

TP N 2

ISTA TINGHIR
Realisé par: Groupe2
Amine ALOUANI
Abderrahmane MOUCHE

le: 12 mai 2023

Module: M205
Développer en Back-End

Plan :

- ❖ 1. Definition de Query Builder Laravel.
- ❖ 2. Les méthodes de Query Builder.
 - a. La Méthode SELECT.
 - b. La Méthode WHERE.
 - c. La Méthode ORDERING.
 - d. La Méthode GROUPING.
 - e. La Méthode LIMIT.
 - f. La Méthode OFFSET.
 - g. La Méthode UNION.
 - h. La Méthode JOIN.

Définition de Query Builder Laravel :

La Query Builder de Laravel est une fonctionnalité du framework Laravel qui permet de créer des requêtes SQL de manière programmatique en utilisant des méthodes en chaîne de type "fluent" en PHP.

La Query Builder de Laravel fournit une syntaxe élégante et facile à utiliser pour créer des requêtes SQL de manière dynamique et sécurisée, sans avoir à écrire directement des requêtes SQL. Cette fonctionnalité prend en charge toutes les opérations SQL courantes telles que SELECT, INSERT, UPDATE, DELETE, UNION, JOIN, WHERE, GROUP BY, HAVING, ORDER BY, LIMIT, OFFSET.

La Query Builder de Laravel est également très flexible, elle permet de manipuler les requêtes pour ajouter ou supprimer des clauses, ou modifier leur ordre d'exécution. Elle offre également une sécurité accrue contre les attaques d'injection SQL en utilisant des requêtes préparées et des liens de paramètres.

L'utilisation de la Query Builder de Laravel permet aux développeurs de travailler de manière plus efficace et de réduire les erreurs de syntaxe dans leurs requêtes SQL. Elle permet également de mieux organiser et structurer les requêtes SQL en les intégrant directement dans le code PHP de l'application, plutôt que de les écrire dans des fichiers SQL séparés.

En somme, la Query Builder de Laravel est une fonctionnalité puissante et flexible qui facilite la création et la maintenance de requêtes SQL dans les applications Laravel, tout en offrant une sécurité accrue contre les attaques d'injection SQL.

Les méthodes de Query Builder

La Méthode SELECT :

La méthode SELECT de la Query Builder permet de spécifier les colonnes à sélectionner dans une requête SELECT SQL. Elle peut être appelée en utilisant la syntaxe de chaîne de méthode en PHP en passant les noms des colonnes à sélectionner comme arguments.

```
14 // la methode Select :  
15 $students = DB::table('students')  
16 ->select('name', 'email', 'phone', 'course_name')  
17 ->get();  
18
```

La Méthode WHERE :

La méthode WHERE de la Query Builder permet de filtrer les résultats d'une requête SELECT SQL en utilisant des conditions spécifiées. Elle peut être appelée en utilisant la syntaxe de chaîne de méthode en PHP en passant les conditions à appliquer.

```
21 // La methode where :  
22  
23 $students = DB::table('students')  
24 ->where('name', '=', "Amine")  
25 ->get();  
26 //-----
```

La Méthode ORDERING :

La méthode "orderBy" de la Query Builder de Laravel permet de trier les résultats d'une requête SELECT SQL par ordre croissant ou décroissant en fonction d'une ou plusieurs colonnes. Elle peut être appelée en utilisant la syntaxe de chaîne de méthode en PHP en passant le ou les noms de colonnes à trier ainsi que l'ordre de tri.

```
28 //La methode ORDERING :|
29
30 $students = DB::table('students')
31     ->orderBy('name', 'desc')
32     ->get();
33
34 //-----
```

La Méthode GROUPING :

La méthode GROUPING (GroupBy) dans Query Builder de Laravel est utilisée pour regrouper les résultats d'une requête selon une ou plusieurs colonnes. Cette méthode est généralement utilisée avec des fonctions d'agrégation telles que SUM, COUNT, AVG, etc., pour effectuer des calculs sur les données regroupées.

```
35 // La methode GROUPING :
36
37 $students = DB::table('students')
38     ->groupBy('course_name')
39     ->having('course_name', '=', "mecanique")
40     ->get();
41 //-----
```

La Méthode LIMIT :

La méthode "Limit" de Query Builder de Laravel est utilisée pour limiter le nombre d'enregistrements retournés par une requête. Elle est généralement utilisée avec la méthode "orderBy" pour contrôler les enregistrements retournés.

```
3 // La méthode LIMIT & OFFSET : Limit is important
4 //limit :
5
6 $students = DB::table('students')
7     ->limit(3)
8     ->get();
9
```

La Méthode OFFSET :

La méthode "offset" de Query Builder de Laravel est utilisée pour sauter un nombre spécifié d'enregistrements avant de retourner les enregistrements restants. Elle est généralement utilisée en combinaison avec la méthode "limit" pour contrôler le nombre d'enregistrements retournés et la position de départ des enregistrements retournés.

```
50 //offset :
51 $students = DB::table('students')
52     ->offset(1)
53     ->limit(10)
54     ->get();
55 //-----
```

La Méthode UNION :

La méthode UNION de la Query Builder permet de combiner le résultat de deux ou plusieurs requêtes SELECT SQL en une seule. Elle peut être appelée en utilisant la syntaxe de chaîne de méthode en PHP en passant les requêtes SELECT à combiner.

```
57 // La methode Union:
58
59 $first = DB::table('students')
60     ->where('email',"=", "");
61
62 $students = DB::table('students')
63     ->where('phone','=', "")
64     ->union($first)
65     ->get();|
66
67
68 //
```

La Méthode JOIN :

La méthode JOIN de la Query Builder permet de joindre deux ou plusieurs tables dans une requête SQL en utilisant une clause JOIN. Elle peut être appelée en utilisant la syntaxe de chaîne de méthode en PHP en passant les noms des tables à joindre et les conditions de jointure.

```
70 // La methode Joins
71
72 $students = DB::table('students')
73     ->join('courses', 'students.course_id', '=', 'courses.id')
74     ->select('students.*', 'courses.name as courses_name')
75     ->get();|
76
77
```