Introduction à la normalisation

Les trois premières formes normales.

Préparé par Saliha Yacoub

Introduction

- Définition: une base de données est dite normalisée si elle est au moins en troisième forme normale (3 FN)
- Pourquoi normaliser ?
  - Pour réduire la redondance de données. → facilité de mise à jour
  - Avoir une bonne structure des données
  - Mettre en évidence les relations entre les données.
- Comment normaliser une base de données ?

On procède à la normalisation en appliquant les règles de normalisation. C'est-à-dire

  - D'abord obtenir la première Forme Normale (1FN)
  - Ensuite obtenir la deuxième Forme Normale (2FN)
  - Et enfin obtenir la troisième Forme Normale (3FN)
- Il y a d'autres FN (FNBC pour la forme normal de Codd,4FN,5FN,). Ces FN ne seront pas abordées ici. On arrête à la 3FN.

La 1FN (1/6)

- La première Forme Normale est appelée et notée 1FN
- La 1FN est appelée la CLÉ.
- La 1FN réduit la redondance
- Une base de données est dite en 1FN si toutes ses tables sont en 1FN

Énoncé 1: Une table est en 1FN si et seulement si:

1. La table a une clé primaire.
2. Aucune donnée de la table n'est calculée.
3. Les données doivent être élémentaires .
4. La table ne doit pas contenir de groupe de données répétitif ce qui revient à dire que tout les attributs de la table ne sont pas multivalués

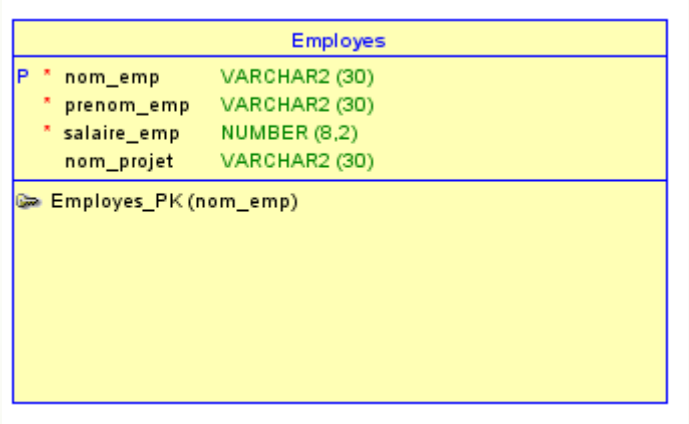
Un attribut est multivalué, s'il peut prendre plusieurs valeurs pour un enregistrement.

Un attribut est élémentaire s'il n'est pas composé

La 1FN (2/6)

- Exemple 1: **Le problème**

Pour la table Employes suivante, on voit bien qu'elle n'est pas en 1FN. Pourquoi ?

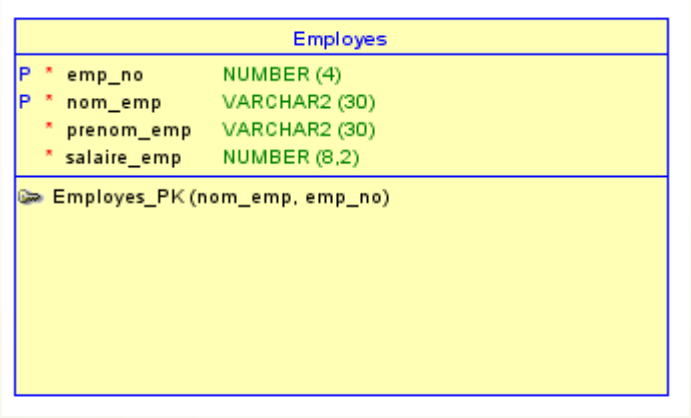


1. Le nom_emp ne peut être une clé primaire car il ne peut pas être UNIQUE
2. nom_projet constitue une donnée répétitive car un employé travaille sur plusieurs projets. En un mot nom_projet peut avoir plusieurs valeurs (multivalué). Par exemple pour l'employés Patoche, il peut avoir 5 projets différents.

La 1FN(3/6)

- Exemple 1: **La solution**

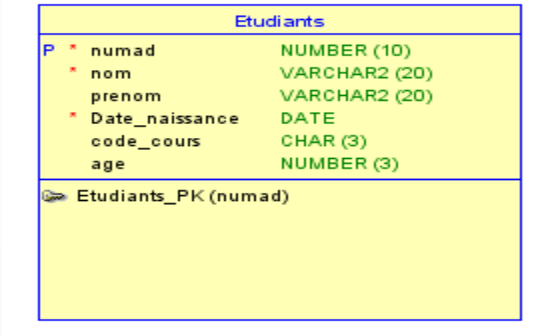
1. Mettre une clé primaire à la table Employés. La clé primaire a la contrainte d'être unique et not null
2. Sortir nom_projet puisqu'il est redondant. (il peut avoir plusieurs valeurs)



La 1FN(4/6)

- Exemple 2: **le problème**

Pour la table Etudiants suivante, on voit bien qu'elle n'est pas en 1FN. Pourquoi ?

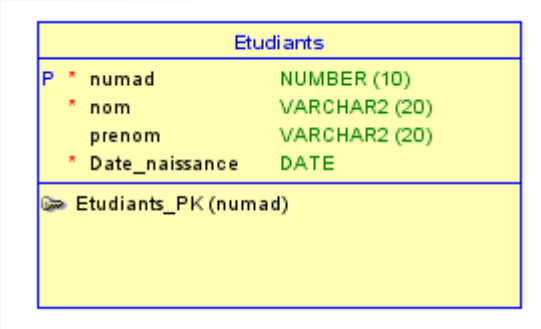


1. Le code_cours est multivalué. Peut prendre plusieurs valeurs pour un etudiants, puisqu'un étudiant suit PLUSIEURS cours. Le code_cours est répétitif. Pensez à vous. Dans combien de cours vous êtes inscrits ? Si vous répondez plus que 1 alors, le code_cours est redondants. Vous êtes dans KB6,KB4,KB5 etc..
2. L'âge de l'étudiant est calculé

La 1FN (5/6)

- Exemple 2: **La solution**

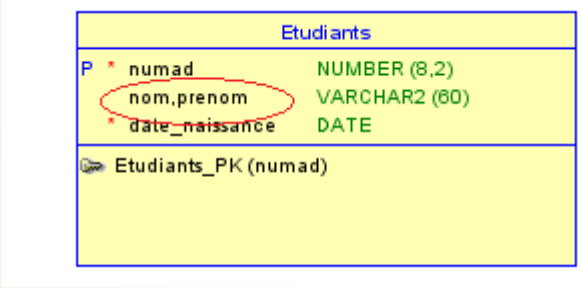
1. Sortir le code_cours puisqu'il est redondant, (multivalué).
2. Enlever age, puisqu'il est calculé



La 1FN (6/6)

- Exemple 3: **Le problème**

La table Etudiants suivante n'est pas en 1FN. Pourquoi ?



1. Car, l'attribut(nom, prenom) n'est pas élémentaire. Il est composée

Exemple 3**: La solution**

Avoir un attribut nom et un attribut prenom (comme on l'a toujours vu en classe)

La 2FN (1/4)

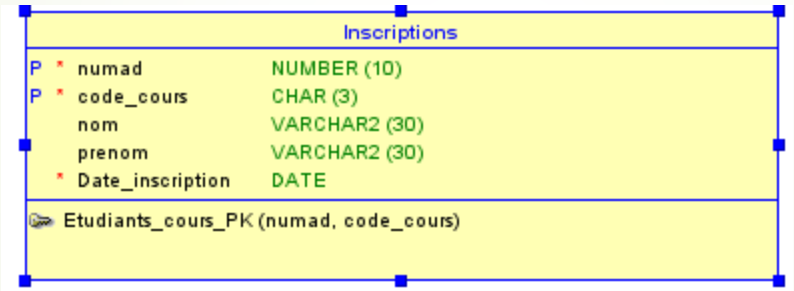
- La deuxième Forme Normale est appelée et notée 2FN
- La 2FN est appelée TOUTE la CLÉ.
- La 2FN garantie l'intégrité des données
- La 2FN concerne les tables avec clé primaire composée.

Énoncé 2: Une table est en 2FN si et seulement si:

1. Elle est en 1FN et
2. Chaque attribut de la table dépend de TOUTE la clé et non pas d'une partie de la clé. Il n'y a pas de dépendance partielle de la clé.

La 2FN(2/4)

- Exemple 1: **Le problème**.
- Le problème. Dans la table Inscriptions suivante, on veut représenter les inscriptions d'un étudiants aux différents cours. La table n'est pas en 2FN. Pourquoi ?



1. L'attribut nom qui représente le nom de l'étudiants dépend uniquement du numad
2. L'attribut prenom dépent uniquement du numad.

Par les points 1 et 2, on constate qu'il existe des attirbuts qui ne dépendent pas de TOUTE la clé mais uniquement d'une partie de la clé.

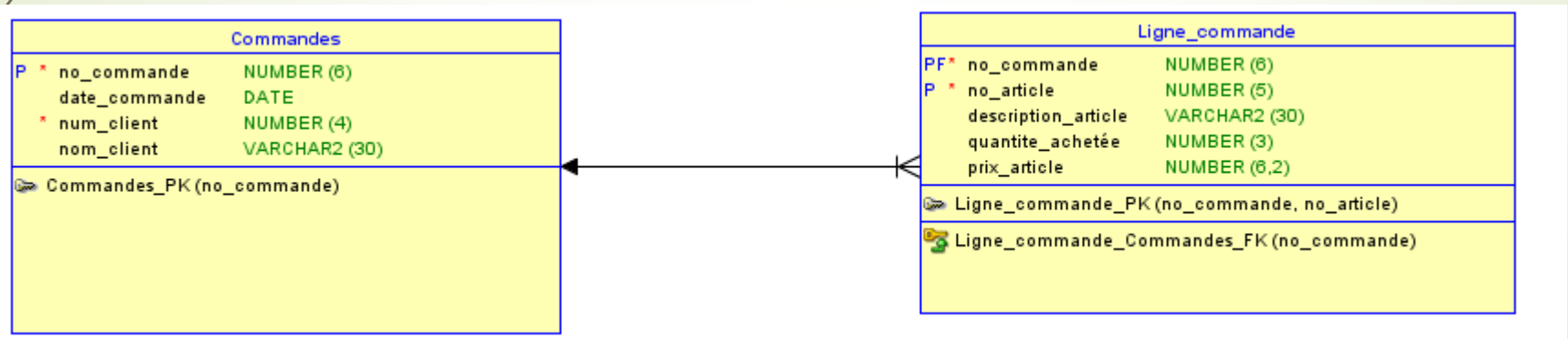
La solution : De la table Inscriptions , on enlève les attributs nom et prenom.

La 2FN(3/4)

- Exemple 2: **Le problème**.

Dans la table Ligne_commade qui représente la liste des produits commandés. Pour chaque commande (no_commande) on retrouve les articles (no_article,description_article) que nous avons commandé.

La Table Ligne_commande n'est pas en 2 FN pourquoi ?

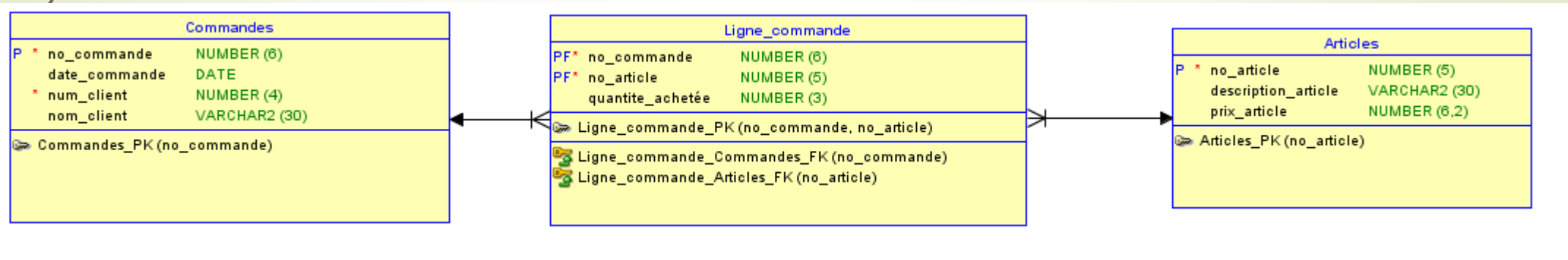


1. L'attribut description_article dépend uniquement de no_article. (ne dépend pas de no_commande)
2. L'attribut prix_article dépend uniquement de no_article (ne dépend pas de no_commande)

La 2FN(4/4)

- Exemple 2: **La solution**.

1. Sortir description_article de la table ligne_commande
2. Sortir le prix_article de la table ligne_commade.
3. Créer une table Articles qui va contenir les données (description_article et prix_article)
4. Vérifier que toutes les tables sont en 2FN.



La 3FN(1/3)

- La troisième Forme Normale est appelée et notée 3FN
- La 3FN est appelée RIEN que la CLÉ.
- La 3FN garantie l'intégrité des données.
- La 3FN concerne les clés étrangères.

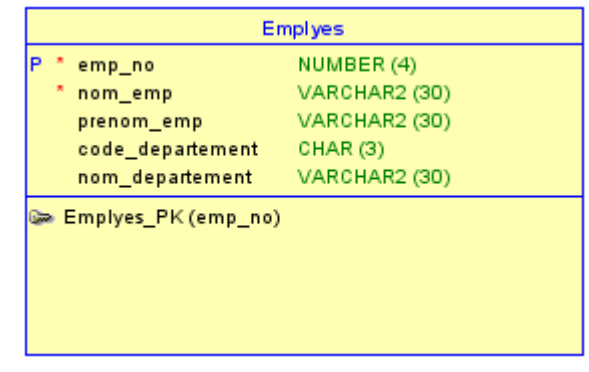
Énoncé 3: Une table est en 3FN si et seulement si:

1. Elle est en 2FN et
2. Chaque attribut de la table NE dépend QUE da la clé primaire ou encore chaque attribut non-clé ne dépend pas d'un autre attribut non-clé

La 3FN(2/3)

- Exemple 1: **Le problème**.

la table Employes suivante n'est pas en 3 FN. Pourquoi ?



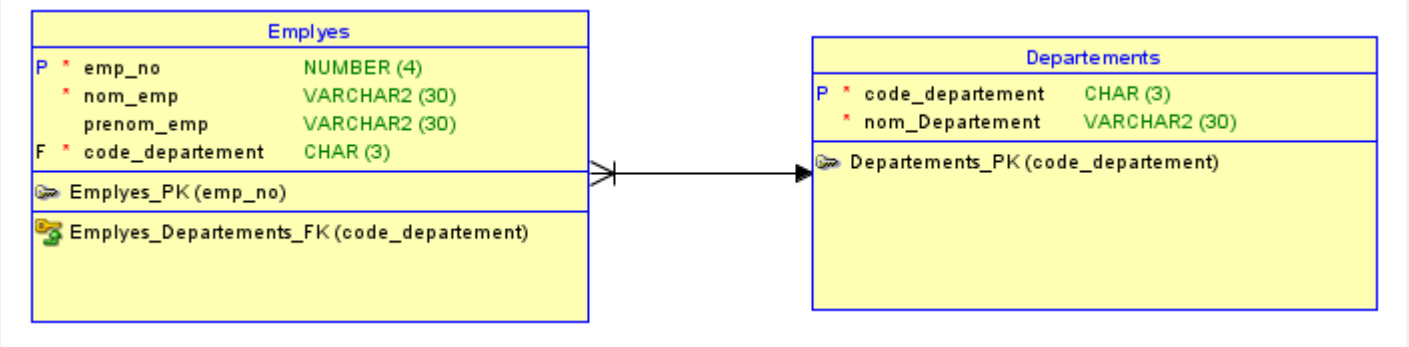
Le nom_département dépend de code_departement. Il y a un attribut non-clé qui dépend d'un autre attribut non-clé.

Solution:

1. Sortir le nom_departement de la table Employes
2. Créer une table département qui va avoir comme clé primaire code_département
3. Faire en sorte que code_departement de la table Employes soit une clé étrangère.

La 3FN(3/3)

- On voit bien que toutes les tables sont en 3FN. Donc la BD est normalisée



Question: La table Commandes de **la page 12**, est-elle en 3FN ?

La réponse est **NON**, puisque le nom_client dépend de num_client.

Il faudra créer une table Clients qui va contenir le num_client et le nom_client. La clé primaire de la table Clients et num_client.

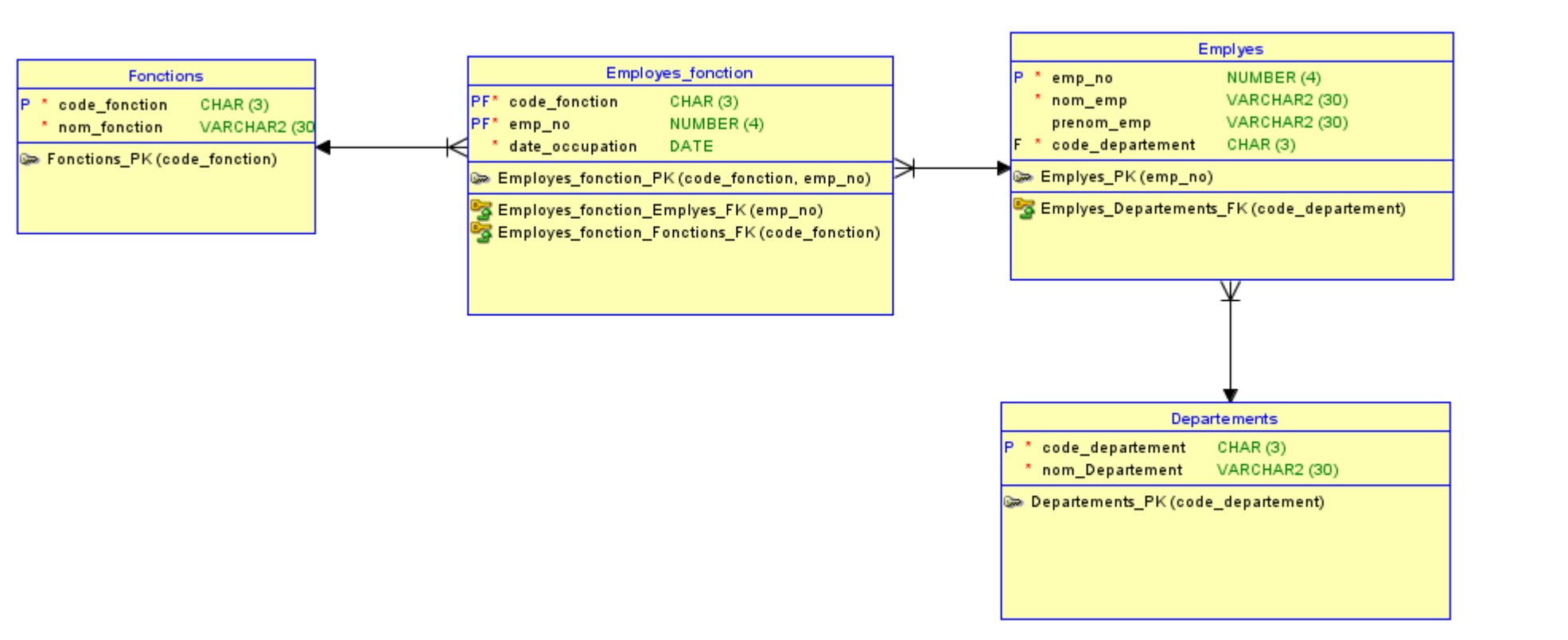
Le num_client de la table Commandes et une clé étrangère (FK)

Application

- On part du modèle précédent (page 14) et on veut représenter les fonctions qu'occupe un employé durant sa carrière.
- Chaque fonction a un code_fonction, un nom_fonction.
- Un employe peut occuper plusieurs fonctions dans sa carrière. Il obtient des promotions par exemple.
- Chaque fonction peut être occupée par plusieurs employés.
- A chaque fois on souhaite garder la date à laquelle il commence la nouvelle fonction.

On vous demande de compléter le modèle pour avoir une BD en 3FN

Solution


Fonctions
P * code_fonction CHAR (3)
* nom_fonction VARCHAR2 (30
Fonctions_PK (code_fonction)
Employes_fonction
PF* code_fonction CHAR (3)
PF* emp_no NUMBER (4)
* date_occupation DATE
Employes_fonction_PK (code_fonction, emp_no)
Employes_fonction_Emplyes_FK (emp_no)
Employes_fonction_Fonctions_FK (code_fonction)
Emplyes
P * emp_no NUMBER (4)
* nom_emp VARCHAR2 (30)
prenom_emp VARCHAR2 (30)
F * code_departement CHAR (3)
Emplyes_PK (emp_no)
Emplyes_Departements_FK (code_departement)
Departements
P * code_departement CHAR (3)
* nom_Departement VARCHAR2 (30)
Departements_PK (code_departement)

Au final

- Quand on lit un énoncé décrivant une bases de données parfois on perd le nord. Ce n'est pas toujours évident de normaliser.
- Le processus de normalisation est un processus pas facile. Ça prend de l'Expérience.
- Le modèle de données obtenu après la normalisation n'est pas unique. (la solution n'est pas unique)
- Alors, comment peut-on arriver à des solutions ?
- Par de la pratique, de la logique .
- Voici quelques étapes qui pourraient vous aider.

Au final

Étape1, Repérez les tables: Quand, vous lisez un énoncé .. Repérer les objets qui ont une exitance propre. Chaque objet devient une table.

Par exemple:

Etudiants, cours, Programmes,Enseignants.

Employes, Departements, formations, conges, Projets.

Étape 2, trouvez la clé primaire pour chaque table

Exemple: pour etudiants →numad, pour Programme →codeProgramme.

Si la clé primaire n'est pas évidente, donner un numéro séquentiel.

Étape 3, reperez les attributs qui dépendent directement de la clé primaire. Cette étapes est un peu intuitive. Comme par exemple: Nom_etudiant, dépend uniquement du numAd. A ce stade pensez uniquement 1FN

Au final

Si vous avez bien fait votre travail, il vous restera:

1. Les attributs qui ne dépendent pas d'une seule clé primaire (comme par exemple une NOTE. Et oui, la note d'un étudiant dépend du NUMAD et du CODE_COURS)
2. Déterminer les liens entre les tables.

Comment faire les liens ?? Par la lecture de l'énoncé encore une fois

Étape4, déterminer les liens:

Lorsque vous lisez votre énoncé concernant les liens possibles, si vous lisez DEUX fois le mot PLUSIEURS… cela veut dire que vous avez une table supplémentaire. Cette table va avoir comme clé primaire, la clé composée des deux autres tables.

La nouvelle table est appelé table de lien et a comme nom table1_table2 (cette appellation n'est pas obligatoire)

Au final

Exemples:1

Un étudiant est inscrit à PLUSIEURS COURS. Dans un COURS il y a PLUSIEURS étudiants

L'énoncé précédent implique qu' il y a une table **Inscriptions.** Cette table a une clé primaire composée qui est (NUMAD , code_cours).

La table inscription est une table de lien. On aurait pu l'appeler: Etidiants_Cours.

Exemple2:

Un employé peut suivre PLUSIEURS Formations. Une formations est suivie par PLUSIEURS employés.

L'énoncé précédent implique qu'il y a une table **Employes_Formations .** Cette tablea une clé primaire composée qui est (emp_no , le code_formation).

Au final

Lorsque vous lisez votre énoncé concernant les liens possibles, si vous lisez une seule fois le mot PLUSIEURS… cela veut dire que vous avez une clé étrangère.
C'est la table dont vous dites UN SEUL qui accueille la clé étrangère de l'autre table

- Exemple1:

Un étudiant est inscrit dans UN SEUL programme. Dans un programme nous avons plusieurs étudiants.

L'énoncé précédent indique que la table ETUDIANTS va accueillir le code_programme pour être une FK

- Exemple2

UN client peut faire plusieurs commandes. Une commande appartient à un SEUL Client.

L'énoncé précédent indique que la table commandes va accueillir le Num_Client pour être une clé Étrangère