

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DATOVÉ SKLADY A OLAP V PROSTŘEDÍ MS SQL SERVERU

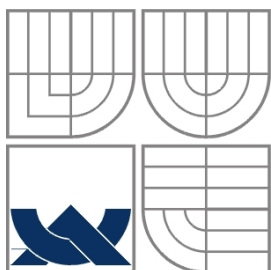
DIPLOMOVÁ PRÁCE

MASTER'S THESIS

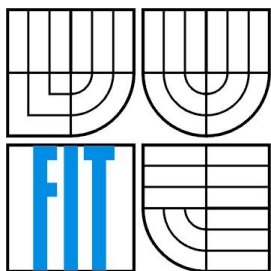
AUTOR PRÁCE
AUTHOR

Bc. LUKÁŠ MADRON

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DATOVÉ SKLADY A OLAP V PROSTŘEDÍ MS SQL SERVERU

DATA WAREHOUSES AND OLAP IN MS SQL SERVER ENVIRONMENT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. LUKÁŠ MADRON

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VLADIMÍR BARTÍK, Ph.D.

BRNO 2008

Abstrakt

Práce se zabývá problematikou datových skladů a OLAP. Tyto technologie jsou zde definovány a popsány. Dále je představena architektura produktu MS SQL Server a jeho nástroje pro práci s datovými sklady a OLAP. Získané poznatky jsou využity k vytvoření ukázkové aplikace.

Klíčová slova

Datový sklad, datový trh, etapa ETL (extract, transform, load), OLTP (online transaction processing), OLAP (online analytical processing), multidimenzionální databázový model, datová kostka, informační systémy pro podporu rozhodování, business intelligence, Microsoft SQL Server, SSIS (integrační služby MS SQL Serveru), SSAS (analytické služby MS SQL Serveru), SSRS (reportovací služby MS SQL Serveru).

Abstract

This paper deals with data warehouses and OLAP. These technologies are defined and described here. Then an introduction of the architecture of product MS SQL Server and its tools for work with data warehouses and OLAP follow. The knowledge gained is used for creation of sample application.

Keywords

Data warehouse, data mart, ETL (extract, transform, load), OLTP (online transaction processing), OLAP (online analytical processing), multidimensional database model, data cube, decision support systems, business intelligence, Microsoft SQL Server, SSIS (MS SQL Server Integration Services), SSAS (MS SQL Server Analysis Services), SSRS (MS SQL Server Reporting Services).

Citace

Madron Lukáš: Datové sklady a OLAP v prostředí MS SQL Serveru. Brno, 2008, diplomová práce, FIT VUT v Brně.

Datové sklady a OLAP v prostředí MS SQL Serveru

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Vladimíra Bartíka, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Madron
19.5.2008

Poděkování

Moje poděkování patří vedoucímu práce Ing. Vladimíru Bartíkovi, Ph.D. za odbornou pomoc a připomínky.

© Lukáš Madron, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	5
1 Úvod	7
2 Business Intelligence (BI)	8
2.1 Definice	8
2.2 Hierarchie informačních úrovní.....	8
2.3 Vývoj systémů pro podporu řídicích činností.....	9
2.4 Architektura systémů Business Intelligence.....	10
2.5 Nasazení Business Intelligence	11
3 Datový sklad.....	13
3.1 Definice	13
3.2 Procesní schéma	14
3.3 Datové trhy	15
3.4 Budování datového skladu	15
3.5 Přístupy k řešení datové architektury	17
3.6 Plnění skladu – etapa ETL.....	18
3.7 Provoz.....	18
4 OLAP	20
4.1 Definice	20
4.2 Rozdíl mezi OLTP a OLAP	20
4.3 Příprava dat pro OLAP analýzy	21
4.4 Multidimenzionální databáze a databázový model	21
4.5 Fakta, dimenze a schémata tabulek dimenzí	22
4.6 Úložiště multidimenzionálních dat.....	22
5 MS SQL Server	23
5.1 Integrovaná služba (SSIS).....	24
5.2 Analytické služby (SSAS).....	25
5.3 Reportovací služby (SSRS)	26
6 Ukázková aplikace	27
6.1 Vybudování datového skladu pomocí SSIS	27
6.1.1 Strategie.....	27
6.1.2 Definice	27
6.1.3 Analýza	28
6.1.4 Návrh.....	28
6.1.5 Sestavení	34

6.1.6	Produkce.....	36
6.2	Vytvoření datové kostky v SSAS	37
6.2.1	Datové zdroje	37
6.2.2	Pohledy na datové zdroje	37
6.2.3	Vypočítané atributy	37
6.2.4	Datová kostka.....	38
6.2.5	Hierarchie dimenzí.....	38
6.2.6	Cacheování, nastavení míry agregace	39
6.3	Tvorba, správa a distribuce reportů s SSRS	41
6.3.1	Filozofie reportovacích služeb	41
6.3.2	Návrh reportu	43
6.3.3	Správa reportu	45
6.3.4	Možnosti doručení reportu	45
6.4	Prezentace dat v klientských aplikacích.....	45
6.4.1	Windows aplikace	46
6.4.2	Webová aplikace	47
7	Závěr.....	48
	Literatura.....	49
	Seznam příloh	51
	Manuál	54

1 Úvod

V dnešní době jsou informační technologie nasazovány do všech oblastí lidské činnosti. Velkou část aplikací pak tvoří tzv. datově náročné, které při svém provozu produkují velké množství rozličných údajů ukládaných do databází. Jde především o různé informační systémy podniků, organizací nebo třeba státní správy.

Primárním účelem většinou bývá, aby tyto systémy zajišťovaly základní chod těchto institucí, což většinou také plní. Pokud však již máme takto fungující systém, generuje nám v čase obrovské množství historických dat, která již nemusí být pro samotný provoz důležitá, ale mohou obsahovat velkou informační hodnotu. Zde se dostávají ke slovu systémy Business Intelligence.

Tato práce se zabývá především problematikou datových skladů a OLAP. Tyto technologie jsou v prvních kapitolách podrobněji popsány a zasazeny do kontextu technologií podporujících procesy rozhodování, dnes často označované jako Business Intelligence.

V následujících kapitolách je představeno prostředí MS SQL Serveru a jeho prostředky pro tvorbu datových skladů, OLAP analýzu a také pro prezentaci dat, která tyto technologie využívá.

V závěru je popsána aplikace, která demonstruje využití popsaných prostředků. Teoreticky vychází z předcházejících kapitol, které byly zpracovány v rámci semestrálního projektu.

2 Business Intelligence (BI)

V názvu práce jsou pojmy datový sklad a OLAP, kterým se budu věnovat v následujících kapitolách. Ještě předtím bych však chtěl představit pojem Business Intelligence, který tato témata zastřešuje a dle mého názoru vyjadřuje cíl, ke kterému směřuje vše, co zde bude dále uvedeno.

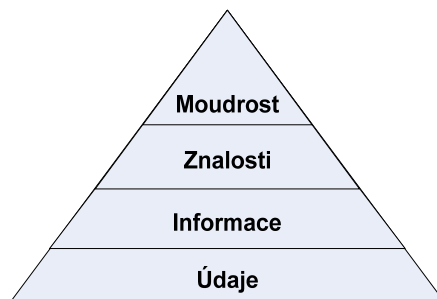
2.1 Definice

Business Intelligence lze definovat jako proces transformace dat na informace a převod těchto informací na poznatky prostřednictvím objevování. [1]

Dá se tedy říci, že účelem je konverze velkých objemů údajů, uložených v databázích na poznatky, které jsou potřebné pro koncové uživatele. Ti je potom mohou efektivně využívat například v procesu rozhodování.

2.2 Hierarchie informačních úrovní

Proces přeměny dat (uvedený v definici výše) dokumentuje hierarchie informačních úrovní na obr. 2.1 (převzat z [2]).



Obr. 2.1: Hierarchie informačních úrovní

Základem jsou **údaje**, které máme k dispozici (obvykle ve formě databáze). Ty však obsahují jenom jednoduchá fakta. Teprve jsou-li tato data pochopena a interpretována člověkem, respektive jiným inteligentním systémem, stávají se smysluplnými **informacemi**.

Další stupeň hierarchie tvoří **znalost**. Ta vypovídá o schopnosti uchovávat, komunikovat a zpracovávat informace do systematicky a hierarchicky uspořádaných znalostních struktur.

Moudrost jsou pak znalosti, spojené s určitým postojem a vyjadřuje komplexní hodnocení či poznání okolního světa. Moudrost vychází z minulosti i současnosti, ale zahrnuje také jistý pohled do budoucnosti.

Některé zdroje uvádí v této hierarchii ještě pátý stupeň a tím je **osvícení** (viz [3]). To pak představuje znalost kompletní pravdy. Člověk nejen že ví proč, ale ví to najisto (zde už jsme ale mimo oblast, pro kterou mohou poskytovat podporu počítačové systémy).

2.3 Vývoj systémů pro podporu řídicích činností

Informační systémy pro podporu řídicích činností se začaly objevovat už od 60. let, brzy po nástupu počítačů. Snažily se poskytovat vedoucím pracovníkům prostředky, které by jim efektivně a účinně pomáhaly v jejich práci a poskytovaly jim takovou informační podporu, jakou potřebují (přehled z [4]).

První takové systémy byly označovány jako **Management Information Systems - MIS** (systémy pro podporu řízení, resp. managementu). Jejich záběr byl ovšem značně široký a obvykle zahrnoval i běžné rutinní agendy, jejichž provozování zajišťovalo vlastní chod příslušné organizace. Potřebám strategického rozhodování a řízení vycházely takovéto systémy vstříc jen nepřímo, především poskytováním nejrůznějších přehledů (například o stavu zásob, o stavu financí na účtech apod.). Navíc v takové formě, která sice byla dobře "stravitelná" pro uživatele zvyklé na práci s počítači, ale která mohla být pro ostatní nepřekonatelnou překážkou.

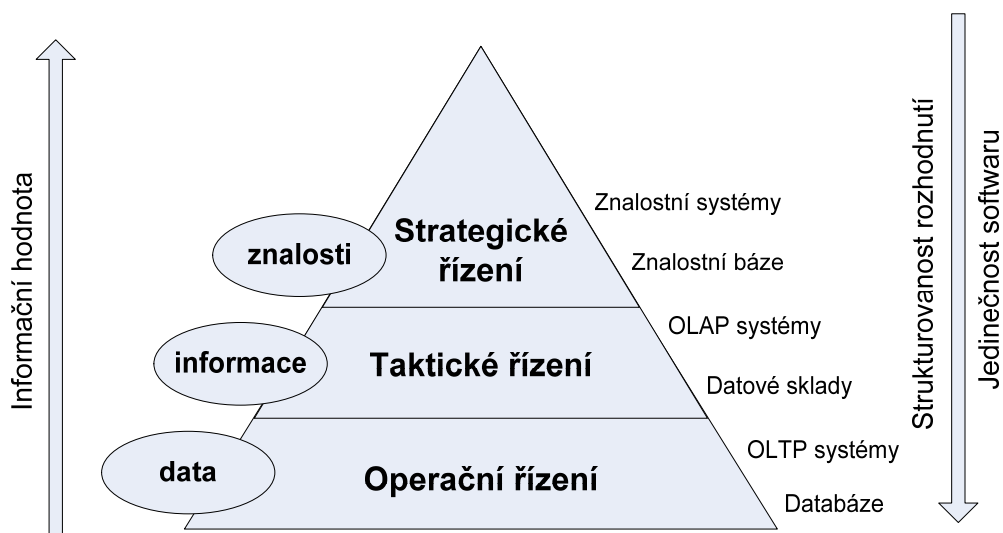
Proto se začaly objevovat snahy vyjít více vstříc také lidem, kteří se místo každodenního "provozního" řízení, obvykle spojeného i s rutinním používáním počítačů, zabývají spíše řízením strategickým a přijímáním koncepčních rozhodnutí. Světlo světa tak spatřily aplikace obvykle vytvářené a provozované v rámci systémů MIS a označované souhrnně jako **Decision Support Systems - DSS** (systémy pro podporu rozhodování). Jejich základním úkolem bylo poskytovat řídicím pracovníkům nejen informace, ale také vhodné nástroje, pomocí kterých by bylo možné vyhodnocovat různé alternativy, které připadají v úvahu, vypočítávat jejich ekonomický efekt apod. Tyto systémy se již poměrně široce prosadily. Stále však jejich využití zůstalo spíše u středních řídicích pracovníků. U vyššího managementu opět příliš neuspěly.

S rozvojem možností informačních technologií se začala objevovat třetí generace těchto prostředků. Jde o systémy označované jako **Executive Information Systems – EIS**, které jsou určeny pro vrcholové řízení. Tyto systémy poskytují přímý (on-line) přístup k aktuálním informacím o stavu řízené organizace, a to s maximální možnou mírou intuitivnosti. Jsou koncipovány tak, aby vycházely vstříc způsobu myšlení řídicích pracovníků a respektovaly to, co tyto pracovníky zajímá nejvíce.

2.4 Architektura systémů Business Intelligence

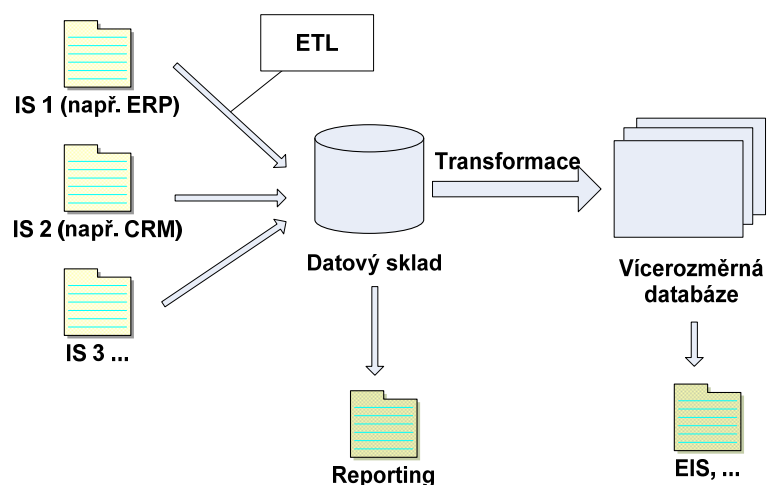
Systémy EIS jsou nyní stále častěji označovány právě termínem **Business Intelligence**, který je zřejmě marketingově výhodnější a pomáhá tak k prosazení těchto systémů i u nejvyššího managementu. Neznamená to však, že by byl určen pouze pro ně. Obecně je možné říci, že systémy vyšší úrovně v sobě zahrnují i systémy nižší úrovně.

Hlavním smyslem systému Business Intelligence je publikovat data v konsolidované formě, vhodné pro podporu rozhodování na různých úrovních v hierarchii podniku či organizace. Viz schéma na obr. 2.2 vycházející z [5] a [6].



Obr. 2.2: Tříúrovňové schéma řízení

Na obrázku 2.3 je pak zobrazeno základní schéma celého systému BI. Obvykle se vychází z centrálního podnikové IS (ERP), popřípadě systému CRM (systém pro podporu řízení vztahů se zákazníky) a dalších podnikových systémů a zdrojů. Odtud jsou data ve fázi ETL převáděna do datového skladu a dále transformována do multidimenzionálních databází. Z těchto zdrojů jsou vytvářeny výsledné reporty a publikována data v aplikacích na podporu rozhodování.



Obr. 2.3: Schéma systému Business Intelligence(převzato ze [7])

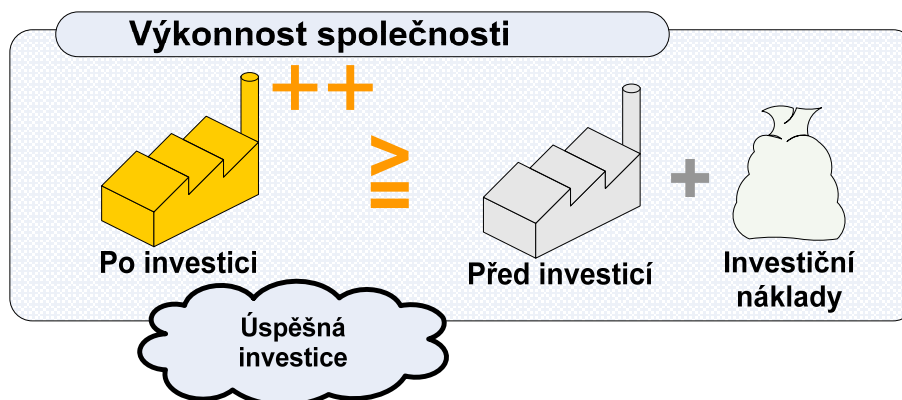
Za srdce celého systému můžeme označit datový sklad, který prezentuje jediný bod datové pravdy pro všechny uživatele. Mezi další důležité komponenty patří:

- aplikační server (OLAP server)
- reportingový server, uživatelský portál
- dataminingové nástroje
- uživatelské nástroje (klientské aplikace pro přístup k datům)
- vývojové nástroje (tvorba reportů, aplikací)

Podrobněji budou jednotlivé komponenty(především datový sklad a OLAP) popsány v dalších kapitolách.

2.5 Nasazení Business Intelligence

Stejně jako u jiných projektů je i při nasazování BI cílem úspěšná investice. Poměrně dlouhý a nákladný vývoj opravdu komplexního systému tak předchází rozhodování, zda bude projekt pro společnost opravdu přínosem.



Obr. 2.4: Úspěšná investice [7]

Zavádění BI systému proto nebývá realizováno jako jeden projekt (vzhledem ke komplexnosti by to nebylo ani možné), ale zahrnuje mnoho menších projektů - vedení společnosti pak může vidět a také využívat dílčí části z postupně budovaného systému.

Plánování a analýzu je samozřejmě nutné provádět na celopodnikové globální úrovni, avšak realizace už může být inkrementální, kdy typicky jeden projekt pokrývá jeden obchodní proces a vytváří například jeden datový trh. Jde tedy spíše o nikdy nekončící iterativní proces, kdy objem práce v jednotlivých iteracích často narůstá a vývojový tým zajišťuje jak rozvoj, tak i údržbu a provoz. [8]

Celkový úspěch systému Business Intelligence je dán mírou využívání a spokojenosti jeho uživatelů a přidanou hodnotou, kterou jim přináší v jejich každodenní obchodní praxi.

Úspěšný BI systém pak má následující vlastnosti [8]:

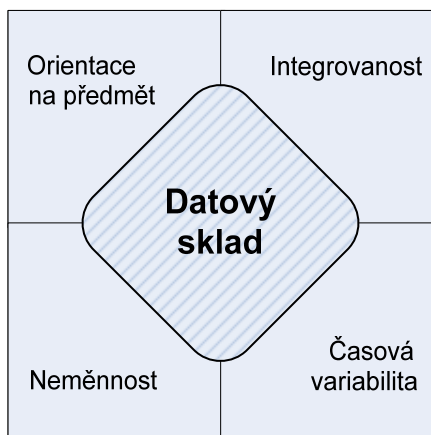
- Umožňuje snadný přístup k informacím organizace – rychlý přístup ke srozumitelným a intuitivním datům
- Prezentuje informace konzistentně a věrohodně
- Je adaptivní a schopný pružně a rychle reagovat na změny
- Je bezpečným místem, které chrání citlivá informační aktiva organizace
- Je přijat a používán uživateli a má jejich důvěru

3 Datový sklad

Po úvodní rámcové kapitole se nyní dostáváme ke kapitolám, které popisují základní komponenty figurující v procesu zpracování dat. Tato kapitola se bude věnovat datovým skladům, především jejich struktuře a výstavbě.

3.1 Definice

Nejznámější obecná definice datového skladu pochází od Billa Inmona. Ten říká, že datový sklad je podnikově strukturovaný depozitář subjektivě orientovaných, integrovaných, časově proměnných, historických dat, použitých na získávání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.



Obr. 3.1: Grafická podoba definice datového skladu

Definice datového skladu je graficky znázorněna na obrázku 3.1. Jednotlivé pojmy, které datový sklad odlišují od operační databáze jsou vysvětlovány následovně (zdroj [1], [9]):

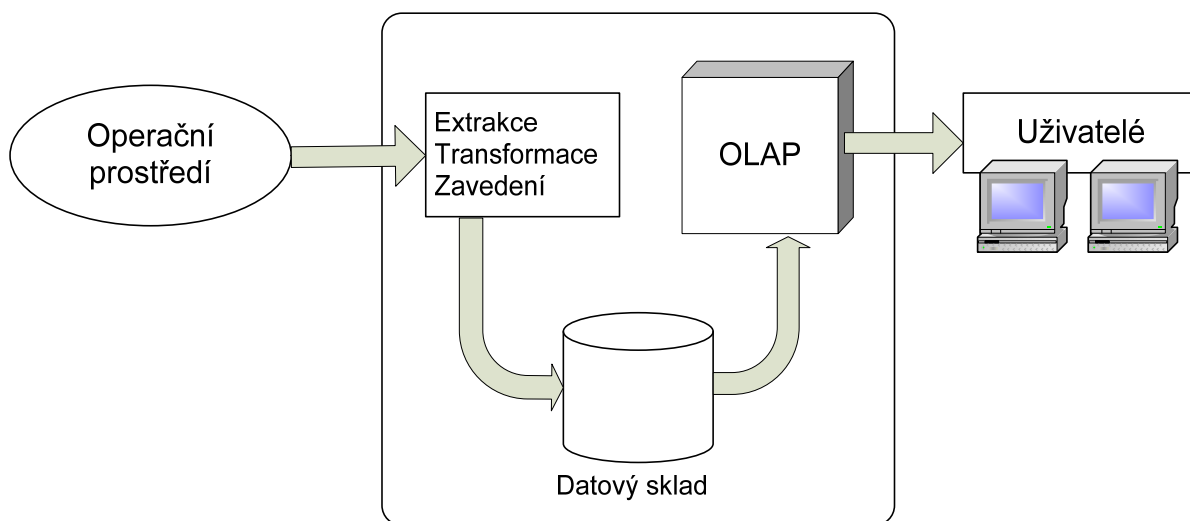
- **Orientace na předmět:** údaje se do databáze zapisují spíše podle předmětu zájmu, než podle aplikace, v níž byly vytvořeny. Z pohledu uživatelů je obsah datového skladu organizovaný podle předmětných oblastí jednotlivých odborných směrů v organizaci.
- **Integrovanost:** datová integrovanost spočívá v tom, že všechna data, která se týkají konkrétního předmětu, se do datového skladu ukládají jen jednou. Data jsou kategorizována a uspořádána v zájmu snazší správy, přehlednosti a dostupnosti. Je tedy nutné zavést jednotnou terminologii, jednotné a konzistentní jednotky veličin apod. To nebývá jednoduché, protože data většinou přicházejí do datového skladu z více (navzájem nekonzistentních) zdrojů. Proto musí být v etapě přípravy a zavedení upravena, vyčištěna a sjednocena.

- **Časová variabilita:** tento pojem znamená, že data jsou ukládána do datového skladu z hlediska času sekvenčně, reprezentují tak určitý časový úsek. Přidávají se nová aktuální data.
- **Neměnnost:** neměnnost vyjadřuje, že starší, neaktuální data se ze skladu neodstraňují ani nemění, ale jsou v něm spravována dále; je tak možné sledovat vývoj datové základny.

Takto definovaný datový sklad pak může splňovat základní požadavky, které jsou na něj kladeny. Stává se zdrojem pro reporty a další formy prezentace dat a tvoří také základnu pro přechod od transakčních systémů OLTP k analytickým systémům OLAP (podrobněji popsány v další kapitole).

3.2 Procesní schéma

Data se v běžných systémech získávají a ukládají do produkčních (operačních) databází, které mohou využívat různá oddělení firmy a jsou tedy často umístěna v různých geografických lokalitách. Tyto data jsou pak v pravidelných intervalech sbírána, předzpracována a zaváděna do datového skladu. Ten pak může být zdrojem pro vykonávání různých analýz. Přístup k těmto datům a analýzám mohou mít manažeři pomocí webu z celého světa. S různými přístupovými právy pak mohou mít přístup také obchodníci, obchodní partneři nebo třeba zákazníci.



Obr. 3.2: Procesní schéma [1]

3.3 Datové trhy

Datové trhy jsou menší podmnožiny datového skladu, určené pro jednotlivá oddělení firmy. Datový trh může existovat také samostatně jako jednoduchý datový sklad, většinou však funguje jako subsystém většího celku.

Důvodem jejich vzniku je jednak logické členění datových skladů, ale také systém budování datového skladu po částech, kvůli náročnosti celkového projektu.

3.4 Budování datového skladu

Vývoj datových skladů začal v době hromadného nasazování provozních informačních systémů, které produkovaly velké množství dat, z nichž bylo potřeba získávat analytické a statistické údaje, a to pokud možno snadno a rychle. To bylo a zůstává základní ideou, které se musíme držet při návrhu a budování datového skladu

Mezi nejčastěji používané metody budování datového skladu patří metoda „velkého třesku“ a přírůstková metoda[1]:

1) Metoda „velkého třesku“

Tato metoda se skládá ze tří etap:

- Analýza požadavků podniku
- Vytvoření podnikového datového skladu
- Vytvoření přístupu buď přímo nebo pomocí datových trhů

Výhodou této metody je skutečnost, že můžeme celý projektový plán kompletně vypracovat ještě před začátkem jeho realizace. To však může být také nevýhoda, protože požadavky uživatelů ale i technologie se mohou v průběhu změnit. Další poměrně významnou slabinou je dlouhá doba, než začne takto budovaný sklad fungovat a začne přinášet užitek.

2) Přírůstková metoda

Přírůstková metoda předpokládá budování datového skladu po jednotlivých etapách. Místo vybudování celého skladu naráz postupně přibývají přírůstková řešení, která samozřejmě zapadají do celkové architektury.

Výhodou je oproti předchozí metodě mnohem dřívější uvedení do provozu (alespoň části systému, například v podobě datového trhu) a také zpětná vazba od uživatelů.

Tato metoda se používá mnohem častěji, popíše tedy podrobněji její fáze:

Strategie: Úloha této fáze je definovat obchodní cíle a také účel datového skladu. Dále je zde vytvořena strategie správy skladu, včetně dokumentace a školení. Měl by být také definován základ architektury datového skladu.

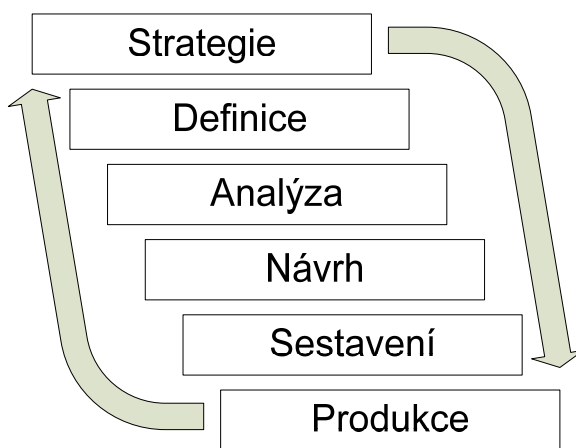
Definice: V této fázi je určen rozsah a cíl přírůstkového vývoje. Je nutné pochopit a zdokumentovat strukturu zdrojů údajů. Vytvoří se počáteční přírůstek a konceptuální modely. Především je zde však navržena architektura datového skladu i technických prostředků.

Analýza: Cílem analýzy je zaměřit se na informace ohledně uživatelů, získávání dat a požadavků na přístup k datům. Dokončí se výběr nástroje pro všechny komponenty datového skladu. Řeší se také problém kvality dat a požadavky na metadata.

Návrh: Při návrhu se transformují požadavky získané při analýze do detailních podmínek návrhu. Vytvářejí se relační a případně multidimenzionální modely. Je dokončena instalace technické architektury.

Sestavení: Cílem fáze sestavení je vytvořit a otestovat vytvořené databázové struktury a veškeré moduly datového skladu.

Produkce: V této fázi je datový sklad nainstalován, začne se používat a je nutné řídit jeho údržbu a růst.

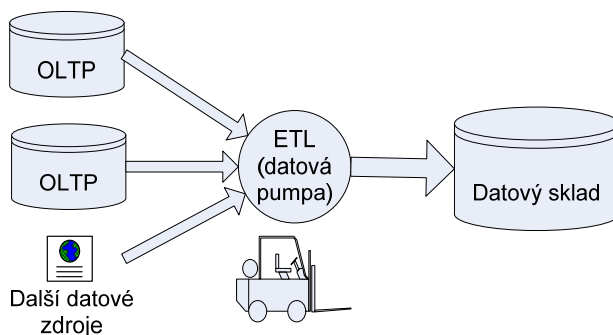


Obr 3.3: Schéma jedné iterace při použití přírůstkové metody, vychází z [1]

Shrnutí

Dobře navržený a úplně implementovaný datový sklad pravidelně přebírá data z jednoho nebo více provozních informačních systémů, příp. jiných zdrojů a ukládá je do své vlastní primární databáze. Do datového skladu se většinou nepřebírají všechna data provozního informačního systému, ale pouze určité podoblasti, které mají být předmětem dalšího zkoumání. V primární databázi datového skladu jsou data stále ještě uložena relačním způsobem a jde vlastně o jakýsi obraz vybrané části provozního systému s tím rozdílem, že se zde uchovávají data včetně historie. Proces, který zajišťuje

převod dat z provozního informačního systému do databáze datového skladu, se nazývá etapa ETL nebo také datová pumpa.



Obr. 3.4: Aktualizace datového skladu

3.5 Přístupy k řešení datové architektury

V současné době existují dva hlavní přístupy k řešení datové architektury datových skladů. Prvním z nich je Corporate Information Factory (CIF) od Billa Inmona a druhým přístupem je tzv. Data Warehouse Bus (DWHBus) od Ralpa Kimballa (při popisu vycházím z [8]).

Obě koncepce jsou poměrně odlišné, mají však některé společné znaky. Mezi ty patří především nutnost počáteční zevrubné analýzy uživatelských požadavků všech potenciálních uživatelů a analýza všech zdrojových systémů. Dále také metoda vytváření shora dolů a inkrementálnost vývoje celého systému. Pro oba přístupy je také společné, že prezentační vrstva obsahuje atomická data, aby mohla uspokojit všechny datové požadavky a samotná forma i obsah prezentační vrstvy vznikají s ohledem na celopodnikové zájmy (nikoli tedy podle jednotlivých oddělení). V obou případech také datové trhy používají dimenzionální model.

První odlišnost obou přístupů spočívá v pohledu na logický model prezentační vrstvy. Inmon preferuje entitně relační návrh (ER model, tabulky v 3NF). Naopak Kimball dává přednost procesnímu návrhu (dimenzionální model, faktové tabulky + dimenzionální tabulky ve 2NF nebo 3NF). Další rozdílnost existuje v interpretaci pojmu datový trh. Podle Inmona je to problémově orientovaná podmnožina dat prezentační vrstvy, která je ve vlastnictví oddělení organizace. Je plněn transformací dat prezentační vrstvy a je uložen v dedikovaném úložišti (data mohou být smazána). Kimballova vize představuje datový trh jako procesně orientovanou podmnožinu tabulek prezentační vrstvy. Prezentační vrstva je pak tvořena sjednocením těchto procesně orientovaných datových trhů.

Souhrnně řečeno je architektura podle Billa Inmona založena na centrálním datovém skladu a závislých datových trzích, jejichž zdrojem dat je právě celopodnikový datový sklad. Naopak idea Ralpa Kimballa je, že datový sklad může být vybudován inkrementálně pomocí série jednotlivých datových trhů s odpovídajícími dimenzemi.

3.6 Plnění skladu – etapa ETL

Úkolem etapy ETL je nejen vybrat specifikovanou část dat z provozního systému a tuto část překopírovat do databáze datového skladu. Proces převodu většinou představuje částečnou nebo i značnou změnu struktury ukládaných dat a hlavně jejich "čištění". V provozních systémech (zvláště pokud nejsou zcela správně navrženy) mohou být nekonzistentní data.

Totéž platí, když se provádí import z různých zdrojů dat. Proces čištění má za úkol zjistit a odstranit nekonzistence ve vstupních datech a může sloužit i jako opravná zpětná vazba pro provozní informační systém [10].

Z uvedených informací je možné odvodit tři základní fáze etapy ETL, které zároveň tvoří tuto zkratku:

Extrakce – výběr dat

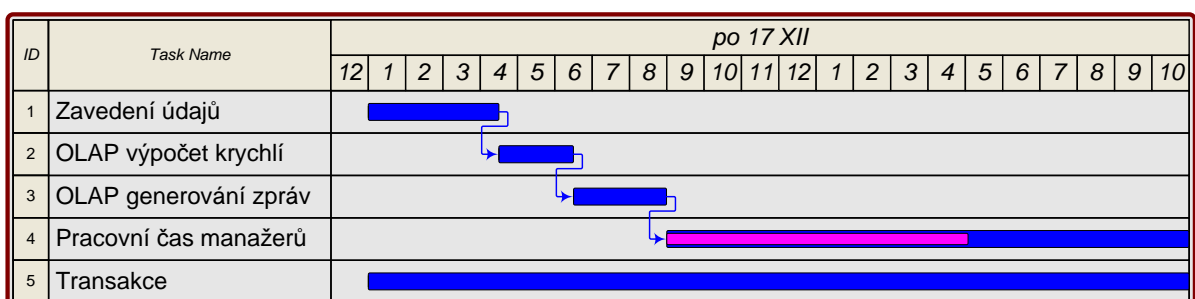
Transformace – ověření, čištění, integrování a časové označení dat

Loading – přemístění dat do datového skladu

Celý proces ETL je komplexní a časově poměrně náročný. Jeho hlavním cílem je centralizace údajů.

3.7 Provoz

Pokud je datový sklad vybudován a naplněn daty, čeká nás běžný provoz systému, který musí zajišťovat dostupnost dat a jejich pravidelnou aktualizaci. Ta může probíhat v denních, týdenních nebo třeba měsíčních cyklech.



Obr. 3.5: Časová návaznost procesu aktualizace datového skladu a procesu OLAP v rámci jednoho dne, [2]

Na obrázku 3.5 je zobrazen příklad časového schématu, podle kterého by mohlo probíhat denně aktualizované generování analýz a reportů. V nočních hodinách probíhá etapa ETL, výpočet OLAP krychlí a následné generování zpráv. Vše je tak připraveno v čase, kdy začíná pracovní doba manažerů. Souběžně s tím mohou stále probíhat transakce systému OLTP, ze kterého se data čerpají.

4 OLAP

Tématem této kapitoly je technologie OLAP, jejímž cílem je analýza dat a podpora rozhodování. Je založena na využití datových skladů, jejichž data prezentuje vhodným způsobem uživatelům.

4.1 Definice

OLAP (Online Analytical Processing) je technologie uložení dat v databázi, která umožňuje uspořádat velké objemy dat tak, aby byla data přístupná a srozumitelná uživatelům zabývajícím se analýzou obchodních trendů a výsledků. Způsob uložení dat se svým zaměřením liší od běžněji užívaného OLTP (Online Transaction Processing), kde je důraz kladen především na snadné a bezpečné ukládání změn v datech v konkurenčním (mnohouživatelském) prostředí. [11]

4.2 Rozdíl mezi OLTP a OLAP

Rozdílnost mezi OLAP a OLTP spočívá v tom, že OLTP systémy uchovávají záznamy o jednotlivých uskutečněných (typicky obchodních) transakcích a jsou obvykle realizovány pomocí dnes nejběžnější – relační – databázové technologie. Data uchovávaná v OLTP databázovém systému jsou (zpravidla periodicky) agregována (typicky sumarizována) a poté ukládána do datového skladu, nad nímž se posléze podle potřeb provádí okamžité zpracování analýz pomocí vrstvy OLAP.

Datový sklad je na rozdíl od OLTP databáze určen výhradně ke čtení dat pro potřeby nejrůznějších analýz. Jedinou výjimkou jsou (obvykle periodické) aktualizace. Tento zásadní rozdíl mezi OLTP systémy a datovými sklady má rozsáhlé důsledky pro způsob jeho implementace, návrhu a tvorby konceptuálního modelu, který je orientován na dosažení co nejrychlejšího zpracování dotazů, kladených datovému skladu vrstvou OLAP.

Základní rozdíly jsou shrnuty v tabulce (vychází z [12], [13]):

Znak	OLTP	OLAP
Charakteristika	Provozní zpracování	Informační zpracování
Uživatelé	Úředník, databázový administrátor	Znalostní pracovník (manažer, analytik)
Funkce	Každodenní operace	Podpora pro rozhodování
Návrh DB	Orientace na aplikaci	Orientace na subjekt
Data	Současná, aktuální	Historická, sumarizovaná, multidimenzionální, integrovaná
Přístup	Čtení, zápis, modifikace	Převážně čtení, periodický zápis

4.3 Příprava dat pro OLAP analýzy

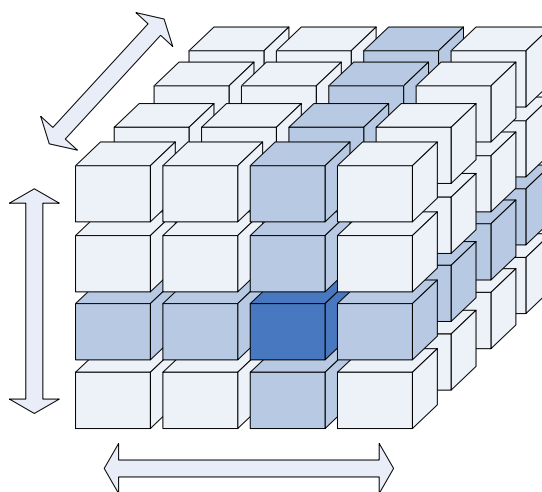
Analytické a statistické dotazy nad daty se vyznačují tím, že v naprosté většině případů směřují k získávání nikoliv detailních dat, nýbrž dat různým způsobem agregovaných. Počítání agregovaných dat je činnost velice časově náročná, proto je v datových skladech žádoucí tuto činnost zbytečně neopakovat. Proto je snaha vypočítat co možno nejvíce agregací dopředu a v hlavní části datového skladu uchovávat data už v agregované podobě. Nad takto uloženými daty je pak mnohem snazší realizovat dotazy, které jsou pro tento typ zpracování dat charakteristické.

4.4 Multidimenzionální databáze a databázový model

Do multidimenzionálních databází ukládáme data již upravená a vyčištěná a to většinou ve formě nenormalizovaných tabulek. Takto vytvořené databáze pak slouží jako podklad pro získání sumarizovaných a agregovaných dat. Pro využití OLAP je dále nutné organizovat data do multidimenzionálních struktur.

Ekvivalentem tabulky v relační databázi je v multidimenzionálním datovém modelu kostka. Datová kostka umožňuje modelování a pohled na data v různých dimenzích, jejichž počet může být, na rozdíl od geometrické kostky, mnohem vyšší. Díky tomu umožňuje OLAP kostka uchovávat velké množství informací. Její naplnění pak ale vyžaduje množství výpočtů a agregací.

I přes svůj velký objem dokáže OLAP kostka poskytovat rychlý přístup k datům. Ty se totiž nachází na průnicích jednotlivých dimenzí, jejichž rozsah již není tak velký.



Obr. 4.1: 3-dimenzionální kostka s vyznačeným průnikem vybraných dimenzí

Mezi výhody tohoto způsobu ukládání dat patří rychlý komplexní přístup k velkému objemu dat, možnost komplexních analýz a silné schopnosti pro modelování a prognózy. Nevýhodou je větší prostorová náročnost a problémy vznikající při změnách dimenzí.

4.5 Fakta, dimenze a schémata tabulek dimenzí

Pro vytvoření OLAP kostky potřebujeme dva druhy dat – fakta a dimenze. Fakta jsou numerické měrné jednotky obchodování (typicky cena, množství, ...). Dimenze obsahují logicky nebo organizačně hierarchicky uspořádaná data. Jsou to vlastně textové popisy obchodování (čas, region, produkt, ...).

Při konceptuálním modelování datových skladů se využívají převážně tyto tři schémata[12]:

- **Schéma hvězdy:** Jedna tabulka faktů je spojená s množinou tabulek dimenzí.
- **Schéma sněhové vločky:** Modifikace schématu hvězdy, kde je provedena normalizace tabulek dimenzí.
- **Schéma souhvězdí:** Jde o kolekci hvězd. Je zde více tabulek faktů, které sdílí tabulky dimenzí.

4.6 Úložiště multidimenzionálních dat

Pro uložení multidimenzionálních dat existuje několik typů architektur. Mezi nejpoužívanější patří relační, multidimenzionální a hybridní OLAP servery.

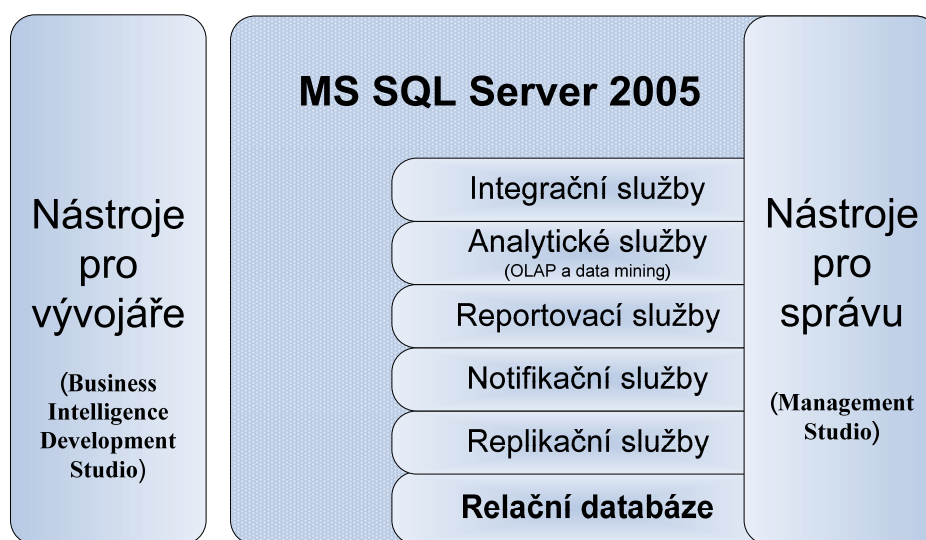
Relační OLAP (ROLAP) využívá pro analýzy data z datového skladu, kde jsou uložena v relačních databázích. Ty musí být zpracovány a pak mohou být předloženy uživateli jako multidimenzionální pohled. Zahrnuje také optimalizaci SŘBD, implementaci agregačních funkcí a další přídatné nástroje a služby.

Multidimenzionální OLAP (MOLAP) získává data z datových skladů nebo i operačních zdrojů. Po zpracování je však ukládá ve vlastních datových strukturách a sumářích. Dochází zde k vypočítání mnoha předběžných výsledků, takže následné získávání informací je velmi rychlé. Problémem však je velké množství ukládaných dat a také redundance.

Hybridní OLAP (HOLAP) představuje kompromis mezi ROLAP a MOLAP, využívající výhody obou. Relační databáze ukládá množství detailních dat a multidimenzionální model ukládá sumární data.

5 MS SQL Server

MS SQL Server je robustní databázový systém, poskytující silnou podporu pro implementaci datových skladů, OLAP analýz a také vytváření reportů a prezentaci dat. V této kapitole se budu věnovat popisu MS SQL Serveru ve verzi 2005.



Obr. 5.1: Platforma MS SQL Serveru 2005

Datová platforma MS SQL Server 2005 zahrnuje následující nástroje(zdroj: [14] a [15]):

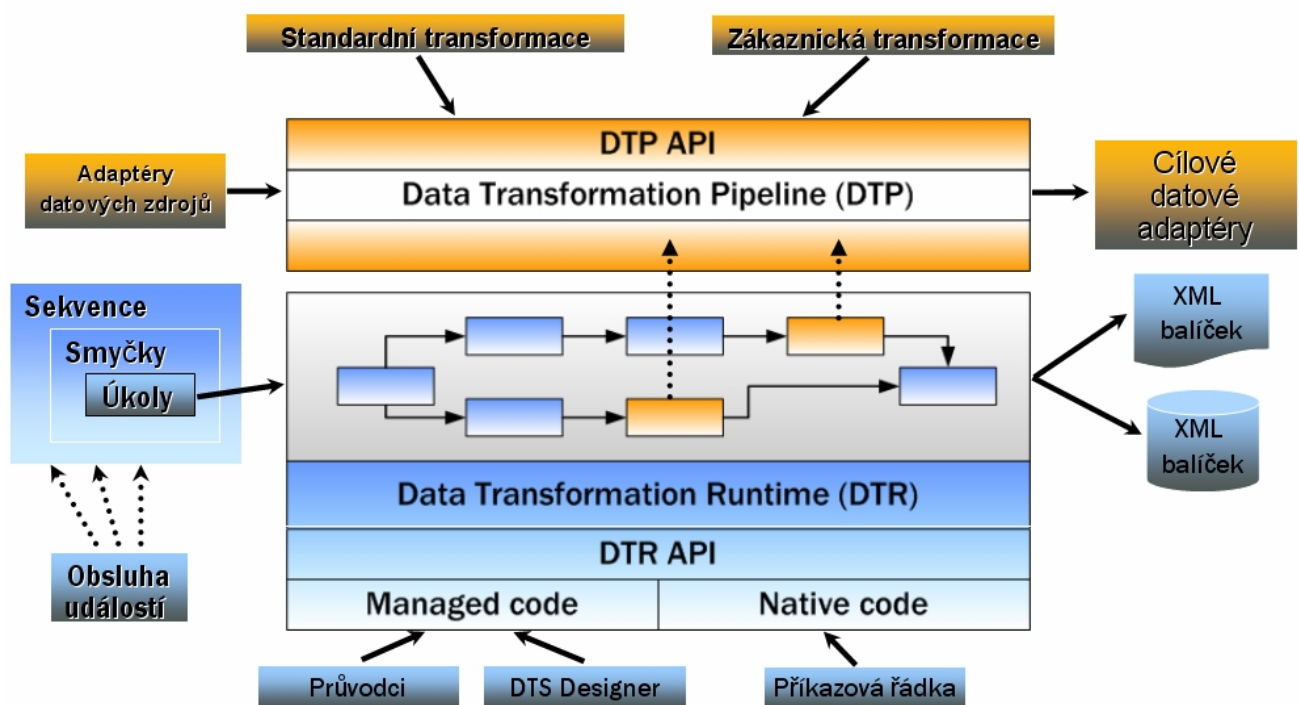
- **Relační databáze:** Relační databázový stroj.
- **Služba Replication Services:** Replikace dat pro aplikace zpracovávající distribuovaná či mobilní data.
- **Služba Notification Services:** Funkce zasílání upozornění pro vývoj a nasazení škálovatelných aplikací. Umožňuje zasílat aktuální informace přizpůsobené individuálním požadavkům na nejrůznější připojená i mobilní zařízení.
- **Služba Integration Services:** Funkce extrakce, transformace a načítání dat (ETL) pro datové sklady a integraci dat v celém podniku.
- **Služba Analysis Services:** Funkce OLAP pro analýzu velkých a složitých datových sad s využitím vícedimenzionálních úložišť.
- **Služba Reporting Services:** Komplexní řešení pro vytváření, správu a zasílání statických i interaktivních webových sestav.
- **Nástroje pro správu (MS SQL Server Management Studio):** SQL Server zahrnuje integrované nástroje pro pokročilou správu a ladění databází.

- **Nástroje pro vývojáře (Business Intelligence Development Studio):** SQL Server nabízí integrované nástroje pro vývojáře, určené pro databázový stroj, extrakci, transformaci a načítání dat (ETL), dolování dat, funkce OLAP a vytváření sestav, které jsou úzce integrovány se sadou Microsoft Visual Studio a poskytují komplexní funkce pro vývoj aplikací.

V následujících podkapitolách se budu věnovat pouze integračním, analytickým a reportovacím službám, které se nejvíce váží k problematice datových skladů a OLAP. Pro implementaci je samozřejmě důležité znát i ostatní části tohoto systému, ty však budou popsány až v rámci řešení praktické části práce.

5.1 Integrační služby (SSIS)

Integrační služby (SSIS – SQL Server Integration Services) zajišťují především zavádění dat do datových skladů (fázi ETL). Využití však naleznou i při instalaci databázové části IS, migraci na jiný informační systém nebo při výměně databázového serveru.



Obr. 5.2: Architektura integračních služeb MS SQL Serveru 2005

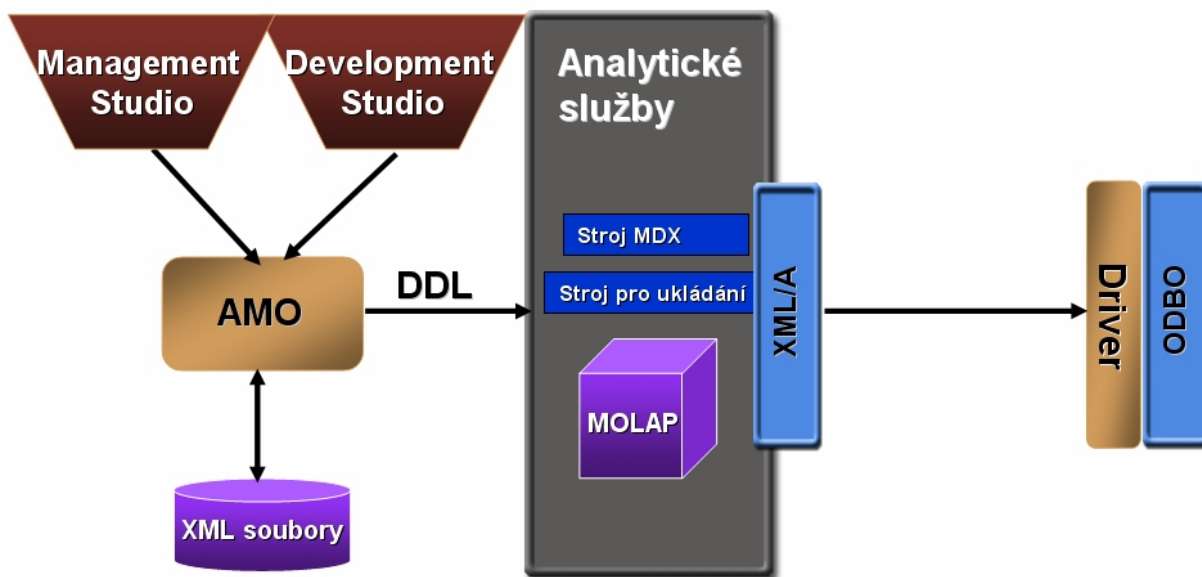
Základem architektury integračních služeb je DTR (Data Transformation Runtime), který provádí vykonávání jednotlivých kroků v rámci workflow. Tento runtime je přístupný přes API funkce pomocí různých průvodců, DTS Designeru ve Visual Studiu nebo také příkazové řádky. Tento

runtime vykonává různé sekvence, smyčky úloh, provádí odchyťování a zpracování událostí a chyb. Dále také provádí logování a monitorování provádění daného balíčku a další služby.

V případě, že se ve workflow nachází dataflow úloha, je vyvolána další část integračních služeb a tou je DTP (Data Transformation Pipeline). Tato část zajišťuje práci uskutečňovanou mezi zdrojovými a cílovými daty. Zajišťuje transformaci, čištění a další zpracovávání dat (tyto úkoly je možné vytvořit v rámci integračních služeb nebo pomocí vlastních aplikací).

5.2 Analytické služby (SSAS)

Analytické služby SQL Serveru (SSAS - SQL Server Analysis Services) poskytují správu databází, analytických služeb a analytických objektů v rámci nástroje MS SQL Server Management Studio a vytváření objektů BI v nástroji MS SQL Server BI Development (součást MS Visual Studio 2005).

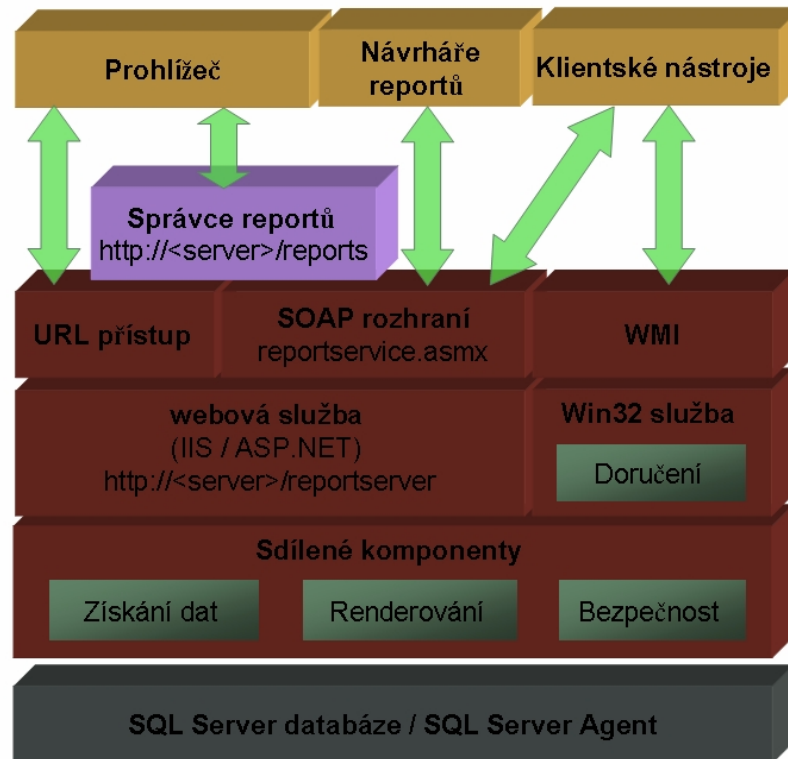


Obr. 5.3: Architektura analytických služeb MS SQL Serveru 2005

Při vývoji analytických služeb nezapisujeme do relační databáze, ale místo toho vytváříme XML soubory s definicí jednotlivých objektů na disku. K těmto souborům můžeme přistupovat prostřednictvím rozhraní AMO (Analysis Management Objects) a k dispozici máme nástroj pro správu analytických služeb - Management Studio a nástroj pro jejich vývoj - Business Intelligence Development Studio, který je součástí Visual Studia. Objektová knihovna AMO potom posílá příkazy na server analytických služeb, kde probíhá zpracování a MDX kalkulace (multidimenzionální výpočty). Rozhraním analytických služeb je XML/A. Klient pak může přistupovat nativně přes toto rozhraní, nebo může použít ovladač, který překládá volání staršího rozhraní ODBO na volání XML/A standardu.

5.3 Reportovací služby (SSRS)

Reportovací služby (SSRS – SQL Server Reporting Services) slouží pro podporu rozhodování na všech stupních organizační struktury. Hlavním důvodem pro jejich nasazení je generování výstupů v elektronické nebo papírové podobě, umožňující efektivní přístup k datům/informacím.



Obr. 5.4: Architektura reportovacích služeb MS SQL Serveru 2005

Architektura reportovacích služeb má blokovou stavbu. Základ tvoří sdílené komponenty, sloužící pro získávání dat, zobrazování sestav a nastavení bezpečnosti.

Další část infrastruktury je orientovaná na klientské aplikace a na zobrazování na klientském počítači. K tomu je určena webová služba, která je schopná doručit sestavu klientské aplikaci. Poskytuje URL přístup, takže umožňuje zobrazování sestav přímo v prohlížeči. Přes SOAP rozhraní je navíc možné použít připravenou webovou službu tak, že je možné zpracovat sestavy přímo do klientských aplikací. I samotné návrhář reportů komunikují s reportovacími službami přes toto SOAP rozhraní.

Doručovací kanály jsou obhospodařované win32 službou. Pro monitorování reportovacích služeb je možné využít infrastrukturu WMI.

Všechny informace, nastavení a předpřipravené reporty jsou ukládané do databáze SQL Serveru, která bude při plánovaném zpracování obsluhována SQL Server agentem.

Správa reportovacích služeb a samotných reportů je zpřístupňována přes url <http://<server>/reports>.

6 Ukázková aplikace

V této kapitole je popsána ukázková aplikace, která demonstruje práci s datovým skladem v prostředí MS SQL Serveru (konkrétně ve verzi 2005 se Service Packem 2).

Jako základ je vzata OLTP databáze „Adventure Works“ dodávaná spolu s MS SQL Serverem 2005. Obsahuje celkem 70 tabulek a svým rozsahem i obsahem odpovídá běžné menší firmě.

Budování celého datového skladu je úkol velmi komplexní a časově náročný i pro takto rozsáhlou databázi, proto jsem se v rámci diplomového projektu zaměřil na uskutečnění první iterace vývoje skladu, jejímž cílem bylo vytvoření datového trhu pro obchodní oddělení této fiktivní firmy.

Součástí této iterace je také vytvoření OLAP analýz a reportů nad tímto datovým trhem a ukázka jejich prezentace v klientské aplikaci a přes webové rozhraní.

6.1 Vybudování datového skladu pomocí SSIS

Při budování datového skladu jsem použil přírůstkovou metodu, tak jak je popsána v kapitole 3.4. a níže popisují průběh jednotlivých fází. Celkově jsem při řešení datové architektury vycházel z koncepce Data Warehouse Bus od Ralpa Kimballa (viz kapitola 3.5, nebo podrobněji v [16]-[19]).

6.1.1 Strategie

Cílem vytvářeného datového skladu je umožnit vedoucím pracovníkům, manažerům a analytikům ve firmě dělat kvalitnější a rychlejší rozhodnutí. Vyžaduje se především poskytování reportů (provozních i strategických), ad-hoc analýzy a provozní podpora (scoring, podklady pro direct-marketingové kampaně, atd.). Z dlouhodobějšího hlediska je požadováno sledování plnění plánů a trendů vývoje.

Projekt vybudování datového skladu má být uskutečněn v časovém horizontu jednoho roku s použitím přírůstkové metody. V souladu s koncepcí Data Warehouse Bus je obsahem každé iterace vybudování jednoho datového trhu pro jednotlivá oddělení firmy. První iterace, kterou zde hodlám popsat, se zabývá datovým trhem pro obchodní oddělení. Součástí tohoto dokumentu budou také informace pro správu datového skladu, jeho dokumentace a podklady pro školení uživatelů.

6.1.2 Definice

První iterace vývoje má za cíl vytvořit datový trh, jež by svým rozsahem pokrýval potřeby obchodního oddělení. Základním zdrojem dat je provozní OLTP databáze firmy. Na základě důkladného prostudování schématu[20] a popisu[21] této databáze byl vytvořen výčet tabulek, z nichž

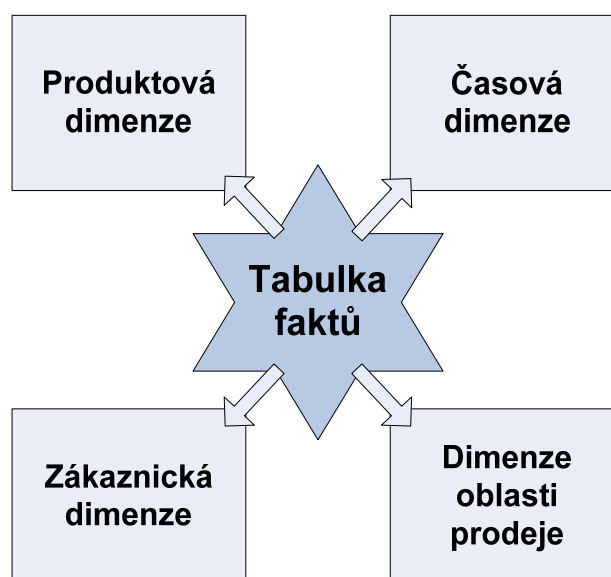
se budou čerpat data pro vybudování datového skladu. Schéma těchto vybraných tabulek je uvedeno jako příloha 1 na konci této práce.

Vzhledem k tomu, že jsou data brána z jednoho zdroje, nepředpokládají se problémy s integritou dat. Návrh OLTP databáze je poměrně kvalitní, takže i v oblasti čištění dat se neočekávají výrazné problémy.

6.1.3 Analýza

Mezi uživatele vytvářeného datového trhu budou patřit všichni pracovníci obchodního oddělení a jim nadřízení pracovníci. Řešení bude postaveno na produktu SQL Server 2005, instalovaném na počítači/serveru, který je určen výhradně k provozu datového skladu.

Na základě zdrojových dat a požadavků na výkon bylo pro konceptuální model skladu vybráno schéma hvězdy, ve kterém bude centrální tabulka faktů obsahovat počet a cenu prodaných produktů. K tabulce faktů se budou vztahovat čtyři dimenze – produktová, časová, zákaznická a oblastní (viz obr. 6.1).



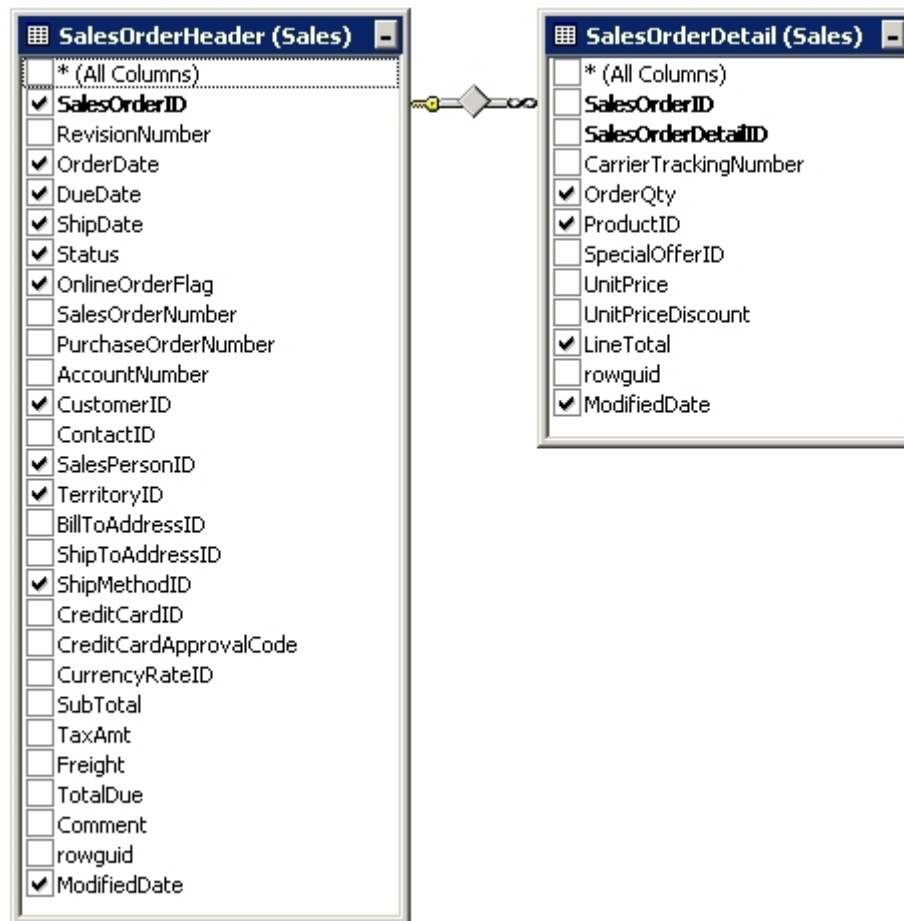
Obr. 6.1: Základní schéma datového skladu

6.1.4 Návrh

Tato fáze transformuje výsledky analýzy do detailních podmínek návrhu. V tomto procesu jsou tabulky ze zdrojové OLTP databáze poměrně silně denormalizovány - kvůli zvýšení výkonu a také přehlednosti při přístupu k nim.

Níže je popsán vytvořený relační model datového skladu (na jehož základě bude v kapitole 6.2 vytvořen model multidimenzionální). Výraznou pomocí při tvorbě jednotlivých tabulek byl průvodce a designer pohledů v produktu MS SQL Server Management Studio, z něhož také pochází přiložené obrázky.

Tabulka faktů

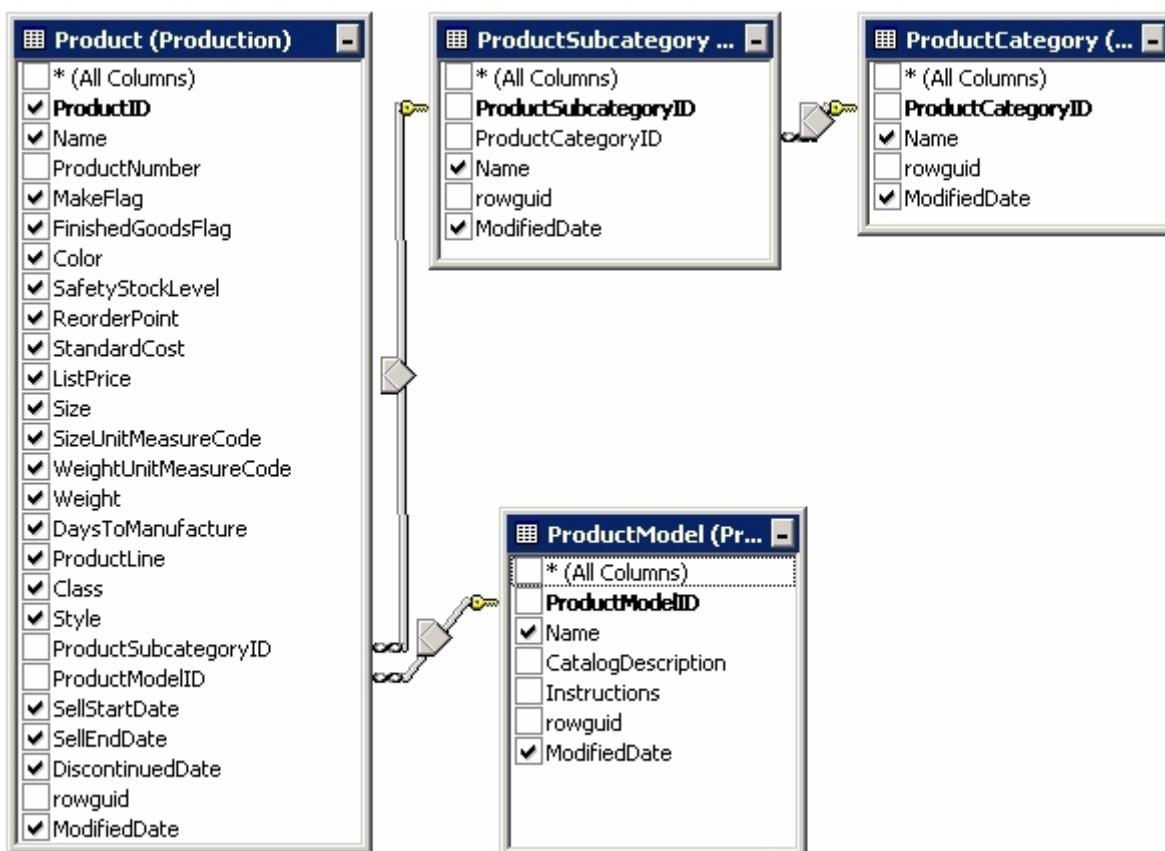


Obr. 6.2: Výběr dat pro přenos do datového skladu

Tabulka faktů factOrders byla vytvořena na základě spojení tabulek SalesOrderHeader a SalesOrderDetail (viz obr. 6.2, zaškrtnuty jsou sloupce přenášené do datového skladu) pomocí následujícího SQL příkazu:

```
SELECT TOP (100) PERCENT Sales.SalesOrderHeader.SalesOrderID,  
Sales.SalesOrderHeader.OrderDate, Sales.SalesOrderHeader.DueDate,  
Sales.SalesOrderHeader.ShipDate, Sales.SalesOrderHeader.Status,  
Sales.SalesOrderHeader.OnlineOrderFlag, Sales.SalesOrderHeader.CustomerID,  
Sales.SalesOrderHeader.SalesPersonID, Sales.SalesOrderHeader.TerritoryID,  
Sales.SalesOrderHeader.ShipMethodID,  
Sales.SalesOrderHeader.ModifiedDate, Sales.SalesOrderDetail.OrderQty,  
Sales.SalesOrderDetail.ProductID, Sales.SalesOrderDetail.LineTotal,  
Sales.SalesOrderDetail.ModifiedDate AS DetModifiedDate  
FROM Sales.SalesOrderHeader INNER JOIN Sales.SalesOrderDetail ON  
Sales.SalesOrderHeader.SalesOrderID = Sales.SalesOrderDetail.SalesOrderID
```

Tabulka produktů



Obr. 6.3: Výběr dat pro přenos do datového skladu

Tabulka produktové dimenze dimProduct byla vytvořena na základě spojení tabulek Product, ProductSubcategory, ProductCategory a ProductModel (viz obr. 6.3) pomocí následujícího SQL příkazu:

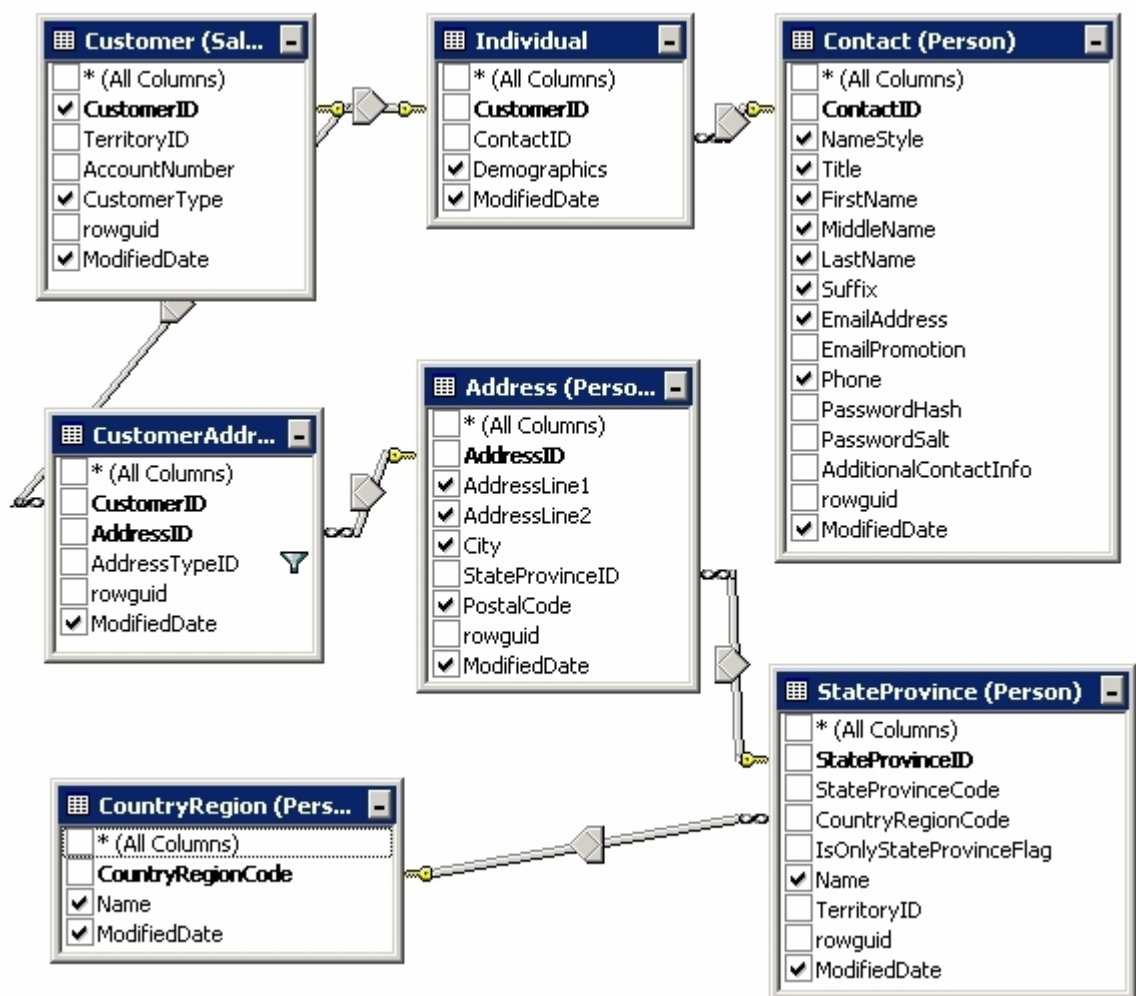
```
SELECT Production.Product.ProductID, Production.Product.Name,  
Production.ProductModel.Name AS ProductModelName, Production.Product.ProductLine,  
Production.ProductSubcategory.Name AS Subcategory, Production.ProductCategory.Name  
AS Category, Production.Product.MakeFlag, Production.Product.FinishedGoodsFlag,  
Production.Product.Color, Production.Product.SafetyStockLevel,  
Production.Product.ReorderPoint, Production.Product.StandardCost,  
Production.Product.ListPrice, Production.Product.Size,  
Production.Product.SizeUnitMeasureCode, Production.Product.WeightUnitMeasureCode,  
Production.Product.Weight, Production.Product.DaysToManufacture,  
Production.Product.Class, Production.Product.Style, Production.Product.SellStartDate,  
Production.Product.SellEndDate, Production.Product.DiscontinuedDate,  
Production.Product.ModifiedDate, Production.ProductSubcategory.ModifiedDate AS  
MD_Subcategory, Production.ProductCategory.ModifiedDate AS MD_Category,
```

```

Production.ProductModel.ModifiedDate AS MD_ProductModel
FROM      Production.Product LEFT OUTER JOIN Production.ProductSubcategory ON
Production.Product.ProductSubcategoryID =
Production.ProductSubcategory.ProductSubcategoryID LEFT OUTER JOIN
Production.ProductCategory ON Production.ProductSubcategory.ProductCategoryID =
Production.ProductCategory.ProductCategoryID LEFT OUTER JOIN
Production.ProductModel ON Production.Product.ProductModelID =
Production.ProductModel.ProductModelID

```

Tabulka zákazníků



Obr. 6.4: Výběr dat pro přenos do datového skladu

Tabulka zákaznické dimenze dimCustomer byla vytvořena na základě spojení tabulek Customer, Individual, Contact, CustomerAddress, Address, StateProvince a CountryRegion (viz obr. 6.4) pomocí následujícího SQL příkazu:

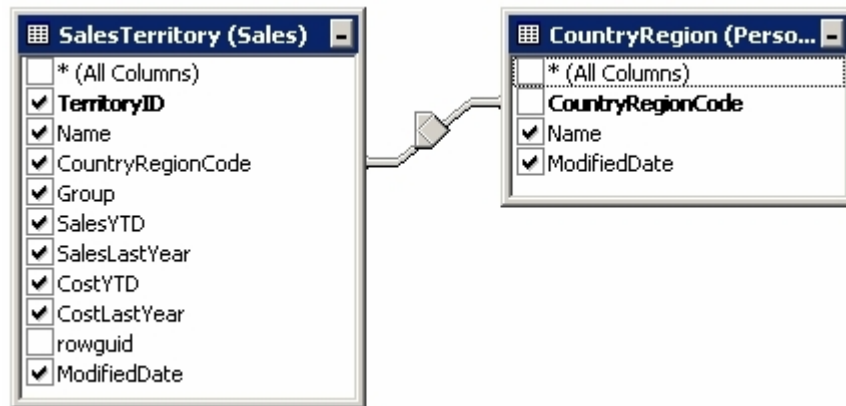
```

SELECT TOP (100) PERCENT Sales.Customer.CustomerID, Sales.Customer.CustomerType,
    Person.Contact.Title, Person.Contact.FirstName, Person.Contact.MiddleName,
    Person.Contact.LastName, Person.Contact.Suffix, Person.Contact.NameStyle,
    Person.Contact.EmailAddress, Person.Contact.Phone, Person.Address.AddressLine1,
    Person.Address.AddressLine2, Person.Address.PostalCode, Person.Address.City,
    Person.StateProvince.Name AS StateProvinceName, Person.CountryRegion.Name AS
    CountryRegionName, Sales.Customer.ModifiedDate AS MD_Customer,
    Sales.CustomerAddress.ModifiedDate AS MD_CustomerAddress,
    Person.Address.ModifiedDate AS MD_Address, Person.StateProvince.ModifiedDate AS
    MD_StateProvince, Person.CountryRegion.ModifiedDate AS MD_CountryRegion,
    i.ModifiedDate AS MD_Individual, Person.Contact.ModifiedDate AS MD_Contact,
    Dem.Col.value(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
    BirthDate', 'datetime') AS BirthDate,
    ...
FROM Sales.Customer LEFT OUTER JOIN Sales.CustomerAddress ON
    Sales.Customer.CustomerID = Sales.CustomerAddress.CustomerID LEFT OUTER JOIN
    Person.Address ON Sales.CustomerAddress.AddressID = Person.Address.AddressID LEFT
    OUTER JOIN Person.StateProvince ON Person.Address.StateProvinceID =
    Person.StateProvince.StateProvinceID LEFT OUTER JOIN Person.CountryRegion ON
    Person.StateProvince.CountryRegionCode = Person.CountryRegion.CountryRegionCode
    LEFT OUTER JOIN Sales.Individual AS i ON Sales.Customer.CustomerID = i.CustomerID
    LEFT OUTER JOIN Person.Contact ON i.ContactID = Person.Contact.ContactID OUTER
    APPLY i.Demographics.nodes(N'declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey";
    IndividualSurvey') AS Dem(Col)
WHERE (Sales.CustomerAddress.AddressTypeID = 2) OR
    (Sales.CustomerAddress.AddressTypeID = 3)

```

(Ve skriptu jsou z důvodu úspory místa vynechány některé sloupce získané z XML atributu Demographics z tabulky Individual. Celkem jde o 13 sloupců: BirthDate, MaritalStatus, Gender, YearlyIncome, TotalChildren, NumberChildrenAtHome, Education, Occupation, HomeOwnerFlag, NumberCarsOwned, TotalPurchaseYTD, DateFirstPurchase, CommuteDistance.)

Tabulka oblastí prodeje



Obr. 6.5: Výběr dat pro přenos do datového skladu

Tabulka dimenze oblasti prodeje pojmenovaná dimTerritory byla vytvořena na základě spojení tabulek SalesTerritory a CountryRegion (viz obr. 6.5) pomocí následujícího SQL příkazu:

```
SELECT      Sales.SalesTerritory.TerritoryID, Sales.SalesTerritory.Name,
            Person.CountryRegion.Name AS CountryRegionName, Sales.SalesTerritory.[Group],
            Sales.SalesTerritory.CountryRegionCode, Sales.SalesTerritory.SalesYTD,
            Sales.SalesTerritory.SalesLastYear, Sales.SalesTerritory.CostYTD,
            Sales.SalesTerritory.CostLastYear, Sales.SalesTerritory.ModifiedDate AS MD_SalesTerritory,
            Person.CountryRegion.ModifiedDate AS MD_CountryRegion
FROM        Sales.SalesTerritory LEFT OUTER JOIN Person.CountryRegion ON
            Sales.SalesTerritory.CountryRegionCode = Person.CountryRegion.CountryRegionCode
```

Tabulka pro časovou dimenzi

U časové dimenze byl zvolen jiný postup tvorby než u předcházejících dimenzí. Data nejsou přenášena ze zdrojové databáze, ale jsou uměle vytvořena. Důvodem je jednodušší způsob výstavby dimenze a také to, že vznikne ucelená řada datumů (což by u přenosu ze zdrojové DB nemuselo platit, například pokud by v určitý den nebyla vytvořena žádná objednávka).

Skript pro vytvoření tabulky dimTime reprezentující časovou dimenzi:

```
CREATE TABLE [dbo].[dimTime](
    [ID] [smallint] NULL,
    [Date] [datetime] NOT NULL,
    [DayNumberOfWeek] [smallint] NULL,
    [Day] [nvarchar](9) NULL,
```

[DayNumberOfMonth] [smallint] NULL,
[DayNumberOfYear] [smallint] NULL,
[WeekNumber] [smallint] NULL,
[Month] [nvarchar](9) NULL,
[MonthNumber] [smallint] NULL,
[Quarter] [smallint] NULL,
[Year] [smallint] NULL,
[Semester] [smallint] NULL)

Tabulka je následně naplněna řadou datumů, pokrývající celé období provozu zdrojové OLTP databáze:

```
INSERT INTO dimTime
VALUES ('1','2001-07-01 00:00:00.000','1','Sunday','1','182','27','July','7','3','2001','2');
...
INSERT INTO dimTime
VALUES ('1158','2004-08-31 00:00:00.000','3','Tuesday','31','244','36','August','8','3','2004','2');
```

Celkové relační schéma datového skladu

Na základě výše uvedených tabulek je vytvořeno relační schéma datového skladu, které přikládám v příloze 2.

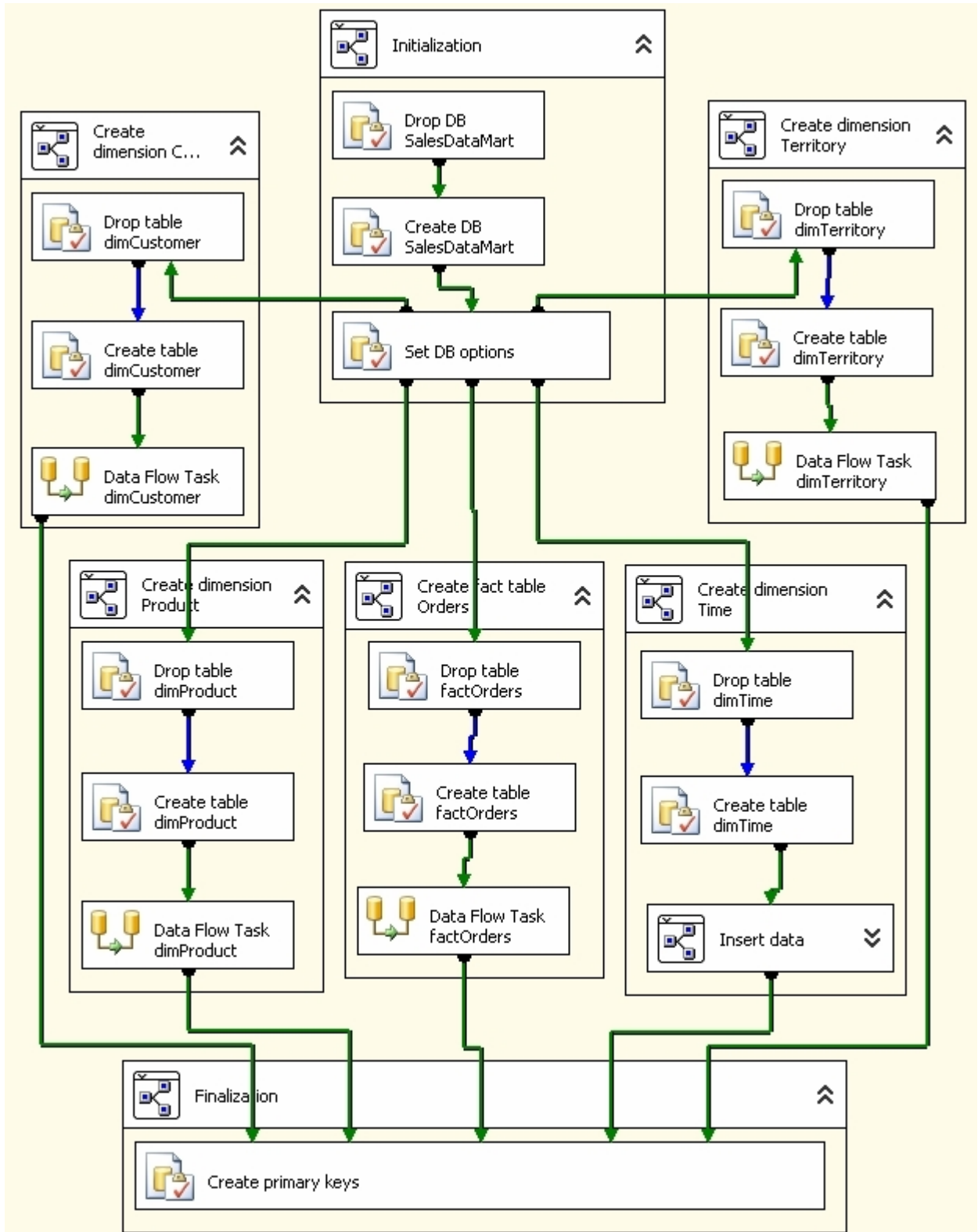
6.1.5 Sestavení

Na základě relačního schématu, vytvořeného ve fázi návrhu, byl vytvořen balíček integračních služeb (v MS SQL Server Integration Services), který vytváří samotnou databázi datového skladu, provádí její nastavení a samozřejmě také tvorbu tabulky faktů a tabulek jednotlivých dimenzí. Viz obr. 6.6 a následující popis.

V režimu návrhu projektu jsou na záložce ControlFlow umístěny ikony jednotlivých částečných úloh, z nichž se projekt skládá. Celkový diagram znázorňuje posloupnost úloh, které je potřeba postupně vykonat, aby se mohl zrealizovat náš záměr. Zásadní jsou v tomto diagramu komponenty Data Flow Task, které představují komplexní úlohy, skládající se obvykle z více bloků pro načítání dat, transformaci a uložení do cílové databáze. Proto je také pro tento modul vyčleněna samostatná záložka pracovní plochy návrhového prostředí.

Inicializace

V rámci inicializace je odstraněna případná, již existující, databáze SalesDataMart. Následně je vytvořena nová databáze s tímto jménem a jsou jí nastaveny základní vlastnosti. Výchozí velikost databáze je nastavena na 10MB s růstem po 10%.



Obr. 6.6: Schéma balíčku integračních služeb

Vytvoření tabulky faktů a dimenzí

Úlohy vedoucí k vytvoření jednotlivých tabulek jsou seskupeny do samostatných kontejnerů, které vždy obsahují odstranění a znovu-vytvoření dané tabulky a následné naplnění daty v komponentě Data Flow Task.

Pro zjednodušení procesu vytváření tabulek lze využít průvodce SSIS Import and Export Wizard. Zde je nutné pouze nastavit ovladač pro přístup k databázi (v našem případě SQL Native Client), název serveru, typ ověřování serveru (authentication) a jméno databáze. To vše samozřejmě pro zdroj i cíl dat. Následně je již nutné pouze specifikovat data, která mají být přenesena. K tomu je možné využít SQL příkazy, které byly vytvořeny ve fázi návrhu.

Závěrečné nastavení

V poslední komponentě, ve které se sbíhají všechny prováděné toky Control Flow, jsou tabulkám nastaveny primární klíče.

6.1.6 Produkce

Po vytvoření a odladění balíčku je možné jej nasadit. U projektu se nastaví parametr Project Deployment na hodnotu True a provede se build. V podadresáři Deployment tím vzniknou soubory „navez_projektu.dtsx“ a „navez_projektu.SSISDeploymentManifest“. Tyto soubory se přenesou na cílový počítač a zde se spustí manifest(soubor „navez_projektu.SSISDeploymentManifest“). Po aktivování manifestu se spustí průvodce „Package Installation Wizard“, ve kterém je možné vybrat, zda bude balíček uložen v souborovém systému, nebo v úložišti MS SQL Serveru.

Z hlediska bezpečnosti mohou být balíčky při přenosu zašifrované a digitálně podepsané. Když je zvoleno jako cílové úložiště MS SQL Server, můžeme nastavit dostupnost balíčku jen pro určité role.

Nyní je možné balíček spustit. To lze provést příkazem „Run package“ u daného balíčku v MS SQL Server Management Studiu (je nutné se připojit k serveru Integration Services).

Vytvořený balíček integračních služeb provádí iniciální naplnění datového skladu. To obvykle probíhá pouze jednou a lze jej provést také jednoduše spuštěním balíčku příkazem Execute Package v Solution Exploreru vývojového prostředí.

Pro průběžné doplňování datového skladu novými daty by bylo nutné balíček upravit tak, aby do skladu přenášel pouze nové položky. Tento balíček by se pak spouštěl v určených intervalech, což je možné zautomatizovat pomocí služby SQL Server Agent.

6.2 Vytvoření datové kostky v SSAS

Pokud máme dobře vytvořený datový sklad, je již poměrně jednoduché nad ním vytvářet OLAP kostky pomocí analytických služeb SQL Serveru 2005.

6.2.1 Datové zdroje

Prvním krokem je definice zdroje dat, kterým je v našem případě datový sklad, vytvořený pomocí integračního balíčku, popsaného v kapitole 6.1. Opět je k tomu možné použít připraveného průvodce Data Source Wizard, který ulehčí nastavení připojení k databázi datového skladu a umožní také vybrat typ autentifikace pro přístup k DB.

6.2.2 Pohledy na datové zdroje

Datový sklad často obsahuje mnoho tabulek, které pro vytváření OLAP kostek nevyužijeme. Pro výběr tabulek, se kterými chceme pracovat, slouží právě pohledy na datové zdroje. K jejich vytvoření slouží průvodce Data Source View Wizard. Pomocí něj je možné vybrat potřebné tabulky a nastavit vztahy mezi nimi (průvodce je dokáže sám navrhnout na základě shody jmen sloupců s primárními klíči). Vztahy, které průvodce nedokázal identifikovat, je možné později doplnit v designeru pohledů na datové zdroje. V našem případě bylo nutné doplnit vztahy mezi tabulkou faktů a tabulkou časové dimenze (sloupce nemohly mít shodný název, protože z tabulky faktů se do časové dimenze odkazuje více položek – datum objednávání, splatnosti a dodání).

6.2.3 Vypočítané atributy

Tabulky dimenzí mohou kromě sloupců tabulky obsahovat i vypočítané atributy. Obvykle se tak vytváří různé složené nebo jinak zkonvertované atributy. Ve vytvořeném pohledu na datové zdroje jsem pomocí této techniky přidal následující atributy:

Tabulka zákaznické dimenze:

- Name of Customer – celé jméno zákazníka složené z jeho rodného jména a příjmení:
FirstName + ' ' + LastName
- Type of Customer – typ zákazníka, rozlišující zda jde o koncového zákazníka nebo prodejce:
CASE WHEN CustomerType = 'I' THEN 'Internet Customer' ELSE 'Reseller' END

Tabulka produktové dimenze:

- NameOfProductLine – celý název produktové linie:
CASE
WHEN ProductLine = 'R' THEN 'Road'

```

WHEN ProductLine = 'M' THEN 'Mountain'
WHEN ProductLine = 'T' THEN 'Touring'
WHEN ProductLine = 'S' THEN 'Standard'
ELSE 'Components'
END

```

- SizeOfProduct – atribut udávající rozměry produktu:

```

CASE
  WHEN SizeUnitMeasureCode IS NULL THEN
    Size
  ELSE
    Size + ' ' + SizeUnitMeasureCode
END

```

- WeightOfProduct – hmotnost produktu:

```

CONVERT (nvarchar(10),Weight) + ' ' + WeightUnitMeasureCode

```

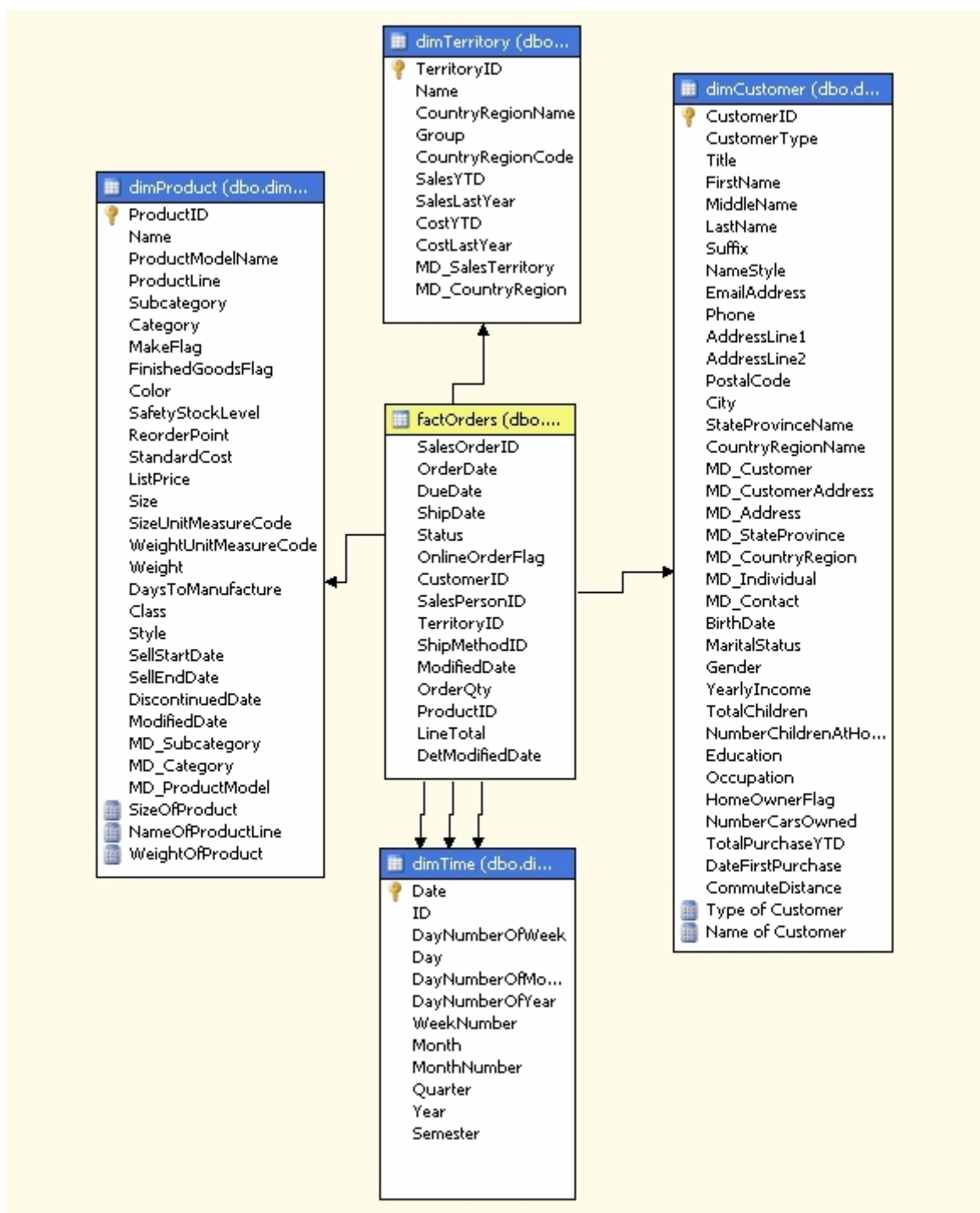
6.2.4 Datová kostka

Datová kostka je vytvářena pomocí průvodce Cube Wizard. Jeho součástí je metoda Auto Build, která analyzuje datové zdroje a identifikuje fakta a dimenze ve vybraném pohledu na zdroj dat. V dalších krocích je možné tuto automatickou analýzu upravit podle svých představ. Dále je nutné označit tabulku, která bude podkladem pro vytvoření časové dimenze. V tomto bodě jsou již všechny nutné údaje zadány a jako poslední úkol zbývá vybrat fakta a dimenze, které budou do právě vytvářené kostky zahrnuty. Průvodce již sám navrhne hierarchie dimenzí. Ty je pak možné dopracovat podle vlastních požadavků. Výsledná struktura datové kostky je uvedena na obrázku 6.7.

6.2.5 Hierarchie dimenzí

Ve vytvářené kostce jsem nadefinoval následující hierarchie:

<u>Časová dimenze:</u>	Rok – kvartál – měsíc – den
<u>Zákaznická dimenze:</u>	Stát – provincie – město – jméno zákazníka
<u>Dimenze oblasti prodeje:</u>	Geografické území – stát – oblast prodeje
<u>Produktová dimenze:</u>	Kategorie – podkategorie – produkt
	Produktová linie – model – produkt



Obr. 6.7: Přehled struktury datové kostky vytvořené pomocí analytických služeb

6.2.6 Cacheování, nastavení míry agregace

Po vytvoření kostky je důležité nastavit optimální kompromis mezi prostorem, který kostka zabere na disku a rychlostí výpočtu a přístupu k datům. To lze zajistit nastavením cacheování a míry agregace.

Běžně známy jsou tři základní typy ukládání analytických dat – vícerozměrné OLAP (MOLAP), relační OLAP (ROLAP) a hybridní OLAP (HOLAP). Při použití MOLAP jsou všechny

hodnoty a agregovaná data uloženy ve vícerozměrném úložišti analytických služeb, což dává rychlejší odezvy na dotazy. ROLAP, kdy hodnoty zůstávají v tabulce faktů v relační databázi a agregace jsou uloženy ve speciálních samostatných relačních tabulkách, dává kratší dobu zpracování rychle, ale pomalejší odezvy na dotazy. Kompromisní volba HOLAP udržuje hodnoty v relační databázi a agregace ukládá vícerozměrně (více viz kapitola 4.6).

V SQL Serveru 2005 byla omezení jednotlivých typů ukládání zmírněna technikou proaktivního cacheování. Ta umožňuje použít jako základové úložiště ROLAP a zajistí cacheování rychle do média MOLAP. Pak je možné zvolit, jak často bude obsah cache uvolněn. Podle nastavení cacheování může být rychle MOLAP ihned po provedení změn v podkladových datech zneplatněna. V takovém případě jsou dotazy provedeny proti základní rychli ROLAP, verze MOLAP je přepočítána na pozadí a po dokončení zpracování je překlopena zpět. Je zde také možnost, aby „stará“ data MOLAP byla používána i během přepočítávání a zpracování změněných dat. Tím se zvyšuje výkon, ale data nejsou vždy zcela aktuální. (Informace o technice proaktivního cacheování čerpány z [22].)

Další možností optimalizace je nastavení míry předpočítaných agregací. Tu je možné nastavit podle prostoru, který agregace zaberou na disku nebo podle procentuálního zvýšení výkonnosti, popřípadě podle doby výpočtu agregací.

Vzhledem k povaze vyvíjené aplikace jsem zvolil jako úložiště multidimenzionálních dat MOLAP s nejvyšší možnou mírou agregace. Předpokládá se totiž pravidelná aktualizace datového skladu v průběhu noci (kdy je vzhledem k rozsahu databáze dostatek času pro generování datových kostek i požadovaných agregací) a potřeba vysokého výkonu během dne.

Nyní následuje fáze zavedení dat, kterou je možné provést příkazem Deploy z menu Build.

6.3 Tvorba, správa a distribuce reportů s SSRS

Reportovací služby představují podstatný základ prezentační vrstvy celého Business Intelligence řešení, postaveného na MS SQL Serveru. V této kapitole bude popsán návrh reportů pro vyvíjenou aplikaci, ale také jejich správa a možnosti doručení cílovým adresátům.

6.3.1 Filozofie reportovacích služeb

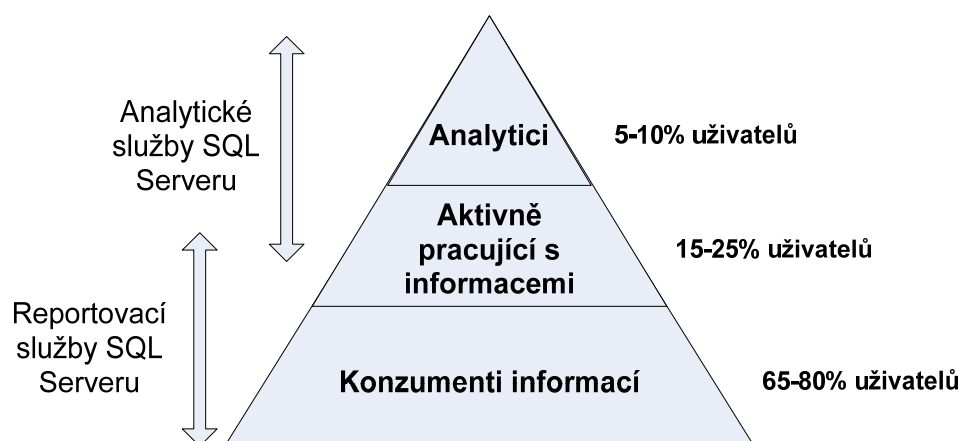
Reporting Services v SQL Serveru 2005 je celá sada technologií, jež nabízejí vývoj, správu a zobrazování datových sestav. Zahrnuje nástroje pro návrh sestav, jednorázové dotazy i administraci.

Definice sestav se ukládají v jazyce XML, publikují do databáze v SQL a ovládají přes webové rozhraní. Služba ve Windows ovládá zpracování sestav, jež odděluje získávání dat od zobrazování sestav. Jednu sestavu tak lze zobrazit mnoha uživatelům v různých výstupních formátech.

Uživatelé

Pro objasnění pozice reportovacích služeb je důležité si uvědomit, kteří uživatelé budou tyto služby využívat. V kontextu celého BI řešení existují tři základní typy uživatelů (viz také obr. 6.8):

- **Analytici:** Jejich úlohou je vybírat data, která jsou vhodná pro analýzy, analyzovat je a na základě výsledků analýz poskytovat informace pro podporu rozhodování.
- **Aktivně pracující s informacemi:** Pracovníci této skupiny jednak data analyzují, zpracovávají a potřebují je v různé formě zobrazovat.
- **Konzumenti informací:** Tvoří většinu uživatelů. Dostávají výpisy dat ve formě dvourozměrných sestav. Přístup k datům ve formě reportů zabezpečují právě reportovací služby.



Obr. 6.8: Zaměření uživatelů využívajících služeb BI SQL Serveru

Ještě zásadnější rozdělení reportů vychází ze skutečnosti, zda bude přístup k datům umožněn pouze pracovníkům firmy nebo například obchodním partnerům. Z tohoto hlediska je možné rozlišovat následující dvě kategorie:

- **Enterprise a Embedded Reporting:** Prezentace dat v rámci podniku pomocí podnikových portálů, intranetu nebo informačních systémů.
- **B2B Reporting:** Prezentace dat obchodním partnerům, například přes firemní webový portál.

Komponenty

Reportovací služby se skládají z těchto komponent:

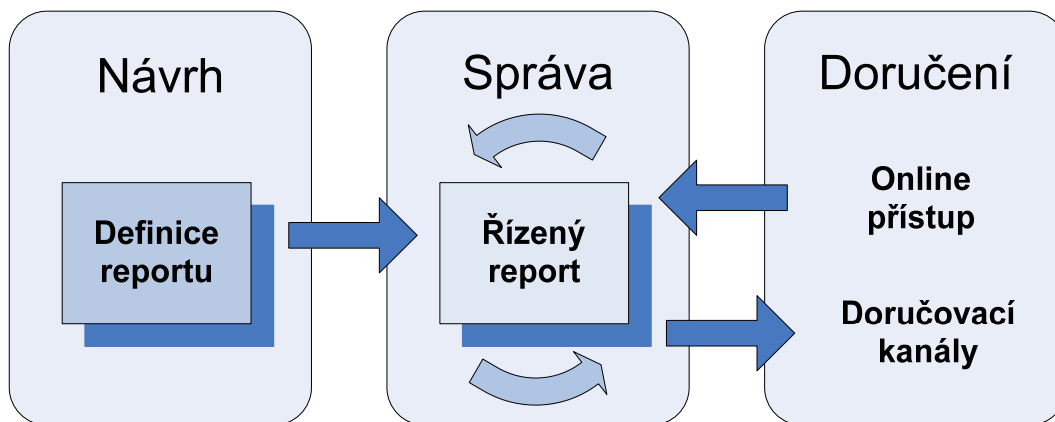
- **Report Server:** Serverová komponenta, zahrnující webové a reportovací služby. (Webové služby představují množinu programátorských rozhraní, která využívají klientské aplikace pro přístup k reportům.)
- **Report Model Designer:** Nástroj pro návrh modelů reportů.
- **Report Designer:** Vizuální nástroj pro vytváření reportů. Je součástí produktu MS Visual Studio.
- **Report Manager:** Webová aplikace pro správu a prohlížení reportů.
- **Report Builder:** Umožňuje vytvářet a nasazovat sestavy pomocí modelu vytvořeného v Report Model Designeru.

Koncepce reportovacích služeb v SQL Serveru je označována jako „managed reporting“. Její hlavní výhodou je skutečnost, že metadata, na jejichž základě se generují reporty, jsou uložena centrálně. Mohou tedy být i centrálně spravována. To znamená, že je možné reporty organizovat do různých sestav, adresářů a zpřístupňovat je na různé úrovni uživatelům, popř. skupinám uživatelů. Přístup k reportům je pak možné řídit také přes přístup k adresářovým strukturám.

Životní cyklus reportu

Pomocí reportů můžeme zobrazovat data z relačních i analytických databází. V případě, že data čerpáme z datového skladu, je nutné při návrhu reportu brát v úvahu i životní cyklus datového skladu. Cílem je, aby plynule navazovaly kroky naplnění datového skladu, vypočítání OLAP kostek a následně generování reportů z těchto zdrojů.

Z jiného hlediska je možné rozdělit životní cyklus reportu na tři základní fáze: návrh, správa a doručení (viz obrázek 6.9). Těmto fázím se věnují i následující tři podkapitoly.



Obr. 6.9: Životní cyklus reportu(převzato z [1])

6.3.2 Návrh reportu

Pro vývoj reportu je možné využít MS Visual Studio, případně Business Intelligence Development Studio, což je verze MS Visual Studia, omezená jen na projekty Business Intelligence, dodávaná spolu s MS SQL Serverem.

Výsledkem návrhu reportu je kód v jazyce RDL, který se zapisuje ve formě XML. Návrh proto velmi usnadňuje vizuální návrhové prostředí Report Designeru.

Data, prezentovaná prostřednictvím reportů, mohou být zobrazována v různých formách tabulek, datových matic, grafů a dokumentů, provázaných hypertextovými odkazy. Variabilitu navíc podporují dynamické a hierarchické parametry, možnosti třídění, filtrování, seskupování dat a výpočet částečných výsledků pomocí agregačních funkcí.

Vytvořené reporty

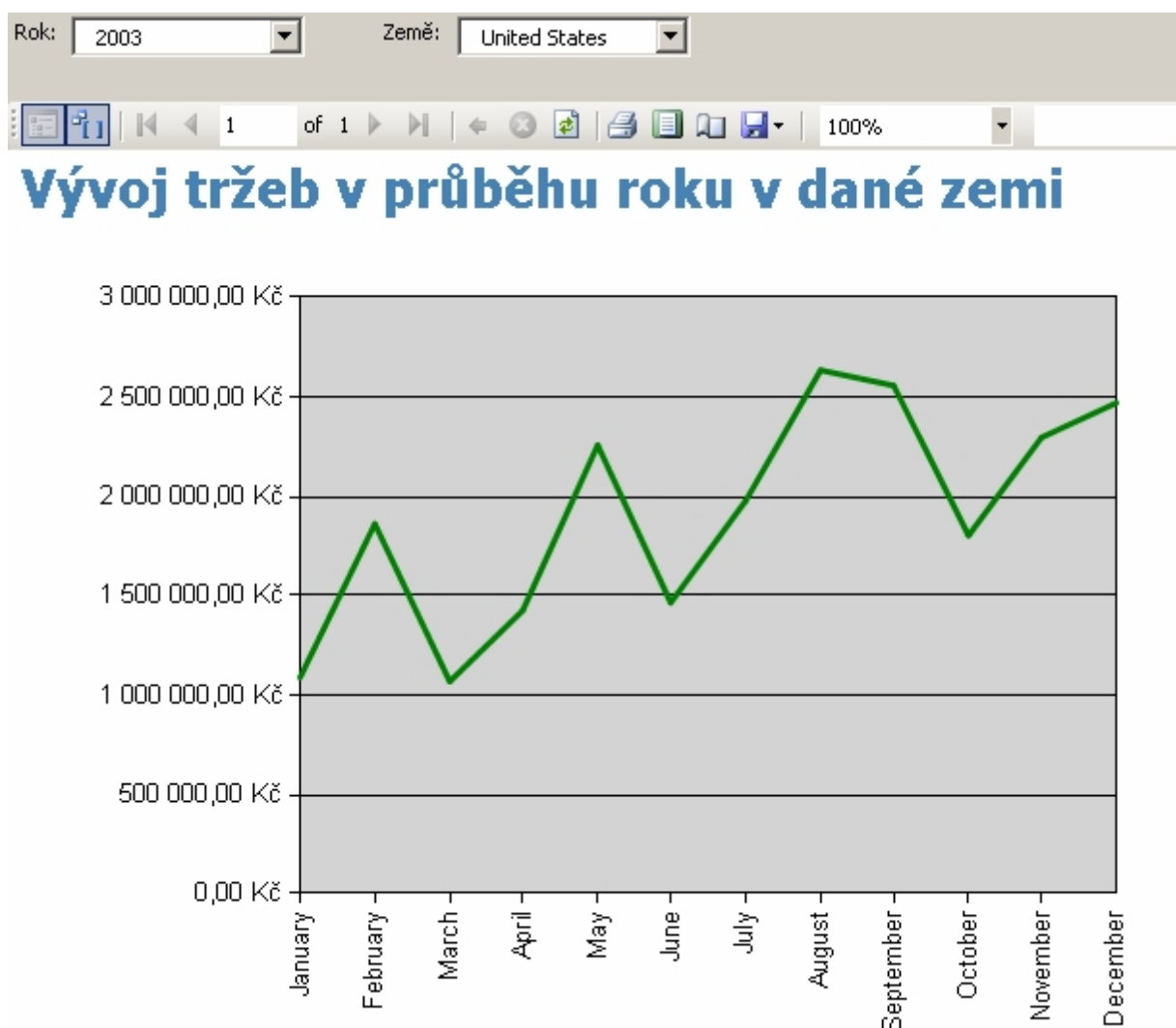
V rámci vyvíjené ukázkové aplikace byl dosud vytvořen datový sklad, respektive datový trh a nad ním OLAP kostka. Na základě této kostky jsem dále pomocí reportovacích služeb vytvořil několik reportů, demonstrujících možnosti této technologie.

Prvním krokem před samotným vytvářením reportů je definice datových zdrojů, které jsou sdílené pro všechny reporty v rámci projektu. Na výběr je několik typů zdrojů dat (MS SQL Server, Oracle, ODBC, XML, ...). Pro přístup k vytvořené OLAP kostce je nutné zvolit jako typ databáze MS SQL Server Analysis Services. Dále je potřeba nastavit název serveru a databáze.

Pokud máme nadefinovány datové zdroje, je možné na jejich základě vytvářet reporty. Opět je k dispozici průvodce, který návrh usnadňuje. Integrovaný Query Builder umožňuje vizuální výběr dimenzí a faktů z datové kostky, nastavení filtrů a také parametrů reportu. Dále je možné vybrat ze dvou typů rozvržení dat - v běžné tabulce nebo matici, způsob uspořádání jednotlivých dimenzí a také šablonu designu, která se má použít. Tvorba reportu se tímto velmi zjednoduší.

Pro tvorbu složitějších reportů můžeme využít návrhové prostředí Report Designeru. Je rozděleno na tři části – pod záložkou Data je možné upravovat výběr dimenzí a faktů použitých v reportu, v záložce Layout se vytváří vizuální vzhled a rozmístění jednotlivých prvků reportu, pod poslední záložkou Preview je možné zobrazit výslednou podobu reportu.

Ve chvíli, kdy máme reporty vytvořené, následuje fáze zavedení dat na server, kterou je možné provést příkazem Deploy z menu Build. Report server je možné ovládat přes web – je přístupný přes URL adresu <http://localhost/ReportServer> (v případě že server běží na lokálním počítači). V případě, že tuto adresu zadáme do prohlížeče, objeví se stránka se seznamem projektů a datových zdrojů. Pokud vybereme konkrétní projekt, zobrazí se nám jednotlivé reporty (ukázka reportu viz obrázek 6.10).



Obr. 6.10: Ukázka reportu

6.3.3 Správa reportu

Správu reportů zajišťuje nástroj Report Manager, který je defaultně přístupný na URL adrese <http://localhost/reports>.

Nástroj umožňuje prohlížet reporty, jejich popis, parametry, datové zdroje, cacheování reportů, historii a zabezpečení.

6.3.4 Možnosti doručení reportu

Hlavním důvodem pro nasazení reportovacích služeb je generování výstupů v elektronické nebo papírové podobě. Výhodou elektronických reportů je pak nejen jejich interaktivita (možnost nastavování parametrů, zanořování do hierarchií dimenzí, použití odkazů na další informace atd.), ale i moderní způsoby doručování reportů.

Způsoby doručování reportů lze rozdělit na dva základní typy:

1) Doručování reportů klientovi

To je možné nastavit v Report Manageru pod záložkou New Subscription u každého reportu. Je třeba nastavit způsob doručení (e-mailem nebo do zvoleného adresáře), interval doručování a formát reportu (XML, CSV, TIFF, PDF, HTML,...).

2) Klient sám přistupuje k úložištím reportů

V tomto případě existuje mnohem více možností, jak se může klient k vytvořeným reportům dostat. Základním způsobem je přístup přes Report Server, kde je možné se dostat pomocí odkazů k jednotlivým reportům, které si může uživatel následně exportovat do zvoleného formátu. Reportovací služby, díky podpoře rozhraní webových služeb, umožňují integrovat reporty do různých firemních aplikací, ať už klasických desktopových nebo webových.

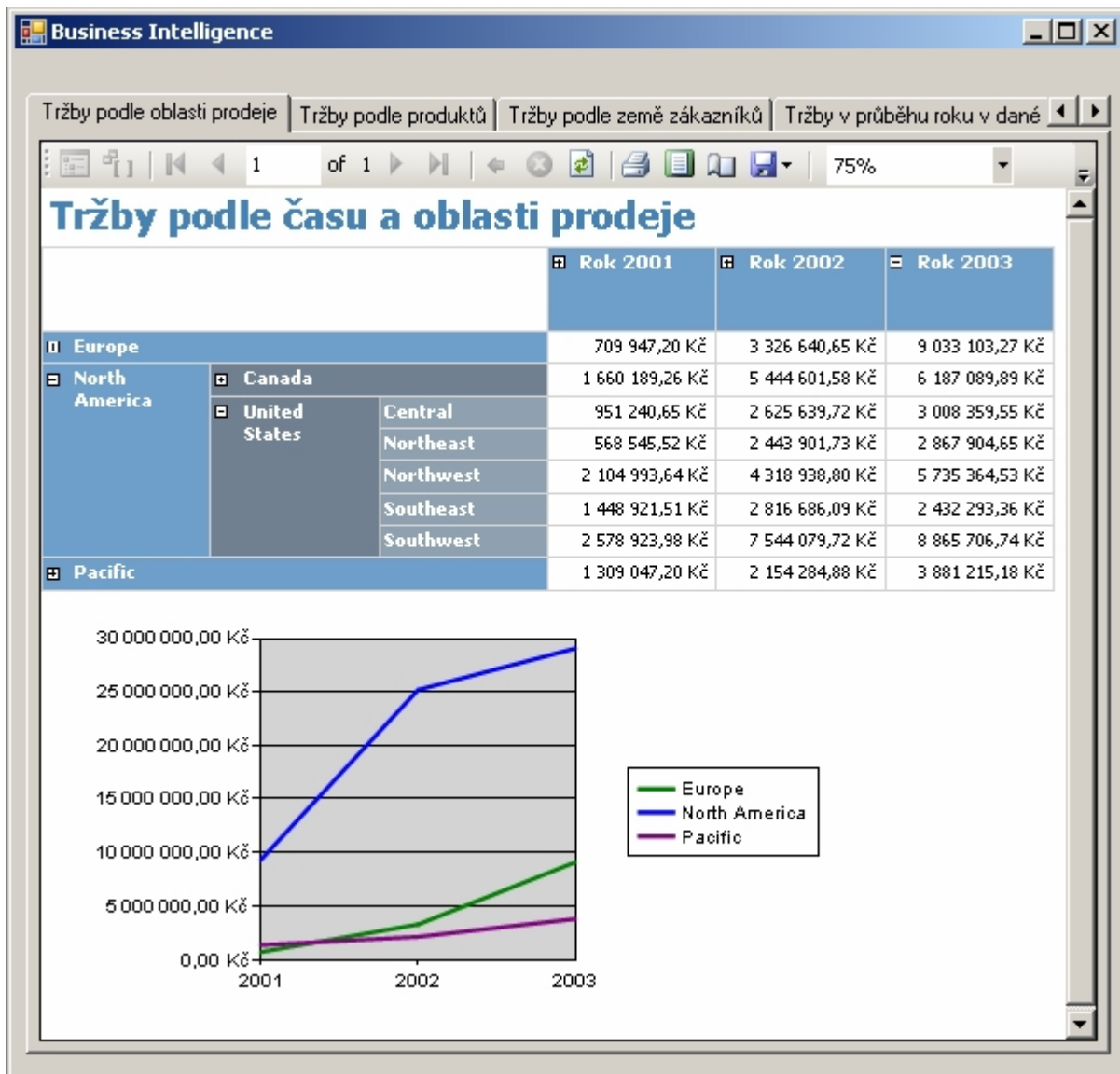
6.4 Prezentace dat v klientských aplikacích

U klientských aplikací se vymezuje základní rozdělení na tlusté a tenké klienty. Výhodou tlustých klientů je využití značného výpočetního výkonu dnešních klientských stanic a také případná práce v odpojeném režimu. Nevýhodou je však náročná a nákladná správa lokálního softwaru a tím pádem i jeho cena. Pod pojmem tenký klient chápeme obecně jakékoliv zařízení (počítač, notebook, mobilní telefon,...) s prohlížečem webového obsahu. Výhodou použití tenkého klienta je hlavně podpora širokého spektra zařízení, platforem a nízké náklady na hardware a software pro uživatele. Hlavní nevýhodou je použití bezstavového protokolu HTTP a nemožnost práce offline. V rámci vyvíjené ukázkové aplikace byly vytvořeny oba typy aplikací, umožňující přístup k datům pomocí reportovacích služeb.

6.4.1 Windows aplikace

Pro vývoj windows aplikací, spolupracujících s reportovacími službami, existuje užitečná komponenta ReportViewer. Po nastavení adresy Report Serveru a cesty ke konkrétnímu reportu dokáže komponenta zvolený report zobrazit v aplikaci včetně všech možností interakce, které report umožňuje.

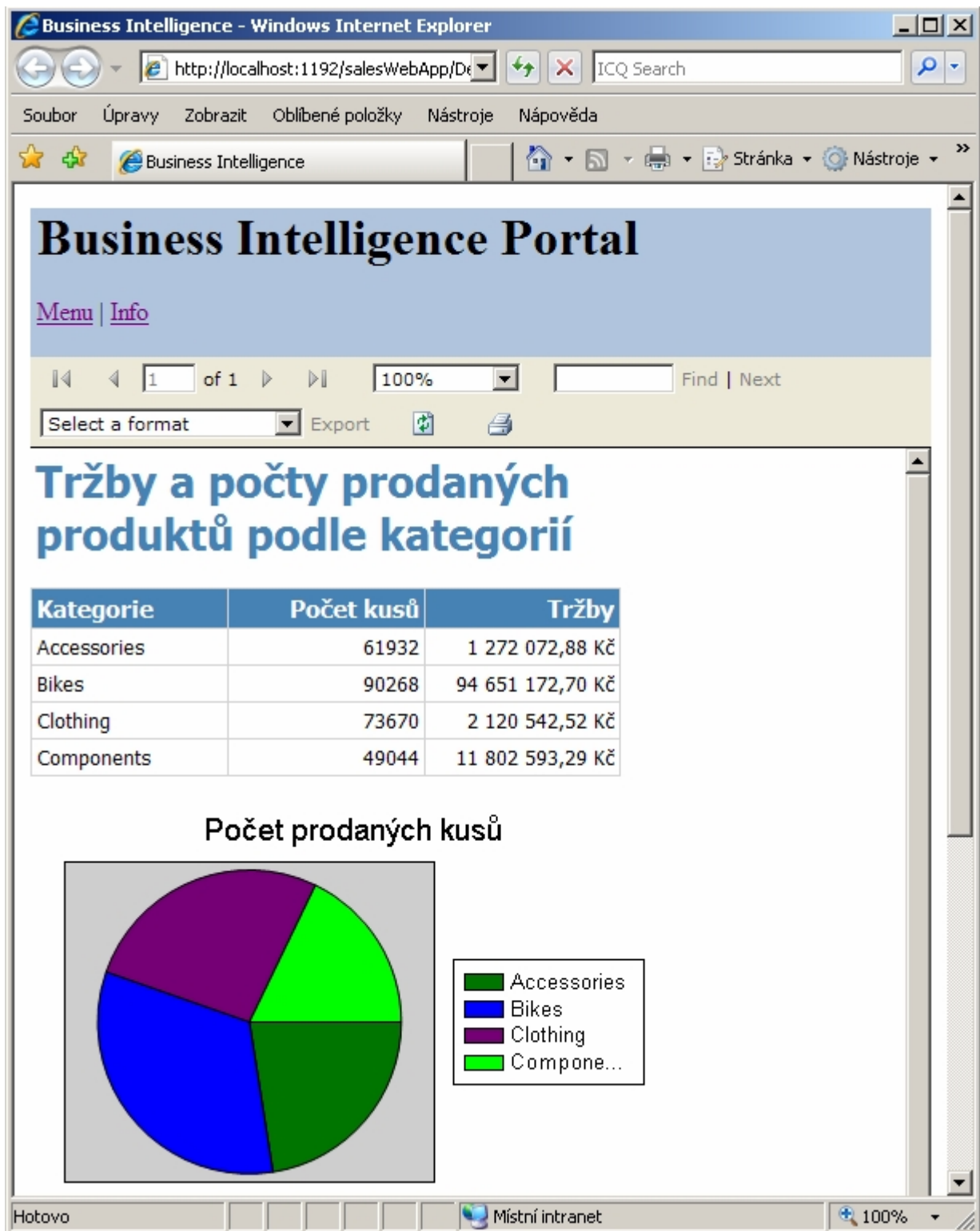
Vytvořená aplikace demonstruje pouze přístup k reportům. Nic by však nebránilo integrovat tuto funkčnost do firemních aplikací a informačních systémů tak, aby koncoví uživatelé našli výsledky řešení BI v jim dobře známých aplikacích.



Obr. 6.11: Windows aplikace

6.4.2 Webová aplikace

Vytvořená webová aplikace používá, stejně jako windows aplikace, komponentu ReportViewer. Oproti prohlížení reportů přímo na Report Serveru je zde opět výhoda v tom, že je možné reporty zakomponovat do stávajícího webového portálu firmy.



Obr. 6.12: Webová aplikace

7 Závěr

Oblast datových skladů a OLAP analýz je velmi široká, přesto jsem se pokusil o její částečné zmapování a zasazení do kontextu Business Intelligence systémů.

Tyto systémy obvykle pracují s velkým množstvím dat, pro jejichž zvládnutí jsou nepostradatelná robustní databázová a vývojová prostředí. Jedním z nich je právě MS SQL Server, s jehož pomocí jsem v rámci diplomového projektu vytvořil aplikaci, která získané teoretické vědomosti demonstruje v praxi.

V návaznosti na tuto práci by bylo možné mnohem šířeji pojednat o prezentační vrstvě Business Intelligence – oblast reportovacích služeb SQL Serveru dává mnohonásobně více možností, než bylo možné popsat v rámci této práce. Zajímavým pro další zkoumání je jistě také segment mobilních zařízení jako klientů pro přístup k datům.

Literatura

- [1] Lacko, Luboslav. *Business Intelligence v SQL Serveru 2005*. Brno, Computer Press, 2006. 388 s. ISBN 80-251-1110-5.
- [2] Lacko, Luboslav. *Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle*. Brno, Computer Press, 2003. 488 s., 1 CD-ROM. ISBN 80-7226-969-0.
- [3] Planková, Jindra. *ITlib*, 2007, č. 3. ISSN 1335-793X. Dostupné na URL:
<http://www.cvtisr.sk/itlib/itlib073/plankova.htm> (květen 2008).
- [4] Peterka, Jiří. *COMPUTERWORLD*. 1994, č. 42. ISSN 1210-9924. Dostupné na URL:
<http://www.earchiv.cz/a94/a442c120.php3> (květen 2008).
- [5] Maciaszek, L.A., Liong, B.L. *Practical Software Engineering*. Pearson 2005. 864 s. ISBN 0 321 20465 4. Dostupné na URL:
https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/AIS-IT/lectures/1_PSE_Chapter1.pdf (květen 2008).
- [6] Loffelman, J., *IT Systems*. 2007, č.5. ISSN 1802-615X. Dostupné na URL:
<http://www.systemonline.cz/rizeni-vyroby/informacni-systemy-v-planovani-a-rizeni-vyroby.htm> (květen 2008).
- [7] Mráz, J., *Podnikové informační systémy*. Dokument dostupný na URL:
https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/AIS-IT/lectures/10_ICT-Podniku.pdf (květen 2008).
- [8] Kotásek, P., *DWH/BI systémy, datové sklady*. Dokument dostupný na URL:
https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ZZN-IT/lectures/P6_DWH_BI.pdf (květen 2008).
- [9] Barták, M., *Vliv IT nástrojů pro udržování dat na komunikaci v podniku*. Dokument dostupný na URL: www.agris.cz/etc/textforwarder.php?iType=2&iId=137620&PHPSESSID=3e (květen 2008).
- [10] Vrána, J., *IT Systems*. 2001, č.5. ISSN 1802-615X. Dostupné na URL:
<http://www.systemonline.cz/clanky/datove-sklady-a-jejich-optimalizace.htm> (květen 2008).
- [11] Wikimedia Foundation, *OLAP*, dokument dostupný na URL:
<http://cs.wikipedia.org/wiki/OLAP> (květen 2008).
- [12] Bartík, V., *Datové sklady a technologie OLAP pro získávání znalostí*. Dokument dostupný na URL: https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ZZN-IT/lectures/02_DWOLAP.pdf (květen 2008).
- [13] *Datové sklady a OLAP*. Dokument dostupný na URL:
<http://datamining.xf.cz/view.php?cislocclanku=2002102808> (květen 2008).

- [14] Microsoft Corporation, *Přehled produktu SQL Server 2005*, dokument dostupný na URL: <http://www.microsoft.com/cze/windowsserversystem/sql/prodinfo/overview/default.aspx> (květen 2008).
- [15] Microsoft Corporation, *Webcasty Business Intelligence*, dokument dostupný na URL: <http://www.microsoft.com/cze/msdn/webcasts/default.aspx> (květen 2008).
- [16] Šanda, A. *COMPUTERWORLD*. 1999, č. 26. ISSN 1210-9924. Dostupné na URL: <http://archiv.computerworld.cz/cwarchiv.nsf/clanky/F738F470BE703B67C12569B00055B833?OpenDocument> (květen 2008).
- [17] Kučera, M. *IT Systems*. 2001, č.5. ISSN 1802-615X. Dostupné na URL: <http://www.systemonline.cz/clanky/dva-zpusoby-budovani-datoveho-skladu.htm> (květen 2008).
- [18] Microsoft Corporation, *Using SQL Server 2005 Integration Services to Populate a Kimball Method Data Warehouse*, dokument dostupný na URL: <http://msevents.microsoft.com/cui/WebCastEventDetails.aspx?culture=en-US&EventID=1032297073&CountryCode=US> (květen 2008).
- [19] Kimball Group, *The Microsoft Data Warehouse Toolkit Book Website*, dokument dostupný na URL: <http://www.msftdwtoolkit.com/ToolsandUtilities/ToolandUtilities.htm> (květen 2008).
- [20] Microsoft Corporation, *AdventureWorks OLTP Database Diagram*, dokument dostupný na URL: <http://www.microsoft.com/Downloads/details.aspx?familyid=0F6E0BCF-A1B5-4760-8D79-67970F93D5FF&displaylang=en> (květen 2008).
- [21] Microsoft Corporation, *AdventureWorks Data Dictionary*, dokument dostupný na URL: <http://msdn.microsoft.com/en-us/library/ms124438.aspx> (květen 2008).
- [22] Brust, A. J., Forte, S. *Mistrovství v programování SQL Serveru 2005*. Brno, Computer Press, 2007. 848 s. ISBN 978-80-251-1607-4.

Seznam příloh

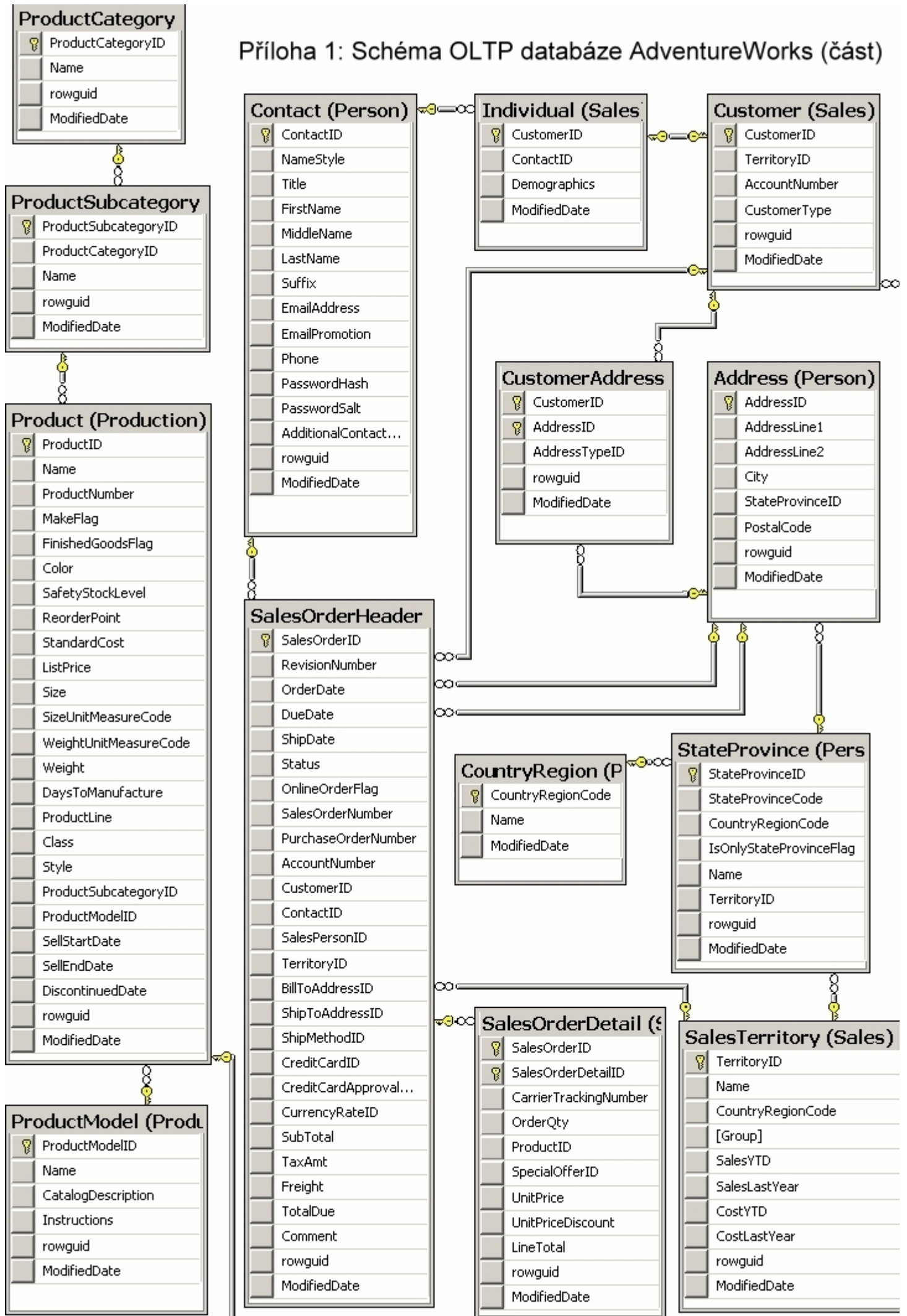
Příloha 1. Schéma OLTP databáze AdventureWorks (část)

Příloha 2. Celkové relační schéma datového skladu

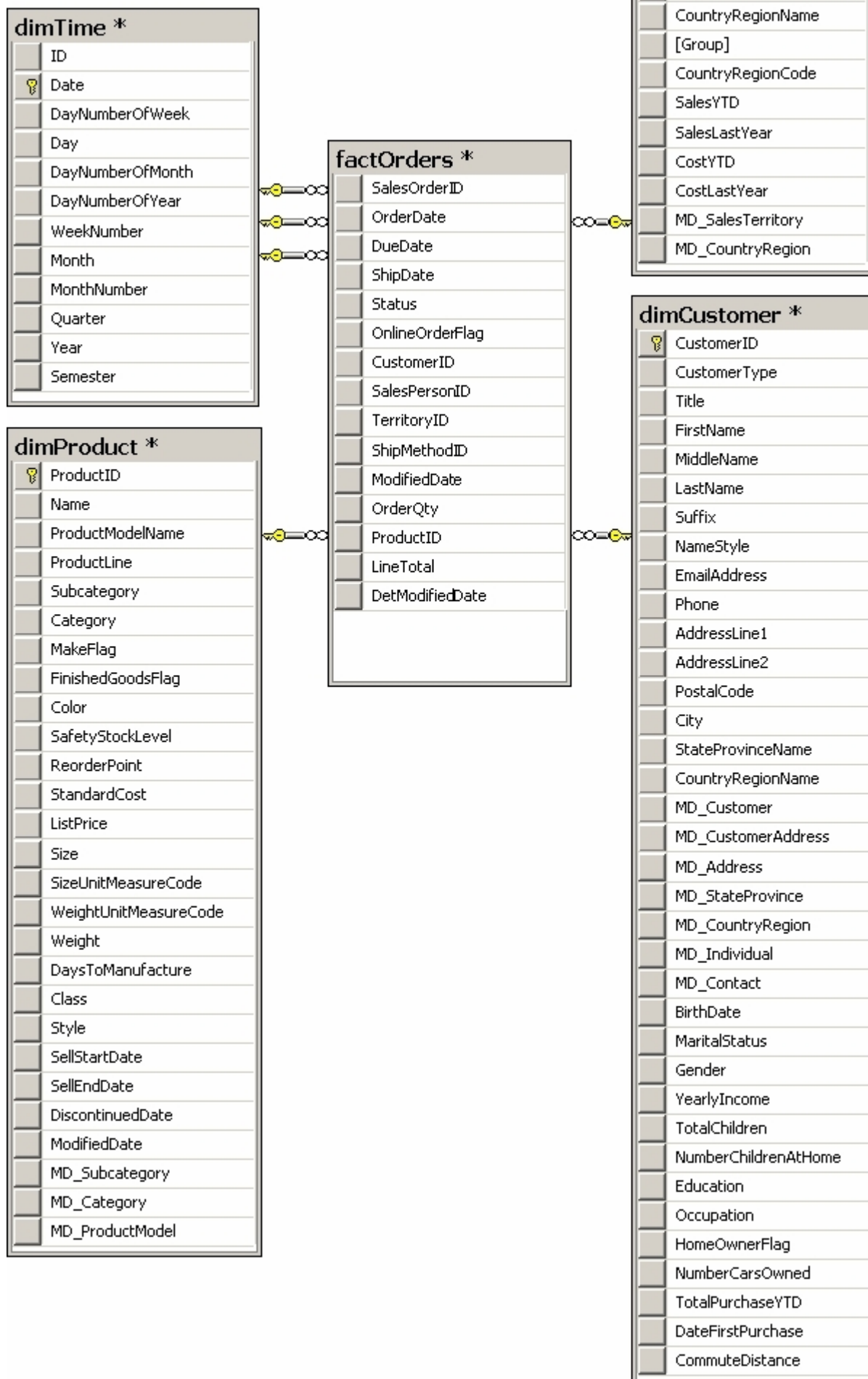
Příloha 3. Manuál

Příloha 4. CD

Příloha 1: Schéma OLTP databáze AdventureWorks (část)



Příloha 2: Celkové relační schéma datového skladu



Manuál

Ukázková aplikace sestává celkem z pěti projektů. K jejich vytvoření byl použit produkt MS SQL Server 2005 SP2 a MS Visual Studio 2005. Ke spuštění výsledných aplikací postačí instalace MS SQL Serveru a webového serveru IIS.

Projekty se nachází na přiloženém CD v adresáři aplikace. Univerzálním návodem k jejich spuštění je otevření projektu ve vývojovém prostředí MS Visual Studio 2005 a stisknutí zelené šipky na pruhu nástrojů. Níže je uveden konkrétnější popis k jednotlivým projektům.

1) SalesDataMart

Balíček integračních služeb, který vytváří datový sklad.

2) SalesDataCube

Olap kostka vytvořená nad datovým skladem.

3) SalesReport

Projekt obsahující reporty vytvořené nad datovým skladem a OLAP kostkou.

4) SalesWinApp

Windows aplikace s integrovanými reporty z projektu SalesReport.

5) SalesWebApp

Webová aplikace demonstrující webový portál s reporty.