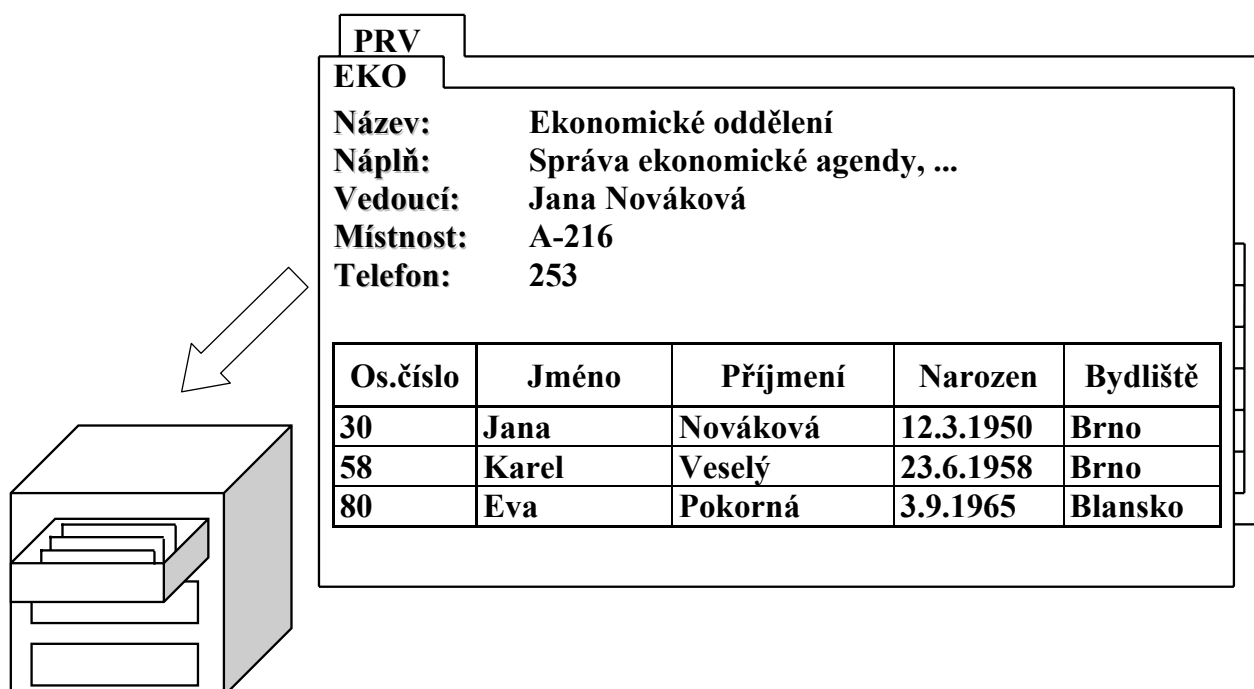


# 1 Úvod

1.1. Intuitivní vymezení pojmu „databáze“ .....	2
1.2. Historie vývoje zpracování dat .....	6
1.3. Základní pojmy .....	9
1.4. Abstrakce pohledu na data v databázi .....	11
1.5. Datové modely .....	13
1.6. Schéma databáze .....	17
1.7. Nezávislost dat .....	18
1.8. Databázové jazyky .....	19
1.9. Funkce SŘBD .....	21
1.10. Uživatelé databázového systému .....	22
1.11. Struktura databázového systému .....	23
1.12. Architektury databázových systémů .....	24
1.13. Typy DBS .....	28
1.14. Databázové systémy Oracle a Gupta .....	29
Literatura .....	30

## 1.1. Intuitivní vymezení pojmu „databáze“

Co je to *databáze*?- registr (papírová agenda) v elektronické podobě, v podobě **perzistentních** dat



- Typická podoba odpovídající databáze

### ODDĚLENÍ

zkratka	název	náplň	vedoucí	místnost	telefon
EKO	Ekonomické oddělení	Správa ekon. agendy, ...	30	A-216	253
PRV	Právní oddělení	Právní záležitosti	10	A-320	301
...	...	...	...	...	...

### ZAMĚŠTNANCI

os_číslo	jméno	příjmení	narozen	bydliště	oddělení
...	...	...	...	...	...
10	Josef	Floryán	18.3.1945	Brno	PRV
...	...	...	...	...	...
30	Jana	Nováková	12.3.1950	Brno	EKO
...	...	...	...	...	...
58	Karel	Veselý	23.6.1958	Brno	EKO
...	...	...	...	...	...
80	Eva	Pokorná	3.9.1965	Blansko	EKO
...	...	...	...	...	...

hodnoty z jiného sloupce (cizí klíč)

unikátní hodnoty (primární klíč)

- Typické operace s daty v databázi

- Vyhledání

„Kteří zaměstnanci pracují v ekonomickém oddělení?“

The screenshot shows the Oracle SQL Worksheet interface. The main window displays the following SQL query and its results:

```
SQLWKS> SELECT os_číslo, jméno, příjmení, bydliště
2> FROM Zaměstnanci
3> WHERE oddělení = 'EKO'
4>
```

OS_ČÍSLO	JMÉNO	PŘÍJMENÍ	BYDLIŠTĚ
30	Jana	Nováková	Brno
58	Karel	Veselý	Brno
80	Eva	Pokorná	Blansko

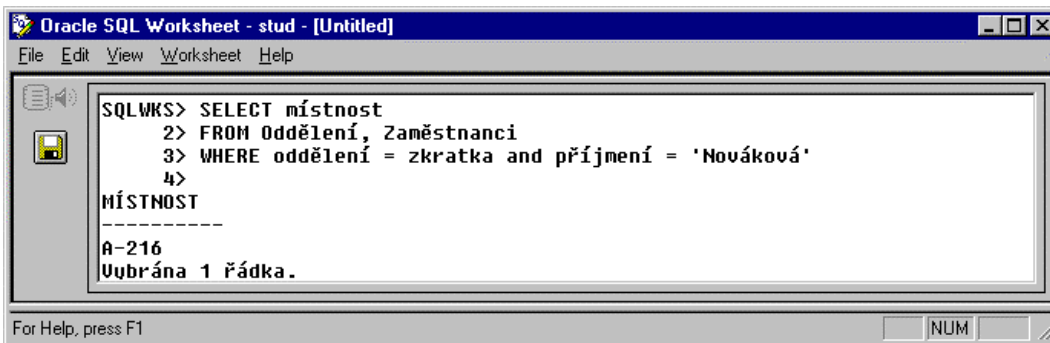
Počet vybraných řádek: 3.

The bottom window shows the same query without the results:

```
SELECT os_číslo, jméno, příjmení, bydliště
FROM Zaměstnanci
WHERE oddělení = 'EKO'
```

- lze spojovat informace z několika tabulek

„Kde sídlí oddělení, v němž pracuje paní Nováková?“



### ➤ Vložení

```
INSERT INTO Oddělení  
VALUES ('PER', 'Personální oddělení', 'Personální  
záležitosti', NULL, 'B-020', 125)
```

### ➤ Aktualizace

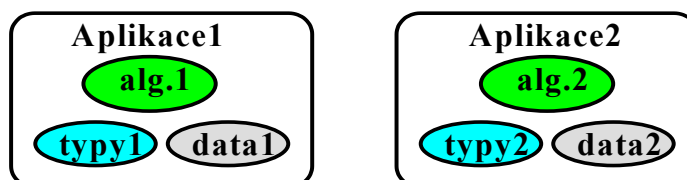
```
UPDATE Zaměstnanci  
SET oddělení = 'PER'  
WHERE os_číslo = 58
```

### ➤ Zrušení

```
DELETE FROM Zaměstnanci WHERE os_číslo = 58
```

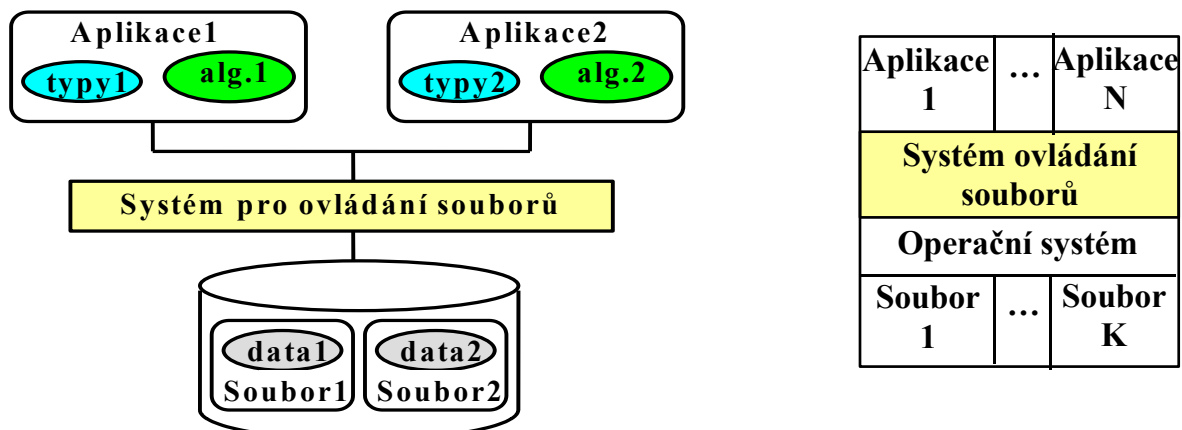
## 1.2. Historie vývoje zpracování dat

- 50. léta - vše v programu



- 60. léta - vytvoření **systemů pro ovládání souborů** (podporován hostitelským operačním systémem), vznik programovacích jazyků pro práci se soubory (např. COBOL)

- klasické „hromadné zpracování dat“



Př) Spořitelna:

Data v souborech: informace o klientech, jejich účtech, pobočkách spořitelny, transakcích s účty.

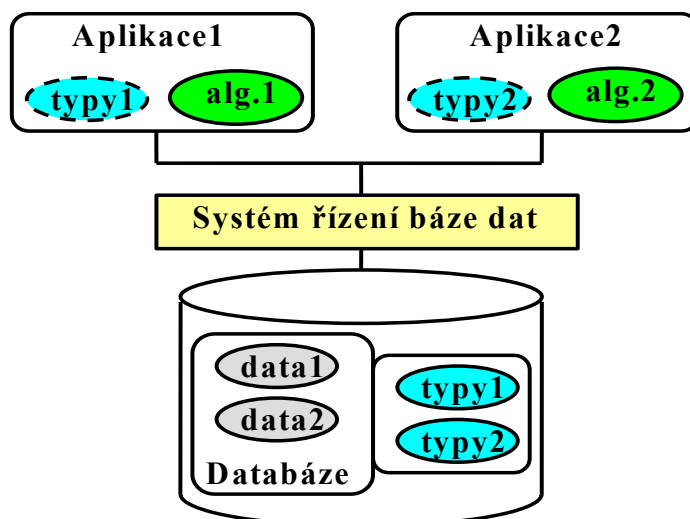
Aplikace: správa dat klientů účtů a poboček, provádění transakcí s účty, výpis z účtu, ...)

- přidávání souborů a aplikačních programů podle potřeby

➤ Nevýhody:

- **redundance** dat (informace se opakuje)
- nebezpečí **nekonzistence** (rozpory v datech)
- problémy s přístupem k datům pro **neplánované (ad hoc) dotazy**
- izolace dat (sbírání dat z jednotlivých souborů)
- problémy s **bezpečností dat** (omezený přístup)
- problémy **integrity** (implementace integritních omezení )

- 2. polovina 60. let – **databázová technologie** - systémy řízení báze dat (SŘBD, DBMS- Database Management System)



Základní přínos databázové technologie – dosažení jisté **nezávislosti dat** na uživatelských programech a naopak.

### 1.3. Základní pojmy

**Databáze (DB)** - **perzistentní** data, využívaná aplikačními systémy dané instituce (v klasické databázi **strukturovaná**).

**Perzistentní data** – data s dobou života překračující běh aplikačního programu i vypnutí počítače.

• Další vlastnosti dat databáze:

- **integrována** - lze chápat jako sjednocení několika datových souborů s odstraněním redundance (úplným nebo částečným)
- **sdílená** - typicky víceuživatelský přístup s případným omezením pohledu
- **bezpečná** - snadněji se realizuje omezení práv přístupu k datům
- snadnější zajištění **integrity dat** (implementace integritních omezení)

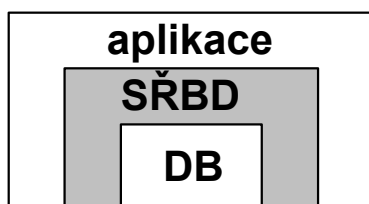
**Integrita dat** – správnost dat z hlediska splnění omezení (tzv. **integritních omezení**), která existují v reálném světě.

**Př) účet musí mít vlastníka, rodné číslo musí splňovat podmínku dělitelnosti 11, zadaná pobočka musí existovat, ...**

**Konzistence dat** – nerozpornost dat . Dojde-li k porušení konzistence, říkáme, že data jsou **nekonzistentní**.

**Př) V databázi je uložena dvakrát adresa klienta s různou hodnotou, po převodu částky z účtu A na účet B je součet na obou účtech jiný než před převodem, ...**

**Systém řízení báze dat** - programová vrstva řešící operace nad DB.



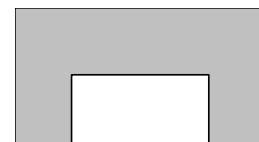
Cíl: odstínění uživatele (aplikace) od technických detailů

- operace: vytvoření DB, tabulky, vyhledávání, vkládání, ...

**Databázový systém (DBS)** - systém, který v širším pojetí zahrnuje:

- technické prostředky
- data - **DB**
- programové vybavení - **SŘBD**, vývojové prostředky, knihovny, ...
- uživatelé DB

**DBS (v užším smyslu) = DB + SŘBD**



## 1.4. Abstrakce pohledu na data v databázi

Jedním z důležitých úkolů DBS je poskytnout uživatelům abstraktní pohled na data (jsou skryty detaily uložení a správy dat).

reálný svět



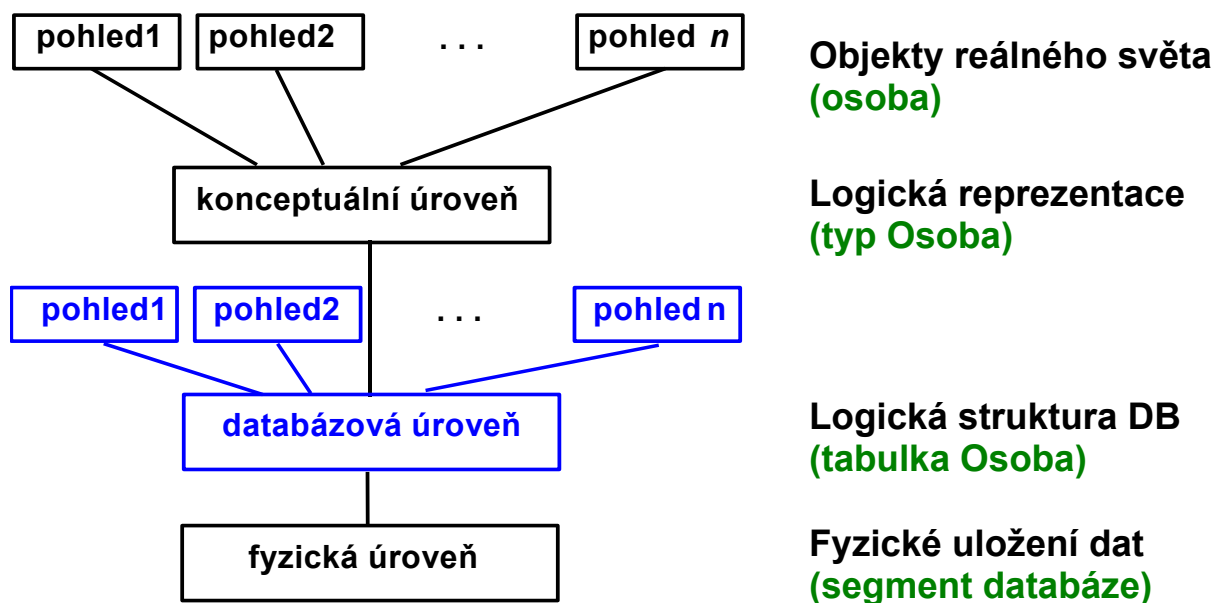
různé abstrakce pohledu na data

uložená data

### • Základní úrovně abstrakce dat:

- **fyzická (interní) úroveň** - popisuje data, **jak** jsou skutečně uložena.
- **konceptuální (logická) úroveň** - popisuje **jaká** data jsou skutečně uložena v databázi a jaké vztahy mezi nimi existují.
- **úroveň pohledů (externí)** – popisuje, jaká data vidí jednotliví uživatelé, tj. obecně pouze část databáze, která představuje data reprezentující objekty reálného světa, viditelná jednotlivými uživateli.

### • Tříúrovňová architektura (ANSI/SPARC) - 70. léta, snaha o standardizaci:



⇒ **schéma**: externí, konceptuální, databázové, fyzické

## 1.5. Datové modely

**Datový model** - kolekce konceptuálních nástrojů pro popis objektů reality, resp. je reprezentujících dat, vztahů mezi nimi, sémantiky a integritních omezení.

- Rozdělení podle úrovně modelování:

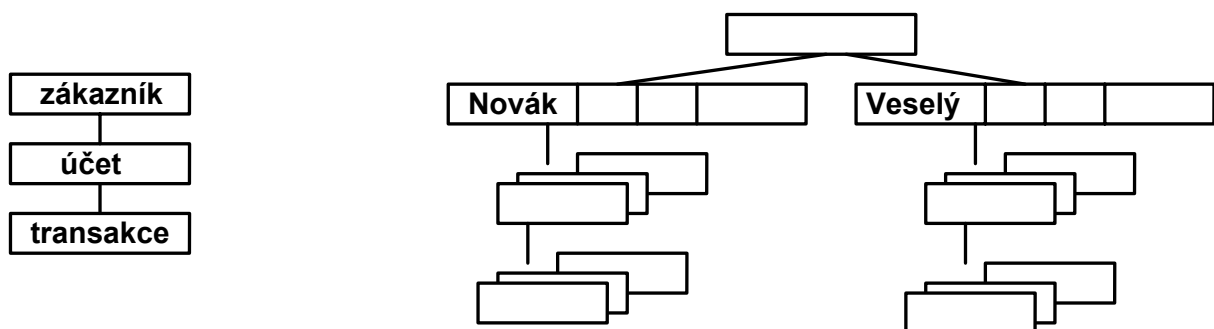
➤ **Logické modely** - popisují data na úrovni konceptuální a pohledů

- modely pro modelování objektů reálného světa (ER model, OO model, funkcionální model, ...) – **konceptuální modelování**.
- databázové modely definující logickou organizaci dat v databázi (relační, síťový, hierarchický, OO, objektově relační, ...)

➤ **Modely fyzických dat** - popisují data na fyzické úrovni.

**Pozn.:** Někdy se v souvislosti s návrhem relační databáze logickým modelem označuje ER diagram a fyzickým modelem podobný diagram ukazující tabulky a vztahy mezi nimi.

- **Hierarchický model** – 2. polovina 60. let. a dále



**Struktura dat:** podobná síťovému modelu, ale záznamy jsou organizovány jako stromy a ne sítě.

**Manipulace s daty:** „**navigační programování**“ po struktuře.

**Př) IMS (IBM, 1968)**

- **Síťový model** – konec 60. let a dále



**Struktura dat** - množina záznamů + pojmenovaných vazeb (obdobu ukazatelů)..

**Manipulace s daty** – navigační: umístění na záznam s danou hodnotou pole, nastavení na první členský záznam, na další, vytvoření/zrušení/modifikace záznamu, zapojení/vyřazení/přepojení z vazby, ...

**Standardizace:** standard DBTG CODASYL (konec 60.let-1981)

### Př) IDMS (Computer Associates)

- **Relační model** – 1970 teorie, 1975 System R (IBM)

**Struktura dat:** Na konceptuální úrovni jsou data strukturována do tabulek (tzv. **normalizovaných** – hodnoty v tabulce musí být z hlediska významu **atomické (skalární)**).

**Manipulace s daty:** výběr z tabulky, vložení/změna/zrušení řádku tabulky a jednoduchá navigace po řádcích tabulky.

**Standardizace:** standard SQL (Structured Query Language) – 1986 – dosud.

č_klienta	jméno	adresa
100	Pavla Zelená	Hájkova 1, Brno
200	Jan Novák	Cejl 8, Brno

č_úctu	stav	majitel
800	10856	100
557	117543	200
486	57800	100

**Výhody:** Jednoduchá struktura, jednoduché operace.

**Nevýhoda (relativní):** Informace je rozptýlena po tabulkách, u složitějších dotazů je třeba informace z tabulek **spojovat**.

**Př) Kdo je majitelem účtu č. 557?**

**Př) Oracle, DB2 (IBM), Microsoft SQL Server, MySQL, Gupta SQLBase**



## 1.6. Schéma databáze

**Schéma databáze** - metainformace popisující data v databázi.

- Schémata v architektuře ANSI/SPARC- externí schéma, konceptuální (logické), interní (fyzické) schéma.
- Logické schéma databáze odráží použitý databázový model.
- Informace o schématu typicky uložena v **systemovém katalogu** (datovém slovníku).

## 1.7. Nezávislost dat

**Nezávislost dat** - schopnost modifikovat definici schématu bez vlivu na schéma na vyšší úrovni .

- Úrovně nezávislosti:

- **Fyzická nezávislost dat** - schopnost modifikace fyzického schématu bez nutnosti přepsat aplikační programy.

**Př) vytvoření efektivnější přístupové metody (index)**

- **Logická nezávislost dat** - schopnost modifikovat logické schéma bez nutnosti přepsat aplikační programy.

**Př) přidání sloupce tabulky**

- jedna ze základních předností databázového přístupu.
- snazší dosažení fyzické nezávislosti dat.
- analogie s abstraktními datovými typy - detaily implementace jsou uživateli skryty.

## 1.8. Databázové jazyky

Jazyk, kterému „rozumí“ SŘBD, někdy se označuje jako dotazovací.

- Součástí databázového jazyka - musí existovat prostředky pro:
  - Specifikaci schématu databáze
    - ⇒ **jazyk pro definici dat (DDL - Data Definition Language)**
      - výsledkem překladu příkazů jsou informace uložené ve **systémovém katalogu (slovníku dat)** + případně vytvoření dané struktury.
  - Pro manipulaci s daty v databázi
    - ⇒ **jazyk pro manipulaci s daty (DML - Data Manipulation Language)**
      - poskytuje prostředky pro databázové operace (vyhledání, vkládání, modifikace, rušení) v daném datovém modelu.
  - Zpravidla ještě další prostředky:
    - pro řízení přístupu k datům, integritní omezení, ...
      - ⇒ **jazyk pro řízení dat (DCL - Data Control Language)**.
    - transakční zpracování, ...

- Přístup k databázi z aplikačních programů:
  - Použití **specializovaných databázových jazyků** – rozšiřují základní, typicky neprocedurální, databázový jazyk (typicky SQL) o další příkazy a operace a datové typy.
    - Př) PL/SQL (Oracle), SAL (Gupta), Transact SQL (Microsoft SQL Server, Sybase Adaptive Server Enterprise), Informix 4GL, ...**
  - Použití **obecných programovacích jazyků**:
    - Poskytnutí rozhraní pro přístup k databázi
      - Př) nativní (OCI – Oracle Call Interface, ...), standardizované (ODBC – Open DataBase Connectivity, JDBC – Java DataBase Connectivity)**
    - Umožněním začlenit příkazy základního databázového jazyka do zdrojového textu programu (typicky se speciální direktivou) a zpracování preprocesorem.
      - Př) Hostitelská verze SQL pro Pascal, C, ..., SQLJ**

## 1.9. Funkce SŘBD

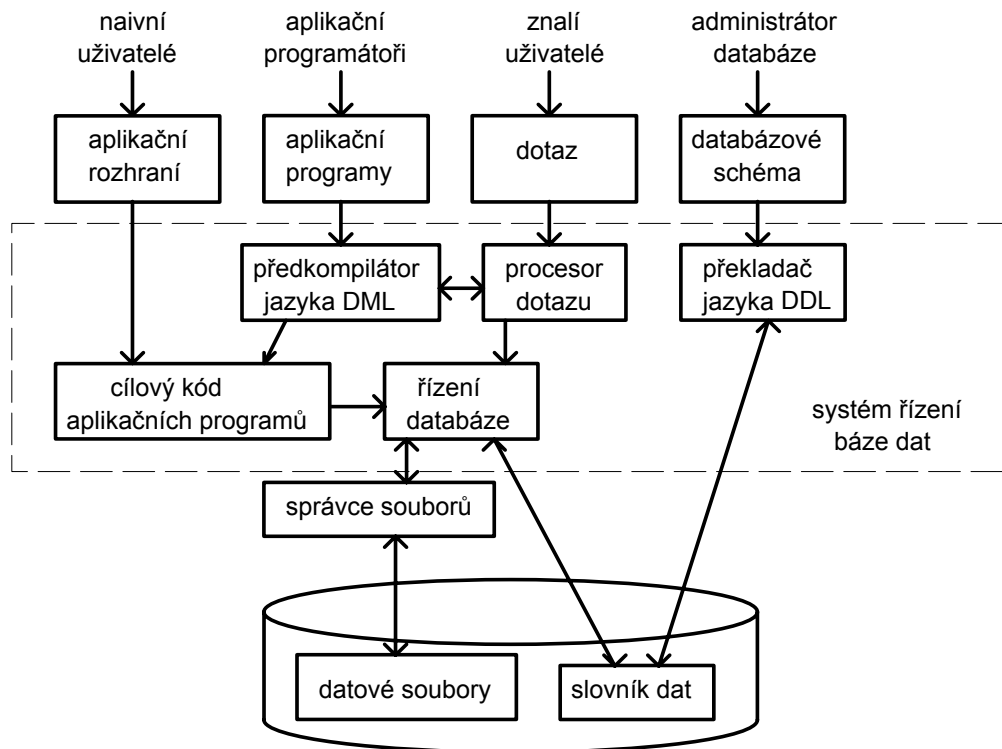
SŘBD tvoří rozhraní mezi aplikačními programy a uloženými daty.

- **Základní funkce SŘBD**
  - definice dat, vytváření slovníku dat
  - manipulace s daty
  - zajištění bezpečnosti a integrity dat
  - zotavení po chybách a souběžný přístup - transakční zpracování
  - zajištění co nejvyšší výkonnosti

## 1.10. Uživatelé databázového systému

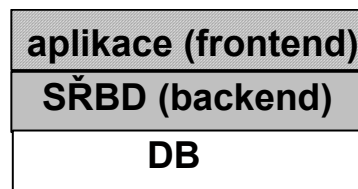
- **Kategorie uživatelů DBS**
  - **Administrátor databáze** - zajišťuje centrální kontrolu nad daty a programy, které je využívají, plní rozhodnutí *správce dat*.  
Mezi jeho funkce patří:
    - definice schématu,
    - definice paměťové struktury a přístupové metody,
    - modifikace schématu a fyzické organizace,
    - přidělování práv přístupu
  - **Aplikační programátor** - vytváří aplikační programy využitím prostředků jazyka pro manipulaci s daty vloženými do *hostitelského jazyka* nebo použitím problémově orientovaných neprocedurálních jazyků.
  - **Znalý uživatel** – nepíše programy, ale je schopen formulovat požadavky v databázovém jazyce pro manipulaci s daty.
  - **Naivní uživatel** - komunikuje se systémem pouze prostřednictvím aplikačního programu.

## 1.11. Struktura databázového systému

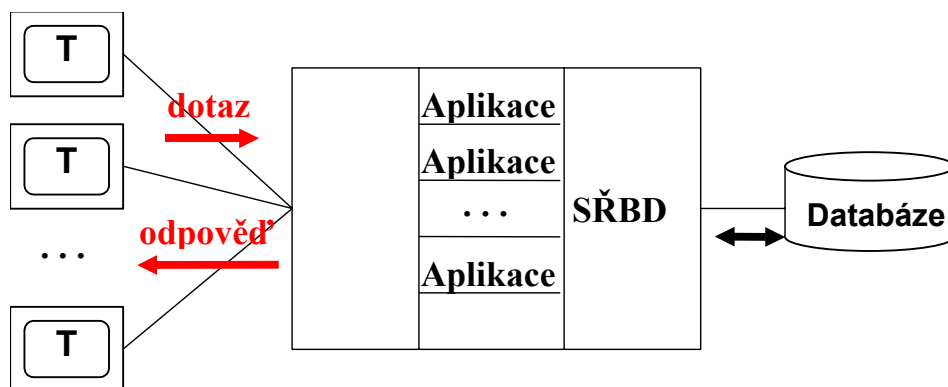


## 1.12. Architektury databázových systémů

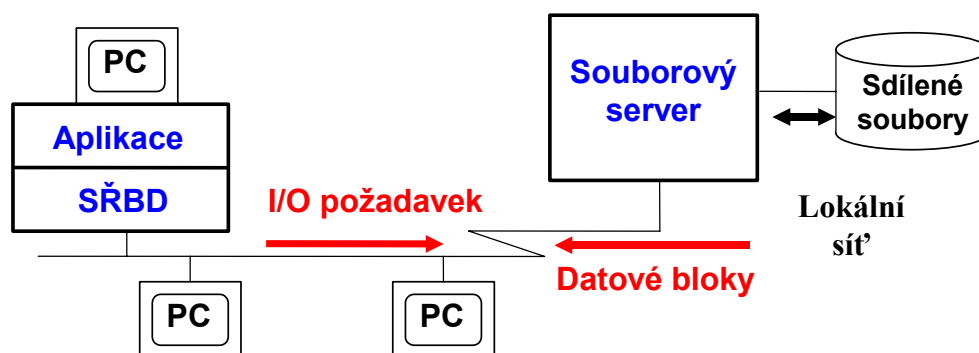
- DBS v širším pojetí (včetně aplikací) zahrnuje dvě základní vrstvy:
  - **SŘBD** (procesy na pozadí – *backend, server*) – procesy realizující všechny základní funkce SŘBD.
  - **Aplikace** (procesy na popředí – *frontend, klient*) – aplikace využívající služeb SŘBD.



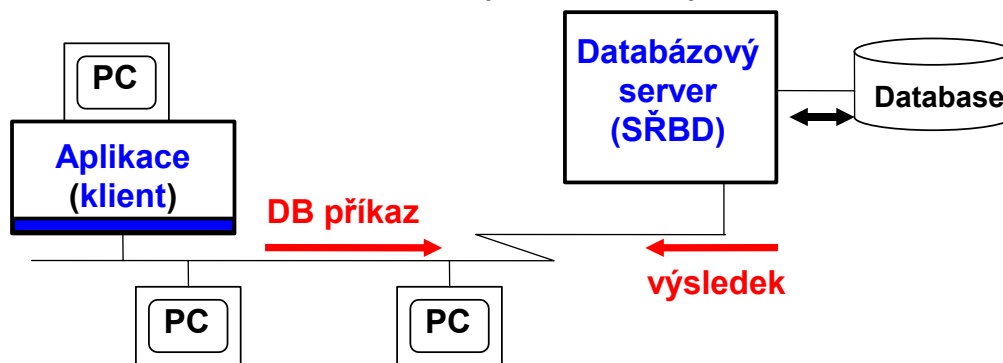
- Architektura typu „mainframe“



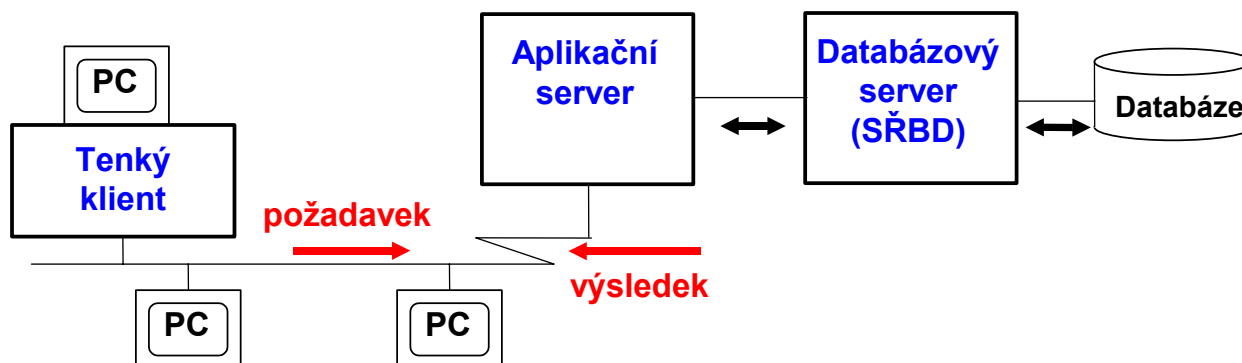
- Architektura typu „PC file server“



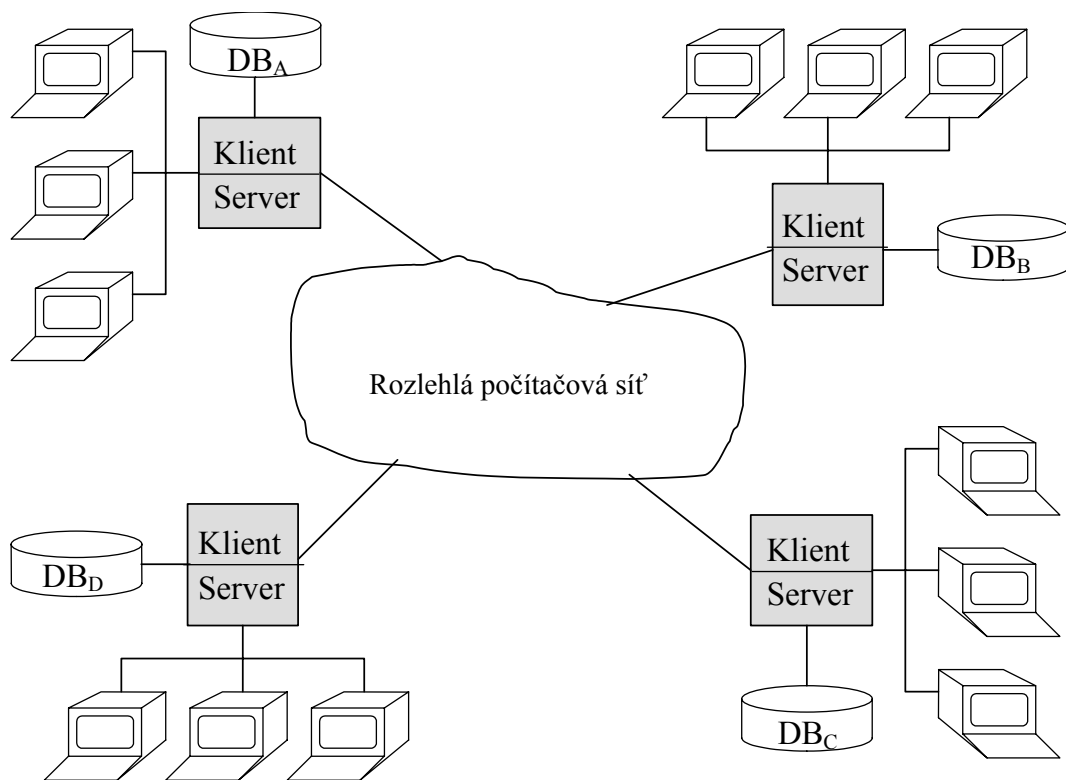
- Architektura „klient/server“ (dvouvrstvá)



- Vícevrstvá architektura



## • Distribuovaný databázový systém



### 1.13. Typy DBS

#### • Podle použitého modelu dat:

➤ **předrelační** (hierarchické, síťové) - IMS, IDMS

➤ **relační**:

- s architekturou typu „mainframe“: systémy v 2.pol. 70 let
- s architekturou typu „PC file-server“: dBASE, FoxBase, ...
- s architekturou klient-server: IBM DB2, Oracle , Microsoft SQL Server, Informix, Gupta SQLBase, ..

➤ **postrelační** (objektově orientované, deduktivní, ...) - GemStone, ODE, O<sub>2</sub>, ObjectStore, Jasmine,...

- Téměř všechny současné komerční databázové systémy jsou relační (případně objektově-relační).

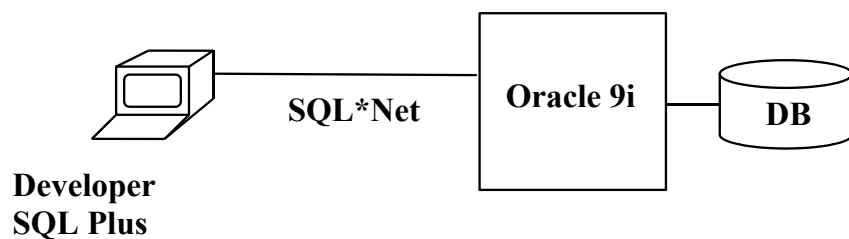
#### • Mezi základní rysy relačních systémů patří:

1. data jsou v databázi na logické úrovni organizována do tabulek (jediná datová struktura na této úrovni)
2. operace nad tabulkami vytvářejí opět nové tabulky

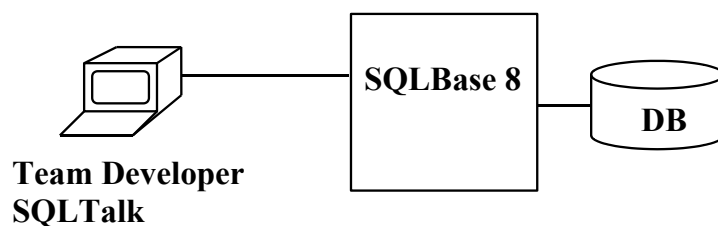
## 1.14. Databázové systémy Oracle a Gupta

- relační systémy s architekturou klient-server

- Oracle



- Gupta



## Literatura

1. Silberschatz, A., Korth H.F, Sudarshan, S.: Database System Concepts. Fourth Edition. McGRAW-HILL. 2001, str. 1 - 24.
2. Pokorný, J.: Databazová abeceda. Science, Veletiny, 1998, str. 15 - 18.
3. Pokorný, J.: Dotazovací jazyky. Science, Veletiny, 1994, str. 12 - 20.