 @Query("SELECT p FROM Person p WHERE LOWER(p.lastName) = LOWER(:lastName)")

    public List<Person> find(@Param("lastName") String lastName);

**Springdatajpacustomqueries**

<https://attacomsian.com/blog/spring-data-jpa-query-annotation>

@Query("SELECT n FROM Note n WHERE n.featured = true")

List<Note> findByActiveNotes();

// SELECT clause is optional - it can be omitted

@Query("FROM Note n WHERE n.featured = true")

List<Note> findByActiveNotes();

@Query(value = "SELECT \* FROM Notes n WHERE n.featured = 1",

nativeQuery = true)

List<Note> findByFeaturedNotesNative();

@Query("SELECT n FROM Note n WHERE n.title = ?1")

List<Note> findByTitlePositionalBind(String title);

// multiple bind parameters

@Query("SELECT n FROM Note n WHERE n.title = ?1 AND n.featured = ?2")

List<Note> findByTitleAndFeaturedPositionalBind(String title, boolean featured);

@Query("SELECT n FROM Note n WHERE n.title = :title")

List<Note> findByTitleNamedBind(@Param("title") String title);

// multiple named parameters

@Query("SELECT n FROM Note n WHERE n.title = :title AND n.featured = :featured")

List<Note> findByTitleAndFeaturedNamedBind(@Param("featured") boolean featured,

@Param("title") String title);

// And / Or

@Query("SELECT n FROM Note n WHERE n.title = ?1 AND n.featured = ?2 OR n.priority = ?3")

List<Note> findByTitleAndFeaturedOrPriority(String title, boolean featured, int priority);

// Equality

@Query("SELECT n FROM Note n WHERE n.title = ?1")

List<Note> findByTitle(String title);

// Ignore Case

@Query("SELECT n FROM Note n WHERE lower(n.title) = lower(?1) ")

List<Note> findByTitleIgnoreCase(String title);

// Not Equal

@Query("SELECT n FROM Note n WHERE n.title <> ?1")

List<Note> findByTitleNotEqual(String title);

// Like / Contains / Starts With / Ends With

@Query("SELECT n FROM Note n WHERE n.title LIKE ?1")

List<Note> findByTitleLike(String pattern);

// Less Than

@Query("SELECT n FROM Note n WHERE n.priority < ?1")

List<Note> findByPriorityLessThan(int priority);

// Greater Than

@Query("SELECT n FROM Note n WHERE n.priority > ?1")

List<Note> findByPriorityGreaterThan(int priority);

// Between

@Query("SELECT n FROM Note n WHERE n.priority BETWEEN ?1 AND ?2")

List<Note> findByPriorityBetween(int start, int end);

// Before

@Query("SELECT n FROM Note n WHERE n.created < ?1")

List<Note> findByCreatedBefore(Date before);

// After

@Query("SELECT n FROM Note n WHERE n.created > ?1")

List<Note> findByCreatedAfter(Date before);

// Null

@Query("SELECT n FROM Note n WHERE n.title IS NULL")

List<Note> findByTitleIsNull();

// Not Null

@Query("SELECT n FROM Note n WHERE n.title IS NOT NULL")

List<Note> findByTitleIsNotNull();

// In

@Query("SELECT n FROM Note n WHERE n.priority IN ?1")

List<Note> findByPriorityIn(Set<Integer> priorities);

// sort notes in ascending order

@Query("SELECT n FROM Note n WHERE n.title = ?1 ORDER BY n.priority ASC")

List<Note> findByTitleOrderByPriorityAsc(String title);

// sort notes in descending order

@Query("SELECT n FROM Note n WHERE n.featured = ?1 ORDER BY n.created DESC")

List<Note> findByFeaturedOrderByCreatedDesc(boolean featured);