

JDBC ir Transakcijos

Parengė Giedrius Graževičius

Transakcija

- Nedaloma operacijų visuma
- Arba viskas, arba nieko

Transakcijos pavyzdys

- Pinigų perlaida
 - Nuskaitome pinigus iš vienos sąskaitos
 - Įrašome sumą į kitą sąskaitą
- Jei po pirmosios operacijos sistema nulūžtų...

Transakcijos DB pavyzdys

- Įsivaizduokime, kad banke sąskaitų informacijos laikomos lentelėje

Accounts

AccountNo	Balance
LT9001	2000,0
LT9003	45,0

Transakcijos DB pavyzdys (2)

- Užklausos pinigų perlaidai:
 - UPDATE Accounts SET Balance = Balance - 400
WHERE AccountNo = 'LT9001'
 - UPDATE Accounts SET Balance = Balance + 400
WHERE AccountNo = 'LT9003'
- O jei programa nulūš po pirmosios?

DB Sprendimas

- SET AUTOCOMMIT FALSE
- UPDATE Accounts SET Balance = Balance - 400 WHERE AccountNo = 'LT9001'
- UPDATE Accounts SET Balance = Balance + 400 WHERE AccountNo = 'LT9003'
- COMMIT

DB Sprendimas (2)

- SET AUTOCOMMIT FALSE - išjungiamia automatinį pakeitimų skelbimą po kiekvieno sakinio
- COMMIT - paskelbiame pakeitimus
- ROLLBACK - atšaukiame visus transakcijos pakeitimus

- Užtikrina duomenų izoliaciją daugelio vartotojų duomenų bazėse
 - Gali užtikrinti, kad nebaigtos transakcijos duomenys nebūtų matomi už transakcijos ribų
 - Gali užtikrinti, kad transakcija dirbs su konkrečiais iš išorės nekeičiamais duomenimis visą savo galiojimo laiką
 - Gali užtikrinti, kad viena transakcija negalės keisti duomenų, kuriuos keitė kita
 - Gali užtikrinti, kad viena transakcijas negalės keisti duomenų, kuriuos skaitė kita
 - ...

Izoliacija (2)

- Skirtingos DBVS palaiko skirtingus izoliacijos lygius

Transakcijos H2

- Standartinis izoliacijos lygis “Read Committed”
 - Transakcijos duomenys matomi kitoms, tik po COMMIT
 - Vienos transakcijos vykdymo metu, jai matomi duomenys gali keistis, kai kitos transakcijos įvykdo COMMIT
- Palaiko “Serializable”
 - Transakcijos duomenys matomi kitoms, tik po COMMIT
 - Kitų transakcijų skaityti duomenys netampa matomi, net joms įvykdžius COMMIT
- Galima izoliaciją išjungti

Demo

```
SET AUTOCOMMIT FALSE;  
UPDATE Projektas SET Pavadinimas = 'Abc' WHERE ID =  
1;  
SELECT * FROM Projektas;  
ROLLBACK;  
SELECT * FROM Projektas;
```

Transakcijos JDBC

```
Connection connection = ...;  
connection.setAutoCommit(false); // Pradedame transakciją  
  
connection.commit();           // Baigiame  
connection.setAutoCommit(true); // transakciją
```

Transakcijos JDBC (2)

```
connection.setAutoCommit(false);  
try {  
    ... // SQL užklauso  
    connection.commit();  
} catch (Exception ex) {  
    connection.rollback(); // Klaidos atveju atšaukiame pakeitimus  
} finally {  
    connection.setAutoCommit(true);  
}
```

1. Programa turi mokėti importuoti projektus iš tekstinio failo, kuriame kiekviena eilutė laiko projekto pavadinimą.
 - a. Programa turi po eilutę įterpinėti naujas įmones
 - b. Jei aptinka dublikatą, visos iš failo įterptos įmonės turi būti pašalintos
 - c. Panaudokite transakcijas, pakeitimams atšaukti po klaidos

Informacija užduočiai

- Sukurkime unikalumo apribojimą projektų pavadinimams
 - ALTER TABLE Projektas ADD CONSTRAINT
UnikalusPavadinimas UNIQUE(PAVADINIMAS);
- Jei bus bandoma įterpti pakartotiną Projektas.Pavadinimas reikšmę, H2 išmes klaidą, kurią JDBC grąžins kaip SQLException

Informacija užduočiai (2)



```
// Paprastas būdas nuskaityti visas eilutes iš failo
```

[illegible]