**مقارنة بين أنواع قواعد البيانات الشائعة :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **وجه المقارنة**  **قاعدة البيانات** | **نوعها** | **دعم Django** | **المميزات** | **العيوب** | **معلومات اخرى** |
| **PostgreSQL** | علائقي (Relational) | مدعوم | - قوي جدًا للبيانات الكبيرة.  - يدعم JSON و GIS وميزات متقدمة.  - متوافق مع ACID بشكل ممتاز. | - أثقل قليلًا في الإعداد مقارنة بـ MySQL.  - يحتاج موارد أكثر قليلاً. | - موصى به للبيئات الإنتاجية. |
| **MariaDB** | علائقي (Relational) | مدعوم | - أداء جيد.  - شائع جدًا.  - بديل مفتوح المصدر لـ MySQL. | - بعض الميزات أقل تطورًا من PostgreSQL. | - مدعوم مثل MySQL. |
| **MySQL** | علائقي (Relational) | مدعوم | - شائع عالميًا.  - أداء جيد للتطبيقات المتوسطة.  - دعم مجتمعي واسع. | - قيود في بعض ميزات ORM.  - أقل مرونة في التعامل مع JSON وميزات متقدمة. |  |
| **Oracle** | علائقي (Relational) | مدعوم | - أداء عالي جدًا للشركات الكبيرة.  - أمان وموثوقية ممتازة. | - مكلف (رخصة).  - إعداد معقد. | - أقل شيوعًا في المشاريع العامة. |
| **SQLite** | علائقي (Relational) | مدعوم  ( الافتراضي ) | - مدمج في بايثون افتراضيًا.  - مناسب للتطوير والتجارب.  - لا يحتاج إعداد خارجي. | - غير مناسب للتطبيقات الكبيرة.  - ضعف في الأداء مع المستخدمين المتعددين. | - ممتاز للتجربة والتطوير المحلي. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **وجه المقارنة**  **قاعدة البيانات** | **نوعها** | **دعم Django** | **المميزات** | **العيوب** | **معلومات اخرى** |
| **Microsoft SQL Server** | علائقي (Relational) | غير مدعوم | - أداء قوي.  - تكامل ممتاز مع منتجات مايكروسوفت. | - رخصة مدفوعة للإصدارات الكاملة.  - دعم Django رسمي غير موجود (مكتبات خارجية مطلوبة). | - ممكن الربط عبر مكتبة mssql-django. |
| **MongoDB** | |  | | --- | |  |  |  | | --- | | NoSQL (وثائق) | | غير مدعوم | - مرن جدًا للبيانات غير المنظمة.  - قابل للتوسع بسهولة. | - لا يدعم المعاملات الكاملة مثل SQL.  - Django لا يدعمه رسميًا. | - يمكن الربط عبر مكتبات djongo أو mongoengine. |
| **قواعد NoSQL أخرى** (Firestore, DynamoDB) | NoSQL | غير مدعوم | - مرونة كبيرة للبيانات الكبيرة والغير منظمة. | - Django ORM مبني لـ SQL.  - يحتاج مكتبات أو Backend مخصص. | - لا يوجد دعم رسمي. |
| **قواعد SQL متخصصة** (IBM DB2, SAP SQL Anywhere) | علائقي  (Relational) | غير مدعوم | - أداء جيد في بيئات الشركات الكبيرة. | - غير مدعومة رسميًا.  - إعداد صعب. | - تحتاج Backends خارجية أو ربط مخصص. |

**جدول الاعدادات عند الربط بمشروع Django بقواعد البيانات المدعومة :**

|  |  |  |
| --- | --- | --- |
| **قاعدة البيانات** | **المكتبة المطلوبة****pip** | **الاعداد في****Settings.py** |
| |  | | --- | |  |  |  | | --- | | **SQLite** | | - مدمج مع بايثون | DATABASES = {  'default' : {  'ENGINE' : 'django.db.backends.sqlite3' ,  'NAME' : BASE\_DIR / 'db.sqlite3' ,  }  } |
| **MySQL / MariaDB** | |  | | --- | | pip install mysqlclient |  |  | | --- | |  | | DATABASES = {  'default' : {  'ENGINE' : 'django.db.backends.mysql',  'NAME' : 'Name\_of\_DB' ,  'USER' : 'User\_Name' ,  'PASSWORD' : 'password' ,  'HOST' : 'localhost' ,  'PORT' : '3306' ,  }  } |
| **PostgreSQL** | |  | | --- | | pip install psycopg2 |  |  | | --- | |  | | DATABASES = {  'default' : {  'ENGINE' : 'django.db.backends.postgresql',  'NAME' : 'Name\_of\_DB' ,  'USER' : 'User\_Name' ,  'PASSWORD' : 'password' ,  'HOST' : 'localhost' ,  'PORT' : '5432' ,  }  } |

|  |  |  |
| --- | --- | --- |
| **قاعدة البيانات** | **المكتبة المطلوبة****pip** | **الاعداد في****Settings.py** |
| **Oracle** | |  | | --- | | pip install cx\_Oracle |  |  | | --- | |  | | DATABASES = {  'default' : {  'ENGINE' :  ' django.db.backends.oracle',  'NAME' : 'Name\_of\_DB' ,  'USER' : 'User\_Name' ,  'PASSWORD' : 'password' ,  'HOST' : 'localhost' ,  'PORT' : '1521' ,  }  } |
| |  | | --- | | **Microsoft SQL Server** |  |  | | --- | |  | | |  | | --- | | pip install mssql-django |  |  | | --- | |  | | DATABASES = {  'default' : {  'ENGINE' : 'mssql',  'NAME' : 'Name\_of\_DB' ,  'USER' : 'User\_Name' ,  'PASSWORD' : 'password' ,  'HOST' : 'localhost' ,  'PORT' : '' ,  }  'OPTIONS' : {  'driver' : 'ODBC Driver 17 for SQL Server' ,  },  }  } |