

Partie 2

Objectifs et Programme

- 1) Comment construire une « bonne » base de données relationnelle ?
- 2) Comment retrouver des informations dans une BD relationnelle ?
 - Langage algébrique
 - Langage SQL

Base de données Buveurs

Buveur (NumBuveur, NomB, PrenomB, VilleB)

Les clés primaires sont soulignées en **trait plein**

Viticulteur (NumVitic, NomV, PrénomV, VilleV)

Vin (NumVin, Cru, Millesime, Région, NumVitic)

Les clés étrangères sont soulignées en **pointillés**

Commande (NumCom, NumBuveur, NumVin, QtteCommandee, DateCom)

Livraison (NumCom, QtteLivree, DateLiv)

Buveur	NumBuveur	NomB	PrenomB	VilleB
1	1800	ECOBET	ALINE	DIJON
2	1900	MAURY	FRANCOIS	PARIS
3	1900	JAMET	FLORENCE	VERSAILLES
4	1900	SAUVINEL	SOPHIE	NICE
5	1900	SAUTHIER	ROBERT	PARIS
6	1900	FRICAL	PYRIS	MACON
7	1900	FRACET	PHILIPPE	MACON
8	1900	DUPOND	ALAIN	PARIS
9	1900	COMBES	EDOUARD	PARIS
10	1900	DARIEUX	BLAISE	CHINON

Viticulteur	NumVitic	NomV	PrenomV	VilleV
1	1	MOULIN	JEAN	MACON
2	2	MOIRIN	JULES	BENAS
3	3	VALLEY	CLAUDE	SAINT NICOLAS
4	4	COULIS DUTHEL		CHINON
5	5	PARENT	PIERRE	POMMARD
6	6	BARANCOURT		TOURS SUR MARNE
7	7	JADOT	LOUIS	BEAUNE
8	8	MEDEVILLE	CHARLES	PREIGNAC
9	9	TISSOT	JACQUES	ARBOIS
10	10	LYNCH BAGES		PAULLAC
11	11	CHATEAU MONBRISON		MARGAUX
12	12	CHATEAU MOULIN A VENT		CASTELNAU DE MEDOC

Vin	NumVin	Cru	Millesime	Région	NumVitic
1	100	SAUTERNES	1982	BORDEAUX	18
2	100	PAULLAC	1982	BORDEAUX	21
3	100	CHINON	1983	LOIRE	5
4	100	CHINON	1978	LOIRE	4
5	100	CHAMBERTIN	1980	BOURGOGNE	1
6	100	MARGAUX	1978	BORDEAUX	22
7	100	MOULIN	1981	BORDEAUX	23
8	100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
9	100	CHINON	1985	LOIRE	4
10	100	BORGUES	1985	LOIRE	4
11	100	POMMARD	1976	BOURGOGNE	10
12	100	MONTHELIE	1978	BOURGOGNE	10
13	100	BORGUES	1984	LOIRE	3
14	100	BOLEY	1984	CHAMPAGNE	15
15	100	MEURSAULT	1980	BOURGOGNE	18
16	100	SAINT EMILION	1981	BORDEAUX	18

Commande	NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1	1800	90	6	10/06/87	
2	1900	100	18	20/11/87	
3	1900	88	24	10/09/87	
4	1800	175	6	12/02/87	
5	1900	140	3	18/01/87	
6	1900	140	8	24/12/87	
7	1900	88	3	24/12/87	
8	1800	140	12	18/12/88	
9	1900	140	12	12/12/88	
10	1900	120	6	13/12/88	

Livraison	NumCom	QtteLivree	DateLiv
1	2	12	20/12/87
2	6	6	18/01/87
3	2	6	22/11/87
4	2	12	20/11/87
5	6	6	20/11/87
6	8	8	20/11/87
7	1	3	18/12/88

1. Comment construire une « bonne » base de données relationnelle ?

SQL : Structured Query Language

Langage de définition des Données (LDD)



- Créer/ modifier/ supprimer les schémas de relation
 - ➔ CREATE TABLE : créer une table (pour DM)
 - ➔ ALTER TABLE : modifier le schéma
 - ➔ DROP TABLE : supprimer une table

Langage de Manipulation des Données (LMD)



- Créer, modifier, supprimer, interroger les données (n-uplet) = contenu

- ➔ **INSERT** : insérer un n-uplet (pour dm)
- ➔ **UPDATE** : mettre à jour
- ➔ **DELETE** : supprimer un n-uplet
- ➔ **SELECT** : interroger la base (récupérer les n-uplet)

CREATE TABLE

Buveur (NumBuveur, NomB, PrenomB, VilleB)

```
CREATE TABLE Buveur
( NumBuveur Integer PRIMARY KEY,
  Nom B Char (25) NOT NULL,
  PrenomB char (25) NOT NULL,
  VilleB Char(30)
);
```

NumBuveur est la **clé primaire**

Domaine de valeur : le numéro du buveur est un entier. Les autres attributs sont des chaînes de caractères

Valeur obligatoire : le prénom du buveur ne peut pas être nul (vide) : il doit être renseigné. La ville est optionnelle

Domaines de valeurs à connaître

- **Types de données**
 - *Numériques* :
 - **INTEGER** : entier long
 - **FLOAT** : Nombre réel
 - *Chaînes de caractères* :
 - **CHAR (n)** : chaîne de caractères de **longueur n**
 - *Temporelles* :
 - **DATE** : date (2009-11-09)
 - **TIME** : heure (00 :11 :22 :333)
 - *Logiques* :
 - **BOOLEAN** : valeur booléenne, vrai ou faux (« True » ou « False »)

Vin (NumVin, Cru, Millesime, Région, NumVitic)

```
CREATE TABLE Vin (
  NumVin INTEGER PRIMARY KEY,
  Cru CHAR (15),
  Millesime INTEGER,
  Region CHAR (15),
  NumVitic INTEGER REFERENCES viticulteur (NumVitic) );
```

NumVitic est une **clé étrangère**

Livraison (NumCom, QtteLivree, DateLiv)

```
CREATE TABLE Livraison (  
    NumCom INTEGER,  
    QtteLivree INTEGER,  
    DateLiv DATE,  
    PRIMARY KEY (NumCom, DateLiv)  
    FOREIGN KEY (NumCom) REFERENCES commande (NumCom) );
```

NumCom et DateLiv
forment la **clé primaire**

NumCom est une **clé étrangère**

Manipulation de la base de données

- **INSERT INTO** : ajout de nouveaux n-uplets

```
INSERT INTO BUVEUR VALUES (1500, 'FRADET', 'PHILIPPE', 'MACON')  
INSERT INTO VITICULEUR VALUES (15, 'BARANCOURT', NULL, 'TOURS SUR MARNE');
```

- **SELECT** : + tard (requêtes)

```
SELECT      NomB, PrenomB  
FROM        Buveur  
WHERE       VilleB = 'Macon'
```

Guillemet = chaîne de caractère, date, heure

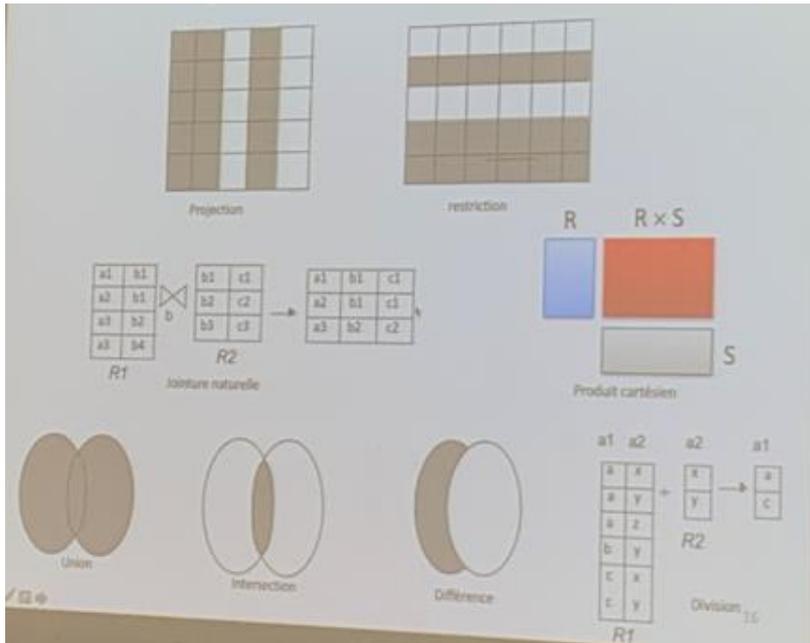
2. Comment retrouver des informations dans une BD relationnelle ?

- Langage algébrique
- Langage SQL

Interrogation des bases de données

Interrogation = requêtes (Query)

- **Algèbre relationnelle**
 - 8 opérations
- **SQL (Structured Query Language)**
 - 8 opérations
 - + fonctions de calculs
 - + opérateurs d'agrégation
- **Algèbre relationnelle**



Projection : on s'intéresse seulement à quelques colonnes

Restriction : on s'intéresse à quelques lignes

Produits cartésiens : assembler deux tables

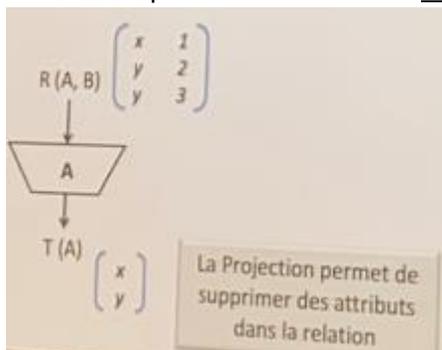
Intersections : on cherche les lignes qui appartiennent simultanément dans 2 tables

Différences : éléments qui sont dans l'un et pas dans l'autres

Algèbre : Projection



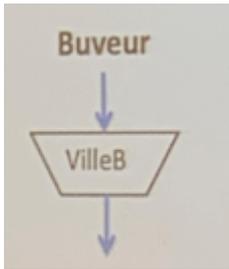
La **projection** d'une **relation** $R(A_1, \dots, A_i, \dots, A_k, \dots, A_n)$ sur les attributs A_i, \dots, A_k est une relation $T(A_i, \dots, A_k)$ dont les n -uplet sont ceux de R en éliminant les attributs non mentionnés et les n -uplet en double.



Exemple :

Quelle sont les villes des buveurs enregistrés dans la base ?

Buveur (NumBuveur, NomB, PrenomB, VilleB)



On peut écrire du haut vers le bas ou vice-versa

NumBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

Quelles sont les villes des buveurs enregistrés dans la base ?

N-uplets en double : supprimés

DIJON
PARIS
VERSAILLES
NICE
MACON
CHINON

Nouveau schéma composé des attributs conservés

Doublets sont enlevés en algèbre mais pas en SQL.

Le schéma d'une table après projection est forcément différent

Quelles sont les noms et prénoms des buveurs enregistrés dans la base ?

NumBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

Quelles sont les noms et prénoms des buveurs enregistrés dans la base ?

NomB PrenomB

COMBET ALINE
MAURY FRANCOIS
JAMET FLORENCE
GRAVINEL SOPHIE
GAUTHIER ROBERT
PASCAL YVES
FRADET PHILIPPE
DUPOND ALAIN
COMBES EDOUARD
DURIEUX BLAISE

Nouveau schéma composé des attributs conservés

SQL : affichage du contenu complet d'une table

SELECT *
FROM table → relation concernée

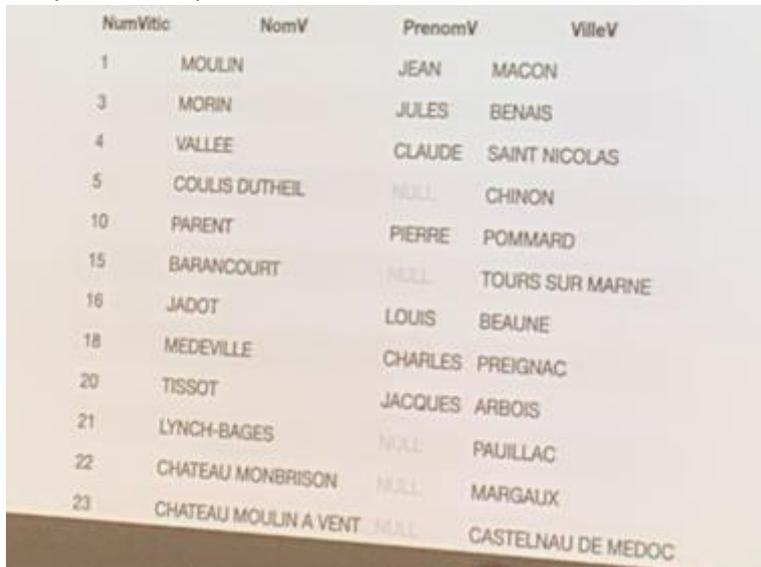
Le **résultat** de l'opération est **l'ensemble de tous les attributs de tous les n-uplets de la table**

Site : bdd.univ-paris1.fr

Exemple : Lister toutes les informations sur les viticulteurs :

SELECT *
FROM Viticulteur ;

Le symbole * représente tous les attributs



NumVitic	NomV	PrenomV	VilleV
1	MOULIN	JEAN	MACON
3	MORIN	JULES	BENAI
4	VALLEE	CLAUDE	SAINT NICOLAS
5	COULIS DUTHEIL	NULL	CHINON
10	PARENT	PIERRE	POMMARD
15	BARANCOURT	NULL	TOURS SUR MARNE
16	JADOT	LOUIS	BEAUNE
18	MEDEVILLE	CHARLES	PREIGNAC
20	TISSOT	JACQUES	ARBOIS
21	LYNCH-BAGES	NULL	PAUILLAC
22	CHATEAU MONBRISON	NULL	MARGAUX
23	CHATEAU MOULIN A VENT	NULL	CASTELNAU DE MEDOC

SQL : Projection

Opérateur de projection : liste des attributs demandés

SELECT attr₁, attr₂...
FROM table (relation concernée par la projection)

Le **résultat** de l'opération **SELECT...FROM**

Est **l'ensemble de TOUS les n-uplet** (doublons conservés) mais contenant uniquement les **attributs** indiqués (attr₁,attr₂...)

SELECT correspond à une projection sur les attributs recherchés. Ce qui vient derrière le **SELECT** correspond à l'information attendue (**résultat**)

Exemple : Quelles sont les villes des buveurs enregistrés dans la base ?

NumBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

Quelles sont les villes des buveurs enregistrés dans la base ?

SELECT VilleB FROM Buveur ;

VilleB

DIJON
PARIS
VERSAILLES
NICE
PARIS
MACON
MACON
PARIS
PARIS
CHINON

N-uplets en double : conservés

Nouveau schéma composé des attributs conservés

UNIVERSITÉ PARIS 1 PANTHÉON SORBONNE

32

SQL : Projection et suppression des doublons

Opérateur de projection : liste des attributs demandés

SELECT DISTINCT attr₁,attr₂...

FROM → table relation concernée par la projection

Le résultat de l'opération **SELECT... FROM**

Est l'ensemble des n-uplet **DISTINCTS** mais contenant uniquement les attributs indiqués (attr₁,attr₂...)

= équivalent de la projection avec l'algèbre relationnelle

Exemple :

Quelles sont les villes des buveurs enregistrés dans la base ?

NumBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

Quelles sont les villes des buveurs enregistrés dans la base ?

SELECT DISTINCT VilleB FROM Buveur ;

VilleB

DIJON
PARIS
VERSAILLES
NICE
MACON
CHINON

N-uplets en double : supprimés

Nouveau schéma composé des attributs conservés

UNIVERSITÉ PARIS 1 PANTHÉON SORBONNE

Quelles sont les noms et prénoms des buveurs enregistrés dans la base ?

Quelles sont les noms et prénoms des buveurs enregistrés dans la base ?

NumBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

`SELECT NomB, PrenomB
FROM Buveur ;`

NomB	PrenomB
COMBET	ALINE
MAURY	FRANCOIS
JAMET	FLORENCE
GRAVINEL	SOPHIE
GAUTHIER	ROBERT
PASCAL	YVES
FRADET	PHILIPPE
DUPOND	ALAIN
COMBES	EDOUARD
DURIEUX	BLAISE

Nouveau schéma composé des attributs conservés

Exercice :

Quels sont les crus de vins enregistrés dans la base

Buveur (NumBuveur, NomB, PrenomB, VilleB)

Viticulteur (NumVitic, NomV, PrénomV, VilleV)

Vin (NumVin, Cru, Millesime, Région, NumVitic)

Commande (NumCom, NumBuveur, NumVin, QtteCommandee, DateCom)

Livraison (NumCom, QtteLivree, DateLiv)

Quels sont les crus des vins enregistrés dans la base ?

NumVin	Cru	Millesime	Région	NumVitic
30	SAUTERNES	1982	BORDEAUX	18
34	PAULLAC	1982	BORDEAUX	21
35	CHINON	1983	LOIRE	5
36	CHINON	1978	LOIRE	4
38	CHAMBERTIN	1980	BOURGOGNE	1
39	MARGAUX	1978	BORDEAUX	22
38	MOULIS	1981	BORDEAUX	23
120	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
120	CHINON	1985	LOIRE	4
128	BOURGUEREL	1985	LOIRE	4
140	POMMARD	1976	BOURGOGNE	10
150	MONTHELIE	1978	BOURGOGNE	10
160	BOURGUEREL	1984	LOIRE	3
160	BOUZY	1984	CHAMPAGNE	15
170	MEURSAULT	1980	BOURGOGNE	16
190	SAINT EMILION	1981	BORDEAUX	18

`SELECT DISTINCT Cru
FROM Buveur ;`

N-uplets en double : supprimés

Nouveau schéma composé des attributs conservés

Vin
↓
Cru

SQL : Variantes du SELECT

- Possibilité d'afficher un autre nom (ALIAS) pour l'attribut
 - o Uniquement pour l'affichage, la table n'est pas modifiée

Exemple : Quelles sont les villes des buveurs enregistrés dans la base ?

Quelles sont les villes des buveurs enregistrés dans la base ?

SQL Query	Result
<code>SELECT VilleB FROM Buveur ;</code>	VilleB DIJON PARIS VERSAILLES NICE PARIS MACON MACON PARIS PARIS CHINON
<code>SELECT VilleB AS 'Ville des Buveurs' FROM Buveur ;</code>	Ville des Buveurs DIJON PARIS VERSAILLES NICE PARIS MACON MACON PARIS PARIS CHINON

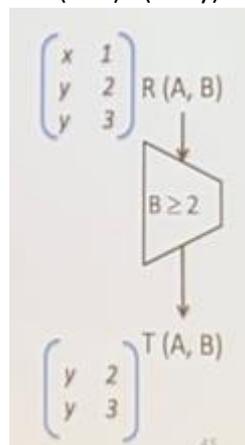
Annotations: A red arrow points from the word "Alias" to the text 'Ville des Buveurs' in the second query. A blue arrow points from the text "Apostrophes ou guillemets" to the single quotes around 'Ville des Buveurs' in the second query.

Algèbre : Restriction

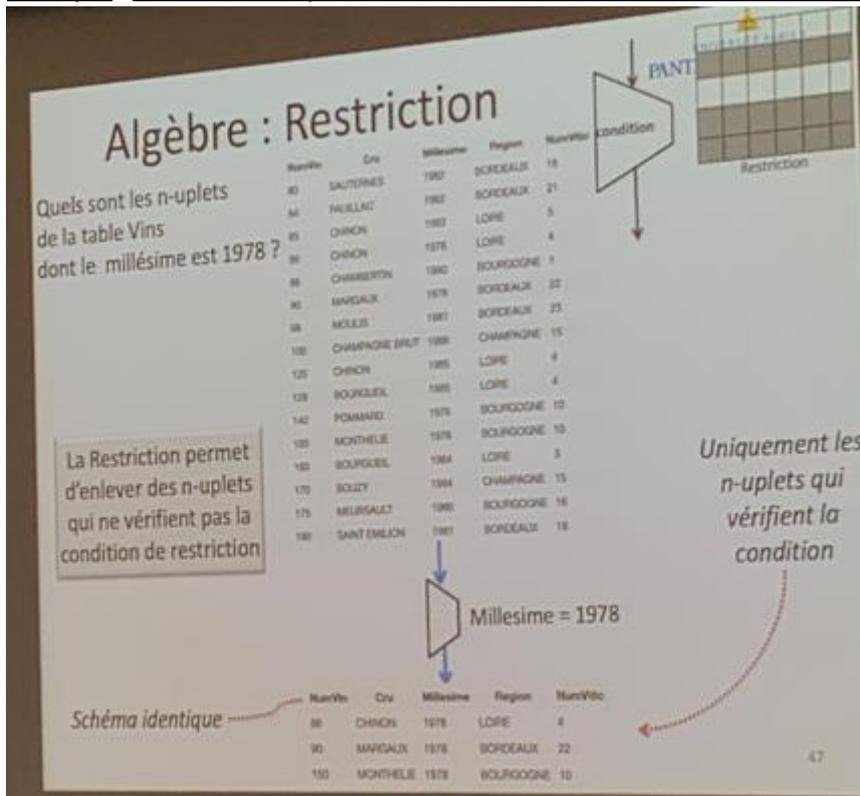


La restriction d'une relation R sur une condition Q est une relation T de même schéma dont les n-uplets sont ceux de R qui vérifient la condition Q

- La condition Q est construite à partir :
 - o D'opérateurs relationnels : $<$, \geq , $>$, \geq , $=$, \neq
Ex : $B > 1$, $A \neq y$
 - o D'opérateurs logiques : et (\wedge), ou (\vee)
Ex : $(B=1) \wedge (A \vee y)$



Exemple : Quels sont n-uplets de la table Vins dont le millésime est 1978 ?



SQL: restriction

Opérateur de projection : liste des attributs demandés

SELECT attr₁, attr₂...

FROM table → relation concernée par la requête

WHERE condition

Opérateurs de restriction : conditions de conservation des n-uplets

Le résultat de l'opération SELECT... FROM... WHERE

Est l'ensemble de tous les n-uplet (pas de suppression des doublons) respectant la condition et contenant uniquement les attributs indiqués (attr₁, attr₂...)

SQL : Restriction

Quels sont les n-uplets de la table Vins dont le millésime est 1978 ?

NumVite	Cru	Millesime	Region	NumVite
80	SALTERNES	1982	BORDEAUX	18
84	FRULLAC	1982	BORDEAUX	21
85	CHINON	1983	LOIRE	3
86	CHINON	1978	LOIRE	4
88	CHAMBERTIN	1980	BOURGOGNE	1
90	MARGAUX	1978	BORDEAUX	22
98	MOULIS	1981	BORDEAUX	23
100	CHAMPAGNE EFFORT	1989	CHAMPAGNE	15
120	CHINON	1985	LOIRE	4
138	BOURGUEIL	1980	LOIRE	4
143	POMMARD	1978	BOURGOGNE	10
150	MONTHÉLIE	1978	BOURGOGNE	10
160	BOURGUEIL	1984	LOIRE	3
170	BOUZY	1984	CHAMPAGNE	15
175	MEURSAULT	1980	BOURGOGNE	16
190	SANT-EMILION	1981	BORDEAUX	18

La Restriction permet d'enlever des n-uplets qui ne vérifient pas la condition de restriction

Uniquement les n-uplets qui vérifient la condition

```
SELECT *
FROM Vin
WHERE Millesime = 1978 ;
```

Schéma identique

NumVite	Cru	Millesime	Region	NumVite
86	CHINON	1978	LOIRE	4
90	MARGAUX	1978	BORDEAUX	22
150	MONTHÉLIE	1978	BOURGOGNE	10

51

Algèbre : Restriction et Projection

Quelle requête pour obtenir le nom et le prénom des buveurs parisiens ?

Table Buveur

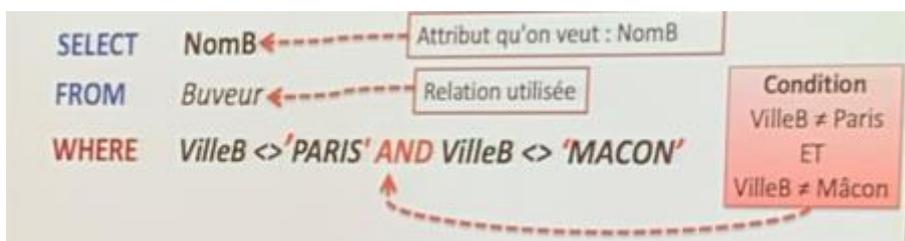
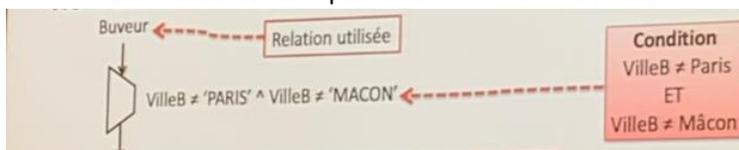
Condition de restriction : VilleB = 'PARIS'

Ce qu'on cherche (projection) NomB, PrenomB

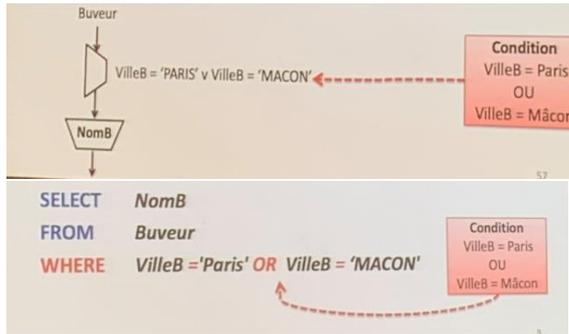


Algèbre : Restriction avec condition complexe

- Nom des buveurs qui n'habite ni Paris ni Macon



- Nom des buveurs qui habitent Paris ou Macon



SQL : Clause WHERE avec opérateur LIKE

L'opérateur **LIKE** permet de tester si un attribut de type chaîne de caractères contient une **sous-chaîne**

« _ » remplace 1 seul caractère : VilleB **LIKE '___ ON'**

	VilleB
SELECT Ville B	
FROM Buveur	→
WHERE VilleB LIKE '___ ON'	DIJON
	MACON
	MACON

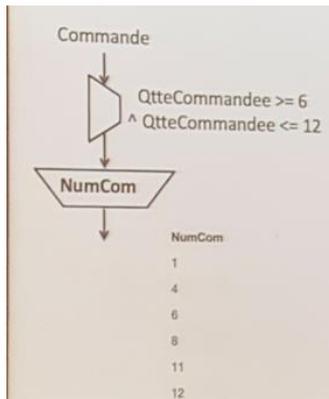
“%” remplace une sous chaîne : Ville **LIKE '%ON'**

	VilleB
SELECT VilleB	
FROM Buveur	→
WHERE VilleB LIKE '%ON'	DIJON
	MACON
	MACON
	CHINON

Algèbre : intervalle

- Numéro des commandes où le nombre de bouteilles est compris entre 6 et 12

NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1	1600	90	6	1987-06-10
2	1500	100	18	1987-11-20
3	1500	88	24	1987-09-10
4	1600	175	6	1987-02-12
5	1300	140	3	1987-03-18
6	1100	140	8	1987-12-24
7	1200	88	3	1987-12-24
8	1400	140	12	1988-12-10
11	1200	140	12	1989-12-12
12	1100	120	6	1989-12-13



```
SELECT NumCom
FROM Commande
WHERE QtteCommandee >= 6 AND QtteCommandee <=12 ;
```

Possibilité d'utiliser l'opérateur BETWEEN

```
SELECT NumCom
FROM Commande
WHERE QtteCommandee BETWEEN 6 AND 12.
```

Il peut s'agir d'intervalles entre des chaînes de caractères, des dates...

SQL: IS NULL/ IS NOT NULL

Condition dans la clause **WHERE**

- **IS NULL**
 - o Si la valeur d'un attribut est inconnue (NULL)
- **IS NOT NULL**
 - o Si la valeur d'un attribut n'est pas inconnue (NOT NULL)

NumVitic	NomV	PrenomV	VilleV
1	MOULIN	JEAN	MACON
3	MORIN	JULES	BENAIS
4	VALLEE	CLAUDE	SAINT NICOLAS
5	COULIS DUTHEIL	NULL	CHINON
10	PARENT	PIERRE	POMMARD
15	BARANCOURT	NULL	TOURS SUR MARNE
16	JADOT	LOUIS	BEAUNE
18	MEDEVILLE	CHARLES	PREIGNAC
20	TISSOT	JACQUES	ARBOIS
21	LYNCH-BAGES	NULL	PAULLAC
22	CHATEAU MONBRISON	NULL	MARGAUX
23	CHATEAU MOULIN A VENT	NULL	CASTELNAU DE MEDOC

Nom des viticulteurs qui n'ont pas renseigné leur prénom :

```
SELECT NomV
FROM Viticulteur
WHERE PrenomV IS NULL
```

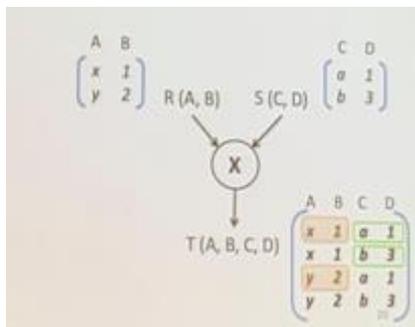
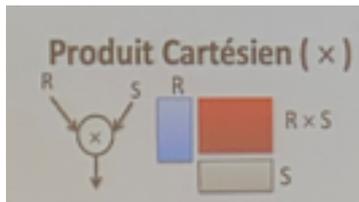
NomV

COULIS DUTHEIL

BARANCOURT
 LYNCH-BAGES
 CHATEAU MONBRISON
 CHATEAU MOULIN A VENT

Algèbre : Produit cartésien (X)

Le **produit cartésien** de 2 relations R et S de schéma quelconque est une relation T ayant pour schéma la concaténation de ceux de R et de S et pour n-uplets toutes les combinaisons des n-uplets de R et de S



Propriétés :

- Commutatif : $R * S = S * R$

Produit cartésien entre Buveur et Commandes

Buveur

NomBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

Commande

NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1	1600	90	6	10/06/87
2	1500	100	18	20/11/87
3	1500	88	24	10/09/87
4	1600	175	6	12/02/87
5	1300	140	3	18/03/87
6	1100	140	8	24/12/87
7	1200	88	3	24/12/87
8	1400	140	12	10/12/88
11	1200	140	12	12/12/89
12	1100	140	6	13/12/89

Concaténation des 2 schémas

NumBuveur	NomB	PreNomB	VilleB	NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1100	COMBET	ALINE	DIJON	1	1600	90	6	1987-06-10
1100	COMBET	ALINE	DIJON	2	1500	100	18	1987-11-20
1100	COMBET	ALINE	DIJON	3	1500	88	24	1987-09-10
1100	COMBET	ALINE	DIJON	4	1600	175	6	1987-02-12
1100	COMBET	ALINE	DIJON	5	1300	140	3	1987-03-18
1100	COMBET	ALINE	DIJON	6	1100	140	8	1987-12-24
1100	COMBET	ALINE	DIJON	7	1200	88	3	1987-12-24
1100	COMBET	ALINE	DIJON	8	1400	140	12	1988-12-10
1100	COMBET	ALINE	DIJON	11	1200	140	12	1989-12-12
1100	COMBET	ALINE	DIJON	12	1100	120	6	1989-12-13
1200	MAURY	FRANCOIS	PARIS	1	1600	90	6	1987-06-10
1200	MAURY	FRANCOIS	PARIS	2	1500	100	18	1987-11-20
1200	MAURY	FRANCOIS	PARIS	3	1500	88	24	1987-09-10
1200	MAURY	FRANCOIS	PARIS	4	1600	175	6	1987-02-12
1200	MAURY	FRANCOIS	PARIS	5	1300	140	3	1987-03-18
1200	MAURY	FRANCOIS	PARIS	8	1100	140	8	1987-12-24
1200	MAURY	FRANCOIS	PARIS	7	1200	88	3	1987-12-24
1200	MAURY	FRANCOIS	PARIS	8	1400	140	12	1988-12-10

Combinaison de toutes les n-uplets de Buveur et de Commande !!

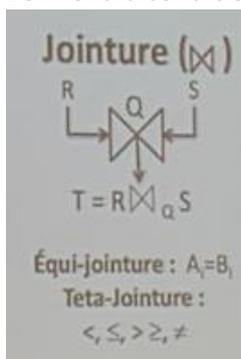
22

SELECT*

FROM Buveur, Commande

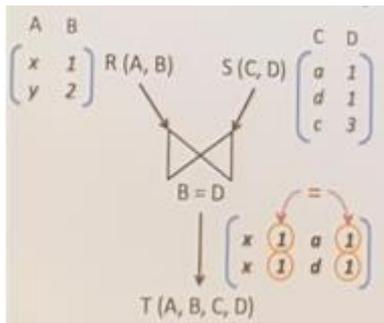
Algèbre : Jointure ☒

La **jointure** de 2 relations **R** et **S** sur **une condition Q** est la combinaison des relations **R** et **S** qui **vérifient la condition Q**



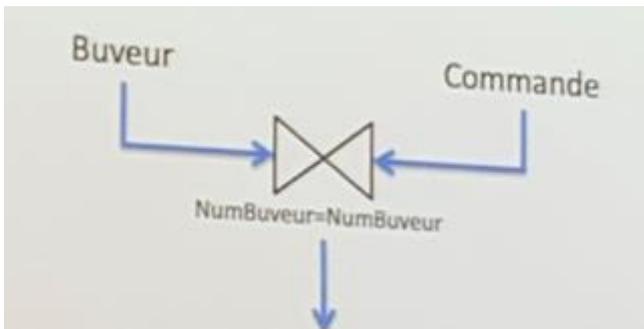
Condition au prédicat de jointure :

- Exemples : $B > 1$, $A \neq y$, $(B=1) \wedge (A = y)$



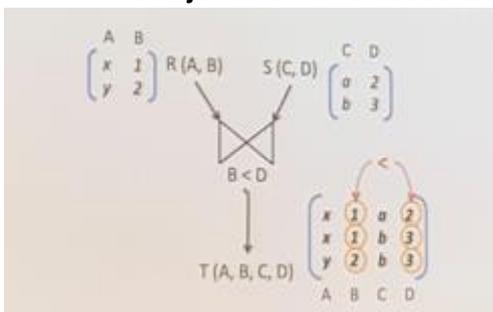
Propriétés :

- Commutative : $R \bowtie_{\theta} S = S \bowtie_{\theta} R$
- Associative : $R \bowtie_{\theta} (S \bowtie_{\theta} T) = (R \bowtie_{\theta} S) \bowtie_{\theta} T$



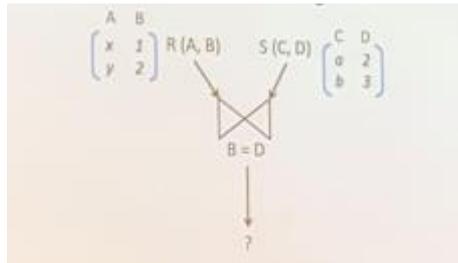
NumBuveur	NomB	PrenomB	VilleB	NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1600	DUPOND	ALAIN	PARIS	1	1600	90	6	1987-06-10
1500	FRADET	PHILIPPE	MACON	2	1500	100	18	1987-11-20
1500	FRADET	PHILIPPE	MACON	3	1500	88	24	1987-09-10
1600	DUPOND	ALAIN	PARIS	4	1600	175	6	1987-02-12
1300	JAMET	FLORENCE	VERSAILLES	5	1300	140	3	1987-03-18
1100	COMBET	ALINE	DIJON	6	1100	140	8	1987-12-24
1200	MAURY	FRANCOIS	PARIS	7	1200	88	3	1987-12-24
1400	GAUTHIER	ROBERT	PARIS	8	1400	140	12	1988-12-10
1200	MAURY	FRANCOIS	PARIS	11	1200	140	12	1989-12-12
1100	COMBET	ALINE	DIJON	12	1100	120	6	1989-12-13

- **Équi-Jointure** : opérateur d'égalité (ex. A=B)
- **Thêta-jointure** : dans les autres cas ($\neq, <, \geq, >, \leq$)



Exercice :

- Jointure
 - Équi-Jointure : condition d'égalité



- ❖ La clause **FROM** indique les **relations** à utiliser
 - Lorsque l'on introduit plusieurs relations, le SELECT porte sur le **produit cartésien**
 - La clause **WHERE** introduit alors la (les) condition(s) de jointure

SELECT*
FROM Buveur, Commande
WHERE NumBuveur = NumBuevur ➔ FAUX !!

SELECT*
FROM Buveur, Commande
WHERE Buveur.NumBuveur= Commande.NumBuveur

SELECT*
FROM AS B, Commande **AS C**
WHERE B.NumBuveur= **C**.NumBuveur

Exemple de jointures :

- **FROM/ WHERE** : quelques exemples de jointure
 - Lister les **noms des buveurs** qui ont fait des commandes et **les dates de ces commandes**

Buveur

NomBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

Commande

NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1	1600	90	6	10/06/87
2	1500	100	18	20/11/87
3	1500	88	24	10/09/87
4	1600	175	6	12/02/87
5	1300	140	3	18/03/87
6	1100	140	8	24/12/87
7	1200	88	3	24/12/87
8	1400	140	12	10/12/88
11	1200	140	12	12/12/89
12	1100	140	6	13/12/89

```
SELECT NomB, DateCom
FROM Buveur AS B, Commande AS C
WHERE B.NumBuveur = C.NumBuveur
```

L'usage d'un **alias (AS...)** permet de **distinguer les attributs** de la table Fournisseur de ceux de la table Livraison

Ou bien

```
SELECT NomB, DateCom
FROM Buveur B, Commande C
WHERE B.NumBuveur = C.NumBuveur
```

NomB	DateCom
DUPOND	1987-06-10
FRADET	1987-11-20
FRADET	1987-09-10
DUPOND	1987-02-12
JAMET	1987-03-18
COMBET	1987-12-24
MAURY	1987-12-24
GAUTHIER	1988-12-10

- Lister les **noms des buveurs** parisiens qui ont fait des commandes et **les dates de ces commandes**

```
SELECT NomB, DateCom
```

Attributs que l'on veut (résultat attendu)

```
FROM Buveur B, Commande C
```

Relations nécessaires

```
WHERE B.NumBuveur = C.NumBuveur AND VilleB = 'PARIS'
```

Condition de jointure

Condition de restriction

Exemples de jointures :

Lister les crus de Vins produits par des viticulteurs dont le nom commence par 'CHATEAU'. On indiquera également le nom du viticulteur associé à chaque cru.

```

SELECT Cru, NomV
FROM Vin V, Viticulteur Vit
WHERE V.NumVitic = Vit. numVitic
AND nomV LIKE 'CHATEAU%'

```

Condition de jointure

Restriction : nom de viticulteur
« ressemble à » 'CHATEAU%'

Quelle opération ?

1)

COMMANDES

N° COMMANDE	PERIODE	MONTANT
28	Octobre	1986
29	Octobre	2024
30	Novembre	1610
52	Décembre	512



N°COMMANDE	PERIODE	MONTANT
28	Octobre	1986
29	Octobre	2024

Restriction

2)

VINS

Cru	Mill	Région	Qualité
VOLNAY	1983	BOURGOGNE	A
VOLNAY	1979	BOURGOGNE	B
CHENAS	1983	BEAUJOLAIS	A
JULIENAS	1986	BEAUJOLAIS	C



Cru	Région
VOLNAY	BOURGOGNE
CHENAS	BEAUJOLAIS
JULIENAS	BEAUJOLAIS

Projection

3)

LIVRE

Titre	Auteur
X	Hugo
Y	Yourcenar

EDITION

Couleur	Edition
Rouge	Luxe
Blanc	Broché
Vert	Cartonné

?



Titre	Auteur	Couleur	Edition
X	Hugo	Rouge	Luxe
X	Hugo	Blanc	Broché
X	Hugo	Vert	Cartonné
Y	Yourcenar	Rouge	Luxe
Y	Yourcenar	Blanc	Broché
Y	Yourcenar	Vert	Cartonné

Produit cartésien

4)

EMPLOYE

Num_emp	Nom_emp	Num_service
02	E1	S1
10	E2	S8
72	E3	S6
62	E4	S1

SERVICE

Num_service	Nom_service
S1	Informatique
S6	Mathématique
S8	Sociologie
S4	Anglais

?



Num_emp	Nom_emp	Num_service	Nom_service
02	E1	S1	Informatique
10	E2	S8	Sociologie
72	E3	S6	Mathématiques
62	E4	S1	Informatique

→ Jointure entre les deux tables

Base de données Buveurs

Buveur (NumBuveur, NomB, PrenomB, VilleB)

Viticulteur (NumVitic, NomV, PrénomV, VilleV)

Vin (NumVin, Cru, Millesime, Région, NumVitic)

Commande (NumCom, NumBuveur, NumVin, QtteCommandee, DateCom)

Livraison (NumCom, QtteLivree, DateLiv)

Create table de 12 bouteilles livré aujourd'hui

```
CREATE TABLE Livraison (  
    NumCom INTEGER REFERENCES Commandes (NumCom),  
    QtteLivree INTEGER,  
    DateLiv DATE,  
    PRIMARY KEY ( NumCom, DateLiv)  
);
```

```
CREATE TABLE Livraison (  
    NumCom INTEGER  
    QtteLivree INTEGER  
    DateLiv DATE  
    PRIMARY KEY ( NumCom, DateLiv)  
    FOREIGN KEY (NumCom) REFERENCES Commandes (NumCom)  
);
```

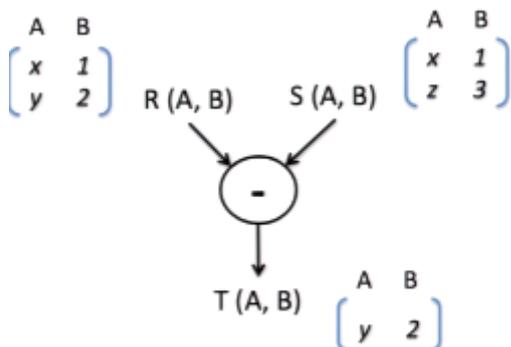
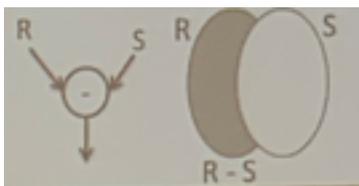
Je fais l'hypothèse que numcom = 345

```
INSERT INTO Livraison VALUES ( 345, NULL, '2021-10-18' );
```

Requête simple : nécessite qu'un seul SELECT

Algèbre : différence (-)

La **différence** de 2 relations **R** et **S** de **même schéma** est une **relation T** de **même schéma** contenant les **n-uplets** qui $\in R$ et $\notin S$.



➔ Nécessite deux SELECT (joueur qui ont joué que dans ce film)

Propriétés :

- NON commutatif : R-S ≠ S-R

Exemple :

- Quels sont les lignes présentes des viticulteurs présents dans la table Viticulteur1 et pas dans la table Viticulteur2 ?

NumVitic	NomV	PrenomV	VilleV
1	MOULIN	JEAN	MACON
3	MORIN	JULES	BENAIS
4	VALLEE	CLAUDE	SAINT NICOLAS
5	COULIS DUTHEL		CHINON
10	PARENT	PIERRE	POMMARD
15	BARANCOURT		TOURS SUR MARNE
16	JADOT	LOUIS	BEAUNE

NumVitic	NomV	PrenomV	VilleV
10	PARENT	PIERRE	POMMARD
15	BARANCOURT		TOURS SUR MARNE
16	JADOT	LOUIS	BEAUNE
18	MEDEVILLE	CHARLES	PREIGNAC
20	TESSOT	JACQUES	ARBOIS
21	LYNCH-BAGES		PAULLIAC
22	CHATEAU MONBRISON		MARGAUX
23	CHATEAU MOULIN A VENT		CASTELNAU DE MEDOC

même schéma

NumVitic	NomV	PrenomV	VilleV
1	MOULIN	JEAN	MACON
3	MORIN	JULES	BENAIS
4	VALLEE	CLAUDE	SAINT NICOLAS
5	COULIS DUTHEL		CHINON

Schéma identique à Viticulteur1 et à Viticulteur2

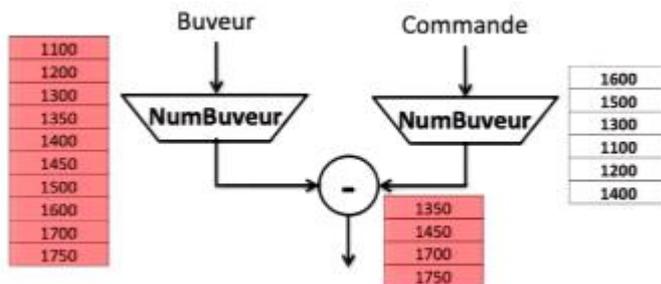
- Quel est le numéro des buveurs qui n'ont pas fait aucune commande ?

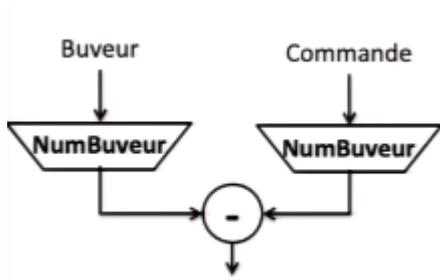
NumBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1	1600	90	6	10/06/87
2	1500	100	18	20/11/87
3	1500	88	24	10/09/87
4	1600	175	6	12/02/87
5	1300	140	3	18/03/87
6	1100	140	8	24/12/87
7	1200	88	3	24/12/87
8	1400	140	12	10/12/88
11	1200	140	12	12/12/89
12	1100	120	6	13/12/89

On cherche les numéros des buveurs présents dans la table buveur mais absents dans la table Commandes

Problème : les tables Buveurs et Commande n'ont pas le même schéma





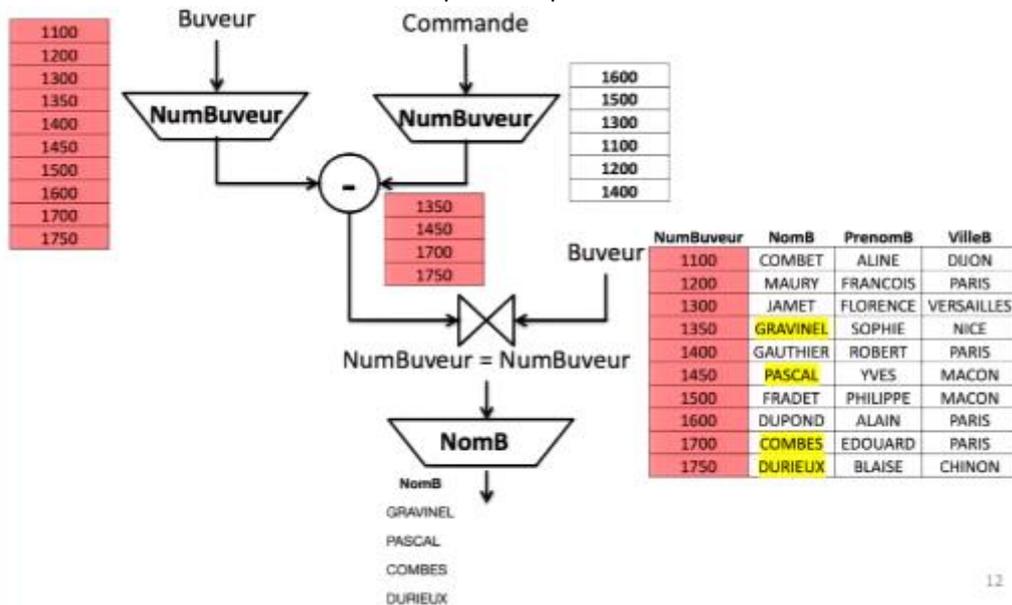
SQL : Différence

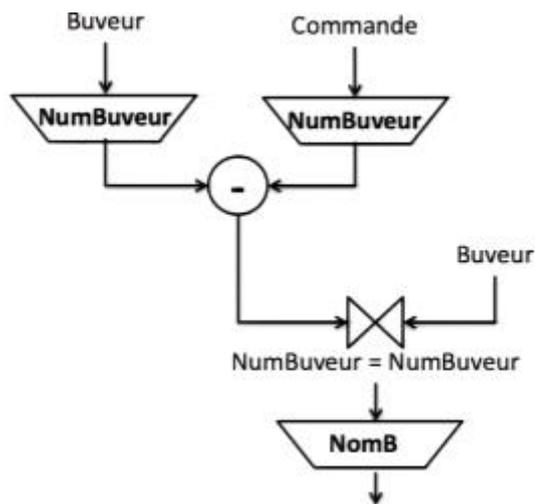
```
SELECT NumBuveur
FROM Buveur B
WHERE B.NumBuveur NOT IN
  (SELECT C.NumBuveur
   FROM Commande C);
```

OU

```
SELECT NumBuveur
FROM Buveur B
WHERE NOT EXISTS
  (SELECT *
   FROM Commande C
   WHERE B.NumBuveur = C.NumBuveur);
```

- Quel est le nom des buveurs qui n'ont pas fait aucune commande ?





```

SELECT NomB
FROM Buveur B
WHERE B.NumBuveur NOT IN
      (SELECT C.NumBuveur
       FROM Commande);

```

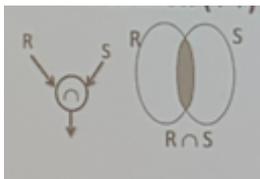
OU

```

SELECT NomB
FROM Buveur B
WHERE NOT EXISTS
      (SELECT *
       WHERE B. NumBuveur = C. NumBuveur);

```

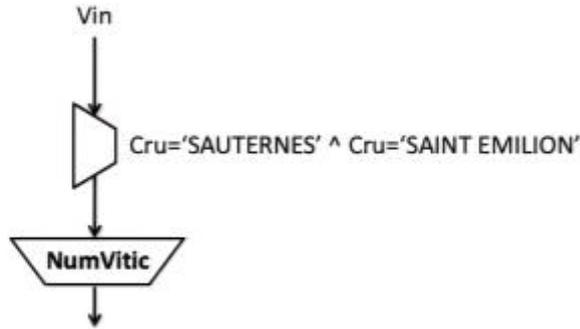
Algèbre : Intersection (\cap)



- Quels sont les numéros des viticulteurs qui produisent du vin de cru SAUTERNES et du vin de cru SAINT EMILION ?

NumVin	Cru	Millesime	Region	NumVitic
80	SAUTERNES	1982	BORDEAUX	18
84	PAUILLAC	1982	BORDEAUX	21
85	CHINON	1983	LOIRE	5
86	CHINON	1978	LOIRE	4
88	CHAMBERTIN	1980	BOURGOGNE	1
90	MARGAUX	1978	BORDEAUX	22
98	MOULIS	1981	BORDEAUX	23
100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
120	CHINON	1985	LOIRE	4
128	BOURGUEIL	1985	LOIRE	4
140	POMMARD	1976	BOURGOGNE	10
150	MONTHELIE	1978	BOURGOGNE	10
160	BOURGUEIL	1984	LOIRE	3
170	BOUZY	1984	CHAMPAGNE	15
175	MEURSAULT	1980	BOURGOGNE	16
190	SAINT EMILION	1981	BORDEAUX	18

?



Résultat : ensemble vide

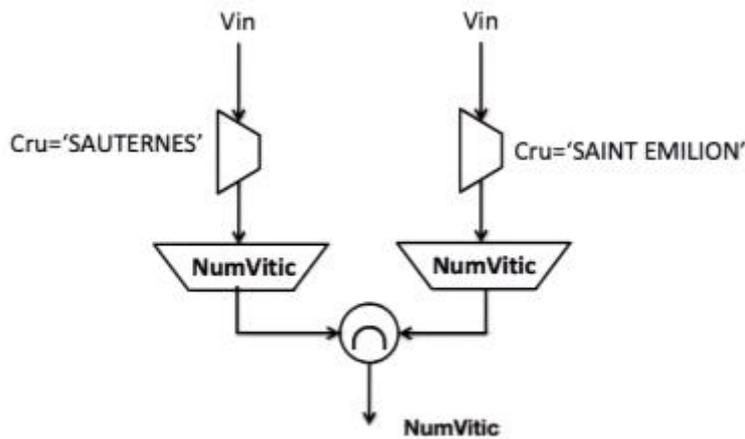
18

Ensemble vide car on demande de prendre deux valeurs distinctes ce qui est impossible.

NumVin	Cru	Millesime	Region	NumVitic
80	SAUTERNES	1982	BORDEAUX	18
84	PAUILLAC	1982	BORDEAUX	21
85	CHINON	1983	LOIRE	5
86	CHINON	1978	LOIRE	4
88	CHAMBERTIN	1980	BOURGOGNE	1
90	MARGAUX	1978	BORDEAUX	22
98	MOULIS	1981	BORDEAUX	23
100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
120	CHINON	1985	LOIRE	4
128	BOURGUEIL	1985	LOIRE	4
140	POMMARD	1976	BOURGOGNE	10
150	MONTHELIE	1978	BOURGOGNE	10
160	BOURGUEIL	1984	LOIRE	3
170	BOUZY	1984	CHAMPAGNE	15
175	MEURSAULT	1980	BOURGOGNE	16
190	SAINT EMILION	1981	BORDEAUX	18



20



SQL: Intersection

SELECT V1.NumVitic

FROM Vin V1

WHERE V1. Cru = 'SAUTERNES'

AND V1.NumVitic IN

(SELECT V2. NumVitic

FROM VIN V2

WHERE V2.Cru = 'SAINT EMILION');

OU

SELECT V1.NumVitic

FROM Vin V1

WHERE V1.Cru = 'SAUTERNES'

AND EXISTS

(**SELECT** *

FROM Vin V2

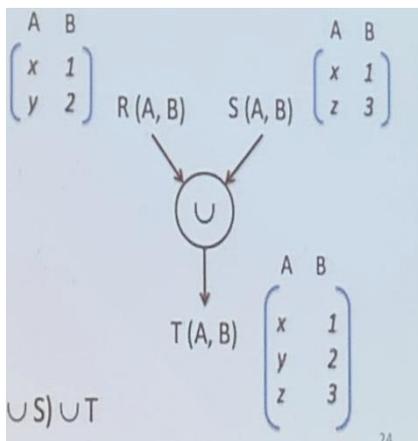
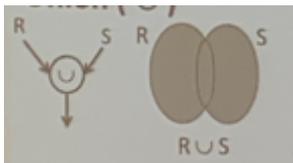
WHERE V2.Cru = 'SAINT EMILION'

AND V1. NumVitic = V2.NumVitic);

Algèbre : Union (U)

L'**union** de 2 relations **R** et **S** de même schéma est une **relation T** de même schéma contenant l'**ensemble des n-uplets** des 2 relations.

On peut souvent s'en sortir avec une restriction



Propriétés :

- Commutative : $R \cup S = S \cup R$
- Associative : $R \cup (S \cup T) = (R \cup S) \cup T$

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

même schéma

Viticulteur1

NumVitic	NomV	PrenomV	VilleV
1	MOULIN	JEAN	MACON
3	MORIN	JULIES	BENAS
4	VALLEE	CLAUDE	SANT NICOLAS
5	COULIS DUTHEL		CHINON
10	PARENT	PIERRE	POMMARD
15	BARANCOURT		TOURS SUR MARNE
16	JADOT	LOUIS	BEAUNE

Viticulteur2

NumVitic	NomV	PrenomV	VilleV
10	PARENT	PIERRE	POMMARD
15	BARANCOURT		TOURS SUR MARNE
16	JADOT	LOUIS	BEAUNE
18	MIEDEVILLE	CHARLES	PREIGNAC
20	TISSOT	JACQUES	ARRIOS
21	LYNCH BAGES		PAULILAC
22	CHATEAU MONBRISON		MARIGALX
23	CHATEAU MOULIN A VENT		CASTELNAU DE MEDOC

Schéma identique à Viticulteur1 et à Viticulteur2

même schéma

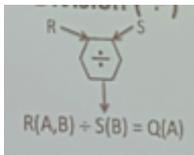
Viticulteur1 Viticulteur2

NumVitic NomV PrenomV VilleV

NumVitic	NomV	PrenomV	VilleV
1	MOULIN	JEAN	MACON
3	MORIN	JULIES	BENAS
4	VALLEE	CLAUDE	SANT NICOLAS
5	COULIS DUTHEL		CHINON
10	PARENT	PIERRE	POMMARD
15	BARANCOURT		TOURS SUR MARNE
16	JADOT	LOUIS	BEAUNE
18	MIEDEVILLE	CHARLES	PREIGNAC
20	TISSOT	JACQUES	ARRIOS
21	LYNCH BAGES		PAULILAC
22	CHATEAU MONBRISON		MARIGALX
23	CHATEAU MOULIN A VENT		CASTELNAU DE MEDOC

ous les n-uplets de Viticulteur1 et de Viticulteur2
n-uplets \in Viticulteur1 ou \in Viticulteur2

Algèbre : Division



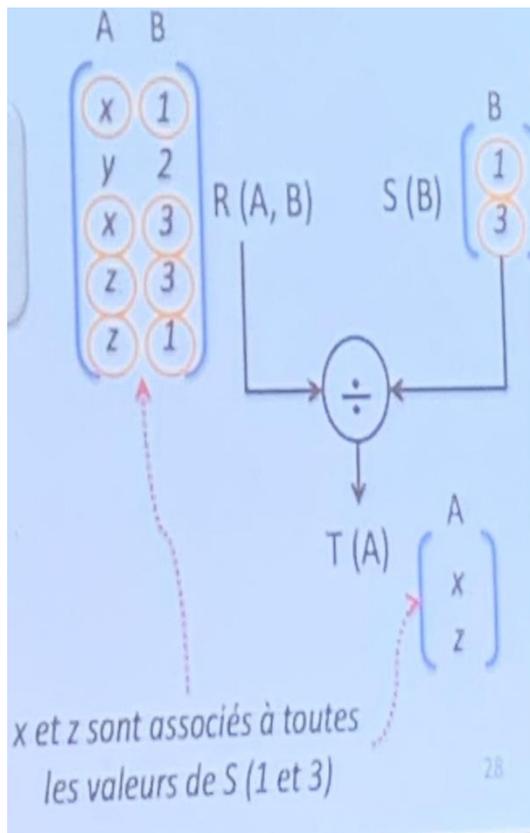
$R(A,B) \div S(B) = D(A)$ avec:

- **N-uplets de R** dont les **valeurs de A** sont associées à toutes les valeurs de **B en S**

« Quelles sont les A de la relation R qui sont associés à tous les B de la relation S ? »

Propriétés :

- NON Commutative : $R S \neq S R$



Exemple :

- Trouver les noms des buveurs qui ont goûté tous les vins



∃ dans Commande 1 n-uplet pour chaque vin dans Vin pour les buveurs que l'on recherche



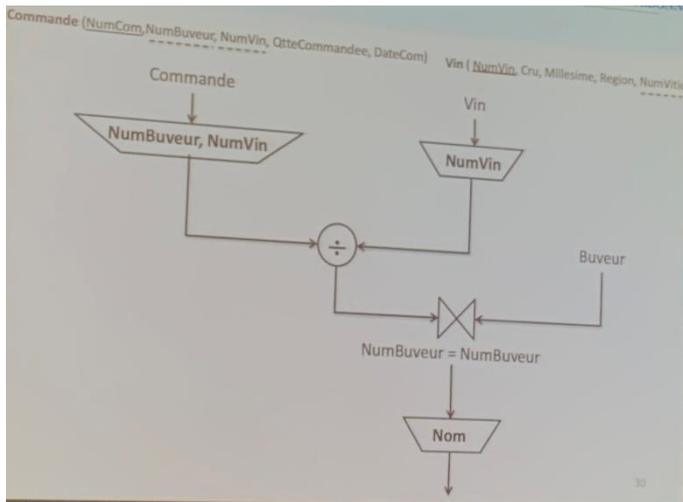
Division de commande par Vin

Commande (NumBuveur, NumVin)

÷

Vin (NumVin)

- 1) Commande : Projection sur NumBuveur et NumVin
- 2) Vin : Projection sur NumVin
- 3) Division
- 4) Jointure avec Buveur
- 5) Projection sur NumBuveur



SQL : Division

- Pas d'opérateur propre de division en SQL
- Utilisation de l'opérateur NOT EXISTS

Nom des buveurs qui ont commandé tous les vins

- ➔ Nom des buveurs pour lesquels il n'existe pas de vin qu'ils n'ont pas commandé
- ➔ **Nom des buveurs pour lesquelles il n'existe pas de Vin pour lequel il n'existe pas de Commande pour ce buveur et pour ce vin**

- Nom des buveurs qui ont commandé tous les vins

```

SELECT B.NomB
FROM Buveur B
WHERE NOT EXISTS
  ( SELECT V.*
    FROM Vin V
    WHERE NOT EXISTS
      (SELECT C.*
       FROM Commande C
       WHERE C.NumBuveur = B.NumBuveur
         AND C.NumVin = V.NumVin
      )
    )
  )
  
```

Nom des **buveurs** pour lesquels il **n'existe pas de Vin** pour lequel il **n'existe pas** une **Commande** pour ce **Buveur**

Quelles opérations ?

1.

INGENIEUR	
Num_empl	Nom_empl
3	Jojo
21	Sophie
15	Loulou
56	Babette

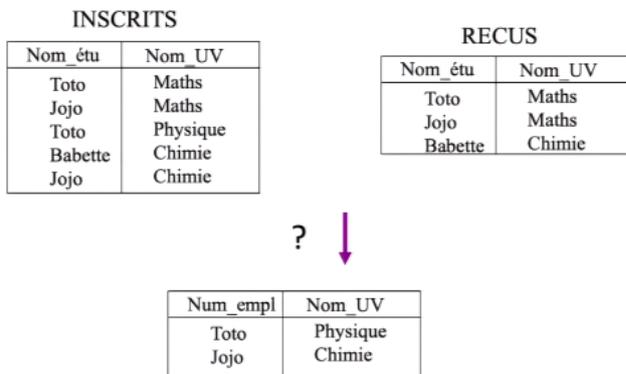
CHEF DE SERVICE	
Num_empl	Nom_empl
3	Jojo
15	Loulou
28	Riri

?

Num_empl	Nom_empl
3	Jojo
15	Loulou

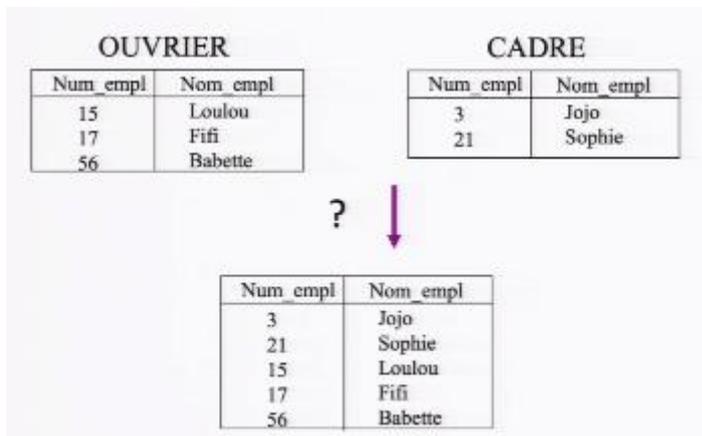
→ intersection

2.



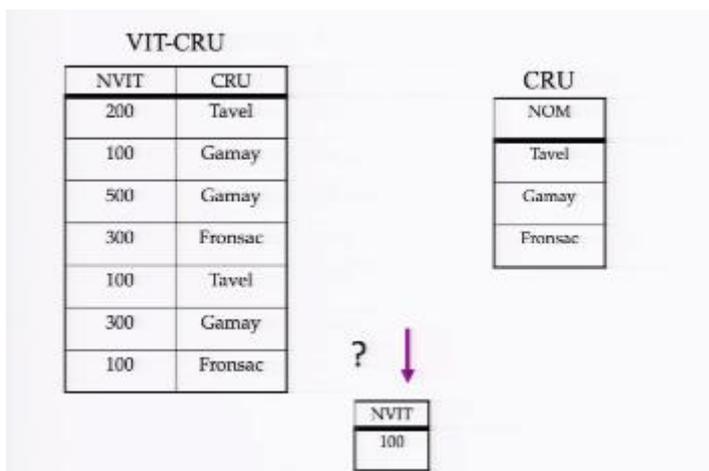
→ Différence

3.



→ Union

4.



→ Division

Base de données Buveur

Algèbre : identification de min/max

Quel est le nom du buveur qui a fait la commande maximale ?

- Identifier d'abord le numéro du buveur ayant effectué la commande la plus importante (en 1 fois, on ne cumule pas les quantités)
- Récupérer le nom de ce(s) buveur(s)

Commande la plus importante (avec qtteCommandee max)

= commande dont qtteCommandee TOUJOURS >= qtteCommandee des autres commandes

= toutes les commandes – commande dont qtteCommandee < QtteCommandee des autres commandes

Différence + thème – jointure avec inégalité stricte

Buveur			
NumBuveur	NomB	PrenomB	VilleB
1100	COMBET	ALINE	DIJON
1200	MAURY	FRANCOIS	PARIS
1300	JAMET	FLORENCE	VERSAILLES
1350	GRAVINEL	SOPHIE	NICE
1400	GAUTHIER	ROBERT	PARIS
1450	PASCAL	YVES	MACON
1500	FRADET	PHILIPPE	MACON
1600	DUPOND	ALAIN	PARIS
1700	COMBES	EDOUARD	PARIS
1750	DURIEUX	BLAISE	CHINON

Viticulteur			
NumVitic	NomV	PrenomV	VilleV
1	MOULIN	JEAN	MACON
3	MORIN	JULES	BENAIS
4	VALLEE	CLAUDE	SAINTE NICOLAS
5	COULIS DUTHEIL		CHINON
10	PARENT	PIERRE	POMMARD
15	BARANCOURT		TOURS SUR MARNE
16	JADOT	LOUIS	BEAUNE
18	MEDEVILLE	CHARLES	PREIGNAC
20	TISSOT	JACQUES	ARBOIS
21	LYNCH-BAGES		PAUILLAC
22	CHATEAU MONBRISON		MARGAUX
23	CHATEAU MOULIN A VENT		CASTELNAU DE MEDOC

Commande				
NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1	1600	90	6	10/06/87
2	1500	100	18	20/11/87
3	1500	88	24	10/09/87
4	1600	175	6	12/02/87
5	1300	140	3	18/03/87
6	1100	140	8	24/12/87
7	1200	88	3	24/12/87
8	1400	140	12	10/12/88
11	1200	140	12	12/12/89
12	1100	120	6	13/12/89

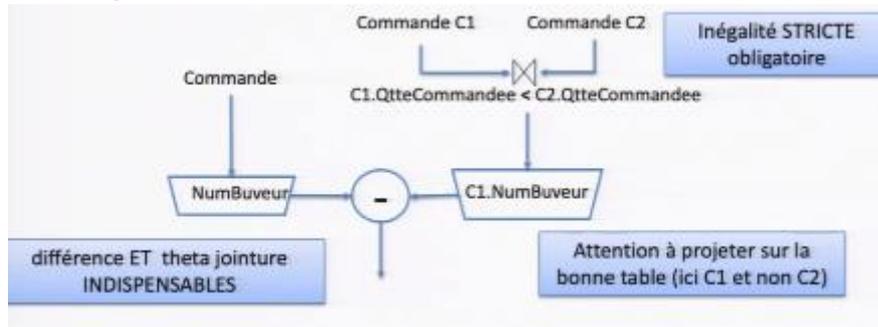
Vin				
NumVin	Cru	Millesime	Region	NumVitic
80	SALTERNES	1982	BORDEAUX	18
84	PAUILLAC	1982	BORDEAUX	21
85	CHINON	1983	LOIRE	5
86	CHINON	1978	LOIRE	4
88	CHAMBERTIN	1980	BOURGOGNE	1
90	MARGAUX	1978	BORDEAUX	22
98	MOULIS	1981	BORDEAUX	23
100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
120	CHINON	1985	LOIRE	4
128	BOURGUEIL	1985	LOIRE	4
140	POMMARD	1976	BOURGOGNE	10
150	MONTHELIE	1978	BOURGOGNE	10
160	BOURGUEIL	1984	LOIRE	3
170	BOUZY	1984	CHAMPAGNE	15
175	MEURSAULT	1980	BOURGOGNE	16
190	SAINT EMILION	1981	BORDEAUX	18

Livraison			
NumCom	QtteLivree	DateLiv	
2	12	30/12/87	
1	5	15/07/87	
2	5	27/12/87	
3	12	20/11/87	
4	5	20/12/87	
6	8	28/12/87	
7	3	30/12/87	

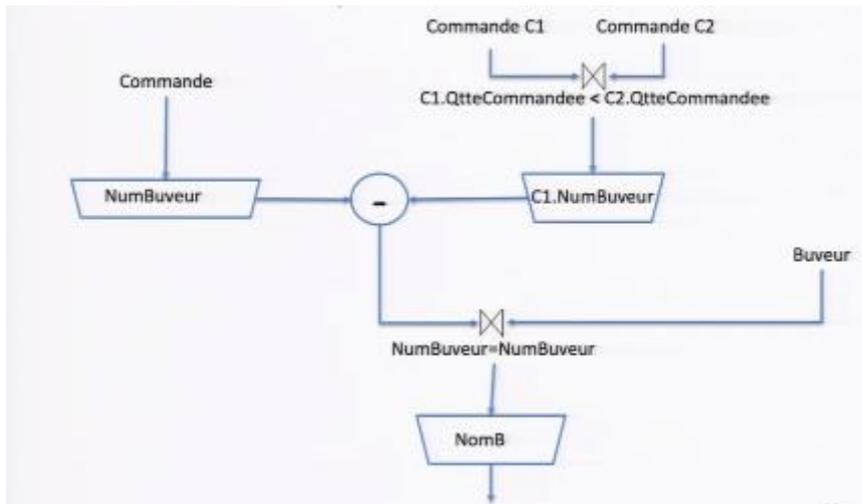
Quel est le numéro du buveur ayant effectué la commande maximale ?

= tous les buveurs ayant fait des commandes – buveurs avec commandes dont qtteCommandee < QtteCommandee des autres commandes

! \ obligé de mettre les alias



Quel est le nom du buveur qui a effectué la commande maximale ?



SQL : identification de min/max

NE SURTOUT PAS S'INSPIRER DE LA SOLUTION EN ALGÈBRE !

Toute tentative de « traduction » de l'algèbre avec une différence et une thêta jointure risque d'être fautive

Fonction de calcul : s'applique sur un ensemble de n-uplets et renvoie une valeur unique

- **MAX (attribut)** : maximum des valeurs prises par un attribut (entier, réel, chaîne de caractères ou date)
- **MIN (attribut)** : minimum des valeurs prises par un attribut (entier, réel, chaîne de caractères ou date)

Quelles est la quantité commandée maximale ?

Reformuler la requête en faisant apparaître le mot max (ou min)

On veut afficher la valeur maximale de QtteCommandee de la table Commande

Fonction MAX indispensable

```
SELECT MAX(QtteCommandee)
FROM Commande ;
```

Fonction MAX dans SELECT, car c'est le résultat du calcul que l'on veut afficher comme résultat

MAX(QtteCommandee)
24

Quel est le nom du buveur qui a fait la commande maximale ?

Reformuler la requête en faisant apparaître le mot max (ou min)

On cherche le nom du buveur dont la QtteCommandee est égale au max des QtteCommandee

Fonction MAX indispensable

```

SELECT NomB
FROM Buveur B, Commande C
WHERE B. NumBuveur = C.NumBuveur
AND QtteCommandee = ( SELECT MAX ( QtteCommandee)
                      FROM Commande);

```

Fonction MAX dans le WHERE
Il faut impérativement ouvrir un 2^{ème} select

NomB
FRADET

Exercice : Algèbre Relationnelle et SQL

- Base de données « CoursesBateaux »
 - Bateau (Nbat, NomBat, Sponsor)
 - Course (Ncomp, NomComp, DateComp, PrixComp)
 - Résultat (Nbat, Ncomp, Score)
- Requêtes en algèbre et en SQL pour afficher :
 - 1) Le nom des bateaux sponsorisés par « AGF »
 - 2) Le numéro des bateaux qui ont été 1^e ou 2^e dans une course
 - 3) Le nom des bateaux qui ont été 1^e ou 2^e dans une course
 - 4) Le nom des bateaux qui ont le même sponsor que le bateau « VDP »
 - 5) Le numéro des bateaux qui ont toujours été classés avant la 4^e place
Le nom des bateaux qui ont toujours été classés avant la 4^e place

- 1) Lister les noms des bateaux sponsorisés par AGF

1) Lister les **noms des bateaux sponsorisés par AGF**

Résultat : NomBat ← Condition : sponsor="AGF"

Bateau (Nbat, **NomBat**, *Sponsor*) ←

```

SELECT NomBat
FROM Bateau
WHERE Sponsor = 'AGF' ;

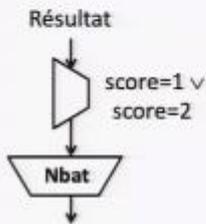
```

- 2) Lister les n° des bateaux qui ont été 1^e ou 2^e à une course
(SCORE = 1)

2) Lister les n° des bateaux qui ont été 1^{ers} ou 2^{emes} à une course

Résultat : Nbat ← Condition : score = 1 ∨ score=2
 Résultat (Nbat, Ncomp, Score)

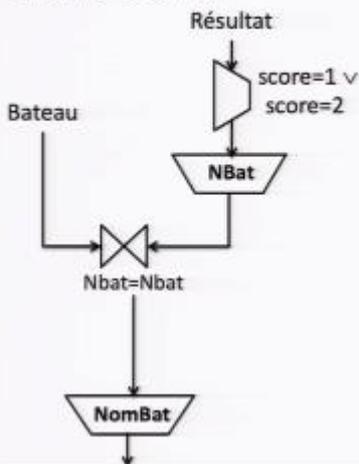
∨ OU
 ∧ ET



```
SELECT Nbat
FROM Resultat
WHERE score=1 OR score=2 ;
```

3) Lister les noms des bateaux qui ont été 1^{er} ou 2^e à une course (SCORE = 1)

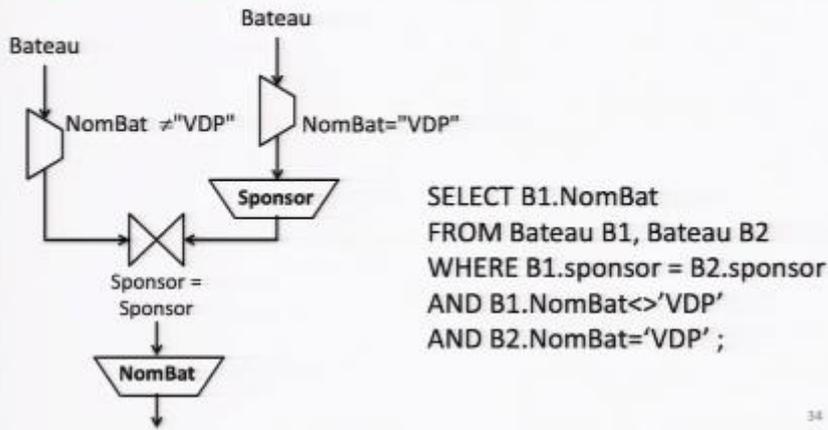
3) Lister les noms des bateaux qui ont été 1^{ers} ou 2^{emes} à une course



```
SELECT NomBat
FROM Bateau B, Resultat R
WHERE B.Nbat = R.Nbat
AND (score=1 OR score=2) ;
```

4) Lister les bateaux qui ont le même sponsor que le bateau VDP
 a. Trouver le sponsor de VDP
 b. Trouver les autres bateaux
 c. Combiner les résultats

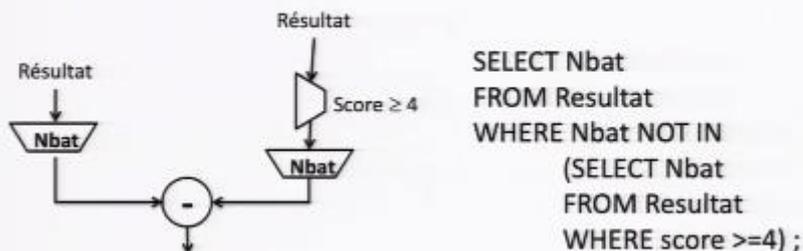
4) Lister les noms des bateaux qui ont le même sponsor que le bateau VDP



- 5) Lister les n° des bateaux qui ont toujours été classés avant la 4^e place
Toujours classés < 4
Jamais eu un score >= 4

Ceux qui ont participé à une course ————— Ceux qui ont été classés après 4^e (score >=4)

5) Lister les n° des bateaux qui ont toujours été classés avant la 4^e place



```

SELECT Nbat
FROM Resultat R1
WHERE NOT EXISTS
(SELECT *
FROM Resultat R2
WHERE score >=4
AND R1.Nbat = R2.Nbat) ;
    
```

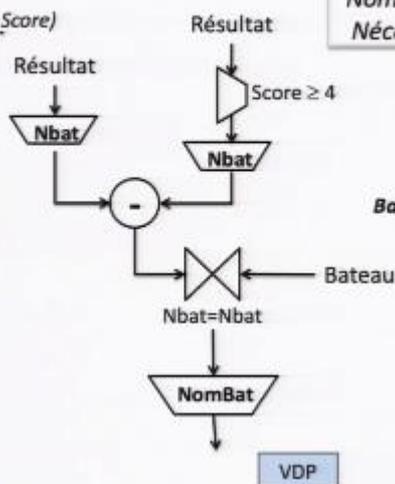
Lister les noms des bateaux qui ont toujours été classés avant la 4^e place

5_{bis}) Lister les noms de ces bateaux

Résultat (Nbat, Ncomp, Score)

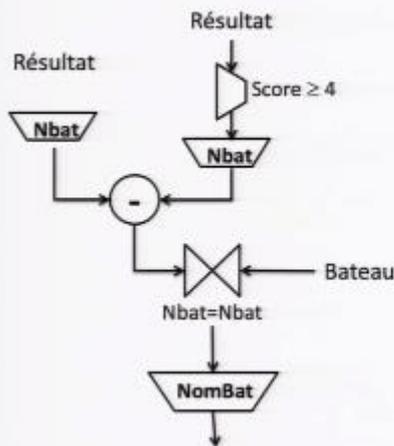
100	10	1
200	10	2
300	10	3
300	20	5
200	20	4
100	20	2

Noms sur la table **Bateau**
Nécessité d'une **Jointure**



39

5_{bis}) Lister les noms de ces bateaux



```
SELECT NomBat
FROM Bateau B, Resultat R
WHERE B.Nbat = R.Nbat
AND B.Nbat NOT IN
(SELECT Nbat
FROM Resultat
WHERE score >=4);
```

40

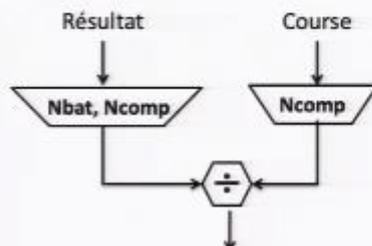
6) Le numéro des bateaux qui ont participé à toutes les courses

6) Lister les n° bateaux qui ont participé à toutes les courses

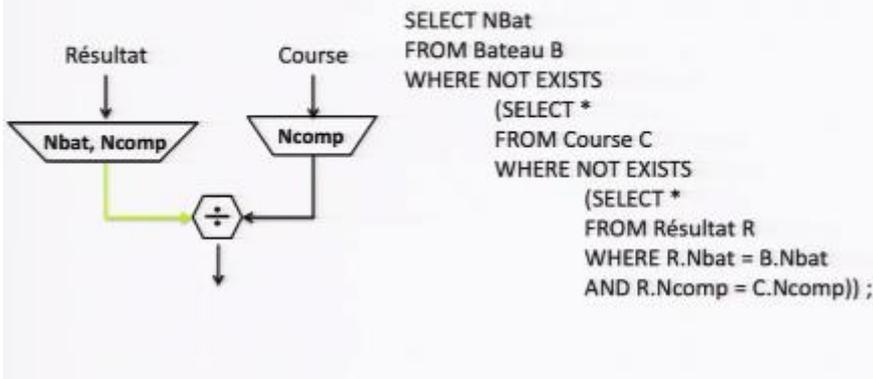
Les bateaux qui ont un résultat pour chaque course

Les bateaux associés à TOUTES les courses

Division

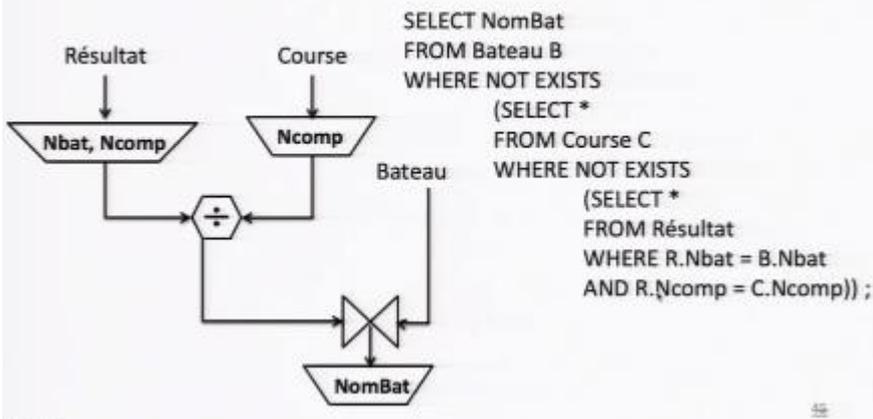


6) Lister les n° des bateaux qui ont participé à toutes les courses



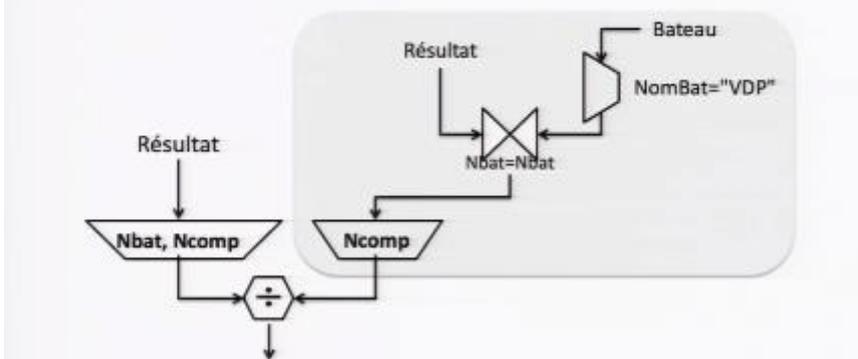
Le nom des bateaux qui ont participé à toutes les courses

6) Lister les noms des bateaux qui ont participé à toutes les courses



- 7) Le nom des courses offrant un prix supérieur à 20K euros auxquelles le bateau VDP a participé
- 8) Le numéro des bateaux qui ont participé à toutes les courses auxquelles le bateau VDP a participé

8) Lister les n° des bateaux qui ont participé à toutes les courses auxquelles le bateau VDP a participé

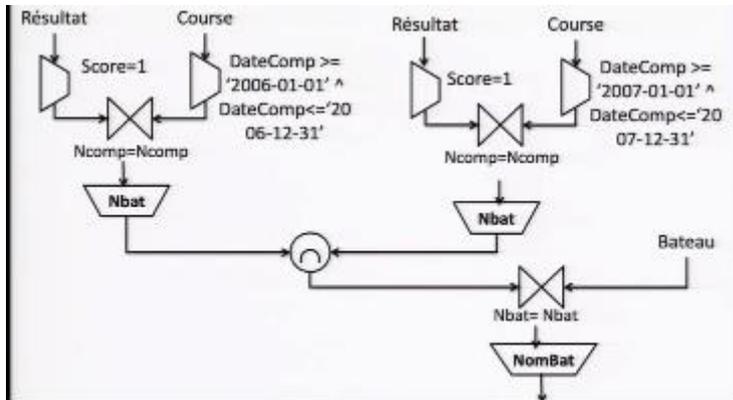


```

SELECT Nbat
FROM Bateau B1
WHERE NOT EXISTS
  (SELECT *
   FROM Résultat R1, Bateau B2
   WHERE R1.Nbat=B2.Nbat
   AND B2.NomBat='VDP '
   AND NOT EXISTS
    (SELECT *
     FROM Résultat R2
     WHERE R2.Nbat = B1.Nbat
     AND R2.Ncomp = R1.Ncomp));

```

- 9) Le nom des bateaux qui ont classés 1^{er} au moins une fois en 2006 et au moins une fois en 2007
- Trouver les bateaux classés en 1^{er} en 2006
 - Trouver les bateaux classés en 1^{er} en 2007
 - Trouver les bateaux qui participent à a et b
 - Intersection de de (a) et (b)
 - Jointure avec la table Bateau
 - Projection sur NomBat



```

SELECT NomBat
FROM Bateau B, Resultat R1, Course C1
WHERE B.Nbat = R1.Nbat
AND C1.Ncomp=R1.Ncomp
AND score = 1
AND DateComp BETWEEN '2006-01-01' AND '2006-12-31'
AND B.Nbat IN
  (SELECT Nbat
   FROM Resultat R2, Course C2
   WHERE R2.Ncomp = C2.Ncomp
   AND score=1
   AND DateComp BETWEEN '2007-01-01' AND '2007-12-31' );

```

```

SELECT NomBat
FROM Bateau B, Resultat R1, Course C1
WHERE B.Nbat = R1.Nbat
AND C1.Ncomp=R1.Ncomp
AND score = 1
AND DateComp BETWEEN '2006-01-01' AND '2006-12-31'
AND EXISTS
  (SELECT *
   FROM Resultat R2, Course C2
   WHERE R2.Ncomp = C2.Ncomp
   AND score=1
   AND DateComp BETWEEN '2007-01-01' AND '2007-12-31'
   AND B.Nbat = R2.Nbat );

```

10) Le numéro de la course la plus récente

Conseil : commencer par SQL et ne surtout pas se laisser influencer par la solution en algèbre, qui n'a rien à voir (pour toute question MIN/MAX)

```

SELECT Ncomp
FROM Course
WHERE DateComp =
  (SELECT MAX(DateComp)
   From Course);

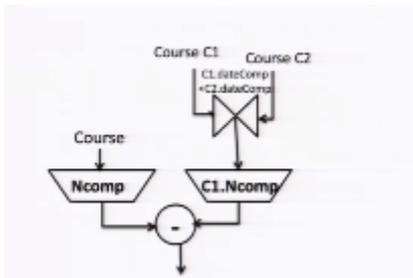
```

= question Min – Max

Ncomp dont la dateComp est toujours >= à la dateComp des autres Ncomp

On fait une différence :

On prend tout les Ncomp et on enlève les Ncomp dont la dateComp est < (strictement inférieure) aux autres dateComp



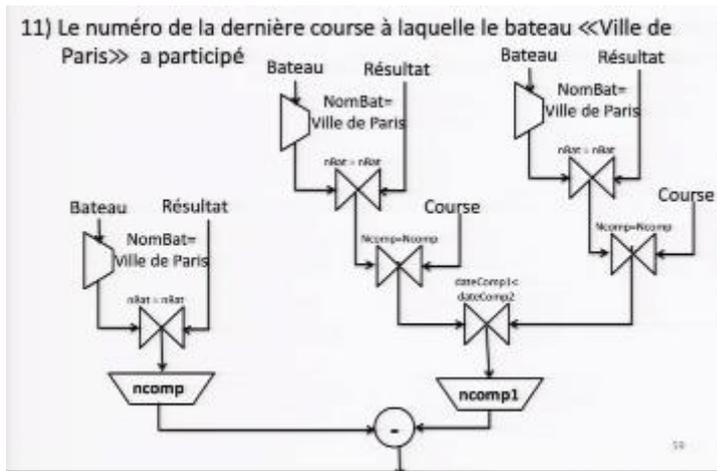
11) Le numéro de la dernière course à laquelle la bateau « Ville de Paris » a participé

Idem que la question précédente, mais au lieu de considérer tous les Ncomp, on conserve uniquement les Ncomp associés au bateau Ville de Paris

```

SELECT Ncomp
FROM Resultat R1, Bateau B1, Course C1
WHERE R1.Ncomp = C1.Ncomp
AND B1.Nbat=R1.Nbat
AND NomBat='Ville de Paris'
AND DateComp =
  (SELECT MAX(DateComp)
   From Resultat R2, Bateau B2, Course C2
   WHERE R2.Ncomp = C2.Ncomp
   AND B2.Nbat=R2.Nbat
   AND NomBat='Ville de Paris' );

```



Tri des résultats : ORDER BY

!/ Pas d'équivalence en Algèbre

- **ORDER BY attr1, attr2...** permet de trier le résultat selon les attributs précisés.
- **ORDER BY attr1 ASC/ DESC** permet de spécifier le sens du tri (ascendant ou descendant). Si non précisé = ascendant

Infos sur les viticulteurs, tirés par leur nom (ordre alphabétique)

NumVitic	NomV	PrenomV	VilleV
15	BARANCOURT		TOURS SUR MARNE
22	CHATEAU MONBRISON		MARGAUX
23	CHATEAU MOULIN A VENT		CASTELNAU DE MEDOC
5	COULIS DUTHEIL		CHINON
16	JADOT	LOUIS	BEAUNE
21	LYNCH-BAGES		PAUILLAC
18	MEDEVILLE	CHARLES	PREIGNAC
3	MORIN	JULES	BENAI
1	MOULIN	JEAN	MACON
10	PARENT	PIERRE	POMMARD
20	TISSOT	JACQUES	ARBOIS
4	VALLEE	CLAUDE	SAINT NICOLAS

SELECT* FROM Viticulteur

ORDER BY NomV ASC ;

Infos sur les vins, triés par millésime décroissant

SELECT* FROM Vin

ORDER BY millésime DESC ;

➔ ORDER BY toujours à la fin

Fonctions de calcul

!/ Pas d'équivalence en Algèbre

- Fonction de calcul : s'applique sur un ensemble de n-uplet et renvoie une valeur unique
- Calcul sur les valeurs prises par un attributs

- **AVG (DISTINCT/ ALL attribut)** : Moyenne des valeurs prises par un attributs (entier ou réel)
 - Avec **DISTINCT** : sans les doublons
 - Avec **ALL** : avec les doublons
- **SUM (DISTINCT / ALL attribut)** : somme des valeurs prises par un attribut (entier ou réel)
- **MAX (attribut)** : maximum des valeurs prises par un attribut (entier, réel, caractères ou date)
- **MIN (attributs)** : minimum des valeurs prises par un attribut (entier, réel, caractères ou date)
- Comptage des n-uplets
 - **COUNT (*)**, **COUNT (ALL/ DISTINCT attribut)**

Exemples :

Quantité totale de bouteilles commandées par le buveur n° 1600

Commande				
NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1	1600	90	6	10/06/87
2	1500	100	18	20/11/87
3	1500	88	24	10/09/87
4	1600	175	6	12/02/87
5	1300	140	3	18/03/87
6	1100	140	8	24/12/87
7	1200	88	3	24/12/87
8	1400	140	12	10/12/88
11	1200	140	12	12/12/89
12	1100	120	6	13/12/89

```
SELECT SUM ( QtteCommandee)
FROM Commande
WHERE NumBuveur = 1600 ;
```

Quantité maximale commandée par le buveur n° 1600

```
SELECT MAX ( QtteCommandee)
FROM Commande
WHERE NumBuveur = 1600;
```

• Comptage

/!\ Pas d'équivalence en Algèbre

- **COUNT** compte le nombre de n-uplet
 - **COUNT (*)**, **COUNT (ALL/ DISTINCT attribut)**
 - Dans le cas de *: y compris les valeurs nulles
 - Avec **DISTINCT** : sans les doublons
 - Avec **ALL** : avec les doublons

Nombre de vins différents commandés

Commande	NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
	1	1600	90	6	10/06/87
	2	1500	100	18	20/11/87
	3	1500	88	24	10/09/87
	4	1600	175	6	12/02/87
	5	1300	140	3	18/03/87
	6	1100	140	8	24/12/87
	7	1200	88	3	24/12/87
	8	1400	140	12	10/12/88
	11	1200	140	12	12/12/89
	12	1100	120	6	13/12/89

SELECT COUNT (DISTINCT NumVin)
FROM Commande ;

Remarque : les fonctions de calcul peuvent être dans la clause **SELECT**

Quantité moyenne commandée

Commande	NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
	1	1600	90	6	10/06/87
	2	1500	100	18	20/11/87
	3	1500	88	24	10/09/87
	4	1600	175	6	12/02/87
	5	1300	140	3	18/03/87
	6	1100	140	8	24/12/87
	7	1200	88	3	24/12/87
	8	1400	140	12	10/12/88
	11	1200	140	12	12/12/89
	12	1100	120	6	13/12/89

SELECT AVG (QtteCommandee)
FROM Commande ;

Remarque : les fonctions de calcul peuvent aussi être utilisé dans la clause **WHERE**

Numéro des buveurs ayant fait des commandes supérieures à la moyenne

Commande	NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
	1	1600	90	6	10/06/87
	2	1500	100	18	20/11/87
	3	1500	88	24	10/09/87
	4	1600	175	6	12/02/87
	5	1300	140	3	18/03/87
	6	1100	140	8	24/12/87
	7	1200	88	3	24/12/87
	8	1400	140	12	10/12/88
	11	1200	140	12	12/12/89
	12	1100	120	6	13/12/89

AVG(QtteCommandee)
9.8000

SELECT DISTINCT NumBuveur
FROM Commande
WHERE QtteCommandee >=
(SELECT AVG (QtteCommandee)
FROM Commande) ;

NumBuveur
1200
1400
1500

Numéro des buveurs ayant TOUJOURS fait des commandes supérieures à la moyenne

(pas demandé en exams)

Commande

NumCom	NumBuveur	NumVin	QtteCommandee	DateCom
1	1600	90	6	10/06/87
2	1500	100	18	20/11/87
3	1500	88	24	10/09/87
4	1600	175	6	12/02/87
5	1300	140	3	18/03/87
6	1100	140	8	24/12/87
7	1200	88	3	24/12/87
8	1400	140	12	10/12/88
11	1200	140	12	12/12/89
12	1100	120	6	13/12/89

AVG(QtteCommandee)
9.8000

```

SELECT DISTINCT NumBuveur
FROM Commande
WHERE numBuveur NOT IN
    (SELECT NumBuveur
     FROM Commande
     WHERE QtteCommandee <=
        ( SELECT ( AVG(QtteCommandee)
          FROM Commande)
        );

```

NumBuveur
1400
1500

Partitionnement

!/ Pas d'équivalence en Algèbre

Objectif : partitionnement horizontal d'une relation en sous-relations selon 1 ou plusieurs attributs

Vin					Partitionnement par Viticulteur				
NumVin	Cru	Millesime	Region	NumVitic	NumVin	Cru	Millesime	Region	NumVitic
80	SAUTERNES	1982	BORDEAUX	18	88	CHAMBERTIN	1980	BOURGOGNE	1
84	PAUILLAC	1982	BORDEAUX	21	160	BOURGUEIL	1984	LOIRE	3
85	CHINON	1983	LOIRE	5	86	CHINON	1978	LOIRE	4
86	CHINON	1978	LOIRE	4	120	CHINON	1985	LOIRE	4
88	CHAMBERTIN	1980	BOURGOGNE	1	128	BOURGUEIL	1985	LOIRE	4
90	MARGAUX	1978	BORDEAUX	22	85	CHINON	1983	LOIRE	5
98	MOULIS	1981	BORDEAUX	23	140	POMMARD	1976	BOURGOGNE	10
100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15	150	MONTHELIE	1978	BOURGOGNE	10
120	CHINON	1985	LOIRE	4	100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
128	BOURGUEIL	1985	LOIRE	4	170	BOUZY	1984	CHAMPAGNE	15
140	POMMARD	1976	BOURGOGNE	10	175	MEURSAULT	1980	BOURGOGNE	16
150	MONTHELIE	1978	BOURGOGNE	10	80	SAUTERNES	1982	BORDEAUX	18
160	BOURGUEIL	1984	LOIRE	3	190	SAINT EMILION	1981	BORDEAUX	18
170	BOUZY	1984	CHAMPAGNE	15	84	PAUILLAC	1982	BORDEAUX	21
175	MEURSAULT	1980	BOURGOGNE	16	90	MARGAUX	1978	BORDEAUX	22
190	SAINT EMILION	1981	BORDEAUX	18	98	MOULIS	1981	BORDEAUX	23

GROUP BY attr1, attr2... permet le partitionnement horizontal d'une relation en sous-relations en fonction des attributs

ATTENTION : On retrouve dans le résultat une **ligne par sous-relation**

Nombre de vins produits par région

```
SELECT*
FROM Vin
GROUP BY region ;
```

SELECT *
FROM Vin
GROUP BY region ;

NumVin	Cru	Millesime	Region	NumVitic
80	SAUTERNES	1982	BORDEAUX	18
88	CHAMBERTIN	1980	BOURGOGNE	1
100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
85	CHINON	1983	LOIRE	5

Vin					Partitionnement par région				
NumVin	Cru	Millesime	Region	NumVitic	NumVin	Cru	Millesime	Region	NumVitic
80	SAUTERNES	1982	BORDEAUX	18	80	SAUTERNES	1982	BORDEAUX	18
84	PAUILLAC	1982	BORDEAUX	21	84	PAUILLAC	1982	BORDEAUX	21
85	CHINON	1983	LOIRE	5	90	MARGAUX	1978	BORDEAUX	22
86	CHINON	1978	LOIRE	4	98	MOULIS	1981	BORDEAUX	23
88	CHAMBERTIN	1980	BOURGOGNE	1	190	SAINT EMILION	1981	BORDEAUX	18
90	MARGAUX	1978	BORDEAUX	22	88	CHAMBERTIN	1980	BOURGOGNE	1
98	MOULIS	1981	BORDEAUX	23	140	POMMARD	1976	BOURGOGNE	10
100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15	150	MONTHELIE	1978	BOURGOGNE	10
120	CHINON	1985	LOIRE	4	175	MEURSAULT	1980	BOURGOGNE	16
128	BOURGUEIL	1985	LOIRE	4	100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
140	POMMARD	1976	BOURGOGNE	10	170	BOUZY	1984	CHAMPAGNE	15
150	MONTHELIE	1978	BOURGOGNE	10	85	CHINON	1983	LOIRE	5
160	BOURGUEIL	1984	LOIRE	3	86	CHINON	1978	LOIRE	4
170	BOUZY	1984	CHAMPAGNE	15	120	CHINON	1985	LOIRE	4
175	MEURSAULT	1980	BOURGOGNE	16	128	BOURGUEIL	1985	LOIRE	4
190	SAINT EMILION	1981	BORDEAUX	18	160	BOURGUEIL	1984	LOIRE	3

Les fonctions **AVG, SUM, MAX, COUNT**, placées dans la clause **SELECT**, s'appliquent à chacune des sous-relations créées par le **GROUP BY**

Nombre de vins produits par région

```
SELECT COUNT(NumVin), region
FROM Vin
GROUP BY region ;
```

COUNT(NumVin)	region
5	BORDEAUX
4	BOURGOGNE
2	CHAMPAGNE
5	LOIRE

- Les attributs présents dans la clause **GROUP BY** sont forcément présents après **SELECT**

HAVING condition permet de poser une **condition** portant **sur les sous-relations** générées par le **GROUP BY**

Numéro des viticulteurs ayant produit au moins 2 vins

Vin					Partitionnement par Viticulteur				
NumVin	Cru	Millesime	Region	NumVitic	NumVin	Cru	Millesime	Region	NumVitic
80	SAUTERNES	1982	BORDEAUX	18	88	CHAMBERTIN	1980	BOURGOGNE	1
84	PAUILLAC	1982	BORDEAUX	21	160	BOURGUEIL	1984	LOIRE	3
85	CHINON	1983	LOIRE	5	86	CHINON	1978	LOIRE	4
86	CHINON	1978	LOIRE	4	120	CHINON	1985	LOIRE	4
88	CHAMBERTIN	1980	BOURGOGNE	1	128	BOURGUEIL	1985	LOIRE	4
90	MARGAUX	1978	BORDEAUX	22	85	CHINON	1983	LOIRE	5
98	MOULIS	1981	BORDEAUX	23	140	POMMARD	1976	BOURGOGNE	10
100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15	150	MONTHELIE	1978	BOURGOGNE	10
120	CHINON	1985	LOIRE	4	100	CHAMPAGNE BRUT	1988	CHAMPAGNE	15
128	BOURGUEIL	1985	LOIRE	4	170	BOUZY	1984	CHAMPAGNE	15
140	POMMARD	1976	BOURGOGNE	10	175	MEURSAULT	1980	BOURGOGNE	16
150	MONTHELIE	1978	BOURGOGNE	10	80	SAUTERNES	1982	BORDEAUX	18
160	BOURGUEIL	1984	LOIRE	3	190	SAINT EMILION	1981	BORDEAUX	18
170	BOUZY	1984	CHAMPAGNE	15	84	PAUILLAC	1982	BORDEAUX	21
175	MEURSAULT	1980	BOURGOGNE	16	90	MARGAUX	1978	BORDEAUX	22
190	SAINT EMILION	1981	BORDEAUX	18	98	MOULIS	1981	BORDEAUX	23

```
SELECT NumVitic
FROM Vin
GROUP BY NumVitic
HAVING COUNT (NumVin) >= 2 ;
```

NumVitic
4
10
15
18

Les sous relations ne vérifiant pas la condition sont écartées résultat

Exercice SQL

- Base de données « CoursesBateaux »
 - Bateau (Nbat, NomBat, Sponsor)
 - Course (Ncomp, NomComp, DateComp, PrixComp)
 - Résultat (Nbat, Ncomp, Score)
- Requête en SQL permettant d'afficher :
 - 12) Nombre de fois que chaque bateau est arrivé premier à une course

```
SELECT COUNT(*), Nbat
FROM Resultat
WHERE score = 1
GROUP BY Nbat ;
```

- 13) Nombre de bateaux par course ayant eu lieu en 2019

```
SELECT COUNT(Nbat), R.NComp
FROM Resultat R, Course C
WHERE R.Ncomp=C.Ncomp AND dateComp
BETWEEN '01/01/2019' AND '31/12/2019'
GROUP BY Ncomp ;
```

- 14) Montant total des prix gagnés par le bateau Ville de Paris

```

SELECT SUM(PrixComp)
FROM Course C, Bateau B, Resultat R
WHERE C.Nbat = R.Nbat
AND R.Ncomp = C.Ncomp
AND score=1
AND NomBat = « Ville de Paris » ;

```

- 15) Noms des courses (et prix associé) auxquelles des bateaux dont le nom commence par « Ville » ont participé, triés par prix décroissant

```

SELECT NomComp, PrixComp
FROM Course C, Bateau B, Resultat R
WHERE C.Nbat = R.Nbat
AND R.Ncomp = C.Ncomp
AND NomBat LIKE « Ville%»
ORDER BY PrixComp DESC ;

```

- 16) Score moyen obtenu pour chaque bateau (uniquement pour ceux qui ont participé à au moins 3 courses)

```

SELECT AVG(score), Nbat
FROM Resultat
GROUP BY Nbat
HAVING COUNT(Ncomp)>=3) ;

```

- 17) Nom des bateaux ayant effectué des courses en 2018, triés par ordre alphabétique

```

SELECT NomBat
FROM Bateau B, Course C, Resultat R
WHERE C.Nbat = R.Nbat
AND R.Ncomp = C.Ncomp
AND DateComp BETWEEN '01/01/2018' AND '31/12/2018'
ORDER BY NomBat ;

```