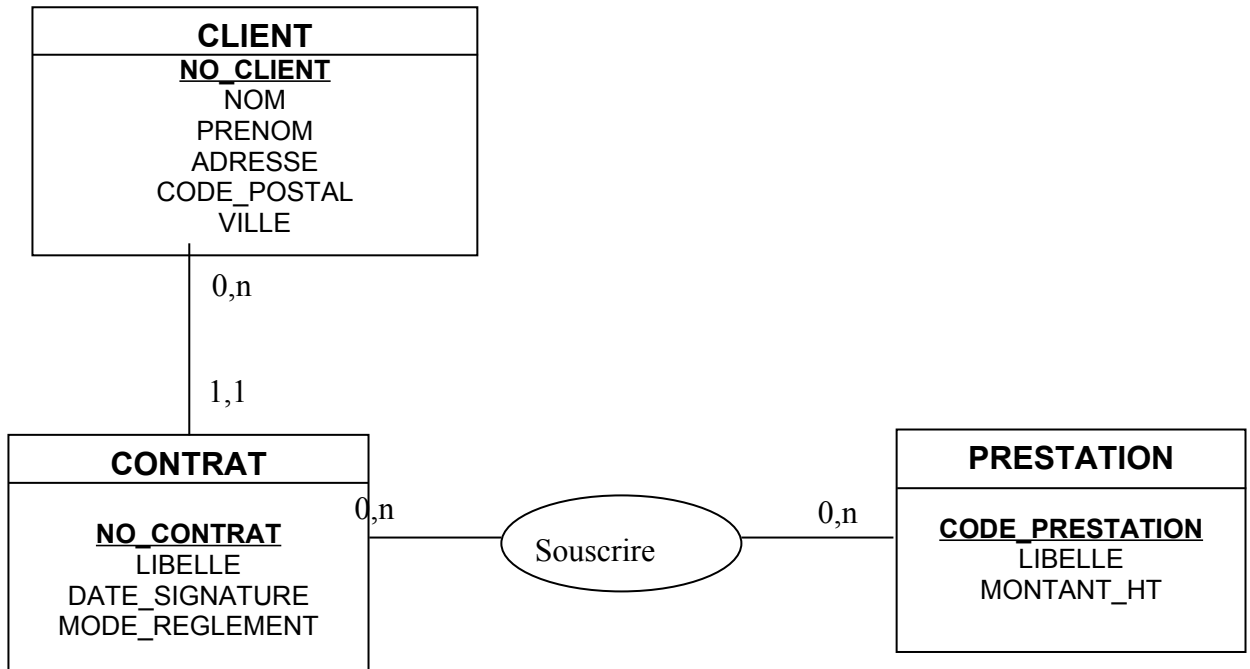


Exercice 1

1. Modèle conceptuel de données

Le besoin de gestion des contrats des clients d'une société est modélisé avec ce schéma Entité / Association :



1. Est-ce que dans ce modèle un client peut-il avoir plusieurs contrats ?

Solution : Oui, cardinalité maximale n sur CLIENT - CONTRAT

2. Est-ce qu'il peut y avoir un client qui n'a pas de contrats ?

Solution : Oui, cardinalité minimale 0 sur CLIENT - CONTRAT

3. Peut-on souscrire plusieurs prestations dans le même contrat ?

Solution : Oui, Cardinalité maximale n sur CONTRAT - PRESTATION

4. Peut-on trouver la même prestation dans plusieurs contrats ?

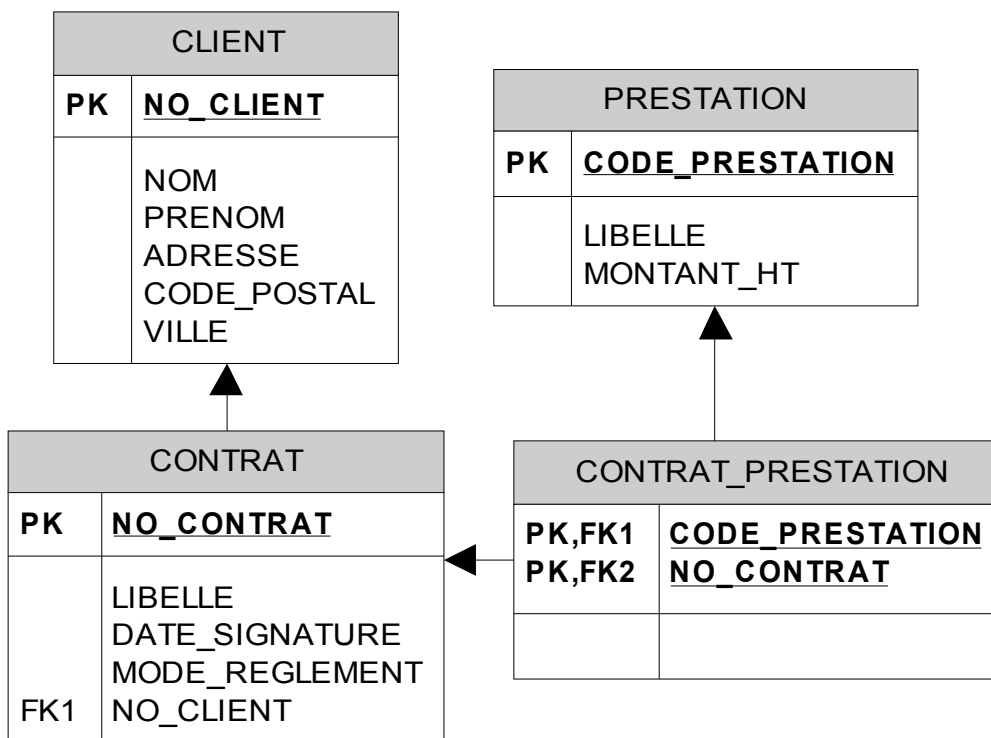
Solution : Oui, Cardinalité maximale n sur PRESTATION - CONTRAT

2. Modèle relationnel de données

Déduire à partir du modèle conceptuel précédant le modèle relationnel, en indiquant pour chaque relation :

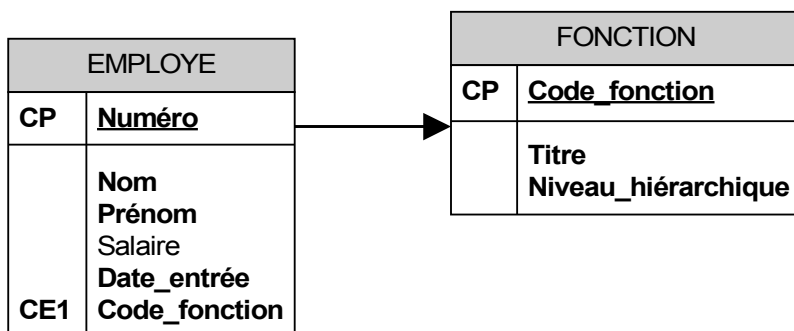
- Les attributs
- La clé primaire
- Les clés étrangères

Solution :



3. Algèbre relationnel

Soit le modèle relationnel qui gère les fonctions des employés dans une société :



Une fonction est identifiée par un code et possède un titre et un niveau hiérarchique.

Un employé est identifié par un numéro et possède les informations suivantes :

- Nom : Nom de l'employé
- Prénom : Prénom de l'employé
- Salaire : Salaire de l'employé en €
- Date_entrée : Date d'entrée de l'employé dans la société
- Code_fonction : Code de la fonction occupée par l'employé (Clé étrangère avec la table FONCTION)

Exprimer les requêtes suivantes en algèbre relationnel :

1. Les différents codes fonctions dans la relation EMPLOYE.

Solution : $\Pi_{\text{Code_fonction}} (\text{EMPLOYE})$

2. Nom et Prénom des DBA qui ont un salaire supérieur à 3000 dans la relation EMPLOYE.

Solution : $\Pi_{\text{Nom}, \text{Prénom}} (\sigma_{\text{Salaire} > 3000 \wedge \text{CODE_FONCTION} = \text{DBA}} (\text{EMPLOYE}))$

3. Nom, Prénom et titre de fonction de tous les employés

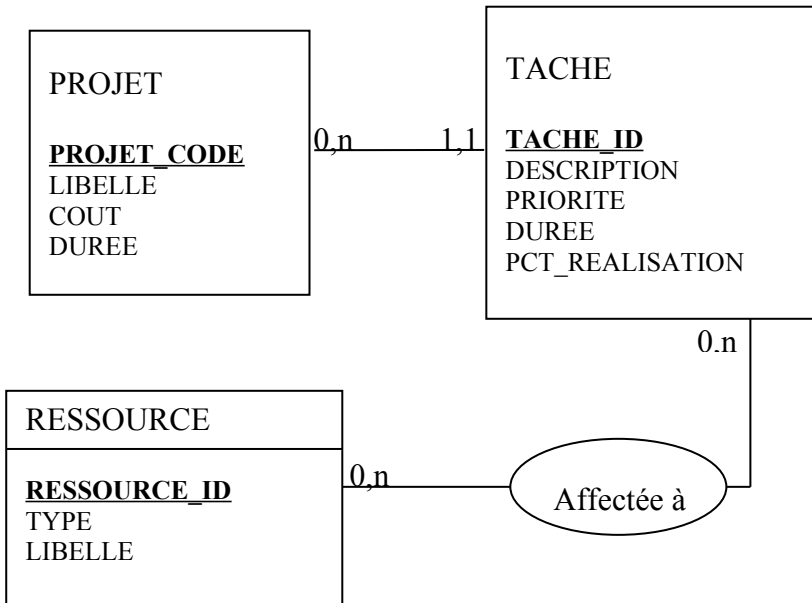
Solution : $\Pi_{\text{Nom, Prénom, Titre}}(\text{EMPLOYEE} \bowtie \text{FONCTION})$

Exercice 2 :

1. Modèle conceptuel de données

Description du schéma :

Une entreprise informatique souhaite gérer ses projets et ses ressources dans son système d'information. Le besoin de la société est modélisé dans le schéma Entité / Association suivant :



Un projet est identifié par un code, et possède un libelle, un coût en Euro et une durée en mois. Une ressource a un identifiant, elle est de type matériel ou travail et un libelle. Une tâche a une description, une priorité, une durée en jours et un pourcentage de réalisation.

Questions :

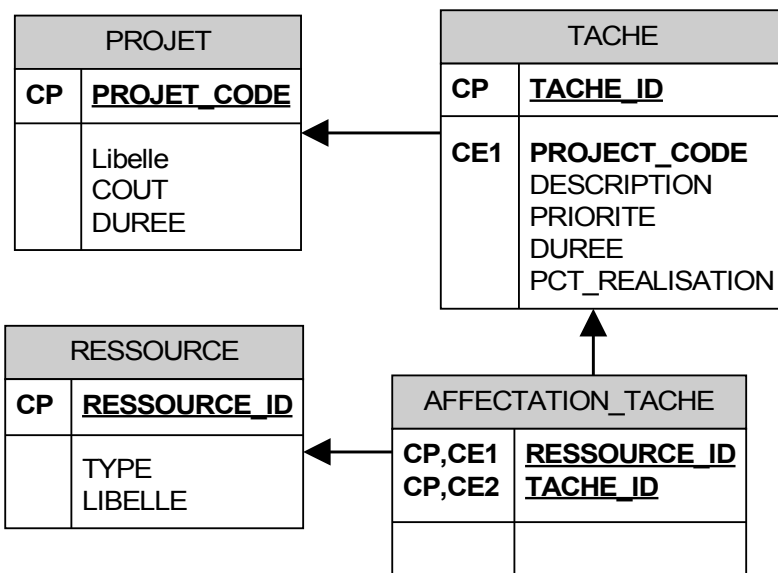
1. Est-ce qu'une ressource peut travailler sur plusieurs tâches à la fois ?
Solution : Oui, cardinalité maximale n sur RESSOURCE - TACHE
2. Est-ce qu'une tâche peut être affectée à plusieurs ressources ?
Solution : Oui, cardinalité maximale n sur TACHE - RESSOURCE
3. Une tâche peut-elle être dans plusieurs projets ?
Solution : NON, Cardinalité maximale 1 sur TACHE - PROJET
4. Est-ce qu'on pourrait représenter dans ce modèle un projet qui ne contient aucune tâche ?
Solution : Oui, Cardinalité minimale 0 sur PROJET – TACHE.

2. Modèle relationnel :

Déduire à partir du modèle conceptuel précédant le modèle relationnel, en indiquant pour chaque relation :

- Les attributs
- La clé primaire
- Les clés étrangères

Solution :



3. Algèbre relationnel

Donner les requêtes suivantes en langage algébrique :

1. Les codes et libelles des projets dépassant une durée de 60 mois et un coût de 100000 €.

Solution : $\Pi_{\text{CODE}, \text{LIBELLE}} (\sigma_{\text{DUREE} > 60 \wedge \text{COUT} > 100000} (\text{PROJET}))$
2. Les id des ressources et descriptions de leurs tâches, n'afficher que les tâches avec un pourcentage de réalisation inférieur à 50 %

Solution : $\Pi_{\text{DESCRIPTION}, \text{LIBELLE}} (\sigma_{\text{PCT_REALISATION} < 50} (\text{TACHE}) \bowtie \text{RESSOURCE})$
3. Afficher les libelles des ressources qui ne travaillent sur aucune tâche

Solution : $\Pi_{\text{LIBELLE}} (\text{RESSOURCE} - (\text{RESSOURCE} \bowtie \text{AFFECTATION_TACHE}))$