مختصر كتابي

From Requirements to Schema



كيف تتحول قصة المتجر إلى Tables وعلاقات واضحة؟

لو وضعت طلب ليلى كله في Row واحد، باسمها واسم السماعة وسعرها واسم لوحة المفاتيح وسعرها، فماذا تفعل حين يأتي طلب فيه عشرة منتجات؟ هل تضيف Columns جديدة كل مرة؟ وإن غيّرت ليلى بريدها، هل تبحث عن كل طلب كتبت فيه بياناتها؟ لذلك نحدد قبل كتابة أي أمر ما الذي نمثّله بالضبط.

نموذج البيانات (Data Model) هو طريقة تمثيل الأشياء المهمة و**العلاقات** (Relationships) بينها والقواعد التي تحكمها، كرسم البيت قبل بنائه. في قصتنا خمسة أشياء: العميل، والمنتج، والمخزون، والطلب، وبنود الطلب. ولا يلزم أن يصبح كل اسم Table؛ نختار الشكل المناسب للأسئلة والتغييرات المتوقعة. وُضع المخزون في Table مستقل لأن رقمه يتغير كثيرًا، ولأننا سنحتاجه في تجارب التزامن، لا لأن كل متجر يجب أن يفصله.

الطلب يشبه رأس الفاتورة: رقمه وصاحبه ووقته وحالته. وكل سطر في الفاتورة بند: منتج وعدد قطعه وسعره وقت الشراء. إن زادت المنتجات أضفنا Rows في Table البنود، ولم نغيّر شكل الـTable. بين العميل والطلب علاقة **واحد إلى متعدد** (One-to-Many)، وبين الطلب والمنتج علاقة **متعدد إلى متعدد** (Many-to-Many) يربطها Table البنود.

```
CREATE TEMP TABLE m_customers(id bigint PRIMARY KEY, name text);
CREATE TEMP TABLE m_products(id bigint PRIMARY KEY, name text, price
numeric(12,2));
CREATE TEMP TABLE m_orders(id bigint PRIMARY KEY, customer_id bigint REFERENCES
m_customers);
CREATE TEMP TABLE m_items(order_id bigint REFERENCES m_orders,
 product_id bigint REFERENCES m_products, quantity integer, unit_price
numeric(12,2),
 PRIMARY KEY(order_id,product_id));
INSERT INTO m_customers VALUES(1,'Layla');
INSERT INTO m_products VALUES(101,'Headphones',500),(102,'Keyboard',750);
INSERT INTO m_orders VALUES(10,1);
INSERT INTO m_items VALUES(10,101,1,500),(10,102,2,750);
SELECT sum(quantity*unit_price) AS total FROM m_items WHERE order_id=10;
```

طلب ليلى رقم 10 فيه سماعة واحدة بخمسمئة جنيه، ولوحتا مفاتيح سعر الواحدة سبعمئة وخمسون، فالإجمالي ألفا جنيه. الربط يتم بالمعرّف (ID) لا بالاسم، فقد يوجد شخصان باسم ليلى.

- حين تزيد الأشياء وتنقص، فالحل Rows مرتّبة لكل منها معرّف واضح، لا Columns إضافية.
- حالة الطلب واحدة من: `Pending` أو `Accepted` أو `Cancelled`، و `Accepted` تعني أن الطلب تأكد وحُجزت قطعه.
- الدفع يتم عند شركة أخرى، وردّها يُحفظ في Table مستقل.

- وجود Column اسمه «الحالة» لا يكفي، فقد تُكتب فيه قيمة خاطئة فيقبلها.
- حدود المثال مقصودة: عميل مسجّل، ومنتج بلا مقاسات، ومخزن واحد؛ وتغيير أي منها يستدعي مراجعة التصميم.
- اسأل عن كل Row: ماذا يمثّل؟ ما الذي يميزه؟ وماذا لو كان منه ألف؟

طلب جديد من ليلى فيه سماعتان لا يحتاج عميلًا ولا منتجًا جديدًا؛ نضيف طلبًا وبندًا واحدًا كميته اثنتان:

```
INSERT INTO m_orders VALUES(11,1);
INSERT INTO m_items VALUES(11,101,2,500);
```

الخلاصة: ارسم العلاقات قبل الكود، وافصل الطلب عن بنوده لأن كلًا منهما يتغير بطريقة مختلفة، واربط بالمعرّفات لا بالأسماء. أما كفاية المخزون فشرط آخر يُحمى عند تشغيل البرنامج.