

UFR des sciences économiques et de gestion

Databases (Bases de données)


Master 1 Monnaie Banque Finance Assurance
 Année 2018-2019
 Jérôme Darmont

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

Actualité du cours


http://eric.univ-lyon2.fr/~jdarmont/?page_id=447


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


https://twitter.com/darmont_lyon2
#bdmbfa

Conception de sites Web http://eric.univ-lyon2.fr/~jdarmont/ 1

Planning des séances

Jour	Horaire	Salle
Vendredi 28/09/18	12h00-13h45	Amphi Fugier
Vendredi 12/10/18	12h00-13h45	Amphi Fugier
Vendredi 19/10/18	12h00-13h45	Amphi Fugier
Vendredi 09/11/18	12h00-13h45	Amphi Fugier
Vendredi 16/11/18	12h00-13h45	Amphi Fugier
Vendredi 23/11/18	12h00-13h45	Amphi Fugier

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


Plan du cours

- Introduction
- Modèle UML
- Modèle relationnel
- Langage SQL

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

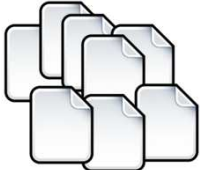
Qu'est-ce qu'une base de données (BD) ?

Collection de données **cohérentes** et **structurées**



Base de données

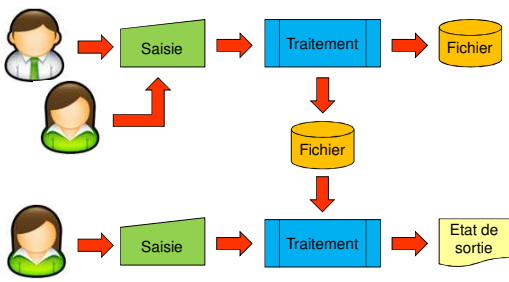
≠



Fichiers

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

Organisation en fichiers

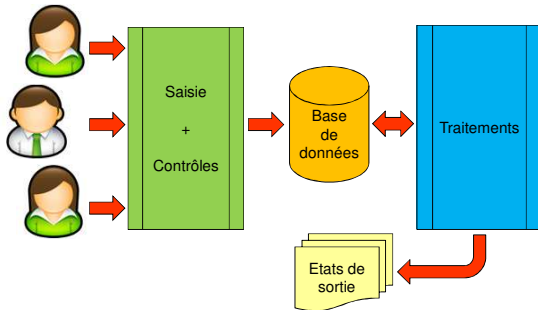


```

    graph TD
      User1[User] --> Saisie1[Saisie]
      Saisie1 --> Traitement1[Traitement]
      Traitement1 --> Fichier1[Fichier]
      Fichier1 --> Fichier2[Fichier]
      Fichier2 --> Traitement2[Traitement]
      Saisie2[Saisie] --> Traitement2
      Traitement2 --> Etat[Etat de sortie]
  
```

Bases de données http://eric.univ-lyon2.fr/~jdarmont/ 5

Organisation base de données



Bases de données

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

6

Avantages de l'organisation BD

- Uniformisation de la saisie
- Standardisation des traitements
- Contrôle immédiat de la validité des données
- Partage de données entre plusieurs traitements

Bases de données

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

7

Propriétés de l'organisation BD

- Usage multiple des données
- Accès facile, rapide, protégé, souple, puissant
- Coût réduit de stockage, de mise à jour et de saisie
- Disponibilité, exactitude, cohérence et protection des données ; **non redondance**
- Évolution aisée et protection de l'investissement de programmation
- Indépendance des données et des programmes
- Conception *a priori*

Bases de données

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

8

Qu'est-ce qu'un SGBD ?

- **Système de Gestion de Bases de Données** : Logiciel(s) assurant structuration, stockage, maintenance, mise à jour et consultation des données d'une BD
- Exemples
 - SGBD « bureautiques » : Access, Base, Filemaker, Paradox...
 - SGBD serveurs : Oracle, DB2, SQL Server, PostgreSQL, MySQL, MariaDB...

Bases de données

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

9

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/~jdarmon/>

10

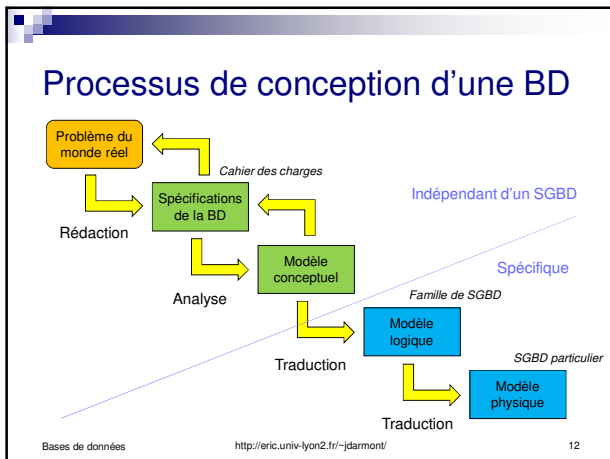
Fonctions des SGBD

- 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/~jdarmon/>

11



Sondage express

N° de la question : 796



Qu'avez-vous compris jusqu'ici ?

- A. L'utilité des bases de données
- B. Le processus de conception des bases de données
- C. Les deux !
- D. Rien du tout...

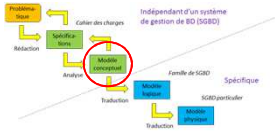
Répondre sur <http://toreply.univ-lille1.fr>

Question n° 796

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

- ## Plan du cours
- ✓ Introduction
 - Modèle UML
 - Modèle relationnel
 - Langage SQL
- Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 14

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/> 15

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/> 16

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/> 17

Instances

- Objets 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	...
Durand	Rachid	15/11/1995	...
Titgoutte	Justine	28/02/1996	...
Dupont	Noémie	18/09/1995	...
Dupont	Albert	23/05/1990	...

Problème : Comment distinguer les Dupont ?

Identifiant (1/2)

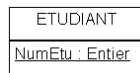
- Solution : Ajouter un attribut **numéro d'étudiant** !

NumEtu	Nom	Prénom	DateNaiss
1110	Dupont	Albertin	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

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



Sondage express



La notion d'identifiant vous paraît-elle claire ?

Répondre sur <http://toreply.univ-lille1.fr>

Question n° 843

Associations

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

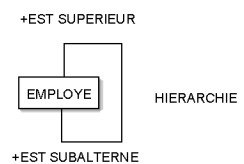


- Les classes ETUDIANT et EPREUVE peuvent être qualifiées de **participantes** à l'association PASSER.
- Degré ou **arité** d'une association : nombre de classes participantes. En général : associations binaires (de degré 2).

Associations récursives

- Association récursive : une même instance de classe peut jouer plusieurs rôles dans la même association.
ex. Employé-es et supérieur-es hiérarchiques

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



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)

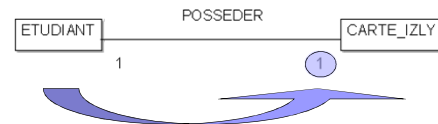
Bases de données

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

24

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/~jdarmon/>

25

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/~jdarmon/>

26

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/~jdarmon/>

27

Associations M-N

- **ex.** Un 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/~jdarmon/>

28

Quizz

« Les clients achètent des produits » est une association :



- A. 1-1
- B. 1-N (1 à plusieurs)
- C. 0 ou 1-N (0 ou 1 à plusieurs)
- D. M-N (plusieurs à plusieurs)

Répondre sur <http://toreply.univ-lille1.fr>

Question n° 936

Bases de données

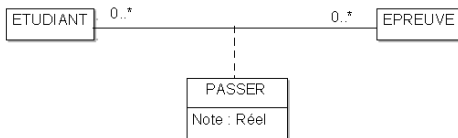
<http://eric.univ-lyon2.fr/~jdarmon/>

29

Classes-associations

- Il est possible de caractériser une association par des attributs.

ex. Un-e étudiant-e obtient une note par épreuve passée.



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

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**.

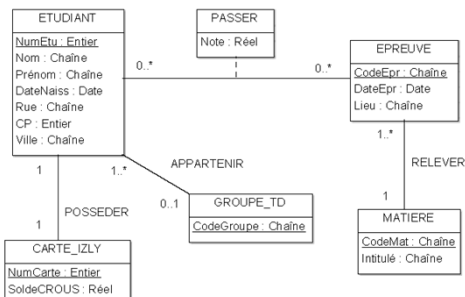
Exemple : spécifications (2/2)

- 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**.

Démarche de modélisation conceptuelle

- Identifier les **classes**
- Identifier les **associations** entre les classes
- Identifier les **attributs** de chaque classe et de chaque classe-association
- Identifier et souligner l'**identifiant** de chaque classe
- Évaluer les **multiplicités** des associations

Exemple : diagramme de classes



Quizz

Dans le diagramme précédent, **PASSER** est :

- Une classe
- Une association
- Les deux !
- Obiwan Kenobi

Répondre sur <http://toreply.univ-lille1.fr>

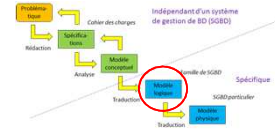
Question n° 256



Plan du cours

- ✓ Introduction
- ✓ Modèle UML
- Modèle relationnel
- Langage SQL

Modèle logique relationnel



- Modèle associé aux SGBD relationnels
- Objectifs du modèle relationnel :
 - Indépendance physique
 - Traitement du problème de redondance des données
 - LMD non procéduraux (faciles à utiliser)
 - Devenir un standard

Caractéristiques des systèmes relationnels

- Langages d'interrogation puissants et déclaratifs
- Accès orienté valeur
- Grande simplicité, absence de considérations physiques
- Description du schéma très réduite
- LDD intégré au LMD
- Grande dynamique de structure
- Optimisation de requêtes
- Utilisation interactive ou à partir d'un langage hôte

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. Note $\in [0, 20]$
Lieu $\in \{\text{'Amphi Aubrac'}, \text{'Amphi Say'}, \text{'Salle D101'}, \text{'Salle D201'}, \dots\}$

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 V_i est la valeur nulle (NULL).
ex. $\langle \text{'InfoS2'}, \text{'30-06-2011'}, \text{'Amphi Aubrac'} \rangle$ est un n-uplet de la relation EPREUVE.

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
 \Rightarrow ajout de l'attribut CodeMat à la relation EPREUVE

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 ≥ 0 ET Note ≤ 20

Contraintes d'intégrité en pratique

EPREUVE				MATIERE	
CodeEpr	DateEpr	Lieu	CodeMat#	Codemat	Intitulé
ECOS101	15/01/2011	Amphi Aubrac	ECO	ECO	Économie
ECOS102	16/01/2011	Amphi Aubrac	ECO	GES	Gestion
GESS201	25/05/2011	Salle 201	GES	INFO	Informatique
INFOS101	20/01/2011	Salle 101	INFO		

Sondage express



La notion de clé étrangère vous paraît-elle claire?

Répondre sur <http://toreply.univ-lille1.fr>

Question n° 475

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)

Traduction UML-relationnel (2/4)

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

ex. CARTE_IZLY (NumCarte, SoldeCROUS)
ETUDIANT (NumEtu, Nom, Prénom, DateNaiss, Rue, CP, Ville, NumCarte#)

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.

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

Traduction UML-relationnel (4/4)

- Chaque association M-N est prise en compte en créant une nouvelle relation dont la clé primaire et 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)

Exemple : modèle logique relationnel

CARTE_IJLY (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)

Traduction d'une association M-N

ETUDIANT

NumEtu	Nom	Prénom
1110	Dupont	Albertine
2002	West	James

PASSER (table « pont »)

NumEtu#	CodeEpr#	Note
1110	INFOS101	15,5
2002	ECOS101	8,5
2002	ECOS102	13
1110	GESS201	14
2002	GESS201	14,5

EPREUVE

CodeEpr	DateEpr	Lieu
ECOS101	15/01/2016	Aubrac
ECOS102	16/01/2016	Aubrac
GESS201	25/03/2016	D201
INFOS101	20/01/2016	D101

Sondage express

Le tableau d'équivalence suivant vous paraît-il correct ?



UML	Relationnel
Classe	Relation
Attribut	Attribut
Identifiant	Clé primaire
Association 1-1 ou 1-N	Clé étrangère
Association M-N	Relation « pont »

Répondre sur <http://toreply.univ-lille1.fr>

Question n° 218

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	INFO1	Amphi Aubrac
1110	14,0	ECO1	Amphi Aubrac
2002	13,0	ECO2	Salle D201
3333	10,5	INFO1	Amphi Aubrac

Cette relation présente différentes anomalies.

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-e n° 2002 fait perdre toutes les informations concernant l'épreuve ECOS2.

Éviter la redondance

- Pourquoi ?
 - Suppression des problèmes de mise à jour
 - Minimisation de l'espace de stockage
- Comment ?
 - Ne pas concevoir directement un modèle logique
 - 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/~jdarmon/>

54

Quizz

La redondance, c'est :

- A. Vraiment Très Mal™
- B. Parfois nécessaire
- C. Une figure de style littéraire

Répondre sur <http://toreply.univ-lille1.fr>

Question n° 473



Bases de données

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

55

Algèbre relationnelle

- Ensemble d'opérateurs qui s'appliquent aux relations
 - **Résultat** : nouvelle relation qui peut à son tour être manipulée
- ⇒ L'algèbre relationnelle permet d'effectuer des recherches dans les relations.

Bases de données

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

56

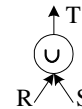
Opérateurs ensemblistes

- Union : $T = R \cup S$ (notation algébrique)
ou $T = \text{UNION}(R, S)$ (notation fonctionnelle)

R et S doivent avoir même schéma.

ex. R et S sont les relations ETUDIANT de deux formations (ex. anciens M1 Finance et Eco-Société) fusionnées pour constituer une liste d'émargement commune.

Notation graphique :



Bases de données

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

57

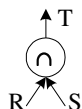
Opérateurs ensemblistes

- Intersection : $T = R \cap S$
ou $T = \text{INTERSECT}(R, S)$

R et S doivent avoir même schéma.

ex. Permet de trouver les étudiant-es commun-es à deux formations.

Notation graphique :



Bases de données

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

58

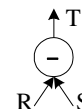
Opérateurs ensemblistes

- Différence : $T = R - S$
ou $T = \text{MINUS}(R, S)$

R et S doivent avoir même schéma.

ex. Permet de retirer les étudiant-es de la relation S existant dans la relation R.

Notation graphique :



Bases de données

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

59

Opérateurs ensemblistes

- **Produit cartésien :** $T = R \times S$
ou $T = \text{PRODUCT}(R, S)$
Associe chaque n-uplet de R à chaque n-uplet de S.
Notation graphique :

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 60

Produit cartésien

ex.

NumEtu	Nom
101	E1
102	E2

 \times

CodeEpr	Lieu
INFO1	Aubrac
ECO1	Aubrac
ECO2	D201

NumEtu	Nom	CodeEpr	Lieu
101	E1	INFO1	Aubrac
102	E2	INFO1	Aubrac
101	E1	ECO1	Aubrac
102	E2	ECO1	Aubrac
101	E1	ECO2	D201
102	E2	ECO2	D201

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 61

Opérateurs ensemblistes

- **Division :** $T = R \div S$
ou $T = \text{DIVISION}(R, S)$
 $R(A_1, A_2, \dots, A_n)$ $S(A_{p+1}, \dots, A_n)$
 $T(A_1, A_2, \dots, A_p)$ contient tous les n-uplets tels que leur concaténation à chacun des n-uplets de S donne toujours un n-uplet de R.
Notation graphique :

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 62

Division

ex.

NumEtu	CodeEpr	Note
101	INFO1	11
101	ECO1	15
101	ECO2	12
102	ECO1	9
103	INFO1	11
103	ECO2	12

 \div

CodeEpr	Note
INFO1	11
ECO2	12

NumEtu
101
103

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 63

Opérateurs spécifiques

- **Projection :** $T = \Pi \langle A, B, C \rangle (R)$
ou $T = \text{PROJECT}(R / A, B, C)$
T ne contient que les attributs A, B et C de R.
ex. Noms et prénoms des étudiant-es.
Notation graphique :

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 64

Opérateurs spécifiques

- **Restriction :** $T = \sigma \langle C \rangle (R)$
ou $T = \text{RESTRICT}(R / C)$
T ne contient que les attributs de R qui satisfont la condition C.
ex. C = Étudiant-es qui habitent à Lyon.
Notation graphique :

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 65

Modèle relationnel

Opérateurs spécifiques

- Jointure naturelle : $T = R \bowtie S$
ou $T = \text{JOIN}(R, S)$

Produit cartésien $R \times S$ et restriction $A = B$ sur les attributs $A \in R$ et $B \in S$.

Notation graphique :

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 66

Modèle relationnel

Exemple de requête*

Notes des étudiant-es en précisant leurs noms (et pas seulement leurs numéros)

*Requête : enchaînement d'opérations

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 67

Modèle relationnel

Exemple de requête

Décomposition des opérations

ETUDIANT		X	PASSER		
NumEtu	Nom		NumEtu	CodeEpr	Note
101	E1		101	INFO1	10
102	E2		103	INFO1	15
103	E3		103	ECO1	12

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 68

Modèle relationnel

Exemple de requête

E.NumEtu	Nom	P.NumEtu	CodeEpr	Note
101	E1	101	INFO1	10
102	E2	101	INFO1	10
103	E3	101	INFO1	10
101	E1	103	INFO1	15
102	E2	103	INFO1	15
103	E3	103	INFO1	15
101	E1	103	ECO1	12
102	E2	103	ECO1	12
103	E3	103	ECO1	12

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 69

Modèle relationnel

Exemple de requête

ETUDIANT \bowtie PASSER

E.NumEtu	Nom	P.NumEtu	CodeEpr	Note
101	E1	101	INFO1	10
103	E3	103	INFO1	15
103	E3	103	ECO1	12

$\Pi \langle \text{Nom}, \text{CodeEpr}, \text{Note} \rangle$ (ETUDIANT \bowtie PASSER)

Nom	CodeEpr	Note
E1	INFO1	10
E3	INFO1	15
E3	ECO1	12

(Projection sur les attributs Nom, CodeEpr et Note)

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 70

N° de la question : 851

Quizz

L'algèbre relationnelle sert à faire :

- Des opérations sur les relations
- Des requêtes sur une base de données relationnelle
- Des nœuds dans le cerveau des étudiant-es

Répondre sur <http://toreply.univ-lille1.fr>

Question n° 851

Bases de données <http://eric.univ-lyon2.fr/~jdamont/> 71

Classification des SGBD relationnels

- Niveau 1 : **Systèmes non relationnels**. Supportent uniquement la structure tabulaire.
- Niveau 2 : **Systèmes relationnellement minimaux**. Permettent les opérations de restriction, projection et jointure.
- Niveau 3 : **Systèmes relationnellement complets**. Toutes les opérations de l'algèbre relationnelle.
- Niveau 4 : **Systèmes relationnellement pleins**. Permettent la définition des contraintes d'intégrité.

Quizz



Quel est le niveau de classification d'un tableur ?

- A. Non relationnel – Niveau 1
- B. Relationnellement minimal – Niveau 2
- C. Relationnellement complet – Niveau 3
- D. Relationnellement plein – Niveau 4

Répondre sur <http://toreply.univ-lille1.fr>

Question n° 769

Plan du cours

- ✓ Introduction
- ✓ Modèle UML
- ✓ Modèle relationnel
- Langage SQL

Qu'est-ce que SQL ?

- Structured Query Language
- Issu de SEQUEL (Structured English as a Query Language)
- Permet la **LDD** définition, la **LMD** manipulation et le **LCD** contrôle d'une base de données relationnelle.
- SQL se base sur l'algèbre relationnelle.
- Standard depuis 1986.

Types de données principaux



- NUMBER(n) : nombre entier à n chiffres
- NUMBER(n, m) : nombre réel à n chiffres au total (virgule comprise) et m chiffres après la virgule
- VARCHAR(n) : chaîne de caractères de taille n
- DATE : date au format 'JJ-MM-AAAA'

Contraintes d'intégrité



- Mot clé **CONSTRAINT**
- Identification par un nom de contrainte
- Clé primaire : PRIMARY KEY (clé)
- Clé étrangère : FOREIGN KEY (clé) REFERENCES table(attribut)
- Contrainte de domaine : CHECK (condition)

SQL LDD

Définition des données

ex.

```
CREATE TABLE Etudiant (
    NumEtu NUMBER(8),
    Nom VARCHAR(255),
    Prenom VARCHAR(255),
    DateNaiss DATE,
    Rue VARCHAR(255),
    CP NUMBER(5),
    Ville VARCHAR(255),

    CONSTRAINT EtuClePri PRIMARY KEY (NumEtu)
)
```

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 78

SQL LDD

Définition des données

ex.

```
CREATE TABLE Passer (
    NumEtu NUMBER(8),
    CodeEpr VARCHAR(10),
    Note NUMBER(5, 2),

    CONSTRAINT PassClePri PRIMARY KEY (NumEtu, CodeEpr),

    CONSTRAINT PassCleEtrEtu FOREIGN KEY (NumEtu)
        REFERENCES Etudiant (NumEtu),

    CONSTRAINT PassCleEtrEpr FOREIGN KEY (CodeEpr)
        REFERENCES Epreuve (CodeEpr),

    CONSTRAINT NoteValide CHECK (Note >= 0 AND Note <= 20)
)
```

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 79

SQL LDD

Modifications structurelles

- Ajout d'attributs
ALTER TABLE nom_table ADD (attribut TYPE, ...)
ex. ALTER TABLE Etudiant ADD (tel NUMBER(8))
- Modifications d'attributs
ALTER TABLE nom_table MODIFY (attribut TYPE, ...)
ex. ALTER TABLE Etudiant MODIFY (tel NUMBER(10))
- Suppression d'attributs
ALTER TABLE nom_table DROP COLUMN attribut, ...
ex. ALTER TABLE Etudiant DROP COLUMN tel

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 80

SQL LDD


Modifications structurelles

- Ajout de contrainte
ALTER TABLE nom_table
ADD CONSTRAINT nom_contrainte définition_contrainte
ex. ALTER TABLE Epreuve
ADD CONSTRAINT LieuValide CHECK (Lieu IN ('Say', 'Aubrac'))
- Suppression de contrainte
ALTER TABLE nom_table DROP CONSTRAINT nom_contrainte
ex. ALTER TABLE Epreuve
DROP CONSTRAINT LieuValide

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 81

SQL LDD

Index



- Définition : Structure de données physique permettant d'accélérer les accès aux données
- Exemple :
CREATE INDEX IdxNomEtu ON Etudiant (Nom)
- NB : La clé primaire d'une relation est automatiquement indexée.

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 82

SQL LMD


Mise à jour des données

- Ajout d'un n-uplet
ex. INSERT INTO Matiere
VALUES ('BDM1MBFA', 'Bases de données')
- Modification de la valeur d'un attribut
ex. UPDATE Etudiant SET Nom='Dudule'
WHERE NumEtu = 333333
ex. UPDATE Passer SET Note = Note + 1
- Suppression de n-uplets
ex. DELETE FROM Etudiant
WHERE Ville = 'Lyon'
ex. DELETE FROM Epreuve

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 83

N° de la question : 752

Quizz



Corriger l'adresse d'un-e étudiant-e qui a déménagé est :

- Une modification structurelle des données
- Une mise à jour des données
- Ni l'une, ni l'autre

Répondre sur <http://toreply.univ-lille1.fr>

Question n° 752

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 84

SQL LMD

Interrogation des données

- Par l'exemple, sur la base ETUDIANTS

CARTE_IJLY (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)

Note : Les symboles [] indiquent une clause optionnelle d'une requête dans les transparents suivants.

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 85

SQL LMD

Étoile, tri et champs calculés

- Tous les n-uplets d'une table : étoile (*)
ex. `SELECT * FROM Etudiant`
- Tri du résultat
ex. Par ordre alphabétique [inverse] de nom
`SELECT * FROM Etudiant ORDER BY Nom [DESC]`
- Champs calculés
ex. Transformation de notes sur 20 en notes sur 10
`SELECT Note / 2 FROM Passer`

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 86

SQL LMD

Projection et restriction

- Projection
ex. Noms et Prénoms des étudiant-es, uniquement (pas les autres attributs)
`SELECT Nom, Prénom FROM Etudiant`
- Suppression des doublons
ex. `SELECT DISTINCT Nom FROM Etudiant`
- Restriction
ex. Étudiant-es qui habitent à Lyon
`SELECT * FROM Etudiant WHERE Ville = 'Lyon'`

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 87

SQL LMD

Opérateurs de restriction (1/3)

ex. Épreuves se déroulant après le 01/01/2016
`SELECT * FROM Epreuve WHERE DateEpr >= '01-01-2016'`

ex. Notes comprises entre 10 et 20
`SELECT * FROM Passer WHERE Note BETWEEN 10 AND 20`

ex. Notes indéterminées (sans valeur)
`SELECT * FROM Passer WHERE Note IS NULL`

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 88

SQL LMD

Opérateurs de restriction (2/3)

ex. Étudiant-es habitant une ville dont le nom se termine par sur-Saône
`SELECT * FROM Etudiant WHERE Ville LIKE '%sur-Saône'`

'sur-Saône%' ⇒ commence par sur-Saône
 '%sur%' ⇒ contient le mot sur

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 89

Opérateurs de restriction (3/3)

ex. Prénoms des étudiant-es dont le nom est Dupont, Durand ou Martin

```
SELECT Prénom FROM Etudiant
WHERE Nom IN ('Dupont', 'Durand', 'Martin')
```

NB : Possibilité d'utiliser la négation pour tous ces prédicats ⇒ NOT BETWEEN, NOT NULL, NOT LIKE, NOT IN.

Opérateurs logiques

- **ET.** **ex.** Épreuves se déroulant le 15/01/2016 en salle D201

```
SELECT * FROM Epreuve
WHERE DateEpr = '15-01-2016' AND Lieu = 'D201'
```
- **OU.** **ex.** Étudiant-es né-es avant 1990 ou habitant hors Lyon

```
SELECT * FROM Etudiant
WHERE DateNaiss < '01-01-1990' OR Ville <> 'Lyon'
```
- **Combinaisons.** **ex.** Étudiant-es né-es après 1990 et habitant Lyon ou Vienne

```
SELECT * FROM Etudiant
WHERE DateNaiss > '31-12-1990'
AND (Ville = 'Lyon' OR Ville = 'Vienne')
```

Fonctions d'agrégat

- Elles opèrent sur un ensemble de valeurs et les agrègent.
- **AVG()**, **VARIANCE()**, **STDDEV()** : moyenne, variance et écart-type des valeurs
- **SUM()** : somme des valeurs
- **MIN()**, **MAX()** : valeur minimum, valeur maximum
- **COUNT()** : nombre de valeurs
- **ex.** Moyenne des notes

```
SELECT AVG(Note) FROM Passer
```

Fonction COUNT et opérateur DISTINCT

ex. Nombre total de notes

```
SELECT COUNT(*) FROM Passer
SELECT COUNT(NumEtu) FROM Passer
```

ex. Nombre d'étudiant-es noté-es

```
SELECT COUNT(DISTINCT NumEtu) FROM Passer
```

Exemple

Table PASSER

NumEtu	CodeEpr	Note
101	INFO1	10
103	INFO1	15
103	ECO1	12

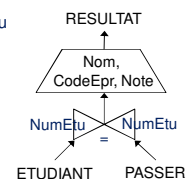
COUNT(NumEtu) ⇒ Résultat = 3

COUNT(DISTINCT NumEtu) ⇒ Résultat = 2

Jointure

ex. Liste des notes avec le nom des étudiant-es

```
SELECT Nom, CodeEpr, Note
FROM Etudiant, Passer
WHERE Etudiant.NumEtu = Passer.NumEtu
```



Jointure

ex. Idem avec le numéro d'étudiant en plus

```
SELECT E.NumEtu, Nom, CodeEpr, Note
FROM Etudiant E, Passer P
WHERE E.NumEtu = P.NumEtu
ORDER BY Nom, Note DESC
```

NB : Utilisation d'alias (E et P) pour alléger l'écriture
+ tri par nom (croissant) et note (décroissante).

Jointure

Jointure exprimée avec le prédicat IN

ex. Notes des épreuves passées le 23 septembre 2013

```
SELECT Note FROM Passer
WHERE CodeEpr IN (
  SELECT CodeEpr FROM Epreuve
  WHERE DateEpr = '23-09-2013')
```

Sous-
requête

NB : Il est possible d'imbriquer des requêtes.

Prédicats d'existence

■ Prédicats EXISTS / NOT EXISTS

ex. Étudiant-es qui ont passé au moins une épreuve
[n'ont passé aucune épreuve]

```
SELECT * FROM Etudiant E
WHERE [NOT] EXISTS (
  SELECT * FROM Passer P
  WHERE E.NumEtu = P.NumEtu)
```

Prédicats de dénombrement

■ Prédicats ALL / ANY

ex. Numéros des étudiant-es qui ont obtenu au moins
une note supérieure à chacune [à au moins une] des
notes obtenues par l'étudiant-e n° 1000.

```
SELECT DISTINCT NumEtu FROM Passer
WHERE Note > ALL [ANY] (
  SELECT Note FROM Passer
  WHERE NumEtu = 101 )
```

Groupement (1/2)

ex. Moyenne de chaque étudiant-e

```
SELECT NumEtu, AVG(Note)
FROM Passer
GROUP BY NumEtu
```

ex. Nombre de notes par étudiant-e

```
SELECT NumEtu, COUNT(*)
FROM Passer
GROUP BY NumEtu
```

Groupement (2/2)

ex. Note moyenne pour les étudiant-es ayant
passé moins de 5 épreuves

```
SELECT NumEtu, AVG(Note)
FROM Passer
GROUP BY NumEtu
HAVING COUNT(*) < 5
```

Attention : La clause HAVING ne s'utilise qu'avec GROUP BY.

NB : HAVING : évaluation de condition sur un résultat
de groupement (*a posteriori*)
≠ WHERE : évaluation de condition *a priori*

SQL LMD

Opérations ensemblistes

INTERSECT, MINUS, UNION


ex. Code des épreuves ayant soit lieu dans l'Amphi Aubrac, soit ayant été passées par l'étudiant-e n° 102

```
SELECT CodeEpr FROM Epreuve
WHERE Lieu = 'Amphi Aubrac'
UNION
SELECT CodeEpr FROM Passer
WHERE NumEtu = 102
```

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 102

N° de la question : 312

Quizz




Tous les opérateurs de l'algèbre relationnelle ont-ils un équivalent en SQL ?

Répondre sur <http://toreply.univ-lille1.fr>
Question n° 312

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 103

Tutoriel SQL

Pour approfondir SQL en ligne...




<http://eric.univ-lyon2.fr/~jdarmon/tutoriel-sql/>

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 104

Plan du cours

- ✓ Introduction
- ✓ Modèle UM
- ✓ Modèle relationnel
- ✓ Langage SQL



Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 105

N° de la question : 531

Sondage express



Que pensez vous de ce cours ?

Répondre sur <http://toreply.univ-lille1.fr>
Question n° 531

Bases de données <http://eric.univ-lyon2.fr/~jdarmon/> 106