

Assignment 17: Query Builder in Laravel

01. Explain what Laravel's query builder is and how it provides a simple and elegant way to interact with databases.

Laravel's query builder is an abstraction layer that provides a simple and elegant way to interact with databases. It can be used to perform most database operations in application and works on all supported database systems.

02. Write the code to retrieve the "excerpt" and "description" columns from the "posts" table using Laravel's query builder. Store the result in the \$posts variable. Print the \$posts variable.

```
$posts = DB::table('posts')->select('title')->distinct()->get();  
print_r($posts);
```

03. Describe the purpose of the distinct() method in Laravel's query builder. How is it used in conjunction with the select() method?

The distinct() method in Laravel's query builder is used to retrieve unique records from a query result. It ensures that the returned result set contains only distinct rows by removing duplicates. It is often used in conjunction with the select() method to specify which columns to select and apply the distinct filter to.

For example:

```
$posts = DB::table('posts')->select('title')->distinct()->get();
```

04. Write the code to retrieve the first record from the "posts" table where the "id" is 2 using Laravel's query builder. Store the result in the \$posts variable. Print the "description" column of the \$posts variable.

```
$posts = DB::table('posts')->where('id', 2)->first();  
echo $posts->description;
```

05. Write the code to retrieve the "description" column from the "posts" table where the "id" is 2 using Laravel's query builder. Store the result in the \$posts variable. Print the \$posts variable.

```
$posts = DB::table('posts')->where('id', 2)->value('description');  
echo $posts;
```

06. Explain the difference between the first() and find() methods in Laravel's query builder. How are they used to retrieve single records?

The first() method is used to retrieve the first record that matches the query conditions, while the find() method is used to retrieve a record by its primary key.

first() example:

```
$posts = DB::table('posts')->where('published', true)->first();
```

find() example:

```
$posts = DB::table('posts')->find(2);
```

07. Write the code to retrieve the "title" column from the "posts" table using Laravel's query builder. Store the result in the \$posts variable. Print the \$posts variable.

```
$posts = DB::table('posts')->pluck('title');  
print_r($posts);
```

08. Write the code to insert a new record into the "posts" table using Laravel's query builder. Set the "title" and "slug" columns to 'X', and the "excerpt" and "description" columns to 'excerpt' and 'description', respectively. Set the "is_published" column to true and the "min_to_read" column to 2. Print the result of the insert operation.

```
$result = DB::table('posts')->insert([  
    'title' => 'Test title',  
    'slug' => 'test-title',  
    'excerpt' => 'test-excerpt',  
    'category' => 'test-category',  
    'description' => 'test-description',  
    'is_published' => true,  
    'min_to_read' => 2,  
    'visited_count' => 0,  
    'author' => 'test-author',  
]);  
echo $result;
```

09. Write the code to update the "excerpt" and "description" columns of the record with the "id" of 2 in the "posts" table using Laravel's query builder. Set the new values to 'Laravel 10'. Print the number of affected rows.

```
$affectedRows = DB::table('posts')
->where('id', 2)
->update([
    'excerpt' => 'Laravel 10',
    'description' => 'Laravel 10'
]);
echo $affectedRows;
```

10. Write the code to delete the record with the "id" of 3 from the "posts" table using Laravel's query builder. Print the number of affected rows.

```
$affectedRows = DB::table('posts')->where('id', 3)->delete();
echo $affectedRows;
```

11. Explain the purpose and usage of the aggregate methods `count()`, `sum()`, `avg()`, `max()`, and `min()` in Laravel's query builder. Provide an example of each.

```
$count = DB::table('posts')->count();
$sum = DB::table('posts')->sum('visited_count');
$average = DB::table('posts')->avg('visited_count');
$maxValue = DB::table('posts')->max('visited_count');
$minValue = DB::table('posts')->min('visited_count');
```

12. Describe how the `whereNot()` method is used in Laravel's query builder. Provide an example of its usage.

```
$posts = DB::table('posts')->whereNot('category', 'technology')->get();
```

13. Explain the difference between the `exists()` and `doesn'tExist()` methods in Laravel's query builder. How are they used to check the existence of records?

```
$exists = DB::table('posts')->where('author', 'John')->exists();
```

14. Write the code to retrieve records from the "posts" table where the "min_to_read" column is between 1 and 5 using Laravel's query builder. Store the result in the `$posts` variable. Print the `$posts` variable.

```
$affectedRows = DB::table('posts')->where('id', 3)->increment('min_to_read');
echo $affectedRows;
```