Hibernate Tuning

Presented by

Sagar
Java Consultant

Dirty Checking

In Hibernate, the **“dirty checking”** mechanism tracks changes in entities that are in the “persistent” state (i.e., associated with a session).

Dirty Checking is after the execution of

session.save()	C
session.get()	R
session.update()	U
session.delete()	D

DTO – Data Transfer Object :
An object that carries data between processes in order to reduce the number of method calls.

If data only need to display and without further modify it, then a DTO projection is much more suitable.

Fetching Techniques

Tuning fetch strategies :

Select fetching (the default) is extremely vulnerable to N+1 selects problems, so we might want to enable join fetching in the mapping document

1-1, 1-M, M-1

The fetch strategy defined in the mapping document affects:

1. Retrieval via get() or load()
2. Retrieval that happens implicitly when an association is navigated
3. Criteria queries
4, HQL queries if sub select fetching is used

Cacheing

- **Cacheing** in Hibernate refers to the technique of storing frequently accessed data in
- memory to improve the performance of an application that uses Hibernate as an
- Object-Relational Mapping (ORM) framework.

- Hibernate provides two levels of caching:

  - **First-Level Cache**: Hibernate uses a session-level cache, also known as a first-level cache, to store the data that is currently being used by a specific session.

  - When an entity is loaded or updated for the first time in a session, it is stored in the session-level cache.

  - **Second-Level Cache**: Hibernate also supports a second-level cache, which is a shared cache across multiple sessions.

  - This cache stores data that is frequently used across different sessions, reducing the number of database queries and improving the overall performance of the application.

Thank You