Starchain Sql

Starchain supports a SQL-like language to facilitate users application development, including deploy of smart contract, schema definition of contract, invoking contracts and query the chan. Users can use client interface to connect to the system.

Deploying Contract:

```
DEPLOY(data dir,contract name);
```

This command deploys a smart contract on Starchain, the command has two arguments: the data directory containing smart contract and the name for the smart contract. This command return the result of deploying the contract. In current system, application can be implemented as smart contract in JAVA.

Creating Schema:

```
CREATE TABLE table_name(col1 type1, col2 type2, ...) ON contract name:function;
```

This command creates schema of a contract. Schema of a contract maps the physical storage of transaction to logical presentation. In the command, col1, col2 ... represent logical presentation of arguments of smart contract. By this way, users can attach semantics to block data and query engine can provide rich query on block data. A contract function is mapped to a transaction table.

Invoking contract:

```
INVOKE contract name(func name, argu1, argu2, ...);
```

This command is used to invoke smart contract, the argument func name specify the function name in contract contract name.Arguments argu1; argu2...are arguments of contract function func name.

Query :

```
SELECT select list FROM BLOCK [WHERE where conditions];
```

```
SELECT select list FROM ONCHAIN.table_name [WHERE where conditions];
```

```
TRACE ["yyyy - mm - dd" to "yyyy - mm - dd"] OPERATOR = '?';
TRACE ["yyyy - mm - dd" to "yyyy - mm - dd"] FUNCTION = '?';
```

```
SELECT ONCHAIN.table.column|OFFCHAIN.table.column[,ONCHAIN.column|OFFCHAIN.column] FROM
ONCHAIN.table_name,OFFCHAIN.table_name [WHERE where_conditions] ON
ONCHAIN.table.column|OFFCHAIN.table.column >|=|< ONCHAIN.table.column|OFFCHAIN.table.column;
```

The first query queries for block content. The select list consists of block fields. For example, query ”SELECT * FROM BLOCK WHERE block id = ？;” queries for the block content of block number ID. The where conditions define selection predicates to filter query result.

The second query queries transaction table. For example, query ”SELECT donor FROM donate WHERE transaction id = ID;” queries for the donor of transaction with id of ”ID".

The third query executes track-trace query. For example, the query ” TRACE [”2018-01-01” to”2018-01-07”] OPERATOR = "Jack";” queries for transactions(operations) sent by Jack from 2018-01-01 to 2018-0107.

The last query joins on-line data or integrats on-line and off-line data. For example, the SQL ”SELECT ONCHAIN.donate.donee, OFFCHAIN.donee.address, OFFCHAIN.donee.age, From ONCHAIN. danate, OFFCHAIN.donee ON ONCHAIN.donate.donee = OFFCHAIN.donee.name queries detailed info of donees.