**CSCI235: Database Systems**

**Assignment 2**

**Nicholas Monteleone 5055076**

**Task 4**



|  |  |
| --- | --- |
| Transaction 1 | Transaction 2 |
|  | SELECT NVL(MAX(unit\_price), 0)  INTO max\_unit\_price  FROM PRODUCT; |
| UPDATE PRODUCT  SET unit\_price = unit\_price + 0.1\*unit\_price  WHERE product\_name = product\_plus; |  |
|  | UPDATE PRODUCT  SET unit\_price = unit\_price + 0.01\*max\_unit\_price  WHERE units\_in\_stock > 60; |
| UPDATE PRODUCT  SET unit\_price = unit\_price - 0.1\*unit\_price  WHERE product\_name = product\_minus; |  |
|  | UPDATE PRODUCT  SET unit\_price = unit\_price + 0.02\*max\_unit\_price  WHERE units\_in\_stock <= 60; |
|  | COMMIT; |
| COMMIT; |  |

Above schedule shows concurrent processing of the procedure from Task 3 (assumed to be changed to serializable isolation level) and the procedure from Task 4. With a serializable isolation level this function fails due to an update being done to the unit\_price variable by T2, causing a conflict with T1.

This will result in a failure of both transactions as they both are both writing to the same variable before the other transaction is completed and committed.