VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

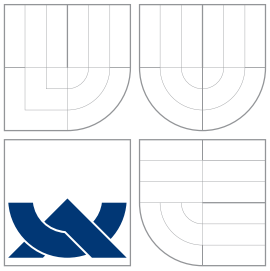
METODY UKLÁDÁNÍ DAT PRO OLAP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

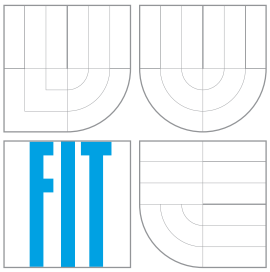
AUTOR PRÁCE
AUTHOR

PETR DITTRICH

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

METODY UKLÁDÁNÍ DAT PRO OLAP

OLAP DATA STORAGE METHODS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR DITTRICH

VEDOUCÍ PRÁCE
SUPERVISOR

Prof. Ing. TOMÁŠ HRUŠKA, CSc.

BRNO 2007

Abstrakt

Tématem bakalářské práce je návrh a implementace jádra OLAP datového skladiště. V úvodu práce je v krátkosti rozebrána problematika datového skladiště, důvody proč v minulosti tato datová skladiště začala vznikat a dále předpoklady jejich praktického uplatnění. V další části jsou popsána některá úskalí, která se vyskytla při implementaci skladiště. Je popsána forma uložení metadat, struktura a tvorba OLAP kostek. V poslední kapitole je na příkladu popsán postup tvorby datové kostky v tomto systému.

Klíčová slova

OLAP, ETL, datové skladiště, dimenze, měřítko, multidimenzionální model

Abstract

The topic of the bachelor work is the OLAP core implementation for the data storage space. All theoretical facts about the data storage space are discussed at the start. The form of metadata elements storage and the OLAP cubes creating are described in the following part. And finally, in the last chapter, it is precisely shown how to create a cube in this system.

Keywords

OLAP, ETL, data warehouse, dimension, measure, multidimensional model

Citace

Petr Dittrich: Metody ukládání dat pro OLAP, bakalářská práce, Brno, FIT VUT v Brně, 2007

Metody ukládání dat pro OLAP

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Prof. Ing. Tomáše Hrušky, CSc..

. . . . . .
Petr Dittrich
9. května 2007

Obsah

Kapitola 1

Úvod

Na konci minulého století došlo k velkému rozvoji informačních technologií. Jako následek tohoto rozvoje dochází k vytváření obrovského množství dat a jejich použití pro řídící a rozhodovací procesy. Tato data můžeme využít k získávání informací, ovšem přesnost těchto informací závisí na množství dostupných dat. Otázka skladování dat se s rostoucími nároky na jejich kvantitu i kvalitu stala zásadním problémem. Tato data můžeme skladovat různě, nejlepší je jejich skladování ve specializovaných datových skladištích.

Datová skladiště jsou jednou ze základních částí informačních systémů. Datová skladiště se velmi rychle rozvíjejí a mění své požadavky nejen z důvodu změn požadavků manažerů a nárůstu množství zpracovávaných dat, ale i z důvodu růstu hardwarového výkonu počítačů, který umožňuje skladování a manipulaci se stále větším množstvím dat.

Data z datových skladišť musíme nějak získávat a analyzovat. Pro tyto účely jsou používány OLAP analýzy a transformace. Tyto analýzy a operace jsou používány při vyhodnocování dotazů na data v datových skladištích a podrobně jsou popisovány v [1].

Práce se zabývá návrhem a realizací prototypu jádra datového skladiště, zvláště jeho naplňováním daty a manipulací s jeho strukturami.

Nejdříve je nastíněna problematika datových skladišť, jejich vývoj a úskalí manipulace s daty. Dále jsou vzneseny požadavky na vytvářený systém, návrh řešení a popis implementace jednotlivých částí systému. Ke konci práce je nastíněn směr možných vylepšení systému datového skladu, ukázka vytvoření datové kostky a manipulace s daty v tomto systému.

Práce navazuje na prezentaci semestrálního projektu, ze kterého byly převzaty informace o komunikaci OLAP systému s uživatelem a metodách ukládání dat v OLAP systémech.

Kapitola 2

Vymezení pojmů

V této kapitole je uveden výčet všech používaných pojmů včetně jejich významů.

Datové skladiště je systém, který skladuje a spravuje data pro OLAP a podporuje ETL operace.

Dimenze je datový prvek, který kategorizuje každou položku souboru dat na nepřekrývající se oblasti.

Dimenzní tabulka je jednou z pomocných tabulek tabulky faktů. Obsahuje vlastnosti dat a bývá využita k omezení a seskupování dat při dotazech.

ETL – Extract, transform, load – získat, převést, načíst jsou procesy načítání dat do datového skladiště.

NULL je označení prázdné nebo nevyplněné hodnoty.

ODS – Operational data store – sklad operačních dat je místo, kde se ukládají data z produkčního systému před natažením do tabulek faktů a dimenzních tabulek.

OLAP – On Line Analytical Processing – systém pro podporu rozhodování.

Sekvence je databázový objekt, který zajišťuje generování unikátních čísel z rostoucí řady.

Servlet je rozhraní, které umožňuje vytvářet dynamické webové aplikace.

Tabulka faktů je hlavní tabulkou kostky, ke které jsou přivázány dimenzní tabulky. Obsahuje měření, míry nebo fakta.

Uložiště je místo v počítačovém systému, kde jsou ukládána data.

Kapitola 3

Uvedení do problematiky

3.1 Co je to datové skladiště?

Pod pojmem datové skladiště rozumíme místo, kde skladujeme historické informace. Datové skladiště se od běžného skladiště liší především v tom, že datové skladiště stále roste a data z něj nejsou v podstatě nikdy odstraňována. Data v datových skladištích jsou následně používána například pro podporu rozhodování řídících pracovníků. Na tento typ datových skladišť je kladeno z technického hlediska několik značně protichůdných požadavků.

Jedná se o tyto požadavky:

- pojmutí velkého množství dat,
- vyhledávání v těchto datech,
- řazení těchto dat,
- možnost stále měnit požadavky.

Cestou pro specifikaci takto nastavených požadavků, rychlé předání požadavků a vyřešení požadavků je použití systému OLAP.

OLAP, dle [4], lze chápat jako skupinu softwarových nástrojů, které jsou určeny k provádění analýzy dat v datových skladištích. Cílem tohoto typu softwarových nástrojů je umožnění uživatelům analyzovat data dle různých úhlů pohledu, data filtrovat, agregovat a sumovat.

3.2 Kostka, fakta a dimenze

Systémy OLAP vyžadují organizaci vstupních dat ve formátu tzv. datových kostek. Datové kostky jsou systémově organizovány do vícerozměrné kostky, kde rozměry jsou dimenzemi a uvnitř se nacházejí fakta.

Všechny datové kostky jsou tvořeny dvěma základními typy údajů: fakty a dimenzemi. Fakta a dimenze představují základní entity datového modelu, které určují, z jakých pohledů bude možno na data v datových kostkách nahlížet (např. čas, produkty, provozovny apod.).

Fakta jsou obsahem datové kostky ve formě numerické měrné jednotky. Jsou tvořeny informacemi. Tabulka faktů je největší tabulkou v datovém skladišti a je obvykle tvořena velkým objemem dat.

Dimenze narozdíl od faktů obsahují logicky nebo organizačně hierarchicky uspořádaná data. Tato data specifikují podrobnosti popisující informace pro podporu rozhodování. Tabulky dimenzí

bývají menší než tabulky faktů a data, která obsahují, nejsou měněna tak často. Nejčastěji se používají časové, produktové a geografické dimenze. Tabulky dimenzí jsou většinou organizovány ve stromové struktuře.

Dimenzi můžeme chápat jako měřítko: kdy, kde, případně co se stalo. Fakta nejčastěji označují množství či míru dějů v prostoru ohraničeném dimenzemi.

Například: dne 4.3.2007 se v samoobsluze na Slovanském náměstí na Králově Poli prodalo 25 jablek.

Jako dimenze můžeme chápat:

- čas prodeje,
- místo prodeje a
- zboží.

Těchto dimenzí může být neomezeně mnoho a často jsou hierarchické:

- rok prodeje – kvartál prodeje – měsíc prodeje – den prodeje – . . . atd.,
- země prodeje – město prodeje – prodejna,
- kategorie zboží – zboží.

Jako fakt lze poté vzít množství, případně utržený obnos peněz. Na tomto příkladu je ukázáno, že každý den může vznikat obrovské množství informací. Skladování takového množství dat a hledání v nich mohou být velmi složité procesy, které vyžadují velké hardwarové i softwarové nároky.

3.3 Vývoj datových skladišť

Systémy datových skladišť, tak jako většina lidských produktů, prošly během svého vývoje různými stádii. První verze byly charakterizovány získáváním informací pro podporu rozhodování přímo z produkčních relačních databází. Tento použitý přístup přestal velmi rychle dostačovat z hlediska výkonnosti. Navíc tento přístup při vyhodnocování dotazu výrazně zpomaloval a zatěžoval produkční databázi.

Nejjednodušší variantou řešení vedoucí k odstranění těchto nedostatků bylo vytvořit pro účely vyhodnocování dotazů kopii databáze a pracovat s touto kopií. Tato kopie databáze je oddělena od produkční databáze a v jistých časových intervalech je synchronizována s produkční databází. Negativem tohoto přístupu je nezbytná potřeba dvojnásobku úložného prostoru a minimálně dvojnásobku procesorového výkonu. Pozitivum tohoto řešení však spočívá v tom, že přetížení systému s datovým skladištěm nezpůsobí přetížení produkčního systému. Produkčnímu systému tak neklesá při vyhodnocování dotazů výkon a pro přechod k tomuto řešení stačí pouze provádět replikaci na druhý systém a využívat jej místo produkčního systému pro vyhodnocování dotazů.

Zanedlouho ovšem ani toto řešení výkonnostně nestačilo. Schéma uložení dat v produkční databázi bylo navrženo primárně pro transakční přístup, není tudíž vhodné pro získávání informací z dat. Proto byl zvolen postup optimalizace uložení dat ve skladišti pro vytváření filtrací, agregací a sumací, které jsou nejčastějším požadavkem pohledu na data ve skladišti. Optimalizace uložení spočívá v uložení nadbytečného množství dat, pro rychlejší manipulaci s daty. Toto řešení je používáno dodnes, nicméně přesun dat mezi formou vhodnou pro OLAP a produkčními systémy je stále velmi náročný proces.

3.4 Rozdíl mezi uložišti pro OLAP a produkčními systémy

Pod pojmem datové skladiště rozumíme uložiště pro OLAP systém. Tento systém se od produkčního systému liší především orientací na tvorbu sestav a jiných podkladů určených pro strategická rozhodnutí. U produkčních systémů je kladen především důraz na konzistenci dat při víceuživatelském přístupu. Datová skladiště pro OLAP používají často nadbytečné (například agregované či předsumované) informace ke zvýšení rychlosti zpracování. Velmi často se tak nedosahuje třetí normální formy, která bývá u produkčních systémů většinou vyžadována, což je zde tolerováno z toho důvodu, že tím dojde k urychlení vyhodnocení dotazů.

Další podstatný rozdíl spočívá ve faktu, že v datovém skladišti pro OLAP při uživatelské práci nedochází často ke změnám v uložených datech. Data většinou nejsou editována či odstraňována, pouze dochází k přidávání, což většinou probíhá při migraci dat z produkčního systému při dávkovém zpracování (často jednou denně, týdně, měsíčně, ročně či jinak) bez zásahu uživatele. Tato operace se označuje ETL. Mezi získáním dat z produkčního systému bývají informace uloženy v ODS skladu, kam se přesunou hrubá data z produkčního systému, aby se minimalizovala doba, kdy je zatěžován produkční systém. Data bývají přesouvána po tzv. balících dat. Balík dat je chápán jako soubor datových položek, které jsou načítány najednou.

Velké balíky dat bývají zpravidla přesouvány v době, kdy má produkční systém nejnižší zatížení, nejčastěji v noci, či během svátků. Přesun malých rozdílových balíků dat je možné provádět i za provozu nebo v době, kdy není systém plně vytížen.

3.5 Natahování dat do datového skladiště

Natahování dat do datového skladiště může probíhat dvěma základními metodami:

- metoda přírůstková,
- metoda velkého třesku.

Při natahování metodou velkého třesku dochází k vyčištění datového skladiště a jeho opětovnému naplnění daty z ODS. ODS je místo, kam se přesunou data z produkčního systému před natažením do datového skladiště. Hlavní nevýhodou tohoto postupu je jeho značná časová náročnost, zvláště při velikosti používaných velkých datových skladišť ve stovkách gigabajtů či jednotkách terabajtů. Při této velikosti může dle výkonnosti serveru trvat natahování i několik dní či týdnů.

Při natahování přírůstkovou metodou dochází k přesunu přírůstků dat do datového skladiště. Tento přesun může být dokonce prováděn přímo za provozu produkčního systému z jeho protokolu o změnách. V tomto případě můžeme dosáhnout téměř přesného zobrazení všech dostupných dat do datového skladiště. Při změnách dat ovšem dochází ke stavu, kdy uživatel má data ze zobrazení kostky, která se již změnila. O tomto problému je pojednáno v kapitole 7.

V praxi je preferována přírůstková metoda z důvodu rozložení zatížení serveru do delšího časového úseku. Metodu velkého třesku však nelze opominout, neboť bývá používána při vytváření skladiště, nebo v případě problému s daty, když je nutné data znovu natáhnout.

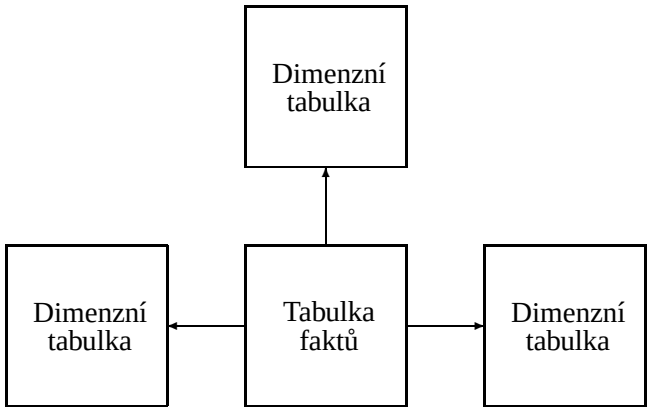
3.6 Schémata dat pro OLAP

Data v datových skladištích pro OLAP jsou uložena ve formátu vícerozměrných kostek. Každá kostka se skládá z tabulky faktů, která zabírá většinou více než 90% prostoru zabraného kostkou, a z několika dimenzních tabulek, které specifikují možné pohledy na data uložená v tabulce faktů.

Dimenzní tabulky mohou být navázány buď pouze přímo na tabulku faktů, nebo na další dimenzní tabulky. Tato specifická uspořádání se nazývají schémata. Mezi nejčastěji používané patří schéma *hvězda*, schéma *sněhová vločka* či jejich kombinace.

3.6.1 Schéma hvězda

Schéma hvězda je schéma uložení dat v datovém skladišti, ve kterém jsou všechny dimenzní tabulky navázány přímo na tabulku faktů. Je to často používané schéma. Příklad použití schématu hvězda je zobrazen v obrázku 3.1. Schéma hvězda je výsledek nejjednodušší metody transformace relačních dat na data multidimenzionální. Každá dimenze je představována právě jednou dimenzní tabulkou. Každá tabulka může mít více atributů, například časová dimenze může mít atributy: rok, kvartál, měsíc a den.



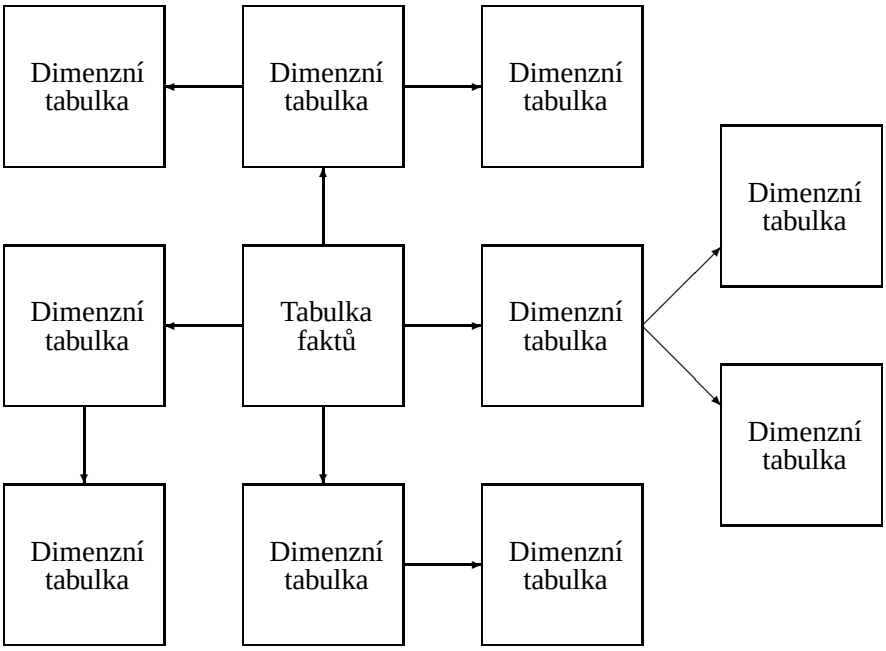
Obrázek 3.1: Nákres příkladu propojení tabulek faktů a dimenzních tabulek ve schématu hvězda

3.6.2 Schéma sněhová vločka

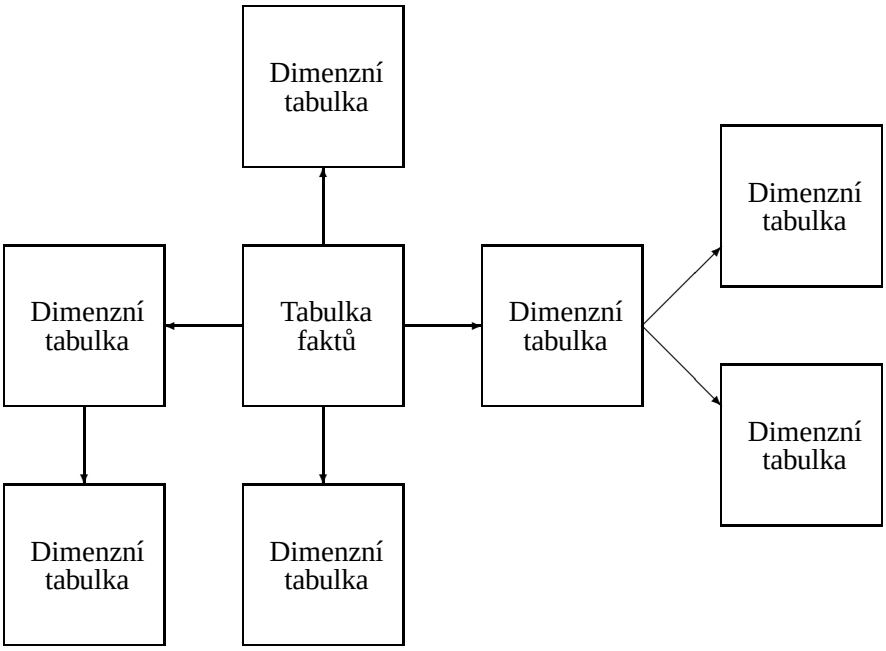
Schéma sněhová vločka je schéma uložení dat v datovém skladišti, ve kterém jsou dimenzní tabulky vázány přes jinou dimenzní tabulku. Ukázka tohoto schématu je v obrázku 3.2. Toto schéma vzniká normalizací schématu hvězda na třetí normální formu. Dochází tak k snížení množství redundantních informací v dimenzních tabulkách. Zvýší se tak udržovatelnost dat v dimenzních tabulkách a sníží se nároky na diskový prostor nutný pro jejich uložení. Přesto je manipulace často pomalejší než při schématu hvězda z důvodu nutnosti spojení více tabulek při dotazu. Toto schéma se používá ojediněle.

3.6.3 Kombinované schéma

Nejčastěji se prvky schématu typu hvězda a schématu typu sněhová vločka kombinují, protože je tím využíváno pozitiv obou přístupů pro maximalizaci výkonu datového skladiště. Některé dimenze jsou v třetí normální formě, některé pouze v druhé normální formě. Příkladem tohoto schématu je obrázek 3.3.



Obrázek 3.2: Nákres příkladu propojení tabulek faktů a dimenzních tabulek ve schématu sněhová vločka



Obrázek 3.3: Nákres propojení tabulek faktů a dimenzních tabulek v kombinovaném schématu

3.7 Komunikace OLAP s uživatelem

Nejčastějším způsobem komunikace uživatele s OLAP systémem je komunikace pomocí relační či kontingenční tabulky. Méně rozšířeným, i když velmi přehledným způsobem je komunikace pomocí kontingenčního grafu. Každý z těchto způsobů má svoje využití a proto nelze některý z nich vyloučit.

Relační tabulka obsahuje na každém řádku jeden fakt.

Tabulka je rozdělena svisle na dvě části:

- ve sloupcích v levé části jsou uvedeny jednotlivé dimenze,
- v pravé části potom jednotlivé hodnoty z tabulky faktů.

Tento způsob je výhodný zvláště při řídkém obsazení dat mezi dimenzemi, protože zobrazuje jen existující fakta. Při zobrazení kontingenční tabulkou můžeme zase jednoduše a názorně detekovat závislosti mezi více dimenzemi. V kontingenčním grafu můžeme nejsnáze vidět trendy vývoje či různé anomálie.

Kapitola 4

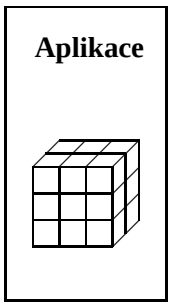
Ukládání dat v systémech OLAP

V principu dělíme systémy OLAP dle umístění datových skladišť na:

- desktop OLAP,
- multidimenzionální OLAP,
- relační OLAP a
- hybridní OLAP.

4.1 Desktop OLAP

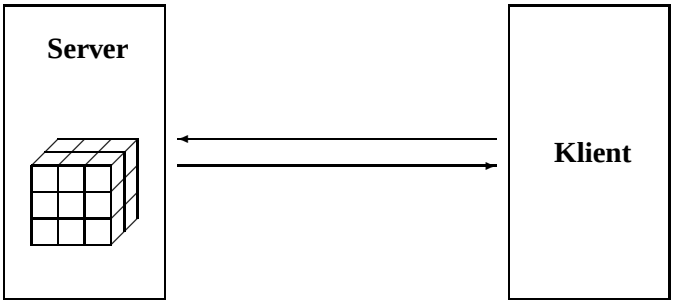
Desktop OLAP je nejjednodušší systém OLAP, který lze vytvořit. Je to jednovrstvá aplikace, jež se vyznačuje limitací svojí velikosti, protože jde o systém běžící celý na počítači uživatele, viz obrázek 4.1. Uživatelský počítač většinou nedisponuje velkými systémovými prostředky v porovnání se servery. Desktop OLAP bývá používán jen v opravdu vyjímečných případech. Jedním z těchto případů může být offline manipulace s malou částí velkého OLAP systému.



Obrázek 4.1: Schéma rozprostření vrstev systému Desktop OLAP

4.2 Multidimenzionální OLAP

Multidimenzionální OLAP je dvouvrstvý OLAP systém, který udržuje všechna data na serveru ve speciálním formátu, tzv. multidimenzním poli. K datům přistupuje uživatel pomocí klientského programu. Způsob uložení v multidimenzním poli je optimalizován pro OLAP, což systému umožňuje mít velmi vysoký výkon při vyřizování dotazů. Zároveň to dovoluje uložit obsah datového skladu do velmi malého diskového prostoru. Multidimenzionální OLAP je vhodný pro malé až středně velké datové sklady. Některá řešení mají však problémy s větším množstvím než cca 10 dimenzí. Záleží samozřejmě na kardinalitě dimenze. Pokud lze kostku celou načíst a manipulovat s ní v paměti serveru, nemá toto řešení výkonovou konkurenci. Schéma častého rozprostření vrstev je znázorněno v obrázku 4.2.

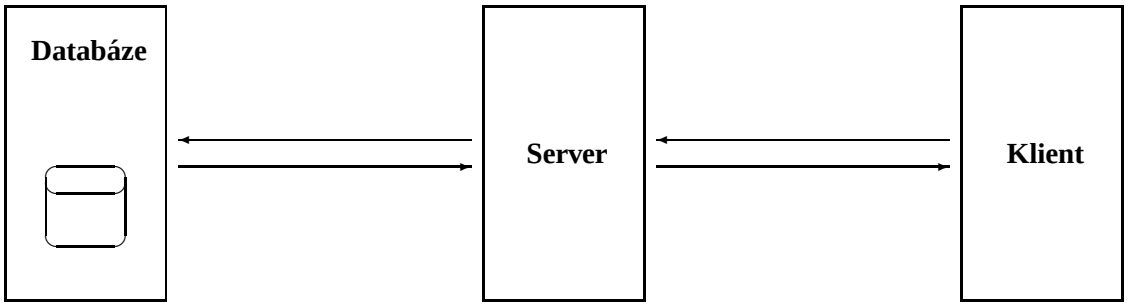


Obrázek 4.2: Schéma rozprostření vrstev systému Multidimensional OLAP

4.3 Relační OLAP

Relační OLAP je třívrstvý OLAP systém, který má veškerá svá data uložena v relační databázi. Toto řešení není omezeno velikostí, která může růst do desítek gigabajtů až jednotek terabajtů dat. Relační databáze mívají o cca jeden řád větší alokovaný diskový prostor, než je objem dat v nich uložených.

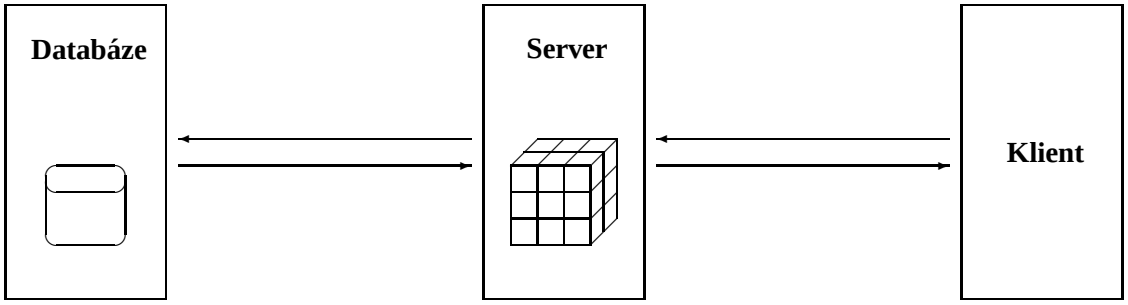
Větší problém zde ale nastává při natahování dat do skladiště. Pro tuto operaci relační databáze nemá žádný nástroj a je nutno vytvořit vrstvu, která bude zajišťovat načítání dat. Tato vrstva je součástí serveru. Server taky zprostředkovává data z relační databáze klientské aplikaci, která je zobrazí uživateli. Na rozdíl od multidimenzionálního OLAPu u relačního OLAPu není problém s konzistencí v případě výpadku databázového serveru. Většina relačních databází má velmi dobře vyřešeno obnovení systému po pádu systému. Relačního OLAPu využívá většina implementací dostupných na trhu. Toto řešení je organizováno tak, jak je zobrazeno v obrázku 4.3.



Obrázek 4.3: Schéma rozprostření vrstev systému Relational OLAP

4.4 Hybridní OLAP

Hybridní OLAP je třívrstvý OLAP systém, který spojuje relační a multidimenzionální OLAP pro zefektivnění a urychlení vyhodnocování dotazů. Snaží se eliminovat nedostatky multidimenzionálního OLAPu a relačního OLAPu a maximalizovat jejich výhody. Toto řešení je velmi efektivní, ale také nejsložitější. Udržuje celý obsah kostky v relační databázi a její agregace v multidimenzionálním poli (viz obrázek 4.4). Hybridní OLAP lze s výhodou použít v případě velkých datových skladišť, ze kterých používáme vždy malou část jako výřez pro dotazování a analýzy. Tato malá část datové kostky může být například při otevření kostky zkopírována do multidimenzního pole, kde manipulace trvají řádově kratší dobu.



Obrázek 4.4: Schéma rozprostření vrstev systému Hybrid OLAP

Kapitola 5

Požadavky

Cílem této práce je dát k dispozici dobře zdokumentovaný a podle posledních poznatků koncipovaný prototyp datového skladiště pro OLAP.

Toto datové skladiště by mělo umožnit spouštění ETL operací, implementovaných jako zásuvný modul, a mělo by být koncipováno jako relační OLAP systém s otevřenou možností upravit systém na hybridní OLAP.

Mezi operace, které by mělo datové skladiště podporovat, musí patřit:

- vytvoření kostky,
- editace struktury kostky,
- volání ETL rutiny, které provede načtení balíku dat do ODS,
- natažení dat z ODS do tabulek faktů a dimenzních tabulek a
- vymazání tabulky faktů pro případ reimportu dat z ODS.

Velmi žádoucí je, aby projekt byl spustitelný na běžně dostupných softwarových platformách:

- Microsoft Windows,
- GNU/Linux,

přičemž případné použité knihovny či projekty by měly být zdarma k dispozici.

Dále by měla být prozkoumána možnost častého načítání dat do ODS a natahování do tabulek faktů a dimenzních tabulek a prověřena náročnost a reálnost průběžného informování klienta o změnách v kostce, kterou aktuálně prohlíží.

Kapitola 6

Návrh řešení

Datové skladiště bude tvořit druhou vrstvu třívrstvého řešení systému OLAP. Hlavní snahou při vytváření je modulárnost a multiplatformnost tohoto řešení. Z těchto důvodů byla jako jazyk zvolena Java[2] od společnosti Sun Microsystems[1].

Toto skladiště bude používat běžnou relační databázi jako primární uložiště dat. V této databázi budou kromě dat soustředěna i veškerá metadata. Jako databáze bude využito projektu PostgreSQL [3], jako běžně dostupné databázové platformy.

S databází bude komunikováno pomocí standardizovaného rozhraní JDBC. Projekt PostgreSQL toto rozhraní obsahuje ve standardní instalaci.

6.1 Schéma ukládání dat a metadat

Hlavní motivací při návrhu řešení je neupnout se příliš k jedné databázové platformě. Proto při vytváření schématu uložení metadat do databáze bude provedena duplikace části systémového katalogu. Systémový katalog je část databáze, která obsahuje informace o tabulkách, sloupcích apod.. Tak bude zajištěna jednoduchá portovatelnost řešení na jiné databázové uložiště. I přesto bude využito jedné specifické vlastnosti systému PostgreSQL, a to automatického doplňování obsahu sloupce ze sekvence a manipulace se sekvencí.

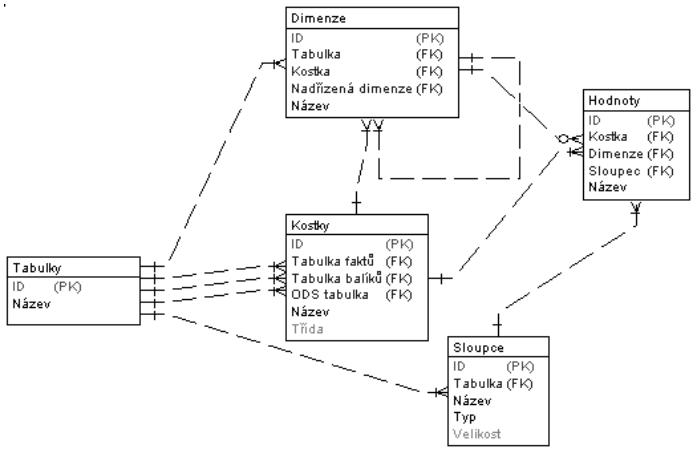
V metadatech budou uloženy informace o všech kostkách, jejich dimenzích, atributech, metrikách a faktech. Každá tabulka bude svázána s kostkou, do které patří. Jejich názvy budou automaticky generovány dle určitého schématu. Nákres provázání metadat je uveden v obrázku 6.1.

Každá kostka se bude skládat z:

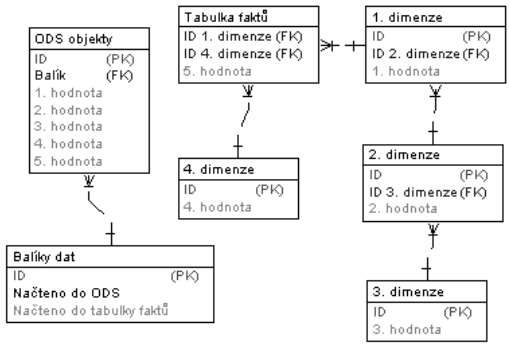
- tabulky faktů,
- dimenzních tabulek,
- tabulky balíků dat a
- tabulky s ODS objekty.

Tabulka balíků dat a tabulka s ODS budou svázány cizími klíči pro zajištění referenční integrity, taktéž budou spojeny dimenzní tabulky a tabulka faktů. Referenční integrita je mechanizmus, který zajišťuje nerozporuplnost obsahu tabulek. Ta je zde velmi důležitá, protože i malá chyba v datech by mohla způsobit vznik redundantních dat, které mohou vytvářet problémy.

[1] Název Sun Microsystems je registrovanou obchodní známkou firmy Sun Microsystems, Inc.



Obrázek 6.1: Logické ERD schéma uložení a provázání metadat



Obrázek 6.2: Logické ERD schéma uložení a provázání tabulek vzorové kostky

Kapitola 7

Manipulace s daty v reálném čase

Manipulace s daty v reálném čase je vlastnost, která není v systémech OLAP ani v jejich datových skladištích řešena. Je předpokládáno, že data v datovém skladišti mají v čase neměnnou podobu. Data v produkčních systémech, ze kterých berou OLAP systémy data, se ale v čase mění. Otázkou je proč tedy nepromítnout tyto změny i do OLAP systémů a neumožnit uživateli sledovat téměř aktuální informace, staré třeba jen několik minut namísto informací, které jsou staré třeba i několik dní. Aktuálnost dat v rozhodovacím procesu může být tím, co může naše řešení odlišovat od jiných řešení a poskytovat možnost být mnohem pružnější v reakci například na vývoj trhu. V dnešní době to může být zásadní a jediná věc, která bude konkurenční výhodou tohoto přístupu k řešení.

Při označování stáří faktu v databázi je několik postupů.

- Prvním postupem je, že pro tento účel bude vytvořena sekvence, jejíž hodnotu budeme přidávat ke každému faktu a která bude označovat, jakou operaci právě provádíme. Tato varianta má nevýhodu v tom, že nemůžeme zjistit aktuální hodnotu sekvence, aniž bychom si od ní vyžádali další číslo.

- Druhou možností postupu je vložení časového razítka (časovou hodnotu typu `timestamp`) ke každému faktu. Toto řešení naráží na problém vznikající z důvodu transakčního zpracování, tj. načítání jen potvrzených transakcí. Může dojít k případu, že je změněn fakt, který však v čase T není ještě potvrzen a tak dojde k načtení původní hodnoty faktu. Protože ale došlo k změně před časem T, je časové razítko starší než čas T, a tak se systém domnívá, že ke změně ve faktu nedošlo.

- Třetí možnost spočívá ve změně označování příznaku načtení balíku z logického typu na číselný typ. Změníme tak typ dat ve sloupci označujícím načtení balíku dat z typu `boolean` na typ `integer`. Číslo ze sekvence, která bude pro tento účel vytvořena při vytváření kostky, bude uloženo do tohoto sloupce při načtení tohoto balíku. Příznak načtení bude hodnota různá od NULL. Toto řešení jednak výrazně šetří místo v uložišti dat, jednak umožňuje snadné nalezení aktuálního stavu sekvence.

Na základě posouzení jednotlivých způsobů řešení byl zvolen třetí způsob jako optimální k implementaci.

Kapitola 8

Implementace datového skladiště

8.1 Struktura ukládání dat a metadat

Data a metadata se nacházejí v relační databázi. Při vytváření datového skladiště jde v podstatě o vytvoření prototypu datového skladiště pro OLAP. V tomto prototypu není pro zjednodušení počítáno přímo s podporou výměny databázové platformy pouze výměnou jednoho modulu. Nicméně by neměl být velký problém podporu vyměnného modulu pro abstrakci databázového přístupu do systému v případě nutnosti doplnit.

Tabulky obsahující metadata jsou `tables`, `columns`, `cubes`, `dimensions` a `values`.

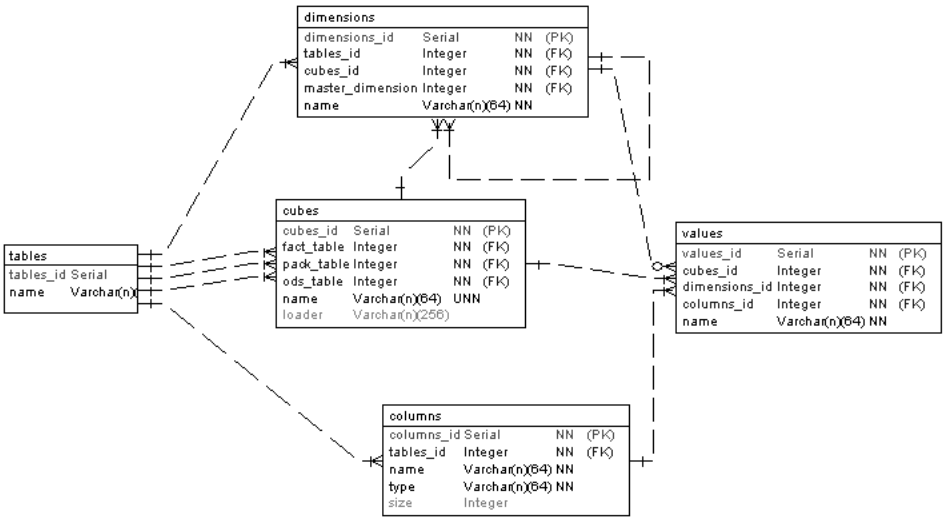
- Tabulka `tables` obsahuje všechny tabulky v databázovém schématu.
- Tabulka `columns` obsahuje sloupce těchto tabulek.
- Tabulka `cubes` obsahuje seznam všech kostek a odkazy na podstatné tabulky, které tvoří kostku. Jsou to: tabulka balíků dat, tabulka s ODS a tabulka faktů.
- V tabulce `dimensions` jsou všechny dimenze kostek včetně jejich zanořených částí.
- Informace o obsazích jednotlivých kostek a dimenzí jsou uloženy v tabulce `values`.

Celé schéma tabulek metadat a jejich provázání je uvedeno v obrázku 8.1.

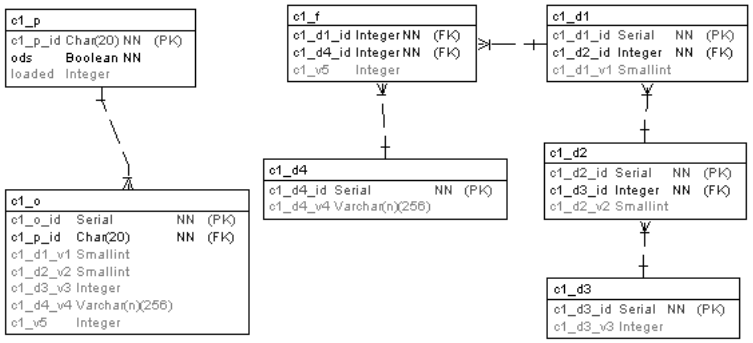
Všechny tabulky, které jsou použity pro uložení dat, jsou dynamicky vytvářeny při tvorbě kostky a jejích částí. Pomocí duplikované části systémového katalogu s nimi lze jednodušeji manipulovat. Názvy tabulek a sloupců jsou generovány systematicky. Všechny tabulky týkající se kostky mají předponu `c<číslo>_`, kde `<číslo>` je pořadové číslo kostky dle sekvence `cubes_cubes_id_seq`. Všechny sloupce s pořadovými čísly mají příponu `_id`. Názvy tabulek s fakty mají příponu `_f`, tabulky se seznamem balíků pro natažení dat `_p`, tabulky s ODS `_o` a tabulky s dimenzemi `_d<číslo>`, kde `<číslo>` je pořadové číslo dimenze dle sekvence `dimensions_dimensions_id_seq`. Sloupce obsahující hodnotu důležitou pro informace uložené v kostce mají příponu `_v<číslo>`, kde `<číslo>` je pořadové číslo dimenze dle sekvence `values_values_id_seq`. Ukázka schématu je v obrázku 8.2.

8.2 Struktura programového řešení

Jak bylo dříve uvedeno, jako programovací jazyk byla použita Java. Pro zjednodušení vytváření serveru bylo použito projektu Jetty, který řeší víceuživatelský přístup k serveru a další problémy, například načítání modulů pro natahování dat do ODS apod..



Obrázek 8.1: Fyzické ERD schéma uložení a provázání metadat



Obrázek 8.2: Fyzické ERD schéma uložení a provázání tabulek vzorové kostky

Pro zjednodušení manipulace a komunikace s datovým skladištěm je v prototypu datového skladiště použito rozhraní přes protokol http. Přes protokol http lze snadno komunikovat pomocí libovolného webového prohlížeče, a protože je implementované rozhraní opravdu jednoduché, lze použít i prohlížeče typu Links či Lynx, pracujícího v konzoli. Toto rozhraní je velmi jednoduché a umožňuje vytvářet, modifikovat a mazat datové kostky. Po prvním importu dat do ODS je zamezeno modifikaci datové kostky, aby nedošlo k poškození dat v ODS. Poté je možno pouze importovat další data do ODS, načítat data z ODS do tabulek faktů a dimenzních tabulek případně vyčistit celý obsah tabulky faktů.

8.3 Rozhraní pro komunikaci se systémem

Celé rozhraní je koncipováno jako jeden servlet, který je namapován do cesty `/olap/` na serveru. V kořenu serveru se nachází úvodní stránka provádějící přesměrování na cestu `/olap/`.

Každá další stránka je tvořena ze dvou částí. V horní části stránky server zobrazuje informace, které si uživatel vyžádal, nebo formulář pro vstup informací od uživatele, spodní část potom obsahuje operace které lze provádět.

Všechny operace, které nevyžadují uživatelský vstup, jsou provedeny ihned po kliknutí na odkaz, proto je nutné zvláště při mazání dávat pozor, aby nedošlo k nežádoucímu vymazání, poněvadž před operací již nebude uživatel varován. Pokud operace vyžaduje údaje od uživatele, je zobrazena stránka s formulářem, po jehož vyplnění a potvrzení je operace provedena.

8.4 Postup importu dat do ODS

O celý import dat do ODS se stará objekt `xdittr03.model.Cube`. Po zavolání metody `importData()` tohoto objektu dojde k vyžádání třídy dle jména uloženého v tabulce `cubes` ve sloupci `loader`. Je testováno, zda je třída implementací rozhraní `xdittr03.data.PackLoader` (viz Dodatek A). V případě, že je shledáno kompatibilním, je zavolána metoda `importData(Cube)`. Jako výsledek volání je očekáván nový objekt třídy `xdittr03.data.Pack`, navázaný na kostku, kterou dostala metoda `importData(Cube)` jako parametr. Třída implementující rozhraní `xdittr03.data.PackLoader` by měla být mostem zprostředkovávajícím data z produkčních systémů a měla by řešit vlastní ETL.

8.5 Postup natahování z ODS

Nejdůležitější operací v datovém skladišti je natažení dat z ODS do tabulky faktů a dimenzních tabulek.

Natažení dat z ODS lze vyvolat dvěma způsoby:

- první způsob je zavolat metodu `load()` na příslušném balíku dat v ODS (objekt třídy `xdittr03.data.Pack`), to vyvolá iterativní natažení celého obsahu tohoto balíku,
- druhým způsobem je zavolat metodu `load()` dané kostky (objekt třídy `xdittr03.model.Cube`). Tento způsob vyvolá metodu `load()` u všech balíků, které mají příznak označující kompletnost balíku v databázi a nemají příznak natažení z ODS do tabulek faktů a dimenzních tabulek.

Natažení jedné položky z ODS spočívá v načtení celého jejího obsahu do záznamu třídy `xdittr03.data.Record` a následném projití všech dimenzí kostky do hloubky. Průchod do hloubky je realizován rekurzivním voláním metody `complete(Record)`. V každé tabulce dimenzí je

nalezen řádek, který odpovídá datům ve třídě `xdittr03.data.Record`. V případě, že řádek není nalezen, je vytvořen podle dat ve třídě `xdittr03.data.Record`. Identifikátor tohoto řádku je přidán jako další část záznamu `xdittr03.data.Record`. Po průchodu všemi dimenzemi je buď upraven řádek v tabulce faktů, nebo je založen nový řádek v případě jeho neexistence. Identifikace řádku tabulky faktů je dána obsahem záznamu, který byl uložen v třídě `xdittr03.data.Record` a následně doplněn průchodem tabulek dimenzí.

Toto řešení má největší výhodu ve své jednoduché paralelizovatelnosti. Jediným limitním faktorem je rychlost databázové platformy.

Kapitola 9

Návrh dalšího rozvoje systému

9.1 Modulární přístup k databázi

V hotovém produktu navrhuji zvážit vytvoření modulární mezivrstvy pro abstrakci přístupu k databázovému stroji a jeho konfiguraci v externím souboru. Primárně je nutné vyřešit jiné názvy typů sloupců na různých databázových platformách. Další problém může vzniknout z důvodu použití sekvencí, protože nejsou součástí normy a každá databázová platforma je má implementovány jinak. V tomto prototypu bylo s výhodou využito automatického použití sekvencí.

9.2 Navržení a implementace protokolu pro komunikaci

Částí, kterou je potřeba přepracovat, je komunikace s klientskou částí systému a s uživatelem. V tomto prototypu je namísto uživatelského rozhraní implementován přímý přístup k serveru. Bylo provedeno pouze zapouzdření přístupu k serveru do protokolu http, z důvodu snadného přístupu k voláním systému. Samozřejmě je nutné vytvořit systém práv a přihlašování, což nebylo předmětem řešení této práce.

9.3 Větší míra zabezpečení importu do ODS

Modulární systém importů do ODS je dalším místem možného vylepšení. V aktuálním stavu je tento systém sice plně funkční, ale bylo by vhodné provést analýzu, zda nemůže dojít k zneužití podstrčením nebezpečné třídy, která by mohla poškodit systém nebo data v něm uložená.

9.4 Automatizované vyvolání importu

Další úpravou by měla být možnost vyvolávat import dat do ODS na vnější podnět, například proběhlou transakci zdrojového systému či plynutí času, což by snižovalo prodlevu mezi vznikem dat ve zdrojovém systému a jejich zobrazením v datovém skladu OLAP systému.

Kapitola 10

Popis použití vytvořeného systému

V rámci práce bylo vytvořeno datové skladiště. Toto datové skladiště naleznete na přiloženém CD viz dodatek B. Po spuštění datového skladiště v adresáři `pgm` příkazem `go.cmd` můžeme přistoupit k rozhraní pomocí libovolného webového prohlížeče na adrese `http://localhost:8080/olap/`.

Pomocí tohoto rozhraní můžeme provádět všechny operace. Vytvoření datové kostky a její naplnění daty se skládá ze 4 operací.

Jedná se o tyto operace:

- vytvoření kostky,
- vytvoření struktury,
- import prvního balíku dat do ODS a
- natažení dat z ODS.

10.1 Vytvoření kostky

Pro vytvoření kostky se na stránce seznam kostek v sekci operace vybere akce: New cube. Na formuláři se zadá unikátní jméno a název třídy, která je implementací rozhraní `xdittr03.data.PackLoader` (viz Dodatek A). Tato třída může obsahovat metodu `createStructure(Cube)`, která slouží k vytvoření struktury kostky. Pokud je v této třídě kód, který vytvoří celou strukturu, je možné přeskočit vytvoření struktury.

Metoda `createStructure(Cube)` je volána pouze při vytvoření nové kostky a není možné ji jakkoli vyvolat znovu.

Pro ukázku je doporučeno použití třídy: `xdittr03.loaders.A`.

10.2 Vytvoření struktury

Pokud není implementováno vytvoření struktury v třídě svázané s kostkou, musí se povést vytvoření struktury kostky ručně, a to tak, že se v operacích kostky vybere Edit cube structure. V případě, že se již struktura vytvořila, je možné tento krok přeskočit.

Potom se mohou vytvářet dimenze a poddimenze kostky příkazem Add dimension, atributy dimenzí příkazem Add value u dané dimenze se může vložit metrika příkazem Add value nad kostkou (položka se nachází téměř úplně dole).

Další možností je vložené dimenze, atributy či metriky kostky mazat pomocí příkazů Delete dimension/value, či přejmenovat příkazem Rename dimension/value. Přejmenovat lze i celou kostku příkazem Rename cube.

Po vložení metriky nebo atributu do kostky není možné již měnit jejich typ z důvodu možného poškození dat uložených v ODS.

10.3 Import balíku dat do ODS a natažení dat z ODS

Vyvolání těchto dvou operací je pro uživatele jednoduché. Po otevření kostky se mohou vyvolat v seznamu akcí kostky pod příkazy:

- Import data to ODS,
- Load ODS data to fact and dimensions tables.

Příslušná operace se zahájí okamžitě po kliknutí na odkaz, po skončení operace se obnoví stránka a změní se statistiky kostky. Při importu vzroste počítadlo Packs in cube o hodnotu 1 a podle množství položek vzroste počítadlo Items in ODS. Při natažení dat do tabulky faktů a dimenzních tabulek z ODS vzroste počítadlo Records in fact table.

Kapitola 11

Závěr

Tato bakalářská práce se zabývá metodami ukládání dat pro OLAP systémy. V úvodu se zabývá problematikou datových skladů a OLAP systémů. V této části bylo využito poznatků ze semestrálního projektu, který byl řešen v zimním semestru akademického roku 2006/2007.

V rámci bakalářské práce byl vytvořen prototyp datového skladiště a byl připraven ukázkový modul pro načítání náhodných dat. Těmito daty bylo naplněno datové skladiště a byla ověřena jeho funkčnost.

Závěrem práce je proveden rozbor dalších možností rozvoje systému a je doporučeno vytvořit klientskou aplikaci, která bude řešit komunikaci s uživatelem a bude izolovat uživatele od volání systému. Na základě poznatků je doporučeno provézt implementaci dotazování a informování klientské aplikace o změnách v datové kostce. Vytvořené datové skladiště by mělo tuto implementaci bez problémů podporovat a umožňovat řešení dané problematiky.

Navržený systém, který splňuje všechny požadavky v rozsahu zadání bakalářské práce, byl realizován a odzkoušen a předpokládá se pokračování a rozšíření v rámci diplomové práce.

Literatura

[1] Wang, J.: *Encyclopedia of data warehousing ang mining*. Idea Group Reference, 2006, ISBN 1-59140-559-9.

[2] WWW stránky: Java. [online], [cit. 2007-04-28].
URL <http://java.sun.com/>

[3] WWW stránky: PostgreSQL. [online], [cit. 2007-04-28].
URL <http://www.postgresql.org/>

[4] WWW stránky: Webopedia. [online], [cit. 2007-04-28].
URL <http://www.webopedia.com/TERM/O/OLAP.html>

Dodatek A

Rozhraní `xdittr03.data.PackLoader`

```
package xdittr03.data;

import xdittr03.model.Cube;

/**
 * Toto rozhrani umoznuje natahovani do ODS
 *
 * @author bodyn
 */
public interface PackLoader {

  /**
   * Toto je metoda, umoznujici vytvorit strukturu automaticky
   *
   * @param   c   Cube ukazujici na kostku, do ktere bude vlozena struktura
   */
  public void createStructure(Cube c);

  /**
   * Toto je hlavni metoda, ktera se o import stara
   *
   * @param   c   Cube ukazujici na kostku, do ktere miri import
   * @return  Pack, ktery obsahuje nova data
   */
  public Pack importData(Cube c);

}
```

Dodatek B

Obsah CD

Na přiloženém CD jsou uloženy:

- zdrojová data pro tisk zprávy v adresáři `text`,
- zdrojové kódy vytvořeného systému v adresáři `olap`,
- záloha databáze obsahující tabulky s metadaty v souboru `database.backup`,
- připravený software Jetty, včetně konfigurace a zkompilovaného projektu v adresáři `pgm`.