

## Re: Comparing database dates: SELECT FROM WHERE Date(today) < MYDATECOL ?

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.databases/2005-02/0112.html>

---

*From:* Bjorn Abelli (*DoNotSpam.bjorn\_abelli\_at\_hotmail.com*)

*Date:* 02/07/05

Date: Mon, 7 Feb 2005 23:02:48 +0100

"Thomas Stein" wrote...

> *I want to get in a java prgm all those database records whose*  
> *MYDATE column contains a date greater equal a given other date*  
> *(say: today). How do I specify this in a SQL query from Java?*  
>  
> *Assume the comparison date in java is in a java.util.Date field. Do I have*  
> *to convert it*  
> *into a variable of type java.sql.Date?*  
>  
> *The resulting code should look similar to:*  
>  
> *java.util.Date somejavadate = new java.util.Date();*  
> *java.sql.Date somesqldate = new java.sql.Date(somejavadate);*  
>  
> *rs = stmt.executeQuery("SELECT ... FROM ... WHERE " + somesqldate + "<="*  
> *MYDATECOL");*

This will fail in most cases, as that construction will make use of the toString-method in java.sql.Date, which makes your querystring look something like:

```
SELECT ... FROM ... WHERE 2005-02-07 <= MYDATECOL
```

...which isn't a valid way to use dates...

You can in many databases use " around the date:

```
rs = stmt.executeQuery  
("SELECT ... FROM ... WHERE " + somesqldate + " <= MYDATECOL");
```

...but, in many databases, such as MS Access, it won't work, as it demands another formatting of a date literal.

> *MYDATECOL is defined with a type "datetime".*

So it's probably an SQL Server or MS Access/Jet... ;-)

> *However this does not work.*

> *How does the exact query look like?*

To make it less database specific, use a `PreparedStatement`, which is the recommended way to deal with insertion of values into query strings (instead of concatenations).

<http://java.sun.com/j2se/1.4.2/docs/api/java/sql/PreparedStatement.html>

... and as you're using a `DateTime` type in the DB, you possibly want to make use of `Timestamp` instead of `java.sql.Date`, as the latter truncates any time information.

<http://java.sun.com/j2se/1.4.2/docs/api/java/sql/Timestamp.html>

```
java.sql.Timestamp somesqldate =  
    new java.sql.Timestamp(somejavadate.getTime());
```

```
String sql =  
    "SELECT ... FROM ... WHERE MYDATECOL >= ?";
```

```
PreparedStatement stmt = con.prepareStatement(sql);
```

```
ps.setTimestamp(1, somesqldate )
```

```
rs = stmt.executeQuery();
```

> *Is the date query database product specific?*

Yes and no, as pointed out above.

By using a `PreparedStatement`, you can make your queries less db specific, but there are still some db:s that won't map timestamps, dates, etc, correctly. In those cases you will probably need to explore the specific `prepareStatement`-class in the driver, i.e. the documentation for the `jdbc-driver` from a specific vendor.

However, the example above will work for most db:s.

// Bjorn A