

Bases de Datos

Modelo relacional

- Modelo Relacional
 - ✧ Definición de relación
 - ✧ Restricciones de integridad

Diseño de bases de datos relacionales

Conversión del modelo Entidad-Asociación o Entidad-Relación al modelo Relacional

Modelo relacional

■ Bibliografía

Elmasri, R., Navathe y S. (2000). Fundamentos de Sistemas de Bases de Datos. España: Pearson Addison-Wesley.

Miguel, A., Piattini M. y Marcos, E. (2000). *Diseño de bases de datos relacionales*. México: Alfaomega:Ra-Ma.

Marqués, M. (2001). Apuntes de ficheros y bases de datos. Universitat Jaume I, Campus de Riu Sec. España. consultado el 29 enero 2017, en:
https://www.researchgate.net/publication/41209120_Apuntes_de_Ficheros_y_Bases_de_Datos

Modelo Relacional

Según Elmasri y Navathe, un **modelo de datos** es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos.

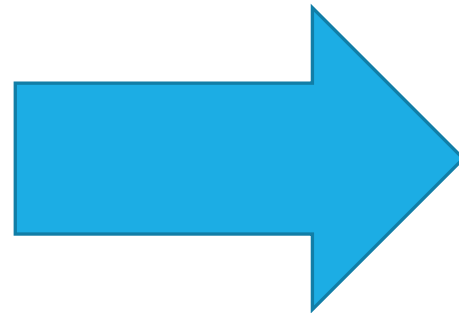
Modelo Relacional

El modelo relacional es un modelo de datos que da estructura a una base de datos mediante tablas con filas y columnas.

Modelo relacional

"Su objetivo es transformar el esquema conceptual obtenido en la etapa anterior, adaptándolo al modelo de datos en el que se apoya el SMDB (modelo relacional en nuestro caso) que se va a utilizar."
(Miguel y Piattini, 1999).

Modelo de datos



Modelo relacional

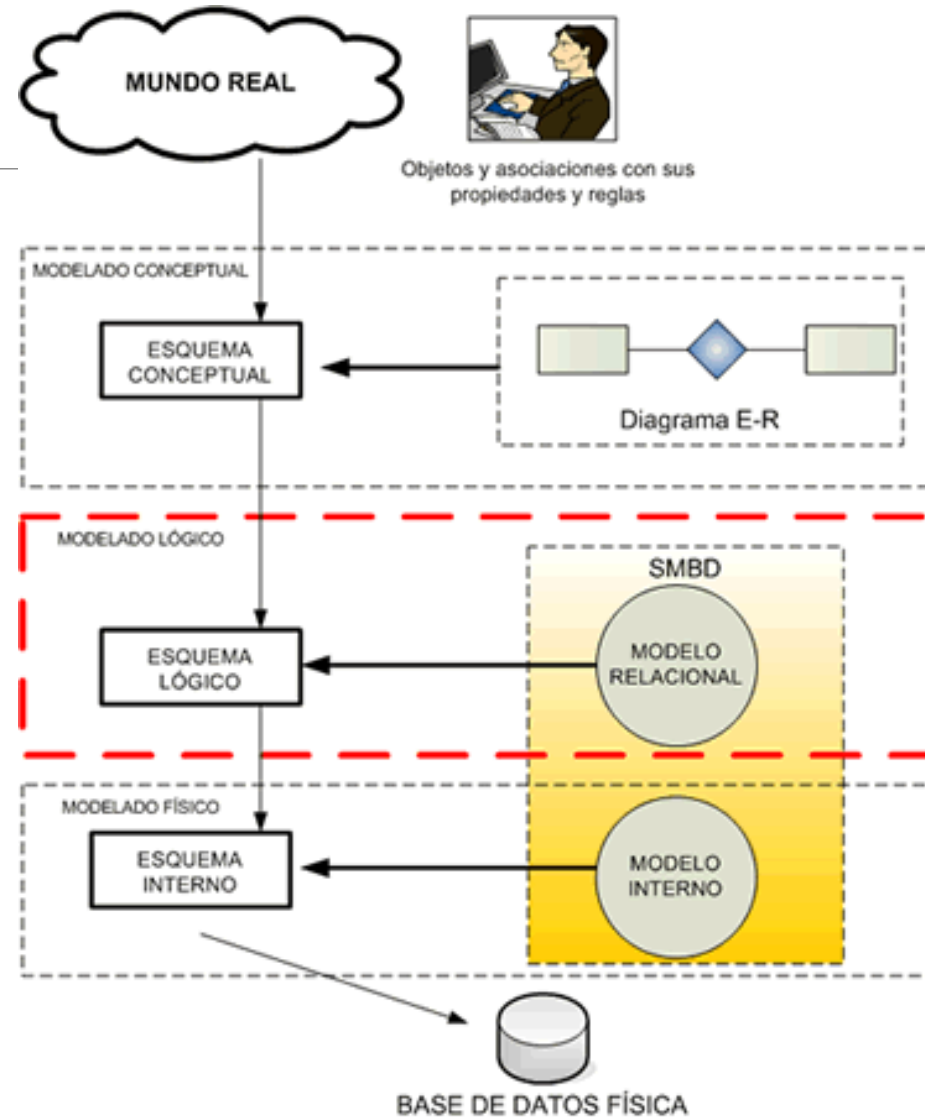
Modelo Relacional

“El modelo relacional usa una colección de tablas para representar tanto los datos como sus relaciones. Cada tabla tiene varias columnas, y cada columna tiene un nombre único. El modelo relacional es un ejemplo de un modelo basado en registros...”
(Silberschatz, Korth, & Sudarshan, 2006)

Modelo relacional

"El modelo relacional fue presentado por E. F. Codd en 1970 en su célebre artículo de ACM titulado 'Un modelo de datos relacional para grandes bancos de datos compartidos', constituyó un hito en la historia de las bases de datos." (Miguel, A. et. al., 2000).

Modelo relacional



Modelo relacional

*Una **relación** es una tabla con columnas y filas que sirve para representar tanto los objetos como las asociaciones entre ellos (relaciones del modelo entidad-relación) (Marqués, 2001).*

*Un **atributo** es el nombre de una columna de una relación y representan a los atributos del modelo entidad-relación.*

Modelo relacional

El ***Dominio*** permite que el usuario defina, el significado y la fuente de los valores que los atributos pueden tomar.

*Una **tupla** es una fila de una relación.*

*La **cardinalidad** de una relación es el número de tuplas que contiene.*

*El **grado** de una relación, es el número de atributos que contiene.*

*Una **base de datos relacional** es un conjunto de relaciones normalizadas*

La estructura relacional

- **Relación** → Corresponde con la idea general de tabla
- **Tupla** → Corresponde con una fila
- **Atributo** → Corresponde con una columna
- **Cardinalidad** → Número de tuplas (m)
- **Grado** → Número de atributos (n)
- **Clave primaria** → Identificador único (no hay dos tuplas con igual identificador)
- **Dominio** → Colección de valores de los cuales el atributo obtiene su valor



Términos importantes en la estructura de datos relacional:

- **Relación:** Película (título, año, duración)

Atributos	Título	Año	Duración
Tuplas	La guerra de las galaxias	1977	123
	El señor de los anillos I	2001	178
	Mar adentro	2004	125

↓

Dominio=textos

↘ ↙

Dominio=enteros

Cardinalidad=3

Grado de la relación=3

Modelo relacional

Esquema de una relación

CURSO(Cod_Curso, Nombre, Num_Horas, Materia)

Instancias de una relación

Cod_Curso	Nombre	Num_Horas	Materia
00012	DISEÑO DE BASES DE DATOS	50	Bases de Datos
00034	BASES DE DATOS ORIENTADAS A OBJETOS	30	Sis. Operativos
00167	SISTEMAS OPERATIVOS AVANZADOS	30	Bases de Datos
01521	ALMACENES DE DATOS	25	Bases de Datos
005142	INTRODUCCIÓN A C++	25	Lenguajes
005143	INTRODUCCIÓN A C++	25	Lenguajes

Modelo relacional

Propiedades de las relaciones

- Cada relación tiene un nombre y éste es distinto del nombre de todas las demás.
- Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están *normalizadas*.
- No hay dos atributos que se llamen igual.
- El orden de los atributos no importa: los atributos no están ordenados.
- Cada tupla es distinta de las demás: no hay tuplas duplicadas.
- El orden de las tuplas no importa: las tuplas no están ordenadas.

Restricciones de integridad

Restricciones de Integridad

Qué son las restricciones de integridad

- ❖ Son condiciones que garantizan que las modificaciones realizadas en la base de datos por los usuarios autorizados no den lugar a una pérdida de la consistencia de los datos.
- ❖ Protegen contra daños accidentales a las bases de datos.



Restricciones de Integridad

Qué son las restricciones de integridad

- ❖ **Consistencia:**
Coherencia entre todos los datos de la base de datos.
- ❖ **Integridad:** *Es la corrección y exactitud de la información contenida en la base de datos.*



Restricciones de Integridad

- ❖ Ejemplos de restricciones de integridad:
 - El nombre de un alumno no puede ser Nulo.
 - No se permite tener dos alumnos con la misma matrícula.
 - Todos los ID de clientes de la relación *Cliente*, deben tener un número ID cliente correspondiente en la relación *Reservacion*.

Restricciones de Integridad

❖ Algunas restricciones sobre una sola relación.

- ✓ Valores no nulos (Not null)
- ✓ Llaves primarias y Llaves foráneas (Primary key, Foreign key)

Restricciones de Integridad

❖ Valores No nulos (Not null)

- Prohíbe la inserción de valores nulos para un atributo.
- Cualquier modificación de la base de datos que haga que se inserte un valor nulo en un atributo declarado como **not null** genera un diagnóstico de error.
- SQL prohíbe los valores nulos en la clave primaria.

Restricciones de Integridad

❖ Llave primaria (Primary key)

- Identifica de forma única cada tupla o registro de una relación.

```
CREATE TABLE Estudiante  
(matricula integer,  
ap_paterno varchar(30),  
ap_materno varchar(30),  
PRIMARY KEY (matricula));
```

Restricciones de Integridad

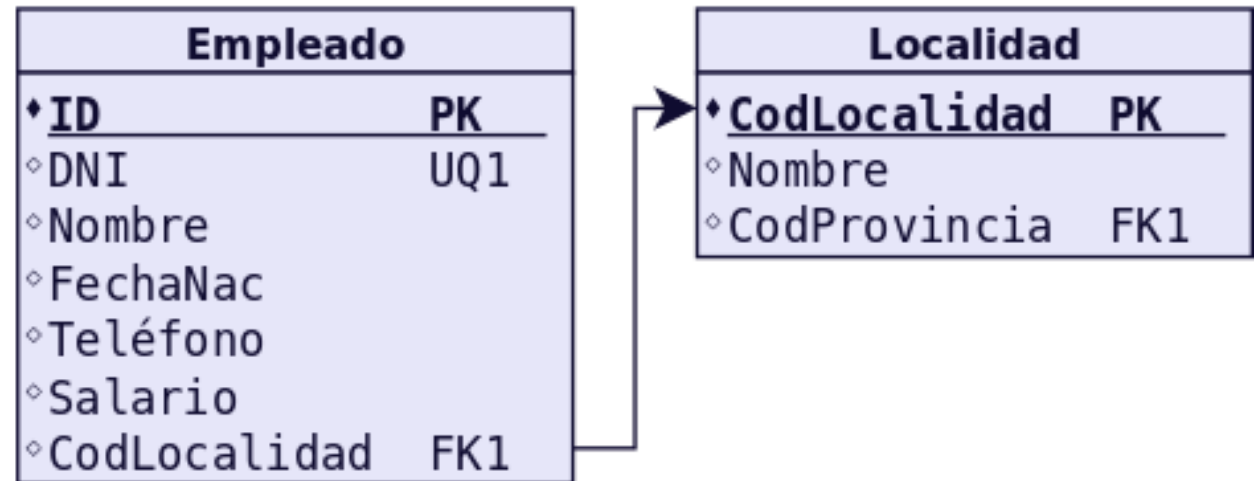
❖ INTEGRIDAD REFERENCIAL.

Tiene el objetivo de garantizar que el valor que aparece en una relación para un atributo aparezca para otro atributo en otra relación.

Restricciones de Integridad

❖ INTEGRIDAD REFERENCIAL.

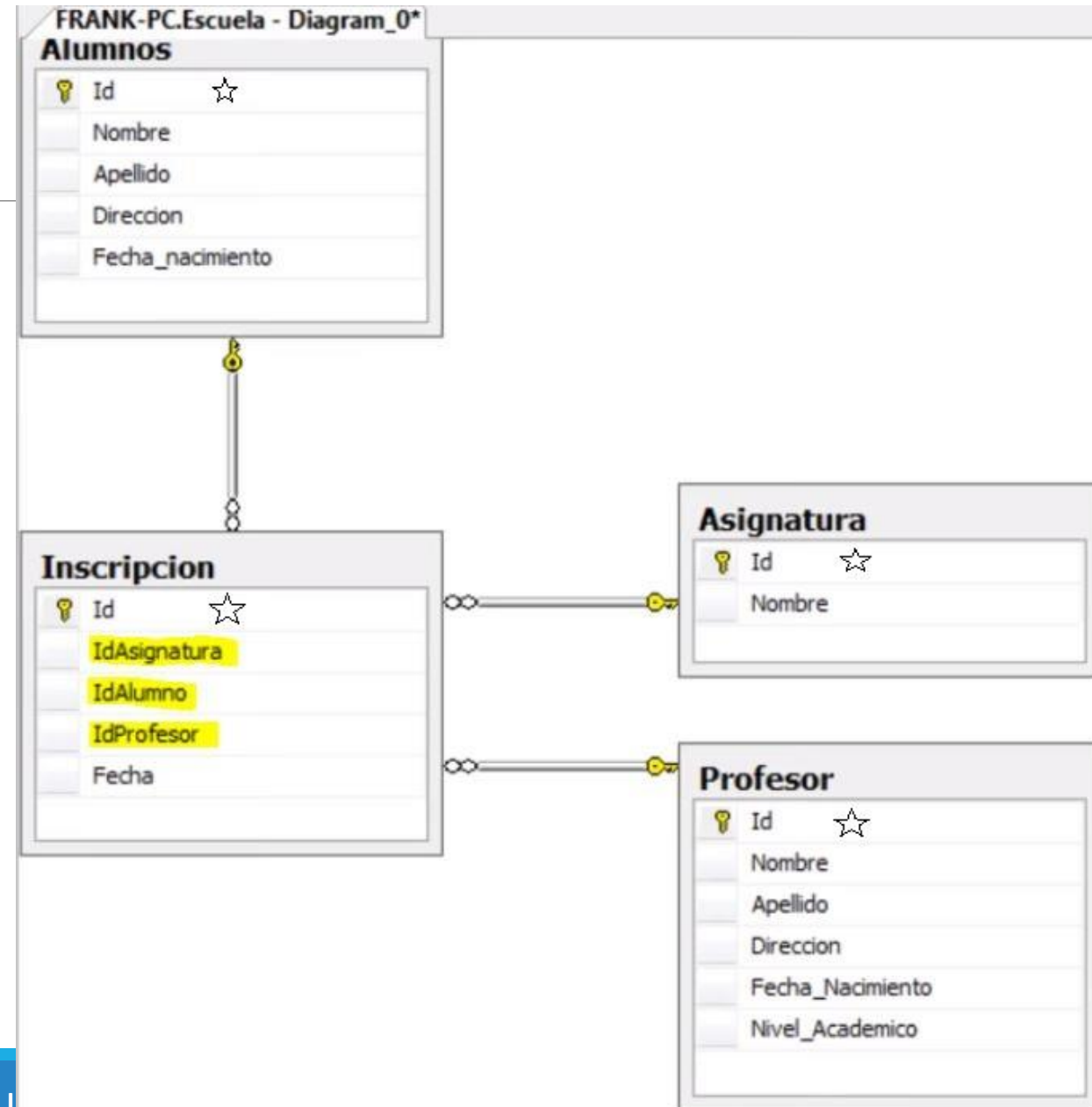
CLAVE FORÁNEA



Restricciones de Integridad

❖ INTEGRIDAD REFERENCIAL.

CLAVE FORÁNEA



Diseño de bases de datos relacionales

Conversión del modelo Entidad
Asociación al modelo Relacional

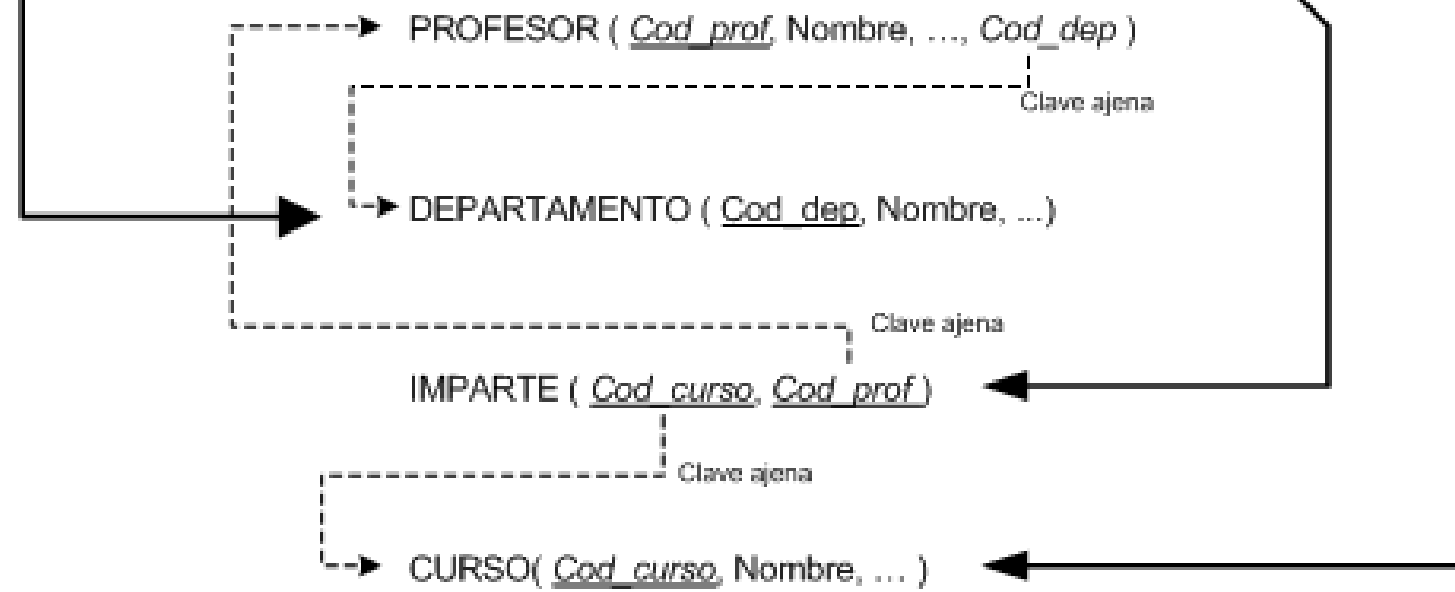
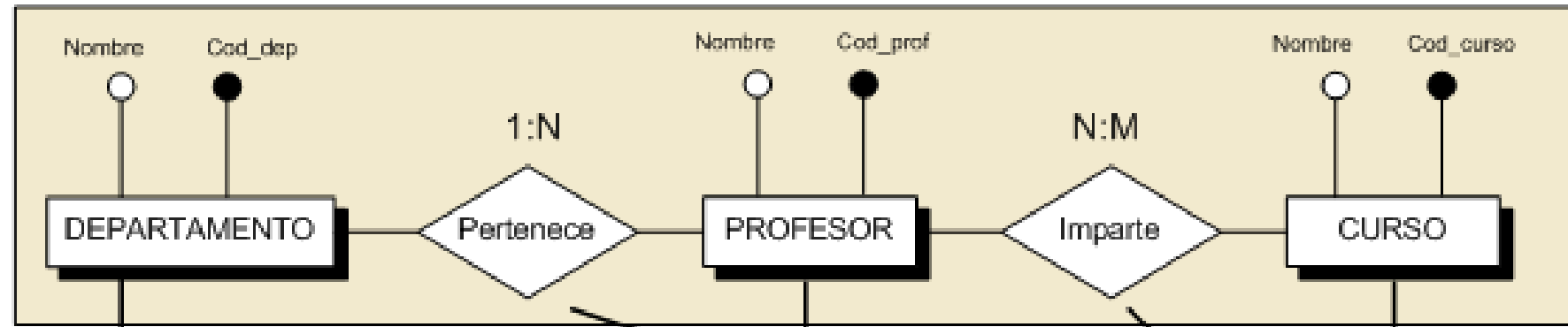
Modelo relacional

Convertir el esquema conceptual en un esquema lógico.

Las tres reglas básicas para convertir un esquema en el modelo entidad-relación al relacional son las siguientes (Miguel, A. et. al., 2000):

- i. Todo tipo de entidad se convierte en una relación.
- ii. Toda relación N:M se transforma en una relación
- iii. Para toda relación 1:N se realiza lo que se denomina *propagación de clave* (regla general).

Esquema Conceptual



Esquema relacional

Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - a) Eliminar las relaciones de muchos a muchos
 - b) Eliminar las relaciones entre tres o más entidades
 - c) Eliminar las relaciones recursivas
 - d) Eliminar las relaciones con atributos
 - e) Eliminar los atributos multivaluados
 - f) Revisar las relaciones de uno a uno
 - g) Eliminar las relaciones redundantes.

Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - a) Eliminar las relaciones de muchos a muchos

Eliminar las relaciones de muchos a muchos, sustituyendo cada una de ellas por una nueva entidad intermedia.

La nueva entidad será débil, ya que sus ocurrencias dependen de la existencia de ocurrencias en las entidades originales.

Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - b) Eliminar las relaciones entre tres o más entidades

Eliminar las relaciones entre tres o más entidades, sustituyéndola por una nueva entidad (débil) intermedia que se relaciona con cada una de las entidades originales.

La cardinalidad de estas nuevas relaciones binarias dependerá de su significado.

Modelo relacional

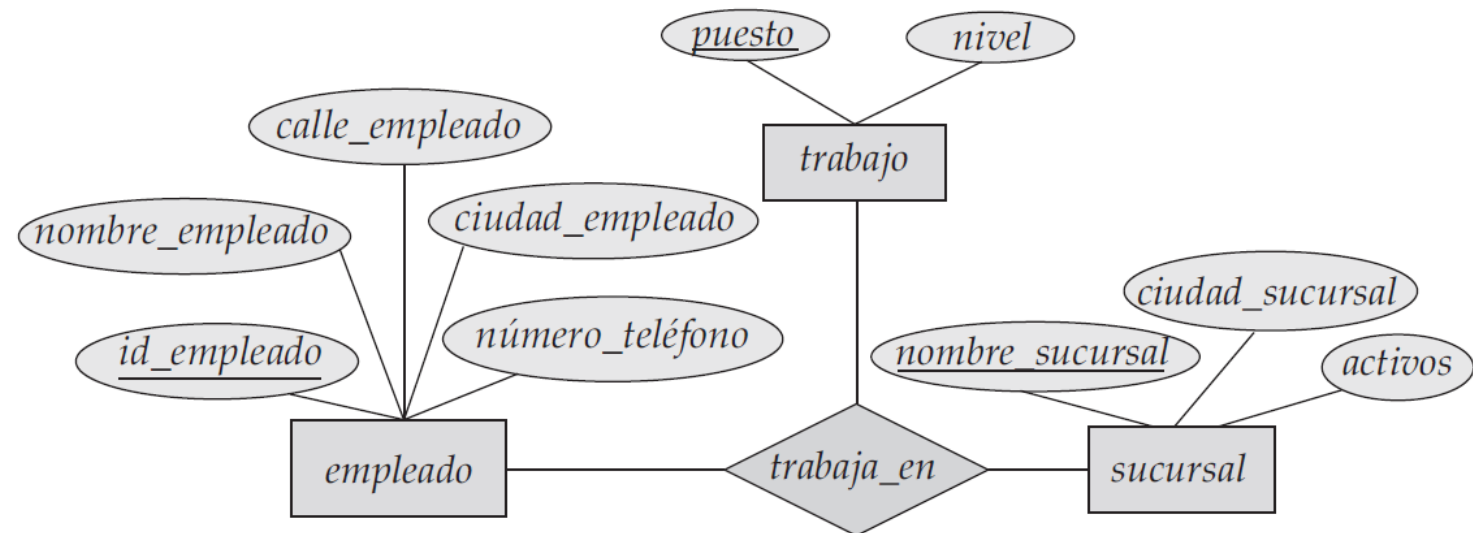
Por ejemplo, considérense los conjuntos de entidades *empleado*, *sucursal* y *trabajo*.

La entidad Trabajo por ejemplo director, cajero, interventor, etc.

La relación *trabaja_en* entre *Empleado*, *Sucursal* y *Trabajo* es un ejemplo de relación ternaria. Una relación ternaria entre *Santos*, *Scotiabank Centro* y *director*, indica que Santos trabaja de director de la sucursal de Scotiabank Centro. Santos también podría actuar como interventor de la sucursal de Scotiabank Ánimas.

Podría haber otra relación entre Gómez, *Scotiabank Urban Center*, *cajero*, que indicaría que Gómez trabaja de cajero en la sucursal de *Scotiabank Urban Center*.

Trabaja_en se convierte en entidad débil.
Trabaja_en(id_empleado,nombre sucursal,puesto)



Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.

c) Eliminar las relaciones recursivas

Eliminar las relaciones recursivas, cuando una entidad se relaciona consigo misma (por ejemplo, una Entidad *empleado* desempeña un rol de supervisor en uno de los lados de la relación y un rol de supervisado en el otro lado).

Sustituyéndola por una nueva entidad (débil).

La cardinalidad de estas relaciones dependerá de su significado.

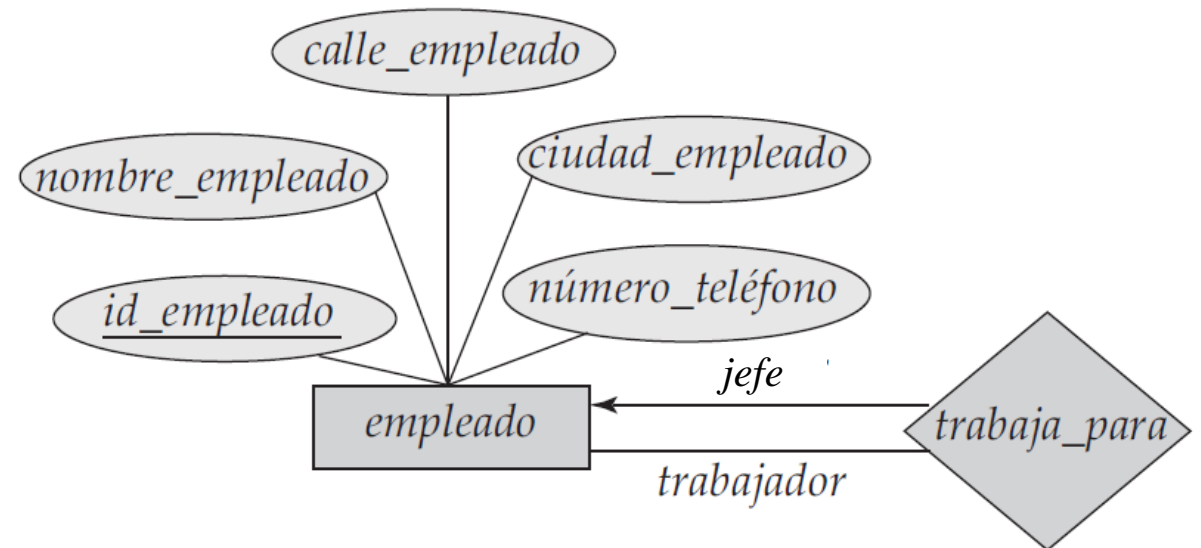
Modelo relacional

Eliminar las relaciones recursivas

En el ejemplo, cada empleado trabaja para un único jefe y cada jefe supervisa a uno o más empleados.

empleado (*id_empleado*, *nombre_empleado*...)

trabaja_para (*id_empleado trabajador*, *id_empleado jefe*)



Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - d) Eliminar las relaciones con atributos

Eliminar las relaciones con atributos, sustituyéndola por una nueva entidad (débil).

La cardinalidad de estas relaciones dependerá del tipo de la relación original y de su significado.

Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - d) Eliminar las relaciones con atributos

Eliminar las relaciones con atributos, sustituyendo cada una de ellas por una nueva entidad (débil) y las relaciones binarias correspondientes de esta nueva entidad con las entidades originales.

La cardinalidad de estas relaciones dependerá del tipo de la relación original y de su significado.

Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - e) Eliminar los atributos multivaluados

Eliminar los atributos multivaluados, sustituyendo cada uno de ellos por una nueva entidad (débil) y una relación binaria de uno a muchos con la entidad original.

Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - f) Revisar las relaciones de uno a uno

Revisar las relaciones de uno a uno, ya que es posible que se hayan identificado dos entidades que representen el mismo objeto (sinónimos). Si así fuera, ambas entidades deben integrarse en una sola.

Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - g) Eliminar las relaciones redundantes.

Eliminar las relaciones redundantes. Una relación es redundante cuando se puede obtener la misma información que ella aporta mediante otras relaciones.

El hecho de que haya dos caminos diferentes entre dos entidades no implica que uno de los caminos corresponda a una relación redundante, eso dependerá del significado de cada relación.

Modelo relacional

2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

- Se obtiene un conjunto de relaciones (tablas) para cada una de las entidades y relaciones entre entidades

Entidades fuertes. Crear una relación para cada entidad fuerte que incluya todos sus atributos simples. De los atributos compuestos incluir sólo sus componentes.

Cada uno de los identificadores de la entidad será una **clave** candidata. Elegir la **clave** primaria; el resto serán **claves** alternativas.

Modelo Relacional y su álgebra asociado.

Claves

Una característica importante de las relaciones en el modelo relacional es:

- ❖ Dos tuplas no pueden ser iguales en todos sus atributos. La manera de lograr que esta regla se cumpla es a través de una **clave primaria**.

Modelo Relacional y su álgebra asociado.

Claves

Una **superclave** es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades.

Por ejemplo, el atributo id-cliente del conjunto de entidades cliente es suficiente para distinguir una entidad cliente de las otras.

Modelo Relacional y su álgebra asociado.

Claves

Análogamente, la combinación de nombre-cliente e id-cliente es una superclave del conjunto de entidades cliente.

El atributo nombre-cliente de cliente no es una superclave por si sola, porque varias personas podrían tener el mismo nombre.

Una superclave puede contener atributos innecesarios.

Modelo Relacional y su álgebra asociado.

Claves

Cada tupla en una relación puede distinguirse de las demás por un número determinado de atributos. Estos atributos se denominan **claves candidatas**.

Una **clave primaria** es una clave candidata que ha sido seleccionada para la identificación de las tuplas de una relación.

Esquema_cuenta = (número_cuenta, nombre_sucursal, saldo)

Modelo relacional

2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

Para escoger la clave primaria entre las claves candidatas:

- Escoger la clave candidata que tenga menos atributos.
- Escoger la clave candidata cuyos valores no tengan probabilidad de cambiar en el futuro.
- Escoger la clave candidata cuyos valores no tengan probabilidad de perder la unicidad en el futuro.
- Escoger la clave candidata con el mínimo número de caracteres (si es de tipo texto).
- Escoger la clave candidata más fácil de utilizar desde el punto de vista de los usuarios.

Modelo relacional

2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

Entidades débiles. Crear una relación para cada entidad débil incluyendo todos sus atributos simples.

- I. Incluir solo los componentes de los atributos compuestos.
- II. Añadir una clave ajena a la entidad de la que depende.
Incluyendo la clave primaria de la relación que representa a la entidad padre en la nueva relación creada para la entidad débil.
- III. Determinar la clave primaria de la nueva relación.

Modelo relacional

2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

Relaciones binarias de uno a muchos.

- I. Se incluyen los atributos de la clave primaria de la entidad padre en la relación (tabla) que representa a la entidad hijo, para actuar como una clave ajena.
- II. La clave primaria de la entidad con cardinalidad 1 pasa a la tabla de la entidad cuya cardinalidad es N

Propagación de clave

Modelo relacional

2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

Relaciones binarias de uno a uno.

- I. Para cada relación binaria se incluyen los atributos de la clave primaria de la entidad padre en la relación (tabla) que representa a la entidad hijo, para actuar como una clave ajena.
- II. La entidad hijo es la que participa de forma total (obligatoria) en la relación
- III. La entidad padre es la que participa de forma parcial (opcional).

Modelo relacional

2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

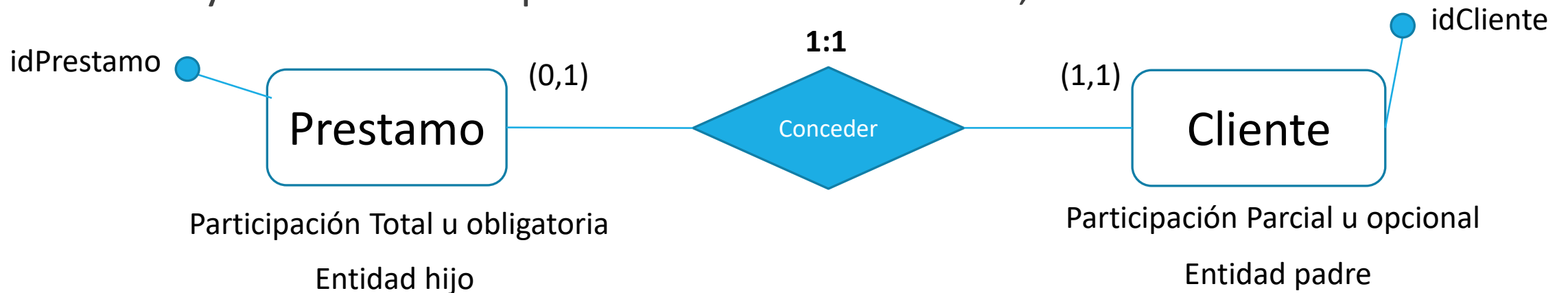
Relaciones binarias de uno a uno.

- IV. Si las dos entidades participan de forma total o parcial en la relación, la elección de padre e hijo es arbitraria.
- V. Si ambas entidades participan de forma total en la relación, se tiene la opción de integrar las dos entidades en una sola relación (tabla). Esto se suele hacer si una de las entidades no participa en ninguna otra relación.

Modelo relacional

Relaciones binarias de uno a uno.

Ejemplo: En una institución bancaria, un Préstamo le es concedido a un Cliente y a un Cliente le puede ser concedido o no, un Préstamo.



Es posible que sólo algunas de las entidades *cliente* estén relacionadas con el conjunto de entidades *préstamo* mediante la relación *Conceder*, y la participación de *cliente* en la relación *Conceder* es, por tanto, parcial. **Prestamo(idPrestamo,...,idCliente)**

Modelo relacional

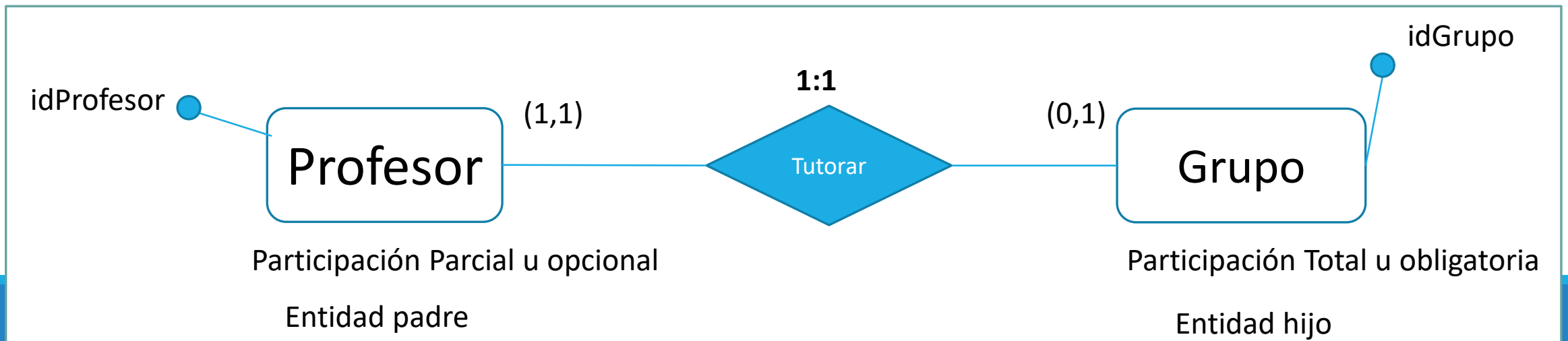
Relaciones binarias de uno a uno.

Ejemplo:

Un profesor puede ser tutor de un sólo grupo y del mismo modo, cada grupo sólo puede tener un tutor.

Es posible que sólo algunas de las entidades *Profesor* estén relacionadas con el conjunto de entidades *Grupo* mediante la relación *Tutorar*, y la participación de *Profesor* en la relación *Tutorar* es, por tanto, parcial. Por otro lado, todos los grupos deben tener un profesor.

Grupo(idGrupo,....,idProfesor)



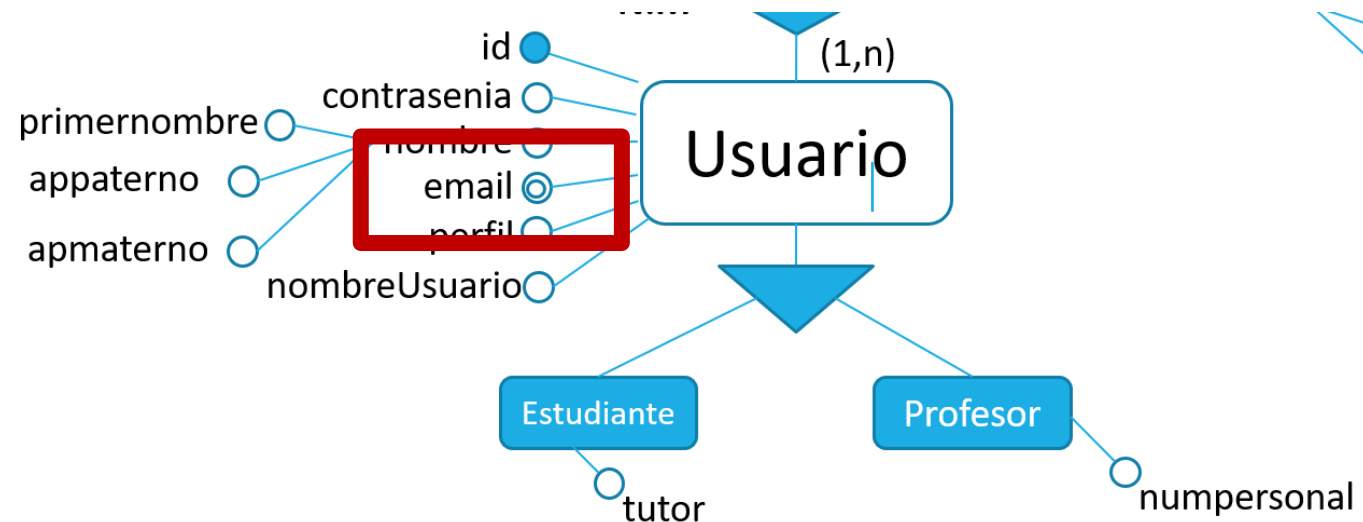
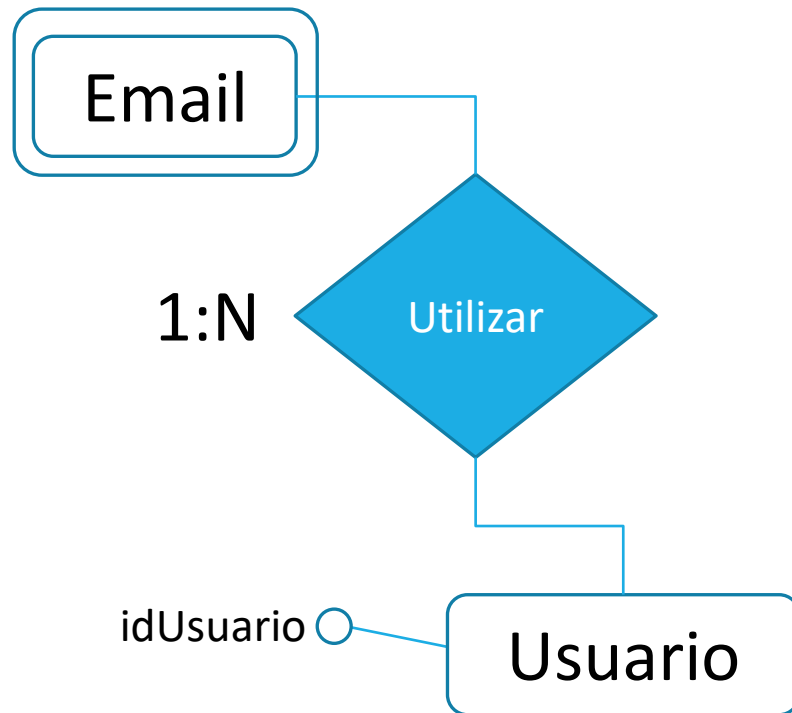
Modelo relacional

1. Convertir el esquema conceptual en un esquema lógico. Detalle.
 - e) Eliminar los atributos multivaluados

Eliminar los atributos multivaluados, sustituyendo cada uno de ellos por una nueva entidad (débil) y una relación binaria de uno a muchos con la entidad original.

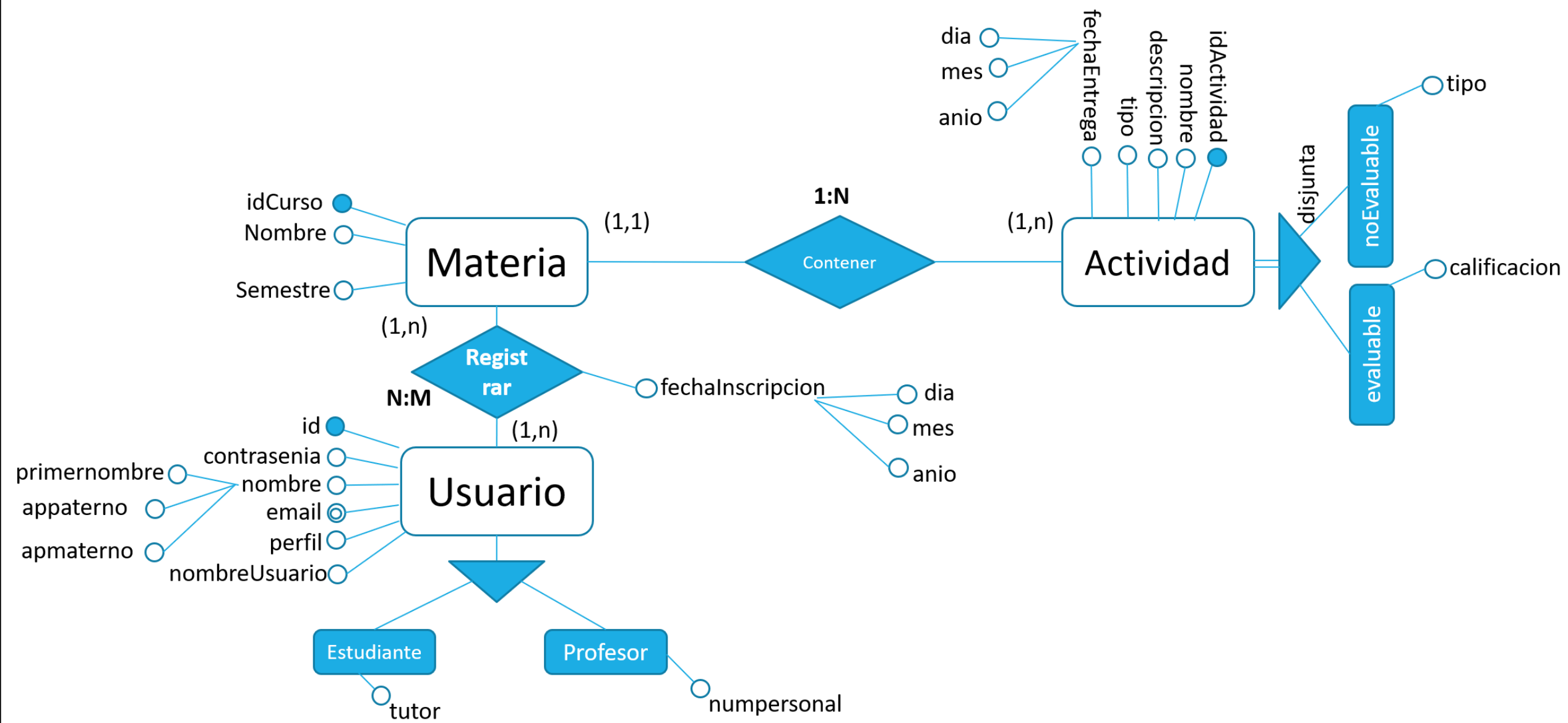
Modelo relacional

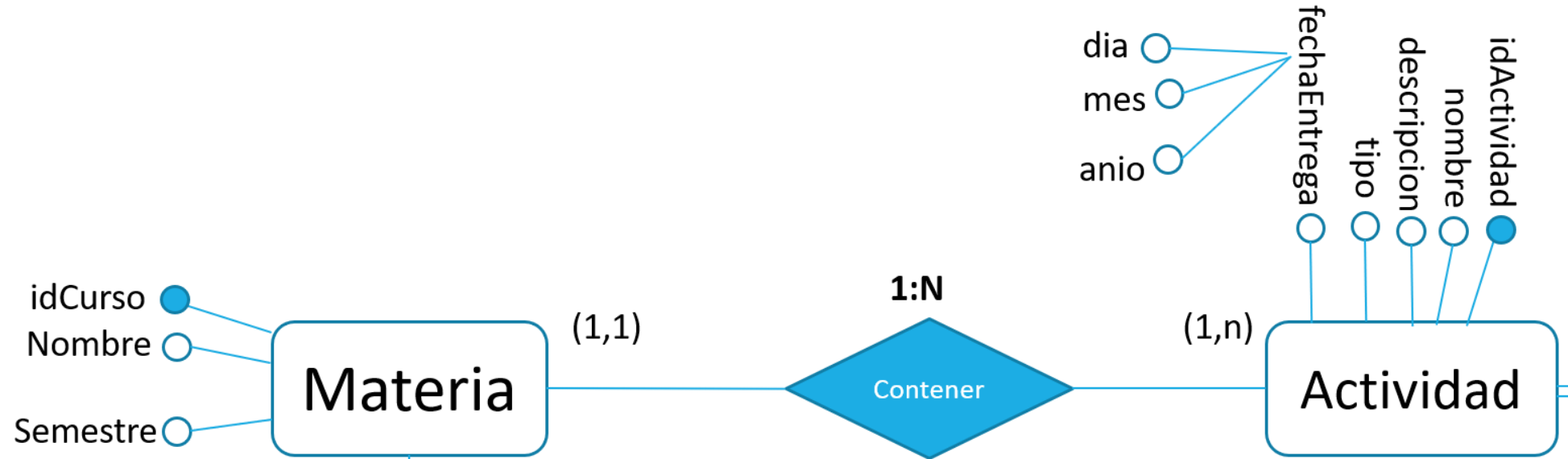
Atributos Multivaluados.



Email (idUsuario,email)

Ejemplo

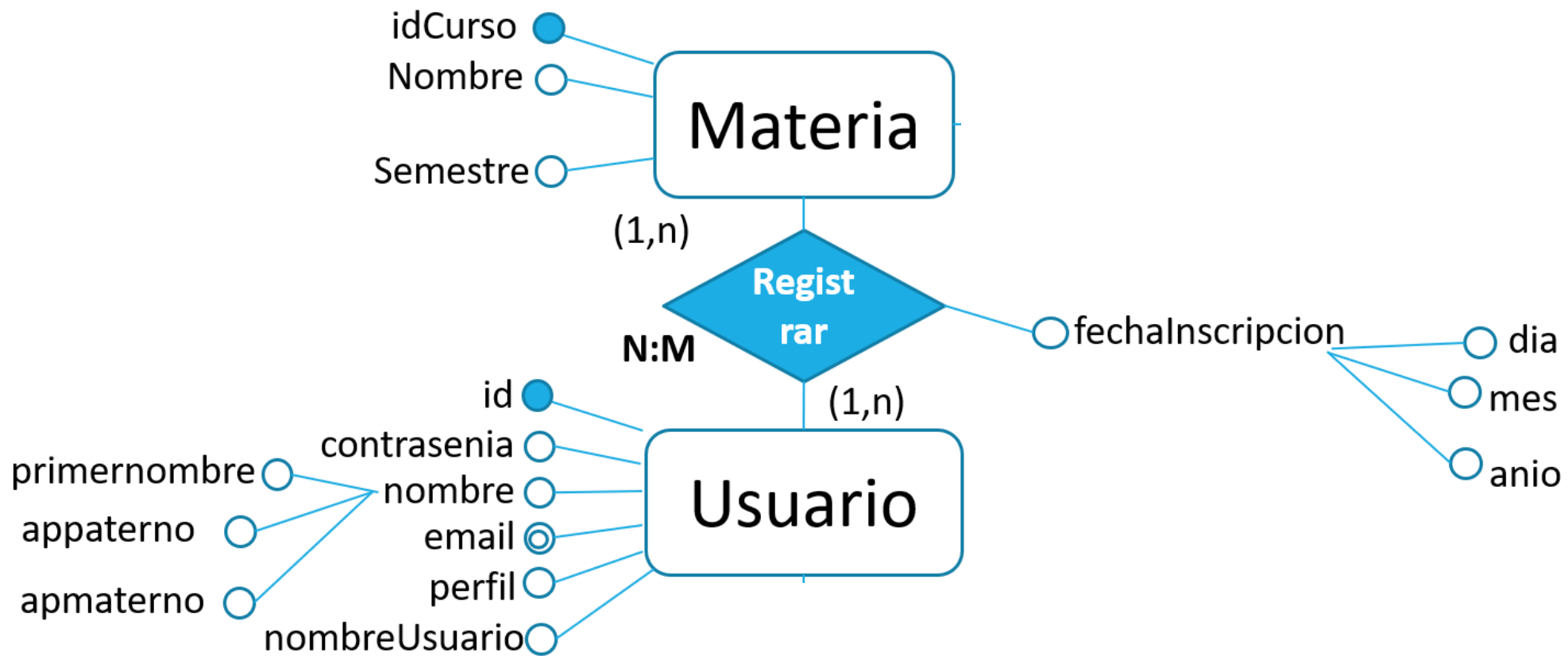




Relaciones resultantes:

Materia (idCurso, Nombre, Semestre)

Actividad (idActividad, nombre, descripcion, tipo, fecha_entrega, idCurso)



Relaciones resultantes:

Usuario (idUsuario, primernombre, appaterno, amaterno, email, usuario, contraseña, perfil)

Registrar (idUsuario, idCurso, día, mes, anio)

Modelo Lógico

Modelo Relacional

Esquemas de las relaciones del Modelo Relacional

Materia (idCurso, Nombre, Semestre)

Actividad (idActividad, nombre, descripcion, tipo, fecha_entrega, *idCurso*)

Registrar (idUsuario, idCurso, fecha_inscripcion)

Usuario (idUsuario, primernombre, appaterno, amaterno, email, usuario, contrasenia, perfil)

Modelo relacional

Jerarquías de generalización. Hay tres opciones distintas para representar las jerarquías. La elección de la más adecuada se hará en función de su tipo (total/parcial, exclusiva/superpuesta).

1. Crear una relación por cada entidad. Las relaciones de las entidades hijo heredan como clave primaria la de la entidad padre.
2. Crear una relación por cada entidad hijo, heredando los atributos de la entidad padre. Esta opción sólo sirve para jerarquías totales y exclusivas.
3. Integrar todas las entidades en una relación, incluyendo en ella los atributos de la entidad padre, los atributos de todos los hijos y un atributo discriminativo para indicar el caso al cual pertenece la entidad en consideración.

Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

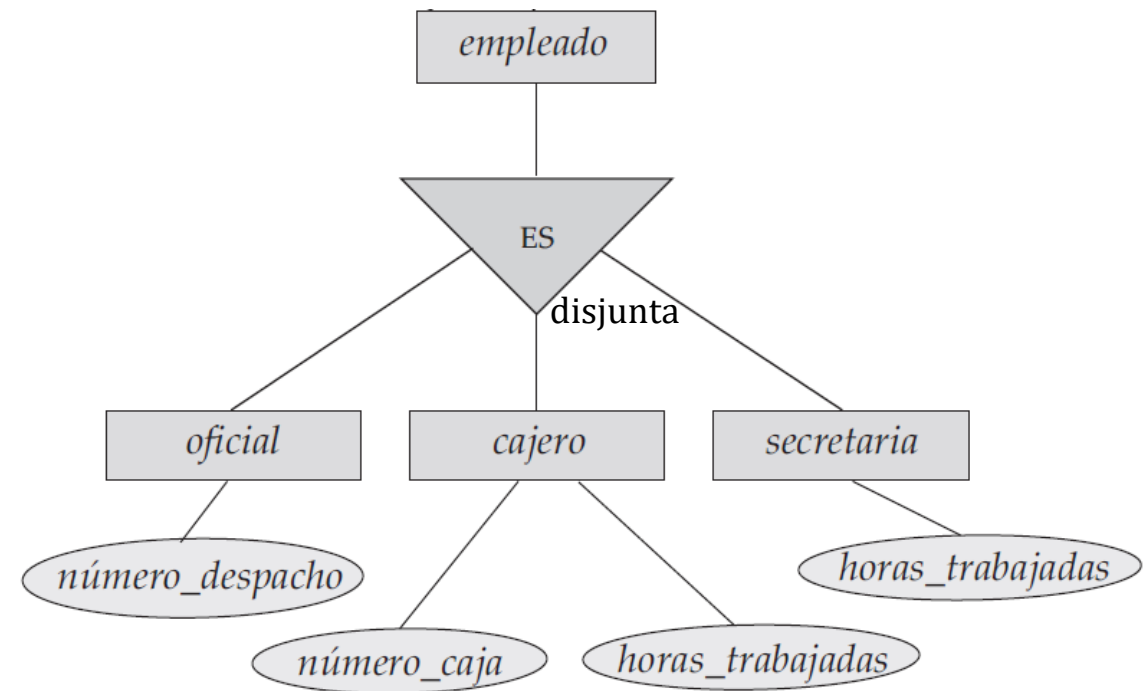
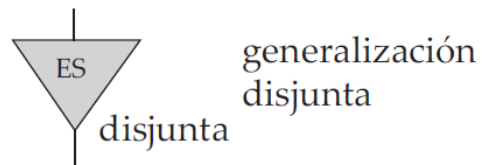
- **Disjuntos o Exclusiva.** La condición de disjunción exige que cada entidad no pertenezca a más de un conjunto de entidades de nivel inferior.

Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

○ **Disjuntos.** Un *empleado* solo puede ser oficial o cajero o secretaria.



Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

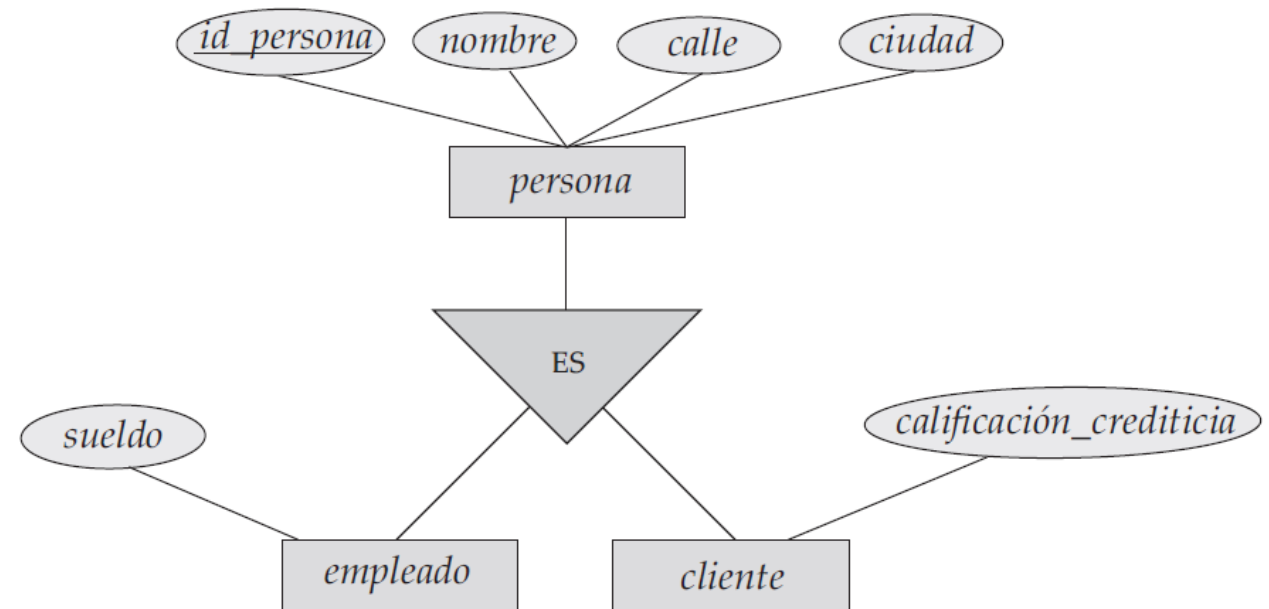
- **Solapados o Superpuesta.** la misma entidad puede pertenecer a más de un conjunto de entidades de nivel inferior de la generalización.

Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

○ **Solapados.** Los empleados también pueden ser clientes.



Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

○ **Restricción de completitud.**

Especifica si una entidad de nivel superior debe pertenecer, al menos, a uno de los conjuntos de entidades de nivel inferior de la generalización o especialización. Esta restricción puede ser de uno de los tipos siguientes:

Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

○ **Restricción de completitud.**

- Generalización o especialización total.
- Generalización o especialización parcial.

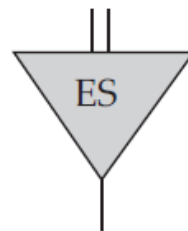
Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

○ Restricción de completitud.

Generalización o especialización total. Cada entidad de nivel superior debe pertenecer a un conjunto de entidades de nivel inferior.



generalización
total

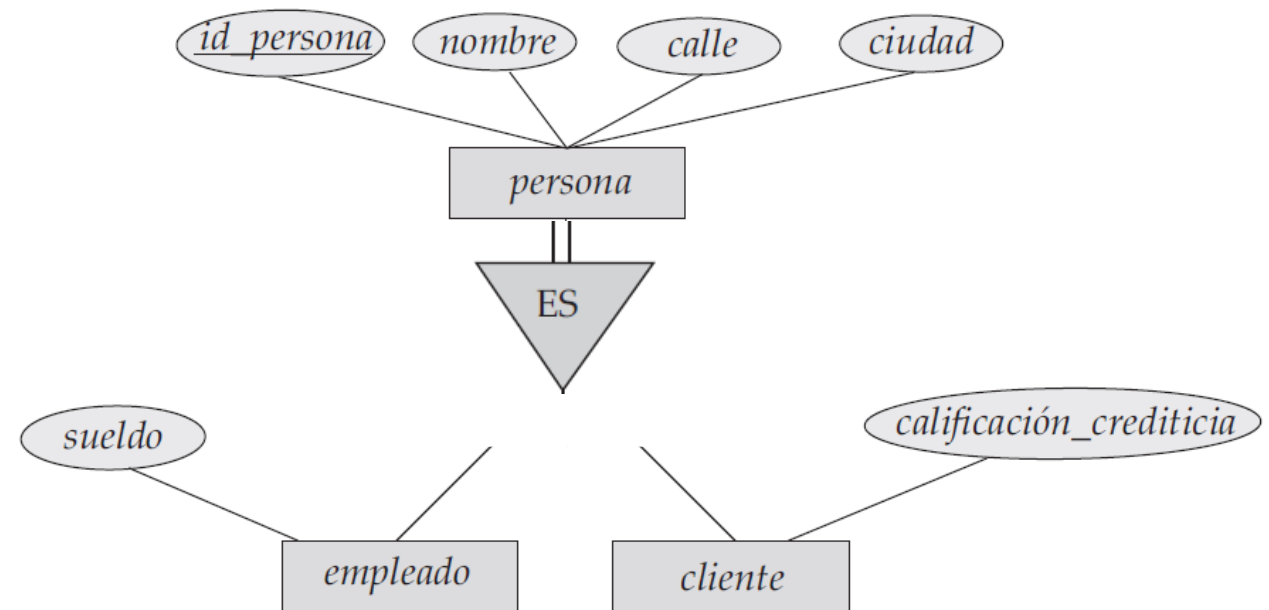
Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

○ Restricción de completitud.

Generalización o especialización total.



Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

○ **Restricción de completitud.**

Generalización o especialización parcial. Puede que alguna entidad de nivel superior no pertenezca a ningún conjunto de entidades de nivel inferior.

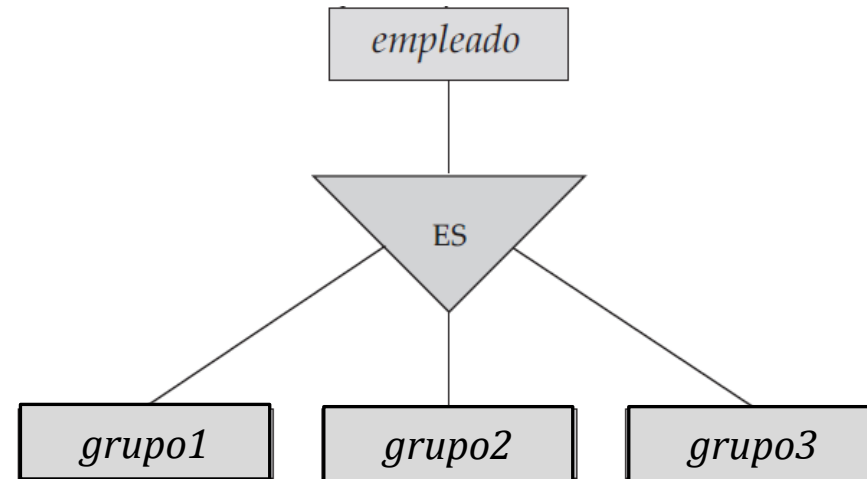
Características del modelo E-R extendido

❖ Especialización y Generalización

✓ Restricciones

○ Restricción de completitud.

Generalización o especialización parcial.



Modelo relacional

Jerarquías de generalización. Hay tres opciones distintas para representar las jerarquías. La elección de la más adecuada se hará en función de su tipo (total/parcial, exclusiva/superpuesta).

1. Crear una relación por cada entidad. Las relaciones de las entidades hijo heredan como clave primaria la de la entidad padre.
2. Crear una relación por cada entidad hijo, heredando los atributos de la entidad padre. Esta opción sólo sirve para jerarquías totales y exclusivas.
3. Integrar todas las entidades en una relación, incluyendo en ella los atributos de la entidad padre, los atributos de todos los hijos y un atributo discriminativo para indicar el caso al cual pertenece la entidad en consideración.

Modelo relacional

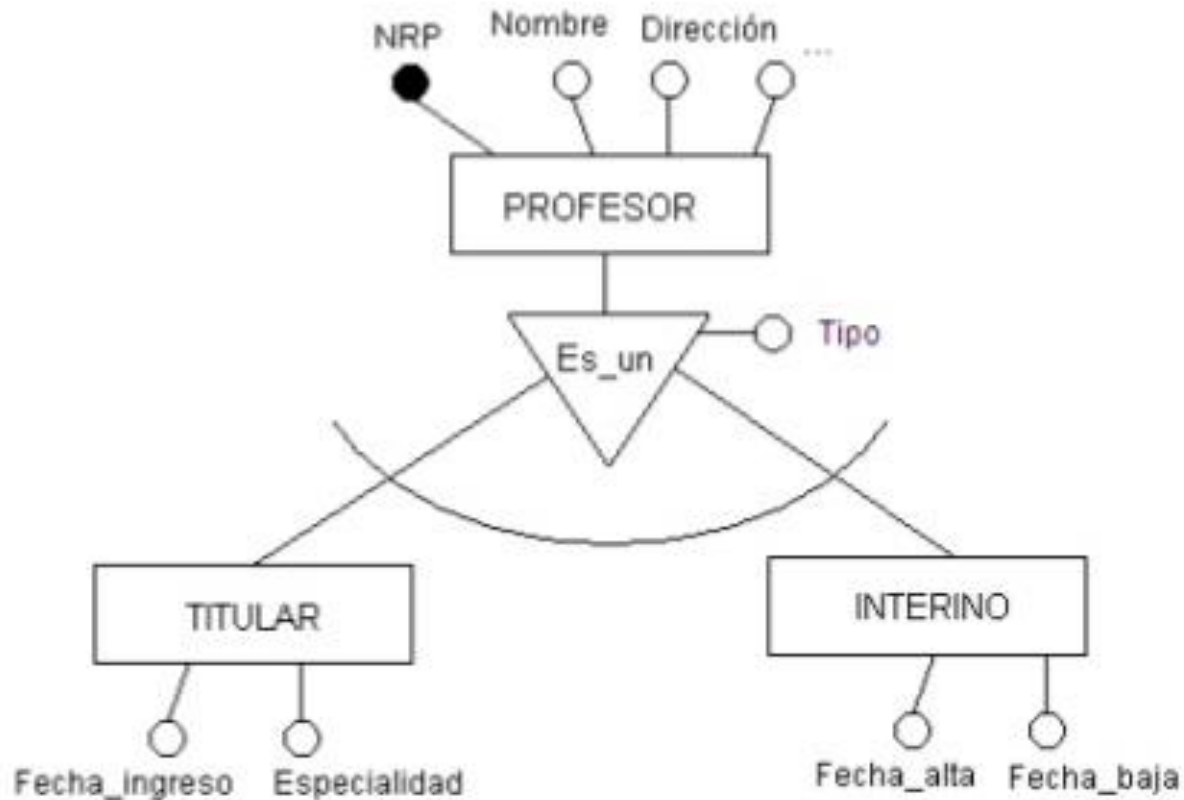
2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

Jerarquías de generalización. Entidad padre = entidad genérica, superclase. Entidades hijo = subentidades, subclases.

A) Crear una relación para la superclase y tantas relaciones como subclases haya.

Contendrán sus atributos propios y los atributos de la clave primaria de la superclase.

Modelo relacional



PROFESOR (NRP, Nombre, Dirección..., Tipo)

TITULAR (NRP, Fecha_ingreso, Especialidad)

INTERINO (NRP, Fecha_alta, Fecha_baja)

Empleada cuando existen muchos atributos distintos entre las subclases y los atributos en común desean mantenerse.

No genera valores nulos: Los atributos exclusivos pertenecen a su correspondiente relación.

Es la menos eficiente cuando se recurre a la combinación de tablas para obtener información.

Modelo relacional

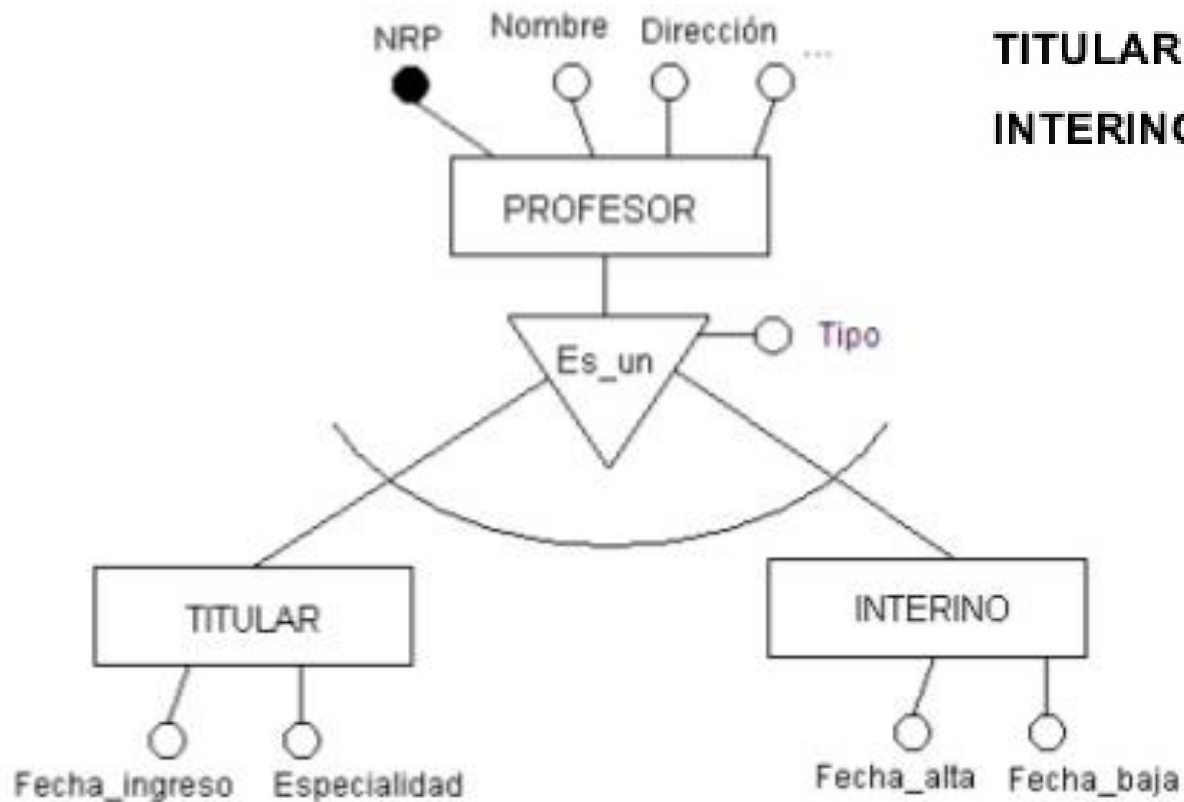
2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

Jerarquías de generalización. Entidad padre = entidad genérica, superclase. Entidades hijo = subentidades, subclases.

B) No crear una relación para la superclase y crear tantas relaciones como subclases haya.

Contendrán además de sus propios atributos los atributos de la superclase (o atributos comunes)

Modelo relacional



TITULAR (NRP, Nombre, Dirección..., Fecha_ingreso, Especialidad)

INTERINO (NRP, Nombre, Dirección..., Fecha_alta, Fecha_baja)

Empleada cuando las subclases son *disjuntas* (Cuando una ocurrencia de la superclase solo aparece en una única subclase) y la especialización es total (Todo profesor debe ser titular o interino).

No genera valores nulos.

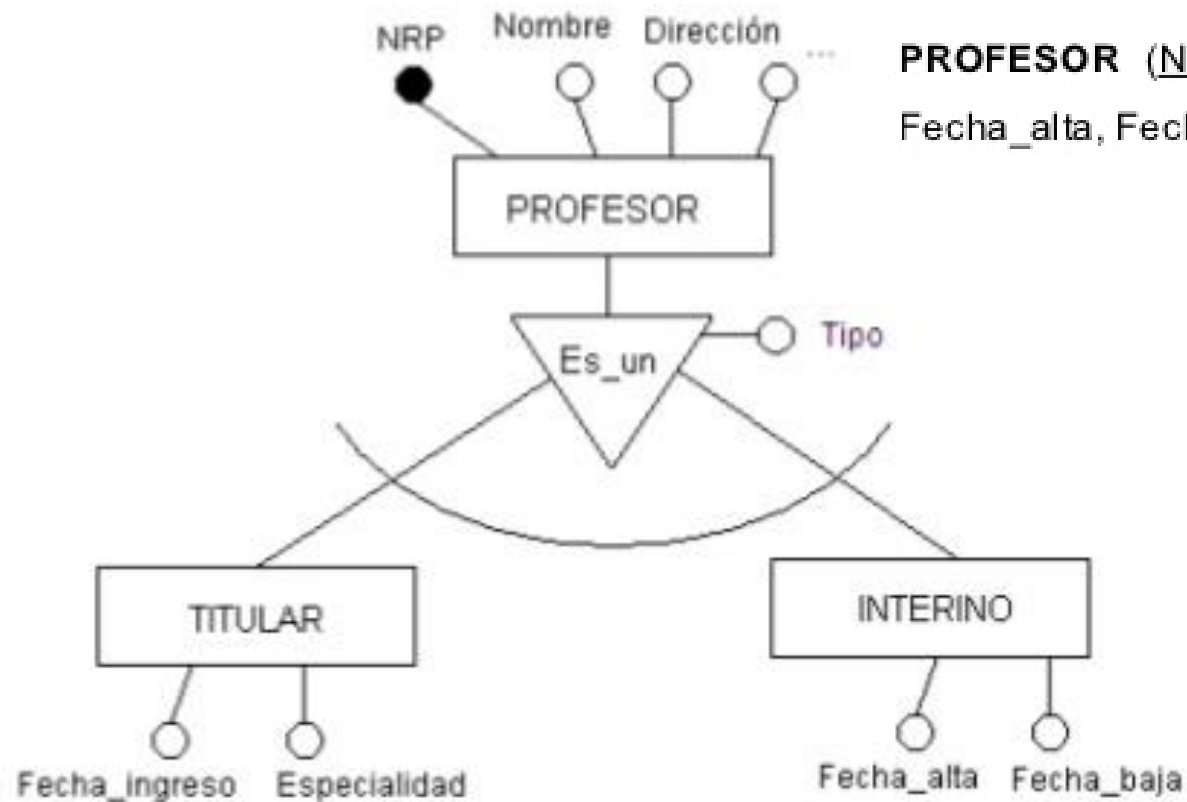
Modelo relacional

2. Derivar un conjunto de relaciones (tablas) para el esquema lógico.

Jerarquías de generalización. Entidad padre = entidad genérica, superclase. Entidades hijo = subentidades, subclases.

C) Englobar todos los atributos de la entidad superclase y subclases en una sola relación, añadiendo el atributo que indica el tipo de subclase.

Modelo relacional

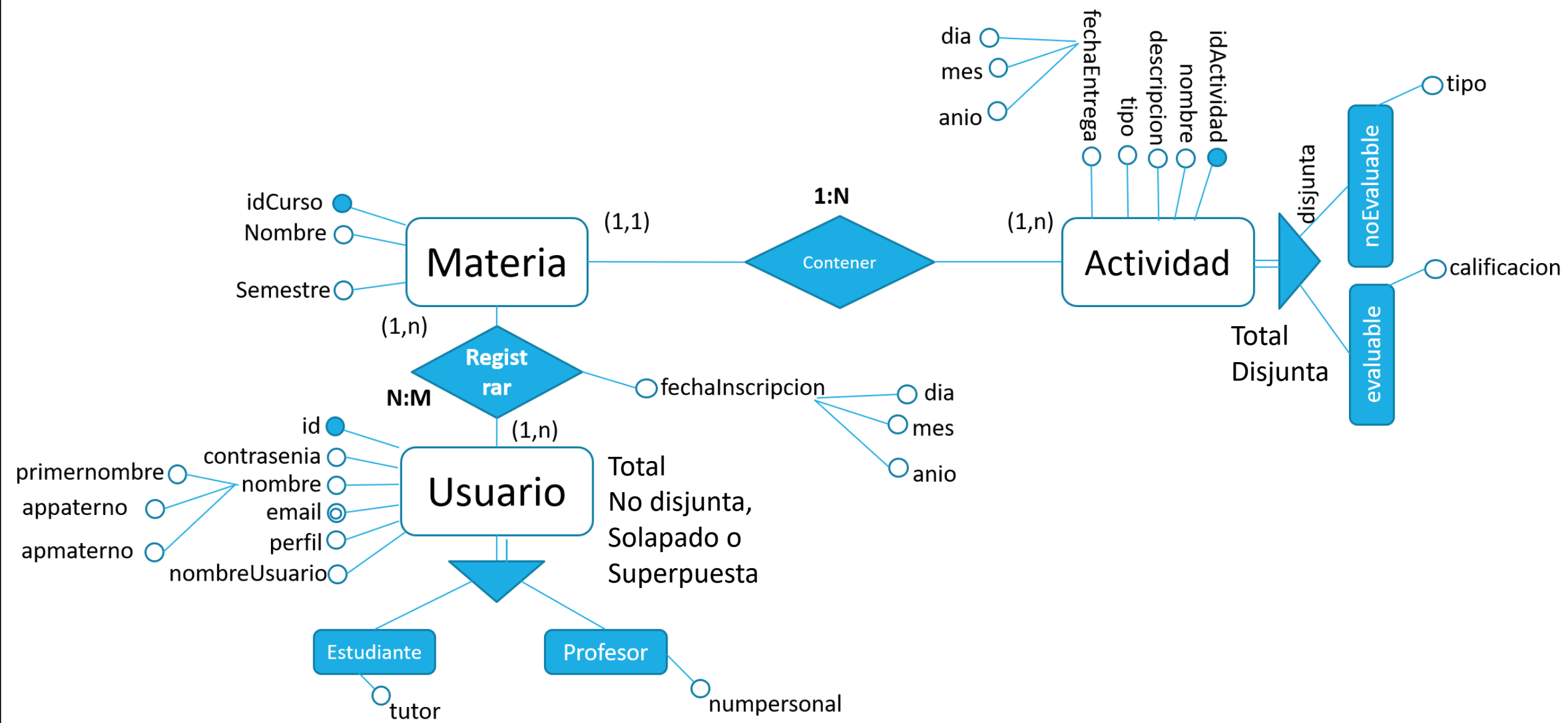


PROFESOR (NRP, Nombre, Dirección... Tipo, Fecha_ingreso, Especialidad, Fecha_alta, Fecha_baja)

Empleada cuando las subclases se diferencian en muy pocos atributos y las interrelaciones que los asocian con el resto de las entidades del esquema son las mismas para todos.

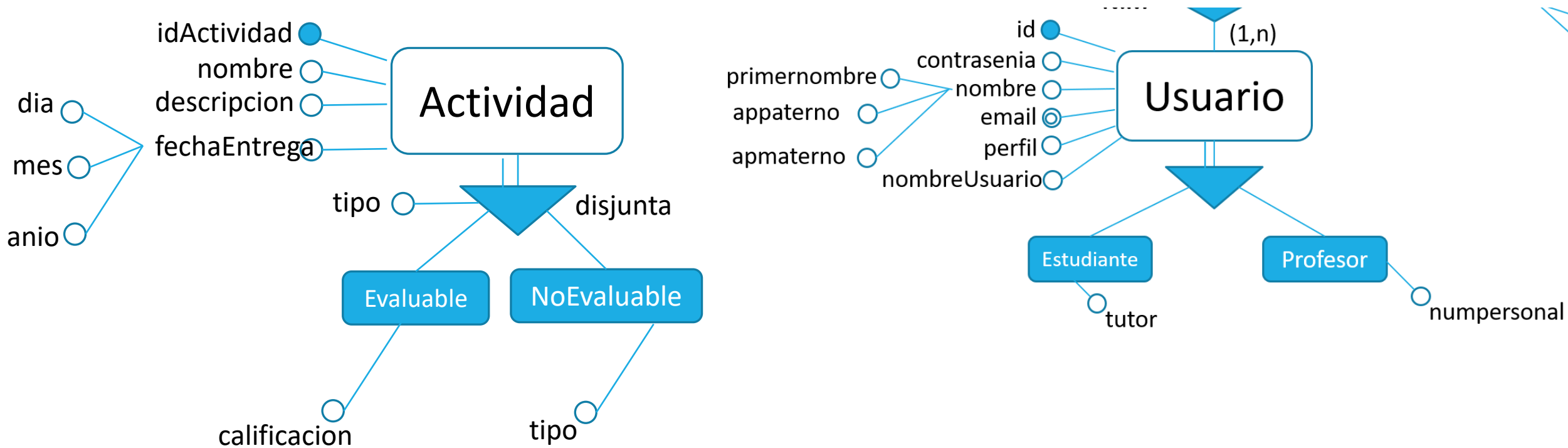
El acceso a los datos de una ocurrencia, es mucho más rápido; pues no se combinan relaciones.

Existen valores nulos en los atributos específicos de cada subclase.



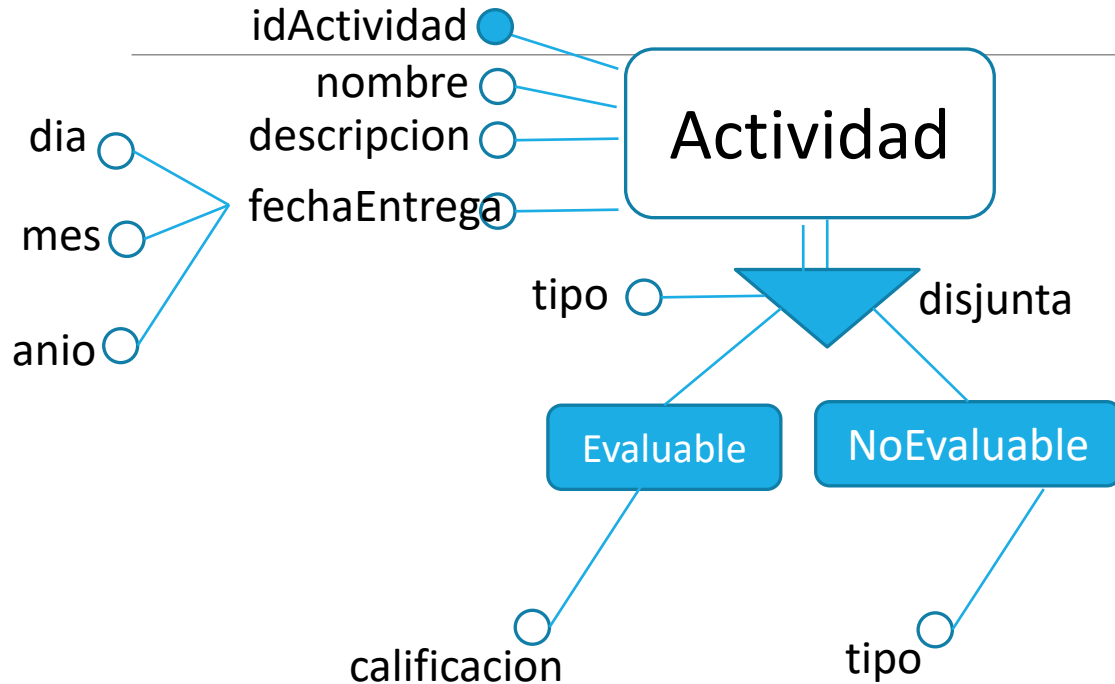
Modelo relacional

Jerarquías de generalización.



Modelo relacional

Jerarquías de generalización.



No crear una relación para la superclase y crear tantas relaciones como subclases haya.

Empleada cuando las subclases son *disjuntas* (Cuando una ocurrencia de la superclase solo aparece en una única subclase) y la especialización es total (Todo profesor debe ser titular o interino).

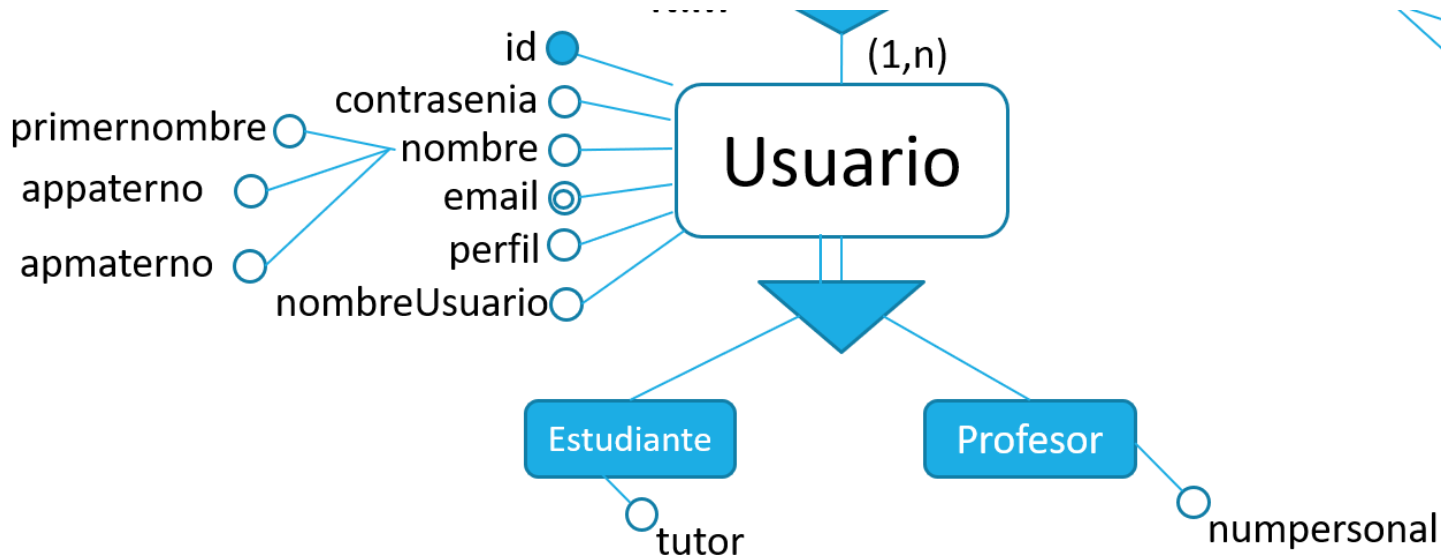
No genera valores nulos.

ActividadEvaluable(idActividad,nombre,descripcion, diaEntrega,mesEntrega,anioEntrega,calificación,tipo)

ActividadNoEvaluable(idActividad,nombre,descripcion, diaEntrega,mesEntrega,anioEntrega,tipoNoEval,tipo)

Modelo relacional

Jerarquías de generalización.



Usuario(idUsuario, primerNombre, apPaterno, apMaterno, email, nombreUsuario, contraseña, tutor, numPersonal, perfil)

Englobar todos los atributos de la entidad superclase y subclases en una sola relación, añadiendo el atributo que indica el tipo de subclase.

Empleada cuando las subclases se diferencian en muy pocos atributos y **las interrelaciones que los asocian con el resto de las entidades del esquema son las mismas para todos.**

El acceso a los datos de una ocurrencia, es mucho más rápido; pues no se combinan relaciones.

Existen valores nulos en los atributos específicos de cada subclase.

Modelo Lógico

Modelo Relacional

Materia (idCurso, Nombre, Semestre)

Actividad (idActividad, nombre, descripcion, tipo, fecha_entrega, *idCurso*)

Registrar (idUsuario, idCurso, fecha_inscripcion)

Usuario (idUsuario, primernombre, apaterno, amaterno, email, usuario, contrasenia, perfil)

Modelo Lógico

Modelo Relacional

Materia (idMateria, nombre, semestre)

Registrar (idUsuario, idCurso, fecha_inscripcion)

Usuario(idUsuario, primerNombre, apPaterno, apMaterno, email, nombreUsuario, contraseña, tutor, numPersonal, perfil)

ActividadEvaluable

(idActividad, nombre, descripcion, diaEntrega, mesEntrega, anioEntrega, calificación, tipo, *idMateria*)

ActividadNoEvaluable(idActividad, nombre, descripcion, diaEntrega, mesEntrega, anioEntrega, tipoNoEval, tipo, *idMateria*)

Esquema de las relaciones:

Materia (idMateria, nombre, semestre)

Registrar (idUsuario, idCurso, fecha_inscripcion)

Usuario(idUsuario, primerNombre, apPaterno, apMaterno, nombreUsuario, contraseña, tutor, numPersonal, perfil)

Email (idUsuario, email)

ActividadEvaluable

(idActividad, nombre, descripcion, diaEntrega, mesEntrega, anioEntrega, calificación, tipo, *idMateria*)

ActividadNoEvaluable(idActividad, nombre, descripcion, diaEntrega, mesEntrega, anioEntrega, tipoNoEval, tipo, *idMateria*)

Modelo relacional

Esquema de la Base de Datos

El diseño general de la base de datos se denomina el esquema de la base de datos.

Cuando se habla de bases de datos se debe diferenciar entre el esquema de la base de datos, o diseño lógico de la misma, y el ejemplar de la base de datos, que es una instantánea de los datos de la misma en un momento dado.

Los **esquemas de las relaciones** incluyen una lista de los atributos y de sus dominios correspondientes. Como una definición de tipos.

Esquema

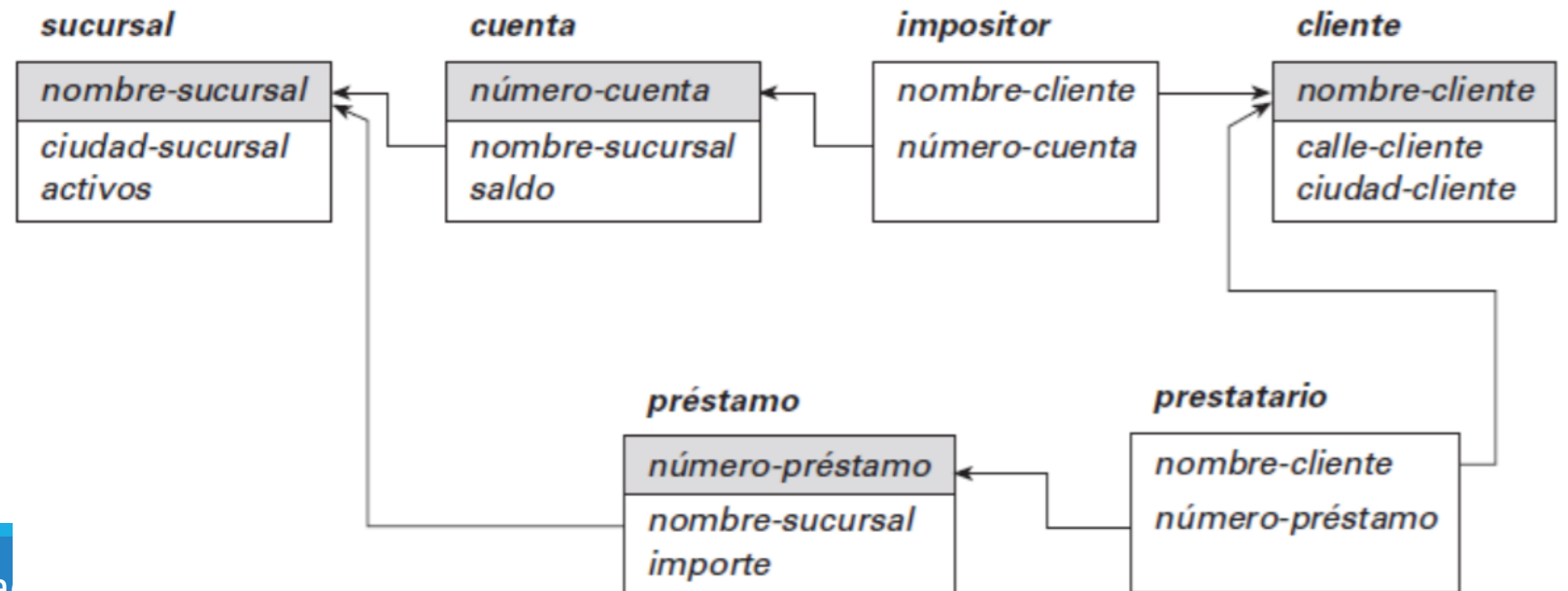
Es el nombre que se le da a una relación y el conjunto de atributos en ella, incluyendo el campo llave.

Materia (idMateria, nombre, semestre)

Registrar (idUsuario, idCurso, fecha_inscripcion)

Diagramas de Esquema

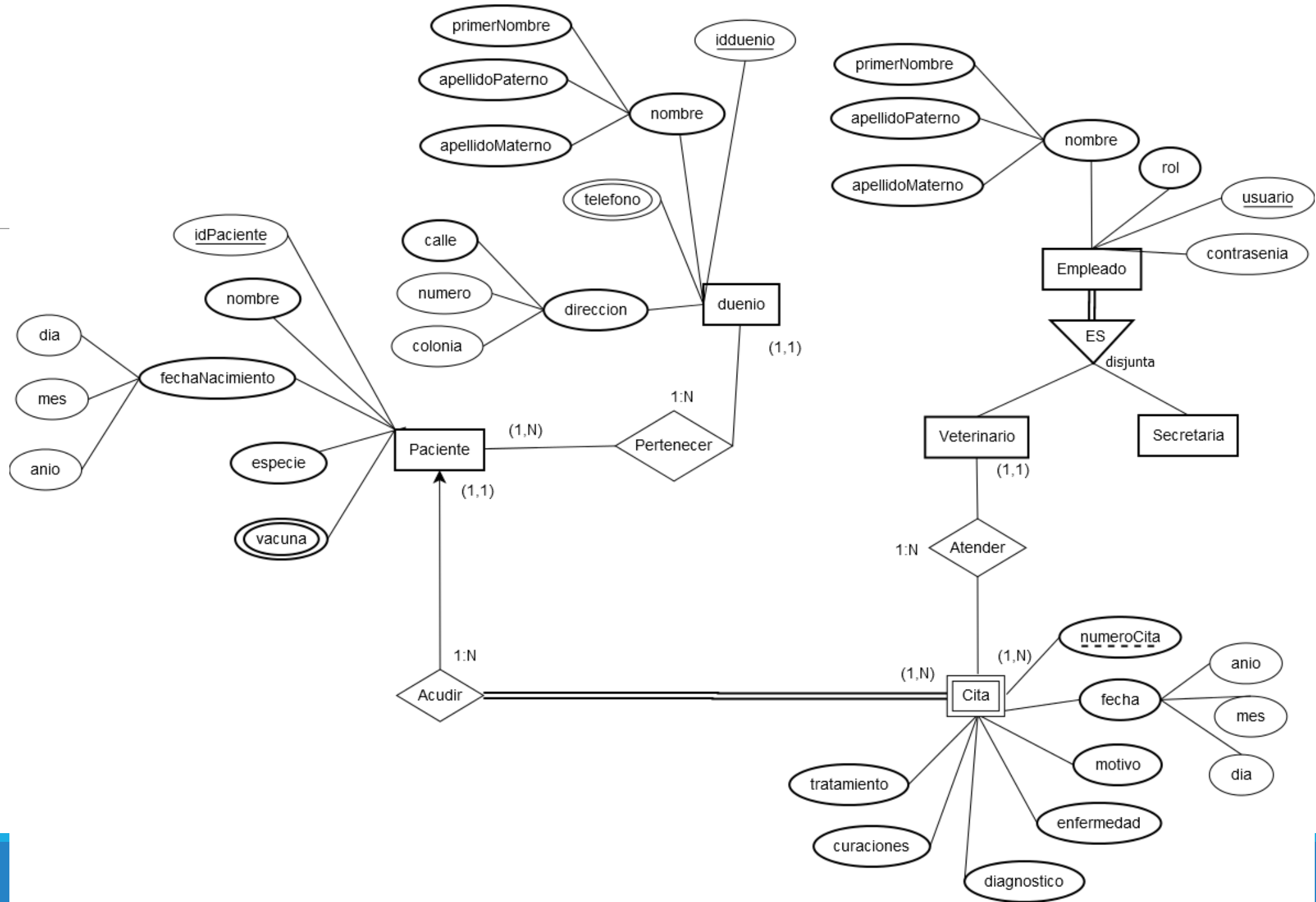
Un esquema de bases de datos, junto con las dependencias de clave primaria y externa, se puede mostrar gráficamente mediante diagramas de esquema.



Actividad

Elabora el Modelo Relacional correspondiente al diagrama Entidad-Relación que se muestra en la siguiente lámina; y que responde a la problemática de la Clínica Veterinaria.

Sigue las reglas estudiadas en la clase.



Solución

Esquema de las relaciones

Base de datos “ClinicaVeterinaria”

Duenio (idDuenio, nombre, apPaterno, apMaterno, calle, numero, colonia)

Paciente (idPaciente, nombre, día_nac, mes_nac, anio_nac, especie, *idDuenio*)

Cita (numeroCita, día_cita, mes_cita, anio_cita, enfermedad, diagnostico, tratamiento, *idPaciente*, *usuario*)

Veterinario (usuario, primerNombre, apPaterno, apMaterno, nombreUsuario, contrasenia)

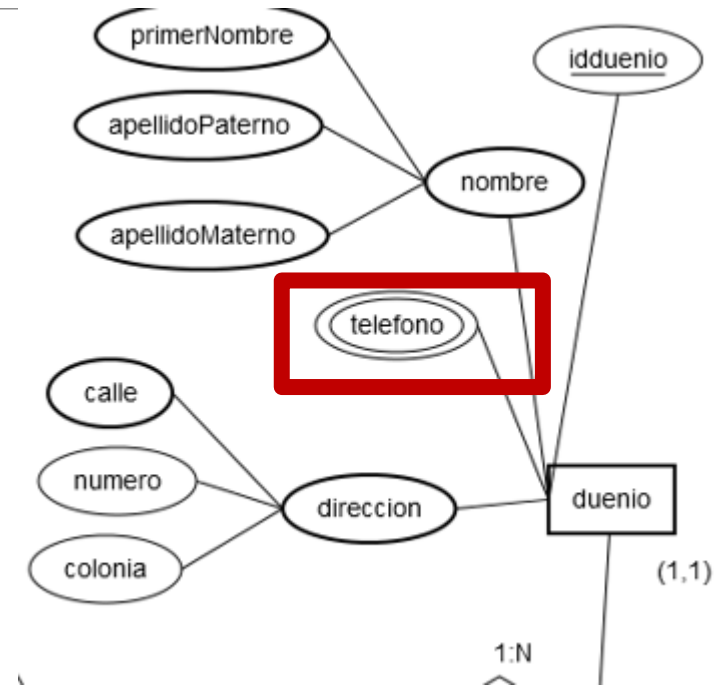
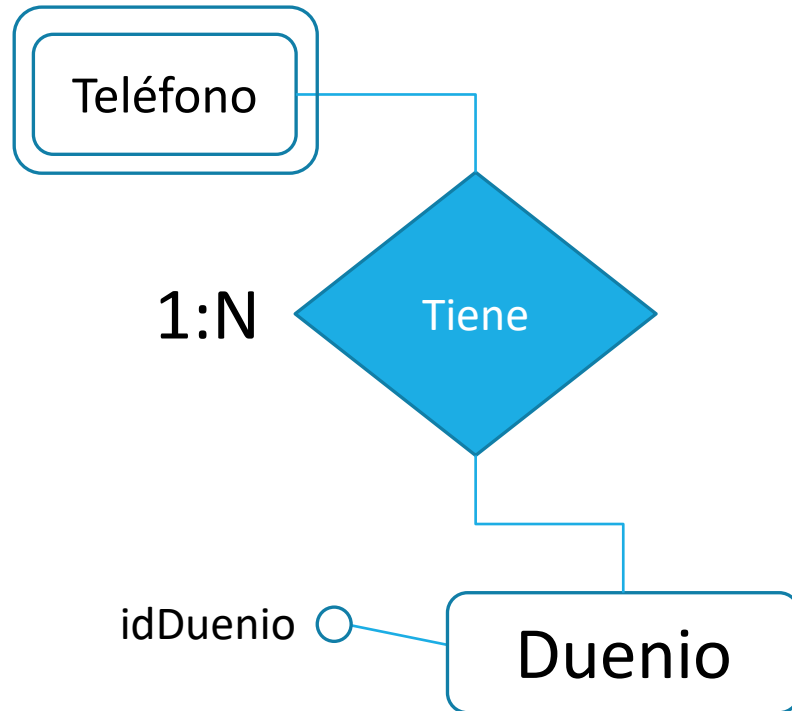
Secretaria (usuario, primerNombre, apPaterno, apMaterno, nombreUsuario, contrasenia)

Vacuna_Paciente (idPaciente, nombreVacuna, diaAplicacion, mesAplicacion, anioAplicacion)

Telefono_duenio (idDuenio, teléfono, tipo)

Modelo relacional

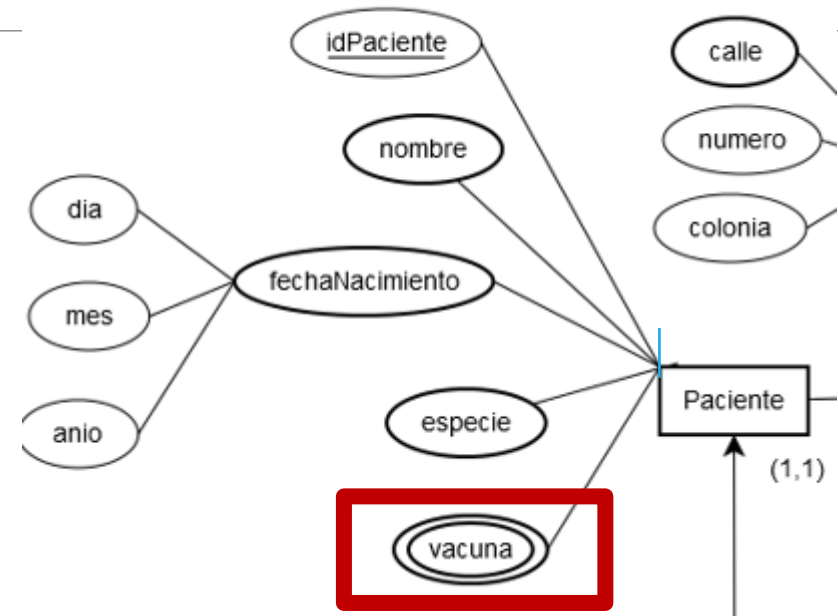
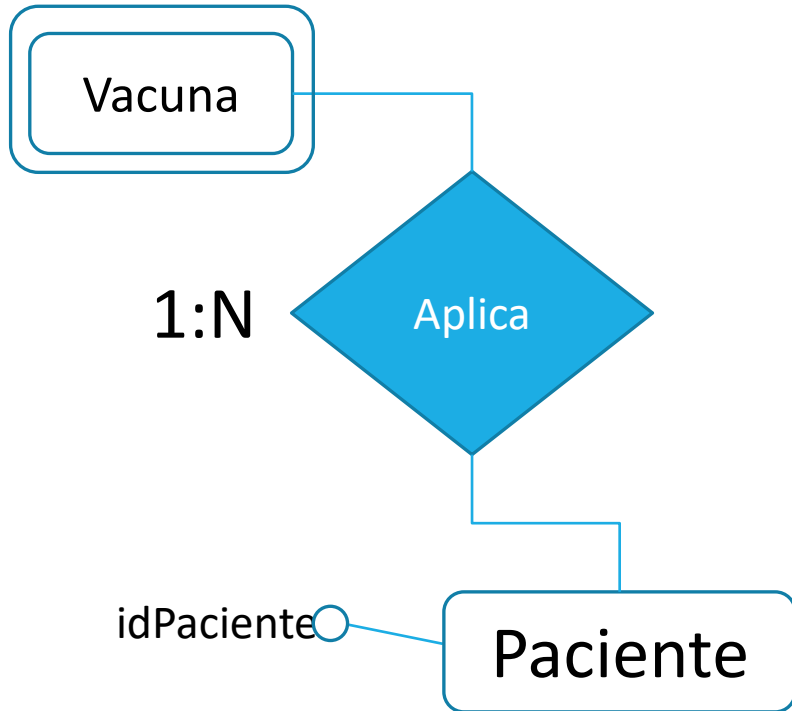
Atributos Multivaluados.



Telefono_duenio (idDuenio, teléfono, tipo)

Modelo relacional

Atributos Multivaluados.



Vacuna_Paciente

(idPaciente, nombreVacuna, diaAplicacion, mesAplicacion, añoAplicacion)

Diseño lógico de base de datos

Validar el esquema mediante la normalización.

La normalización se utiliza para **mejorar el esquema lógico**, de modo que satisfaga ciertas restricciones que **eviten la duplicidad** de datos.

El objetivo es generar un conjunto de esquemas de relaciones que permita almacenar información sin redundancias innecesarias, pero que también permita recuperar la información con facilidad.

- La normalización garantiza que el esquema resultante se encuentra más **próximo al modelo de la empresa**.
- Reglas que se basan en la **clave primaria** y las **dependencias funcionales**.
- Cada regla que se cumple aumenta el grado de normalización.