

Querying Container Data Objects

By Ahmed Baraka

Objectives

In this lecture, you will learn how to perform the following:

- Understand querying container data objects
- Set **CONTAINER_DATA** attributes for a common user
- Obtain information about **CONTAINER_DATA** attributes set in a multitenant environment
- Remove **CONTAINER_DATA** attributes for a user

What are Container Data Objects?

- Tables or views containing data pertaining to multiple containers
- Examples: `V$`, `CDB_*`, and `DBA_HIST*` views
- They have a `CON_ID` column:

CON_ID	The Container the Rows Pertain to
0	Whole CDB
1	The root container (<code>CDB\$ROOT</code>)
2	The seed container (<code>PDB\$SEED</code>)
All other IDs	User-created PDBs, application roots, or application seeds

- To identify container data objects:

```
SELECT VIEW_NAME FROM CDB_VIEWS WHERE CONTAINER_DATA = 'Y';  
SELECT TABLE_NAME FROM CDB_TABLES WHERE CONTAINER_DATA = 'YES';
```

Querying Container Data Objects: Default Behavior and Modifying It

- By default, common users created by the DBA can retrieve rows related to the current container. To view rows of another container, the user must connect to it

- To allow a common user to see the rows of all the PDBs:

```
ALTER USER c##user1 SET CONTAINER_DATA=ALL CONTAINER=CURRENT;
```

- To revert it back to the default behavior:

```
ALTER USER c##user1 SET CONTAINER_DATA=DEFAULT CONTAINER=CURRENT;
```

- To allow a common user to see rows of only specific PDBs:
 - The root must always be included

```
ALTER USER c##user1 SET CONTAINER_DATA=(CDB$ROOT,PDB1)  
CONTAINER=CURRENT;
```

Adding and Removing Specific Container from the CONTAINER_DATA Attributes

- To add specific PDBs to the CONTAINER_DATA attributes of a user:

```
ALTER USER c##user1 ADD CONTAINER_DATA=(PDB3) CONTAINER=CURRENT;
```

- To remove specific PDBs from the CONTAINER_DATA attributes of a user:

```
ALTER USER c##user1 REMOVE CONTAINER_DATA=(PDB3) CONTAINER=CURRENT;
```

- Setting the attributes overwrites the existing ones:

```
ALTER USER c##user1 SET CONTAINER_DATA=(CDB$ROOT, PDB1, PDB2)  
CONTAINER=CURRENT;
```

Setting CONTAINER_DATA Attributes for Specific Objects

- To set CONTAINER_DATA attributes for specific objects, use the FOR clause:

```
SQL> ALTER USER c##user2 SET CONTAINER_DATA = (CDB$ROOT, PDB1)
FOR sys.V_$SESSION CONTAINER=CURRENT;
```

```
SQL> conn C##USER2/ABcd##1234
```

```
SQL> SELECT USERNAME, CON_ID FROM sys.V_$SESSION ;
```

```
SQL> SELECT USERNAME, CON_ID FROM V$SESSION;
```

USERNAME	CON_ID
-----	-----
SYS	0
C##USER2	1
SYS	3

To Obtain Information about the CONTAINER_DATA Attributes Set in a Multitenant Environment

```
SELECT USERNAME, DEFAULT_ATTR, OWNER, OBJECT_NAME,  
ALL_CONTAINERS, CONTAINER_NAME  
FROM CDB_CONTAINER_DATA  
ORDER BY USERNAME, OBJECT_NAME;
```

USERNAME	DEFAULT	OWNER	OBJECT_NAME	ALL	CONTAINER_NAME
-----	-----	-----	-----	---	-----
C##USER2	N	SYS	V_\$SESSION	N	CDB\$ROOT
C##USER2	N	SYS	V_\$SESSION	N	PDB1
SYS	Y			Y	
SYSTEM	Y			Y	
...					

Removing CONTAINER_DATA Attributes for a User

- To remove all PDB-level CONTAINER_DATA attributes:

```
ALTER USER c##user2 SET CONTAINER_DATA=DEFAULT CONTAINER=CURRENT;
```

- It does not remove the attributes at the object-level

- To remove object-level CONTAINER_DATA attributes:

```
ALTER USER c##user2 SET CONTAINER_DATA=DEFAULT FOR V_$SESSION  
CONTAINER=CURRENT;
```


Summary

In this lecture, you should have learnt how to perform the following:

- Understand querying container data objects
- Set **CONTAINER_DATA** attributes for a common user
- Obtain information about **CONTAINER_DATA** attributes set in a multitenant environment
- Remove **CONTAINER_DATA** attributes for a user



Ahmed Baraka
Oracle Database Administrator