

Systèmes de Gestion de Bases de Données — 2e

Chapitre 1 : Concepts de base

Daniel Schreurs

19 septembre 2021

Haute École de Province de Liège

1. Base de données
2. Système de gestion de base de données
3. Indépendance données
4. Architecture d'un SGBD
5. Avantages des bases de données

6. Fonctionnement d'un SGBD

Base de données

Table des matières de la section

1. Base de données

1.1 BD et SGBD

1.2 Propriétés

2. Système de gestion de base de données

3. Indépendance données

4. Architecture d'un SGBD

5. Avantages des bases de données

- *BD* : collection de données concernant un sujet enregistrées sur un support permanent accessible par l'ordinateur.
- *SGBD* : logiciel qui permet à un utilisateur d'exploiter une BD

Table des matières de la sous-section : Propriétés i

1.1 BD et SGBD

1.2 Propriétés

Être un ensemble organisé/structuré

Être un ensemble intégré

Correspondre fidèlement à la réalité

Contenir les données opérationnelles sur un sujet donné

Être multi-utilisateurs

Être non-redondante

Propriétés : Être un ensemble organisé/structuré

Stocker les données de manière à ce que leur exploitation soit efficace !

L'organisation doit tenir compte du mode d'accès le plus courant aux données.

Exemple : Catalogue d'une bibliothèque : lecture principalement
Fichier des prêts : lecture et écriture

1. Catalogue d'une bibliothèque : lecture principalement
2. Fichier des prêts : lecture et écriture

Propriétés : Être un ensemble intégré

Exemple : chaque service de la bibliothèque accède aux mêmes données centralisées.

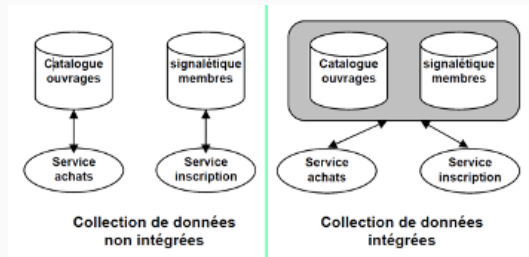


Figure 1 – Exemple d'une gestion intégrée

Propriétés : Correspondre fidèlement à la réalité

- Pour forcer les données à rester fidèles à la réalité, on définit, des contraintes d'intégrité sur la BD.
- Ces contraintes d'intégrité sont la traduction informatique des règles de fonctionnement.
- On ne stocke pas que des données, mais on stocke aussi des contraintes portant sur ces données.

Propriétés : Contenir les données opérationnelles sur un sujet donné

- Les membres et les ouvrages sont appelés *entités* dans le jargon des bases de données. Les entités sont stockées dans la base de données. Un emprunt associe un membre et un ouvrage : un emprunt est une association. Les associations sont aussi stockées dans la base de données.
- Les entités et les associations sont appelées données opérationnelles.

- Les données doivent être accessibles en même temps par plusieurs utilisateurs. Chaque utilisateur doit avoir l'impression qu'il est seul à utiliser la base.
- Le partage simultané des données implique l'existence de mécanismes de protections : confidentialité, gestions des accès concurrents, sauvegarde, reprise après panne.

- La non-redondance implique que chaque donnée ne soit stockée qu'une seule fois dans la base. Elle assure la cohérence. De plus, elle permet d'économiser la place disque.
- La redondance contrôlée peut être utilisée pour satisfaire à des besoins de vitesse de traitements et de ce fait, gagner en temps de réponse.
- Des trigger (déclencheurs) seront utilisés pour éviter les incohérences

Système de gestion de base de données

2. Système de gestion de base de données

2.1 Les Fonctions

2.2 Architecture d'un SGBD

2.1 Les Fonctions

Description et définition

Manipulation

Intégrité

Confidentialité

Concurrence d'accès

2.2 Architecture d'un SGBD

Le concepteur ou administrateur de la base doit pouvoir :

- Créer la base de données ;
- Définir ses paramètres physiques ;
- Définir les objets qu'elle contient.

La manipulation des données est un terme générique pour désigner :

- La *recherche* d'information ;
- L'*ajout* ;
- La *modification* ;
- La *suppression* de données.

Le SGBD doit :

- Permettre de définir des règles d'intégrité¹.
- Il doit aussi assurer qu'à tout moment², les valeurs présentes dans la base ou qu'un utilisateur tente d'introduire respectent ces contraintes.

1. représentant les règles de gestion du système informatisé

2. Au moins avant et après chaque transaction.

La fonction de confidentialité permet :

- D'assurer que chaque utilisateur n'effectue que les manipulations³ autorisées sur certaines données.

3. Tel que défini au slide 15.

- Les accès concurrents⁴ doivent pouvoir être traités. Sinon, ils peuvent introduire des incohérences dans la base de données !

4. Quand des utilisateurs différents tentent d'accéder en même temps aux mêmes données.

Cette liste de fonctions n'est pas exhaustive. On peut aussi évoquer :

- La reprise après panne ;
- Précompilateur ;
- Optimiseur ;
- Journalisation ;
- Gérer les accès concurrents ;
- Gérer le "cache" ;
- Gérer les sauvegardes ;
- Gérer l'espace disque, les index, etc.

Système de gestion de base de données : Architecture d'un SGBD

En général, les SGBD relationnels sont constitués de deux parties

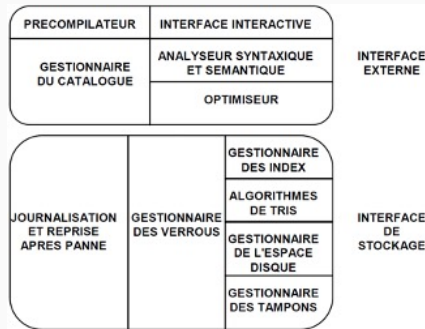


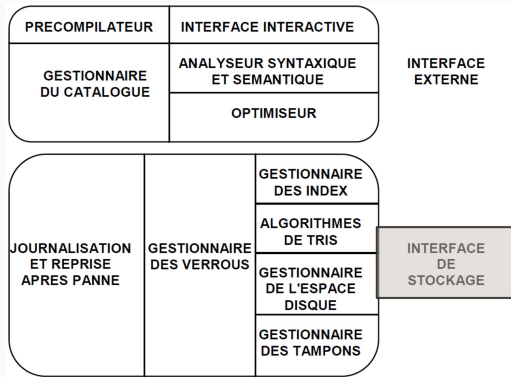
Figure 2 – Architecture d'un SGBD

Important

Le *Catalogue* ou *Méta base* ou *Dictionnaire* ou *Tables Systèmes*

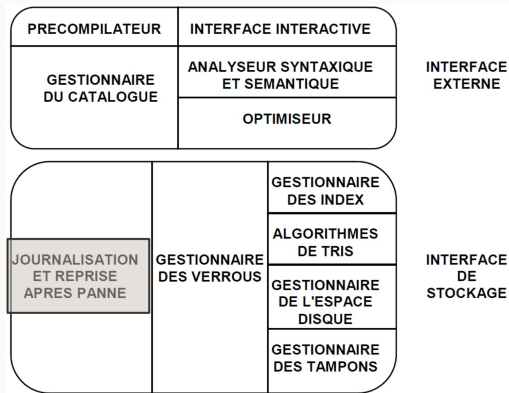
- Est mis à jour automatiquement par le SGBD
- Constitue une mini base de données contenant des informations sur la BD
- Est stocké dans la BD elle-même et peut être interrogé comme les données de la BD elle-même
- Contient la description de tous les objets présents dans la base : tables, domaines, contraintes, privilèges, fichiers, index, procédures stockées, déclencheurs, vues, objets, ...
- Il NE s'agit PAS d'un fichier spécial en dehors de la base

Système de gestion de base de données : Architecture d'un SGBD



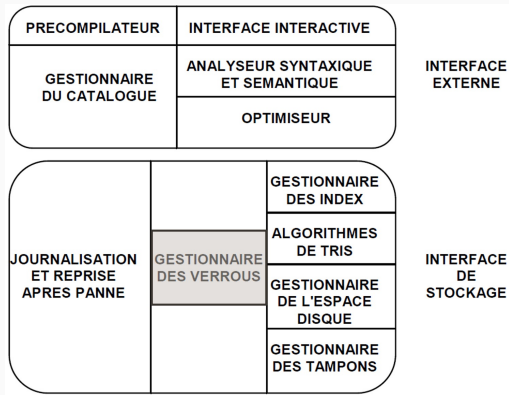
L'interface de stockage s'occupe de tout ce qui concerne l'accès aux données stockées sur disques

Système de gestion de base de données : Architecture d'un SGBD



- La journalisation et reprise après pannes assure la fonction de sécurité de fonctionnement.
- Nous ne l'aborderons pas cette année.

Système de gestion de base de données : Architecture d'un SGBD



- Le gestionnaire des verrous assure la fonction de concurrence d'accès.
- Les 4 gestionnaires restants servent à minimiser le nombre d'entrées/sorties, l'espace mémoire alloué sur disque, etc.

Indépendance données

Table des matières de la sous-section : Programmes i

3.1 Programmes

L'approche traditionnelle par les fichiers

Organisation autour d'une base de données

Indépendance des données et des programmes

Important

Il est important de pouvoir changer la structure logique ou physique d'une base sans devoir changer les programmes d'applications qui l'utilisent.

Exemple :

- On ajoute de nouvelles entités ou associations ;
- On décompose une entité en sous-entités.

Programmes : L'approche traditionnelle par les fichiers

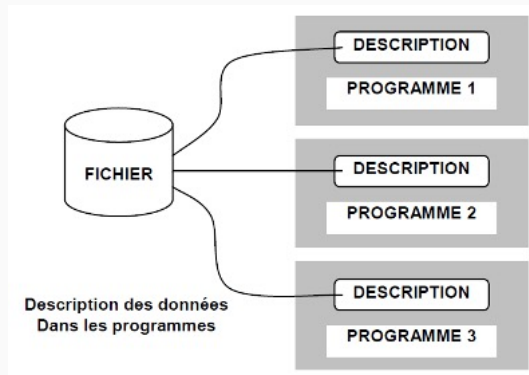


Figure 3 – L'approche traditionnelle par les fichiers

Programmes : L'approche traditionnelle par les fichiers

- Les données contenues dans les fichiers sont directement associées aux programmes par une description contenue dans le programme lui-même.
- Dans chaque programme utilisant le fichier, on aura une déclaration du type (structure par exemple) permettant la manipulation des données du fichier.
- Aucune indépendance possible entre les données et les programmes.

Programmes : Organisation autour d'une base de données

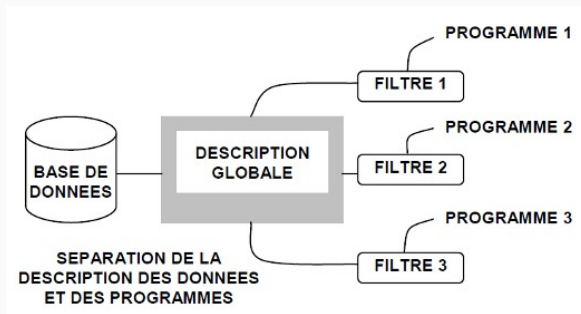


Figure 4 – Organisation autour d'une base de données

- La description des informations est centralisée ;
- Chaque programme utilise un filtre pour désigner les informations qu'il utilise.

Objectifs fondamentaux d'un système base de données :

- L'indépendance des données par rapport aux programmes de traitements
- La prise en compte des associations entre les différentes données
- Le partage simultané des données entre plusieurs utilisateurs

Programmes : Indépendance des données et des programmes

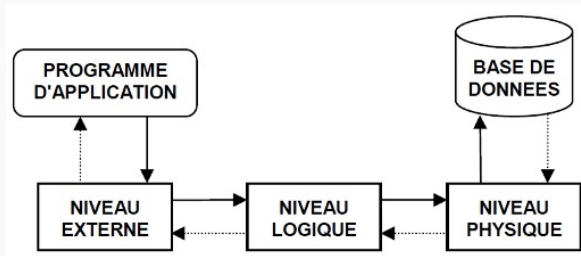


Figure 5 – Indépendance des données et des programmes

Programmes : Indépendance des données et des programmes

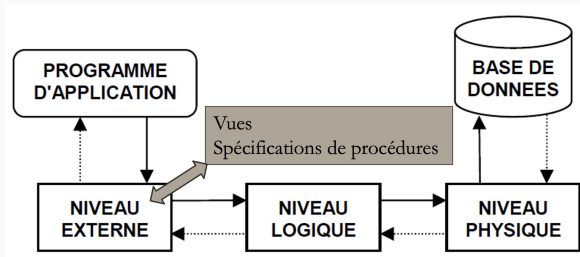


Figure 6 – Indépendance des données et des programmes

Programmes : Indépendance des données et des programmes

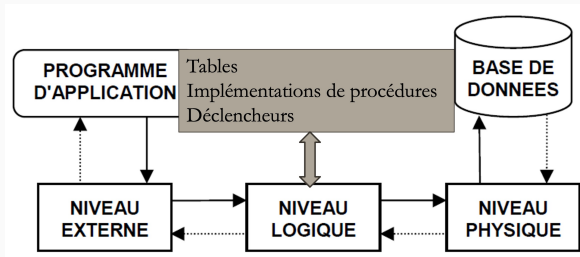


Figure 7 – Indépendance des données et des programmes

Programmes : Indépendance des données et des programmes

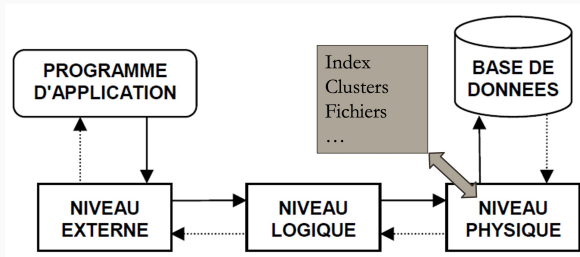


Figure 8 – Indépendance des données et des programmes

Programmes : Indépendance des données et des programmes

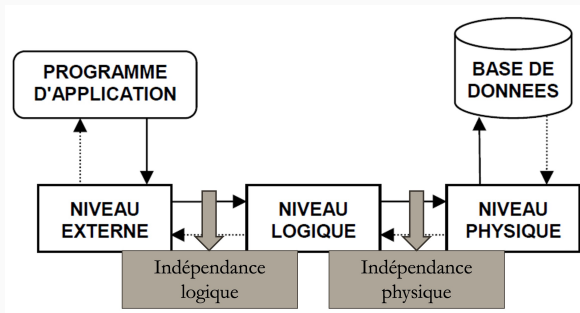


Figure 9 – Indépendance des données et des programmes

Important

- L'**indépendance des données au niveau LOGIQUE** signifie que l'on peut changer la structure logique globale sans devoir changer les programmes d'applications.
- L'**indépendance des données au niveau PHYSIQUE** signifie que la couche physique et l'organisation des données peuvent changer sans devoir changer la structure logique globale ou les programmes d'applications.

Programmes : Indépendance des données et des programmes

Pas de changement pour	Pgm	niv log	Org mém
Ajout d'1 nv pgm utilisant des données existantes	*	*	*
1 pgm utilise une nvelle représentation de données existantes	*	*	*
Ajout d'un nv pgm utilisant de nvelles données	*		
Description log glob améliorée / ajout nvelles assoc entre données	*		
Fusion de 2 BD	*	*	
Organisation physique améliorée, éventuellement nvelle représentation de données	*	*	
Méthodes d'accès modifiées	*	*	
Données déplacées sur d'autres volumes	*	*	
Logiciel est changé (nvelle version)	*	*	
Matériel est changé	*	*	

Architecture d'un SGBD

Architecture d'un SGBD : Vue d'ensemble

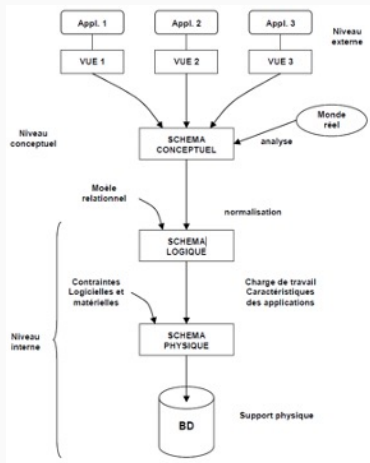


Figure 10 – Vue d'ensemble

Avantages des bases de données

- La redondance peut être réduite (exception à la règle de non-redondance : pour améliorer vitesse de traitement et temps de réponse)
- L'incohérence peut être évitée
- Les données peuvent être partagées
- Des règles de sécurité peuvent être établies
- L'intégrité peut être maintenue
- Les conflits d'accès peuvent être équilibrés

Fonctionnement d'un SGBD

Fonctionnement d'un SGBD

Déroulement d'une recherche lancée par un programme d'application

