

# Théorie relationnelle

Frédéric DADEAU

Institut FEMTO-ST – Département d’Informatique des Systèmes Complexes

Bureau 410 C

Email : [frederic.dadeau@univ-fcomte.fr](mailto:frederic.dadeau@univ-fcomte.fr)

Licence 1 – Semestre 2



# Objectifs de ce chapitre

---

Être capable d'appréhender le fonctionnement d'une base de données relationnelle en identifiant les entités, leurs liens et leurs propriétés.

# Plan du cours

---

1. Définitions de base
2. Intégrité des données
3. Un exemple
4. Les opérateurs relationnels

# La théorie relationnelle

## Historique

La théorie relationnelle a été proposée pour la première fois par Edgar Frank Codd, dans l'article

*A relational Model of Data for Large Shared Data Banks*, CACM, Juin 1970

Depuis, aucune modification révolutionnaire.

Il s'agit d'une approche basée sur la *théorie des ensembles* (mathématiques).

# Plan du cours

---

## 1. Définitions de base

Définitions sur les relations

Propriétés des relations

Propriétés des schémas relationnels

## 2. Intégrité des données

## 3. Un exemple

## 4. Les opérateurs relationnels

# Les relations

## Définition des relations

Une relation  $R$ , sur un ensemble de domaines  $D_1, D_2, \dots, D_N$ , est constituée de deux parties :

- ▶ l'en-tête : ensemble fixé d'attributs
- ▶ le corps : ensemble de t-uplets

Une relation peut être assimilée à un tableau.

## Attributs et T-uplets

Un **attribut** correspond à une colonne. Un **t-uplet** correspond à une ligne.

# La théorie relationnelle

Un petit dessin vaut mieux qu'un long discours :

Nom attribut 1	Nom attribut 2	Nom attribut 3	...	Nom attribut $N$
		$v \in D_3$		

Les attributs sont en bleu, les t-uplets sont en rouge.

On trouve des valeurs  $D_3$  désigne le domaine de l'attribut 3.

# Les relations

## Domaines de valeurs

Un domaine  $D$  est un ensemble de valeurs dans lequel des attributs puisent leurs valeurs.

# Les relations

## Domaines de valeurs

Un domaine  $D$  est un ensemble de valeurs dans lequel des attributs puisent leurs valeurs.

## Cardinalité d'une relation

La cardinalité d'une relation correspond à son nombre de t-uplets (i.e. nombre de lignes).

# Les relations

## Domaines de valeurs

Un domaine  $D$  est un ensemble de valeurs dans lequel des attributs puisent leurs valeurs.

## Cardinalité d'une relation

La cardinalité d'une relation correspond à son nombre de t-uplets (i.e. nombre de lignes).

## Degré d'une relation

Le degré d'une relation correspond à son nombre d'attributs (i.e. nombre de colonnes).

# Les relations

## Clé primaire

La *clé primaire* est un groupe d'attribut dont la valeur permet d'identifier de manière unique un t-uplet (i.e. une ligne) de la relation.

# Les relations

## Clé primaire

La *clé primaire* est un groupe d'attribut dont la valeur permet d'identifier de manière unique un t-uplet (i.e. une ligne) de la relation.

## Schéma (de base de données) relationnel(le)

Le *schéma de base de données relationnelle* ou *schéma relationnel* est un ensemble de relations, qui ont toutes des noms différents.

# Un exemple de relation

## La relation ETUDIANT

Les attributs de la relation ETUDIANT sont les suivants :

- ▶ NumeroEtudiant : entier positif
- ▶ NomEtudiant : chaîne de caractères
- ▶ PrenomEtudiant : chaîne de caractères
- ▶ DateNaissanceEtudiant : date
- ▶ LoginEtudiant : chaîne de 8 caractères maxi
- ▶ MotDePasseEtudiant : chaînes de 8 caractères maxi

On dénote cette relation **ETUDIANT**(NumeroEtudiant\*, NomEtudiant, PrenomEtudiant, DateNaissanceEtudiant, LoginEtudiant, MotDePasseEtudiant).

# Un exemple de relation

## La relation ETUDIANT – Contenu

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

- ▶ L'en-tête correspond à la ligne de titre,
- ▶ Degré de la relation ETUDIANT : 6
- ▶ Cardinalité de la relation ETUDIANT : 4
- ▶ Clé primaire : NumeroEtudiant

# Les propriétés des relations

Définition mathématique de la notion de relation

- ▶ Pas de duplication des t-uplets
- ▶ Pas d'ordonnancement des t-uplets
- ▶ Pas d'ordonnancement des attributs
- ▶ Atomicité des valeurs des attributs
- ▶ Pour un attribut donné, toutes les valeurs de cet attribut appartiennent au même domaine

# Les propriétés des relations

## Pas de duplication des t-uplets

- ▶ Ensemble de t-uplets d'une relation = Ensemble au sens des mathématiques
- ▶ Une même valeur ne peut pas apparaître deux fois dans un ensemble

En conséquence, toute relation possède une clé.

## La relation ETUDIANT erronée

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

Cette relation ne respecte pas la propriété d'absence de doublons dans les t-uplets.

# Les propriétés des relations

## Pas d'ordre des t-uplets

- ▶ Ensemble de t-uplets d'une relation = Ensemble au sens des mathématiques
- ▶ Les ensembles ne sont pas ordonnés

En conséquence, il n'existe pas de 2<sup>ème</sup> t-plet, pas plus que de t-plet suivant.

## La relation ETUDIANT

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski



Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard

# Les propriétés des relations

## Pas d'ordre des attributs

- ▶ En-tête d'une relation = Ensemble d'attributs
- ▶ Comme pour les t-uplets, les attributs ne sont pas ordonnés.

En conséquence, les attributs sont référencés par un nom et pas par une position dans l'en-tête.

## La relation ETUDIANT

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
---------	-----	--------	---------------	-------	------------



Login	Prenom	MotDePasse	Numero*	Nom	DateNaissance
-------	--------	------------	---------	-----	---------------

# Les propriétés des relations

## Atomicité des valeurs des attributs

- ▶ Les domaines contiennent des valeurs atomiques
- ▶ Dans une relation, à l'intersection d'une ligne et d'une colonne, il ne peut y avoir qu'une seule valeur.

On parle de relation normalisée, en 1<sup>ère</sup> forme normale.

## La relation ETUDIANT erronée

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998 30-05-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	mssmolinski 1234

Cette relation ne respecte pas la propriété d'atomicité des valeurs.

# Les propriétés des relations

## Domaines uniques

- ▶ Dans une relation, les valeurs possibles des attributs sont données par son domaine.
- ▶ Il n'est pas possible d'associer plusieurs domaines à un attribut (pas d'union de type).

# Les propriétés des schémas relationnels

## Noms uniques

- ▶ Un schéma de base de données est un ensemble de relations
  - ⇒ Dans un schéma relationnel, chaque relation a un nom unique.
- ▶ Une relation est un ensemble d'attributs
  - ⇒ Dans une relation, chaque attribut a un nom unique.

Justification : il n'est pas possible d'avoir deux fois le même nom dans un ensemble  
→ référencement ambigu

# Les propriétés des schémas relationnels

## Convention de représentation

On précise toujours le nom et les attributs de la relation. Le contenu de la relation (les-tuplets) sont toujours donnés en extension (leurs valeurs sont énumérées).

## Lien avec les bases de données relationnelles

Base de données relationnelle



Ensemble de relations normalisées

# Plan du cours

---

## 1. Définitions de base

## 2. Intégrité des données

Notions préliminaires : clé, lien, NULL

Les règles d'intégrité : unicité de la clé

Les règles d'intégrité : contraintes de domaine

Les règles d'intégrité : contraintes de références

Les règles d'intégrité : intégrité des entités

## 3. Un exemple

## 4. Les opérateurs relationnels

# L'intégrité des données

## L'intégrité des données

Bases pour définir des outils permettant une gestion automatique au niveau du SGBD de certaines propriétés de la base de données.

## Intégrité des données relationnelles

- ▶ Une base de données contient une configuration particulière des valeurs de données.
- ▶ Certaines configurations de valeurs n'ont pas de sens  
⇒ Définition d'une base de données étendue pour inclure certaines *contraintes d'intégrité*.

# L'intégrité des données

## L'intégrité des données

Bases pour définir des outils permettant une gestion automatique au niveau du SGBD de certaines propriétés de la base de données.

Les *contraintes d'intégrité* sont des règles qui vont permettre au SGBD de **conserver automatiquement la cohérence de la base de données**.

# L'intégrité des données

## Comment s'assurer de l'intégrité des données

- ▶ en vérifiant les données lors de leur chargement,
- ▶ en vérifiant les données lors de toute modification (saisie, mise à jour)
- ▶ en répercutant certaines mises à jour entre les tables
- ▶ en gérant les références entre les tables

## Les règles d'intégrité que nous allons voir

- ▶ l'unicité de la clé
- ▶ les contraintes de domaines
- ▶ les contraintes de références
- ▶ la règle d'intégrité des entités

# Les clés relationnelles

---

Une **clé** identifie un t-uplet.

Nous allons voir les notions de :

- ▶ Super-clé
- ▶ Clé candidate
- ▶ Clé primaire
- ▶ Clé étrangère

# Les clés relationnelles

## Définition d'une super-clé

Une **super-clé** pour une relation  $R$  est un sous-ensemble de l'ensemble des attributs de  $R$  qui identifie de manière unique chaque t-uplet de  $R$ .

## Propriété d'unicité de la super-clé

Une super-clé possède la propriété d'**unicité** : il n'existe pas deux t-uplets distincts de  $R$  ayant la même valeur pour cet ensemble d'attributs.

# Les clés relationnelles

## Définition d'une clé candidate

Une **clé candidate** pour une relation  $R$  est une super-clé qui possède la propriété d'être *irréductible*.

## Propriété d'irréductibilité (des clés candidates)

Une clé candidate composée d'un ensemble d'attributs  $K$  est dite **irréductible** si aucun sous-ensemble strict de  $K$  n'est une super-clé.

# Les clés relationnelles

## Utilisation des clés candidates

Les **clés candidates** fournissent un mécanisme d'adressage dans la relation car elles permettent de répéter **un** t-uplet dans la relation.

## Référencement par clé-candidate

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinksi	msmolinksi

Données correspondant à l'étudiant numéro 23794 → 1 seul t-uplet.

Données correspondant au prénom d'étudiant 'Simon' → ? ? ? t-uplets.

# Les clés relationnelles

## Définition d'une clé primaire

Une **clé primaire** est une clé choisie arbitrairement parmi les clés candidates.

## Unicité de la clé primaire dans une relation

S'il est possible de définir plusieurs clé candidates pour une relation, il n'existe en revanche qu'une seule clé primaire.

⇒ une clé primaire est donc unique pour une relation donnée.

# Les clés relationnelles

## La relation ETUDIANT

On considère la relation ETUDIANT :

ETUDIANT (NumeroEtudiant, NomEtudiant, PrenomEtudiant, DateNaissanceEtudiant,  
LoginEtudiant, MotDePasseEtudiant)

Le couple d'attributs (NumeroEtudiant, NomEtudiant) représente une super-clé, car deux étudiants distincts ne peuvent pas avoir le même numéro et le même nom.

# Les clés relationnelles

## La relation ETUDIANT

On considère la relation ETUDIANT :

ETUDIANT (NumeroEtudiant, NomEtudiant, PrenomEtudiant, DateNaissanceEtudiant,  
LoginEtudiant, MotDePasseEtudiant)

L'attribut NumeroEtudiant est une clé candidate, car deux étudiants ne peuvent pas avoir le même numéro.

La clé est réduite à un seul attribut, elle est donc *irréductible*.

# Les clés relationnelles

## La relation ETUDIANT

On considère la relation ETUDIANT :

ETUDIANT (NumeroEtudiant, NomEtudiant, PrenomEtudiant, DateNaissanceEtudiant,  
LoginEtudiant, MotDePasseEtudiant)

Le login d'un étudiant étant également unique, LoginEtudiant est également clé candidate.

# Les clés relationnelles

## La relation ETUDIANT

On considère la relation ETUDIANT :

ETUDIANT (NumeroEtudiant, NomEtudiant, PrenomEtudiant, DateNaissanceEtudiant,  
LoginEtudiant, MotDePasseEtudiant)

Si on suppose qu'il n'existe pas deux étudiants avec les mêmes noms, prénoms et dates de naissance, alors le triplet (NomEtudiant, PrenomEtudiant, DateNaissanceEtudiant) est également une clé candidate.

# Les clés relationnelles

Certains attributs (ou ensembles d'attributs) peuvent apparaître dans plusieurs relations.

⇒ On identifie alors des liaisons (ou liens) entre plusieurs relations.

## Définition d'une clé étrangère

Une clé étrangère est un ensemble d'attributs d'un relation  $R'$  qui est clé candidate dans une relation  $R$ .

# Les clés relationnelles

Certains attributs (ou ensembles d'attributs) peuvent apparaître dans plusieurs relations.

⇒ On identifie alors des liaisons (ou liens) entre plusieurs relations.

## Propriété d'une clé étrangère

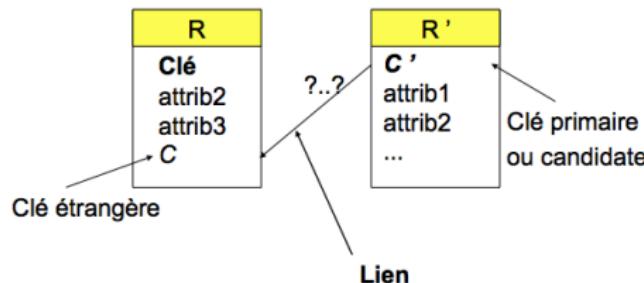
Soit  $R$  une relation, une **clé étrangère** dans  $R$  est un sous-ensemble  $C$  de l'ensemble des attributs de  $R$  tel que :

- ▶ Il existe une relation  $R'$  avec une clé candidate  $C'$ , et
- ▶ Chaque valeur de  $C$  dans  $R$  est identique à la valeur de  $C'$  dans  $R'$ .

⇒ On définit ainsi un lien entre les relations  $R'$  et  $R$ .

# Les liens

En notation graphique :



# Les liens

## Types de lien

Différents types de lien existent selon que le nombre de fois où la valeur de la clé candidate peut apparaître en tant que clé étrangère dans la relation cible.

- 1..N une valeur de la clé candidate peut apparaître **un nombre indéterminé de fois** (potentiellement nul) en tant que clé étrangère.
- 1..1 une valeur de la clé candidate ne peut apparaître qu'**une et une seule fois** en tant que clé étrangère.

Remarque : les liens N..N n'existent pas !

## Sens d'un lien

Un lien est toujours défini d'une **clé primaire** vers une **clé étrangère**.

Cela se comprend comme : “*J'ai une valeur dans une relation (clé primaire), elle peut être référencée dans une autre table (clé étrangère).*”

# Les liens

## 1. Repérer les liens potentiels

Il faut chercher dans les relations si une clé candidate d'une relation se retrouve comme clé étrangère dans une autre relation. On cherche donc des groupes attributs (ou ensembles d'attributs) "compatibles" d'une relation à l'autre et représentant la même information.

ETUDIANT
<b>NumeroEtudiant</b>
NomEtudiant
PrenomEtudiant
DateNaissanceEtudiant
LoginEtudiant
MotDePasseEtudiant

INSCRIPTION
<b>NumeroEtudiant</b>
<b>CodeModule</b>

# Les liens

## 1. Repérer les liens potentiels

Il faut chercher dans les relations si une clé candidate d'une relation se retrouve comme clé étrangère dans une autre relation. On cherche donc des groupes attributs (ou ensembles d'attributs) "compatibles" d'une relation à l'autre et représentant la même information.

ETUDIANT
<b>NumeroEtudiant</b>
NomEtudiant
PrenomEtudiant
DateNaissanceEtudiant
LoginEtudiant
MotDePasseEtudiant

INSCRIPTION
<b>NumeroEtudiant</b>
CodeModule

# Les liens

## 2. Définir le sens et le type du lien

Le sens est toujours le même : de la clé primaire vers la clé étrangère.

Le type de la relation dépend du nombre d'apparition d'une même valeur des attributs représentant la clé étrangère :

- ▶ Plusieurs occurrences possibles : lien de type 1..N
- ▶ Une seule occurrence possible : lien de type 1..1

Autrement dit : si la clé étrangère est une clé candidate dans sa relation, on a un lien 1..1.

ETUDIANT
<b>NumeroEtudiant</b>
NomEtudiant
PrenomEtudiant
DateNaissanceEtudiant
LoginEtudiant
MotDePasseEtudiant

INSCRIPTION
<b>NumeroEtudiant</b>
CodeModule

# Les liens

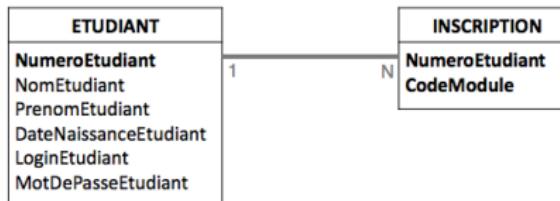
## 2. Définir le sens et le type du lien

Le sens est toujours le même : de la clé primaire vers la clé étrangère.

Le type de la relation dépend du nombre d'apparition d'une même valeur des attributs représentant la clé étrangère :

- ▶ Plusieurs occurrences possibles : lien de type 1..N
- ▶ Une seule occurrence possible : lien de type 1..1

Autrement dit : si la clé étrangère est une clé candidate dans sa relation, on a un lien 1..1.



# La valeur NULL

## Qu'est-ce que c'est ?

Il s'agit d'une valeur conventionnelle, introduite dans une relation lorsque la valeur de l'un des attributs est inconnue, inapplicable ou non-spécifiée.  
⇒ NULL représente l'absence de valeur

NULL est une valeur "classique" en programmation.

# La valeur NULL

## La valeur NULL chez les attributs

Cette valeur d'attribut s'autorise ou s'interdit au moment de la conception de la base de données. Un SGBD pourra donc affecter cette valeur à un attribut, si celui-ci n'est pas renseigné lors d'un ajout ou d'une mise à jour.

## Deux politiques

Si, lors d'un ajout ou d'une mise à jour d'un t-uplet, l'utilisateur ne précise pas une valeur pour un attribut, on a deux possibilités :

- ▶ si NULL est autorisé pour l'attribut, le SGBD lui affectera la valeur NULL,
- ▶ si NULL n'est pas autorisé pour l'attribut, le SGBD rejetttera tout t-uplet où la valeur de l'attribut n'est pas renseigné.

# La valeur NULL

## La relation ETUDIANT – Contenu

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

Supposons que la valeur NULL soit autorisée pour l'attribut *DateNaissanceEtudiant*.

L'ajout d'un t-uplet (32823, Marche, Claire, NULL, cmarche, cmarche) est possible.

# La valeur NULL

## La relation ETUDIANT – Contenu

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

Supposons que la valeur NULL soit interdite pour l'attribut *MotDePasseEtudiant*.

L'ajout d'un t-uplet (32823, Marche, Claire, 30-07-1999, cmarche, NULL) est impossible.

# Les règles d'intégrité

---

Elles sont au nombre de 4 :

1. L'unicité de la clé
2. Les contraintes de domaine
3. Les contraintes de référence
4. La règle d'intégrité des entités

# Les règles d'intégrité

Cette première règle est basée sur les notions de clé candidate et de clé primaire.

## Règle de l'unicité de la clé

Dans une base de données, toutes les relations doivent posséder une clé unique, appelée **clé primaire**.

Cette règle permet de mettre en place des mécanismes de contrôle, qui vérifient que deux informations identiques ne peuvent pas être présentes dans la base de données.

# Les règles d'intégrité

## Unicité de la clé et contrôle

Considérons la relation ETUDIANT ci-dessous

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

L'ajout d'un t-uplet (23794, Marche, Claire, 30-07-1999, cmarche, cmarche) est impossible, car ce t-uplet duplique une des valeurs de la clé primaire.

# Les règles d'intégrité

Chaque attribut d'une relation prend ses valeurs dans un domaine (i.e. un ensemble de valeurs).

Les domaines sont des types de base, tels que les entiers, les réels, les chaînes de caractères, agrémentés de quelques types dédiés, tels que les dates, les horaires, ou les types énumérés (principalement).

## Les contraintes de domaine

Une contrainte de domaine est une contrainte d'intégrité qui définit la propriété que doit vérifier toute valeur d'un attribut d'une relation donnée.

Il s'agit de vérifier que la valeur est correctement **typée**.

# Les règles d'intégrité

## Illustration des contraintes de domaines

Considérons l'attribut *DateNaissanceEtudiant*. Son domaine est celui donné par le type DATE.

Lors d'un ajout ou d'une mise à jour de cet attribut, le SGBD s'assurera que la nouvelle valeur appartient bien au domaine défini, pour vérifier la cohérence des données.

# Les règles d'intégrité

## Qu'est-ce que c'est ?

Les contraintes de références permettent au SGBD de gérer automatiquement la présence de **données référencées** dans les différentes relations de la base.

## Intégrité référentielle

La base de données ne doit pas contenir de valeurs de clés étrangères non unifiables.  
⇒ Si une clé étrangère prend une valeur, ce doit être une valeur qui existe dans la relation référencée par le lien.

# Les règles d'intégrité

## Relations ETUDIANT et INSCRIPTION

On considère la relation INSCRIPTION(*NumeroEtudiant\**, *CodeModule\**).

- ▶ L'attribut *NumeroEtudiant* fait référence à l'étudiant (il est lié à l'attribut *ETUDIANT.NumeroEtudiant*) dans lequel le cours où il est inscrit est identifié par son *CodeModule*.
- ▶ Le t-uplet (*23794, BD\_L1*) signifie que l'étudiant numéro 23794 est inscrit dans le module BD\_L1.
- ▶ La valeur prise par l'attribut *INSCRIPTION.NumeroEtudiant* ne peut être qu'une valeur apparaissant dans la relation *ETUDIANT*, pour l'attribut *NumeroEtudiant*.
- ▶ Il ne serait pas cohérent de décrire les inscriptions d'un étudiant qui n'existe pas.

# Les règles d'intégrité

Prise en compte de l'intégrité référentielle lors d'une saisie.

- ▶ Le SGBD effectue un contrôle qui permet de savoir si la valeur existe dans la relation référencée.
- ▶ C'est un contrôle immédiat qui empêche l'ajout en cas d'absence de valeur unifiable.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Politique RESTRICTED

L'opération de mise à jour ou de suppression est restreinte au cas où il n'existe aucune référence à la valeur dans les autres relations en lien avec la relation initiale. Dans les autres cas, elle est interdite.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Application d'une politique RESTRICTED

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1

On veut supprimer le t-uplet correspondant à l'étudiant 23794.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Application d'une politique RESTRICTED

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1

On veut supprimer le t-uplet correspondant à l'étudiant 23794.

Avec la règle RESTRICTED sur le lien entre ETUDIANT et INSCRIPTION, la suppression est impossible, car il existe des enregistrements dans INSCRIPTION référençant cet étudiant.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Application d'une politique RESTRICTED

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1

On veut supprimer le t-uplet correspondant à l'étudiant 33818.

Avec la règle RESTRICTED sur le lien entre ETUDIANT et INSCRIPTION, la suppression est possible, car il n'existe pas d'enregistrements dans INSCRIPTION référençant cet étudiant.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Politique CASCADE

L'opération de mise à jour ou de suppression est réalisée en "cascade" dans les autres relations, c'est-à-dire qu'elle est propagée.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Application d'une politique CASCADE

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1

On veut supprimer le t-uplet correspondant à l'étudiant numéro 32794.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Application d'une politique CASCADE

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1

On veut supprimer le t-uplet correspondant à l'étudiant numéro 23794.

Avec la règle CASCADE sur le lien entre ETUDIANT et INSCRIPTION, la suppression est possible, même s'il existe des enregistrements dans INSCRIPTION référençant cet étudiant. Ces enregistrements seront d'ailleurs également supprimés.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Application d'une politique CASCADE

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
32911	BD_L1

La suppression a lieu dans les deux tables.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Une petite remarque

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1

Supposons que l'on souhaite supprimer les deux enregistrements de INSCRIPTION qui concernent l'étudiant 23794.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Une petite remarque

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
32911	BD_L1

C'est toujours possible, quelque soit la politique de mise à jour/suppression définie.

En effet, les règles de suppression “restreinte” ou “en cascade” ne s’appliquent que dans un seul sens (la suppression d’une clé primaire entraîne/est bloquée par la clé étrangère, et pas l’inverse !).

→ La relation ETUDIANT est inchangée !

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Politique SET NULL

L'opération de mise à jour ou de suppression de la clé candidate est répercutée en mettant les valeurs des clés étrangères à NULL.

Attention : la valeur NULL doit être autorisée pour ces attributs clé étrangère.

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Politique SET DEFAULT

L'opération de mise à jour ou de suppression de la clé candidate est répercutée en mettant les valeurs des clés étrangères à une valeur par défaut (si celle-ci existe).

# Les règles d'intégrité

En cas de modification ou de suppression de données, plusieurs politiques de mise à jour/suppression existent. Ces politiques sont appliquées aux liens entre les relations.

## Politique NO ACTION

Similaire à RESTRICTED, sauf que la vérification se fait a posteriori de l'exécution de la requête.

Cela permet à des triggers (procédures qui peuvent s'exécuter automatiquement lors de certaines actions dans la base) de s'exécuter pour corriger d'éventuelles incohérences ainsi créées.

# Les règles d'intégrité

## Règle d'intégrité des données

Tous les attributs appartenant à la clé primaire d'une relation ne sont pas autorisés à prendre la valeur NULL.

### La relation ETUDIANT

Dans la relation ETUDIANT, l'attribut NumeroEtudiant est clé primaire, il ne peut donc pas prendre la valeur NULL.

Un t-uplet (NULL, Marche, Claire, 30-07-1999, cmarche, NULL) n'est donc pas autorisé dans ETUDIANT.

# Les règles d'intégrité

## Règle d'intégrité des données

Tous les attributs appartenant à la clé primaire d'une relation ne sont pas autorisés à prendre la valeur NULL.

### La relation ETUDIANT

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1

On veut supprimer le t-uplet correspondant à l'étudiant numéro 23794 pour lequel on a une politique SET NULL entre ETUDIANT et INSCRIPTION.

# Les règles d'intégrité

## Règle d'intégrité des données

Tous les attributs appartenant à la clé primaire d'une relation ne sont pas autorisés à prendre la valeur NULL.

## Application d'une politique CASCADE

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

NumeroEtudiant*	CodeModule*
NULL	BD_L1
NULL	PROG_L1
32911	BD_L1

On veut supprimer le t-uplet correspondant à l'étudiant numéro 23794.

La suppression mettrait à NULL les valeurs du NumeroEtudiant de INSCRIPTION pour les deux enregistrements. Or NumeroEtudiant est une partie de la clé primaire de INSCRIPTION, et ne peut donc pas être NULL.

La suppression sera alors refusée.

# Les règles d'intégrité

## Les clés étrangères et NULL

L'intégrité référentielle impose que toute valeur d'une clé étrangère contienne une valeur unifiable.

- ▶ On étend cette définition pour prendre en compte la valeur NULL, unifiable avec toutes les valeurs.
- ▶ Le concepteur de la base de données peut préciser si, oui ou non, une clé étrangère est autorisée à prendre la valeur NULL.  
Attention : si cette clé étrangère est également clé primaire, elle ne peut pas prendre la valeur NULL (cf. NumeroEtudiant dans INSCRIPTION).

# Plan du cours

---

1. Définitions de base
2. Intégrité des données
3. Un exemple
4. Les opérateurs relationnels

# Un exemple

---

Considérons l'ensemble de relations suivant :

- ▶ **ETUDIANT**(NumeroEtudiant\*, NomEtudiant, PrenomEtudiant, DateNaissanceEtudiant, LoginEtudiant, MotDePasseEtudiant)
- ▶ **INSCRIPTION**(NumeroEtudiant\*, CodeModule\*, Annee\*, NoteSession1, NoteSession2)
- ▶ **MODULE**(CodeModule\*, LibelleModule, VolumeCM, VolumeTD, VolumeTP, Responsable)
- ▶ **ENSEIGNANT**(NumeroHarpege\*, NomEnseignant, PrenomEnseignant, Grade)

qui représente une version simplifiée des inscriptions d'étudiants à des modules de l'université.

# Un exemple

Détaillons chacune des relations.

## La relation ETUDIANT

Les attributs de la relation ETUDIANT sont les suivants :

- ▶ NumeroEtudiant\* : entier positif
- ▶ NomEtudiant : chaîne de caractères
- ▶ PrenomEtudiant : chaîne de caractères
- ▶ DateNaissanceEtudiant : date
- ▶ LoginEtudiant : chaîne de 8 caractères maxi
- ▶ MotDePasseEtudiant : chaînes de 8 caractères maxi

Remarque : LoginEtudiant est clé candidate de la relation.

# Un exemple

Détaillons chacune des relations.

## La relation INSCRIPTION

Les attributs de la relation INSCRIPTION sont les suivants :

- ▶ NumeroEtudiant\* : entier positif
- ▶ CodeModule\* : chaîne de 8 caractères maxi
- ▶ Année\* : entier positif sur 4 chiffres
- ▶ NoteSession1 : réel entre 0 et 20
- ▶ NoteSession2 : réel entre 0 et 20, NULL autorisé.

# Un exemple

Détaillons chacune des relations.

## La relation MODULE

Les attributs de la relation MODULE sont les suivants :

- ▶ CodeModule\* : chaîne de 8 caractères maxi
- ▶ LibelleModule : chaîne de caractères
- ▶ VolumeCM : entier positif ou nul
- ▶ VolumeTD : entier positif ou nul
- ▶ VolumeTP : entier positif ou nul
- ▶ Responsable : entier positif

# Un exemple

Détaillons chacune des relations.

## La relation ENSEIGNANT

Les attributs de la relation ENSEIGNANT sont les suivants :

- ▶ NumeroHarpege\* : entier positif à 5 chiffres
- ▶ NomEnseignant : chaîne de caractères
- ▶ PrenomEnseignant : chaîne de caractères
- ▶ Grade : "MCF", "PR", "PRAG", "Vacataire", "Doctorant"

# Un exemple

Essayons de trouver les liens.

ETUDIANT
<b>NumeroEtudiant</b>
NomEtudiant
PrenomEtudiant
DateNaissanceEtudiant
LoginEtudiant
MotDePasseEtudiant

INSCRIPTION
<b>NumeroEtudiant</b>
<b>CodeModule</b>
<b>Annee</b>
NoteSession1
NoteSession2

MODULE
<b>CodeModule</b>
LibelleModule
VolumeCM
VolumeTD
VolumeTP
Responsable

ENSEIGNANT
<b>NumerоХарпеге</b>
NomEnseignant
PrenomEnseignant
Grade

Méthodologie :

1. repérer les attributs compatibles (domaine + sémantique)
2. définir le sens et le type du lien (1..N ou 1..1)

# Un exemple

Essayons de trouver les liens.

ETUDIANT
<b>NumeroEtudiant</b>
NomEtudiant
PrenomEtudiant
DateNaissanceEtudiant
LoginEtudiant
MotDePasseEtudiant

INSCRIPTION
<b>NumeroEtudiant</b>
CodeModule
Annee
NoteSession1
NoteSession2

MODULE
CodeModule
LibelleModule
VolumeCM
VolumeTD
VolumeTP
Responsable

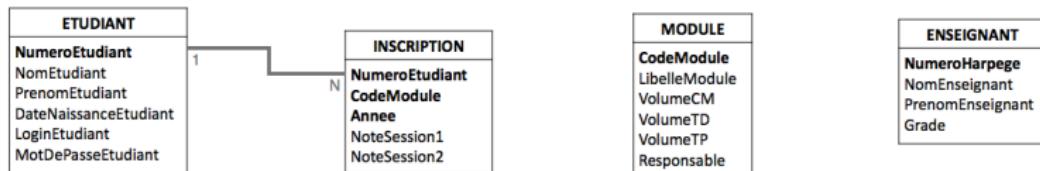
ENSEIGNANT
<b>NumeroHarpege</b>
NomEnseignant
PrenomEnseignant
Grade

Méthodologie :

1. repérer les attributs compatibles (domaine + sémantique)
2. définir le sens et le type du lien (1..N ou 1..1)

# Un exemple

Essayons de trouver les liens.

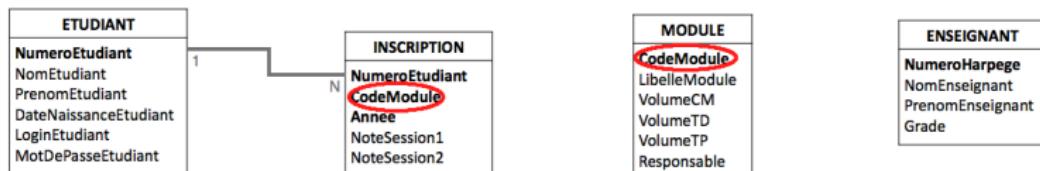


Méthodologie :

1. repérer les attributs compatibles (domaine + sémantique)
2. définir le sens et le type du lien (1..N ou 1..1)

# Un exemple

Essayons de trouver les liens.

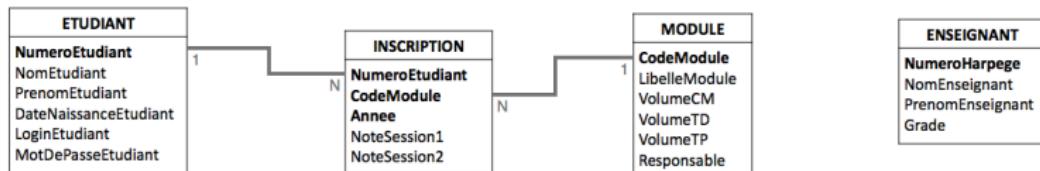


Méthodologie :

1. repérer les attributs compatibles (domaine + sémantique)
2. définir le sens et le type du lien (1..N ou 1..1)

# Un exemple

Essayons de trouver les liens.

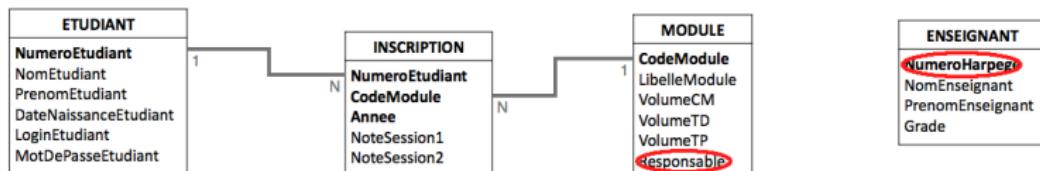


Méthodologie :

1. repérer les attributs compatibles (domaine + sémantique)
2. définir le sens et le type du lien (1..N ou 1..1)

# Un exemple

Essayons de trouver les liens.

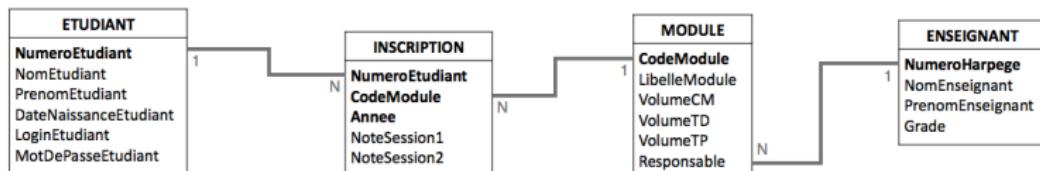


Méthodologie :

1. repérer les attributs compatibles (domaine + sémantique)
2. définir le sens et le type du lien (1..N ou 1..1)

# Un exemple

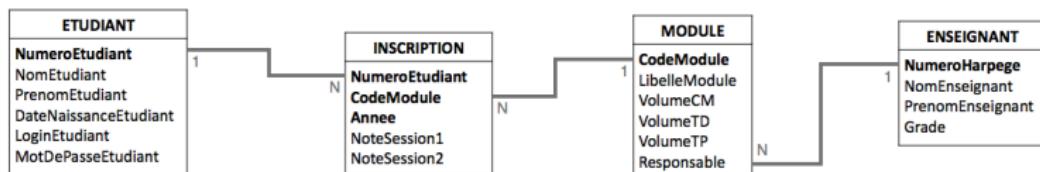
Essayons de trouver les liens.



Méthodologie :

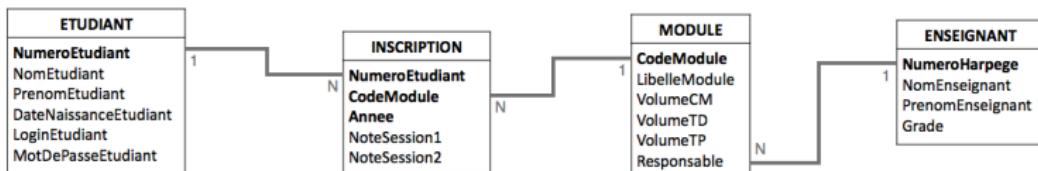
1. repérer les attributs compatibles (domaine + sémantique)
2. définir le sens et le type du lien (1..N ou 1..1)

# Un exemple



Un étudiant peut-il être inscrit plusieurs fois à un même module ?

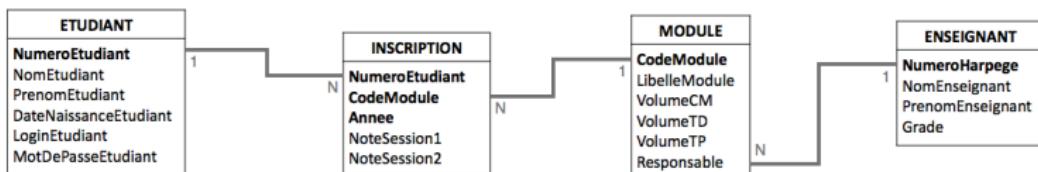
# Un exemple



Un étudiant peut-il être inscrit plusieurs fois à un même module ?

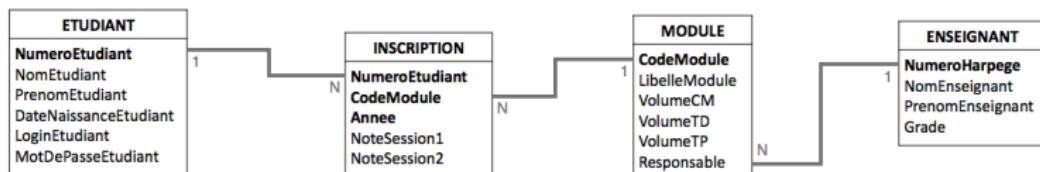
- ⇒ Oui, mais pas la même année : la clé primaire de la relation INSCRIPTION est composée des 3 attributs (NumeroEtudiant, CodeModule, Annee) on peut donc avoir plusieurs fois le même couple de valeurs pour (NumeroEtudiant, CodeModule), mais pour des valeurs différentes d'Anneee.

# Un exemple



Un enseignant peut-il être responsable de plusieurs modules ?

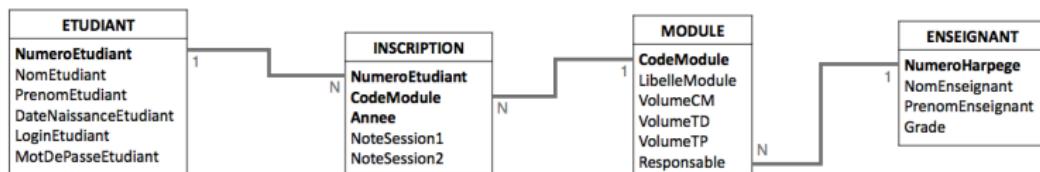
# Un exemple



Un enseignant peut-il être responsable de plusieurs modules ?

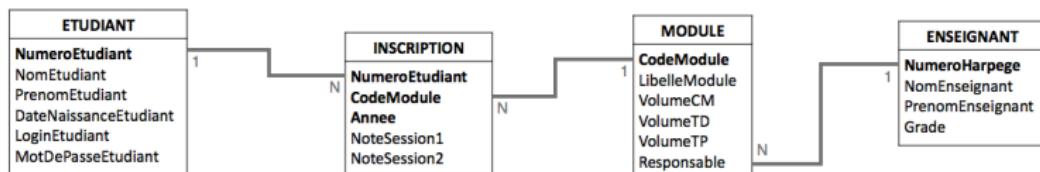
⇒ Oui, par définition du lien 1..N entre ENSEIGNANT et MODULE.

# Un exemple



Plusieurs enseignants peuvent-il être responsable d'un même module ?

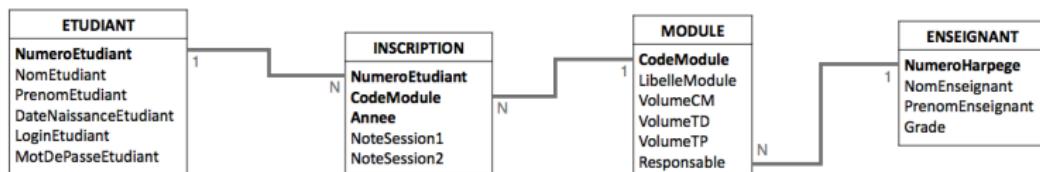
# Un exemple



Plusieurs enseignants peuvent-il être responsable d'un même module ?

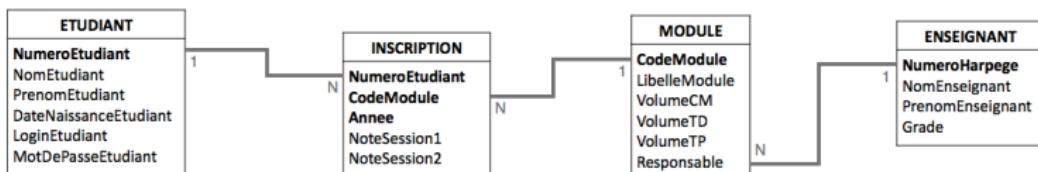
- ⇒ Non, car pour une valeur de CodeModule, identifiant un module, on a qu'une seule valeur de Responsable.

# Un exemple



Supposons une politique de suppression RESTRICTED sur le lien entre MODULE et ENSEIGNANT. Que se passe-t-il si on supprime un enseignant ?

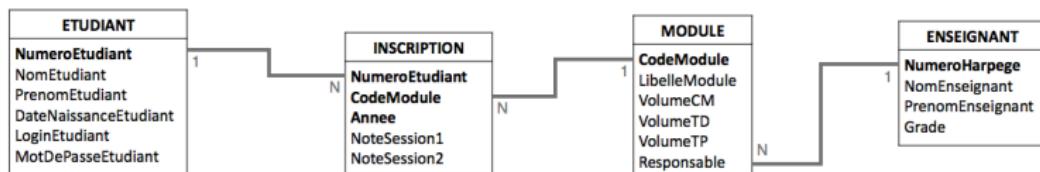
# Un exemple



Supposons une politique de suppression RESTRICTED sur le lien entre MODULE et ENSEIGNANT. Que se passe-t-il si on supprime un enseignant ?

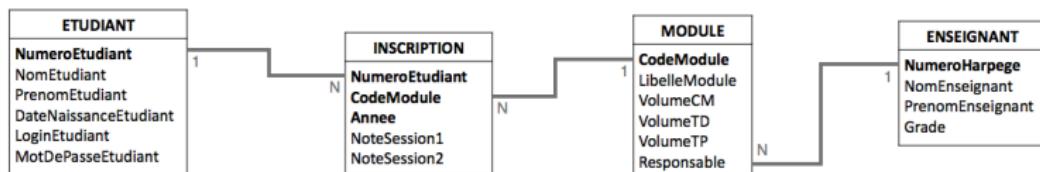
- ⇒ S'il est référencé comme Responsable dans la relation MODULE, alors cette suppression sera interdite.
- ⇒ S'il n'est pas référencé comme Responsable dans la relation MODULE, alors cette suppression sera possible.

# Un exemple



Supposons une politique de suppression CASCADE sur le lien entre MODULE et ENSEIGNANT, et une politique RESTRICTED entre MODULE et INSCRIPTION. Que se passe-t-il si on supprime un enseignant qui est responsable d'un module où des étudiants sont inscrits ?

# Un exemple



Supposons une politique de suppression CASCADE sur le lien entre MODULE et ENSEIGNANT, et une politique RESTRICTED entre MODULE et INSCRIPTION. Que se passe-t-il si on supprime un enseignant qui est responsable d'un module où des étudiants sont inscrits ?

- ⇒ Cette suppression sera interdite, car elle suppose de supprimer en cascade le module, ce qui est impossible, car celui-ci est référencé dans la relation INSCRIPTION.

# Plan du cours

---

1. Définitions de base

2. Intégrité des données

3. Un exemple

4. Les opérateurs relationnels

Quelques rappels : logique, théorie des ensembles

Les opérateurs ensemblistes

Les opérateurs relationnels spécifiques

Les opérateurs relationnels dérivés

# Les opérateurs relationnels

- ▶ Relations  $\Leftrightarrow$  Ensembles
- ▶ Opérateurs relationnels  $\Leftrightarrow$  Opérateurs ensemblistes
- ▶ A partir d'une ou de deux relations, un opérateur relationnel permet de calculer une nouvelle relation.
- ▶ Les opérateurs relationnels sont utilisés en particulier pour effectuer des requêtes de récupération de données.

# Les opérateurs relationnels

On trouve différents types d'opérateurs relationnels.

## Opérateurs relationnels

Ils se classent en trois catégories :

- ▶ les opérateurs ensemblistes classiques : union, intersection, différence, produit cartésien.
- ▶ les opérateurs relationnels spécifiques : sélection, projection, jointure.
- ▶ les opérateurs dérivés : division, jointure externe, etc.

# Logique et théorie des ensembles

## Un peu de logique

Les connecteurs logique “standards” sont généralement les suivants :

- ▶ et logique, opérateur binaire de conjonction, de notation mathématique  $\wedge$
- ▶ ou logique, opérateur binaire de disjonction, de notation mathématique  $\vee$
- ▶ non logique, opérateur unaire de négation, de notation mathématique  $\neg$
- ▶ implication, opérateur binaire, de notation mathématique  $\Rightarrow$
- ▶ équivalence, opérateur binaire, de notation mathématique  $\Leftrightarrow$

Ces opérateurs prennent comme opérandes des expressions booléennes (qui s'évaluent soit à vrai/true/1 soit à faux/false/0) et forment une expression booléenne.

Ces connecteurs sont utilisés dans tous les langages de programmation, mais nous allons également les utiliser pour définir formellement d'autres opérateurs qui dérivent de leur application.

# Logique et théorie des ensembles

Quelques tables de vérité des connecteurs standards.

## Les tables de vérité

Et logique :

a	b	$a \wedge b$
faux	faux	faux
faux	vrai	faux
vrai	faux	faux
vrai	vrai	vrai

# Logique et théorie des ensembles

Quelques tables de vérité des connecteurs standards.

## Les tables de vérité

Ou logique :

a	b	$a \vee b$
faux	faux	faux
faux	vrai	vrai
vrai	faux	vrai
vrai	vrai	vrai

# Logique et théorie des ensembles

Quelques tables de vérité des connecteurs standards.

## Les tables de vérité

Non logique :

a	$\neg a$
faux	vrai
vrai	faux

# Logique et théorie des ensembles

Quelques tables de vérité des connecteurs standards.

## Les tables de vérité

Implication :

a	b	$a \Rightarrow b$
faux	faux	vrai
faux	vrai	vrai
vrai	faux	faux
vrai	vrai	vrai

L'implication  $a \Rightarrow b$  peut se réécrire  $(\neg a) \vee b$

# Logique et théorie des ensembles

Quelques tables de vérité des connecteurs standards.

## Les tables de vérité

Equivalence logique :

a	b	$a \Leftrightarrow b$
faux	faux	vrai
faux	vrai	faux
vrai	faux	faux
vrai	vrai	vrai

L'équivalence  $a \Leftrightarrow b$  peut se réécrire  $(a \Rightarrow b) \wedge (b \Rightarrow a)$ .

# Logique et théorie des ensembles

## Un peu de théorie des ensembles

On va définir les opérateurs ensemblistes basiques suivants :

- ▶ Le constructeur d'ensemble en extension  $\{a, b, c, d\}$
- ▶ Le constructeur d'ensemble en compréhension  $\{x \mid P(x)\}$
- ▶ Le prédictat d'appartenance d'un élément à un ensemble  $x \in S$
- ▶ Les prédictats d'égalité et d'inégalité entre ensembles  $S = T$ ,  $S \neq T$
- ▶ Le prédictat d'inclusion (inclusion stricte) d'un ensemble dans un autre  $S \subseteq T$  ( $S \subset T$ )

Les deux premiers permettent de construire (définir) des ensembles. Les suivants permettent de donner des propriétés sur les ensembles.

# Logique et théorie des ensembles

## Constructeurs d'ensembles

- ▶ Définition en extension de l'ensemble  $S$ , contenant les 6 éléments 4, 8, 15, 16, 23, et 42.

$$S = \{4, 8, 15, 16, 23, 42\}$$

- ▶ Définition en compréhension de l'ensemble  $T$ , contenant les éléments de  $S$  supérieurs à 10.

$$T = \{x \mid x \in S \wedge x \geq 10\}$$

Cette expression se lit “l'ensemble des  $x$  *tels que*  $x$  appartienne à  $S$  et soit supérieur ou égal à 10”.

- ▶ Une constante désigne l'ensemble qui ne contient aucun élément. Elle s'appelle l'*ensemble vide* et se note  $\emptyset$  (i.e. équivalent à  $\{ \}$ ).

# Logique et théorie des ensembles

## Définition de l'opérateur d'appartenance

L'opérateur d'appartenance est un opérateur binaire, dénoté  $x \in S$  qui s'évalue à vrai si l'élément désigné par  $x$  appartient à l'ensemble désigné par  $S$ .

Attention, cet opérateur n'est pas commutatif, pour des raisons de typage des opérandes : sa première opérande est un élément, sa seconde est un ensemble d'éléments.

## Utilisation de l'opérateur d'appartenance

Supposons  $S = \{4, 8, 15, 16, 23, 42\}$ .

- ▶ l'expression  $42 \in S$  est évaluée à vrai.
- ▶ l'expression  $5 \in S$  est évaluée à faux.
- ▶ l'expression  $S \in S$  est mal formée (erreur de typage).
- ▶ l'expression  $x \in \emptyset$  est évaluée à faux (quelque soit  $x$ ).

# Logique et théorie des ensembles

## Définition de l'égalité entre ensembles

Deux ensembles  $S$  et  $T$  sont égaux si et seulement si ils possèdent exactement les mêmes éléments.

$$(\forall x. x \in S \Rightarrow x \in T) \wedge (\forall y. y \in T \Rightarrow y \in S)$$

Ils sont différents dans le cas contraire.

$$(\exists x. x \in S \wedge x \notin T) \vee (\exists y. y \in T \wedge y \notin S)$$

## Egalité et inégalité d'ensembles

Supposons  $S = \{a, b, c, d\}$ ,  $T = \{a, b, c\}$ , et  $U = \{a, b, c\}$ . On a alors :

- ▶  $S \neq T$
- ▶  $S \neq U$
- ▶  $T = U$

# Logique et théorie des ensembles

## Définition de l'opérateur d'inclusion

L'opérateur d'inclusion est un opérateur binaire, dénoté  $S \subseteq T$ , qui s'évalue à vrai si tous les éléments de  $S$  appartiennent à  $T$ .

$$S \subseteq T \Leftrightarrow \forall x. x \in S \Rightarrow x \in T$$

L'inclusion stricte, dénotée  $S \subset T$ , s'évalue à vrai si tous les éléments de  $S$  sont dans  $T$  et s'il existe des éléments de  $T$  qui ne sont pas dans  $S$  (les ensembles ne sont pas égaux).

$$S \subset T \Leftrightarrow S \subseteq T \wedge S \neq T$$

# Logique et théorie des ensembles

## Inclusion stricte et non-stricte

Supposons  $S = \{4, 8, 15, 16, 23, 42\}$  et  $T = \{x \mid x \in S \wedge x \geq 10\} = \{15, 16, 23, 42\}$

- ▶ l'expression  $T \subseteq S$  est évaluée à vrai.
- ▶ l'expression  $T \subset S$  est évaluée à vrai.
- ▶ l'expression  $S \subseteq T$  est évaluée à faux (non-commutativité).
- ▶ l'expression  $S \subseteq S$  est évaluée à vrai.
- ▶ l'expression  $S \subset S$  est évaluée à faux.

Deux cas particuliers :

- ▶ l'expression  $\emptyset \subseteq S$  est toujours évaluée à vrai (quelque soit  $S$ ).
- ▶ l'expression  $\emptyset \subset S$  est toujours évaluée à vrai (sauf si  $S = \emptyset$ ).

On remarque que :  $S = T \Leftrightarrow (S \subseteq T \wedge T \subseteq S)$

# Les opérateurs ensemblistes

La théorie des ensemble est prépondérante en informatique théorique, mais aussi appliquée (pas uniquement pour les bases de données).

Les opérateurs vus précédemment se transposent au niveau des relations, les relations étant des ensembles de t-uplets.

# Les opérateurs ensemblistes

## Définition de l'union ensembliste

L'union de deux ensembles  $S$  et  $T$ , dénotée  $S \cup T$ , produit un nouvel ensemble contenant tous les éléments de  $S$  et tous les éléments de  $T$ .

$$S \cup T = \{x \mid x \in S \vee x \in T\}$$

## Exemple d'union ensembliste

Soient  $S = \{4, 8, 15, 16, 23\}$  et  $T = \{4, 15, 23, 42\}$ , on a :

$$S \cup T = \{4, 8, 15, 16, 23, 42\}$$

(Rappel : il n'y a pas d'ordre dans les ensembles)

# Les opérateurs ensemblistes

## Union de deux relations

L'**union** permet de créer une relation dans laquelle apparaissent tous les t-uplets apparaissant dans au moins une des deux relations de départ. L'union de deux relations  $R_1$  et  $R_2$  se dénote  $R_1 \text{ UNION } R_2$  ou  $R_1 \cup R_2$ .

## Exemple d'union de relations

ETUDIANTSINFO

Numero*	Nom
23794	Avenia
32911	Gigant
33818	Maillard
34812	Smolinski

ETUDIANTSPASINFO

Numero*	Nom
32936	Perriot
33061	Chatelier
35684	Souvignes

ETUDIANTSINFO  $\cup$   
ETUDIANTSPASINFO

Numero*	Nom
23794	Avenia
32911	Gigant
32936	Perriot
33061	Chatelier
33818	Maillard
34812	Smolinski
35684	Souvignes

# Les opérateurs ensemblistes

## Définition de l'intersection ensembliste

L'intersection de deux ensembles  $S$  et  $T$ , dénotée  $S \cap T$ , produit un nouvel ensemble contenant tous les éléments de  $S$  qui sont également dans  $T$ .

$$S \cap T = \{x \mid x \in S \wedge x \in T\}$$

## Exemples d'intersections ensembliste

Soient  $S_1 = \{4, 8, 15, 16, 23\}$  et  $T_1 = \{4, 15, 23, 42\}$ , on a :

$$S_1 \cap T_1 = \{4, 15, 23\}$$

Soient  $S_2 = \{4, 8, 16, 23\}$  et  $T_2 = \{15, 42\}$ , on a :

$$S_2 \cap T_2 = \emptyset$$

# Les opérateurs ensemblistes

## Intersection de deux relations

L'**intersection** permet de créer une relation dans laquelle apparaissent tous les t-uplets contenus à la fois, dans la première et dans la seconde relation de départ.

L'intersection de deux relations  $R_1$  et  $R_2$  se dénote  $R_1 \text{ INTER } R_2$  ou  $R_1 \cap R_2$ .

## Exemple d'intersection de relations

ETUDIANTSGRB

Numero*	Nom
23794	Avenia
32911	Gigant
32936	Perriot
33061	Chatelier
33818	Maillard
34812	Smolinski
35684	Souvignes

ETUDIANTSINFO

Numero*	Nom
23485	Cabodi
24871	Riblet
23794	Avenia
32911	Gigant
33818	Maillard
34812	Smolinski

ETUDIANTSINFO  $\cap$

ETUDIANTSGRB

Numero*	Nom
23794	Avenia
32911	Gigant
33818	Maillard
34812	Smolinski

# Les opérateurs ensemblistes

## Définition de la différence ensembliste

La différence entre deux ensembles  $S$  et  $T$ , dénotée  $S - T$ , produit un nouvel ensemble contenant tous les éléments de  $S$  qui ne sont pas dans  $T$ .

$$S - T = \{x \mid x \in S \wedge x \notin T\}$$

## Exemples de différence ensembliste

Soient  $S = \{4, 8, 15, 16, 23\}$  et  $T = \{4, 15, 23, 42\}$ , on a :

$$S - T = \{8, 16\}$$

# Les opérateurs ensemblistes

## Définition de la différence ensembliste

La différence entre deux ensembles  $S$  et  $T$ , dénotée  $S - T$ , produit un nouvel ensemble contenant tous les éléments de  $S$  qui ne sont pas dans  $T$ .

$$S - T = \{x \mid x \in S \wedge x \notin T\}$$

## Réécriture de l'intersection avec des différences

Soient  $S = \{4, 8, 15, 16, 23\}$  et  $T = \{4, 15, 23, 42\}$ , on a :

$$S \cap T = S - (S - T) = S - \{8, 16\} = \{4, 15, 23\}$$

ou alors

$$S \cap T = T - (T - S) = T - \{42\} = \{4, 15, 23\}$$

# Les opérateurs ensemblistes

## Différence de deux relations

La **différence** de deux relations  $R_1$  et  $R_2$  permet de créer une relation dans laquelle apparaissent tous les t-uplets de  $R_1$  n'apparaissant pas dans  $R_2$ .

La différence de deux relations  $R_1$  et  $R_2$  se dénote  $R_1 - R_2$ .

## Exemple de différence de relations

ETUDIANTSGRB

Numero*	Nom
23794	Avenia
32911	Gigant
32936	Perriot
33061	Chatelier
33818	Maillard
34812	Smolinski
35684	Souvignes

ETUDIANTSINFO

Numero*	Nom
23485	Cabodi
24871	Riblet
23794	Avenia
32911	Gigant
33818	Maillard
34812	Smolinski

ETUDIANTSGRB –  
ETUDIANTSINFO

Numero*	Nom
32936	Perriot
33061	Chatelier
35684	Souvignes

# Une remarque sur ces opérateurs

## Remarque sur UNION, INTERSECTION, et DIFFERENCE

Attention, ces trois opérateurs ne s'appliquent qu'entre des relations qui possèdent les mêmes attributs (et donc au moins le même degré).

Autrement, elles n'ont pas de sens.

# Les opérateurs ensemblistes

## Définition du produit cartésien de deux ensembles

Le produit cartésien de deux ensembles  $S$  et  $T$ , dénoté  $S \times T$ , est l'ensemble des couples dont la première composante est issue de  $S$  et la seconde composante est issue de  $T$ .

$$S \times T = \{(x, y) \mid x \in S \wedge y \in T\}$$

Dans ce contexte, tous les éléments de  $S$  sont associés à tous les éléments de  $T$ . La cardinalité (taille) du produit cartésien est égale au produit des cardinalités des opérandes.

## Produit cartésien

Soient  $S = \{x, y, z\}$  et  $T = \{1, 2, 3, 4\}$  deux ensembles. Le produit cartésien de  $S$  et de  $T$ , est l'ensemble des couples

$$S \times T = \{(x, 1), (x, 2), (x, 3), (x, 4), (y, 1), (y, 2), (y, 3), (y, 4), (z, 1), (z, 2), (z, 3), (z, 4)\}$$

On remarque que :  $\text{card}(S \times T) = \text{card}(S) \times \text{card}(T) = 3 \times 4 = 12$

# Les opérateurs ensemblistes

## Produit cartésien de deux relations

Le **produit cartésien** de deux relations  $R_1$  et  $R_2$  permet de créer une relation dans laquelle apparaissent tous les t-uplets de  $R_1$  auxquels sont concaténés les t-uplets de  $R_2$ .

Le produit cartésien de deux relations  $R_1$  et  $R_2$  se dénote  $R_1 \times R_2$ . On note que le degré de  $R_1 \times R_2 = \text{degré de } R_1 + \text{degré de } R_2$ .

## Exemple de produit cartésien de relations

ETUDIANTS1

Numero*	Nom
23794	Avenia
32911	Gigant
34812	Smolinski

MODULES1

CodeModule*
BD_L1
PROG_L1

ETUDIANTS1  $\times$  MODULES1

Numero*	Nom	CodeModule*
23794	Avenia	BD_L1
23794	Avenia	PROG_L1
32911	Gigant	BD_L1
32911	Gigant	PROG_L1
34812	Smolinski	BD_L1
34812	Smolinski	PROG_L1

# Les opérateurs relationnels spécifiques

## Définition

La **sélection**, également appelée **restriction**, permet d'extraire d'une relation les t-uplets satisfaisant une condition donnée.

La sélection des t-uplets de  $R$  satisfaisant un critère  $C$  se dénote  $S(C)R$ .

## Quelques propriétés

- ▶ La relation résultante de la sélection possède la même en-tête que la relation initiale  $R$ , mais une cardinalité différente (inférieure ou égale à celle de  $R$ ).
- ▶ Seuls les t-uplets satisfaisant le critère de sélection apparaissent dans le résultat.

# Les opérateurs relationnels spécifiques

## Critère de sélection $C$

Le critère de sélection  $C$  doit respecter une syntaxe particulière :

$< Attribut > < Operateur > < Valeur >$

- ▶ Les attributs sont les attributs de la relation  $R$ .
- ▶ Les opérateurs sont les opérateurs classiques de comparaison :  $=, \neq, >, <, \geq, \leq$ . Les conditions atomiques peuvent se connecter à l'aide des opérateurs logiques classiques ET, OU et NON.
- ▶ Les valeurs sont celles du domaine de l'attribut considéré.

# Les opérateurs relationnels spécifiques

## Sélection dans une relation

Considérons la relation ETUDIANT vue précédemment.

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

On souhaite calculer :

ETUDIANTS A VANT M E S 18 = S(DateNaissanceEtudiant  $\leq$  30-06-1998) ETUDIANT

# Les opérateurs relationnels spécifiques

## Sélection dans une relation

On sélectionne les t-uplets satisfaisant le critère.

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

On souhaite calculer :

ETUDIANTS A VANT M ES18 = S(DateNaissanceEtudiant  $\leq$  30-06-1998) ETUDIANT

# Les opérateurs relationnels spécifiques

## Sélection dans une relation

On obtient le résultat : ETUDIANTS A VANT MES18

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard

# Les opérateurs relationnels spécifiques

## Définition

La **projection** permet de définir une relation consistant en l'ensemble de tous les t-uplets de la relation de départ dans laquelle seuls les attributs de projection sont conservés.

La projection de  $R$  sur ses attributs  $Att_1, Att_2, \dots$  se dénote  $[Att_1, Att_2, \dots]R$ .

## Quelques propriétés

- ▶ Le degré de la relation résultante est inférieur (ou égal) au degré de la relation de départ.
- ▶ La cardinalité peut également être diminuée, si le résultat contient des doublons (par définition interdits dans les ensembles, ils n'apparaîtront qu'une fois).

# Les opérateurs relationnels spécifiques

## Application de la projection

Considérons la relation ETUDIANT

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
32827	Signoret	Adrien	3-10-1992	asignoret	asignoret
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

On veut calculer ETUDIANTSNP = [NomEtudiant, PrenomEtudiant] ETUDIANTS

# Les opérateurs relationnels spécifiques

## Application de la projection

On réalise la projection sur les deux attributs mentionnés

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
32827	Signoret	Adrien	3-10-1992	asignoret	asignoret
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

On veut calculer  $\text{ETUDIANTSNP} = [\text{NomEtudiant}, \text{PrenomEtudiant}] \text{ ETUDIANTS}$

# Les opérateurs relationnels spécifiques

## Application de la projection

On obtient le résultat ETUDIANTSNP

Nom	Prenom
Avenia	Adrien
Gigant	Simon
Maillard	Cynthia
Signoret	Adrien
Smolinski	Marie

# Les opérateurs relationnels spécifiques

## Application de la projection

Considérons la relation ETUDIANT

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
32827	Signoret	Adrien	3-10-1992	asignoret	asignoret
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

On veut calculer ETUDIANTS.PRENOM = [PrenomEtudiant] ETUDIANTS

# Les opérateurs relationnels spécifiques

## Application de la projection

On réalise la projection sur le seul attribut mentionné

Numero*	Nom	Prenom	DateNaissance	Login	MotDePasse
23794	Avenia	Adrien	18-08-1998	aavenia	aavenia
32911	Gigant	Simon	26-02-1998	sgigant	sgigant
33818	Maillard	Cynthia	06-05-1998	cmaillard	cmaillard
32827	Signoret	Adrien	3-10-1992	asignoret	asignoret
34812	Smolinski	Marie	21-11-1998	msmolinski	msmolinski

On veut calculer ETUDIANTSPRENOM = [PrenomEtudiant] ETUDIANTS

# Les opérateurs relationnels spécifiques

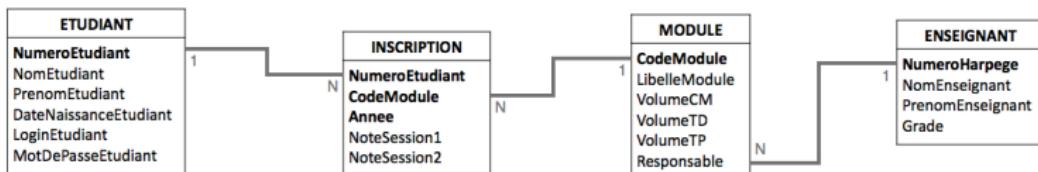
## Application de la projection

On obtient le résultat ETUDIANTS.PRENOM

Prenom
Adrien
Simon
Cynthia
Marie

Les relations étant des ensembles, la valeur “Adrien” n’apparaît qu’une seule fois.

# A vous de jouer



Donnez les requêtes permettant de calculer, sur notre exemple :

- ▶ Le libellé des modules qui ont plus de 18h de CM.
- ▶ Le nom et prénom des étudiants nés avant 1998.
- ▶ Le numéro des étudiants ayant validé leurs modules en session 1 en 2017.
- ▶ Les nom et prénom des enseignants ayant le grade “MCF”.
- ▶ Le numéro des étudiants inscrits en BD\_L1, mais qui ne l'ont jamais validé.

# Les opérateurs relationnels spécifiques

La jointure est un opérateur essentiel pour la manipulation des informations dans une base de données.

## Définition de la jointure

Une jointure de relations  $R_1$  et  $R_2$  consiste à construire une relation dont les t-uplets sont la concaténation d'un t-uplet de  $R_1$  et d'un t-uplet de  $R_2$  vérifiant une condition dite de jointure.

La jointure de deux relations  $R_1$  et  $R_2$  avec la condition de jointure  $C$  se note :  $R_1[C]R_2$ .

# Les opérateurs relationnels spécifiques

La jointure est un opérateur essentiel pour la manipulation des informations dans une base de données.

## Application de la jointure

Une jointure entre deux relations est possible lorsque celles-ci possède au moins un attribut commun au sens sémantique.

En pratique, le résultat de la jointure est équivalent à l'application successive d'un produit cartésien, suivi d'une sélection des t-uplets respectant le critère de jointure.

$$R_1[C]R_2 \Leftrightarrow S(C)(R_1 \times R_2)$$

# Les opérateurs relationnels spécifiques

## Exemple de jointure

On souhaite réaliser la jointure :

MOD1 [ MOD1.Responsable = ENS1.NumeroHarpege ] ENS1

	CodeModule*	LibelleModule	Responsable
MOD1	BD_L1	Bases de données	7914
	PROG_L1	Programmation	7358
	TEST_L3	Test de logiciels	7914

	NoHarpege*	Nom	Prenom
ENS1	7358	Janey	Nicolas
	7914	Dadeau	Frédéric
	32598	Paquette	Guillaume

# Les opérateurs relationnels spécifiques

## Exemple de jointure

On souhaite réaliser la jointure :

MOD1 [ MOD1.Responsable = ENS1.NumeroHarpege ] ENS1

	CodeModule*	LibelleModule	Responsable
MOD1	BD_L1	Bases de données	7914
	PROG_L1	Programmation	7358
	TEST_L3	Test de logiciels	7914

	NoHarpege*	Nom	Prenom
ENS1	7358	Janey	Nicolas
	7914	Dadeau	Frédéric
	32598	Paquette	Guillaume

# Les opérateurs relationnels spécifiques

## Exemple de jointure

On souhaite réaliser la jointure :

MOD1 [ MOD1.Responsable = ENS1.NumeroHarpege ] ENS1

CodeModule*	LibelleModule	Responsable	NoHarpege*	Nom	Prenom
BD_L1	Base de données	7914	7914	Dadeau	Frédéric
PROG_L1	Programmation	7358	7358	Janey	Nicolas
TEST_L3	Test de logiciels	7914	7914	Dadeau	Frédéric

Remarque : on aurait obtenu le même résultat avec  
ENS1 [ ENS1.NumeroHarpege = MOD1.Responsable ] MOD1

# Les opérateurs relationnels spécifiques

On trouve en réalité trois types de jointures, en fonction de la nature de la condition de jointure.

- ▶ La jointure naturelle
- ▶ L'équi-jointure
- ▶ La  $\theta$ -jointure

# Les opérateurs relationnels spécifiques

## Définition de la jointure naturelle

Une jointure naturelle de relations  $R_1$  et  $R_2$  consiste à construire une relation dont les t-uplets sont la concaténation d'un t-uplet de  $R_1$  et d'un t-uplet de  $R_2$  pour lesquels la condition de jointure spécifie que des attributs de même nom doivent porter la même valeur.

$$R_1[R_1.att = R_2.att]R_2 \text{ également noté } R_1[att]R_2$$

En conséquence, le résultat ne conservera qu'une seule des deux colonnes communes.

# Les opérateurs relationnels spécifiques

Jointure naturelle

On souhaite réaliser : ETU1 [ NumeroEtudiant ] MOD1

	NumeroEtudiant*	Nom	Prenom
ETU1	23794	Avenia	Adrien
	32911	Gigant	Simon
	33818	Maillard	Cynthia
	34812	Smolinski	Marie

	NumeroEtudiant*	CodeModule*
MOD1	23794	BD_L1
	23794	PROG_L1
	32911	BD_L1

# Les opérateurs relationnels spécifiques

Jointure naturelle

On souhaite réaliser : ETU1 [ NumeroEtudiant ] MOD1

	NumeroEtudiant*	Nom	Prenom
ETU1	23794	Avenia	Adrien
	32911	Gigant	Simon
	33818	Maillard	Cynthia
	34812	Smolinski	Marie

	NumeroEtudiant*	CodeModule*
MOD1	23794	BD_L1
	23794	PROG_L1
	32911	BD_L1

# Les opérateurs relationnels spécifiques

## Jointure naturelle

On souhaite réaliser : ETU1 [ NumeroEtudiant ] MOD1

NumeroEtudiant*	Nom	Prenom	CodeModule*
23794	Avenia	Adrien	BD_L1
23794	Avenia	Adrien	PROG_L1
32911	Gigant	Simon	BD_L1

On remarque que la colonne *NumeroEtudiant* n'apparaît qu'une seule fois dans le résultat.

# Les opérateurs relationnels spécifiques

## Définition de l'équi-jointure

L'équi-jointure de deux relations  $R_1$  et  $R_2$  est une jointure dont la condition de jointure est l'égalité des valeurs d'attributs  $att_1$  et  $att_2$ , appartenant respectivement aux relations  $R_1$  et  $R_2$ .

$$R_1[R_1.att_1 = R_2.att_2]R_2$$

Remarque : la jointure naturelle, vue précédemment, est un cas particulier d'équi-jointure, où les deux attributs sont les mêmes.

# Les opérateurs relationnels spécifiques

## Exemple d'équi-jointure

C'est le cas de la jointure déjà montrée précédemment :

MOD1 [ MOD1.Responsable = ENS1.NumeroHarpege ] ENS1

	CodeModule*	LibelleModule	Responsable
MOD1	BD_L1	Bases de données	7914
	PROG_L1	Programmation	7358
	TEST_L3	Test de logiciels	7914

	NoHarpege*	Nom	Prenom
ENS1	7358	Janey	Nicolas
	7914	Dadeau	Frédéric
	32598	Paquette	Guillaume

# Les opérateurs relationnels spécifiques

## Exemple d'équi-jointure

C'est le cas de la jointure déjà montrée précédemment :

MOD1 [ MOD1.Responsable = ENS1.NumeroHarpege ] ENS1

	CodeModule*	LibelleModule	Responsable
MOD1	BD_L1	Bases de données	7914
	PROG_L1	Programmation	7358
	TEST_L3	Test de logiciels	7914

	NoHarpege*	Nom	Prenom
ENS1	7358	Janey	Nicolas
	7914	Dadeau	Frédéric
	32598	Paquette	Guillaume

# Les opérateurs relationnels spécifiques

## Exemple d'équi-jointure

C'est le cas de la jointure déjà montrée précédemment :

MOD1 [ MOD1.Responsable = ENS1.NumeroHarpege ] ENS1

CodeModule*	LibelleModule	Responsable	NoHarpege*	Nom	Prenom
BD_L1	Base de données	7914	7914	Dadeau	Frédéric
PROG_L1	Programmation	7358	7358	Janey	Nicolas
TEST_L3	Test de logiciels	7914	7914	Dadeau	Frédéric

# Les opérateurs relationnels spécifiques

## Définition de la $\theta$ -jointure

La  $\theta$ -jointure de deux relations  $R_1$  et  $R_2$  selon une condition de jointure  $C$  est l'ensemble des t-uplets du produit cartésien de  $R_1$  et  $R_2$  satisfaisant la condition  $C$ .

$$R_1[C]R_2$$

Remarque : l'équi-jointure, et, par extension, la jointure naturelle, sont des cas particuliers de  $\theta$ -jointure, où la condition de jointure est une égalité.

# Les opérateurs relationnels spécifiques

## Exemple de $\theta$ -jointure

La condition d'une  $\theta$ -jointure n'est pas obligée de contenir une égalité :  
MOD1 [ MOD1.Responsable  $\neq$  ENS1.NumeroHarpege ] ENS1

	CodeModule*	LibelleModule	Responsable
MOD1	BD_L1	Bases de données	7914
	PROG_L1	Programmation	7358
	TEST_L3	Test de logiciels	7914
	NoHarpege*	Nom	Prenom
7914	Dadeau	Frédéric	
32598	Paquette	Guillaume	

# Les opérateurs relationnels spécifiques

## Exemple de $\theta$ -jointure

La condition d'une  $\theta$ -jointure n'est pas obligée de contenir une égalité :  
MOD1 [ MOD1.Responsable  $\neq$  ENS1.NumeroHarpege ] ENS1

	CodeModule*	LibelleModule	Responsable
MOD1	BD_L1	Bases de données	7914
	PROG_L1	Programmation	7358
	TEST_L3	Test de logiciels	7914
ENS1	NoHarpege*	Nom	Prenom
	7358	Janey	Nicolas
	7914	Dadeau	Frédéric
	32598	Paquette	Guillaume

# Les opérateurs relationnels spécifiques

## Exemple de $\theta$ -jointure

La condition d'une  $\theta$ -jointure n'est pas obligée de contenir une égalité :  
MOD1 [ MOD1.Responsable  $\neq$  ENS1.NumeroHarpege ] ENS1

	CodeModule*	LibelleModule	Responsable
MOD1	BD_L1	Bases de données	7914
	PROG_L1	Programmation	7358
	TEST_L3	Test de logiciels	7914
ENS1	NoHarpege*	Nom	Prenom
	7358	Janey	Nicolas
	7914	Dadeau	Frédéric
	32598	Paquette	Guillaume

# Les opérateurs relationnels spécifiques

## Exemple de $\theta$ -jointure

La condition d'une  $\theta$ -jointure n'est pas obligée de contenir une égalité :  
MOD1 [ MOD1.Responsable  $\neq$  ENS1.NumeroHarpege ] ENS1

	CodeModule*	LibelleModule	Responsable
MOD1	BD_L1	Bases de données	7914
	PROG_L1	Programmation	7358
	TEST_L3	Test de logiciels	7914
ENS1	NoHarpege*	Nom	Prenom
	7358	Janey	Nicolas
	7914	Dadeau	Frédéric
	32598	Paquette	Guillaume

# Les opérateurs relationnels spécifiques

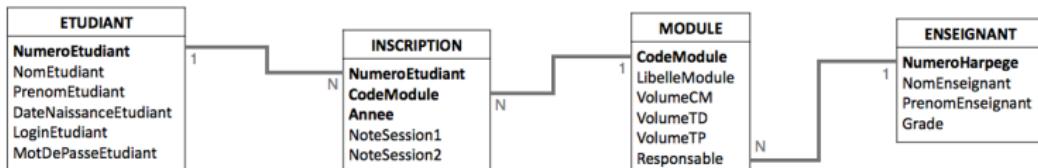
## Exemple de $\theta$ -jointure

La condition d'une  $\theta$ -jointure n'est pas obligée de contenir une égalité :

MOD1 [ MOD1.Responsable  $\neq$  ENS1.NumeroHarpege ] ENS1

CodeModule*	LibelleModule	Resp.	NoHarpege*	Nom	Prenom
BD_L1	Base de données	7914	7358	Janey	Nicolas
BD_L1	Base de données	7914	32598	Paquette	Guillaume
PROG_L1	Programmation	7358	7914	Dadeau	Frédéric
PROG_L1	Programmation	7358	32598	Paquette	Guillaume
TEST_L3	Test de logiciels	7914	7358	Janey	Nicolas
TEST_L3	Test de logiciels	7914	32598	Paquette	Guillaume

# A vous de jouer



Donnez les requêtes permettant de calculer, sur notre exemple :

- ▶ Le libellé des modules où est inscrit l'étudiant numéro 23794 en 2017.
- ▶ Le nom et prénom des étudiants inscrits dans un module de l'enseignant 7914.
- ▶ Les noms et prénoms des enseignants qui ont côtoyé l'étudiant numéro 23794.
- ▶ Les numéros des étudiants qui n'ont jamais côtoyé l'étudiant numéro 23974.
- ▶ Le libellé des modules communs aux étudiants 23974 et 33818.

# Les opérateurs relationnels dérivés

## Définition de la division

La division de la relation  $R_1(A_1, \dots, A_p, A_{p+1}, \dots, A_n)$  par la relation  $R_2(A_{p+1}, \dots, A_n)$  retourne une relation  $R_3(A_1, \dots, A_p)$  telle que tous les t-uplets de  $R_3$  concaténés à  $R_2$  apparaissent dans la relation  $R_1$ .

On a :  $R_3 \times R_2 \subseteq R_1$

Cette division se note :  $R_3 = R_1[R_1.A_{p+1}, \dots, R_1.A_n / R_2.A_{p+1}, \dots, R_2.A_n]R_2$

# Les opérateurs relationnels dérivés

## Exemple de division

$R_1$	
$A_1$	$A_2$
a	x
a	y
a	z
b	x
b	z
c	x

$R_2$	
$A_2$	
x	
z	

$$R_1 [R_1.A_2 / R_2.A_2] R_2$$

# Les opérateurs relationnels dérivés

## Exemple de division

$R_1$	
$A_1$	$A_2$
a	x
a	y
a	z
b	x
b	z
c	x

$R_2$	
$A_2$	
x	
z	

$R_1[R_1.A_2 / R_2.A_2]R_2$	
$A_1$	

# Les opérateurs relationnels dérivés

## Exemple de division

$R_1$	
$A_1$	$A_2$
a	x
a	y
a	z
b	x
b	z
c	x

$R_2$	
$A_2$	
x	
z	

$R_1[R_1.A_2 / R_2.A_2]R_2$	
$A_1$	
a	

# Les opérateurs relationnels dérivés

## Exemple de division

$R_1$	
$A_1$	$A_2$
a	x
a	y
a	z
b	x
b	z
c	x

$R_2$	
$A_2$	
x	
z	

$R_1[R_1.A_2 / R_2.A_2]R_2$	
$A_1$	
a	
b	

# Les opérateurs relationnels dérivés

## Quelle utilité pour la division

Cet opérateur permet :

- ▶ de rechercher dans une relation tous les t-uplets qui sont complétés par concaténation par ceux d'une autre relation.
- ▶ de répondre à des questions de la forme : "*quelque soit x, trouver y*" de façon simple.

## Notation

La relation par laquelle on divise ne doit pas contenir d'autres attributs que ceux intervenants dans la division. Il est donc possible de simplifier l'écriture de la division :

$$R_1 \text{ DIV } R_2$$

# Les opérateurs relationnels dérivés

Un premier exemple de division

INSCRIPTION	
NoEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1
32911	PROG_L1
32911	TEST_L3
33818	BD_L1
34812	PROG_L1
34812	BD_L1
34812	TEST_L3

MOD1	
CodeModule*	
BD_L1	
PROG_L1	
TEST_L3	

INSCRIPTION DIV MOD1	
NoEtudiant*	
32911	
34812	

On calcule ainsi les étudiants qui sont inscrits dans tous les modules de MOD1.

# Les opérateurs relationnels dérivés

Un second exemple de division

INSCRIPTION

NoEtudiant*	CodeModule*
23794	BD_L1
23794	PROG_L1
32911	BD_L1
32911	PROG_L1
32911	TEST_L3
33818	BD_L1
34812	PROG_L1
34812	BD_L1
34812	TEST_L3

ETU1

NoEtudiant*
23794
32911
33818
34812

INSCRIPTION DIV ETU1

CodeModule*
BD_L1

On calcule ainsi les modules auxquels sont inscrits tous les étudiants de ETU1.

# Les opérateurs relationnels dérivés

## Jointure interne ...

Dans le cas où, pour un t-uplet de la relation  $R_1$  il n'existe pas de t-uplet de la relation  $R_2$  satisfaisant la condition de jointure, ce t-uplet n'apparaît pas dans le résultat de la jointure.

## ... vs. jointure externe

La **jointure externe** permet de garder les t-uplets de  $R_1$  même si aucun t-uplet correspondant n'existe dans la relation  $R_2$ .

# Les opérateurs relationnels dérivés

## Définition de la jointure externe

La jointure externe permet de créer une relation à partir de deux relations  $R_1$  et  $R_2$  par jointure des deux relations, et ajout des t-uplets des relations  $R_1$  et  $R_2$  ne participant pas à la jointure, avec des valeurs NULL pour les attributs de l'autre relation.

La notation est similaire à celle d'une jointure, on rajoute un + du côté de la relation pour laquelle on souhaite conserver tous les t-uplets.

$$R_1 + [C] + R_2$$

# Les opérateurs relationnels dérivés

## Exemple de jointures

**MOD1**

CodeModule*	LibelleModule	Responsable
BD_L1	Bases de données	7914
PROG_L1	Programmation	7358
SR_L3	Systèmes et réseaux	22223

**ENS1**

NoHarpege*	Nom	Prenom
7358	Janey	Nicolas
7914	Dadeau	Frédéric
32598	Paquette	Guillaume

# Les opérateurs relationnels dérivés

## Exemple de jointures

MOD1

CodeModule*	LibelleModule	Responsable
BD_L1	Bases de données	7914
PROG_L1	Programmation	7358
SR_L3	Systèmes et réseaux	22223

ENS1

NoHarpege*	Nom	Prenom
7358	Janey	Nicolas
7914	Dadeau	Frédéric
32598	Paquette	Guillaume

Jointure interne : MOD1 [ MOD1.Responsable = ENS1.NoHarpege ] ENS1

CodeModule*	LibelleModule	Resp.	NoHarpege	Nom	Prenom
BD_L1	Bases de données	7914	7914	Dadeau	Frédéric
PROG_L1	Programmation	7358	7358	Janey	Nicolas

# Les opérateurs relationnels dérivés

## Exemple de jointures

**MOD1**

CodeModule*	LibelleModule	Responsable
BD_L1	Bases de données	7914
PROG_L1	Programmation	7358
SR_L3	Systèmes et réseaux	22223

**ENS1**

NoHarpege*	Nom	Prenom
7358	Janey	Nicolas
7914	Dadeau	Frédéric
32598	Paquette	Guillaume

Jointure externe : MOD1 + [ MOD1.Responsable = ENS1.NoHarpege ] ENS1

CodeModule*	LibelleModule	Resp.	NoHarpege	Nom	Prenom
BD_L1	Bases de données	7914	7914	Dadeau	Frédéric
PROG_L1	Programmation	7358	7358	Janey	Nicolas
SR_L3	Systèmes et réseaux	22223	NULL	NULL	NULL

# Les opérateurs relationnels dérivés

## Exemple de jointures

MOD1

CodeModule*	LibelleModule	Responsable
BD_L1	Bases de données	7914
PROG_L1	Programmation	7358
SR_L3	Systèmes et réseaux	22223

ENS1

NoHarpege*	Nom	Prenom
7358	Janey	Nicolas
7914	Dadeau	Frédéric
32598	Paquette	Guillaume

Jointure externe : MOD1 [ MOD1.Responsable = ENS1.NoHarpege ] + ENS1

CodeModule*	LibelleModule	Resp.	NoHarpege	Nom	Prenom
BD_L1	Bases de données	7914	7914	Dadeau	Frédéric
PROG_L1	Programmation	7358	7358	Janey	Nicolas
NULL	NULL	NULL	32598	Paquette	Guillaume

# Les opérateurs relationnels dérivés

## Exemple de jointures

**MOD1**

CodeModule*	LibelleModule	Responsable
BD_L1	Bases de données	7914
PROG_L1	Programmation	7358
SR_L3	Systèmes et réseaux	22223

**ENS1**

NoHarpege*	Nom	Prenom
7358	Janey	Nicolas
7914	Dadeau	Frédéric
32598	Paquette	Guillaume

Jointure externe : MOD1 + [ MOD1.Responsable = ENS1.NoHarpege ] + ENS1

CodeModule*	LibelleModule	Resp.	NoHarpege	Nom	Prenom
BD_L1	Bases de données	7914	7914	Dadeau	Frédéric
PROG_L1	Programmation	7358	7358	Janey	Nicolas
SR_L3	Systèmes et réseaux	22223	NULL	NULL	NULL
NULL	NULL	NULL	32598	Paquette	Guillaume