

Consultas Base de Datos VideoClub

1. Muestra el nombre y apellido de todos los actores.

```
SELECT actor.nombre,actor.apellidos from actor;
```

2. Muestra el nombre y apellido de cada actor en una sola columna, en mayúscula. Nombra la columna "Nombre del actor".

```
SELECT UPPER(concat(actor.nombre," ",actor.apellidos))as 'Nombre del Actor' from actor;
```

3. Muestra el ID, nombre y apellido de un actor, de quien solo tienes el nombre "Joe".

```
SELECT actor.id_actor,actor.nombre,actor.apellidos from actor where actor.nombre='JOE';
```

4. Encuentra los actores cuyo apellido contenga "GEN".

```
SELECT actor.nombre , actor.apellidos FROM actor WHERE apellidos LIKE "%GEN%";
```

5. Encuentra los actores cuyo apellido contenga "LI". Ordena las filas por apellido y nombre (en ese orden).

```
SELECT nombre , apellidos FROM actor WHERE apellidos LIKE '%LI%' order by apellidos,nombre asc;
```

6. Usando la función IN, muestra el nombre y apellido de todos los clientes llamados "Terry", "Jessie" o "Alice".

```
SELECT cliente.nombre,cliente.apellidos FROM cliente WHERE nombre IN ('Terry', 'Jessie', 'Alice');
```

7. Muestra el apellido de cada actor y la cantidad de actores que tienen ese apellido.

```
SELECT actor.apellidos, COUNT(actor.apellidos) as 'Cantidad de actores' from actor GROUP by actor.apellidos;
```

8. Muestra el apellido y la cantidad de actores que tienen ese apellido, pero solo los apellidos compartidos por dos o más actores.

```
SELECT actor.apellidos, COUNT(actor.apellidos) as 'Cantidad de actores' from actor GROUP by actor.apellidos HAVING COUNT(actor.apellidos)>=2;
```

9. Usando *joins*, muestra el nombre, apellido y dirección de cada miembro del *staff*.

```
SELECT empleado.nombre,empleado.apellidos,direccion.direccion,direccion.direccion2,direccion.distrito,direccion.codigo_postal from empleado join direccion on direccion.id_direccion=empleado.id_direccion;
```

10. Muestra el total de dinero recaudado por cada empleado durante agosto del 2005.

```
SELECT SUM(pago.total) as 'Total  
Vendido' from pago WHERE (pago.fecha_pago BETWEEN '2005-08-  
01' and '2005-08-31') GROUP by pago.id_empleado;
```

```
SELECT empleado.id_empleado, empleado.nombre, SUM(pago.total) as 'Total  
Recaudado' from empleado join pago on empleado.id_empleado=pago.id_emp  
leado where pago.fecha_pago>='2005-08-01' and pago.fecha_pago<'2005-  
09-01' GROUP by empleado.id_empleado;
```

11. Despliega la cantidad de actores por cada película.

```
SELECT pelicula.titulo, COUNT(actor.id_actor) as 'Nº  
Actores' from pelicula, actor, pelicula_actor WHERE pelicula.id_pelicula  
=pelicula_actor.id_actor AND pelicula_actor.id_actor=actor.id_actor GR  
OUP BY pelicula.titulo;
```

12. ¿Cuántas copias hay inventariadas en el sistema de la película "Hunchback Impossible"?

```
SELECT pelicula.titulo, count(inventario.id_pelicula) as 'Nº  
Copias' from pelicula, inventario WHERE pelicula.id_pelicula=inventario  
.id_pelicula GROUP by pelicula.titulo;
```

13. Muestra el total de dinero pagado por cada cliente, solo si ha realizado compras. Ordena los clientes por apellido de forma ascendente.

```
SELECT cliente.nombre, cliente.apellidos, sum(pago.total) as 'Total' fro  
m cliente, pago where cliente.id_cliente=pago.id_cliente GROUP BY clien  
te.nombre ORDER BY cliente.apellidos ASC;
```

14. Se debe realizar una campaña de marketing en Canada. Para esto necesitas el nombre y correo electrónico de todos los clientes canadienses. Despliega esta información.

```
SELECT cliente.nombre, cliente.apellidos, cliente.email from cliente, pai  
s, direccion, ciudad WHERE pais.nombre='Canada' and cliente.id_direccion  
=direccion.id_direccion AND direccion.id_ciudad=ciudad.id_ciudad AND c  
iudad.id_pais=pais.id_pais;
```

15. Identifica todas las películas categorizadas como familiares (categoría "family").

```
SELECT pelicula.titulo from pelicula, pelicula_categoria, categoria WHER  
E pelicula.id_pelicula=pelicula_categoria.id_pelicula AND categoria.id  
_categoria=pelicula_categoria.id_categoria AND categoria.nombre='famil  
y';
```

16. Muestra las películas más arrendadas en orden descendente.

```
SELECT pelicula.titulo, COUNT(alquiler.id_alquiler) as 'Nº  
Alquileres' from cliente, pago, alquiler, inventario, pelicula WHERE clien
```

```
te.id_cliente=pago.id_pago AND pago.id_alquiler=alquiler.id_alquiler A  
ND alquiler.id_inventario=inventario.id_inventario AND inventario.id_p  
elicula=pelicula.id_pelicula GROUP BY pelicula.id_pelicula order by CO  
UNT(alquiler.id_alquiler) DESC;
```

17. Despliega el dinero recaudado por cada almacén.

```
SELECT almacen.id_almacen, sum(pago.total) from pago, alquiler, inventari  
o, almacen WHERE pago.id_alquiler=alquiler.id_alquiler AND alquiler.id_  
inventario=inventario.id_inventario AND inventario.id_almacen=almacen.  
id_almacen GROUP BY almacen.id_almacen;
```