

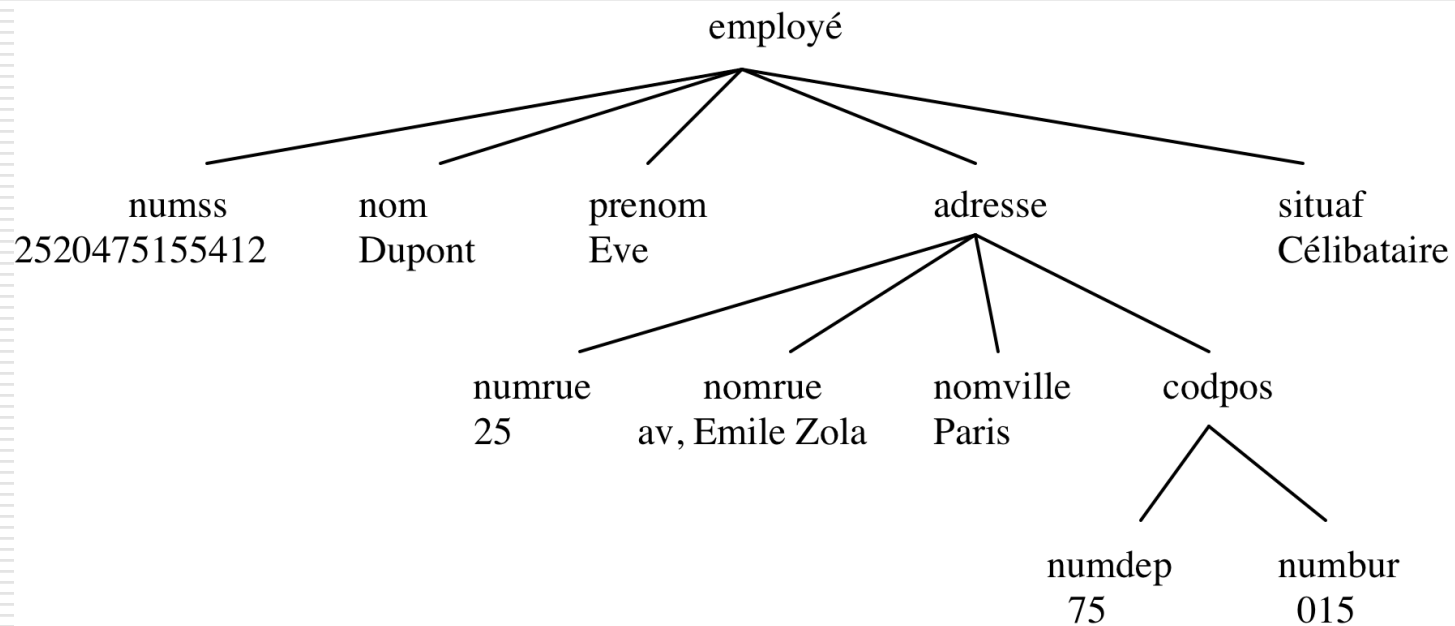
Chapitre 7 – Les enregistrements

C'est une structure de données hiérarchisée.

C'est une cellule composée d'éléments qui peuvent être de types différents.

Exemple

- ❑ Employé : Numss, Nom, Prénom, Adresse, Situaif.
- ❑ Adresse : Numrue, Nomrue, Nomville, Codpos.
- ❑ Codpos : Numdep, Numbur.



La variable « employé »

- ❑ La variable enregistrement « employé » est représentée par une arborescence de racine « employé ».
- ❑ A chaque niveau de l'arborescence, on trouve les champs.
- ❑ Un champ correspond à un sommet distinct de la racine ; on l'identifie à l'aide des sommets du chemin unique qui lie la racine au champ.
- ❑ Les champs feuilles de l'arborescence sont dits champs élémentaires.
- ❑ Un employé particulier est une valeur de la variable employé.
- ❑ Chacune des informations placées aux feuilles est la valeur d'un champ élémentaire.
- ❑ L'ensemble de toutes ces valeurs constitue la valeur de la variable employé.

Déclaration du type et de la variable

- Un type enregistrement est décrit par les types des champs structurés hiérarchiquement.

```
Type code-postal = Enregistrement
    numdep, numbur : chaine
    FinEnregistrement
adresse-pos = Enregistrement
    numrue, nomrue, nomville : chaine
    codpos : code-postal
    FinEnregistrement
employé = Enregistrement
    numss, nom, prénom, situaif : chaine
    adresse : adresse-pos
    FinEnregistrement
Var e : employe
```

Utilisation

- ❑ Numéro du département où habite l'employé e :
`e.adresse.codpos.numdep`
- ❑ Les seules opérations globales possibles sur les enregistrements de mêmes types sont l'affectation et la comparaison ($=$ ou \neq).
- ❑ Les opérations possibles sur les champs sont celles réalisables sur les variables du type de ces champs.

Exemple : tableau de points

Const max = 30

Type coord = Enregistrement

 x,y : entier

FinEnregistrement

 tab-coord = tableau[1..max]
 de coord

Var t : tab-coord

 i : entier

 p : coord

Début

 p.x ← 0

 p.y ← 0

Pour i ← 1 à max Incr +1 Faire

 lire(t[i].x,t[i].y)

 p.x ← p.x + t[i].x

 p.y ← p.y + t[i].y

FinPour

 p.x ← p.x div max

 p.y ← p.y div max

Fin