Бази даних

Лекція 10

Тематика лекції

- Транзакції
- ACID
- Рівні ізоляції
- SELECT ... FOR UPDATE

Приклад

```
UPDATE account_balance SET balance = balance - 100 WHERE account_id = 1;

UPDATE account_balance SET balance = balance + 100 WHERE account_id = 2;
```

Приклад

```
UPDATE account_balance SET balance = balance - 100 WHERE account_id = 1;

---------- тут втратився доступ до бази даних

-- запит не виконався
-- гроші списано з account_id 1, проте не зараховано account_id 2
UPDATE account_balance SET balance = balance + 100 WHERE account_id = 2;
```

Транзакції

Транзакція - це набір операцій, які виконуються як одна неподільна задача - виконуються або всі запити, або жоден.

Транзакції допомагають виконувати кілька запитів, як одну логічну задачу і також надають фінкціонал для повернення бази даних до стану перед початком виконання даної задачі.

Приклад - вірний підхід

```
-- Початок транзакції
BEGIN;

UPDATE account_balance SET balance = balance - 100 WHERE account_id = 1;

UPDATE account_balance SET balance = balance + 100 WHERE account_id = 2;

-- Виконання транзакції
COMMIT;
```

Приклад - вірний підхід

```
-- Початок транзакції
BEGIN;

UPDATE account_balance SET balance = balance - 100 WHERE account_id = 1;

---------------- тут втратився доступ до бази даних
-- COMMIT не виконався - з account_id 1 не списались гроші
```

Контроль транзакцій

- **BEGIN - початок транзакції.**
- **COMMIT - збереження транзакції.**
- **ROLLBACK - відміна транзакції.**

Властивості транзакцій - ACID

- Atomicity - Атомарність
- Consistency - Консистентність
- Isolation - Ізольованість
- Durability - Персистентність

Атомарність (Atomicity)

Операції в транзакції виконуються всі, або не виконується жодна з них.

Консистентність (Consistency)

Транзакція переводить базу даних з одного коректного стану в інший - в результаті транзакції жодні обмеження та відношення не будуть поламані.

Ізоляція (Isolation)

Окремі транзакції незалежні - паралельні транзакції не впливають одна на одну.

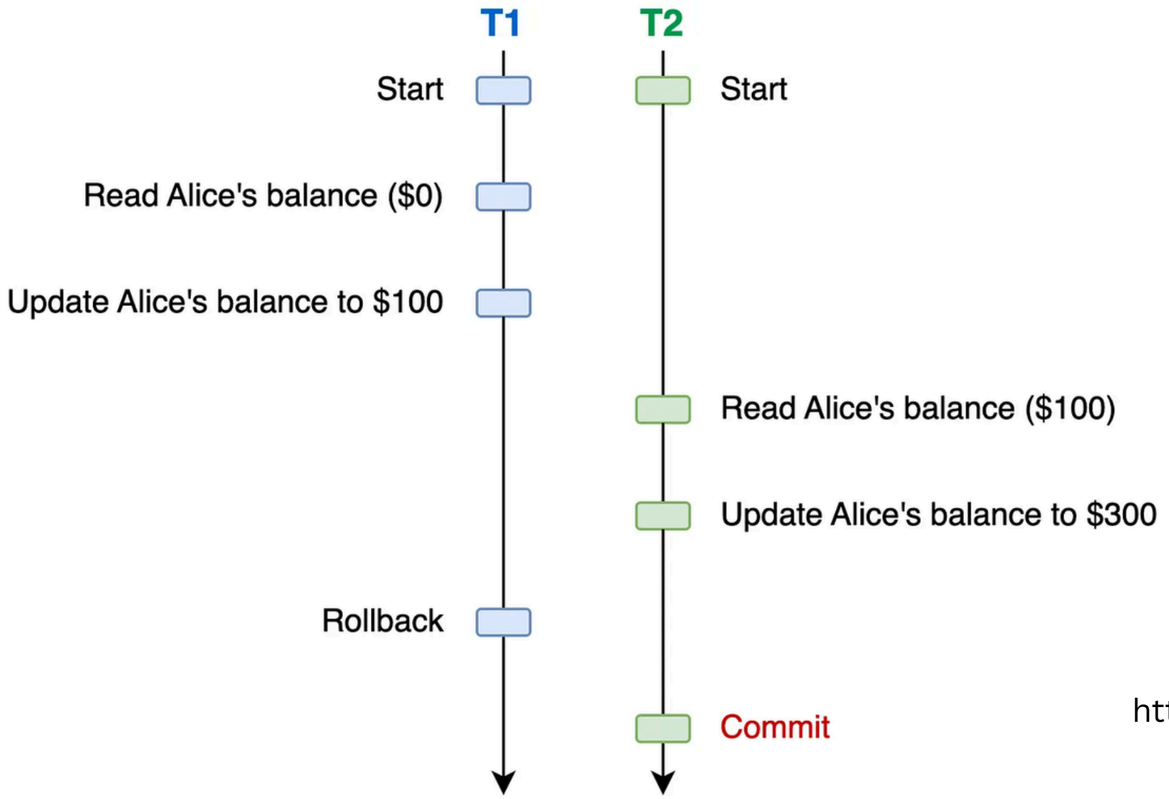
Персистентність (Durability)

Зміни з успішно завершеної транзакції (committed) не буде втрачено.

Проблеми паралельних транзакцій

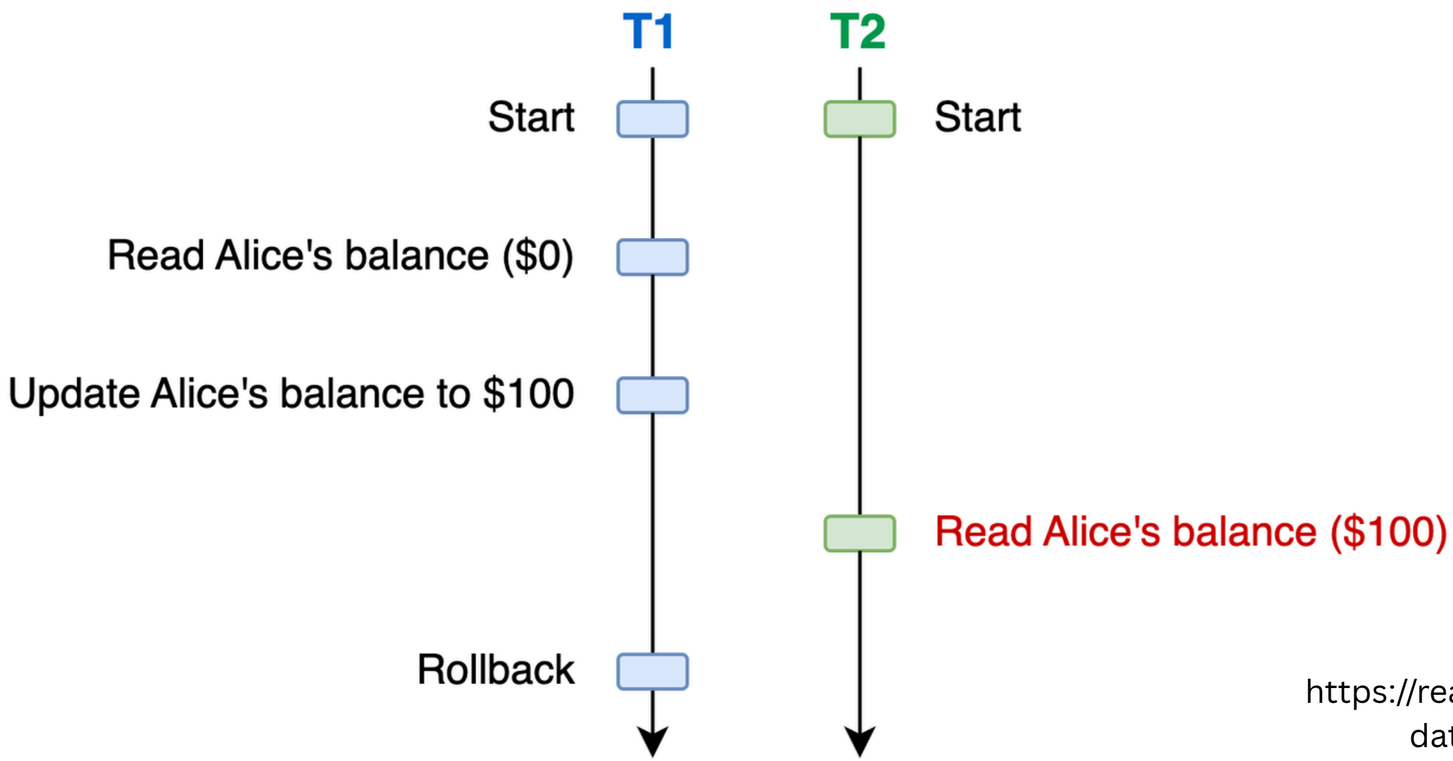
- Dirty write
- Dirty read
- Non-repeatable read / Phantom read
- Lost update
- Write skew

Dirty write



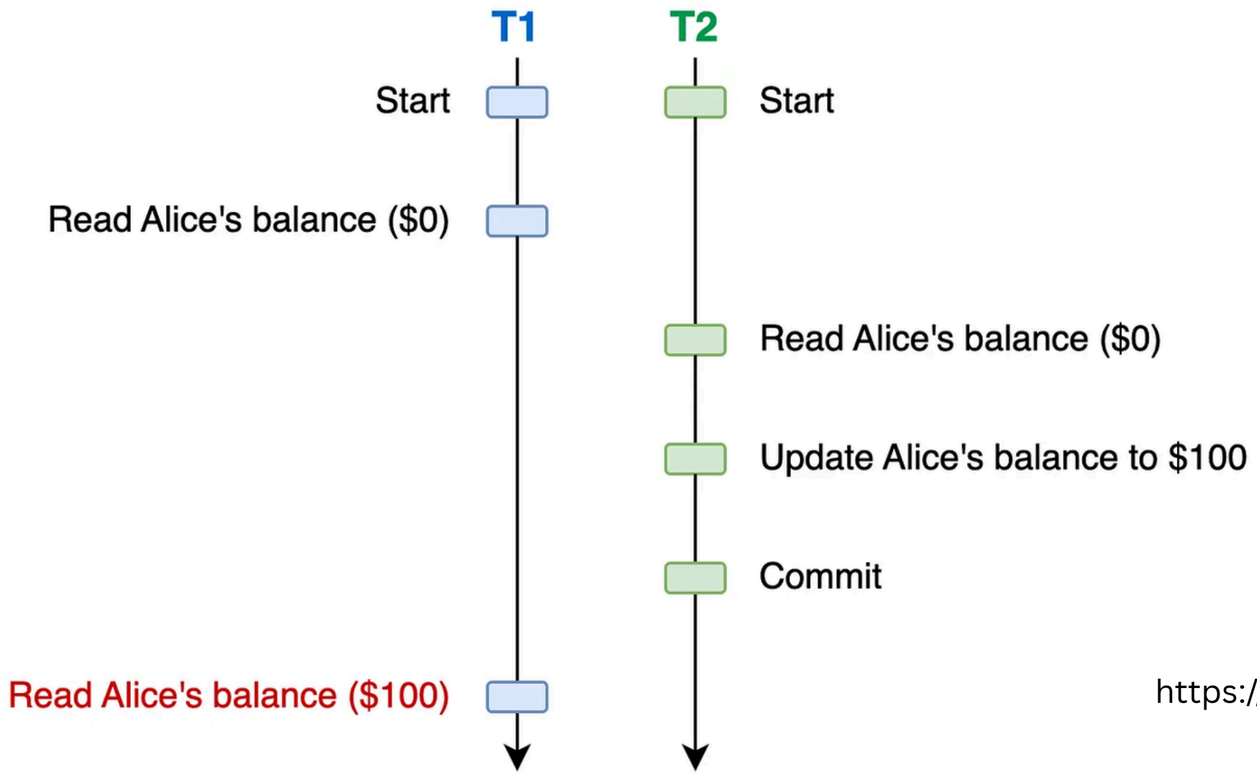
https://read.thecoder.cafe/p/exploring-database-isolation-levels

Dirty read



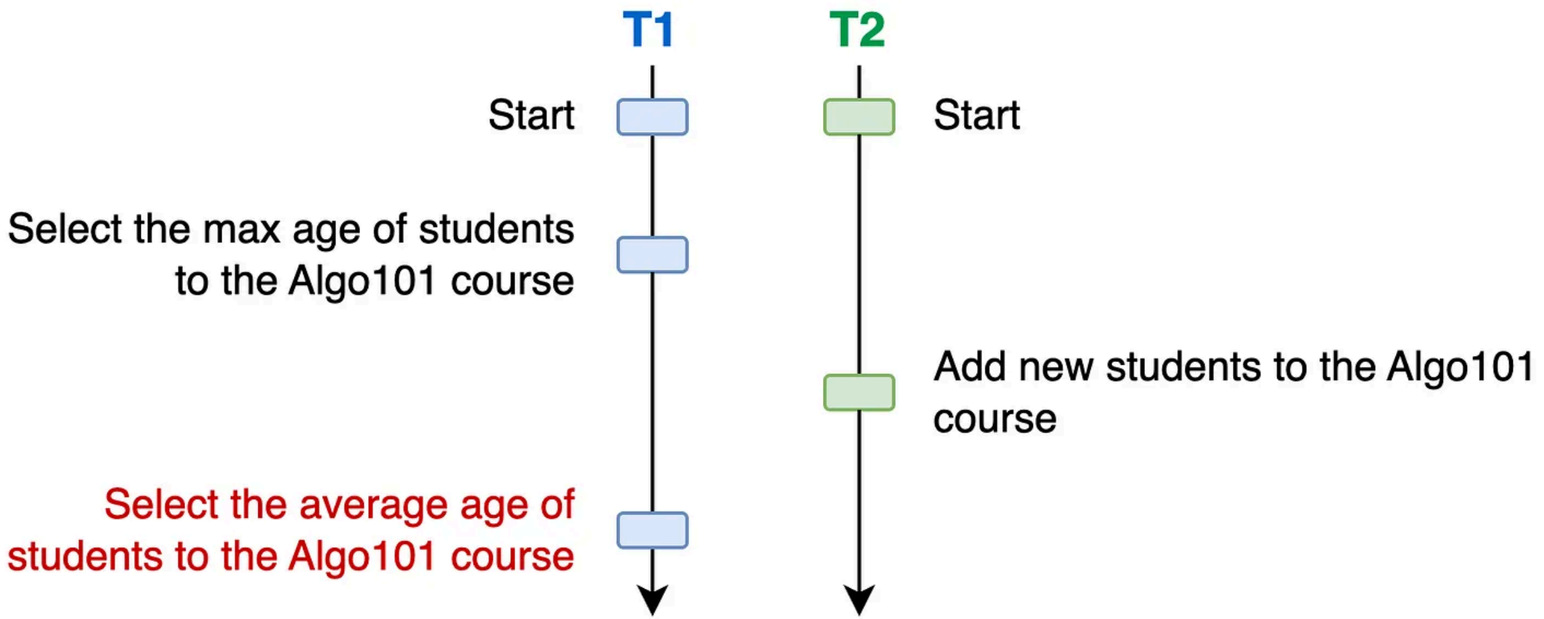
https://read.thecoder.cafe/p/exploring-database-isolation-levels

Non-repeatable read



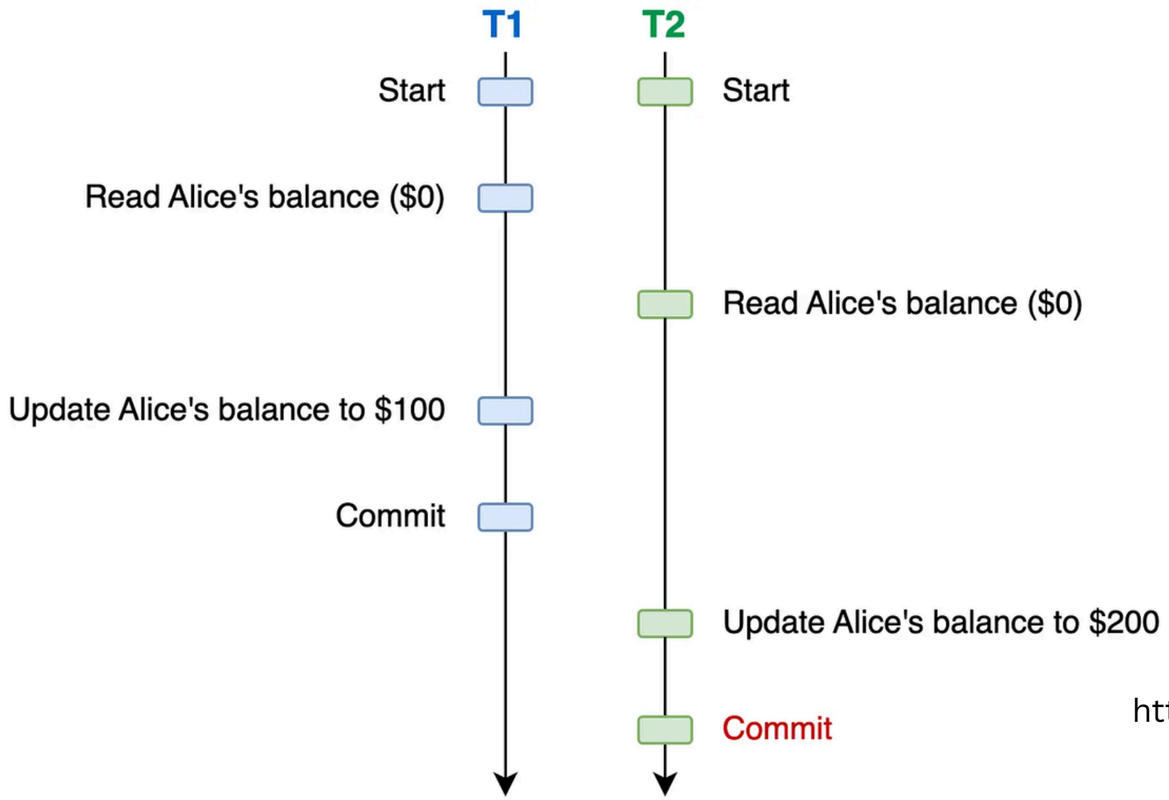
https://read.thecoder.cafe/p/exploring-database-isolation-levels

Phantom read



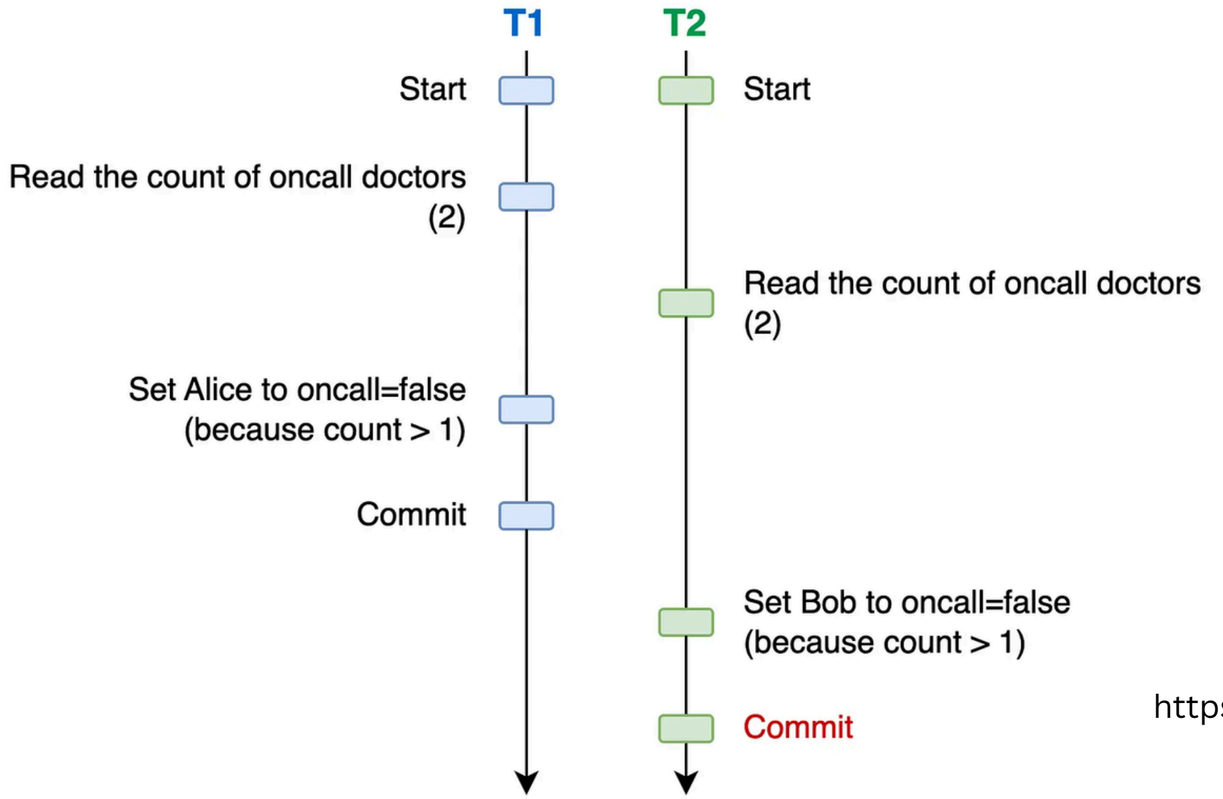
https://read.thecoder.cafe/p/exploring-database-isolation-levels

Lost update



https://read.thecoder.cafe/p/exploring-database-isolation-levels

Write skew



https://read.thecoder.cafe/p/exploring-database-isolation-levels

Рівні ізоляції транзакцій

1. **No Isolation - жодної ізоляції, всі проблеми можливі**
2. **Read Uncommitted - унеможливлює “Dirty write”**
3. **Read Committed - унеможливлює “Dirty read”**
4. **Repeatable Read - унеможливлює “Non-repeatable read”**
5. **Snapshot Isolation - унеможливлює “Lost update”**
6. **Serializable - унеможливлює “Phantom read”, “Write skew”**

Рівні ізоляції транзакцій - PostgreSQL

1. **Read Uncommitted = Read Committed - унеможливлює “Dirty read”, “Dirty write”**
2. **Repeatable Read = Snapshot Isolation - унеможливлює “Non-repeatable read”, “Lost update”, “Phantom read”**
3. **Serializable - унеможливлює “Write skew”**

Рівні ізоляції транзакцій - PostgreSQL

BEGIN TRANSACTION ISOLATION LEVEL <>;

SELECT ... FOR UPDATE

Альтернативний спосіб боротьби з “Lost update”, без збільшення рівня ізоляції транзакції.

Встановлює лок на рядки, які повертаються з SELECT. Лише поточна транзакція зможе їх оновлювати, а будь-які інші оновлення будуть очікувати, поки поточна транзакція не завершиться.

SELECT ... FOR UPDATE

Блокує операції на ті ж самі рядки:

- **UPDATE**
- **DELETE**
- **SELECT ... FOR UPDATE**

Не блокує:

- **SELECT**

Варіації SELECT ... FOR UPDATE

- FOR UPDATE - ексклюзивний лок на весь рядок
- FOR NO KEY UPDATE - ексклюзивний лок на запис на не ключові атрибути
- FOR SHARE - лок на рядок, який дозволяє багато одночасних FOR SHARE, проте не дозволяє UPDATE / DELETE
- FOR KEY SHARE - те ж саме, що і FOR SHARE, проте лише на ключові атрибути

Практичні поради

- **Транзакції дозволяють виконувати кілька запитів “як один” - отже, якщо одна задача потребує кількох запитів - їх потрібно виконувати в транзакції.**

- **Варто використовувати найнижчий з необхідних рівнів ізоляції, адже чим вищий рівень ізоляції - тим менша швидкодія.**

Практичні поради

- **SELECT ... FOR UPDATE дозволяє явно встановити лок для набору рядків, тож може бути використаний замість рівня ізоляції Repeatable Read для сценарію read - process - update. Проте може призвести до дедлоків, тож варто використовувати з обередністю.**

- **Транзакції варто використовувати для небезпечних запитів, які виконуються “вручну”, адже поки транзакція не збережена - зміни завжди можна “відкотити”.**

Запитання