**Assignment 4**

Name: Jain Samkitkumar Hasmukhlal

Class: TEIT Roll. No.: 20 Batch: T1

Date:31/8/2015

Implement any one Concurrency Control Protocol using MongoDB and .net/Java 5

Microsoft Windows [Version 6.3.9600]

(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\STUDENT>cd C:\Program Files\MongoDB 2.6 Standard\bin

C:\Program Files\MongoDB 2.6 Standard\bin>mongo

MongoDB shell version: 2.6.2

connecting to: test

Server has startup warnings:

2014-09-23T11:53:20.386+0530 [initandlisten]

2014-09-23T11:53:20.386+0530 [initandlisten] \*\* NOTE: This is a 32 bit MongoDB b

inary.

2014-09-23T11:53:20.386+0530 [initandlisten] \*\* 32 bit builds are limited

to less than 2GB of data (or less with --journal).

2014-09-23T11:53:20.386+0530 [initandlisten] \*\* Note that journaling defau

lts to off for 32 bit and is currently off.

2014-09-23T11:53:20.386+0530 [initandlisten] \*\* See http://dochub.mongodb.

org/core/32bit

2014-09-23T11:53:20.387+0530 [initandlisten]

> show databases;

Library 0.078GB

admin (empty)

local 0.078GB

> use Library

switched to db Library

> db.createCollection("accounts")

{ "ok" : 1 }

**Initialize Source and Destination Accounts**

> db.accounts.insert(

... [

... { \_id:"A", balance:1000, pendingTransactions: [] },

... { \_id:"B", balance:1000, pendingTransactions: [] }

... ]

... )

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 2,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

> db.accounts.find()

{ "\_id" : "A", "balance" : 1000, "pendingTransactions" : [ ] }

{ "\_id" : "B", "balance" : 1000, "pendingTransactions" : [ ] }

**Initialize Transfer Record**

> db.transactions.insert( { \_id:1, source:"A", destination:"B", value:100, state

:"initial", lastModified:new Date() } )

WriteResult({ "nInserted" : 1 })

**Step 1: Transfer Funds Between Accounts Using Two-Phase Commit**

> var t =db.transactions.findOne( { state:"initial" } )

> t

{

"\_id" : 1,

"source" : "A",

"destination" : "B",

"value" : 100,

"state" : "initial",

"lastModified" : ISODate("2014-09-23T06:37:14.348Z")

}

**Step 2: Update transaction state to pending.**

> db.transactions.update(

... { \_id:t.\_id, state:"initial" },

... {

... $set:{ state:"pending" },

... $currentDate:{ lastModified:true }

... }

... )

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

**Step 3:** **Apply the transaction to both accounts.**

> db.accounts.update(

... { \_id:t.source, pendingTransactions: { $ne:t.\_id } },

... { $inc: { balance:-t.value }, $push: { pendingTransactions:t.\_id } }

... )

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.accounts.update(

... { \_id:t.destination, pendingTransactions: { $ne:t.\_id } },

... { $inc: { balance:t.value }, $push: { pendingTransactions:t.\_id } }

... )

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

**Step 4:** **Update transaction state to applied.**

> db.transactions.update(

... { \_id:t.\_id, state:"pending" },

... {

... $set:{ state:"applied" },

... $currentDate:{ lastModified:true }

... }

... )

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

**Step 5:** **Update both accounts’ list of pending transactions.**

> db.accounts.update(

... { \_id:t.source, pendingTransactions:t.\_id },

... { $pull: { pendingTransactions:t.\_id } }

... )

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.accounts.update(

... { \_id:t.destination, pendingTransactions:t.\_id },

... { $pull: { pendingTransactions:t.\_id } }

... )

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

>

**Step 6:** **Update transaction state to done.**

> db.transactions.update(

... { \_id:t.\_id, state:"applied" },

... {

... $set:{ state:"done" },

... $currentDate:{ lastModified:true }

... }

... )

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.accounts.find();

{ "\_id" : "A", "balance" : 900, "pendingTransactions" : [ ] }

{ "\_id" : "B", "balance" : 1100, "pendingTransactions" : [ ] }

>