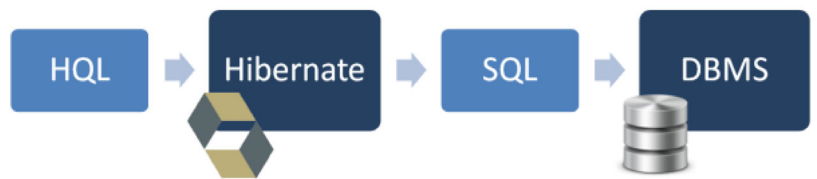


# UA06. LENGUAJE HQL

## UA06. Lenguaje HQL

Hibernate utiliza un lenguaje de consulta potente (**HQL, Hibernate Query Language**) que se parece a SQL. Sin embargo, **HQL es completamente orientado a objetos** y comprende nociones como herencia, polimorfismo y asociación.



Las consultas se escriben en **HQL** y **Hibernate se encarga de convertirlas al SQL** usado por la base de datos con la que estemos trabajando y ejecutarla para realizar la operación indicada.

**HQL es case-insensitive**, o sea que sus sentencias pueden escribirse en mayúsculas y minúsculas. Por lo tanto "SeLeCt", "seleCT", "select", y "SELECT" se entienden como la misma cosa. Lo único con lo que debemos tener cuidado es con los nombres de las clases que estamos recuperando y con sus propiedades, ahí si se distinguen mayúsculas y minúsculas. O sea, en este caso "pruebas.Hibernate.Usuario" NO ES LO MISMO que "PrueBAs.Hibernate.UsuArio".

Entre las características más importantes de HQL:

- **Soporte completo para operaciones relacionales:** HQL permite representar consultas SQL en forma de objetos. HQL usa clases y atributos o propiedades en vez de tablas y columnas.
- **Regresa sus resultados en forma de objetos:** Las consultas realizadas usando HQL regresan los resultados de las mismas en la forma de objetos o listas de objetos, que son más fáciles de usar.
- **Consultas Polimórficas:** Podemos declarar el resultado usando el tipo de la superclase e Hibernate se encargara de crear los objetos adecuados de las subclases correctas de

forma automática.

- **Soporte para características avanzadas:** HQL contiene muchas características avanzadas que son muy útiles y que no siempre están presentes en todas las bases de datos, o no es fácil usarlas, como paginación, fetch joins con perfiles dinámicos, inner y outer joins, etc. Además soporta proyecciones, funciones de agregación (max, avg), y agrupamientos, ordenamientos, y subconsultas.
- **Independiente del manejador de base de datos:** Las consultas escritas en HQL son independientes de la base de datos (siempre que la base de datos soporte la característica que estamos intentando utilizar).
- **Clausula from:** La consulta más simple que se puede realizar con Hibernate, es utilizando la cláusula from, la siguientes sería una consulta que mostraría todos los datos de una tabla de nombre Alumnos: **from Alumnos**
- **Cláusula select:** La cláusula select escoge qué objetos y propiedades devolver en el conjunto de resultados de la consulta. Un ejemplo de consulta podría ser **select alumno.nombre from Alumnos alumno where alumno.nombre like 'A%'**
- **La cláusula where:** La cláusula where nos permite refinar la lista de instancias retornadas. Si no existe ningún alias, puede referirse a las propiedades por nombre: **from Alumnos where nombre='Francisco'**. Si existe un alias, usaremos un nombre de propiedad calificado: **from Alumnos as alumnos where alumnos.nombre='Francisco'**. Esto retorna instancias de Alumnos llamados "Francisco".
- **Funciones de agregación.** Las consultas HQL pueden retornar resultados de funciones de agregación sobre propiedades: **select avg(alumnos.nota), sum(alumnos.nota), max(alumnos.nota), count(alumnos) from Alumnos alumnos**.
- **Expresiones.** Las expresiones utilizadas en la cláusula **where** incluyen lo siguiente: operadores matemáticos, operadores de comparación binarios, operadores lógicos , paréntesis ( ) que indican agrupación, funciones Java, etc.
- **La cláusula order by.** La lista retornada por una consulta se puede ordenar por cualquier propiedad de una clase retornada o componentes. La palabra **asc** o **desc** opcionales indican ordenamiento ascendente o descendente respectivamente.
- **La cláusula group by.** Una consulta que retorna valores agregados se puede agrupar por cualquier propiedad de una clase retornada o componentes.
- **Subconsultas.** Para bases de datos que soportan subconsultas, Hibernate soporta subconsultas dentro de consultas. Una subconsulta se debe encerrar entre paréntesis (frecuentemente por una llamada a una función de agregación SQL). Incluso se

permiten subconsultas correlacionadas (subconsultas que se refieren a un alias en la consulta exterior).

El ejemplo más simple de una sentencia HQL es el siguiente:

```
session.createQuery("from Person").getResultList();
```

Su equivalente en JPQL sería el siguiente, donde se requiere la cláusula select:

```
entityManager.createQuery("select p from Person p", Person.class).getResultList();
```

Obra publicada con **Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0** <<http://creativecommons.org/licenses/by-nc-sa/4.0/>>