Assignment - 17

Ans - 1:

Laravel's query builder is a powerful feature that provides a simple and elegant way to interact with databases. It allows developers to build database queries using a fluent, chainable interface instead of writing raw SQL queries. The query builder provides methods for constructing queries, selecting columns, specifying conditions, joining tables, ordering results, and more. It abstracts the underlying database system, making it easier to work with different database engines without changing the code.

Ans -2 :

```
$posts = DB::table('posts')

        ->select('excerpt', 'description')
        ->get();
print_r($posts);
```

Ans - 3 :

The distinct() method in Laravel's query builder is used to retrieve distinct rows from a table based on a specified column. It eliminates duplicate rows from the result set. It is often used in conjunction with the select() method to retrieve distinct values from a specific column.

For example, if you want to retrieve distinct values from the "category" column of the "posts" table, you can use the following code:

```
$distinctCategories = DB::table('posts')

    ->select('category')
    ->distinct()
    ->get();
```

Ans - 4 :

```
$posts = DB::table('posts')

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

Ans - 5 :

```
$posts = DB::table('posts')

        ->where('id', 2)
        ->pluck('description');
print_r($posts);
```

Ans - 6 :

The first() and find() methods in Laravel's query builder are used to retrieve single records, but they have slight differences:

- **first() method retrieves the first record matching the query conditions. It returns a single object or null if no matching record is found. It is often used when you need to retrieve a single record based on certain conditions.**
- **find() method retrieves a record by its primary key. It returns a single object or null if the record is not found. It is commonly used when you know the primary key value and want to retrieve the corresponding record.**

Example usage of first():

```
$firstPost = DB::table('posts')
            ->where('category', 'News')
            ->first();
```

Example usage of find():

```
$post = DB::table('posts')

        ->find(1);
```

Ans - 7 :

```
$posts = DB::table('posts')

        ->pluck('title');
         print_r($posts);
```

Ans - 8 :

```
$result = DB::table('posts')->insert([
            'title' => 'X',
            'slug' => 'X',
            'excerpt' => 'excerpt',
            'description' => 'description',
            'is_published' => true,
            'min_to_read' => 2
          ]);
print_r($result);
```

Ans - 9 :

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

Ans - 10 :

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

Ans -11 :

Aggregate methods in Laravel's query builder are used to perform calculations on a set of values from the database. Here's an explanation of each method with an example:

- **count(): Returns the number of rows matching the query.**
  $count = DB::table('posts')->count();

- **sum(): Calculates the sum of a column's values.**
  $total = DB::table('orders')->sum('amount');

- **avg(): Calculates the average of a column's values.**

  $average = DB::table('products')->avg('price');

- **max(): Retrieves the maximum value of a column.**

  $maxPrice = DB::table('products')->max('price');

- **min(): Retrieves the minimum value of a column.**

  $minPrice = DB::table('products')->min('price');

Ans - 12 :

The whereNot() method in Laravel's query builder is used to add a "not equal" condition to a query. It retrieves records where the specified column value is not equal to the provided value.

Here's an example of its usage:

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

In this code, the whereNot() method is used to retrieve records where the "category" column value is not equal to 'News'. This is useful when you want to exclude certain values from the result set.

Ans - 13 :

The exists() and doesntExist() methods in Laravel's query builder are used to check the existence of records in a table:

- **exists(): Returns true if the query has any matching records, and false otherwise.**

  ```
  $exists = DB::table('posts')->where('id', 1)->exists();
  ```

- **doesntExist(): Returns true if the query does not have any matching records, and false otherwise.**

  ```
  $doesntExist = DB::table('posts')->where('id', 1)->doesntExist();
  ```

Ans - 14 :

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 would be:

```
$posts = DB::table('posts')
        ->whereBetween('min_to_read', [1, 5])
        ->get();
print_r($posts);
```

Ans - 15 :

The code to increment the "min_to_read" column value of the record with the "id" of 3 in the "posts" table by 1 using Laravel's query builder would be:

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