

Expressions de requêtes en calcul predicatif

Question 1

select prenomEnf, ageEnf from LesEnfants;

$$\{ a \mid \exists b \in \text{LesEnfants}, a.\text{prenomEnf} = b.\text{prenomEnf} \wedge a.\text{ageEnf} = b.\text{ageEnf} \}$$

LesActivites[prenomEnf,nomAct] / LesActivites[nomAct]

$$\{ a \mid \exists b \in \text{LesActivites}, \forall c \in \text{LesActivites}, a.\text{prenomEnf} = b.\text{prenomEnf} \wedge b.\text{nomAct} = c.\text{nomAct} \}$$

select prenomEnf from LesActivites group by prenomEnf having count(distinct nomAct) =2

$$R1 = \{ a \mid \exists b \in \text{LesActivites}, \exists c \in \text{LesActivites}, \\ a.\text{prenomEnf} = b.\text{prenomEnf} \\ \wedge b.\text{prenomEnf} = c.\text{prenomEnf} \\ \wedge b.\text{nomAct} \neq c.\text{nomAct} \}$$
$$R2 = \{ a \mid \exists b \in \text{LesActivites}, \exists c \in \text{LesActivites}, \exists d \in \text{LesActivites}, \\ a.\text{prenomEnf} = d.\text{prenomEnf} \\ \wedge b.\text{prenomEnf} = c.\text{prenomEnf} \\ \wedge b.\text{prenomEnf} = d.\text{prenomEnf} \\ \wedge b.\text{nomAct} \neq c.\text{nomAct} \\ \wedge b.\text{nomAct} \neq d.\text{nomAct} \\ \wedge c.\text{nomAct} \neq d.\text{nomAct} \}$$
$$R = R1 - R2$$

Programmes Datalog

Question 2

1/

```
/assert lesEnfants (alice, alison, parent, 5).  
/assert lesResponsables (alison, grenoble, 0766554411).
```

2/

```
/assert lesEnfants (margaux, thomas, tuteur, 10).
/assert lesEnfants (matilde, thomas, parent, 12).
/assert lesResponsables (thomas, vizille, 0445541444).
```

3/

```
/assert lesActivites (pingpong, 23122015, alex, alice).
/assert lesActivites (pingpong, 23122015, alex, margaux).
/assert lesActivites (macrame, 22122015, thomas, margaux).
/assert lesEmployes (thomas, moniteur).
```

4/

```
/assert lesEmployes (pierre, directeur).
/assert lesEmployes (josiane, secretaire).
```

Question 3

On crée une assert qui « cherche » ceux qui ne respecteront pas la contrainte :

```
/assert peutIlEncadrer(A) :- lesActivites(_,_,A,_), lesEmployes (A, B),
not(B=moniteur).
```

...Ainsi, si la requête retourne aucun résultat, alors la base de donnée respecte la contrainte.

Question 4

```
/assert cercle(E1,E2) :- lesActivites (A, B, _, E1), lesActivites (A, B, _, E2), not(E1=E2).
```

Question 5

```
/assert cercleElargi(E1, E2) :- cercle(E1, E2), not(E1=E2).
/assert cercleElargi(E1, E2) :- lesEnfants(E1, R, _, _), lesEnfants (E2, R, _, _), not(E1=E2).
/assert cercleElargi(E1, E2) :- cercleElargi(E1, E3), cercleElargi (E2, E3), not(E1=E2),
not(E3=E2), not(E3=E1).
```

Question 6

```
/assert PasDansMemeCercle(E1,E2) :-  
lesEnfants(E1,-,-,-),lesEnfants(E2,-,-,-),not(E1=E2),  
not(cercle(E1,E2)).
```