JPA

1. JDBC: So many connections, resultSets, Prepared statements, Lots of code.
   1. *PreparedStatement st =Conn.preparedStatement(“Update Student set user=?, desc=?”);*

St.setString(1, Student.getUser());

St.setString(2, Student.getDesc());

1. Spring JDBC: JDBC template
   1. *JdbcTemplate.update(“Update Student set user =?, desc=?”, Student.getUser(), Student.getDesc());*
   2. Implemented RowMapper

*JdbcTemplate.queryForObject(“Select \* from Student where id=?”, new Object[]{id}, new StudentMapper());*

**Spring JPA**

EntityManagerFactory emf = Persistance.CreateEntity(“unit1”); // “unit1” defines connections in xml

EntityManager em = emf.createEntityManager();

* *Fetch student with id 2*

Student s1 = em.find(Student.class, 2)

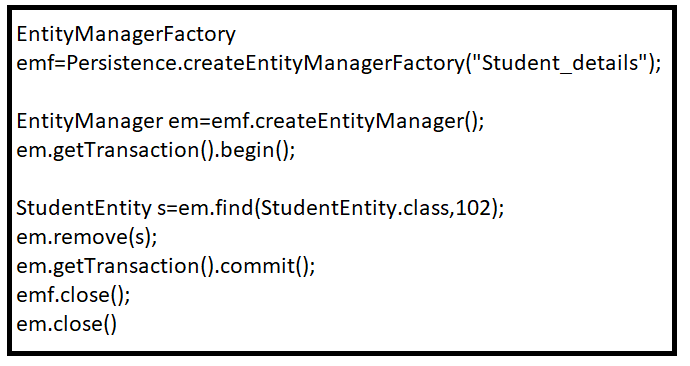
* *SAVE: To change data in database we have to use Transactions*

Em.getTransaction().begin(); // Start transaction

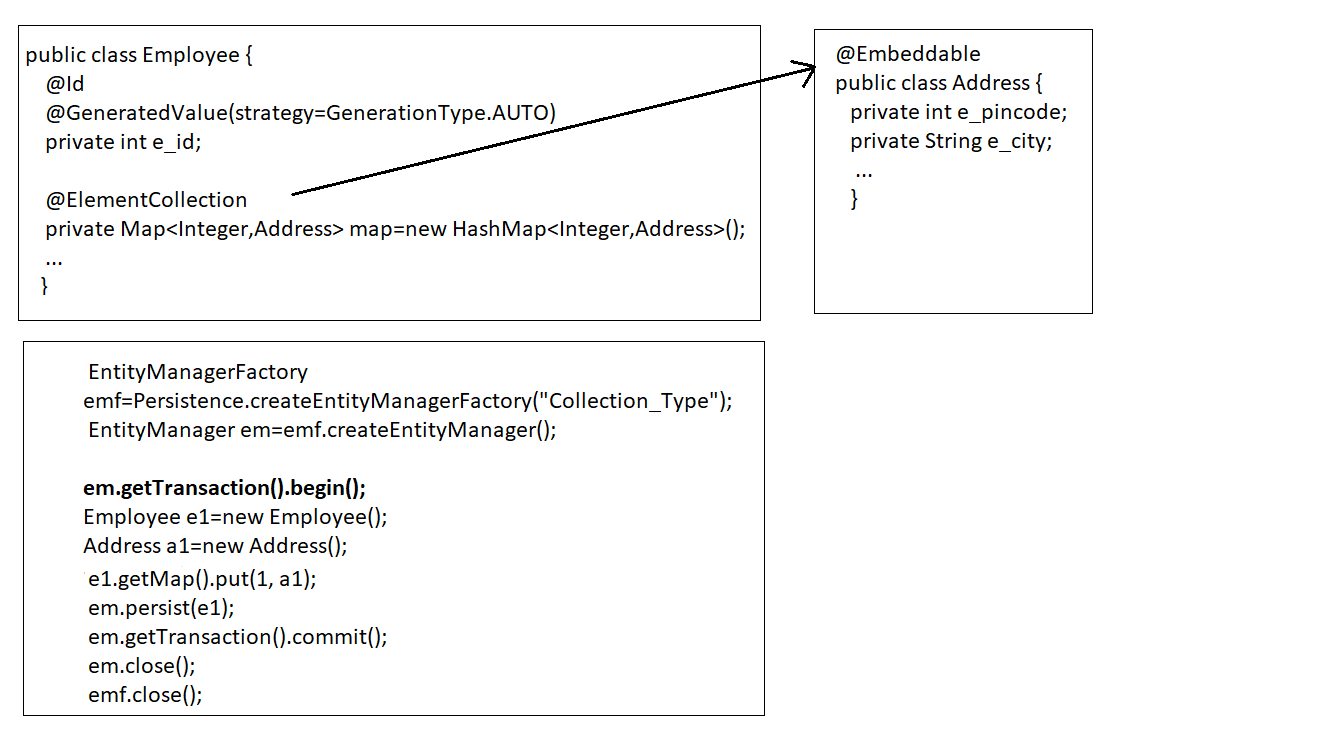
em.persist(s2); // Save student and s2 is tracked in Persistence context

em.getTransaction().commit();

Instead of using transactions for every method use @Transactional at class level.



* Mapping:

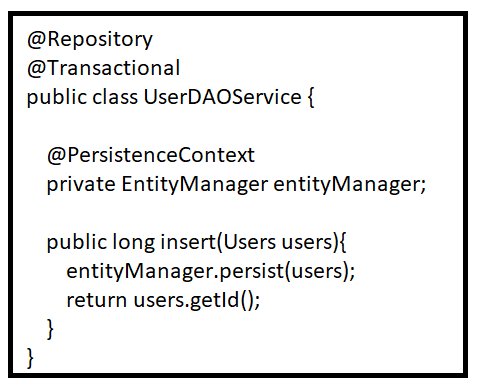


**Spring boot JPA**

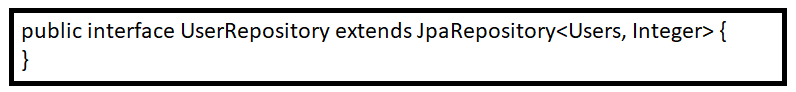
**@**Transactional – Transactions for every method

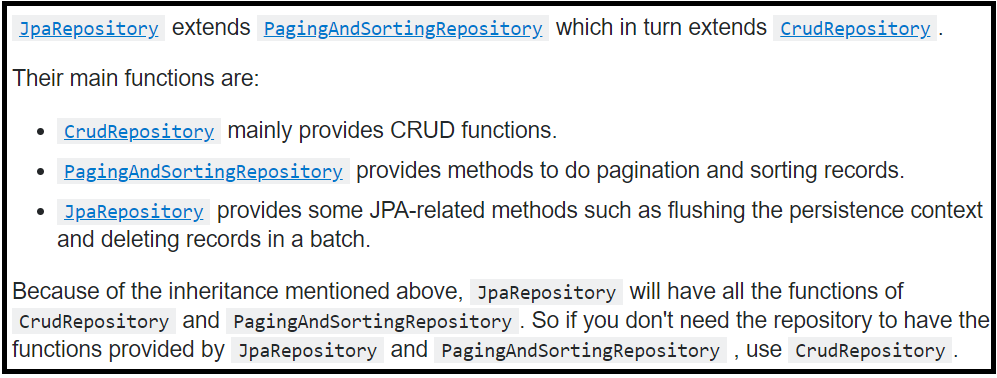
@PersistanceContext - Track Persistence context object

Example :

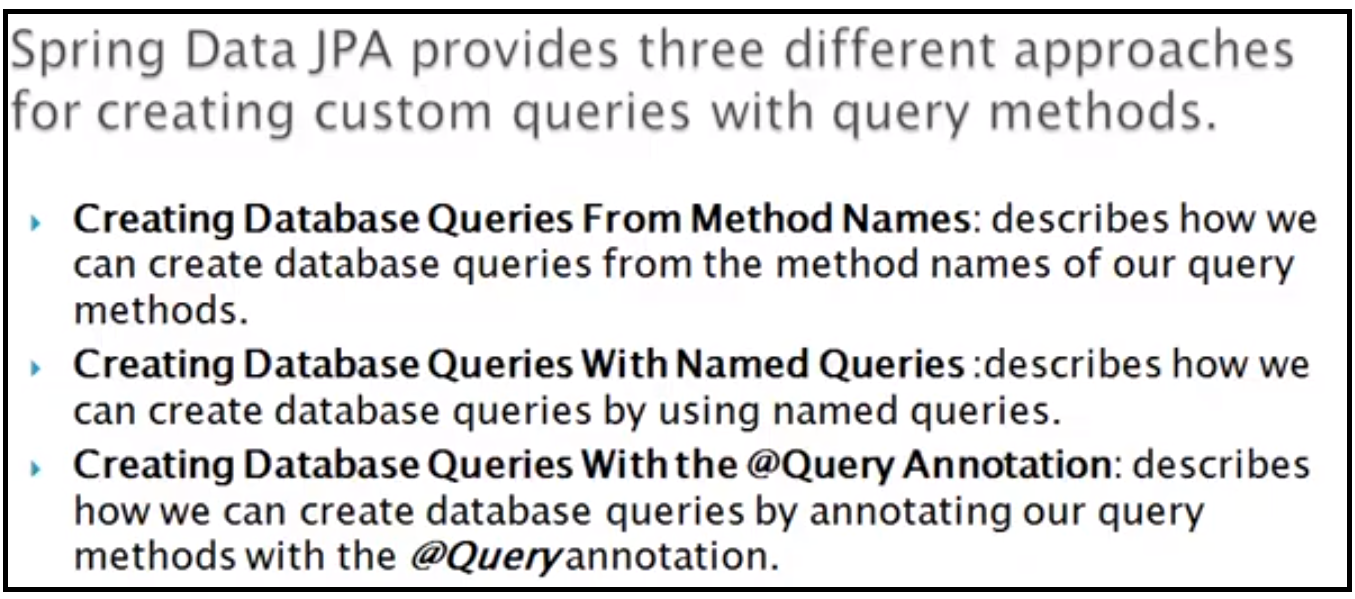


**Spring Data JPA**





* Spring data provides the abstract repositories that are implemented at run time by the spring container and perform CRUD operations.



* If there is a collision between **@NamedQuery** and **@Query**, the query specified using @Query is executed



* There are two types of named queries, JPQL named queries and native SQL named queries.
  + @NamedQuery
    - “SELECT \* FROM person p WHRE p.lastName = ?1”
  + @NamedNativeQuery
    - “SELECT \* FROM person\_table WHRE last\_name =?1”
* To Specify sql query in @Query, then use :
  + @Query(value=”select \* from employee where email = ?1**”, nativeQuery= true**)
* @Query has two formats:
  + *@Query(value=”SELECT \* FROM PERSON p WHERE p.name =?1)*

*Person findByName(String name);*

* + *@Query(value=”SELECT \* FROM PERSON p WHERE p.name : name)*

*Person findByName(@Param(“name”) String name);*

**Note:**

* JPA is a Specification and needs to be implemented.
* By default the Application Server provides the implementation of JPA as it is J2EE standards.
* Hibernate, Ibatis, topLink are some implementations.