



# SQL

## Structured Query Language

*comment interroger et mettre à jour des données dans une VRAIE base de données ?*



Département  
INFormatique



# Plan du document

- **Introduction** **slide 87**
- **BD Exemple : les vins** **slide 91**
- **Manipulation des données** **slide 93**
  - **Recherche** **slide 94**
  - **Mises à jour** **slide 121**
- **Traitement d'une requête** **slide 125**

# Langages de requêtes

- **Langage de requêtes**
  - Langage de Définition de Données (LDD)
  - Langage de Manipulation de Données (LMD)
- **Langages de manipulation formels**
  - Algèbre relationnelle
  - Calcul relationnel (basé sur la logique des prédicats)
- **Langages de requêtes orientés utilisateur**
  - Structured Query Language (SQL)
  - QUery Language (QUEL)
  - Query By Example (QBE)
- **Lien avec les langages de programmation**
  - Immersion d'un langage de requêtes dans un langage de programmation : approche Embedded SQL (Java, Pascal, C, ...)

# Présentation de SQL (1)

- **Fonctionnalités**
  - Définition et manipulation de données au format relationnel
  - Contrôle des données
- **Le langage de manipulation**
  - Déclaratif, non procédural
  - Emprunté à l'algèbre relationnelle et au calcul relationnel de tuples
- **Puissance du langage de manipulation**

Algèbre Relationnelle

+

Fonctions-Agrégats

+

Tri



Requête SQL (sans fonctions et tri)

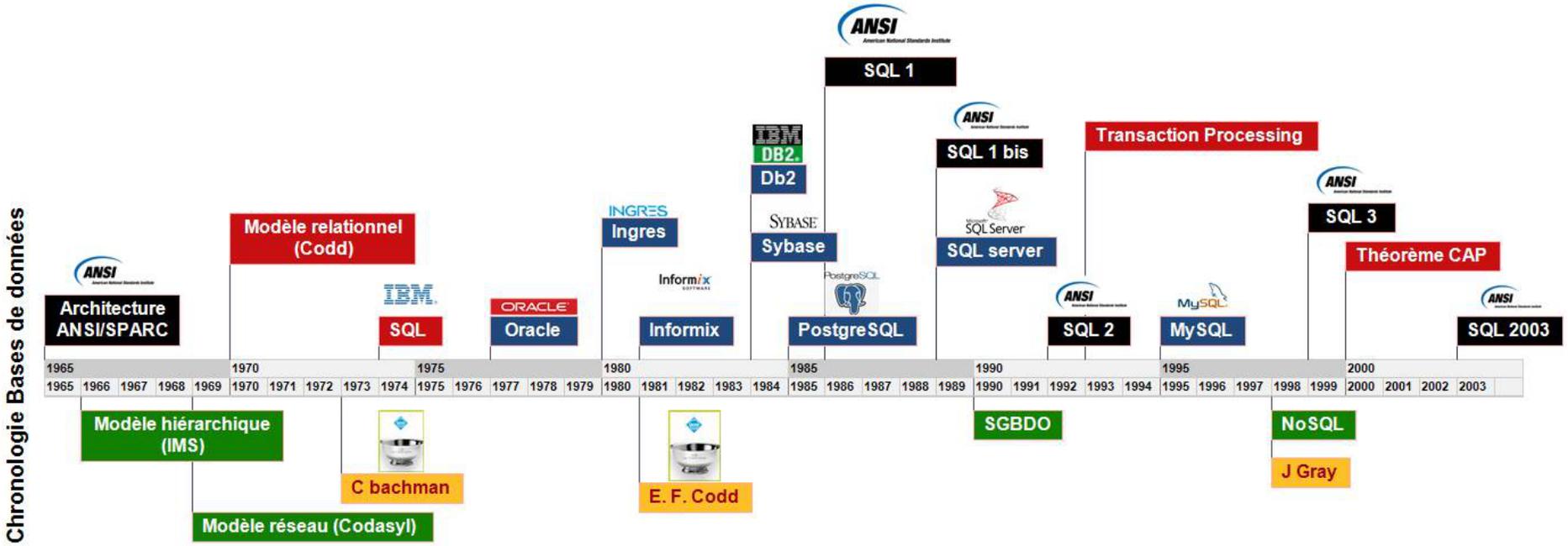


Composition d'opérations de l'algèbre relationnelle

# Présentation de SQL (2)

**Légende**

- Alternatives
- Théorie BD
- Produits SGBD
- Prix Turing
- Normalisation



# Plan du document

- Introduction
- **BD Exemple : les vins**
- Manipulation des données
  - Recherche
  - Mises à jour
- Traitement d'une requête

## BD Exemple : les vins



Vins( <u>num</u> , cru, annee, degre)	V
Recoltes( <u>nvin</u> , nprod, quantite)	R
Producteurs( <u>num</u> , nom, prenom, region)	
P	
Clients( <u>num</u> , nom, prenom, ville)	
Cl	
Commandes( <u>ncde</u> , date, ncli, nvin, qte)	C
Livraisons( <u>ncde</u> , no_ordre, qteLivree)	L

# Plan du document

- Introduction
- BD Exemple : les vins
- Manipulation des données
  - Recherche
    - Syntaxe générale de recherche
    - Restriction et projection
    - Jointures
    - Opérateurs ensemblistes
    - Fonctions agrégats
    - Partitionnement
    - Prédicats et division
    - Synthèse
    - Exemple complet
  - Mises à jour
- Traitement d'une requête
- Et moi que dois-je faire ?

# Syntaxe générale de recherche

Syntaxe	
<b>SELECT</b>	<liste d'attributs projetés>
<b>FROM</b>	<liste de relations>
[WHERE et de	<liste des critères de restriction jointure>]

Comment remplir les clauses ?



- ③ Quel résultat souhaite voir l'utilisateur (schéma du résultat) ?
- ① Où sont les attributs dont j'ai besoin ?
- ② Y-a t-il
  - des conditions sur les valeurs d'attributs exprimées dans ma requête ?
  - plusieurs relations dans ma clause FROM ? Si oui, quelles sont les conditions de jointure(s) ?

## Restriction ( $\sigma$ ) et projection ( $\Pi$ )

- *"Donner les vins de cru Pommard"*

```
SELECT num, annee, degre
```

```
FROM Vins
```

```
WHERE cru = 'Pommard'
```

Valeur des chaînes de caractères délimitées par des “

NUM	ANNEE	DEGRE
5	1976	11.70
23	1972	12.00

Nombre de tuples accédés : 2

## Projection ( $\Pi$ )

- *"Donner tous les vins"*

```
SELECT *
```

```
FROM Vins ;
```

- *"Donner la liste de tous les crus, avec élimination des doublons"*

```
SELECT DISTINCT cru
```

```
FROM Vins;
```

## Restriction

- *"Donner les vins de degré compris entre 8 et 12"*

```
SELECT *  
FROM Vins  
WHERE degre >=8 AND degre <=12 ;
```

```
SELECT *  
FROM Vins  
WHERE degre BETWEEN 8 AND 12 ;
```

```
SELECT *  
FROM Vins  
WHERE degre IN (8, 9, 10, 11, 12);
```

## Restriction et tri

- *"Donner les vins dont le cru commence par p ou P"*

```
SELECT *
```

```
FROM Vins
```

```
WHERE cru LIKE 'p%' OR cru LIKE 'P%';
```

- *"Donner les crus des vins de millésime 1995 et de degré 12, triés par ordre croissant"*

```
SELECT cru
```

```
FROM Vins
```

```
WHERE annee=1995 AND degre = 12
```

```
ORDER BY cru ;
```

# Jointure

- *"Donner les noms des producteurs de Pommard"*

```
SELECT nom
FROM Vins V, Recoltes R, Producteurs P
WHERE V.num = R.nvin
AND R.nprod = P.num
AND cru = 'Pommard';
```

Produit  
cartésien

Conditions  
de jointures

Nom d'attribut  
non ambigu

## Jointure (syntaxe SQL2)

- Syntaxe plus proche de l'algèbre relationnelle (directement exprimée dans le FROM)
- Supportée dans plusieurs SGBD ( $\geq$  Oracle 9, MySQL, SQLServer, ...)

```
SELECT nom
FROM Vins V JOIN Recoltes R ON (V.num = R.nvin)
      JOIN Producteurs P ON (R.nprod=P.num)
WHERE cru = 'Pommard' ;
```

- Jointure « naturelle »
  - L'égalité sur les attributs de même nom peut être remplacée par NATURAL JOIN, ou JOIN ... USING (attributs)

## Jointure « procédurale » ou ensembliste

```
SELECT nom
FROM Producteurs
WHERE num IN (
    SELECT nprod
    FROM Recoltes
    WHERE nvin IN (
        SELECT num
        FROM Vins
        WHERE cru = 'Pommard' )) ;
```

# Auto-jointure

- Jointure d'une relation avec elle-même

→ synonymes

- « *Donner les couples de producteurs produisant le même vin* »

```
SELECT P1.num, P2.num
```

```
FROM Producteurs P1, Producteurs P2, Recoltes R1,  
Recoltes R2
```

```
WHERE P1.num = R1.nprod AND P2.num = R2.nprod
```

```
AND R1.nvin = R2.nvin
```

```
AND P1.num > P2.num ;
```

# Opérateurs ensemblistes

- **Union (norme SQL1)**
  - **Élimination automatique des doublons**

```
SELECT num FROM Producteurs
UNION
SELECT num FROM Clients ;
```

- **Intersection (norme SQL2 !)**

```
SELECT num FROM Producteurs
INTERSECT
SELECT num FROM Clients ;
```

- **Différence (norme SQL2 !)**



```
SELECT num FROM Clients
EXCEPT
SELECT num FROM Producteurs ;
```

(ou MINUS)