



DeepL

Suscríbete a DeepL Pro para poder traducir archivos de mayor tamaño.
Más información disponible en www.DeepL.com/pro.



I.E.S. Rodrigo Caro

Departamento de Informática

www.informaticarodrigocaros.es

Informática
Rodrigo
Caro

Unidad 3: Modelos relacionales

ASIR

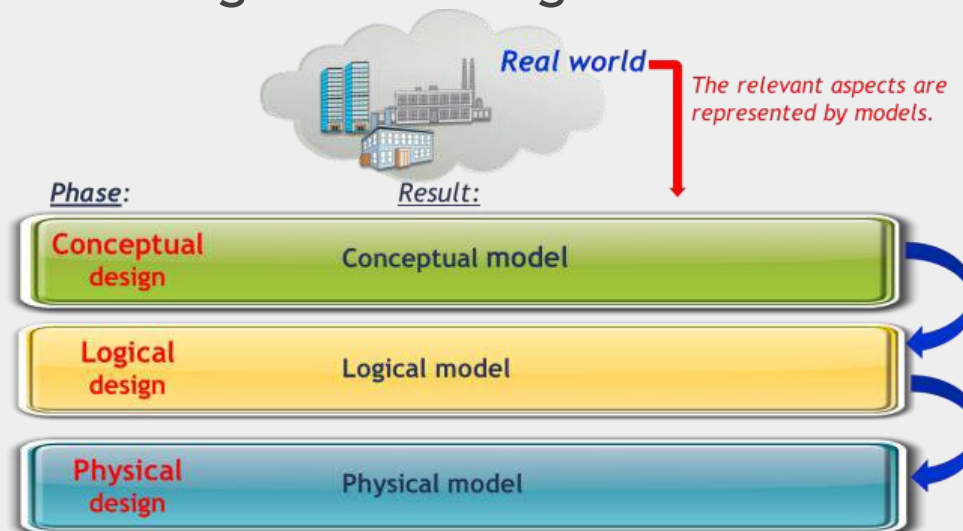
Índice

1. Modelo relacional: Definición
2. Elementos
3. Restricciones
4. Entidades modelo
5. Relación de modelos: N:M
6. Relación de modelos: 1:M
7. Relaciones con el modelo: 1:1
8. Relaciones reflexivas
- 9.
10. Normalización: 1FN
11. Normalización: 2FN
12. Normalización: 3FN

Modelo relacional: Definición

El modelo relacional es un **modelo lógico** para diseñar bases de datos. Actualmente, el modelo relacional es el más extendido ya que ha ido sustituyendo a otros modelos como el de red o el jerárquico.

Recuerda que estamos trabajando a nivel lógico. El almacenamiento físico de los datos es independiente de la forma en que están organizados lógicamente..



Modelo relacional: Definición

Las principales razones de su éxito son:

- La información se representa y manipula de forma sencilla. Básicamente, se trata de **tablas de dos dimensiones interrelacionadas**: formadas por **filas** (registros o **tuplas**) y **columnas** (**atributos** o campos).
- Se basan en el **álgebra relacional**, que es un modelo matemático con fundamentos sólidos.

Principios básicos del modelo relacional:

- Estructura de datos relacional
- Normas de integridad.

Modelo relacional: Concepto de relación

En el modelo relacional, la relación es la definición de la estructura de la tabla, es decir, su nombre y la lista de atributos que la componen.

La clave primaria se utiliza para distinguir un registro de otro. Puede haber más combinaciones de atributos en una relación que puedan identificar de forma exclusiva una fila (se denominarán "claves candidatas"), pero de ellas sólo **se elegirá una** para utilizarla como clave **primaria**.

Modelo relacional: Elementos

Relación (tabla): Representan las entidades cuya información se va a almacenar en la BD. Se compone de:

- **Filas (registros o tuplas):** Corresponden a cada ocurrencia o valor de la entidad.
- **Columnas (Atributos o campos):** Corresponden a las propiedades de la entidad.

Condiciones:

- Cada relación tiene un nombre y este nombre es diferente del nombre de todas las demás relaciones de la misma BD.
- No hay dos atributos en la misma relación que tengan el mismo nombre.
- El orden de los atributos (columnas) no importa.
- El orden de las tuplas (filas) no importa.
- Cada tupla es distinta de las demás: no hay duplicados tuplas (al menos se diferenciarán en la clave primaria).

Modelo relacional: Elementos

- **Clave candidata:** atributo que identifica de forma única una tupla. Cualquiera de las claves candidatas puede ser elegida como clave primaria.
- **(PK) Clave primaria:** Clave candidata que elegimos como identificador de la tupla. Una clave primaria no puede asumir un valor nulo (**integridad de la entidad**).
- **Clave alternativa:** Cualquier clave candidata que no sea una clave primaria (aquellas que no hemos elegido como clave primaria).
- **(FK) Clave ajena:** Atributo o conjunto de atributos que forman la clave primaria de otra relación. En otras palabras, los valores presentes en la clave foránea deben corresponder a los valores presentes en la clave primaria correspondiente (**Integridad Referencial**).

Modelo relacional: Elementos

- **Dominio de un atributo:** Conjunto de valores que puede asumir ese atributo.
- **Grado:** El número total de atributos que en la relación se llama el grado de la relación.
- **Cardinalidad:** Número total de filas presentes en la Tabla.

Modelo relacional: Restricciones

Hay muchos tipos de restricciones de integridad en los SGBD. Las restricciones en el sistema de gestión de bases de datos relacionales se dividen principalmente en tres categorías principales son:

- Restricciones de dominio
- Restricciones clave
- Restricciones de integridad referencial

Modelo relacional: Restricciones

Restricciones de dominio

Las restricciones de dominio pueden ser violadas si un valor de atributo no aparece en el dominio correspondiente o no es del tipo de datos apropiado. Por ejemplo:

```
Crear DOMAIN CustomerName  
CHECK (valor no NULL)
```

El ejemplo mostrado demuestra la creación de una restricción de dominio tal que CustomerName no es NULL

Modelo relacional: Restricciones

Restricciones clave

Un atributo que puede identificar de forma exclusiva una tupla en una relación se denomina clave de la tabla. El valor del atributo para diferentes tuplas en la relación tiene que ser único.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

CustomerID es un atributo clave de la tabla de clientes.

Modelo relacional: Restricciones

Restricciones de integridad referencial

Una clave externa es un atributo importante de una relación al que se debe hacer referencia en otras relaciones. El estado de la restricción de integridad referencial ocurre cuando la relación se refiere a un atributo clave de una relación diferente o de la misma. Sin embargo, ese elemento clave debe existir en la tabla.

De E / R al modelo relacional: Entidades

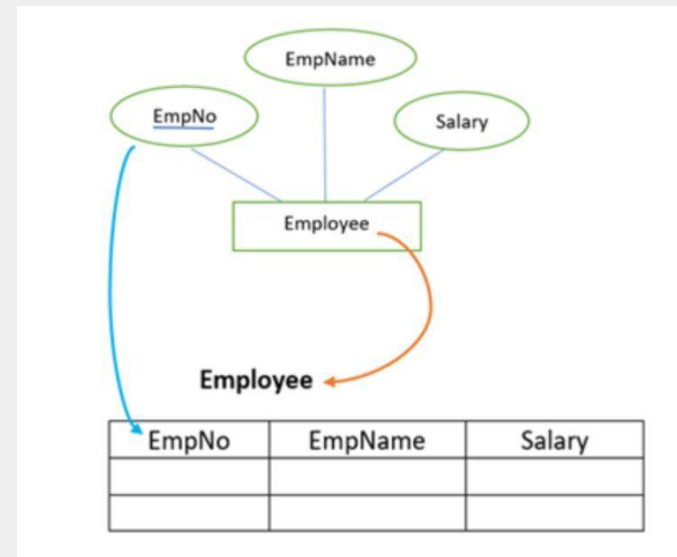
El modelo ER está más orientado semánticamente que el modelo relacional. Reglas:

- Cada entidad debe transformarse en una tabla. El nombre de la tabla debe llamarse con el nombre que empieza por T Ejemplo Entidad Empleado → Tabla TEEmpleado
- Todo atributo simple se transformará en un campo de una tabla.
- La clave primaria será la clave principal para identificar la relación. (Debe aparecer subrayada). La **clave** alternativa debe ser el **vínculo**
- Los atributos obligatorios no pueden ser nulos (restricción de dominio)

De E / R al modelo relacional: Entidades

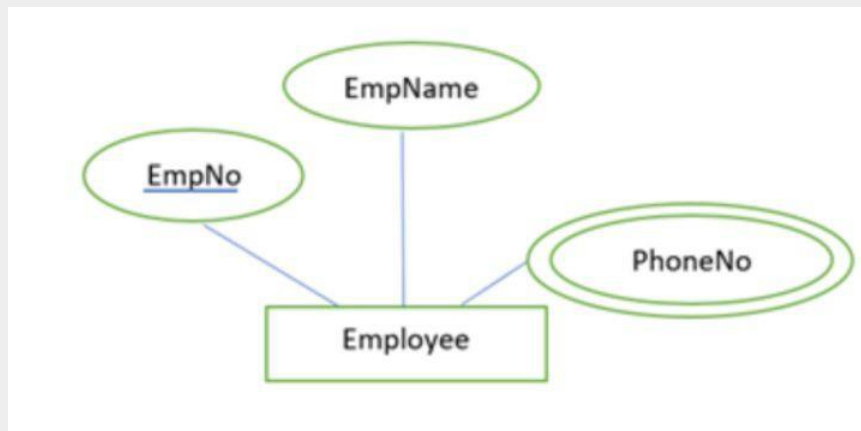
- Los atributos obligatorios no pueden ser nulos (restricción de dominio)

TEmpleado		
<u>EmpNo</u>		
EmpName		
Salario		



De E / R al modelo relacional: Entidades

- **Atributos multivaluados:** Se crea una nueva relación con la clave primaria de la entidad y el atributo multivaluado, siendo ambos la clave primaria de la nueva relación.

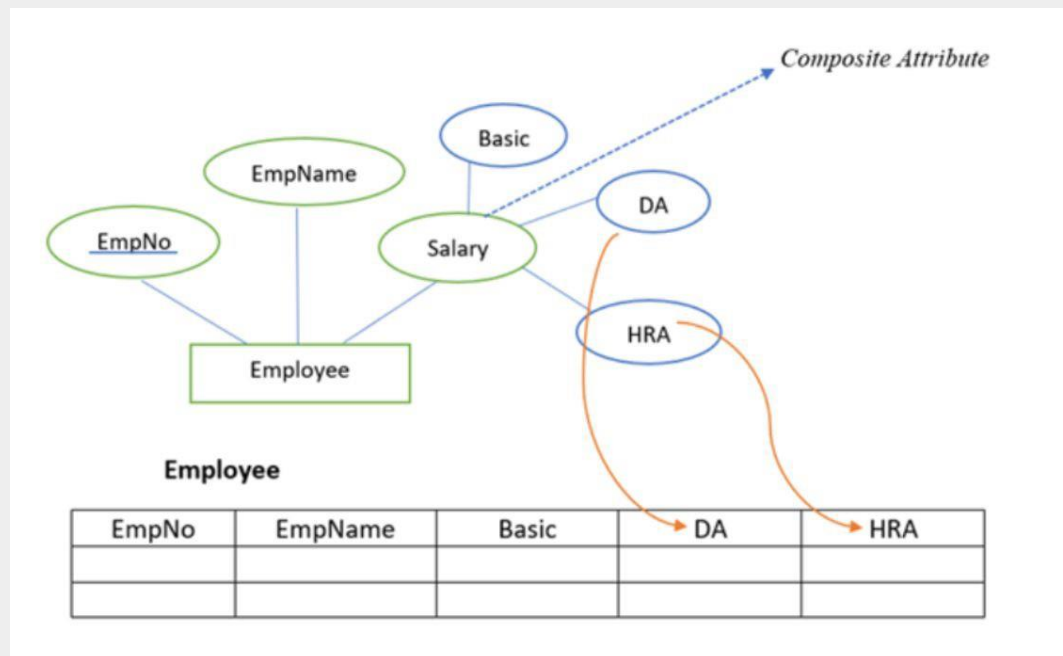


EmpNo	EmpName

EmpNo	PhoneNo

De E / R al modelo relacional: Entidades

- Los atributos **compuestos** se transformarán en atributos simples, es decir, en una columna más de la tabla.



De E / R al modelo relacional: Relaciones N: M

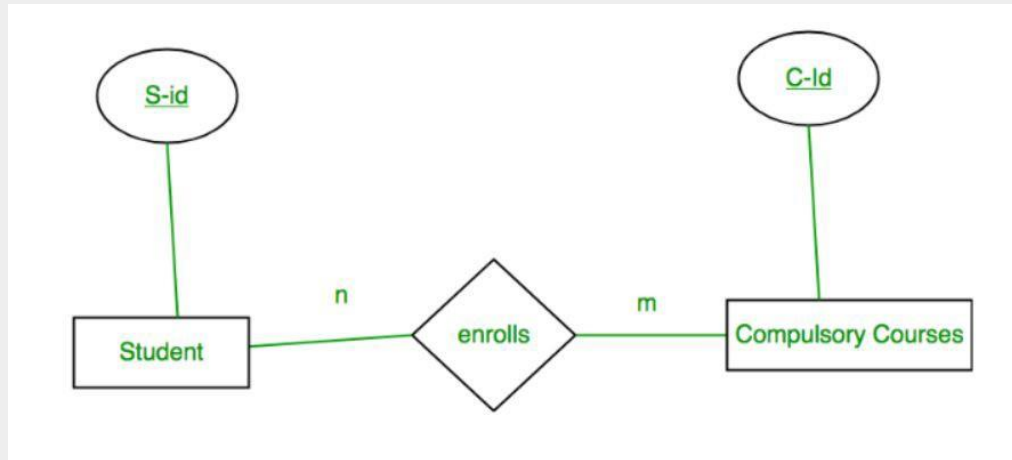
Binario N:M Relaciones entre la entidad A y la entidad B

Construir una nueva tabla con:

- Añade una columna para cada clave primaria como clave foránea
- La clave primaria contendrá estas claves foráneas y puede otra columna para identificar la tupla.
- Los atributos de la relación se añadirán como columna a esta nueva tabla.

De E / R al modelo relacional: Relaciones N: M

Relaciones binarias N:M



Los atributos de la relación deben añadirse como un campo de su tabla

TStudent		Cursos_Obligatorios
<u>S-id</u>		<u>C-id</u>
	TEmpleado	
	<u>S-id</u>	
	<u>C-id</u>	

De E / R al modelo relacional: Relación 1:N

Relaciones binarias 1:N

Para este tipo hay 3 posibilidades:

Primer escenario

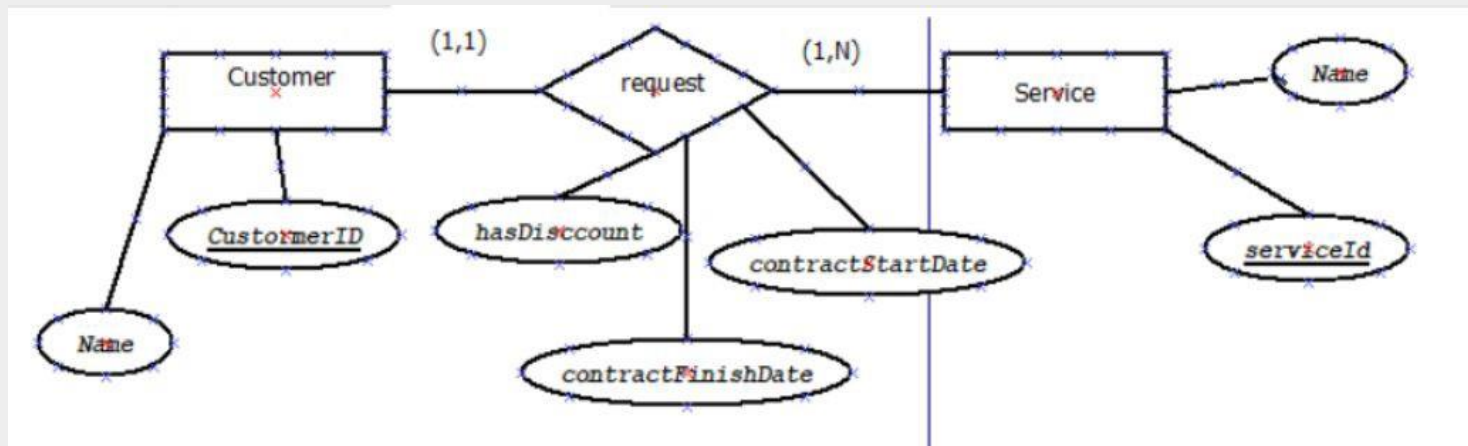
Si hay al menos un **atributo para la relación**, construya una **nueva tabla** añadiendo las dos claves primarias como claves ajenas y los atributos de la relación y define la clave primaria para esta nueva tabla.

De E / R al modelo relacional: Relación 1:N

Relaciones binarias 1:N

Para este tipo hay 3 posibilidades:

Primer escenario



De E / R al modelo relacional: Relación 1:N

Relaciones binarias 1:N

Para este tipo hay 3 posibilidades:

Primer escenario

TCliente		TService
<u>customerID</u>		<u>serviceID</u>
nombre		descripción
	TRequest	
	<u>customerID</u>	
	<u>serviceID</u>	
	<u>contractStartDate</u>	
	contractFinishDate	
	hasDiscount	

De E / R al modelo relacional: Relación 1:N

Binario 1:N Relaciones entre la entidad A y la entidad B

Segundo escenario

Si allí una de las cardinalidades es $(0,1)$ (cardinalidad A) y la otra es $(0,n)$ (cardinalidad B)

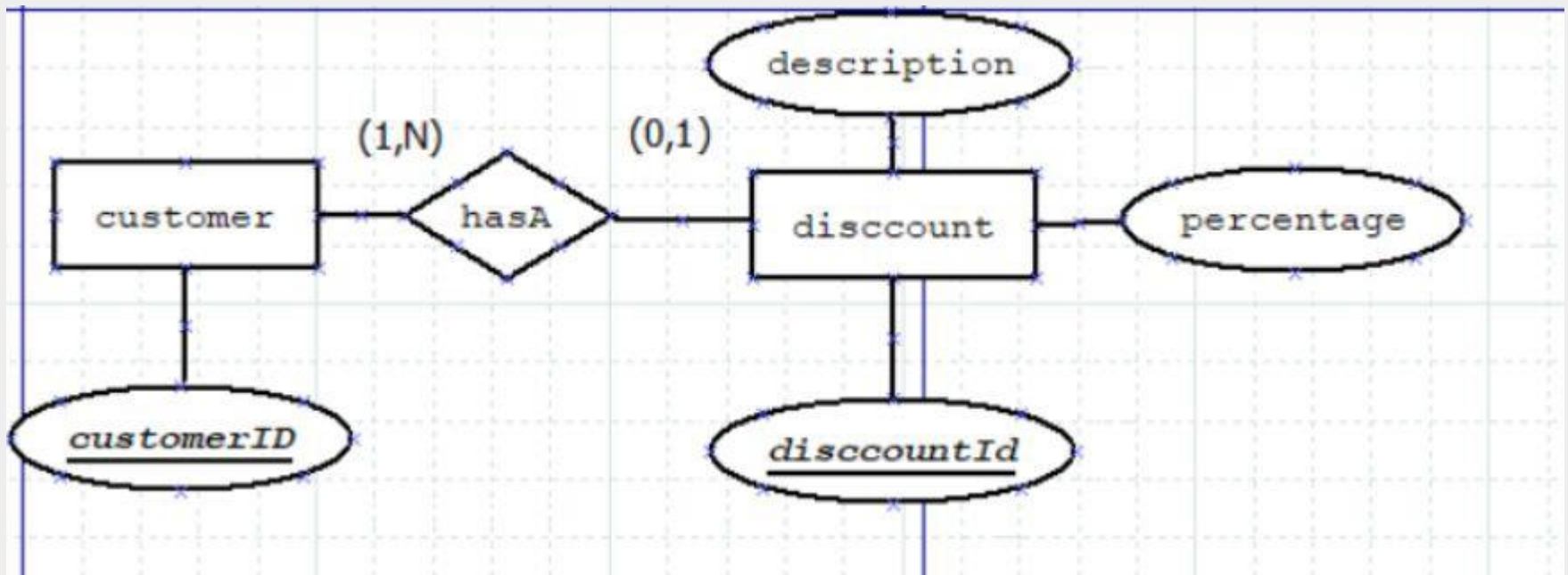
La cardinalidad $(0,1)$ significa que la entidad A podría existir sin relación con B.

En este escenario, deberíamos **construir una nueva tabla** para la relación añadiendo las dos claves primarias como claves foráneas y definiendo la clave primaria para esta tabla. Esta clave primaria podría ser la combinación de las dos claves externas y quizás otro campo más.

De E / R al modelo relacional: Relación 1:N

Binario 1:N Relaciones entre la entidad A y la entidad B

Segundo escenario



De E / R al modelo relacional: Relación 1:N

Binario 1:N Relaciones entre la entidad A y la entidad B

Segundo escenario

TCliente		TDiscount
<u>customerID</u>		<u>disccountID</u>
Nombre		descripción
	ThasA	porcentaje
	<u>customerID</u>	
	<u>disccountID</u>	
	descripción	
	porcentaje	

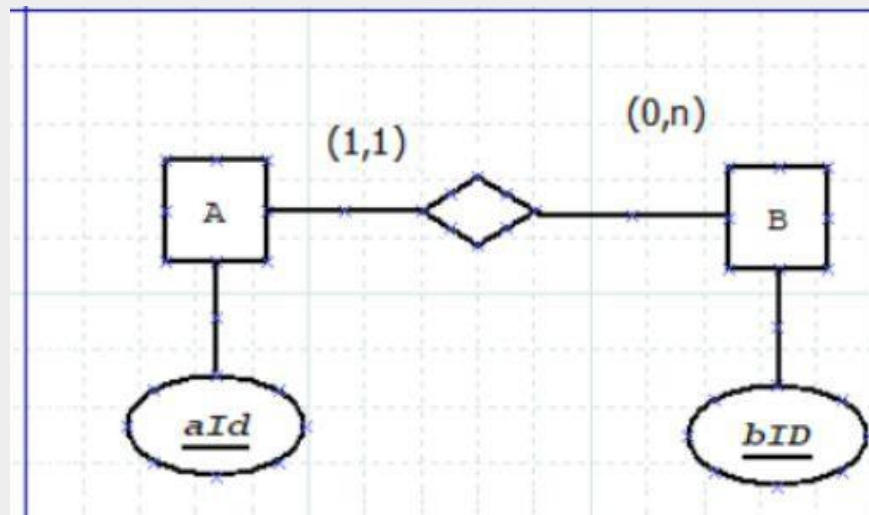
De E / R al modelo relacional: Relación 1:N

Binario 1:N Relaciones entre la entidad A y la entidad B

Tercer escenario

Si no hay atributos de **relación** y una de las cardinalidades es (1,1) (entidad a) y la otra es (0,n) (cardinalidad B)

La cardinalidad (1,1) implica que una entidad no existirá sin relación

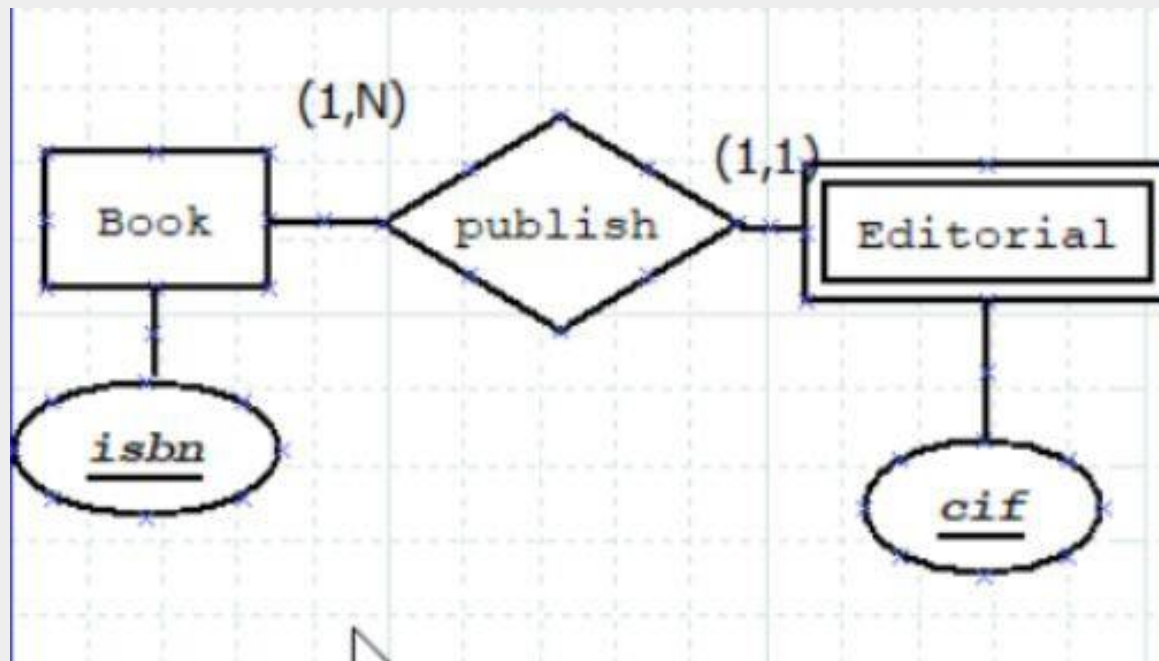


De E / R al modelo relacional: Relación 1:N

Binario 1:N Relaciones entre la entidad A y la entidad B

Tercer escenario

Por ejemplo:



De E / R al modelo relacional: Relación 1:N

Relaciones binarias 1:N

Tercer escenario

Si no hay atributos de relación y una de las cardinalidades es (1,1).

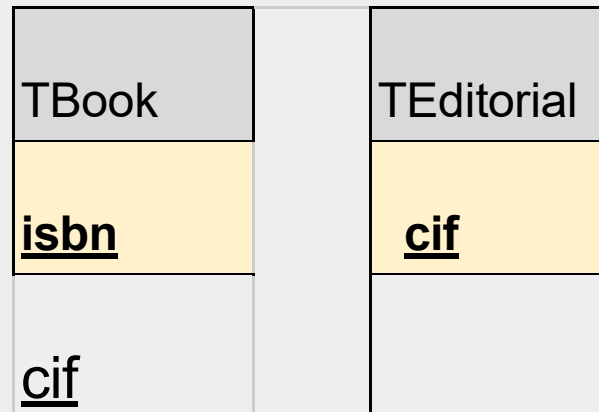
Debemos **propagar la clave primaria** de la entidad que tiene cardinalidad máxima 1 a la que tiene la cardinalidad máxima N

De E / R al modelo relacional: Relación 1:N

Relaciones binarias 1:N

Tercer escenario

Para nuestro ejemplo, Libro (1,n) y Editorial (1,1). Debemos propagar cif a la tabla TBook como clave foránea.



De E / R al modelo relacional: Relación 1:1

Binario 1:1 Relaciones entre la entidad A y la entidad B

El objetivo es minimizar los files vacíos

Primer escenario: Ambas cardinalidades son (0,1) o hay relaciones atributos

(0,1)

(0,1)

Un estudiante puede ser el líder de un equipo y un equipo puede tener o no un líder.

De E / R al modelo relacional: Relación 1:1

Primer escenario: Ambas cardinalidades son (0,1) o hay relaciones atributos

En este escenario, construimos una nueva tabla añadiendo ambas claves primarias. De este modo, minimizamos los campos vacíos.

Tstudent		TTeam
<u>studentID</u>		<u>teamID</u>
Name		Name
	TisLeaderOf	
	<u>studentID</u>	
	<u>teamID</u>	

De E / R al modelo relacional: Relación 1:1

Segundo escenario: Para todos los demás escenarios

Deberíamos añadir la clave primaria a la otra tabla. ¿Pero en cuál?

- Si hay una cardinalidad (0,1) (entidad A) y otra (1,1) (entidad B). Esto implica que la entidad A puede no existir sin la relación (entidad débil). En este caso la tabla A recibirá la clave primaria de la entidad B

De E / R al modelo relacional: Relación 1:1

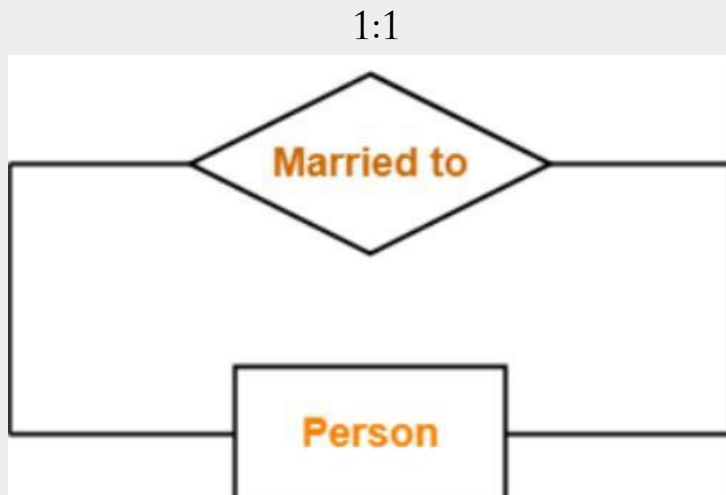
Segundo escenario: Para todos los demás escenarios

- Si ambos son (1,1), debes revisar:
 - Si una entidad debe crearse antes que la otra. En este caso, la segunda recibirá la clave primaria de la primera.
 - Si ambas entidades existen sin depender de la otra. Para este escenario, no hay ninguna regla. Se puede añadir una nueva tabla o se puede gestionar añadiendo la clave primaria como clave ajena a la otra

De E / R al modelo relacional: Relaciones Reflexivas

Básicamente, aplica las **mismas reglas que las relaciones binarias**

- **Relación 1:1** Una tabla con dos campos: uno como clave primaria y otro como clave externa.



Person a		
<u>idPerson a (PK)</u>	nombre	<i>casadoCon (FK)</i>

De E / R al modelo relacional: Relaciones Reflexivas

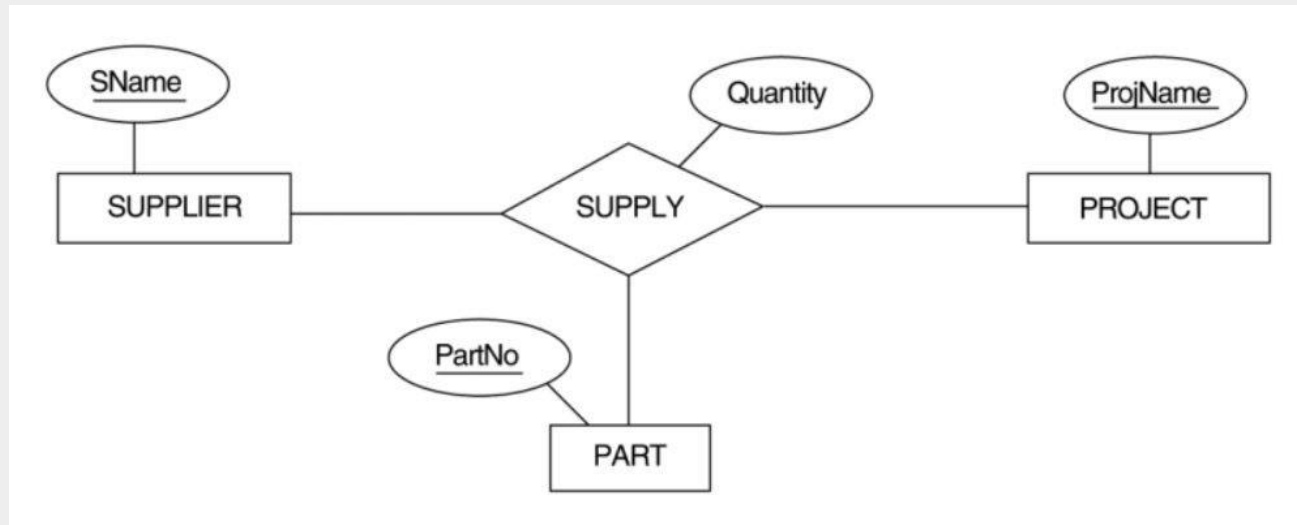
- Relación N:M Crear una nueva tabla con las dos claves primarias
- Relación 1:N Tenemos dos casos:
 - En el que la entidad muchos es obligatoria, se procede como en el caso (1,1), propagando de la clave primaria de la tabla de cardinalidad 1 a la de cardinalidad N
 - Si no es obligatorio, se crea una nueva tabla cuya clave será la de la entidad y la clave se propaga también a la nueva tabla como clave ajena.

De E / R al modelo relacional: Relación ternaria o más

N-ario donde N debe ser > 2

Debemos añadir una nueva tabla que incluya todas las claves primarias de las relaciones y también los atributos de las mismas.

Después, debemos revisar la cardinalidad para comprobar si se puede simplificar.



De E / R al modelo relacional: Relación ternaria o más

SUPPLIER

<u>SNAME</u>	...
--------------	-----

PROJECT

<u>PROJNAME</u>	...
-----------------	-----

PART

<u>PARTNO</u>	...
---------------	-----

SUPPLY

<u>SNAME</u>	PROJNAME	<u>PARTNO</u>	QUANTITY
--------------	----------	---------------	----------

De E / R al modelo relacional: Relaciones jerárquicas

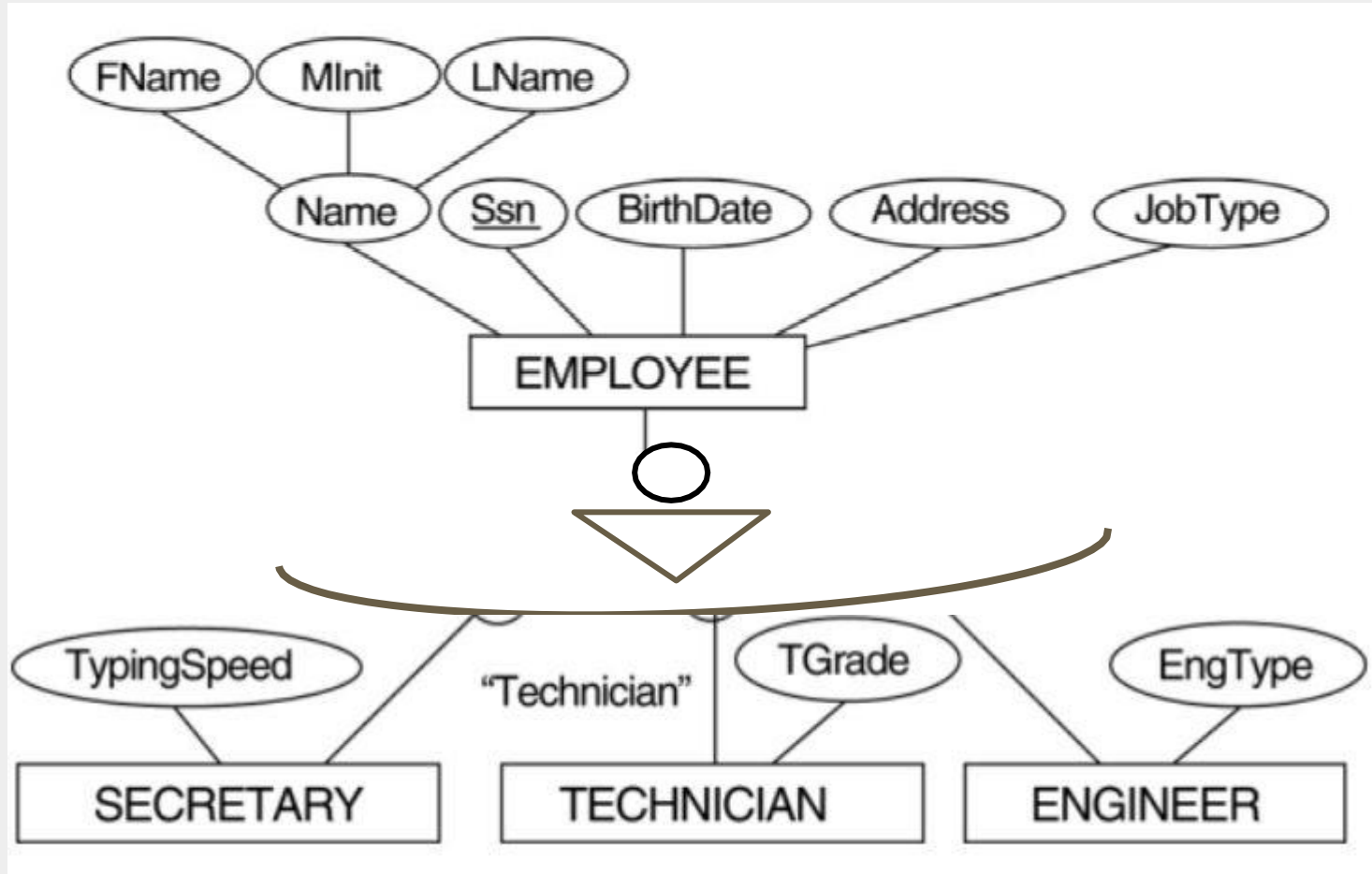
El modelo relacional no tiene ningún mecanismo para la representación de las relaciones jerárquicas. Por lo tanto, estas relaciones tienen que ser eliminadas. Esto debe implicar perder la semántica. Por ejemplo:

- Relaciones exclusivas → No hay forma de modelar con E/R. Se gestionará como una restricción en la base de datos.

Para gestionar eso hay diferentes opciones:

1. Primera opción, construir una tabla única con los atributos del padre y todos los atributos **de las entidades hijas al padre** Podría implicar muchos atributos nulos y tablas grandes.
2. Segunda opción, todos los atributos comunes que se asignen en la entidad padre se **propagarán a los hijos**. Esta solución sólo es posible **para las relaciones totales y exclusivas**.
3. Tercera opción, transformar a **una relación 1:1 entre el padre y cada uno de los hijos**.

De E / R al modelo relacional: Relaciones jerárquicas



De E / R al modelo relacional: Relaciones jerárquicas

1. Primera opción:

TEmployee(SSN, Name, Fname, Minit, LName, BirthDate, Address, JobType, TypingSpeed, TGrade, EngType)

2. Segunda opción: No es posible porque se superpone.
3. Tercera opción:

Empleado(SSN, Nombre, Fname, Minit, LName, BirthDate, Address)

Secretaria(SSN, TypingSpeed)

Técnico (SSN, TGrade) Ingeniero

(SSN, EngType)

Resumen de las transformaciones de E/R a relacionales

Modelo E/R			Modelo relacional
Entidad			Crear una nueva tabla (PK)
Atributo	Simple		Nueva Columna en la tabla de Entidades Obligatoria no puede ser nula
	Compuesto		Nuevas columnas para cada uno de los atributos simples
	Multivalores		Nueva tabla con las claves primarias de las entidades y el atributo, añadir una fila por valor
Relaciones	Binario N:M Cardinalidad		Añadir una nueva tabla con las dos claves primarias
	Unario, Binario 1:N	Al menos un atributo del relación	Añadir una nueva tabla con las dos claves primarias
		Una cardinalidad (0,1) (A)	Añadir una nueva tabla con las dos claves primarias
		ONE cardinalidad (1,1) (A)	Propagar la clave primaria de A (cardinalidad 1) a la tabla B (cardinalidad N) como clave externa
	Unario, Binario 1:1	Al menos un atributo del relación	Añadir una nueva tabla con las dos claves primarias
		Ambos (0,1)	Añadir una nueva tabla con las dos claves primarias
		(0,1)A (1,1) B o ambos (1,1)	Propagar la clave primaria B (cardinalidad 1,1) a la tabla A (cardinalidad 0,1) como clave externa
	Ternario o más		Añade una nueva tabla añadiendo todas las claves primarias y después de revisarla se podría simplificar

Normalización: 1FN

La normalización es el proceso de estructuración de una base de datos relacional para reducir la redundancia de los datos y mejorar su integridad.

Hay diferentes niveles de normalización.

- Primera forma normal (1FN) Una relación está en primera forma normal,

TABLE_PRODUCT

Product ID	Color	Price
1	red, green	15.99
2	yellow	23.99
3	green	17.50
4	yellow, blue	9.99
5	red	29.99

TABLE_PRODUCT_PRICE

Product ID	Price
1	15.99
2	23.99
3	17.50
4	9.99
5	29.99

TABLE_PRODUCT_COLOR

Product ID	Color
1	red
1	green
2	yellow
3	green
4	yellow
4	blue
5	red

si no hay atributos multivaluados.

Transformación a

1FN

Normalización: 2FN

- Una relación está en la Segunda Forma Normal (**2FN**) si cumple los dos requisitos siguientes:
 - Está en la primera forma normal.
 - Todos los atributos que **no forman parte de la clave primaria dependen totalmente de ella**. Esto significa que cada campo que no sea clave debe referirse a lo mismo que la clave primaria.

Normalización: 2FN

TABLE_PURCHASE_DETAIL

Customer ID	Store ID	Purchase Location
1	1	Los Angeles
1	3	San Francisco
2	1	Los Angeles
3	2	New York
4	3	San Francisco

Esta tabla tiene una **clave primaria compuesta** [ID de cliente, ID de tienda]. Hay un atributo no clave [Ubicación de la compra]. En este caso, [Ubicación de la compra] **sólo depende de [ID de la tienda]**, que es sólo una parte de la clave primaria. Por lo tanto, esta tabla no satisface la segunda forma normal.

Normalización: 2FN

TABLE_PURCHASE

Customer ID	Store ID
1	1
1	3
2	1
3	2
4	3

TABLE_STORE

Store ID	Purchase Location
1	Los Angeles
2	New York
3	San Francisco

Transformación a 2FN

Normalización: 3FN

- Una relación está en la Tercera Forma Normal (**3FN**) si cumple los dos requisitos siguientes:
 - Está en segunda forma normal.
 - Si todos los atributos que **no forman parte de la clave primaria son independientes entre sí**, es decir, no dan información sobre otros atributos de la relación. Este tipo de dependencia se denomina dependencia funcional en el esquema relacional. En otras palabras, se considera que una tabla está en tercera normal si la tabla/entidad ya está en segunda normal y las columnas de la tabla/entidad son no depende transitivamente de la clave primaria.

Normalización: 3FN

Dependencia transitiva. Ocurre cuando un atributo depende, además de la clave primaria (2FN), de otro atributo no clave, entonces se dice que tiene dependencia funcional transitiva.

3FN implica no tener dependencias transitivas.

Normalización: 3FN

player_id	player_rating	player_skill_level
jane	Intermediate	6
john	Beginner	3
mary	Advanced	8
lisa	Advanced	9

¿1FN? Sí, no hay atributos multivalentes

¿2FN? Sí, la clave primaria es player_id y el resto de los atributos dependen de la clave primaria.

¿3FN? No, porque player_skill_level depende del player_rating

Normalización: 3FN

Tercera forma Transformación

Dividir el en dos tabletas: una con la clave primaria y el nivel y una segunda con el nivel y la calificación

player_id	player_skill_level
jane	6
john	4
mary	8
lisa	9

player_skill_level	player_rating
1	Beginner
2	Beginner
3	Beginner
4	Intermediate
5	Intermediate
6	Intermediate
7	Advanced
8	Advanced
9	Advanced



Referencias

- Rafael Lozano, Introducción a las Bases de Datos
- Arturo Mora Rioja. Base de datos. Diseño y Gestión. Editorial Sintesis Madrid 2014
- Antonio Postigo Palacios, Gestión de base de datos Editorial Paraninfo Madrid 2022
- <https://www.guru99.com/relational-data-model-dbms.html>
- <https://www.scaler.com/topics/dbms/relational-model-in-dbms/>