VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Datové sklady a Business Intelligence příklady a porovnání

Data warehouses and Business Intelligence examples and comparison

2009 Radek Tomášek

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2009 .....

Rád bych na tomto místě poděkoval svému vedoucímu, Ing. Radoslavu Fasugovi, Ph.D,
za jeho vedení, cenné rady a celkovou odbornou pomoc, bez níž by tato práce nevznikla.

Abstrakt

Úkolem této bakalářské práce je vhodně demonstrovat klíčové prvky tvorby datových skladů s použitím nástrojů Business Intelligence. Celé řešení je postaveno na MS SQL Serveru 2005 s využitím produktu SQL Server Business Intelligence Development Studio. Součástí této práce je příručka, v níž je zobrazena postupná tvorba jednoduchého datového skladu a jeho možné využití včetně reportů a multidimenzionálních OLAP databází. Nechybí ani propojení s externími aplikacemi (MS Excel 2003 a 2007). Práce má být pomůckou pro studenty kurzu Informační systémy a datové sklady (INS), který je součástí navazujícího magisterského studia.

Klíčová slova: bakalářská práce, datový sklad, Business Intelligence, MS SQL 2005, SQL Server Business Intelligence Development Studio, OLAP, reporty, MS Office, Informační systémy a datové sklady (INS)

Abstract

The main goal of this Bachelor thesis is suitably demonstrate key features of implementation of data warehouses using Business Intelligence tools. Therefore, MS SQL Server 2005 and SQL Server Business Intelligence Development are primary platforms. Our thesis contains user's manual where is shown a step by step implementation of data warehouse and its applications including reports and OLAP multidimensional databases with MS Excel 2003 and 2007 extension. This work should be suitably for course in Information Systems (INS) for graduate students.

Keywords: bachelor thesis, data warehouse, Business intelligence, MS SQL 2005, SQL Server Business Intelligence Development Studio, OLAP, reports, MS Office, Information Systems (INS)

Seznam použitých zkratek a symbolů

SŘBD	–	Systém řížení báze dat
DWH	–	Datový sklad (Data Warehouse)
BI	–	Business Intelligence
OLTP	–	Online Transaction Processing
OLAP	–	Online Analytical Processing
MOLAP	–	Multidimensional Online Analytical Processing
ROLAP	–	Relational Online Analytical Processing
HOLAP	–	Hybrid Online Analytical Processing
SCD	–	Slowly Changing Dimension
DDL	–	Data Definition Language
AMEX	–	American Stock Exchange
BIDS	–	Business Intelligence Development Studio

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Vývoj je k nezastavení. To, co bylo ještě před několika desítkami let pouhým snem několika jedinců, je dnes běžnou realitou všedního života. A nemusí jít jen o oblast informačních technologií, i když tam je každá změna patrná nejvíce. Když bylo nutné zvolit bakalářskou práci, osobně jsem měl velmi těžké rozhodování. Nebyl problém vybrat osvědčená témata, ale mnohem větším lákadlem bylo se pustit do něčeho méně známého. Výběr proto nakonec padl na problematiku datových skladů a Business Intelligence. Práce si klade za cíl seznámit čtenáře s problematikou budování datových skladů, důraz však není kladen na detailní rozbor jednotlivých technologií. Probrány jsou spíše obecné principy a ty nejdůležitější části jsou na vhodných příkladech implementovány a zařazeny do uživatelské příručky, která je součástí této práce.

Čtenář bude průběžně seznámen se zajímavostmi, jež často přesahují rámec oboru. Chybět nebudou ani ukázkové výstupy z různých nástrojů pro tvorbu BI. Zájemci o hlubší studium se dočkají pečlivě vybranné literatury. Ta je k dispozici na závěr každé důležité kapitoly. Protože jde o oblast poměrně rozsáhlou, nebylo v mých silách pokrýt vše, co bylo původně zamýšleno. Přesto nepochybuji, že čtenáři, který je začátečník a má o tuto problematiku skutečný zájem, přinese práce spoustu hodnotných a užitečných informací.

Kapitoly jsou sestavovány tak, aby skutečně pokryly většinu prvků z praxe. Vše je samozřejmě velmi relativní, při práci se muselo vycházet s velmi omezenými možnostmi, at'už po stránce hardwarové, tak po stránce datové.

2 Historický vývoj DWH a potřeba skladování dat

2.1 Stručný přehled

Při studiu na vysoké škole se lze hlouběji seznámit s teorií databázových systémů a primárním důvodem, proč je potřeba se evidencí dat zabývat. Tato příčina nevznikla sama od sebe, ale je poznamenaná určitým historickým vývojem, jenž sahá hluboko do minulosti. Ještě před dobu velkého rozmachu výpočetní techniky. V raných dobách stačil obyčejný zápisník a pero. Později se přešlo na psací stroje. Kupříkladu obchodníci zapisovali průběh transakcí nebo lékaři vedli statistiky svých pacientů. S přibývajícími léty však množství dat neustále rostlo a začalo se objevovat množství problémů, s nimiž se původně nepočítalo. Mezi ně patřila možnost editace dat. Kompletní přepis daného listu ale z časových důvodů často možný nebyl, proto se vše v praxi řešilo tak, že se problematické místo mnohdy pracně retušovalo a nahradilo novým údajem.

Problémů ubylo s rozvojem výpočetní techniky, kdy se začalo k mnoha věcem přistupovat odlišně. Začala vznikat sada programů - agendy, jež řešily určitou množinu úloh nad daty. Tomuto případu se začalo říkat *Agendové zpracování dat* a v praxi to znamenalo, že se například vytvořila aplikace v jazyce C++, a ta se používala pro potřeby firmy. S přibývajícím časem však i původní programové řešení přestalo stačit. Společnost se začala rozrůstat a původní program bylo potřeba přepsat pro nové účely. Bohužel při úprávách kódu docházelo v určitém procentu k zanášení nových chyb. Není se proto čemu divit, že se postupně hledaly nové cesty, jak si práci co nejvíce usnadnit.

Než však došlo k prosazení relačních databází tak, jak jsou známy dnes, uplynulo ještě spoustu let. Mezitím jsme zde měli *hiearchický a síťový databázový model*. Hierarchickou databázi si lze představit jako strom, který má své kořenové uzly (*rodič*) a k nim přidružené další uzly (*potomek*). Tento typ databázového modelu trpí, i přes řadu výhod, kterými jsou uspořádání a rychlost, jednou obrovskou nevýhodou. Při vkládání poduzlu se mohlo stát, že nebyl k dispozici vhodný nadřazený rodičovský prvek. Ten se často vytvořil uměle, což mohlo způsobovat mnoho problémů narušení konzistence nebo zvýšenou redundanci dat.

Tyto problémy byly typické i při agendovém zpracování dat. Od databázového modelu se však očekávalo něco jiného. Byl navržen síťový databázový model, kde hlavní výhodou byl rychlý přístup k datům. Uživatelé se setkávali s pojmy *uzel* a *množinová struktura*. Nevýhodou celého řešení bylo potřeba znát celou strukturu datábáze. A tak se pomalu přistupilo k navržení relačního databázového modelu, jenž by veškeré nedostatky obou výše zmíněných modelů vyřešil. S teoretickým řešením přišel v roce 1969 Dr. Edgar F. Codd, který jako výzkumník firmy IBM definoval sadu pravidel pro splnění relačnosti. Postupně se začaly objevovat firmy, jež celé řešení implementovaly a z dnešního pohledu lze bezpečně usoudit, že původní nápad byl více než úspěšný. Hlavním paradigmatem celé databázové technologie bylo *oddělení datových struktur od programů* a aby byl využit potencionál relačních databází naplno, bylo nezbytné zvládnout několik klíčových problémů.

Dříve trápila vývojáře již zmíněná redundance, neboli nadbytečnost. Mohlo se stát, že díky špatnému návrhu byla stejná data uložena na více místech a snadno mohlo dojít k nekonzistenci dat. Pro řešení se často přistupuje k tzv. procesu normalizace.

Definice 2.1 *Redundantní data jsou údaje, které se v poli opakují v důsledku účasti pole ve vztahu mezi dvěma tabulkami nebo jako důsledek nějaké anomálie pole či tabulky.*

Definice 2.2 *Normalizace je proces rozkladu velkých tabulek na menší, vedoucí k eliminaci redundantních a duplicitních dat.*

Proces normalizace je pak zpravidla aplikací několika pravidel normálních forem, z nichž ty nejdůležitější je vhodné formulovat.

Definice 2.3 *Tabulka splňuje podmínku první normální formy tehdy, když všechny atributy (sloupce) jsou atomické, tedy dále nedělitelné.*

Definice 2.4 *Tabulka splňuje podmínku pro zařazení do druhé normální formy tehdy, když splňuje podmínku první normální formy a každý atribut kromě primárního klíče musí být úplně závislý na celém primárním klíči.*

Definice 2.5 *Tabulka je ve třetí normální formě tehdy, když je ve druhé normální formě a zároveň neexistují závislosti neklíčových sloupců tabulky.*

V žádném případě ale nelze tvrdit, že když se bude striktně dodržovat proces normalizace, pak redundance existovat nebude. Ta bude součástí řešení vždy, ať už v rámci zlepšení výkonu nebo jiné optimalizace. Jen je ji potřeba udržet na přijatelné a co nejmenší úrovni.

Velmi důležitými pojmy jsou také *transakce* a *operační databáze*.

Definice 2.6 *Transakcí nazýváme základní logickou jednotku zpracování. Aby byla zachována konzistence dat, musí transakce proběhnout buď celá, nebo je nutné obnovit původní stav a spustit transakci znovu. Říkáme, že transakce musí být atomická, tj. dále logicky nedělitelná.*

Definice 2.7 *Operační databáze je typ databáze, která ukládá dynamická data a používá se v situacích, kdy je třeba shromažďovat, modifikovat a udržovat data*

Právě operační databáze jsou nejčastěji používany v *OLTP úložištích*.

Definice 2.8 *OLTP (Online Transaction Processing) je systém v SŘBD určený pro okamžité zpracování transakcí a odpovídající modifikování hlavních souborů.*

Jinými slovy lze říci, že OLTP přístup je tradiční pojetí databázového zpracování, jež je velmi dobře známo po celém světě. Informační systémy bank, letišť či obchodů denně zaplňují dobře definované relační struktury, provadí úpravu vybranných záznamů či je rovnou odstraňují. Při návrhu podobných systémů může analytik/programátor často vycházet pouze z potřeb zadavatele. Ten zná velmi dobře pozadí svého businessu a při

důkladné konzultaci s vývojářem tak může vzniknout velmi rigorózní návrh, jenž pak není složité naprogramovat.

Firma může používat více podobných systémů a často nastává problém, jak vše mezi sebou propojit. Protože konkurence nikdy nespí, může se řešit problematika, jak lze z dat *vyčíst konkurenční výhodu*. S klasickým databázovým pojetím si však vystačit nelze. Data se při řešení problémů musí přesouvat do speciálně upravených datovýh struktur, jež se pak dále používají pro analýzy a podporu rozhodování. Data se ukládají do tzv. *datových skladů*.

Definice 2.9 *Datový sklad je podnikově strukturovaný depozitář subjektově orientovaných, integrovaných, časově proměnných, historických dat, použitých na získávání a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.*

Definice vypadá komplikovaně jen zdánlivě. Při bližším průzkumu lze zjistit, že mnoho pojmů jde výjadřit intuitivně. Tato bakalářská práce se celou problematikou zabývá do hloubky a všechny klíčové pojmy budou postupně vysvětleny. Už nyní si však hlavní pozornost zaslouží pasáže o historických datech a časových proměnných. To je hlavní rozdíl oproti klasickému databázovému pojetí, kde se data pravidelně modifikují a nejen proto je hlubší analýza téměř nemožná. V praxi je potřeba zodpovědět několik otázek: *Jaké zboží se nejlépe prodávalo v uplynulých třech letech? Jak efektivně pracují zaměstnanci na svěřených úkolech? Je přijatelné udělit zákazníkovi hypotéku?*

Možných cest, jak tohle zjistit, je několik. Lze použít sofistikované statistické metody (ty v pozdějších fázích s touto problematikou úzce souvisí), či provést rozumný odhad na základě pozorování. Když je zdrojových systémů několik, je vhodné data předzpracovat a k tomu poslouží datový sklad. V tomto ohledu se lze často setkat s pojmy *tabulka fakt* a *tabulká dimenzí*, neboli zkráceně s *fakty* a *dimenzemi* v *hvězdicovém schématu* (Obrázek 1).

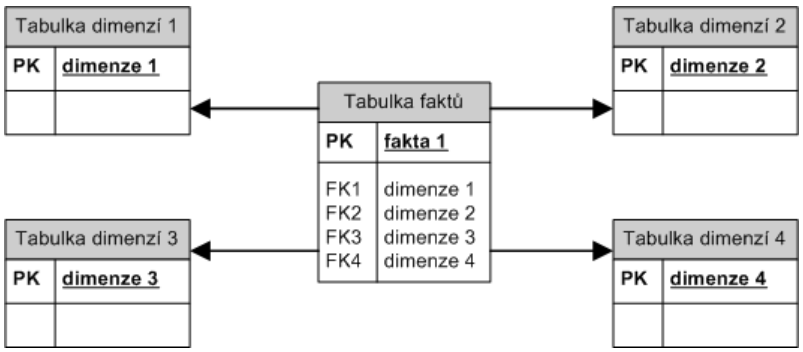
Definice 2.10 *Fakta jsou numerické měrné jednotky obchodování.*

Definice 2.11 *Dimenze obsahují logicky nebo organizačně hierarchicky uspořádaná data. Jsou to vlastně textové popisy obchodování.*

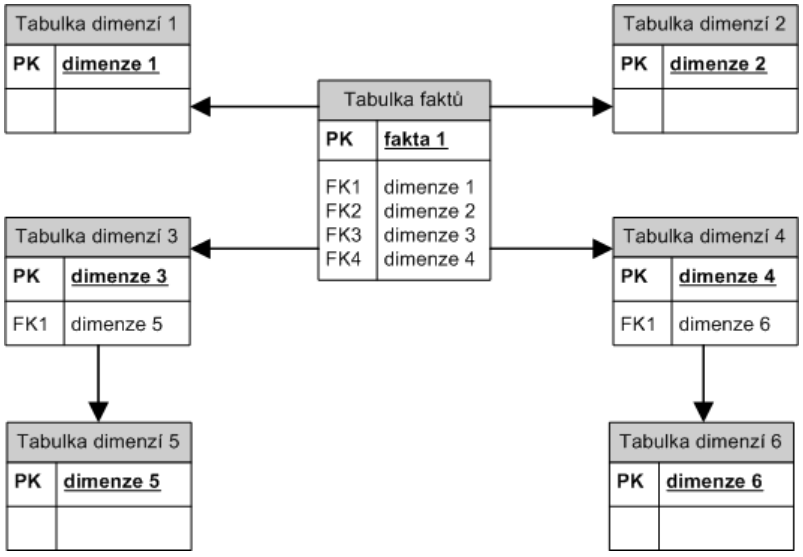
Definice 2.12 *Hvězdicové schéma se skládá z tabulky faktů a obsahuje cizí klíče, které se vztahují k primárním klíčům v tabulkách dimenzí.*

Čas od času se také mluví o schématu sněhové vločky. Situace je podobná jako u hvězdicového schématu s tím rozdílem, že tabulky dimenzí obsahují ještě další primární klíče, které se váží k dalším tabulkám. Situaci dokresluje Obrázek 2.

Při pokusu o porovnání jednotlivých přístupů je nutné si uvědomit jeden podstatný fakt. *Data v datovém skladu jsou zpravidla vysoce nenormalizovaná, zejména v tabulkách dimenzí*. Z definice 2.2 lze snadno vyčíst, proč se u OLTP přístupu podobné procesy tvoří (snaha o elimaci redundatních dat a získání přesných výstupů). U datových skladů je situace odlišná. Hlavní snahou je zjistit něco o samotných datech a co možná nejkomplexněji je analyzovat. Data v dimenzích budou *předpočítaná*. Častá bude například datumová



Obrázek 1: Ilustrace hvězdicového schématu



Obrázek 2: Ilustrace schématu sněhové vločky

Kraj	Město	Počet obyvatel
Hlavní město Praha	Praha	1 000 000
Středočeský	Dobřichovice	3 000
	Čenošice	8000
Jihomoravský	Brno	400 000
	Vyškov	30 000
Moravskoslezský	Ostrava	320 000
	Havířov	60 000

Tabulka 1: Počet obyvatel ČR v roce 1998

Kraj	Počet obyvatel
Hlavní město Praha	1 000 000
Středočeský	11 000
Jihomoravský	430 000
Moravskoslezský	380 000

Tabulka 2: Počet obyvatel ČR v roce 1998 dle krajů

dimenze, kde budou jak atributy *den*, *měsíc*, *rok*, tak i plný formát datumu ve formě *rok-měsíc-den*. Použít lze i *rok/měsíc/den*. Lehce lze také zjistit číslo kvartálu v daném roce, číslo trimestru nebo dokonce čísla týdnů a dnů v roce. Vše jde jednoduše uložit k příslušným atributům a připravit si tak podklady pro další zpracování.

Proč ale něco takového dělat? Nešlo by vše vypočítavat za chodu? Teoreticky by to mělo jít, ovšem z praktického hlediska jde o výpočetně náročný krok. Kdyby se vše počítalo až za chodu, server s celým řešením by vykazoval prudký pokles ve výkonu. Lze si představit velkou firmu, kde data dosahují často až terabajtových velikostí a jednotlivé dimenze mají klidně i více než milión záznamů. Předpočítaní těchto údajů v dimenzích je jediné řešení, jak celý proces urychlit. Lehkou paralelu lze spatřit i v jiných oborech, například v trojrozměrné počítačové grafice. Z důvodu optimalizace výkonu jsou často nejdůležitější matematické operace předpočítány a ve výsledku je tento fakt znát.

Celkovému návrhu dimenzí je také potřeba věnovat náležitou pozornost. Finální *návrh nesmí obsahovat žádné redundance mezi záznamy*. Fakta bývají zpravidla normalizována. Z praktického hlediska v letové evidenci lze mezi dimenzemi spatřit různé datumy, u faktů to jsou naopak často počty prodaných kusů zboží či ceny letenek. Ukázku jak vypadají fakta a dimenze v praxi si lze prohlédnout v Tabulce 1 a Tabulce 2. Data jsou smyšlená, důležité je si uvědomit, že fakta jsou v tomto případě pouze hodnoty atributu *Počet obyvatel*. Mezi dimenze pak patří *Kraj*, *Město* a *Rok*. Obě tabulky ukazují různý stupeň zanoření a tomu odpovídající i výsledky.

Když je hotový návrh, následuje samotná implementace. Důležitou roli hraje výběr vhodného SŘBD. Obecně na výběru prostředí až tak nezáleží, protože veškeré principy jsou univerzální a vše bude fungovat jak na MySQL, tak specializovaném MS SQL Serveru. Z praktického hlediska je nutné brát na zřetel, za jakých okolností bude daný

datový sklad používán, kolik bude obsahovat záznamů a kdo k němu bude mít přístup. Základem je implementovat definované struktury tabulek. Často stačí použít skripty pro implementaci tabulek přesně dle definovaných požadavků. Pokročilejší systémy pak nabízejí různá vylepšení jako možnost tvorby oddílů či optimalizace výkonů. Vše potom záleží na potřebách daného řešení.

Implementace definovaných struktur je však jen začátek. Komplikace nastanou v případě, kdy je potřeba celé prostředí naplnit daty ze zdrojů. Je potřeba sestavit speciální skripty, kterým se říká *datové pumpy*, neboli odborně *ETL procesy*. Ty se skládají ze tří částí - Extrakce (Extract), Transformace (Transform) a nahrávaní (Load). Význam je patrný už z názvů. Extrakce slouží k vytažení dat z různých zdrojů. V praxi lze jako zdroj použít cokoliv, od jednoduchých textových souborů, až po sofistikovaná databázová řešení. Protože původní data často nelze použít vzhledem k odlišnosti datových skladů, je nutno přistoupit k transformacím (například zjištění popisu z datumu čísla týdne v roce). Jakmile je vše připraveno, nová data se nahrají do cílového úložiště, což jsou v našem případě předem definované struktury.

Jedná se o stěžejní část celé problematiky budování datových skladů. Nejdůležitějším problémem tedy je vždy promyslet, jak data vhodně integrovat. Jakmile je ale tato část zvládnutá, samotná implementace je velmi snadnou záležitostí. Nejdůležitější roli má tak analytik, na němž závisí úspěch či neúspěch celého řešení. Tento člověk tak často komunikuje s majitelem dat a snaží se co nejvěrohodněji zachytit celou realitu. V praxi jsou bohužel data u zavedených firem velmi složitá a často je téměř nemožné napoprvé vytvořit komplexní datový sklad, který by najednou zvládal splnit všechny požadavky zadavetele. Často lze také slyšet názor, že datový sklad jako projekt nelze prakticky nikdy dokončit.

Je to možná překvapivá informace, ale část pravdy na ni je. Chce-li nějaká firma udržet krok před konkurencí, měla by jít neustále kupředu. Nové potřeby firmy je nutné zaznamenat, což má za důsledek změny v celém datovém skladu. Je nutné poznamenat, že by problematika měla být řešena co nejobecněji, protože nelze dopředu určit dotazy příslušných uživatelů. Důležitým faktem je, že datový sklad je průběžně naplňován ze zdrojových souborů, a proto je potřeba po celou dobu provozu sledovat všechny důležité uzly v celém řešení a v případě výpadků co nejrychleji zasáhnout. Z důvodu výše uvedené složitosti se v praxi setkáváme s pojmem *Datové tržiště*.

Definice 2.13 *Datové trhy jsou určité přesně specifikované podmnožiny datového skladu, které jsou určeny pro menší organizační složky firmy.*

Zjednodušeně lze říci, že datové tržiště je menší část komplexního datového skladu, která může fungovat samostatně. Chybou není, když se nově vytvořenému datovému tržišti říká datový sklad. Při vývoji se lze také setkáváme s přírustkovou metodou. Ta může být buď *zdola nahoru*, anebo *zhora dolů*. Vše souvisí s již zmíněnými trhy. U přírustkové metody zdola nahoru se řeší nejprve implementace datových trhů a až poté celého datového skladu. U metody zhora dolů pak analogicky jdou nejprve sklady a až potom tržiště. Z logiky věci je pak zřejmé, že pro implementaci je výhodnější použití přírustkové metody zdola nahoru. Jsou zde sice počáteční zvýšené náklady, avšak návratnost inves-

tice je mnohem větší a i samotné analýzy budou jednodušší, než při použití metody zhora dolů.

U velkých datových úložišť je kladen důraz na *datovou kvalitu* a *metadata*. Datová kvalita je důležitá z toho důvodu, že různorodost systémů může způsobovat různé nepřesnosti. Například pro Spojené státy americké může existovat hned několik zkratek jako *US*, *U.S.A.* či *USA*. Při podcenění této složky mohou být výsledná data hodně zkreslena. Metadata, neboli data o datech, jsou také velmi důležitou složkou, podobně jako třeba dokumentace. Metadat existuje několik druhů. Obecně jde o speciální databázi, kde jsou uloženy popisky k celému řešení datového skladu, veškeré datové toky a spousta dalších důležitých údajů. V praxi se bohužel tato složka často opomíjí, což může způsobovat problémy zejména v případech, kdy je nutné datový sklad rozšířit, nebo na projektu mají pracovat zcela odlišní lidé než ti, již se podíleli na řešení původním.

Vytvořit datový sklad je pouze začátek. S celým řešením je nutné dále pracovat. Nejčastější aplikací datového skladu je vytvoření tzv. kostky, neboli *multidimenzionální databáze*.

Definice 2.14 *Multidimenzionální databáze je forma databáze, kde jsou data uložena v buňkách a pozice každé buňky je definovaná číslem hiearchie zvané dimenze.*

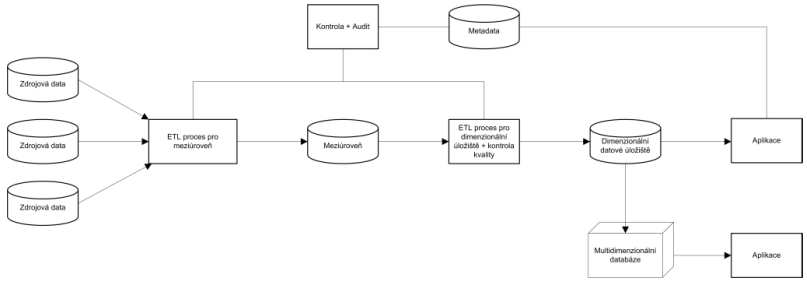
Při správném pochopení definic 2.10 pro fakta a 2.11 pro dimenze, lze bez problémů spatřit další souvislosti. Fakta jsou totiž jednotlivé buňky v multidimenzionální databázi. Tomuto způsobu uložení dat se také říká *OLAP*. Jakmile je vytvořená kostka, lze s ní provádět mnoho úkonů jako vytvořit reporty (sestavy), které si budou moci prohlédnout oprávnění uživatelé. Kostku pak lze integrovat jako součást dalších projektů a aplikací, vč. webových stránek a MS Excelu. Nad multidimenzionálními databázemi existují specialní dotazovací jazyky. Firma Microsoft vyvinula jazyk s názvem MDX, jenž je součástí MS SQL Serveru. Představen byl v roce 1997. Samostatnou kapitolou je data mining (dolování dat). Lze jej provádět samostatně nad obyčejnými textovými soubory, ale častější využití je u datových skladů a multidimenzionálních databází. Obecně je zde snaha o nalezení nových informací na základě důkladné analýzy dat s pomocí sofistikovaných algoritmů. Spousta firem si začíná uvědomovat, že bohatství jejich podnikaní je ukryto právě v samotných datech.

Celá problematika počínaje analýzou dat, přes budovaní datových skladů, až po tvorbu příslušných aplikací, je označovaná jako proces Business Intelligence (BI). Pod tímto označením se může skrývat mnoho. Spousta odborníků označuje tento pojem markentingovým trikem.

Definice 2.15 *Business Intelligence je proces transformace údajů na informace a převod těchto informací na poznatky prostřednictvím objevování*

2.2 Stručný historický vývoj a pohled do budoucna

Kapitola o historickém vývoji této problematiky byla ponechána až na konec, aby byl čtenář schopen díky předchozímu výkladu některé prvky snadněji rozlišit. Často lze



Obrázek 3: Blokové schéma dimenzionálního modelu a možné aplikace

slyšet názor, který říká, že datové sklady vrátily databáze tam, odkud vzešly. Původně se zdál návrh skladového řešení jako neřešitelný problém, i když byla definovaná poměrně striktní pravidla, jak by podobné řešení mělo vypadat.

Průlom nastal v devadesátých letech, kdy pánové Bill Inmon a Ralph Kimball úspěšně realizovali své pojetí datových skladů. Jejich definice byly lehce odlišné a zajímavostí je, že obě se podařilo úvest do praxe. Pohled Billa Inmona je velmi podobný definici 2.9. Pro úplnost je uveden názor Ralpha Kimballa.

Definice 2.16 *Datový sklad je systém, kde je ze zdrojových dat provedena extrakce, čištění, přizpůsobení a vše je doručeno do dimenzionálního datového úložiště a následné dotazy a analýzy slouží jako podklady pro podporu rozhodnutní.*

Oba pánové jsou bráni jako otcové datových skladů a oba přístupy se hodí jako řešení pro jiné oblasti. Žádný jiný obor informačních technologií nepřináší takový potenciál jako právě oblast BI. I dnes, kdy se hlasitě mluví o krizi, je zřejmé, že využití bude obrovské. Firmy různě optimalizují náklady a za všech okolností potřebují mít co nejpřesnější výsledky. BI toto dokáže splnit bez větších problémů.

3 Tvorba datových skladů krok za krokem

Následující kapitola si klade za cíl seznámit s jednotlivými fázemi při tvorbě datových skladů, od návrhu, až po aplikaci celého řešení.

3.1 Přípravná fáze, sběr požadavků a analýza

V oblasti DWH/BI nelze rozhodně podceňovat důkladnou analýzu a přípravu. Analýza je základem celého úspěchu, protože pokud se nepromyslí jednotlivé fáze vývoje, lze v lepším případě počítat se ztátou firemních zdrojů (čas a peníze). V horším případě naopak se ztátou dobré pověsti.

Jen pro zajímavost, cena za vytvoření DWH/BI řešení se pohybuje v řádech milionů korun. Velkou část těchto nákladů přitom tvoří zakoupení licencí na specializované produkty. Například nástroj z rodiny ETL od firmy BusinessObjects - Data Integrator, jde pořídit (v době psaní této Bakalářské práce) za 900 000 Kč.

Oproti klasickému databázovému přístupu je potřeba uvědomit si jeden zásadní rozdíl. Když se buduje klasický informační systém, lze vše postavit na zelené louce. *Nejsou potřeba žádná vstupní data*. Analytik příjde za majitelem firmy, sepíše podrobně všechny požadavky na systém a na základě svých odborných znalostí a zkušeností se pustí do práce, popřípadě implementací pověří spolupracovníky. Když se však staví projekt datového skladu, analytik může být v kontaktu s majitelem dané firmy, ale *musí mít také k dispozici vstupní data*. Čím více kvalitnějších dat bude, tím lépe, nicméně je nevhodné stavět podobná řešení, pokud data nejsou k dispozici.

Jsou-li data k dispozici, je potřeba vybrat metodu, s jakou se bude sklad budovat. Existuje *Metoda velkého třesku*. Neznamená to, že by se vývoj založený na této metodě prováděl zcela bez rozmyslu. Je stanovena prvotní analýza, jež se společně s dalšími podklady připraví a pak se vše implementuje téměř najednou. Převažuje velká část nevýhod. Může se stát, že se data v průběhu změní. Pak je nutné začít od začátku. Mnohem lepší je použít přírustkovou metodu. O ní byla zmínka v předchozí kapitole. Pro připomenutí lze dodat, že existují dvě verze. Shora dolů a zdola nahoru, kde se buď řeší nejprve celý datový sklad a až potom tržiště, anebo nejprve tržiště a až potom datový sklad. Bylo řečeno, že pro úspech mnoha projektů je výhodnější použití metody přírustkové metody zdola nahoru.

3.1.1 Návrh fakt, dimenzí a Slowly Changing Dimension

Když jsou připraveny požadavky a úvodní analýzy, je potřeba všechny návrhy ještě více zjemnit. Zde bude situace velmi individuální, protože bude záležet na tom, na jaké úrovni jsou vstupní data a čeho se chce při implementaci docílit. Volit se bude, zda použít k modelování hvězdicové schéma, anebo schéma sněhové vločky. V předchozí kapitole bylo nadefinováno, že hvězdicové schéma používá klasické tabulky dimenzí a fakt s tím, že použije-li se schéma sněhové vločky, může se snadno stát, že tabulky dimenzí obsahují odkazy ještě k dalším doprovodným tabulkám.

Název atributu	Datový typ
date_key	int
date	nchar(10)
ansi_date	nchar(10)
day	tinyint
day_of_the_week	tinyint
day_name	nvarchar(9)
day_of_the_year	smallint
week_number	tinyint
month	tinyint
month_name	nvarchar(9)
short_month_name	nchar(3)
quarter	nchar(2)
year	smallint
create_timestamp	datetime
update_timestamp	datetime

Tabulka 3: Návrh datumové dimenze

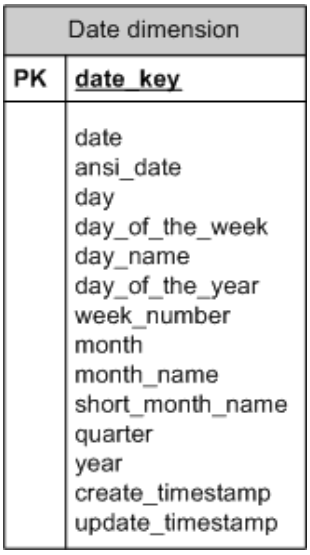
Oba přístupy jsou odlišné a oba odrážejí realitu tak, jak ji dokázali definovat pánové Kimball a Inmon. V praxi lze zaslechnout názory, že modelování za pomocí schématu sněhové vločky se použije v případě, kdy existují zavedené systémy, které spolupracují se zavedenou relační databází. V takovém případě je pak snažší takové systémy integrovat do řešení datového skladu.

Řekněme, že se navrhuje tabulka dimenzí a fakt. Pro připomenutí uveďme, že když se navrhuje tabulka dimenzí, přistupuje se k nenormalizovanému přístupu, jenž však musí být proveden velmi precizně. Jako ilustrace může posloužit Tabulka 3, kde je vymodelovaná ukázková datumová dimenze.

Jedná se o ukázkový návrh dimenze, která je použita i v doprovodném projektu této bakalářské práce. Na obrázku 4 je stejná dimenze zobrazená ve formě diagramu.

Jeden atribut *date* je zde rozložen na několik dalších. Není obtížné zjistit den v roce, stejně jako číslo týdne. Vše má praktické upotřebení. Datumová dimenze byla vybrána zcela záměrně. Při dobrém pochopení definice datového skladu (2.9), lze rozpoznat, proč je čas v těchto projektech velmi důležitý. Každé DWH/BI řešení bude totiž obsahovat alespoň jednu dimenzi s časovými údaji, proto by měla být věnována zvýšená pozornost jejímu návrhu. Ukázka výše je hodně zjednodušená. V praxi se lze často setkat s daňovým kalendářem, jehož použití má svůj význam.

Dobrá dimenze se pozná tak, že je od ostatních dobře izolována a na první pohled vypadá staticky. Bohužel v praxi je často potřeba řešit problém. V dimenzi je potřeba nějakým způsobem zaznamenat jemné rozdíly. Firma se totiž neustále vyvíjí a data se mění. Jak ale zachytit tuto skutečnost? Při vývoji datových skladů se lze často setkat s dobře zavedeným pojmem *Slowly Changing Dimension - SCD* (Pomalu se měnící dimenze). Pro lepší pochopení je zde celá problematika demonstrována na příkladu.



Obrázek 4: Schéma návrhu datumové dimenze

ID	Předmět	Garant
123	SQL pro pokročilé	Holub

Tabulka 4: Dimenze s kurzy

Mějme situaci znázorněnou v Tabulce 4. Jde o tabulku kurzů, které mají své ID, název a lektora. Časem se může stát, že tutor z nějakého důvodu opustí zaměstnání a místo něj příjde nová posila. Tato skutečnost by měla být nějak zaznamenána. Může přijít nová posila - učitel s příjmením Petržela. Nejjednodušším řešením je příslušnou hodnotu atributu přepsat. Toto je možné provést a z teoretického hlediska se jedná o SCD typu 1. Výsledek si lze prohlédnout v Tabulce 5.

Ideální řešení v tom však hledat nelze. Při implementaci SCD 1 se odstraní veškerá důležitá historie. Během roku se může prostřídat dalších pět učitelů a zaznamenán bude jen ten poslední. Lepší bude využít další typ SCD s pořadovým číslem 2. Její princip spočívá v tom, že každá změna bude evidovaná na samostatném řádku. Situaci dokresluje Tabulka 6.

V tomhto případě je situace lepší a je pravděpodobné, že se v praxi setkáme s tímto postupem. Lze však ještě použít SCD třetího typu (Tabulka 7).

ID	Předmět	Garant
123	SQL pro pokročilé	Petřzela

Tabulka 5: Dimenze s kurzy

ID	Předmět	Garant
123	SQL pro pokročilé	Holub
128	SQL pro pokročilé	Petřzela

Tabulka 6: Dimenze s kurzy

ID	Předmět	Garant	Dřívější garant
123	SQL pro pokročilé	Petřzela	Holub

Tabulka 7: Dimenze s kurzy

Oproti SCD druhého typu se liší v tom, že příslušný atribut je zaznamenán v novém sloupci. I toto řešení je bez problému použitelné, ovšem setkat se s ním lze o trochu méně, než v případě SCD 2.

Když jsou připravené dimenze, je čas začlenit i fakta. Ty bývají oproti dimenzionálním tabulkám spíše normalizované a měly by mít jednotnou granualitu. Často zde budou vystupovat různé měny a další veličiny, se kterými lze provádět výpočty. Jak taková tabulka faktů může vypadat, si lze prohlédnout v Tabulce 8.

Začleněnou tabulku fakt k naší datumové dimenzi si lze prohlédnout na Obrázku 5.

3.1.2 Doporučená literatura

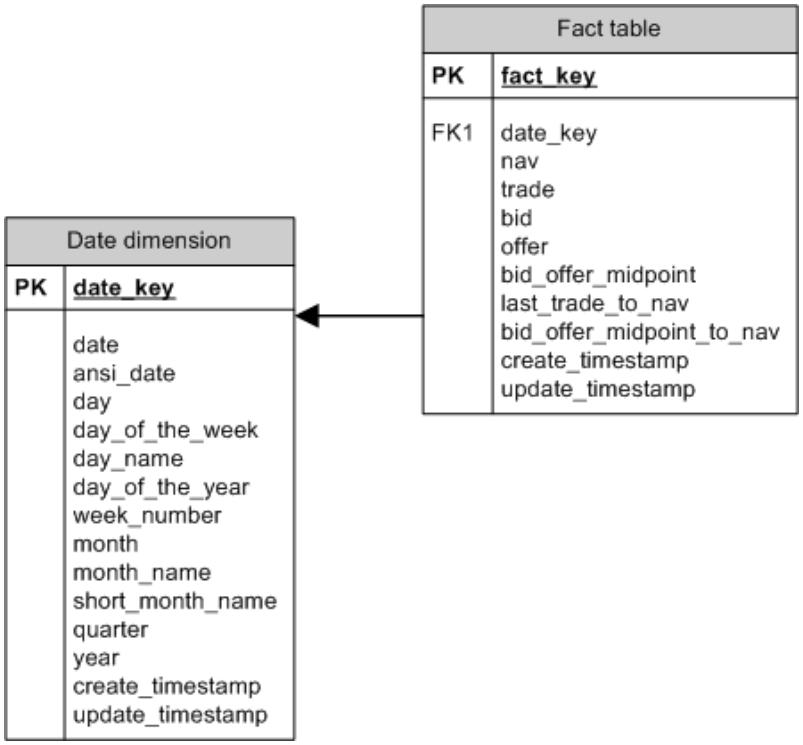
Zájemci o dimenzionální modelování najdou podrobný výklad v knize [1]. Pro úplné začátečníky publikace vhodná není, nicméně má-li čtenář alespoň minimální povědomí o problematice budování datových skladů, najde tu naprosto vyčerpávající informace. Úvodem může být také kniha [2], stejně jako [7]. V obou textech jsou důkladně probrány jednotlivé aspekty tvorby datových skladů. Pro zájemce o publikace v českém jazyce, lze doporučit díla [6] a [3].

3.2 Implementace definovaných struktur

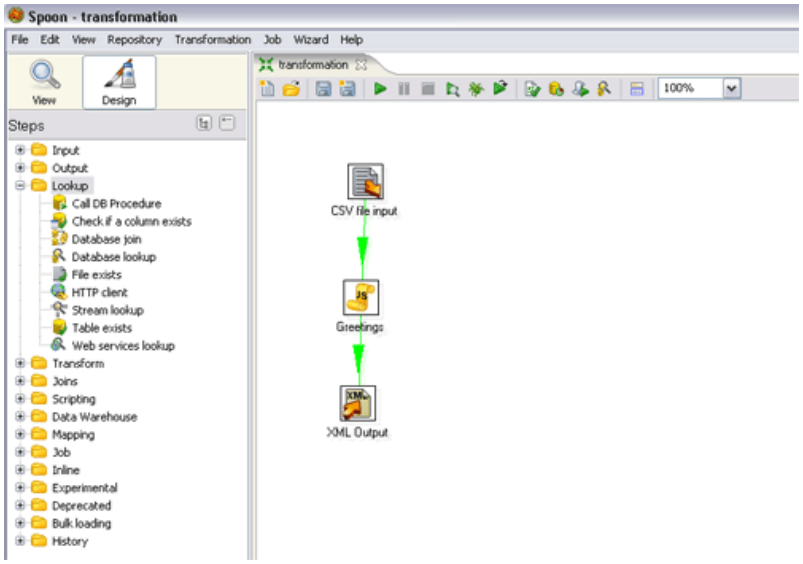
Tento krok je jednoduchý. Je-li provedena analýza, stačí vybrat vhodný SŘBD a pomocí jazyku pro definici dat, kde u SQL patří například příkazy *Create*, *Drop* a *Alter*, danou strukturu implementovat. Jak již bylo popsáno výše, na výběru SŘBD nezáleží. Je tedy jedno, zda-li se použije MySQL databáze nebo Oracle. Důležité je si uvědomit, jakým způsobem se bude s celým projektem pracovat. Moderní databázové systémy navíc obsahují grafické nástroje k tomu, aby šlo vše vytvořit velmi jednoduše. Doprovodný projekt k této Bakalářské práci je implementován v MS SQL Server 2005 - Developer Edition. Ve výpisu 1 je pro ilustraci vypsán zdrojový kód v SQL jazyce, jenž vytváří dříve definovanou datumovou tabulku.

Název atributu	Datový typ
fact_key	int
date_key	int
nav	smallmoney
trade	smallmoney
bid	smallmoney
offer	smallmoney
bid_offer_midpoint	smallmoney
last_trade_to_nav	smallmoney
bid_offer_midpoint_to_nav	smallmoney
create_timestamp	datetime
update_timestamp	datetime

Tabulka 8: Návrh tabulky faktů



Obrázek 5: Ilustrace schématu sněhové vločky



Obrázek 6: Prostředí ETL nástroje Pentaho Kettle - Spoon

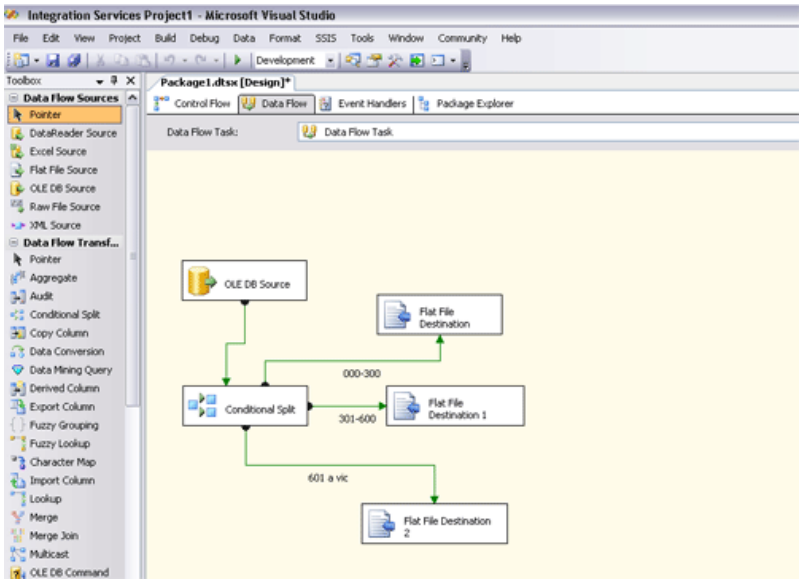
3.3 Sestavení datových pump - ETL procesů

Nyní bude následovat mnohem náročnější problém. Hlavním úkolem bude definované struktury naplnit. K tomuto účelu se vytváří tzv. datové pumpy, neboli ETL procesy. Jde o sadu speciálních skriptů, jež vytáhnou data z produkčních systémů (extract), transformují je (transform) a nahrají do cílové databáze, kde jsou uloženy dimenzionální struktury (load). Čas od času se lze setkat s označením ELT. Jediný rozdíl je ten, že data se v cílové databázi transformují až po celkovém nahrání (například s pomocí uložených procedur).

Jako produkční systém může být použito cokoliv. Datový sklad také vyžaduje pravidelný přísun dat, takže je nutné sestavit časové periody, ve kterých se budou dané skripty provádět. Otázkou ale zůstává, jak postupovat při tvorbě takových skriptů? Je-li produkční a cílová databáze na jednom stroji, lze bez větších problémů sáhnout po uložených procedurách nebo triggerech a použít procedurální nadstavby SQL dotazovacího jazyka. Vzhledem k faktům, že v praxi jde pouze o sled několika základních postupů. Dnešní pokročilá databázová prostředí vybízejí k tomu, aby se v nich programovalo.

Mnohem lepší je využít specializované ETL nástroje, které se mnohdy dodávají jako součásti instalací velkých SŘBD. Firma Microsoft vyvinula komponentu pro Visual Studio s názvem SQL Server Business Intelligence Development Studio, Oracle naopak používá Warehouse Builder. Možná pro někoho méně známé, avšak ve světě hojně používané produkty jsou nástroje firmy Informatica s názvem Powercenter a Data Integrator od firmy BusinessObjects. Často jde o špičkové komerční nástroje. Za zmínku také stojí produkt Data Integration, který je součástí většího open-source projektu Pentaho a jehož kvalita je na velmi vysoké úrovni.

Obrázky 6 a 7 pro porovnání zobrazují prostředí produktů Pentaho Kettle a Microsoft Business Intelligence Development Studio.



Obrázek 7: Prostředí ETL nástroje Business Intelligence Development Studio

Jde o nástroje konkurenčních produktů, nesou však spoustu společných rysů. V mnoha knihách lze najít spoustu zajímavých výroků, ocitujme si jednu krásnou myšlenku, s níž se lze setkat v publikaci [5]. *Soustřeďte se na koncept nebo techniku a zamýšlený výsledek, nikoliv na příklad, který je má demonstrovat*. Jakmile celou problematiku vývojář správně pochopí, na vybraném ETL nástroji nebude vůbec záležet.

Pozornost je nutné věnovat datové kvalitě. Může se stát, že určitý údaj bude v produkční databázi uložen s více odlišnými hodnotami. Například Spojené státy americké mohou mít zkratku *US*, *U.S.A.*, *USA* a mnohé další. V datovém skladu by však vše mělo být sjednocené, proto je potřeba být co nejpečlivější.

3.3.1 Doporučená literatura

Produkt Business Intelligence Development Studio od firmy Microsoft je velmi dobře popsán v publikaci [7].

3.4 Metadata

Fotografie, webové stránky, hudba, filmy. To je jen drobný výčet subjektů, kde se lze setkat s pojmy Metadata. Jsou to *data o datech*, tedy popisky, jež charakterizují daný subjekt. Těchto metadat existuje několik druhů, nejčastěji je pomocí nich popsán určitý datový tok nebo tabulka faktů či dimenzí. Obecně jde o speciální databázi, která v určitém ohledu zastupuje dokumentaci. V praxi se často na tvorbu těchto popisků neklade takový důraz a větší projekty jsou často částečně nebo zcela bez metadat.

Jak ale takové údaje zaznamenat? Většinu důležité práce obstarají samotné ETL nástroje. Existuje ale spousta software třetích stran, které se specializují pouze na tuto

oblast. I rodina produktů Pentaho má svého zástupce pro metadata. Díky přehlednému interface je pak práce s nimi velice komfortní. Některá metadata se ale generují automaticky.

3.4.1 Doporučená literatura

Na problematiku metadat je vhodná publikace [7]. Spíše teoreticky, ovšem velmi dobře, je vše popsáno v knize [1].

3.5 Aplikace datových skladů

Když už je datový sklad hotov, je potřeba ho smysluplně využít. Možností je několik a níže budou probrány ty nejzákladnější.

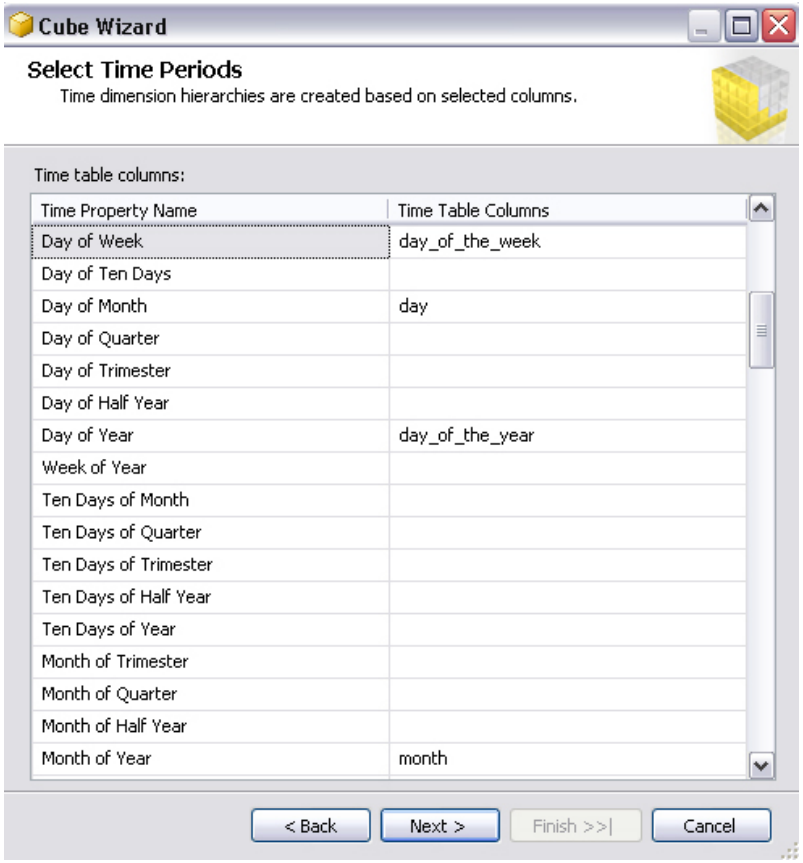
3.5.1 Multimenzionální databáze

První aplikací je tvorba multidimenzonální databáze, neboli kostky OLAP. Velmi záleží na konkrétních potřebách, obecně jsou tyto úložiště velmi škálovatelné. Mnoho aplikací navíc podporuje OLAP klienty (například většina produktů od firmy Microsoft), a tak je kostky možné integrovat jako součást i jiných produktů.

Tvorba analytických databází je poměrně robustní, protože data lze načítat přímo z datového skladu, anebo z jakéhokoliv dalšího zdroje (druhá varianta se hodí pro aplikaci data miningu, o které bude řeč později). Díky mnohým průvodcům pak nebude problém vytvořit správný cílový objekt. Nástrahy číhají pouze v podobě správné identifikace faktů a dimenzí. Je-li datový sklad korektně navržen, neměl by být problém vše správně rozpoznat (systém spoustu věcí identifikuje automaticky, přesto není na škodu být pečlivý a vše řádně překontrolovat), stejně jako by neměl být problém správně identifikovat datumové atributy.

Nejdůležitějšími veličinami, při budování datových skladů a OLAP databází, jsou datum a čas. Dané implementační nástroje s tím počítají a nabídnou vhodné prostředky pro výběr korektních atributů. Jako příklad lze uvést analytické služby Business Intelligence Development Studia (Obrázek 8).

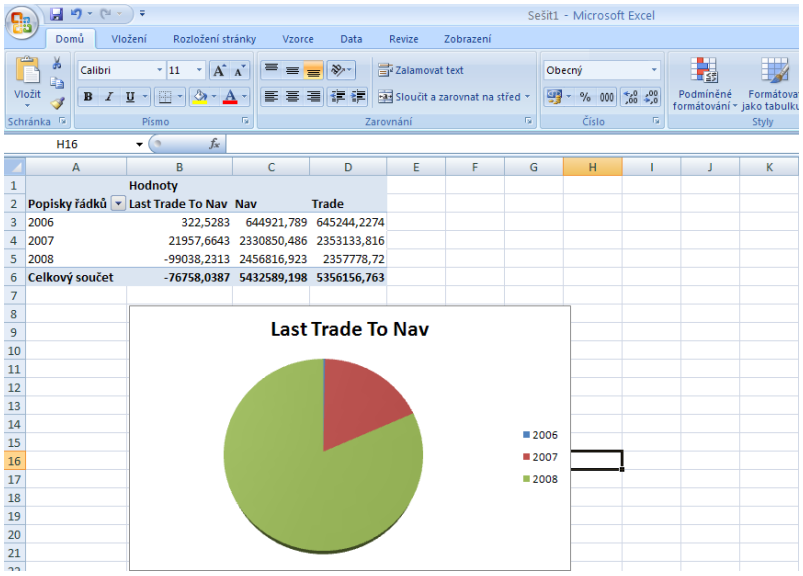
Čistý OLAP je však spíše teoretický pojem. V praxi se jde spíše setkat s pojmy *Multidimensional Online Analytical Processing (MOLAP)*, *Relational Online Analytical Processing (ROLAP)* a *Hybrid Online Analytical Processing (HOLAP)*. Když se mluví o OLAP, nejčastěji se používá tradiční MOLAP. V praxi se vytvoří multidimenzionální kostka, data se zagregují a nejsou-li na systém nějaké zvláštní požadavky, lze pak danou databázi používát nezávisle na původním datovém skladu. Je-li potřeba udržovat kontakt s původním zdrojem, pravděpodobně se zvolí ROLAP. Může se totiž stát, že z nějakého důvodu bude potřeba, aby kostka byla aktualizovaná při každé změně datového skladu. V takovém případě ovšem není problém sáhnout i po HOLAPu, jenž kombinuje přístupy obou předchozích řešení. Lze počítat s mnohonásobným nárustem výkonu, oproti přímému získávání dat z datového skladu. Příčinou nárustu výkonu je právě předpočítaní dat.



Obrázek 8: Ukázka dialogu pro výběr časových atributů

			Trade	Bid Offer Midpoint	Last Trade To Nav	Bid Offer Midpoint To Nav	Nav
2006	Q2	Apr	26231.87	26264.655	152.1577	184.9427	26079.7122
2006	Q2	Jun	59495.0701000001	59674.45	-8.64390000000006	170.816	59503.6343
2006	Q2	May	57477.6699	57487.715	50.7571000000001	60.8019000000001	57426.9125
2006	Q3	Aug	79608.3700999999	79801.8699999999	34.6637	228.1636	79573.7064
2006	Q3	Jul	66438.9100000001	67038.17	29.7831	629.0432	66409.1269
2006	Q3	Sep	70422.0802	70419.1449999999	15.5135	12.5778	70406.5666999998
2006	Q4	Dec	100409.9324	101798.0301	6.42989999999998	1394.5774	100403.4527
2006	Q4	Nov	97710.4464000002	97786.235	22.6215999999999	98.1903000000002	97688.0446
2006	Q4	Oct	87449.8783000001	87476.255	19.2456000000001	45.6221999999999	87430.6327000001
2007	Q1	Feb	143713.8611	143845.4098	32.025	163.4838	143681.9262
2007	Q1	Jan	107550.8365	107593.3542	27.2318999999999	69.7191000000001	107523.6345
2007	Q1	Mar	184646.7379	185926.805	-16.7578	1263.2594	184663.5458

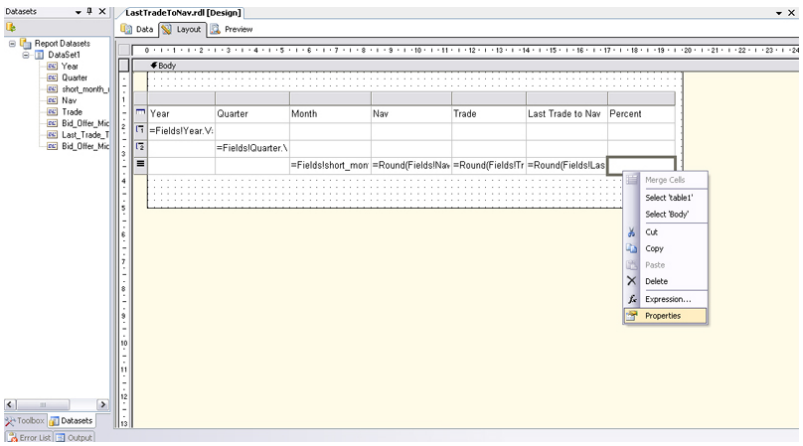
Obrázek 9: Výsledek ukázkového MDX dotazu



Obrázek 10: Možná aplikace OLAPu v Excelu

Jak k datům v těchto úložištích přistupovat? U klasického OLTP přístupu problém nebyl, stačilo použít dotazovací jazyk SQL. Lze ho však použít i u OLAP přístupu? Bohužel nelze. Je to z důvodu přítomnosti jiného typu úložiště, pro které nebyl tento jazyk navržen. Teoreticky by to bylo možno, ovšem mnohem důmyslnější a vhodnější je využít specializované dotazovací jazyky. Firma Microsoft například vyvinula dotazovací jazyk *MDX*, jenž lze najít i u mnoha dalších nástrojů jiných výrobců (Pentaho) a je pravděpodobné, že jeho pozice bude do budoucna posilovat. Ukázku, jak MDX dotaz může vypadat, si lze prohlédnout ve výpise 2. Výsledek lze spatřit na Obrázku 9.

OLAP lze integrovat do zcela jiných produktů, které disponují tzv. OLAP klientem. U firmy Microsoft se v tomto ohledu často používá klasický MS Excel. Na verzi pochopitelně nezáleží. Vhodná je verze 2003. Navíc díky potenciálu, jaký v sobě ukrývá právě Excel v podobě kontingenčních tabulek, grafů a VBA skriptů, má vývojář téměř neomezené možnosti využití. Jednoduchou aplikaci OLAP přístupu si lze prohlédnout na Obrázku 10.



Obrázek 11: Reportovací služby SQL Serveru

Možností je mnohem více. Toto byl jen stručný přehled, který je však v praxi nejčastěji používán. Zájemci o hlubší studium nechť nehlédnou mezi publikace z doporučené literatury.

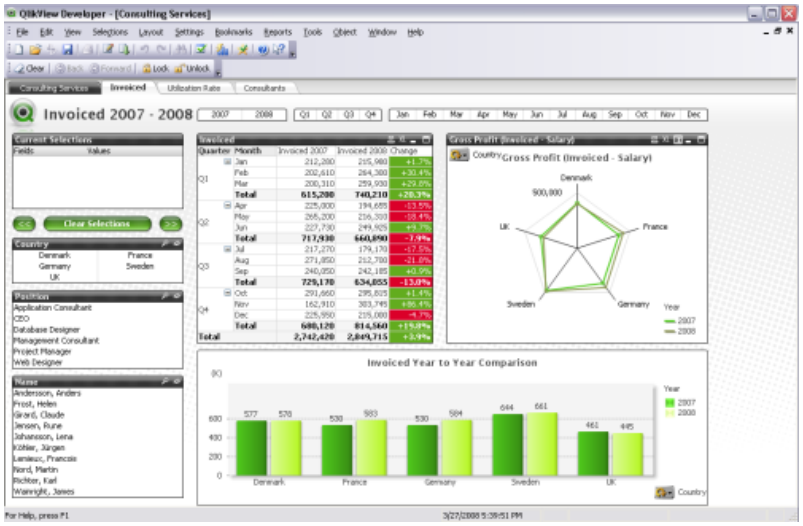
3.5.2 Reporty

Reporty neboli sestavy jsou další možností, jak aplikovat datový sklad. Existuje mnoho specializovaných nástrojů, firma Microsoft například jako součást svého řešení dodává poměrně robustní projekt s názvem Reporting Services. Sestavy jsou často spojovány s tiskem. Samozřejmě nic nebrání tomuto využití, nicméně doba pokročila a s reporty toho lze provádět mnohem více.

Primárním účelem reportovacích nástrojů je doručit správné výsledky lidem, kteří s nimi dále pracují. Například majitelé autopůjčoven bude nejčastěji zajímat, o jaká auta byl největší zájem v posledních dvou letech. Vhodný výstup pak mnohé napoví. Nějak se však musí daná sestava k cílovému uživateli dopravit. V praxi se často vytvoří specializovaný reportovací server, jenž cílové sestavy vhodně distribuuje. Špičkové produkty pak umí rozlišit uživatele dle přístupových oprávnění.

Výsledný report může být téměř v jakémkoliv formátu, nejčastěji však mají uživatelé reportovací klienty, kde mohou daný report interaktivně ovládat. Reporty jsou totiž statické a dynamické. Ta první skupina je téměř ideální pro tiskový výstup, zbytek je však vhodnější pro dynamickou práci. Sestavy lze také obohatit o grafy a ty nejlepší nástroje dokonce umožňují, aby si uživatel upravoval strukturu dat přímo za běhu (ad-hoc). To se hodí pouze v případě, kdy je cílový uživatel zkušený. Reporty lze vytvářet nad jakýmkoliv zdrojem (vč. relační databáze). Z důvodu rychlosti je však při práci s BI vhodné použít OLAP databázi. Jak vypadá prostředí pro návrh reportu v Business Intelligence Development Studio, si lze prohlédnout na Obrázku 11.

Pro porovnání je zde přiloženo prostředí produktu Qlikview a práce s ukázkovou strukturou dat amerických prezidentů. Vše je na Obrázku 12.



Obrázek 12: Ukázka návrhu reportu v Qlikview

Při práci s Qlikview lze počítat s vyšší hardwarovou náročností. Na druhou stranu se výrobce Qliktech snaží o to, aby práce s celým nástrojem byla co nejintuitivnější. Jednak můžete použít jakýkoliv datový zdroj a veškeré transformace můžete provádět přímo za chodu. Není možné tvořit ani OLAP kostky. Vše je ale zatíženo obrovskou hardwarovou náročností. Na druhou stranu jde o alternativní a velmi zajímavý přístup. Z dalších nástrojů stojí za zmínku produkty firem Cognos, Business Objects nebo Advizor.

3.5.3 Data mining

Data mining v posledních letech zažívá opravdový nárust popularity u odborné veřejnosti. Každý by chtěl najít ve svých datech skrytý poklad, jak o tom barvitě píše nejedna marketingová příručka pro Data mining. Takový poklad se sice ne vždy podaří najít, přesto je nutné přiznat, že jde o velmi pozoruhodné techniky, které lze provádět nejen nad datovými sklady či OLAP kostkami, ale také nad obyčejnými dokumenty. Základem, stejně jako ve zbylých partiích informatiky, jsou pečlivě vybranné matematické postupy, zejména z oblasti statistiky. O popularitě svědčí i fakt, že například v analytických službých SQL Serveru, jehož součástí je právě i data mining, je s každou novou verzí implementováno několik dalších algoritmů. Velmi častou technikou je použití tzv. rozhodovacích stromů. I produkty rodiny Oracle mají svou sadu standardních nástrojů.

Existují specializované utility, které implementují ne tak známé algoritmy. Jedním z nich je například produkt KXEN, jenž má implementovanou pouze Vapnikovou statistickou teorii učení. Stručně řečeno jde o metodu, která má pečlivě zmapovány jak stávající data, tak i data nově přibývající. Na základě speciálního algoritmu dokáže z těchto dat KXEN vybrat optimální vzorek dat, jež je vhodný pro další účely. Podobných specializovaných frameworků pak existuje celá řada. Jejich výhoda je snadná integrovatelnost do jiných řešení a škálovatelnost.

Při používání produktů firmy Microsoft, jistě nikomu neunikne pojem *DMX*. Jak lze tušit, jedná se o další dotazovací jazyk, který plní podobnou funkci jako SQL nebo MDX. Primární cíl je však oblast data miningu, kde základem je potřeba mít dostatečný vzorek dat, což je stejné jako u datových skladů. Čím kvalitnější pak tato data budou, tím lepších výsledků lze dosáhnout.

3.5.4 Doporučená literatura

Je více než zřejmé, že vybudovat datový sklad je pouze počátek celého úsilí. Mnohem důležitější je pak smysluplně rozhodnout, jak lze celý projekt využít. Pro hlubší pochopení OLAP problematiky je opět výborným zdrojem kniha [7], jež poskytne solidní úvod také do problematiky reportovacích služeb. Pro reporty lze doporučit i publikaci [6], kde je popsaná problematika všech důležitých aplikací Business Intelligence včetně úvodu do data miningu.

4 Amex - praktická aplikace teoretických poznatků

Tato bakalářská práce obsahuje, kromě teoretické části, také část praktickou. Ta je součástí uživatelské příručky, která je k dispozici na přiloženém CD. Mimo teoretický popis bylo nutné vytvořit doprovodnou příručku, jež by čtenáři na vhodných příkladech demonstrovala nutné procesy pro sestavení jednoduchého řešení datového skladu. Problémem byla data. Ta jsou pro úspěšný datový sklad nezbytnou součástí. Najít vhodný volně přístupný zdroj nebylo snadné. Použita nakonec byla data z Americké burzy cenných papírů (American Stock Exchange - Amex). Data byla zaměřená na podílové fondy, které mají k dispozici své akcie a jsou obchodovány na speciálních burzách. Nesou označení ETF (Exchange Traded Funds).

I přes některá omezení se práce snaží pokrýt veškeré důležité aspekty tvorby datových skladů. Příručka je rozdělená na několik kapitol. První dvě sekce se věnují obecnému popisu dat a architektury. Zdrojem jsou dva xls soubory obsahující přibližně 120 000 záznamů. Třetí kapitola je zaměřená na datové modelování. Výsledkem jsou dvě tabulky dimenzí (datumová a subjektová dimenze) a jedna tabulka fakt. Z důvodu snížení hardwarové zátěže je implementační část úměrně rozložená na několik částí. Dva zdrojové xls soubory jsou převedeny do databáze v MS SQL 2005 a sjednoceny. Tabulky fakt a dimenzí jsou implementovány dle analýz. Jako ETL nástroj použit Business Intelligence Development Studio (BIDS).

Zdrojová data obsahovala množství záznamů s NULL hodnotami. Pomocí jazyka pro definici dat byl vytvořen pohled nad zdrojovou databází, jenž pracoval pouze s úplnými záznamy. V BIDS byly sestaveny celkem čtyři ETL procesy - konsolidace zdrových dat (Obrázek 13), naplnění subjektové dimenze (Obrázek 14), naplnění datumové dimenze (Obrázek 15) a naplnění tabulky fakt (Obrázek 16).

Nad datovým skladem je postaveno multidimenzionální úložiště, které slouží jako podklad pro další aplikace. OLAP úložiště slouží jako zdroj dat pro reporty vytvořené v reportovacích službých SQL Serveru 2005. Díky předpočítaným a zagregovaným hodnotám v OLAP databázi je znát znatelný nárust výkonu oproti OLTP úložišti. Aplikován byl i externí produkt Microsoft Excel, a to ve verzi 2003 a 2007. V obou případech byl použit integrovaný OLAP klient a výstupem byla, vyjma kontigenčních tabulek, také dvojice grafů.

Přiložené CD médium obsahuje navíc videotutoriály ve flashovém formátu (SWF), které krok za krokem zobrazují sestavení ETL procesů, OLAP úložiště a reportů.

5 Závěr

Úkolem této bakalářské práce bylo získat přehled v oblasti datových skladů a Business Intelligence a na několika názorných příkladech demonstrovat použití jednotlivých technik. Implementační část si lze prohlédnout na přiloženém médiu.Protože jde o velmi rozsáhlou problematiku, kladl jsem důraz na to, aby byla podána co nejsrozumitelnějším způsobem. Velmi důležité bylo získat vhodná data. K tomuto účelu nakonec posloužila americká burza cenných papírů (AMEX), kde nebyl problém zajistit dostatečné množství dat. Z velké části šlo sice o již předzpracována data, takže plná síla dimenzionálního modelování demonstrována nebyla, nicméně všechny klíčové prvky tvorby datových skladů byly provedeny.

Praktická část byla tvořena v prostředí Business Intelligence Development Studio, kde byly použity komponenty *Integration Services*, *Analysis Services* a *Reporting Services*. Podařilo se mi však zjistit, že BI řešení nemusí být postaveno pouze na produktech jedné firmy, ale že lze bez problémů použít kombinace několika nástrojů různých firem. Díky této provázanosti lze ušetřit značné množství finančních prostředků, protože například takový MS Excel, kde lze aplikovat OLAP struktury, či data mining, je dostupný za minimální finanční náklady.

Přínos implementace podobných řešení je pak více než značný a v případě velkých firem dokáží datové sklady, i přes počáteční značné investice, ušetřit velmi výrazné finanční prostředky. Do budoucna pak lze počítat se stále větším rozšiřováním oblasti BI, protože příslušné technologie se stávají stále více dostupnější a uživatelé stále více kvalifikovanější.

6 Reference

[1] Kimball, Ralph; Ross, Margy Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, John Wiley & Sons, 2002. ISBN 978-0471200246

[2] Humphries, Mark, *Data warehousing - návrh a implementace*, Computer Press, 2002. ISBN 80-7226-560-1

[3] Lacko, Luboslav, *Databáze: datové sklady, OLAP a dolování dat*, Computer Press, 2003. ISBN 80-7226-969-0

[4] Groff, James; Weinberg, Paul, *SQL - kompletní průvodce*, Computer Press, 2005. ISBN 80-251-0369-2

[5] Hernandez, Michael, *Návrh databází*, Grada, 2006. ISBN 80-247-0900-7

[6] Lacko, Luboslav, *Business Inteligence v SQL Serveru 2005*, Computer Press, 2006. ISBN 80-251-1110-5

[7] Rainardi, Vincent, *Building a Data Warehouse: With Examples in SQL Server*, Apress, 2007. ISBN 978-1590599310

[8] Informatica, http://www.informatica.com/Pages/

[9] Pentaho, http://www.pentaho.com/

[10] Cognos, http://www.cognos.com/

[11] Advizor, http://www.advizorsolutions.com/

[12] MySQL 5.4, http://www.mysql.com/

[13] PostgreSQL 8.4, http://www.postgresql.org/

[14] Oracle 11g, http://www.oracle.com

[15] Microsoft SQL Server 2005, http://www.microsoft.com/cze/windowsserversystem/sql

[16] BusinessObjects, http://www.sap.com/solutions/sapbusinessobjects/

[17] KXEN, http://www.kxen.com/

[18] SpagoBI, http://spagobi-info.eng.it/ecm/faces/public/guest/home/solutions/spagobi

[19] JasperSoft Business Intelligence Suite, http://www.jaspersoft.com/

[20] Qlikview, http://www.qlikview.com/

A Vybranné zdrojové kódy

```
create table dim_date
( date_key  int  not null  identity(1,1)
, date  nchar(10)  not null
, ansi_date nchar(10)  not null
, day  tinyint  not null
, day_of_the_week tinyint  not null
, day_name nvarchar(9) not null
, day_of_the_year smallint not null
, week_number tinyint not null
, month tinyint not null
, month_name nvarchar(9) not null
, short_month_name nchar(3) not null
, quarter  nchar(2) not null
, year  smallint  not null
, create_timestamp     datetime    not null
, update_timestamp     datetime    not null
, constraint pk_dim_date
  primary key clustered (date_key)
)
```

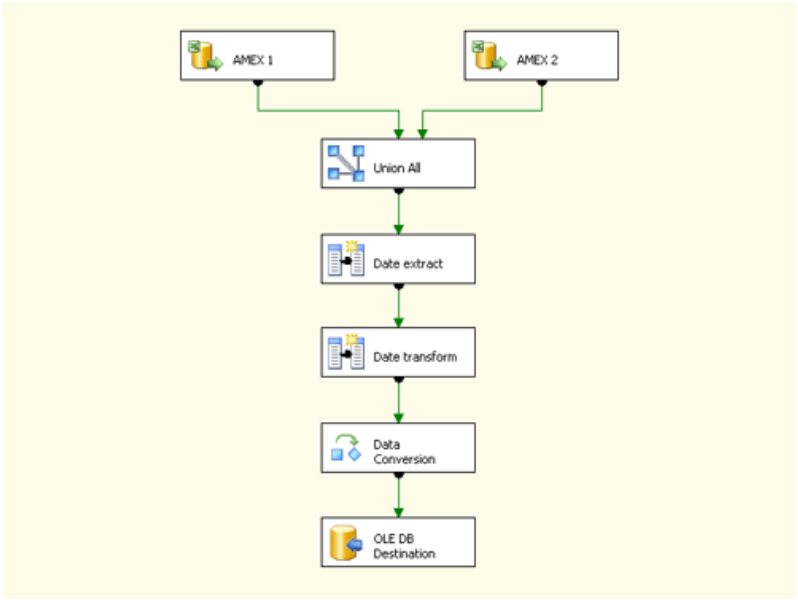
Výpis 1: Implementace datumové dimenze v SQL Serveru 2005

```
SELECT
   { [Measures].[Trade], [Measures].[Bid Offer Midpoint],  [Measures].[Last Trade To Nav],
   [Measures].[Bid Offer Midpoint To Nav],  [Measures].[Nav] } ON COLUMNS,
   { ([Dim Date].[Year].[Year].ALLMEMBERS * [Dim Date].[Quarter].[Quarter].ALLMEMBERS *
  [Dim Date].[short_month_name].[short_month_name].ALLMEMBERS ) } ON ROWS
FROM [Dwproject]
)
```

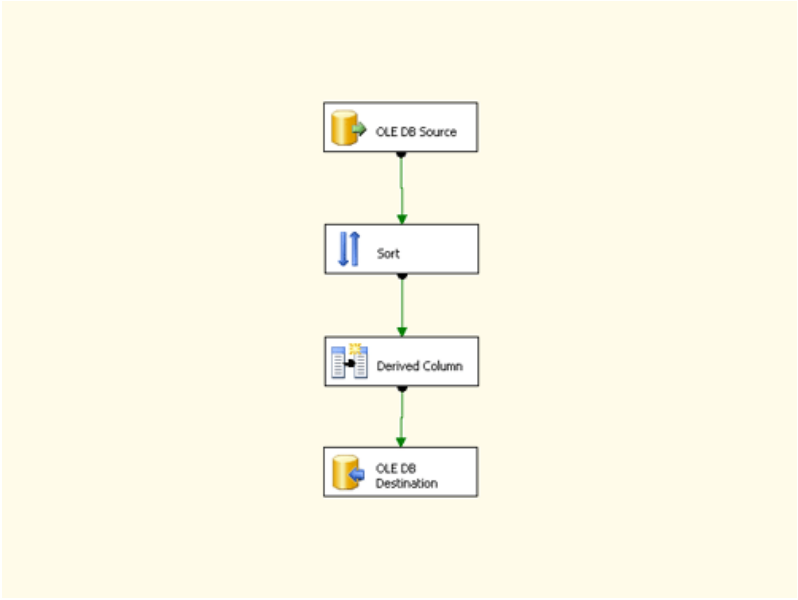
Výpis 2: Ukázka MDX dotazu

B ETL procesy použité v projektu AMEX

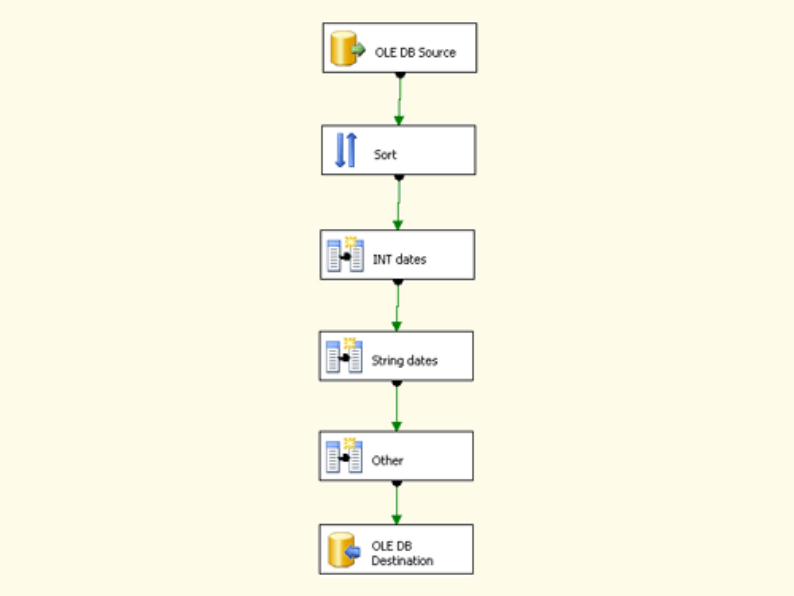
Pro úplnost je zde uveden grafický přehled sestavených ETL procesů, použitých v praktické části této bakalářské práce.



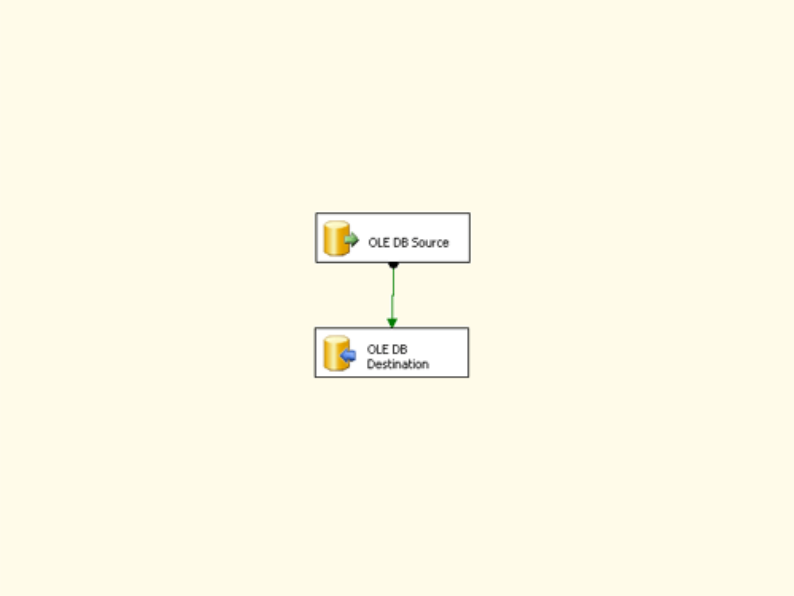
Obrázek 13: Sloučení datových zdrojů a přesun do databáze



Obrázek 14: Naplnění subjektové dimenze



Obrázek 15: Naplnění datumové dimenze



Obrázek 16: Naplnění tabulky fakt