

# Types of Databases

---

## ◆ Relational Databases (RDBMS)

قاعدة البيانات العلائقية (Relational Database) هي نوع من قواعد البيانات ييخزن البيانات في جداول (Tables) مكونة من صفوف (Rows) و أعمدة (Columns)، والعلاقات (Relations) بين الجداول بتحدد بالمفاتيح (Primary Key & Foreign Key).

النظام اللي بيدير النوع ده من قواعد البيانات بيتسمى RDBMS – Relational Database Management System.

---

## Advantages of Relational Databases (RDBMS)

1. الهيكلية المنظمة (Structured Data Organization)
    - البيانات بتتخزن في Rows & Columns → Tables، وده بيخليها سهلة الفهم والإدارة.
  2. العلاقات (Relationships)
    - بتسمح بربط البيانات باستخدام Primary Key و Foreign Key.
    - مثال: جدول Customers مرتبط بجدول Orders.
  3. التكامل والاتساق (Data Integrity & Consistency)
    - عن طريق Constraints زي (NOT NULL, UNIQUE, FOREIGN KEY) البيانات بتفضل صحيحة ومترابطة.
  4. لغة الاستعلام الموحدة (SQL Standard)
    - التعامل كله بيبكون من خلال لغة موحدة اسمها SQL، متعلمة ومعروفة عالميًا.
  5. منع التكرار (Data Redundancy Control)
    - بفضل Normalization، بتمنع تكرار البيانات وتقلل مشاكل التناقض.
  6. المعاملات (Transactions)
    - بتدعم مفهوم ACID Properties (Atomicity, Consistency, Isolation, Durability) لضمان إن العمليات تتم صح 100%.
    - مثال: لو عميل بيحوّل فلوس، العملية كلها تتم أو كلها تتلغي، مفيش نص نص.
  7. التوسع الرأسى (Vertical Scalability)
    - ممكن تزود قدرات السيرفر (CPU, RAM) وال Database تفضل شغالة بكفاءة.
  8. الدعم الواسع (Wide Support & Tools)
    - موجودة في كل الشركات ومجتمع المطورين عنده خبرة كبيرة بيها.
    - مدعومة من أدوات قوية زي Power BI, Tableau، وغيرهم.
-

## ◆ أشهر أنظمة RDBMS

PostgreSQL	MySQL	SQL Server	العنصر
Open Source community	Oracle (Open Source كان) ولسه منه (نسخة مجانية)	Microsoft	المالك (Owner)
مستخدم في الأنظمة المعقدة والـ Startups Enterprise والـ	مستخدم بكثرة في تطبيقات الويب (Web apps)	مستخدم بقوة في المؤسسات الكبيرة (Enterprise)	الانتشار (Usage)
مجاني بالكامل (Open Source)	مجاني (Community Edition) + مدفوع للشركات	مدفوع (لكن في نسخة مجانية اسمها SQL Server Express)	التكلفة (Cost)
ممتاز مع البيانات الضخمة والتحليلات المعقدة	سريع وخفيف، ممتاز للـ Websites	قوي جدًا مع الـ Enterprise apps ويدعم التحليلات (BI)	الأداء (Performance)
دعم قوي من المجتمع (Community)	دعم واسع من المجتمع + Oracle	دعم رسمي من Microsoft	الدعم (Support)
يدعم JSON, GIS, Extensible Big Data، مناسب للـ	خفيف، بسيط، سهل الدمج مع PHP و WordPress	تكامل مع Windows + Power BI .NET	المميزات الفريدة (Unique Features)
أنظمة تحليل بيانات، GIS systems، شركات تقنية ناشئة أو كبيرة	مواقع ويب، CMS، تطبيقات صغيرة ومتوسطة	بنوك، مؤسسات حكومية، شركات ضخمة	المجالات الأنسب (Best Use Cases)

## ◆ Brief Intro to Non-Relational (NoSQL) Databases

- الـ **Relational Databases (RDBMS)** بتخزن البيانات في شكل **Tables → Rows & Columns** والعلاقات بتكون عن طريق **Primary Key / Foreign Key**.
- لكن **NoSQL Databases** مش معتمدة على الجداول، بتخزن البيانات بطرق مختلفة حسب النوع.
- اتعملت عشان تحل مشاكل الـ **Big Data** والـ **Scalability** اللي صعب على الـ RDBMS التقليدية.

## Advantages of NoSQL

### 1. مرونة (Flexibility):

- مش محتاج "Schema ثابت"، يعني تقدر تضيف Data مختلفة في نفس الـ Collection.
- مثال: مستخدم عنده (Name, Age) ومستخدم ثاني عنده (Name, Age, Address, Phone).

### 2. التوسع الأفقي (Horizontal Scalability):

- تقدر توزع البيانات على **Servers** كتير (Sharding)، عشان تتحمل ملايين أو مليارات البيانات.

3. سرعة عالية (High Performance):
    - معمولة عشان تستجيب بسرعة في تطبيقات الويب والموبايل.
  4. أنواع بيانات متنوعة (Variety of Data):
    - تقدر تخزن نصوص، صور، JSON، بيانات جغرافية، ... إلخ.
- 

## NoSQL Databases

### 1. Document-oriented

- بتخزن البيانات في شكل Documents (غالبًا JSON أو BSON).
- مثال: MongoDB, CouchDB.

### 2. Key-Value Stores

- كل Data بتتخزن في شكل Key → Value زي Dictionary.
- مثال: Redis, DynamoDB.

### 3. Column-oriented

- بتخزن البيانات بالـ Columns بدل الـ Rows.
- مناسبة للتحليلات الضخمة.
- مثال: Cassandra, HBase.

### 4. Graph Databases

- بتخزن البيانات في شكل Nodes & Edges (علاقات رسومية).
  - ممتازة للشبكات الاجتماعية وتحليل العلاقات.
  - مثال: Neo4j.
- 

## متى نستخدم NoSQL؟

- لما البيانات غير منظمة (Unstructured) أو شبه منظمة (Semi-structured).
  - في تطبيقات الويب والموبايل اللي محتاجة سرعة عالية جدًا.
  - في الأنظمة اللي بتتعامل مع Big Data.
  - لو في تغيير مستمر في شكل البيانات (Schema-less).
- 

## أمثلة مشهورة

- MongoDB → الأكثر شهرة، Document-based.
- Redis → Key-Value سريع جدًا كـ.
- Cassandra → قوي جدًا مع البيانات الضخمة.
- Neo4j → والعلاقات Graphs متخصص في الـ.

---

## ♦ Important concepts

### Consistency (الاتساق)

- **المعنى:** البيانات في قاعدة البيانات لازم تظل صحيحة ومتوافقة بعد أي عملية.
- مثال: لو عندك حساب بنكي، مجموع أرصدة الحسابات لازم يظل مضبوط بعد كل عملية تحويل.
- ضمن مفهوم الـ **ACID**، الاتساق يعني أي عملية تغيّر البيانات لازم تترك النظام في حالة صحيحة.

---

### ACID Properties

ده اختصار لأربع خصائص مهمة في الـ RDBMS لضمان سلامة البيانات:

#### 1. Atomicity (الذرية / الذرية)

- العملية تتم بالكامل أو لا تتم إطلاقاً. (All or None)
- مثال: تحويل \$100 من حساب A إلى B → لو حصل خطأ، العملية كلها تلغى.

#### 2. Consistency (الاتساق)

- البيانات تبقى صحيحة بعد كل عملية.
- مثال: رصيد الحسابات لا يمكن أن يصبح سالب إذا القاعدة تمنع ذلك.

#### 3. Isolation (العزل)

- العمليات المتزامنة لا تؤثر على بعضها.
- مثال: لو شخصان يحاولان تعديل نفس الحساب في نفس الوقت، كل عملية تتم بشكل مستقل حتى الانتهاء.

#### 4. Durability (الدوام / الديمومة)

- بمجرد اكتمال العملية، نتائجها تحفظ حتى لو حصل عطل أو انقطاع.
- مثال: بعد تحويل الأموال بنجاح، حتى لو وقع انقطاع كهرباء، التغييرات تبقى محفوظة.

---

### Vertical Scalability (التوسع الرأسي)

- **المعنى:** زيادة قدرات السيرفر الواحد (CPU, RAM, Storage) لتحمل المزيد من البيانات أو المستخدمين.
- مثال: رفع RAM و CPU على قاعدة البيانات نفسها لتحمل عدد مستخدمين أكبر.
- غالباً يكون أسهل لكن محدود → السيرفر له حد معين.

---

### Horizontal Scalability (التوسع الأفقي)

- **المعنى:** إضافة المزيد من السيرفرات لتوزيع البيانات والجمل.

- مثال: بدل أن نرفع قدرات سيرفر واحد، نضيف 5 سيرفرات جديدة ووزع البيانات بينهم (Sharding).
  - ممتاز لـ **NoSQL** أو أنظمة الـ Big Data.
  - يضمن سرعة أعلى وقدرة على معالجة ملايين العمليات في نفس الوقت.
-