



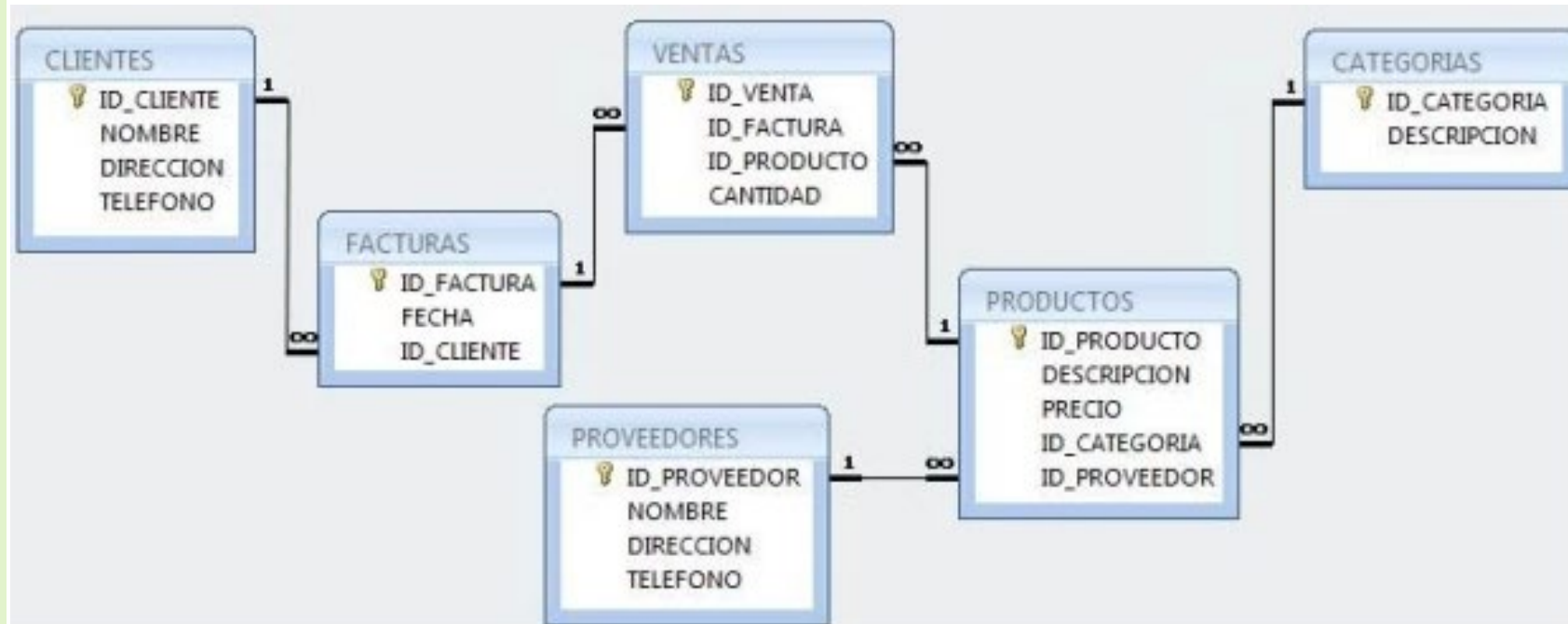
EFA
MORATALAZ

*1º CFGS Desarrollo de
Aplicaciones Multiplataforma*

BASES DE DATOS

JESÚS SANTIAGO RICO

UT3 – EL MODELO RELACIONAL





EFA
MORATALAZ

*1º CFGS Desarrollo de Aplicaciones
Multiplataforma*

BASES DE DATOS

UT3 – EL MODELO RELACIONAL

1. INTRODUCCIÓN MODELO RELACIONAL
2. VENTAJAS Y DESVENTAJAS DEL MODELO RELACIONAL
3. NOTACIÓN DE UNA RELACIÓN
4. ELEMENTOS DEL MODELO RELACIONAL
5. OBTENER EL MODELO RELACIONAL A PARTIR DEL DIAGRAMA E/R
6. GENERACIÓN BBDD A PARTIR DEL MODELO RELACIONAL

INTRODUCCIÓN MODELO RELACIONAL

1

- Definido por Edgar F. Codd en 1970 en el artículo: “A **Relational Model** of Data for Large Shared Data Banks.
- El modelo relacional buscaba dar una solución a los problemas derivados de la rigidez estructural de las bases de datos jerárquicas de la época.
- El elemento básico del modelo relacional es la **Relación**, una estructura bidimensional que representa las entidades y algunas relaciones de los diagramas de entidad-relación.
- Informalmente en el modelo relacional: **Entidad = Tabla**.
- Generó el nacimiento de los **RDBMS** (SGBD Relacionales).

VENTAJAS Y DESVENTAJAS DEL MODELO RELACIONAL



- **Ventajas**

- Sencillo y fácil de modificar.
- Operadores con gran poder de manipulación de datos.
- Compatibilidad y estandarización.
- Estabilidad.
- Conectividad garantizada con los lenguajes de programación estándar (Java, C++...).

- **Desventajas**
 - No incluye comportamiento de los datos.
 - No se puede manejar herencia.
 - Descompone los elementos de interés en varias tablas.
 - Presenta dificultades para el manejo de datos no atómicos.

NOTACIÓN DE UNA RELACIÓN



- El esquema de una relación R (una entidad en los diagramas de Entidad/Relación) se denota: $R(A_1, A_2, \dots, A_n)$.
- En esta notación, R es el nombre de la relación y A_1, A_2, \dots, A_n son los atributos de R.
- Ejemplo:
 - **EMPLEADO(num_empleado, nombre, apellido, sueldo)**

NOTACIÓN HUNGARA



La notación húngara, es una forma nombrar a las variables para poder identificarlas dentro de un programa. Consiste en añadir prefijos en minúsculas al nombre de la variable para identificar de que tipo son.

Para las bases de datos usaremos una aproximación a este concepto.

Regla	Procedimiento	Ejemplo
Nombre de tabla	Se añade una “T” al comienzo del nombre de la tabla	Tautor
Campo tipo numérico	Se añade “n” al comienzo del nombre del campo	nIdentificador
Campo tipo cadena	Se añade “c” al comienzo del nombre del campo	cNombre
Campo tipo fecha	Se añade “d” al comienzo del nombre del campo	dNacimiento / dFechaNacimiento
Campo tipo lógico	Se añade “l” al comienzo del nombre del campo	lSexo
Campo de tipo objeto	Se añade “o” al comienzo del nombre del campo	oFoto
Identificar campo PK	Se añade ID al final del nombre del campo	cIdProductoID

ELEMENTOS DEL MODELO RELACIONAL

5

- **TUPLA:** Cada instancia, fila o registro de una relación es una tupla. Un conjunto de tuplas es una relación.
- **ATRIBUTO:** Cada campo o columna de una relación es un atributo. El conjunto de atributos forman la cabecera de la relación.
- **DOMINIO:** Es el conjunto de los posibles valores que puede tomar un atributo, es decir, el **tipo de datos** que puede almacenar un atributo: Boolean, Integer, Character, etc.
- **CLAVE CANDIDATA:** Atributo o atributos que identifican de manera única una tupla. Por ejemplo, un coche se puede identificar por su número de matrícula o por el número de bastidor.

5. Elementos del modelo relacional


valor

atributo

tupla

NIF	Nombre	Apellidos	Teléfono	Fecha de nacimiento	Fecha de alta
00789521T	Paula	Sanz González	619554687	13/09/1983	20/09/2012
09653801B	José Luis	García Viñals	667859621	05/02/1963	14/09/2011
38546958X	Javier	Peinado Martín	666932541	24/10/1978	05/05/2009
50687452Y	Ruth	Lázaro Cardenal	689330247	15/05/1981	04/09/2013

CLAVES CANDIDATAS A SER CLAVE PRIMARIA



NMatrícula	NMotor	Marca	Modelo	...
CCA-341	91234908123	Toyota	Yaris	
OFG-851	53489787679	Fiat	Fiorino	
XTV-657	30752312386	Ford	Mustang	
WGB-959	50934187123	Toyota	Avensis	

- **CLAVE PRIMARIA (Primary Key):** Elegida a partir de las claves candidatas de la relación. El resto de posibles claves candidatas quedan como claves alternativas o secundarias. La clave primaria no puede contener valores nulos (aparece subrayada en la notación)
- **CLAVE AJENA (Clave Foránea o Foreign Key):** Atributo o conjunto de atributos que establecen la forma en que dos tablas se relacionan (aparece subrayada en puntos discontinuos en la notación) Todo valor de clave ajena debe existir como clave primaria en la tabla maestra (Integridad Referencial).

OBTENER EL MODELO RELACIONAL A PARTIR DEL DIAGRAMA E/R



- **Reglas de transformación del Diagrama E/R a un Modelo Relacional**
 - **Transformación de dominios:** es equivalente en ambos modelos.
 - Atributo → Atributo
 - **Transformación de entidades**
 - Entidad → Relación/Tabla
 - **Transformación de atributos**
 - Atributo principal → Clave primaria
 - Atributos no clave → Columnas de la tabla
 - Atributos multivaluados → Nueva tabla cuyo identificador es el atributo principal de la entidad original más el propio atributo

NOTACION EN FORMA DE TABLA

TAutor
<u>nAutorID</u>
cNombre
cApellidos

NOTACION EN FORMA RELACIONAL

AUTOR(nAutorID, cNombre, cApellidos)

- Reglas de transformación del Diagrama E/R a un Modelo Relacional

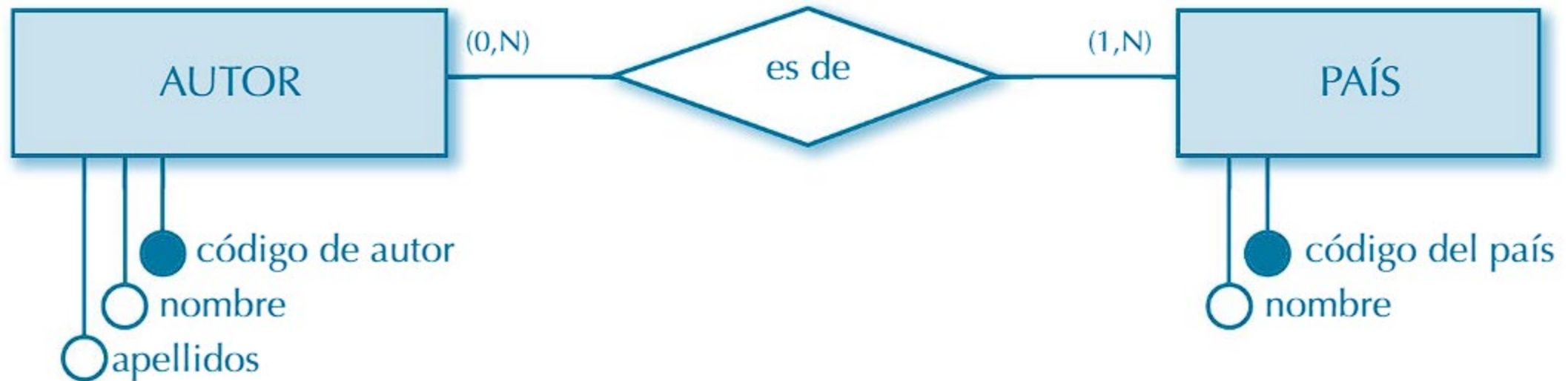
- Entidades fuertes:

- Por cada entidad fuerte del diagrama E/R crearemos una tabla, cuyo nombre será el nombre de la entidad (ahora es una relación/tabla), y que contiene todos sus atributos simples (ahora serán las columnas de dicha tabla). En la notación de la relación el atributo clave aparece subrayado.



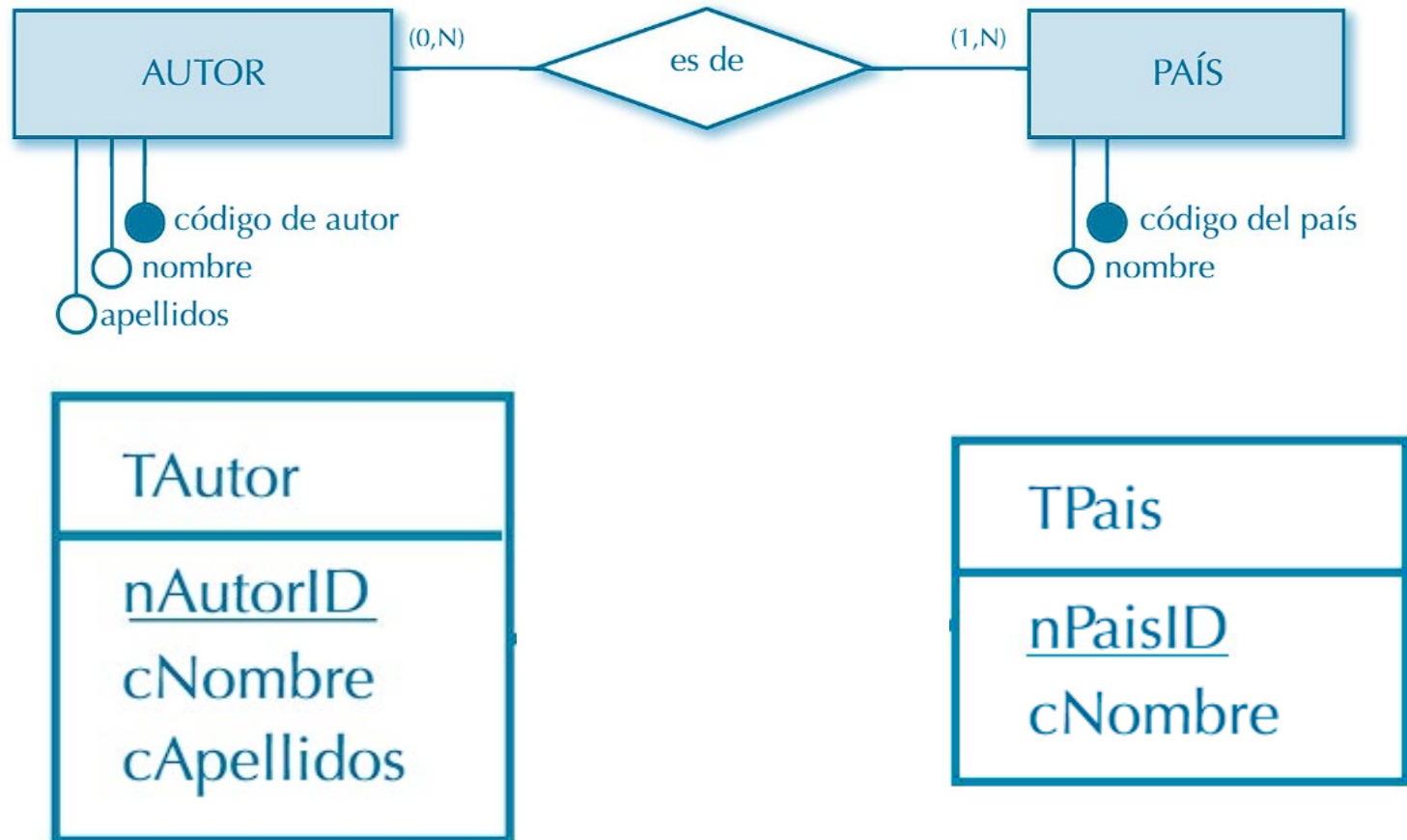
- Empleado(cDNIID, cNombre, cApellido,...)
- Departamento(nCod_depID, cNombreDept, nTelefono,...)

- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Entidades fuertes:



- AUTOR(nCodAutorID, cNombre, cApellidos)
- PAIS(nCodPaisID, cNombre)

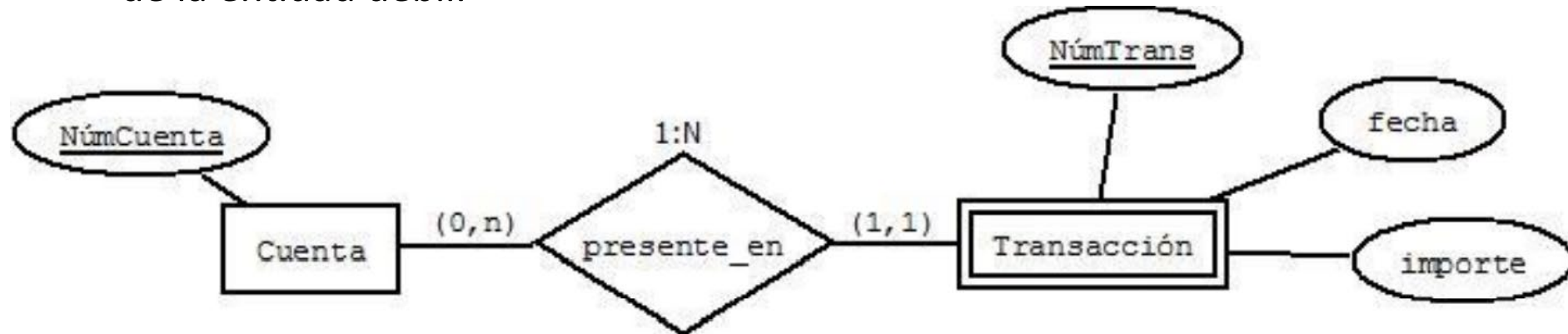
- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Entidades fuertes (Otra forma de representarlas):



- Reglas de transformación del Diagrama E/R a un Modelo Relacional

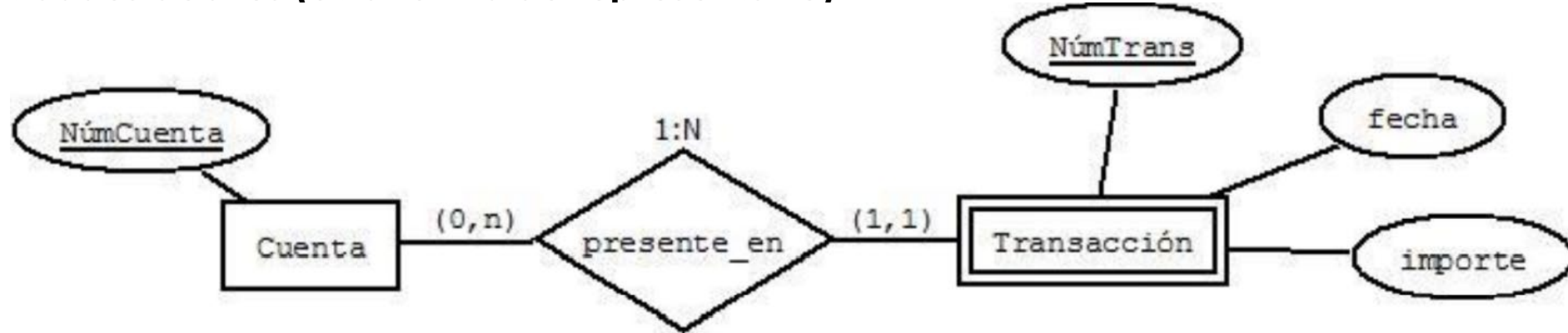
- Entidades débiles:

- Por cada entidad débil del diagrama E/R crearemos una tabla, cuyo nombre será el nombre de la entidad y que contiene todos sus atributos simples. Además, se añade como atributo (será una clave ajena) el atributo principal de la entidad fuerte de quien depende. La clave principal estará formada por la clave de la entidad fuerte y la clave de la entidad débil.



- Cuenta(nNumCuentaID)
- Transaccion(nNumCuentaID, nNumTransID, dFecha, nImporte)

- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Entidades débiles (otra forma de representarlo):



TCuenta
nNumCuentaID

Ttransacion
nNumCuentaID
nNumTransID
Fecha
importe

- Representación de relaciones

- Modalidad 0:1



- Modalidad 1:1



- Representación de relaciones

- Modalidad 1:N



- Modalidad 0:N



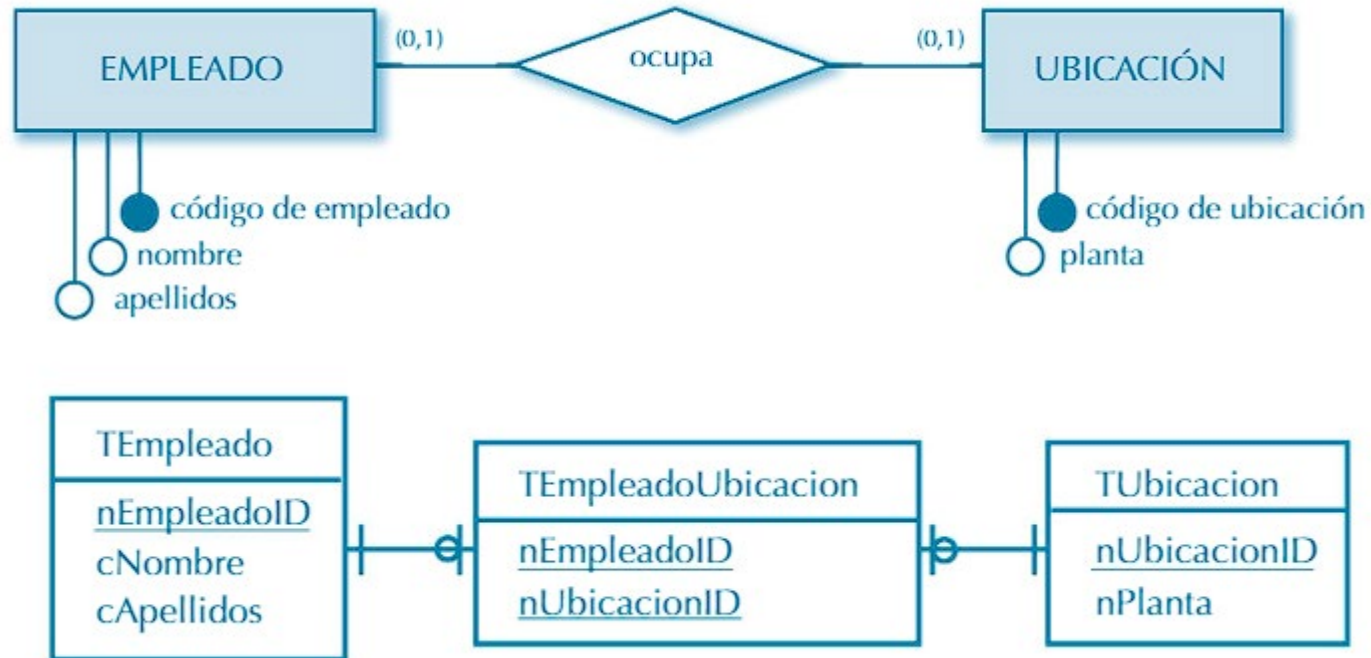
- **Reglas de transformación del Diagrama E/R a un Modelo Relacional**
 - **Relaciones 1:1**
 - Este tipo de relaciones es el más complejo y se pueden dar 4 tipos de casuísticas:
 - ✓ Si ambas modalidades (cardinalidades en los extremos) es 0:1.
 - ✓ Si una modalidad es 0:1 y otra 1:1.
 - ✓ Si ambas modalidades 1:1 y una de las entidades es débil.
 - ✓ Si ambas modalidades 1:1

- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:1 (Ambas modalidades 0:1)
 - Se genera una tabla nueva y los atributos son las PK'S de las tablas con las que se relaciona, siendo estos atributos a su vez PK de la nueva tabla.

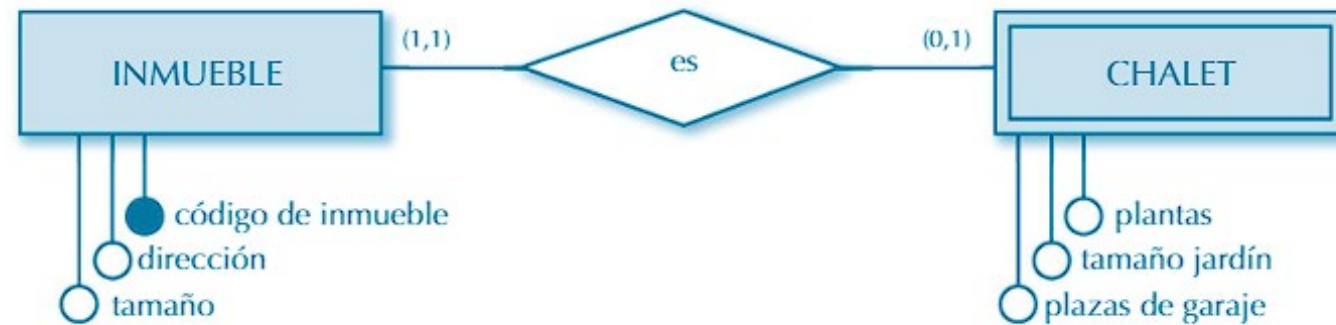


- EMPLEADO(nEmpleadoID, cNombre, cApellidos)
- UBICACION(nUbicacionID, nPlanta)
- EMPLEADOUbicACION(nEmpleadoID, nUbicacionID)

- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:1 (Ambas modalidades 0:1)
 - Se genera una tabla nueva y los atributos son las PK'S de las tablas con las que se relaciona, siendo estos atributos a su vez PK de la nueva tabla.

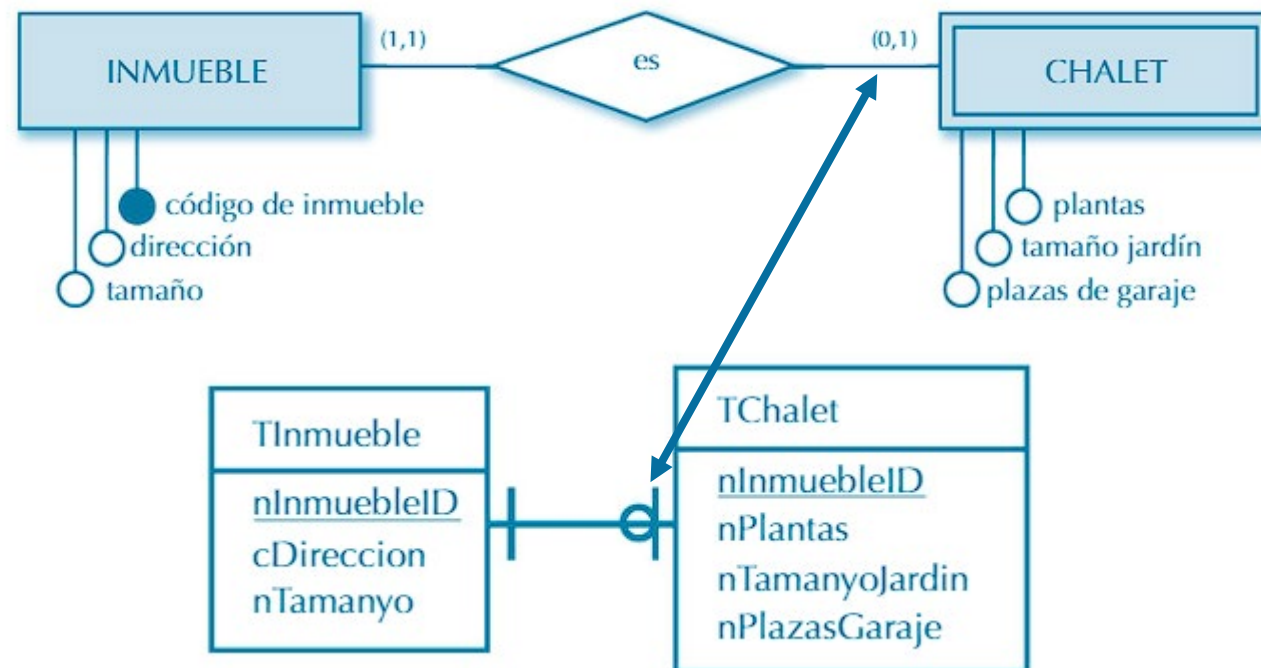


- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:1 (Si una modalidad es 0:1 y otra 1:1)
 - La PK de la entidad con modalidad mínima 1 se propaga a la entidad con modalidad mínima 0, haciendo que esta clave propagada sea PK en la nueva entidad.



- INMUEBLE(nInmuebleID, cDireccion, nTamanio)
- CHALET(nInmuebleID, nPlantas, nTamanioJardin, nPlazasGaraje)

- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:1 (Si una modalidad es 0:1 y otra 1:1)
 - La PK de la entidad con modalidad mínima 1 se propaga a la entidad con modalidad mínima 0, haciendo que este atributo propagado sea PK en la entidad destino.

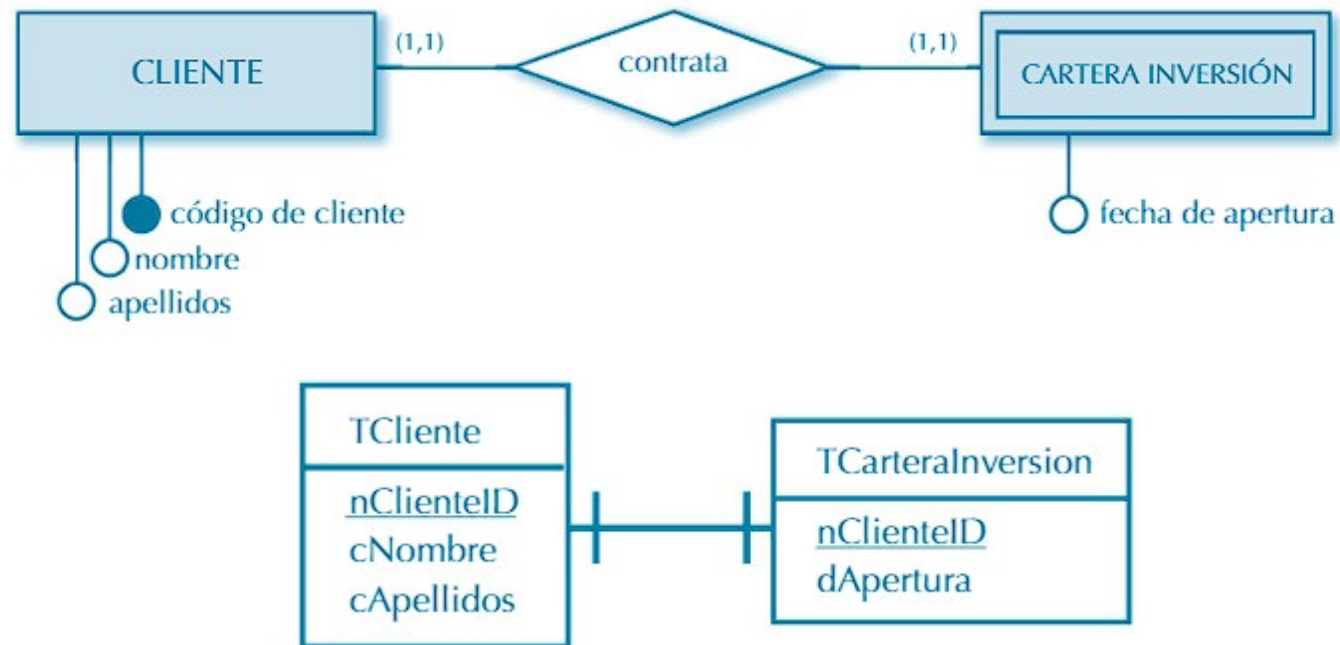


- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:1 (Si ambas modalidades 1:1 y una de las entidades es débil)
 - La PK de la entidad fuerte se propaga a la entidad débil, siendo también PK en la entidad propagada.



- CLIENTE(nClienteID, cNombre, cApellidos)
- CARTERAINVERSION(nClienteID, dApertura)

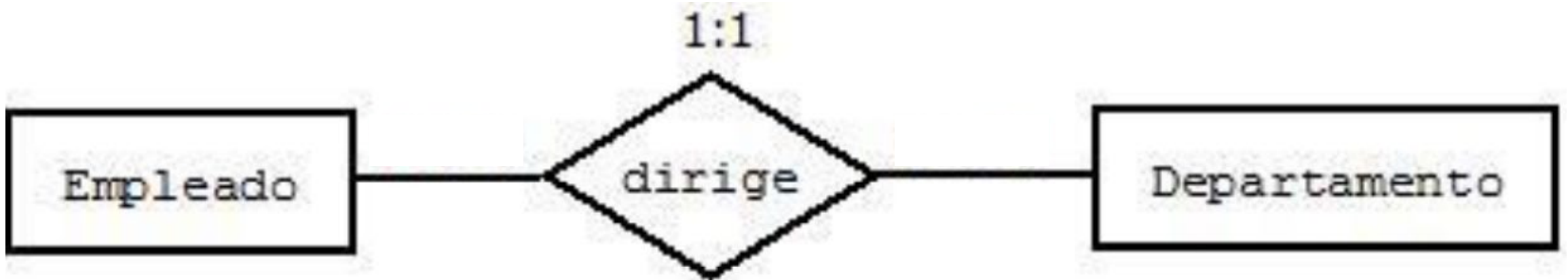
- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:1 (Si ambas modalidades 1:1 y una de las entidades es débil)
 - La PK de la entidad fuerte se propaga a la entidad débil, siendo también PK en la entidad propagada.



- Reglas de transformación del Diagrama E/R a un Modelo Relacional

- Relaciones 1:1

- Por cada relación 1:1 del Diagrama E/R, identificamos las tablas que participan en la relación 1:1 (R y T). Elegimos una de las tablas (R, por ejemplo) y añadimos como clave ajena de R la clave principal de T.



- Empleado(nDNIID, Nombre, Apellido,...)
- Departamento(nCodDepID, NombreDept, Telefono,..., nDNI)

- **Reglas de transformación del Diagrama E/R a un Modelo Relacional**
 - **Relaciones 1:N**
 - En este tipo de relaciones nos podemos encontrar 3 tipos de casuísticas:
 - ✓ Si la relación tiene atributos.
 - ✓ Si la relación no tiene atributos y la cardinalidad mínima del lado 1 es 1.
 - ✓ Si la relación no tiene atributos y la cardinalidad mínima del lado 1 es 0.

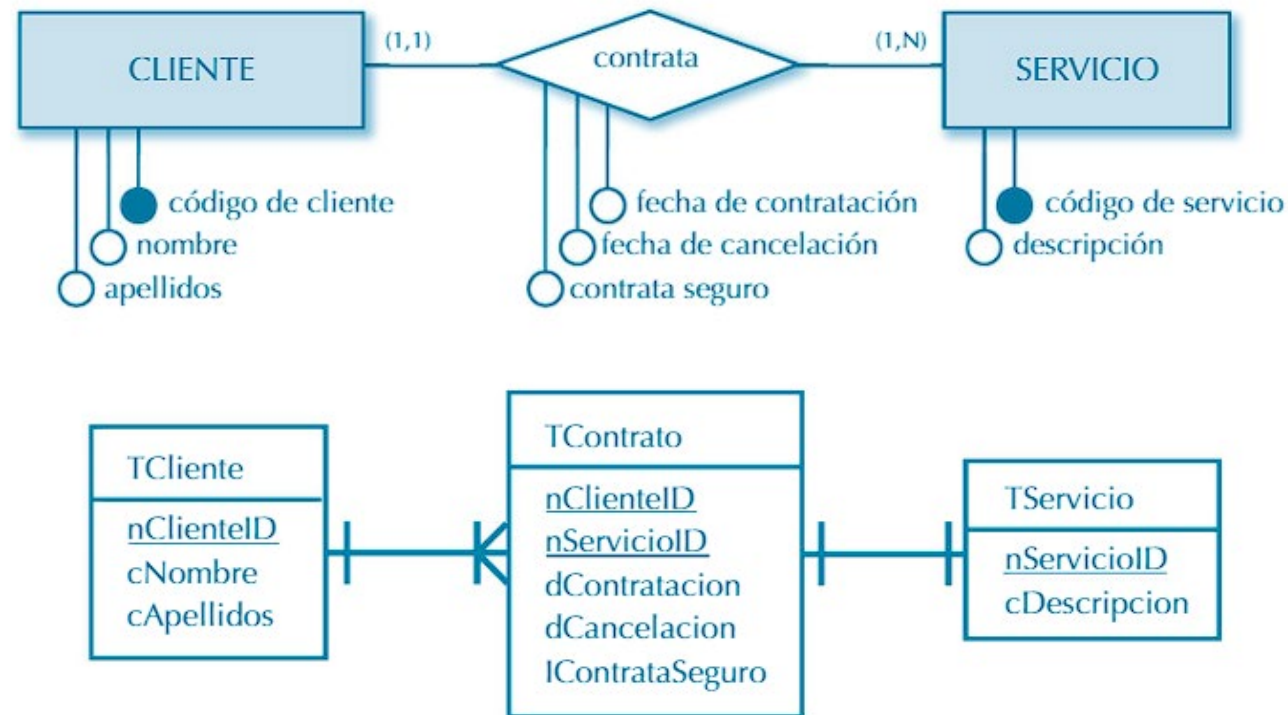
- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:N (Con Atributos en la relación)**
 - La relación donde haya atributos se transforma en tabla y tiene como atributos las PK de las tablas que relaciona, adicionalmente hay que añadir el resto de campos de la relación. T.



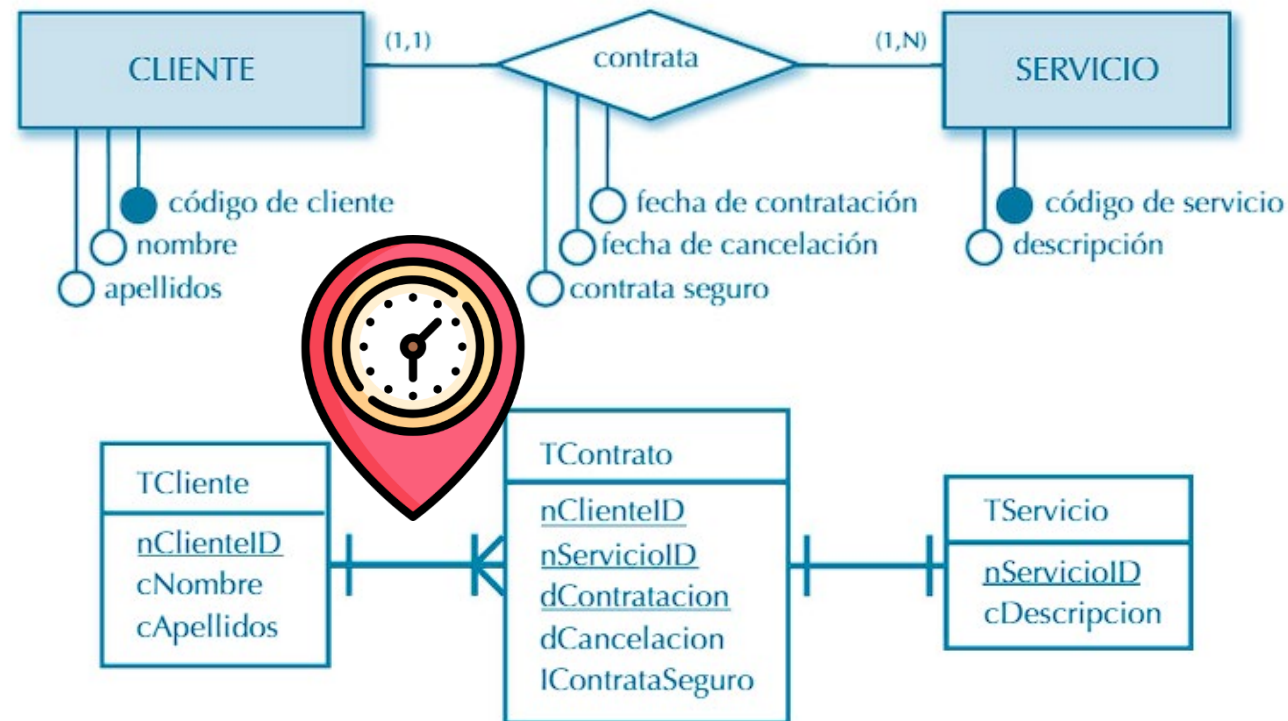
- CLIENTE(nClienteID, cNombre, cApellidos)
- SERVICIO(nServicioID, cDescripcion)
- CONTRATO (nClienteID, nServicioID, dContratacion, dCancelacion, lContrataSeguro)



- **Reglas de transformación del Diagrama E/R a un Modelo Relacional**
 - **Relaciones 1:N (Con Atributos en la relación)**
 - La relación donde haya atributos se transforma en tabla y tiene como atributos las PK de las tablas que relaciona, adicionalmente hay que añadir el resto de campos de la relación. T.



- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:N (Con Atributos en la relación)**
 - La relación donde haya atributos se transforma en tabla y tiene como atributos las PK de las tablas que relaciona, adicionalmente hay que añadir el resto de campos de la relación. T.

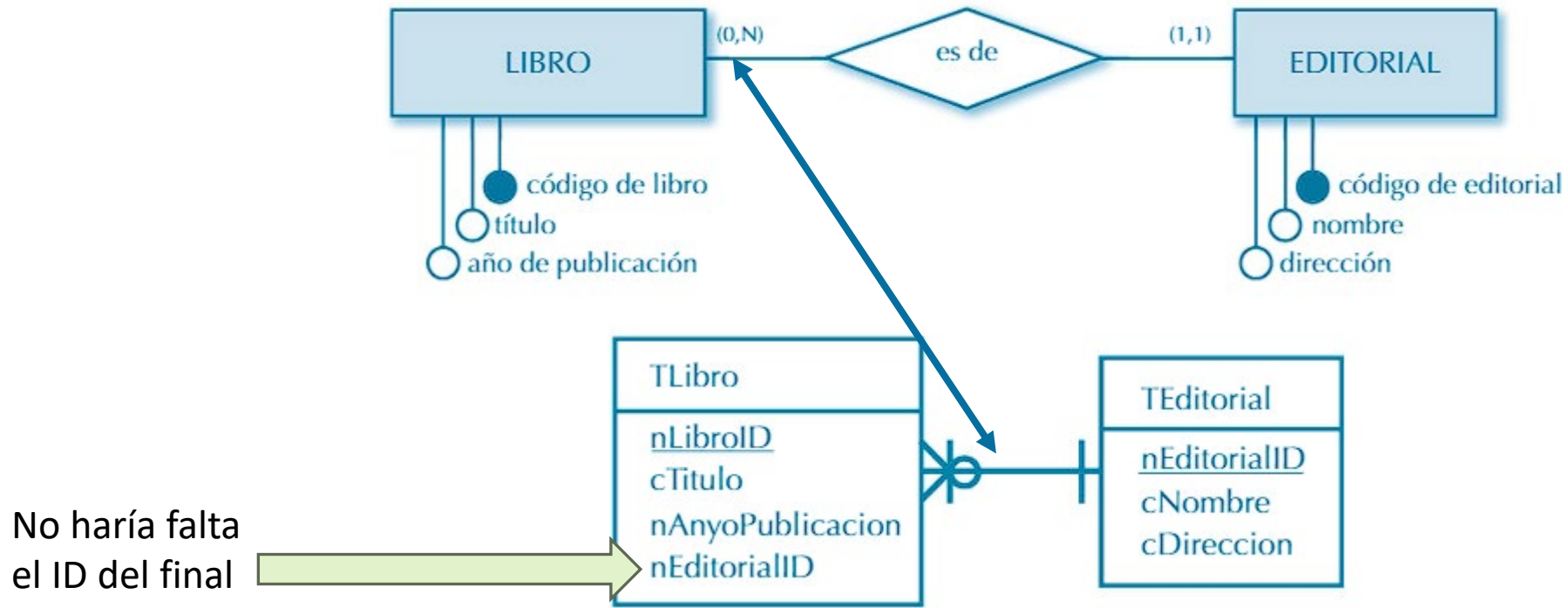


- **Reglas de transformación del Diagrama E/R a un Modelo Relacional**
 - **Relaciones 1:N (Sin Atributos en la relación y la cardinalidad mínima del lado 1 es 1)**
 - La PK de la entidad cuya cardinalidad máxima es 1, se propaga como atributo (FK) a la entidad con cardinalidad máxima N.

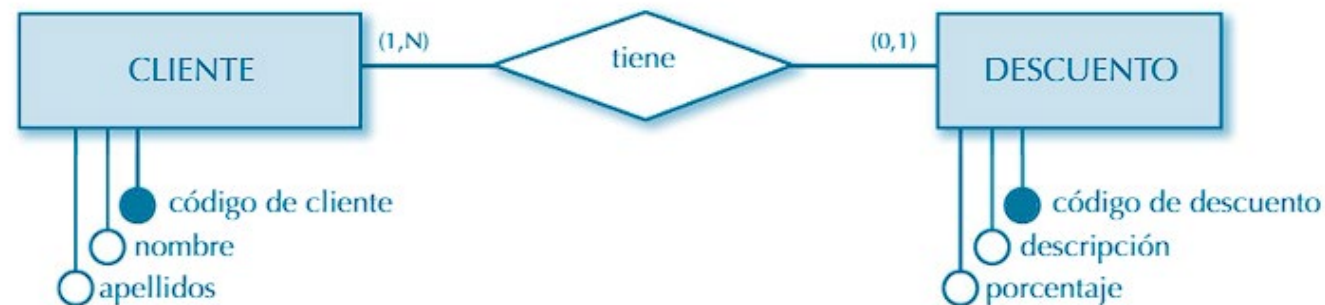


- LIBRO(nLibroID, cTitulo, nAnioPublicacion, nEditorial)
- EDITORIAL(nEditorialID, cNombre, cDireccion)

- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:N (Sin Atributos en la relación y la cardinalidad mínima del lado 1 es 1)
 - La PK de la entidad cuya cardinalidad máxima es 1, se propaga como atributo (FK) a la entidad con cardinalidad máxima N.

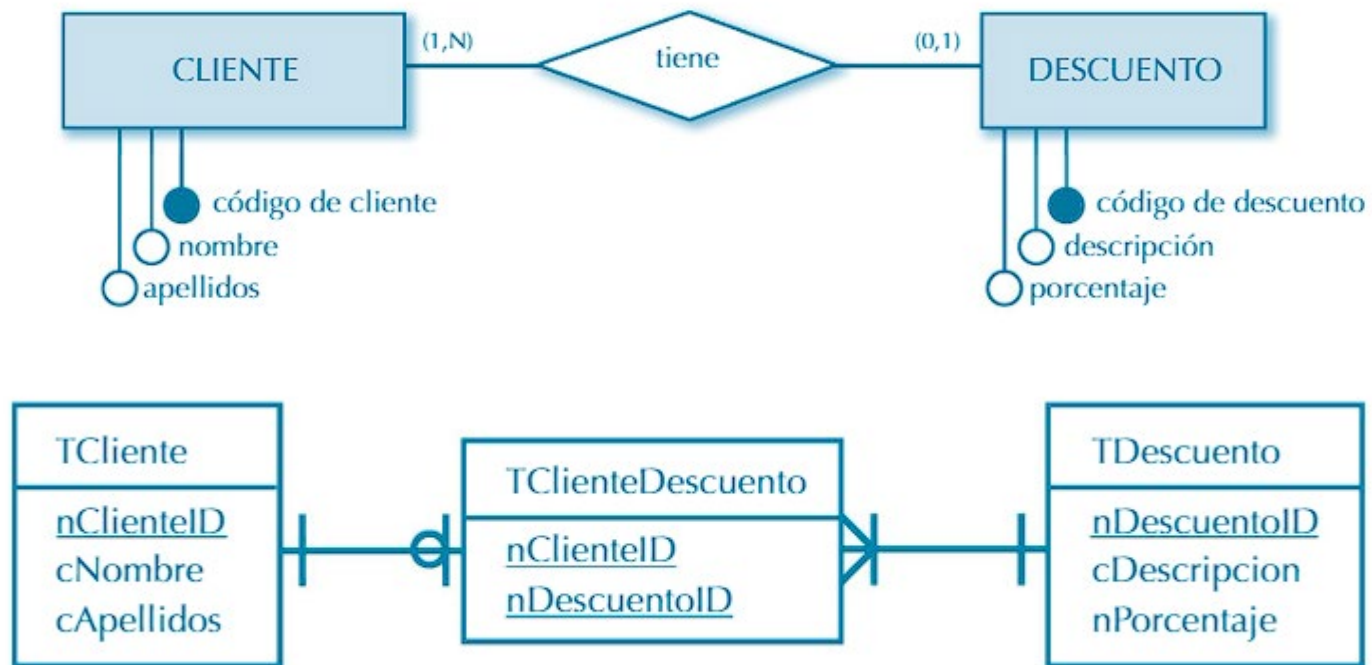


- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:N (Sin Atributos en la relación y la cardinalidad mínima del lado 1 es 0)
 - La relación se convierte en tabla y esta tabla tiene como atributos las PK'S de las tablas que relaciona y en esta nueva tabla esos campos también son PK.



- CLIENTE(nClienteID, cNombre, cApellidos)
- DESCUENTO(nDescuentoID, cDescripcion, nPorcentaje)
- CLIENTEDESCUENTO(nClienteID, nDescuentoID)

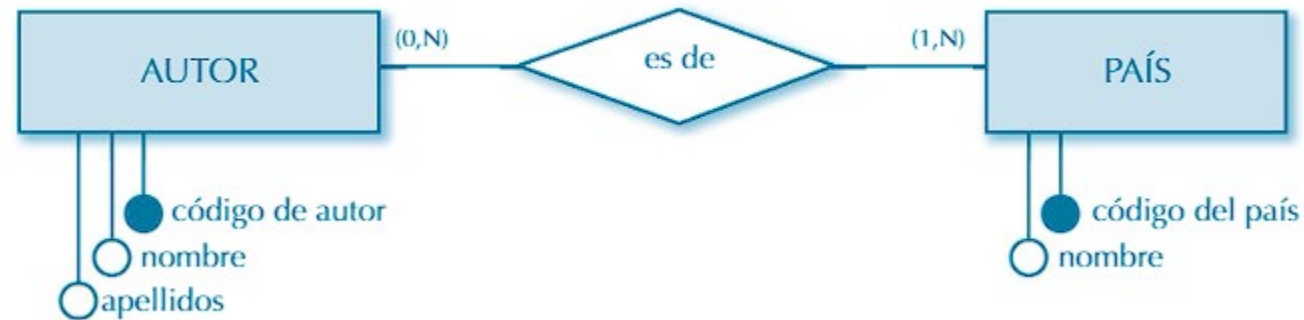
- Reglas de transformación del Diagrama E/R a un Modelo Relacional
 - Relaciones 1:N (Sin Atributos en la relación y la cardinalidad mínima del lado 1 es 0)**
 - La relación se convierte en tabla y esta tabla tiene como atributos las PK'S de las tablas que relaciona y en esta nueva tabla esos campos también son PK.



• Reglas de transformación del Diagrama E/R a un Modelo Relacional

• Relaciones N:M

- La relación se convierte en una tabla y tiene como atributos las PK'S de las tablas con las que se relaciona, siendo estos atributos PK también de la nueva tabla.

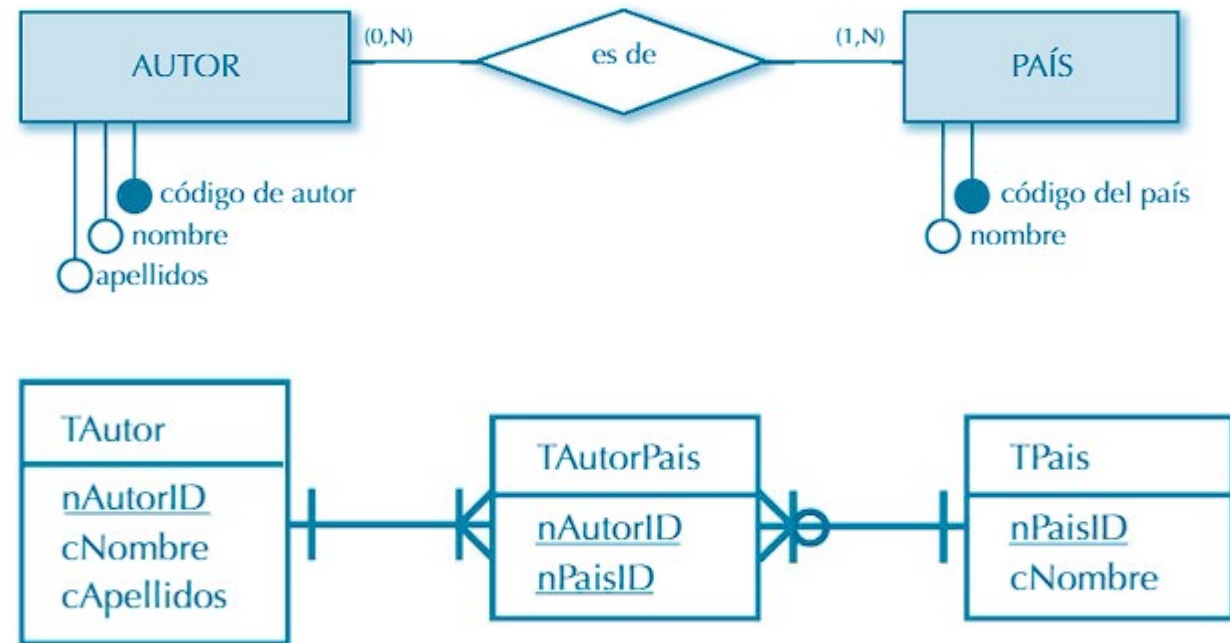


- **AUTOR** (nAutorID, cNombre, cApellidos)
- **PAIS**(nPaisID, nNombre)
- **AUTORPAIS**(nAutorID, nPaisID)

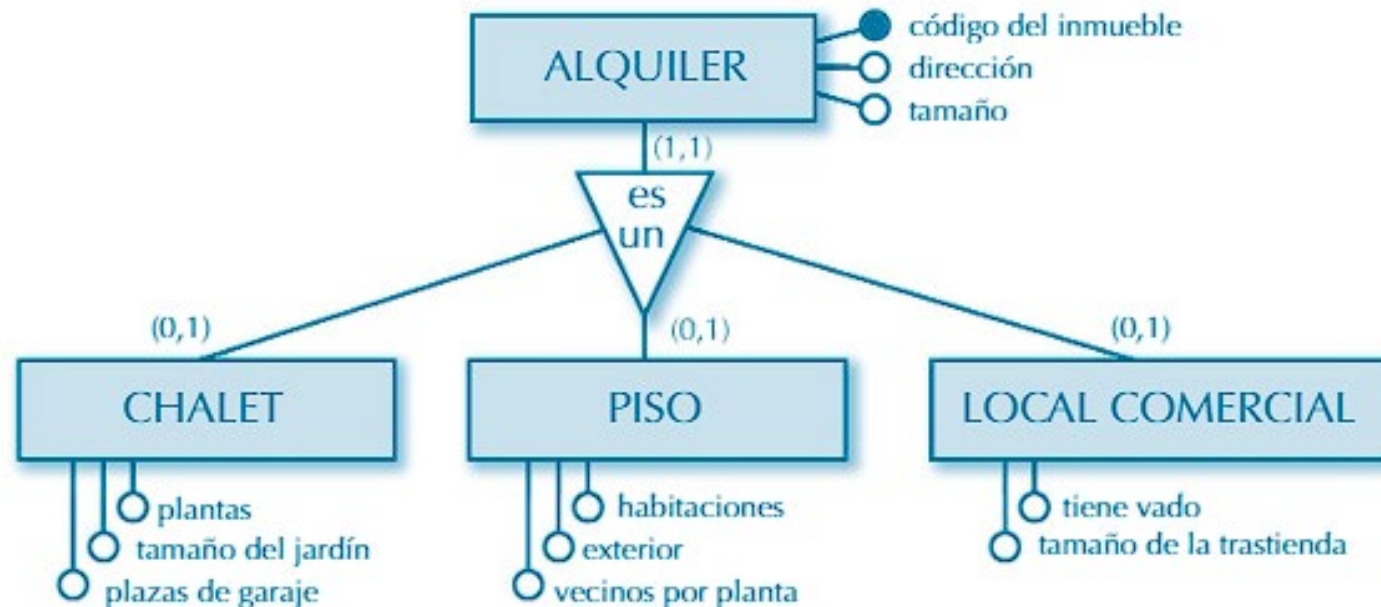
- Reglas de transformación del Diagrama E/R a un Modelo Relacional

- Relaciones N:M

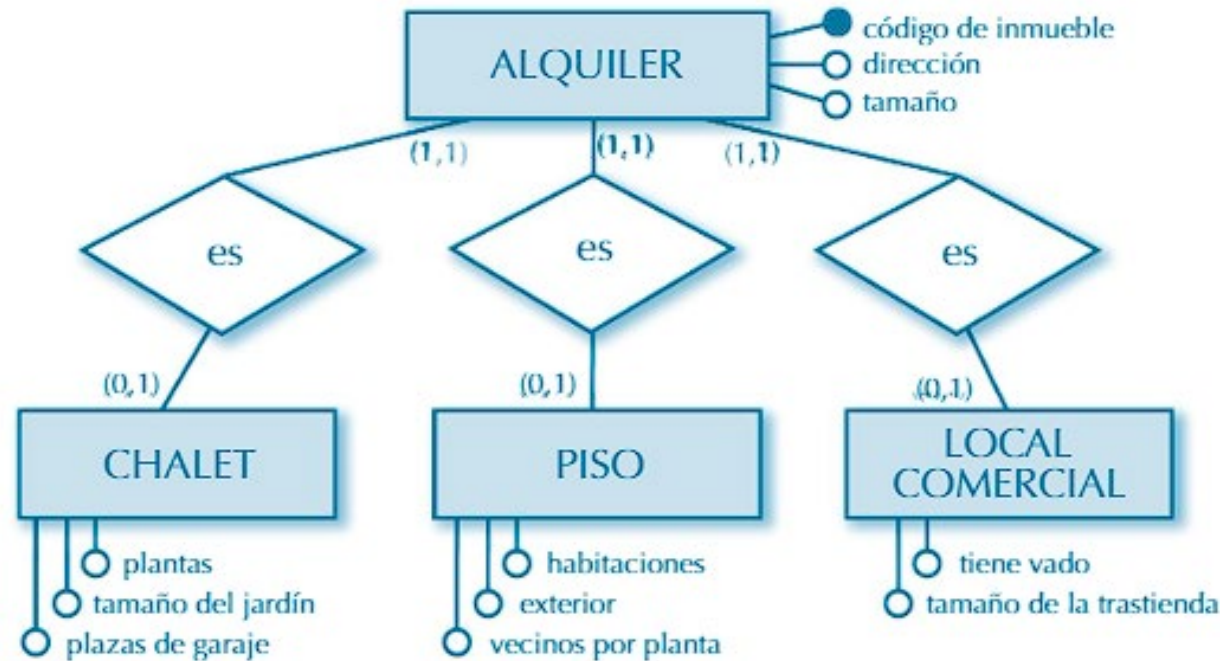
- La relación se convierte en una tabla y tiene como atributos las PK'S de las tablas con las que se relaciona, siendo estos atributos PK también de la nueva tabla. (Si la relación tiene atributos también se añaden a la nueva tabla)



- **Reglas de transformación del Diagrama E/R a un Modelo Relacional**
 - **Transformación de jerarquías:** Se convierten en relaciones normales con modalidades (0,1) y (1,1)



- Reglas de transformación del Diagrama E/R a un Modelo Relacional



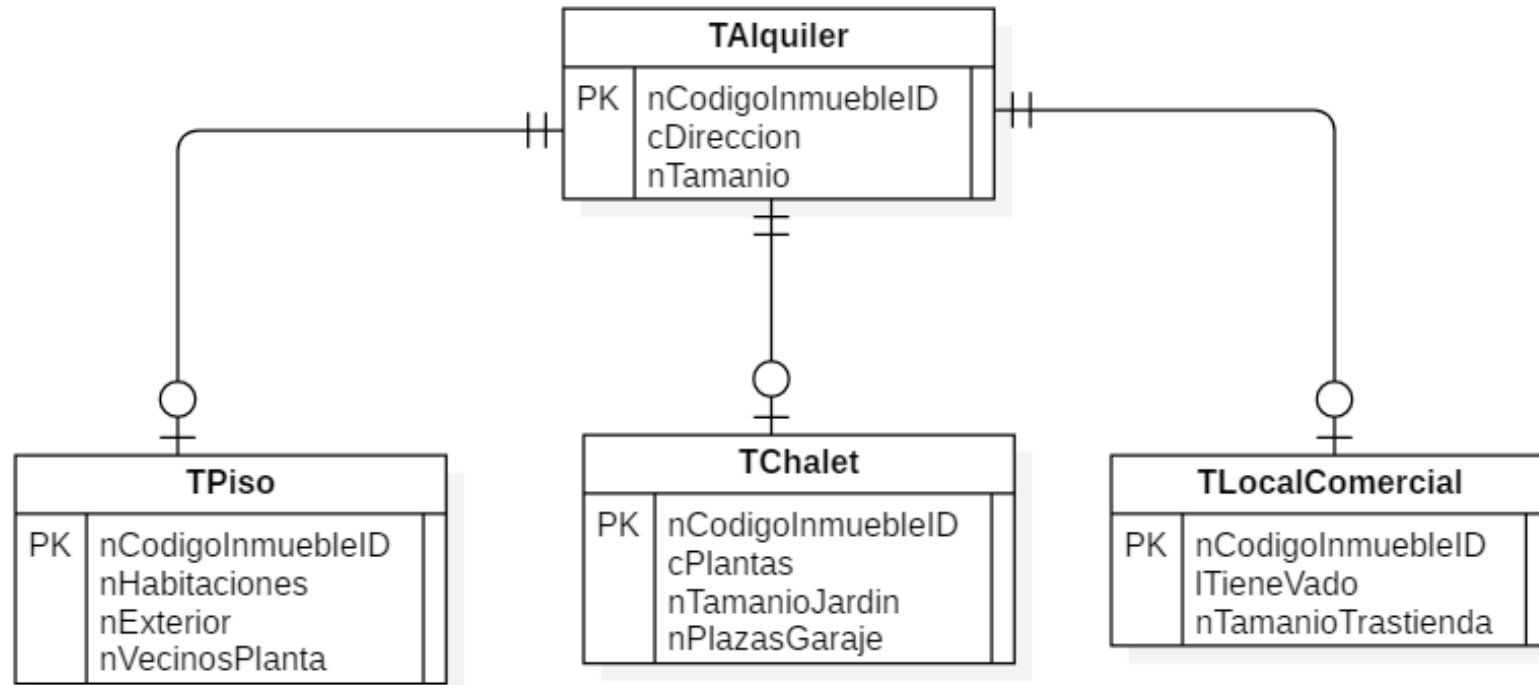
Alquiler(nCodigoInmuebleID, cDireccion, nTamanio)

Chalet(nCodigoInmuebleID, nHabitaciones, nExterior, nVecinosPlanta)

Piso(nCodigoInmuebleID, nPlantas, nTamanioJardin, nPlazasGaraje)

LocalComercial(nCodigoInmuebleID, lTieneVado, nTamanioTrastienda)

- Reglas de transformación del Diagrama E/R a un Modelo Relacional



GENERACIÓN BBDD A PARTIR DEL MODELO RELACIONAL



- Instrucciones para crear y definir tablas:

- **CREATE TABLE** nombreTabla (

nombreCampo1 **DataType** NOT NULL ... **COMMENT** '...'

nombreCampo2 **DataType** NULL **COMMENT** '...'

CONSTRAINT

CONSTRAINT

)

- **RESTRICCIONES (CONSTRAINT)**
- Para asegurar la integridad de los datos, en cada una de las tablas creadas en nuestra BBDD debemos establecer una serie de restricciones (CONSTRAINT). Algunas de estas restricciones son:
 - **PRIMARY KEY:** Cada tabla de nuestra BBDD debe tener una clave primaria que identifique de manera única a dicha tabla.
 - **CONSTRAINT nombreRestreccion PRIMARY KEY** (campo principal de la tabla)

- **RESTRICCIONES (CONSTRAINT)**

- **UNIQUE:** Establece que uno o varios campos de la nueva tabla van a tener un valor único. A diferencia de la Primary Key, los campos Unique si pueden ser NULL, pero este valor aparecerá como máximo en un registro, ya que su valor es único y no se puede repetir en ningún otro.
 - `CONSTRAINT nombreRestreccion UNIQUE(campo1, campo2....)`
- **FOREIGN KEY:** Campo (o campos) de la nueva tabla que sirve como enlace hacia otra tabla. La Foreign Key (clave ajena) de la nueva tabla será PK de la tabla con la que se relaciona.
 - `CONSTRAINT nombreRestreccion FOREIGN KEY (campo1, campo2....) REFERENCES tablaRelacion (campo principal de la tabla)`

Muchas gracias por vuestra atención

