**Hands on 4**

**Difference between JPA, Hibernate and Spring Data JPA**

**JPA (Java Persistence API):**

JPA is a specification that facilitates object-relational mapping to manage relational data in Java applications. It provides a platform to work directly with objects instead of using SQL statements.

**Hibernate Framework:**

Hibernate is an object-relational mapping solution for Java environments. Object-relational mapping or ORM is the programming technique to map application domain model objects to the relational database tables.

Hibernate provides a reference implementation of the Java Persistence API that makes it a great choice as an ORM tool with the benefits of loose coupling.

Hibernate is a Java-based ORM tool that provides a framework for mapping application domain objects to relational database tables and vice versa.

**Example:**The diagram below shows an Object-Relational Mapping between the Student Java class and the student's table in the database.

A diagram of hibernate framework

Description automatically generated

**Spring Data JPA:**

Spring Data is a part of the [Spring Framework](http://www.javaguides.net/p/spring-framework.html). The goal of Spring Data repository abstraction is to significantly reduce the amount of boilerplate code required to implement data access layers for various persistence stores.

Spring Data JPA is a library/framework that adds an extra layer of abstraction on the top of our JPA provider (like Hibernate).

Example:

**Hibernate**

/\* Method to CREATE an employee in the database \*/

   public Integer addEmployee(Employee employee){

      Session session = factory.openSession();

      Transaction tx = null;

      Integer employeeID = null;

      try {

         tx = session.beginTransaction();

         employeeID = (Integer) session.save(employee);

         tx.commit();

      } catch (HibernateException e) {

         if (tx != null) tx.rollback();

         e.printStackTrace();

} finally {

         session.close();

      }

      return employeeID;

   }

**Spring Data JPA**  
EmployeeRespository.java

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

EmployeeService.java

@Autowired

  private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

  employeeRepository.save(employee);

  }

​​​​​​​ 