

Les Extensions MERISE

Partie I: Extensions MCD

Compléter les modèles conceptuels de données (MCD) suivants par les extensions appropriées pour prendre en compte les contraintes exprimées en langage naturel.

1) Gestion de stocks et approvisionnement

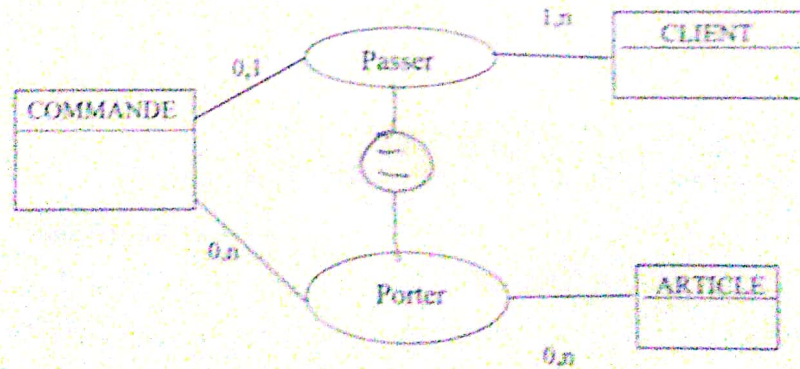
Un article peut être acheté chez des fournisseurs, approvisionné par des unités de production extérieures, ou élaboré (assemblé) directement dans le domaine; il ne peut cependant être à la fois acheté et approvisionné.



Un article qui participe à acheter ne peut pas participer à approvisionnement

2) Prise de commande

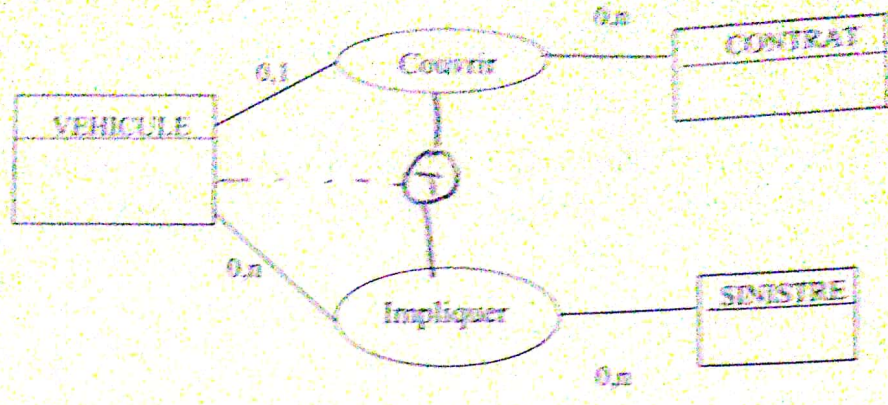
Une commande portant sur des articles est obligatoirement passée par un client et réciproquement.



→ TT (Acheter) ∩ TT (Approv) = ∅
TT Article

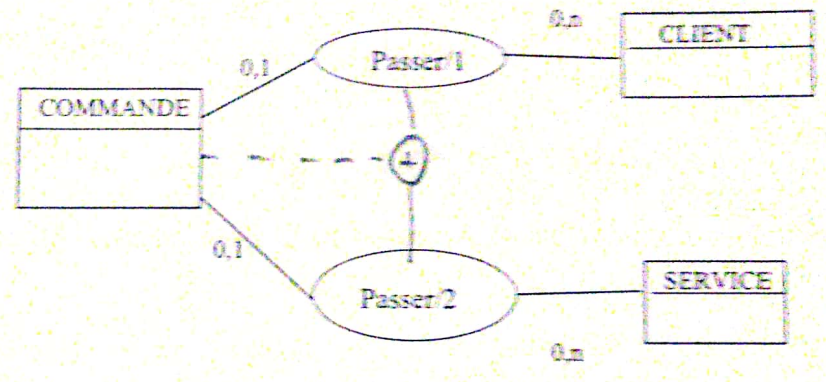
3) Contrat d'assurance

Tout véhicule est au minimum associé soit à un contrat par l'association Couvrir, soit à un sinistre par l'association Impliquer, soit aux deux.



4) Gestion de commandes

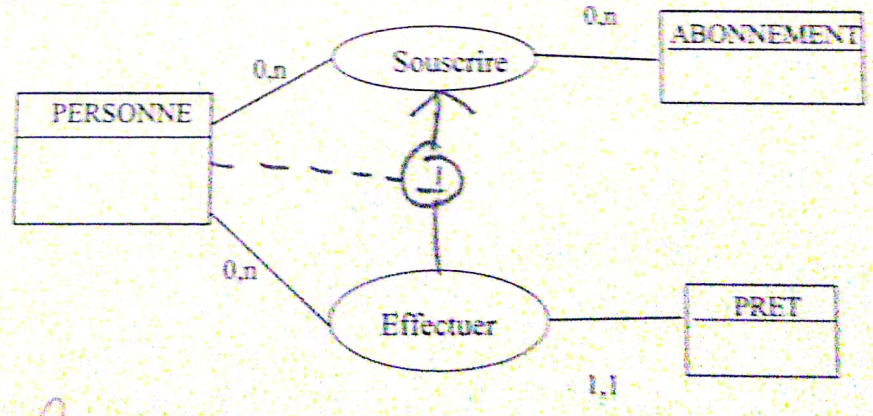
Soit une entité type **COMMANDE** participant à deux associations types **Passer 1** et **Passer 2**. toute occurrence de l'entité type **COMMANDE** participa soit à l'association **Passer 1**, soit à l'association **Passer 2**.



Handwritten note: + = T x +

5) Gestion de prêts dans une bibliothèque

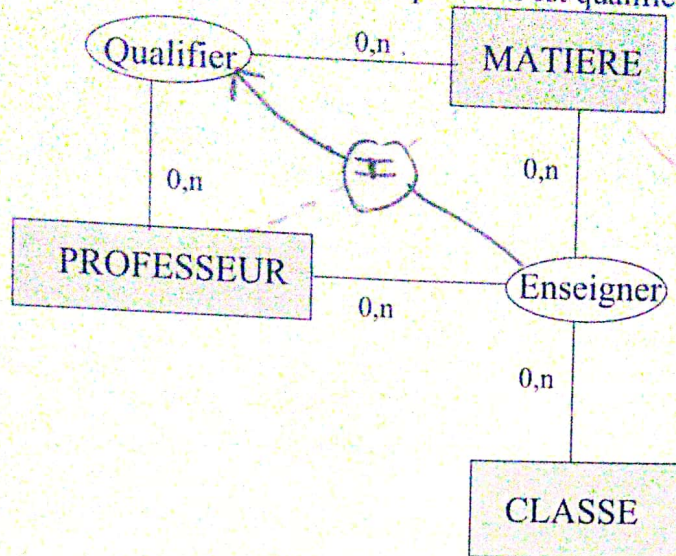
Toute personne qui effectue un prêt doit avoir souscrit un abonnement.



Handwritten note: On met le pivot car c'est pas les mêmes objets au bout des 2 assoc.

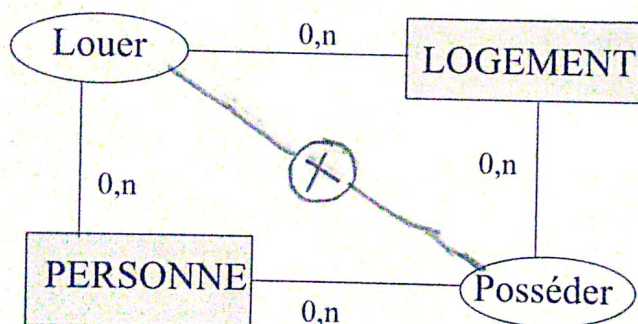
6) Gestion de enseignements

Un professeur n'enseigne que les matières pour lesquelles il est qualifié.



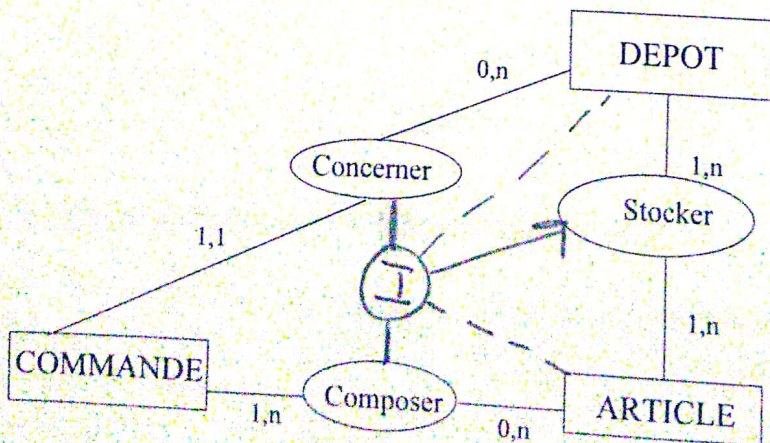
7) Agence immobilière

Une personne ne peut pas être à la fois le propriétaire et le locataire d'un logement.



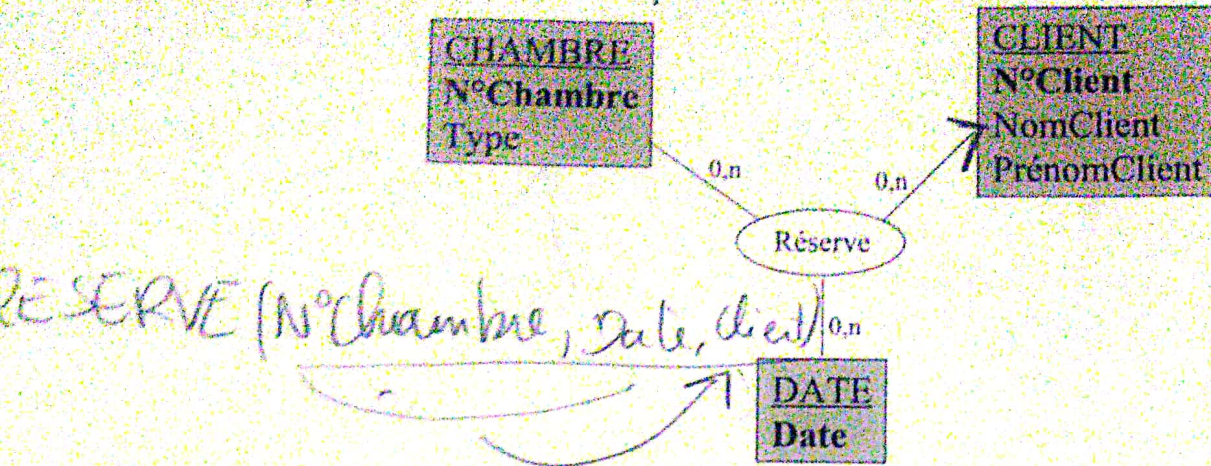
8) Préparation de commande

Quand un article entre dans la composition d'une commande, et que cette commande concerne un dépôt, alors cet article est obligatoirement stocké dans ce dépôt.

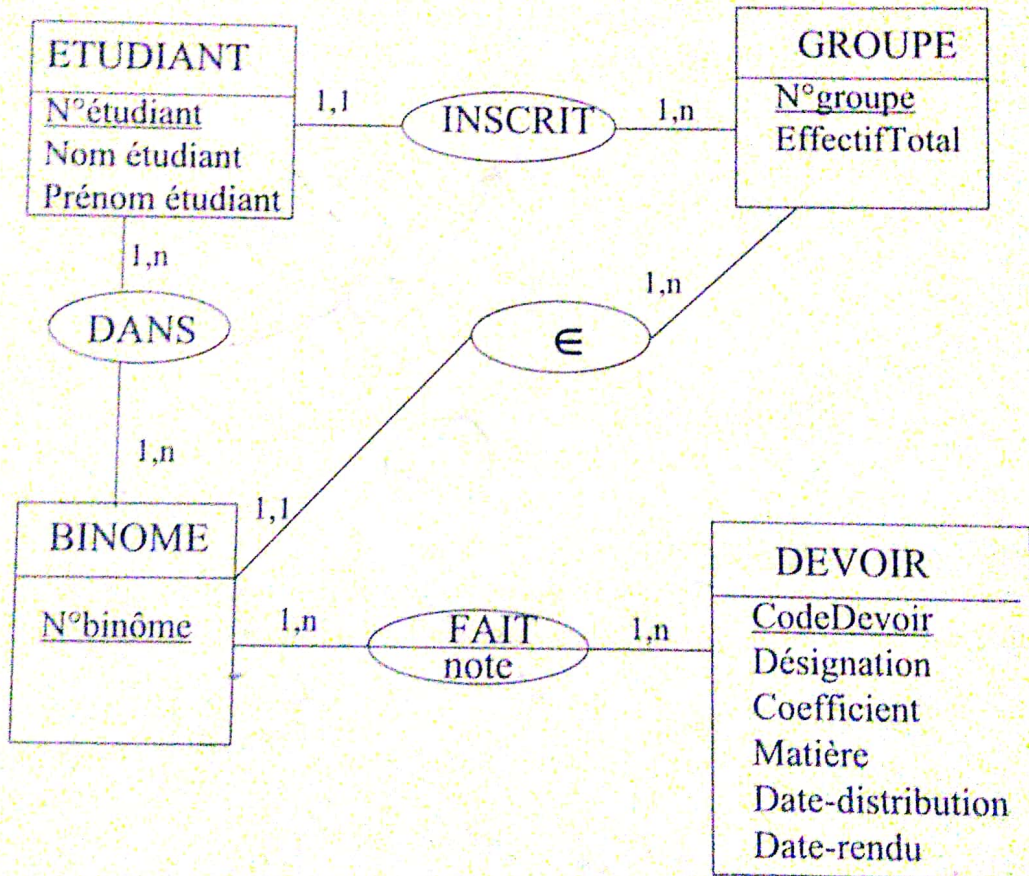


9) Réservation de chambre d'hôtel

On sait qu'une chambre d'hôtel ne peut être louée à deux clients à une même date.



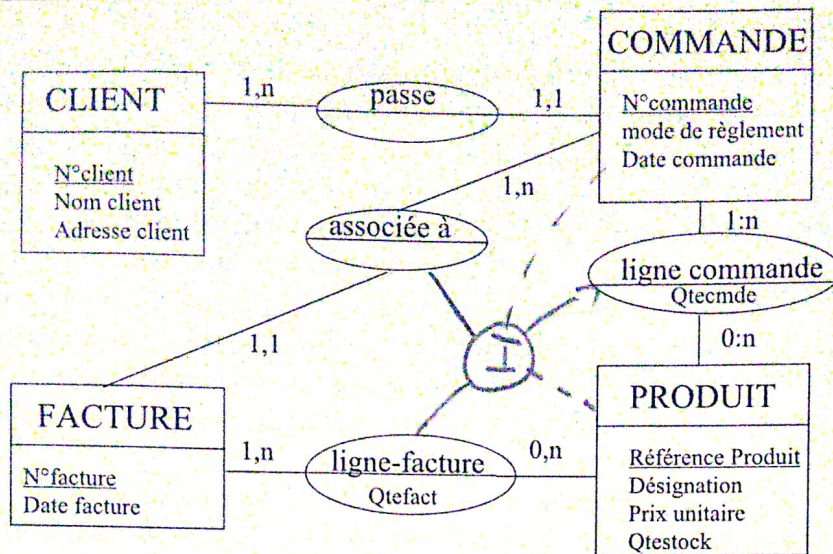
10) Gestion des binômes



10.1 Comment indiquer que l'étudiant doit avoir une seule note par devoir.

10.2 Comment indiquer que les binômes ne peuvent pas être formés par des étudiants appartenant à des groupes différents.

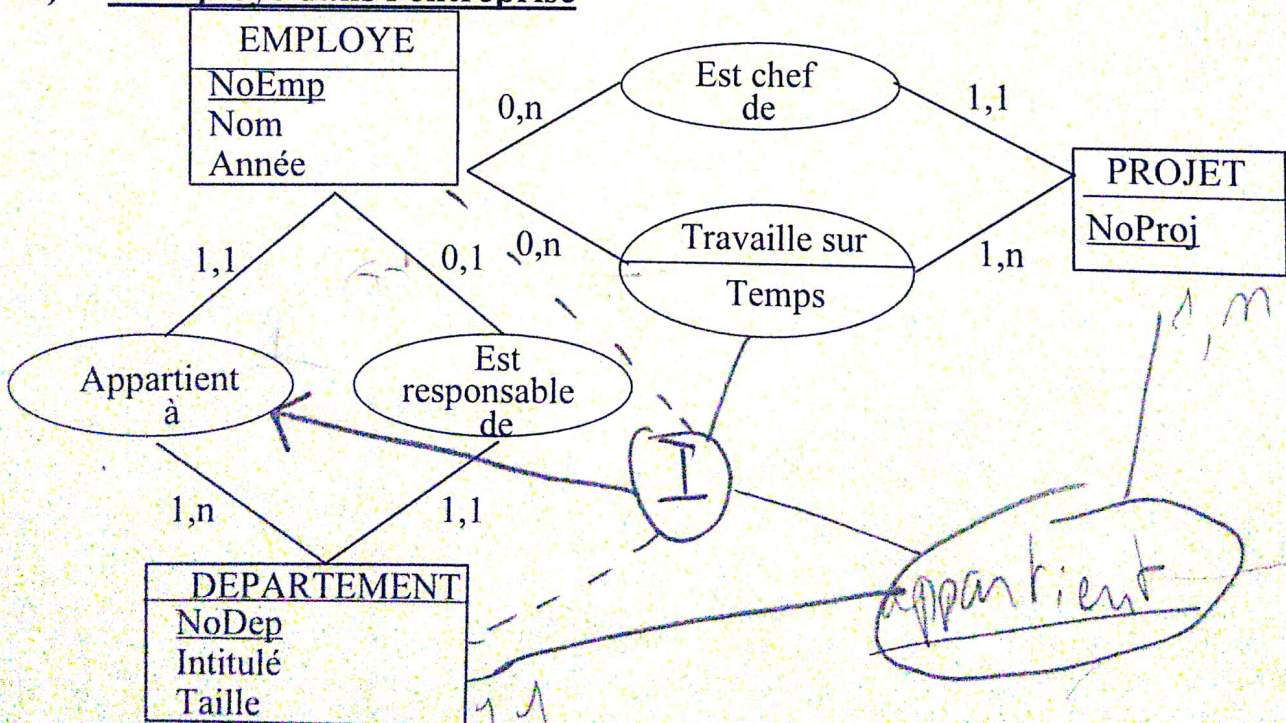
11) PME Luminare



11.1 On ne peut pas facturer des produits qui n'ont pas été commandés (Les produits facturés doivent être inclus dans les produits commandés).

11.2 La quantité facturée pour chaque produit ne peut être supérieure à la quantité commandée.

12) L'employé dans l'entreprise



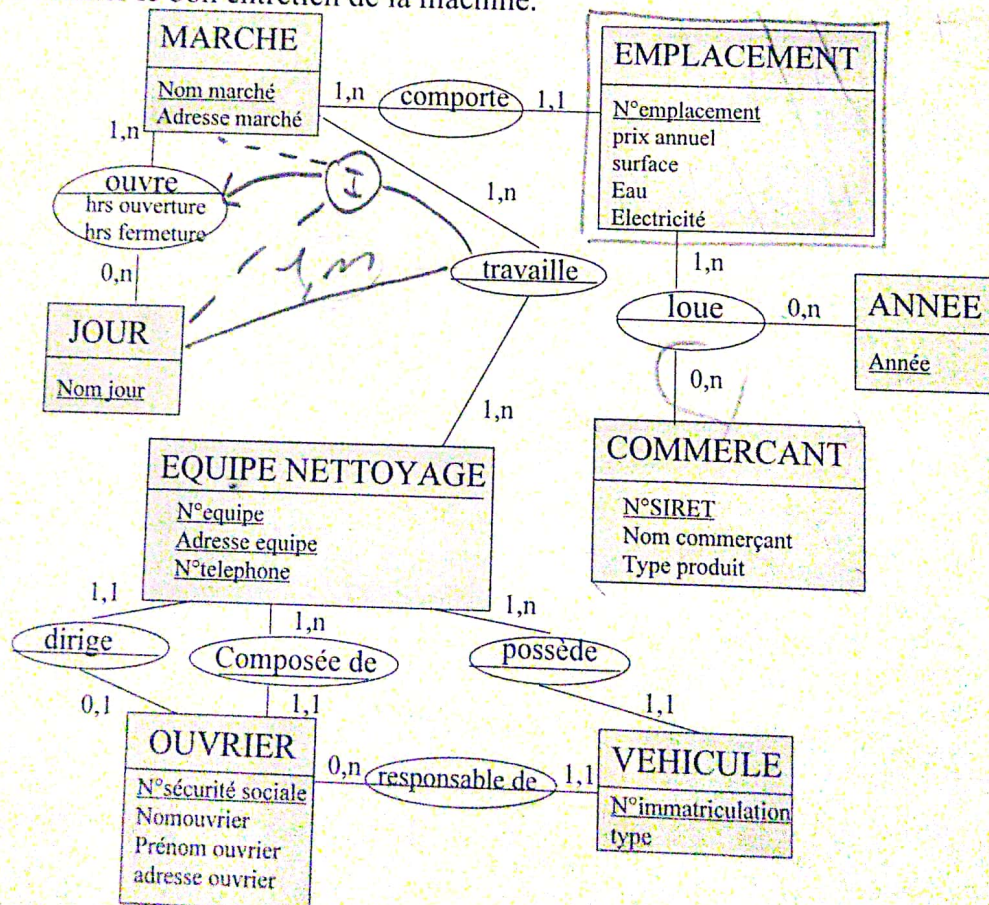
12.1 Représentez graphiquement le fait que des employés travaillent sur des projets dépendant du même département.

13) Les marchés de Paris

La ville de Paris souhaite informatiser la gestion des emplacements ainsi que le nettoyage des marchés (puces, fruits & légumes, etc.).

Un marché est décrit par un nom unique et une adresse et comporte de nombreux emplacements. Il est ouvert certains jours de la semaine, mais selon le jour, un marché commence et termine à des heures différentes. Le système doit permettre de connaître précisément quand un marché est ouvert. Un commerçant loue un emplacement à l'année pour vendre ses produits. Le système doit garder trace des locations d'emplacement. Pour chaque commerçant, le système doit mémoriser son numéro de SIRET, son nom ainsi que le type de produit vendu. Le prix de location annuel et la surface de chaque emplacement doivent être mémorisés par le système. Il faut aussi savoir si l'emplacement possède l'eau courante et/ou l'électricité.

D'autre part, le système doit permettre de rendre compte des équipes effectuant le nettoyage des marchés. Une équipe est basée à une adresse, il faut aussi mémoriser son numéro de téléphone. Les équipes sont très mobiles, elles peuvent travailler sur différents marchés, aucune n'est affectée à un marché particulier. Une équipe est composée de plusieurs ouvriers et peut posséder différents véhicules. Pour chaque ouvrier, le système doit mémoriser son numéro de sécurité sociale, son nom, son prénom ainsi que son adresse. Un des ouvriers est nommé responsable de l'équipe. Un véhicule est identifié par son numéro d'immatriculation, il est d'un certain type. Pour chaque véhicule, un ouvrier est désigné comme responsable, c'est lui qui se charge de vérifier le bon entretien de la machine.



13.1 On souhaite affecter les équipes à des marchés par jour. Comment s'assurer que les équipes seront affectées à des marchés uniquement leur jour d'ouverture?

13.2 Comment modifier le modèle si l'on considère que la numérotation des emplacements commence à 1 sur chaque marché?

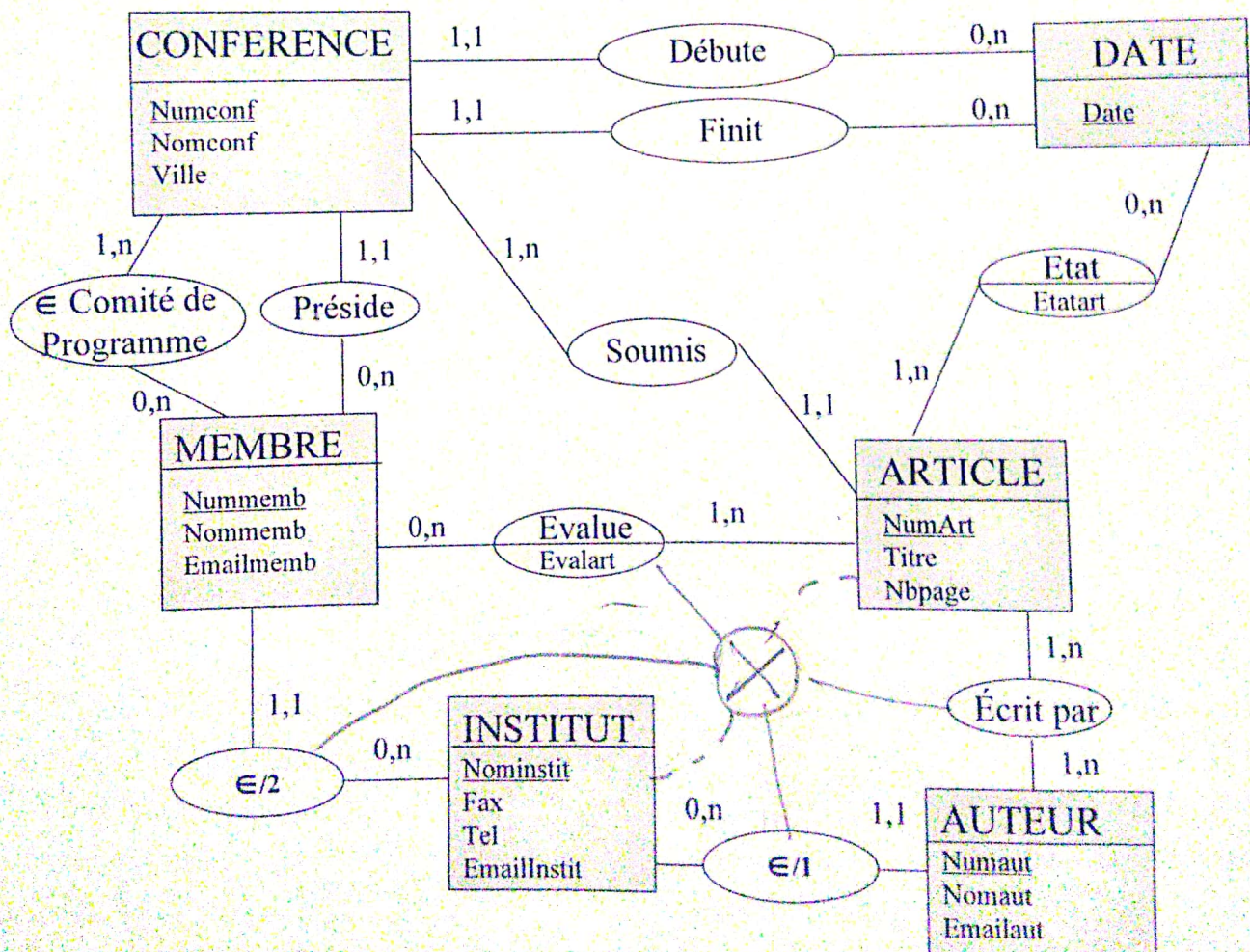
14) L'organisation de conférences

On s'intéresse à l'organisation des programmes scientifiques d'un ensemble de conférences. Les différentes personnes participant au programme scientifique (présidents de conférence, membres du comité de programme et auteurs d'articles) sont identifiées par un numéro unique. Le système d'information doit mémoriser pour chacune d'elle, leur nom, le nom de l'institution à laquelle il appartient, leur fax, téléphone et leur adresse de courrier électronique.

Une conférence est identifiée par un numéro unique, elle a lieu dans une ville, est décrite par un nom et se déroule entre deux dates. Une conférence a un président et un ensemble de personnes qui sont les membres du comité de programme chargés de l'évaluation des articles soumis à la conférence.

Un article est numéroté de manière unique, il possède un titre et a un nombre de pages. Un article peut être écrit par un ensemble d'auteurs qui sont des personnes, une personne peut soumettre plusieurs articles à une ou plusieurs conférences. L'état d'un article évolue au cours du temps (soumis, accepté, refusé, en attente), on veut conserver la trace de cette évolution.

Une fois soumis, un article est lu par plusieurs membres du comité de programme de la conférence. Chaque membre du comité de programme lit en moyenne sept articles. Le résultat de chacune de ces lectures est mémorisé dans la base de données sous forme d'une note comprise entre 0 et 20.



14.1: Un article ne doit pas être évalué par un membre de la commission appartenant au même institut (ou laboratoire) que l'un des auteurs de l'article.

$$1) \pi_{N^{\circ} \text{Article}}(\text{Acheter}) \cap \pi_{N^{\circ} \text{Article}}(\text{Approu}) = \emptyset$$

2)

$$\pi_{N^{\circ} \text{Commande}} \left[\sigma_{\text{Idclient IS NOT NULL}}(\text{Commande}) \right] \subset \pi_{N^{\circ} \text{Commande}}(\text{Baker})$$

3)

$$\pi_{N^{\circ} \text{Vehicule}}(\text{Vehicule}) \subset \pi_{N^{\circ} \text{Vehicule}} \left(\sigma_{\text{Idcontrat IS NOT NULL}}(\text{Vehicule}) \right)$$

$$\cup \pi_{N^{\circ} \text{vehicule}}(\text{Implique})$$

Autre solution:

$$\pi_{N^{\circ} \text{Vehicule}} \left(\sigma_{\text{Idcontrat IS NULL}}(\text{Vehicule}) \right) \subset \pi_{N^{\circ} \text{Vehicule}}(\text{Implique})$$

4)

a) Totalité

$$\pi_{N^{\circ} \text{Commande}}(\text{Commande}) \subset \left(\pi_{N^{\circ} \text{Commande}} \left(\sigma_{\text{Idclient IS NOT NULL}}(\text{Commande}) \right) \right)$$

$$\cup \pi_{N^{\circ} \text{Commande}} \left(\sigma_{\text{Idservice IS NOT NULL}}(\text{Commande}) \right)$$

b) Exclusion

$\Pi_{\text{N°commande}} (\sigma_{\text{idClient IS NOT NULL}} (\text{Commande})) \cap \Pi_{\text{N°commande}} (\sigma_{\text{idService IS NOT NULL}} (\text{Commande}))$

5)

$\Pi_{\text{idPersonne}} (\text{Vot}) \subset \Pi_{\text{idPersonne}} (\text{Souscription})$

6)

$\Pi_{\text{idProfesseur}, \text{idMatiere}} (\text{Enseigner}) \subset \Pi_{\text{idProfesseur}, \text{idMatiere}} (\text{Qualifier})$

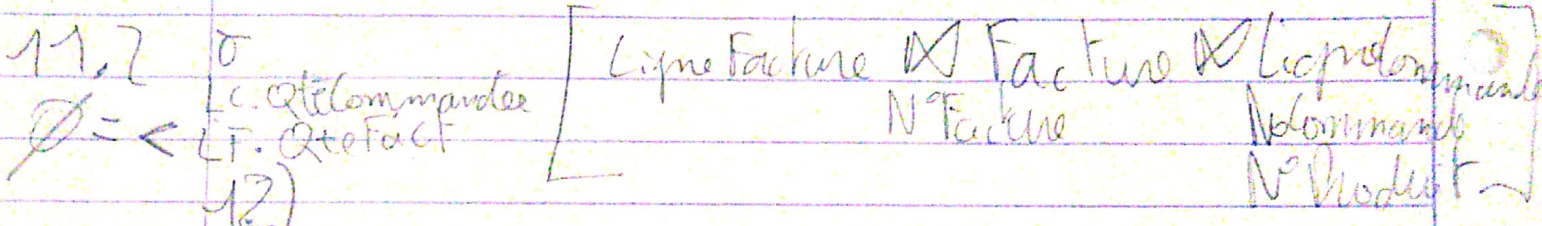
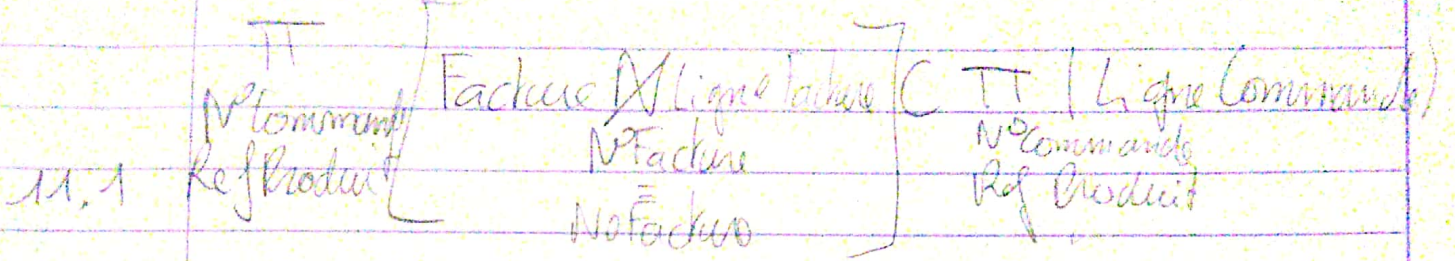
7)

$\Pi_{\text{idPersonne}, \text{idLogement}} (\text{Louer}) \cap \Pi_{\text{idPersonne}, \text{idLogement}} (\text{posséder}) = \emptyset$

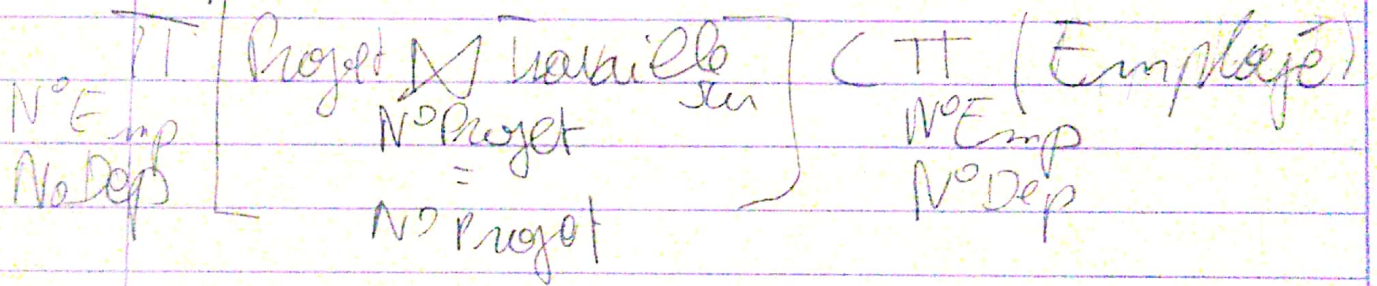
$\Pi_{\text{IdArticle}, \text{IdDepot}} (\text{compose}) \bowtie \Pi_{\text{IdCommande}, \text{IdCommande}} (\text{Commande}) \subset \Pi_{\text{IdArticle}, \text{IdDepot}} (\text{Stock})$

9) Réserve (# N°Chambre, # Date, # Numéro)

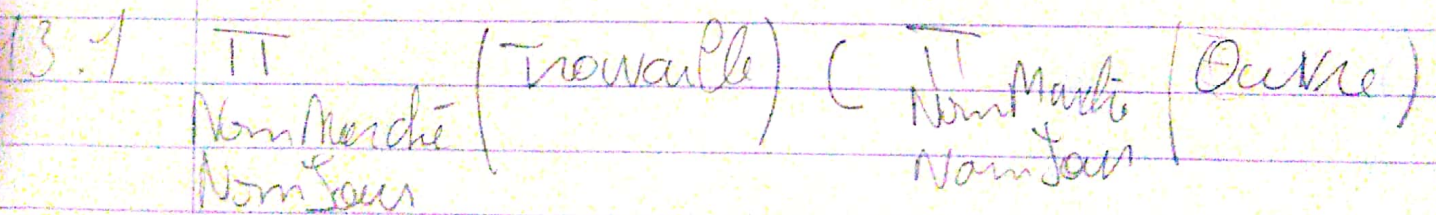
11)



12)

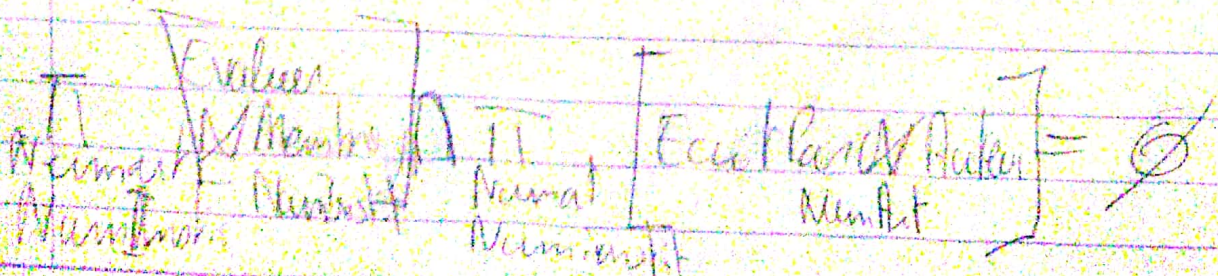


13)



13.7 EMPACEMENT

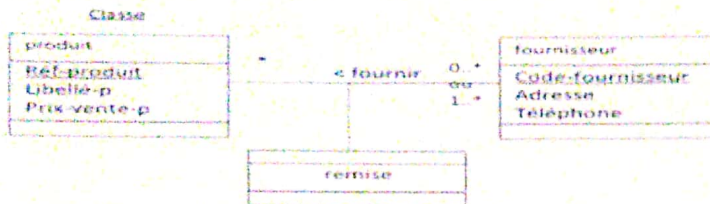
14)



TD 5 et 6 Transformation Diagrammes de Classes en Modèle Relationnel

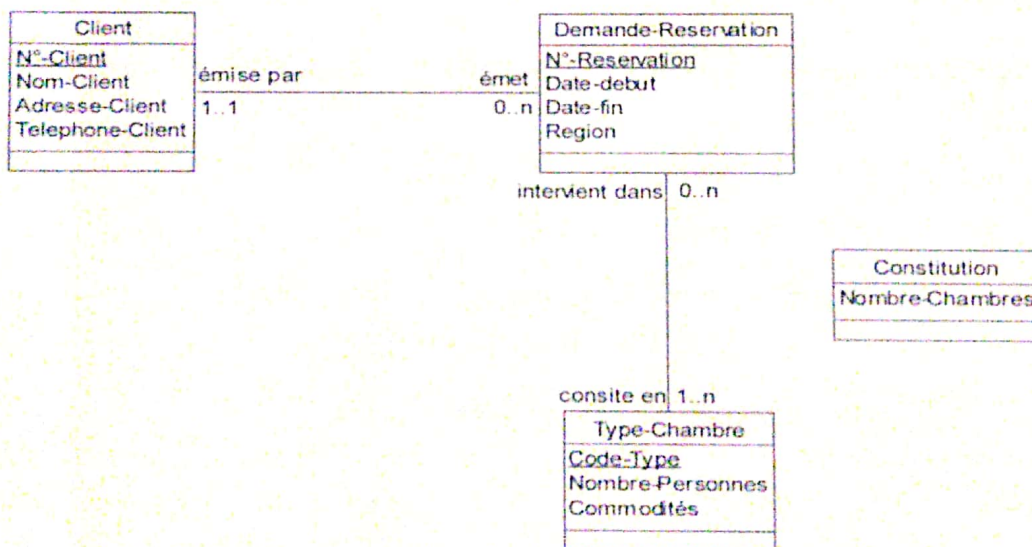
Exercice 1

Construire le modèle relationnel correspondant au diagramme de classe suivant:



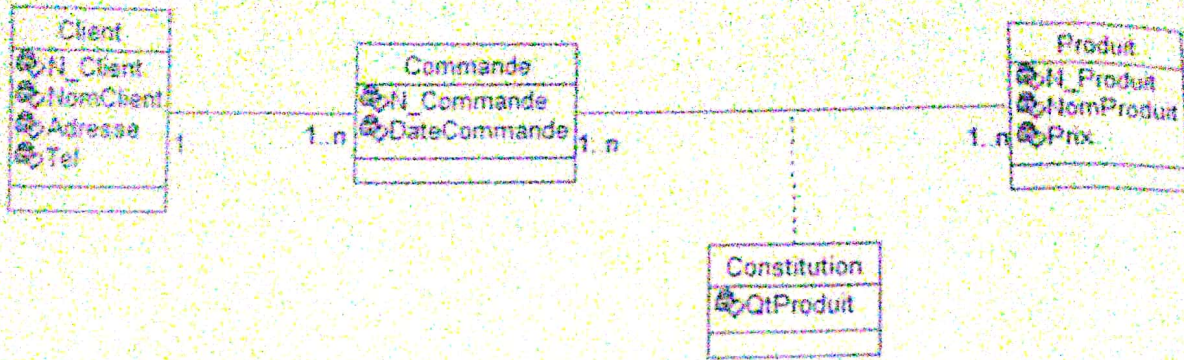
Exercice 2

Construire le modèle relationnel correspondant au diagramme de classe suivant:



Exercice 3

-Construire le modèle relationnel correspondant au diagramme de classe suivant:



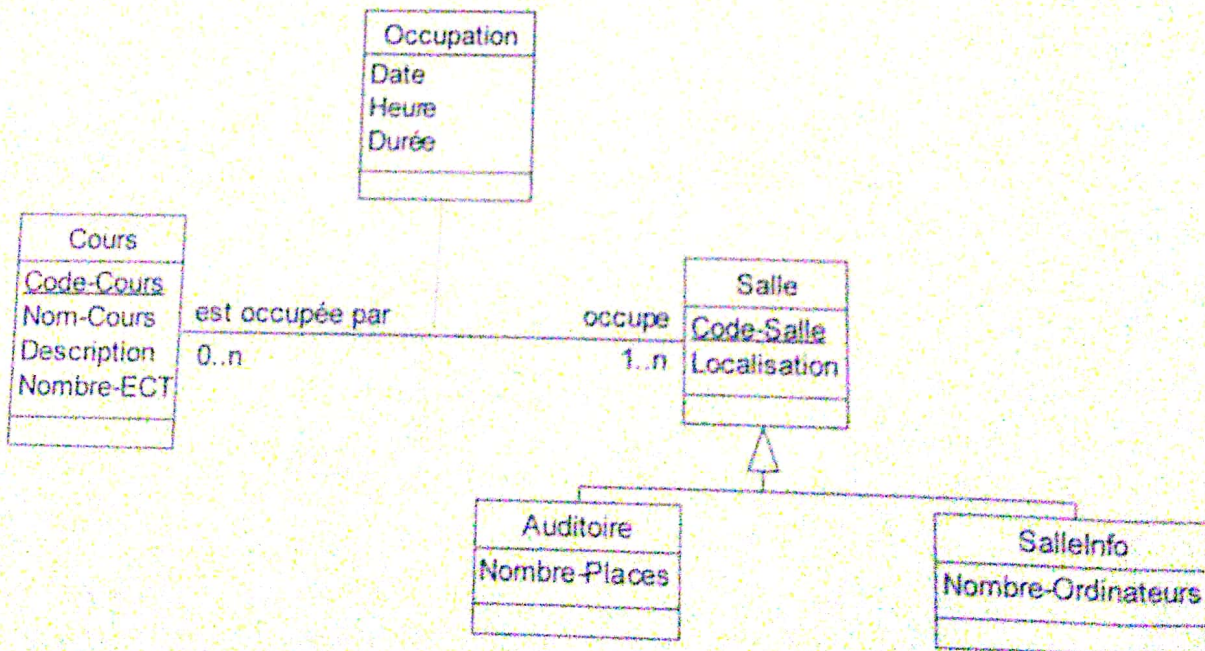
-Répondre aux requêtes suivantes pour les tables suivantes

- Quel est le *prix* et *numéro produit* du *Samsuung S4*
- Quels sont les clients (*nom*) qui habitent à *Bruxelles*
- Quels sont les clients (*nom* et *adresse*) qui ont achetés de produits le *12/01/2012*
- Qui (*nom* et *adresse*) ont achetés le chocolat *Leonidas*
- Quels sont les produits (*nomProduit*) achetés par *Toto*

-Quelles sont les contraintes d'intégrités

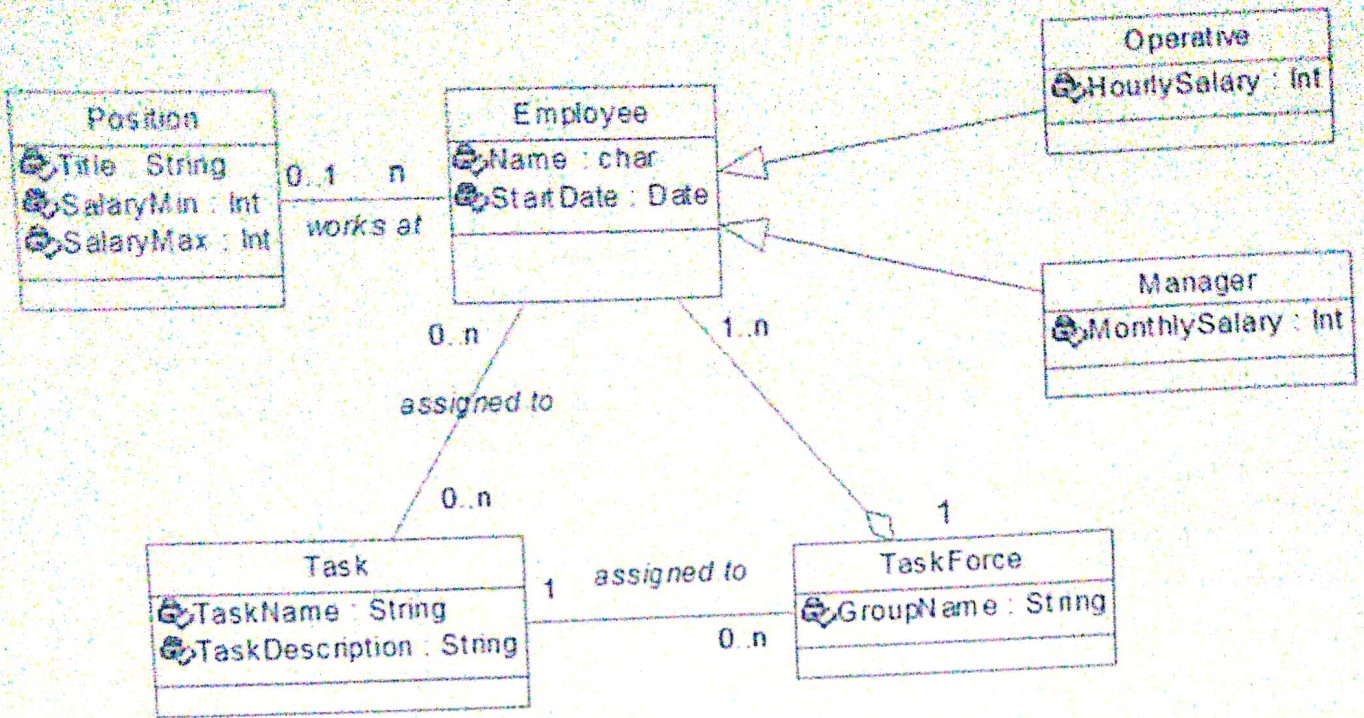
Exercice 4

Construire le modèle relationnel correspondant au diagramme de classe suivant:



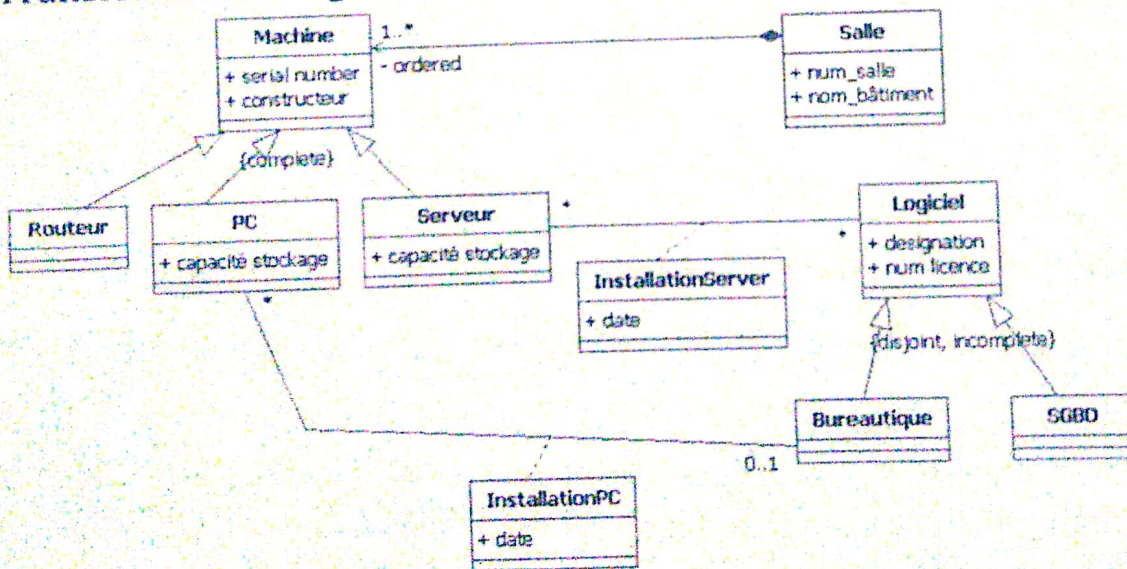
Exercice 5

Même question



Exercice 6

Transformer le diagramme de classe en modèle relationnel



Exercice 1

PRODUIT (Réf-produit, Libellé, Prix Vente,
* Code-Fournisseur)

FOURNISSEUR (Code-Fournisseur, Adresse,
Téléphone)

PROMOTION (* Réf-produit, * Code-Fournisseur,
Remise)

Exercice 2

CLIENT (N°-Client, Nom-Client, Adresse-
Client, Téléphone-Client)

DEMANDE - RÉSERVATION (N°-Réservation,
Date-debut, Date-fin, Région, * N°-Client)

TYPE-CHAMBRES (Code-Type, Nombre-
Personnes, Commodités)

CONSTITUTION (* Code-Type, * Numéro-
Réservation, Nombre-Chambres)

Exercice 3

CLIENT (N-Client, NomClient, Adresse, Tel)

COMMANDE (N-Commande, DateCommande,
* N-Client)

PRODUIT (N-Produit, NomProduit, Prix)

CONSTITUTION (* N-Commande, * N-Produit,
QteProduit)

1) SELECT N.Produit, Prix FROM produit
WHERE NomProduit = 'Galaxy S10';

2) SELECT nom FROM client
WHERE Adresse LIKE '%Bruxelles%';

3) SELECT C.Nom, Adresse FROM client C,
Commande Com
WHERE Com.N.Client = C.Client
AND Com.DateCommande = '12/01/2012';

4) SELECT C.nom, Adresse FROM
client C, Commande Com, Constitution
Const, Produit P;
WHERE C.N.Client = Com.N.Client
AND Com.N.Commande = Const.N.Commande
AND Const.N.Produit = P.N.Produit
AND P.NomProduit = 'Leonidas';

5) SELECT P.nomProduit FROM
client C, Commande Com, Constitution
Const, Produit P,
WHERE P.N.Produit = Const.N.Produit
AND Const.N.Commande = Com.N.Commande
AND Com.N.Client = C.N.Client
AND C.nomClient = 'Toto';

Exercice 4

COURS (Code-Cours, Nom-Cours, description, Nombre-ECT)
 OCCUPATION (X Code-Cours, X Code-Salle, Date, Heure, Durée)
 SALLE (Code-Salle, Localisation)
 AUDITOIRE (X Code-Salle, Nombre-Places)
 SALLEINFO (X Code-Salle, Nombre-Ordinateurs)

Exercice 5

POSITION (Title, SalaryMin, SalaryMax)
 EMPLOYEE (Name, startDate, X GroupName)
 OPERATIVE (X Name, HourlySalary)
 MANAGER (X Name, MonthlySalary)
 TASK (TaskName, TaskDescription)
 TASKFORCE (GroupName, X TaskName)
 ASSIGNEMENT (X Name, X TaskName)

Exercice 6

SALLE (num_salle, nom_batiment)
 MACHINE (SerialNumber, Constructeur, X num_salle)
 ROUTEUR (X SerialNumber)
 PC (X SerialNumber, Capacité_Stockage)
 INSTALLATION_PC (X SerialNumber, X Designation, date)
 SERVEUR (X SerialNumber, Capacité_Stockage)
 INSTALLATION_SERVEUR (X SerialNumber, X Designation, date)
 LOGIPEL (Designation, num_casa)

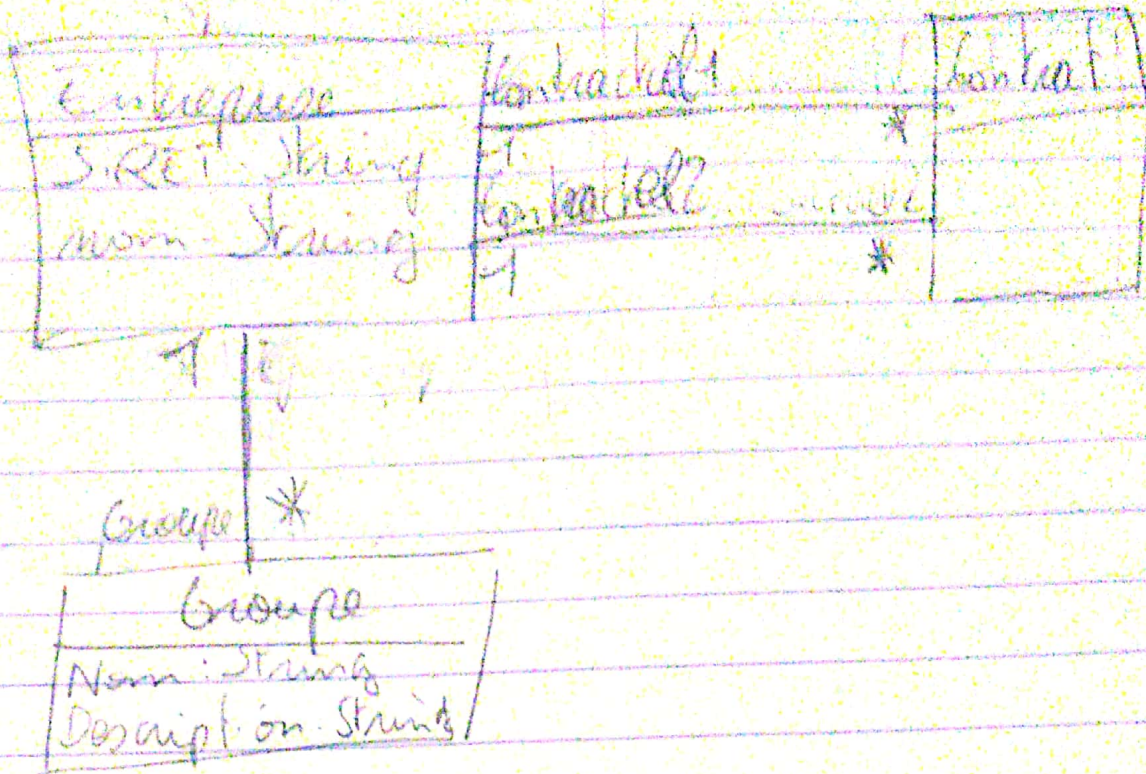
COCREATIVE (* Designation)
SGOD (* Designation)

Exercice – Expression de contraintes OCL

1. Un vol relie deux villes : une ville de départ et une ville d'arrivée. La ville de départ et la ville d'arrivée d'un vol doivent être différentes.
2. Un vol est assuré par un avion ayant une certaine capacité (nombre maximum de places). Le nombre de réservations enregistrées pour un vol ne doit pas dépasser la capacité de l'avion utilisé.
3. Un contrat est passé entre deux entreprises (SIRET, nom entreprise). Une entreprise appartient à un groupe (nom du groupe, description).
 - a. Une entreprise ne peut pas passer un contrat avec elle-même.
 - b. Un contrat ne peut pas être passé entre deux entreprises du même groupe.
4. Plusieurs opérations (nature, montant, date) peuvent être enregistrées sur un compte bancaire (n°, date).
 - a. Le montant des opérations enregistrées sur un compte ne peut pas être nul.
 - b. La date de création d'un compte doit être antérieure à la date d'enregistrement de toutes les opérations le concernant.
 - c. Seules les opérations de virement nécessitent un compte émetteur et un compte bénéficiaire.
5. Une tâche est affectée à des développeurs et des testeurs. Tout développeur dans une tâche est aussi testeur dans cette tâche.
6. Un employé est affecté à un service qui possède des projets. Un employé ne peut travailler que sur les projets de ses services.
7. Lorsqu'on crée un nouveau projet, on lui affecte des employés (nom, salaire). Un chef de projet est désigné parmi les employés affectés au projet. Le salaire d'un employé ne peut pas dépasser celui de son chef de projet. Nous supposons qu'un projet peut avoir plusieurs chefs de projet.

Travail demandé

Faire un diagramme de classes et représenter les contraintes en OCL.



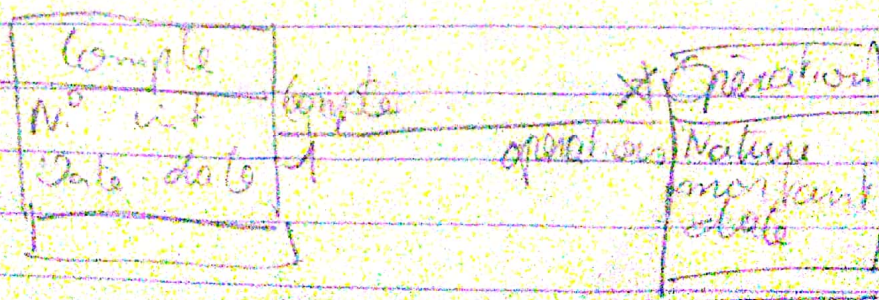
3) a)

context = Contrat
 Invariant = self.contratuel1 <>
 self.contratuel2

b)

context = Contrat
 Invariant = self.contratuel1.groupe
 <> self.contratuel2.groupe

c)



a)

context: Compte
 invariant: self.operation For All (o.operation / o.montant > 0)

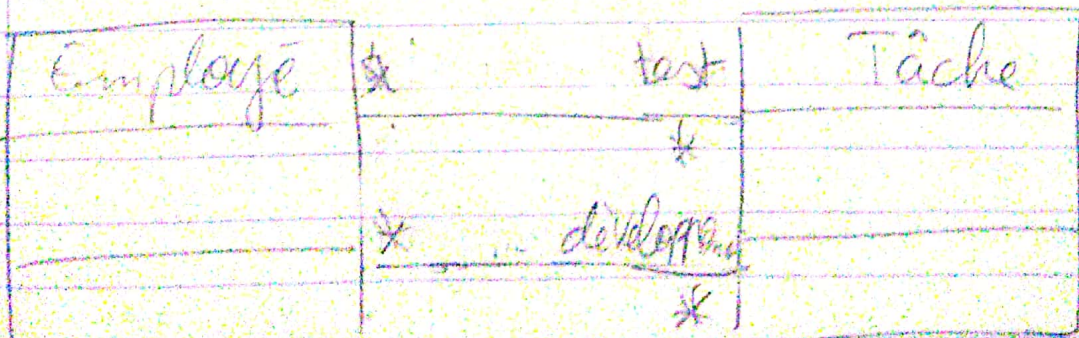
b)

context: compte
 invariant: self.operation For All (o.operation / o.date < self.date)

c)

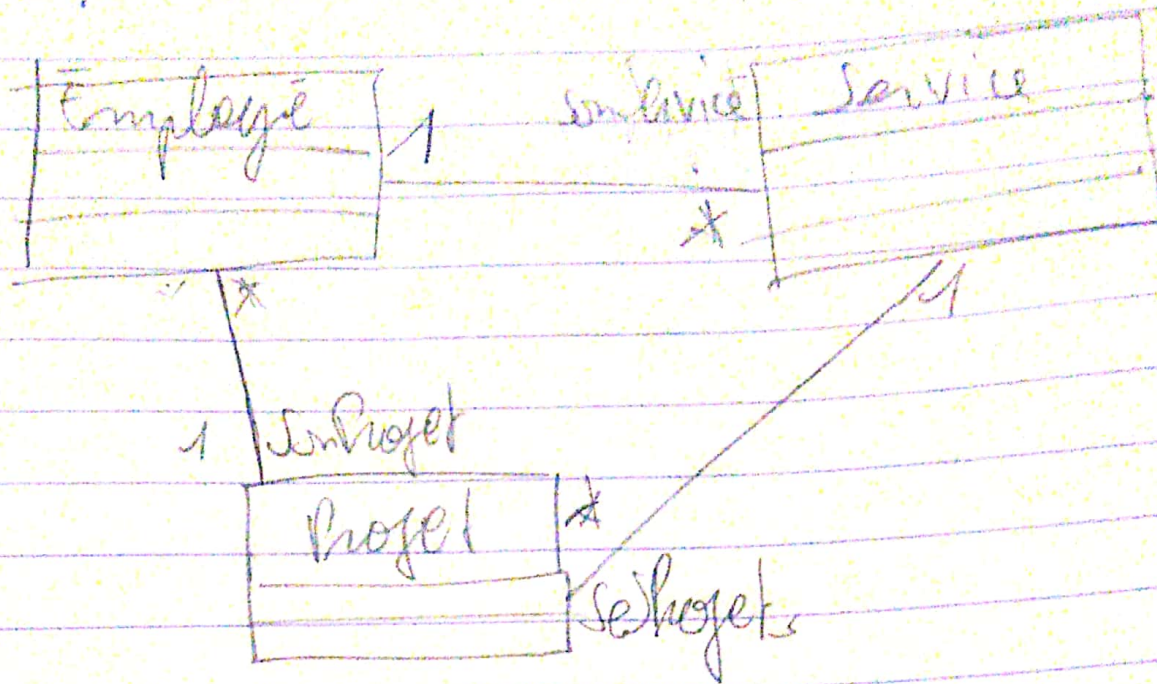
context:
 invariant:

d)



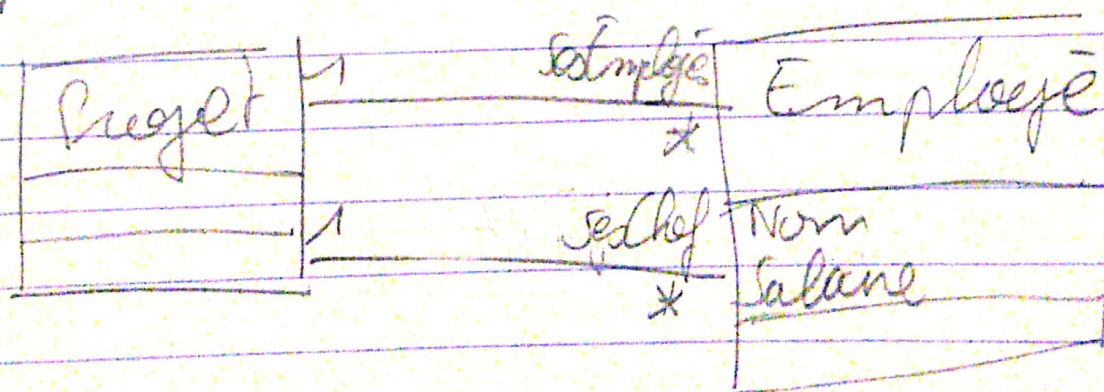
context: employé
 invariant: self.test = self.developpement

6)



Context = Employee
 invariant = self, son Service, sesProjets
 includes All (self, sonChef)

7)



Context: Projet
 invariant: self, sesEmployes, for All
 (e: Employee / self, sesChef, for All (c:
 Employee / e. salaire < c. salaire))