

**LOG660 - Bases de données de haute performance**

**Conception de BD relationnelles**

**ÉTS** Département de génie logiciel et des TI

---

---

---

---

---

---

---

---

**Modélisation des données**

■ Schéma conceptuel

- Modélise les classes, leurs attributs et leurs relations (ex: association, agrégation, spécialisation, etc.)
- *Exemple* diagramme de classe UML.

■ Schéma relationnel (conceptuel)

- Traduit le schéma conceptuel sous la forme d'un modèle relationnel (ex: tables, colonnes, clés, contraintes, etc.)
- Indépendant de la plateforme/BD utilisée (ex: Oracle versus SQL Server)
- *Exemple* diagramme UML avec tables, diagramme entités-associations

**ÉTS** Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

2

---

---

---

---

---

---

---

---

**Modélisation des données**

■ Schéma relationnel (MSP)

- MSP: Modèle spécifique à la plateforme
- Implémente le schéma relationnel conceptuel en considérant une plateforme spécifique
- Tient compte de la syntaxe spécifique à la plateforme/BD.
- *Exemple* SQL LDD Oracle (CREATE TABLE, VARCHAR2, etc.)

**ÉTS** Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

3

---

---

---

---

---

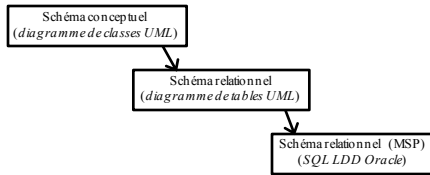
---

---

---

## Processus de conception

### ■ Objectif:

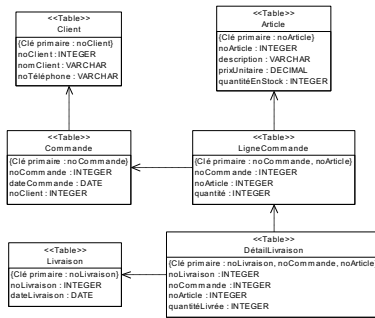


Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

4

## Schéma relationnel en UML



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

5

## Nom clé étrangère ≠ nom clé primaire

### ■ Étiquette de la relation de dépendance



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

6

## Traduction du schéma conceptuel en schéma relationnel

### ■ Étapes principales:

1. Traduire les classes en tables
2. Traduire les attributs et leur type
3. Définir la clé primaire
4. Traduire les associations



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

7

---

---

---

---

---

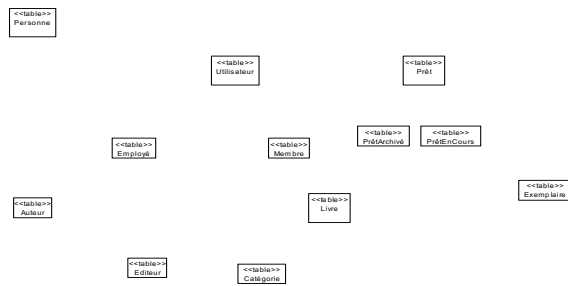
---

---

---

## Étape 1 : Traduire les classes en tables

### ■ Cas normal: une table par classe



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

8

---

---

---

---

---

---

---

---

## Étape 2 : Traduire les attributs et leur type

### ■ Cas possibles:

1. Type simple (ex: Integer)
2. Type énuméré (ex: enum)
3. Type complexe (ex: struct en C/C++)
4. Attributs multivalués (ex: tableau Integer[0..\*])
5. Attributs de classe (ex: static)



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

9

---

---

---

---

---

---

---

---

## Attributs de type simple

- Attribut de la classe → colonne de la table

Livre
(UNIQUE: ISBN)
ISBN : String
titre : String
annéeParution : TypeDonnéesAnnée

<<table>> Livre
{Clé candidate: ISBN}
ISBN : CHAR(13)
titre : VARCHAR(50)
annéeParution : DomaineAnnée

Note: les attributs UNIQUE deviennent des clés candidates



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

10

## Traduction des types de données

Type OCL	Type SQL2	Oracle 11
Boolean	BIT(1)	BOOLEAN ou CHAR(1) + CHECK
Integer	INTEGER ou SMALLINT	NUMBER(n), INTEGER
String	CHARACTER (CHAR) (n), CHARACTER VARYING (VARCHAR) (n)	VARCHAR2(n) (chaîne jusqu'à 4000 bytes), LONG ou LONG VARCHAR (chaîne jusqu'à 2G), CLOB (chaîne jusqu'à 4G), NCLOB (chaîne pour caractères encodés sur plusieurs octets)
Real	NUMERIC(p,s) (précision exacte), DECIMAL(p,s), REAL, DOUBLE PRECISION, FLOAT(n)	NUMBER(p,s), FLOAT, DOUBLE
Enum{v1,...,vn}	CHARACTER (CHAR) ou VARCHAR + CHECK ... IN (v1,...,vn) (possibilité de création de domaine)	Domaine non supporté
	DATE	DATE inclut TIME
	TIME	
	TIMESTAMP	
	BIT(n), BIT VARYING(n)	RAW(n : max = 255), LONG RAW (binaire jusqu'à 2G), BLOB (binaire jusqu'à 4G)
		BFILE (pointeur à un fichier externe)



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

11

## Types énumérés (cas 1)

- Petit domaine invariant
  - création d'un domaine VARCHAR + CHECK

Exemple
(UNIQUE: idExemple)
idExemple : String
dateAchat : Date
statut : enum(prêté, disponible, retiré)

<<table>> Exemple
{Clé candidate : idExemple}
idExemple : VARCHAR(10)
dateAchat : Date
statut : DomaineStatut

<<domain>> DomaineStatut
{VARCHAR(15) CHECK value IN ('prêté','disponible','retiré')}



Département de génie logiciel et des TI

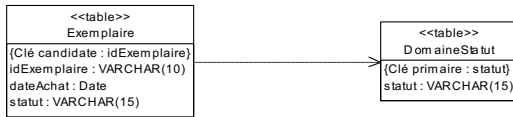
© R. Godin, C. Desrosiers - Hiver 2011

12

## Types énumérés (cas 2)

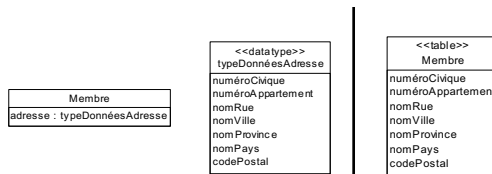
### ■ Gros domaine ou extensible

- création d'une table à part
- utilisé comme liste de valeurs (LOV Designer)
- introduction d'une clé primaire artificielle ?



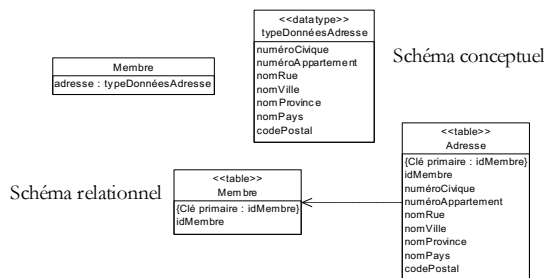
## Types complexes (cas 1)

### ■ Représentation explicite des attributs du type complexe



## Types complexes (cas 2)

### ■ Création d'une nouvelle table



### Attributs multivalués

- Table à part
  - clé étrangère + colonne pour l'attribut
- Petit tableau de taille fixe
  - $n$  colonnes (valeurs nulles)
- Encodage
  - invisible au SGBD
- Oracle8
  - VARRAY, NESTED TABLE



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

16

---

---

---

---

---

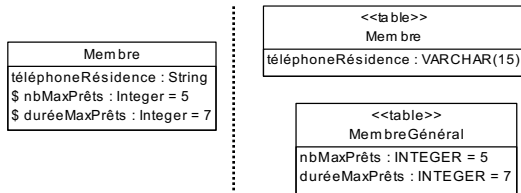
---

---

---

### Attributs de classe

- Création de tables supplémentaires



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

17

---

---

---

---

---

---

---

---

### Étape 3 : Définir la clé primaire

- Choix possibles:
  1. Clé artificielle générée
  2. Clé naturelle
  3. Simple ou composée



Département de génie logiciel et des TI

© R. Godin, C. Desrosiers - Hiver 2011

18

---

---

---

---

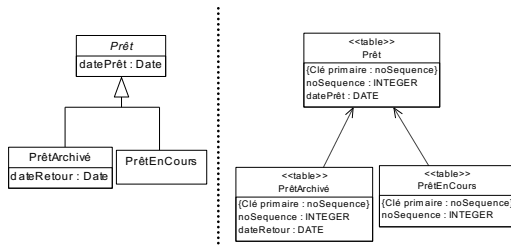
---

---

---

---

### Clés primaires: valeur générée



- De manière systématique ?
- Mécanisme de SEQUENCE Oracle

---

---

---

---

---

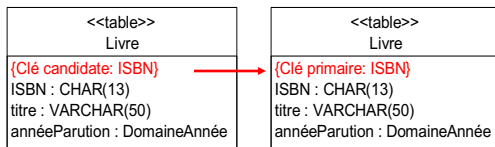
---

---

---

### Clés primaires: identifiant naturel

- Utilisation d'un identifiant naturel (UNIQUE)



- Si identifiant naturel trop lourd:
  - introduire clé primaire artificielle

---

---

---

---

---

---

---

---

### Clés primaires composées

- Formées de plusieurs colonnes de la table
- Performance réduite pour l'indexage
- Utilisées lorsque les lignes de la table ne sont pas référencées (ex: table de jointure)

---

---

---

---

---

---

---

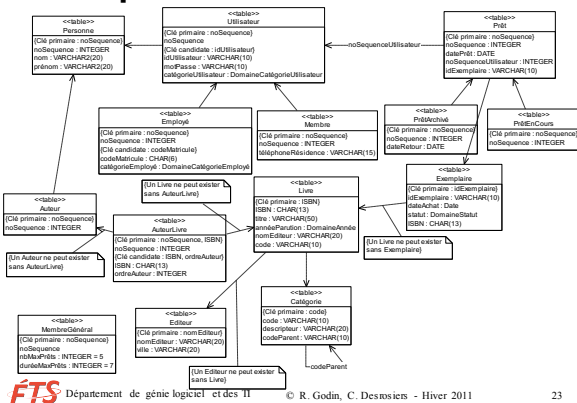
---

## Étape 4 : Traduire les relations

### ■ Cas possibles:

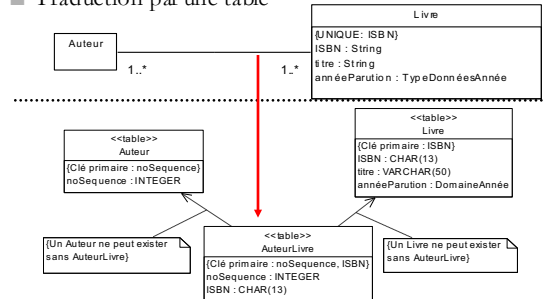
1. Plusieurs à plusieurs
2. Un à plusieurs
3. Un à un
4. Agrégation, composition
5. Spécialisation

## Exemple de schéma



## Association plusieurs à plusieurs

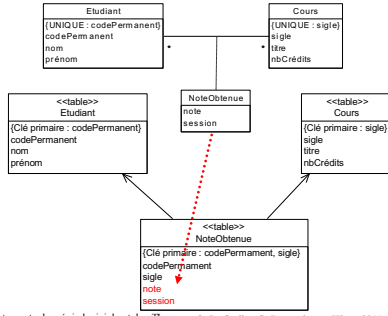
### ■ Traduction par une table





## Classe associative

### ■ Classe qualifiant une association



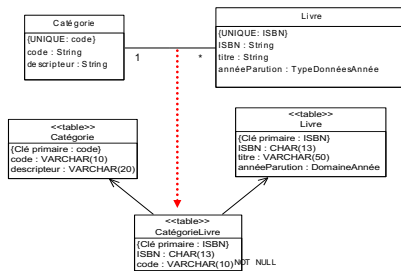
ÉTS

Département de génie logiciel et des TI © R. Godin, C. Desrosiers - Hiver 2011

25

## Association un à plusieurs (cas 1)

### ■ Traduction par une table



ÉTS

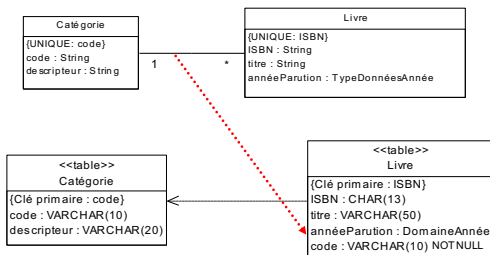
Département de génie logiciel et des TI © R. Godin, C. Desrosiers - Hiver 2011

26

## Association un à plusieurs (cas 2)

### ■ Ajout d'une clé étrangère

### ■ Navigation plus performante



ÉTS

Département de génie logiciel et des TI © R. Godin, C. Desrosiers - Hiver 2011

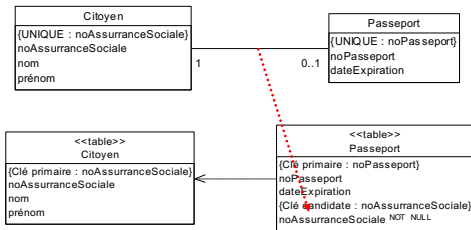
27

## Renommer la clé étrangère au besoin

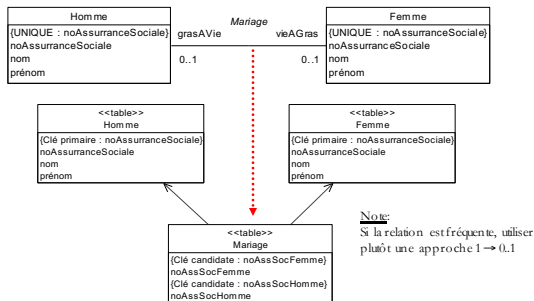


## Association un à un (cas 1 → 0..1)

■ Une clé étrangère (du côté obligatoire)



## Association un à un (cas 0..1 → 0..1)



### Association un à un (cas 1 → 1)

- Fusion des deux classes dans une seule table
- Permet d'éviter la jointure

---

---

---

---

---

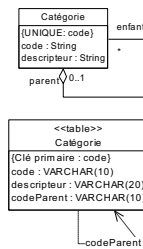
---

---

---

### Relation d'agrégation

- Comme une association normale




---

---

---

---

---

---

---

---

### Relation de composition

- Cas 1 → 1
  - ~ attribut complexe
- Mode SQL CASCADE
- Oracle8:
  - VARRAY, NESTED TABLE

---

---

---

---

---

---

---

---

## Relation de généralisation / spécialisation

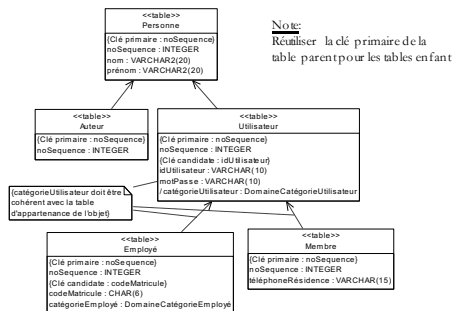
### ■ Choix possibles:

1. Délégation
2. Fusion
3. Concaténation

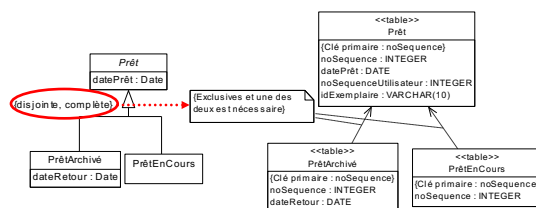
### ■ À considérer:

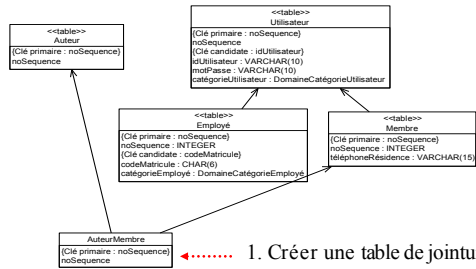
1. Spécialisation complète ou incomplète (extensible)
2. Spécialisation disjointe ou non (plusieurs sous-classes)

## Approche 1 : délégation

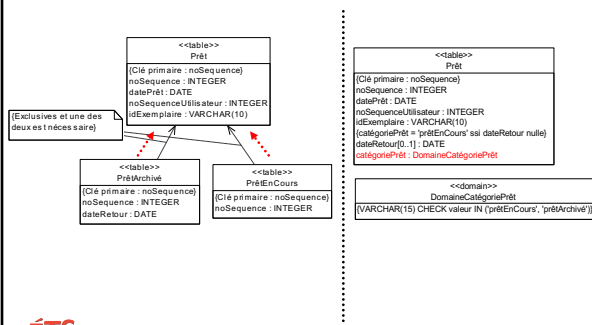


## Contrainte {disjointe, complète}

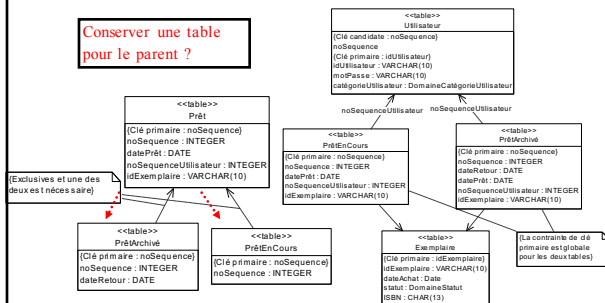




## Approche 2 : fusion



### Approche 3 : concaténation



## Comparaison

		Décomposition	Fusion	Concaténation
Disjointe	Oui	~	✓	~
	Non	✓	✗	✓
Complète	Oui	~	✓	~
	Non	✓	~	✓

---

---

---

---

---

---

---

---