

# Systèmes de Gestion de Base de Données



Un système de gestion de bases de données (SGBD, DBMS en anglais) regroupe des outils logiciels permettant de gérer, manipuler, stocker et sécuriser des informations organisées dans une base de données.

La gestion d'une base de données (BDD) est complexe du fait de sa taille, du nombre de ses utilisateurs, de la nécessité d'assurer la cohérence et la sécurité des informations.

Un SGBD vise à sécuriser et fiabiliser l'accès et sécuriser l'intégrité de la base de donnée. Ce logiciel offre aux utilisateurs un langage deux fonctions :

- x Créer la structure de la base ainsi que les contraintes d'intégrité ;
- x Interroger la base de données (à l'aide de requêtes via une interface) et aussi d'effectuer des mises à jour sur les données (insertion, modification et suppression).

Les avantages qui en découlent sont considérables :

- x Le programmeur est déchargé d'une grande partie de la programmation puisqu'il n'a plus à écrire ses requêtes. Le code des applications est en conséquence notablement réduit et devient plus lisible
- x Les deux langages sont de haut niveau, l'utilisation ne nécessite donc pas beaucoup de compétence pour les opérations courantes.

## 1. Services rendus par un SGBD

Le principal objectif d'un SGBD est d'assurer l'indépendance données/traitement et assurer la sécurité de la base afin de proposer un service robuste et multi-utilisateur. Les services rendus pour atteindre ces objectifs sont :

- ➔ **la persistance des données** : Chaque transaction est tout ou rien. Soit la transaction arrive à son terme et les données sont alors modifiées, soit elle a échoué et toutes les modifications sont annulées pour restaurer la base de données dans l'état où elle était avant la transaction.
- ➔ **La sécurisation des accès** : Le SGBD gère les autorisations d'accès à la base afin de proposer un service multi-utilisateurs.
- ➔ **La gestion des accès concurrents** : Si deux transactions s'exécutent simultanément, alors leur exécution doit produire le même effet que si on les avait exécutées l'une après l'autre. Ainsi le système interdit l'altération de la base de données qui intervient lors d'accès simultanés aux données.
- ➔ **L'efficacité de traitement des requêtes** : Dans les langages de haut niveau comme SQL, l'utilisateur formule sa requête objectivement sans se préoccuper de l'aspect technique, notamment la durée d'exécution. Il appartient au SGBD de chercher une méthode optimale d'évaluation de la requête.

## 2. Architecture matérielle d'un SGBD

Un SGBD est basé sur une architecture client/serveur. Suivant les performances recherchées, on distingue alors les architectures Client/Serveur à 2 ou 3 niveaux et l'architecture distribuée.

### L'architecture Client/Serveur à deux niveaux

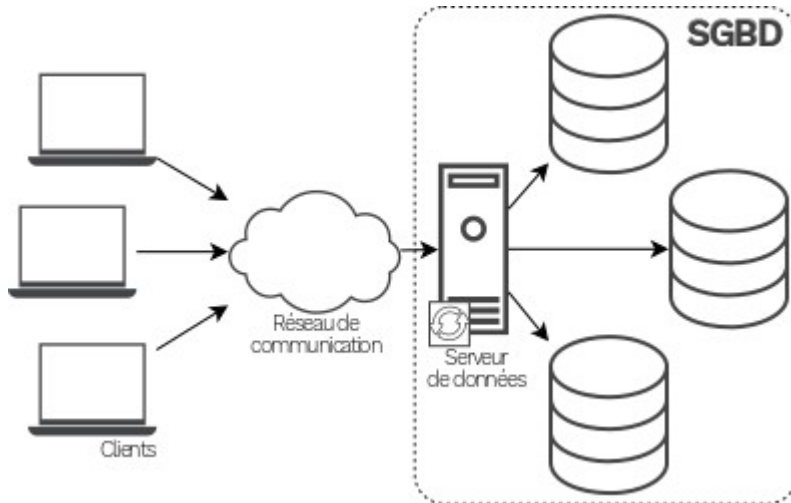


Figure 1: Architecture Client/Serveur à 2 niveaux

Cette architecture a été développée vers le début des années 90, elle est fondée sur :

- x un ordinateur puissant dénommé serveur et spécialement dédié à la gestion des données ;
- x d'autres ordinateurs qualifiés de clients dont le rôle est de prendre en charge les applications.

Les clients sont reliés au serveur à l'aide d'un réseau de communication (voir figure 1).

Avec cette architecture, les applications (situées dans les clients) transmettent leurs requêtes au SGBD (situé dans le serveur), ce dernier les traite et renvoie ensuite les résultats. Le coût croît rapidement avec l'augmentation du nombre de postes de travail (les clients). La solution s'avère donc efficace au sein des entreprises pour les applications de type département avec un nombre de clients limité.

### L'architecture Client/Serveur à trois niveaux

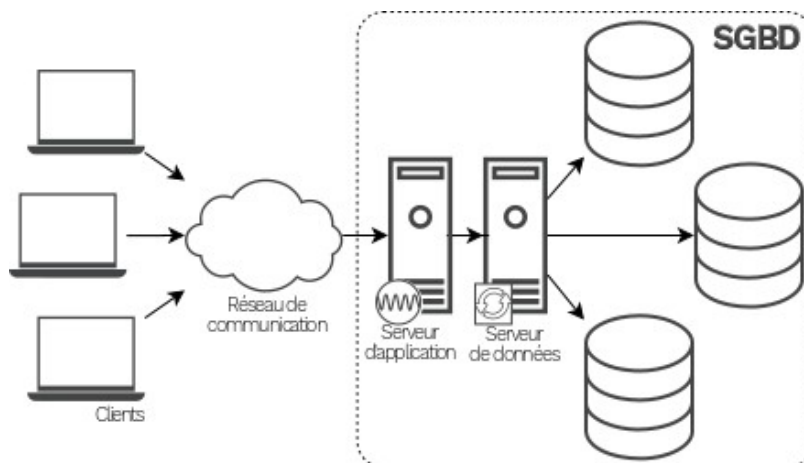


Figure 2: Architecture Client/Serveur à 3 niveaux

Elle est formée par :

- x un serveur de données : machine puissante dédiée spécialement à la gestion des données ;
- x un serveur d'applications : machine puissante pour exécuter le traitement des différentes applications ;
- x des clients : ordinateurs en principe de très faible puissance réservés à la partie présentation des applications.

Avec cette configuration, les programmes du serveur d'applications transmettent les requêtes au serveur de données, ce dernier les traite puis leur renvoie les résultats. La solution est particulièrement économique pour des applications faisant intervenir une grande partie d'une entreprise importante (nombre de clients élevé) ; dans ces conditions, les coûts sont considérablement réduits par rapport au client/serveur à deux niveaux.



## L'architecture distribuée

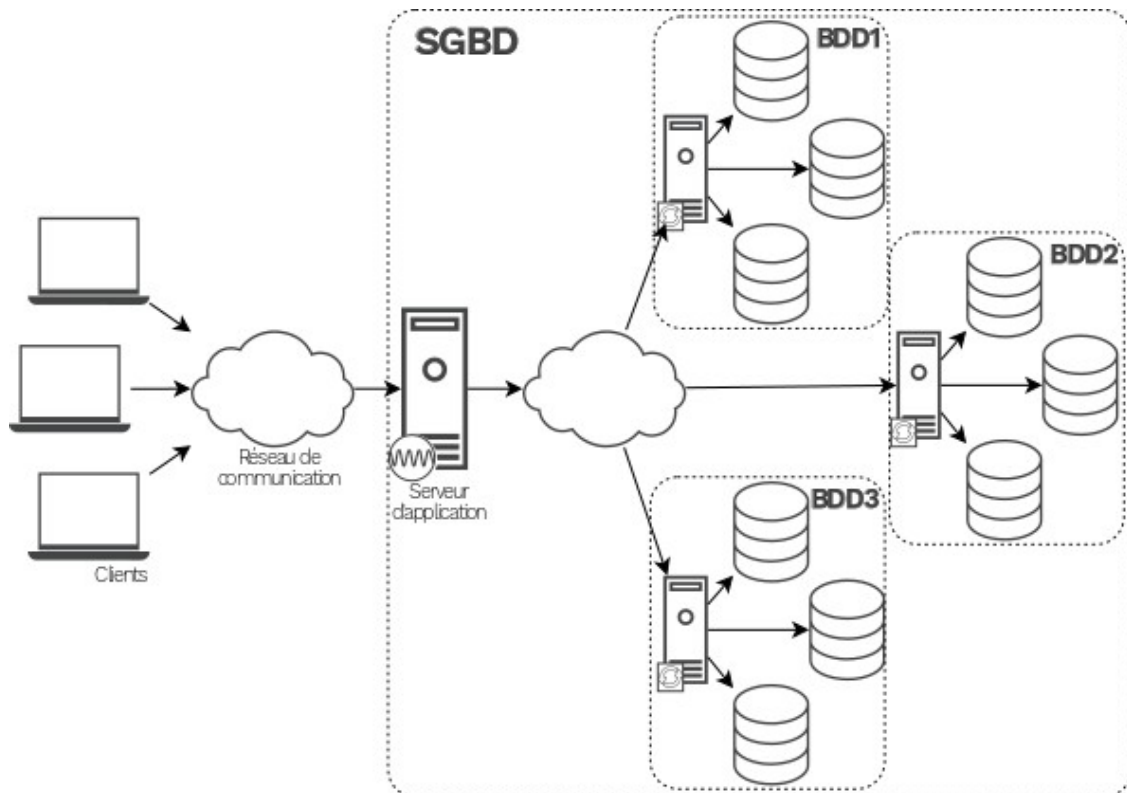


Figure 3: Architecture distribuée

La base de données est conceptuellement divisée et chaque partie est installée physiquement sur un site distinct. Ce site prend en charge la gestion des données couramment nécessaires à ses utilisateurs locaux.

On parle de base de données distribuée pour se référer à l'ensemble des bases situées sur les différents sites (dans la figure les bases BDD1, BDD2 et BDD3 forment donc la base de données distribuées).

### 3. SQL : un langage de définition de données

Directement inspiré par le modèle relationnel, SQL (sigle de *Structured Query Language*, en français *langage de requête structurée*) est un langage de programmation qui permet au client d'interagir avec le SGBD. Ce langage est normalisé par la norme ISO/IEC 9075 et sert à :

- x manipuler les données afin de rechercher, ajouter, modifier ou supprimer des données dans les bases de données relationnelles.
- x modifier l'organisation de la base de données afin de créer ou modifier des relations et des définir les contraintes d'intégrité.
- x contrôler les transactions afin de commencer et terminer des transactions et autoriser ou d'interdire l'accès à certaines données à certaines personnes.

Sur l'année de terminale dans le cours de spécialité NSI, seules seront abordées les manipulations de données avec le langage SQL.