Opdracht 3: Performance

1 Doel

Het leerdoel van deze opdracht is: kan een database kwa performance kan benchmarken, analyseren en optimaliseren.

Om dit leerdoel aan te tonen, zal je de performance van een database gaan meten d.m.v. een zelf geschreven Java programma; indexes toepassen en de invloed van de indexes op de performance onderzoeken.

2 Uitvoering

- Groepsgrootte: maximaal 2 studenten.
- De deadline: de eerst volgende ingeroosterde les.

Elk groepslid is verantwoordelijk voor alle gemaakte keuzes van het ingeleverde werk. Het cijfer wordt bepaald door de zwakste schakel binnen de groep. Je bent dus eigenlijk verplicht om het werk dat je hebt gemaakt uit te leggen aan je mede groepsgenoot, anders gaat het ten koste van je eigen groepscijfer.

3 Case

Als case wordt de case van Opdracht 1 gebruikt.

4 Opdracht

Houd er in de opdracht rekening mee dat indien je continue achter elkaar veel naar het scherm uitprint, dat de benchmark tijden kan beïnvloeden.

4.1 Creëer de database

Creëer de database zoals beschreven in opdracht 1, maar laat stored procedures achterwege.

4.2 Simuleer en benchmark

Schrijf in Java de volgende simulaties die gebruiker acties nabootsen.

4.2.1 Simulatie: Gebruikers die data invoegen

Simuleer een gebruiker die herhaaldelijk (loop/iteratie) het volgende uitvoert:

1. Start een transactie
2. Voeg een student toe
3. Voeg met een kans van 1/30 een nieuwe klas toe. (Indien er nog geen klas is, maak dan een klas aan.)
4. Koppel de student aan een willekeurige klas.

5. Voeg met een kans van 1/30 een nieuwe module toe, en koppel elke klas met een kans van 15% aan die module. (je mag, i.p.v. de roostertabel, een aparte koppeltabel maken en gebruiken.)
6. Commit de transactie
7. Wacht vervolgens 100 milliseconden. Pas de wachttijd aan indien het te snel/traag gaat.

Laat deze simulatie uitvoeren door 3 gebruikers (Java threads) tegelijkertijd, en meet de tijdsduur van het uitvoeren van stap 2 t/m 5 door de gemiddelde tijden per iteratie te berekenen.

Laat elke gebruiker 600 iteraties uitvoeren.

Noteer de gemiddelde tijd.

Je kan de tijdverschillen in Java op de volgende manier berekenen:

```
// Import boven in Java code
import java.util.Date;

//

long beginTijd = System.currentTimeMillis();

// query wordt uitvoerd......

long eindTijd = System.currentTimeMillis();
long duurInMS = eindTijd - beginTijd;
```

4.2.2 Simulatie: Gebruikers die data opvragen

Simuleer een gebruiker die herhaaldelijk (loop/iteratie) het volgende uitvoert:

1. Haal alle studenten op en kies uit deze list een willekeurige/random student. (Meet deze tijd niet mee in de benchmark.)
2. Start een transactie.
3. Haal op basis van de student voornaam i.c.m. de achternaam, de student opnieuw op uit de database.
4. Geef vervolgens alle modules weer die de student heeft gevolgd.
5. Sluit de transactie.

Laat deze simulatie uitvoeren door 2 gebruikers (Java threads) tegelijkertijd, en meet de tijdsduur van het uitvoeren door de gemiddelde tijden per iteratie te berekenen.

Laat elke gebruiker 600 iteraties uitvoeren.

Noteer de gemiddelde tijd.

4.3 Pas indexes toe

Pas indexen toe op een plek waar je denkt dat dit een performance verbetering zal geven. Beschrijf je motivatie voor deze keuze.

4.4 Controleer of de indexes gebruikt worden door Postgresql

Het toevoegen van indexen betekent niet meteen dat indexen ook gebruikt worden bij het uitvoeren van query's. Ze worden alleen uitgevoerd indien PostgreSQL denkt dat de indexen die je hebt gemaakt, bruikbaar zijn.

Gebruik PostgreSQL's EXPLAIN statement (mag buiten Java, gewoon in een PostreSQL client als pgAdmin) om aan te tonen dat er een simulatie is die gebruik maakt van je indexen. Beschrijf waaruit blijkt dat PostgreSQL ook daadwerkelijk je indexen gebruikt.

(Je kan eventueel indexen uitschakelen met [set enable_seqscan = true] (per database connectie).)

4.5 Simuleer opnieuw

Voer de simulaties opnieuw uit, en beschrijf of er verbeteringen zijn in performance.

4.6 Extra

- Gebruik R om de benchmarkresultaten in diagrammen weer te geven. Gebruik als verschillende variabelen (aantal gelijktijdige gebruikers, wachttijd, insert kansen, etc.) als functie van de responstijd.
- Experimenteer met het effect van verschillende transaction isolation levels op de performance.

5 Inlevervorm

De uitwerkingen van de opdrachten dienen op papier (in een verslag) te worden genoteerd en mondeling te worden besproken en gedemonstreerd aan de docent.

Noteer op het verslag je voor- en achternaam, studentnummer, klas en datum van inleveren. Het verslag dient op de deadline dag te worden besproken met de docent. Verder dienen de simulaties op een computer gedemonstreerd te worden aan de docent.

6 Beoordelingscriteria

Onderwerp	Percentage van het opdrachtcijfer
Simulatie: Gebruikers die data invoegen	20%
Simulatie: Gebruikers die data opvragen	20%
Pas indexes toe en controleer.	20%
Motivatie in het verslag (Compleetheid van de beschrijving van de gemaakte keuzes. M.a.w. in hoeverre vallen de gemaakte keuzes af te leiden uit de tekstuele beschrijving/motivatie in het verslag.)	20%
Extra (alleen mogelijk indien alle andere opdrachtonderdelen afgerond zijn.)	20%