esprit
Se former autrement

Chapitre VII:
Préparation à l'implémentation
Partie 1
De l'UML à SQL

Module Langage de modélisation UML

Année Universitaire
2016-2017

PLAN

Introduction

A Traduction des classes

- Les attributs
- Les méthodes

B Traduction des associations

- Association simple et réflexive
- Les associations attribuées
- Composition et agrégation

C Traduction de la généralisation

Exercices et conclusion

Introduction

Pour conceptualiser, on utilise **le modèle objet** , pour stoker , on utilise **le modèle relationnel**.

Le modèle relationnel , la cible, utilise structurellement trois concepts :

1. La relation qui regroupe un ensemble d'attributs.
2. L'attribut qui est définit sur un domaine.
3. Le domaine qui est définit sur un ensemble de valeurs.

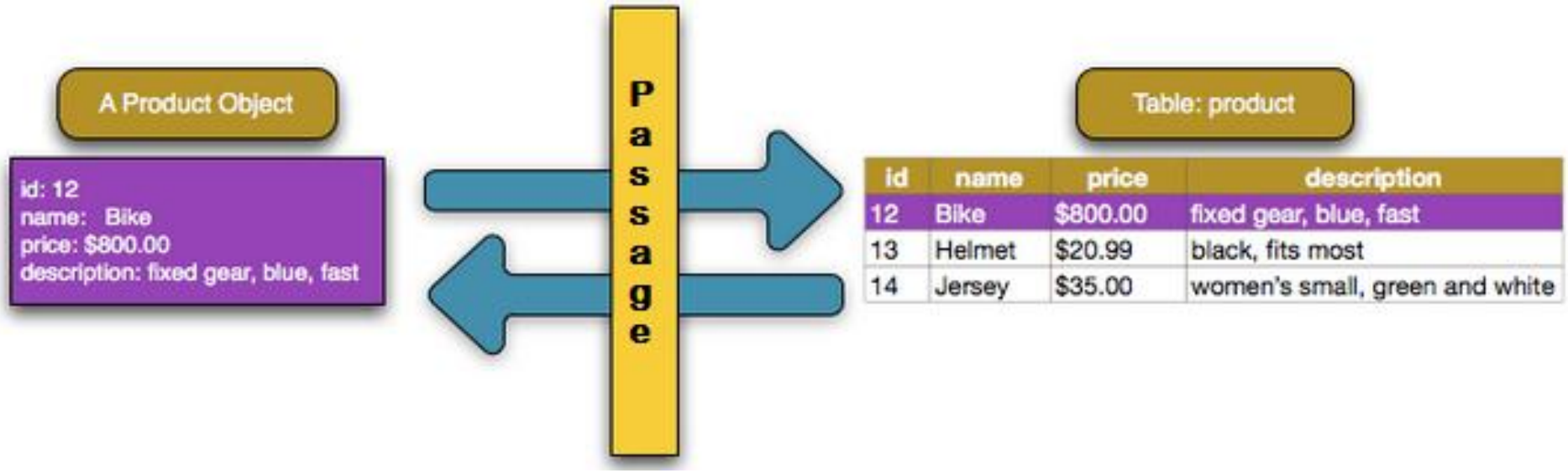
✓ *La notion de **domaine** est identique à celle de **type** pour les diagrammes de classes. on utilise donc **des domaines identiques***

.

Introduction

- Il faut choisir les classes à persister généralement à partir du diagramme de classe de conception, qui sont les entités dont on a besoin de stocker leurs données.

- Une classe persistante ne peut pas avoir uniquement des attributs non-persistants..

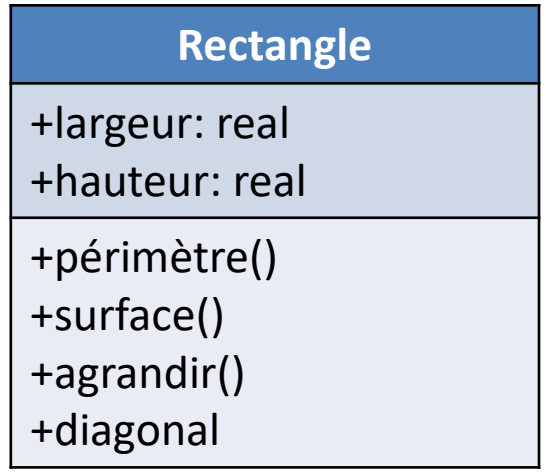


id	name	price	description
12	Bike	$800.00	fixed gear, blue, fast
13	Helmet	$20.99	black, fits most
14	Jersey	$35.00	women's small, green and white

Traduction des classes

Chaque classe devient une relation:

La règle est assez simple *« Chaque classe devient une relation ; chaque attribut de la classe devient un attribut de la relation »*
Dans l'exemple suivant *Rectangles* , nous aurons une relation *Rectangles avec deux attributs largeur et hauteur .*



Rectangle (largeur, hauteur)

Clef primaire manquante
Voir diapo 5

Traduction des classes

Comment identifier les objets dans une relation(table)?

Dans le monde des objets, il existe par définition un **Oid** (Object identifier) pour tout objet.

Ce n'est pas le cas pour les relations. Soit il existe **une clé naturelle** , par exemple un numéro d'emloyé ,(ou un groupe d'attribut), alors on choisira ces attributs comme clé de la relation.

Sinon, on ajoute **une clé artificiel** à la relation.

Exemple : **Rectangle (Id_rectangle, largeurnumber, hauteurnumber)**

Traduction des classes

Que faire avec les méthodes?

- Il n’existe pas de solution unique pour les méthodes.
- Dans le cas des SGBD relationnels, les traitement doivent être décrit dans un langage procédural tel que PL/SQL .
- Les traitements peuvent être extérieurs à la base de données ou stockés dans le serveur de la base de données.

Traduction des classes

Il y a trois possibilités pour traiter des méthodes sans faire appel à des traitements procéduraux:

1. Mémoriser les attributs calculés

Pour les méthodes qui ne modifient pas l'état de l'objet et dont le but est de retourner une valeur sur l'état de l'objet, il est possible de rendre statique ces méthodes en leur substituant un attribut qui correspond à la valeur de l'objet. Pour notre exemple,

nous obtenons la déclaration suivante:

```
Create Table Rectangle1 (
    Id_rectangle integer primary key,
    largeurnumber,
    hauteurnumber,
    surfacenumber,
    perimetre number,
    diagonalnumber)
```

- L'inconvénient de ce choix réside dans **l'aspect statique,** Lors de modification des attributs dont dépend ces méthodes, il faut mettre à jour les attributs correspondants aux méthodes. Dans notre cas, si l'on modifie la largeur d'un rectangle, il faut mettre à jour les attributs surface, périmètre et diagonal .

Traduction des classes

2. Utiliser les vues:

Les vues peuvent être une solution dynamique aux méthodes qui ne modifient pas l'état de l'objet et dont le but est de retourner une valeur sur l'état de l'objet.

✓ On peut dire que la vue va simuler une table semblable à celle de *rectangle*, Mais les attributs *surface, périmètre, diagonal* seront calculés à chaque fois que la table sera requise dans une sélection.

```
Create vue Rectangle2 as
        Select
          Id_rectangle,
          largeur,
          hauteur,
         largeur*hauteur surface,
         2*(largeur+hauteur) perimetre,
         sqrt(largeur*largeur+hauteur*hauteur) diagonal)
```

Traduction des classes

3. Utiliser les méthodes de mise à jour:

Cette technique peut être parfois utiliser pour écrire des scripts de mise à jour correspondant à une méthode modifiant l'état de l'objet.

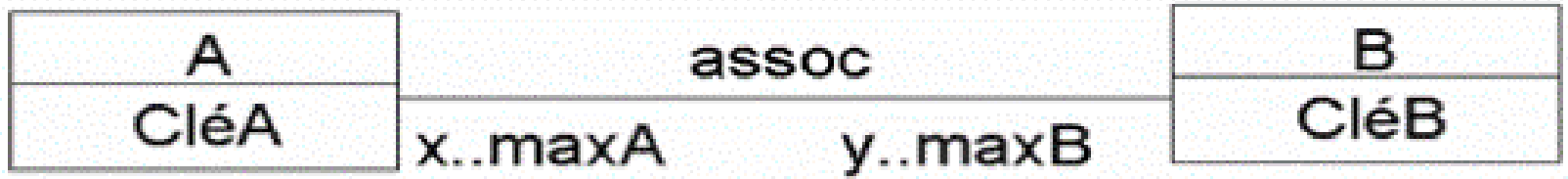
On peut transcrire la méthode doubler de rectangle dans la requête SQL suivante :

```
Update Rectangle
        Set largeur=2*largeur, hauteur=2*hauteur
      Where num_rectangle= …
```

✓ Il existe pratiquement toujours une solution à la transcription des méthodes que l'utilisation d'un langage procédural qui permet le traitement de tous les cas, Mais ceci ne remplace pas complétement les méthodes surtout dans l'aspect d'héritage !

Traduction des associations

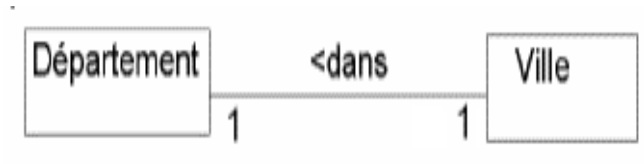
La solution dépend des *m*ultiplicités :



	maxA = 1	maxA > 1
maxB = 1	une des autres solutions ou fusion en une seule relation **1**	CléB dans A comme clé étrangère **2**
maxB > 1	CléA dans B comme clé étrangère **3**	créer relation assoc avec comme clé CléA et CléB **4**

Traduction des associations: exemples

1. Simple association : multiplicité [0..1,0..1] [0..1,1] [1,1] [1,1..0]

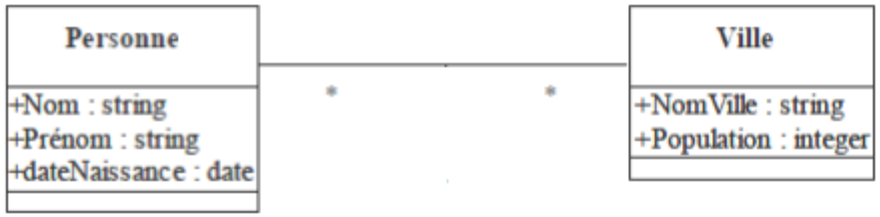


departement(n° dep,….)
ville(n° Ville, # n° dep , nom_ville,…)

2,3. Simple association : multiplicité [1,*],[*,1]

departement(n° dep,….)
ville(n° ville, , # n° dep ,nom_ville,…)

4. Simple association : multiplicité [*,*]



Ville(nom-ville , population)

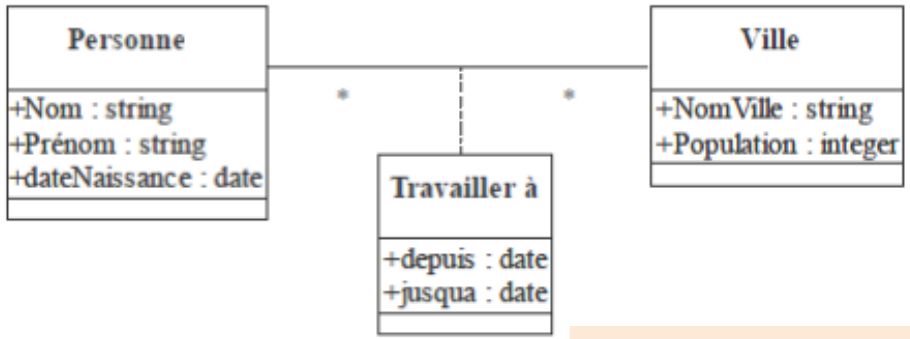
Personne(id-personne , nom, prenom, date_naiss)

Travailler_a(#Id_personne,# NomVille)

Traduction des associations: exemples

D. Simple attribué : [*,*]

- Association attribué est traitée comme l'association à laquelle elle est attachée.
- Dans le cas où il suffit d'ajouter un attribut clé dans une des deux relations, il suffit aussi d'ajouter les attributs de l'association attribuée dans cette même relation.
- Dans le cas où il a fallu créer une nouvelle relation, il faut ajouter les attributs de l'association attribuée dans cette relation. Examinons l'exemple suivant:



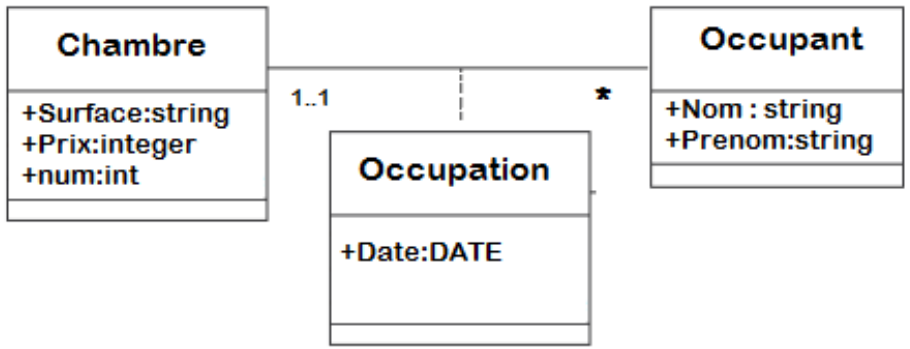
Ville(<u>nom-ville</u> , population)

Personne(<u>id-personne</u> , nom, prenom, date_naiss)

Travailler_a((<u>#Id_personne, #NomVille</u>), depuis, jusqua)

Traduction des associations: exemples

D. Simple attribué : [*,1..0]



chambre(id_chambre,surface,prix,num)

occupant(id-occupant ,#id_chambre,nom, prenom)

occupation(#id-occupant,date)

Traduction des associations: exemples

A,B,C. Agrégation et composition :

L'agrégation et la composition se traitent comme les associations. (La dépendance entre les classes est souvent une indication pour l'utilisation du ***delete cascade*** *)*

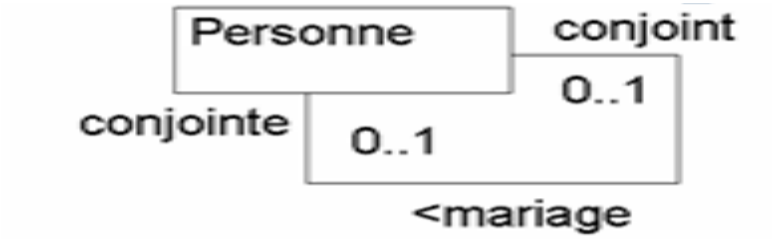
Dans l'exemple suivant, les polygomes sont composés de points et en plus il existe une contrainte d'ordre sur les points le premier, second, etc.. Point du polygones. Cette contrainte d'ordre est traduite par un attribut num_ordre qui permet d'ordonner les points. On remarquera aussi que la relation Points ne possède pas de clé propre, la clé est formée par les attributs *IdPolygone et num_ordre*

Polygones(IdPolygone ,Couleur)

Points((IdPoint, # IdPolygone) , x , y,) : entité faible par rapport à l'entité polygone

Traduction des associations: exemples

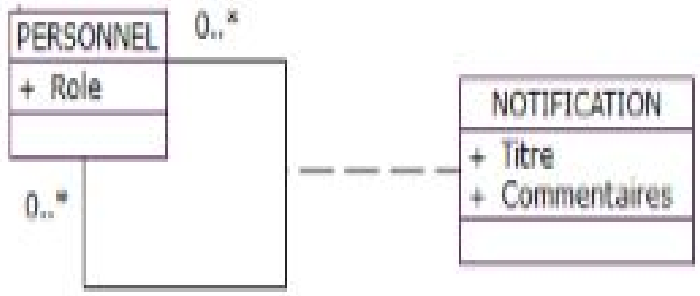
2. Association réflexive :



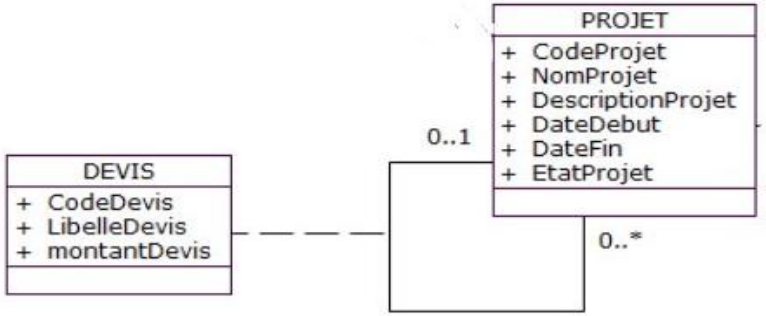
Personne(N° Pers,# N° PersConjoint,……) avec valeurs nulles

Ou

personne(N° Pers….) et Mariage(#N° PersConjoint,#N° PersConjointe)



personnel(id-personnel,role,……)
notification((#id-personnel,#id-pers-parent) titre,commentaire)



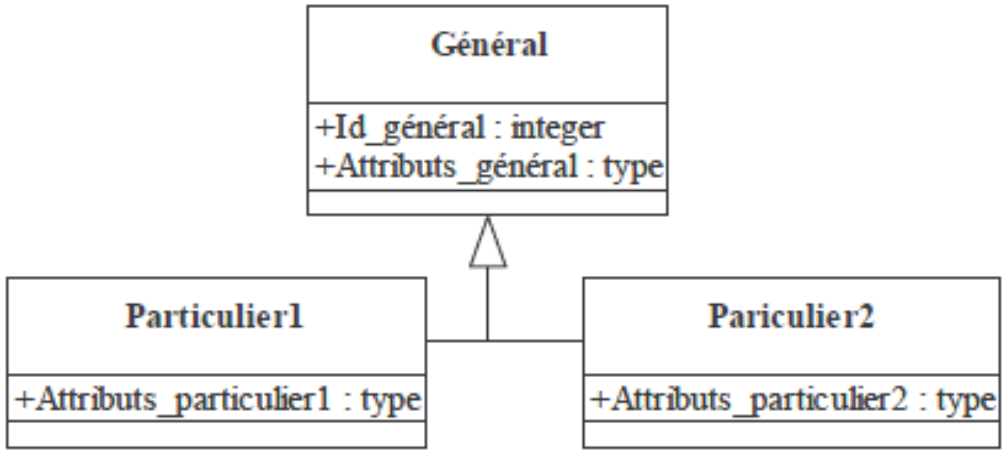
projet(id-projet,#id-projet-parent,code_projet, numProjet,……)
devis(#id-projet,code_devis,libelle_devis,montant_devis)

Traduction de la généralisation

La généralisation ou l'héritage est un concept qui n'a pas d'équivalent dans le monde relationnel. Il faut donc faire un choix entre regrouper toutes les particularité dans une seule relation ou conserver chaque particularité dans une relation propre. On examinera ces deux solutions .

1. Tous dans un

Cette solution consiste à mettre tous les attributs dans la même table, à ajouter un constituant donnant le genre de l'objet.



Général(Id_général,attributs_de_général …, …attributs _de_particulier1…, …attributs_de_particulier2…, …)

- ✓ ***Les attributs non utilisés sont laissés à null.***
- ✓ ***On peut ajouter un attribut genre dont les valeurs possibles sont G,P1,P2***

Traduction de la généralisation

2. Chacun à sa place

- Dans cette solution, chaque classe devient une relation.
- On ajoute éventuellement un attribut de genre dans la relation Général dont les valeurs possibles sont G,P1,P2.

Général(id_général, …attribut_de_général …)

Particulier1(# id_général, attribut_de_particulier2 …)

Particulier2(#id_général,attribut_de_particulier2 …)

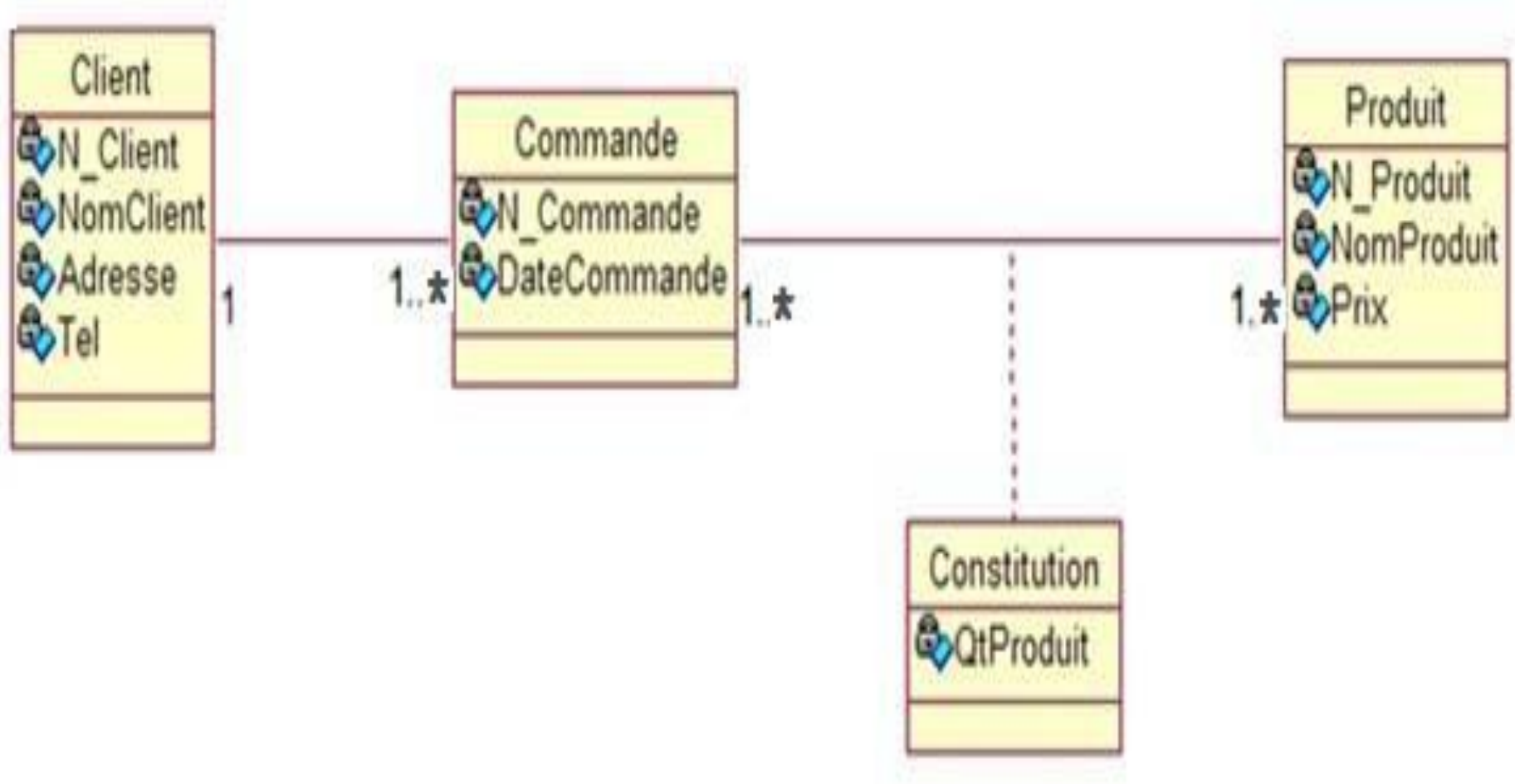
3. Deux tables filles

Particulier1(id_particulier1, attribut_de_général ,**attribut_de_particulier2 …)**

Particulier2(id_particulier2, attribut_de_général ,attribut_de_particulier2 …)

Exercice 1

Construire le modèle relationnel correspondant au diagramme de classe suivant :



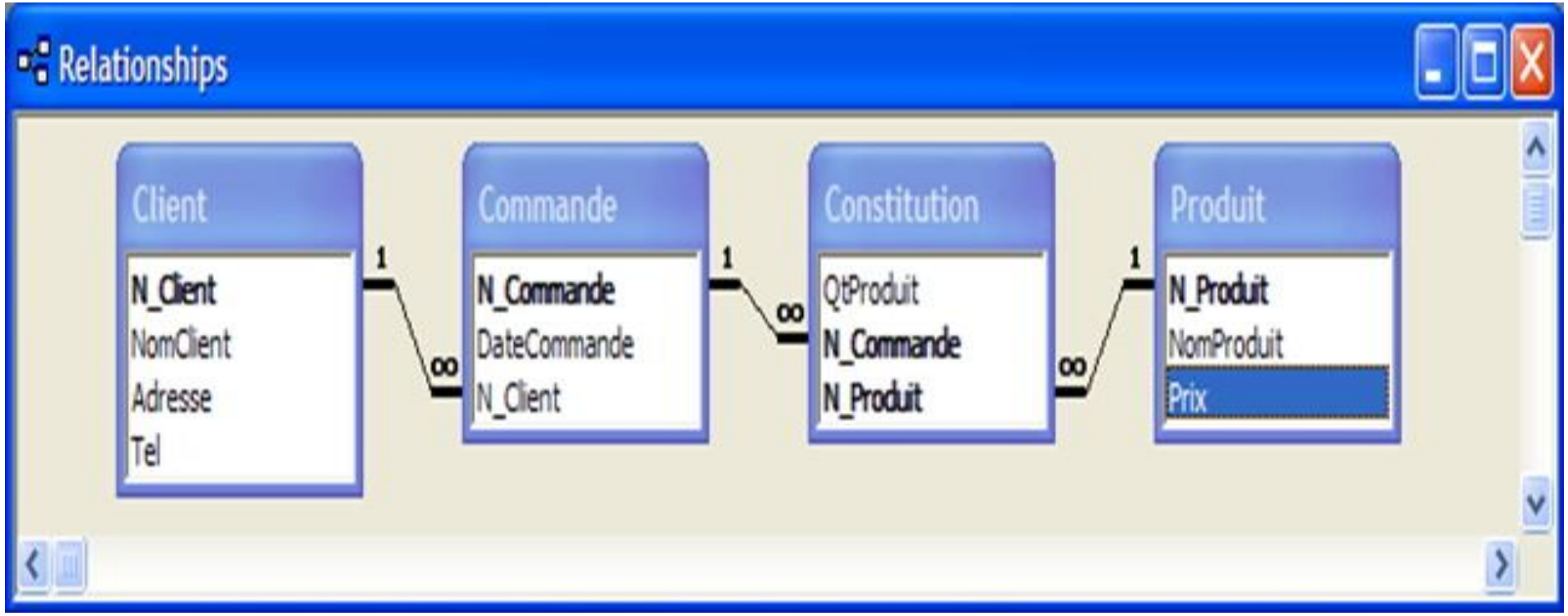
Solution 1

client(n_client,nom_client,adresse,tel)

commande(n_commande,#n_client,date_commande)

constitution((#n_commande,#n_produit),qt_produit)

Produit(n_produit,nom_produit,prix)



Client

N_Client	Nom Client	Adresse	Tel
1	Michel	Bruxelles	123456
2	David	Namur	456298
3	Manuel	Dinant	876230
4	Lucas	Bruge	937402
5	Tintin	Bruxelles	384043

Commande

N_Commande	DateCommande	*N_Client*
1	12/09/98	1
2	15/03/97	1
3	12/09/98	3
4	10/01/00	3
5	20/10/00	4
6	15/02/00	5

Constitution

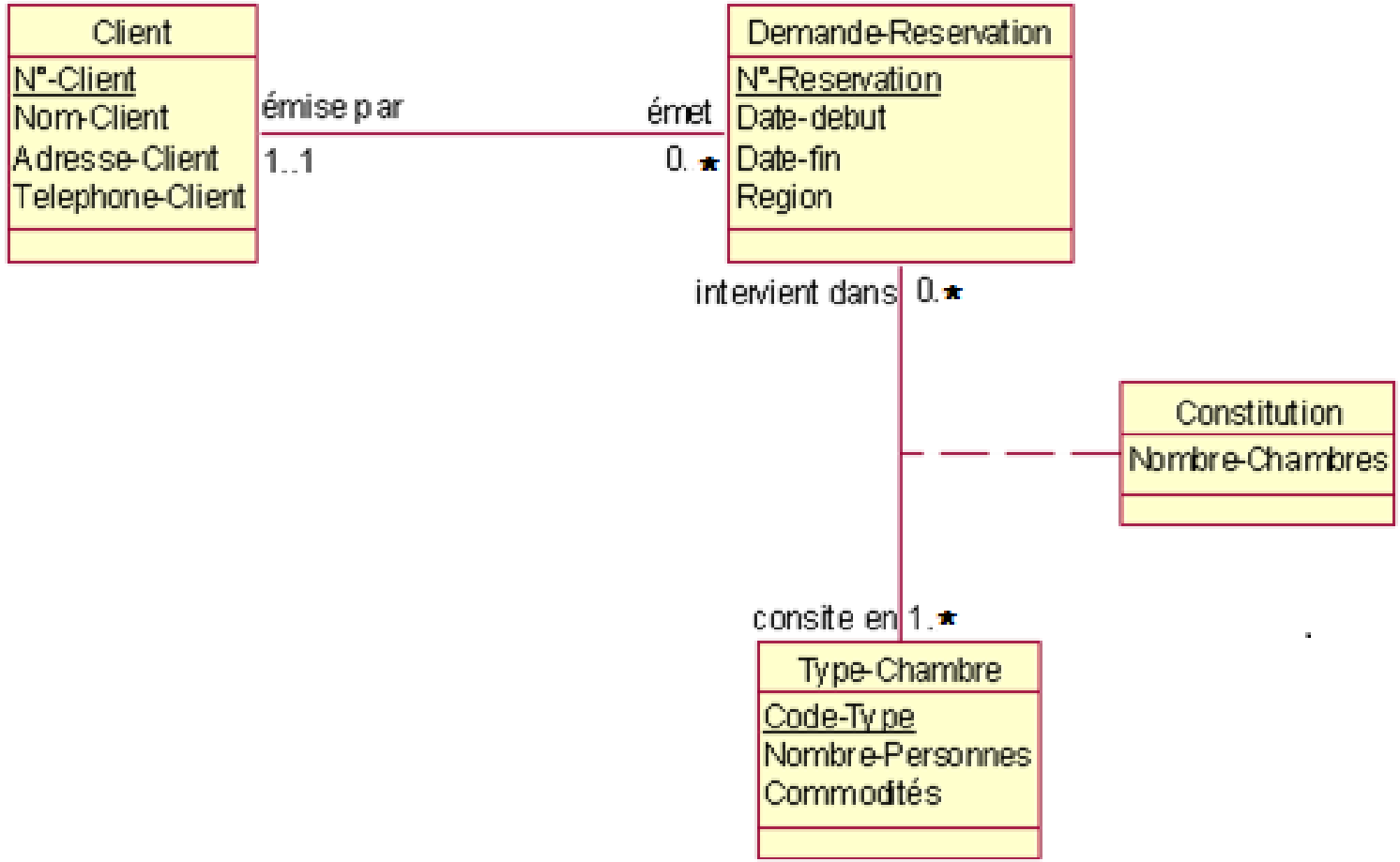
QtProduit	*N_Commande*	*N_Produit*
1	5	200
1	5	400
2	3	500
3	6	500
2	5	600
2	4	600

Produit

N_Produit	NomProduit	Prix
100	Walkman	5.000,00 FB
200	TV	20.000,00 FB
300	GSMNokia	10.000,00 FB
400	PlayStation2	200.000,00 FB
500	Leonidas	500,00 FB
600	Godiva	1.200,00 FB

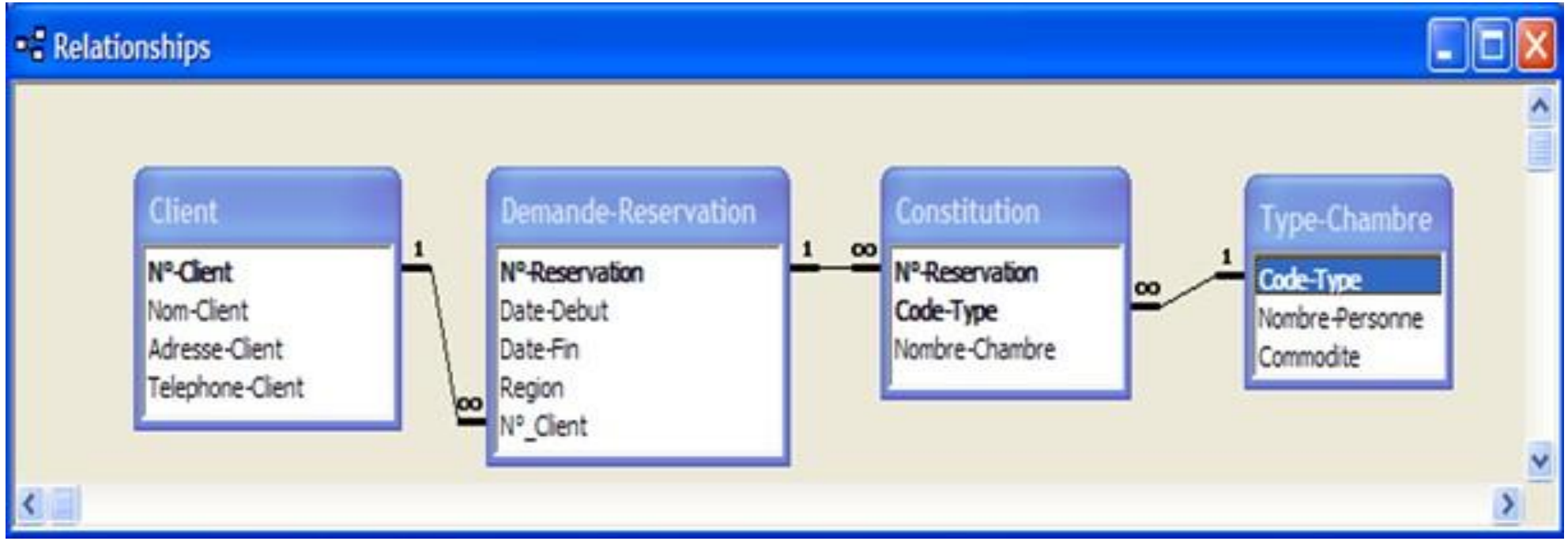
Exercice 2

Construire le modèle relationnel correspondant au diagramme de classe suivant :



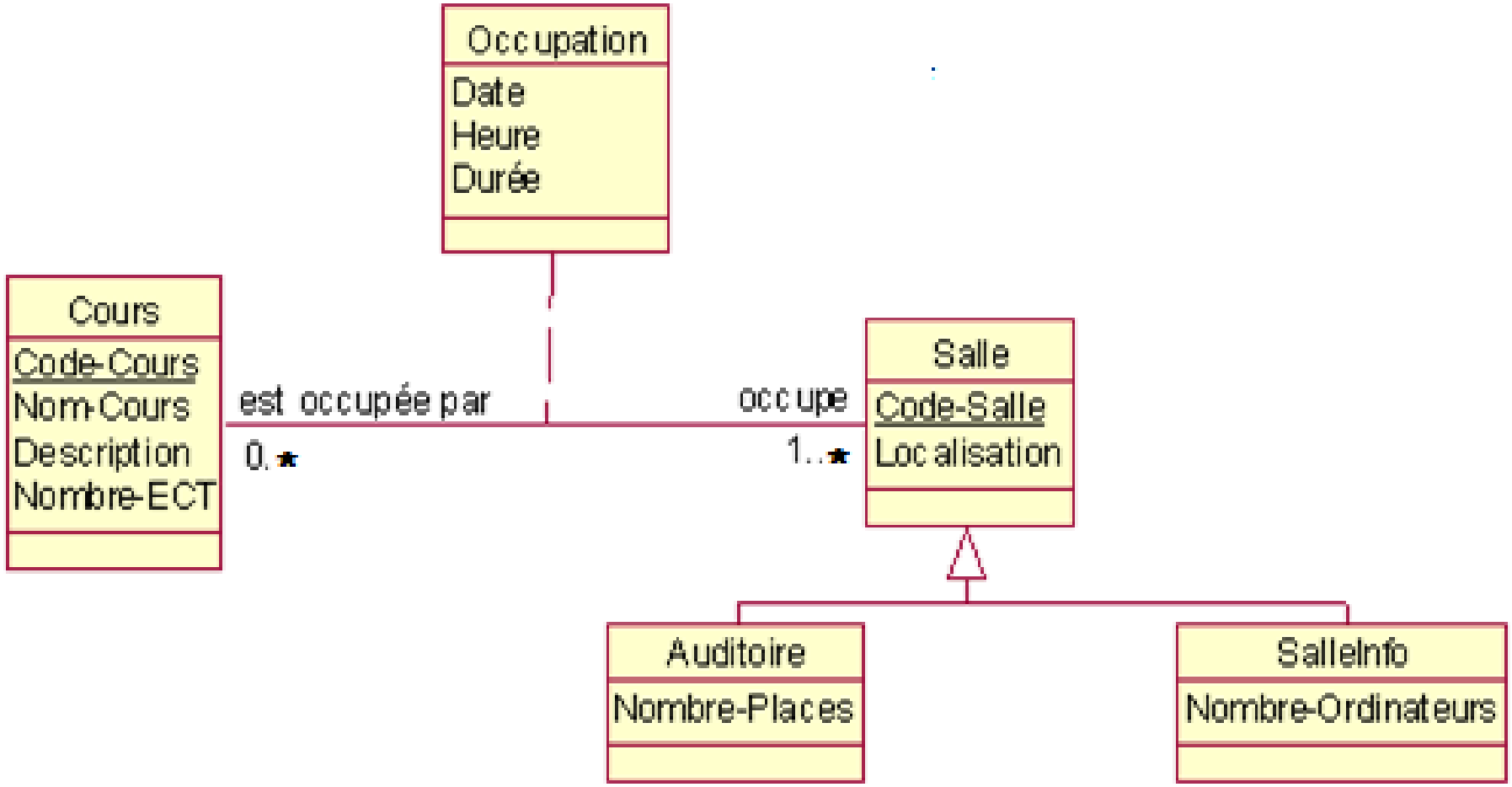
Solution 2

client(n_client,nom_client,adresse_client,telephone_client)
demande_reservation(n_reservation,#n_client,date_debut,date_fin,region)
constitution((#n_reservation,#code_type),nombre_chambre)
type_chambre(code_type,nombre_personne,commodite)



Exercice 3

Construire le modèle relationnel correspondant au diagramme de classe suivant :



Solution 3

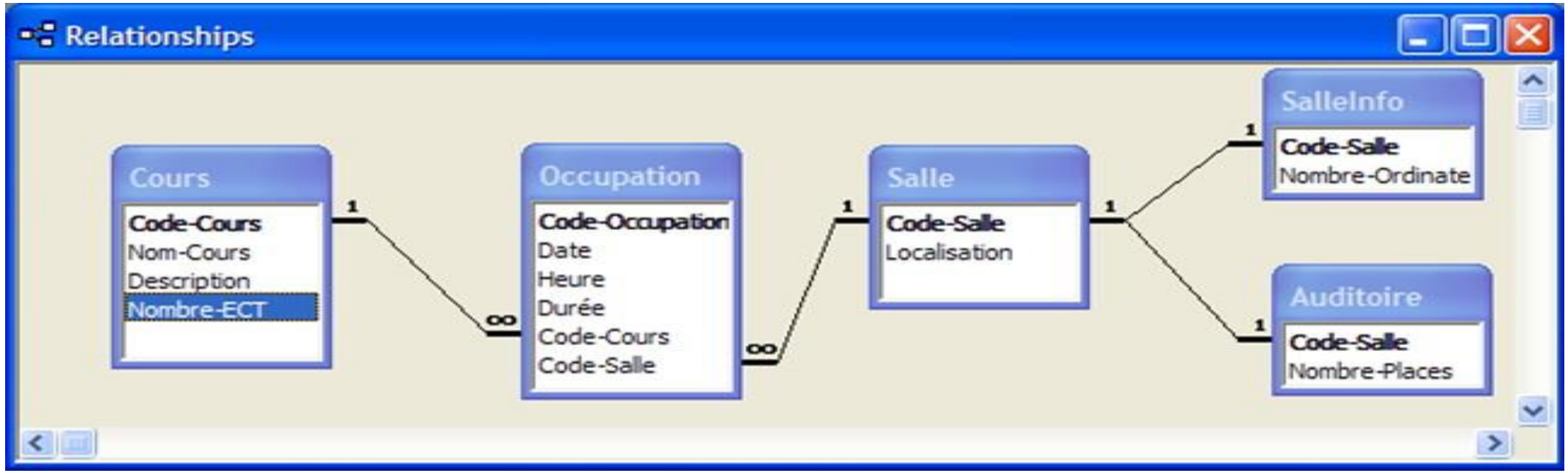
cours(code_cours,nom_cours,description,nombre_ect)

occupation((#code_cours, #code_salle),date,heure,durée)

salle(code_salle,localisation)

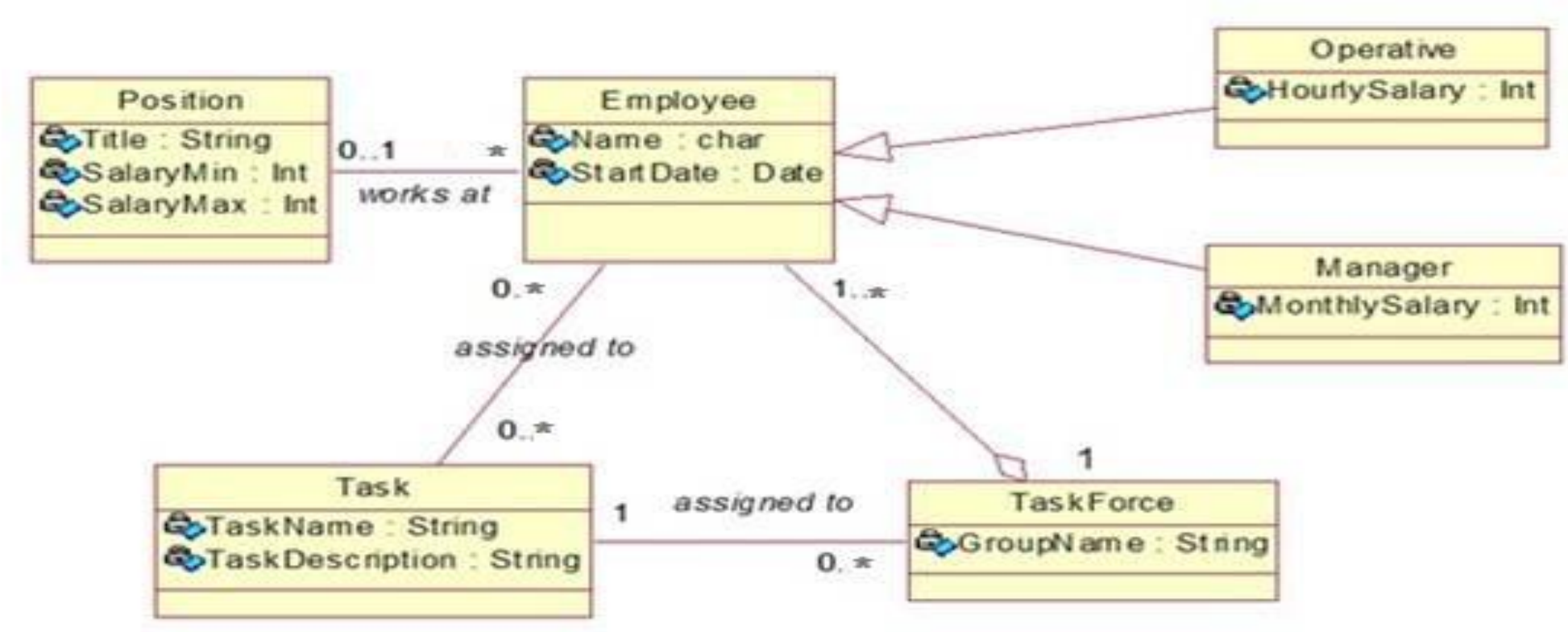
salle_info(#code_salle,nombre_ordinateurs)

auditoire(#code_salle,nombre_places)



Exercice 4

Construire le modèle relationnel correspondant au diagramme de classe suivant :



Solution 4

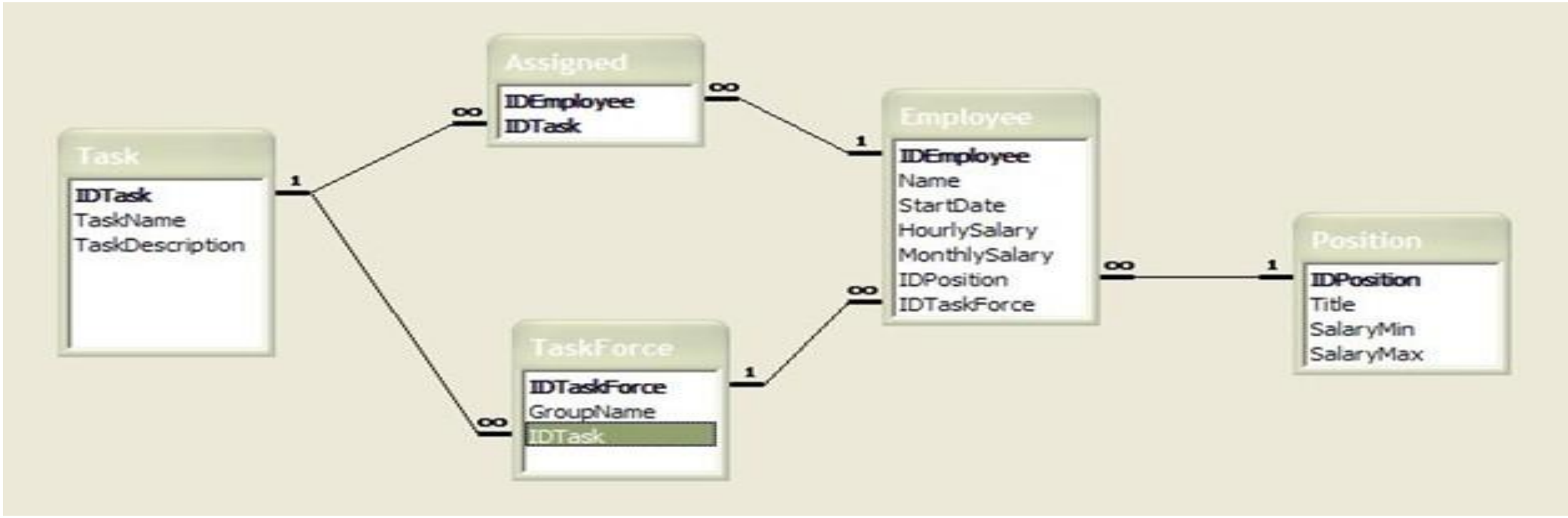
task(id_task,task_name,task_description)

assigned(#id_task,#id_employee)

task_force(id_task_force,#id_task,group_name)

employee(id_employee, ,#id_position,#id_task_force,name,start_date,hourly_salary,monthly_salary)

position (id_position,title,salary_min,salary_max)



Bibliographie

http://www.irit.fr/~Thierry.Millan/CNAM-NFP107/UML%20et%20les%20Bases%20de%20Donn%C3%A9es.pdf

http://www.essai.rnu.tn/Ebook/Informatique/uml2enaction.pdf

http://www.essai.rnu.tn/UML2_par_la_pratique.pdf

http://www.essai.rnu.tn/Ebook/Informatique/conceptiondebasesdedonneesavecuml.pdf