

Nová implementácia dynamického SQL

Kedysi dávno sme sa venovali tvorbe dopytov s použitím ručne písaného kódu bez jazyka SQL. V tejto časti si, naopak, ukážeme, akým spôsobom sa v Caché vyvíjali dopyty, a oboznámime sa s najnovšou implementáciou dynamických dopytov SQL.

Už od prvopočiatkov Caché existovala v ňom možnosť pracovať s príkazmi jazyka SQL zapuzdrenými do makrokódu jazyka ObjectScript, teda nášho natívneho skriptovacieho jazyka. Tento spôsob práce s SQL je stále najrýchlejší pri vykonávaní dopytov, ale má tú necnosť, že príkaz SQL musí byť zostavený už v čase tvorby zdrojového kódu, teda musí byť „zadrôtovaný“ natvrdo. Síce do a zo zapuzdreného SQL možno odovzdávať parametre, ale stále príkazy sa nedajú za behu meniť.

Preto veľmi skoro, súčasne s prvotnou implementáciou objektov v Caché, bola vytvorená systémová trieda **% Library.ResultSet**, ktorá umožňovala definovať tak statické otázky, zapuzdrené do definícií tried, a to pomocou jazyka SQL alebo (ako sme videli v 2. časti nášho seriálu) aj pomocou ObjectScriptu, ako aj dynamické dopyty SQL.

Postup bol a stále je jednoduchý. Vytvoríme inštanciu triedy **% Library.ResultSet** a pseudokonštruktor – metóde **% New ()** odovzdáme reťazec informujúci o tom, že budeme naďalej pracovať nie s vopred definovaným dopytom, ale s dynamicky zostaveným. Hoci takto:

```
SAMPLES>s rs=##class(%ResultSet).%New("%DynamicQuery:SQL")
SAMPLES>s sc=rs.Prepare("select * from Sample.Person")
SAMPLES>s sc=rs.Execute()
SAMPLES>w rs.Next()
1
SAMPLES>w rs.Get("ID")
1
SAMPLES>w rs.Get("Name")
Quince,Edward K.
```

Ak chceme odovzdávať parametre za behu programu, použijeme jednoduchú modifikáciu uvedeného príkladu:

```
SAMPLES>s rs=##class(%ResultSet).%New("%DynamicQuery:SQL")
SAMPLES>s sc=rs.Prepare("select * from Sample.Person where ID=? And Name %STARTSWITH ?")
SAMPLES>s sc=rs.Execute(10,"T")
SAMPLES>w rs.Next()
1
SAMPLES>w rs.Get("ID")
41
SAMPLES>w rs.Get("Name")
Tweed,Imelda D.
```

Jednoduché, prehľadné, ale nie zvlášť výkonné, navyše aj pomerne nepohodlné v porovnaní s prácou s relačnými databázami či s objektmi. Práve vzhľadom na zvýšenie výkonnosti dynamických dopytov boli vytvorené nové triedy na implementovanie dynamických dopytov SQL. Tieto triedy boli začlenené do balíčka **% ResultSet**. Zajímavé sú najmä triedy **% ResultSet.SQL**, **% ResultSet.Static** a triedy používané bránami ODBC a JDBC. Výhoda tohto novšieho API pre dynamické príkazy SQL je podpora kešovania a statických snímok dát.

Vo verzii Caché 2010.1 bola pridaná tretia generácia API pre dynamické SQL, a to prostredníctvom balíčka **% SQL**. A o tejto najnovšej im-

plementácii uvedieme viac podrobností. Najskôr zhrňme základné prínosy tejto najnovšej implementácie oproti skorším:

- Pôvodný **% Library.ResultSet** podporoval prenos iba obmedzeného počtu parametrov.
- **% ResultSet.SQL** umožňoval len príkaz SQL SELECT.
- Pôvodné API neumožňovalo pracovať s viacnásobnými resultsetmi.
- Väčšia rýchlosť vykonania kódu.
- Resultset plne podporuje objekty, teda ak je vráteným prvkom SQL SELECT ukazovateľ na objekt (referenčný atribút), je (voliteľne) inštalovaný a naďalej sa s ním pracuje plne objektovo.
- Nahrádza tak triedy z balíčka **% Library**, ako aj **% ResultSet** a tvorí nové, jednotné rozhranie.

Základná trieda na prácu s novým API je trieda **% SQL.Statement**. Tá obsahuje niekoľko metód na prácu:

Názov	Opis
%Prepare	Na základe zaslaného reťazca SQL pripraví príkaz SQL, reťazec SQL môže byť odovzdaný aj ako pole pozostávajúce z viacerých častí.
%PrepareClassQuery	Pripraví príkaz SQL pre dopyty (SQL) pevne definované v iných triedach.
%Execute	Vykoná príkaz SQL a vráti množinu záznamov ako inštanciu % SQL.StatementResult.
%ExecDirect	Pripraví a zároveň vykoná príkaz SQL, spravidla sa používa na vykonanie uložených procedúr, výsledkom je % SQL.StatementResult.

Ďalšia dôležitá trieda je **% SQL.StatementResult**, opisujúca výsledok operácie SQL. Niektoré z metód sú uvedené v nasledujúcej tabuľke:

Názov	Opis
%GetMetadata	Vracia inštanciu triedy % SQL.StatementMetadata, opisujúcu resultset.
%Nex	Posúva kurzor na ďalší záznam.
%NextOutputValue	Vracia nasledujúcu hodnotu vrátenú uloženou procedúrou.
%NextResult	Vracia nasledujúci resultset, ak existuje.

Použitie uvedených tried ukážeme na jednoduchom príklade:

```
>set statement=##class(%SQL.Statement).%New()
>set sql=3
>set sql(1)="select id,Name,DOB,Home_State"
>set sql(2)="from Sample.Person"
>set sql(3)="where id?"
>set status=statement.%Prepare(.sql)
>set result=statement.%Execute(6)
>do result.%Display() // pomocna metoda
tridy %SQL.StatementResult pro zobrazeni
vysledku
id Name DOB
Home_State
7 Finn,George V. 03/23/1928 MA
8 Hanson,James K. 04/02/1928 VT
9 Klein,Michael X. 06/17/1923 WV
10 Klingman,Brenda U. 07/09/1924 WA
11 Paladino,Rhonda H. 06/29/1923 AR
12 Vonnegut,Nellie K. 03/24/1926 HI
... atd...
```

Komplikovanejšiu ukážku nájdete v priestore názvov SAMPLES, ide o triedu **Sample.ResultSets**, ktorá je súčasťou predvolenej inštalácie Caché od verzie 2010.1.

A na záver: **% SQL.StatementResult** je praktický predstaviteľ tried poskytujúcich dynamické od-bavenia, ako o ňom bolo napísané v predchádzajúcej časti tohto seriálu. Lahko si to môžeme demonštrovať na jednoduchom príklade, opäť nad dátami v priestore názvov SAMPLES:

```
>set statement=##class(%SQL.Statement).%New()
>set statement.%ObjectSelectMode=1
>set sql="select Title, Name, Company from Sample.Employee"
>set status=statement.%Prepare(.sql)
>set result=statement.%Execute()
>do result.%Next(.status) // posun na prvni zaznam
>write !,result.Company.Name // pozor na mala a velka pismena

YoyoSonics Partners
```

Poznámka: Urobte si malú MAC rutinu a z nej uvedený príklad spustíte miesto vykonávania jednotlivých príkazov z príkazového riadka.

DAN KUTÁČ, InterSystems



Ďalšie zaujímavé diely seriálu *Tipy a triky s Caché*, ktoré vychádzajú už od roku 2005, nájdete na stiahnutie tu: <http://www.intersystems.cz/education/university/serialy/tipyTrikyCache.html>