Spring, Hibernate and MySQL integration Document

Prepared by:- Vijay Chouhan

1. Download the Eclipse,

eclipse-java-juno-SR2-win32-x86\_64.zip

1. Download the JDK 1.7 and set the class path

jdk1.7.0\_79

1. Download the Spring Framework binaries from below url,

<https://repo.spring.io/release/org/springframework/spring>

spring-framework-4.1.6.RELEASE-dist.zip

1. Download Hibernate from <http://www.hibernate.org/downloads>.

hibernate-distribution-3.6.4.Final-dist.zip

1. Download the latest version of Apache Commons Logging API from [https://commons.apache.org/logging/](https://commons.apache.org/logging/download_logging.cgi).

commons-logging-1.1.1-bin.zip

1. Download the Others jar files used for integrate or configure the mySql database in this application as follows below,

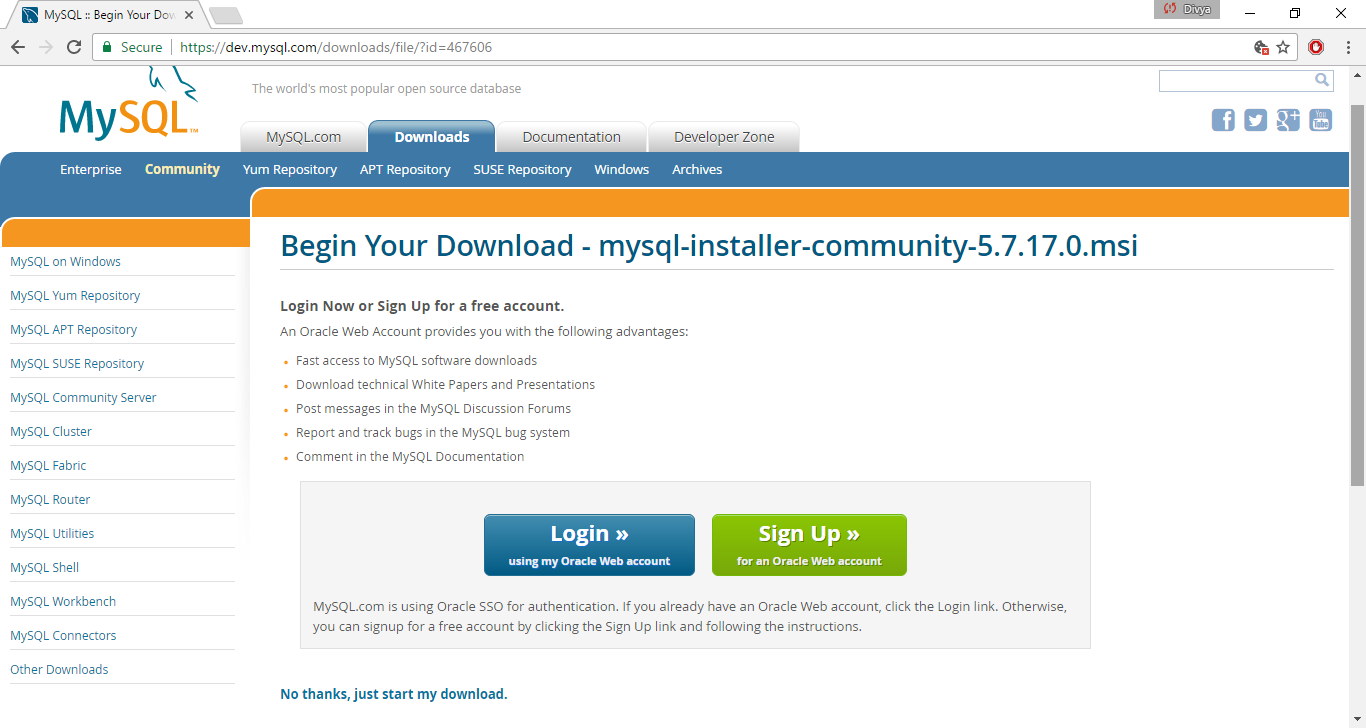
commons-dbcp-1.4-bin.zip

commons-pool-1.6-bin.zip

1. Download the Driver class to connect to the MySQL Database,

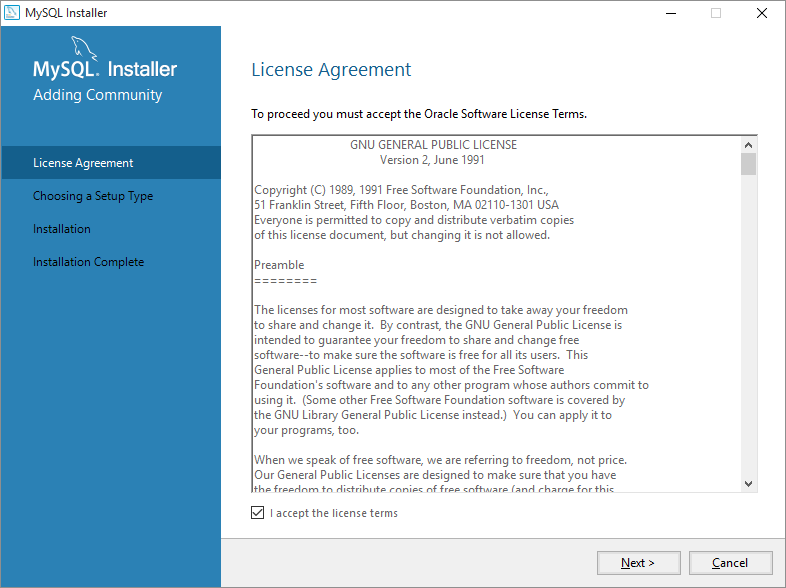
mysql-connector.jar

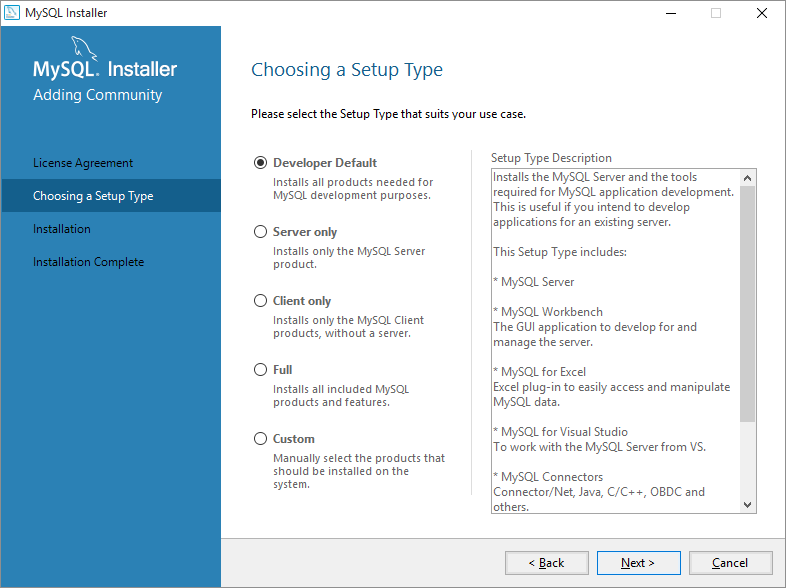
1. Download the MySQL database

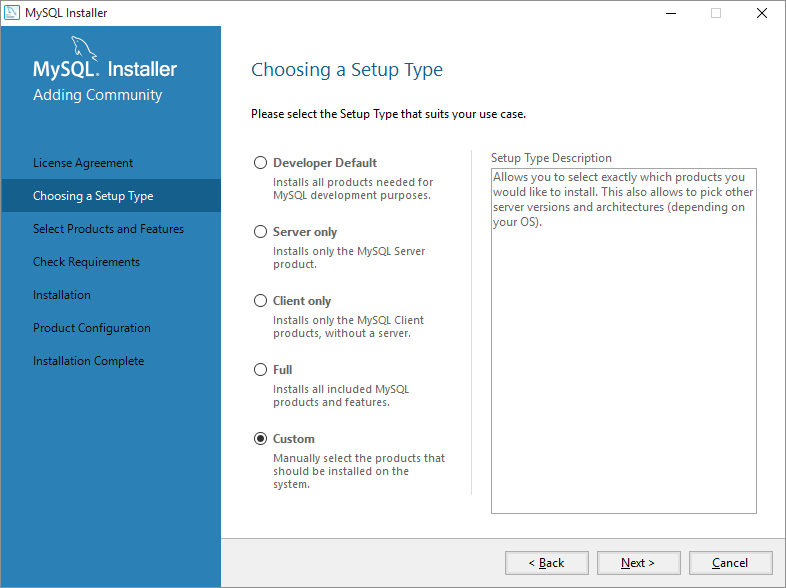


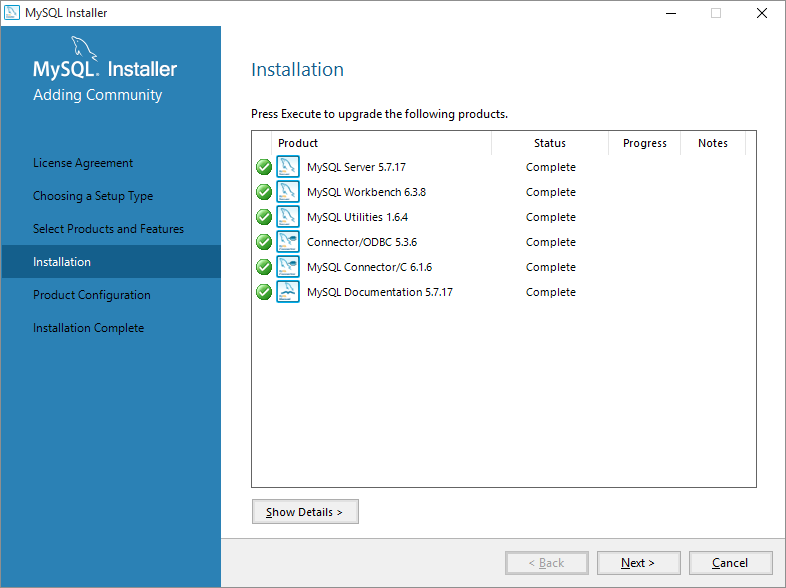
mysql-installer-community-5.7.17.0.msi file gets downloaded,

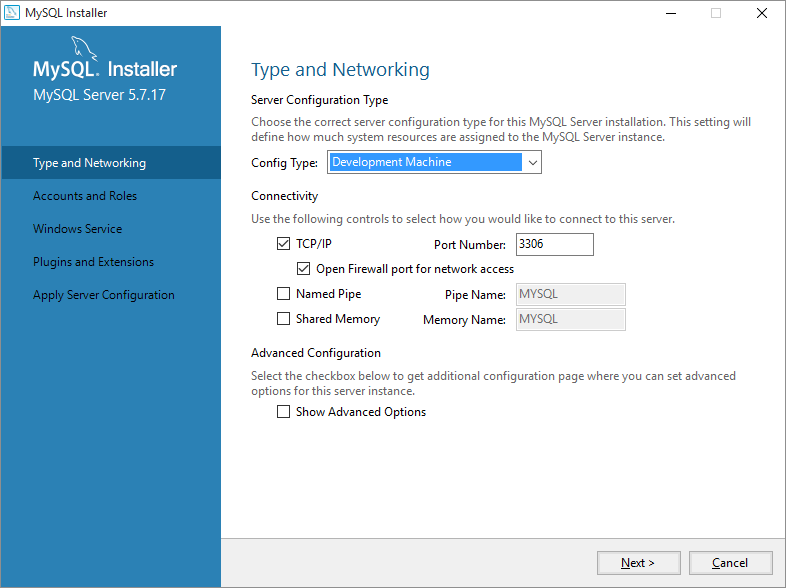
1. Install the MySQL database,

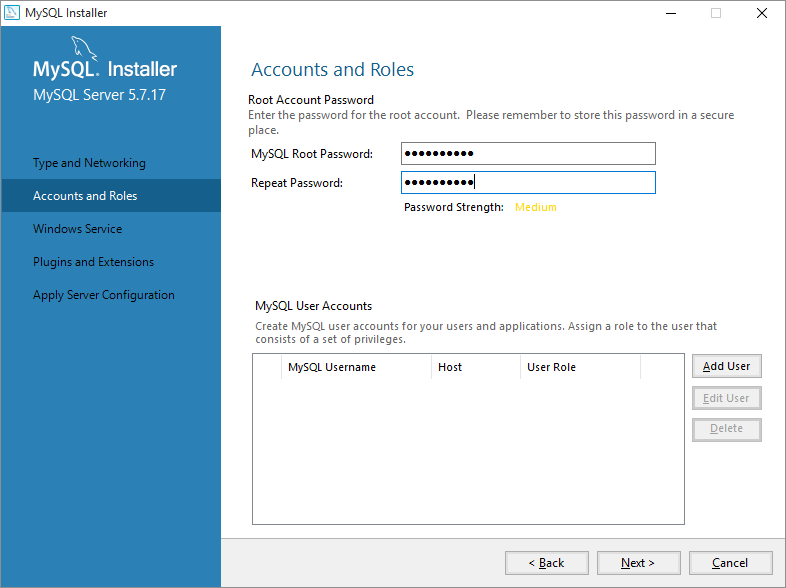




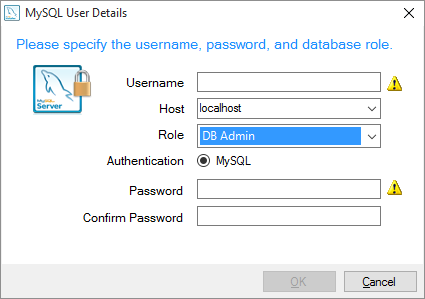








Root password:- Vijay@1234

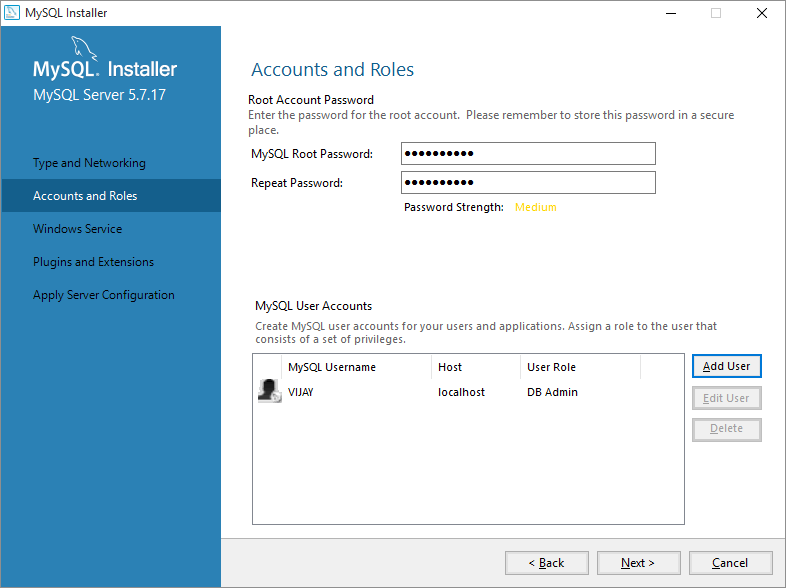


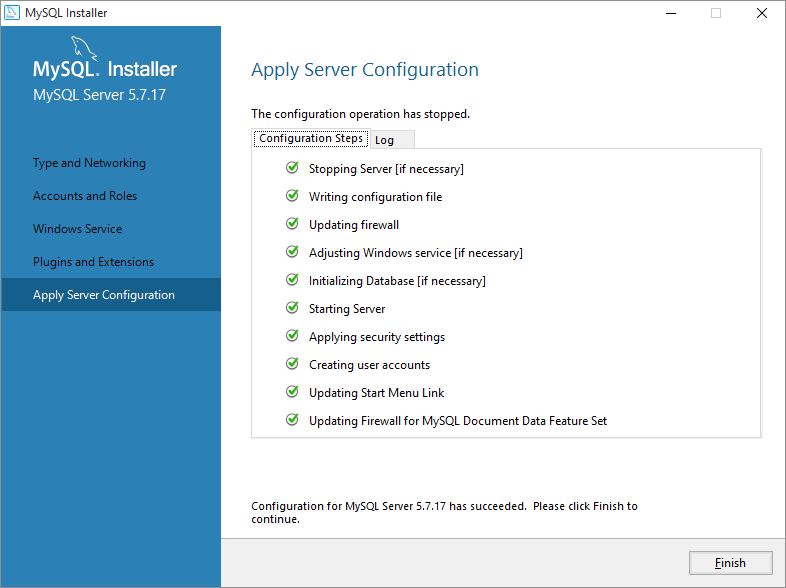
User:- VIJAY

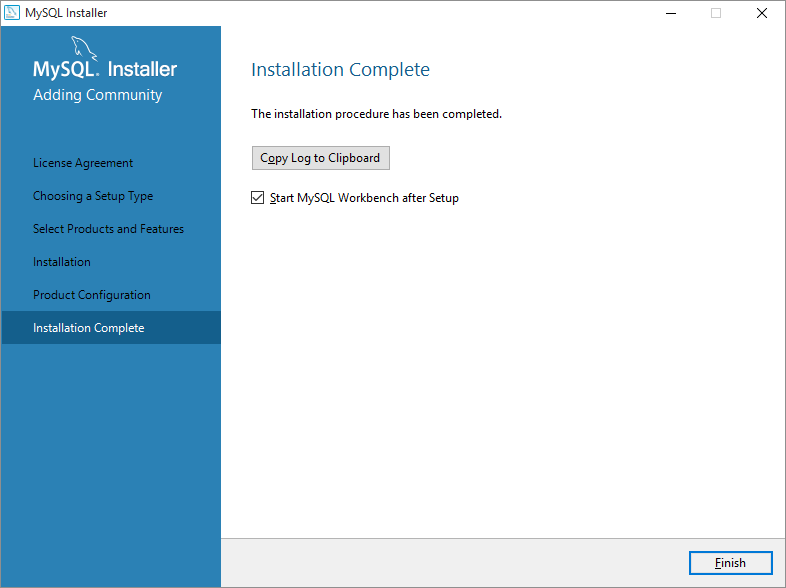
Host:- localhost

Role:- DB Admin

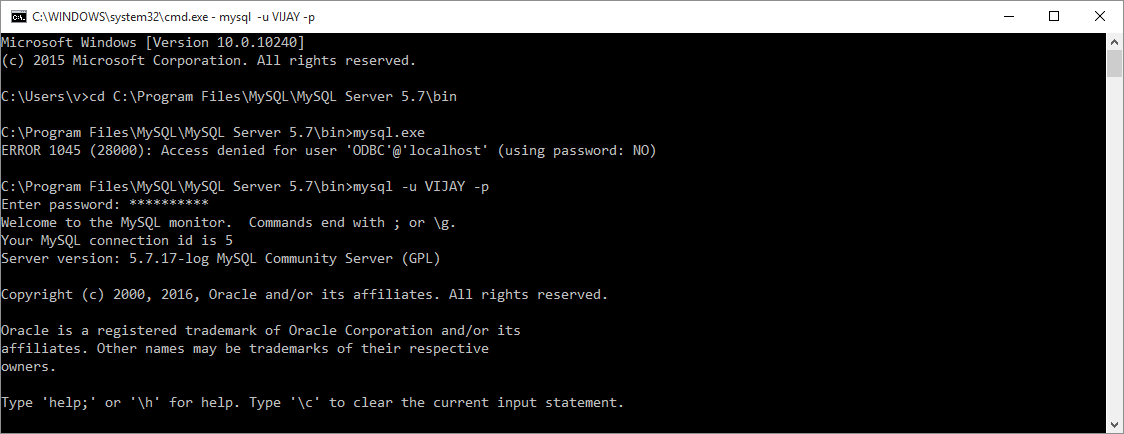
Password:- Vijay@1234

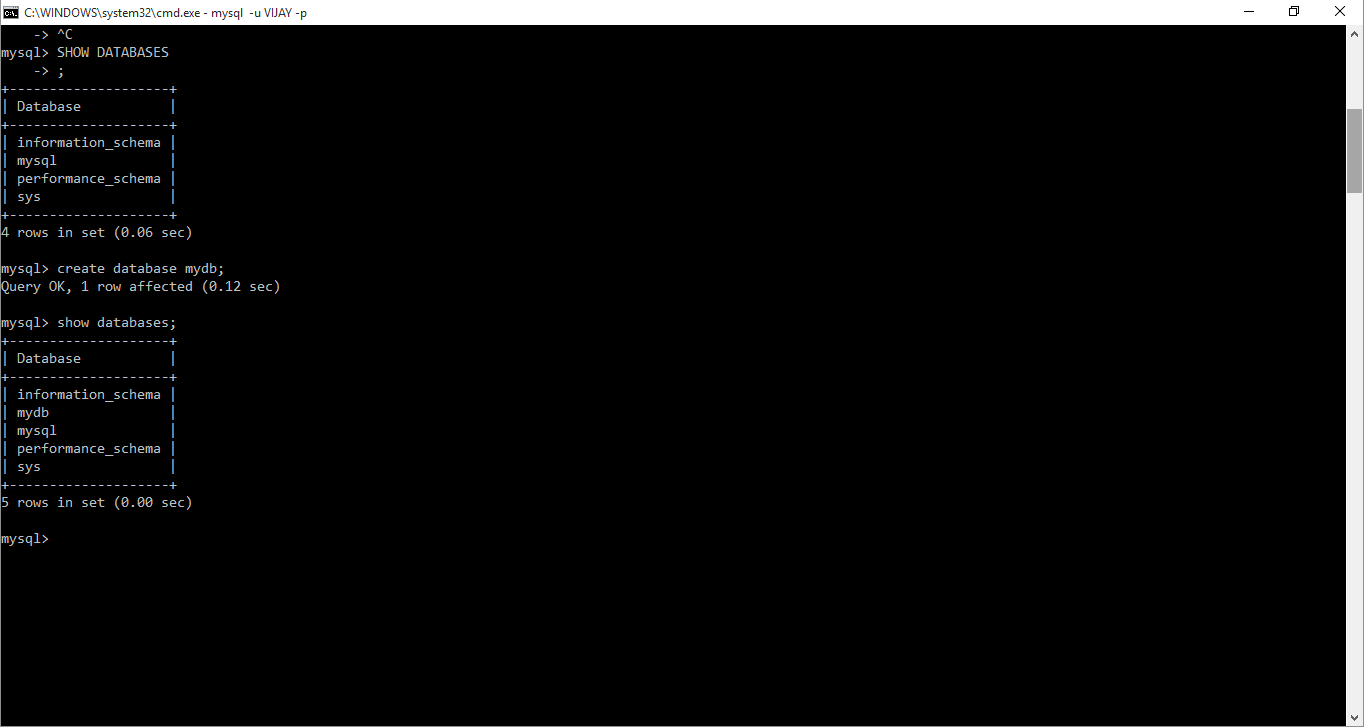




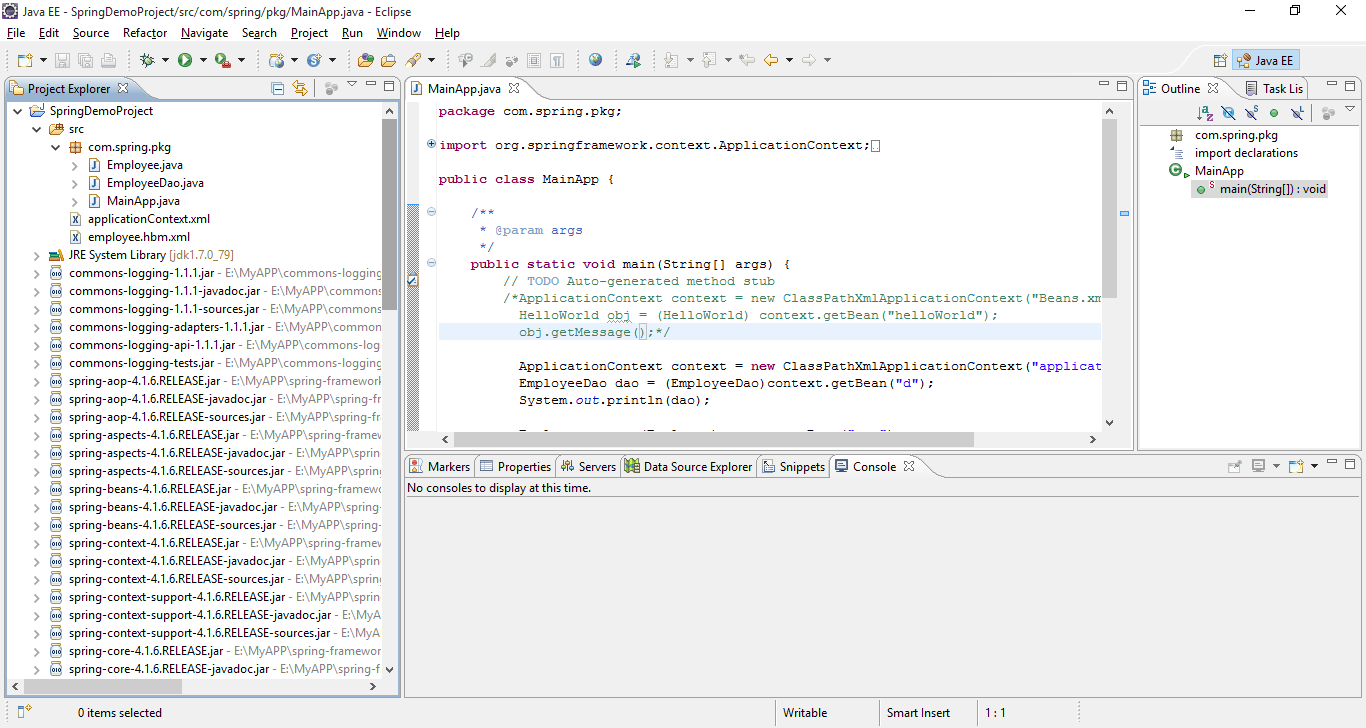


1. Connect to MySQL and create Database as **mydb**

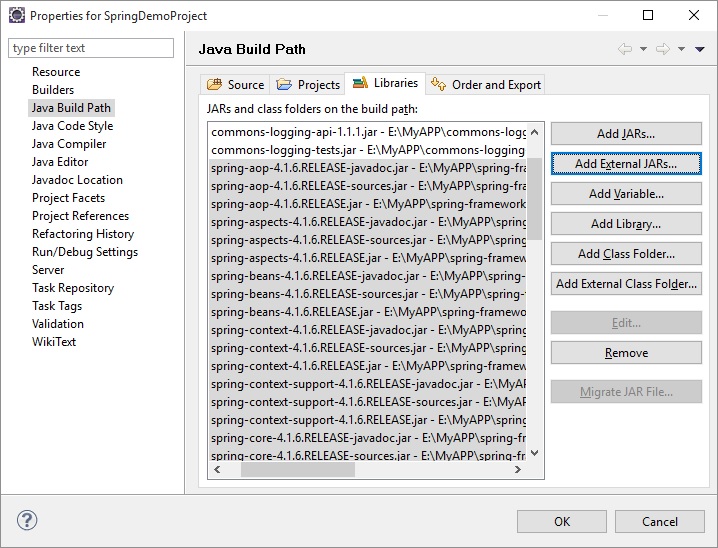


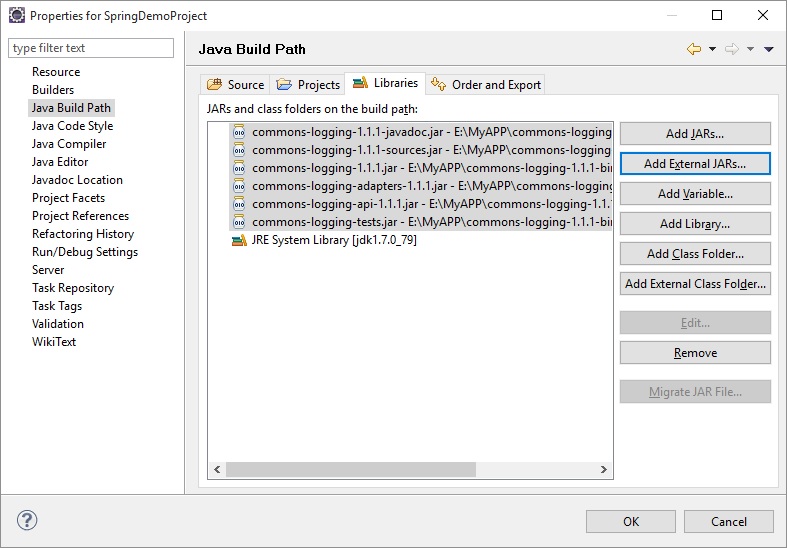


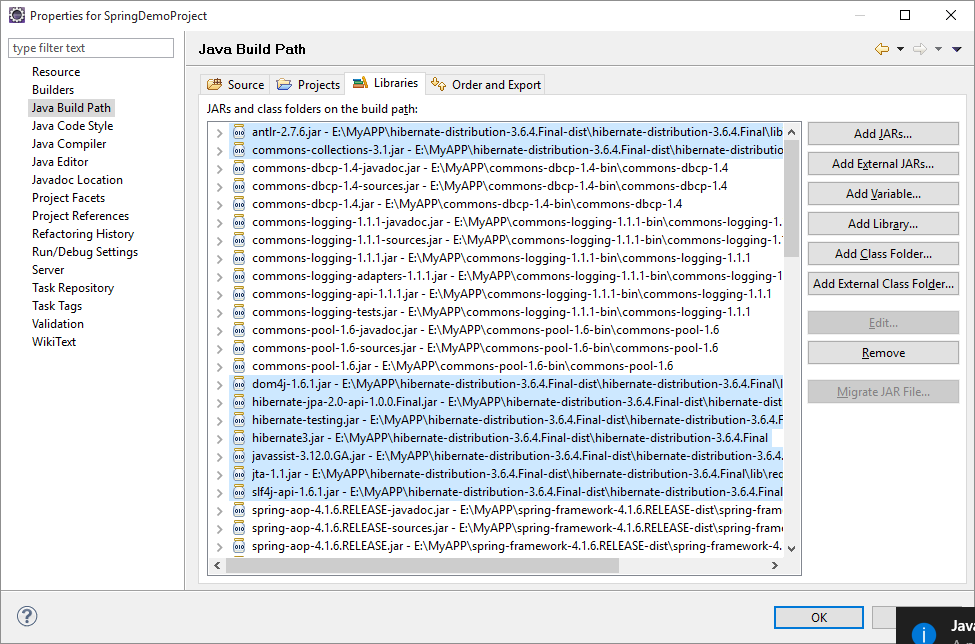
1. Create java project in eclipse named as **SpringDemoProject**

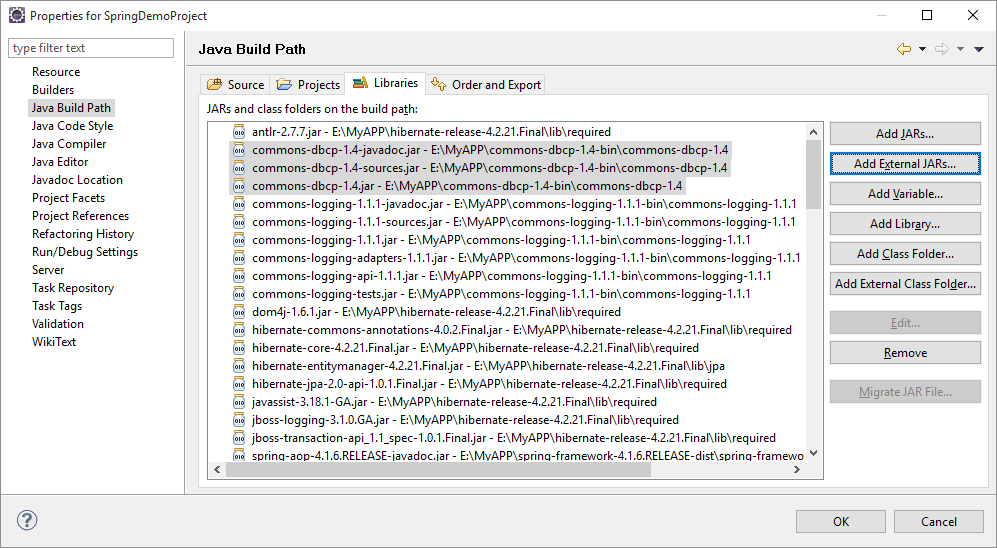


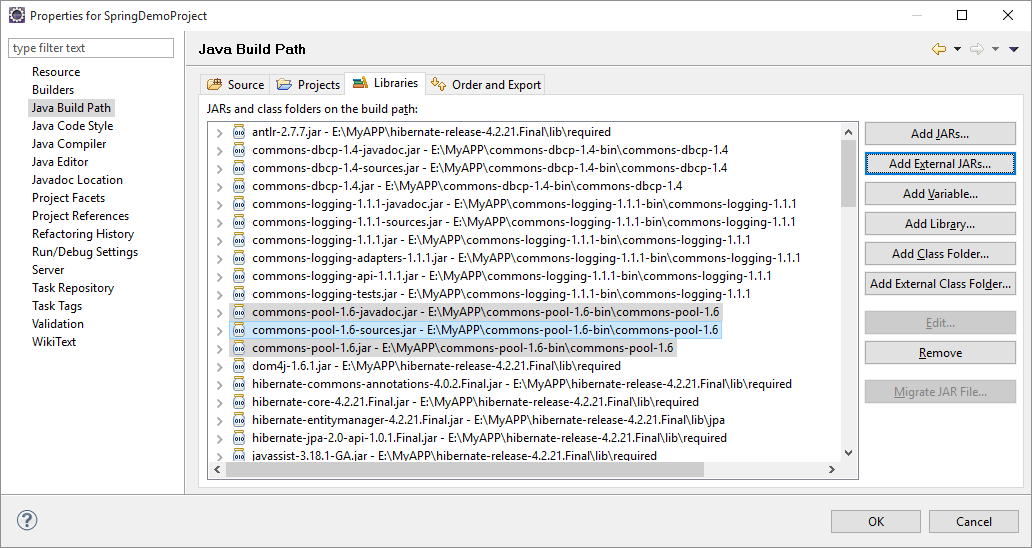
And add all mentioned jar files.

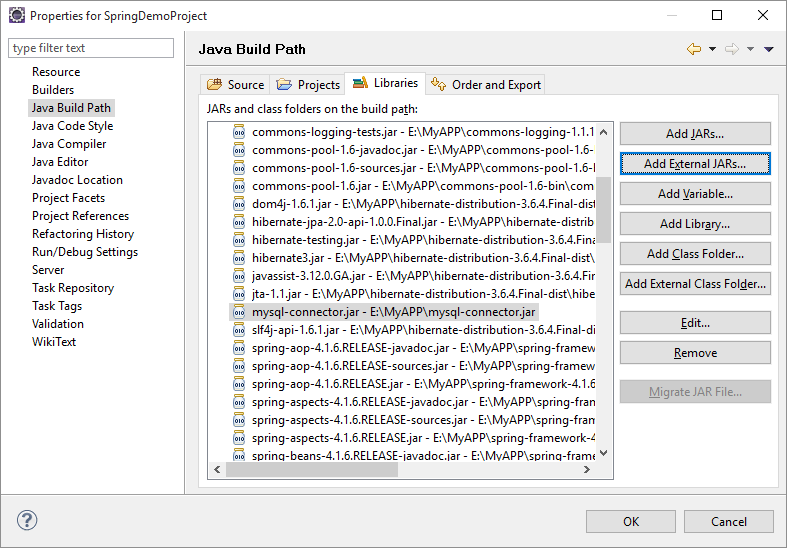






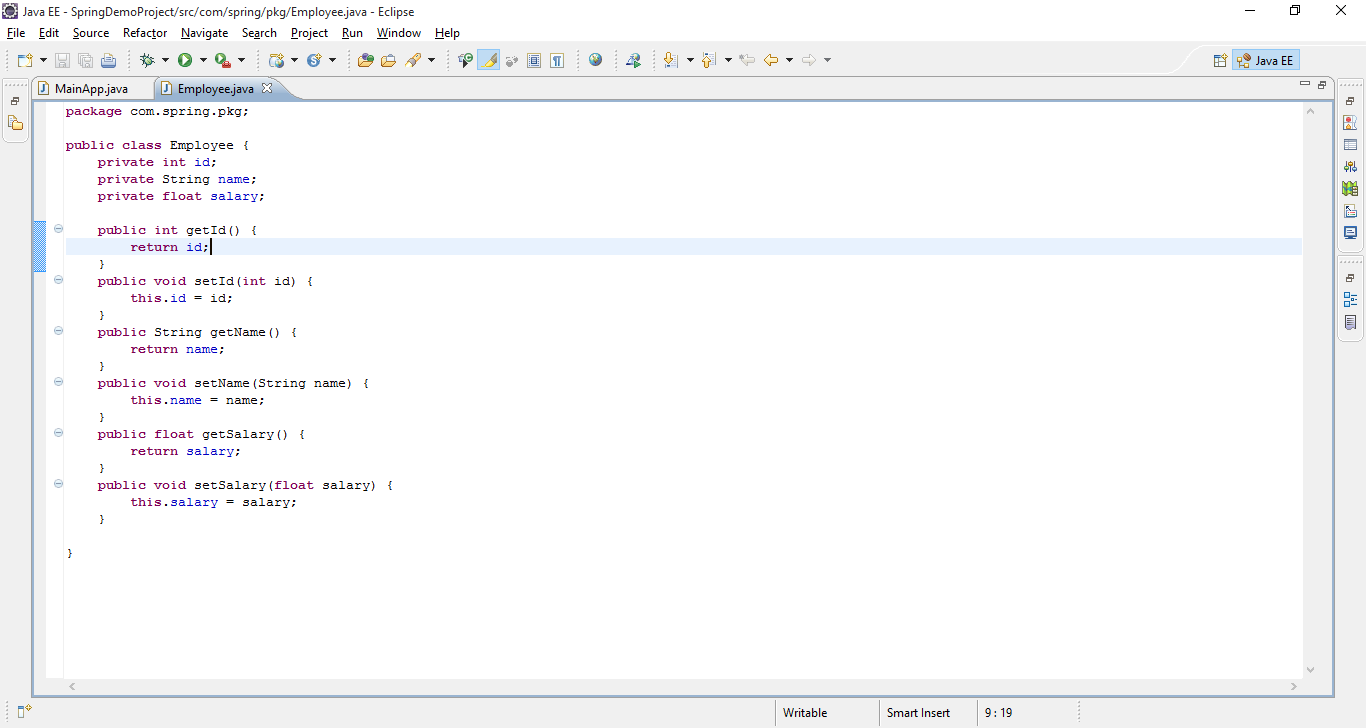




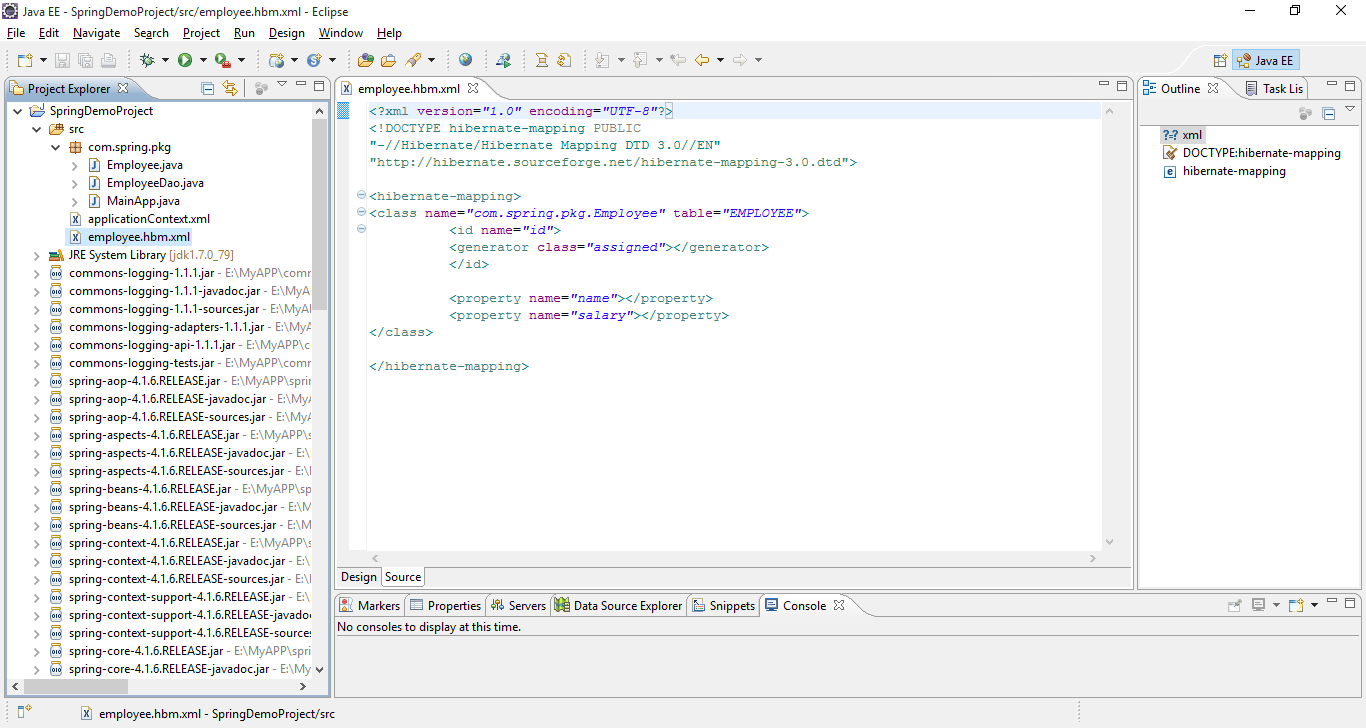


This project contains following files

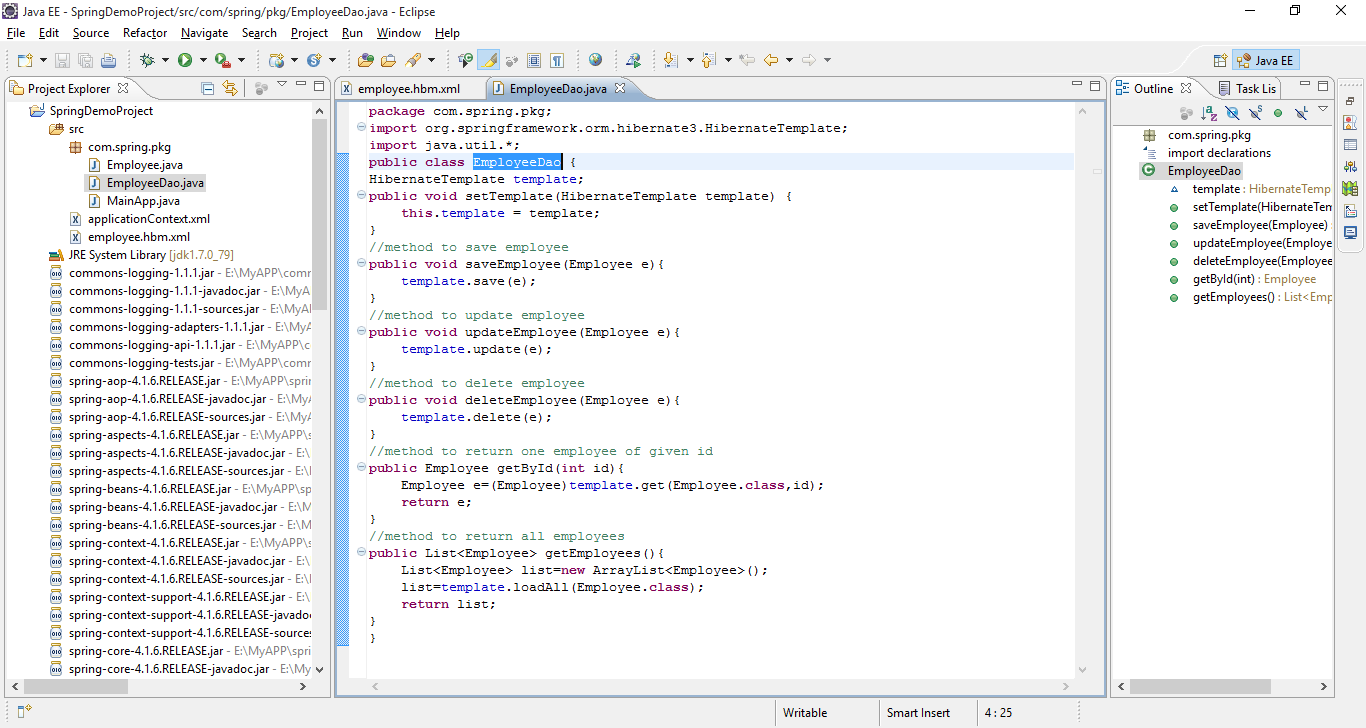
Employee.java



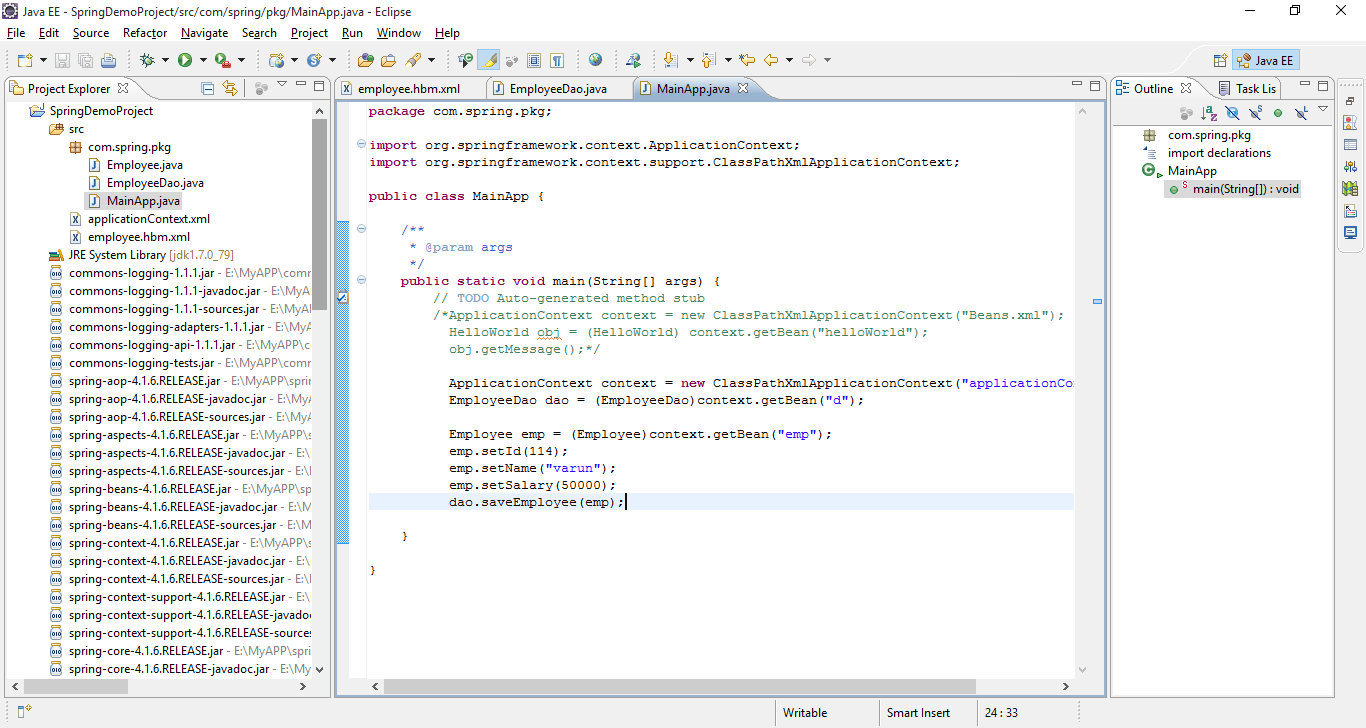
employee.hbm.xml



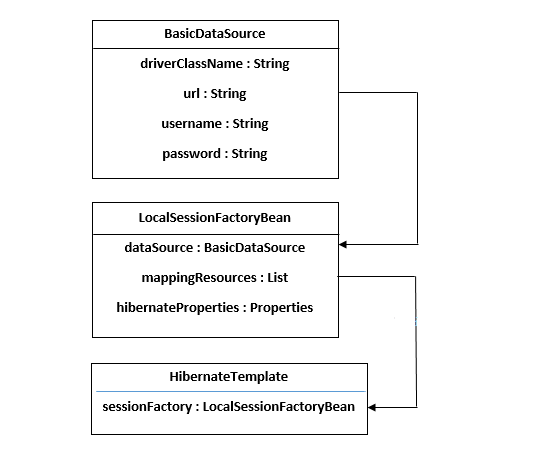
EmployeeDao.java

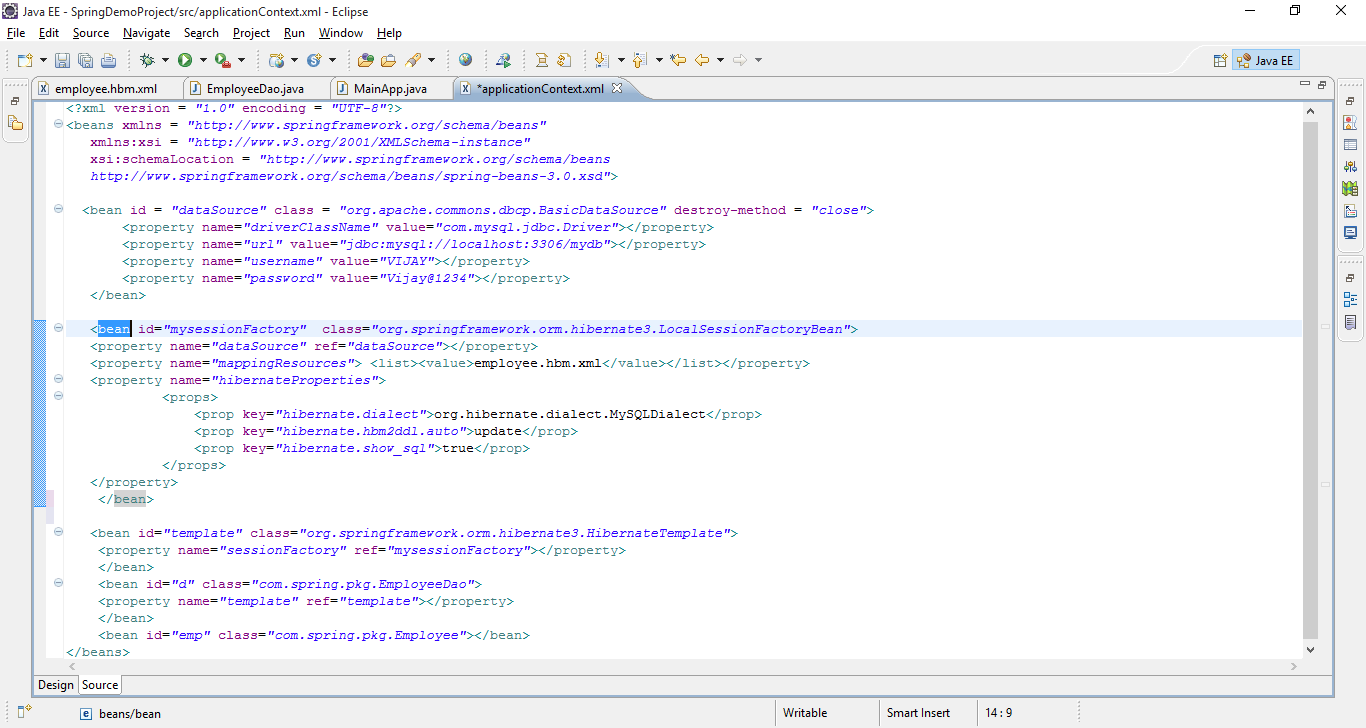


MainApp.java

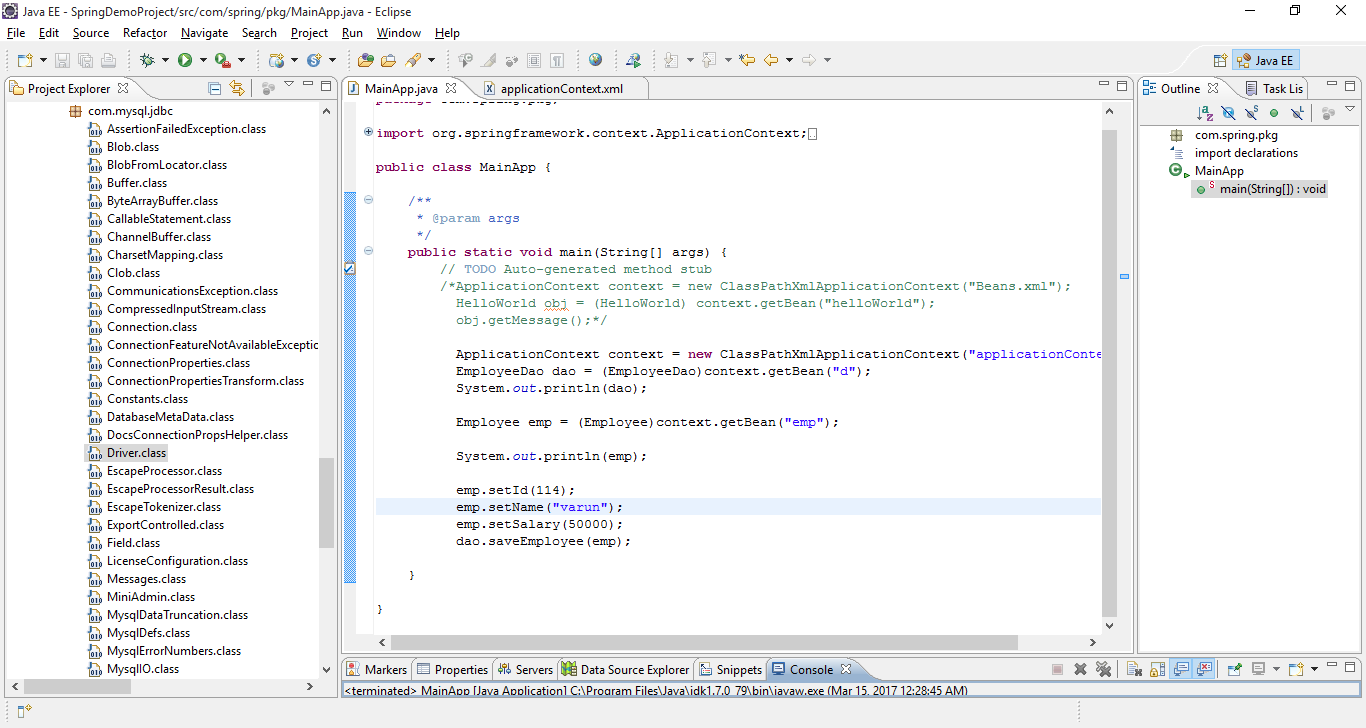


ApplicationContext.xml

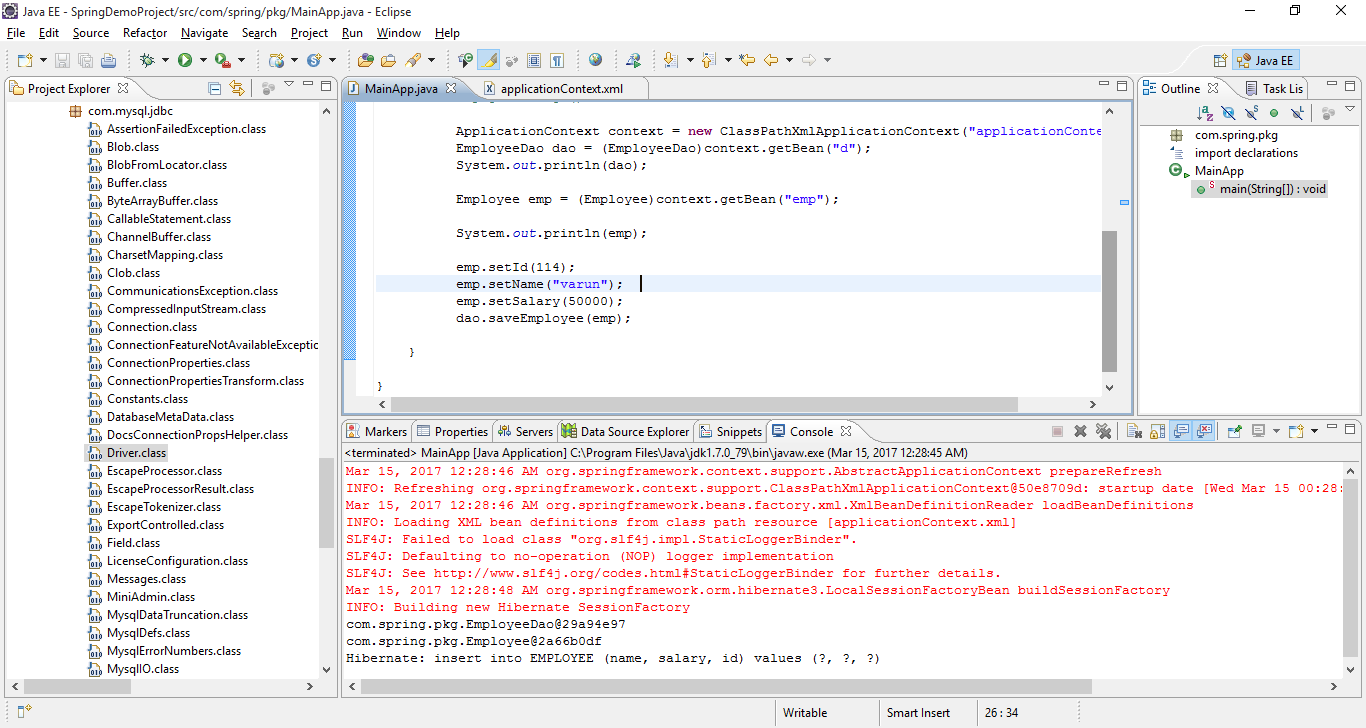




**CRUD Operation:-**

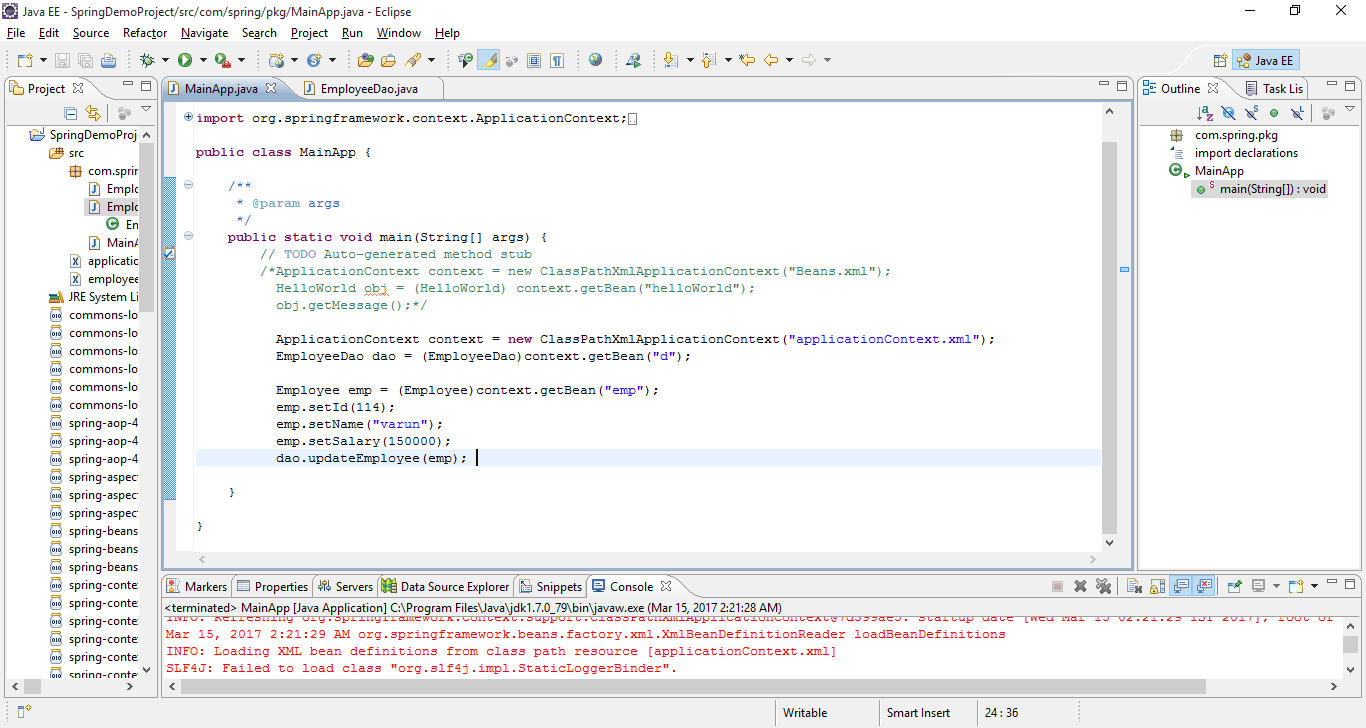


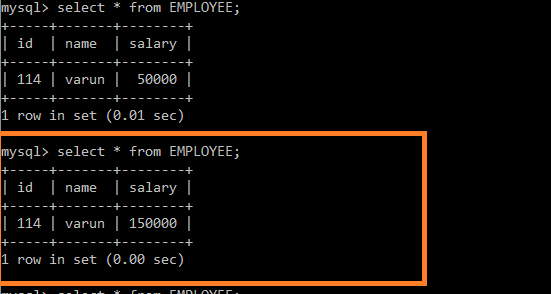
Insert first record:-



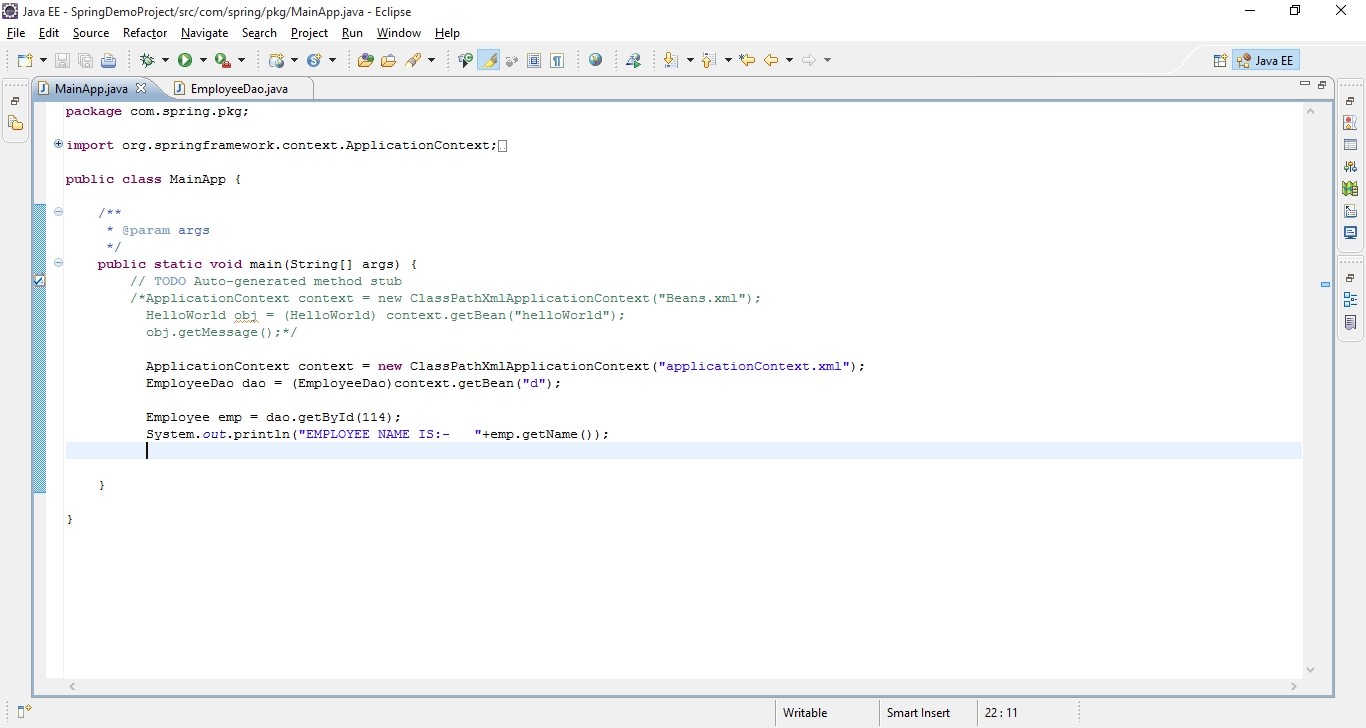


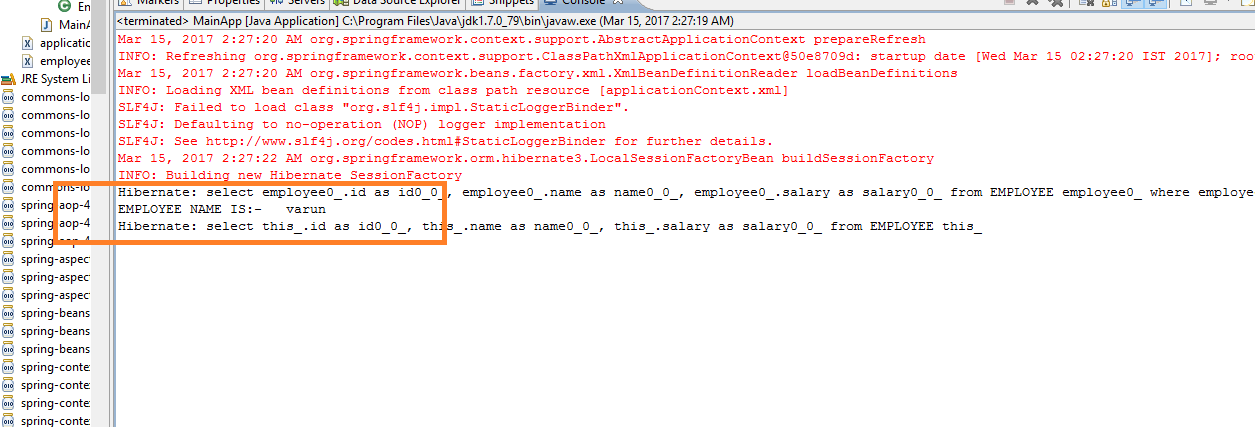
**Modify Record:-**



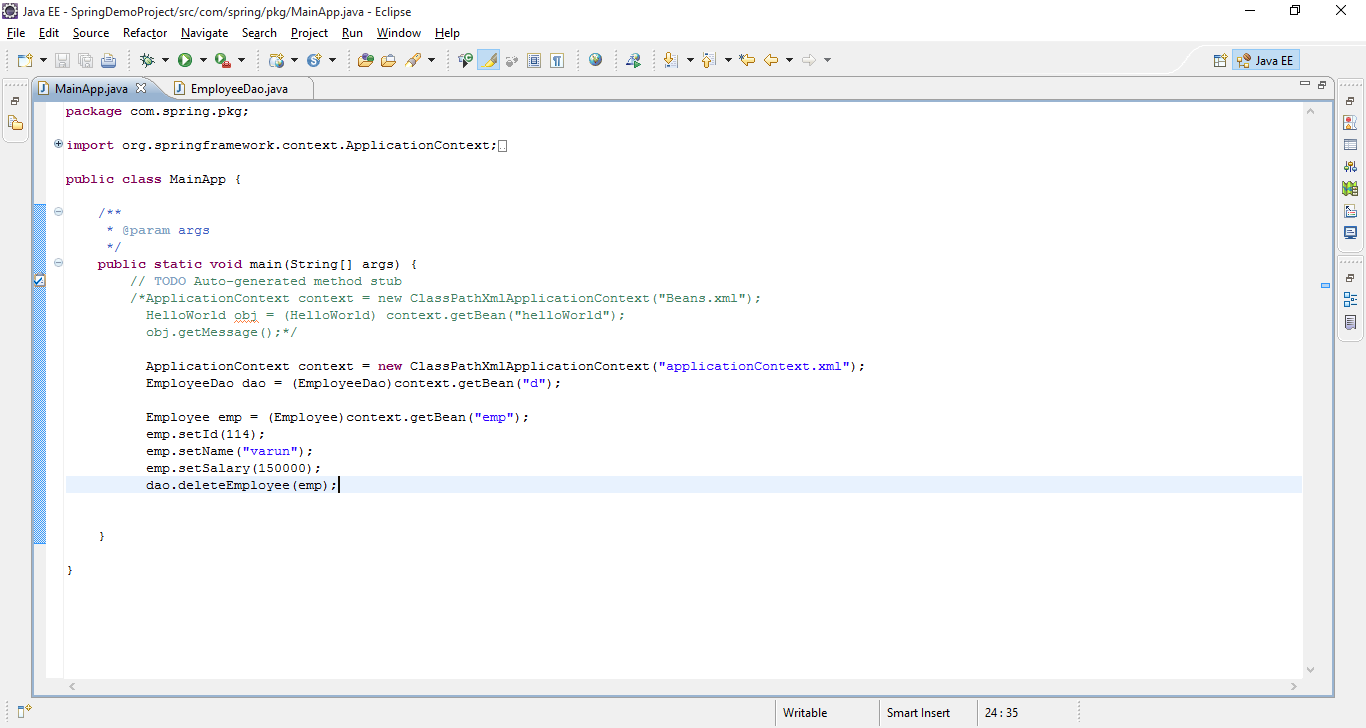


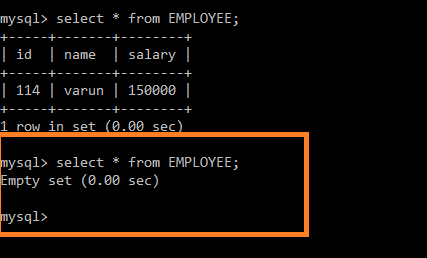
**Select Record :-**





**Delete Record:-**

****

****

Similarly Integration with any other database like mongoDb, We need to set the following configuration

1. In Spring configuration file,

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation=

"http://www.springframework.org/schema/beans

[http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"](http://www.springframework.org/schema/beans/spring-beans-3.0.xsd)>

<import resource="spring-mongodb-config.xml"/>

</beans>

And **spring-mongodb-config.xml** file as below,

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:mongo="http://www.springframework.org/schema/data/mongo"

xsi:schemaLocation=

"http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/data/mongo

http://www.springframework.org/schema/data/mongo/spring-mongo-1.0.xsd">

<!-- Default bean name is 'mongo' -->

<mongo:mongo host="localhost" port="27017">

<!-- OPTIONAL: configure <mongo:options /> -->

</mongo:mongo>

<mongo:db-factory dbname="pascalalma" mongo-ref="mongo"/>

<bean id="mongoTemplate" class="org.springframework.data.document.mongodb.MongoTemplate">

<constructor-arg ref="mongoDbFactory"/>

</bean>

</beans>

--------------------------------------------------------------------------------------------------------------------------------