



BASES DE DATOS RELACIONALES

Modelado

Por
Ing. Elizabeth León Guzmán, PhD.
Profesora
Ingeniería de Sistemas y Computación

Contenido

- Diagrama E/R
 - Superclases y Subclases
- Modelo Relacional
 - Relación
 - Terminología
- Conversión E/R Modelo Relacional
- Normalización

Subclases

Cuando una entidad tiene instancias (tuplas) con **especificaciones particulares en sus atributos o relaciones** se generan subclases

Ejemplo

Persona (cedula, nombre, apellido, edad, correo, teléfono)

Estudiante(cedula, nombre, apellido, edad, correo, teléfono, **código**)

Profesor (cedula, nombre, apellido, edad, correo, teléfono,
oficina, profesión)

Atributo único de estudiante

Atributos únicos
de profesor

Estudiante y profesor SON personas
son Subclases de Persona

Subclases - Notación en E/R

Subclases que se especializan por **atributos**.

Tanto estudiante como profesor tienen atributos específicos

SUBCLASES

PERSONA

per_cedula
per_nombre
per_edad
per_telefono
per_correo

SUPERCLASE

ESTUDIANTE

est_codigo

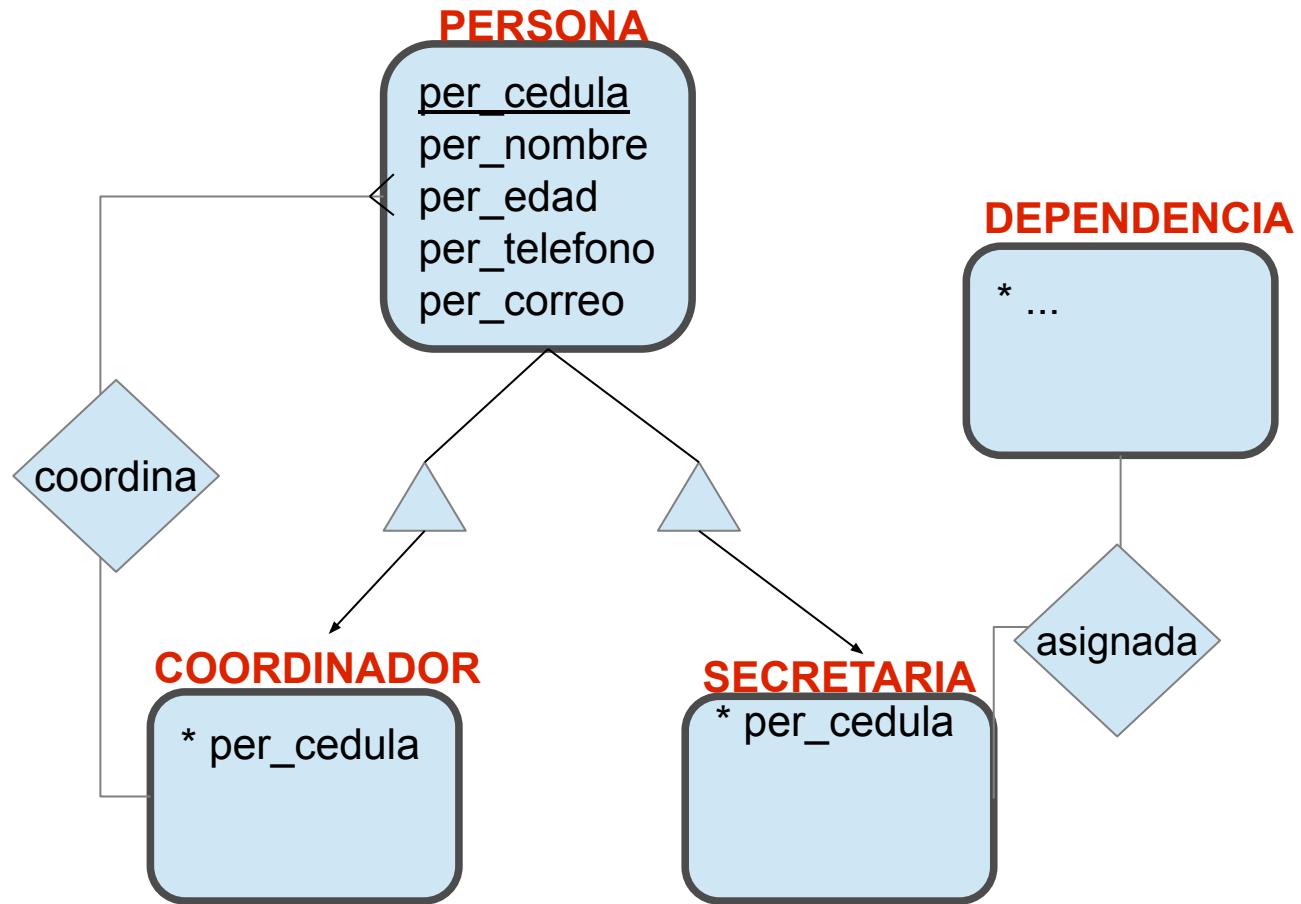
PROFESOR

pro_oficina
pro_profesión

Las subclases “heredan” los atributos de la superclase

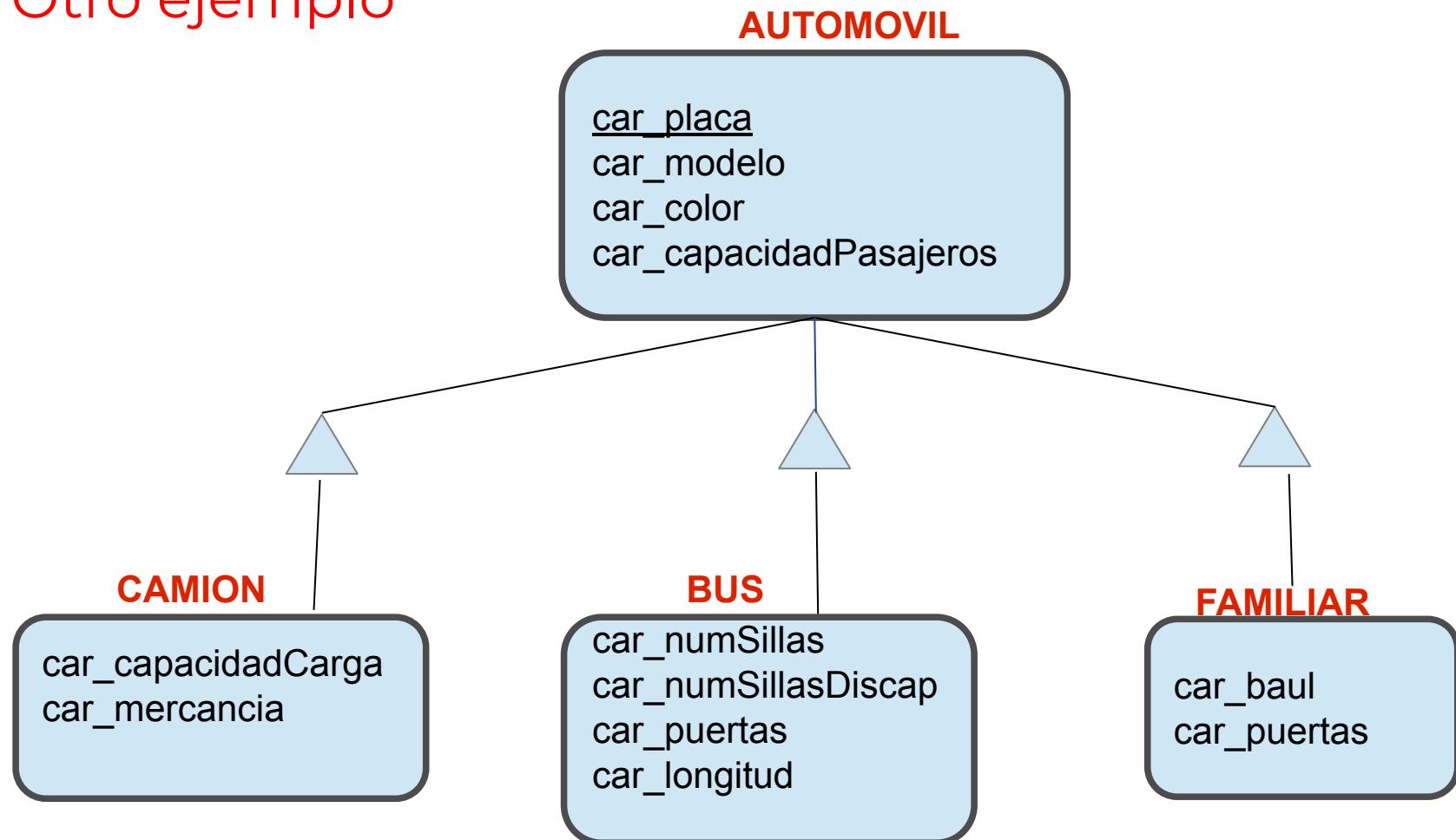
Subclases - Notación en E/R

Subclases que se especializan por **relaciones**. secretaria tiene relación con la entidad dependencia, y Coordinador con Persona



Subclases

Otro ejemplo



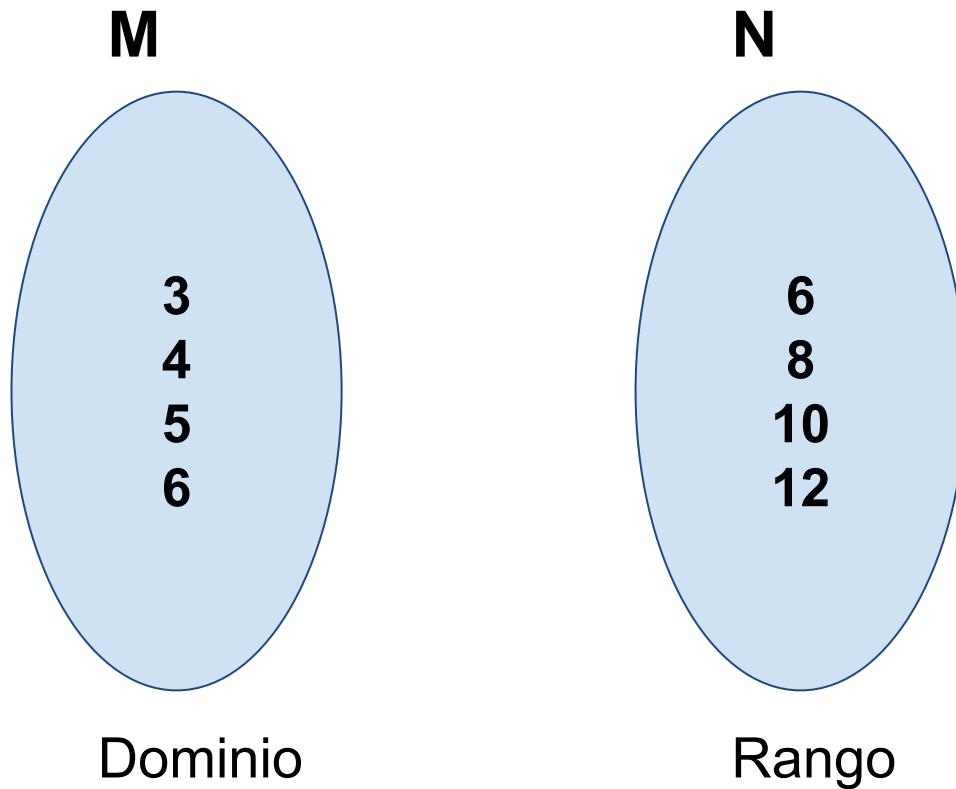
Modelo Relacional

Una base de datos relacional es una **colección de relaciones** que contienen los datos que describen un ambiente de negocios

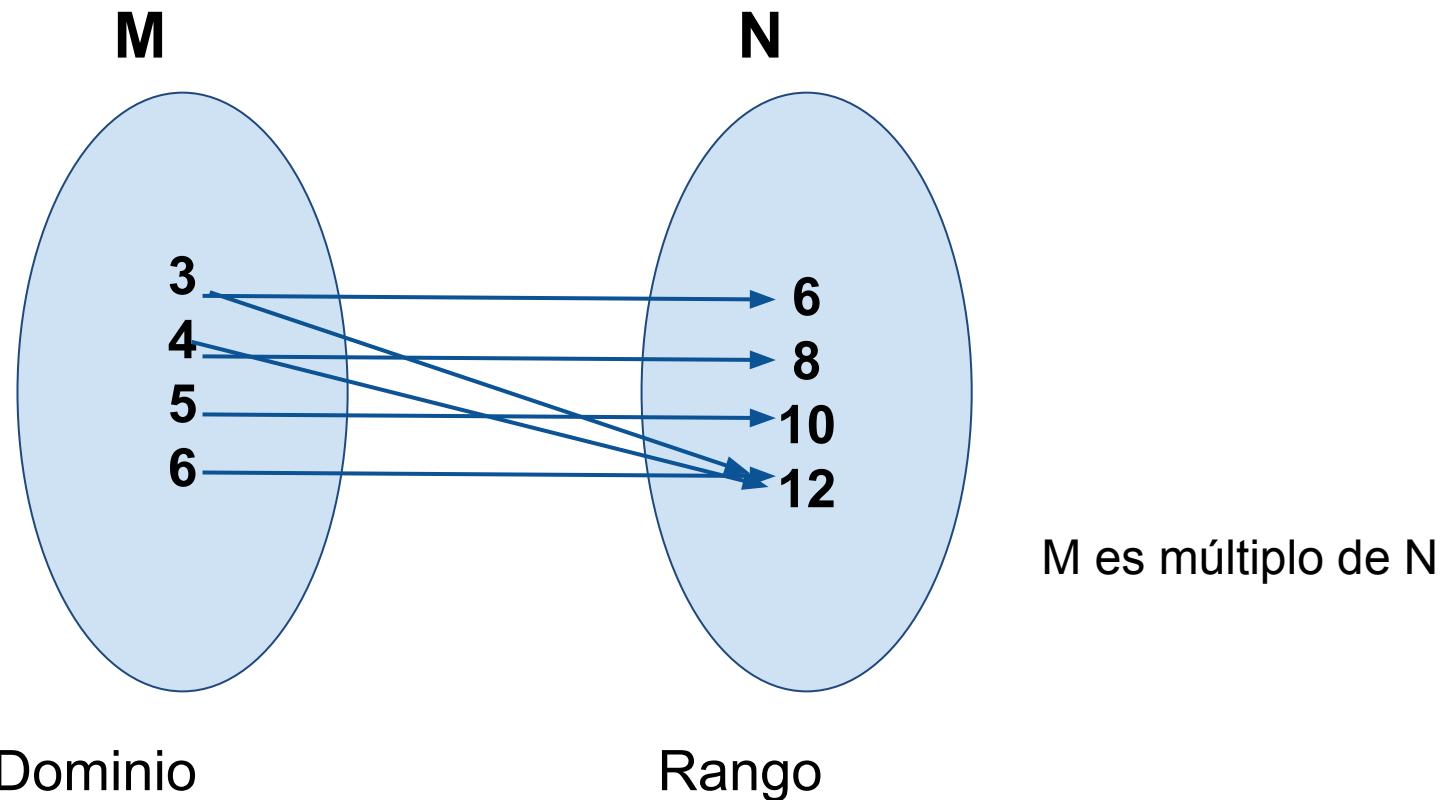
Las relaciones se representan en **tablas** y cada una tiene un nombre exclusivo

Basado en **matemáticas**

Relación



Relación



Relación es la correspondencia entre dos conjuntos.
Correspondencia entre el Dominio y el Rango. Cada elemento del dominio le corresponde uno o más elementos del rango

Relación

- Una relación matemática de **M** en **N** será un subconjunto del producto cartesiano **M x N**.

$$M \times N = \{ (3, 6), (3, 8), (3, 10), (3, 12), \\ (4, 6), (4, 8), (4, 10), (4, 12), \\ (5, 6), (5, 8), (5, 10), (5, 12) \\ (6, 6), (6, 8), (6, 10), (6, 12) \}$$

- Las relaciones serán pares ordenados que vinculan elementos de **M** con elementos de **N**.

Relación

- Una relación matemática de **M** en **N** será un subconjunto del producto cartesiano **M x N**.

$$M \times N = \{ (\mathbf{3}, \mathbf{6}), (3, 8), (3, 10), (\mathbf{3}, \mathbf{12}), \\ (4, 6), (\mathbf{4}, \mathbf{8}), (4, 10), (\mathbf{4}, \mathbf{12}), \\ (5, 6), (5, 8), (\mathbf{5}, \mathbf{10}), (5, 12) \\ (6, 6), (6, 8), (6, 10), (\mathbf{6}, \mathbf{12}) \}$$

$$\{(3, 6), (3, 12), (4, 8), (4, 12), (5, 10), (6, 12)\} \subset M \times N$$

- Las relaciones serán pares ordenados que vinculan elementos de **M** con elementos de **N**.

Relación

Relación → Tabla

Columna
Campo
Atributo

Estudiante

Codigo	Nombre	Edad	Genero	Telefono
259875	Carlos	18	m	3114123456
259040	Jorge	19	m	3126543211
256734	Ana	18	f	3001298765
250021	Maria	20	f	3109977564

fila
registro
tupla

Relación

Conjunto de valores permitido para un atributo → “Dominio” del atributo

Estudiante

Código	Nombre	Edad	Género	Teléfono
259875	Carlos	18	m	3114123456
259040	Jorge	19	m	3126543211
256734	Ana	18	f	3001298765
250021	Maria	20	f	3109977564

Conjunto de todos
los códigos

→ D₁ D₂ D₃ D₄ D₅

Cada fila consisten en una tupla (v₁, v₂, v₃, v₄, v₅) donde v₁ está en el dominio D₁,
v₂ está en el dominio D₂ ...

Por lo tanto:

$$\text{ESTUDIANTE}(\text{codigo}, \text{nombre}, \text{edad}, \text{genero}, \text{telefono}) \subset D_1 \times D_2 \times D_3 \times D_4 \times D_5$$

Relación

Matemáticamente las relaciones son subconjuntos del producto cartesiano de la lista de dominios.

Conjunto de tuplas

Cada tupla de una relación es una lista de componentes
La relación tiene aridad fija (número de componentes)

Relación

Estudiante

Codigo	Nombre	Edad	Genero	Telefono
259875	Carlos	18	m	3114123456
259040	Jorge	19	m	3126543211
256734	Ana	18	f	3001298765
250021	Maria	20	f	3109977564

Relación

Estudiante

Codigo	Nombre	Edad	Genero	Telefono
	259875	Carlos	18	m
259040	Jorge	19	m	3126543211
256734	Ana	18	f	3001298765
250021	Maria	20	f	3109977564

$t = <259875, \text{Carlos}, 18, m, 3114123456> \in \text{ESTUDIANTE}$

Relación

Codigo	Nombre	Edad	Genero	Telefono
259040	Jorge	19	m	3126543211
256734	Ana	18	f	3001298765
250021	Maria	20	f	3109977564
259875	Carlos	18	m	3114123456



Llaves

Una relación **siempre tiene** una llave primaria

Una llave primaria (llamada clave) es **un atributo** o un **grupo de atributos** en los que los valores son **únicos** en todas las tuplas (filas) de la relación.

Estudiante

Codigo	Nombre	Edad	Genero	Telefono
259875	Carlos	18	m	3114123456
259040	Jorge	19	m	3126543211
256734	Ana	18	f	3001298765
250021	Maria	20	f	3109977564

Llaves candidatas

- Cuando una relación tiene más de un atributo (o grupos de atributos) que representen las tuplas como únicas.
- Se debe escoger una como llave primaria.

Codigo	Nombre	Edad	Genero	Telefono	Cedula	PAPA
259875	Carlos	18	m	3114123456	121212	3.8
259040	Jorge	19	m	3126543211	343434	3.5
256734	Ana	18	f	3001298765	767756	4.1
250021	Maria	20	f	3109977564	367787	4.0

Llave primaria



Llaves Candidatas

Llaves foráneas o externas

Son llaves que son primarias en una relación, y aparecen como atributos en otra relación.

Son llaves foráneas para la relación en la que aparecen como atributos.

LIBRO

<u>ISBN</u>	Título	Precio	Páginas	<u>Id_Editorial</u>
19839	Redes de Computadores	100000	100	259875
22343	Matemáticas Básicas	110000	230	256734
33432	Programación en Python	80000	90	259040
46767	Programación	75000	110	256734

Atributo que es llave foránea en la relación LIBRO

EDITORIAL

<u>Id_Editorial</u>	Nombre	Teléfono	Dirección
259875	Norma	3119282	Bogotá
259040	IEEE	3125663	NY
256734	Springer	3109374	Madrid

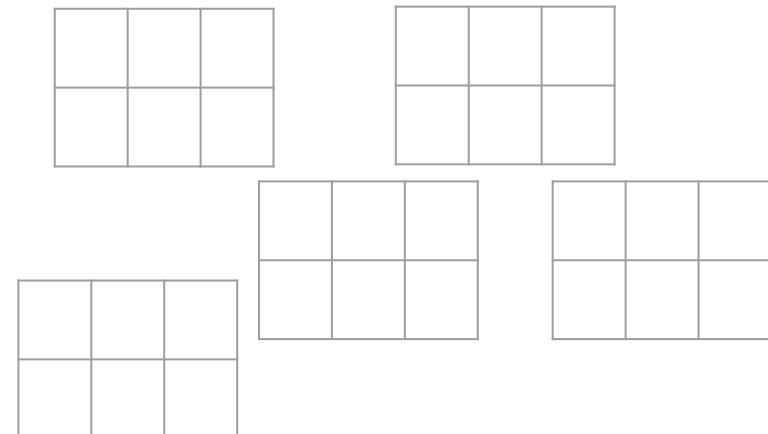
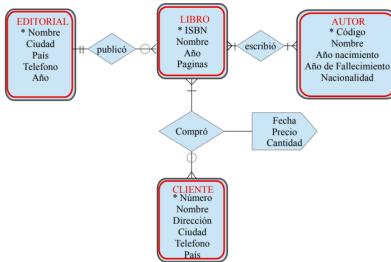
Conversión Diagrama E-R a Modelo Relacional

E/R

- Entidad
- Relaciones

Relacional

- Relación (tabla)
- Relación/Llaves foráneas



Conversión Diagrama E-R a Tablas relacionales

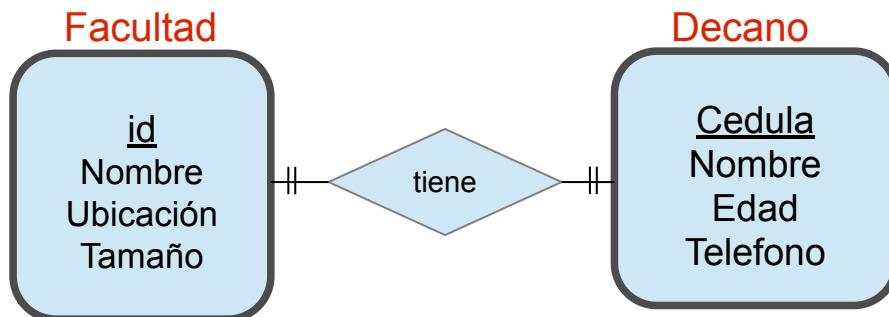
1. Toda **entidad** es una relación o tabla. Los atributos son columnas

2. Relaciones:
 - **1 a 1:** i) Atributos de una entidad pasan a ser atributos de la otra, o ii) Una sola relación con los atributos de las dos entidades

 - **1 a muchos:** Atributo llave primaria de la entidad con cardinalidad uno pasa a ser llave foránea (atributo) de la entidad con cardinalidad múltiple. Se crean dos relaciones.

 - **Muchos a muchos:** Cada entidad es una relación, y se crea una nueva relación con el nombre de la relación de las dos entidades y con atributos las llaves primarias de las entidades relacionadas (llaves foráneas). La llave primaria de la nueva relación son el grupo de llaves foráneas, o se crea una llave artificial

Uno a Uno



- 1 Una sola tabla (se unen)

Facultad

<u>Id</u>	Nombre	Ubicación	Tamaño	cedulaDecano	NombreDecano	Edad	telefono

- 2 Dos tablas. Una recibe llave foránea/externa

Facultad

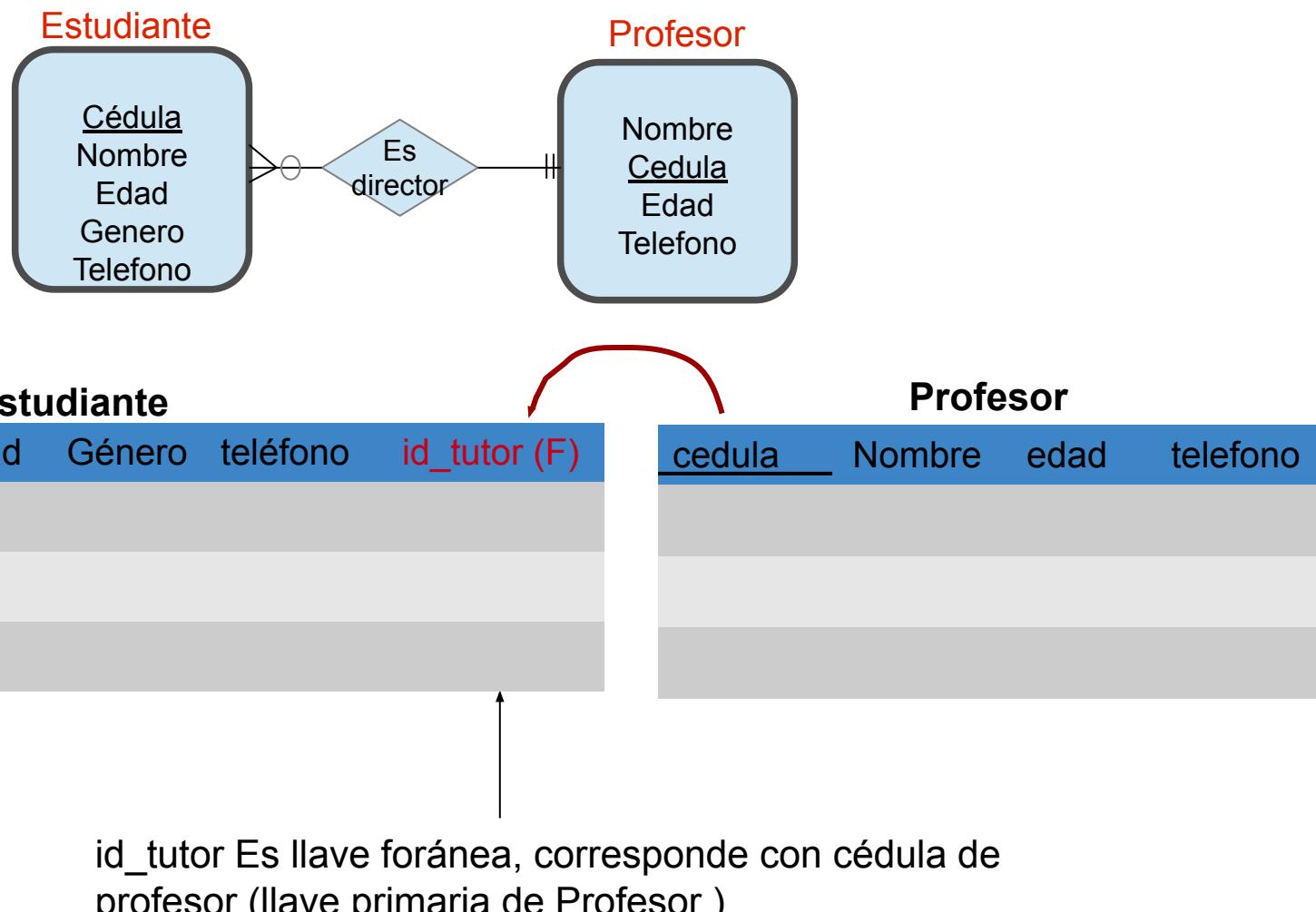
<u>Id</u>	Nombre	Ubicacion	Tamaño	cedula_decano

Decano

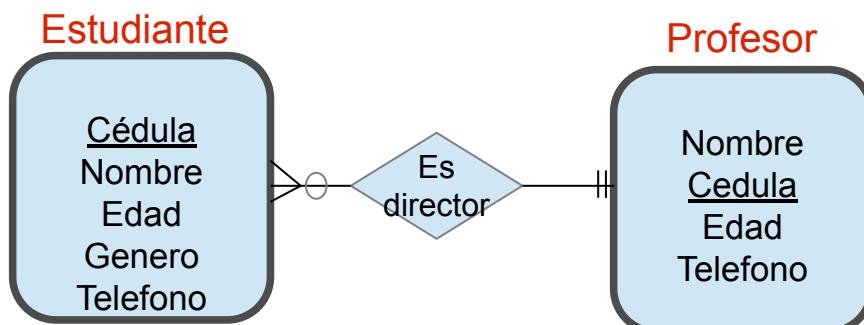
<u>cedula</u>	Nombre	Edad	telefono

* Llaves primarias subrayadas

Uno a Muchos



Uno a Muchos



Estudiante

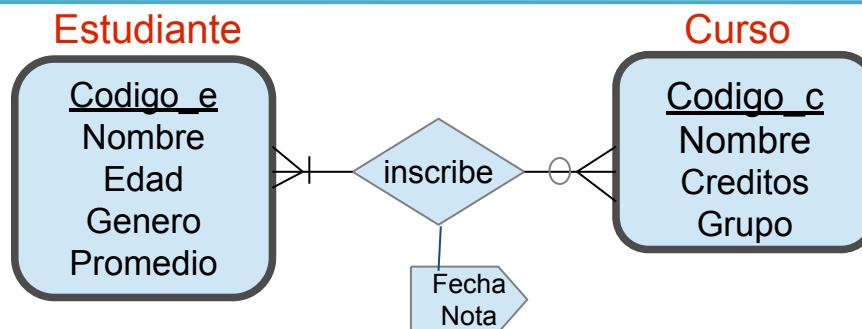
Profesor

Cédula	Nombre	Edad	Género	teléfono	id_tutor (F)
2344	Alicia	19	f	238231	22
1896	Fernando	20	m	492828	10
5678	Maria	20	f	671906	22

cedula	Nombre	edad	telefono
10	Pedro Diaz	35	2134562
22	Ana Suarez	40	2456783

id_tutor Es llave foránea, corresponde con cédula de profesor (llave primaria de Profesor)

Muchos a Muchos



<u>Codigo_e</u>	Nombre	Edad	Género	Promedio

<u>Codigo_c</u>	Nombre	Créditos	Grupo

Dos formas de crear **inscripción**:

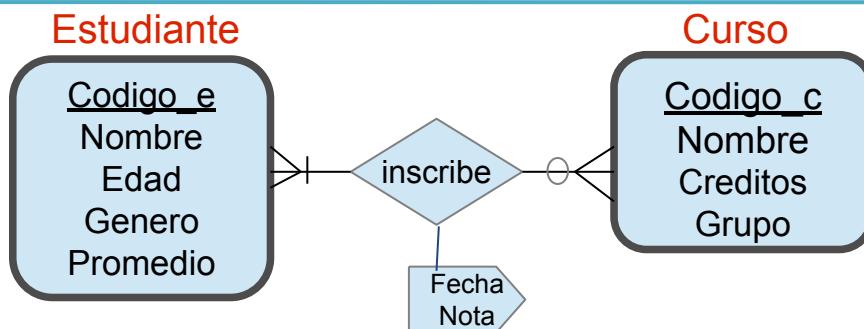
- 1 Llave primaria compuesta por llaves primarias de estudiante y curso

Inscripcion			
<u>Codigo_e</u>	<u>Codigo_c</u>	fecha	Nota

- 2 Se puede crear una llave artificial! Como llave primaria: id_incripcion. codigo_e y código_c serán solo llaves foráneas

Inscripcion				
<u>id_incripcion</u>	<u>Codigo_e</u>	<u>Codigo_c</u>	fecha	Nota

Muchos a Muchos

**Estudiante**

<u>Codigo_e</u>	Nombre	Edad	Género	Promedio
2343	Andrés	20	m	4,3
1325	Lucía	19	f	4,4

Curso

<u>Codigo_c</u>	Nombre	Créditos	Grupo
10	BD	3	1
20	Cálculo	4	1

Dos formas de crear **inscripción**:

- 1 Llave primaria compuesta por llaves primarias de estudiante y curso

Inscripcion

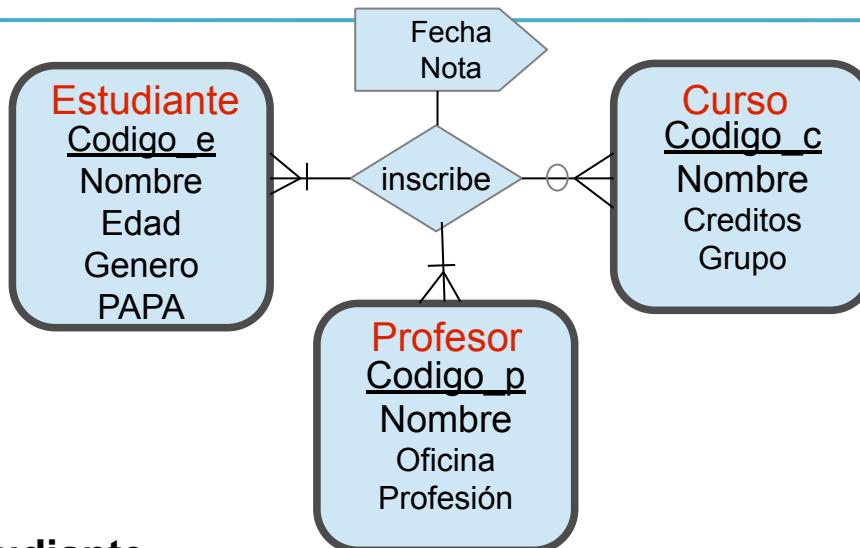
<u>Codigo_e</u>	<u>Codigo_c</u>	fecha	Nota
2343	20	2/7/2019	4,5
2343	10	2/7/2019	4,3

- 2 Se puede crear una llave artificial! Como llave primaria: id_incripcion. codigo_e y código_c serán solo llaves foráneas

Inscripcion

<u>id_incripcion</u>	<u>Codigo_e</u>	<u>Codigo_c</u>	fecha	Nota
1	2343	20	2/7/2019	4,5
2	2343	10	27/2019	4,3

Muchos a Muchos - Relación ternaria



Estudiante

Código_e	Nombre	Edad	Género	Promedio

Curso

Código_c	Nombre	Créditos	Grupo

Profesor

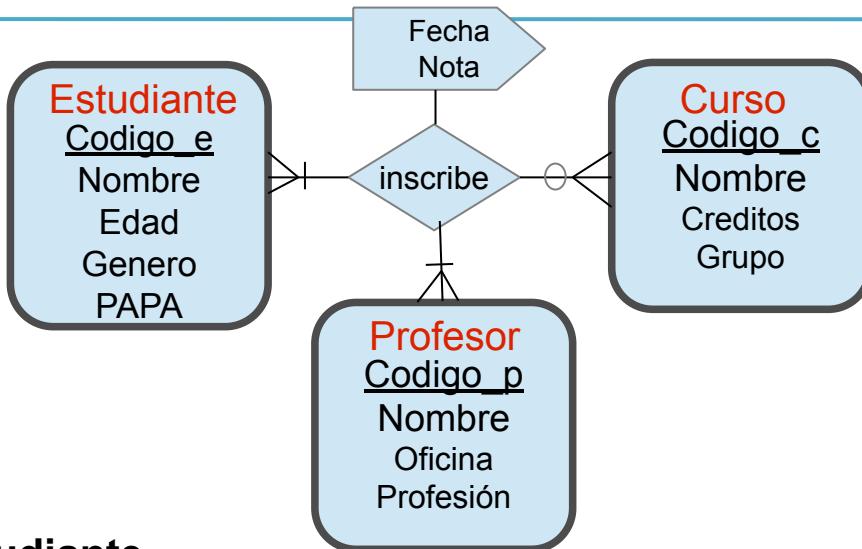
Código_p	Nombre	Oficina	Profesión

Inscripción

Código_e	Código_c	Código_p	fecha	Nota

Para Inscripción la Llave primaria es compuesta por las tres llaves primarias de las entidades relacionadas

Valores Nulos



Estudiante

Código_e	Nombre	Edad	Género	Promedio

Curso

Código_c	Nombre	Créditos	Grupo

Profesor

Código_p	Nombre	Oficina	Profesión

Inscripción

id_inscripción	Código_e	Código_c	Código_p	fecha	Nota

Para Inscripción la Llave primaria es compuesta por las tres llaves primarias de las entidades relacionadas

Valores Nulos

- En ocasiones, no se conoce el valor de un atributo para una determinada tupla.
- En esos casos, a ese atributo de esa tupla se le asigna un **valor nulo (NULL)**, que indica que el valor de ese atributo para esa tupla es *desconocido* o, simplemente, que ese atributo no es aplicable a esa tupla.

Estudiante

Codigo	Nombre	Edad	Género	Telefono	Promedio
23323	Jorge	18	M	3119019287	4.6
57485	Maria	19	F	NULL	4.5
43344	Andres	18	M	NULL	4.3
54556	Lucia	17	F	2829182277	4.3

Valores Nulos

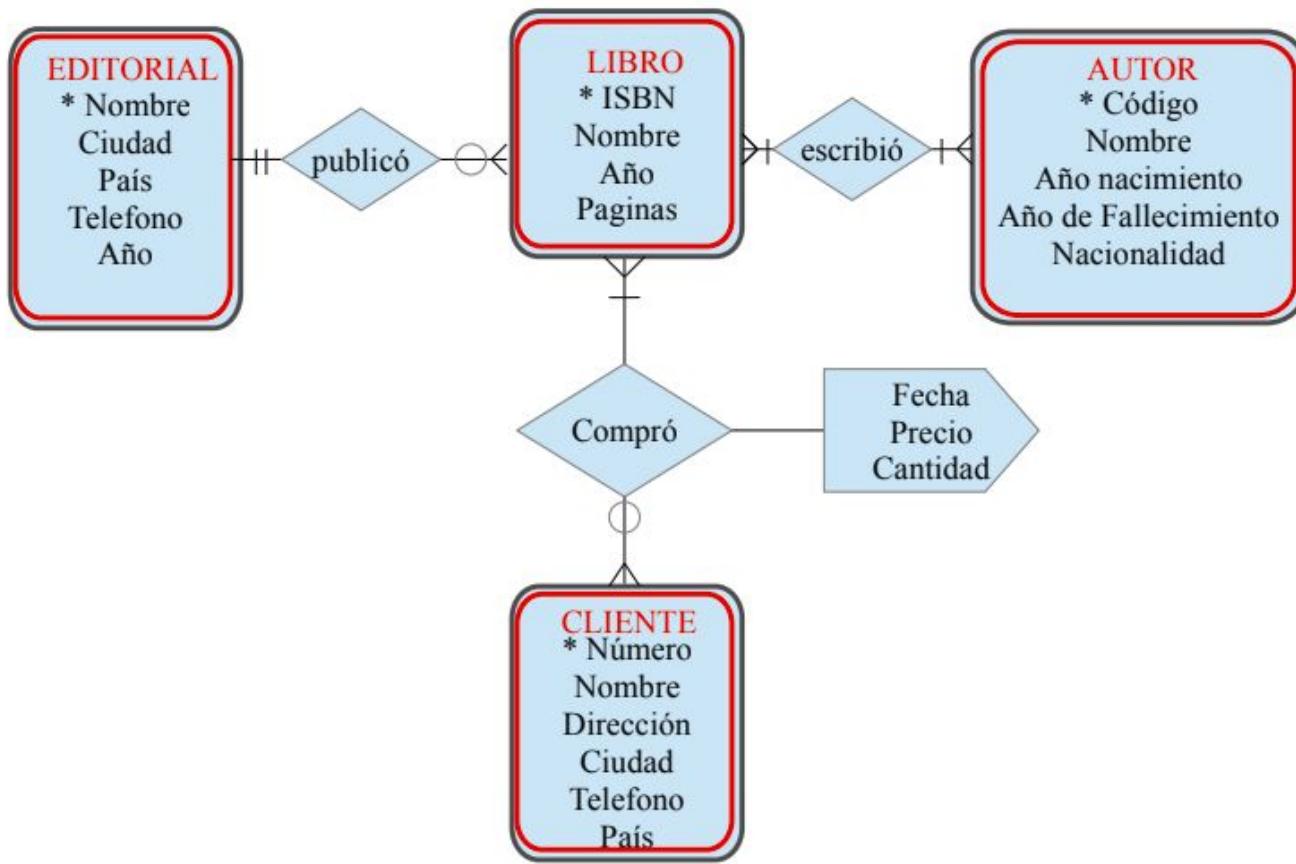
- Las modalidades de las relaciones se garantizan asignando la propiedad al atributo de permitir o no permitir valores NULL (parte de los metadatos).

Estudiante					
Codigo	Nombre	Edad	Género	Promedio	id_tutor
25754	Arturo	18	m	4,5	NULL

Si se permiten valores NULL en id_tutor (llave foranea). Indica que la modalidad es mínimo 0 (opcional)

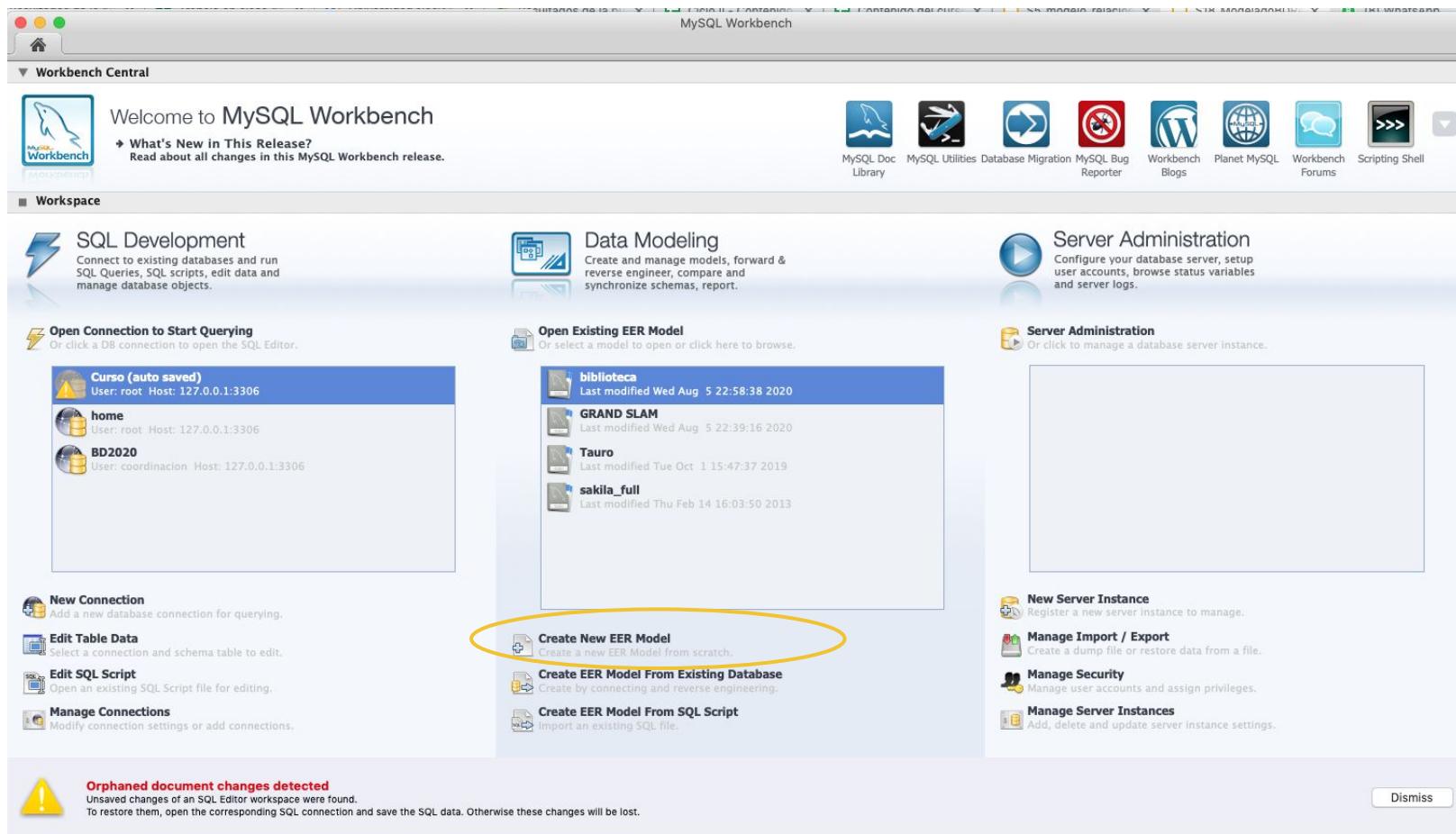
Ejercicio

Convertir a modelo Relacional el siguiente Diagrama E/R
Indicando llaves primarias y llaves foráneas

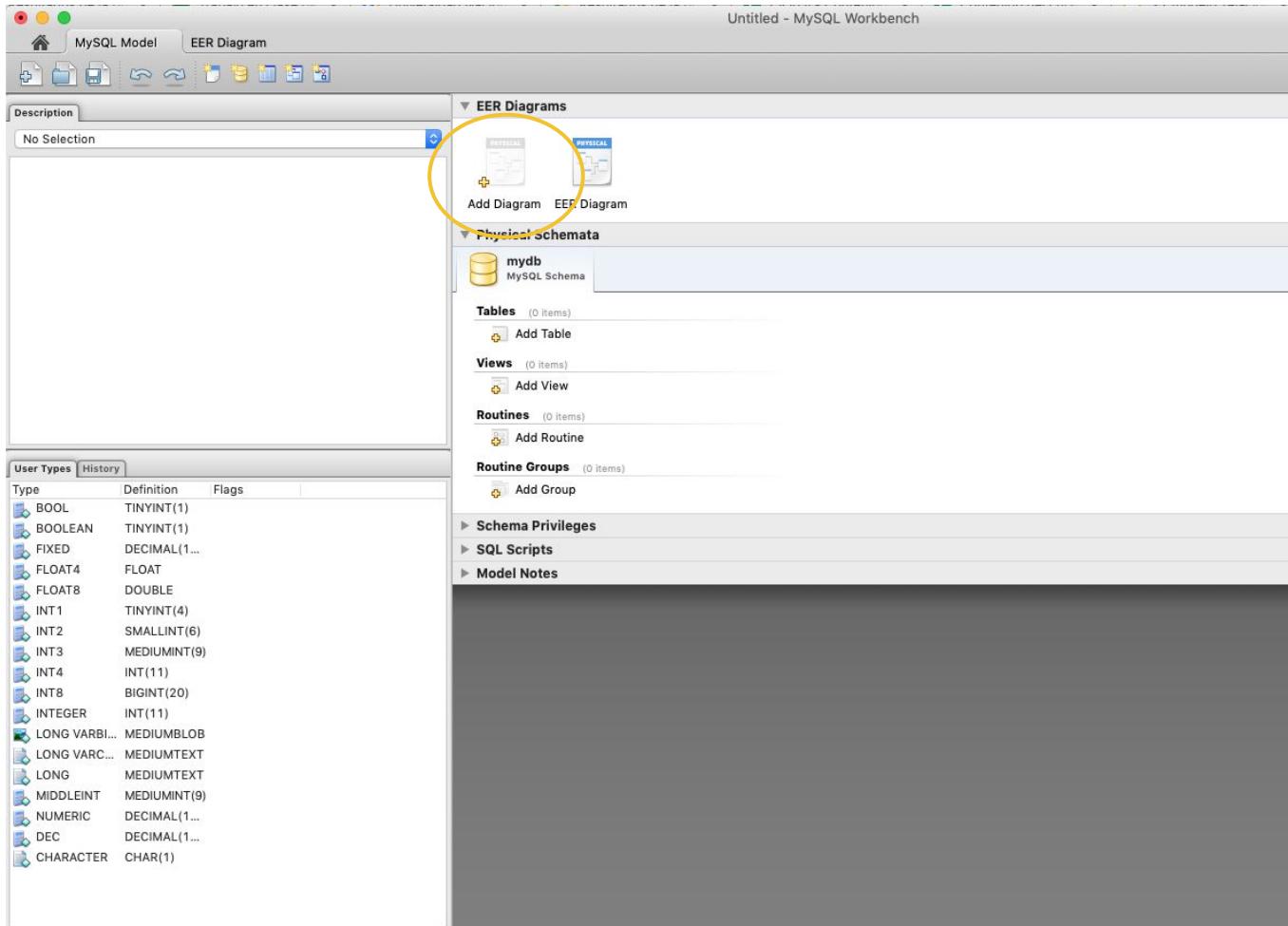


Ejercicio

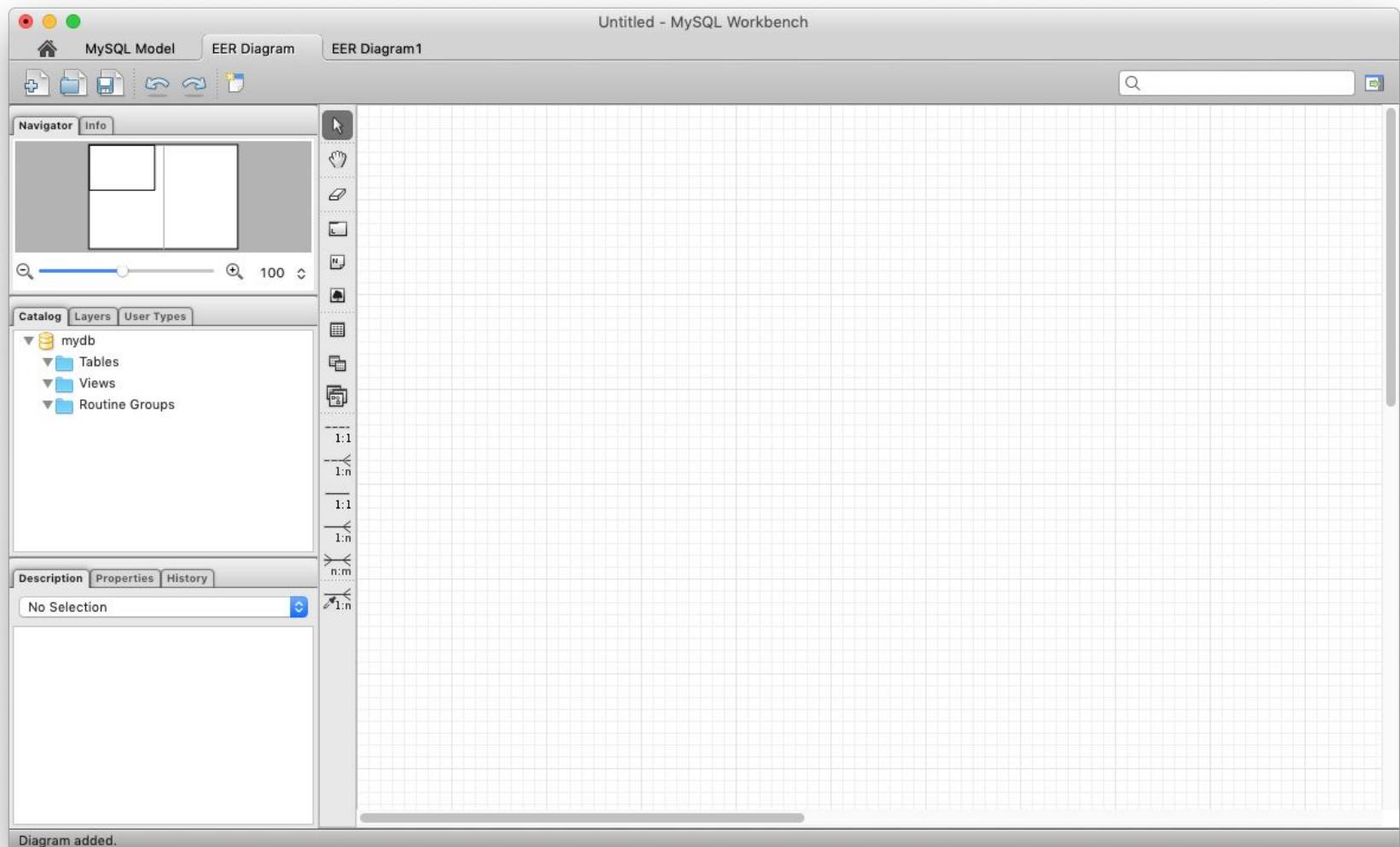
Entrar al workbench de MySQL. Crear un nuevo modelo EER



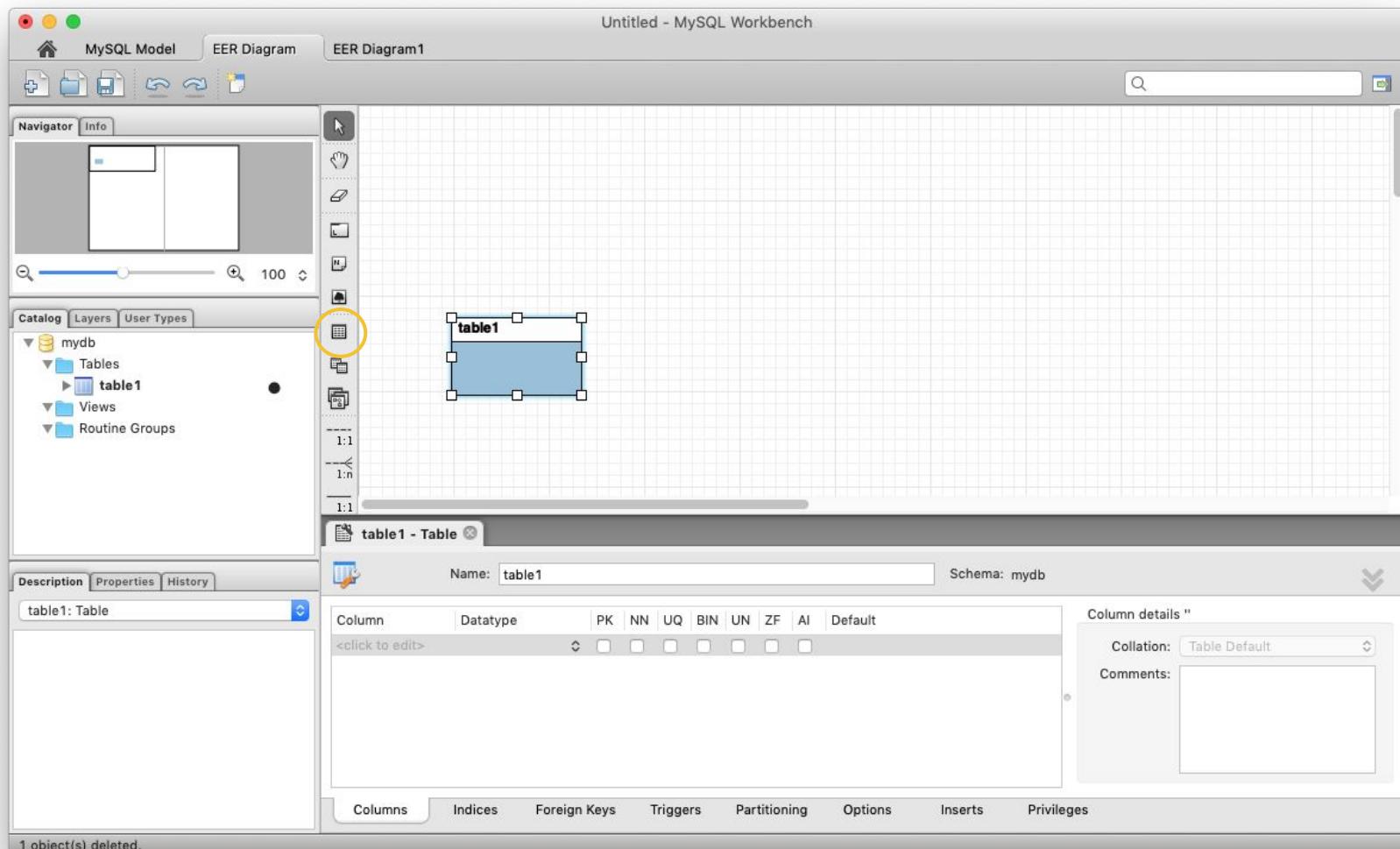
Ejercicio



Ejercicio



Ejercicio



Ejercicio

Untitled - MySQL Workbench

MySQL Model EER Diagram EER Diagram 1

Navigator Info

Catalog Layers User Types

mydb

Tables Libro

Views

Routine Groups

Libro

ISBN INT
titulo VARCHAR(45)
precio FLOAT

1:1
1:n
1:1

Libro - Table

Name: Libro Schema: mydb

table1: Table

Column Datatype PK UNN UC BIN UN ZF AI Default

ISBN INT ✓ ✓

titulo VARCHAR(45)

precio FLOAT

Column details 'precio'

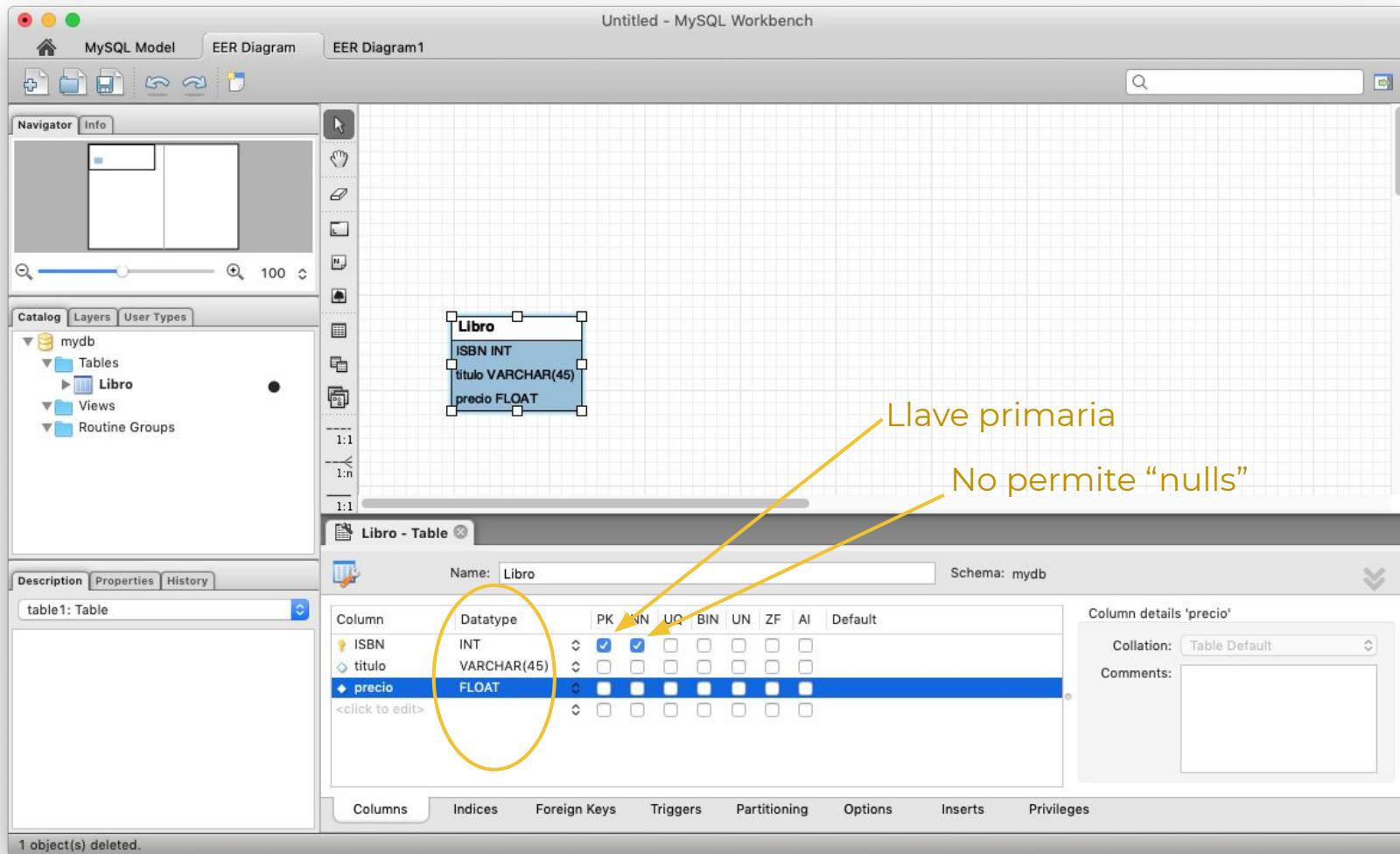
Collation: Table Default

Comments:

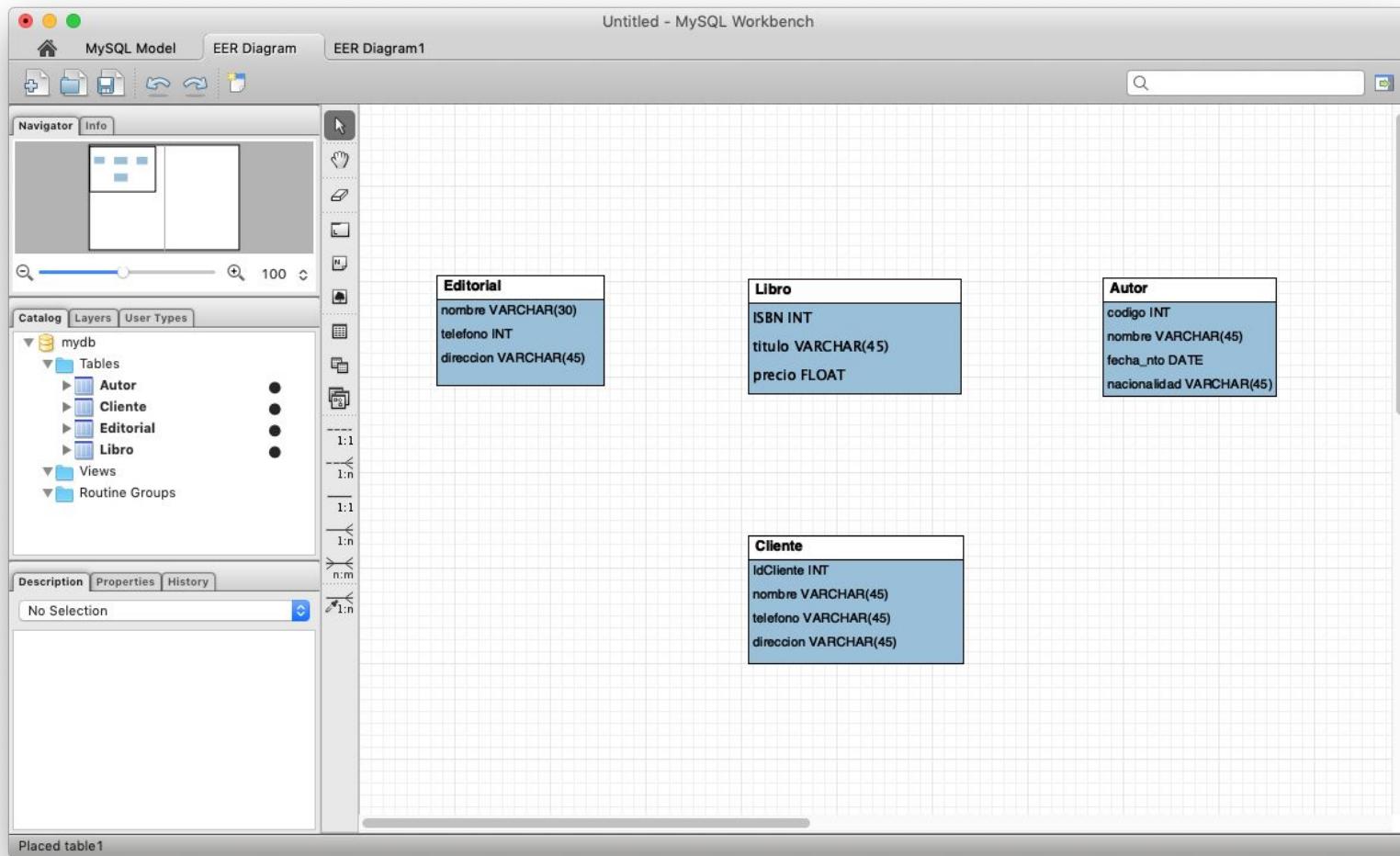
1 object(s) deleted.

Llave primaria

No permite "nulls"

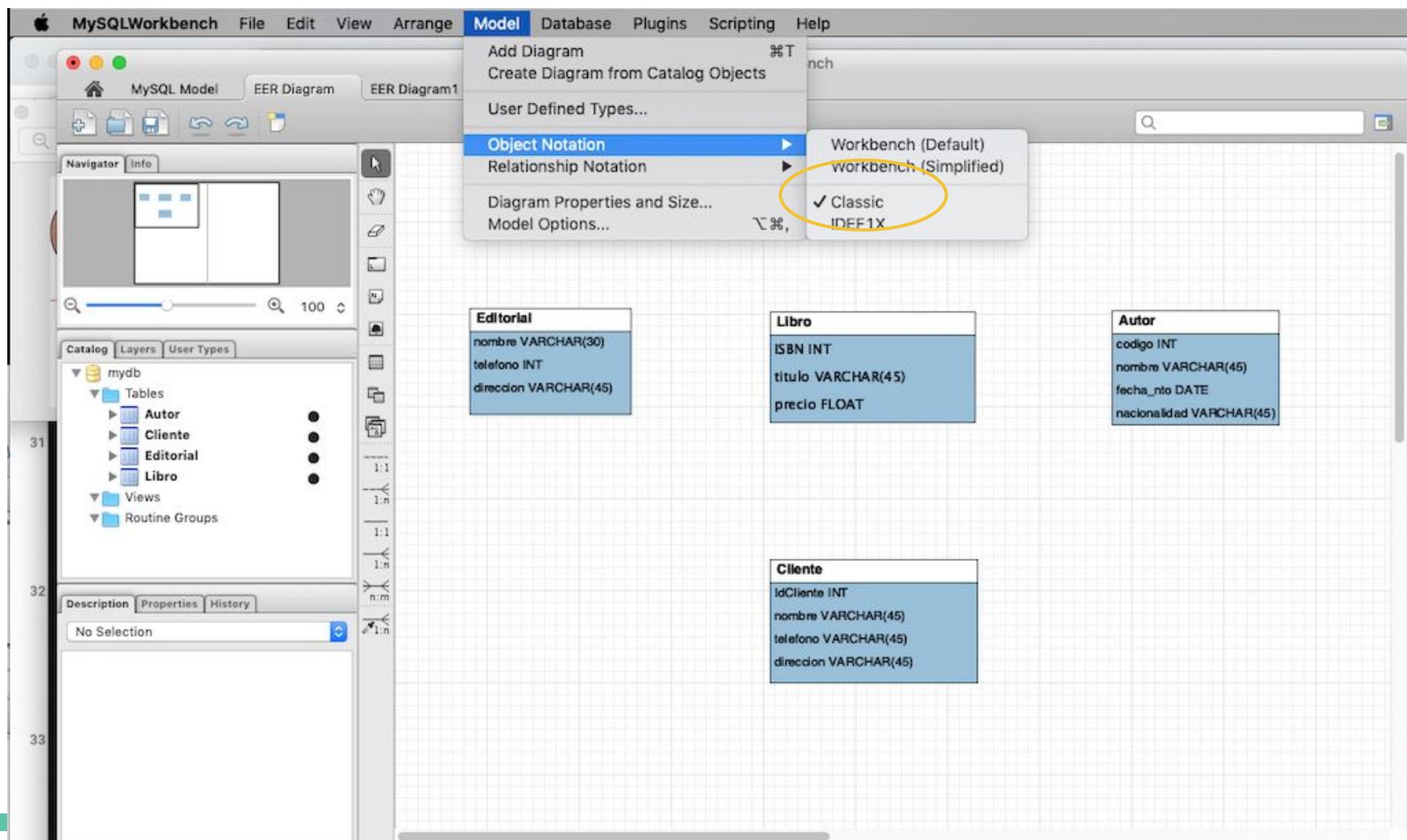


Ejercicio

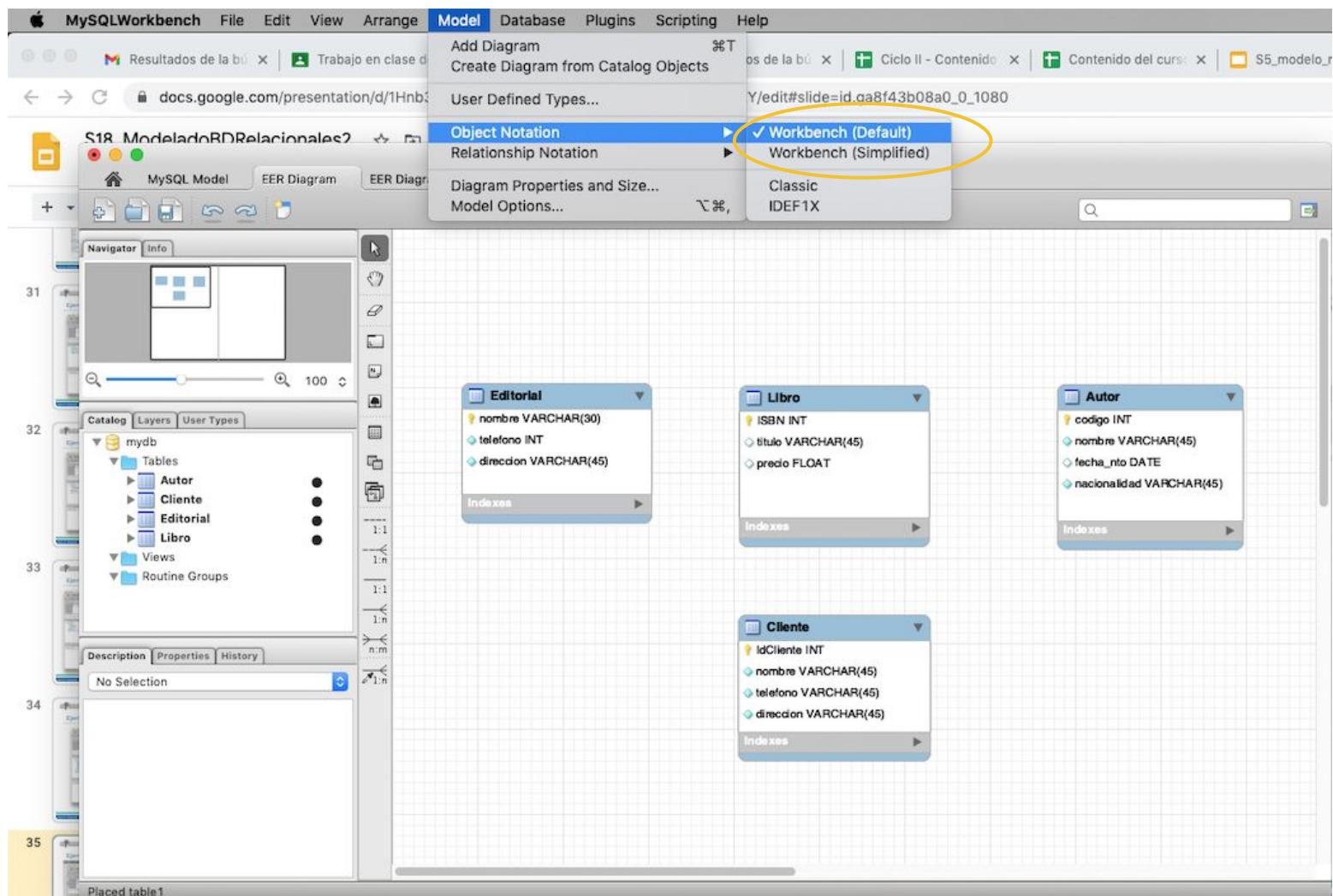


Ejercicio

Notación del modelo



Ejercicio



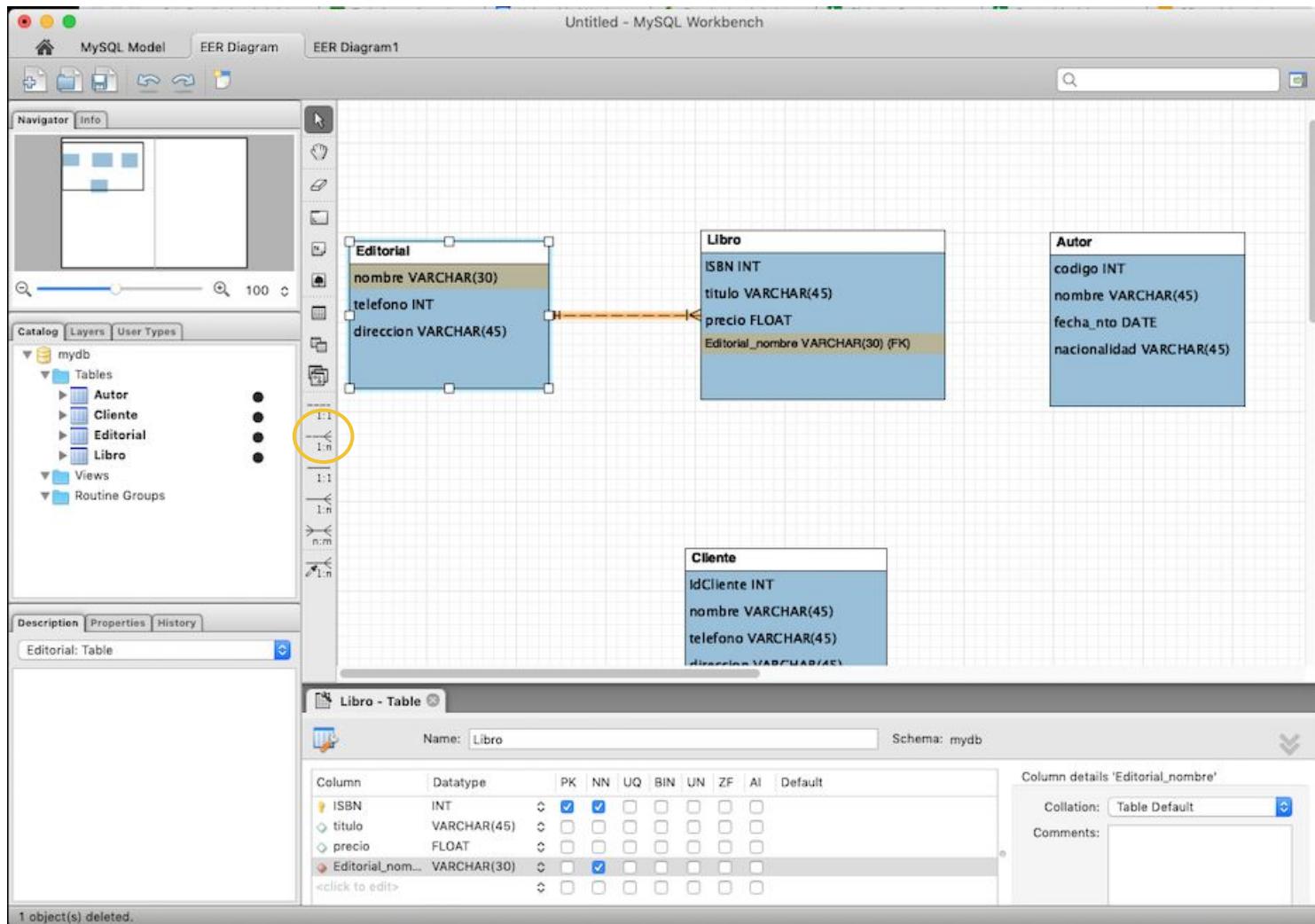
The screenshot shows the MySQL Workbench interface with an EER diagram titled "S18_ModeladoRDRelacionales2". The "Model" menu is open, displaying options for Object Notation, Relationship Notation, Diagram Properties and Size..., and Model Options... The "Object Notation" option is currently selected, and its submenu shows "Workbench (Default)" (which is checked) and "Workbench (Simplified)".

The EER diagram displays four tables:

- Editorial**: nombre VARCHAR(30), telefono INT, dirección VARCHAR(45)
- Libro**: ISBN INT, título VARCHAR(45), precio FLOAT
- Autor**: código INT, nombre VARCHAR(45), fecha_nacimiento DATE, nacionalidad VARCHAR(45)
- Cliente**: idCliente INT, nombre VARCHAR(45), teléfono VARCHAR(45), dirección VARCHAR(45)

The Navigator panel on the left shows the database structure under "mydb": Catalog, Layers, User Types, Tables (Autor, Cliente, Editorial, Libro), Views, and Routine Groups. The Properties panel at the bottom left shows "No Selection".

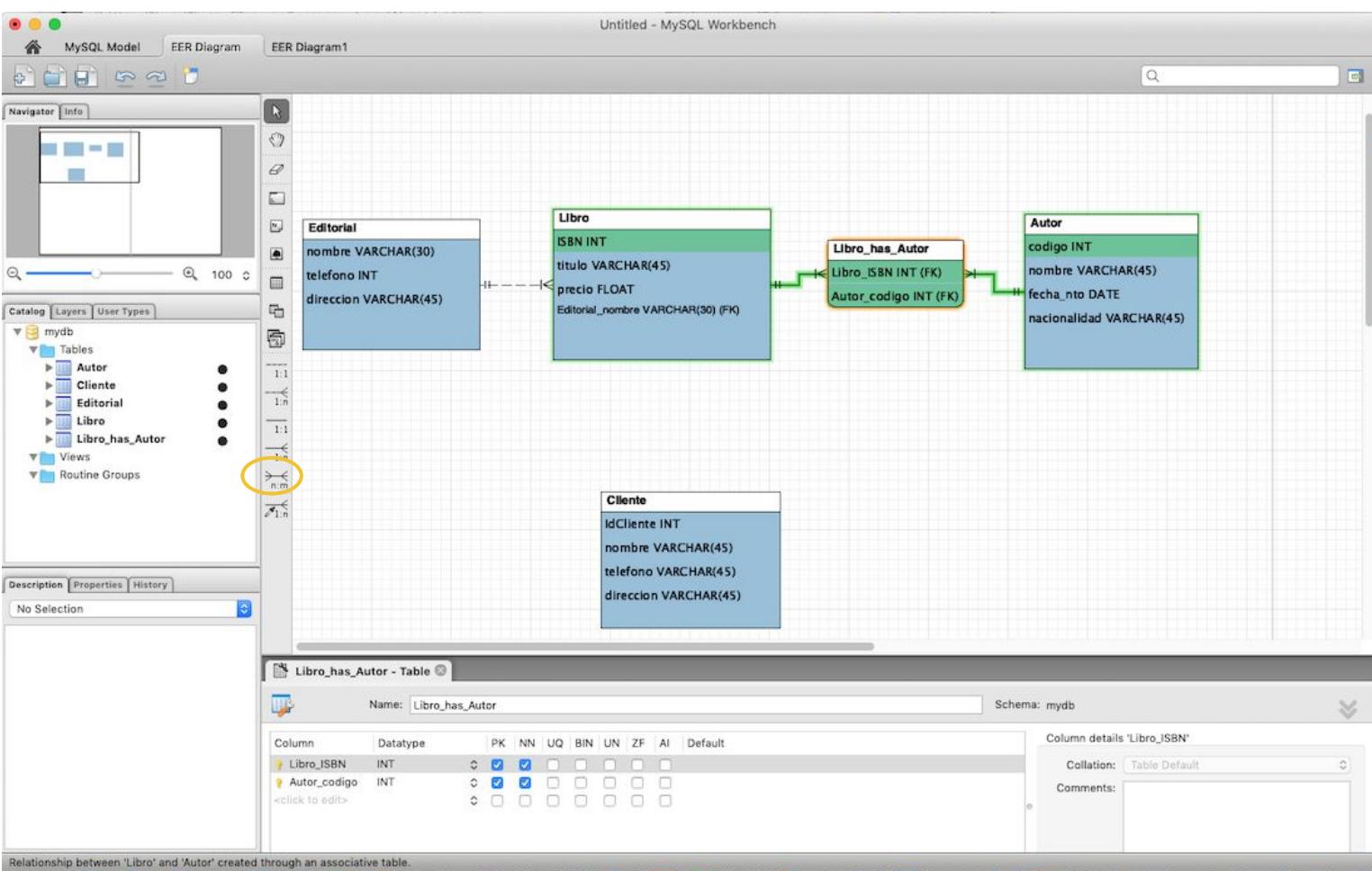
Ejercicio



Relación Uno a Muchos

1. Se selecciona la relación de uno a muchos punteada (no es débil)
2. Primero click en la relación Libro (cardinalidad muchos) y luego en Editorial (cardinalidad uno)
3. Automáticamente la llave primaria de editorial pasa a ser llave foránea de libro.

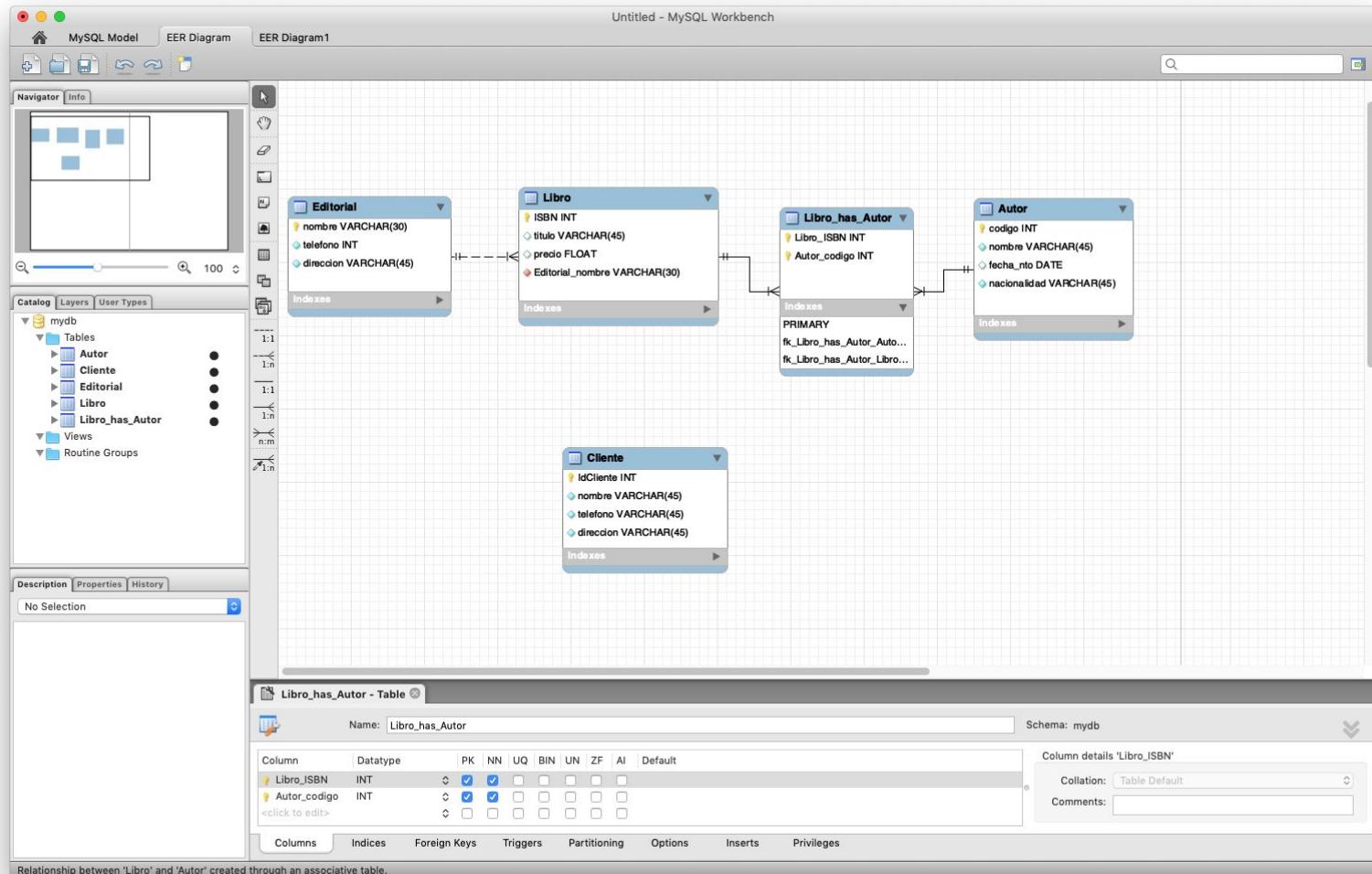
Ejercicio



Relación Muchos a Muchos

1. Se selecciona la relación de muchos a muchos
2. click en una de las entidades implicadas y luego en la otra
3. Automáticamente se creará la tabla que las relaciona. Tendrá como atributos las llaves primarias de las entidades, que a su vez son foráneas

Ejercicio



Con notación clásica.

No olvidar salvar el modelo. También se puede exportar como imagen (png)

Terminar el ejercicio. Crear la relación de muchos a muchos entre Cliente y Libro. Si lo considera puede crear una llave artificial para la compra de libros.

Normalización

Método para construir el modelo relacional, a partir de una relación, de forma que **evite redundancia de datos**

Objetivos:

- Mantener la Integridad de los datos.
- Eliminar información redundante siempre que sea posible.
- Mantener el número de relaciones al mínimo entre los componentes de la base de datos (fácil programación y uso por parte del usuario).
- Minimizar problemas de actualización y borrado.

Normalización

La Normalización es un proceso de descomposición o subdivisión de una relación en dos o más relaciones para evitar la redundancia.

También se puede ver como una serie de **reglas** que ayudan a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica.

Se puede utilizar para refinar las relaciones construidas a partir del diagrama E/R

Normalización

El proceso de normalización es definido por una serie de fases cada una con una serie de reglas específicas:

1FN, 2FN, 3FN, 4FN

Primera Forma Normal - 1FN

*Una relación está en primera forma normal si todo atributo contiene un valor **indivisible (atómico)**.*

Estudiante		
est_nombre	est_dirección	est_teléfono
Ana Díaz	Calle 50 # 30-20 Bogota, Colombia	(1)2307645, 311456289
Luis Avila	Cra 10 # 4-12 Medellín, Colombia	312432167 311876532

Se pueden dividir: No son atómicos

Primera Forma Normal - 1FN

est_nombre	est_apellido	est_calle	est_dirección	est_paia	est_teléfono
Ana	Díaz	Calle 50 # 30-20	Bogota	Colombia	(1)2307645
Ana	Díaz	Calle 50 # 30-20	Bogota	Colombia	311456289
Luis	Avila	Cra 10 # 4-12	Medellín	Colombia	312432167
Luis	Avila	Cra 10 # 4-12	Medellín	Colombia	311876532

Ahora se encuentra en 1FN. Pero presenta **anomalías de almacenamiento** por la datos **redundantes**.

Estás anomalías se deben a la presencia de campos no clave (llave) en la relación.

Primera Forma Normal - 1FN

Otro ejemplo

OrdenesVenta

ClienteID	Nombre	Localidad	CostoTransporte	ArtículoID	Artículo	Cantidad	Fecha
11	Luis	Suba	50.000	A1	Papel	100	3/5
11	Luis	Suba	50.000	A3	Cinta	50	5/5
11	Luis	Suba	50.000	A9	Lápiz	200	7/5
44	Ana	Centro	65.000	A1	Papel	100	10/5
55	José	Puente Aranda	70.000	A4	Grapas	30 50	3/5 5/5

Primera Forma Normal - 1FN

Otro ejemplo

OrdenesVenta

ClienteID	Nombre	Localidad	CostoTransporte	ArtículoID	Artículo	Cantidad	Fecha
11	Luis	Suba	50.000	A1	Papel	100	3/5
11	Luis	Suba	50.000	A3	Cinta	50	5/5
11	Luis	Suba	50.000	A9	Lápiz	200	7/5
44	Ana	Centro	65.000	A1	Papel	100	10/5
55	José	Puente Aranda	70.000	A4	Grapas	30	3/5
55	José	Puente Aranda	70.000	A4	Grapas	50	5/5

Segunda Forma Normal - 2FN

Las anomalías pueden subsanarse de la siguiente forma:

- Dividiendo la relación en nuevas relaciones.
- Eligiendo una **clave primaria** que represente de forma única a cada registro de las nuevas relaciones. Cada nueva relación tiene la propiedad de que su clave es necesaria para definir cada uno de los campos no clave.
- Para dividir las tablas se realiza un análisis de dependencias funcionales de los atributos

¡No perder información al dividir la relación!

Dependencia Funcional

La Normalización se basa en la dependencia funcional:

Dependencia funcional

Dados dos atributos A y B de una relación R, se dice que **B es funcionalmente dependiente de A** si para cada valor de A existe un valor de B, y sólo uno, asociado con él.

Se denota como

$$A \longrightarrow B$$

Estudiante		
Código	Nombre	Apellido
253420	Juan	Ruiz
254210	Maria	López
255820	Juan	Diaz
263100	Luisa	Avila

Código \longrightarrow Nombre

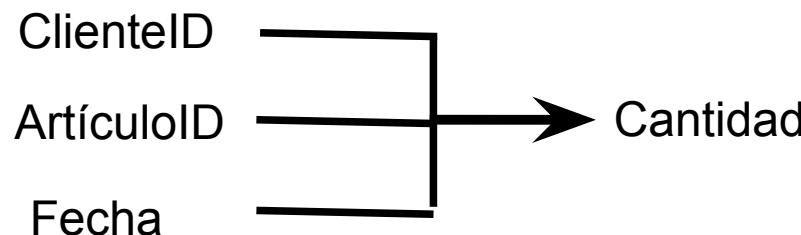
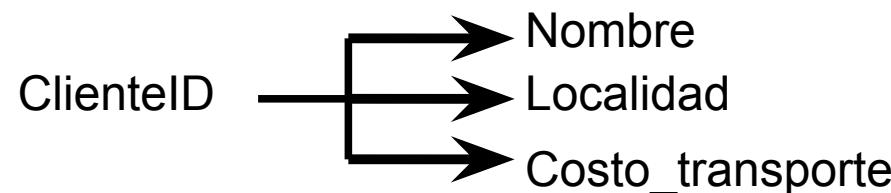
Código \longrightarrow Apellido

Nombre y apellido dependen funcionalmente de Código

Análisis de Dependencias Funcionales

Ejemplo

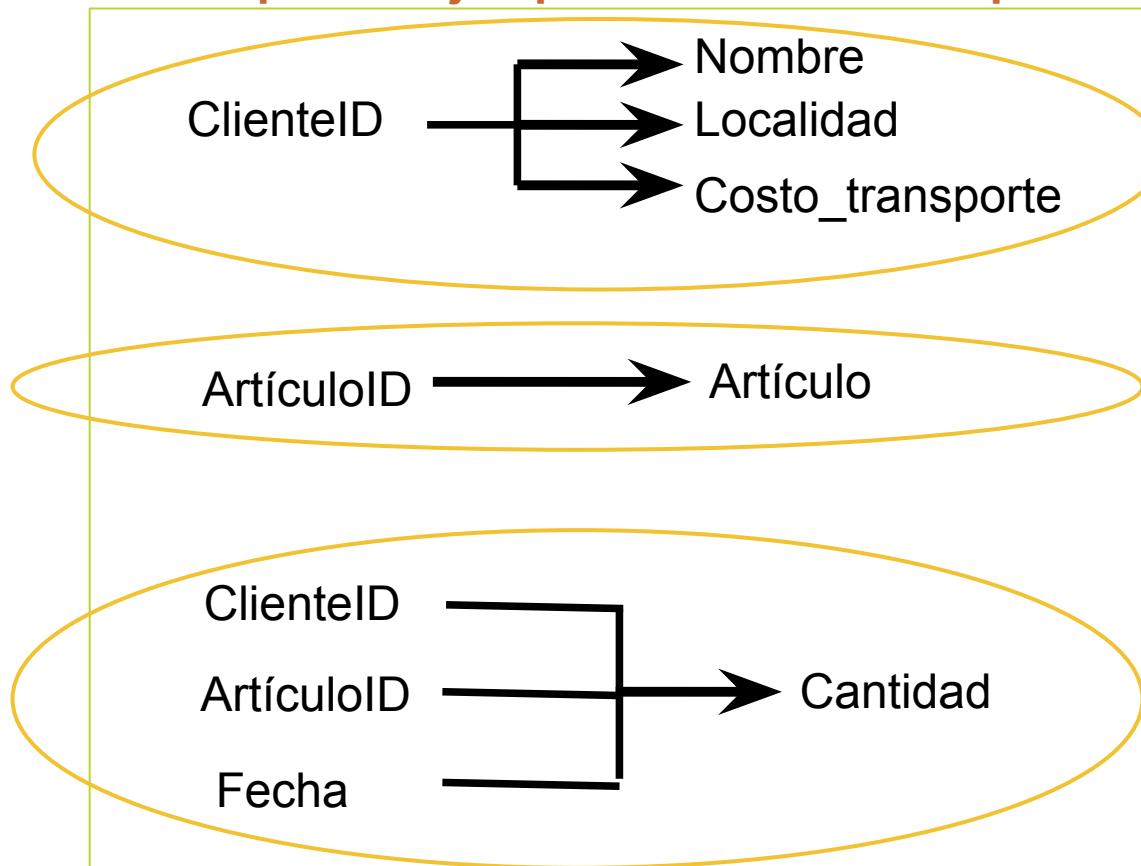
Análisis dependencia funcional para el ejemplo de OrdenesCompra



Análisis de Dependencias Funcionales

Ejemplo

Análisis dependencia funcional para el ejemplo de OrdenesCompra



El análisis indica que la tabla original se debe dividir en tres (3) nuevas tablas/relaciones

Segunda Forma Normal - 2FN

Relación CLIENTE

<u>ClienteID</u>	Nombre	Localidad	CostoTransporte
11	Luis	Suba	50.000
44	Ana	Centro	65.000
55	José	Puente Aranda	70.000

Relación VENTA

<u>ClienteID</u>	<u>ArtículoID</u>	Cantidad	Fecha
11	A1	100	3/5
11	A3	50	5/5
11	A9	200	7/5
44	A1	100	10/5
55	A4	30	3/5
55	A4	50	5/5

Relación ARTÍCULO

<u>ArtículoID</u>	Artículo
A1	Papel
A3	Cinta
A4	Grapas
A9	Lápiz

Segunda Forma Normal - 2FN

Una relación está en segunda forma normal si, y sólo si:

- Está en **1FN**
- Todo atributo que no pertenezca a la clave debe depender de la clave en su **totalidad**. Es decir, los registros no deben depender de nada aparte de su clave primaria.

Dependencia transitiva

Ejemplo

Las relaciones CLIENTES, ARTICULOS y VENTAS estan en 2FN. Sin embargo, la relación CLIENTE presenta **anomalías** de almacenamiento debido a que el atributo **CostoTransporte** es funcionalmente dependiente de la **Localidad**, que a su vez depende de **ClienteID**.

Hay una **dependencia transitiva** que ocasiona problemas a la hora de hacer actualizaciones

Relación CLIENTE

ClienteID	Nombre	Localidad	CostoTransporte
11	Luis	Suba	50.000
44	Ana	Centro	65.000
55	José	Puente Aranda	70.000

Tercera Forma Normal - 3FN

Una relación está en 3FN si, y sólo si:

- Está en 2FN.
- Todo atributo que no pertenezca a la clave no depende de un atributo no clave.

La 3FN elimina las redundancias ocasionadas por las **dependencias transitivas**.

Tercera Forma Normal - 3FN

- Aplicando la 3FN a la relación CLIENTES

Relación CLIENTE

<u>ClienteID</u>	Nombre	<u>Id_Localidad</u>
11	Luis	1
44	Ana	2
55	José	3

Relación TRANSPORTE

<u>Id_Localida</u>	Localidad	CostoTransporte
1	Suba	50.000
2	Centro	65.000
3	Puente Aranda	70.000

BCFN - Forma Normal de Boyce Codd

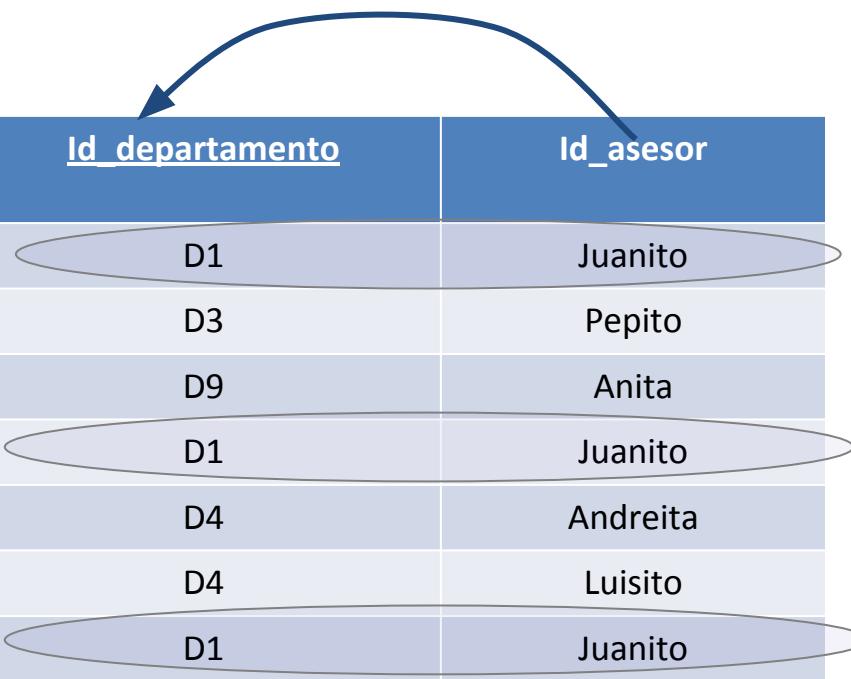
<u>Id_empleado</u>	<u>Id_departamento</u>	<u>Id_asesor</u>
11	D1	Juanito
11	D3	Pepito
11	D9	Anita
44	D1	Juanito
44	D4	Andreita
55	D4	Luisito
55	D1	Juanito



Un atributo de la llave depende de un atributo no llave (determinante)

BCFN - Forma Normal de Boyce Codd

<u>Id_empleado</u>	<u>Id_departamento</u>	<u>Id_asesor</u>
11	D1	Juanito
11	D3	Pepito
11	D9	Anita
44	D1	Juanito
44	D4	Andreita
55	D4	Luisito
55	D1	Juanito



Un atributo de la llave depende de un atributo no llave (determinante)

¡redundancia!

BCFN

<u>Id_asesor</u>	<u>Id_departamento</u>
Juanito	D1
Pepito	D3
Anita	D9
Andreita	D4
Luisito	D4

<u>Id_empleado</u>	<u>Id_asesor</u>
11	Juanito
11	Pepito
11	Anita
44	Juanito
44	Andreita
55	Luisito
55	Juanito

Cuarta Forma Normal - 4FN

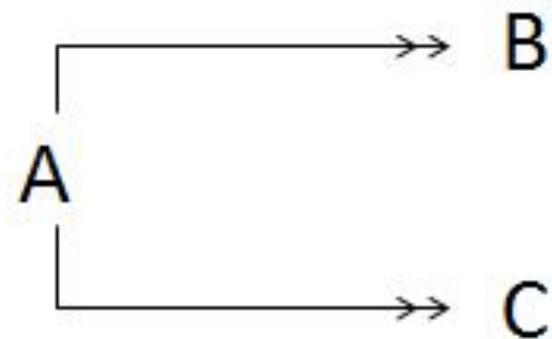
La 4FN es una generalización de la BCFN para descomponer relaciones que posean dependencias multivaluadas.

Una relación R está en 4FN si, y sólo si:

- Es BCFN
- No contiene dependencias multivaluadas

Dependencia multivaluada

La relación $R(\underline{A},B,C)$ con las dependencias multivaluadas:



Se puede descomponer sin pérdida en dos relaciones 4FN: $R1(\underline{A},B)$ y $R2(\underline{A},C)$.

Dependencia Multivaluada

Ejemplo

La relación estudiantes indica el código del estudiante, el curso donde está matriculado y los deportes que practica.

ESTUDIANTE		
<u>CódigoEst</u>	Curso	Deporte
1	Bases de Datos	Baloncesto
1	Bases de Datos	Natación
2	Bases de Datos	Tenis
3	Física	Baloncesto
3	Física	Esgrima

Los atributos *Curso* y *Deporte* son dependientes multivalentes de *CódigoEst*, es decir, cualquier valor de *CódigoEst* determina una serie de valores de los atributos *Curso* y *Deporte*.

Cuarta Forma Normal - 4FN

InscripcionesCurso

<u>CodigoEst</u>	Curso
1	Base de datos
2	Base de datos
3	Física

PracticaDeporte

<u>CodigoEst</u>	Deporte
1	Baloncesto
1	Natación
2	Tenis
3	Baloncesto
3	Esgrima

Ejercicio

Construir un modelo relacional a partir de esta relación

Fecha_presentacion	Id_museo	Nom_museo	Id_expo	Nom_expo	Id_obra	Nom_obra	Tipo_obra	Costo
Ene-mar-2019, Oct-dic-2019	101	Louvre	1003	Da Vinci	111	Mona lisa	Pintura	\$1000
Ene-mar-2019, Oct-dic-2019	101	Louvre	1003	Da Vinci	112	Ultima cena	Pintura	\$800
May-sept-2019	101	Louvre	1004	Renacimiento	200	Fornarina	Pintura	\$400
May-sept-2019	101	Louvre	1004	Renacimiento	201	David	Escultura	\$700
Oct-dic-2019	101	Louvre	1003	Da Vinci	113	Hombre vitruvio	Boceto	\$400
Ene-jul-2019	102	Met	1005	Cubismo	300	Violin and candlestick	Pintura	\$300
Ene-jul-2019	102	Met	1005	Cubismo	301	Les demoiselles d'avignon	Pintura	\$350
Abr-jun-2019, Sept-2019	102	Met	1003	Da Vinci	113	Hombre vitruvio	Boceto	\$400
Abr-jun-2019, Sept-2019	102	Met	1003	Da Vinci	112	Ultima cena	Pintura	\$800
Ene-abr-2019	102	Met	1004	Renacimiento	202	Nacimiento venus	Pintura	\$250
Ene-abr-2019	102	Met	1004	Renacimiento	201	David	Escultura	\$700
jul.-2019	203	Shangai	1003	Da Vinci	113	Hombre vitruvio	Boceto	\$400
jul.-2019	203	Shangai	1003	Da Vinci	114	Planos	Planos	\$200
Marz-jun-2019	203	Shangai	1006	Impresionismo	401	La parade	Pintura	\$300
Marz-jun-2019	203	Shangai	1006	Impresionismo	402	Solei levant	Pintura	\$300
Oct-dic-2019	203	Shangai	1005	Cubismo	301	Les demoiselles d'avignon	Pintura	\$350
Oct-dic-2019	203	Shangai	1005	Cubismo	302	Cabeza de mujer	Escultura	\$300
Oct-dic-2019	203	Shangai	1004	Renacimiento	200	Fornarina	Pintura	\$400
Oct-dic-2019	203	Shangai	1004	Renacimiento	202	Nacimiento venus	Pintura	\$250
Ago-nov-2019	304	Británico	1006	Impresionismo	402	Solei levant	Pintura	\$300
Ago-nov-2019	304	Británico	1006	Impresionismo	400	Autorretrato	Pintura	\$100
Jul-agosto-2019	304	Británico	1003	Da Vinci	114	Planos	Planos	\$200
Jul-agosto-2019	304	Británico	1003	Da Vinci	111	Mona lisa	Pintura	\$1000

Referencias

- [1] Gillenson, M. *Administración de Bases de Datos*. LIMUSA WILEY (Cap 1 y 2)
- [2] Coronel, Morris, Rob. *Bases de Datos: Diseño, Implementación y Administración*. CENGAGE Learning
- [3] Elmasri, R.; Navathe, S.B. *Fundamentos de Sistemas de Bases de Datos*. 3^a ed. Addison-Wesley, (Cap. 3 y 4)
- [4] Silberschatz, A;Korth, H; Sudarshan, S. *Fundamentos de Bases de Datos*. 3^a edición. Madrid: McGraw-Hill. (Cap. 2)
- [5] León, E. Notas curso Bases de Datos. Universidad Nacional de Colombia