**EJERCICIOS VISTAS 2**

**Realiza sobre el esquema videoclub:**

1. Crea una vista llamada **preciofinal** en la que guardarás el título de la película y el precio final de alquiler resultante de restar la oferta al precio de alquiler

CREATE VIEW preciofinal (titulo,preciofinal) as select titulo, truncate(precioalquiler - precioalquiler\*oferta,2) from peliculas;

1. Crea una vista llamada **multas** en la que mostrarás los nombres de los socios del video club a los que se ha multado por entregar la película tarde y el total que tienen acumulado en multas.

CREATE VIEW multas as select concat(socios.nombre," ",socios.apellidos) as NOMBRE\_SOCIO, sum((datediff(alquiler.fechadevolucion, alquiler.fechasalida) - peliculas.MAXIMODIAS) \* 2) as MULTA from alquiler,peliculas,stock,socios where alquiler.CODIGOSOCIO = socios.CODSOC and alquiler.CODIGOSTOCK = stock.CODSTOCK and peliculas.CODPEL = stock.CODPELIC and alquiler.FECHADEVOLUCION is not null and (datediff(alquiler.fechadevolucion, alquiler.fechasalida) - peliculas.MAXIMODIAS) > 0 group by nombre\_socio;

1. Crea una vista llamada **categ** en la que mostrarás los nombres de las categorías y el total gastado en alquiler de películas en dichas categorías después de restar la oferta correspondiente.

CREATE VIEW categ as select categorias.DESCRIPCIÓN, truncate(sum(peliculas.PRECIOALQUILER - peliculas.PRECIOALQUILER \* oferta),2) as TOTAL from peliculas, stock, alquiler, categorias where peliculas.CODCATEG = categorias.CODCAT and peliculas.CODPEL = stock.CODPELIC and alquiler.CODIGOSTOCK = stock.CODSTOCK group by categorias.CODCAT;

1. Crea una vista llamada **mas8** en la que mostrarás nombre y apellidos de los socios en un único campo llamado NCOMPLETO y el total gastado en alquiler por los mismos después de restar la oferta, pero sólo de aquellos socios que hayan gastado más de 8 euros en dicho alquiler. Ordena descendentemente por el total.

CREATE VIEW mas8 as select concat(socios.nombre," ",socios.apellidos) as NCOMPLETO, truncate(sum(peliculas.PRECIOALQUILER - peliculas.PRECIOALQUILER \* oferta),2) as TOTAL from peliculas, alquiler, stock, socios where peliculas.CODPEL = stock.CODPELIC and alquiler.CODIGOSTOCK = stock.CODSTOCK and alquiler.CODIGOSOCIO = socios.CODSOC group by ncompleto having total > 8 order by total desc;

1. Crea una vista llamada **meses** en la que mostrarás el nombre de los meses y el total recaudado por el alquiler en cada uno, después de descontado el campo oferta. Ordena descendentemente por el total.

CREATE VIEW meses as select monthname(alquiler.fechasalida) as mes, truncate(sum(peliculas.PRECIOALQUILER - peliculas.PRECIOALQUILER \* oferta),2) as TOTAL from alquiler, peliculas, stock where peliculas.CODPEL = stock.CODPELIC and alquiler.CODIGOSTOCK = stock.CODSTOCK group by mes;

**Realiza sobre el esquema otra biblioteca:**

1. Crea una vista llamada **sanciones** en la que mostrarás nombre y apellidos de los socios en un solo campo llamado NOMBRECOMPLETO y el total de euros que tendrán que pagar como consecuencia de la devolución de los libros más tarde de lo debido. Sabemos que por cada día de retraso se paga 1.50 euros. Muestra sólo aquellos socios que se hayan pasado de días.

CREATE VIEW sanciones as select concat(usuarios.nombre, " ", usuarios.apellidos) as NOMBRECOMPLETO, sum((datediff(prestamos.fdevol, prestamos.fmax)) \* 1.5) as TOTAL from usuarios, prestamos where usuarios.Codusuario = prestamos.Codusuario and (datediff(prestamos.fdevol, prestamos.fmax)) \* 1.5 > 0 group by nombrecompleto;

1. Crea una vista llamada **librosgenero** en la que guardarás el nombre del género y el número de libros que hay de cada uno de ellos. Prueba después a añadir un libro en la tabla libros y comprueba que se ha sumado uno más en el género al que haga referencia dicho libro.

CREATE VIEW librosgenero as select genero.nombre, count(\*) from libros, genero where libros.Genero = genero.idgenero group by genero.idgenero;

INSERT INTO libros values (999, "Iván García Prieto", "San Andrés", "Yo", 5, "España", 100, 2020, 50);

El libro se ha sumado a la vista.

1. Crea una vista llamada **preciogenero** en la que guardarás el nombre del género y total gastado en los libros que pertenecen al mismo.

CREATE VIEW PRECIOGENERO as select genero.nombre, sum(libros.Preciolibro) from libros, genero where libros.Genero = genero.idgenero group by genero.idgenero;

**Realiza sobre el esquema supermercado:**

1. Crea una vista llamada **precioiva** en la que mostrarás el total en euros vendido por cada planta del supermercado, después de aplicar el 21% de IVA a las ventas. Ordena descendentemente por el total.

CREATE VIEW precioiva as select departamentos.Planta ,truncate(sum(articulos.PRECIOUNIDAD\*ventas.Unidvend)\*1.21,2) as Total from departamentos, articulos, ventas, vendedores where vendedores.DEPARTAMENTO = departamentos.DEPART and vendedores.CODVEND = ventas.Codvend and articulos.CODART = ventas.Codartic group by departamentos.DEPART order by total desc;

1. Crea una vista llamada **totalmes** en la que mostrarás el total en euros vendido por cada mes, después de aplicar el 21% de IVA a las ventas, pero sólo de los meses que hayan vendido más de 50 euros.

CREATE VIEW totalmes as select monthname(ventas.fechaventa) as Mes,truncate(sum(articulos.PRECIOUNIDAD\*ventas.Unidvend)\*1.21,2) as Total from departamentos, articulos, ventas, vendedores where vendedores.DEPARTAMENTO = departamentos.DEPART and vendedores.CODVEND = ventas.Codvend and articulos.CODART = ventas.Codartic group by mes having total > 50;