**Hibernate Annotation-**

We can create hibernate application with annotation. There are many annotation that can be used to create the hibernate application such as @Entity, @Id, @Table, etc.

Package for annotation is javax.persistence.\*;

**Why?**

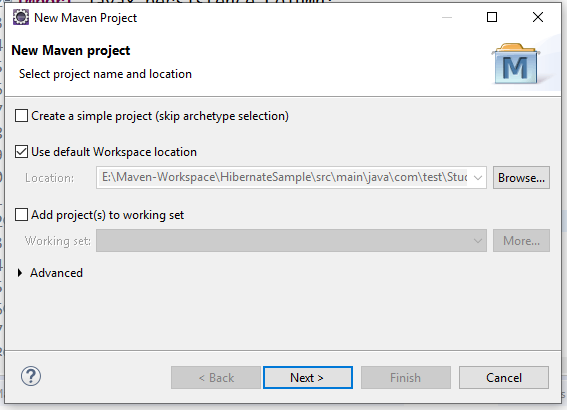
You don’t need to create mapping (hbm) file.

CRUD operation in hibernate

**Hibernate- Insert operation**

Maven project

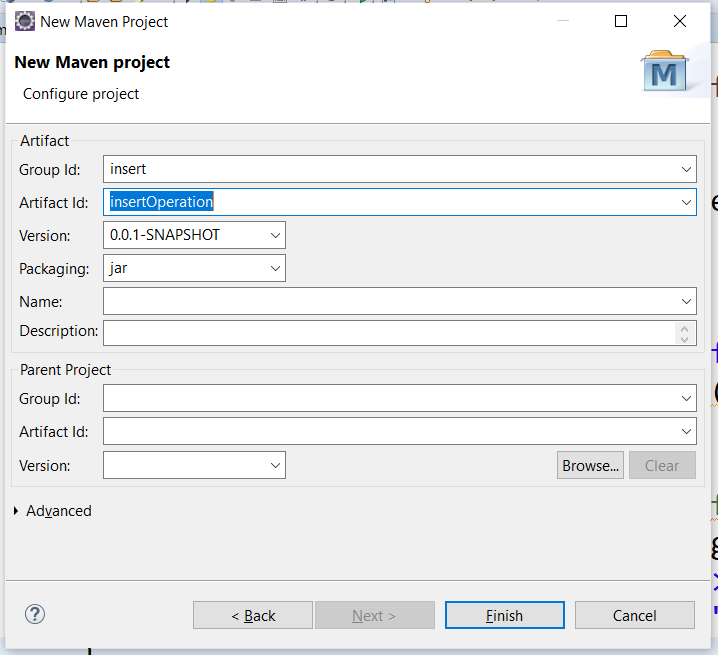
File->New->Maven Project->



Click on create simple project

Click on Next button

Enter the Group id and Artifact id



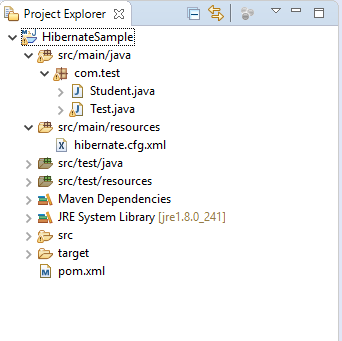
Click on Finish button

**Note-**

Mention the group id as any name, artifact id as any name.

Where group id is the project id and artifact id is project name.

Maven project structure looks like as



Go to pom.xml add hibernate and MySQL jar dependencies in that file

**Pom.xml**

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.1.4.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.21</version>

</dependency>

</dependencies>

**Student.java- POJO class**

**package** com.test;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="student")

**public** **class** Student {

@Id

@Column(name = "id")

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** **int** id;

@Column(name = "name")

**private** String name;

@Column(name = "city")

**private** String city;

@Column(name = "mobile")

**private** String mobile;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getMobile() {

**return** mobile;

}

**public** **void** setMobile(String mobile) {

**this**.mobile = mobile;

}

}

**hibernate.cfg.xml**

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

<!DOCTYPE hibernate-configuration SYSTEM

"classpath://org/hibernate/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/test</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*>root</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL57Dialect</property>

<property name=*"hbm2ddl.auto"*>create</property>

<property name=*"show\_sql"*>true</property>

<mapping class=*"com.test.Student"*></mapping>

</session-factory>

</hibernate-configuration>

**Dilect Meaning:-**

The dialect specifies the type of database used in hibernate so that hibernate generate appropriate type of SQL statements. For connecting any hibernate application with the database, it is required to provide the configuration of SQL dialect.

**Test.java**

**package** com.test;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Configuration configuration = **new** Configuration();

configuration.configure("hibernate.cfg.xml");

SessionFactory sessionFactory = configuration.~~buildSessionFactory~~();

Session session = sessionFactory.openSession();

Transaction transaction = session.beginTransaction();

// insert data into database

Student student = **new** Student();

student.setName("ram");

student.setCity("pune");

student.setMobile("9595972678");

session.save(student);

transaction.commit();

session.close();

System.***out***.println("Record saved successfully.");

}

}

Output-



Go to database and check the results whether record is inserted or not.

**Hibernate- Update operation**

**Student.Java**

**package** com.demo;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="student")

**public** **class** Student {

@Id

@Column(name = "id")

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** **int** id;

@Column(name = "name")

**private** String name;

@Column(name = "city")

**private** String city;

@Column(name = "mobile")

**private** String mobile;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getMobile() {

**return** mobile;

}

**public** **void** setMobile(String mobile) {

**this**.mobile = mobile;

}

}

**hibernate.cfg.xml**

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

<!DOCTYPE hibernate-configuration SYSTEM

"classpath://org/hibernate/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/test</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*>root</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL57Dialect</property>

<property name=*"hbm2ddl.auto"*>update</property>

<property name=*"show\_sql"*>true</property>

<mapping class=*"com.test.Student"*></mapping>

</session-factory>

</hibernate-configuration>

**Test.java**

**package** com.demo;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Configuration configuration = **new** Configuration();

configuration.configure("hibernate.cfg.xml");

SessionFactory sessionFactory = configuration.~~buildSessionFactory~~();

Session session = sessionFactory.openSession();

Transaction transaction = session.beginTransaction();

Student student = (Student) session.get(Student.**class**, 1);

student.setName("rohit");

session.update(student);

transaction.commit();

session.close();

sessionFactory.close();

System.***out***.println("Record updated successfully.");

}

}

**transaction.commit():-**

In the context of a database transaction, a commit refers to the saving of data permanently after a set of tentative changes. A commit ends a transaction within a relational database and allows all other users to see the changes.

Run the application

Go to database and check the results.



**Hibernate- Delete operation**

**Student class**

**package** com.demo;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="student")

**public** **class** Student {

@Id

@Column(name = "id")

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** **int** id;

@Column(name = "name")

**private** String name;

@Column(name = "city")

**private** String city;

@Column(name = "mobile")

**private** String mobile;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getMobile() {

**return** mobile;

}

**public** **void** setMobile(String mobile) {

**this**.mobile = mobile;

}

}

**hibernate.cfg.xml**

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

<!DOCTYPE hibernate-configuration SYSTEM

"classpath://org/hibernate/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/test</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*>root</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL57Dialect</property>

<property name=*"hbm2ddl.auto"*>update</property>

<property name=*"show\_sql"*>true</property>

<mapping class=*"com.test.Student"*></mapping>

</session-factory>

</hibernate-configuration>

**Test.Java**

**package** com.demo;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Configuration configuration = **new** Configuration();

configuration.configure("hibernate.cfg.xml");

SessionFactory sessionFactory = configuration.~~buildSessionFactory~~();

Session session = sessionFactory.openSession();

Transaction transaction = session.beginTransaction();

Student student = (Student) session.get(Student.**class**, 1);

session.delete(student);

transaction.commit();

session.close();

sessionFactory.close();

System.***out***.println("Record deleted successfully.");

}

}

Run the application

Hibernate: delete from student where id=? Record deleted successfully.

**Hibernate- Select operation**

**Student.java**

**package** com.demo;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="student")

**public** **class** Student {

@Id

@Column(name = "id")

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** **int** id;

@Column(name = "name")

**private** String name;

@Column(name = "city")

**private** String city;

@Column(name = "mobile")

**private** String mobile;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getMobile() {

**return** mobile;

}

**public** **void** setMobile(String mobile) {

**this**.mobile = mobile;

}

}

**hibernate.cfg.xml**

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

<!DOCTYPE hibernate-configuration SYSTEM

"classpath://org/hibernate/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/test</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*>root</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL57Dialect</property>

<property name=*"hbm2ddl.auto"*>update</property>

<property name=*"show\_sql"*>true</property>

<mapping class=*"com.test.Student"*></mapping>

</session-factory>

</hibernate-configuration>

**Test.Java**

**package** com.demo;

**import** java.util.\*;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Configuration cfg = **new** Configuration();

cfg.configure("hibernate.cfg.xml");

SessionFactory sessionFactory = cfg.~~buildSessionFactory~~();

Session session = sessionFactory.openSession();

// pass the class name

Query query = session.createQuery("from Student");

List<Student>students = query.list();

**for** (Student student : students) {

System.***out***.println("ID="+student.getId());

System.***out***.println("City="+student.getCity());

System.***out***.println("Mobile="+student.getMobile());

System.***out***.println("Name="+student.getMobile());

}

session.close();

sessionFactory.close();

System.***out***.println("Record retrieved successfully.");

}

}

Run the application

Output

Hibernate: select student0\_.id as id1\_0\_, student0\_.city as city2\_0\_, student0\_.mobile as mobile3\_0\_, student0\_.name as name4\_0\_ from student student0\_

ID=1

City=pune

Mobile=7575858595

Name=7575858595

ID=2

City=mumbai

Mobile=8585959575

Name=8585959575

Record retrieved successfully.