**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 AS**

**SELECT peliculas.TITULO, PRECIOALQUILER-precioalquiler\*oferta AS TOTAL**

**FROM peliculas**

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

**CREATE VIEW multas AS**

**SELECT concat\_ws(" ",nombre, socios.apellidos) as nombrecompleto, sum((DATEDIFF(FECHADEVOLUCION,FECHASALIDA)-MAXIMODIAS)\*2) AS MULTA**

**FROM ((PELICULAS INNER JOIN STOCK ON PELICULAS.CODPEL=STOCK.CODPELIC)**

**INNER JOIN ALQUILER ON ALQUILER.CODIGOSTOCK=STOCK.CODSTOCK) INNER JOIN SOCIOS ON SOCIOS.CODSOC=ALQUILER.CODIGOSOCIO**

**WHERE FECHADEVOLUCION IS NOT NULL and DATEDIFF(FECHADEVOLUCION,FECHASALIDA)-MAXIMODIAS>0**

**group by nombrecompleto**

**TENED EN CUENTA QUE LO QUE MOSTRAMOS EN EL SELECT ES EL IMPORTE DE LA MULTA Y LO QUE QUEREMOS COMPROBAR ES QUE LA PELÍCULA HA ESTADO MÁS TIEMPO DE LA CUENTA FUERA DEL VIDEOCLUB. POR ELLO ESTA CONDICIÓN VAMOS A PONERLA EN EL WHERE. NO HACE FALTA HAVING, PUESTO QUE TODOS LOS CAMPOS QUE UTILIZAMOS SON CAMPOS DE LA TABLA.**

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.DESCRIPCION, round(SUM(PRECIOALQUILER-precioalquiler\*oferta),2) AS TOTAL**

**FROM (((PELICULAS INNER JOIN STOCK ON PELICULAS.CODPEL=STOCK.CODPELIC) INNER JOIN ALQUILER ON ALQUILER.CODIGOSTOCK=STOCK.CODSTOCK) INNER JOIN categorias on categorias.CODCAT=peliculas.CODCATEG)**

**group by categorias.DESCRIPCION**

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\_ws(" ",SOCIOS.NOMBRE,SOCIOS.APELLIDOS) AS NCOMPLETO, SUM(PRECIOALQUILER-precioalquiler\*oferta) AS TOTAL**

**FROM PELICULAS, STOCK, ALQUILER, SOCIOS**

**WHERE PELICULAS.CODPEL=STOCK.CODPELIC AND ALQUILER.CODIGOSTOCK=STOCK.CODSTOCK AND SOCIOS.CODSOC=ALQUILER.CODIGOSOCIO**

**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(fechasalida) as MESSALIDA, round(SUM(PRECIOALQUILER-precioalquiler\*oferta),2) AS TOTAL**

**FROM (((PELICULAS INNER JOIN STOCK ON PELICULAS.CODPEL=STOCK.CODPELIC) INNER JOIN ALQUILER ON ALQUILER.CODIGOSTOCK=STOCK.CODSTOCK)**

**group by MESSALIDA**

**order by total desc**

**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\_WS(" ", usuarios.Nombre, usuarios.Apellidos) AS NOMBRECOMPLETO, SUM(DATEDIFF(Fdevol, Fmax)\*1.5) AS EurosSANCION  
FROM usuarios INNER JOIN prestamos ON usuarios.Codusuario=prestamos.Codusuario  
WHERE DATEDIFF(Fdevol, Fmax)>0  
GROUP BY usuarios.Codusuario;**

**PARA SABER EL NÚMERO DE DÍAS QUE SE HA TENIDO EL LIBRO HAY QUE RESTAR LA FECHA DE DEVOLUCIÓN Y LA FECHA MÁXIMA HASTA LA QUE SE PUEDEN TENER. SI SALE MAYOR DE 0, ES QUE EL LIBRO HA ESTADO FUERA MÁS TIEMPO DE LA CUENTA, Y HAY QUE PONER MULTA. TAMPOCO HAY QUE USAR HAVING, PUESTO QUE TANTO FDEVOL COMO FMAX SON CAMPOS DE LA TABLA.**

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(\*) as numlibros**

**FROM genero**

**inner join libros on libros.Genero=genero.idgenero**

**group by genero.nombre**

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(preciolibro) as total**

**FROM genero**

**inner join libros on libros.Genero=genero.idgenero**

**group by genero.nombre**

**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 planta, sum(ventas.Unidvend\*articulos.PRECIOUNIDAD\*1.21) as total FROM ventas inner join articulos on articulos.codart=ventas.codartic inner join departamentos**

**on departamentos.DEPART=articulos.DPTO**

**group by planta;**

**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(fechaventa) as mes, sum(ventas.Unidvend\*articulos.PRECIOUNIDAD\*1.21) as total FROM ventas inner join articulos on articulos.codart=ventas.codartic**

**group by monthname(fechaventa)**

**having total>50;**