

Řízení databázových systémů

Přednáška 1

Roman Špánek

- Ing. Roman Špánek, Ph.D.
- Kancelář: budova A, 3. patro
- Tel. (48 535) 3519
- email: roman.spanek@tul.cz
- Přednášky na:
 - <https://elearning.fm.tul.cz/>

Náplň předmětu RDB

- Úvod, relační model dat, normální formy
- Návrh databáze a normální formy
- Indexování a optimalizace
- Uložené procedury, spouště
- NoSQL
- Programování databázových aplikací
- ...

Miniprojekty

- Cílem předmětu je osvojit si a umět použít nabyté znalosti ohledně databází
- Naučit se komunikovat s případným zadavatel (zaměstnavatelem)
- Podmínkou zápočtu bude vytvořit DB aplikaci a také ji prezentovat před publikem (v rámci cvičení či přednášek)

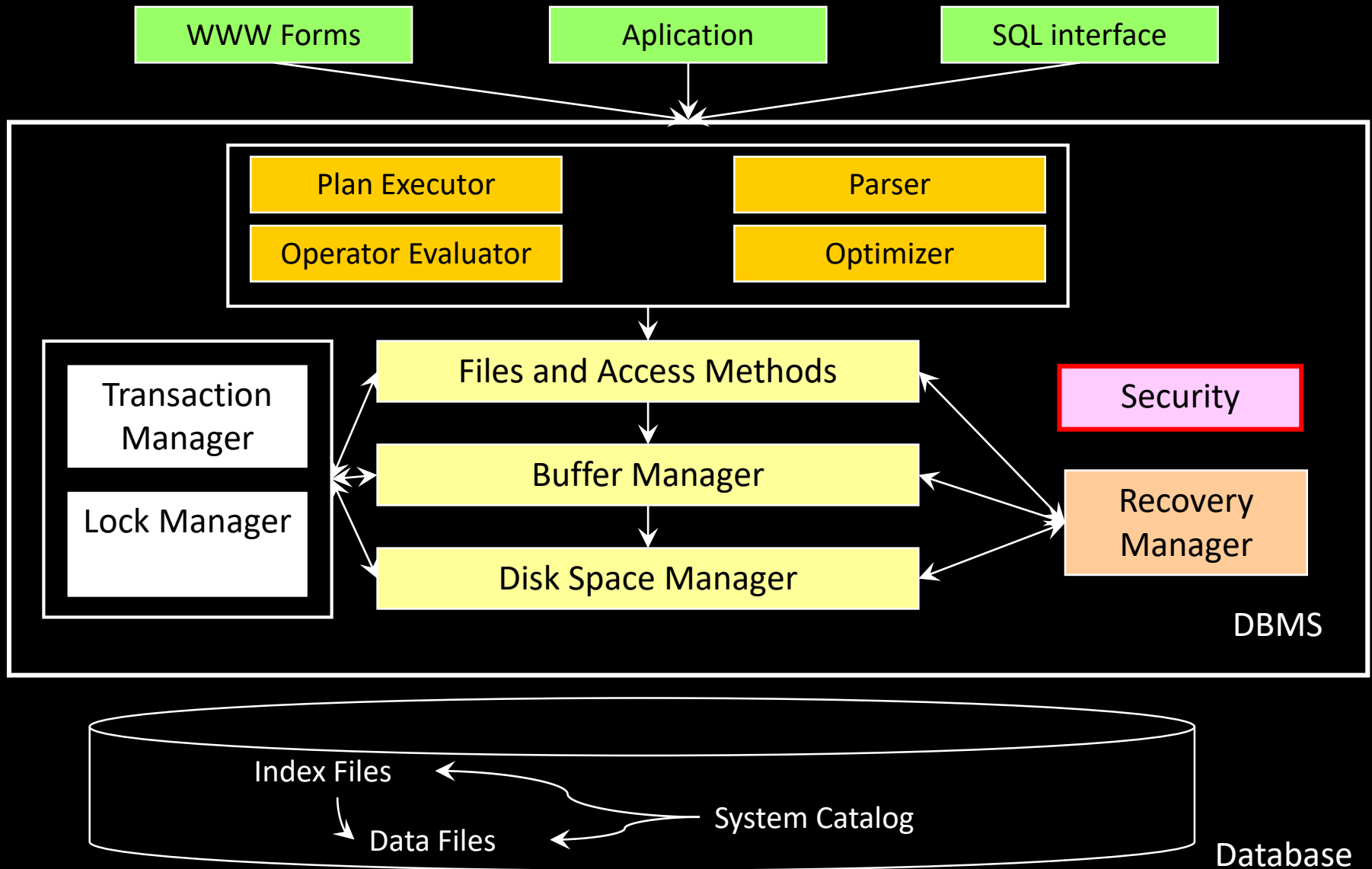
Miniprojekty

1. Sami si vytvoříte skupinky po max. 3 studentech
2. Zadání bude oznámeno během prvních cca 3 týdnů semestru
3. Na konci (cca poslední dva týdny) by jste odevzdávali/prezentovali vaše práce

Historický vývoj zpracování dat

- Systémy pro zpracování souborů
- Nevýhody:
 - redundance dat,
 - nekonzistentnost dat (problém aktualizace od různých uživatelů).
- Přístupy jsou sice ve vyšších programovacích jazycích, ale tzv. natvrdo.
- Při rozvoji systému to vede k neustálému předělávání a dodělávání aplikačních programů.
- Soubory mohou mít různé formáty, je obtížné psát moduly pro řízení přístupu k datům do již hotových programů.
- Problém se zabezpečením. Soubory jsou obvykle dostupné v rámci OS.

Struktura SŘBD



Relační model dat

- 1970 E.F. Codd
- Založen na **pevném matematickém** základu
- Def: Matematická relace
 - Relace R^* nad množinou atributů
 $\Omega = \{A_1:\text{dom}(A_1), \dots, A_n:\text{dom}(A_n)\}$
je podmnožina kartézského součinu
 $\text{dom}(\Omega) = \text{dom}(A_1) \times \dots \times \text{dom}(A_n)$,
kde pro každé A_i je $\text{dom}(A_i)$ neprázdným oborem (doménou)
hodnot, dále nedělitelných.

Relační model dat

- Základní rysy relačního modelu:
 - RMD **důsledně odděluje data**, která jsou chápána jako relace, **od jejich implementace**
 - přístup k datům je **symetrický**, tj. při manipulaci s daty se nezajímáme o přístupové mechanismy k datům
 - pro manipulaci s daty jsou k dispozici dva silné prostředky - **relační kalkul a relační algebra**

- RMD má jediný konstrukt: **databázovou relaci**
 - Mějme množiny $D_1, D_2, D_3, \dots, D_n$. Z každé vybereme 1 prvek. Tím vytvoříme uspořádanou n -tici.
 - Kartézský součin $D_1 \times D_2 \dots$ je množina všech posloupností (x_1, x_2, \dots) kde x_1 je prvkem $D_1 \dots$
 - Z hlediska databázových systémů jsou množiny D_1, D_2, \dots množinami hodnot atributů a značí se jako **domény**.

- **Schéma relace** $R(A_1:D_1, \dots, A_n:D_n)$, kde
 - **A_i** jsou jména atributů
 - **D_i** jsou domény atributů $\text{dom}(A_i)$ (obvykle primitivní typy (STRING, INTEGER...)).)
 - Dvojici **$A_i:D_i$** se říká **atribut relace**.
- Relace R nad množinou A je libovolná **podmnožina** kartézského součinu domén $D_1 \times \dots \times D_n$.
- Prvkům relace se říká **n -tice**, přičemž **n** určuje **řád relace**.

- **Relační schéma databáze** je dvojice (R, I) , kde
 - R je množina schémat relací,
 - I je množina integritních omezení.
 - Relační databáze se schématem (R, I) je **konzistentní**, pokud prvky všech relací relačního schématu vyhovují I .
- Protože relace jsou množiny, nesmí relace obsahovat duplicitní prvky.

Integritní omezení

- **Primární klíč** je množina atributů K z A , jejichž hodnoty jednoznačně určují n -tice relace R .
- **Kandidáti primárního klíče**: primární klíč - pouze jeden, alternativní klíče - více
- K musí být minimální (nelze z ní odebrat žádný atribut, aniž by to narušilo jednoznačnost identifikace)
- Atribut, který je součástí nějakého klíče se nazývá **klíčový**.
- Atributy, které nejsou součástí žádného klíče se nazývají **neklíčové**.
- Protože relace jsou množiny, nesmí relace obsahovat **duplicitní prvky**.
- **Referenční integrita**. Popisuje vztahy mezi daty ve dvou relacích. Atribut, kterého se referenční integrita týká se nazývá **cizí klíč** (foreign key).

Podmínky, které musí splňovat relační tabulka:

- sloupce mohou být v libovolném pořadí
- řádky mohou být v libovolném pořadí
- sloupce musí být homogenní = ve sloupci musí být údaje stejného typu
- každému sloupci musí být přiřazeno jednoznačné jméno (tzv. atribut)
- v relační tabulce nesmí být dva zcela stejné řádky (z čeho to plyne?).

- Dle relační teorie lze pomocí základních operací
 - sjednocení,
 - kartézský součin,
 - rozdíl,
 - selekce,
 - projekce,
 - přejmenování

uskutečnit veškeré operace s daty

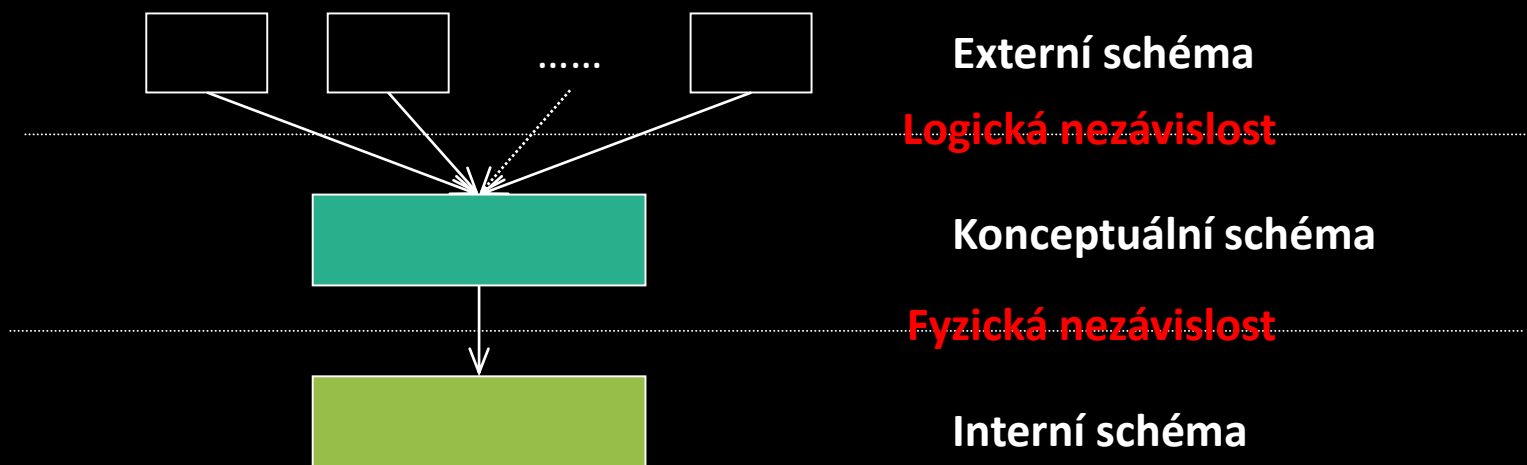
- ostatní operace jsou již jen kombinacemi těchto pěti.

Historie relačních databází

- 1975
 - ANSI SPARC
 - Do té doby pouze **schéma a subschéma**
 - Schéma: celý obsah databáze
 - Subschéma: pohledy na data, např. pouze výseky
 - Návrh **3-úrovňové architektury** databázových systémů
 - Přibylo **konceptuální schéma**
 - Konceptuální modelování

Databázový systém

- DBS=SŘBD+DB
- Architektura SŘBD



- První SQL databázi na trhu byl v roce 1980 Oracle. Firmu Oracle založil Larry Ellison a jeho počáteční inspirace byla v System-R od IBM.
- Posléze po určitých pochybnostech managementu přišla na trh i IBM se svým produktem DB2.
- ERP (Enterprise Resource Planning) a MRP (Management Resource Planning)
 - Aplikační vrstva nad DBMS (Bean, Oracle, SAP,...)
 - Běžné úkoly plánování, lidské zdroje, sklady, finanční analýza, Business Intelligence,...

Historie

- Deduktivní databázové systémy
- 90. léta:
 - Objektově orientované databázové systémy
 - post-relační databázové systémy
 - multimediální databáze
 - Spatial Databases (GIS)
 - databáze na internetu + skriptovací jazyky (PHP, APS, .NET, JSP,...)

- Multimediální DB
 - text, grafika, číslíkově zpracovaný obraz a zvuk
 - Informační systémy úřadů (Office Information Systems)
 - dokumenty různých typů, vzájemná provázanost objektů, toky dokumentů, plánovací kalendáře apod.
- XML (Extendable Markup Language)
 - Standard pro výměnu strukturovaných dat v podobě tzv. XML
 - XML databáze
 - efektivní ukládání dokumentů a vyhledávání

Objektové modely dat

- Hierarchický, síťový i relační model jsou záznamově orientované modely, které popisují prvky databáze formou jednotlivých záznamů.
- V 90. letech se začaly objevovat první objektově orientované DBMS (**OODBMS**), které umožňují pracovat s datovou abstrakcí na úrovni objektů a tak **přirozeněji a věrněji popisovat skutečný svět**.
- Hlavní výhodou tohoto přístupu je **snadnější aktualizace dat**. Pro objektově orientovaný model (stejně jako pro dále zmíněný objektově relační model **ORDBMS**) neexistuje žádný standard.

Objektové modely dat

- Vývoj a návrh objektově orientovaných modelů v jejich univerzálnosti a komplexnosti je velmi složitý proces, což se projevuje zatím menším uplatněním tohoto typu modelu v reálných aplikacích.

Objektově-orientovaný datový model

- V roce 1991 vznikla skupina ODMG (Object Database Management Group), jejíž cílem byla snaha o standardizaci v oblasti OO databázových jazyků (<http://www.odmg.org/>).
- GemStone, Jasmine, O2, ODE, Objectivity, ObjectStore

Objektově relační SŘBD

- Komplikovaný vývoj OOSŘBD vedl ke vzniku **objektově relačních SŘBD (ORSŘBD)**, které spojují pozitivní vlastnosti relačních SŘBD
 - výkonný jazyk SQL,
 - jednoduchá relační implementace složitých dat a objektově orientovaných SŘBD,
 - vazba na reálné objekty,
 - nové typy dat,
 - zapouzdření metod a objektů,
 - dědičnost vlastností tříd,
 - připojování procedur k záznamům

Objektově-relační databáze

- podpora nenormalizovaných (které nejsou v 1NF) relací, rozšíření relačního modelu o bohatší typový systém a OO rysy
- vnořené relace (nested relations)
- domény atributů mohou obsahovat buď atomické (skalární) nebo relační (zanořené relace) hodnoty
 - Titul (nazev, *seznam_autoru*, vydavatel, rok_vydani, *klicova_slova*)
- Složité datové typy
 - kolekce
 - množiny, multimnožiny, pole

NoSQL

- NoSQL je nerelační systém řízení báze dat
- Navržen pro distribuovaná datová uložště (např. Google, Facebook).
- Nevyžadují pevné schéma databáze
- Nepoužívá se JOIN operace
- Škálují horizontálně

Proč NoSQL

- Dostupnost dat je dnes mnohem větší než byla v minulosti
- Osobní data uživatelů, sociální vazby (grafy), geografická data, logy systémů jsou příklady kdy množství dat roste exponenciálně.
- SQL databáze nebyly navrženy na zacházení s tak obrovským množstvím dat

SŘBD vs. NoSQL

- **SŘBD**

- Strukturovaná a organizovaná data
- SQL
- Data a jejich vztahy jsou uloženy v oddělených tabulkách
- DML a DDL
- Konzistence
- Transakce

- **NoSQL**

- Nemá deklarativní dotazovací jazyk
- Nemá definované schéma
- Ukládá dvojice klíč-hodnota
- Uložení sloupců, dokumentů, grafů
- Případná konzistence upřednostněna před ACID vlastnostmi
- Podpora nestrukturovaných a nepředvídatelných dat
- CAP teorém
- Upřednostňují vysoký výkon, dostupnost a rozšiřitelnost

Jaký DB model?

- Relační
- Objektový
- NoSQL
- Jak se rozhodovat?
 - Cena
 - Výkon
 - Škálovatelnost
 - ...