

Ans to the Ques no – 1

Laravel's query builder is a feature that simplifies working with databases in the Laravel framework. It provides a fluent interface for building and executing database queries, abstracting the underlying SQL syntax. With a chainable API, it allows developers to construct queries concisely and readably. The query builder is database-agnostic, supporting multiple database systems. It offers automatic parameter binding for security against SQL injection attacks. Integration with the Eloquent ORM enables seamless transition between the query builder and object-oriented database operations. Additionally, query logging aids in debugging and optimizing database interactions. Overall, Laravel's query builder offers a simple and elegant way to interact with databases, promoting efficient and maintainable code.

Ans to the Ques no – 2

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

print_r($posts);
```

Ans to the Ques no – 3

The method `distinct()` is used to retrieve distinct values from a specific column in a database table. It helps to eliminate duplicate records and show values only once.

Conjunction with `select()` method example-

```
$uniqueNames = DB::table('users')
->select('name')
->distinct()
->get();
```

Ans to the Ques no – 4

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

if ($posts) {
    echo $posts->description;
}
```

Ans to the Ques no – 5

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

print_r($posts);
```

Ans to the Ques no – 6

The first() method is used to retrieve the first record that matches the query conditions. It is typically used when you want to fetch a single record based on specific criteria, such as filtering by a column value or applying sorting. It returns an object representing the first matching record or null if no matching record is found.

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

The find() method is used to retrieve a record by its primary key. It takes the primary key value as a parameter and returns the corresponding record as an object. It is commonly used when you know the primary key value of the record

you want to retrieve. If no record is found with the specified primary key, it returns null.

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

Ans to the Ques no – 7

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

print_r($posts);
```

Ans to the Ques no – 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 to the Ques no – 9

```
$affectedRows = DB::table('posts')
->where('id', 2)
```

```
->update([
    'excerpt' => 'Laravel 10',
    'description' => 'Laravel 10',
]);

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

Ans to the Ques no – 10

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

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

Ans to the Ques no – 11

The count() method is used to get the number of records or the count of specific column values.

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

The sum() method calculates the sum of values in a specific column.

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

The avg() method calculates the average value of a specific column.

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

The max() method retrieves the maximum value from a specific column.

```
$highestScore = DB::table('scores')->max('score');
```

The min() method retrieves the minimum value from a specific column.

```
$lowestAge = DB::table('users')->min('age');
```

Ans to the Ques no – 12

In Laravel's query builder, the whereNot() method is used to add a "not equal" condition to a query. It allows you to specify a column and a value that should not match in order for a record to be included in the query results.

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

Ans to the Ques no – 13

The exists() method is used to check if records exist based on a query. It returns true if there is at least one record that matches the query conditions, and false otherwise. It is commonly used when you want to verify if records exist before performing certain actions.

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

The doesntExist() method is the inverse of exists(). It returns true if no records match the query conditions, and false if there is at least one matching record. It is commonly used when you want to check if no records exist before taking specific actions.

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

Ans to the Ques no – 14

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

print_r($posts);
```

Ans to the Ques no – 15

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

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