List<DropDownResA> resList = **new** ArrayList<DropDownResA>();

**try** {

String sumInsured = StringUtils.*isBlank*(req.getSumInsured()) ? "0" : req.getSumInsured() ;

// Criteria

CriteriaBuilder cb = em.getCriteriaBuilder();

CriteriaQuery<UnderWritterRelation> query = cb.createQuery(UnderWritterRelation.**class**);

List<UnderWritterRelation> list = **new** ArrayList<UnderWritterRelation>();

// Find All

Root<ClaimLoginMaster> l = query.from(ClaimLoginMaster.**class**);

Root<UnderWritterRelation> uw = query.from(UnderWritterRelation.**class**);

// Select

query.select(uw );

// Order By

List<Order> orderList = **new** ArrayList<Order>();

orderList.add(cb.asc(uw.get("loginId")));

// Where

javax.persistence.criteria.Predicate n1 = cb.equal(uw.get("status"), "Y");

javax.persistence.criteria.Predicate n2 = cb.equal(uw.get("classId"), req.getClassId());

javax.persistence.criteria.Predicate n3 = cb.lessThanOrEqualTo(uw.get("suminsuredStartLimit"), sumInsured);

javax.persistence.criteria.Predicate n4 = cb.greaterThanOrEqualTo(uw.get("suminsuredEndLimit"), sumInsured);

javax.persistence.criteria.Predicate n5 = cb.like(l.get("branchCode"),"%" + req.getBranchCode() + "%" ) ;

javax.persistence.criteria.Predicate n6 = cb.equal(l.get("companyId"), req.getInsCompanyId());

javax.persistence.criteria.Predicate n7 = cb.equal(l.get("agencyCode"), uw.get("uwCode"));

javax.persistence.criteria.Predicate n8 = cb.equal(l.get("loginId"), uw.get("loginId"));

query.where(n1,n2,n3,n4,n5,n6,n7,n8).orderBy(orderList);

// Get Result

TypedQuery<UnderWritterRelation> result = em.createQuery(query);

list = result.getResultList();

**for** (UnderWritterRelation data : list ) {

DropDownResA res = **new** DropDownResA();

res.setCode(data.getUwCode());

res.setCodeDesc(data.getLoginId());

resList.add(res);

}

} **catch** (Exception e) {

e.printStackTrace();

log.info("Log Details" + e.getMessage());

**return** **null**;

}

**return** resList;

}