

Aide Mémoire – Bases de données

- **Projection** sur (A_1, \dots, A_n) de R :

```
SELECT A1, ..., An FROM R;
```

- **Selection** selon le prédicat p de R :

```
SELECT * FROM R WHERE p;
```

- **Opérations ensemblistes** (même schéma) : $R_1 \cup R_2, R_1 \cap R_2, R_1 - R_2$.

```
SELECT ... INTERSECT SELECT ...;
SELECT ... UNION SELECT ...;
SELECT ... UNION ALL SELECT ...; -- Avec les doublons
SELECT ... EXCEPT SELECT ...; -- MINUS en MySQL
```

- **Jointure** de R_1 et R_2 selon $R_1.X = R_2.Y$:

```
SELECT * FROM R1 JOIN R2 ON R1.X = R2.Y; -- A préférer
SELECT * FROM R1, R2 WHERE R1.X = R2.Y;
```

- **Renommage** de X en A dans R :

```
SELECT R.X AS A, ... FROM R;
```

- **Produit cartésien** de R_1 et R_2 :

```
SELECT * FROM R1, R2;
```

- **Requêtes de base** :

```
SELECT ... FROM ... WHERE ...;
SELECT DISTINCT ... FROM ... WHERE ...;
SELECT ... JOIN ... ON ... WHERE ...;
SELECT ... FROM ... WHERE ... ORDER BY ...;
SELECT ... FROM ... WHERE ... ORDER BY ... DESC;
SELECT ... FROM ... WHERE ... LIMIT ... OFFSET ...;
```

- **Renommages** :

```
SELECT A1 AS nom_1, ..., An AS nom_n FROM table1 nomTable_1 ...;
```

- **Conditions** : = ; != OU <> ; < ; > ; <= ; >= ; IS NULL ; IS NOT NULL ; AND ; OR ; NOT ; BETWEEN ... AND ... ; LIKE '...' (_ remplace un caractère, % remplace un nombre quelconque de caractères.) ; IN ; NOT IN ; (comparateur) ANY ; (comparateur) ALL .

- **Fonctions d'agrégat** : SUM, MIN, MAX, AVG, COUNT.

```
SELECT fonction(attribut(s)) FROM ...;
```

(ne renvoie qu'un résultat.)

- **Existence**

```
SELECT ... FROM ... WHERE EXISTS (SELECT ...);
SELECT ... FROM ... WHERE NOT EXISTS (SELECT ...);
```

- **Limit et offset**

LIMIT n = se limiter aux n premiers tuples.

OFFSET m = sauter les m premiers tuples.

LIMIT n OFFSET m = renvoyer les tuples numéros $m+1$ à $m+n$.

- **Regroupement**

```
SELECT ... FROM ... WHERE ... GROUP BY ...;
SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ...;
```

Le contenu du SELECT doit être des attributs du GROUP BY ou des attributs identiques pour chaque élément du groupe ou des fonctions d'agrégat qui porteront sur chaque groupe.

Le contenu du HAVING porte sur des attributs n'apparaissant pas dans le GROUP BY.

Renvoie un résultat par groupe.

- **Agrégat de double niveau**

```
SELECT fonction2(attribut2)
FROM (SELECT fonction1(attribut1) AS attribut2
      FROM ... GROUP BY ...);
```