Beaconfire Inc, Home Work, Week4 Day18.
By (Ping) Nalongsone Danddank.
ndanddank@gmail.com
wechatID: ndanddank
Short Answer:

1. What is ORM and why is it helpful?

-> is technique that let us processing on data base by mapping out object structure or Object-oriented paradigm to relational database table column. We just interact directly with an object. So we do not use SQL anymore.

2. What is Hibernate and what is its advantage?

-> is a tool of ORM, it is an open source and light weight and use for developing DB independent persistence logic in java based application. The advantage is let developers a way to map the object structures in java classes to relational database table. And provide APIs to store and retrieving objects directly to database.

3. What is Connection Pool and what is its advantage?

-> is the reuse of connection objects and reduce the number of times that created connection objects. Improve performance for database-intensive application because creating connection object is expensive processing.

4. What are some of the configurations for Connection Pool?

-> Max pool size, Max idle size, min idle size, idle waiting time.

5. What is a Session and how do we get a Session?

->The session object provides an interface between the application and data stored in the database.
• A Session is a light weight and a non-threadsafe object that represents a single unit-of-work with the database. It is a short-lived object and wraps the JDBC connection. It is factory of Transaction, Query and Criteria.

6. What is a SessionFactory and how do we get a SessionFactory?

-> SessionFactory is Hibernate's concept of a single datastore and is thread-safe so that many threads can access it concurrently and request for sessions and immutable cache of compiled mappings for a single database.

7. If an application uses Hibernate for persistent logic and uses MySQL as persistent store, what are the steps that we need to do if the application want to switch to OracleDB from MySQL? (Assume table structures remain the same)

-> just change parameter on the hibernate.cfg.xml for your project or change some sessionFactory bean in your Spring ApplicationContext.xml .

8. What are different entity states defined in Hibernate? Explain each state.

-> •there are the three main states in relation to the Session persistence context:
• transient — this instance is not, and never was, attached to a Session; this instance has no corresponding rows in the database; it's usually just a new object that you have created to save to the database;
• persistent — this instance is associated with a unique Session object; upon flushing the Session to the

database, this entity is guaranteed to have a corresponding consistent record in the database;
• detached — this instance was once attached to a Session (in a persistent state), but now it's not; an instance enters this state if you evict it from the context, clear or close the Session, or put the instance through serialization/deserialization process.

9. What are the methods can be used to transfer an entity between different states?
-> Transient -> Persistent are: save(), persist(), saveOrUpdare()
Persistent -> Transient: delete().
Persistent -> Detached: evict(), session.close()
Detached -> Persistent: lock(), update(), merge(), saveOrUpdate().
In Persistent itself are: get(), load(), HQL quey.

10. What is the difference between get() and load()?
-> get(): Used to fetch data from the database for a given identifier • Return null if no object can be found using given Identifier • Eager loading - Return a fully initialized object • Slower performance
-> load(): Also used to fetch data from the database for a given identifier • Throw exception if not object can be found using the given identifier • Lazy Loading - Return proxy object • Slightly faster performance
LazyInitializationException (very importance) load() may throw this exception
11. What is the difference between save() and persist()?
-> save(): The method strictly states that it persists the instance, "first assigning a generated identifier".
• The method is guaranteed to return the Serializable value of this identifier.
• The method has a return type of Serializable
• The reference of the passed in object pointing to the persisted object.
• Note: it does not conform to the JPA specification.
-> persist(): The persist method is intended for adding a new entity instance to the persistence context,
• i.e. transitioning an instance from transient to persistent state.
• We usually call it when we want to add a record to the database (persist an entity instance)
• The persist method has void return type. It operates on the passed object "in place", changing its state.
• The object passed in now actually pointing to the persisted object
• Note: This method does NOT guarantee that the id of the object will be generated after calling the method. It follows JPA specification.

12. What is the difference between update(), merge() and saveOrUpdate()?

-> update(): It acts almost same as Save and Persist method, with small different:
• It acts upon passed object (its return type is void)
• The update method transitions the passed object from detached to persistent state
• This method throws an exception if you pass it a transient entity
• Note: it does not conform to the JPA specification.

-> merge(): • The main intention of the merge method is to update a persistent entity instance with new field values from a detached entity instance
• Suppose we have a RESTful interface with a method for retrieving an JSON-serialized object by its id to the caller and a method that receives an updated version of this object from the caller.

• An entity that passed through such serialization/deserialization will appear in a detached state. So the merge method does exactly that:
• Finds an entity instance by id taken from the passed object
• Copies fields from the passed object to this instance
• Returns newly updated instance
• The return type of the method is an Object — It is the object loaded into the persistent state and updated, not the object passed as the argument.
• Note: It follows JPA specification.

-> saveOrUpdate(): Similar to update, it also may be used for reattaching instances
• The main difference of saveOrUpdate method is that it does not throw exception when applied to a transient instance; instead, it makes this transient instance persistent.
• Note: it does not conform to the JPA specification.