All Question Answers:

Question 1:

Laravel's query builder is a feature in the Laravel framework that makes it easy to interact with databases. It provides a simple and elegant way to write database queries using a clean and readable syntax.

With the query builder, you can fetch, insert, update, and delete data from your database tables. It has a fluent API, meaning you can chain methods together to construct your queries in a natural and intuitive way.

The query builder is designed to work with different database systems, and it handles parameter binding automatically, ensuring the security of your queries. It integrates seamlessly with Laravel's Eloquent ORM, allowing you to work with database data as objects.

The query builder also supports advanced operations like aggregates, joins, and subqueries, giving you powerful capabilities to manipulate and retrieve data.

In summary, Laravel's query builder simplifies database interactions by providing a user-friendly and secure way to work with databases, making it a popular choice for developers building web applications.

Question 2:

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

print_r($posts);
```

Question 3:

The distinct() method in Laravel's query builder is used to retrieve unique records from a database table. It ensures that the result set contains only distinct (unique) values for the specified columns.

When used in conjunction with the select() method, the distinct() method modifies the query to retrieve unique values based on the selected columns.

Question 4:

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

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

Question 5:

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

print_r($posts);
```

Question 6:

```
$user = DB::table('users')
           ->where('status', 'active')
           ->orderBy('created_at')
           ->first();
$user = DB::table('users')->find(1);
```

Question 7:

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

print_r($posts);
```

Question 8:

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

print_r($result);
```

Question 9:

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

echo $affectedRows;
```

Question 10:

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

echo $affectedRows;
```

Question 14:

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

print_r($posts);
```

Question 15:

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

echo $affectedRows;
```