Server Setup Cheat sheet

Preconditions:

User must have permission to execute PowerShell scripts.

Java must be installed with Environment Variables:

JAVA\_HOME C:\Program Files\Java\jre-10.0.1

PATH ;C:\Program Files\Java\jre-10.0.1\bin\server

Download

neo4j-community-3.4.0-windows.zip

Create Operating Folder

C:\graph

Copy in Zip contents

Start Server from CMD Run as Administrator

cd /.

cd graph/bin

neo4j console

Open Browser

localhost:7474

Username: neo4j

Password: neo4j

//Add a deal node (edit and repeat for deal 2 and 3) Note: Entities:PascalCase, Properties:camelCase  
CREATE (n:Deal {dealId:"1", dealName:"SQ101"}) edit and repeat for 2, 131 and 3, 132

//Add loan nodes   
CREATE (n1:Loan {loanId: "1", dealId: "1", maturityDate: "2026-07-01", originalAmount: toInteger("243546")})  
CREATE (n2:Loan {loanId: "2", dealId: "1", maturityDate: "2031-02-01", originalAmount: toInteger("465764")})  
CREATE (n3:Loan {loanId: "3", dealId: "1", maturityDate: "2035-03-01", originalAmount: toInteger("354657")})  
CREATE (n4:Loan {loanId: "4", dealId: "1", maturityDate: "2029-06-01", originalAmount: toInteger("645342")})  
CREATE (n5:Loan {loanId: "5", dealId: "1", maturityDate: "2027-11-01", originalAmount: toInteger("534231")})  
CREATE (n6:Loan {loanId: "6", dealId: "2", maturityDate: "2034-12-01", originalAmount: toInteger("523421")})  
CREATE (n7:Loan {loanId: "7", dealId: "2", maturityDate: "2033-11-01", originalAmount: toInteger("445231")})  
CREATE (n8:Loan {loanId: "8", dealId: "2", maturityDate: "2037-09-01", originalAmount: toInteger("601201")})

//Add LoanPeriod nodes  
CREATE (n1:LoanPeriod {periodId:"1", loanId:"1", balance:241844.00, delinquencyStatus:0})

CREATE (n2:LoanPeriod {periodId:"1", loanId:"2", balance:463521.00, delinquencyStatus:0})

CREATE (n3:LoanPeriod {periodId:"1", loanId:"3", balance:352657.00, delinquencyStatus:0})

CREATE (n4:LoanPeriod {periodId:"1", loanId:"4", balance:645342.00, delinquencyStatus:3})

CREATE (n5:LoanPeriod {periodId:"1", loanId:"5", balance:531210.00, delinquencyStatus:0})

CREATE (n6:LoanPeriod {periodId:"1", loanId:"6", balance:521201.00, delinquencyStatus:0})

CREATE (n7:LoanPeriod {periodId:"1", loanId:"7", balance:443231.00, delinquencyStatus:0})

CREATE (n8:LoanPeriod {periodId:"1", loanId:"8", balance:601201.00, delinquencyStatus:1})

CREATE (n9:LoanPeriod {periodId:"2", loanId:"1", balance:241844.00, delinquencyStatus:1})

CREATE (n10:LoanPeriod {periodId:"2", loanId:"2", balance:451321.00, delinquencyStatus:0})

CREATE (n11:LoanPeriod {periodId:"2", loanId:"3", balance:350543.00, delinquencyStatus:0})

CREATE (n12:LoanPeriod {periodId:"2", loanId:"4", balance:645342.00, delinquencyStatus:4})

CREATE (n13:LoanPeriod {periodId:"2", loanId:"5", balance:529321.00, delinquencyStatus:0})

CREATE (n14:LoanPeriod {periodId:"2", loanId:"6", balance:519321.00, delinquencyStatus:0})

CREATE (n15:LoanPeriod {periodId:"2", loanId:"7", balance:441209.00, delinquencyStatus:0})

CREATE (n16:LoanPeriod {periodId:"2", loanId:"8", balance:601201.00, delinquencyStatus:2})

//Add Deal:Loan relationships  
MATCH (d:Deal), (l:Loan) WHERE d.dealId = l.dealId CREATE (d)-[:CONTAINS]->(l)

//Add Loan:LoanPeriod relationships  
MATCH (l:Loan), (p:LoanPeriod) WHERE l.loanId = p.loanId CREATE (l)-[:HAS\_BALANCE]->(p)

//Returns Average balance of Deals in Period 2  
MATCH (d:Deal)-[:CONTAINS]->(l:Loan)-[:HAS\_BALANCE]->(p:LoanPeriod) WHERE p.periodId = "2"   
RETURN d.dealName AS `Deal`, AVG(p.balance) AS `Average balance`

//Returns list of loans with change in delinquencyStatus from Period 1 to Period2  
MATCH (p1:LoanPeriod), (p2:LoanPeriod)  
WHERE p1.loanId = p2.loanId  
AND p1.periodId = "1"  
AND p2.periodId = "2"  
AND p1.delinquencyStatus <> p2.delinquencyStatus  
RETURN p2.loanId  
ORDER BY p2.loanId