



# BKM\_DATS: Databázové systémy

## 1. Úvod

Vlastislav Dohnal

# Databázové systémy

- Účel
- Pohled na data
- Databázové jazyky
- Architektura databázových systémů
  - Datové modely
  - Struktura databázového systému
  - Relační databáze
  - Objektové databáze
  - Historie
- Návrh databáze
- Ukládání dat a dotazování
- Zpracování transakcí
- Uživatelé a správci databáze

# Databázový systém

- Systém řízení báze dat (SŘBD)
  - Database Management System (DBMS)
- DBMS zpracovávají informace konkrétní firmy
  - Kolekce vzájemně propojených dat
  - Programy přistupující k datům
  - Prostředí (rozhraní) pro používání
    - má být použitelné, ale i efektivní
- Databázové aplikace
  - Bankovníctví – všechny transakce, co známe
  - Aerolinky – rezervace, plánování
  - Univerzity – registrace, zápis, hodnocení, ...
  - Obchod – zákazníci, produkty, prodeje
    - Online – sledování zásilek, doporučení podle uživatelů (customized recommendations)
  - Výroba – produkce, inventář, objednávky, doprava, dodavatelé
  - Státní správa – evidence obyvatel, žádosti, daňová správa, ...
- S databázemi se lze setkat (téměř) všude
  - Servisní orientace (SOA), microservices => databáze vložené v aplikaci

# Účel databázového systému

- Dříve byly databáze vytvořeny přímo nad souborovým systémem.
- Nevýhody ukládání dat přímo v souborech:
  - Redundance a nekonzistence dat
    - Různé souborové formáty, duplikace informace do více souborů
  - Data jsou obtížně přístupná
    - Pro novou úlohu naprogramovat novou aplikaci.
  - Izolovanost dat
    - Oddělené soubory, různé formáty
  - Problémy s udržením integrity
    - Integritní omezení jsou implementované v uživatelských programech, např. zůstatek účtu  $\geq 0$ .
    - Jsou skryté v programech, nikde nejsou „explicitně prezentovány“
    - Obtížné přidat nové omezení nebo změnit stávající

# Účel databázového systému

- Nevýhody ukládání dat přímo v souborech:
  - Atomičnost aktualizací dat
    - Výpadky mohou způsobit nekonzistentní stav
      - Pouze některé úkoly byly provedeny
    - Např. převod částky z účtu na účet – musí být provedeno komplet nebo vůbec
  - Souběžný přístup více uživatelů
    - Důležité pro výkon systému
    - Bez kontroly souběžného přístupu lze vytvořit nekonzistence
      - Např. dva uživatelé přistupují a aktualizují zůstatek téhož účtu
  - Omezení přístupu k datům (data security)
    - Obtížné omezit přístup k vybraným datům (část některého souboru)
- Databázové systémy
  - = řešení těchto problémů

# Datové modely

- Datový model = sada nástrojů pro popis dat
  - a vztahů mezi nimi
  - sémantiky dat
  - integritních omezenía manipulaci s daty
  
- Příklady
  - Relační model
  - Entitně-relační model (zejména při návrhu databáze)
  - Objektově-relační model, objektově-orientovaný model
  - Model pro semistrukturovaná data (XML)
  - Starší modely
    - Hierarchický model
    - Síťový model

# Relační model

## ■ Příklad dat v relačním modelu

### □ Tabulková reprezentace

<i>customer_id</i>	<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>	<i>account_number</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-101
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-201
677-89-9011	Hayes	3 Main St.	Harrison	A-102
182-73-6091	Turner	123 Putnam St.	Stamford	A-305
321-12-3123	Jones	100 Main St.	Harrison	A-217
336-66-9999	Lindsay	175 Park Ave.	Pittsfield	A-222
019-28-3746	Smith	72 North St.	Rye	A-201

# Příklad relační databáze

<i>customer_id</i>	<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>account_number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

<i>customer_id</i>	<i>account_number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table



# Jazyk pro definici dat

- Data Definition Language (DDL)
- Poskytuje výrazy pro definici schéma databáze
  - Např.: 

```
create table account (  
    account-number char(10),  
    balance integer  
);
```
- Kompilátor DDL vygeneruje množinu tabulek
  - Ty jsou uloženy v datovém slovníku
- *Datový slovník* obsahuje metadata (data o datech)
  - Databázové schéma
  - Integritní omezení
    - Omezení domény
    - Referenční integrita (cizí klíč)
    - Tvrzení (Assertions)
  - Přístupová práva
  - Metody uložení dat

# Jazyk pro manipulaci s daty

- Data Manipulation Language (DML)
- Jazyk pro přístup a manipulaci s daty organizovanými v konkrétním modelu
  - Často nazývaný jako dotazovací jazyk (query language)
- Dvě třídy jazyků
  - Procedurální – uživatel specifikuje jak data, která chce, tak i, jak se k nim dostat.
  - Deklarativní (neprocedurální) – uživatel specifikuje pouze data bez postupu, jak je získat.
- SQL = nejčastější (dotazovací) jazyk

# SQL

- SQL – Structured Query Language
- Často používaný neprocedurální jazyk

- Např.: Zjisti jméno zákazníka s id 7465

```
select customer_name  
from customer  
where customer_id = 7465
```

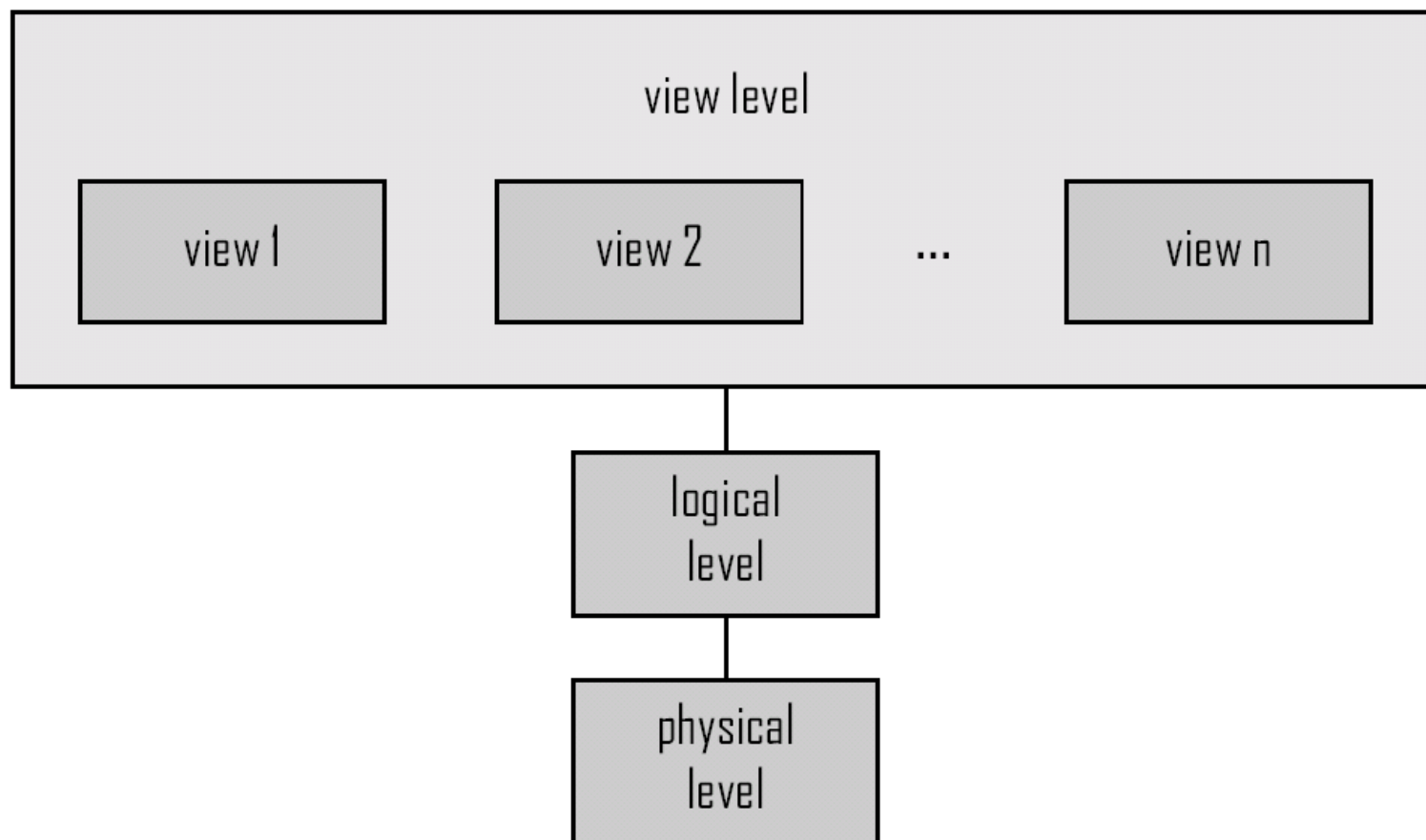
- Např. Vypiš zůstatky všech účtů vlastněných zákazníkem majícím id 7465

```
select account.balance  
from depositor, account  
where depositor.customer_id = 7465 and  
depositor.account_number = account.account_number
```

- Aplikace obecně přistupují k databázi pomocí
  - rozšíření programovacího jazyka o zapouzdření SQL
  - rozhraní (např. ODBC, JDBC) umožňující posílání SQL výrazů
  - knihoven automaticky generujících SQL dotazy (Spring.io & JPA)

# Pohled na data

- Architektura databázového systému



# Úrovně abstrakce dat

## ■ Fyzická úroveň

- Popisuje, jak je datový záznam (např. objednávka) uložen.

## ■ Logická úroveň

- Popisuje strukturu dat uložených v databázi a vztahy mezi daty.

**type order = record**

```
    order_id : integer;  
    created  : date;  
    goods   : string;  
    customer_id : integer;
```

**end;**

## ■ Úroveň pohledů

- Pohled může selektivně ukrývat části dat – řádky i sloupce
  - Např. výši platu (z bezpečnostních důvodů)
- Výpočet hodnot na základě uložených dat
  - Stáří zákazníka
- Kombinace více relací

# Úrovně abstrakce dat

- Fyzická datová nezávislost
  - = Schopnost změnit fyzické schéma bez změny logického schématu
- Aplikace jsou závislé na logickém schématu
- Nutné definovat rozhraní mezi jednotlivými úrovněmi
  - Změny v jedné úrovni co nejméně ovlivnily ostatní úrovně

# Instance a schéma databáze

## ■ Schéma databáze

- = Struktura databáze
  - Např. kolekce zákazníků, účtů a vztahů mezi nimi
- Logické schéma
  - Návrh databáze na logické úrovni
- Fyzické schéma
  - Návrh databáze na fyzické úrovni

## ■ Databáze, resp. *instance databáze*

- Aktuální datový obsah (v daném čase)

## ■ Databázový systém

- Implementace konkrétního dat. modelu včetně dalšího SW
- Např. MariaDB, PostgreSQL, ....

# Návrh databáze

## ■ Proces návrhu databáze:

### 1. Logický návrh

- Rozhodnutí, jak má databázové schéma vypadat
- Vyžaduje nalezení vhodných relačních schémat
- Rozhodnutí zákazníka
  - Jaké informace (popisy) budeme v databázi ukládat?
- Rozhodnutí IT
  - Jaké relace vytvoříme a jaké budou mít atributy

### 2. Fyzický návrh

- Rozhodnutí o fyzickém rozložení databáze



# Návrh databáze

## ■ Požadavky

- Chceme evidovat učitele univerzity a jejich příslušnost k fakultě
- Učitel má jméno a plat
- Fakulta má své sídlo a rozpočet

## ■ Příklad dat

- Einstein bere 95000 a patří na Physics ve Watson s rozpočtem 70000.

# Návrh databáze

## ■ Je tento návrh správný?

Tabulka *instructor*

<i>name</i>	<i>salary</i>	<i>dept_name</i>	<i>building</i>	<i>budget</i>
Einstein	95000	Physics	Watson	70000
Wu	90000	Finance	Painter	120000
El Said	60000	History	Painter	50000
Katz	75000	Comp. Sci.	Taylor	100000
Kim	80000	Elec. Eng.	Taylor	85000
Crick	72000	Biology	Watson	90000
Srinivasan	65000	Comp. Sci.	Taylor	100000
Califieri	62000	History	Painter	50000
Brandt	92000	Comp. Sci	Taylor	100000
Mozart	40000	Music	Packard	80000
Gold	87000	Physics	Watson	70000
Singh	80000	Finance	Painter	120000

# Návrh databáze

Tabulka *instructor*

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Tabulka *department*

<i>dept_name</i>	<i>building</i>	<i>budget</i>
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

# Návrh databáze

## ■ Teorie o normalizaci

- Formální nástroje, které určují, co je správně a co ne.
- Postupy, jak udělat správný návrh

## ■ Entitně-relační model

- Modeluje data ISu jako kolekce *entit a vztahů*
  - Entita je „věc“ nebo „objekt“ evidovaná v IS, která je jednoznačně odlišitelná od ostatních
    - Popsána sadou atributů
  - Vztah je vazba mezi entitami
- Reprezentován pomocí E-R diagramu

# Entitně-relační model

## ■ Entitní množiny

- *instructor* a *department*

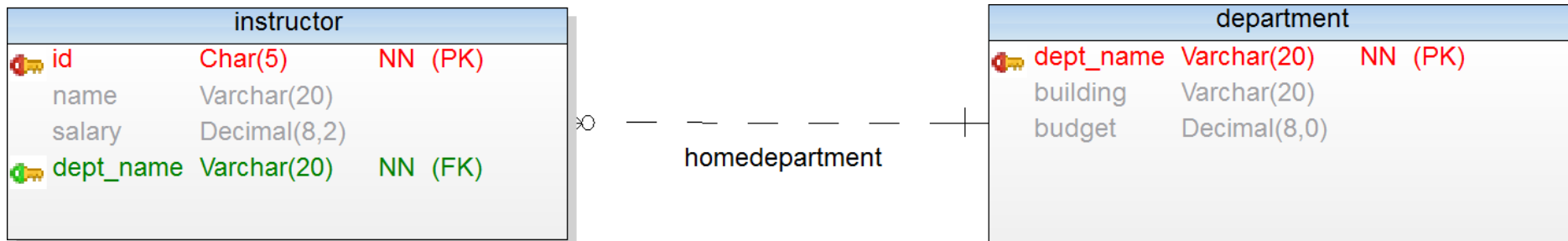
- vyznačené primární a cizí klíče

## ■ Vztahy, např. *homedepartment*

- kardinalita n:1 (many-to-one)

- totální (ze strany *instructor*)

- označeno kolmou čarou u *department*



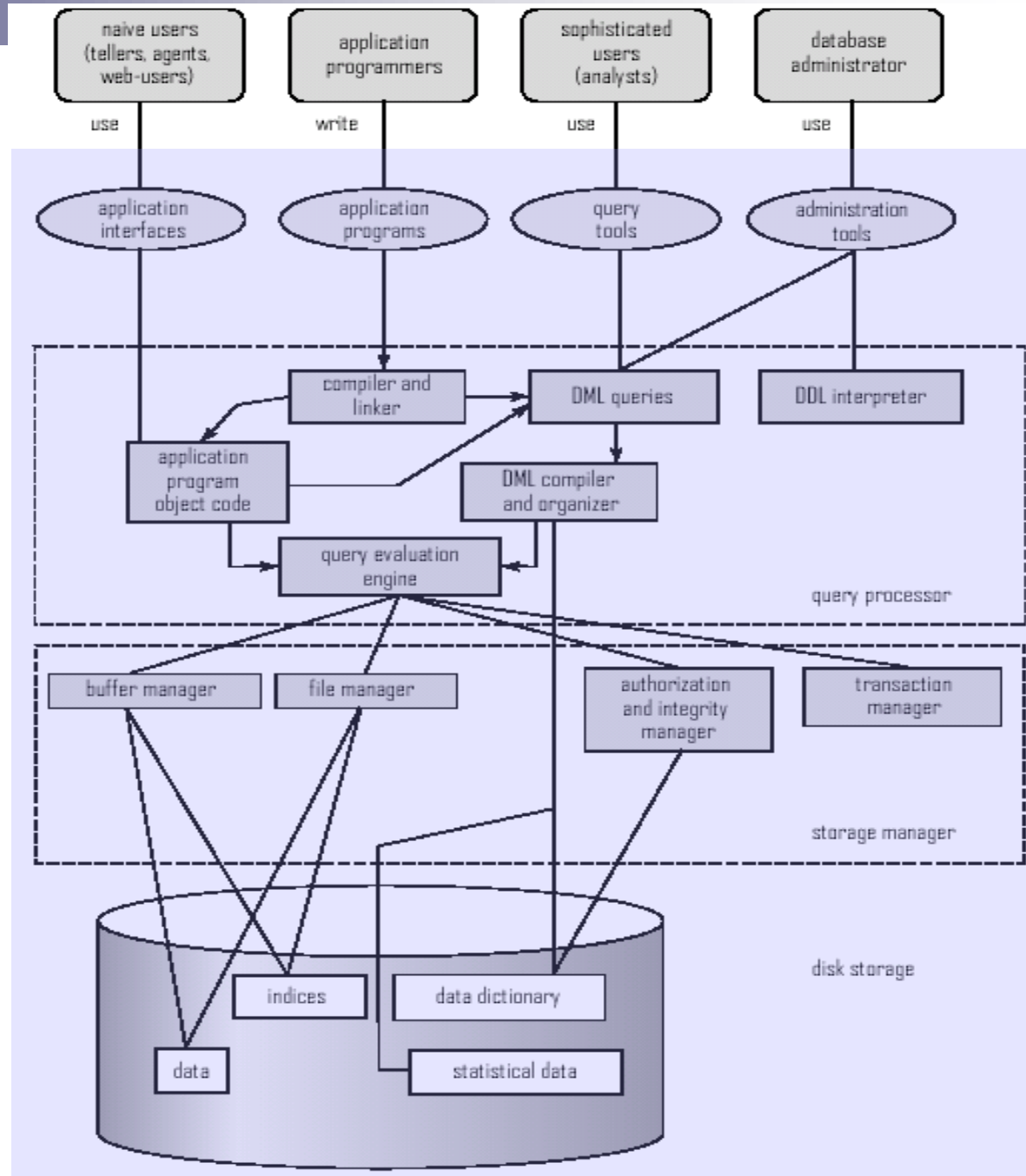
# Objektově-relační model dat

- Rozšiřuje relační model
  - o objekty a konstrukce pro práci s nimi
- Umožňuje, aby atributy měly komplexní struktury
  - Např. vnořené relace
- Zachovává relační přístup k datům
  - Rozšiřuje jej
  - Je zpětně kompatibilní s existujícími relačními jazyky
    - `SELECT object.name FROM table WHERE object.isValid();`

# XML: Extensible Markup Language

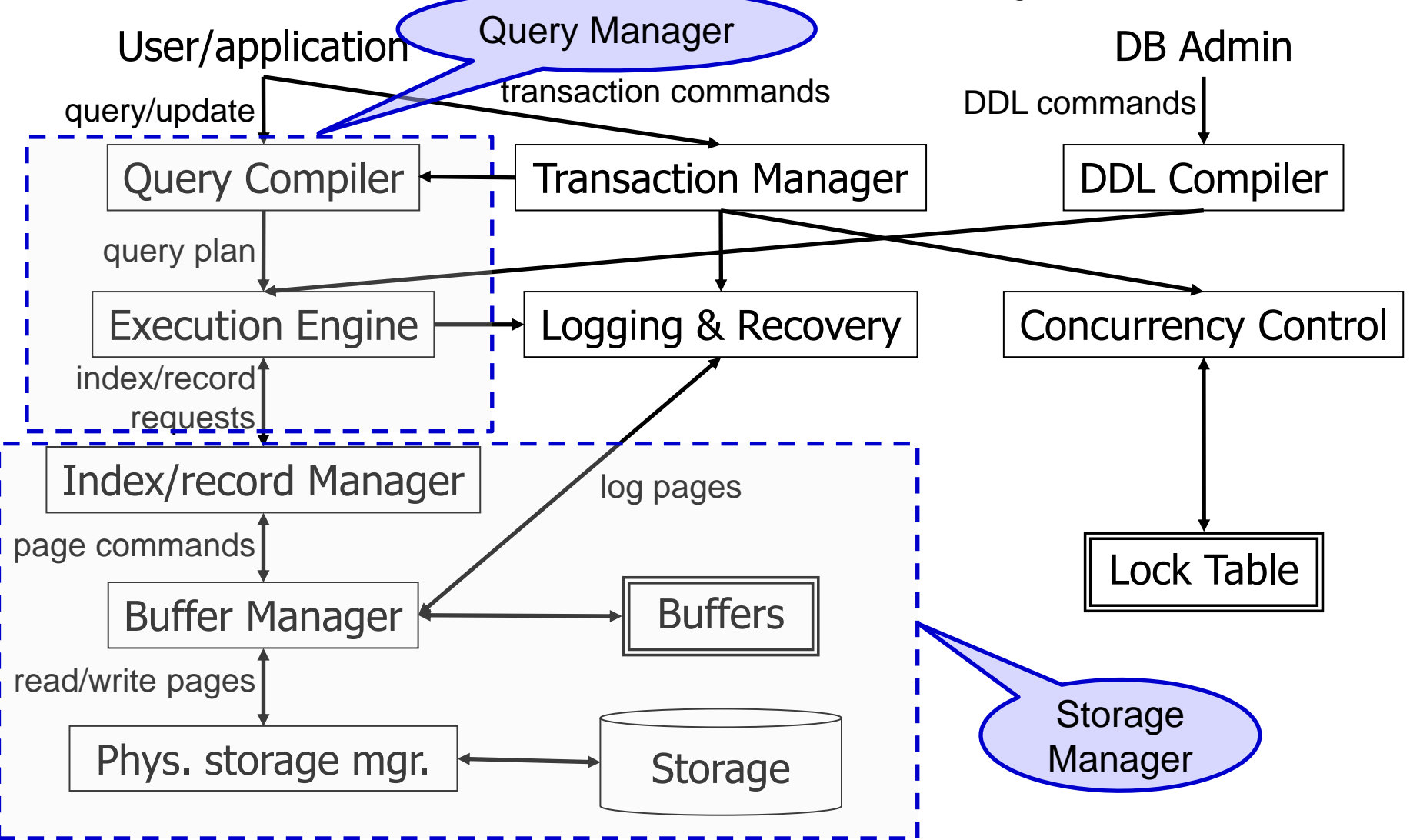
- Definovaný W3C konzorciem
- Původně jako značkovací jazyk pro dokumenty
  - Nikoli jako databázový jazyk
- Lze definovat nové značky
- Značky se mohou vnořovat
  - Velmi vhodné pro výměnu dat
- XML je základy moderních výměnných formátů
- Nástroje pro zpracování a dotazování XML

# Struktura databázového systému





# Struktura databázového systému



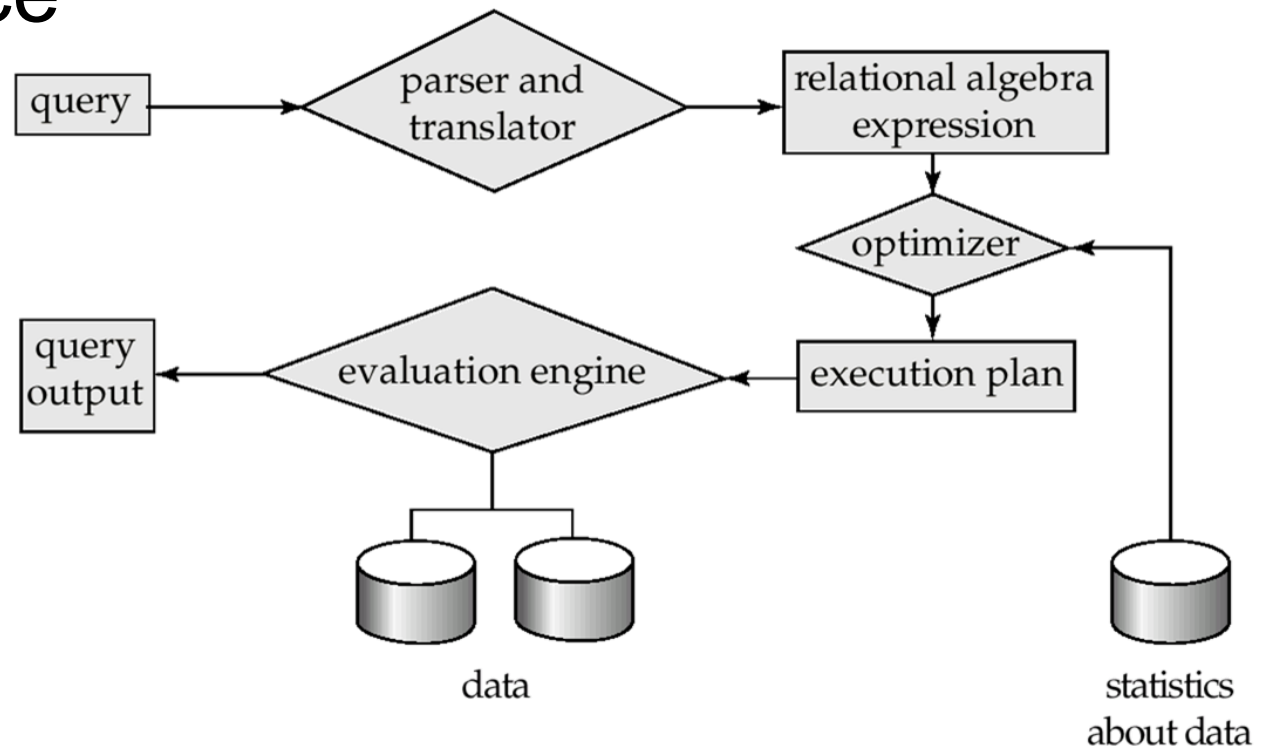
# Správa úložiště

## ■ Správce úložiště (storage manager)

- Modul, který poskytuje rozhraní pro fyzické ukládání dat
- Úkolem je
  - Spolupráce se souborovým systémem
  - Efektivní ukládání, aktualizace a opětovné načítání dat
- Spravuje
  - Přístup k úložišti
  - Organizuje data, např. do souborů
  - Provádí indexování

# Zpracování dotazu

- Analýza a překlad
  - Parsing and translation
- Optimalizace
- Vykonání
  - Execution



# Optimalizace dotazu

- Více možností zpracování stejného dotazu
  - Ekvivalence výrazů
  - Různé algoritmy pro jednotlivé operace
- Rozdílné náklady jednotlivých možností
  - Cena zpracování, obvykle čas
  - Rozdíly mohou být velmi významné
- Potřeba tyto náklady odhadovat
  - Používání statistik o datech (velikost relací, ...)
  - Statistiky o mezivýsledcích dotazů
    - Pro odhady složitých dotazů

# Zpracování transakcí

## ■ Transakce

- Posloupnost operací, které tvoří logický blok, funkci, databázové aplikace

## ■ Transakční zpracování (správce transakcí)

- Zajištění konzistentního (správného) stavu
  - bez ohledu na výpadky databázového systému
  - např. výpadek proudu, pád OS, chyba paměti.

## ■ Souběžné zpracování

- Zajišťuje konzistenci dat, pokud je spuštěno více transakcí současně
  - Tj. zajišťuje izolovanost transakcí

# Databázoví uživatelé

- Typy uživatelů podle jejich činností
  - Aplikační programátor
    - Používá DML
  - Znalí uživatelé
    - Používají přímo dotazovací jazyk databáze
  - Specializovaní uživatelé
    - Vytvářejí aplikace, kterým tradiční zpracování dat nepostačuje
  - Naivní uživatelé
    - Používají připravené aplikační programy, rozhraní
      - Např. webové stránky

# Databázový správce

- Koordinuje činnosti prováděné s databázovým systémem
  - Má znalosti o dostupných zdrojích a potřebách firmy
- Činnosti:
  - Definice schéma databáze
  - Definice metod ukládání a přístupu k datům
  - Změny schématu a fyzické organizace
  - Povolování přístupových práv, vytváření uživatelů
  - Definice integritních omezení
  - Prostředník pro komunikaci mezi uživateli
  - Sledování výkonnosti a ladění systému

# Databázové architektury

- Architektura databázového systému je ovlivněna prostředím, ve kterém má DB systém běžet.
  - Centralizovaná (vnitřní součást aplikace)
  - Klient-server
  
  - Paralelní (multiprocesory)
  - Distribuované



# Historie databázových systémů

- 50tá a počátek 60tých let
  - Zpracování dat na magnetických páskách
    - Pouze sekvenční přístup
  - Děrné štítky pro vstup dat
- 60tá a 70tá léta
  - Pevný disk a přímý přístup k datům
  - Síťový a hierarchický model dat
  - Ted Codd definuje relační model
    - IBM Research začíná implementovat System R (dnes jako DB2)
    - UC Berkeley – prototyp systému Ingres
  - Vysoce výkonné transakční zpracování
    - Na svoji dobu

# Historie databázových systémů

## ■ 80tá léta

- Výzkum relačních systémů vedl ke komerčním systémům
  - SQL se stalo průmyslovým standardem
- Paralelní a distribuované databáze
- Objektově-orientované databáze

## ■ 90tá léta

- Podpora pro rozhodovací procesy (decision support) a dolování znalostí
- Obrovské datové sklady (terabajty)
- Počátek internetového (webového) obchodování

## ■ 2000-dnes

- Standardy pro XML a Xquery, JSON
- Automatická administrace a ladění databáze
- Specializované databázové systémy:
  - Obrovské datové úložiště, distribuované, NoSQL databáze
    - Google BigTable, Yahoo PNuts, Amazon Redshift, ...

# Praktická cvičení

## ■ PostgreSQL

- open-source, dostupné pro Linux, macOS, Windows



## ■ Řádkové rozhraní

- `psql -h <host> -U <user> -W <db_name>`

## ■ Grafické rozhraní

- pgAdmin

- pro Linux, macOS, Windows

The screenshot shows the pgAdmin interface. On the left is a tree view of the database structure, including 'Local Servers (7)', 'Databases (2)', and 'public' schema. The main window displays a 'Query Editor' with the following SQL query:

```
1 SELECT * FROM public.statement
2 ORDER BY id ASC
```

Below the query editor is a 'Data Output' table with the following columns: id, text, search\_text, conversation\_text, and created\_at. The table contains 20 rows of data.

id	text	search_text	conversation_text	created_at
1	What's the name of that package fo...	DT:language_uni...	training	2008-10-10 05:02:00+01
2	Hello!	hello	training	2008-10-10 05:03:00+01
3	what's the name of the compiz man...	DT:language_uni...	training	2008-10-10 05:04:00+01
4	lcompiz	lcompiz	training	2008-10-10 05:05:00+01
5	you looking for it or just want the na...	PRP:looking RB...	training	2008-10-10 05:05:00+01
6	the name so i can install it	DT:language_uni...	training	2008-10-10 05:05:00+01
7	hmm im pretty sure its under addr...	NN:im NN:pretty...	training	2008-10-10 05:06:00+01
8	You can install Compiz by using the ...	MD:put VB:com...	training	2008-10-10 05:08:00+01
9	are you still here?	PRP:still	training	2008-10-10 05:10:00+01
10	acke. did u try installing flash using...	JJ:activity NN:p...	training	2007-07-18 14:10:00+01
11	yeah??	NN?	training	2007-07-18 14:10:00+01
12	lol	lol	training	2007-07-18 14:10:00+01
13	sorry...noob	NN:noob	training	2007-07-18 14:11:00+01
14	here	here	training	2007-07-18 14:11:00+01
15	lol	lol	training	2007-07-18 14:11:00+01
16	i try not to use it as much as possibl...	NNS:act TO:use...	training	2007-07-18 14:11:00+01
17	apt-get 99% of the time works though	JJ:99% DT:case...	training	2007-07-18 14:11:00+01
18	automatix seems to not support pp...	RB:be RB:suppo...	training	2007-07-18 14:13:00+01
19	on windows, i can make my printer...	IN:windows, MD...	training	2010-05-25 01:49:00+01
20	Manual feeder does not work for me	JJ:domestic_ani...	training	2010-05-25 01:49:00+01