Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Diplomová práce

Analytické databáze

Plzeň, 2012 Michal Šlajs

Poděkování

Na tomto místě bych rád poděkoval vedoucímu diplomové práce Ing. L. Stauberovi za odborné vedení a předané zkušenosti. Dále děkuji garantovi Ing. J. Weinrebovi CSc. za praktické rady a doporučení.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni, dne 20. května 2012 .....

Michal Šlajs

Anotace

Diplomová práce se zabývá analýzou vybraného Business Intelligence řešení a možnostmi jeho implementace do existujícího systému Rendite. Pozastavuje se nad klady a zápory obou variant s cílem vylepšit, urychlit a zjednodušit práci s tímto ERP sytémem.

Klíčová slova

Datový sklad, OLAP, dolování dat.

Summary

This thesis deals with an analysis of choosed BI solution and its implementation into Rendite system. Those steps should lead to improvement and easier and more powerful work with this ERP system.

Keywords

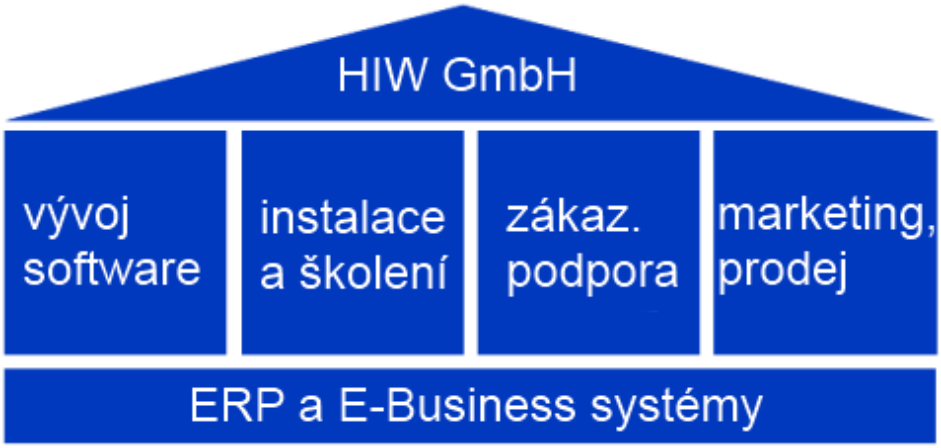
Data warehouse, OLAP, data mining.

Obsah

1 Úvod do problematiky a cíl práce

1.1 Představení společnosti Český software s.r.o.

Český software s.r.o. je dceřinou společností H.I.W. Gesellschaft für Warenwirtschaftssysteme mbH od roku 2006. Č.S. a H.I.W. (dále jen firma) se zabývají zejména vývojem a dodávkami modulárních ERP H.I.W. systémů Rendite (viz Ilustrace 1). Je jedním z předních poskytovatelů ERP pro prodejce techniky v Německu.



Ilustrace 1: oblast působnosti H.I.W. GmbH [2]

Rendite bylo původně určeno především pro prodejce techniky, avšak časem dospělo v univerzální řešení. Zahrnuje moduly pro účetnictví, nákup, příjem, odbyt zboží, kontakt se zákazníky, předpovídání prodejních trendů, sklad, logistiku, servis, manažerskou analýzu dat, plánování důležitých strategických kroků, pokladní systémy, opravnu a servis, interní organizace, mobilní komunikace, výměna dat a další [1].

Firma se dále zabývá těmito oblastmi:

- Podpora (HIW Helpdesk)
- Řízení projektů (HIW Project)
- Intranet (HIW Intranet)
- Internetové stránky (HIW CMS)
- Zákaznická a dodavatelská řešení
- E-Shop

1.2 Motivace a trendy v BI

S přibývajícími roky dochází k čím dál většímu nárůstu shromažďovaných dat. Kromě toho se data sbírají tam, kde se předtím nesbírala. To umožňuje mimo jiného zejména dostupnost dostatečných úložných kapacit. Výhodu, informace a znalosti pak získává ten, kdo dokáže s těmito daty pracovat a efektivně je vyhodnocovat. Termín, který zaštiťuje tuto oblast přístupu k datům se nazývá Business Intelligence (BI).

> „Business Intelligence můžeme chápat jako ucelený a efektivní přístup k práci s firemními daty, který má vliv na správnost strategických rozhodnutí, a tím i na obchodní úspěch společnosti. V současném vysoce konkurenčním prostředí představuje informovanost jednu z hlavních konkurenčních výhod. Tato výhoda spočívá ve schopnosti efektivně využít data nashromážděná ve firmách k tvorbě informací a znalostí, na základě kterých můžeme reagovat na rychle se měnící požadavky trhu a našich zákazníků.“[3]

Využití BI se plynule posouvá od velkých do středních a malých firem. Dopomáhá k tomu pravděpodobně také fakt, že aplikovat a využívat BI mohou uživatelé bez statistických či jiných vědomostí a tyto pak více či méně nahrazuje použitý nástroj. Tendence je poskytnout výstupy, a tedy vědomosti a fakta, nejen vrcholným manažerům firem, ale také zaměstnancům z nižších stupňů hierarchie. Oblastmi působení BI jsou zejména obchodní a finanční instituce, marketing, ale v poslední době také sociální sítě, které se stávají velmi významným zdrojem pro analýzy dat. Ze svého pohledu si myslím, že nepřímý vliv na rozvoj a vylepšení formy zpřístupnění BI má rozvoj mobilních zařízení, zejména pak tabletů.

Lze se domnívat, že BI se bude ubírat cestou poskytnout okamžitě a nejpohodlnější formou co nejpřesnější data k tomu kterému rozhodnutí. Dále se bude ve vyšší míře aplikovat vliv externích elementů na rozhodnutí a podpora tohoto v současných nástrojích. Nástroje BI by měly co nejvíce zjednodušit opakovatelné činnosti při budování či implementaci BI.

1.3 Cíl práce

Systém Rendite poskytuje řadu nástrojů pro podporu rozhodování. Cílem této práce je zanalyzovat co tyto nástroje umožňují, jaké jsou jejich vlastnosti, přednosti a zápory a zjistit, co by znamenalo aplikovat místo nich nebo současně s nimi části vybraného řešení BI. To znamená odpovědět na otázku, zda je nutné nebo výhodné využít multidimenzionální databázi, jak přínosné je dolování dat a jaké výstupy lze díky řešení BI získat oproti současným.

2 Analýza současného stavu

2.1 Architektura Rendite

Hlavní vývojovou větví je desktopová aplikace, která komunikuje přímo s databází. V poslední době se staví na architektuře zaměřené na služby. Služeb využívá tenký klient v podobě webového prohlížeče nebo mobilního zařízení. Oba typy klientů komunikují s webovým serverem pomocí služeb (xml, json).

2.1.1 Klient

- Desktopová aplikace – Přístup ke všem modulům. (Windows XP a vyšší.)
- Internetový prohlížeč – Přístup pouze k některým modulům (kalendář,zprávy,statistiky apod.). Konkrétními aplikacemi, které využívají služby napojené na databázi Rendite, jsou zejména intranet a elektronický obchod.
- Mobilní zařízení – Přístup k vybraným modulům. (iOS, Android)

2.1.2 Zdroje dat

- Rendite DB – Primární zdrojová databáze. Přes 1000 tabulek. Podpora MSSQL 2000 a vyšší.
- Externí databáze – Zejména práce s externí databází Rendite při analytických operacích.
- Datové soubory – Importovací moduly umožňují práci se soubory třetích stran (zejména formát XML).

2.2 Analytický manažer

První z popisovaných analytických nástrojů obsahuje mimo jiné následující oblasti předdefinovaných analýz:

- Nákup
- Prodej
- Finance
- Produkty a sklady
- Analýza zákazníků

Obergruppe	Fabrikat	Artikel	VK-Brutto (Liste)	Verkau Menge Januar
Rechner	Apple	iMac 21,5" i3 3,06GHz 4GB MC508	1.099,00 €	
	Datalogic	FPS18 Netzteil für Ladestation	47,60 €	
		Netzkabel mit Kaltgerätestecker	3,50 €	
		Single Cradle USB/RS232	178,50 €	
		Skorpio 701-902-455 LA RF BT	1.307,81 €	
		Skorpio Lilon Akku	59,50 €	
		Skorpio Multi Battery Charger	208,25 €	
	Fujitsu-Siemens	Esprimo P2560 DC E5500	452,20 €	
		Esprimo P2560 DC E5700	404,60 €	
		Esprimo P2560 DC E5800	416,50 €	
Peripherie	APC	Ersatzakku RBC48 f. Smart UPS 750VA	107,10 €	
		Smart UPS 1000iNET	416,50 €	
		Smart UPS 1500 VA (Powershut-SW)	565,25 €	
		Smart UPS 750iNET LCD (Powershut-SW)	297,50 €	
	Elo	Touchscreen 17" 1729L 43cmAccuT. USB dark	708,05 €	
		Touchscreen 19" 1928L 48cmAccuT. USB dark	844,90 €	

Ilustrace 2: Ukázka výsledku analýzy v Rendite

Tyto kategorie se dále člení na jednotlivé analýzy, které kopírují pro uživatele potřebné analýzy. Jedná se čistě o rozdělení do tématických skupin, jak naznačuje Ilustrace 3.V případě vytvoření nové analýzy nebo modifikace se postupuje tak, že si uživatel vybere jednu nebo více oblastí, v kterých se pak vyskytují konkrétní sloupce s daty(fakta).Dalším krokem je vložení kritérii do osy X nebo Y, podle kterých se budou hledané hodnoty seskupovat či třídit(dimenze). Případně je k dispozici možnost filtrovat hodnoty. Pro prohlédnutí výsledku je potřeba stisknout tlačítko „spočítat“ (Berechnen).Výsledkem (viz Ilustrace 2) je tabulka s barevným rozlišením jednotlivých skupin na každé úrovni hierarchie nebo graf. Na pozadí celého předchozího procesu se odehrávají následující kroky:

- Výběr oblasti, seznam možných faktů a seznam dimenzí je napevno definován ve zdrojovém kódu. Vše s potřebnými parametry a definicemi. Jako např. zdrojová tabulka, zda je možno filtrovat, připojené tabulky atd.
- Sestavení a vykonání SQL dotazu.
- Složení a vygenerování výsledné tabulky nebo grafu.
- Pokud se změní pouze filtr, nemusí se znovu sestavovat, pouze se upraví výsledná tabulka.

Ukázka definice oblasti Einkaufrabatt ve zdrojovém kódu:

```
(name: 'Einkaufrabatt'; // název srozumitelný uživateli
zeitmodus: zmPerioden; //způsob zadávání časové dimenze
```

```
erlaubtezeiteditors: [ezePerioden, ezeBuchungsperioden]; //způsob zadávání časové
dimenze
querystart: 'einkauffil ef'; // základní tabulka s fakty
queryend: ':INPERIOD(ef.lidatum)'; // časová dimenze v tabulce faktů
joins: @EinkaufEinkaufRabattJoins; // seznam doplňujících tabulek a způsobů jejich
napojení (fakta i dimenze)
joincount: length(EinkaufEinkaufRabattJoins); // jejich počet
defaulttablesneeded: [EinkaufRabattT, EinkaufFilT]; // doplňující tabulky k tabulce
základní
hierarchiedefs: @EinkaufRabattHierarchieDefs; // popis napojení na uživatelské
datové hierarchie
hierarchiecount: length(EinkaufRabattHierarchieDefs); // počet popisu napojení
hierarchieSQLMode: hsmJoinLimited; // způsob napojení na uživatelské hierarchie
// přídavná pole pro detailní náhled
detailadditfields: nil;
detailadditfieldcount: 0;
detailtablesneeded: [];
// přídavné filtry (jiné, než fakta nebo dimenze)
existfilters: nil;
existfiltercount: 0;)
```

2.3 Skupinové analýzy (Gruppenanalysis)

Skupinové analýzy poskytují takové přehledy, které lze v základní formě dosáhnout i pomocí analytického manažeru. Nabízejí však vyšší uživatelský komfort týkající se ovládání, větší záběr sledovaných dat a lepší přehlednost.

Kurzfristige Auswertung	Přehled prodejů za poslední čtvrtletí, přehled stavu skladu na jednotlivých pobočkách.
Periodenvergleich	Porovnání prodejů (zisků, stavů na skladě, atd.) s předchozím rokem.
Fabrikatsauswertung	Přehled prodejů pro zvolené období podle jednotlivých výrobců.
Verkäuferauswertung	Přehled prodejů pro zvolené období podle jednotlivých prodavačů.
Filialauswertung	Roční přehled prodejů v jednotlivých pobočkách.
Planvergleich	Porovnání skutečných prodejů se zvoleným plánem.
Frequenzanalyse	Přehled prodejů v jednotlivých dnech v týdnu (nebo hodinách).
Umsatzübersicht	Porovnání prodejů s předchozím rokem na úrovni dnů.
PLZ-Analyse	Přehled prodejů podle místa bydliště zákazníka.

Artikelranking	Vyhodnocení nejúspěšnějších výrobků podle zisku, počtu prodaných kusů, nebo obratu.
Verkäuferprovisionierung	Přehled vyplácených provizí pro prodavače a simulace různých provizních modelů.
Gewählte Altersstrukt.	Přehled stáří výrobků na skladě.
Untergeordnete Altersstrukt.	Přehled stáří výrobků na skladě.

2.4 Podpora plánování – Unternehmensplanung

Uživatel Rendite má možnost naplánovat si veličiny týkající se prodeje v následujícím období a odráží se přitom od prodejů z předchozích let. Na základě sezónních křivek(statistika prodeje) za předchozí roky je zobrazena tabulka, ve které jsou hlavní kategorie a podkategorie produktů (viz. Ilustrace 4) s hodnotami prodejů za jednotlivé měsíce. Jednotlivým křivkám lze přiřazovat prioritu. Uživatel pak může jednotlivé hodnoty upravovat na základě svých rozhodnutí (např. předpoklad zvýšení prodeje v měsíci na základě předchozí reklamní kampaně) a je schopen získat celkovou předpokládanou hodnotu na konci zvoleného období. Plánování probíhá v několika krocích, v nichž je plán postupně upřesňován a dolaďován. Posléze je možné plán uložit a v běhu času plán editovat či porovnáván s aktuálními hodnotami prodejů.

Geschäftsbereich	Obergruppe	March	April	May
Datenimport				
	FERNSEHER	47.16%	0.00%	
Fremdware				
	Fremdware	25.31%	22.64%	

Ilustrace 4: Plánování prodejů.

3 Použitelná řešení a jejich analýza

3.1 Výběr BI technologie – Microsoft

Ze všech možných firem, která poskytují BI řešení (Oracle, IBM, Microstrategy, atd.) jsem z níže uvedených důvodů vybral řešení od firmy Microsoft:

- Microsoft je jedním z předních poskytovatelů BI řešení a poskytuje kompletní portfolio podpůrných nástrojů při aplikaci BI.
- Český software s.r.o. je Microsoft Gold Certified Partner a z toho vyplývají další body uvedené níže.
- Rendite staví na MS SQL.
- Uživatelé napříč celé hierarchie společnosti používají Microsoft Excel a mají minimálně obecné znalosti týkající se ovládání dalších programů od společnosti Microsoft. To se týká i většiny zákazníků.

K dispozici jsou pak pod jednou „střechou“:

- Databázový stroj (Microsoft SQL Server)
- Analyzační služby (SSAS)
- Integrační služby (SSIS)
- Reportovací služby (SSRS)

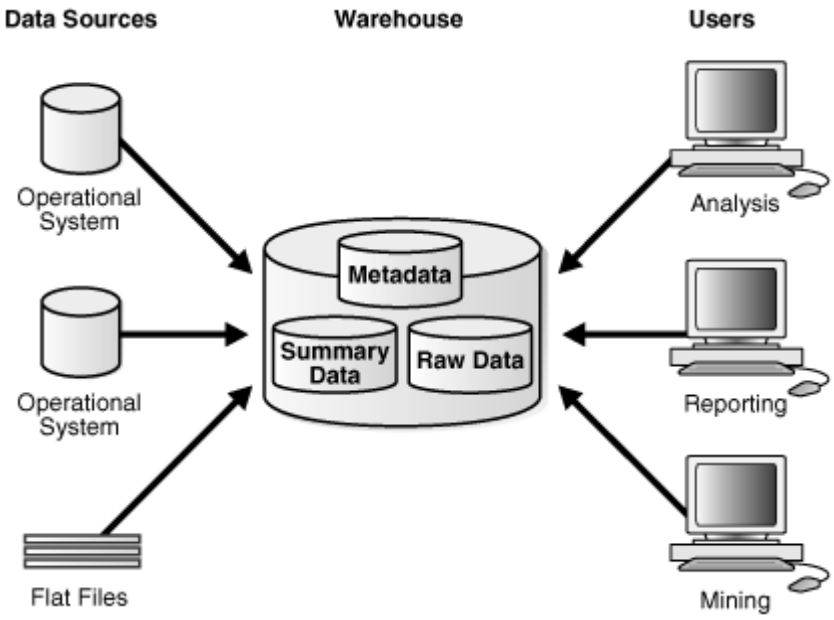
3.2 Datový sklad

3.2.1 Definice a architektura

Prvním krokem při implementaci BI řešení bývá datový sklad. Píšu záměrně bývá, neboť analytické databáze lze vybudovat i nad databázemi OLTP. Nad výhodami a nevýhodami se pozastavím později. Existuje mnoho definic na to co to je datový sklad. Vezměme definici Billa Inmona, kterému je autorství tohoto pojmu připisováno[4]:

> „Datový sklad je podnikově strukturovaný depozitář subjektově orientovaných, integrovaných, časově proměnlivých, historických dat použitých na získávání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.“

Architektura datového skladu se skládá z úložiště, do kterého se zapisují údaje nejen z produkčních databází, ale také z dalších zdrojů, jak je znázorněno na obrázku Ilustrace 5.



Ilustrace 5: Schéma datového skladu. [17]

Datové pumpy dle kritérií transformují data z produkčních databází do datového skladu. K práci multidimenzionálních databází slouží OLAP server. Uživatelé jsou pak schopni za pomoci konkrétních nástrojů získávat různé výstupy.

3.2.2 Multidimenzionální modelování a jeho entity

Předmětem je schopnost pracovat s vícerozměrnou datovou krychlí a poskytnout uživateli náhled na obsažená data dle různých, právě vyžadovaných pohledů.

Základními stavebními kameny jsou[22]:

- Fakta - Prvky krychle. Hodnoty, které lze agregovat či seskupovat dle dimenzí.. Jedná se např. o hodnoty prodejů, stav na skladě apod.
- Dimenze - Rozměry krychle. Např. kategorie, datum.

3.2.3 OLTP a OLAP

Podrobnému popisu co to je OLTP a OLAP, a jaké jsou hlavní rozdíly mezi relační a multidimenzionální databází se již dříve ve svých pracích podrobně věnovali moji bývalí kolegové[9][10], ale protože význam těchto termínů není jednoznačný, ve stručnosti zmíním

interpretaci těchto zkratek a jak budu na tyto termíny pohlížet.

•OLTP (Online Transaction Processing) – Transakční databáze, označovaná také jako databáze produkční. Využívá relační schéma.

•OLAP (Online Analytical Processing) – Termín představuje data s odlišnou strukturou než ve výše uvedeném případě, ale zároveň analytické nástroje. Touto zkratkou budu označovat zejména analytické multidimenzionální databáze.

Když jsem se s datovým skladem seznamoval blíže, hlavně s jeho realizací, nebylo mi dostatečně jasné, zda datový sklad využívá transakční databázi nebo (i) OLAP. V několika zdrojích[5][4] jsem se setkal s tím, že je srovnáváno OLTP vůči datovému skladu.V jiných zase, že datový sklad neobsahuje tabulky. Při realizaci (uvedené níže) jsem si ověřil, že datový sklad je v případě použití Microsoft řešení OLTP databáze, zatímco analytická databáze může ležet na jiném serveru. Obecné srovnávání OLTP a datového skladu je tedy mírně matoucí. Ve zdrojích jsem se později dozvěděl, že datových skladů je několik typů a různí autoři zahrnují do pojmu datový sklad něco jiného[6]. Pro sjednocení názvů, pokud budu hovořit o datovém skladu, budu mít na mysli shromaždiště dat využívající relační databázové schéma a do pojmu OLAP budu zahrnovat analytické databáze.

Rozdíly OLTP a OLAP [7][4]:

Zdroj dat	Operační data; zdroje dat jsou OLTP databáze.	Konsolidovaná data.
Účel dat	Podpora provozních aplikací.	Podpora plánování, řešení problémů a podpora rozhodování.
Rychlost zpracování	Velmi rychlé.	Záleží na velikosti dat.
Návrh databáze	Normalizované, mnoho tabulek.	Typicky nenormalizované, tématicky vytvářené.
Záloha a obnovení	Operační data jsou kritická pro běh společnosti. Ztráta dat má dalekosáhlé důsledky.	Spíše než záloha se využívá znovu načtení dat ze všech zdrojů.
Stáří dat	Současná.	Historická.
Dotazování	Relativně jednodušší dotazy.	Komplexní dotazy zahrnující agregace.
Základní operace nad daty	Přidávání, změna, mazání, čtení	Čtení.
Velikost	Malá až velká.	Velká až velmi velká.
Původ dat	6 – 18 měsíců.	2 – 7 let.

3.2.4 Struktura OLAP databáze

Dimenze, fakta a kostky se řadí do základní terminologie při práci s analytickými databázemi. Jejich základní popis a operace, které s nimi mohou být prováděny, jsou popsány v pracích mých předchozích kolegů [9][10].

3.2.5 Úložiště multidimenzionálních dat

Služby analytického serveru MSSQL (SSAS) poskytují několik modifikací základních modelů MOLAP, ROLAP a HOLAP, jimiž se také zabývali kolegové [9]. Jednotlivé části analytické databáze pak mohou využívat různé modely[11]:

Real Time ROLAP	OLAP v reálném čase. Data a agregace jsou uložena v relačním formátu. Jakmile dojde ke změnám, server okamžitě aktualizuje(nulová latence). Toto nastavení je obvykle využíváno pro datové zdroje s velmi častými průběžnými změnami, kdy je uživateli vyžadováno mít vždy aktuální data.
Real Time HOLAP	OLAP v reálném čase. Data jsou uložena v relačním formátu, zatímco agregace v multidimenzionálním. Nepoužívá se žádná MOLAP vyrovnávací paměť. Nastavení podobně jako v prvním případě; pro ne tak časté aktualizace.
Low Latency MOLAP	Přepínání mezi ROLAP a MOLAP(objekty využívají vyrovnávací paměť)
Medium Latency MOLAP	Data i agregace uloženy multidimenzionálním formátu.
Automatic MOLAP	Podobně jako výše uvedený model. Typické využití pakliže má rychlost provádění dotazů klíčovou důležitost.
Scheduled MOLAP	Každých 24 hodin se aplikují změny.
MOLAP	Vyrovnávací paměť se nevyužívá. Aplikace změn manuálně nebo naplánovaně.

Pohledy na datové zdroje (Data Source Views)

V případě BI technologie Microsoft se využívá pohledu na datové zdroje jako logické vrstvy, která je abstrahující vrstvou nad jednou nebo více fyzickými databázemi. To přináší výhody, které umožňují vytvářet logická spojení tabulek, definování primárních a cizích klíčů, vytváření nových tabulek v této vrstvě, přidávání počítaných sloupců apod. Další výhodou je možnost při změně fyzické struktury pouze změnit mapování. Této vrstvě není možné se vyhnout a musí být využívána, avšak doporučuje se využívat této vrstvy minimálně [19], využít pouze automatického namapování fyzických tabulek dimenzí a faktů na logické a mít veškerá spojení a definice realizována již na fyzické vrstvě. Hlavně proto, že kdybychom chtěli využít datový

sklad za pomoci jiné BI technologie, museli bychom všechno, co je v logické vrstvě, definovat znovu.

3.2.6 Dotazy v OLAP databázi – MDX

MDX (Multidimensional Expressions) je dotazovací jazyk vytvořeným za účelem práce s daty z multidimenizonálních modelů. Syntakticky se podobá SQL jazyku. Vyskytují se zde stejná klíčová slova SELECT, FROM, WHERE, avšak principiálně se od jazyka SQL – sloužícímu pro operace nad dvou-dimenzionálním modelem – liší. Jako ilustrace slouží následující příklad:

```
SELECT {
      Measures.[internet sales amount] } ON COLUMNS, --sloupce
      non empty --odstranění řádků s Internet Sales Amount = null
      { filter(
            [date].[calendar].[calendar year]
            ,[internet sales amount] > 4000000) -- odfiltrování
      } -- řádky
ON ROWS
FROM [Adventure Works]
```

Dotaz vrátí výsledek z cvičné databáze Microsoft Adventure Works DW 2008 R2 [21]. Z krychle Internet.SalesAmount jsou získány prodeje(fakta) vyšší než zadané hodnotě za období 2006-2008(dimenze) :

	Internet Sales Amount
CY 2006	$6,530,343.53
CY 2007	$9,791,060.30
CY 2008	$9,770,899.74

3.2.7 Způsoby budování datového skladu

Nad způsobem budováním datového skladu, má-li být datovým „srdcem“ firmy, je potřeba se velice dobře zamyslet. Otázka zní, zda využít datových skladů nebo datových tržišť [10], a jak při jejich implementaci postupovat. V literatuře[12] je zmiňován termín „velký třesk“, jež se dá přirovnat k vývoji software a termínu vodopádového modelu. Nevýhody tohoto modelu jsou zřejmé a kromě několika málo typů projektů tento způsob nelze než nedoporučit. Protipólem je přírůstková metoda, která je rozdělena na dva typy[12]:

- „Shora dolů“ – Na základě požadavků uživatelů, s přihlédnutím na hierarchie předmětných

oblastí, se vytvoří konceptuální model datového skladu, kde se postupně vytvářejí datové trhy, tedy datové sklady předmětných oblastí v rámci datového skladu. Jako nevýhoda je uváděna vlastnost zvýšených vstupních nákladů, aniž bychom znali návratnost investic.

• „Zdola nahoru“ – Konceptuální model se vytváří na základě zdrojových dat, hlavní roli zde hraje IT oddělení. Tento model se považuje za nevýhodný, neboť IT oddělení nemusí mít povědomí o tom, co vlastně uživatel potřebuje a vyžaduje.

Přírůstková metoda zahrnuje iterační kroky (strategie, definice, analýza, návrh, sestavení, produkce), které se následně opakují – opět víceméně totožné s iteračními kroky při použití některých postupů vývoje software.

Mírně odlišným pohledem rozdělení datových skladů (datových tržišť) je následující[13]:

• Datový sklad jako množina datových tržišť.

  - S celopodnikovými dimenzemi, fakty (Ralph Kimball).

  - Bez celopodnikových dimenzí, faktů.

• Celopodnikový datový sklad (Bill Inmon).

Obecně lze říci, že budování jednotlivých datových tržišť je výhodné v rychlosti implementace. Je však potřeba si dát pozor na použití stejných (celopodnikových) dimenzí a faktů. Při vytváření datových tržišť není možné vyhnout se redundancím.

3.2.8 ETL – Plnění datového skladu

Proces ETL je mechanismus, který zahrnuje přenos dat ze zdrojových systémů do systémů cílových (datový sklad). Během tohoto přenosu většinou dochází k úpravě těchto dat. Microsoft pro tento účel poskytuje integrační služby SSIS.

Extrakce

Prvním krokem ETL procesu je schopnost získat požadovaná data ze systémů různých formátů fungujících v různých prostředích na různých platformách. Může se jednat nejen o data podniková, ale i o externí. Laicky řečeno je potřeba získat cokoli odkudkoli.

Transformace

Nepřímým pozitivním efektem implementace BI bývá ve většině případů vyčištění dat, která

chceme analyzovat. V operačních databázích se chtě nechtě vyskytuje několik problémů týkajících se kvality dat[12]:

- Nejednoznačnost. Jednou je např. pravdivostní hodnota označována jako „ano/ne“, jednou jako „0/1“.
- Chybějící položky.
- Duplicita.
- Různé názvy stejných objektů.
- Odlišné měny.
- Odlišné formáty čísel a textových řetězců.
- Problémy s referenční integritou.
- Chybějící datum nebo chybný formát.

Cílem této fáze je tyto problémy eliminovat a poskytnout datovému skladu kvalitní vyčištěná data. Transformace v sobě zahrnuje nejen operace s existujícími daty, ale je také možné data generovat (např. spouštění SQL skriptu pro generování časové dimenze).

Nahrávání

Finální fáze procesu ETL. Transformovaná data se fyzicky přesunou do cílového úložiště. Během procesu ETL nejsou primární a cizí klíče brány v potaz, proto je potřeba nově vytvářeným tabulkám nastavit pro jednoznačnou identifikaci klíče nové. Stejně tak je možné nastavit indexování.

3.2.9 ETL – Vlastnosti MSSQL 2008

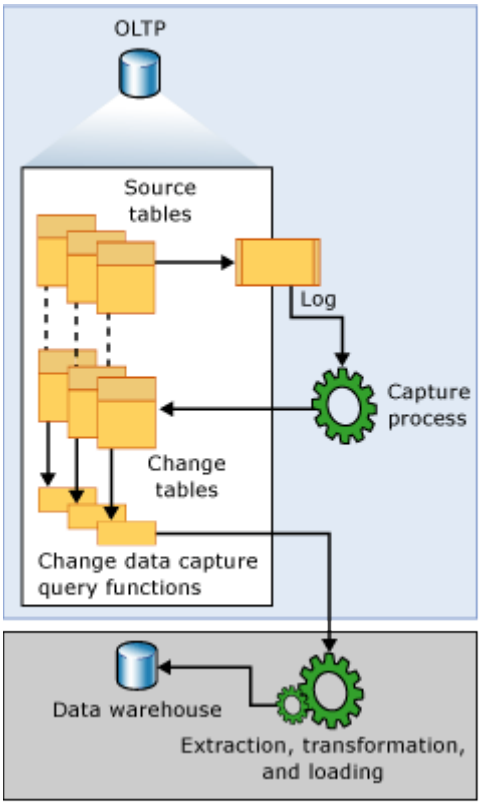
Change Tracking a Change Data Capture

U běžných projektů dochází při prvotním naplnění skladu k přesunu velkého množství dat, a pak se už data – zejména u objemných tabulek – přesouvají v pravidelných intervalech v řádu hodin nebo dní. Pro zachytávání změn v tabulkách byly do SQL Serveru 2008 přidány dvě funkcionality:

- Change Tracking – Synchronní zapisování změn pomocí triggerů.
- Change Data Capture (CDC) – Asynchronní analýza transakčního logu.

Porovnání obou funkcí [12]:

Vlastnost	Change Tracking	Change Data Capture
Mechanismus	Synchronní(trigger)	Asynchronní(trans.log)
Ukládá změněné údaje	Ne, jen primární klíče	Ano
Ukládá metadata o změnách	Ano	Ano
Sledování po sloupcích	Ano	Ano
Filtrování podle verze	Ano	Ano
Sledování DDL příkazů	Ne	Ano
Automatické „uklízení“	Ano	Ano
Edice	Všechny	Enterprise, Developer, Evaluation



Ilustrace 6: Princip Change Data Capture.[16]

Princip fungování průběžného zachytávání změn (CDC) popisuje obrázek Ilustrace 6. Nejprve je potřeba zapnout CDC pro konkrétní databázi a tabulky pomocí systémových procedur sp_cdc_enable_db a sp_cdc_enable_table za využití SQL Server Agenta pro účely zachytávání změn. To se týka mimo jiné transakčního logu, do kterého se promítnou změny provedené ve zdrojových tabulkách. Příkladem bude realizace zachytávání změn tabulky FactInternetSales v databázi Adventure Works 2008 R2. Procedura sp_cdc_enable_table vyžaduje následující

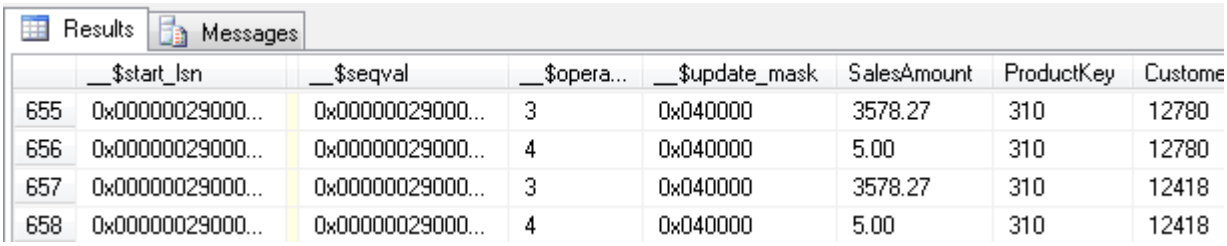
parametry:

```
EXEC sys.sp_cdc_enable_table
@source_schema = N'dbo',
@source_name = N'FactInternetSales',
@capture_instance = N'dbo_factisales',
@role_name=N'public',
@supports_net_changes = 1;
--@captured_column_list='' V případě zachycení změn pouze v některých sloupcích je
možno definovat v tomto parametru, avšak sloupce jako primární klíče musí být také
uvedeny.
```

Z tabulky sys.tables lze vyčíst, že tabulka pro sledování změn má název dbo_factisales_ct. Provedeme nějaké změny dat v tabulce FactInternet Sales:

```
update FactInternetSales set SalesAmount=5 where ProductKey=310
```

Ilustrace 7 ukazuje obsah tabulky dbo_factisales_ct, přičemž jednotlivé parametry znamenají


Results Messages

	__$start_lsn	__$seqval	__$opera...	__$update_mask	SalesAmount	ProductKey	Custome
655	0x00000029000...	0x00000029000...	3	0x040000	3578.27	310	12780
656	0x00000029000...	0x00000029000...	4	0x040000	5.00	310	12780
657	0x00000029000...	0x00000029000...	3	0x040000	3578.27	310	12418
658	0x00000029000...	0x00000029000...	4	0x040000	5.00	310	12418

Ilustrace 7: Obsah dbo_factsales_ct.

následující :

- __$start_lsn – pořadové číslo v sekvenčím logu. Tento údaj lze pomocí funkce převést na datum. Lze tedy celkem jednodušše zjistit datum a čas poslední změny:

```
select sys.fn_cdc_map_lsn_to_time(sys.fn_cdc_get_max_lsn())
```

- __$end_lsn – není podporováno, je vždy NULL v SQL Serveru 2008
- __$update_mask – bitová maska ukazuje, které sloupce byly změněny v DML operaci
- __$seqval – číslou složící k seřazení operací v rámci transakce
- __$operation – určuje typ DML operace

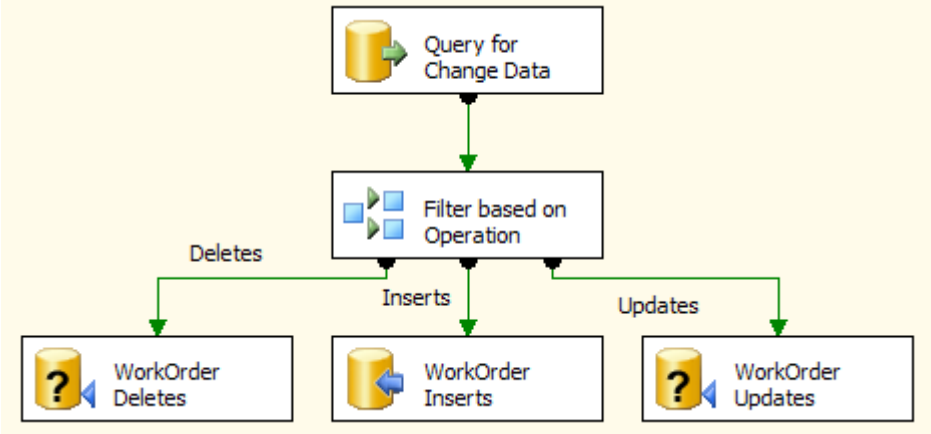
  1 = smazání (delete)

  2 = vložení (insert)

3 = změna (update(old)), hodnota před provedením

4 = změna (update(new)), hodnota po provedení

Nastavení CDC v SQL Serveru 2008 nepodporují vizualizační nástroje a je potřeba provádět je pomocí SQL dotazů a vestavěných procedur a funkcí. Realizovat CDC za pomoci SSIS je možné tak, jak je to uvedeno v příkladu CDC Adventure Works 2008 R2. Schéma, během něhož dochází ke zjištění změn a jejich aplikace na datový sklad je uvedeno na obrázku Ilustrace 8.V prvním kroku dochází ke zjištění všech záznamů, u kterých došlo ve vybraném časovém intervalu ke změně. Dle operací (delete,insert,update(n)) jsou pak prováděny úpravy na datovém skladu.



Ilustrace 8: Schéma CDC v SSIS.

Merge

Funkce Merge, dostupná od verze SQL Serveru 2008, slouží k synchronizace tabulek. V jedné transakci je možné přidávat, mazat nebo měnit záznamy jedné tabulky na základě jiné. Následující příklad ilustruje použití funkce merge. První tabulka produkty slouží jako zdrojová a produktyKopie jako cílová, která se bude synchronizovat. Záměrně obsahuje méně sloupců pro ověření toho, že tabulky nemusí být identické.

```
create table produkty(
id int primary key,
nazev varchar(max),
vyrobce varchar(max),
barva varchar(200),
cena float
)
create table produktyKopie(
id int primary key,
```

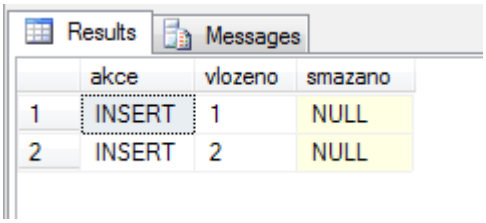
```
nazev varchar(max),
barva varchar(200),
cena float
)

insert into produkty(id,nazev,vyrobce,barva,cena)
values(1,'Video','Sony','černá',5),(2,'Televize','LG','modrá',1)
```

Tabulky nesplňují kritéria vytváření tabulek v transakční databázi, budou sloužit pouze jako názorný příklad. Jakmile je nalezen odpovídající záznam dle id, provede akce dle definovaných pravidel.

```
merge produktyKopie as pk
using produkty as p
on pk.id = p.id
when matched and (p.nazev!=pk.nazev or p.barva!=pk.barva or p.cena!=pk.cena)then
update set pk.nazev = p.nazev, pk.barva = p.barva, pk.cena=p.cena
when not matched then
insert values(p.id,p.nazev,p.barva,p.cena)
when not matched by source then
delete
output $action as akce,inserted.id as vlozeno,deleted.id as smazano;
```

Výstup funkce output je na obrázku Ilustrace 9. Požadované záznamy z tabulky produkty byly vytvořeny v tabulce produktyKopie.



Ilustrace 9: Výstup funkce output.

3.3 Dolování dat

3.3.1 Definice

Jedním z dalších prostředků spadajících do BI technologií a podporujících rozhodování je dolování dat (data mining).

> „Data mining je prostředek pro získávání informací pro podporu rozhodování. Samotné rozhodování musí udělat příslušný zodpovědný pracovník“[12]

> „Data mining je netriviální proces zjišťování platných, neznámých, potenciálně užitečných a snadno pochopitelných závislostí v datech.“[12]

„Data mining je proces analýzy dat z různých perspektiv a jejich přeměna na užitečné informace. Z matematického a statistického hlediska jde o hledání korelací, tedy vzájemných vztahů nebo vzorů v datech. Data mining je proces, jehož cílem je těžba informací v databázích. Využívá statistické metody a další metody hraničící s oblastí umělé inteligence.“[12]

Dolování dat je relativně nová disciplína. Metodologie použité v této disciplíně pochází ve většině případů ze dvou vědních oborů, a těmi je strojové učení a statistika. Oblast využití je v podstatě neomezená, mezi nejčastější patří [15]:

- Bankovní sektor – zacílení na konkrétní zákazníky, ziskovost zákazníka
- Obchod – segmentace zákazníků, nabídka relevantních produktů, odhalení specifických typů zákazníků
- Pojišťovnictví – odhalení pojistných podvodů
- Zdravotnictví – odhalení chorob
- Veřejný sektor – daňové podvody, anomálie, kriminální zločiny
- Výroba – analýzy spojené se zárukou, spolehlivostí, výnosy
- Telekomunikace – neoprávněné vstupování do sítě

3.3.2 DM versus OLAP

Dolování dat nutně nevyžaduje využití OLAP, avšak tyto dvě technologie se mohou vzájemně užitečně doplňovat. Lze využít vyčištěná data z kostek, což odfiltruje nepodstatná data a usnadní či urychlí jejich pochopení. Výsledky dolování dat pak lze prezentovat v OLAP kostkách. Rozdíly mezi DM a OLAP nejlépe ozřejmí konkrétní otázky:

Dotazy směřující k využití OLAP:

- Jak se liší počet nehod kuřáků a nekuřáků?
- Jaký je průměrný objem nákupu s falešnou kreditní kartou a nefalešnou?

Dotazy směřující k využití DM:

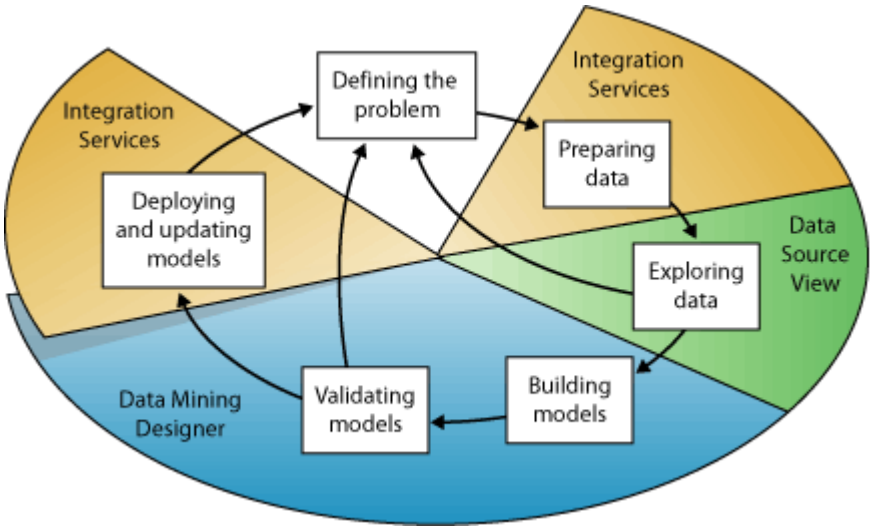
- Jaké vzory nakupování jsou spojené s falešnou kreditní kartou?
- Opustí X společnost?
- Co způsobuje nehody?

- Zákazník Y si koupil televizi, koupí si i video?

3.3.3 Proces DM

Proces dolování dat bývá obvykle složen z několika kroků, které nemusí jít nezbytně postupně za sebou, ale mohou se vzájemně prolínat a probíhají v iteracích jak ukazuje obrázek Ilustrace 10:

- Definice problému
- Příprava dat
- Prozkoumání dat
- Vytvoření modelů
- Prozkoumání a ověření modelů
- Nasazení a úprava modelů



Ilustrace 10: Proces DM.[17]

Definice problému

Definice problému a zvážení všech možných variant jak docílit odpovědi na něj je jedna z nejsložitějších částí procesu. Je třeba velmi dobře rozumět obchodním procesům, znát rozsah celého problému, jaká kritéria se musí vzít v úvahu, v jakých měřítkách bude výstup celé analýzy probíhat. Je třeba nezapomenout na důležité faktory, které mohou výsledek analýzy znehodnotit, pokud by nebyly do procesu zahrnuty.

Příprava dat a ověření

Tato fáze se přímo dotýká téma čistých dat, neboť do analýzy nesmí vstupovat data obsahující

chybějící nebo nesprávné údaje. Proto je vhodné použít existující datový sklad jako zdroj. Kromě kvality dat z pohledu datového skladu je třeba zvolit data nejpřesnější, kde se nevyskytují nějaké skryté a nežádoucí korelace. Jakékoli nesrovnalosti v datech mohou negativně ovlivnit výsledek. Je více než vhodné aplikovaná data znát a vyhnout se tomu, že v některých úsecích nedávají smysl.

Vytvoření modelu

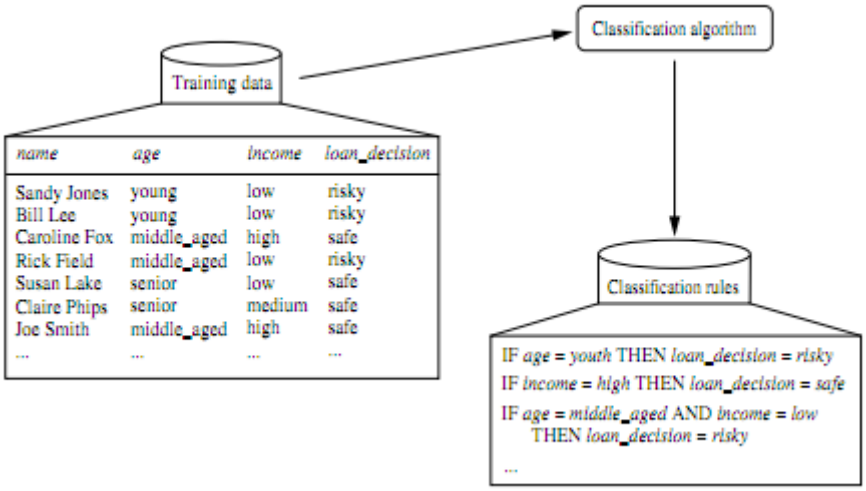
Je potřeba vytvořit model se zvoleným algoritmem. V tomto kroku se definují vstupní, klíčové a predikční atributy. Volí se množina testovacích a trénovacích dat, aby bylo možné ověřit vhodnost algoritmu pro danou úlohu a správné navržení modelu. K dispozici je řada nástrojů a výstupů, které pomohou sestavený model analyzovat.

Predikce

V tomto kroku lze aplikovat ověřený model na nová data, ve kterých chceme hledat vzory.

3.3.4 Oblast a výběr algoritmu

Ještě než budu popisovat algoritmy DM, které se dají použít k vytvoření DM modelu, chtěl bych jednotlivé algoritmy kategorizovat a uvést několik případů, podle kterých je možné určit, která kategorie a potažmo algoritmus, je vhodný nebo nutný k řešení té které úlohy. Výběr správného algoritmu je obtížný úkol a to nejen proto, že pro danou úlohu je možné vybrat více algoritmů, z nichž jeden může být výhodnější oproti jinému, ale oba mohou být použitelné. Oracle[16] rozlišuje DM algoritmy a funkce. Funkce dělí na řízené (prediktivní modely) a neřízené (detekce vzorů). Microsoft uvádí termín typy algoritmů a tyto typy dále nekategorizuje. Pro sjednocení názvů budu místo typů nebo funkcí používat termín oblast[12]. Zaměřím-li se na konkrétní oblasti, pak je charakteristika více méně totožná. U každé oblasti zmíním několik případů či otázek, které do konkrétní kategorie spadají, a jejichž smyslem je výběr algoritmu usnadnit.



Ilustrace 11: Analýza trénovacích dat.[20]

Klasifikace(prediktivní model)

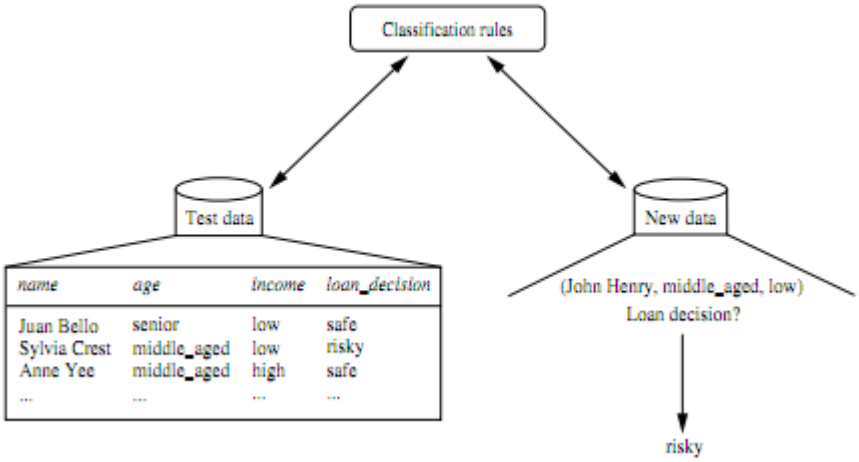
Predikuje jednu nebo více diskrétních hodnot na základě jiných atributů v datech[16]. Tento proces se dělí na dvě fáze:

- Učící – Klasifikační algoritmus analyzuje data a získá z nich pravidla.
- Kvalifikační – Získaná pravidla se aplikují na nová data.

Ilustrace 11 představuje analýzu trénovacích dat klasifikačním algoritmem, jejíž výsledkem jsou klasifikační pravidla. Klasifikační pravidla jsou posléze aplikována na testovací data. Pokud jsou výsledky klasifikace akceptovatelné, použijí se pravidla na klasifikaci nových dat jako je na obrázku Ilustrace 12.

Regrese (prediktivní model)

Predikuje jednu nebo více spojitých hodnot na základě jiných atributů v datech[16]. Od klasifikace se liší tím, že jde o numerickou predikci. Předmětem je závislost náhodné veličiny na nenáhodné proměnné. Využívá se aproximace metodou nejmenších čtverců. Pakliže se jedná o logickou regresi (závislá proměnná je diskrétní) je potřeba přetransformovat ji na regresi lineární. Příkladem použití může být predikce zisku, ztrát, prodeje, teploty atd.



Ilustrace 12: Klasifikace nových dat. [20]

Segmentace

Rozděluje data do skupin v nichž mají jednotlivé prvky podobné vlastnosti.

Asociace

Tato oblast sleduje pravděpodobnost spolu výskytu některých prvků v množině. Algoritmus nalezne asociace mezi hodnotami a posléze je potřeba rozhodnout o jejich relevanci. Vztahy mezi těmito prvky jsou pak popsány pomocí asociačních pravidel. Jedná se nejčastěji o obchodní otázky, vyplývající ze znalosti toho, které produkty byly nakoupeny společně, anebo o koupěschopnost klientů. Konkrétně to znamená, že přestože jsou citrony umístěny v sekci zelenina a ovoce, mnohdy jsou k nalezení vedle tequily v sekci alkoholických nápojů.

DM algoritmy pro jednotlivé oblasti použití [12]:

Decision trees	Naïve Bayes	Clustering	Seq. Clustering	Time Series	Association Rules	Neural Networks	
nejlepší	vhodný	vhodný	vhodný		vhodný	nejlepší	Classification
vhodný	nejlepší	nejlepší	nejlepší				Regression
		vhodný	vhodný			nejlepší	Segmentation
vhodný	vhodný	nejlepší	nejlepší		vhodný	vhodný	Assoc. Analysis
		vhodný	vhodný			nejlepší	Anomaly Detect.
			vhodný				Seq. Analysis
				vhodný			Time Series

3.3.5 Algoritmy DM

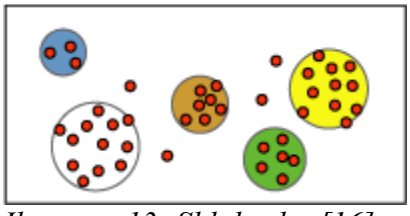
Níže uvedený výčet obsahuje algoritmy, které má SQL Server 2008 implementovány. Co se týká názvosloví, tyto algoritmy jsou označeny názvem firmy Microsoft a nejedná se tedy o konkrétní algoritmy určené pro dolování dat obecně. Např. Microsoft Association Algorithm využívá DM algoritmus Apriori. Pokud je z nějakého důvodu potřeba implementovat algoritmus třetích stran, tedy například místo uvedeného Apriori využít třeba FP-Tree, je to možné přes poskytované rozhraní.

Rozhodovací stromy

Jak už název napovídá, princip tohoto algoritmu (Microsoft Decision Trees) spočívá v realizaci stromu, který posléze poslouží k vytvoření pravidel, potřebným k predikčnímu modelu. Algoritmus podporuje jak spojité, tak diskrétní hodnoty. Nejprve se vybere jeden atribut(vlastnost) jako kořen stromu. Musí to být atribut, který od sebe objekty maximálně odliší. K tomu se využívá míra informační hodnoty atributu[12]. Z kořenového atributu se pak vytvoří větve(hrany), které rozdělí objekty splývající s kořenovým atributem do podmnožin, definovaných dle kritérií. Všechny objekty jsou zařazeny do nějaké podmnožiny. Strom se může dále dělit na další podmnožiny, avšak je potřeba nastavit optimální velikost stromu, aby nebyl ani příliš stručný, ani příliš obsáhlý.

Shlukování

Tento algoritmus (Microsoft Clustering Algorithm) identifikuje vztahy mezi daty a na základě těchto vztahů vytváří jednotlivé shluky, jak je patrné na obrázku Ilustrace 13.



Ilustrace 13: Shluky dat.[16]

Sekvenční shlukování

Hybridní algoritmus (Microsoft Sequence Algorithm), který vyhledává shluky za pomoci algoritmu shlukování a Markovských procesů a modelů.

Asociační pravidla

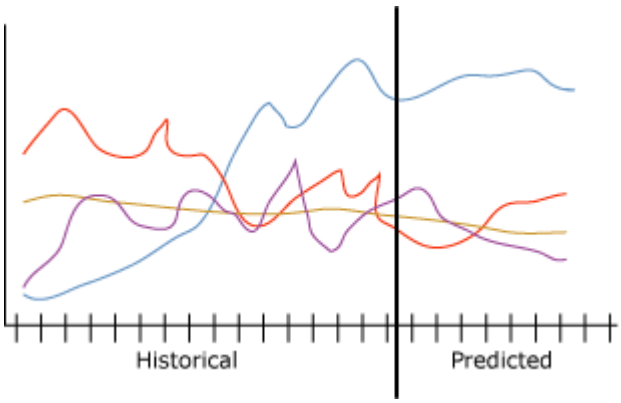
Realizuje hledání vztahů v datech (Microsoft Association Algorithm). Nejčastějším případem je analýza nákupního košíku, kdy se nakupujícímu nabízí výrobky, které jsou k vybíranému nějakým způsobem vztaženy. Charakteristika byla uvedena výše v sekci výběr algoritmu. Výstup algoritmu lze ořezat o nedůležité případy pomocí nastavení atributů algoritmu jako je podpora a spolehlivost.

• podpora (support) – Počet případů, které obsahují cílovou položku. Pakliže je nastavena hodnota minimum_support a této hodnoty není dosaženo, nejsou případy zahrnuty do modelu.

•spolehlivost (probability,confidence) – Hodnota, která udává počet výskytů nějaké kombinace prvků dělená počtem všech případů.

Časové řady

Na základě trénovacích dat lze předpovědět vývoj proměnné v následujícím časovém horizontu (Microsoft Time Series Algorithm) jak naznačuje obrázek Ilustrace 14. Většinou se jedná o předpověď prodeje toho kterého výrobku, vývoj obchodních ukazatelů jako např. obrat, zisk, apod.



Ilustrace 14: Ukázka grafu časové řady. [16]

Neuronové sítě

Neuronová síť představuje podobnost s lidským mozkem – princip rozpoznávání vzorů. Skládá se ze vzájemně spojených prvků (neuronů), které od sebe přijímají podněty. Algoritmus (The Microsoft Neural Network) je užitečný pro analyzování komplexních dat z oblasti výroby nebo obchodu [16]. Dále pro marketingové analýzy, rychle se měnící informace z oblasti financí, text miningu.

Síť je složena ze tří vrstev neuronů[16]:

- Vstupní vrstva
- Skrytá vrstva
- Výstupní vrstva

Zjednodušeně lze říci, že na trénovacích datech upravujeme jednotlivé váhy ve skryté vrstvě tak, abychom dostali co nejpřesnější výstupy a poté aplikujeme neuronovou síť na testovacích datech k ověření výsledků.

Naive Bayes

Rychlý algoritmus (Microsoft Naive Bayes Algorithm) založený na Bayesově větě.

„Dobrým hypotetickým příkladem pro vysvětlení Bayesovy věty je novorozenec, který pozoruje, zda bude v noci venku zima. První den to neumí posoudit, protože to ještě nikdy nezažil, a tedy pravděpodobnost bude 0.5, tedy 50%. Každý další den když nastane noc, se jeho odhad pravděpodobnosti tohoto jevu zpřesňuje, v tomto případě zvyšuje.“[12]

3.3.6 Dotazy v DM – Data Mining Extensions

DMX je jazyk ne nepodobný SQL, který umožňuje vytvářet a modifikovat DM modely a současně data v těchto modelech prohlížet a realizovat predikce. Ukázka použití tohoto jazyka je v kapitole 4.3.

3.4 Výstupy analytických databází

Jako je důležité analýzy realizovat, je neméně důležité tyto analýzy vhodnou formou poskytnout uživatelům. Prezentování těchto faktů a vědomostí je možné kategorizovat několika způsoby. Ať už by se jednalo o vzhled, formu, cílovou skupinu nebo použité technologie. Microsoft dodává SSRS, které více či méně kvalitně podporují prezentaci vytvořených analytických modelů. Vzhledem k tomu, že jsem se zaměřil na BI řešení od firmy Microsoft, nebudu úmyslně zmiňovat nástroje třetích stran. V dalších kapitolách budu využívat následujících služeb a nástrojů:

- SSRS
- Microsoft Excel 2010

- Microsoft SQL Management Studio
- Microsoft Visual Studio

4 Implementace

4.1 Technické požadavky

Datový sklad bude využívat MSSQL 2008 R2 běžící na Microsoft Windows 2003.Transakční databáze byla vytvořena z kopie a poběží na témže serveru. Analytická databáze a predikční modely budou uloženy na lokálním počítači.

4.2 Realizace datového skladu

Na serveru jsem za pomoci MS SQL Manageru vytvořil databázi RenditeDW, která bude sloužit jako datový sklad. Dalším krokem bude popis zdrojových dat a návrh a definice dimenzí a tabulek faktů. Praktická realizace proběhne v sekci ETL následovaná vytvořením OLAP krychlí, pokrývajícími vybrané oblasti. Při popisu dimenzí a tabulek faktů již budu uvádět z jakých tabulek jsou tyto dimenze a fakta vytvořeny, přestože to fakticky patří až do jednotlivých procesů ETL.

4.2.1 Popis zdrojových dat

Jediným zdrojem při plnění datového skladu bude transakční databáze RenditeTest, která je sice testovací databází staršího data, avšak obsahuje relevantní údaje, a to v dostatečném množství. Při realizaci datového skladu jsem se zaměřil na některé z oblastí, které jsou zavedeny v současném analytickém manageru, a které jsou velmi často používány v reálném provozu. Vybral jsem ty, které jsou uvedeny v analytickém manažeru shora, neboť jsou teoreticky nejpoužívanější. Jednotlivé oblasti pokrývá desítky relačně svázaných tabulek. Místo ERA modelu zvolím pro jejich popis seznam vybraných oblastí se stručnou charakteristikou klíčových tabulek, které budou důležité pro tvorbu tabulek dimenzí a faktů.

Oblasti zaměření při implementaci :

- Prodej (Verkauf)
- Produkty a sklady (Artikel & Lager)
- Nákup (Einkauf)
- Faktury (Kundenrechnungen, Lieferantenrechnungen)
- Zákazníci(Kunden)

Jelikož se tabulky ve vybraných oblastech vzájemně překrývají, nebudou dále nijak kategorizovány. Názvy tabulek budu dodržovat původní pro snazší orientaci v ukázkových příkladech, avšak rozlišení velkých či malých písmen nemá v současnosti žádný význam.

- VERKAUF – Prodané výrobky (pozice na faktuře). Cena výrobku, datum, atd.
- VERKAUF_INFO – Popis pozice na faktuře.
- ARTSTAMM – Informace o výrobku.
- FABRIKAT – Výrobci.
- KUNDSTAMM – Zákazníci.
- PERSONAL – Osobní údaje uživatelů.
- GBEREICH – První kategorie produktů, např. Unterhaltungselektronik.
- OBERGRUPPE – Druhá kategorie produktů.
- UNTERGRUPPE – Třetí kategorie produktů.
- gruppen – Čtvrtá kategorie produktů.
- FARBEN – Barvy.
- REGION – Geografické údaje zákazníků.
- FIRMEN – Firmy.
- FILSETUP – Pobočky ve firmách.
- LAGERTAB – Sklady.
- LIEFERANT – Dodavatelé.

4.2.2 Analýza tabulek dimenzí a faktů

Prvním krokem je nadefinování samotné struktury těchto tabulek, tedy jaké atributy mají obsahovat, co je potřeba sledovat. Atributy tabulek faktů víceméně kopírují atributy faktů použité v analytickém manažeru. Nejsou využity všechny, ale pouze ty, které jsou významnější (častěji sledované) a současně ty, které budou využity k analýze. Jedním z kritérií pro porovnání vybraných technologií je obtížnost přidání nové dimenze nebo faktu do stávajícího datového skladu, případně modifikace jejich struktury. Následující přehled je tedy produktem hrubého předpokladu(s vědomím, že struktura datového skladu by měla být dobře promyšlena dopředu)

co vše bude potřeba a dle následujících požadavků pak bude doplněno. Na druhou stranu lze předpokládat, že v reálném provozu se tato situace určitě vyskytne.

Tabulky dimenzí budou označeny prefixem Dim, zatímco tabulky faktů prefixem Fact.

DimFiliale

Dimenze jejíž předmětem je seznam poboček ve firmách, ke kterým se vztahují jednotlivé prodeje a nákupy. V případě potřeby odlišit pobočky jednotlivých firem by bylo záhodno vytvořit hierarchickou strukturu firma > pobočka. Pro zjednodušení je však tato dimenze složena ze dvou relačních tabulek a název firmy je zohledněn pouze sloupcem firma.

SQL příkaz k vytvoření:

```
create table DimFiliale(
nr int,
bez varchar(500),
firma varchar(500),
strasse varchar(500),
plz varchar(100),
ort varchar(500),
land varchar(100)
)
```

DimArtikel, DimGruppen, DimUntergruppen, DimObergruppen, DimGBereich

K vytvoření dimenze DimArtikel slouží tabulka ARTIKELSTAMM. Tabulky GBEREICH, OBERGRUPPEN, UNTERGRUPPEN, GRUPPEN pro ostatní výše uvedené dimenze.

Tyto dimenze poslouží k vytvoření produktové hierarchické dimenze:

- Gbereiche
- Obergruppen
- Untergruppen
- Gruppen
- Artikel

Je doporučováno nastavit relace mezi jednotlivými relacemi. Dosáhne se tak vyššího výkonu. Standardně je každý atribut implicitně závislý na primárním klíči produktové tabulky (Artikel). Po vytvoření relací bude závislost následující:

Artikel > Gruppen > Untergruppen > Obergruppen > Gbereiche

DimDatum

V případě vytvoření dimenze času je potřeba se zamyslet nad tím, v jakých minimálních časových úsecích bude potřeba data prohlížet. Například jednotlivé prodeje jsou zaznamenávány řádech sekund, což by ovšem kladlo obrovské nároky na prostor a mělo negativní dopad na výkonnost zpracování. Vzhledem k tomu, že to není přímo vyžadováno a pro realizaci a porovnání je dle mého názoru dostačující měřítko jeden den, obsahuje časová dimenze jako klíč pouze datum bez času.

Tabulka DimDatum je vytvořena následujícím SQL dotazem:

```
create table dimDatum(
   [FullDate] [datetime] NOT NULL, --např. 2012-01-30 0:00:00
   [DateName] [char](11) NOT NULL, -- datum
   [DayOfWeek] [tinyint] NOT NULL, -- den v týdnu
   [DayNameOfWeek] [char](10) NOT NULL, -- název dne v týdnu
   [GermanDayNameOfWeek] [char](10) NOT NULL, -- německá lokalizace dne
   [DayOfMonth] [tinyint] NOT NULL, -- den v měsíci
   [DayOfYear] [smallint] NOT NULL, -- den v roce
   [WeekdayWeekend] [char](7) NOT NULL, -- víkendový den
   [WeekOfYear] [tinyint] NOT NULL, --týden v roce
   [MonthName] [char](10) NOT NULL, -- název měsíce
   [MonthOfYear] [tinyint] NOT NULL, --měsíc v roce
   [CalendarQuarter] [tinyint] NOT NULL, -- kvartál
   [CalendarYear] [smallint] NOT NULL, -- rok
   [CalendarYearMonth] [char](7) NOT NULL, --
   [CalendarYearQtr] [char](15) NOT NULL,
CONSTRAINT PK_Date_DateID PRIMARY KEY (FullDate)
) ON [PRIMARY]
```

Co se týká počátečního a koncového data při plnění této tabulky, zvolil jsem datum prvního záznamu v tabulce VERKAUF (bylo by však možné např. umístit generování datové dimenze do úlohy po vytvoření dimenze FactVerkauf a získat tak datum prvního záznamu) a jako cílový datum rok od dne generování. V reálném využití by záleželo na konkrétních požadavcích. To se týká také volby sloupců časové dimenze, kde by mohly přibýt např. lokalizované nebo fiskální položky. SQL skript, který slouží k naplnění časové dimenze je spuštěn vzápětí po vytvoření tabulky DimDatum :

```
DECLARE @StartDate datetime, @EndDate datetime
SELECT @StartDate = '1.1.2005 00:00:00'
--CAST(FLOOR( CAST( (select MIN(datum) from factverkauf) AS FLOAT ) )AS DATETIME)
```

```
select @EndDate = CAST(FLOOR( CAST( dateadd(year,1,GETDATE()) AS FLOAT ) )AS
DATETIME)
WHILE (@StartDate <= @EndDate )
BEGIN
INSERT INTO dimDatum
SELECT
   @StartDate AS [Date]
   ,CONVERT(varchar(20),@StartDate,106) AS DateName
   ,DATEPART(DW,@StartDate)   [DayOfWeek]
   ,DATENAME(DW,@StartDate) [DayNameOfWeek]
   ,'' [GermanDayNameOfWeek]
   ,DATENAME(DD,@StartDate)  [DayOfMonth]
   ,DATENAME(DY,@StartDate)  [DayOfYear]
   ,CASE WHEN DATEPART(DW,@StartDate) THEN 'WeekEnd'
             ELSE 'WeekDay' END     [WeekdayWeekend]
   ,DATEPART(WW,@StartDate) [WeekOfYear]
   ,DATENAME(MM ,@StartDate) [MonthName]
   ,DATEPART(MM ,@StartDate)   [MonthOfYear]
   ,DATEPART(QQ,@StartDate)     [CalendarQuarter]
   ,DATEPART(YY ,@StartDate)    [CalendarYear]
   ,DATENAME(YY,@StartDate)+'-'+RIGHT('0'+CAST(Month(@StartDate) as varchar),2)
[CalendarYearMonth]
   ,DATENAME(YY,@StartDate)+'-Q'+DATENAME(QQ,@StartDate) [CalendarYearQtr]

   SET @StartDate =DATEADD(day,1, @StartDate) --nastavení granularity
END
GO
SET NOCOUNT OFF
SET LANGUAGE german
update dimDatum set GermanDayNameOfWeek = DATENAME(DW,FullDate)
SET LANGUAGE english
```

V časové dimenzi je vytvořena hierarchie Calendar:

- Calendar Year
- - Calendar Quarter
- - - Month Of Year
- - - - Day Of Month
- - - - - Full Date

Opět pro ni platí, stejně jako u produktové dimenze, že je vhodné nastavit relace mezi atributy:

Full Date > Day Of Month > Month Of Year > Calendar Quarter > Calendar Year

V některých případech je důležité analyzovat konkrétní hodnoty v intervalech po hodinách v jednotlivých měsících. K tomuto účelu by bylo potřeba vytvořit další časovou dimenzi, kde

budou položky hodina od – do, k jejímuž zkonstruování poslouží SQL příkaz case. Druhým sloupcem by byl měsíc. Tabulka faktů by pak byla pomocí těchto sloupců svázána s časovou dimenzí svázána.

DimLieferant

Seznam dodavatelů je založen na tabulce LIEFERANT.

FactKundenrechnungen

Tabulka zahrnuje vydané faktury, které obsahující následnující sloupce:

- Rechnungsbuch-Brutto – Suma faktur.
- Belegzahl – Počet faktur.
- Kundenzahl – Počet různých zákazníků.
- Offener Betrag – Nezaplacená suma.
- Bezahlter Betrag – Zaplacená suma.

FactLieferantrechnungen

Obsahem tabulky jsou přijaté faktury.

FactVerkauf

Pozice na vydaných fakturách:

- Verkaufte Menge – Prodané množství.
- Verkauf-Brutto – Hrubá prodejní částka.
- Verkauf-Netto – Čistá prodejní částka.
- Verkauf-Rohertrag – Čistý výnos, bez daně.
- Verkauf-EK-Umsatz – Suma faktur v nákupních cenách.
- Verkauf-Abw. Netto – Suma slev bez daně (netto).
- Verkauf-Abw. Brutto – Suma slev s daní (brutto).
- Kundenzahl – Verkauf – Počet různých zákazníků.

FactArtikelstamm

Výrobky s těmito položkami:

- Lagerbestand – Stav na skladu.
- Lagerwert – Cena produktu na skladě.
- Bestellmenge – Množství.
- Bestellwert – Objednávací cena.
- Reservierte Menge – Rezervované množství.

FactEinkauf

Jedná se o příjem zboží (na úrovni jednotlivých pozic). Tabulka obsahuje položky:

- Einkaufswert (Netto-Netto) – Čistá nákupní cena.
- Einkaufsmenge – Množství.
- Einkauswert(Rechnung)
- Einkaufswert(VK)
- Einkaufsmenge noch im Bestand

FactEinkaufrabatt

Rabaty na příjmu zboží:

- Rabatt

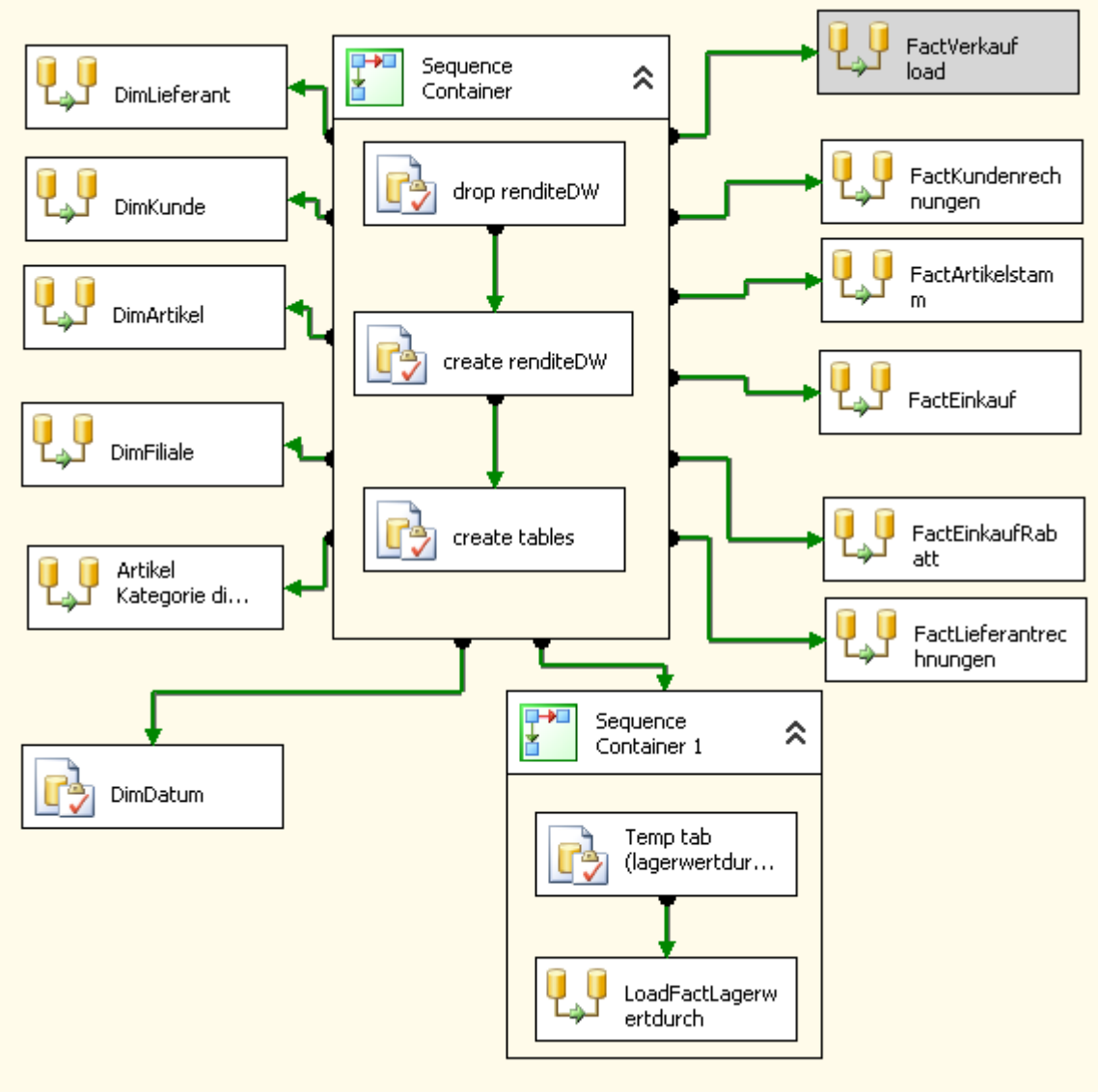
FactLagerwertdurch

Existuje požadavek na zobrazení průměrné hodnoty zboží na skladu za určitý časový interval, avšak účel analytické databáze není počítat v reálném čase data ze zvolených parametrů a je potřeba zvolit jiné řešení. Tato tabulka představuje průměrnou hodnotu zboží na skladě po jednotlivých dnech, jak ukazuje následující SQL dotaz:

```
create table templager(
datum datetime not null,
artnr int not null,
filiale int not null,
lagerwertdurch float not null
```

```
)

declare @StartDate datetime, @EndDate datetime
select @StartDate ='1.1.2012'
select @EndDate = '1.2.2012'
while (@StartDate <= @EndDate )
begin
insert into templager
       select @StartDate AS datum,dlwt.artnr,dlwt.filiale, dlwt.durchlagerwert as
lagerwertdurch
       from artikelfildurchlagerwerte(@StartDate, dateadd(day,1,@StartDate)) dlwt
   set @StartDate =dateadd(day,1, @StartDate)
end
```



Ilustrace 15: Úlohy SSIS projektu.

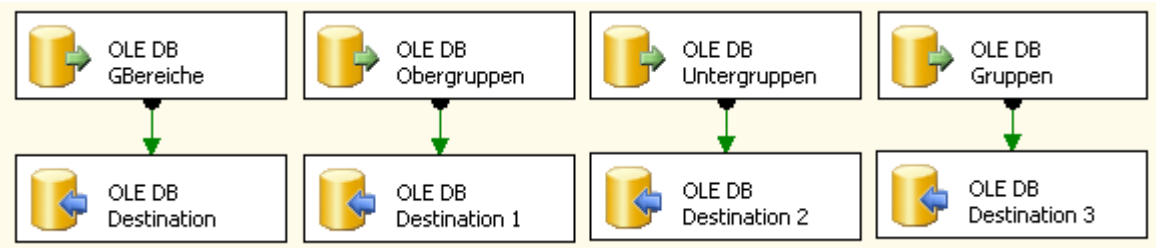
Tabulka obsahuje kombinace produktu a pobočky za jednotlivé dny za zvolené časové období. Pro výpočet hodnoty za jeden den je použita interní funkce databáze Rendite

artikelfildurchlagerwerte (od,do). Dočasná tabulka je po procesu ETL smazána. Teoreticky by šla použít přímo SQL temp tabulka, avšak pro zjednodušení (je potřeba zachovat v projektu temp tabulku v několika úlohách) jsem použil tabulku klasickou.Proces ETL

Projekt SSIS obecně

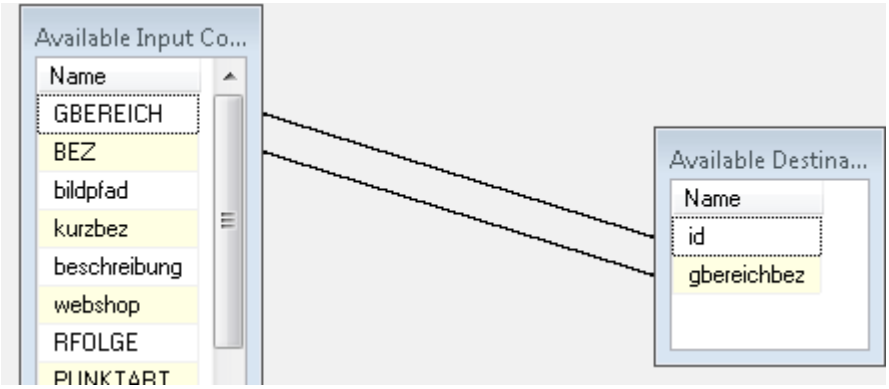
Ilustrace 15 ukazuje výřez SSIS projektu vytvořeného v BIDS, jehož účelem je jednorázové naplnění datového skladu daty z transakční databáze RenditeTest dle předchozích požadavků. Konkrétně se jedná o schéma složené z řídících úloh (Control Flow), které v sobě dále zahrnují vnořené úlohy (Data Flow) viz Ilustrace 16. Výhodou řídících úloh je možnost jejich deaktivace a správa chybových stavů (deaktivace úloh na úrovni Data Flow není možná).

Prvním krokem procesu ETL je smazání již existující databáze RenditeDW. Tato úloha (Execute SQL Task) má definovanou referenci na soubor (který může nebo nemusí být součástí projektu) s SQL příkazem, který smazání provádí. Úlohy na vytvoření prázdné databáze RenditeDW a vytvoření všech tabulek dimenzí a faktů jsou stejného typu a jsou umístěné v sekvenčním kontejneru, aby byly prováděny postupně. Každá úloha typu Execute SQL Task musí mít definované připojení k databázi (Connection Manager) v jehož kontextu se daný SQL příkaz, ať už ze souboru nebo vložený přímo do úlohy, spouští. Obsah řídící úlohy Artikel Kategorie Dimensions je na obrázku Ilustrace 16. Jedná se o schéma (Data Flow) naplnění tabulek datového skladu týkajících se produktové dimenze. OLE DB zdroje v horní řadě využívají připojení k transakční databázi, v dolní řadě připojení k datovému skladu. Zatímco u Execute SQL Task úlohy jsem k úloze připojoval přímo externí soubor s SQL dotazem, v případě OLE DB zdrojů je využito SQL dotazu vloženého přímo do projektu a vztaženého ke konkrétní úloze nebo přímá volba tabulky nebo pohledu. Jedním z důvodů, že jsem nevyužil externího souboru je to, že jsem nebyl schopen dohledat, jak jednoduše externí soubor k OLE DB zdroji přiřadit. V případech, kdy jako zdroj účinkuje pouze jedna fyzická tabulka, jsem zvolil přímo tabulku. V případě spojení dvou a více tabulek SQL dotaz.



Ilustrace 16: Data Flow úlohy.

Zelená šipka na obrázku Ilustrace 16 mezi zdrojem a cílem představuje tok dat mezi jednotlivými úlohami. Možnost přiřazení jednotlivých sloupců zdrojové tabulky sloupcům cílové tabulky ukazuje obrázek Ilustrace 17. Vzhledem k výkonu by měly být v budoucím projektu využity jen využité sloupce, ale vzhledem k tomu, že se jedná o jednorázový ETL proces, upřednostnil jsem rychlost provedení, a tedy výběr tabulky z datového zdroje, nikoli SQL dotaz s filtrem na sloupce GBEREICH a BEZ.



Ilustrace 17: Přiřazení sloupců.

Mezi zdroje a cíle lze zařadit celou škálu úloh jako je pivot, spojení tabulek, modifikace sloupců apod. Jelikož je mi bližší SQL jazyk tyto úlohy jsem nevyužíval. Krom jiného také proto, že například u modifikace sloupců je zapotřebí využít výrazů (SSIS Expressions) mírně odlišných od SQL a opět z hlediska rychlosti provedení bylo pro mne jednodušší využít SQL a modifikovat sloupec přímo ve zdrojovém dotazu.

Primární a cizí klíče, indexy

Při procesu ETL jsou všechny klíče a indexy vymazány a je potřeba je definovat následně po transferu dat. Teoreticky je lze nastavit v pohledu na datového zdroje(data source views) což ale s sebou přináší mnohá úskalí, jak bylo popsáno v kapitole 3.2.5.

Spuštění integračního projektu

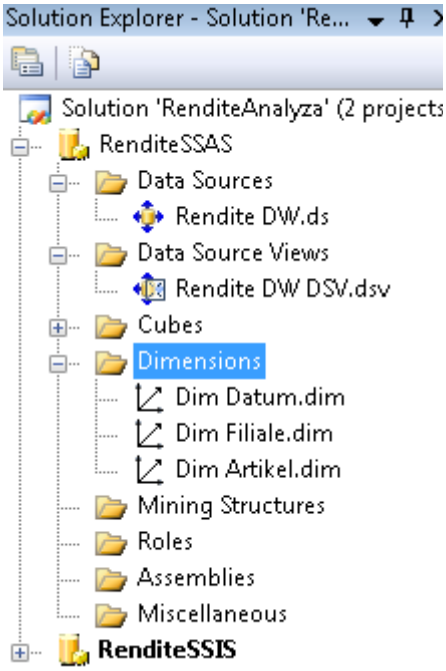
Při tvorbě a ladění SSIS projektu stačí spouštět projekt či jednotlivé úlohy z prostředí Visual Studia. Pro nasazení do praxe a spouštění jednoho nebo více projektů automaticky v naplánovanou dobu je potřeba využít SQL Server Agent. Při vytvoření nové úlohy(new job) je potřeba změnit typ na SSIS a specifikovat cestu k projektovému souboru dtsx.

4.2.3 Realizace krychlí

Při realizaci krychlí byl přikládán význam porovnání jejich pracnosti vytváření s analytickým manažerem v Rendite. Kromě „obyčejných“ krychlí s jednou tabulkou faktů byly vybrány krychle se dvěma nebo více. Výběr byl zaměřen taktéž na výpočet náročnější přehledy např. výpočet průměrné hodnoty zboží na skladu za určité období.

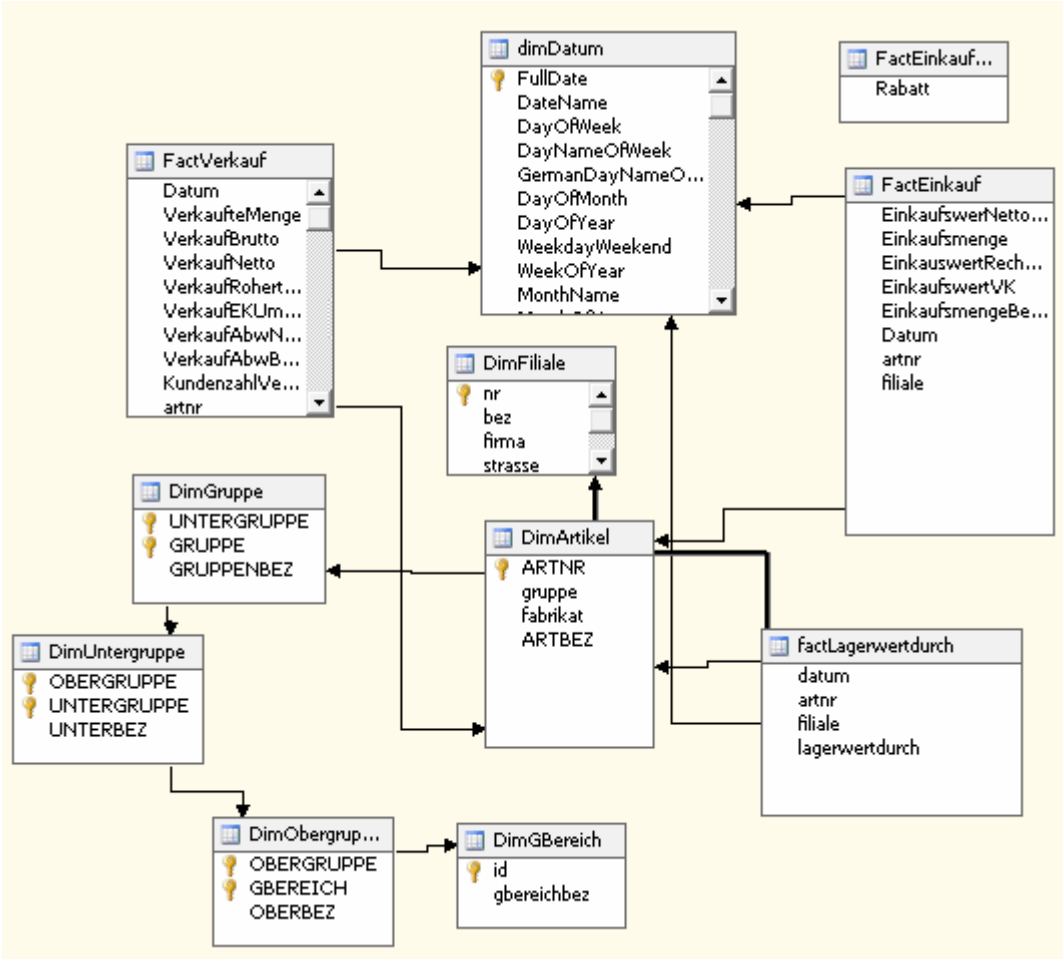
Projekt analytických služeb (SSAS) obecně

Ve fázi přípravy na vytváření OLAP krychlí jsou již k dispozici naplněné tabulky dimenzí



Ilustrace 18: Projektový strom RenditeAnalyza.

a faktů v datovém skladu po procesu ETL. Prvním krokem je vytvoření projektu SSAS a definování spojení k databázi RenditeDW. Poté je třeba vytvořit pohled na datový zdroj a vložit více méně automaticky všechny tabulky dostupné v databázi RenditeDW (viz Ilustrace 19). Tento krok je důležitý, neboť dimenze, fakta a potažmo datové krychle je možné vytvářet pouze ze zdrojů dostupných v některém z pohledů na datové zdroje. V této chvíli jsou v logické vrstvě pohledu na datový zdroj všechny dostupné tabulky a je potřeba přidat dimenzi do OLAP databáze. Přidané dimenze jsou vidět v projektovém stromu (Ilustrace 18). Vytvoření dimenze je stejně jako při vytváření datového pohledu téměř automatické a je založeno na již existující tabulkách v datovém skladu, respektive v pohledu datové zdroje. Následují kapitoly na vytváření konkrétních krychlí.



Ilustrace 19: Výřez části schéma datového pohledu (DSV).

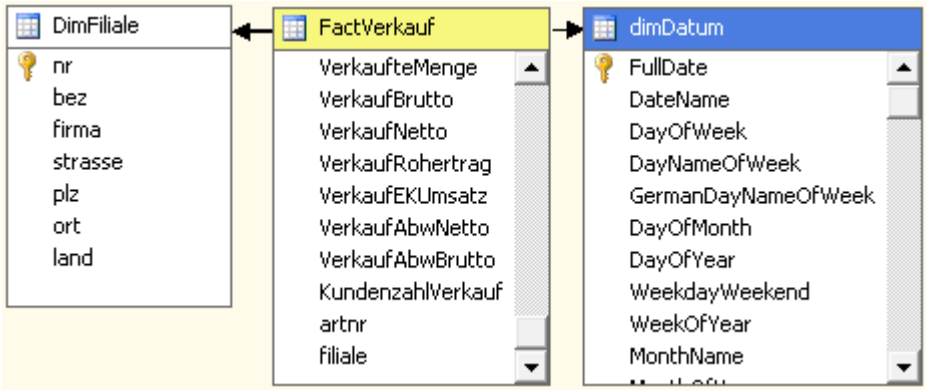
Erfolgsauswertung pro Tag

Krychle poskytuje Verkauf-Netto a Verkauf-Rohertrag z tabulky faktů FactVerkauf rozlišitelné podle roku, měsíce a dne. Případně i dle poboček firem. K tomu poslouží dimenze DimDate a DimFiliale. Schéma krychle je zobrazeno na Ilustrace 20.

Průměrná hodnota zboží na skladě (Lagerwertdurch)

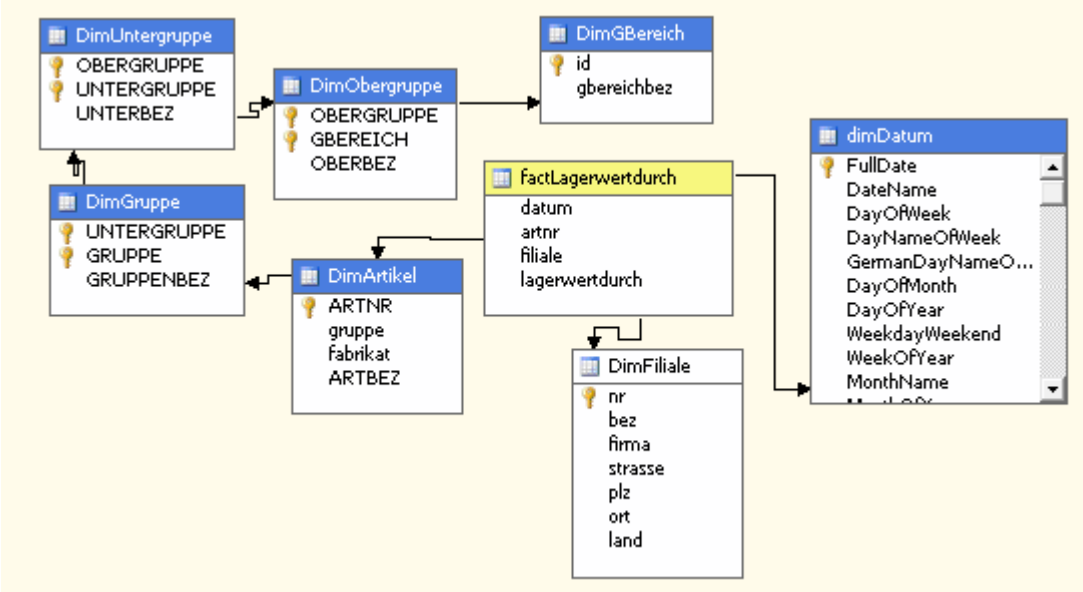
Schéma krychle je na obrázku Ilustrace 21. Uživatel by měl být schopen pohlížet na data s pomocí dimenzí času a produktu, případně pobočky. V případě Rendite uživatel zadá časový interval pro nějž jsou získány potřebné hodnoty. Zadávání vstupních parametrů není v případě kostky určené výhradně pro čtení možné, a proto je potřeba připravit kostky s požadovanými

časovými úseky (např. pro každý rok).



Ilustrace 20: Schéma krychle Erfolgsauswertung pro Tag

Úskalím při vytváření této krychle bylo to, že je potřeba na každé úrovni zohlednit průměrnou hodnotu produktu na skladu za zvolené období. To je realizováno pomocí dvou počítaných sloupců:



Ilustrace 21: Schéma krychle Lagerwertdurch.

Počet dní na dané úrovni :

```
--[Pocet dni]
Count(Descendants([Dim Datum].[Calendar].currentmember, [Dim Datum].[Calendar].
[Full Date]), excludeempty)
```

Vypočítaný průměr:

```
--[Prumer]
Measures.[lagerwertdurch]/[pocet dni]
```

Na obrázku Ilustrace 22 jsou zobrazené průměrné hodnoty na skladu v hlavní kategorii produktů

Drop Filter Fields Here

			Drop Column
Calend.	Month	Gbereichbez	Prumer
⊟ 2012	⊟ 1	⊞ Car-Multimedia / Navigation	134,234 €
		⊞ Datenimport	6,227 €
		⊞ Elektro-Haushaltsgeräte	1,869,800 €
		⊞ Fremdware	65,413 €
		⊞ michal	50 €
		⊞ PC / Multimedia	923,478 €
		⊞ Telekommunikation	1,042,848 €
		⊞ Unterhaltungselektronik	1,768,692 €
		⊞ Unterhaltungsmedien/Software	154,418 €
		⊞ Werbemittel	-200 €
		⊞ Werkstatt	56 €
		Total	5,965,017 €
	⊞ 2		206,597 €
	Total		521,603 €
Grand Total			64,802 €

Ilustrace 22: Průměry v prohlížeči kostky (OLAP).

pomocí prohlížeče kostek ve Visual Studiu 2008. Obrázek Ilustrace 23 slouží k porovnání s hodnotami, které byly za období od 1.1.2012 – 31.1.2012 vygenerovány pomocí analytického manažeru v Rendite.

01/01/2012 - 31/01/2012

11. 2011 - 11. 2011

Name test

0 Tabelle | 1 Parameter | 2 Gitter | 3 Graph | 4 Graph 2

Y-Achse

☑ Geschäftsbereich

Geschäftsbereich	Lagerwert-durchschn.
Datenimport	6,227.47 €
Fremdware	65,412.74 €
Unterhaltungselektronik	1,768,691.98 €
Car-Multimedia / Navigation	134,234.26 €
Telekommunikation	1,042,847.83 €
PC / Multimedia	923,478.25 €
Unterhaltungsmedien/Software	154,418.47 €
Elektro-Haushaltsgeräte	1,869,799.85 €
Werkstatt	55.77 €
Werbemittel	-200.00 €
michal	50.00 €
Gesamtsumme	**5,965,016.63 €**

Ilustrace 23: Konkrétní průměry v Rendite.

Sloupce lagerwertdurch a [pocet dni], které byly využity pro výpočet průměru, jsou skryté. Sloupec [Prumer] je zaokrouhlen a naformátován, aby zobrazoval znak € a odděloval čárkou tisíce.

4.3 Aplikace dolování dat

Obsahem této kapitoly je využití služeb BI z oblasti dolování dat. Budou vybrány takové úlohy, které jsou v praxi nejčastěji využívané.

4.3.1 Oslovení cílové skupiny s nabídkou produktu

Motivace

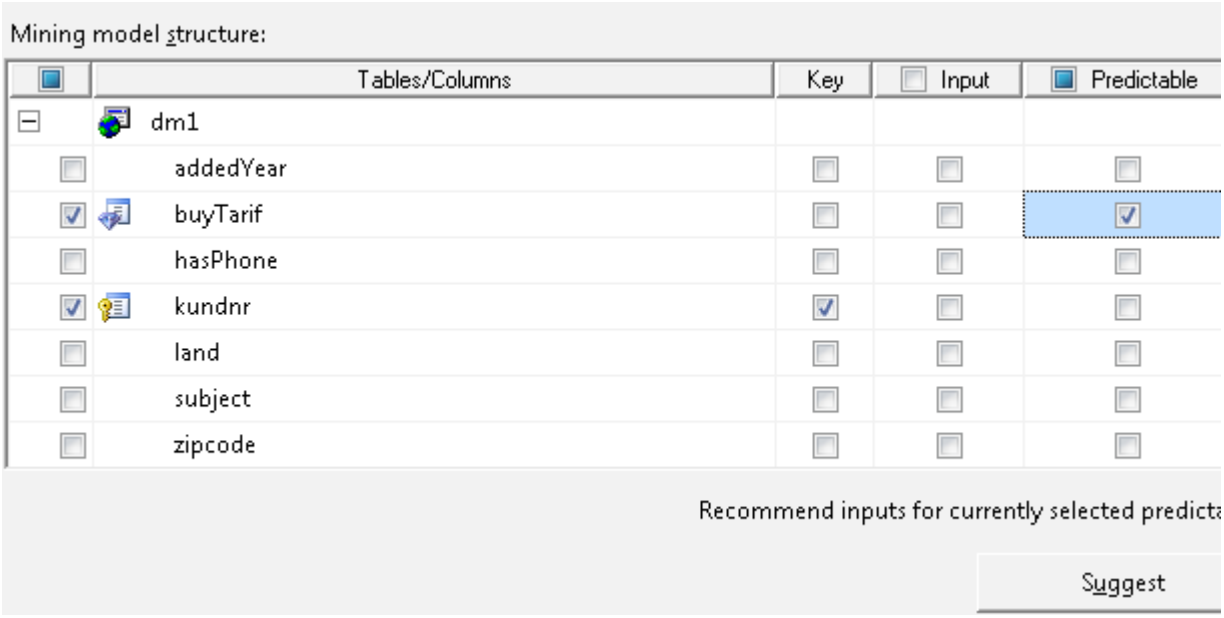
Cílem je oslovit vybrané zákazníky s tím, že jim bude nabídnut vybraný produkt. Je jedno, zda-li to bude telefonickou formou nebo poštou. Problémem je relativně velké množství zákazníků a omezené prostředky na to, aby mohli být osloveni všichni zákazníci.

Zdrojová data

Jelikož je teoreticky možné využívat služby dolování dat bez zavedení datového skladu a procesu SSIS, poslouží jako zdroj dat pohled z transakční databáze RenditeTest, který sice bude uložen v datovém skladu, avšak neprojde procesem SSIS, nýbrž položky budou upraveny v těle SQL příkazu pohledu:

```
use renditeDW
go
create view dm1 AS
select k.kundnr,
lower(k.land) as land,
(case when exists (SELECT r.kundnr FROM [renditetest].[dbo].verkauf v
left join [renditetest].dbo].rechbuch r on r.rechnr = v.rechnr
WHERE (r.kundnr = k.kundnr and r.art = 'R')
and ((v.artnr = '794'))) then 1 else 0 end) as buyTarif,
(case when k.telefon1 is null and k.telefon2 is null then 0 else 1 end) as hasPhone,
(
case k.anrede
when 2 then 'M'
when 13 then 'M'
when 3 then 'F'
when 14 then 'F'
else 'O'
end) as [subject],
k.plz as zipcode,
datepart(year,k.erf_dat) as addedYear
--mobiltel malo
from [renditetest].[dbo].kundstamm k
--794
where k.land is not null and k.land !='' and k.anrede is not null
```

Výše uvedený pohled v sobě zahrnuje vybraná data zákazníků z tabulky KUNDSTAMM v závislosti na prodejích (RECHBUCH,VERKAUF) produktu s identifikačním číslem 794, což je jeden z nejprodávanějších výrobků v této databázi a jedná se o obecný tarifový balíček k mobilnímu telefonu. Predikovaný údaj tedy bude ten, zda si zákazník koupil nebo nekoupil tento balíček.



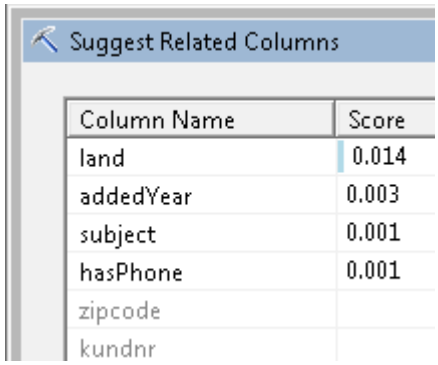
Ilustrace 24: Nastavení charakteru sloupců.

Data byla vybírána z pohledu toho, zda jsou relevantní ke sledované položce a také v jaké míře se v databázi vyskytují. Údaje jako adresa, jméno a příjmení byly vynechány. Sloupec mobiltel byl ve většině případů prázdný a lze se tedy domnívat, že se nejedná o to, že by zákazník mobilní telefon neměl, nýbrž jen nebyl vyplněn. Naopak sloupce telefon1 a telefon2 byly zastoupeny v takové míře, že tento údaj byl zahrnut jako relevantní. Tabulka KUNDSTAMM neobsahuje pohlaví, avšak je využito oslovení(anrede),které je uvedeno téměř u všech zákazníků, k získání tohoto údaje. Kromě poštovního směrovacího čísla a roku zavedení zákazníka se vyskytuje také země odkud zákazník pochází. Je použito funkce lower, neboť v databázi jsou ve velké míře uvedeny kódy stejné země jako např. „de“, „DE“. Vyskytuje se také pár případů, kdy je kód země buď prázdný nebo vyplněn chybně (číslem), avšak prázdné záznamy jsou filtrovány a chybně vyplněné, s ohledem na počet záznamů v řádu desítek tisíc, ignorovány.

Tabulka KUNDSTAMM obsahuje mnoho dalších sloupců, ale bohužel nebyly vyplněny(např. datum narození) a je otázkou, zda jsou použité údaje dostatečné k predikci. To bude v dalších krocích ověřeno.

Realizace

Nejprve je potřeba vytvořit novou strukturu(Mining Structure), což znamená definovat zdroj dat, na kterých se bude model učit, a zvolit jaká technika(algoritmus) bude pro model využita. Pro

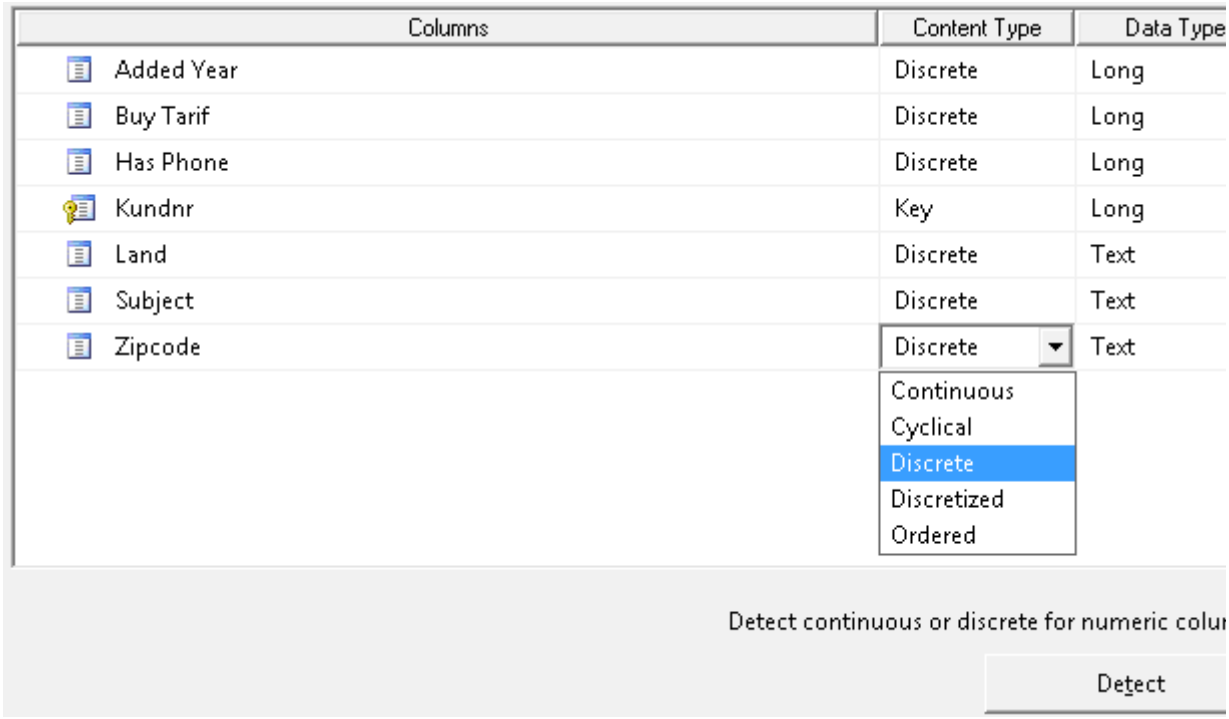


Ilustrace 25: Výsledek detekce atributů.

tento případ bude zvolen algoritmus Microsoft Decision Trees. Další modely využívající jiné algoritmy pak mohou být a budou vytvořeny dodatečně. Jako zdroj dat je zvolen pohled dm1. Poté, jak je uvedeno na obrázku Ilustrace 24, je potřeba nastavit charakter jednotlivých sloupců. Klíčovým sloupcem je jednoznačně kundnr. Po označení sloupce buyTarif jako predikovaného se aktivuje tlačítko suggest, které slouží k automatické detekci vstupních atributů, to znamená označení významných a méně významných atributů. Výsledek detekce je na obrázku Ilustrace 25.

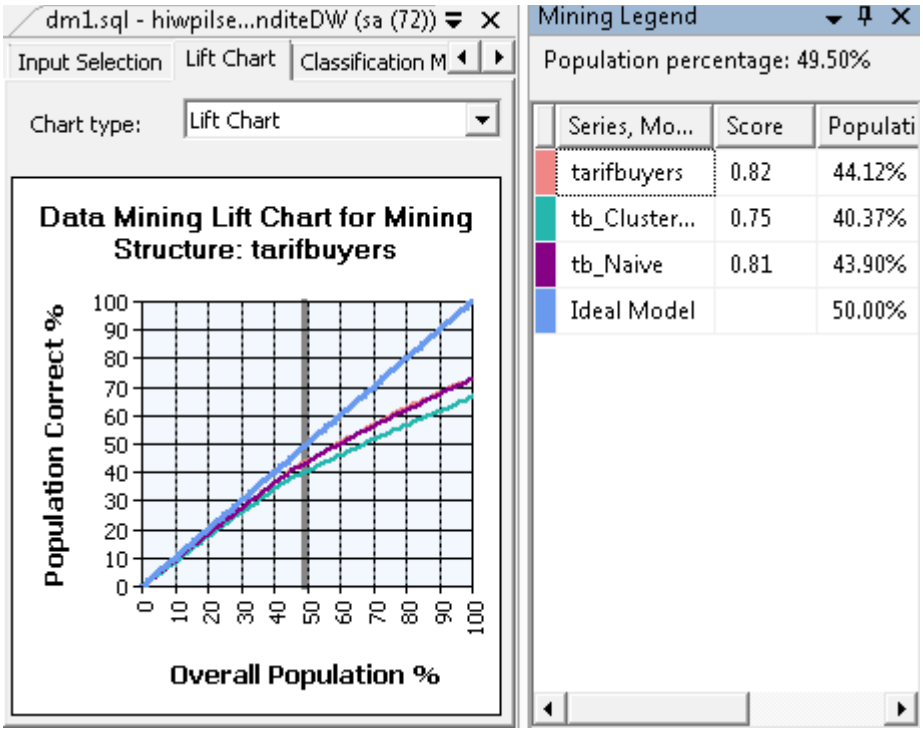
Skóre celkově není příliš vysoké, což ukazuje na to, že závislost ostatních atributů na predikovaném atributu není uspokojivá. Nejvyšší hodnotu má land, naopak údaj zipcode není vůbec doporučen jako vstupní sloupec.

Bez ohledu na výsledek předpokladu byly zvoleny všechny dostupné sloupce jako vstupní a je potřeba určit jakého datového typu a typu obsahu jsou jednotlivé sloupce jak ukazuje obrázek Ilustrace 24. K dispozici je tlačítko detect, které nastaví typy automaticky. Po jeho aplikaci se přiřadilo všem sloupcům(content type), kromě klíčového, diskrétní hodnota, datový typ zůstává nezměněn. Posledním krokem před vytvořením modelu je potřeba nastavit velikost testovací množiny ze vstupních dat v procentech. Trénovací množina slouží k vytvoření modelu, zatímco testovací množina ověřuje přesnost modelu. Případně je možné nastavit minimální počet případů, který obsahuje testovací množina. Po sestavení a nasazení na server je možné na každý vytvořený model nahlížet specifickými prohlížeči modelů.



Ilustrace 26: Nastavení typu sloupců.

V záložce Mining Models byly přidány další dva modely s algoritmy Microsoft Clustering a Microsoft Naive Bayes. Kroky při vkládání jsou pouze výběr algoritmu a název modelu, struktura zůstává stejná. Důvodem je, že ukázkový případ klasifikace umožňuje zvolit více algoritmů a lze pak zvolit ten nejpřesnější, což ukazuje obrázek Ilustrace 27.



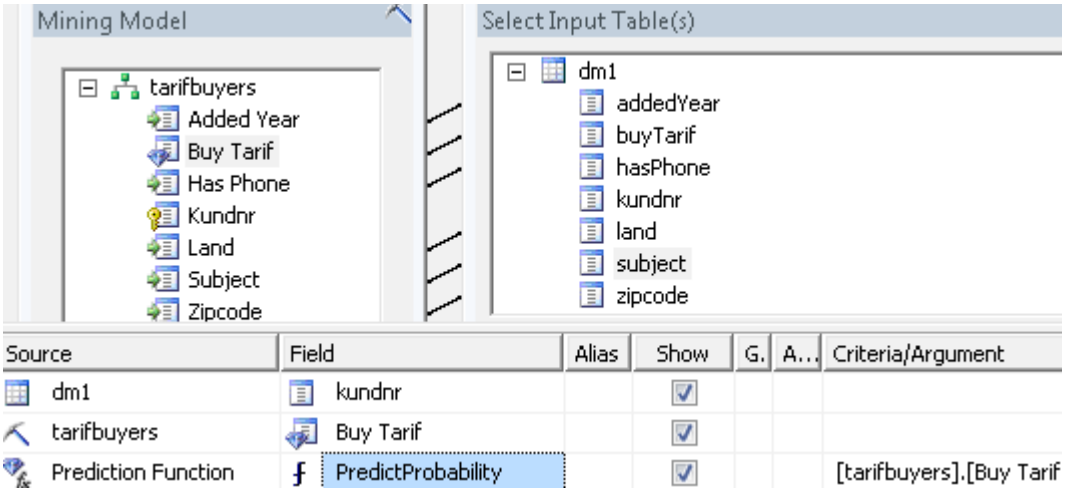
Ilustrace 27: Porovnání algoritmů.

Na grafu (lift chart) vychází nejhůře tb_Clustering (Microsoft Clustering), který se nejvíce odkloňuje od ideální křivky. Zbylé dva algoritmy jsou na tom víceméně podobně. Celkové odklonění všech křivek od ideální se však zdá být relativně velké, což poukazuje buď na špatně zvolený algoritmus, což je v tomto případě eliminováno zvolením několika algoritmů, anebo na nevhodně zvolený model, který se opírá o zvolená data. K zobrazení grafu je použito SQL Management Studio, neboť Visual Studio 2008 toto neumožnilo kvůli blíže nespecifikované chybě, přestože byla v obou případech použita stejná databáze na stejném serveru.

Pro predikci nových případů bude dle předchozích informací využit model tarifbuyers(Microsoft Decision Trees). Na obrázku Ilustrace 28 je zobrazen návrhář predikčního dotazu. Jako vstupní zdrojová tabulka k ověření modelu poslouží už existující pohled dm1. Je potřeba určit pole s klíčem a predikované pole v modelu. Následně je potřeba vložit predikční funkci, která bude mít jako argument predikovaný sloupec modelu.

Tímto je sestaven predikční dotaz uvedený níže, který vrátí výsledky v podobě jak je uvedeno na obrázku Ilustrace 29.

```
SELECT
  t.[kundnr],
  [tarifbuyers].[Buy Tarif],
  PredictProbability([tarifbuyers].[Buy Tarif])
From
  [tarifbuyers]
PREDICTION JOIN
  OPENQUERY([Rendite DW ds],
    'SELECT
      [kundnr],[land],[buyTarif],[hasPhone],[subject],[zipcode],[addedYear]
    FROM
      [dbo].[dm1]
    ') AS t
ON
  [tarifbuyers].[Land] = t.[land] AND
  [tarifbuyers].[Buy Tarif] = t.[buyTarif] AND
  [tarifbuyers].[Has Phone] = t.[hasPhone] AND
  [tarifbuyers].[Subject] = t.[subject] AND
  [tarifbuyers].[Zipcode] = t.[zipcode] AND
  [tarifbuyers].[Added Year] = t.[addedYear]
```



Ilustrace 28: Návrhář predikčního dotazu.

Výsledná tabulka z obrázku Ilustrace 29 byla uložena do tabulky dm1predictRes. Zjištění, kolik případů při predikci bylo chybných, ověří následující SQL dotaz:

```
select count(*)
from dm1predictres p
left join dm1 v on p.kundnr = v.kundnr
where v.buytarif <> p.[buy tarif]
```

Výsledkem je 18775 chybných predikcí z celkových 68557. Jinými slovy bylo za použití vybraného modelu chybně předpovězeno 18775 zákazníků v otázce koupě vybraného produktu.

kundnr	Buy Tarif	Expression
100215	0	0.9492441279...
100216	0	0.9492441279...
100217	0	0.9492441279...

Ilustrace 29: Výstup dm1predictRes.

4.3.2 Analýza nákupního košíku

Motivace

Aplikace tohoto modelu je zřejmě nejčastěji využívána v elektronických obchodech za účelem zvýšení prodeje. Po vybrání jednoho výrobku jsou nějakým způsobem zobrazeny ty výrobky, které si ostatní uživatelé obchodu již zakoupili.

Zdrojová data

Jako zdroj poslouží dva pohledy. První obsahuje seznam objednávek a druhý seznam produktů,

které každá objednávka obsahovala:

```
use renditeDW
go

create view orders
as
select r.rechnr as id,r.kundnr
from tg.dbo.rechbuch r
where r.art='R' and r.kundnr is not null and
not exists (SELECT v.rechnr from tg.dbo.verkauf v where v.rechnr = r.rechnr and
v.artbez is null)
go

create view ordersItems
as
select
rechnr as id, nr as lineID, artbez as productName
from tg.dbo.verkauf
where artbez is not null
```

Pro lepší přehlednost je vybrán název produktu a vzhledem k tomu, že některé objednávky obsahují produkty s hodnotu artbez = null – proto, že se dají prodávat výrobky i jen podle skupiny (časopisy, kdyz se něco prodává na objednávku a normálně to není vedeno, drobné zboží atd.) – jsou z pohledu vyřazeny.

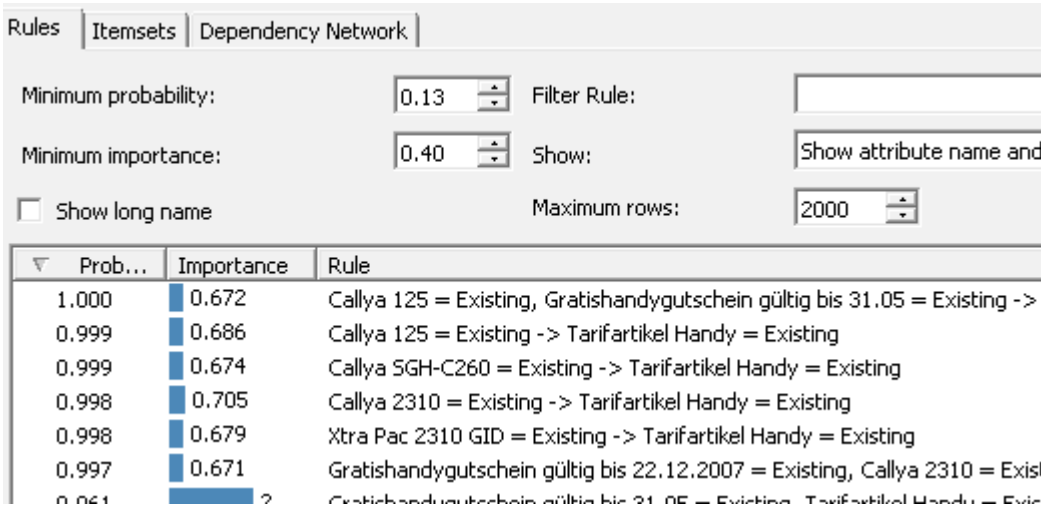
Realizace

Stejně jako v prvním případě dolování dat je potřeba zvolit zdrojová data a algoritmus, což je Microsoft Association Rules. Ilustrace 30 ukazuje nutné nastavení charakteru obou pohledů. Objednávky orders jsou případy (case) a jednotlivé položky ordersItems vnořené (nested).

Ilustrace 30: Nastavení vstupních tabulek.

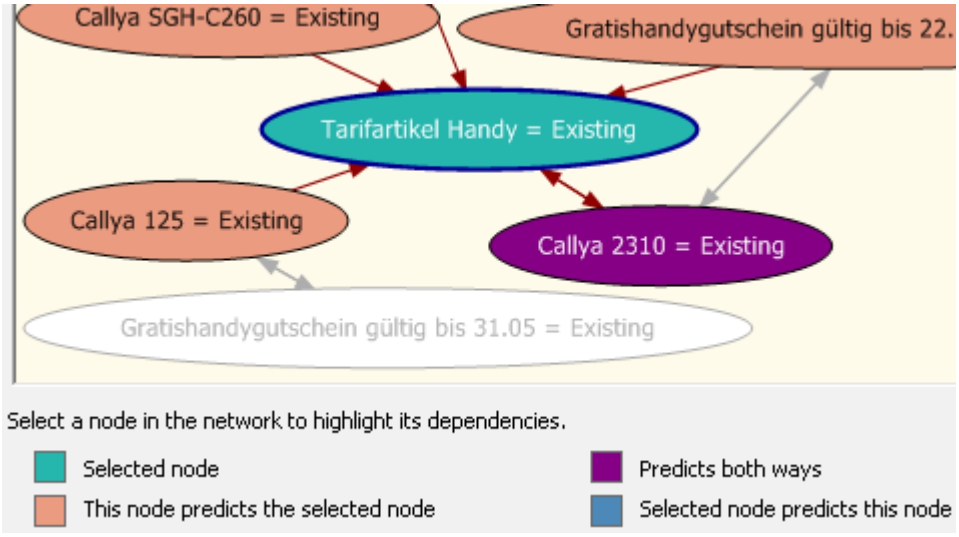
V datovém pohledu (RenditeDW DSV) je potřeba nastavit vztah 1:M mezi orders a ordersItems. Dalším krokem je definice typů jednotlivých atributů. Orders.id bude klíč a ordersItems.productName bude klíč, vstup i predikovaná hodnota. OrdersItems.lineID nebude brát v potaz, neboť v tomto případě nezáleží na pořadí produktů v košíku. Typy sloupců id a

productName jsou detekovány automaticky a korektně. Průvodce nabízí nastavení trénovací množiny, přestože to není při volbě Microsoft Association Rules podporováno. Pro odfiltrování nedůležitých pravidel je zvýšena hodnota podpora (support) ve vlastnostech modelu na 0.01. Výsledný vytvořený a sestavený model je zobrazen v prohlížeči modelů (Mining Model Viewer) na obrázku Ilustrace 31.



Ilustrace 31: Prohlížení modelu.

U každého pravidla je zobrazena hodnota důležitosti (importance). Čím je vyšší, tím lépe. Tato hodnota pomáhá určit spolu s pravděpodobností charakter jednotlivých pravidel.

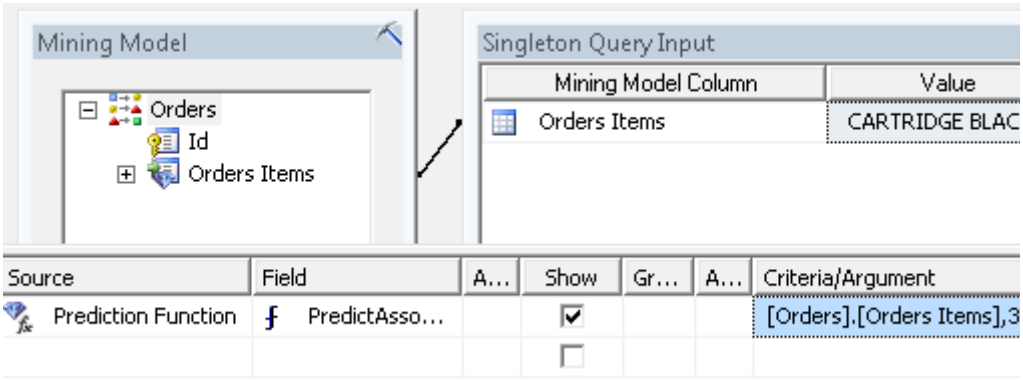


Ilustrace 32: Síť závislostí.

Stejné vztahy, ale jiným způsobem, prezentuje síť závislostí (Dependency Network). Na obrázku Ilustrace 32 je znázorněna míra závislosti na vybraném produktu a to v obou směrech. Tarifartikel Handy je kupován současně s přístrojem Callya 125. Callya 2310 je kupována spolu

s Tarifartikel a naopak. Produkt Gratishandygutschein je označen jako neaktivní z důvodu nesplnění hodnoty síly vztahu, která je nastavena ve vlastnostech grafu.

K praktickému využití tohoto modelu bude potřeba sestavit predikční model podobně, jak je uvedeno v případě prvního modelu. Vzhledem k tomu, že se jedná o asociační algoritmus, je potřeba vybrat predikci asociační funkcí a posléze zvolit kritérium jak je uvedeno na obrázku Ilustrace 33.



Ilustrace 33: Výběr asociační funkce.

Výsledkem návrháře je následující vygenerovaný příkaz, který po spuštění zobrazí 3 (hodnota 3 je zvolena v kritériu) produkty, které se k vybranému produktu vztahují:

```
SELECT
  PredictAssociation([Orders].[Orders Items],3)
From
  [Orders]
NATURAL PREDICTION JOIN
(SELECT (SELECT ' CARTRIDGE BLACK T055140' AS [Product Name]) AS [Orders Items]) AS
t
```

Výsledek dotazu je za použití produktu CARTRIDGE BLACK stejný, jako kdyby žádný parametr nebyl definován, protože pro tento produkt nebyla vytvořena žádná pravidla. Dotaz tedy vrátí tři produkty Tarifartikel Handy, Callya 2310 a Gratishandygutschein gültig bis 22.12.2007. V případě pokud je parametrem Tarifartikel Handy, výsledky jsou Callya 2310, Gratishandygutschein gültig bis 22.12.2007 a Gratishandygutschein gültig bis 30.5. Při praktickém nasazení bude samozřejmě potřeba odfiltrovat nežádoucí produkty, v tomto případě již s neplatným datem. V praxi by také bylo bezesporu výhodnější použít jako parametr číslo výrobku spíše než jeho název, aby se předešlo duplikacím.

4.3.3 Předpovídání prodeje produktů

Motivace

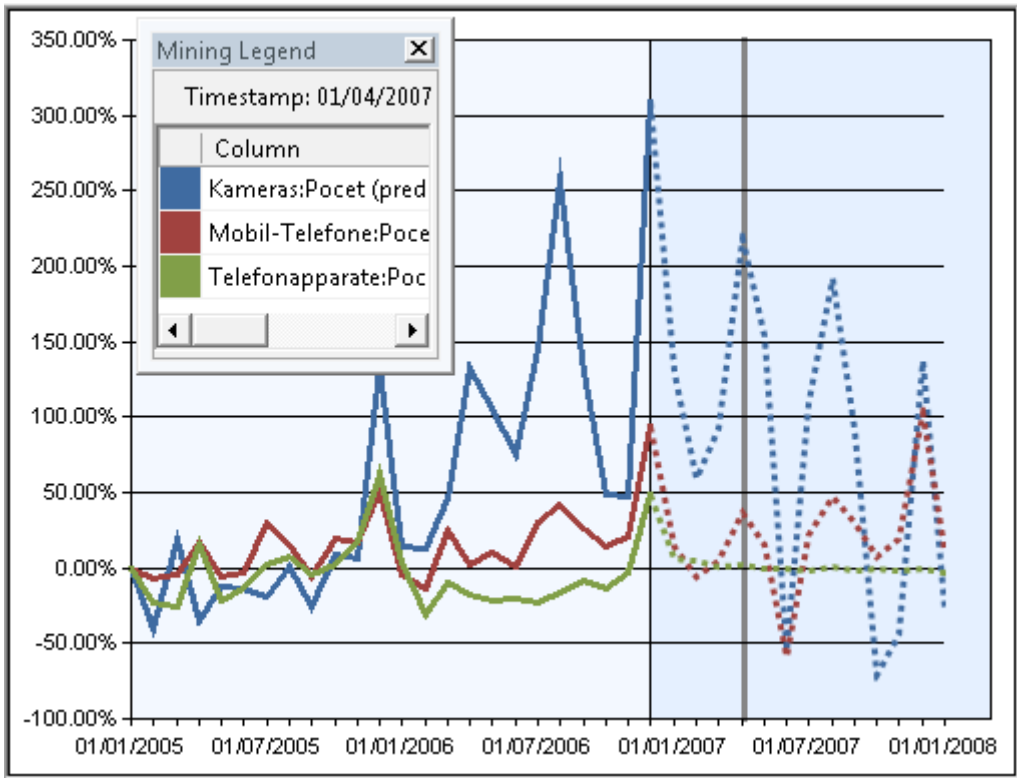
Předmětem je předpovědět jak se bude prodávat jeden nebo více produktů v následujících časovém horizontu, a to na základě dat týkajících se prodejů z předchozích let. Cílem je podpořit tímto obchodní rozhodnutí a dle výsledku s jednotlivými produkty naložit.

Zdrojová data

Jako zdroj slouží následující pohled:

```
use renditedw
go
create view monthsales as --nebo monthsales2007
select convert(datetime,('1.'+cast(MONTH(vdatum) as varchar)+'.'+cast(YEAR(vdatum)
as varchar)))  as datum, count(v.gruppe) as pocet,u.bez
from tg.dbo.verkauf v
left join tg.dbo.gruppen g on v.gruppe=g.gruppe
left join tg.dbo.untergruppen u on g.untergruppe=u.untergruppe
where (u.untergruppe=151010 or u.untergruppe=155010 or u.untergruppe=169995)
and (v.vdatum < '1.1.2007 0:00:00') --nebo >= '1.1.2007'
group by convert(datetime,('1.'+cast(MONTH(vdatum) as varchar)
+'.'+cast(YEAR(vdatum) as varchar))),u.bez
```

Místo konkrétních tří produktů jsou vybrány všechny produkty z tabulky VERKAUF tří podkategorií, které se řadí mezi nejprodávanější a současně prodávané většinou v delším časovém horizontu. Důvodem proč nejsou vybrány konkrétní produkty je jejich krátká životnost, protože např. konkrétní model mobilního telefonu se prodává krátký čas, a proto je využito kategorií, do kterých totožné produkty avšak s jiným názvem spadají. První pohled monthsales zahrnuje interval od 2005-2007, druhý pohled monthsales2007 interval 2007-2012 a poslouží pouze jako ověření predikce. Vybranými sloupci jsou počet prodaných kusů, název podkategorie a měsíc a rok do kterého prodej spadá. Algoritmus Microsoft Time Series vyžaduje sloupec typu datetime, a proto jsou jednotlivé prodeje za měsíc seskupeny dle měsíců a roků (den je reprezentován číslem 1 a nehraje žádnou roli).



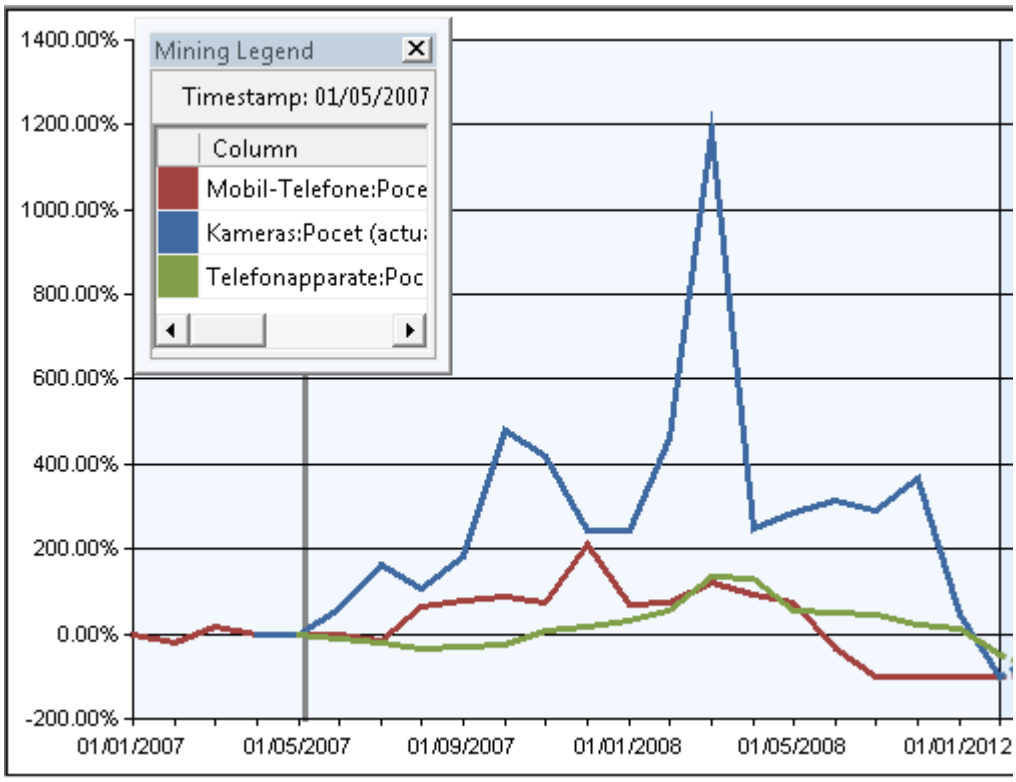
Ilustrace 34: Predikce od roku 2007.

Postup při tvorbě modelu je totožný s předchozími případy, až na algoritmus, kterým je Microsoft Time Series. Jako klíčové sloupce jsou nastaveny datum a identifikační číslo. Atribut počet je vstupní a zároveň predikovaný. Při sestavení modelů je potřeba nastavit chybějící standardní hodnoty v měsících, kdy nebyly provedeny žádné prodeje vybraných produktů. K tomu slouží položka MISSING_VALUE_SUBSTITUTION v nastavení algoritmu modelu se zvolenou hodnotou 0.

Pro prezentaci predikce poslouží dva grafy. První (na obrázku Ilustrace 32) představuje prodeje grafů od 1.1.2005 do 1.1.2007, přičemž následující časové období je predikováno (počet predikčních kroků lze do určité míry zvyšovat, avšak s každým krokem se predikce stává méně přesnou). Ilustrace 34 představuje skutečné hodnoty od roku 2007.

Výsledek skutečné křivky od roku 2007 nesporně ovlivňuje fakt, že pro Telefonapparate neexistují žádná data(prodeje) pro měsíce 1-5. Ostatní dvě kategorie na skutečných datech velmi zhruba naznačují obdobný vývoj jaký je předpokládán, to znamená nárůst prodeje Kameras a kolísání Mobil-Telefone.

Pro praktické využití poslouží návrhář (viz Ilustrace 35), kde je potřeba vybrat jako predikční funkci PredictTimeSeries a výsledný dotaz aplikačně použít s potřebnými parametry.



Ilustrace 35: Skutečný prodej od roku 2007

5 Zhodnocení

Obsahem této kapitoly je najít plusy a mínusy jednotlivých řešení a napomoci k rozhodování nad tím, zda implementovat celé řešení BI, jeho části, nebo neimplementovat vůbec. Z pohledu trendů lze rovnou říci, že BI přinášejí jednoznačně užitek a následující informace by měly napovědět, zda to lze říci i o implementaci do systému Rendite. V podkapitolách jsou zmiňovány jednotlivé části BI pospolu a nejsou dále děleny na samostatná řešení jako datový sklad nebo dolování dat. Naopak je uvedeno dělení na možné kategorie dle různých typů aktérů.

5.1 Možnosti prezentace a uživatelská přívětivost

V této kapitole bude obsahem z části subjektivní hodnocení či porovnání vizuálních vlastností obou řešení. Subjektivní znamená moje osobní a objeví se zde prostor pro případný nesouhlas s individuální pozitivní či negativní kritikou. Ilustrace 2 ukazuje výstup analytického manažeru v Rendite. Z uživatelského hlediska lze uvést dvě hlavní citelné nevýhody a tou je nemožnost využít seskupení jednotlivých větví stromu a jejich následné rozbalení se zaměřením na ty nejdůležitější. V případě většího množství řádek s mnoha sloupci je dohledání konkrétních položek obtížné. Buď je možné použít filtry, anebo vybrat pouze kritické sloupce, což ale působí poněkud těžkopádně. Principiálně lze vizuální podobu výstupu analýzy v analytickém manažeru Rendite (dále AMR) srovnat s reporty v MSSQL, které však mohou být interaktivní (podporují drilldown). Co se týká prohlížení kostek, v případě řešení MSSQL je možno použít prohlížeč kostek ve Visual Studio 2008, MSSQL manažeru nebo Excelu, případně využít komponentu třetí strany a zakomponovat ji do Rendite.

Miscrosoft BI řešení poskytuje SSRS, ovšem jednotlivé reporty je potřeba připravit. Výhodou obsluhování a zobrazení výstupu v AMR pak může být celkem intuitivní ovládání (identické s ovládáním jiných formulářů v Rendite) realizovatelné takřka ihned bez hlubších speciálních znalostí.

Slabou stránkou je beze sporu tisk, neboť AMR sice tisk podporuje, avšak styl a kvalita je ne úplně přijatelná. Na druhou stranu je potřeba říci, že SSRS je možné využít v případě BI stejně dobře jako v případě AMR, neboť zdrojem vytvářených reportů může být téměř jakýkoli zdroj.

Do uživatelské (ne)přívětivosti bych zařadil taktéž nutnost přepočítávat výstup v AMR v případě vložení nové dimenze, neboť je výstup znovu generován (netýká se rychlého přepočítání, které je použito v případě úpravy filtrů). To je v případě prohlížeče OLAP kostek téměř (v závislosti

na objemu dat, kvalitě navržení kostky či použitého prohlížeče) okamžité.

5.2 Technické možnosti jednotlivých řešení

Co se týká současného stavu, AMR neobsahuje žádné statistické metody, které by sloužily k rozsáhlým analýzám. Teoreticky by bylo možné chybějící prvky doprogramovat či použít nástroje třetích stran, ovšem to už se týká dalších kapitol v této sekci. Lze dle mého názoru říci, že sumovací a agregační schopnosti obou řešení jsou na přibližně stejné úrovni.

Zásadní otázkou je potřeba shromažďovat data z více databází. Zatím toto není potřeba, avšak jakmile by vznikl takový požadavek, vše by hovořilo pro tvorbu datového skladu, sloužící jako centrální úložiště. Pak by bylo nutné udělat razantní zásah do AMR. V současnosti je tendence mít jednu hlavní databázi pro všechny pobočky (i přes možné výpadky internetového připojení). Dalším souvisejícím aspektem je splnění homogenního prostředí. V opačném případě by byl problém shromažďovat data k analýze z více různých prostředí.

Systém Rendite uchovává historii zpětně přímo v transakční databázi. To by mělo mít negativní vliv na výkon, avšak předpokládá se, že spolu s rostoucími daty vzrůstá i výkon hardware.

5.3 Výkonnost

V části o prezentačních schopnostech byla zmíněna nutnost přepočítávání (online dotaz do transakční databáze Rendite) pro získání výsledků dle navolených parametrů v AMR. Komplexní dotazy jsou počítány delší dobu, což je v případě OLAP eliminováno. Výsledek průměrného stavu jednotlivých produktů na skladu je počítán v databázi RenditeTest v řádu sekund, avšak u Lagerbestandalter je to již v řádu minut pro interval jeden rok na nezatížené databázi.

Negativním faktorem spouštění analýz na transakční databázi je její zatížení. To je eliminováno vytvořením kopie transakční databáze, na níž se pak analýzy spouští. Je otázka, jaké analýzy jsou vyžadovány a jak často by měla být kopie vytvářena, a je-li takový postup vhodný či dostačující.

5.4 Náročnost a cena jednotlivých řešení

Cena řešení je přespříliš obecné spojení, než aby se nedalo rozčlenit na několik úhlů pohledu. V kapitole 5.2 lze dohledat cenu za rychlost zobrazení výsledku u OLAP a tou je velikost

úložiště pro předpřipravená data. Zatímco transakční velikost databáze Rendite se v průměru pohybuje v řádu jednotek až desítek GB, pak velikost datového skladu se pohybuje minimálně o řád výše. Předpokládám, že bude datový sklad přijímat data pouze z jedné databáze Rendite. Počet předpřipravených tabulek generovaných analýz (typu Lagerwertdurch) bude v řádu desítek. Konkrétní tabulka factLagerwertdurch, obsahující data v časovém úseku jen tří měsíců využívá 490MB prostoru na pevném disku. Pokud by nebylo nalezeno úspornější řešení, pak by byl nárůst požadavků na prostor extrémní. I přesto, že v dnešní době jsou ceny za pevný disk nižší než před několika lety.

Do kategorie náročnosti se řadí proces vytváření tabulek datového skladu. Většina je časově zanedbatelná, avšak v případě factLagerwertdurch byla doba vytváření kolem hodiny a půl – jak bylo zmíněno – pro záznamy tří měsíců. Pro každý rok v různých analýzách by se jednalo o poměrně dlouhou dobu a bylo by potřeba toto brát v potaz.

Nespornou (ne)výhodou stávajícího systému je jeho samotná existence. Pokud by vznikl požadavek na implementaci rozbalování stromu vybraných kategorií (drilldown), cena by se pohybovala přibližně v relaci dvou „člověkotýdnů“. V případě nutnosti implementace výraznějších změn by byla nutná kompletní přestavba vyžadující několik měsíců.

6 Závěr

Tato práce byla vytvořena jako podpora pro rozhodování vedení firmy Český software s.r.o., zda implementovat analytické databáze jako celek, jednotlivé komponenty, anebo vůbec. Realizovat celý proces implementace BI a na základě toho najít výhody a nevýhody stávajícího řešení a individuálně je posoudit.

Prvním krokem byla analýza systému Rendite z uživatelského i vývojářského pohledu, konkrétně jaké poskytuje nástroje pro podporu rozhodování, a jak snadné a efektivní je tyto nástroje používat v praxi. Následoval výběr řešení BI a prozkoumání komponent, ze kterých se skládá. Zejména datový sklad, OLAP databáze a dolování dat. V závislosti na zdrojových datech nejpoužívanějších analýz v analytickém manažeru Rendite jsem realizoval datový sklad včetně procesu ETL. Dalším krokem bylo navržení a vytvoření OLAP kostek z datového skladu. Při jejich tvorbě byl důraz kladen zejména na to, jak lze zobrazit data různých časových intervalů, což analytický manažer Rendite umožňuje. Dále jsem použil data mining při hledání vzorů v datech Rendite v několika typických úlohách.

Z analýzy, její implementace a porovnání vyplývá, že v současného chvíli a za současného stavu není nutné využít kompletních služeb BI. Služby dolování dat by byly zcela jistě přínosem. Analytické nástroje v Rendite jsou i přes některé nedostatky z praktického hlediska dostačující a uživateli jsou schopny se svými předdefinovanými analýzami nabídnout plnohodnotnou podporu pro rozhodování, zvláště pak po nezávislé implementaci dolování dat.

Je velmi důležité, aby do procesu rozhodování byly zapojeny všechny zainteresované strany. Faktorů, dle kterých lze rozhodnout, zda použít BI či nikoli, je celá řada. Architektura u všech zákazníků je v současné chvíli prezentována jednou databází Rendite, k níž jsou připojeni klienti. Pokud by nastala situace, že by měl zákazník více databází Rendite nebo data systémů třetích stran a vyžadoval je k analýzám, pak by využití datového skladu bylo přímo nutností.

Závěrem lze říci, že pokud by se v budoucnosti změnily požadavky zákazníků nebo nastavená pravidla implementace a využívání systému Rendite, pak nezbývá než doporučit BI řešení k využití.

Přehled zkratek

BI – Business Intelligence

OLAP – Online Analytical Processing

OLTP – Online Transaction Processing

SSAS – SQL Server Analysis Services

SSIS – Integrační služby SQL Server 2008

SSRS – SQL Server Reporting Services

MSSQL – Microsoft SQL Server

ETL – Extraction, Transformation, Loading

BIDS – Business Intelligence Development Studio

DM – Dolování dat (Data mining)

Zdroje

Literatura:

[4] Lacko, L. – Datové sklady a analýza OLAP a dolování dat. Brno: Computer Press, 2003. ISBN 80-7226-969-0.

[5] Reeves L. – A Manager's Guide to: Data Warehousing. USA: Wiley Publishing, 2009. ISBN 978-0-470-17638-2.

[6] Kouba Z. – Datové sklady, Dobývání znalostí z databází 2000, Sborník přednášek, FIS VŠE Praha (28.12.2011)

[9] Hulová, H. – Aplikace vybraných metod prostorového dolování dat v databázových systémech (Diplomová práce). ZČU v Plzni. 2010.

[10] Vajgant, J. – Business warehouse řešení pro HP Service Desk (Diplomová práce). ZČU v Plzni. 2007.

[12] Lacko, L. – Business Intelligence v SQL Serveru 2008. Brno: Computer Press, 2009. ISBN 978-80-251-2887-9.

[19] Webb, Ch. – Ferrari, A. – Russo, M. – Expert Cube Development with Microsoft SQL Server 2008 Analysis services. Birmingham: Packt Publishing, 2009. ISBN 978-1-847197-22-1.

[20] Han, J. – Kamber, M. – Datamining: Concepts and Techniques. 2. vydání. USA:Morgan Kaufmann Publishers. 2005. ISBN 978-1-55860-901-3.

[22] Ing. Josef Weinreb, CSc. – Přednášky KIV/PSDS. 2012

Internet:

[1] Český software s.r.o., www.ceskysoftware.cz (10.12.2011)

[2] H.I.W. 24, www.hiw24.de (10.12.2011)

[3] System online, http://www.systemonline.cz/clanky/co-je-to-business-intelligence.htm (28.12.2011)

[7] SAP, http://wiki.sdn.sap.com/wiki/display/BI/OLAP+vs+OLTP (29.12.2011)

[8] System online,http://www.systemonline.cz/clanky/dva-zpusoby-budovani-datoveho-skladu.htm (10.1.2012)

[11] Microsoft MSDN, http://msdn.microsoft.com/en-us/library/ms175646.aspx (12.1.2012)

[13] System online, http://www.systemonline.cz/clanky/dva-zpusoby-budovani-datoveho-skladu.htm (28.11.2011)

[15] Oracle, http://www.oracle.com/technetwork/database/enterprise-edition/odm-business-solutions-084229.html (4.1.2012)

[16] Microsoft MSDN, www.msdn.com (3.1.2012)

[17] Oracle, www.oracle.com (27.12.2011)

[18] Doc. Ing. Petr Berka, CSc., http://sorry.vse.cz/~berka/docs/izi456/sl-idt.pdf (18.11.2011)

[21] Databáze AdventureWorks 2008 DW, Microsoft, http://msftdbprodsamples.codeplex.com/

Seznam ilustrací

Přílohy

Příloha CD : Diplomová práce ve formátu PDF

Neořízнuté obrázky

SSIS projekt

SSAS projekt

SQL skripty