

```
// Q1
```

The query builder acts as a wrapper around the database connection and provides a set of methods that can be chained together to construct complex queries. It supports various types of database operations, such as selecting data, inserting, updating, and deleting records.

Laravel's query builder seamlessly integrates with the Eloquent ORM (Object-Relational Mapping) system, which provides an ActiveRecord implementation in Laravel. The query builder uses a natural and intuitive syntax that closely resembles SQL, making it easy to understand and work with. It provides methods like `select`, `where`, `orderBy`, and `join`, which allow you to build complex queries step by step.

```
// Q2
```

```
$posts = DB::table('posts')->select('excerpt', 'description')->get();  
  
print_r($posts);
```

```
// Q3
```

`distinct()` is used for finding distinct element is selected row.

```
//Q4
```

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

```
//Q5
```

```
$posts = DB::table('posts')->select('description')->where('id', 2)->get();  
  
print_r($posts->description);
```

```
// Q6
```

The first() method is used to retrieve the first record that matches the query criteria.

The find() method is used to retrieve a record by its primary key value

```
// Q7
```

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

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

```
// Q8
```

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

```
print_r($result);
```

```
// Q9
```

```
$posts = new Post();  
$posts->except = "Laravel 10";  
$posts->description = "dshdfhsf sdfhs ";  
$posts->update();
```

```
// Q10
```

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

```
echo "Number of affected rows: " . $affectedRows;
```

```
// Q11
```

```
$count = DB::table('users')->count();
```

find row match will contain criteria

```
$totalSales = DB::table('orders')->sum('amount');
```

find sum of specific column value

```
$averageScore = DB::table('scores')->avg('score');
```

find avg of specific column value;

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

find max of specific column value;

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

find min of specific column value;

```
// Q12
```

```
$users = DB::table('users')
->select('name', 'email')
->whereNot('status', 'active')
->get();
```

The whereNot() method takes two arguments: the column name and the value or an array of values to be excluded

```
// Q13
```

The exists() method is used to check if any records exist in the specified table that match the query criteria.

It returns true if at least one matching record is found; otherwise, it returns false.

The doesntExist() method is used to check if no records exist in the specified table that match the query criteria.

It returns true if no matching record is found; otherwise, it returns false.

```
// Q14
```

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

```
print_r($posts);
```

```
// Q14
```

```
$affectedRows = DB::table('posts')  
->where('id', 3)  
->increment('min_to_read', 1);
```

```
echo "Number of affected rows: " . $affectedRows;
```