

////////// 1 //////////

Laravel Query Builder:-

Laravel Query Builder is feature of laravel which used for querying data from database without writing raw sql code. It provide some built in function for querying.

It is actually simple to use and also elegant because we can query once by laravel query builder and fetch data from different database like MySQL, PostgreSQL, SQLite etc.

////////// 2 //////////

```
use Illuminate\Support\Facades\DB;
```

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

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

////////// 3 //////////

The purpose of distinct() method in laravel query builder is to retrieve only distinct value of a column in the result set of query. It eliminates duplicated value. Only one instance of each value is returned.

use case:-

```
use Illuminate\Support\Facades\DB;
```

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

```
print_r($uniqueNames);
```

////////// 4 //////////

```
use Illuminate\Support\Facades\DB;
```

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

```
if ($posts) {  
    echo $posts->description;  
} else {  
    echo "No record found."  
}
```

////////// 5 //////////

```
use Illuminate\Support\Facades\DB;
```

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

```
echo $posts;
```

////////// 6 //////////

first() method is used when we need first record from a queried result and find() method used when we know the primary key and we want record of that primary key.

////////// 7 //////////

```
use Illuminate\Support\Facades\DB;
```

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

```
->select('title')
```

```
->get();
```

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

////////// 8 //////////

```
use Illuminate\Support\Facades\DB;
```

```
$result = DB::table('posts')->insert([
```

```
    'title' => 'X',
```

```
    'slug' => 'X',
```

```
    'excerpt' => 'excerpt',
```

```
    'description' => 'description',
```

```
    'is_published' => true,
```

```
    'min_to_read' => 2,
```

```
]);
```

```
echo $result ? "Successful!" : "Failed";
```

////////// 9 //////////

```
use Illuminate\Support\Facades\DB;
```

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

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

////////// 10 //////////

```
use Illuminate\Support\Facades\DB;
```

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

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

////////// 11 //////////

count() method returns the total count of rows of a query result

sum() method calculate summation of all number of a column

avg() method calculate average of all number of a column

max() method return maximum value from a column

min() method return minimum value from a column

///////// 12 //////////

whereNot() method used when we want to retrieve certain value which do not match a specific condition.

use :-

```
use Illuminate\Support\Facades\DB;
```

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

this code will return all the records which status column value is not equal to "active".

///////// 13 //////////

exists() method returns true if any records match the query condition and false if not.

use :-

```
use Illuminate\Support\Facades\DB;
```

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

doesn'tExist() method returns false if any records match the query condition and true if not.

```
use Illuminate\Support\Facades\DB;
```

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

///////// 14 //////////

```
use Illuminate\Support\Facades\DB;
```

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

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

///////// 15 //////////

```
use Illuminate\Support\Facades\DB;
```

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

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