

# Chapitre 5: le langage SQL

But: accéder aux données de la base de données

SQL = **S**tructured **Q**uery **L**anguage

Langage pour les bases de données relationnelles

SQL est défini par une norme ISO

Existe sur tous les SGBD relationnels (Oracle, Access,...)

Origine : SEQUEL (**S**tructured **E**nglish **Q**Uery **L**anguage) développé par IBM pour le SGBD appelé SYSTEM R (au milieu des années 70)

Première réalisation commerciale de SQL: 1979 (Oracle Corporation)

Concepts de base: table, colonne, ligne

## Le langage SQL

SQL est un langage *déclaratif*

(par opposition à un langage procédural)

SQL fournit un ensemble de *commandes* pour une variété de tâches, dont:

- l'interrogation de la base de données
- l'insertion, la mise à jour et la suppression des données dans la base de données
- la création et la modification du schéma de la BD
- la définition de vues
- le contrôle de l'accès aux données
- la création d'index pour accélérer les interrogations

Une commande SQL est aussi appelée instruction SQL ou **requête SQL**.

## Terminologie SQL

### *Terminologie*

Relationnel:

relation

tuple

attribut

domaine

SQL:

table (table)

ligne (row)

colonne (column)

type de donnée (datatype)

### *Autre différence*

les tables SQL peuvent contenir des doublons (lignes)

-> ce ne sont pas vraiment des relations.

## L'interrogation en SQL: la commande SELECT

En SQL, l'interrogation d'une base de données se fait avec la commande SELECT

*Forme de base de la commande SELECT*

SELECT < liste des colonnes de la table résultat >

FROM < liste des tables impliquées dans l'interrogation >

WHERE < condition de sélection des tuples >

*Exemple ( → A-3 pour la définition de "Chambres")*

SELECT NumChambre, Confort

FROM Chambres

WHERE Confort='bain' AND Equipement='TV'

## De l'algèbre relationnelle à l'interrogation en SQL

La relation R

$R \equiv \text{select } * \text{ from } R$

La projection  $\pi_{A_u, A_v, \dots, A_z}(R)$

$\pi_{A_u, A_v, \dots, A_z}(R) \equiv \text{select } A_u, A_v, \dots, A_z \text{ from } R$

La sélection  $\sigma_F(R)$

$\sigma_F(R) \equiv \text{select } * \text{ from } R \text{ where } F$

Le produit cartésien  $R \times S$

$R \times S \equiv \text{select } R.*, S.* \text{ from } R, S$

La jointure  $R_{(A_{i1} \theta_1 B_{j1}) \wedge (A_{i2} \theta_2 B_{j2}) \wedge \dots \wedge (A_{im} \theta_m B_{jm})} \bowtie S$

$R_{(A_{i1} \theta_1 B_{j1}) \wedge (A_{i2} \theta_2 B_{j2}) \wedge \dots \wedge (A_{im} \theta_m B_{jm})} \bowtie S \equiv$

$\text{select } R.*, S.* \text{ from } R, S$

$\text{where } R.A_{i1} \theta_1 S.B_{j1}$

$\text{and } R.A_{i2} \theta_2 S.B_{j2}$

...

$\text{and } R.A_{im} \theta_m S.B_{jm}$

où  $\theta_i$  est un opérateur de comparaison ( $=, <, \leq, >, \geq, \neq$ )

notation SQL:  $=, <, <=, >, >=, <>$  (noté aussi  $\neq$  et  $\wedge =$ )

## De l'algèbre relationnelle à SQL (suite)

La jointure naturelle  $R \bowtie S$

si les attributs de jointure de R et S sont  $A_u, A_v, \dots, A_z$   
et les attributs propres de S sont  $B_p, B_q, \dots, B_t$

$R \bowtie S \equiv \text{select } R.*, B_p, B_q, \dots, B_t \text{ from } R, S$

where  $R.A_u = S.A_u$

and  $R.A_v = S.A_v$

...

and  $R.A_z = S.A_z$

L'union  $R \cup S$

$R \cup S \equiv \text{select } * \text{ from } R \text{ union select } * \text{ from } S$

L'intersection  $R \cap S$

$R \cap S \equiv \text{select } * \text{ from } R \text{ intersect select } * \text{ from } S$

La différence  $R - S$

$R - S \equiv \text{select } * \text{ from } R \text{ minus select } * \text{ from } S$

( $\rightarrow$  A-24 pour ces trois opérations ensemblistes)

## De l'algèbre relationnelle à SQL: cas général

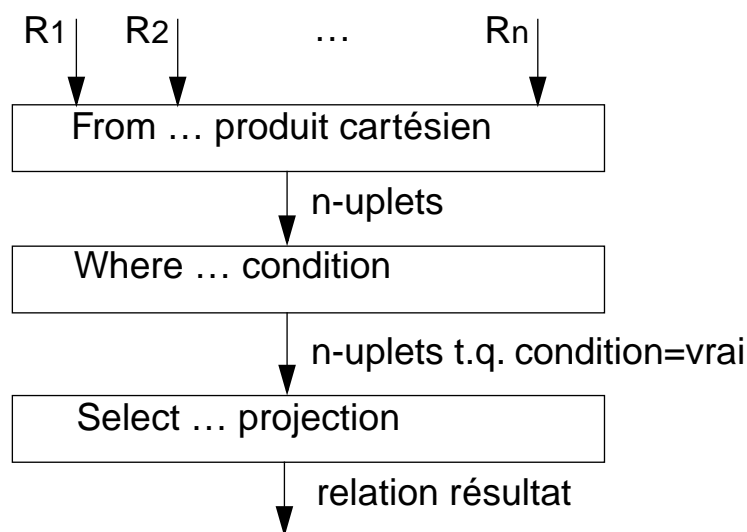
D'une manière générale, la requête SQL

**select** A<sub>1</sub>,A<sub>2</sub>,...A<sub>m</sub> **from** R<sub>1</sub>, R<sub>2</sub>,...,R<sub>n</sub> **where** F

est équivalente à

$$\pi_{A_1, A_2, \dots, A_m}(\sigma_F(R_1 \times R_2 \times \dots \times R_n))$$

Cette équivalence nous donne le *modèle d'exécution* simplifié de la machine SQL:



Remarques:

- ce modèle est un modèle logique destiné à l'utilisateur; il lui fournit la sémantique des requêtes **select** de SQL;
- l'exécution réelle d'une requête **select** est différente: propriétés de l'algèbre relationnelle -> optimisations.

## De l'algèbre relationnelle à SQL: exemples

→ A-8

Soit les instances de **cru**, **vins** et **cepage\_region**:

instance de **cru**:

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
Ch. Myrat	Barsac	Bordeaux	blanc
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge
Les Perrieres	Meursault	Bourgogne	blanc
Les Charmes	Meursault	Bourgogne	blanc
La Grappe d'Or	Meursault	Bourgogne	blanc

instance de **vins**:

REGION	COUL	MILLESIME	QUALITE
Bordeaux	rouge	1990	tres bonne
Bordeaux	blanc	1990	excellente
Bourgogne	rouge	1990	excellente
Bourgogne	blanc	1990	bonne
Bordeaux	rouge	1991	moyenne
Bordeaux	blanc	1991	mediocre
Bourgogne	rouge	1991	moyenne
Bourgogne	blanc	1991	moyenne

instance de **cepage\_region**:

CEPAGE	R_PROD	COUL
Cabernet-Sauvignon	Bordeaux	rouge
Pinot noir	Bourgogne	rouge
Semillon	Bordeaux	blanc
Chardonnay	Bourgogne	blanc
Chardonnay	Champagne	blanc



## De l'algèbre relationnelle à SQL: exemples (suite)

Q1 : "Tous les crus"

Expression algébrique: cru

SQL> select \* from cru;

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
Ch. Myrat	Barsac	Bordeaux	blanc
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge
Les Perrieres	Meursault	Bourgogne	blanc
Les Charmes	Meursault	Bourgogne	blanc
La Grappe d'Or	Meursault	Bourgogne	blanc

14 rows selected.

Q2: "Tous les crus rouges"

E.a:  $\sigma_{\text{COUL}=\text{rouge}}(\text{cru})$

SQL> select \* from cru where coul='rouge';

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge

9 rows selected.

## De l'algèbre relationnelle à SQL: exemples (suite)

Q3: "La liste des noms de crus rouges"

E.a:  $\pi_{\text{NOM\_CRU}}(\sigma_{\text{COUL.}=\text{rouge}}(\text{cru}))$

SQL> select nom\_cru from cru where coul='rouge';

```
NOM_CRU
-----
Ch. Margaux
Ch. Rausan-Segla
Ch. Latour
Ch. Lynch-Bages
Ch. Lagrange
Clos Vougeot
Corton
Les Epenots
Les Gravieres

9 rows selected.
```

Q4: "À partir de quel cépage principal est produit le Meursault blanc ?"

E.a:  $\pi_{\text{CÉPAGE}}(\sigma_{\text{COMMUNE}=\text{Meursault}} \wedge (\text{COUL.}=\text{blanc}))$   
 $(\text{cru} \bowtie (\text{COUL.}=\text{COUL.}) \wedge (\text{RÉGION} = \text{R.PROD}))$   
cépage\_région))

SQL> select cepage from cepage\_region, cru  
where region=r\_prod and cepage\_region.coul=cru.coul  
and commune='Meursault' and cru.coul='blanc';

```
CEPAGE
-----
Chardonnay
Chardonnay
Chardonnay
```

Remarque: nous verrons plus loin comment éliminer les doublons

## De l'algèbre relationnelle à SQL: exemples (suite)

Q5: “Quels sont les bons millésimes du Château Latour?” (i.e. bon, très bon ou excellent)

E.a:  $\pi_{\text{MILL.}}(\sigma_{\text{QUALITÉ} \geq \text{bonne} \wedge \text{NOM\_CRU} = \text{Ch. Latour}}(\text{cru} \bowtie \text{vins}))$

```
SQL> select millesime from vins, cru
      where vins.region=cru.region and vins.coul=cru.coul
      and nom_cru='Ch. Latour'
      and (qualite='bonne' or qualite='tres bonne'
           or qualite = 'excellente');
```

```
MILLESIME
-----
      1990
```

Q6 : “Quels sont les crus rouges et leurs millésimes qui sont de bonne qualité?”

E.a:  $\pi_{\text{NOM-CRU, COMMUNE, RÉGION, COUL., MILL.}}(\sigma_{\text{COUL.} = \text{rouge} \wedge \text{QUALITÉ} \geq \text{bonne}}(\text{cru} \bowtie \text{vins}))$

```
SQL> select cru.*, millesime from cru, vins
      where cru.region=vins.region and cru.coul=vins.coul
      and cru.coul='rouge'
      and (qualite='bonne' or qualite='tres bonne'
           or qualite = 'excellente');
```

## De l'algèbre relationnelle à SQL: exemples (suite)

NOM_CRU	COMMUNE	REGION	COUL	MILLESIME
Ch. Margaux	Margaux	Bordeaux	rouge	1990
Ch. Rausan-Segla	Margaux	Bordeaux	rouge	1990
Ch. Latour	Pauillac	Bordeaux	rouge	1990
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge	1990
Ch. Lagrange	St. Julien	Bordeaux	rouge	1990
Clos Vougeot	Vougeot	Bourgogne	rouge	1990
Corton	Aloxe-Corton	Bourgogne	rouge	1990
Les Epenots	Pommard	Bourgogne	rouge	1990
Les Gravieres	Santenay	Bourgogne	rouge	1990

9 rows selected.

Q7: “Quels sont les millésimes où les Bourgognes blancs sont de qualité supérieure au Bordeaux blancs?”

E.a:  $\pi_{\text{MILL.}}(\sigma_{(\text{REGION}=\text{Bourgogne}) \wedge (\text{COUL.}=\text{blanc})}(\text{vins}))$   
 $\bowtie (\text{MILL.}=\text{MILL.}) \wedge (\text{QUALITÉ} > \text{QUALITÉ})$   
 $\sigma_{(\text{REGION}=\text{Bordeaux}) \wedge (\text{COUL.}=\text{blanc})}(\text{vins})$

```
SQL> select bourg.millesime from vins bord,vins bourg
      where bord.region='Bordeaux' and bord.coul='blanc'
      and bourg.region='Bourgogne' and bourg.coul='blanc'
      and bourg.millesime=bord.millesime
      and bourg.qualite > bord.qualite;
```

```
MILLESIME
-----
      1991
```

Remarque: pour que cette interrogation SQL donne toujours le résultat escompté, il faudrait que le type de données de la colonne “qualité” soit numérique (voir plus loin).

## La forme complète de la commande SELECT

A l'aide d'exemples, nous allons examiner en détail les différentes parties de la commande SELECT, à savoir

- les colonnes du résultat
  - l'utilisation de \*
  - les noms de colonnes ambigus
  - les fonctions d'agrégation
  - alias pour une colonne
- les tables de l'interrogation
  - alias pour une table
- la table comme ensemble(élimination des doublons)
- la condition de sélection
  - la selection sans condition
  - les opérateurs de comparaison
  - les interrogations imbriquées
  - les quantificateurs en SQL ( $\exists$  ,  $\forall$  )
- la jointure externe
- le regroupement
- les opérations ensemblistes
- le tri du résultat

## Les types de données en SQL

- type de données d'une colonne  $\equiv$  domaine d'un attribut
- il y a essentiellement trois types de données en SQL:
  - **char(n)** et **varchar2(n)**: chaîne de n caractères
  - **number(p)** et **number(p,f)** et : nombre de p chiffres dont f chiffres après la virgule
  - **date**: une valeur de type date s'écrit sur le modèle **'21-feb-94'**
- Ex: create table vins1 (region char(16), coul char(5), millesime number(4), qualite number(1));
- Instance de la table **vins1**:

REGION	COUL	MILLESIME	QUALITE
Bordeaux	rouge	1985	7
Bordeaux	blanc	1985	5
Bourgogne	rouge	1985	7
Bourgogne	blanc	1985	7
Bordeaux	rouge	1986	6
Bordeaux	blanc	1986	7
Bourgogne	rouge	1986	5
Bourgogne	blanc	1986	7
Bordeaux	rouge	1987	4
Bordeaux	blanc	1987	4
Bourgogne	rouge	1987	5
Bourgogne	blanc	1987	5
Bordeaux	rouge	1988	5
Bordeaux	blanc	1988	7
Bourgogne	rouge	1988	7
Bourgogne	blanc	1988	7
Bordeaux	rouge	1989	6
Bordeaux	blanc	1989	7
Bourgogne	rouge	1989	7
Bourgogne	blanc	1989	7
Bordeaux	rouge	1990	6
Bordeaux	blanc	1990	7
Bourgogne	rouge	1990	7
Bourgogne	blanc	1990	5
Bordeaux	rouge	1991	4
Bordeaux	blanc	1991	3
Bourgogne	rouge	1991	4
Bourgogne	blanc	1991	4

## Les colonnes du résultat (displayed column)

### *Cas de base*

les colonnes que l'on désire afficher comme résultat doivent être énumérées après "select "

Q3: "La liste des noms de crus rouges"

```
select nom_cru from cru where coul='rouge'
```

```
NOM_CRU
```

```
-----  
Ch. Margaux  
Ch. Rausan-Segla  
Ch. Latour  
Ch. Lynch-Bages  
Ch. Lagrange  
Clos Vougeot  
Corton  
Les Epenots  
Les Gravieres
```

### *Utilisation de \**

"select \* from ..." entraîne l'affichage de toutes les colonnes de toutes les tables citée après "from"

Q8: "La description des crus des bourgognes 1990"

```
select * from cru, vins  
where cru.region=vins.region and cru.coul=vins.coul  
and region=bourgogne and millesime=1990
```

```
NOM_CRU      COMMUNE      REGION      COUL      REGION      COUL      MILL      QUALITE  
-----  
Les Perrieres  Meursault  Bourgogne  blanc  Bourgogne  blanc  1990  bonne  
Les Charmes   Meursault  Bourgogne  blanc  Bourgogne  blanc  1990  bonne  
La Grappe d'Or Meursault  Bourgogne  blanc  Bourgogne  blanc  1990  bonne  
Clos Vougeot  Vougeot    Bourgogne  rouge  Bourgogne  rouge  1990  excellente  
Corton        Aloxe-Corton Bourgogne  rouge  Bourgogne  rouge  1990  excellente  
Les Epenots   Pommard    Bourgogne  rouge  Bourgogne  rouge  1990  excellente  
Les Gravieres Santenay    Bourgogne  rouge  Bourgogne  rouge  1990  excellente
```

## Les colonnes du résultat (suite)

la requête “select **t.\***,... from t,...” a pour effet d’afficher toutes les colonnes de la table t

Q6 : “Quels sont les crus rouges et leurs millésimes qui sont de bonne qualité?”

```
select cru.*, millesime from cru, vins
      where cru.region=vins.region and cru.coul=vins.coul
      and cru.coul='rouge'
      and (qualite='bonne' or qualite='tres bonne'
           or qualite = 'excellente');
```

### *Noms de colonne ambigus*

ce cas se produit lorsque l’interrogation se réfère à des colonnes qui portent le même nom mais qui appartiennent à des tables différentes -> pour lever l’ambiguïté, il faut préfixer le nom de la colonne par le nom de la table à laquelle elle appartient

Q4: “À partir de quel cépage principal est produit le Meursault blanc ?”

```
select cepage from cepage_region, cru
      where region=r_prod and cepage_region.coul=cru.coul
      and commune='Meursault';
```

CEPAGE

-----  
Chardonnay  
Chardonnay  
Chardonnay



## Les colonnes du résultat (suite)

### *Les fonctions d'agrégation*

opèrent sur une liste de valeurs; on les utilise généralement sur une colonne

- **avg**: calcule la moyenne d'une liste
- **count**: compte le nombre d'éléments d'une liste
- **min**: donne la valeur minimum d'une liste
- **max**: donne la valeur maximum d'une liste
- **sum**: calcule la somme d'une liste

Q9: "La capacité théorique d'accueil de l'hôtel"

```
select sum(nbr_pers) from CHAMBRES
```

```
SUM(NBR_PERS)
```

```
-----  
56
```

## Les colonnes du résultat (suite)

### *Alias pour une colonne*

permet de donner un nom à une colonne du résultat; lors de l'utilisation d'une fonction d'agrégation:

Q9a:

```
select sum(nbr_pers) 'capacité de l'hôtel'  
from chambres
```

```
capacité de l'hôtel  
-----  
56
```

ou pour expliciter une colonne:

Q10: "les qualités du millésime 1986"

```
select region, coul, millesime, qualite 'QUALITE (1 à 7)'  
from vins1 where millesime=1986;
```

REGION	COUL	MILLESIME	QUALITE(1 à 7)
Bordeaux	rouge	1986	6
Bordeaux	blanc	1986	7
Bourgogne	rouge	1986	5
Bourgogne	blanc	1986	7

## Les tables de l'interrogation (selected table)

### *Cas de base*

toutes les tables impliquées dans l'interrogation doivent être citées après "from"

Q5: "Quels sont les bons millésimes du Château Latour?"

```
select millesime from vins, cru
where vins.region=cru.region and vins.coul=cru.coul
and nom_cru='Ch. Latour' and (qualite='bonne'
    or qualite='tres bonne' or qualite = 'excellente');
```

MILLESIME

-----  
1990

### *Alias pour une table*

lorsqu'il faut joindre plusieurs fois la même table;

Q7: "Quels sont les millésimes où les Bourgognes blancs sont de qualité supérieure au Bordeaux blancs?"

```
select bourg.millesime from vins1 bord,vins1 bourg
where bord.region='Bordeaux' and bord.coul='blanc'
and bourg.region='Bourgogne' and bourg.coul='blanc'
and bourg.millesime=bord.millesime
and bourg.qualite > bord.qualite;
```

MILLESIME

-----  
1985  
1987  
1988  
1991

## La table comme ensemble

Rappel: une table SQL peut contenir plusieurs tuples identiques

### *Utilisation de **distinct***

élimine les doublons du résultat

Q4a: “À partir de quel cépage principal est produit le Meursault blanc ?”

```
select distinct cepage from cepage_region, cru
where region=r_prod and cepage_region.coul=cru.coul
      and commune='Meursault' and cru.coul='blanc';
```

CEPAGE

-----  
Chardonnay

Remarques:

- l'élimination des doublons est “coûteuse” car elle nécessite le tri préalable des tuples
- les opérateurs ensemblistes **union**, **intersect** et **minus** éliminent automatiquement les doublons du résultat (voir plus loin)

## La condition de sélection

### *Cas de base*

la commande SELECT sélectionne tous les tuples de la table (ou des tables) spécifiée après FROM qui satisfont la condition de sélection spécifiée après WHERE

Q2: "Tous les crus rouges"

```
select * from cru where coul='rouge';
```

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge

### *Sélection sans condition*

l'omission de la partie WHERE indique qu'il n'y a aucune condition de sélection des tuples -> tous les tuples de la table spécifiée après WHERE sont sélectionnés

Q1 : "Tous les crus"

```
select * from cru;
```

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
...	...	...	...

14 rows selected.

## La condition de sélection (suite)

### *Les opérateurs de comparaison*

- =, <, <=, >, >=, <> (noté aussi != et ^=)
- **in** et **not in**: appartenance (resp. non appartenance) à une liste

Q5a: “Quels sont les bons millésimes du Château Latour?”

```
select millesime from vins, cru
```

```
where vins.region=cru.region and vins.coul=cru.coul  
and nom_cru='Ch. Latour'
```

```
and qualite in ('bonne', 'tres bonne', 'excellente');
```

```
MILLESIME
```

```
-----  
1990
```

- **like**: appartenance à une chaîne de caractère  
Q11: “La liste des articles qui contiennent le mot 'ordinateur' dans leur titre  
select \* from article where titre like '%ordinateur%';

```
TITRE
```

```
NOM_AUTEUR
```

```
-----  
PRENOM_AUTEUR
```

```
COD NUM
```

```
ANNEE
```

```
-----  
L'ordinateur va-t'il imposer sa maniere de penser? Moeckli  
Gustave TS 10 1984
```

```
Quand je pense "a", l'ordinateur repete "a" Demenjoz  
Luc LNQ 658 1993
```

(explication de % et \_ → A-16, A-32 et A-33)

## Les opérateurs de comparaison (suite)

- **between**: appartenance à un intervalle

Q12: “Nombre de chambres dont le prix est compris entre 85 et 120 francs”

```
select count(*) from chambres
where prix between 85 and 120
```

```
count(*)
-----
      12
```

- **is null** et **is not null**: comparaison avec la valeur **null**; utilisé généralement pour rechercher dans une table les tuples dont la valeur d'un attribut est indéfinie;

Q13: “Le nom des clients qui n'ont pas annoncé leur date de départ”

```
select nom from clients, reservation
where clients.num_client=reservations.num_client
and date_dep is null;
```

```
NOM
-----
DUPONT
```

## La condition de sélection (suite)

### *Interrogations imbriquées*

certaines interrogations nécessite la connaissance préalable de certaines données de la bd pour pouvoir spécifier la condition de sélection

Q14: “Quels sont les millésimes du Bordeaux rouge dont la qualité est supérieure à la moyenne ( qualité moyenne des millésimes du Bordeaux rouge)”

```
select millesime from vins1
where region='Bordeaux' and coul='rouge'
and qualite > (select avg(qualite) from vins1 where
               region='Bordeaux' and coul='rouge');
```

```
MILLESIME
-----
    1985
    1986
    1989
    1990
```

Q15: “La recette du 25 décembre 1989”

```
select sum(prix) from chambres
where num_chambre in (select num_chambre
                      from reservations
                      where date_arr <= '25-dec-89'
                      and date_dep > '25-dec-89')
```

```
SUM(PRIX)
-----
        600
```



## Les quantificateurs en SQL ( $\exists$ , $\forall$ )

### Exists ( $\exists$ ), not exists( $\exists$ )

- permet de tester si le résultat d'une interrogation imbriquée contient au moins un tuple (respectivement aucun tuple).

### Soit la relation **langue**

NOM_CANTON	LANGUE_PARLEE
Appenzell R.E.	allemand
Appenzell R.I.	allemand
Argovie	allemand
Bale-Campagne	allemand
Bale-Ville	allemand
Berne	allemand
Berne	français
Fribourg	allemand
Fribourg	français
Geneve	français
Glaris	allemand
Grisons	allemand
Grisons	romanche
Grisons	italien
Jura	français
...	...

Q16: "Quels sont les cantons suisse où l'on ne parle pas l'allemand?"

```
select * from canton where not exists
  (select * from langue
   where canton.nom_canton=langue.nom_canton
   and langue_parlee='allemand');
```

NOM_CANTON	CHEF_LIEU	DATE_ENTRE
Geneve	Geneve	1815
Jura	Delemont	1979
Neuchatel	Neuchatel	1815
Tessin	Bellinzone	1803
Vaud	Lausanne	1803

**All ( $\forall$ ):** → A-19 et A-35

## La jointure externe

En postfixant dans la condition les colonnes d'une table par (+), on spécifie une jointure externe.

*Exemple: soit les tables "cabine" et "reservation":*

*cabine*

NUM_CABINE	NBR_PERS	CONFORT	-----	PRIX
1	2	bain	-----	4000
2	2	douche	-----	3500
3	1	bain	-----	2500
4	2	douche	-----	3000
5	2	douche	-----	3000
6	4	douche	-----	5000

*reservation*

NUM_CABINE	NOM_CLIENT
1	Arditi P.
5	Dupont J.
6	Dupont J.

Requête: "Afficher toutes les cabines et le nom du client qui a réservé la cabine".

```
select cabine.*, nom_client from cabine, reservation
where cabine.num_cabine = reservation.num_cabine
```

NUM_CABINE	NBR_PERS	CONFORT	-----	PRIX	NOM_CLIENT
1	2	bain	-----	4000	Arditi P.
5	2	douche	-----	3000	Dupont J.
6	4	douche	-----	5000	Dupont J.

```
select cabine.*, nom_client from cabine, reservation
where cabine.num_cabine = reservation.num_cabine(+)
```

NUM_CABINE	NBR_PERS	CONFORT	-----	PRIX	NOM_CLIENT
1	2	bain	-----	4000	Arditi P.
2	2	douche	-----	3500	
3	1	bain	-----	2500	
4	2	douche	-----	3000	
5	2	douche	-----	3000	Dupont J.
6	4	douche	-----	5000	Dupont J.

## Les regroupements

permet d'appliquer les fonctions d'aggrégation à des sous-groupes de tuples

Q17: "Le prix minimum et maximum des chambres par type de confort"

```
select confort, min(prix) 'PRIX MINIMUM', max(prix)
'PRIX MAXIMUM' from chambres group by confort;
```

CONFORT	PRIX MINIMUM	PRIX MAXIMUM
BAIN	120	180
DOUCHE	100	100
WC	80	90

Q18: "Combien de crus rouges et de crus blancs sont produits dans chaque commune?"

```
select commune, coul, count(*) from cru
group by commune, coul;
```

COMMUNE	COUL	COUNT( *)
Aloxe-Corton	rouge	1
Barsac	blanc	1
Margaux	rouge	2
Meursault	blanc	3
Meursault	rouge	1
Pauillac	rouge	2
Pommard	rouge	1
Santenay	rouge	1
Sauternes	blanc	1
St. Julien	rouge	1
Vougeot	rouge	1

11 ligne(s) sélectionnée(s).

## Les opérations ensemblistes

opérateurs: **union** ( $\cup$ ), **intersect** ( $\cap$ ) et **minus** ( $-$ )

- les opérands doivent avoir le même nombre de colonnes et les colonnes correspondantes doivent être égales en type (pas forcément en taille)
- l'utilisation de ces opérateurs implique implicitement la clause **distinct**

Q19: “Quels sont les termes employés pour décrire l'équipement et le confort d'une chambre?”

select confort 'TERMES' from chambres

**union**

select equipement from chambres;

TERMES

-----

BAIN

DOUCHE

NON

TV

WC

## Le tri du résultat

- le résultat est trié selon une ou plusieurs clés de tri;
- s'il y a plusieurs clés de tri, la première de la liste est la clé primaire, la deuxième est la clé secondaire etc.
- **asc** -> ordre croissant (par défaut)
- **desc** -> ordre décroissant

Q11: "les cantons suisses dans l'ordre d'entrée dans la confédération et par ordre alphabétique"

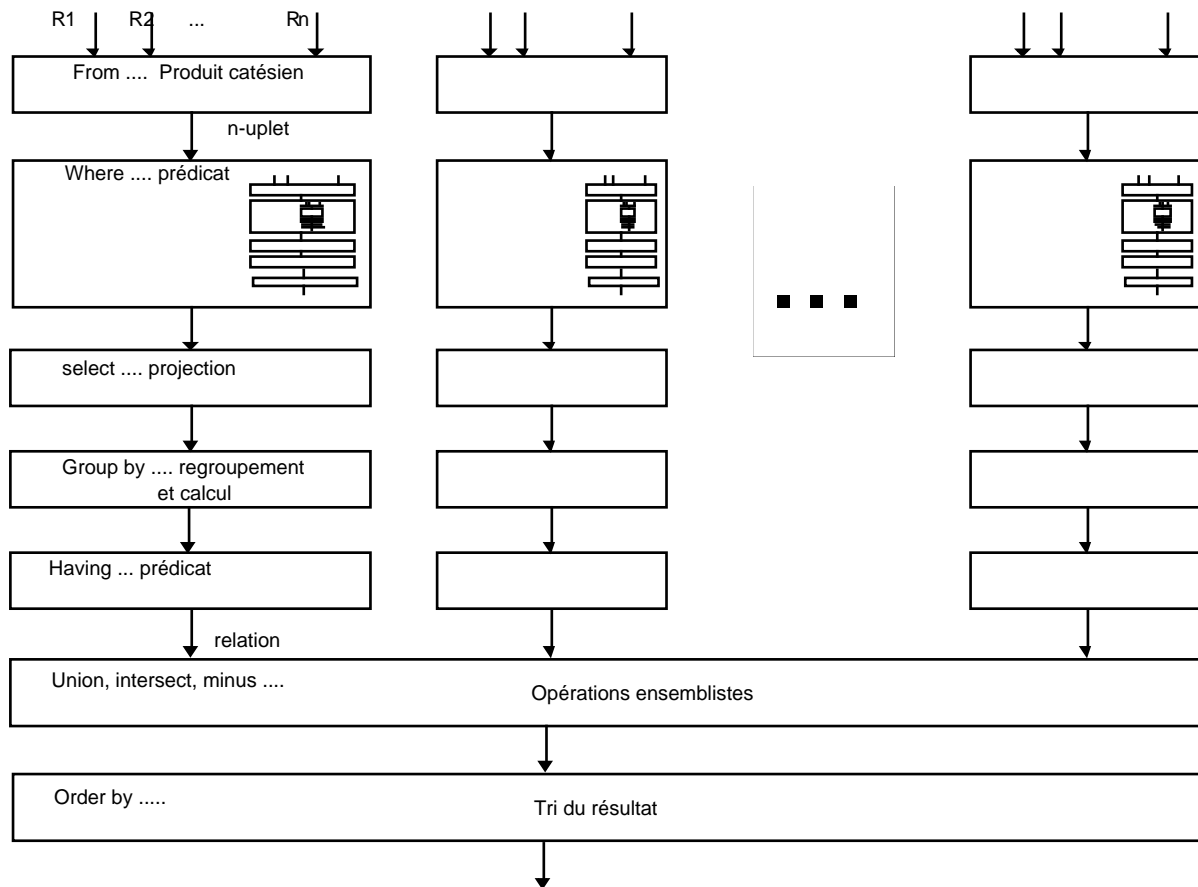
```
select * from canton
order by date_entree asc, nom_canton asc
```

NOM_CANTON	CHEF_LIEU	DATE_ENTREE
Nidwald	Stans	1291
Obwald	Sarnen	1291
Schwytz	Schwytz	1291
Uri	Altdorf	1291
Lucerne	Lucerne	1332
Zurich	Zurich	1351
Glaris	Glaris	1352
Zoug	Zoug	1352
Berne	Berne	1353
Fribourg	Fribourg	1481
Soleure	Soleure	1481
Bale-Campagne	Liestal	1501
Bale-Ville	Bale	1501
Schaffhouse	Schaffhouse	1501
Appenzell R.E.	Herisau	1513
Appenzell R.I.	Appenzell	1513
Argovie	Aarau	1803
Grisons	Coire	1803
Saint-Gall	Saint-Gall	1803
Tessin	Bellinzone	1803
Thurgovie	Frauenfeld	1803
Vaud	Lausanne	1803
Geneve	Geneve	1815
Neuchatel	Neuchatel	1815
Valais	Sion	1815
Jura	Delemont	1979

26 ligne(s) sélectionnée(s).

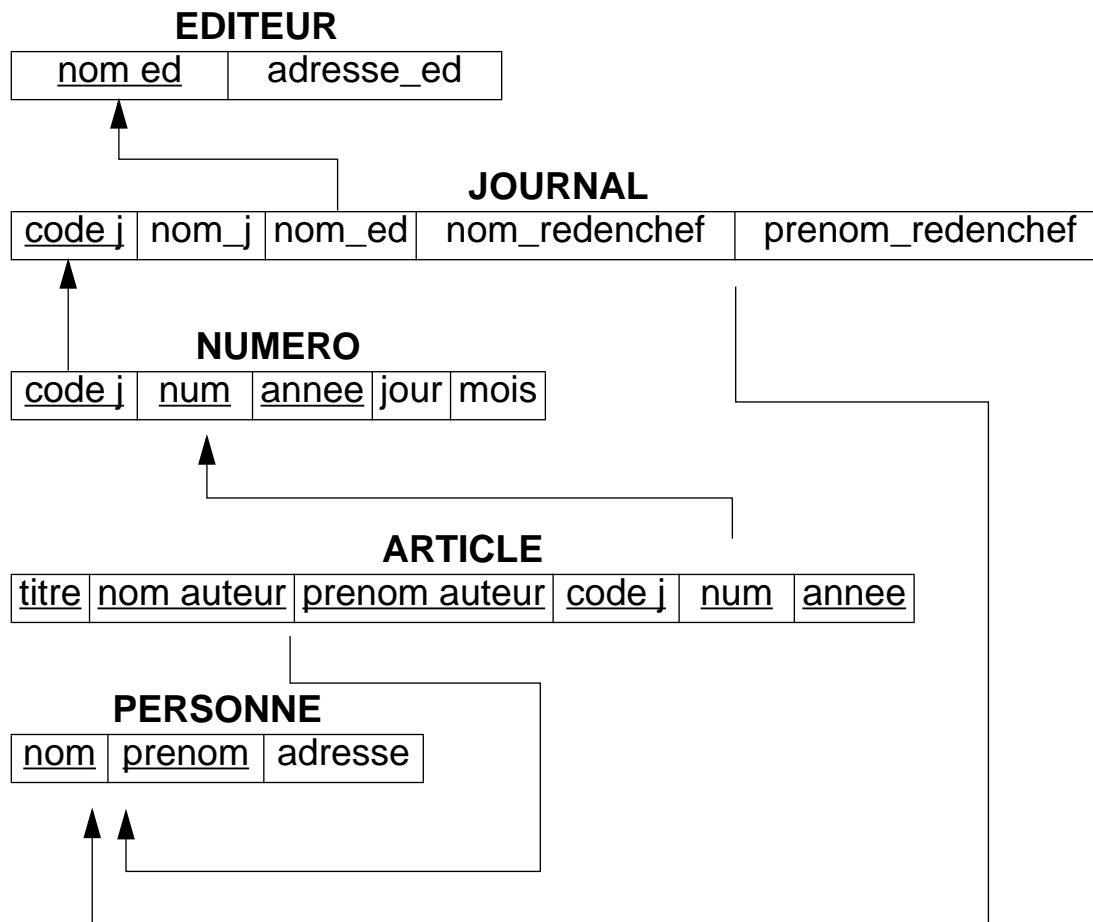
# Modèle d'exécution complet de la machine SQL

Modèle d'exécution complet (tiré de "Modèle relationnel et SQL, théorie et pratique", J. Guyot)



## Définition des schémas de relation en SQL

Soit le schéma relationnel de la bd "articles de journaux":



En SQL:

```
create table editeur (nom_ed char(28),  
                    adresse_ed char(16));
```

```
create table journal (code_j char(3), nom_j char(20),  
                    nom_ed char(28), nom_redenchef char(16),  
                    prenom_redenchef char(16));
```

## Définition des schémas de relation en SQL (suite)

```
create table article (titre varchar2(50),  
                        nom_auteur char(16),  
                        prenom_auteur char(16),  
                        code_j char(3),  
                        num char(6), annee number(4));
```

```
create table personne (nom char(16), prenom char(16),  
                        adresse char(16));
```

```
create table numero (code_j char(3), num char(6),  
                     annee number(4), jour number(2),  
                     mois char(3));
```

### *Interrogation du dictionnaire de SQL*

```
SQL> select table_name, column_name, data_type,  
           data_length, data_precision, data_scale  
from user_tab_columns where table_name='NUMERO'
```

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION	DATA_SCALE
NUMERO	CODE_J	VARCHAR2	3		
NUMERO	NUM	VARCHAR2	6		
NUMERO	ANNEE	NUMBER	22	4	0
NUMERO	JOUR	NUMBER	22	2	0
NUMERO	MOIS	VARCHAR2	3		



## Définition des schémas de relation en SQL (suite)

Soit le schéma de la bd "les grands crus de France":

### CRU

<u>nom_cru</u>	commune	region	coul
----------------	---------	--------	------

### VINS1

<u>region</u>	<u>coul</u>	<u>millesime</u>	qualite
---------------	-------------	------------------	---------

### CEPAGE\_REGION

cepage	<u>r_prod</u>	<u>coul</u>
--------	---------------	-------------

*Définition des schémas de relation en SQL:*

```
create table cru (nom_cru char(20),  
                 commune char(16),  
                 region char(16),  
                 coul char(5));
```

```
create table vins1 (region char(16),  
                   coul char(5),  
                   millesime number(4),  
                   qualite char(1));
```

```
create table cepage_region (cepage char(20),  
                            r_prod char(16),  
                            coul char(5));
```

## Insertion, modif et suppression des données en SQL

*Insertion d'un tuple (exemple):*

### avant l'insertion

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
Ch. Myrat	Barsac	Bordeaux	blanc
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge
Les Perrieres	Meursault	Bourgogne	blanc
Les Charmes	Meursault	Bourgogne	blanc
La Grappe d'Or	Meursault	Bourgogne	blanc

14 ligne(s) sélectionnée(s).

**insert into cru (nom\_cru, commune, région, coul) values ('Sous-le-Dos-d'Ane', 'Meursault', 'Bourgogne', 'rouge');**

1 ligne créée.

### après l'insertion

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
Ch. Myrat	Barsac	Bordeaux	blanc
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge
Les Perrieres	Meursault	Bourgogne	blanc
Les Charmes	Meursault	Bourgogne	blanc
La Grappe d'Or	Meursault	Bourgogne	blanc
<b>Sous-le-Dos-d'Ane</b>	<b>Meursault</b>	<b>Bourgogne</b>	<b>rouge</b>

15 ligne(s) sélectionnée(s).

## Modification des données en SQL

### Modification d'un tuple (exemple):

#### avant la modification

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
Ch. Myrat	Barsac	Bordeaux	blanc
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge
Les Perrieres	Meursault	Bourgogne	blanc
Les Charmes	Meursault	Bourgogne	blanc
La Grappe d'Or	Meursault	Bourgogne	blanc
Sous-le-Dos-d'Ane	Meursault	Bourgogne	rouge

15 ligne(s) sélectionnée(s).

```
SQL> update cru set nom_cru='Goutte d'Or'  
      where nom_cru='La Grappe d'Or';
```

1 ligne mise à jour.

#### après la modification

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
Ch. Myrat	Barsac	Bordeaux	blanc
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge
Les Perrieres	Meursault	Bourgogne	blanc
Les Charmes	Meursault	Bourgogne	blanc
<b>Goutte d'Or</b>	Meursault	Bourgogne	blanc
Sous-le-Dos-d'Ane	Meursault	Bourgogne	rouge

15 ligne(s) sélectionnée(s).

## Suppression des données en SQL

### Suppression d'un tuple (exemple):

#### avant la suppression

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
Ch. Myrat	Barsac	Bordeaux	blanc
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge
Les Perrieres	Meursault	Bourgogne	blanc
<b>Les Charmes</b>	<b>Meursault</b>	<b>Bourgogne</b>	<b>blanc</b>
Goutte d'Or	Meursault	Bourgogne	blanc
Sous-le-Dos-d'Ane	Meursault	Bourgogne	rouge

15 ligne(s) sélectionnée(s).

**SQL> delete from cru where nom\_cru='Les Charmes';**

1 ligne supprimée.

#### après la suppression

NOM_CRU	COMMUNE	REGION	COUL
Ch. Margaux	Margaux	Bordeaux	rouge
Ch. Rausan-Segla	Margaux	Bordeaux	rouge
Ch. Latour	Pauillac	Bordeaux	rouge
Ch. Lynch-Bages	Pauillac	Bordeaux	rouge
Ch. Lagrange	St. Julien	Bordeaux	rouge
Ch. d'Yquem	Sauternes	Bordeaux	blanc
Ch. Myrat	Barsac	Bordeaux	blanc
Clos Vougeot	Vougeot	Bourgogne	rouge
Corton	Aloxe-Corton	Bourgogne	rouge
Les Epenots	Pommard	Bourgogne	rouge
Les Gravieres	Santenay	Bourgogne	rouge
Les Perrieres	Meursault	Bourgogne	blanc
Goutte d'Or	Meursault	Bourgogne	blanc
Sous-le-Dos-d'Ane	Meursault	Bourgogne	rouge

14 ligne(s) sélectionnée(s).

*(Pour en savoir plus → A-38 à A-44)*

## La définition des contraintes d'intégrité en SQL

“Une contrainte d'intégrité est une règle qui restreint les valeurs possibles pour une ou plusieurs colonnes dans une table”.

Nous ne verrons que la définition des contraintes de clé (*primary key*) et des contraintes de référence (*foreign key*) qui correspondent aux trois règles d'intégrité minimum du modèle relationnel (voir pp. 123-125)

### *Définition d'une clé primaire (primary key)*

- Définition de la clé d'une table -> unicité des valeurs et valeurs non nulles
- Cette clé pourra être référencées dans une contrainte de référence.

*Exemple: base de données “articles de journaux” (pour le schéma, voir p. 178)*

```
alter table editeur add  
(constraint pk_editeur primary key (nom_ed));
```

```
alter table journal add  
(constraint pk_journal primary key (code_j));
```

```
alter table numero add  
(constraint pk_numero primary key (code_j, num,  
annee));
```

## Définition des contraintes d'intégrité en SQL (suite)

```
alter table personne add  
(constraint pk_personne primary key (nom, prenom));
```

*Exemple de violation d'une contrainte de clé:*

```
SQL> insert into personne(nom,prenom) values  
('Monnier','Claude');
```

1 ligne créée.

```
SQL> insert into personne(nom,prenom) values  
('Monnier','Claude');
```

```
insert into personne(nom,prenom)values ('Monnier','Claude')
```

```
*  
ERREUR à la ligne 1:  
ORA-00001: Présence d'une clé dupliquée dans l'index
```

## La définition des contraintes de référence

“Une contrainte de référence désigne une colonne ou une combinaison de colonnes comme clé externe (*foreign key*) et établit une association entre cette clé externe et la clé primaire de la table référencée”.

### *Définition d'une clé externe (foreign key)*

- la table qui contient la clé externe est appelée table *enfant*
- la table qui contient la clé primaire référencée est appelée table *parent*
- contrainte de référence -> la valeur de la clé externe de chaque tuple de la table enfant doit exister comme valeur de clé d'un tuple de la table parent (c-à-d que chaque tuple de la table enfant doit faire référence à un tuple qui *existe* dans la table parent)

### *Exemples:*

**alter table journal add**

**(constraint fk\_ed\_journal foreign key(nom\_ed)**

**references editeur,**

**constraint fk\_red\_journal foreign key(nom\_redenchef,  
prenom\_redenchef)references personne(nom,prenom))**

**alter table numero add**

**(constraint fk\_numero foreign key (code\_j) references  
journal);**

## La définition des contraintes de référence (suite)

```
alter table article add  
(constraint fk_n_article foreign key(code_j, num,  
annee) references numero,  
constraint fk_aut_article foreign key(nom_auteur,  
prenom_auteur)references personne(nom,prenom));
```

*Exemple de violation d'une contrainte de référence:*

```
SQL> insert into article (titre, nom_auteur,  
prenom_auteur, code_j,num,annee)  
values('La SBS n'aime pas les extraterrestres', 'Genoud',  
'Madeleine', 'LAS', '11', 1994);
```

```
insert into article (titre,nom_auteur,prenom_auteur,code_j,num,annee)  
*  
ERREUR à la ligne 1:  
ORA-02291: violation de contrainte NERIMA.FK_AUT_ARTICLE d'intégrité  
- touche parent introuvable
```

il y a violation de la contrainte de référence  
fk\_aut\_article car 'Genoud Madeleine' n'existe pas dans  
dans la table PERSONNE



## Contraintes d'intégrité et dictionnaire SQL

*Interrogation du dictionnaire de SQL:*

Q: "Quelles sont les contraintes d'intégrité qui concernent la table JOURNAL?"

```
SQL> select owner, constraint_name, constraint_type,
table_name, r_owner, r_constraint_name, status
from user_constraints where table_name='JOURNAL';
```

OWNER	CONSTRAINT_NAME	C	TABLE_NAME	R_OWNER	R_CONSTRAINT_NAME	STATUS
NERIMA	PK_JOURNAL	P	JOURNAL			ENABLED
NERIMA	FK_ED_JOURNAL	R	JOURNAL	NERIMA	PK_EDITEUR	ENABLED
NERIMA	FK_RED_JOURNAL	R	JOURNAL	NERIMA	PK_PERSONNE	ENABLED

## Définition de vue en SQL

- “Une vue est une *table logique* basée sur une ou plusieurs tables ou vues”
- On définit une vue en spécifiant une interrogation

Exemple: la vue “les cantons romands”

```
SQL> create view canton_romand as
      select canton.* from canton, langue where
      canton.nom_canton=langue.nom_canton
      and langue_parlee='français';
```

Vue créée.

Exemples d'interrogation:

Q: “Tous les cantons romands”

```
SQL> select * from canton_romand;
```

NOM_CANTON	CHEF_LIEU	DATE_ENTREE
Berne	Berne	1353
Fribourg	Fribourg	1481
Geneve	Geneve	1815
Jura	Delimont	1979
Neuchatel	Neuchatel	1815
Valais	Sion	1815
Vaud	Lausanne	1803

7 ligne(s) sélectionnée(s).

## Définition de vue en SQL (suite)

Q: “Quels sont les cantons romands entrés dans la confédération au 19ème siècle?”

```
SQL> select * from canton_romand
where date_entree between 1800 and 1899
order by date_entree;
```

NOM_CANTON	CHEF_LIEU	DATE_ENTREE
Vaud	Lausanne	1803
Geneve	Geneve	1815
Neuchatel	Neuchatel	1815
Valais	Sion	1815

### *Remarques*

- en interrogation, une vue se comporte exactement comme une table
- il n'est par contre pas possible d'insérer, de modifier ou de supprimer un tuple dans une vue

## SQL: une histoire d'amour...

On se propose de donner toutes les variantes en français de la célèbre phrase tirée du Bourgeois Gentilhomme "Belle marquise, vos beaux yeux me font mourir d'amour."

Solution:

1° on découpe la phrase en cinq parties indivisibles que l'on insère dans une table (à une seule colonne) appelée "marquise":

```
marquise P
-----
belle marquise
vos beaux yeux
me font
mourir
d'amour
```

2° on produit toutes les permutations possibles en joignant 5 fois la table "marquise" avec elle-même et en éliminant les tuples dont deux colonnes ont même valeur:

```
SQL> select * from marquise m1, marquise m2,
marquise m3, marquise m4,marquise m5
where m2.p <> m1.p and m3.p not in (m1.p,m2.p)
and m4.p not in (m1.p,m2.p,m3.p)
and m5.p not in (m1.p,m2.p,m3.p,m4.p);
```

P	P	P	P	P
-----	-----	-----	-----	-----
d'amour	mourir	me font	vos beaux yeux	belle marquise
mourir	d'amour	me font	vos beaux yeux	belle marquise
d'amour	me font	mourir	vos beaux yeux	belle marquise
me font	d'amour	mourir	vos beaux yeux	belle marquise
mourir	me font	d'amour	vos beaux yeux	belle marquise
me font	mourir	d'amour	vos beaux yeux	belle marquise

d'amour	mourir	vos beaux yeux	me font	belle marquise
mourir	d'amour	vos beaux yeux	me font	belle marquise
d'amour	vos beaux yeux	mourir	me font	belle marquise
vos beaux yeux	d'amour	mourir	me font	belle marquise
mourir	vos beaux yeux	d'amour	me font	belle marquise
vos beaux yeux	mourir	d'amour	me font	belle marquise
d'amour	me font	vos beaux yeux	mourir	belle marquise
me font	d'amour	vos beaux yeux	mourir	belle marquise
d'amour	vos beaux yeux	me font	mourir	belle marquise
vos beaux yeux	d'amour	me font	mourir	belle marquise
me font	vos beaux yeux	d'amour	mourir	belle marquise
vos beaux yeux	me font	d'amour	mourir	belle marquise
mourir	me font	vos beaux yeux	d'amour	belle marquise
me font	mourir	vos beaux yeux	d'amour	belle marquise
mourir	vos beaux yeux	me font	d'amour	belle marquise
vos beaux yeux	mourir	me font	d'amour	belle marquise
me font	vos beaux yeux	mourir	d'amour	belle marquise
vos beaux yeux	me font	mourir	d'amour	belle marquise
d'amour	mourir	me font	belle marquise	vos beaux yeux
mourir	d'amour	me font	belle marquise	vos beaux yeux
d'amour	me font	mourir	belle marquise	vos beaux yeux
me font	d'amour	mourir	belle marquise	vos beaux yeux
mourir	me font	d'amour	belle marquise	vos beaux yeux
me font	mourir	d'amour	belle marquise	vos beaux yeux
d'amour	mourir	belle marquise	me font	vos beaux yeux
mourir	d'amour	belle marquise	me font	vos beaux yeux
d'amour	belle marquise	mourir	me font	vos beaux yeux
belle marquise	d'amour	mourir	me font	vos beaux yeux
mourir	belle marquise	d'amour	me font	vos beaux yeux
belle marquise	mourir	d'amour	me font	vos beaux yeux
d'amour	me font	belle marquise	mourir	vos beaux yeux
me font	d'amour	belle marquise	mourir	vos beaux yeux
d'amour	belle marquise	me font	mourir	vos beaux yeux
belle marquise	d'amour	me font	mourir	vos beaux yeux
me font	belle marquise	d'amour	mourir	vos beaux yeux
belle marquise	me font	d'amour	mourir	vos beaux yeux
mourir	me font	belle marquise	d'amour	vos beaux yeux
me font	mourir	belle marquise	d'amour	vos beaux yeux
mourir	belle marquise	me font	d'amour	vos beaux yeux
belle marquise	mourir	me font	d'amour	vos beaux yeux
me font	belle marquise	mourir	d'amour	vos beaux yeux
belle marquise	me font	mourir	d'amour	vos beaux yeux
d'amour	mourir	vos beaux yeux	belle marquise	me font
mourir	d'amour	vos beaux yeux	belle marquise	me font
d'amour	vos beaux yeux	mourir	belle marquise	me font
vos beaux yeux	d'amour	mourir	belle marquise	me font
mourir	vos beaux yeux	d'amour	belle marquise	me font
vos beaux yeux	mourir	d'amour	belle marquise	me font
d'amour	mourir	belle marquise	vos beaux yeux	me font
mourir	d'amour	belle marquise	vos beaux yeux	me font
d'amour	belle marquise	mourir	vos beaux yeux	me font
belle marquise	d'amour	mourir	vos beaux yeux	me font
mourir	belle marquise	d'amour	vos beaux yeux	me font
belle marquise	mourir	d'amour	vos beaux yeux	me font
d'amour	vos beaux yeux	belle marquise	mourir	me font
vos beaux yeux	d'amour	belle marquise	mourir	me font
d'amour	belle marquise	vos beaux yeux	mourir	me font
belle marquise	d'amour	vos beaux yeux	mourir	me font
vos beaux yeux	belle marquise	d'amour	mourir	me font
belle marquise	vos beaux yeux	d'amour	mourir	me font
mourir	vos beaux yeux	belle marquise	d'amour	me font

vos beaux yeux mourir	belle marquise d'amour	me font
mourir	belle marquise vos beaux yeux d'amour	me font
belle marquise mourir	vos beaux yeux d'amour	me font
vos beaux yeux belle marquise mourir	d'amour	me font
belle marquise vos beaux yeux mourir	d'amour	me font
d'amour	me font	vos beaux yeux belle marquise mourir
me font	d'amour	vos beaux yeux belle marquise mourir
d'amour	vos beaux yeux me font	belle marquise mourir
vos beaux yeux d'amour	me font	belle marquise mourir
me font	vos beaux yeux d'amour	belle marquise mourir
vos beaux yeux me font	d'amour	belle marquise mourir
d'amour	me font	belle marquise vos beaux yeux mourir
me font	d'amour	belle marquise vos beaux yeux mourir
d'amour	belle marquise me font	vos beaux yeux mourir
belle marquise d'amour	me font	vos beaux yeux mourir
me font	belle marquise d'amour	vos beaux yeux mourir
belle marquise me font	d'amour	vos beaux yeux mourir
d'amour	vos beaux yeux belle marquise me font	mourir
vos beaux yeux d'amour	belle marquise me font	mourir
d'amour	belle marquise vos beaux yeux me font	mourir
belle marquise d'amour	vos beaux yeux me font	mourir
vos beaux yeux belle marquise d'amour	me font	mourir
belle marquise vos beaux yeux d'amour	me font	mourir
me font	vos beaux yeux belle marquise d'amour	mourir
vos beaux yeux me font	belle marquise d'amour	mourir
me font	belle marquise vos beaux yeux d'amour	mourir
belle marquise me font	vos beaux yeux d'amour	mourir
vos beaux yeux mourir	me font	mourir
me font	mourir	vos beaux yeux belle marquise d'amour
mourir	vos beaux yeux me font	vos beaux yeux belle marquise d'amour
vos beaux yeux mourir	me font	belle marquise d'amour
me font	vos beaux yeux mourir	belle marquise d'amour
vos beaux yeux me font	mourir	belle marquise d'amour
mourir	me font	belle marquise vos beaux yeux d'amour
me font	belle marquise vos beaux yeux d'amour	vos beaux yeux d'amour
mourir	belle marquise me font	vos beaux yeux d'amour
belle marquise mourir	me font	vos beaux yeux d'amour
me font	belle marquise mourir	vos beaux yeux d'amour
belle marquise me font	mourir	vos beaux yeux d'amour
mourir	vos beaux yeux belle marquise me font	d'amour
vos beaux yeux mourir	belle marquise me font	d'amour
mourir	belle marquise vos beaux yeux me font	d'amour
belle marquise mourir	vos beaux yeux me font	d'amour
vos beaux yeux belle marquise mourir	me font	d'amour
belle marquise vos beaux yeux mourir	me font	d'amour
me font	vos beaux yeux belle marquise mourir	d'amour
vos beaux yeux me font	belle marquise mourir	d'amour
me font	belle marquise vos beaux yeux mourir	d'amour
belle marquise me font	vos beaux yeux mourir	d'amour
vos beaux yeux mourir	me font	d'amour
belle marquise vos beaux yeux me font	mourir	d'amour

120 ligne(s) sélectionnées