ITESO

Universidad Jesuita
de Guadalajara

Ingeniería en Sistemas Computacionales

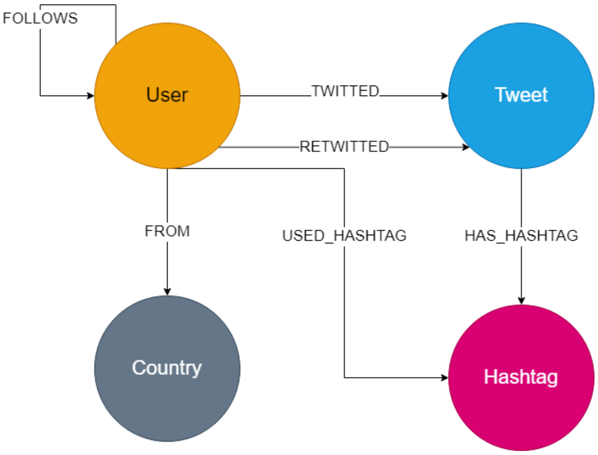
Bases de Datos No Relacionales

Práctica en Clase: Merge y Constraints en Neo4j

Marco Ricardo Cordero Hernández

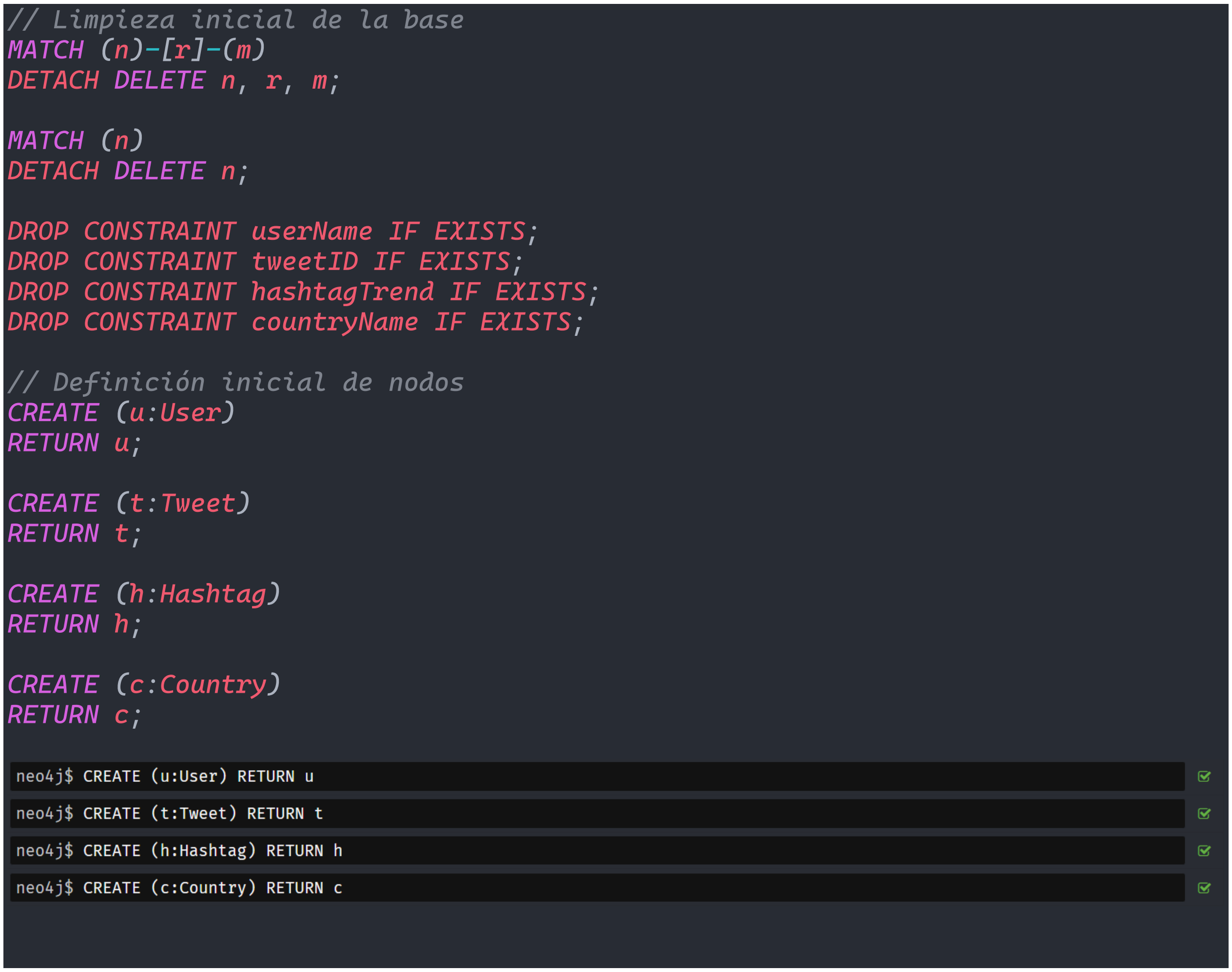
Tlaquepaque, Jal., 11 de abril de 2023

Dado el siguiente diagrama



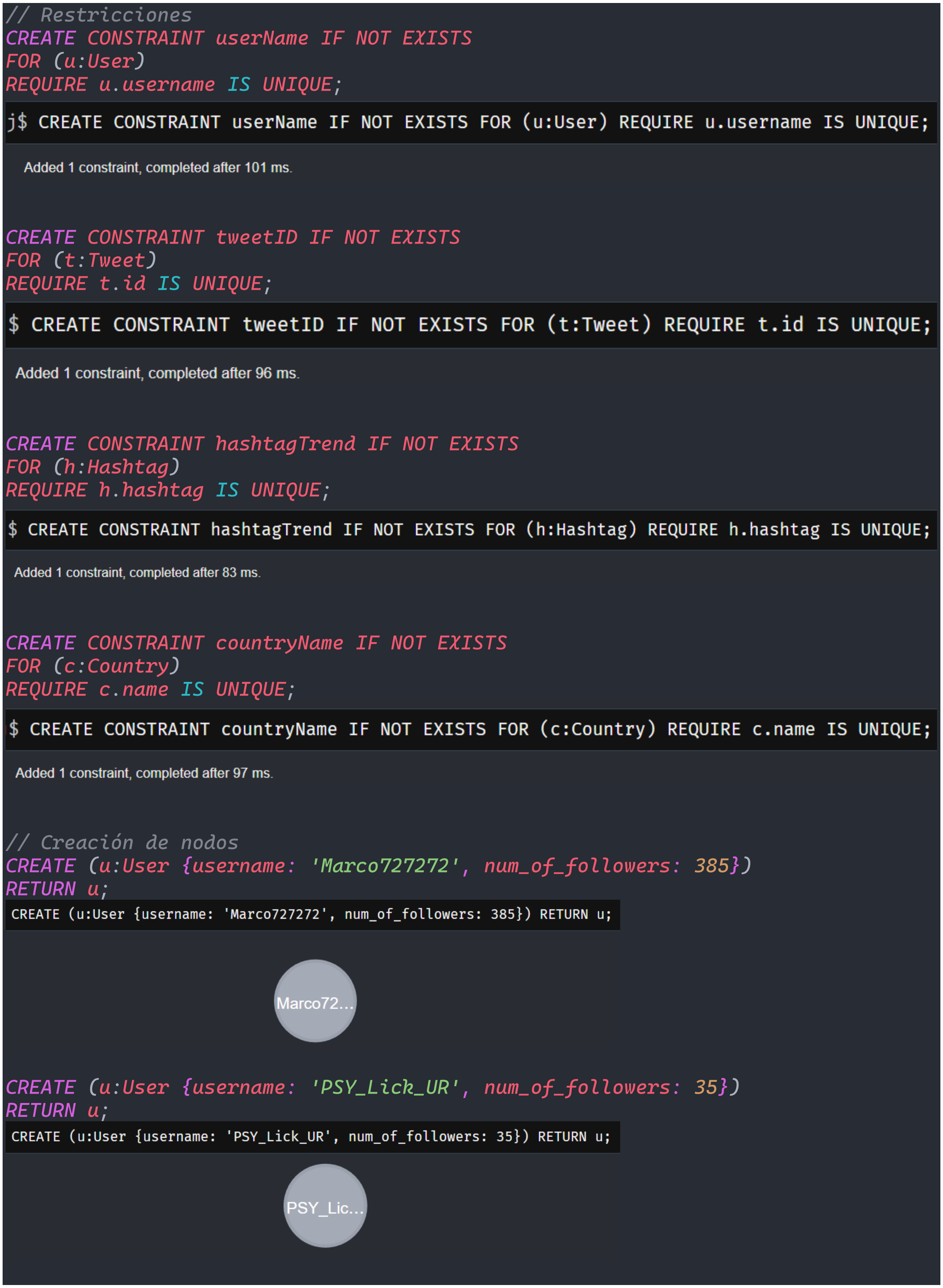
Se solicita realizar las consultas necesarias para replicar el mismo comportamiento dentro de Neo4j a través de su lenguaje de consulta Cypher.

Las sentencias utilizadas fueron las siguientes:



```
// Limpieza inicial de la base
MATCH (n)-[r]-(m)
DETACH DELETE n, r, m;

MATCH (n)
DETACH DELETE n;

DROP CONSTRAINT userName IF EXISTS;
DROP CONSTRAINT tweetID IF EXISTS;
DROP CONSTRAINT hashtagTrend IF EXISTS;
DROP CONSTRAINT countryName IF EXISTS;

// Definición inicial de nodos
CREATE (u:User)
RETURN u;

CREATE (t:Tweet)
RETURN t;

CREATE (h:Hashtag)
RETURN h;

CREATE (c:Country)
RETURN c;
```

```
neo4j$ CREATE (u:User) RETURN u
neo4j$ CREATE (t:Tweet) RETURN t
neo4j$ CREATE (h:Hashtag) RETURN h
neo4j$ CREATE (c:Country) RETURN c
```



```
// Restricciones
CREATE CONSTRAINT userName IF NOT EXISTS
FOR (u:User)
REQUIRE u.username IS UNIQUE;
```

```
j$ CREATE CONSTRAINT userName IF NOT EXISTS FOR (u:User) REQUIRE u.username IS UNIQUE;
```

```
Added 1 constraint, completed after 101 ms.
```

```
CREATE CONSTRAINT tweetID IF NOT EXISTS
FOR (t:Tweet)
REQUIRE t.id IS UNIQUE;
```

```
$ CREATE CONSTRAINT tweetID IF NOT EXISTS FOR (t:Tweet) REQUIRE t.id IS UNIQUE;
```

```
Added 1 constraint, completed after 96 ms.
```

```
CREATE CONSTRAINT hashtagTrend IF NOT EXISTS
FOR (h:Hashtag)
REQUIRE h.hashtag IS UNIQUE;
```

```
$ CREATE CONSTRAINT hashtagTrend IF NOT EXISTS FOR (h:Hashtag) REQUIRE h.hashtag IS UNIQUE;
```

```
Added 1 constraint, completed after 83 ms.
```

```
CREATE CONSTRAINT countryName IF NOT EXISTS
FOR (c:Country)
REQUIRE c.name IS UNIQUE;
```

```
$ CREATE CONSTRAINT countryName IF NOT EXISTS FOR (c:Country) REQUIRE c.name IS UNIQUE;
```

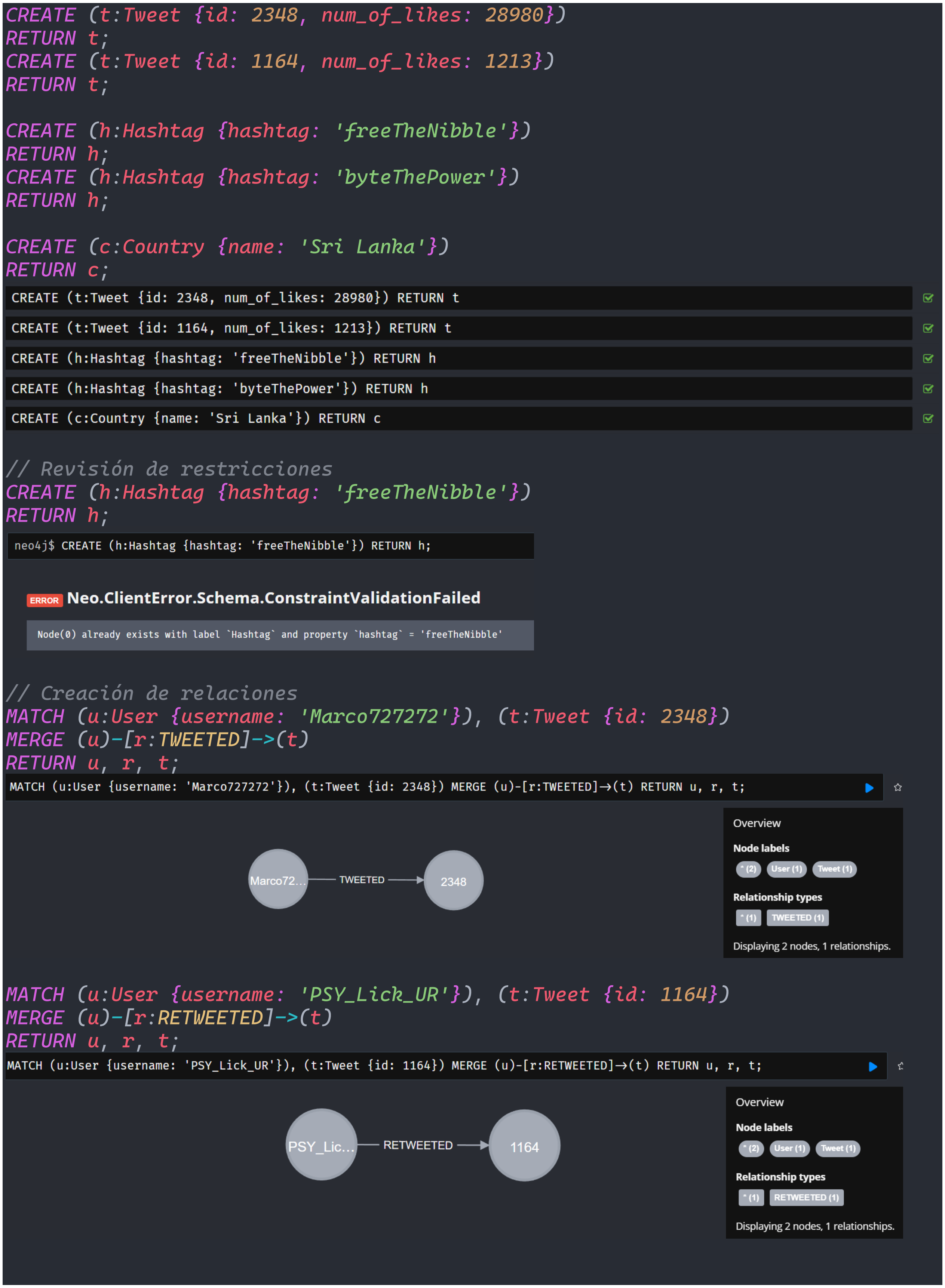
```
Added 1 constraint, completed after 97 ms.
```

```
// Creación de nodos
CREATE (u:User {username: 'Marco727272', num_of_followers: 385})
RETURN u;
```

```
CREATE (u:User {username: 'Marco727272', num_of_followers: 385}) RETURN u;
```

```
CREATE (u:User {username: 'PSY_Lick_UR', num_of_followers: 35})
RETURN u;
```

```
CREATE (u:User {username: 'PSY_Lick_UR', num_of_followers: 35}) RETURN u;
```



```
CREATE (t:Tweet {id: 2348, num_of_likes: 28980})
RETURN t;
CREATE (t:Tweet {id: 1164, num_of_likes: 1213})
RETURN t;

CREATE (h:Hashtag {hashtag: 'freeTheNibble'})
RETURN h;
CREATE (h:Hashtag {hashtag: 'byteThePower'})
RETURN h;

CREATE (c:Country {name: 'Sri Lanka'})
RETURN c;
```

```
CREATE (t:Tweet {id: 2348, num_of_likes: 28980}) RETURN t
CREATE (t:Tweet {id: 1164, num_of_likes: 1213}) RETURN t
CREATE (h:Hashtag {hashtag: 'freeTheNibble'}) RETURN h
CREATE (h:Hashtag {hashtag: 'byteThePower'}) RETURN h
CREATE (c:Country {name: 'Sri Lanka'}) RETURN c
```

```
// Revisión de restricciones
CREATE (h:Hashtag {hashtag: 'freeTheNibble'})
RETURN h;
```

```
neo4j$ CREATE (h:Hashtag {hashtag: 'freeTheNibble'}) RETURN h;
```

ERROR **Neo.ClientError.Schema.ConstraintValidationFailed**

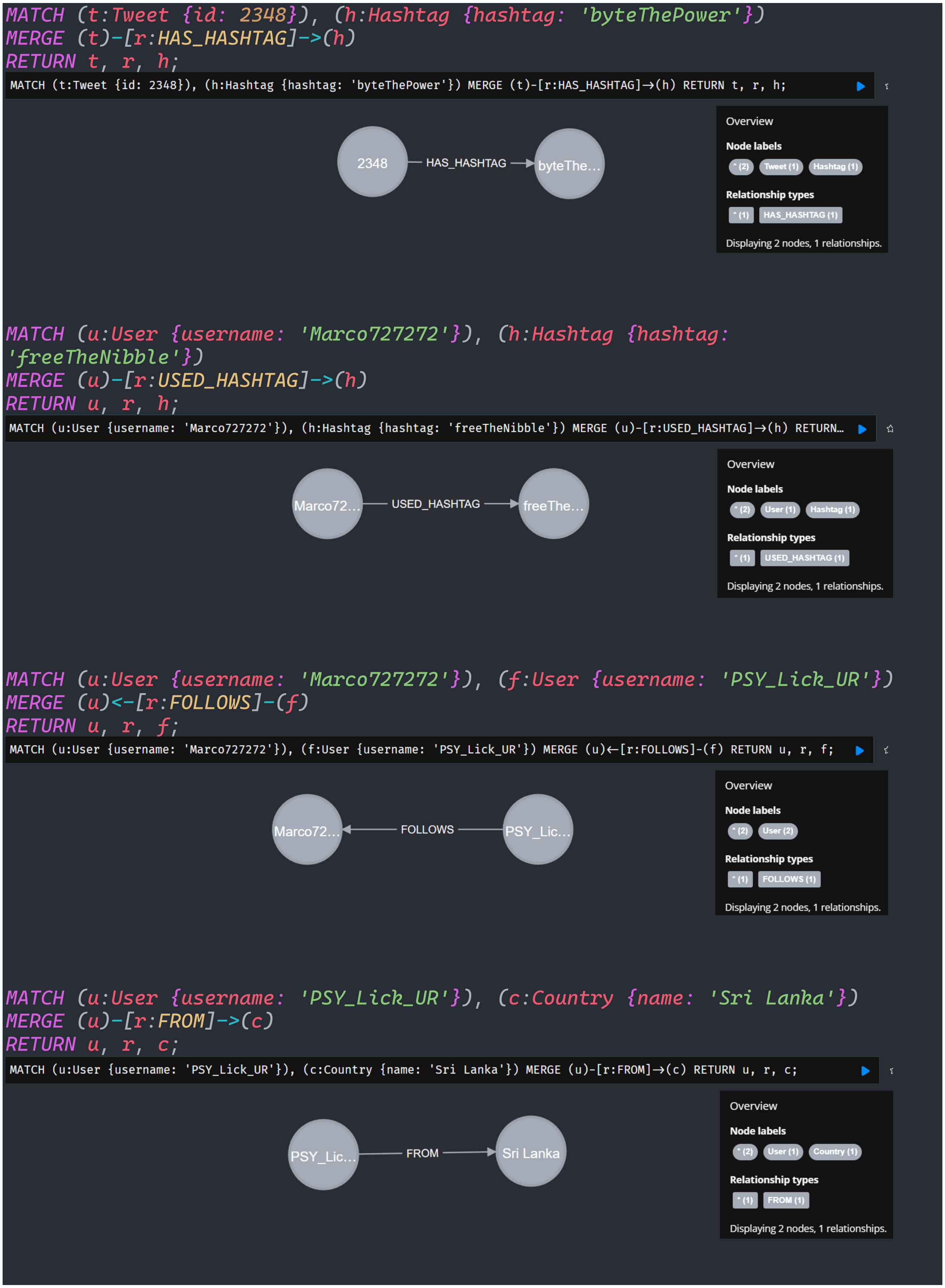
```
Node(0) already exists with label `Hashtag` and property `hashtag` = 'freeTheNibble'
```

```
// Creación de relaciones
MATCH (u:User {username: 'Marco727272'}), (t:Tweet {id: 2348})
MERGE (u)-[r:TWEETED]->(t)
RETURN u, r, t;
```

```
MATCH (u:User {username: 'Marco727272'}), (t:Tweet {id: 2348}) MERGE (u)-[r:TWEETED]→(t) RETURN u, r, t;
```

```
MATCH (u:User {username: 'PSY_Lick_UR'}), (t:Tweet {id: 1164})
MERGE (u)-[r:RETWEETED]->(t)
RETURN u, r, t;
```

```
MATCH (u:User {username: 'PSY_Lick_UR'}), (t:Tweet {id: 1164}) MERGE (u)-[r:RETWEETED]→(t) RETURN u, r, t;
```



```
MATCH (t:Tweet {id: 2348}), (h:Hashtag {hashtag: 'byteThePower'})
MERGE (t)-[r:HAS_HASHTAG]->(h)
RETURN t, r, h;
```

```
MATCH (t:Tweet {id: 2348}), (h:Hashtag {hashtag: 'byteThePower'}) MERGE (t)-[r:HAS_HASHTAG]→(h) RETURN t, r, h;
```

```
MATCH (u:User {username: 'Marco727272'}), (h:Hashtag {hashtag:
'freeTheNibble'})
MERGE (u)-[r:USED_HASHTAG]->(h)
RETURN u, r, h;
```

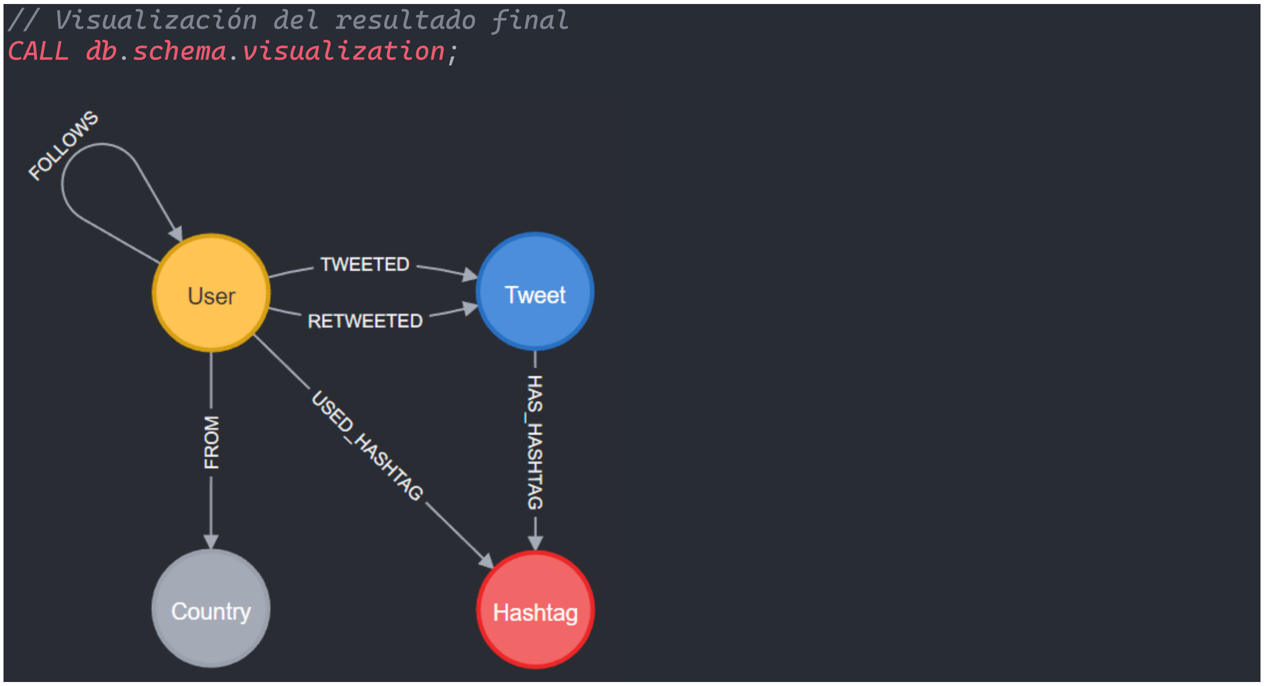
```
MATCH (u:User {username: 'Marco727272'}), (h:Hashtag {hashtag: 'freeTheNibble'}) MERGE (u)-[r:USED_HASHTAG]→(h) RETURN…
```

```
MATCH (u:User {username: 'Marco727272'}), (f:User {username: 'PSY_Lick_UR'})
MERGE (u)<-[r:FOLLOWS]-(f)
RETURN u, r, f;
```

```
MATCH (u:User {username: 'Marco727272'}), (f:User {username: 'PSY_Lick_UR'}) MERGE (u)←[r:FOLLOWS]-(f) RETURN u, r, f;
```

```
MATCH (u:User {username: 'PSY_Lick_UR'}), (c:Country {name: 'Sri Lanka'})
MERGE (u)-[r:FROM]->(c)
RETURN u, r, c;
```

```
MATCH (u:User {username: 'PSY_Lick_UR'}), (c:Country {name: 'Sri Lanka'}) MERGE (u)-[r:FROM]→(c) RETURN u, r, c;
```



```
// Visualización del resultado final
CALL db.schema.visualization;
```