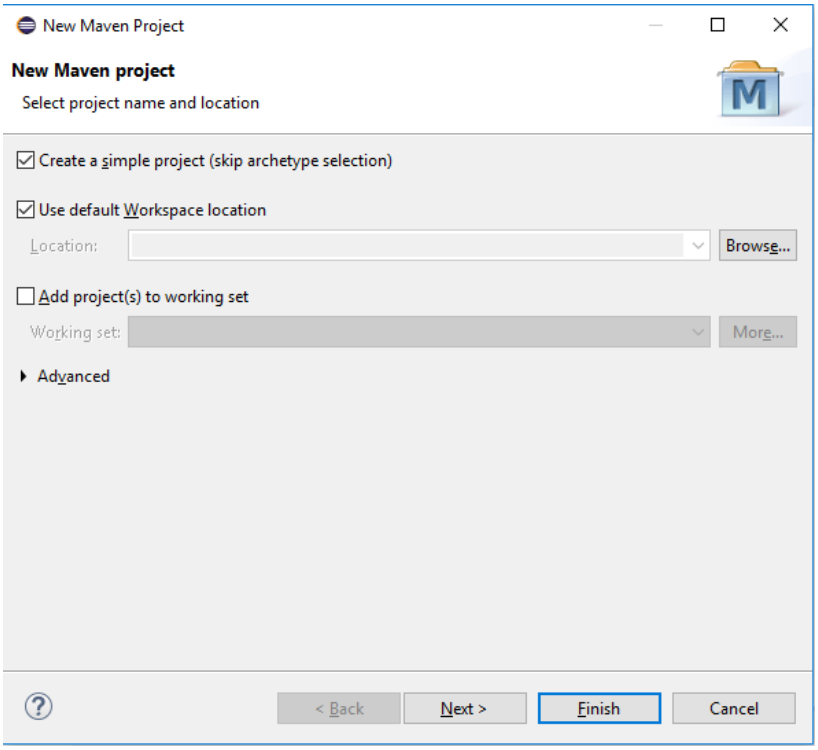
First Hibernate Demo Program

Step by Step creation of Hibernate application using maven

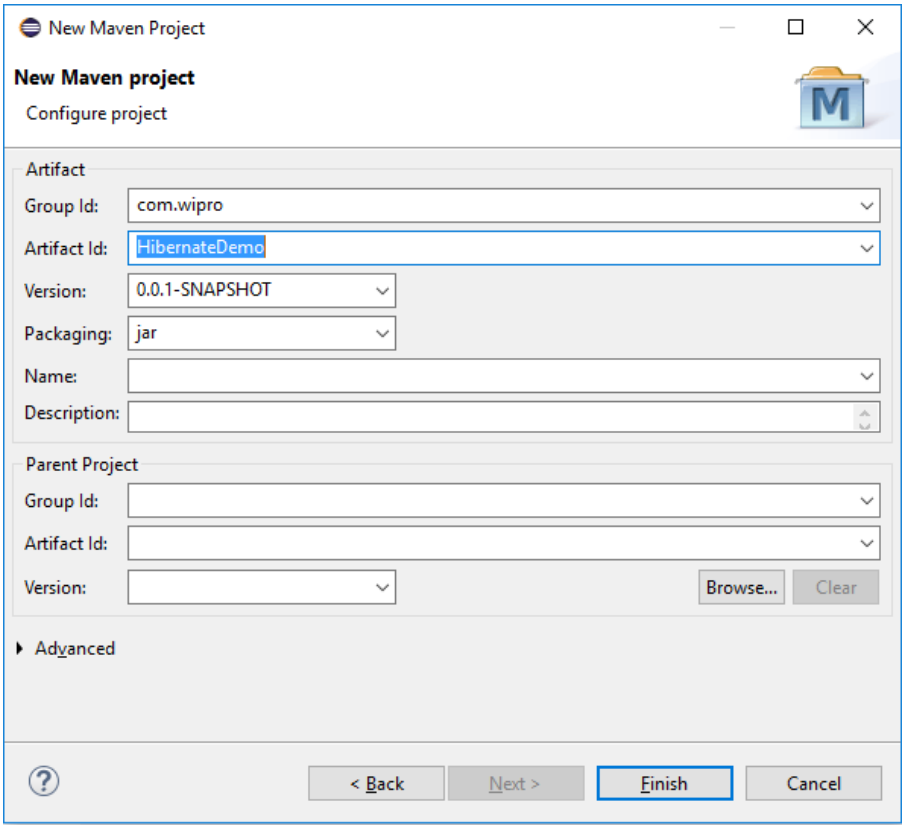
1. To start with let us take any table from any database.
    a. Here we would be using Oracle 11g
    b. Login into SCOTT schema
    c. Verify dept table .

```
Command Prompt - sqlplus

SQL> desc dept
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 DEPTNO                                             NUMBER(3)
 DNAME                                              VARCHAR2(14)
 LOC                                                VARCHAR2(13)

SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

SQL>
```

2. Now let us Create a maven Project using hibernate to insert a record into the dept table.
    a. Create a simple project

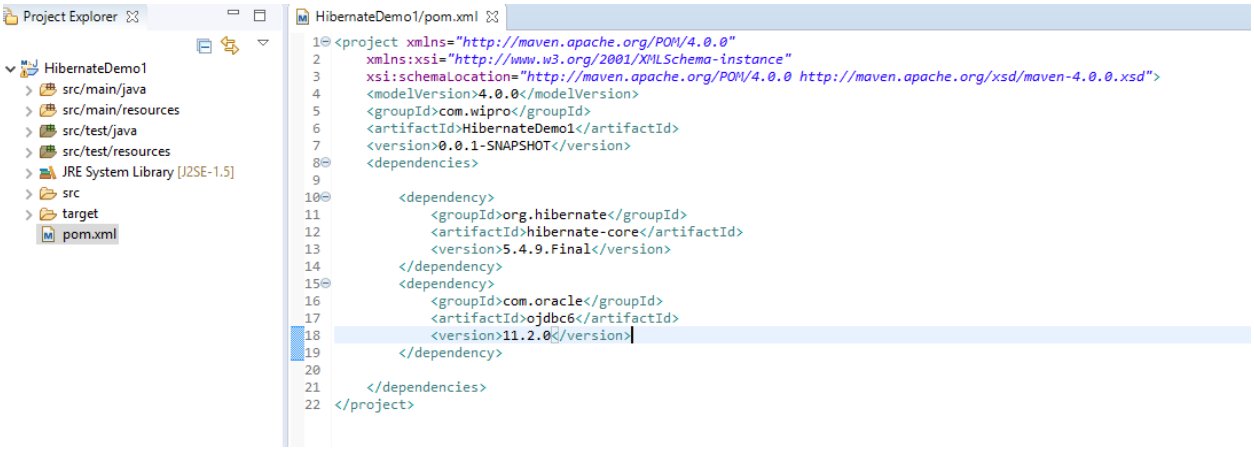


b. Provide the artifacts



3. Add the hibernate and Oracle dpendency in the POM file
   a. To get the required version details check the below link
      https://mvnrepository.com/artifact/org.hibernate/hibernate-core
   b. Add the dependency code and save POM file

```
<dependency>
<groupId>org.hibernate</groupId>
<artifactId>hibernate-core</artifactId>
<version>5.4.9.Final</version>
</dependency>
<dependency>
<groupId>oracle</groupId>
<artifactId>ojdbc6</artifactId>
<version>11.2.0.3.0</version>
</dependency>
```



c. In case oracle depency throws Missing artifact error, load the ojdbc6.jar as external jar file [Just like how we do for JDBC apps]

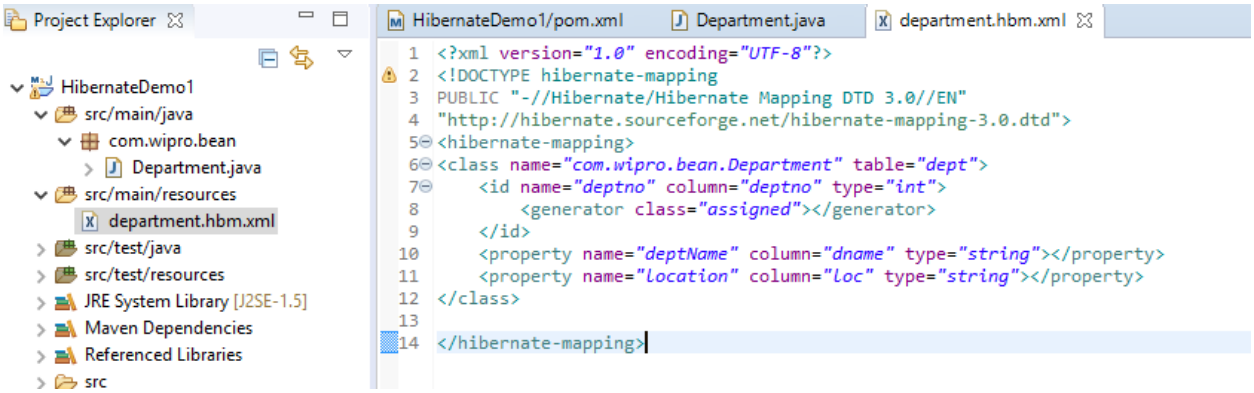
4. Create the Department bean class to represent the dept table structure
   a. Create a package called com.wipro.bean
   b. Under the package create class called Department

```
package com.wipro.bean;

public class Department {
	private int deptno;
	private String deptName;
	private String location;
	public Department() {

	}
	public Department(int deptno, String deptName, String location) {
		super();
		this.deptno = deptno;
		this.deptName = deptName;
		this.location = location;
	}
	public int getDeptno() {
		return deptno;
	}
	public void setDeptno(int deptno) {
		this.deptno = deptno;
	}
	public String getDeptName() {
		return deptName;
	}
	public void setDeptName(String deptName) {
		this.deptName = deptName;
	}
	public String getLocation() {
		return location;
	}
	public void setLocation(String location) {
		this.location = location;
	}
```

```
	@Override
	public String toString() {
		return "Department [deptno=" + deptno + ", deptName=" + deptName
+ ", location=" + location + "]";
	}

}
```

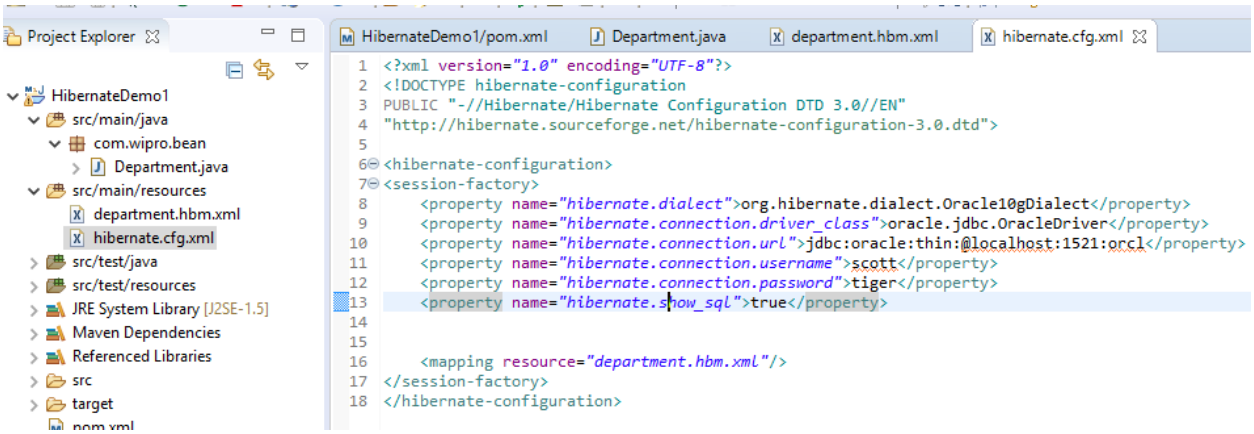
5. Create a xml file name department.hbm.xml in src/main/resources
    a. Hibernate mapping file to map the Department class and variables with dept table and columns in Oracle database.
    b. Right click on src/main/resources new → others → XML → XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping
PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="com.wipro.bean.Department" table="dept">
	<id name="deptno" column="deptno" type="int">
		<generator class="assigned"></generator>
	</id>
	<property name="deptName" column="dname" type="string"></property>
	<property name="location" column="loc" type="string"></property>
</class>

</hibernate-mapping>
```



6. Create Hibernate configuration file in src/main/resources called **hibernate.cfg.xml** here this name is important as it is the default name the config method would search

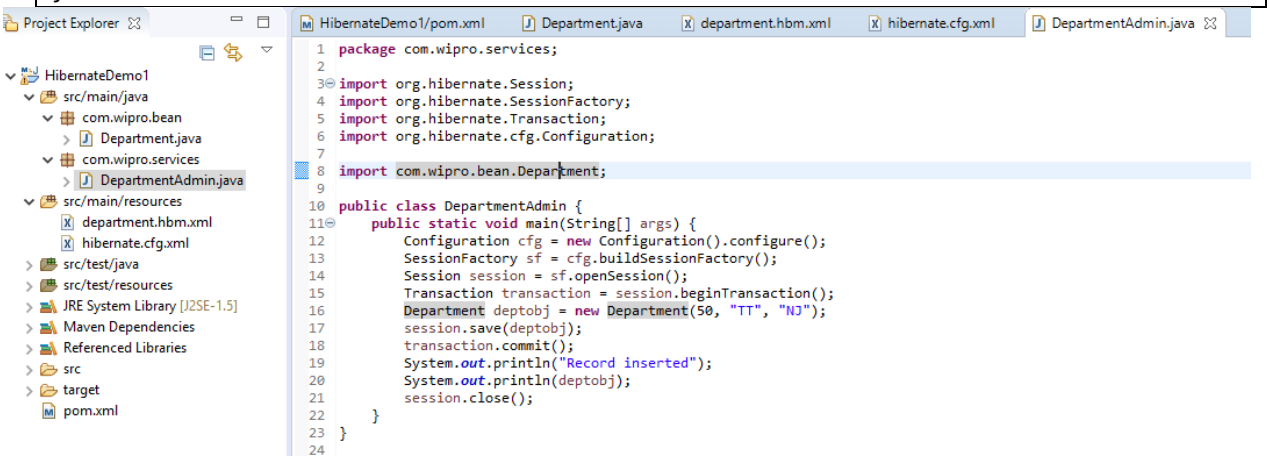
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
<session-factory>
	<property name="hibernate.dialect">
org.hibernate.dialect.Oracle10gDialect</property>
	<property name="hibernate.connection.driver_class">
oracle.jdbc.OracleDriver</property>
	<property name="hibernate.connection.url">
jdbc:oracle:thin:@localhost:1521:orcl</property>
	<property name="hibernate.connection.username">scott</property>
	<property name="hibernate.connection.password">tiger</property>
	<property name="hibernate.show_sql">true</property>


	<mapping resource="department.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```



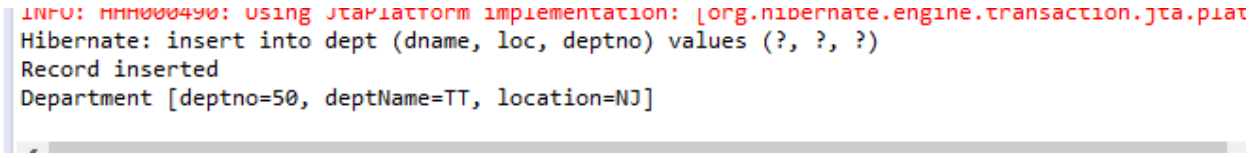
7. Create the main class named DepartmentAdmin under the package com.wipro.service to insert records into the dept table.

```
package com.wipro.services;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

import com.wipro.bean.Department;
```

```java
public class DepartmentAdmin {
    public static void main(String[] args) {
            Configuration cfg = new Configuration().configure();
            SessionFactory sf = cfg.buildSessionFactory();
            Session session = sf.openSession();
            Transaction transaction = session.beginTransaction();
            Department deptobj = new Department(50, "TT", "NJ");
            session.save(deptobj);
            transaction.commit();
            System.out.println("Record inserted");
            System.out.println(deptobj);
            session.close();
    }
}
```



8. Run the application
    a. Eclipse Console output



    b. Database output

```
SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 TT             NJ
```