**Hiebarnet\_Interview\_Questions**

**Q1) What is the way to use second level cache in hibernate**

**A)** A second-level cache is Session Factory-scoped, meaning it's shared by all sessions created with the same session factory. When an entity instance is looked up by its id (either by application logic or by Hibernate internally, e.g. when it loads associations to that entity from other entities), and second-level caching is enabled for that entity, the following happens:

* If an instance is already present in the first-level cache, it's returned from there.
* If an instance isn't found in the first-level cache, and the corresponding instance state is cached in the second-level cache, then the data is fetched from there and an instance is assembled and returned.
* Otherwise, the necessary data are loaded from the database and an instance is assembled and returned.

Different vendors have provided the implementation of second level cache

* EH Cache
* OS Cache
* Swarm Cache
* Jboss Cache

Each implementation provides different cache usages functionality

1. Read-only
2. Nonstrict read-write
3. Read-write
4. Transactional

**Q2) Advantages of hibernates**

**A)**

**ORM:** Hibernate RM easily solves the data mismatch found between the object-oriented classes of an application and relational database.

**Transparent Persistence:** Hibernates transparent persistence ensures the automatic connection between the application objects with the database tables.

**Database Independent:** Hibernate is database independent. It can be used to connect with any database like MySQL, Oracle and DB2.

**HQL:** Hibernate supports a powerful query language called HQL (hibernate query language). This query language is more powerful than SQL and completely object oriented.

**Dual-Layer Caching:** Hibernate supports both first level and second level caching mechanisms.

The first level caching is associated with session object which is used by default. The second level caching is related to the session factory object.

**Open Source:** Hibernate is available as an open source software with zero cost product license.

**Scalability:** Hibernate is highly scalable it adapts itself in any environment.

**Lazy-Loading:** The Lazy-Loading concepts fetches only the necessary object that is required for the execution of an application.

**Q3) What is lazy loading and eager loading**

**A)**

|  |  |
| --- | --- |
| **Lazy-Loading** | **Eager-Loading** |
| In lazy loading associated with data loads only when we explicitly call getter or size method | In eager data loading happens at the time of their parents fetched |
| Many to many and one to many associations used lazy loading strategy by default | Many to one and one to one associations used in lazy loading strategy by default |
| It can be enabled by using the annotation parameter fetch=FETCHTYPE.LAZY | It can be enabled by using the annotation parameter fetch=FETCHTYPE.EAGER |

**Q4) What is the difference between JPA and Hiebarnet**

**A)**

|  |  |
| --- | --- |
| **JPA** | **Hibernate** |
| JPA : It is a Java **specification**for mapping relational data in Java application. It is not a framework | Hibernate is an **ORM framework** and in that way data persistence is possible. |
| In JPA, no implementation classes are provided. | In Hibernate, implementation classes are provided. |
| Main advantage is It uses JPQL (Java Persistence Query Language) and it is platform-independent query language. | Here it is using HQL (Hibernate Query Language). |
| It is available under javax.persistence package. | It is available under org.hibernate package. |
| In Hibernate, EclipseLink, etc. we can see its implementation. | Hibernate is the provider of JPA. |
| Persistence of data is handled by EntityManager. | Persistence of data is handled by Session. |

**Q5) What is Session Factory Object and Session Object**

**A) Session Factory Object:** Session factory is thread safe object and used by all the threads of an application

Session factory is heavy weight object. It is created during start-up and kept for later use. We need one session factory object per database using a separate configuration file.

**Session Object:** A session object is used to get a physical connection with a database. The session object is light weight and designed to be instantiated each time an interaction is needed with the database. Persistence object are saved and retrieved through a session object.

**Q6) What is caching, explain about first level caching and second level caching**

**A)** Caching is a mechanism to enhance the performance of a system. It is a buffer memory that lies between the application and the database.

Cache memory stores recently used data items in order to reduce the number of database hits as much as possible.

**First-Level Cache:** The first level cache is the session cache and it is a mandatory cache through which all requests must pass. The session object keeps an object under it’s own power before committing it to the database.

If we close the session, all the objects being cached are lost and either persisted or updated in the database.

**Second-Level Cache:** Second level cache is an optional cache and first level cache will always be consulted before any attempt us made to locate an object in second level cache.

Hibernate uses first level cache by default and we have to do nothing to use first level cache.

The hibernate second level cache is set up in two steps. First, we have decided which concurrency strategy to use. After that we have to configure cache expiration and physical cache attributes using the cache provider.

1. Read-only – Caching will work for read only
2. Nonstrict read-write – Caching will work for read and write but one at a time
3. Read-write – Caching will work for read and write. Can be use both simultanesly
4. Transactional – caching will work for concurrency transactions.

**Q7) What is concurrency strategies**

**A)** A concurrency strategy is mediator, which is responsible for storing items of data in the cache and retrieving them from the cache. If we are going to enable second level cache, we will have to decide for each persistent class and collection.