2. Explain the need and benefit of Criteria Query

Criteria Query

- A Criteria Query is a type-safe, dynamic, and programmatically constructed query API provided by JPA.
- It is part of the JPA Criteria API, which allows the creation of queries without writing strings (as in JPQL/HQL).
- It is especially useful when building complex or conditional queries in Java code.

Benefits of Criteria Query

- **Type Safety:** Queries are constructed using Java objects and types.
- **Dynamic Query Generation**: Queries can be built conditionally using logic in Java code.
- **IDE Support**: Full auto-completion, refactoring, and validation from modern IDEs.
- **No Hardcoded Strings**: Eliminates syntax errors in JPQL/HQL strings.

Key Components of Criteria API

1. **CriteriaBuilder**
    - Used to construct different parts of the query (SELECT, WHERE, ORDER BY, etc.).
    - Obtained from EntityManager.getCriteriaBuilder().
2. **CriteriaQuery<T>**
    - Represents the actual query structure.
    - Defines result type and query operations like `select`, `where`, `groupBy`.

3. **Root<T>**

   - Represents the root entity in the FROM clause**.**
   - Used to refer to entity attributes (e.g., root.get("name")).

4. **TypedQuery<T>**

   - Created from `EntityManager.createQuery(criteriaQuery).`

Criteria Query Example

```
@Autowired
private EntityManager entityManager;
public List<Student> findStudentsByDepartment(String dept) {
   CriteriaBuilder cb = entityManager.getCriteriaBuilder();
   CriteriaQuery<Student> cq = cb.createQuery(Student.class);
   Root<Student> root = cq.from(Student.class);
   // Predicate: WHERE department = :dept
   cq.select(root).where(cb.equal(root.get("department"), dept));
   TypedQuery<Student> query = entityManager.createQuery(cq);
   return query.getResultList();
}
```

Explanation of Sample Code

- CriteriaBuilder creates expressions, predicates, and queries**.**
- CriteriaQuery<Student> defines a query that returns Student bjects.
- Root<Student> refers to the entity being queried (like FROM Student s in JPQL).