enq: TX - row lock contention

1.
   Lock held on a particular row by a transaction to prevent other transactions from modifying it
2. sessionSQL

```
select sid,sql_text
from
v$session s,v$sql q
where  sid in
(select sid from v$session
    where state in ('WAITING')
    and wait_class != 'Idle'
    and event='enq: TX - row lock contention'
    and (q.sql_id = s.sql_id or q.sql_id = s.prev_sql_id));
```

3. blockedsession

```
select
    blocking_session,
    sid, serial#,
    wait_class,
    seconds_in_wait
from
    v$session
where
    blocking_session is not NULL
order by
    blocking_session;
```

4. blockedsessionSQL

```
SELECT DISTINCT
    -- Snapshot ID
    MIN(blocked.snap_id) AS first_snap_id,
    MAX(blocked.snap_id) AS last_snap_id,

    -- Sample ID and Time
    MIN(blocked.sample_id) AS first_sample_id,
    MIN(blocked.sample_id) AS last_sample_id,
    to_char(MIN(blocked.sample_time), 'YYYY-MM-DD HH24:MI:SS') AS first_sample_time,
    to_char(MAX(blocked.sample_time), 'YYYY-MM-DD HH24:MI:SS') AS last_sample_time,

    -- Session causing the block
    blocker.instance_number    AS blocker_instance_number,
    blocker.machine            AS blocker_machine,
    blocker.program            AS blocker_program,
    blocker.session_id         AS blocker_sid,
    blocker_user.username      AS blocker_username,
    ' -> ' AS is_blocking,

    -- Sesssion being blocked
    blocked.instance_number    AS blocked_instance_number,
    blocked.machine            AS blocked_machine,
    blocked.program            AS blocked_program,
    blocked.session_id         AS blocked_sid,
    blocked_user.username      AS blocked_username,
    blocked.session_state      AS blocked_session_state,
    blocked.event              AS blocked_event,
    blocked.blocking_session   AS blocked_blocking_session,
    blocked.sql_id             AS blocked_sql_id,
    blocked.sql_child_number   AS blocked_sql_child_number,
    sys_obj.name               AS blocked_table_name,
    dbms_rowid.rowid_create(rowid_type => 1, object_number => blocked.current_obj#, relative_fno =>
blocked.current_file#, block_number
    => blocked.current_block#, row_number => blocked.current_row#) AS blocked_rowid,
    blocked_sql.sql_text       AS blocked_sql_text
```

```
FROM
    dba_hist_active_sess_history    blocker
    INNER JOIN dba_hist_active_sess_history    blocked ON blocker.session_id = blocked.blocking_session
    INNER JOIN sys.obj$                        sys_obj ON sys_obj.obj# = blocked.current_obj#
    INNER JOIN dba_users                       blocker_user ON blocker.user_id = blocker_user.user_id
    INNER JOIN dba_users                       blocked_user ON blocked.user_id = blocked_user.user_id
    LEFT OUTER JOIN v$sql                      blocked_sql ON blocked_sql.sql_id = blocked.sql_id
                                            AND blocked_sql.child_number = blocked.sql_child_number
    LEFT OUTER JOIN v$sql                      blocker_sql ON blocker_sql.sql_id = blocker.sql_id
                                            AND blocker_sql.child_number = blocker.sql_child_number
WHERE
    blocked.snap_id BETWEEN &begin_snap_id AND &end_snap_id
    AND blocked.event = 'enq: TX - row lock contention'
GROUP BY
    blocker.instance_number,
    blocker.machine,
    blocker.program,
    blocker.session_id,
    blocker_user.username,
    ' -> ',
    blocked.instance_number,
    blocked.machine,
    blocked.program,
    blocked.session_id,
    blocked_user.username,
    blocked.session_state,
    blocked.event,
    blocked.blocking_session,
    blocked.sql_id,
    blocked.sql_child_number,
    sys_obj.name,
    dbms_rowid.rowid_create(rowid_type => 1, object_number => blocked.current_obj#, relative_fno =>
blocked.current_file#, block_number
    => blocked.current_block#, row_number => blocked.current_row#),
    blocker_sql.sql_text,
    blocked_sql.sql_text
ORDER BY
    first_sample_id;
```