**TAREA 2**

/\* PRACTICAS MYSQL \*/

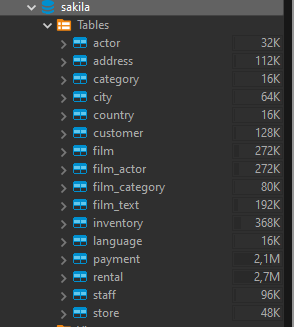
/\* Comandos WHERE, BINARY \*/

❖ Accedemos con mysql, mysql workbench o cualquier otra de las herramientas con la que estás trabajando.

❖ Descarga y carga la base de datos SAKILA, que es oficial de MYSQL: AQUÍ • Instala primero sakila-schema. • Instala segundo sakila-data

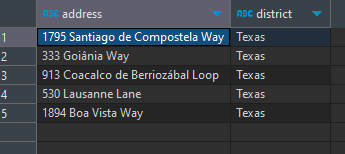
❖ Entra en la base de datos sakila”. Utiliza capturas de pantalla.

* Entrar en base de Datos SAKILA



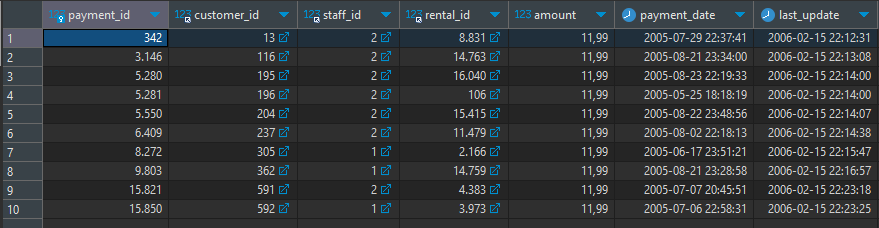
❖ Seleccionar los campos “address” y “district” de la tabla “address” donde distrito sea Texas.

**select** address, district **from** address *a* **where** district = **"Texas"**



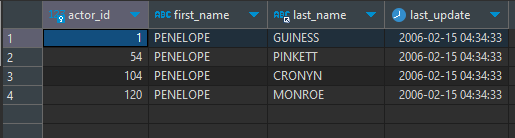
❖ Seleccionar todas las columnas de la table “payment” cuyo campo “amount” sea mayor que 11.

**select** \* **from** payment p **where** amount > 11;

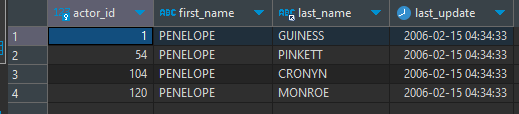


❖ Visualizar todos los actores de la tabla “actor” cuyo first\_name sea “Penelope”. Prueba con mayúscula y minúsculas para ver si devuelve el mismo resultado.

**select** \* **from** actor *a* **where** first\_name = **"Penelope"**

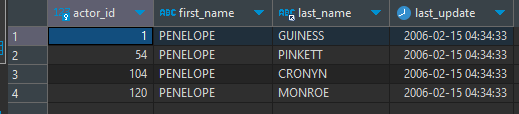


**select** \* **from** actor *a* **where** first\_name = **"penelope"**;



❖ Hacer lo mismo, pero con la cláusula BINARY para comprobar la diferencia.

**select** \* **from** actor *a* **where** **binary** first\_name = **"PENELOPE"**;



pero

**select** \* **from** actor *a* **where** **binary** first\_name = **"Penelope"**;

o

**select** \* **from** actor *a* **where** **binary** first\_name = **"penelope"**;

NO arrojan resultados pero si una advertencia de deprecated para usar el CAST.



❖ 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 *c* **where** active = 0;



❖ Prueba cambiando la igualdad del CERO por un False. ¿funciona? Si es así ¿Cuál es el motivo?

R: Si funciona y entrega el mismo resultado porque, si bien la columna es de tipo tinyint, al solo existir 0 y 1, el motor los interpreta como booleano (true o false).

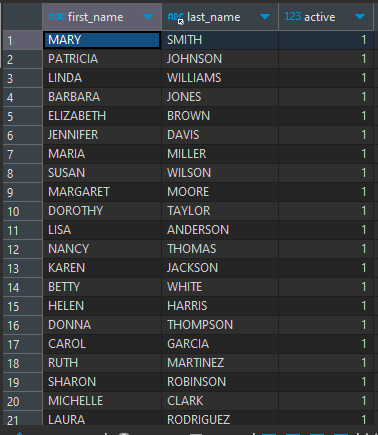
**select** first\_name, last\_name **from** customer *c* **where** active = **false**;



❖ ¿Qué resultados daría el comando siguiente? ¿Los clientes activos o los inactivos : • select first\_name,last\_name,active from customer where active;

R: La columna “active” sin condición, el motor MySQL lo interpreta como TRUE, o sea, un 1, y devuelve todos los customer que estén con ese valor, o sea, todos los que están activados.

**select** first\_name,last\_name,active **from** customer **where** active;

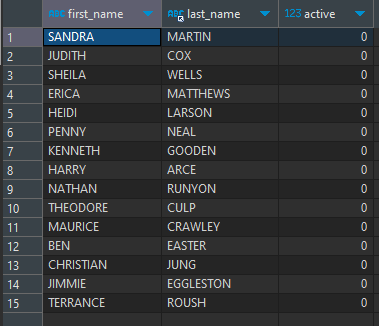


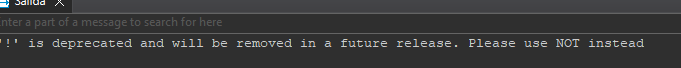
❖ ¿Y el siguiente?¿Activos o inactivos? • select first\_name,last\_name,active from customer where !active;

R: El Simbolo “!” está negando la condición de la columna “active”, o sea, está buscando los FALSE.

Nota: el símbolo “!” para consultas MySQL también tiene un aviso de deprecado, se recomienda usar el NOT.

**select** first\_name,last\_name,active **from** customer **where** !active

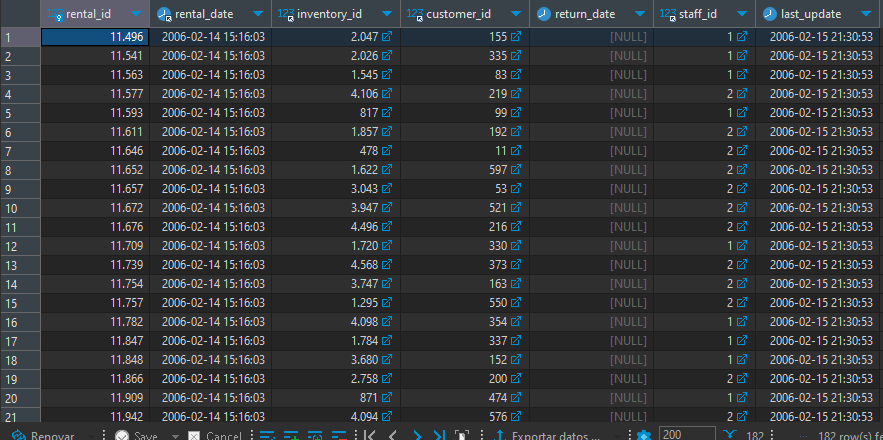




❖ Si ejecutamos esta query contra la tabla “rental”: select \* from rental where rental\_date="2006-02-14"; Verás que no aparece ninguna fila, a pesar de que si existen registros en esa fecha. ¿A qué es debido? ¿Cómo podríamos visualizar alguno de los registros?

R: El problema es que debemos informarle a MySQL que la columna a consultar es de tipo DATE.

**select** \* **from** rental **where** **date**(rental\_date) = **'2006-02-14'**;



❖ PISTA. Si hacemos un desc de la tabla Rental comprobamos que la columna rental\_date es de tipo DATETIME, no es de tipo DATE