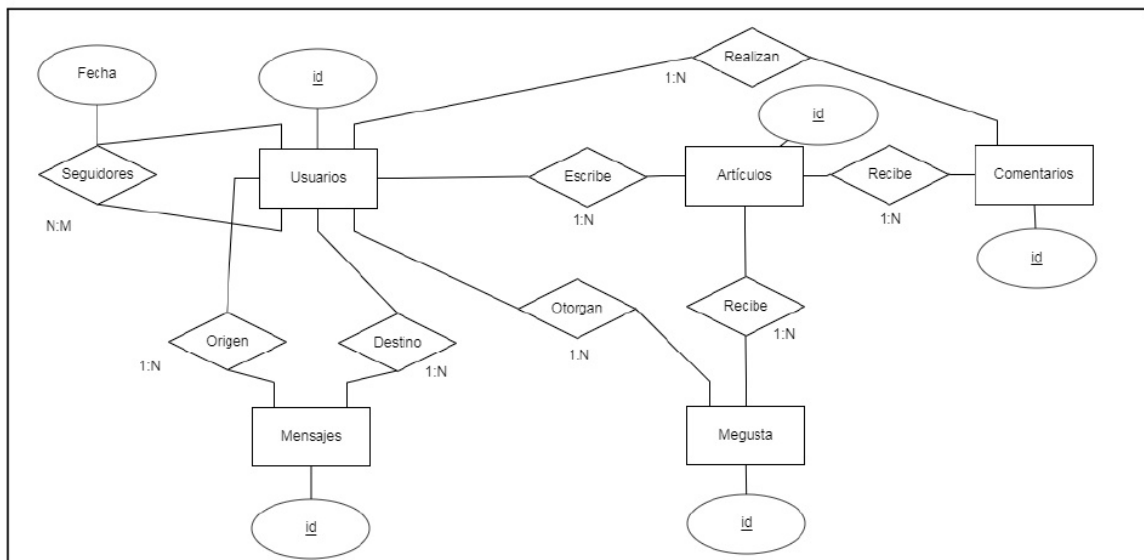


Nombre		Nota	
--------	--	------	--

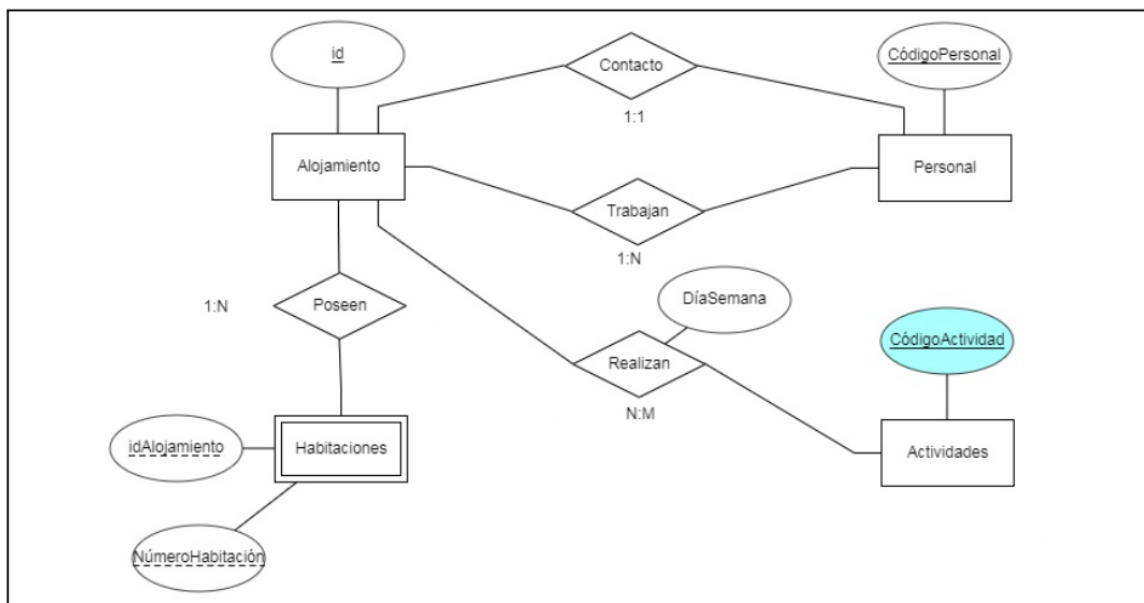
**Ejercicio 1.** Construye el diagrama entidad-relación para la base de datos relacionada con el siguiente supuesto. Representa de forma clara la clave primaria de cada entidad así como los atributos de las relaciones.

Base de datos para la red social "Oceana".

- De los **usuarios** de la red se debe guardar identificador único, pseudónimo (único), nombre, apellidos, correo electrónico, foto de perfil, fecha de alta y fecha de nacimiento.
- Los usuarios escriben **artículos**. Un usuario puede escribir varios artículos y un artículo sólo pertenece a un usuario. De cada artículo se guarda identificador único, título, fecha de publicación, url y texto.
- Un usuario puede tener varios **seguidores** en la red social. Los seguidores son también usuarios de dicha red. Se debe guardar la fecha desde que un usuario sigue a otro. Un usuario puede seguir a varios, y además un usuario puede ser seguido por varios.
- Los artículos pueden recibir **comentarios** por parte de los usuarios. De cada comentario se necesita guardar su identificador único, usuario que lo realizó, contenido, fecha y el artículo al que se refiere. Un artículo puede recibir varios comentarios, y además un usuario puede realizar varios comentarios
- Por otro lado, los usuarios pueden intercambiar **mensajes** entre ellos. Cada mensaje tiene un número que lo identifica, usuario origen, usuario destino, contenido y fecha de creación. Es evidente que un usuario puede enviar y recibir varios mensajes.
- Finalmente, los artículos pueden recibir "**me gusta**" por parte de los usuarios de la red. De cada "me gusta" se guarda un identificador único, usuario que lo concede, artículo con el que está relacionado, texto y fecha en que se otorgó. Un usuario puede conceder varios "me gusta".



**Ejercicio 2.** A partir del diagrama E-R contenido en la imagen, obtén el modelo lógico relacional correspondiente a la base de datos “Alojamientos Rurales” con las características que se citan más adelante. En este modelo lógico deben aparecer todos los campos, pero presta especial atención a claves primarias y ajenas, haciendo que se diferencien claramente e indicando la clave principal a la que apunta cada clave ajena.



- En la imagen sólo aparecen entidades, relaciones, clave principal de cada entidad y atributos de alguna relación N:M.
- De cada **alojamiento** se guarda id, nombre, dirección, localidad, teléfono y correo electrónico.
- Un alojamiento puede tener varias habitaciones, y estas se numeran a partir de 1 para cada alojamiento. De cada **habitación** se registra su capacidad, precio base, si tiene baño o no y si tiene chimenea o no.
- En los alojamientos trabajan personas. De cada **persona** se necesita un código, nombre, correo electrónico y teléfono.
- Una persona sólo trabaja en un alojamiento, pero en un mismo alojamiento trabajan varias personas.
- Cada alojamiento tiene una única persona de contacto, que lo será de un solo alojamiento.
- En los alojamientos se organizan actividades. De cada **actividad** se guarda un código, nombre y descripción.
- Una actividad se puede practicar en varios alojamientos y en un mismo alojamiento se practican varias actividades. Se debe guardar el día de la semana en el que se practica cada actividad en cada alojamiento (que es solo uno).

**Habitaciones** (idAlojamiento, numeroHabitación, capacidad, precio, baño, chimenea)

idAlojamiento apunta a Alojamiento (id)

**Alojamiento** (id, nombre, dirección, localidad, teléfono, email, idCódigoPersonalContacto)

idCódigoPersonalContacto apunta a Personal(CodigoPersonal). Este campo es además único, para mantener la relación 1:1

**Personal** (CodigoPersonal, nombre, email, teléfono, idAlojamiento)

idAlojamiento apunta a Alojamiento (id)

**ActividadesAlojamiento** (idAlojamiento, codActividad, diaSemana)

idAlojamiento apunta a Alojamiento (id)

codActividad apunta a Actividades(código)

**Actividades** (código, nombre, descripción)

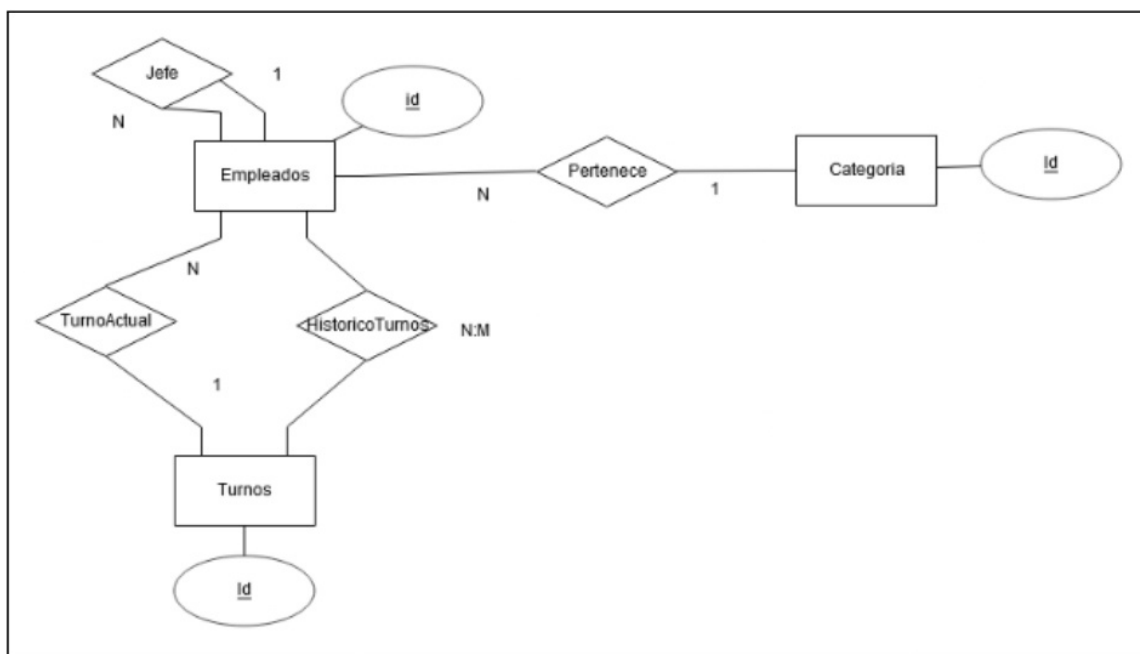
**Ejercicio 3.** Escribe las sentencias SQL que formen el script que sirva para crear la base de datos que se detalla a continuación. Ten en cuenta que partes del diagrama E-R.

Nombre base de datos: GestionTurnos.

Tablas y campos:

- Categoría: id, nombre.
- Empleados: id, nombre, apellidos, dirección, email y teléfono.
- Turnos: id, nombre, descripción.
- De una categoría existen varios empleados. Un empleado solo pertenece a una categoría.
- Un empleado puede ser jefe de varios. Un empleado solo tiene un jefe directo.
- Un empleado tiene un turno de trabajo actual. A lo largo del tiempo, un empleado ha podido trabajar en varios turnos y se necesita conocer entre qué fechas estuvo trabajando en cada turno.

La imagen muestra el diagrama E-R, en el que se observan entidades, relaciones y claves primarias de esas entidades.



Otras consideraciones::

- Los campos que sean claves primarias serán enteros y autoincrementales.
- Todos los campos que almacenan fechas serán de tipo fecha.
- Todas las relaciones se crearán con la opción referencial RESTRICT.

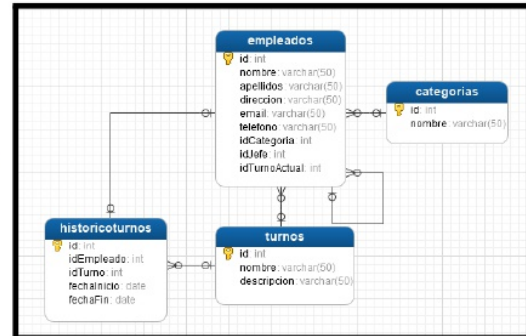
Create database Ejercicio3;  
Use Ejercicio3;

```
create table Categorías(
    id int primary key auto_increment,
    nombre varchar(50) not null
);
```

```
create table Turnos (
    id int primary key auto_increment,
    nombre varchar(50) not null,
    descripcion varchar(50) not null
);
```

```
create table Empleados (
    id int primary key auto_increment,
    nombre varchar(50) not null,
    apellidos varchar(50) not null,
    direccion varchar(50) not null,
    email varchar(50) not null,
    telefono varchar(50) not null,
    idCategoría int,
    idJefe int,
    idTurnoActual int,
    FOREIGN key FK_Cat (idCategoría) REFERENCES Categorías(id),
    Foreign key Fk_Jefe (idJefe) REFERENCES Empleados (id),
    Foreign key Fk_TurnoActual (idTurnoActual) REFERENCES Turnos (id)
);
```

```
create table HistoricoTurnos(
    id int primary key auto_increment,
    idEmpleado int,
    idTurno int,
    fechaInicio date,
    fechaFin date,
    unique (idEmpleado, fechaInicio),
    Foreign key Fk_Empleado (idEmpleado) REFERENCES Empleados (id),
    Foreign key Fk_TurnoHistorico (idTurno) REFERENCES Turnos (id),
);
```





**Ejercicio 4.** Escribe las sentencias SELECT para cada uno de los apartados siguientes. Utiliza una única sentencia en cada caso. La base de datos utilizada será ClassicModels. Puedes ver sus tablas en última página

1. Muestra los clientes con y sin pedidos. Las columnas a mostrar serán: número de cliente, nombre de cliente, número de pedido, fecha del pedido.

```
select c.numeroCliente, c.nombre, p.numeroPedido, p.fechaPedido  
from clientes c left join pedidos p  
on c.numeroCliente=p.numeroCliente
```

2. Muestra todos los datos de los empleados que tienen el mismo puesto de trabajo que el empleado con nombre "Leslie" y apellido "Jennings". Utilizar subselects.

```
select * from empleados where puestoTrabajo  
= (select puestoTrabajo from empleados where  
nombre='Leslie' and apellido='Jennings')
```

3. Mostrar número de cliente, nombre y cuántos pedidos ha realizado. Deben aparecer los clientes con y sin pedidos.

```
select c.numeroCliente, c.nombre, count(*)  
from clientes c left join pedidos p  
on c.numeroCliente=p.numeroCliente  
group by c.numeroCliente
```

4. Mostrar código de producto, nombre de producto, categoría, precio de venta y precio de compra y diferencia entre estos precios. Sólo deben aparecer los productos de las categorías "Classic Cars" y "Planes".

```
select p.codigoProducto, p.nombreProducto, p.categoriaProducto, p.precioCompra,  
p.precioVenta  
from productos p  
where p.categoriaProducto in ('classic Cars','Planes')
```

5. Contar el número de pedidos por estado existente en febrero de 2021. Es decir, cuántos hay entregados, cuántos pendientes de envío, cuántos en reparto, etc. Debes mostrar dos columnas: estado y la cuenta.

```
select estado, count(*) as numeroPedidos  
from pedidos  
GROUP BY estado
```

Notas:

- Puntuación de cada apartado:
  - Ejercicio 1: 3 puntos. Cada elemento mal creado o inexistente resta 0,5 puntos.
  - Ejercicio 2: 2 puntos. Cada error resta 0,4 puntos.
  - Ejercicio 3: 2 puntos (0,5 cada tabla creada correctamente).
  - Ejercicio 4: 3 puntos (0,6 cada consulta correcta).
- No olvides numerar y poner tu nombre en todos los folios que presentes.