

Databázové systémy

Obmedzenia a spúšťače
(constraints & triggers)

Integritné obmedzenia a spúšťače

- súčasť SQL štandardu
 - dostupné implementácie sa líšia

Integritné obmedzenia a spúšťače

- súčasť SQL štandardu
 - dostupné implementácie sa líšia
- Integritné obmedzenia
 - obmedzujú prípustné stavy databázy
 - statické
- Spúšťače
 - sledujú zmeny v databáze, kontrolujú podmienky a inicializujú akcie
 - dynamické

Integritné obmedzenia

- Kladú obmedzenia na prípustné dáta
 - “nadstavba” nad štruktúrou a dátovými typmi
- Príklady
 - $0.0 < VSP \leq 4.0$
 - `capacity < 300`
 - `was_tasty`: 'T', 'F', NULL
 - `restaurant = 'horna' ==> was_tasty = 'T'`

Integritné obmedzenia - motivácia

- Zachytiť chyby od používateľa (INSERT)
- Zabezpečiť korektnosť manipulácie (UPDATE)
- Zabezpečiť konzistenciu dát
- Poskytnúť DBMS dodatočné informácie o dátach a ich štruktúre
 - uloženie dát
 - spracovanie dopytov

Integritné obmedzenia

- NOT NULL
- Obmedzenia na kľúčoch (key-constraints)
- Referenčná integrita (cudzie kľúče)
- Obmedzenia atribútov
- Obmedzenia n-tice
- Všeobecné tvrdenia
 - ľubovoľné SQL
 - zatiaľ neexistuje implementácia

Deklarácia a kontrola obmedzení

- Deklarácia
 - pri tvorbe schémy – skontrolované po hromadnom importe dát
 - neskôr – kontrola podľa aktuálneho stavu DB
- Kontrola
 - po každej modifikácii údajov
 - optimalizovane – len relevantné obmedzenia
 - po ukončení transakcie (odložená kontrola)
 - deferred

DEMO

Pre úplnosť: referenčná integrita

- Čo skončí s chybou
 - INSERT INTO referencing_table
 - UPDATE referencing_attribute
- Zmena v referencovanej tabuľke
 - NO ACTION (default, deferred)
 - RESTRICT
 - SET NULL
 - CASCADE

Polymorfické vzťahy

- Príklad: komentovanie na stackoverflow
- Komentár môžem pridať
 - K otázke
 - K odpovedi
- Otázka a odpoveď sú *komentovateľné*
- Tabuľka komentárov
 - commentable_id
 - commentable_type

Polymorfické vzťahy

- Podpora ORMs/frameworkov
- Bez referenčnej integrity na strane databázy
 - Dá sa zabezpečiť triggermi (o tom zachvíľu)
- Alternatíva: mať v tabuľke N cudzích kľúčov a N-1 bude vždy NULL

Spúšťáče / triggers

- Pravidlá Udalosť – Podmienka – Akcia
 - ak nastane *udalosť*, skontroluj *podmienku* a v prípade potreby vykonaj *akciu*
- Príklady
 - UPDATE capacity < 50 ==> zmen was_tasty na 'T'
 - UPDATE vsp < 2 ==> vymaz všetky jeho obedy
 - INSERT capacity > 500 ==> vyhlás chybu

Spúšťáče - Motivácia

- presun logiky z aplikácie do DBMS
- pre vynútenie/kontrolu obmedzení
 - možnosť reakcie na porušenie obmedzení
 - napr. MySQL CHECK constraint
 - Od v8.0.16 už nejaký CHECK podporujú

Spúšťáče všeobecne

Create Trigger name

Before|After|Instead Of events

[referencing variables]

[For each row]

when (condition)

action

Pozor

- For each row vs. jedno odpálenie
- before/after/instead
- viacero triggerov naraz
 - poradie?
- akcia triggera môže odpáliť ďalší trigger
 - cykly?
- Podmienka vo WHEN alebo v ACTION

Implementácie

- PostgreSQL
 - podporuje takmer plný štandard
 - Samotná *action* funkcia musí byť naskriptovaná pred definovaním triggru
 - PL/Python, PL/Perl, PL/pgSQL, PL/Tcl
- SQLite
 - row-level only
 - okamžitá aktivácia (aj keby som updatoval 100 riadkov naraz)
- MySQL
 - ako SQLite
 - iba jeden trigger pre jeden typ udalosti

CREATE TRIGGER

- WHEN
 - BEFORE
 - AFTER
 - (INSTEAD OF)
- trigger_event (v zmysle operácie nad tabuľkou)
 - INSERT
 - UPDATE
 - DELETE/TRUNCATE
- referencovanie
 - OLD.col_name
 - NEW.col_name

DEMO

Zhrnutie

- Je dobré mať stráženie konzistencie dát čo najbližšie pri dátach
 - Ak viac aplikácií zdieľa db, tak to žiadna nepokazí
- Constraints vs Triggers
 - statické vs. dynamické
- Pozor na komplikovanú logiku v triggroch
 - Udržiavateľnosť, prehľadnosť...