**N+1 Problem in Hibernate and its solution-Precise-2022**

N+1 problem is a performance issue in Object Relational Mapping that fires multiple select queries (N+1 to be exact, where N = number of records in table) in database for a single select query at application layer.

**N+1 Resolution**

Hibernate & Spring Data JPA provide mechanism to solve the N+1 ORM issue.

At SQL level, what ORM needs to achieve to avoid N+1 is to fire a query that joins the two tables and get the combined results in single query.

**Plain SQL**

**SELECT p.id, p.name FROM Parent p LEFT OUTER JOIN child c ON p.id = c.parent\_id**

**SELECT \* FROM t\_mobile\_vendor vendor LEFT OUTER JOIN t\_phone\_model model ON model.vendor\_id=vendor.vendor\_id**

**1. Spring Data JPA Approach**

If we are using Spring Data JPA, then we have two options to achieve this - using EntityGraph or using select query with fetch join.

public interface UserRepository extends CrudRepository<User, Long> {

List<User> findAllBy();

**@Query("SELECT u FROM User u LEFT JOIN FETCH u.roles")**

List<User> findWithoutNPlusOne();

**@EntityGraph(attributePaths = {"roles"})**

List<User> findAll();

}

**2. Hibernate Approach**

If its pure Hibernate, then the following solutions will work.

**Using HQL Query**

"from User u join fetch u.roles roles roles"

**Using Hibernate Criteria API**

Criteria criteria = session.createCriteria(User.class);

criteria.setFetchMode("roles", FetchMode.EAGER);