Database Systems
Transactions - Examples

Martin Docsek

FH University of Applied Sciences TECHNIKUM WIEN

DML Row Locks (TX Locks) - Demo

DEMO_LockSession1.sql – DEMO_LockSession2.sql

Session #1	Session #2

UPDATE person
SET vorname = 'Ronald‘
WHERE personid = 1;
	<= *sets exclusive transaction lock (TX)*
	set shared DML lock
	shared table lock (TM)

vorname returns unchanged name =>	SELECT vorname
	FROM person
	WHERE personid = 1;

requests locks
session hangs (is waiting), because
of exclusive lock from session #1 =>
	UPDATE person
	SET vorname = 'Ronald‘
	WHERE personid = 1;

ROLLBACK;	<= *commit or rollback release locks*

vorname returns changed name
from session #2 =>
	SELECT vorname
	FROM person
	WHERE personid = 1;

commit or rollback release locks =>	ROLLBACK;

DDL Locks - Demo

DEMO_DML_DDL_Session1.sql – DEMO_DML_DDL_Session2.sql

Session #1	Session #2

UPDATE person
SET vorname = 'Ronald‘
WHERE personid = 1;
	<= *sets exclusive transaction lock (TX)*
	set shared DML lock
	shared table lock (TM)

ERROR at line 1: *ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired* =>	DROP TABLE person;

Manual Data Locks

- It is possible to override Oracle’s default locking mechanisms

- Should be avoided unless there is a justified application requirement
  - Example: Transaction needs exclusive access to resource and must not wait for other transactions

- Manual row-level locks:
  - **SELECT … FOR UPDATE** statement

- Manual table-level locks:
  - **LOCK TABLE** statement

Manual Row Locks - Demo

DEMO_Manual_Row_Lock_Session1.sql – DEMO_Manual_Row_lock_Session2.sql

Session #1	Session #2

SELECT *
FROM person
WHERE personid = 2 <u>FOR UPDATE</u>;
	<= *sets row level locks*

1 row updated =>	UPDATE person
	SET vorname = 'Ronald‘
	WHERE personid = 1;

waiting – session is blocked =>	UPDATE person
	SET vorname = Ronny‘
	WHERE personid = 2;

Manual Table Locks - Demo

DEMO_Manual_Table_Lock_Session1.sql – DEMO_Manual_Table_lock_Session2.sql

Session #1	Session #2

SELECT *
FROM person
WHERE personid = 2 FOR UPDATE;
	<= *sets row level locks*

table is locked in row share mode =>	LOCK TABLE person IN ROW SHARE MODE;
waiting – session is blocked =>	LOCK TABLE person IN EXCLUSIVE MODE;
ROLLBACK;	<= *commit or rollback release locks*
table is locked in exclusive mode =>	
	ROLLBACK;

Deadlocks - Demo

DEMO_Deadlocks_Session1.sql – DEMO_Deadlocks_Session2.sql

Session #1	Session #2

UPDATE person
SET vorname = ‘Harry'
WHERE personid = 1;
	<= *sets row level locks*

sets row level locks =>	UPDATE person
	SET vorname = ‘Chris'
	WHERE personid = 2;

UPDATE person
SET vorname = ‘Chrissi'
WHERE personid = 2;
	<= *ORA-00060: deadlock detected*
	while waiting for resource
	statement has been rolled-back

waiting – session is blocked =>	UPDATE person
	SET vorname = ‘Harri'
	WHERE personid = 1;

ROLLBACK;	<= *commit or rollback release lockes*