

**Hibernate.cfg.xml**:

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

<!DOCTYPE hibernate-configuration SYSTEM

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

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

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/empdb?useSSL=false</property>

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

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

<property name=*"dialect"*>org.hibernate.dialect.MySQL5Dialect</property>

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

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

<!-- tag added to create a new table from the class. -->

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

<mapping class=*"com.hibersessions.Address"* />

</session-factory>

</hibernate-configuration>

Address.java:

**package** com.hibersessions;

**import** java.util.Date;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Lob;

**import** javax.persistence.Table;

**import** javax.persistence.Temporal;

**import** javax.persistence.TemporalType;

**import** javax.persistence.Transient;

@Entity

@Table(name = "stu\_address")

**public** **class** Address {

@Id

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

@Column(name = "address\_id")

**private** **int** addressId;

@Column(length = 50, name = "stu\_street")

**private** String street;

@Column(length = 30, name = "Visakhapatnam")

**private** String city;

**private** **boolean** isOpen;

@Transient

**private** **double** tollCharges;

@Column(name = "street\_renvo\_dt")

@Temporal(TemporalType.***DATE***) //displays only date but not time

**private** Date renovationDate;

@Lob

**private** **byte**[] image;

**public** Address() {

**super**();

}

**public** Address(**int** addressId, String street, String city, **boolean** isOpen, **double** tollCharges, Date renovationDate,

**byte**[] image) {

**super**();

**this**.addressId = addressId;

**this**.street = street;

**this**.city = city;

**this**.isOpen = isOpen;

**this**.tollCharges = tollCharges;

**this**.renovationDate = renovationDate;

**this**.image = image;

}

**public** **int** getAddressId() {

**return** addressId;

}

**public** **void** setAddressId(**int** addressId) {

**this**.addressId = addressId;

}

**public** String getStreet() {

**return** street;

}

**public** **void** setStreet(String street) {

**this**.street = street;

}

**public** String getCity() {

**return** city;

}

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

**this**.city = city;

}

**public** **boolean** isOpen() {

**return** isOpen;

}

**public** **void** setOpen(**boolean** isOpen) {

**this**.isOpen = isOpen;

}

**public** **double** getTollCharges() {

**return** tollCharges;

}

**public** **void** setTollCharges(**double** tollCharges) {

**this**.tollCharges = tollCharges;

}

**public** Date getRenovationDate() {

**return** renovationDate;

}

**public** **void** setRenovationDate(Date renovationDate) {

**this**.renovationDate = renovationDate;

}

**public** **byte**[] getImage() {

**return** image;

}

**public** **void** setImage(**byte**[] image) {

**this**.image = image;

}

}

Main.java:

**package** com.hibersessions;

**import** java.util.Date;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

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

**public** **class** TestHiberAnnotation\_2 {

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

SessionFactory factory = **new** Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

System.***out***.println(factory);

Address address = **new** Address();

address.setAddressId(530020);

address.setCity("Visakhapatnam");

address.setStreet("Dabagardens");

address.setOpen(**true**);

address.setRenovationDate(**new** Date());

address.setTollCharges(500.34);

address.getImage();

// Triggering the session

Session session = factory.openSession();

Transaction tx = session.beginTransaction();

session.save(address);

tx.commit();

session.close();

System.***out***.println("Completed Saving !!!");

}

}

**Console Output:**

Oct 09, 2023 4:33:52 PM org.hibernate.Version logVersion

INFO: HHH000412: Hibernate ORM core version 5.6.15.Final

Oct 09, 2023 4:33:54 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>

INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}

Oct 09, 2023 4:33:55 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure

WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)

Oct 09, 2023 4:33:55 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/empdb?useSSL=false]

Oct 09, 2023 4:33:55 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001001: Connection properties: {password=\*\*\*\*, user=root}

Oct 09, 2023 4:33:55 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001003: Autocommit mode: false

Oct 09, 2023 4:33:55 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>

INFO: HHH000115: Hibernate connection pool size: 20 (min=1)

Oct 09, 2023 4:33:56 PM org.hibernate.dialect.Dialect <init>

INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect

Oct 09, 2023 4:33:58 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection

INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@1dbd580] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.

Hibernate: create table stu\_address (address\_id integer not null auto\_increment, Visakhapatnam varchar(30), image longblob, isOpen bit not null, street\_renvo\_dt date, stu\_street varchar(50), primary key (address\_id)) engine=MyISAM

Oct 09, 2023 4:33:58 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService

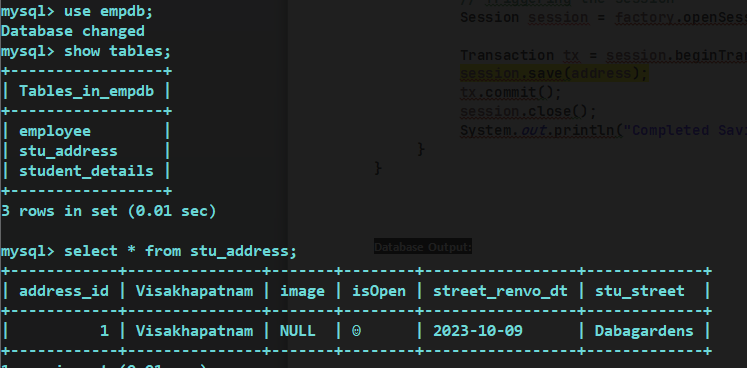
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]

org.hibernate.internal.SessionFactoryImpl@4eb166a1

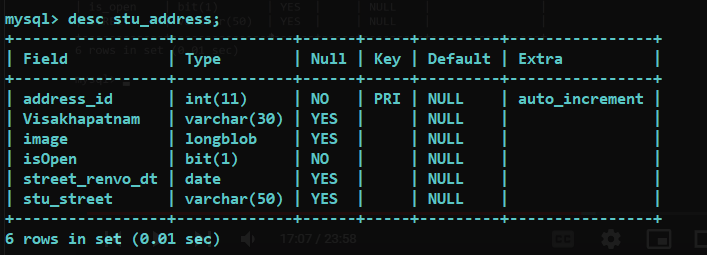
Hibernate: insert into stu\_address (Visakhapatnam, image, isOpen, street\_renvo\_dt, stu\_street) values (?, ?, ?, ?, ?)

Completed Saving !!!

**Database Output:**

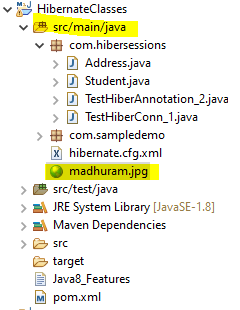


**Table that is Created:**



**NOTE: CRemove the ‘Transient’ annotation and try again.**

**After adding the image the code and output will be like the following:**



**Main class:**

**package** com.hibersessions;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.util.Date;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

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

**public** **class** TestHiberAnnotation\_2 {

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

SessionFactory factory = **new** Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

System.***out***.println(factory);

Address address = **new** Address();

address.setAddressId(530020);

address.setCity("Visakhapatnam");

address.setStreet("Dabagardens");

address.setOpen(**true**);

address.setRenovationDate(**new** Date());

address.setTollCharges(500.34);

//adding image

String file = "E:\\Programming\\eclipse-workspace\\HibernateClasses\\src\\main\\java\\madhuram.jpg";

FileInputStream fis = **new** FileInputStream(file);

//gets the image in 'fis' object, converts to byte and sends to 'setImage()' setter class.

address.setImage(**new** **byte**[fis.available()]);

// Triggering the session

Session session = factory.openSession();

Transaction tx = session.beginTransaction();

session.save(address);

tx.commit();

session.close();

fis.close();

System.***out***.println("Completed Saving !!!");

}

}

**Console Output:**

Oct 09, 2023 5:29:03 PM org.hibernate.Version logVersion

INFO: HHH000412: Hibernate ORM core version 5.6.15.Final

Oct 09, 2023 5:29:04 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>

INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}

Oct 09, 2023 5:29:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure

WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)

Oct 09, 2023 5:29:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/empdb?useSSL=false]

Oct 09, 2023 5:29:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001001: Connection properties: {password=\*\*\*\*, user=root}

Oct 09, 2023 5:29:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001003: Autocommit mode: false

Oct 09, 2023 5:29:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>

INFO: HHH000115: Hibernate connection pool size: 20 (min=1)

Oct 09, 2023 5:29:05 PM org.hibernate.dialect.Dialect <init>

INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect

Oct 09, 2023 5:29:06 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection

INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@1dbd580] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.

Oct 09, 2023 5:29:06 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService

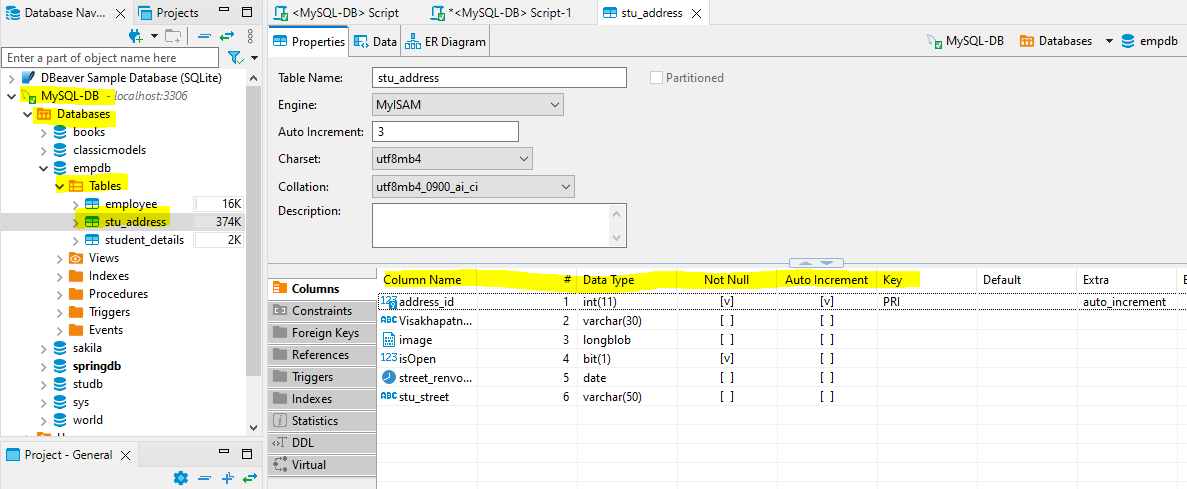
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]

org.hibernate.internal.SessionFactoryImpl@46383a78

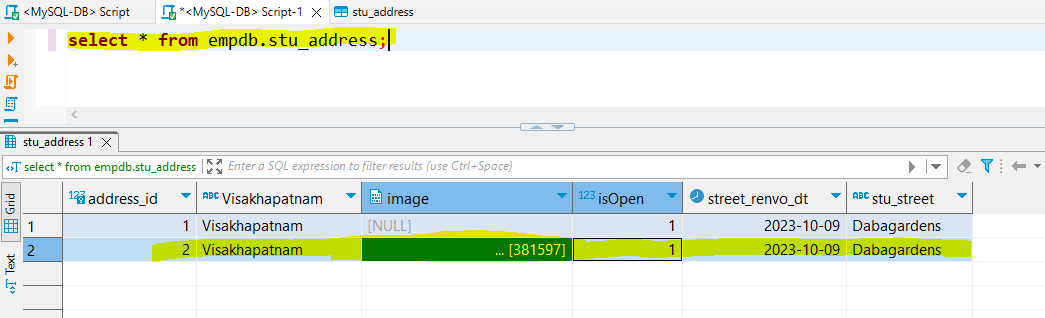
Hibernate: insert into stu\_address (Visakhapatnam, image, isOpen, street\_renvo\_dt, stu\_street) values (?, ?, ?, ?, ?)

Completed Saving !!!

**Database Screen Shots:**



**Table screen shot:**



***Note: The image is inserted in the table.***