Introducción a bases de datos
Sesión 4

Adrian González Pardo

20 de mayo de 2021

1. MySQL

1.1. Ejemplos

```
# Ejemplo de creacion de BD en caso de existir no lo hace
CREATE DATABASE IF NOT EXISTS Nombre;

# Ejemplo de eliminacion de BD en caso de existir
DROP DATABASE IF EXISTS Nombre;

CREATE DATABASE IF NOT EXISTS MiNombre;

USE MiNombre;
```

```
USE MiNombre;
# Creacion de tabla 'users' en caso de no existir
CREATE TABLE IF NOT EXISTS users (
   user_id INT PRIMARY KEY,
   genero VARCHAR(1),
   edad INT,
   ocup INT,
   cp VARCHAR(20)
);

# En caso de existir o no
# DROP TABLE IF EXISTS nombre;
```

```

# Para cargar los datos desde el terminal de nuestro servidor


SHOW VARIABLES LIKE "local_infile";
# En caso de off modificamos el acceso
 SET GLOBAL local_infile = 'ON';
SHOW VARIABLES LIKE "local_infile";

# En caso de que el INFILE mande un error como
# The MySQL server is running with the
#   --secure-file-priv option so it cannot execute this statement
# Ejecutar lo siguiente:
LOAD DATA LOCAL INFILE "ml-1m/users.dat"
INTO TABLE users
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n';
```

1.2. Retos

```
USE MiNombre;
# Creacion de tabla "movies" si no existe
CREATE TABLE IF NOT EXISTS movies (
  movie_id INT PRIMARY KEY,
  titulo VARCHAR(200),
  genero VARCHAR(200)
);
# Creacion de tabla "ratings" si no existe
CREATE TABLE IF NOT EXISTS ratings (
  user_id INT,
  movie_id INT,
  rating INT,
  time_stamp BIGINT,
  FOREIGN KEY(user_id) REFERENCES users(user_id),
  FOREIGN KEY(movie_id) REFERENCES movies(movie_id)
);

SELECT * FROM users LIMIT 10;

SELECT * FROM movies LIMIT 10;

SELECT * FROM ratings LIMIT 10;
```

```
USE MiNombre;
# Carga de datos en "movies" del archivo "movies.dat"
LOAD DATA LOCAL INFILE "ml-1m/movies.dat"
INTO TABLE movies
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n';

# Carga de datos en "ratings" del archivo "ratings.dat"
LOAD DATA LOCAL INFILE "ml-1m/ratings.dat"
INTO TABLE ratings
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n';
```

1.3. Capturas de ejecución en servidor

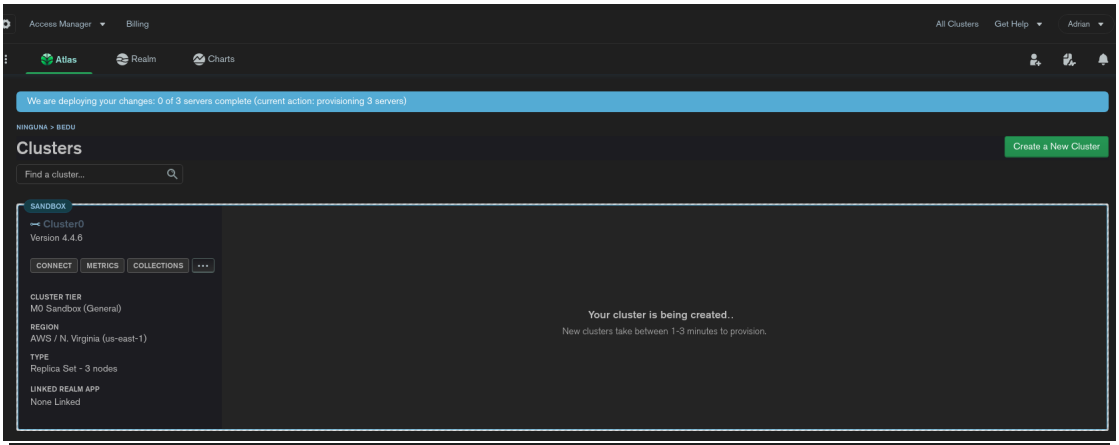
```
mysql> CREATE TABLE IF NOT EXISTS users (
    ->   user_id INT PRIMARY KEY,
    ->   genero VARCHAR(1),
    ->   edad INT,
    ->   ocup INT,
    ->   cp VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE IF NOT EXISTS movies (
    ->   movie_id INT PRIMARY KEY,
    ->   titulo VARCHAR(200),
    ->   genero VARCHAR(200)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> CREATE TABLE IF NOT EXISTS raitings (
    ->   user_id INT,
    ->   movie_id INT,
    ->   rating INT,
    ->   time_stamp BIGINT,
    ->   FOREIGN KEY(user_id) REFERENCES users(user_id),
    ->   FOREIGN KEY(movie_id) REFERENCES movies(movie_id)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> LOAD DATA INFILE "./ml-1m/users.dat"
    -> INTO TABLE users
    -> FIELDS TERMINATED BY ';'
    -> LINES TERMINATED BY '\n';
ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement
```

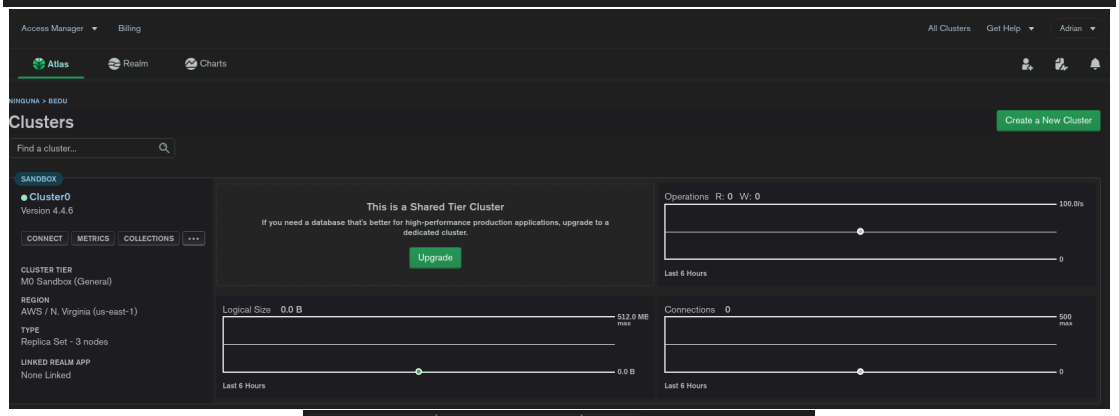
```
Database changed
mysql> LOAD DATA LOCAL INFILE "ml-1m/users.dat"
    -> INTO TABLE users
    -> FIELDS TERMINATED BY ';'
    -> LINES TERMINATED BY '\n';
Query OK, 6040 rows affected (0.51 sec)
Records: 6040  Deleted: 0  Skipped: 0  Warnings: 0

mysql> show tables;
+--------------------+
| Tables_in_MiNombre |
+--------------------+
| movies             |
| raitings           |
| users              |
+--------------------+
3 rows in set (0.00 sec)

mysql> LOAD DATA LOCAL INFILE "ml-1m/movies.dat"
    -> INTO TABLE movies
    -> FIELDS TERMINATED BY ';'
    -> LINES TERMINATED BY '\n';
Query OK, 3883 rows affected, 1 warning (0.40 sec)
Records: 3883  Deleted: 0  Skipped: 0  Warnings: 1
```

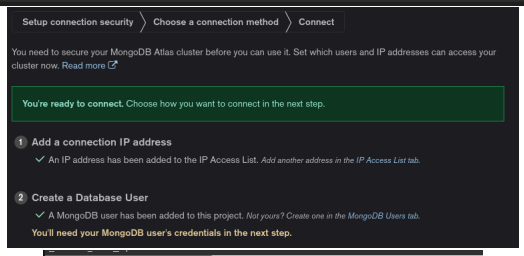
```
mysql> LOAD DATA LOCAL INFILE "ml-1m/ratings.dat"
    -> INTO TABLE ratings
    -> FIELDS TERMINATED BY ';'
    -> LINES TERMINATED BY '\n';
Query OK, 1000209 rows affected (41.32 sec)
Records: 1000209  Deleted: 0  Skipped: 0  Warnings: 0
```

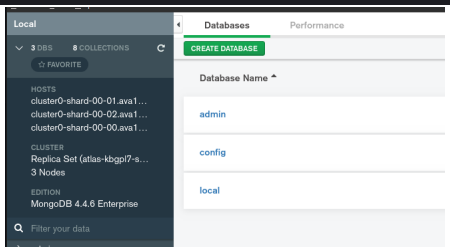
```
mysql> SELECT * FROM users LIMIT 10;
+---------+--------+------+------+-------+
| user_id | genero | edad | ocup | cp    |
+---------+--------+------+------+-------+
|       1 | F      |    1 |   10 | 48067 |
|       2 | M      |   56 |   16 | 70072 |
|       3 | M      |   25 |   15 | 55117 |
|       4 | M      |   45 |    7 | 02460 |
|       5 | M      |   25 |   20 | 55455 |
|       6 | F      |   50 |    9 | 55117 |
|       7 | M      |   35 |    1 | 06810 |
|       8 | M      |   25 |   12 | 11413 |
|       9 | M      |   25 |   17 | 61614 |
|      10 | F      |   35 |    1 | 95370 |
+---------+--------+------+------+-------+
10 rows in set (0.00 sec)

mysql>
mysql> SELECT * FROM movies LIMIT 10;
+----------+-----------------------------------+-----------------------------+
| movie_id | titulo                            | genero                      |
+----------+-----------------------------------+-----------------------------+
|        1 | Toy Story (1995)                  | Animation|Children's|Comedy   |
|        2 | Jumanji (1995)                    | Adventure|Children's|Fantasy |
|        3 | Grumpier Old Men (1995)           | Comedy|Romance              |
|        4 | Waiting to Exhale (1995)          | Comedy|Drama                |
|        5 | Father of the Bride Part II (1995) | Comedy                      |
|        6 | Heat (1995)                       | Action|Crime|Thriller       |
|        7 | Sabrina (1995)                    | Comedy|Romance              |
|        8 | Tom and Huck (1995)               | Adventure|Children's        |
|        9 | Sudden Death (1995)               | Action                      |
|       10 | GoldenEye (1995)                  | Action|Adventure|Thriller    |
+----------+-----------------------------------+-----------------------------+
10 rows in set (0.00 sec)

mysql>
mysql> SELECT * FROM ratings LIMIT 10;
+---------+----------+--------+------------+
| user_id | movie_id | rating | time_stamp |
+---------+----------+--------+------------+
|       1 |     1193 |      5 |  978300760 |
|       1 |      661 |      3 |  978302109 |
|       1 |      914 |      3 |  978301968 |
|       1 |     3408 |      4 |  978300275 |
|       1 |     2355 |      5 |  978824291 |
|       1 |     1197 |      3 |  978302268 |
|       1 |     1287 |      5 |  978302039 |
|       1 |     2804 |      5 |  978300719 |
|       1 |      594 |      4 |  978302268 |
|       1 |      919 |      4 |  978301368 |
+---------+----------+--------+------------+
10 rows in set (0.01 sec)
```

2. MongoDB

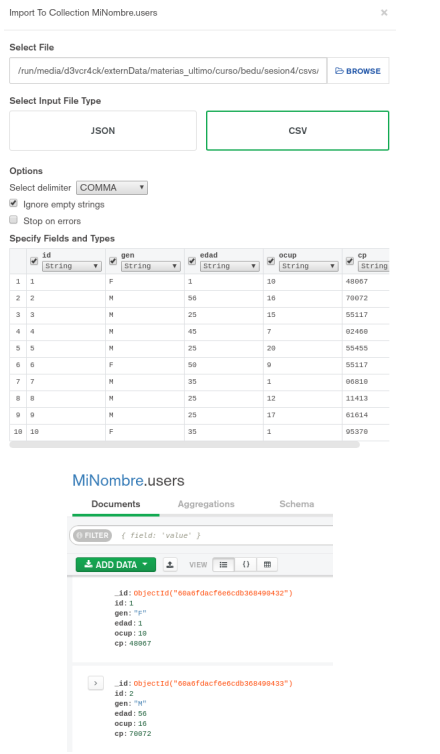




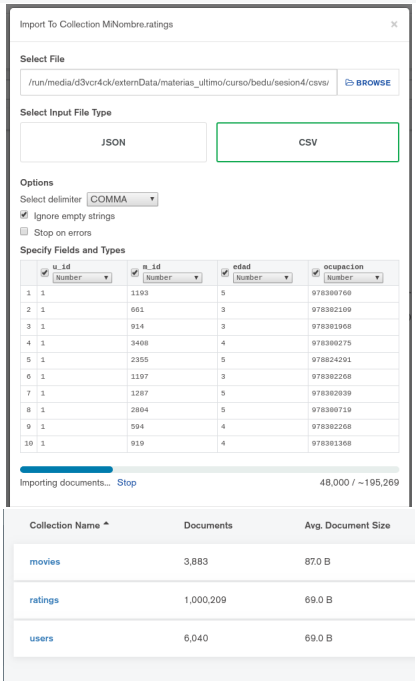




Usuario y contraseña: introabd:introabd1234
Más la creación del servidor de MongoDB

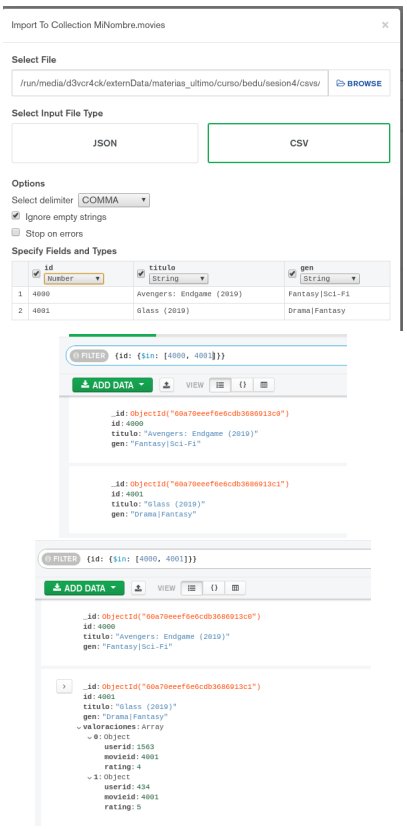


Creación de colección users



Creación de colección movies y ratings con sus archivos csv respectivo

Uso de la herramienta de MongoDB Compass para añadir, modificar, buscar información dentro de nuestra coleccion users



Ejercicio de MongoDB en la colección movies