1.2 Metodiky

S postupem doby začaly vznikat metodiky, které si kladou za cíl poskytnout uživatelům jednotný rámec pro řešení různých úloh z oblasti dobývání znalostí. Tyto metodiky umožňují sdílet a přenášet zkušenosti z úspěšných projektů. Za některými metodikami stojí producenti programových systémů (metodika *5A* firmy SPSS nebo metodika *SEMMA* firmy SAS, obr. 9), jiné vznikají ve spolupráci výzkumných a komerčních institucí jako „softwarově nezávislé" (*CRISP-DM*).

1.2.1 Metodika 5A

Metodiku 5A nabízí firma SPSS jako svůj pohled na proces dobývání znalostí. Název metodiky je akronymem pro jednotlivé prováděné kroky (5A, 2000):

- *Assess* – posouzení potřeb projektu,
- *Access* – shromáždění potřebných dat,
- *Analyze* – provedení analýz,
- *Akt* – přeměna znalostí na akční znalosti,
- *Automate* – převedení výsledků analýzy do praxe.

Žádná data nemají význam, jestliže jsou oddělena od kontextu. Prvním krokem v analytickém procesu je tedy stanovení kontextu – cílů, strategií a procesů. Materiál SPSS (SPSS, 2000) k tomu říká:

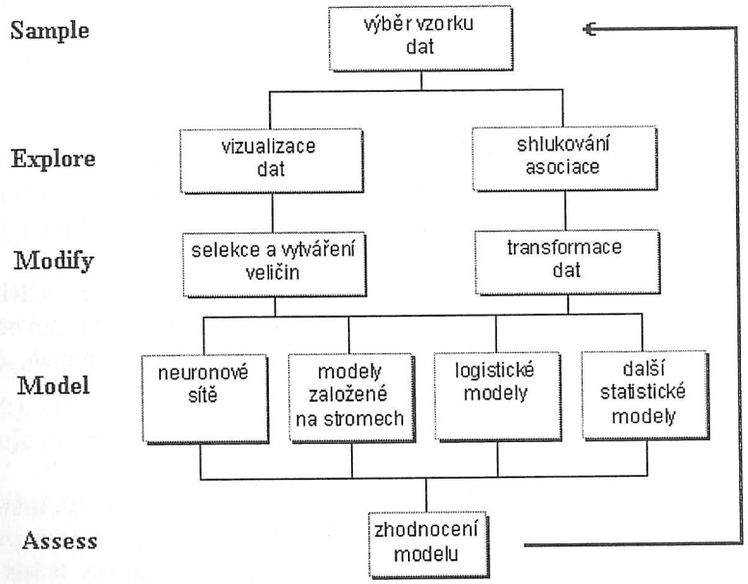
- Určete data, jejichž sběr, pořízení a skladování je nutné zajistit k provedení takových analýz, které chcete realizovat.
- Připravte se na své projekty a obory, v nichž rozhodujete – jejich porozuměním zabezpečte ty analytické nástroje, které potřebujete.
- Vzdělávejte a trénujte všechny lidi, kteří myslí analyticky a používají efektivně software jako součást přemýšlení nad problémy a analýzu dat jako nedílnou složku rozhodovacího procesu.

Druhým krokem v metodice 5A je sběr a příprava dat. Je třeba získat vhodné soubory z podnikových datových skladů, datových bází, odkazových systémů a jiných interních zdrojů. Lze využít i data týkající se daného problému, která jsou nabízena veřejně (oficiální statistiky, rezortní data, demografické a psychografické údaje apod.). Data lze rovněž získat vlastními průzkumy nebo od výzkumné firmy.

Třetím krokem je používání různých analytických postupů k tomu, abychom nalezli odpovědi na otázky stanovené v prvním kroku. V tomto kroku se data přeměňují na informace a znalosti. Firma SPSS doporučuje širokou škálu nástrojů pro zkoumání a porozumění datům počínaje deskriptivní statistikou, přes metodu OLAP až po metody strojového učení (rozhodovací stromy, neuronové sítě). Doporučení je zřejmé: *„Použijte více metod a porovnejte jejich výsledky a vhodnost, abyste získali nejlepší řešení a navíc rychle a jednoduše"*.

Čtvrtý krok procesu obsahuje doporučení, řadu dodatečných otázek a následné rozhodnutí. Znalosti nalezené v předcházejícím kroku se zde mění na znalosti akční. Nalezené výsledky by měly být předkládány v jasné a srozumitelné podobě.

Pátým krokem je převedení výsledků analýzy do praxe. Tento krok obsahuje všechny činnosti, kterými lze zajistit aplikaci učiněných rozhodnutí. Sem patří například to, že vytvoříme praktické rozhraní, abychom rozvinuli nalezené modely do takového formátu, který je snadný pro užívání a porozumění v běžné a opakované praxi organizace a pro monitorování výsledků (a důsledků) prováděných rozhodnutí. Další z doporučení zní *„Automatizujte své analýzy tak, aby opakující se úlohy nezabíraly čas a abyste mohli snadno aktualizovat své modely s tím, jak přicházejí nové výsledky"*.



Obr. 9 Metodika SEMMA.

1.2.2 Metodika SEMMA

Enterprise Miner, softwarový produkt firmy SAS, vychází z vlastní metodiky pro dobývání znalostí z databází. Název SEMMA opět charakterizuje jednotlivé prováděné kroky:

- *Sample* (vybrání vhodných objektů),
- *Explore* (vizuální explorace a redukce dat),
- *Modify* (seskupování objektů a hodnot atributů, datové transformace),

- *Model* (analýza dat: neuronové sítě, rozhodovací stromy, statistické techniky, asociace a shlukování),
- *Assess* (porovnání modelů a interpretace).

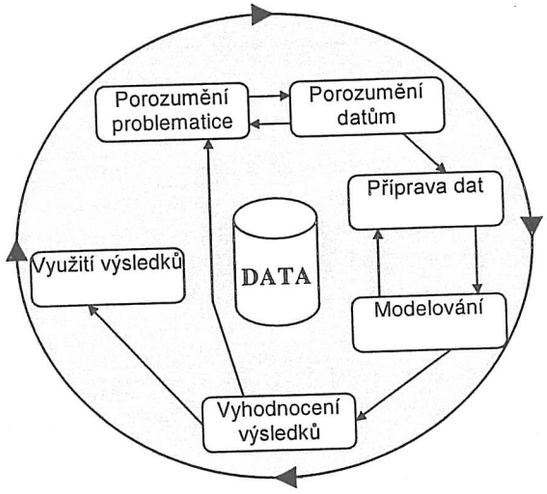
Důraz se klade na snadnou interpretaci výstupů ve formě srozumitelné obchodnímu uživateli.

1.2.3 Metodika CRISP-DM

Metodika *CRISP-DM* (CRoss-Industry Standard Process for Data Mining) vznikla v rámci Evropského výzkumného projektu. Cílem projektu bylo navrhnout univerzální postup (tzv. standardní model procesu dobývání znalostí z databází), který bude použitelný v nejrůznějších komerčních aplikacích (Chapman a kol, 2000). Vytvoření takovéto metodiky umožní řešit rozsáhlé úlohy dobývání znalostí rychleji, efektivněji, spolehlivěji a s nižšími náklady. Kromě návrhu standardního postupu má CRISP-DM nabízet „průvodce" potenciálními problémy a řešeními, které se mohou vyskytnout v reálných aplikacích.

Na projektu spolupracovaly firmy NCR (přední dodavatel datových skladů), DaimlerChrysler, ISL (tvůrce systému *Clementine*) a OHRA (velká holandská pojišťovna). Všechny tyto firmy mají bohaté zkušenosti s reálnými úlohami dobývání znalostí z databází.

Životní cyklus projektu dobývání znalostí je podle metodiky CRISP-DM tvořen šesti fázemi (obr. 10). Pořadí jednotlivých fází není pevně dáno. Výsledek dosažený v jedné fázi ovlivňuje volbu kroků následujících, často je



Obr. 10 Metodika CRISP-DM.

třeba se k některým krokům a fázím vracet. Vnější kruh na obrázku symbolizuje cyklickou povahu procesu dobývání znalostí z databází jako takovou.

Jednotlivé fáze metodiky CRISP-DM bude ilustrovat poměrně reálný příklad úlohy dobývání znalostí z dat v bance XY (Berka, 1999).

Nejprve několik slov k pozadí úlohy. Banka XY je zaměřena na drobné klienty, kterým vede účty, poskytuje půjčky apod. Pod rostoucím tlakem konkurence chce tato banka zlepšit své služby. Management banky má jen velmi vágní představu, co je možné od metod dobývání znalostí očekávat. Doufá ale, že mu tyto nové metody umožní lépe pochopit klienty, a tak například cíleněji nabízet své produkty, nebo rozlišovat mezi různými skupinami klientů (bonitní, resp. problémoví).

Porozumění problematice (Business Understanding)

Tato úvodní fáze je zaměřena na pochopení cílů úlohy a požadavků na řešení formulovaných z manažerského hlediska. Manažerská formulace musí být následně převedena do zadání úlohy pro dobývání znalostí z databází.

Manažerský problém, ke kterému pomocí metod KDD hledáme informace, může být formulován (téměř) bez vazby na informace získávané pomocí metod KDD z dostupných dat. Příkladem může být snaha nabídnout uložení části peněz na zvláštní účet s delší výpovědní lhůtou pomocí reklamy vhodně zacílené na v tomto směru nadějnou skupinu klientů (i potenciálních) banky. Pro KDD to znamená nalézt takovou charakteristiku klientů, která zajišťuje, že ve skupině klientů s touto charakteristikou bude velká část klientů mít stále dostatečně vysoký zůstatek na účtu. V tomto případě je zadání pro KDD formulováno relativně přesně, přesto je však třeba počítat s možností přeformulovat nebo upřesnit manažerský problém na základě provedených analýz. Jinou možnou úlohou je otázka včasného rozpoznání klientů, kteří představují rizikovou skupinu z hlediska splácení poskytnutého úvěru.

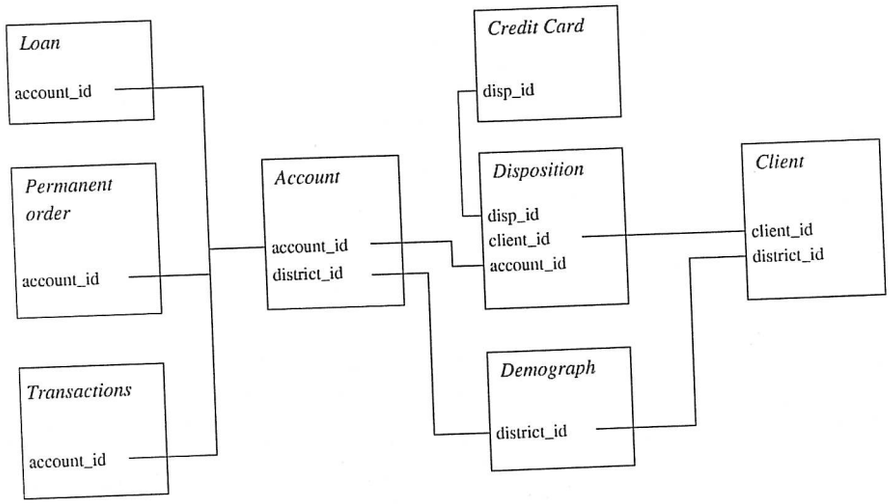
V této fázi se rovněž provádí inventura zdrojů (datových výpočetních i lidských), hodnotí se možná rizika, náklady a přínos použití metod KDD a stanovuje se předběžný plán prací.

Porozumění datům (Data Understanding)

Fáze porozumění datům začíná prvotním sběrem dat. Následují činnosti, které umožní získat základní představu o datech, která jsou k dispozici (posouzení kvality dat, první „vhled" do dat, vytipování zajímavých podmnožin záznamů v databázi...). Obvykle se zjišťují různé deskriptivní charakteristiky dat (četnosti hodnot různých atributů, průměrné hodnoty, minima, maxima apod.), s výhodou se využívají i různé vizualizační techniky.

Data sledovaná bankou XY mají podobu několika navzájem propojených tabulek (obr. 11). Základní tabulkou je tabulka *Account* (účty). S každým účtem může disponovat nějaký klient (tabulka *Client*). K jednomu účtu může mít přístup více klientů, jeden klient může mít zřízeno více účtů; tato skutečnost je zachycena v tabulce *Disposition*, která přiřazuje klienty k účtům. Klientovi, který disponuje nějakým účtem, může být k tomuto účtu vydána kreditní karta (tabulka *Credit Card*). Nejdůležitějšími údaji o účtech jsou údaje o prováděných operacích, to je zachyceno v tabulce *Transactions* (transakce). Na některých účtech mohou být zřízeny trvalé platební příkazy (tabulka *Permanent order*), na základě některých účtů banka poskytuje úvěr (tabulka *Loan*).

Banka poskytla pro analýzu jen určitý (relativně malý) vzorek těchto dat; první představu o podobě dat tedy bylo možno získat relativně jednoduchými nástroji (Access, Excel). Bylo konstatováno, že některé údaje v tabulce transakce (např. konstantní symbol) mají mnoho chybějících hodnot.



Obr. 11 Data banky XY.

Příprava dat (Data Preparation)

Příprava dat zahrnuje činnosti, které vedou k vytvoření datového souboru, který bude zpracováván jednotlivými analytickými metodami. Tato data by tedy měla

- obsahovat údaje význačné pro danou úlohu,
- mít podobu, která je vyžadována vlastními analytickými algoritmy.

Příprava dat tedy zahrnuje selekci dat, čištění dat, transformaci dat, vytváření dat, integrování dat a formátování dat. Tato fáze je obvykle nejpracnější částí řešení celé úlohy. Jednotlivé úkony jsou obvykle prováděny opakovaně, v nejrůznějším pořadí.

Vzhledem k tomu, že data o klientech a jejich účtech jsou uložena v několika tabulkách navzájem spojených relacemi $1:n$, $n:1$ a $n:m$, velkou část předzpracování představovalo vytvoření jediné tabulky obsahující údaje vybrané z více tabulek. Příslušné operace vedoucí k tomuto cíli tedy zahrnují agregování hodnot odpovídajících jednomu klientovi. Dalšími operacemi bylo například vypočtení průměrných měsíčních zůstatků, průměrných měsíčních příjmů, převod rodného čísla klienta na jeho věk a pohlaví apod.

Modelování (Modeling)

V této fázi jsou nasazeny analytické metody (algoritmy pro dobývání znalostí). Obvykle existuje řada různých metod pro řešení dané úlohy, je tedy třeba vybrat ty nejvhodnější (doporučuje se použít více různých metod a jejich výsledky kombinovat) a vhodně nastavit jejich parametry. Jde tedy opět o iterativní

činnost (opakovaná aplikace algoritmů s různými parametry), použití analytických algoritmů může navíc vést k potřebě modifikovat data, a tedy k návratu k datovým transformacím z předcházející fáze.

Pro hledání zajímavých skupin klientů je možné použít metody shlukování nebo asociační pravidla. Pro rozpoznání rizikových klientů z hlediska půjček jsou (vzhledem k tomu, že jedna z tabulek obsahuje informace o průběhu splácení) vhodné například algoritmy pro tvorbu rozhodovacích stromů nebo rozhodovacích pravidel. Tyto metody je vhodné kombinovat, například shluková analýza může rozdělit klienty do skupin a rozhodovací strom pak umožní jednotlivé skupiny charakterizovat dostatečně srozumitelným způsobem.

Součástí této fáze je rovněž ověřování nalezených znalostí z pohledu metod dobývání znalostí. To může představovat např. testování klasifikačních znalostí na nezávislých datech.

Znalosti „deskriptivní" (charakteristiky skupiny klientů „zajímavých" z hlediska připravovaného produktu) byly předloženy expertům z banky. Znalosti klasifikační (umožňující „rozpoznat" klienty, kteří nesplácejí úvěr) byly testovány na novém vzorku dat.

Vyhodnocení výsledků (Evaluation)

V této fázi jsme se dopracovali do stavu, kdy jsme nalezli znalosti, které se zdají být v pořádku z hlediska metod dobývání znalostí. Dosažené výsledky je ale ještě třeba vyhodnotit z pohledu manažerů, zda byly splněny cíle formulované při zadání úlohy.

Některé nalezené skupiny klientů experty nepřekvapily, vědělo se o nich a banka se připravovala je oslovit dopisem. Jiné (rovněž bonitní skupiny) byly shledány zajímavými, ale budou ještě podrobeny dalšímu zkoumání. Výsledky testování klasifikačních znalostí ukázaly, že systém byl příliš „přísný", tedy správně rozpoznával klienty rizikové, ale v určitých případech (obzvláště u vyšších půjček) za rizikové označil i klienty bonitní. Bylo tedy rozhodnuto, že ve všech pobočkách banky bude využíván program, který bude rozhodovat o úvěrech do určité částky.

Na závěr této fáze by mělo být přijato rozhodnutí o způsobu využití výsledků.

Využití výsledků (Deployment)

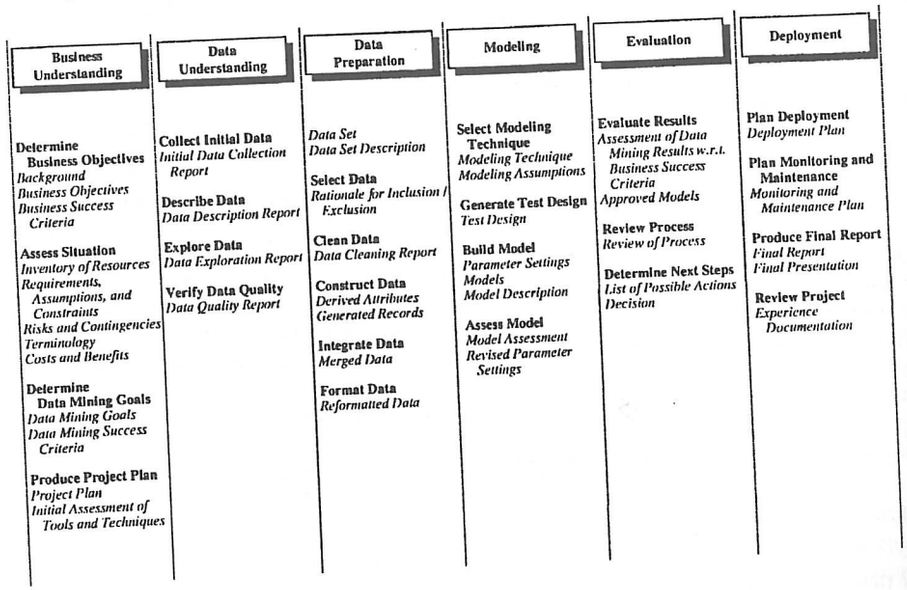
Vytvořením vhodného modelu řešení úlohy obecně nekončí. Dokonce i v případě, že řešenou úlohou byl pouze popis dat, je třeba získané znalosti upravit do podoby použitelné pro zákazníka (manažera – zadavatele úlohy). Podle typu úlohy tedy využití (nasazení) výsledků může na jedné straně znamenat prosté sepsání závěrečné zprávy, na straně druhé pak zavedení (hardwarové, softwarové, organizační) systému pro automatickou klasifikaci nových případů.

Ve většině případů je to zákazník a nikoliv analytik, kdo provádí kroky vedoucí k využívání výsledků analýzy. Proto je důležité, aby pochopil, co je nezbytné učinit pro to, aby mohly být dosažené výsledky využívány efektivně.

Systém pro rozhodování o půjčkách bude nasazen ve dvou fázích. V první fázi bude systém dán jen do vybraných poboček, po tomto poloprovozním ověření pak bude využíván všude. Toto rozhodnutí vyžaduje:

- implementaci klasifikačního algoritmu v uživatelsky přátelské podobě,
- přípravu uživatelského manuálu,
- instalaci programu ve všech pobočkách banky,
- zaškolení uživatelů,
- změnu metodiky poskytování úvěrů a tomu odpovídající změnu vnitřních předpisů banky.

Celkové schéma jednotlivých kroků metodiky CRISP-DM ukazuje obr. 12. Jednotlivé kroky procesu dobývání znalostí jsou různě časově náročné a mají i různou důležitost pro úspěšné vyřešení dané úlohy. Praktici v oboru uvádějí[6], že nejdůležitější je fáze porozumění problému (80 % významu, 20 % času) a časově nejnáročnější je fáze přípravy dat (80 % času, 20 % významu). Překvapivě málo práce zaberou vlastní analýzy (5 % času, 5 % významu).



Obr. 12 Úlohy v metodice CRISP-DM (Chapman a kol. 2002).

Literatura

Anand S. a kol.: *Towards Real-World Data Mining*. In: Practical Aspects of Knowledge Management, Schweizer Informatiker Gesellschaft, Basel, 1996.

[6] Následující údaje jsou převzaty z vystoupení J.U. Kietze na konferenci PKDD 2000.

Berka P.: *Guide to Financial Data Set*. In: Workshop notes on Discovery Challenge, PKDD'99, 1999.

Fayyad U., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. eds: *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, 1996.

Chapman P., Clinton J., Kerber R., Khabaza T., Reinartz T., Shearer C., Wirth R.: *CRISP-DM 1.0 Step-by-step data mining guide*. SPSS Inc. 2000.

Klosgen W., Zytkow J.: *Knowledge Discovery and Data Mining*. Tutorial Notes. PKDD'97, Trondheim 1997.

Rauch J.: *Získávání znalostí z databází*. Podklady pro posluchače semináře IISEM 491 v letním semestru 1998/99, VŠE, 1999.

SASS: SEMMA. Internet: http://www.sas.com/products/miner/semma.html, 2002.

SPSS: Metoda 5A. Internet: http://www.spss.cz/datamin_jak.html, 2000.

II. TŘI ZDROJE

2 Databáze

2.1 Relační databáze

V prehistorii databází byla data ukládána v jednom velkém „plochém“ souboru (tzv. flat file), ke kterému se přistupovalo indexovanými sekvenčními metodami (ISAM). Soubor byl indexován na základě předpokládaných způsobů dotazování. Nevýhodou bylo, že se informace v záznamech opakovaly (viz tab. 1), další nevýhodou bylo předurčení typu dotazů (daných dopředu zvoleným způsobem indexování).

Tabulka 1 Plochý soubor s daty

datum	jmeno	prijmeni	adresa_ulice	adresa_mesto	cislo_uctu	platba	zustatek
980103	Jan	Novak	Dlouha 5	Praha 1	9945371	100,00	100,00
980105	Jan	Novak	Dlouha 5	Praha 1	9945371	1500,00	1600,00
980106	Jan	Novak	Dlouha 5	Praha 1	9945371	–1550,00	50,00
980106	Karel	Nemec	Podolska 4	Praha 2	24867134	3000,00	6000,00
980107	Karel	Nemec	Podolska 4	Praha 2	24867134	–4000,00	2000,00
980108	Jan	Novak	Dlouha 5	Praha 1	9945371	–150,00	–100,00
980111	Karel	Nemec	Podolska 4	Praha 2	24867134	5000,00	7000,00
				...			

Velkým krokem kupředu bylo zavedení relačních databází. Jeden velký datový soubor byl rozdělen do řady relací (tabulek). Relační databáze je tedy tvořena:

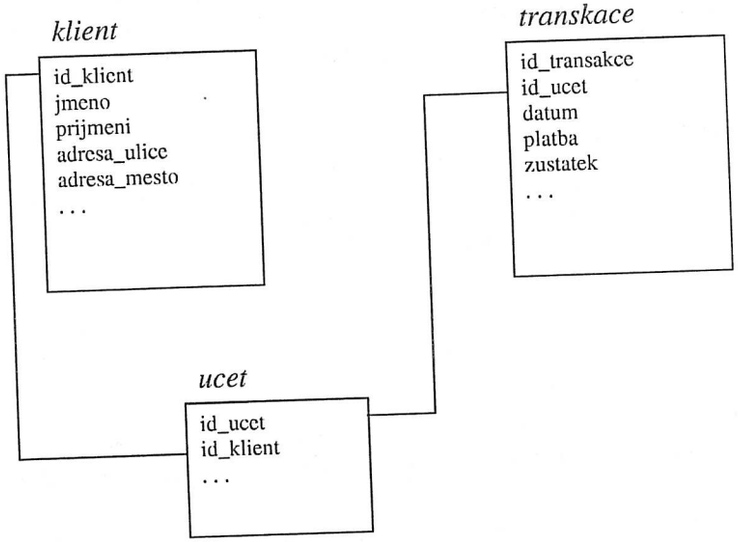
- množinou relací – relace je reprezentována dvourozměrnou tabulkou (řádky odpovídají záznamům, sloupce atributům, jednotlivé záznamy jsou jednoznačně identifikovány pomocí primárního klíče),
- operacemi selekce, projekce a spojení pro manipulaci s tabulkami – selekce slouží k výběru záznamů (řádků tabulky), projekce slouží k výběru atributů (sloupců tabulky) a spojení slouží k propojování tabulek (spojují se řádky se stejnou hodnotou nějakého atributu – obvykle klíče).

Pro kladení dotazů nabízejí relační databáze dva způsoby:

- QBE (query by example),
- SQL (structured query language).

Oba tyto způsoby navrhla v 70. letech minulého století firma IBM. QBE nabízí uživateli relativně jednoduchý, intuitivní způsob kladení dotazů. V předem připraveném formuláři uživatel vyplní (vybere), co ho zajímá, zadá tedy jakousi „masku“ které budou odpovídat nalezené záznamy. Je tedy tento způsob vhodnější pro méně zkušené uživatele. SQL je naopak určen uživatelům

zkušeným. Jde vlastně o jednoduchý programovací jazyk pro definování dat a pro manipulaci s nimi (obr. 14). SQL je daleko mocnější a flexibilnější nástroj než dotazování (vyhledávání) pomocí indexů. Klade však zvýšené nároky na uživatele. Uživatel musí znát syntaxi jazyka, navíc musí znát i detailní strukturu databáze (názvy souborů a polí). Tento způsob dotazování tedy příliš nepřirostl k srdcím manažerů a analytiků. Pokud se nechtěli učit jazyku SQL, byli nuceni připravit dotaz v přirozeném jazyce, zajít za programátorem, který jejich dotaz převedl do jazyka SQL a provedl jej. V případě, že obdržené výsledky nepřinesly požadovanou odpověď, museli dotaz přeformulovat a znovu se vydat do výpočetního centra atd.



Obr. 13 Relační databáze.

```
SELECT klient.jmeno, klient.prijmeni,
       klient.adresa_ulice,
       klient.adresa_mesto, ucet.cislo_uctu,
       transkace.zustatek
FROM   klient, ucet, transakce
WHERE  klient.id_klient  = ucet.id_klient;
AND    transakce.id_ucet = ucet.id_ucet;
AND    transakce.zustatek < 100;
GROUP BY  klient.adresa_mesto
```

Obr. 14 Dotaz v jazyce SQL .

2.2 EIS

EIS (Executive Information Systems) byl první pokus jak přiblížit dotazování do databáze manažerům. Zavádění EIS bylo spojeno se zaváděním osobních počítačů v dané firmě; počítače přestaly být doménou programátorů, objevily se na stolech „prostých" uživatelů. Základním požadavkem se tedy stalo snadné ovládání. Uživatelsky přátelský interface exekutivních informačních systémů „prosté" uživatele odstínil od syntaxe SQL a od nutnosti znát strukturu databáze, se kterou chtěli pracovat. Analýzu tedy mohl analytik provádět sám, z počítače, který měl na svém pracovním stole. Dotaz, který si uživatel vybral v menu, byl pak převeden do jazyka SQL a proveden standardním způsobem. Nevýhodou tohoto přístupu bylo, že uživatel měl k dispozici pouze určitý soubor připravených dotazů. Chtěl-li se zeptat na něco, co tvůrce daného EIS nepředpokládal, byl opět nucen připravit si dotaz v přirozeném jazyce, zajít za programátorem, který dotaz převedl do jazyka SQL.

EIS tedy byly sice uživatelsky přátelské, ale málo flexibilní nástroje pro analýzu dat v databázích.

2.3 OLAP

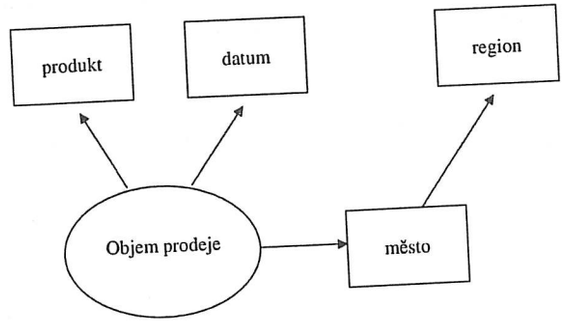
OLAP (On-Line Analytical Processing) konečně nabídl uživatelům obojí; flexibilitu (a rychlost) i příjemné, intuitivní ovládání. OLAP umožňuje analytikům firmy snadno získávat odpovědi na otázky typu „Co má největší vliv na růst tržeb v severních Čechách?" ve velice srozumitelné podobě. Typické pro OLAP jsou totiž možnosti vizualizace. Grafické rozhraní umožňuje uživateli nahlížet na data jak v numerické podobě, tak v podobě nejrůznějších grafů.

Podle E. F. Codda, který v 80. letech 20. století přišel s touto koncepcí, je pro metodu OLAP typické (Thomsen, 2002):

- multidimenzionální koncept uložení i manipulace s daty,
- intuitivní manipulace s daty,
- práce s daty z heterogenních datových zdrojů – provádí se konverze dat,
- použití analytických metod – statistické přehledy, what-if analýzy,
- Client/Server architektura,
- podpora multiuživatelského pohledu,
- ukládání výsledků OLAP mimo zdrojová data,
- dynamická manipulace s řídkými maticemi,
- zpracování chybějících hodnot,
- neomezený počet dimenzí a agregačních úrovní.

Základem OLAP je pohled na data jako na mnohorozměrnou tabulku nazývanou *datová krychle* (data cube). Obrázek 15 ukazuje strukturu jednoduché databáze. Záznamy v databázi obsahují údaje o prodeji různých produktů v různých dnech v různých městech (tab. 2). Tuto databázi můžeme převést na datovou krychli tak, že jednotlivé sledované atributy budou tvořit dimenze krychle, buňky krychle pak odpovídají jednotlivým záznamům v databázi. Tento způsob uložení umožňuje různé pohledy na data (natáčení krychle, provádění řezů) ale plýtvá se při něm místem. Řada buněk v krychli je prázdných[7] (tab. 3).

Datová krychle obsahuje jak data z operačních databází, tak dílčí souhrny (viz obr. 16). Právě tyto souhrny umožňují rychlou odezvu na ad-hoc (nepřed-připravené) dotazy uživatele a flexibilitu systému. Práce s krychlí spočívá



Obr. 15 Struktura databáze.

Tabulka 2 Záznamy v databázi PRODEJ

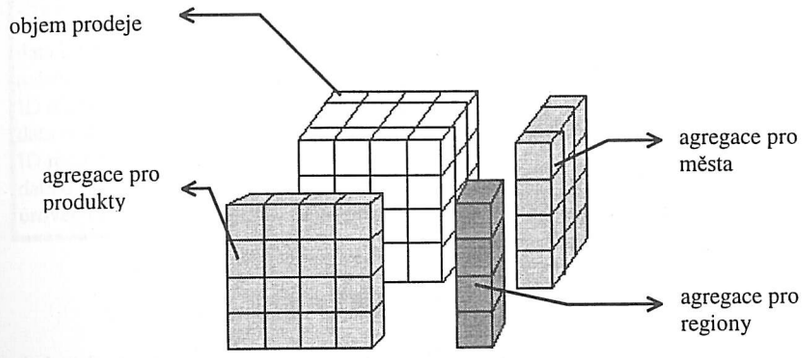
datum	produkt	město	množství
10.1.	šrouby	Praha	241
10.1.	matky	Praha	61
10.1.	šrouby	Brno	17
10.1.	podložky	Brno	42
10.2.	šrouby	Praha	92
10.2.	podložky	Praha	27
10.2.	šrouby	Kladno	35

Tabulka 3 Řídká matice

	Praha			Brno			Kladno		
	šrouby	matky	podložky	šrouby	matky	podložky	šrouby	matky	podložky
10.1	241	61		17		42			
10.2	92		27				35		

[7] Matematika má pro takovou strukturu označení řídká matice.

v různém natáčení (pivot), provádění řezů (slice), výběru určitých částí (dice) a zobrazování různých agregovaných hodnot. Velmi často lze hodnoty atributů sdružovat do hierarchií (v databázi na obr. 16 jsou např. města zařazena do regionů). Úrovní v hierarchii bývá obvykle více; (např. *město* → *okres* → *region* → *země*). Tyto hierarchie se využívají při práci s krychlí při operacích roll-up a drill-down. Při roll-up se přechází na hierarchicky vyšší, obecnější úroveň – zobrazované údaje mají podobu souhrnů, při drill-down se přechází na podrobnější pohled na data; někdy se mluví o různých úrovních podrobnosti (granularity) pohledu na data.



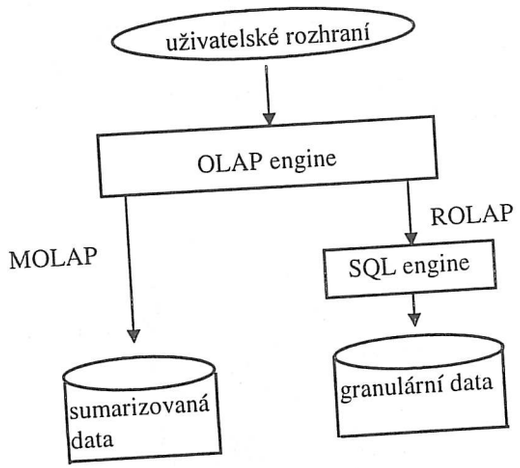
Obr. 16 Multidimenzionální model.

Základní multidimenzionální model má podobu *n*-rozměrné krychle. To je však podoba v logickém smyslu. Z hlediska implementačního se nabízí několik možností jak tuto strukturu uložit v počítači. Hlavní důvod pro odlišné fyzické implementace logického modelu je především velká řídkost dat a jejich nestejnoměrné rozmístění. Pro implementaci se nabízejí v zásadě dva přístupy:

- hyperkrychle (hypercube) – jedna velká krychle, která obsahuje nástroje pro práci s řídkými daty; výhodou je jednoduchá struktura a srozumitelnost pro uživatele
- multikrychle (multicube) – více navzájem propojených menších krychlí obsahujících jen několik dimenzí; výhodou je efektivní uložení dat.

Uložení dat v krychli ale nepřináší jen výhody. Za rychlost přístupu k datům platíme zvýšenými nároky (a tedy i cenou) na datový server. Tyto nároky někdy vedou k tomu, že se místo „čistého" řešení OLAP, založeného na datové krychli (někdy nazývaného MOLAP – multidimenzionální OLAP), použije tzv. ROLAP – relační OLAP založený na klasické relační databázi. V tomto druhém případě se dotazy OLAP převádějí do klasických dotazů SQL (viz obr. 17).

MOLAP se hodí pro středně velké, statické aplikace. Příkladem může být analýza historických dat o prodeji nějakého produktu. Vzhledem k tomu, že výpočty souhrnů vyžadují jistý čas, není MOLAP vhodný pro dynamické aplikace, kde jsou požadovány informace z průběžně aktualizovaných dat.



Obr. 17 Porovnání přístupu MOLAP a ROLAP.

ROLAP je vhodný pro rozsáhlé aplikace hojně využívající transakční data. Výhodou je schopnost zpracovávat rozsáhlá data za použití existujících databázových technologií. Nepoužívá se ale příliš pro obchodní nebo finanční aplikace. Podobně jako u MOLAP, jsou i zde různé možnosti fyzické implementace systému:

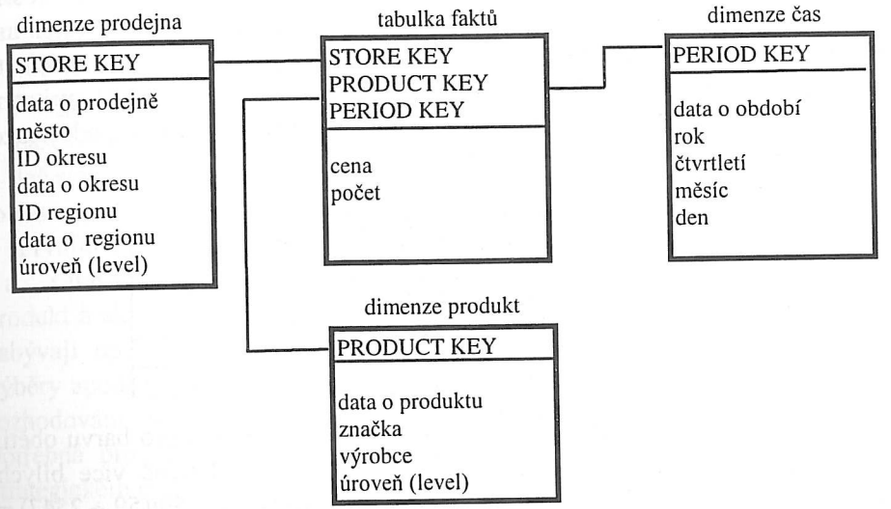
- schéma hvězdy (star schema),
- schéma sněhové vločky (snowflake schema).

Rozdíl mezi oběma přístupy ilustruje opět příklad evidence prodeje nějakých výrobků. Naše databáze má opět tři dimenze: *prodejna*, *produkt* a *čas*. Dimenze prodejen je tvořena hierarchií *obchod* → *okres* → *region*, dimenze produktu je tvořena hierarchií *výrobek* → *značka* → *výrobce*, a časová dimenze je tvořena hierarchií *datum* → *měsíc* → *čtvrtletí* → *rok* (dbms, 2000).

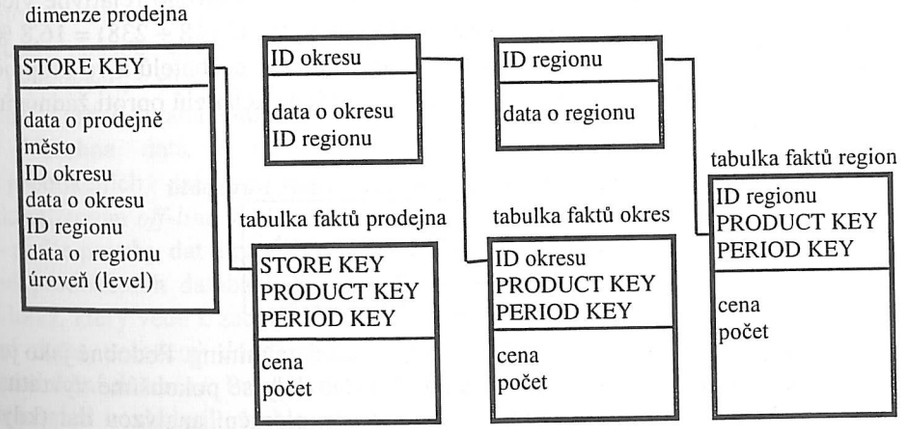
Schéma hvězdy (obr. 18) vychází z jedné centrální tabulky – *tabulky faktů* (fact table), která obsahuje složený primární klíč (jeden segment klíče pro každou dimenzi) a detailní data (objem prodeje daného výrobku v daném obchodu za dané období); může rovněž obsahovat data agregovaná. Pro každou dimenzi existuje jedna tabulka, která obsahuje údaje na různé úrovni příslušné hierarchie, tzv. *tabulka dimenzí*. Úroveň v hierarchii (level) se rovněž zaznamenává jako další indikátor do tabulky dimenzí; je totiž třeba mít přehled o tom, jaké úrovně granularity se konkrétní záznam týká. Indikátor úrovně je

tedy nutný při dotazování do tabulky, která obsahuje současně data detailní i agregovaná. Výhodou hvězdy je její srozumitelnost, snadné definování hierarchií, jednoduchá metadata a rychlost přístupu k datům. Nevýhodou jsou problémy s velkými tabulkami dimenzí a předpoklad, že pracujeme pouze se statickými daty, která nejsou aktualizována *on-line*.

Alternativou ke schématu hvězdy je schéma sněhové vločky. Zde se pracuje s normalizovanými tabulkami dimenzí tak, že každá tabulka nějaké dimenze ukazuje na odpovídající agregovanou tabulku faktů. Tabulky dimenzí obsahují



Obr. 18 Hvězda.



Obr. 19 Sněhová vločka.

jediný primární klíč pro danou úroveň dimenze spolu s odkazem na nejbližšího rodiče v hierarchii dimenzí. Obrázek 19 ukazuje takto upravenou dimenzi prodejen. Při použití sněhové vločky odpadá nutnost používat indikátor úrovně v hierarchii; v každé tabulce jsou údaje jen z jedné úrovně. Výhody této koncepce se projeví ve chvílích, kdy dotazy do databáze se týkají agregovaných hodnot. Nevýhodou je složitá údržba a nárůst počtu tabulek v databázi.

Snadnost použití OLAP v sobě skrývá nebezpečí chybné interpretace dat způsobené nesprávným zobecněním závěrů při přechodu mezi jednotlivými úrovněmi granularity. Na dílčí souhrny uvedené v procentech (relativní četnosti) totiž nelze použít běžné aritmetické operace, což si ne každý uživatel asi uvědomí. Na tento paradox relativních četností upozorňuje (v souvislosti s pravděpodobnostními modely závislostí mezi daty) S. Lauritzen (Lauritzen, 1996). Ve svém příkladu použil data o rozsudcích vynesených v případě vražd na Floridě v letech 1973–79 (viz tab. 4).

Tabulka 4 Podrobný přehled vražd na Floridě

	oběť běloch		oběť černoch	
	pachatel běloch	pachatel černoch	pachatel běloch	pachatel černoch
rozsudek smrt	72	48	0	11
jiný rozsudek	2074	238	111	2309

Provedeme-li roll-up (přechod na méně podrobnou úroveň) pro barvu oběti, můžeme dospět k závěru (tab. 5), že bylo popraveno relativně více bílých pachatelů (72/(72 + 2185) = 3,2 %) než černých pachatelů (59/(59 + 2547) = 2,3 %). Při pohledu do původní tabulky 4 ale zjistíme, že jak v případě, že obětí byl běloch, tak v případě, že obětí byl černoch, bylo popraveno relativně více černých pachatelů. Ve skupině bílých obětí to bylo 48/(48 + 238) = 16,8 % černých pachatelů oproti 72/(72 + 2074) = 3,4 % bílých pachatelů a ve skupině černých obětí to bylo 11/(11 + 2309) = 0,5 % černých pachatelů oproti žádnému bílému pachateli.

Tabulka 5 Přehled vražd na Floridě bez vztahu k barvě oběti

	pachatel běloch	pachatel černoch
rozsudek smrt	72	59
jiný rozsudek	2185	2547

Nutno ještě podotknout, že OLAP není totéž co data mining. Podobně jako je ve statistice rozdíl mezi konfirmační analýzou dat (kdy se pokoušíme vyvrátit, popř. potvrdit, dříve formulovanou hypotézu) a explorační analýzou dat (kdy hledáme nějaké zajímavé souvislosti), je rozdíl i mezi OLAP (kdy získáváme z dat sumární charakteristiky na zvolené podrobnosti pohledu) a data mining

(kdy hledáme v datech nějaké zajímavé souvislosti). OLAP tedy přináší odpovědi na konkrétní, přesně specifikované otázky, ale sám o sobě nic „neobjevuje".

2.4 Datové sklady a datová tržiště

Zatímco OLAP představuje nástroj pro analýzu (a vizualizaci) dat o firmě, datový sklad představuje místo, kde jsou analyzovaná data uložena. Podle W. H. Inmona, který v 80. letech zformuloval koncept datového skladu, je datový sklad

- subjektově orientovaný,
- integrovaný,
- časově proměnný,
- leč stálý

soubor dat, který slouží pro podporu rozhodování (Inmon, 1999).

Prvním charakteristickým rysem datového skladu je, že je orientován na subjekty, kterými se daná firma zabývá (zákazník, dodavatel, produkt a aktivita). To je výrazný rozdíl od tzv. *produkčních databází,* které se zabývají operacemi a transakcemi, jako např. půjčkami, fakturami, vklady, výběry apod. Datový sklad neuchovává data, která nejsou vhodná pro podporu rozhodování na manažerské úrovni, produkční databáze uchovávají data potřebná pro operativní řízení bez ohledu na to, zda budou využitelná při strategickém rozhodování.

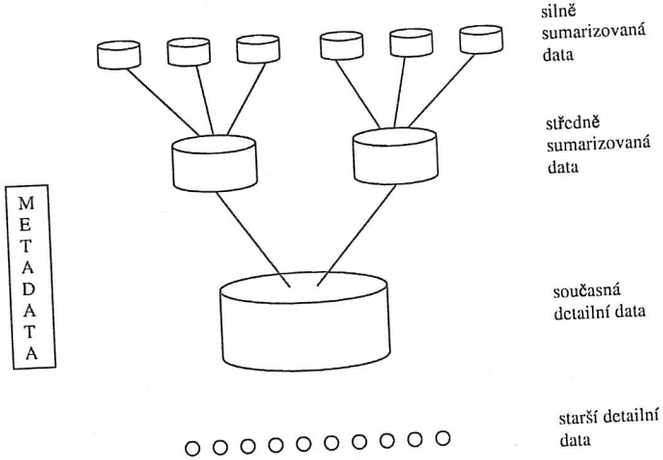
Vzhledem k tomu, že do datového skladu vstupují data z různých produkčních databází firmy, je důležitá *integrace a sjednocení dat.* Toto integrování zahrnuje sjednocení názvů stejných ukazatelů, sjednocení měřítek (různý způsob měření týchž veličin – např. finanční částky v tisících nebo v jednotkách, v různých měnách apod.), sjednocení kódování (např. pohlaví kódované v jedné databázi hodnotami „M" a „Z", v jiné databázi „0" a „1") apod.

Všechna data v datovém skladu představují „časový snímek" dat z produkčních databází sejmutý v určitém okamžiku. Datový sklad je aktualizován *off-line* v určitých časových intervalech (měsíčně, čtvrtletně, ročně – podle povahy dat a předpokládaných analýz) a je rovněž analyzován odděleně od produkčních databází. Výhodou je, že nešetrný zásah do datového skladu (dotaz, který vede k zacyklení) neovlivní operativní řízení firmy. Rovněž odezva na dotaz položený do datového skladu je rychlejší, než by byla odezva do produkční databáze. Produkční databáze je totiž plně vytížena zaznamenáváním transakcí a analytikovi by odpovídala jen okrajově. Nevýhodou je, že data v datovém skladu postupně stárnou. *Časovou proměnností* se tedy myslí v první řadě toto zafixování dat z produkčních databází. Druhé časové hledisko spočívá

v tom, že časové údaje jsou v datovém skladu explicitně přítomny jako jedna z důležitých informací.

Dotazy, které do datového skladu směřují uživatelé-analytici, nezpůsobují změnu zde uložených dat. Je tedy datový sklad v tomto smyslu s t á l ý .

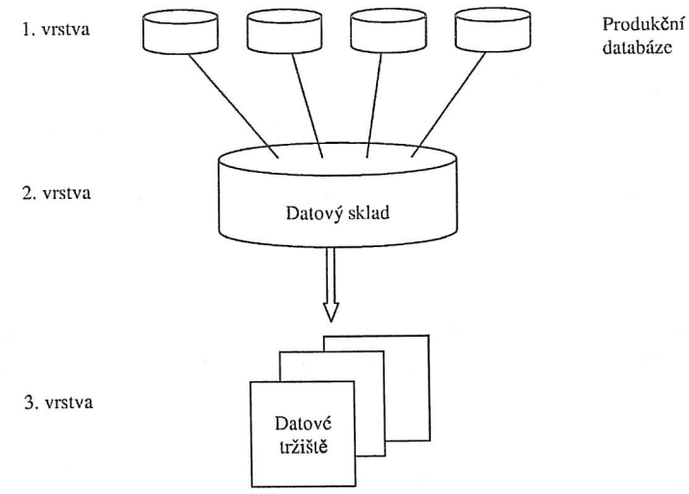
Struktura datového skladu je uvedena na obrázku 20. Datový sklad obsahuje obvykle operační data uložená v daném okamžiku, starší (historická) operační data, souhrny na různých úrovních abstrakce a tzv. *metadata*, zachycující



Obr. 20 Struktura datového skladu.

informace o datech. Vezměme jako příklad datový sklad obsahující údaje o prodeji nějakých výrobků. Současnými detailními daty pak mohou být údaje o prodeji v roce 2003, starší detailní data mohou být údaji o prodeji v letech 1998 až 2002, středně sumarizovaná data mohou udávat souhrnný prodej po skupinách výrobků za jednotlivé týdny nebo souhrnný prodej jednotlivých výrobků po okresech za jednotlivé týdny, silně sumarizovaná data mohou představovat souhrnný celkový měsíční prodej nebo souhrnný týdenní prodej za jednotlivé kraje, metadata mohou být údaji o tom, které okresy tvoří jednotlivé kraje, které výrobky vytvářejí jednotlivé skupiny, jakým způsobem byly kategorizovány numerické veličiny (např. jak byla rozdělena cena výrobku na cenu nízkou, střední a vysokou) apod.

Vytvoření datového skladu zahrnuje úkoly jako načtení dat, konverzi dat, čištění a transformaci. Data uložená v datovém skladu představují jakýsi neutrální datový prostor, který není vytvářen s myšlenkou konkrétních analýz. Proto se doporučuje vytvářet v návaznosti na datový sklad řadu specializovanějších *datových tržišť* (data mart), kam se z datového skladu přesunou data relevantní pro určitý typ analýz (pro určité oddělení firmy). Mluví se pak o třívrstvé (three tiered) architektuře datového skladu (Symons, 2000).



Obr. 21 Třívrstvá (three tiered) architektura datového skladu.

2.5 Dotazovací jazyky pro dobývání znalostí z databází

Klasický jazyk SQL, obdobně jako OLAP, umožňuje najít v databázích jen to, co hledáme. V tomto smyslu se tedy ani v jednom případě nejedná o dobývání znalostí (kdy přesně nevíme, co chceme). V posledních letech se ale objevilo několik rozšíření SQL směrem k řešení deskriptivních úloh KDD (Boulicaut, 1999).

Jedním takovým systémem je MINE RULE umožňující klást dotazy na asociační pravidla[8]. Příklad dotazu do databáze uvedené v tabulce 2 je na obrázku 22. Syntaxe systému umožňuje provádět výběr záznamů jako v běžném SQL (příkazy `FROM`, `WHERE`, `GROUP BY`, `CLUSTER BY`), na vybrané záznamy se pak aplikuje algoritmus pro hledání asociací (příkazy `SELECT`, `EXTRACTING RULES`). V uvedeném příkladu dotazu tedy pracujeme jen s produkty, které byly zakoupeny ve stejný den ve stejném městě. V takto zvolených záznamech hledáme pravidla tvaru

`IF` produkt_1 `&` produkt_2 `& ...` produkt_n `THEN produkt (SUPPORT,CONFIDENCE)`

kde

[8] O asociačních pravidlech pojednává podrobněji oddíl 5.2. Zde se spokojíme s intuitivním chápáním pravidla jako implikace „jestliže platí předpoklad, platí i závěr" doplněné o kvantitativní charakteristiky odvozené z počtu záznamů v databázi splňujících předpoklad, resp. závěr pravidla.

- SUPPORT (*podpora*) je podíl počtu záznamů, ve kterých současně platí předpoklad (BODY) i závěr (HEAD) pravidla, a celkového počtu záznamů vybraných na základě podmínky WHERE,
- CONFIDENCE (*spolehlivost*) je podíl počtu záznamů, ve kterých současně platí předpoklad (BODY) i závěr (HEAD) pravidla, a počtu záznamů, ve kterých platí pouze předpoklad.

Dotazem z obr. 22 do tab. 2 tedy budou nalezena pravidla:

```
IF podložky THEN šrouby (0.67, 1.00)
IF šrouby THEN podložky (0.67, 0.67)
```

V dotazovacím jazyce MSQL (Imielinski a Virmani, 1999) je od sebe oddělena fáze generování pravidel a fáze dotazování na výsledky analýzy.

```
MINE RULE Priklad AS
SELECT DISTINCT 1..n produkt AS BODY, 1..1 produkt AS
       HEAD, SUPPORT, CONFIDENCE
FROM Prodej
WHERE BODY.město = HEAD.město
   AND BODY.datum = HEAD.datum
EXTRACTING RULES WITH SUPPORT: 0.1, CONFIDENCE: 0.5
```

Obr. 22 Dotaz v MINE RULE.

```
Emp(Id,Age,Sex,Salary,Position,Car)

GetRules (Emp)
into R
where support > 0.1 and confidence > 0.9

SelectRules (R)
where body has {(Age=*), (Sex=*)}
   and head is {(Car=*)}
```

Obr. 23 Dotaz v MSQL – hledání pravidel.

```
Select *
from Emp
where violates all (GetRules (Emp)
                    where body is {(Age=*)}
                       and head is {(Salary=*)}
                       and confidence > 0.3)
```

Obr. 24 Dotaz v MSQL – hledání výjimek.

V příkladu z obr. 23 hledáme v databázi zaměstnanců pravidla, která by nám umožnila na základě věku a pohlaví odvodit, jaké má dotyčný zaměstnanec auto. Kromě hledání asociací nabízí MSQL i paralelní pohled na data a pravidla. Příklad z obr. 24 ukazuje, jak je možné nalézt všechny záznamy z databáze Emp, které odporují pravidlům tvaru `Age => Salary` s alespoň 30% spolehlivostí.

Literatura

Berson A., Smith S. J.: *Data Warehousing, Data Mining, and OLAP*. McGraw Hill, 1997.

Boulicaut J. F.: *Query languages for Knowledge Discovery in Databases*. Tutorial PKDD'99.

dbms, 2000 Internet. http://warehouse.chime-net.org/software/genappsw/dbms.

Dhar V., Stein R.: *Seven methods for transforming corporate data into usiness Intelligence*. Prentice Hall, 1997.

Humphries M., Hawkins M. W., Dy M. C.: *Data warehousing návrh a implementace*. Computer Press, 2002.

Imielinski T., Virmani A.: *A query language for database mining*. Data Mining and Knowledge Discovery, Vol. 3 No. 4, 1999, s. 373–408.

Information Discovery: *OLAP and Data Mining. Bridging the gap*. White Paper. DM Review. Internet. http://www.datawarehouse.com.

Inmon W. H.: *What is a Data Warehouse*. Tech Topic. Internet. http://www.cait.wustl.edu/papers/prism/vol. 11_no1.

Inmon W. H.: *Building the Data Warehouse* (3. vydání). John Wiley & Sons, 2002.

Lauritzen S.: *Graphical Models*. Oxford Statistical Science Series, Clarendin Press, Oxford 1996.

Pokorný J., Halaška I: *Databázové systémy*. ČVUT, 1998.

ShowCase Corp.: *Accelerating the deployment of a business intelligence system*. White Paper.

Symons V.: *Three tiered Data Warehouse structure*. White Paper. DM Review. Internet, http://www.datawarehouse.com.

Thomsen E.: *OLAP Solutions, Building Multidimensional Information Systems*. (2. vydání) 2002.

TM1 White Paper. Internet. http://www.csm.uwe.ac.uk/~jharney/tm1wppr.htm.