Le langage SQL

SQL = Structured Query Language

Langage pour les bases de données relationnelles

But: manipuler et accéder aux données de la base de données

Existe sur tous les SGBD relationnels (Oracle, Access, MySQL, SQL Server, Sybase...), mais avec des extensions (+/- proche de la norme)

Historique

Origine : SEQUEL(Structured English QUEry Language) développé par IBM pour le SGBD appelé SYSTEM R (au milieu des années 70) D. Chamberlin et R.F. Boyce

Première réalisation commerciale de SQL: 1979 (Oracle Corporation)

SQL est défini par une norme ISO:
SQL - 1986 - niveau I
SQL - 1989 - niveau II
SQL 2 - 1992
SQL 3 - SQL 99 - 1999
SQL 2003
SQL 2008
SQL 2011
SQL 2016
... futur ... ?

Caractéristiques du 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

Mais nous ne verrons que l'interrogation

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

Terminologie SQL

Modèle relationnel	SQL	Access
relation	table (table)	table
tuple ou n-uplet	ligne (row)	enregistrement
attribut	colonne (column)	champ
domaine	type de donnée (data type)	type de donnée

Autre différence

les tables SQL (et Access) peuvent contenir des doublons (plusieurs lignes identiques)
-> 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

- La commande SELECT comprend trois parties:

  SELECT < liste des colonnes du résultat>

  FROM < liste des tables impliquées dans l'interrogation>

  WHERE < condition de sélection des tuples>

- Exemple: la base de données "Hôtel"

  Chambres (NumChambre, Prix, NbrLit, NbrPers, Confort, Equipement)

  Clients (NumClient, Nom, Prenom, Adresse)

  Réservation (NumChambre, NumClient, DateArr, DateDep)

- Exemple d'interrogation:

  "Quelles sont les chambres avec bain et télévision ?"

  SELECT NumChambre, Confort

  FROM Chambres

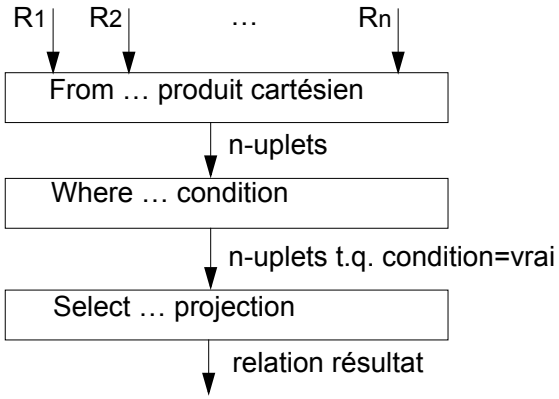
  WHERE Confort='bain' AND Equipement='TV'

La machine SQL

select A_1,A_2,...A_m from R_1, R_2,...,R_n where condition

Exécution:

- faire le produit cartésien de toutes les tables citées après from
- sélectionner les entités satisfaisant la condition décrite après where
- projeter sur les colonnes citées après select



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.

SQL: exemples

Soit les instances de cru, vins et cepage_region:

instance de la table 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 la table vins:

REGION	COUL	MILLESIME	QUALITE
Bordeaux	rouge	2007	moyenne
Bordeaux	blanc	2007	excellente
Bourgogne	rouge	2007	bonne
Bourgogne	blanc	2007	excellente
Bordeaux	rouge	2008	excellente
Bordeaux	blanc	2008	très bonne
Bourgogne	rouge	2008	bonne
Bourgogne	blanc	2008	très bonne

instance de la table cepage_region:

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

SQL: exemples d'interrogation

- Schéma:

  cru (nom_cru, commune, region, coul)

  vins (region, coul, millesime, qualite)

  cepage_region (cepage, r_prod, coul)

- Interrogations:

  Q1 : "Tous les crus"

  Q2: "Tous les crus rouges"

  Q3: "La liste des noms de crus rouges"

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

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

  Q6 : "Quels sonts les crus rouges et leurs millésimes qui sont de bonne qualité?"

  Q7: "Quels sont les millésimes où les Bordeaux rouges sont de qualité supérieure aux Bourgogne rouges?"

Interrogations SQL: solution

Q1 : “Tous les crus”

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”

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.
```


Interrogations SQL: solution (suite)

Q3: “La liste des noms de crus rouges”

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 ?”

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

Interrogations SQL: solution (suite)

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

```
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
----------
      2008
```

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

```
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');
```


Interrogations SQL: solution (de plus en plus difficile)

```
NOM_CRU              COMMUNE          REGION          COUL   MILLESIME
-------------------- ---------------- --------------- ----- ----------
Ch. Margaux          Margaux          Bordeaux        rouge       2008
Ch. Rausan-Segla     Margaux          Bordeaux        rouge       2008
Ch. Latour           Pauillac         Bordeaux        rouge       2008
Ch. Lynch-Bages      Pauillac         Bordeaux        rouge       2008
Ch. Lagrange         St. Julien       Bordeaux        rouge       2008
Clos Vougeot         Vougeot          Bourgogne       rouge       2007
Corton               Aloxe-Corton     Bourgogne       rouge       2007
Les Epenots          Pommard          Bourgogne       rouge       2007
Les Gravieres        Santenay         Bourgogne       rouge       2007
Clos Vougeot         Vougeot          Bourgogne       rouge       2008
Corton               Aloxe-Corton     Bourgogne       rouge       2008
Les Epenots          Pommard          Bourgogne       rouge       2008
Les Gravieres        Santenay         Bourgogne       rouge       2008

13 rows selected.
```

Q7: “Quels sont les millésimes où les Bordeaux rouges sont de qualité supérieure aux Bourgognes rouges?”

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

 MILLESIME
----------
      2008
```

Remarque: pour que cette interrogation SQL donne le résultat escompté, il faudrait que le type de données de la colonne “qualite” soit numérique (voir vins1 plus loin).

De l'algèbre relationnelle à l'interrogation en SQL

La relation R

R ≡ select ∗ from R

La projection $\pi_{Au,Av,...Az}(R)$

$\pi_{Au,Av,...Az}(R)$ ≡ select Au, Av, ..., Az from R

La sélection $\sigma_F(R)$

$\sigma_F(R)$ ≡ select ∗ from R where F

Le produit cartésien R × S

R × S ≡ select R.∗, S.∗ from R, S

La jointure $R_{(Ai_1\ \theta_1\ Bj_1)\ \wedge\ (Ai_2\ \theta_2\ Bj_2)\ \wedge ...\ \wedge\ (Ai_m\ \theta_m\ Bj_m)} \bowtie S$

$R_{(Ai_1\ \theta_1\ Bj_1)\ \wedge\ (Ai_2\ \theta_2\ Bj_2)\ \wedge ...\ \wedge\ (Ai_m\ \theta_m\ Bj_m)} \bowtie S$ ≡

select R.∗, S.∗ from R, S

where R.Ai_1 θ_1 S.Bj_1

and R.Ai_2 θ_2 S.Bj_2

...

and R.Ai_m θ_m S.Bj_m

où θ_i est un opérateur de comparaison (=, <, ≤, >, ≥, ≠)

notation SQL: =, <, <=, >, >=, <> (noté aussi != et ^=)

De l'algèbre relationnelle à SQL (suite)

La jointure naturelle $R \bowtie S$

si les attributs de jointure de R et S sont Au, Av,...,Az

et les attributs propres de S sont Bp, Bq,..., Bt

$R \bowtie S$ ≡ select R.∗, Bp, Bq,..., Bt from R, S

where R.Au = S.Au

and R.Av = S.Av

...

and R.Az = S.Az

L'union R ∪ S

R ∪ S ≡ select ∗ from R union select ∗ from S

L'intersection R ∩ S

R ∩ S ≡ select ∗ from R intersect select ∗ from S

La différence R − S

R − S ≡ select ∗ from R minus select ∗ from S

De l'algèbre relationnelle à SQL: cas général

D'une manière générale, la requête SQL

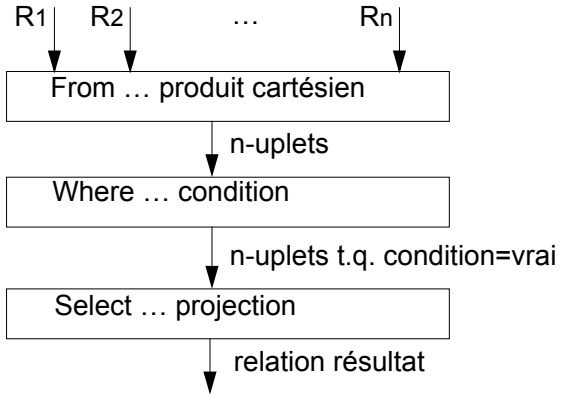
select $A_1,A_2,...A_m$ from $R_1, R_2,...,R_n$ where F

est équivalente à

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

C'est cette équivalence qui nous donne le modèle d'exécution simplifié de la machine SQL de la page 189.

Rappel:



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'aggrégation
  - 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

Instance de la table vins1

Pour faciliter l'écriture des conditions sur la qualité du vin

- Instance de la table vins1:

REGION	COUL	MILLESIME	QUALITE
Bordeaux	rouge	2002	4
Bordeaux	blanc	2002	5
Bourgogne	rouge	2002	6
Bourgogne	blanc	2002	7
Bordeaux	rouge	2003	5
Bordeaux	blanc	2003	5
Bourgogne	rouge	2003	5
Bourgogne	blanc	2003	4
Bordeaux	rouge	2004	6
Bordeaux	blanc	2004	4
Bourgogne	rouge	2004	4
Bourgogne	blanc	2004	6
Bordeaux	rouge	2005	7
Bordeaux	blanc	2005	6
Bourgogne	rouge	2005	7
Bourgogne	blanc	2005	7
Bordeaux	rouge	2006	5
Bordeaux	blanc	2006	6
Bourgogne	rouge	2006	6
Bourgogne	blanc	2006	7
Bordeaux	rouge	2007	3
Bordeaux	blanc	2007	6
Bourgogne	rouge	2007	4
Bourgogne	blanc	2007	6
Bordeaux	rouge	2008	6
Bordeaux	blanc	2008	5
Bourgogne	rouge	2008	4
Bourgogne	blanc	2008	5

Source: The Economist Slimline Diary 2011, The International Wine & Food Society, 2009

Les colonnes du résultat (dispayed 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 de bourgogne 2007"

select $*$ from cru, vins
where cru.region=vins.region and cru.coul=vins.coul
and region=bourgogne and millesime=2007

```
NOM_CRU        COMMUNE      REGION    COUL  REGION    COUL  MILL QUALITE
-------------- ------------ --------- ----- --------- ----- ---- --------
Les Perrieres  Meursault    Bourgogne blanc Bourgogne blanc 2007 excellente
Les Charmes    Meursault    Bourgogne blanc Bourgogne blanc 2007 excellente
La Grappe d'Or Meursault    Bourgogne blanc Bourgogne blanc 2007 excellente
Clos Vougeot   Vougeot      Bourgogne rouge Bourgogne rouge 2007 bonne
Corton         Aloxe-Corton Bourgogne rouge Bourgogne rouge 2007 bonne
Les Epenots    Pommard      Bourgogne rouge Bourgogne rouge 2007 bonne
Les Gravieres  Santenay     Bourgogne rouge Bourgogne rouge 2007 bonne
```

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 sonts 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’aggrégation

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'acceuil de l'hôtel”

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

```
SUM(NBR_PERS)
-------------
           56
```

Les alias dans les colonnes du résultat

Alias pour une colonne

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

Q9a:
select sum(nbr_pers) as [capacité de l'hôtel]
from chambres

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

ou pour expliciter une colonne:

Q10: "les qualités du millésime 2005"

select region,coul,millesime,qualite as [QUALITE (1 à 7)]
from vins1 where millesime=2005;

```
REGION          COUL   MILLESIME QUALITE(1 à 7)
--------------- ----- ---------- --------------
Bordeaux        rouge        2005              7
Bordeaux        blanc        2005              6
Bourgogne       rouge        2005              7
Bourgogne       blanc        2005              7
```


Les expressions dans les colonnes du résultat

- arithmétiques
  - opérateur +, -, *, /
  - fonctions: sin, cos, exp, ln, mod, floor(tronquer),etc.
- sur les chaînes de caractères
  - opérateur || (concaténation)
  - fonctions: concat, lower, replace, substr, etc.

Q11: "Quelle la population et la densité (population au km^2) de chaque pays"

select nom, population, round(population/surface) as densité
from pays;

```
NOM         POPULATION    DENSITÉ
----------- ---------- ----------
Suisse         7489370        181
France        60656178        111
```

Q12: "Quelles son les initiales des clients"

select nom, prenom, substr(nom,1,1) || substr(prenom,1,1)
from clients;

```
NOM         PRENOM      SU
----------- ----------- --
GASCON      GASTON      GG
DUPONT      PIERRE      DP
DUFOUR      JEAN        DJ
ZORO        DIEGO       ZD
EINSTEIN    ALBERT      EA
DUMAS       ALEXANDRE   DA
NOBODY      FRANCOISE   NF
ROMULUS     BERNADETTE  RB
AGDA        BRUNO       AB
CHADOK      AMELIE      CA

10 rows selected.
```

Les expressions dans les colonnes du résultat(suite)

- expressions avec les dates
  - opérateurs -
  - fonctions add_months, last_day, months_between, sysdate, etc.

select num_chambre, date_arr, date_dep, date_dep - date_arr as durée from reservations;

```
NUM_CHAMBR DATE_ARR  DATE_DEP       DURÉE
---------- --------- --------- ----------
11         11-JAN-90 15-JAN-90          4
21         10-JAN-90 02-MAR-90         51
34         20-DEC-89 27-DEC-89          7
44         24-DEC-89 27-DEC-89          3
45         23-DEC-89 28-DEC-89          5
14         01-DEC-89 28-DEC-89         27
23         01-DEC-89 02-DEC-89          1
23         08-DEC-89 09-DEC-89          1
23         15-DEC-89 16-DEC-89          1
23         22-DEC-89 23-DEC-89          1
23         29-DEC-89 30-DEC-89          1

11 rows selected.
```


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
----------
      2008
```

Alias pour une table
lorsqu'il faut joindre plusieurs fois la même table:

Q7: "Quels sont les millésimes où les Bordeaux rouges sont de qualité supérieure aux Bourgognes rouges?"

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

```
 MILLESIME
----------
      2004
      2008
```

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

- =, <, <=, >, >=, <>
- 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
----------
      2008
```

- 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
```


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 jointure

- Généralement, lorsque plusieurs tables sont impliquées dans l'interrogation, il faut spécifier la condition de jointure
- La condition de jointure est placée au même endroit que les autres conditions de sélection, dans la partie where
- Style d'écriture: on spécifie de préférence la condition de jointure juste après le mot réservé where
- rappel: si les mêmes noms de colonne apparaissent dans plusieurs tables, il faut les préfixer par le nom de la table (pour lever l'ambiguité)

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

```
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
```


Jointure - syntaxe alternative: INNER JOIN

- Il existe une syntaxe alternative: INNER JOIN ... ON
- Dans ce cas la condition de jointure est spécifiée dans la partie from de la requête
- Cette notation sépare condition de jointure et autres conditions de sélection
- Le résultat de la requête est identique à celui de la requête exprimée dans la syntaxe vue précédemment

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

```
SQL>  select cepage from
  cepage_region INNER JOIN cru
  ON region=r_prod and cepage_region.coul = cru.coul
  where commune='Meursault' and cru.coul='blanc';

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

La condition de sélection - interrogation imbriquée

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 au dessus de la moyenne (qualité > 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
----------
      2004
      2005
      2008
```

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 (∃), not 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            francais
Fribourg         allemand
Fribourg         francais
Geneve           francais
Glaris           allemand
Grisons          allemand
Grisons          romanche
Grisons          italien
Jura             francais
...              ...
```

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
```

La jointure externe

Problème:

- Lorsque l'on fait une jointure entre deux tables, on "perd" les tuples qui n'ont pas la même valeur dans les deux tables

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.

Q17: "Afficher 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.

- Dans ce cas on a "perdu" les cabines qui n'ont pas été réservées -> on peut remédier à cet effet indésirable en spécifiant une jointure externe

La jointure externe - syntaxe Access

- Spécification d'une jointure externe - syntaxe Access
- On utilise une syntaxe semblable à la jointure mais en spécifiant LEFT JOIN (ou RIGHT JOIN)
- LEFT (ou RIGHT) indique la table ou l'on veut préserver tous les tuples dans le résultat de la requête

Q18: "Afficher toutes les cabines et le nom du client pour les cabines qui ont été réservées".

select cabine.∗, nom_client from cabine
LEFT JOIN reservation
ON 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.

La jointure externe - syntaxe Oracle

- la(es) colonne(s) d'une des deux table est postfixée par (+)
- mnémotechnique: (+) indique la table où il faut ajouter des tuples fictifs; dans l'exemple Q18 ça revient à ajouter dans la table reservation des tuples fictifs pour toutes les cabines qui n'ont pas de réservation (avec un nom de client null):

reservation

```
NUM_CABINE NOM_CLIENT
---------- ----------
         2
         3
         4
```

Q18: "Afficher toutes les cabines et le nom du client pour les cabines qui ont été réservées".

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

Q19: "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
```

Q20: "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).
```

La clause having: la sélection après regroupement

- permet de ne sélectionner que les groupes satisfaisant une certaine condition
- opération non réalisable avec la clause where (where effectue la sélection avant le regroupement)

Q21: “La liste des cantons bilingues (ou trilingues)”

SQL> select nom_canton, count(*) from langue group by nom_canton having count(*) > 1;

```
NOM_CANTON         COUNT(1)
---------------- ----------
Berne                     2
Fribourg                  2
Grisons                   3
Valais                    2
```


Les opérations ensemblistes

opérateurs: union (∪), intersect (∩) et minus (–)

- les opérandes 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

Q22: “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
```

Attention: dans Access seul l’union est disponible

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

Q23: "les cantons suisses dans l'ordre d'entrée dans la confédération et par ordre alphabétique"

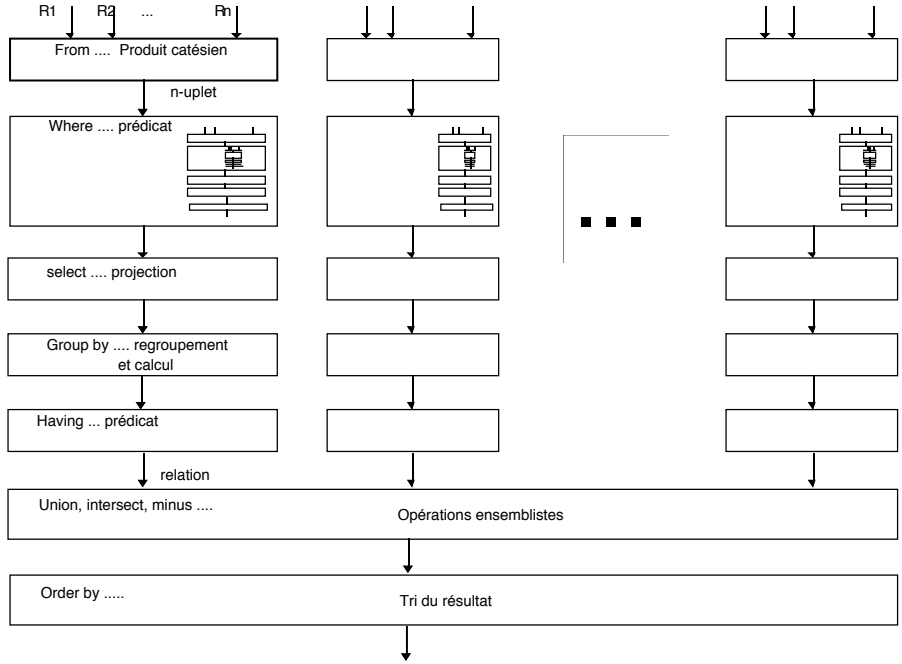
select * from canton
order by date_entree asc, nom_canton asc

```
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)



Les vues

Le concept de vue dans les bases de données

- Jusqu'à présent: une table (ou relation) est associée à une instance où les tuples sont stockés physiquement
- "Une vue est une table logique définie à partir d'autres tables (ou vues)"
- On définit une vue en spécifiant une requête d'interrogation SQL
- L'instance d'une vue est évaluée (calculée) au moment où elle est activée dans une requête

Soit la base de données des cantons suisses

relation canton

```
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).
```

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            francais
Fribourg         allemand
Fribourg         francais
Geneve           francais
Glaris           allemand
Grisons          allemand
Grisons          romanche
Grisons          italien
Jura             francais
...              ...
```


Exemple: la vue “les cantons romands”

En SQL

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

Exemples d’interrogation:

Q24: “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).
```

Vues en SQL (suite)

Q25: “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

Vues: modèle d’exécution

Comment cela fonctionne-t-il?

Les termes de la requête sont substitués lexicalement par les termes qui ont été utilisé dans la définition de la vue:

a) la table dans la clause FROM de la requête est substitué par les tables spécifiés dans la clause FROM de la vue.

b) les colonnes dans la clause SELECT de la requête sont substituées par les expressions spécifiées dans la clause SELECT de la vue.

c) la clause WHERE de la vue est ajoutée à la clause WHERE de la requête.

d) Dans les cas les plus simples, la clause GROUP BY de la vue (s’il elle a été définie) est ajoutée à la requête.

Exemple d'exécution

Soit la vue

create view canton_romand as select canton.∗ from canton, langue where canton.nom_canton = langue.nom_canton and langue_parlee='francais';

et la requête:

select ∗ from canton_romand where date_entree between 1800 and 1899 order by date_entree;

Application des règles:

a) select * from canton, langue where date_entree between 1800 and 1899 order by date_entree;

b) select canton.* from canton, langue where date_entree between 1800 and 1899 order by date_entree;

c) select canton.* from canton, langue where date_entree between 1800 and 1899 and canton.nom_canton = langue.nom_canton and langue_parlee='francais'
order by date_entree;

C'est la requête que nous aurions écrite si les vues n'existaient pas !

Vues: utilité

- Vues: multiplier les représentations logiques
- Idée centrale: créer une indépendance logique entre le schéma de la base de données et les applications qui l'utilisent

Typologie des vues

Nous étudierons 4 types de vues

- Vues ”contextuelles”
- Vues “interfaces”
- Vues “attributs calculés”
- Vues “déductives”

(1) Vues “contextuelles”

On construit une nouvelle relation (vue) à partir d’un ensemble de relations

On utilise ensuite la vue (contextuelle) au lieu des schémas sous-jacents

Exemple

```
CREATE VIEW hotel
     (NUM_CLIENT, NOM, PRENOM, ADRESSE,
      NUM_CHAMBRE, PRIX, NBR_LITS,
      NBR_PERS, CONFORT, EQUIPEMENT,
      DATE_ARR, DATE_DEP)
AS SELECT CLIENTS.NUM_CLIENT, NOM, PRENOM, ADRESSE,
        CHAMBRES.NUM_CHAMBRE, PRIX, NBR_LITS,
        NBR_PERS, CONFORT, EQUIPEMENT,
        DATE_ARR, DATE_DEP
                FROM CHAMBRES, CLIENTS, RESERVATIONS
                WHERE Clients.num_client=Reservations.num_client
                AND Chambres.num_chambre=Reservations.num_chambre
```

Exemple de vue contextuelle: les lexiques du LATL

Chaque lexique monolingue est structuré en deux tables:

- la table des mots qui contient toutes les formes fléchies des mots. Ex: pour le verbe jouer, “joue”, “joues”, “jouons” etc. donneront lieu à autant de tuples dans la table

- la table des lexèmes qui contient toutes les lectures syntaxiques (et sémantiques) d’un mot. Ex: pour le verbe jouer, on aura une lecture “intransitive” (les enfants jouent), “transitive avec prép. objet” (jouer du piano), etc

```
create table f_word(key char(24), word_index number(9), lexeme_index
number(9), cat number(2), mode number(10), tense number(10), ...

create table f_lexeme(lexeme_index number(9), word_index number(9),
type number(3), features number(10), nb_arg number(1),...
```

Vue des items lexicaux (~ ce que l’on trouve dans un dictionnaire traditionnel):

```
create view f_item as select key, f_lexeme.lexeme_index lexeme_index,
f_lexeme.cat cat, features,...
from f_word, f_lexeme where f_word.word_index = f_lexeme.word_index
```


Vue contextuelle: implémentation des sous-classes

Exemple: les périodiques

```
create table périodique(ISSN number(10), titre char(30),
        est_un_quotidien char(1),
        est_une_revue char(1),
        fréquence_de_parution char(20))

create table JourDeParution(ISSN number(10), jour
number(1))
```

- Les revues:

```
create view revue as
select ISSN, titre, fréquence_de_parution
from périodique
where est_une_revue=’o’
```

- Les quotidiens:

```
create view quotidien as
select ISSN, titre, ‘quotidien’ fréquence_de_parution
from périodique
where est_un_quotidien=’o’
```

(2) Vues “interfaces”

Interfacer un schéma

Supposons qu’on doit utiliser une bd qui possède déjà un schéma de relations (qu’on n’a pas le droit de modifier)

-> on crée des vues sur les relations existantes

- les applications (p.e. requêtes SQL) seront basées sur les vues
- les vues sont définies de manières à faciliter l’écriture des applications
- si le schéma de la base change, il suffit d’adapter les vues, sans avoir à modifier les applications
- changer la terminologie dans les schémas

Exemple: multilinguisme

create view Customer as select num_client cust_no, nom surname, prenom firstname, adresse address from clients

(3) Vues “attributs calculés”

Lorsque la valeur d’un attribut est entièrement déterminée par de l’information existant déjà dans la base

Exemple:

create view personne_avec_age as select nom, prenom, date() - date_de_naissance age from personne

Remarque: date() donne la date système d’aujourd’hui

(4) Vues “déductives”

Programmation logique: “à partir d’un ensemble de faits et de règles de déduction, on déduit de nouvelles informations.”

Rappel: en PROLOG, faits + règles (clauses de Horn)

En SQL, faits seront représentés par les tuples contenus dans les tables les règles seront définies par les vues

Exemples: soit le schéma de l’arbre généalogique:

```
create table pers(nom char(20), sexe char(1));
create table geni(parent char(20), enfant char(20));
```

A partir des données contenues dans ces 2 tables, nous aimerions poser les questions:

? - qui est la soeur de ... ?
? - qui sont les grand-parents ... ?
? - qui est le cousin de ... ?
? - qui est l'ancÍtre de ... ?

Les vues suivantes répondent à ces questions:

```
CREATE VIEW femme
  AS SELECT nom
       FROM pers
       WHERE sexe='F';

CREATE VIEW homme
  AS SELECT nom
       FROM pers
       WHERE sexe='H';
```


Vues “déductives” (suite)

```
CREATE VIEW pere_de
  AS SELECT parent pere ,enfant
       FROM geni,homme
       WHERE geni.parent=homme.nom;

CREATE VIEW mere_de
  AS SELECT parent mere,enfant
       FROM geni,femme
       WHERE geni.parent=femme.nom;

CREATE VIEW soeur_de
  AS SELECT a.enfant soeur,b.enfant nom
       FROM geni a,geni b,femme
       WHERE a.parent=b.parent
         AND a.enfant=femme.nom
         AND a.enfant<>b.enfant ;

CREATE VIEW frere_de
  AS SELECT a.enfant frere,b.enfant nom
       FROM geni a,geni b,homme
       WHERE a.parent=b.parent
         AND a.enfant=homme.nom
         AND a.enfant<>b.enfant;

CREATE VIEW grandpere_de
  AS SELECT a.pere grandpere,b.enfant petitenfant
       FROM pere_de a,geni  b WHERE a.enfant=b.parent;

CREATE VIEW grandmere_de
  AS SELECT a.mere grandmere,b.enfant petitenfant
       FROM mere_de a,geni b WHERE a.enfant=b.parent;

CREATE VIEW grandparent_de(grandparent, petitenfant)
  AS SELECT grandpere, petitenfant
       FROM grandpere_de
     union
       SELECT grandmere, petitenfant
       FROM grandmere_de;
```

Questions sur vues “déductives”

Question: “Les soeurs de Jacques ?”

```
SELECT distinct * FROM soeur_de
   WHERE nom='jacques';
```

Question: “Les grand-parents d’Amélie?”

```
SELECT distinct * FROM grandparent_de
   WHERE petitenfant='amélie';
```

La question des ancêtres est plus difficile car elle fait intervenir la notion de récursivité.

Définition d’ancêtre:

X est ancêtre de Z
si 1)X est le géniteur de Z
ou bien si 2) X est ancêtre de Y et Y est le géniteur de Z

En toute généralité, il n'est pas possible de créer des vues récursives.

Avec la clause connect on peut définir certaines vues récursives

```
CREATE VIEW ancetre_de
AS SELECT a.nom ancetre, b.nom descendant
  FROM pers a, pers b
  WHERE b.nom in (SELECT enfant FROM geni
            connect by prior enfant=parent
            start with parent=a.nom);
```


Commentaires sur les vues

- en interrogation, une vue se comporte exactement comme une table
- les modifications à travers les vues sont restreintes aux vues ne contenant pas les opérations de:
  - la clause distinct
  - jointure
  - les aggrégations (group by)
  - les connexions (connect by)

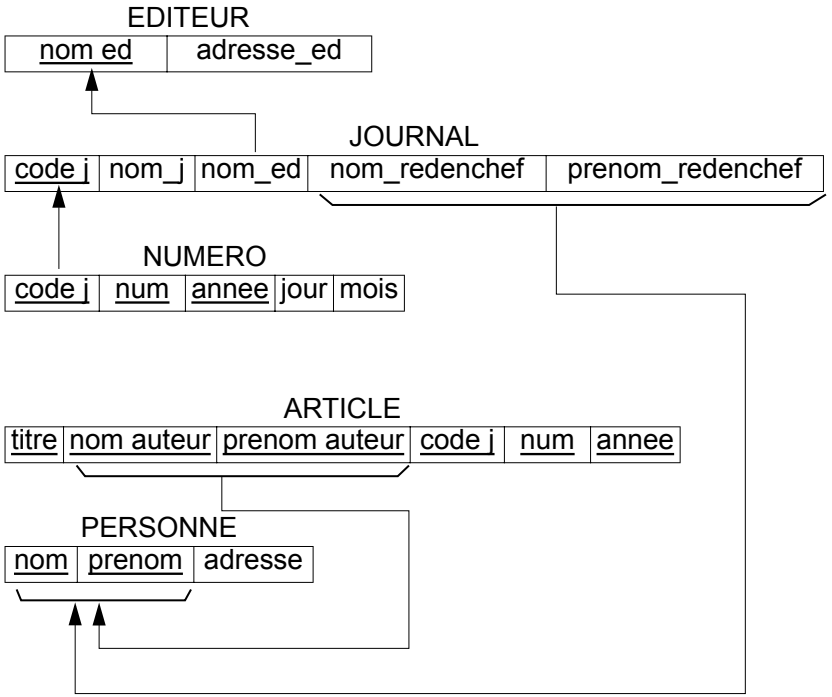
En Access

- créer une requête (p.e. en SQL)

- sauver la requête -> on pourra l’utiliser comme vue dans d’autres requêtes

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

nom_cru	commune	region	coul

VINS1

region	coul	millesime	qualite

CEPAGE_REGION

cepage	r_prod	coul

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
Sous-le-Dos-d’Ane    Meursault        Bourgogne        rouge

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
Goutte d'Or	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
Les Charmes	**Meursault**	**Bourgogne**	**blanc**
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 supression

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).

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

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. 33)

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

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

120 ligne(s) sélectionnées
```