مشروع نهاية الفصل 9

مقرر قواعد البيانات — Transactions — كله أو ولا حاجة

الهدف: تحمي عملية من خطوتين بـ transaction وتشوف lost update بالمحاكاة.
الأدوات: SQL على SQLite، وتقدر تشغّل الكود في صفحة المحاضرة أو في أي أداة SQLite مجانية.

الداتا: بيانات تجريبية مكتوبة خصيصًا للمقرر ومتاحة لإعادة الاستخدام.

المطلوب

1 جهّز (Prepare)

- .account(id, balance CHECK >= 0)

2 تحويل (Transfer)

- حوّل 30 بـ transaction وCOMMIT.

3 فشل (Failure)

- جرّب تحويل أكبر من الرصيد واعمل ROLLBACK.

4 Lost update (Lost update)

- حاكي lost update بقيم ثابتة (الصح 80 مش 90).

بونس

- بونس: اعرض الأرصدة بعد كل خطوة.

أسئلة تفكير (اكتب إجابة في سطرين)

1. إيه معنى ACID؟
2. ليه الـ serial schedule دايماً صح؟

ماذا تسلّم؟

- ملف SQL بيشتغل من الأول للآخر.
- فقرة قصيرة (٥ أسطر) بتلخّص النتيجة وإيه اللي اتعلمته.

الحل النموذجي

جرّب بنفسك الأول، وبعدين قارن. الأرقام الحقيقية في النتيجة تحت.

الكود

```
CREATE TABLE account (id INTEGER PRIMARY KEY, balance INTEGER CHECK (balance >= 0));
INSERT INTO account VALUES (1,100), (2,50);
BEGIN;
UPDATE account SET balance = balance - 30 WHERE id = 1;
UPDATE account SET balance = balance + 30 WHERE id = 2;
COMMIT;
BEGIN;
UPDATE account SET balance = balance + 500 WHERE id = 2;
UPDATE account SET balance = balance - 500 WHERE id = 1;
ROLLBACK;
SELECT * FROM account ORDER BY id;
UPDATE account SET balance = 70 WHERE id = 1;
UPDATE account SET balance = 70 - 10 WHERE id = 1;
UPDATE account SET balance = 70 + 20 WHERE id = 1;
SELECT balance AS lost_update FROM account WHERE id = 1;
```

النتيجة

```
Error: CHECK constraint failed: balance >= 0
id | balance
---+--------
1  | 70
2  | 80
(2 rows)
lost_update
-----------
90
(1 row)
```

ملاحظات على الإجابات

- Atomicity وConsistency وIsolation وDurability.
- لأن مفيش تداخل بين الـ transactions.

Credits: topic follows the matching topic of LiU TDDD12 Database Technology and the Elmasri and Navathe textbook. Questions, data, code and explanations are original.