

IMPLEMENTATION OF TEMPORAL DATABASES IN CACHE

Lukáš RYCHNOVSKÝ, Master Degree Programme
Dept. of Information Systems, FIT, VUT
E-mail: xrychn01@stud.fit.vutbr.cz

Supervised by: Dr. Dušan Kolář

ABSTRACT

A common requirement in database applications is the need to handle data with a time dimension. It's very complex problem to write your own temporal support in database systems like Oracle, Informix or MySQL. The most suitable database system for this task seems to be postrelational database Caché developed by Intersystems. This paper covers basic ideas and some problems of implementation.

1 ÚVOD

Temporální databáze jsou založeny na úvaze, že záznamy v databázi nejsou statické, ale mění se v čase. Tato myšlenka přináší do klasického SQL podstatná rozšíření. Lze uchovávat a dotazovat se na data aktuální k libovolnému časovému okamžiku či intervalu v minulosti. Dotazovací jazyk nad temporálními databázemi se označuje jako TSQL2 a je rozšířením SQL.

V rámci mého ročníkového projektu vzniká implementace temporální databáze v objektově orientovaném databázovém systému Caché.

Postrelační databáze Caché spojuje výhody objektového přístupu k datům s klasickým SQL přístupem. Na data lze pohlížet jako na objekty (objektově orientované struktury), které se více blíží reálnému pohledu na svět, ale Caché také umožňuje použití velkého množství nástrojů, které umí komunikovat s databázemi i pomocí jazyka SQL.

Do mého projektu jsem si zvolil právě Caché, pro její snadnou rozšiřitelnost, možnost měnit datové struktury pro ukládání a tvořit vlastní indexační metody.

2 TEORIE

Pro implementaci jsem zvolil valid-time model. V tomto modelu je každý záznam v tabulce doplněn dvěma časovými razítky (Ts a Te), které zaznamenávají začátek a konec platnosti příslušného záznamu. Klíčovým slovem NOW jsou označeny záznamy platné v současnosti. Příklad uložení dává následující tabulka.

<i>ID_Zam</i>	<i>Jméno</i>	<i>Částka</i>	<i>Ts</i>	<i>Te</i>
1	Alice	15000	1.1.2001	NOW
2	Bob	6000	1.7.2002	31.8.2002

Modelovaná situace vypadá tak, že Alice pobírá plat 15000,- od 1.1.2001 dosud a Bob pobíral plat 6000 v době letních prázdnin od 1.7. do 31.8. v roce 2002.

Dotazování do tohoto datového modelu pomocí TSQL2 může vypadat třeba takto: `SELECT Jméno FROM tabulka WHERE VALID_IN('1.1.2002')='true'`. Tento dotaz vrátí všechny záznamy platné k 1.1.2002. Konstrukce `VALID_IN`, `VALID_BETWEEN`, `OVERLAPS`, `BEFORE` a další jsou právě temporálním rozšířením SQL. Dále je také potřeba v klauzuli `SELECT` mít možnost vybrat vlastnosti související s temporálním rozšířením, tedy například `VALID_FROM` nebo `VALID_TO`, označující začátek a konec platnosti vybíraných záznamů. Dalo by se sice použít `SELECT Ts` nebo `SELECT Te`, ale je třeba si uvědomit, že sloupce `Ts` a `Te` jsou systémové a nemusí být možné takto dotazy klást.

2.1 CACHE

Pro implementaci v Caché je třeba nejdříve porozumět ukládání dat. Caché v tomto případě používá tzv. globály. Pro každou perzistentní třídu je definováno několik globálů, do kterých jsou data nějakým způsobem zapsána. Nejčastěji se objekty ukládají tzv. serializované do jednorozměrného pole označeného primárním klíčem. V našem případě tedy například: `^SampleApp.Plat.D(1) = << Alice, 15000 >>`.

Vlastní temporální chování Caché zajistíme předefinováním metod `Save`, `Load` a `Delete`, které na základní úrovni zajišťují ukládání objektových dat. Pro SQL přístup je třeba nadefinovat trigger pro zmíněné operace. Požadovaný efekt pak je při jakékoliv operaci nad datovým globálem (`Plat.D`) obohacen příslušnou operací s historickým (`Plat.H`) a indexovými (`Plat.HT` a `Plat.HA`) globály. Operace `DELETE` nad datovým globálem ukončí platnost historického globálu, operace `INSERT` vloží data do datového i historického globálu a `UPDATE` změní data v datovém globálu, ukončí platnost příslušného historického globálu a vloží nový historický globál s novými daty.

2.2 VLASTNÍ IMPLEMENTACE

S využitím výše uvedených znalostí jsem implementaci temporálních databází navrhnul následujícím způsobem. Data jsou uložena v celkem 5 globálech.

- Datový globál `.D` má strukturu kompatibilní se standardním Caché globálem.
- Historický globál `.H` obsahuje kromě všech dat také obě časová razítka a `idH`, jednoznačné v rámci tohoto globálu (je umístěno na druhé pozici). Časové razítko `Te`, označující konec platnosti, je v tomto globálu součástí identifikátoru.

```
^SampleApp.Plat.H(2, '31.8.2002') = << '1.7.2002', 2, Bob, 6000 >>
```

- Indexový globál `.HT` je zaveden pro rychlejší zpracování dotazů s klauzulí `SELECT VALID_FROM` a `VALID_TO`. Jako identifikátor je zvolen `idH` odvozený v globálu `.H`. Globál samotný pak obsahuje časová razítka `Te` a `Ts` a také původní (v tomto případě nejednoznačné) `id`.

```
^SampleApp.Plat.HT(2) = << 2, '31.8.2002', '1.7.2002' >>
```

- Indexový globál .HA je definován pro rychlý UPDATE záznamů. Globál .HA obsahuje vazbu mezi id a aktuálním idH.

`^SampleApp.Plat.HA(2) = << 2 >>`

- Indexový globál .HX je kompatibilní se standardním indexovým Caché globálem.

Dalším krokem je nadefinování metody RetrieveData, která je volána při získávání dat z globálů. Při zpracování dotazu je pak místo standardního datového globálu .D procházen globál .H. Metody VALID_IN a VALID_BETWEEN u příslušného záznamu v .H zkontrolují shodu časových razítek s požadovaným časovým údajem nebo intervalem.

V neposlední řadě je třeba implementovat funkce pro správu temporálního schématu. Měly by umožňovat změnu časového razítka záznamu, smazání či vložení záznamu s platností v minulosti atp. Tyto funkce jsou v projektu nazvány TimeINSERT pro vložení, TimeUPDATE pro změnu platnosti a TimeDELETE pro odstranění záznamu z historického globálu.

2.3 UKÁZKOVÉ DOTAZY

Jméno českého prezidenta, který zastával úřad k 1.1.1970 :

```
SELECT Name FROM Presidents WHERE State="Czech Republic" AND
ValidIn(idH,time)="true", kde time=DisplayToLogical("1970-01-01 00:00:00.000")
```

Výsledek dotazu: Ludvík Svoboda

Jména českých prezidentů a konce jejich funkčních období, jejichž funkční období se překrývá [overlaps | during] s funkčním obdobím Harry S. Trumana

```
SELECT A.Name, ValidTo(A.idH) As ValidTo FROM Presidents A, Presidents B WHERE
Overlaps(A.idH,B.idH)="true" OR During(A.idH,B.idH)="true" AND A.idH <> B.idH AND
B.Name = "Harry S. Truman" AND A.State = "Czech Republic"
```

Výsledek dotazu:

Name: Eduard Beneš ValidTo: 1948-06-07 00:00:00.000

Name: Klement Gottwald ValidTo: 1953-03-21 00:00:00.000

Zjistěte, zda Antonín Novotný zastával úřad před Antonínem Zapotockým

```
SELECT Before(A.idH,B.idH) As Before FROM Presidents A, Presidents B WHERE
A.Name = "Antonin Novotny" AND B.Name = "Antonin Zapotocky"
```

Výsledek dotazu: false

PODĚKOVÁNÍ

Chtěl bych poděkovat Ing. Danielu Kutáčovi z Intersystems za podporu při studiu systému Caché a vedoucímu tohoto projektu Dr. Ing. Dušanu Kolářovi za cenné rady a odkazy na literaturu.

LITERATURA

[1] Kim, W.:Modern Database Systems, New York , Addison Wesley 1995

[2] Online Dokumentace k DBS Caché.