Laboratorij programskog inženjerstva i informacijskih sustava 2
odjeljak Sustavi baza podataka

Laboratorijske vježbe 2. dio
Optimizacija upita u sustavu PostgreSQL 9.4

Cilj ovih laboratorijskih vježbi je upoznavanje s osnovama optimizacije upita u sustavu za upravljanje bazama podataka PostgreSQL. Zadatak je kreirati i podacima napuniti vlastite testne relacije, te osmisliti i obaviti niz eksperimenata pomoću kojih će se steći uvid u osnovne postupke nadgledanja i upravljanja optimizacijom upita.

Podrazumijeva se da je prva vježba - Instalacija programske potpore za laboratorij PIIS 2, uspješno obavljena prema uputama (http://www.fer.hr/predmet/sbp/laboratorij_piis_2).

Za uspješno obavljanje ovih vježbi potrebno je proučiti sljedeću literaturu (zbog referenci na stranice podrazumijeva se korištenje PDF formata navedenog priručnika):

PostgreSQL 9.4.0 Documentation
http://www.postgresql.org/files/documentation/pdf/9.4/postgresql-9.4-A4.pdf

- Poglavlje 5.4
  - dovoljno je proučiti samo sistemske atribute oid i ctid
- Poglavlje 11.
  - pri čemu se može **preskočiti** sljedeće:
    - 11.9 do 11.10
- Poglavlje 14.
  - pri čemu se može **preskočiti** sljedeće:
    - 14.4.1
    - 14.4.5 do 14.4.7
    - 14.4.9
    - 14.5
- Poglavlje 23.1
  - pri čemu se može **preskočiti** sljedeće:
    - od 23.1.4 do kraja poglavlja
- Poglavlje VI Reference
  - naredbe:
    - CLUSTER
    - CREATE INDEX
- Poglavlja 47.1 i 47.5
- Poglavlja 48.11 i 48.69

Svu ostalu dostupnu literaturu, internet, itd. koristiti prema potrebi.

Uvodne napomene:

- za izvršavanje SQL naredbi preporuča se alat PostgreSQL pgAdmin III, ali je dopušteno korištenje bilo kojeg drugog sličnog alata
- prije provođenja eksperimenata, na globalnoj razini isključiti automatsko pokretanje brisanja zastarjelih n-torki i reorganizacije fizičkog prostora (*vacuum*), te ažuriranje statističkih podataka (*track_counts*). Naputak: u ljusci operacijskog sustava (pomoću VmWare konzole ili putty emulatora), kao korisnik root obaviti sljedeće:
  - u datoteci /var/lib/pgsql/9.4/data/postgresql.conf, editorom zamijeniti
    `#autovacuum = on` → `autovacuum = off` (uočiti: ukloniti oznaku komentara #)
    `track_counts = on` → `track_counts = off`

    - zaustaviti i ponovo pokrenuti postgresql. Obaviti naredbu
      `service postgresql-9.4 restart`
    - U SQL editoru možete provjeriti nove vrijednosti parametara izvršavanjem "SQL" naredbi `SHOW autovacuum` ili `SHOW track_count`
    - Napomena: na ovaj način promijenjeni parametri ostaju trajno zapisani tako da ovaj postupak nije potrebno ponavljati kod svakog ponovnog pokretanja SUBP-a ili virtualnog stroja. Kada 2. laboratorijsku vježbu dovršite, parametre možete vratiti na pretpostavljenu vrijednost (*on*).
- kreiranje testnih relacija (koje će se koristiti u eksperimentima) i obavljanje eksperimenata obavljati kao korisnik postgres
- u naredbama za kreiranje vlastitih testnih relacija NE navoditi eksplicitno primarne, alternativne i strane ključeve, kako bi se izbjeglo kreiranje indeksa koji bi mogli utjecati na rezultate eksperimenata. Bez obzira na navedeno, testni podaci moraju biti takvi da zadovoljavaju entitetski integritet i integritet stranog ključa. Ostala integritetska ograničenja u testnim podacima nije nužno definirati niti ih testni podaci moraju zadovoljavati.

Korisne informacije:

- kako postaviti format za datum na dd.mm.yyyy
  ```
  SET dateStyle ='German, DMY';
  ```

- prikupljanje, pregled i brisanje statističkih podataka o tablici (relaciji)

```
--  prikupljanje za relaciju osoba (i pripadne indekse i histograme)
VACUUM ANALYZE VERBOSE osoba;

-- pregled
SELECT * FROM pg_class WHERE relname = 'osoba' AND relkind = 'r';
-- 'r' znači 'relacija', indeksi se dobiju pomoću relkind = 'i'

SELECT * FROM pg_stats WHERE tablename = 'osoba';

-- brisanje svih statističkih podataka za relaciju osoba
UPDATE pg_class SET relpages = 0
                  , reltuples = 0
                  , relallvisible = 0
   WHERE oid = (SELECT oid FROM pg_class
                WHERE relname = 'osoba'
                AND relkind = 'r');
DELETE FROM pg_statistic
   WHERE starelid = (SELECT oid FROM pg_class
                     WHERE relname = 'osoba' AND relkind = 'r');
```

- grafički prikaz plana izvršavanja može se dobiti tako da se u SQL editoru označi upit i odabere gumb "Explain query" ili tipka F7 ili Shift-F7.

- vrijednost konfiguracijskog parametra može se u SQL editoru ispisati naredbom `SHOW konf_parametar` ili naredbom `SHOW ALL`.

1. ZADATAK

- nacrtati i opisati ER model (vrlo ukratko opisati značenje podataka), pripadnu relacijsku shemu definirati u obliku SQL naredbi za kreiranje relacija. Model ne bi trebao obuhvaćati više od dvije-tri relacije s po nekoliko atributa. Ako se takav manji model ne pokaže prikladnim za obavljanje svih eksperimenata, dopušteno je načiniti veći model ili više različitih modela, ali nastojati veličinu i broj različitih modela svesti na nužni minimum
    - modelirati se smije bilo koji dio realnog svijeta. Jedino ograničenje jest da se model mora potpuno razlikovati od modela korištenih na domaćim zadaćama predmeta Sustavi baza podataka, uključujući i model baze podataka koji se koristi za zadatke s pohranjenim procedurama
- generirati ili na drugi način pripremiti testne podatke (npr. preuzimanjem s neke web stranice). Za pripremu podataka i punjenje relacija tim podacima može se koristiti bilo koji prikladan postupak. Neki od mogućih postupaka su:
    - generiranje pomoću pohranjenih procedura
    - generiranje pomoću programskog jezika java (jdbc)
    - generiranje pomoću programskog jezika C# (odbc)
        - pripremanjem podataka u datotekama (u tekstualnom obliku) i punjenje (*load*) u tablice pomoću COPY ... FROM ... WITH DELIMITER ...
    - generiranje niza INSERT naredbi (na bilo koji način) i izvršavanje dobivenih SQL skripta
- količina testnih podataka nije propisana, treba ih biti "dovoljno da bi se uspješno obavili eksperimenti"
- istinitost i "uvjerljivost" testnih podataka nije propisana. Npr. imena osoba mogu biti Xusdzgsf, Tasudfu, Qdhh, ... itd, ili John, Ivo, Pero, ... Međutim, s pojednostavljivanjem se ne smije ići u krajnost, npr. tako da se 10 000 n-torki neke relacije generira kopiranjem jedne, uvijek iste n-torke.
- prije nego se krene u osmišljavanja modela i testnih podataka, treba pogledati što će se s tim podacima trebati prikazati u eksperimentima (pogledati 2. ZADATAK)
- nije dopušteno koristiti eksplicitno ograničavanje odabira metode obavljanja operacije (npr. `SET enable_hashjoin TO off` ili `SET join_collapse_limit to 1`) s ciljem postizanja traženog efekta u eksperimentu

Što treba predati kao rezultat 1. zadatka:

- Slika (ili slike) i opis (ili opise) ER modela
- SQL naredbe za kreiranje relacija
- Programe i/ili skripta i/ili podatke i/ili ... ovisno o tome kojom metodom su pripremljeni testni podaci - ukratko, sve što je potrebno (uključujući i kratko objašnjenje kako se do testnih podataka došlo) da bi osoba koja ponavlja eksperimente (npr. nastavnik koji će ocjenjivati vježbu) mogao bez teškoća pripremiti testne relacije i podatke.

2. ZADATAK

Osmisliti, provesti i ukratko komentirati (jednom do najviše nekoliko rečenica) eksperimente za sljedeće zadatke:

1. Primjerom prikazati kako procijenjeni broj n-torki rezultata upita utječe na odabir metode *sequential scan* ili *index scan* pri obavljanju jednostavne selekcije s uvjetom koji sadrži neki atribut B nad kojim je izgrađen indeks, odnosno zašto se *sequential scan* koristi za jedan uvjet selekcije nad atributom B, a *index scan* za neki drugi uvjet selekcije nad atributom B, iako i u jednom i drugom slučaju "postoji indeks koji bi se mogao koristiti")
2. Na primjeru pokazati kako je pristup n-torci putem sistemskog atributa *ctid* efikasniji od pristupa preko primarnog ključa čak i onda kada je za primarni ključ kreiran indeks. Odgovoriti na pitanje: zašto se *ctid* u pravilu ne smije koristiti za dohvat n-torki?
3. Načiniti primjer upita u kojem će se optimizator odlučiti za spajanje pomoću metode *hash join*
4. Načiniti primjer upita u kojem će se optimizator odlučiti za spajanje pomoću metode *nested loop join*
5. Primjerom prikazati kako na kvalitetu procjene broja n-torki u rezultatu nekog upita utječe ima li optimizator na raspolaganju dovoljno kvalitetne podatke iz prikladnog histograma. Pomoć: broj razreda koji će biti korišten u histogramu može se promijeniti pomoću `SET default_statistics_target`.
6. Prikazati primjerom kako upotreba funkcijskog indeksa može pridonijeti efikasnosti obavljanja upita (uočiti: PostgreSQL, za razliku od IBM IDS-a, dopušta korištenje i ugrađenih funkcija za izgradnju funkcijskih indeksa).
7. a) Prikazati primjer u kojem će optimizator uspjeti transformirati upit s podupitom u upit bez podupita (taj se postupak naziva *subquery unnesting, collapsing of subqueries into their parent query*)

   b) Prikazati primjer u kojem optimizator neće uspjeti upit s podupitom transformirati u upit bez podupita
8. Kako se može doći do podatka o stupnju grupiranja n-torki, odnosno korelaciji između njihovog fizičkog i logičkog poretka (*clustering factor*) za zadani indeks? Primjerom prikazati kako *clustering factor* utječe na odabir *sequential scan* ili *index scan* metode pri obavljanju jednostavne selekcije s uvjetom koji sadrži atribut nad kojim je izgrađen indeks (zašto se *sequential scan* koristi u jednom, a *index scan* u drugom slučaju, iako i u jednom i drugom slučaju "postoji indeks koji bi se mogao koristiti").
9. Kako se može doći do podatka o broju blokova koje zauzima indeks? Primjerom pokazati kako se parametrom za inicijalnu popunjenost B-stabla (FILLFACTOR) može utjecati na broj blokova koje će indeks zauzeti. Koje su dobre, a koje loše strane korištenja niskog faktora inicijalne popunjenosti B-stabla?
10. Primjerom prikazati korist od primjene parcijalnih indeksa (*partial index*)

Što treba predati kao rezultat 2. zadatka:

- Uz svaki eksperiment:
  - tekst zadatka
  - sve naredbe koje su se koristile tijekom provođenja eksperimenta
  - komentare uz naredbe tamo gdje je to potrebno
  - plan izvršavanja
  - ako je za eksperiment značajno, statističke podatke (vrstu i vrijednosti) koji su bili relevantni u tom eksperimentu

    - odgovore na potpitanja, ako ih ima (npr. zašto se *ctid* u pravilu ne smije koristiti za dohvat n-torki)
    - kratki komentar eksperimenta (jedna do nekoliko rečenica)
    - ne treba priložiti rezultate upita (podatke)
- plan izvršavanja kojeg treba priložiti jest rezultat naredbe EXPLAIN ANALYZE.
- nastojati koristiti čim jednostavnije primjere (*make things as simple as possible, but not simpler*)

ZAVRŠNE NAPOMENE:

- eventualne programe, skripta, itd. koji su korišteni za generiranje testnih podataka ili punjenje tablica testnim podacima, priložiti kao zasebne datoteke
- eventualne testne podatke koji su od nekud preuzeti, priložiti kao zasebne datoteke
- sve ostalo (što je navedeno da treba predati kao rezultate 1. i 2. zadatka) predati u jednoj MS Word datoteci (ne .txt, ne .pdf, ...).
    - ime datoteke treba biti: Lp2**Vašeprezime**.doc ili Lp2**Vašeprezime**.docx
    - napisati uredno zaglavlje (naslov, podaci o studentu, datum predaje, ...)
- uzeti u obzir da je urednost i organiziranost predanih rezultata jedan od elemenata koji će utjecati na ocjenu