DAL: OUTERJOIN and Multiple Sybase Servers Support (5/93)

Article Created: 12 May 1993

TOPIC ---

1) Does DAL support the SYBASE outerjoin operation? OUTERJOIN (see manual of SYBASE 4.9.1 page 2-192). It does the following:

Suppose we have two tables, t1 and t2 with columns colt1 and colt2:

```
table t1 with colt1         table t2 with colt2
     a                              a
     b                              b
     c
```

Now with 'select colt1, colt2 from t1, t2 where colt1 *= colt2'

```
outerjoin result
     a, a
     b, b
     c, null
```

So the *= join operator includes all rows from the first table that meets the statement's restrictions. The second table generates values if there is a match on the join condition. Otherwise, the second table generates null values.

2) Can I query two Sybase data bases, installed on the same VAX with DAL?

I installed SYBASE 4.0 on a VAX/VMS; DAL 1.3 is used (together with Clear/Access) to interrogate the database. This is all in a test environment but if everything turns out right, between 10 and 30 Mac's will follow. I also installed Sybase 4.9 on the same machine. (This apears as a SYBASE_49 process in the $SHO PROCESS table). Of course I want to access this second db also with DAL. The question now is how to connect to this second db. Can DAL connect to such a SYBASE_49 process? How to let DAL know that this second SYBASE exsists? (perhaps by the definition of a logical ??? What logical???)

Must we reinstall DAL, and how or when must we define the link to

SYBASE_49 process?

DISCUSSION ----------

1) DAL does not support outer joins currently. Engineering is considering this support in a release in the future, but the timeframe is not known.

2) The installation of Multiple Servers are described in the SYBASE Installation and Operations manual, section Installing Multiple Servers and is summarized below:

When you have a single server running, system logical names are used to determine where the interfaces file is and where the database devices are and so on. When two version of SQL Server are present on a single machine, PROCESS-level logical names must be used to keep the two systems separate. The definition for the logical name SYBASE_SYSTEM (normally defined as a system logical name) must differ from one SQL server version to another; therefore, you must use process-level logical name definitions for SYBASE_SYSTEM.

All users who are to have access to the secondary system must execute the command file SYLOGICALS.COM, located in the directory SYBASE_SECONDARY_DEVICE:[SYBASE.INSTALL]. You may wish to add the line:

@SYBASE_SECONDARY_DEVICE:[SYBASE.INSTALL]SYLOGICALS.COM

to the LOGIN.COM files of all users who will be using the secondary system. An easier way to define the process level logical names for all secondary systm users is to create a command file (for example, SYBASE_LOGIN.COM) which executes the file:

SYBASE_SECONDARY_DEVICE:[SYBASE.INSTALL]SYLOGICALS.COM

DAL will act as a user to the SQL Server, and it will choose the PRIMARY or SECONDARY SQL Server depending on what either system logical names, or process-level logical names are executed by the user account that DAL is logged on under. Once your customer successfully multiple servers with appropriate logical names, they should test access to these servers in native Transaction SQL first. DAL should then be used without further modifications.

Keywords: <None>

==========

This information is from the Apple Technical Information Library.

19960215 11:05:19.00