Schema Annotations

Data Use Case Domains

ORACLE Database 23ai

Read-Only Per-PDB Standby

Operational Property Graphs

Real-time SQL Plan Management

SQL Enhancements

Lock-Free Reservation

JSON Relational Duality

AI Vector Search

True Cache

SQL Firewall

Priority Transactions

Javascript in Oracle Database

Up to 4096 Columns per Table

Automatic In-Memory

Globally Distributed Database

Row Locks and Oracle Database

Transaction 1

```
UPDATE hr.employees SET
phone_number='515.555.1234'
WHERE employee_id=118
AND email='GHIMURO'
AND phone_number =
'515.127.4565';

1 row updated.
```

Transaction 2

```
UPDATE hr.employees SET
phone_number='515.555.1235'
WHERE employee_id=118
AND email='GHIMURO'
AND phone_number =
'515.127.4565';

-- SQL*Plus does not show
-- a row updated message or
-- return the prompt.
```

... is waiting until ...

How can you solve this?

- If a transaction acquires row locks and is not committed or aborted, it can block other transactions trying to update the same rows.
  This can lead to lengthy application stalls and typically requires the hands-on involvement of a DBA.

Solutions:

- DBAs manually terminate the blocking transaction by using the **ALTER SYSTEM KILL SESSION** command.

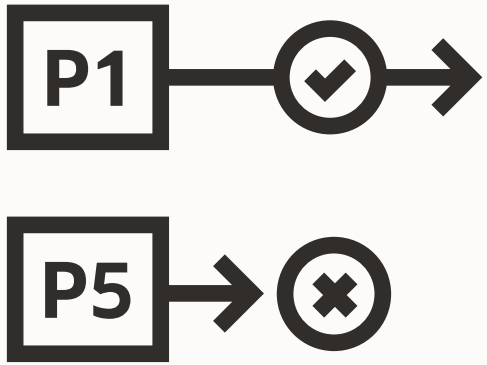
```
     Oracle     Table                          Table
  Sid Username   Locked                         Owner      MODE_HELD
----- ---------- ------------------------------ ---------- ---------------
  468 APP1       EMPLOYEES                      HR         Row Excl (SX)
  629 APP2       EMPLOYEES                      HR         Row Excl (SX)
```

- DBAs create a **Database Resource Manager Plan** with plan directives such as MAX_EST_EXEC_TIME, MAX_IDLE_TIME, MAX_IDLE_BOCKER_TIME.

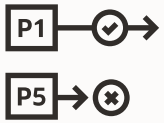
New In 23ai

Priority Transactions in 23ai

- Priority Transactions roll automatically back low-priority transactions that are blocking higher-priority transactions from obtaining row locks.
- In 23ai, users can enable functionality that automatically prioritizes high-priority transactions over low-priority transactions.
- Low-priority transactions that block high-priority transactions will be automatically aborted.



- Priority Transaction reduces the admin burden on the DBA and avoids long transactions.



Configuring Priority Transaction

1. DBA configures the maximum time duration, in seconds, a transaction with priority HIGH and MEDIUM will wait before the database rolls back a lower priority transaction.

```
priority_txns_high_wait_target   = {1 to unlimited} in seconds
priority_txns_medium_wait_target = {1 to unlimited} in seconds

alter system set priority_txns_high_wait_target=10 scope=both;
```

2. Applications specify the priority of their transactions with ALTER SESSION.

```
txn_priority =  {HIGH|MEDIUM|LOW}

alter session set txn_priority=LOW;
```

Priority Transaction is only enabled when **both** the transaction priority and the wait target parameters are set.

3. DBA can specify or change the mode for Priority Transactions.

```
priority_txns_mode = {ROLLBACK|TRACK}
```


Priority Transactions in Oracle Database 23ai

Low priority

```
UPDATE hr.employees SET
phone_number='515.555.1234'
WHERE employee_id=118
AND email='GHIMURO'
AND phone_number =
'515.127.4565';

1 row updated.
```

```
SELECT * from hr.employees;
*
ERROR at line 1:
ORA-63302: Transaction must roll
back

ROLLBACK;
```

High/Medium priority

```
UPDATE hr.employees SET
phone_number='515.555.1235'
WHERE employee_id=118
AND email='GHIMURO'
AND phone_number =
'515.127.4565';
```

... is waiting 10 seconds (*)
and then the update succeeds!

(*) PRIORITY_TXNS_**HIGH**_WAIT_TARGET=10
PRIORITY_TXNS_**MEDIUM**_WAIT_TARGET=10

Example

See posting
[Priority Transactions with high, medium and low priority]

Time	Transaction1 Low	Transaction2 Low	Transaction3 High (Default)

t1	alter session
	set txn_priority = low;
	
	select
	sys_context('userenv','SID');
	SID
	----- 	
	630
	
	update scott.mycheck set t=0;
	1 rows updated.		

t2		alter session
		set txn_priority = low;
		
		select
		sys_context('userenv','SID');
		SID
		----- 	
		940
		
		update scott.mycheck set t=10;
		-- is waiting	

t3			select SYS_CONTEXT('USERENV','SID'); SID ----- 1093 update scott.mycheck set t=1000; -- is waiting
t4=t3+10		1 row updated.	
t5=t4+10	SQL> select * from scott.mycheck; * ERROR at line 1: ORA-63302: Transaction must roll back ORA-63300: Transaction is automatically rolled back since it is blocking a higher priority transaction from another session. Help: https://docs.oracle.com/error-help/db/ora-63302/	SQL> select * from scott.mycheck; * ERROR at line 1: ORA-63302: Transaction must roll back ORA-63300: Transaction is automatically rolled back since it is blocking a higher priority transaction from another session. Help: https://docs.oracle.com/error-help/db/ora-63302/	1 row updated. SQL> select * from scott.mycheck; --------- 1000

Priority Transaction

- Allows transactions to be automatically rolled back and includes parameters to control this behavior
- Simple and easy to configure
- Not time-consuming
- Can be monitored in alert file, v$transactions, v$sysstat or v$session
- More information
  - Priority Transactions with high, medium and low priority
  - Documentation: Database Administrator's Guide: Managing Transactions