1. ✅ **Legacy Hibernate Criteria API** (pre-JPA, available till Hibernate 5.x).
2. ✅ **DTO Projection using Emp\_Dept** class.

**🔧 First, enable legacy criteria support in Hibernate 5.x:**

If you use Hibernate 5.2+, make sure hibernate-core supports legacy org.hibernate.Criteria. It’s deprecated but still works in 5.x.

**🧱 Legacy Hibernate Criteria Examples**

**✅ 1. e\_name and salary of all employees**

Criteria crit = session.createCriteria(Emp.class)

.setProjection(Projections.projectionList()

.add(Projections.property("e\_name"))

.add(Projections.property("salary")));

List<Object[]> result = crit.list();

for (Object[] row : result) {

System.out.println("Name: " + row[0] + ", Salary: " + row[1]);

}

**✅ 2. Top 3 employees by salary**

Criteria crit = session.createCriteria(Emp.class);

crit.addOrder(Order.desc("salary"));

crit.setMaxResults(3);

List<Emp> topEmployees = crit.list();

**✅ 3. Aggregate functions**

Criteria crit = session.createCriteria(Emp.class)

.setProjection(Projections.projectionList()

.add(Projections.avg("salary"))

.add(Projections.sum("salary"))

.add(Projections.max("salary"))

.add(Projections.count("e\_name")));

Object[] result = (Object[]) crit.uniqueResult();

System.out.println("Avg: " + result[0] + ", Sum: " + result[1] +

", Max: " + result[2] + ", Count: " + result[3]);

**✅ 4. Name contains 'a' and salary > value**

Criteria crit = session.createCriteria(Emp.class);

crit.add(Restrictions.like("e\_name", "%a%"));

crit.add(Restrictions.gt("salary", 25000.0));

List<Emp> result = crit.list();

**✅ 5. Between two dates**

SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

Date d1 = sdf.parse("01-01-2003");

Date d2 = sdf.parse("31-12-2003");

Criteria crit = session.createCriteria(Emp.class);

crit.add(Restrictions.between("join\_date", d1, d2));

List<Emp> result = crit.list();

**🎯 DTO Projection using Emp\_Dept**

Suppose you want to project data directly into your custom Emp\_Dept class:

**HQL with DTO Constructor:**

String hql = "select new com.bean.Emp\_Dept(e.d\_code, e) from Emp e";

List<Emp\_Dept> results = session.createQuery(hql, Emp\_Dept.class).getResultList();

for (Emp\_Dept ed : results) {

System.out.println(ed.getE\_name() + " | " + ed.getD\_name() + " | " + ed.getSalary());

}

**🛑 Limitation in Legacy Criteria for DTO**

Legacy Criteria **does not support constructor expressions** directly. So for Emp\_Dept, you must:

* Either use HQL with new as shown above.
* Or manually map from List<Object[]> to Emp\_Dept.

**Example:**

Criteria crit = session.createCriteria(Emp.class, "e")

.createAlias("e.d\_code", "d")

.setProjection(Projections.projectionList()

.add(Projections.property("e.e\_code"))

.add(Projections.property("e.e\_name"))

.add(Projections.property("e.salary"))

.add(Projections.property("e.join\_date"))

.add(Projections.property("d.d\_code"))

.add(Projections.property("d.d\_name"))

);

List<Object[]> rows = crit.list();

List<Emp\_Dept> edList = new ArrayList<>();

for (Object[] row : rows) {

Dept dept = new Dept((int) row[4], (String) row[5]);

Emp emp = new Emp((int) row[0], (String) row[1], (double) row[2], (Date) row[3]);

edList.add(new Emp\_Dept(dept, emp));