

Univerzální řešení domén v relační databázi

Martin Zíma, Michal Nykl, Martin Dostal

Katedra informatiky a výpočetní techniky,
Západočeská univerzita v Plzni,
Univerzitní 8, 306 14 Plzeň, Česká republika

{zima, nyklm, madostal}@kiv.zcu.cz

Abstrakt. Tento příspěvek je zaměřen na relační databáze a představuje náš inovativní přístup k problematice využívání atributů s výčtovým datovým typem (dále označovány jako doménové atributy). Velkým nedostatkem existujících řešení daného problému je např. nezbytný zásah do navrženého schématu databáze při každém požadavku na přidání nového doménového atributu. Námí navržený způsob správy doménových atributů využívá některé dobré vlastnosti dřívějších řešení, není závislý na zvoleném relačním systému řízení báze dat a nevyžaduje zásah do schématu databáze kromě přidání nového doménového atributu. Navržený způsob správy doménových atributů v relační databázi bude využit v námi řešeném projektu *Zpřístupnění dotazů jazykové poradny v lingvisticky strukturované databázi*.

Klíčová slova: databázové systémy, integritní omezení, doménové atributy.

1 Úvod

Nezbytnou součástí komplexních schémat relačních databází bývají také atributy tabulek, které mohou nabývat pouze hodnoty z omezeného výčtu hodnot. Přesto, že každý atribut je definován svým jménem a doménou zastupující datový typ [2], budeme tyto atributy nazývat *doménové atributy*. Pokud je výčet hodnot doménového atributu neměnný, tak je jeho definice jednoduchá. Prostřednictvím klauzule `CHECK` jazyka SQL vytvoříme pro daný doménový atribut integritní doménové omezení, přičemž hodnoty dané domény definujeme operátorem `IN` [3]. Některé databázové systémy nabízí pro řešení tohoto problému speciální datové typy, např. MySQL zavádí datový typ `ENUM` [4]. Problém ale nastává, pokud chceme doménový atribut přidat do tabulky nebo pokud chceme změnit výčet jeho hodnot. Zde žádné existující řešení není dostatečně univerzální, a proto v tomto článku představujeme náš inovativní přístup ke správě doménových atributů, který umožňuje přidávat do tabulek doménové atributy, aniž by vyžadoval změnu schématu databáze.

V sekci 2 shrneme existující způsoby správy doménových atributů a v sekci 3 představíme vlastní způsob řešení tohoto problému. Na závěr shrneme kladné vlastnosti navrženého způsobu správy doménových atributů a představíme projekt, ve kterém bude tento způsob použit.

*J. Steinberger, M. Zíma, D. Fiala, M. Dostal, M. Nykl (eds.)
Data a znalosti 2017, Plzeň, 5. - 6. října 2017, pp. 116-119.*

2 Existující způsoby správy doménových atributů

Jedním z nejznámějších způsobů správy doménových atributů, který nabízí možnost měnit, přidávat či odebrat položky domény, je tvorba tzv. *číselníků*, či též doménových tabulek. Každý číselník obsahuje minimálně dva atributy: celočíselný primární klíč a (obvykle) textovou hodnotu dané domény. Číselník je na tabulku využívající doménový atribut vázán relací 1:N, přičemž doménový atribut je v tabulce nahrazen cizím klíčem odkazujícím do číselníku. Uvedený přístup má několik výhod: změna hodnoty v číselníku se okamžitě promítá do stávajících dat a stejnou doménu lze použít v jiném doménovém atributu libovolné tabulky.

Nevýhodou uvedeného přístupu je nutnost založení nového číselníku vždy, když přijde požadavek na přidání nové domény, tj. je zde nutný zásah do schématu relační databáze. Nicméně snahou databázových architektů je navrhnout schéma tak, aby výše uvedené změny nevyžadovaly žádný zásah do schématu a vše bylo řešeno jen uloženými daty v databázi.

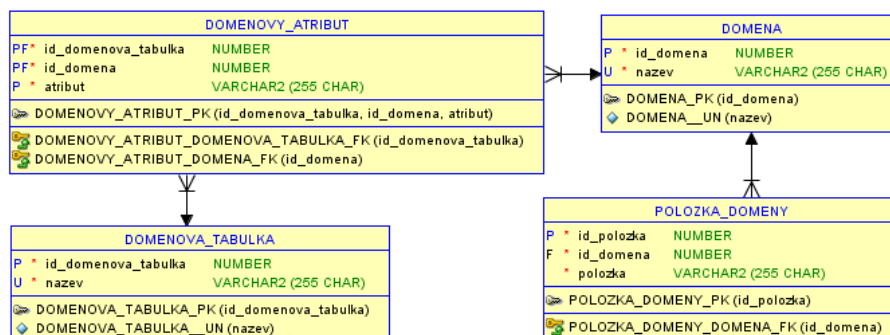
Vlastní způsob správy doménových atributů nabízí také společnost Oracle, která do produktu *Oracle Designer* zabudovala speciální tabulku `CG_REF_CODES` [1]. Ta je navržena tak, aby zahrnuje všechny zakládané domény (výčty hodnot) v databázi, ale není propojena žádnou relací s ostatními tabulkami. Doménový atribut musí být textového datového typu. Při definici doménového atributu tabulky je nezbytné vygenerovat tzv. *table API* pro danou tabulku. *Table API* se skládá z uloženého balíku procedur a funkcí a sady triggerů, které při nastalé události volají metody z tohoto balíku. Jedna z metod hlídá, že daný doménový atribut může nabývat pouze zvolených hodnot z tabulky `CG_REF_CODES`.

3 Navržený univerzální způsob správy doménových atributů

Při řešení větších projektů je stěžejní správný a univerzální návrh schématu relační databáze. Nicméně, i po dokončení návrhu schématu mohou vznikat požadavky na nové doménové atributy, což vede k rozšiřování databáze o nové číselníky. Naším cílem je zabránit rozšiřování schématu relační databáze a přitom uspokojit nové požadavky na doménové atributy. Za tímto účelem jsme navrhli nový způsob práce s doménami, který využívá dobré vlastnosti výše popisovaných přístupů a není závislý na zvoleném databázovém systému.

3.1 Schéma navrženého způsobu správy doménových atributů

Námi navržený univerzální způsob správy doménových atributů využívá vlastní schéma databáze obsahující čtyři tabulky, jak ukazuje obrázek 1. Schéma je vytvořeno nástrojem *Oracle SQL Developer Data Modeler*. Atributy mohou být označeny písmeny P, F či U, pokud je daný atribut prvkem množiny definující integritní omezení. Těmi mohou být primární klíč (P, *primary key*), cizí klíč (F, *foreign key*) a unikátní hodnota (U, *unique*). Hvězdička u atributu označuje integritní omezení NOT NULL.



Obr. 1. Navržené schéma databáze pro podporu doménových atributů

Jak to celé funguje, si ukážeme na příkladu. Při návrhu schématu relační databáze si představme např. tabulku NAHRAVKA, do které chceme vložit doménový atribut FORMAT, který může nabývat pouze hodnot *mp3*, *wma* a *wav*. Dále pro jednoduchost výkladu předpokládejme, že tabulky z obrázku 1 jsou prázdné. Nyní budeme postupovat podle těchto bodů:

1. Do tabulky DOMENOVA_TABULKA vložíme nový záznam o tabulce, tj. NAHRAVKA.
2. Do tabulky DOMENA vložíme nový záznam o doméně, tj. FORMAT_NAHRAVKY.
3. Do tabulky POLOZKA_DOMENY vložíme odpovídající tři hodnoty dané domény, tj. *mp3*, *wma* a *wav*.
4. Do tabulky DOMENOVY_ATRIBUT vložíme nový záznam o tom, ve které tabulce byla použita která doména a jak se daný doménový atribut v tabulce jmenuje, tj. tabulka NAHRAVKA, doména FORMAT_NAHRAVKY a atribut FORMAT.

Nyní víme, že atribut FORMAT tabulky NAHRAVKA je doménovým atributem, ale aktuálně není nikterak zajištěno, že může nabývat pouze hodnot dané domény.

3.2 Validace hodnot doménových atributů

Aby nově přidáný doménový atribut mohl v tabulce nabývat pouze hodnot ze své domény, je nezbytné vytvořit uloženou funkci pro kontrolu vstupu a řádkový trigger pro danou tabulku. Funkce bude mít 4 parametry: název tabulky a jejího doménového atributu, název domény tohoto atributu a do tabulky vkládaná či aktualizovaná hodnota atributu. Trigger při vkládání či aktualizaci hodnot v tabulce dovolí zapsat do doménového atributu pouze hodnotu, kterou obsahuje příslušná doména. K tomu účelu bude volat připravenou funkci, která svou návratovou hodnotou stanoví, zda vkládaná či aktualizovaná hodnota doménového atributu je validní, tj. jedná se o hodnotu správné domény.

4 Závěr

Navržený univerzální způsob správy doménových atributů přebírá dobré vlastnosti existujících řešení a eliminuje vlastnosti nepříznivé. Použití navrženého způsobu je jednoduché, protože je potřeba pouze rozšířit stávající schéma relační databáze o čtyři tabulky a v databázi zkompilevat uloženou funkci. Pro každý doménový atribut je nutné tyto tabulky odpovídajícím způsobem naplnit a současně každému atributu vytvořit v databázi řádkový trigger pro jeho kontrolu. Výhodou námi navržené správy doménových atributů je také skutečnost, že do doménového atributu je ukládána příslušná hodnota místo hodnoty cizího klíče odkazujícího na záznam v číselníku.

Navržený způsob správy domén v relační databázi bude využit v projektu *Zpřístupnění dotazů jazykové poradny v lingvisticky strukturované databázi*, jehož cílem je vytvořit podpůrný nástroj pro pracovníky Jazykové poradny Ústavu pro jazyk český AV ČR. Nástroj by měl usnadnit pracovníkům poradny jejich poradenskou činnost a současně by měl osobám z široké veřejnosti umožnit nalezení správných informací o aktuálních jazykových jevech, které stávající jazykové příručky dosud neobsahují. Tento projekt bude dokončen v roce 2019.

Literatura

1. COXALL, Malcolm a CASWELL, Guy. *Oracle Quick Guides Part 2 - Oracle Database Design*. Cornelio Books, Spain, 2014. ISBN 978-84-940853-5-2
2. CONNOLLY, Thomas a BEGG, Carolyn. *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Addison Wesley, 2004. ISBN 978-0-321-21025-8
3. STEPHENS, Ryan a PLEW, Ron: *Sams teach yourself SQL in 21 days*. 4th ed. Indianapolis, SAMS, 2003. ISBN 978-0672324512
4. The ENUM Type. *MySQL 5.7 Reference Manual* [online]. [cit. 2017-06-07]. Dostupné z: <https://dev.mysql.com/doc/refman/5.7/en/enum.html>

Poděkování: Tento příspěvek byl podpořen grantem Ministerstva Kultury České republiky číslo DG16P02B009 *Zpřístupnění dotazů jazykové poradny v lingvisticky strukturované databázi*.

Annotation:

Universal solution of domains in relational database

This article describes the problem of domain attributes management in relational databases and it describes our innovative approach to this issue. The core problem of other solutions is in a need to extend relational database schema when a new domain attribute has to be added. Our solution is RDMS independent and it not requires modifications in relational database schema. The proposed solution will be used in our project: *Access to a Linguistically Structured Database of Enquiries from the Language Consulting Centre*.