Vysoké učení technické v Brně
Fakulta informačních technologií

Databázové systémy
2019/2020

Oficiálna dokumentácia projektu

Téma: Vegánska reštaurácia

Eva Moresová (xmores01)
Jiří Žák (xzakji02)

22. apríla 2020

1 Popis

Vytvorená databáza je navrhnutá pre systém reštaurácie, ktorý umožňuje zaznamenávať objednávky a rezervácie, vystavovať účtenky a spravovať položky menu.

2 Implementácia

2.1 Vzťah generalizácie/špecializácie

Databáza obsahuje jeden vzťah generalizácie/špecializácie a to medzi entitami Jedlo, Nápoj (podtypy) a Položka menu (nadtyp).

Špecializácie Jedlo a Nápoj sú disjunktné, pretože Položka menu môže patriť iba do jednej z nich a špecializácia je úplná. K jednotlivým špecializáciam nevedú vzťahy, vedú iba k nadtypu a každá z entít obsahuje relatívne malý počet atribútov. Preto bola pre implementáciu tohto vzťahu zvolená varianta, kedy je všetko uložené v jednej tabuľke. Špecializácia je rozlíšená podľa prázdnych hodnôt.

2.2 Triggery

Skript obsahuje 3 triggery. Sú využité na výpočet stĺpcov s odvodenými hodnotami a implementáciu integritných obmedzení.

2.2.1 `generate_uzivatel_id`

Trigger pre automatické generovanie primárneho kľúča (ID) v tabuľke `uzivatel`. Trigger vloží do stĺpca `uzivatel.id` číslo, ktoré je o jedna vyššie než najvyššie nájdené. Trigger sa spúšťa pred operáciou `INSERT` v prípade, že je pri vkladaní hodnota primárneho kľúča nedefinovaná.

2.2.2 `trig_check_rezervation`

Trigger slúži na kontrolu správneho formátu telefónneho čísla, e-mailu a mena v tabuľke `rezervace`. Trigger sa spúšťa vždy pred použitím operácií `INSERT` alebo `UPDATE` na danej tabuľke.

2.2.3 `vypocet_ceny_objednavky`

Trigger pre aktualizáciu ceny objednávky v tabuľke `objednavka`. Trigger spúšťa procedúru, ktorá vypočíta novú cenu objednávky a starú hodnotu aktualizuje. Keďže táto procedúra číta údaje z tabuľky, ktorá je zároveň menená, bolo potrebné použiť zložený trigger. Spúšťa sa vždy pred operáciou `INSERT` do tabuľky `objednavka_polozka`.

2.3 Procedúry

Skript obsahuje 2 procedúry. Obe procedúry využívajú cursor a dátový typ, ktorý odkazuje na riadok tabuľky.

2.3.1 `set_objednavka_cena`

Táto procedúra sa využíva pre aktualizáciu ceny v tabuľke `objednavka` a spúšťa ju trigger `vypocet_ceny_objednavky`. Procedúra pomocou cursoru a cyklu nájde všetky položky, ktoré objednávka obsahuje a sčíta ich ceny. Po prehľadaní tabuľky aktualizuje cenu v tabuľke `objednavka`.

2.3.2 proc_delete_user

Táto procedúra sa využíva na zmazanie užívateľa podľa ID. Procedúra najskôr podľa zadaného ID vyhľadá, či užívateľ existuje. Ďalej potom pomocou cursoru a cykla nájde vo všetkých odkazovaných tabuľkách jeho ID a zmení ho na ID majiteľa. Týmto bude možné užívateľa odstrániť bez porušenia integrity.

2.4 Explain plan

Explain plan zobrazuje postup databázy pri vykonávaní `SELECT`u, poradie a spôsob prechodu nad dátami v tabuľkách a jeho časovú, procesorovú a pamäťovú náročnosť. Bol použitý na `SELECT` príkaz, ktorý pre každého užívateľa v databázi vypíše sumu cien z tabuľky `zaznam_platby` za určité obdobie. Prvýkrát bol Explain plan vykonaný bez použitia indexu, ktorý by urýchlil hľadanie v databáze. Jeho priebeh:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	82	5 (20)	00:00:01
1	HASH GROUP BY		1	82	5 (20)	00:00:01
2	NESTED LOOPS		1	82	4 (0)	00:00:01
3	NESTED LOOPS		1	82	4 (0)	00:00:01
* 4	TABLE ACCESS FULL	ZAZNAM_PLATBY	1	35	3 (0)	00:00:01
* 5	INDEX UNIQUE SCAN	SYS_C00986727	1		0 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	UZIVATEL	1	47	1 (0)	00:00:01

Pred vykonaním druhého Explain plan bol vytvorený index, ktorý odkazuje na dátum v tabuľke `zaznam_platby`. Druhý Explain plan prebehne už s použitím vytvoreného indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	82	4 (25)	00:00:01
1	HASH GROUP BY		1	82	4 (25)	00:00:01
2	NESTED LOOPS		1	82	3 (0)	00:00:01
3	NESTED LOOPS		1	82	3 (0)	00:00:01
4	TABLE ACCESS BY INDEX ROWID BATCHED	ZAZNAM_PLATBY	1	35	2 (0)	00:00:01
* 5	INDEX RANGE SCAN	ZAZNAM_DATUM_IDX	1		1 (0)	00:00:01
* 6	INDEX UNIQUE SCAN	SYS_C00986727	1		0 (0)	00:00:01
7	TABLE ACCESS BY INDEX ROWID	UZIVATEL	1	47	1 (0)	00:00:01

Z výsledkov je možné pozorovať, že `SELECT` s indexom je vykonaný rýchlejšie a s menšou záťažou počítača, pretože namiesto sekvenčného prechodu tabuľkou `zaznam_platby` sa využíva indexovaný prístup nad stĺpcom `datum`.

2.5 Materializovaný pohľad

Materializovaný pohľad je databázový objekt na zrýchlenie prístupu k dátam, vďaka uloženiu výsledku po vykonaní dotazu. V skripte bol vytvorený pohľad pre dotaz, ktorý zobrazí názov položky a názov alergénu, ktorý obsahuje, z tabuliek druhého člena tímu, pre ktoré boli definované prístupové práva. Pohľad je nakoniec testovaný pomocou príkazov `INSERT` a `COMMIT`.