Databáze

Základy databáze

> SŘBD Označení databáze je vlastně nepřesné a v odborné literatuře se setkáme s označením RDBMS (Relation DataBase Management System). Česky je to přeloženo jako "systém řízení báze dat" SŘBD, což zní opravdu hrozně a proto budu dále používat označení databázový stroj nebo SŘBD.

Databáze (neboli datová základna, též databanka) je systém souborů s pevnou strukturou záznamů. Tyto soubory jsou mezi sebou navzájem propojeny pomocí klíčů. V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento software se v české odborné literatuře nazývá systém řízení báze dat (SŘBD). Běžně se označením databáze – v závislosti na kontextu – myslí jak uložená data, tak i software (SŘBD).

MySQL

MySQL je tzv. relační databáze. Tento pojem označuje databázi založenou na tabulkách. Každá tabulka obsahuje položky jednoho typu. Můžeme mít tedy tabulku uživatelé, další tabulku články a další třeba komentáře. Tabulka je vnímána v oblasti databáze stejně jako tabulku v Excelu

Jméno	Příjmení	RČ	Datum nástupu
Karel	Novák	9907020001	25.03.2019
Michal	Novák	7211150525	01.06.2009
Anna	Kovářová	5315050007	13.11.2013

Každý sloupec má svůj specifický význam a specifickou datovou strukturu. V každém řádku jsou pak uložena konkrétní propojená data. Každý takový řádek, by pak měl být označený unikátním identifikátorem, například pořadové číslo, číslo O.P., rodné číslo, aby se na něj dalo jednoznačně zavolat. Pokud by identifikátorem bylo jméno, mohlo by se stát, že v tabulce seznamu lidí, Janů bylo více a tak v případě odstranění záznamu o Janovi, by došlo k odstranění všech Janů. Může se zdát, že je pro nás použití nějakého databázového systému zbytečné a bylo by jednodušší použít nějakého tabulkového programu. Ano, opravdu dá se databázový stroj (SW) nahradit např. Excelem, ale v takovém případě je potřeba dbát na mnoho aspektů, které s MySQL, jsou již automatické, nebo se snadněji z databáze získávají

 # ACID

je akronym slov Atomicity (nedělitelnost), Consistency (validita), Isolation (izolace) a Durability (trvanlivost).

Jednotlivé složky mají následující význam:

- Atomicity - Operace v transakci se provedou jako jedna atomická (nedělitelná) operace. Tzn. že pokud nějaká část operace selže, vrátí se databáze do původního stavu a žádné části transakce nebudou provedeny. Reálný příklad je např. převod peněz na bankovním účtu. Pokud se nepodaří peníze odečíst z jednoho účtu, nebudou ani připsány na účet druhý. Jinak by byla databáze v nekonzistentním stavu. Pokud bychom si práci s daty řešili sami, mohlo by se nám toto velmi jednoduše stát.
- Consistency - Stav databáze po dokončení transakce je vždy konzistentní, tedy validní podle všech definovaných pravidel a omezení. Nikdy nenastane situace, že by se databáze nacházela v nekonzistentním stavu.
- Isolation - Operace jsou izolované a navzájem se neovlivňují. Pokud se sejde v jeden okamžik více dotazů na zápis do stejného řádku, jsou vykonávány postupně, jako ve frontě.
- Durability - Všechna zapsaná data jsou okamžitě zapsána na trvanlivá úložiště (na pevný disk), v případě výpadku el. energie nebo jiného přerušení provozu RDBMS vše zůstane tak, jak bylo těsně před výpadkem.

Databáze (přesněji databázový stroj) je tedy černá skříňka, se kterou naše aplikace komunikuje a do které ukládá veškerá data. Její použití je velmi jednoduché a je odladěna tak, jak bychom si sami zápis dat v programu asi těžko udělali. Vůbec se nemusíme starat o to, jak jsou data fyzicky uložena, s databází komunikujeme pomocí jednoduchého dotazovacího jazyka SQL, viz dále. V dnešní době se vůbec nevyplatí zatěžovat se otázkou ukládání dat, jednoduše sáhneme po hotové databázi, kterých je obrovský výběr a jsou většinou zadarmo. O databázi občas hovoříme jako o 3. vrstvě aplikace (1. vrstva je uživatelské rozhraní, 2. vlastní logika aplikace, 3. je právě datová vrstva).

Jak vytvořit první databázi

Než se člověk vrhne na vytvoření opravdové databáze měl by se řídit pravidlem, dvakrát měř jednou řež. Pokud bude muset dojít ke změně některých zásadních částí, je tak ohrožená celková stabilita systému, který se tak může stát nepoužitelným, nebo bude často padat z důvodu chybovosti. Před nasazením, je nutné funkce databáze ozkoušet, pomocí naplnění dat.

 # Proces návrhu

Proces návrhu se skládá z následujících kroků:

- **Určení účelu databáze**: Tato část vám pomůže s přípravou na další kroky.
- **Vyhledání a uspořádání požadovaných informací**: Získejte všechny typy údajů, které chcete zaznamenat do databáze, například název produktu a číslo objednávky.
- **Rozdělení informací do tabulek**: Rozdělte jednotlivé údaje do hlavních skupin či předmětů, například Produkty nebo Objednávky. Každý předmět pak bude představovat

tabulku.

- **Převod jednotlivých informací do sloupců**: Rozhodněte se, jaké informace chcete ukládat v jednotlivých tabulkách. Každý údaj tvoří pole a je zobrazen jako sloupec v tabulce. Tabulka Zaměstnanci například může obsahovat pole Příjmení a Datum nástupu do zaměstnání.
- **Zadání primárních klíčů** :Pro každou tabulku zvolte primární klíč. Jedná se o sloupec, který slouží k jednoznačné identifikaci jednotlivých řádků. Příkladem může být ID produktu nebo ID objednávky. Existují tři druhy primárních klíčů, přirozený klíč, umělý klíč a složený.Přirozený primární klíč je údaj, který přímo evidujeme o předmětu, který ukládáme do tabulky např. rodné číslo, SPZ, VIN. Umělý primární klíč je údaj, který je uměle vytvořený za účelem jednoznačné identifikace zápisu, nejčastěji je to ID nebo pořadí. Složený klíč se skládá z více sloupců v tabulce.
- **Vytvoření relací mezi tabulkami**: Prohlédněte si tabulky a rozhodněte, jak spolu data v různých tabulkách souvisejí. Podle potřeby přidejte pole do tabulek nebo vytvořte nové tabulky, abyste objasnili relace. (Pokud v relaci předáváme PK, následně se ve druhé tabulce nazývá cizím klíčem.
- **Úprava návrhu**: Proveďte analýzu návrhu a vyhledejte chyby. Vytvořte tabulky a přidejte několik vzorových záznamů. Zjistěte, zda z tabulek získáte požadované výsledky. Podle potřeby návrh upravte.
- **Použití normalizačních pravidel**: Použijte normalizační pravidla dat a ověřte, že jsou tabulky strukturovány správně. Podle potřeby tabulky upravte.
- Relace typu 1:N
  Vezměme si jako příklad databázi pro sledování objednávek, jež obsahuje tabulky Zákazníci a Objednávky. Zákazník může vytvořit libovolný počet objednávek. To znamená, že pro každého zákazníka uvedeného v tabulce Zákazníci může existovat celá řada objednávek zaznamenaných v tabulce Objednávky. Typ relace mezi tabulkami Zákazníci a Objednávky je 1:N.Chcete-li znázornit relaci 1:N v návrhu databáze, vezměte primární klíč na straně 1 relace a přidejte jej jako další pole do tabulky na straně N relace. V tomto případě například přidáte nové pole – pole Kód z tabulky Zákazníci – do tabulky Objednávky a nazvete je Kód zákazníka. Aplikace Access může potom pomocí kódu zákazníka v tabulce Objednávky vyhledat u každé objednávky správného zákazníka.
- Relace typu M:N
  Nyní se podívejme na relaci mezi tabulkami Výrobky a Objednávky. Jedna objednávka může obsahovat více výrobků. Na druhou stranu se jeden výrobek může objevit v mnoha objednávkách. Z tohoto důvodu může pro každý záznam v tabulce Objednávky existovat mnoho záznamů v tabulce Výrobky. Navíc pro každý záznam v tabulce Výrobky může existovat celá řada záznamů v tabulce Objednávky. Tato relace se nazývá M:N. Všimněte si, že ke zjištění existující relace typu M:N mezi tabulkami je důležité vzít v úvahu obě strany relace.Chcete-li vyjádřit relaci typu M:N, je nutné vytvořit třetí tabulku, která se často nazývá spojená tabulka, jež rozdělí relaci typu M:N na dvě relace typu 1:N. Primární klíč z těchto dvou tabulek vložíte do třetí tabulky. Výsledkem je, že třetí tabulka zaznamená každý výskyt nebo instanci relace. V relaci M:N jsou například tabulky Objednávky a Výrobky a tato relace je definována vytvořením dvou relací 1:N s tabulkou Rozpis objednávek. V každé objednávce může být uvedeno více výrobků a každý výrobek může být uveden ve více objednávkách.
- Relace typu 1:1
  V relaci 1:1 odpovídá jednomu záznamu v první tabulce maximálně jeden záznam v druhé

tabulce a naopak jednomu záznamu v druhé tabulce maximálně jeden záznam v první tabulce. Tato relace není obvyklá, protože většina takto souvisejících informací by byla obvykle uložena ve stejné tabulce. Relaci 1:1 můžete použít k rozdělení rozsáhlé tabulky, k oddělení části tabulky z důvodů zabezpečení nebo k uložení informací, které mají vztah pouze k části hlavní tabulky. Při určování relace musí obě tabulky sdílet společné pole.

Revision #1
Created 2025-05-26 07:06:35 UTC by Magdalena Dobešová
Updated 2025-05-28 05:59:48 UTC by Magdalena Dobešová