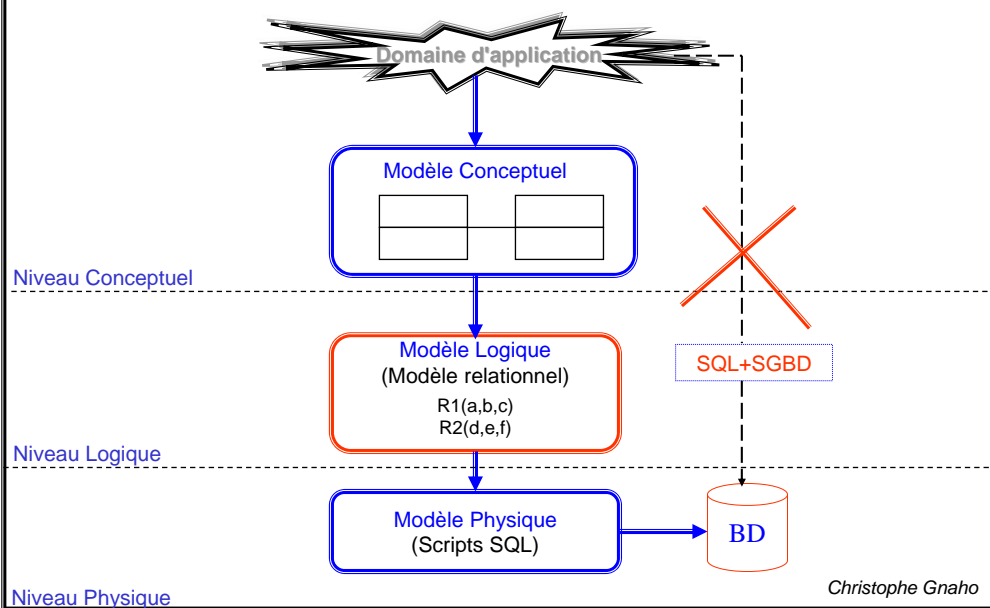


DEMARCHE DE CONCEPTION D'UNE BD RELATIONNELLE

Christophe Gnaho

DEMARCHE DE CONCEPTION



DEMARCHE DE CONCEPTION

Le niveau conceptuel

Ce niveau décrit une représentation abstraite de la base de données à l'aide de diagrammes de type entité-association ou de diagrammes de classes UML.

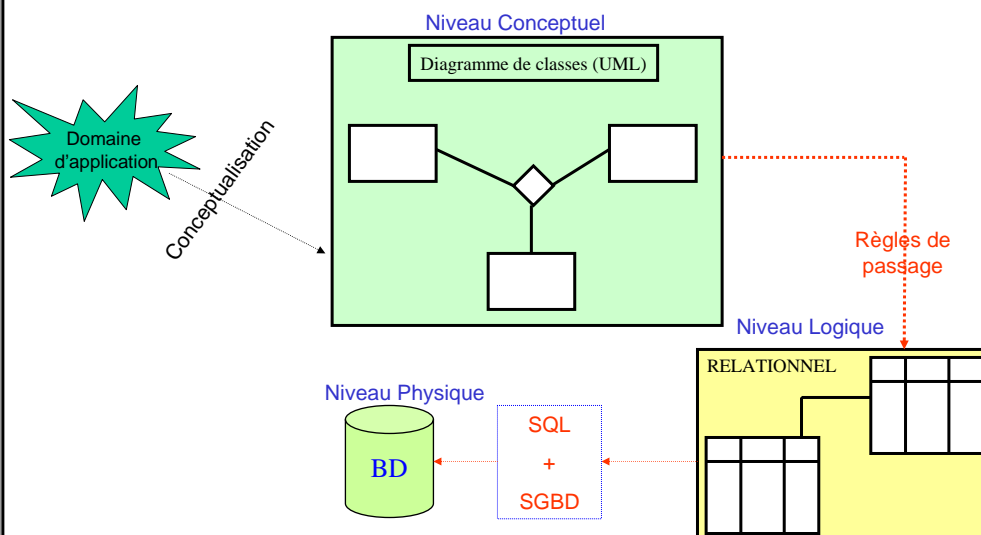
Le niveau logique

Le niveau logique détaille une représentation intermédiaire entre le niveau conceptuel et le niveau physique. Les diagrammes logiques peuvent être exprimés soit à l'aide d'une notation mathématique, soit sous forme de tabulaire, soit à l'aide d'un diagramme de classes UML (les classes auront le stéréotype particulier <<Table>>).

Le niveau Physique

Ce niveau concerne les structures de données à mettre en œuvre dans la base de données. Il traduit, à l'aide du langage SQL, le schéma logique. *Christophe Gnaho*

DEMARCHE DE CONCEPTION



Christophe Gnaho

LE MODELE RELATIONNEL (Rappel)

Christophe Gnaho

LES CONCEPTS DU MODELE RELATIONNEL (Relation)

UNE **RELATION** = UNE **TABLE** A DEUX DIMENSIONS

Schéma			Un attribut
Numéro Client	Nom Client	Ville Client	données
C1	Dulac	Paris	Un tuple
C2	Gobert	Toulon	
C3	Baudry	Toulouse	

Christophe Gnaho

LES CONCEPTS DU MODELE RELATIONNEL

(Domaine)

Le concept de **domaine** désigne l'ensemble des valeurs que peut prendre une donnée.

Exemple

Numéro Client sera définie sur le domaine	D1 = Entiers
Marque sera définie sur le domaine	D2 = {Renault, Peugeot, }
Couleur sera définie sur le domaine	D3 = {Blanche, Noire, ...}
Poids sera définie sur le domaine	D4 = {10 à 100 kilos, ...}
Salaires employés sera définie sur le domaine	D5 = Réels

Christophe Gnaho

LES CONCEPTS DU MODELE RELATIONNEL

(Attribut)

Le concept **d'attribut** représente une colonne d'une relation caractérisée par un nom.

Exemple

Numéro Client, Nom Client et Ville Client sont les attributs de la relation *Client*
Marque et Couleur sont les attributs de la relation *Voiture*.

Il est important de ne pas confondre la notion d'attribut avec celle de domaine.
Par exemple, l'attribut *Quantité_Produit* d'une relation *Produit* correspond au domaine *Entier*.

Le domaine représente l'ensemble des valeurs que peut prendre un attribut.

Le nom des attributs est unique dans une relation.

Christophe Gnaho

LES CONCEPTS DU MODELE RELATIONNEL

(Tuple)

Chaque ligne du tableau correspond à un *tuple* ou une *occurrence* de la relation.

Exemple

- C1 Dulac Paris.
- C2 Gobert Toulouse

Christophe Gnaho

LES CONCEPTS DU MODELE RELATIONNEL

(Clé primaire)

Chaque relation contient un attribut (ou un ensemble d'attributs) appelé *Clé primaire*, dont la valeur permet de distinguer de façon sûre une occurrence de toutes les autres.

Par exemple

Le *numéro d'immatriculation* d'une voiture permet de l'identifier, cet attribut est une clé de la relation Voiture.

Le *nom d'un client*, dans une relation Client, ne peut pas être une clé ; en effet, deux clients différents peuvent avoir le même nom.

Christophe Gnaho

LES CONCEPTS DU MODELE RELATIONNEL

(Schéma d'une relation)

Le *schéma d'une relation* désigne le nom de la relation suivi par la liste des attributs qui la composent et (éventuellement) par la définition de leurs domaines.

$R (A1 : D1, ..., An : Dn)$ où R est le nom de la relation, Ai les attributs et Di les domaines associés.

Par exemple

Client (NoClient : Entiers, Nom : Caractères, Ville : Caractères)

Client (NoClient, Nom , Ville)

Voiture (Immatriculation, Marque, Couleur)

Christophe Gnaho

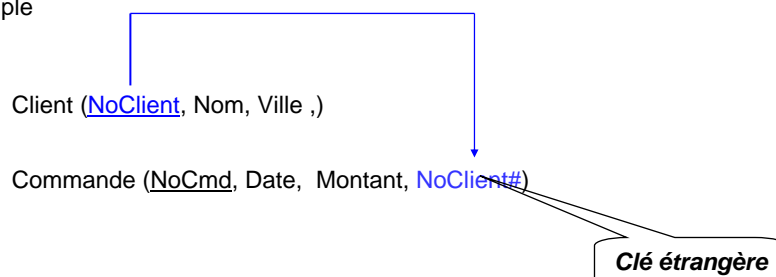
LE MODELE RELATIONNEL

(Règles d'intégrité structurelle)

Contrainte référentielle

Une *référence* (ou *clé étrangère*) est un attribut (ou un groupe d'attributs) dont les valeurs sont celles d'une clé d'une autre relation

Exemple



Christophe Gnaho

LE MODELE RELATIONNEL

(Règles d'intégrité structurelle)

Toute valeur de l'attribut *NoClient* dans la relation *Commande* doit se trouver dans la relation *Client*

Relation Client

NoClient	Nom	Ville
001	Dulac	Paris
002	Gomis	Madrid
003	Dubois	Londres

Clé primaire

Relation Commande

NoCmd	Date	Montant	NoClient
C1	21/11/07	10256	002
C2	10/11/07	20538	001
C3	2/012/07	45689	002
C4	4/12/07	10000	002
C5	30/11/07	2000	001

Clé étrangère

Christophe Gnaho

LE MODELE RELATIONNEL

(Règles d'intégrité structurelle)

Valeur nulle

La valeur nulle (*NULL*) est une valeur conventionnellement introduite dans une relation pour représenter une information inconnue ou inapplicable.

Ex: Employé (NoEmpl, Nom, Ville, Téléphone)

(11, Dulac, Nice, *NULL*)

ou

(15, Martin, Paris, 01 45 98 36 89)

Christophe Gnaho

LE MODELE RELATIONNEL

(Représentation graphique)

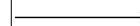
Client (noClient, nom, ville)

Client
<u>noClient</u>
nom
ville

Client (noClient, nom, ville ,)

Commande (noCmd, date, montant, noClient#)

Commande
<u>noCmd</u>
date
montant
noClient#



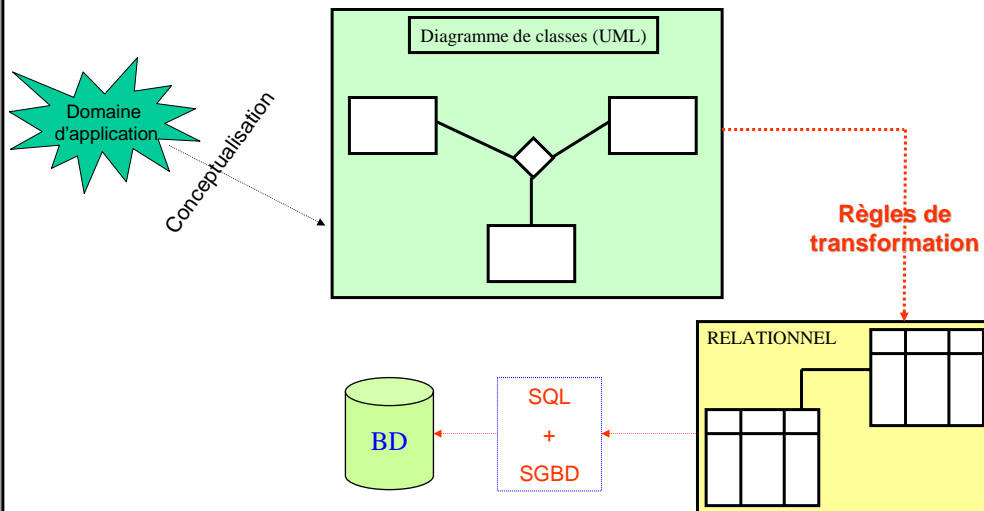
Client
<u>noClient</u>
nom
ville

Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD



Christophe Gnaho

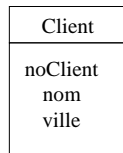
REGLES DE TRANSFORMATION DU MCO EN MLD

- ❖ Règles de transformation de la classe d'objet
- ❖ Règles de transformation des associations
- ❖ Règles de transformation de l'héritage
- ❖ Règles de transformation des contraintes en langage SQL2

Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Transformation des Classes d'objets



- la classe se transforme en une table (relation)
- les attributs de la classe deviennent des attributs de la table
- choisir un attribut (ou groupe d'attribut) de la classe pouvant jouer le rôle de clé primaire; si aucun attribut ne convient, il faut en ajouter un à la table

TClient (noClient, nom, ville)

<u>noClient</u>	nom	ville
001	Dulac	Paris
002	Gomis	Madrid
003	Dubois	Londres

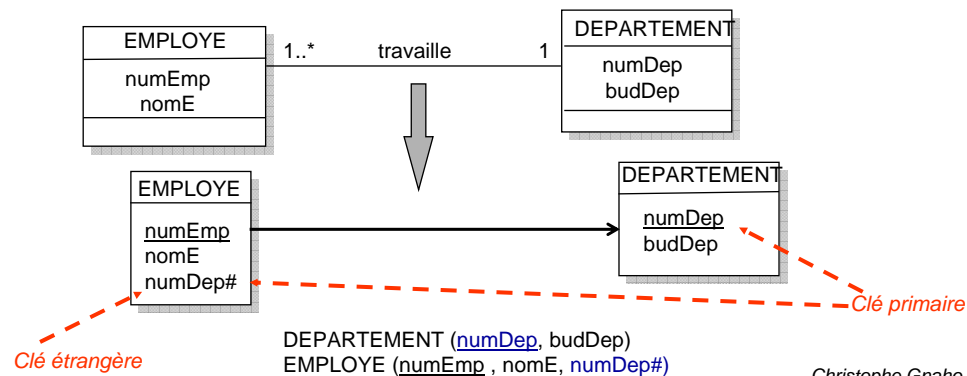
Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Transformation des associations

Cas Association binaire de type **plusieurs** [(1..*) ou (0..*)] à un [-(1..1) ou (0..1)]

ajouter un attribut de type clé étrangère dans la relation à multiplicité plusieurs de l'association.

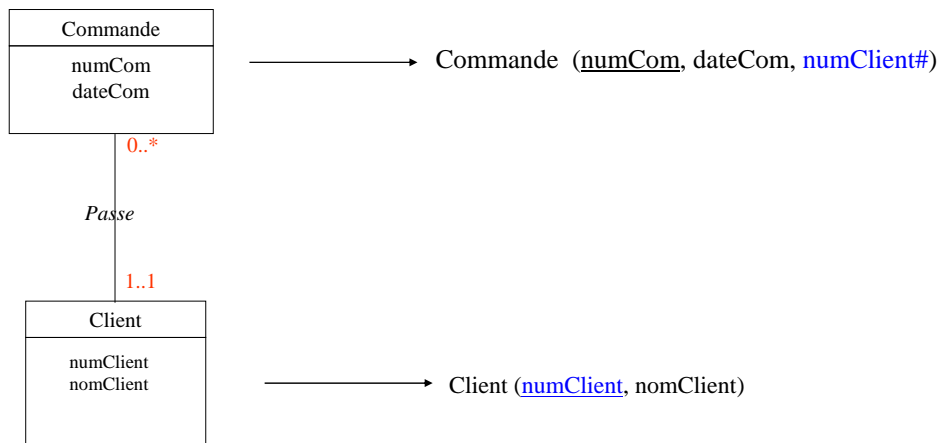


Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Diagramme de classes

Modèle Logique



Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Transformation des associations

Cas d'une liaison plusieurs (0..* ou 1..*) à plusieurs (0..* ou 1..*)

L'association ou la classe-association devient une relation. La clé primaire est composée des clés primaires des relations obtenues.

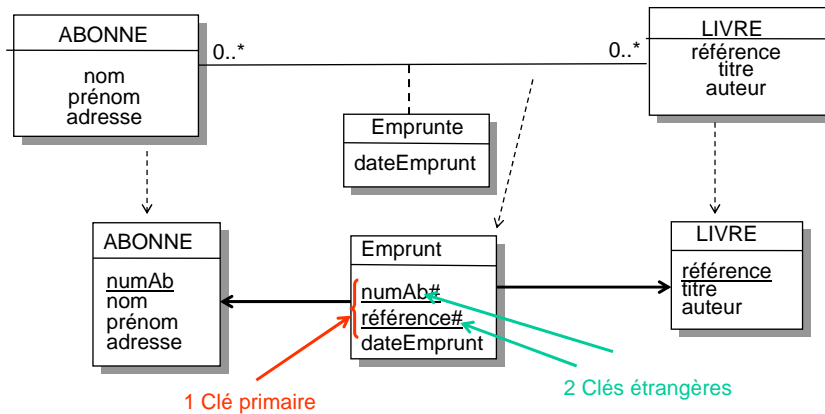
Les éventuelles attributs de la classe-association deviennent des attributs de la nouvelle relation.

Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

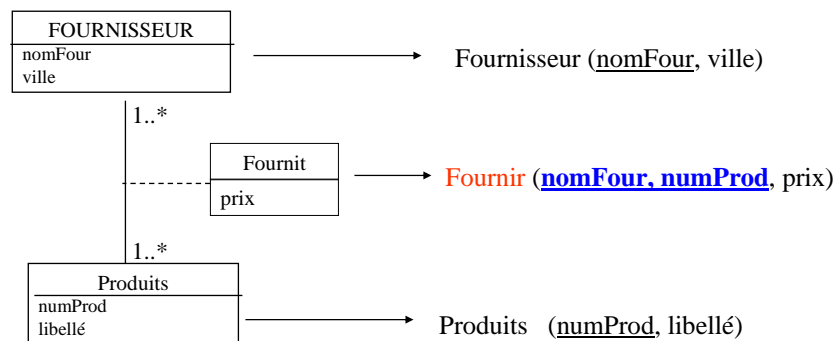
Transformation des associations

Cas d'une liaison plusieurs (0..* ou 1..*) à plusieurs (0..* ou 1..*)



Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD



Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Transformation des associations

Cas Association binaire de type un [(0..1) ou (1..1)] à un [(0..1) ou (1..1)]

Il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un.

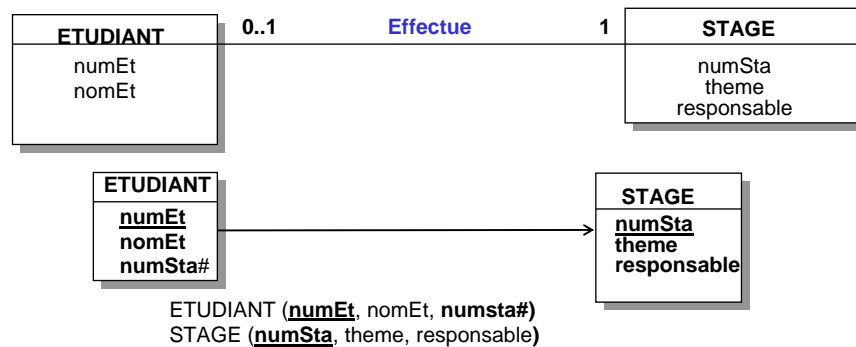
Dans le cas où les deux multiplicités minimales sont à 1, il est préférable de fusionner les deux entités

Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Transformation des associations

Cas Association binaire de type un [(0..1) ou (1..1)] à un [-(0..1) ou (1..1)]



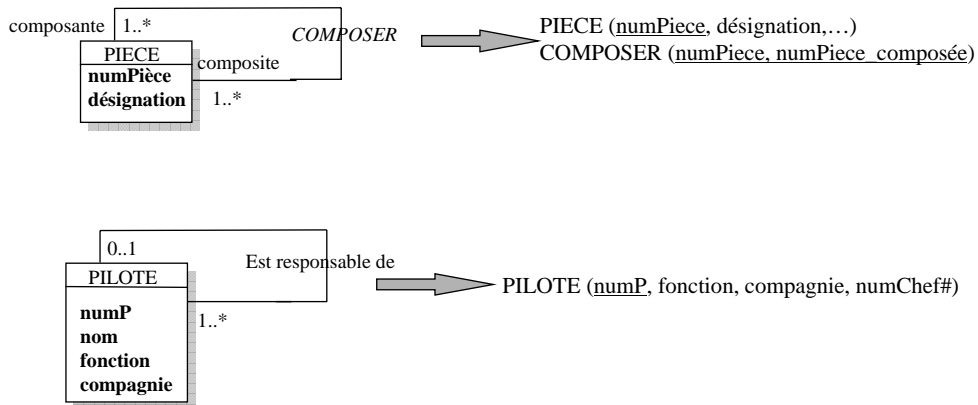
Ceci présuppose que le stage existe déjà à la création d'un étudiant

Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Transformation des associations

Association Reflexive

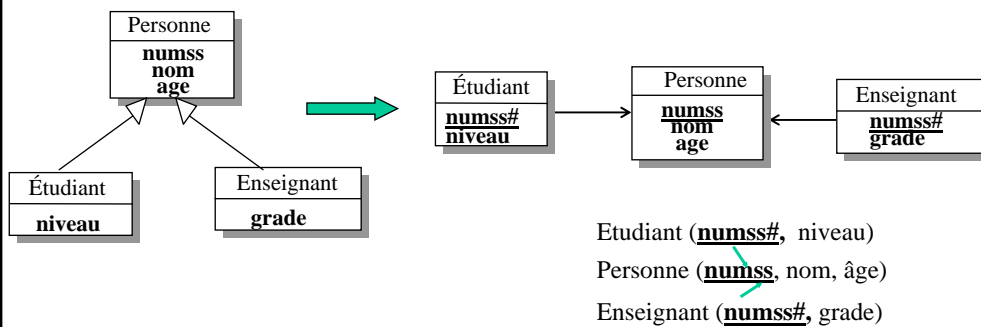


Christophe Gnaho

REGLES DE TRANSFORMATION DU MCO EN MLD

Cas de l'héritage

Transformer chaque sous classe en une relation. La clé primaire de la superclasse devient clé primaire de chaque sous classe



Christophe Gnaho

TRADUCTION D'UN MLD EN SQL

Christophe Gnaho

TRADUCTION D'UN MLD RELATIONNEL EN LANGAGE SQL

Exemple

```
CREATE TABLE stage
(nstage VARCHAR(4),
entreprise VARCHAR(30)
CONSTRAINT pk_sage PRIMARY KEY (nstage));

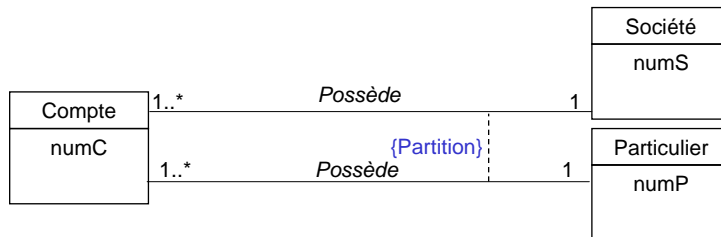
CREATE TABLE etudiant
(netu VARCHAR(2),
nometu VARCHAR (30),
nstage VARCHAR(4),
CONSTRAINT pk_etudiant PRIMARY KEY (netu)
CONSTRAINT fk_etudiant FOREIGN KEY (nstage) REFERENCES stage(nstage),
CONSTRAINT nn_etudiant_nstage
CHECK (nstage is NOT NULL),
CONSTRAINT unique_etudiant_nstage Unique(nstage));
```

Christophe Gnaho

TRADUCTION D'UN MLD RELATIONNEL EN LANGAGE SQL

Expression de la contrainte de partition

Tous les objets d'une classe participent à l'une des deux associations mais pas aux deux, ni à aucune de deux.



```
CONSTRAINT ck_compte_partition
CHECK ((numS# is NOT NULL OR numP# is NOT NULL)
AND
NOT (numS# is NOT NULL AND numP# is NOT NULL))
```

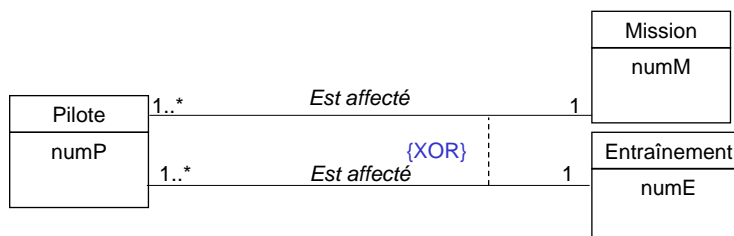
Christophe Gnaho

TRADUCTION D'UN MLD RELATIONNEL EN LANGAGE SQL

Expression de la contrainte d'exclusion

Tous les objets d'une classe peuvent participer à l'une des deux associations, mais pas aux deux à la fois.

On peut imaginer par exemple un pilote au repos. Si un pilote est affecté à une mission, alors il ne peut être affecté à un vol d'entraînement et réciproquement.



```
CONSTRAINT ck_compte_exclusion CHECK (numM# is NULL OR numE# is NULL);
```

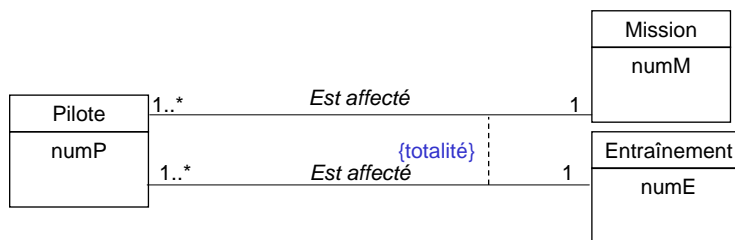
Christophe Gnaho

TRADUCTION D'UN MLD RELATIONNEL EN LANGAGE SQL

Expression de la contrainte de totalité

Tous les objets d'une classe participent au moins à une association

On peut imaginer par exemple qu'un pilote soit affecté à la fois à une mission et à un vol d'entraînement, et tous les pilotes participent à au moins une mission.



```
CONSTRAINT ck_compte_totalite CHECK (NOT (numM# is NULL AND numE# is NULL));
```

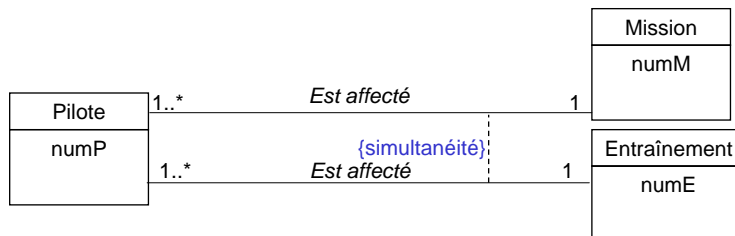
Christophe Gnaho

TRADUCTION D'UN MLD RELATIONNEL EN LANGAGE SQL

Expression de la contrainte de simultanété

Si un objet d'une classe participe à l'une des deux associations, alors elle participe également à l'autre

Un pilote peut être affecté à la fois à une mission et à un vol d'entraînement. Il peut également n'être affecté à aucune mission.



```
CONSTRAINT ck_compte_simultanété CHECK ((numM# is NULL AND numE# is NULL) OR (numM# is NOT NULL AND numE# is NOT NULL) );
```

Christophe Gnaho