NOMBRE DEL TEMA:	Operadores Set, Joins y Subcosultas			No.	4
Asignatura:	TALLER DE BASE DE DATOS	Carrera:	ISIC	Duración de la práctica (Hrs)	5

Jocelin Reyes Rodriguez

I. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

III. Material empleado:

Equipo de Cómputo

IV. Desarrollo de la práctica:

CONSULTAS Y SUBCONSULTAS EN MySQL

```
2  -- Calcular el promedio de la longitud de 'first_name' en la tabla 'actor'
3  SELECT AVG (LENGTH(first_name)) FROM actor;
```

Message | Summary | Result 1

Data | Info — Cell Editor | Data Profiling

AVG (LENGTH(first_nam
5.3050

```
5  -- Seleccionar actores con longitud de 'first_name' mayor al promedio
6  SELECT  * FROM actor WHERE LENGTH(first_name) > (SELECT AVG(LENGTH(first_name
   )) FROM actor);
7
```

Message | Summary | Result 1

Data | Info — Cell Editor | Data Profiling | Export

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINESS	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33
14	VIVIEN	BERGEN	2006-02-15 04:34:33
20	LUCILLE	TRACY	2006-02-15 04:34:33
21	KIRSTEN	PALTROW	2006-02-15 04:34:33
23	SANDRA	KILMER	2006-02-15 04:34:33

-- Seleccionar actores con longitud de 'first_name' mayor al promedio SELECT * FROM actor WHERE LENGTH(first_name) > (SELECT AV(| Record 1 of 82 | Ln 4, Col 1 | Elapsed Time

```
 8  -- Enceuntra los actores que han participado en películas de la categoría
    comedy
 9    -- Hechos por separado
10  SELECT category_id FROM category WHERE name = "Comedy";
```

Message Summary Result 1

Data Info — Cell Editor | Data Profiling | Export

category_id
5

```
12  SELECT film_id FROM film_category WHERE category_id = 5;
13
```

Message Summary Result 1

Data Info — Cell Editor | Data Profiling | Export | Pin

film_id
7
28
99
119

SELECT film_id FROM film_category WHERE category_id = 5 | Read Only | Record 1 of 58 | Ln 11, Col 56 | Elapsed Time: 0.346

```
14    -- Uniendo ambas QUERY
15  SELECT film_id FROM film_category WHERE category_id = (SELECT category_id FROM
     category WHERE name = "Comedy");
16
```

Message Summary Result 1

Data Info — Cell Editor | Data Profiling | Export | Pin

film_id
7
28
99
119

SELECT film_id FROM film_category WHERE category_id = (SELECT category_id FROM category WHERE name = "Comedy") | Read Only | Record 1 of 58 | Ln 13, Col 3 | Elapsed Time: 0.31

```
16    -- Para listas hacemos uso del IN y del NOT IN
17  SELECT  actor_id FROM film_actor WHERE film_id IN (7,28,99);
18
```

Message Summary Result 1

Data Info — Cell Editor | Data Profiling | Export | Pin

actor_id
99
133
162
170

SELECT actor_id FROM film_actor WHERE film_id IN (7,28,99) | Read Only | Record 1 of 9 | Ln 15, Col 1 | Elapsed Time: 0.3

```
19    -- Uniendo QUERY
20  SELECT  actor_id FROM film_actor WHERE film_id IN (SELECT film_id FROM
    film_category WHERE category_id = (SELECT category_id FROM category WHERE name
     = "Comedy"));
21
```

Message Summary Result 1

Data Info — Cell Editor | Data Profiling | Export | Pin

actor_id
99
133
162
170
185

-- Uniendo QUERY SELECT actor_id FROM film_actor WHERE film_id IN (SELECT film_id FROM film_category WHERE category_id = (SELE | Read Only | Record 1 of 286 | Ln 18, Col 1 | Elapsed Time: 0.3

```
22    -- Haciendo el último SELECT
23    SELECT first_name, last_name FROM actor WHERE actor_id IN (SELECT  actor_id
  FROM film_actor WHERE film_id IN (SELECT film_id FROM film_category WHERE
  category_id = (SELECT category_id FROM category WHERE name = "Comedy")));
```

Message Summary Result 1

Data Info — Cell Editor | Data Profiling | Export | Pin

first_name	last_name
JIM	MOSTEL
RICHARD	PENN
OPRAH	KILMER
MENA	HOPPER
MICHAEL	BOLGER

SELECT first_name, last_name FROM actor WHERE actor_id IN (SELECT actor_id FROM film_actor WHERE film_id IN (SELECT film_id FRO | Read Only | Record 1 of 147 | Ln 21, Col 1 | Elapsed Time: 0.3

```
1  -- Encuentra los títulos de las películas que nunca han sido alquiladas
2  -- Averiguar que ID estan en la tabla
3  SELECT rental_id FROM rental WHERE rental_id ;
4
```

Message Summary Result 1 Explain 1

Data Info — Cell Editor | Data Profiling | Export | Pin

rental_id
1
2
3
4
5
6

```sql
-- Inventario
SELECT film_id FROM inventory WHERE inventory_id IN ( SELECT inventory_id FROM
rental WHERE rental_id );
```

Message Summary Result 1 Explain 1

Data Info Cell Editor Data Profiling Export

film_id
1
1
1
1
1
1
1
2
2
2

SELECT film_id FROM inventory WHERE inventory_id IN (SELECT inventory_id FROM rental WHERE rental_id) Read Only Record 1 of 4580 Ln 5, Col 14 Elapsed Time:

```sql
-- FILM

SELECT title FROM film WHERE film_id NOT IN (SELECT film_id FROM inventory WHERE
inventory_id IN ( SELECT inventory_id FROM rental WHERE rental_id ));
```

Message Summary Result 1 Explain 1

Data Info Cell Editor Data Profiling Export Pin

title
ALICE FANTASIA
APOLLO TEEN
ARGONAUTS TOWN
ARK RIDGEMONT

SELECT title FROM film WHERE film_id NOT IN (SELECT film_id FROM inventory WHERE inventory_id IN (SELECT inventory_id FROM rent Read Only Record 1 of 42 Ln 7, Col 1 Elapsed Time: 0.34

JOINS EN MySQL

Objects * functions @sakila (mysql8-localhost) ... peliculas @sakila (mysql8-localhost) - ... JOIN @sakila (mysql8-localhost) - Query

Save Query Builder Beautify SQL Code Snippet

mysql8-localhost sakila Run Stop Explain

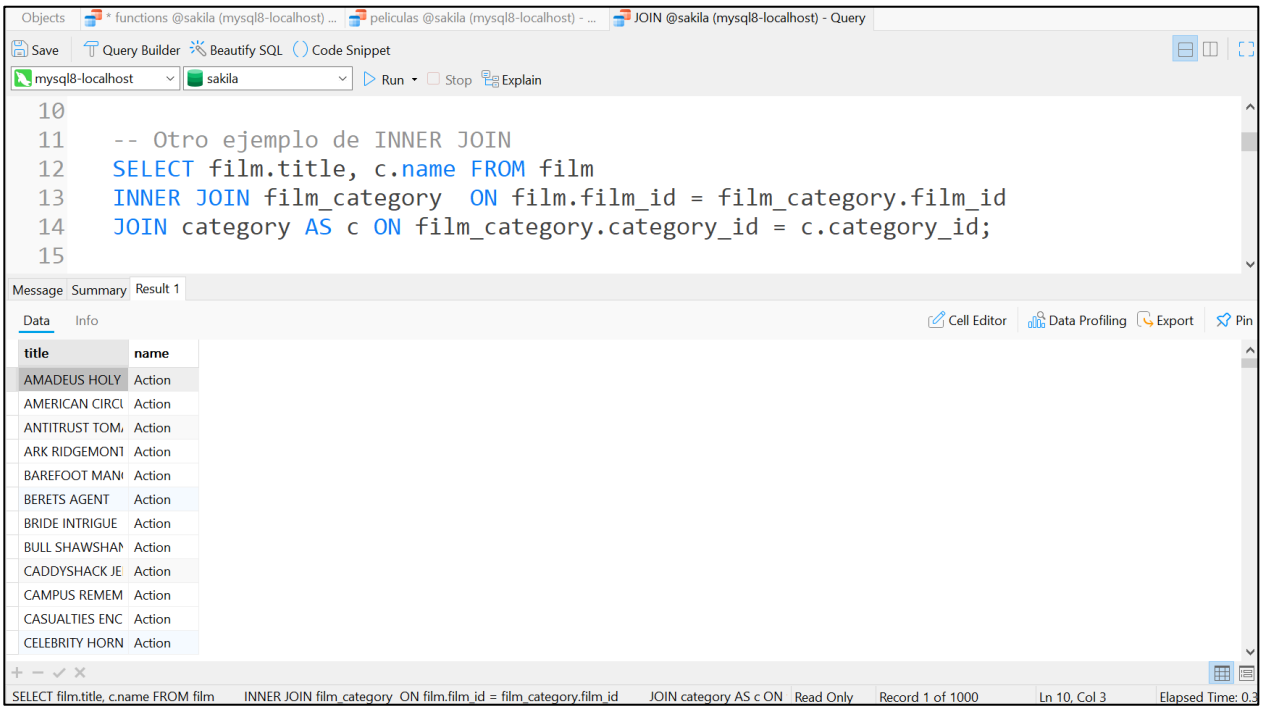
```sql
-- JOINS en MySQL
  -- INNER JOIN : cuando coinciden ambas tablas
  SELECT f.title, c.name AS category_name
  FROM film AS f
  INNER JOIN film_category AS fc
  ON f.film_id = fc.film_id
  INNER JOIN category AS c
  ON fc.category_id = c.category_id
  WHERE c.name = "Comedy";

```

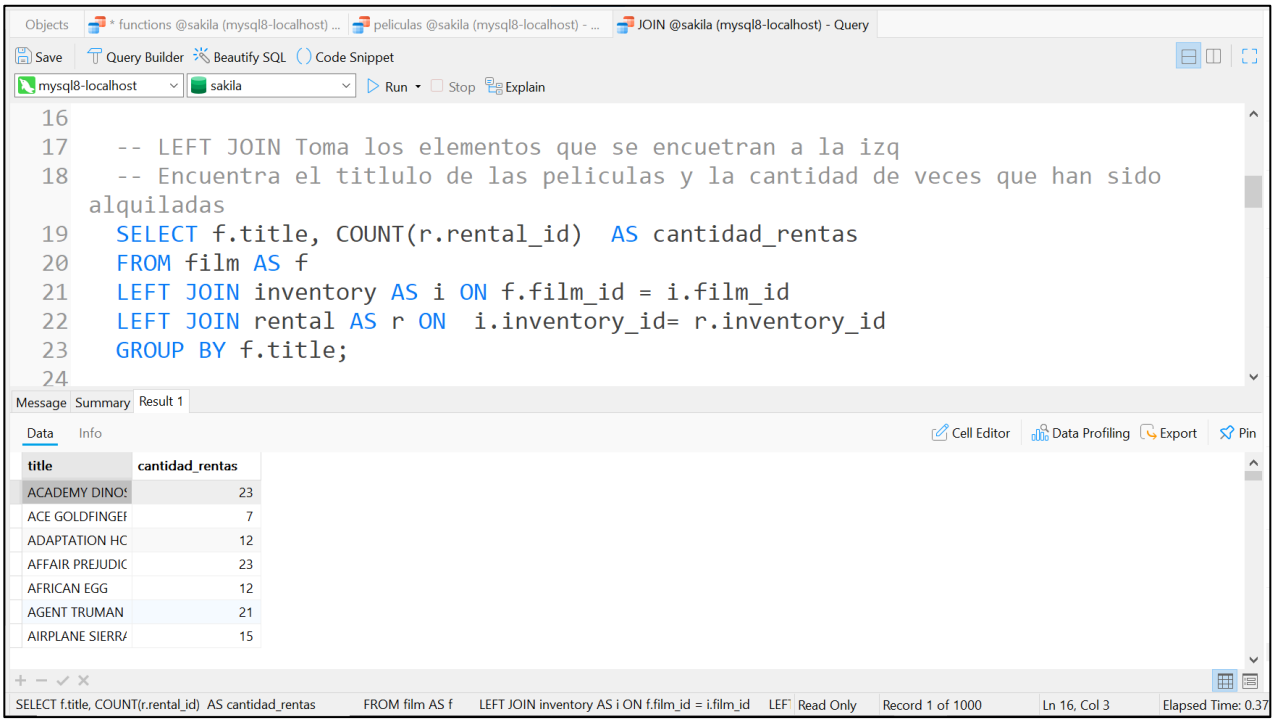
Message Summary Result 1

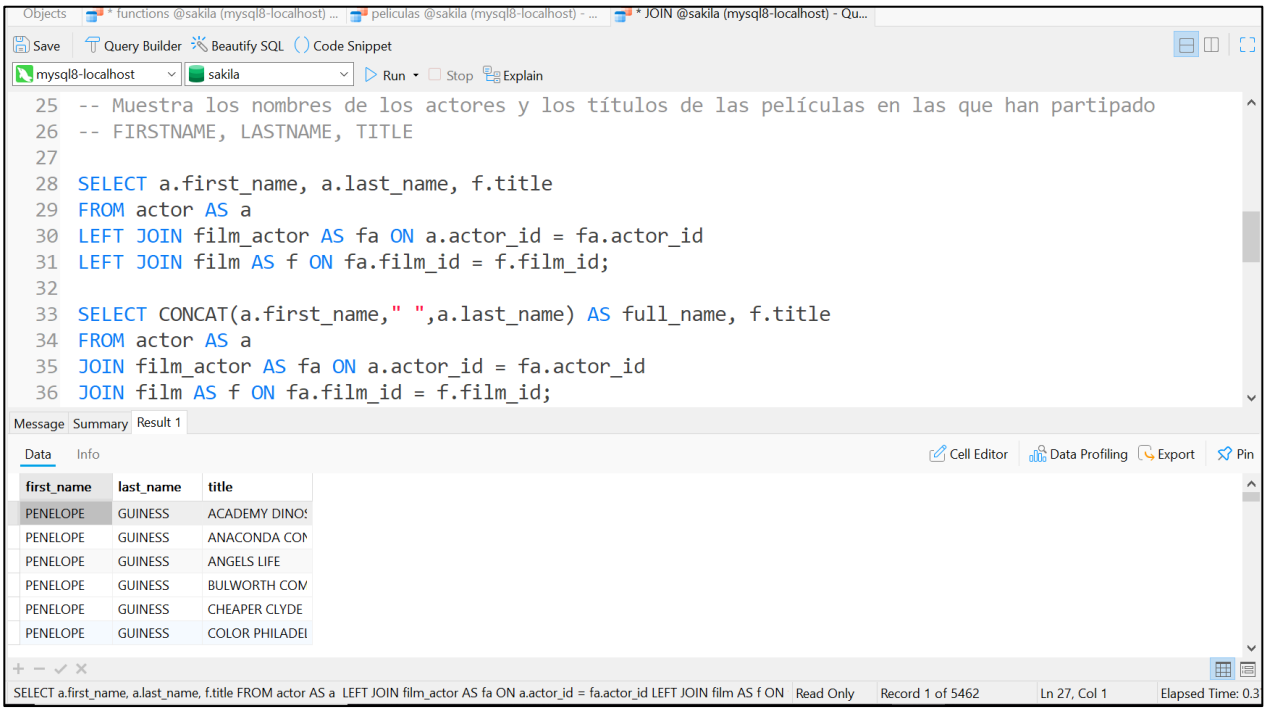
Data Info Cell Editor Data Profiling Export Pin

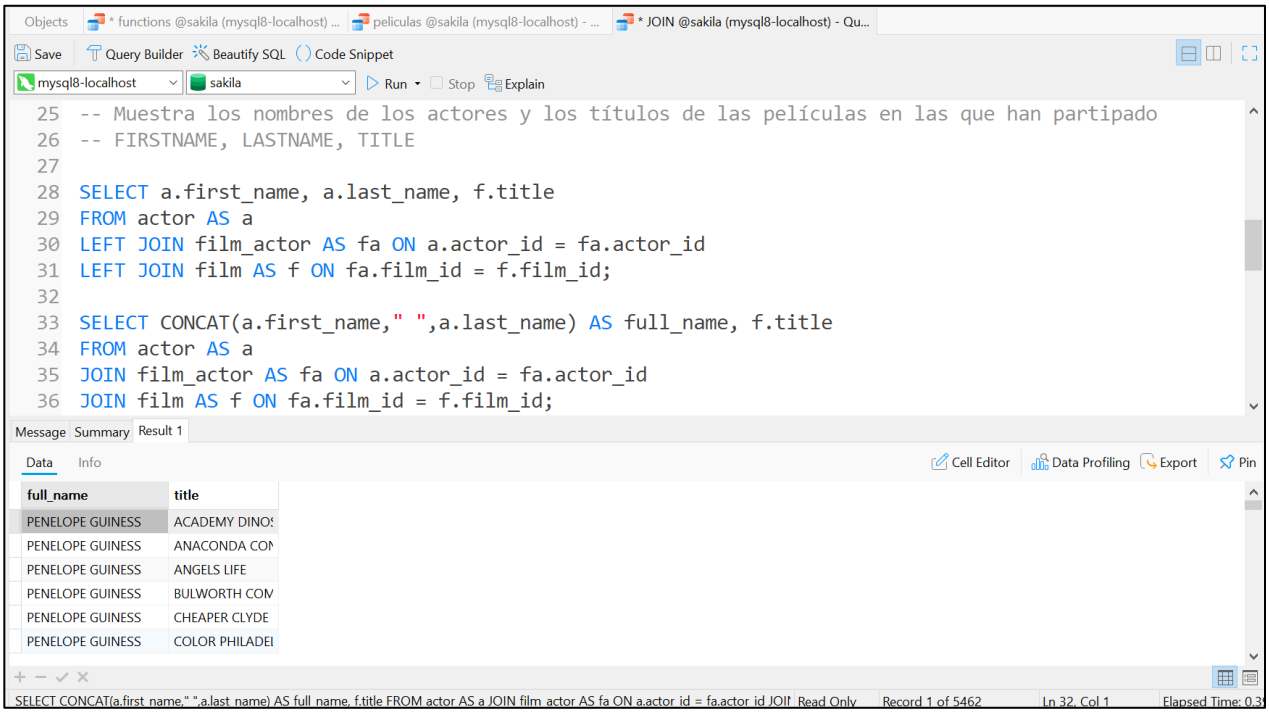
title	category_name
AIRPLANE SIERRA	Comedy
ANTHEM LUKE	Comedy
BRINGING HYSTE	Comedy
CAPER MOTIONS	Comedy
CAT CONEHEADS	Comedy
CLOSER BANG	Comedy
CONNECTION MI	Comedy

SELECT f.title, c.name AS category_name FROM film AS f INNER JOIN film_category AS fc ON f.film_id = fc.film_id INNER JOIN cate Read Only Record 1 of 58 Ln 2, Col 3 Elapsed Time: 0.3

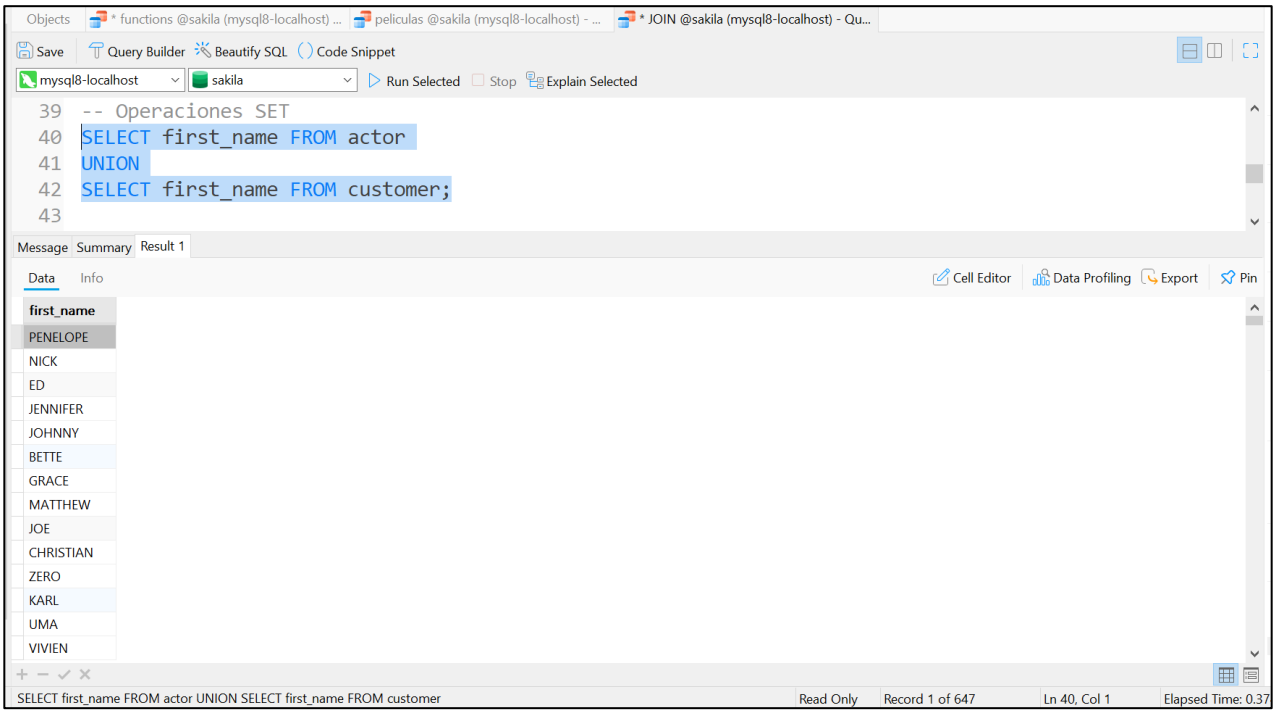

Objects
* functions @sakila (mysql8-localhost) ...
peliculas @sakila (mysql8-localhost) - ...
JOIN @sakila (mysql8-localhost) - Query
Save
Query Builder
Beautify SQL
Code Snippet
mysql8-localhost
sakila
Run
Stop
Explain
10
11 -- Otro ejemplo de INNER JOIN
12 SELECT film.title, c.name FROM film
13 INNER JOIN film_category ON film.film_id = film_category.film_id
14 JOIN category AS c ON film_category.category_id = c.category_id;
15
Message
Summary
Result 1
Data
Info
Cell Editor
Data Profiling
Export
Pin
title
name
AMADEUS HOLY Action
AMERICAN CIRCU Action
ANTITRUST TOM Action
ARK RIDGEMONT Action
BAREFOOT MAN Action
BERETS AGENT Action
BRIDE INTRIGUE Action
BULL SHAWSHAN Action
CADDYSHACK JE Action
CAMPUS REMEM Action
CASUALTIES ENC Action
CELEBRITY HORN Action
SELECT film.title, c.name FROM film
INNER JOIN film_category ON film.film_id = film_category.film_id
JOIN category AS c ON
Read Only
Record 1 of 1000
Ln 10, Col 3
Elapsed Time: 0.3

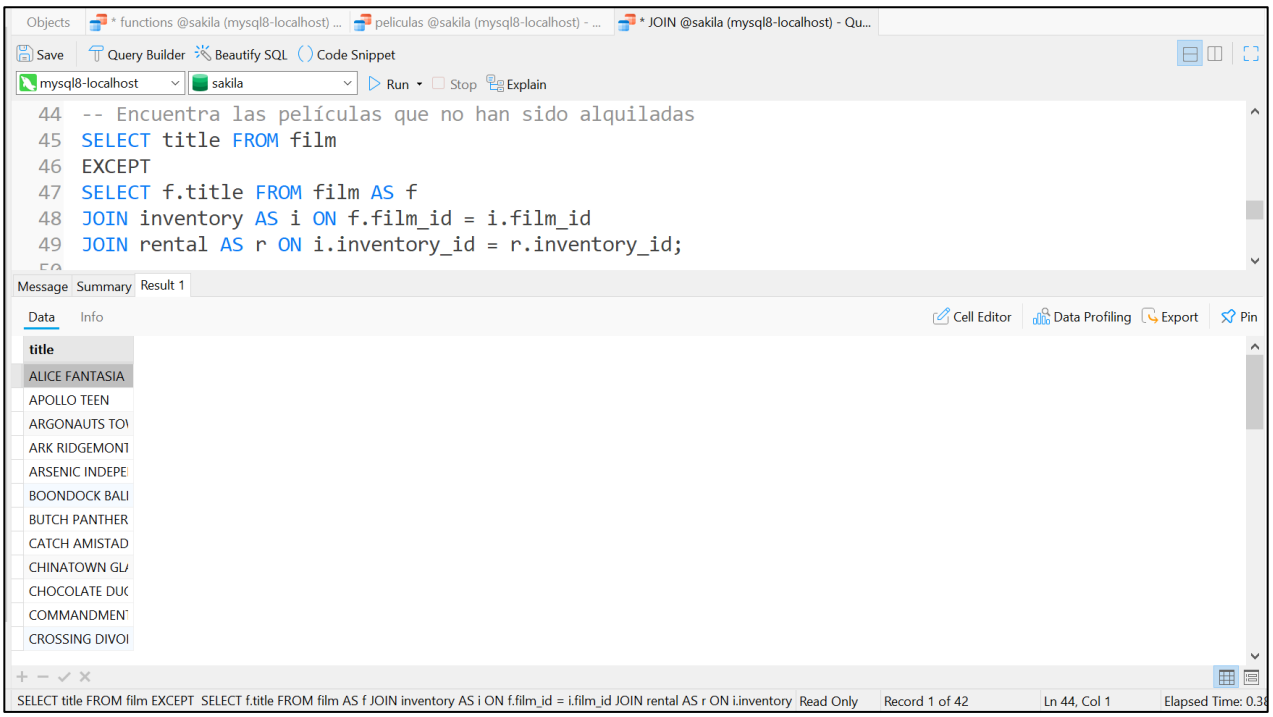

Objects
* functions @sakila (mysql8-localhost) ...
peliculas @sakila (mysql8-localhost) - ...
JOIN @sakila (mysql8-localhost) - Query
Save
Query Builder
Beautify SQL
Code Snippet
mysql8-localhost
sakila
Run
Stop
Explain
16
17 -- LEFT JOIN Toma los elementos que se encuetran a la izq
18 -- Encuentra el titlulo de las peliculas y la cantidad de veces que han sido alquiladas
19 SELECT f.title, COUNT(r.rental_id) AS cantidad_rentas
20 FROM film AS f
21 LEFT JOIN inventory AS i ON f.film_id = i.film_id
22 LEFT JOIN rental AS r ON i.inventory_id= r.inventory_id
23 GROUP BY f.title;
24
Message
Summary
Result 1
Data
Info
Cell Editor
Data Profiling
Export
Pin
title
cantidad_rentas
ACADEMY DINOS 23
ACE GOLDFINGEF 7
ADAPTATION HC 12
AFFAIR PREJUDIC 23
AFRICAN EGG 12
AGENT TRUMAN 21
AIRPLANE SIERRA 15
SELECT f.title, COUNT(r.rental_id) AS cantidad_rentas
FROM film AS f
LEFT JOIN inventory AS i ON f.film_id = i.film_id
LEF
Read Only
Record 1 of 1000
Ln 16, Col 3
Elapsed Time: 0.37

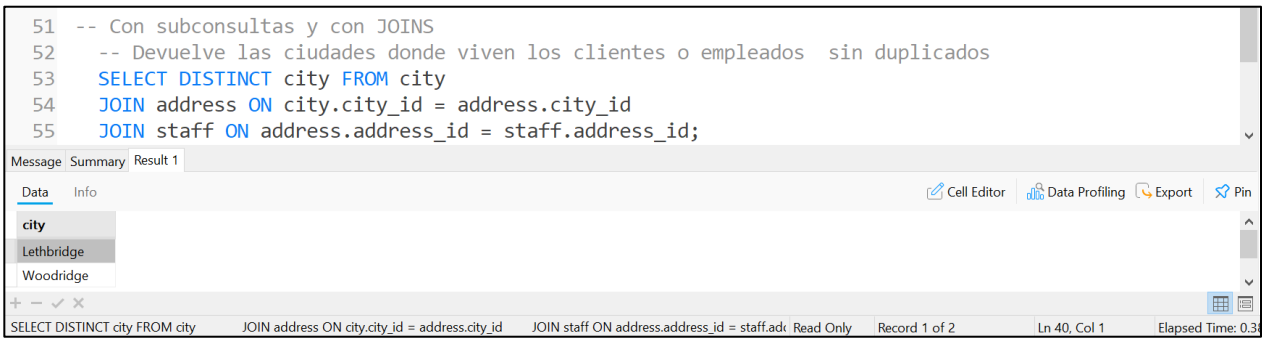

Objects
* functions @sakila (mysql8-localhost) ...
peliculas @sakila (mysql8-localhost) - ...
* JOIN @sakila (mysql8-localhost) - Qu...
Save
Query Builder
Beautify SQL
Code Snippet
mysql8-localhost
sakila
Run
Stop
Explain
25 -- Muestra los nombres de los actores y los títulos de las películas en las que han partipado
26 -- FIRSTNAME, LASTNAME, TITLE
27
28 SELECT a.first_name, a.last_name, f.title
29 FROM actor AS a
30 LEFT JOIN film_actor AS fa ON a.actor_id = fa.actor_id
31 LEFT JOIN film AS f ON fa.film_id = f.film_id;
32
33 SELECT CONCAT(a.first_name," ",a.last_name) AS full_name, f.title
34 FROM actor AS a
35 JOIN film_actor AS fa ON a.actor_id = fa.actor_id
36 JOIN film AS f ON fa.film_id = f.film_id;
Message
Summary
Result 1
Data
Info
Cell Editor
Data Profiling
Export
Pin
first_name
last_name
title
PENELOPE GUINESS ACADEMY DINOS
PENELOPE GUINESS ANACONDA CON
PENELOPE GUINESS ANGELS LIFE
PENELOPE GUINESS BULWORTH COM
PENELOPE GUINESS CHEAPER CLYDE
PENELOPE GUINESS COLOR PHILADEI
SELECT a.first_name, a.last_name, f.title FROM actor AS a LEFT JOIN film_actor AS fa ON a.actor_id = fa.actor_id LEFT JOIN film AS f ON
Read Only
Record 1 of 5462
Ln 27, Col 1
Elapsed Time: 0.3

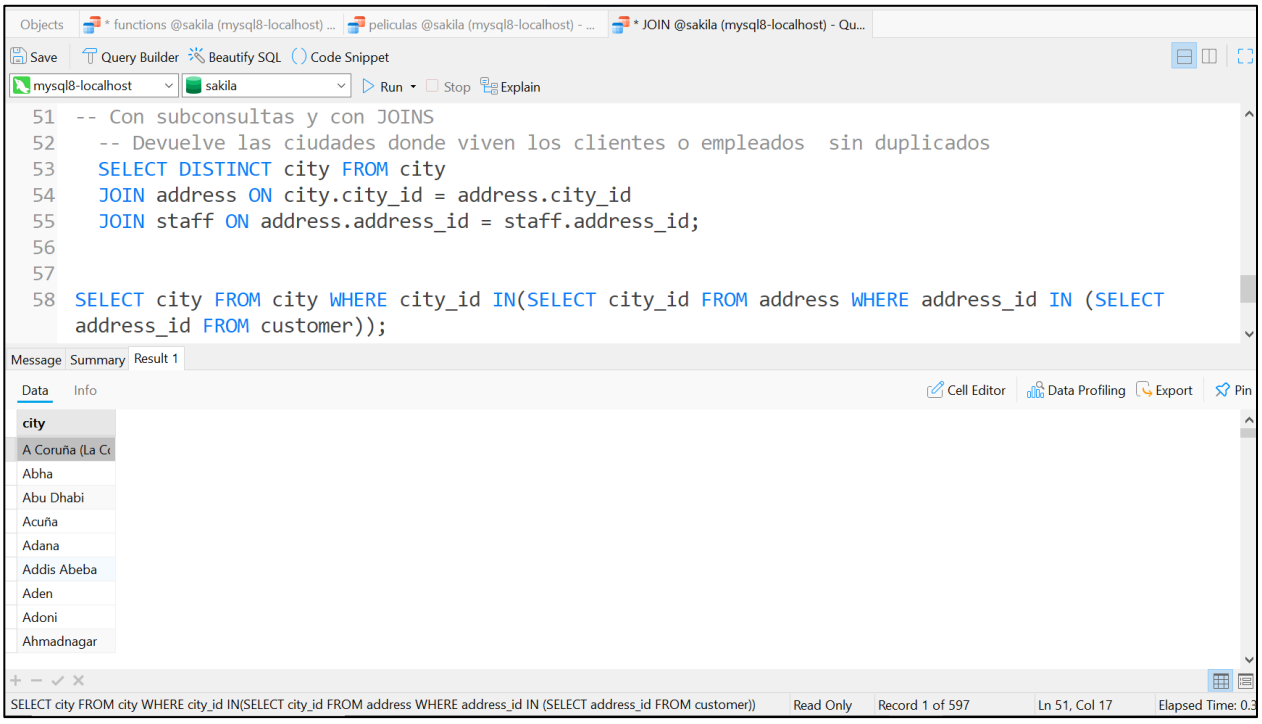

Objects
* functions @sakila (mysql8-localhost) ...
peliculas @sakila (mysql8-localhost) - ...
* JOIN @sakila (mysql8-localhost) - Qu...
Save
Query Builder
Beautify SQL
Code Snippet
mysql8-localhost
sakila
Run
Stop
Explain
25 -- Muestra los nombres de los actores y los títulos de las películas en las que han partipado
26 -- FIRSTNAME, LASTNAME, TITLE
27
28 SELECT a.first_name, a.last_name, f.title
29 FROM actor AS a
30 LEFT JOIN film_actor AS fa ON a.actor_id = fa.actor_id
31 LEFT JOIN film AS f ON fa.film_id = f.film_id;
32
33 SELECT CONCAT(a.first_name," ",a.last_name) AS full_name, f.title
34 FROM actor AS a
35 JOIN film_actor AS fa ON a.actor_id = fa.actor_id
36 JOIN film AS f ON fa.film_id = f.film_id;
Message
Summary
Result 1
Data
Info
Cell Editor
Data Profiling
Export
Pin
full_name
title
PENELOPE GUINESS ACADEMY DINOS
PENELOPE GUINESS ANACONDA CON
PENELOPE GUINESS ANGELS LIFE
PENELOPE GUINESS BULWORTH COM
PENELOPE GUINESS CHEAPER CLYDE
PENELOPE GUINESS COLOR PHILADEI
SELECT CONCAT(a.first_name," ",a.last_name) AS full_name, f.title FROM actor AS a JOIN film_actor AS fa ON a.actor_id = fa.actor_id JOI
Read Only
Record 1 of 5462
Ln 32, Col 1
Elapsed Time: 0.3

OPERACIONES SET EN MySQL









V. CONCLUSIONES:

Las consultas en MySQL son una parte esencial de la gestión y manipulación de bases de datos. Al analizar el uso de las consultas, se pueden extraer diversas conclusiones sobre su funcionamiento, eficiencia y buenas prácticas.

Las consultas en MySQL son fundamentales para interactuar con bases de datos, y su correcta optimización y uso adecuado son claves para lograr aplicaciones eficientes y escalables. El dominio de buenas prácticas como el uso de índices, la selección precisa de columnas, la optimización de JOINs, el manejo de transacciones y el monitoreo del rendimiento son esenciales para mantener el rendimiento y la integridad de los datos en sistemas de bases de datos MySQL.