

Výuková prezentace 4

6MDBS1

Databázové systémy

Ing. Vladimír Přibyl, Ph.D.



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MŠMT
MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY

Architektury IS



Typy architektur

- Existuje řada typů architektur (architektonických vzorů), které se používají v různých modifikacích pro návrh informačních systémů
 - Např. Master – Slave, architektura se společným úložištěm, Pipe and Filter atd.
- Pro systémy založené na práci s daty a tedy postavené na databázových systémech je však velmi obvyklá architektura Klient – Server. Naprostá většina podnikových IS má tuto architekturu.
- Čím dál častěji se však uplatňují také distribuované architektury typu peer to peer, kde není dedikovaný databázový server. Např. často ve spojení se zpracováním tzv. Big Data.

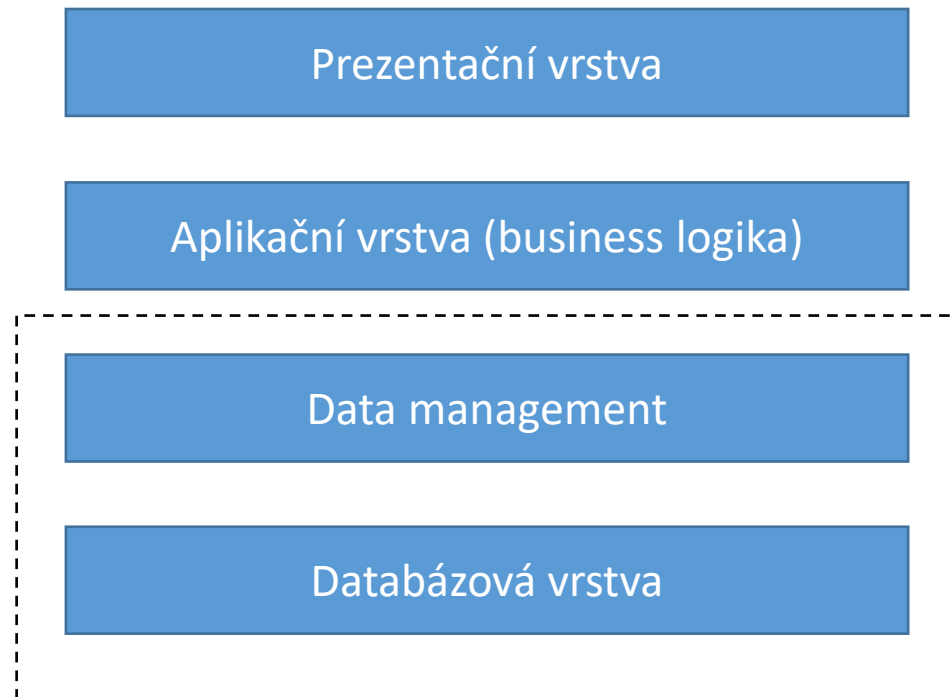


Architektura Klient - Server

- Klientské aplikace (část např. webový prohlížeč) jsou rozptýleny mezi uživateli systému a ten je využívá pro interakci se systémem.
- Server poskytuje pak těmto klientům požadované služby



Model vrstev pro klient-server aplikace

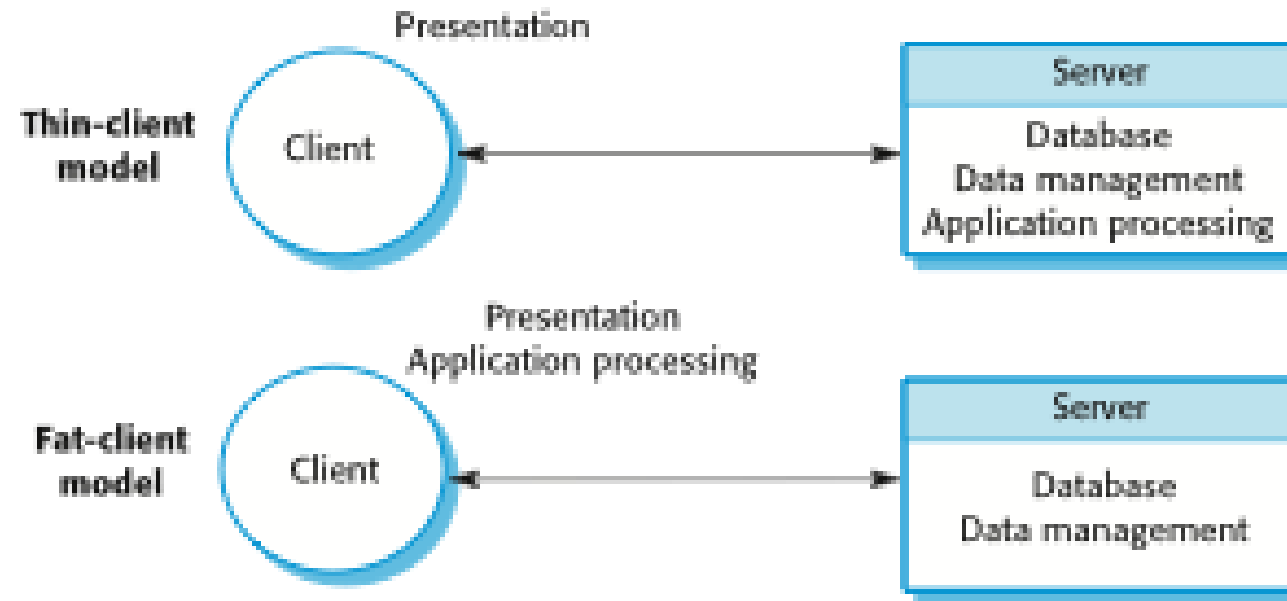




Architektura klient server se dvěma stupni

- V tomto typu architektury existuje server a blíže neurčený počet klientských aplikací. Obvykle fyzicky oddělených.
- V tomto případě existují dvě varianty klientských aplikací, které se od sebe liší tím, které vrstvy modelu, resp. které služby realizují:
 - Tenký klient (Thin-client model), kdy klient realizuje je prezentační vrstvu a služby ostatních vrstev realizuje server
 - Tlustý klient (Fat-client model), kdy klient realizuje také aplikační vrstvu.

Tlustý a tenký klient





Tenký klient

- Výhody:
 - Snadná aktualizace aplikační logiky (jen na serverové straně).
 - Nízké požadavky na klientskou stranu, většinou jen webový prohlížeč.
 - Klient nepotřebuje žádný speciální SW
- Nevýhody:
 - Velká zátěž pro server a tedy problém ve škálování a vyšší nároky a tedy i náklady na serverovou stranu



Tlustý klient

- Nevýhody:
 - Nesnadná aktualizace aplikační logiky (nutno distribuovat nové verze klientů).
 - Vyšší požadavky na klientskou stranu.
 - Klient potřebuje speciální SW
- Výhody:
 - Menší zátěž pro server a tedy menší problém ve škálování a nižší nároky a tedy i náklady na serverovou stranu

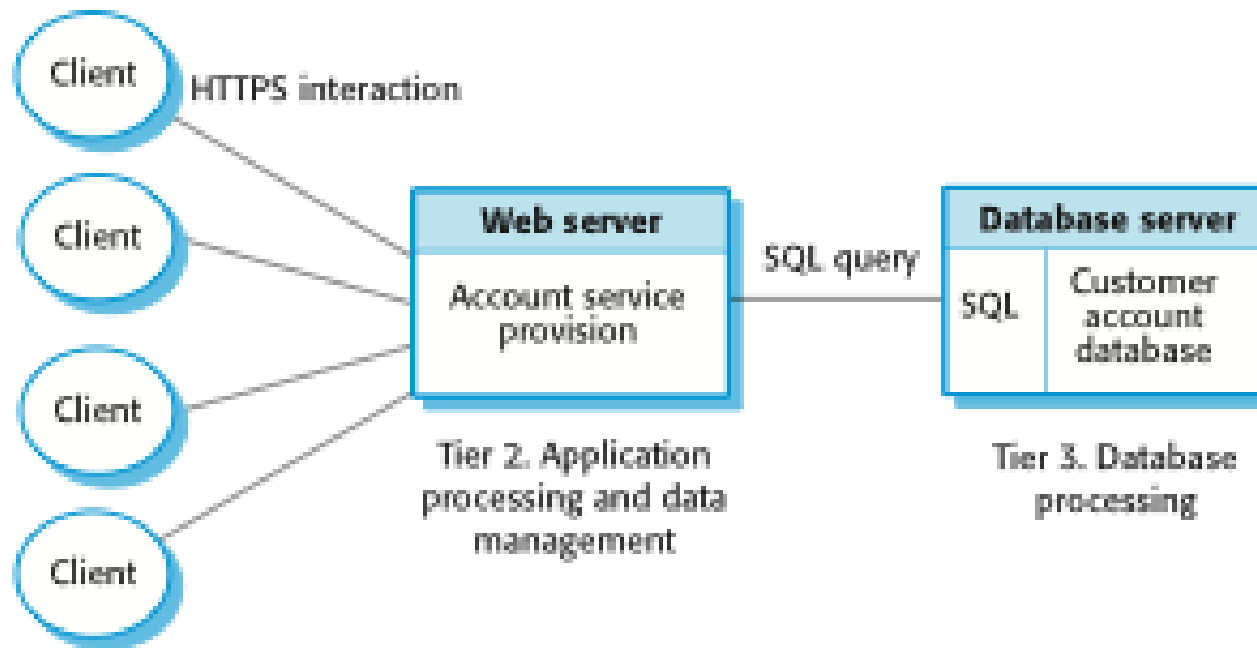


Vícestupňová architektura klient-server

- Jednotlivé logické vrstvy modelu jsou rozděleny mezi více fyzických stupňů (procesorů)
 - Je možné zachovat výhody tenkého klienta a přitom se lépe vyrovnat s problémy se škálováním na serverové straně
 - Často se jedná o třístupňovou architekturu, kdy aplikační vrstva je realizována samostatně.

Příklad třístupňové architektury

Tier 1. Presentation





Přístup k datům – klient/server

- Spojení klienta se serverem
 - ODBC ovladač
 - Zejména produkty Microsoft (MS SQL server, MS Access, FoxPro, Excel ..)
 - Musí být definován v systému před spuštěním klienta
 - Speciální klienti
 - např. Oracle ..

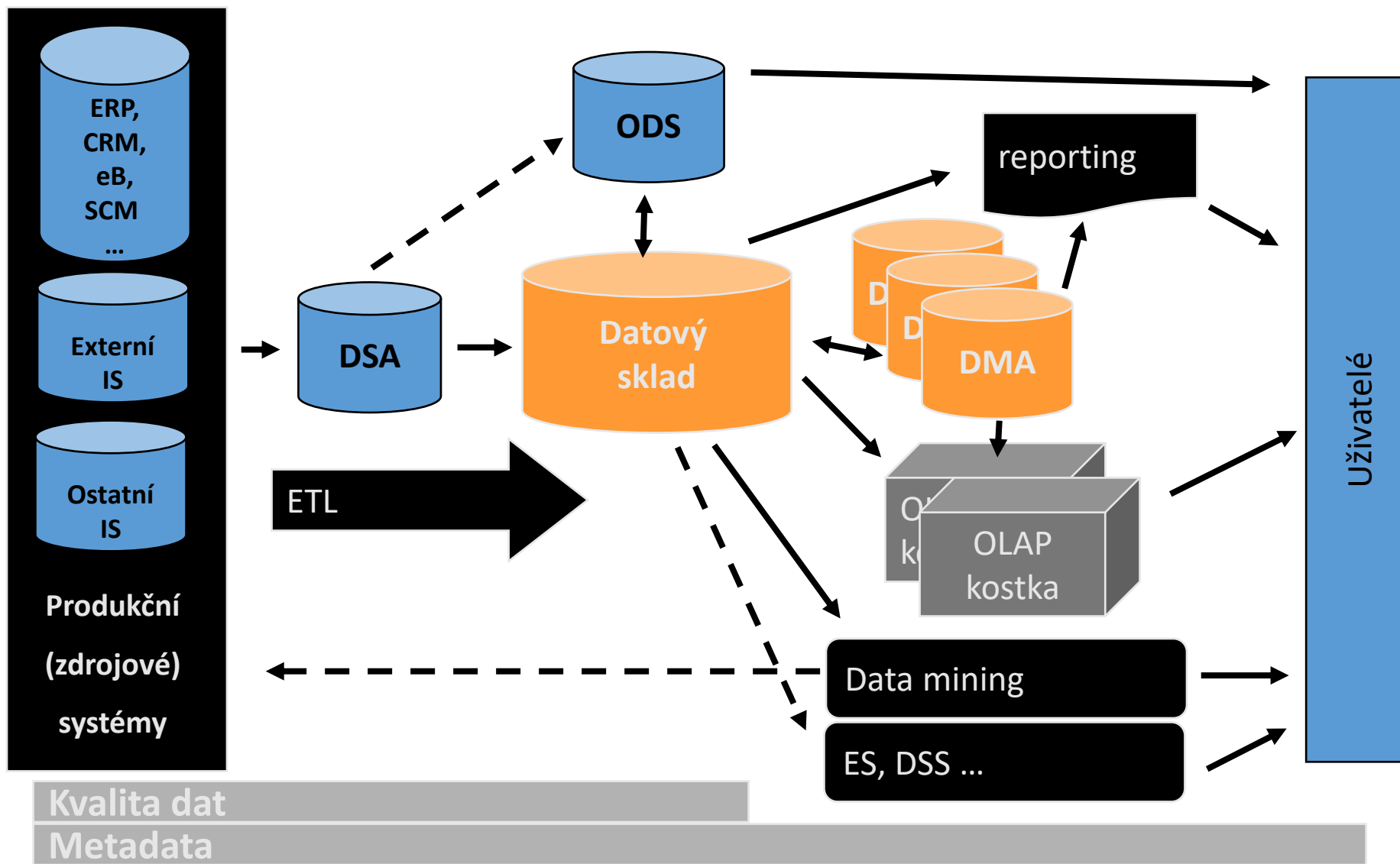
Multidimenzionální databáze



Business Intelligence (BI)

- Pojem „**multidimenzionální databáze**“ je spojen s oblastí BI, což je sada procesů, aplikací a technologií, jejichž cílem je účinně a účelně podporovat zejména analytické a plánovací činnosti v organizacích a které jsou právě postaveny na principech multidimenzionálních pohledů na data.
- Transakční systémy (zejména v rámci ERP) mají datové základny založeny nejčastěji na relačních databázích a optimalizovány pro efektivní provádění relativně jednoduchých, ale velmi frekventovaných transakcí. Takto koncipované datové základny mají z hlediska úloh BI zejména tato omezení
 - Neumožňují rychle a pružně měnit kritéria pro analýzu dat (resp. odezva na tyto změny není obvykle dost rychlá pro potřeby analýzy a rozhodování)
 - Jsou velmi vytíženy pořizováním a aktualizací dat a neposkytují tedy dostatek výkonu pro rychlé zpracování složitých analytických úloh BI
 - Často obsahují nadměrné množství příliš detailních dat
- Transakční systémy (jsou také často nazývány produkční, primární, OLTP či „legacy“) tedy nepatří do BI, ale jsou pro BI zdrojem dat (často jediným)
- BI aplikace využívají jiné datové struktury pro ukládání dat a v porovnání s transakčními systémy pracují s méně detailními (agregovanými) daty

Vztah komponent komplexního řešení BI





Typy dat v (multi)dimenzionálních databázích

Data lze rozdělit na dva základní typy:

Fakta: velmi rozsáhlé a postupně narůstají (např. data o provedených obchodních transakcích)

- často “insert-only”, tedy neustále se kumulují

Data dimenzí: obvykle malý rozsah v porovnání s fakty, v podstatě statické. Informace o entitách vstupujících do fakt

- mají obvykle hierarchickou strukturu
- jednou z dimenzí je vždy čas



Realizace dimenzionální struktury

- Star schéma (hvězda)
 - dvouúrovňová struktura
 - centrální tabulka faktů a k ní připojené tabulky dimenzí, které nemusí být normalizovány
- Snowflake schéma (sněhová vločka)
 - tabulky dimenzí tvoří hierarchickou strukturu, jsou normalizovány



Převod dat do datového skladu

- Zdroj dat tvoří produkční systémy, případně externí datové zdroje (statistické údaje, registry ...)
- **ETL** (Extraction, Transformation and Loading)
 - Jedna z nejvýznamnějších komponent BI. Často jsou tyto prostředky nazývány „Datová pumpa“. Jejich úkolem je data ze zdrojových systémů získat a vybrat (extraction), upravit do požadované formy a vyčistit (transformation) a uložit do specifických datových struktur (loading).
 - Umožňují přenos dat z různorodých systémů



OLAP

- zpracování dat zaměřené na analytickou práci (On Line Analytical Processing)
- využívá multidimenzionálního uspořádání dat
- urychluje odpovědi na analytické otázky především prostřednictvím předspočítaných agregačních dat (samozřejmě na úkor flexibility)
- multidimenzionálnímu uspořádání dat se v souvislosti s OLAP říká obvykle „datové kostky“
- základní způsoby realizace multidimenzionální struktury
 - ROLAP
 - MOLAP
 - HOLAP



- ROLAP

- Využívá přímo dimenzionálního uspořádání dat v datovém skladu.
- Používají se přímo dotazy do struktury tabulek
 - vysoká flexibilita dotazů, která je ovšem vykoupená jejich podstatně vyšší časovou náročností

- MOLAP

- struktura založená na binární realizaci „datových kostek“, jejichž osy jsou klíčové atributy dimenzí
- kostky také obsahují již agregované údaje pro jednotlivé dimenze, resp. i pro podmnožiny dimenzí
 - rychlá odezva na dotazy, které odpovídají předpřipravené struktuře



- HOLAP

- Kombinace (H - hybridní) obou předchozích přístupů.
- Podle typu dotazu se používají buď data v tabulkách nebo v multidimenzionální struktuře



Základní operace v OLAP analýze

- Rozbalování (Drill-down)
 - zvyšování granularity pohledu na data (více detailů)
- Sbalování (Roll-up)
 - opak předešlé operace.
- Řezy (Slicing resp. Dicing)
 - fixování jedné, resp. více dimenzí
- Pivoting
 - změna struktury dimenzí, tj. jiné pohledy na data



Jazyk MDX

- jazyk pro získávání dat z multidimenzionálních databází
- podobná syntaxe jako v případě SQL, ale koncepce je jiná (poplatná multidimenzionálnímu charakteru databáze)
- je podporován řadou dodavatelů OLAP systémů

Distribuované databáze



Distribuované databáze

- databáze je zpracovávána na více než jednom počítači (serveru)
- cesta k řešení problému s kapacitou
- typy distribuovaných databází
 - rozdělování dat
 - replikace dat
 - kombinace rozdělování a replikace
- distribuované databáze s sebou přináší další komplikace (v porovnání s provozem na jednom serveru)
 - nutnost komunikačního prostředí, na jehož spolehlivosti a rychlosti závisí funkčnost dat. systému
 - potřeba synchronizace dat (zejména při replikaci), resp. problémy spojené s její konečnou rychlostí
 - komplikovanější problematika zamykáním a sním spojenými problémy
- distribuované databáze nacházejí uplatnění zejména v souvislosti s „Big Data“



Pojem „Big Data“

- Definice není jasně daná a jedná se v podstatě o „relativní“ pojem
- Data, která jsou charakteristická především následujícími vlastnostmi
 - **Objem** (Volume)
 - Exponenciální růst objemu dat
 - **Různorodost** (Variety)
 - **Rychlost**
 - Požadavek na rychlost zpracování i v situaci nárůstu rychlosti vzniku dat
- Další vlastnosti
 - Veracity – problém s věrohodností
 - Value – problém s identifikací důležitých dat
 - Vizualization – problém s vizualizací
 - Variability – různé významy podle kontextu
 - Volatility – časově omezená platnost dat



NoSQL databáze

- Jiné přístupy k práci s daty, než je uplatňován u relačních databází.
- Nejsou a ani nemohou být kladeny tak přísné požadavky na konzistenci a integritu dat.
- Jedná se o distribuované databáze.
- Rozvoj souvisí zejména s rozvojem internetových společností (Google, Facebook, Amazon ...).
- Existuje a vyvíjí se řada různých řešení a v této oblasti rozhodně není možné mluvit o standardizaci, jako je to v případě relačních databází.
- Obvykle jsou zaměřeny více na čtení dat , než na jejich modifikaci a zápisy.



Typy NoSQL databází

- Databáze typu klíč-hodnota
 - jen jednoduché operace, vysoká propustnost, vyhledávání jen podle klíčů
- Dokumentové databáze
 - slouží k ukládání dokumentů
 - formáty JSON, XML (samopopisné)
- Sloupcové databáze
 - podobné relačním, ale řádky mohou mít různý počet sloupců
- Grafové databáze
 - založeny na grafech, jakožto množinách uzlů propojených hranami
 - není předem daná struktura



Zdroje

- Holubová, Irena et al. Big Data a NoSQL databáze. První vydání. Praha: Grada, 2015. 281 stran. Profesional. ISBN 978-80-247-5466-6.
- Kroenke, David a Auer, David J. Databáze. 1. vyd. Brno: Computer Press, 2015. 496 s. ISBN 978-80-251-4352-0.
- GÁLA, L., POUR, J., ŠEDIVÁ, Z. Podniková informatika. Praha: Grada 2015. ISBN 978 80 2475457 4