

Retos Sesión 4

Hurtado Gutiérrez Marco Antonio
markhg@ciencias.unam.mx

3 de Diciembre de 2020

Reto 1: Realizando operaciones con tablas.

1. ¿Definir los campos y tipos de datos para la tabla movies haciendo uso de los archivos movies.dat y README.?

Respuesta:

- a) Primero revisamos el contenido del archivo

```
Help
movies.dat
1 1::Toy Story (1995)::Animation|Children's|Comedy
2 2::Jumanji (1995)::Adventure|Children's|Fantasy
3 3::Grumpier Old Men (1995)::Comedy|Romance
4 4::Waiting to Exhale (1995)::Comedy|Drama
5 5::Father of the Bride Part II (1995)::Comedy
6 6::Heat (1995)::Action|Crime|Thriller
7 7::Sabrina (1995)::Comedy|Romance
8 8::Tom and Huck (1995)::Adventure|Children's
9 9::Sudden Death (1995)::Action
10 10::GoldenEye (1995)::Action|Adventure|Thriller
11 11::American President, The (1995)::Comedy|Drama|Romance
12 12::Dracula: Dead and Loving It (1995)::Comedy|Horror
13 13::Balto (1995)::Animation|Children's
14 14::Nixon (1995)::Drama
15 15::Cutthroat Island (1995)::Action|Adventure|Romance
16 16::Casino (1995)::Drama|Thriller
17 17::Sense and Sensibility (1995)::Drama|Romance
18 18::Four Rooms (1995)::Thriller
19 19::Ace Ventura: When Nature Calls (1995)::Comedy
20 20::Money Train (1995)::Action
21 21::Get Shorty (1995)::Action|Comedy|Drama
22 22::Copycat (1995)::Crime|Drama|Thriller
23 23::Assassins (1995)::Thriller
24 24::Powder (1995)::Drama|Sci-Fi
25 25::Leaving Las Vegas (1995)::Drama|Romance
26 26::Othello (1995)::Drama
27 27::Now and Then (1995)::Drama
28 28::Persuasion (1995)::Romance
29 29::City of Lost Children, The (1995)::Adventure|Sci-Fi
30 30::Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)::D
```

Figura 1: Vemos que tiene nombre de la película, año y genero .

b) Luego se revisa la documentación, dentro del archivo README.

```
136
137 MOVIES FILE DESCRIPTION
138 =====
139
140 Movie information is in the file "movies.dat" and is in the following
141 format:
142
143 MovieID::Title::Genres
144
145 - Titles are identical to titles provided by the IMDB (including
146 year of release)
147 - Genres are pipe-separated and are selected from the following genres:
148
149 * Action
150 * Adventure
151 * Animation
152 * Children's
153 * Comedy
154 * Crime
155 * Documentary
156 * Drama
157 * Fantasy
158 * Film-Noir
159 * Horror
160 * Musical
161 * Mystery
162 * Romance
163 * Sci-Fi
164 * Thriller
165 * War
```

Figura 2: Vemos los campos que va a tener la tabla movies que son: MovieID, Title, Genres.

c) Así que se definen los siguientes datos y tipo para crear la tabla Movies en MySQL:

- id_movie INT PRIMARY KEY
- title VARCHAR(100)
- genres VARCHAR(100)

2. Crear la tabla movies (recuerda usar el mismo nombre del archivo sin la extensión para vincular nombres de tablas con archivos).

Respuesta:

```
-- Crear la tabla movies (recuerda usar el mismo nombre del archivo sin
-- la extensión para vincular nombres de tablas con archivos).

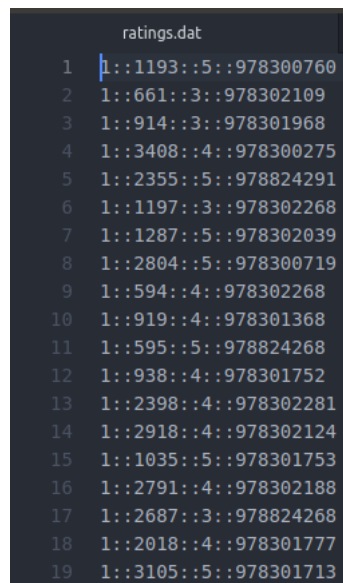
CREATE TABLE movies (
  id_movie INT PRIMARY KEY,
  title VARCHAR(100),
  genres VARCHAR(100)
);
```

Figura 3: Creamos la tabla movies.

3. Definir los campos y tipos de datos para la tabla ratings haciendo uso de los archivos ratings.dat y README.

Respuesta:

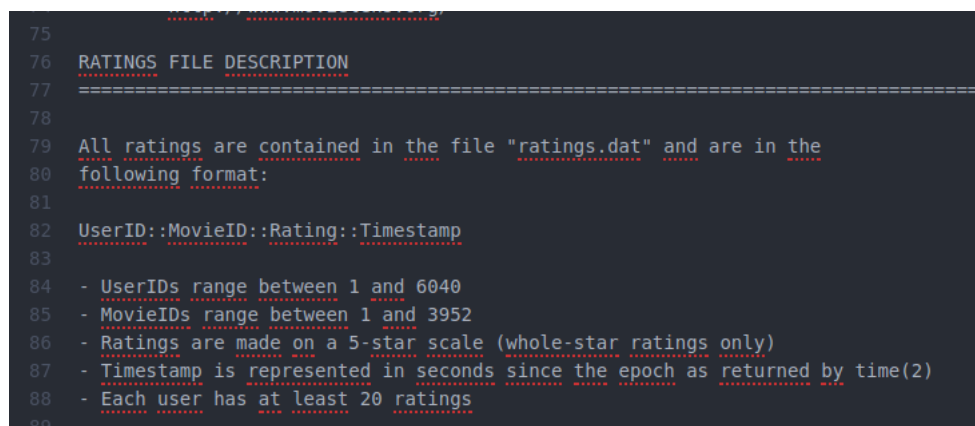
- a) Primero revisamos el contenido del archivo



```
ratings.dat
1 1::1193::5::978300760
2 1::661::3::978302109
3 1::914::3::978301968
4 1::3408::4::978300275
5 1::2355::5::978824291
6 1::1197::3::978302268
7 1::1287::5::978302039
8 1::2804::5::978300719
9 1::594::4::978302268
10 1::919::4::978301368
11 1::595::5::978824268
12 1::938::4::978301752
13 1::2398::4::978302281
14 1::2918::4::978302124
15 1::1035::5::978301753
16 1::2791::4::978302188
17 1::2687::3::978824268
18 1::2018::4::978301777
19 1::3105::5::978301713
```

Figura 4: Vemos muchos números como campos y no queda nada claro que pueda ser cada uno .

- b) Luego se revisa la documentación, dentro del archivo README.



```
75
76 RATINGS FILE DESCRIPTION
77 =====
78
79 All ratings are contained in the file "ratings.dat" and are in the
80 following format:
81
82 UserID::MovieID::Rating::Timestamp
83
84 - UserIDs range between 1 and 6040
85 - MovieIDs range between 1 and 3952
86 - Ratings are made on a 5-star scale (whole-star ratings only)
87 - Timestamp is represented in seconds since the epoch as returned by time(2)
88 - Each user has at least 20 ratings
89
```

Figura 5: Revisando el README ya podemos ver los campos que va a tener la tabla Ratings que son: UserID,MovieID,Rating,Timestamp.

- c) Así que se definen los siguientes datos, tipo y llaves foráneas para crear la tabla Ratings en MySQL:
- id_users INT
 - id_movie INT
 - ratings INT
 - time_stamp BIGINT
 - FOREIGN KEY (id_users) REFERENCES users(id),
 - FOREIGN KEY (id_movie) REFERENCES movies(id_movie)
4. Crear la tabla ratings (recuerda usar el mismo nombre del archivo sin la extensión para vincular nombres de tablas con archivos).

Respuesta:

```
28 -- Crear la tabla ratings (recuerda usar el mismo nombre
29 -- del archivo sin la extensión para vincular nombres de tablas con archivos)
30
31 CREATE TABLE rating (
32     id_users INT,
33     id_movie INT,
34     ratings INT,
35     time_stamp BIGINT,
36     FOREIGN KEY (id_users) REFERENCES users(id),
37     FOREIGN KEY (id_movie) REFERENCES movies(id_movie)
38 );
39
```

Figura 6: Creamos la tabla rating.

Reto 2: Importando datos a una tabla en formato CSV.

1. Usando como base el archivo movies.dat, limpiarlo e importar los datos en la tabla movies creada en el Reto 1.

Respuesta:

- Abrimos con el editor de texto Atom el archivo movies.dat y agregamos la siguiente línea al inicio:

```
id_movie,title,genres
```

Figura 7: id_movie,title,genres.

- Ahora reemplaza todos los símbolos, :: por ,. El archivo final debe verse así:

```

s Help
movies.csv
1 id_movie,title,genres
2 1,Toy Story (1995),Animation|Children's|Comedy
3 2,Jumanji (1995),Adventure|Children's|Fantasy
4 3,Grumpier Old Men (1995),Comedy|Romance
5 4,Waiting to Exhale (1995),Comedy|Drama
6 5,Father of the Bride Part II (1995),Comedy
7 6,Heat (1995),Action|Crime|Thriller
8 7,Sabrina (1995),Comedy|Romance
9 8,Tom and Huck (1995),Adventure|Children's
10 9,Sudden Death (1995),Action
11 10,GoldenEye (1995),Action|Adventure|Thriller
12 11,American President- The (1995),Comedy|Drama|Romance
13 12,Dracula: Dead and Loving It (1995),Comedy|Horror
14 13,Balto (1995),Animation|Children's
15 14,Nixon (1995),Drama
16 15,Cutthroat Island (1995),Action|Adventure|Romance
17 16,Casino (1995),Drama|Thriller
18 17,Sense and Sensibility (1995),Drama|Romance
19 18,Four Rooms (1995),Thriller
20 19,Ace Ventura: When Nature Calls (1995),Comedy
21 20,Money Train (1995),Action
22 21,Get Shorty (1995),Action|Comedy|Drama
23 22,Copcat (1995),Crime|Drama|Thriller

```

Figura 8: Así queda después de reemplazar los símbolos :: por , y lo guardamos con el nombre movies.csv.

- Para importar archivos desde MySQL Workbench damos clic en Table Import Data Wizard y seguimos los pasos del ejemplo 3. ¹.

Así:

```

25 -- Usando como base el archivo movies.dat, limpiarlo e importar los
26 -- datos en la tabla movies creada en el Reto 1.
27
28 • SELECT *
29 FROM movies;

```

#	id_movie	title	genres
1	1	Toy Story (1995)	Animation Children's Comedy
2	2	Jumanji (1995)	Adventure Children's Fantasy
3	3	Grumpier Old Men (1995)	Comedy Romance
4	4	Waiting to Exhale (1995)	Comedy Drama
5	5	Father of the Bride Part II (1995)	Comedy
6	6	Heat (1995)	Action Crime Thriller
7	7	Sabrina (1995)	Comedy Romance

Figura 9: Después de importar movies.csv, hacemos la consulta SELECT * FROM movies;

¹<https://github.com/beduExpert/A1-Introduccion-a-Bases-de-Datos-SG2/blob/main/Sesion-04/Ejemplo-03/Readme.md>

2. Usando como base el archivo ratings.dat, limpiarlo e importar los datos en la tabla ratings creada en el Reto 2.

Respuesta:

- Abrimos con el editor de texto Atom el archivo ratings.dat y agrega la siguiente línea al inicio:

```
1 id_users,id_movie,ratings,time_stamp
```

Figura 10: id_users,id_movie,ratings,time_stamp.

- Ahora reemplaza todos los símbolos :: por ,. El archivo final debe verse así:

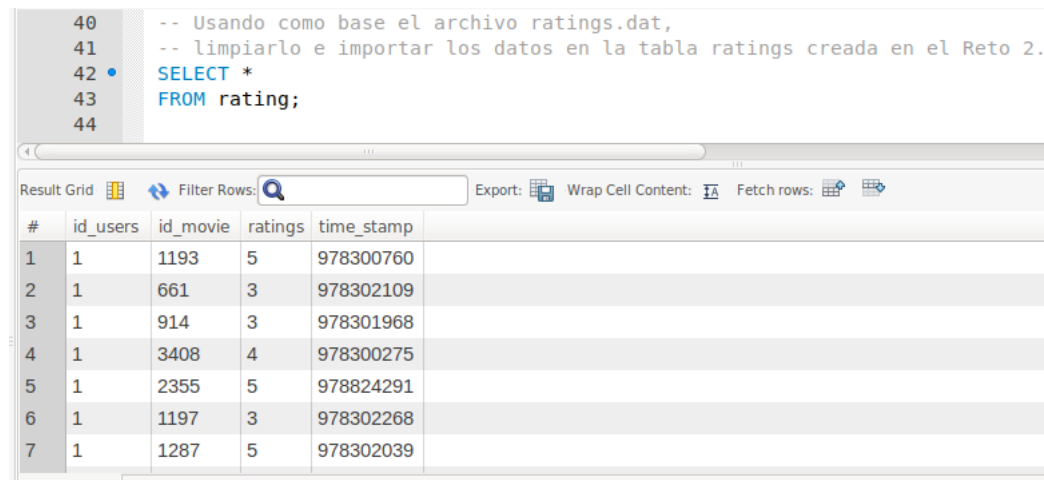
```
ratings.csv
1 id_users,id_movie,ratings,time_stamp
2 1,1193,5,978300760
3 1,661,3,978302109
4 1,914,3,978301968
5 1,3408,4,978300275
6 1,2355,5,978824291
7 1,1197,3,978302268
8 1,1287,5,978302039
9 1,2804,5,978300719
10 1,594,4,978302268
11 1,919,4,978301368
12 1,595,5,978824268
13 1,938,4,978301752
14 1,2398,4,978302281
15 1,2918,4,978302124
16 1,1035,5,978301753
17 1,2791,4,978302188
18 1,2687,3,978824268
19 1,2018,4,978301777
20 1,3105,5,978301713
21 1,2797,4,978302039
22 1,2321,3,978302205
23 1,720,3,978300760
24 1,1270,5,978300055
```

Figura 11: Así queda después de reemplazar los símbolos :: por , y lo guardamos con el nombre ratings.csv.

- Para importar archivos desde MySQL Workbench damos clic en Table Import Data Wizard y seguimos los pasos del ejemplo 3. ².

Así:

```
40 -- Usando como base el archivo ratings.dat,  
41 -- limpiarlo e importar los datos en la tabla ratings creada en el Reto 2.  
42 • SELECT *  
43 FROM rating;  
44
```



The screenshot shows the MySQL Workbench interface. At the top, a SQL editor contains a query to select all data from the 'rating' table. Below the editor, the 'Result Grid' tab is active, displaying the query results in a table format. The table has five columns: '#', 'id_users', 'id_movie', 'ratings', and 'time_stamp'. It contains seven rows of data, all with 'id_users' equal to 1. The interface also includes a 'Filter Rows' search bar and buttons for 'Export', 'Wrap Cell Content', and 'Fetch rows'.

#	id_users	id_movie	ratings	time_stamp
1	1	1193	5	978300760
2	1	661	3	978302109
3	1	914	3	978301968
4	1	3408	4	978300275
5	1	2355	5	978824291
6	1	1197	3	978302268
7	1	1287	5	978302039

Figura 12: Después de importar ratings.csv, hacemos la consulta SELECT * FROM rating;

²<https://github.com/beduExpert/A1-Introduccion-a-Bases-de-Datos-SG2/blob/main/Sesion-04/Ejemplo-03/Readme.md>

Reto 3: Realizando operaciones con Colecciones e importando datos.

Se deberá de crear colecciones e importar los datos de los archivos movies.csv y ratings.csv.

Respuesta:

Primero creamos la base de datos MarcoHurtado:

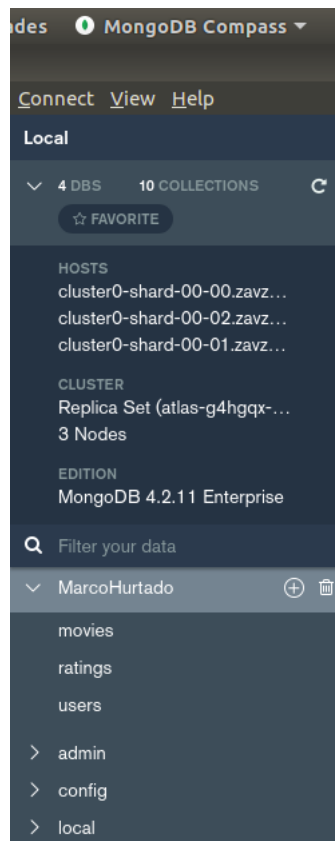






Figura 13: Base de datos MarcoHurtado.

- Creamos la colección movies y ratings.
- Importamos datos a la colección movies y ratings desde el archivo movies.csv y ratings.csv.
- Validamos que cada campo tenga nombre y que estén todos los campos, además validamos la cantidad de documentos.

Así:

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
movies	3,883	101.1 B	383.6 KB	1	60.0 KB	
ratings	1,000,209	100.6 B	95.9 MB	1	9.4 MB	
users	6,040	88.2 B	520.5 KB	1	80.0 KB	

(a) Colecciones

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
movies	3,883	101.1 B	383.6 KB	1	60.0 KB	

(b) Validamos la cantidad de documentos en movies que en este caso debería de ser 3883

```

_id: ObjectId("5fc9c06a927603381e7381d9")
id_movie: "1"
title: "Toy Story (1995)"
genres: "Animation|Children's|Comedy"

>
_id: ObjectId("5fc9c06a927603381e7381da")
id_movie: "2"
title: "Jumanji (1995)"
genres: "Adventure|Children's|Fantasy"

_id: ObjectId("5fc9c06a927603381e7381db")
id_movie: "3"
title: "Grumpier Old Men (1995)"
genres: "Comedy|Romance"

_id: ObjectId("5fc9c06a927603381e7381dc")
id_movie: "4"
title: "Waiting to Exhale (1995)"
genres: "Comedy|Drama"

```

(c) Validamos los campos en movies.

Figura 14: Creamos la colección movies e Importamos datos a la colección movies desde el archivo movies.csv.

ratings	1,000,209	100.6 B	95.9 MB	1	9.4 MB	
---------	-----------	---------	---------	---	--------	--

(a) Validamos la cantidad de documentos en ratings que en este caso debería de ser 1000209 documentos.

```

_id: ObjectId("5fc9ceb927603381e739104")
id_users: "1"
id_movie: "1193"
ratings: "5"
time_stamp: "978300760"

_id: ObjectId("5fc9ceb927603381e739105")
id_users: "1"
id_movie: "661"
ratings: "3"
time_stamp: "978302109"

> _id: ObjectId("5fc9ceb927603381e739106")
id_users: "1"
id_movie: "914"
ratings: "3"
time_stamp: "978301968"

```

(b) Validamos los campos en ratings

Figura 15: Creamos la colección ratings e Importamos datos a la colección ratings desde el archivo ratings.csv..