COMPARISON ON ORM VS PLAIN SQL

SQL

- Databases are the core of storing state for almost all web applications. The way to interact with most relational databases is SQL — the Structured Query Language.
- SQL makes it incredibly simple to switch the actual database system or the client using that database. With SQL, a developer can pull information from a single table, or from a range of tables using joins. A join condition pulls data from multiple tables, based on specific columns.

  For example, a database containing customer information might be called "Customers." This table would have fields for unique information, such as the customer name, date of birth, a unique customer ID,

  A separate table perhaps called "CustomerPhone" could contain every phone number associated with a given customer. This second table would be linked to the first via the customer ID field. A join statement could pull information from both tables, capturing all records that match that customer ID.
- Needless to say, however, becoming proficient with SQL takes time and effort. Like any programming language, SQL has its own rules, syntax, and requirements. One small mistake can completely break an otherwise well-written query.

ORM: The Alternative Method

- Object-Relational Mapping (ORM) is another way of interacting with a database. Rather than using SQL to interact with the database, ORM provides a method of interacting with a database using an object-oriented language.

This functionality is usually provided through a library, framework or API that works with an object-oriented language. As a result, there is a level of abstraction between the database and the programmer, with the library acting as the go-between.

Advantages and Disadvantages of ORM

- ➢ ORM allows a programmer to interact with a relational database without becoming an expert in SQL.
- ➢ Second, it allows a developer to use the language of their choice. This offers significant advantages for developers who are constantly switching back-and-forth between different types of databases.
- ➢ Another advantage of ORM is that it can offer better performance than standard SQL, especially when the developer is not that familiar with SQL.

Disadvantage of ORM is

- ➢ that very abstraction makes it so easy to use. There comes a point in many development lifecycles when the developer needs to know exactly what is happening under-the-hood. The abstraction ORM provides, however, insulates the developer from that layer, potentially making low-level troubleshooting more difficult.
- ➢ ORMs are slower as they act as a middleware between your code and the query execution. In fact, ORMs internally create the same raw query to get the desired result.
- ➢ ORM may restrict the scope of your implementation. As I mentioned, they act as a middleware. There is a possibility that your database engine supports some functionality but that was not implemented in the ORM. But in such scenarios you always have the option to write a raw SQL query to get the desired result .

Advantages and Disadvantages of SQL

- One of its biggest benefits is the ability to make out the best possible performance. Although ORM can offer better out-of-the-box performance, developers proficient in SQL can use it to the fullest and achieve a higher level of performance.
- Similarly, SQL allows a developer to see exactly what is happening in the underlying database, making troubleshooting complex problems easier than when using ORM.

The disadvantage of using SQL

- is that it requires the developer to learn the language and become proficient enough to unlock its full potential.

ADVANTAGE OF USING ORM OVER RAW SQL:

Robustness:

You don't need to worry about the syntax errors you might make in writing the SQL query for different Database sources. In fact you do not need to know the syntax of all the DB sources. Same ORM query works for all. Whether it is SQL based engine like MySQL, or NoSQL based engine like MongoDB.

Scalability:

With change in business requirement, or kind/amount of data you are handling. It is very common to change the database engine. You don't have to worry about the breakage in the query, as ORM handles that. The only condition is your ORM should support that data source.

Security:

You need not to worry about the security breaches due to SQL Injections etc as the ORM already acts a protective shield against them