

Resolución ejercicios 2

Autor: Juan V. Pino Contreras

```
/*Entra en la base de datos sakila*/
```

```
USE sakila;
```

#	Time	Action
1	19:53:02	USE sakila

```
4 /*Seleccionar los campos "address" y "district" de la tabla "address" donde distrito sea Texas. */
5 • SELECT address, district FROM address WHERE district = "Texas";
```

	address	district
▶	1795 Santiago de Compostela Way	Texas
	333 Goiânia Way	Texas
	913 Coacalco de Berriozábal Loop	Texas
	530 Lausanne Lane	Texas
	1894 Boa Vista Way	Texas

```
/*Seleccionar todas las columnas de la table "payment" cuyo campo "amount" sea mayor que 11.*/
```

```
SELECT * FROM payment WHERE amount > 11;
```

payment_id	customer_id	staff_id	rental_id	amount	payment_date	last_update
342	13	2	8831	11.99	2005-07-29 22:37:41	2006-02-15 22:12:31
3146	116	2	14763	11.99	2005-08-21 23:34:00	2006-02-15 22:13:08
5280	195	2	16040	11.99	2005-08-23 22:19:33	2006-02-15 22:14:00
5281	196	2	106	11.99	2005-05-25 18:18:19	2006-02-15 22:14:00
5550	204	2	15415	11.99	2005-08-22 23:48:56	2006-02-15 22:14:07
6409	237	2	11479	11.99	2005-08-02 22:18:13	2006-02-15 22:14:38
8272	305	1	2166	11.99	2005-06-17 23:51:21	2006-02-15 22:15:47
9803	362	1	14759	11.99	2005-08-21 23:28:58	2006-02-15 22:16:57
15821	591	2	4383	11.99	2005-07-07 20:45:51	2006-02-15 22:23:18
15850	592	1	3973	11.99	2005-07-06 22:58:31	2006-02-15 22:23:25
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
> /*Visualizar todos los actores de la tabla "actor" cuyo first_name sea "Penelope".
- Prueba con mayúscula y minúsculas para ver si devuelve el mismo resultado.*/
-- En ambas consultas devuelve el mismo resultado
```

	first_name
▶	PENELOPE
	PENELOPE
	PENELOPE
	PENELOPE

```
SELECT first_name FROM actor WHERE first_name = "penelope";
SELECT first_name FROM actor WHERE first_name = "PENELOPE";
```

	first_name
▶	PENELOPE
	PENELOPE
	PENELOPE
	PENELOPE

```
SELECT first_name FROM actor WHERE first_name = "penelope";
SELECT first_name FROM actor WHERE first_name = "PENELOPE";
```

```

/*Hacer lo mismo, pero con la cláusula BINARY para comprobar la diferencia. */
-- La consulta con "penelope" con minuscula no devuelve resultado.
-- Usando BINARY las letras mayúsculas y minúsculas las trata como distintas durante la comparación.

```

```

SELECT first_name FROM actor WHERE BINARY first_name = "PENELOPE";
SELECT first_name FROM actor WHERE BINARY first_name = "penelope";

```

first_name
PENELOPE
PENELOPE
PENELOPE
PENELOPE

```

SELECT first_name FROM actor WHERE BINARY first_name = "PENELOPE";
SELECT first_name FROM actor WHERE BINARY first_name = "penelope";

```

first_name

```

/*Visualizar las columnas first_name y last_name de la tabla "customer" de aquellos clientes que
esté inactivos (campo "active" igual a 0) .*/

```

```

SELECT first_name, last_name FROM customer WHERE active = 0;

```

first_name	last_name
SANDRA	MARTIN
JUDITH	COX
SHEILA	WELLS
ERICA	MATTHEWS
HEIDI	LARSON
PENNY	NEAL
KENNETH	GOODEN
HARRY	ARCE
NATHAN	RUNYON
THEODORE	CULP
MAURICE	CRAWLEY
BEN	EASTER
CHRISTIAN	JUNG
JIMMIE	EGGLESTON
TERRANCE	ROUSH

```
/*Prueba cambiando la igualdad del CERO por un False. ¿funciona? Si es así ¿Cuál es el motivo?*/
-- Si funciona. MySQL interpreta el valor BOOLEANO FALSE como 0 y TRUE como 1.
```

```
SELECT first_name, last_name FROM customer WHERE active = false;
```

first_name	last_name
SANDRA	MARTIN
JUDITH	COX
SHEILA	WELLS
ERICA	MATTHEWS
HEIDI	LARSON
PENNY	NEAL
KENNETH	GOODEN
HARRY	ARCE
NATHAN	RUNYON
THEODORE	CULP
MAURICE	CRAWLEY
BEN	EASTER
CHRISTIAN	JUNG
JIMMIE	EGGLESTON
TERRANCE	ROUSH

```

> /*¿Qué resultados daría el comando siguiente? ¿Los clientes activos o los
· inactivos : */
-- Muestra los clientes activos. Los valores 0 y 1 MySQL los interpreta como valores Boolean, por tanto en la consulta
-- 'WHERE active'; lo interpreta como active = true

```

```
SELECT first_name,last_name,active FROM customer WHERE active;
```

first_name	last_name	active
MARY	SMITH	1
PATRICIA	JOHNSON	1
LINDA	WILLIAMS	1
BARBARA	JONES	1
ELIZABETH	BROWN	1
JENNIFER	DAVIS	1
MARIA	MILLER	1
SUSAN	WILSON	1
MARGARET	MOORE	1

```
/*¿Y el siguiente?¿Activos o inactivos? */
```

```
-- Muestra los clientes inactivos. En MySQL con el simbolo ! evaluamos el valor falso o valor contrario del campo comparatorio
```

```
SELECT first_name,last_name,active from customer where !active;
```

first_name	last_name	active
SANDRA	MARTIN	0
JUDITH	COX	0
SHEILA	WELLS	0
ERICA	MATTHEWS	0
HEIDI	LARSON	0
PENNY	NEAL	0
KENNETH	GOODEN	0
HARRY	ARCE	0
NATHAN	RUNYON	0
THEODORE	CULP	0

```

/* Si ejecutamos esta query contra la tabla "rental":*/
-- Devuelve valores nulos. Esto porque el campo de comparacion rental_date se lo estamos pasando con un tipo de dato DATE,
-- y el campo rental_date de la tabla esta en DATETIME, es decir fecha y hora.
SELECT * FROM rental where rental_date='2006-02-14';
DESCRIBE rental;

```

rental_date	datetime	NO	MUL	NULL		
rental_id	rental_date	inventory_id	customer_id	return_date	staff_id	last_update
NULL	NULL	NULL	NULL	NULL	NULL	NULL

/*Verás que no aparece ninguna fila, a pesar de que si existen registros en esa fecha.
 ¿A qué es debido? ¿Cómo podríamos visualizar alguno de los registros?*/

/* Esto porque el campo de comparacion rental_date se lo estamos pasando con un tipo de dato DATE, y el campo rental_date de la tabla esta en DATETIME, es decir fecha y hora.*/

```
-- Con DATE(rental_date) extraemos el campo de tipo fecha del valor de rental_date.
```

- DESC rental;
- SELECT * FROM rental WHERE DATE(rental_date) = '2006-02-14';

11496	2006-02-14 15:16:03	2047	155	NULL	1	2006-02-15 21:30:53
11541	2006-02-14 15:16:03	2026	335	NULL	1	2006-02-15 21:30:53
11563	2006-02-14 15:16:03	1545	83	NULL	1	2006-02-15 21:30:53
11577	2006-02-14 15:16:03	4106	219	NULL	2	2006-02-15 21:30:53
11593	2006-02-14 15:16:03	817	99	NULL	1	2006-02-15 21:30:53
11611	2006-02-14 15:16:03	1857	192	NULL	2	2006-02-15 21:30:53
11646	2006-02-14 15:16:03	478	11	NULL	2	2006-02-15 21:30:53
11652	2006-02-14 15:16:03	1622	597	NULL	2	2006-02-15 21:30:53
11657	2006-02-14 15:16:03	3043	53	NULL	2	2006-02-15 21:30:53
11673	2006-02-14 15:16:03	2047	524	NULL	2	2006-02-15 21:30:53