Working with Objects

Agenda

1 **Hibernate Object States**

2 **Insert Object**

3 **Retrieve Object**

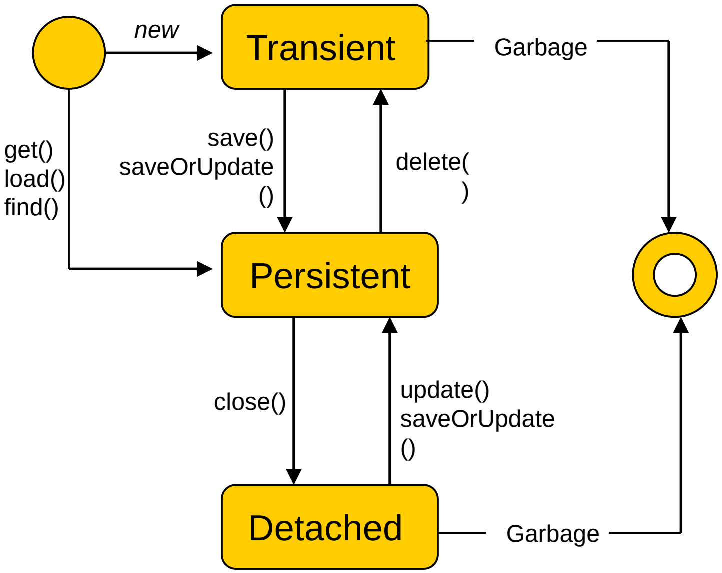
4 **CURD Operations**

Hibernate Object States

- An object is **transient** when instantiated with new operator
- Not associated with Hibernate session
- No persistent representation in database & no identifier value yet assigned

- A **persistent** object has a database identity & an identifier value assigned
- Associated with Hibernate session – saved or loaded
- Participates in a transaction

- A **detached** instance is an object that has been persistent, but its *Session has been closed*
- The state of detached objects is not guaranteed to be synchronized with database state



States of an object and transitions in a Hibernate Application

Insert Object

- Hibernate considers newly instantiated instances of a persistent class as *transient*
- A transient instance can be made *persistent* by associating it with a session and calling save() method
- Hibernate provides save() method to persist object to the database

```
public Object save(Object obj);
```

- Save method has to be called from a session inside a transaction
- Example:

```
// Create a Department object and save it
Transaction transaction = session.beginTransaction();
Department deptobj = new Department(50, "TT", "NJ");
session.save(deptobj);
transaction.commit();
```

Make a transient object persistent

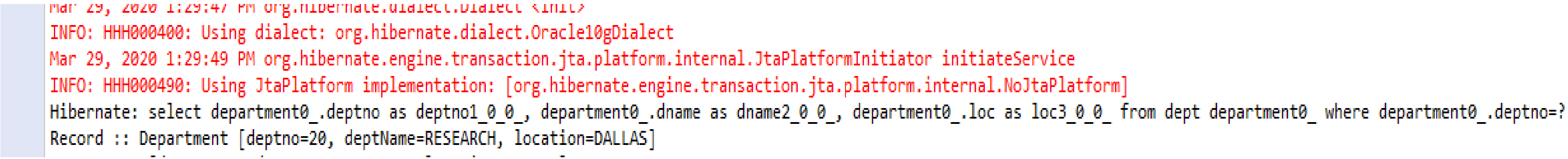
Retrieve object

- You can query the database using Session to retrieve existing persistent objects
- Two special methods are provided for the simplest kind of query: retrieval by identifier
  - The get() method
  - The load() method
- These methods return a single Object mapping to the record retrieved based on the given identity value

Retrieve object – The Get Method

- The **get** method of the session object retrieves a persistent instance of the given Persistent Class and the identifier value

```
Department deptobj = new Department();
deptobj=session.get(Department.class, 20);
System.out.println("Record :: "+deptobj);
```



```
Mar 29, 2020 1:29:47 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.Oracle10gDialect
Mar 29, 2020 1:29:49 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select department0_.deptno as deptno1_0_0_, department0_.dname as dname2_0_0_, department0_.loc as loc3_0_0_ from dept department0_ where department0_.deptno=?
Record :: Department [deptno=20, deptName=RESEARCH, location=DALLAS]
```

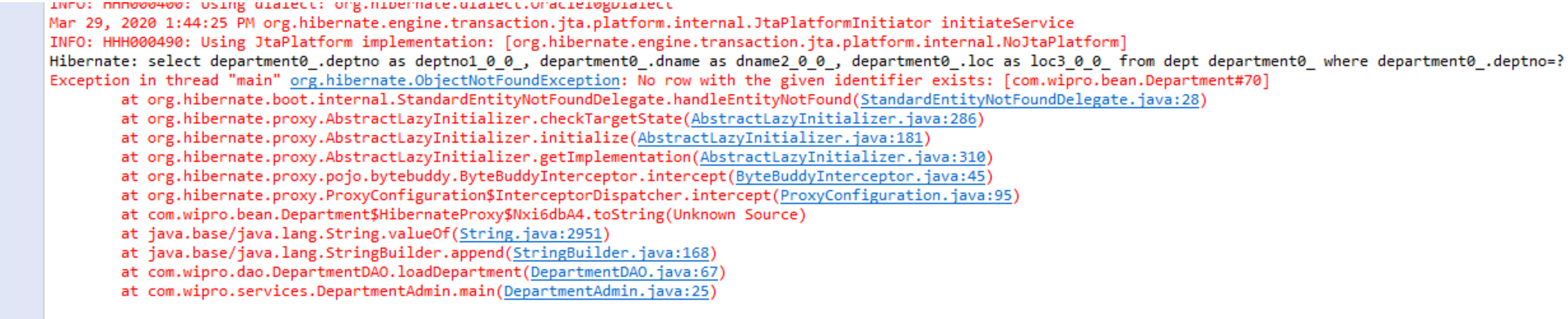
- If the given ID is not available it returns null

Retrieve object – The Load Method

- The **load** method of the session object retrieves a persistent instance of the given Persistent Class and the identifier value

```
Department deptobj = new Department();
deptobj=session.load(Department.class, 88);
System.out.println("Record :: "+deptobj);
```

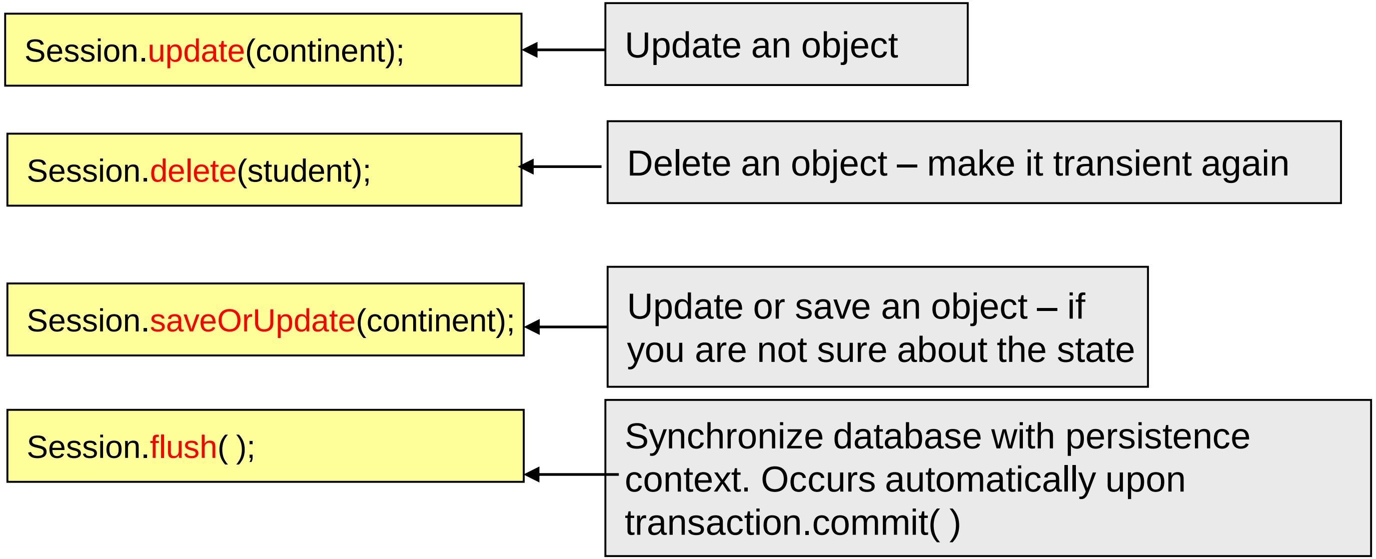
- If the given ID is not available load method throws ObjectNotFoundException



```
INFO: HHH000400: Using dialect: org.hibernate.dialect.Oracle10gDialect
Mar 29, 2020 1:44:25 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select department0_.deptno as deptno1_0_0_, department0_.dname as dname2_0_0_, department0_.loc as loc3_0_0_ from dept department0_ where department0_.deptno=?
Exception in thread "main" org.hibernate.ObjectNotFoundException: No row with the given identifier exists: [com.wipro.bean.Department#70]
	at org.hibernate.boot.internal.StandardEntityNotFoundDelegate.handleEntityNotFound(StandardEntityNotFoundDelegate.java:28)
	at org.hibernate.proxy.AbstractLazyInitializer.checkTargetState(AbstractLazyInitializer.java:286)
	at org.hibernate.proxy.AbstractLazyInitializer.initialize(AbstractLazyInitializer.java:181)
	at org.hibernate.proxy.AbstractLazyInitializer.getImplementation(AbstractLazyInitializer.java:310)
	at org.hibernate.proxy.pojo.bytebuddy.ByteBuddyInterceptor.intercept(ByteBuddyInterceptor.java:45)
	at org.hibernate.proxy.ProxyConfiguration$InterceptorDispatcher.intercept(ProxyConfiguration.java:95)
	at com.wipro.bean.Department$HibernateProxy$Nxi6dbA4.toString(Unknown Source)
	at java.base/java.lang.String.valueOf(String.java:2951)
	at java.base/java.lang.StringBuilder.append(StringBuilder.java:168)
	at com.wipro.dao.DepartmentDAO.loadDepartment(DepartmentDAO.java:67)
	at com.wipro.services.DepartmentAdmin.main(DepartmentAdmin.java:25)
```

CRUD operations

- Session object supports with update and delete methods to modify the database table contents



Session.update(continent);

Update an object

Session.delete(student);

Delete an object – make it transient again

Session.saveOrUpdate(continent);

Update or save an object – if you are not sure about the state

Session.flush( );

Synchronize database with persistence context. Occurs automatically upon transaction.commit( )

For Sample Application Refer: Hibernate - CRUD Operations.pdf

Summary

In this module you were able to :

- Understand Hibernate Object States
- Make Objects persistent
- Retrieve a persistent Object
- Modify Objects
- Delete Objects

Thank You