|  |  |  |
| --- | --- | --- |
| **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) |

**The following table shows Supported keywords inside method names**

## Spring Data JPA Method Naming Conventions for Query Example

Now let us see one example how method naming conventions and strategies play a role in spring data JPA.

#### Database script (MySQL)

|  |
| --- |
| **CREATE** **TABLE** STUDENT (         STUDENTID **INT** **NOT** **NULL** AUTO\_INCREMENT,         FIRSTNAME **VARCHAR**(20) **DEFAULT** **NULL**,         LASTNAME **VARCHAR**(20) **DEFAULT** **NULL**,         MARKS **INT**(20) **DEFAULT** **NULL**,  **PRIMARY** **KEY** (STUDENTID)  );    **INSERT** **INTO** STUDENT (FIRSTNAME,LASTNAME,MARKS) **VALUES** ('Manu','Manjunatha', 100);  **INSERT** **INTO** STUDENT (FIRSTNAME,LASTNAME,MARKS) **VALUES**('Advith','Tyagraj', 100);  **INSERT** **INTO** STUDENT (FIRSTNAME,LASTNAME,MARKS) **VALUES**('Likitha','Tyagraj', 98);    **COMMIT**; |