مختصر كتابي

Database, DBMS, and Storage Engine



ما الفرق بين البيانات، والبرنامج الذي يديرها، والمحرك الذي يخزّنها؟

حين نقول إن البرنامج جلب المخزون من الـDatabase، فإن الجملة تخفي أكثر من دور. إن لم نفرّق بينها، فقد نظن أن البرنامج الذي نكتب فيه SQL هو نفسه الذي يخزّن البيانات، ويصعب علينا حينها معرفة مصدر المشكلة.

تخيّل مكتبة: فيها الكتب ومعلوماتها، ومكتب استعلامات يستقبل طلبك، وطريقة لترتيب الرفوف والوصول إلى كل كتاب. **الـDatabase** هي الكتب ومعلوماتها: مجموعة بيانات منظمة، كبيانات العملاء والمنتجات والطلبات والعلاقات بينها. أما البرنامج الذي يتحكم في البيانات ويستقبل طلبات القراءة والكتابة فهو **نظام إدارة الـDatabases** (DBMS)؛ يفهم الأمر، ويراجع الصلاحيات (Privileges)، ويختار طريقة جلب البيانات، وينسّق العمل مع العمليات الأخرى.

في **نموذج العميل والخادم** (Client-Server) يرسل المتجر Query إلى Database Server الذي يعمل عليه Postgres، فيصل الـServer إلى Row المخزون ويعيد قيمة واحدة. لا يحتاج المتصفح إلى فتح ملفات البيانات، ولا يحتاج التطبيق إلى معرفة الصفحة التي يقع فيها الـRow. وقد يكون العميل والـServer على الجهاز نفسه أو على جهازين مختلفين.

```
SELECT quantity FROM store.inventory WHERE product_id = 101;
```

أما الوصول إلى الـRow وتغييره وتنظيم الـIndexes، فهو دور **محرك التخزين** (Storage Engine)، وقد يشارك أيضًا في المعاملات والـLocks بحسب تصميم النظام. في MySQL مثلًا، InnoDB اسم محرك التخزين لا اسم النظام كله، فلا تصح مقارنته باسم منتج كامل. وهذه أدوار برمجية، لا أشخاص منفصلون ولا أجهزة نشتريها، وتنظيمها يختلف من نظام لآخر.

وفي **الـEmbedded Database** مثل SQLite، يستخدم التطبيق Library داخل العملية نفسها تقرأ ملف البيانات وتكتب فيه، دون Server منفصل. وهذا لا يعني أن البيانات مؤقتة أو قليلة، ولا أنه لا توجد معاملات أو Indexes؛ فالضمانات تُعرف من تصميم المنتج وإعداداته.

```
import sqlite3
from contextlib import closing
with closing(sqlite3.connect('notes_demo.sqlite')) as con:
    con.execute('CREATE TABLE IF NOT EXISTS notes(id INTEGER PRIMARY KEY, body
TEXT)')
    con.execute('INSERT INTO notes(body) VALUES (?)', ('First note',))
    con.commit()
with closing(sqlite3.connect('notes_demo.sqlite')) as con:
    print(con.execute('SELECT count(*) FROM notes').fetchone()[0])
```

يطبع المثال واحدًا إن كان الملف جديدًا، واثنين إن أضفت ملاحظة أخرى. بقيت البيانات لأنها سُجّلت في الملف، لا لأن نافذة العرض ما زالت مفتوحة.

- إغلاق برنامج العميل ليس أمر حذف للبيانات؛ في Postgres الـServer منفصل عن العميل.
- تعديل لم يُنفَّذ له `COMMIT` قد يضيع إذا انقطع الـConnection.
- رحلة الطلب: المتصفح، ثم التطبيق، ثم الـ DBMS، ثم محرك التخزين، ثم تعود النتيجة للتطبيق.
- في SQLite تختفي خطوة الـServer المنفصل، ويصبح محرك التخزين داخل عملية التطبيق.
- تثبيت عميل جديد على جهازك لا ينقل بيانات Server بعيد إليك؛ أنت تغيّر أداة الطلب فقط.
- إن منعت مشكلةُ الـConnection الـQuery من الوصول، فتغيير الـIndex لن يصلحها.

الخلاصة: البيانات، ونظام إدارتها، ومحرك تخزينها أدوار مختلفة، والتفريق بينها يساعدك على تحديد مصدر أي مشكلة وطرح أسئلة أدق عنها.