

**Software\_Engineering**

**Lab4**

**Heyam Hameed**

## مقارنة قواعد البيانات بشكل عام

### ١. قواعد البيانات العلائقية (Relational Databases)

**التعريف:** تخزن البيانات في جداول مترابطة، بحيث كل جدول يحتوي على صفوف (Rows) وأعمدة (Columns)، والعلاقات بين الجداول تحدد كيفية ارتباط البيانات ببعضها.

**أمثلة شائعة:** MySQL، PostgreSQL، Oracle، SQL Server.

**لغة الاستعلام:** SQL (Structured Query Language) للتعامل مع البيانات، سواء للإضافة أو التعديل أو الاستعلام.

**العلاقات:** قوية جدًا، يمكن إنشاء علاقات معقدة بين الجداول مثل One-to-One، One-to-Many، Many-to-Many.

**سلامة البيانات:** ممتازة، توفر قواعد البيانات العلائقية ضمان الاتساق (Consistency) وسلامة البيانات عبر القيود والقواعد (Constraints).

**الأداء والتوسع:** عادةً أفضل عند التوسع الرأسي (Vertical Scaling)، أي زيادة موارد الخادم مثل المعالج والذاكرة.

**المرونة:** أقل مرونة، يجب تحديد هيكل البيانات (Schema) مسبقًا قبل إدخال البيانات.

**الاستخدام الشائع:** مناسب للتطبيقات التي تحتاج دقة وسلامة عالية للبيانات، مثل الأنظمة المصرفية، إدارة المخازن، والتطبيقات التجارية الكبيرة.

---

### ٢. قواعد البيانات غير العلائقية (NoSQL)

**التعريف:** تخزن البيانات بصيغ غير جدولة، مثل المستندات (Documents)، المفتاح-القيمة (Key-Value)، الأعمدة (Column-Family)، أو الرسوم البيانية (Graph).

**أمثلة شائعة:** MongoDB، Redis، Cassandra، CouchDB.

لغة الاستعلام: تختلف حسب نوع قاعدة البيانات، ولا تعتمد على SQL، مثل MongoDB Query Language.

العلاقات: ضعيفة أو محدودة مقارنة بالعلائقية، بعض الأنواع تدعم Graph Relations بشكل خاص.

سلامة البيانات: أقل من العلائقية في الاتساق الكامل، لكنها تدعم أداء سريع ومرونة عالية في البيانات الكبيرة.

الأداء والتوسع: ممتازة عند التوسع الأفقي (Horizontal Scaling)، أي توزيع البيانات على أكثر من خادم.

المرونة: عالية، يمكن تخزين البيانات بدون تحديد هيكل محدد مسبقاً (Schema-less).

الاستخدام الشائع: مناسب للتطبيقات التي تحتاج سرعة عالية، مثل تطبيقات التواصل الاجتماعي، أنظمة التخزين الكبيرة، و Big Data.

---

## قواعد البيانات التي يتعامل معها Django

### :Django

هو إطار عمل ويب بلغة Python، ويستخدم (ORM (Object Relational Mapping لربط الكائنات في Python مع قواعد البيانات. بشكل رسمي، Django يدعم قواعد البيانات العلائقية فقط:

#### ١. SQLite

قاعدة بيانات خفيفة، مدمجة مع Python. مناسبة للتطبيقات الصغيرة أو للاختبارات والتطوير المحلي.

#### ٢. PostgreSQL

قاعدة بيانات قوية ومفتوحة المصدر.

تدعم ميزات متقدمة مثل JSON، Full-text search، GIS، والوظائف المعقدة.

مناسبة للتطبيقات الكبيرة والمعقدة

### ٣. MySQL / MariaDB

سريعة وشائعة الاستخدام.

مناسبة للتطبيقات متوسطة وكبيرة الحجم.

سهولة الإدارة وتوفر دعمًا واسعًا من المجتمع.

### ٤. Oracle

مدعومة رسميًا في Django، لكنها أقل استخدامًا مقارنة بـ PostgreSQL وMySQL.

< ملاحظة: قواعد البيانات غير العلائقية (NoSQL) مثل MongoDB أو Redis تحتاج مكتبات خارجية للتعامل معها في Django، وليست مدعومة رسميًا بواسطة ORM.

---