



### ► Bases de données

Master 1 Humanités numériques  
2018-2019  
Jérôme Darmont  
<http://eric.univ-lyon2.fr/~jdarmont/>

### Actualité du cours

 [http://eric.univ-lyon2.fr/~jdarmont/?page\\_id=3139](http://eric.univ-lyon2.fr/~jdarmont/?page_id=3139)


 <http://eric.univ-lyon2.fr/~jdarmont/?feed=rss2>

 [https://twitter.com/darmont\\_lyon2](https://twitter.com/darmont_lyon2) #hnbd

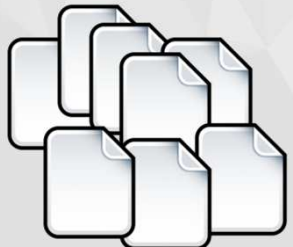
Bases de données <http://eric.univ-lyon2.fr/~jdarmont/> 2

### Définition

Base de données (BD) : Collection de données **cohérentes** et **structurées**





 Base de données




$\neq$



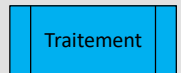
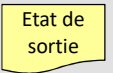
 Fichiers

Bases de données <http://eric.univ-lyon2.fr/~jdarmont/> 3

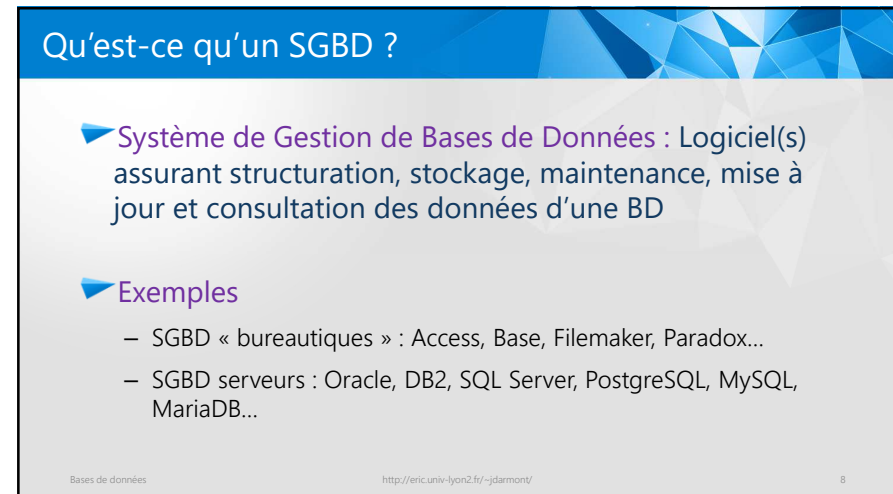
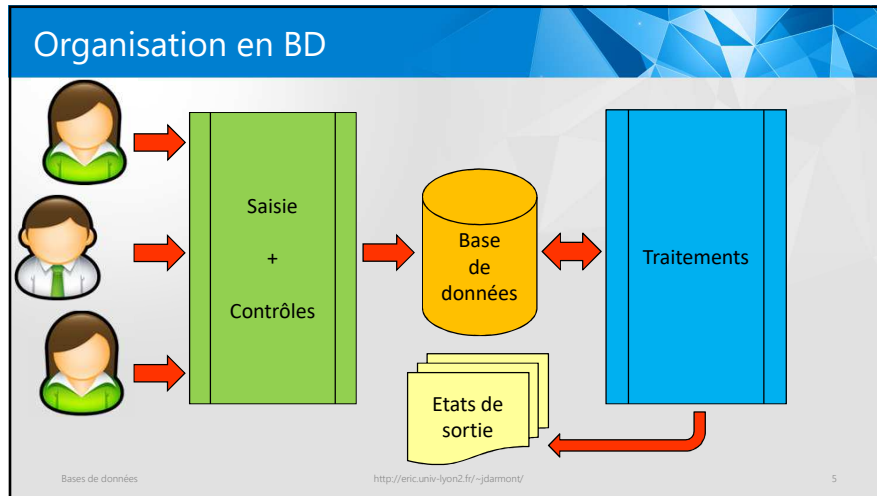
### Organisation en fichiers

 →  →  → 

 →  → 

 →  →  → 

Bases de données <http://eric.univ-lyon2.fr/~jdarmont/> 4



## Objectifs des SGBD

- ▶ Indépendance physique
- ▶ Indépendance logique
- ▶ Manipulation facile des données
- ▶ Administration facile des données
- ▶ Efficacité des accès aux données
- ▶ Redondance contrôlée des données
- ▶ Cohérence des données (contraintes d'intégrité)
- ▶ Partage des données
- ▶ Sécurité des données

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 9

## Fonctions des SGBD

- ▶ Description des données : Langage de Définition de Données (LDD)
- ▶ Recherche des données
- ▶ Mise à jour des données
- ▶ Transformation des données
- ▶ Contrôle de l'intégrité des données
- ▶ Gestion de transactions et sécurité

} Langage de Manipulation de Données (LMD)

} Langage de Contrôle de Données (LCD)

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 10

## Processus de conception d'une BD

Problématique → Rédaction → Spécifications → Analyse → Modèle conceptuel → Traduction → Modèle logique → Traduction → Modèle physique

*Cahier des charges* (between Problématique and Spécifications)

Indépendant d'un système de gestion de BD (SGBD) (above Spécifications)

*Famille de SGBD* (above Modèle conceptuel)

*SGBD particulier* (above Modèle physique)

Spécifique (to the right of the process flow)

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 11

## Plan du cours

- ▶ Séances 1 et 2 : Conception de bases de données
  - UNIFIED MODELING LANGUAGE
  - Modèle relationnel
  - Access
- ▶ Séances 3 et 4 : Interrogation de bases de données
  - Access
  - SQL
  - Excel

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 12

## Partie 1 Modélisation conceptuelle

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 13

## Modèle conceptuel UML

- ▶ Standard de l'Object Management Group
- ▶ Ensemble de formalismes graphiques
- ▶ Diagramme de classes

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 14

## Classes et attributs

- ▶ **Classe** : Groupe d'entités du monde réel ayant les mêmes caractéristiques et le même comportement  
ex. **ETUDIANT**
- ▶ **Attribut** : Propriété de la classe  
ex. **Nom et Prénom de l'étudiant-e**
- ▶ **Représentation graphique** :

| ETUDIANT |
|----------|
| Nom      |
| Prénom   |

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 15

## Types des attributs

- ▶ **Type d'attribut** :
  - Nombre entier (**Entier**)
  - Nombre réel (**Réel**)
  - Chaîne de caractères (**Chaîne**)
  - Date (**Date**)

| ETUDIANT         |
|------------------|
| Nom : Chaîne     |
| Prénom : Chaîne  |
| DateNaiss : Date |
| Rue : Chaîne     |
| CP : Entier      |
| Ville : Chaîne   |

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 16

## Instances

- Objets (individus) de la classe ETUDIANT = les étudiant-es

| Nom       | Prénom    | DateNaiss  | Etc. |
|-----------|-----------|------------|------|
| Dupont    | Albertine | 01/06/1993 | ...  |
| West      | James     | 03/09/1994 | ...  |
| Martin    | Marie     | 05/06/1995 | ...  |
| Abidi     | Rachid    | 15/11/1995 | ...  |
| Titgoutte | Justine   | 28/02/1996 | ...  |
| Dupont    | Noémie    | 18/09/1995 | ...  |
| Dupont    | Albert    | 23/05/1990 | ...  |

**Problème :** Comment distinguer les Dupont ?

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 17

## Identifiant (1/2)

- Solution : Ajouter un attribut **numéro d'étudiant** !

| NumEtu | Nom       | Prénom    | DateNaiss  |
|--------|-----------|-----------|------------|
| 1110   | Dupont    | Albertine | 01/06/1993 |
| 2002   | West      | James     | 03/09/1994 |
| 3333   | Martin    | Marie     | 05/06/1995 |
| 4042   | Durand    | Rachid    | 05/11/1995 |
| 5552   | Titgoutte | Justine   | 28/02/1996 |
| 6789   | Dupont    | Noémie    | 18/09/1995 |
| 7000   | Dupont    | Albert    | 23/05/1990 |

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 18

## Identifiant (2/2)

- Le numéro d'étudiant est un attribut **identifiant**.
- Un identifiant caractérise **de façon unique** les instances d'une classe.
- Convention graphique :**  
NB : Ne pas confondre avec les attributs de classe UML dont c'est la notation usuelle

|                        |
|------------------------|
| ETUDIANT               |
| <u>NumEtu</u> : Entier |

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 19

## Associations

- Association :** liaison perçue entre des classes  
**ex. Les étudiant-es passent des épreuves.**

```

graph LR
    ETUDIANT --- PASSER --- EPREUVE
  
```

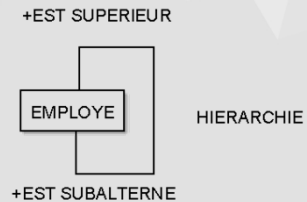
- Les classes ETUDIANT et EPREUVE peuvent être qualifiées de **participantes** à l'association PASSER.
- Degré ou arité** d'une association : nombre de classes participantes.  
En général : **associations binaires** (de degré 2).

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 20

## Associations récursives

- Une classe peut être associée à elle-même, chaque instance pouvant jouer plusieurs rôles dans l'association.  
ex. Employés et supérieurs hiérarchiques

- Rôle : fonction de chaque classe participante (+).



Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

21

## Multiplicité (ou cardinalité)

- Définition :** Indicateur qui montre combien d'instances de la classe considérée peuvent être liées à une instance de l'autre classe participant à l'association

- 1 Un et un seul
- 0..1 Zéro ou un
- 0..\* ou \* Zéro ou plus
- 1..\* Un ou plus
- M..N De M à N (M, N entiers)  
ex. 4..10 (de 4 à 10)

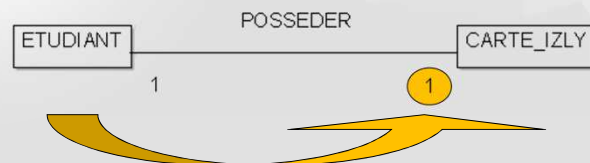
Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

22

## Associations 1-1

- ex. Un-e étudiant-e possède une et une seule carte Izly. Cette dernière n'est possédée que par un-e seul-e étudiant-e.



Lire « Un-e étudiant.e possède multiplicité (1) carte Izly ».

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

23

## Associations 1-N

- ex. Une épreuve relève d'une et une seule matière. Une matière peut donner lieu à plusieurs épreuves.



**NB :** La multiplicité un à plusieurs (1..\*) peut aussi être zéro à plusieurs (0..\* ou \*).

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

24

## Associations 0 ou 1-N

- ex. Un-e étudiant-e peut appartenir ou non à un groupe de TD. Un groupe de TD réunit plusieurs étudiant-es.



**NB :** La multiplicité un à plusieurs (1..\*) peut aussi être zéro à plusieurs (0..\* ou \*).

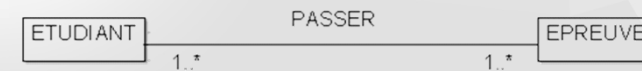
Bases de données

http://eric.univ-lyon2.fr/~jdamont/

25

## Associations M-N

- ex. Un-e étudiant-e peut passer plusieurs épreuves. Une épreuve peut être passée par plusieurs étudiant-es.



**NB :** Les multiplicités un à plusieurs (1..\*) peuvent aussi être zéro à plusieurs (0..\* ou \*).

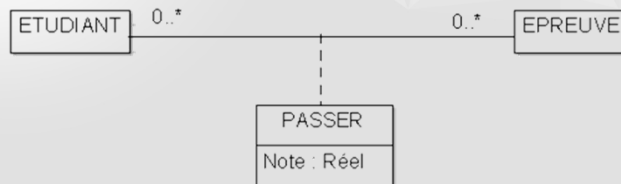
Bases de données

http://eric.univ-lyon2.fr/~jdamont/

26

## Classes-associations

- Il est possible de caractériser une association par des attributs.
- ex. Un-e étudiant-e qui passe une épreuve obtient une note.



**NB :** Une classe-association demeure une association.

Bases de données

http://eric.univ-lyon2.fr/~jdamont/

27

## Exemple : Spécifications (1/2)

- Les étudiant-es sont caractérisé-es par un numéro unique, leur nom, prénom, date de naissance, rue, code postal et ville.
- Les étudiant-es possèdent une carte Izly caractérisée par un numéro unique et un solde d'argent utilisable au CROUS.
- Selon qu'ils ou elles sont dispensé-es ou non d'assiduité, les étudiant-es appartiennent à un groupe de TD caractérisé par un code unique.

Bases de données

http://eric.univ-lyon2.fr/~jdamont/

28

## Exemple : Spécifications (2/2)

- Les étudiant-es passent des **épreuves** et obtiennent une note pour chacune.
- Les épreuves sont caractérisées par un **code**, ainsi que la **date** et le **lieu** auxquels elles se déroulent.
- Chaque épreuve relève d'une **matière** unique (mais une matière donnée peut donner lieu à plusieurs épreuves).
- Les matières sont caractérisées par un **code** et un **intitulé**.

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

29

## Démarche de modélisation conceptuelle

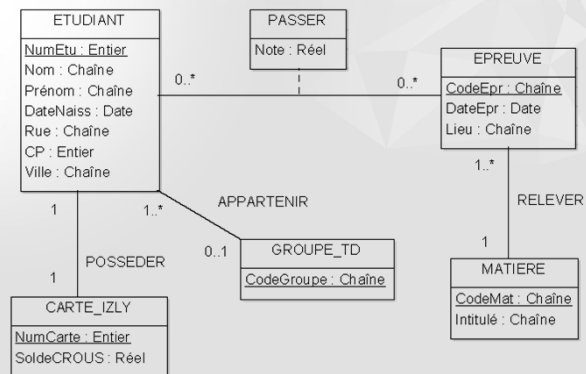
1. Identifier les **classes**
2. Identifier les **associations** entre les classes
3. Identifier les **attributs** de chaque classe et de chaque classe-association
4. Identifier et souligner l'**identifiant** de chaque classe
5. Évaluer les **multiplicités** des associations

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

30

## Exemple : Diagramme de classes

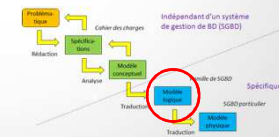


Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

31

## Partie 2 Modélisation logique



Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

32



## Modèle logique relationnel

- ▶ **Modèle** associé aux SGBD relationnels  
(ex. Oracle, SQL Server, DB2, MySQL, Access...)
- ▶ **Objectifs du modèle relationnel**
  - Indépendance physique
  - Traitement du problème de redondance des données
  - Langages non procéduraux (faciles à utiliser)
  - Devenir un standard

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

33

## Relations et attributs

- ▶ Une **relation**  $R$  est un ensemble d'**attributs**  $\{A_1, A_2, \dots, A_n\}$ .  
ex. La relation EPREUVE est l'ensemble des attributs  $\{\text{CodeEpr}, \text{DateEpr}, \text{Lieu}\}$ .
- ▶ Chaque attribut  $A_i$  prend ses valeurs dans un **domaine**  $\text{dom}(A_i)$ .  
ex.  $\text{Note} \in [0, 20]$   
 $\text{Lieu} \in \{\text{'Amphi Say'}, \text{'Amphi Aubrac'}, \text{'Salle D101'}, \dots\}$

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

34

## N-uplets

- ▶ **Notation d'une relation** :  $R(A_1, A_2, \dots, A_n)$   
ex. EPREUVE (CodeEpr, DateEpr, Lieu)
- ▶ Un **n-uplet**  $t$  est un ensemble de valeurs  $t = \langle V_1, V_2, \dots, V_n \rangle$   
où  $V_i \in \text{dom}(A_i)$  ou bien  $V_i$  est la valeur nulle (NULL).  
ex.  $\langle \text{'InfoS2'}, \text{'30-06-2016'}, \text{'Amphi Aubrac'} \rangle$  est un n-uplet de la relation EPREUVE.

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

35

## Contraintes d'intégrité (1/2)

- ▶ **Clé primaire** : Ensemble d'attributs dont les valeurs permettent de distinguer les n-uplets les uns des autres.  
ex. CodeEpr est clé primaire de la relation EPREUVE.
- ▶ **Clé étrangère** : Attribut qui est clé primaire d'une autre relation.  
ex. Connaître la matière dont relève chaque épreuve  
⇒ ajout de l'attribut CodeMat à la relation EPREUVE

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

36

## Contraintes d'intégrité (2/2)

- ▶ **Notations** : Clés primaires soulignées, clés étrangères postfixées par le caractère #.

ex. EPREUVE (CodeEpr, DateEpr, Lieu, CodeMat#)

- ▶ **Contraintes de domaine** : Les attributs doivent respecter une condition logique.

ex. Note  $\geq 0$  ET Note  $\leq 20$

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

37

## Contraintes d'intégrité en pratique

### EPREUVE

| CodeEpr  | DateEpr    | Lieu         | Codemat# |
|----------|------------|--------------|----------|
| ECOS101  | 15/01/2016 | Amphi Aubrac | ECO      |
| ECOS102  | 16/01/2016 | Amphi Aubrac | ECO      |
| GESS201  | 25/05/2016 | Salle 201    | GES      |
| INFOS101 | 20/01/2016 | Salle 101    | INFO     |

### MATIERE

| CodeMat | Intitulé     |
|---------|--------------|
| ECO     | Économie     |
| GES     | Gestion      |
| INFO    | Informatique |

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

38

## Traduction UML-relationnel (1/4)

- ▶ Chaque **classe** devient une **relation**.
- ▶ Les **attributs** de la classe deviennent **attributs** de la relation.
- ▶ L'**identifiant** de la classe devient **clé primaire** de la relation.

ex. ETUDIANT (NumEtu, Nom, Prénom, DateNaiss, Rue, CP, Ville)

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

39

## Traduction UML-relationnel (2/4)

- ▶ Chaque **association 1-1** est prise en compte en incluant la clé primaire d'une des relations participante comme **clé étrangère** dans l'autre relation.

ex. CARTE\_IZLY (NumCarte, SoldeCROUS)

ETUDIANT (NumEtu, Nom, Prénom, DateNaiss, Rue, CP, Ville, **NumCarte#**)

Bases de données

<http://eric.univ-lyon2.fr/~jdamont/>

40

### Traduction UML-relationnel (3/4)

- Chaque association 1-N est prise en compte en incluant la clé primaire de la relation dont la multiplicité maximale est 1 comme clé étrangère dans l'autre relation participante.

ex. EPREUVE (CodeEpr, DateEpr, Lieu, CodeMat#)

MATIERE (CodeMat, Intitulé)

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 41

### Traduction UML-relationnel (4/4)

- Chaque association M-N est prise en compte en créant une nouvelle relation dont la clé primaire est la concaténation des clés primaires des relations participantes. Les attributs de la classe-association sont insérés dans cette nouvelle relation si nécessaire.

ex. PASSER (NumEtu#, CodeEpr#, Note)

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 42

### Exemple : Modèle logique relationnel

CARTE\_IZLY (NumCarte, SoldeCROUS)  
 GROUPE\_TD (CodeGroupe)  
 ETUDIANT (NumEtu, Nom, Prénom, DateNaiss, Rue, CP, Ville, NumCarte#, CodeGroupe#)  
 MATIERE (CodeMat, Intitulé)  
 EPREUVE (CodeEpr, DateEpr, Lieu, CodeMat#)  
 PASSER (NumEtu#, CodeEpr#, Note)

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 43

### Traduction d'une association M-N

| ETUDIANT |        |           |
|----------|--------|-----------|
| NumEtu   | Nom    | Prénom    |
| 1110     | Dupont | Albertine |
| 2002     | West   | James     |

| EPREUVE  |            |        |
|----------|------------|--------|
| CodeEpr  | DateEpr    | Lieu   |
| ECOS101  | 15/01/2016 | Aubrac |
| ECOS102  | 16/01/2016 | Aubrac |
| GESS201  | 25/05/2016 | D201   |
| INFOS101 | 20/01/2016 | D101   |

| PASSER (table « pont ») |          |      |
|-------------------------|----------|------|
| NumEtu#                 | CodeEpr# | Note |
| 1110                    | INFOS101 | 15,5 |
| 2002                    | ECOS101  | 8,5  |
| 2002                    | ECOS102  | 13   |
| 1110                    | GESS201  | 14   |
| 2002                    | GESS201  | 14,5 |

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 44

## Problème de la redondance

- ▶ Lorsque l'on effectue directement une modélisation logique  
ex. Soit la relation `PASSER_EPREUVE`.

| NumEtu | Note | CodeEpr  | Lieu         |
|--------|------|----------|--------------|
| 1110   | 15,5 | INFOS101 | Amphi Aubrac |
| 1110   | 14,0 | ECOS101  | Amphi Aubrac |
| 2002   | 13,0 | ECOS102  | Salle D201   |
| 3333   | 10,5 | INFOS101 | Amphi Aubrac |

Cette relation présente différentes anomalies.

Bases de données

<http://eric.univ-lyon2.fr/~jdarmont/>

45

## Anomalies liées à la redondance

- ▶ Anomalies de modification : Si l'on souhaite mettre à jour le lieu d'une épreuve, il faut le faire pour tous les n-uplets concernés.
- ▶ Anomalies d'insertion : Pour ajouter une nouvelle épreuve, il faut obligatoirement fournir des valeurs pour `NumEtu` et `Note`.
- ▶ Anomalies de suppression  
ex. La suppression de l'étudiant n° 2002 fait perdre toutes les informations concernant l'épreuve `ECOS102`.

Bases de données

<http://eric.univ-lyon2.fr/~jdarmont/>

46

## Éviter la redondance

- ▶ Pourquoi ?
  - Suppression des problèmes de mise à jour
  - Minimisation de l'espace de stockage
- ▶ Comment ?
  - Dans le modèle conceptuel, ne spécifier que des attributs **non décomposables** (première forme normale).  
ex. Une adresse doit être décomposée en rue, code postal, ville...
  - C'est tout !

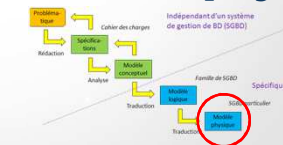
Bases de données

<http://eric.univ-lyon2.fr/~jdarmont/>

47

## Partie 3 Modélisation physique

TP sur



Bases de données

<http://eric.univ-lyon2.fr/~jdarmont/>

48