

CLIENT (**nclt**, nom, age, adresse)
 PROD (**nprod**, design, couleur, volume)
 CMD (**nclt**, **nprod**, quantité, date)

$$A \Rightarrow B \leftrightarrow \neg A \vee B$$

$$\neg A \leftrightarrow \neg A$$

$$A \wedge B \leftrightarrow \neg(A \Rightarrow \neg B) \\ \neg(\neg A \vee \neg B)$$

$$A \vee B \leftrightarrow \neg A \Rightarrow B$$

$$A \Rightarrow B \leftrightarrow A \Rightarrow B$$

$$A \Leftrightarrow B \leftrightarrow (A \Rightarrow B) \wedge (B \Rightarrow A) \\ \neg((A \Rightarrow B) \Rightarrow \neg(B \Rightarrow A))$$

A	B	$A \wedge B$	$A \Rightarrow B$
1	1	1	1
1	0	0	0
0	1	0	1
0	0	0	1

♣ CRT (Calculs Relationnels sur Tuples)

♥ CRD (Calculs Relationnels sur Domaines)

1) Liste des noms de clients qui ont un age > 20

♣ { t[nom] | t ∈ CLIENT ∧ t[age] > 20 }

♥ { <n> | ∃ clt, a (< clt, n, a, _ > ∈ CLIENT ∧ a > 20) }

```
select nom
from client
where age > 20;
```

2) Liste des noms de clients ayant commandé le produit n°13

♣ { t[nom] | t ∈ CLIENT ∧ ∃ c (c ∈ CMD ∧ c[nprod] = 13 ∧ t[nelt] = c[nelt]) }

♥ { <n> | ∃ clt (< clt, n, _, _ > ∈ CLIENT ∧ < clt, 13, _, _ > ∈ CMD) }

```
select nom
from client natural join cmd
where nprod = 13;
```

```
select nom
from client join cmd on client.nclt = cmd.nclt
where nprod = 13;
```

```
select nom
from client, cmd
where client.nclt = cmd.nclt and nprod = 13;
```

```
select nom
from client
where nclt in (select nclt
               from cmd
               where nprod = 13);
```

3) Liste des noms de clients ayant commandé un produit de couleur rouge

♣ { t[nom] | t ∈ CLIENT ∧ ∃ c (c ∈ CMD ∧ c[nclt] = t[nclt] ∧ ∃ p (p ∈ PROD ∧ p[nprod] = c[nprod] ∧ p[couleur] = "rouge")) }

♥ { <n> | ∃ clt (< clt, n, _, _ > ∈ CLIENT ∧ ∃ prod (< clt, prod, _, _ > ∈ CMD ∧ < prod, _, "rouge", _ > ∈ PROD)) }

```
select nom
from client natural join cmd natural join prod
where couleur = 'rouge';
```

```
select distinct nom
from client natural join cmd natural join prod
where couleur = 'rouge';
```

```
select nom
from client
where nclt in (select nclt
               from cmd natural join prod
               where couleur = 'rouge');
```

4) La couleur des produits commandés par Monsieur Dupont

♣ { p[couleur] | p ∈ PROD ∧ ∃ t (t[nom] = "Dupont" ∧
∃ c (c ∈ CMD ∧ c[nprod] = p[nprod] ∧ p[nclt] = t[nclt])) }

♥ { <c> | ∃ clt (< clt, "Dupont", _, _ > ∈ CLIENT ∧
∃ prod (< clt, prod, _, _ > ∈ CMD ∧ < prod, _, c, _ > ∈ PROD)) }

```
select distinct couleur
from client, prod, cmd
where client.nclt = cmd.nclt and
      cmd.nprod = prod.nprod and
      lower(client.nom) = 'dupont';
```

```
select distinct couleur
from client natural join cmd
      natural join prod
where lower(client.nom) = 'dupont';
```

```
create view tuple_dupont
as (select *
    from client
    where lower(client.nom) = 'dupont');
create view cmd_dupont
as (select *
    from cmd
    where nclt in (select nclt
                  from tuple_dupont));
select distinct couleur
from prod
where nprod in (select nprod
                from cmd_dupont);
```

5) Liste des noms de clients ayant commandé au moins un produit

♣ { t[nom] | t ∈ CLIENT ∧ ∃ c (c ∈ CMD ∧ c[nclt] = t[nclt]) }

♥ { <n> | ∃ clt (< clt, n, _, _ > ∈ CLIENT ∧ < clt, _, _, _ > ∈ CMD) }

```
select nom
from client natural join cmd;
```

```
select nom
from client
where nclt in (select nclt
               from cmd);
```

```
select nom
from (select distinct nclt, nom
      from client natural join cmd);
```

6) Liste des noms de clients ayant commandé un produit vert ou rouge

♣ { t[nom] | t ∈ CLIENT ∧ ∃ p (p ∈ PROD ∧
 (p[couleur] = "rouge" ∨ p[couleur] = "vert") ∧
 ∃ c (c ∈ CMD ∧ c[nclt] = t[nclt] ∧ c[nprod] = p[nprod])) }

♥ { <n> | ∃ clt (< clt, n, _, _ > ∈ CLIENT ∧ ∃ prod (< clt, prod, _, _ > ∈ CMD
 ∧ ∃ c (< prod, _, c, _ > ∈ PROD ∧ (c = "rouge" ∨ c = "vert"))) }

```
select nom
from client natural join cmd natural join prod
where couleur in ('rouge', 'vert');
```

7) Liste des noms de clients ayant commandé un produit vert ou bien rouge

♣ { t[nom] | t ∈ CLIENT ∧ ∃ p (p ∈ PROD ∧
 [p[couleur] = "rouge" ∧ ∃ c (c ∈ CMD ∧ c[nclt] = t[nclt] ∧
 c[nprod] = p[nprod] ∧ ¬(∃ p2 (p2 ∈ PROD ∧ p2[couleur] = "vert" ∧
 ∃ c2 (c2 ∈ CMD ∧ c2[nclt] = t[nclt] ∧ c2[nprod] = p2[nprod])))] ∨
 [p[couleur] = "vert" ∧ ∃ c (c ∈ CMD ∧ c[nclt] = t[nclt] ∧
 c[nprod] = p[nprod] ∧ ¬(∃ p2 (p2 ∈ PROD ∧ p2[couleur] = "rouge" ∧
 ∃ c2 (c2 ∈ CMD ∧ c2[nclt] = t[nclt] ∧ c2[nprod] = p2[nprod])))]) }

♣ { t[nom] | t ∈ CLIENT ∧ ∃ c (c ∈ CMD ∧ c[nclt] = t[nclt] ∧
 ∃ p1 (p1 ∈ PROD ∧ p1[nprod] = c[nprod] ∧ ∀ c2 ∀ p2 (c2 ∈ CMD ∧
 p2 ∈ PROD ∧ c2[nclt] = t[nclt] ∧ c2[nprod] = p2[nprod]) ⇒
 [(p1[couleur] = "rouge" ⇒ p2[couleur] <> "vert") ∧
 (p1[couleur] = "vert" ⇒ p2[couleur] <> "rouge")]) }

♥ { <n> | ∃ clt (< clt, n, _, _ > ∈ CLIENT ∧
 ∃ prod1 (< clt, prod1, _, _ > ∈ CMD ∧ ∃ c1 (< prod1, _, c1, _ > ∈ PROD ∧
 ∀ prod2 (∀ c2 (< clt, prod2, _, _ > ∈ CMD ∧ < prod2, _, c2, _ > ∈ PROD ⇒
 [[c = "rouge" ⇒ c2 <> "vert"] ∧ [c = "vert" ⇒ c2 <> "rouge"]]))))) }

```
select nom
from client
where nclt in (((select nclt
                 from client natural join cmd natural join prod
                 where couleur = 'rouge')
              except
              (select nclt
               from client natural join cmd natural join prod
               where couleur = 'vert'))
             union
             ((select nclt
              from client natural join cmd natural join prod
              where couleur = 'vert')
            except
            (select nclt
             from client natural join cmd natural join prod
             where couleur = 'rouge'))));
```

```
select nom
from (select nclt, nom
      from client natural join cmd natural join prod
      where couleur in ('rouge', 'vert')
      group by nclt, nom
      having count(distinct couleur) = 1);
```

8) Liste des noms de clients ayant commandé au moins deux produits

♣ { t[nom] | t ∈ CLIENT ∧ ∃ c1 (c1 ∈ CMD ∧ ∃ c2 (c2 ∈ CMD ∧
 c1[nclt] = t[nclt] ∧ c2[nclt] = t[nclt] ∧ c1[nprod] <> c2[nprod])) }

♥ { <n> | ∃ clt (< clt, n, _, _ > ∈ CLIENT ∧
 ∃ prod1 (< clt, prod1, _, _ > ∈ CMD ∧
 ∃ prod2 (< clt, prod2, _, _ > ∈ CMD ∧ prod1 <> prod2)) }

```
select nom
from client
where (select count(*)
       from cmd
       where cmd.nclt = client.nclt) >= 2;
```

```

select nom
from (select nclt, nom
      from client natural join cmd
      group by nclt, nom
      having count(distinct (nprod)) >= 2;

```

9) Liste des noms de clients dont l'age est > 50 et qui n'ont pas commandé un produit vert

♣ { t[nom] | t ∈ CLIENT ∧ t[age] > 50 ∧ ∀ c (c ∈ CMD ∧ c[nclt] = t[nclt]) ⇒ ∃ p (p ∈ PROD ∧ p[nprod] = c[nprod] ∧ p[couleur] ≠ "vert") }

♥ { <n> | ∃ clt, a (< clt, n, a, _ > ∈ CLIENT ∧ a > 50 ∧ ∀ prod (< clt, prod, _, _ > ∈ CMD ⇒ ¬< prod, _, "vert", _ > ∈ PROD)) }

```

select nom
from prod join cmd on
couleur = 'vert' and prod.nprod = cmd.nprod
natural right outer join client
where couleur is NULL and age > 50;

```

```

select nom
from client
where age > 50 and
not exists (select *
            from cmd natural join prod
            where couleur = 'vert' and cmd.nclt = client.nclt);

```

10) Liste des noms de clients qui ont commandé tous nos produits

♣ { t[nom] | t ∈ CLIENT ∧ ∀ p (p ∈ PROD ⇒ ∃ c (c ∈ CMD ∧ c[nclt] = t[nclt] ∧ p[nprod] = c[nprod])) }

♣ { t[nom] | t ∈ CLIENT ∧ ¬(∃ p (p ∈ PROD ∧ ¬(∃ c (c ∈ CMD ∧ c[nclt] = t[nclt] ∧ p[nprod] = c[nprod])))) }

♥ { <n> | ∃ clt (< clt, n, _, _ > ∈ CLIENT ∧ ∀ prod (< prod, _, _, _ > ∈ PROD ⇒ < clt, prod, _, _ > ∈ CMD)) }

