

Databáze

[Nástin vývoje databázových systémů](#)

[Základní pojmy databázových systémů](#)

[Modely databázových systémů](#)

[Hlavní problémy tradičního zpracování souborů](#)

[Výkonné databáze](#)

Nástin vývoje databázových systémů

Na začátku 60. let minulého století byla v USA utvořena konference o jazycích systémů na bázi dat (*Conference on Data System Languages = CODASYL*), jejímž výsledkem byl programovací jazyk určený pro zpracování dat **COBOL** (Common Business Oriented Language). COBOL obsahuje pohodlné prostředky pro práci s externími soubory.

Koncem 60. let minulého století se začínají objevovat první **systémy pro podporu řízení databází (SŘBD)**.

V roce 1970 vzniká databázový model, který je nazván **relačním databázovým modelem** a jsou stanoveny tři minimální **podmínky relačnosti**:

1. **všechna data a všechny záznamy o datech v databázi jsou uloženy v tabulkách**
2. **uživatel je odstíněn od fyzické struktury dat** (nepotřebuje vědět jak jsou data fyzicky uložena)
3. **musí existovat jazyk, pomocí něhož lze s daty nakládat – tj. zajistit minimálně operace selekce, filtrace, projekce a spojení** (viz dále).

Základním databázovým systémem (DS), na který navázala řada dalších úspěšných produktů, byl systém **dBASE II** od americké firmy Ashton-Tate.

Produkty firmy Fox Holdings Inc. vycházely z DS dBASE II. První verze produktů této firmy se nazývaly **FoxBase**. Nad nimi byla posléze vystavěna nadstavba - **Fox-Pro**. Ta umožňuje příjemný způsob ovládání pomocí **systému menu a dialogů**. DS FoxPro se stal u nás nejrozšířenějším prostředkem pro tvorbu informačních systémů.

Vývoj tradičních relačních produktů pokračoval dále, např. Microsoft má jednak následovníka FoxPro - **Visual FoxPro** a vlastní nový relační DS **Access**.

Existují i další DS - **Informix, Oracle** aj. Vedle tradičních relačních DS existují i objektové DS a objektově relační DS.

U nás vznikly v 80. letech zcela originální DS pro DOS (Redap a FAND). Firma Software602 vytvořila pozoruhodný produkt pro Windows pod názvem **WinBase602**.

V současné době se relační a objektové DS se propojují s **internetovými technologiemi**, takže součástí technologie pro Windows .NET je **SQL Server** a DS **MySQL**, který je propojen jak s PHP skripty, tak i s celou řadou dalších programovacích jazyků (včetně .NET) na různých platformách – Windows, Linux aj.

Základní pojmy databázových systémů

Databáze (banka dat, angl. *Database*) je souhrn údajů o nějakých objektech (např. o účastnících telefonní sítě nebo o knihách uložených v knihovně). Všechny

objekty jedné databáze jsou stejného druhu, jsou popsány nějakou předem určenou množinou údajů, jako je např. příjmení, křestní jméno, adresa a telefonní číslo.

Informační systém je širší význam pojmu databáze. Znamená nejen vlastní datovou základnu, ale i souhrn metod a postupů, které umožňují s údaji v této datové základně manipulovat.

Záznam (věta, angl. *record*) je souhrn všech údajů o jednom objektu uloženém v databázi (např. o jedné knize nebo o jednom účastníkovi telefonní sítě).

Všechny záznamy v jedné databázi mají stejnou **strukturu**: Obsahují několik vždy stejných **položek** (polí, angl. *fields*), které odpovídají jednému údaji sledovanému u objektů databáze (např. jméno autora, signatura knihy nebo telefonní číslo). Každá položka má svůj **název a typ**. Název musí jednoznačně identifikovat, o kterou položku se jedná (např. JMÉNO, PŘÍJMENÍ, ADRESA a TELEFONNÍ ČÍSLO). Typ položky určuje, jakého druhu jsou údaje v položce obsažené. Název a typ položky zůstává stejný pro všechny záznamy jedné databáze.

Klíčové položky (klíče, angl. *key*) jsou položky, podle jejichž hodnoty je např. možné vyhledat v databázi zadaný záznam.

Pod pojmem **databázový systém (DS)** budeme rozumět soubor programových prostředků, které usnadňují přístup k datům uloženým v databázi. Databázový systém je tedy prázdný, pro jeho konkrétní využití je potřeba umístit do databází konkrétní data. Zde je třeba si uvědomit, že když říkáme „používáme databázi FoxPro“ (např.), dopouštíme se chyby, protože program FoxPro není databáze, ale databázový systém.

Mezi českými pojmy pro teorii databází existuje stále ještě určitá nejednoznačnost. DS ve významu uvedeném o pár řádků výše se v některé literatuře nazývá **Systém pro řízení báze dat (SRBD, angl. Database Management System)**. Pod pojmem DS se pak rozumí DS naplněný určitými daty, tedy SRBD spolu s jeho **datovou základnou**.

My budeme DS chápat jako systém prázdný. DS spolu s konkrétními daty a konkrétními způsoby, jak s těmito konkrétními daty zacházet, budeme nazývat **informační systém**.

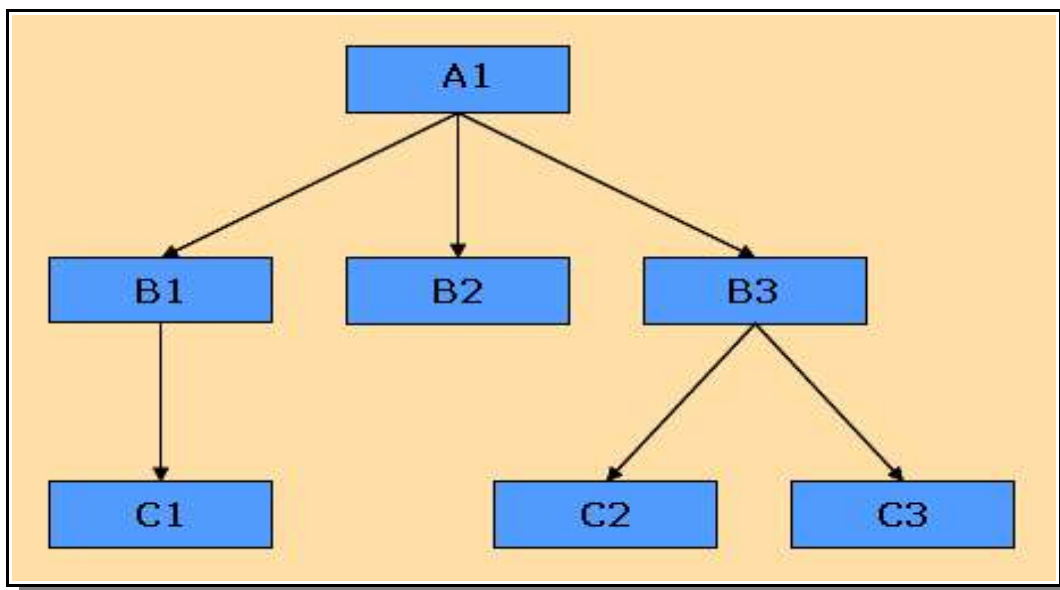
Pod pojmem **databáze** pak budeme i nadále rozumět jeden **databázový soubor**, tedy množinu dat, která se vyznačují stejnou strukturou a stejným použitím.

Modely databázových systémů

Hierarchický

Veškeré **vztahy mezi prvky** (uzly) jsou **binární** – tj. vztahy mezi dvěma prvky. Existují vztahy:

- **1:1** - prvek z vyšší úrovně (rodič) je vztahu právě k jednomu prvku z úrovně o jednu nižší (potomek) a naopak
- **1:N** – rodič je ve vztahu k více než jednomu potomkovi. Každý potomek má právě jednoho rodiče.

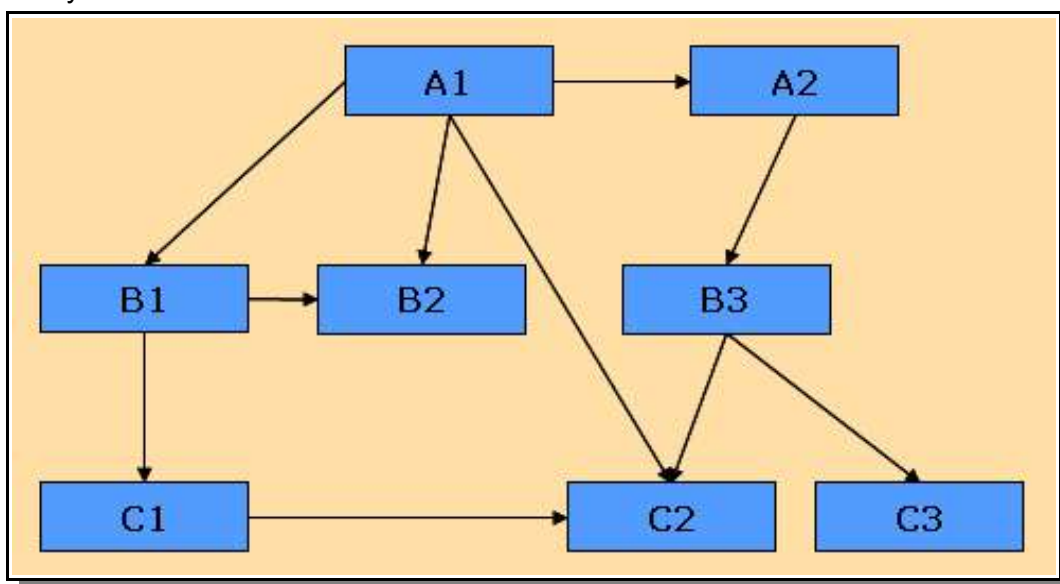


Obr. 1 – Hierarchický model

Síťový

Tento datový model překonává omezení síťového modelu dat týkající se vztahů mezi prvky na jedné hierarchické úrovni a zavádí vztah **M:N**.

Jeho nevýhodou je (stejně jako u síťového modelu) náročná každá změna jeho struktury.



Obr. 2 – Síťový model

V prvních dvou modelech jsou data uložena víceméně nezávisle na sobě a vztahy mezi nimi jsou vyjádřeny pomocí **směrníků** (odkazů, angl. *pointers*). U **hierarchického modelu** jsou směrníky vedeny **systémem shora dolů**, např. od dat popisujících nějaký celek k datům popisujícím jeho části. U **síťových modelů** jsou směrníky vedeny i **mezi daty na stejné hierarchické úrovni**.

Relační model

Je nejrozšířenější model DS. Data seřazena do **tabulek**. Jedna tabulka odpovídá tomu, co nazýváme **databázový soubor**. Skládá se z řádků a sloupců. **Každému objektu (entitě)** je přiřazen **jeden řádek** (záznam). **Sloupce** reprezentují jednotlivé **atributy** objektů (jednotlivé druhy údajů o objektech, které chceme uchovávat). Do-

sud jsme je nazývali položkami databáze. Datový objekt je plně charakterizován seznamem svých atributů.

Pořadí sloupců v tabulce není podstatné. Na sloupec, tedy na hodnotu atributu, se odvoláváme pouze pomocí názvu tohoto sloupce. Stejně tak není podstatné ani pořadí řádků. Pokud chceme objekty uložené v databázi, tedy řádky tabulky, nějak seřadit, použijeme k tomu hodnotu nějakého jejich atributu. Takový atribut nazýváme **jednoduchý klíč**. Pokud se nějaký jednoduchý klíč vyskytuje ve více tabulkách, je možné spojit pomocí tohoto klíče více tabulek do jedné.

Atribut musí mít jedinečný název v rámci jedné tabulky, avšak v rámci jedné databáze se může stejný název atributu vyskytovat ve více tabulkách. Proto když budeme chtít vybírat data z těchto tabulek najednou, je nutno přistupovat k těmto atributům přes jméno tabulky, ve které jsou umístěny.

Základní operace relačního databázového modelu:

1. **Spojování tabulek** na základě shodných hodnot některých atributů,
2. **Projekce tabulky** - zobrazení pouze určité podmnožiny atributů - sloupců tabulky, položek databáze,
3. **Filtrace tabulky** - zobrazení údajů o některých objektech, zobrazení jen některých řádků tabulky či záznamů databáze na základě vyhodnocení nějaké logické podmínky,
4. **Faktorizace tabulky** na základě spojení těch řádků tabulky, které mají stejnou hodnotu některých atributů.

Počítačové DS založené na relačním modelu obsahují obvykle prostředky pro řešení těchto čtyř operací, a mimo to i příjemné **uživatelské rozhraní (interface)**.

Hlavní problémy tradičního zpracování souborů, které vedly ke vzniku databázových systémů

- **Redundance a konzistence.** V případě, že data jsou v roztroušených souborech, dochází k opakování stejných údajů v různých souborech. Např. týž pracovník je zanesen v souborech dat osobních, kvalifikačních, mzdových aj. Opakovanému výskytu stejného údaje se říká **redundance** (nadbytečnost). Redundance nad potřebnou míru (někdy je jistá nadbytečnost potřebná – např. z důvodu vzájemných kontrolních vazeb dat) je negativním jevem. Při změně údaje (např. pracovnice se vdá a změní příjmení) je nutné zajistit **konzistentnost** dat, tj. aby změna byla provedena ve všech souborech, a to lze v případě izolovaného zpracování souborů zajistit jen těžko.
- **Obtížná dosažitelnost dat.** Pokud se zpracování dat provádělo speciálními programy, které obsahovaly pevné dotazy, které neumožňovaly jejich úpravu, bylo obtížné získat odpovědi na jiné než předem připravené dotazy.
- **Současný přístup více uživatelů.** Tradiční souborový přístup se ukázal jako nevhodný např. v systémech rezervace letenek, kdy v jednom okamžiku si někteří uživatelé prohlížejí možné lety či obsazení letadel, jiní rezervují letenky, další ruší rezervace apod.
- **Ochrana dat před zneužitím.** V tradičním souborovém přístupu se zajišťuje špatně, neboť uživatel musí při spouštění každého uživatelského programu procházet kontrolou oprávnění, přičemž neexistuje centrální zpráva hesel, takže musí znát své heslo pro každý uživatelský program.
- **Integrita.** Integrita znamená, že data musí odpovídat reálným vlastnostem objektů. Takže např. hmotnost musí být číslo nezáporné, neexistuje datum 30. února, zboží musí mít stejné číslo v ceníku i na faktuře. Udržet integritu v roztroušených souborech je obtížné.

Jak uvedené řeší DS?

1. Aplikační programy a datové soubory jsou odděleny.
2. Přístup k datům je možný pouze prostřednictvím DS, nikoli přímo.
3. Dotazy nejsou pevné, existují prostředky pro tvorbu téměř libovolné dotazy.
4. DS umožňuje přístup více uživatelům najednou při zajištění ochrany před zneužitím.

Výkonné databáze

V dnešní době se stále více prosazují výkonné síťové databáze, které pracují na základě dotazovacího jazyka SQL. Databáze pracují technologiemi:

- a) **file server** - řídicí počítač na požadavek poskytne potřebná data,
- b) **klient server** - řídicí počítač vyhodnotí dotaz a pošle zpět odpověď, třeba jen jedinou hodnotu - odpadá přenos velkých objemů dat a šetří se tak telekomunikační poplatky.

Objektově orientovaný model

Data jsou vztažena k **objektům**, které odpovídají objektům z reálného světa. Objektově orientovaný model se skládá z množiny objektů, které spolu a se systémem komunikují prostřednictvím **zpráv**.

Objekt je možno popsat:

- ◆ množinou vlastností objektu – tzv. **instančních proměnných**, které obsahují data objektu a odpovídají s atributy v relačním modelu,
- ◆ množinou metod,
- ◆ množinou zpráv, na které objekt reaguje.

Objektově relační model

Do tabulek jsou připojeny objektové vlastnosti dat, přičemž některé položky mohou mít poněkud rozšířenou strukturu dat, která je nazývána **abstraktním datovým typem**.

Literatura:

- [1] Softwarové noviny 9/95 – Cover Story - Databáze
- [2] Vaníček, T.: FoxPro 2.5 for DOS snadno a rychle, Grada, Praha 1993
- [3] Kořínek, M.: Access 2000, Kopp, České Budějovice 2000
- [4] různé materiály z internetu