Werken met transakties

1 Gebruiker

Probeer het basisgebruik van transakties uit voor 1 gebruiker. Dit kan je individueel doen. Let hierbij op de effecten van START TRANSACTION, COMMIT/ROLLBACK, SAVEPOINT

Algemene voorbeeldoefeningen:

1. Verbind met een databank (bv probeer en werk dan je eigen schema, niet in public)
2. Vastleggen van wijzigingen
   1. Start een transaktie.
   2. Maak een tabel naar keuze aan (bv *oliebol(id,kleur,..)*).
   3. Leg de transaktie vast.
   4. Kijk of de tabel *oliebol* bestaat.
3. Ongedaan maken van wijzigingen
   1. Maak de transaktie ongedaan.
   2. Kijk of de tabel *oliebol* nog bestaat.
   3. Start een transaktie.
   4. Maak een tweede tabel naar keuze aan (bv *bloedwaarde(id,cholesterol,..)*).
   5. Maak de transaktie ongedaan.
   6. Kijk of de tabel *bloedwaarde* nog bestaat.
4. Werken met savepoints.
   1. Start een transaktie.
   2. Maak een derde tabel naar keuze aan (bv *kraam(naam,type,..)*).
   3. Maak een savepoint naar keuze aan (bv *buikvol*).
   4. Maak een vierde tabel naar keuze aan (bv *beoordeling(kraam\_naam,waarde,..)*).
   5. Toon de tabel *beoordeling*.
   6. Maak de wijzigingen ongedaan tot savepoint *buikvol*.
   7. Toon de tabel *beoordeling*.
   8. Toon de tabel *kraam*.
   9. Leg de wijzigingen vast
   10. Toon de tabel *kraam*.

|  |  |
| --- | --- |
| **TIP** | Zie elektronisch materiaal en cursus voor het expliciet gebruik van transakties. |

Standaardinstellingen

Zo kom je meer te weten over de standaardinstellingen van de databank.

Voer de volgende instrukties uit en interpreteer hun uitvoer:

**SHOW** default\_transaction\_isolation;

**SHOW** transaction\_isolation;

**SHOW** autocommit;

Meerdere gebruikers

Probeer het basisgebruik van transakties uit voor meerdere gebruikers. Dus voor deze oefening heb je verschillende gebruikers nodig. Let hierbij op de effecten van START TRANSACTION, COMMIT/ROLLBACK, SAVEPOINT, ISOLATION LEVEL.

1. Verbind met een databank waar alle gebruikers aan kunnen (bv probeer)
   1. Gebruiker1 start een transaktie
      1. Gebruiker1 maakt een tabel naar keuze aan (bv *koek*).
      2. Gebruiker2 probeert de tabel *koek* te lezen.
      3. Gebruiker1 legt de transaktie vast.
      4. Gebruiker2 probeert de tabel *koek* te lezen.
      5. Ah, vergeet ook niet de juiste rechten toe te kennen.
   2. Probeer de *problemen bij multi-user gebruik* aktief uit te lokken, bv *nonreadable read* …​ Welke van deze problemen uit de presentatie (of cursus) kan je uitlokken?
   3. Probeer nu de *problemen bij multi-user gebruik* aktief te vermijden, maak hiervoor gebruik van het *isolation level*
      1. Vergeet hierbij niet expliciet het *isolation level* in te stellen.

|  |  |
| --- | --- |
| **TIP** | * Je kan deze oefeningen met je gebuur doen, of door met verschillende gebruikers op de databank aan te melden. Bv r1234567 en local\_r1234567 * De transakties dienen de overlappen in de tijd om deze problemen aktief uit te lokken, anders is gewoon alsof ze na elkaar gebeuren. |
| **NOTE** | * <https://www.postgresql.org/docs/current/transaction-iso.html> * <https://www.postgresql.org/docs/current/tutorial-transactions.html> |

Kip ei probleem

Hoe kunnen we transakties en DEFERRABLE gebruiken om het kip ei probleem op te lossen? Bij het kip ei probleem, verwijzen 2 tabellen naar elkaar. In het begin zijn beide tabellen leeg, hoe kunnen we dan toch data toevoegen aan deze tabellen?

|  |  |
| --- | --- |
| **NOTE** | * <https://www.postgresql.org/docs/current/sql-set-constraints.html> * <https://www.postgresql.org/docs/current/sql-createtable.html> * <https://www.postgresql.org/docs/current/sql-altertable.html> |

Problemen en Isolatie nivos

Welke problemen zijn toegelaten op welk isolatie nivo volgens de standaard? Concrete databanken mogen strenger zijn om te voldoen aan de ISO standaard. Geef een overzicht in tabel vorm van de 5 problemen die we gezien hebben en de isolatie nivos.

Last updated 2024-02-19 16:53:44 +0100