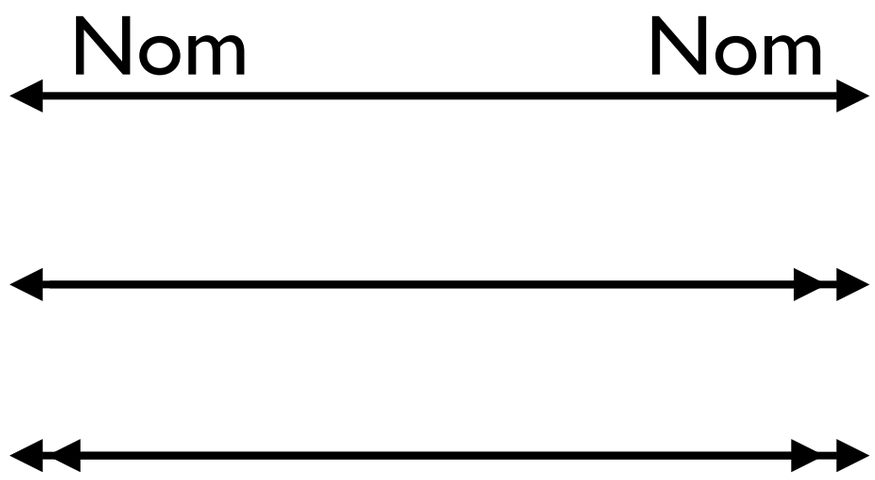
INFO-H-511

Bases de données objet
Exercices

ODMG Schemas : Notations

- Classes et héritage :
  - comme en UML
- Relations :
  - 1-1 :
  - 1-n :
  - m-n :



OQL : Définition de classe

- Les relations sont bi-directionnelles
- Les 1-n / m-n utilisent des collections :
  - set (pas de doublons), bag
  - list, array

```
class Person
(extent Persons){
  attribute string name;
  attribute struct Address{
    unsigned short number,
    string street
  } address;
  relationship Person spouse inverse Person::spouse;
  relationship set<Person> children inverse Person::parents;
  relationship list<Person> parents inverse Person::children;
}
```

Requêtes OQL

- ```
  select p
  from p in Persons
  where p.name=”Pat”
  ```
  - retourne une collection d’objets
- ```
  select p.name
  from p in Persons
  ```
  - retourne une collection de littéraux

Requêtes OQL

- ```
  select struct( name    : x.name,
                 address : x.address )
  from x in Persons
  ```
  - retourne une collection de `struct`
  ```

Requêtes OQL

- ```
  select struct( parent : p.name,
                 child  : c.name )
  from p in Persons,
       c in p.children
  ```
  - jointure
  - ((p1, c1),(p1,c2),(p2,c1),(p2,c2),(p3,c3),
    (p4,c3)...)
  ```

Navigation

- Pour les attributs ou relations 1-1 :
  - `s.spouse.address.city.name`
  - le dernier élement peut être multivalué
- Pour les relations m-n
  - utiliser des clauses “select from” (jointures)

Fonctions

- `distinct(1,2,2,2)` retourne `(1,2)`
- `count(1,2,3)` retourne 3
- `not(true)` rentourne `false`
- `max(), sum(),…`
- `flatten((1,2,3),(1,2)) = (1,2,3,1,2)`
- …

Partitionnement

- ```
  select *
  from Employees e
  group by low :      salary <  1000,
            medium :  salary >= 1000 and
                      salary <  10000,
           high :     salary >= 10000
  ```

- Renvoie des données selon cette structure

  - ```
    set<struct(
       low:boolean,
       medium:boolean,
       high:boolean,
       partition:bag<struct(e.Employee)>
    )>
    ```