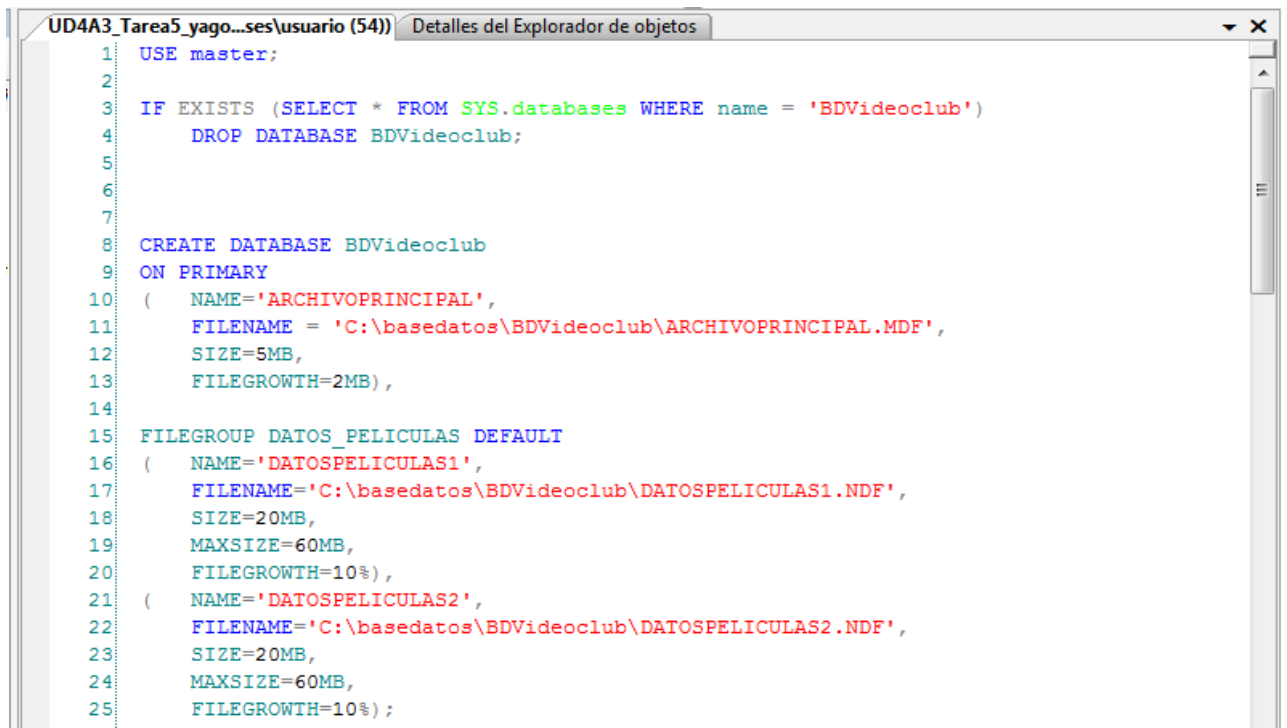


Ejercicio 5 bases de datos

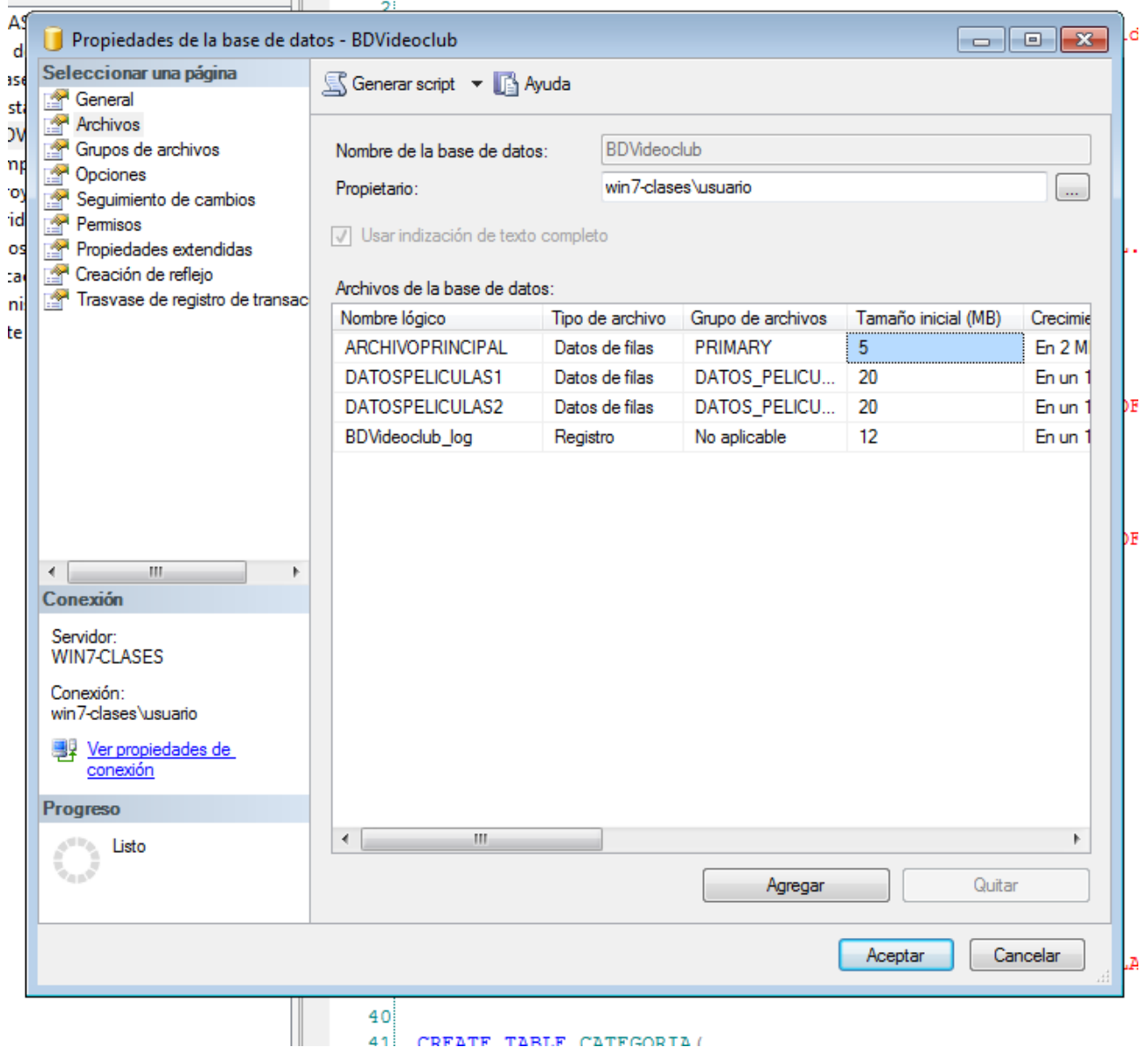
A) Creación de la BD



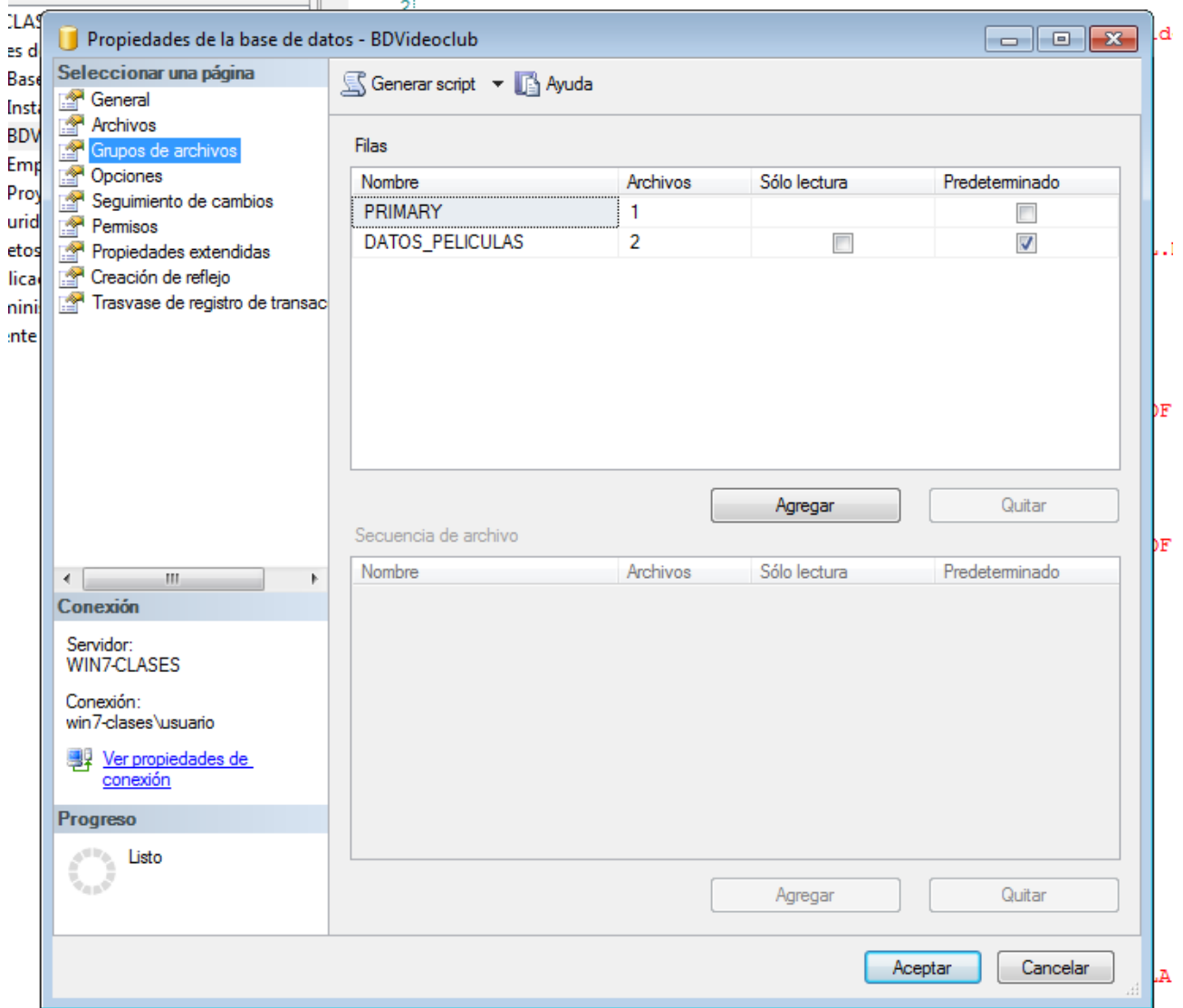
The screenshot shows a window titled "UD4A3_Tarea5_yago...ses\usuario (54)) Detalles del Explorador de objetos". The main area displays a T-SQL script for creating a database named "BDVideoclub". The script includes a "USE master;" statement, a check for the database's existence, and a "DROP DATABASE" statement if it exists. It then proceeds to "CREATE DATABASE BDVideoclub ON PRIMARY" with a primary data file named "ARCHIVOPRINCIPAL.MDF" located at "C:\basedatos\BDVideoclub\ARCHIVOPRINCIPAL.MDF", with a size of 5MB and a file growth of 2MB. Finally, it creates a "FILEGROUP DATOS_PELICULAS DEFAULT" with two secondary data files: "DATOSPELICULAS1.NDF" and "DATOSPELICULAS2.NDF", both located at "C:\basedatos\BDVideoclub\DATOSPELICULAS1.NDF" and "C:\basedatos\BDVideoclub\DATOSPELICULAS2.NDF" respectively, with sizes of 20MB, a maximum size of 60MB, and a file growth of 10%.

```
1 USE master;
2
3 IF EXISTS (SELECT * FROM SYS.databases WHERE name = 'BDVideoclub')
4     DROP DATABASE BDVideoclub;
5
6
7
8 CREATE DATABASE BDVideoclub
9 ON PRIMARY
10 ( NAME='ARCHIVOPRINCIPAL',
11   FILENAME = 'C:\basedatos\BDVideoclub\ARCHIVOPRINCIPAL.MDF',
12   SIZE=5MB,
13   FILEGROWTH=2MB),
14
15 FILEGROUP DATOS_PELICULAS DEFAULT
16 ( NAME='DATOSPELICULAS1',
17   FILENAME='C:\basedatos\BDVideoclub\DATOSPELICULAS1.NDF',
18   SIZE=20MB,
19   MAXSIZE=60MB,
20   FILEGROWTH=10%),
21 ( NAME='DATOSPELICULAS2',
22   FILENAME='C:\basedatos\BDVideoclub\DATOSPELICULAS2.NDF',
23   SIZE=20MB,
24   MAXSIZE=60MB,
25   FILEGROWTH=10%);
```

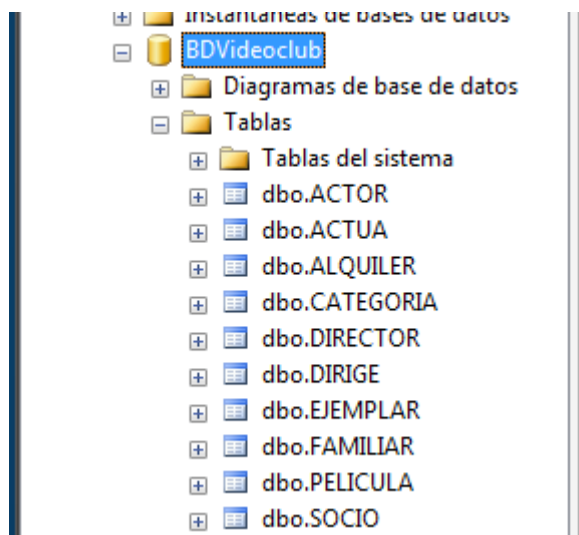
Archivos



Grupo de archivos



B) Creación de todas las tablas y restricciones.



Tipos

```
30 |
31 | IF EXISTS (SELECT * FROM SYS.types WHERE name = 'FECHA')
32 |     DROP TYPE FECHA
33 |
34 |
35 | CREATE TYPE FECHA
36 |     FROM SMALLDATETIME NOT NULL
37 |
38 | IF EXISTS (SELECT * FROM SYS.TYPES WHERE NAME ='NACIONALIDAD')
39 |     DROP TYPE NACIONALIDAD;
40 |
41 | CREATE TYPE NACIONALIDAD
42 |     FROM VARCHAR(15);
43 |
```

Tabla categoría

```
48 |
49 | CREATE TABLE CATEGORIA(
50 |     CODCATEGORIA    TINYINT,
51 |     PRECIO MONEY,
52 |     DESCRIPCION VARCHAR(30) NULL,
53 |
54 |     CONSTRAINT PK_CATEGORIA PRIMARY KEY (CODCATEGORIA)
55 | )
56 |
```

Película

```

CREATE TABLE PELICULA(
    IDPELICULA VARCHAR(7),
    TITULO VARCHAR(20),
    AÑOPRODUCCION SMALLINT,
    GENERO VARCHAR(15),
    DURACION SMALLINT,
    NACIONALIDAD VARCHAR(15) NULL,
    CODCATEGORIA TINYINT,

    CONSTRAINT FK_CODCATEGORIA FOREIGN KEY (CODCATEGORIA)
    REFERENCES CATEGORIA(CODCATEGORIA) ON UPDATE
    CASCADE ON DELETE NO ACTION,

    CONSTRAINT CK_IDPELICULA CHECK (
        (IDPELICULA LIKE '[PF][A-Z][-][0-9][0-9][0-9][0-9]')
    ),

    CONSTRAINT PK_PELICULA PRIMARY KEY (IDPELICULA),
- ) ON [PRIMARY];

```

Alquiler

```

CREATE TABLE ALQUILER(
    IDEJEMPLAR TINYINT,
    IDPELICULA VARCHAR(7),
    FECHAALQ DATE,
    NUMEROSOCIO SMALLINT,
    FECHAESTIMADADEV DATE DEFAULT DATEADD(DD, 15, 'FECHAALQ'),
    FECHADEV DATE NULL,

    CONSTRAINT PK_ALQUILER PRIMARY KEY (IDEJEMPLAR, IDPELICULA, FECHAALQ),
    CONSTRAINT FK_IDEJEMPLAR FOREIGN KEY (IDPELICULA, IDEJEMPLAR) REFERENCES
    EJEMPLAR(IDPELICULA, IDEJEMPLAR) ON DELETE CASCADE ON UPDATE CASCADE,
- )

```

Actor

```

CREATE TABLE ACTOR(
    NOMBRE VARCHAR(20),
    NACIONALIDAD NACIONALIDAD,
    SEXO CHAR(1),

    CONSTRAINT PK_ACTOR PRIMARY KEY (NOMBRE),
    CONSTRAINT CK_SEXO CHECK(
        (SEXO LIKE '[HM]')
    )
- )

```

Actua

```

CREATE TABLE ACTUA (
    NOMBREACTOR VARCHAR(20),
    IDPELICULA VARCHAR(7),
    PRINSEC CHAR(1),

    CONSTRAINT CK_PRINSEC CHECK(
        (PRINSEC LIKE '[PS]')
    ),

    CONSTRAINT PK_ACTUA PRIMARY KEY (NOMBREACTOR, IDPELICULA),

    CONSTRAINT FK_NOMBREACTOR FOREIGN KEY (NOMBREACTOR) REFERENCES ACTOR (NOMBRE)
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,

    CONSTRAINT FK_IDPELICULA_ACTOR FOREIGN KEY (IDPELICULA) REFERENCES PELICULA
)

```

Director

```

CREATE TABLE DIRECTOR (
    NOMBRE VARCHAR(20),
    NACIONALIDAD NACIONALIDAD,

    CONSTRAINT PK_DIRECTOR PRIMARY KEY (NOMBRE)
)

```

Dirige

```

CREATE TABLE DIRIGE (
    IDPELICULA VARCHAR(7),
    NOMBREDIRECTOR VARCHAR(20),

    CONSTRAINT PK_DIRIGE PRIMARY KEY (IDPELICULA, NOMBREDIRECTOR),

    CONSTRAINT FK_IDPELICULA_DIRIGE FOREIGN KEY (IDPELICULA) REFERENCES PELICULA (IDPELICULA) ON UPDATE NO ACTION ON DELETE NO ACTION,

    CONSTRAINT FK_NOMBREDIRECTOR FOREIGN KEY (NOMBREDIRECTOR) REFERENCES DIRECTOR (NOMBRE) ON UPDATE CASCADE ON DELETE NO ACTION
)

```

Familiar

```

CREATE TABLE FAMILIAR (
    DNI CHAR(9),
    NOMBRE VARCHAR(60),
    PARENTESCO VARCHAR(15),
    FECHANACIMIENTO DATE,
    NUMEROSOCIO TINYINT,

    CONSTRAINT PK_FAMILIAR PRIMARY KEY (DNI),

) ON [PRIMARY];

```

```

0 CREATE TABLE EJEMPLAR (
1     IDPELICULA VARCHAR(7),
2     IDEJEMPLAR TINYINT,
3     ESTADO CHAR(1),
4     FECHACOMPRA DATE DEFAULT DATEADD(dd, -2, getDate()),
5
6
7     CONSTRAINT CK_ESTADO CHECK (
8         (ESTADO LIKE '[BRM]')
9     ),
10
11     CONSTRAINT FK_IDPELICULA FOREIGN KEY (IDPELICULA) REFERENCES PELICULA(IDPELICULA),
12
13     CONSTRAINT PK_EJEMPLAR PRIMARY KEY (IDPELICULA, IDEJEMPLAR)
14 ) ON DATOS_PELICULAS;
15

```

socio

```

06
07 CREATE TABLE SOCIO (
08     NUMERO SMALLINT IDENTITY,
09     DNI TINYINT,
10     NOMBRE VARCHAR(20),
11     APELLIDOS VARCHAR(20),
12     DIRECCION VARCHAR(30),
13     TELEFONO CHAR(9),
14     FECHADEALTA DATE,
15
16     CONSTRAINT PK_SOCIO PRIMARY KEY (NUMERO),
17     CONSTRAINT CK_DNI CHECK (
18         (TELEFONO LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
19     ),
20     CONSTRAINT UNI_DNI UNIQUE (DNI)
21 )
22

```

Ejemplar

3) Modificaciones

3.1) Añade una columna a la tabla Alquiler llamada PrecioAlquiler con el tipo de datos money. Es obligatorio. Por defecto toma el valor 4 euros y todos los registros que ya existan en la tabla tomarán este valor.

```

3
4 ALTER TABLE ALQUILER
5     ADD PRECIOALQUILER MONEY NOT NULL DEFAULT 4 WITH VALUES
6
7

```

2) Haz que las siguientes columnas tenga los siguientes valores predeterminados:

- En FechadeAlquiler y en la fecha de alta de un socio, la fecha actual

```

1 ALTER TABLE ALQUILER
2     ADD CONSTRAINT DF_FECHADEALQUILER DEFAULT GETDATE() FOR FECHAALQ

```

3.3) Añade la restricción para que el precio en la tabla categoría no pueda ser inferior a 1 euro ni superior a 300 euros. Se ha de validar que los datos ya existentes en esa columna son los mencionados.

```
ALTER TABLE CATEGORIA
    WITH CHECK ADD CONSTRAINT CK_PRECIO CHECK (
        ((PRECIO) > 1 AND (PRECIO) < 300)
    )
```

3.4) Borra el campo PrecioAlquiler de la tabla Alquiler.

```
ALTER TABLE ALQUILER
    DROP CONSTRAINT DF__ALQUILER__PRECIO__1ED998B2,
    COLUMN PRECIOALQUILER
```



```

229
230 CREATE TABLE DISTRIBUIDORA(
231     ID TINYINT IDENTITY(1,1),
232     NOMBRE VARCHAR(20),
233     DIRECCION VARCHAR(50),
234     FAX CHAR(9) NULL,
235     EMAIL VARCHAR(60) NULL,
236
237     CONSTRAINT UQ_NOMBRE UNIQUE (NOMBRE),
238     CONSTRAINT PK_DISTRIBUIDORA PRIMARY KEY (ID)
239 )
240
241 CREATE TABLE DISTRIBUYE(
242     ID_DISTRIBUIDORA TINYINT ,
243     ID_PELICULA VARCHAR(7),
244
245     CONSTRAINT PK_DISTRIBUYE PRIMARY KEY (ID_DISTRIBUIDORA , ID_PELICULA),
246     CONSTRAINT FK_DISTRIBUIDORA FOREIGN KEY (ID_DISTRIBUIDORA) REFERENCES DISTRIBUIDORA(ID),
247     CONSTRAINT FK_PELICULA FOREIGN KEY (ID_PELICULA) REFERENCES PELICULA(IDPELICULA)
248 )
249
250
251

```

```

:7     CONSTRAINT FK_PELICULA FOREIGN KEY (ID_PELICULA) REFERENCES PELICULA(IDPELICULA)
:8     ON DELETE NO ACTION ON UPDATE CASCADE
:9 )
:

```

```
0 |
1 | CREATE TABLE SOCIO_RECOMENDADO(
2 |     ID_RECOMENDACION SMALLINT,
3 |     ID_RECOMENDADO SMALLINT,
4 |
5 |     CONSTRAINT FK_RECOMENSACION FOREIGN KEY (ID_RECOMENDACION) REFERENCES SOCIO(NUMERO)
6 |         ON DELETE SET NULL ON UPDATE CASCADE
7 | )
8 |
```