

# Oracle Sessions

Under some circumstances, an Oracle user initiated session can remain connected after real work has been completed, sometimes with locks in place. These locks will prevent you from being able to continue to work normally with your tables.

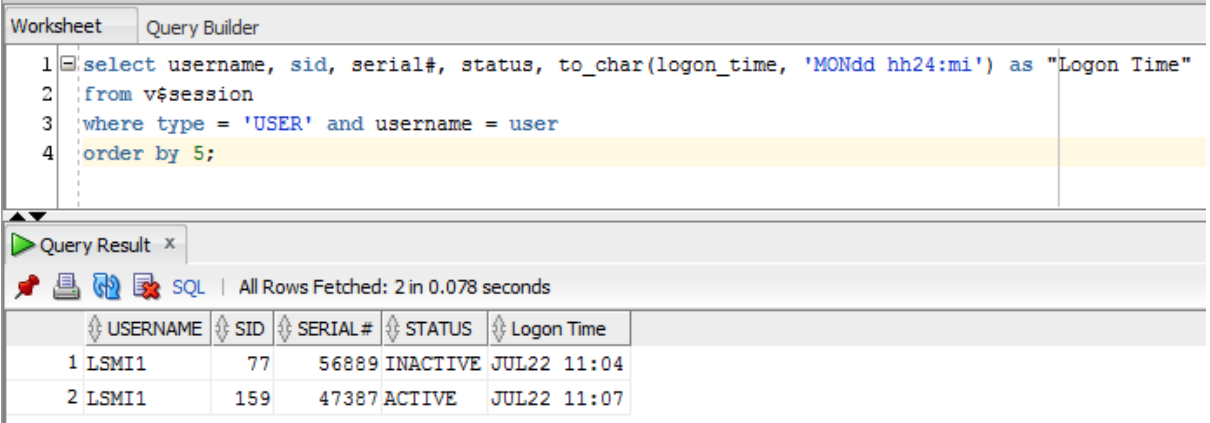
A common trigger for this situation is where a user on a laptop drops the VPN before they disconnect from the database. In such a situation when the user reconnects to the database they may find that they cannot complete any useful work because of the locks still in place - a typical message under such a scenario is "RESOURCE BUSY AND ACQUIRE WITH NOWAIT SPECIFIED".

To solve this situation the Monash Oracle databases have been configured so that you can kill your old process/session. If you find that you are experiencing problems such as above, follow these steps:

## 1. Connect using sql developer and run the SQL command:

```
select username, sid, serial#, status,
to_char(logon_time, 'MONdd hh24:mi') as "Logon Time"
from v$session
where type = 'USER' and username = user
order by 5;
```

This will provide details of all the sessions currently running in Oracle under your username:



The screenshot shows the SQL Developer interface. The top pane displays a query: `select username, sid, serial#, status, to_char(logon_time, 'MONdd hh24:mi') as "Logon Time" from v$session where type = 'USER' and username = user order by 5;`. The bottom pane shows the query results in a table with 2 rows and 5 columns: USERNAME, SID, SERIAL#, STATUS, and Logon Time. The first row shows an INACTIVE session for user LSMI1 with SID 77. The second row shows an ACTIVE session for user LSMI1 with SID 159.

	USERNAME	SID	SERIAL#	STATUS	Logon Time
1	LSMI1	77	56889	INACTIVE	JUL22 11:04
2	LSMI1	159	47387	ACTIVE	JUL22 11:07

Each SQL Developer session you initiate will often open several connections to the database - *if there are multiples, you will need to kill off all **inactive** sessions.*

## 2. Kill off the 'hung' connections:

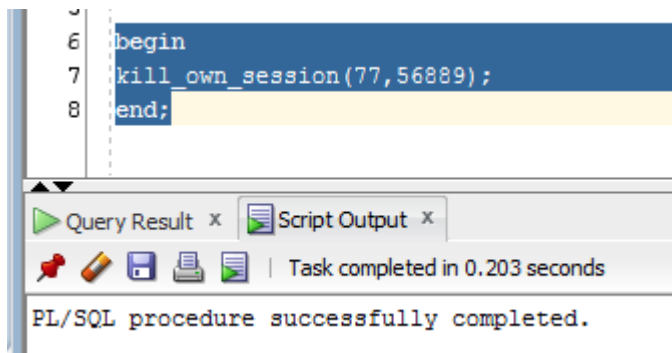
For example, in the above display the connection with a logon time of 11:04 represents a 'hung' session - to kill such sessions off we have an Oracle procedure available called **kill\_own\_session**. The parameters to this procedure are the SID and the SERIAL# - for example to kill the INACTIVE session above you would use:

```
kill_own_session(77,56889)
```

In SQL Developer the procedure is called inside an anonymous PL/SQL block (surrounded by begin/end). To kill the session above, the command would be:

```
begin
  kill_own_session(77,56889);
end;
```

The command operation is shown below:



Following this, a repeat of the original select will show that the connections have now been killed:

The screenshot shows the SQL Developer interface with a query executed. The query is:

```
select username, sid, serial#, status, to_char(logon_time, 'MONdd hh24:mi') as "Logon Time"
from v$session
where type = 'USER' and username = user
order by 5;
```

The 'Query Result' tab is active, showing the following data:

	USERNAME	SID	SERIAL#	STATUS	Logon Time
1	LSMI1	77	56889	KILLED	JUL22 11:04
2	LSMI1	159	47387	ACTIVE	JUL22 11:07

The status bar indicates 'All Rows Fetched: 2 in 0.047 seconds'.

Oracle will automatically clean up these killed sessions. With the killing of these sessions, any locks associated with them will also be cleared and you will be able to proceed to work normally.