```
//==============1=================
```

Laravel's query builder is a feature that allows developers to build database queries using a fluent and expressive syntax in PHP. It provides a simple and elegant way to interact with databases by abstracting the underlying SQL statements and providing a more intuitive and developer-friendly interface.

```
//==========answer to the question no 2==================
$posts = DB::table('posts')
->select('excerpt', 'description')
->get();

return($posts);
```

```
//===============3===============
```

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

When used in conjunction with the select() method, the distinct() method specifies that the select query should return unique values for the specified columns.

```
//==========answer to the question no 4==================
$posts = DB::table('posts')->where('id', 2)->first();
return $posts;
Your paragraph text
//==========answer to the question no 5==================
$posts = DB::table('posts')->where('id', 2)->pluck('description')->first();
return $posts;
```

```
//===============6===============
```

In Laravel's query builder, the first() and find() methods are used to retrieve single records from a database table, but they differ in their usage and behavior.

first(): The first() method is used to retrieve the first record that matches the query conditions. It returns a single object representing the first matching record found. It is commonly used when you want to retrieve a single record based on specific conditions.
Example usage of first():

```
$user = DB::table('users')->where('name', 'John')->first();
```

find(): The find() method is used to retrieve a record based on its primary key value. It expects the primary key value as an argument and returns a single object representing the matching record. It is commonly used when you want to retrieve a record by its unique identifier.
Example usage of find():

```
$user = DB::table('users')->find(1);
```

```
//==========answer to the question no 7==================
$posts = DB::table('posts')->pluck('title');
return $posts;

//==========answer to the question no 8==================
$result=DB::table('posts')->insert([

'title' => 'X',
'slug' => 'X',
'excerpt' => 'excerpt',
'description' => 'description',
'is_published' => true,
'min_to_read' => 2
]);
return $result;

//==========answer to the question no 9==================
$data = [
'excerpt' => 'Laravel 10',
'description' => 'Laravel 10'
];

$data_1 = DB::table('posts')->where('id', 2)->update($data);
return $data_1;

//==========answer to the question no 10==================
$data = DB::table('posts')->where('id', 3)->delete();
return $data;
```

```
//===============11===============
```

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

sum(): The sum() method is used to calculate the sum of a specific column's values.

avg(): The avg() method is used to calculate the average value of a specific column.

max(): The max() method is used to retrieve the maximum value of a specific column.

min(): The min() method is used to retrieve the minimum value of a specific column.

```
//===============12===============
```

whereNot() method takes two arguments: the column name and the value or an array of values to exclude from the query results. It is commonly used to filter out records that do not meet certain criteria.

example the whereNot() method:

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

```
//===============13===============
```

exists(): The exists() method is used to check if any records exist in the query result. It returns a boolean value indicating whether there are matching records or not.
Example usage of exists():

```
$exists = DB::table('users')->where('status', 'active')->exists();
```

doesntExist(): The doesntExist() method is used to check if no records exist in the query result. It returns a boolean value indicating whether there are no matching records.
Example usage of doesntExist():

```
$doesntExist = DB::table('users')->where('status', 'inactive')->doesntExist();

//==========answer to the question no 14==================
$posts = DB::table('posts')
->whereBetween('min_to_read', [1, 5])
->get();

return($posts);

//==========answer to the question no 15==================
$data = DB::table('posts')
->where('id', 3)
->increment('min_to_read', 1);

return $data;
```