1.jpa内置函数接口

| **Keyword** | **Sample** | **JPQL snippet** |
| --- | --- | --- |
| And | findByLastnameAndFirstname | … where x.lastname = ?1 and x.firstname = ?2 |
| Or | findByLastnameOrFirstname | … where x.lastname = ?1 or x.firstname = ?2 |
| Is,Equals | findByFirstname,findByFirstnameIs,findByFirstnameEquals | … where x.firstname = ?1 |
| Between | findByStartDateBetween | … where x.startDate between ?1 and ?2 |
| LessThan | findByAgeLessThan | … where x.age < ?1 |
| LessThanEqual | findByAgeLessThanEqual | … where x.age <= ?1 |
| GreaterThan | findByAgeGreaterThan | … where x.age > ?1 |
| GreaterThanEqual | findByAgeGreaterThanEqual | … where x.age >= ?1 |
| After | findByStartDateAfter | … where x.startDate > ?1 |
| Before | findByStartDateBefore | … where x.startDate < ?1 |
| IsNull | findByAgeIsNull | … where x.age is null |
| IsNotNull,NotNull | findByAge(Is)NotNull | … where x.age not null |
| Like | findByFirstnameLike | … where x.firstname like ?1 |
| NotLike | findByFirstnameNotLike | … where x.firstname not like ?1 |
| StartingWith | findByFirstnameStartingWith | … where x.firstname like ?1(parameter bound with appended %) |
| EndingWith | findByFirstnameEndingWith | … where x.firstname like ?1(parameter bound with prepended %) |
| Containing | findByFirstnameContaining | … where x.firstname like ?1(parameter bound wrapped in %) |
| OrderBy | findByAgeOrderByLastnameDesc | … where x.age = ?1 order by x.lastname desc |
| Not | findByLastnameNot | … where x.lastname <> ?1 |
| In | findByAgeIn(Collection<Age> ages) | … where x.age in ?1 |
| NotIn | findByAgeNotIn(Collection<Age> ages) | … where x.age not in ?1 |
| True | findByActiveTrue() | … where x.active = true |
| False | findByActiveFalse() | … where x.active = false |
| IgnoreCase | findByFirstnameIgnoreCase | … where UPPER(x.firstame) = UPPER(?1) |

2. Using @Query

*Example Declare query at the query method using @Query*

public interface UserRepository extends JpaRepository<User, Long> {

@Query("select u from User u where u.emailAddress = ?1")

User findByEmailAddress(String emailAddress);

}

3. 修改查询

@Modifying

@Query("update User u set u.firstname = ?1 where u.lastname = ?2")

int setFixedFirstnameFor(String firstname, String lastname);

##### 4. 派生删除查询

@Modifying

@Transactional

@Query("delete from User u where user.role.id = ?1")

void deleteInBulkByRoleId(long roleId);

*//JPA 执行update/delete query 需要加上事务*@Transactional

经验之谈，表名最好使用Abb格式命名，否则会出现respository无法创建成功的情况。

5.Pageable的使用

可以在使用时添加PageableDefault注解，改变分页参数：

Size：页面大小

Sort：排序依照物

Direction：排序规则

@GetMapping("/")

public void index(@PageableDefault(size = 10, sort = "id", direction = Sort.Direction.DESC) Pageable pageable) {

List<Demo> result = demoRespository.findByName("glq", pageable).getContent();

for (Demo demo : result

) {

System.out.println(demo.toString());

}

System.out.println(pageable.toString());

也可以在配置文件中修改页面大小，但是优先级小于方法内修改。

**data:  
 web:  
 pageable:  
 default-page-size:** 30