17

Monitoring and Resolving Lock Conflicts

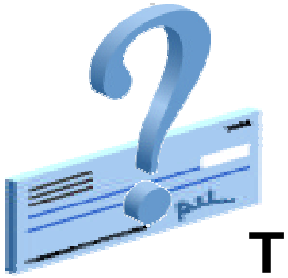
Objectives

After completing this lesson you should be able to do the following:

- **Detect and resolve lock conflicts**
- **Manage deadlocks**

Locks

- Prevent multiple sessions from changing the same data at the same time
- Automatically obtained at the lowest possible level for a given statement



Transaction 1

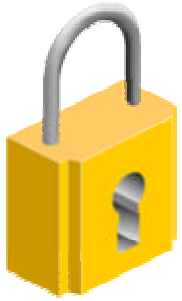
```
SQL> UPDATE hr.employees
  2  SET salary=salary+100
  3  WHERE employee_id=100;
```

Transaction 2

```
SQL> UPDATE hr.employees
  2  SET salary=salary*1.1
  3 WHERE employee_id=100;
```

Locking Mechanism

- High level of data concurrency
  - Row-level locks for inserts, updates, and deletes
  - No locks required for queries
- Automatic queue management
- Locks held until transaction ends (with commit or rollback operation)



Transaction 1

```
SQL> UPDATE hr.employees
  2  SET salary=salary+100
  3  WHERE employee_id=100;
```

Transaction 2

```
SQL> UPDATE hr.employees
  2  SET salary=salary*1.1
  3  WHERE employee_id=101;
```

Data Concurrency

Time:	Transaction 1	UPDATE hr.employees SET salary=salary+100 WHERE employee_id=100;
	Transaction 2	UPDATE hr.employees SET salary=salary+100 WHERE employee_id=101;
09:00:00	Transaction 3	UPDATE hr.employees SET salary=salary+100 WHERE employee_id=102;
	…	…
	Transaction *x*	UPDATE hr.employees SET salary=salary+100 WHERE employee_id=*xxx;*

DML Locks

Transaction 1

```
SQL> UPDATE employees
  2  SET salary=salary*1.1
  3  WHERE employee_id= 107;
1 row updated.
```

Transaction 2

```
SQL> UPDATE employees
  2  SET salary=salary*1.1
  3  WHERE employee_id= 106;
1 row updated.
```

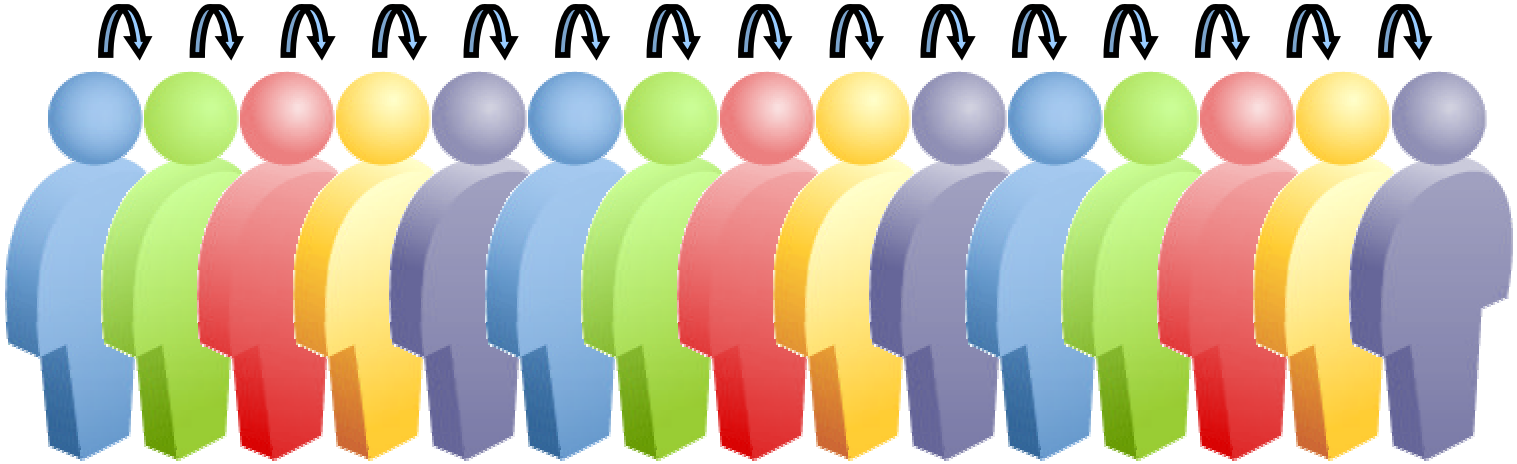
Each DML transaction must acquire *two* locks:

- **Row-exclusive lock for the row or rows being updated**
- **Shared table-level lock for the table containing the rows**

Enqueue Mechanism

The enqueue mechanism keeps track of:

- **Sessions waiting for locks**
- **The requested lock mode**
- **The order in which sessions requested the lock**



Lock Conflicts

Transaction 1	Time	Transaction 2
`UPDATE hr.employees SET salary=salary+100 WHERE employee_id=100;` `1 row updated.`	9:00:00	`UPDATE hr.employees SET salary=salary+100 WHERE employee_id=101;` `1 row updated.`

`UPDATE hr.employees SET COMMISION_PCT=2 WHERE employee_id=101;`
Session waits enqueue due to lock conflict.
	9:00:05 	`SELECT sum(salary) FROM hr.employees;`
		`SUM(SALARY)`
		`----------`
		`692634`

Session still waiting!	16:30:00	Many selects, inserts, updates, and deletes during the last 7.5 hours, but no commits or rollbacks!
`1 row updated.` Session continues.	16:30:01	`commit;`

Possible Causes of Lock Conflicts

- Uncommitted changes
- Long-running transactions
- Unnecessarily high locking levels

Detecting Lock Conflicts

Select Blocking Sessions from the Performance page.

Database: dba10g > Blocking Sessions

Blocking Sessions

Page Refreshed **Nov 25, 2003 6:53:21 PM**

View Session | Kill Session

Expand All | Collapse All

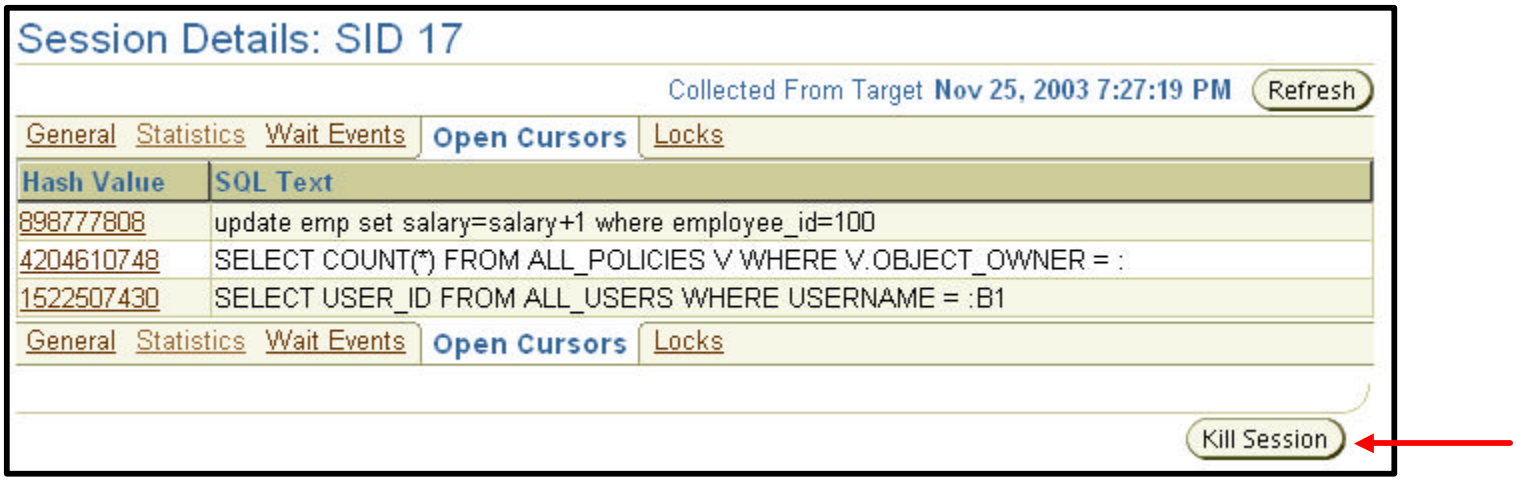
Select	Username	Sessions Blocked	Session ID	Session Serial Number	SQL Hash Value	Wait Class	Wait Event	P1	P2	P3	Seconds in Wait
	Blocking Sessions										
	HR	1	17	28225	0	Idle	SQL*Net message from client	1413697536	1	0	44
	HR	0	32	13948	898777808	Application	enq: TX - row lock contention	1415053318	5373958	39	35

Click the Session ID link to view information about the locking session, including the actual SQL statement.

Resolving Lock Conflicts

To resolve a lock conflict:

- **Have the session holding the lock commit or roll back.**
- **Terminate the session holding the lock as a last resort.**



Deadlocks

Transaction 1		Transaction 2

`UPDATE employees`
`SET salary = salary x 1.1`
`WHERE employee_id = 1000;`
	9:00	`UPDATE employees`
		`SET manager  = 1342`
		`WHERE employee_id = 2000;`

`UPDATE employees`
`SET salary = salary x 1.1`
`WHERE employee_id = 2000;`
	9:15	`UPDATE employees`
		`SET manager  = 1342`
		`WHERE employee_id = 1000;`

`ORA-00060:`
`Deadlock detected while`
`waiting for resource`
	9:16	

Summary

In this lesson you should have learned how to:

- **Detect and resolve lock conflicts**
- **Manage deadlocks**

Practice 17:
Locks in the Oracle Database

This practice covers common administrative tasks relating to locks in Oracle Database 10*g*, including:

- **Detecting which session is causing the locking conflict**
- **Resolving locking conflicts**