

# Operaciones CRUD en CQL

Antes de empezar a ver las operaciones CRUD en CQL hemos de crear un *keyspace* y una tabla de ejemplo.

Para crear el *keyspace* ejecutaremos el siguiente comando:

```
CREATE KEYSPACE sbd
WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 2 };
```

a continuación creamos la tabla `miembros`:

```
CREATE TABLE IF NOT EXISTS sbd.miembros (
    id int PRIMARY KEY,
    nombre text,
    apellidos text,
    email text,
    rol text static,
    fecha_alta timestamp,
    fecha_de_nacimiento date
)
WITH comment = 'Tabla con datos de prueba.';
```

Si la *partition key* es compuesta la definiremos de la siguiente forma:

```
CREATE TABLE IF NOT EXISTS sbd.miembros (
    id int,
    nombre text,
    apellidos text,
    email text,
    rol text static,
    fecha_alta timestamp,
    fecha_de_nacimiento date,
    PRIMARY KEY (id, fecha_alta)
) WITH comment = 'Tabla con datos de prueba.',
AND CLUSTERING ORDER BY (fecha_alta DESC);
```

Como se puede comprobar este código es cada vez más incómodo de escribir en la consola de CQLSH. Por ello, a partir de ahora utilizaremos un fichero de texto para almacenar las sentencias CQL que queremos ejecutar.

Un *script* de CQL es un fichero de texto con extensión `.cql` que contiene sentencias CQL. Para ejecutar un *script* de CQL desde dentro de CQLSH utilizaremos la sentencia `SOURCE`:

```
SOURCE 'path/to/script.cql';
```

Si lo que necesitamos es ejecutar un *script* de CQL desde la línea de comandos utilizaremos el siguiente comando:

```
cqlsh -f path/to/script.cql
```

Finalmente veamos cómo crear un *keyspace* y una tabla de ejemplo utilizando un *script* de CQL.

```
CREATE KEYSPACE sbd
  WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 2 };

CREATE TABLE IF NOT EXISTS sbd.miembros (
  id int PRIMARY KEY,
  nombre text,
  apellidos text,
  email text,
  rol text static,
  fecha_alta timestamp,
  fecha_de_nacimiento date
  PRIMARY KEY (id, fecha_alta)
);

// Creamos una segunda tabla de asignaturas

CREATE TABLE IF NOT EXISTS sbd.asignaturas (
  id int PRIMARY KEY,
  nombre text,
  curso int,
  profesor map <text, text>,
  alumnos list<map<text, text>>
  fecha_inicio timestamp,
  fecha_fin timestamp
);

// Comprobamos que se han creado el keyspace y las tablas

DESCRIBE KEYSPACES;

USE sbd;

DESCRIBE TABLES;
```

Guardamos el código anterior en un fichero llamado `script01.cql` y lo ejecutamos:

```
docker exec -it cass1 cqlsh -f /scripts/script01.cql
```