

Conception des systèmes d'information

Méthodes & Outils

Extensions du MEA :
Contraintes d'intégrité

Christian FISCHER – © Octobre 2002

1.A. Les Contraintes d'intégrité

Les contraintes d'intégrité expriment certaines caractéristiques sémantiques du système d'information et permettent de s'assurer de la cohérence des informations.

Au niveau conceptuel, les contraintes d'intégrité permettent :

- d'affiner la description, l'image donnée du monde réel (plus sémantique). Toute contrainte d'intégrité est porteuse de sens,
- d'exprimer des contraintes sur l'état actuel du système d'information et sur son évolution dans le temps (vie du système).

Remarque préalable : aucune des contraintes présentées dans cette partie n'est prise en charge par AMC*Designor.

1.A.1 Les contraintes entre valeurs de propriétés

Ces contraintes n'apparaissent généralement pas dans la représentation graphique de la modélisation des données. Elles visent à assurer que la modélisation des données possède et conserve un sens et qu'il n'est pas possible, éventuellement au travers de manipulations intempestives, d'aboutir à une représentation incohérente de la réalité.

Ces contraintes peuvent être réparties en deux classes principales suivant les éléments de la modélisation des données et le type de règle qu'elles font intervenir.

Les deux classes de contraintes principales sont les suivantes :

1. **Les contraintes statiques** : elles concernent l'état du système d'information indépendamment du temps.
Par exemple la contrainte "*le salaire d'un employé doit être supérieur à un seuil constitué par le SMIC*" est statique.
2. **Les contraintes de transition (ou dynamiques)** concernent quant à elles l'évolution dans le temps des données du système d'information. Toute valeur d'une propriété dépendra d'une valeur prise précédemment.
Par exemple la contrainte "*le salaire d'un employé ne peut que croître*" est dynamique, toute nouvelle valeur du salaire doit donc être supérieure à l'ancienne.

A l'intérieur de cette première classification, est introduite une classification de second niveau, venant se combiner à la précédente. Elle porte sur un nombre d'occurrences impliquées dans l'expression de la contrainte. Les deux sous-classes de contraintes sont les suivantes :

1. **Les contraintes simples** font intervenir une ou plusieurs occurrences dont on compare simplement certaines des composantes comme par exemple le fait d'exprimer que "*la date de prestation d'un sinistre doit être postérieure ou égale à la date d'effet du contrat considéré*".

2. **Les contraintes ensemblistes** font intervenir un ensemble d'occurrences sans comporter de comparaison entre les composantes. Elles font appel à des opérateurs du type SOM, MOY, MIN, MAX, etc., comme par exemple le fait d'exprimer que *"la somme des salaires des employés doit être inférieure à 1 Million de Francs"*.

Les critères, statique-transition d'une part et simple-ensembliste de l'autre, peuvent être croisés. Le tableau ci-dessous illustre le croisement possible entre les deux classes de critères :

Contraintes	STATIQUES	TRANSITION
SIMPLE	Le salaire des employés doit être supérieur au SMIC	Le nouveau salaire d'un employé doit être supérieur à l'ancien
ENSEMBLISTE	La moyenne des salaires des employés doit être Inférieure à 15 000 F	La moyenne des salaires des employés ne doit pas dépasser de plus de 5 % la moyenne des anciens salaires

L'expression de ces différentes contraintes pourra également s'exprimer à l'aide du langage SQL sous la forme de triggers.

1.A.2 Les contraintes ensemblistes entre associations

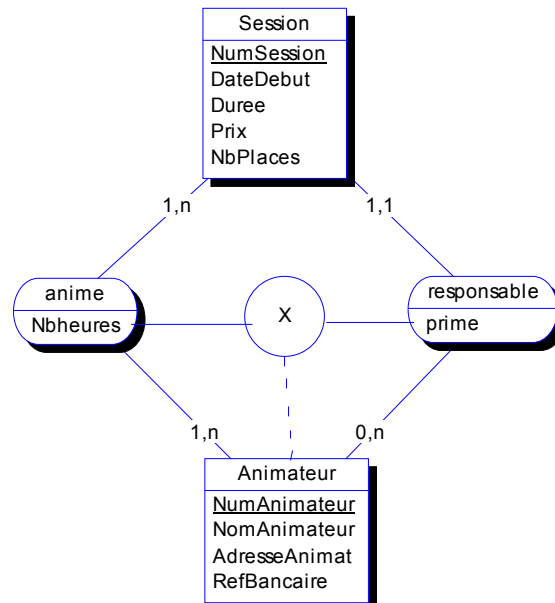
Les représentations présentées ont été réalisées avec AMC*Designor en utilisant les outils de dessin (trait, rond et texte). Ces contraintes ne sont pas prises en charge par le logiciel, de ce fait la prise en compte de ces contraintes au niveau de la base de données nécessite l'écriture d'un code PL/SQL pour Oracle sous forme de déclencheurs (*triggers*).



a. La contrainte d'exclusion :

Cette contrainte traduit le fait que toute occurrence d'une entité pivot participe à l'une ou l'autre des associations de la contrainte ou à aucune des deux mais pas aux deux. Rappelons que par défaut, le pivot désigne l'ensemble des entités communes aux associations concernées par la contrainte.

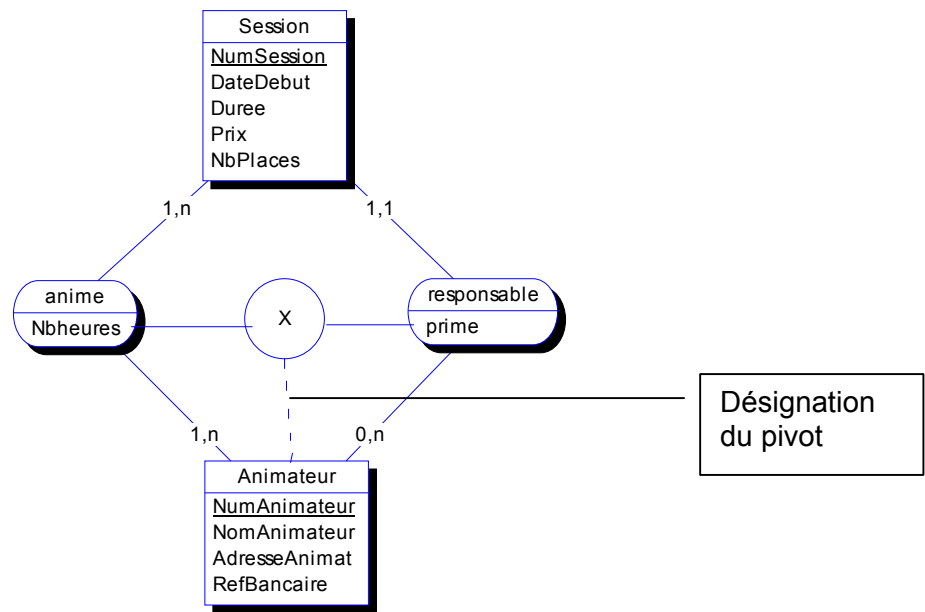
Exemple :



Dans notre exemple, l'entité pivot est ANIMATEUR. La contrainte exprimée est «Un animateur peut animer ou être responsable d'une session mais ne peut pas à la fois animer et être responsable ».

Dans le livre «Merise/2 : Modèles et techniques MERISE avancés» de Panet et Letouche, le pivot est précisé graphiquement . Le formalisme proposé consiste à relier la contrainte et l'entité ANIMATEUR par un trait en pointillé.

Exemple



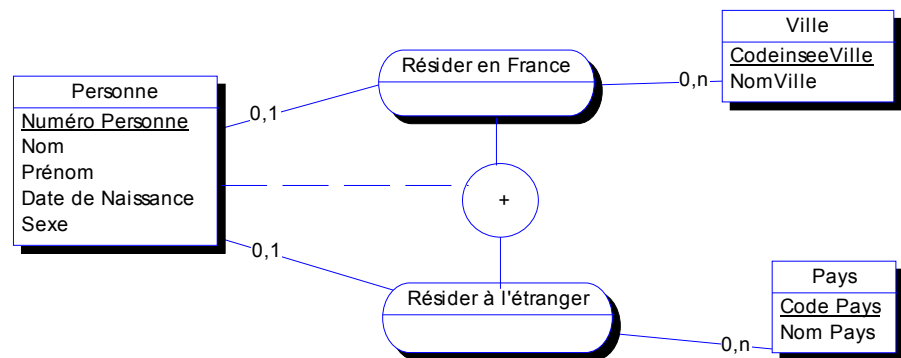


b. La contrainte de partition :

Cette contrainte traduit le fait que toute occurrence du pivot participe à l'une ou l'autre des associations de la contrainte mais pas aux deux.

Exemple

Toute personne est soit résidente en France, soit résidente à l'étranger. Elle ne peut pas être les deux.



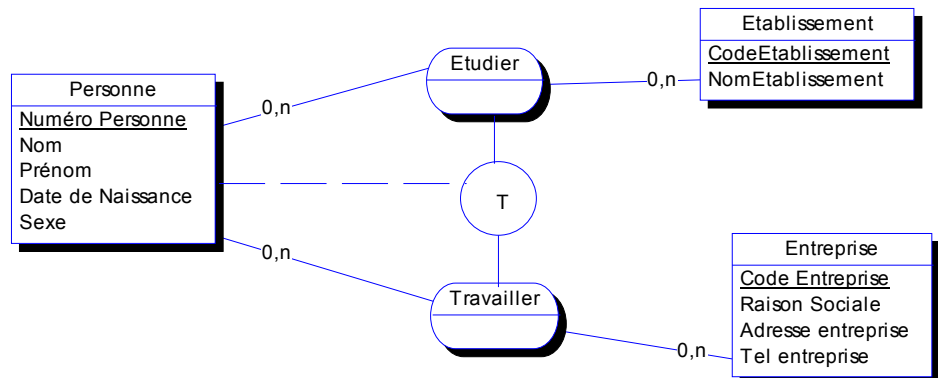


c. La contrainte de totalité :

Cette contrainte traduit le fait que toute occurrence du pivot participe à l'une ou l'autre des associations de la contrainte ou aux deux.

Exemple

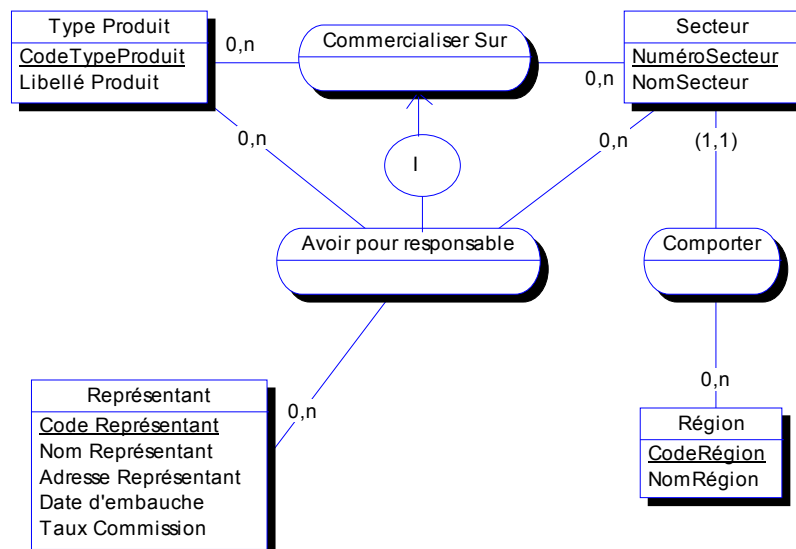
Une personne est étudiant dans un établissement de formation ou salarié dans une entreprise ou les deux à la fois (cas d'un étudiant en formation continue par exemple).



d. La contrainte d'inclusion :

Cette contrainte est orientée, elle relie deux associations dont l'une est source (ou portée) de la contrainte et l'autre cible. Elle traduit le fait que l'ensemble des couples du pivot qui participent à l'association source est inclus dans l'ensemble de ceux qui participent à l'association cible.

Exemple



L'ensemble des couples (type-produit, secteur) qui participent à l'association AVOIR POUR RESPONSABLE est inclus dans l'ensemble des couples (type-produit, secteur) qui participent à l'association COMMERCIALISER. Cela signifie que l'ensemble des produits du secteur qui ont un responsable est constitué uniquement de produits qui sont commercialisés sur le secteur.

Par son orientation dans la représentation graphique, la contrainte d'inclusion exprime également une contrainte temporelle. Une occurrence de l'association source ne peut être créée que si l'occurrence de l'association but correspondante existe.

Exemples de prise en compte de cette contrainte par l'écriture de triggers pour Oracle V7 :

```
CREATE TRIGGER commercialiser_UPDATE
AFTER UPDATE OF coderegion, numerosecteur, codetypeproduit
ON commercialiser
FOR EACH ROW
DECLARE
    check_val integer;
BEGIN
    -- Inclusion (Table <- Contrainte)
    SELECT COUNT(*) INTO check_val
    FROM avoir_pour_responsable
    WHERE avoir_pour_responsable.coderegion = :old.coderegion
    AND
    avoir_pour_responsable.numerosecteur=old.numerosecteur    AND
    avoir_pour_responsable.codetypeproduit=old.codetypeproduit;
    IF check_val != 0
    THEN
        UPDATE avoir_pour_responsable
        SET
            coderegion = :new.coderegion,
            numerosecteur = :new.numerosecteur,
            codetypeproduit = :new.codetypeproduit
        WHERE
            coderegion = :old.coderegion AND
            numerosecteur = :old.numerosecteur AND
            codetypeproduit = :old.codetypeproduit ;
    END IF;
END;
/
```



```

CREATE TRIGGER commercialiser_DELETE
AFTER DELETE
ON commercialiser
FOR EACH ROW
DECLARE
    check_val integer;
BEGIN
-- Inclusion (Table <- Contrainte)
    DELETE FROM avoir_pour_responsable
    WHERE avoir_pour_responsable.coderegion = :old.coderegion
    AND
    avoir_pour_responsable.numerosecteur=old.numerosecteur    AND
    avoir_pour_responsable.codetypeproduit=old.codetypeproduit ;
END;
/

```

```

CREATE TRIGGER avoir_pour_responsable_INSERT
AFTER INSERT
ON avoir_pour_responsable
FOR EACH ROW
DECLARE
    check_val integer;
BEGIN
-- Inclusion (Table -> Contrainte)
    SELECT COUNT(*) INTO check_val
    FROM commercialiser
    WHERE commercialiser.coderegion = :new.coderegion
    AND commercialiser.numerosecteur = :new.numerosecteur
    AND commercialiser.codetypeproduit = :new.codetypeproduit;
    IF check_val = 0
    THEN
        Erreur
        ...
        RETURN;
    END IF;
END;
/

```

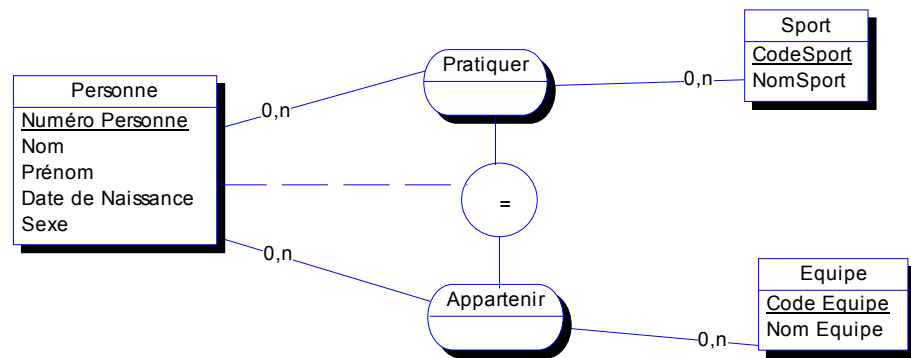
```
CREATE TRIGGER avoir_pour_responsable_UPDATE
AFTER UPDATE OF coderegion, numerosecteur, codetypeproduit
ON avoir_pour_responsable
FOR EACH ROW
DECLARE
    check_val integer;
BEGIN
-- Inclusion (Table -> Contrainte)
    SELECT COUNT(*) INTO check_val
    FROM commercialiser
    WHERE commercialiser.coderegion = :new.coderegion
    AND commercialiser.numerosecteur = :new.numerosecteur
    AND commercialiser.codetypeproduit = :new.codetypeproduit;
    IF check_val = 0
    THEN
        Erreur
        ...
        ...
        RETURN;
    END IF;
END;
/
```



e. La contrainte d'égalité :

Cette contrainte est également nommée contrainte de simultanéité.

Cette contrainte traduit deux inclusions symétriques. L'ensemble des occurrences du pivot est donc le même pour chacune des associations participant à la contrainte.



Toute personne qui pratique un sport fait partie d'une équipe et vice-versa. Les personnes qui participent aux deux associations sont donc les mêmes.



f. La contrainte d'unicité :

Elle permet de préciser l'identifiant de l'association lorsque la dimension de celle-ci est supérieure strictement à 2. Cette contrainte permet d'éviter les erreurs liées à la 2^{ème} forme normale lors du passage au modèle logique de données relationnel normalisé.

Exemple

La contrainte d'unicité permet de représenter le fait qu'une personne à une date donnée ne peut occuper qu'une et seule fonction. Sans la contrainte une personne pourrait occuper plusieurs fonctions différentes à une date donnée.

