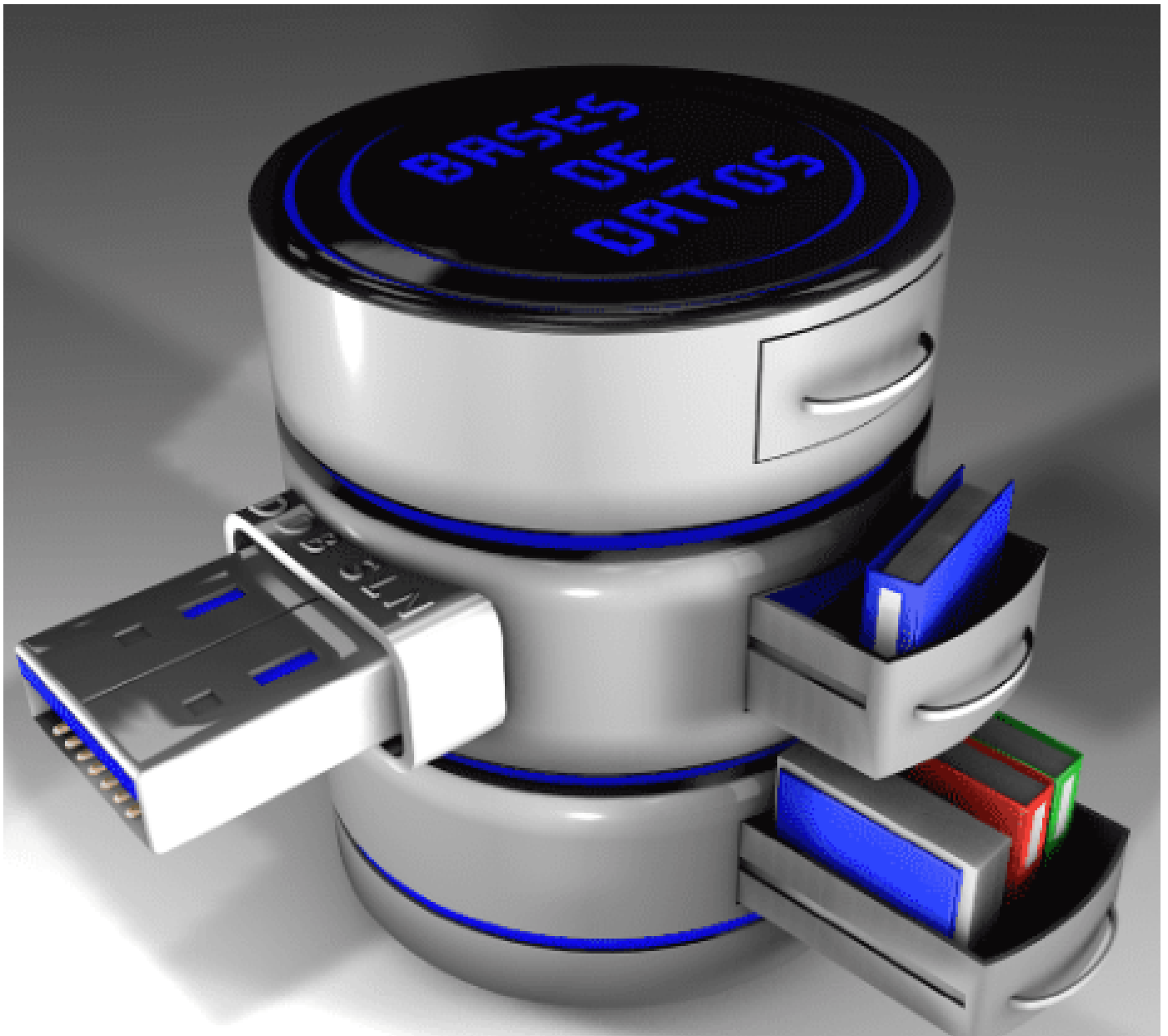


PARCIAL 1 FEBRERO



ÍNDICE

Introducción.....	3
Codigo ejercicio 1.....	4
Segundo Ejercicio.....	7
CONCLUSIÓN.....	9

Introducción

Una página de streaming de películas y series pretende almacenar, en una base de datos llamada "icseflix", información sobre los usuarios y los títulos que consumen.

De los usuarios se almacenan los siguientes datos:

Nombre.

Apellidos.

Correo electrónico.

Teléfono.

Tarjeta de crédito (24 dígitos). Ejemplo: ES00 1111 2222 3333 4444 5555

Al registrarse, los usuarios tiene que marcar sus preferencias (género), la cual, como mínimo, debe ser una (este apartado es para nota, no es necesario resolverlo, también puede explicarse y comentar cómo lo resolverían).

De los títulos (series o películas) se almacenan los siguientes datos:

Nombre del título.

Año de publicación.

Sinopsis.

Rango de edad, que son: Todos los públicos, 13 años, 16 años y 18 años).

Una vez que un usuario ve una serie (capítulo) o película queda registrado que la ha visto y, además, puede hacer una valoración sobre la misma.

De las series, mantenemos la misma información que de las películas (id y nombre), pero también tenemos que saber el capítulo que se está viendo y la temporada. No es necesario aplicar normalización entre películas y series.

Por último, la web tiene un componente social que permite que los usuarios tengan seguidores, de forma que se conocer las películas que ha visto un determinado usuario. No es necesario normalizar usuarios y seguidores.

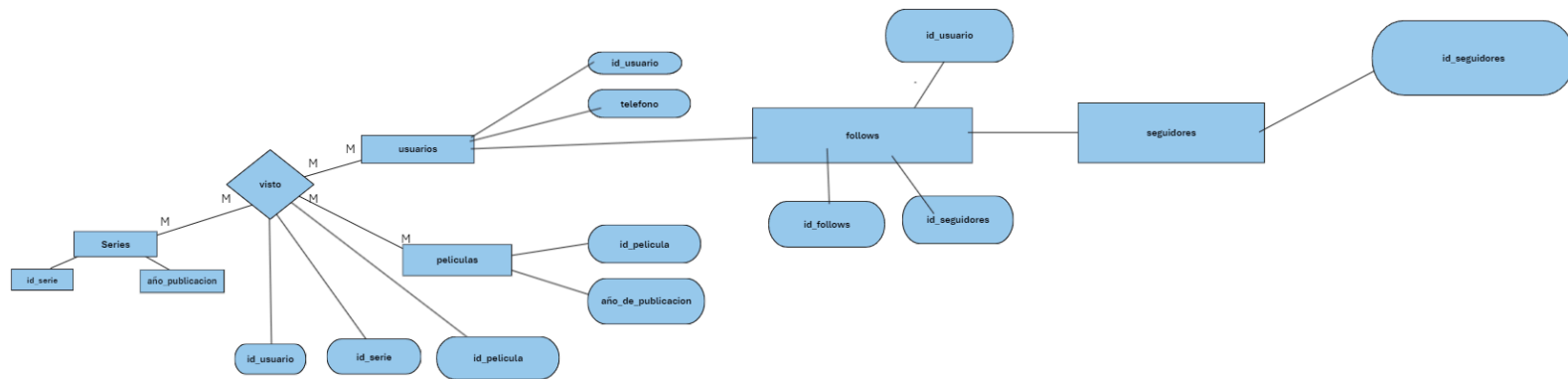
Crear el correspondiente modelo E-R:

Representación de las entidades y relaciones, cardinalidades y atributos (3 puntos).

Implementación en MySQL o similar del modelo E-R (3 puntos).

Codigo ejercicio 1

```
Create Database icseflix;
USE DATABASE icseflix;
CREATE TABLE usuarios (
    id_usuario INT (100) Primary KEY auto_increment,
    nombre varchar (100) not null,
    apellidos varchar(100) not null,
    correoelectronico varchar(100) not null,
    telefono INT (100) not null,
    TarjetadeCredito varchar(100) not null
);
CREATE TABLE pelicula (
    id_pelicula INT (100) Primary KEY auto_increment,
    nombre varchar (100) not null,
    año_de_publicacion date not null,
    sinopsis varchar (100) not null,
    rango_de_edad ENUM("Todos_los_públicos","13años","16años","18años") not null
);
CREATE TABLE series (
    id_serie INT (100) Primary KEY auto_increment,
    nombre varchar (100) not null,
    año_de_publicacion date not null,
    sinopsis varchar (100) not null,
    rango_de_edad ENUM("Todos_los_públicos","13años","16años","18años") not null
);
CREATE TABLE vistos (
    id_usuario INT (100),
    id_serie INT(100) ,
    id_pelicula INT(100),
    temporada INT(100),
    capitulo INT(100),
    fecha_visto DATE,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
    FOREIGN KEY (id_serie) REFERENCES series(id_serie),
    FOREIGN KEY (id_pelicula) REFERENCES pelicula(id_pelicula),
    visto boolean not null
);
CREATE TABLE seguidores(
    id_seguidores INT (100) Primary Key auto_increment
);
CREATE TABLE follows(
    id_seguidores INT (100),
    id_usuario INT (100),
    id_follows INT (100),
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
);
```



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0009 segundos.)

```
Create Database icseflix;
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0106 segundos.)

```
CREATE TABLE usuarios ( id_usuario INT (100) Primary KEY auto_increment, nombre varchar (100)
not null, apellidos varchar(100) not null, correoelectronico varchar(100) not null, telefono INT
(100) not null, TarjetadeCredito varchar(100) not null );
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0064 segundos.)

```
CREATE TABLE pelicula ( id_pelicula INT (100) Primary KEY auto_increment, nombre varchar (100)
not null, año_de_publicacion date not null, sinopsis varchar (100) not null, rango_de_edad
ENUM("Todos_los_públicos", "13años", "16años", "18años") not null );
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0068 segundos.)

```
CREATE TABLE series ( id_serie INT (100) Primary KEY auto_increment, nombre varchar (100) not
null, año_de_publicacion date not null, sinopsis varchar (100) not null, rango_de_edad
ENUM("Todos_los_públicos", "13años", "16años", "18años") not null );
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0190 segundos.)

```
CREATE TABLE vistos ( id_usuario INT (100), id_serie INT(100) , id_pelicula INT(100), temporada
INT(100), capitulo INT(100), fecha_visto DATE, FOREIGN KEY (id_usuario) REFERENCES
usuarios(id_usuario), FOREIGN KEY (id_serie) REFERENCES series(id_serie), FOREIGN KEY
(id_pelicula) REFERENCES pelicula(id_pelicula), visto boolean not null );
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0087 segundos.)

```
CREATE TABLE seguidores( id_seguidores INT (100) Primary Key auto_increment );
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0189 segundos.)

```
CREATE TABLE follows( id_seguidores INT (100), id_usuario INT (100), id_follows INT (100),  
FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario) );
```

La explicación de mi ejercicio lo que hago es hacer la base de datos, después hice 4 tablas que entre ellas se comunican por vistos debido a que 1 usuario puede ver muchas películas y series, entonces todas se enlazan por vistos, que en ella pongo las foreign key para poder entrelazarlas y uso en la tabla vistos pongo visto con un boolean, que eso se usa para que solo pueda ser o verdadero o falso y encima le pongo un not null para que sea obligatorio.

Segundo Ejercicio

Consulta para obtener la información detallada de todos los pedidos realizados por un cliente específico, incluyendo los nombres de los componentes de cada pedido.

✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0004 segundos.)

```
SELECT Pedidos.PedidoID, Pedidos.FechaPedido, Pedidos.EstadoPedido, Componentes.Nombre AS NombreComponente,
PedidoComponente.Cantidad FROM Pedidos JOIN PedidoComponente ON Pedidos.PedidoID = PedidoComponente.PedidoID JOIN
Componentes ON PedidoComponente.ComponenteID = Componentes.ComponenteID JOIN Clientes ON Pedidos.ClienteID =
Clientes.ClienteID WHERE Clientes.Nombre IN ('Juan', 'Ana') AND Clientes.Apellido IN ('Pérez', 'García');
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Ordenar según la clave: Ninguna

Opciones extra

PedidoID	FechaPedido	EstadoPedido	NombreComponente	Cantidad
1	2024-02-27	En Proceso	Procesador Intel i7	1
1	2024-02-27	En Proceso	Tarjeta Gráfica Nvidia GTX 1080	1
1	2024-02-27	En Proceso	Disco Duro SSD 1TB	1

Consulta para listar todos los componentes que están actualmente en stock pero que no han sido incluidos en ningún pedido reciente.

✓ 1 fila insertada.
La Id de la fila insertada es: 4 (La consulta tardó 0,0014 segundos.)

```
INSERT INTO Componentes (Nombre, Descripcion, Precio, Stock) VALUES ('Tarjeta de video', 'NVIDIA GeForce RTX 3080, 10GB GDDR6X', 799.99, 0);
```

✓ 1 fila insertada.
La Id de la fila insertada es: 5 (La consulta tardó 0,0011 segundos.)

```
INSERT INTO Componentes (Nombre, Descripcion, Precio, Stock) VALUES ('Memoria RAM', 'Corsair Vengeance RGB Pro, 16GB (2 x 8GB), DDR4 3200MHz', 129.99, 0);
```

✓ Mostrando filas 0 - 4 (total de 5, La consulta tardó 0,0001 segundos.)

```
SELECT ComponenteID, Nombre, Descripcion, Precio, Stock FROM Componentes;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Ordenar según la clave: Ninguna

Opciones extra

	ComponenteID	Nombre	Descripcion	Precio	Stock
<input type="checkbox"/> Editar Copiar Borrar	1	Procesador Intel i7	Procesador de alta velocidad	300.00	10
<input type="checkbox"/> Editar Copiar Borrar	2	Tarjeta Gráfica Nvidia GTX 1080	Tarjeta gráfica para juegos	700.00	5
<input type="checkbox"/> Editar Copiar Borrar	3	Disco Duro SSD 1TB	Disco duro sólido de alta capacidad	150.00	20
<input type="checkbox"/> Editar Copiar Borrar	4	Tarjeta de video	NVIDIA GeForce RTX 3080, 10GB GDDR6X	799.99	0
<input type="checkbox"/> Editar Copiar Borrar	5	Memoria RAM	Corsair Vengeance RGB Pro, 16GB (2 x 8GB), DDR4 32...	129.99	0

Consulta para identificar clientes que han realizado pedidos que incluyen un componente específico, por ejemplo, todos los clientes que han pedido un ordenador con una tarjeta gráfica particular.

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0005 segundos.)

```
SELECT DISTINCT Clientes.ClienteID, Clientes.Nombre, Clientes.Apellido FROM Clientes JOIN Pedidos ON Clientes.ClienteID = Pedidos.ClienteID JOIN Ordenadores ON Pedidos.PedidoID = Ordenadores.PedidoID JOIN
PedidoComponente ON Ordenadores.PedidoID = PedidoComponente.PedidoID JOIN Componentes ON PedidoComponente.ComponenteID = Componentes.ComponenteID WHERE Componentes.Nombre = 'Tarjeta Gráfica Nvidia GTX 1080';
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

ClienteID	Nombre	Apellido
1	Juan	Pérez

Consulta para obtener un listado de todos los ordenadores montados que están listos para entrega, mostrando el detalle de cada componente incluido en el montaje junto con la información del pedido correspondiente.

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0005 segundos.)

SELECT Ordenadores.OrdenadorID, Pedidos.PedidoID, Pedidos.FechaPedido, Pedidos.EstadoPedido, Componentes.ComponenteID, Componentes.Nombre AS NombreComponente, Componentes.Descripcion, Componentes.Precio FROM Ordenadores JOIN Pedidos ON Ordenadores.PedidoID = Pedidos.PedidoID JOIN PedidoComponente ON Pedidos.PedidoID = PedidoComponente.PedidoID JOIN Componentes ON PedidoComponente.ComponenteID = Componentes.ComponenteID WHERE Pedidos.EstadoPedido = 'En Proceso';

Perfilando [Editar en línea][Editar][Explicar SQL][Crear código PHP][Actualizar]

Mostrar todo

Número de filas: 25

Filtrar filas: Buscar en esta tabla

Ordenar según la clave: Ninguna

Opciones extra

OrdenadorID	PedidoID	FechaPedido	EstadoPedido	ComponenteID	NombreComponente	Descripcion	Precio
1	1	2024-02-27	En Proceso	1	Procesador Intel i7	Procesador de alta velocidad	300.00
1	1	2024-02-27	En Proceso	2	Tarjeta Gráfica Nvidia GTX 1080	Tarjeta gráfica para juegos	700.00
1	1	2024-02-27	En Proceso	3	Disco Duro SSD 1TB	Disco duro sólido de alta capacidad	150.00

Mostrar todo

Número de filas: 25

Filtrar filas: Buscar en esta tabla

Ordenar según la clave: Ninguna

```
SELECT Pedidos.PedidoID, Pedidos.FechaPedido, Pedidos.EstadoPedido, Componentes.Nombre AS NombreComponente, PedidoComponente.CantidadFROM Pedidos
JOIN PedidoComponente ON Pedidos.PedidoID = PedidoComponente.PedidoID
JOIN Componentes ON PedidoComponente.ComponenteID = Componentes.ComponenteID
JOIN Clientes ON Pedidos.ClienteID = Clientes.ClienteID
WHERE Clientes.Nombre IN ('Juan', 'Ana') AND Clientes.Apellido IN ('Pérez', 'García');
```

```
SELECT ComponenteID, Nombre, Descripcion, Precio, Stock
FROM Componentes;
```

```
SELECT DISTINCT Clientes.ClienteID, Clientes.Nombre, Clientes.Apellido
FROM Clientes
JOIN Pedidos ON Clientes.ClienteID = Pedidos.ClienteID
JOIN Ordenadores ON Pedidos.PedidoID = Ordenadores.PedidoID
JOIN PedidoComponente ON Ordenadores.PedidoID = PedidoComponente.PedidoID
JOIN Componentes ON PedidoComponente.ComponenteID = Componentes.ComponenteID
WHERE Componentes.Nombre = 'Tarjeta Gráfica Nvidia GTX 1080';
```

```
SELECT Ordenadores.OrdenadorID, Pedidos.PedidoID, Pedidos.FechaPedido, Pedidos.EstadoPedido,
Componentes.ComponenteID, Componentes.Nombre AS NombreComponente, Componentes.Descripcion, Componentes.Precio
FROM Ordenadores
JOIN Pedidos ON Ordenadores.PedidoID = Pedidos.PedidoID
JOIN PedidoComponente ON Pedidos.PedidoID = PedidoComponente.PedidoID
JOIN Componentes ON PedidoComponente.ComponenteID = Componentes.ComponenteID
WHERE Pedidos.EstadoPedido = 'En Proceso';
```


OpenAI(2024).Enum para tipos de películas.ChatGPT.<https://chat.openai.com/share/d44b8e02-9fc6-4622-9c40-f0c9c5a20f7d>

CONCLUSIÓN

Este examen para mi me ha parecido, que ha sido un poco confuso debido a que el enunciado está un enrevesado y lo de dibujar (Entidad Relacion) habría que explicar como se hace estilo que no me he enterado bien de como hacer tal cosa, que si una cosa en rombo, cuadrado, etc...