

HIBERNATE ASSIGNMENT Day_1

Harshit Kushmakar | 16896

1. Create an entity Customer according to the given class diagram:

Write a hibernate program to perform the below:

a. DML (insert, update, delete) operations of Customer entity in database.

INSERT

```
package com.hibernate;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import java.time.LocalDate;

public class CreateCustomer {
    public static void main(String[] args) {
        //Create student entity object
        Customer customer = new Customer();
        customer.setCustomerID(16893);
        customer.setCustomerName("Aman verma");
        customer.setDateOfBirth(LocalDate.parse("2019-10-02"));
        customer.setMonthlyIncome(3656);
        customer.setProfession("Doctor");
        customer.setDesignation("surgeon ");
        customer.setCompanyName("AIIMS");
        // customer.setCustomerID(16896);
        // customer.setCustomerName("Harshit");
        // customer.setDateOfBirth(LocalDate.parse("2018-11-01"));
        // customer.setMonthlyIncome(36756);
        // customer.setProfession("Engineer");
        // customer.setDesignation("Assistant software");
        // customer.setCompanyName("Nucleus");
        //Create session factory object
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        //getting session object from session factory
        Session session = sessionFactory.openSession();
        //getting transaction object from session object
        session.beginTransaction();
        session.save(customer);
        System.out.println("Inserted Successfully");
        session.getTransaction().commit();
        sessionFactory.close();
    }
}
```

```

package com.hibernate;

import javax.persistence.*;
import java.io.Serializable;
import java.time.LocalDate;

@Entity
@Table(name="CUSTOMER_16896")
public class Customer implements Serializable {
    private static final long serialVersionUID = 8633415090390966715L;

    @Id
    @Column(name="customerID")
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int customerID ;

    @Column(name="customerName")
    private String customerName;

    @Column(name="dateOfBirth")
    private LocalDate dateOfBirth;

    @Column(name="monthlyIncome")
    private double monthlyIncome;

    @Column(name="profession")
    private String profession;

    @Column(name="designation")
    private String designation;

    @Column(name="companyName")
    private String companyName;

    public int getCustomerID() {
        return customerID;
    }

    public void setCustomerID(int customerID) {
        this.customerID = customerID;
    }

    public String getCustomerName() {
        return customerName;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public LocalDate getDateOfBirth() {
        return dateOfBirth;
    }

    public void setDateOfBirth(LocalDate dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }

    public double getMonthlyIncome() {

```

```

        return monthlyIncome;
    }

    public void setMonthlyIncome(double monthlyIncome) {
        this.monthlyIncome = monthlyIncome;
    }

    public String getProfession() {
        return profession;
    }

    public void setProfession(String profession) {
        this.profession = profession;
    }

    public String getDesignation() {
        return designation;
    }

    public void setDesignation(String designation) {
        this.designation = designation;
    }

    public String getCompanyName() {
        return companyName;
    }

    public void setCompanyName(String companyName) {
        this.companyName = companyName;
    }

    @Override
    public String toString() {
        return "Customer{" +
            "customerID=" + customerID +
            ", customerName=" + customerName +
            ", dateOfBirth=" + dateOfBirth +
            ", monthlyIncome=" + monthlyIncome +
            ", profession=" + profession + '\'' +
            ", designation=" + designation + '\'' +
            ", companyName=" + companyName + '\'' +
            '}';
    }

    public Customer() {}

    public Customer(int customerID, String customerName, LocalDate
dateOfBirth, double monthlyIncome, String profession, String designation,
String companyName) {
        this.customerID = customerID;
        this.customerName = customerName;
        this.dateOfBirth = dateOfBirth;
        this.monthlyIncome = monthlyIncome;
        this.profession = profession;
        this.designation = designation;
        this.companyName = companyName;
    }
}





```

```

package com.hibernate;
import java.util.Properties;import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.*; import org.hibernate.service.ServiceRegistry;
public class HibernateUtil {
    private static SessionFactory sessionFactory;
    public static SessionFactory getSessionFactory() {
        if (sessionFactory == null) {
            try {
                Configuration configuration = new Configuration();
                Properties settings = new Properties();
                settings.put(Environment.DRIVER,
"oracle.jdbc.driver.OracleDriver");
                settings.put(Environment.URL,
"jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c");
                settings.put(Environment.USER, "sh");
                settings.put(Environment.PASS, "sh");
                settings.put(Environment.DIALECT,
"org.hibernate.dialect.Oracle10gDialect");
                settings.put(Environment.SHOW_SQL, "true");
                settings.put(Environment.CURRENT_SESSION_CONTEXT_CLASS,
"thread");
                // settings.put(Environment.HBM2DDL_AUTO, "update");
                configuration.setProperties(settings);
                configuration.addAnnotatedClass(Customer.class);
                ServiceRegistry serviceRegistry = new
StandardServiceRegistryBuilder().applySettings(configuration.getProperties(
)).build();
                sessionFactory =
configuration.buildSessionFactory(serviceRegistry);
            } catch (Exception e) { e.printStackTrace(); }
        } return sessionFactory;
    } }

```

Query Result x

 | All Rows Fetched: 2 in 0.182 seconds

	CUSTOMERID	COMPANYNAME	CUSTOMERNAME	DATEOFBIRTH	DESIGNATION	MONTHLYINCOME	PROFESSION
1	18	AIIMS	Aman verma	02-10-19	surgeon	3656	Doctor
2	17	Nucleus	Harshit	01-11-18	Assistant software	36756	Engineer

UPDATE:

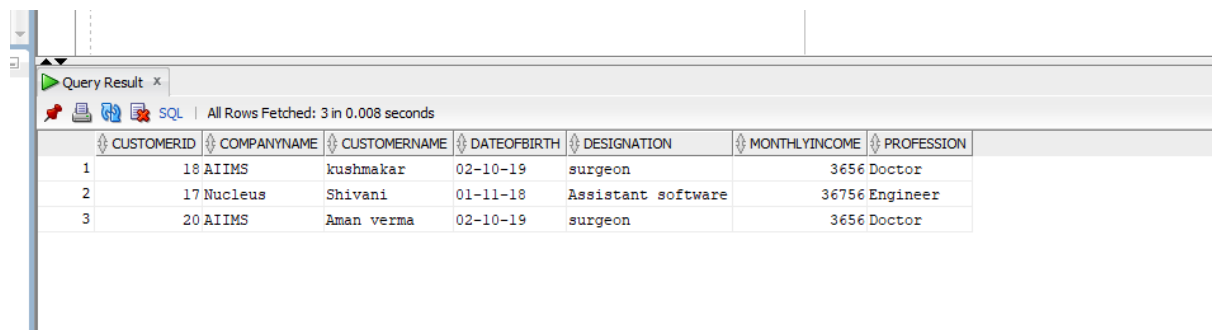
```
package com.hibernate;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

public class UpdateCustomer {
    public static void main(String[] args) {

        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        Session session = sessionFactory.openSession();
        session.beginTransaction();
        Customer customer = (Customer) session.get(Customer.class, 17);
        //select * from Student where ID=16896;
        customer.setCustomerName("Shivani"); //update where id=16896
        System.out.println("Updated Successfully");
        session.save(customer);
        session.getTransaction().commit();
        sessionFactory.close();
        customer.setCustomerName("shivani");

    }
}
```



Query Result x

All Rows Fetched: 3 in 0.008 seconds

CUSTOMERID	COMPANYNAME	CUSTOMERNAME	DATEOFBIRTH	DESIGNATION	MONTHLYINCOME	PROFESSION
1	18 AIIMS	kushmakar	02-10-19	surgeon	3656	Doctor
2	17 Nucleus	Shivani	01-11-18	Assistant software	36756	Engineer
3	20 AIIMS	Aman verma	02-10-19	surgeon	3656	Doctor

DELETE:

```
package com.hibernate;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

public class DeleteCustomer {

    public static void main(String[] args) {
        //Create session factory object
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        //getting session object from session factory
        Session session = sessionFactory.openSession();
        //getting transaction object from session object
```

```

        session.beginTransaction();
//        Customer customer = (Customer)session.load(Customer.class, 18);
        Customer customer = (Customer)session.load(Customer.class, 17);
//        Customer customer = (Customer)session.load(Customer.class, 20);
        session.delete(customer);
        System.out.println("Deleted Successfully");
        session.getTransaction().commit();
        sessionFactory.close();
    }
}

```

```

Apr 10, 2023 1:44:55 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.Oracle10gDialect
Hibernate: select customer0_.customerID as customerID1_0_0_, customer0_.companyName as companyName2_0_0_, custom
Deleted Successfully
Hibernate: delete from CUSTOMER_16896 where customerID=?
Apr 10, 2023 1:44:57 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolS
INFO: HHH10001008: Cleaning up connection pool [jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c]
Class transformation time: 0.050754177s for 3946 classes or 1.2862183730359858E-5s per class

```

OUTPUT:

Query Result x							
All Rows Fetched: 1 in 0.01 seconds							
	CUSTOMERID	COMPANYNAME	CUSTOMERNAME	DATEOFBIRTH	DESIGNATION	MONTHLYINCOME	PROFESSION
1	20	AIIMS	Aman verma	02-10-19	surgeon	3656	Doctor

b. Print details of one customer by giving the customerId.

```

package com.hibernate;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

public class RetrieveCustomer {
    public static void main(String[] args) {

        //Create session factory object
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        //getting session object from session factory
        Session session = sessionFactory.openSession();

        Customer customer = session.get(Customer.class, 18);
        System.out.println("CustomerID : "+customer.getCustomerId());
        System.out.println("CustomerName : "+customer.getCustomerName());
        System.out.println("CompanyName : "+customer.getCompanyName());
        System.out.println("DateOfBirth : "+customer.getDateOfBirth());
        System.out.println("MonthlyIncome : "+customer.getMonthlyIncome());
    }
}

```

```

        System.out.println("Designation : "+customer.getDesignation());
        System.out.println("Profession : "+customer.getProfession());

        sessionFactory.close();
    }
}

```

```

Apr 10, 2023 2:49:03 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.Oracle10gDialect
Hibernate: select customer0_.customerID as customerID1_0_0_, customer0_.companyName as companyName2_0_0_, customer0_.customerName
CustomerID : 20
CustomerName : Aman verma
CompanyName : AIIMS
DateOfBirth : 2019-10-02
MonthlyIncome : 3656.0
Designation : surgeon
Profession : Doctor
Apr 10, 2023 2:49:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c]

```

c. Print List of all Customers

```

package com.hibernate;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import java.util.ArrayList;
import java.util.List;

public class ReadCustomer {
    public static void main(String[] args) {
        //Create session factory object
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        //getting session object from session factory
        Session session = sessionFactory.openSession();
        //getting transaction object from session object
        session.beginTransaction();
        List<Customer> customers = new ArrayList<Customer>();
        for (Object oneObject : session.createQuery("FROM
Customer").getResultList()) {
            customers.add((Customer) oneObject);
        }
        // List<Student> students =
(List<Student>)session.createQuery("FROM Student").getResultList();
        //System.out.println(students); //if Student.toString() exists
        for(Customer customer : customers) {
            System.out.println("Customer ID:"+customer.getCustomerID()
+",Customer Name:"+customer.getCustomerName()+ ",Date of Birth:
"+customer.getDateOfBirth()+",Monthly
Income"+customer.getMonthlyIncome()+",profession"+customer.getProfession()+
"designation"+customer.getDesignation());
        }

        session.getTransaction().commit();
        sessionFactory.close();
    }
}

```

```

INFO: HHH10001005: Autocommit mode: false
Apr 10, 2023 3:20:27 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Apr 10, 2023 3:20:28 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.Oracle10gDialect
Hibernate: select customer0_.customerID as customerID1_0_, customer0_.companyName as companyName2_0_, customer0_.customerName as customerName3_0_,
customer0_.dateOfBirth as dateOfBirth4_0_, customer0_.designation as designation5_0_, customer0_.monthlyIncome as monthlyIncome6_0_, customer0_
.profession as profession7_0_ from CUSTOMER_16896 customer0_
Customer ID:20,Customer Name:Aman verma,Date of Birth: 2019-10-02,Monthly Income3656.0,professionDoctor,designationsurgeon
Apr 10, 2023 3:20:32 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c]

```

d. Print list of all customers who have the same designation and who belong to same company.

3. Create a Hibernate application to perform CRUD operations on the given entity.

```

package loan;

import com.hibernate.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

import java.time.LocalDate;

public class CreateLoanAgreement {
    public static void main(String[] args) {
        //Create entity object
        LoanAgreement loanAgreement = new LoanAgreement();
        loanAgreement.setLoanAgreementId(101);
        loanAgreement.setLoanAmount(140000);
        loanAgreement.setTenure(10);
        loanAgreement.setRoi(1000);
        loanAgreement.setEmiPerMonth(500);
        loanAgreement.setLoanDisbursalDate(LocalDate.ofEpochDay(2017-05-14));
        loanAgreement.setRepaymentFrequency(134000);
        //Create session factory object
        SessionFactory sessionFactory =
loan.HibernateUtil.getSessionFactory();
        //getting session object from session factory
        Session session = sessionFactory.openSession();
        //getting transaction object from session object
        session.beginTransaction();
        session.save(loanAgreement);
        System.out.println("Inserted Successfully");
        session.getTransaction().commit();
        sessionFactory.close();
    }
}

```

```

package loan;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.cfg.Environment;

```



```

import org.hibernate.service.ServiceRegistry;

import java.util.Properties;

public class HibernateUtil {
    private static SessionFactory sessionFactory;
    public static SessionFactory getSessionFactory() {
        if (sessionFactory == null) {
            try {
                Configuration configuration = new Configuration();
                Properties settings = new Properties();
                settings.put(Environment.DRIVER,
"oracle.jdbc.driver.OracleDriver");
                settings.put(Environment.URL,
"jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c");
                settings.put(Environment.USER, "sh");
                settings.put(Environment.PASS, "sh");
                settings.put(Environment.DIALECT,
"org.hibernate.dialect.Oracle10gDialect");
                settings.put(Environment.SHOW_SQL, "true");
                settings.put(Environment.CURRENT_SESSION_CONTEXT_CLASS,
"thread");
                settings.put(Environment.HBM2DDL_AUTO, "update");
                configuration.setProperties(settings);
                configuration.addAnnotatedClass(loan.LoanAgreement.class);

//configuration.addAnnotatedClass(CreateLoanAgreement.class);
//    configuration.addAnnotatedClass(DeleteCustomer.class);
//
configuration.addAnnotatedClass(UpdateCustomer.class);
//
configuration.addAnnotatedClass(RetrieveCustomer.class);
//
configuration.addAnnotatedClass(ReadCustomer.class);
                ServiceRegistry serviceRegistry = new
StandardServiceRegistryBuilder().applySettings(configuration.getProperties(
)).build();

                sessionFactory =
configuration.buildSessionFactory(serviceRegistry);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
        return sessionFactory;
    }
}

```

```

package loan;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import java.io.Serializable;
import java.time.LocalDate;

```

```

@Entity
@Table(name="LoanAgreement_16896")
public class LoanAgreement implements Serializable {
    private static final long serialVersionUID = 8633415090390966715L;

    @Id
    @Column(name="loanAgreementId")

    private int loanAgreementId ;

    @Column(name="loanAmount")
    private double loanAmount;

    @Column(name="tenure")
    private int tenure;

    @Column(name="roi")
    private double roi;

    @Column(name="emiPerMonth")
    private double emiPerMonth;

    @Column(name="loanDisbursalDate")
    private LocalDate loanDisbursalDate;

    @Column(name="repaymentFrequency")
    private int repaymentFrequency;

    public int getLoanAgreementId() {
        return loanAgreementId;
    }

    public void setLoanAgreementId(int loanAgreementId) {
        this.loanAgreementId = loanAgreementId;
    }

    public double getLoanAmount() {
        return loanAmount;
    }

    public void setLoanAmount(double loanAmount) {
        this.loanAmount = loanAmount;
    }

    public int getTenure() {
        return tenure;
    }

    public void setTenure(int tenure) {
        this.tenure = tenure;
    }

    public double getRoi() {
        return roi;
    }

    public void setRoi(double roi) {
        this.roi = roi;
    }
}

```

```

public double getEmiPerMonth() {
    return emiPerMonth;
}

public void setEmiPerMonth(double emiPerMonth) {
    this.emiPerMonth = emiPerMonth;
}

public LocalDate getLoanDisbursalDate() {
    return loanDisbursalDate;
}

public void setLoanDisbursalDate(LocalDate loanDisbursalDate) {
    this.loanDisbursalDate = loanDisbursalDate;
}

public int getRepaymentFrequency() {
    return repaymentFrequency;
}

public void setRepaymentFrequency(int repaymentFrequency) {
    this.repaymentFrequency = repaymentFrequency;
}

    public LoanAgreement(int loanAgreementId, double loanAmount, int
tenure, double roi, double emiPerMonth, LocalDate loanDisbursalDate, int
repaymentFrequency) {
        this.loanAgreementId = loanAgreementId;
        this.loanAmount = loanAmount;
        this.tenure = tenure;
        this.roi = roi;
        this.emiPerMonth = emiPerMonth;
        this.loanDisbursalDate = loanDisbursalDate;
        this.repaymentFrequency = repaymentFrequency;
    }




    public LoanAgreement() {








    }

    @Override
    public String toString() {
        return "LoanAgreement{" +
            "loanAgreementId=" + loanAgreementId +
            ", loanAmount=" + loanAmount +
            ", tenure=" + tenure +
            ", roi=" + roi +
            ", emiPerMonth=" + emiPerMonth +
            ", loanDisbursalDate=" + loanDisbursalDate +
            ", repaymentFrequency=" + repaymentFrequency +
            '}';
    }
}

```

Query Result x

   SQL | All Rows Fetched: 1 in 2.102 seconds

 LOANAGREEMENTID	 EMIPERMONTH	 LOANAMOUNT	 LOANDISBURSALDATE	 REPAYMENTFREQUENCY	 ROI	 TENURE	
1	101	500	140000	22-06-75	134000	1000	10

