**21.2. Hibernate CRUD**

**Hibernate CRUD**:

1. Configure Hibernate
2. Create Entity class
3. Create object and store it in database
4. Read Object
5. Update Object
6. Delete Object

**1) Configure Hibernate (File: hibernate.cfg.xml):**

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

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

<hibernate-configuration>

<session-factory>

<!-- JDBC Database connection settings -->

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

<property name=*"connection.url"*>

jdbc:mysql://localhost:3306/hb\_student\_tracker?useSSL=false&amp;serverTimezone=UTC

</property>

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

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

<!-- JDBC connection pool settings ... using built-in test pool -->

<property name=*"connection.pool\_size"*>1</property>

<!-- Select our SQL dialect -->

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

<!-- Echo the SQL to stdout -->

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

<!-- Set the current session context -->

<property name=*"current\_session\_context\_class"*>thread</property>

</session-factory>

</hibernate-configuration>

**2) Create Entity class (Student.class)**

**package** com.ruhul.hibernate.entity;

**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

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

@Column(name = "id")

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

@Column(name = "first\_name")

**private** String firstName;

@Column(name = "last\_name")

**private** String lastName;

@Column(name = "email")

**private** String email;

**public** Student() {

}

**public** Student(String firstName, String lastName, String email) {

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.email = email;

}

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

**return** id;

}

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

**this**.id = id;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

@Override

**public** String toString() {

**return** "Student [id=" + id + ", firstName=" + firstName + ", lastName=" + lastName + ", email=" + email + "]";

}

}

**3) Create and Store Object (CreateStudentDemo.class)**:

**package** com.ruhul.hibernateDemo;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

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

**import** com.ruhul.hibernate.entity.Student;

**public** **class** CreateStudentDemo {

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

//create session factory

SessionFactory factory = **new** Configuration()

.configure("hibernate.cfg.xml")

.addAnnotatedClass(Student.**class**)

.buildSessionFactory();

//create session

Session session = factory.getCurrentSession();

**try** {

//create a student object

System.***out***.println("Create new student object...");

Student tempStudent = **new** Student("Md. Ruhul Amin", "Ruhul", "ruhul@gmail.com");

Student tempStudent1 = **new** Student("Md. Rezaul Islam", "Reza", "reza@gmail.com");

Student tempStudent2 = **new** Student("Md. Ariful Islam", "Arif", "arif@gmail.com");

Student tempStudent3 = **new** Student("Md. Rafiul Islam", "Rafi", "rafi@gmail.com");

//start transaction

session.beginTransaction();

// save the student

System.***out***.println("Save the student...");

session.save(tempStudent);

session.save(tempStudent1);

session.save(tempStudent2);

session.save(tempStudent3);

// commit the transaction

session.getTransaction().commit();

System.***out***.println("Done!!!");

}

**finally** {

factory.close();

}

}

}

**Output**:

Create new student object...

Save the student...

Hibernate: insert into student (email, first\_name, last\_name) values (?, ?, ?)

Hibernate: insert into student (email, first\_name, last\_name) values (?, ?, ?)

Hibernate: insert into student (email, first\_name, last\_name) values (?, ?, ?)

Hibernate: insert into student (email, first\_name, last\_name) values (?, ?, ?)

Done!!!

**4) Read Object**:

For retrieve hibernate data we have to know about Hibernate Query Language.

**Hibernate Query Language (HQL)**:

Hibernate Query Language (HQL) is an object-oriented query language, similar to SQL, but instead of operating on tables and columns, HQL works with persistent objects and their properties. HQL queries are translated by Hibernate into conventional SQL queries, which in turns perform action on database.

**Retrieving all Students**:

List<Student> theStudents =

session.createQuery("from Student").getResultList();

**Retrieve student with lastName: Ruhul:**

List<Student> theStudents =

session.createQuery("from Student s where s.lastName = 'Ruhul'").getResultList();

**Retrieve students with lastName="Ruhul" OR email="reza@gmail.com":**

List<Student> theStudents = session

.createQuery("from Student s where s.lastName = 'Ruhul' or s.email = 'reza@gmail.com'").getResultList();

**QueryStudentDemo.class**:

**package** com.ruhul.hibernateDemo;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

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

**import** com.ruhul.hibernate.entity.Student;

**public** **class** QueryStudentDemo {

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

// create session factory

SessionFactory factory = **new** Configuration()

.configure("hibernate.cfg.xml")

.addAnnotatedClass(Student.**class**)

.buildSessionFactory();

// create session

Session session = factory.getCurrentSession();

**try** {

// start a transaction

session.beginTransaction();

// display all students

System.***out***.println("Display all students");

List<Student> theStudents = session

.createQuery("from Student").getResultList();

*displayStudents*(theStudents);

// Display student wth last name: 'Ruhul'

System.***out***.println("\n\nDisplay student wth last name: 'Ruhul'");

theStudents = session

.createQuery("from Student s where s.lastName = 'Ruhul'")

.getResultList();

*displayStudents*(theStudents);

// display students with lastName="Ruhul" OR email= "reza@gmail.com"

System.***out***.println("\n\nStudents who have last name of : Ruhul OR email of: reza@gmail.com");

theStudents = session

.createQuery("from Student s where s.lastName = 'Ruhul' or s.email = 'reza@gmail.com'")

.getResultList();

*displayStudents*(theStudents);

// commit the transaction

session.getTransaction().commit();

System.***out***.println("Done!!!");

} **finally** {

factory.close();

}

}

// method for display students

**private** **static** **void** displayStudents(List<Student> theStudents) {

**for** (Student tempStudent : theStudents) {

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

}

}

}

**Output**:

Display all students

Hibernate: select student0\_.id as id1\_0\_, student0\_.email as email2\_0\_, student0\_.first\_name as first\_na3\_0\_, student0\_.last\_name as last\_nam4\_0\_ from student student0\_

Student [id=1, firstName=Md. Ruhul Amin, lastName=Ruhul, email=ruhul@gmail.com]

Student [id=2, firstName=Md. Rezaul Islam, lastName=Reza, email=reza@gmail.com]

Student [id=3, firstName=Md. Ariful Islam, lastName=Arif, email=arif@gmail.com]

Student [id=4, firstName=Md. Rafiul Islam, lastName=Rafi, email=rafi@gmail.com]

Student [id=5, firstName=Mst. Shanjana Hossain, lastName=Shanju, email=shanju@gmail.com]

Display student wth last name: 'Ruhul'

Hibernate: select student0\_.id as id1\_0\_, student0\_.email as email2\_0\_, student0\_.first\_name as first\_na3\_0\_, student0\_.last\_name as last\_nam4\_0\_ from student student0\_ where student0\_.last\_name='Ruhul'

Student [id=1, firstName=Md. Ruhul Amin, lastName=Ruhul, email=ruhul@gmail.com]

Students who have last name of : Ruhul OR email of: reza@gmail.com

Hibernate: select student0\_.id as id1\_0\_, student0\_.email as email2\_0\_, student0\_.first\_name as first\_na3\_0\_, student0\_.last\_name as last\_nam4\_0\_ from student student0\_ where student0\_.last\_name='Ruhul' or student0\_.email='reza@gmail.com'

Student [id=1, firstName=Md. Ruhul Amin, lastName=Ruhul, email=ruhul@gmail.com]

Student [id=2, firstName=Md. Rezaul Islam, lastName=Reza, email=reza@gmail.com]

Done!!!

**ReadStudentDemo.class**:

**package** com.ruhul.hibernateDemo;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

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

**import** com.ruhul.hibernate.entity.Student;

**public** **class** ReadStudentDemo {

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

// create session factory

SessionFactory factory = **new** Configuration()

.configure("hibernate.cfg.xml")

.addAnnotatedClass(Student.**class**)

.buildSessionFactory();

// create session

Session session = factory.getCurrentSession();

**try** {

// create a student object

System.***out***.println("Creating new student object...");

Student tempStudent = **new** Student("Mst. Shanjana Hossain", "Shanju", "shanju@gmail.com");

// start transaction

session.beginTransaction();

// save the student

System.***out***.println("Save the student...");

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

session.save(tempStudent);

// commit the transaction

session.getTransaction().commit();

//find out the student's id: primary key

System.***out***.println("Save student. Generate id: " + tempStudent.getId());

//now get a new session and start transaction

session = factory.getCurrentSession();

session.beginTransaction();

//retrieve student based on the id: primary key

System.***out***.println("\nGetting student with id:" + tempStudent.getId());

Student myStudent = session.get(Student.**class**, tempStudent.getId());

System.***out***.println("Get complete: "+myStudent);

//commit the transaction

session.getTransaction().commit();

System.***out***.println("Done!!!");

} **finally** {

factory.close();

}

}

}

**Output**:

Creating new student object...

Save the student...

Student [id=0, firstName=Mst. Shanjana Hossain, lastName=Shanju, email=shanju@gmail.com]

Hibernate: insert into student (email, first\_name, last\_name) values (?, ?, ?)

Save student. Generate id: 10

Getting student with id:10

Hibernate: select student0\_.id as id1\_0\_0\_, student0\_.email as email2\_0\_0\_, student0\_.first\_name as first\_na3\_0\_0\_, student0\_.last\_name as last\_nam4\_0\_0\_ from student student0\_ where student0\_.id=?

Get complete: Student [id=10, firstName=Mst. Shanjana Hossain, lastName=Shanju, email=shanju@gmail.com]

Done!!!

**5) UpdateStudentDemo.class**:

**package** com.ruhul.hibernateDemo;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

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

**import** com.ruhul.hibernate.entity.Student;

**public** **class** UpdateStudentDemo {

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

// create session factory

SessionFactory factory = **new** Configuration()

.configure("hibernate.cfg.xml")

.addAnnotatedClass(Student.**class**)

.buildSessionFactory();

// create session

Session session = factory.getCurrentSession();

**try** {

**int** studentId = 1;

//start a transaction

session.beginTransaction();

//retrieve student based on the id: primary key

System.***out***.println("\nGetting student with id:" + studentId);

Student myStudent = session.get(Student.**class**, studentId);

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

//update student

System.***out***.println("\n\nUpdating student...");

myStudent.setFirstName("Md. Ruhul Amin");

session.getTransaction().commit();

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

//update email for all student

System.***out***.println("\n\nupdate email for all student");

session = factory.getCurrentSession();

session.beginTransaction();

session.createQuery("update Student set email = 'demo\_email@gmail.com'").executeUpdate();

session.getTransaction().commit();

System.***out***.println("\n\nDone!!!");

}

**finally** {

factory.close();

}

}

}

**Output**:

Getting student with id:1

Hibernate: select student0\_.id as id1\_0\_0\_, student0\_.email as email2\_0\_0\_, student0\_.first\_name as first\_na3\_0\_0\_, student0\_.last\_name as last\_nam4\_0\_0\_ from student student0\_ where student0\_.id=?

Student [id=1, firstName=Md. Ruhul, lastName=Ruhul, email=demo\_email@gmail.com]

Updating student...

Hibernate: update student set email=?, first\_name=?, last\_name=? where id=?

Student [id=1, firstName=Md. Ruhul Amin, lastName=Ruhul, email=demo\_email@gmail.com]

update email for all student

Hibernate: update student set email='demo\_email@gmail.com'

Done!!!

**6) DeleteStudentDemo.class**:

**package** com.ruhul.hibernateDemo;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

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

**import** com.ruhul.hibernate.entity.Student;

**public** **class** DeleteStudentDemo {

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

// create session factory

SessionFactory factory = **new** Configuration()

.configure("hibernate.cfg.xml")

.addAnnotatedClass(Student.**class**)

.buildSessionFactory();

// create session

Session session = factory.getCurrentSession();

**try** {

// start a transaction

session.beginTransaction();

// display all students

System.***out***.println("Display all students");

List<Student> theStudents = session.createQuery("from Student").getResultList();

*displayAllStudent*(theStudents);

//delete student id=1

System.***out***.println("\nDeleting student id=1");

session.createQuery("delete from Student where id = 1").executeUpdate();

session.getTransaction().commit();

// display all students

System.***out***.println("\nDisplay all students after delete id: 1");

session = factory.getCurrentSession();

session.beginTransaction();

List<Student> theStudentNew =

session.createQuery("from Student").getResultList();

*displayAllStudent*(theStudentNew);

session.getTransaction().commit();

System.***out***.println("Done!!!");

} **finally** {

factory.close();

}

}

**private** **static** **void** displayAllStudent(List<Student> theStudents) {

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

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

}

}

}

**Output**:

Display all students

Hibernate: select student0\_.id as id1\_0\_, student0\_.email as email2\_0\_, student0\_.first\_name as first\_na3\_0\_, student0\_.last\_name as last\_nam4\_0\_ from student student0\_

Student [id=1, firstName=Md. Ruhul Amin, lastName=Ruhul, email=ruhul@gmail.com]

Student [id=2, firstName=Md. Rezaul Islam, lastName=Reza, email=reza@gmail.com]

Student [id=3, firstName=Md. Ariful Islam, lastName=Arif, email=arif@gmail.com]

Student [id=4, firstName=Md. Rafiul Islam, lastName=Rafi, email=rafi@gmail.com]

Deleting student id=1

Hibernate: delete from student where id=1

Display all students after delete id: 1

Hibernate: select student0\_.id as id1\_0\_, student0\_.email as email2\_0\_, student0\_.first\_name as first\_na3\_0\_, student0\_.last\_name as last\_nam4\_0\_ from student student0\_

Student [id=2, firstName=Md. Rezaul Islam, lastName=Reza, email=reza@gmail.com]

Student [id=3, firstName=Md. Ariful Islam, lastName=Arif, email=arif@gmail.com]

Student [id=4, firstName=Md. Rafiul Islam, lastName=Rafi, email=rafi@gmail.com]

Done!!!

**Note**:

When we start a new operation in database, we must have to

1. Create a new session

session = factory.getCurrentSession();

1. Start a transection

session.beginTransaction();

21.2. Hibernate CRUD