

Re: Server-side cursors for ResultSets – a pattern? (and what about Oracle?)

## Re: Server-side cursors for ResultSets – a pattern? (and what about Oracle?)

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.databases/2006-12/msg00012.html>

---

- *From:* "StM" <stmsat@xxxxxxxx>
  - *Date:* 4 Dec 2006 09:43:27 -0800
- 

Thank you for your clear response, Joe.

On 4 Dic, 17:41, "joeNOS...@xxxxxxxx" <joe.weinst...@xxxxxxxx> wrote:

With Oracle, you typically will get limited amounts of result set data at a time anyway. Just define the statement as you have, with forward-only and read-only. Setting a fetch size is OK, but not min-int ;). Set it to something reasonable so you can get some rows at once for some performance, say 100.

This is great news :)

With Sybase or SQLServer, by default, you'd get all the data sent out at once, but it doesn't immediately invade client memory as long as you read it and process it immediately. It is buffered up in the network and at the DBMS while you read it. Only if you try to run another statement before fully processing the query, do you cause the driver to have to buffer everything. `selectMethod=cursor` does work for MS drivers, and there will be a Sybase analog, but as I said, it isn't necessary if you just process the result as it shows up.

I don't know if this applies to SQL Server 2000 (and I'm using, if I'm not wrong, the SQL Server 2005 JDBC drivers). If it does, then I'm doing something wrong, as I get a JVM error saying that I exceed the JVM thread memory limit (or something like that, I'm not at the office now), with no work done (it should populate another table on another database). Besides, I'm not sure I've ever encountered a ResultSet that didn't need the complete fetch of the results before getting processed. Has this something to do with CachedResultSets (that, sadly, are only an interface in standard java)?

Joe Weinstein at BEA Systems

Re: Server-side cursors for ResultSets – a pattern? (and what about Oracle?)