

L'approche Bases de données

Notions générales

Définition

Une **BD** un ensemble d'informations structurées **non redondant** permettant la **mise en place d'une série d'applications** informatiques **destinées** à une **grande variété d'utilisateurs**.

• Exemple:

Dans une entreprise, les informations concernant son fonctionnement :

- Employés
- Produits Fabriqués
- Moyens matériels (Machines, Véhicules, Magasins, etc.)

peuvent être rassemblées sous forme de BD et mises à la disposition de nombreux utilisateurs (cadres de l'entreprise, gestionnaires, opérateurs, clients, fournisseurs, etc.).

Objectifs d'une base de données

Parmi les principaux objectifs visés par une base de données :

➤ Permettre le Partage de l'information

Une **BD** permet le **partage** d'un **ensemble unique d'informations** par **plusieurs utilisateurs**.



il faut que cette mise en commun soit faite :

1. tout en préservant **la vue particulière** que chaque utilisateur peut avoir des informations,
2. et en s'assurant que la **simultanéité des traitements** qui peuvent être effectués ne risque pas **d'altérer l'intégrité** de la base de données.

➤ Organiser les données indépendamment des programmes :

☒ Afin de construire un ensemble d'informations structurées non redondant et qui soit partageable par plusieurs utilisateurs :

- il est nécessaire de **faire abstraction des traitements** particuliers de tel ou tel utilisateur (ou programme)
- **organiser les informations en fonction** de leur **nature** et des **liens réels** qui existent entre elles **et non en fonction des traitements**.

➔ C'est de cette manière qu'on arrivera à garantir le maximum d'indépendance entre données et programmes.

Avantages de l'approche B.D.

✓ Contrairement aux approches classiques, l'approche BD est le reflet d'une évolution dans la gestion de l'entreprise.

✓ Elle rend possible :

La centralisation de l'information :

l'information n'est plus éparpillée dans différents fichiers à différents endroits

L'intégration :

tout ce qui se fait dans un service est visible par d'autres services

La diffusion de l'information archivée :

si l'information est disponible à un seul endroit, elle est facile à diffuser

Ceci a pour avantages :

1- Améliorer la cohérence de l'information :

une seule valeur pour une même information

2- Réduire les redondances :

une même information n'est stockée si possible qu'une seule fois

3- Réduire les efforts de saisie et de mise à jour :

conséquence de 2. une information qui doit être stockée une seule fois ne sera saisie qu'une seule fois. De même que sa mise à jour ne se fera qu'une seule fois

Démarche de conception d'une BD

- la conception d'une base de données doit se faire en utilisant une **méthode de conception** qui définit la démarche à suivre.
- Plusieurs méthodes de conception existent

Exemple :

la méthode **MERISE** basée sur le modèle **E/R** de Chen

✍ Pour certaines méthodes, on dispose même d'un outil logiciel **d'aide à la conception** appelé aussi un **Atelier de Génie Logiciel (AGL)**

❖ un ensemble de logiciels permettant l'automatisation d'un certain nombre de tâches lors des différentes phases du processus de conception :

- génération automatique de la structure de la B.D.,
- de programmes d'accès et de manipulation,
- etc..

☞ Quelle que soit la méthode utilisée, la conception d'une base de données passe par **un processus de modélisation** :

- Modéliser une certaine **partie du monde réel** afin de caractériser les **entités qu'on manipule** (étudiants, Comptes Bancaires, Ouvrages, etc.)
- Caractériser les **attributs de ces entités** en fonctions des problèmes que doit résoudre l'existence de la B.D. :
 - Gestion de la scolarité,
 - Gestion de Prêt d'ouvrages dans une bibliothèque,
 - etc.

Exemple :

- Pour un étudiant on aura besoin d'un **attribut relatif au résultat** dans un module dans le cas d'une **Gestion de la scolarité**, alors que ce type d'attribut **ne sera pas nécessaire ?** dans le cas d'une application de **Gestion de Prêt**

➤ Le cas le plus général est celui où la B.D. est partagée par plusieurs utilisateurs.

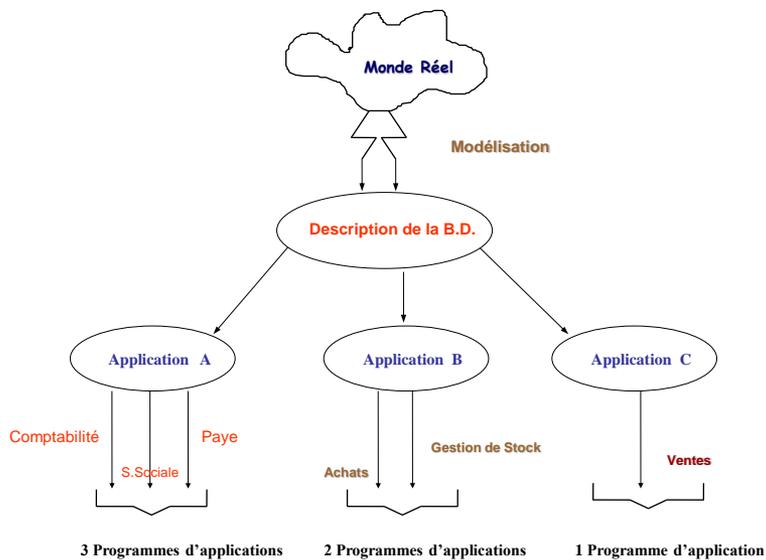
- Ces utilisateurs n'ont pas tous la même vue des données de la base,
- Ils n'ont pas tous à voir la base dans sa totalité
- Chaque utilisateur n'est concerné que par une partie de celle-ci.

Exemple :

Dans une entreprise l'ensemble des informations sur les :

- départements,
 - les employés,
 - les produits,
 - le matériel,
 - etc.
- peuvent être rassemblées sous forme d'une B.D. et il est bien rare qu'un utilisateur de cette base ait besoin de toutes ces informations à la fois.

Processus de Modélisation d'une B.D.



Remarque

- ☝ Il serait plus juste de remplacer le terme **Utilisateur** par celui d'**Application**
- ➔ On définit en général une B.D. afin de mettre en place une série d'applications ayant chacune ses programmes et ses propres utilisateurs exploitant le même sous-ensemble de données.

Exemple :

Dans une B.D. de Gestion Universitaire, on peut mettre en place une application de gestion de Prêts qui sera **composée d'un ensemble de programmes** et dont les **utilisateurs seront ceux** concernés par les traitements réalisés par ces programmes (Secrétaire, Documentaliste, Archiviste, etc.).

Problèmes posés par la Centralisation de l'information sous forme de B.D.

- la **centralisation** de l'information sous forme **d'une B.D. unique pose un certain nombre de problèmes** liés directement à l'**intégrité** et à la **sécurité** de ces informations.
- **Parmi ces problèmes**, on peut énumérer les suivants :

❖ Nature et type de l'information

➤ Lors de la mise en place d'une B.D., **l'utilisateur décrit les propriétés que doivent vérifier ses données et qui doivent être préservées tout au long de l'existence de la base de données.**

❑ **Ces propriétés peuvent se situer à différents niveaux :**

- **Appartenance d'une donnée à un ensemble de valeurs**

Ex : l'âge d'un employé est un entier positif compris entre 0 et 150
 $0 < \text{âge} < 150$

- **Déclaration de propriétés invariantes au cours du temps**

Ex : Un enseignant à une heure Donnée ne peut se trouver que dans une seule salle

- **Relation d'ordre total à respecter lors du stockage des données**

Ex : Les employés doivent être stockés dans la B.D. par ordre croissant de leur numéro ou par ordre alphabétique sur leur nom

❖ Sûreté physique, sûreté de fonctionnement et point de reprise

➤ Il s'agit de protéger l'information contre **un mauvais fonctionnement** soit de la **machine**, soit du **système** (logiciel) qui gère la base de données.

❑ Dans le **premier cas**, on peut **délimiter les enregistrements** qui ont été altérés ou perturbés

❑ Dans le **second cas** le problème est beaucoup **plus complexe**.

- Une des solutions en usage consiste à **prendre à intervalles réguliers des copies de la B.D.** et à enregistrer l'ensemble des transactions (opérations) effectuées sur la base.

- Ceci permettra en cas d'incident de **régénérer une copie consistante** (i.e. sans défauts) de la base.

❖ Partage de l'information

- Lorsque deux programmes P1 et P2 veulent se partager la même donnée A, il peut y avoir **perte d'intégrité**.

Exemple :

P1 accède à	A	et la transfère dans son buffer propre
P2 accède à	A	et la transfère dans son buffer propre
P1 modifie	A	dans son buffer puis la recopie dans la B.D.
P2 modifie	A	dans son buffer puis la recopie dans la B.D.

venant ainsi écraser les modifications faites par P1.

- La solution à ce problème serait par exemple celle de **l'exclusion mutuelle** qui est une technique utilisée dans les systèmes d'exploitation.

❖ Problème des données confidentielles

- Il s'agit de **protéger les données** contre des **utilisateurs indiscrets**
- On dispose en général pour cela de **procédures sélectionnant les accès à la base de données**.
- Lorsqu'un utilisateur veut faire un accès, on distingue **deux phases** :

1. La phase d'identification :

a pour but **d'identifier l'utilisateur qui veut se connecter** à la base de données. Ceci est possible grâce à un mot de passe, une carte spéciale, etc.

2. La phase d'autorisation :

après identification de l'utilisateur, **permet de déterminer ce que peut faire cet utilisateur sur telle ou telle données** (consulter seulement, consulter et mettre à jour, etc.)

Niveaux de description d'une Base de Données

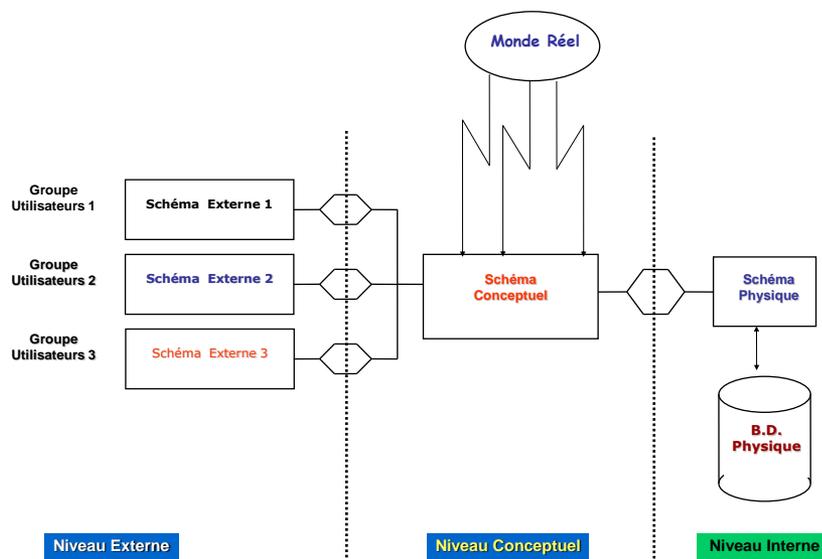
- La description d'une base de données peut se faire à **différents niveaux**, suivant que l'on regarde plus du côté de l'utilisateur que du côté du stockage des données sur les supports physiques.
- On distingue communément trois (3) niveaux de description d'une B.D.

1- niveau conceptuel

2- niveau externe

3- niveau interne

Niveaux de description d'une Base de Données



Le Niveau conceptuel

- Le **schéma conceptuel** est la **partie fondamentale** dans l'architecture d'une base de données.
- Il a pour but de **décrire en termes abstraits mais fidèles** une certaine réalité d'une organisation et de ses processus de gestion qui ont nécessité la mise en place d'une B.D.
- Le passage du **monde réel au schéma conceptuel** correspond à un **processus de modélisation** :
 - les objets du monde réel ayant les mêmes caractéristiques sont classés en catégories et désignés par des noms (Etudiants, Véhicules, etc..)
- Dans le processus de modélisation, le concepteur de la B.D. spécifie le schéma conceptuel en utilisant les possibilités offertes par un **modèle de données**.

Définition d'un modèle de données

- Un modèle de données est un **outil formel** destiné à **décrire la réalité** de manière **indépendante** de **tout traitement informatique**.
- Un modèle de données **doit permettre de regrouper** les objets du monde réel auxquels on s'intéresse en classes d'objets de nature identique.

Exemple :

- ▶ Dans une application de gestion universitaire, on pourra regrouper les étudiants dans une classe d'objets qu'on appellera ETUDIANTS et les modules dans une autre classe d'objets qu'on appellera MODULES.
- ▶ Par la suite on ne fera référence aux objets de ces classes que par l'intermédiaire de ces noms.

- Un modèle de données doit aussi permettre de décrire les **liaisons** ou **associations** qui peuvent exister entre les classes d'objets.

Exemple :

- Dans une application de gestion universitaire, l'inscription est un phénomène qui associe un objet étudiant appartenant à la classe ETUDIANTS à un ou plusieurs objets modules appartenant à la classe MODULES.
 - On pourra dans ce cas créer une association qu'on nommera **INSCRIPTION** entre la classe ETUDIANTS et la classe MODULES afin de modéliser cette réalité.
- Un schéma conceptuel est donc le **résultat d'un processus de modélisation** fait **en respectant** les possibilités d'un **modèle de données**.

Classification des modèles

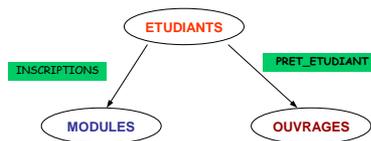
- Il existe **trois grandes classes de modèles** de données qui se **distinguent par** la nature des associations qu'ils permettent de modéliser.

Ce sont :

- ⊗ Les modèles **hiérarchiques**
- ⊗ Les modèles **réseaux**
- ⊗ Les modèles **relationnels**

Le modèle hiérarchique

- A l'aide du modèle hiérarchique, le schéma conceptuel peut être vu comme un **graphe arborescent** dont **les nœuds** correspondent aux classes d'objets (**entités**) et les **arcs** entre deux nœuds aux **liaisons** ou associations entre les entités.
- Un tel graphe possède donc un **nœud racine** (sur lequel n'arrive aucun arc !) et les autres nœuds sont des fils, petit-fils, etc., de cette racine.
- Avec le modèle hiérarchique, le nombre de flèches pouvant arriver sur un nœud est donc égal à un (sauf pour le nœud racine).

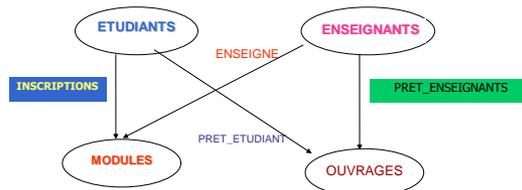


Cours: BDD. – Année: 2019/2020 Ens. S. MEDILEH (Univ. El-Oued) L'approche Base de Données

23

Le modèle réseau

- A l'aide de ce modèle, le schéma conceptuel peut être vu comme un graphe général où les nœuds correspondent aux classes d'objets et les arcs entre deux nœuds aux associations.
- A la différence du modèle hiérarchique on peut avoir ici **plusieurs arcs** qui **arrivent sur le même nœud**.
- De même que la notion de **nœud racine n'existe pas** avec le modèle réseau.



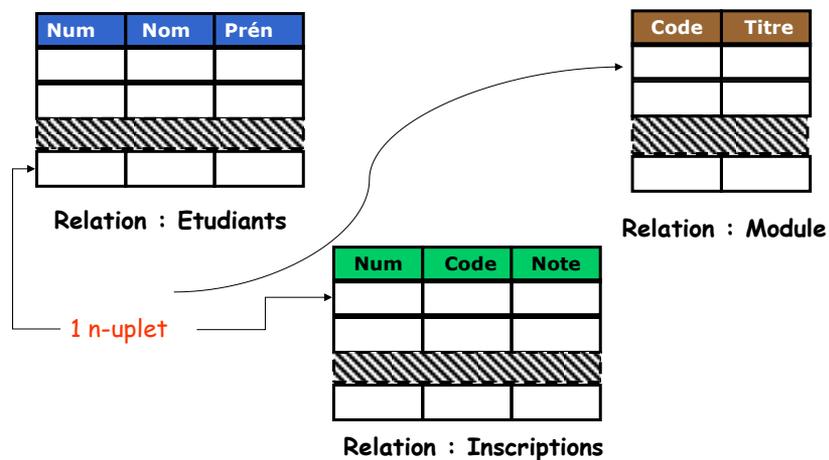
Cours: BDD. – Année: 2019/2020 Ens. S. MEDILEH (Univ. El-Oued) L'approche Base de Données

24

Le modèle relationnel

- Ce modèle est fondé sur la théorie mathématiques des relations.
- Le **schéma conceptuel** peut être vu comme **un ensemble de tables** (ou relations) à **n colonnes**, **n** désignant **le degré de la relation**.
- Avec le modèle relationnel, **une table** sert à représenter **aussi bien** une **classes d'objets** qu'une **association** entre des classes d'objets.
- ✂ Ainsi, la distinction entre nœud et arc comme dans les autres modèles n'est pas nécessaire avec le modèle relationnel.
- Chaque élément d'une table est appelé un **n-uplet**.

- Par exemple la table **INSCRIPTIONS** décrit l'association entre la classe d'objets **ETUDIANTS** et la classe **MODULES** et qui permet de modéliser le fait qu'un étudiant peut s'inscrire à 0, 1 ou plusieurs modules.



Le Niveau Externe

- Ce niveau correspond à la vision de tout ou partie du schéma conceptuel par un groupe d'utilisateurs concerné par une application.
- Il s'agit de décrire à l'aide d'un **schéma externe ou Vue** la façon dont seront perçues les données par un programme d'application.

Exemple :

Dans une base de données de gestion universitaire, un groupe d'utilisateurs concerné par les **INSCRIPTIONS** des étudiants

- n'a pas besoin d'avoir une vision globale de la base et peut se limiter à la partie qui englobe les informations relatives aux **étudiants et aux Modules**.

Le Niveau Externe

- Un schéma externe peut donc être considéré comme **un sous schéma du schéma conceptuel**.
- Grâce à cette notion de schéma externe, **chaque groupe d'utilisateurs perçoit les données à sa façon**
 - Par exemple, une donnée vue comme donnée numérique par un groupe peut être vue comme une chaîne de caractères par un autre (cas d'une date par exemple).
 - Un groupe peut ne pas voir certaines caractéristiques (attributs) d'une entité (ex : note obtenue dans un module) qui seront par contre visibles par un autre groupe (ou application).

Le Niveau Interne

- Ce niveau a pour but de spécifier **comment les données** sont **stockées** sur les **supports physiques**.
- Cette spécification est faite par le biais d'un **schéma physique** ou **schéma de stockage**.
- Ce schéma permettra par exemple de :
 - **Décrire la structure des fichiers qui constituent la base de données** (nom d'un fichier, organisation, adresse sur le support, etc.)
 - **Définir les méthodes d'implantation** (fichier plat, inversé, etc.)
 - **Préciser les chemins d'accès aux enregistrements** (index, chaînage, calcul d'adresse, etc.)

- 👉 Il faut noter que tous ces aspects ne doivent pas affecter les applications **sauf sur le plan des performances** d'accès à la base
- Afin de garantir l'un des objectifs visés par une base de données à savoir : **l'indépendance entre les données et les programmes**.
- ❖ Le souci du schéma physique est donc de **pouvoir changer l'organisation physique des données** sans modifier le schéma conceptuel ni les programmes d'application.
- 📖 **Par exemple**, pour augmenter les performances d'accès à la base de données, on peut être amené à :
 - Changer l'organisation d'un fichier (Passer par exemple d'une organisation initialement séquentielle à une organisation séquentielle indexée ou directe).
 - Déplacer physiquement le fichier vers une autre adresse sur le support
 - Modifier les chemins d'accès aux enregistrements (changer d'index, ajouter d'autres indexes, etc.).