

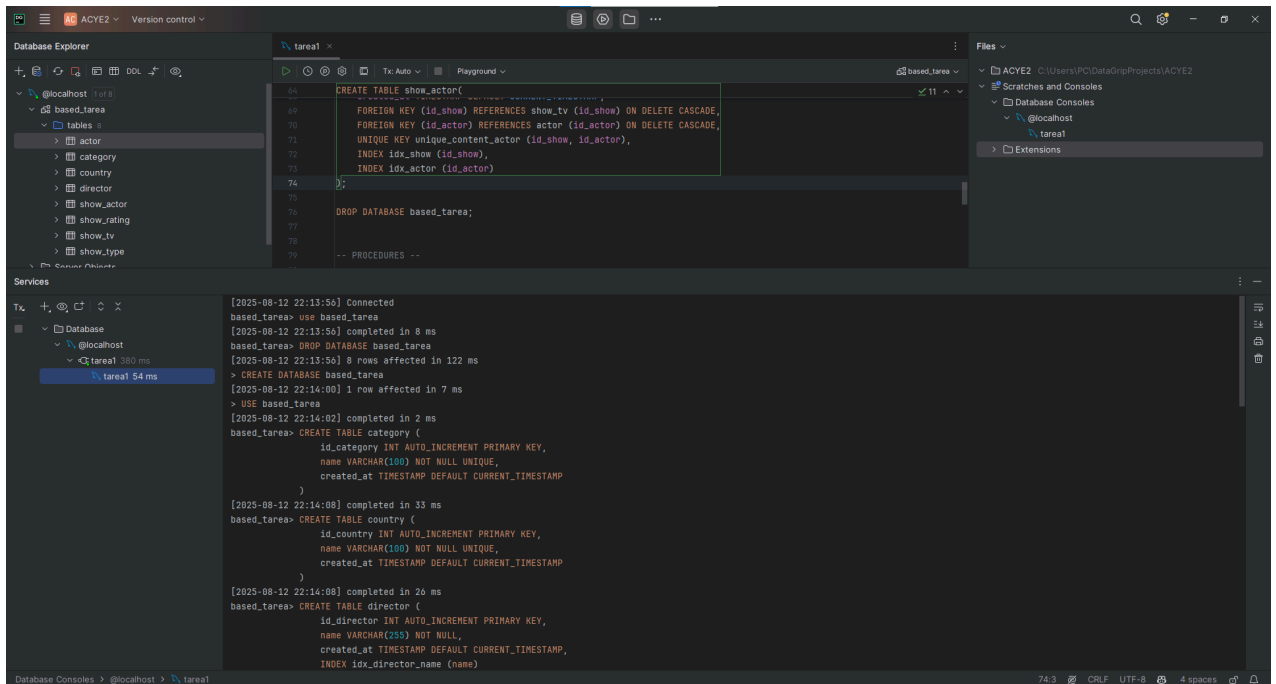
Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Bases de Datos 2
Segundo Semestre 2025



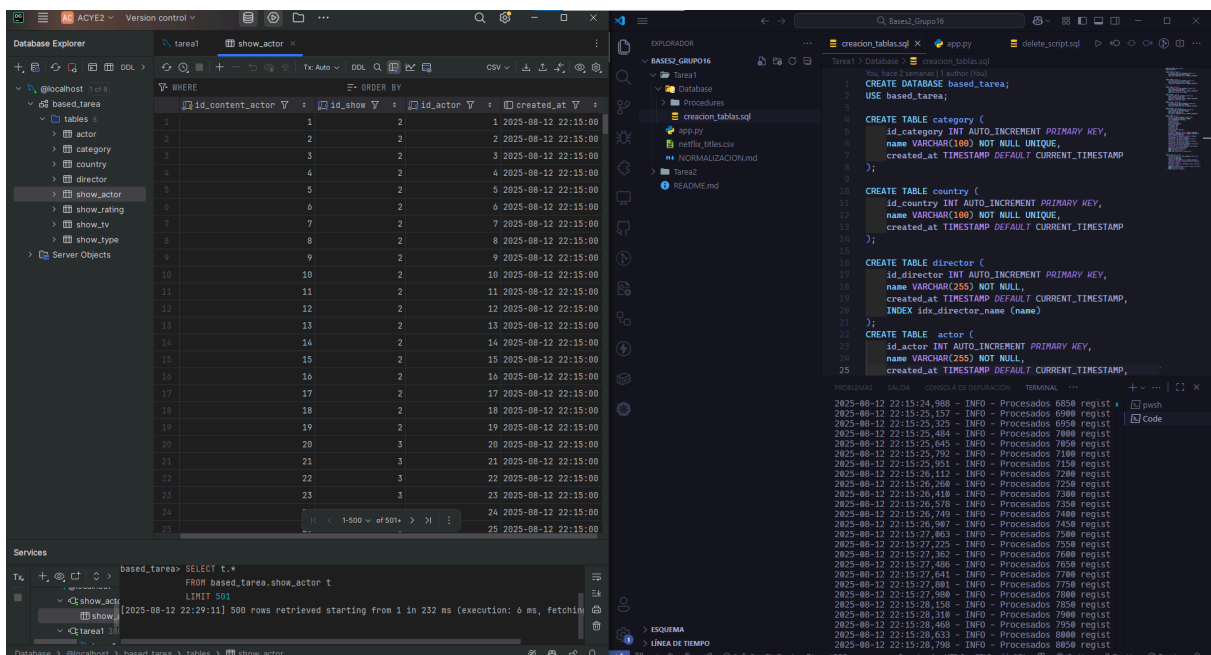
Tarea 2

Nombre	Carnet
Susan Pamela Herrera Monzon	201612218
Josué Nabí Hurtarte Pinto	202202481
Naomi Rashel Yos Cujcuj	202001814

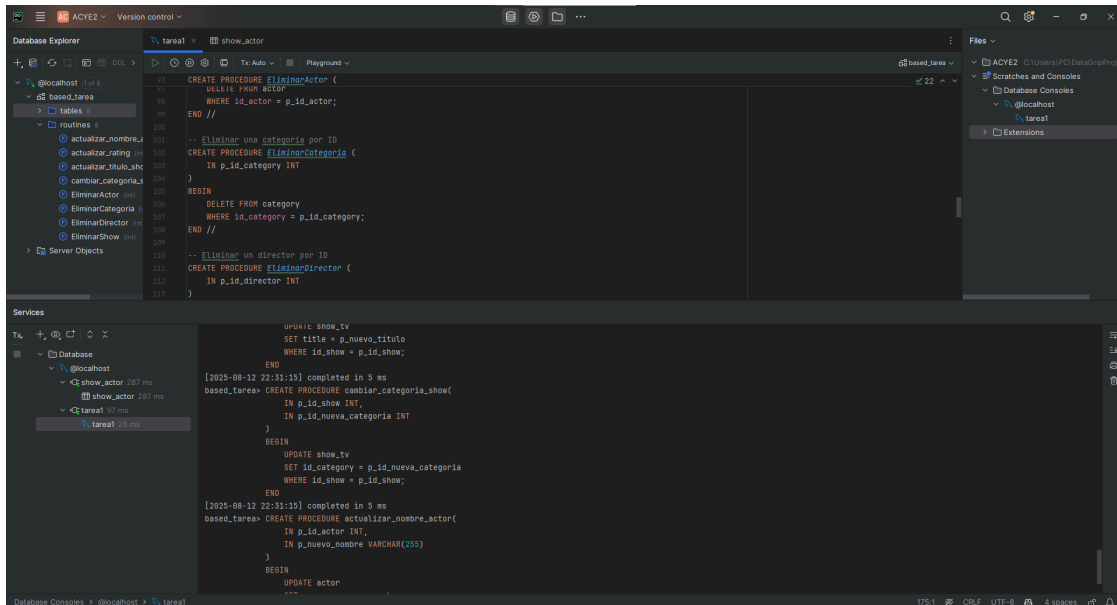
1. Ejecutar el script de la tarea 1



2. Cargar los datos de la tarea 1



3. Cargar los procedimientos de la tarea 1



4. Realizar respaldo completo de la backup

Para esta tarea utilizamos MySQL, y el respaldo completo se realiza mediante el comando ***mysqldump***.

La forma más básica de hacerlo es la siguiente, la cual genera un backup con la estructura y los datos de la base:

mysqldump -u root -p based_tarea > respaldo_completo_antes.sql

Sin embargo, existe una opción más avanzada que incluye parámetros adicionales para respaldar no solo los datos y la estructura, sino también objetos como procedimientos almacenados, triggers y eventos:

mysqldump -u root -p --single-transaction --routines --triggers --events based_tarea > respaldo_completo_antes.sql

A continuación, explicamos los parámetros más relevantes:

- single-transaction: Asegura consistencia de datos usando una transacción
- routines: Incluye procedimientos almacenados y funciones
- triggers: Incluye triggers
- events: Incluye eventos programados

En nuestro caso, utilizaremos esta opción avanzada, ya que la base de datos contiene procedimientos almacenados e índices.

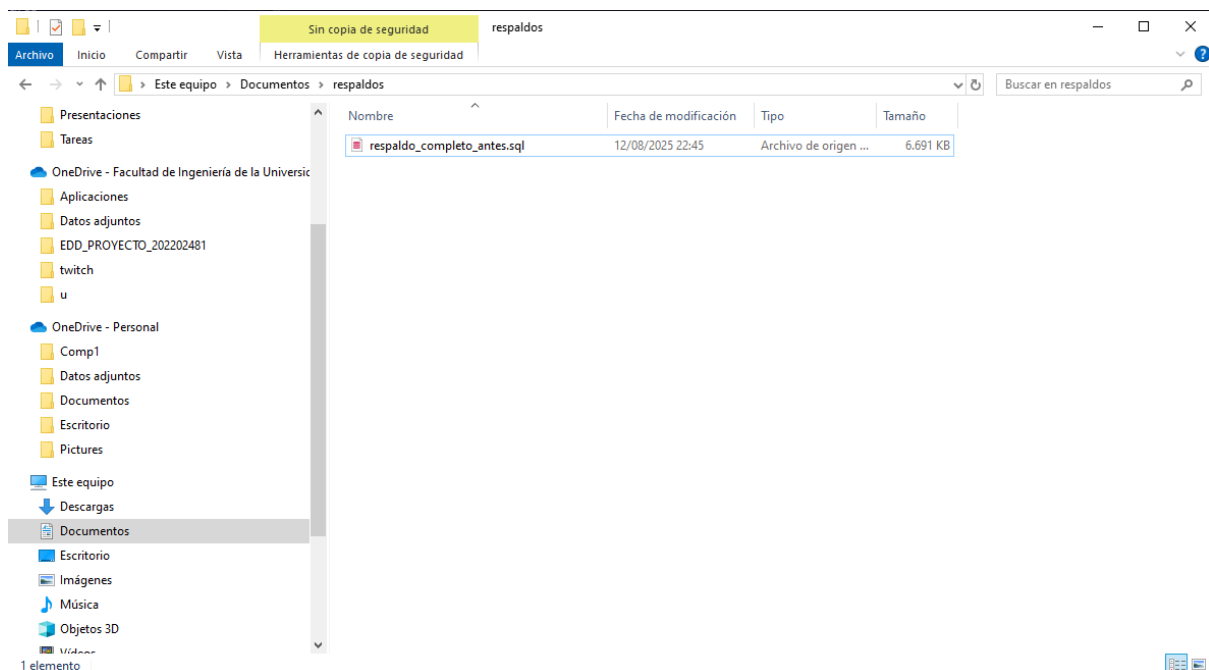
Además, para que el respaldo se guarde en una carpeta específica y el script generado incluya las sentencias **CREATE DATABASE** y **DROP DATABASE IF EXISTS**, utilizamos el parámetro **--add-drop-database** y especificamos la ruta de salida:

```
mysqldump -u root -p --single-transaction --routines --triggers --events --add-drop-database --databases based_tarea >C:\Users\PC\Documents\respaldos\respaldo_completo_antes.sql
```

Al ejecutarlo nos pide la contraseña de MySQL que en mi caso es “root”

```
2025-08-12 22:15:31,246 - INFO - [Normalización completada exitosamente!]  
[22:15] Shell main = 31s 190ms  
~\OneDrive - Facultad de Ingeniería de la Universidad de San Carlos de Guatemala\u\clases ocatvo semestre\Bases2\Lab\Tarea1\Base  
s2_Grupo16  
> mysqldump -u root -p --single-transaction --routines --triggers --events --add-drop-database --databases based_tarea >C:\Users\PC\Documents\respaldos\respaldo_completo_antes.sql  
Enter password: 
```

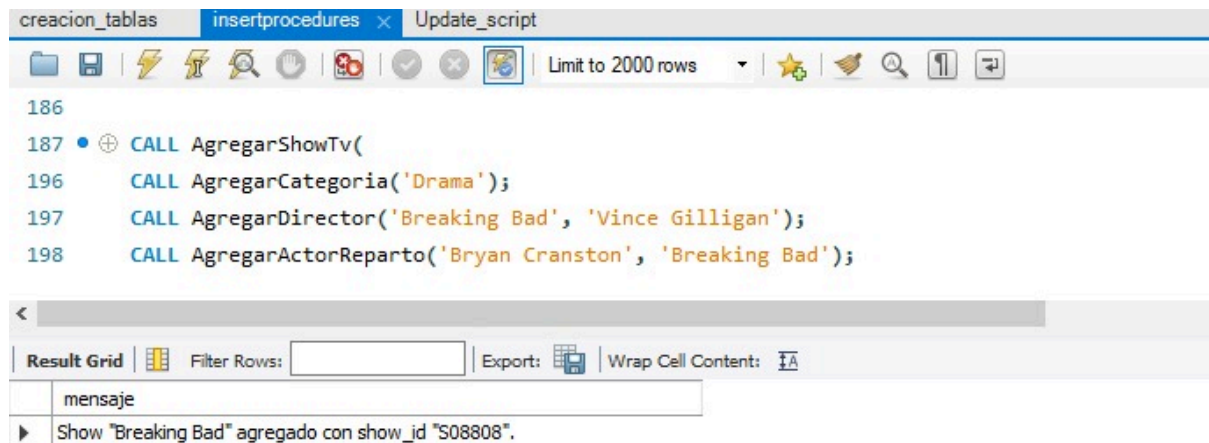
Como podemos observar ya lo creo perfectamente en el directorio especificado.



5. Ejecutar los procedimientos creados con al menos 3 datos distintos

Se ejecutaron los cuatro inserts y se llevaron a cabo los 4 updates.

- captura Insert / Updates

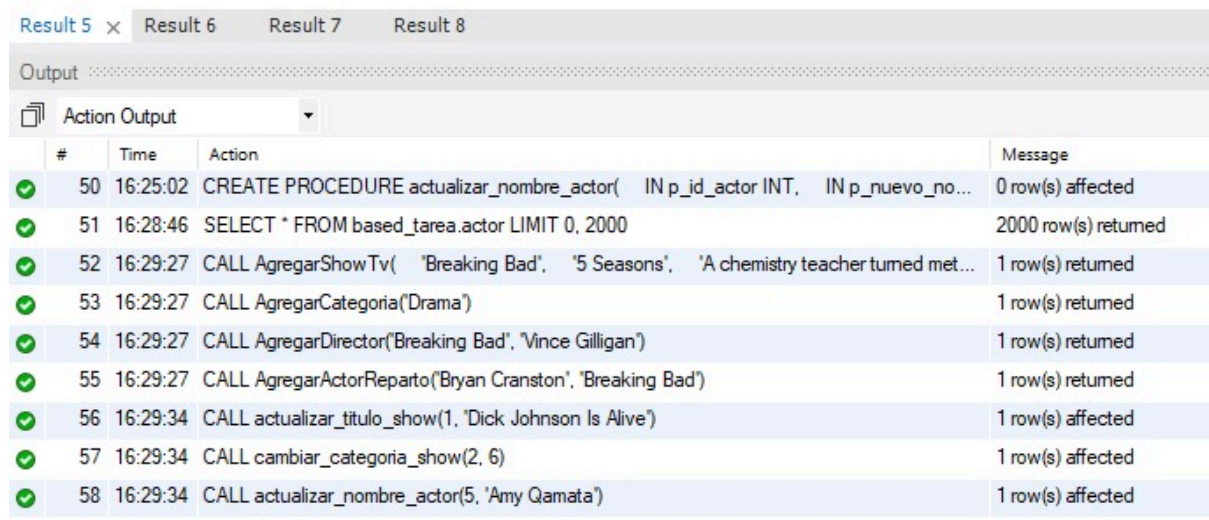


```
186
187 • CALL AgregarShowTv(
196   CALL AgregarCategoria('Drama');
197   CALL AgregarDirector('Breaking Bad', 'Vince Gilligan');
198   CALL AgregarActorReparto('Bryan Cranston', 'Breaking Bad');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

mensaje

Show "Breaking Bad" agregado con show_id "S08808".

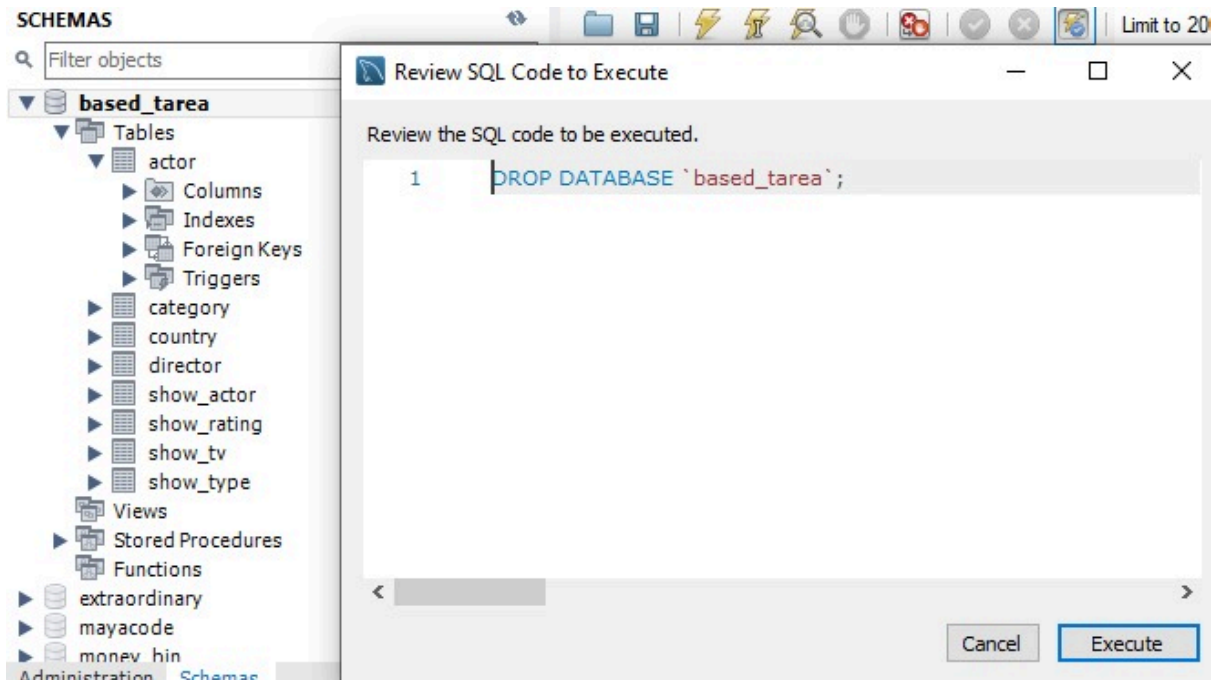


#	Time	Action	Message
✓ 50	16:25:02	CREATE PROCEDURE actualizar_nombre_actor(IN p_id_actor INT, IN p_nuevo_no...	0 row(s) affected
✓ 51	16:28:46	SELECT * FROM based_tarea.actor LIMIT 0, 2000	2000 row(s) returned
✓ 52	16:29:27	CALL AgregarShowTv('Breaking Bad', '5 Seasons', 'A chemistry teacher turned met...	1 row(s) returned
✓ 53	16:29:27	CALL AgregarCategoria('Drama')	1 row(s) returned
✓ 54	16:29:27	CALL AgregarDirector('Breaking Bad', 'Vince Gilligan')	1 row(s) returned
✓ 55	16:29:27	CALL AgregarActorReparto('Bryan Cranston', 'Breaking Bad')	1 row(s) returned
✓ 56	16:29:34	CALL actualizar_titulo_show(1, 'Dick Johnson Is Alive')	1 row(s) affected
✓ 57	16:29:34	CALL cambiar_categoria_show(2, 6)	1 row(s) affected
✓ 58	16:29:34	CALL actualizar_nombre_actor(5, 'Amy Qamata')	1 row(s) affected

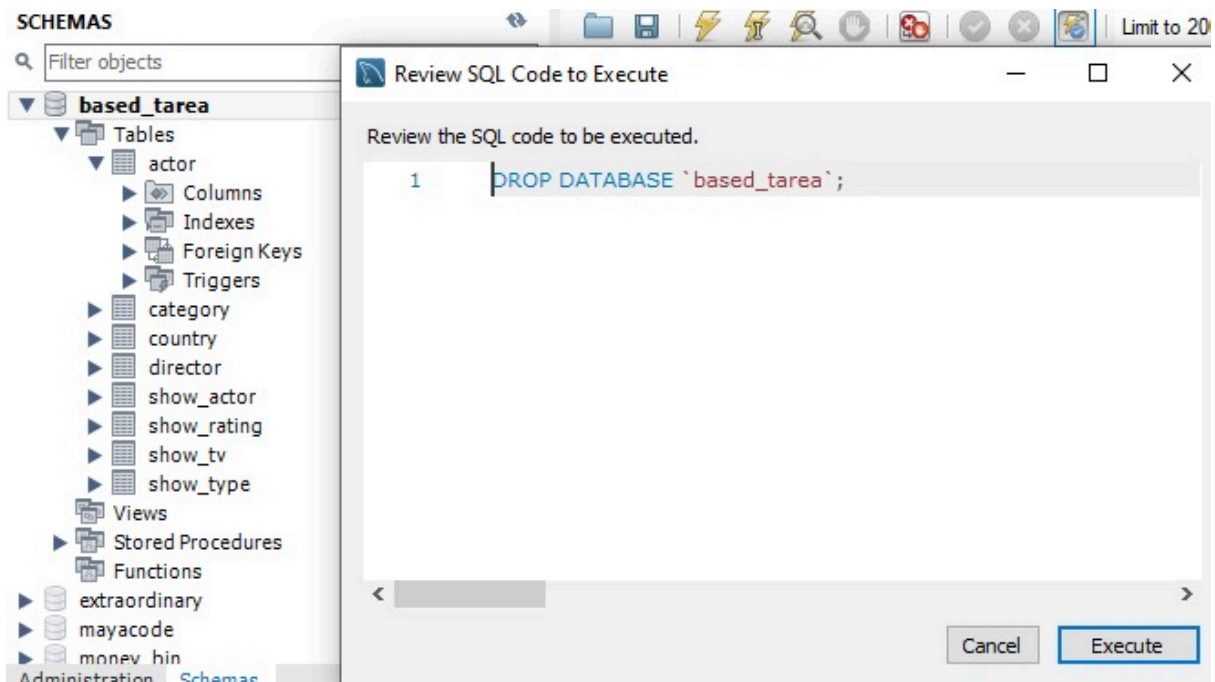
En la captura se puede apreciar que se insertaron los registros y actualizaron de manera correcta.

6. Realizar nuevamente el backup completo

- Captura de existencia de esquemas



- Captura de executed



- Ejecución de Backup

```

1  -- MySQL dump 10.13  Distrib 8.0.32, for Win64 (x86_64)
2  --
3  -- Host: localhost    Database: based_tarea
4  --
5  -- Server version  8.0.32
6
7  /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8  /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9  /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --

```

#	Time	Action	Message	Duration / Fetch
211	16:35:50	/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */	0 row(s) affected	0.000 sec
212	16:35:50	/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */	0 row(s) affected	0.000 sec
213	16:35:50	/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */	0 row(s) affected	0.000 sec
214	16:35:50	/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */	0 row(s) affected	0.000 sec
215	16:35:50	/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */	0 row(s) affected	0.000 sec

Se corrieron nuevamente los comando anteriormente descritos para los backups

#	Time	Action	Message	Duration / Fetch
210	16:35:50	/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */	0 row(s) affected	0.000 sec
211	16:35:50	/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */	0 row(s) affected	0.000 sec
212	16:35:50	/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */	0 row(s) affected	0.000 sec
213	16:35:50	/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */	0 row(s) affected	0.000 sec
214	16:35:50	/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */	0 row(s) affected	0.000 sec
215	16:35:50	/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */	0 row(s) affected	0.000 sec
216	16:36:24	SELECT * FROM based_tarea.actor LIMIT 0, 2000	2000 row(s) returned	0.000 sec / 0.000 sec
217	16:36:42	SELECT * FROM based_tarea.director LIMIT 0, 2000	2000 row(s) returned	0.000 sec / 0.000 sec

En esta captura se observa que ya hay datos existentes nuevamente actualizados.

7. Respaldo Incremental

Debido a que trabajamos en la base de datos de manera local primero hice un backup completo, para eso utilicé mysqldump, además que lo trabajé desde un contenedor de docker, así que esto fueron los pasos que realicé:

1. Desde CMD puse el siguiente comando para realizar un backup completo:

```
docker exec -i mysql-server mysqldump -u root -p123456 --single-transaction
--routines --triggers --events --add-drop-database --databases based_tarea >
"C:\Users\Naomi Rashel\Documents\2do semestre 2025\Lab
bases2\Bases2_Grupo16\Tarea1\Database\Procedures\Respaldo\respaldo_com
pleto.sql"
```

2. Modifiqué la tabla show_tv por medio de un procedimiento de actualización:

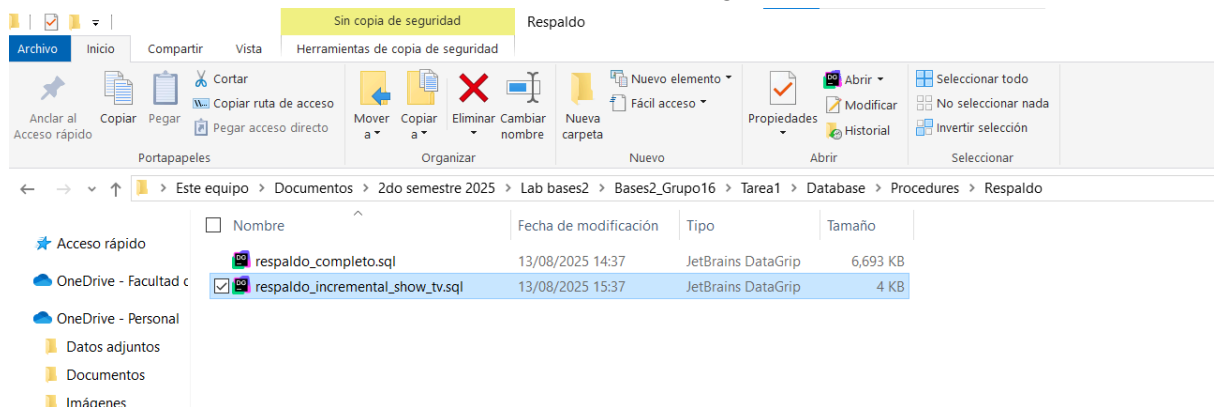
```
CALL cambiar_categoria_show(3, 6);
CALL actualizar_titulo_show(3, 'El show de Naomi');
```

3. Ya que lo estaba realizando en un contenedor para simular el backup incremental utilicé el siguiente comando

```
docker exec -i mysql-server mysqldump -u root -p123456 based_tarea show_tv
--where="updated_at >= '2025-08-13 14:37:26'" > "C:\Users\Naomi
Rashel\Documents\2do semestre 2025\Lab
bases2\Bases2_Grupo16\Tarea1\Database\Procedures\Respaldo\respaldo_incr
emental_show_tv.sql"
```

En donde --where="updated_at >= '2025-08-13 14:37:26'" en donde se incluyen los registros cuyos campos sean posteriores a la fecha y hora indicada.

y > "C:\Users\Naomi Rashel\Documents\2do semestre 2025\Lab bases2\Bases2_Grupo16\Tarea1\Database\Procedures\Respaldo\respaldo_incremental_show_tv.sql" indica la ruta donde se desea guardar el backup.



8. Conclusión

Se aprecia en el backup completo las inserciones que se realizaron, luego de los datos cargados del archivo csv.