Identifying databases that might exceed the OBID limit

Last Updated: 2022-06-21

You can determine if a database has enough available object identifiers (OBIDs) to accommodate the OBID usage of an operation.

About this task

Certain operations, such as creating a table space, use OBIDs. If the number of OBIDs for a database exceeds the limit of 32,767, the operation fails and SQLCODE -497 is issued.

Procedure

To identify databases that might exceed the OBID limit, run the following query:

```
WITH MTTS_DB AS
( SELECT     DBNAME,
             DBID,
             COUNT(DBNAME) AS NUMTS,
             SUM(NTABLES)  AS NUMTB
   FROM      SYSIBM.SYSTABLESPACE
   WHERE     NTABLES > 1
   GROUP BY  DBNAME, DBID
)
,
MTTS_OBIDS AS
(
  SELECT DBID, OBID AS OBDID
   FROM SYSIBM.SYSTABLES
   WHERE DBID IN ( SELECT DBID FROM MTTS_DB )
   AND   OBID <> 0
  UNION
  SELECT DBID, OBID AS OBDID
   FROM SYSIBM.SYSTABLESPACE
   WHERE DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT DBID, PSID AS OBDID
   FROM SYSIBM.SYSTABLESPACE
```

```
    FROM SYSIBM.SYSTABLESPACE
    WHERE DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT DBID, OBID AS OBDID
    FROM SYSIBM.SYSINDEXES
    WHERE DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT DBID, ISOBID AS OBDID
    FROM SYSIBM.SYSINDEXES
    WHERE DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT DBID, OBID AS OBDID
    FROM SYSIBM.SYSTRIGGERS
    WHERE DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT DBID, OBID AS OBDID
    FROM SYSIBM.SYSCHECKS
    WHERE DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT TAB.DBID, REL.RELOBID2 AS OBDID
    FROM  SYSIBM.SYSRELS   AS REL,
          SYSIBM.SYSTABLES AS TAB
    WHERE REL.CREATOR = TAB.CREATOR
    AND   REL.TBNAME  = TAB.NAME
    AND   TAB.DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT TAB.DBID, REL.RELOBID1 AS OBDID
    FROM  SYSIBM.SYSRELS   AS REL,
          SYSIBM.SYSTABLES AS TAB
    WHERE REL.REFTBCREATOR = TAB.CREATOR
    AND   REL.REFTBNAME  = TAB.NAME
    AND   TAB.DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT TAB.DBID, AUX.AUXRELOBID AS OBDID
    FROM  SYSIBM.SYSAUXRELS AS AUX,
          SYSIBM.SYSTABLES  AS TAB
    WHERE AUX.TBOWNER = TAB.CREATOR
    AND   AUX.TBNAME  = TAB.NAME
    AND   TAB.DBID IN ( SELECT DBID FROM MTTS_DB )
  UNION
  SELECT TAB.DBID, XML.XMLRELOBID AS OBDID
    FROM  SYSIBM.SYSXMLRELS AS XML,
          SYSIBM.SYSTABLES  AS TAB
    WHERE XML.TBOWNER = TAB.CREATOR
    AND   XML.TBNAME  = TAB.NAME
    AND   TAB.DBID IN ( SELECT DBID FROM MTTS_DB )
)
SELECT   CURRENT SERVER                AS DB2LOC,
         MTTS_DB.DBNAME                AS DBNAME,
         MTTS_DB.DBID                  AS DBID,
         MTTS_DB.NUMTS                 AS NUM_MTTS,
         MTTS_DB.NUMTB                 AS NUM_TB,
         COUNT(MTTS_OBIDS.OBDID)       AS OBIDS_USED,
         32767 - COUNT(MTTS_OBIDS.OBDID) AS OBIDS_AVAILABLE,
         (MTTS_DB.NUMTB * 2)           AS OBIDS_NEEDED
FROM     MTTS_DB,
         MTTS_OBIDS
WHERE    MTTS_DB.DBID = MTTS_OBIDS.DBID
GROUP BY MTTS_DB.DBNAME, MTTS_DB.DBID, MTTS_DB.NUMTS, MTTS_DB.NUMTB
ORDER BY DBNAME
;
```

What to do next

If not enough OBIDs are available to accommodate an operation, take one of the following actions:

- Specify a different database.
- Drop all unused table spaces or indexes in the database and issue a COMMIT.
- If the database contains multi-table table spaces that contain a mix of used and unused tables, drop all unused tables and issue a COMMIT. Run the REORG utility on each affected table space, and then run the MODIFY RECOVERY utility to reclaim the dropped table OBIDs. For more information, see Reclaiming space in the DBD.

For more information, see -497.

Parent topic:

→ Implementing your database design