

Module 1

GÉNÉRALITÉS SUR LES BASES DE DONNÉES

Agenda

- Qu'est-ce qu'une base de données?
- Notion de modèle
- Qu'est-ce qu'un SGBD ?
- Description de données
- Architecture fonctionnelle du SGBD

Qu'est-ce qu'une base de données

- C'est un ensemble **cohérent et structuré** d'informations concernant un domaine particulier, **interrogeables par requêtes**, mémorisées sur **un support informatique**

- Cohérent et structuré
 - Les données sont caractérisées par des propriétés et des relations (i.e. des liens) entre elles.
 - Ces données concernent un domaine particulier
 - Les données modélisent une partie du « monde réel »
 - On retrouve dans le modèle de données, les concepts du domaine
- Question : Quels concepts pour la base de données d'une école?

Qu'est-ce qu'une base de données

- Interrogeable par requêtes
 - On doit pouvoir retrouver les données qui satisfont certains critères
 - On doit par ailleurs pouvoir retrouver la structure des données

- Support informatique
 - **Représentation des données** et des **relations** entre les données sous une forme « **numérique** », **interprétable** et **manipulable** par un ordinateur.
 - La Base de Données est une partie de la science informatique
 - Les **données** sont **stockées** sur les **mémoires secondaires** (Disque dur, CDRom, etc.) et on y **accède** grâce à des logiciels.

- Construire une base de données sur un domaine c'est avant tout **construire un modèle**
 - Modèle d'un domaine = représentation abstraite de ce domaine
 - Abstraction signifie éloignement vis-à-vis des considérations matérielles et logicielles

- Pourquoi modéliser ?
 - Pour construire une **représentation du domaine** qui soit **manipulable** et **interprétable** par un ordinateur
 - La **modélisation** est une **simplification de la réalité**
 - Une **modélisation** relève toujours **d'un point de vue** ou d'un **cadrage**.
 - On ne garde que les éléments **pertinents** par rapport à **ce point de vue**
 - Aussi complète soit-elle, une **modélisation** est **nécessairement « limitée »**.

- Représenter implique de :
 - **choisir** les objets, abstraits ou concrets, que l'on va décrire
 - **nommer** ces objets pour « pouvoir en parler »
 - –Un **modèle** est avant tout un **outil de communication**
 - **définir** clairement
 - Pourquoi on choisit de décrire ces « objets »
 - Les **propriétés importantes** de ces objets
 - Un modèle doit pouvoir se justifier par rapport à des choix effectués par le concepteur, le client, etc.

- Un modèle est :
 - une **interprétation explicite** par son **utilisateur** de la **compréhension** d'une **situation** ou plus exactement de l'idée que cet utilisateur se fait de la situation.
- – Pour simplifier un modèle en informatique est :
 - une **description d'entités** et de **relations** entre **ces entités**
 - synonyme : représentation
- Modéliser, en quoi cela consiste ?
 - C'est d'une certaine manière élaborer une vue partielle, plus ou moins abstraite de l'existant
 - **Principe de pertinence** : supprimer des détails et ne garder que ce qui est important par rapport à l'objectif de la BDD
 - On fait un modèle dans un but précis, et on ne garde dans ce modèle que ce qui est **pertinent** par rapport à ce but.
 - Un modèle est correct s'il permet de répondre aux questions qu'on se pose.

- **La modélisation en informatique :**
 - c'est le passage du domaine du problème à celui de sa solution informatique.
- **La difficulté de modéliser**
 - Il est difficile d'appréhender la sémantique du monde réel et de la transformer en signes manipulables par des ordinateurs

Qu'est-ce qu'un SGBD ?

- Un **SGBD** est un **ensemble de logiciels** qui permet de **créer**, de **gérer** et d'**interroger efficacement** une base de données indépendamment du domaine d'application.
- Exemples :
 - Access de Microsoft™
 - Oracle ? <http://www.oracle.com/>
 - MySQL ? <http://www.mysql.com/>
 - PostgreSQL ? <http://www.postgresql.org/>

- Le SGBD se compose de 3 couches :
 - **Le SGF : Système de Gestion de Fichier**
 - Gestion des données sur les mémoire secondaires (disque dur, bande magnétique, etc.)
 - **Le SGBD Interne**
 - Il comprend le **système d'accès aux données**. Il est quasiment invisible pour l'utilisateur, depuis les programmes d'application.
 - Il gère les données dans les fichiers
 - Il s'occupe du placement et de l'assemblage des données
 - Il gère les **liens** entre les **données** et la **structure de la base**
 - **Le SGBD externe assure**
 - – l'**analyse** et l'**interprétation** des **requêtes** des utilisateurs.
 - – la **mise en forme** des données **résultats**

- **Les principaux objectifs des SGBD sont :**
 - **Indépendance physique** des données
 - **Indépendance** logique
 - **Manipulation** des données par des **non-informaticiens**
 - **Efficacité des accès** aux données
 - **Administration cohérente** des données
 - **Non redondance** des données
 - **Cohérence** des données
 - **Partageabilité** des données
 - **Sécurité** de données
 - Il faut noter que les SGBD, même récents, n'atteignent que partiellement l'ensemble de ses objectifs

- – Séparation entre le monde réel et le monde informatique
 - La **représentation physique** des informations au niveau des mémoires secondaires informatiques doit être **indépendante** de la **représentation des informations dans le monde réel**
 - Dans le monde réel, les informations sont organisées d'une certaine manière.
 - Pour des raisons d'efficacité d'accès et de manipulation, les informations ne sont pas organisés sur les mémoires secondaires de la même manière qu'elles sont organisées dans le monde réel.

- **Toutes les informations n'ont pas la même importance selon l'utilisateur**
- Tous les utilisateurs n'ont pas les mêmes droits sur les informations
- **Notion de vue**
 - Chaque type d'utilisateur à une certaine « **vue** » des données de la base
 - Il est important qu'il y ait une **indépendance** entre cette « **vue** » propre à un utilisateur ou un groupe d'utilisateur et les données elle-même.
 - C'est ce qu'on nomme **l'indépendance logique**.
- **Intérêt**
 - Chaque groupe d'utilisateur de la base a **une vue** sur les données **correspondant à ces besoins**
 - On peut faire évoluer les vues existantes ou créer de nouvelles vues sans modifier la structure des données dans la base et vice versa.

- Critère de performance dans l'accès et le stockage des données
 - **Point de vue « utilisateur » du SGBD**
 - Les bases de données doivent être « efficaces »
 - On utilise un SGBD pour gagner du temps. Si une recherche manuelle est plus rapide, le SGBD perd de son intérêt !
 - Il faut notamment des accès aux mémoires secondaires efficaces
 - **Point de vue « programmeur »**
 - un SGBD doit pouvoir s'utiliser facilement avec un langage de programmation «classique»
 - Dans de nombreuses applications on peut avoir besoin d'accéder par programme à une base de données
 - par exemple, les langages PHP ou ASP couplés avec une base de données dans la création d'un site Web dynamique.

- Manipulation des données et de la base par des non-informaticiens
 - Les utilisateurs des **BDD** ne sont pas nécessairement des informaticiens
 - Il faut donc que le **SGBD** permette aux usagers d'**accéder**, de **manipuler**, de **modifier**, de **mettre à jour** les informations de la base facilement, sans une connaissance approfondie de la manière dont la base est « implémentée » sur la machine
 - Utilisation de **langage de requête** « simple », non-procéduraux comme **QBE** ou **SQL**.
 - Utilisation de requêtes effectuées « graphiquement »

- Deux fonctions essentielles liés à un SGBD
 - **Définition** des **structures de stockage** et des **structures de données**
 - Suivi de l'évolutions des données
- On parle **d'administration des données**

- **La non-redondance est liée à une administration cohérente des données**
 - Il faut éviter la **duplication « physique »** des données
 - C'est-à-dire à la répétition en mémoire des mêmes informations
 - La redondance d'information entraîne
 - De la **perte** de mémoire secondaire
 - La nécessité de faire des **mise à jour multiples**
 - Des incohérences dans la base de données
 - – La **redondance d'information** peut entraîner
 - Des incohérences dans la base et donc des **incohérences dans les réponses aux requêtes**
 - Des **problèmes d'optimisation** dans l'accès aux informations

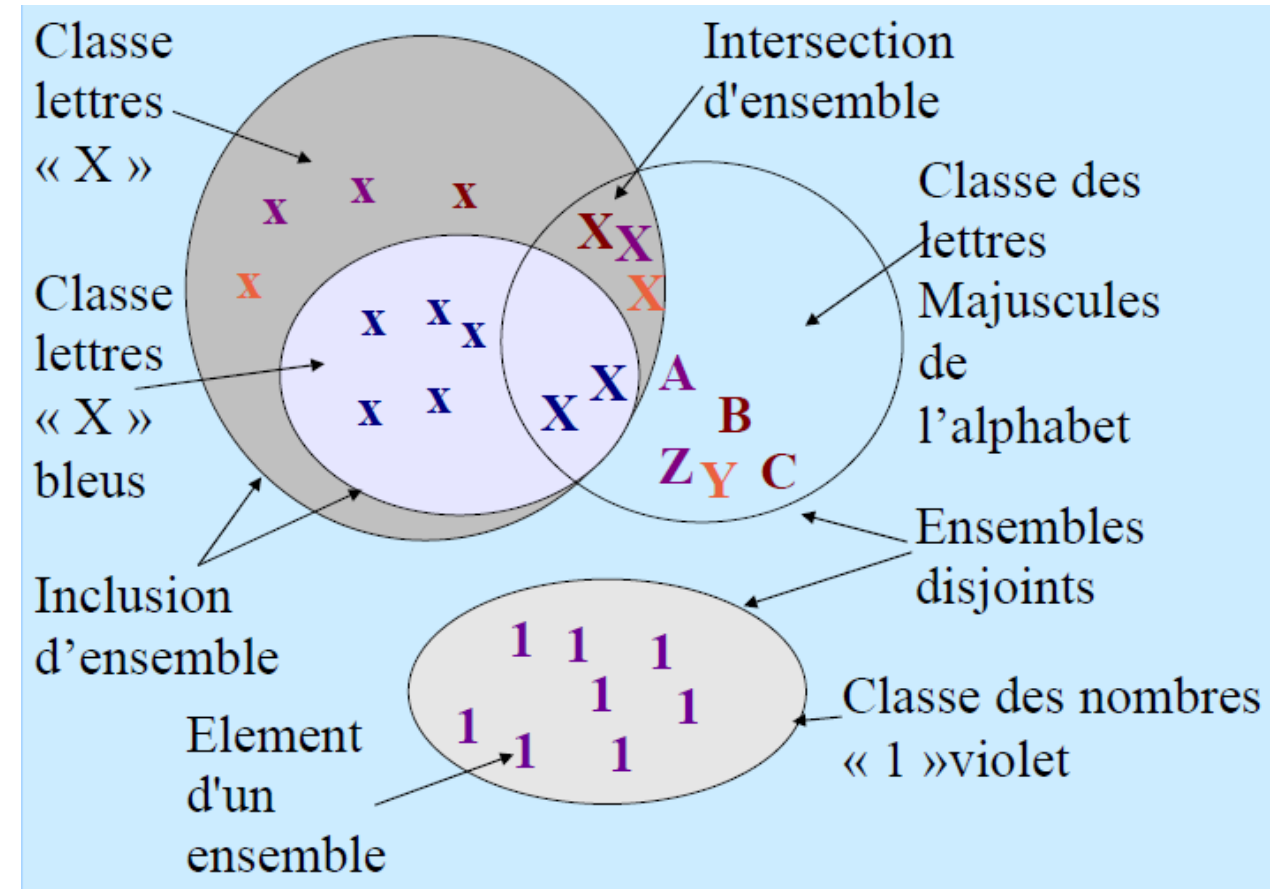
- Respects des contraintes de définition des données
 - La vérification de la **non-redondance** des informations (transparent précédent) permet d'assurer une **cohérence « structurelle »** de la base
 - Les informations ne sont **pas dupliquées** et la structure de la base respecte certaines règles facilitant son administration.
 - Il s'agit ici de vérifier la **cohérence « sémantique »** des informations, c'est-à-dire la cohérence de leur valeur.
 - Chaque donnée stockée dans la base est définie par un certain nombre de **propriétés** elles-mêmes **associées à des valeurs**.
 - Ces valeurs ont un « **domaine de définition** », c'est-à-dire qu'elle respecte certaines contraintes
 - Un âge ne peut pas être un entier négatif

- « Partageabilité » des données
 - Partage des données **simultanément/concurremment**
 - Problématique de la **cohérence de la base**, mise à jour en même temps, etc.
 - Partage des données à des moments différents
 - Optimisation de la place mémoire utilisée
 - Problématique de cohérence des données, de non-duplication.
 - Il est important de savoir que **plusieurs applications** différentes peuvent **modifier les mêmes données**.

- Il est important de pouvoir protéger les données
 - Certaines données peuvent être « sensibles » et doivent n'être **accessibles qu'à une certaine catégorie d'utilisateurs**.
 - Il faut se prémunir contre les **accès non autorisés** et/ou **mal intentionnés**
 - C'est aussi un moyen de **limiter les modifications accidentelles** des données par des utilisateurs non avertis
- – Il est donc nécessaire d'avoir une politique de sécurité
 - Problématique similaire à celle des systèmes d'exploitation
 - Présence de mécanismes pour **autoriser, contrôler** ou **enlever les droits d'accès** de n'importe quel usager (en général gérer par le(s) **DBA**)

- Toute description de données consiste à définir les caractéristiques **d'ensembles d'objets** modélisés dans la base de données.
 - On modélise des **ensembles d'objets** - des « **classes** » d'objets - et non juste des objets particuliers.
 - Exemple dans une école, on modélise la classe d'objet « Etudiants » et non chaque étudiant en particulier
- Les **objets particuliers** sont définis par les programmes d'applications lorsque les utilisateurs insèrent ou mettent à jour des données.
 - Les **objets particuliers** doivent alors **vérifier les propriétés** de l'ensemble auquel ils appartiennent.

- Intuitivement un **ensemble** est une **collection d'objets**
 - On désigne souvent ces objets comme les **éléments** de l'ensemble
 - En général, un ensemble est formé d'**éléments** qui possèdent certaines **propriétés communes**.
 - C'est notamment le cas des ensembles que nous définirons dans le cadre des Bases de Données
- En informatique pour désigner un **ensemble** on utilise également souvent le terme de « **classe** ».



- **Notion de type d'objet**

- – **Ensemble d'objets** (ou **classe d'objets**) possédant des **caractéristiques similaires** et **manipulables** par des **opérations identiques**
 - – **exemple 1** : le type d'objet Entier
 - représente les entiers relatifs $\{0, +/-1, +/-2, \dots, +/-N, \dots +/-\infty\}$
 - est associé aux opérations $\{+, -, /, *, \}$.
 - – **exemple 2** : le type d'objet Livre
 - représente un livre dans la base d'un centre de documentation.
 - est caractérisé par son titre, son ou ses auteurs, un numéro identifiant, etc.
 - est associé des opérations de création, modification, destruction, consultation, duplication.
 - Ces opérations sont des opérations standards sur de nombreux objets

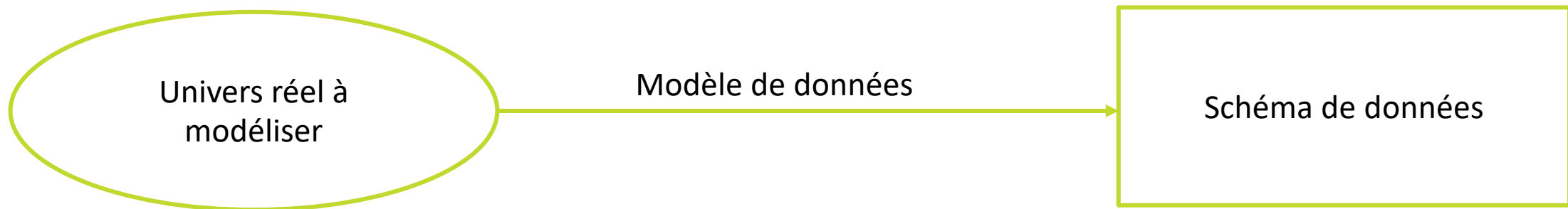
- Notion d'instance d'objet
 - Une **instance** d'objet est un **élément particulier** d'un **ensemble** d'objets.
 - Le terme “**occurrence** d'objet” est aussi parfois utilisé.
 - Par abus de langage on parle souvent directement d'« **objet** ».
 - Une **instance** est **caractérisée** par un **identifiant**, des **propriétés** et des **valeurs** pour ces propriétés
- **Exemple 1** : l'entier 10 est une **instance** du **type d'objet Entier**
- **Exemple 2** : le livre « Le nom de la rose » de Umberto Eco est une **instance** du **type d'objet Livre**.
 - caractérisé par les **propriétés** “titre”, “auteur”, etc.
 - pour cette **instance particulière** de livre, les **valeurs** pour les **propriétés** “titre” et “auteur” sont “Le nom de la rose” et “Umberto Eco”.

- Modèle de description des données (Data Model)
 - **Décrire des données** se ramène à **définir un type d'objet**
 - Des éléments descriptifs permettent de décrire les propriétés d'ensembles d'objets :
 - Ces **éléments descriptifs** sont nommés **caractéristiques, propriétés** ou **attributs**
 - A partir de ces **éléments descriptifs** on compose un **modèle de description** de données qui peut se définir comme :
 - Un ensemble de concepts, et de règles de composition de ces concepts, permettant de décrire des données.

- **Langage de description de données (LDD)**

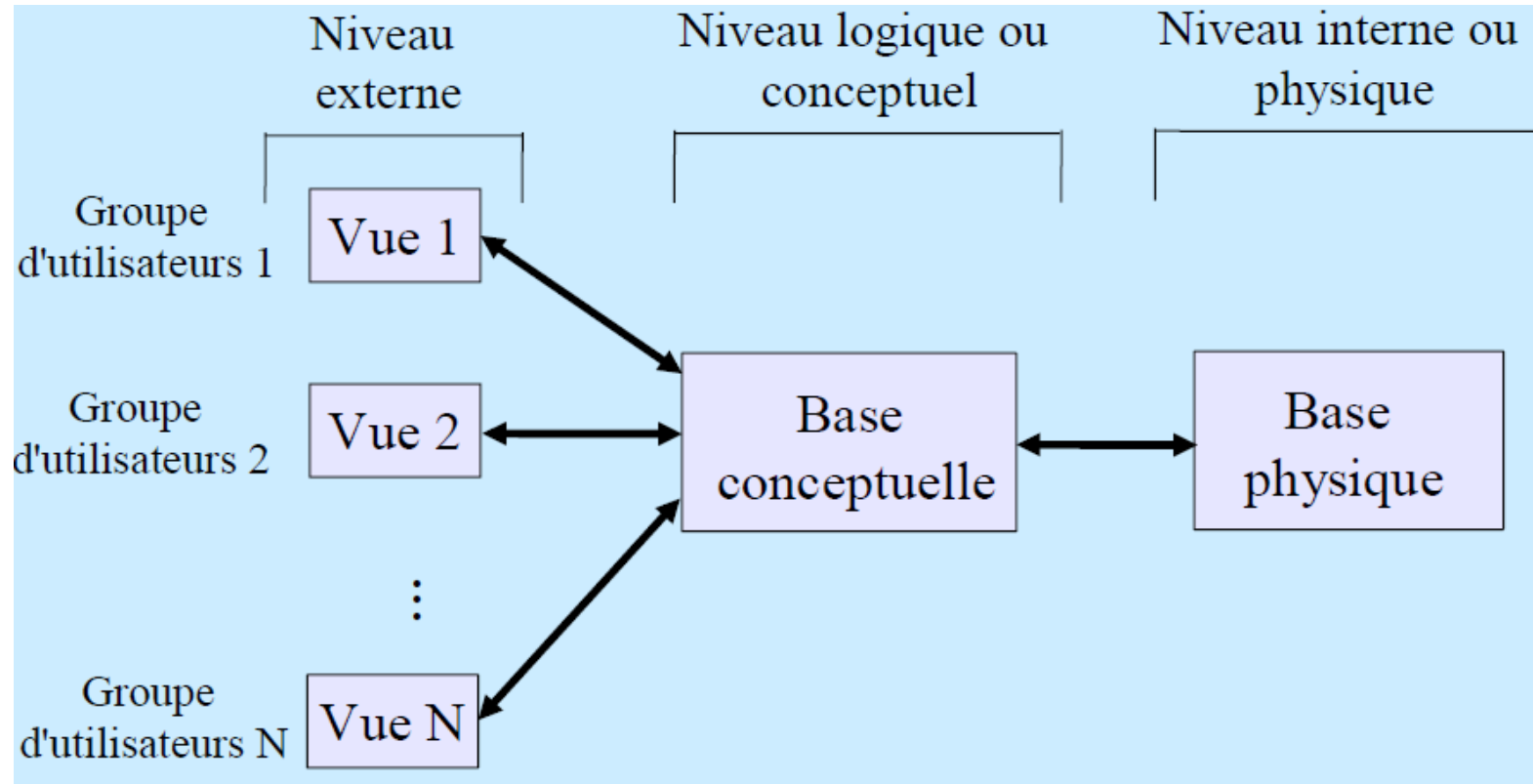
- On parle aussi de langage de définition de données
- Le **modèle de description** des données doit être **exprimée** par un **langage**
- Le modèle de description de données est souvent représenté de manière graphique
 - – Le **modèle entité – association** par exemple fournit un tel formalisme graphique permettant de décrire des modèles de données.
- Un **langage de description de données** est donc :
 - Un langage supportant un modèle et permettant de décrire les données d'une base de données d'une manière assimilable par une machine.

- Schéma de données
 - La description d'un ensemble de données particulier, correspondant par exemple à une application, à l'aide d'un **langage de description** donne naissance à un **schéma**.
 - Un schéma est une description au moyen d'un langage déterminé d'un ensemble de données particulier.
 - C'est donc une **abstraction** résultant de l'application d'un **modèle de données** à l'univers réel.



- Afin d'assurer les objectifs d'**indépendance logique** et **physique**, on distingue **3 niveaux de description de données** pour l'**architecture fonctionnelle** d'un **SGBD** :
 - Niveau externe
 - **Niveau logique** ou niveau **conceptuel**
 - **Niveau interne** ou niveau **physique**
- ces 3 niveaux de description de données sont définis par la norme AINSI/SPARC.

Architecture fonctionnelle d'un SGBD



- La **base conceptuelle** est une **abstraction** de la **base physique**
- Les **vues** définies au niveau externe sont des **abstractions** de la **base conceptuelle**

- **Niveau conceptuel**

- Il correspond à la **structure canonique des données**
- Il décrit :
 - les données qui sont actuellement stockées dans la base
 - les liens entre ces données,
 - **ne se soucie pas** de la manière dont cela doit être implantée sur la machine
- – Il permet de définir :
 - Les **types de données élémentaires** qui définissent les **attributs** des objets que l'on souhaite modéliser.
 - Par exemple : nom, prénom, matricule étudiant, date de naissance, etc.
 - Les **types de données composées** qui permettent de décrire les entités et les associations du monde réel
 - – Par exemple : Etudiants, Enseignants, Classes, Utilisateurs, etc.
 - Les éventuelles **règles** et **contraintes d'intégrité**.

- **Niveau physique**
 - Il correspond à la **structure de stockage des données**
 - Il permet de décrire :
 - Les **fichiers** avec leur nom, leur organisation, leur localisation, etc.
 - Les **articles** de ces fichiers (longueur, champs, composants, etc.)
 - Les articles : ce qui est contenu dans ces fichiers
 - Les **chemins d'accès aux données** (index, chaînage, etc.)
 - **Attention** : on est au niveau “**implémentation**”, on se pose des questions d'ordre “technique

- **Administrateur de données**

- La ou les personnes **responsables** de la **définition de schémas** de bases de données.
- Elle regroupe **3 rôles distincts** :
 - **Administrateur de bases de données** : rôle de **définition** du **schéma interne** (i.e. de la base physique) et des **règles de correspondance** entre les **schémas interne** et **conceptuel**.
 - Administrateur d'entreprise : rôle de définition du schéma conceptuel.
 - – Administrateur d'application : rôle de **définition** des **schémas externes** et des **règles de correspondances** entre les **schémas externe** et **conceptuel**.

- **Dictionnaire des données**
 - Ensemble des **schémas** et des **règles** de passage entre les **schémas associés à une base de données**, combinés à une **description de la signification des données**
- **Métabase**
 - **Dictionnaire de données** organisé sous forme de **base de données** qui décrit les autres bases.

- **Un seul schéma interne, un seul schéma conceptuel, plusieurs schémas externes**
 - Pour une BDD particulière, il existe **un seul schéma interne** et **un seul schéma conceptuel**, mais par contre il peut exister **plusieurs schémas externes**.
- **– Schéma conceptuel**
 - **Description** des données d'une entreprise ou d'un domaine en terme de **types d'objets** et de **liens logiques**, **indépendamment de toute représentation en machine**.
 - Cette description correspond à **une vue canonique** globale de l'entreprise ou du domaine modélisé.
 - **canonique** signifie qui se conforme à un **ensemble de règles**.
 - – **vue canonique** signifie donc **une vue qui se conforme à un ensemble de règles propres à l'entreprise ou au domaine**

- Schéma interne
 - **Description des données** d'une base en termes de **représentation physique en machine**.
 - Cette description correspond à une **spécification des structures de mémorisation** et des **méthodes de stockage et d'accès** utilisées pour ranger et retrouver les données sur disques
- Schéma externe
 - **Description d'une partie de la base** de données,
 - Les vues sont **extraites** ou **calculées** à partir de la base physique,
 - Elles correspondent à la vision d'un programme ou d'un utilisateur
 - Un **schéma externe** correspond donc à un **arrangement particulier** de **certaines données**.



SEJEN

POWERING DECISION MAKING