1. Laravel's query builder is a fluent, expressive interface for building and executing database queries in Laravel. It provides a simple and elegant way to interact with databases by allowing developers to construct queries using chainable methods instead of writing raw SQL statements. The query builder abstracts the underlying database engine and provides a unified syntax for interacting with different database systems. It offers various methods for selecting, inserting, updating, and deleting records, as well as for applying conditions, ordering results, joining tables, and more. With the query builder, developers can write database queries in a more readable and maintainable manner, reducing the risk of SQL injection and promoting database agnostic code.
2. $posts = DB::table('posts')->select('excerpt', 'description')->get();
   print_r($posts);
3. $distinctValues = DB::table('table_name')

    ->select('column_name')

    ->distinct()
    ->get();

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

```
if ($posts) {

   echo $posts->description;

}
```

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

print_r($posts);

6. $post = DB::table('posts')->where('category', 'news')->orderBy('created_at', 'desc')->first();

7. $posts = DB::table('posts')->pluck('title');

print_r($posts);

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

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

   'slug' => 'X',

   'excerpt' => 'excerpt',

   'description' => 'description',

   'is_published' => true,

   'min_to_read' => 2

]);
```

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

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

echo $affectedRows;
11.
 Ex. $totalPosts = DB::table('posts')->count();
echo $totalPosts;
ex. $totalLikes = DB::table('posts')->sum('likes');
echo $totalLikes;
ex. $averagePrice = DB::table('products')->avg('price');
echo $averagePrice;
ex. $maxScore = DB::table('students')->max('score');
echo $maxScore;
ex. $minAge = DB::table('users')->min('age');
echo $minAge;
12.
$posts = DB::table('posts')->whereNot('status', 'published')->get();
13.
$exists = DB::table('users')->where('email', 'john@example.com')->exists();
if ($exists) {
    echo 'User exists';
} else {
    echo 'User does not exist';
}
```

```
Ex. $doesntExist = DB::table('users')->where('role', 'admin')->doesntExist();

if ($doesntExist) {

    echo 'No admin users found';

} else {

    echo 'Admin users found';

}
```

14.

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

print_r($posts);
```

15.

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

echo $affectedRows;
```