

Vysoké učení technické v Brně

Fakulta informačních technologií

DATABÁZOVÉ SYSTÉMY

2016/2017

Konceptuální model

Zadání: Půjčovna kostýmů

Jan Kotas (xkotas07)

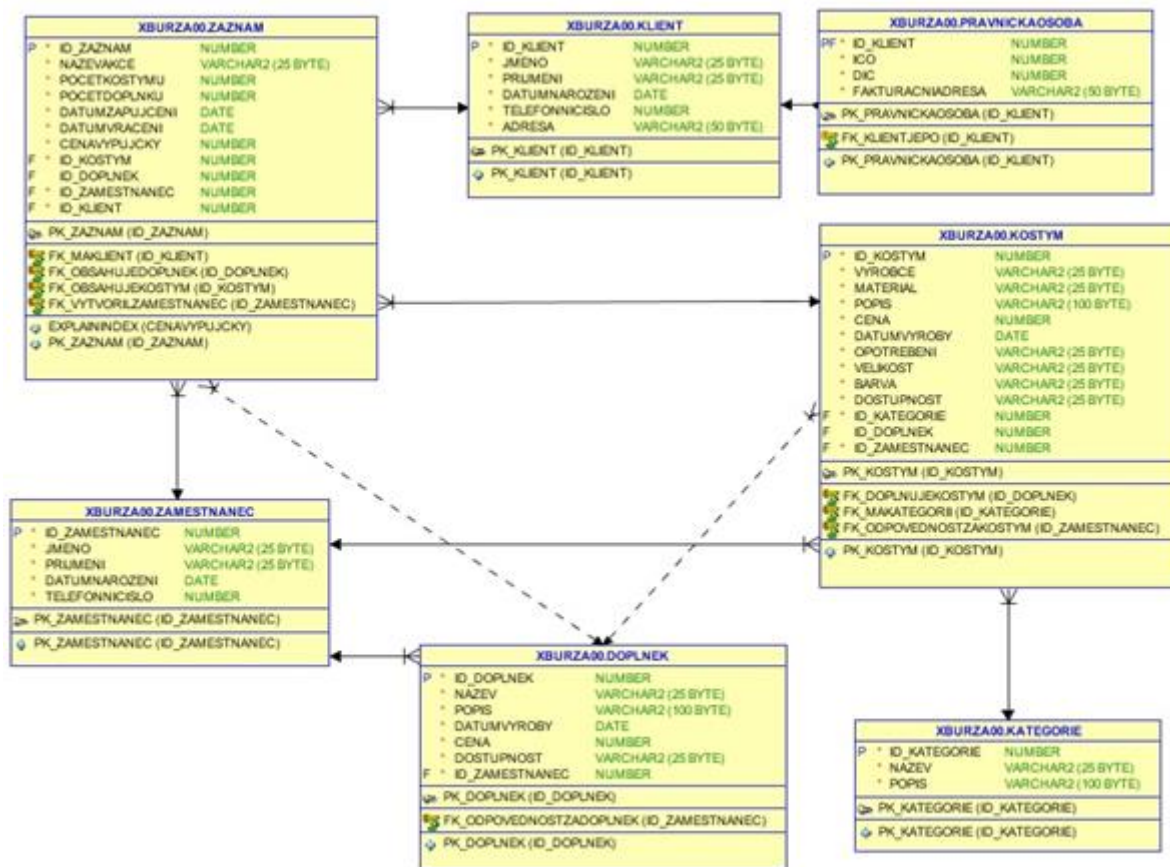
Matúš Burzala (xburza00)

29.4.2017

Zadání

Navrhněte informační systém půjčovny kostýmů, která nabízí kostýmy pro různé akce. U kostýmů je možné dohledat od jakého výrobce je, z jakého materiálu, jeho stručný popis. Každý kostým spadá do určité kategorie, podle možnosti využití (hororová maska, maškarní oblek, převlek zvířete, dětský kostým atd.). U některých kostýmů půjčovna nabízí více velikostí a více barev, zároveň lze dohledat stáří a opotřebení konkrétního kostýmu. U různých typů kostýmů půjčovna nabízí možnost zapůjčení doplňků (u některých typů kostýmů je doporučeno i více doplňků). Samozřejmě i o doplňcích systém poskytuje více informací (název, stáří, popis možného využití, datum výroby, atd.). Při zapůjčení doplňků/kostýmů je vytvořen záznam na kterou akci a kdy byly zapůjčeny, zda již byly vráceny, samozřejmě základní informace o člověku, který si kostýmy zapůjčil a celková cena výpůjčky. Klient má možnost si vyhledat svoje výpůjčky, jak pro kontrolu, kdy má kostýmy vrátit, tak i pro historii svých výpůjček. Půjčovna má více zaměstnanců, u kterých je možné dohledat, jaké výpůjčky zprostředkovali a zároveň každý kostým a doplněk má odpovědnou osobu, která konkrétní zboží spravuje.

Schéma databáze



Generalizace/specializace

Vztah generalizace/specializace jsme použili u tabulky reprezentující klienta. V případě, že je klientem právnická osoba, je nutné do systému ukládat informace jako IČO, DIČ a fakturační adresa. Proto je vytvářena tabulka PravnickaOsoba, která je specializací tabulky Klient.

Implementace

Skript nejdřív vytvoří tabulky reprezentující základní objekty databáze, dále jsou pak nastaveny integritní omezení (primární a cizí klíče), triggerů pro nastavování hodnot některých atributů. Do tabulek jsou pak vloženy ukázkové data, nad kterými je provedeno několik dotazů SELECT. Skript dále obsahuje procedury, funkce EXPLAIN PLAN, definici přístupových práv a materializovaný pohled.

Triggery

Ve skriptu se nachází několik triggerů implementovaných formou SEQUENCE, které automaticky doplňují hodnoty id. Dále je pak implementován trigger na kontrolu formátu telefonního čísla vkládaného do tabulky Klient, který ověří délku zadaného čísla a dále první tři znaky, které reprezentují předvolbu.

Procedury

Podle zadání skript obsahuje implementaci dvou netriviálních procedur. První celkovaUtrataKlienta() sečte celkovou útratu klienta, tedy projede všechny záznamy daného klienta a vrátí součet cen výpůjček. Druhá procedura podilPraceZamestnance() vrátí procentuální hodnotu podílu práce daného zaměstnance. Tato hodnota je výsledkem podílu počtu kostýmů a doplňků, které spravuje daný zaměstnanec a počtu všech kostýmů a doplňků v systému.

EXPLAIN PLAN

EXPLAIN PLAN jsme použili na demonstraci toho, jak databáze zpracovává dotaz SELECT. Jako vzorový příklad jsme použili dotaz spojující dvě tabulky Kostým a Záznam, s využitím agregační funkce GROUP BY. Následně jsme vytvořili index a opětovně použili uvedený dotaz s tím, že jsme jej donutili daný index použít.

Výstup EXPLAIN PLAN bez použití indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		7	735	4 (25)	00:00:01
1	HASH GROUP BY		7	735	4 (25)	00:00:01
2	NESTED LOOPS		7	735	3 (0)	00:00:01
3	NESTED LOOPS		8	735	3 (0)	00:00:01
4	TABLE ACCESS FULL	ZAZNAM	8	416	3 (0)	00:00:01
*5	INDEX UNIQUE SCAN	PK_KOSTYM	1		0 (0)	00:00:01
*6	TABLE ACCESS BY INDEX ROWID	KOSTYM	1	53	0 (0)	00:00:01

Výstup EXPLAIN PLAN s použitím indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		7	735	6 (17)	00:00:01
1	HASH GROUP BY		7	735	6 (17)	00:00:01
*2	HASH JOIN		7	735	5 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID BATCHED	KOSTYM	7	371	2 (0)	00:00:01
4	IDEX FULL SCAN	EXPLAININDEX	7		1 (0)	00:00:01
5	TABLE ACCESS FULL	ZAZNAM	8	416	3 (0)	00:00:01

Po použití indexu se zvýšila hodnota cost, ale snížil se počet operací. První dvě operace jsou v obou případech stejné. SELECT STATEMENT znamená, že se provedl dotaz, HASH GROUP BY indikuje seskupování položek dle hashovacího klíče. V případě nepoužití indexu následuje dvakrát po sobě operace NESTED LOOPS, která znamená spojení, kdy se pro každou položku první tabulky prochází všechny řádky druhé tabulky. Další operace je TABLE ACCESS FULL, která představuje průchod celou tabulkou bez použití indexu. Následuje operace INDEX UNIQUE SCAN, která představuje přístup k tabulkám přes B-STROM (B-Tree Index Access) a hledání rovnosti podle hodnot PK_KOSTYM. Poslední operace je TABLE ACCESS BY INDEX ROWID znamenající přímý přístup k tabulkám a dohledání po průchodu indexem. V případě, když byl použit index, následovala po prvních dvou operacích operace HASH JOIN, značící párování záznamů spojovaných tabulek přes hash klíče spojení. Dále operace TABLE ACCESS BY INDEX ROWID BATCHED, která představuje přímý přístup k tabulkám a dohledání hodnot po průchodu indexem. Po ní následovala operace IDEX FULL SCAN představující přístup k tabulkám přes B-STROM a výpis hodnot z indexovaných sloupců. Poslední operace TABLE ACCESS FULL představuje přechod celou tabulkou bez použití indexu.

Definici přístupových práv

Při definici přístupových práv si můžeme představit, že druhý člen týmu reprezentuje běžného zaměstnance, který pracuje s databází. Jsou mu přiděleny práva ke všem tabulkám, kromě práv k tabulce Zaměstnanec a proceduře podilPraceZamestnance(). Důvodem je fakt, že běžný zaměstnanec nemůže mít přístup k informacím o ostatních zaměstnancích.

Materializovaný pohled

Materializovaný pohled, patřící druhému členovi týmu v naší implementaci, poskytuje uživateli informace o dostupnosti jednotlivých kostýmů. Nejdřív je vytvořen LOG nad tabulkou Kostym a dále pak samotný materializovaný pohled kostymDostupnost, který je

reprezentován výslednou tabulkou, obsahující jednotlivé dostupnosti a jejich počty výskytů v databázi. Ukázka funkčnosti je pak demonstrována pomocí příkazu SELECT a následným vložením nových hodnot do tabulky. Následuje opětovný příkaz SELECT, který demonstruje, jak se změny projeví v materializovaném pohledu.

Závěr

Skript byl vypracován a otestován v prostředí Oracle na školním serveru Oracle 12.c. K vypracování byla použita aplikace SQL Developer.