

Implementačná dokumentácia

Zadanie číslo č.:40 - Internetový obchod

Členovia tímu: Michal Mikuš Richard Seipel
Login: xmikus18 xseipe00

1 Úvodný opis nášho SQL skriptu

1.1 Úloha 1 až 3

Pomocou `CREATE TABLE` sme si vytvorili potrebné tabuľky, podľa predom vytvoreného ER diagramu. Následne sme pomocou `INSERT INTO` naplnili jednotlivé tabuľky testovacími hodnotami.

V ďalšej fáze projektu sme pomocou príkazov `SELECT` otestovali, či naša databáza obsahuje správne dáta a postupne sme začali pridávať aj zložitejšie konštrukcie `SELECT` príkazov. Napríklad príkaz `SELECT` využívajúci spojenie 2 tabuliek do jednej, vnorený príkaz `SELECT` alebo príkaz `SELECT` s predikátmi `IN` a `EXISTS`. Neskôr sme takiež pridali `SELECT` s využitím **agregačnej funkcie** a klauzúl `GROUP BY`.

1.2 Úloha 4

V poslednej časti sme implementovali viacero databázových **triggrov** pre automatickú inkrementáciu identifikátorov, trigger `zmena_stavu_objednavky` pre zmenu stavu objednávky po schválení zamestnancom a `zisti_cenu_objednavky` pre nastavenie výšky platby podľa ceny objednávky.

Následne sme vytvorili dve **procedúry**: `pridaj_zamestnanca` a `vypocitaj_novu_cenu`. Prvá slúži na pridanie zamestnanca spolu s prepojením na vedenie ktoré ho spravuje a druhá na vypočítanie aktuálnej ceny pre vybranú objednávku. V druhej procedúre sme využili aj **kurzor** a **ošetrenie výnimiek** ako bolo v zadaní požadované. Taktiež typy premených nastavujeme pomocou: `nazov_tabulky.nazov_stlpca%TYPE`.

Následovalo vytvorenie **indexu** pre optimalizáciu prístupu k dátam, ktorý je bližšie popísaný v sekcii 2. Pri ňom prebehne výpis plánu prevedenia nad dotazom `SELECT` so spojením dvoch tabuliek, využitím **agregačnej funkcie** a klauzúl `GROUP BY`.

Ďalej sme vytvorili **materializované pohľady** na databázove objekty pre člena tímu `XMIKUS18` a priradili mu prístupové práva na tieto pohľady, využívajúce tabuľky definované druhým členom `XSEIPE00`. Pre kontrolu funkčnosti je napísaný dotaz `SELECT`, ktorý funguje aj pri lokálne nevytvorených tabuľkách.

2 Optimalizácia prevedenia dotazu

Najprv si necháme pomocou `EXPLAIN PLAN` vypísať plán prevedenia dotazu `SELECT` so spojením dvoch tabuliek, využitím **agregačnej funkcie** a klauzúl `GROUP BY`, na výpis priemerného hodnotenia, názvu a ceny produktu.

Vo výstupe bolo vidno priamy prístup dotazu k tabuľke `Hodnotenie` a jej postupné prehľadanie. Potom sme pomocou `CREATE INDEX` vytvorili index s názvom `hodnotenie_pocet_i` nasledne sme na tabuľke `Hodnotenie` znova nechali vypísať plán prevedenia, ale tentokrát už s vytvoreným indexom. Vo výpise bolo vidno, že už dotaz nepristupuje priamo k hodnotám tabuľky `Hodnotenie`, ale pristupuje k indexovému bloku, v ktorom sa nachádza nami dotazovaná hodnota.

Nakoľko naša tabuľka nemá veľa záznamov zrýchlenie hľadania nebolo až tak signifikantné. Výraznejší rozdiel v efektívite prístupu by sa prejavil pri tabuľke s väčším počtom hodnôt a indexovaním aj drhej vyberanej tabuľky.