

CST4125: Blockchain Development

Week: 9

Title: Queries

Dr Ian Mitchell



smerf.net
Bedfordshire,
UK

2023

Lecture Objectives

Knowledge

- Queries
- Static
- Dynamic
- Resource

Disclaimer

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

Contact and Office Hours

Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Contact and Office Hours

Contact Details

- Name: Dr Ian Mitchell
- Room: TG10
- Address: Middlesex University, Computer Science, London, NW4 4BT
- email: smerf.net

Office Hours

- During term time only
- When: Autumn Term: Mondays 1100-1300hrs
- Please read notifications or emails
- There are occasions that these could be arranged online, e.g., due to industrial action or inclement weather

Deadlines

Description	Submission	Weight	Deadline	Feedback	
				Formative	Summative
1. Hyperledger	MyLearning	50%	18 th December 2022	LW11-12	12/01/2023
2. Ethereum	MyLearning	50%	2 nd April 2023	LW23-24	24/04/2023
Resits	MyLearning	50-100%	1 st July 2023	None	None
Deferrals	MyLearning	50-100%	1 st July 2023	None	None

Problem Definition

- Remove Staff from Trader Network
- Remove Staff from Trader Registry
- Update assets with nominated member of staff
- Update nominated member of staff asset ownership
- Can this be done with Queries?

CTO I

```
1 namespace org.trader.net
2 enum Grade{
3     o Manager
4     o Consultant
5     o Trader
6     o Clerk
7     o Intern}
8 asset Commodity identified by tradingSymbol {
9     o String tradingSymbol
10    o String description
11    o Double quantity
12    -> Trader owner
13 }
14 participant Trader identified by tradeld {
15     o String tradeld
16     o String firstName
17     o String lastName
18     o Grade grade
19     o String [] commoditiesOwned
20 }
21 transaction Trade {
22     -> Commodity commodity
23     -> Trader newOwner
24 }
25 transaction removeStaff {
26     ->Trader removedStaff
27     ->Trader nominatedStaff
28 }
29 transaction initialise {}
30 transaction initialiseAll {}
```

smerf.net

CST4125:L9

Winter 2023

6 / 36

CTO II

smerf.net

CST4125:L9

Winter 2023

7 / 36

Permissions I

```
1 rule traderToSubmitTX{
2     description: "Any trader can submit a transaction"
3     participant: "org.trader.net.Trader"
4     operation: ALL
5     resource: "org.trader.net.*"
6     action: ALLOW
7 }
```

smerf.net

CST4125:L9

Winter 2023

8 / 36

JS - Trade Commodity I

```
1 var ns='org.trader.net';
2 /**
3  * transaction of a commodity from one trader to another
4  * @param {org.trader.net.Trade} tx - tx to be processed
5  * @transaction
6  */
7 async function tradeCommodity(tx) {
8     let buyerAndSeller = new Array();
9     let traderReg = await getParticipantRegistry(ns+'.Trader');
10    let exist = await traderReg.exists(tx.newOwner.getIdentifer());
11    //test if buyer exists
12    if (exist){
13        //buyer
14        let buyer = await traderReg.get(tx.newOwner.getIdentifer());
15        //seller
16        let seller = await traderReg.get(tx.commodity.owner.getIdentifer());
17        // add to the buyer
18        buyer.commoditiesOwned.push(tx.commodity.getIdentifer().toString());
19        buyerAndSeller.push(buyer);
20        //remove from the seller
21        let needle = tx.commodity.getIdentifer().toString();
22        let haystack = seller.commoditiesOwned;
23        let filteredHaystack = haystack.filter((item)>=>item!=needle);
24        seller.commoditiesOwned = filteredHaystack;
25        buyerAndSeller.push(seller);
26        //updateAll traders
27        await traderReg.updateAll(buyerAndSeller);
28        //update the commodity
29        tx.commodity.owner = tx.newOwner;
30        let commodityReg = await getAssetRegistry(ns+'.Commodity');
```

smerf.net

CST4125:L9

Winter 2023

9 / 36

JS - Trade Commodity II

```
31     await commodityReg.update(tx.commodity);
32 } else
33     throw new Error('Buyer does not exists');
34 }
```

smerf.net

CST4125:L9

Winter 2023

10 / 36

JS - Remove I

```
1 }
2 /**
3  * transaction to remove staff
4  * @param {org.trader.net.removeStaff} staff leaving
5  * @transaction
6  */
7 async function removeStaff(tx){
8     let traderReg = await getParticipantRegistry(ns+'.Trader');
9     let selectedTrader = await traderReg.get(tx.nominatedStaff.getIdentifer());
10    let removedTrader = await traderReg.get(tx.removedStaff.getIdentifer());
11    let commodityReg = await getAssetRegistry(ns+'.Commodity');
12    let commodityRegAll = await commodityReg.getAll();
13    let commodityArray = new Array();
14    removedTrader.commoditiesOwned.forEach(item =>selectedTrader.commoditiesOwned.push(
15        item));
16    await traderReg.update(selectedTrader);
17    commodityRegAll.forEach(function (item) {
18        if (item.owner.getIdentifer() == tx.removedStaff.getIdentifer()){
19            let newCommodity = factory.newResource(ns, 'Commodity', item.getIdentifer());
20            let ownerRel = factory.newRelationship(ns, 'Trader', tx.nominatedStaff.
21                getIdentifer());
22            newCommodity.owner = ownerRel;
23            newCommodity.description = item.description;
24            newCommodity.quantity = item.quantity;
25            commodityArray.push(newCommodity);
26        }
27    });
28    await traderReg.remove(removedTrader);
29    await commodityReg.updateAll(commodityArray);
```

smerf.net

CST4125:L9

Winter 2023

11 / 36

JS - Remove II

```
29 }
```

JS - Initialise All I

```
1 /*  
2 * @param {org.trader.net.initialiseAll} no param  
3 * @transaction  
4 */  
5 async function initialiseAll(){  
6 //values to populate staff  
7 let ids = ['1000', '2000', '3000'];  
8 let firstNames = ['Mohamed', 'Pauline', 'Sharon'];  
9 let lastNames = ['Sherrif', 'Tyler', 'Brown'];  
10 let grades = ['Manager', 'Consultant', 'Trader'];  
11 //values to populate commodities  
12 let commodityIDs= ['51', '52', '53', '54', '55'];  
13 let commodityOwner = ['1000', '2000', '2000', '3000', '1000'];  
14 let commodityDesc = ['tea', 'coffee', 'milk', 'chocolate', 'grain'];  
15 let commodityQuan = [100.0, 140, 200.0, 55.5, 400];  
16 // arrays for staff and commodities - push new item every iteration and then addAll to registry  
17 let staff = new Array();  
18 let commodities = new Array();  
19 //have to update staff first since there is a relationship of owner in commodity  
20 for (let i=0; i<ids.length; i++){  
21 let factory = getFactory();  
22 let newStaff = factory.newResource(ns, 'Trader', ids[i]);  
23 newStaff.firstName = firstNames[i];  
24 newStaff.lastName = lastNames[i];  
25 newStaff.grade = grades[i];  
26 newStaff.commoditiesOwned = new Array();  
27 let needle = ids[i];  
28 let haystack = commodityOwner;  
29 //filteredHaystack is commoditiesOwned array
```

JS - Initialise All II

```
30 haystack.filter((element, index) => {if (needle === element) newStaff.commoditiesOwned.  
31 push(commodityIDs[index]) });  
32 staff.push(newStaff);  
33 }  
34 let traderReg = await getParticipantRegistry(ns+'.Trader');  
35 await traderReg.addAll(staff);  
36 commodityIDs.forEach((commodityValue, commodityIndex) => {  
37 let factory = getFactory();  
38 let newCommodity = factory.newResource(ns, 'Commodity', commodityValue);  
39 let ownerRel = factory.newRelationship(ns, 'Trader', commodityOwner[commodityIndex]);  
40 newCommodity.owner = ownerRel;  
41 newCommodity.description = commodityDesc[commodityIndex];  
42 newCommodity.quantity = commodityQuan[commodityIndex];  
43 commodities.push(newCommodity);  
44 });  
45 let commodityReg = await getAssetRegistry(ns+'.Commodity');  
46 await commodityReg.addAll(commodities);
```

at bottom

Static Query Structure

- Open a query file, e.g., query.qry
- Start with keyword query
- Must have a unique name, e.g., selectTrader
- Must have a description
- Must have a statement
- statement is followed by query

```
1 query selectCommodityHigh{  
2 description: "select all the commodities  
3 with quantity higher than 125"  
4 statement:  
5 SELECT org.trader.net.Commodity WHERE  
6 (quantity > 125)
```

Blockchain SQL

Command	Description
SELECT	defines selected registry to query
WHERE	conditions to be applied to selected registry
AND	boolean and operator, defines additional conditions
OR	boolean or operator, defines alternative conditions
CONTAINS	conditions for array values
ORDER BY	sorting of results
SKIP	number of results to skip
LIMIT	defines the max. number of results to return from a query, default value is 25.

Static Query

- Name: selectCommodityHigh
- Registry: org.trader.net.Commodity
- action: extract commodities with quantities higher than 125

```
1 query selectCommodityHigh{  
2 description: "select all the commodities  
3 with quantity higher than 125"  
4 statement:  
5 SELECT org.trader.net.Commodity WHERE  
6 (quantity > 125)
```

Dynamic Queries



- value 125, to be variable?
- parameterized
- undefined parameters using `_$_`
- value is provided with execution
- parameter must be primitive type, a relationship or an enumeration

Dynamic Query



- Name: `selectCommodityHigh`
- Registry: `org.trader.net.Commodity`
- action: extract commodities with quantities higher than 125

```
1 query selectCommodityHigh{
2   description: "select all the commodities
3     with quantity higher than 125"
4   statement:
5     SELECT org.trader.net.Commodity WHERE
      (quantity > $_$value)
```

Transaction & Static Queries



- query
- `let qryRes = query()`
- For now, output results in `console.log`
- `CTRL+SHIFT+i`

Static Query in Transaction



CTO

```
1 transaction pingNetwork{
```

Query

```
1 query selectCommodityHigh{
2   description: "select all the commodities
3     with quantity higher than 125"
4   statement:
5     SELECT org.trader.net.Commodity WHERE
      (quantity > 125)
```

Node.js

```
1 /**
2  * transaction to swap commodities from
3  * 1000 to 2000
4  * @param {org.trader.net.pingNetwork} no
5  * params
6  * @transaction
7  */
8 async function pingNetwork(){
9   let results = await query('
10    selectCommodityHigh');
11   console.log('The results length: '+
12     results.length);
13   results.forEach( item=>console.log(item)
14 );
```

Static Queries and Transactions



Node.js

```
1 /**
2  * transaction to swap commodities from 1000 to 2000
3  * @param {org.trader.net.pingNetwork} no params
4  * @transaction
5  */
6 async function pingNetwork(){
7   let results = await query('selectCommodityHigh');
8   console.log('The Results length: '+results.length);
9   results.forEach( item=>console.log(item) );
10 }
```

Dynamic Query in Transaction



Query

```
1 query individualParam{
2   description: "Exercise 3: select an individual given a tradeID"
3   statement: SELECT org.trader.net.Trader WHERE (tradeId=$_$tID )
4 }
```

CTO

```
1 transaction queryIndividualDynamic{
2   --> Trader trader
3 }
```

Node.js

```
1 /* Exercise 3
2  * @param {org.trader.net.queryIndividualDynamic} tx trader
3  * @transaction
4  */
5 async function queryIndividualDynamic(tx){
6   let tx_trader = tx.trader.tradeId;
7   let queryResult = await query('individualParam', {tID: tx.trader});
8   if(queryResult.length > 0){
9     queryResult.forEach(item=>console.log(item));
10  } else
11  {
12    throw new Error('No results');
```

Dynamic Queries and Transactions

Node.js

```
1 /**
2  * @param {org.trader.net.queryNetwork1} trader
3  * @transaction
4  */
5 async function queryNetwork1(tx){
6   var tx.trader = tx.owner.tradId;
7   let results = await query('selectTrader', {tID: tx.trader});
8   console.log('The results: '+results.length);
9   results.forEach(item=>console.log(item));
10 }
11 /**
```

smerf.net

CST4125:L9

Winter 2023

24 / 36

Queries with relationships

- Commodities contains a relationship with Trader
- Parameter values?
- owner == "1000"
- owner == "resource:namespace.identifier#value"
- getFullyQualifiedIdentifier()

smerf.net

CST4125:L9

Winter 2023

25 / 36

Queries with relationships

Update Asset

In registry: org.trader.net.Commodity

JSON Data Preview

```
1 {
2   "$class": "org.trader.net.Commodity",
3   "tradingSymbol": "51",
4   "description": "tea",
5   "quantity": 100,
6   "owner": "resource:org.trader.net.Trader#1000"
7 }
```

smerf.net

CST4125:L9

Winter 2023

26 / 36

Executing the Query

Query

```
1 query selectCommodityOwnerParam{
2   description: "select all commodities belonging to staff"
3   statement:
4     SELECT org.trader.net.Commodity
5     WHERE (owner==.$owner)
6 }
```

CTO

```
1 transaction queryNetwork2{
2   ->Trader owner
3 }
```

Node.js

```
1 /**
2  * @param {org.trader.net.queryNetwork2} trader
3  * @transaction
4  */
5 async function queryNetwork2(tx){
6   let tx.trader = tx.owner;
7   let resource = 'resource:'+tx.trader.getFullyQualifiedIdentifier();
8   //console.log('Trader: '+tx.trader.getFullyQualifiedIdentifier());
9   console.log('Resource: '+resource);
10  let results = await query('selectCommodityOwnerParam', {owner: resource});
11  console.log('The results: '+results.length);
12  results.forEach(item=>console.log(item));
13 }
```

smerf.net

CST4125:L9

Winter 2023

27 / 36

getFullyQualifiedIdentifier

```
let resource = 'resource:' + tx.trader.getFullyQualifiedIdentifier()
```

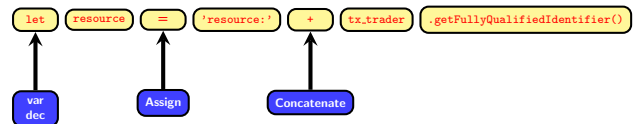
smerf.net

CST4125:L9

Winter 2023

28 / 36

getFullyQualifiedIdentifier



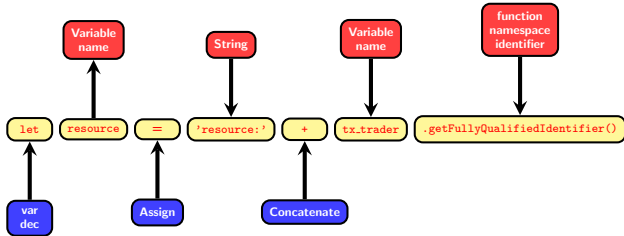
smerf.net

CST4125:L9

Winter 2023

28 / 36

getFullyQualifiedIdentifier



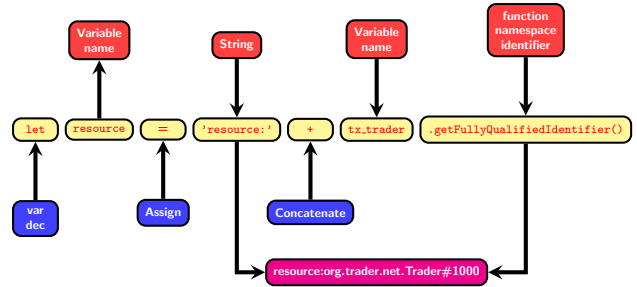
smerf.net

CST4125:L9

Winter 2023

28 / 36

getFullyQualifiedIdentifier



smerf.net

CST4125:L9

Winter 2023

28 / 36

Problem Definition

- Remove a trader
- Nominated trader
- Assign the commodities owned by the removed trader to the nominated trader
- Without using arrays
- Using Queries

smerf.net

CST4125:L9

Winter 2023

29 / 36

Arrays?

Pre - with Array

```
1
2 participant Trader identified by tradeId {
3   o String tradeId
4   o String firstName
5   o String lastName
6   o Grade grade
7   o String[] commoditiesOwned
8 }
```

Post - without Array

```
1
2 participant Trader identified by tradeId {
3   o String tradeId
4   o String firstName
5   o String lastName
6   o Grade grade
7 }
8 transaction removeTrader{
9   -->Trader removedTrader
10  -->Trader nominatedTrader
11 }
```

smerf.net

CST4125:L9

Winter 2023

30 / 36

Query

Query

```
1 query commodityOwner{
2   description: "Exercise 6 & 7: select all commodities owned by a given trader"
3   statement: SELECT org.trader.net.Commodity WHERE (owner == $_owner)
4 }
```

smerf.net

CST4125:L9

Winter 2023

31 / 36

Node.js

```
1 var ns='org.trader.net';
2 /* Exercise 7
3  * @param {org.trader.net.removeTrader} tx
4  * @transaction
5  */
6 async function removeTrader(tx){
7   let tx_rm_trader = tx.removedTrader;
8   let resource = 'resource:'+tx_rm_trader.getFullyQualifiedIdentifier();
9   let queryResults = await query('commodityOwner', {owner: resource});
10
11   let commodityReg = await getAssetRegistry(ns+'.Commodity');
12   let traderReg = await getParticipantRegistry(ns+'.Trader');
13
14   let commodityArray = new Array();
15
16   if(queryResults.length>0){
17     queryResults.forEach(async function(item){
18       let singleCommodity = await commodityReg.get(item.getIdentifer().toString());
19       singleCommodity.owner = tx.nominatedTrader;
20       commodityArray.push(singleCommodity);
21     });
22   }
23   await traderReg.remove(tx_rm_trader.getIdentifer());
24   await commodityReg.updateAll(commodityArray);
25 } else {
26   await traderReg.remove(tx_rm_trader.getIdentifer());
27 }
28 }
```

smerf.net

CST4125:L9

Winter 2023

32 / 36

Access Control



- results from query
- do access control policy apply

Access Control



- results from query
- do access control policy apply
- Yes. ACL will apply to results generated from query
- Any items the current participant does not have access to, will be omitted from the results
- For example.

Access Control



- results from query
- do access control policy apply
- Yes. ACL will apply to results generated from query
- Any items the current participant does not have access to, will be omitted from the results
- For example.
- Current participant does not have read permission to a particular asset, then that asset will not be generated in the results.
- Access control should be defined with queries in mind

Code Summary



- Await
- Var/Let
- Update / UpdateAll
- Add / AddAll
- Get / GetAll
- Asset / Participant Registry
- Remove
- **Queries**
- **Static**
- **Dynamic**
- **JSON resource:namespace:identifier**
- **getFullyQualifiedIdentifier**
- **Queries in Transactions**

References I



- [1] The Linux Foundation. *Hyperledger Architecture, Volume 1*. hyperlink. [Accessed: Jan 2021]. 2017.
- [2] The Linux Foundation. *Hyperledger Architecture, Volume 2*. hyperlink. [Accessed: Jan 2021]. 2018.
- [3] Nitin Gaur et al. *Hands-on Blockchain with Hyperledger: Building Decentralised Applications with Hyperledger Fabric and Composer*. Packt, 2018. ISBN: 9781788994521.

Web Resources



- <http://hyperledger.org>
- <https://nodejs.org>
- <https://hyperledger.github.io/composer/latest/api/runtime-factory>
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array
- <https://github.com/hyperledger/composer-sample-networks>
- <https://hyperledger.github.io/composer/latest/business-network/bnd-create>