12/12/2018

bella

Bella

TP DE SEMINAIRE INFO

**Nom : BELANGANI**

POSTNOM :**BELLA**

PRENOM : **PATRICK**

PROMOTION :**L2 BDD**

Tp Dirigé par le CT : **MATETA WANG**

| **Nom SGBD**  **COMPARATIF DES DIFFERENTS SGBD** |  | **Année** | **Editeur** | **Caractéristiques** | **type de logiciel** | **SQL** | **Multivalué** | **Licence** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Apache Derby |  | 1996 | Apache Software Foundation | embarqué16, relationnel, centralisé34 | Composant logiciel |  |  | Apache |
| DB2 |  | 1983 | IBM | pour entreprises, groupes de travail, particuliers35 | serveur | Fait |  | propriétaire |
| dBase |  | 1978 | Ashton-Tate | relationnel, pour particuliers36 | L4G |  |  | propriétaire |
| FileMaker Pro |  | 1985 | *FileMaker* | relationnel, pour groupes de travail37 | logiciel applicatif |  | Fait | propriétaire |
| 4D |  | 1985 | *4D* | relationnel, pour groupes de travail et entreprises38 | logiciel applicatif | Fait | Fait | propriétaire |
| Firebird |  | 1981 | Firebird Foundation | relationnel, centralisé, embarqué, pour groupes de travail et entreprises16,39 | serveur | Fait |  | Interbase |
| HSQLDB |  | 2000 | *Thomas Mueller* | relationnel, embarqué, centralisé, pour groupes de travail et particuliers16,40 | Composant logiciel | Fait |  | BSD |
| HyperFile |  | 1993 | PC Soft |  | composant logiciel41 | Fait | Fait | propriétaire |
| Informix |  | 1981 | IBM | pour entreprises, groupes de travail, distribué35 | serveur | Fait |  | propriétaire |
| Ingres |  | 1974 | *Ingres Corporation* | relationnel, spatial, centralisé, distribué42 | serveur | Fait |  | GPL |
| Caché |  | 1997 | InterSystems | objet, pour entreprises, distribué43 | serveur | Fait | Fait | propriétaire |
| MariaDB |  | 2009 | *Monty Program Ab* |  | serveur | Fait |  | GPL |
| MaxDB44,45 |  | 1977 | SAP AG et MySQL AB | objet-relationnel, pour entreprises et groupes de travail, centralisé46 | composant logiciel | Fait |  | GPL |
| Microsoft Access |  | 1992 | Microsoft | relationnel, pour particuliers et groupes de travail37,47 | L4G | Fait | Fait | propriétaire |
| Microsoft SQL Server |  | 1989 | Microsoft48 | entreprises, groupes de travail, particuliers, relationnel, distribué49 | serveur | Fait |  | propriétaire |
| MySQL |  | 1995 | Oracle Corporation50 et MySQL AB | centralisé, embarqué16, distribué, pour entreprises51, groupes de travail et particuliers45 | serveur | Fait |  | GPL |
| OpenOffice.org Base |  | 2002 | Oracle Corporation50 |  | Logiciel applicatif |  |  | LGPL |
| Oracle Database |  | 1979 | Oracle Corporation | entreprises, groupes de travail, particuliers, relationnel, spatial, distribué52 | serveur | Fait | Fait | propriétaire |
| Paradox |  | 198753 | Corel54 |  | logiciel applicatif | Fait | Fait | propriétaire |
| Pick |  | 1968 | Pick System |  | serveur |  | Fait | propriétaire |
| PostgreSQL |  | 1985 | Michael Stonebraker |  | serveur | Fait |  | BSD |
| Progress 4GL |  | 1981 | Progress Software Corporation |  | L4G |  |  | propriétaire |
| SQLite |  | 2000 | *D. Richard Hipp* | embarqué16 | composant logiciel | Fait |  | Domaine public |

**AVANTAGES ET INCONVENNIENTS DES DIFERRENTS SGBD**

SQL server

**Avantages**

•Administration aisée.

•Fonction d'audit évoluée.

•Indépendance entre les diverses bases, facilitant l'intégration de plusieurs applicatifs dans une même instance.

•Une des bases les plus performantes sous Windows en configuration par défaut.

•Optimiseur statistique enrichi à flux tendu.

•Langage T-SQL très convivial, intégration de CLR.

•Gestion de l'indexation textuelle.

•Niveau de SQL très près de la norme SQL et implémente presque toutes les possibilités de SQL.

•Services Web et Support XML.

•Ordonnanceur intégré.

•Compression des données et des sauvegardes.

**Inconvénients**

•Distributions fortement liées au système d'exploitation.

•Jungle des versions, mais fonctionnalités cantonnées dans les éditions Enterprise, Developer et Standard.

•Mono-plateforme (MS Windows).

•Depuis la version2005, plus de prise directe sur les tables système (remplacées par de vues système).

•Toujours pas de cluster (hormis en actif-passif, sur le cluster OS)

•Pas certifié SQLJ, pas d'intégration Java, orientation C#.

•Pas de contraintes d'unicité multinull.

Oracle DB

**Avantages**

• Fonction d'audit évolué.

• Row Level Storage Security (RLSS)

• Parallélisme, caches nommés, haute disponibilité.

• Procédures stockés en PL-Sql (langage propriétaire Oracle, orienté ADA) ou... en JAVA (depuis la version 8.1.7) ce qui peut s'avérer utile pour les équipes de développement

• Assistants performants via Oracle Manager Server, possibilité de gérer en interne des tâches et des alarmes

• plus portable (machines/architectures)

• plus utilisé (48% marché global),

• plus riche en termes de fonctionnalités

• plus tenable

• solutions d'accès (tous langages et toutes technologies)

• Pas mauvais sur le marché de la haute disponible, gestion des clusters…

**Inconvénients**

• Incapacité pour mettre en œuvre un traitement récursif

• incohérence et l'incompatibilité des données dans les domaines du temps et de la syntaxe de date, la concaténation de chaînes et de sensibilité à la casse

•Intolérance potentielle de l'application aux comportements dépendant du protocole durant la reprise

•Possibilité que le logiciel de cluster tombe lui-même en panne ou qu'il induise une panne dans un autre •sous-système qui n'aurait pas eu lieu dans une opération autonome

•Plus grande complexité de gestion et plus forte probabilité que des erreurs opérateur surviennent lors de l'exécution de tâches de gestion

•Possibilité que plusieurs erreurs ou une erreur opérateur grave induise une perte ou une altération des données qui n'aurait pas lieu dans une configuration autonome

•Plus grande difficulté de récupération à partir des états logiciels et/ou matériels non anticipés.

**Mysql**

•**Avantages**

Compatibilité croisée MySQL est un autre avantage. Il peut être installé dans tous les principaux systèmes d'exploitation tels que UNIX, Solaris, Linux ainsi que Windows sans perte de performance. Il travaille également avec différents développement interfaces such intransigeant que JDBC, ODBC, Pearl, Python, PHP, C ++, et bien d'autres. Ceci est principalement dû à l'API de développement qui sont intégrés avec elle.

**Inconvénients**

•L’immaturité

•MySQL n'est pas aussi mature que d'autres systèmes de gestion de base de données relationnelle. MySQL n'a pas démarré comme un système de gestion de base de données relationnelle (SGBDR), mais a changé de direction pour englober davantage de fonctionnalités. Certains SGBDR plus matures, comme PostgreSQL, sont considérés comme plus riches en fonctionnalités.

• MySQL est propriété d'Oracle plutôt que de la communauté

•MySQL n'a pas changé de direction de façon spectaculaire puisqu'il a été acquis par Oracle, mais Oracle le possède encore, ce qui rend certains développeurs nerveux.

•contrôle de SQL contraintes ne sont pas pris en charge dans MySQL.

•ne supporte pas les transactions et est sujette à la corruption de données.

**MongoDB**

**Avantages**

•La rapidité

•Les requêtes MongoDB peuvent être beaucoup plus rapides dans certains cas, d'autant plus que vos données sont généralement toutes situées en une fois et peuvent être récupérées dans une seule recherche. Cependant, cet avantage n'existe que lorsque vos données sont vraiment un document. Lorsque vos données imitent essentiellement un modèle relationnel, votre code finit par effectuer de nombreuses requêtes indépendantes afin de récupérer un document unique et peut devenir beaucoup plus lent qu'un SGBDR classique.

•La Flexibilité

MongoDB n'exige pas une structure de données unifiée sur tous les objets, de sorte qu'il n'est pas possible de s'assurer que vos données seront structurées de manière cohérente, MongoDB peut être beaucoup plus simple à utiliser qu'un SGBDR. Cependant, la cohérence des données est une bonne chose, alors, si possible, vous devez toujours essayer de vous assurer qu'une structure unifiée sera appliquée.

**Inconvénients**

•Pas de jointure

•Dans MongoDB, il n'existe aucune possibilité de jointure comme dans une base de données relationnelle. •Cela signifie que lorsque vous avez besoin de ce type de fonctionnalité, vous devez effectuer plusieurs requêtes et joindre les données manuellement dans votre code (ce qui peut entraîner un code lent et laid, et une flexibilité réduite lors de la modification de la structure).

•Utilisation de la mémoire

•MongoDB a tendance naturellement à utiliser plus de mémoire car il doit stocker les noms de clés dans chaque document. Ceci est dû au fait que la structure des données n'est pas nécessairement cohérente parmi les objets de données.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SGBDR**  **Quelques spécifications de stockage de SGBD** | **Paradox 7** | **Access 97** | **Sybase adaptive 11** | **SQL Server 7** | **Oracle 8** |
| Nature | Service de fichier | Service de fichier | Serveur de données | Serveur de données | Serveur de données |
| Nb utilisateur (max / en pratique) | 255 / 50 | 300 / 10 |  |  |  |
| Taille max de la base | illimitée | 1 Go |  |  |  |
| Taille max d'une table | 2 Go (hors BLOBS) | 1 Go |  |  |  |
| Normalisation | SQL 92 | SQL 89 ? | SQL 92 | SQL 92 | SQL 89 |
| DDL | Oui (a) | Oui | Oui | Oui | Oui |
| DML | Oui | Oui | Oui | Oui | Oui |
| DCL | Non | Oui | Oui | Oui | Oui |
| TCL | Oui (b,c) | Non | Oui | Oui | Oui |
| DCL | Non | Oui | Oui | Oui | Oui |
| TCL | Oui (b,c) | Non | Oui | Oui | Oui |
| CHAR | limité à 255 car. | limité à 255 car. | limité à 255 car. | limité à 8000 car. | limité à 2000 car. |
| VARCHAR | Non | Non | Oui | Oui | limité à 4000 car. |
| NUMERIC | Oui, avec 15 chiffres significatifs | sous types comprenant des entiers et des réels | Oui | Oui | Oui |
| INTEGER | Oui | Non | Oui | Oui | Oui |
| SMALLINT | Oui | Non | Oui | Oui | Oui |
| FLOAT | Oui, en fait NUMERIC | Voir NUMERIC | Oui | Oui | Oui |
| DATE | Oui | Non | Non | Non | Oui |
| TIME | Oui | Non | Non | Non | Non |
| TIMESTAMP | Oui | Oui | Oui | Oui | Non |
| INTERVAL | Non | Non | Non | Non | Non |
| BIT | Non | Oui | Oui |  |  |
| BOOLEAN | Oui (LOGICAL) | Oui | Non | Non | Non |
| MONEY | Oui | Oui | Oui | Oui | Non |
| BYTES | Oui | Non | Non | Non | Oui (RAW) |
| AUTOINC | Oui | Oui | Non (d) | Non (d) | Non (d) |
| BLOB | Oui (4 types différents : MEMO, MEMO FORMATE en RTF, IMAGE et BINARY) limités à 2 Go | Oui (2 types différents : MEMO, HYPERLIEN limité à 64 ko) | Oui (2 types différents TEXT IMAGE) | Oui (2 types différents IMAGE, TEXT) | Oui (7 types différents : LONG, LONG RAW, LONG VARCHAR, BFILE, BLOB, CLOB, NCLOB) |
| Autres types | OCTET (1 à 255), OLE | OLE, liste de choix | BIT, BINARY | BIT, BINARY, CURSOR, TINYINT, GUID | ROWID (N° d'enregistrement) |
| INTEGRITÉ RÉFÉRENTIELLE | Oui, stricte ou cascade (suppression et modif.) | Oui, stricte ou cascade (suppression et modif.) | Oui | Oui, pas en cascade | Oui |
| TRIGGERS | Non | Non | Oui | Oui, limités | BEFORE INSERT, BEFORE UPDATE, BEFORE DELETE, AFTER INSERT, AFTER UPDATE, AFTER DELETE |
| PROCÉDURES STOCKÉES | Non | Non | Oui, langage propriétaire Transact SQL | Oui, langage propriétaire Transact SQL | Oui, langage propriétaire PL/SQL |