

## Devoir Sur Table « La chocolaterie du 5<sup>ème</sup> »

Vous êtes employés d'une chocolaterie localisée à Paris au 5<sup>ème</sup>. Vous avez à votre disposition une base de données contenant les produits et les commandes. Chaque produit a un *type* (*blanc, noir, au lait, diet ou composition*) et peut être proposé pour une *occasion* précise (*noël, pâques, saint valentin, etc.*). Les compositions sont des produits mélangeant d'autres produits du catalogue.

**Produit** (codeProd, nomProd, prixKg, type, occasion)

**Composition** (codeCompo, codeProduit) \*

**Clients** (ncli, nomC, tel, dateEntrée, codePostal)

**Commande** (ncom, dateCom, ncli, prixTotal, dateLiv)

**ContenuCom** (ncom, codeProd, qte)

\* L'attribut codeCompo correspond à un codeProd de type « composition »

## Questions algèbre et SQL

Répondre aux questions suivantes utilisant l'algèbre relationnelle et le langage SQL :

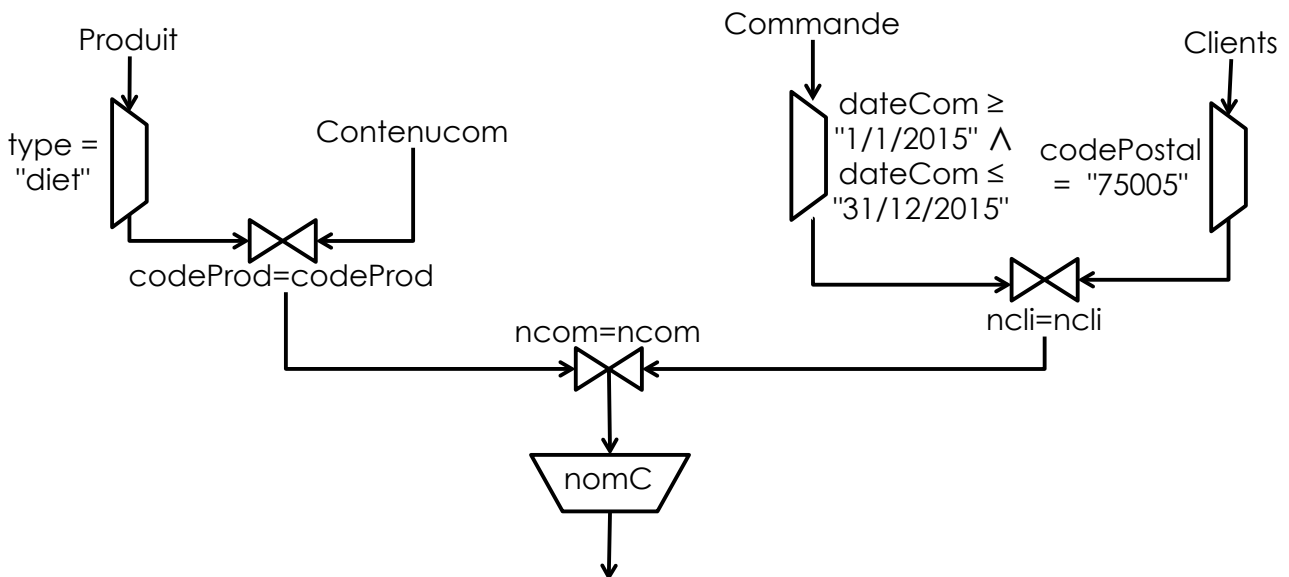
- 1) Les noms des clients du 5<sup>ème</sup> (code postal « 75005 ») qui ont commandé des chocolats « diet » cette année (2015).
- 2) Les clients (nom et téléphone) qui ont commandé des produits de Noël en 2014 mais pas en 2015.
- 3) Les clients (nom) qui ont déjà commandé des produits de pâques et des produits de Noël.
- 4) Les produits (nom produit) mélangeant chocolat noir et chocolat blanc.
- 5) Les clients (nom et code postal) qui ont déjà commandé tous les produits de Noël diet proposés.
- 6) Les clients (nom et téléphone) ayant effectué la commande la plus élevée cette année.

## Questions SQL uniquement

Répondre aux questions suivantes uniquement en langage SQL.

- 7) Le prix moyen des produits par type en ordre décroissant.
- 8) Les noms des clients en ordre alphabétique ayant commandé plus de 5 produits diet en 2015.
- 9) Le prix moyen des compositions contenant plus de 3 produits.

(1) Les noms des clients du 5<sup>ème</sup> (code postal « 75005 ») qui ont commandé des chocolats « diet » cette année (2015). (1,0)



dateCom est une date,  
donc jour/mois/année

(1) Les noms des clients du 5<sup>ème</sup> (code postal « 75005 ») qui ont commandé des chocolats « diet » cette année (2015). (1,0)

```

SELECT  C.nomC
FROM    Clients as C, Produit as P, Commande as Co, ContenuCom as Cc
WHERE   C.ncli = Co.ncli
        AND Co.ncom = Cc.ncom
        AND Cc.codeProd = P.codeProd
        AND P.type = "diet"
        AND C.codePostal = "75005"
        AND Co.dateCom BETWEEN "1/1/2015" AND "31/12/2015"
    
```

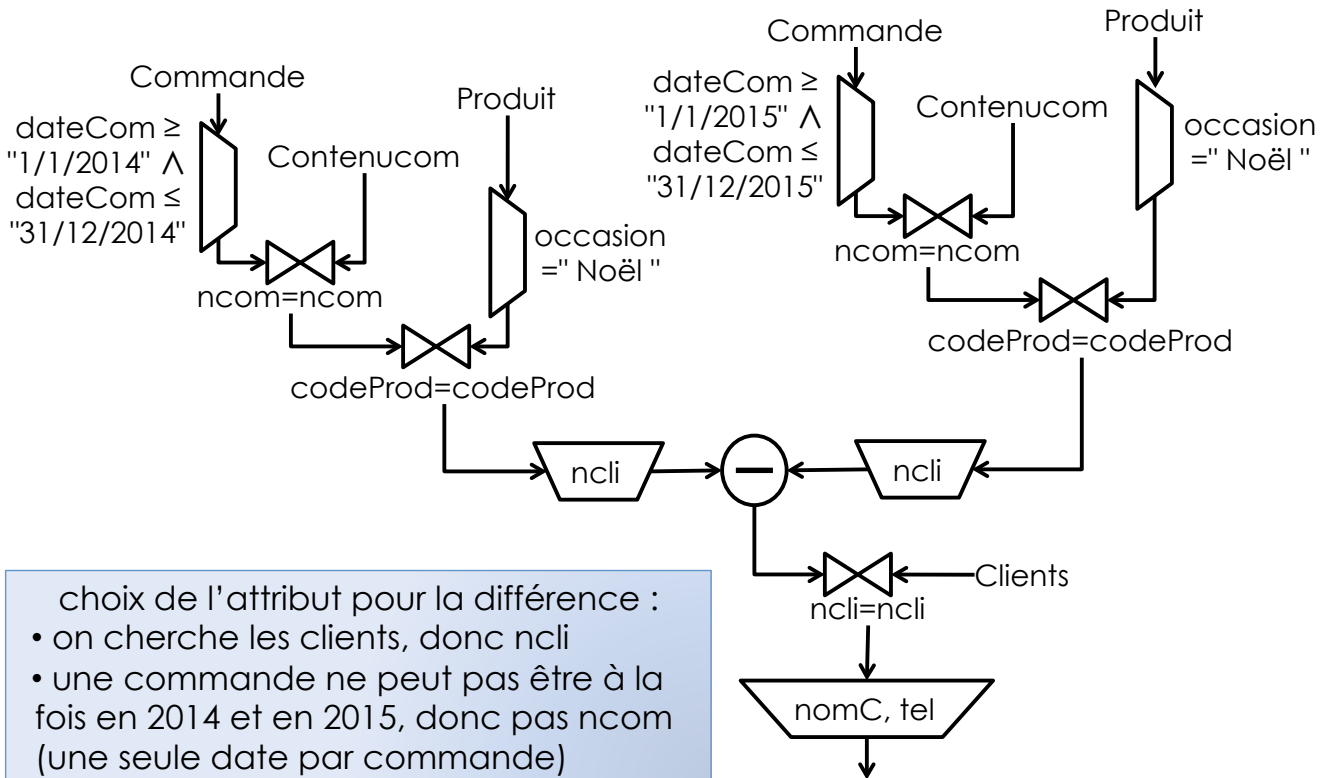
les « alias » (C, p, Co, Cc)  
aident à indiquer de  
quel table on parle

on fait attention aux nom des  
attributs et des tables :

Client**S** C  
P.**codeProd**

dateCom est une date,  
donc jour/mois/année

**(2) Les clients (nom et téléphone) qui ont commandé des produits de Noël en 2014 mais pas (de produit de Noël) en 2015. (1,5)**



**(2) Les clients (nom et téléphone) qui ont commandé des produits de Noël en 2014 mais pas (de produit de Noël) en 2015. (1,5)**

```

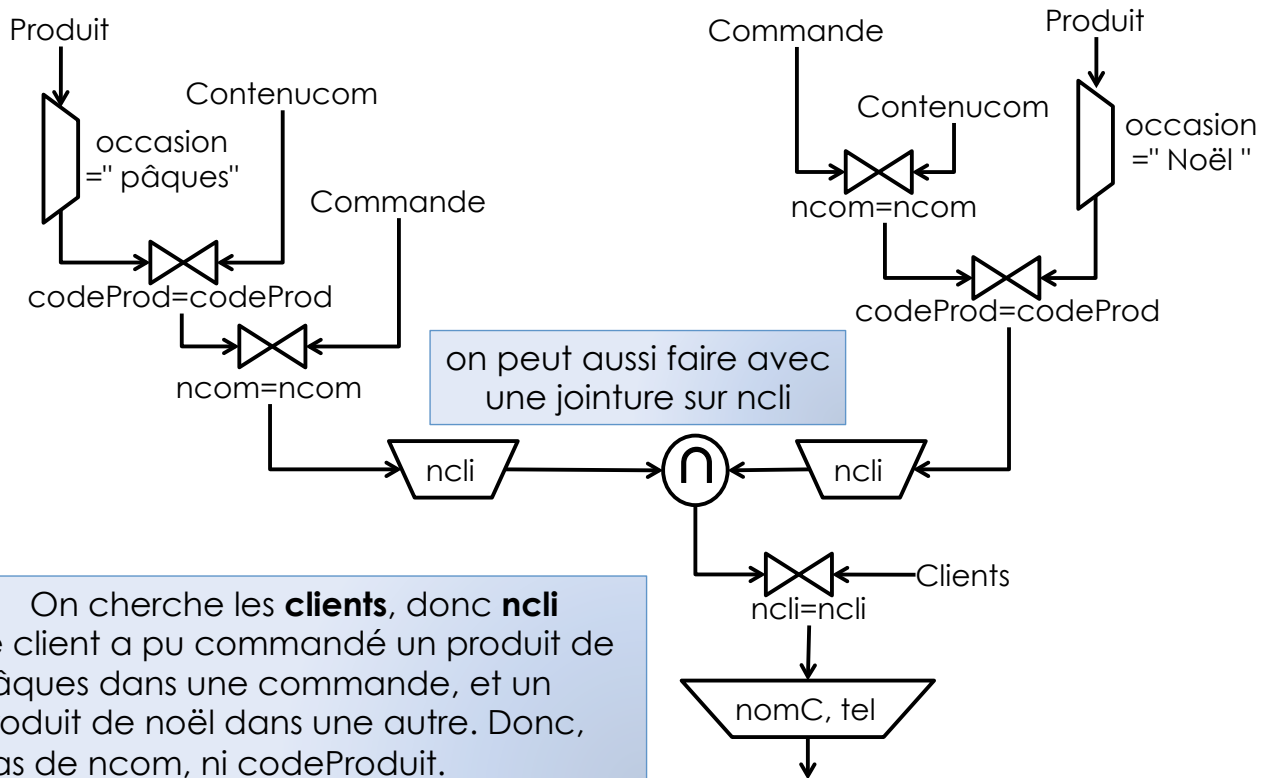
SELECT C.nomC, C.tel
FROM Clients as C, Produit as P, Commande as Co, ContenuCom as Cc
WHERE C.ncli = Co.ncli
      AND Co.ncom = Cc.ncom
      AND Cc.codeProd = P.codeProd
      AND P.occasion = "Noël"
      AND Co.dateCom BETWEEN "1/1/2014" AND "31/12/2014"
      AND Co.ncli NOT IN ( SELECT Co2.ncli
FROM Commande as Co2, Produit as P2,
  ContenuCom as Cc2
WHERE Co2.ncom = Cc2.ncom
      AND Cc2.codeProd = P2.codeProd
      AND P2.occasion = "Noël"
      AND Co2.dateCom BETWEEN "1/1/2015"
      AND "31/12/2015" )
  
```

c'est le **client** qui n'a pas commandé en 2015 **même attribut** de la différence dans le **NOT IN**

Les noms des tables ne changent pas, juste leur **alias**  
**Commande Co**  
**Commande Co2**

On choisit l'alias qu'on veut :  
**Clients Toto** est aussi bon que **Clients C**

### (3) Les clients (nom) qui ont déjà commandé des produits de pâques et des produits de Noël. (1,0)



### (3) Les clients (nom) qui ont déjà commandé des produits de pâques et des produits de Noël. (1,0)

```
SELECT C.nomC
FROM Clients as C, Produit as P, Commande as Co, ContenuCom as Cc
WHERE C.ncli = Co.ncli
      AND Co.ncom = Cc.ncom
      AND Cc.codeProd = P.codeProd
      AND P.occasion = "pâques"
      AND Co.ncli IN ( SELECT Co2.ncli
```

```
FROM Commande as Co2, Produit as P2, ContenuCom as Cc2
WHERE Co2.ncom = Cc2.ncom
      AND Cc2.codeProd = P2.codeProd
      AND P2.occasion = "Noël" )
```

Un produit ne peut pas avoir occasion=« pâques » et occasion=« Noël », ces sont donc deux produits différents.

On cherche les **clients** qui ont pu commandé un produit de Noël dans une commande, puis un produit de pâques dans une autre. On va donc utiliser le **ncli** avec le **IN**.

#### (4) Les produits (nom produit) mélangeant chocolat noir et chocolat blanc

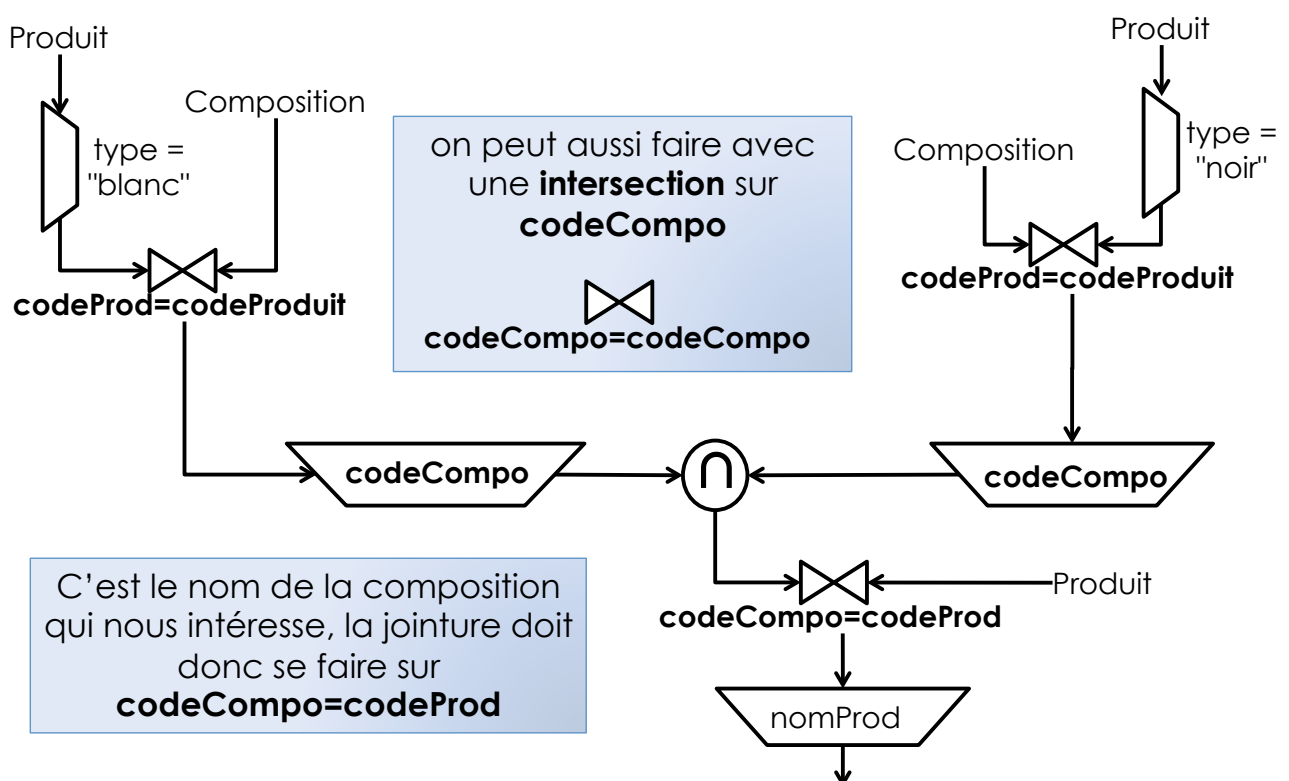
- Un produit qui mélange chocolat noir et blanc est forcément une composition.
- Le produit qu'on cherche est donc une composition qui contient au moins un produit de type=« blanc » et au moins un autre produit de type=« noir ».
- On va donc devoir regarder le contenu des compositions (table composition) pour trouver celle qui contient un produit en chocolat blanc (type=« blanc ») et un autre en chocolat noir (type=« noir »).
- Dans la table Composition, codeCompo est le codeProd du produit de type composition, alors que codeProduit est le codeProd de ceux qui participent à la composition.

Le produit n° 3 (Poule en chocolat) est une composition avec des œufs en chocolat noir et des œufs en chocolat blanc.

codeProd	type	nomProd
1	noir	œuf noir...
2	blanc	œuf blanc...
3	composition	Poule avec des œufs

codeCompo	codeProduit
3	1
3	2

#### (4) Les produits (nom produit) mélangeant chocolat noir et chocolat blanc (1,0)



#### (4) Les produits (nom produit) mélangeant chocolat noir et chocolat blanc (1,0)

```

SELECT  Pcompo.nomProd
FROM    Produit as Pcompo, Produit as Pblanc,
        Composition C
WHERE   Pblanc.type= "blanc"
        AND Pblanc.codeProd = C.codeProduit
        AND Pcompo.codeProd = C.codeCompo
        AND C.codeCompo  IN ( SELECT C2.codeCompo
                              FROM  Produit as Pnoir, Composition as C2
                              WHERE Pnoir.type = "noir"
                              AND  Pnoir.codeProd = C2. codeProduit )
    
```

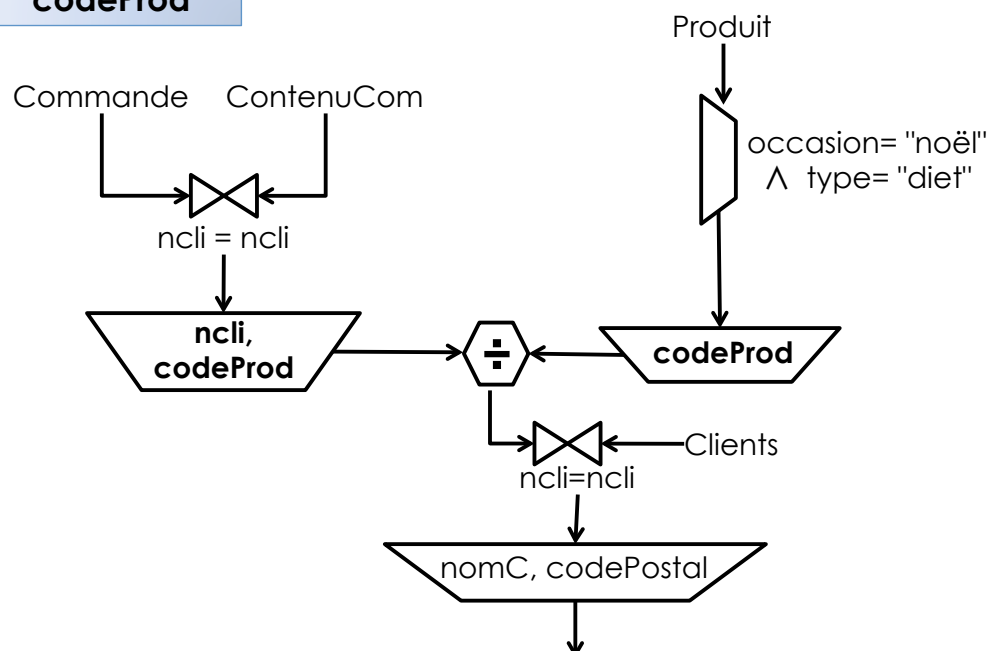
Le produit qui nous intéresse est la **composition**.  
Le **codeProd** de celui qui nous intéresse correspond au **codeCompo**.

Un produit n'a qu'un type, il ne peut pas être blanc, noir et une composition à la fois. On a donc à faire à **trois produits** différents : la composition qu'on cherche, un en chocolat blanc et un en chocolat noir.

C'est la composition qui contient les deux autres produit, il faut donc utiliser le **codeCompo** pour l'**intersection**.

#### (5) Les clients (nom et code postal) qui ont déjà commandé tous les produits de Noël diet proposés. (1,5)

$R(A, B) \div S(B)$   
 $\downarrow$              $\downarrow$   
**ncli**            **codeProd**



### (5) Les clients (nom et code postal) qui ont déjà commandé tous les produits de Noël diet proposés. (1,5)

```
SELECT C.nomC, C.codePostal
FROM Clients as C
WHERE NOT EXISTS (
  SELECT *
  FROM Produit as P
  WHERE P.type = "diet" AND P.occasion="noël"
```

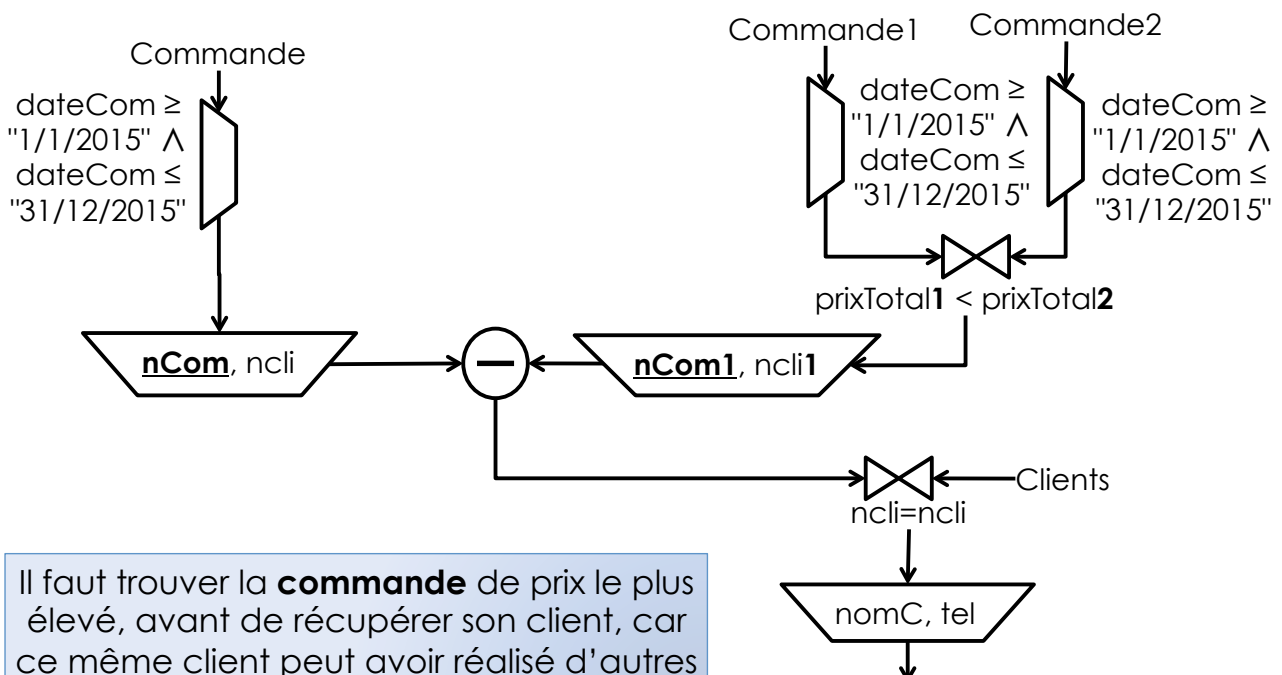
Il y a des conditions sur notre B (être de Noël diet), on les trouvera dans le WHERE du 2<sup>e</sup> SELECT.

```
AND NOT EXISTS (
  SELECT *
  FROM Commande as Co, ContenuCom as Cc
  WHERE Co.ncom = Cc.ncom
  AND C.ncli = Co.ncli
  AND P.codeProd = Cc.codeProd ) )
```

Ce qu'on cherche (**notre A**)  
→ clients (**ncli**)  
Pour tous les produit (**notre B**)  
→ Produit (**codeProd**)  
Ce qui permet de **relier A et B**  
→ **commande** et **contenuCom**

Dans le **dernier SELECT** on relie les tables entre elles, mais aussi avec A (clients) et avec B (codeProd).

### (6) Les clients (nom et téléphone) ayant effectué la commande la plus élevée cette année. (1,5)



Il faut trouver la **commande** de prix le plus élevé, avant de récupérer son client, car ce même client peut avoir réalisé d'autres commandes de valeur inférieur

**(6) Les clients (nom et téléphone) ayant effectué la commande la plus élevée cette année. (1,5)**

```
SELECT  C.nomC, C.tel
FROM    Clients as C, Commande as Co
WHERE   C.ncli = Co.ncli
        AND Co.dateCom BETWEEN "1/1/2015" AND "31/12/2015"
        AND Co.prixTotal = ( SELECT MAX (Co2.prixTotal)
                             FROM  Commande as Co2
                             WHERE Co2.dateCom BETWEEN "1/1/2015"
                             AND "31/12/2015" )
```

La date dans les 2 select est importante car le client qui a fait la commande la plus élevée cette année ne l'a pas fait les autres années.

**(7) Le prix moyen des produits par type en ordre décroissant (par type en ordre alphabétique inversée). (1,5)**

```
SELECT  AVG (prixKg) , type
FROM    Produit
GROUP BY type
ORDER BY type DESC
```

On souhaite la moyenne par type, il faut donc grouper les produits par type pour calculer la moyenne → **GROUP BY**

Le **DESC** va permettre d'afficher en ordre décroissant

L'attribut qui est dans l'**ORDER BY** se trouve forcément dans le **SELECT**



## (8) Les noms des clients en ordre alphabétique ayant commandé plus de 5 produits diet en 2015. (2,0)

```
SELECT C.nomC, C.ncli
FROM Clients as C, Commande as Co, ContenuCom as Cc, Produit as P
WHERE C.ncli = Co.ncli AND Co.ncom=Cc.ncom AND Cc.codeProd=P.codeProd
AND P.type = "diet" AND Co.dateCom BETWEEN "1/1/2015" AND "31/12/2015"
GROUP BY C.ncli
HAVING Count(*)>5
ORDER BY C.nomC
```

Les 5 produits n'ont pas été forcément acquis dans une même commande.  
On doit donc **grouper (GROUP BY) par client.**

Il n'y a pas d'attribut pouvant nous donner le nombre d'articles d'un type. Il faut donc compter les n-uplets. → **HAVING**  
Et si on a un **HAVING**, on a un **GROUP BY**

On peut aussi utiliser  
**Count(P.codeProd)**

## (9) Le prix moyen des compositions contenant plus de 3 produits. (1,5)

```
SELECT AVG (P.prixKg) , Cp.codeCompo
FROM Produit as P, Composition as Cp
WHERE P.codeProd = Cp.codeCompo
GROUP BY Cp.codeCompo
HAVING Count(*) > 3
```

On s'intéresse aux compositions, c'est donc **Cp.codeCompo** qui identifie les produits qui nous intéressent.