

Use Case 3

BookShop

Use Cases Description

A chain of bookstores requires to develop a data intensive application that can contain book sales transactions. Some of these transactions have been reported by customers as unknown or fraudulent.

There are millions of transactions per day and a customer can buy on different book stores on the same day.

In order to implement a database, suppose there are 16 bookstores, 10 customers, each customer can has bought on at least three different stores. The database contains a history of 60 transactions, 16 of them are fraudulent and have been marked as “complained” the rest are normal transactions or no complained.

Companies lose billions of dollars every year to credit card fraud. Credit card data can be stolen by criminals using a variety of methods. For instance, Bluetooth enabled data skimming devices can be placed on the card reader. The data might be stolen in a mass breach by hackers of a large retailer. Sometimes the thief is simply the clerk at the checkout line at the bookstore, where the victim's card is swiped through a small device or surreptitiously jotted down.

Design a data intensive application that can help to detect frauds with simple queries like:

- a) Obtain all customers that have complained about fraudulent transactions or books they have not actually bought, show which customers and which Bookshops are involved in the fraud cases
- b) The bookstore chain not only want the illegitimate transactions but also the transactions happening before the theft.
- c) Find the common Bookshop in all these fraudulent transactions

What solution would you bring to the company?

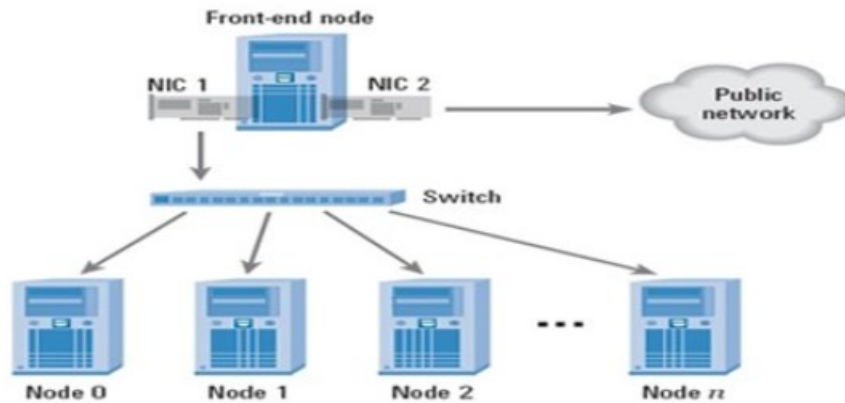
The student should submit its proposal and justification of the following elements:

1. Typical architecture according to the type of information system
2. Kind of database to implement and its design
3. Answers to the queries required
4. How does it support scalability
5. How does it support maintainability
6. How does it support security and reliability

Solution

1. Typical architecture according to the type of information system

The best architecture would be a distributed parallel cluster environment.



2. Kind of database to implement and its design

We have following key requirements here: 1) ACID transactions 2) NoSQL Database

Thus, would suggest a NoSQL graph database. The characteristics of the graph data model:

- By representing transactions as a graph, we can look for the common denominator in the fraud cases and find the point of origin of the scam.
- A series of credit card transactions can be represented as a graph.
- Each transaction involves two nodes: a Customer and a Bookshop.
- The nodes are linked by the transaction itself.
- A transaction has a date and a status.
- Legitimate transactions have the status "UnComplained". Fraudulent transactions are "Complained".

3. Answers to the queries required

The Database, queries and results are shown below.

4. How does it support scalability

Scalability can be easily achieved by adding a new node to the cluster.

5. How does it support maintainability

As it is a graph, it is flexible to changes, thus supports easier maintainability.

6. How does it support security and reliability

The Neo4j database supports ACID properties and authentication.

Book Shop Database:

```
// Create customers
CREATE (Peter:Customer {id:'1', name:'Peter', gender:'man', age:'50'})
CREATE (Joanny:Customer {id:'2', name:'Joanny', gender:'man', age:'48'})
CREATE (Dan:Customer {id:'3', name:'Dan', gender:'man', age:'23'})
CREATE (Michael:Customer {id:'4', name:'Michael', gender:'man', age:'30'})
CREATE (Frederic:Customer {id:'5', name:'Frederic', gender:'man', age:'31'})
CREATE (Mariana:Customer {id:'6', name:'Mariana', gender:'woman', age:'52'})
CREATE (Louisa:Customer {id:'7', name:'Louisa', gender:'woman', age:'23'})
CREATE (Maria:Customer {id:'8', name:'Maria', gender:'woman', age:'58'})
CREATE (Maryleen:Customer {id:'9', name:'Maryleen', gender:'woman', age:'51'})
CREATE (Rose:Customer {id:'10', name:'Rose', gender:'woman', age:'37'})

// Create Bookshops
CREATE (JOANNE_HENDRICKS_COOKBOOKS:Bookshop {id:'11',
name:'JOANNE_HENDRICKS_COOKBOOKS', street:'2626 Wilkinson Court',
address:'Saint Paul, CA 92410'})
CREATE (FAULKNERHOUSE:Bookshop {id:'12', name:'FAULKNERHOUSE',
street:'4355 Walnut Street', age:'Saint Paul, CA 92410'})
CREATE (BARTS_BOOKS:Bookshop {id:'13', name:'BARTS_BOOKS',
street:'2092 Larry Street', age:'Saint Paul, CA 92410'})
CREATE (SPOTTY_DOG_BOOKS_AND_ALE:Bookshop {id:'14',
name:'SPOTTY_DOG_BOOKS_AND_ALE', street:'1870 Caynor Circle', age:'Saint
Paul, CA 92410'})
CREATE (POLITICS_AND_PROSE:Bookshop {id:'15', name:'
POLITICS_AND_PROSE', street:'1381 Spruce Drive', age:'Saint Paul, CA 92410'})
CREATE (POLITICS_BOOKS:Bookshop {id:'16', name:' POLITICS_BOOKS',
street:'826 Anmoore Road', age:'Saint Paul, CA 92410'})
CREATE (THE_LAST_BOOKSTORE:Bookshop {id:'17',
name:'THE_LAST_BOOKSTORE', street:'1925 Spring Street', age:'Saint Paul, CA
92410'})
CREATE (BRATTLE_BOOK_SHOP:Bookshop {id:'18',
name:'BRATTLE_BOOK_SHOP', street:'4209 Elsie Drive', age:'Saint Paul, CA
92410'})
CREATE (BOOKS_FOR_THE_CITY:Bookshop {id:'19', name:'
BOOKS_FOR_THE_CITY ', street:'86 D Street', age:'Saint Paul, CA 92410'})
CREATE (JOHN_K_KING_USED_AND_RARE:Bookshop {id:'20',
name:'JOHN_K_KING_USED_AND_RARE', street:'945 Kinney Street', age:'Saint
Paul, CA 92410'})
CREATE (BEST_BOOKS_FOREVER:Bookshop {id:'21',
name:'BEST_BOOKS_FOREVER', street:'3810 River Lane', age:'Saint Paul, CA
92410'})
CREATE (OREALLY:Bookshop {id:'22', name:'OREALLY', street:'3778 Tenmile
Road', age:'Saint Paul, CA 92410'})
CREATE (WATERSTONES:Bookshop {id:'23', name:' WATERSTONES ',
street:'349 Bel Meadow Drive', age:'K Rivers, MO 64105'})
CREATE (MY_BEST_BOOKS:Bookshop {id:'24', name:' MY_BEST_BOOKS',
street:'99 Strother Street', age:'K Rivers, MO 64105'})
CREATE (OLDIES_BUT_GOODIES:Bookshop {id:'25',
name:'OLDIES_BUT_GOODIES', street:'3306 Douglas Dairy Road', age:'K Rivers,
```

```

MO 64105'})
CREATE (RARE_BOOKS:Bookshop {id:'26', name:'RARE_BOOKS', street:'2912
Nutter Street', age:'K Rivers, MO 64105'})
// Create transaction history
CREATE (Peter)-[:HAS_BOUGHT_AT {salesalesamount:'986.50',
time:'4/17/2018', status:'UnComplained'}}->(POLITICS_AND_PROSE)
CREATE (Peter)-[:HAS_BOUGHT_AT {salesalesamount:'239.99',
time:'5/15/2018', status:'UnComplained'}}->(BEST_BOOKS_FOREVER)
CREATE (Peter)-[:HAS_BOUGHT_AT {salesalesamount:'475.55',
time:'3/28/2018', status:'UnComplained'}}->(BRATTLE_BOOK_SHOP)
CREATE (Peter)-[:HAS_BOUGHT_AT {salesalesamount:'654.00',
time:'3/20/2018', status:'UnComplained'}}->(BARTS_BOOKS)
CREATE (Joanny)-[:HAS_BOUGHT_AT {salesalesamount:'196.75',
time:'7/24/2018', status:'UnComplained'}}->(BOOKS_FOR_THE_CITY)
CREATE (Joanny)-[:HAS_BOUGHT_AT {salesalesamount:'502.50',
time:'4/9/2018', status:'UnComplained'}}->(FAULKNERHOUSE)
CREATE (Joanny)-[:HAS_BOUGHT_AT {salesalesamount:'848.00',
time:'5/29/2018', status:'UnComplained'}}->(BARTS_BOOKS)
CREATE (Joanny)-[:HAS_BOUGHT_AT {salesalesamount:'802.30',
time:'3/11/2018', status:'UnComplained'}}-
>(JOANNE_HENDRICKS_COOKBOOKS)
CREATE (Joanny)-[:HAS_BOUGHT_AT {salesalesamount:'203.34',
time:'3/27/2018', status:'UnComplained'}}->(OREALLY)
CREATE (Dan)-[:HAS_BOUGHT_AT {salesalesamount:'35.20', time:'1/23/2018',
status:'UnComplained'}}->(SPOTTY_DOG_BOOKS_AND_ALE)
CREATE (Dan)-[:HAS_BOUGHT_AT {salesalesamount:'605', time:'1/27/2018',
status:'UnComplained'}}->(SPOTTY_DOG_BOOKS_AND_ALE)
CREATE (Dan)-[:HAS_BOUGHT_AT {salesamount:'62.60', time:'9/17/2018',
status:'UnComplained'}}->(BOOKS_FOR_THE_CITY)
CREATE (Dan)-[:HAS_BOUGHT_AT {salesamount:'141.45', time:'11/14/2018',
status:'UnComplained'}}->(JOANNE_HENDRICKS_COOKBOOKS)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'134.00', time:'4/14/2018',
status:'UnComplained'}}->(JOANNE_HENDRICKS_COOKBOOKS)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'336.45', time:'4/3/2018',
status:'UnComplained'}}->(MYOPIC_BOOKS)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'964.50', time:'3/22/2018',
status:'UnComplained'}}->(BARTS_BOOKS)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'430.00', time:'8/10/2018',
status:'UnComplained'}}->(BRATTLE_BOOK_SHOP)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'11.00', time:'9/4/2018',
status:'UnComplained'}}->(BOOKS_FOR_THE_CITY)
CREATE (Frederic)-[:HAS_BOUGHT_AT {salesamount:'545.00', time:'10/6/2018',
status:'UnComplained'}}->(BOOKS_FOR_THE_CITY)
CREATE (Frederic)-[:HAS_BOUGHT_AT {salesamount:'457.50', time:'10/15/2018',
status:'UnComplained'}}->(JOHN_K_KING_USED_AND_RARE)
CREATE (Frederic)-[:HAS_BOUGHT_AT {salesamount:'468.00', time:'7/29/2018',
status:'UnComplained'}}->(THE_LAST_BOOKSTORE)
CREATE (Frederic)-[:HAS_BOUGHT_AT {salesamount:'768.50', time:'11/28/2018',
status:'UnComplained'}}->(MYOPIC_BOOKS)
CREATE (Frederic)-[:HAS_BOUGHT_AT {salesamount:'921.00', time:'3/12/2018',
status:'UnComplained'}}->(POLITICS_AND_PROSE)

```

CREATE (Mariana)-[:HAS_BOUGHT_AT {salesamount:'740.50', time:'12/15/2018', status:'UnComplained'}]->(SPOTTY_DOG_BOOKS_AND_ALE)

CREATE (Mariana)-[:HAS_BOUGHT_AT {salesamount:'510.00', time:'11/27/2018', status:'UnComplained'}]->(FAULKNERHOUSE)

CREATE (Mariana)-[:HAS_BOUGHT_AT {salesamount:'414.50', time:'1/20/2018', status:'UnComplained'}]->(POLITICS_AND_PROSE)

CREATE (Mariana)-[:HAS_BOUGHT_AT {salesamount:'721.50', time:'7/17/2018', status:'UnComplained'}]->(JOANNE_HENDRICKS_COOKBOOKS)

CREATE (Mariana)-[:HAS_BOUGHT_AT {salesamount:'353.00', time:'10/25/2018', status:'UnComplained'}]->(OREALLY)

CREATE (Louisa)-[:HAS_BOUGHT_AT {salesamount:'681.00', time:'12/28/2018', status:'UnComplained'}]->(BRATTLE_BOOK_SHOP)

CREATE (Louisa)-[:HAS_BOUGHT_AT {salesamount:'87.50', time:'2/19/2018', status:'UnComplained'}]->(BARTS_BOOKS)

CREATE (Louisa)-[:HAS_BOUGHT_AT {salesamount:'533.00', time:'8/6/2018', status:'UnComplained'}]->(MYOPIC_BOOKS)

CREATE (Louisa)-[:HAS_BOUGHT_AT {salesamount:'723.00', time:'1/8/2018', status:'UnComplained'}]->(MYOPIC_BOOKS)

CREATE (Louisa)-[:HAS_BOUGHT_AT {salesamount:'627.00', time:'5/20/2018', status:'UnComplained'}]->(POLITICS_AND_PROSE)

CREATE (Maria)-[:HAS_BOUGHT_AT {salesamount:'74.00', time:'9/4/2018', status:'UnComplained'}]->(BOOKS_FOR_THE_CITY)

CREATE (Maria)-[:HAS_BOUGHT_AT {salesamount:'231.00', time:'7/12/2018', status:'UnComplained'}]->(BARTS_BOOKS)

CREATE (Maria)-[:HAS_BOUGHT_AT {salesamount:'924.00', time:'10/4/2018', status:'UnComplained'}]->(BOOKS_FOR_THE_CITY)

CREATE (Maria)-[:HAS_BOUGHT_AT {salesamount:'742.00', time:'8/12/2018', status:'UnComplained'}]->(POLITICS_AND_PROSE)

CREATE (Maryleen)-[:HAS_BOUGHT_AT {salesamount:'276.00', time:'12/24/2018', status:'UnComplained'}]->(BOOKS_FOR_THE_CITY)

CREATE (Maryleen)-[:HAS_BOUGHT_AT {salesamount:'66.40', time:'4/16/2018', status:'UnComplained'}]->(BEST_BOOKS_FOREVER)

CREATE (Maryleen)-[:HAS_BOUGHT_AT {salesamount:'467.40', time:'12/23/2018', status:'UnComplained'}]->(SPOTTY_DOG_BOOKS_AND_ALE)

CREATE (Maryleen)-[:HAS_BOUGHT_AT {salesamount:'830.40', time:'3/13/2018', status:'UnComplained'}]->(BRATTLE_BOOK_SHOP)

CREATE (Maryleen)-[:HAS_BOUGHT_AT {salesamount:'240.40', time:'7/9/2018', status:'UnComplained'}]->(JOANNE_HENDRICKS_COOKBOOKS)

CREATE (Maryleen)-[:HAS_BOUGHT_AT {salesamount:'164.50', time:'12/26/2018', status:'UnComplained'}]->(BOOKS_FOR_THE_CITY)

CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'630.50', time:'10/6/2018', status:'UnComplained'}]->(SPOTTY_DOG_BOOKS_AND_ALE)

CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'19.50', time:'7/29/2018', status:'UnComplained'}]->(FAULKNERHOUSE)

CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'352.50', time:'12/16/2018', status:'UnComplained'}]->(OREALLY)

CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'147.50', time:'8/3/2018', status:'UnComplained'}]->(JOANNE_HENDRICKS_COOKBOOKS)

CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'91.50', time:'6/29/2018', status:'UnComplained'}]->(BARTS_BOOKS)

CREATE (Peter)-[:HAS_BOUGHT_AT {salesamount:'1021.50', time:'7/18/2018',

```

status:'Complained'}}->(WATERSTONES)
CREATE (Peter)-[:HAS_BOUGHT_AT {salesamount:'1732.50', time:'5/10/2018',
status:'Complained'}}->(MY_BEST_BOOKS)
CREATE (Peter)-[:HAS_BOUGHT_AT {salesamount:'1415.50', time:'4/1/2018',
status:'Complained'}}->(OLDIES_BUT_GOODIES)
CREATE (Peter)-[:HAS_BOUGHT_AT {salesamount:'1849.50', time:'12/20/2018',
status:'Complained'}}->(RARE_BOOKS)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'1914.50', time:'7/18/2018',
status:'Complained'}}->(WATERSTONES)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'1424.50', time:'5/10/2018',
status:'Complained'}}->(MY_BEST_BOOKS)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'1721.50', time:'4/1/2018',
status:'Complained'}}->(OLDIES_BUT_GOODIES)
CREATE (Michael)-[:HAS_BOUGHT_AT {salesamount:'1003.50',
time:'12/20/2018', status:'Complained'}}->(RARE_BOOKS)
CREATE (Maria)-[:HAS_BOUGHT_AT {salesamount:'1149.50', time:'7/18/2018',
status:'Complained'}}->(WATERSTONES)
CREATE (Maria)-[:HAS_BOUGHT_AT {salesamount:'1152.50', time:'8/10/2018',
status:'Complained'}}->(MY_BEST_BOOKS)
CREATE (Maria)-[:HAS_BOUGHT_AT {salesamount:'1884', time:'8/1/2018',
status:'Complained'}}->(OLDIES_BUT_GOODIES)
CREATE (Maria)-[:HAS_BOUGHT_AT {salesamount:'1790.00', time:'12/20/2018',
status:'Complained'}}->(RARE_BOOKS)
CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'1925.00', time:'7/18/2018',
status:'Complained'}}->(WATERSTONES)
CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'1374.00', time:'7/10/2018',
status:'Complained'}}->(MY_BEST_BOOKS)
CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'1368.00', time:'7/1/2018',
status:'Complained'}}->(OLDIES_BUT_GOODIES)
CREATE (Rose)-[:HAS_BOUGHT_AT {salesamount:'1816.00', time:'12/20/2018',
status:'Complained'}}->(RARE_BOOKS)

```

The screenshot shows a code editor window with a light blue background. The editor contains 8 lines of Cypher queries for creating customers. To the right of the code is a toolbar with icons for search, undo, redo, and close. Below the code editor, a status bar displays the message: "Added 26 labels, created 27 nodes, set 296 properties, created 64 relationships, completed after 387 ms." On the left side of the editor, there is a sidebar with two icons: a table icon labeled "Table" and a code icon labeled "Code".

```

1 // Create customers
2 CREATE (Peter:Customer {id:'1', name:'Peter', gender:'man', age:'50'})
3 CREATE (Joanny:Customer {id:'2', name:'Joanny', gender:'man', age:'48'})
4 CREATE (Dan:Customer {id:'3', name:'Dan', gender:'man', age:'23'})
5 CREATE (Michael:Customer {id:'4', name:'Michael', gender:'man',
  age:'30'})
6 CREATE (Frederic:Customer {id:'5', name:'Frederic', gender:'man',
  age:'31'})
7 CREATE (Mariana:Customer {id:'6', name:'Mariana', gender:'woman',
  age:'52'})
8 CREATE (Louisa:Customer {id:'7', name:'Louisa', gender:'woman',
  age:'23'})

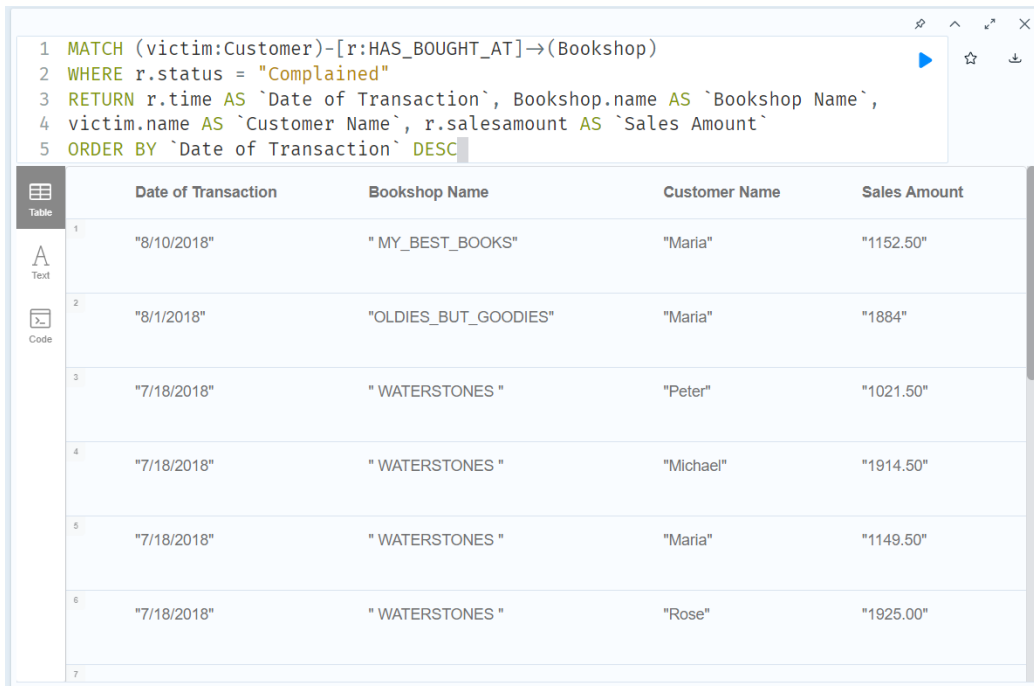
```

Added 26 labels, created 27 nodes, set 296 properties, created 64 relationships, completed after 387 ms.

Queries:

a) Obtain all customers that have complained about fraudulent transactions or books they have not actually bought, show which customers and which Bookshops are involved in the fraud cases

```
MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]->(Bookshop)
WHERE r.status = "Complained"
RETURN r.time AS `Date of Transaction`, Bookshop.name AS `Bookshop Name`,
victim.name AS `Customer Name`, r.salesamount AS `Sales Amount`
ORDER BY `Date of Transaction` DESC
```



The screenshot shows a query execution interface. At the top, a text area contains the following Cypher query:

```
1 MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]->(Bookshop)
2 WHERE r.status = "Complained"
3 RETURN r.time AS `Date of Transaction`, Bookshop.name AS `Bookshop Name`,
4 victim.name AS `Customer Name`, r.salesamount AS `Sales Amount`
5 ORDER BY `Date of Transaction` DESC
```

Below the query, a table displays the results. The table has five columns: "Date of Transaction", "Bookshop Name", "Customer Name", and "Sales Amount". The results are ordered by "Date of Transaction" in descending order.

	Date of Transaction	Bookshop Name	Customer Name	Sales Amount
1	"8/10/2018"	" MY_BEST_BOOKS"	"Maria"	"1152.50"
2	"8/1/2018"	"OLDIES_BUT_GOODIES"	"Maria"	"1884"
3	"7/18/2018"	" WATERSTONES "	"Peter"	"1021.50"
4	"7/18/2018"	" WATERSTONES "	"Michael"	"1914.50"
5	"7/18/2018"	" WATERSTONES "	"Maria"	"1149.50"
6	"7/18/2018"	" WATERSTONES "	"Rose"	"1925.00"
7				

b) The bookstore chain not only want the illegitimate transactions but also the transactions happening before the theft.

```
MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]->(Bookshop)
WHERE r.status = 'Complained'
MATCH (victim)-[t:HAS_BOUGHT_AT]->(otherBookshops)
WHERE t.status = 'UnComplained' AND t.time < r.time
WITH victim, otherBookshops, t ORDER BY t.time DESC
RETURN t.time AS `Date of Transaction`, otherBookshops.name AS `Book shop`, victim.name AS `Customer Name`, t.salesamount AS `Sales Amount`, t.status
ORDER BY `Date of Transaction` DESC
```

neo4j\$ MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]→(Bookshop) WHERE r.status =...

	Date of Transaction	Book shop	Customer Name	Sales Amount	t.status
1	"7/12/2018"	"BARTS_BOOKS"	"Maria"	"231.00"	"UnComplained"
2	"7/12/2018"	"BARTS_BOOKS"	"Maria"	"231.00"	"UnComplained"
3	"7/12/2018"	"BARTS_BOOKS"	"Maria"	"231.00"	"UnComplained"
4	"6/29/2018"	"BARTS_BOOKS"	"Rose"	"91.50"	"UnComplained"
5	"6/29/2018"	"BARTS_BOOKS"	"Rose"	"91.50"	"UnComplained"
6	"6/29/2018"	"BARTS_BOOKS"	"Rose"	"91.50"	"UnComplained"
7					

Started streaming 34 records after 2 ms and completed after 7 ms.

c) Find the common Bookshop in all these fraudulent transactions

```

MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]→(Bookshop)
WHERE r.status = "Complained"
MATCH (victim)-[t:HAS_BOUGHT_AT]→(otherBookshops)
WHERE t.status = "UnComplained" AND t.time < r.time
WITH (victim), otherBookshops, t ORDER BY t.time DESC
RETURN DISTINCT otherBookshops.name AS `Suspicious Store`,
count(DISTINCT t) AS Count, collect(DISTINCT victim.name) AS Victims
ORDER BY Count DESC

```

neo4j\$ MATCH (victim:Customer)-[r:HAS_BOUGHT_AT]→(Bookshop) WHERE r.status =...

	Suspicious Store	Count	Victims
1	"BARTS_BOOKS"	4	["Maria", "Rose", "Michael", "Peter"]
2	"BEST_BOOKS_FOREVER"	1	["Peter"]
3	null	1	["Michael"]
4	"POLITICS_AND_PROSE"	1	["Peter"]
5	"JOANNE_HENDRICKS_COOKBOOKS"	1	["Michael"]
6	"BRATTLE_BOOK_SHOP"	1	["Peter"]
7			

Started streaming 9 records after 20 ms and completed after 33 ms.