Rich Queries

Fabric ledger

- **Transaction Log**: Stores the details of the transaction (the transaction logs/history) in a tamper-proof and sequential manner.
- **World state**: Stores the current state of the business object (asset).
  - **LevelDB** stores chaincode data as simple key-value pairs.
  - **CouchDB** is an optional, alternate state database that allows you to model data on the ledger as **JSON** and issue queries against data values rather than the keys. It allows **indexing** by which we can query large datasets.

Rich Queries

- **GetQueryResult**
- **GetStateByRange**
- **GetHistoryForKey**
- **GetQueryResultWithPagination**

Rich Queries

It allows to fetch data more easily from the ledger

GetQueryResult API

we can perform JSON queries against the data in the state database by using the **GetQueryResult** API and passing a **CouchDB query string**

CouchDB query string

selector (*json*) – JSON object describing criteria used to select documents.
sort (*json array*)– JSON array with list of field name and direction pairs. Used to sort the result based on different fields.

Indexing

- It is used to optimize the performance of database by minimizing number of disk accesses required when a query is processed.
- Indexes allow a database to be queried without having to examine every row with every query.
- Normally, indexes are created for frequently occurring queries allowing the data to be queried more efficiently.
- If sorting is required in a query, CouchDB requires an index that includes the sorted fields.

To define an index, we need to provide three points:

- **fields**: these are the fields to query
- **name**: name of the index
- **type**: always "json" in the context

Sample:File name:**indexColor.json**

{META-INF/statedb/couchdb/indexes}

```
{  "index": {"fields": ["color"]  }, "ddoc":
"indexColorDoc",  "name": "indexColor",  "type": "json"}
```

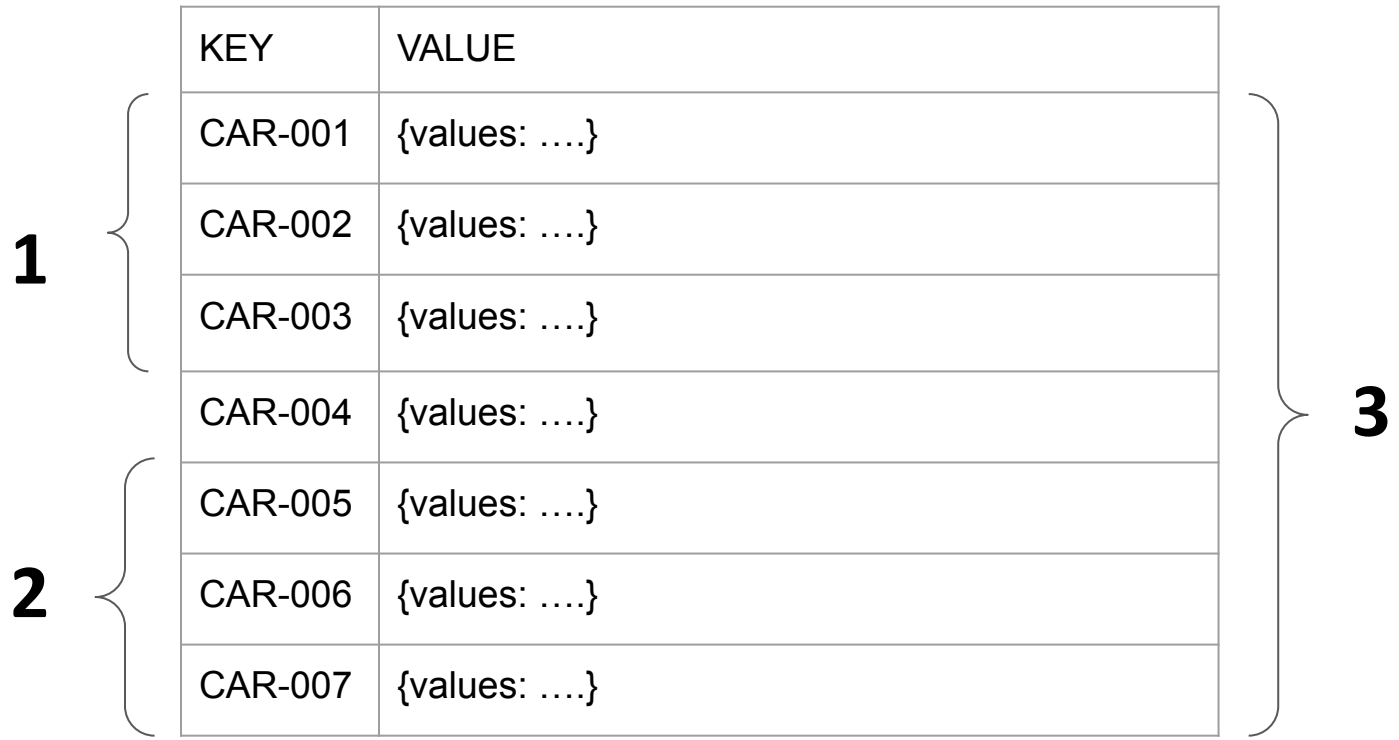
GetStateByRange

- Returns a range of data over a set of keys in the ledger
- The iterator can be used to iterate over all keys between the start key(inclusive) and end key (exclusive)

getStateByRange(start key, end key)

GetStateByRange

- getStateByRange(CAR-001, CAR-004)
- getStateByRange(CAR-005, “ ”)
- getStateByRange(“ ”, “ ”)



Queries on Private Data

getPrivateDataQueryResult ()

getPrivateDataByRange()

THANK YOU