Step 3

1)

Transaction 1	Transaction 2
CREATE OR REPLACE PROCEDURE INSERT_employee_isolation (emp# IN char, emp_name IN VARCHAR, date_of_birth IN date, sup# IN CHAR, dept_name IN VARCHAR) IS intial_emp_count NUMBER; intial_emp_count_corrupt Number; final_emp_count NUMBER; BEGIN SELECT total_emp INTO intial_emp_count FROM DEPARTMENT WHERE DName = dept_name;	
	SET TRANSACTION ISOLATION LEVEL READ COMMITTED; UPDATE DEPARTMENT SET total_emp = (SELECT select count(DName)- 2 from employee WHERE DEPARTMENT.DName = employee.DName group by DName); COMMIT;
SELECT total_emp INTO intial_emp_count_corrupt FROM DEPARTMENT WHERE DName = dept_name; SELECT (2 * intial_emp_count_corrupt) - intial_emp_count + 1 INTO final_emp_count FROM DUAL ; INSERT INTO employee VALUES(emp#, emp_name, date_of_birth, sup#, dept_name); UPDATE department SET total_emp = final_emp_count WHERE DName = dept_name;	

In any serial execution, the value of total employee must reflect the number of employees there are in the department.

After the update performed by transaction 2, the values of total number of employees has been reduced by 2, this maybe because employees left work and the employee records are being updated. However, when the 2nd transaction is processed concurrently with the first one. the new update is saved into the intial_emp_count_corrupt variable. The final_emp_count is calculated using the two variables intial_emp_count and intial_emp_count_corrupt. When no concurrent transactions take place at Isolation Level Read Committed, the final_emp_count calculated by the formula (2 * intial_emp_count_corrupt) - intial_emp_count + 1 produces correct output as intial_emp_count_corrupt and intial_emp_count are equal.

But when Transaction 2 happens concurrently, only the value of intial_emp_count_corrupt has changed. The final_emp_count calculated is corrupted as intial_emp_count_corrupt and intial_emp_count are not equal.

e.q. if there were 3 employees initially, then the transactions makes the following changes when no concurrent transaction happens at Read Committed Isolation level.

Transaction 1	Transaction 2
Actual employee count = 3 Initial_emp_count = 3 Initial_emp_count_corrupt = 3	
Actual employee count = 3 +1 = 4 Final_emp_count = (3*2) -3 +1 = 4	

The transactions make the following changes when a transaction 2 concurrently happens at Read Committed Isolation level and removes two employees:

Transaction 1	Transaction 2
Actual employee count = 3 Initial_emp_count = 3	
	Actual employee count = 3-2 = 1
Initial_emp_count_corrupt = 1 Initial_emp_count = 3 Final_emp_count = (2*1) – 3 +1 = 0 Actual employee count = 2	