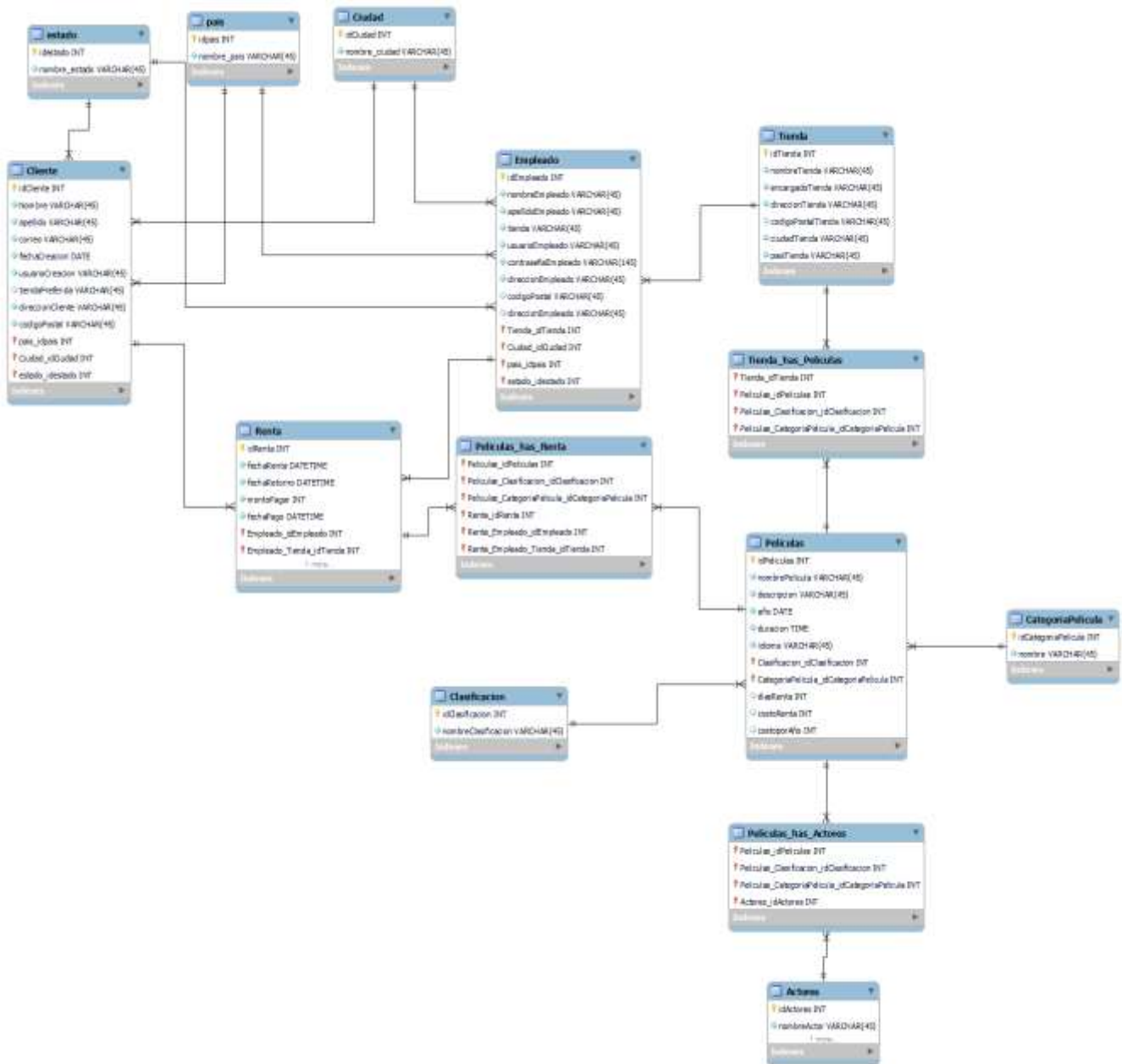


## Documentación Técnica:

Se realizó el diseño de la base de datos sobre el enunciado dado. Se ocupó para su creación el diagrama entidad relación en el cual se ve representada la idea y la manera de uso de la base de datos.



A continuación se realizó la carga de datos con una tabla temporal en la cual se manejaron los datos para la distribución en el modelo relacional con el fin de manejar los datos correctamente.

```
create table public.temporal(  
  NOMBRE_CLIENTE varchar(45),  
  CORREO_CLIENTE varchar(45),  
  CLIENTE_ACTIVO varchar(10),  
  FECHA_CREACION varchar(50),  
  TIENDA_PREFERIDA varchar(45),  
  DIRECCION_CLIENTE varchar(50),  
  CODIGO_POSTAL_CLIENTE varchar(50),  
  CIUDAD_CLIENTE varchar(50),  
  PAIS_CLIENTE varchar(50),  
  FECHA_RENTA varchar(50),  
  FECHA_RETORNO varchar(50),  
  MONTO_A_PAGAR varchar(50),  
  FECHA_PAGO varchar(50),  
  NOMBRE_EMPLEADO varchar(50),  
  CORREO_EMPLEADO varchar(50),  
  EMPLEADO_ACTIVO varchar(10),  
  TIENDA_EMPLEADO varchar(20),  
  USUARIO_EMPLEADO varchar(10),  
  CONTRASEÑA_EMPLEADO varchar(250),  
  DIRECCION_EMPLEADO varchar(100),  
  CODIGO_POSTAL_EMPLEADO varchar(10),  
  CIUDAD_EMPLEADO varchar(100),  
  PAIS_EMPLEADO varchar(100),  
  NOMBRE_TIENDA varchar(100),  
  ENCARGADO_TIENDA varchar(50),  
  DIRECCION_TIENDA varchar(45),  
  CODIGO_POSTAL_TIENDA varchar(10),  
  CIUDAD_TIENDA varchar(50),  
  PAIS_TIENDA varchar(50),  
  TIENDA_PELICULA varchar(50),  
  NOMBRE_PELICULA varchar(50),  
  DESCRIPCION_PELICULA varchar(500),  
  ANIO_LANZAMIENTO varchar(50),  
  DIAS_RENTA varchar(50),  
  COSTO_RENTA varchar(50),  
  DURACION varchar(50),  
  COSTO_POR_DAÑO varchar(50),  
  CLASIFICACION varchar(10),  
  LENGUAJE_PELICULA varchar(45),  
  CATEGORIA_PELICULA varchar(50),  
  ACTOR_PELICULA varchar(50),  
);
```

Se realizaron las siguientes consultas para solucionar las interrogantes que se solicitaron.  
);

```
1 ---Consulta1-----
2 select a.id_pelicula,b.id_tienda
3 from pelicula as a, pelicula_tienda as b
4 where a.nombre_pelicula='SUGAR MONKA' and a.id_pelicula= b.id_pelicula
5
6 ---Consulta2-----
7 select (b.nombre||' '||b.apellido) as name1, sum(monto_pagar)as total_a_pagar
8 from renta as a, cliente as b
9 group by a.id_cliente,name1,b.id_cliente having count(a.id_cliente)>=40 and a.id_cliente=b.id_cliente
10
11 -----Consulta3-----
12 select * from actores
13 where nombre_actores like '%son%'
14
15 -----Consulta4-----
16 select distinct c.nombre_actores,b.pelicula_año
17 from pelicula_actor as a, pelicula as b, actores as c
18 group by a.id_pelicula,c.nombre_actores,b.pelicula_año, b.descripcion, b.id_pelicula
19 having b.descripcion like '%Shark%' and b.descripcion like '%Crocodile%' and a.id_pelicula=b.id_pelicula
20
21 -----Consulta5-----
22
23 select (select pa.nombre_pais from pais as pa where pa.id_pais= b.pais),(b.nombre||' '||b.apellido) as name1,
24 (((select distinct max(count (id_cliente)) OVER () as maxima_renta from renta
25 group by id_cliente)/(select distinct count (renta.id_cliente)
26 from renta as renta inner join cliente cl on cl.id_cliente= renta.id_renta
27 inner join ciudad ci on ci.id_ciudad=cl.id_cliente
28 inner join pais pa on pa.id_pais=cl.pais
29 group by renta.id_cliente, cl.id_cliente, pa.id_pais)))
30 from renta as a, cliente as b
31 group by a.id_cliente,name1,b.id_cliente having count(a.id_cliente)=(select distinct max(count (id_cliente)) OVER () as maxima_renta
32 from renta
33 group by id_cliente)
34 and a.id_cliente=b.id_cliente
35
```