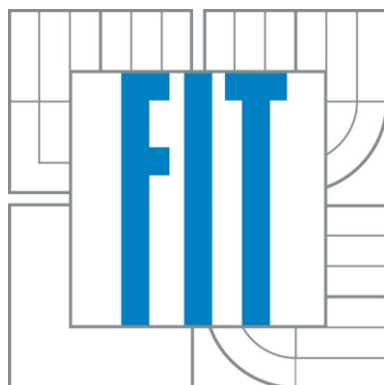


VYSOKÉ UČENIE TECHNICKÉ V BRNE

Fakulta informačných technológií



Databázové systémy

2017/2018

Zadanie – Řetězec multikin

Peter Andris (xandri04)

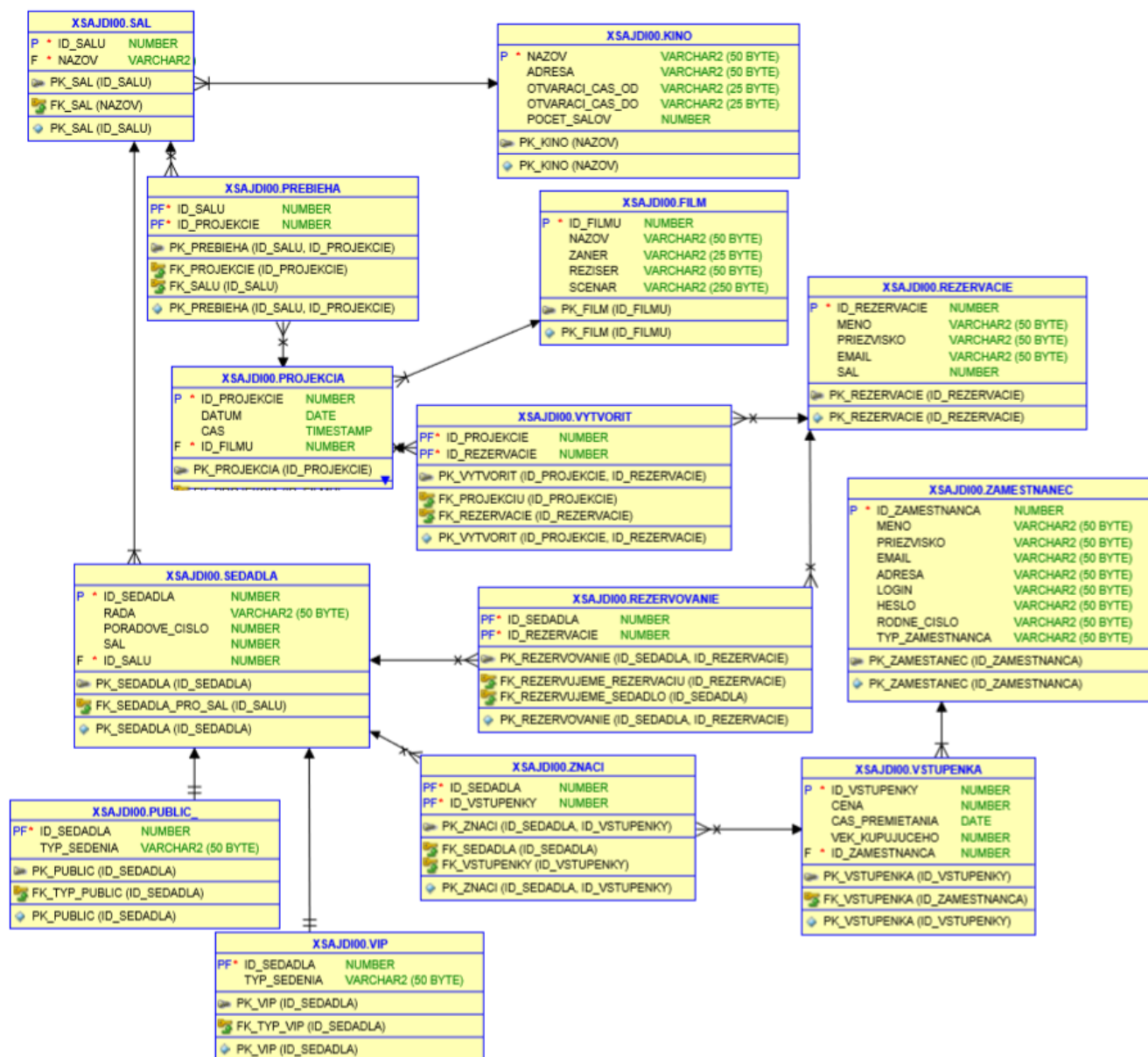
Michal Šajdík (xsajdi00)

27.4.2018

1. Zadanie

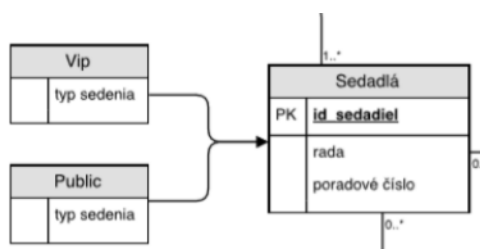
Vytvořte návrh informačního systému pro řetězec multikin. Řetězec vlastní několik multikin a každé multikino obsahuje několik promítacích sálů. V sálech probíhají projekce filmů. Systém musí umožnit klientům připojeným přes webové rozhraní vyhledávání projekcí podle názvu filmu, žánru, kina, apod. Systém umožní klientovi zarezervovat si na zvolenou projekci konkrétní sedadla (jeden klient si může zarezervovat i více sedadel)(není třeba uvažovat překročení kapacity sedadel promítacího sálu), či dříve zadanou rezervaci zrušit. Cena za vstupenku se liší podle toho, zda je divák dítě, mládež, dospělý, či důchodce, zda se jedná o dopolední, odpolední, či večerní představení, atd. Pokladní musí mít možnost prodat vstupenku, ať už při koupi na místě, tak i na základě webové rezervace. Není třeba pamatovat si, který prodavač prodal kterou vstupenku. Vedoucí pracovníci jednotlivých poboček mají možnost zjistit tržby jednotlivých multikin, filmů, apod. Zákazníci si mohou přes webové rozhraní prohlížet program kin.

2 Finálne schéma databázy



2.1 Generalizácia / Špecializácia

Generalizáciu / Špecializáciu sme v našej databáze využili pri tabuľke **Sedadlá**. Rozdelili sme ju tak, že sedadlá sú typu **Vip** alebo **Public**. **Vip** sedadlá majú svoje špecifické sedenie oproti sedadlám typu **Public**.



3. SQL Skript

Skript na začiatok vytvorí databázové objekty pomocou DROP a následne sa vytvoria tabuľky, priradia sa primárne a cudzie kľúče. Potom naplníme tabuľky dátami. V ďalšom kroku sú vytvorené SELECT príkazy, trigger, procedúry, materializovaný pohľad, použitý EXPLAIN PLAN a pridelenie práv druhému členovi tímu.

3.1 Triggery

V projekte sme implementovali dva trigger. Prvý trigger, ktorý bol určený priamo v zadaní má automaticky generovať hodnoty primárneho kľúča: „Autoincrement“. Automaticky inkrementuje hodnotu primárneho kľúča v tabuľke **Zamestnanec**. Trigger sme realizovali pomocou sekvencie „Id_zamestnanec_auto“, ktorá slúžila pre pamätanie si posledného pridaného kľúča. Druhým triggerom bol trigger na kontrolu správnosti zadaného rodného čísla v tabuľke **Zamestnanec**. Trigger kontroluje počet čísel, správnosť dátumu a ďalej či je rodné číslo deliteľné 11.

3.2 Procedúry

V projekte sme implementovali tri procedúry, v ktorých sú použité kurzory kvôli práci s viacerými riadkami databáze. Ďalej v každej procedúry bola použitá premenná s dátovým typom odkazujúcim sa na riadok či typ stĺpca tabuľky (table_name.column_name%TYPE alebo table_name%ROWTYPE).

Prvá procedúra „Filmy_reziser“ zisťuje počet filmov v tabuľke **Film**, ktoré sú natočené zadaným režisérom v argumente. Pri zlyhaní používame sa vypíše chybová hláška s kódom - 20005.

Druhá procedúra „Procedure_zamestnanec“ obsahuje dva argumenty: **meno** a **typ_zamestnanca**. Procedúra zo zadaných argumentov zistí celkový počet zamestnancov, počet zamestnancov podľa typu zamestnania, počet zamestnancov podľa zadaného mena a počet zamestnancov, ktorý odpovedajú obidvom argumentom. Pri zlyhaní používame sa vypíše chybová hláška s kódom -20005.

Tretia procedúra „Vypis_kina“ nemá žiadne argumenty a vypíše z tabuľky **Kino** názov kina, adresu a počet sálov. Pri zlyhaní používame sa vypíše chybová hláška s kódom -20005.

3.3 Explain plan a vytvorenie indexu

Explain plan sme demonštrovali na jednoduchom SELECT dotaze. Najprv sme spustili explain plan, alebo teda plán ako databáza spracováva daný select dotaz, na SELECTe bez použitia indexu, potom sme zadefinovali index a spustili explain plan znovu. Index sme vytvorili na tabuľke s viacerými dátami.

Bez použitia indexu

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		3	198	4 (25)	00:00:01
1	HASH GROUP BY		3	198	4 (25)	00:00:01
2	NESTED LOOPS		3	198	3 (0)	00:00:01
3	NESTED LOOPS		3	198	3 (0)	00:00:01
4	TABLE ACCESS FULL	VSTUPENKA	3	78	3 (0)	00:00:01
* 5	INDEX UNIQUE SCAN	PK_ZAMESTANEC	1		0 (0)	00:00:01

SELECT STATEMENT znamená, že sa uskutočnil select dotaz, HASH GROUP BY znamená, že sa zoskupujú položky podľa hashovacieho kľúča. Dva krát NESTED LOOPS reprezentujú spojenie, kde pre každú položku prvej tabuľky sa všetky riadky druhej tabuľky. TABLE ACCESS FULL značí prejdienie celou tabuľkou bez použitia indexov. INDEX UNIQUE SCAN reprezentuje prístup k tabuľkám cez B-strom, kde získame jeden riadok podľa primárneho kľúča v tabuľke ZAMESTNANEC.

S použitím indexu

Databáza index nepoužila, ale pri použití indexu by sa mali znížiť prístupy na disk a CPU by sa mal zvýšiť. S použitím indexu sa pristupuje do tabuľky cez konkrétny riadok (náš index).

3.4 Prístupové práva

Druhý člen tímu má pridelené práva na úrovni bežného zamestnanca. Tento zamestnanec má obmedzený prístup k tabuľke ZAMESTNANEC. K ostatným tabuľkám má prístup plne povolený a taktiež k procedúram **Filmy_reziser(Meno)** a **Vypis_kina()** má práva na spúšťanie týchto procedúr. Na pridelenie prístupu k tabuľkám sme použili GRANT ALL ON „name_table“ TO „login“

3.5 Materializovaný pohľad

Pri implementácii materializovaného pohľadu patriacemu druhému členovi tímu, ktorý používa tabuľky prvého člena tímu. Najskôr sme si vytvorili materializované záznamy obsahujúce zmeny v hlavnej tabuľke, aby sa mohol pri zmenách tabuľky využívať „fast refresh on commit“, namiesto „complete refresh“. Po vytvorení logov, sme vytvorili samotný materializovaný pohľad a predviedli jeho funkčnosť na príklade.

CACHE - postupne optimalizuje čítanie z pohľadu

BUILD IMMEDIATE - naplní pohľad hneď po jeho vytvorení

REFRESH FAST ON COMMIT - postupne optimalizuje čítanie z pohľadu

ENABLE QUERY REWRITE - bude používaný optimalizátorom

Skontrolované na jednoduchom príklade, kde sme najskôr nechali vypísať žánre filmov a následne sme pridali nový film z novým žánrom a znovu nechali vypísať.

4. Záver

Skript sme vypracovali v nástroji SQL Developer, v prostredí Oracle. Na otestovanie nám poslužil takisto SQL Developer. K úspešnému vypracovaniu tejto úlohy sme si našťudovali učebnú látku z predmetu IDS. Ostatné informácie sme získavali z rôznych internetových zdrojov, hlavne w3schools alebo stack overflow.