



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Úvod do IS

VIKBA18 Informační systémy

jan.matula@fpf.slu.cz

21. 2. 2014 & 28. 2. 2014

# Doporučená literatura

- SODOMKA, P. Informační systémy v podnikové praxi. 1. vyd. Praha: Computer Press, 2006. 351 s. ISBN 80-251-1200-4.
- DOUCEK, P. Řízení projektů informačních systémů. 2. vyd. Praha: Professional Publishing, 2006. 180 s. ISBN 80-86946-17-7.
- KAJZAR, D. – POLÁŠEK, I. Projektování informačních systémů. 1. vyd. Opava: Ediční středisko FPF SU v Opavě, 2003. 219 s. ISBN 80-7248-214-9.
- VLASÁK, R. – BULÍČKOVÁ, S. Základy projektování informačních systémů. 1. vyd. Praha: Karolinum, 2003. 144 s. ISBN 80-246-0727-1.
- KIMLIČKA, Š. Princípy informačných systémov. Bratislava: STU, 2006. 250 s.

# Ukončení předmětu

Úspěšné absolvování písemného testu (nad 70 %) – 2 varianty:

- Průběžné testy (celkem 3 v průběhu semestru, v termínech výuky kombinovaného studia) – ve všech 3 testech je třeba získat nad 70 %. Chybějící body (do 70 %) lze získat za aktivitu v hodinách (nelze je však použít u testu, ve kterém bylo získáno  $\geq 70$  %). 1 získaný bod na hodině reprezentuje 10 % v testu.
- Písemný test (shrnující za celý semestr) v zápočtovém týdnu či zkouškovém období.

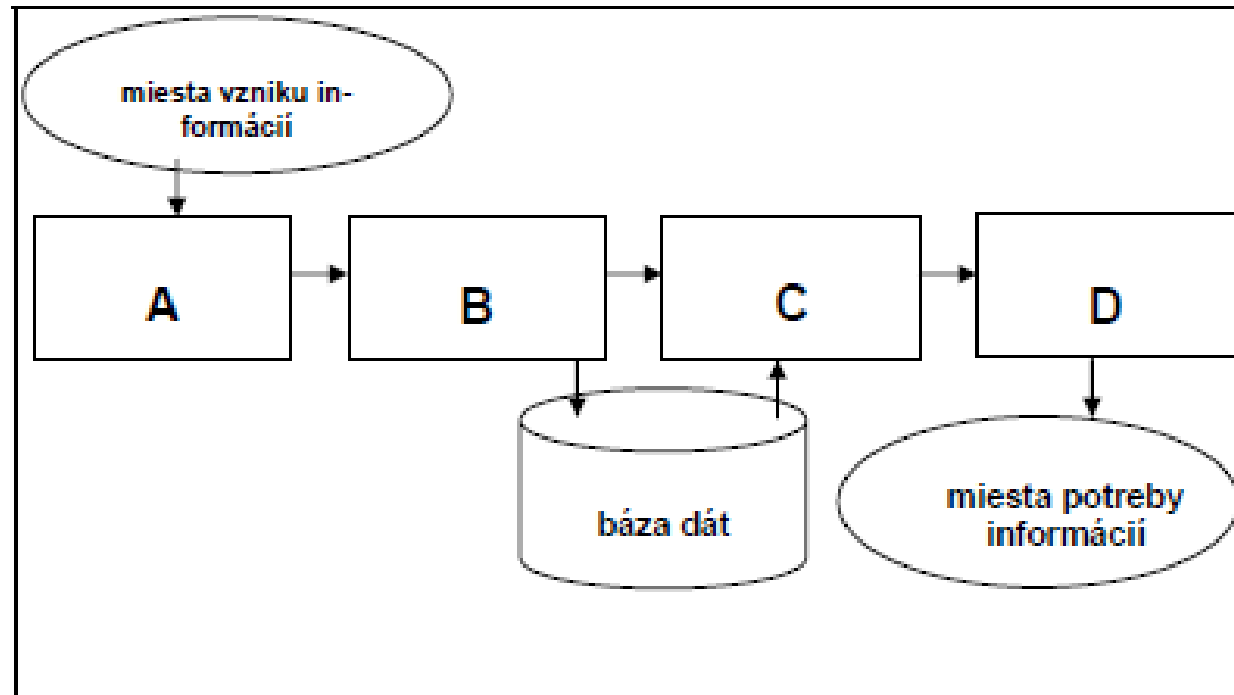
# Cíle předmětu

- Cílem předmětu je seznámit posluchače s teoretickými základy informačních systému, se základními principy fungování IS. Pozornost je také věnována analýze, projektování a zabezpečení IS. Na konci tohoto kurzu bude student schopen: porozumět a vysvětlit základní principy teorie systému; vytvořit projektový úkol a navrhnout IS.

# Obecná charakteristika IS

- systém umožňující komunikaci a transformaci informací - časově, prostorově i co do formy tak, aby byly lépe využity než v původním stavu (systém, který přidává hodnotu k zpracovávaným či komunikovaným informacím)
- speciální typ komunikačního média, jehož cílem je odstranit bariéry v přístupu k informacím
- účelové uspořádání vztahu a informačních toků mezi informačními zdroji, lidmi a technologickými prostředky spolu s procesy zpracování a komunikace informací
- model reálného světa, jehož základními prvky jsou informace

# Konceptuální model IS



Obr. 1.16 – Konceptuálny model informačného systému

# Jaké informace zpracovává IS?

- **strukturované data popisující neprostorové objekty** (záznamy v databázích, souborech a pod.) – dělíme na numerické a nenumerické
- **strukturované data popisující prostorové objekty** ve formě souřadnic (geografické informační systémy) – převážně numerické data,
- **nestrukturované data** (volné texty, záznamy rozhovorů a pod.),
- **metadata** (popis dat pomocí SGML jazyků – HTML, XML, struktury typu MARC, Dublin Core a pod.), které jsou často spojené s nestrukturovanými daty (plné texty dokumentů typu články, zpráva, kniha, ...) nebo obrázky, mapami, schémata, multimediálními dokumenty atd.

# Typické problémy řešené IS

- potřeba informací (pro poznání, pro rozhodování, pro realizaci určité činnosti)
- složitost (complexity)
- Znovu použitelnost (reusability)
- automatizace
- komunikace
- bezpečnost, spolehlivost, minimalizace rizik...



# Automatizovaný IS

- informační systém fungující s podporou informačních a komunikačních technologií
- automatizace procesu
- digitalizace datové základny

# Úvod do problematiky

- s rozvojem lidského poznání roste množství informací
- pro efektivní práci s informacemi začaly vznikat specializované IS
- IS = systémy pro sběr, uchování, vyhledávání a zpracování informací (dat, údajů) za účelem jejich poskytování

# Historie IS – agendové zpracování

- rozvoj spjat s vývojem ICT zejména PC
- na počátku – zpracování velkých informačních objemů na jednom PC  
= SYSTÉMY HROMADNÉHO ZPRACOVÁNÍ DAT nebo AGENDOVÉ ZPRACOVÁNÍ.
- Data se zaznamenávala do formulářů, následoval přepis na vhodné médium (děrné štítky, disketa), poté primární a sekundární zpracování, výsledkem jsou tištěné výstupní sestavy.

# Agendové zpracování dat

## ZÁVISLOST DAT A PROGRAMŮ

- Každý program řeší nejen vlastní aplikační problém, ale i formát **fyzického uložení** dat na médiu.
- **Navazující úlohy** musí **respektovat již vytvořené – deklarované fyzické struktury dat**.
- Při změně datové struktury v jednom programu je nutné měnit a kompilovat i všechny další programy, které s touto strukturou pracují, i když se v jejich funkčnosti nic nemění.
- Nízká efektivnost datových struktur i programu.

# Agendové zpracování dat

## PROBLÉMY AGENDOVÉHO ZPRACOVÁNÍ

- **Redundance:** některé informace ve více souborech opakují, jsou redundandní. Redundance je zdrojem mnoha dalších problémů
- **Konzistence:** vzájemná shoda údajů. Postupem času – vlivem nedostatečné kontroly v programech se stejné hodnoty na různých místech v datových souborech, začnou rozcházet.
- **Integrita:** data aktuální, odrážejí skutečnost z reálného světa. Problémem tedy je **zabezpečit, aby chybou či nedůsledností uživatele** nebyla porušena integrita a konzistence dat.
- **Obtížná dosažitelnost dat:** aplikační programy pro konkrétní požadavky; pro nový požadavek nutno napsat nový aplikační program - bez pomoci programátora nelze.
- **Izolovanost dat** - data roztroušena v různých souborech, soubory mohou být různě organizovány, data různě formátována. To komplikuje tvorbu nových aplikačních programů a možnost realizovat vazby mezi datovými strukturami.

# Agendové zpracování dat

## PROBLÉMY AGENDOVÉHO ZPRACOVÁNÍ

- **Současný přístup více uživatelů:** větší systémy vyžadují současný přístup k datům více uživatelů. Pak je nutné, aby programy vzájemně spolupracovaly, jejich činnosti byly koordinovány.
- **Ochrana proti zneužití:** při zpracování důvěrných či tajných dat není přípustné, aby měl kdokoliv přístup ke všem informacím. Při klasickém zpracování však musí mít programátor aplikačních programů k dispozici tolik podrobností, že to ochranu dat prakticky znemožňuje.

# Agendové zpracování dat

## POUČENÍ Z AGENDOVÉHO ZPRACOVÁNÍ

- Vedlo k návrhu a vytvoření programových systému (systému řízení báze dat – SŘBD) následujících vlastností:
- existuje **seznam datových typů**, které jsou v programovém systému definovány; pro tyto typy dat programový systém **vytváří fyzickou strukturu** na disku a automaticky řeší všechny přístupy k datům
- existují prostředky pro definování všech **sledovaných vlastností** popisovaných objektu;

# Agendové zpracování dat

## POUČENÍ Z AGENDOVÉHO ZPRACOVÁNÍ

- existuje **soubor instrukcí**, které nad definovanými daty provádějí jednotlivé operace; každá instrukce je vlastně mohutnou procedurou, v níž je řešen fyzický přístup k datum i realizace vlastní operace; jinak než prostřednictvím systému není možno s daty pracovat;
- programový systém řeší způsob, jak zaznamenat **vztahy mezi objekty**.



# Úvod do terminologie - pokračování

**Systém řízení báze dat (SŘBD)** lze chápat jako 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.

**Pro práci s daty SŘBD podporují zejména tyto funkce:**

- vytvoření báze dat (CREATE),
- vkládání dat (INSERT),
- aktualizace dat (UPDATE),
- rušení dat (DELETE),
- výběr z báze dat (SELECT).

Dále je podporována tvorba **formulářů** (vstupních obrazovek, Forms), **výstupních sestav** (Reports) a aplikačních programů.

# Úvod do terminologie

## Data

- údaje, které mají určitou vypovídací schopnost. Mohou být určitým způsobem uspořádány (seřazeny, např. podle velikosti, chronologicky atd.) a jsou uživateli k dispozici v různých formách (tabulky, grafy, zvukové signály, grafická forma atd.). Data jsou obvykle rozdělena na dílčí údaje (**atributy**) o dané množině objektů (entit), na základě nichž lze získat určitou informaci, která může vést k rozhodovacímu procesu.

# Úvod do terminologie

## Záznam

- je souhrn údajů (**atributů**) o dané části objektu, které jsou uloženy v **položkách (polích, Fields)** charakterizovaných názvem a datovým typem. Význam pojmu **záznam (Record)** a **položka** lze snadno ukázat na bázi dat STUDENT.

Abychom mohli u každého studenta zaznamenat potřebné údaje, musí mít záznam 5 položek, které mohou mít tyto názvy:

- identifikační číslo,
- jméno,
- příjmení,
- datum narození,
- bydliště,
- pohlaví.

# Úvod do terminologie

## Datové typy

každá položka musí být určitého datového typu. Obecně akceptované jsou tyto typy dat :

- **Textový typ** - textový řetězec, zpravidla do max. délky 255 znaků.
- **Číselné typy** - pro uložení celých a reálných čísel s pevnou i plovoucí desetinnou tečkou.
- **Logický typ** - slouží k uložení logické hodnoty Ano/Ne (True/False, Yes/No).
- **Memo** - pro uložení textu proměnné délky.
- **Datumový typ** - pro uložení datumových a časových hodnot.

# Úvod do terminologie

## Model

- je souhrn pravidel pro reprezentaci logické organizace dat v databázi.
- Základní typy modelů: hierarchický, síťový, relační, objektový.

**KONCEPTUÁLNÍ MODEL** je formalizovaný popis zájmové reality. Popisuje fakta o reálném světě, která jsou v čase neměnná nebo se mění pouze málo. Nejedná se o popis dat v počítači. Důvodem zavedení konceptuálního schématu je odstranit náhlý přechod od zájmové reality přímo k logickému schématu báze dat.

# Relační model – rozšíření

- byl popsán v roce 1970 Dr. Coddem. V současnosti je tento model **nejčastěji využíván u komerčních SŘBD**. Relační databázový model má jednoduchou strukturu. Data jsou organizována v tabulkách, které se skládají z řádků a sloupců. Všechny databázové operace jsou prováděny na těchto tabulkách.

# Relační model - rozšíření

Dr. Codd definoval jako minimalisticky relační ty systémy, které splňují tyto dvě vlastnosti:

1. Databáze je chápána uživatelem jako množina relací a nic jiného.
2. V relačním SŘBD jsou k dispozici minimálně operace selekce, projekce a spojení, aniž by se vyžadovaly explicitně předdefinované přístupové cesty pro realizaci těchto operací.

# 12 pravidel pro relační SŘBD

Dále definoval Dr. Codd dalších 12 pravidel pro relační SŘBD:

## **1) Informační pravidlo**

Všechny informace v relační databázi jsou vyjádřeny explicitně na logické úrovni jediným způsobem - hodnotami v tabulkách.

## **2) Pravidlo jistoty**

Všechna data v relační databázi jsou zaručeně přístupná kombinací jména tabulky s hodnotami primárního klíče a jménem sloupce.

## **3) Systematické zpracování nulových hodnot**

Nulové hodnoty jsou plně podporovány relačním SŘBD pro reprezentaci informace, která není definována a to nezávisle na datovém typu.



# 12 pravidel pro relační SŘBD

## 4) Dynamický on-line katalog založený na relačním modelu

Popis databáze je vyjádřen na logické úrovni stejným způsobem jako zákaznická data, takže autorizovaný uživatel může aplikovat stejný relační jazyk ke svému dotazu jako uživatel při práci s daty.

## 5) Obsáhlý datový podjazyk

Relační systém může podporovat několik jazyků a různých módů použitých při provozu terminálu. Nicméně musí být nejméně jeden příkazový jazyk s dobře definovanou syntaxí, který obsáhle podporuje definici dat, definici pohledů, manipulaci s daty jak interaktivně, tak programem, integritní omezení, autorizovaný přístup k databázi, transakční příkazy apod.

## 6) Pravidlo vytvoření pohledů

Všechny pohledy, které jsou teoreticky možné, jsou také systémem vytvořitelné.

# 12 pravidel pro relační SŘBD

## **7) Schopnost vkládání, vytvoření a mazání**

Schopnost zachování relačních pravidel u základních i odvozených relací je zachována nejen při pohledu na data, ale i při operacích průniku, přidání a mazání dat.

## **8) Fyzická datová nezávislost**

Aplikační programy jsou nezávislé na fyzické datové struktuře.

## **9) Logická datová nezávislost**

Aplikační programy jsou nezávislé na změnách v logické struktuře databázového souboru.

# 12 pravidel pro relační SŘBD

## **10) Integritní nezávislost**

Integritní omezení se musí dát definovat prostředky relační databáze nebo jejím jazykem a musí být schopna uložení v katalogu a nikoliv v aplikačním programu.

## **11) Nezávislost distribuce**

Relační SŘBD musí být schopny implementace na jiných počítačových architekturách.

## **12) Pravidlo přístupu do databáze**

Jestliže má relační systém jazyk nízké úrovně, pak tato úroveň nemůže být použita k vytváření integritních omezení a je nutno vyjádřit se v relačním jazyce vyšší úrovně.

# ARCHITEKTURA DATABÁZÍ

U databází rozlišujeme 3 základní architektury:

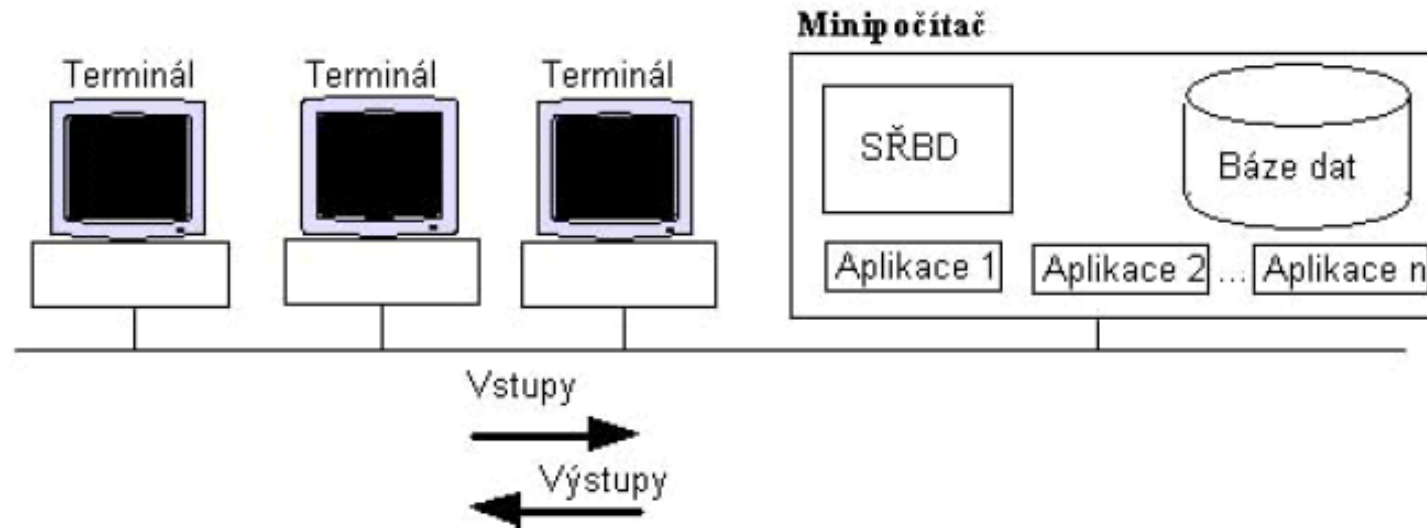
- **Centrální architektura**
- **Architektura file – server**
- **Architektura klient - server**

# Centrální architektura

- **data i SŘBD jsou uložena v centrálním počítači.**

Tato architektura je typická pro terminálovou síť, kdy se po síti přenáší vstupní údaje z terminálu na centrální počítač do příslušné aplikace, výstupy z této aplikace se přenáší na terminál. Protože aplikační program i vlastní zpracování probíhá na centrálním počítači, který může zpracovávat více úloh, mají odezvy na dotazy určité zpoždění.

# Obr. Centrální architektury



# Architektura file - server

- Tato metoda souvisí zejména s rozšířením osobních počítačů a sítí LAN. SŘBD a příslušné databázové aplikace jsou provozovány na jednotlivých počítačích, data jsou umístěna na file-serveru a mohou být sdílena. Aby nedocházelo ke kolizím při přístupu více uživatelů k jednomu datům, musí SŘBD používat vhodný systém zamykání (položek nebo celých tabulek).

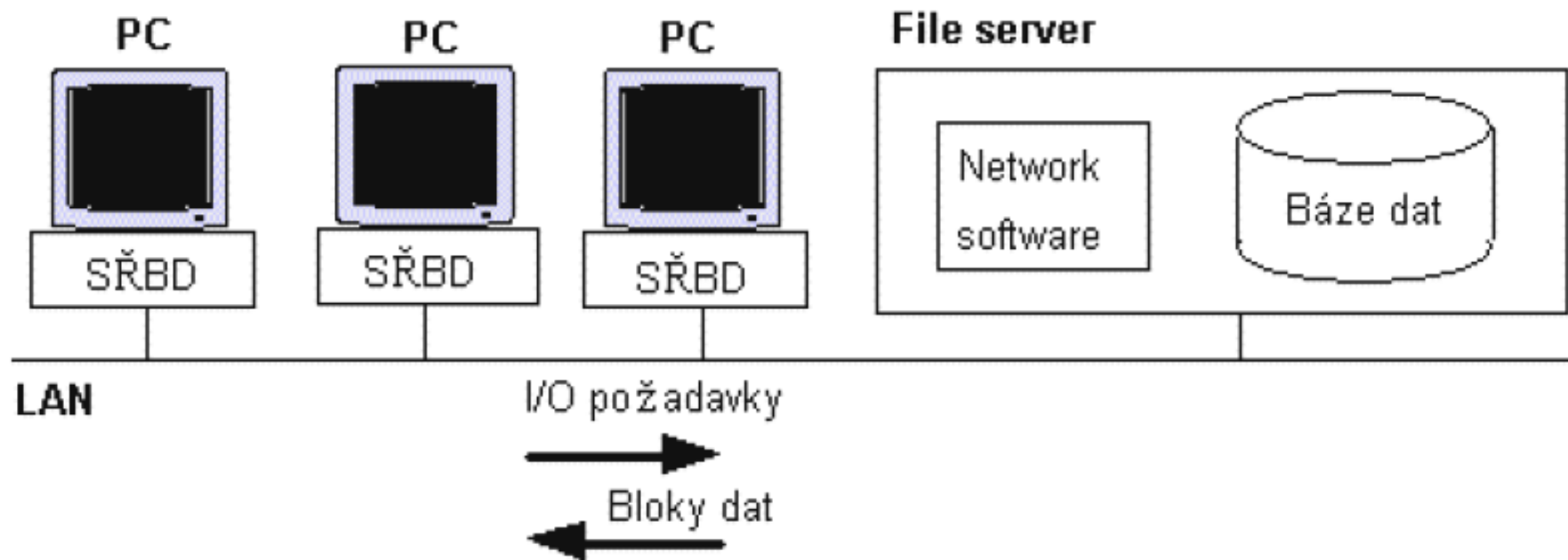
# Architektura file - server

Komunikace uživatele se systémem probíhá následujícím způsobem:

- uživatel zadá dotaz,
- SŘBD přijme dotaz, zasílá požadavky na data file-serveru,
- file-server posílá bloky dat na lokální počítač, kde jsou data zpracovávána podle zadaného dotazu (vyhledávání, setřídění atd.),
- výsledek dotazu se zobrazí na obrazovce osobního počítače.



# Obrázek architektury file - server



# Architektura klient - server

V podstatě je založena na lokální síti (LAN), personálních počítačích a databázovém serveru. Na personálních počítačích běží program podporující např. vstup dat, formulaci dotazu atd.

**Dotaz se dále předává pomocí jazyka SQL** (Structured Query Language) na databázový server, který jej vykoná a vrátí výsledky zpět na personální počítač. Databázový server je tedy nejvíce zatíženým prvkem systému a musí být tvořen dostatečně výkonným počítačem.

# Architektura klient - server

Komunikace probíhá tímto způsobem:

- uživatel zadává dotaz (buď přímo v SQL nebo musí být do tohoto jazyka přeložen),
- dotaz je odeslán na databázový server,
- databázový server vykoná dotaz,
- výsledek dotazu je poslán zpět na vysílací počítač, kde je zobrazen.

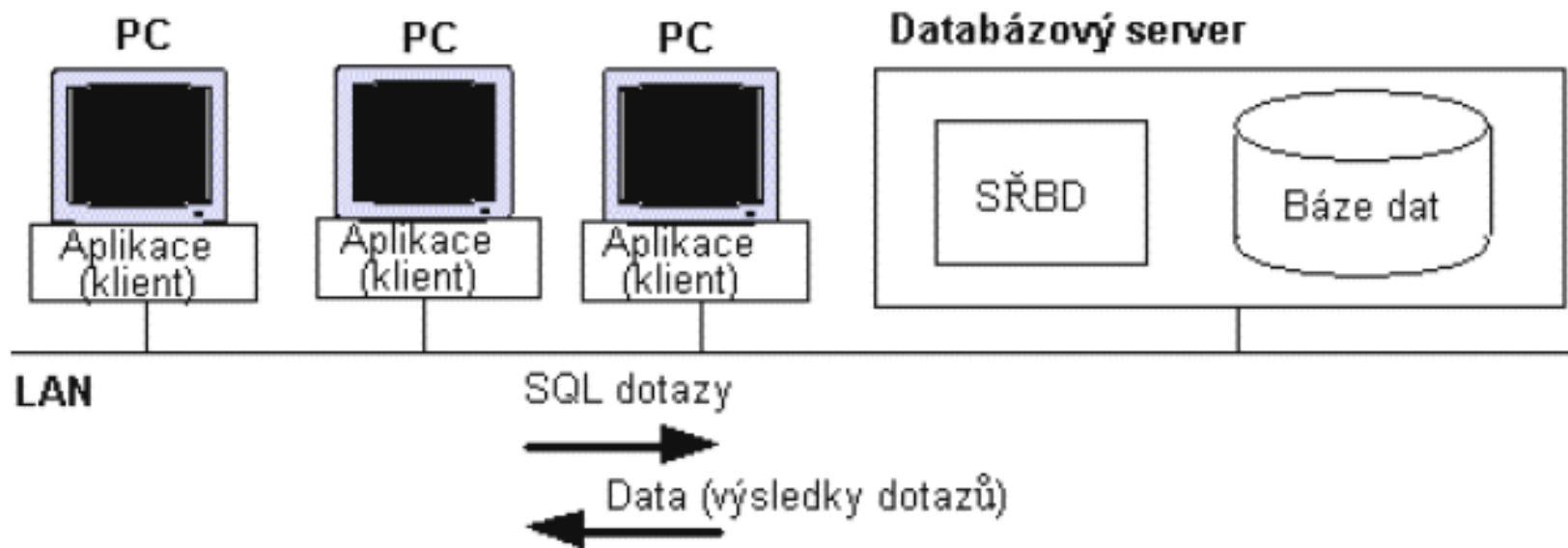
# Architektura klient - server

Architektura klient-server redukuje přenos dat po síti, protože dotazy jsou prováděny přímo na databázovém serveru a na personální 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, tam se teprve provede dotaz a zpracuje nalezených 100 záznamů.*

Architektura klient-server vyhovuje i náročným aplikacím a je využívána většinou renomovaných databázových firem.

# Obrázek architektury klient - server



# DISTRIBUOVANÁ DATABÁZE

Distribuovaná databáze je množina databází, která je uložena na několika počítačích. Uživatelé se však jeví jako jedna velká databáze.

**Distribuovanou databázi charakterizujeme 3 vlastnostmi:**

- Transparentnost
- Autonomnost
- Nezávislost na počítačové síti

# DD - TRANSPARENTOST

- Z pohledu klienta se zdá, že všechna data jsou zpracovávána na jednom serveru v lokální databázi. Uživatel používá syntakticky shodné příkazy pro lokální i vzdálená data, nespecifikuje místo uložení dat, o to se stará distribuovaný SŘBD.

# DD - AUTONOMNOST

- S každou lokální bází dat zapojenou do distribuované databáze je možno pracovat nezávisle na ostatních databázích.
- Lokální databáze je funkčně samostatná, propojení do jiné části distribuované databáze se v případě potřeby zřizují dynamicky.
- V distribuované databázi neexistuje žádný centrální uzel nebo proces odpovědný za vrcholové řízení funkcí celého systému, což výrazně zvyšuje odolnost systému proti výpadkům jeho částí.



# DD – NEZÁVISLOST NA POČ. SÍTI

Jsou podporovány různé typy architektur lokálních i globálních počítačových sítí (LAN, WAN). V jedné distribuované databázi tedy mohou být zapojeny počítače i počítačové sítě různých architektur, pro komunikaci se používá jazyk SQL.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ