Lab 09

Transaction Processing

(Source: taken from Prof S Sudarshan's[1] Assignments and modified to company schema)

IT214 Database Management System, Autumn'2018; pm_jat

In this lab, you experience, different isolation levels in SQL in general and postgreSQL in particular. Have two client windows connected to your postgresql server connected to company schema. Windows1 and Window2 refer to these clients respectively in following explanations.

Exercise-1

Window1: begin;
Window1: update employee set salary = 45000 where ssn = '123456789';
Window2: begin;
Window2: select * from employee where ssn = '123456789';

Look at the value of salary.
Can you figure out why you got the result that you saw?
What does this tell you about concurrency control in PostgreSQL?

Window1: commit;
Window2: select * from employee where ssn = '123456789';

Look at the value of salary again. Why is it so?

Windows2 now shows updated salary of the employee.
Window2: commit;

Exercise-2

Window1: begin;
Window1: update employee set salary = 45000 where ssn = '123456789';
Window2: begin;
Window2: update employee set salary = salary+3000 where ssn = '123456789';

Hold for a moment, and see result of this query

Window1: commit;
Window2: commit;
Window2: select salary from employee where ssn = '123456789';

Why do you get the result what you see? What if T1 aborts?

[1] http://www.cse.iitb.ac.in/~sudarsha/

Exercise-3

Window1: begin;
Window1: update employee set salary = 45000 where ssn = '123456789';
Window2: begin;
Window2: update employee set salary = (select min(salary) from employee where ssn = '123456789')+3000 where ssn = '123456789';

Hold for a moment, and see result of this query
Window1: commit;
Window2: commit;
Window2: select * from employee where ssn = '123456789';

Why do you get the result what you see?
What isolation level you observe?

Exercise-4

Set isolation to serializable by running following statement after begin in both windows, and rerun exercise-3 again
set transaction isolation level serializable;

Exercise-5

Window1: select ssn, salary from employee where ssn in('888665555', '123456789');
note the results
Window1: begin;
Window1: set transaction isolation level serializable;
Window2: begin;
Window2: set transaction isolation level serializable;

Window 1: update employee set salary = (select salary from employee where ssn = '888665555') where ssn = '123456789';

Window 2: update employee set salary = (select salary from employee where ssn = '123456789') where ssn = '888665555';

Window1: commit;
Window2: commit;

Window1: select ssn, salary from employee where ssn in('888665555', '123456789');

Compare the results. Is this equivalent to any serializable schedule?

What is wrong, and what is solution?

Now, just before the update statement in window 1, add the line

select salary from employee where ssn = '888665555' for update;

Do you see any change?