



Ingeniería en Sistemas Computacionales

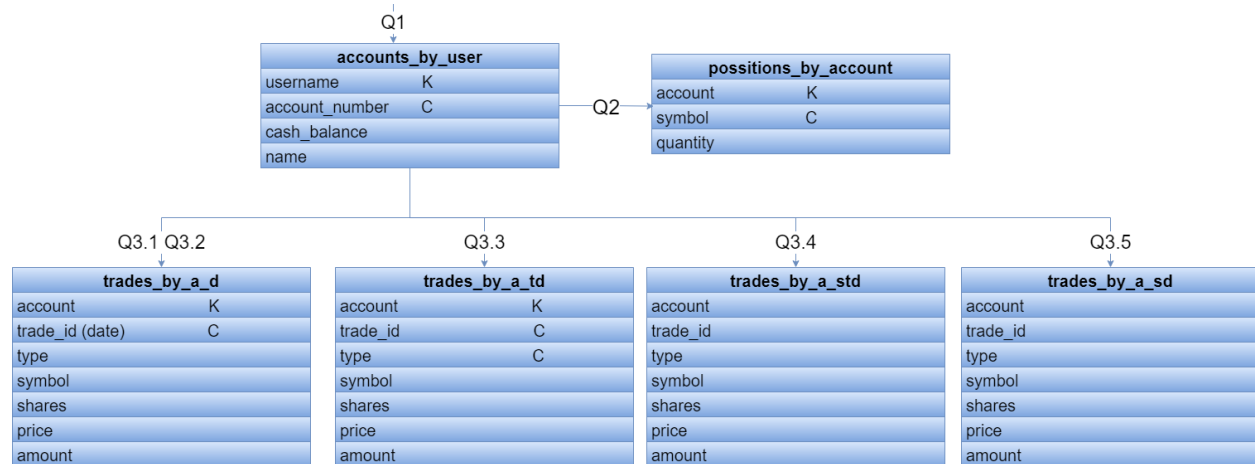
Bases de Datos No Relacionales

Cassandra Lab 2 – CQL

Marco Ricardo Cordero Hernández

Tlaquepaque, Jal., 09 de febrero de 2023

Continuando con el modelo visto anteriormente, ahora se realizarán algunas consultas a través de CQL. Para esto, se proponen las siguientes tablas:



El siguiente fragmento de código CQL define la creación de dichas tablas (con algunas instrucciones adicionales).

```

-- Creación de keyspace inicial y cambio
CREATE KEYSPACE cassandra_lab2
WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 3};

-- Asegurar que no existen las tablas (medida adicional)
DROP TABLE accounts_by_user;
DROP TABLE positions_by_account;
DROP TABLE trades_by_a_d;
DROP TABLE trades_by_a_td;
DROP TABLE trades_by_a_std;
DROP TABLE trades_by_a_sd;

-- Creación de tablas requeridas
CREATE TABLE IF NOT EXISTS accounts_by_user (
    username text,
    account_number varint,
    cash_balance double,
    name text,
    PRIMARY KEY ((username), account_number)
);

CREATE TABLE IF NOT EXISTS positions_by_account (
    account varint,
    symbol text,
    quantity float,
    PRIMARY KEY ((account), symbol)
);

CREATE TABLE IF NOT EXISTS trades_by_a_d (
    account varint,
    trade_id date,
    type boolean,
    symbol text,

```

```

    shares text,
    price double,
    amount float,
    PRIMARY KEY ((account), trade_id)
);

CREATE TABLE IF NOT EXISTS trades_by_a_td (
    account varint,
    trade_id date,
    type boolean,
    symbol text,
    shares text,
    price double,
    amount float,
    PRIMARY KEY ((account), type, trade_id)
);

CREATE TABLE IF NOT EXISTS trades_by_a_std (
    account varint,
    trade_id date,
    type boolean,
    symbol text,
    shares text,
    price double,
    amount float,
    PRIMARY KEY ((account), symbol, type, trade_id)
);

CREATE TABLE IF NOT EXISTS trades_by_a_sd (
    account varint,
    trade_id date,
    type boolean,
    symbol text,
    shares text,
    price double,
    amount float,
    PRIMARY KEY ((account), symbol, trade_id)
);

```

```

cassandra@cqlsh:cassandra_lab2> DESCRIBE TABLES

accounts_by_user      trades_by_a_d  trades_by_a_std
positions_by_account  trades_by_a_sd trades_by_a_td

```

Ojo: el acomodo de las llaves primarias es importante, es decir, el orden en el que se declaran en las tablas afectará a las consultas posteriores.

Posteriormente, se insertan los datos “dummy”.

```

-- Inserción de datos
-- accounts_by_user | text, varint, double, text
INSERT INTO accounts_by_user (username, account_number, cash_balance, name)
VALUES ('MarCordero', 410, 27532.45, 'Marco Cordero');

```

```

INSERT INTO accounts_by_user (username, account_number, cash_balance, name)
VALUES ('AdriD', 777, 60000.9, 'Adriana Díaz');

INSERT INTO accounts_by_user (username, account_number, cash_balance, name)
VALUES ('MarCordero', 430, 5000, 'MarContingence plan');

INSERT INTO accounts_by_user (username, account_number, cash_balance, name)
VALUES ('MFD00M', 31102020, 1000000, 'Daniel Dumile');

-- positions_by_account | varint, text, float
INSERT INTO positions_by_account (account, symbol, quantity)
VALUES (410, 'MRVL1', 2);

INSERT INTO positions_by_account (account, symbol, quantity)
VALUES (410, 'SPOTN', 3.4);

INSERT INTO positions_by_account (account, symbol, quantity)
VALUES (31102020, 'KMD', 100);

INSERT INTO positions_by_account (account, symbol, quantity)
VALUES (777, 'ORCL', 13.12);

-- trades_by_a_* | varint, date, boolean, text, text, double, float
-- Para type -> boolean : 0 = compra, 1 = venta
INSERT INTO trades_by_a_d (account, trade_id, type, symbol, shares, price, amount)
VALUES (410, '2023-02-09', false, 'SPOTN', 'Stock', 500, 3.4);

INSERT INTO trades_by_a_d (account, trade_id, type, symbol, shares, price, amount)
VALUES (410, '2023-02-01', false, 'MRVL1', 'MF', 1000, 2);

INSERT INTO trades_by_a_d (account, trade_id, type, symbol, shares, price, amount)
VALUES (777, '2023-01-09', true, 'ORCL', 'ETF', 2500, 0.88);

INSERT INTO trades_by_a_d (account, trade_id, type, symbol, shares, price, amount)
VALUES (31102020, '2020-01-01', true, 'KMD', 'Stock', 8003, 64);

INSERT INTO trades_by_a_td (account, trade_id, type, symbol, shares, price, amount)
VALUES (410, '2023-02-09', false, 'SPOTN', 'Stock', 500, 3.4);

INSERT INTO trades_by_a_td (account, trade_id, type, symbol, shares, price, amount)
VALUES (410, '2023-02-01', false, 'MRVL1', 'MF', 1000, 2);

INSERT INTO trades_by_a_td (account, trade_id, type, symbol, shares, price, amount)
VALUES (777, '2023-01-09', true, 'ORCL', 'ETF', 2500, 0.88);

INSERT INTO trades_by_a_td (account, trade_id, type, symbol, shares, price, amount)
VALUES (31102020, '2020-01-01', true, 'KMD', 'Stock', 8003, 64);

INSERT INTO trades_by_a_std (account, trade_id, type, symbol, shares, price, amount)
VALUES (410, '2023-02-09', false, 'SPOTN', 'Stock', 500, 3.4);

INSERT INTO trades_by_a_std (account, trade_id, type, symbol, shares, price, amount)
VALUES (410, '2023-02-01', false, 'MRVL1', 'MF', 1000, 2);

INSERT INTO trades_by_a_std (account, trade_id, type, symbol, shares, price, amount)
VALUES (777, '2023-01-09', true, 'ORCL', 'ETF', 2500, 0.88);

INSERT INTO trades_by_a_std (account, trade_id, type, symbol, shares, price, amount)
VALUES (31102020, '2020-01-01', true, 'KMD', 'Stock', 8003, 64);

INSERT INTO trades_by_a_sd (account, trade_id, type, symbol, shares, price, amount)

```

```
VALUES (410, '2023-02-09', false, 'SPOTN', 'Stock', 500, 3.4);

INSERT INTO trades_by_a_sd (account, trade_id, type, symbol, shares, price, amount)
VALUES (410, '2023-02-01', false, 'MRVL1', 'MF', 1000, 2);

INSERT INTO trades_by_a_sd (account, trade_id, type, symbol, shares, price, amount)
VALUES (777, '2023-01-09', true, 'ORCL', 'ETF', 2500, 0.88);

INSERT INTO trades_by_a_sd (account, trade_id, type, symbol, shares, price, amount)
VALUES (31102020, '2020-01-01', true, 'KMD', 'Stock', 8003, 64);
```

Una vez que existen los datos, se realizan las consultas.

```
-- Queries
-- Q1. Consulta de cuentas de un usuario
SELECT name, account_number, cash_balance
FROM accounts_by_user
WHERE username = 'MarCordero';
```

name	account_number	cash_balance
Marco Cordero	410	27532.45
MarContingence plan	430	5000

```
-- Q2. Encontrar todas las posiciones en una cuenta, ordenar por símbolo (ASC)
SELECT symbol, quantity
FROM positions_by_account
WHERE account = 410
ORDER BY symbol ASC;
```

symbol	quantity	trade_id	account	type	symbol
MRVL1	2	2023-02-09	410	False	SPOTN
SPOTN	3.4	2023-02-01	410	False	MRVL1

```
/* Q3. Encontrar todas las transacciones de una cuenta (rango de fechas
opcional),
* tipo de transacción, símbolo; ordenar por fecha de transacción (DESC) */
SELECT trade_id, account, type, symbol
FROM trades_by_a_d
WHERE account = 410 AND trade_id < '2023-03-01' AND trade_id > '2023-01-01'
ORDER BY trade_id DESC;
```

trade_id	account	type	symbol
2023-02-09	410	False	SPOTN
2023-02-01	410	False	MRVL1

```
-- Q3.1 Encontrar todas las transacciones de una cuenta; ordenar por fecha
SELECT trade_id, account, type, symbol
FROM trades_by_a_d
WHERE account = 410
ORDER BY trade_id;
```

trade_id	account	type	symbol
2023-02-01	410	False	MRVL1
2023-02-09	410	False	SPOTN

```
/* Q3.2 Encontrar todas las transacciones de una cuenta y utilizar un rango de
* fecha; ordenar por fecha */
SELECT trade_id, account, type, symbol
FROM trades_by_a_d
WHERE account = 410 AND trade_id < '2023-03-01' AND trade_id > '2023-02-08'
ORDER BY trade_id;
```

trade_id	account	type	symbol
2023-02-09	410	False	SPOTN

```
/* Q3.3 Encontrar todas las transacciones de una cuenta en un rango de fechas,
* y tipo de transacción; ordenar por fecha */
SELECT trade_id, account, type, symbol
FROM trades_by_a_td
WHERE account = 410 AND type = false
AND trade_id < '2023-02-04' AND trade_id > '2023-01-01'
ORDER BY trade_id;
```

trade_id	account	type	symbol
2023-02-01	410	False	MRVL1

```
/* Q3.4 Encontrar todas las transacciones de una cuenta en un rango de fechas,
* tipo de transacción y símbolo de instrumento; ordenar por fecha */
SELECT trade_id, account, type, symbol
FROM trades_by_a_std
WHERE account = 777 AND type = true AND symbol = 'ORCL'
AND trade_id < '2023-01-31' AND trade_id > '2023-01-01'
ORDER BY trade_id;
```

trade_id	account	type	symbol
2023-01-09	777	True	ORCL

```

/* Q3.5 Encontrar todas las transacciones de una cuenta en un rango de fechas
* y símbolo de instrumento; ordenar por fecha */
SELECT trade_id, account, type, symbol
FROM trades_by_a_sd
WHERE account = 31102020 AND symbol = 'KMD'
AND trade_id < '2020-10-31' AND trade_id > '1971-07-13'
ORDER BY trade_id;

```

trade_id	account	type	symbol
2020-01-01	31102020	True	KMD

Una vez que se demostraron las consultas solicitadas, se realizarán un par de ejemplos con las sentencias UPDATE y DELETE.

```

-- UPDATE
UPDATE accounts_by_user SET cash_balance = 385
WHERE username = 'MarCordero' and account_number = 410;
SELECT * FROM accounts_by_user;

```

username	account_number	cash_balance	name
AdriD	777	60000.9	Adriana Díaz
MarCordero	410	385	Marco Cordero
MarCordero	430	5000	MarContingence plan
MFDoom	31102020	1e+06	Daniel Dumile

```

UPDATE positions_by_account SET quantity = 3.14
WHERE account = 410 and symbol = 'SPOTN';
SELECT * FROM positions_by_account;

```

account	symbol	quantity
777	ORCL	13.12
31102020	KMD	100
410	MRVL1	2
410	SPOTN	3.14

```

UPDATE accounts_by_user SET name = 'King Geedorah'
WHERE username = 'MFDoom' and account_number = 31102020;
SELECT * FROM accounts_by_user;

```

username	account_number	cash_balance	name
AdriD	777	60000.9	Adriana Díaz
MarCordero	410	385	Marco Cordero
MarCordero	430	5000	MarContingence plan
MFDoom	31102020	1e+06	King Geedorah

```
-- DELETE
DELETE name FROM accounts_by_user
WHERE username = 'MarCordero' AND account_number = 430;
SELECT * FROM accounts_by_user;
```

username	account_number	cash_balance	name
AdriD	777	60000.9	Adriana Díaz
MarCordero	410	385	Marco Cordero
MarCordero	430	5000	null
MFDoom	31102020	1e+06	King Geedorah

```
DELETE quantity FROM positions_by_account
WHERE account = 410 and symbol = 'SPOTN';
SELECT * FROM positions_by_account;
```

account	symbol	quantity
777	ORCL	13.12
31102020	KMD	100
410	MRVL1	2
410	SPOTN	null

```
DELETE cash_balance FROM accounts_by_user
WHERE username = 'MFDoom' AND account_number = 31102020;
SELECT * FROM accounts_by_user;
```

username	account_number	cash_balance	name
AdriD	777	60000.9	Adriana Díaz
MarCordero	410	385	Marco Cordero
MarCordero	430	5000	null
MFDoom	31102020	null	King Geedorah

Aprendizaje adquirido

Al ver por primera vez un lenguaje de consultas como lo es CQL y con un trasfondo de aprendizaje “anticuado” (más no inútil) encontrado en el curso de bases de datos relacionales, uno podría llegar a pensar que este en conjunto de lo que significa Cassandra puede parecer una cosa sencilla, cuando la realidad dista de ello.

El aprendizaje primordial en esta ocasión resulta ser el que no se debe de dar todo por sentado, puesto que el exceso de confianza hará que las piedras fundamentales del conocimiento actual se derrumben y no habrá nada más que un fallido complejo de superioridad.

Ahora, hablando de un aspecto más técnico, el desarrollo de este ejercicio ha resultado sumamente grato pero solo se ha de aceptar eso después de una sufrida elaboración. Desde el descubrimiento y dolorosa instalación de Docker, hasta la conexión a CQL, resultó ser un camino de múltiples aprendizajes en distintas áreas del conocimiento informático contemporáneo.

La construcción de las consultas no resultó ser tan compleja, no obstante, el desconocimiento en primera instancia (u olvido) de la relevancia del orden en la definición de las llaves primarias en la declaración de las tablas ha sido una de las cosas que más enseñanzas me ha dado el semestre actual, ya que ese simple error desencadenó en una serie de búsquedas y el deseo de una suscripción premium a Stack Overflow.

Finalmente, se puede decir que el curso va por buen rumbo. Quizás para esta tecnología el paso fue un poco apresurado pero no llega a sobrepasar lo justo de una buena clase de nivel licenciatura, claro, si se olvida que antes solo se impartía en maestría...