



CS544 EA

Applications

Spring Data: Finder Method Features

List of Keywords

And	After	Containing
Or	Before	OrderBy
Is, Equals	IsNull	Not
Between	IsNotNull, NotNull	In
LessThan	Like	NotIn
LessThanEqual	NotLike	True
GreaterThan	StartingWith	False
GreaterThanEqual	EndingWith	IgnoreCase (AllIgnoreCase)

Examples:

```
findByFirstname, findByFirstnameIs, findByFirstnameEquals // all the same
findByStartDateBetween // expects two parameters
findByAgeLessThanEqual
findByAgeIsNotNull, findByAgeNotNull // same
findByFirstnameLike // first parameter matched as Like (including %'s)
findByAgeIn(Collection<Age> ages) // or subclass of collection
```

IgnoreCase

- You can ignore case for a single property

```
List<Person> findByFirstNameAndLastnameIgnoreCase(String firstname, String lastname);
```

Only LastName is
case insensitive

- Or ignore case for all properties

```
List<Person> findByLastnameAndFirstnameAllIgnoreCase(String lastname, String firstname);
```

PagingAndSortingRepository

- The PagingAndSortingRepository interface adds methods to sort and paginate entities
 - findAll(Pageable pageable), findAll(Sort sort)
- Enables you to use the orderBy keyword

```
public List<Student> findByLastNameOrderByLastNameAsc(String lastName);  
public List<Student> findByLastNameOrderByLastNameDesc(String lastName);
```

- You can request a page using PageRequest

```
PageRequest pageRequest = new PageRequest(0, 10);  
Page<Student> page = studentDao.findAll(pageRequest);  
Slice<Student> slice = studentDao.findByFirstName("Lisa", pageRequest);
```

Difference Between Page and Slice

- Page extends Slice
 - Slice only knows if there is more
 - Not how much more
- Page executes a select count to find out

Limiting Results

- The keywords first and top will limit the amount of rows that will be returned
 - Optional numeric value can be added

```
User findFirstByOrderByLastnameAsc();  
User findTopByOrderByAgeDesc();  
Page<User> queryFirst10ByLastname(String lastname, Pageable pageable);  
Slice<User> findTop3ByLastname(String lastname, Pageable pageable);  
List<User> findFirst10ByLastname(String lastname, Sort sort);  
List<User> findTop10ByLastname(String lastname, Pageable pageable);
```

- If pagination or slicing is applied it is within the limited result