

PROCESSOR OF TSQL2 ON POST-RELATIONAL DATABASES IN ORACLE DATABASE

Jan Szkandera

Master Degree Programme (2), FIT BUT

E-mail: xszkan00@stud.fit.vutbr.cz

Supervised by: Marek Rychlý

E-mail: rychly@fit.vutbr.cz

Abstract: The paper presents the most important concepts of temporal databases and introduces an original interpreter of TSQL2 language. The interpreter supports temporal queries in post-relational multimedia and spatial databases. A possible solution of referential integrity control in temporal databases is also discussed.

Keywords: temporal database, TSQL2, post-relational database, referential integrity

1 ÚVOD

Výzkum na poli temporálních databází probíhá už více než třicet let. Přesto však doposud neexistuje standardizované temporální rozšíření jazyka SQL, ani efektivní implementace temporální databáze, která by našla uplatnění v praxi.

Aplikací, kde je potřeba data skladovat v časovém kontextu, je celá řada — mezi jinými lze jmenovat finančnictví či medicínu. Kromě ryze relačních bych mohl uvést i příklady postrelačních databází využívajících při skladování dat časová hlediska — například lékaři potřebují ukládat obrazová data, jako jsou rentgenové snímky nebo záznamy z magnetické rezonance. U těchto záznamů může docházet k vývoji v čase. Příkladem prostorových databází, které vyžadují temporální podporu, jsou databáze pro uchovávání katastrálních map.

Úkolem tohoto článku je představit návrh interpretu pracujícího nad postrelační databází, který rozšiřuje funkcionalitu této databáze o podporu temporálních dat a dále diskutovat problém kontroly referenční integrity v temporální databázi.

2 JAZYK TSQL2

Richard T. Snodgrass a kolektiv jeho spolupracovníků navrhl jazyk TSQL2 jako temporální rozšíření jazyka SQL [1]. Jde o dotazovací jazyk s podporou temporálních tabulek a dotazů, který zavádí vedle času definovaného uživatelem, jenž je součástí SQL, další dvě nová časová hlediska – **čas platnosti** a **transakční čas**.

O práci s uživatelským časem, jak už název napovídá, se stará sám uživatel – využívá k tomu dotazy stejně jako v jazyce SQL. Zásadním přínosem TSQL2 je čas platnosti. Jde o časové hledisko vyjadřující, kdy je daný záznam v temporální tabulce platný vzhledem k realitě, kterou daná tabulka modeluje. Může se jednat o interval či časový okamžik. Úkolem transakčního času je zachovávat hodnoty záznamů, které byly v databázi uloženy v minulosti – vytváří tak historii tabulky a nachází uplatnění při zálohování a obnově dat.

Pokud tabulka nepodporuje ani jedno z nově zavedených časových hledisek, pak mluvíme o tzv. **snímkové tabulce**. Tabulka podporující pouze čas platnosti se může nazývat buď tabulkou **stavovou** (podporuje intervaly času), nebo tabulkou **událostní** (podporuje časové okamžiky). **Transakční**

tabulka podporuje transakční časové hledisko. Jestliže tabulka podporuje obě nově zavedená časová hlediska, pak se jedná o tabulku **bitemporální**.

Kromě nových časových hledisek se TSQL2 snaží zavést i další funkcionalitu pro práci s časem jako jsou nové datové typy, podpora jiných kalendářů než gregoriánského nebo neomezené rozlišení dvou časových údajů.

Závěrem tedy lze konstatovat, že dvěma hlavními cíli TSQL2 je zjednodušení dotazů nad temporálními daty a rozšíření temporální funkcionality SQL.

3 TSQL2LIB

V roce 2009 vytvořil Ing. Jiří Tomek v rámci své diplomové práce [2] implementaci temporální databáze s názvem TSQL2lib. Tato databáze umožňuje uchovávání pouze relačních dat a podporuje jak čas platnosti, tak i transakční čas. Ing. Tomek se ve své práci zabýval i kontrolou referenční integrity, kterou však výsledná databáze nepodporuje.

V tomto článku navrhovaný interpret z TSQL2lib vychází, v diplomové práci autor článku tuto databázi rozšíří o podporu multimediálních a prostorových tabulek a bude detailněji diskutovat způsob řešení kontroly referenční integrity v temporální databázi.

4 ARCHITEKTURA SYSTÉMU

Výsledná temporální databáze bude tvořena dvěma hlavními částmi – interpretem a databází Oracle. Dotazy uživatele v jazyce TSQL2 budou pomocí navrhovaného interpretu překládány na sérii dotazů v jazyce SQL tak, aby tato série mohla být předána databázi, ve které bude vyhodnocena a výsledek předán zpět uživateli. Temporální tabulky budou mít pro uživatele skryté sloupce uchovávající informaci o čase platnosti či o transakčním čase daného řádku.

5 PŘEKLAD POSTRELAČNÍCH DOTAZŮ

Práce s postrelačními daty, tedy v případě navrhovaného interpretu multimediálními a prostorovými, je podobná jako u dat relačních. Postrelační data jsou reprezentována pomocí objektových datových typů, nad kterými lze budovat podobné dotazy jako u relačních dat. Objektové typy navíc implementují velkou škálu metod, které slouží například ke zjištění formátu dat nebo velikosti plochy geometrického útvaru. Při syntaktické analýze dotazů nad postrelačními datovými typy je velká rozmanitost metod překážkou, která může být řešena dvěma způsoby.

Prvním způsobem je definovat všechna pravidla pro syntaktickou analýzu vstupních řetězců včetně těch částí, ve kterých se vyskytují multimediální a prostorové syntaktické konstrukce. Pak ovšem bude syntaktická analýza těchto částí probíhat vlastně dvakrát – jednou proběhne při překladu dotazu z jazyka TSQL2 do SQL a podruhé při analýze SQL v databázi, nad kterou je interpret postaven.

Mnohem vhodnějším způsobem překladu se zdá být přístup, kdy postrelační části dotazu interpret nebude nijak překládat a během syntaktické analýzy bude brát jako terminální symboly. Pokud navrhovaný interpret bude podporovat stejné datové typy a metody nad nimi jaké podporuje relační databáze, pak můžeme syntaktickou analýzu těchto částí dotazů v interpretu vynechat. Např. z dotazu `"SELECT r.shape.st_isvalid() from rivers r"` bude část `"r.shape.st_isvalid()"` při syntaktické kontrole vynechána.

6 REFERENČNÍ INTEGRITA

V práci [2] byla diskutována kontrola referenční integrity v temporální databázi. Tento článek tuto diskuzi ještě více rozvíjí jak ze stránky návrhu logické kontroly, tak i její technické realizace.

Kontrola referenční integrity v relační databázi bez logické kontroly temporálních vztahů mezi odkazujícími záznamy je triviální. Prakticky mohou nastat dva případy porušení referenční integrity, a to při vkládání do odkazující tabulky a mazání z odkazované, kde jde pouze o kontrolu výskytu cizích klíčů.

6.1 NÁVRH KONTROLY

Kontrola referenční integrity v temporální databázi je výrazně složitější. Nelze se spoléhat pouze na přítomnost cizího klíče v odkazované tabulce, navíc je potřeba ověřit, zda odkazující záznam leží v časovém intervalu odkazovaného záznamu.

Mějme situaci, kdy záznam ze stavové tabulky odkazuje do jiné stavové tabulky. Nyní v důsledku modifikace záznamu v odkazované tabulce dojde k rozdělení tohoto záznamu na dva s disjunktními intervaly času platnosti. Záznam z odkazující tabulky, který původně odkazoval na jeden řádek, nyní musí odkazovat na dva. Prvním řešením je rozdělení odkazujícího řádku na dva nové. Pokud však uvážíme, že i na odkazující záznam může odkazovat jiný z jiné tabulky, pak dojdeme k závěru, že se tato operace může rekurzivně šířit na velké množství tabulek. Lepším řešením tohoto problému je zavedení nové tabulky, ve které se bude uchovávat informace o referenčním vztahu mezi konkrétními řádky. Tímto způsobem dojde pouze k rozdělení záznamu v tabulce referencí a dále tato změna nebude šířena. Odkazující záznam tak bude odkazovat na dva nové.

Logika transakčních tabulek způsobuje další problém. Tyto tabulky uchovávají historii svých záznamů, tzn. záznamy z těchto tabulek, kromě výjimečných případů, nejsou fyzicky mazány. Pro zachování referenční integrity je tedy nutné zachovat i ty záznamy, se kterými je transakční tabulka v referenčním vztahu. Řešením tohoto problému je zakázání referencí mezi transakčními tabulkami a tabulkami, které toto časové hledisko nepodporují.

6.2 PROSTŘEDKY PRO KONTROLU

Návrh interpretu počítá se dvěma způsoby kontroly referenční integrity. Prvním je kontrola již v průběhu interpretace, kdy samotný interpret ověří, zda záznam, se kterým se pracuje, splní integritní podmínku po jeho modifikaci. Druhý způsob využívá databázových triggerů a uložených procedur. Při vložení, modifikaci nebo smazání řádku tabulky bude automaticky spouštěn trigger nebo uložená procedura, která ověří zachování referenční integrity.

7 ZÁVĚR

Článek čtenáře seznámil se základními koncepty jazyka TSQL2 a temporálních databází. Dále představil návrh interpretu jazyka TSQL2 postaveném nad databázovým systémem Oracle, který počítá s možností uchovávání multimediálních a prostorových dat. V závěru článek diskutuje některé problémy související s kontrolou referenční integrity v temporální databázi a navrhuje jejich řešení.

Tento příspěvek byl podpořen výzkumným záměrem č. MSM 0021630528.

REFERENCE

- [1] Snodgrass, J. R. et al.: *TSQL2 language specification*. ACM SIGMOD Record, 1994, roč. 23, č. 1, s. 65-86, ISSN 0163-5808.
- [2] Tomek, J.: *TSQL2 interpret nad relační databází*. Brno, 2009, 79 s., Diplomová práce, VUT v Brně, Fakulta informačních technologií.