

SGBD(R)

Système de Gestion de Bases de Données (Relationnels)

OLIVARÈS Georges

25/05/2016

© Groupe CGI inc. CONFIDENTIEL

CGI

La force de l'engagement[MD]

SGBDr
Langages relationnels



CGI

La force de l'engagement[MD]

Plan – Chapitre 3 – Langages relationnels

1. CRUD
2. Les expressions
3. Les opérateurs
4. Les conditions
5. Les tris
6. Les jointures
7. Les groupements

CRUD – Create, Read, Update, Delete

Il existe un langage SQL pour faire différentes actions dans SQL

- Create with INSERT
- Read with SELECT
- Udpdate with UPDATE
- Delete with DELETE

Dans nous exemples, nous utiliserons

MUSICIEN (NM, NOM, PRENOM, QUALITE)

ŒUVRE (NO, NOM, TYPE, DATE, PERIODE, DUREE, NC)

COMPOSITEUR (NC, NOM, NATIONALITE)

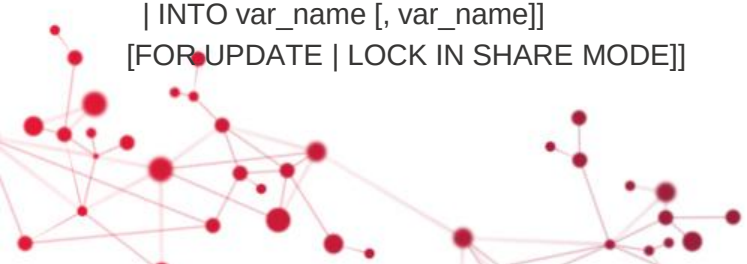
JOUE (NM, NO, LIEU, DATE)



CGI

CRUD – Create, Read, Update, Delete

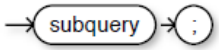
```
SELECT
    [ALL | DISTINCT | DISTINCTROW ]
      [HIGH_PRIORITY]
      [MAX_STATEMENT_TIME = N]
      [STRAIGHT_JOIN]
      [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
      [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
    select_expr [, select_expr ...]
    [FROM table_references
      [PARTITION partition_list]
    [WHERE where_condition]
    [GROUP BY {col_name | expr | position}
      [ASC | DESC], ... [WITH ROLLUP]]
    [HAVING where_condition]
    [ORDER BY {col_name | expr | position}
      [ASC | DESC], ...]
    [LIMIT {[offset,] row_count | row_count OFFSET offset}]
    [PROCEDURE procedure_name(argument_list)]
    [INTO OUTFILE 'file_name'
        [CHARACTER SET charset_name]
        export_options
      | INTO DUMPFILE 'file_name'
      | INTO var_name [, var_name]]
    [FOR UPDATE | LOCK IN SHARE MODE]]
```



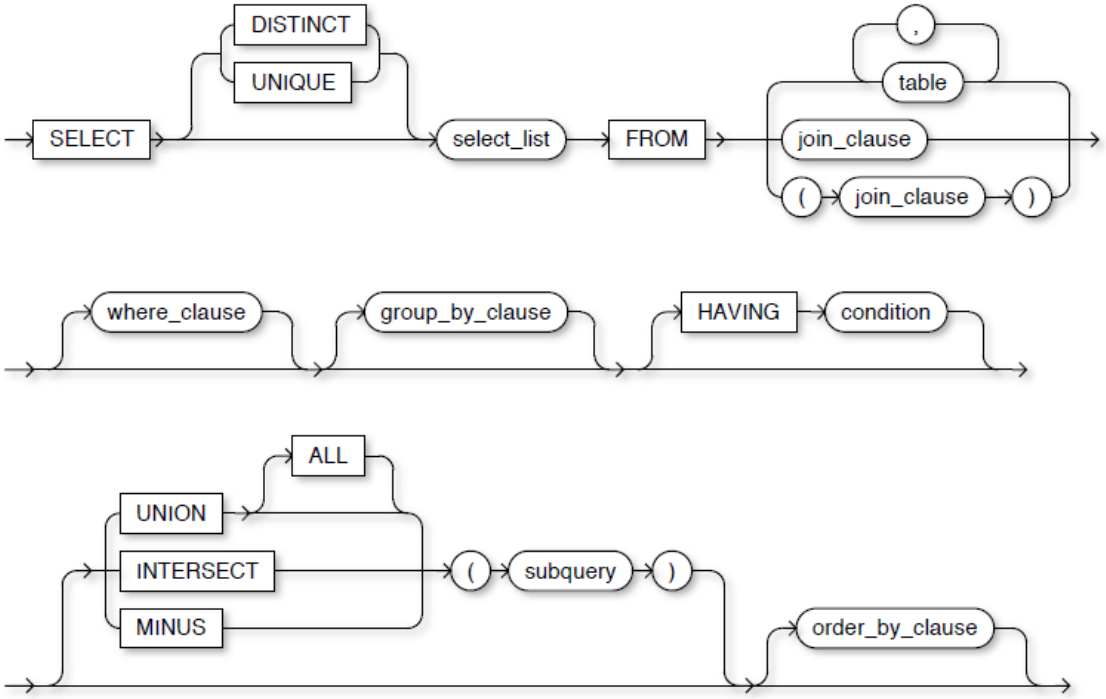
CGI

CRUD - SELECT

select ::=



subquery ::=



Expression des projections

Sélectionner les attributs (projection):

```
SELECT * FROM musiciens
SELECT nom, qualite FROM musiciens
```

L'élimination des doubles n'est pas automatique:

```
SELECT DISTINCT nom FROM musiciens
```

On peut renommer des champs:

```
SELECT DISTINCT nom, qualite as qualite2 FROM musiciens
```



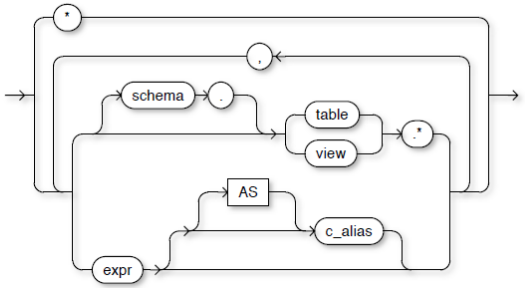
CGI

Expression des selections

- Doit être préfixé par les noms des relations si deux attributs portent le même nom
- Peut être renommer par un alias
- *: retourne toutes les colonnes (peu recommander dans une application client)

```
SELECT nom, qualite FROM musiciens
SELECT musiciens.nom, musiciens.qualite FROM musiciens
SELECT nom as nom_musicien, qualite FROM musiciens
```

select_list ::=



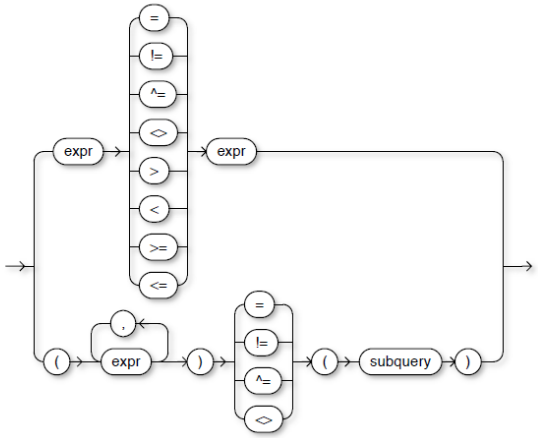
Expression des restrictions

- Le critère de restriction est un ensemble de critères logiques séparés par **AND** et **OR**
- Chaque critère possède un comparateur { =, <>, <, <=, >, >=, ... }

Filtrer les données:

```
SELECT * FROM musiciens WHERE qualite = 'Virtuose'
SELECT * FROM musiciens
    WHERE qualite = 'Virtuose' OR prenom <> 'Jean'
```

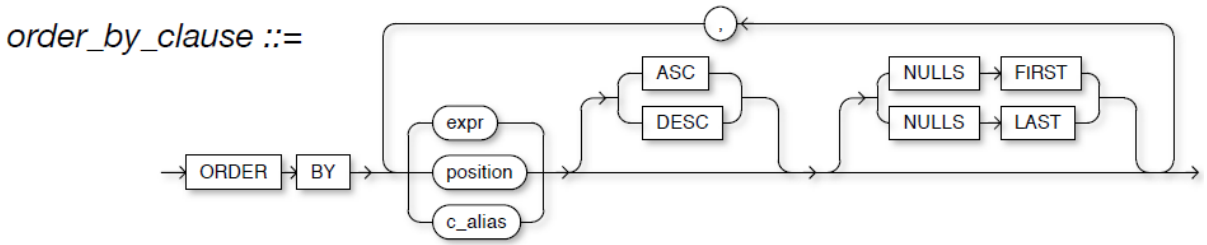
simple_comparison_condition ::=



Expression des tris

Ordonner les données (trier par ordre alphabétique décroissant sur la période et croissant sur la date

```
SELECT * FROM musiciens
     WHERE type = 'Symphonie'
     ORDER BY periode DESC, "DATE" ASC
```



Expression des groupements

- Le groupement se fait via « GROUP BY attr, … »
- S'utilise généralement avec des fonctions d'aggrégat (MIN, MAX, COUNT, …)

Grouper les données (retourne le nombre de musiciens par type (ordonné)):

```
SELECT type, COUNT(nm) FROM musiciens
     WHERE type = 'Symphonie'
     GROUP BY type
     ORDER BY type DESC
```

Expression des conjugaisons

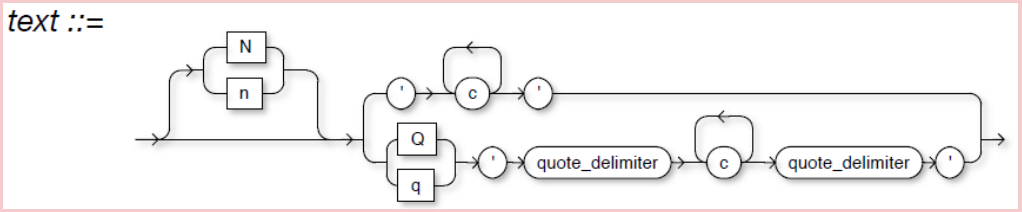
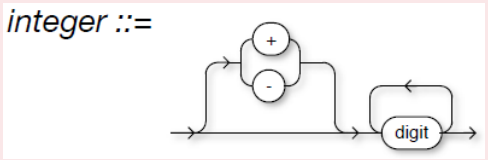
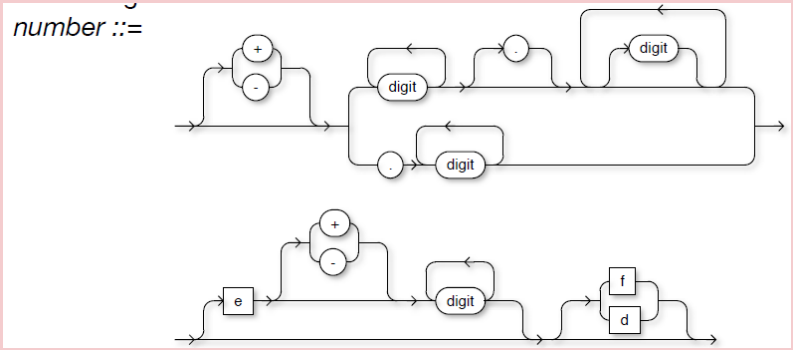
- Opérateurs:
  - **UNION** (supprime les doublons)
  - **UNION ALL** (conserve les doublons)
  - **INTERSECT**
  - **MINUS** (différence ensembliste)

- Les relations combinées doivent être de même arité (dimension) et comporter des types de même nature

```
(SELECT nom FROM musiciens WHERE type = 'Symphonie')
    UNION
(SELECT nom FROM musiciens WHERE nom <> 'John')
```


CGI

Les littéraux / constantes / texte

Schéma	Exemples
	'Dupont' 'L''article'
	421 +10 -20
	25 +6.34 25e-03 -1D 25f

Les littéraux / constantes / texte

Litteraux	Exemples

Dates	`DATE '2007-11-14'`
	`'14-NOV-07 14:50'`
	…

Timestamp	`TIMESTAMP '2007-11-14 15:50:00 Europe/Paris'`

Intervalles	`INTERVAL '123' YEAR(3)`
	`INTERVAL '4' MONTH`
	`INTERVAL '5-3' YEAR TO MONTH + INTERNVAL '20' MONTH`

Null (pas de valeur) /!\ NULL <> 0	

a IS NULL	*10*	*FALSE*
a IS NULL	*NULL*	*TRUE*
a = NULL	*10*	*UNKNOWN*
a = NULL	*NULL*	*UNKNOWN*

Les commentaires

Une ligne:

```
SELECT *                    -- Sélectionne le nom
    FROM musiciens          -- de tous les musiciens
```

Multi-lignes:

```
SELECT *
    /* Sélectionne le nom de
       tous les musiciens    */
    FROM musiciens
```

Les autres opérateurs

Opérateurs arithmétiques

- `SELECT * FROM oeuvre WHERE -duree > -30`

- `SELECT * FROM oeuvre WHERE duree + 29 = 60`
- `SELECT * FROM oeuvre WHERE duree * 2 >= 90`

- +, -, *, /

Opérateur de concaténation (II)

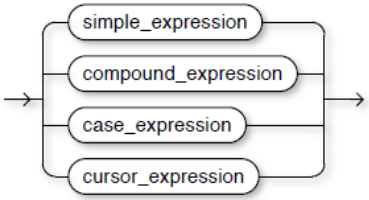
- `SELECT 'Le nom est ' || nom FROM musicien`



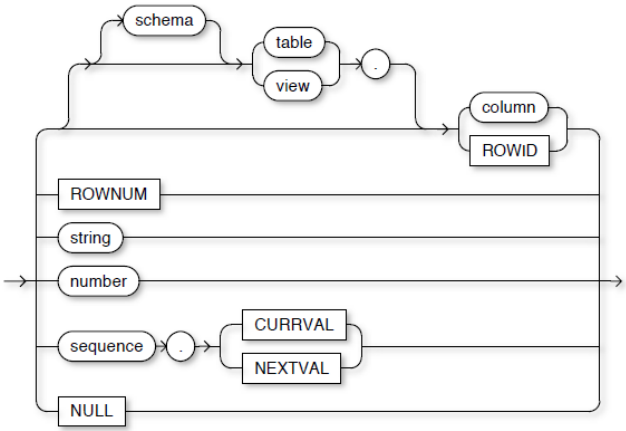
CGI

Expressions

expr ::=

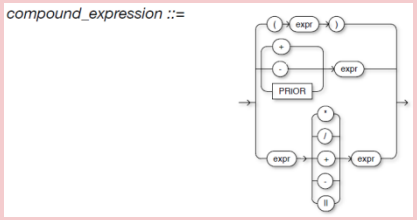


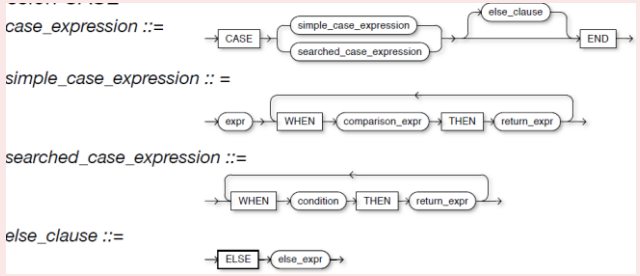
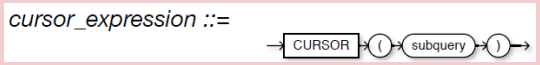
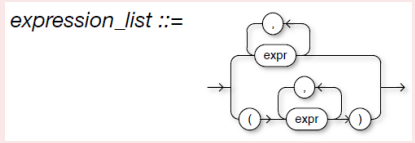
simple_expression ::=



Expressions

Schéma	Exemples

	`('Paul' || 'Dupont')`
	`LENGTH('Don')+2`
	`SQRT(121)/11`

	`SELECT nom, CASE WHEN duree < 60 THEN 'Short' ELSE 'Not short' END isShort FROM oeuvre`
	
	`WHERE (nom, prenom) IN (('Dupont', 'JEAN'), ('Durand', 'Paul'))`

Les jointures

Toujours utiliser un opérateur de jointure normalisé = « join »

Equi-jointure

- `SELECT * FROM musiciens m, œuvre as o WHERE m.nm = o.nm`

Jointure interne (par défaut)

- `SELECT * FROM TA [INNER]JOIN TB ON <cond. entre A et B>`

Jointure externe

Lorsqu'une ligne d'une table figurant dans une jointure n'a pas de correspondant dans les autres tables, elle ne satisfait pas au critère d'équi-jointure et donc ne figure pas dans le résultat de la jointure. Une option permet de faire figurer dans le résultat les lignes satisfaisant la condition d'équi-jointure plus celles n'ayant pas de correspondant.

- `SELECT * FROM musiciens m, œuvre as o WHERE m.nm(+) = o.nm`
- `SELECT * FROM TA LEFT|RIGHT|FULL OUTER JOIN TB ON <cond. entre A et B>`

Jointure naturelle

- `SELECT * FROM TA NATURAL JOIN TB [USING <noms de colonnes>]`

http://sqlpro.developpez.com/cours/sqlaz/jointures/

Les jointures

Toujours utiliser un opérateur de jointure normalisé = « join »

Les mots clefs **LEFT**, **RIGHT** et **FULL** indiquent la manière dont le moteur de requête doit effectuer la jointure externe. Il font référence à la table située à gauche (**LEFT**) du mot clef **JOIN** ou à la table située à droite (**RIGHT**) de ce même mot clef. Le mot **FULL** indique que la jointure externe est bilatérale.

Jointure croisée (produit cartésient de deux tables)

- `SELECT * FROM TA CROSS JOIN TB`

Jointure d'union (union de structures quelconque)

- `SELECT * FROM TA UNION JOIN TB`

Sélectionnez

```
SELECT *
FROM T_TITRE UNION JOIN T_TYPE
```

Sélectionnez

TIT_CODE	TIT_LIBELLE	TYP_CODE	TYP_LIBELLE
M.	Monsieur	NULL	NULL
Melle.	Mademoiselle	NULL	NULL
Mme.	Madame	NULL	NULL
NULL	NULL	FAX	Télécopie
NULL	NULL	GSM	Téléphone portable
NULL	NULL	TEL	Téléphone fixe

http://sqlpro.developpez.com/cours/sqlaz/jointures/

Les jointures

	Paradox	Access	PostGreSQL	Sybase	SQL Server 7	Oracle 8	DB2 (400)
INNER JOIN	Oui	Oui	Oui	Oui	Oui	Non (1)	Oui
OUTER JOIN	LEFT, RIGHT, FULL	LEFT, RIGHT	LEFT, RIGHT, FULL	LEFT, RIGHT, FULL	LEFT, RIGHT, FULL	Non (1)	LEFT (4)
UNION JOIN	Non (2)	Non	Non	Non (2)	Non (2)	Non (1) (2)	Non
CROSS JOIN	Non (3)	Non (3)	Oui	Non (3)	Oui	Non (1) (3)	Oui

(1) Oracle ne connaît toujours pas le **JOIN** (ça fait quand même plus de dix ans de retard pour cet éditeur pionnier qui semble s'endormir sur ses lauriers). Il faut donc utiliser une syntaxe propriétaire. Exception : la version 9 supporte enfin les jointures normalisées.

(2) Possible avec un **FULL OUTER JOIN**

(3) Possible avec l'ancienne syntaxe sans précision de critère **WHERE**

http://sqlpro.developpez.com/cours/sqlaz/jointures/

Prédicats

ALL

Compare une valeur à toutes les valeurs dans une liste ou un résultat de sous-reqûete. TRUE si aucun tuple

```
SELECT * FROM oeuvres WHERE duree >= ALL(SELECT DISTINCT duree FROM oeuvres)
```

no	nom	type	date	periode	duree	nc
18	Don Giovanni	Opéra	1787	Classique	172	110

ANY

Compare une valeur à chaque valeur dans une liste ou un résultat de sous-reqûete. FALSE si aucun tuple

```
SELECT * FROM oeuvres WHERE duree >= ANY(SELECT DISTINCT duree FROM oeuvres)
```

no	nom	type	date	periode	duree	nc
5	Concerto n°5	Concerto	1808	Classique	17	100
19	6e Symphonie	Symphonie	1805	Classique	43	100
18	Don Giovanni	Opéra	1787	Classique	172	110
10	Concerto n°2	Concerto	1829	Romantique	31	120

EXISTS: tester l'existance de tuples dans une sous reqûete

IN: tester si une valeur est dans une liste

Tables de vérité – conditions logiques

Tables de vérité

- ***NOT***

¬	TRUE	FALSE	UNKNOWN
NOT	FALSE	TRUE	UNKNOWN

- ***AND***

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

- ***OR***

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

Conditions de reconnaissances & cie

LIKE

Compare une chaine de caractère à un motif.

_ : 1 caractère ; % : 0..n caractères ; caractère d'échapement \ (_, \%)

```
SELECT * FROM compositeur WHERE nom LIKE 'Moz%'
SELECT * FROM joue WHERE lieu LIKE 'Salle\_%' ESCAPE '\'
```

REGEXP_LIKE

Compare une chaine de caractère à une expression régulière (standards POSIX).

```
SELECT * FROM musiciens WHERE REGEXP_LIKE(nom, '^Dupont(t|d)$')
```

Between

```
SELECT * FROM oeuvre WHERE duree BETWEEN 10 AND 20
```

NULL

```
SELECT * FROM musicien WHERE qualite IS [NOT] NULL
```

CRUD

On peut faire la même chose pour:

- Insérer

```
INSERT INTO musiciens VALUES(3, 'Bowi', 'David', 'Débutant')
INSERT INTO musiciens(nom, prenom) VALUES('Bowi', 'David')
INSERT INTO musiciens SET nom = 'Bowie', prenom = 'David'
```

- Modifier

```
UPDATE musiciens SET qualite = 'Expert', nom = 'Bowie' WHERE nm = 3
UPDATE musiciens SET qualite = NULL
```

- Supprimer

```
DELETE FROM musiciens
DELETE FROM musiciens WHERE conditions
```

MUSICIEN (NM, NOM, PRENOM, QUALITE)