

Analyse conceptuelle

Projet base de données – 1A – ENS Cachan

GROSSHANS Nathan

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

3 et 10 février 2017

Introduction

Motivation

- ▶ Utilité d'une base de données : modéliser un **univers** donné à propos duquel l'on stocke, extrait et gère des données.
- ▶ Univers modélisé par un schéma logique de base de données, en fonction de sa nature et d'un certain nombre de besoins auxquels la modélisation devra répondre.
- ▶ Conception d'un tel schéma : **tâche** plutôt **complexe**.
- ▶ Idée : prendre du recul, s'abstraire du modèle de données en produisant d'abord un **schéma conceptuel**.

Livre de référence utilisé pour cette séance

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

Introduction

Formalisme

Formalisme classique utilisé pour l'analyse conceptuelle : le modèle entité-association (EA).

Principe

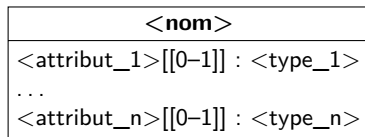
- ▶ L'univers considéré est peuplé d'un nombre fini d'**entités**.
- ▶ Chacune de ces entités est une instance d'un concept abstrait, un **type d'entités** décrivant comment cette entité est caractérisée.
- ▶ Les entités peuvent être liées l'une à l'autre à travers des **associations**, qui décrivent une relation entre les entités impliquées.
- ▶ Un **type d'associations** permet de décrire, de manière abstraite, les associations entre entités d'un certain ensemble de types d'entité.

Types d'entités

Définitions

- ▶ Chaque **type d'entités** est caractérisé par un **nom** (unique) et un ensemble d'**attributs**.
- ▶ Chaque attribut a un **nom** (unique pour ce type d'attributs), un **type** (de données) et peut soit être **obligatoire** (par défaut), soit être **optionnel** (c.-à-d. non spécifié).
- ▶ Une **entité** est une instance d'un type d'entités associant, à chaque attribut, une valeur prise dans le domaine défini par son type de données ou la valeur non spécifiée NULL, si autorisée.

Représentation graphique



Types d'entités

Exemples

PERSONNE
nss : uint
nom : varchar(30)
prenom : varchar(30)
date_naissance : date

nss ↦ 3 34 06...

nom ↦ Duck

prenom ↦ Donald

date_naissance ↦ 1934-06-09

INSTITUTION
nom : varchar(50)
ville : varchar(50)
budget[0-1] : uint

nom ↦ U.Duckburg

ville ↦ Duckburg

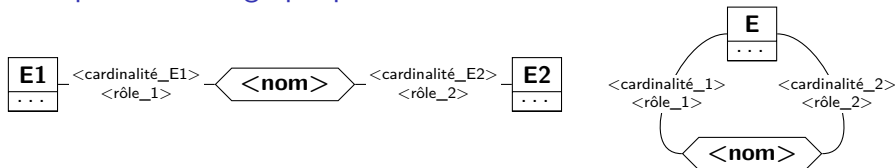
budget ↦ NULL

Types d'associations

Définitions

- ▶ Chaque **type d'associations** est caractérisé par un **nom** (unique), une **paire de types d'entités** (non nécessairement distincts), chacun jouant un **rôle** (distinct) et ayant une **cardinalité** dans ce type d'associations.
- ▶ Pour un type d'associations non cyclique, le rôle joué par chaque type d'entités est par défaut implicitement son nom.
- ▶ Une **association** est une instance d'un type d'association, c.-à-d. simplement une paire d'entités des types impliqués.

Représentation graphique



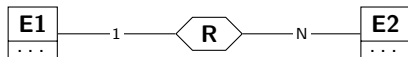
Types d'associations

Cardinalité

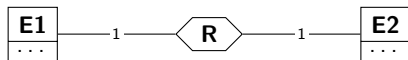
Prenons deux types d'entités E1 et E2, liés entre eux par le type d'associations R.

Classe fonctionnelle

- ▶ **Un-à-plusieurs** : chaque entité de E1 est associée à travers R à au plus une entité de E2, tandis que chaque entité de E2 est associée à travers R à zéro, une ou plusieurs entités de E1.



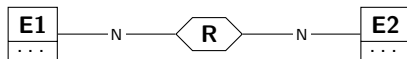
- ▶ **Un-à-un** : chaque entité de E1 est associée à travers R à au plus une entité de E2, et chaque entité de E2 est associée à travers R à au plus une entité de E1.



Types d'associations

Cardinalité

- ▶ **Plusieurs-à-plusieurs** : chaque entité de E1 est associée à travers R à zéro, une ou plusieurs entités de E2, et chaque entité de E2 est associée à travers R à zéro, une ou plusieurs entités de E1.

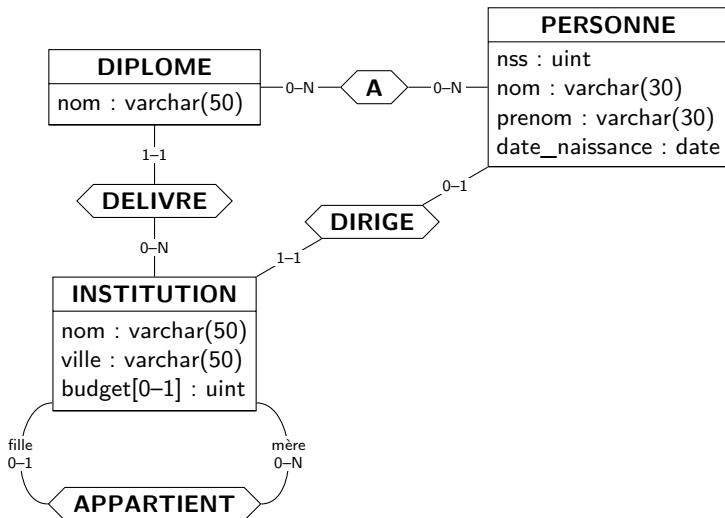


Cardinalité

- ▶ En plus de la classe fonctionnelle, on indique pour E1 et E2 si chaque entité concernée doit obligatoirement participer dans au moins une association de R ou non. Les cardinalités possibles sont donc : 0-1, 1-1, 0-N ou 1-N.
- ▶ Plus généralement, la cardinalité de E1 et E2 peut être tout intervalle $i-j$ où i est un entier naturel, j est $N (+\infty)$ ou un entier naturel plus grand que 0.

Types d'associations

Exemple



Identifiants

Définitions

- ▶ Chaque type d'entités a un ou plusieurs **identifiants**, chacun de ceux-ci identifiant de manière unique les entités de ce type.
- ▶ Étant donné un type d'entités E , un identifiant pour E est un **ensemble non vide minimal** pour l'inclusion :
 - ▶ d'attributs a_i de E ;
 - ▶ de rôles opposés $R_i.E_i$, où E_i est un type d'entités distinct de E associé à celui-ci via le type d'associations R_i , où $R_i.E$ a 0–1 ou 1–1 pour cardinalité ;

vérifiant que quelque soit la valuation donnée aux attributs a_i et les entités e_i de E_i , il existe au plus une entité de E ayant ces valeurs pour les attributs a_i et associée à chacune des entités e_i via R_i .

- ▶ On peut par exemple poser que :
 - ▶ $\{\text{nss}\}$ est un identifiant pour PERSONNE ;
 - ▶ $\{\text{nom, DELIVRE.INSTITUTION}\}$ est un identifiant pour DIPLOME.

Identifiants

- ▶ Les composants d'un identifiant peuvent soit être tous obligatoires (identifiant obligatoire), soit être tous facultatifs (identifiant facultatif).
- ▶ L'un de ces identifiants est déclaré **primaire**, et doit nécessairement être obligatoire ; les autres sont déclarés **secondaires**.

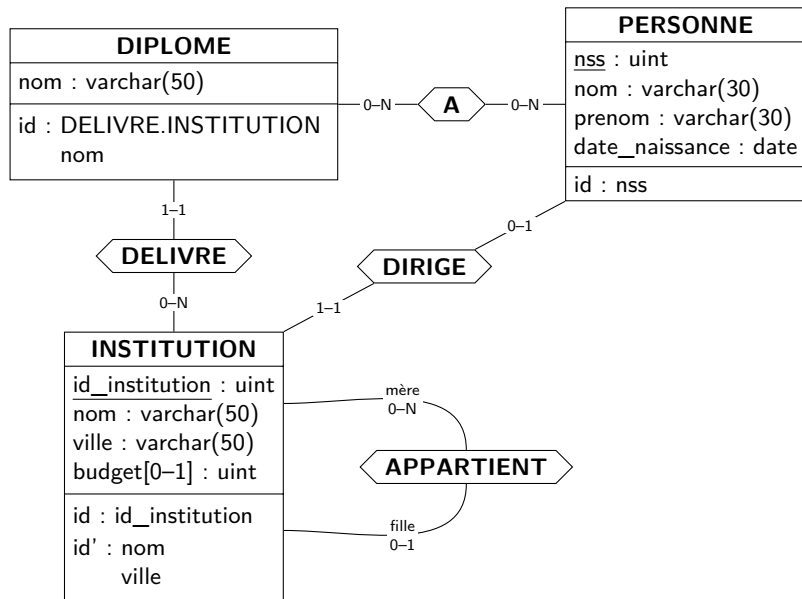
Représentation graphique

E
...
id : <liste_identifiant_primaire> id' : <liste_identifiant_secondaire_1> ... id' : <liste_identifiant_secondaire_k>

Lorsque l'identifiant primaire est composé uniquement d'attributs, ces derniers sont également soulignés.

Identifiants

Exemple



Contraintes d'intégrité

Une **contrainte d'intégrité** est une propriété que les instances décrites par un schéma (les entités, les associations et les valeurs d'attributs) doivent respecter. Deux types :

- ▶ les **contraintes statiques** : propriétés devant être vérifiées à tout instant par les instances ;
- ▶ les **contraintes dynamiques** : propriétés devant être vérifiées lors de l'ajout, la suppression ou la modification d'une ou plusieurs instances.

Contraintes de base

Elles sont **déjà données** directement dans le schéma entité-association, à savoir :

- ▶ les identifiants ;
- ▶ les attributs obligatoires ;
- ▶ les contraintes de cardinalité ;
- ▶ le type de données des attributs.

Contraintes d'intégrité

Contraintes additionnelles

Elles sont données **en supplément** du schéma entité-association et sont exprimées de manière semi-formelle ou en langage courant.

On peut trouver :

- ▶ des **contraintes de domaine** sur les valeurs données aux attributs (ex. : $\forall i \in \text{INSTITUTION}, 10000 \leq i.\text{budget} \leq 1000000000$);
- ▶ des **contraintes entre valeurs** d'attributs (ex. : $\forall (m, f) \in \text{APPARTIENT}, m.\text{id_institution} < f.\text{id_institution}$);
- ▶ des **contraintes d'existence** (ex. : $\forall m \in \text{INSTITUTION}, (\exists f \in \text{INSTITUTION t.q. } (m, f) \in \text{APPARTIENT}) \Rightarrow m.\text{budget} \neq \text{NULL}$);
- ▶ des **dépendances fonctionnelles** (voir plus loin);
- ▶ etc.

Contraintes d'intégrité

Dépendances fonctionnelles

Soient un type d'entités E et deux ensemble A, B non vides

- ▶ d'attributs de E ;
- ▶ de rôles opposés $R_i.E_i$, où E_i est un type d'entités distinct de E associé à celui-ci via le type d'associations R_i , où $R_i.E$ a 0-1 ou 1-1 pour cardinalité.

On a la **dépendance fonctionnelle** $A \rightarrow B$ ssi pour toute entité de E , la valeur donnée aux attributs de A et les entités associées des rôles opposés de A déterminent de manière unique la valeur donnée aux attributs de B et les entités associées des rôles opposés de B .

Par exemple :

- ▶ dans DIPLOME, on a $\{\text{nom, DELIVRE.INSTITUTION}\} \rightarrow \{\text{nom}\}$ (identifiant) ;
- ▶ on pourrait imposer que dans INSTITUTION, $\{\text{nom}\} \rightarrow \{\text{budget}\}$.

Élaboration d'un schéma conceptuel

Objectif

À partir d'une **spécification informelle** d'un certain univers, produire un **schéma EA** modélisant les **concepts** ainsi que les **faits** au sujet de ces concepts donnés par la spécification.

Procédure suggérée

La construction d'un schéma EA peut se faire de manière itérative au cours de la lecture de la spécification informelle, où chaque pas consiste à :

- ▶ **repérer** la prochaine **proposition élémentaire irréductible** de la forme $C_1 L_1 C_2 \cdots L_{k-1} C_k$ où les C_i sont des concepts et les L_i expriment des liens entre ces concepts ;
- ▶ **intégrer**, si ce n'est déjà le cas, le **fait exprimé** par l'ajout d'attributs, de types d'entités, de types d'associations, en restructurant éventuellement le schéma.

Élaboration d'un schéma conceptuel

Contraintes d'intégrité

À ajouter ensuite. Deux sources possibles :

- ▶ celles données explicitement par la spécification informelle ;
- ▶ celles issues d'une analyse critique ou d'un certain choix de conception (nécessaire pour les contraintes de base).

Normalisation du schéma

- ▶ **Simplification du schéma** : éliminer les constructions inutilement compliquées.
- ▶ **Élimination des redondances internes** : pour chaque type d'entités E , les seules dépendances fonctionnelles $A \rightarrow B$ doivent être telles que A est un sur-ensemble d'un identifiant de E , autrement il faut décomposer.

Élaboration d'un schéma conceptuel

Documentation du schéma

À chaque type d'entités, attribut et type d'associations est associée une **description** indiquant quel concept ou fait de la spécification informelle il représente, ou, à défaut, son utilité en regard de la spécification.

Complétude

Simplement **vérifier** que tous les concepts et faits de l'énoncé formel sont bien pris en compte et que tout ce qui est nécessaire est présent dans le schéma (identifiants, types de données, cardinalité, etc.).

Extensions du modèle

Utilité

- ▶ Augmenter l'**expressivité** du modèle EA, ainsi que la **lisibilité** et l'**évolutivité** des schémas.
- ▶ Elles ne sont pour ainsi dire **pas nécessaires**, car remplaçables par des constructions du modèle de base.

Remarques

- ▶ Seules quelques extensions, que j'ai jugées les plus utiles, sont rapidement présentées ici ; il en existe plus.
- ▶ Selon Hainaut, au-delà de ce qui correspond au modèle de base, peu de choses sont vraiment standardisées.

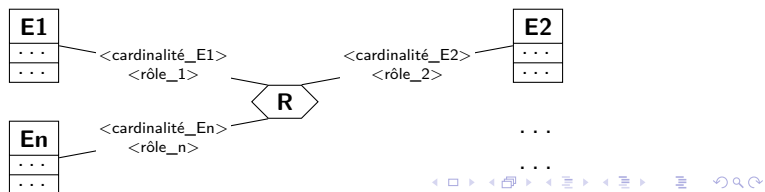
Extensions du modèle

Types d'associations n-aires

Définitions

- ▶ On définit des types d'associations à $n \geq 2$ rôles.
- ▶ Une association (instance) d'un tel type est donc un ***n*-uplet d'entités** des types impliqués.
- ▶ Chaque type d'entités impliqué joue **un ou plusieurs rôles**, chacun de ceux-ci a une **cardinalité** donnant, pour chaque entité de ce type, le nombre d'associations minimum et maximum dans lesquelles elle peut participer pour ce rôle.

Représentation graphique



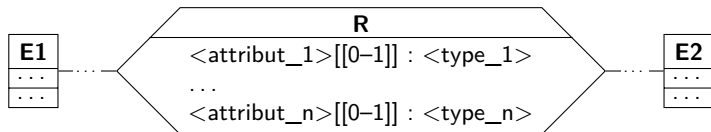
Le modèle EA étendu

Attributs de type d'associations

Définitions

- ▶ À chaque type d'associations, on peut ajouter une **liste d'attributs** (chacun ayant un nom, un type de données et un caractère obligatoire ou facultatif).
- ▶ Chaque association (instance) de ce type vient donc en plus avec une valuation de ces attributs.

Représentation graphique



Le modèle EA étendu

Identifiants de type d'associations

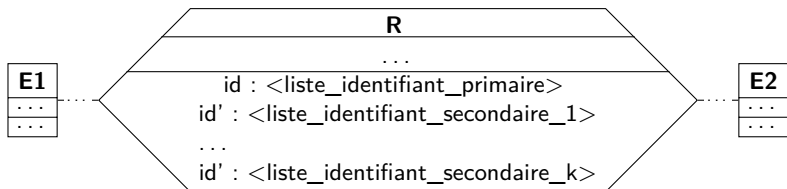
Définitions

Pour R un type d'associations, sont des identifiants :

- ▶ tout rôle de cardinalité $i-1$;
- ▶ tout ensemble d'attributs et de rôles de R déclaré comme tel ;
- ▶ l'ensemble des rôles de R , si R ne possède pas de rôle de cardinalité $i-1$ ou d'identifiant explicite.

Il ne doit pas exister de contraintes de dépendance fonctionnelle violant la minimalité des identifiants tels que définis ci-dessus.

Représentation graphique



Le modèle EA étendu

Relation de généralisation-spécialisation (est-un)

Définitions

- ▶ Soit A un **objet** (un type d'entités ou d'associations), on appellera **population de A** et on notera $\text{pop}(A)$ l'ensemble des instances de A à un moment déterminé.
- ▶ Dans le modèle EA de base, pour A et B deux types d'entités distincts, on a à tout instant $\text{pop}(A) \cap \text{pop}(B) = \emptyset$. Par la **relation est-un**, on va pouvoir exprimer le fait qu'une entité d'un certain type est aussi d'un **surtype plus général** (généralisation) et/ou d'un ou plusieurs **sous-types plus spécifiques** (spécialisation).

Exemple

On peut par exemple exprimer qu'une entité du type DIRECTEUR est aussi, plus généralement, du surtype PERSONNE et qu'une entité de ce type PERSONNE peut être, plus spécifiquement, du sous-type ALUMNI ou DIRECTEUR.

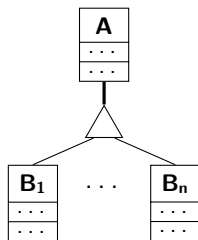
Le modèle EA étendu

Relation de généralisation-spécialisation (est-un)

Variantes et représentation graphique

Supposons que l'on veuille spécifier que A est un surtype de sous-types B_1, \dots, B_n . Il existe **plusieurs variantes** de relation est-un que l'on peut imposer entre A et B_1, \dots, B_n , selon les contraintes que l'on souhaite voir respectées par les populations de A et B_1, \dots, B_n .

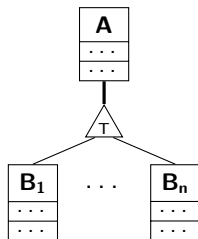
- ▶ **Sans contraintes** : $\bigcup_{i=1}^n \text{pop}(B_i) \subseteq \text{pop}(A)$.



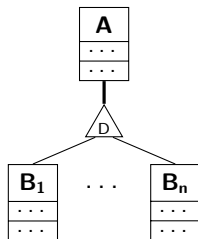
Le modèle EA étendu

Relation de généralisation-spécialisation (est-un)

- **Recouvrement total** : $\bigcup_{i=1}^n \text{pop}(B_i) = \text{pop}(A)$.



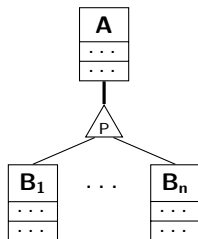
- **Disjonction** : $\bigcup_{i=1}^n \text{pop}(B_i) \subseteq \text{pop}(A)$ et pour tous $i, j \in \mathbb{N}$, $1 \leq i, j \leq n$, $\text{pop}(B_i) \cap \text{pop}(B_j) = \emptyset$.



Le modèle EA étendu

Relation de généralisation-spécialisation (est-un)

- ▶ **Partition** : $\bigcup_{i=1}^n \text{pop}(B_i) = \text{pop}(A)$ et pour tous $i, j \in \mathbb{N}$, $1 \leq i, j \leq n$, $\text{pop}(B_i) \cap \text{pop}(B_j) = \emptyset$.



Propriétés

- ▶ Toute entité est d'un **type propre** et de tous ses **surtypes directs ou indirects** ; elle possède les attributs, joue les rôles et est soumise aux contraintes de tous ses types (notion d'**héritage**).

Le modèle EA étendu

Relation de généralisation-spécialisation (est-un)

- ▶ Une **caractéristique héritée** peut être **restreinte** par rapport à sa définition dans le surtype (domaine, cardinalité, etc.).

Règles

- ▶ Le graphe des relations est-un (où l'on va des surtypes aux sous-types) doit former un **graphe orienté sans circuit**.
- ▶ Les **attributs propres et hérités** d'un type d'entités doivent avoir des **noms distincts**.
- ▶ Les **contraintes d'intégrité propres et héritées** d'un type d'entités doivent être **non contradictoires** (en particulier, seul un type d'entités source peut avoir un identifiant primaire).
- ▶ Si A et B sont deux types d'entités n'ayant **aucun surtype** direct ou indirect **commun**, alors $\text{pop}(A) \cap \text{pop}(B) = \emptyset$.