



# Integrita, konzistence a transakce

Business vrstva informačního systému

**doc. Ing. Radek Burget, Ph.D.**

[burgetr@fit.vutbr.cz](mailto:burgetr@fit.vutbr.cz)

# Integrita a konzistence

# Databázová integrita

- Databáze vyhovuje zadaným pravidlům – ***integritním omezením (IO)***. Tato integritní omezení bývají nejčastěji součástí definice databáze a za jejich splnění zodpovídá **system řízení báze dat (SŘBD)**
- Mohou být zadána výrazem (*deklarativně*) nebo programem (*procedurálně*)
- Integritní omezení se mohou týkat *jednotlivých hodnot* vkládaných do polí databáze (například známka z předmětu musí být v rozsahu 1 až 5)

# Databázová integrita

- Může jít o podmínku na *kombinaci hodnot* v některých polích jednoho záznamu (například datum narození nesmí být pozdější než datum úmrtí).
- Může se 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).

# Integrita datovým typem

- *Datový typ* je množina hodnot spolu s operacemi, které je možné nad těmito hodnotami provádět.
- Je vlastností jisté části modelu (proměnné, části jiného datového typu apod.) a *omezuje* její použití (jde vlastně o *integritní omezení* části modelu). Omezuje je tak, že tato část modelu může:
  - *nabývat pouze jisté množiny hodnot a*
  - *může s ní být prováděna pouze jistá omezená množina operací.*
- Výhodou zavedení datového typu pro jistou část modelu je zejména možnost kontrolovat (a to nejčastěji předem), zda se s touto částí zachází korektně (zda ukládaná hodnota je správná a použitá operace správně použita).

# Integrita v relačních databázích

# Primární klíč

- Pole nebo kombinace polí, jednoznačně identifikující každý záznam v relaci. Žádné pole, které je součástí primárního klíče, nesmí obsahovat nedefinovanou hodnotu. Každá tabulka má mít definovaný právě jeden primární klíč (*entitní integrita*).
- *Primární klíč* má dvě základní vlastnosti: *jedinečnost* v rámci tabulky a *definovanou hodnotu*.

# Primární klíč

- Databázový systém by měl být navržen a udržován tak, aby se primární klíč založeného záznamu nikdy nemusel měnit.
- Typickým příkladem primárního klíče je katalogové číslo u výrobků apod. Pokud u záznamu neexistuje žádný přirozený primární klíč, nebo je jeho použití problematické (změny, GDPR, apod.), používá se obvykle primární klíč, které záznamu přidělí SŘBD.



# Referenční integrita

- Častou v relačních databázích je tzv. *referenční integrita*.
- Jedná se o požadavek, aby pro pole záznamu, jež má obsahovat odkaz na jiný záznam někde v databázi, takový odkazovaný záznam skutečně existoval, tedy aby takový odkaz nevedl *do prázdna* a nejednalo se o tzv. *databázového sirotka*.
- Další integritní omezení lze definovat za pomoci tzv. *triggerů*. Jde o komplexnější (programově definované) definice kontrol, jež se budou provádět při každém pokusu o zápis záznamu do databáze.

# Referenční integrita

- Referenční integrita se definuje *cizím klíčem*, a to pro dvojici tabulek nebo nad jednou tabulkou, která obsahuje na sobě závislá data (například stromové struktury).
- Tabulka, v níž je pravidlo uvedeno, se nazývá podřízená tabulka (používá se také anglický termín *slave*). Tabulka, jejíž jméno je v omezení uvedeno, je nadřízená tabulka (*master*).

# Konzistence

- DB musí splňovat všechna *integritní omezení* (IO)
- Udržování *interní konzistence* (redundantních, vícenásobně uložených/replikovaných dat v distribuovaných systémech)
  - Máme více lokálních kopií dat (replikace)
  - Jak zajistíme, že jsou kopie vzájemně konzistentní?
- Dodržování *pravidel daných modelovaným systémem*

# Konzistence replikovaných dat

- Data-centrické modely
  - Několik procesů, jedna skupina dat – distribuovaný výpočet
  - Může být náročná na výměnu dat (objem zpráv)
  - Sekvenční konzistence – výsledek stejný, jako sekvenční provedení operací
- Klient-centrické modely
  - Jeden proces, několik skupin dat
  - Např. eventual consistency

# Transakce

# Motivační otázky vzniku transakcí

- Co se stane, pokud dojde k poruše během práce s důležitými zdroji dat?
- Které operace prováděné nad daty systém stihl před poruchou skutečně provést a které ne?
- Co se stane, když více uživatelů současně bude modifikovat tentýž údaj?
- Budou údaje v databázi stále smysluplné ?
- Hledáním odpovědí a jejich aplikací při zajištění spolehlivosti se zabývají ***transakční modely*** a celý obor ***transakčního zpracování***.

# Pojem transakce

- **Transakce** představuje jednotku práce vykonávanou **souvislým a bezpečným způsobem nezávisle na jiných transakcích**.
  - *Databázové transakce* – omezeno pouze na DB vrstvu (zajišťuje DB server)
  - *Business transakce* – jednotka business logiky, nutno řešit na business vrstvě
    - Podpora na pokročilejších platformách pro implementaci IS, např. Java (ne PHP, Python)
- Transakce má dva základní účely:

# Pojem transakce

1. Poskytnout bezpečnou jednotku práce, která dovoluje správné **zotavení z poruch** a udržuje systém v konzistentním stavu i v případě poruchy systému, když je zastaveno provádění (úplně nebo částečně) a některé operace zůstávají nedokončené nebo v nejistém stavu
2. Poskytnout **izolaci souběžně prováděných operací**. Pokud tato izolace není poskytnuta, výstupy jsou potenciálně chybové (souběh).



# Pojem transakce

- **Skupina operací** (akcí) prováděných jako celek (buď celá dávka nebo nic)
- Modelování stavu popisovaného výseku reálného světa
  - popis a provádění nerozlučných příkazů
  - první historické zmínky – 60. léta
  - důležitý pojem v oblasti databází
- **Transakce** je speciální druh programu, který je spouštěn v aplikaci **OLTP** (On Line Transaction Processing)

# System pro zpracování transakcí - TPS

- Systém (platforma, databázový systém) **podporující provádění transakcí** – transakční systém
- Zajišťuje speciální **vlastnosti transakcí** (atomičnost, nezávislost, trvanlivost)
- Angl. **Transactional Processing System** (zkratka **TPS**)

# Základní vlastnosti transakce

- Žádoucí vlastnosti transakcí jsou:
  - **Atomičnost** (**A**tomicity) – každá transakce je dokončena zcela nebo vůbec
  - **Konzistence** (**C**onsistence) – databázová konzistence (správná reflexe stavu reálného světa a dodržování omezujících pravidel pro hodnoty)
  - **Izolovanost** (**I**solation, **I**ndependence) – souběžné provádění má totožný efekt jako sekvenční
  - **Trvanlivost** (**D**urability) – odolnost proti ztrátě již dokončených změn
- V databázové praxi se pro tyto vlastnosti užívá akronym **ACID**.

# Kdo co zajišťuje

- Programátor je zodpovědný za vytváření konzistentních transakcí. TPS považuje
  - ***konzistenci*** za zajištěnou programátorem (případně částečně systémem pro kontrolu integritních omezení)
- a zaopatřuje
  - ***atomičnost***,
  - ***izolovanost*** a
  - ***trvanlivost***,
- což jsou vlastnosti nutné pro zajištění ***souběžného spouštění konzistentních transakcí*** a případného ***zotavení z chyb či poruch***.

# Transakce v business vrstvě IS

- Jednotlivé business operace obvykle představují elementární transakce
  - Podpora v některých SW frameworkcích, např. Java EE
  - Např. převod zboží mezi sklady
- Někdy je nutno zajistit vlastnosti transakce pro skupinu transakcí
  - Např. kino: výběr vstupenky – výběr místa – platba
  - Nutno zajistit na aplikační úrovni
  - Vede na hierarchické transakce

A to je vše!

Dotazy?

