

Nombre de la práctica	SET, JOINS Y SUBCONSULTAS			No.	6
Asignatura:	Taller de Base de Datos	Carrera:	ISIC	Duración de la práctica (Hrs)	4

Ana Edith Hernández Hernández

I. Realizar consultas con set, joins y subconsultas

II. Competencia(s) específica(s):

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

IV. Material empleado:

Computadora

V. Desarrollo de la práctica:

```

1  -- JOINS en MySQL
2  -- INNER JOIN : cuando coinciden ambas tablas
3  SELECT f.title, c.name AS category_name
4  FROM film AS f
5  INNER JOIN film_category AS fc
6  ON f.film_id = fc.film_id
7  INNER JOIN category AS c
8  ON fc.category_id = c.category_id
9  WHERE c.name = "Comedy";
10
11 -- Otro ejemplo de INNER JOIN

```

Message Summary Result 1

Data Info

title	category_name
AIRPLANE SIERRA	Comedy
ANTHEM LUKE	Comedy
BRINGING HYSTERICAL	Comedy
CAPER MOTIONS	Comedy
CAT CONEHEADS	Comedy
CLOSER BANG	Comedy
CONNECTION MICROCOSI	Comedy
CONTROL ANTHEM	Comedy
CRAZY HOME	Comedy
DADDY PITTSBURGH	Comedy
DOOM DANCING	Comedy
DOWNHILL ENOUGH	Comedy
DYING MAKER	Comedy



```
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
16
17 -- LEFT JOIN Toma los elementos que se encuentran a la izq
```

Message Summary Result 1

Data Info

Cell Editor Data Profiling Export Pin

title	name
AMADEUS HOLY	Action
AMERICAN CIRCUS	Action
ANTITRUST TOMATOES	Action
ARK RIDGEMONT	Action
BAREFOOT MANCHURIAN	Action
BERETS AGENT	Action
BRIDE INTRIGUE	Action
BULL SHAWSHANK	Action
CADDYSHACK JEDI	Action
CAMPUS REMEMBER	Action
CASUALTIES ENCINO	Action
CELEBRITY HORN	Action
CLUELESS BUCKET	Action

```
16
17 -- LEFT JOIN Toma los elementos que se encuentran a la izq
18 -- Encuentra el titulo de las películas 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
25
26 -- Muestra los nombres de los actores y los títulos de las películas en las que han participado
```

Message Summary Result 1

Data Info

Cell Editor Data Profiling Export Pin

title	cantidad_rentas
ACADEMY DINOSAUR	23
ACE GOLDFINGER	7
ADAPTATION HOLES	12
AFFAIR PREJUDICE	23
AFRICAN EGG	12
AGENT TRUMAN	21
AIRPLANE SIERRA	15
AIRPORT POLLOCK	18
ALABAMA DEVIL	12
ALADDIN CALENDAR	23
ALAMO VIDEOTAPE	24
ALASKA PHANTOM	26
ALI FOREVER	9



```
25
26 -- Muestra los nombres de los actores y los títulos de las películas en las que han participado
27 -- FIRSTNAME, LASTNAME, TITLE
28
29 SELECT a.first_name, a.last_name, f.title
30 FROM actor AS a
31 LEFT JOIN film_actor AS fa ON a.actor_id = fa.actor_id
32 LEFT JOIN film AS f ON fa.film_id = f.film_id;
33
34 SELECT CONCAT(a.first_name," ",a.last_name) AS full_name, f.title
35 FROM actor AS a
36 JOIN film_actor AS fa ON a.actor_id = fa.actor_id
37 JOIN film AS f ON fa.film_id = f.film_id;
38
```

Message Summary Result 1 Result 2

Data Info

Cell Editor Data Profiling Export Pi

first_name	last_name	title
PENELOPE	GUINNESS	ACADEMY DINOSAUR
PENELOPE	GUINNESS	ANACONDA CONFESSION
PENELOPE	GUINNESS	ANGELS LIFE
PENELOPE	GUINNESS	BULWORTH COMMANDM
PENELOPE	GUINNESS	CHEAPER CLYDE
PENELOPE	GUINNESS	COLOR PHILADELPHIA
PENELOPE	GUINNESS	ELEPHANT TROJAN
PENELOPE	GUINNESS	GLEAMING JAWBREAKER
PENELOPE	GUINNESS	HUMAN GRAFFITI
PENELOPE	GUINNESS	KING EVOLUTION

```
39
40 -- Operaciones SET
41 SELECT first_name FROM actor
42 UNION
43 SELECT first_name FROM customer;
44
```

Message Summary Result 1

Data Info

first_name

PENELOPE
NICK
ED
JENNIFER
JOHNNY
BETTE
GRACE
MATTHEW
JOE
CHRISTIAN



```

45 -- Encuentra las películas que no han sido alquiladas
46 SELECT title FROM film
47 EXCEPT
48 SELECT f.title FROM film AS f
49 JOIN inventory AS i ON f.film_id = i.film_id
50 JOIN rental AS r ON i.inventory_id = r.inventory_id;

```

Message	Summary	Result 1																								
		<table border="1"> <thead> <tr> <th>Data</th> <th>Info</th> </tr> </thead> <tbody> <tr> <td>title</td> <td></td> </tr> <tr> <td>ALICE FANTASIA</td> <td></td> </tr> <tr> <td>APOLLO TEEN</td> <td></td> </tr> <tr> <td>ARGONAUTS TOWN</td> <td></td> </tr> <tr> <td>ARK RIDGEMONT</td> <td></td> </tr> <tr> <td>ARSENIC INDEPENDENCE</td> <td></td> </tr> <tr> <td>BOONDOCK BALLROOM</td> <td></td> </tr> <tr> <td>BUTCH PANTHER</td> <td></td> </tr> <tr> <td>CATCH AMISTAD</td> <td></td> </tr> <tr> <td>CHINATOWN GLADIATOR</td> <td></td> </tr> <tr> <td>CHOCOLATE DUCK</td> <td></td> </tr> </tbody> </table>	Data	Info	title		ALICE FANTASIA		APOLLO TEEN		ARGONAUTS TOWN		ARK RIDGEMONT		ARSENIC INDEPENDENCE		BOONDOCK BALLROOM		BUTCH PANTHER		CATCH AMISTAD		CHINATOWN GLADIATOR		CHOCOLATE DUCK	
Data	Info																									
title																										
ALICE FANTASIA																										
APOLLO TEEN																										
ARGONAUTS TOWN																										
ARK RIDGEMONT																										
ARSENIC INDEPENDENCE																										
BOONDOCK BALLROOM																										
BUTCH PANTHER																										
CATCH AMISTAD																										
CHINATOWN GLADIATOR																										
CHOCOLATE DUCK																										

```

51
52 -- Con subconsultas y con JOINS
53 -- Devuelve las ciudades donde viven los clientes o empleados sin duplicados
54 SELECT DISTINCT city FROM city
55 JOIN address ON city.city_id = address.city_id
56 JOIN staff ON address.address_id = staff.address_id
57

```

Message	Summary	Result 1								
		<table border="1"> <thead> <tr> <th>Data</th> <th>Info</th> </tr> </thead> <tbody> <tr> <td>city</td> <td></td> </tr> <tr> <td>Lethbridge</td> <td></td> </tr> <tr> <td>Woodridge</td> <td></td> </tr> </tbody> </table>	Data	Info	city		Lethbridge		Woodridge	
Data	Info									
city										
Lethbridge										
Woodridge										



```
58
59 SELECT city FROM city WHERE city_id IN(SELECT city_id FROM address WHERE address_id IN (SELECT
60 address_id FROM customer));
61
62 INSERT actor VALUES (DEFAULT, "Lucio", "Hernandez", NOW());
```

Message Summary Result 1

Data Info

Cell Editor Data Profiling Export Pin

city
A Coruña (La C...
Abha
Abu Dhabi
Acuña
Adana
Addis Abeba
Aden
Adoni
Ahmadnagar
Akishima

```
61
62 INSERT actor VALUES (DEFAULT, "Lucio", "Hernandez", NOW());
63 CREATE VIEW prueba AS
64 SELECT first_name,last_name FROM actor
65 WHERE YEAR (last_update)>2020;
66
```

Message Summary

Processed Query: 2 Start Time: 2024-11-08 21:06:36
Success: 2 End Time: 2024-11-08 21:06:36
Error: 0 Elapsed Time: 0.064s ☐ Show Errors Only

Query	Message	Query Time	Fetch Time
INSERT actor VALUES (DEFAULT, "Lucio", "Hernandez", NOW())	Affected rows: 1	0.017s	0.000s
CREATE VIEW prueba AS	OK	0.025s	0.000s

```
71
72 SELECT AVG (LENGTH(first_name)) FROM actor;
```

Message Summary Result 1

Data Info

AVG (LENGTH(first_name))
5.3035



```
73  
74 SELECT * FROM actor WHERE LENGTH(first_name) > (SELECT AVG(LENGTH(first_name)) FROM actor);  
75
```

Message Summary Result 1

Data

Info

Cell Editor

Data P

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
24	CAMERON	STREEP	2006-02-15 04:34:33

```
75  
76 -- Encuentra los actores que han participado en películas de la categoría comedy  
77 -- Hechos por separado  
78 SELECT category_id FROM category WHERE name = "Comedy";  
79 SELECT film_id FROM film_category WHERE category_id = 5;  
80  
81 -- Uniendo ambas QUERY  
82 SELECT film_id FROM film_category WHERE category_id = (SELECT category_id FROM category WHERE name = "Comedy");
```

Message Summary Result 1 Result 2

Data

Info

Cell

category_id

5



```
80
81 -- Uniendo ambas QUERY
82 SELECT film_id FROM film_category WHERE category_id = (SELECT category_id FROM
83 category WHERE name = "Comedy");
```

Message Summary Result 1

Data Info [Cell Editor](#) [Data Profiling](#) [Export](#)

film_id
7
28
99
119
127
159
178
182
188
202
242
247

```
83
84 -- Para listas hacemos uso del IN y del NOT IN
85 SELECT actor_id FROM film_actor WHERE film_id IN (7,28,99);
86
```

Message Summary Result 1

Data Info [Cell Editor](#) [Data Profiling](#) [Export](#)

actor_id
99
133
162
170
185
74
162
48
144



```
87 -- Uniendo QUERY
88 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 P

actor_id

99
133
162
170
185
74
162
48
144
17
34

```
89
90 -- Haciendo el último SELECT
91 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 P

first_name last_name

JIM MOSTEL
RICHARD PENN
OPRAH KILMER
MENA HOPPER
MICHAEL BOLGER
MILLA KEITEL
FRANCES DAY-LEWIS
ANGELA WITHERSPOON
HELEN VOIGHT
AUDREY OLIVIER
VAL BOLGER

VI. Conclusiones:

SET, JOINS y SUBCONSULTAS son herramientas fundamentales en SQL que permiten manipular y consultar datos en bases de datos relacionales de forma eficiente. La instrucción `SET` se usa principalmente para asignar valores a variables y configurar opciones de sesión, lo que facilita la personalización y el control del entorno de ejecución. Los `JOINS`, por su parte, son esenciales para combinar datos de dos o más tablas relacionadas, permitiendo crear conexiones lógicas y extraer información coherente en función de las relaciones entre las tablas. Existen varios tipos de JOINS, como INNER JOIN, LEFT JOIN y RIGHT JOIN, cada uno adaptado a diferentes necesidades según se busquen coincidencias exactas o se desee incluir datos de una tabla aunque no tengan correspondencia en otra. Por último, las `SUBCONSULTAS` o consultas anidadas son consultas internas que se ejecutan dentro de otra consulta, permitiendo trabajar con datos intermedios o utilizar los resultados de una consulta en otra. Estas tres herramientas son cruciales para realizar consultas complejas, optimizar el manejo de grandes volúmenes de datos y extraer información significativa que permita tomar decisiones informadas.