

Współbieżność: poziomy izolacji i blokady

Czas: 2 × 45 min

praca indywidualna (dwa okna konsoli A i B na jednym komputerze)

Cel: Zaobserwowanie w dwóch oknach konsoli problemów współbieżnego dostępu do danych i sprawdzenie, jak chronią przed nimi poziomy izolacji i blokady InnoDB.

Po zajęciach potrafisz:

- ustawia poziom izolacji sesji i rozpoznaje brudny odczyt, niepowtarzalny odczyt i fantom
- pokazuje utraconą aktualizację i usuwa ją atomowym UPDATE albo SELECT ... FOR UPDATE
- interpretuje błędy 1205 (lock wait timeout) i 1213 (deadlock)
- odczytuje informacje o blokadach z information_schema i SHOW ENGINE INNODB STATUS
- stosuje stałą kolejność blokowania, aby uniknąć zakleszczeń

Sprzęt i materiały

- komputer z XAMPP (MariaDB 10.4) lub MySQL 8.4
- plik `r2_13_bank.sql` z folderu Pliki_do_zadan (ten sam co w zadaniu 13)

Oprogramowanie

- **XAMPP — konsola mysql** — dwa okna „Shell”, w każdym `mysql -u root` apachefriends.org/download.html
- **Edytor tekstu (VS Code)** — plik `r2_14_wspolbieznosc.sql` code.visualstudio.com

Przygotowanie przed zajęciami

- Przypomnij sobie z prezentacji 2.08 cztery poziomy izolacji i pojęcia: brudny odczyt, niepowtarzalny odczyt, fantom, utracona aktualizacja.

Bezpieczeństwo

- Zaimportuj od nowa `r2_13_bank.sql` — plik usuwa i tworzy bazę `bank`. Nie wykonuj ćwiczeń na bazie `sklep`.
- Po każdym scenariuszu zakończ transakcje w obu oknach (COMMIT lub ROLLBACK). Zawieszona transakcja trzyma blokady i zatrzymuje inne sesje.
- Poziom izolacji zmieniaj tylko poleceniem `SET SESSION ...` — `SET GLOBAL` zmienia ustawienie całego serwera.

Przebieg zadania

1 Przygotowanie

Zapisuj, w którym oknie wykonujesz polecenie (A albo B). Zapis „B:” w poleceniu oznacza okno B.

1. Zaimportuj `r2_13_bank.sql`. Otwórz dwa okna konsoli i w obu wykonaj `USE bank;`.
2. Z1. Sprawdź w oknie B domyślny poziom izolacji: w MariaDB `SELECT @@tx_isolation;`, w MySQL 8.4 `SELECT @@transaction_isolation;` oraz wartość `@@innodb_lock_wait_timeout`.

2 Poziomy izolacji

Poziom izolacji ustawiaj w oknie B przed START TRANSACTION, np. `SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;`.

1. Z2. Brudny odczyt — B: poziom READ UNCOMMITTED. A: rozpocznij transakcję i zdejmij 1000 zł z konta 1 (bez COMMIT). B: odczytaj saldo konta 1. A: ROLLBACK. B: odczytaj saldo jeszcze raz.
2. Z3. Powtórz Z2 z poziomem READ COMMITTED w oknie B. Co widzi B?
3. Z4. Niepowtarzalny odczyt — B: READ COMMITTED, START TRANSACTION i odczyt salda konta 2. A: dopisz 100 zł do konta 2 (zwykły UPDATE, autocommit). B: odczytaj saldo konta 2 ponownie w tej samej transakcji, potem COMMIT.
4. Z5. Powtórz Z4 z poziomem REPEATABLE READ. Sprawdź saldo w B przed COMMIT i po COMMIT.
5. Z6. Fantom — B: READ COMMITTED, START TRANSACTION, policz konta z saldem co najmniej 1000 zł. A: dodaj konto `(6, 'Maja Kamińska', 3000)`. B: policz ponownie, COMMIT.
6. Z7. Powtórz Z6 z poziomem REPEATABLE READ i kontem `(7, 'Tomasz Lewandowski', 4000)`. W oknie B wykonaj policzenie trzy razy: zwykłe, z dopiskiem `FOR UPDATE` i znowu zwykłe. Wyjaśnij różnicę (migawka MVCC a odczyt blokujący).

3 Utracona aktualizacja i FOR UPDATE

Symulujemy aplikację, która odczytuje saldo, liczy nową wartość „u siebie” i zapisuje liczbę.

1. Z8. W obu oknach START TRANSACTION i `SELECT saldo INTO @s FROM konta WHERE nr_konta = 3;`. A: `UPDATE konta SET saldo = @s - 100 WHERE nr_konta = 3; COMMIT;`. B: to samo z kwotą 200 i COMMIT. Jakie saldo powinno zostać, a jakie zostało? Przywróć saldo 2300.
2. Z9. Powtórz Z8, ale odczyt w obu oknach zakończ `FOR UPDATE`. Zaobserwuj, że B czeka, aż A zatwierdzi. Jaką wartość odczytał B i jakie jest saldo końcowe?
3. Z10. Zapisz, jak poprawić Z8 bez blokady jawnej — jednym atomowym poleceniem UPDATE — i wykonaj je w obu oknach.

4 Czekanie na blokadę i zakleszczenie

1. Z11. B: `SET SESSION innodb_lock_wait_timeout = 30;`. A: START TRANSACTION i zablokuj konto 4 przez `SELECT ... FOR UPDATE`. B: START TRANSACTION, dopisz 1 zł do konta 5, potem spróbuj zdjąć 1 zł z konta 4. Gdy B czeka, w oknie A wyświetl `trx_id, trx_state, trx_query` z `information_schema.INNODB_TRX`. Odczekaj do błędu w B i zanotuj jego numer. Sprawdź w B, czy zmiana konta 5 nadal istnieje. Zakończ obie transakcje ROLLBACK.
2. Z12. A: zablokuj konto 4 jak w Z11. B: spróbuj `SELECT ... FOR UPDATE NOWAIT` na koncie 4. Zakończ obie transakcje.
3. Z13. Zakleszczenie — A i B: START TRANSACTION. A: zdejmij 10 zł z konta 1. B: zdejmij 10 zł z konta 2. A: dopisz 10 zł do konta 2 (A czeka). B: dopisz 10 zł do konta 1. Zanotuj błąd w B, sprawdź w B saldo konta 2, zakończ obie transakcje COMMIT i wyświetl salda kont 1 i 2.
4. Z14. Wyświetl `SHOW ENGINE INNODB STATUS\G` i odszukaj sekcję LATEST DETECTED DEADLOCK. Wypisz, które polecenie czekało na którą blokadę.
5. Z15. Powtórz Z13, ale w obu oknach najpierw zmieniaj konto o niższym numerze. Czy zakleszczenie wystąpiło?

W sprawozdaniu umieść

- ☐ tabela zapytań Z1–Z15 z oznaczeniem okna (A/B) i wynikiem
- ☐ tabela porównania poziomów izolacji wypełniona na podstawie Z2–Z7
- ☐ zrzuty ekranu dwóch okien z Z8 (utracona aktualizacja) i Z13 (błąd 1213)
- ☐ fragment sekcji LATEST DETECTED DEADLOCK z Z14 z opisem
- ☐ plik `r2_14_wspolbieznosc.sql`

☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Dlaczego brudny odczyt jest groźny? Podaj przykład decyzji podjętej na podstawie danych, które nigdy nie zostały zatwierdzone.
2. Czym różni się zwykły SELECT od SELECT ... FOR UPDATE w transakcji REPEATABLE READ?
3. Co serwer robi z transakcją po błędzie 1213, a co po błędzie 1205? Jak powinna zareagować aplikacja?
4. Podaj dwa sposoby uniknięcia utraconej aktualizacji w sklepie internetowym.

Kryteria oceny

Kryterium	Punkty
Scenariusze poziomów izolacji Z2–Z7 i tabela porównania	6
Utracona aktualizacja i jej usunięcie (Z8–Z10)	4
Blokady, timeout i zakleszczenie z analizą (Z11–Z15)	6
Odpowiedzi na pytania kontrolne	4
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).