

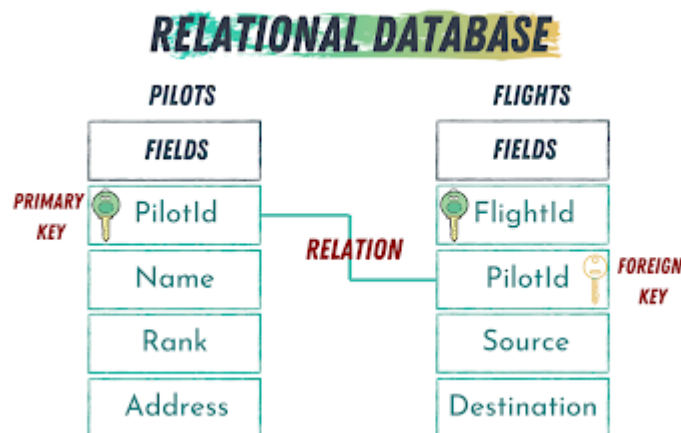
لو عندك مكتبة ضخمة. السؤال هو: **هترتب الكتب دي إزاي؟**
هل هتخطهم في رفوف منظمة، كل رف ليه عنوان وتصنيف واضح (زي مكتبة الجامعة)؟ ولا هتسيبهم في صناديق مفتوحة تقدر تحط فيها أي كتاب بأي شكل (زي مكتبة البيت)؟
بنفس المنطق، في عالم قواعد البيانات عندنا نوعين أساسيين:

- **SQL**: بيحب النظام والترتيب، بيعتمد على الجداول والعلاقات الواضحة.
- **NoSQL** مرن، مش بيقيدك بشكل معين، تقدر تخزن البيانات زي ما هي.

--: SQL VS NOSQL

SQL: يعني إيه وليه نستخدمه؟

(Structured Query Language) : هي لغة معمولة مخصوص عشان تدير قواعد البيانات العلائقية.



- البيانات بتتخزن في جداول (Tables).
- كل جدول فيه صفوف (Rows) وأعمدة (Columns).
- الجداول دي بتترابط ببعض من خلال مفاتيح:
 - **Primary Key**: العمود اللي فيه قيم فريدة.
 - **Foreign Key**: نفس العمود لما يربط جدول بجدول تاني.

أشهر أمثلة على قواعد بيانات **MySQL**: **SQL** و **PostgreSQL**.

مميزاته:

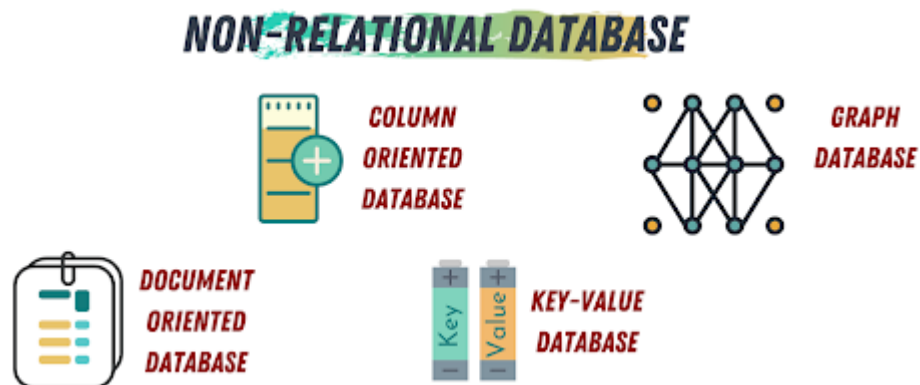
- أسرع وأدق في الاستعلامات المعقدة.
- ثابت جدًا ومناسب للبيانات المنظمة.
- مثالي في التعامل مع الـ **Transactions** (زي البنوك أو أنظمة الدفع).

عيوبه:

- محتاج تحدد الـ **Schema** و (بنية البيانات) من البداية.
- التوسع عنده رأسي (تزود إمكانيات السيرفر نفسه).

NoSQL: هنا تيجي المرونة

(Not Only SQL) مصطلح يشمل أي قاعدة بيانات غير علائقية. هنا البيانات مش مضطرة تتخزن في جداول، ممكن تتخزن بأكثر من شكل:



- **Document-Oriented** : زي **MongoDB** (بتخزن البيانات كـ JSON).
- **Key-Value Stores** : زي **Redis** (زي **Dictionary** في **Python**).
- **Column-Oriented** : زي **Cassandra** (بيخزن البيانات في أعمدة)

- **Graph Databases** : زي Neo4j (مناسبة جدا للشبكات الاجتماعية).

مميزاته:

- مرن جدا ومش محتاج Schema ثابت.
- بيستوعب بيانات ضخمة (Big Data) ومن مصادر مختلفة.
- التوسع عنده أفقي (تزود سيرفرات جديدة بسهولة).

عيوبه:

- أقل دقة واتساق مقارنة بSQL.
- مش بنفس القوة في الاستعلامات المعقدة.

الفرق في الأداء والتوسع

- **SQL**: سريع جدا في قراءة البيانات المنظمة والمعقدة، بس محتاج تجهيز مسبق. بيتوسع رأسي (تزود قوة السيرفر).
- **NoSQL**: مثالي للبيانات الكبيرة والمتنوعة، مرن وسهل التوسع أفقي (توزيع الحمل على أكثر من سيرفر).

إمتى تستخدم SQL وإمتى NoSQL؟

- لو بياناتك منظمة وثابتة (زي البنوك، المستشفيات، أنظمة الدفع) افضل تستخدم **SQL**
- لو بياناتك متنوعة وضخمة وجاية من مصادر كتير (زي السوشيال ميديا، إنترنت الأشياء) افضل استخدم **NoSQL**

الخلاصة :

مفيش واحد أحسن من الثاني بشكل مطلق. كل واحد ليه مكانه الصح.
قبل ما تختار, اسأل نفسك:

- هل بياناتي منظمة وواضحة العلاقات؟ : **SQL** .
- هل بياناتي ضخمة, متنوعة, وبتتغير كثير؟ : **NOSQL**

المصادر :

<https://builtin.com/data-science/sql-vs-nosql>