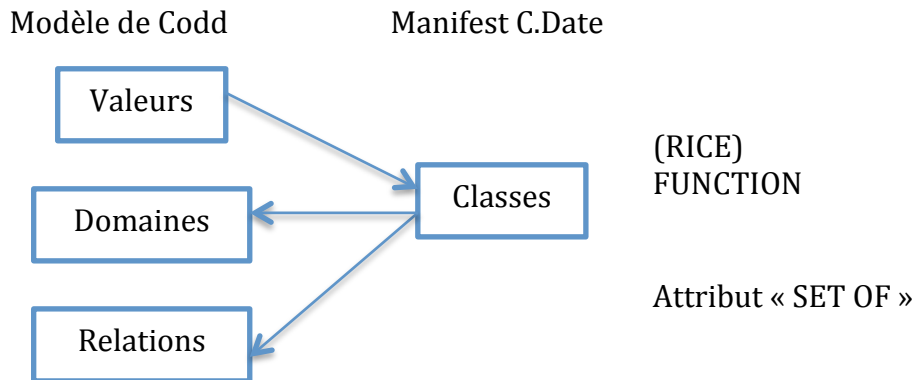


Base de donnée

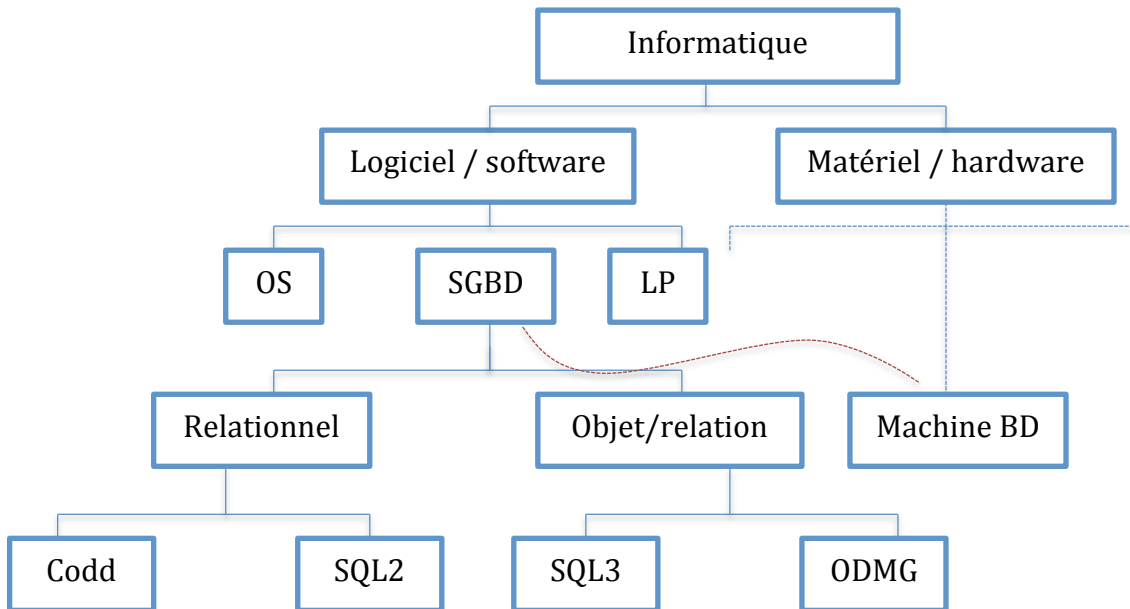
Cours 3



La dernière fois nous avons vu le manifest de Date appliqué sur du 2D
 Aujourd'hui nous allons voir avec un système documentaire

Pour un système de documentaire, nous avons besoin de la notion de thésaurus
 Thésaurus : Super index utilisé lors de l'indexation et utilisable dans la manipulation. Il permet d'étendre ou de réduire un champ de recherche. Le cœur sémantique d'un thésaurus est un arbre de mot-clé.

Au niveau informatique :



/ pour synonyme

----- pour lien de voisinage

Un thésaurus c'est un domaine de valeur organisé autour d'un arbre (hiérarchie) -> réseau sémantique

LIVRE(L#, titre, editeur, npages, {auteurs}, {motclés})

On veut que le thésaurus étende sémantiquement la recherche.

Réduire permet de descendre dans la hiérarchie et réduire le bruit (nombre de résultat), à l'inverse de l'extension.

Lorsqu'on utilise les liens de voisinage, (SGBD – machine BD) on fait une recherche étendue.

1. Comment représenté LIVRE et THESAURUS en SL2.
2. Convertir en modèle OR de Date : « SET OF » → attribut multilié possible

Repondre à la question pour ces 2 modèles :

Q : Quel sont les livres dont le titre commence par « concept » et qui concerne les SGBD ?

1.

LIVRE(L#, titre, editeur, npages, {auteurs}, {motclés})

LIVRE1(L#, titre, editeur, nbpage)

LAUTEUR(A#, L#)

LMOTCLES(MC#, L#)

HIERARCHIE(MCP, MCF)

SYNONYME(MC1, MC2)

VOISINAGE(MC1, MC2)

Ou on indique tout les couples possibles.

THESAURUS	MC1	MC2	Type Relation
	Informatique	Logiciel	H
	Informatique	CS	S

Requette SQL de base :

```
SELECT * FROM LIVRE1, LAUTEUR, LMOTCLES
```

```
WHERE LIVRE1.L# = LAUTEUR.L#
```

```
AND LIVRE1.titre LIKE 'Concept%'
```

```
AND (MC = 'SGBD'
```

```
OR MC IN (SELECT MCF FROM HIERARCHIE WHERE MCP = 'SGBD')
```

```
OR MC IN (SELECT MC2 FROM SYNONYME WHERE MC1 = 'SGBD')
```

```
OR MC IN (SELECT MC2 FROM VOISINAGE WHERE MC1 = 'SGBD') )
```

2.

LIVRE(L#, titre, editeur, npages, setof(auteur), setof(MC))

1° on crée le domaine (classes pour Date)

```
Create Domain THESAURUS
```

```
Create function HIERARCHIE(MC)
```

```
Return setof(MC)
```

```
Create function SYNONYME(MC)
```

```
Return setof(MC)
```

```
Create function VOISINAGE(MC)
```

```
Return setof(MC)
```

2° on crée les tables

Create table LIVRE

(L# definie sur LIVNO

⋮

setof(MC) definie sur THESAURUS)

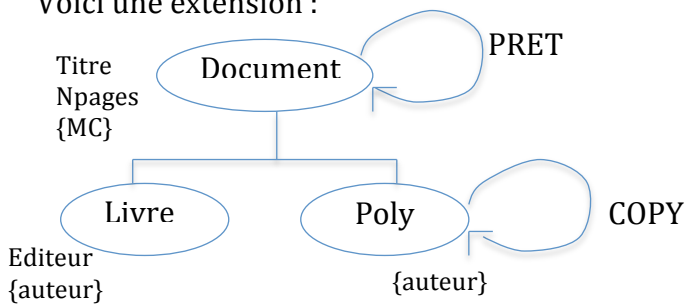
Q :

SELECT * FROM LIVRE

WHERE titre = 'concet%'

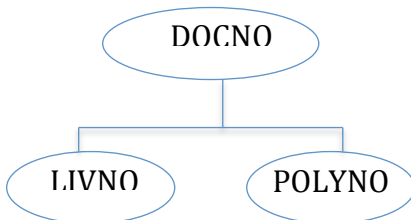
AND MC IN (SGBD, HIERARCHIE(SGBD), VOISINAGE(SGBD), SYNONYME(SGBD))

Voici une extension :



Quels sont les nombres de pages des livres publiés chez dunod qui ont été prêtés ?
On peut soit définir le domaine en compréhension (on donne toutes les valeurs) ou en intention comme suit

On a une hiérarchie de domaine primaire :



Create domain DOCNO primary INT

Create function PRET(id)

Return true/false

Create domain LIVNO primary under DOCNO

Create domain POLYNO primary under DOCNO

Create function COPY(id)

Return true/false

Create domain DTITRE VARCHAR(15)

Create domain DPAGE INT

Create domain DEDITEUR VARCHAR(12)

```

Create domain DAUTEUR VARCHAR(12)
Create Domain THESAURUS
  Create function HIERARCHIE(MC)
    Return setof(MC)
  Create function SYNONYME(MC)
    Return setof(MC)
  Create function VOISINAGE(MC)
    Return setof(MC)

```

```

Create table DOCUMENT
  (D#          definie sur  DOCNO
  NBPAGES     definie sur  DPAGE
  TITRE       definie sur  DTITRE
  Setof(MC)   definie sur  THESAURUS)

```

```

Create table LIVRE
  (L#          definie sur  LIVNO
  EDITEUR     definie sur  DEDITEUR
  Setof(Auteur)definie sur DAUTEUR)

```

```

Create table POLY
  (P#          definie sur  LPOLYNO
  Setof(Auteur)definie sur DAUTEUR)

```

```

Q :
SELECT L# → NBPAGES
FROM LIVRE
WHERE EDITEUR = 'Dunod' AND PRET = 'true'

```

a→* signifie que pour a une clé primaire, on va chercher n'importe quel attribut sur l'héritage