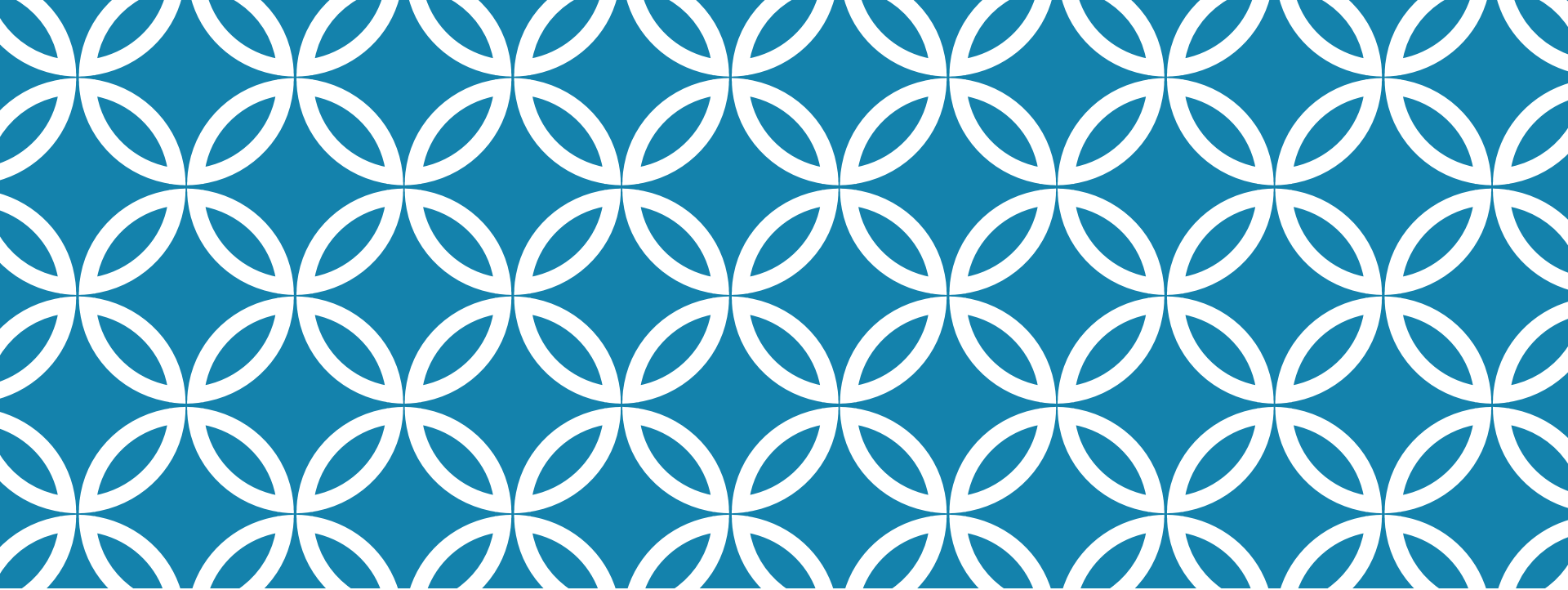


ISKB12 INFORMAČNÍ SYSTÉMY
- DATABÁZE

Ing. Mgr. Pavel Synek

8.3.2024

DNEŠNÍ AGENDA

1. Databáze a jejich historie
2. Modely
3. Architektura IS
4. Analýza a tvorba IS
5. Abstrakce vs. Konkretizace

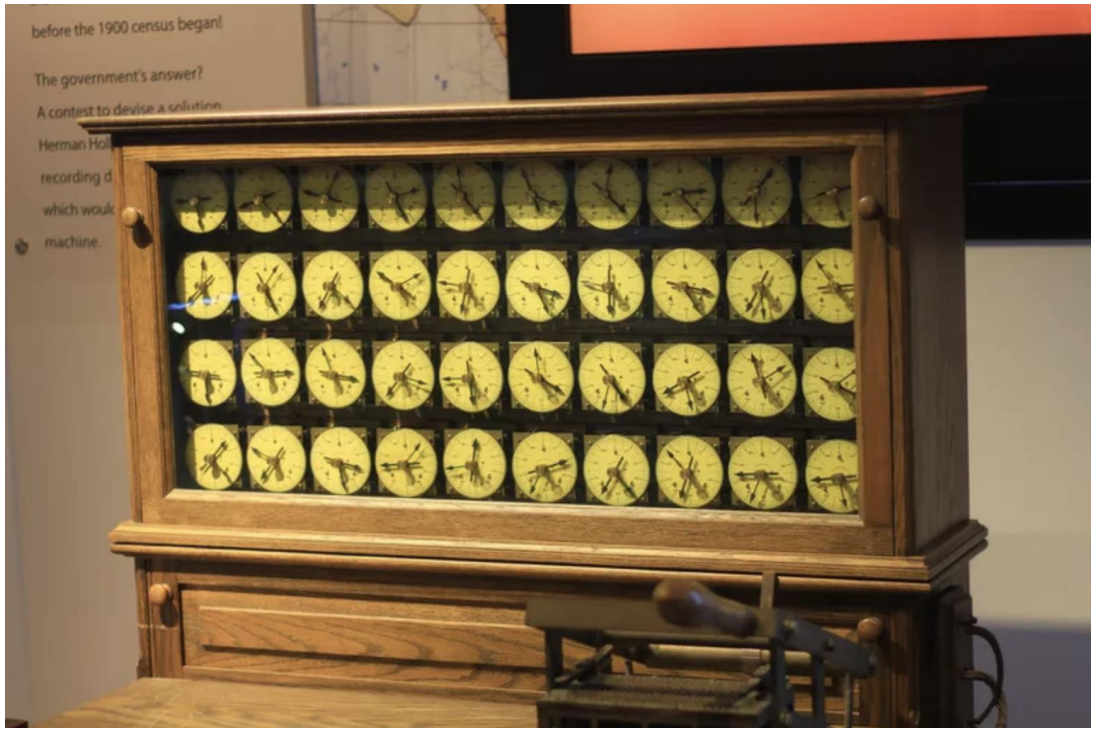
DATABÁZE

- Nárůst množství dat a vzájemných vazeb - potřeba je hierarchizovat a spravovat
- Databáze = soubor dat organizovaných dle předem stanoveného řádu uložených k dalšímu zpracování
- Databáze je zjednodušeným modelem reálného světa a popisuje děje v něm probíhající
- Rozsáhlejší systémy pro zpracování dat se pak nazývají informační systémy, které můžeme definovat jako „systémy pro sběr, uchovávání, vyhledávání a zpracovávání dat za účelem poskytování informací“.

DATABÁZE

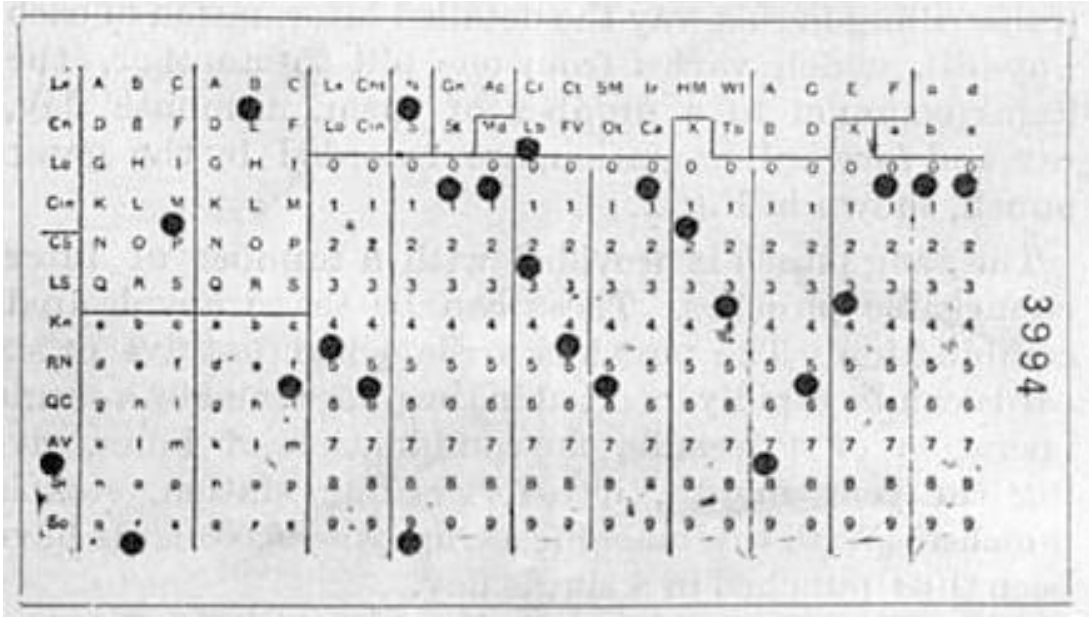
Knihovní katalog

DATABÁZE



Herman Hollerith`s Tabulating Machine - 1890

DATABÁZE



Děrný štítek

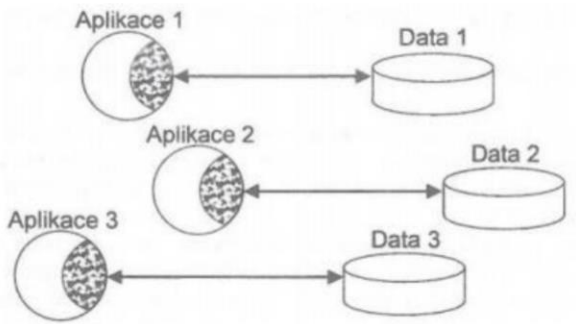
DATABÁZE

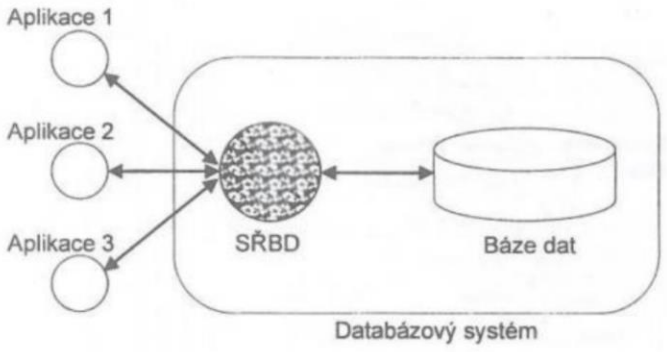
- 1888 – Oberlin Smith publikoval článek "Some Possible Forms of the Phonograph" v časopisu Electrical World, kde představuje tezi o technologickém řešení záznamu informací na elektromagnetický nosič
- 1959 - první jazyk pro databázové systémy - COBOL (common business-oriented language)
- 1965 - První Systémy řízení báze dat provozované na sálových počítačích
- 1969 – Edgar. F. Codd : Návrh relačního datového modelu
- 1974 - vzniká databázový jazyk Sequel (později SQL)
- 1980 - První SQL datáze Oracle, spol. Relational Software, Inc. (1980),
- následovala DB2 od IBM (a řada další Progres, Informxix, SyBase,...).

HISTORIE DATABÁZOVÉHO ZPRACOVÁNÍ

- V historii hromadného zpracování dat lze sledovat dva přístupy k datům:
  - Souborový přístup – data uložena do jednoho nebo více datových souborů uložených na paměťovém médiu, součástí souboru dat je i jejich popis. Aplikace je přímo navázána na strukturalizaci dat. Toto řešení přineslo celou řadu problémů (redundance, izolovanost dat, nekonzistence dat, nezajištění integrity dat, nemožnost přístupu více uživatelů, atd.)
  - Databázový přístup – odstraňuje nevýhody souborového přístupu. Báze dat je řízena Systémem řízení báze dat.

SOUBOROVÝ VS. DATABÁZOVÝ PŘÍSTUP





VÝHODY DATABÁZOVÉHO PŘÍSTUPU

- Zamezení redundance
- Zajištění konzistence dat
- Integrita dat
- Sdílení dat
- Ochrana dat před zneužitím
- Nezávislost dat na aplikaci
- Přístup k datům výhradně prostřednictvím databázových programů
- Možnost využití grafických přehledů
- Možnost ukládání velkých objemů dat
- Využití jazyka SQL jako standardu pro práci s daty

SYSTÉM ŘÍZENÍ BÁZE DAT (SŘBD)

- SŘBD je programový systém, který umožňuje definování struktury, ukládání, výběr a ochranu dat, zabezpečuje databázi a komunikaci mezi uživatelem a systémem.
- Zjednodušeně jde tedy o softwarový prostředek, který řídí sdílený přístup k bázi dat a poskytuje mechanismy určené k zajištění bezpečnosti a integrity dat.
- SŘBD tvoří souhrn procedur a datových struktur, které zajišťují nezávislost databázových aplikací na detailech vytváření, výběru, uchování, modifikaci a zabezpečení ochrany databází na fyzických paměťových strukturách počítače.
- Příklady SŘBD: Microsoft SQL server, My SQL, Oracle, Informix, SyBase, Microsoft Access a další.

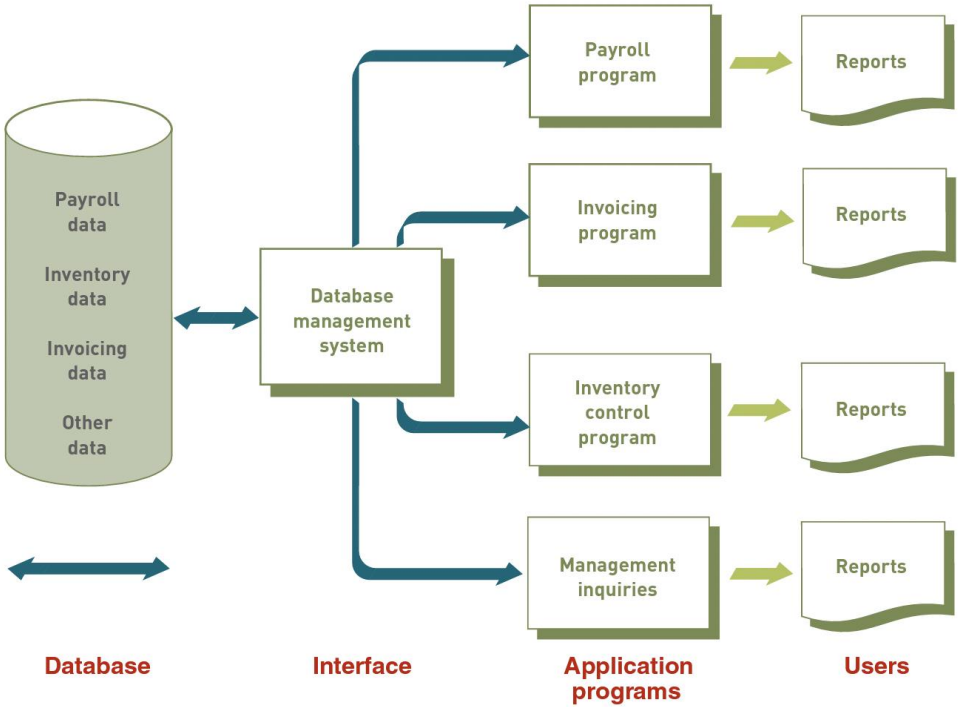
SYSTÉM ŘÍZENÍ BÁZE DAT (SŘBD)

- SŘBD umožňují:
  - vytvoření báze dat
  - vkládání dat
  - aktualizace dat
  - rušení dat
  - výběr z báze dat
  - tvorbu vstupních a výstupních formulářů, výstupních sestav a vytváření aplikací.

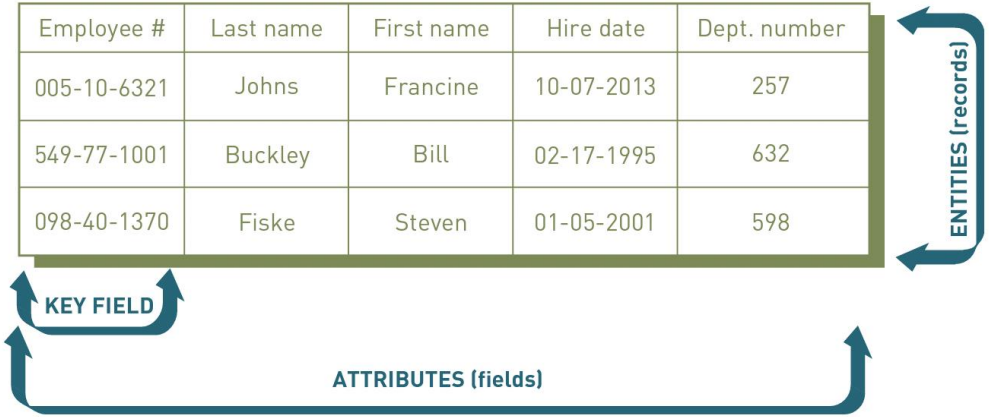
SYSTÉM ŘÍZENÍ BÁZE DAT (SŘBD)

- Systém řízení báze dat zahrnuje:
  - **Jazyk pro definici dat** – DDL (data definition language) – prostředky pro popis dat, sloužící k vytvoření všech definic uživatelských dat potřebných v aplikaci, včetně určení omezujících podmínek.
  - **Jazyk pro manipulaci s daty** – DML (data manipulation language) – prostředky pro popis algoritmu, které se používají k aktualizaci dat (přidávání, změny a rušení dat) a k výběru dat z databáze na základě kladených požadavků.

DATABÁZOVÝ SYSTÉM = BÁZE DAT + SYSTÉM ŘÍZENÍ BÁZE DAT



DATABÁZE


Employee #
Last name
First name
Hire date
Dept. number
005-10-6321
Johns
Francine
10-07-2013
257
549-77-1001
Buckley
Bill
02-17-1995
632
098-40-1370
Fiske
Steven
01-05-2001
598
ENTITIES (records)
KEY FIELD
ATTRIBUTES (fields)

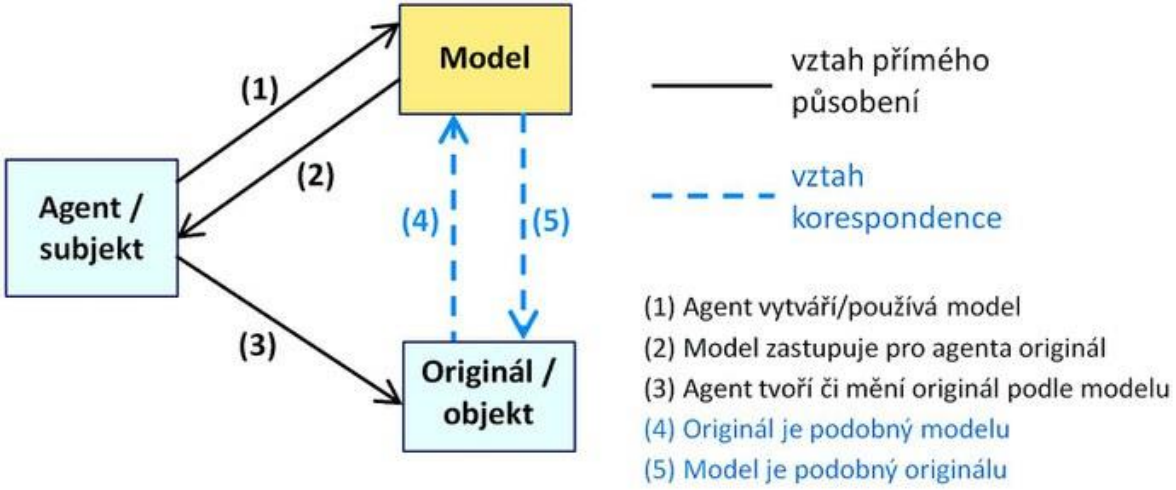
BÁZE DAT (BD) A.K.A. DATABÁZE

- BD je množina souborů a jejich popisu, které jsou vzájemně v určitém logickém vztahu.
- Jde o komplikovanou centrálně zpracovávanou strukturu dat.
- Pro databáze je vytvořena jediná interní organizace dat, společná pro všechny oblasti a způsoby využití.

CO JE MODEL?

- Model budeme chápat jako umělý výtvor, tj. artefakt, záměrně za určitým cílem vytvářený agentem ve funkci subjektu. Tímto výtvorem může být jakákoli myšlenková nebo materiální konstrukce. Od ostatních umělých výtvorů se model odlišuje svým specifickým účelem, jímž je reprezentace originálu ve funkci objektu (předmětu).

CO JE MODEL?



Znázornění modelu
https://knihovnarevue.nkp.cz/archiv/2018-2/recenzovane-prispevky/pojem-modelu-a-pojmovy-model-v-informacni-vede

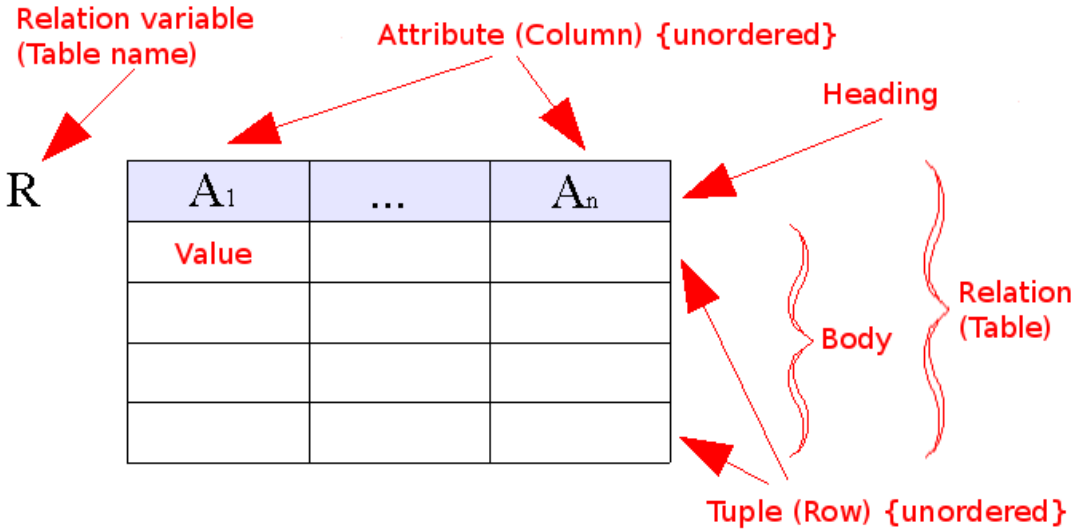
JAKÉ JSOU DRUHY MODELŮ?

- Mentální model
- Konceptuální model
- Matematický model
- Procesní model
- Fyzický model
- Architektonický model
- Datový model
- ….

DATABÁZOVÉ MODELY

- Databázový model je typ datového modelu, který určuje logickou strukturu databáze a zásadně určuje, jakým způsobem lze data ukládat, organizovat a manipulovat.
- Podle typu modelovaných vztahů mezi záznamy se v databázi na technologické úrovni rozlišují následující modely:
  - relační
  - hierarchický
  - síťový
  - objektový model

RELAČNÍ DATOVÝ MODEL



RELAČNÍ DATOVÝ MODEL

- Relace = tabulka se sloupci a řádky vyjadřující jejich vzájemný vztah/vazby
- Atribut = pojmenovaný sloupec s nějakou "relací"
- Doména = rozsah hodnot přípustných v daném atributu
- N-tice (Tuple) = řádek tabulky

- Výhody - přirozená reprezentace dat, jednoduché zachycení struktury a vazeb

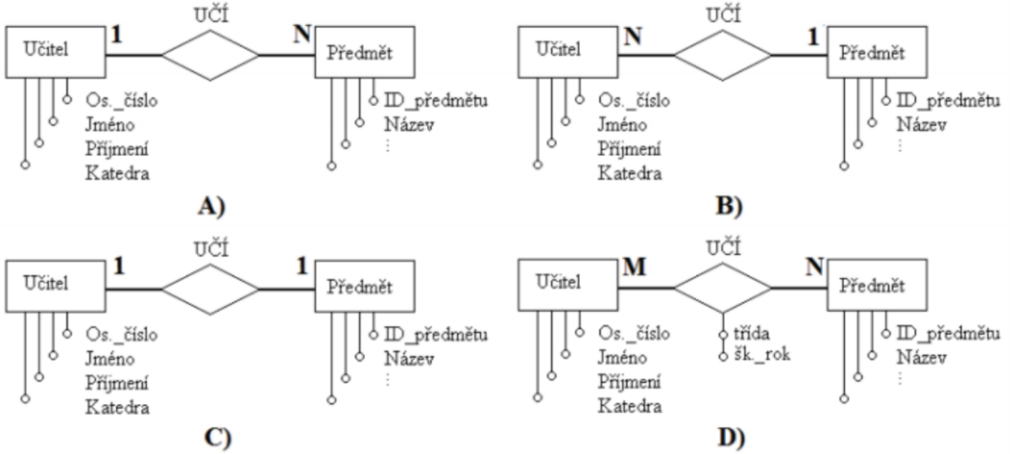
RELAČNÍ MODEL

- Nejrozšířenější
- Relační databázový model má jednoduchou strukturu, kdy data jsou organizována v tabulkách, které se skládají z řádků a sloupců.
- Všechny databázové operace jsou pak prováděny na těchto tabulkách.

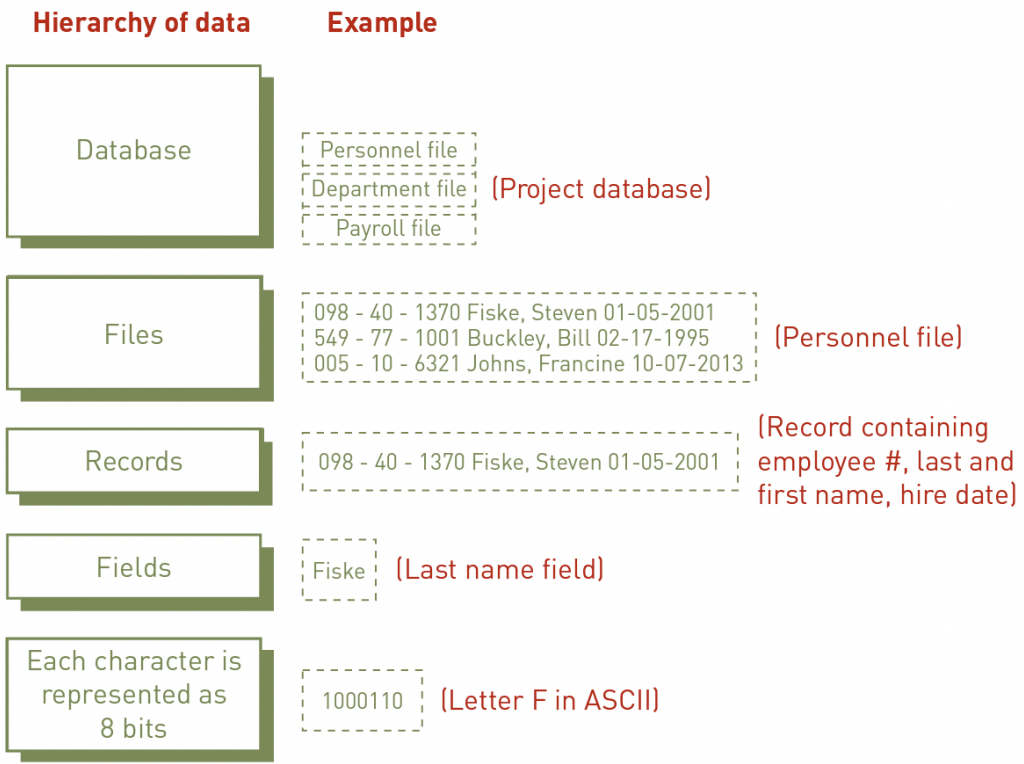
RELAČNÍ DATOVÝ MODEL

- Příklady:
  - **Knihovna**: Uložená kniha, autor, datum vydání a další informace.
  - **E-shop**: Uložený produkt, cena, popis a další informace o produktu.
  - **Sociální síť**: Uložená uživatel, profil, příspěvek a další informace o uživatelích.
- Příklady systémů:
  - MySQL
  - PostgreSQL
  - Oracle
  - ...

RELAČNÍ MODEL



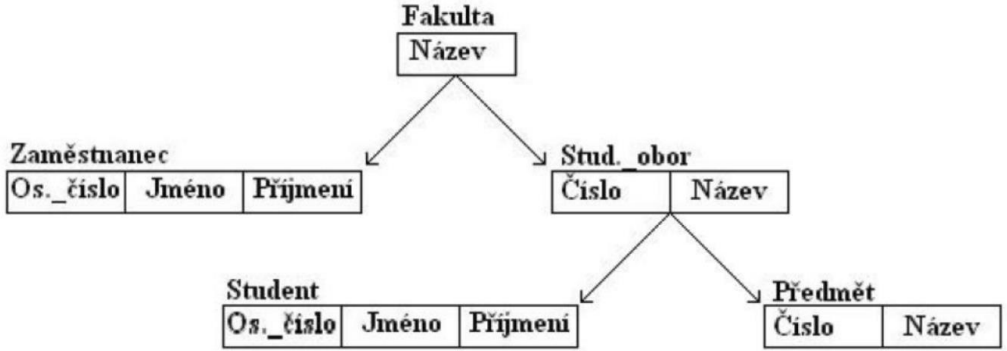
HIERARCHIE DAT



HIERARCHICKÝ MODEL

- Logické uspořádání dat v tomto modelu je založeno na stromové struktuře, která umožňuje vyjádřit jednosměrné vztahy typu 1 - více ve směru shora dolů.
- Každý záznam představuje uzel ve stromové struktuře a vzájemný vztah mezi záznamy je typu Rodič/Potomek.
- Použití hierarchického modelu je vhodné tam, kde i realita má hierarchickou strukturu, např. organizační či skladové systémy.
- Nalezení dat v této struktuře vyžaduje navigaci přes záznamy směrem dolů (Potomek), nahoru (Rodič), do stran (Sourozenec).

HIERARCHICKÝ MODEL

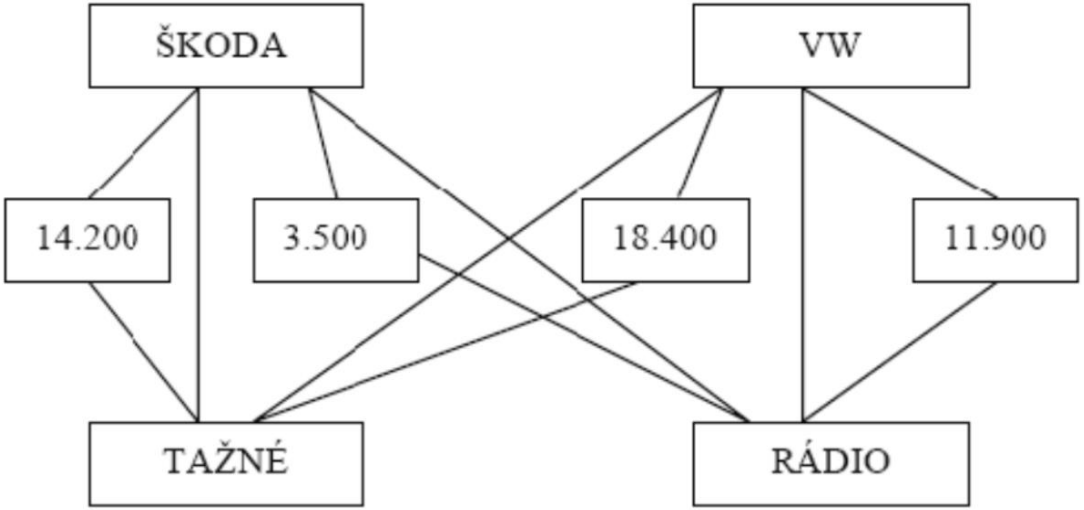


HIERARCHICKÝ MODEL

- Příklady:
  - Dublin Core
  - MARC 21

SÍŤOVÝ MODEL

- V tomto modelu jsou data logicky i fyzicky uspořádána jako uzly rovinného grafu, v němž může být každý záznam spojený s libovolným počtem dalších záznamů.
- Tím může obsahovat odkazy na jiné entity v bázi dat, které s ní logicky souvisí – je tedy možné využít mnohonásobného rodičovství.
- Jde v podstatě o zobecněný hierarchický model, který je doplněn o mnohonásobné vztahy. Jednotlivé entity tak mohou vytvořit síť s velmi rozmanitou strukturou.



SÍŤOVÝ MODEL

SÍŤOVÝ MODEL

- Příklady:
  - Modely sociálních sítí a vazeb
  - Dopravní sítě
  - Molekulární struktury
  - FRBR - Funkční model bibliografických záznamů, který definuje entity a vztahy mezi nimi v bibliografických datech

OBJEKTOVÝ MODEL

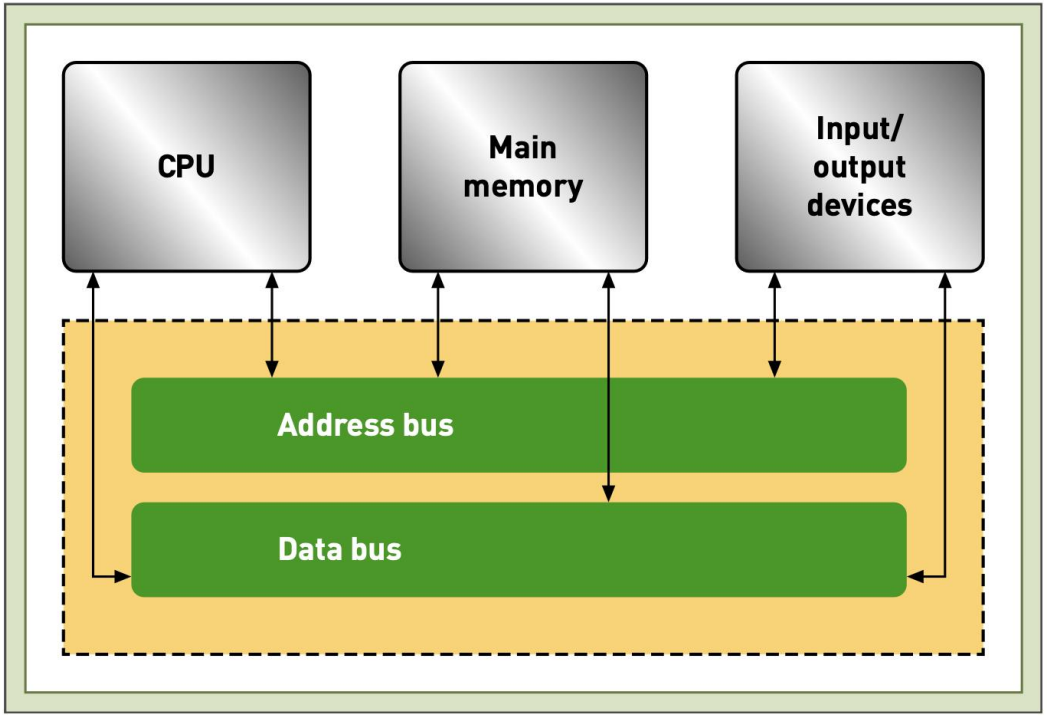
- Pod obecné označení „objektové databáze“ patří dva odlišné datové modely:
  - Objektově relační datový model, který je doplněním relačního datového modelu o možnost práce s některými datovými strukturami známé z oblasti objektově orientovaných programovacích jazyků. Objektově relační datový model však ve svých principech zůstává původním relačním datovým modelem.
  - Objektově orientovaný datový model je nový datový model, který není odvozen od relačního datového modelu. Do jisté míry jde o přetvoření a vylepšení původního síťového datového modelu, který je doplněn o možnost práce s objekty známými z objektového programování.

OBJEKTOVÝ MODEL

- Data = objekty
- Každý objekt obsahuje data (atributy) a má nadefinované určité chování (metody)
- Příklady:
  - GUI (tlačítka a funkce na webové stránce)
  - CMS
- Nevýhody:
  - Přílišná složitost a ve výsledku I snížená výkonnost při využívání

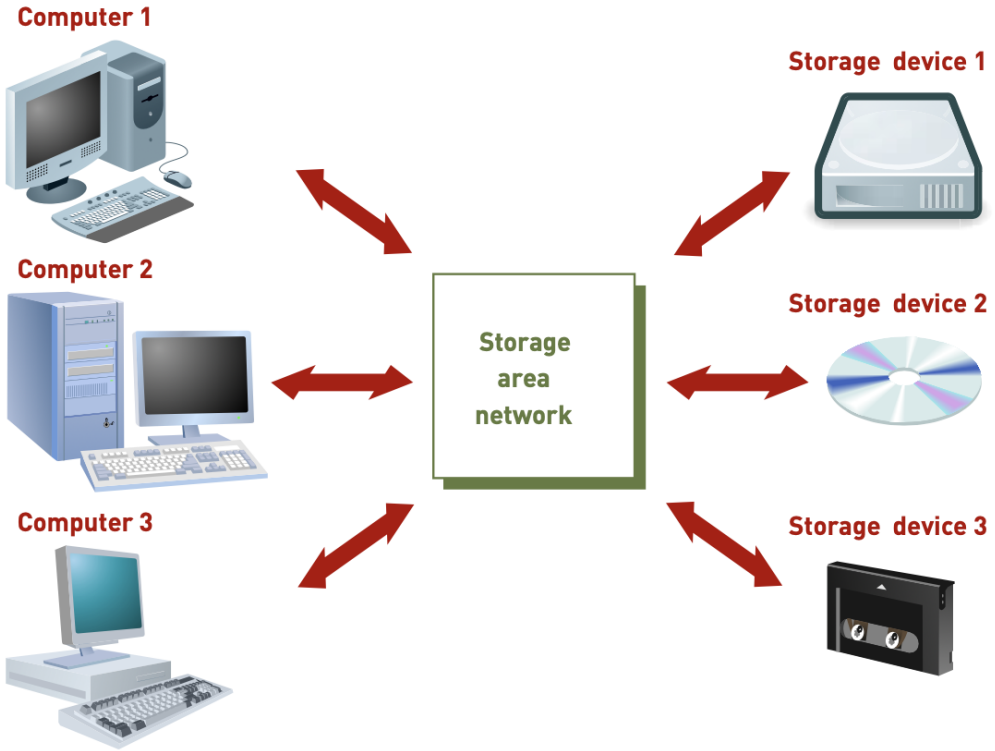
ARCHITEKTURA DATABÁZOVÝCH SYSTÉMŮ

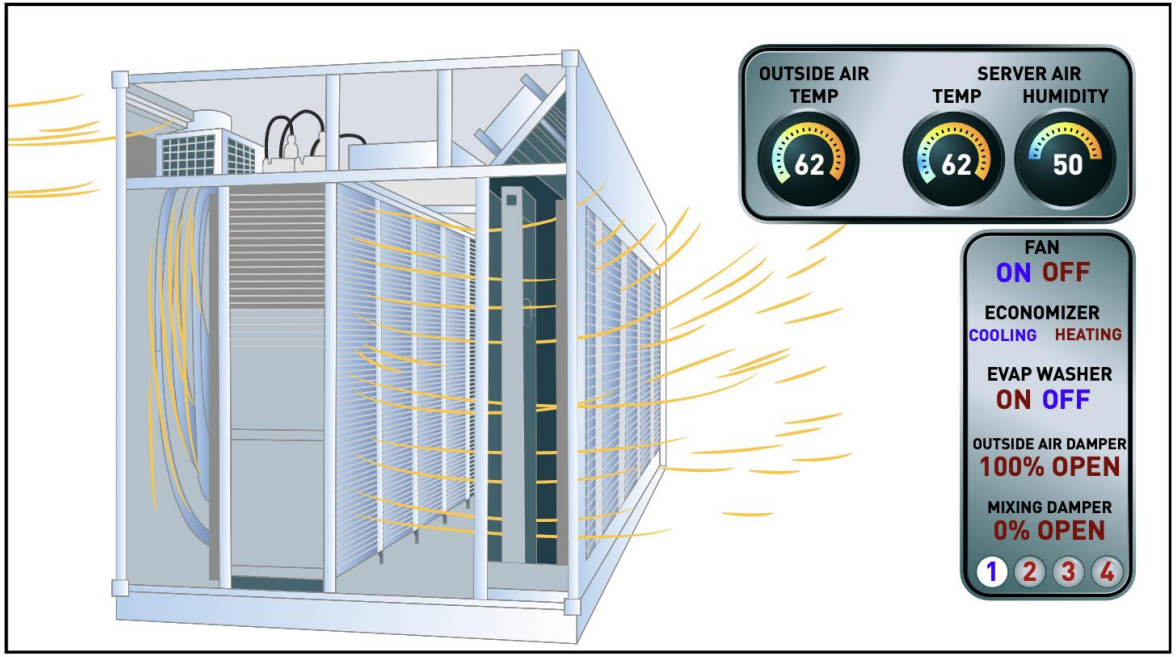
ZJEDNODUŠENÁ ANATOMIE POČÍTAČE



STORAGE AREA NETWORK

Dedikovaná datová síť, která umožňuje připojení externích datových nosičů se serverem.

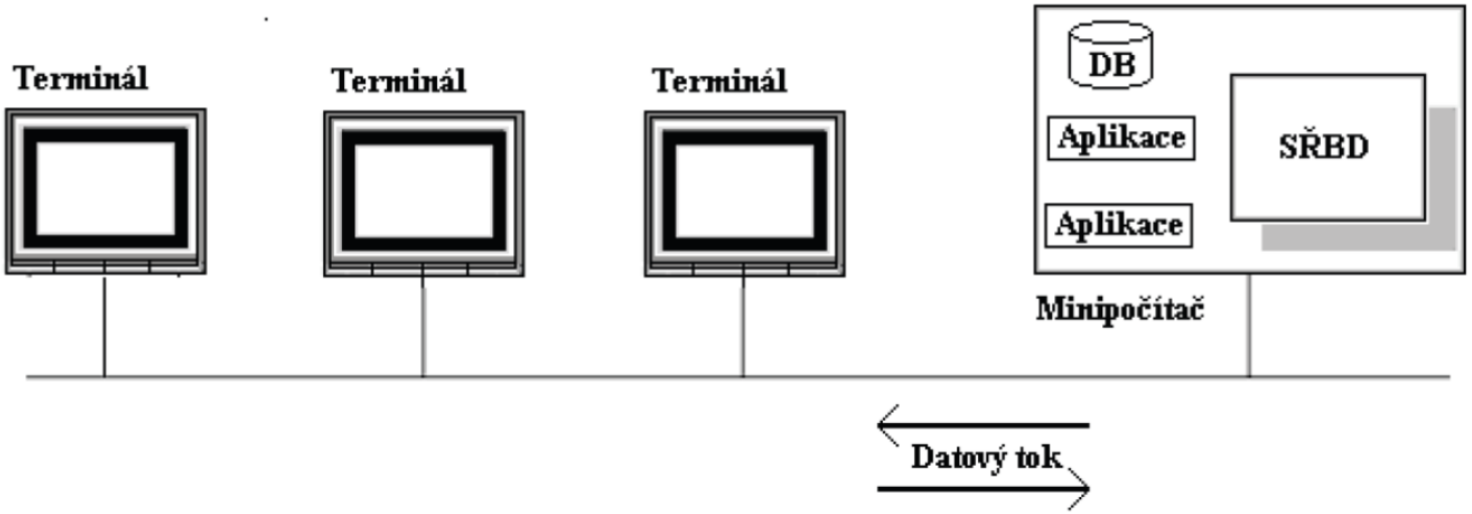




MODULÁRNÍ DATOVÉ CENTRUM

ARCHITEKTURA DATABÁZOVÝCH SYSTÉMŮ

- Rozlišujeme následující typy databázových architektur:
  - Jednovrstvá (centralizovaná) architektura
  - Dvouvrstvá architektura (File-Server nebo Klient-Server)
  - Vícevrstvá architektura



JEDNOVRSTVÁ (CENTRALIZOVANÁ) ARCHITKTURA

JEDNOVRSTVÁ (CENTRALIZOVANÁ) ARCHITEKTURA

- Zastaralý model architktury db systémů s použitím centrálního počítače
- BD dat i SŘBD jsou společně na centrálním počítači a komunikaci s uživatelem zprostředkovává pouze terminál umístěný na pracovišti uživatele.
- Vstupní data a požadavky se přenášejí z terminálu po síti do centrálního počítače, kde dochází ke zpracování a následnému zpětnému odeslání na terminál uživatele.

JEDNOVRSTVÁ (CENTRALIZOVANÁ) ARCHITEKTURA

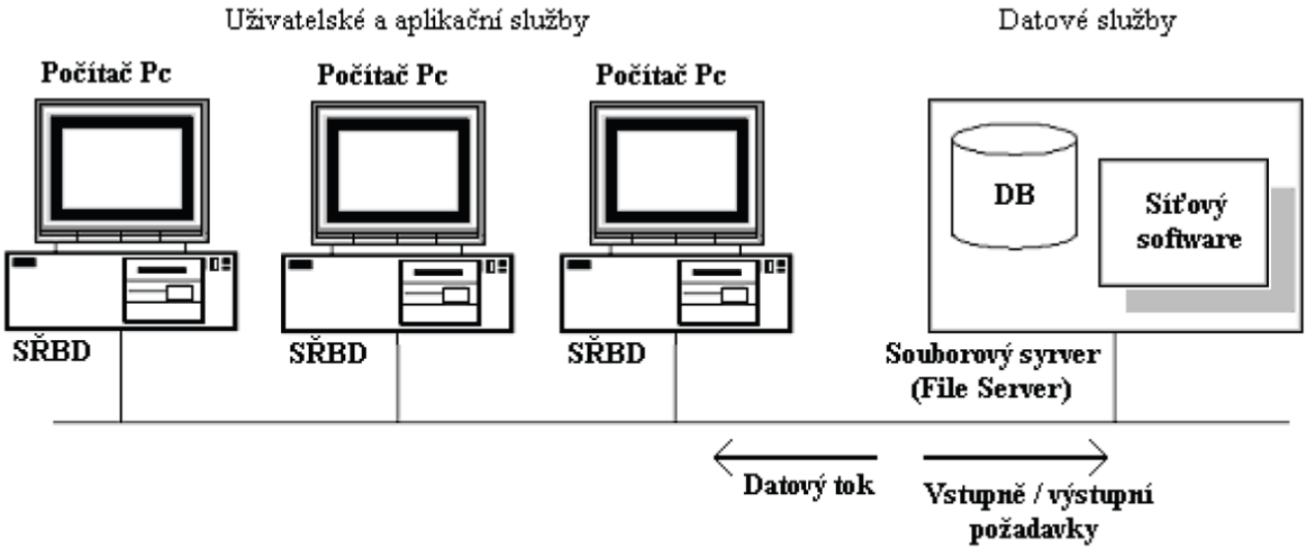
- Vhodné pro aplikace (databáze), ve kterých se informace nesdílejí mezi několika uživateli.
- Řešení má výhodu v rychlosti přístupu a zpracování dat, protože data jsou uložena lokálně a nevyžadují samostatný databázový server a počítačovou síť.

JEDNOVRSTVÁ (CENTRALIZOVANÁ) ARCHITEKTURA

- Všechny komponenty v jedné vrstvě
- Vhodné pro webové stránky s nízkým počtem funkcí, nebo malé desktopové aplikace
- Špatně se škáluje

DVOUVRSTVÁ ARCHITEKTURA

- Tvorbou dvouvrstvých aplikací poskytneme víceuživatelskou podporu a získáme možnost používat velké vzdálené databáze, které mohou ukládat značně velké množství informací.
- Dvě vrstvy = prezenční a datová
- Dvouvrstvou architekturu lze rozdělit do dvou skupin:
  - Architektura File-Server - výkon spojený s aplikačními službami je umístěn na straně klienta
  - Architektura Klient-Server - výkon spojený s aplikačními službami je umístěn na straně serveru



ARCHITEKTURA FILE-SERVER

ARCHITEKTURA FILE-SERVER

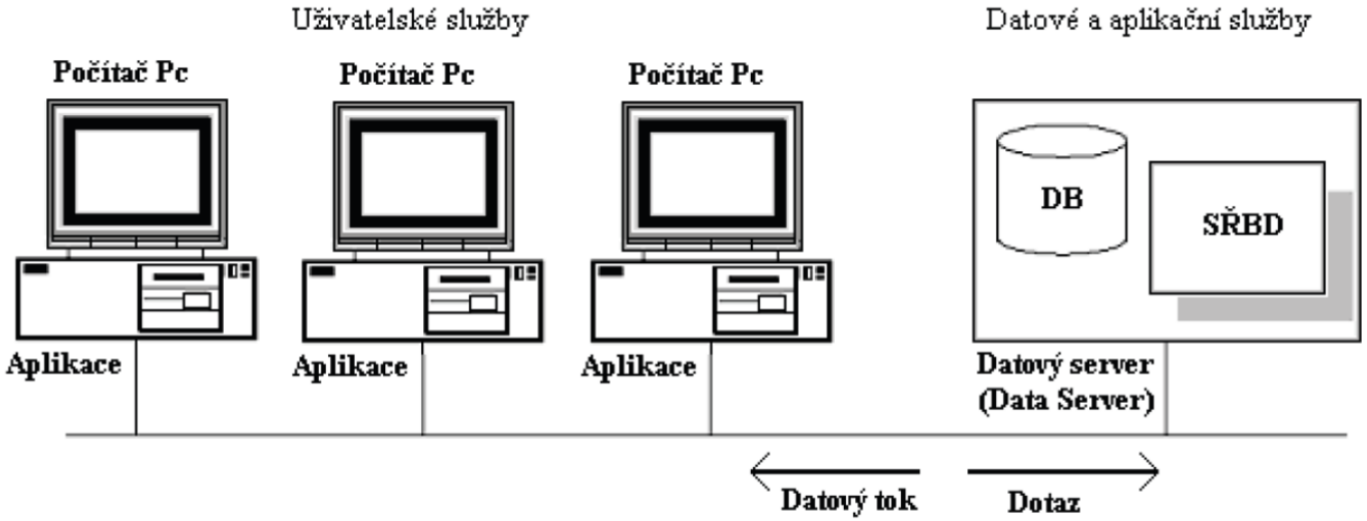
- Databáze (DB) s daty je umístěna na počítači, který pracuje jako File-Server a prostřednictvím sítě a jednotlivých systémů řízení báze dat (SŘBD), umístěných na počítačích uživatelů, poskytuje jednotlivá data a umožňuje sdílení.
- V tomto případě může k datům přistupovat více aplikací v podobě SŘBD najednou a proto je nutné zajistit ochranu používaných záznamů.

ARCHITEKTURA FILE-SERVER

- Nevýhodou je, že veškeré aplikační a uživatelské služby se zpracovávají u klienta, kterého v tomto případě nazýváme Tlustý klient.
- Velkou slabinou této architektury jsou rovněž velké nároky na přenosové kapacity, kdy mezi klientem a serverem musí probíhat velký počet datových přenosů.

ARCHITEKTURA FILE-SERVER

- Příklady:
  - Ulož.to
  - MS Excel v počítači



ARCHITEKTURA KLIENT-SERVER

ARCHITEKTURA KLIENT-SERVER

- U této architektury běží na počítačích aplikace, které předávají dotazy a požadavky na datový server, který je zpracovává a výsledky posílá zpět na počítač uživatele.
- Server je v tuto chvíli nejvíce zatíženým počítačem, protože na něm běží SŘBD, který vše zpracovává.

ARCHITEKTURA KLIENT-SERVER

- K uživateli jsou přesunuty pouze uživatelské služby a získává pouze požadované informace. V tomto případě hovoříme, že uživatel je tzv. Tenký klient.
- Aplikační a datové služby probíhají na serveru. Tato architektura snižuje požadavky na množství dat pohybujících se v síti a tím vyhovuje i rozsáhlým aplikacím.
- Výhodou je tedy minimální zatížení sítě, vysoká pružnost aplikací a rozdělení zpracování záznamů.

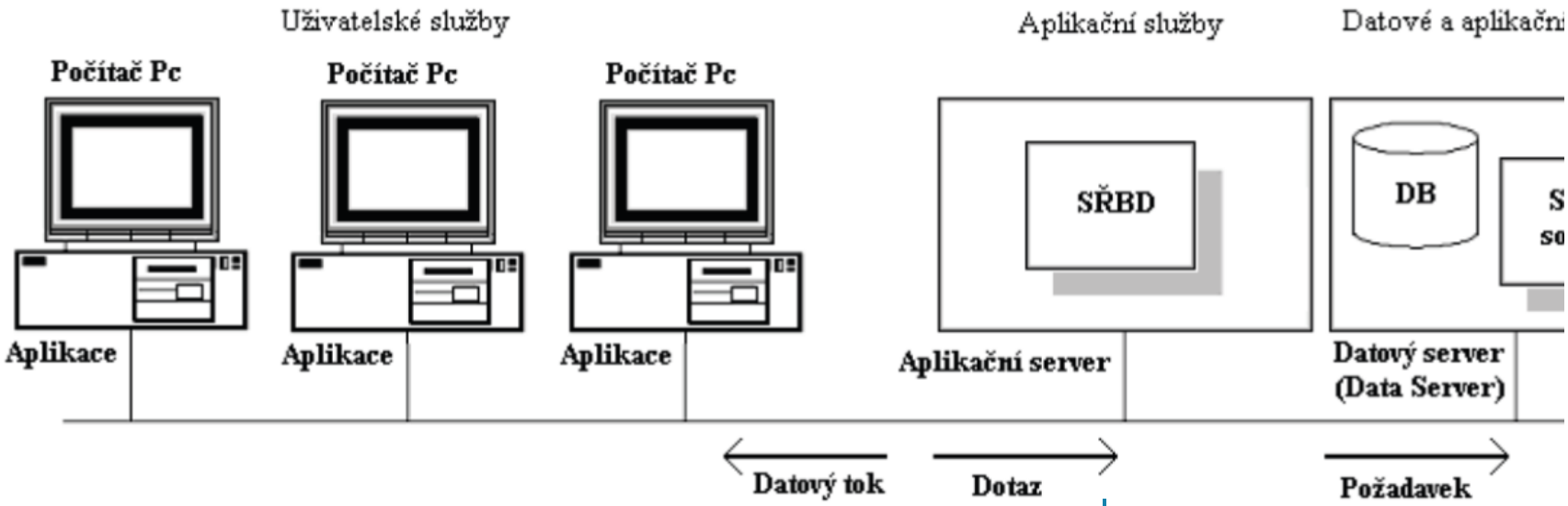
ROZDÍL MEZI FILE-SERVER A KLIENT-SERVER

Architektura Klient-Server redukuje přenos dat po síti, protože dotazy jsou prováděny přímo na databázovém serveru a na počítač jsou posílány pouze výsledky. Např. pokud je mezi 10 000 záznamy pouze 100 záznamů, které splňují podmínku dotazu, pak na personální počítač putuje pouze těchto 100 záznamů.

V případě architektury File-Server je však nutné poslat všech 10 000 záznamů na personální počítač (tedy celou tabulku – blok), tam se teprve provede dotaz a zpracuje nalezených 100 záznamů.

VÍCEVRSTVÁ ARCHITEKTURA

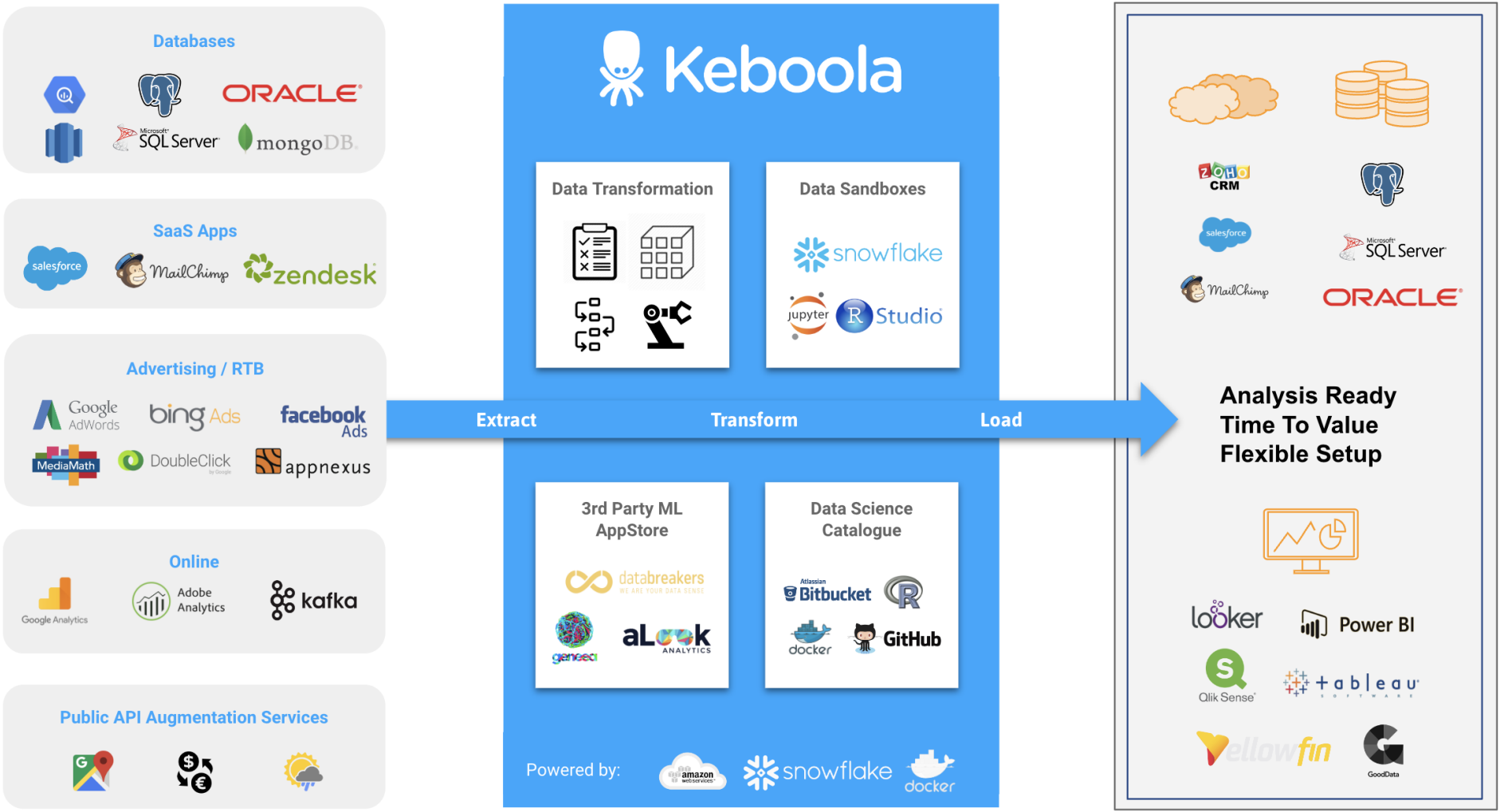
- Obsahují-li databázové informace komplikované vazby mezi několika tabulkami nebo zvyšuje-li se počet uživatelů (klientů), pak lze používat vícevrstvé aplikace.
- Vícevrstvé aplikace obsahují střední vrstvu, která centralizuje logiku ovládání databázových interakcí (umožňuje různým klientským aplikacím používat stejná data se zajištěním konzistentní datové logiky).

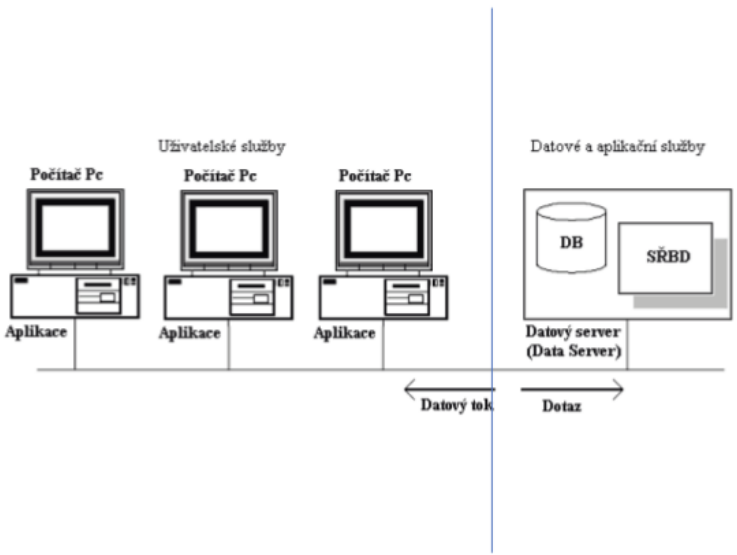


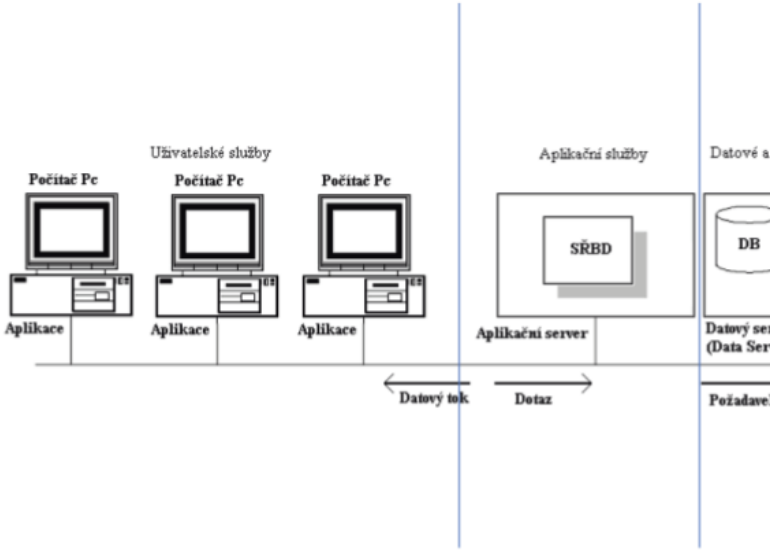
VÍCEVRSTVÁ ARCHITEKTURA

VÍCEVRSTVÁ ARCHITEKTURA

- U vícevrstvé architektury lze shledat určitou podobnost s již uvedeným modelem architektury Klient-Server, kdy je výkon spojený s aplikačními službami soustředěn na serveru.
- Klient, potažmo uživatel pracuje pouze s uživatelským rozhraním, přičemž datové a aplikační služby jsou od sebe odděleny do samostatných logických celků, které mohou být umístěny buď na stejném serveru, nebo na dvou různých serverech.
- Vícevrstvá architektura umožňuje získat vyšší úroveň stability, protože provozní zátěž může být rozložena na dva nebo více servery.


Databases
ORACLE
SQL Server
mongoDB
SaaS Apps
salesforce
MailChimp
zendesk
Advertising / RTB
Google AdWords
bing Ads
facebook Ads
MediaMath
DoubleClick
appnexus
Online
Google Analytics
Adobe Analytics
kafka
Public API Augmentation Services
Keboola
Data Transformation
Data Sandboxes
snowflake
jupyter
R Studio
3rd Party ML AppStore
databreakers
aLook ANALYTICS
Data Science Catalogue
Atlassian Bitbucket
docker
GitHub
Powered by:
amazon web services
snowflake
docker
Extract
Transform
Load
ZOHO CRM
salesforce
MailChimp
Microsoft SQL Server
ORACLE
Analysis Ready
Time To Value
Flexible Setup
looker
Power BI
Qlik Sense
tableau
yellowfin
GoodData

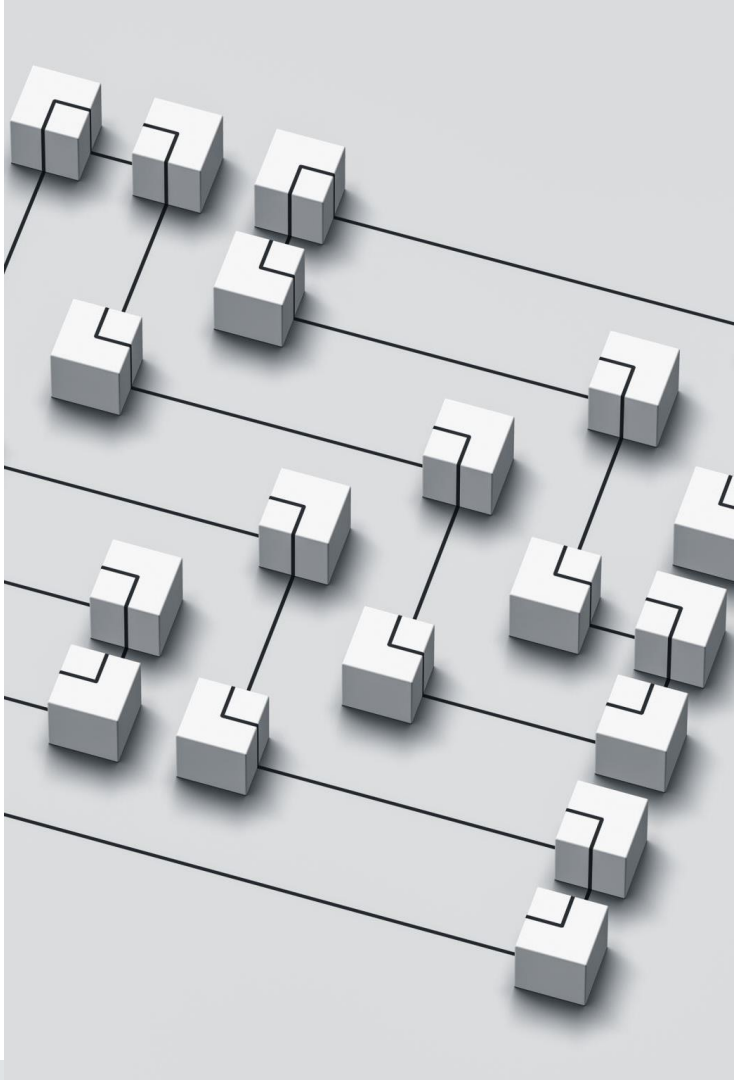




DVOUVRSTVÁ VS. TŘÍVRSTVÁ ARCHITEKTURA

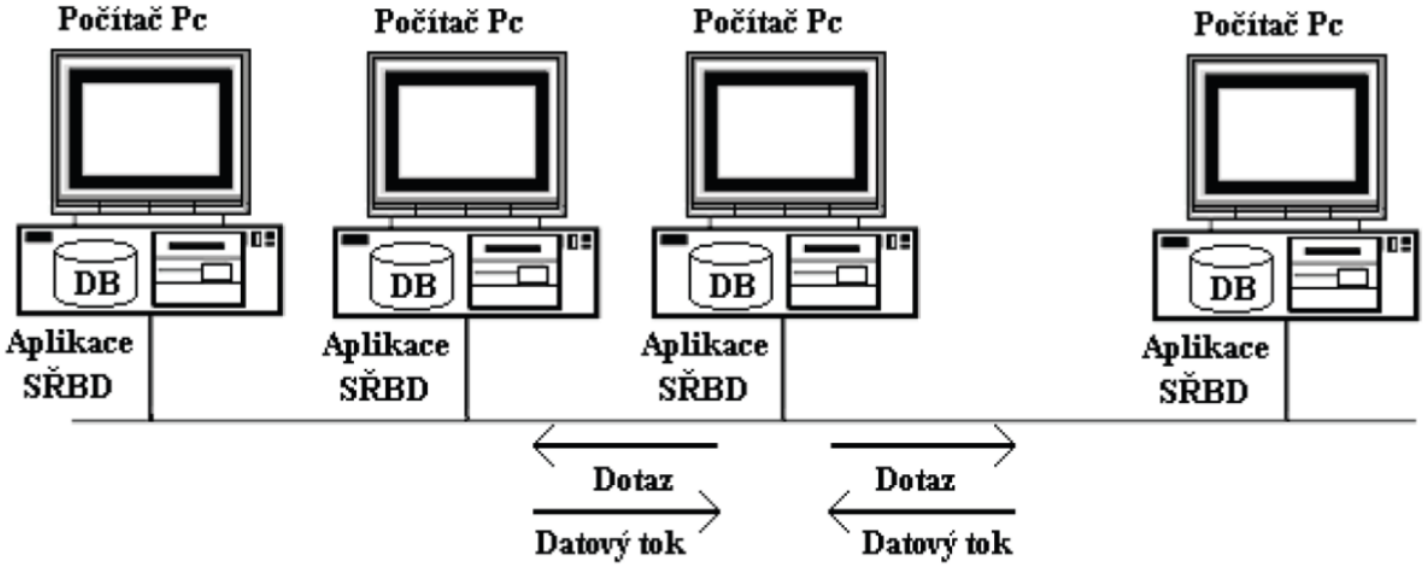
ARCHITEKTURA DISTRIBUOVANÝCH DB SYSTÉMŮ

- Distribuovaná databáze je kolekce dat, které logicky patří do jednoho systému, ale fyzicky jsou rozprostřena mezi jednotlivými místy (uzly) počítačové sítě.
- K rozvoji DDBS vedla řada faktorů:
  - distribuovaná povaha některých databázových aplikací
  - zvýšená spolehlivost a dostupnost
  - řízené sdílení dat
  - zvýšený výkon.

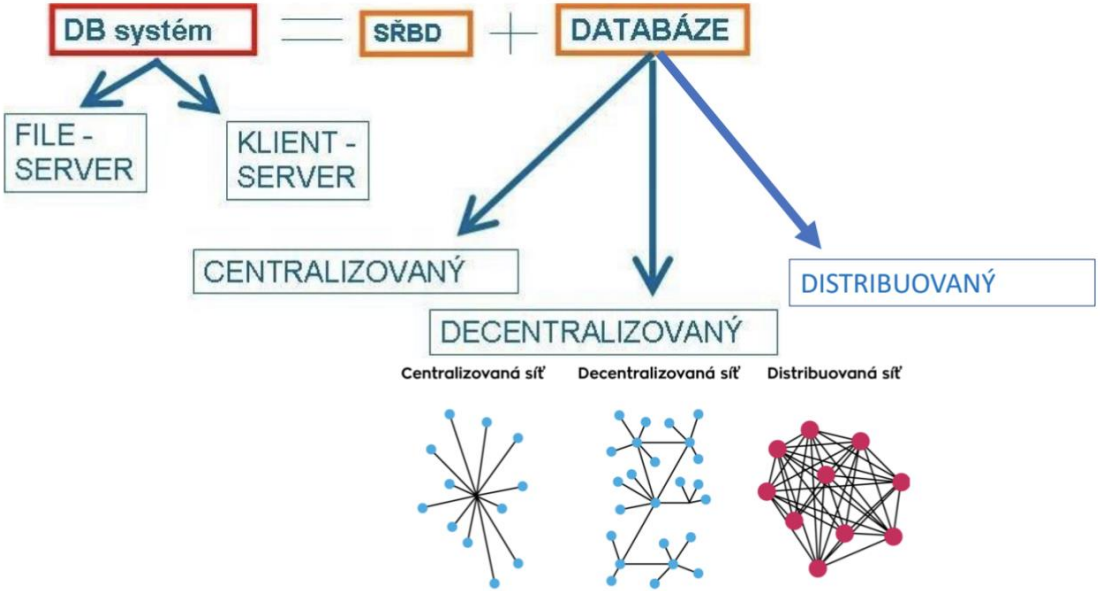


ARCHITEKTURA DISTRIBUOVANÝCH DB SYSTÉMŮ

- Při použití této architektury jsou data rozložena v několika počítačích, což znamená, že databáze je rozdělena do několika částí.
  - Navenek se však uživateli jeví jako jediná celistvá databáze.
  - Distribuované databázové systémy se vyznačují třemi základními vlastnostmi:
  - Transparentnost
  - Autonomnost
  - Nezávislost na typu sítě



ARCHITEKTURA DISTRIBUOVANÝCH SYSTÉMŮ



KONTEXT DBS = SŘBD + DB

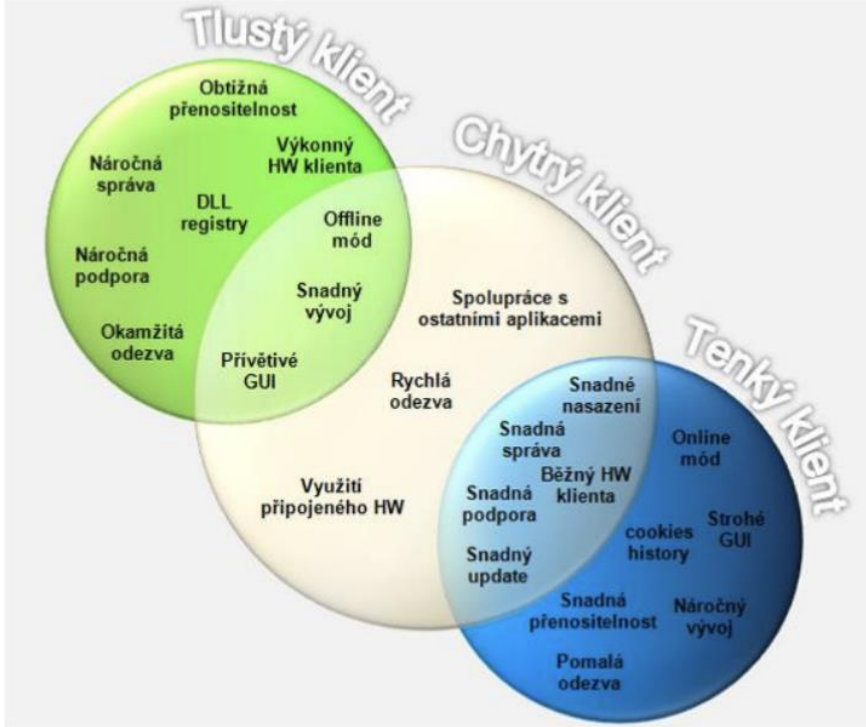
TYPY SW KLIENTŮ A JEJICH VÝHODY A NEVÝHODY

- **Tlustý klient** v sobě obvykle obsahuje jak presentační tak i aplikační vrstvu a připojuje se přímo k databázovému nebo jinému serveru. Další typickou vlastností tlustého klienta je, že si přes síť stahuje velký objem dat, která zpracuje a výsledek pak přenese zpět na server.
- **Tenkým klientem** je obvykle webový prohlížeč, který s presentační vrstvou komunikuje přes bez stavový HTTP protokol a stará se tak pouze o zobrazování dat. Na tenkém klientovi neprobíhá žádná rozhodovací logika, pokud nebereme v úvahu např. validaci dat, zadávaných do webového formuláře.

TYPY SW KLIENTŮ A JEJICH VÝHODY A NEVÝHODY

- Chytrý klient vhodně kombinuje výhody tenkého a tlustého klienta a potlačuje jejich nevýhody. Aby mohl pracovat off-line, obsahuje určitou logiku a drží data, která se v okamžiku navázání spojení synchronizují. Využívá místní systémové zdroje jako je např. paměť, procesor a diskový prostor, ale může komunikovat a využívat I připojených zařízení.

Vlastnost/klient	Tlustý klient (desktopová aplikace)		Chytrý klient		Tenký klient (webová aplikace)	
Nároky na HW konfiguraci klienta	-	vysoké, je potřeba výkonný HW	0	postačí běžný HW	+	nízké, není potřeba výkonný HW
Nároky na HW konfiguraci serveru	+	nízké, není potřeba výkonný HW	0	postačí běžný HW	-	vysoké, je potřeba výkonný HW
Objem přenesených dat	-	značný, neboť data se zpracovávají lokálně	0	malý, pořízená data se musí přenést na server	+	nepatrný, neboť data se zpracovávají na serveru
Nároky na vývoj aplikace	+	nízké, snadný a levný vývoj	0	střední, snadný, ale drahý vývoj	-	vysoké, náročný a drahý vývoj
Nároky na nasazení aplikace	-	časově náročný instalační proces setup.exe	0	nainstalovaný po zadání URL	+	k dispozici po připojení na dané URL
Nároky na update aplikace	-	nutno naplánovat způsob jak update proběhne	0	po připojení proběhne automatické update	+	zobrazena vždy aktuální verze
Nároky na správu aplikace	-	vysoké	0	střední/nízké	+	nízké/žádné
Nároky na podporu uživatelů aplikace	-	vysoké	0	střední/nízké	+	nízké/žádné
Přenositelnost aplikace	-	musí se instalovat různé verze pro různé platformy	0	požaduje runtime nebo pluginy do prohlížeče	+	běží v každém internetovém prohlížeči
Možnost práce uživatele offline	+	ano	0	ano – synchronizace	-	ne
Uživatelské rozhraní aplikace	+	přívětivé	0	přívětivé	-	strohé
Odezva aplikace	+	okamžitá	0	rychlá (JAVA, Flash, Silverlight, AJAX)	-	pomalá (reload stránky po každém kliknutí)
Stopy aplikace v systému	-	velký (vlastní program, DLL, registry)	0	střední (vlastní program a soubory v tempu)	+	malý (cookies, soubory v tempu, historie)
Způsob uložení dat	0	data jsou stahovaná na klienta	0	data se nachází dočasně na klientovi	0	data jsou uložena výhradně na serveru



DÁME PAUZU?

JAKÉ JSOU TEDY DŮVODY K ZAVEDENÍ NOVÉHO IS V ORGANIZACI?

ANALÝZA A TVORBA IS

CO JE TO ANALÝZA

- Z řečtiny: *ana-lysis*, tedy *rozebrání, rozpuštění, rozvázání*
- = Rozložení problému na menší celky, které jsou snadněji uchopitelné
- Analýza systému
  - Rozložení systému na jednotlivé části tak, aby byly snadno uchopitelné.
  - Nutno brát v potaz kontext mezi jednotlivými částmi, jelikož se jedná o komplexní problém a jednotlivé části spolu vždy nějak souvisejí
  - Systém je tedy vhodné analyzovat z různých úhlů pohledu
- Analýza systému vs. Analýza businessu
- Umožňuje mimo jiné odstraňovat nejednoznačnosti, snižovat entropii a redundanci v systému

ANALÝZA SYSTÉMU

- Organizace jsou zpravidla komplexními systémy - proto je nutné, aby analýzu vytvářelo více lidí (business analytik, IT, process owner, atp.)
- Výstup analýzy musí být univerzálně srozumitelný

ZÁKLADNÍ PROSTŘEDKY PRO BOJ SE SLOŽITOSTÍ

- Hierarchický rozklad problematiky
- Etapizace a iterace postupu řešení
- Modelování a srovnávání modelů
- Použití grafických vyjadřovacích prostředků

ANALÝZA - ABSTRAKCE VS. KONKRETIZACE

- **Abstrakce:** myšlenkový proces, vylučuje odlišnosti a zvláštnosti jednotlivých objektů či jevů a zdůrazňuje společné, obecné, podstatné vlastnosti sledované množiny objektů či jevů.
- 3 stupně ABSTRAKCE:
  - Kategorizace
  - Agregace
  - Generalizace

- **Konkretizace:** přístup, při němž postupně vyčleňujeme z obecného specifické vlastnosti sledovaných objektů či jevů

ANALÝZA - ABSTRAKCE VS. KONKRETIZACE

- KATEGORIZACE
  - nejnižší stupeň abstrakce, znamená seskupování prvků (jevů) do tříd (kategorií) podle kritérií, které si zvolíme k účelu sledování těchto prvků (jevů)
- AGREGACE
  - abstrakce, při níž považujeme prvek za část většího celku. Jde o účelové sdružení prvků (tzv. abstrakce typu „část-celek“). Při agregaci nejde o zobecnění společných vlastností těchto prvků.
- GENERALIZACE
  - abstrakce typu „specifický typ – obecný nadtyp“. Při generalizaci hledáme společné vlastnosti nadřízeného celku jakožto nositele specifikovaných společných vlastností (atributů).

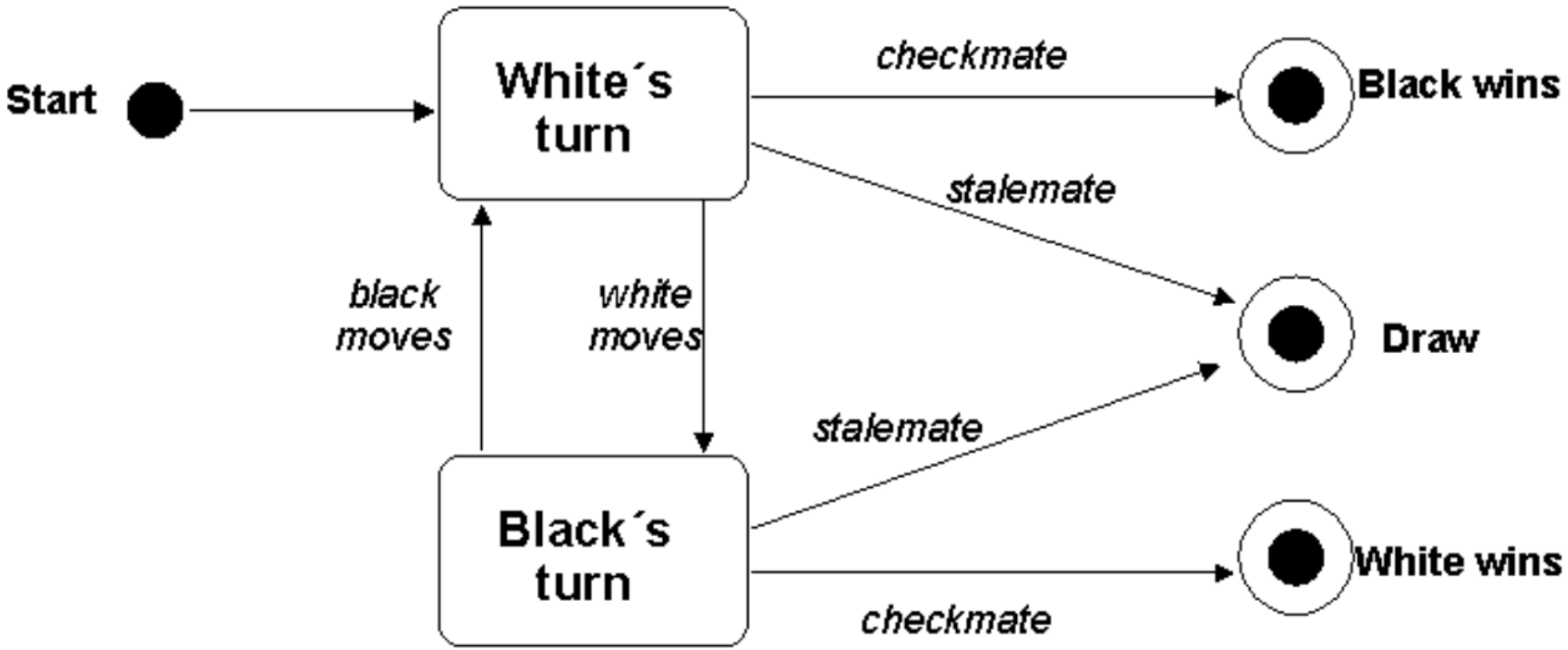
DRUHY PŘÍSTUPŮ K ANALÝZE A NÁVRHU IS

- V průběhu historického vývoje se vyprofilovaly dva základní přístupy k analýze a návrhu IS:
  - **STRUKTUROVANÝ** přístup (70. léta 20. stol.),
  - **OBJEKTOVĚ ORIENTOVANÝ** přístup (90. léta 20. stol.).

STRUKTUROVANÝ PŘÍSTUP

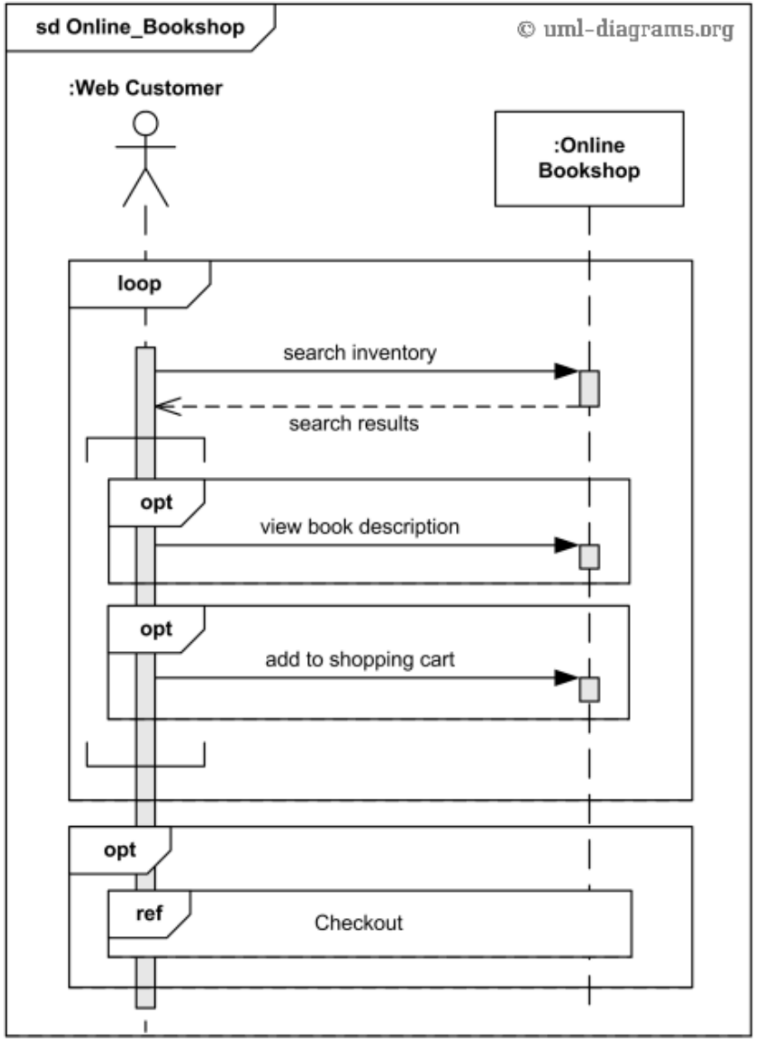
- Tok dat a transformace dat
- Jasná definice procesů a dat
- Nízká flexibilita, obtížné modelování komplexních systémů

Chess game


Start
White´s turn
checkmate
Black wins
stalemate
black moves
white moves
Draw
stalemate
Black´s turn
White wins
checkmate

OBJEKTOVÝ PŘÍSTUP

- Objekty a jejich interakce
- Vysoká flexibilita, snadné modelování komplexních systémů
- Složitější návrh, náročnější na pochopení


sd Online_Bookshop
© uml-diagrams.org
:Web Customer
:Online Bookshop
loop
search inventory
search results
opt
view book description
opt
add to shopping cart
opt
ref
Checkout

DĚKUJI ZA POZORNOST.