

## CS186 Week 10 - Discussion Notes

**Compare locking and optimistic concurrency control (OCC). List two advantages of locking and two advantages of OCC.**

**Thinking about race conditions: What can query results can transaction 2 return if the two transactions below are executed concurrently and we didn't make isolation guarantees?**

### **Transaction 1**

```
INSERT INTO ratings VALUES('larry', 9.5);  
INSERT INTO ratings VALUES('kevin', 2.5);  
INSERT INTO ratings VALUES('george', 2.3);
```

### **Transaction 2**

```
SELECT MIN(rating) FROM ratings;  
SELECT MAX(rating) FROM ratings;  
SELECT COUNT(*) FROM ratings WHERE rating=9.5;
```

### **Ratings**

<b>Name</b>	<b>Rating</b>
'John'	9.4
'Foo'	4.3
'Bar'	2.8
'Eric'	9.4
'Jack'	7.7

**What's one case where snapshot isolation would not guarantee serializability?**

**What values can the program below print? How can we modify the code to make it deterministic?**

```
public class TestConcurrent {
    public static void main(String[] args) {
        Incrementer incr = new Incrementer();
        Thread t1 = new Thread(incr);
        Thread t2 = new Thread(incr);
        t1.start();
        t2.start();
        try {
            t1.join();
            t2.join();
        } catch (InterruptedException e) {
            System.err.println("Oops something wrong");
        }
        System.out.println(incr.count);
    }

    private static class Incrementer implements Runnable {
        int count = 0;
        public void run() {
            for (int i = 0; i < 3; i++) {
                increment();
            }
        }
        public void increment() {
            count = count + 1;
        }
    }
}
```