مشروع نهاية الفصل 7

مقرر قواعد البيانات — Normalization — تنضيف جدول

الهدف: تكتشف dependencies وتقسّم جدول مكرر لـ 3NF.
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)

- جدول flat فيه order وcustomer وproduct.

2 Dependencies (Dependencies)

- افحص customer_id -> customer_name بـ COUNT(DISTINCT).

3 قسّم (Split)

- اعمل customer وproduct وorder_line.

4 اتأكد (Check)

- الجداول واعدّ الصفوف. join

بونس

- بونس: ضيف PRIMARY KEY لكل جدول جديد.

أسئلة تفكير (اكتب إجابة في سطرين)

1. إيه الـ transitive dependency؟
2. إيه معنى lossless؟

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 flat (order_id INTEGER, customer_id INTEGER, customer_name TEXT, product_id
INTEGER, product_title TEXT, qty INTEGER);
INSERT INTO flat VALUES (1,1,'Mona',1,'Pen',2), (2,1,'Mona',2,'Book',1),
(3,2,'Omar',2,'Book',1), (4,2,'Omar',1,'Pen',4);
SELECT customer_id, COUNT(DISTINCT customer_name) AS names FROM flat GROUP BY
customer_id;
CREATE TABLE customer AS SELECT DISTINCT customer_id, customer_name FROM flat;
CREATE TABLE product AS SELECT DISTINCT product_id, product_title FROM flat;
CREATE TABLE order_line AS SELECT order_id, customer_id, product_id, qty FROM flat;
SELECT COUNT(*) AS rows_back FROM order_line o JOIN customer c ON c.customer_id =
o.customer_id JOIN product p ON p.product_id = o.product_id;
```

النتيجة

```
customer_id | names
------------+------
1           | 1
2           | 1
(2 rows)
rows_back
---------
4
(1 row)
```

ملاحظات على الإجابات

- X -> Y و Y -> Z يعني X -> Z ومشكلتها في 3NF.
- الـ join بيرجّع الصفوف الأصلية بالظبط.

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