Modelování multidimenzionality pro relační databáze

Jiří Webr, Software AG s.r.o., Vyšehradská 53, 120 00 Praha 2, Česká republika

Abstrakt

Jedním z nejdůležitějších faktorů ovlivňujících využívání datových skladů je správný datový model. Modelování pro datové sklady se v mnohém liší od modelování standardních relačních databází, na jaké jsme byli zvyklí pro operativní databanky. Pro optimalizaci přístupu k datům z hlediska multidimenzionality a sumarizace jsou zcela záměrně porušována některá pravidla pro normální formy. V tomto příspěvku chceme uvést nástin principů modelování multidimenzionality pro relační databanky používané v datových skladech.

1. Úvod

V posledních letech zasáhl trend rozvoje a budování informační technologie i do oblasti datových skladů. Jednou z hlavních podmínek jejich úspěchu je vytvoření správného datového modelu pro velmi náročné analytické procesy (OLAP) poskytující důležité informace pro podporu rozhodování obchodních analytiků a managementu. Pouze optimální uložení dat do struktury umožňující multidimenzionální pohledy a intuitivní manipulace s daty může využít mohutnou sílu datových skladů. Data ukládaná do datových skladů jsou jednak data získaná z různých zdrojů a jednak data vypočítaná (agregovaná, sumarizovaná). Jestliže by se agregace dat prováděla v okamžiku runtime, silně by narůstala doba odezvy, protože by byl veliký počet načítaných tabulek (často i celá databáze) a potřebných joins. Snížení doby odezvy při intuitivních dotazech koncového uživatele u často opakovaných požadavků na přístup k datům, zejména při drilování (drill-down, drill-up, drill-across, drill-within) a "hraní" si s daty, sebou nese potřebu předem vypočítaných sumarizací z různých hledisek. Dalším faktorem silně ovlivňujícím efektivní využívání datových skladů jsou multidimenzionální pohledy na data, tj. pohledy z více hledisek. Nestačí nám již údaje o odbytu určitého výrobku, ale potřebujeme tyto údaje blíže specifikovat např. vzhledem k časovému období nebo regionu. To sebou nese potřebu nějakým způsobem zprostředkovat vztah mezi daty a pohledem na ně. Tento problém multidimenzionality je v současné době řešen dvěma způsoby. Jedním z nich je ukládání dat ve virtuálním multidimenzionálním datovém prostoru, v tzv. hypercube resp. multicube (tj. v jedné nebo více navzájem logicky propojených "datových krychlích") [1]. Jsou to tzv. proprietární multidimenzionální databanky. Jejich nespornou výhodou je velmi rychlý přístup k datům, na druhé straně však mají dost nevýhod (např. malé využití předdefinovaného prostoru, neprůhlednost struktury takovéto databanky a tím i přílišná závislost na dodavateli, náročnost při jakýchkoliv změnách jako přidání agregačního stupně nebo další dimenze, data nebývají standardně přístupná pomocí SQL apod.)

[2], [3]. *Druhým* způsobem je využívání standardních relačních databází a multidimenzionalitu modelovat jiným způsobem. Tento přístup šetří investice, neboť podnik může využívat databázový systém, který má již instalovaný, k datům může přistupovat pomocí SQL a využívat všech služeb, které SŘBD poskytuje. V tomto příspěvku se proto chceme zabývat principy modelování multidimenzionality pro relační databanky.

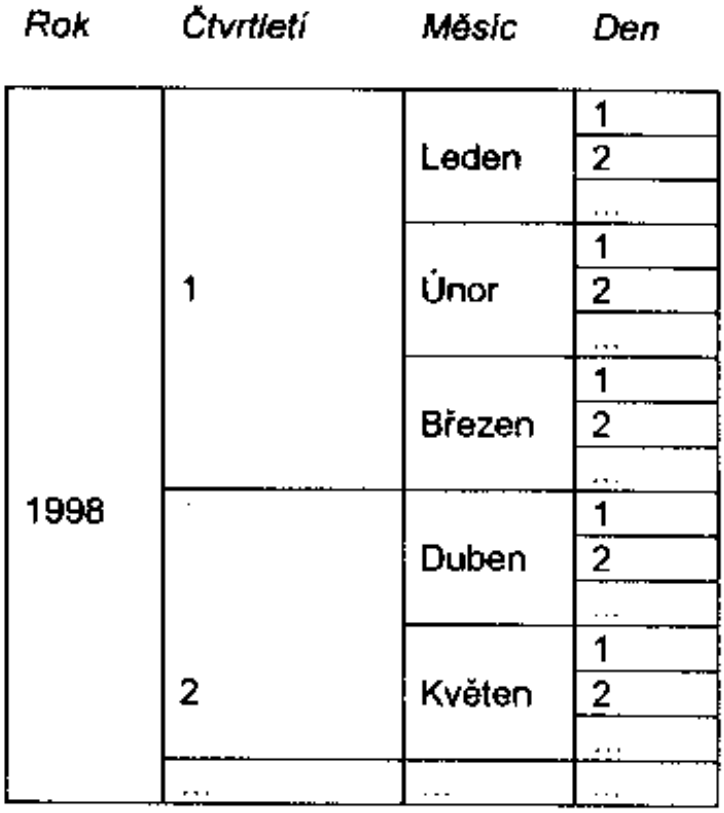
2. Datový model

Při modelování relační databanky pro datové sklady je nutno vzít v úvahu specifika datového modelu z hlediska multidimenzionality a sumarizace dat. Atributy, které se definují pro data, se sdružují do logických skupin nazývaných dimenze, které je kvalifikují na obecné úrovni. Tak např. údaj "Prodej" může být určen na obecné úrovni dimenzemi Produkt, Čas a Území, ale na specifické úrovni je určen kombinací atributů těchto dimenzí, jako např. Datum, Prodejna a Výrobek - viz Obr.1. Sloupec Datum reprezentuje dimenzi Čas, sloupec Položka dimenzi Produkt a sloupec Prodejna dimenzi geografickou - Území. Hierarchie atributů v rámci dimenze - její stromová struktura - je určena požadavky na úroveň detailů, na které je nutné se "provrtat". Každý datový model musí mít alespoň jednu dimenzi, ale jinak počet dimenzí není omezen, a každá dimenze musí mít alespoň jeden atribut.

Datum	Položka	Prodejna	Prodej_ks	Prodej_Kč
5.2.98	Chléb Šumava	Vodička a syn	123	2410,80
5.2.98	Chléb Šumava	Rytina	98	1920,80
5.2.98	Chléb Šumava	Včela	236	4272,80
...	...	...	...	...

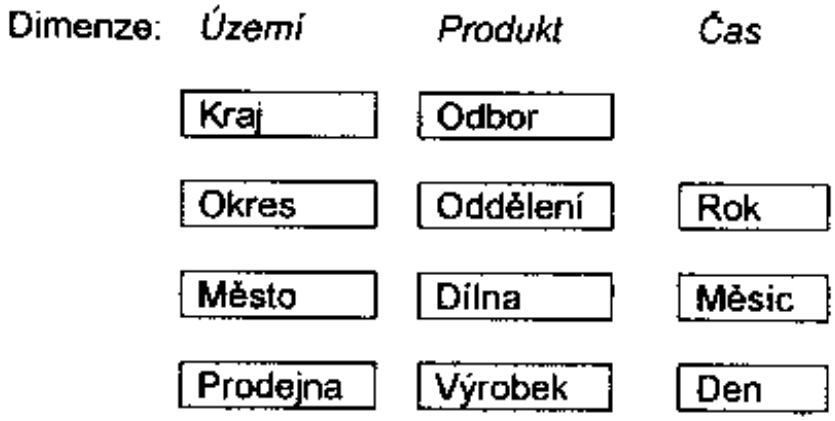
Obr. 1. Tabulka faktů

Pro účely analýzy z hlediska obchodní strategie, např. odbytu výrobků, nebývá tak moc zajímavý odbyt jednotlivého výrobku v určitém dni a v určité prodejně, ale spíše skupiny výrobků v nějakém časovém období (měsíc, čtvrtletí, letní či zimní sezóna, ...) a/nebo geografickém území, (v okrese, kraji apod.). Proto musí být v příslušných rozměrech *Čas, Produkt a Území* definovány příslušné atributy (např. Rok, Čtvrtletí, Měsíc a Den; Prodejna, Město, Okres, Kraj, ...) a jejich hierarchie, tj. vztah "parent-child" - viz příklad pro dimenzi Čas na Obr. 2. Podobně mohou být hierarchie atributů definovány i pro další rozměry. Struktura tohoto hierarchického stromu závisí na tom, do jaké hloubky detailu se budeme chtít "provrtávat", a jaké potřebujeme sumarizace.



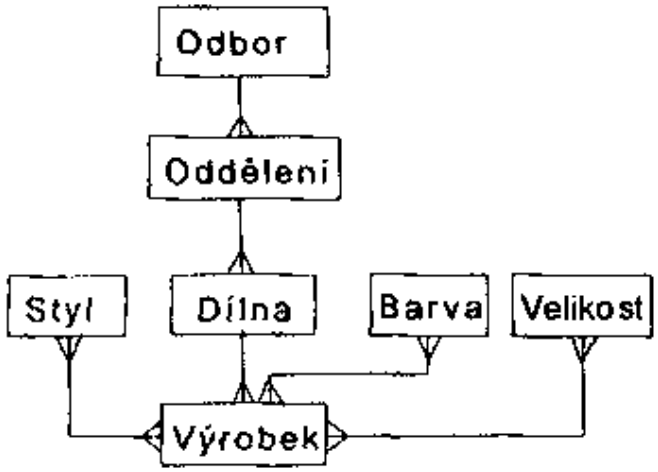
Obr. 2. Hierarchie atributů pro rozměr *Čas*.

Příklad hierarchie atributů všech uvedených dimenzí ukazuje příklad na Obr. 3.



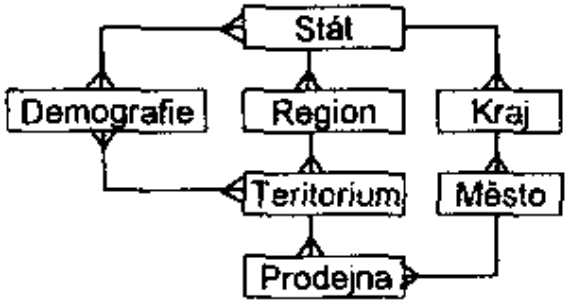
Obr. 3. Hierarchie atributů v jednotlivých dimenzích.

V takovémto znázornění hierarchie atributů v dimenzích je ještě nutné definovat vztahy M:N a vedlejší atributy. V rozměru Produkt mohou být jednotlivé výrobky či jejich skupiny rozčleněny např. podle výrobního organizačního schématu (dílna, oddělení, odbor, ...), které jsou definovány v hierarchii hlavních atributů (Odbor, Oddělení, Dílna). Každý výrobek může existovat v různých velikostech, barvě, stylu apod., které jsou definovány vedlejšími atributy, např. Styl, Barva a Velikost, jak ukazuje Obr. 4.



Obr. 4. Stromová architektura - hlavní a vedlejší atributy.

Stromová struktura atributů umožňuje větvení hlavní hierarchické struktury. To může nastat v případě, že atribut má více "dětí" nebo naopak více "rodičů". Např. na Obr.4 atribut Výrobek má jako rodiče atributy Styl, Barva a Velikost. Možné větvení pro dimenzi Území ukazuje Obr. 5.



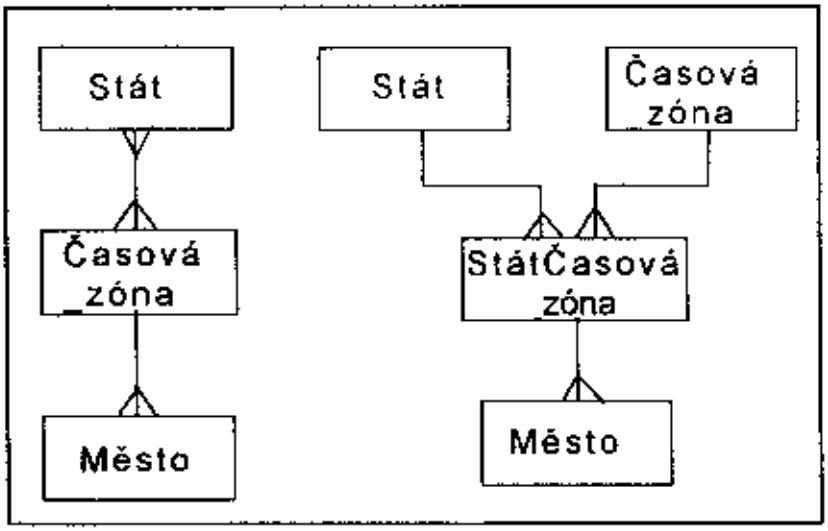
Obr.5. Rozvětvený strom hierarchie atributů.

Mají-li atributy vztah M:1 nebo M:N, mohou být modelovány buďto jako složená struktura atributů nebo jako dva separátní atributy - jeden z nich jako "rodič" a druhý "dítě" - mezi nimiž je vztah M:1 nebo M:N. Pochopit tyto rozdíly je velmi důležité, neboť může docházet k častým nedorozuměním. Objasnění rozdílů mezi oběma možnostmi bude snáze pochopitelné na příkladu. Vezměme si atributy Výrobek a Barva, jejichž vztah je M:N (výrobek může být v různých barvách a naopak různé výrobky mohou mít stejnou barvu).

Předpokládejme nejdříve, že atributy Barva a Výrobek tvoří složený atribut Barevný_výrobek. V tomto případě existují data pro kombinaci výrobku a barvy, a proto mezi atributy Barevný_výrobek a Barva není vztah M:N. Proto je možné zjistit např. prodej všech modrých obleků. Nyní předpokládejme, že atributy Výrobek a Barva mají vztah M:N, kde atribut Barva je "rodičem" atributu Výrobek. V tomto případě se nemůžeme dotazovat na prodej všech modrých obleků, protože datový sklad neobsahuje žádná data na požadované úrovni specifičnosti. Co existuje, to jsou data, která nám ukazují barvy, ve kterých jsou různé výrobky vyráběny. To nám umožňuje

klást např. dotazy typu "Jaký je prodej všech výrobků v modré barvě?". Takovýto dotaz však ukáže i prodej výrobků, které existují ve více barvách.

Jiným příklad je znázorněn na Obr. 6. Předpokládejme, že atributy Stát, Časová_zóna a Město vytvářejí hierarchii znázorněnou v levé části obrázku. Atributy Časová_zóna a Stát mají vztah M:N, neboť v jedné zóně může ležet více států a naopak pro jeden stát může existovat více zón. Vztah mezi atributy Město a Časová_zóna není vztah M:N, protože města nemohou ležet ve více časových zónách. Jestliže nyní atributy Stát a Časová_zóna tvoří složený atribut Stát_Časová_zóna, jak je znázorněno v pravé části obrázku 6, neexistuje mezi atributy Stát a Stát_Časová_zóna a mezi atributy Časová_zóna a Stát_Časová_zóna vztah M:N, a je možné klást dotazy jako např. "Jaký je prodej určitého výrobku ve všech časových zónách v USA". Není možné ovšem klást stejný dotaz, jestliže Stát a Časová_zóna netvoří složený atribut. V tomto případě je možno klást dotazy jako: "Jaký je prodej určitého výrobku v určité časové zóně"- odpovědí bude prodej tohoto výrobku ve všech státech, které spadají do uvedené časové zóny.



Obr. 6. Složená struktura a vztah M:N

3. Tabulky ve schématu Sněhová vločka

Z datového modelu se potom odvíjí modelování multidimenzionality u relačních databázích, které se realizuje strukturou nazvanou "Snowflake" - "Sněhová vločka". V tomto modelu se používají tři druhy tabulek. Tabulky faktů (Fact Tables, Base Tables či Primary Data Tables) obsahují vlastní data vyhledávaná uživatelem ať již v atomární formě nebo sumarizované. Řádky těchto tabulek jsou indexovány složeným primárním klíčem tvořeným kombinací atributů jednotlivých dimenzí. Příklad jednoduché tabulky faktů byl uveden na Obr. 1. "Datum", "Položka" a "Prodejna" označují sloupce atributů a "Prodej_ks" a "Prodej_Kč" označují sloupce faktů. Každý ze tří sloupců atributů se vztahuje na jednu dimenzi v datovém modelu, v tomto případě třírozměrném. Pro jednoznačnou identifikaci hodnot ve sloupcích faktů je nutný jeden atribut z každé dimenze. Na příklad, existuje pouze jeden údaj o prodeji chleba Šumava v prodejně Rytina dne pátého února 1998. Sloupec "Datum" reprezentuje rozměr *Čas* (zde např. na atomární, nejnižší agregační úrovni), sloupec "Položka" rozměr *Produkt* a sloupec "Prodejna" geografický rozměr *Území*.

Druhým typem tabulek jsou Prohledávací tabulky (Lookup Tables) obsahující např. verbální popis kódovaných údajů, a Tabulky dimenzí (Dimensional Tables či Relational Tables) obsahující hierarchie atributů v jednotlivých dimenzích. Oba tyto druhy tabulek jsou "navěšeny" na tabulky faktů. Příklad těchto tabulek je uveden na Obr.7.

Prohledávací tabulka

PROD_ID	PROD_POPIS
1	Bílá Labuť
2	Máj
3	Kotva
4	Prior
5	Brouk&Babka

Tabulka dimenzí

Prod_ID	Město_ID
1	2
2	2
3	2
4	1
5	1

Obr. 7. Příklad prohledávací tabulky a tabulky dimenzí

Vzhledem k tomu, že tyto tabulky se často vztahují ke stejnému atributu v Tabulce faktů, je možné spojovat je do jedné tabulky. Jednoduchý příklad je uveden na Obr. 8.

LOOKUP_PROD

Prod_ID	Prod_popis	Město_ID
1	Bílá Labuť	2
2	Máj	2
3	Kotva	2
4	Prior	1
5	Brouk&Babka	1

Obr. 8. Spojení Prohledávací tabulky a Tabulky dimenzí

Důležité je, že z primárních klíčů tabulek příslušných dimenzí se potom vytváří složený primární klíč v Tabulce faktů, takže přístup k požadovaným datům je velmi rychlý. Na Obr 9 je jako příklad uvedena tabulka faktů z Obr.1, kde místo slovního vyjádření jsou ve sloupcích Datum, Položka a Prodejna použity primární klíče z příslušných tabulek dimenzí tvořící složený klíč v tabulce faktů, např. pro údaj "Prodej chleba Šumava dne 5.2.96 v prodejně Vodička a syn", tj. pro první řádek, bude klíč kombinací klíčů 5, 1 a 2 z příslušných tabulek dimenzí Čas, Výrobek a Území.

Datum	Položka	Prodejna	Prodej_ks	Prodej_Kč
5	1	2	123	2410,80
5	1	3	98	1920,80
5	1	4	236	4272,80
...	...	...	...	...

Obr. 9. Tabulka faktů

4. Závěr

Modelování databanky pro datové sklady je důležitým faktorem pro efektivnost datových skladů a je proto nutné zvládnout jeho principy. Tento příspěvek by měl být prvním krokem pro hlubší studium této problematiky datových skladů. Spolu s optimalizačními technikami ukládání, jako je rozčlenění tabulek a analýza agregací, tvoří základ úspěšného projektování datových skladů.

Literatura

[1] Diepolt J.: Multidimenzionální data. Datové sklady - profily produktů na českém trhu. Systémová integrace, Praha, ČR, č.3, září, 1997

[2] Bannister F. E.: OLAP - A Question of Definition. DATASEM '97, Praha, ČR, říjen, 20-22, 1996

[3] Lhoták M., Benešovský M. Janský P.: Data Warehousing aneb nic nového pod sluncem. DATASEM '97, Praha, ČR, říjen, 20-22, 1996

[4] DSSAgent User Guide, Microstrategy.