

# Langage SQL

Opération	Résultat (copies d'écran SQLiteStudio)	Requête en langage SQL
<b>Affichage</b>		SELECT * FROM Auteurs
<b>Projection</b> (Sélection de colonnes)		SELECT nom,prénom FROM Auteurs
<b>Sélection ou restriction</b> <sup>(1)</sup> (Sélection de lignes)		SELECT * FROM Livres WHERE titre = 'Aphorismes'
<b>Jointure</b> (Graphe des clés indispensable)		SELECT * FROM (Livres JOIN Genre ON Livres.genre=Genre.id)
<b>Agrégation</b> <sup>(2)</sup>		SELECT COUNT(*) FROM Emprunteurs
<b>Renommage</b>		SELECT rendu AS dateRetour FROM Emprunts
<b>Filtrage</b> <sup>(3)</sup>		SELECT * FROM Livres ORDER BY titre
<b>Regroupement</b>		SELECT genre, COUNT(*) FROM Livres GROUP BY genre
<b>Sélection dans un regroupement</b>		SELECT genre, COUNT(*) AS n FROM Livres GROUP BY genre HAVING n>1

Toutes ces opérations élémentaires peuvent être combinées entre elles pour former des requêtes complexes (voir TP).

- (1) On appelle **formule de sélection**  $F$  une formule construite à partir des attributs, des fonctions usuelles, de constantes, des opérateurs de comparaison ( $\leq, \geq, >, <, =$ ), de connecteurs logiques (**AND, OR, NOT**), LIKE, BETWEEN, IS NULL.  
Exemple : nom = « Loutard » AND date < 1/1/2011.
- (2) Fonctions d'agrégation : **COUNT, SUM, MAX, MIN, AVG** (Average = moyenne).  
Ces fonctions s'appliquent aux **attributs** d'une table et sont souvent utilisées avec les clauses **GROUP BY** (**regroupement** de n-uplets) et **HAVING** (**sélection** à l'intérieur de GROUP BY) : voir exemples en TP.
- (3) Filtrage **ORDER BY** ou ORDER BY DESC (tri croissant/décroissant), **LIMIT, OFFSET**

Autres mots clés (opérateurs ensemblistes) : **UNION, INTERSECT, EXCEPT**

Ordre des mots clés (**mots clés facultatifs entre crochets** ; ne pas mettre de crochets si le mot clé est utilisé) :

```
SELECT [DISTINCT] ... [AS] ...
FROM ... [JOIN ... ON ... ]
[WHERE [NOT]... ]
[GROUP BY ... [HAVING ... ]]
[ORDER BY ... ][DESC]] [LIMIT ... ][OFFSET...]
```

## Savoir-faire

**Pour comprendre l'élaboration des requêtes en TP, ne pas hésiter à tester des requêtes partielles pour juger des effets des perfectionnements successifs.**