# The Yacht Club Example (model 2)

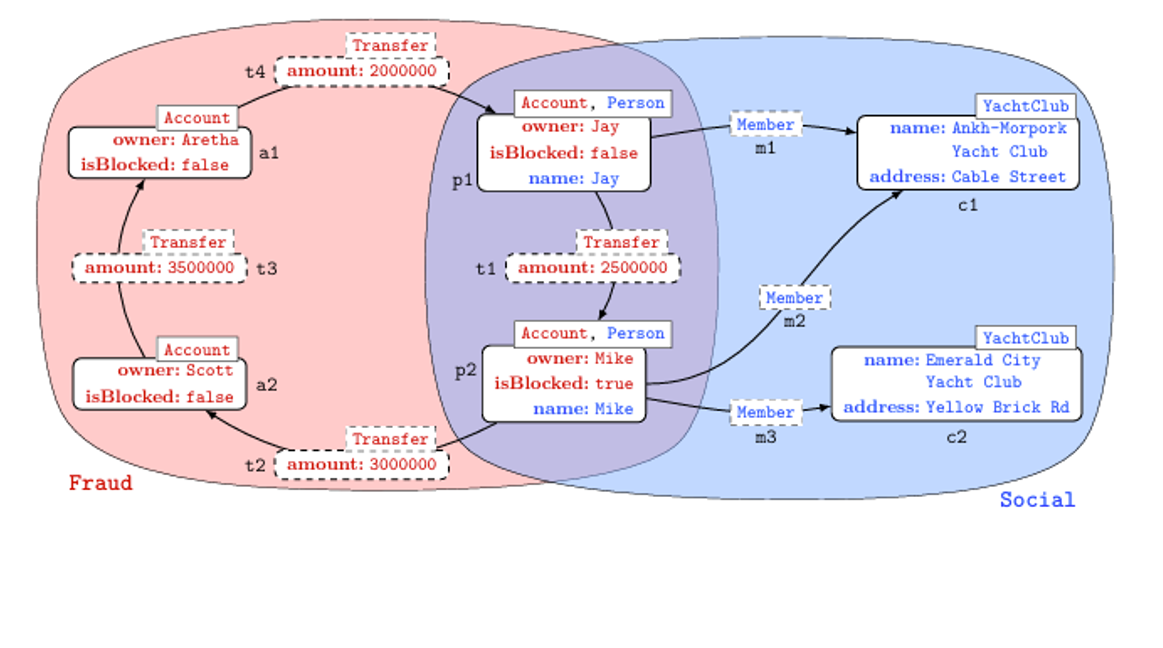


Figure 1. A database with graphs Fraud and Social ( [1))

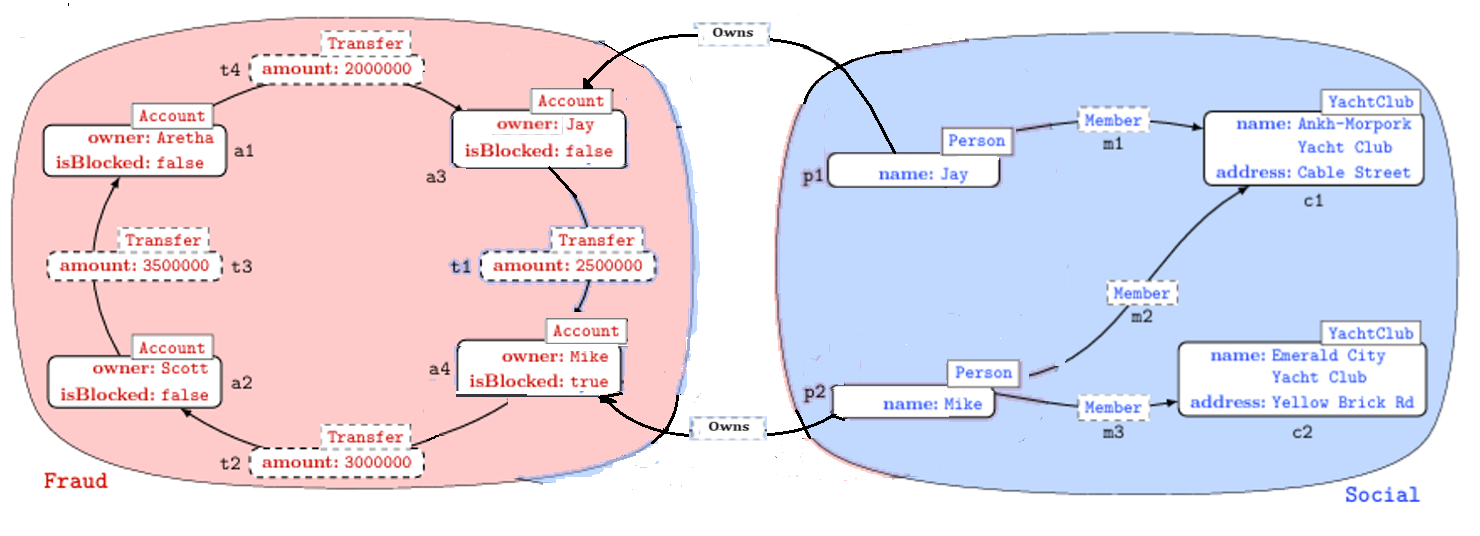


Figure 2. A database with edges linking graphs Fraud and Social (cf[2])

In this model of the example database, we separate the Person and Account nodes, and create edges for Owns, in a similar way to the LDBC Financial Benchmark [2], as shown in Figure 2.

## Initial configuration

The initial state of the database should include graph types that can be informally described as follows:

### Node Types

|  |  |
| --- | --- |
| **Name** | **Record type** |
| Account | (id string, owner string, isBlocked boolean) |
| Person | (id string, name string) |
| YachtClub | (id string, name string, address string) |

### Edge Types

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Record Type** | **Source type** | **Destination type** |
| Member | (id string, personid string, clubid string) | Person | YachtClub |
| Owns | () | Person | Account |
| Transfer | (id string, fromid string, toid string, amount int) | Account | Account |

## Graphs

|  |  |  |
| --- | --- | --- |
| **Name** | **Node Types** | **Edge Types** |
| Fraud | Account | Transfer |
| Social | Person, YachtClub | Member |

## Constructing the nodes and edges using GQL

The initial state of the database is as shown in Figure 1, which is equivalent to populating the database with the following GQL statements (single quotes should be straight quotes here):

insert (:Account{id:'a1',owner:'Aretha',isBlocked:false});

insert (:Account{id:'a2',owner:'Scott', isBlocked:false});

insert (:Account{id:'a3',owner:'Jay',isBlocked:false})<-[:Owns]- (:Person {id:'p1', name:'Jay' });

insert (:Account{id:'a4',owner:'Mike',isBlocked:true})<-[:Owns]- (:Person {id:'p2',name:'Mike' });

insert (:YachtClub{id:'c1',name:'Ankh-Morpork Yacht Club',address:'Cable Street'});

insert (:YachtClub{id:'c2',name:'Emerald City Yacht Club',address:'Yellow Brick Road'});

match (x:Account{id:'a3'}),(y:Account{id:'a4'}) insert (x)-[:Transfer{id:'t1', amount:2500000}]->(y);

match (x:Account{id:'a4'}),(y:Account{id:'a2'}) insert (x)-[:Transfer{id:'t2', amount:3000000}]->(y);

match (x:Account{id:'a2'}),(y:Account{id:'a1'}) insert (x)-[:Transfer{id:'t3', amount:350000}]->(y);

match (x:Account{id:'a1'}),(y:Account{id:'a3'}) insert (x)-[:Transfer{id:'t4', amount:2000000}]->(y);

match (x:Person {id:'p1'}),(y:YachtClub {id:'c1'}) insert (x)-[:"Member" {id:'m1'}]->(y)

match (x:Person {id:'p2'}),(y:YachtClub {id:'c1'}) insert (x)-[:"Member" {id:'m2'}]->(y)

match (x:Person {id:'p2'}),(y:YachtClub {id:'c2'}) insert (x)-[:"Member" {id:'m3'}]->(y)

Alternatively, the same data can be imported using the csv files provided as below.

### Account.csv

accid,owner,isBlocked

a1,Aretha,FALSE

a2,Scott,FALSE

a3,Jay,Jay,FALSE

a4,Mike,Mike,TRUE

### Club.csv

clubid,name,address

c1,Ankh-Morpork Yacht Club,Cable Street

c2,Emerald City Yacht Club,Yellow Brick Rd

### PersonMemberYachtClub.csv

memid,personid,clubid

m1,p1,c1

m2,p2,c1

m3,p2,c2

### PersonOwnsAccount.csv

personid,accid

p1,a3

p2,a4

### Person.csv

id,name

p1,Jay

p2,Mike

### AccountTransferAccount.csv

transid,fromac,toac,amount

t4,a1,a3,2000000

t3,a2,a1,350000

t2,a4,a2,3000000

t1,a3,a4,2500000

### References

[1] Francis, N., Guagliardo, P., Marsault, V., Murlak, F., Rogova, A., Gheerbrant, A., Libkin, L., Martens, W., Peterfreund, L., Vrgoč, D.: A Researcher’s Digest of GQL, 26th International Conference on Database Theory (ICDT 2023), Mar 2023, Ioannina, Greece. doi: 10.4230/LIPIcs.ICDT.2023.1 . https://hal.science/hal-04094449 (retrieved: Dec 2023)

[2] LDBCouncil.org: The LDBC Financial Benchmark (version 0.1.0), https://arxiv.org/pdf/2306.15975.pdf (retrieved February 2024)