

# Úvod do IS

Teoretické základy projektování IS

[jan.matula@fpf.slu.cz](mailto:jan.matula@fpf.slu.cz)

I. blok

# 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 % (včetně). 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.
- Písemné testy (chybějící nebo případně všechny 3 najednou) v zápočtovém týdnu či zkuškovém období.

# 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.

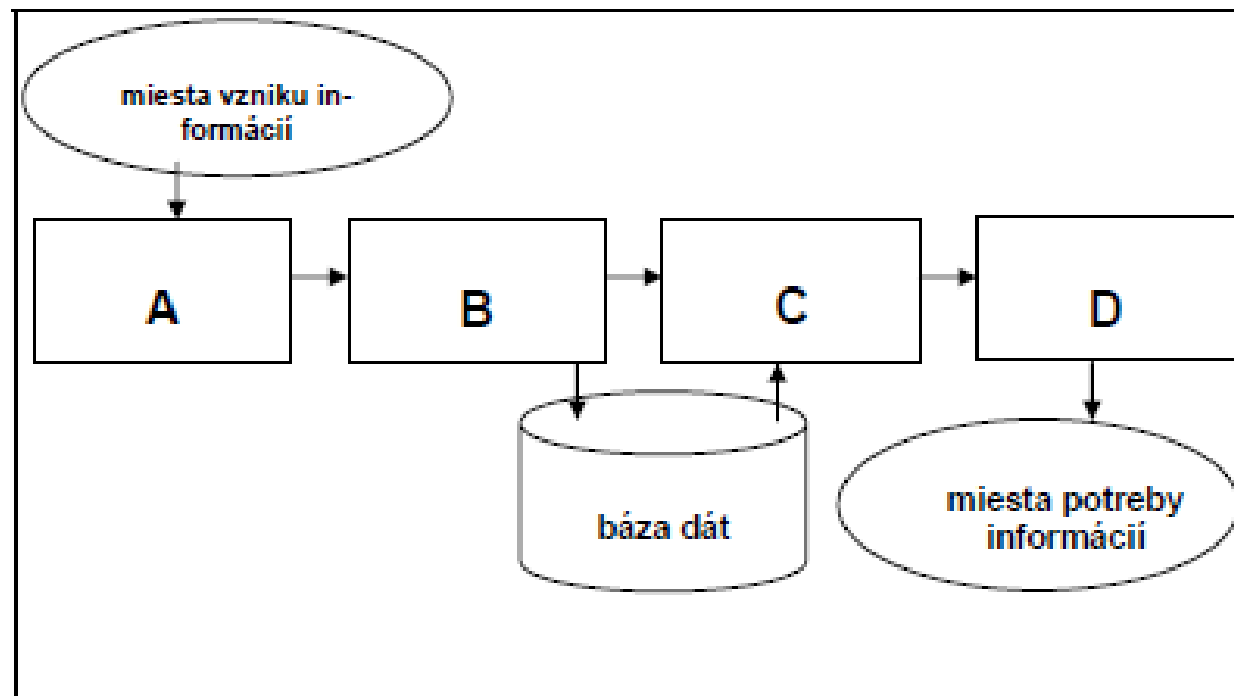
# Cíle předmětu

- 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:

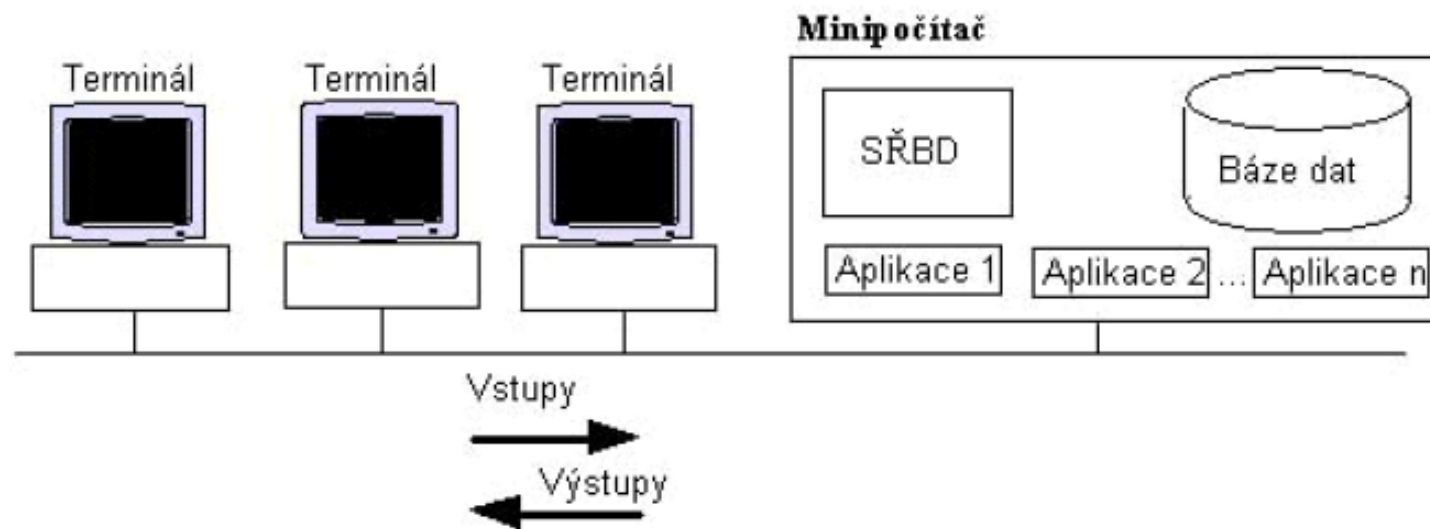
- **Centrální architektura**
- **Architektura file – server**
- **Architektura klient - server**

# 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).

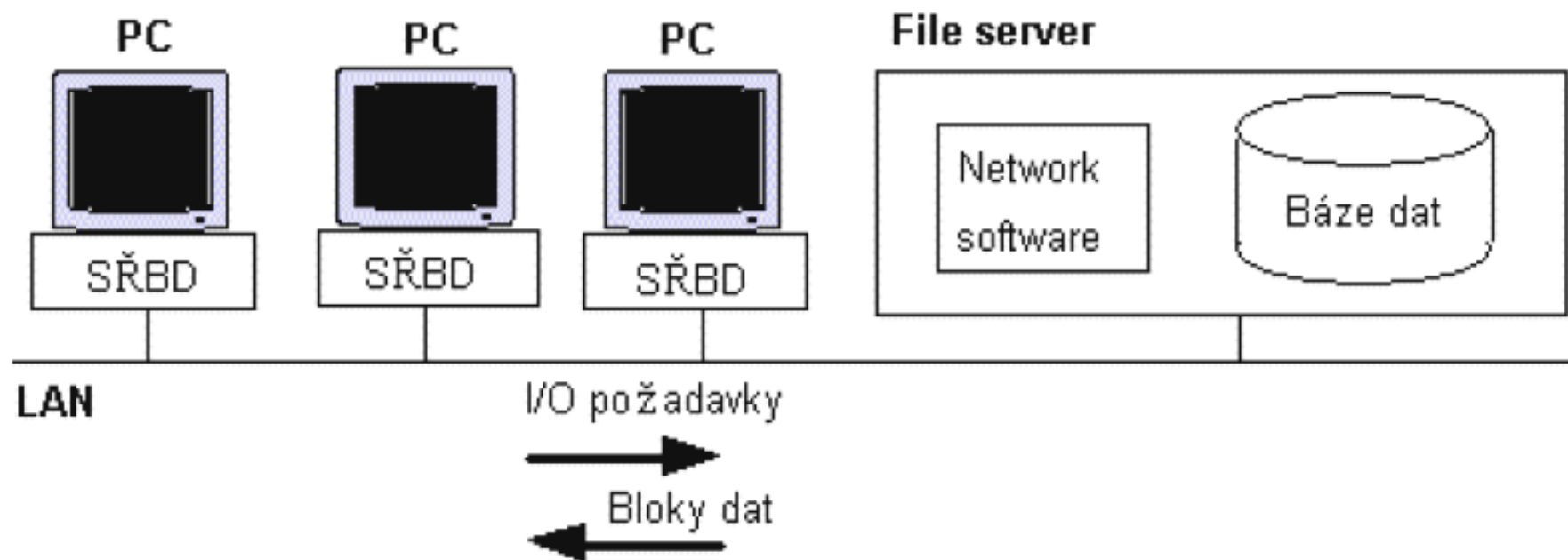
# Architektura file - server

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.

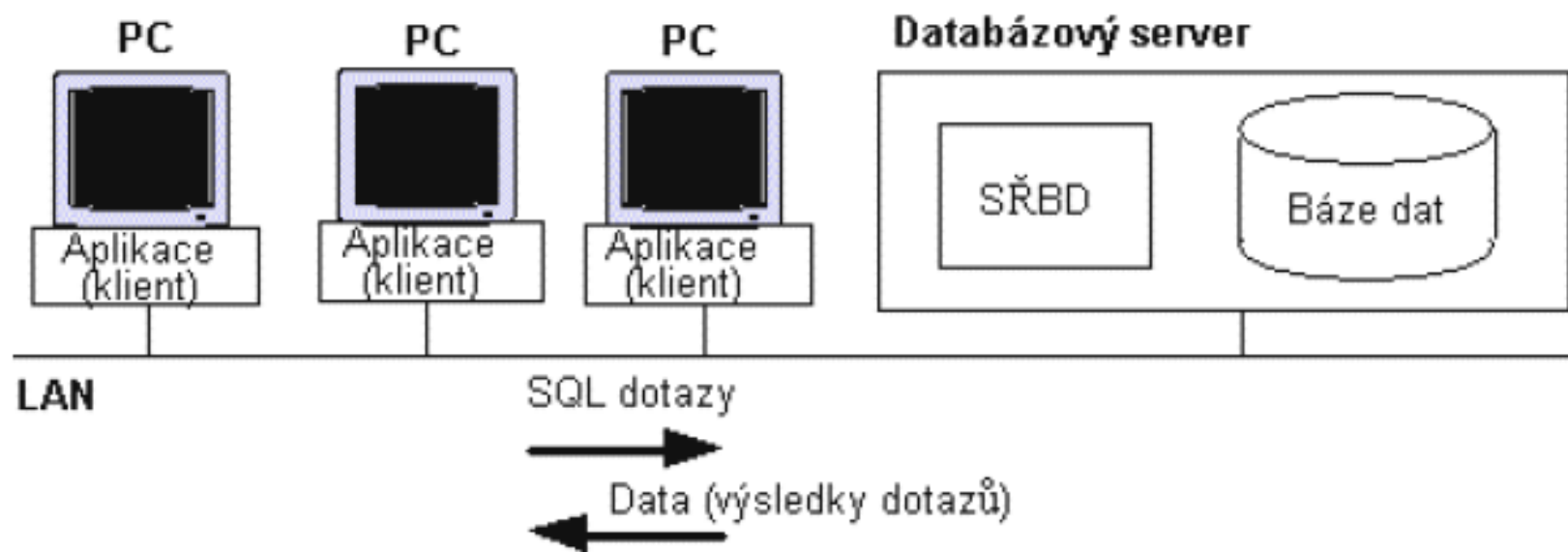
# Architektura klient - server

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.

# Základní pojmy DTB zpracování

## ENTITA - OBJEKT

celá posloupnost položek popisuje objekt. Taková struktura položek, která má ucelený význam (zachycuje všechny potřebné údaje o sledovaném objektu) se nazývá **záznamem** (větou, recordem). Je to obvykle skupinová položka.

## MNOŽINA ENTIT – MNOŽINA OBJEKTŮ – DATOVÝ SOUBOR – OBSAH TABULKY

množinu záznamů stejného typu, zaznamenávající ucelenou informaci o množině sledovaných objektů a uloženou na paměťovém médiu, nazýváme **datovým souborem**. Množiny záznamů si můžeme snadno představit ve tvaru **tabulky**, kde každý objekt je popsán jedním řádkem a každý atribut objektu je v jednom sloupci.

## DATABÁZE

Množinu datových souborů, uchovávajících data o nějakém uceleném úseku reality, nazýváme **databází**.

# Základní pojmy DTB zpracování

## **SYSTEM ŘÍZENÍ BÁZE DAT – SŘBD**

programový systém (prázdný, bez datových souborů a bez aplikačních programů), umožňující definování datových struktur a datových souborů, řešící fyzické uložení dat ve vnější paměti počítače, umožňující manipulaci s daty a formátování vstupních i výstupních informací, nazýváme **systemem řízení báze dat**.

## **APLIKAČNÍ ÚLOHA**

**Aplikační úlohou** nad SŘBD nazýváme konkrétní program napsaný pomocí programových prostředků použitého SŘBD nad konkrétní databází, pro tuto úlohu vytvořenou.

# Základní pojmy DTB zpracování

## INFORMAČNÍ SYSTÉM

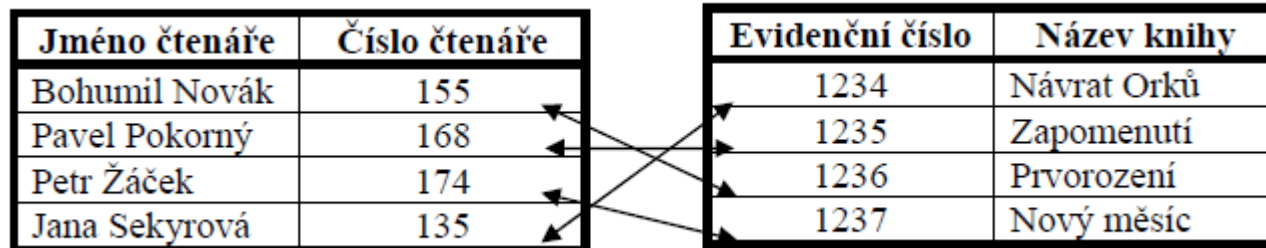
Aplikační úlohy nad společnou databází tvoří ucelený systém, nazývaný **databázovým nebo informačním systémem** (dále jen IS) nad použitým SŘBD.

V tomto pojetí tedy IS rozumíme celek, řešící rozsáhlejší oblast aplikační, naprogramovaný v jednom SŘBD s vhodně navrženými datovými strukturami tak, aby všechny aplikační úlohy k nim měly optimální přístup. Řeší **uložení, uchování, zpracování a vyhledávání informací a umožňuje jejich formátování** do uživatelsky přívětivého tvaru.

# Vztahy mezi entitami (relace)

1:1

V relaci 1:1 odpovídá jednomu záznamu v první tabulce maximálně jeden záznam v druhé tabulce a naopak jednomu záznamu v druhé tabulce maximálně jeden záznam v první tabulce.



# Vztahy mezi entitami (relace)

## 1:N

V relaci 1:N odpovídá jednomu záznamu v první tabulce žádný, jeden nebo více záznamů v druhé tabulce a naopak jednomu záznamu v druhé tabulce maximálně jeden záznam v první tabulce.

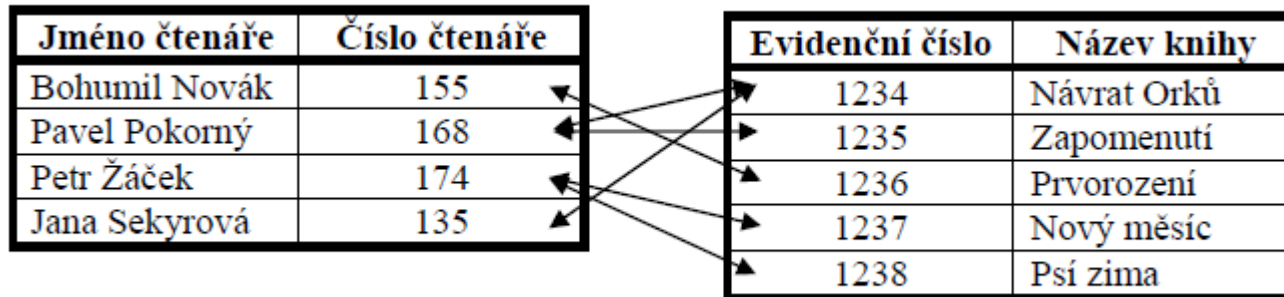
# Vztahy mezi entitami (relace)

## N:N

- V relaci N:N odpovídá jednomu záznamu v první tabulce žádný, jeden nebo více záznamů v druhé tabulce a naopak jednomu záznamu v druhé tabulce žádný, jeden nebo více záznamů v první tabulce.
- Chcete-li vyjádřit relaci typu N:N, musíte vytvořit třetí tabulku, která se často nazývá spojená tabulka, jež rozdělí relaci typu N:N na dvě relace typu 1:N. Primární klíč z těchto dvou tabulek vložíte do třetí tabulky. Výsledkem je, že třetí tabulka zaznamená každý výskyt nebo instanci relace

# Vztahy mezi entitami (relace)

N:N





# Úrovně DBS

## Způsoby pohledu na data v databázi

- FYZICKÝ (INTERNÍ) – způsob fyzického uložení dat na disk
- KONCEPTUÁLNÍ – popis struktury databáze (tabulek), popis vztahů mezi uloženými daty
- EXTERNÍ – popis dat z pohledu uživatele, tj. uživatelské prostředí (formuláře, sestavy,...)

# Datové modely

Způsob uložení dat v databázi

- Hierarchický
- Síťový
- Relační
- Objektový

# Hierarchický DM

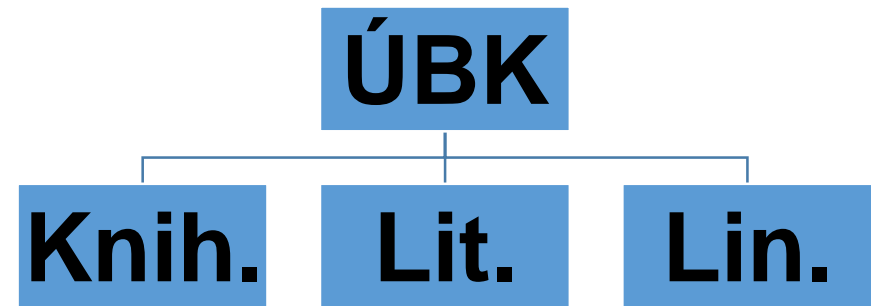
Záznamy jsou organizovány ve stromové struktuře

VÝHODY:

- Řeší snadno a rychle vztahy 1:N
- Nezáleží na fyzické struktuře dat

NEVÝHODY:

- Problémy při řešení vztahů M:N
- Problémy při změně struktury dat



# Síťový DM

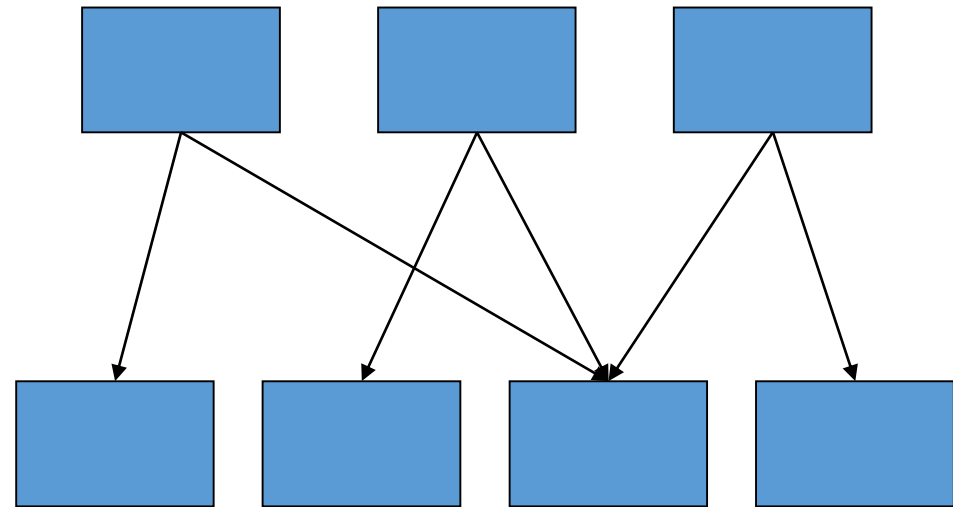
Data jsou reprezentována kolekcemi záznamů a vztahů mezi nimi.

## VÝHODY:

- Řeší snadno a rychle vztahy 1:N i M:N
- Nezáleží na fyzické struktuře dat
- Rychlé vyhledávání

## NEVÝHODY:

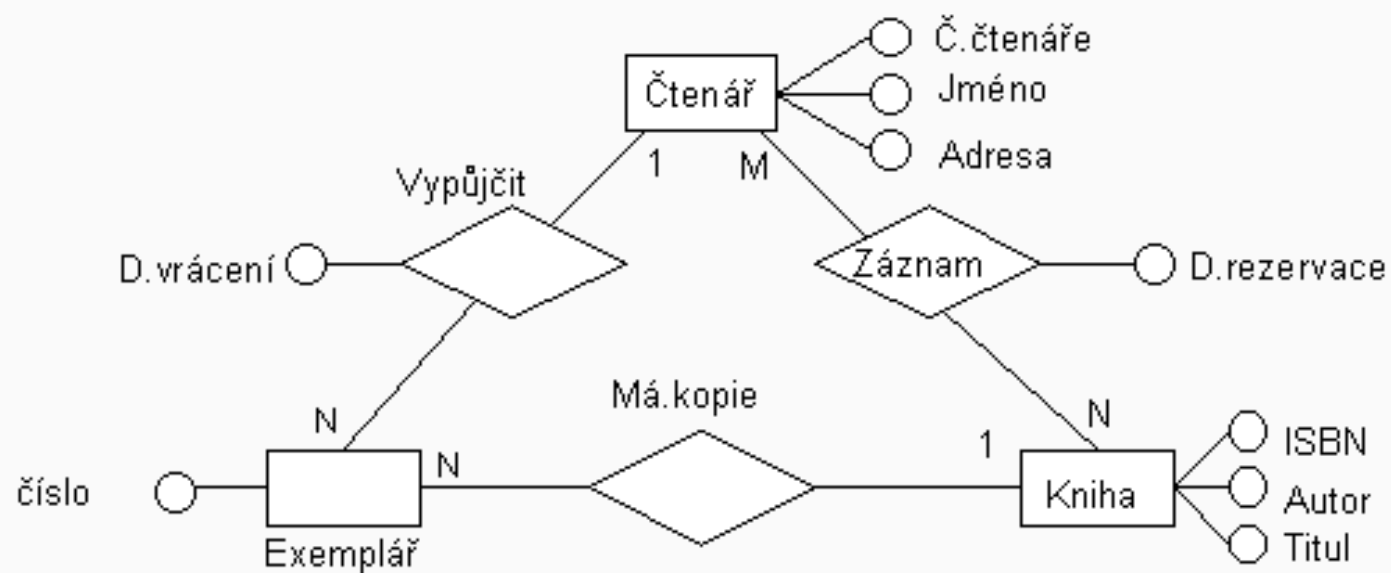
- Problémy při změně struktury dat



# Relační DM

- **RDBS** – relační databázový systém
  - Informace uchovávané v jednom typu objektu jsou uchovávané v tabulkách s určitou strukturou
  - Tabulky jsou navzájem provázány relacemi
  - Relace usnadňují vyhledávání různých informací uložených v těchto tabulkách
- **V SOUČASNOSTI NEJPOUŽÍVANĚJŠÍ MODEL**

# Schéma entitně-relačního modelu



# Relační DM

Číslo objednávky	Název výrobku
10456	003
10456	004
10457	101
10457	002
10457	001
10457	102
10458	006
10459	001
10459	004

Id zaměstnance	Jména zaměstnanců	Zaměstnán od
01	Jan Novák	1.2.2005
02	Emil Král	15.2.2001
03	Václav Nový	24.8.2002

Id výrobku	Název výrobku	Cena výrobku
001	Fausto	155
002	Funghi	125
003	Carpaccio	135
004	Hawai	140
005	Rustica	139
006	Trapolla	155
101	Instalate Beluco	125
102	Instalate Caesar	128

Číslo objednávky	Datum objednávky	Id zaměstnance
10456	12.8.2005	01
10457	12.8.2005	02
10458	13.8.2005	02
10459	13.8.2005	03

# Relační tabulka

**atributy**

<b>Id výrobku</b>	<b>Název výrobku</b>	<b>Cena výrobku</b>
001	Fausto	155
002	Funghi	125
003	Carpaccio	135
004	Hawai	140
005	Rustica	139
006	Trapolla	155
101	Instalate Beluco	125
102	Instalate Caesar	128

**záznamy**




# Objektový DM

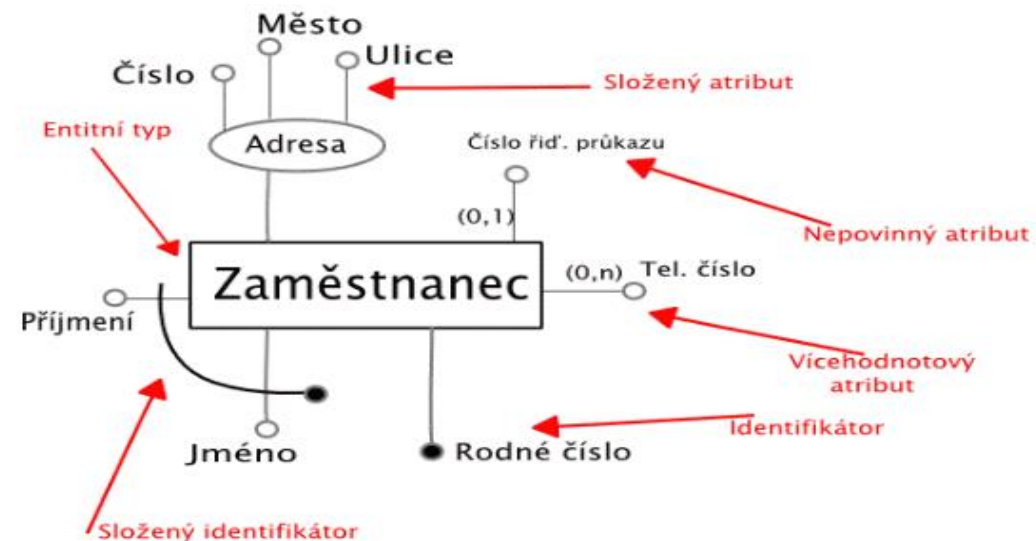
- Vychází z principů objektově orientovaného přístupu
- Objekt – datová struktura definovaná jako třída s určitými vlastnostmi a metodami
- Komunikace mezi objekty probíhá pomocí zpráv
- Výhody
  - Nejen statické, ale i dynamické chování objektů
  - Možné vytváření složitějších objektů
  - Snadnější zadávání dotazů

# Konceptuální model


- Slouží k popisu dat v databázi nezávisle na jejich fyzickém uložení
- Umožňuje zobrazit a popsat objekty v databázi a vztahy mezi nimi z hlediska jejich významu a chování
- Výsledkem je implementačně nezávislé schéma obecně aplikovatelné v jakémkoli prostředí
- Znázorňuje se v podobě **ER diagramu**, který definuje entity (třídy prvků), jejich atributy a relace (vztahy) mezi nimi

# Pojmy v ER-modelu

- ENTITNÍ TYP (v diagramu ) reprezentuje třídu entit (např. Zaměstnanec).
- Každý ENTITNÍ TYP má nějaké atributy (např. Jméno), z nichž některé mohou být identifikátory (jednoznačně určují instanci entity).
- Pokud ET nemá žádné identifikátory explicitně označené, jsou jimi všechny atributy dohromady (tzv. složený indikátor). Identifikátory mohou být více atributové.



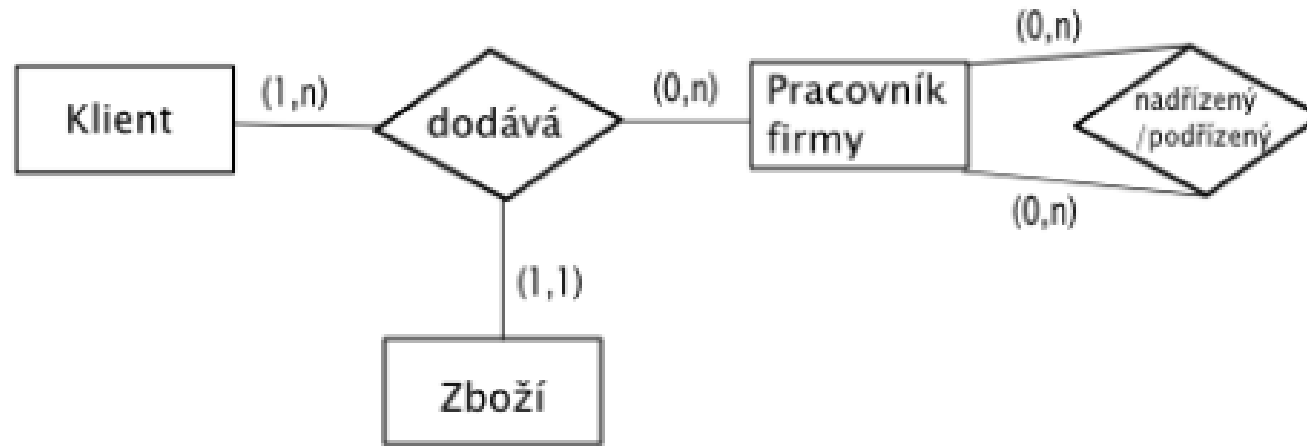
# Vztahový typ

- VT (v diagramu ) popisuje vztahy mezi jednotlivými entitami – s těmi entitami, se kterými je v nějakém vztahu, je spojen čarou.
- Vztah může mít danou i KARDINALITU (kolik entit z každé strany do vztahu vstupuje), která může být typu 1:1, 1:n, n:n a je značena vedle čáry spojující vztahový typ s entitou.
- Entity ve vztahu mohou mít navíc povinné či nepovinné členství (vstupovat do něj vždy či někdy)

# Vztahový typ

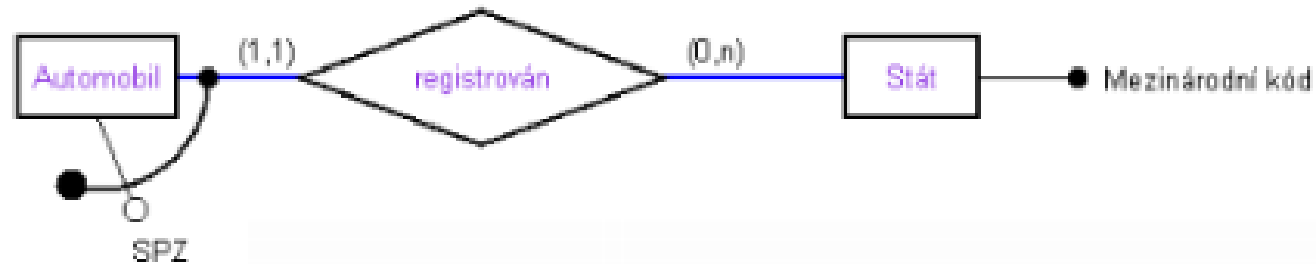
- Vztahy mohou být buď binární nebo **n-ární**, ale více než ternární vztahy se většinou neobjevují. Vztahy mohou být i **REKURZIVNÍ**, tj. do vztahů vstupují entity stejného typu.
- Instance vztahového typu je jednoznačně určena identifikátory instancí ve vztahu.
- Některé entitní typy mohou být spolu identifikovány (nebo přímo identifikovány) vztahem – pak se nazývají slabé entitní typy.

# Vztahové typy



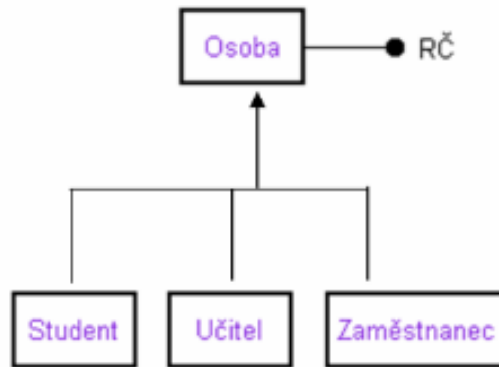
Obr. Ukazuje ternární vztah s různými kardinalitami – klientovi někdo dodává zboží jednou až n-krát, pracovník dodává nula až n-krát zboží (tj. jde o nepovinné členství ve vztahu, mohou existovat pracovníci, kteří nic nedodávají) a zboží je vždy dodáváno právě jednou. Na zaměstnancích je zároveň ukázán rekurzivní binární vztah.

# Slabý entitní typ



Obr. Ukazuje slabý entitní typ – automobil je identifikován SPZ a zároveň státem, ve kterém je registrován.

# ISA hierarchie



ISA hierarchie je rozšíření ER diagramů o dědičnost entit - tj. rozdělení entitních typů na subtypy (a přidání dalších vztahů nebo atributů pro subtypy). V ISA hierarchii se povoluje pouze jednonásobná dědičnost, navíc potomci nějakého entitního typu musí být jednoznačně identifikováni předkem (tj. všechny entity v hierarchii sdílí identifikátor).



# Základní pojmy

- **Klíč (key)**
  - Jeden nebo několik atributů tabulky určený pro setřídění záznamů podle hodnot v tomto poli (numerický, textový)
- **Unikátní klíč**
  - Klíč tabulky, ve kterém se každá hodnota atributu vyskytuje nejvýše jedenkrát
- **Duplicitní klíč**
  - Klíč tabulky, ve kterém se každá hodnota atributu může vyskytovat vícekrát (u více různých záznamů)

# Základní pojmy

- **Jednoduchý klíč**
  - Klíč tabulky, který je tvořen pouze jediným atributem
  - Nejčastěji se vyskytující typ klíče
  - I uměle vytvořený (id)
- **Složený klíč**
  - Klíč tabulky, který je tvořen alespoň dvěma atributy
  - Často součást tzv. spojovací tabulky

# Základní pojmy

- **Primární klíč (primary key)**

- Klíč tabulky, který slouží k jednoznačné identifikaci záznamu
- Musí být unikátní
- Označuje se PK
- V jedné tabulce může být nanejvýše jeden primární klíč

- **Cizí klíč (foreign key)**

- Klíč tabulky, který slouží k propojení (vytvoření relace) s primárním klíčem jiné tabulky
- Často obsahuje duplicitní hodnoty
- Označuje se FK
- V jedné tabulce může být i více cizích klíčů

# ER model (postup vytváření)

## 1. Určení typu entit

- zvolení množiny objektů stejného typu
- např. Objednávka, Zaměstnanec, Výrobek

## 2. Určení typů relací

- vztahů, do kterých mohou příslušné entity vstupovat
- např. objednávka obsahuje výrobek

## 3. Určení atributů

- přiřazení jednotlivým entitám a vztahům
- např. Objednávka (číslo, datum, ...)

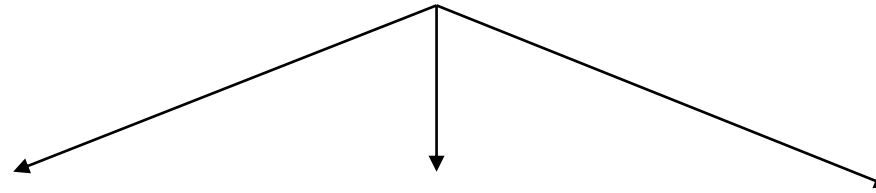
## 4. Určení integritních omezení

- zpřesnění navrženého modelu
- např. atribut datum je datového typu Datum a čas

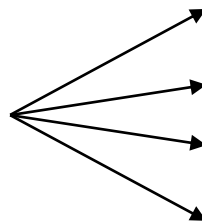
# ER model

Relační tabulka

atributy



záznamy



Id výrobku	Název výrobku	Cena výrobku
001	Fausto	155
002	Funghi	125
003	Carpaccio	135
004	Hawai	140
005	Rustica	139
006	Trapolla	155
101	Instalate Beluco	125
102	Instalate Caesar	128

**FK** **FK**

Číslo objednávky	Název výrobku
10456	003
10456	004
10457	101
10457	002
10457	001
10457	102
10458	006
10459	001
10459	004

**PK**

Id zaměstnance	Jména zaměstnanců	Zaměstnan od
01	Jan Novák	1.2.2005
02	Emil Král	15.2.2001
03	Václav Nový	24.8.2002

**PK** **FK**

Číslo objednávky	Datum objednávky	Id zaměstnance
10456	12.8.2005	01
10457	12.8.2005	02
10458	13.8.2005	02
10459	13.8.2005	03

**PK**

Id výrobku	Název výrobku	Cena výrobku
001	Fausto	155
002	Funghi	125
003	Carpaccio	135
004	Hawai	140
005	Rustica	139
006	Trapolla	155
101	Instalate Beluco	125
102	Instalate Caesar	128

# QBE (Query By Example)

- QBE = dotaz podle vzoru, je rozhraní pro zadávání dotazů grafickou formou. Znalost konkrétního dotazovacího jazyka není vyžadováno.
- Lze provádět základní relační operace (selekce, projekce, spojení), ale také další operace jako třídění, součty, apod. (v závislosti na konkrétním SŘBD).

Č.Studenta	Příjmení	Jméno	Adresa	Věk
	√ No..	√	√	√ >18, <=25

# Transakce

- TRANSAKCE = posloupnost operací nad objekty BÁZE DAT, které realizují jednu nebo více ucelených operací z hlediska uživatele.
- Transakce začíná vykonáním prvního příkazu nebo speciálním příkazem <např. BeginTrans>.
- Transakce může skončit úspěšně nebo neúspěšně.
- Neúspěch může vzniknout: poruchou hardware, chybou programového vybavení, chybnými daty, apod.



# Transakce

- Bod, od kterého lze považovat transakci za úspěšnou se nazývá **bodem potvrzení**.
- Informace o změnách během transakce se ukládají do tzv. **žurnálového (transakčního) souboru**, teprve po dosažení bodu potvrzení se promítnou do báze dat.
- Při vlastní transakci se data v bázi dat nemění, tzn. při chybě nebo poruše nedojde k porušení konzistentního stavu báze dat.

# SQL (Structured Query Language)

- SQL = strukturovaný dotazovací jazyk.
- Standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích.
- Vznik v 70. letech 20. století (IBM – výzkum relačních databází) jako sada příkazů pro ovládání RD – vznik jazyka SEQUEL (Structured English Query Language).
- Cílem bylo vytvořit jazyk, ve kterém se příkazy tvoří syntakticky a v návaznosti na přirozený jazyk (angličtina).

# SQL

- Jazyk zahrnuje nástroje pro tvorbu databází (tabulek) a dále nástroje pro manipulaci s daty (vkládání dat, mazání, vyhledávání, atd.).
- SQL patří mezi tzv. DEKLARATIVNÍ PROGRAMOVACÍ JAZYKY, což znamená, že kód jazyka SQL nepíšeme v samostatném programu, ale vkládáme jej do jiného programovacího jazyka, který je již procedurální.
- Se samotným SQL můžeme pracovat pouze v případě terminálového přístupu na SQL server (skrze příkazový řádek).

# Příkazy jazyka SQL

SQL příkazy se dělí do 4 zákl. skupin:

- Příkazy pro MANIPULACI S DATY

(např.: SELECT, INSERT, UPDATE,...)

- Příkazy pro DEFINICI DAT

(např.: CREATE, DROP,...)

- Příkazy pro ŘÍZENÍ PŘÍSTUPOVÝCH PRÁV

(např.: GRANT, REVOKE,...)

- Příkazy pro ŘÍZENÍ TRANSAKČÍ

(např.: START TRANSACTION, COMMIT,...)

# SQL

SQL se skládá z několika částí:

- **DDL – Data Definition Language** – jazyk pro vytváření databázových schémat a katalogů,
- **SDL – Storage Definition Language** – jazyk pro definici způsobu ukládání tabulek,
- **VDL – View Definition Language** – jazyk pro návrháře a správce, určuje vytváření pohledů na tabulky,
- **DML – Data Manipulation Language** – jazyk obsahující základní příkazy INSERT, UPDATE, DELETE, SELECT.

S jazykem DML pracují nejvíce koncoví uživatelé a programátoři databázových aplikací.

# SQL - uplatnění

SQL plní v současnosti celou řadu úloh, např.:

- Interaktivní dotazovací jazyk - uživatel zapíše příkazy SQL do interaktivního SQL programu, který najde potřebná data a zobrazí je na obrazovce.
- Databázový programovací jazyk - k použití při tvorbě databázové aplikace.
- Jazyk pro správu databází - správci databází využívají SQL pro definici datových struktur a kontrolu uložených dat.

# SQL - uplatnění

- Klient/server jazyk - programy pro počítače používají SQL pro komunikaci v lokální počítačové síti (LAN) s databázovými stanicemi (database servers), kde jsou uložena sdílená data.
- Jazyk distribuované báze dat - systémy spravující distribuované databáze, používají SQL jako doplněk pro distribuci dat mezi několika spojenými počítačovými systémy.
- Komunikační jazyk SŘBD v LAN - v síti s několika různými SRBD je SQL jedinou cestou pro jejich vzájemnou komunikaci.

# Metodiky, metody, techniky, nástroje

Metodika = souhrn etap, přístupů zásad.

Metodika stanovuje – co, kdo, kdy a proč má dělat během procesu vývoje.

Zahrnuje:

- organizace práce vývojového týmu
- metody práce s informacemi o vyvíjeném IS
- ekonomické otázky
- vedení projektové a provozní dokumentace
- způsob řízení v jednotlivých fázích vývoje IS
- SW a HW prvky doporučené pro vývoj IS



# Metodiky, metody, techniky, nástroje

- Metoda – určuje, co je třeba dělat v určité etapě vývoje IS. Bývá spojená s určitým přístupem (strukturovaný, objektový).
- Technika – určuje, jak se dobrat požadovaného výsledku, tj. určuje přesný postup kroků, způsob použití nástrojů apod.  
příklad technik: prototypování, normalizace datového modelu, transformační a transakční analýza při tvorbě struktury programového systému.

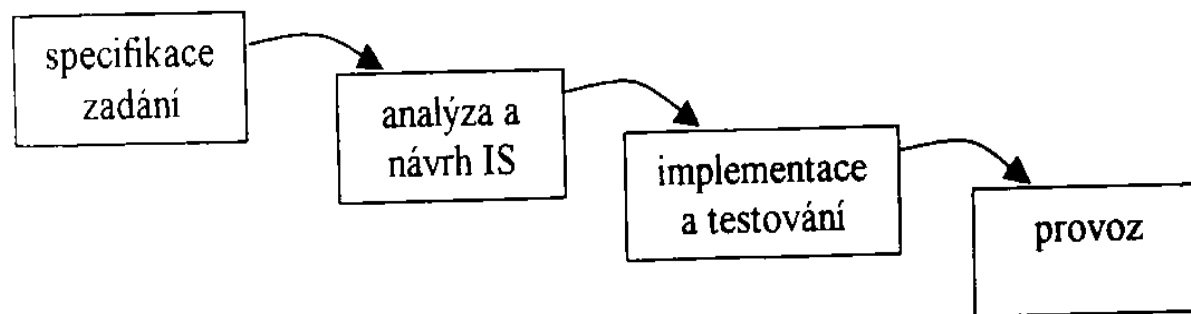
# Metodiky, metody, techniky, nástroje

- Nástroj = prostředek k uskutečnění určité činnosti, resp. k vyjádření výsledku dané činnosti (formalizuje vyjádření výsledku). Může být svázán s konkrétní technikou, např. CASE nástroje, modely IS (datový, funkční, stavový diagram).

# Vazby metodika-metoda-technika-nástroj

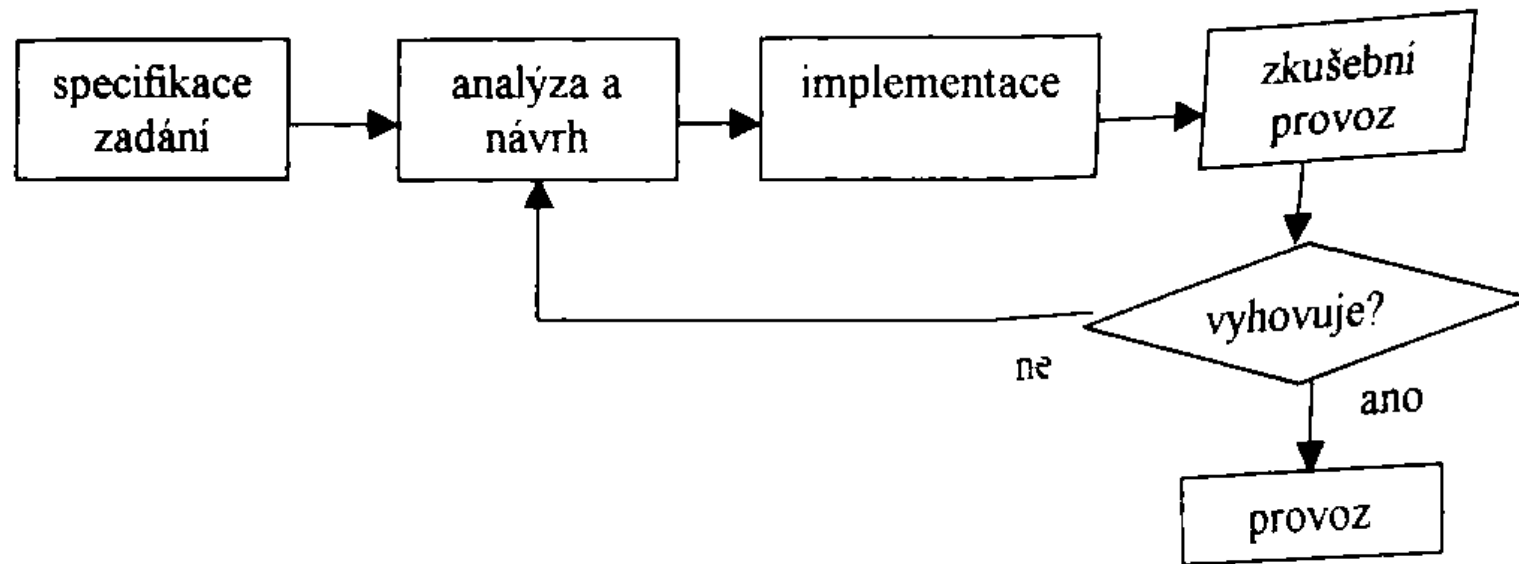
- Metodika doporučuje použití určitých metod v průběhu vývoje IS, metody pak využívají určitých technik a nástrojů. Není však možné prohlásit, že daná metoda patří jednoznačně k určité metodice. Některé metody jsou specificky využívány konkrétní metodikou. Většina metod je univerzálních, využívají různé metodiky, v různých etapách vývoje IS.
- Metodologie vývoje IS = zobecňující nauka o metodikách a metodách vývoje IS.

# Životní cyklus IS



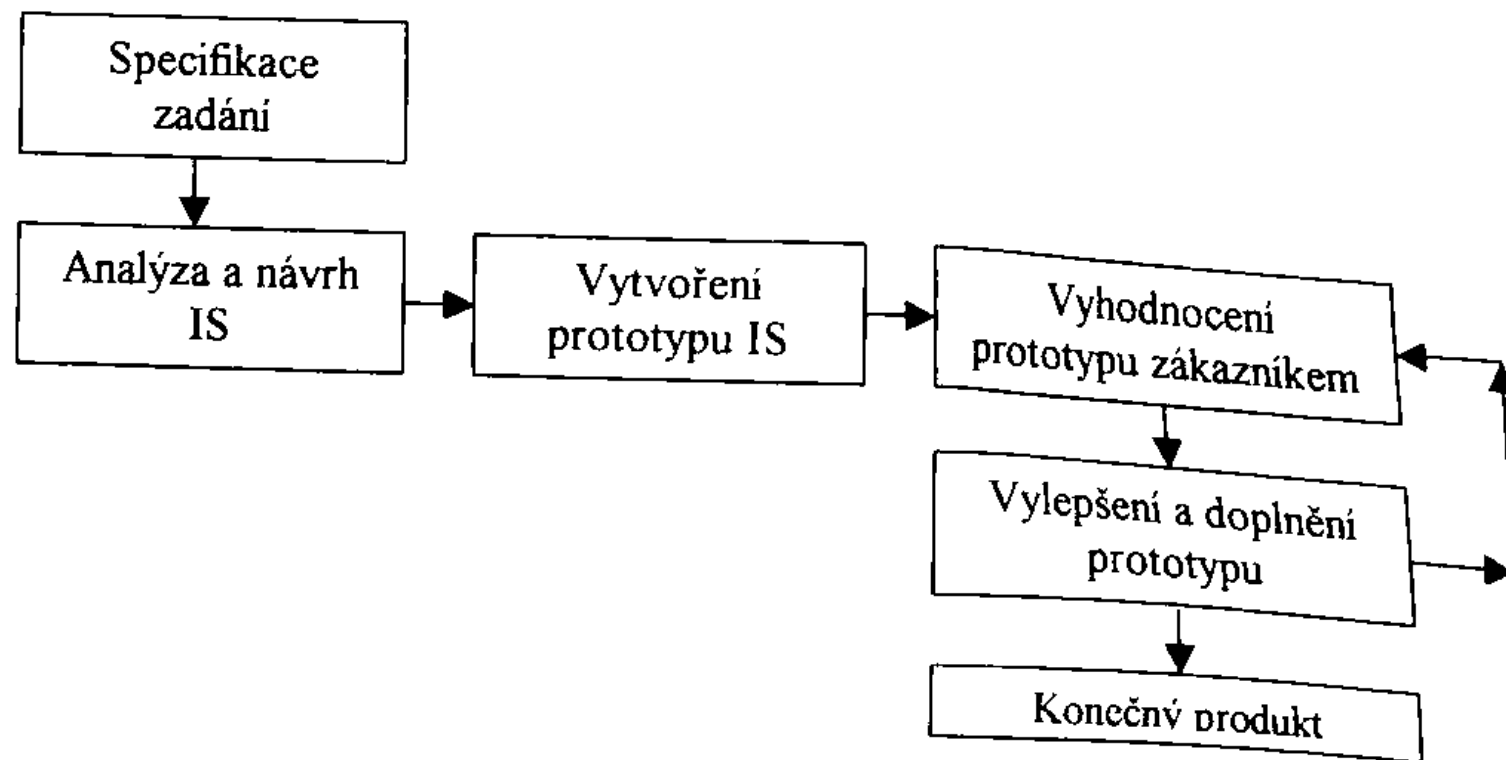
Obr 2.1 Model životního cyklu IS - "vodopád"

# Životní cyklus IS



Obr 2.2 Model životního cyklu IS - "výzkumník"

# Životní cyklus IS



Obr 2.3 Model životního cyklu IS - "prototyp"

# Životní cyklus IS

- Model „spirála“

Kombinuje prototypování s analýzou rizik. Jednotlivé etapy jsou cyklicky procházené vždy na vyšší úrovni podrobnosti analýzy, návrhu i implementace systému.

# Základní prostředky pro boj se složitostí vývoje IS (1)

- **Hierarchický rozklad problematiky**

rozdělení složitého systému na subsystemy a to až do potřebné úrovně podrobnosti.

Hierarchické rozdělení systému na subsystemy napomáhá plánovat, organizovat a kontrolovat práci vývojového týmu.



# Základní prostředky pro boj se složitostí vývoje IS (2)

- **Etapizace a iterace postupu řešení**

rozdělení složitého procesu vývoje IS na dílčí etapy. Každé etapě jsou přiřazeny cíle, úkoly, vstupy, výstupy, dokumentace, rizika, dílčí činnosti, odpovědné osoby, finanční náklady, apod.

Iterace znamená opakované provádění činností jednotlivých etap vždy na vyšším stupni porozumění problému. Účelem iterace je postupné zpracování problému na různých úrovních rozlišení – od hrubé představy o řešení až k podrobnému návrhu systému.

# Základní prostředky pro boj se složitostí vývoje IS (3)

- **Modelování a srovnávání modelů**  
základní technika používaná během vývoje IS.

# Základní prostředky pro boj se složitostí vývoje IS (4)

- **Použití grafických vyjadřovacích prostředků**

umožňují vytvořit si názornou představu o vyvíjeném IS.

Grafické vyjadřovací prostředky jsou součástí CASE (Computer Aided System Engineering) tj. nástroje pro podporu vývoje IS – automatizují rutinní činnost.

# Analýza, návrh IS

- **Princip ABSTRAKCE**

myšlenkový proces, vylučuje odlišnosti a zvláštnosti jednotlivých objektů či jevů a zdůrazňuje společné, obecné, podstatné vlastnosti sledované množiny objektů či jevů.

opakem

- **KONKRETIZACE**

přístup, při němž postupně vyčleňujeme z obecného specifické vlastnosti sledovaných objektů či jevů

# Analýza, návrh IS

## **3 stupně ABSTRAKCE:**

- Kategorizace
- Agregace
- Generalizace

**KATEGORIZACE:** nejnižší stupeň abstrakce, znamená seskupování prvků (jevů) do tříd (kategorií) podle kritérií, které si zvolíme k účelu sledování těchto prvků (jevů)

# Analýza, návrh IS

**AGREGACE** je abstrakcí, 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.

Př. agregace prvků – komponenty pc: monitor, klávesnice, HDD – jde o prvky daného celku, tj. počítače.

# Analýza, návrh IS

**GENERALIZACE** – abstrakce typu „specifický typ – obecný nadtyp“. Při generalizaci hledáme společné vlastnosti nadřízeného celku jakožto nositele specifikovaných společných vlastností (atributů).

Př. – společné vlastnosti prvků „správce počítače“, „operátor počítače“, „správce dtb. systémů“ - PRACOVNÍK SYSTÉMOVÉ PODPORY

Opakem GENERALIZACE je myšlenkový postup zvaný SPECIALIZACE.

# Vývojová klasifikace IS

Aspekt / typ IS	TPS	MIS	DSS	Expertné systémy	EIS
ÚČEL	len transakcie s dátami	najrôznejšie práce s dátami	pružné rozhodovanie	využitie znalostí experta	podpora vrcholového vedenia
VÝSTUPNÉ INFORMÁCIE	agregované správy	správy na vyžiadanie	informácie na rozhodovanie	hodnotenie, rady, vysvetlenia	kľúčové indikátory pre podniky
ORGANIZÁCIA DATABÁZY	jednotlivé súbory	súbory v interakcii	databázy a bázy modelov	databázy a bázy znalostí	napojenie: MIS a mimo podnik
PODPORA ROZHODOVANIA	slabá	pre rutinné úlohy	semištruktúrované problémy	aj neštruktúrované problémy	len orientačná
TYPICKÉ APLIKÁCIE	mzdy, evidencia materiálu	riadenie výroby, inventúry	strategické plánovanie	úvahy o investíciách	reakcia na okolie podniku
OBDOBIE VÝVOJA	1955-1960	1962-1970	pred r. 1980	po r. 1980	okolo r. 1990



# Typologie IS

**Průzkumové IS (Information Retrieval Systems)** *definované* jako množina lidí, technologií a procedur (software), které pomáhají vyhledávat údaje, informace a poznatkové zdroje lokalizované částečně v knihovnách nebo mimo ně. Informace o dostupných zdrojích jsou získávány, ukládány, vyhledávány a zpřístupňovány dle potřeb uživatelů.

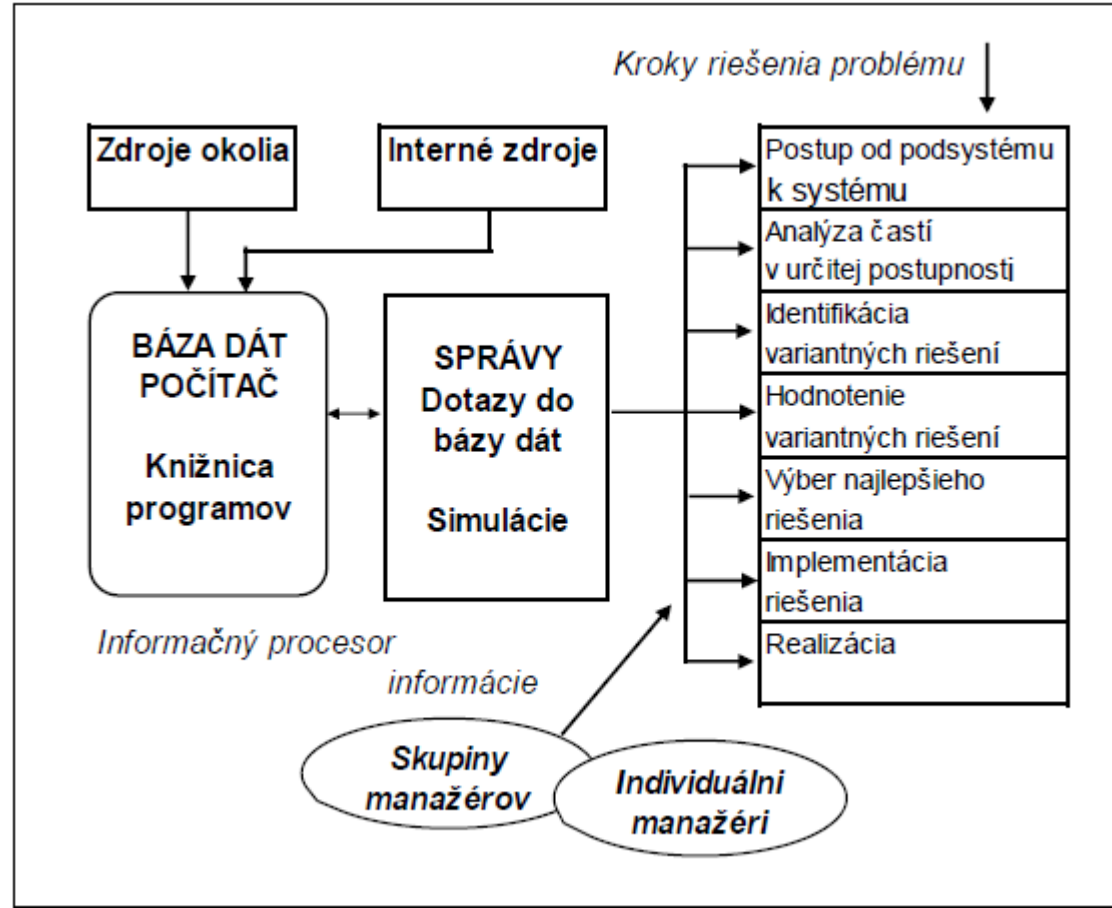
# Typologie IS (pokračování)

## **Informační systémy pro podporu rozhodování (Decision Support Systems) DSS**

jsou systémy se specifickými funkcemi orientovanými na pomoc manažerům při řešení problémů a v rozhodovacích procesech. Zahrnují lidi, procedury, software a účelové databáze.

Pomáhají identifikovat faktory, které vytváří problémy; poskytují možné cesty řešení problémů; pomáhají vybírat možnosti, které jsou k dispozici k řešení problémů.

# Obr. DSS



# Typologie IS (pokračování)

## **Expertní systémy (Expert Systems)**

jsou specifickým druhem informačních systémů, které pomocí software poskytují služby, které se očekávají od expertů. Jsou naprogramované imitovat myšlenkové postupy expertů a připravit návrhy rozhodnutí na výběr nejlepších partikulárních řešení problémových situací.

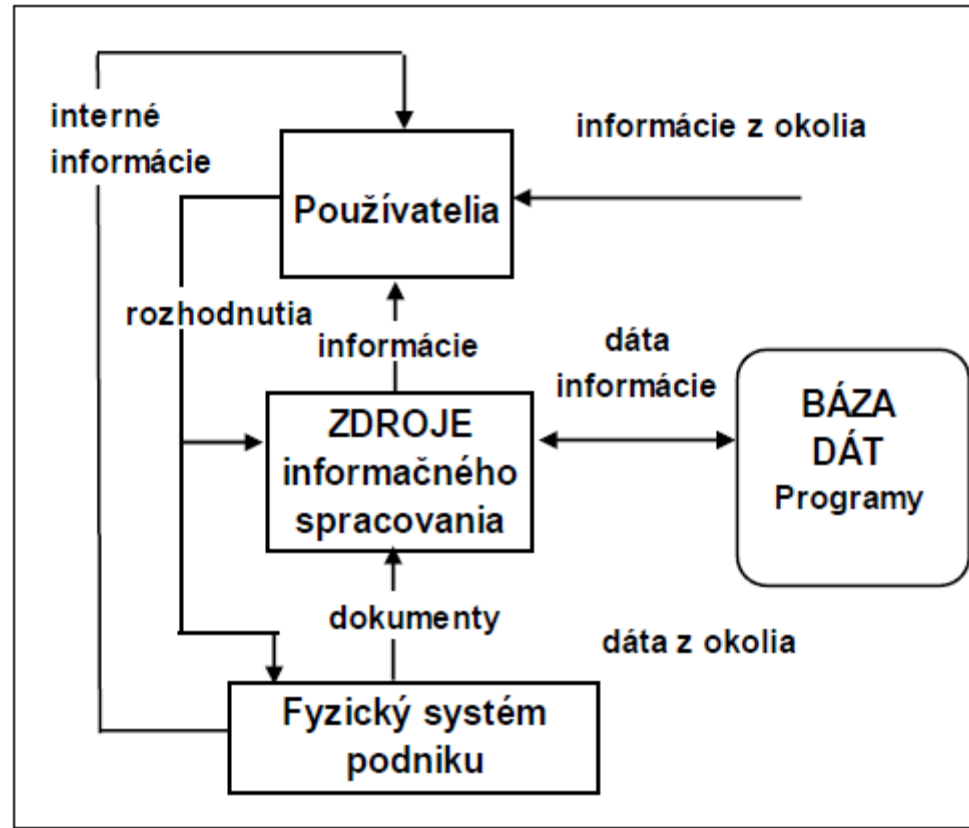
# Typologie IS (pokračování)

## **Manažerské informační systémy**

### **(Management Information Systems)**

zahrnují lidi, technologie a procedury, které slouží na organizační plánování, operační a řídicí přístup a využívání lidských a materiálních zdrojů.

# Obr. MIS



# Typologie IS (pokračování)

**Systemy na přímé řízení technologických procesů.**

**Jsou to systémy pracující v on-line-real-time (OLRT)** režimu určené na přímé řízení technologických procesů, např. prostřednictvím NC strojů (numeric control) připojených na počítače.

Integrováním přímého řízení procesů s organizací výroby, zásobování a expedice vznikají integrované výrobní informační systémy (Computer Integrated Manufacturing – CIM).

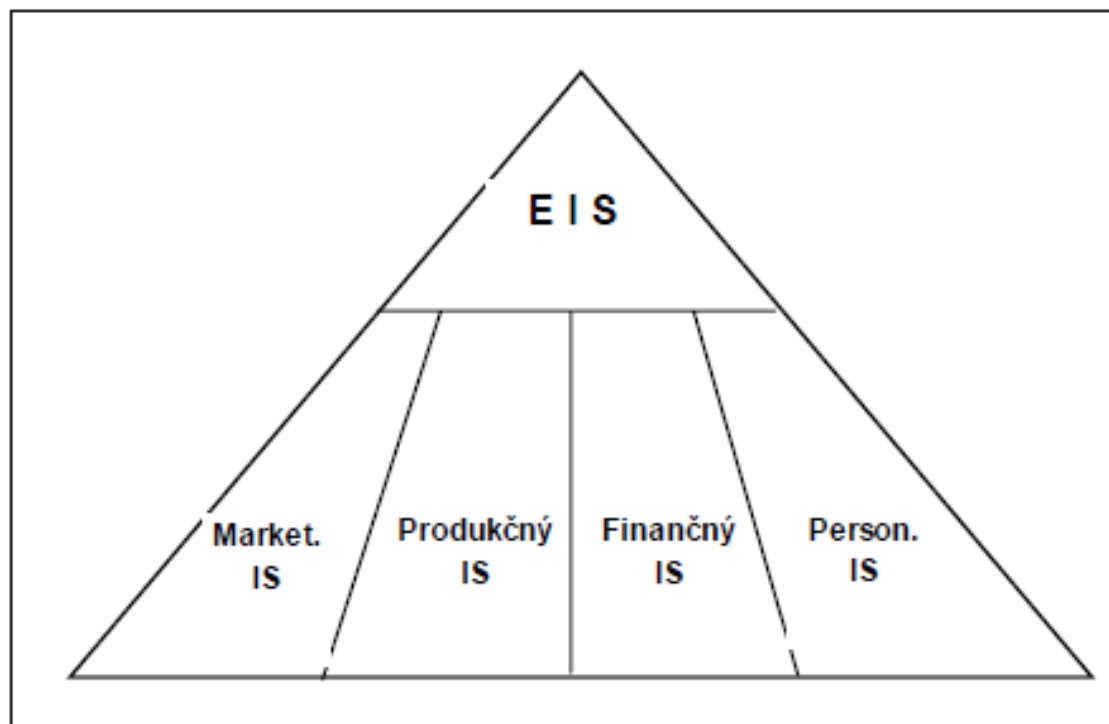
# Typologie IS (pokračování)

**Informační systémy pro podporu vrcholového řízení (EIS – IS)**, které zabezpečují vrchol řídicí pyramidy, slouží

především vrcholovému managementu podniku. Jsou to „osobní“ IS pro manažery na úrovni strategického plánování. Na rozdíl od MIS se EIS zajímá o informace z okolí podniku (technické inovace, trh, banka, konkurence apod.). EIS umožňují přístup k externím datům a sumarizují interní podnikové informace do nejvyšší úrovně agregace.



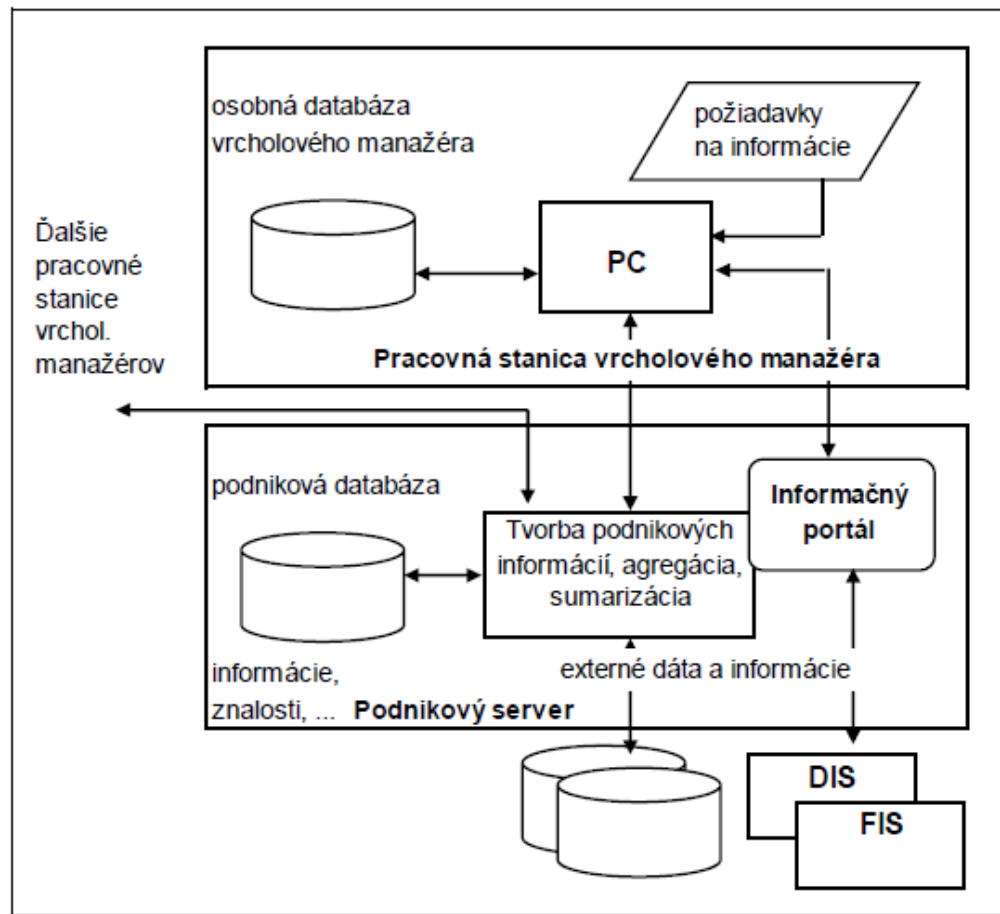
# Obr. EIS



# Srovnání MIS & DSS

<i>Kritérium</i>	<i>MIS</i>	<i>DSS</i>
Hľadisko podpory	organizácia	individuálne
Typ podpory	nepriama	priama
Podpora fáz riešenia problému	identifikácia, pochopenie a realizácia riešenia	všetky
Typy podporovaných problémov	všetky	semištruktúrované
Dôraz na	informácie	rozhodnutia

# Obr. EIS a jeho propojení na DIS a FIS přes IS



# Podpůrné IS

## **Kancelářské IS (Office Automation – OA)**

Obsahují textové procesory, faxy, kopírovací přístroje, zařízení na optické čtení dokumentů, el. Poštu apod.

# Podpůrné IS

## **Útvarové systémy (Departmental Systems – DS)**

Jsou často spojením TPS, DSS a OA, ale jejich rozsah je redukován na určitý útvar nebo místo.

**Dokumentografické (DIS) a faktografické (FIS) IS** zpracovávají a poskytují odborné a vědecké informace sloužící k podpoře strategického rozhodování a plánování. Nejčastěji existují propojení z EIS na DIS nebo FIS přes informační portály.

# Dělení IS dle obsahu výstupu

- agregované zprávy pro management (typické pro transakční IS),
- zprávy na vyžádání (Manažerské IS),
- informace pro rozhodování (IS na podporu rozhodování),
- hodnocení, rady, vysvětlení (expertní systémy),
- klíčové indikátory na řízení a strategické rozhodování v podnicích (exekutivní IS),
- adresy, příp. plné texty dokumentů (dokumentografické IS),
- fakta, souvislosti, sémantické mapy (znalostní a zprávečeské IS)

# Dělení IS dle jejich vztahu k systému řízení

