

Question

[Guillaume Rongier](#) · Sep 20, 2018

JDBC Large query optimisation

Hi,

We recover a large amount of data from an external database (SQLServer, about 1 million rows in JDBC).

However, we have treatment time issue.

This process takes more than 30 minutes whereas on a "classic" SQL Server Management Studio type request takes less than a minute.

While searching on the internet, I came across this article: <http://makejavafaster.blogspot.com/2015/06/jdbc-fetch-size-performance.html>

It explains how in Java, we can tweak the FetchSize parameter of the JDBC driver to optimize this kind of process.

Is there a way to access this JDBC parameter through the EnsLib.SQL.OutboundAdapter adapter?

If not, how do you deal with this kind of situation of large volumetry?

Code used in the EnsLib.SQL.OutboundAdapter :

```
Method GetResultSetView(pRequest As Ens.StringRequest, Output pResponse As Ens.String
Response) As %Status
{
    set tStatus = $$$OK

    //Set ..Adapter.StatementAttrs = "fetchSize:200" <-- Make no difference
    //Set ..Adapter.StatementAttrs = "Type=TYPE_FORWARD_ONLY" <-- Make no difference

    $$$ThrowOnError(..Adapter.SetAutoCommit(0))

    try{

        set pResponse = ##class(Ens.StringResponse).%New()
        set sqlGetView = "SELECT * from TableOfMillionsLignes"

        Set pSnap = ##class(EnsLib.SQL.Snapshot).%New()

        //dimension du snapshot
        // -1 = Max
        set pSnap.MaxRowsToGet = -1

        $$$ThrowOnError(..Adapter.ExecuteQuery(.pRS,sqlGetView))

        //Most of the time is spend here
        $$$ThrowOnError(pSnap.ImportFromResultSet(pRS,0))
    }
```

```
$$$ThrowOnError(pSnap.%Save())

    set pResponse.StringValue = pSnap.%GblRef
}
catch exp
{
    Set tStatus = exp.AsStatus()
}
Quit tStatus
}
```

[#Database Transaction Processing](#) [#Ensemble](#) [#Business Operation](#) [#JDBC](#)

Source URL: <https://community.intersystems.com/post/jdbc-large-query-optimisation>