

# Rappel sur les Bases de Données

---

**Didier DONSEZ**

Université Joseph Fourier

PolyTech'Grenoble –LIG/ADELE


**Didier.Donsez@imag.fr,**

**Didier.Donsez@ieee.org**

Didier Donsez, 1998-2007, Rappel  
Bases de Données

# Concepts de Base

---

- |   |                   |
|---|-------------------|
| ■ <b>Modèle Relationnel</b>   | <b>SQL</b>        |
| ■ <b>Concept de Base</b>  |                   |
| ■ Relation  | Table             |
| ■ Tuple   | Ligne             |
| ■ Attribut  | Colonne           |
| ■ <b>Manipulation</b>   |                   |
| ■ <i>Opérateurs Relationnels</i>  | <i>Ordres</i>     |
| $\cup, -, \Pi, \sigma, \times, \cap, \div,$  | <b>SELECT ...</b> |
| ■ <i>Composition d'Opérateurs</i>   | <b>FROM ...</b>   |
| <b>associativité des jointures, ... WHERE ...</b>   |                   |

# Exemples de Requêtes

- Soit la base constituée par les relations  
**BUVEURS(NB, NOM, VILLE)**  
**VINS(NV, CRU, MILL, REGION, COULEUR)**  
**CONSO(NB, NV, QTE)**

- Mono-Relation

- Noms des buveurs parisiens

$\Pi ( \sigma( \text{Buveurs/ Ville} = \text{' Paris ' } ) / \text{Nom} )$

- Crus des bordeaux rouges de millésime 1994

$\Pi ($

$\sigma( \text{Vins/ Region} = \text{' Bordelais ' } \wedge \text{Mill} = 1994 \wedge$   
 $\text{Couleur} = \text{' Rouge ' } )$

$/ \text{Cru} )$

# La Restriction $\sigma$ et la Projection $\Pi$

Buv	NB	Nom	Ville
	100	Ben	Nice
	101	Ben	Lille
	102	Charlie	Paris
	103	David	Paris

$\sigma(\text{Buveurs/ Ville=' Paris '})$

Rest	NB	Nom	Ville
	102	Charlie	Paris
	103	David	Paris

$\Pi(\text{Buveurs/Ville})$

Proj	Ville
	Nice
	Lille
	Paris

# Exemples de Requêtes

- Soit la base constituée par les relations

BUVEURS(NB, NOM, VILLE)

VINS(NV, CRU, MILL, REGION, COULEUR)

CONSO(NB, NV, QTE)

- Multi-Relations

- Noms des buveurs de vins

$$\Pi ( \sigma( (\text{Buveur} \times \text{Conso}) / \text{Buveurs.NB} = \text{Conso.NB}) / \text{Nom} )$$

$$\Pi ( \succ\langle (\text{Buveur}, \text{Conso} / \text{Buveurs.NB} = \text{Conso.NB}) / \text{Nom} )$$

- Noms des buveurs qui ont consommé un vin rouge

$$\Pi( \succ\langle (\sigma( \text{Vins} / \text{Couleur} = ' \text{Rouge} '),$$

$$\succ\langle (\text{Buveur}, \text{Conso} / \text{Buveurs.NB} = \text{Conso.NB})$$

$$/ \text{Conso.NV} = \text{Vins.NV} )$$

$$/ \text{Nom} )$$

# L'union $\cup$ et l'intersection $\cap$

Buv1	NB	Nom	Ville
	100	Ben	Nice
	101	Ben	Lille

Buv2	NB	Nom	Ville
	101	Ben	Lille
	102	Charlie	Paris
	103	David	Paris

Buveurs1  $\cup$  Buveurs2

Buveurs1  $\cap$  Buveurs2

Union	NB	Nom	Ville
	100	Ben	Nice
	101	Ben	Lille
	102	Charlie	Paris
	103	David	Paris

Inter	NB	Nom	Ville
	101	Ben	Lille

# La Jointure ><

Buv	NB	Nom
	100	Al
	101	Ben
	102	Charlie
	103	David

nées

Conso	NB	Qte
	100	50
	100	25
	100	75
	101	100
	101	50
	102	25

Didier Donsez, 1998-2007, Rappel Bases

NB	Nom	NB	Qte
100	Al	100	50
100	Al	100	25
100	Al	100	75
101	Ben	101	100
101	Ben	101	50
102	Charlie	102	25

# Propriétés de l'algèbre relationnelle

## ■ Associativité des jointures

- Noms des buveurs qui ont consommé un vin rouge

$$\Pi( \begin{array}{l} \bowtie( \quad \sigma( \text{Vins/Couleur}=' \text{Rouge} '), \\ \bowtie( \text{Buveur,Conso/ Buveurs.NB}=\text{Conso.NB} \\ / \text{Conso.NV}=\text{Vins.NV} \\ / \text{Nom} \end{array} )$$

$$\Pi( \begin{array}{l} \bowtie( \text{Buveurs} \\ \bowtie( \sigma( \text{Vins/Couleur}=' \text{Rouge} '), \text{Conso/ Vins.NV}=\text{Conso.NV} \\ / \text{Conso.NB}=\text{Buveurs.NB} \\ / \text{Nom} \end{array} )$$



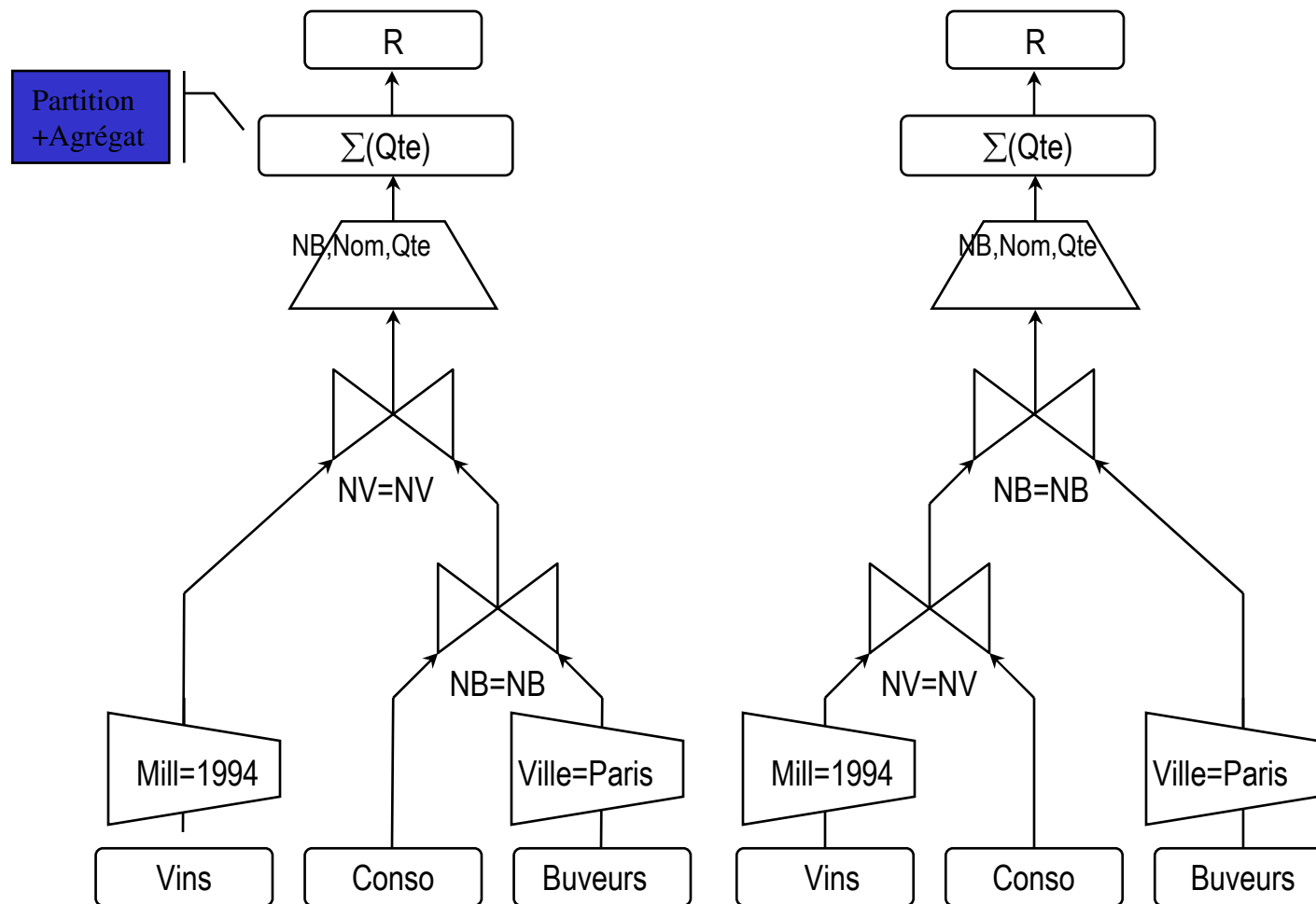
# Propriétés de l'algèbre relationnelle

- Commutativité des jointures et des restrictions
  - Noms des buveurs qui ont consommé un vin rouge

$\Pi( \sigma( \bowtie( \text{Buveurs}$   
 $\bowtie(\text{Vins}, \text{Conso} / \text{Vins.NV} = \text{Conso.NV} )$   
 $/ \text{Conso.NB} = \text{Buveurs.NB} )$   
 $/ \text{Couleur} = ' \text{Rouge} ' )$   
 $/ \text{Nom} )$

# Représentation en Arbre Algébrique

Utile à l'optimiser pour déterminer un plan d'exécution de la requête optimal



# La Notion de Clé

---

- Clé d'un relation
  - Groupe minimum d'attributs dont la connaissance de la valeur permet de connaître celle des autres attributs
  - identifie un tuple parmi les autres
- Clés candidates (*Candidate Keys*)
  - plusieurs clés possibles pour la même relation
    - **CodePostal(Ville,Rue,CodPost) -> (Ville,Rue) ou (CodPost)**
- Clé primaire (*Primary Key*)
  - une clé choisie parmi les clés candidates (**CodPost**)
- Clés étrangères (*Foreign Keys*)
  - clé primaire C d'une relation A  
apparaissant parmi les attributs d'une relation B
  - on dit : C de B référence A
  - définit une contrainte d'intégrité référentielle

# Création des Tables en SQL

```

CREATE TABLE BUVEURS(
    NB          DECIMAL(5),
    NOM         CHAR(20),
    VILLE       CHAR(15),
    PRIMARY KEY(NB)                -- clé primaire
);
CREATE TABLE VINS(
    NV          CHAR(5),
    CRU         CHAR(20),
    MILL        INTEGER,
    REGION     CHAR(10),
    COULEUR    CHAR(10)
    PRIMARY KEY(NV)                -- clé primaire
);
CREATE TABLE CONSO(
    NB          DECIMAL(5),
    NV          CHAR(5),
    QTE         DECIMAL(5)
    PRIMARY KEY(NB,NV),           -- clé primaire
    FOREIGN KEY (NB) REFERENCES BUVEURS, -- clé étrangère
    FOREIGN KEY (NV) REFERENCES VINS,   -- clé étrangère
);

```

# Requêtes : ordre SELECT

SELECT	Colonnes à "Projeter"
FROM	Tables Sources
WHERE	Conditions sur les Lignes
GROUP BY	Colonnes de Groupage
HAVING	Condition sur les groupes
ORDER BY	Colonnes du tri

- Requêtes Mono-Tables
  - 1 seule table source
- Requêtes Multi-Tables
  - N tables sources
  - Condition testée sur chaque ligne issu de leur produit cartésien

# Exemples de Requêtes

---

## ■ Mono-Table

- Noms des buveurs parisiens
- Crus des bordeaux rouges de millésime 1994

## ■ Multi-Tables

- Noms des buveurs de vins
- Noms des buveurs parisiens qui ont consommé un vin rouge
  - *Exprimez les jointures de 4 manières différentes*
- Noms des buveurs qui ne boivent pas de vins

# ON

(Buveurs.NB=Conso.NB)

Buv	NB	Nom
	100	Al
	101	Ben
	102	Charlie
	103	David

Didier Donsez, 1998-2007, Rappel Bases

Conso	NB	Qte
	100	50
	100	25
	100	75
	101	100
	101	50
	102	25

NB	Nom	NB	Qte
100	Al	100	50
100	Al	100	25
100	Al	100	75
101	Ben	101	100
101	Ben	101	50
102	Charlie	102	25

## Réponses avec X et JOIN

```
SELECT BUVEURS.NOM
FROM   BUVEURS, CONSO, VINS
WHERE  BUVEURS.VILLE='Paris ' AND VINS.COULEUR='Rouge'
AND    BUVEURS.NB = CONSO.NB AND CONSO.NV = VINS.NV;
```

■ Et ...

```
SELECT BUVEURS.NOM
FROM   (BUVEURS JOIN CONSO ON BUVEURS.NB = CONSO.NB)
        JOIN VINS ON CONSO.NV = VINS.NV
WHERE  BUVEURS.VILLE='Paris' AND VINS.COULEUR='Rouge';
```

■ Et ...

```
SELECT BUVEURS.NOM
FROM   (BUVEURS JOIN CONSO USING(NB)) JOIN VINS USING(NV)
WHERE  BUVEURS.VILLE='Paris' AND VINS.COULEUR='Rouge';
```

■ Et ...



# Réponses avec IN et EXISTS

```

SELECT  BUVEURS.NOM
FROM    BUVEURS
WHERE   BUVEURS.VILLE='Paris'
AND     BUVEURS.NB IN (
        SELECT  CONSO.NB
        FROM    CONSO
        WHERE   CONSO.NV IN (
                SELECT  VINS.NV
                FROM    VINS
                WHERE   VINS.COULEUR='Rouge' )
        );

```

```

SELECT  BUVEURS.NOM
FROM    BUVEURS
WHERE   BUVEURS.VILLE='Paris'
AND     EXISTS(
        SELECT  *
        FROM    CONSO
        WHERE   CONSO.NB = BUVEURS.NB
        AND     EXISTS(
                SELECT  *
                FROM    VINS
                WHERE   VINS.NV=CONSO.NV
                AND     VINS.COULEUR='Rouge'
                )
        );

```

# Réponses avec NOT IN et NOT EXISTS

---

Noms des buveurs qui ne boivent pas de vins

```
SELECT  BUVEURS.NOM
FROM    BUVEURS
WHERE   BUVEURS.NB NOT IN (          SELECT  CONSO.NB
                                   FROM    CONSO
                                   );
```

```
SELECT  BUVEURS.NOM
FROM    BUVEURS
WHERE   NOT EXISTS(                SELECT  *
                                   FROM    CONSO
                                   WHERE   CONSO.NB = BUVEURS.NB
                                   );
```

# Agrégats et Groupage (i)

- Fonctions d 'agrégat
  - SUM, AVG (*Moyenne*), MIN, MAX, COUNT
  
- Agrégat Simple
  - Moyenne des consommations de vins rouges
  
- Agrégat Groupé (clause GROUP BY)
  - Moyenne des consommations par couleur de vin
  - Quantité totale de vins consommée par chaque buveur
  
- Condition sur les Groupes (clause HAVING)
  - Nom des buveurs qui consomment plus que la moyenne

# Agrégats et Groupage (ii)

Quantité totale de vins consommée par chaque buveur ?

Buv	NB	Nom
	100	Al
	101	Ben
	102	Charlie

NB	Nom	NB	Qte
100	Al	100	50
100	Al	100	25
100	Al	100	75
101	Ben	101	100
101	Ben	101	50
102	Charlie	102	25

Nom	Qte
Al	150
Ben	150
Charlie	25

Didier Donsez, 1998-2007, Rappel Bases

Conso	NB	Qte
	100	50
	100	25
	100	75
	101	100
	101	50
	102	25

54,166

108,33

Moyenne des quantités de vin consommées

Moyenne de la quantité totale de vins consommée par chaque buveur

## Agrégats et Groupage (Réponses)

- Moyenne des consommations de vins rouges

```
SELECT AVG(QTE) FROM CONSO, VINS
WHERE VINS.COULEUR='Rouge' AND CONSO.NV = VINS.NV;
```

- Moyenne des consommations par couleur de vin

```
SELECT COULEUR, AVG(QTE) FROM CONSO, VINS
WHERE CONSO.NV = VINS.NV GROUP BY COULEUR;
```

- Quantité totale de vins consommée par chaque buveur

```
SELECT NB, SUM(QTE) AS QTOT FROM CONSO GROUP BY NB;
```

- Nom des buveurs qui consomment plus que la moyenne

```
SELECT NOM FROM BUVEUR
WHERE NB IN (
  SELECT NB FROM CONSO GROUP BY NB
  HAVING SUM(QTE) >
    (SELECT AVG(QTOT) FROM
      (SELECT NB, SUM(QTE) AS QTOT FROM CONSO GROUP BY NB)));
```

## La Jointure externe (i)

- Liste des Buveurs avec leur consommation

```
SELECT BUVEURS.NOM, SUM(CONSO.QTE)
FROM   BUVEUR JOIN CONSO ON BUVEURS.NB = CONSO.NB;
```

- Remarque

- les buveurs n 'ayant pas bu n 'apparaissent pas

- Solution (avec des INNER JOIN)

```
(SELECT      BUVEURS.NOM, SUM(CONSO.QTE)
  FROM      BUVEUR JOIN CONSO ON BUVEURS.NB = CONSO.NB)
UNION
(SELECT      NOM, 0
  FROM      BUVEURS
  WHERE     NB NOT IN (SELECT NB FROM CONSO));
```

## La Jointure externe (ii)

- Définition

- Jointure

- Les lignes qui ne satisfont jamais la condition de jointure, sont complétés (padded) avec des nulls. Le "padding" peut être complet (FULL), à droite (RIGHT) ou à gauche (LEFT)

- La réponse

```
SELECT    BUVEURS.NOM, BUVEURS.NB, SUM(NVL(CONSO.QTE,0))
FROM      BUVEURS LEFT OUTER JOIN CONSO
          ON BUVEURS.NB = CONSO.NB
GROUP BY  BUVEURS.NOM, BUVEURS.NB;
```

# La Jointure externe (iii)

## LEFT OUTER JOIN

Buv	NB	Nom
	100	Al
	101	Ben
	102	Charlie
	103	David

Conso	NB	Qte
	100	50
	100	25
	100	75
	101	100
	101	50
	102	25

NB	Nom	NB	Qte
100	Al	100	50
100	Al	100	25
100	Al	100	75
101	Ben	101	100
101	Ben	101	50
102	Charlie	102	25
103	David	Null	Null

Nom	Qte
Al	150
Ben	150
Charlie	25
David	0



# La Jointure externe (iv)

## FULL OUTER JOIN

Buv	NB	Nom
	100	Al
	101	Ben
	102	Charlie
	103	David

Conso	NB	Qte
	100	50
	100	25
	100	75
	101	100
	101	50
	102	25
	104	125

NB	Nom	NB	Qte
100	Al	100	50
100	Al	100	25
100	Al	100	75
101	Ben	101	100
101	Ben	101	50
102	Charlie	102	25
103	David	Null	Null
Null	Null	104	125

# La Jointure externe (v)

## ■ Syntaxe Oracle

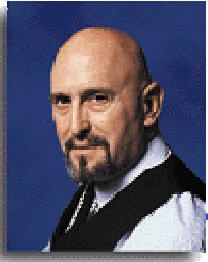
```
SELECT    BUVEURS.NOM, BUVEURS.NB, SUM(NVL(CONSO.QTE,0))
FROM      BUVEURS, CONSO
WHERE     BUVEURS.NB (+) = CONSO.NB
GROUP BY  BUVEURS.NOM, BUVEURS.NB;
```

## ■ Syntaxe SQLBase, DB2

```
SELECT    BUVEURS.NOM, BUVEURS.NB, SUM(NVL(CONSO.QTE,0))
FROM      BUVEURS (+), CONSO
WHERE     BUVEURS.NB = CONSO.NB
GROUP BY  BUVEURS.NOM, BUVEURS.NB;
```

## ■ Syntaxe Sybase

```
SELECT    BUVEURS.NOM, BUVEURS.NB, SUM(NVL(CONSO.QTE,0))
FROM      BUVEURS, CONSO
WHERE     BUVEURS.NB += CONSO.NB
GROUP BY  BUVEURS.NOM, BUVEURS.NB;
```



## Puzzle de Joe Celko ([www.dbmsmag.com](http://www.dbmsmag.com))

- La table NAME contient ces 13 noms
- On veut ce tableau

name1	name2	name3	name4
Al	Ben	Charlie	David
Ed	Frank	Greg	Howard
Ida	Joe	Ken	Larry
Mike	NULL	NULL	NULL

## Élément de réponse

- Que donne cette requête ?

```
SELECT N1.name, MIN(N2.name) AS name2
FROM Names AS N1 LEFT OUTER JOIN Names AS N2 ON N1.name < N2.name
WHERE N1.name
  IN (
    SELECT A.name FROM Names AS A INNER JOIN Names AS B ON A.name <=
      B.name
      GROUP BY A.name
      HAVING MOD(COUNT(B.name), 2) = (SELECT MOD(COUNT(*),2) FROM Names)
  )
GROUP BY N1.name
ORDER BY N1.name;
```

# La réponse

---

```

SELECT N1.name,
       MIN(N2.name) AS name2,
       MIN(N3.name) AS name3,
       MIN(N4.name) AS name4,
       MIN(N5.name) AS name5
FROM (Names AS N1
      LEFT OUTER JOIN
      Names AS N2
      ON N1.name < N2.name)
LEFT OUTER JOIN
Names AS N3
ON N1.name < N2.name
   AND N2.name < N3.name
LEFT OUTER JOIN
Names AS N4
ON N1.name < N2.name
   AND N2.name < N3.name
   AND N3.name < N4.name

```

```

LEFT OUTER JOIN
Names AS N5
ON N1.name < N2.name
   AND N2.name < N3.name
   AND N3.name < N4.name
   AND N4.name < N5.name
WHERE N1.name IN (SELECT A.name
                  FROM Names AS A
                  INNER JOIN
                  Names AS B
                  ON A.name <= B.name
                  GROUP BY A.name
                  HAVING MOD(COUNT(B.name), 5) =
                  (SELECT MOD(COUNT(*),5)
                   FROM Names))
GROUP BY N1.name ORDER BY N1.name;

```

## Le calcul du Rang (Top Five)

- On veut connaître le rang des buveurs en consommation (jusqu'à la 5ème place) par couleur

Conso	Nom	Couleur	Qte
	Alice	Rouge	1000
	Bob	Rouge	900
	Cary	Rouge	800
	Daniel	Rouge	800
	Eric	Rouge	700
	Frank	Rouge	600
	Gilles	Rouge	600

Rang
1
2
3
3
4
5
5

Rang
1
2
4
4
5
7
7

COUNT(Distinct Qte)

COUNT(Qte)

## Le calcul du Rang (Top5)

- La réponse

```
SELECT C0.Couleur, C0.Nom, C0.Qte,  
       (SELECT COUNT(DISTINCT Qte) FROM Conso AS C1  
        WHERE (C1.Qte >=C0.Qte) AND (C1.Couleur=C0.Couleur))  
       AS RANG  
FROM Conso AS C0  
WHERE RANG<= :n; -- n est ajustable (5 pour un Top5)
```

# Les Vues (*Views*)

## ■ Schéma Externe

- définit à partir du schéma conceptuel ou d'une autre vue

## ■ Exemple

- Vue sur les vins rouges
- Vue sur les gros buveurs (quantité totale > 100)

```
CREATE VIEW GROSBUEURS(NB, NOM, VILLE, QTE)
AS SELECT  NB, NOM, VILLE, SUM(QTE)
  FROM    BUEURS, CONSO
  WHERE   BUEURS.NB=CONSO.NB
  GROUP BY NB
  HAVING SUM(QTE) > (SELECT AVG(QTETOT) FROM
                    (SELECT SUM(QTE) AS QTETOT FROM CONSO GROUP BY NB))
```

## ■ Modification sur une vue

- pas toujours possible (à moins d'un trigger INSTEAD OF)



# Les Modifications

---

## ■ Insertion

- “ Insérer un nouveau tuple décrivant un Riesling de 1997 ”  
**INSERT INTO VINS (NV, CRU, MILL, COULEUR)**  
**VALUES ('RIES1', 'Riesling', 1997, 'Blanc')**

## ■ Modification de tuples

- “ Augmenter de 1 les consommations du buveur 101 ”  
**UPDATE CONSO SET QTE = CONSO.QTE + 1**  
**WHERE (CONSO.NB=101)**

## ■ Suppression de tuples

- “ Supprimer les consommations des buveurs habitant Paris ”  
**DELETE FROM CONSO**  
**WHERE NB IN (SELECT NB FROM BUVEURS WHERE VILLE = 'Paris')**

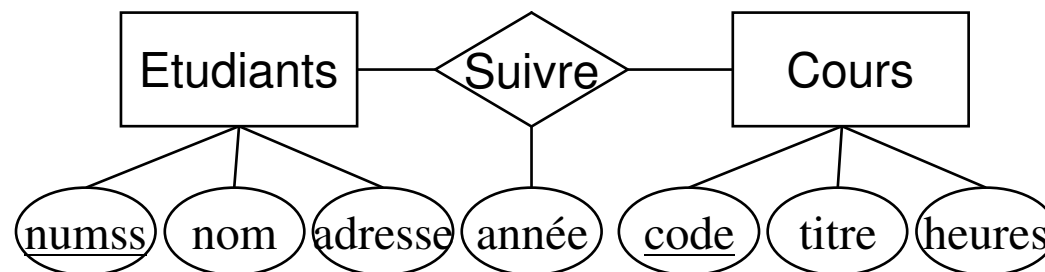
# Les limites de SQL

---

- Compatibilité SQL92 des SGBDs
  - 3 niveaux (Full, Intermediate, Entry)
- Programmation Procédurale
  - puzzle de Joe Celko, rang (Top Ten), ...
  - Réponse : Embedded SQL, ODBC/JDBC, PL/SQL, ...
- Récursion
  - relation composant/composé, graphe, arbre, ...
  - Réponse : SQL3
- Orientation-Objet
  - Abstraction, Spécialisation, Généralisation
  - Réponse : SQL3 et le modèle Objet-Relationnel

# Conception de Bases de Données

- 2 approches complémentaires
- Théorie de la Normalisation
  - Déterminer les attributs et leurs dépendances
  - Décomposer la relation universel en 3<sup>ème</sup> forme normale
    - suppression des redondances
- Modèle Entité-Association E/A [Chen79]
  - Concept repris par tous (MERISE, OMT, ..., UML)
  - Modéliser en Entité et en Association



- Transformation vers le modèle relationnel

# Bibliographie

---

- Chris Date, "Introduction aux Bases de Données", 6<sup>ème</sup> édition, Ed Intl Thomson Publ. ISBN 2-84180-964-1, 970 pp
  - *7ème édition en anglais*
- Jeffrey D. Ullman, Jennifer Widom, "A First Course in Database Systems", 1<sup>ère</sup> édition, Ed. Prentice Hall Engineering, Science & Math, Avril 1997, ISBN 0-13-861337-0, 470 pp.
- G. Gardarin, "Bases de Données Objet et Relationnel", Ed Eyrolles, 1999, ISBN 2-212-09060-9
- Joe Celko, "SQL Avancé", 1997, Ed Intl Thomson Publ., ISBN 2-84180-141-1
- [www.dbmsmag.com](http://www.dbmsmag.com)