

Conception logique

Projet base de données – 1A – ENS Cachan

GROSSHANS Nathan

`nathan.grosshans@lsv.ens-cachan.fr`

3 mars 2017

Introduction

Motivation

- ▶ Les **données** d'une base de données sont **organisées**, au **niveau logique**, d'une certaine manière conforme à un certain **modèle logique** de données.
- ▶ Parmi ces modèles logiques on notera le modèle relationnel, ou encore XML.
- ▶ Après l'élaboration du schéma conceptuel, il est nécessaire de **traduire** celui-ci en schéma logique **équivalent** conforme à un certain modèle.
- ▶ Ceci s'effectue par **applications successives** au schéma source de **transformations préservant la sémantique** du schéma.

Livre de référence utilisé pour cette séance

Hainaut J.-L., *Bases de données : Concepts, utilisation et développement*. 1ère édition (2009).

Introduction

Modèle

Modèle logique de données classique : le **modèle relationnel**.

Définition

Un schéma est dit **relationnel** s'il respecte les contraintes suivantes.

1. Le schéma comporte des types d'entités (appelés **tables**).
2. Tout type d'entités possède au moins un attribut (appelé **colonne**).
3. Un attribut est mono-valué et atomique ; il est obligatoire ou facultatif.
4. Les seules contraintes sont celles qui sont induites par les identifiants (**identifiant primaire** ou **secondaire**) et les attributs de référence (**clés étrangères**).
5. Le schéma relationnel ne contient pas d'autres constructions.
6. Les noms ne contiennent pas d'espaces, de tirets, de signes de ponctuation, d'accents ou d'autres symboles spéciaux autres que ' _ '.

Introduction

Clés étrangères

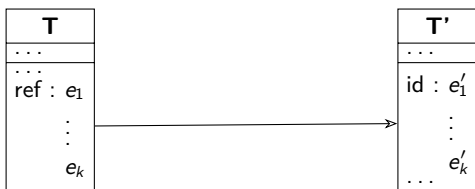
Définitions

- ▶ Étant donné deux tables T et T' , une **clé étrangère** de T vers T' est un ensemble de colonnes E de T , copie d'un identifiant E' de T' vérifiant que pour tout n-uplet t de T , il existe un n-uplet t' de T' tel que la valuation donnée aux colonnes de E dans t corresponde à la valuation donnée aux colonnes de E' dans t' (**contrainte référentielle**).
- ▶ Les colonnes de E et E' sont donc en **même nombre** et de **mêmes types**.
- ▶ Les colonnes de E sont soit toutes obligatoires, soit toutes facultatives et coexistantes (tout n-uplet a soit toutes ses colonnes à NULL, soit elles sont toutes différentes de NULL et référencent un n-uplet de T').
- ▶ En général, E' est l'identifiant primaire de T' .

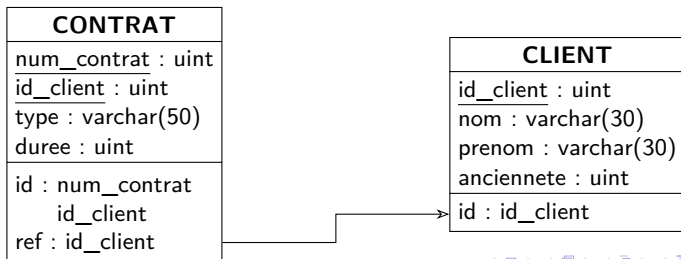
Introduction

Clés étrangères

Représentation graphique



Exemple



Traduction

Principe

- ▶ **Application successive** de **règles de traduction** à un schéma EA source jusqu'à obtenir un schéma relationnel.
- ▶ De telle manière que le schéma obtenu après chaque étape soit **sémantiquement équivalent** au schéma original.
- ▶ On supposera ici que le schéma EA source est **simple**, c'est-à-dire qu'il vérifie les propriétés suivantes.
 - ▶ Les types d'associations sont simples, c'est-à-dire binaires, sans attributs ni identifiants, et chacun des rôles ayant pour cardinalité 0-1, 1-1 ou 0-N.
 - ▶ Le schéma ne contient pas de relation est-un.

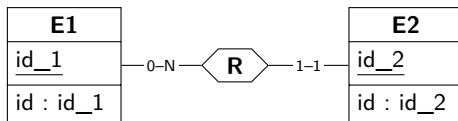
Types d'entités

Les types d'entités se traduisent directement, sans transformation.

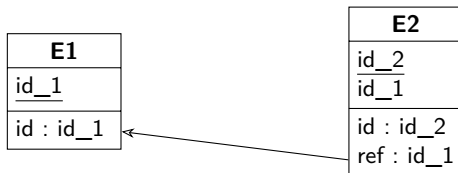
Traduction

Types d'associations

Un-à-plusieurs



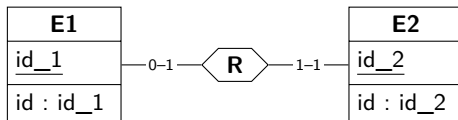
↕



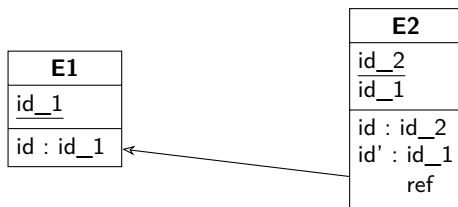
Traduction

Types d'associations

Un-à-un



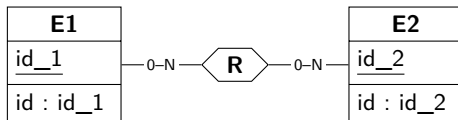
↕



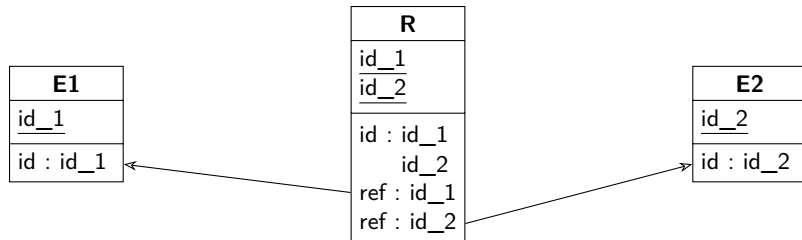
Traduction

Types d'associations

Plusieurs-à-plusieurs



}



Traduction

Divers

Identifiants

Soit E un type d'entités et un identifiant de E contenant :

- ▶ les **attributs** a_1, \dots, a_k ;
- ▶ les **rôles opposés** $R_1.E_1, \dots, R_l.E_l$.

Soit T la table obtenue par traduction de E , ainsi que T_1, \dots, T_l les tables obtenues par traduction, respectivement, de E_1, \dots, E_l .

On notera C_1, \dots, C_l les ensembles des colonnes référençant, respectivement, les identifiants primaires de T_1, \dots, T_l .

L'identifiant correspondant de T sera $\{a_1, \dots, a_k\} \cup \bigcup_{i=1}^l C_i$.

Noms

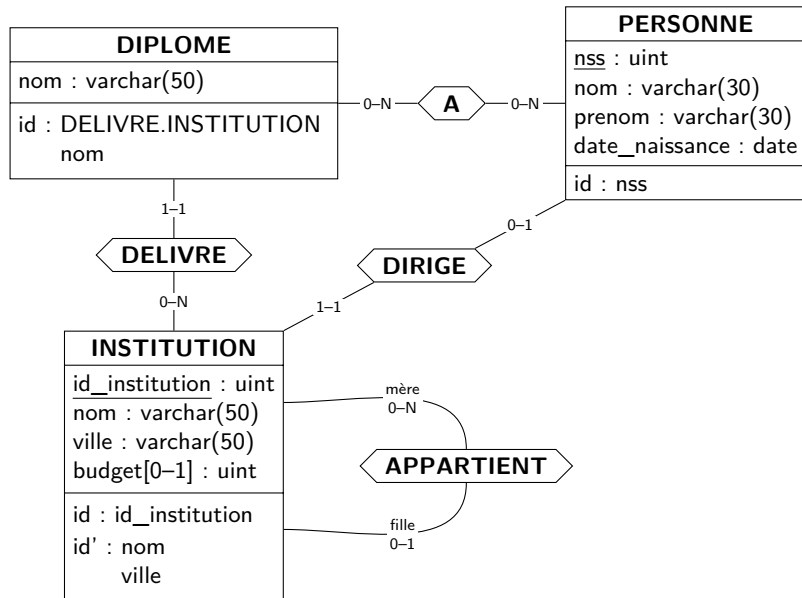
Veiller à ce que les noms soient **conformes** aux restrictions imposées dans la définition d'un schéma relationnel ; les convertir sinon.

Contraintes d'intégrité additionnelles

Il faut les **réexprimer** sur le schéma relationnel.

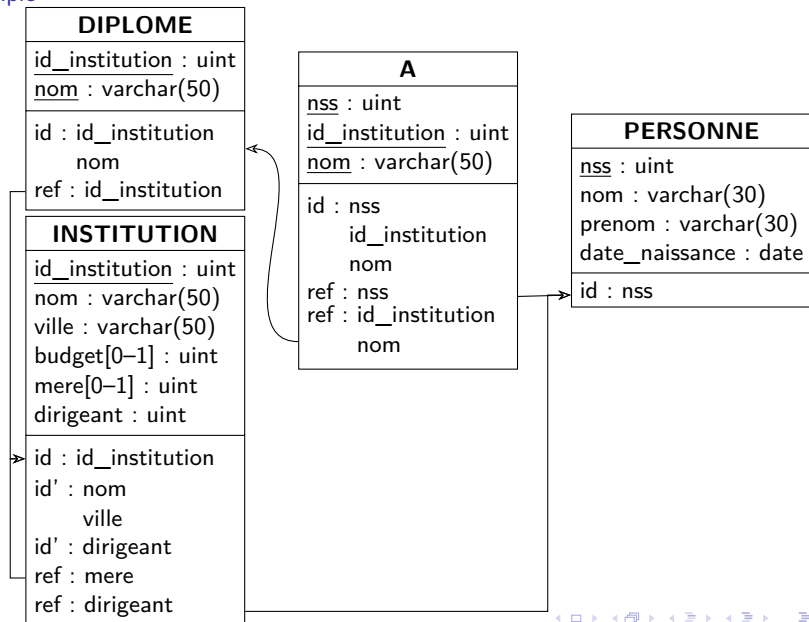
Traduction

Exemple



Traduction

Exemple



Restructuration

Principe

- ▶ Si le schéma EA source n'est pas simple, il faut d'abord le **restructurer** pour que ce soit le cas.
- ▶ Cela se fait à nouveau par **transformations successives préservant la sémantique**.

Cardinalité

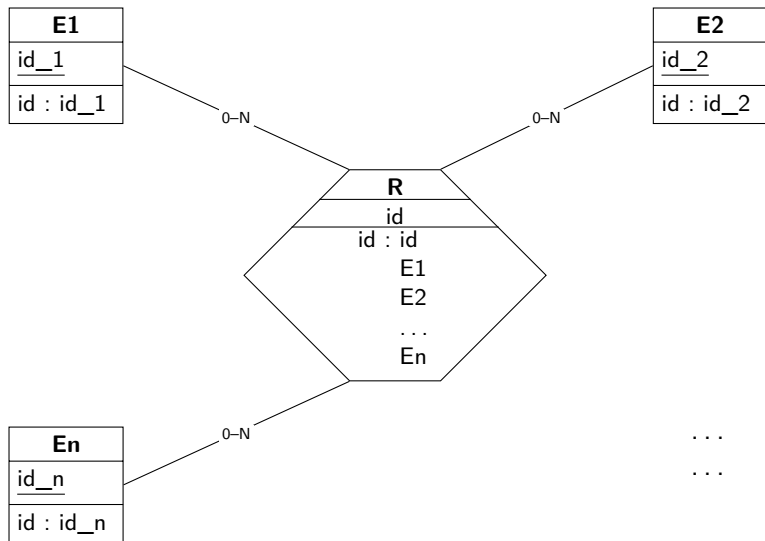
Si la cardinalité i - j d'un rôle dans un type d'associations n'est pas 0-1, 1-1 ou 0-N, alors on la remplace par 0-N et l'on ajoute une contrainte supplémentaire forçant la bonne cardinalité.

Contraintes d'intégrité additionnelles

Il faut veiller, si nécessaire, à les adapter suite à chaque étape de restructuration.

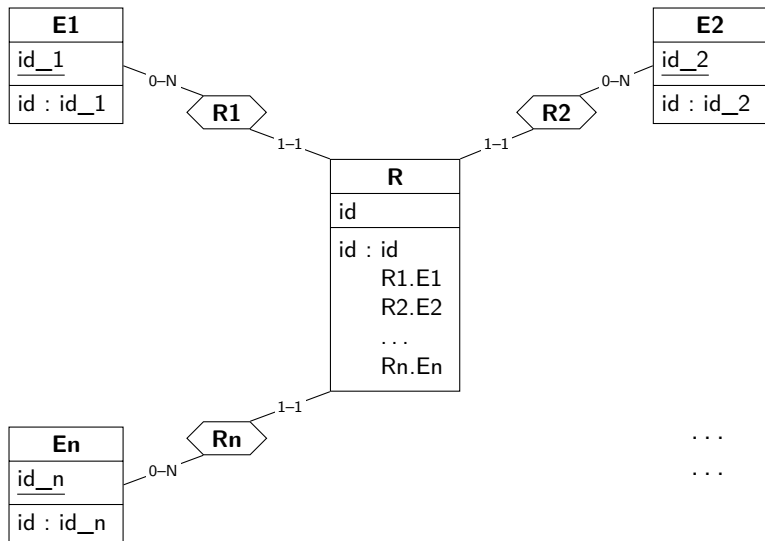
Restructuration

Types d'associations complexes



Restructuration

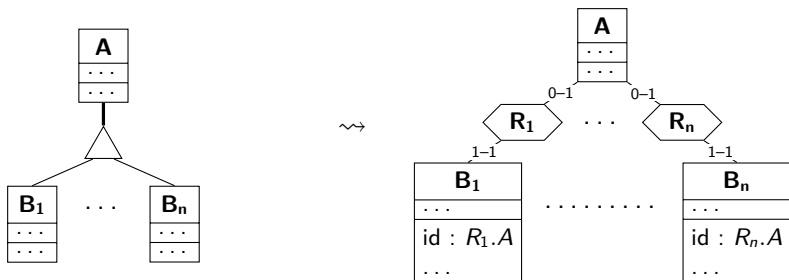
Types d'associations complexes



Restructuration

Relations est-un

Matérialisation



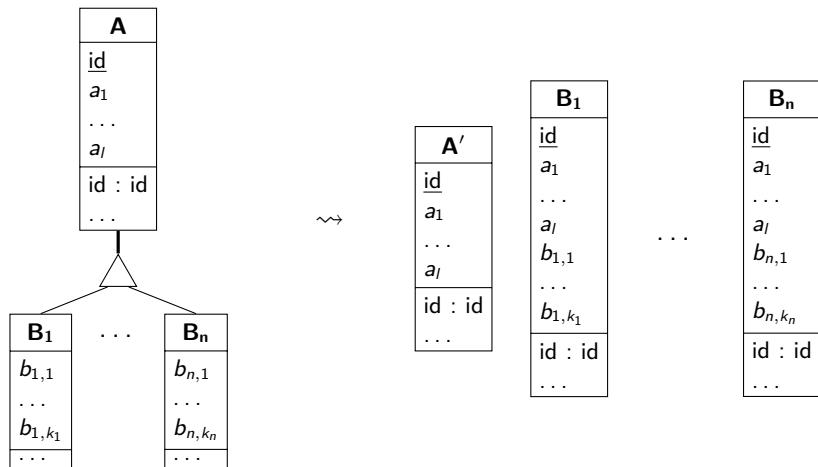
En fonction de la variante, il faut ajouter diverses contraintes.

- ▶ **Sans contraintes** : aucune contrainte.
- ▶ **Recouvrement total** :
 $\forall a \in A, |\{i \in [n] \mid \exists b \in B_i, b.R_i.A = a\}| \geq 1.$
- ▶ **Disjonction** : $\forall a \in A, |\{i \in [n] \mid \exists b \in B_i, b.R_i.A = a\}| \leq 1.$
- ▶ **Partition** : $\forall a \in A, |\{i \in [n] \mid \exists b \in B_i, b.R_i.A = a\}| = 1.$

Restructuration

Relations est-un

Héritage descendant



Restructuration

Relations est-un

Il faut obligatoirement ajouter les contraintes suivantes.

- ▶ $\forall i, j \in [n], \forall (b, b') \in B_i \times B_j, (b.id = b'.id \Rightarrow \forall \iota \in [l], b.a_\iota = b'.a_\iota)$.
- ▶ $\forall a' \in A', \nexists i \in [n], \exists b \in B_i, b.id = a'.id$.

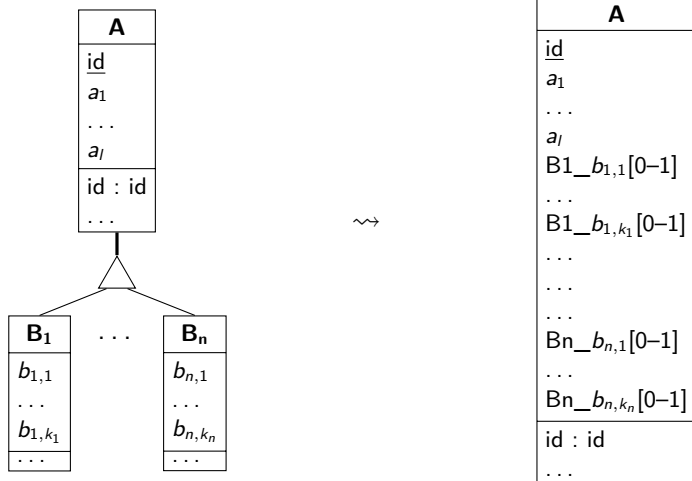
De plus, en fonction de la variante, il faut ajouter diverses contraintes.

- ▶ **Sans contraintes** : aucune contrainte.
- ▶ **Recouvrement total** : $|A'| = 0$.
- ▶ **Disjonction** :
 $\forall i, j \in [n], (i \neq j \Rightarrow \{b.id \mid b \in B_i\} \cap \{b'.id \mid b' \in B_j\} = \emptyset)$.
- ▶ **Partition** : $|A'| = 0 \wedge \forall i, j \in [n], (i \neq j \Rightarrow \{b.id \mid b \in B_i\} \cap \{b'.id \mid b' \in B_j\} = \emptyset)$.

Restructuration

Relations est-un

Héritage ascendant



Restructuration

Relations est-un

Il faut obligatoirement ajouter la contrainte suivante : $\forall a \in A, \forall s \in [n], \forall i, j \in [k_s], (Bs_b_{s,i} \neq \text{NULL} \Leftrightarrow Bs_b_{s,j} \neq \text{NULL})$.

Cette contrainte est plus complexe si au moins un des attributs d'un des sous-types est facultatif.

De plus, en fonction de la variante, il faut ajouter diverses contraintes.

- ▶ **Sans contraintes** : aucune contrainte.
- ▶ **Recouvrement total** :
 $\forall a \in A, |\{i \in [n] \mid Bi_b_{i,1} \neq \text{NULL}\}| \geq 1$.
- ▶ **Disjonction** : $\forall a \in A, |\{i \in [n] \mid Bi_b_{i,1} \neq \text{NULL}\}| \leq 1$.
- ▶ **Partition** : $\forall a \in A, |\{i \in [n] \mid Bi_b_{i,1} \neq \text{NULL}\}| = 1$.