Document

Question:1

Laravel's query builder is a powerful feature that allows developers to interact with databases using a fluent and intuitive syntax. It provides a simple and elegant way to construct database queries and perform various operations such as retrieving, inserting, updating, and deleting data.

We can write database queries in a more readable and maintainable manner compared to writing raw SQL statements. It abstracts the underlying database system, allowing you to write database-agnostic code that can be easily switched between different database engines, such as MySQL, PostgreSQL, SQLite, and SQL Server.

Question 2:

Retrieve the "excerpt" and "description" columns data using laravel query builder.

Query Builder Short:

```
1 reference | 0 overrides
public function excerptDescription()
{
    $posts = DB::table('posts')
        ->select('excerpt', 'description')
        ->get();

    return $posts;
}
```

Document

Question 3:

Distinct Method:

The distinct method in Laravel's query builder is used to retrieve unique values from a specific column or a combination of columns in the result set. It ensures that duplicate values are eliminated, and only distinct values are returned.

Short:

```php
//    3
    1 reference | 0 overrides
    public function distinctSelectMethod()
    {
        $posts = DB::table('posts')->select('min_to_read')->distinct()->get();
        return $posts;
    }
```

In this case, we retrieve distinct combinations of min_to_read from the "posts" table. The select('min_to_read') method specifies the columns we want to select, and the distinct() method ensures that only unique combinations of those columns are returned.

Question 4:

Retrieve the first record from the "posts" table where the "id" is 2 using Laravel's query builder

Short:

```php
// 4
    1 reference | 0 overrides
    public function retriveFristData ()
    {
        $posts = DB::table('posts')
            ->where('id', 2)
            ->first();
        return $posts->description;
    }
```

Document

Question 5:

Retrieve the "description" column from the "posts" table where the "id" is 2 using Laravel's query builder

short:

```php
// 5
    1 reference | 0 overrides
    public function description()
    {
        $posts = DB::table('posts')
            ->where('id', 2)
            ->value('description');
        return $posts;
    }
```

Question 6:

Difference between the first() and find() method

first() method: The first() method retrieves the first record that matches the query conditions. It is typically used when you want to retrieve the first occurrence of a record based on the specified conditions.

find() method: The find() method is used to retrieve a record based on its primary key value. It is commonly used when you know the specific identifier (primary key) of the record we want to retrieve.

Document

Question 7:

Retrieve the "title" column from the "posts" table using Laravel's query builder

Short:

```php
// 7
1 reference | 0 overrides
public function titleColumn()
{
    $posts = DB::table('posts')
        ->pluck('title');

    return $posts;
}
```

In the output we got all data form title column.

Question 8:

Insert new data using laravel query builder:

```php
// 8
1 reference | 0 overrides
public function insertData()
{

    $posts = DB::table('posts')->insert([
        'title' => 'X',
        'slug' => 'X',
        'excerpt' => 'excerpt',
        'description' => 'description',
        'is_published' => true,
        'min_to_read' => 3,
    ]);
    return $posts;
}


//9
```

Document

Question 9:

Update the "excerpt" and "description" columns using using laravel query builder:

Short:

```php
//9
    1 reference | 0 overrides
    public function update()
    {
        $update = DB::table('posts')
            ->where('id',  2)
            ->update([
                'excerpt'     => 'Laravel 10',
                'description' => 'Laravel 10',
            ]);
        return $update;
    }
```

Question 10:

Delete data using laravel query builder:

```php
//     10
    1 reference | 0 overrides
    public function delete()
    {
        $delete = DB::table('posts')
            ->where('id', 3)
            ->delete();
        return $delete;
    }
```

Document

Question 11:

1. count() : The count() method is used to calculate the number of records that match a specific condition. It returns the count as an integer value.

2. sum(): The sum() method calculates the sum of a numeric column in the selected records. It returns the sum as a numeric value.

3. avg(): The avg() method calculates the average value of a numeric column in the selected records. It returns the average as a numeric value.

4. max(): The max() method retrieves the maximum value from a column in the selected records. It returns the maximum value of the column.

5. min(): The min() method retrieves the minimum value from a column in the selected records. It returns the minimum value of the column.

Question 12:

The whereNot() method can be used in two ways:

1. Simple whereNot() use: The whereNot() method adds a "not equal" condition to the query, filtering out records with the specified value.

2. whereNot() with an array of values: The whereNotIn() method is used to specify an array of values to compare against, and records with any of those values in the column will be excluded from the result.

Question 13:

The difference between the exists() and doesntExist() methods in Laravel's query builder

1. exists() method: The exists() method is used to check if any records exist in the result set of a query. It returns a boolean value (true or false) indicating whether the query has any matching records.

2. doesntExist() method: The doesntExist() method is the opposite of the exists() method. It is used to check if a query does not have any matching records. It also returns a boolean value (true or false).

Document

Question 14:

Retrieve records from the "posts" table where the "min_to_read" column is between 1 and 5 using Laravel's query builder;

Short:

```php
//14

1 reference | 0 overrides
public function minReadData()
{
    $posts = DB::table('posts')->whereBetween('min_to_read', [1, 4])->get();
    return $posts;
}
```

Question 15:

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.

Short:

```php
//15
1 reference | 0 overrides
public function incrementByOne()
{
    $posts = DB::table('posts')
        ->where('id', 3)
        ->increment('min_to_read');
    return $posts;
}
```