

3.36pt

Bases de Données

Atef Hasni

AFPA - Formation diplômante

Cursus DWWM

- ▶ Introduction aux bases de données
- ▶ Modèle Conceptuel de Données (MCD)
- ▶ Modèle Logique de Données (MLD)
- ▶ Passage d'un schéma E/A à un MLD relationnel
- ▶ Interrogations des bases de données par requêtes SQL

- ▶ Redondance de l'information si les données sont fortement corrélées
- ▶ Certains calculs sont difficiles voire impossibles à effectuer
- ▶ L'information peut être difficilement partagée

Base de données : collection importante de données

- ▶ représentant une partie du monde réel
- ▶ structurées
- ▶ cohérentes
- ▶ conservées de manière permanente (persistance)
- ▶ manipulables/modifiables/interrogeables
- ▶ partageables
- ▶ sécurisées

Système de gestion de bases de données (SGBD)

- ▶ définition de données
- ▶ manipulation des données
- ▶ contrôle des données

Exemple de bases de données :

- ▶ Gestion d'entreprises : stocks, personnel, clients, ...
- ▶ Banques : comptes, emprunts, ...
- ▶ Systèmes de réservation : avions, trains, spectacles, ...
- ▶ Bibliothèques : ouvrages, emprunteurs, prêts, ...
- ▶ Universités : étudiants, salles, cours, enseignants, ...
- ▶ Etc ...

Modèle Conceptuel de Données (MCD)

- ▶ Le MCD correspond à une représentation graphique des données manipulées dans le système d'information
- ▶ Il sera utilisé pour :
 - ▶ décrire le système d'information en place (*étude de l'existant*)
 - ▶ proposer le système d'information à venir (*prototypage*)
- ▶ Il est défini par le *modèle Entité/Association (E/A)*
- ▶ Formalisme retenu par l'ISO pour décrire l'aspect conceptuel des données à l'aide d'*entités* et d'*associations*

Le concept d'entité :

- ▶ représentation d'un objet matériel ou immatériel
- ▶ peut être vue comme un ensemble de propriétés ou de caractéristiques
- ▶ les propriétés :
 - ▶ données élémentaires relatives à une entité
 - ▶ les propriétés d'une entité sont également appelées des **attributs**, ou des **caractéristiques** de cette entité
- ▶ l'identifiant : groupe minimal d'attributs qui permet d'identifier une entité de manière unique
- ▶ représentation graphique :

Nom de l'entité
Nom de propriété 1
Nom de propriété 2
Nom de propriété 3
...

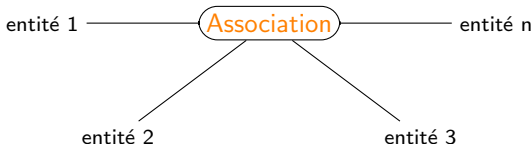
- ▶ **Exemple** : Gestion d'une bibliothèque
 - ▶ Un livre donné ne peut être emprunté que par un seul étudiant.
 - ▶ Un étudiant emprunte plusieurs livres.

Etudiant
<u>Numéro</u>
Nom
Prénom
Filière

Livre
<u>ISBN</u>
Titre
Auteur
Catégorie

Le concept d'association :

- ▶ représentation d'un lien entre deux entités ou plus (binaire, ternaire, n-aire)
- ▶ dépourvue d'existence propre (son existence est conditionnée par celle des entités qui la composent)
- ▶ une association peut avoir des propriétés particulières
- ▶ représentation graphique :

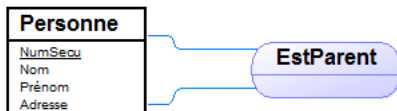


- ▶ **Exemple** : Gestion d'une bibliothèque
 - ▶ Un livre donné ne peut être emprunté que par un seul étudiant.
 - ▶ Un étudiant emprunte plusieurs livres.



Le concept d'association :

- ▶ une association **réflexive** est une association qui fait intervenir au moins deux fois la même entité
- ▶ Exemple :



Le concept de cardinalité :

- ▶ La cardinalité d'une association pour une entité constituante est constituée d'une borne minimale et d'une borne maximale
 - ▶ **Minimale** : nombre de fois minimum qu'une occurrence de l'entité participe aux occurrences de l'association, généralement 0 ou 1
 - ▶ **Maximale** : nombre maximum de fois qu'une occurrence de l'entité participe aux occurrences de l'association, généralement 1 ou n

- ▶ **Exemple** : Gestion d'une bibliothèque
 - ▶ Un livre donné ne peut être emprunté que par un seul étudiant.
 - ▶ Un étudiant emprunte plusieurs livres.



Le concept de cardinalité :

Cardinalité minimale	Signification
0	il peut exister une occurrence de l'entité qui ne participe pas à l'association
1	toute occurrence de l'entité participe obligatoirement à l'association une seule fois au moins
2	toute occurrence de l'entité participe obligatoirement à l'association deux fois au moins
.	.
.	.
.	.

Le concept de cardinalité :

Cardinalité maximale	Signification
1	une occurrence de l'entité peut participer une seule fois au plus à l'association
2	une occurrence de l'entité peut participer deux fois au plus à l'association
.	.
.	.
.	.
n	une occurrence de l'entité peut participer plusieurs fois à l'association

Le concept de cardinalité :

Min	Max	Signification
0	1	une occurrence de l'entité peut participer une seule fois au plus à l'association
1	1	toute occurrence de l'entité participe exactement une seule fois à l'association
.	.	.
.	.	.
.	.	.
0	n	une occurrence de l'entité peut participer plusieurs fois à l'association
1	n	une occurrence de l'entité participe une ou plusieurs fois à l'association

- ▶ **Contraintes d'intégrité fonctionnelle (CIF) :**
 - ▶ une CIF est une contrainte sur des entités participant à une même association
 - ▶ elle exprime que l'une des entités (entité **Père**) est totalement déterminée par une ou plusieurs autres entités (entité **Fils**)
 - ▶ elle consiste en une cardinalité égale à **0,1** ou **1,1** sur l'une des pattes
- ▶ **Exemple :** entité **Père** : Etudiant, entité **Fils** : Livre



- ▶ MCD : description des données indépendamment des choix techniques
- ▶ MLD : description des données à partir du MCD en tenant compte de l'orientation des **choix techniques** de la gestion des données
- ▶ L'orientation la plus répandue est celle de base de données de type **relationnel**
- ▶ On n'étudiera dans ce cours que la construction du **MLD relationnel** à partir du MCD

- ▶ Modèle théorique introduit par Codd en 1970, basé sur la théorie mathématique des ensembles
- ▶ Il permet de représenter toutes les données dans des **tables** (ou **relations**)
- ▶ Les données dans chaque table sont constituées par l'ensemble des **tuples** (lignes) fournissant des valeurs pour les **attributs** (colonnes)
- ▶ Pas d'ordre sur les tuples (relations = ensembles non ordonnés)

Le concept de table :

- ▶ une table contient un ensemble de relations définies par les mêmes propriétés
- ▶ chaque **colonne** ou **attribut** représente une information élémentaire prenant ses valeurs dans un domaine
- ▶ chaque domaine a un type de données
- ▶ Exemple : **Etudiant(Numéro, Nom, Prénom, Filière)**

Numéro	Nom	Prénom	Filière
20143657	Brasseur	Adrian	Energie
20132347	Macq	Jordan	Informatique
20150621	Cagin	Marine	Génie Civil
20152718	Felder	André	Mathématiques

Le concept de clé primaire :

- ▶ toute table possède une **clé primaire** : un ensemble minimal d'attributs
- ▶ la clé primaire permet d'identifier de manière **unique** une tuple de la table
- ▶ il n'y a qu'**une seule** clé primaire par table
- ▶ la clé primaire est **soulignée**
- ▶ **Exemple** : Voiture(NumImm, Couleur, NomModèle)

NumImm	Couleur	NomModèle
123XY34	Jaune	106
34UV62	Verte	106
234XU45	Bleu	Clio
123XY34	Verte	106

Le concept de clé étrangère :

- ▶ une table peut posséder des **clés étrangères** : ensemble d'attributs dont les valeurs appartiennent à l'ensemble des valeurs d'une clé primaire d'une autre table
- ▶ contrairement aux clés primaires, les clés étrangères peuvent prendre plusieurs fois la même valeur dans une table
- ▶ la clé étrangère est **soulignée**
- ▶ Il existe d'autres types de contraintes ...

► Exemple :

- Modèle(NomModèle, Marque)
- Voiture(NumImm, Couleur, NomModèle)

NomModèle	Marque
106	Peugeot
206	Peugeot
306	Peugeot
Clio	Renault

NumImm	Couleur	NomModèle
123XY34	Jaune	106
34UV62	Verte	106
345RT62	Verte	Mégane
234XU45	Bleue	Clio
123XY34	Verte	106

Passage d'un schéma E/A à un MLD relationnel

- ▶ Le modèle relationnel :
 - ▶ utilisé dans les bases de données
 - ▶ consultable à partir de langages de haut niveau
 - ▶ efficace
- ▶ Les équivalences : passage d'un modèle disposant de deux structures (entités et associations) à un modèle disposant d'une seule structure (tables)

Modèle Entité/Association	Modèle relationnel
Entité ou association	Table (relation)
Propriété	Attribut
Identifiant	Clé primaire et parfois clé étrangère

Passage d'un schéma E/A à un MLD relationnel

Etape 1 : Traitement des entités

- 👉 Toute entité est transformée en une table
- 👉 L'identifiant de l'entité sera la clé primaire de la table
- 👉 Chaque propriété de l'entité devient un attribut de la table

Exemple :

Etudiant
<u>Numéro</u>
Nom
Prénom
Filière

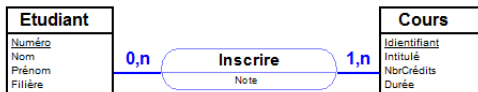
Etudiant(Numéro, Nom, Prénom, Filière)

Passage d'un schéma E/A à un MLD relationnel

Etape 2 : Traitement des associations de plusieurs à plusieurs

- 👉 Toute association ayant des cardinalités 0..n ou 1..n des deux côtés est transformée en une table
- 👉 La clé primaire de la table est constituée des identifiants des entités qui y participent
- 👉 Les propriétés de l'association deviennent des attributs de la table

Exemple :



Etudiant(Numéro, Nom, Prénom, Filière)

Cours(Identifiant, Intitulé, NbrCrédits, Durée)

Inscrire(Numéro, Identifiant, Note)

Passage d'un schéma E/A à un MLD relationnel

Etape 3 : Traitement des associations Père-Fils

- 👉 Toute association ayant sur l'une des pattes des cardinalités 0..1 ou 1..1 est supprimée
- 👉 L'identifiant de l'entité Père deviendra une clé étrangère dans l'entité Fils
- 👉 Les propriétés de l'association seront des attributs de la table correspondant à l'entité Fils

Exemple :



Etudiant(Numéro, Nom, Prénom, Filière)

Livre(ISBN, Titre, Auteur, Catégorie, Numéro)

Comment vérifier que les relations n'induisent pas des redondances/incohérences ?

Il faut normaliser les relations !

- 👉 Notion de dépendance fonctionnelle
- 👉 Première Forme Normale
- 👉 Deuxième Forme Normale
- 👉 Troisième Forme Normale
- 👉 ...

- ▶ **Langage de requêtes** (Structured Query Language)
- ▶ **Origine**
 - ▶ 1975 : QUEL
 - ▶ 1976 : Structured English QUery Langage (SEQUEL) par IBM
 - ▶ 1981 : SQL par IBM
- ▶ **Standard** ANSI-ISO, Normes SQL-86,89, SQL2 (92), SQL3 (99), ...
- ▶ **SGBDR** : Oracle, Sybase, DB2, Ingres, MS SQL Server, **MySQL**, MS Access, Postgresql, ...
- ▶ Il subsiste quelques différences selon le SGBD utilisé

- ▶ Manipulation du schéma de la base de données - LDD
création des tables, suppression des tables, création des contraintes d'intégrité, ...
- ▶ Requêtes de modification du contenu des tables - LMD
ajout, suppression, modification des n-uplets des tables, ...
- ▶ Requêtes d'interrogation sur le contenu des tables - LID
projection, tri, agrégation, sélection, jointure, ...

- Une requête interrogative (simple) est de la forme :

SELECT *⟨cible⟩* : ce que l'on veut garder

FROM *⟨liste de table(s)⟩* : à partir de quelles tables

WHERE *⟨condition(s)⟩* ; : selon des critères d'extraction éventuels

- Le résultat de la requête sera toujours une table/une relation
- Les clauses **SELECT** et **FROM** sont obligatoires

Requête interrogative - Projection

Sélectionner un ensemble de colonnes/d'attributs dans une table

Table Employes

Num	Nom	Prénom	Age	NomServe
1	Martin	Paul	31	Informatique
2	Durant	Sandrine	34	Juridique
3	Hasni	Atef	31	Informatique
4	Cure	Hector	45	Juridique
5	Condotta	Jean	42	Comptabilité

Quels sont les noms et prénoms des employés ?

```
SELECT Employes.Nom, Employes.Prénom FROM Employes ;
```

Nom	Prénom
Martin	Paul
Durant	Sandrine
Atef	Hasni
Cure	Hector
Condotta	Jean

Requête interrogative - Projection

Pour tout garder (projection sur l'ensemble des attributs) : *

```
SELECT Employes.* FROM  
Employes ;
```

OU simplement

```
SELECT * FROM Employes ;
```

Num	Nom	Prénom	Age	NomServe
1	Martin	Paul	31	Informatique
2	Durant	Sandrine	34	Juridique
3	Atef	Hasni	31	Informatique
4	Cure	Hector	45	Juridique
5	Condotta	Jean	42	Comptabilité

Requête interrogative - DISTINCT

Possibilité de doublons (en SQL), suppression avec le mot clé **DISTINCT**

Quels sont les noms des différents services des employés ?

```
SELECT NomService FROM Employes ;
```

NomService
Informatique
Juridique
Informatique
Juridique
Comptabilité

```
SELECT DISTINCT NomService FROM Employes ;
```

NomService
Informatique
Juridique
Comptabilité

Requête interrogative - Tri

- ▶ Pas d'ordre d'affichage implicite des lignes/des tuples
- ▶ Possibilité d'effectuer un tri avec la clause **ORDER BY**

Quels sont les noms et prénoms des employés triés par ordre alphabétique (lexicographique) des noms ?

```
SELECT Nom, Prénom FROM Employes ORDER BY  
Nom ASC ;
```

Nom	Prénom
Condotta	Jean
Cure	Hector
Durant	Sandrine
Hasni	Atef
Martin	Paul

```
SELECT Nom, Prénom FROM Employes ORDER BY  
Nom DESC ;
```

Nom	Prénom
Martin	Paul
Hasni	Atef
Durant	Sandrine
Cure	Hector
Condotta	Jean

Requête interrogative - Tri

- ▶ Pas d'ordre d'affichage implicite des lignes/des tuples
- ▶ Possibilité de contrôler le nombre de lignes retournées avec la clause **LIMIT**

Quels sont les noms et prénoms des trois premiers employés triés par ordre alphabétique (lexicographique) des noms ?

```
SELECT Nom, Prénom FROM Employes ORDER BY  
Nom ASC LIMIT 3;
```

Nom	Prénom
Condotta	Jean
Cure	Hector
Durant	Sandrine

Quels sont les noms et prénoms des quatre derniers employés triés par ordre alphabétique (lexicographique) des noms ?

```
SELECT Nom, Prénom FROM Employes ORDER BY  
Nom DESC LIMIT 4;
```

Nom	Prénom
Martin	Paul
Hasni	Atef
Durant	Sandrine
Cure	Hector

Requête interrogative - Fonctions d'agrégat

Elles opèrent sur un ensemble de valeurs :

- ▶ **AVG()** : moyenne des valeurs
- ▶ **SUM()** : somme des valeurs
- ▶ **MIN()** : valeur minimum
- ▶ **MAX()** : valeur maximum
- ▶ **COUNT()** : nombre de valeurs
- ▶ **COUNT(DISTINCT)** : nombre de valeurs sans les doublons

Quel est l'âge minimum des employés ?

```
SELECT MIN(Age) AS 'AgeMin' FROM Employes ;
```

AgeMin
31

Quel est l'âge moyen des employés ?

```
SELECT AVG(Age) AS 'AgeMoyen' FROM Employes ;
```

AgeMoyen
36,6

Requête interrogative - Fonctions d'agrégat

Elles opèrent sur un ensemble de valeurs :

- ▶ **AVG()** : moyenne des valeurs
- ▶ **SUM()** : somme des valeurs
- ▶ **MIN()** : valeur minimum
- ▶ **MAX()** : valeur maximum
- ▶ **COUNT()** : nombre de valeurs
- ▶ **COUNT(DISTINCT)** : nombre de valeurs sans les doublons

Quel est le nombre d'employés ?

```
SELECT COUNT(*) AS 'NbEmployes' FROM Employees;
```

NbEmployes
5

Quel est le nombre de services ?

```
SELECT COUNT(DISTINCT NomService) AS 'NbServices' FROM Employees;
```

NbServices
3

Requête interrogative - Sélection

Possibilité de sélectionner des lignes/des tuples avec différents critères

```
SELECT ...  
FROM ...  
WHERE <condition>;
```

<condition> est une condition portant sur des attributs et/ou valeurs et peut utiliser différents opérateurs :

- ▶ **opérateurs de comparaison** : <, >, <=, >=, =, <>
- ▶ **opérateurs booléens** : AND, OR, NOT
- ▶ **prédicat d'appartenance à un ensemble de valeurs** : IN
- ▶ **prédicat d'appartenance à un intervalle** : BETWEEN
- ▶ **opérateur de comparaison partielle** : LIKE
- ▶ ...

Seuls les lignes/les tuples satisfaisant la condition sont gardés

Quels sont les noms et prénoms des employés travaillant au service Informatique ?

```
SELECT Nom, Prénom FROM EMPLOYES WHERE  
NomService = 'Informatique' ;
```

Nom	Prénom
Martin	Paul
Hasni	Atef

Quels sont les noms et prénoms des employés de plus 40 ans travaillant au service Informatique ou au service Juridique ?

```
SELECT Nom, Prénom FROM Employes WHERE (Nom-  
Service='Informatique' OR NomService='Juridique')  
AND (Age > 40) ;
```

Nom	Prénom
Cure	Hector
Condotta	Jean

Seuls les lignes/les tuples satisfaisant la condition sont gardés

Quels sont les noms et prénoms des employés dont l'âge est compris entre 30 et 40 ans ?

```
SELECT Nom, Prénom FROM Employés WHERE Age  
BETWEEN 30 AND 40;
```

Nom	Prénom
Martin	Paul
Durant	Sandrine
Hasni	Atef

Quels sont les noms et prénoms des employés dont le nom fini par n ?

```
SELECT Nom, Prénom FROM EMPLOYES WHERE  
Nom LIKE '%n';
```

Nom	Prénom
Martin	Paul

Seuls les tuples/les lignes satisfaisant la condition sont gardés

Quels sont les noms et prénoms des employés dont le nom commence par la lettre C ?

```
SELECT Nom, Prénom FROM EMPLOYES WHERE  
Nom LIKE 'C%';
```

Nom	Prénom
Cure	Hector
Condotta	Jean

Quels sont les noms et prénoms des employés dont le nom contient la chaîne de caractère 'ur' ?

```
SELECT Nom, Prénom FROM EMPLOYES WHERE  
Nom LIKE '%ur%';
```

Nom	Prénom
Durant	Sandrine

Fonctions de groupe :

```
SELECT ...  
FROM ...  
WHERE <condition>  
[GROUP BY attributs]  
[HAVING attributs];
```

avec :

- ▶ **GROUP BY** : grouper les résultats d'une requête selon les valeurs d'un ou plusieurs attributs en vue d'appliquer des fonctions d'agrégation à ces groupes.
- ▶ **Attributs** : représentent une liste d'attributs séparés par des virgules.
- ▶ **HAVING** : la clause HAVING est pour les groupes de tuples ce qu'est la clause WHERE pour les tuples individuellement.

Table Employes

Num	Nom	Prénom	Age	NomService
1	Martin	Paul	31	Informatique
2	Durant	Sandrine	34	Juridique
3	Atef	Hasni	31	Informatique
4	Cure	Hector	45	Juridique
5	Condotta	Jean	42	Comptabilité

Quel est le nombre d'employés par service ?

```
SELECT NomService, count(*) as 'NbEmployes' FROM  
Employes GROUP BY NomService;
```

NomService	NbEmployes
Informatique	2
Juridique	2
Comptabilité	1

Table Employes

Num	Nom	Prénom	Age	NomService
1	Martin	Paul	31	Informatique
2	Durant	Sandrine	34	Juridique
3	Atef	Hasni	31	Informatique
4	Cure	Hector	45	Juridique
5	Condotta	Jean	42	Comptabilité

Quels sont pour chaque service et pour chaque âge le nombre d'employés ?

```
SELECT NomService, Age, count(*) as 'NbEmp' FROM  
Employes GROUP BY NomService, Age ;
```

NomService	Age	NbEmp
Informatique	31	2
Juridique	34	1
Juridique	45	1
Comptabilité	42	1

Table Employes

Num	Nom	Prénom	Age	NomService
1	Martin	Paul	31	Informatique
2	Durant	Sandrine	34	Juridique
3	Atef	Hasni	31	Informatique
4	Cure	Hector	45	Juridique
5	Condotta	Jean	42	Comptabilité

Quelle est la valeur minimale et la valeur maximale des âges des employés par service, uniquement pour les services dont le nom commence par 'J' ou 'C' ?

NomService	Min	Max
Juridique	34	45
Comptabilité	42	42

```
SELECT NomService, min(Age) as 'Min', max(Age) as  
'Max' FROM Employes GROUP BY NomService HA-  
VING NomService like 'J%' OR NomService like 'C%';
```

Table Employes

Num	Nom	Prénom	Age	NomService
1	Martin	Paul	31	Informatique
2	Durant	Sandrine	34	Juridique
3	Atef	Hasni	31	Informatique
4	Cure	Hector	45	Juridique
5	Condotta	Jean	42	Comptabilité

Quel est le nombre de personnes par service dont l'âge est supérieure ou égale à 25 pour les services dont le nom commence par 'I' ?

NomService	NbEmp
Informatique	2

```
SELECT NomService, count(*) as 'NbEmp' FROM Em-  
ployes WHERE age >= 25 GROUP BY NomService HA-  
VING NomService like 'I%';
```


Une jointure permet de relier entre elles les données réparties dans différentes tables

- ▶ Jointure avec un critère d'égalité d'attributs : **equi-jointure**
- ▶ Jointure avec un critère d'égalité d'attributs de même nom : **jointure naturelle**
- ▶ Jointure avec qualification : **jointure avec un critère quelconque**
- ▶ Etc ...

Requête interrogative - Jointure

Une jointure permet de relier entre elles les données réparties dans différentes tables

Num	Nom	Prénom	Age	NomServe	IndiceSal
1	Martin	Paul	31	Informatique	234
2	Durant	Sandrine	34	Juridique	234
3	Atef	Hasni	31	Informatique	128
4	Cure	Hector	45	Juridique	484
5	Condotta	Jean	42	Comptabilité	350

Indice	Montant
128	1200
234	2100
350	3300
484	4700

Quels sont les salaires des employés de plus de 33 ans ?

```
SELECT Nom, Prénom, Montant FROM Employes, Salaires WHERE Employes.IndiceSal = Salaires.Indice AND Age > 33;
```

Nom	Prénom	Montant
Durant	Sandrine	2100
Cure	Hector	4700
Condotta	Jean	3300