- Hromadné zpracování dat na počítači
- Databázový systém a jeho struktura
- Role uživatelů databázových systémů
- Klasifikace databází
- Předrelační databáze
- Relační databáze
- Objektové databáze
- Multidimensionální databáze
- Datové sklady
- Data mining

04

Databázové systémy

VÝUKOVÝ TEXT PRO STUDENTY INFORMAČNÍCH TECHNOLOGIÍ

Marek Lučný, SŠPU OPAVA 2015

A Hromadné zpracování dat na počítači

Elektronické databázové systémy vznikly z potřeby přechovávat a zpracovávat velké množství dat. Postupně nahradily **papírové kartotéky** a plní například také funkci rejstříků v knihovnách či archívech. Patří k nejpoužívanějším počítačovým programům a zpracovávání rozsáhlých dat bylo velkou výzvou od počátku éry výpočetní techniky.

Před vznikem prvních databázových systémů bylo hromadné zpracování dat řešeno formou **systémů pro správu souborů** (*FMS - File Management System*). Tyto programy umožňovaly vytváření tzv. **prostých databázových souborů** (*flat files*) se strukturou podobající se tabulce; jednotlivé řádky jsou nazývány **záznamy** (*records*) a jsou složeny z dílčích **polí** (*fields*). Každá taková struktura byla uložena v samostatném souboru a neměla vztah k jiným souborům.

Souborový systém přitom mohl v daném čase využívat pouze jediný soubor-tabulku, a ten tedy musel obsahovat všechna potřebná data. Důsledkem byla mnohem vyšší úroveň **redundance** (nadbytečné opakování dat) než v pozdějších databázových systémech. Redundance přirozeně vede k častějšímu výskytu chyb v datech, potřebě větší úložné kapacity, delšímu operačnímu času při manipulacích s daty (například kvůli **sekvenčnímu vyhledávání údajů**). Problematické je rovněž zajištění paralelního přístupu více uživatelů a zabezpečení dat.

Kvůli těmto omezením byly systémy pro správu souborů ve většině případů nahrazeny plnohodnotnými databázovými systémy. Využití jednoduchých souborů se vyplatí jen u některých specifických aplikací, kde nedochází k složitým manipulacím s daty, u nichž se předem počítá s jen omezeným množství dat s neměnnou strukturou a přístupem jednoho kvalifikovaného uživatele.

Prostý databázový soubor může mít podobu jednoduchého textového souboru, který většinou obsahuje jeden záznam na jednom řádku (např. soubory typu **CSV**), nebo v sobě může kombinovat text a binární data (např. soubory typu *DBF* nebo *DBM*). Moderní aplikace využívají pro ukládání dat strukturované datové soubory typu **XML** (*eXtensible Markup Language*) nebo **JSON** (*JavaScript Object Notation*).

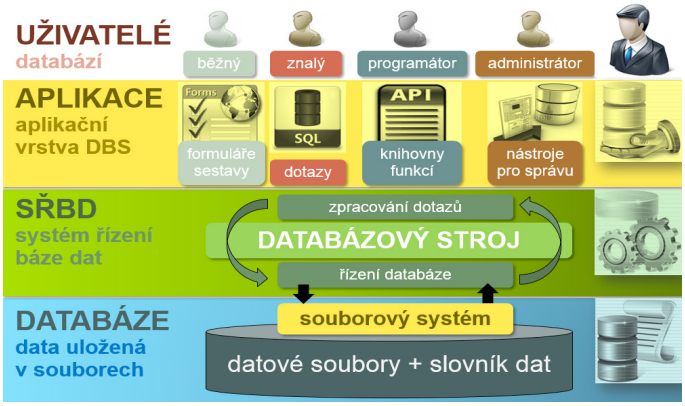
B Databázový systém a jeho struktura

Databáze je soubor souvisejících dat, které jsou ukládány a organizovány za účelem získávání a zpracovávání potřebných informací. Databázový software bývá označován zkratkou **DBMS** (*Database Management System* = systém pro správu báze dat). Umožňuje nejen vytváření a správu databáze, ale zajišťuje rovněž potřebné zabezpečení dat.

Klíčovou komponentou DBMS je **databázový stroj** (*database engine*), tedy ta část programu, která se stará o bezpečné ukládání dat, nebo naopak získávání dat.

Databázový stroj obsahuje vrstvu, jež přímo komunikuje s **databází**, která obsahuje samotná data. Informace o struktuře databáze jsou uloženy v tzv. **datovém slovníku** (*data dictionary*). V případě nejčastěji používaných relačních databází tuto strukturu tvoří **tabulky** složené z jednotlivých **polí** (sloupců). Každé pole má svůj název, příslušný datový typ a další sadu vlastností. Součástí datového slovníku jsou rovněž informace o klíčích, indexech, vazbách mezi tabulkami apod. Datový slovník je přístupný pouze databázovému stroji, aby byla zajištěna **integrita databáze** a nedocházelo k narušení konzistence uložených dat.

Jiná vrstva databázového stroje zajišťuje komunikaci s uživatelskými nástroji, které jsou buď součástí samotného DBMS (jako v případě *MS Access*), nebo samostatnými aplikacemi, jež služeb DBMS využívají. **Databázové aplikace** nabízejí **formuláře** pro zadávání vstupních údajů, **sestavy** pro výstupní data i nástroje pro vytváření struktury databáze, zadávání dotazů či správu databázového systému. Administrátorům a programátorům poskytují DBMS rozhraní s podporou dotazovacích i programovacích jazyků – **API** (*Application Programming Interface* = rozhraní pro programování aplikací).



Standard označovaný zkratkou **ODBC** (*Open Database Connectivity*) tvoří aplikační rozhraní pro snadný a jednotný přístup k různým databázovým systémům nezávisle na programovacím jazyku i operačním systému. ODBC zprostředkovává komunikaci klientské aplikace se serverem. Klientský program se může prostřednictvím ODBC současně připojit k několika různým databázovým systémům a pomocí dotazů (nejčastěji v jazyce SQL) provádět různé operace s daty. V ODBC musí samozřejmě být nainstalován potřebný ovladač (driver) pro příslušný databázový systém, který komunikaci zajistí. Obdobou ODBC je standard **JDBC** (*Java Database Connectivity*), který využívají ke komunikaci s databázemi programy vyvíjené v jazyce Java.

C Role uživatelů databázového systému

Vzhledem ke značnému rozšíření a využití databázových systémů je nutné počítat, že s nimi pracují uživatelé na různých úrovních a vystupujících v různých rolích.

Databázoví analytikové (*DA*). Jsou zodpovědni za návrh databáze. Základem návrhu musí být důkladná a přesná analýza všech požadavků na systém, včetně identifikace typů ukládaných údajů, vztahů mezi daty, žádoucích výstupů apod.

Vývojáři a programátoři. Vytvářejí aktuální databáze a databázové aplikace podle návrhů databázových analytiků. K procesu vývoje patří především vytvoření struktury databáze a uživatelského rozhraní, často s využitím nástrojů, které jsou k tomu v DBMS určeny. V jiných případech (např. u webových aplikací) vyvíjejí programátoři vlastní klientskou aplikaci, která využívá serverových služeb DBMS.

Administrátoři databází (*DBA*). Jsou lidé, kteří se starají o spolehlivý chod databází. Provádějí pravidelnou údržbu, řídí uživatelské přístupy, monitorují výkon databáze, provádějí pravidelné zálohování, starají se o její zabezpečení a konfiguraci. Úzce spolupracují s návrháři i vývojáři databáze na všech změnách systému, protože jsou nejlépe obeznámeni s jeho požadavky i možnými riziky.

Běžní uživatelé. Využívají služeb databáze; plní ji daty a provádějí aktualizace údajů, tisknou výstupní sestavy. K databázi přistupují prostřednictvím uživatelského rozhraní a obvykle nemají podrobnější znalosti o fungování samotného systému ani o způsobu organizace dat. Mohou se proto snadno dopustit chyby zejména během vyplňování formulářů, a je proto nezbytné, aby všechny vstupní údaje byly důkladně ověřovány (validovány).

D Klasifikace databází

Na databázové systémy můžeme pohlížet z různých pohledů a podle toho je rozdělovat na různé typy.

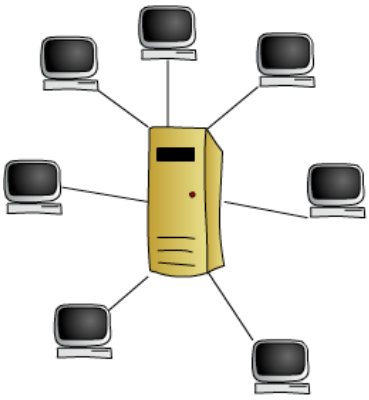
ROZDĚLENÍ PODLE MOŽNOSTI PŘÍSTUPU UŽIVATELŮ

Jednouživatelské systémy. Jsou umístěny na jednom počítači a určeny k přístupu jednoho uživatele. Často se využívají pro vedení osobní agendy nebo pro účely soukromého podnikání.

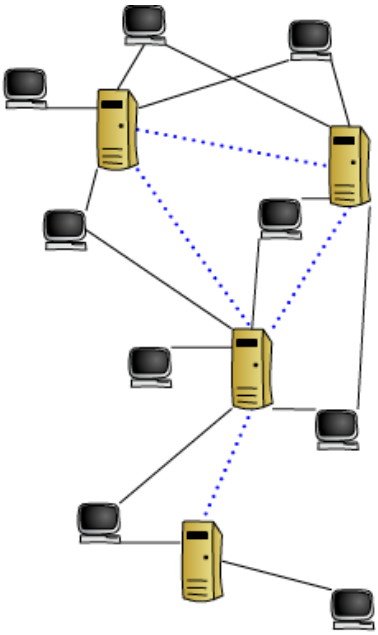
Víceuživatelské systémy. Většina databázových systémů je navržena pro víceuživatelské použití. Obsahují určitý typ uzamykání databázových objektů, aby nemohlo docházet ke konfliktům kvůli současnému přístupu více uživatelů ke stejným datům.

ROZDĚLENÍ PODLE ZPŮSOBU ORGANIZACE PROVOZU DBMS

Databáze typu klient-server. Jsou obdobou sítí typu klient-server; i v tomto případě je systém rozdělen na klientskou aplikaci (**front-end**) a serverovou aplikaci (**back-end**). Server obsahuje DBMS i samotnou databázi a reaguje na příkazy, které přicházejí z klientských aplikací. Typicky jde o internetové databázové aplikace, kdy front-end aplikace se ve formě webové stránky zobrazuje v okně webového prohlížeče.



Centralizované databáze. Databáze i obslužné aplikace jsou centrálně umístěny na jednom počítači, kterým je server nebo mainframe. Uživatelé k centrálnímu systému přistupují prostřednictvím terminálů vybavených základními vstupními zařízeními (klávesnice, čtečka čárových kódů apod.) a displejem. **Terminály** mohou být například i velmi jednoduché bezdiskové počítače, protože veškeré výpočetní a paměťové operace jsou prováděny centrálně. Centralizované systémy jsou i kvůli bezpečnosti hojně využívány v bankách, supermarketech, letištních terminálech apod.



Distribuované databáze. Data jsou rozdělena do několika databází, z nichž každá je uložena na odlišném počítači, často i na různých místech (centrála firmy, sklad…). Počítače jsou propojeny v síti a nastaveny tak, aby se přihlášenému uživateli jevily jako jeden celistvý systém. Často je toto řešení používáno kvůli rozdělení zátěže serverů v případě velmi využívaných aplikací. Distribuované databáze, které podporují *cloud computing*, bývají označovány jako **cloudové databáze**. Kromě podnikových databází velkých firem běží distribuované databáze v pozadí všech významných internetových služeb (*Google*, *Facebook*, *YouTube* a mnoha dalších).

ROZDĚLENÍ DBMS PODLE POUŽITÉHO DATOVÉHO MODELU

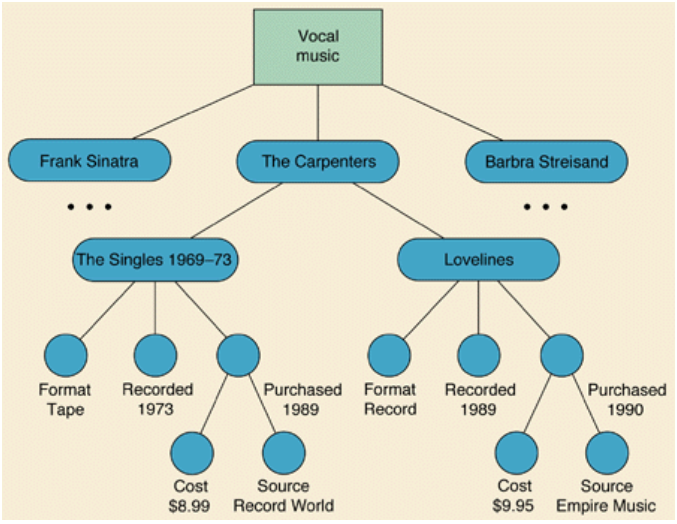
Nejvýznamnějším kritériem pro rozlišení typů databázových systémů je použitý **datový model**. Datový model určuje, jak jsou uvnitř databáze uspořádána data a jakým způsobem je s nimi nakládáno. Protože se od osmdesátých let 20. století nejvíce rozšířil relační datový model, jsou často starší typy databází, využívající převážně hierarchické nebo síťové datové modely, označovány jako *předrelační*, zatímco moderní databáze (objektově-orientované, multidimensionální, hybridní) bývají souhrnně pojmenovány jako *postrelační*.

E Předrelační databáze

Používaly se před rozšířením relačních databází, především v 60. a 70. letech minulého století, a byly založeny na hierarchickém nebo síťovém datovém modelu.

HIERACHICKÉ DATABÁZE

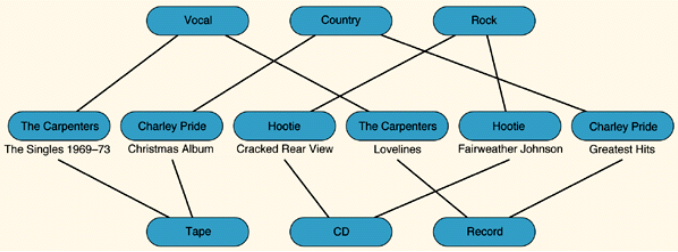
V hierarchické databázi jsou pole a záznamy uspořádány do struktury, která připomíná **strom generací** (rodokmen) – záznamy na vyšších úrovních označujeme jako **předky** (*parents*) a jim podřízené záznamy jako **potomky** (*childs*). Záznam na nejvyšší úrovni je nazýván pojmem **kořenový záznam**



(*root record* nebo *root parent*). Hierarchický datový model je nejstarším a nejjednodušším z datových modelů, který byl hojně používán v době mainframů a magnetických pásek, ale dnes přežívá spíše jen v některých typech rezervačních systémů. Výhodou hierarchického modelu je velmi rychlý přístup k datům díky předem definovaným vztahům. Na druhou stranu právě předdefinovaná struktura přináší velká omezení: přidání nového záznamu obvykle znamená časově náročnou aktualizaci celé databáze, přísná závislost potomků a předků (potomek může mít vždy pouze jednoho předka) je problematická v případě potřeby vícenásobných vztahů mezi daty – vzniká velká redundance.

SÍŤOVÉ DATABÁZE

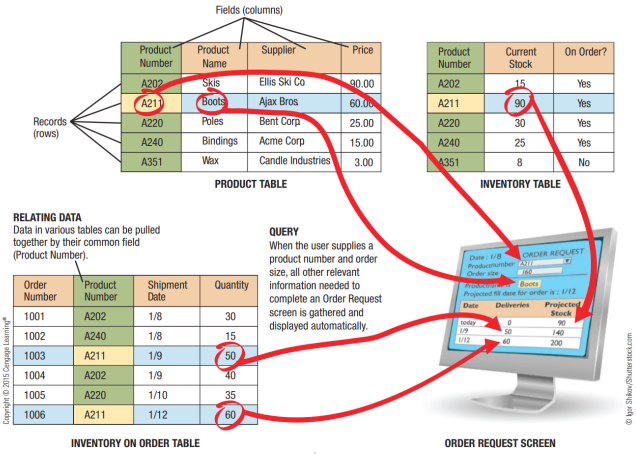
Síťové databáze byly zčásti vyvinuty i proto, aby řešily některé problémy hierarchických databází. V síťovém datovém modelu - na rozdíl od hierarchického - může mít jeden potomek více než jednoho předka. I z toho důvodu se



používá odlišná terminologie: záznamy nejsou označovány slovy potomek a předek, nýbrž pojmy **člen** (*member*) a **vlastník** (*owner*). Síťový datový model je flexibilnější než hierarchický a dokáže zachytit složitější datové struktury. Ovšem i v tomto případě platí, že struktura musí být definována předem a není snadné ji aktualizovat. Existují také limity pro počet možných spojení mezi záznamy a vždy je nutné načíst celý záznam byť jen kvůli jedinému údaji. Ani síťové databáze se proto dlouhodobě neprosadily a postupně byly vytlačeny relačními databázemi.

F Relační databáze

V roce 1970 vystoupil anglický vědec E. F. Codd s návrhem relačního datového modelu, který se stal základem relačních databází. V relačních databázových systémech (**RDBMS**) jsou data uložena v tabulkách (z pohledu samotné relační teorie je označujeme matematickým pojmem **relace** (*relation*), které jsou tvořeny **řádky** (*rows*) a **sloupci** (*columns*). Každý řádek je jednoznačně identifikován tzv. **primárním klíčem** a každý sloupec má unikátní název. Kvůli minimalizaci redundance (nadbytečného opakování dat) jsou v relačních databázích data rozdělena do většího množství tabulek a ty jsou mezi sebou provázány díky společně sdíleným sloupcům – primárním a cizím klíčům.



Podle relační teorie je možné pomocí základních operací (sjednocení, kartézský součin, rozdíl, selekce, projekce a spojení) uskutečnit nad relační databází veškeré operace s daty. Tyto operace provádí databázový stroj na podkladě **dotazů** (*queries*). Dotazy mohou být vytvářeny pomocí dotazovacích jazyků, jako jsou **SQL** (*Structured Query Language*) či **QBE** (*Query by Example*).

První komerční implementace relačního modelu byly uvedeny do praxe na počátku 80. let minulého století. Pomocí relačních databází lze vyřešit poměrně širokou škálu zadání, a proto jsou i v současnosti nadále nejrozšířenějším typem databázových systémů. Základem podnikových informačních systémů jsou komerční databázové systémy vyvíjené nejvýznamnějšími firmami v oboru: *Oracle*, *IBM* (systém *DB2*), *Sybase* nebo *Microsoft* (*SQL Server*). Mnoho internetových aplikací využívá k svému provozu open source databáze - *MySQL* nebo *PostgreSQL*.

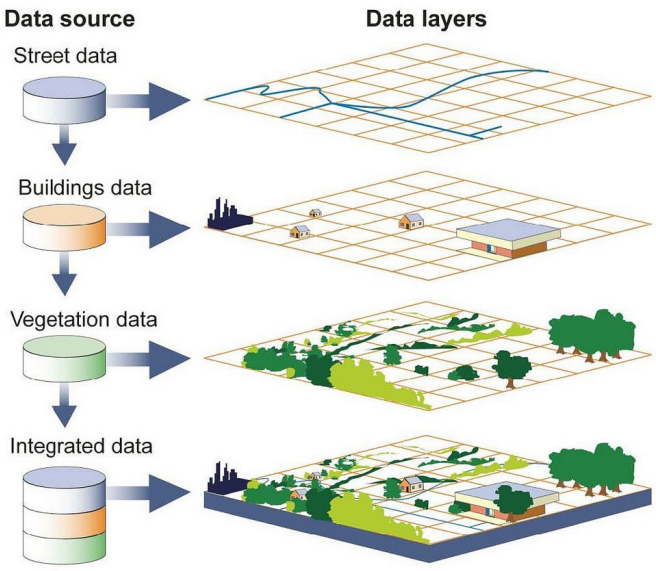
K trvalé oblibě relačních databází přispívají zejména srozumitelná a jednoznačná struktura dat, propracovaná metodika návrhu databáze, možnost využití pro velké množství rozličných úloh, dostupnost kvalitních programových nástrojů a další.

Samozřejmě narážíme i na řadu omezení, které mohou být důvodem pro volbu modernějších typů databází – například slabá podpora objektově orientovaného programování může vést vývojáře k upřednostnění objektově orientovaných nebo objektově relačních databází, specifické požadavky na analytické zpracování rozsáhlých dat zase k užití databází multidimensionálních.

G Objektově-orientované databáze

V souvislosti se vznikem objektově orientovaného programování (OOP) se objevily **objektově-orientované databázové systémy** (*ODBMS - object-oriented database management system*) založené na **objektovém datovém modelu**. Jeho základní prvek – **objekt** – neobsahuje pouze **data**, ale zahrnuje současně i **metody**, které mohou s daty provádět určité operace (např. kontrolu vstupů, formátování výstupů, různé výpočty apod.).

Kromě toho je možné využívat i dalších výhod spojených s koncepcí OOP, včetně **dědičnosti** a **polymorfismu**. Díky dědičnosti se může například objekt student stát potomkem obecného objektu člověk, zdědit jeho data (např. jméno, příjmení, adresu, datum narození) i metody (např. tisk identifikační karty) a rozšířit je o nové atributy i metody, které jsou specifické pro situaci studenta (označení třídy, výpočet průměrné známky atd.).



Objekty navíc nemusí být předem daným a uzavřeným celkem a mohou obsahovat i nestrukturovaná data v různých podobách, včetně grafiky, audia či videa. Velmi efektivní může být rovněž přístup k datovým objektům; v tomto případě se k manipulaci s daty používá jazyk **OQL** (*Object Query Language*), obdoba relačního dotazovacího jazyka SQL.

Mezi čistě objektově-orientované DBMS patří například *GemStone*, *FastObjects* nebo *Objectivity DB*. Velké firmy, jako IBM, Oracle, Microsoft nebo Sybase, vkládají řadu objektových rysů do svých původně relačních databází, a proto se často mluví o tzv. **objektově-relačních** nebo **hybridních databázích**.

OBLASTI VYUŽITÍ ODBS

Multimediální databáze pro ukládání fotek nebo klipů.

Geografické informační systémy (GIS - Geographic Information System), které nabízejí několik různých vrstev map napojených na aktuální data.

Databáze pro groupware sloužící k ukládání dokumentů jako jsou kalendáře, časové plány, manuály, poznámky nebo zprávy.

Databáze pro CAD systémy, které mohou obsahovat uložené grafické komponenty nebo třeba předchozí verze návrhů technických výkresů a modelů.

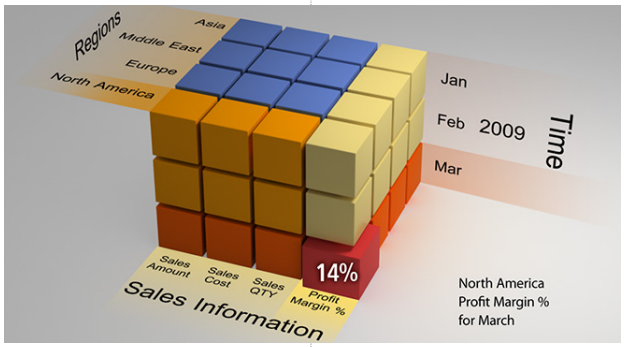
Hypertextové databáze obsahující odkazy na různé zdroje informací umístěné na Internetu; využívají je různé vyhledávací služby.

ORM

Zatímco je v relační databázi entita reprezentována jako řádek, resp. množina řádků v databázových tabulkách, tak v objektově orientovaném jazyce je entita zpravidla reprezentována jako instance nějaké třídy. Tato rozdílná reprezentace entit vedla ke vzniku programovací techniky označované jako **ORM** (*Object-relational mapping*), která se stará o konverzi mezi relační databází a objekty v objektově orientovaném jazyce. Technika se snaží nabídnout vývojářům jednotný přístup k libovolné datové entitě, s níž se v aplikaci pracuje. Díky tomu jsou vývojáři při práci s daty, které jsou v aplikaci reprezentovány právě pomocí objektů, do značné míry odstíněni od nutnosti pracovat s SQL dotazy konkrétní relační databáze. Použití ORM usnadňuje zejména provádění běžných databázových operací, jako jsou čtení, zápis, úprava a mazání dat (*CRUD operace*). Jednou z výhod ORM je nezávislost na konkrétním databázovém systému, což poskytuje programátorům prostor pro volbu nejvhodnějšího databázového systému nebo jiného datového úložiště podle požadavků aplikace.

H Multidimenzionální databáze

Multidimenzionální databáze (**MDDB**) byly vyvinuty spíše za účelem **analýzy dat** než kvůli provádění online transakcí. Oproti běžným relačním databázím zvládají mnohonásobně rychleji zpracování rozsáhlých souhrnných informací.



Data, která jsou shromažďována z různých zdrojů (často z relačních a objektových databází), jsou sumarizována a uspořádána do speciální vícerozměrné struktury – tzv. **hyperkostky** (*hypercube*). Takový multidimenzionální datový model umožňuje předdefinovat pohledy na data z různých úhlů – tzv. **dimenzí** (*dimensions*).

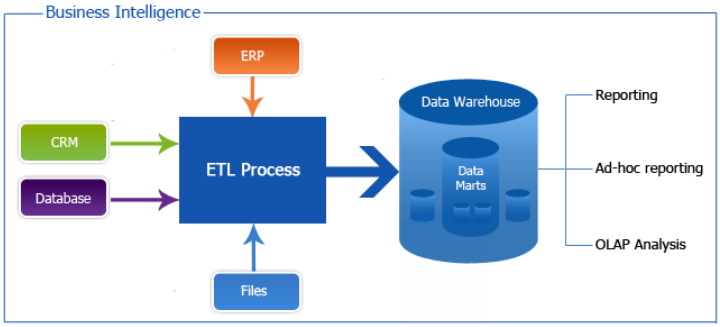
Například v obchodní společnosti mohou být prodejní aktivity zkoumány z pohledu úspěšnosti prodeje jednotlivých produktů, z časové i geografické perspektivy, z předdefinované dimenze porovnávající preference různých skupin zákazníků apod.

Mezi programové prostředky, které jsou používány ve spojení s multidimenzionálními databázemi, patří tzv. **OLAP technologie** (*Online Analytical Processing*). Výstupními informacemi těchto programů jsou především analyticky zaměřené tabulky, grafy a nejrůznější diagramy, kterými lze disponovat při tvorbě obchodních zpráv nebo při strategickém rozhodování na podkladě zjištěných trendů.

Příkladem multidimenzionálních databází jsou systémy *ContourCube* nebo *Cognos PowerPlay*.

I Datové sklady

Datové sklady (*data warehouse*) jsou obrovské databáze, které přechovávají a spravují velké objemy dat. V datových skladech jsou uloženy informace nejen o aktuálních transakcích, ale často i těch provedených v minulosti. Vzhledem k svému rozsahu bývají datové sklady distribuovanými databázemi, které jsou rozděleny na serverech umístěných v různých lokalitách a propojeny prostřednictvím Internetu tak, aby fungovaly jako jeden celek.



Menší verze datových skladů jsou odborně nazývány **data mart**. Jsou k dispozici specifickým skupinám uživatelů nebo slouží pro potřeby jednoho oddělení či pobočky velké společnosti.

Datové sklady i data mart jsou využívány jednak jako zdroj dat pro analytické účely (např. pro výše zmíněné multidimenzionální databáze), jednak pro tzv. „*dolování dat*“ (*data mining*). Prostřednictvím webových aplikací mohou datové sklady posloužit i běžným uživatelům – například k zjištění historie provedených transakcí v souvislosti s konkrétní kreditní kartou, bankovním účtem apod.

J Data mining

Díky celosvětovému propojení počítačů a celé řadě programů, které s vědomím i bez vědomí svých uživatelů permanentně shromažďují ohromné množství dat, je možné s využitím důmyslných metod zkoumat tato „velká data“ (*big data*) a získávat z nich nové informace i ve zcela nečekaných souvislostech. **Data mining**, neboli „*dolování dat*“, je proces procházení a analýzy rozsáhlých dat za účelem objevování skrytých vzorů i významů, které umožňují odhalit některé trendy v minulosti a lépe předvídat budoucí vývoj.

Na počátku procesu „dolování dat“ musí být pochopitelně konkrétní zadání, podle něhož je prováděn i **sběr dat**. Ten je obvykle realizován s využitím mnoha zdrojů – od nejrůznějších databází, přes datové soubory (např. logovací soubory, cookies), online informace pocházejících z internetových článků, blogů a třeba i diskuzí na sociálních sítích, až po údaje získané z datových skladů, ale třeba i z procesů běžících na pozadí operačních systémů.

Shromážděná data poté procházejí **procesem čištění** (*data cleansing* nebo také *data scrubbing*), v němž jsou zbavována možných chyb a nekonzistencí. Velký význam má také odlišení dat od metadat. **Metadata** jsou vlastně data o datech – popisují zejména způsob uložení dat (datové slovníky, hlavičky datových formátů), mohou obsahovat časové informace o uložení souborů apod.

Dalším krokem je **analýza dat**, kdy jsou kombinovány nejrůznější metody – statistikou počínaje a konče vysoce sofistikovanými přístupy (využití generické ho programování, neuronových sítí apod.). Výstupními informacemi této fáze jsou jednak obecnější znalosti (např. že svobodní klienti nejčastěji nakupují pozdě večer, zatímco ženatí po obědě), jednak přísně **matematické modely** (např. postup jak vytipovat potenciálního klienta pro daný produkt).

Aplikace zjištěných poznatků v praxi může být opět velice rozmanitá. Výsledkem komerčního data miningu bývá snaha žádoucím směrem ovlivnit chování zákazníků (např. reorganizací webových stránek, přesně mířenou reklamní kampaní). Data mining pochopitelně může stejně dobře posloužit pokroku vědy (např. přínosným zkoumáním komplikovaných sociálních jevů) jako se stát podpůrným nástrojem pro aktivity politické, vojenské, nebo i zločinecké.