

MODULE 2 :

DRL (Data Retrieval Language - Langage d'extraction de données)

Partie I : SELECT ... FROM ...

Exercice 2.1.1 – Les requêtes suivantes fonctionnent-elles sous SQL-Server ? Si non, comment les corriger ?

N'hésitez pas à tester vos requêtes directement sous Management Studio !

```
1 SELECT last_name, first_name AS F name
2 FROM student;
3
4 SELECT last_name lname, first_name AS fname
5 FROM student;
6
7 SELECT last_name|| _ ||first_name AS name
8 FROM student;
9
10 SELECT last_name+first_name AS name, Year_result x 10 result,
11 FROM student;
```

Solution :

- Première requête, ligne 1 : il faut rajouter des guillemets (simples ou doubles) autour de « F name » ou bien remplacer l'espace par un « underscore »
- La deuxième requête fonctionne très bien car le « AS » est facultatif
- Troisième requête, ligne 7 : la concaténation se fait avec le caractère « + », sous SQL-Server, pas les « || » qu'il faut donc remplacer ; ensuite, le caractère « _ » (underscore) doit être entre guillemets simples
- Quatrième requête, ligne 10 : la multiplication se symbolise par l'étoile « * » et non le « x » qu'il faut donc remplacer ; la virgule en fin de ligne créera une erreur, elle est à supprimer

```
1 SELECT last_name, first_name AS 'F name'
2 FROM student;
3
4 SELECT last_name lname, first_name AS fname
5 FROM student;
6
7 SELECT last_name + '_' + first_name AS name
8 FROM student;
9
10 SELECT last_name+first_name AS name, Year_result * 10 result
11 FROM student;
```

Exercice 2.1.2 – Ecrire une requête pour présenter, pour chaque étudiant, le nom de l'étudiant, la date de naissance, le login et le résultat pour l'année de l'ensemble des étudiants.

```
1 SELECT last_name, birth_date, login, year_result
2 FROM student
```

Exercice 2.1.3 – Ecrire une requête pour présenter, pour chaque étudiant, son nom complet (nom et prénom séparés par un espace), son id et sa date de naissance.

```
1 SELECT last_name + ' ' + first_name as 'Nom complet', student_id, birth_date
2 FROM student
```

Exercice 2.1.4 – Ecrire une requête pour présenter, pour chaque étudiant, dans une seule colonne (nommée « Info Étudiant ») l'ensemble des données relatives à un étudiant séparées par le symbole « | ». Il est nécessaire d'avoir recours à la fonction de conversion CONVERT(type, champs).

```
1 SELECT CONVERT(VARCHAR, student_id) + '|' + last_name + '|'
2      + first_name + '|' + CONVERT(VARCHAR, birth_date) + '|'
3      + login + '|' + CONVERT(VARCHAR, section_id) + '|'
4      + CONVERT(VARCHAR, year_result) as 'Info Étudiant'
5 FROM student
```

Partie II : SELECT ... FROM ... WHERE ... ORDER BY

Exercice 2.2.1 – Ecrire une requête pour présenter le login et le résultat de tous les étudiants ayant obtenu un résultat annuel supérieur à 16

```
1 SELECT login, year_result
2 FROM student
3 WHERE year_result > 16
```

Exercice 2.2.2 – Ecrire une requête pour présenter le nom et l'id de section des étudiants dont le prénom est Georges

```
1 SELECT last_name, section_id
2 FROM student
3 WHERE first_name like 'Georges'
```

Exercice 2.2.3 – Ecrire une requête pour présenter le nom et le résultat annuel de tous les étudiants ayant obtenu un résultat annuel compris entre 12 et 16

```
1 SELECT last_name, year_result
2 FROM student
3 WHERE year_result BETWEEN 12 AND 16
```

Exercice 2.2.4 – Ecrire une requête pour présenter le nom, l'id de section et le résultat annuel de tous les étudiants qui ne font pas partie des sections 1010, 1020 et 1110

```
1 SELECT last_name, section_id, year_result
2 FROM student
3 WHERE section_id NOT IN (1010, 1020, 1110)
```

Exercice 2.2.5 – Ecrire une requête pour présenter le nom et l'id de section de tous les étudiants qui ont un nom de famille qui termine par « r »

```
1 SELECT last_name, section_id
2 FROM student
3 WHERE last_name LIKE '%r'
```

Exercice 2.2.6 – Ecrire une requête pour présenter le nom et le résultat annuel de tous les étudiants qui ont un nom de famille pour lequel la troisième lettre est un « n » et qui ont obtenu un résultat annuel supérieur à 10

```
1 SELECT last_name, year_result
2 FROM student
3 WHERE last_name LIKE '__n%'
4 AND year_result > 10
```

Exercice 2.2.7 – Ecrire une requête pour présenter le nom et le résultat annuel classé par résultats annuels décroissants de tous les étudiants qui ont obtenu un résultat annuel inférieur ou égal à 3

```
1 SELECT last_name, year_result
2 FROM student
3 WHERE year_result <= 3
4 ORDER BY year result DESC
```

Exercice 2.2.8 – Ecrire une requête pour présenter le nom complet (nom et prénom séparés par un espace) et le résultat annuel classé par nom croissant sur le nom de tous les étudiants appartenant à la section 1010

```
1 SELECT last_name + ' ' + first_name as 'Nom complet', year_result
2 FROM student
3 WHERE section_id = 1010
4 ORDER BY last_name ASC
```

Exercice 2.2.9 – Ecrire une requête pour présenter le nom, l'id de section et le résultat annuel classé par ordre croissant sur la section de tous les étudiants appartenant aux sections 1010 et 1020 ayant un résultat annuel qui n'est pas compris entre 12 et 18

```
1 SELECT last_name, section_id, year_result
2 FROM student
3 WHERE section_id IN (1010, 1020)
4     AND year_result NOT BETWEEN 12 AND 18
5 ORDER BY section_id
```

Exercice 2.2.10 – Ecrire une requête pour présenter le nom, l'id de section et le résultat annuel sur 100 (nommer la colonne « Résultat sur 100 ») classé par ordre décroissant du résultat de tous les étudiants appartenant aux sections commençant par 13 et ayant un résultat annuel sur 100 inférieur ou égal à 60

```
1 SELECT last_name, section_id, (year_result*5) as 'Résultat sur 100'
2 FROM student
3 WHERE section_id like '13__' AND year_result*5 <= 60
4 ORDER BY year result DESC
```