Základní pojmy databáze

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Databáze je soubor dat (informací o objektech reálného světa), která spolu nějakým způsobem souvisí. Data je výraz pro údaje, používané pro popis nějakého jevu nebo vlastnosti pozorovaného objektu. Představují formu prezentace reálných objektů (znaky, symboly, obrázky, fakta, události), odrážejí tedy stav reality v určitém časovém okamžiku. Informace je zpráva, že nastal určitý jev. Vzniká přiřazením významu datům a existuje ve vztahu k příjemci. Slouží k informování o změnách ve vnímané realitě.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

S databázemi se v běžném životě setkáváme velmi často. Uvádíme běžné použití databází velkého rozsahu:

- databáze jízdních řádů,
- databáze státní správy,
- informační systémy bank, škol, úřadů
- nemocniční systémy evidence pacientů,
- databáze knihoven.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Základní pojmy

Entita

- Libovolný objekt (osoba, zvíře, věc či jev) reálného světa, který je zachycen v datovém modelu. Entita musí být rozlišitelná od ostatních entit a existovat nezávisle na nich.

Data

- Výraz pro údaje používané pro popis nějakého jevu nebo vlastnosti pozorovaného objektu. Data se získávají měřením nebo pozorováním, a lze je dělit na data spojitá a data atributivní. Data spojitá se přitom vztahují k nějaké spojité stupnici, zatímco data atributivní nikoliv

Informace

- Informace jsou data, která nám přinášejí nové poznatky.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Záznamy a atributy

- Řádky tabulky s hodnotami atributů pro jeden objekt (entitu), které se musí od sebe lišit. Atributy, jsou sloupce tabulky, Atributy mají určen svůj konkrétní datový typ a doménu, což je množina přípustných hodnot daného atributu. Řádek je řezem přes sloupce tabulky a slouží k vlastnímu uložení dat.

Atributy, pole

- Vlastnosti, které se u objektů (entit) sledují:
  – tvoří sloupce tabulky
  – mohou nabývat různých hodnot
  – pole jsou určitého datového typu (číslo, text, datum, …)

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Primární klíč

- Jednoznačně identifikuje záznam (řádek tabulky).
  Je to takový atribut (pole), který má pro každou entitu jedinečnou hodnotu, např. rodné číslo, většinou je to pomocné pole s identifikačním číslem záznamu (ID)

Cizí klíč

- Takový atribut, který je v jiné tabulce primárním klíčem.

Index

- Jedná se o způsob řazení tabulky.
  – pořadí záznamů se v tabulce během „života“ databáze nemění; index pomáhá k rychlému hledání dat v tabulce
  – index vytvoří pomocný soubor s řazením tabulky podle určitého pole
  – k jedné tabulce může být více indexů s řazením podle různých atributů
  – primární klíč je vždy indexem

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Struktura databáze

Nejpoužívanějšími databázemi jsou relační databáze. V nich jsou data ukládána v menších tabulkách, aby se zajistila minimální redundance (nadbytečnost) dat. Tabulky jsou vzájemně propojeny pomocí relací. Relace určují vztahy mezi tabulkami a zjišťují provázanost jednotlivých tabulek. Každá z těchto tabulek by měla obsahovat data týkající se pouze jednoho druhu objektu (např. tabulka objednávek, klientů, cen, zboží atd.). Pro vytvoření dobré databáze je nutné nejdříve navrhnout správnou strukturu jednotlivých tabulek. Tyto tabulky je pak nutné propojit pomocí relací. Tabulky tvoří základ celé struktury databáze. Základní pravidla pro návrh tabulek jsou následující:

- každá informace by měla být v databázi obsažena pouze jednou,
- každá tabulka by měla obsahovat informace o jednom typu objektu,
- při návrhu tabulek by se měl vzít do úvahy rozsah budoucích dat.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Databázová tabulka

V jedné tabulce by měly být informace o jednom typu objektu. Databázová tabulka je podobná běžné tabulce. Řádky obsahují záznamy (o jednom objektu) a sloupce označujeme položky či pole. Polem je totiž někdy zván i průsečík určitého řádku a sloupce, který obsahuje jedinou hodnotu (datový prvek).

Zaměstnanci

ID_zamestn	Jméno	Příjmení	Datum naro	Pohlaví	Telefonní čí:	ID_funkce
1	Tomáš	Novák	28.4.1980	muž	723 123 456	F_03
2	Josef	Koblížek	15.3.1976	muž	728 452 123	F_01
3	Petra	Maková	5.2.1985	žena	724 556 115	F_02
4	Václav	Sýkorka	30.6.1970	muž		F_06
5	Denisa	Rosolová	12.12.1956	žena		F_05
6	Michal	Aspik	3.6.1976	muž		F_04
7	Dominik	Kokeš	14.2.1981	muž		F_04
8	Tereza	Železná	25.10.1978	žena		F_02
9	Vladimíra	Mimořádná	17.11.1989	žena		F_02
10	Jakub	Pekelník	5.12.1973	muž		F_05

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Relace

Typy relací

Rozlišujeme tři základní typy relací:

- - **Relace typu 1:1** (výjimečný) – jednomu záznamu primární tabulky odpovídá právě jeden záznam v sekundární tabulce. Například jedné osobně je přiděleno právě jedno rodné číslo.
- - **Relace typu 1:N** (nejčastější) – jednomu záznamu primární tabulky odpovídá jeden nebo více záznamů v sekundární tabulce. Tento příklad jsme popsali ve výše uvedeném příkladu se zákazníky a objednávkami.
- - **Relace typu M:N** (velmi často) – jednomu nebo více záznamům primární tabulky odpovídá jeden nebo více záznamů v sekundární tabulce. Tyto relace řešíme pomocí spojovací tabulky, kterou pomocí dvou relací typu 1:N propojíme s původními tabulkami.

Referenční integrita

Referenční integrita udržuje neporušenost relací mezi tabulkami. Nedovolí nám vložit do sekundární tabulky záznam, který by neměl odpovídající záznam v primární tabulce. Dále si pohlídá si změnu hodnot cizího klíče při změně primárního klíče.
V rámci referenční integrity je možné nastavit pravidla pro odstraňování záznamů.

Databázové modely

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

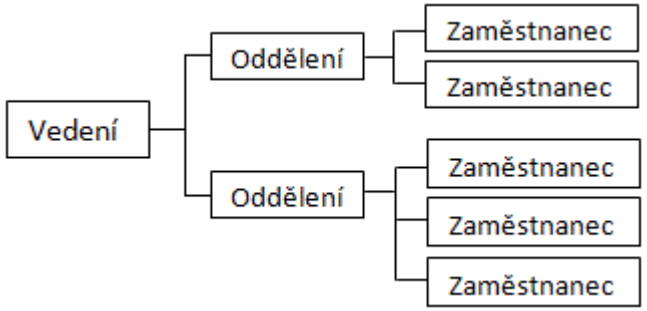
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Hierarchický model dat

Data jsou organizována do stromové struktury. Každý záznam představuje uzel ve stromové struktuře, vzájemný vztah mezi záznamy je typu rodič/potomek. Nalezení dat v hierarchické databázi vyžaduje navigaci přes záznamy směrem na potomka, zpět na rodiče nebo do strany na dalšího potomka. Největšími nevýhodami hierarchického uspořádání je složitá operace vkládání a rušení záznamů a v některých případech i nepřirozená organizace dat.

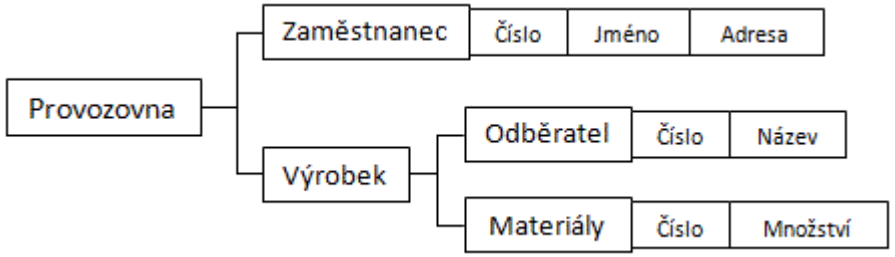


Obr. 1 - Hierarchický model

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Síťový model dat

Síťový model dat je v podstatě zobecněním hierarchického modelu, který doplňuje o mnohonásobné vztahy (sety). Tyto sety propojují záznamy různého či stejného typu, přičemž spojení může být realizováno na jeden nebo více záznamů. Přístup k propojeným záznamům je přímý bez dalšího vyhledávání, k dispozici jsou operace: nalezení záznamu podle klíče, posun na prvního potomka v dílčím setu, posun stranou na dalšího potomka v setu, posun nahoru z potomka na jeho rodiče v jiném setu. Nevýhodou síťové databáze je zejména nepružnost a obtížná změna její struktury.

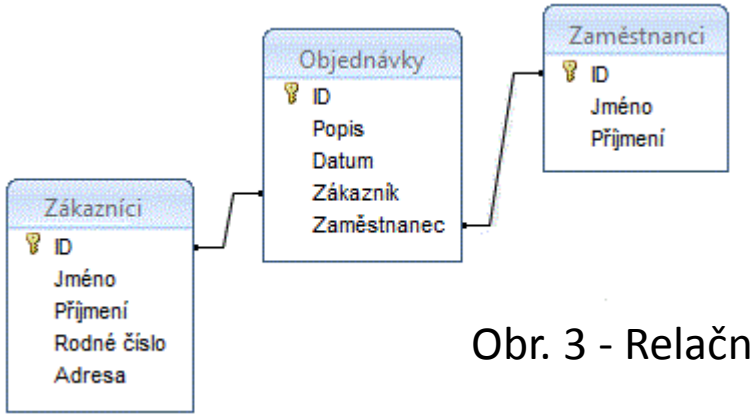


Obr. 2 - Síťový model

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Relační model dat

Relační databázový model je z uvedených nejmladší a zároveň nejpoužívanější. V současnosti je nejčastěji využíván u komerčních SŘBD. Model má jednoduchou strukturu, data jsou organizována v tabulkách, které se skládají z řádků a sloupců. V těchto tabulkách jsou prováděny všechny databázové operace.



Obr. 3 - Relační model

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Objektové databáze

Kromě databází relačních existují i již zmíněné databáze objektové. Ty řeší problém neslučitelnosti objektového a relačního přístupu. Poskytují stejný komfort, jako ORM, ale vnitřně není třeba data převádět do tabulek, ukládají se rovnou jako objekty. Teoreticky neexistuje výkonnostní ani jiný důvod, proč by neměly nahradit databáze relační. V praxi se ale bohužel téměř nepoužívají a můžeme jen doufat, že se to časem změní.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Základy objektové orientace. Objekty a třídy

Objektově orientovaný model je založen na dekompozici informací z reálného světa na tzv. objekty. Objektem se rozumí každá (i strukturovaná) entita, která je jednoznačně a nezávisle identifikovatelná v rámci určitého kontextu okolního světa. Objekt tak má jednoznačnou identitu, každé dva i jinak datově shodné objekty jsou vzájemně odlišitelné. Identita objektu je určena identifikátorem (object identifier – oid), který je generovaný systémem, unikátní, neměnný po dobu existence objektu, skrytý pro programátora i koncového uživatele. Objekty jsou charakterizovány pomocí tříd. Třída je abstraktní popis objektu, určuje datové složky objektu a operace (nazývané metody), které lze nad objektem provádět. Každý objekt je instancí nějaké třídy, od jedné třídy je možné instanciovat obecně neomezený počet strukturálně shodných objektů.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Literály

Kromě objektů se v rámci objektově orientovaného modelu zavádí i pojem literálu. Literál je datová entita určitého datového typu, která však na rozdíl od objektu nemá vlastní identitu. Literály se obvykle vyskytují jako datové atributy objektů. Množina operací nad datovým typem literálu je pevně stanovená, není možné ji měnit. S objekty a literály souvisí pojem proměnlivosti (mutability). Proměnlivost je chápána jako schopnost měnit data při zachování identity – v tomto smyslu jsou objekty proměnlivé, protože je možné měnit hodnoty jejich datových složek a přitom si ponechávají původní identitu. Naproti tomu literály proměnlivé nejsou.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Objektově relační datový model

Objektově-relační datový model je klasická tabulková databáze rozšířená o **abstraktní datové typy** (ADT). ADT jsou uživatelem definované typy skládající se ze základních datových typů databáze. Čímž porušuje 1NF??

Objektové rysy jsou dnes implementovány ve všech velkých SŘBD.

Objektové typy a jejich metody jsou uloženy spolu s daty v databázi, programátor tedy nemusí vytvářet podobné struktury v každé aplikaci.

Programátor může přistupovat k množině objektů, jako by se jednalo o jeden objekt.

Objekty mohou jednoduše reprezentovat vazby, kdy jedna entita se skládá z jiných entit (bez nutnosti použít vazeb).

Metody jsou spouštěny na serveru – nedochází k neefektivnímu přenosu dat po síti.

Integrita databáze

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Integrita databáze znamená, že data v ní uložená jsou konzistentní vůči definovaným pravidlům. Lze zadávat pouze data, která vyhovují předem definovaným kritériím (např. musí respektovat datový typ nastavený pro daný sloupec tabulky, či další omezení hodnot přípustných pro daný sloupec). K zajištění integrity slouží integritní omezení. Jedná se o nástroje, které zabrání vložení nesprávných dat či ztrátě nebo poškození stávajících záznamů v průběhu práce s databází. Například je možné zajistit mazání dat, která již ztratila svůj význam - například smažeme-li uživatele, odstraní se i zbytek jeho záznamů v ostatních databázových tabulkách.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Druhy integritních omezení

- *Entitní integritní omezení* – povinné integritní omezení, které zajišťuje úplnost primárního klíče tabulky (zamezí uložení dat, jež by v těchto polích byla stejná jako v nějakém jiném řádku tabulky)
- *Doménová integritní omezení* – zajišťují dodržování datových typů/domén definovaných u sloupců databázové tabulky
- *Referenční integritní omezení* – zabývají se vztahy dvou tabulek, kde jejich relace je určena vazbou primárního a cizího klíče
- *Aktivní referenční integrita* – definuje činnosti, které databázový systém provede, pokud jsou porušena některá pravidla

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Vztahy mezi tabulkami

Relace slouží ke svázání dat, která spolu souvisejí a jsou umístěny v různých databázových tabulkách. V zásadě rozlišujeme čtyři typy vztahů.

- 1:1 (záznamu odpovídá právě jeden záznam v jiné databázové tabulce a naopak)
- 1:N (přiřazuje jednomu záznamu více záznamů z jiné tabulky)
  - jedná se o nejpoužívanější typ relace, jelikož odpovídá mnoha situacím v reálném životě
- M:N (umožňuje několika záznamům z jedné tabulky přiřadit několik záznamů z tabulky druhé)
  - tento vztah bývá z praktických důvodů nejčastěji realizován kombinací dvou vztahů 1:N a 1:M, které ukazují do pomocné, tzv. vazební tabulky složené z kombinace obou použitých klíčů

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Normální formy

- Pod pojmem normalizace rozumíme proces zjednodušování a optimalizace navržených struktur databázových tabulek. Hlavním cílem je navrhnout databázové tabulky tak, aby obsahovaly minimální počet redundantních dat. Správnost navržení struktur lze ohodnotit některou z následujících normálních forem.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Databázová integrita

- **Integrita databáze** znamená, že databáze vyhovuje zadaným pravidlům – integritním omezením. Tato integritní omezení jsou součástí definice databáze, a za jejich splnění zodpovídá systém řízení báze dat.
- Integritní omezení se mohou týkat jednotlivých hodnot vkládaných do polí databáze (například známka z předmětu musí být v rozsahu 1 až 5), či může jít o podmínku na kombinaci hodnot v některých polích jednoho záznamu (například datum narození nesmí být pozdější než datum úmrtí). Integritní omezení se může týkat i celé množiny záznamů daného typu – může jít o požadavek na unikátnost hodnot daného pole či kombinace polí v rámci celé množiny záznamů daného typu, které se v databázi vyskytují (například číslo průkazu v záznamech o osobách).

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Integritní omezení

Je třeba zajistit, aby se do databáze dostaly jen takové záznamy, které odpovídají vztahům v reálném světě (např. atribut věk by neměl nabývat záporných hodnot). K ošetření takových případů se používají integritní omezení, která slouží ke stanovení určitého rozmezí hodnot, jakých může konkrétní atribut nabývat. Integritní omezení specifikují, jaká omezení a vnitřní vztahy musí splňovat data v databázi.

Relační databázový systém

- musí umožňovat efektivní správu a analýzu uložených dat
- musí plnit tyto tři základní funkce:
- umožnit definovat typ dat
- umožnit pracovat s daty
- systém by měl obsahovat prostředky umožňující provést požadované činnosti s daty - např. řazení dat, filtrování, výpočty s daty, řídit správu dat
- systém kontroluje zejména přístup k datům, tedy kdo je oprávněn k datům přistupovat a kdo může provádět jejich aktualizaci
- systém řídí sdílení dat mezi více uživateli

Relační databázový model

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Databáze dle relačního modelu musí splňovat tyto dvě vlastnosti:

- Databáze je chápana uživatelem jako množina relací a nic jiného.
- V relačním SŘBD jsou k dispozici minimálně operace selekce, projekce a spojení, aniž by se vyžadovaly explicitně předdefinované přístupové cesty pro realizaci těchto operací.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

12 pravidel pro relační SŘBD:

1. Informační pravidlo:

Všechny informace v relační databázi jsou vyjádřeny explicitně na logické úrovni jediným způsobem - hodnotami v tabulkách.

2. Pravidlo jistoty:

Všechna data v relační databázi jsou zaručeně přístupná kombinací jména tabulky s hodnotami primárního klíče a jménem sloupce.

3. Systematické zpracování nulových hodnot:

Nulové hodnoty jsou plně podporovány relačním SŘBD pro reprezentaci informace, která není definována a to nezávisle na datovém typu.

4. Dynamický on-line katalog založený na relačním modelu:

Popis databáze je vyjádřen na logické úrovni stejným způsobem jako zákaznická data, takže autorizovaný uživatel může aplikovat stejný relační jazyk ke svému dotazu jako uživatel při práci s daty.

5. Obsáhlý datový podjazyk:

Relační systém může podporovat několik jazyků a různých módů použitých při provozu terminálu. Nicméně musí být nejméně jeden příkazový jazyk s dobře definovanou syntaxí, který obsáhle podporuje definici dat, definici pohledů, manipulaci s daty jak interaktivně, tak programem, integritní omezení, autorizovaný přístup k databázi, transakční příkazy apod.

6. Pravidlo vytvoření pohledů

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

7. Schopnost vkládání, vytvoření a mazání:

Schopnost zachování relačních pravidel u základních i odvozených relací je zachována nejen při pohledu na data, ale i při operacích průniku, přidání a mazání dat.

8. Fyzická datová nezávislost:

Aplikační programy jsou nezávislé na fyzické datové struktuře.

9. Logická datová nezávislost:

Aplikační programy jsou nezávislé na změnách v logické struktuře databázového souboru.

10. Integritní nezávislost:

Integritní omezení se musí dát definovat prostředky relační databáze nebo jejím jazykem a musí být schopna uložení v katalogu a nikoliv v aplikačním programu.

11. Nezávislost distribuce:

Relační SŘBD musí být schopny implementace na jiných počítačových architekturách.

12. Pravidlo přístupu do databáze:

Jestliže má relační systém jazyk nízké úrovně, pak tato úroveň nemůže být použita k vytváření integritních omezení a je nutno vyjádřit se v relačním jazyce vyšší úrovně.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- **Doména** je množina všech hodnot, kterých může nabývat atribut. Jinak řečeno obor hodnot atributu. V praxi je doména dána integritním omezením (IO).
  *Doména atributu Přijmení z obrázků je množina {Dudak, Novák, Dvořák}. *pozn.*
- **Atribut** je vlastnost entity. Z pohledu tabulky jde o sloupec.
- **Relační schéma** můžeme chápat jako strukturu tabulky (atributy a domény).
- **Příklad pro tabulku (relaci) Učitel:**
- *Atributy:* ID, jméno, příjmení, funkce, kancelář

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Vlastnosti relačního datového modelu

- Z definice relace vyplývají tyto jejich tabulkové vlastnosti:
- homogenita sloupců (prvky domény)
- každý údaj (hodnota atributu ve sloupci) je atomickou položkou
- na pořadí řádků a sloupců nezáleží (jsou to množiny prvků/atributů)
- každý řádek tabulky je jednoznačně identifikovatelný hodnotami jednoho nebo několika atributů (primárního klíče)

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

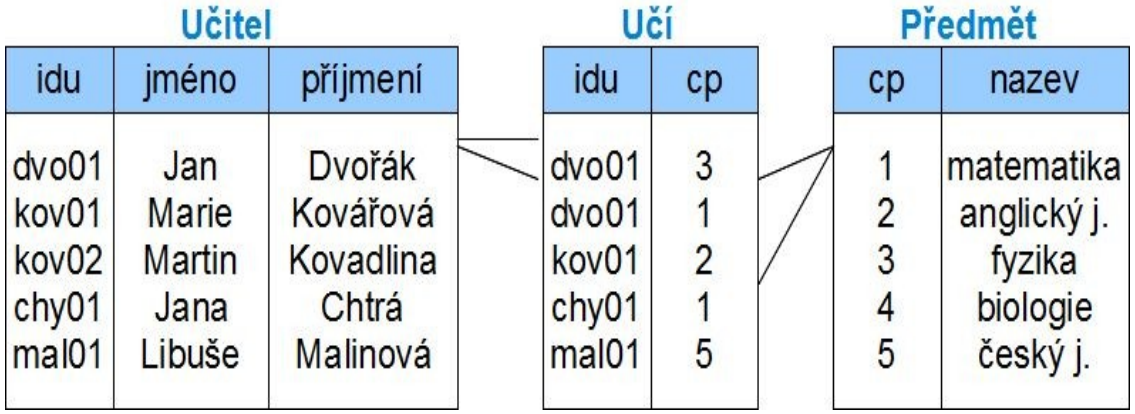
Vazby v relačním modelu

Obecně se vazby v relačním modelu realizují pomocí další relace (tabulky). Jedná se o tzv. vazební tabulku. Ta obsahuje ty atributy relací (tabulek, které se vazby účastní), které jednoznačné identifikují jejich entity - primární klíče. Obsahuje-li tabulka atribut, který slouží jako primární klíč v jiné tabulce, pak obsahuje cizí klíč. Vazební tabulka tedy obsahuje cizí klíče.

Na obrázku dole máme zobrazenou relaci Učitel s primárním klíčem idu a relaci Předmět s primárním klíčem cp. K vyjádření vztahu Učitel UČÍ Předmět byla vytvořena nová relace Učí, která obsahuje dva cizí klíče (idu odkazující na učitele a cp odkazující na entitu předmětu). Nyní tedy můžeme přes vazební tabulku pospojovat učitele s předměty, podle toho kdo co učí.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

A proč jsme do tabulky Učitel nepřidali jen atribut předmět? Jednoduše proto, že učitel může učit více předmětů. Mezi učitelem a předmětem je vazba M:N. Takže ani kdybychom přidali do tabulky Předmět atribut učitel, bychom tento vztah nevyřešili (předmět může být učen více učiteli). Pro vztah 1:N by to však šlo a v praxi se to tak i dělá.
Takže vazební tabulka je nutná jen k realizací vztahů M:N.



Učitel

idu	jméno	příjmení
dvo01	Jan	Dvořák
kov01	Marie	Kovářová
kov02	Martin	Kovadlina
chy01	Jana	Chtrá
mal01	Libuše	Malinová

Učí

idu	cp
dvo01	3
dvo01	1
kov01	2
chy01	1
mal01	5

Předmět

cp	nazev
1	matematika
2	anglický j.
3	fyzika
4	biologie
5	český j.

Ukázka vazební tabulky pro vztah Učí mezi tabulkami Učitel a Předmět. Vztah je M:N, tedy že jeden učitel může učit N předmětů a jeden předmět může být učen M učiteli.

Základy SQL - vytváření dotazů

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

SQL - *Structured Query Language* (Strukturovaný dotazovací jazyk) - je obecný nástroj pro manipulaci, správu a organizování dat uložených v databázi počítače. Je v prvé řadě určen uživatelům, i když jej v mnoha směrech využívají i tvůrci aplikací. Je adaptovatelný pro jakékoliv prostředí.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

SQL je specializovaný programovací jazyk, který se používá ve vhodném prostředí buď uživatelsky nebo interaktivně k okamžitému řešení úloh (nejčastěji dotazy), nebo se jeho příkazy vkládají do hostitelského jazyka. SQL není však plnohodnotným samostatným programovacím jazykem, např. proto, že se v něm ve většině implementací nenachází řídicí programové konstrukce a další požadované prvky, které by měl obsahovat každý obecný programovací jazyk. SQL je tedy standardizovaný nástroj pro práci s relačními databázemi. Nepředstavuje databázový systém, ale různě integrovanou součást systému řízení bází dat.

Pracuje s relačními databázemi, ve kterých se uživatel dívá na data v podobě soustavy provázaných tabulek. Každá tabulka představuje množinu dat, která je uspořádaná v řádcích (záznamech) a sloupcích (položkách). Na hodnotu dat se uživatel odkazuje jako na prvek v matici.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Klíčovým pojmem v jazyku SQL je ***příkaz***. Každý příkaz začíná klíčovým slovem. Slovo vyjadřuje orientačně, jakou činnost daný příkaz provádí. Za klíčovým slovem následuje jedna nebo více volitelných klauzulí, které blíže specifikují povahu vykonávané činnosti, nebo určují data, s nimiž má příkaz pracovat.
- Každá klauzule začíná klíčovým slovem, jako jsou např. FROM nebo WHERE. Některé klauzule jsou povinné, jiné volitelné. Každá implementace SQL používá kromě standardních klauzulí, daných konvencemi ANSI/ISO, své vlastní klauzule, někdy se i činnost standardních klauzulí mírně či více liší.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Databáze SQL dotazy

- Mezi nejdůležitější příkazy patří: SELECT, INSERT, DELETE, CREATE, FROM, WHERE a mnohé další v následujících kapitolách si na příkladech jednotlivé příkazy představíme. Nejprve si ale uděláme jednoduchou tabulku, na které si příkazy představíme (to aby bylo jasnější, co který ten příkaz dělá)

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Příkaz SELECT

Důležitý příkaz, Použijeme naši vytvořenou tabulku a otestujeme si na ní aplikace tohoto příkazu. Nejjednodušší forma příkazu je tato:

```
SELECT *
FROM zamestanci;
```

Vypíše obsah celé tabulky "zamestananci„

```
SELECT jmeno
FROM zamestanci;
```

Vypíše obsah sloupce "jmeno" z tabulky zamestanci

```
SELECT AVG(plat)
FROM zamestanci;
```

```
SELECT SUM(plat)
FROM zamestanci;
```

AVG(sloupec) vypočte půměr z daného sloupce, SUM(sloupec) vypočte sumu z hodnot v daném sloupci, případně můžeme i v samotném dotazu provádět výpočty.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Obecná syntaxe jednoduchého dotazu

SELECT *seznam sloupců* **FROM** *tabulka* **WHERE** *vyhledávací podmínky* **ORDER BY** *sloupce řazení*

Seznam sloupců je buď seznam sloupců oddělených čárkou, nebo znak * pro výběr všech sloupců. Název sloupce je buď pouze název sloupce nebo **název_tabulky.název_sloupce.** Také je možné použít zápis **tabulka.*** pro výběr všech sloupců z tabulky. *Tabulka* může být cokoliv z:

relace (reálná tabulka v databázi)

výsledek jiného SELECT dotazu

pohled

výsledek operátoru JOIN (virtuální tabulky).

Vyhledávací podmínka je podmínka, které musí být splněna pro každý záznam, který se vypíše ve výsledcích dotazu. Samozřejmě pokud chceme nějaký záznam vypsat, tak musí nejprve existovat ve zdrojové tabulce. Část WHERE je tedy *omezující* podmínka, pokud není uvedena, tak se vypisují všechny řádky z tabulky. *Sloupce řazení* je seznam sloupců oddělených čárkou, podle kterých bude výsledná tabulka seřazena, první sloupec je primární kritérium řazení, druhý sloupec je sekundární řazení - pokud nelze rozhodnout podle prvního sloupce, …

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

SQL – složitější dotazy

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Příklad 1

Zadání: napište SQL dotaz, který vybere z databáze všechny osoby, které mají ICQ číslo; vyberte jméno, příjmení a ICQ číslo osoby; seřaďte podle příjmení vzestupně.

Varianta 1

Nejjednodušší - počítá s tím, že v tabulce typy_kontaktu hodnota id_typy_kontaktu = 1 odpovídá typu ICQ. Operátor JOIN pracuje vždy se dvěma tabulkami. Spojovací podmínka je vždy rovnost hodnot v nějakých sloupcích. Spojovací podmínka musí obsahovat obě tabulky, které se vyskytují v části JOIN. Spojovací podmínka musí obsahovat sloupce, které zajišťují vazbu mezi tabulkami. V tabulce osoby není žádný odkaz na tabulku kontakty. V tabulce kontakty je jeden odkaz na tabulku osoby - sloupec id_osoby. INNER JOIN vybere z tabulky pouze ty záznamy, které vyhovují spojovací podmínce - tedy pouze osoby, které mají nějaký kontakt (protože je však v dotazu zadaná i vyhledávací podmínka je v tomto konkrétním případě možné použít i LEFT i RIGHT JOIN, je to ovšem jen shoda okolností).

- SELECT je opět příkaz, který může mít spoustu dalších slov, kombinovat několik tabulek do sebe atd. Ale úplný základ sytaxe je: SELECT, pak následuje seznam sloupců, které chceme získat, případně agregační či jiné funkce, FROM a název tabulky, ze které chceme data získat, WHERE a výraz s omezeními, která musí platit pro žádané řádky. V našem případě chceme, aby sloupec mesto obsahoval přesně hodnotu “Chomutov”.
- S agregační funkcí to vypadá třeba takto – chceme zjistit kolik řádků, tedy lidí je z Chomutova:
- SELECT COUNT(*) FROM lide WHERE mesto = "Chomutov";
- Vrátí nám to jeden řádek v jednom sloupci a bude obsahovat číslo 2.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Databáze SQL dotazy

- Když místo “SELECT sloupce” zadáme “DELETE”, máme dotaz pro smazání řádků. Funguje podobně jako SELECT (je tam podmínka WHERE) ale místo získání výsledků smaže řádky odpovídající podmínce. Pro smazání Adama stačí zadat:
- DELETE FROM lide WHERE jmeno = "Adam";
- Z těch nejpoužívanějších SQL dotazů ještě UPDATE. Ten provede úpravu řádku. Pokud Kateřina zestárne o jeden rok, provedeme aktualizaci třeba takto:
- UPDATE lide SET vek = 30 WHERE jmeno = "Kateřina";
- můžeme ale taký použít matematický výraz a reference na existující hodnoty sloupců. Takže zvýšit hodnotu sloupce vek o jedna jde i takto:
- UPDATE lide SET vek = vek+1 WHERE jmeno = "Kateřina";

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Napojení dalších tabulek do SELECT dotazu se dělá pomocí JOIN. To téma vyžaduje mít víc než jednu tabulku a nějakou relaci mezi nima. Například tabulka lide jak to máme, kde každý člověk má unikátní id a pak třeba tabulku napady se sloupci clovek_id a napad. Ve sloupci clovek_id by byly cisla lidi, kteří daný nápad vymysleli a ve sloupci nápad by byl text nápadu. Pak bychom mohli udělat dotaz, který by vypsal všechny nápady a ke každému přidal i sloupec s názvem člověka. Delaily jsou ale nad rámec tohoto úvodu:
- SELECT napady.napad, lide.jmeno FROM napady LEFT JOIN lide ON lide.id = napady.clovek_id;

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Příkaz SELECT

Důležitý příkaz, Použijeme naši vytvořenou tabulku a otestujeme si na ní aplikace tohoto příkazu. Nejjednodušší forma příkazu je tato:

```
SELECT *
FROM zamestanci;
```

Vypíše obsah celé tabulky "zamestananci„

```
SELECT jmeno
FROM zamestanci;
```

Vypíše obsah sloupce "jmeno" z tabulky zamestanci

```
SELECT AVG(plat)
FROM zamestanci;
```

```
SELECT SUM(plat)
FROM zamestanci;
```

AVG(sloupec) vypočte půměr z daného sloupce, SUM(sloupec) vypočte sumu z hodnot v daném sloupci, případně můžeme i v samotném dotazu provádět výpočty.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Obecná syntaxe jednoduchého dotazu

SELECT *seznam sloupců* **FROM** *tabulka* **WHERE** *vyhledávací podmínky* **ORDER BY** *sloupce řazení*

Seznam sloupců je buď seznam sloupců oddělených čárkou, nebo znak * pro výběr všech sloupců. Název sloupce je buď pouze název sloupce nebo **název_tabulky.název_sloupce.** Také je možné použít zápis **tabulka.*** pro výběr všech sloupců z tabulky. *Tabulka* může být cokoliv z:

relace (reálná tabulka v databázi)

výsledek jiného SELECT dotazu

pohled

výsledek operátoru JOIN (virtuální tabulky).

Vyhledávací podmínka je podmínka, které musí být splněna pro každý záznam, který se vypíše ve výsledcích dotazu. Samozřejmě pokud chceme nějaký záznam vypsat, tak musí nejprve existovat ve zdrojové tabulce. Část WHERE je tedy *omezující* podmínka, pokud není uvedena, tak se vypisují všechny řádky z tabulky. *Sloupce řazení* je seznam sloupců oddělených čárkou, podle kterých bude výsledná tabulka seřazena, první sloupec je primární kritérium řazení, druhý sloupec je sekundární řazení - pokud nelze rozhodnout podle prvního sloupce, …

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- To jsou všechny nejčastěji používané SQL dotazy (klauzule). Jednotlivé dotazy mohou mít víc slov pro složitější a přesnější dotazy. Pro jejich pochopení je však nutno studovat víc jednotlivé dokumentace, nebo hledat konkrétní problém. Například když použijeme v SELECT agregační funkci, získáme výsledek agregace celé tabulky. Pokud ale třeba chceme mít průměry teplot v rocích a máme teploty po měsících, musíme použít GROUP BY a když dáme do google group by years datetime mysql, zjistíme, že potřebujeme DATETIME funkci YEAR:
- GROUP BY YEAR(record_date)
- Kromě GROUP BY má select i ORDER BY, kde si zadáme podle jakých sloupců výsledky řadit a jestli sestupně nebo vzestupně. Důležité je někdy pořadí slov. Pro SELECT je opět v dokumentaci a vidíme, že nejdřív v dotazu musíme mít GROUP BY … a pak ORDER BY … .

Grafové databáze

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Graf je datová struktura skládající se z vrcholů a hran. Grafová databáze je systém na ukládání a zpracování dat v podobě grafu. V mnoha případech je modelování domény grafem velmi přirozené – mezi takové domény patří vztahy mezi lidmi, mezi geny a proteiny, mobilní sítě, distribuční sítě různého druhu, neuronové sítě nebo třeba ekologické sítě zachycující interakce organismů.
- Mezi grafové databáze se často řadí různé systémy, které spravují data v podobě grafů. Na základě takovéto definice by bylo teoreticky možné pohlížet i na relační databáze, resp. na jejich nadstavby jako na grafové databáze. Relační databáze ovšem neumožňují efektivní uložení a dotazování grafových dat.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Opravdové grafové databáze

- Opravdovými grafovými databázemi nazveme pro účel tohoto článku takové databáze, které vyžadují pouze konstantní čas O(1) pro průchod hranou grafu.
- Opravdové grafové databáze, podobně jako jiné NoSQL databáze, ukládají data v denormalizované (již „zjoinované“) podobě. U opravdové grafové databáze se tedy platí především u zápisu, zatímco dotazování (čtení) je levnější.
- Příkladem opravdové grafové databáze je systém Neo4j, který je ve vývoji již více než deset let a je dostatečně vyspělý pro produkční nasazení, nebo například novější systém OrientDB.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Příkladem databází, které jsou schopné ukládat a dotazovat data v podobě grafů, ale ne nutně efektivně procházet grafy, je většina triplestorů (Sesame, Jena, Virtuoso) nebo FlockDB (projekt Twitteru).
- Podobně jako je tomu u většiny NoSQL databází, ani pro grafové databáze neexistuje jednotný dotazovací jazyk. Nicméně mnoho grafových databází implementuje jazyk na procházení grafů Gremlin vytvořený ve spolupráci s autory Neo4j.
- Přechod z relačního světa do světa grafů vyžaduje posun v uvažování a pohledu na data. I když grafy jsou často mnohem intuitivnější než tabulky, postřehl jsem stále opakující se chyby u lidí, kteří s modelováním grafů začínají. V tomto článku se podíváme na jednu z nejčastějších chyb – modelování obousměrných vztahů, a nakonec si ukážeme reálný příklad, ve kterém budeme nadále v seriálu pokračovat.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Vztahy v Neo4j musí být nějakého typu, který dává vztahu sémantický význam a také musí mít určený směr. Právě ten vyjadřuje smysl mezi entitami. Jinými slovy, vztah je nejednoznačný bez určení směru vztahu.
- Například následující graf ukazuje, že Česká republika porazila (DEFEATED) Švédsko v ledním hokeji. Kdyby se směr vztahu obrátil, Švédové by byli mnohem šťastnější. Bez směru vůbec nevíme, kdo je vítězem, a proto je takový vztah nejednoznačný.
- Všimněte si, že z existence tohoto vztahu vyplývá vztah opačném směru, jak je ukázáno níže v dalším grafu. Jedná se o velmi častý případ. Ještě jednou si popíšeme na jiném příkladu, že film Pulp Fiction režíroval (DIRECTED) Quentin Tarantino znamená také, že Quentin Tarantino je režisérem (IS_DIRECTOR_OF) filmu Pulp Fiction. A tímto by mohlo dojít k velmi mnoho párovým vtahům.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

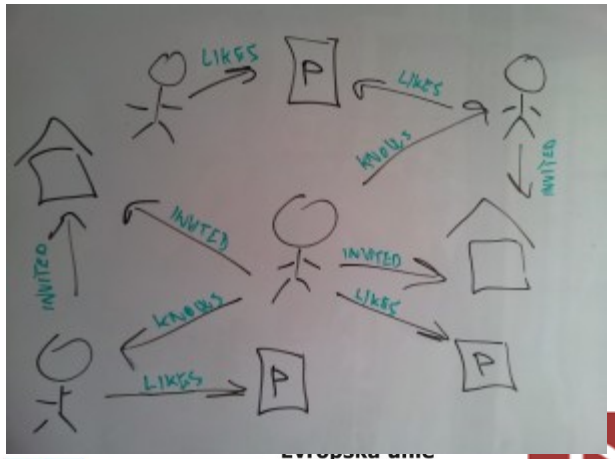
Metodika modelování grafové databáze

- V následujících článcích si ukážeme návrh, implementaci a testování grafové databáze Neo4j na projektu BestPartyToday (jedná se celosvětový pártylist, který nabízí detailní statistiky k akcím). V současné době se v projektu používá databáze MySQL, která obsahuje desítky milionů záznamů, takže nás také čeká přesunutí dat z relačního databázového úložiště do grafového. Ovšem v dnešním příspěvku si pouze namodelujeme podobu grafové databáze z vybraných tabulek a atributů současné struktury MySQL databáze.
- O konceptuálním návrhu grafové databáze se pojednává jako o tzv. Whiteboard friendliness. Pro rychlé promítnutí všech myšlenek a postřehů, které byly vzneseny během návrhového brainstormingu, postačí pouze flipchart. Výsledný nákres je velmi jednoduché následně prezentovat ostatním projektovým skupinám (retail department, accound managers, sales managers atd.), které nemají technické vzdělání. Pochopení návrhu pro ně už není překážkou, jelikož nákres je možné jednoduše přizpůsobit do podoby reálného života.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Databázová tabulka

- Na fotografii finálního návrhu grafové databáze pro projekt BestPartyToday je využito pěti nejvýznamnějších tabulek současné relační databáze, které jsou reprezentovány ikonami znázorňující nodes a směrovými šipkami prorelationships. Skrze entity (nodes a relationships) budeme procházet graf a získávat požadovaná data, která budou zpracována a zobrazena uživateli aplikace. Jejich význam je popsán níže v tabulce. Přehled entit grafového modelu.



NoSql databáze

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

NoSQL databáze NoSQL databáze jsou tématem této práce, proto je jim věnována největší část. Nejdříve je třeba říct, co to vlastně NoSQL databáze jsou. Dále je kapitola věnována CAP teorému, na kterém je vysvětleno, že jedním z důvodů, proč existuje tolik NoSQL produktů, je to, že každý z nich může zajistit jen určité vlastnosti na úkor jiných a uživatel se pak musí rozhodnout, která kombinace vlastností je pro něj nejdůležitější a podle toho vybrat produkt. Dále se práce věnuje škálovatelnosti, tedy schopnosti navyšování výkonu databázového 15 serveru, ať již samotným vylepšováním serveru o výkonnější hardware nebo rozprostřením databáze na více serverů. Spolu se škálovatelností je třeba se taky zmínit o shardingu, tedy o technice, která rozděluje databáze na více částí, jež mohou fungovat samostatně a rychleji, než jeden celek. Poté následuje seznámení s nejznámějšími NoSQL produkty, kde jsou následně vybráni tři zástupci, se kterými je provedeno bližší seznámení. Popis Jak již zaznělo v úvodu, NoSQL databáze jsou ty, které nejdou relační cestou řízení dat, ale jinou. Nejsou primárně postavené na tabulkách a nevyužívají SQL pro práci s daty. Jejich doména je vysoká optimalizace pro vyhledávání, na úkor malé funkcionality, která se často omezuje na prosté ukládání dat. Tyto nedostatky, v porovnání s SQL, jsou kompenzovány škálovatelností a výkonem u určitých datových modelů. NoSQL se hodí na ukládání velkého objemu dat, u kterých není třeba uchovávat vzájemné vztahy.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Síťový model dat

NoSQL je doslovně kombinací dvou slov: No a SQL. Je to tedy technologie, která stojí proti SQL. Zkratka může být matoucí a mezi lidmi neexistuje jednotný názor na to, co znamená. Nejvíce rozšířený je ovšem ten, který tvrdí, že jde o acronym „Not only SQL.“ Ať už zkratka znamená cokoliv, NoSQL je dnes zastřešující název pro všechny databáze, které nejdou cestou známého RDBMS (relační systém řízení báze dat), ale jdou svojí vlastní, často spojovanou s velkými objemy dat.

Popis Jak již zaznělo v úvodu, NoSQL databáze jsou ty, které nejdou relační cestou řízení dat, ale jinou. Nejsou primárně postavené na tabulkách a nevyužívají SQL pro práci s daty. Jejich doména je vysoká optimalizace pro vyhledávání, na úkor malé funkcionality, která se často omezuje na prosté ukládání dat. Tyto nedostatky, v porovnání s SQL, jsou kompenzovány škálovatelností a výkonem u určitých datových modelů. NoSQL se hodí na ukládání velkého objemu dat, u kterých není třeba uchovávat vzájemné vztahy. Strukturovaná data jsou ovšem povolena. U transakcí nemohou NoSQL databáze nabídnout plnou podporu ACID, nýbrž pouze Eventual consistency (Výsledná shoda). Jde o situaci, kdy díky zanedbání ACID získáme větší dostupnost a také lepší škálovatelnost. Tento přístup bez „silné konzistence“ je vhodný pro NoSQL, které ho taky často aplikují. NoSQL má distribuovanou architekturu, která je odolná vůči chybám. Některá data bývají umístěna na několika serverech a tím selhání jednoho se dá tolerovat. Tyto databáze většinou škálují horizontálně a spravují velké objemy dat, u kterých je důležitější výkon než konzistence.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Relační model dat

Především je důležité zdůraznit, co NoSQL není – rozhodně to není NO SQL, tedy trendy odmítání relačních databází. Zkratka NoSQL znamená Not Only SQL, tedy uvědomění si toho, že relační databáze není jediná možnost řešení persistence, že existují i alternativy, které mohou být v některých případech vhodnější.

U rozsáhlejších aplikací pak můžeme dojít k tomu, že na některá data je vhodná relační databáze, na jiná NoSQL databáze a na další data úplně jiná NoSQL databáze. Tento pragmatický přístup, kdy se mixuje použití více databází v jednom projektu, se často označuje jako polyglot persistence.

NoSQL databáze vznikaly (a vznikají) jako řešení reálných problémů – není to tedy bláznivý výmysl akademiků odtržených od reality, ale vysoce praktická záležitost. Zrod mnoha NoSQL databází je spjat s projekty, které se musely vypořádat s obrovským množstvím dat – Facebook (Cassandra), Google (BigTable), Amazon (Dynamo), LinkedIN (Voldemort) atd.

Použití NoSQL databáze může mít ale smysl i tehdy, pokud máme méně dat (které by v pohodě zvádla relační databáze) – třeba protože nějaká NoSQL databáze nabízí datový model, který je pro naši aplikaci přirozenější.

Ale zpět k úvodní otázce. NoSQL databáze je software pro perzistenci dat, který je alternativou ke klasickým relačním databázím (nic objevného). NoSQL databází existuje mnoho a dají se rozdělit z mnoha hledisek. Ve velmi specifických případech může dávat smysl vyvinout i vlastní NoSQL databázi (ale myslete na NIH).

Obr. 3 – Relační model

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- NoSQL databáze např. často nemají takové možnosti dotazování jako relační databáze. Zkuste se třeba zamyslet nad tím, jak by ovlivnilo vaši aplikaci, kdyby jedinou podporovanou DB operací bylo uložení řádku a jeho načtení podle nějakého ID (nepřeháním!). To vede k tomu, že je na vaši aplikaci přenášena odpovědnost (např. generování sekundárních indexů), kterou jste dosud považovali za samozřejmou součást databázového enginu. Použití NoSQL databáze tak často znamená kompletní redesign celé aplikace, takže hodně učení a práce pro aplikační programátory (kteří nepracují zadarmo).
- Většina dnešních programátorů vyrůstala ve světě, kde jedinou možností uložení dat byla relační databáze. Umí tak s ní dobře pracovat (v čemž jim pomáhají léty vyladěné tooly), jsou na ni schopni napasovat jakýkoliv model a už při úvodním seznamování s novým projektem jim v hlavě automaticky naskakují tabulky a vazby mezi nimi. Takže stojí za zvážení, zda se opravdu vyplatí investovat čas (peníze) do učení diametrálně odlišných technologií a zda nemůže být ekonomičtější použít léty ověřené relační databáze namísto nové NoSQL databáze.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- NoSQL databáze např. často nemají takové možnosti dotazování jako relační databáze. Zkuste se třeba zamyslet nad tím, jak by ovlivnilo vaši aplikaci, kdyby jedinou podporovanou DB operací bylo uložení řádku a jeho načtení podle nějakého ID (nepřeháním!). To vede k tomu, že je na vaši aplikaci přenášena odpovědnost (např. generování sekundárních indexů), kterou jste dosud považovali za samozřejmou součást databázového enginu. Použití NoSQL databáze tak často znamená kompletní redesign celé aplikace, takže hodně učení a práce pro aplikační programátory (kteří nepracují zadarmo).
- Většina dnešních programátorů vyrůstala ve světě, kde jedinou možností uložení dat byla relační databáze. Umí tak s ní dobře pracovat (v čemž jim pomáhají léty vyladěné tooly), jsou na ni schopni napasovat jakýkoliv model a už při úvodním seznamování s novým projektem jim v hlavě automaticky naskakují tabulky a vazby mezi nimi. Takže stojí za zvážení, zda se opravdu vyplatí investovat čas (peníze) do učení diametrálně odlišných technologií a zda nemůže být ekonomičtější použít léty ověřené relační databáze namísto nové NoSQL databáze.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- NoSQL databáze - Systém databází s možností horizontálního i vertikálního škálování. NoSQL databázový systém nevyužívá tabulky, jako je tomu u relačních databází, v tomto případě je cílem jednoduchost vzhledu a možnost horizontálního i vertikálního škálování. NoSQL je optimalizovanou databází, kdy jsou využívány klíče i hodnoty. Struktura ukládání dat je u NoSQL rovněž odlišná, kupříkladu se jedná o struktury stromovou nebo grafovou. NoSQL databázové systémy získávají na stále větší popularitě, především pokud hovoříme o segmentu real-time web. Dosud ale nejsou tolik rozšířené, což je dáno především absencí podpory transakčního modelu ACID či neúplnou standardizací rozhraní. Navíc jsou pro některé firmy NoSQL databáze dosti nákladné.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- NoSQL je databázový koncept, ve kterém datové úložiště i zpracování dat používají jiné prostředky než tabulková schémata tradiční relační databáze. Motivací k tomuto přístupu mohou být jednoduchost designu, horizontální i vertikální škálovatelnost a jemnější kontrola dostupnosti. Databáze bez SQL jsou často vysoce optimalizovaná úložiště typu klíč-hodnota (ne vždy). Díky odlišné struktuře ukládání dat (např. stromová, grafová) oproti RDBMS, je i algoritmická složitost pro různé operace odlišná. Obecně se vhodnost aplikace daného typu databáze liší podle řešeného problému.
- Segment NoSQL databází v současnosti významně roste a prospívá především v oblasti big data a real-time webu. NoSQL systémům se také občas říká „nejen SQL“ pro zdůraznění faktu, že často umožňují dotazy v SQL (či podobném) jazyce. V kontextu CAP teorému NoSQL úložiště často potlačují konzistenci ku prospěchu dostupnosti a tolerance k narušení sítě.

Transakce

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Transakce je logická jednotka práce sestávající z jednoho nebo více SQL příkazů, které jsou atomické z hlediska zotavení se z chyb. SQL transakce automaticky začíná SQL inicializačním příkazem (SELECT, INSERT). Změny, které realizuje jedna transakce, nejsou viditelné pro ostatní konkurenčně probíhající transakce, pokud daná transakce neskončí.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Transakce

- Transakce může skončit jedním ze čtyř způsobů: – COMMIT končí transakci úspěšně a změny jsou trvale zaznamenány – ROLLBACK přeruší transakci a všechny změny se anulují, databáze se vrátí do stavu před transakcí – v rámci programu - skončí-li program úspěšně, končí úspěšně i SQL transakce – v rámci programu - končí-li program chybou, zruší se transakce SQL.
- Řízení přístupu do databáze SQL definuje dva příkazy na řízení přístupu k tabulkám: GRANT a REVOKE Bezpečnostní mechanizmus je založený na – autorizačních identifikátorech – vlastnictví – privilegiích.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Řešení: Transakce

Transakce lze chápat jako sekvence databázových operací, které splňují ACID:

Atomicita – Všechny operace v rámci transakce se provedou buď všechny nebo žádná.

Konzistence – Před i po vykonání transakce se databáze nachází v konzistentním stavu. Musí být splněna všechna integritní omezení.

Izolace – Jednotlivé transakce jsou izolované. Změny prováděné v průběhu zpracovávání dané transakce nejsou viditelné v ostatních transakcích.

Trvalost (durability) – Po úspěšném ukončení transakce jsou data uložena a nemohou být ztracena.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- V jazyce SQL máme pro práci s transakcemi k dispozici následující příkazy:
- BEGIN; -- *spouští transakci*
- ...příkazy v rámci transakce...
- COMMIT; -- *uloží a ukončí transakci*
- BEGIN;
- ...příkazy v rámci transakce...
- ROLLBACK; -- *vrátí změny provedené transakcí a ukončí ji*

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

1. READ UNCOMMITTED

Transakci je umožněno číst data zapisovaná jinou transakcí, aniž by tato jiná transakce byla ukončená pomocí COMMIT. Čtení takových dat označujeme jako tzv. **dirty read**. Takto přečtená data mohou být nekonzistentní (čas plyne shora dolů):

2. READ COMMITTED

Na této úrovni izolace již nemůže dojít k **dirty read**. Data, která přečteme, byla vždy zapsána transakcí, která byla úspěšně ukončena pomocí COMMIT. Avšak může dojít k tzv. **non-repeatable read**. Pokud v rámci transakce čteme nějaká data vícekrát, může se stát, že se při opakovaném čtení pokaždé nevrátí stejný výsledek. Mezi čteními dat naší transakcí totiž mohla být úspěšně ukončena souběžná transakce, která zapsala/upravila/smazala nějaká data.

3. REPEATABLE READ

Na této úrovni je zajištěno, že nedojde k non-repeatable read. Data, která jednou přečteme, se již nemohou při dalším čtení změnit. Může však nastat tzv. phantom read. Pokud v rámci transakce čteme nějaká data vícekrát, může se stát, že výsledek opakovaného čtení bude obsahovat nové řádky, které při předchozím čtení ve výsledku nebyly. Mezi čteními dat naší transakcí totiž mohla být úspěšně ukončena souběžná transakce, která zapsala nová data.

4. SERIALIZABLE

Nemůže dojít k žádnému výše uvedenému fenoménu, vynucuje ACID.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Procedury a funkce, triggery a sekvence

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- **Funkce, metody a procedury – jaký je mezi nimi rozdíl**
- Všechny tři (funkce, metody i procedury) jsou si poměrně dost podobné – je to POJMENOVANÁ posloupnost příkazů, část programu, kterou můžeme opakovaně volat z jiných částí programu.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Formální pohled

Z formálního hlediska se řídíme názvosloví daného programovacího jazyka.

- **Funkce** je pojem z funkcionálního programování, vyskytuje se v jazycích jako je JavaScript nebo třeba Haskell.
- Jako **metody** označujeme *funkce* v objektově orientovaném programování (OOP), kde dochází k propojení datových struktur a funkčního kódu do objektů. Metody nalezneme v jazycích jako Java, Smalltalk a dalších.
- **Procedura** je pojem známý z procedurálních jazyků. Nalezneme je např. v Pascalu, ale také v některých databázových systémech – tzv. uložené procedury.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Funkce

- Funkce v programování má velmi blízko k funkcím, jak je chápeme v matematice. Funkce má určitý definiční obor (typ vstupních parametrů) a určitý obor hodnot (typ návratové hodnoty).
- Příklad jednoduché funkce v JavaScriptu:
- function sečti(a, b) { return a + b};
- V objektových jazycích jako Java máme sice metody a ne funkce, to nám ale nebrání v nich funkce psát:
- public static int sečti(int a, int b) { return a + b};
- Přestože sečti() je formálně metoda (veřejná a statická), ve skutečnosti se jedná o funkci a třída zde plní pouze úlohu jmenného prostoru (společně s názvem balíčku) a nevytváříme její instance (resp. nemusíme). Jedná se o návrhový vzor Knihovní třída.
- V Javě najdeme takové funkce např. ve třídě java.lang.Math. Následující funkce vrací větší ze dvou čísel:
- int x = java.lang.Math.max(a, b);

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Procedura

- Procedura je zvláštním případem funkce – NEMÁ návratovou hodnotu a nemusí mít ani vstupní parametry. Používají se často při dávkovém zpracování – např. každou hodinu zavoláme proceduru, která zpracuje objednávky, které se nashromáždily v databázi, a předá je do jiného systému.
- Příklad velmi jednoduché procedury v PostgreSQL, která sečte u dosud nesečtených řádků hodnoty sloupců a a b a výsledek uloží do sloupce c:
- CREATE OR REPLACE FUNCTION sčítej()
- RETURNS void AS
- $BODY$
- UPDATE tabulka_součtů
- SET c = a + b
- WHERE c IS NULL
- $BODY$
- LANGUAGE sql VOLATILE;

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Tato procedura je napsaná v jazyce SQL – kromě toho můžeme psát plpgsql, který nám umožní tvořit i velmi složité procedury, případně můžeme použít jazyk C či jiný.
- Výše uvedená procedura (formálně funkce) nemá žádné vstupní parametry ani návratovou hodnotu, je to prostě jen posloupnost příkazů jazyka, kterou jsme si nějak pojmenovali a uložili.
- Zajímavostí je, že procedury můžou mít parametry – a to nejen klasické vstupní, ale i výstupní (nebo vstupně/výstupní). Procedura s výstupním parametrem nám slouží podobně jako funkce – přestože nemá klasickou návratovou hodnotu. Proceduře můžeme např. předat nějaká data a ona nám je upraví.
- Takové procedury si můžeme simulovat i v jazyce Java. Mějme nějakou přepravku (strukturu):

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Metoda

- Metody jsou součástí třídy a (pokud nejsou statické) jsou úzce spjaté s danou instancí třídy (objektem). Nejsou to tedy funkce pracující s nějakými globálními proměnnými a daty, ale mají přístup k proměnným dané instance (v tomto případě osoby).
- Předchozí příklad můžeme přepsat „objektovějším“ způsobem:
- public class Osoba { public String jméno; public String příjmení; public String getCeléJméno() { return jméno + " " + příjmení}};
- Třídu s uloženými procedurami vůbec nepotřebujeme a požadovanou funkcionalitu přesuneme blíže k datům (do třídy Osoba). Všimněte si, že ani NEMUSÍME ukládat vypočtenou hodnotu celéJméno – vypočteme ji až ve chvíli, kdy ji někdo bude potřebovat, tzn. zavolá metodu get CeléJméno().

Co to je trigger ?

- **Co to je trigger ?**
  Trigger je procedura, která je automaticky spouštěna, když se něco stane. Mohu ho definovat na DML (modifikace) i DDL (vytváření) operace. A navíc mohu říct, jestli se má provést před, nebo po dané operaci. (BEFORE, AFTER) Zajímavostí je, že trigger se může jmenovat stejně jako tabulka (avšak není to doporučováno) a že lze definovat libovolný počet triggerů na stejnou tabulku i událost. Avšak pozor: nelze určit pořadí triggerů, ve kterém budou volány a triggery nesmí na sobě vzájemně záviset. V triggerech se také nesmí využívat rekurzivní volání. V těle triggeru lze provádět libovolné SQL dotazy - INSERT, UPDATE, DELETE. Avšak pokud chcete použít dotaz SELECT, musíte ho využít v kombinaci s klíčovým slovem INTO, které zajistí uložení načtených dat do lokálních proměnných, nebo kurzoru.

Triggery a transakce

- Trigger se vždy provádí jako součást transakce, v níž je spuštěn příkaz, který trigger spustil. Chyba, která případně nastane při provádění triggeru, má stejné důsledky, jako chyba ve spouštěcím příkazu - trigger nelze odvolat bez odvolání tohoto příkazu, příkaz nelze odvolat bez odvolání změn provedených triggerem.
- Chyby kategorie "rollback exception condition", pokud se vyskytnou uvnitř triggeru, ignorují vnitřní deklarace handlerů (tj. handlery v triggeru a procedurách volaných tímto triggerem). Zachytit je lze pouze handlerem pro spouštěcí příkaz.
- Uvnitř triggerů se nesmí provádět žádné explicitní transakční příkazy, v opačném případě nastane chybový sqlstate 2D000 (SQ_INVAL_TRANS_TERM).

Analytické nástroje. OLAP

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- **Co a jak zjednodušuje práci**
  Cesta k úspěšné analýze dat začíná už při jejich přípravě. Prvním důležitým krokem je shromáždění dat z podnikových systémů do formy vhodné pro analýzu a reportování. Již při této přípravě je nutné dbát na správné sjednocení různorodých dat a ošetření jejich kvality. Musíme vždy důvěryhodně zdokumentovat vazby na zdrojové systémy, aby výsledky byly vždy věrohodné a doložitelné - pro případ řešení chyb, které se v původních datech vyskytují. Výsledná data se ukládají do relační databáze, kde jsou k dispozici v rozsahu potřebném k operativním analýzám. Takto relačně připravená data slouží k vytváření detailních reportů o výkonnostních parametrech společnosti a jednodušším analýzám založeným na detailních datech. V následujícím zjednodušeném příkladu si však představme požadavek na jiný typ informací: Firma s celostátní působností prodává různé kategorie výrobků. Manažer člení jednotlivé pobočky do regionů. Kategorie prodávaných výrobků mají tři úrovně - kategorie, podkategorie, konkrétní výrobek. Firma má logicky definováno, že prodej se řídí plánem. Reálné plnění plánu je pak dáno součtem za všechny výrobky a všechny pobočky.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- **Jak tyto informace získat?**
  Pro tyto potřeby slouží pokročilé analýzy, které jsou označovány jako multidimenzionální. Manažer může sledovat měřitelné parametry firmy v souvislostech, různých úhlech a různých mírách detailu. K takové analýze je nutná technologie OLAP (on-line analytical processing), která zajistí uložení a předpočítání dat takovým způsobem, že následné dotazy trvají přiměřenou dobu. Technologie OLAP přináší možnost pracovat s daty na sumární úrovni, zde identifikovat problém či zajímavou oblast a tzv. drilováním postupovat k takové úrovni detailu, která je pro naše rozhodování zajímavá. Typickou oblastí, na které se výhody technologie OLAP demonstrují, je analýza prodeje.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Manažer může prostřednictvím nástrojů pro práci s OLAPem kontrolovat, jak se vyvíjí realita proti plánu. Pokud plán není plněn, trendová křivka včas tuto informaci ukazuje, a konec roku nemusí skončit fiaskem. Sumární čísla je možné snadno rozdrilovat o úroveň níže, pak jsou vidět sumární plnění plánu v jednotlivých regionech. Analýza může ukázat, že problém spočívá pouze v jediném z nich, ostatní plán plní. Dalším drilováním se dostane na úroveň poboček, kde identifikuje pět takových, které v problematickém regionu výrazně zaostávají. Systém umožní rozdrilování na úroveň kategorií a subkategorií až na jednotlivé výrobky, které se v daném místě nedaří prodávat. Na základě takové informace pak lze navrhnout a provádět korektivní kroky, které situaci zlepší. Z příkladu je patrné, že data při použití technologie OLAP jsou vlastně uložena ve stromové struktuře - kostce. Zde jsou pro každou úroveň hloubky tohoto stromu předpočítány hodnoty pro příslušný podstrom. Taková hierarchická struktura je definovatelná na většině podnikových dat. Výrazně zpřehledňuje a zjednodušuje analýzu při jejich vyhodnocování. Cenou za toto zjednodušení je čas a místo potřebné k vypočítání a uložení zmíněného stromu. Je-li však kostka naplněna, dokáže přesně odpovídat na velmi komplikované dotazy, zejména provádět srovnání časových řezů omezených složitými podmínkami. Typický komplikovaný dotaz zní: "Kterých deset zákazníků zaznamenalo nejvyšší nárůst obratu s mojí firmou ve srovnání s minulým rokem, a to v okresech, které patří mezi 20 % mých nejméně ziskových?" Výsledek pak může sloužit pro nadstandardní péči o tyto zákazníky. S technologií OLAP je možno dotazovat data z mnoha úhlů pohledu - mnoha dimenzí - a tyto úhly libovolně parametrizovat. Jak jsme si ukázali, pro tyto typy dotazů není vhodné ukládat data v relační podobě.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Analytické nástroje

Až do tohoto okamžiku jsme popisovali způsoby a technologie uložení dat. K práci s nimi (provádění analýz) jsou zapotřebí vhodné nástroje. Volba vhodného analytického nástroje je pro efektivní získávání informací naprosto klíčovou záležitostí. Je nutné zohlednit zejména zkušenosti analytiků, kteří ve firmě již dnes pracují s prostředky, které mají k dispozici - velmi často jsou používány nástroje z rodiny Microsoft Office. Je proto důležité, aby zvolené řešení umožňovalo dostatečnou spolupráci s nástroji, se kterými jsou lidé zvyklí pracovat - to vede k nižším nákladům na adoptování efektivnějšího způsobu práce s informacemi. Zároveň musí být také zajištěna bezpečnost přístupu k informacím, automatizovaná distribuce reportů, kvalitní grafický výstup pro prezentování, případně integrace s firemním intranetem.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

OLAP kostka

- OLAP krychle (online analytical processing) je způsob organizace dat, který rozšiřuje dvojrozměrně tabulkové uspořádání tak, že každá datová dimenze je uložena v jedné ose kostky. Tím překonává některá omezení relačních databází.
- Uspořádání dat do vektorů kostek umožňuje k nim zpětně přistupovat z různých hledisek (dimenzí) stejným způsobem. Odpadá tím na výkonnost systému náročné spojování mnoha tabulek RDBMS. Nicméně fyzické ukládání dat do kostek neumožňuje rychlou editaci, v takovém případě je třeba přepracovat celou kostku. Kostka je tvořena hodnotami, které jsou kategorizovány do dimenzí. Struktura je implementována relačními tabulkami ve hvězdicovém schématu či schématu sněhové vločky. Jedná se typicky o rodič-potomek (parent-child) strukturu, kde rodičovské prvky reprezentují konsolidaci potomků a zároveň ony samy mohou být agregovány do svých rodičovských prvků.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Základní operace s datovými kostkami

- Pro analytické potřeby se provádí následující operace s datovými kostkami, které mají za účel zpracování a projekci dat tak, aby usnadnily jejich pochopení.
- Krájení kostky: omezení jedné nebo více dimenzí na podmnožinu o jednom prvku
- Kostkování: omezení jedné nebo více dimenzí na podmnožinu o dvou a více prvcích
- Roll up a drill down: jedná se o navigaci datovou hierarchií směrem nahoru a dolů
- Pivotování: otáčení kostky za účelem získání jiné perspektivy na vztahy dat
- Agregace: Konsolidace podle vztahů určených vzorci

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Specifika databázových systémů.

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

Interreg
EVROPSKÁ UNIE
Rakousko-Česká republika
Evropský fond pro regionální rozvoj

Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj

VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- 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).
- Přehled přístupu k datům v technologii ASP.NET

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- **Visual Studio 2010**
- Webové aplikace běžně přistupují ke zdrojům dat pro ukládání a načítání dynamických dat. Můžete psát kód pro přístup k datům pomocí tříd z oboru názvů System.Data (obecně označované jako technologie ADO.NET) a z oboru názvů System.Xml. Tento přístup byl běžný v předchozích verzích technologie ASP.NET.
- Avšak technologie ASP.NET také umožňuje provádět vázání dat deklarativně. Pro většinu běžných scénářů není zapotřebí psát žádný kód, tyto mohou být:
- Výběr a zobrazení dat.
- Řazení, stránkování a také cachování dat.
- Aktualizace, vkládání a mazání dat.
- Filtrování dat pomocí parametrů za běhu.
- Vytvoření scénářů předloha-podrobnosti (master-detail) pomocí parametrů.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Technologie ASP.NET obsahuje několik typů serverových ovládacích prvků, které jsou součástí modelu deklarativního vázání dat, včetně ovládacích prvků zdroje dat, ovládacích prvků vázání dat a rozšiřujícího ovládacího prvku pro dotazy. Tyto ovládací prvky zvládají základní úlohy, které jsou vyžadovány modelem bezstavového webu s cílem zobrazit a aktualizovat data na webových stránkách technologie ASP.NET. Ovládací prvky umožňují přidat na stránku funkcionalitu vázání dat, aniž by bylo nutné porozumět podrobnostem životního cyklu požadavku stránky.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- **Databázové objekty**
- Pojem „databáze“ je často zjednodušován na to, co je ve skutečnosti databázový systém (databázový stroj) nebo též systém řízení báze dat. Ten neobsahuje pouze tabulky – ty jsou jedny z mnoha tzv. databázových objektů (někdy též databázových entit). Pokročilejší databázové systémy dále obsahují:
- pohledy neboli views – SQL příkazy, pojmenované a uložené v databázovém systému. Lze z nich vybírat (aplikovat na ně příkaz SELECT) jako na ostatní tabulky.
- indexy pro každou tabulku. Klíče jsou definovány nad jednotlivými sloupci tabulek (jeden klíč jich může zahrnovat i více) 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.
- spouště neboli Trigger – mechanizmus nad jednotlivými řádkami tabulky (případně samotnou tabulkou), který se vyvolá po změně, odstranění nebo přidání řádky, případně smazání tabulky a provede předprogramovanou akci (například kontrolu integrity dat, doplnění hodnot...)

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- **Úvod k databázovým technologiím**
- S příchodem mikropočítačů a jejich vstupem do života každého z nás se zdá, že vzrůstá zájem o počítačově zpracovávaná data, a to ne pouze ve smyslu něco si vypočítat, ale především ve smyslu něco se dozvědět. Vysvětlení je zcela pochopitelné, uvážíme-li, že i ty nejdokonalejší počítačové hry časem omrzí a že programovat v Basicu výpočty typu řešení soustavy dvou rovnic o dvou neznámých nepřináší až tak velké uspokojení.
- Člověk by si rád uspořádal a vyhledával některé informace, které často používá a které se v průběhu používání mění. Za předpokladu vhodných prostředků pro ukládání takových informací ve formě dat by byl ochoten prosedět několik večerů u klávesnice a příslušná data si sám vyrobit, koupit nebo vyměnit s jiným podobným nadšencem. Příkladem může být knihovna, úřad, zpracovávání letenek, městské muzeum, nemocnice, podnik apod.

Interreg Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

- Naším cílem bude ukázat si tzv. databázovou technologii zpracování dat. Databázová technologie je soubor pojmů, prostředků a technik sloužící pro vytváření informačních systémů (IS). Na nejzákladnější úrovni si můžeme představit architekturu IS s databází takto: data jsou organizována v databázi (DB), jsou řízena balíkem programů, který se nazývá systém řízení bází dat (SŘBD). Databáze a SŘBD tvoří tzv. databázový systém (DBS). V naší terminologii můžeme jednoduše psát DBS = DB + SŘBD. IS využívá data z DBS buď přímo, nebo je zpracovává aplikačními programy.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Geografický informační systém

(GIS; anglicky Geographic information system) je geografický informační systém, který umožňuje ukládat, spravovat a analyzovat prostorová data.

Většina objektů a jevů reálného světa se vyskytuje na některém místě zemského povrchu (např. strom, dům, řeka) nebo má vztah k některému místu na zemském povrchu (občan má někde trvalé bydliště, výrobek byl vyroben v určité továrně). Zároveň se tyto objekty vyskytují v daném prostoru společně s mnoha dalšími objekty a navzájem se ovlivňují. Proto znalost umístění a vzájemných prostorových souvislostí mezi objekty je velmi významná a může sehrát důležitou roli v řadě oborů lidské činnosti, od návrhu umístění jaderné elektrárny až po návrh obchodní sítě a vyhodnocování její úspěšnosti.

Prakticky to znamená, že v našich datech v počítači musíme mít zaznamenáno obojí současně, tj. jak vlastní údaje o objektu, tak údaje o jeho poloze. Tomuto typu dat říkáme geografická (nebo prostorová) data a počítačovému systému, který umožňuje ukládat a využívat taková data, říkáme geografický informační systém, zkráceně GIS.

Interreg
Rakousko-Česká republika
Evropský fond pro regionální rozvoj
Europäische Union
Evropská unie
Europäischer Fonds für regionale Entwicklung
Evropský fond pro regionální rozvoj
VŠTE
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Základní komponenty geografických informačních systémů

Základní komponenty geografických informačních systémů jak uvádí Vít Voženílek ve své knížce Geografické informační systém I. Pojetí, historie, základní komponenty (1998) jsou:

Hardware

Software

Data

Obsluhující personál

Pro efektivní práci systémů je nezbytná jejich vyváženost. Hardware, neboli technické vybavení, představuje technickou základnu geografických informačních systémů. Software, neboli programové vybavení, představuje soubor programů vykonávající veškeré operace systému. Data jsou klíčovým prvkem každého geografického informačního systému. GIS je z pohledu organizační struktury skutečným systémem. Jeho fungování je souhrnem činností, které zabezpečují jednotlivé funkce systému.