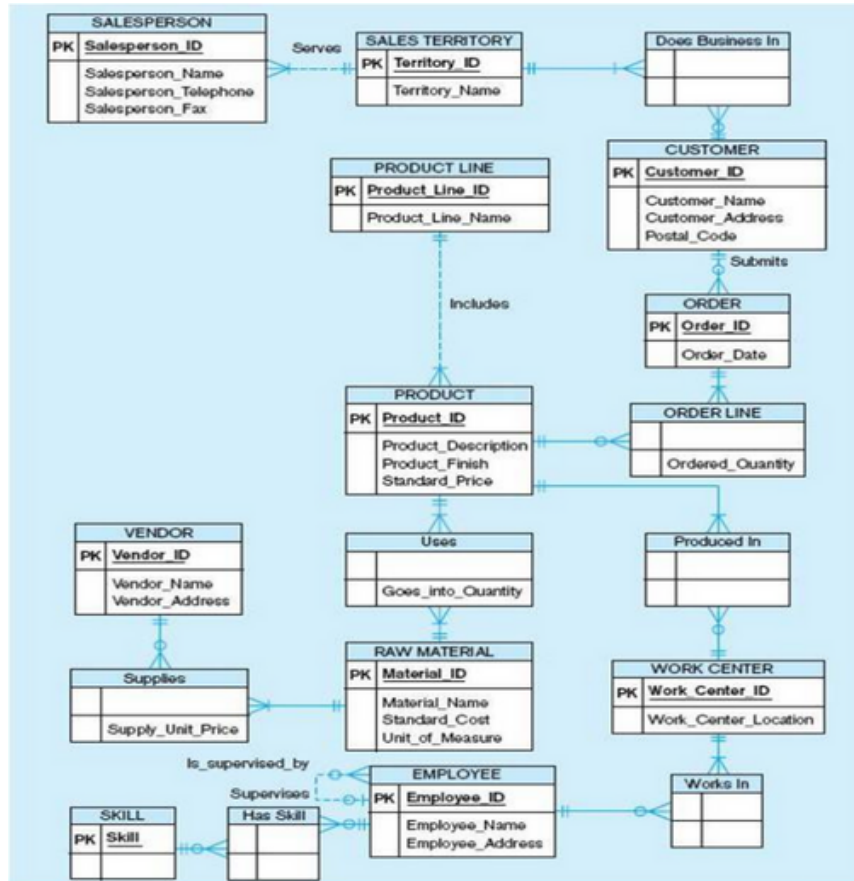


## Chapitre 3: Exercices en ordre comme mes notes

Ch3. Problems and exercises. [MgtDB Jeffrey P 177]

Exo 1: Feuille 8:



1. Répondez aux questions suivantes concernant le modèle de données pour Pine Valley Company (Fig.1 jointe)

a. Nommez une relation unaire, que signifie-t-elle et pour quelles raisons les cardinalités pourraient-elles être différentes dans d'autres organisations ?

Employee est une relation Unaire (1-To-Many).

b. Pourquoi "Include" est-il une relation un-à-plusieurs et pourquoi cela pourrait-il être différent dans d'autres organisations ?

C'est obligatoire d'avoir au moins un produit pour la création d'un product line. Donc product line n'existe pas s'il n'y a pas au minimum un produit.

d. Supposons qu'il y ait une personne de contact client différente pour chaque territoire de vente dans lequel un client a fait des affaires, où placerions-nous le nom de cette personne dans le modèle de données ?

On la place dans la table 'DoesBusiness'.

f. De quelle manière Pine Valley pourrait-elle changer sa façon de faire des affaires, ce qui entraînerait l'élimination de l'entité associative d'approvisionnement et la modification des relations autour d'elle ?

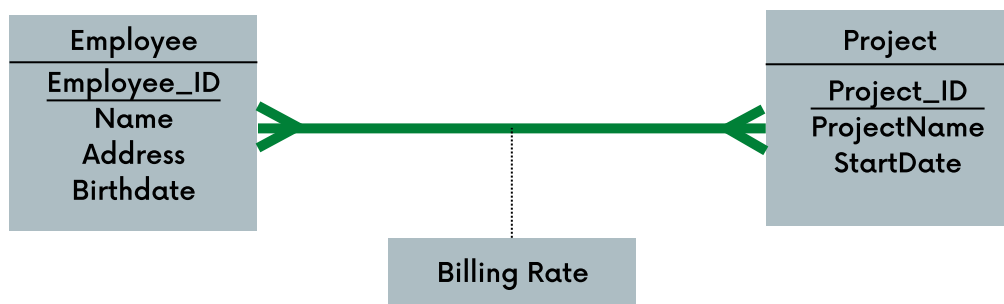
Entre vendors, supplies et Raw Material (1-To-Many instead of Many-To-Many) Chaque Raw Material possède son propre vendor.

#### Exercise #5 [p177]

a. Dessinez un ERD pour chacune des situations suivantes.

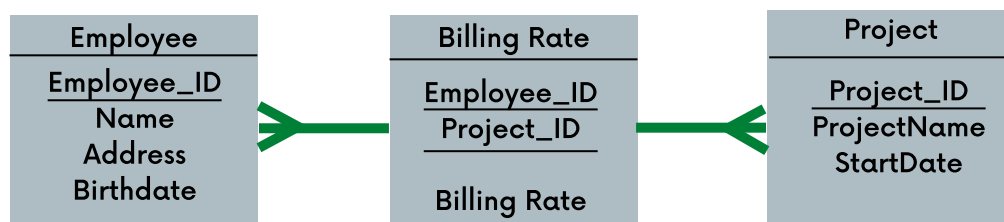
Une entreprise compte plusieurs salariés. Les attributs de EMPLOYEE incluent Employee\_ID (identifiant), Name, Address et Birthdate. L'entreprise a plusieurs projets. Les attributs de PROJECT incluent Project\_ID (identifiant), Project\_Name et Start\_Date. Chaque employé peut être affecté à un ou plusieurs projets ou ne pas être affecté à un projet. Un projet doit avoir au moins un employé affecté et peut avoir n'importe quel nombre d'employés affectés. Le taux de facturation d'un employé peut varier selon le projet, et l'entreprise souhaite enregistrer le taux de facturation applicable (Billing\_rate) pour chaque employé lorsqu'il est affecté à un projet particulier.

#### ERD - Many-To-Many



Attribut de la relation

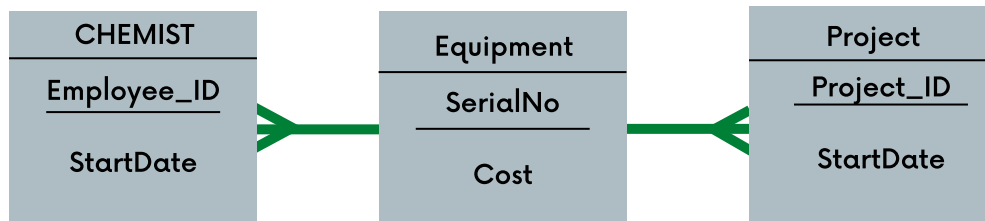
#### Logical Design



La relation Many-To-Many se transforme en une table intermédiaire 1-To-Many, qui possède comme clé, Employee\_ID et Project\_ID (aussi comme foreign keys) et les attributs de la relation donnés par hypothèse.

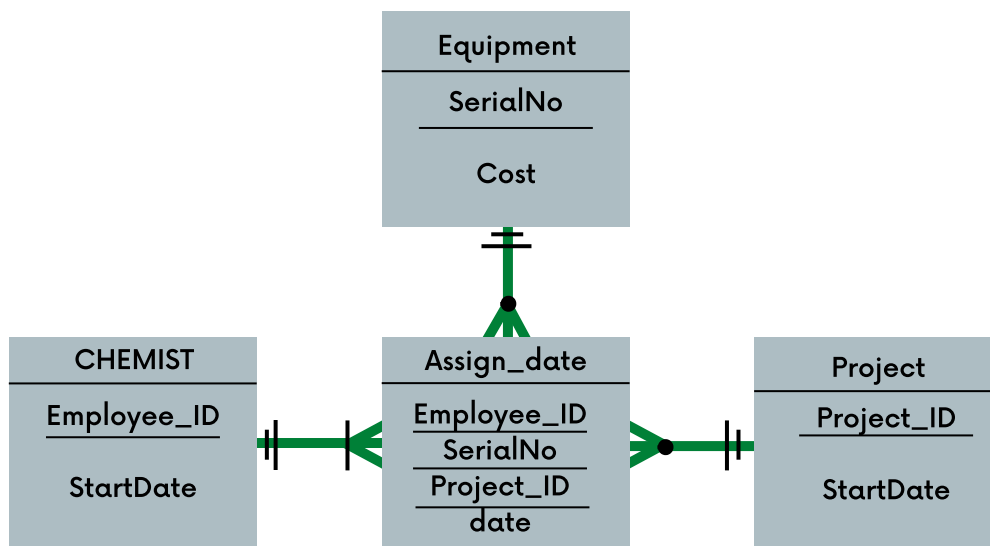
b. Un laboratoire compte plusieurs chimistes qui travaillent sur un ou plusieurs projets. Les chimistes peuvent également utiliser certains types d'équipements sur chaque projet. Les attributs de CHEMIST incluent EMPLOYEE\_ID (identifiant) et Start\_Date. Les attributs de PROJECT incluent PROJECT\_ID (identifiant) et Start\_Date. Les attributs de EQUIPMENT incluent Serial\_No et Cost. L'organisation souhaite enregistrer Assign\_Date, c'est-à-dire la date à laquelle un élément donné a été affecté à au moins un projet et un équipement. Un équipement donné n'a pas besoin d'être affecté, et un projet donné n'a pas besoin d'être affecté ni à un chimiste ni à un équipement. Fournissez de bonnes définitions pour toutes les relations dans cette situation.

ERD - Many-To-Many ternaire:

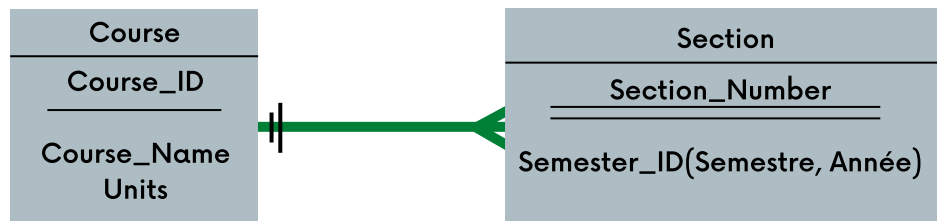


Une relation ternaire est toujours Many-To-Many donc on va créer une nouvelle table intermédiaire qui possède comme clé primaire les 3 autres clés primaires des autres tables. Et qui vont aussi être des foreign keys. Many-To-Many va être 3 relations 1-To-Many.

Logical Design



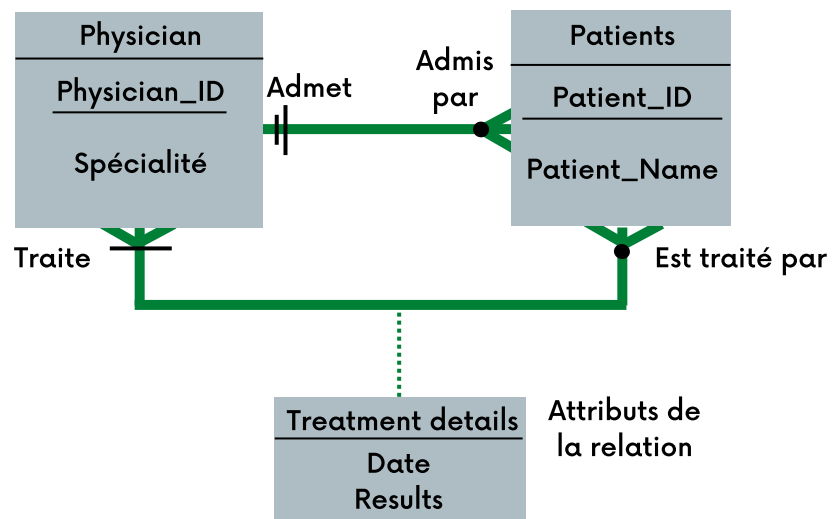
c. Un cours peut avoir une ou plusieurs sections programmées, ou peut ne pas avoir de section programmée. Les attributs de COURSE incluent Course\_ID, Course\_Name et Units. Les attributs de SECTION incluent Section\_Number et Semester\_ID. Semester\_ID est composé de deux parties : semestre et année. Section\_Number est un nombre entier qui distingue une section d'une autre pour le même cours, mais n'identifie pas une section de manière unique. Comment avez-vous modélisé SECTION ?



Explication: L'attribut Section Number va définir plusieurs sections du même cours et non pas une section entière seule, donc cet attribut va dépendre de l'entité Course. Donc Section possède une clé partielle. C'est le cas d'une weak entity. Enfin, pour représenter le logical design, c'est le cas d'une relation 1-To-Many donc on ajoute la clé primaire de l'entité course à section, en tant que clé étrangère.

d. Un hôpital compte un grand nombre de médecins agréés. Les attributs de PHYSICIAN incluent Physician\_ID (l'identifiant) et la spécialité. Les patients sont admis à l'hôpital par des médecins. Les attributs de PATIENT incluent Patient\_ID (l'identifiant) et Patient\_Name. Tout patient admis doit avoir minimum un médecin traitant. Un médecin peut éventuellement admettre n'importe quel nombre de patients. Une fois admis, un patient donné doit être traité par au moins un médecin. Un médecin particulier peut traiter n'importe quel nombre de patients ou ne pas traiter de patients. Chaque fois qu'un patient est traité par un médecin, l'hôpital souhaite enregistrer les détails du traitement (traitement\_détails). Les composants de treatment\_Detail incluent la date, l'heure et les résultats. Avez-vous établi plus d'une relation entre le médecin et le patient ? Pourquoi ou pourquoi pas?

ERD:



Pour le logical design: la relation Many-To-Many est décomposée en 2 1-To-Many relationships:

