# **TASK:** Query Builder in Laravel

Task 1: **Explain what Laravel's query builder is and how it provides a simple and elegant way to interact with databases.**

## Explanation:

Laravel's query builder is a database abstraction layer that allows developers to interact with databases using a simple and elegant API. It provides a fluent, chainable interface for building and executing database queries, eliminating the need for writing raw SQL statements.

With the query builder, you can perform various operations like selecting, inserting, updating, and deleting data. The fluent interface allows you to chain methods together, making the code concise and readable.

The query builder also ensures database agnosticism, meaning you can switch between different database systems without changing your code. It handles parameter binding, protecting against SQL injection attacks, and provides query logging and debugging capabilities. Overall, Laravel's query builder simplifies database interactions by abstracting the complexity of SQL, offering a clean and intuitive syntax. It enhances productivity, improves code maintainability, and provides a convenient way to work with databases in Laravel applications.

Here's a description of the functional work for each question:

2. **getExcerptDescription**: This function retrieves the "excerpt" and "description" columns from the "posts" table using Laravel's query builder. It selects only these two columns and returns the result as a collection of posts.
3. **firstRecordDescription**: This function retrieves the first record from the "posts" table where the "id" is 2 using Laravel's query builder. It fetches the "description" column of the selected record and returns it.
4. **getDescription**: This function retrieves the "description" column from the "posts" table where the "id" is 2 using Laravel's query builder. It returns the result as a collection.
5. **getAllTitle**: This function retrieves the "title" column from the "posts" table using Laravel's query builder. It fetches all the titles and returns them as a collection.
6. **insertPost**: This function inserts a new record into the "posts" table using Laravel's query builder. It sets the values for various columns like "title", "slug", "excerpt", "description", "is_published", and "min_to_read". It returns the result of the insert operation.
7. **update**: This function updates the "excerpt" and "description" columns of the record with the "id" of 2 in the "posts" table using Laravel's query builder. It sets the new values for these columns and returns the number of affected rows.
8. **delete**: This function deletes the record with the "id" of 3 from the "posts" table using Laravel's query builder. It returns the number of affected rows.
9. **getPostByMinRead**: This function retrieves records from the "posts" table where the "min_to_read" column is between 1 and 5 using Laravel's query builder. It returns the result as a collection.
10. **increments**: This function increments the value of the "min_to_read" column of the record with the "id" of 4 in the "posts" table by 1 using Laravel's query builder. It returns the number of affected rows.