**Fiche de comparaison des bases des données**

1. Présentation

Une base de données permet d'enregistrer des données de façon organisée et hiérarchisée. Il va bien falloir stocker quelque part la liste des élèves et professeur, le nombre de repas restants, les heures de passages… Les bases de données constituent le meilleur moyen de faire cela de façon simple et propre. Elle permet aussi de classé les informations afin de les retrouver plus rapidement. Une base de données seule ne suffit donc pas, il est nécessaire d'avoir également :

* un système permettant de gérer cette base ;
* un langage pour transmettre des instructions à la base de données (par l'intermédiaire du système de gestion).

Il y a deux types de base de données :

**SGBD**

Un Système de Gestion de Base de Données (SGBD) est un logiciel (ou un ensemble de logiciels) permettant de manipuler les données d'une base de données. Manipuler, c'est-à-dire sélectionner et afficher des informations tirées de cette base, modifier des données, en ajouter ou en supprimer (ce groupe de quatre opérations étant souvent appelé "CRUD", pour Create, Read, Update, Delete).  
MySQL est un système de gestion de bases de données.

**SGBDR**

Le R de SGBDR signifie "relationnel". Un SGBDR est un SGBD qui implémente la théorie relationnelle. MySQL implémente la théorie relationnelle ; c'est donc un SGBDR.

La théorie relationnelle dépasse le cadre de la simple explication, il n'est pas nécessaire de la maîtriser pour être capable d'utiliser convenablement un SGBDR. Il vous suffit de savoir que dans un SGBDR, les données sont contenues dans ce qu'on appelle des relations, qui sont représentées sous forme de tables. Une relation est composée de deux parties, l'en-tête et le corps. L'en-tête est lui-même composé de plusieurs attributs. Par exemple, pour la relation "Client", on peut avoir l'en-tête suivant :

| Numéro | Prénom | Nom | Email |
| --- | --- | --- | --- |

Quant au corps, il s'agit d'un ensemble de lignes composées d'autant d'éléments qu'il y a d'attributs dans le corps. Voici donc quatre lignes pour la relation "Client" :

| Numéro | Prénom | Nom | Email |
| --- | --- | --- | --- |
| 1 | Jean | Dupont | jdupont@email.com |
| 2 | Marie | Malherbe | mama@email.com |
| 3 | Nicolas | Jacques | Jacques.nicolas@email.com |
| 4 | Hadrien | Piroux | happi@email.com |

1. Comparaison

Il existe différent SGBD et SGBDR, ils ont tous leurs avantages et leurs inconvénients.

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | TYPE | AVANTAGE | INCONVENIENT |
| MySQL | SGBDR | * C'est un des SGBDR les plus utilisés (communauté présente en cas de problème) * Open Source * On peut modifier MySQL pour l'améliorer ou l'adapter à ses besoins * Gratuit * Très bonne intégration dans l'environnement Apache/PHP | * il ne suit pas toujours la norme officielle * il n'implémente pas certaines fonctionnalités avancées * il est très permissif, et acceptera donc des requêtes qui généreraient une erreur sous d'autres SGBDR. |
| Oracle database | SGBDR | * Oracle gère très bien de grands volumes de données | * Payant * Utiliser principalement par les entreprises * inutile d'acheter une licence oracle pour un projet de petite taille * Oracle dispose d'un langage procédural très puissant (du moins plus puissant que le langage procédural de MySQL) : le PL/SQL. |
| PostgreSQL | SGBDR | * Open Source | * Le langage procédural utilisé par PostgreSQL s'appelle le PL/pgSQL |
| MS Access | SGBDR | * l'interface graphique intuitive qui vient avec le logiciel | * édité par Microsoft * ne fonctionne que sous Windows * - beaucoup moins de fonctionnalités que les autres SGBDR |
| SQLite | SGBDR | * Pour de très petits volumes de données, SQLite est très performant | * le fait que les informations soient simplement stockées dans des fichiers rend le système difficile à sécuriser * -n’utilise pas le schéma client-serveur habituel |
| [Informix](https://fr.wikipedia.org/wiki/Informix) | SGBD | * Moteur avant-gardiste, en son temps * Administration simpliste et efficace * Performances au rendez-vous * Stabilité | * Pérennité de la solution * Très peu d'outils graphiques disponibles pour l'administration (hormis Server Studio JE payant) |
| [DB2](https://fr.wikipedia.org/wiki/IBM_DB2) | SGBD | * SQL proche de la norme * Nombreux assistants qui auraient dûs permettre une administration plus aisée (mais très gourmande en ressource) * Richesse fonctionnelle du language et des jointures * Gestion centralisée de plusieurs instances * Compression des données stockées | * Interface client Java lourd instable, peu conviviale avec des menus contextuels à rallonge. Dans les faits, seule la ligne de commandes semble stable ! * Journalisation gourmande en disques, même avec des jounaux tournants * Particulièrement instable sous Windows en configuration mémoire dynamique * Renommage de colonnes impossible = faiblesse de DDL, bien que les assistants tentent maladroitement de cacher ces faiblesses en exécutant des traitemetns lourds * Prix exhorbitant, tant au point de vue des licences que des composants matériels (RAM, CPU) à fournir pour de bonnes performances * Gestion des utilisateurs extrêmement limitative, dédiée à l'OS * Complexité du language procédural |