

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

**VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA STROJNÍ**



ZÁKLADY INFORMATIKY

Ing. Roman Danel, Ph.D.

Ostrava 2013

© Ing. Roman Danel, Ph.D.

© Vysoká škola báňská – Technická univerzita Ostrava

ISBN 978-80-248-3052-0



Tento studijní materiál vznikl za finanční podpory Evropského sociálního fondu (ESF) a rozpočtu České republiky v rámci řešení projektu: CZ.1.07/2.2.00/15.0463, MODERNIZACE VÝUKOVÝCH MATERIÁLŮ A DIDAKTICKÝCH METOD

OBSAH

1	DATABÁZE A DATABÁZOVÉ SYSTÉMY	3
1.1	Úvod	4
1.2	Typy databází	4
1.3	Relační databáze	5
1.3.1	Transakce	6
1.3.2	SQL	7
1.3.3	Některé databázové objekty	7
1.3.4	Databázový stroj (Systém řízení báze dat)	7
1.3.5	Hlavní databázové systémy	8
1.4	Ostatní typy databází - Flat file	8
1.5	Datové sklady	9
1.6	Hierarchická db	9
1.7	Objektově orientovaný DB model	9
1.8	NoSql databáze	9
1.9	Kontrolní otázky	10



1 DATABÁZE A DATABÁZOVÉ SYSTÉMY



OBSAH KAPITOLY:

Typy databází

Relační databáze

Ostatní typy databází - Flat file

Datové sklady

Hierarchická db

Objektově orientovaný DB model

NoSql databáze



MOTIVACE:

V této kapitole se seznámíte s databázovými systémy. Databáze slouží k ukládání dat a tvoří základní kámen při tvorbě informačních systémů. Naučíte se, jaké existují typy databází podle modelu ukládání dat. Dočtete se, jaké jsou základní vlastnosti nejpoužívanější architektury databází v současnosti – relačního datového modelu. Naučíte se také, k čemu je dotazovací jazyk SQL. V závěru kapitoly se zmíníme o objektově orientovaném modelu ukládání dat a o bezeschémových databázích.



CÍL:

Po prostudování kapitoly budete znát základní strukturu a vlastnosti relačního datového modelu. Budete vědět, co to je transakce a co pro transakci znamená vlastnost ACID.



1.1 ÚVOD

Databáze je soubor dat, která slouží pro popis reálného světa.

Za formu databáze z dřívějších dob můžeme považovat i papírovou kartotéku. S nástupem počítačů se začaly používat agendové systémy. Požadavky na větší pružnost zpracování vedly k vytvoření databází s hierarchickou nebo síťovou strukturou.

Dnes je drtivá většina databází vytvořena na základě relačního modelu, jehož autorem je E. F. Codd. Tyto databáze se označují jako relační. Okrajově se můžete setkat také s databázemi, které fungují na objektově orientovaném principu (např. databáze Caché).

Dalším standardem, se kterým se v oblasti databází můžeme setkat, je dotazovací jazyk SQL (Structure Query Language). SQL je deklarativní jazyk, který se používá k práci s objekty a daty uloženými v databázi. Deklarativní jazyk na rozdíl od procedurálních jazyků funguje tak, že počítači deklarujeme, co chceme udělat a ne jak to má dělat. Provádění SQL příkazů a jejich optimalizace je tak vnitřní záležitostí interpretera příkazů SQL. Jednotlivé příkazy SQL vycházejí z angličtiny a původní ideou jazyka bylo umožnit uživatelům – neprogramátorům provádět sofistikované operace nad daty.

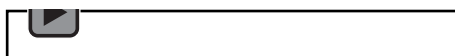
Z toho co jsme si nyní řekli, plyne jeden závěr. Ne každá databáze je relační a ne každá relační databáze má nativní podporu SQL jazyka (příkladem takové databáze je třeba dBase v éře MS DOS a její datový formát DBF, tato relační databáze je primárně řádkově orientovaná).

Pokud chceme používat SQL jazyk, musíme si především uvědomit, že SQL pracuje s daty jako s množinami. Každá SQL operace, podle definovaných podmínek, tak ovlivní určitou množinu dat, a tato množina má nula až n prvků (tzn. SQL příkaz může proběhnout úspěšně, přičemž ovlivnil 0 řádků v databázi, protože žádný řádek nevyhověl zadané podmínce).

Data jsou v databázi uložena v tabulkách, což jsou objekty, které mají dvojrozměrnou strukturu – pevně definované sloupce (jsou určeny při vytvoření tabulky) a proměnný počet řádků. To je hlavní rozdíl oproti zpracování dat v tabulkovém procesoru, jakým je například Excel. Excel umožňuje uživateli pracovat s maticí dat, která má proměnný počet sloupců i řádků.



Audio 1.1



1.2 TYPY DATABÁZÍ

- Flat-file – textový soubor, sekvenční přístup, položky odděleny např. středníkem
- Fixed File – soubor s pevnou délkou jednotlivých položek
- Hierarchické – stromová struktura
- Síťová – rozšíření hierarchické struktury
- Relační
- Objektové (nebo kombinace objektově-relační)
- Deduktivní
- NoSql – document stored, key stored



1.3 RELAČNÍ DATABÁZE

Relační databáze jsou nejčastěji používaným modelem pro ukládání dat.

Na začátku máme nějaký objekt reálného světa nebo nějakou funkcionalitu, o nichž potřebujeme mít uložená data. Musíme tedy provést analýzu, jaké data jsou pro nás potřebná, pro další zpracování a na základě tohoto rozboru můžeme navrhnout strukturu databáze. Strukturou máme na mysli jednotlivé tabulky, které mají definované sloupce (s určenými datovými typy) a jsou definovány vazby (vztahy) mezi tabulkami. Pokud je databáze složitější - obsahuje spoustu provázaných tabulek - je vhodné pro návrh databáze použít nějaký modelovací nástroj. Obecný návod, jak vytvořit databázi tak, aby byla v každém případě optimální, neexistuje, pro dobrý návrh je třeba mít zkušenosti. Výchozí předpoklad je, že jedna tabulka obsahuje informace o jednom objektu reality. V teorii relačního modelu se tomuto objektu říká **entita**. Objekt (entita) má své vlastnosti, označované jako **atributy**. Jako příklad můžeme uvést objekt „telefonní seznam“, atributem bude jméno a příjmení a telefonní číslo.

Modelovacích nástrojů je mnoho. Společným prvkem těchto nástrojů je snaha o grafickou prezentaci struktury databáze v podobě diagramů. Nejčastěji se používají entitně relační diagramy, které umožňují přehledně znázornit strukturu tabulek a vazby mezi nimi. Pokročilejší nástroje pak umí z takto vytvořených diagramů vytvořit i fyzickou strukturu databáze v konkrétním databázovém produktu.

Teorie relačního modelu tedy zavádí následující prvky:

- **Entita** – objekt, který je předmětem zájmu, abstrakce reality (existujících věcí)
- **Atribut** – elementární datový prvek, který entitu blíže charakterizuje
- **Relace** – vztah mezi dvěma entitami
- **Kardinalita** vztahu – mocnost vztahu mezi entitami: 1:1, 1:N, N:1, M:N

Vlastnosti atributů:

- Musí být atomické
- Na pořadí sloupců nezáleží

Relace = vztah:

- Příkladem vztahu 1:1 je vztah manželství mezi muži a ženami v evropské části světa. Jeden muž může mít současně nejvýše jednu manželku a jedna žena nejvýše jednoho manžela.
- V arabském světě jde v uvedeném příkladě o vztah 1: N. Jeden muž může současně mít více než jednu ženu, žena může mít nejvýše jednoho manžela.
- Pokud bychom upustili v daném příkladě od požadavku současnosti, pak může jít o vztah M:N. Každý muž mohl být několikrát ženatý, stejně jako žena vícekrát vdaná.

Vztah je informace, kterou si systém musí pamatovat, nelze ji odvodit.

Ve fyzické realizaci databáze odpovídají entitám z relačního modelu tabulky. Tabulky, které „patří k sobě“ a jsou mezi nimi definovány relace (neb popisují určitou realitu) označujeme jako **schéma**.

Proces optimalizace relační databáze se nazývá **normalizace**. Normalizovaná databáze se nachází v tzv. normální formě, první až páté, podle stupně optimalizace. Pro praxi je postačující normalizace na úrovni třetí normální formy.



Jedním z cílů normalizace je odbourání tzv. **redundance**, tj. opakovaných výskytů stejného údaje v databázi (např. jméno firmy by v databázi mělo být pouze na jediném místě, v případě opravy jména se oprava provádí pouze na tomto jediném místě). U nenormalizované databáze s vícenásobnými výskyty údajů může dojít k porušení **integrity**.

Postup při tvorbě databáze:

- Analýza zadání
- Pojmenování logických jednotek (entit) a analýza, jaké jsou mezi nimi vztahy
- Návrh struktury (může a nemusí být ERD)
- Optimalizace – normální formy
- Stanovení pravidel integrity

Integrita:

- Entitní – jedinečná identifikace každé entity - pomocí primárního klíče (PK)
- Referenční – pomocí cizího klíče (FK) – integrita vazeb

Primární klíč – definuje se nad jedním nebo více sloupci tabulky. Hlavním účelem primárního klíče je zajistit jednoznačnost záznamů v tabulce.

Data primárního klíče nesmí obsahovat duplicitu nebo hodnotu Null.

Cizí klíč se definuje na sloupci tabulky, která je navázána na primární tabulku. Pomocí něho jsou realizovány vztahy v relační databázi.

Cizí klíč se odkazuje na sloupec primární tabulky. Do sloupce s definovaným cizím klíčem nelze vložit hodnotu, která není vložena v nadřazeném sloupci primární tabulky.

Referenční integrita:

- Restrict** – z nadřazené nelze smazat, je-li odkaz
- Cascade** – po výmazu z nadřazené smaže i odkazy v podřazené
- Set null** – po výmazu v nadřazené dostanou odkazy v podřazené hodnotu NULL

1.3.1 Transakce

skupina příkazů, které převedou databázi z jednoho konzistentního stavu do druhého.

Databázové transakce musí splňovat tzv. vlastnosti **ACID**:

A - Atomicity – atomicita (nedělitelnost)

C - Consistency - konzistence (není porušeno integritní omezení)

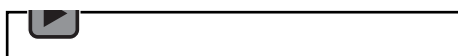
I - Isolation - izolovanost (ostatní nevidí, dokud není ukončena)

D - Durability - trvalost (změny, které se provedou potvrzenou transakcí, jsou v databázi trvalé a nemohou být ztraceny)

Transakce v databázových systémech je tedy **skupina databázových operací, která je provedena buď jako celek, nebo není provedena vůbec.**



Audio 1.2



1.3.2 SQL

SQL (**Structure Query Language**) je deklarativní dotazovací jazyk, který slouží k práci s databázovým systémem.

Obsahuje dvě skupiny příkazů:

- DDL – Data Definition Language
 - příkazy pro vytváření, modifikaci nebo rušení databázových objektů
 - SELECT, UPDATE, INSERT, DELETE
- DML – Data Manipulation Language
 - příkazy pro vytváření, modifikaci nebo rušení dat
 - CREATE, DROP, GRANT, ALTER

1.3.3 Některé databázové objekty

- **pohledy** neboli **views** – SQL příkazy, pojmenované a uložené v databázovém systému. Lze z nich vybírat (aplikovat na ně příkaz SELECT) jako na ostatní tabulky. View je tedy jakási virtuální tabulka.
- **indexy**. Indexy jsou definovány nad jednotlivými sloupci tabulek a jejich funkce je vést si v tabulkách rychlé LUT (*look-up tables* – „pořadníky“) na sloupce, nad nimiž byly definovány, vyloučit duplicitu v záznamech nebo zajišťovat fulltextové vyhledávání. Indexy zrychlují čtení z databáze, ale mohou zpomalovat zápis.
- **triggery** neboli **spouště** – SQL procedura, která je spjata s tabulkou a je automaticky spuštěna na základě definované události (delete, insert, update).
- **uživatelé definované procedury a funkce** – některé databázové stroje podporují ukládání pojmenovaných kusů kódu, které provedou v databázi nad danými tabulkami určitou sekvenci příkazů (procedury) nebo navíc vrátí nějaký výsledek (uživatelské funkce). Mohou mít parametry, které se většinou dělí na vstupní (*IN*), výstupní (*OUT*) a vstupně-výstupní (*INOUT*).
- **události** též „eventy“ – procedury, spouštěné v určitý (uživatelé definovaný) datum a čas nebo opakovaně s definovatelnou periodou nebo na základě určité události. Mohou sloužit k údržbě, promazávání dočasných dat či kontrolování referenční integrity
- **uživatelská oprávnění** – u databázových systémů je samozřejmostí nabídnout možnosti, jak oddělit jednotlivé úrovně přístupu k ostatním objektům databáze jejich uživatelům. Možností bývají desítky, s rozlišením na jednotlivé typy příkazů, které uživatel bude nebo nebude mít oprávnění spustit.
- **collation** – znakové sady a porovnávání, souhrnně nazývané *collation*. Nastavení collation může být provedeno na jednotlivé textové sloupce, celé tabulky i celé databáze (s kaskádovitou dědičností). Collation ovlivňuje i třídění dat, například hodnota *utf8_czech_ci* zajistí správné třídění podle češtiny (tedy včetně diakritiky a včetně písmene „ch“).

1.3.4 Databázový stroj (Systém řízení báze dat)

Software pro práci s databází, vytváří rozhraní mezi binární strukturou uložených dat a uživatelem databáze. Obsahuje:

- Optimalizátor dotazů
- Uživatelské rozhraní – grafický nástroj nebo příkazový klient



- Přístup na data přes rozhraní - např. ODBC, ADO, JDBC, DBI (Perl), DAO, ODMA
 - o ODBC – Open DataBase Connectivity
 - o ADO – ActiveX Data Object

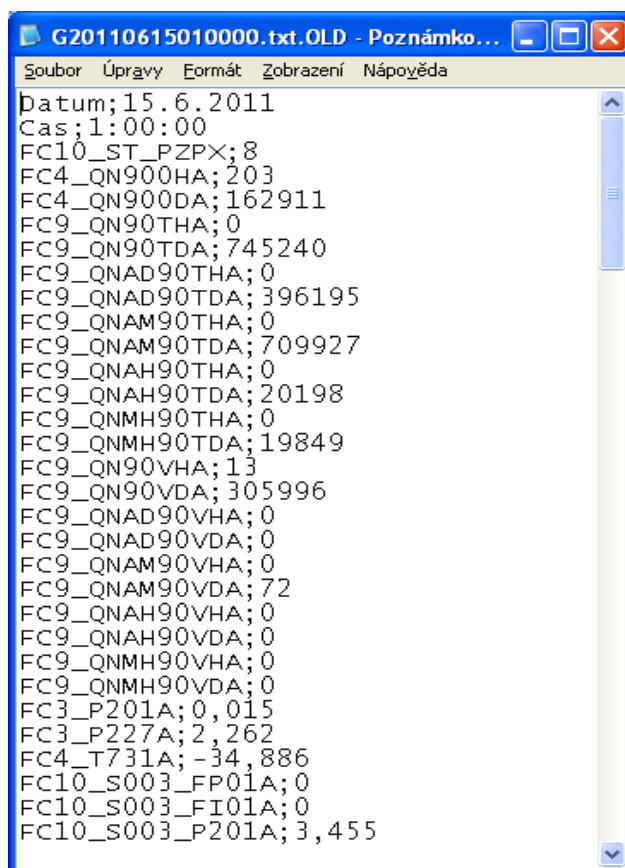
Typy uživatelů v databázi:

- Správce databáze (administrátor)
- Aplikační programátor
- Koncový uživatel

1.3.5 Hlavní databázové systémy

- Oracle – světově nejrozšířenější databázový systém, 40 % komerčního trhu
- Microsoft SQL Server
- Sybase
- MySQL (MariaDB)
- Postgre
- Informix (IBM)
- DB-2 (IBM)
- Interbase, Firebird

1.4 OSTATNÍ TYPY DATABÁZÍ - FLAT FILE



Obr. 1 Flat file – textový soubor, vytvořený standardním editorem – seznam s položkami, oddělenými středníkem. [Zdroj: RWE, a.s.]



1.5 DATOVÉ SKLADY

Speciálním typem databáze je tzv. **datový sklad (data warehouse)**, který slouží k integraci podnikových dat z provozních relačních databází. Má odlišnou strukturu, kde data nejsou normalizovány a nevadí nám redundance dat – cílem je vytvořit strukturu, optimální pro následnou analýzu dat, založenou na použití statistických metod. Datové sklady jsou součástí tzv. OLAP systémů a slouží jako datová základna pro dolování dat (Data Mining). Datový sklad, nástroje OLAP a Data Mining plus prezentační vrstva se souhrnně označují jako software **Business Intelligence**.

1.6 HIERARCHICKÁ DB

- Umožňuje vyjádřit ve směru shora dolů jednosměrné vztahy typu 1:N
- Speciální typ síťového modelu – hierarchické vztahy
- Manipulace s daty – sekvenční procházení stromu
- Tento typ databáze se začíná znovu používat na webu jako alternativa k relačnímu modelu

Příkladem hierarchické databáze je **file systém**.

1.7 OBJEKTOVĚ ORIENTOVANÝ DB MODEL

- Informace jsou reprezentovány ve formě objektů jako u OO programování
- Objektový model není založen na klíčích jako relační
- Objekt je definován identifikátorem
- OQL – Object Query Language (objektová alternativa SQL)
- Rychlejší přístup na data, protože oproti relačním není třeba vytvářet join spojení
- Programovací jazyky a definice databázových objektů používají stejné struktury

Databáze: Caché, Gemstone, Objectivity/DB, ...

1.8 NOSQL DATABÁZE

- CouchDB
- Cassandra
- MongoDB
- BigTable
- Redis a další

Výhody:

- Rychlá práce s velkým objemem dat
- Jednoduchost
- Specializované na řešení určité problematiky
- Zpracování nestrukturovaných dat

NoSQL databáze se označují také jako bezeschémové. Pro ukládání dat používají např. formát JSON nebo XML.

Nejsou vhodné pro úlohy mající relační charakter.



1.9 KONTROLNÍ OTÁZKY

1. Co je to databáze?
2. Jaký je princip relační databáze?
3. Co je to integrita?
4. Co je to primární a cizí klíč?
5. Co je to normalizace relační databáze?
6. Jaké znáte typy databázových systémů?
7. Co je to SQL?
8. Co je to transakce?
9. Co znamenají ACID vlastnosti databáze?
10. Jaké jsou nejrozšířenější databázové systémy?
11. Jaké jsou výhody bezschémových (NoSQL) databází?

