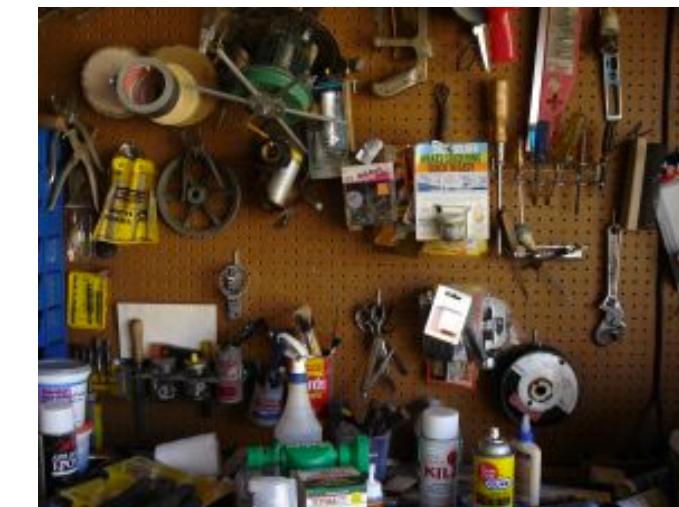
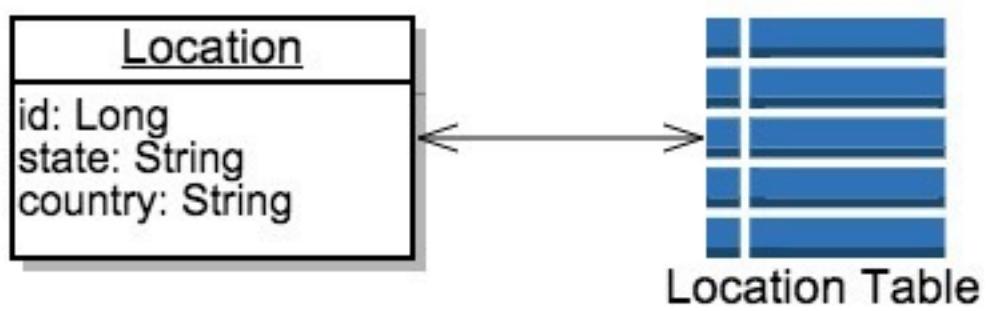
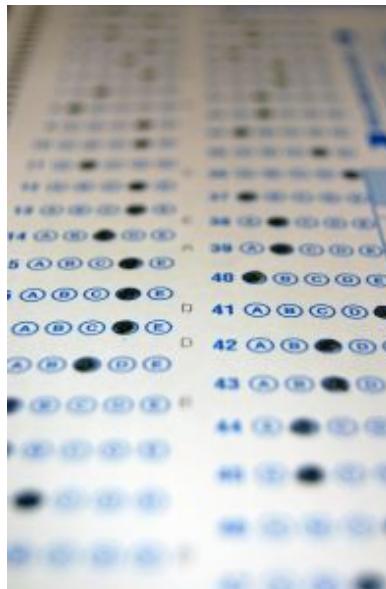


# Advantages of Using a Query DSL

- Utilize your work spent on creating your JPA entities
- Less code, less to maintain



- Check your queries at startup rather than runtime



# DSL - Domain Specific Language

A domain specific language (DSL) is a customized extension of a software programming language that addresses a specific business or domain.

# Query Method Syntax Basics

## Query Methods

- Query parser will match the following:
  - find..By, query..By, read..By, count..By, get..By
- Criteria uses JPA entity attribute names
- Multiple criteria combined with ["And", "Or"]

```
public interface LocationJpaRepository extends JpaRepository<Location, Long>{  
    findByStateLike(String stateName);  
}
```

## SQL

```
select * from Location l where l.state like ?
```

## JPQL

```
select l from Location l  
where l.state like :state
```



# Query Method Return Types

```
public interface LocationJpaRepository extends JpaRepository<Location, Long>{  
    Location findFirstByState(String stateName);  
    List<Location> findByStateLike(String stateName);  
    Long countByStateLike(String stateName);  
}
```

# Query Method Learning Instructions

an AIA or other numbers in  
x film and higher numbers in  
batter speed, the length of  
exposure film. 6. the largest  
be used. 6. Slang: a stimula-  
tamine or amphetamine. 7.  
has auto's ability, per-  
sonal people are more  
or prospects. 8. at full  
speed possible. 9. to the  
with great rapidity. He  
speed, a operating at  
or producing at an  
ive level up to par; a  
11. to promote the  
etc.; further, forward,  
ups, course, way, etc.)  
**speed-star** (spēd'star), *n.* a person who travels at  
high speed. [1915-20; SPEED + -STER]

**speed' trap'**, a section of a road where hidden police,  
radar, etc., carefully check the speed of motorists and  
strictly enforce traffic regulations; sometimes character-  
ized by hard-to-see signals, hidden traffic signs, etc.  
[1920-25]

**speed-up** (spēd'up'), *n.* 1. an increasing of speed. 2.  
an imposed increase in the rate of production of a  
worker without a corresponding increase in the rate of  
pay. [1920-25; *n. use of v. phrase speed up*]

**speed-walk** (spēd'wôk'), *n.* an endless conveyor belt  
moving walk, or the like used to transport standing per-  
sons from place to place. [SPEED + WALK]

**speed-way** (spēd'wā'), *n.* 1. a road or course for  
driving, motoring, or the like, or on which more than  
ordinary speed is allowed. 2. a track on which  
motorcycle races are held. —*speed-*

**spell-bind'er** (spel'bīn'dər), *n.* a person or thing that  
spellbinds, esp. a powerful speaker who can captivate an  
audience. [1885-90, Amer.; SPELLBIND + -ER<sup>1</sup>]

**spell-bound** (spel'bound'), *adj.* bound by or as if by a  
spell; enchanted, entranced, or fascinated: a spellbound  
audience. [1790-1800; SPELL<sup>2</sup> + -BOUND<sup>1</sup>]

**spell' check'er**, a computer program for checking  
the spelling of words in an electronic document. Also,  
**spell'ing check'er**. [1980-85]

**spell-down** (spel'doun'), *n.* a spelling competition  
in which all the contestants standing and that  
but one, the winner, have been required to  
e to a specified number of misspellings.  
er.; from phrase *spell down* to outspell oth-  
ng match]

**spell'er**, *n.* 1. a person who spells words. 2.  
**spell'ing book'**. an elementary textbook or  
ach spelling. [1400-50; late ME; see SPELL<sup>1</sup>,

**spell'ar** (spel'ər di vī'dər), *n.* a reference  
ts words in alphabetical order to show spell-  
abification.

**spell'ing**, *n.* 1. the

# Keyword: And and Or

<b>Uses</b>	<p><b>Combines multiple criteria query filters together using a conditional And or Or</b></p>
<b>Keyword Example</b>	<p><b>findByStateAndCountry("CA", "USA");</b> <b>findByStateOrState("CA", "AZ");</b></p>
<b>JPQL Example</b>	<p><b>... where a.state = ?1 <i>and</i> a.country = ?2</b> <b>... where a.state = ?1 <i>or</i> a.state = ?2</b></p>

# Keyword: Equals, Is and Not

<b>Uses</b>	<p><b><i>The default '=' when comparing the criteria with the filter value. Use Not when wanting to compare not equals</i></b></p>
<b>Keyword Example</b>	<p><b><i>findByState("CA");</i></b> <b><i>findByStateIs("CA");</i></b> <b><i>findByStateEquals("CA");</i></b> <b><i>findByStateNot("CA");</i></b></p>
<b>JPQL Example</b>	<p><b><i>... where a.state = ?1</i></b> <b><i>... where a.state = ?1</i></b> <b><i>... where a.state = ?1</i></b> <b><i>... where a.state &lt;&gt; ?1</i></b></p>

# Keyword: Like and NotLike

<b>Uses</b>	<p><b><i>Useful when trying to match, or not match, a portion of the criteria filter value</i></b></p>
<b>Keyword Example</b>	<p><b><code>findByState<b>Like</b>("Cali%");</code></b> <b><code>findByState<b>NotLike</b>("Al%");</code></b></p>
<b>JPQL Example</b>	<p><b><i>... where a.state <b>like</b> ?1</i></b> <b><i>... where a.state <b>not like</b> ?1</i></b></p>

# Keyword: StartingWith, EndingWith and Containing

<b>Uses</b>	<p><b>Similar to the “Like” keyword except the % is automatically added to the filter value</b></p>
<b>Keyword Example</b>	<p><b>findByState<b>StartingWith</b>("Al"); //Al%</b> <b>findByState<b>EndingWith</b>("ia"); //%ia</b> <b>findByState<b>Containing</b>("in"); //%in%</b></p>
<b>JPQL Example</b>	<p><b>... where a.state <b>like</b> ?1</b> <b>... where a.state <b>like</b> ?1</b> <b>... where a.state <b>like</b> ?1</b></p>

# Keyword: LessThan(Equal) and GreaterThan(Equal)

<b>Uses</b>	<p><b><i>When you need to perform a &lt;, &lt;=, &gt;, or &gt;= comparison with number data types</i></b></p>
<b>Keyword Example</b>	<p><b><i>findByPriceLessThan(20);</i></b> <b><i>findByPriceLessThanEqual(20);</i></b> <b><i>findByPriceGreaterThan(20);</i></b> <b><i>findByPriceGreaterThanOrEqual(20);</i></b></p>
<b>JPQL Example</b>	<p><b><i>... where a.price &lt; ?1</i></b> <b><i>... where a.price &lt;= ?1</i></b> <b><i>... where a.price &gt; ?1</i></b> <b><i>... where a.price &gt;= ?1</i></b></p>

```
findByPriceGreaterThanOrEqual(10, 20);
```

# Keyword: Before, After and Between

<b>Uses</b>	<p><b><i>When you need to perform a less than, greater than or range comparison with date/time data types</i></b></p>
<b>Keyword Example</b>	<p><b><i>findByFoundedDateBefore(dateObj);</i></b> <b><i>findByFoundedDateAfter(dateObj);</i></b> <b><i>findByFoundedDateBetween(startDate, endDate);</i></b></p>
<b>JPQL Example</b>	<p><b><i>... where a.foundedDate &lt; ?1</i></b> <b><i>... where a.foundedDate &gt; ?1</i></b> <b><i>... where a.foundedDate between ?1 and ?2</i></b></p>

# Keyword: True and False

<b>Uses</b>	<p><b><i>Useful when comparing boolean values with true or false.</i></b></p>
<b>Keyword Example</b>	<p><b><i>findByActiveTrue();</i></b> <b><i>findByActiveFalse();</i></b></p>
<b>JPQL Example</b>	<p><b><i>... where a.active = true</i></b> <b><i>... where a.active = false</i></b></p>

# Keyword: IsNull, IsNotNull and NotNull

<b>Uses</b>	<p><b>Used to check whether a criteria value is null or not null</b></p>
<b>Keyword Example</b>	<p><b>findByStateIsNull();</b> <b>findByStateIsNotNull();</b> <b>findByStateNotNull();</b></p>
<b>JPQL Example</b>	<p><b>... where a.state is null</b> <b>... where a.state not null</b> <b>... where a.state not null</b></p>

# Keyword: In and NotIn

<b>Uses</b>	<p><b><i>When you need to test if a column value is part of a collection or set of values or not</i></b></p>
<b>Keyword Example</b>	<p><b><code>findByStateIn(Collection&lt;String&gt; states);</code></b> <b><code>findByStateNotIn(Collection&lt;String&gt; states);</code></b></p>
<b>JPQL Example</b>	<p><b><i>... where a.state <code>in</code> ?1</i></b> <b><i>... where a.state <code>not in</code> ?1</i></b></p>

# Keyword: IgnoreCase

<b>Uses</b>	<p><b><i>When you need to perform a case insensitive comparison</i></b></p>
<b>Keyword Example</b>	<p><b><code>findByStateIgnoreCase("ca");</code></b> <b><code>findByStateStartingWithIgnoreCase("c");</code></b></p>
<b>JPQL Example</b>	<p><b><code>... where UPPER( a.state ) = UPPER( ?1 )</code></b> <b><code>... where UPPER( a.state ) like UPPER( ?1% )</code></b></p>

# Keyword: OrderBy

<b>Uses</b>	<p><b><i>Used to setup an order by clause on your query</i></b></p>
<b>Keyword Example</b>	<p><b><i>findByStateOrderByCountryAsc();</i></b> <b><i>findByStateOrderByCountryDesc();</i></b></p>
<b>JPQL Example</b>	<p><b><i>... where a.state order by a.country asc</i></b> <b><i>... where a.state order by a.country desc</i></b></p>

# Keyword: First, Top and Distinct

<b>Uses</b>	<p><b><i>Used to limit the results returned by the query</i></b></p>
<b>Keyword Example</b>	<p><b><i>findFirstByStateLike("A");</i></b> <b><i>findTop5ByStateLike("A");</i></b> <b><i>findDistinctManufacturerByStateLike("A");</i></b></p>
<b>JPQL Example</b>	<p><b><i>... where a.state like ?1 limit 1</i></b> <b><i>... where a.state like ?1 limit 5</i></b> <b><i>select distinct ... where a.state like ?1</i></b></p>

# Summary



Query DSL advantages  
Basic Syntax  
Keywords, keywords and keywords