

## UDT en Apache Cassandra

### Objetivo

Aprender a definir y utilizar User-Defined Types (UDT) en Apache Cassandra, integrándolos en un modelo de datos práctico. El taller incluye la creación de tres UDT, su implementación en tablas, y consultas para explorar sus usos.

### Caso de Uso: Gestión de una Biblioteca Digital

El modelo representará:

1. Información de **usuarios**.
2. Información de **libros**.
3. Historial de **préstamos**.

### Parte 1: Definición de UDT

#### 1. Crear UDT para direcciones:

- Este UDT almacenará los datos de la dirección de los usuarios.

```
CREATE TYPE address (  
    street TEXT,  
    city TEXT,  
    state TEXT,  
    zip_code TEXT,  
    country TEXT  
);
```

#### 2. Crear UDT para información del libro:

- Este UDT representará los detalles de un libro, como su título, autor y año de publicación.

```
CREATE TYPE book_details (  
    title TEXT,  
    author TEXT,  
    publication_year INT  
);
```

#### 3. Crear UDT para detalles del préstamo:

- Este UDT almacenará información sobre préstamos, como la fecha de préstamo y la fecha de devolución.

```
CREATE TYPE loan_details (  
    loan_date TIMESTAMP,  
    return_date TIMESTAMP  
);
```

## Parte 2: Crear Tablas Usando UDT

### 1. Tabla de usuarios:

- Almacena los datos personales de los usuarios, incluyendo su dirección como un campo UDT.

```
CREATE TABLE users (  
    user_id UUID PRIMARY KEY,  
    first_name TEXT,  
    last_name TEXT,  
    email TEXT,  
    phone TEXT,  
    address FROZEN<address>  
);
```

### 2. Tabla de libros disponibles:

- Almacena información de los libros, utilizando el UDT book\_details para detallar cada libro.

```
CREATE TABLE books (  
    book_id UUID PRIMARY KEY,  
    details FROZEN<book_details>,  
    available_copies INT  
);
```

### 3. Tabla de préstamos:

- Almacena los préstamos realizados por usuarios, utilizando book\_details y loan\_details.

```
CREATE TABLE loans (  
    loan_id UUID PRIMARY KEY,  
    user_id UUID,  
    book FROZEN<book_details>,  
    loan FROZEN<loan_details>  
);
```

## Parte 3: Insertar Datos de Ejemplo

### 1. Insertar datos en la tabla de usuarios:

```
INSERT INTO users (user_id, first_name, last_name, email, phone,  
address)  
VALUES (  
    uuid(),  
    'Alice',  
    'Johnson',  
    'alice.johnson@example.com',  
    '1234567890',  
    {street: '123 Elm St', city: 'Springfield', state: 'IL',  
zip_code: '62704', country: 'USA'}  
);
```

```
INSERT INTO users (user_id, first_name, last_name, email, phone,
address)
VALUES (
    uuid(),
    'Bob',
    'Smith',
    'bob.smith@example.com',
    '0987654321',
    {street: '456 Oak St', city: 'Columbus', state: 'OH',
zip_code: '43215', country: 'USA'}
);
```

## 2. Insertar datos en la tabla de libros:

```
INSERT INTO books (book_id, details, available_copies)
VALUES (
    uuid(),
    {title: '1984', author: 'George Orwell', publication_year:
1949},
    5
);
```

```
INSERT INTO books (book_id, details, available_copies)
VALUES (
    uuid(),
    {title: 'To Kill a Mockingbird', author: 'Harper Lee',
publication_year: 1960},
    3
);
```

## 3. Insertar datos en la tabla de préstamos:

```
INSERT INTO loans (loan_id, user_id, book, loan)
VALUES (
    uuid(),
    <user_id>, -- Reemplaza con el UUID del usuario
    {title: '1984', author: 'George Orwell', publication_year:
1949},
    {loan_date: toTimestamp(now()), return_date:
toTimestamp(now() + 604800000)} -- 7 días después
);
```

```
INSERT INTO loans (loan_id, user_id, book, loan)
VALUES (
    uuid(),
    <user_id>, -- Reemplaza con el UUID del usuario
    {title: 'To Kill a Mockingbird', author: 'Harper Lee',
publication_year: 1960},
    {loan_date: toTimestamp(now()), return_date:
toTimestamp(now() + 1209600000)} -- 14 días después
);
```

## Parte 4: Consultas de Ejemplo

### 1. Consultar información de un usuario:

```
SELECT * FROM users WHERE user_id = <user_id>;
```

### 2. Consultar todos los libros disponibles:

```
SELECT * FROM books;
```

### 3. Consultar los préstamos realizados por un usuario:

```
SELECT * FROM loans WHERE user_id = <user_id>;
```

### 4. Consultar el título y autor de un libro en un préstamo:

```
SELECT book.title, book.author FROM loans WHERE loan_id =  
<loan_id>;
```

## Parte 5: Actividad Práctica

### 1. Diseñar un UDT adicional:

- Diseña un UDT que almacene el historial de pagos de un usuario, incluyendo el monto, la fecha de pago y el método (por ejemplo, tarjeta, efectivo).

#### Ejemplo:

```
CREATE TYPE payment_details (  
    amount DECIMAL,  
    payment_date TIMESTAMP,  
    method TEXT  
);
```

### 2. Agregar el UDT a una nueva tabla:

- Crea una tabla payments que relacione pagos con usuarios.

#### Ejemplo:

```
CREATE TABLE payments (  
    payment_id UUID PRIMARY KEY,  
    user_id UUID,  
    payment FROZEN<payment_details>  
);
```

### 3. Insertar datos y realizar consultas:

- Inserta pagos realizados por usuarios y consulta los detalles.