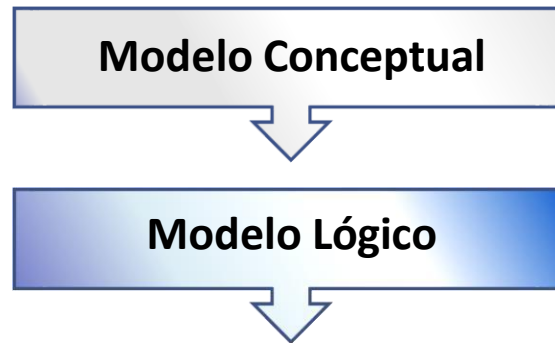


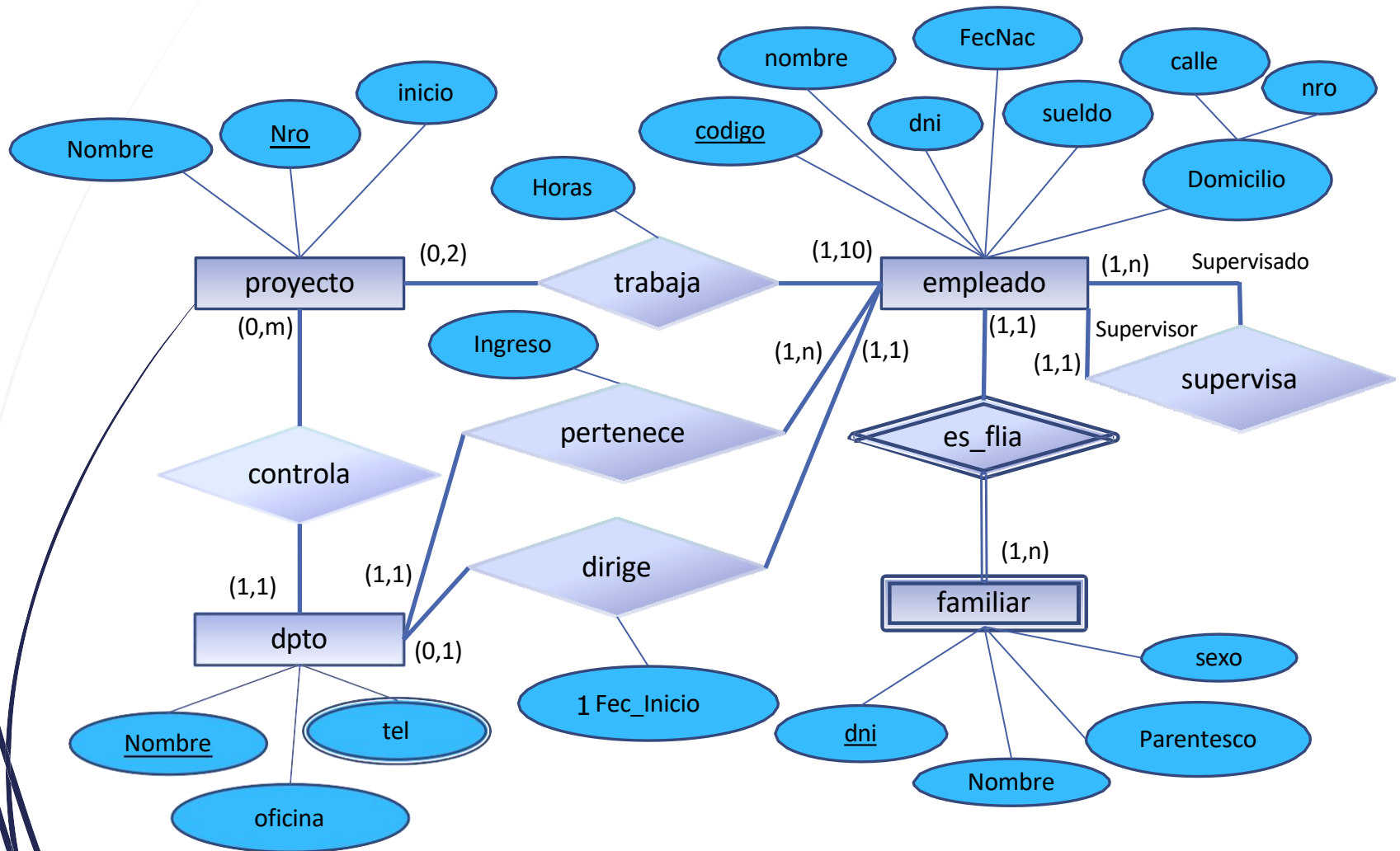
Del modelo conceptual al modelo lógico



Transformación de ER a MR

- Es posible derivar (obtener) un esquema de bases de datos relacional *a partir* de un esquema conceptual creado empleando el modelo ERE.
- Método de transformación utilizaremos **Algoritmo de Siete Pasos**, puesto que *una base de datos que se ajusta a un diagrama ER puede representarse por medio de una colección de tablas*.

Pasos para la Transformación

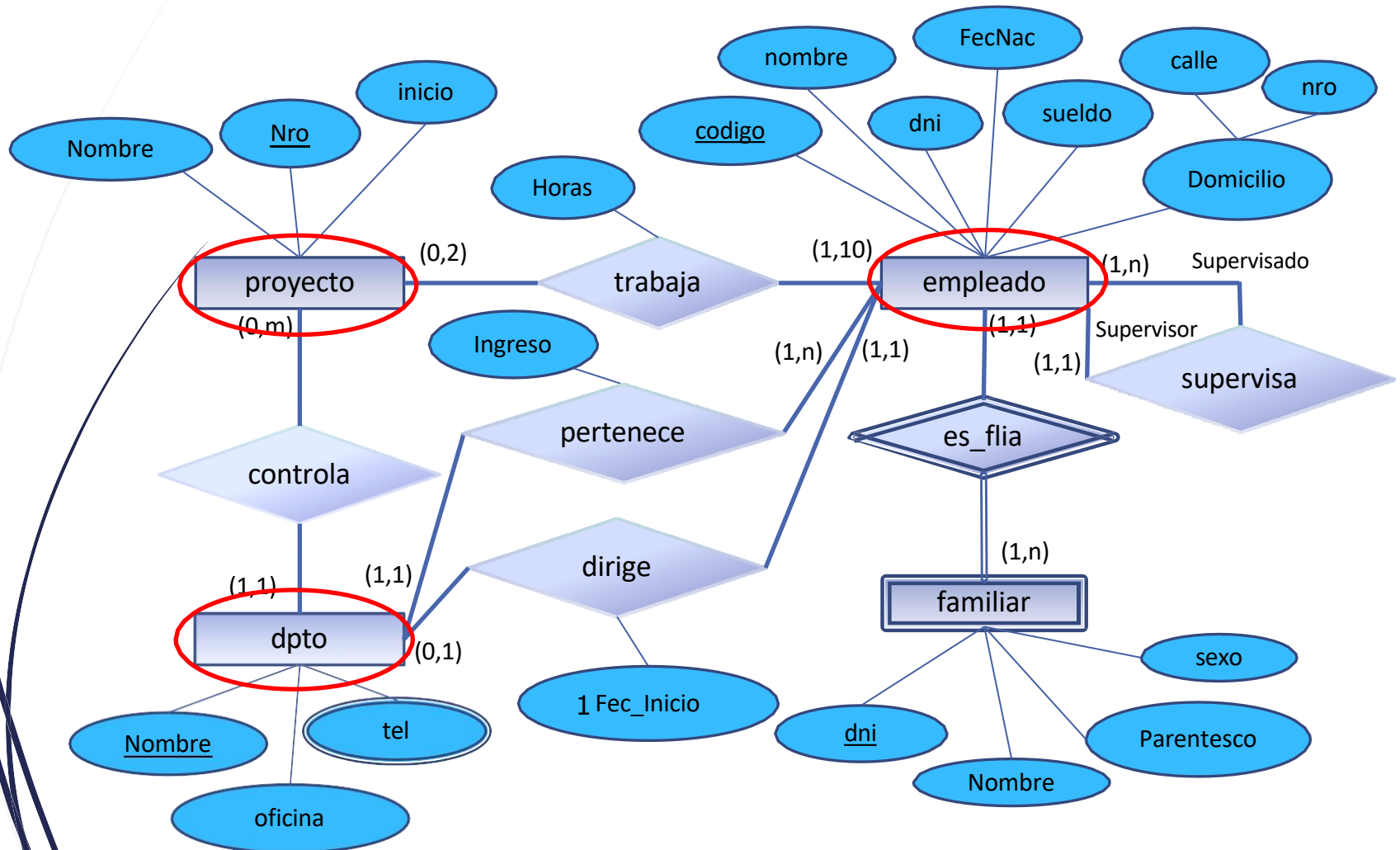


Pasos para la Transformación

1. Representación de conjuntos de Entidades **Fuertes**
2. Conversión de Entidades **Débiles**.
3. Representación de conjunto de relaciones o vínculos **1:1**
4. Representación de conjunto de relaciones o vínculos **1:N**
5. Representación de conjunto de relaciones o vínculos **M:N**
6. Conversión de atributos **multivaluados**.
7. Representación de conjunto de relaciones o vínculos n-arios R, con $n > 2$. (ADICIONAL)

Pasos para la Transformación

Paso 1: Representación de conjuntos de Entidades Fuertes



Pasos para la Transformación

Paso 1:

- Por cada entidad E del esquema E-R, se crea una relación R que contenga todos los atributos simples de E.
- Se incluyen sólo los atributos simples de un atributo compuesto.
- Luego se elige uno de los atributos como clave primaria.
- En el ejemplo los tipos de entidades a representar en este primer paso son: **empleado**, **dpto** y **proyecto**.

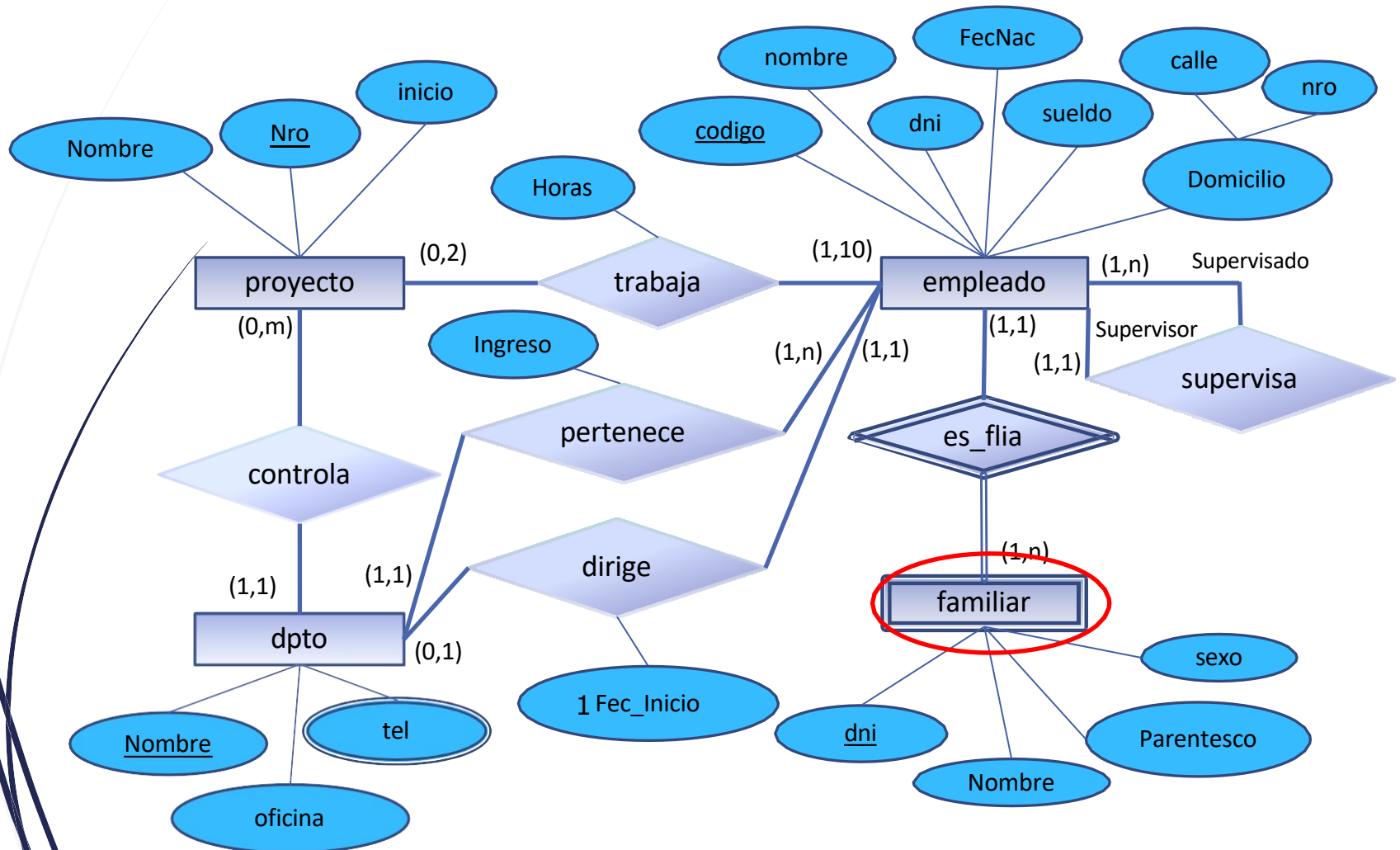
empleado (codigo, dni, nombre, fecNac, sueldo, calle, nro)

proyecto (Nro, nombre, inicio)

dpto (nombre, oficina)

Pasos para la Transformación

Paso 2: Conversión de Entidades Débiles



Pasos para la Transformación

Paso 2:

- Por cada entidad débil D del esquema ER con tipo de entidades propietarias E, se crea una relación o TABLA R y se incluyen todos los atributos simples (o componentes simples de los atributos compuestos) de D como atributos de R.
- Además se incluyen como atributos de clave externa de R los atributos de clave primaria de la relación o relaciones que corresponden al tipo de entidades propietarias.

Pasos para la Transformación

Paso 2:

La clave primaria de R es la combinación de las claves primarias de las propietarias y la clave parcial de D, si existe.

En el ejemplo se crea entonces la relación Dependiente que corresponde al tipo de entidades débil Dependiente. La relación o tipo de vínculo Dependiente_De no se transformará en los pasos correspondientes a conversión de tipos de vínculos puesto que es una ***relación mandatoria***.

empleado (codigo, dni, nombre, fecNac, sueldo, calle, nro)

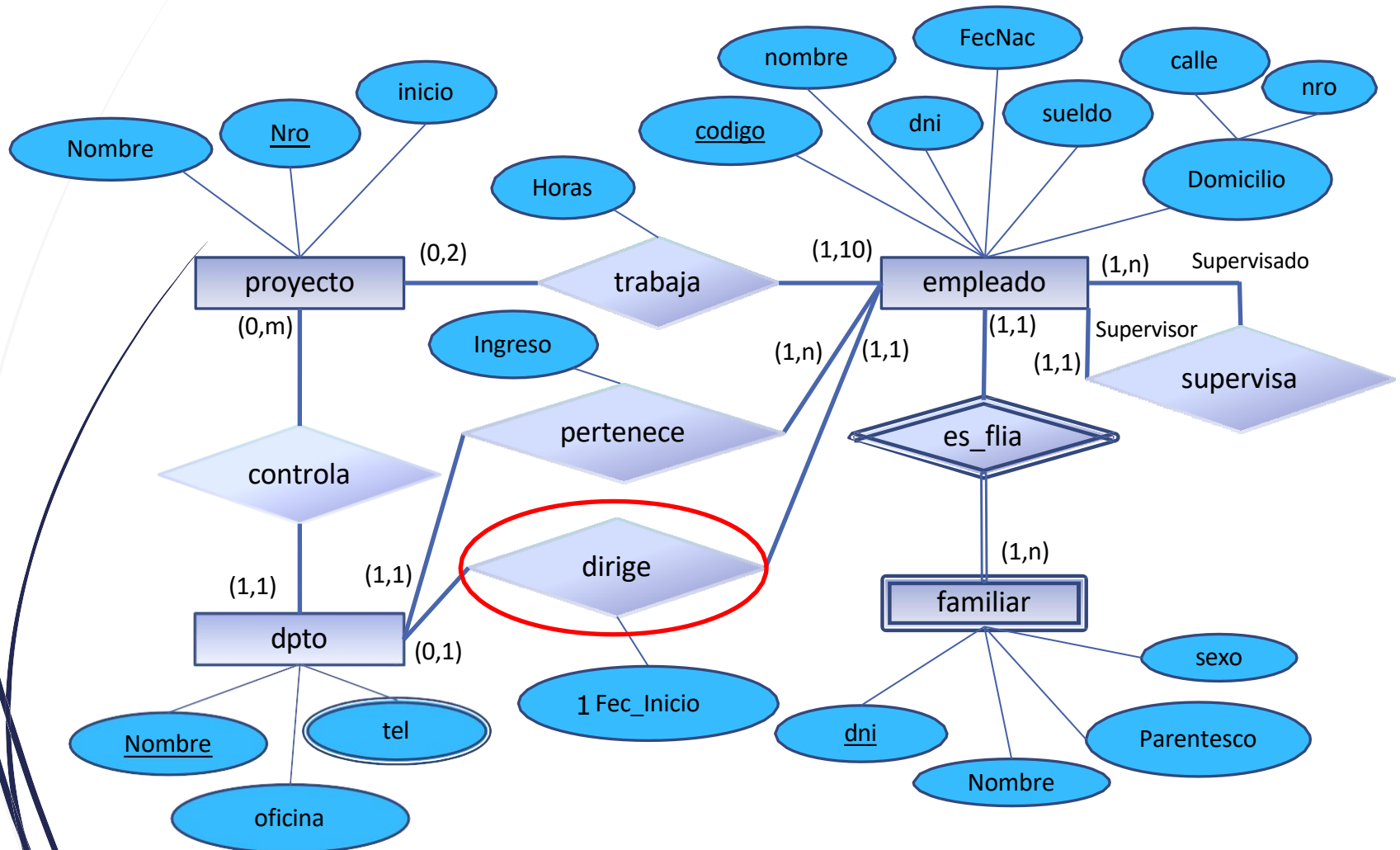
proyecto (Nro, nombre, inicio)

dpto (nombre, oficina)

familiar (codigoE, dni, nombre, sexo parentesco)

Pasos para la Transformación

Paso 3: Representación de conjunto de relaciones o vínculos 1:1



Pasos para la Transformación

Paso 3:

- Por cada relación binaria 1:1 **R** del esquema E-R, se identifican las relaciones **S** y **T** que corresponden a las entidades que participan en **R**.
- Se escoge una de las relaciones (supongamos **S**) y se incluye como clave externa en **S** la clave primaria de **T**.
- Es mejor elegir una entidad con participación Total en **R** en el papel de **S**. Se incluyen todos los atributos simples (o componentes simples de atributos compuestos) de la relación 1:1 **R** como atributos de **S**.

Pasos para la Transformación

Paso 3:

En el ejemplo la relación 1:1 **dirige** de la figura será el que se transformará, eligiendo **dpto** para desempeñar el papel de S, debido a que su participación en Dirige es TOTAL (todo dpto tiene un “gerente”). Se incluirá la clave primaria de la relación **empleado** como clave externa de la relación **dpto** y se cambia el nombre a codigoE. También se incluye el atributo simple fec_Inicio de **dirige** en la relación **dpto** y cambiamos su nombre a fec_InicioE.

empleado (codigo, dni, nombre, fecNac, sueldo, calle, nro)

proyecto (Nro, nombre, inicio)

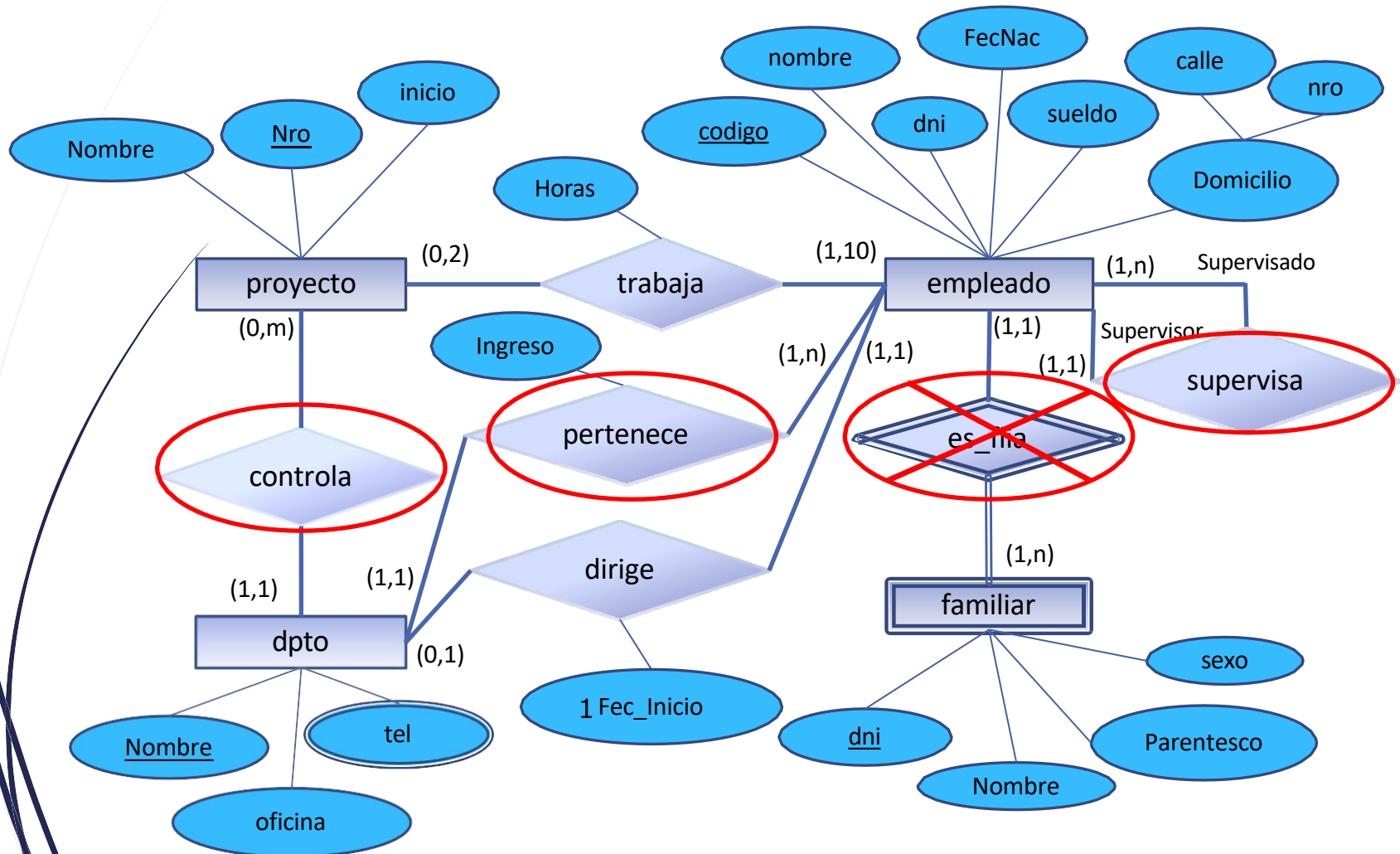
dpto (nombre, oficina, codigoE, fec_InicioE)

familiar (codigoE, dni, nombre, sexo parentesco)

Campos
agregados

Pasos para la Transformación

Paso 4: Representación de conjunto de relaciones o vínculos 1:N



Pasos para la Transformación

Paso 4:

- Por cada relación (no débil) de 1:N R, se identifica la relación S que representa la entidad participante de lado :N de la relación.
- Se incluye como clave externa en S la clave primaria de la relación T que representa a la otra entidad que participa en R; *(la razón es que cada ejemplar de entidad del lado N está relacionado con un máximo de un ejemplar de entidad del lado 1)*:
- **Se incluyen todos los atributos simples (o componentes simples de los atributos compuestos) de la relación 1:N como atributos de S.**

Pasos para la Transformación

Paso 4: En este paso se deben transformar las relaciones **pertenece**, **controla** Y **supervisa**.

En el caso de **pertenece** se incluye la clave primaria de **dpto (nombre)** como clave ajena en **empleado**. En el caso de la relación **controla** se agrega el atributo de clave externa de **dpto** a proyecto.

En el caso de **supervisa**, se incluye la clave primaria de **empleado** como clave externa en la relación **empleado**

empleado (codigo, dni, nombre, fecNac, sueldo, calle, nro, nomD, ingreso, codE)

proyecto (Nro, nombre, inicio, nomD)

dpto (nombre, oficina, codigoE, fec_InicioE)

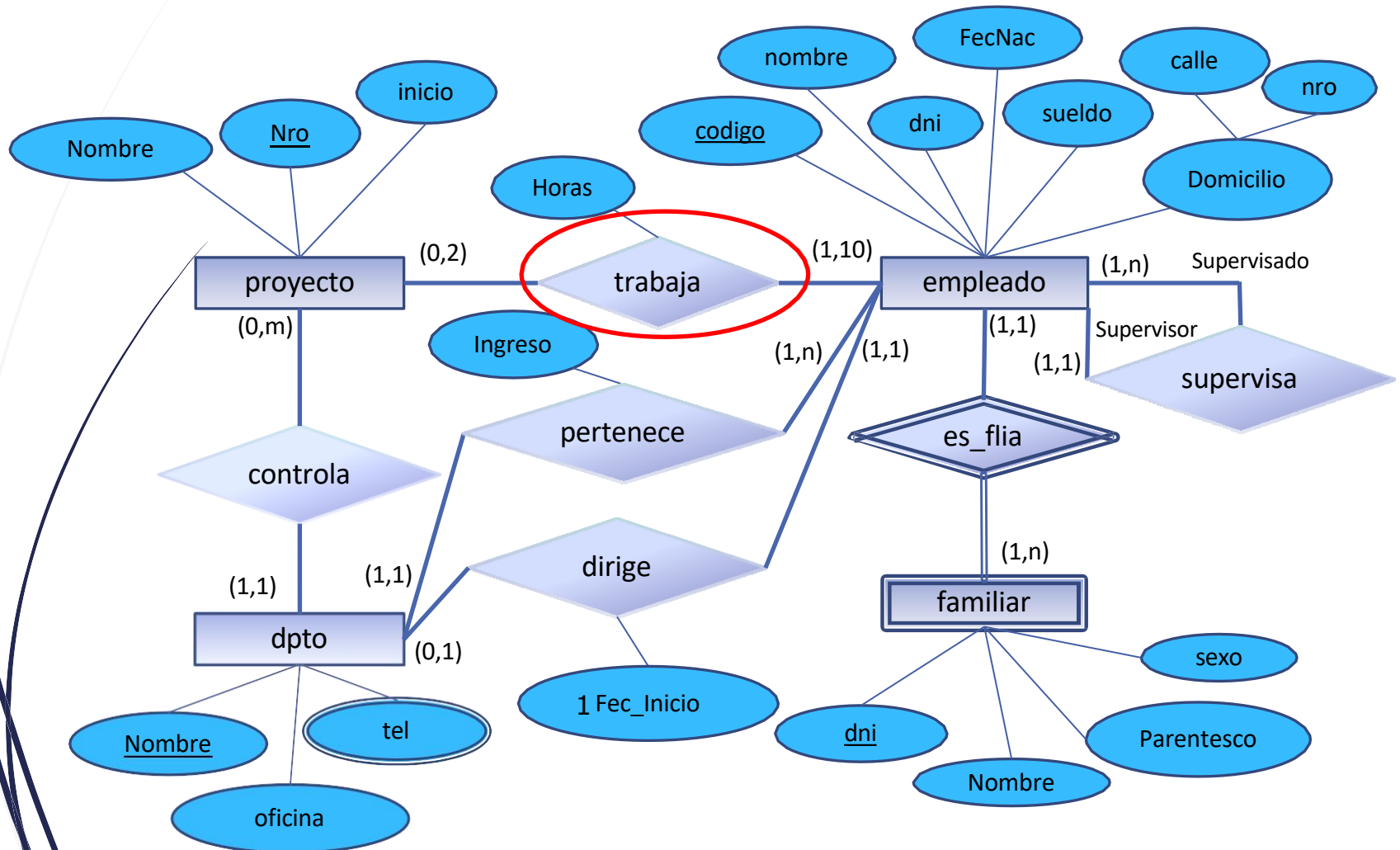
familiar (codigoE, dni, nombre, sexo parentesco)

**Campos
agregados**



Pasos para la Transformación

Paso 5: Representación de conjunto de relaciones o vínculos **M:N**



Pasos para la Transformación

Paso 5:

- Por cada relación binaria M:N , **se crea una nueva TABLA S** para representar R.
- Se incluyen como atributos de clave externa en S las claves primarias de las relaciones que representan las entidades participantes; su combinación constituirá la clave primaria de S.
- También se incluyen todos los atributos simples (o componentes simples de atributos compuestos) del tipo de vínculos M:N como atributos de S.
- No se puede representar una relación M:N con un solo atributo de clave externa en una de las relaciones participantes (como se hizo en el caso de las relaciones 1:1 y 1:N) debido a la razón de cardinalidad M:N.

Pasos para la Transformación

Paso 5:

En el ejemplo, se establece la transformación de la relación **trabaja**, agregando las claves primarias de las relaciones **proyecto** y **empleado** como claves ajenas en **trabaja**. También se incluyen todos los atributos que tiene la relación trabaja, en este caso, horas. La clave primaria de la relación trabaja es la combinación de los atributos de clave externa NSSE, NúmP.

empleado (codigo, dni, nombre, fecNac, sueldo, calle, nro, nomD, ingreso, codE)

proyecto (Nro, nombre, inicio, nomD)

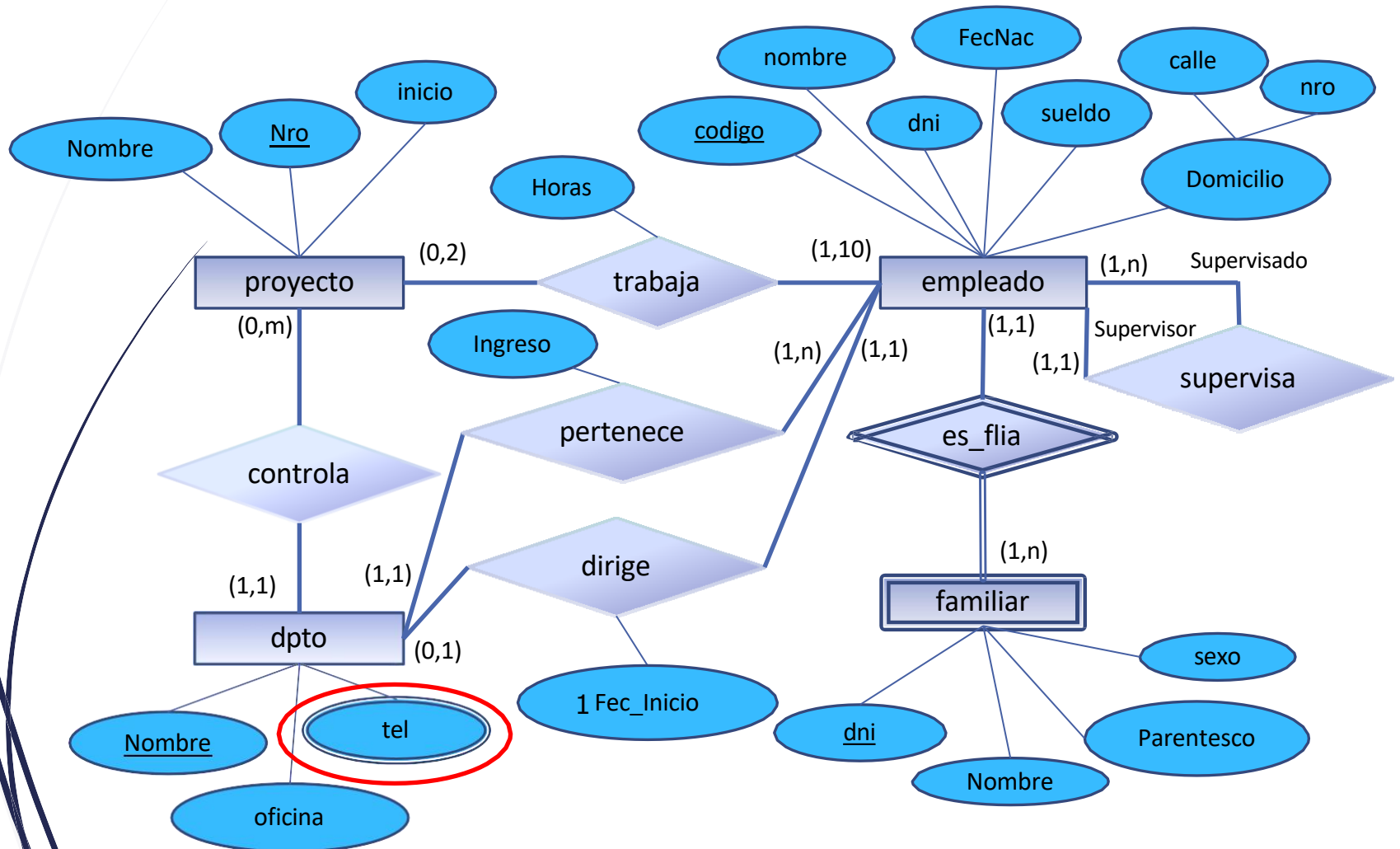
dpto (nombre, oficina, codigoE, fec_InicioE)

familiar (codigoE, dni, nombre, sexo parentesco)

trabaja (códigoE, nombreD, horas) ← Nueva Tabla

Pasos para la Transformación

Paso 6: Conversión de atributos **multivaluados**.



Pasos para la Transformación

Paso 6:

- Por cada atributo multivaluado A se crea una relación nueva R que contiene un atributo correspondiente a A más el atributo de clave primaria K (como clave externa de R) de la relación que representa la entidad o relación que contiene a A como atributo. La clave primaria de R es la combinación de A y K.
- Si el atributo multivaluado es compuesto, se incluyen sus componentes simples.

Pasos para la Transformación

Paso 6:

En el ejemplo, se crea la relación `tel_dpto`. El atributo `telD` representa el atributo multivaluado `tel` de `dpto`, en tanto que `nombreD` (como clave ajena) representa la clave primaria de la relación `dpto`.

La clave primaria de `tel_dpto` es la combinación (`nombreD`, `telD`). Habrá una tupla en `tel_dpto` por cada número de teléfono de `dpto`.

empleado (codigo, dni, nombre, fecNac, sueldo, calle, nro, nomD, inicio codE)

proyecto (Nro, nombre, inicio, nomD)

dpto (nombre, oficina, codigoE, fec_InicioE)

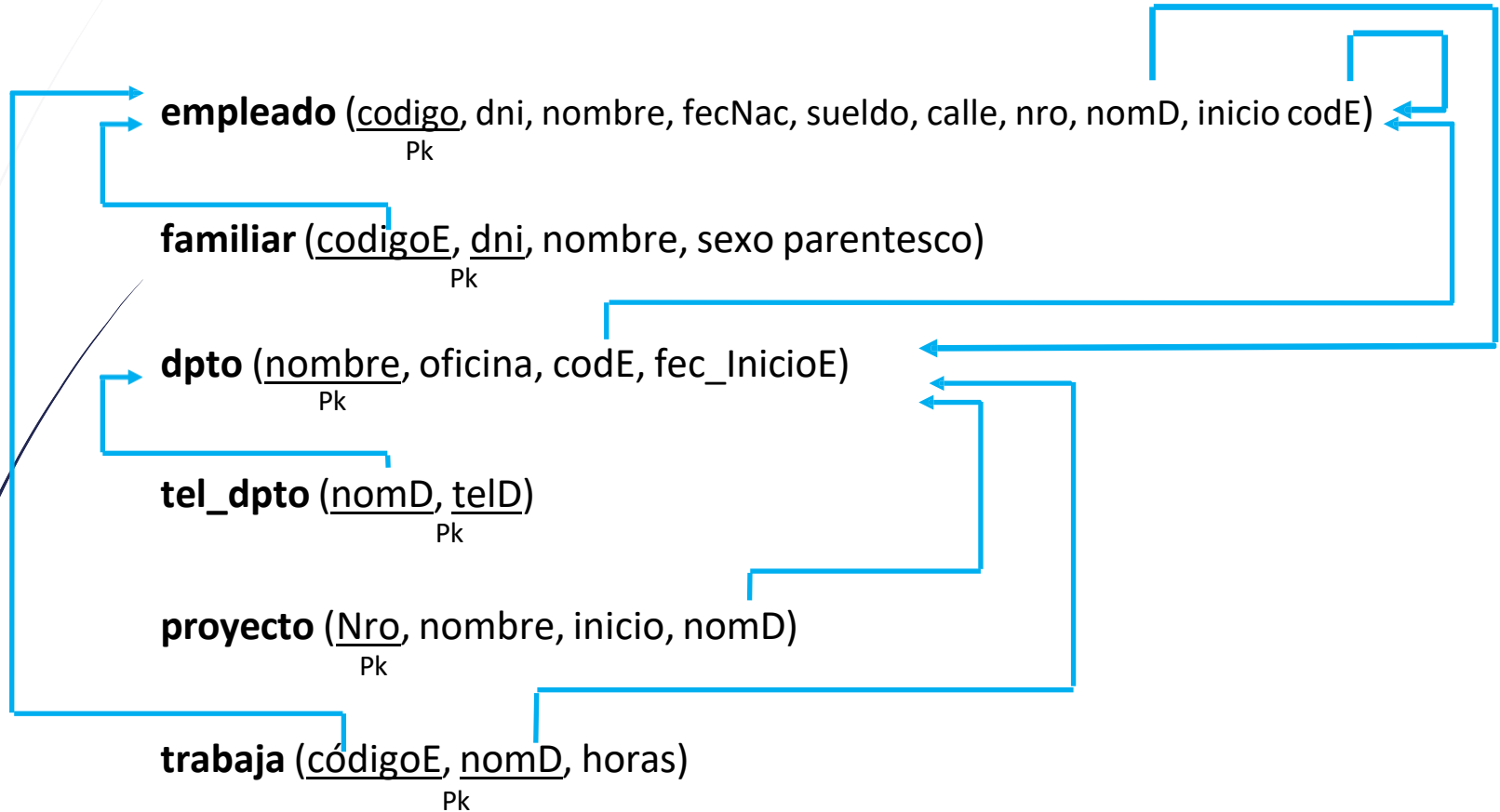
familiar (codigoE, dni, nombre, sexo parentesco)

trabaja (códigoE, nombreD, horas)

tel_dpto (nombreD, telD)

← Nueva Tabla

Pasos para la Transformación



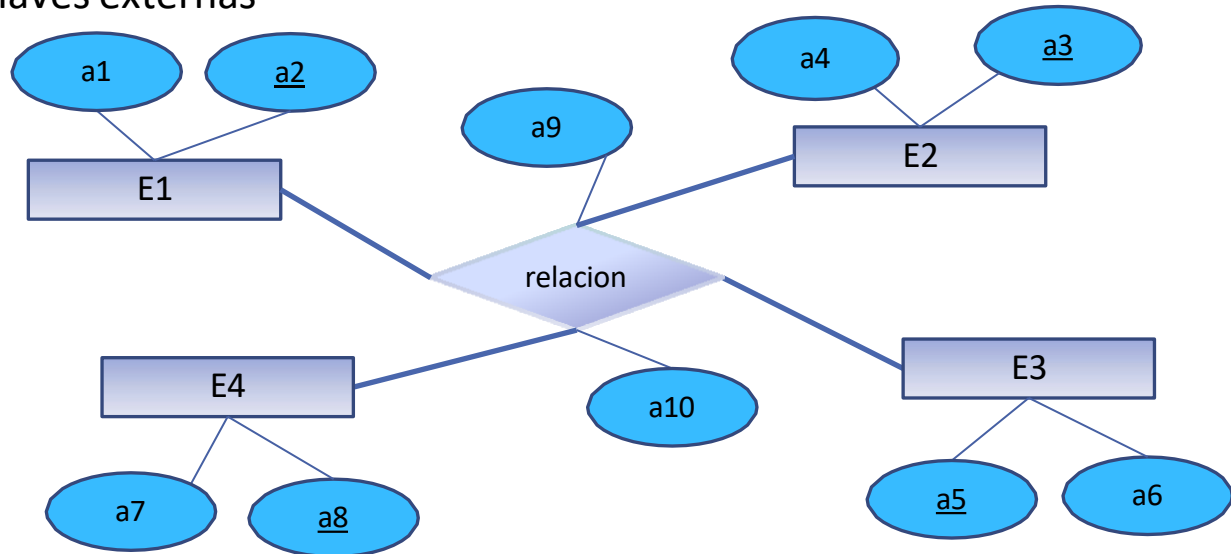
Pasos para la Transformación

Paso 7:

- Por cada relación n -aria R , $n > 2$, se crea una nueva relación S que represente a R . Se incluyen como atributos de clave externa en S las claves primarias de las relaciones que representan las entidades participantes. También se incluyen los atributos simples de la relación n -aria como atributos de S .
- **La clave primaria de S casi siempre es una combinación de todas las claves externas que hacen referencia a las relaciones que representan las entidades participantes.**
- No obstante, si la restricción de participación (min, máx) de uno de los tipos de entidades E que participan en R tiene máx=1, la clave primaria de S podrá ser el único atributo de clave externa que haga referencia a la relación E' que corresponde a E ; la razón es que cada una de las entidades e de E participará, en cuando más, un ejemplar de vínculo R , y por tanto, podrá identificar de manera única ese ejemplar.

Pasos para la Transformación

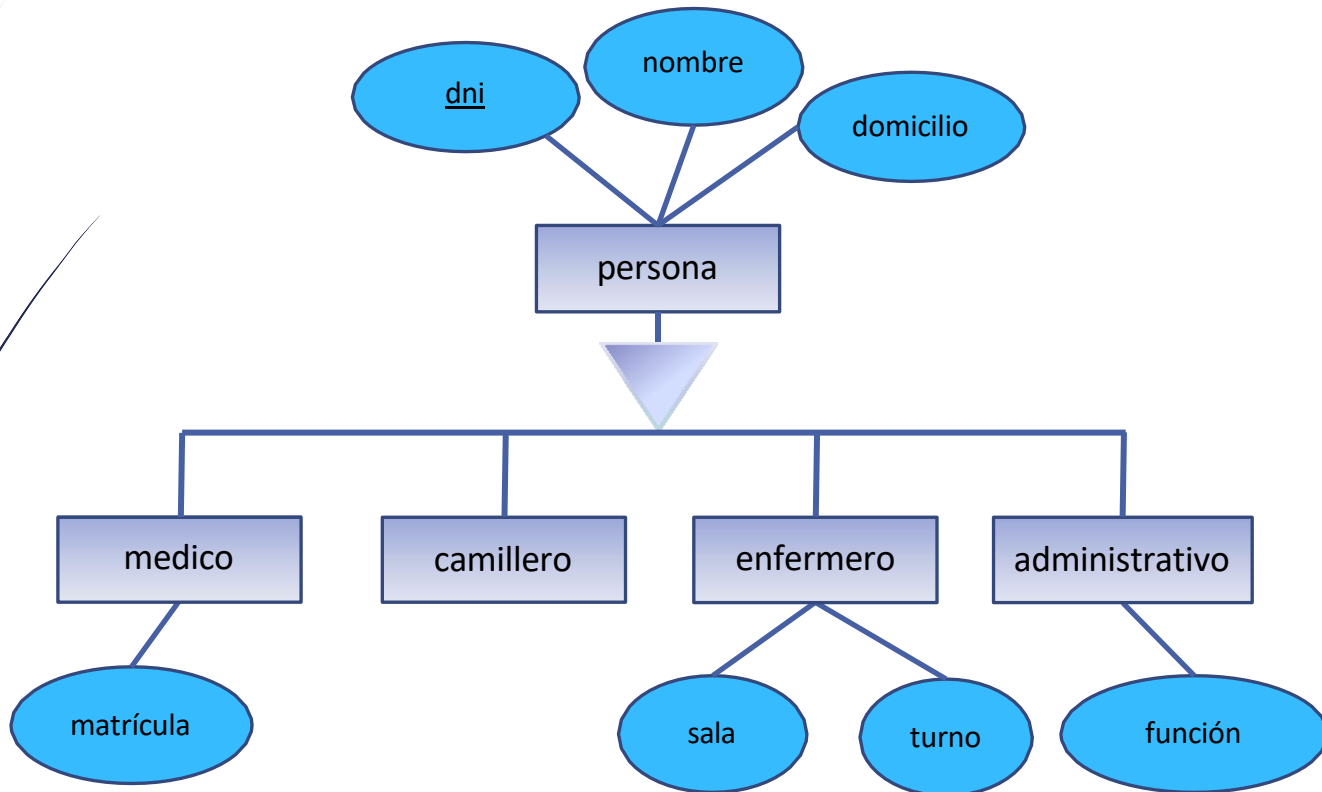
Paso 7: Las relaciones ternarias, cuaternarias y n-arias se transforman en una tabla que contiene los atributos de la relación mas las claves primarias de las entidades . La clave primaria esta formada por las claves externas



relacion (a2, a3, a5, a8, a9, a10)

Pasos para la Transformación

Generalización



Transformar Generalizaciones

1. Pasar sólo el conjunto de entidades superior.

Todos los atributos de la entidad superior, más cada uno de los atributos de las entidades inferiores, más un atributo distintivo (que indique el tipo de personal).

persona (dni, nombre, domicilio, matricula, función, sala, turno)

Es la menos recomendada, pues promueve atributos nuevos y dificulta la búsqueda

Cuando la instanciación sea de un camillero, los últimos 4 atributos serán **nulos!!!**

Transformar Generalizaciones

2. **Pasar las entidades de todos los niveles a tablas y en las entidades de nivel inferior, se pone la clave del nivel superior.**

persona (dni, nombre, domicilio)

médico (dni, matricula)

enfermero (dni, sala, turno)

camillero (dni)

administrativo (dni, función)

3. **Representar sólo los niveles inferiores con los atributos propios, más lo de la entidad superior.**

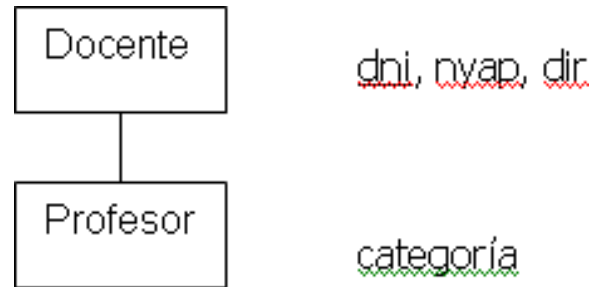
médico (dni, nombre, domicilio, matricula)

enfermero (dni, nombre, domicilio, sala, turno)

camillero (dni, nombre, domicilio)

administrativo (dni, nombre, domicilio, función)

Transformar Especializaciones



También existen dos formas de conversión, que ***coinciden con las dos primeras de la generalización.***

- La tercera no es viable porque pueden quedar filas en la entidad superior que no pertenezcan a ninguna entidad inferior y se perdería información.

1) *Docente* (dni, nyap, dir, categoría, tipo)

2) **Docente** (dni, nyap, dir) –

Profesor (dni, categoría)

Evolución del Modelo

Cuando el MR triunfó comercialmente, muchos fabricantes que tenían productos “antiguos” **no relacionales** optaron por retocarlos o “**camuflarlos**” añadiéndoles la etiqueta relacional.

- Esto supuso una confusión que **Codd intentó arreglar publicando sus 12+1 reglas**, que indican las características que debe tener un DBMS para ser auténticamente relacional.

Regla 0:

Un DBMS relacional debe emplearse para gestionar en la BD exclusivamente sus facilidades relacionales. → De esta regla genérica se derivan 12 reglas detalladas.

Evolución del Modelo

1. Regla de información: Toda la información en la Base de datos es representada en una y solo una forma: valores en columnas de filas de tablas.
2. Regla de acceso garantizado: Cada valor escalar individual puede ser direccionado indicando los nombres de la tabla, columna y valor de la clave primaria de la fila correspondiente.
3. Tratamiento sistemático de valores nulos: El DBMS debe soportar la representación y manipulación de información desconocida y/o no aplicable.
4. Catálogo en línea (diccionario de datos) basado en el modelo relacional.

Evolución del Modelo

5. Sublenguaje de datos completo: El DBMS debe soportar al menos un lenguaje relacional:
 - a) con sintaxis lineal.
 - b) que pueda ser usado interactivamente o en programas (embebido).
 - c) con soporte para operaciones de:
 - definición de datos (p.e. declaración de vistas).
 - manipulación de datos (p.e. recuperación y modificación de tuplas).
 - restricciones de seguridad e integridad.
 - gestión de transacciones.
6. Actualización de vistas: todas las vistas teóricamente actualizables deben poder serlo en la práctica.

Evolución del Modelo

7. Inserción, modificación y borrado de tuplas de alto nivel: todas las operaciones de manipulación de datos deben operar sobre conjuntos de filas (lenguaje de especificación en vez de navegacional).

8. Independencia física de los datos: cambios en los métodos de acceso físico o la forma de almacenamiento no deben afectar al acceso lógico a los datos.

9. Independencia lógica de los datos: los programas de aplicación no deben ser afectados por cambios en las tablas que preservan la integridad.

Evolución del Modelo

10. Independencia de la integridad: Las restricciones de integridad deben estar separadas de los programas, almacenadas en el catálogo de la BD para ser editadas mediante un sub lenguaje de datos.

11. Independencia de la distribución: Las aplicaciones no deben verse afectadas al distribuir (dividir entre varias máquinas), o al cambiar la distrib. ya existente de la BD.

12. Regla de no subversión: Si el sistema posee un interface de bajo nivel (p.e. mediante llamadas en C), éste no puede subvertir el sistema pudiendo evitar restricciones de seguridad o integridad.



Muchas gracias