

# **Environnement d'un SGBD**

# Plan

1. Définition
2. Historique
3. Fonctions
4. Architecture

# 1. Définition

Un SGBD est un ensemble de logiciels systèmes permettant aux utilisateurs d'**insérer, modifier et rechercher** efficacement des données spécifiques dans une grande masse d'information partagée par de multiples utilisateurs aussi bien en **interrogation** qu'en **mise à jour**. Il rend ainsi transparent le partage et donne l'illusion à chaque utilisateur qu'il est seul à travailler avec les données.  
**[Gardarin02]**

# 1. Définition

Un SGBD est un logiciel qui permet de :

- Décrire,
- Stocker,
- Modifier,
- Interroger (consulter),
- Administrer les données d'une BD.

Il permet à l'utilisateur de manipuler les données dans des termes abstraits sans se soucier de la façon de leur implémentation.

## 2. Historique

En BD, il existe 4 modèles, qui ont donné lieu à différents types de SGBD :

1. Modèle Hiérarchique → IMS (Premier SGBD, 1965)
2. Modèle en Réseau CODASYL → Socrate-Clio, IDS2 (BD SINDA - La douane).
3. Modèle Pseudo-réseau → Total, Image, ...etc.

## 2. Historique

4. Modèle Relationnel : Historiquement, c'est l'article de Codd [Codd70] du laboratoire de IBM à San José qui fonda le modèle relationnel en y exposant la simplicité de la représentation de la relation, une forme normale pour décomposer une relation afin d'éviter des redondances, et les principaux opérateurs de l'algèbre relationnelle. Les SGBD relationnels sont les plus répandus dans l'industrie.

→ Oracle, DB2, SQLServer, Sybase, Informix, Ingres, SQL/DS, ...etc. Dans le domaine du logiciel libre : PostGres, MySQL.

## 2. Historique – Bilan

- 1960 : SGF
- 1970 : Début des SGBD hiérarchiques et réseaux proches des SGF
- 1970 : Article de Codd sur la théorie des relations, fondement des BD relationnelles
- 1980 : Apparition des SGBD Relationnels sur le marché (Oracle, Ingres, Informix, Sybase, DB2, ...)
- 1990 : Début des SGBD Orientés Objet (Gemstone, O2, Orion, Objectstore, Versant, Matisse, ...)
- Ensuite : Relationnel-Objet, Semi-structuré, multimédia, ...

# 3. Fonctions

- Définition des données
- Gestion des données
- Contrôle d'intégrité
- Accès aux données
- Contrôle d'accès
- Gestion de la concurrence / transactions
- Protection contre les incidents



# 3. Fonctions

- **Description et définition**

- Définition de la structure de la BD
- Création / modification / suppression de la structure de données
- Description des structures de données :  
**dictionnaire de données**

- **Manipulation**

- Manipulation des données de la BD
- Consultation / mise à jour des données
- Manipulation navigationnelle ou déclarative
- Interface de programmation (API)
- Interface utilisateur : textuelle ou graphique

# 3. Fonctions

- **Intégrité**

- Définition des contraintes d'intégrité
- Respect des contraintes d'intégrité

- **Confidentialité**

- Les données sont partagées par plusieurs utilisateurs :
  - définition des règles de confidentialité
  - vérification du respect des règles

- **Concurrence d'accès**

- Accès concurrent aux données (des utilisateurs différents tentent d'accéder en même temps aux mêmes données)

# 3. Fonctions

- **Gestion des transactions**

- Transaction : unité de traitement cohérente et protégée
- Propriétés d'une transaction : ACID (Atomique, Cohérente, Isolée, Durable)

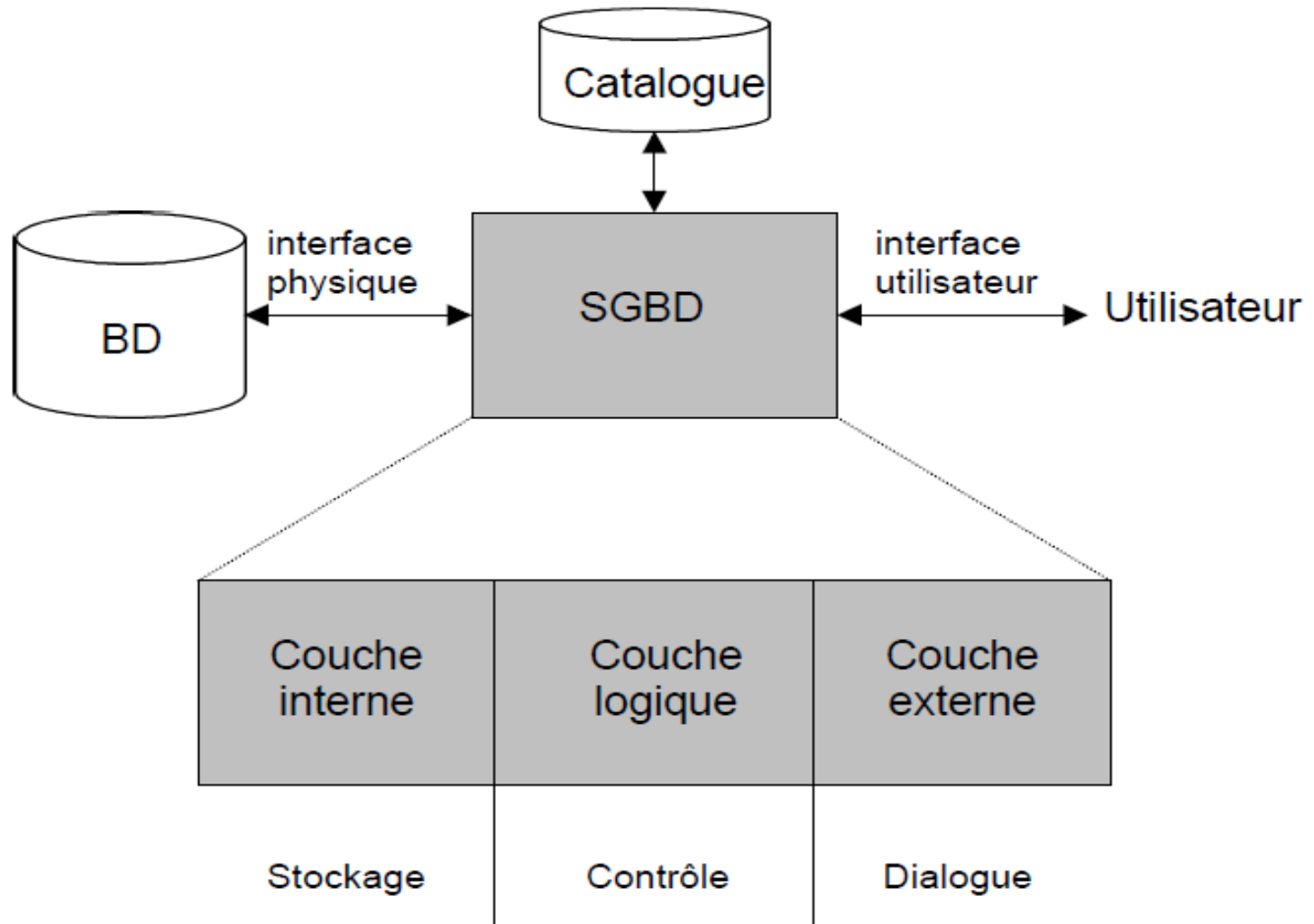
- **Protection contre les incidents**

- Événements extérieurs arrêtant l'exécution d'une transaction
- Reprise après pannes (journal)

- **Indépendance Données/Programme**

- Organisation physique des données transparente aux programmes
- Changement de l'organisation physique d'une BD sans impact sur les programmes

# 4. Architecture



# 4. Architecture

- **Modèle en trois couches**

- **Trois couches :**

- Couche interne : stockage
    - Couche logique : contrôle
    - Couche externe : dialogue

- **Catalogue du SGBD :**

- ***Base de données*** contenant les informations d'une BD
    - Description des structures présentes dans la BD, des contraintes d'intégrités, des autorisations d'accès, ... etc.

# 4. Architecture

## Modèle en trois couches : Fonctions

- **Couche interne : Stockage**
  - Stockage des données sur les supports physiques
  - Gestion de l'espace disque
  - Gestion des accès physiques (index, chemins d'accès ...)
  - Gestion sécurisée de fonctionnement (reprise après panne)

# 4. Architecture

## Modèle en trois couches : Fonctions

- **Couche logique : Contrôle**
  - Optimisation des requêtes
  - Gestion des conflits d'accès simultanés des utilisateurs
  - Contrôle général de la cohérence

# 4. Architecture

## Modèle en trois couches : Fonctions

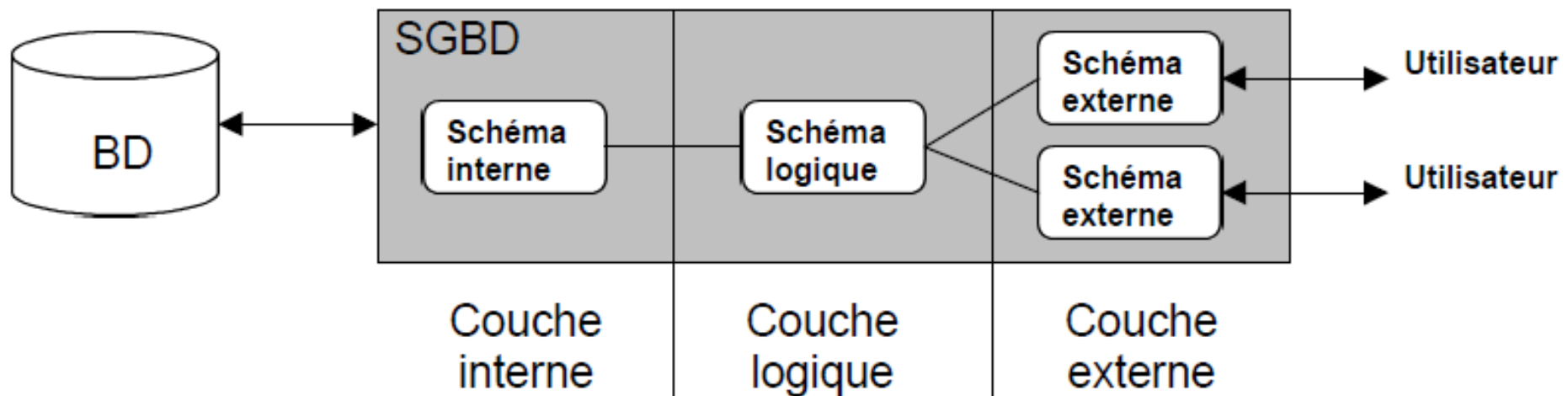
- **Couche externe : Dialogue**
  - Analyse des demandes de l'utilisateur
  - Contrôle des droits d'accès de l'utilisateur
  - Présentation des résultats



# 4. Architecture

## Modèle en trois couches : Représentations

Schémas basés sur des modèles



# 4. Architecture

## Modèle en trois couches : Représentations

- **Couche interne :**
  - Modèle interne : concepts de fichiers, organisation, index, chemin d'accès ...
  - Schéma interne : description technique de l'organisation des données
- **Couche logique :**
  - Modèle logique : modèle de données défini par le SGBD
  - Schéma logique : traduction du schéma conceptuel en un schéma conforme au modèle logique du SGBD
- **Couche externe :**
  - Modèle externe = Modèle logique
  - Schémas externes : vues utilisateur du schéma logique

# 4. Architecture

## Modèle en trois couches : Mécanismes

- **Au niveau externe :**
  - Réception des instructions
  - Analyse syntaxique (conformité à la grammaire du langage)  
et sémantique (selon la construction du schéma externe)
  - Optimisation et traduction des instructions
  - Si validation, traduction au niveau logique

# 4. Architecture

## Modèle en trois couches : Mécanismes

- **Au niveau logique :**
  - Contrôle de confidentialité (droits d'accès) et concurrence (accès concurrents aux données)
  - Si la requête est acceptée, optimisation et découpage en sous-requêtes élémentaires transférées au niveau interne

# 4. Architecture

## Modèle en trois couches : Mécanismes

- **Au niveau interne :**

- Traduction des sous-requêtes en requêtes physiques correspondantes
- Accès physiques aux données

**Exemple :** requête d'interrogation:

- Données extraites passées à la couche logique puis externe, réorganisées et présentées selon le schéma externe de l'utilisateur.