Practice: Automatic Transaction Rollback

Practice Target

This practice demonstrates how Oracle Database 23ai automatically rolls back low-priority blocking transactions to allow higher-priority waiting transactions to proceed after a configurable wait period.

Practice Overview

In high level, you will perform the following tasks:

1. View and modify automatic transaction rollback parameters at the PDB level
2. Create and execute blocking and waiting transactions with different priorities
3. Observe automatic rollback behavior via system views

View and Configure Automatic Rollback Parameters

In this section, you will configure the wait time for automatic rollback of blocking transactions.

1. Open a Putty session, connect as oracle to the vm then invoke SQL*Plus with connecting to freepdb1 as SYS

```
sqlplus sys/oracle@freepdb1 as sysdba
```

2. View the current automatic rollback parameters.
   The waiting time is set to an enormous value (2147483647).

```
COL name FORMAT a40 COL
value FORMAT a15
SELECT name, value FROM v$parameter WHERE name LIKE 'priority_txn%';
```

3. Set wait targets to 60 seconds (1 minutes).
   The parameter take effect at the PDB level.

```
ALTER SYSTEM SET priority_txns_high_wait_target = 60; ALTER SYSTEM SET
priority_txns_medium_wait_target = 60;
```

4. Re-query to verify changes.

```
SELECT name, value FROM v$parameter WHERE name LIKE 'priority_txn%';
```

Simulate Automatic Rollback with Two Sessions

In this section, you will simulate blocking and waiting transactions using two sessions and observe rollback behavior.

Session 1 (low priority - blocking transaction):

5. Connect as HR

```
conn hr/oracle@freepdb1
SET SQLPROMPT "session 1 low> "
```

6. Set the current session to low priority.

```
ALTER SESSION SET txn_priority = low;
```

7. Create the testing table.

```
CREATE TABLE dept_demo (
        dept_id         NUMBER PRIMARY KEY,
        dept_name VARCHAR2(50)
);

INSERT INTO dept_demo VALUES (1,NULL); COMMIT;
```

8. Issue the following update statement.

```
UPDATE dept_demo SET dept_name = 'Finance' WHERE dept_id = 1;
-- Do not commit
```

Session 2 (high priority - waiting transaction):

9. Open a Putty session, connect as oracle to the vm then invoke SQL*Plus with connecting to freepdb1 as HR. This is Session 2.

```
sqlplus hr/oracle@freepdb1
SET SQLPROMPT "session 2 high> "
```

10. Set high priority

```
ALTER SESSION SET txn_priority = high;
```

11. Try to modify same row (will wait).

```
UPDATE dept_demo SET dept_name = 'Accounting' WHERE dept_id = 1;
```

Wait more than 1 minute. Oracle will roll back the blocking transaction from Session 1, and Session 2 will proceed.

12. In session 1, try committing the transition.

```
Commit;
```

You should receive the following error:

```
ORA-63302: Transaction must roll back
ORA-63300: Transaction is automatically rolled back since it is blocking a
higher priority transaction from another session.
```

Monitor Statistics and System Behavior

You can monitor rollback statistics using system views:

13. In session 2, connect to the pdb as sys

```
conn sys/oracle@freepdb1 as sysdba
```

14. See rollback events triggered

```
SELECT name, value FROM v$sysstat WHERE name LIKE '%txns rollback%';
```

15. View the tail of the alter log file.
    The alter log file contains information about the automatic rollback transaction.

```
host tail /opt/oracle/diag/rdbms/free/FREE/trace/alert_FREE.log
```

16. As a cleanup drop the table and reset the parameters to their default values.

```
conn sys/oracle@freepdb1 as sysdba
ALTER SYSTEM RESET priority_txns_high_wait_target; ALTER SYSTEM RESET
priority_txns_medium_wait_target;

conn hr/oracle@freepdb1 DROP TABLE
dept_demo purge;
```

Summary

You have configured the automatic transaction rollback feature in Oracle Database 23ai and tested its behavior by simulating transaction blocking scenarios. This feature helps maintain concurrency and responsiveness by enforcing fairness based on transaction priority and timeout thresholds.