

RE: OLE DB Provider – how tell to not use cursor internally for jo

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2005-01/1256.html>

From: chosse (*chosse_at_discussions.microsoft.com*)

Date: 01/07/05

Date: Fri, 7 Jan 2005 04:59:03 -0800

For improve performance I build SP, which return me rowset. This SP process data on server and return only useful data and evidently decrease a data flow from server to client.

I need operate it under SQL 7.0 and 2000 too, in the select statement and more times, therefore i use openquery function for call SP, which return me rowset. And when i write select stament from me first subscription, it return me under SQL 2000 error, because DBE OLE provider use cursor for join tables. Under SQL 7.0 it run well, the cursor isn't use for join tables.

â€ Mike Epprecht (SQL MVP)" napÃ-sal (napÃ-sala):

> *Hi*

>

> *You can't control it. Why do you need to? It tries to use the most optimal access method for a target server.*

>

> *Regards*

> *Mike*

>

> *"chosse" wrote:*

>

> > *The OLE DB provider find way for join table in distributed query.*

> > *In some case it use cursor for join table dependent from capability of*

> > *target servers.*

> >

> > *My question is – May I tell to the OLE DB provider to prohibit use te cursor*

> > *for join tables by set*

> > *some parameter (for example provider string in the linkedserver) ?*

> >

> > *In some case OLE DB provider use cursor and it cause error. You can see it*

> > *in SQL Profiler.*

> > *It is example which*

> > *– run well under SQL 7.0 (target server for distibuted query), where OLE DB*

> > *provider doesn't use cursor*

> > *– doesn't run wel under SQL 2000 (target server for distibuted query), where*

```
>> OLE DB provider use cursor
>>
>> ~~~~~
>> Declaration part
>> ~~~~~
>> sp_serveroption @@servername, 'data access', 'true'
>> – this option enables a linked server for distributed query access
>> ~~~~~
>> CREATE PROCEDURE dbo.z_j_test_x
>> @as_param_1 VARCHAR(20),
>> @ai_param_2 INTEGER
>> AS
>> SET FMTONLY OFF
>> SET NOCOUNT ON
>> DECLARE @li_i INTEGER
>> CREATE TABLE #lt_hniezdo (
>> s_pk_1 VARCHAR(20) NOT NULL,
>> s_pk_2 INTEGER NOT NULL
>> )
>> SET @li_i = 0
>> WHILE (@li_i < 2) BEGIN
>> INSERT INTO #lt_hniezdo (s_pk_1, s_pk_2)
>> SELECT @as_param_1, (@ai_param_2 + @li_i)
>> SET @li_i = @li_i + 1
>> END
>> SELECT * FROM #lt_hniezdo ORDER BY s_pk_1, s_pk_2
>> DROP TABLE #lt_hniezdo
>> GO
>> ~~~~~
>>
>> SELECT * FROM
>> OPENQUERY(<SERVER NAME>, 'EXEC <schema name>.dbo.z_j_test_x "test1", 10')
>> ,OPENQUERY(<SERVER NAME>, 'EXEC <schema name>.dbo.z_j_test_x "test2", 100')
>>
```