Procurement Desktop-Defense™

Clearing Database Blocks*

March 21, 2000

* Because this topic is not covered by the current PMO funded SPS Helpdesk Agreement, this document has been provided to help you resolve this issue. If you still need assistance after reviewing this document, please contact a representative from your Customer Support Team.

Table of Contents

1. Problem

When users are working in PD² and suddenly the application appears to "hang" or "freeze up" it is possible that a user is creating a ***block*** in the database.

When a user connects to the database using PD², Sybase Central or SQL Advantage the server identifies the connection as a process. Each process runs SQL commands, some of which are used to manipulate data in the database. Any command that attempts to insert, delete or update a row in a table in the database creates a temporary lock on that table. These locks block other processes from accessing that table until the transaction is committed to the database. Blocks occur when several users are attempting to modify the same table in the database before the current transaction has been committed. Blocks can also occur when the transaction log is full. As a result, no transactions can be committed until the log is cleared out.

In a multiple user environment a certain level of blocking is natural in order for users to have access to the same data. Each user process is temporarily blocked while waiting for the active process to complete. These types of blocks occur constantly throughout the day totally unnoticeable to the user. These blocks become apparent when a process that is running remains open. The other processes that are waiting for this process to complete are now "hung up" on the server until the current running process is finished.

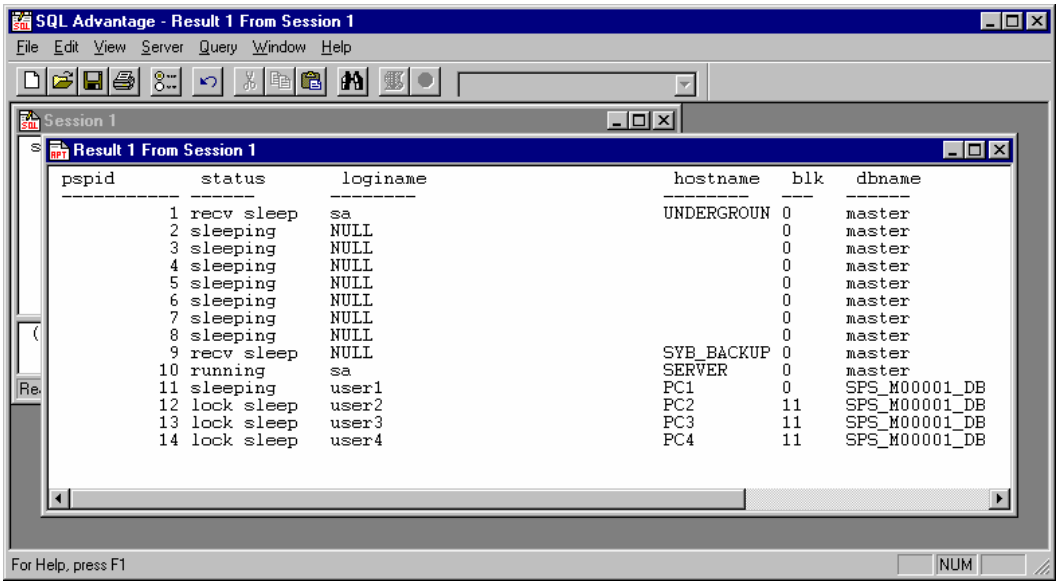
The following paper will guide you through the process of identifying and clearing database blocks.

2. Solution

Open SQL Advantage and connect to the server using your ‘sa’ id and password. Once you are connected to the server execute the following stored procedure.

```
sp_who
```

The results will appear as follows (See Figure 1).



pspid	status	loginame	hostname	blk	dbname
1	recv sleep	sa	UNDERGROUN	0	master
2	sleeping	NULL		0	master
3	sleeping	NULL		0	master
4	sleeping	NULL		0	master
5	sleeping	NULL		0	master
6	sleeping	NULL		0	master
7	sleeping	NULL		0	master
8	sleeping	NULL		0	master
9	recv sleep	NULL	SYB_BACKUP	0	master
10	running	sa	SERVER	0	master
11	sleeping	user1	PC1	0	SPS_M00001_DB
12	lock sleep	user2	PC2	11	SPS_M00001_DB
13	lock sleep	user3	PC3	11	SPS_M00001_DB
14	lock sleep	user4	PC4	11	SPS_M00001_DB

Figure 1: Blocked by spid 11 (Screen 1)

Notice the column on the far right labeled “blk”. Under normal circumstances this column contains all zeros. But when a process is blocking other users in PD² a process id will appear in this column. In this case the process 11 is blocking three other processes in the database. Make a note of this number then go back to the very first column on the left and scroll down until you locate process 11. Once you have found process 11, scroll across to the login name associated with process 11. In this case the user is “user1”. Now look at the last column in the sp_who results to determine what type of command is running that caused the block. In this case the command column for process 11 reads “LOG SUSPEND”(See Figure 2).

Note: If a process is in “LOG SUSPEND” that usually means that the transaction log is full and the transaction has been suspended until the log is cleared out. You can resolve all blocks in the database by dumping the transaction log. You **DO NOT** need to go through the following steps. However, if the process is not in log suspend then continue with this resolution. (Log Suspend was chosen for this example because it is easy to replicate and can be used to demonstrate how blocks will appear on the server.)

At this point you should contact user1 and find out what they are doing in PD² and have them log out of the application.

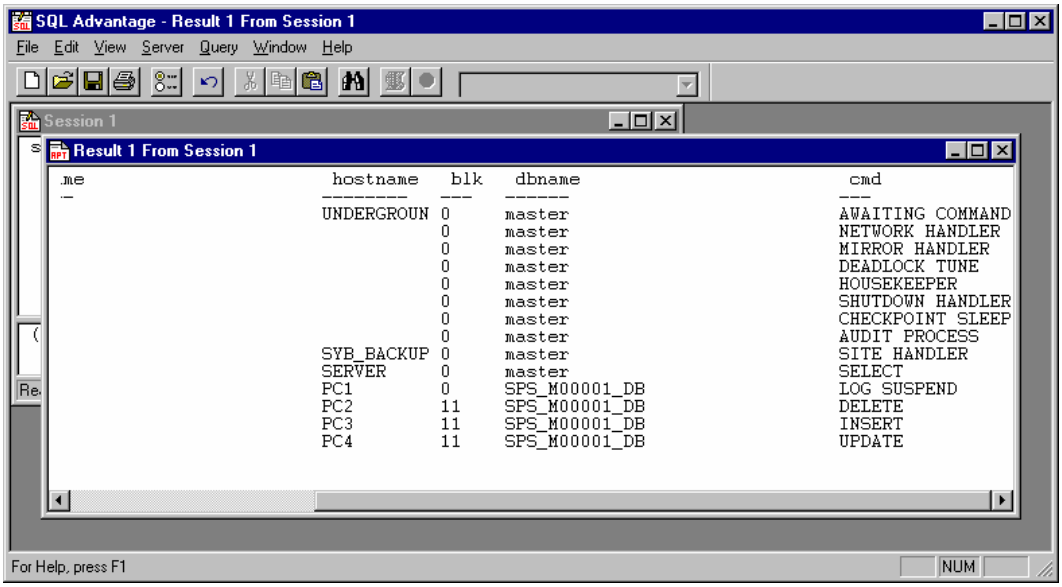


Figure 2: Blocked by spid 11 (Screen 2)

Once user1 has logged out of PD². Run the sp_who command again to make sure all blocks have been cleared. In this case process 12 is now causing a block. Process 12 is associated with user2 who is in log suspend (See Figure 3 and Figure 4). Have user2 log out of PD² and run sp_who again.

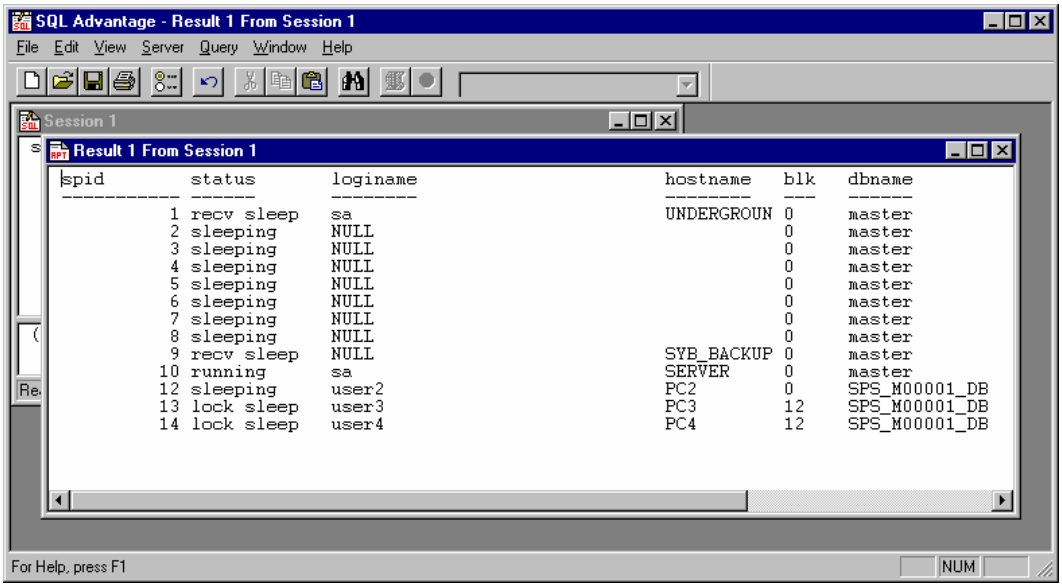


Figure 3: Blocked by spid 12 (Screen 1)

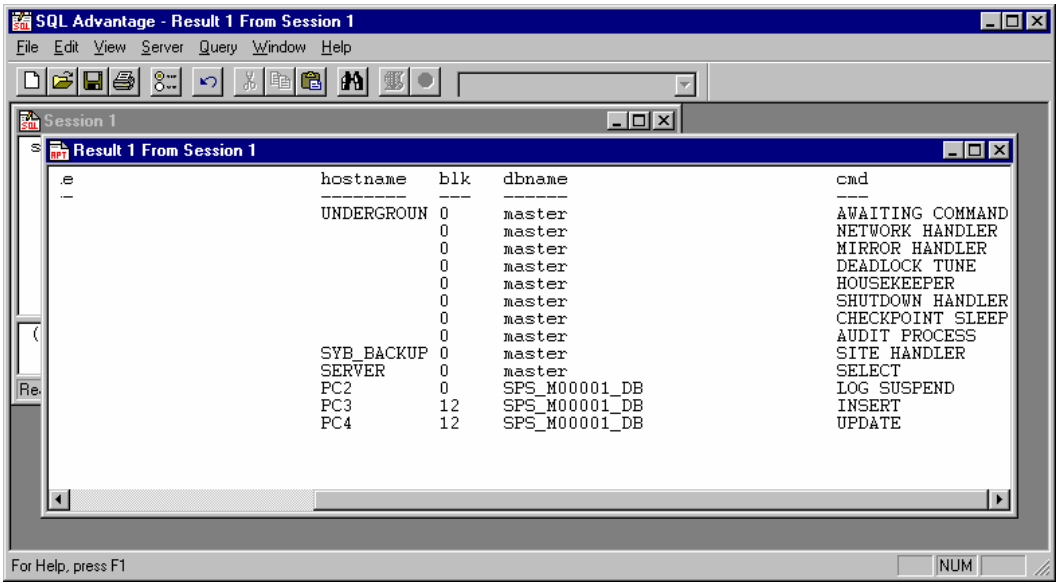


Figure 4: Blocked by spid 12 (Screen 2)

When the results from the sp_who command appear process 14 is now creating a block in the database. Process 14 is associated with user4 who is currently in log suspend (See Figure 5 and Figure 6). Have user4 log out of PD² and run sp_who again.

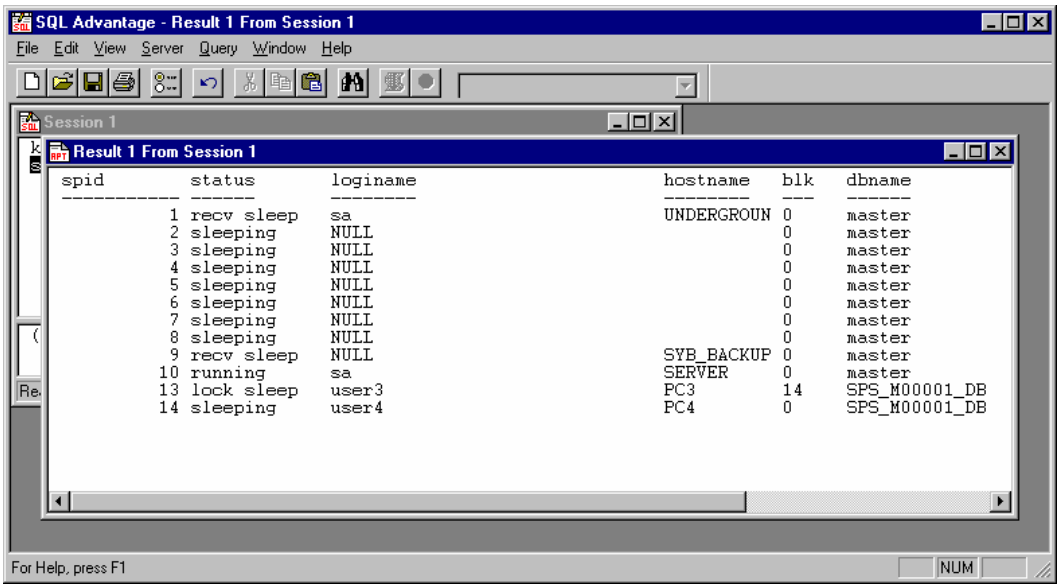


Figure 5: Blocked by spid 14 (Screen 1)

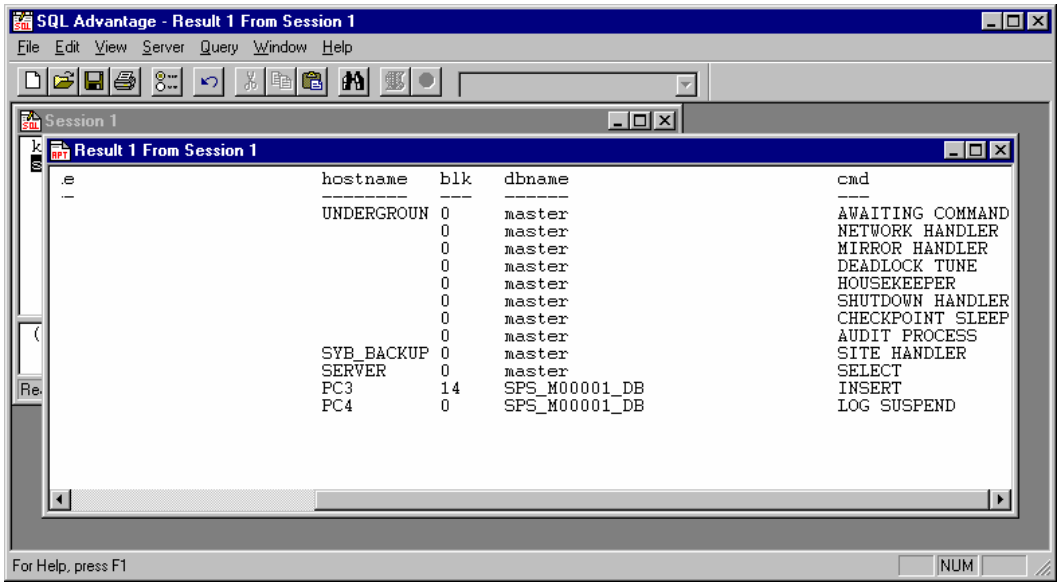


Figure 6: Blocked by spid 14 (Screen 2)

At this point there are no more blocks in the database (See Figure 7). However, in this example, user3 is currently in log suspend (See Figure 8). In order to resolve this problem you will have to dump the transaction log which is probably full.

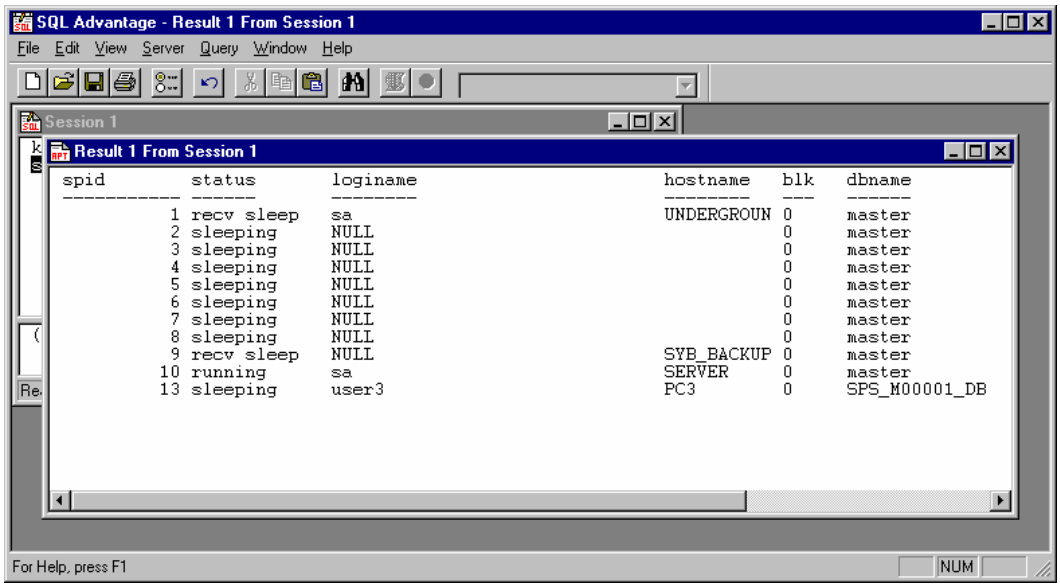


Figure 7: No blocks (Screen 1)

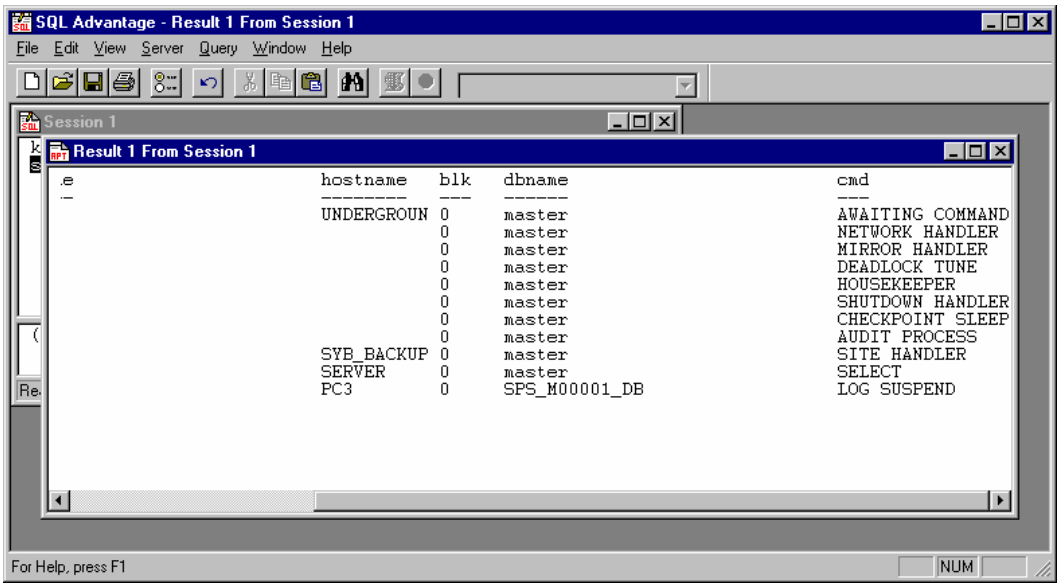


Figure 8: No Blocks (Screen 2)

Once all the blocks have been cleared, users should be able to log back into PD² and continue working.