Introduction aux bases de données et SQL

CM5 : Langage de manipulation de données (LMD)

Mickaël Martin Nevot

V1.0.0

Introduction aux bases de données et SQL

SQL

- *Structured query language* : langage de requête structurée
- **Un seul langage** général :
  - Langage de description de données (LDD)
  - Langage de manipulation de données (LMD)
  - Langage de description des schémas physiques (LDSP)
  - Contrôle et administration

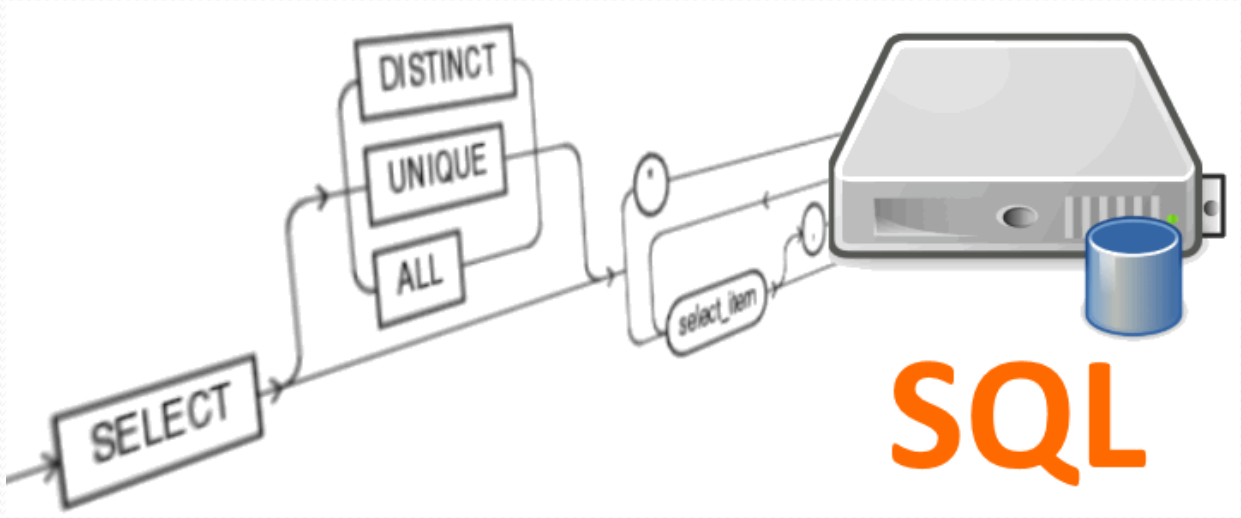
Pour tester

- Interpréteur en ligne : https://livesql.oracle.com
- Documentation : https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf

En SQL, les retours à la ligne sont non déterministes

LMD

- **Langage de manipulation de données** (LMD)
- Recherche des données d'une BD
- Mise à jour des données d'une BD



Le résultat d'une requête relationnelle est une relation

Recherche de données

- Sélection :
  - SELECT : définit les attributs de la relation résultante
  - FROM : spécifie les relations sur lesquelles porte la recherche
  - WHERE : indique des conditions de restriction
  - Les autres clauses seront traitées ultérieurement

Syntaxe :

```
[...] SELECT [...] [DISTINCT] ...
FROM ...
[WHERE ...]
[...]
[GROUP BY ...]
[HAVING ...]
[ORDER BY ...]
[...]
```

SELECT, FROM, WHERE, etc. sont des clauses

Opérateurs SQL

- **Projection**
- **Sélection**
- Union
- Différence
- Intersection
- **Produit cartésien**
- **Jointure**
- Division

Opérateurs ensemblistes

Opérateurs primitifs

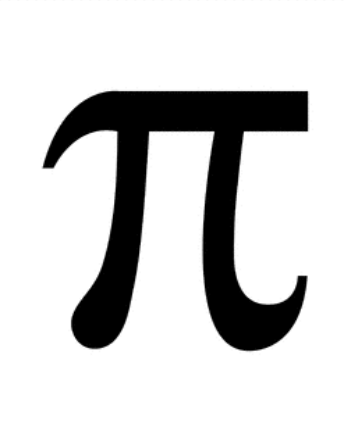
Opérateurs dérivés

Projection

- Préserve (projette) certains **attributs** d'une relation

```
SELECT nom, prenom
FROM Etudiant;

SELECT *
FROM Convention;
```

Tous les attributs sont préservés



Sélection

- Filtre certains **tuples** d'une relation (avec condition[s])

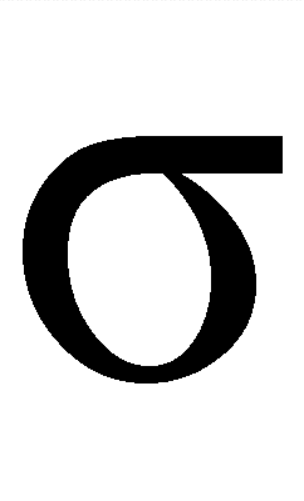
```
SELECT nom, prenom FROM Etudiant WHERE annee = 2;

SELECT * FROM Etudiant WHERE annee = 1 AND sexe = 'F';
```

- Condition :

Une condition est un prédicat

Expression logique vraie ou fausse

  - Attribut(s)
  - Constantes : 2, 'F', etc.
  - Opérateur(s) :
    - De comparaison : =, <>, >, >=, <, <=, etc.
    - Booléens : AND, OR, NOT, etc.
    - IS NULL, IN, ALL, ANY, BETWEEN, EXISTS, LIKE, etc.



Les opérateurs de comparaison peuvent être utilisés avec des attributs textuels

Opérateurs de sélection

- IN : comparaison avec un ensemble

```
SELECT nom, prenom
FROM Etudiant
WHERE annee IN (2, 3);
```

- BETWEEN : intervalle de valeur

```
SELECT nom, prenom
FROM Etudiant
WHERE annee BETWEEN 1 AND 3;
-- Equivalent à : annee >= 1 AND annee <= 3
```

- LIKE : comparaison de chaîne de caractères

```
SELECT nom, raisons
FROM Societe
WHERE adresse LIKE '_ar%';
```

Jokers : _ (un caractère)
% (n'importe quel nombre de caractères)

- IS (NOT) NULL : traitement des valeurs nulles

```
SELECT ide, ids
FROM Convention
WHERE date_deb IS NULL;
```

Projection et sélection

- Projection et sélection sont combinables

```
SELECT nom, prenom
FROM Etudiant
WHERE sexe = 'F' AND annee <= 2;

SELECT nom, raisons
FROM Societe
WHERE adresse LIKE '%ar%'
  AND activite BETWEEN 7 AND 15;
```

Et ce ne sont pas les seuls !

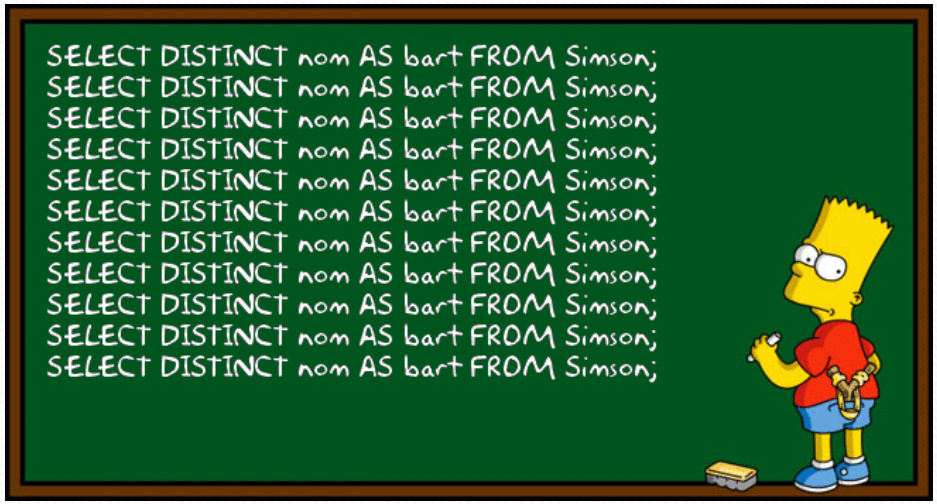
Autres mots clef de SELECT

- DISTINCT : élimination des doublons

```
SELECT DISTINCT nom
FROM Etudiant
WHERE prenom <> 'Lulu';
```

- AS : renommage :

```
SELECT raisons AS formej, adresse AS "Adresse postale"
FROM Societe
WHERE ids = 8;
```



ORDER BY

- Tri des résultats :
  - ASC : ascendant (par défaut)
  - DESC : descendant

```
SELECT nom
FROM Etudiant
WHERE prenom <> 'Lulu'
ORDER BY nom ASC;

SELECT DISTINCT nom
FROM Etudiant
WHERE annee = 2
ORDER BY nom DESC;
```

Fonctions d'agrégation

- Agrège les tuples (un seul résultat)
- Uniquement dans clause `SELECT` (ou `HAVING`)
- Fonctions :
  - `SUM(...)` : somme
  - `AVG(...)` : moyenne algébrique
  - `MIN(...)`, `MAX(...)` : valeur min., max.
  - `COUNT(...)` : dénombrement
  - Etc.

Uniquement pour des attributs numériques

Aussi appelées fonctions de calcul intégrées

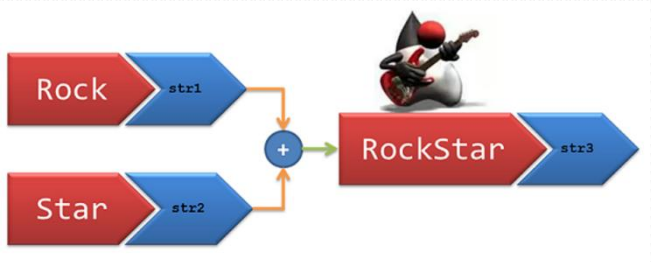
Pas de fonction d'agrégat en paramètre d'une fonction d'agrégat

Fonctions mathématiques

- Uniquement dans clause `SELECT` (ou `HAVING`) et `WHERE`
- Opérateurs : +, –, *, /, etc.
- Fonctions :
  - `ABS(x)` : valeur absolue
  - `ROUND(x, y)` : arrondi
  - `CEIL(x)`, `FLOOR(x)` : arrondi à l'inférieur, supérieur
  - `TRUNC(x)` : troncature
  - `POWER(x, y)` : puissance
  - `SIGN(x)` : signe
  - `SQRT(x)` : racine carrée
  - Etc.

Fonctions de chaînes

- Uniquement dans clause `SELECT` (ou `HAVING`) et `WHERE`
- Opérateurs : `||`
- Fonctions :
  - `LOWER(str)`, `UPPER(str)` : conversion en minuscule, majuscule
  - `REPLACE(txt, str1, str2)` : remplacement de chaîne
  - `SUBSTR(txt, i, n)` : extrait une sous-chaîne
  - `LENGTH(str)` : taille de la chaîne
  - `INITCAP(str)` : premier caractère en majuscule
  - Etc.



Fonctions date/heure

- CURRENT_DATE() : date et heure courantes
- TO_CHAR(..., txt) : conversion de date en chaîne

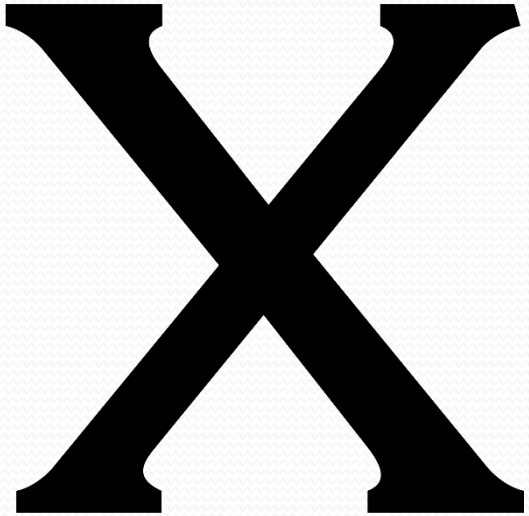
```
SELECT TO_CHAR(current_timestamp, 'HH12:MI:SS') FROM T;
```

- TO_DATE(str1, str2) : conversion de chaîne en date

```
INSERT INTO T VALUES (TO_DATE('05 Dec 2020', 'DD Mon YYYY'));
```

Produit cartésien

```
SELECT *
FROM Etudiant, Societe;
```



Très peu utile (seul)

Jointure

Permet de retrouver des données organisées sur plusieurs relation indépendantes

- Jointure ensembliste : imbrication de sous-requête(s)

```
SELECT nom
FROM Etudiant
WHERE ide IN (
    SELECT ide
    FROM Convention
    WHERE duree >= 3);
```

IN, ALL, ANY, etc.

- Jointure prédicative : utilisation de prédicats

```
SELECT nom
FROM Etudiant INNER JOIN Convention
  ON Etudiant.ide = Convention.ide
WHERE duree >= 3;
```

Version normalisée

Qualification

```
SELECT nom
FROM Etudiant E, Convention C
WHERE E.ide = C.ide AND duree >= 3;
```

Alias (à ne pas confondre avec AS)

```
SELECT E.nom, raisons
FROM Etudiant E, Convention C, Societe S
WHERE E.ide = C.ide
  AND S.ids = C.ids;
```

Généralement on effectue des équi-jointure (=), mais pas toujours (>, <, etc.)

Auto-jointure

```
SELECT nom
FROM Etudiant NATURAL JOIN Etudiant
WHERE nom = 'Dupond';

SELECT ETR.nom
FROM Etudiant ETR INNER JOIN Etudiant ETS
  ON ETR.daten = ETS.daten
WHERE ETS.nom = 'Dupond';

SELECT nom
FROM Etudiant
WHERE daten IN (
    SELECT daten
    FROM Etudiant
    WHERE nom = 'Dupond');
```

Autres opérateurs multilignes

- ANY : n'importe quel résultat de l'ensemble (imbriqué) vrai
- ALL : tous les résultats de l'ensemble (imbriqué) vrais
- =ANY : équivalent à IN
- <>ALL : équivalent à NOT IN

ETUDIANT	Nom	DateN
	Dupond	16-03-81
	Duvent	20-04-81
	Dupond	17-04-82
	Durand	18-04-82

```
SELECT nom, daten
FROM Etudiant
WHERE daten > ALL (SELECT daten
                   FROM Etudiant
                   WHERE nom = 'Dupond');
```

ETUDIANT	Nom	DateN
	Durand	18-04-82

```
SELECT nom, daten
FROM Etudiant
WHERE daten > ANY (SELECT daten
                   FROM Etudiant
                   WHERE nom = 'Dupond');
```

ETUDIANT	Nom	DateN
	Duvent	20-04-81
	Dupond	17-04-82
	Durand	18-04-82

Op. ensemblistes, parti., division

Crédits

Auteur

Mickaël Martin-Nevot
mmartin.nevot@gmail.com

- Laurent Carmignac

Carte de visite électronique

Relecteurs

Cours en ligne sur : www.mickael-martin-nevot.com