TD 1 – Modélisation conceptuelle et logique des données

# Partie 1

## Cas du parc informatique

* Les contraintes
  + Il faut que la date de maintenance soit inférieure à la date d’achat
* Pour que le schéma soi complet, il faut documenter les entités et ajouter le   
  dictionnaire des donnéesUne image contenant capture d’écran

  Description générée automatiquement



* Building (BuildingNo)
* Office(#BuildingNo, OfficeNo)
* PC (SerialNo, purchaseDate, maintenanceDate, model, #buildingNo, #OfficeNo)
* Device(NetworkAddr, purchaseDate, maintenanceDate, type, #builingNo, #officeNo)
* Employé(Name, surname, role, #SerialNo)
* PCDevice(#SerialNO, #NetworkAdress, priority)
* 1FN
  + Aucun attribute multivalué
  + Validé
* 2FN
  + On vérifie que l’on est en 1FN
  + Aucun attribut ne dépend d’une partie de la clé
  + Validé
* 3FN
  + Aucune dépendance fonctionnel entre les attibuts
  + Validée

## Cas du tournois de tennis

* Non, ce n’est pas possible (Participe -> minimum 2, max 4 et contrainte à 3)
* Oui
  + Il n’y a pas de règles sur les joueurs, d’après le modèle
  + Correction : Ajout d’une contrainte
* Oui
  + Aucune contrainte sur les horaires des matchs

## Cas des invitations sociales

* Une personne ne peut être ami et ennemie de la même personne. Il s’agit de la même chose pour Aime et DétesteUne image contenant texte, carte

  Description générée automatiquement

Personne(Nom, Prenom, Sexe, Age, profession)

Reception(Date)

Vin(Région, Type)

Repas(Nom, Nature)

Est\_serv(#nom, #nature, #date)

Ami(#nom1, #prenom1, #nom2, #prenom2)

Ennemi(#nom1, #prenom1, #nom2, #prenom2)

Aime(#nomPersonne, #prenom, #nomPlat, #nature)

Aime(#nomPersonne, #prenom, #nomPlat, #nature)

Va\_avec(#nom, #nature, #type, #region)

# Partie 2

## Les clés

Clé de relation : sous ensemble d’attribut minimal qui permet d’identifier d’une façon unique les tuples d’une entité

Clé atomique : Clés composées d’un seul attribut

1. Oui R inter S est contenu dans R et K clé e R -> K est une clé
2. L’union reprend tout ce qu’il y a dans R et K

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S |  |  | R |  |  |
| K | X | Y | K | A | B |
| 0 | 1 | 2 | 0 | 1 | 2 |
| 1 | 2 | 4 | 1 | 1 | 2 |
| 2 | 3 | 1 | 2 | 2 | 3 |

On aurait des tuples avec des valeurs de K égales, donc K n’est pas la clé de

1. S – R inclus dans R, or comme K est la clé de R, K reste la clé de la différence
2. : Clairement, cela reste la clé de la projections même si on supprimer des attributs

## Dépendance fonctionnelle et normalisation

* Les clés
  + Abonnée
    - Login à partir de la DF1
  + Abonnement
    - Login à partir de la DF4
  + Connexion
    - Login, Date, Heure à partir de la DF5

Abonné (Login, Nom, Prénom, Tél, Ville)

Abonnement (Login, TypeAbonnement, Espace, nbAdresses, EspaceMax, nbAdressesMax,) Connexion (Login, Date, Heure, Durée)

* Abonnée
  + 1FN : Tous les attributs sont atomiques
  + 2FN : Pas de dépendance entre les attributs
  + 3FN : Pas en 3eme forme normale :DF1 : Téléphone -> Login
    - Une dépendance entre les attributs hors clés …
    - Étapes :
      * Isoler la DF qui pose problème
        + Abonné2(Tél, Nom, prénom, adresse)
      * Suppression de la partie de la DF
* Abonnement,
  + 1FN : Tous les attriburs sont atomiques
  + 2FN : Pas de dépendance
  + - 1 isoler la DF qui pose problème
      * TypeAbonnement(TypeAbonnement, EspaceMax, nbAdressesMax)
    - Suppression de la partie de droite de la DF
      * Abonnement2(Login, espace, nbAdresse, #typeAbonnement)
* Connexion :
  + 3FN

Abonné2 (Tel, Nom, Prénom, Ville)

Abonné (Login, #Tél)

Abonnement (Login, TypeAbonnement, Espace, nbAdresses, EspaceMax, nbAdressesMax,) Connexion (Login, Date, Heure, Durée)