Difference between JPA, Hibernate and Spring Data JPA

Java Persistence API (JPA)

- JSR 338 Specification for persisting, reading and managing data from Java objects

- Does not contain concrete implementation of the specification

- Hibernate is one of the implementation of JPA

Hibernate

- ORM Tool that implements JPA

Spring Data JPA

- Does not have JPA implementation, but reduces boiler plate code

- This is another level of abstraction over JPA implementation provider like Hibernate

- Manages transactions

Comparison Table:

Feature	JPA	Hibernate	Spring Data JPA
Type	Specification	Implementation	Abstraction Framework
Provides	Interfaces/Annotations	JPA Implementation + More	Repository Support, Query Simplification
Boilerplate	High	Medium	Very Low
Transactions	Needs manual handling	Needs manual handling	Managed automatically by Spring
Common Usage	Annotations, Interfaces	SessionFactory, HQL	JpaRepository,

Method-based queries |

Code Comparison:

Hibernate (Manual):

```
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 (Simplified):

EmployeeRepository.java

```
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {}
```

EmployeeService.java

```
@Autowired
private EmployeeRepository employeeRepository;

@Transactional
public void addEmployee(Employee employee) {
    employeeRepository.save(employee);
}
```