



IDS projekt 4,5

25.4.2022

Milan Hrabovský (xhrabo15)
Jakub Jůlius Šmýkal (xsmyka01)

Obsah

| | |
|---------------------------|---|
| Cieľ projektu | 2 |
| Trigger | 2 |
| Procedure | 2 |
| Explain Plan | 2 |
| Bez použitia indexu | 2 |
| S použitím indexu | 3 |
| Materialized View..... | 3 |

Cieľ projektu

Cieľom 4. časti projektu bolo implementovať do nášho SQL skriptu 2 trigger, 2 procedúry, explain plan, explain plan urýchlenný pomocou indexu a materialized view. Cieľom 5. časti projektu je spísať dokumentáciu k implementácii daných príkazov.

Trigger

V našom projekte sme implementovali dva rôzne trigger add_points a add_ban.

Trigger add_points sa spustí pred pridaním riadku do tabuľky pokuta. Jeho funkciou získať hodnotu bodov z novo pridanej pokuty a aktualizovať hodnotu skóre vodiča, ktorý túto pokutu dostal. Nová hodnota skóre sa bude rovnať bodom z pokuty + aktuálne skóre vodiča.

Trigger add_ban sa spustí po pridaní riadku do tabuľky pokuta, kedy je hodnota skóre vodiča už aktualizovaná po vykonaní triggeru add_points. Funkciou tohto triggeru je udeliť vodičovi zákaz šoférovania v prípade, že presiahol určitú hranicu skóre – 10 bodov. Údaje o zákaze šoférovania sú prevzaté zo zadanej pokuty.

Procedure

Ďalej sme implementovali dve procedúry recovered_prec, ktorá vracia percentuálnu úspešnosť nájdených vozidiel, ktoré boli ukradnuté a procedúru find_car, ktorá má vstupný parameter špz vozidla. Podľa špz nájde vozidlo v databáze a vypíše rok jeho výroby a názov vozidla. V prípade, že sa vozidlo s danou špz v databáze nenachádza, informuje o tom pomocou výpisu na konzolu.

V implementácii procedúry recovered_perc sme použili cursor na odkazovanie sa na stav vyriešenia krádeží. Pomocou tohto cursoru sme v cykle spočítali počet krádeží, ktoré skončili nájdením ukradnutého vozidla. Výsledky obidvoch procedúr sa vypisujú do konzole pomocou príkazu DBMS_OUTPUT.PUT_LINE.

Explain Plan

Obidve spustenia operácie EXPLAIN PLAN obsahujú operáciu SELECT. Tento SELECT získava dáta o počte krádeží a priemernej škody za krádeže evidovaných každým policajtom spojením tabuliek osoba a krádež. Daná operácia bola prevzatá z 2. časti projektu.

Bez použitia indexu

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-------------------|--------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | 4 | 204 | 7 (15) | 00:00:01 |
| 1 | HASH GROUP BY | | 4 | 204 | 7 (15) | 00:00:01 |
| * 2 | HASH JOIN | | 4 | 204 | 6 (0) | 00:00:01 |
| 3 | TABLE ACCESS FULL | KRADEZ | 4 | 80 | 3 (0) | 00:00:01 |
| 4 | TABLE ACCESS FULL | OSOBA | 6 | 186 | 3 (0) | 00:00:01 |

Operácia SELECT najprv využíva tabuľku s rozptýlenými prvkami, do ktorej vkladá dáta z tabuliek krádež a osoba. Aby získala tieto dáta, musí operácia pristúpiť ku každej tabuľke zvlášť a prečítať ich všetky riadky. Na toto musí dva krát zavolať funkciu TABLE ACCESS FULL.

S použitím indexu

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-------------------------------------|----------------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | 4 | 308 | 5 (0) | 00:00:01 |
| * 1 | HASH JOIN | | 4 | 308 | 5 (0) | 00:00:01 |
| 2 | VIEW | VW_GBC_5 | 4 | 184 | 2 (0) | 00:00:01 |
| 3 | HASH GROUP BY | | 4 | 80 | 2 (0) | 00:00:01 |
| 4 | TABLE ACCESS BY INDEX ROWID BATCHED | KRADEZ | 4 | 80 | 2 (0) | 00:00:01 |
| 5 | INDEX FULL SCAN | POLICAJT_INDEX | 4 | | 1 (0) | 00:00:01 |
| 6 | TABLE ACCESS FULL | OSOBA | 6 | 186 | 3 (0) | 00:00:01 |

Vytvorili sme index pre tabuľku krádež. Tento index odkazuje na prvok policajt, ktorý obsahuje odkaz na položku v tabuľke osoba. Tento krát operácia SELECT môže využiť nami vytvorený index, takže nemusí pristupovať k obidvom tabuľkám pomocou funkcie TABLE ACCESS FULL. Namiesto toho pristúpi k tabuľke krádež pomocou indexu cez funkciu TABLE ACCESS BY INDEX ROWID BATCHED. Použitím indexu sme dosiahli nižšiu záťaž na procesore pri vykonávaní operácie SELECT.

Materialized View

MATERIALIZED VIEW pri zavolaní vytvorí novú tabuľku z dát, ktoré získame pomocou operácie SELECT. Tieto dáta sa štandardne neaktualizujú pri zmene dát v tabuľke, z ktorej sú odvodené. Táto aktualizácia sa dá nastaviť pri vytváraní MATERIALIZED VIEW, kedy sa jeho dáta aktualizujú pri určenej zmene vykonanej nad pôvodnou tabuľkou. V tomto projekte sme vytvorili MATERIALIZED VIEW not_found s dátami z tabuľky krádež bez aktualizácie. Tieto dáta sme zobrazili pomocou operácie SELECT. Potom sme pridali novú položku do tabuľky krádež a vypísali obsah not_found, aby sme ukázali že dáta v MATERIALIZED VIEW sa nezmenili.

```
SQL> SELECT *
2 FROM xhrabo15.not_found;
```

| | ID_KRADEZE | UKRADNUTE_VOZIDLO | POLICAJT | DATUM |
|-------------------------------------|------------|-------------------|------------|-----------|
| <input checked="" type="checkbox"/> | 2 | 9KGSW5TR5KG867461 | 0107177562 | 01-APR-21 |
| <input checked="" type="checkbox"/> | 3 | 9ARCZ5TR5FT985871 | 9904171158 | 11-FEB-21 |
| <input checked="" type="checkbox"/> | 4 | 1TTCZ5FG5GM869876 | 0002092658 | 04-JUL-17 |

Page 1 of 1 |< < > >| (1-3 of 3 rows)

```
SQL> INSERT INTO xhrabo15.kradez(id_kradeze, ukradnute_vozidlo, policajt, datum, miesto, vycislene_skody, stav)
2 VALUES (DEFAULT, '7SKCZ5AN5GM864566', '0002092658', date '2022-04-28', 'základná škola v piešťanoch', 200, 'nenájdené');
```

1 row created.

Commit complete.

```
SQL> SELECT *
2 FROM xhrabo15.not_found;
```

| | ID_KRADEZE | UKRADNUTE_VOZIDLO | POLICAJT | DATUM |
|-------------------------------------|------------|-------------------|------------|-----------|
| <input checked="" type="checkbox"/> | 2 | 9KGSW5TR5KG867461 | 0107177562 | 01-APR-21 |
| <input checked="" type="checkbox"/> | 3 | 9ARCZ5TR5FT985871 | 9904171158 | 11-FEB-21 |
| <input checked="" type="checkbox"/> | 4 | 1TTCZ5FG5GM869876 | 0002092658 | 04-JUL-17 |

Page 1 of 1 |< < > >| (1-3 of 3 rows)