

Intégrité sémantique avancée dans les BD relationnelles : de la spécification en ER à l'implémentation



Bernard ESPINASSE
Professeur à Aix-Marseille Université (AMU)
Ecole Polytechnique Universitaire de Marseille



Février 2013

- 1 – Dépendances fonctionnelles dans les relations
- 2 – Contraintes d'intégrité (CI) générales
- 4 – CI générales associées ou non à un événement
- 3 – CI de stabilité de propriétés
- 5 – CI de participation à d'une entité à des relations
- 6 – Types et sous types d'entités en spécialisation/généralisation

Plan

1. Dépendances fonctionnelles dans les relations
2. Contraintes d'intégrité (CI) générales
3. CI générales associées ou non à un événement
4. CI de stabilité de propriétés
5. CI de participation à d'une entité à des relations
 - a. Exclusion de participation
 - b. Simultanéité de participation
 - c. Totalité de participation
 - d. Partition de participation – OU exclusif
 - e. Inclusion de participation
 - f. Contraintes d'inclusion de relations sur d'autres relations
 - g. Contraintes d'exclusion sur participation de plusieurs entités à plusieurs relations
6. Types et sous types d'entités en spécialisation/généralisation

Dépendances fonctionnelles dans relations (conceptuelle) n-aire dans les modèles Entité-Relation (ER)

2 cas de figure :

1. Cas de dépendances fonctionnelles **n'englobant pas la totalité** de la collection de la relation conceptuelle :

=> il est conseillé de **décomposer la relation n-aire** (en la projetant sur n relations binaires)

2. Cas de dépendances fonctionnelles englobant la totalité de la collection de la relation :

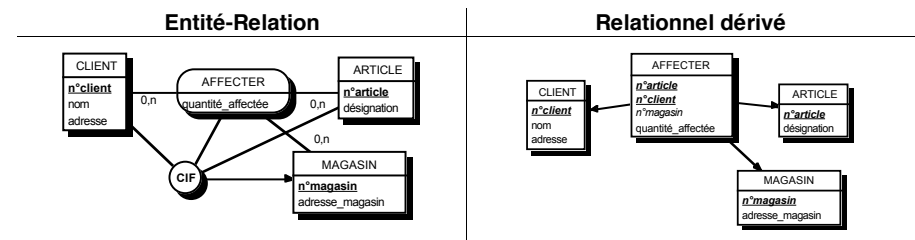
=> la spécification de **clés primaires** et **clés étrangères** dans la table associée à la relation conceptuelle.

Dépendances fonctionnelles dans relation n-aire (2)

Contrainte à exprimer :

« Un client ne s'approvisionne pour un article donné que dans un magasin donné »

Soit : Client X Article -> Magasin :



Schémas relationnels associés :

- Table CLIENT (n°client, nom, adresse) ;
- Table ARTICLE (n°article, désignation) ;
- Table MAGASIN (n°magasin, adresse_magasin) ;
- Table AFFECTER (n°client, n°article, n°magasin, quantité_affectée)

Dépendances fonctionnelles dans relation n-aire (3)

la dépendance fonctionnelle est traduite ici par la définition, dans la table Affecter :

- des 3 clés étrangères, cli_num, art_num, mag_num, et
- de la clé primaire composée (cli_num, art_num).

La création de la table Affectée est :

```
CREATE TABLE Affecter
(cli_num CHAR(5) NOT NULL,
(art_num CHAR(5) NOT NULL,
(mag_num CHAR(5) NOT NULL...)
PRIMARY KEY (cli_num, art_num)
CONSTRAINT PRIMARY_AFFECTER
FOREIGN KEY (cli_num) REFERENCES Client ;
CONSTRAINT AFFECTER_CLIENT
FOREIGN KEY (art_num) REFERENCES Article ;
CONSTRAINT AFFECTER_ARTICLE
FOREIGN KEY (mag_num) REFERENCES Magasin ;
CONSTRAINT AFFECTER_MAGASIN
```

Contraintes d'intégrité (CI) générales

- Généralement des **contraintes sur la prise de valeurs de propriétés** en fonction de valeurs déjà prises par d'autres propriétés de la même ou d'autres entités ou relations du MCD
- Ces contraintes se traduisent par des **CI générales sur les attributs** du MLD relationnel
- **2 grands types de CI générale :**
 - **Contraintes non associées à un événement**
 - assez bien prises en compte au travers de la déclaration d'une clause CHECK avec ou sans création d'une ASSERTION.
 - **Contraintes associées à un événement :**
 - souvent assez complexes, doivent être exécutées dès l'apparition d'un événement donné
 - le programmeur peut déclencher la CI dès la survenance d'un événement par la création d'un **TRIGGER**.

CI générales non associées à un événement

Soit la contrainte générale suivante :

« le prix unitaire de l'article lcd_pu de la table LIGNE_CMD ne peut être remis à plus de 25% du prix de vente de l'article (prix catalogue) art_pv défini dans la table ARTICLE »

1. Traitement par la clause **CHECK** déclarée dans la table LIGNE_CMD :

```
CREATE TABLE Ligne_cmd
...
CHECK (lcd_pu >= 0.75 *
(SELECT art_pv FROM Article WHERE art_num = cmd_art))
CONSTRAINT PU_LIGNCDE);
```

2. Traitement par une clause **ASSERTION** non rattachée à une table en particulier (souvent utilisée pour spécifier une contrainte d'intégrité définie sur plusieurs tables) :

```
...
CREATE ASSERTION PU_PV
CHECK (NOT EXISTS
(SELECT * FROM Ligne_cmd WHERE lcd_pu < 0.75 *
(SELECT art_pv FROM Article WHERE art_num = lcd_art )));
```

CI générales associées à un événement

Soit la contrainte d'intégrité générale suivante :

« lors d'un ajout d'une nouvelle ligne de commande à une commande donnée, le montant de cette commande, l'attribut cmd_montant de la table COMMANDE, doit être égal à la somme des montants associés à chacune des lignes de commande de la commande, montant calculé comme le produit de la quantité commandée d'article par un prix unitaire (lcd_qte * lcd_pu) »

Dans le SGBD Oracle :

```
CREATE TRIGGER Calcul_montant_si_ajout
AFTER INSERT ON ligne_cmd
BEGIN
UPDATE Commande
SET cmd_montant = cmd_montant + :new.lcd_qte * :new.lcd_pu
WHERE cmd.num = :new.lcd_cmd;
END;
```

- instructions comprises entre le BEGIN et le END sont exprimées en PL/SQL
- variable :new associée à un nouveau tuple inséré (INSERT) ou mis à jour (UPDATE)
- variable :old associée à l'ancien tuple mis à jour (UPDATE) ou supprimé (DELETE).

CI de stabilité de propriété

Contrainte à traiter :

Exemple : stabilité de l'attribut `cmd_date` de la table `COMMANDE` (correspondant à la propriété stable date de l'entité `COMMANDE`) :

Les **TRIGGERS** permettent aussi de traiter la plupart de ces contraintes de stabilité

Dans le SGBD Oracle traitement de cette contrainte de stabilité par un **TRIGGER** sur la table `Commande` :

```
CREATE TRIGGER Stable_cmd_date
AFTER INSERT ON cmd_date OF Commande
ON EACH ROW
BEGIN
    IF :new.cmd_date != :old.cmd_date
        raise_application_error (-20001, 'Date non modifiable ');
    END IF;
END;
```

Remarque : "!=" exprime la négation dans le PL-SQL d'Oracle et que `raise_application_error` permet d'affecter un code d'erreur pouvant être ensuite traité ainsi qu'un message à l'utilisateur.

CI inter-relation sur la participation d'une entité à plusieurs relations

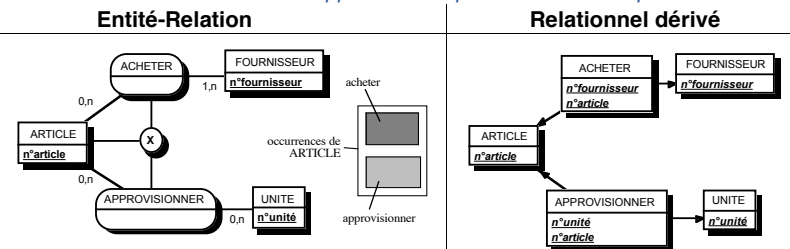
Différentes CI inter-relation :

- Exclusion de participation
- Simultanéité de participation
- Totalité de participation
- Partition de participation – OU exclusif
- Inclusion de participation
- Contraintes d'inclusion de relations sur d'autres relations
- Contraintes d'exclusion sur participation de plusieurs entités à plusieurs relations

Exclusion de participation d'une entité à plusieurs relations (1)

Contrainte à traiter :

« relations `Acheter` et `Approvisionnement` mutuellement exclusives : un article donné ne peut être à la fois acheté à un fournisseur et approvisionné par d'une unité de production »



- Table `ARTICLE` (`n°article`, ...);
- Table `ACHERER` (`n°article`, `n°fournisseur`, ...);
- Table `FOURNISSEUR` (`n°fournisseur`, ...);
- Table `APPROVISIONNER` (`n°article`, `n°unité`);
- Table `UNITÉ` (`n°unité`...).

Exclusion de participation d'une entité à plusieurs relations (2)

1 - Exclusion de participation d'une entité à plusieurs relations par **ASSERTION** :

```
CREATE ASSERTION CX
CHECK (NOT EXISTS
(SELECT * FROM Acheter WHERE n°article IN
(SELECT n°article FROM Approvisionnement))
UNION
(SELECT * FROM Approvisionner WHERE n°article IN
(SELECT n°article FROM Acheter)));
```

Exclusion de participation d'une entité à plusieurs relations (3)

2 - Exclusion de participation d'une entité à plusieurs relations par 2 **TRIGGERS** :

Trigger 1 sur table Acheter :

```
CREATE TRIGGER Exclusion_acheter_approvisionner
BEFORE INSERT ON Acheter
ON EACH ROW
DECLARE
    nb_article_approvisionner number;
BEGIN
    SELECT COUNT(*) INTO nb_article_approvisionner FROM Approvisionner
    WHERE n°article = :new.n°article;
    IF nb_article_approvisionner > 0 THEN
        raise_application_error (-20002, 'Article N° ', :new.n°article,
        ' est un article d approvisionnement' );
    END IF;
END;
```

Exclusion de participation d'une entité à plusieurs relations (4)

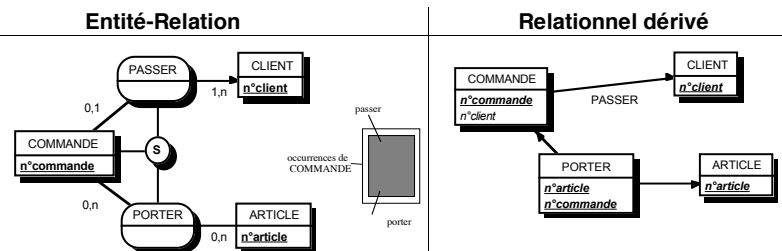
Trigger 2 sur table Approvisionner :

```
CREATE TRIGGER Exclusion_approvisionner_acheter
BEFORE INSERT ON Approvisionner
ON EACH ROW
DECLARE
    nb_article_acheter number;
BEGIN
    SELECT COUNT(*) INTO nb_article_acheter FROM Acheter
    WHERE n°article = :new.n°article;
    IF nb_article_acheter > 0 THEN
        raise_application_error (-20003, 'Article N° ', :new.n°article,
        ' est un article à acheter' );
    END IF;
END;
```

Simultanéité de participation d'une entité à plusieurs relations (1)

Contrainte à traiter :

« une Commande portant sur des Articles est obligatoirement destinée à un Client »



- Table COMMANDE (n°commande, n°client, ...);
- Table CLIENT (n°client, ...);
- Table ARTICLE (n°article, ...);
- Table PORTER (n° article, n°commande).

Simultanéité de participation d'une entité à plusieurs relations (2)

1 - Simultanéité de participation d'une entité à plusieurs relations traitée par une **ASSERTION** :

```
CREATE ASSERTION CS
CHECK (NOT EXISTS
(SELECT n°commande FROM Commande
WHERE n°commande NOT IN
(SELECT n°commande FROM Porter)
AND n°client IS NOT NULL
UNION
(SELECT n°commande FROM Porter
WHERE n°commande NOT IN
(SELECT n°commande FROM Commande)
WHERE n°client IS NOT NULL
```

Simultanéité de participation d'une entité à plusieurs relations (3)

1 - Simultanéité de participation d'une entité à plusieurs relations par 2 **TRIGGERS** :

Trigger 1 sur table Commande :

```
CREATE TRIGGER Simultaneité_Passer_Porter
BEFORE INSERT OR UPDATE n°client ON Commande
ON EACH ROW
WHEN new.n°client IS NOT NULL
DECLARE
    nb_article_passer    number;
BEGIN
    SELECT COUNT(*) INTO nb_article_passer FROM Passer
    WHERE n°commande = :new.n°commande;
    IF nb_article_passer = 0
        raise_application_error (-20004, 'La commande n'est pas porteuse d'articles' );
    END IF;
END;
```

Simultanéité de participation d'une entité à plusieurs relations (4)

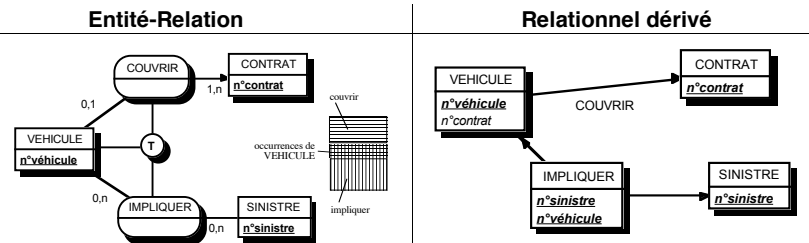
Trigger 2 sur table Porter :

```
CREATE TRIGGER Simultaneité_Porter_Passer
BEFORE INSERT ON Porter
ON EACH ROW
WHEN new.n°client IS NOT NULL
DECLARE
    nb_article_porter    number;
BEGIN
    SELECT COUNT(*) INTO nb_article_porter FROM Commande
    WHERE n°client IS NOT NULL;
    IF nb_article_porter = 0
        raise_application_error (-20005, 'La commande n'est pas encore
        passée par un client' );
    END IF;
END;
```

Totalité de participation d'une entité à plusieurs relations (1)

Contrainte à traiter :

« tout Véhicule est reliée soit à Contrat par la relation Couvrir, soit à Sinistre par la relation Impliquer, soit les deux »



- Table VEHICULE (n°véhicule, n° contrat, ...);
- Table CONTRAT (n°contrat, ...);
- Table SINISTRE (n°sinistre, ...);
- Table IMPLIQUER (n° sinistre, n°véhicule).

Totalité de participation d'une entité à plusieurs relations (2)

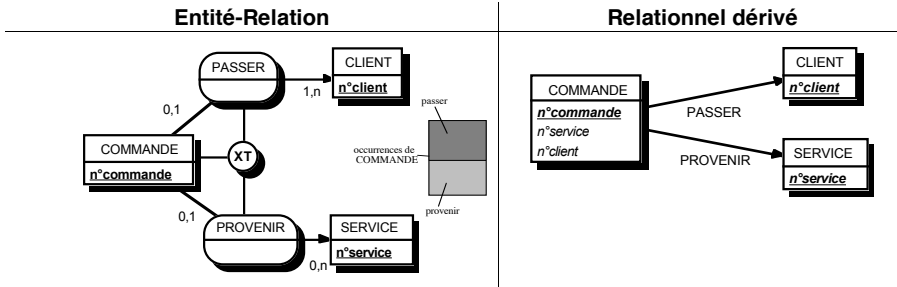
Totalité de participation d'une entité à plusieurs relations traitée par une **ASSERTION** :

```
CREATE ASSERTION CT
CHECK (NOT EXISTS
(SELECT n°véhicule FROM Véhicule
WHERE n°véhicule NOT IN
(SELECT n°véhicule FROM Véhicule
WHERE n°contrat IS NOT NULL)
UNION
SELECT n°véhicule FROM Impliquer));
```

Partition de participation d'une entité à plusieurs relations – OU exclusif (1)

Contrainte à traiter :

« une Commande est, soit destinée à un Client, soit à un Service interne de l'entreprise, mais pas aux 2 à la fois, on a ici une contrainte d'exclusion de participation de l'entité Commande aux relations Passer et Provenir »



- Table COMMANDE (n°commande, n°service, n°client) ;
- Table CLIENT (n°client, ...).
- Table SERVICE (n°service, ...).

Partition de participation d'une entité à plusieurs relations – OU exclusif (2)

Partition de participation d'une entité à plusieurs relations traitée par une **ASSERTION** :

CREATE ASSERTION CXT

CHECK (NOT EXISTS

(SELECT n°commande FROM Commande

WHERE n°commande NOT IN

(SELECT n°commande FROM Commande

WHERE n°client IS NOT NULL)

UNION

SELECT n°commande FROM Commande));

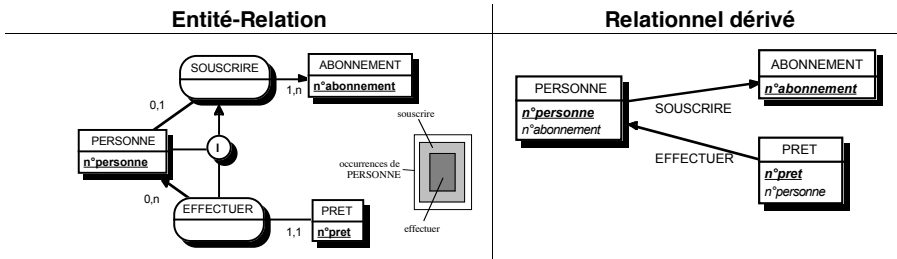
WHERE n°service IS NOT NULL)

Inclusion de participation d'une entité à plusieurs relations (1)

Contrainte à traiter :

« toute personne qui effectue un prêt doit avoir souscrit un abonnement »

Par rapport à l'entité PERSONNE, la relation EFFECTUER est incluse dans la relation SOUSCRIRE



- Table PERSONNE (n°personne, n°abonnement, ...)
- Table ABONNEMENT (n°abonnement, ...)
- Table PRET (n°prêt, n°personne, ...).

Inclusion de participation d'une entité à plusieurs relations (2)

1 - Inclusion de participation d'une entité à plusieurs relations traitée par une **ASSERTION** :

CREATE ASSERTION I

CHECK (NOT EXISTS

(SELECT DISTINCT n°personne FROM Personne A

WHERE NOT EXISTS

(SELECT DISTINCT n°personne FROM Prêt B

WHERE (A.n°personne= B.n°personne))

AND n°abonnement IS NOT NULL));

Inclusion de participation d'une entité à plusieurs relations (3)

2 - Inclusion de participation d'une entité à plusieurs relations par 2 TRIGGERS :

Trigger 1 sur la table Pret :

```
CREATE TRIGGER Inclusion_Effectuer_Souscrire
BEFORE INSERT ON Pret
ON EACH ROW
WHEN new.n°personne IS NOT NULL
DECLARE
    nb_abonnement    number;
BEGIN
    SELECT COUNT(*) INTO nb_abonnement FROM Personne
    WHERE n°personne = :new.n°personne;
    IF nb_abonnement = 0 THEN
        raise_application_error (-20006, 'Un abonnement n'a pas été souscrit' );
    END IF;
END;
```

Inclusion de participation d'une entité à plusieurs relations (4)

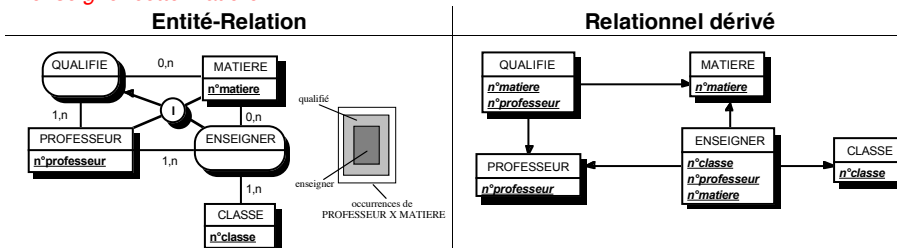
Trigger 2 sur la table Porter :

```
CREATE TRIGGER Simultaneité_Porter_Passer
BEFORE INSERT ON Porter
ON EACH ROW
WHEN new.n°client IS NOT NULL
DECLARE
    nb_article_porter    number;
BEGIN
    SELECT COUNT(*) INTO nb_article_porter FROM Commande
    WHERE n°client IS NOT NULL;
    IF nb_article_porter = 0
        raise_application_error (-20005, 'La commande n'est pas encore
        passée par un client' );
    END IF;
END;
```

Contraintes d'inclusion de relations sur d'autres relations

Contrainte à traiter :

« pour qu'un professeur enseigne une matière à une classe donnée, il faut qu'il sache enseigner cette matière »



- Table PROFESSEUR (n°professeur,...) ;
- Table MATIÈRE (n°matière,...) ;
- Table CLASSE (n°classe,...) ;
- Table QUALIFIE (n° matière, n°professeur) ;
- Table ENSEIGNER (n°classe, n°professeur, n°matière)

Contraintes d'inclusion de relations sur d'autres relations

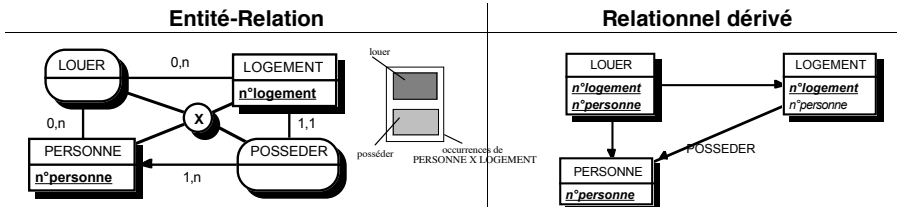
Contrainte d'inclusion de relations sur d'autres relations traité par un TRIGGER sur la table Enseigner :

```
CREATE TRIGGER Inclusion_Enseigner_Qualifier
BEFORE INSERT ON Enseigner
ON EACH ROW
DECLARE
    est_qualifié    number;
BEGIN
    SELECT COUNT(*) INTO est_qualifié FROM Qualifier
    WHERE n°professeur = :new.n°professeur
    AND n°matière = :new.n°matière;
    IF est_qualifié = 0 THEN
        raise_application_error (-20007, 'Ce professeur n'est pas
        qualifié pour cette matière' );
    END IF;
END;
```

Contraintes d'exclusion sur participation de plusieurs entités à plusieurs relations

Contrainte à traiter :

« une personne ne peut pas être locataire et propriétaire d'un même logement »



Schémas relationnels associés :

- Table LOUER (n° logement, n° personne) ;
- Table LOGEMENT (n°logement, n°personne,...) ;
- Table PERSONNE (n°personne, ...)

Contraintes d'exclusion sur participation de plusieurs entités à plusieurs relations

Contrainte d'exclusion de relations sur d'autres relations traité par 2 **TRIGGER** :

Trigger 1 sur la table Louer :

```
CREATE TRIGGER Exclusion_Louer_Logement
BEFORE INSERT ON Louer
ON EACH ROW
DECLARE
    possède_déjà number;
BEGIN
    SELECT COUNT(*) INTO possède_déjà FROM Louer
    WHERE n°personne = :new.n°personne
    AND n°logement = :new.n°logement ;
    IF possède_déjà > 0 THEN
        raise_application_error (-20008, 'Cette personne est déjà
        propriétaire de ce logement' );
    END IF;
END;
```

Contraintes d'exclusion sur participation de plusieurs entités à plusieurs relations

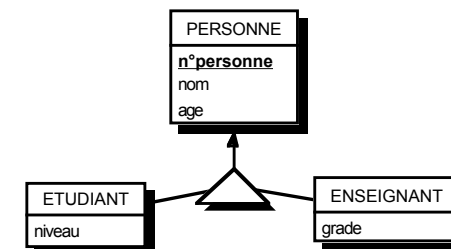
Contrainte d'inclusion de relations sur d'autres relations traité par 2 **TRIGGER** :

Trigger 2 sur la table Logement :

```
CREATE TRIGGER Exclusion_Logement_Louer
BEFORE INSERT ON Logement
ON EACH ROW
    WHEN new.n°personne IS NOT NULL
DECLARE
    nb_location number;
BEGIN
    SELECT COUNT(*) INTO nb_location FROM Louer
    WHERE n°personne = :new.n°personne
    IF nb_location > 0 THEN
        raise_application_error (-20009, 'La personne n° ', new.n°personne,
        ' est déjà locataire' );
    END IF;
END;
```

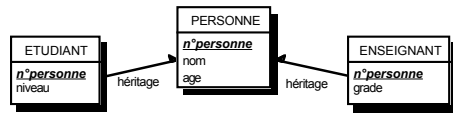
Prise en compte des sous-types de spécialisation d'entités (1)

Entité-Relation :



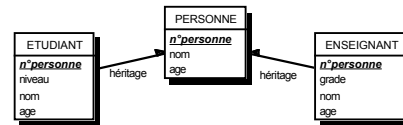
Prise en compte des sous-types de spécialisation d'entités (2)

Solution 1



- On exprime les sous-types par des tables spécifiques, correspondant à des relations (0,1)-(1,1).
- Il y a migration de l'identifiant du sur-type dans les sous-types

Solution 2



- On duplique la totalité du contenu du sur-type dans les sous-types associés et on supprime le sur-type.
- Pas conseillée dans le cas où il existe, dans le modèle ER des relations portant sur le sur-type

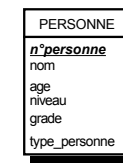
Prise en compte des sous-types de spécialisation d'entités (3)

Solution 3



- On duplique la totalité du contenu du sur-type dans les sous-types associés et on supprime le sur-type.
- Solution pas conseillée dans le cas où il existe, dans le modèle ER des relations portant sur le sur-type

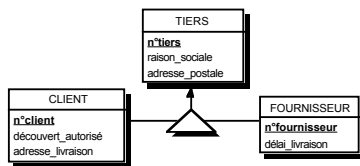
Solution 4



- On transfère la totalité des propriétés des sous-types dans la table PERSONNE correspondant au sur-type.
- On exprime ensuite les sous-types ETUDIANT et ENSEIGNANT par des vues de la table PERSONNE globale, rassemblant l'ensemble des propriétés spécifiques aux sous-types

Prise en compte de généralisation de sous-types d'entités

Entité-relation



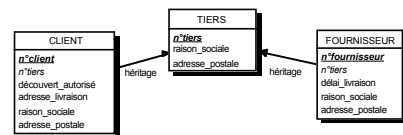
Schémas relationnels associés :

- Table TIERS (n°tiers, raison_sociale, adresse_postale) ;
- Table CLIENT (n°client, n°tiers, découvert_authorized, adresse_livraison, raison_sociale, adresse_postale) ;
- Table FOURNISSEUR (n°fournisseur, n°tiers, délai_livraison, raison_sociale, adresse_postale).

Remarques :

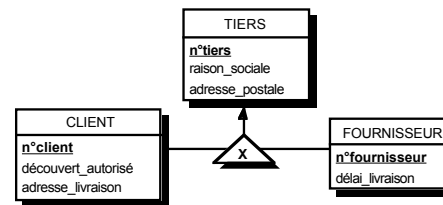
- On est dans le cas d'une relation (0,1)-(1,1)
- Dans CLIENTS et FOURNISSEURS l'attribut n°tiers sera déclaré comme une clé candidate ou clé alternative et sera déclaré dans le schéma avec une clause UNIQUE.

Relationnel dérivé



Prise en compte de généralisation de sous-types d'entités avec exclusion

Entité-relation



Schémas relationnels associés :

- Table TIERS (n° tiers, n° client, n° fournisseur, raison sociale, adresse,...) ;
- Table CLIENT (n° client, n° tiers, découvert autorisé, adresse de livraison) ;
- Table FOURNISSEUR (n° fournisseur, n° tiers, délai de livraison).

Remarque : La contrainte d'exclusion implémentée par triggers spécifiques.

Relationnel dérivé

