datový typ.
Všimnìte si èetvrtého parametru, je v nìm èárka a znak pro konec øádky. Protože se èetvrtý parametr vyhodnotí pro všechny sloupce kromì posledního, bude èárka za všemi sloupci kromì posledního.
V pátém parametru ke uzavírající pravá závorka.

Výsledek mùže vypadat takto:

```
Create table Adresar (
ico char(8),
dic char(15),
jmeno char(30),
prijmeni char(30) )
```

Úvod.

Poøadí volání šablon a maker a jejich viditelnost

Pokud je definováno více šablon nebo maker se stejný názvem, je voláno to s vyšší prioritou. Priority jsou seøazeny takto:

1. **Šablony pro entity.** Jsou uložené v modelu u entity, jsou viditelné jen z této entity.
2. **Šablony pro Model**. Jsou uloženy v modelu, jsou viditelné pouze z modelu.
3. **Uživatelské šablony pro databází**. Jsou uloženy v samostatném souboru, jsou viditelné pro všechny modely dané databáze.
4. **Šablony pro databázi.** Souèást dodávky, jsou viditelné pro všechny modely dané databáze.
5. **Systémové uživatelské šablony.** Jsou uloženy v samostatném souboru, jsou viditelné pro všechny modely.
6. **Systémové šablony.** Souèást dodávky, jsou viditelén pro všechny modely.

Systém hledá nejprve šablonu (makro) s nejvyšší prioritou, pokud ji nenajde, hledá tu s nižší prioritou.

Úvod.

COM rozhraní

IModel Je pøístupné vždy
ITable Je pøístupné uvnitø makra ForTable
IColumn Je pøístupné uvnitø maker ForCol,ForPkCol,ForPfkcol,ForRelPk
IDictType Je pøistupné uvnitø makra ForDict, a také uvnitø maker ForCol, ForPkCol,ForPfkcol,ForRelPk, pokud ma atribut slovníkový typ.
IAlterKey Je pøístupné uvnitø makra ForAlterKey
IIndex Je pøístupné uvnitø makra ForIndex
IIndexCol Je pøístupné uvnitø makra ForIndexCol
IRelation Je pøístupné uvnitø maker ForParent, ForChild
ITrigger Je pøístupné uvnitø maker ForTrigger,ForTriggerr
IProcedure Je pøístupné uvnitø maker ForProcedure a ForProcedurer
IView Je pøístupné uvnitø maker ForView a ForViewr.

Pokud o pøístup k rohzhraní, které v té dobì není pøístupné, vyvolá chybové hlášení a ukonèí zpracování šablon.

Více podrobností o struktuøe rozhraní je v helpu tlb.hlp

Úvod.

ForTable

Prochází všechny tabulky v modelu.
Parametry.

Pø: vypíše všechny tabulky

```
ForTable("","",TableName+cr,"","")
```

Úvod.

ForCol

Prochází všechny atributy v tabulce (entitì). Funguje pouze uvnitø makra ForTable.
Parametry

Pø: Vypíše všechny tabulky a sloupce. Je zde vidìt zanoøení Forcol uvnitø ForTable

```
ForTable("", "", "Table %TableName%"+ForCol("", "", "Col %ColName%", "", ""),
"", "")
```

Úvod.

ForDict

Prochází všechny slovníkové typy v modelu.
Parametry

Pø. vypíše všechny slovníkové typy v modelu:

```
ForDict("","",DictName+cr,"","")
```

Úvod.

ForTrigger

Prochází všechny triggry v modelu.
Parametry

Úvod.

ForPkCol

Prochází všechny atributy, které jsou souèástí primárního klíèe aktuální entity. Funguje pouze uvnitø makra ForTable.
Parametry

Úvod.

ForPFkCol

Prochází všechny atributy, které jsou souèástí primárního nebo cizího klíèe aktuální entity. Funguje pouze uvnitø makra ForTable.
Parametry

Úvod.

ForIndex

Prochází všechny indexy v tabulce (entitì0. Je funkcní jen uvnitø makra ForTable.
Parametry

Úvod.

ForIndexCol

Prochází všechny položky indexu. Je funkèní jen uvnitø makra ForIndex.
Parametry

Úvod.

ForRelPk

Prochází všechny atributy, kterými patøí aktuální relaci. Je funkcní jen uvnitø maker ForParent a ForChild.
Parametry

TableName

Název tabulky. Je pøístupný pouze uvnitø makra ForTable.
Jde o fyzický název, logický název je EntName.

Úvod.

DictName

Název slovníkového typu. Je dostupný v makru ForDict, a také v makrech:
ForCol
ForPkCol
ForPFkcol
ForRelPk
kde je souèástí atributu.

Úvod.

EntName

Název entity. Je pøístupný pouze uvnitø makra ForTable.
Jde o logický název, fyzický název je TableName.

Úvod.