Hibernate ORM Framework in Java

**Introduction**

Java is an object-oriented programming language that includes pre-written code that developers may use to create Java applications. These codes are essentially a set of pre-defined Java classes and functions that developers can use to handle data and govern data flow between software and hardware. The basic goal of frameworks is for developers to be able to reuse them instead of having to recreate everything from scratch each time. Hibernate, which we'll look at in the next few paragraphs, Spring, Blade, Google Web Toolkit, Java Server Faces, and others are examples of prominent Java frameworks.

**Hibernating**

Hibernate ORM (Object Relational Model) is a mediator between Java based objects and relational database which shapes the entities into object-oriented domain. Hibernation enables malleable access to relational databases with high level object-based functions.

**What is Hibernate in Java?**

Hibernate is a Java framework that includes an abstraction layer and manages implementations internally. Writing a query for CRUD activities or establishing a connection with the databases are examples of implementation activities.

What is an ORM Tool?

It is a mechanism for mapping objects recorded in a database. An ORM tool makes data generation, modification, and access easier. To communicate with the databases, it internally employs the Java API.

**Java ORM Frameworks**

There are several persistent frameworks and ORM options in Java. A persistent framework is an ORM service that stores and retrieves objects into a relational database. Enterprise JavaBeans, Entity Beans, Java Data Objects, Castor TopLink Spring, DAO Hibernate etc.

**Benefits of frameworks in software development**

Frameworks are significant because they increase the overall efficiency of developing software and applications by providing developers with pre-built functionalities, allowing them to avoid the time-consuming process of writing programs from the ground up. Without needing a lot of code syntax, frameworks dramatically increase the speed of a software program and its functional output. Frameworks can help developers achieve more efficiency, productivity, maintainability, and dependability.

With the support of multiple surface levels for user preferences, today's frameworks provide a large space for program and software customization. Frameworks may be used to increase the security of apps, and the framework's website may simply provide solutions for any security flaws discovered. The cost of maintaining a framework is often cheap since frameworks improve developer performance, resulting in less time spent building applications, lowering costs in terms of both time and effort.

**Hibernates advantages as a Java framework**

The Hibernate ORM framework for Java is a well-known object-relational mapping utility. The major characteristics of this framework are that it is lightweight since it only employs functions that are necessary for object-relational mapping. Hibernate is also a scalable framework, which means it can be used in applications of all sizes, both large and little. It also allows caching, which dramatically increases application speed.

Hibernate ORM features its own database-independent query language called Hibernate Query Language, which gives developers the advantage of not having to learn SQL to connect their front-end application to the backend. Hibernate is open-source software, which means that anybody who wants to use it can do so for free. Hibernate is also database agnostic, which provides it versatility because it can handle a wide range of databases, including MySQL, Oracle, and others.

**Benefits** of using Hibernate ORM

In comparison to other frameworks, Hibernate ORM has a number of benefits.

* The first benefit is that the hibernate ORM framework offers caching, which increases the efficiency of the application.
* It quickly corrects data mismatches between a relational database and an application's object-oriented class.
* Because the Hibernate ORM framework makes use of JPA annotations, the application code may be ported to other ORM frameworks.
* Hibernate ORM has a locking technique that allows several transactions to access the same object at the same time without degrading the application's performance.
* The Hibernate framework also allows inheritance, which means that when a user saves a child class object, the parent class is also saved in the database.
* -Hibernate ORM also offers a capability that produces primary keys automatically when the user enters data into the database.
* Another benefit of the Hibernate framework is that it builds database tables automatically, eliminating the need to construct them manually in the database.
* Hibernate ORM is a free open-source framework that may be used by anybody.
* The Hibernate ORM framework employs a technique known as lazy-loading, in which only the most necessary objects necessary for application execution are fetched.
* Hibernate can handle a variety of connections, including one-to-one, one-to-many, many-to-many, and many-to-one.
* Hibernate also has an exception converter, which translates checked JDBC exceptions to unchecked hibernate exceptions. Exceptions in hibernate are unchecked exceptions, which means the developer does not need to explicitly manage them.
* Hibernate also includes time stamps, which allow developers to keep track of how many times data has been updated.

**Negative Effects Of Using Hibernate ORM**

Although the Hibernate ORM framework offers many advantages, it does have a few drawbacks

* The Hibernate ORM has the issue of being too heavy for small projects or applications with fewer tables, since it would have a significant impact on speed by utilizing too much memory.
* Hibernate ORM does not support multiple object inserts into the database using a single query. To insert each item, the developer must create a new query.
* Another downside of hibernate ORM is that it performs poorly in batch processing.
* As the number of mappings and joins in tables grow, the code for hibernate ORM gets more difficult to comprehend since the mapping and joining information must be written in XML.
* Another downside is that HQL's runtime creation of numerous SQL statements limits its overall performance.
* Hibernate ORM, like many frameworks, requires a substantial amount of effort to master because it exposes a large number of APIs.
* The Hibernate ORM framework can be challenging to troubleshoot and optimize in terms of performance.
* The hibernate framework's debugging and performance optimization might be tough at times.

**Reference List**

(G, 2016)

(Mihalcea, 2006)

(J, 2016)

(Saddar, 2018)

Bakang Monei Motshegwe

Research Assignment