HELMO
Haute Ecole
Libre Mosane

INFORMATIQUE Développement d'applications
BLOC 1
UE05 Bases de données

Chapitre 3.2 : SQL - DML

Vincent Reip

Septembre 2022

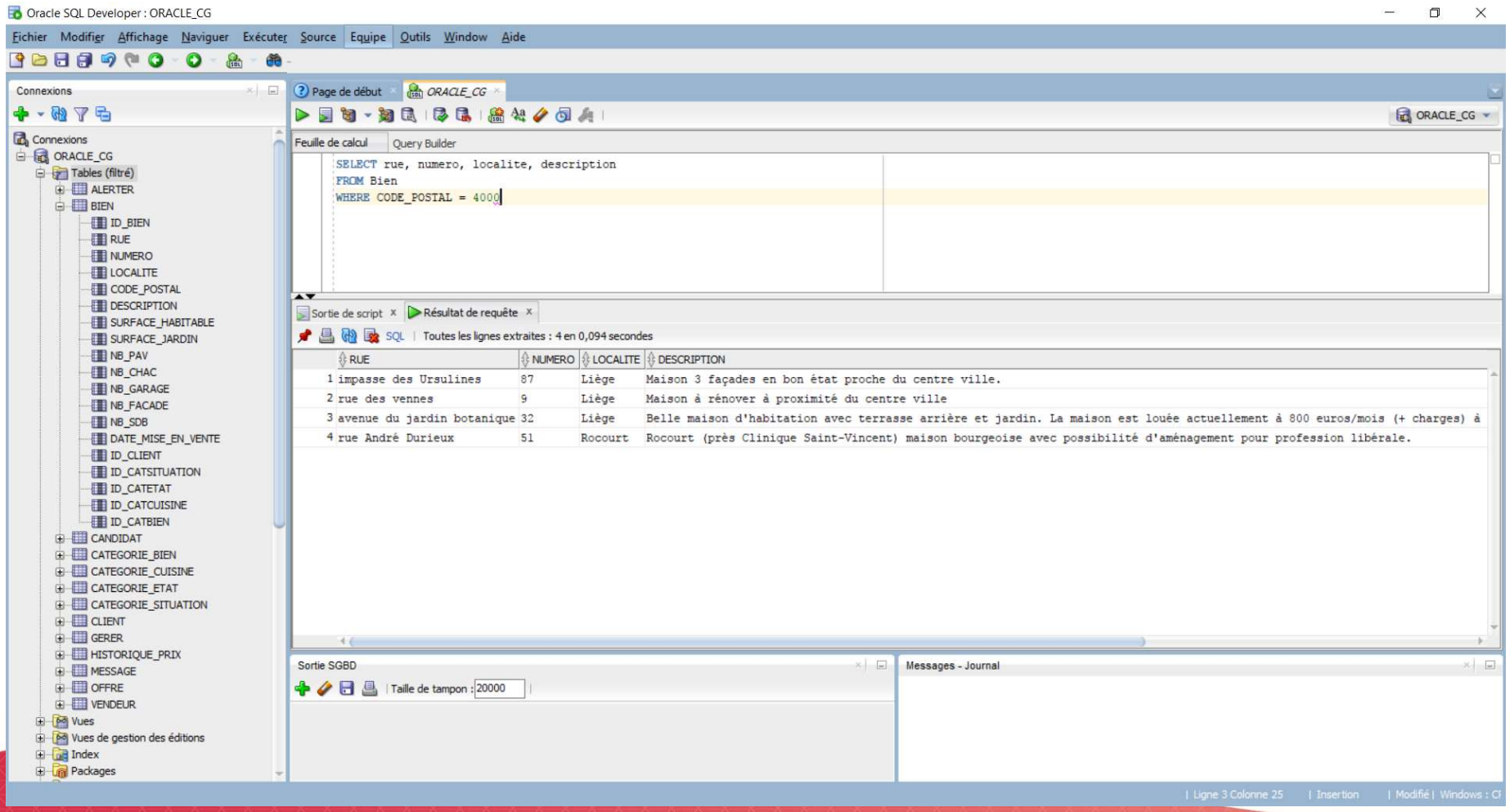
Objectif

- Au terme de ce chapitre, l'étudiant sera capable de :
  - Manipuler les données au sein d'un schéma relationnel

DML : manipulation des données

- Le langage SQL permet de rechercher de l'information dans une ou plusieurs tables de la base de données
- Le langage SQL s'appuie sur l'algèbre relationnelle pour fournir les opérations de base (projection et sélection)
  - Les deux opérations sont généralement utilisées conjointement
- Le langage SQL propose quelques particularités et outils additionnels
  - Tri, groupement, opérateurs spéciaux…
  - Manipulation de dates et de chaînes de caractères…

SGBD : interface utilisateur



SELECT simple

```
SELECT [ ALL | DISTINCT] nom_col1, nom_col2…
FROM nom_table
WHERE expression_conditionnelle
```

- ALL | DISTINCT : DISTINCT permet de supprimer les éventuels doublons (valeur par défaut = ALL)
- nom_col : liste des attributs ou d'expressions que l'on désire retrouver dans le résultat.
  - Les items sont séparés par une virgule
  - Une expression peut faire intervenir un/plusieurs attributs et différents opérateurs/fonctions sur ces attributs
- nom_table: nom de la table sur laquelle on désire effectuer la recherche
- expression_conditionnelle : condition (optionnelle) de sélection des tuples faisant intervenir des prédicats éventuellement combinés grâce aux opérateurs booléens

Enoncé dont le sens logique peut être vrai ou faux en fonction de la valeur des arguments.

SELECT simple

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru, region
FROM vin
```

cru	region
VOLNAY	Bourgogne
VOLNAY	Bourgogne
CHENAS	Beaujolais
VILLAGE	Beaujolais Sud
JULIENAS	Beaujolais

```
SELECT DISTINCT cru, region
FROM vin
```

cru	region
VOLNAY	Bourgogne
CHENAS	Beaujolais
VILLAGE	Beaujolais Sud
JULIENAS	Beaujolais

SELECT simple

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru, millesime, degre*100
FROM vin
```

cru	millesime	degre*100
VOLNAY	1983	12
VOLNAY	1979	13.5
CHENAS	1983	14
VILLAGE	1998	12.5
JULIENAS	1986	11.5

Expressions arithmétiques autorisées sur des attributs de type numérique.

SELECT simple

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT *
FROM vin
WHERE qualite = 'A'
```

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
3	CHENAS	1983	Beaujolais	A	0.14

SELECT simple

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru, region, qualite, degre
FROM vin
WHERE qualite = 'A' OR qualite = 'B'
```

cru	region	qualite	degre
VOLNAY	Bourgogne	A	0.12
VOLNAY	Bourgogne	B	0.135
CHENAS	Beaujolais	A	0.14

SELECT simple

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru || '(' || region || ')',  degre * 100 || '°'
FROM vin
WHERE (qualite = 'A' OR qualite = 'B') AND millesime <> 1983
```

Attention à la priorité des opérateurs…
AND est évalué avant OR

cru || ' (' || region || ')'	degre * 100 || '°'
VOLNAY (Bourgogne)	13.5°

SELECT simple

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru || '(' || region || ')',  degre * 100 || '°'
FROM vin
WHERE qualite = 'A' OR qualite = 'B' AND millesime <> 1983
```

cru || ' (' || region || ')'	degre * 100 || '°'
VOLNAY (Bourgogne)	13.5°
VOLNAY (Bourgogne)	12°
CHENAS (Beaujolais)	14°

SELECT simple

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru || '(' || region || ')‘ AS VIN,  degre * 100 || '°' AS DEGRE
FROM vin
WHERE (qualite = 'A' OR qualite = ‘B‘) AND millesime <> 1983
```

Les colonnes du résultat peuvent être renommées.

VIN	DEGRE
VOLNAY (Bourgogne)	13.5°

Conditions de sélection

- Les prédicats désignent des relations élémentaires entre
  - des attributs comparables
  - des attributs et une ou plusieurs constantes
- Il existe plusieurs types de prédicats
  - Prédicats de comparaisons (=,>=, <=,<>, <, >)
  - Prédicats d'appartenance
    - à un intervalle continu (BETWEEN)
    - à une énumération discrète (IN)
  - Prédicat de valeur inconnue (IS NULL)
  - Prédicat de comparaison de textes par masque (LIKE combiné avec un masque contenant % et/ou _)
    - % : représente 0 ou n caractères quelconques
    - _ : représente 1 seul caractère

BETWEEN

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru, region, millesime
FROM vin
WHERE millesime BETWEEN 1983 AND 1987
```

cru	region	millesime
VOLNAY	Bourgogne	1983
CHENAS	Beaujolais	1983
JULIENAS	Beaujolais	1986

BETWEEN : appartenance à un intervalle

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru, region, millesime
FROM vin
WHERE qualite BETWEEN A AND C
```

cru	region	millesime
VOLNAY	Bourgogne	1983
VOLNAY	Bourgogne	1979
CHENAS	Beaujolais	1983
JULIENAS	Beaujolais	1986

Ordre lexicographique : généralise l'ordre alphabétique aux chaînes de caractères (en se basant par exemple sur le code ASCII pour comparer 2 caractères).

IN : appartenance à une énumération

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru, region, millesime
FROM vin
WHERE millesime IN (1979,1983, 1998)
```

cru	region	millesime
VOLNAY	Bourgogne	1983
VOLNAY	Bourgogne	1979
CHENAS	Beaujolais	1983
VILLAGE	Beaujolais Sud	1998

IS NULL : vérification de valeur inconnue

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	NULL
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	NULL

```
SELECT cru, region, degre
FROM vin
WHERE degre IS NULL
```

cru	region	degre
CHENAS	Beaujolais	NULL
JULIENAS	Beaujolais	NULL

```
SELECT cru, region, degre
FROM vin
WHERE degre IS NOT NULL
```

cru	region	degre
VOLNAY	Bourgogne	0.12
VOLNAY	Bourgogne	0.135
VILLAGE	Beaujolais Sud	0.125

LIKE : comparaison par masque

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru, region, degre
FROM vin
WHERE cru LIKE '__L%'
```

cru	region	degre
VOLNAY	Bourgogne	0.12
VOLNAY	Bourgogne	0.135
VILLAGE	Beaujolais Sud	0.125
JULIENAS	Beaujolais	0.115

L'opérateur LIKE ne doit pas être utilisé comme un opérateur d'égalité simple.

Combinaison de conditions

vin

	idVin	cru	millesime	region	qualite	degre	
V	1	VOLNAY	1983	Bourgogne	A	0.12	V
V	2	VOLNAY	1979	Bourgogne	B	0.135	V
X	3	CHENAS	1983	Beaujolais	A	0.14	V
V	4	VILLAGE	1998	Beaujolais Sud	D	0.125	V
V	5	JULIENAS	1986	Beaujolais	C	0.115	X

```
SELECT cru, region, degre*100
FROM vin
WHERE cru NOT LIKE 'C%' AND (qualite IN ('A', 'B') OR millesime = 1998)
```

cru	region	degre*100
VOLNAY	Bourgogne	12
VOLNAY	Bourgogne	13.5
VILLAGE	Beaujolais Sud	12.5

Exercices

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

1. Sélectionner les vins dont le millésime est 1983 et le degré supérieur à 0.12
2. Sélectionner les noms de cru et le degré des vins dont le millésime est 1983 et dont le degré est compris entre 0.10 et 0.13
3. Sélectionner les noms de cru, l'âge et la qualité des vins du Beaujolais.
4. Sélectionner les vins dont le nom de cru est inconnu (laissé vide dans la table)
5. Sélectionner les noms de cru correspondant à des vins dont la qualité est A ou B
6. Sélectionner les noms de cru et la région des vins dont le millésime multiplié par le degré est plus grand que 180

Tri sur une sélection

```
SELECT [ ALL | DISTINCT] nom_col1, nom_col2…
FROM nom_table
WHERE expression_conditionnelle
ORDER BY nom_col1 [ASC | DESC], nom_col1 [ASC | DESC]…
```

- Il est possible d'ordonner les tuples issus d'une sélection
  - sur un ou plusieurs attributs
  - en spécifiant l'ordre du tri pour chacun des attributs mentionnés
    - ASC : croissant (valeur par défaut)
    - DESC : décroissant

L'ordre des attributs mentionnés dans la clause ORDER BY est important…

Tri sur une sélection

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT cru, region, millesime
FROM vin
WHERE cru <> 'CHENAS'
ORDER BY millesime ASC
```

cru	region	millesime
VOLNAY	Bourgogne	1979
VOLNAY	Bourgogne	1983
JULIENAS	Beaujolais	1986
VILLAGE	Beaujolais Sud	1998

Tri sur une sélection

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
SELECT *
FROM vin
ORDER BY qualite, degre DESC
```

cru	millesime	region	qualite	degre
CHENAS	1983	Beaujolais	A	0.14
VOLNAY	1983	Bourgogne	A	0.12
VOLNAY	1979	Bourgogne	B	0.135
JULIENAS	1986	Beaujolais	C	0.115
VILLAGE	1998	Beaujolais Sud	D	0.125

L'ordre de tri sur l'attribut *qualite* est croissant (ASC – valeur par défaut)

Insertion de tuples

```
INSERT INTO nom_table [(nom_attr1, nom_attr2, …)]
VALUES (valeur_1, valeur_2, …)
```

- nom_table : le nom de la table dans laquelle on désire insérer un tuple

- [(nom_attr1, nom_attr2,…)] : la liste (optionnelle) des attributs, entre parenthèses et séparés par une virgule, pour lesquels on va fournir une valeur
    - NULL ou valeur par défaut pour les attributs non-spécifiés
    - liste absente ≈ liste complète des attributs de la relation

- (valeur_1, valeur_2,…) : la liste des valeurs qui seront associées aux attributs listés
    - respect de la séquence et du type
    - peut inclure une requête de sélection dont le schéma est compatible avec l'attribut concerné
    - peut inclure des expressions

- une requête INSERT INTO permet d'insérer un ou des tuples dans une seule table

Insertion de plusieurs tuples

```
INSERT INTO nom_table [(nom_attr1, nom_attr2, …)]
VALUES (valeur_1, valeur_2, …),
       (valeur_3, valeur_4, …),
       (valeur_5, valeur_6, …),
       (valeur_7, valeur_8, …),
       (valeur_9, valeur_10, …),
       …
```

Insertion de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
INSERT INTO vin (qualite,degre, idVin, cru,millesime,region)
VALUES ('B', 0.11, 6, 'BOURGEOIS', 1997, 'Bordeaux')
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115
6	BOURGEOIS	1997	Bordeaux	B	0.11

Insertion de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
INSERT INTO vin
VALUES (6, 'BOURGEOIS',1997, 'Bordeaux', 'B',0.11)
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115
6	BOURGEOIS	1997	Bordeaux	B	0.11

Insertion de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
INSERT INTO vin (idVin, cru,millesime,region, degre)
VALUES (6, 'BOURGEOIS',1997, 'Bordeaux', 0.11)
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115
6	BOURGEOIS	1997	Bordeaux	NULL	0.11

Hypothèse : pas de contrainte NOT NULL ni de valeur par défaut

Que se passerait il sinon ?

Insertion de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VILLAGE	1989	Beaujolais	A	0.11
2	JULIENAS	1986	Beaujolais	C	0.115

```
INSERT INTO vin (idVin, cru,millesime,region, qualite, degre)
VALUES (3, 'JULIENAS',
          (SELECT millesime+1
           FROM vin
           WHERE idvin = 2)
          , 'Beaujolais', 'B', 0.24/2)
```

Les valeurs listées pour l'insertion du tuple peuvent provenir d'une expression (arithmétique ou autre) voire d'une sous-requête (mais elle doit obligatoirement renvoyer une seule valeur)

vin

idVin	cru	millesime	region	qualite	degre
1	VILLAGE	1989	Beaujolais	A	0.11
2	JULIENAS	1986	Beaujolais	C	0.115
3	JULIENAS	1987	Beaujolais	B	0.12

Modification de tuples

```
UPDATE nom_table
SET nom_attr1 = valeur littérale | expression | NULL ,
    nom_attr2 = valeur littérale | expression | NULL ,
    …
WHERE expression conditionnelle
```

- nom_table : table dans laquelle on désire modifier un ou plusieurs tuples
- nom_attr1 = valeur littérale | expression | NULL : liste d'affectations sur les attributs de la table
- WHERE expression conditionnelle : condition de sélection des tuples à modifier (optionnelle)
- une requête UPDATE permet de modifier plusieurs tuples dans une seule table
  - le nombre de tuples modifiés dépend de la clause WHERE

Modification de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
UPDATE vin
SET qualite = 'A'
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	A	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	A	0.125
5	JULIENAS	1986	Beaujolais	A	0.115

Modification de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
UPDATE vin
SET qualite = 'A'
WHERE region LIKE 'Bea%'
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	A	0.125
5	JULIENAS	1986	Beaujolais	A	0.115

Modification de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
UPDATE vin
SET qualite = NULL
WHERE cru = 'CHENAS'
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	NULL	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

Modification de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
UPDATE vin
SET degre = degre * 100
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	12
2	VOLNAY	1979	Bourgogne	B	13.5
3	CHENAS	1983	Beaujolais	A	14
4	VILLAGE	1998	Beaujolais Sud	D	12.5
5	JULIENAS	1986	Beaujolais	C	11.5

Modification de tuples

cru

idCru	nom
1	VOLNAY
2	CHENAS
3	JULIENAS
4	VILLAGE
5	BOURGEOIS

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
UPDATE vin
SET cru = (SELECT nom FROM cru WHERE idCru = 3) , qualite = 'B'
WHERE degre = 0.14
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	JULIENAS	1983	Beaujolais	B	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

Exercices

- Augmenter de 10% la valeur du degré d'alcool des vins provenant de Bourgogne
- Donner une qualité 'A' aux vins provenant de Bourgogne et dont le millésime est compris entre 1970 et 1990
- Donner une qualité 'B' aux vins dont le millésime est inférieur à 1980 et une qualité 'C' aux vins dont le millésime est supérieur ou égal à 1980
- Rajouter 5 ans aux millésimes des vins dont le cru contient la lettre 'A'

Suppression de tuples

```
DELETE FROM nom_table
WHERE expression conditionnelle
```

- nom_table: nom de la table dans laquelle on désire supprimer un ou plusieurs tuples
- WHERE expression conditionnelle : condition de sélection des tuples à supprimer (optionnelle)
- une requête DELETE permet de supprimer plusieurs tuples dans une seule table
  - le nombre de tuples supprimés dépend de la clause WHERE
  - tous les tuples sont supprimés si la clause WHERE est omise

Suppression de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

```
DELETE FROM vin
WHERE cru IN ('CHENAS', 'JULIENAS')
```

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
4	VILLAGE	1998	Beaujolais Sud	D	0.125

Suppression de tuples

vin

idVin	cru	millesime	region	qualite	degre
1	VOLNAY	1983	Bourgogne	A	0.12
2	VOLNAY	1979	Bourgogne	B	0.135
3	CHENAS	1983	Beaujolais	A	0.14
4	VILLAGE	1998	Beaujolais Sud	D	0.125
5	JULIENAS	1986	Beaujolais	C	0.115

DELETE FROM vin

vin

idVin	cru	millesime	region	qualite	degre

Produit cartésien

```
SELECT nom_attr1, nom_attr2, …
FROM nom_table_1
CROSS JOIN nom_table_2
CROSS JOIN nom_table_3
 …
```

- Le produit cartesien de deux tables permet de retourner l'ensemble des combinaisons des tuples de la première table avec les tuples de la deuxième table.
- Le produit cartésien peut être combiné avec d'autres opérations (sélection, groupement, tri,...)
- Les doublons ne sont pas supprimés
- Ne présente pas un grand intérêt

Jointure

```
SELECT nom_attr1, nom_attr2, …
FROM nom_table_1
JOIN nom_table_2 ON condition de jointure
JOIN nom_table_3 ON condition de jointure
…
```

- La jointure permet de combiner les tuples de la première table et ceux de la deuxième table, pour autant qu'ils respectent la condition de jointure
  - C'est un sous-ensemble du produit cartésien des 2 tables
  - La condition de jointure porte généralement sur des clés primaires et étrangères
  - Equi-jointure : la condition utilise l'opérateur =
  - Theta-jointure : la condition utilise un des opérateurs <,>,<=, >=

Il existe une ancienne syntaxe (à ne plus utiliser) :

SELECT …
FROM nom_table1], nom_table2,…
WHERE [conditions de jointure]

Jointure

vente

idVente	magasin	montant	date
1	Los Angeles	1500	1999-06-01
2	San Diego	250	1999-06-05
3	Boston	800	1999-10-02
4	San Diego	300	1999-12-10
5	Los Angeles	300	1999-04-02
6	Boston	700	1999-06-20

vente_par_internet

idVPI	date	total
1	1999-06-01	250
2	1999-01-10	535
3	1999-04-02	320
4	1999-01-12	750

```
SELECT v.date AS Date_Vente,
       v.montant AS Vente_Magasin,
       vi.total AS Vente_Internet
FROM vente v
JOIN vente_par_internet vi ON v.date = vi.date
```

L'utilisation d'alias (courts) plutôt que le nom complet de la table permet de lever l'ambiguïté liée aux attributs de même nom sans alourdir la requête.

Montants des ventes pour les jours où les deux types de vente ont eu lieu.

Date_Vente	Vente_Magasin	Vente_Internet
1999-06-01	1500	250
1999-04-02	300	320

Jointure

etudiant

idEtudiant	Nom	Prenom	idCursus
1	Dupont	Luc	1
2	Durand	Noémie	2
3	Frion	Stéphane	4
4	Marcoli	Diana	3
5	Préant	Marc	2
6	El Madri	Dris	1

cursus

idCursus	nom
1	Informatique – Développement d'applications
2	Informatique – Cybersécurité
3	Automatisation
4	Robotique

```
SELECT e.nom, e.prenom, cu.nom AS Cursus
FROM etudiant e
JOIN cursus cu ON e.idcursus = cu.idcursus
```

La condition de jointure est très souvent une condition d'égalité entre une clé étrangère et une clé primaire (ou candidate).

Nom	Prenom	Cursus
Dupont	Luc	Informatique – Développement d'applications
Durand	Noémie	Informatique – Cybersécurité
Frion	Stéphane	Robotique
Marcoli	Diana	Automatisation
Préant	Marc	Informatique – Cybersécurité
El Madri	Dris	Informatique – Développement d'applications

Jointure naturelle

```
nom_attr1, nom_attr2, …
FROM nom_table_1
NATURAL JOIN nom_table_2
NATURAL JOIN nom_table_3
…
```

- La jointure naturelle est une équi-jointure opérant sur les attributs de même nom.
  - on conseille généralement de lui préférer une équijointure avec une condition explicite pour éviter des problèmes

L'utilisation de jointures naturelles est proscrite dans ce cours…

Jointures externes

```
SELECT nom_attr1, nom_attr2, …
FROM nom_table_1
(LEFT |RIGHT | FULL) OUTER JOIN nom_table_2 ON condition de jointure
(LEFT |RIGHT | FULL) OUTER JOIN nom_table_3 ON condition de jointure
…
```

- Dans une jointure, un tuple célibataire est un tuple qui n'a pas de tuple correspondant dans l'autre table (par rapport à la condition de jointure)
- La jointure externe permet de faire apparaître dans le résultat les tuples célibataires. Les attributs « manquants » seront mis à NULL
  - LEFT : on préserve les tuples célibataires de la table à gauche des mots-clés OUTER JOIN
  - RIGHT : on préserve les tuples célibataires de la table à droite des mots-clés OUTER JOIN
  - FULL : on préserve les tuples célibataires des deux tables
-

LEFT OUTER JOIN

vente

idVente	magasin	montant	date
1	Los Angeles	1500	1999-06-01
2	San Diego	250	1999-06-05
3	Boston	800	1999-10-02
4	San Diego	300	1999-12-10
5	Los Angeles	300	1999-04-02
6	Boston	700	1999-06-20

vente_par_internet

idVPI	date	total
1	1999-06-01	250
2	1999-01-10	535
3	1999-04-02	320
4	1999-01-12	750

```
SELECT v.*, vi.*
FROM vente v
LEFT OUTER JOIN vente_par_internet vi ON v.date = vi.date
```

Tuples célibataires de 'vente'

idVente	magasin	montant	date	idVPI	date	total
1	Los Angeles	1500	1999-06-01	1	1999-06-01	250
2	San Diego	250	1999-06-05	NULL	NULL	NULL
3	Boston	800	1999-10-02	NULL	NULL	NULL
4	San Diego	300	1999-12-10	NULL	NULL	NULL
5	Los Angeles	300	1999-04-02	3	1999-04-02	320
6	Boston	700	1999-06-20	NULL	NULL	NULL

LEFT OUTER JOIN

etudiant

idEtudiant	Nom	Prenom	idCursus
1	Dupont	Luc	1
2	Durand	Noémie	2
3	Frion	Stéphane	4
4	Marcoli	Diana	NULL
5	Préant	Marc	2
6	El Madri	Dris	1

cursus

idCursus	nom
1	Informatique – Développement d'applications
2	Informatique – Cybersécurité
3	Automatisation
4	Robotique

```
SELECT e.nom, e.prenom, cu.nom AS Cursus
FROM etudiant e
LEFT OUTER JOIN cursus cu ON e.idcursus = cu.idcursus
```

Nom	Prenom	Cursus
Dupont	Luc	Informatique – Développement d'applications
Durand	Noémie	Informatique – Cybersécurité
Frion	Stéphane	Robotique
Marcoli	Diana	NULL
Préant	Marc	Informatique – Cybersécurité
El Madri	Dris	Informatique – Développement d'applications

LEFT OUTER JOIN

etudiant

idEtudiant	Nom	Prenom	idCursus
1	Dupont	Luc	1
2	Durand	Noémie	2
3	Frion	Stéphane	4
4	Marcoli	Diana	4
5	Préant	Marc	2
6	El Madri	Dris	1

cursus

idCursus	nom
1	Informatique – Développement d'applications
2	Informatique – Cybersécurité
3	Automatisation
4	Robotique
5	Technico-commercial.e

```
SELECT cu.nom
FROM cursus cu
LEFT OUTER JOIN etudiant e ON e.idcursus = cu.idcursus
WHERE e.idetudiant IS NULL
```

nom
Automatistion
Technico-commercial.e

On isole les tuples célibataires grâce à la condition IS NULL ce qui nous permet de répondre à la question « *Quels sont les cursus pour lesquels il n'y a aucun étudiant inscrit ?* »

RIGHT OUTER JOIN

etudiant

idEtudiant	Nom	Prenom	idCursus
1	Dupont	Luc	1
2	Durand	Noémie	2
3	Frion	Stéphane	4
4	Marcoli	Diana	NULL
5	Préant	Marc	2
6	El Madri	Dris	1

cursus

idCursus	nom
1	Informatique – Développement d'applications
2	Informatique – Cybersécurité
3	Automatisation
4	Robotique

```
SELECT e.nom, e.prenom, cu.nom AS Cursus
FROM etudiant e
RIGHT OUTER JOIN cursus cu ON e.idcursus = cu.idcursus
```

Nom	Prenom	Cursus
Dupont	Luc	Informatique – Développement d'applications
Durand	Noémie	Informatique – Cybersécurité
Frion	Stéphane	Robotique
NULL	NULL	Automatisation
Préant	Marc	Informatique – Cybersécurité
El Madri	Dris	Informatique – Développement d'applications

FULL OUTER JOIN

etudiant

idEtudiant	Nom	Prenom	idCursus
1	Dupont	Luc	1
2	Durand	Noémie	2
3	Frion	Stéphane	4
4	Marcoli	Diana	NULL
5	Préant	Marc	2
6	El Madri	Dris	1

cursus

idCursus	nom
1	Informatique – Développement d'applications
2	Informatique – Cybersécurité
3	Automatisation
4	Robotique

```
SELECT e.nom, e.prenom, cu.nom AS Cursus
FROM etudiant e
FULL OUTER JOIN cursus cu ON e.idcursus = cu.idcursus
```

Nom	Prenom	Cursus
Dupont	Luc	Informatique – Développement d'applications
Durand	Noémie	Informatique – Cybersécurité
Frion	Stéphane	Robotique
NULL	NULL	Automatisation
Marcoli	Diana	NULL
Préant	Marc	Informatique – Cybersécurité
El Madri	Dris	Informatique – Développement d'applications

Exercices

Candidat

idCandidat	Nom	Prenom	Localite
1	Huard	Nathalie	Liège
2	Dupont	Noémie	Verviers
3	Foucroule	Stéphane	Spa
4	Nandeit	Luc	Liège
5	Puris	Sophiane	Waremme
6	Castro	Cédrid	Verviers
7	Jonlet	Eric	Liège

Offre

idOffre	Description	Salaire_annuel	Societe
1	Analyste programmeur Java	38.000	1
2	Architecte IT	45.000	1
3	Database administrator	44.000	3
4	Project manager	50.000	4

Candidature

idCandidat	idOffre	DateCandidature
1	2	01/09/2022
1	3	01/09/2022
3	4	12/09/2022
4	4	19/09/2022
7	3	03/10/2022

Societe

idSociete	Nom
1	NSI
2	NRB
3	EVS
4	Afelio
5	Computerland

Exercices

- Dessiner le schéma relationnel en indiquant les clés primaires et les clés étrangères
- Ecrivez les requêtes suivantes :
  - Afficher les offres publiées par la société NSI (on veut voir la description et le salaire annuel)
  - Afficher les offres dont le salaire annuel est supérieur à 40.000 € (on veut voir la description, le salaire annuel et le nom de la société)
  - Afficher les candidatures des candidats de Liège, Verviers ou Spa (on veut voir les nom et prénom des candidats, leur localité, la date de la candidature et la description de l'offre)
  - Afficher les offres qui n'ont reçu aucune candidature (on veut voir la description et le salaire)
  - Pour toutes les offres, afficher les candidatures reçues (on veut voir le nom de la société, la description de l'offre, la date de candidature et les noms et prénom du candidat). Si une offre n'a reçu aucune candidature, elle doit quand même apparaître dans le résultat.