Zaawansowane technologie bazodanowe
Mikołaj Wielgos
Lab4

Ćwiczenie 1: Odczyty niestabilne (Non-repeatable reads)

```
BEGIN;
SELECT vacation_days FROM employees WHERE id = 2;
-- 10
BEGIN;
UPDATE employees SET vacation_days = 15 WHERE id = 2;
COMMIT;
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
COMMIT;
```

Ćwiczenie 2: Jawne blokowanie

```
BEGIN;
LOCK TABLE employees IN SHARE MODE;
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
BEGIN;
UPDATE employees SET vacation_days = 25 WHERE id = 2; -- oczekiwanie na lock…
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
COMMIT;
UPDATE employees SET vacation_days = 25 WHERE id = 2; -- wykonanie operacji
COMMIT;
```

Ćwiczenie 3: Poziom izolacji *repeatable read*

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
UPDATE employees SET vacation_days = 25 WHERE id = 2;
COMMIT;
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
COMMIT;
```

Ćwiczenie 4: Błędy serializacji

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
UPDATE employees SET vacation_days = vacation_days + 100 WHERE id = 2;
COMMIT;
UPDATE employees SET vacation_days = vacation_days + 5 WHERE id = 2;
 [40001] ERROR: could not serialize access due to concurrent update
```

Ćwiczenie 5: Blokady na poziomie rekordów

```
BEGIN;
SELECT vacation_days FROM employees WHERE id = 2 FOR UPDATE;
-- 15
BEGIN;
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
UPDATE employees SET vacation_days = vacation_days + 5 WHERE id = 2;
-- waits for lock...
SELECT vacation_days FROM employees WHERE id = 2;
-- 15
COMMIT;
UPDATE employees SET vacation_days = vacation_days + 5 WHERE id = 2;
-- completes
COMMIT;
```

Ćwiczenie 6: Transakcje a wydajność

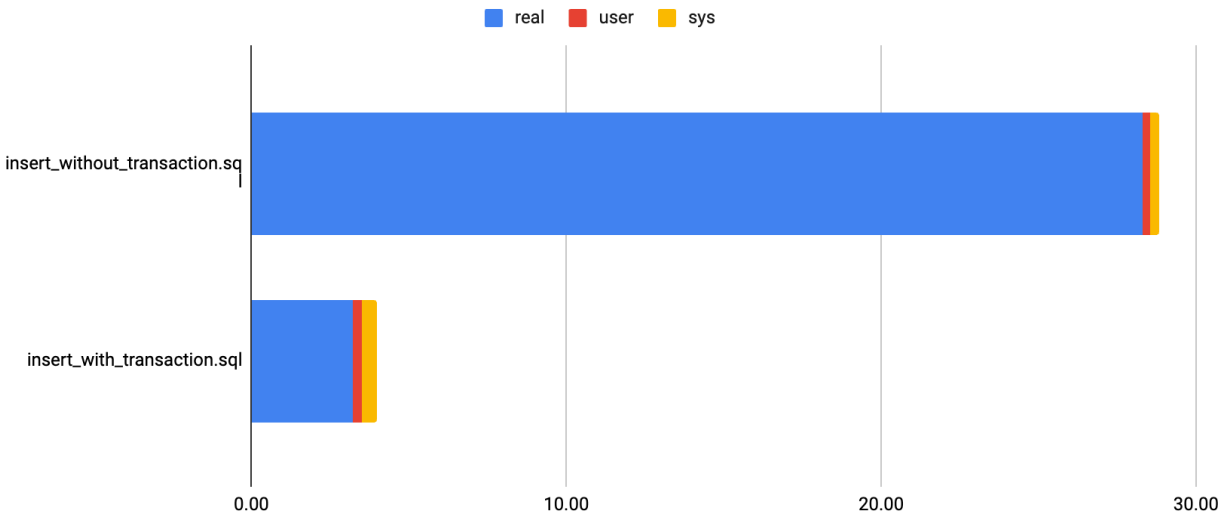
insert_with_transaction.sql

```sql
DROP TABLE IF EXISTS performance_test;
CREATE TABLE performance_test (
   id SERIAL PRIMARY KEY,
   name VARCHAR(50),
   description TEXT
);
BEGIN;
INSERT INTO performance_test (name, description) VALUES ('hognpG5cMhMhJuMky66F',
'FYpb1M21V1yFaipgkFYa4jkNt1ARMTxSwslURqw87Hb44nWJUXWJRqXHqSAx1mYYTM8ufRpnFLQs6gKP');
COMMIT;
```

insert_without_transaction.sql

```sql
DROP TABLE IF EXISTS performance_test;
CREATE TABLE performance_test (
   id SERIAL PRIMARY KEY,
   name VARCHAR(50),
   description TEXT
);
INSERT INTO performance_test (name, description) VALUES ('hognpG5cMhMhJuMky66F',
'FYpb1M21V1yFaipgkFYa4jkNt1ARMTxSwslURqw87Hb44nWJUXWJRqXHqSAx1mYYTM8ufRpnFLQs6gKP');
```

insert_without_transaction.sql								
real	27.02	29.40	28.77	27.93	27.59	27.02	28.28	29.40
user	0.28	0.25	0.25	0.29	0.29	0.25	0.27	0.29
sys	0.29	0.27	0.33	0.24	0.29	0.24	0.28	0.33
insert_with_transaction.sql								
real	3.348	3.109	3.192	3.151	3.216	3.109	3.2	3.348
user	0.324	0.299	0.315	0.28	0.277	0.277	0.3045	0.324
sys	0.528	0.487	0.482	0.516	0.538	0.482	0.50325	0.538


real
user
sys
insert_without_transaction.sq
insert_with_transaction.sql
0.00
10.00
20.00
30.00