

Transformación

Modelo ER a Relacional Lógico

Tres reglas básicas

Las tres reglas básicas para convertir un esquema en el modelo E/R al relacional son las siguientes:

- Todo tipo de entidad se convierte en una relación.
 - La tabla o relación toma el nombre de la entidad. Los atributos de la entidad serán las columnas de la tabla y el atributo identificador principal será la clave primaria. En caso de que se indique lo contrario los atributos no identificadores podrán tomar valores nulos.
- Para todo tipo de relación 1:N se realiza lo que se denomina propagación de clave (regla general), o se crea una nueva relación.
 - Las interrelaciones 1:N o 1:1 se transforman propagando el atributo identificador principal de la entidad que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N. Si la relación fuese 1:1 la propagación de clave podría hacerse en cualquier sentido. El atributo propagado es una clave foránea que referencia a la tabla con cardinalidad máxima de 1.

Tres reglas básicas

Las tres reglas básicas para convertir un esquema en el modelo E/R al relacional son las siguientes:

- Para todo tipo de relación 1:N se realiza lo que se denomina propagación de clave (regla general) o se crea una nueva tabla
- Las interrelaciones 1:N o 1:1 se transforman propagando el atributo identificador principal de la entidad que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N. Si la relación fuese 1:1 la propagación de clave podría hacerse en cualquier sentido. El atributo propagado es una clave foránea que referencia a la tabla con cardinalidad máxima de 1.

Tres reglas básicas

- Todo tipo de relación N:N (muchos a muchos) se transforma en una relación.
- Las interrelaciones N:N se transforman en una tabla cuya clave primaria será la concatenación de los atributos principales de las entidades que se asocia; estos atributos serán claves foráneas que referencian a las respectivas tablas donde son claves primarias. Los atributos de la interrelación serán columnas de la tabla.

Reglas detalladas de Transformación

Transformación de relaciones 1:N

Existen dos soluciones:

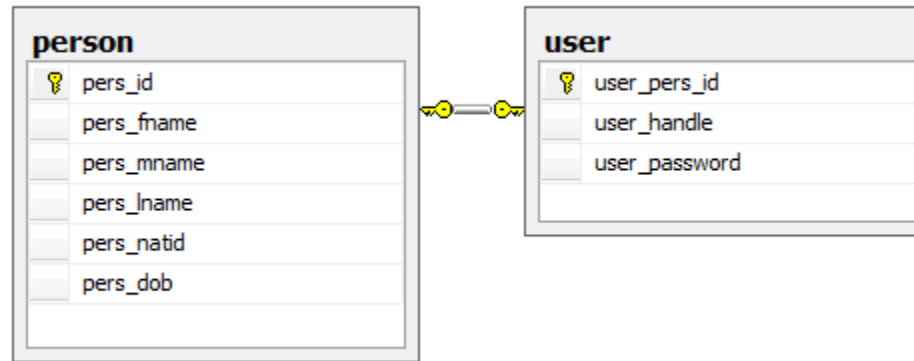
- Propagar la clave principal del tipo de entidad que tiene la cardinalidad máxima 1 a la que tiene N (propagación de clave). Esta es la regla habitual.
- Transformar la relación en una tabla como si se tratara de una relación M:M; pero ahora la clave primaria de la tabla creada es sólo la clave primaria de la tabla a la que le corresponde la cardinalidad n.

La opción b) se utiliza cuando:

- El número de ejemplares relacionados de la entidad que propaga su clave es muy pequeño y, por tanto, existirían muchos valores nulos en la clave propagada.
- Se prevé que la relación en un futuro se convertirá en un tipo M:M.
- La relación tiene atributos propios y no es deseable propagarlos (a fin de conservar la semántica).

Tres reglas básicas – Ejemplo

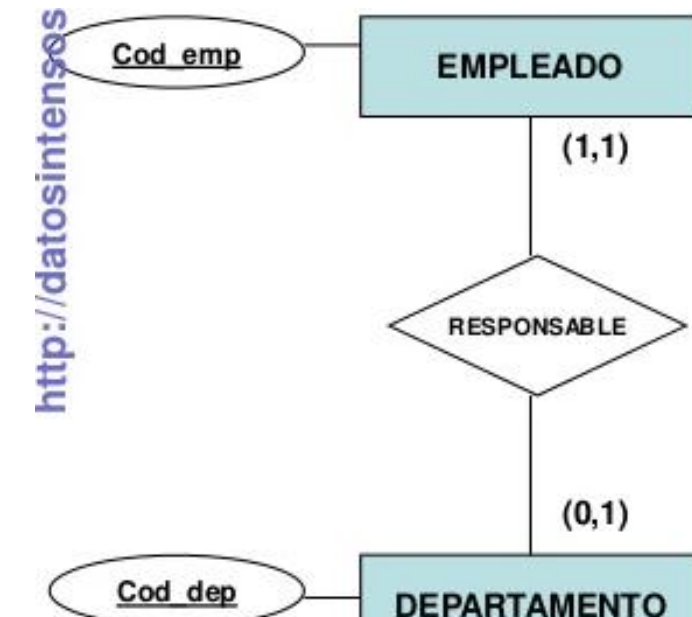
Ejemplo de una relación 1:1



Reglas detalladas de Transformación

Transformación de relaciones 1:1

- Si las entidades que participan en la interrelación poseen cardinalidades (0,1) y (1,1), conviene propagar la clave de la entidad con cardinalidades (1,1) a la tabla resultante de la entidad con cardinalidad.



Modelo Relacional

EMPLEADO (Cod_emp)

DEPARTAMENTO (Cod_dep, Cod_emp)

Clave Ajena

Reglas detalladas de Transformación

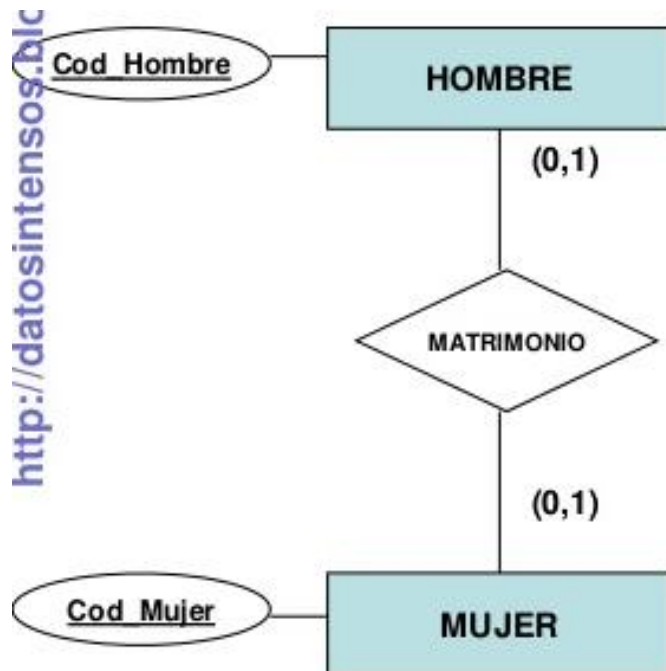
Transformación de relaciones 1:1

- Una relación de tipo 1:1 es un caso particular de una relación 1:N, por lo que se puede aplicar dos opciones : crear una nueva tabla o realizar propagación de clave, en este caso la propagación se puede hacer en ambos sentidos).
- Los criterios para aplicar una u otra regla y para propagar la clave se basan:
 - Las cardinalidades mínimas.
 - Recoger la mayor cantidad de semántica posible.
 - Evitar los valores nulos o aumentar la eficiencia.

Reglas detalladas de Transformación

Transformación de relaciones 1:1

- Si las entidades que se asocian poseen cardinalidades (0,1), suele ser conveniente transformar la relación 1:1 en una tabla.



Modelo Relacional

MATRIMONIO (Cod Hombre, Cod_Mujer)

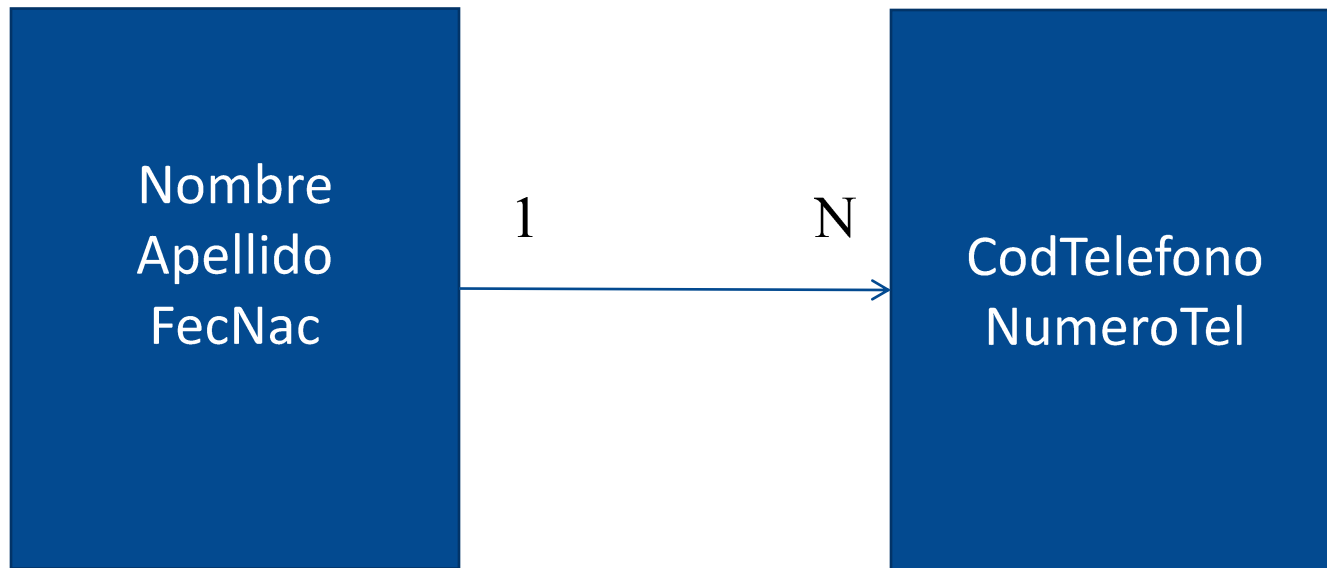
HOMBRE (Cod Hombre)

MUJER (Cod Mujer)

Clave Alternativa
UNIQUE, NOT NULL

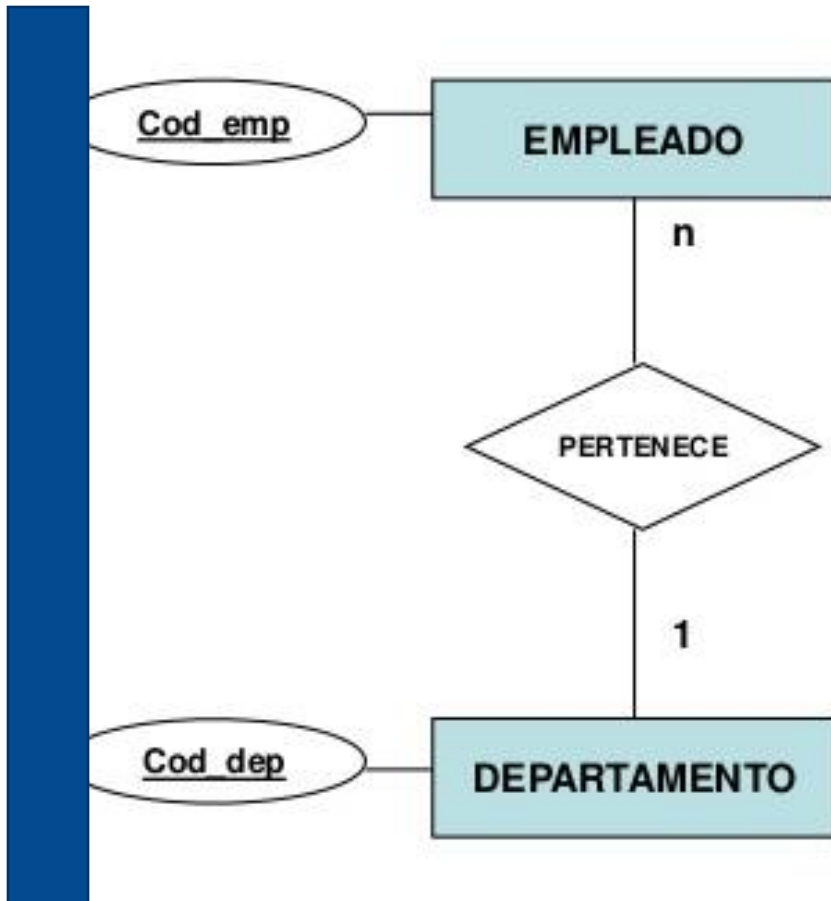
Tres reglas básicas – Ejemplo

- Ejemplo de una relación 1:N



Reglas detalladas de Transformación - Ejemplo

Transformación de relaciones 1:N



Modelo Relacional

EMPLEADO(Cod_emp,, Cod_dep)

DEPARTAMENTO (Cod_dep, ...)

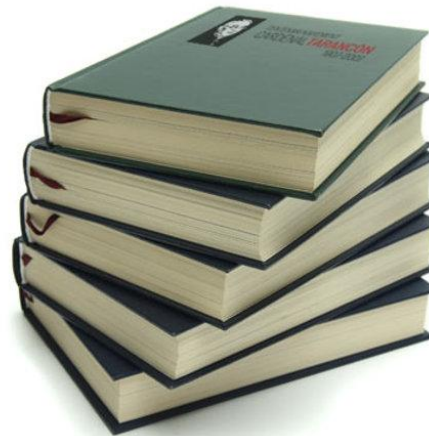
Reglas detalladas de Transformación

Transformación de Dependencias en Identificación y en Existencia



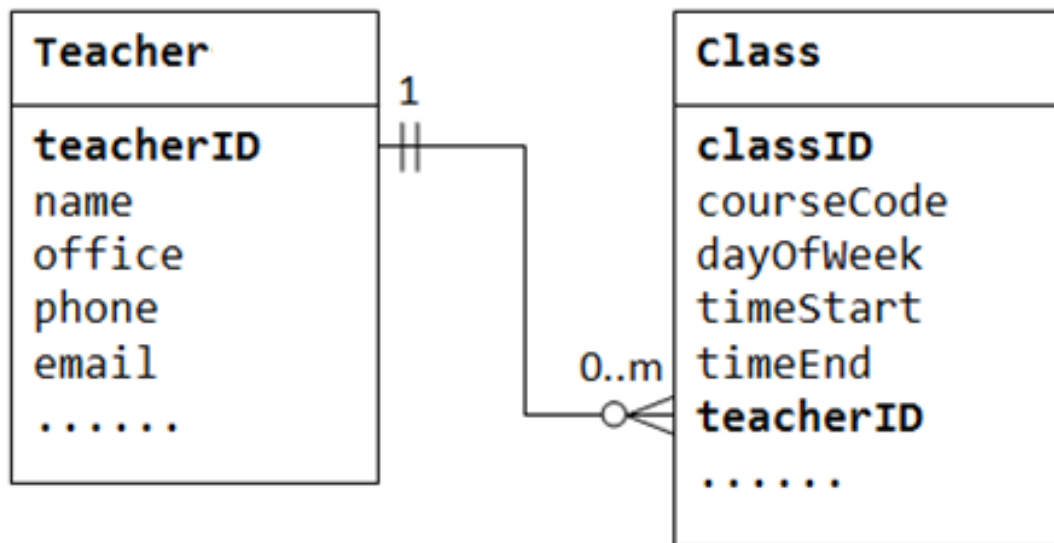
LIBRO(cod_libro,...)

EJEMPLAR(cod_libro, cod_ejemplar,...)



Reglas detalladas de Transformación - Ejemplo

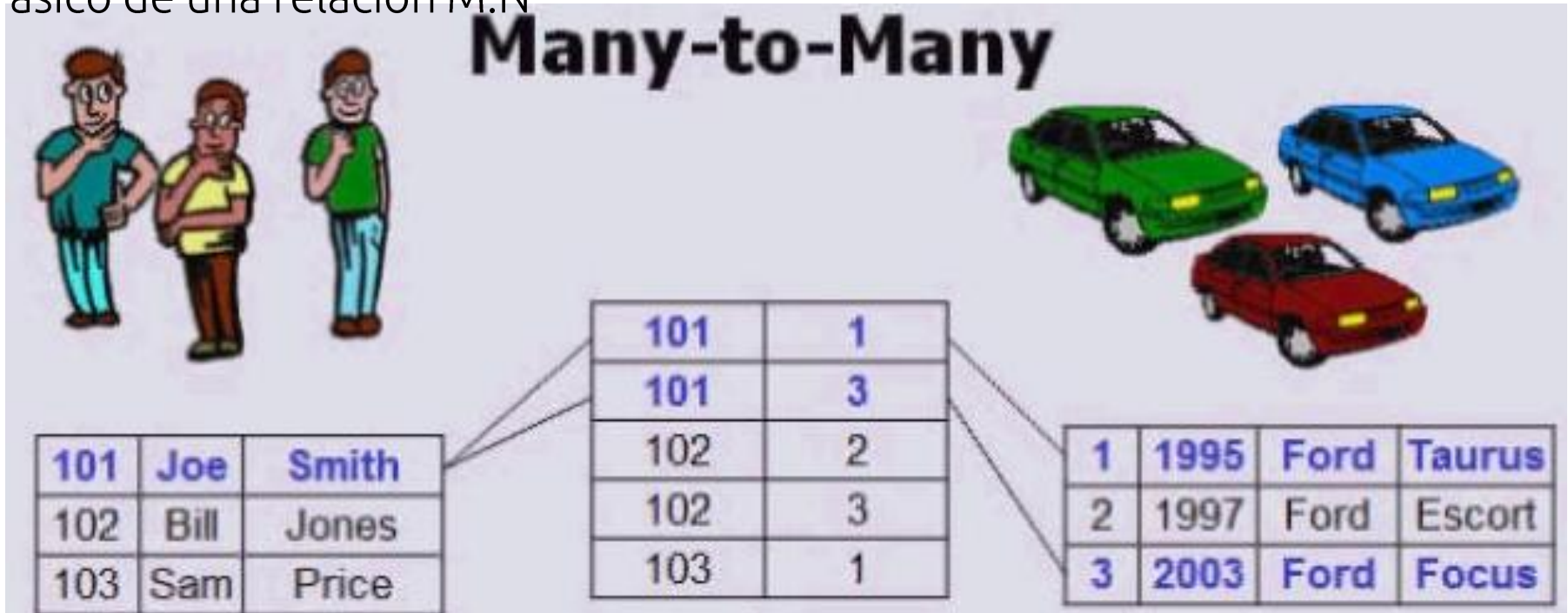
Transformación de relaciones 1:N



Tres reglas básicas – Ejemplo

Ejemplo b

básico de una relación M:N



Nueva tabla

Reglas detalladas de Transformación

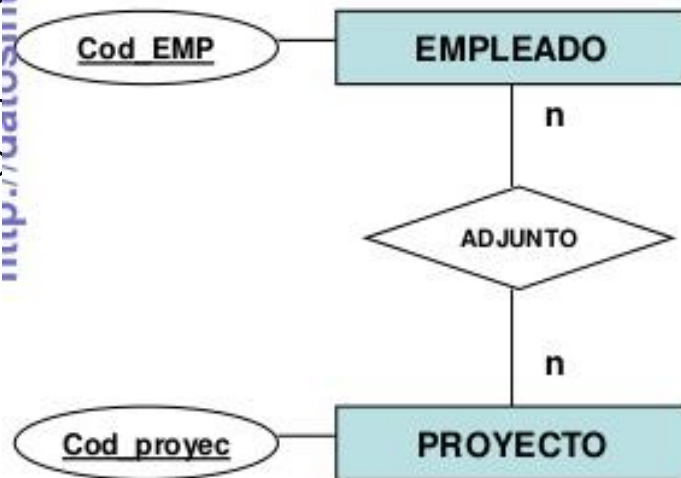
Transformación de relaciones M:M

- Un tipo de relación M:M se transforma en una tabla que tendrá como clave primaria COMPUESTA las claves primarias (PK) de los tipos de entidades que asocia.

- Además,

- para cada uno de los atributos que forman la clave primaria de esta tabla

- será necesario



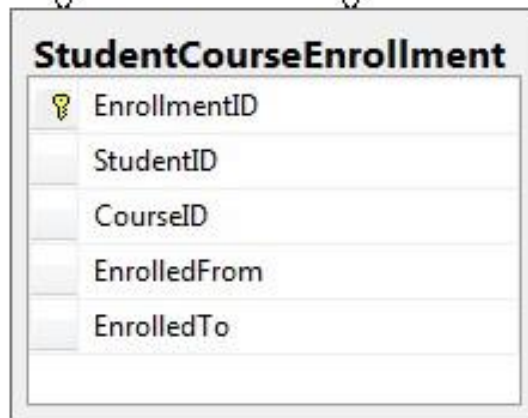
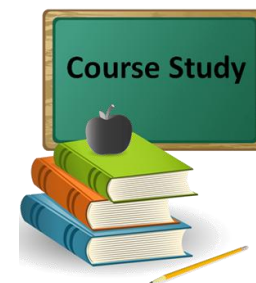
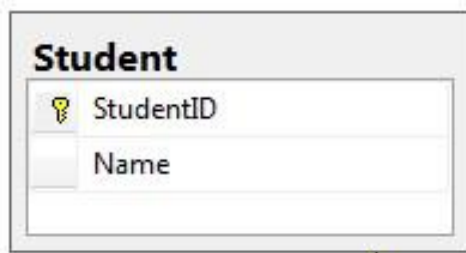
Modelo Relacional

EMPLEADO(Cod_EMP,)

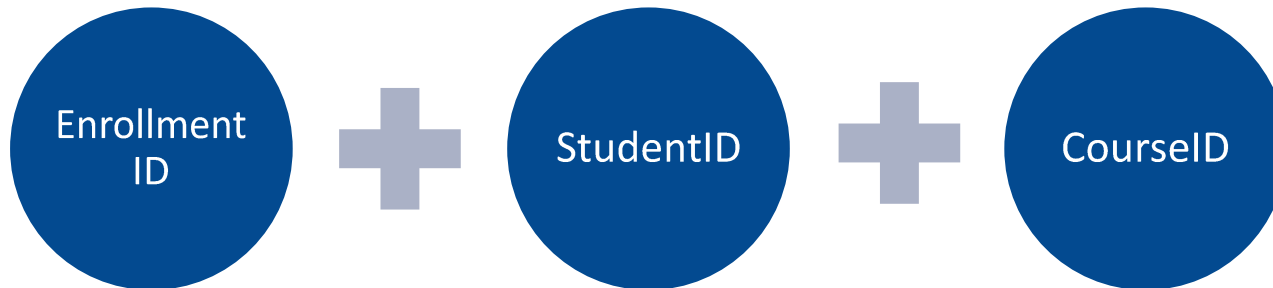
PROYECTO(Cod_proyec)

ADJUNTO(Cod_proyec, Cod_EMP)

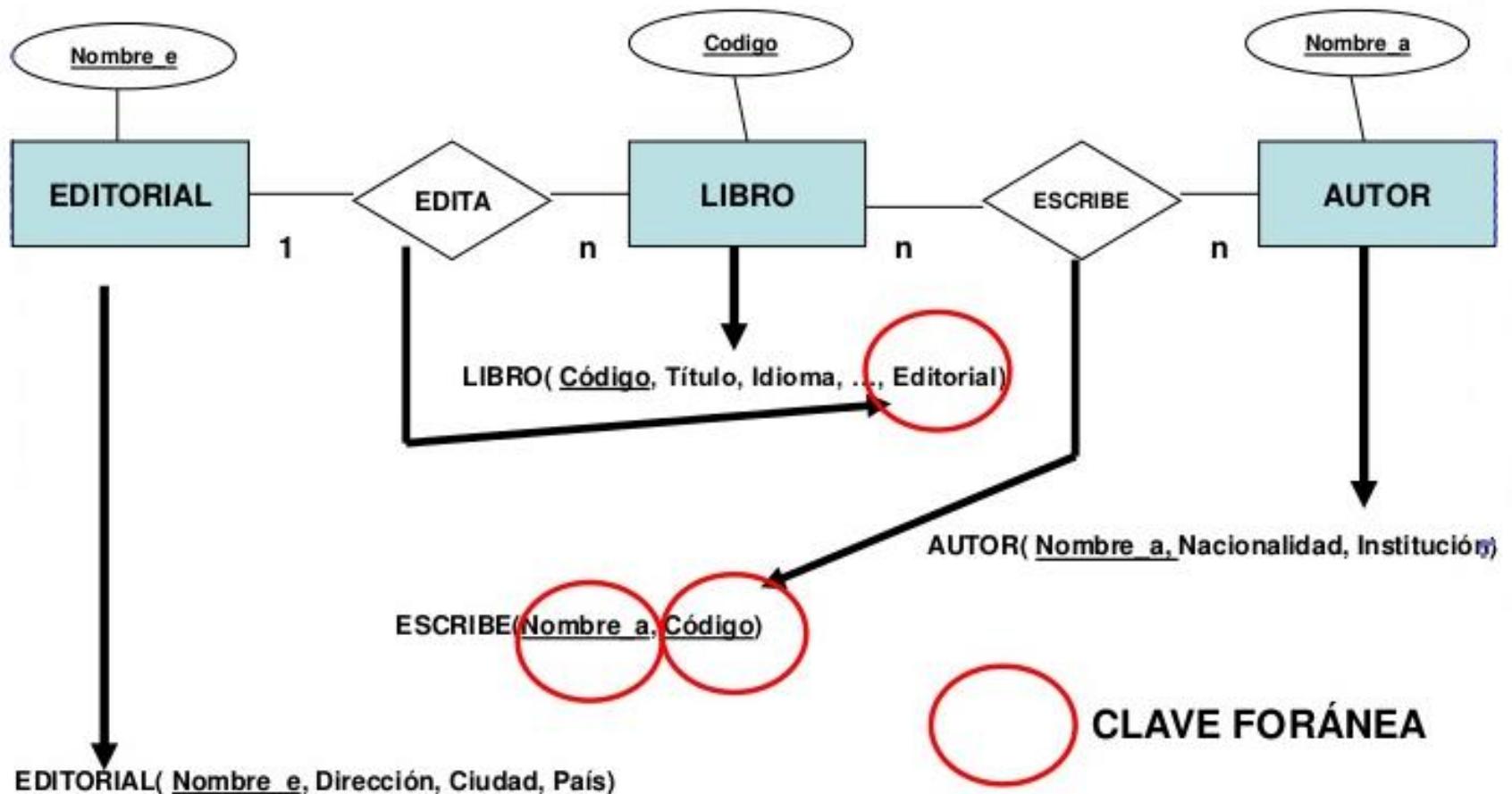
Reglas detalladas de Transformación - Ejemplo



Reglas detalladas de Transformación - Ejemplo



Tres reglas básicas – Ejemplo



Otras reglas de transformación

Reglas detalladas de Transformación

Transformación de Generalización (tipos y subtipos)

- Existen tres soluciones de transformación al modelo relacional:
 - Englobar todos los atributos de la entidad y sus subtipos en una sola tabla. Esta solución se utiliza cuando la cantidad de atributos de subtipos se diferencien muy poco y las relaciones que los asocian con el resto de las entidades del esquema sean las mismas para todos (o casi todos) los subtipos.
 - Crear una tabla para el supertipo y tantas tablas como subtipos haya, con sus atributos correspondientes. Esta solución es adecuada cuando existen muchos atributos distintos entre los subtipos y se desea mantener los atributos comunes a todos ellos en una misma tabla.

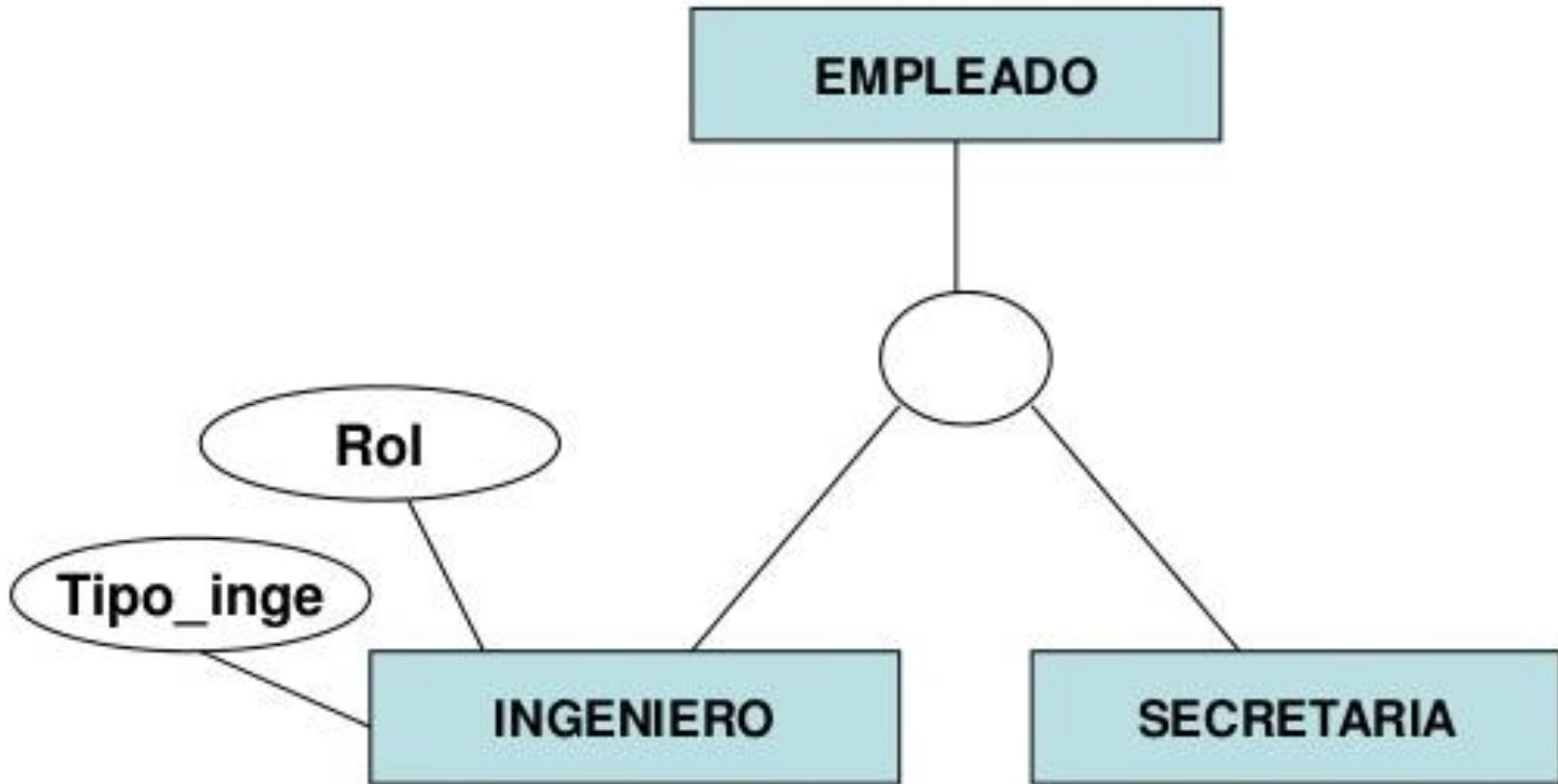
Reglas detalladas de Transformación

Transformación de Generalización (tipos y subtipos)

- Considerar tablas distintas para cada subtipo, que contengan, además de los atributos propios, los atributos comunes. Se elegirá esta opción cuando se den las mismas condiciones anteriores.

Reglas detalladas de Transformación

Transformación de Generalización (tipos y subtipos) – Ejemplo



Reglas detalladas de Transformación

Transformación de Generalización (tipos y subtipos) – Ejemplo

Opción a) **EMPLEADO** (Cod_emp, nombre,, tipo, Tipo_ing, Rol,...)

Opción b) **EMPLEADO** (Cod_emp, Nombre,)

INGENIERO (Cod_emp,)

SECRETARIA (Cod_emp,)

Opción c) **INGENIERO** (Cod_emp, Nombre,, Tipo_ing,)

SECRETARIA (Cod_emp, Nombre,)

Reglas detalladas de Transformación

Transformación de Generalización (tipos y subtipos)

- Se puede elegir cualquiera de las tres estrategias para la transformación de un tipo y sus subtipos al modelo relacional:
 - desde lo semántico la opción b es la mejor.
 - desde la eficiencia deberá tenerse en cuenta que:
 - Opción a: el acceso a una fila es mas rápido a nivel operativo del motor (no hace falta combinar varias tablas).
 - Opción b: la menos eficiente aunque es la mejor desde el mundo semántico, esta opción puede mejorar el desempeño creando una vista materializada que refleje la opción a.
 - Opción c: Con esta solución se aumenta la eficiencia ante determinadas consultas (las que afectan a todos los atributos, tanto comunes como propios, de un subtipo) pero se disminuye ante otras.

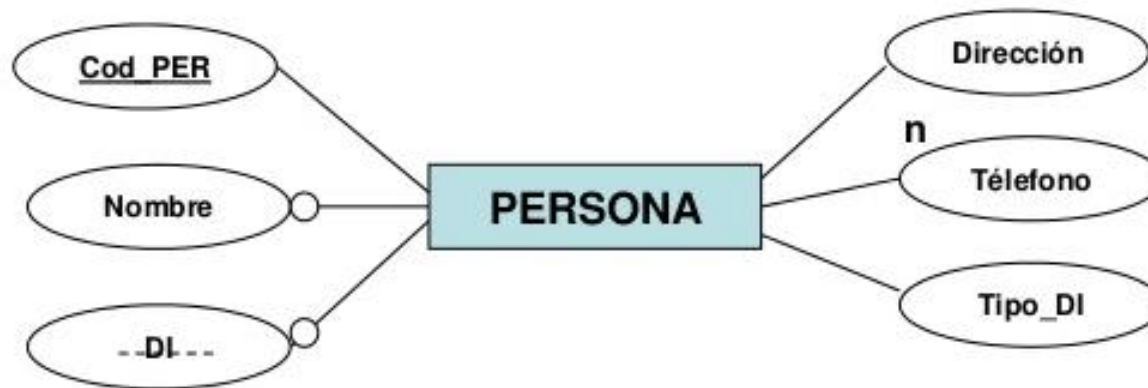
Reglas detalladas de Transformación

Transformación de Generalización (tipos y subtipos)

- Cual estrategia elegir depende de lo que le interese al USUARIO generalmente es la INMEDIATEZ.

Esquema conceptual y Diccionario de datos

Transformación de Atributos - Ejemplo



Diccionario de datos esquema conceptual

ENTIDAD	ATRIBUTOS	DOMINIOS	DESCRIPCION	VALORES		RESTRICCIONES	
				DEFECTO	VALORES		
PERSONA	<DESCRIPCION DE LA ENTIDAD PERSONA>						
	Cod_PER	Numerico	{Digito}8	Números autoincrementales			NOT NULL, PK AUTOINCREMENTO
	DI	Cadena(20)	{Digito}8	Número de identificación del documento			UNIQUE
	Tipo_DI	Cadena(4)	{Letra}	Tipo de documento	NA	CC, TI,NA	Chequeos de valores permitidos
	Direccion	Cadena(40)	{Digito} + {Letra}	Dirección de ubicación	NA		
	Nombre	Cadena(30)	{Letra}				NOT NULL

Transformación de Atributos - Ejemplo

Diccionario de datos esquema conceptual

ENTIDAD	ATRIBUTOS	DOMINIOS	DESCRIPCION	VALORES		RESTRICCIONES
				DEFECTO	VALORES	
TELEFONO	<DESCRIPCION DE LA ENTIDAD TELEFONO>					
	Cod_PER	Numerico	{Digito}8	Números relaionados con Cod_PER		PK ,FK AUTOINCREMENTO
	Numero	Cadena(14)	{Digito} + {Símbolos}	Número de teléfono		PK
[<EL ATRIBUTO Cod_PER ES UNA CLAVE FORANEA CON LA ENTIDAD PERSONA]						
[<LOS ATRIBUTOS Cod_PER y Numero SON LA CLAVE PRIMARIA]						

```
CREATE TABLE persona
(cod_per INTEGER PRIMARY KEY,
nombre VARCHAR(30) NOT NULL,
"DI" VARCHAR(20) UNIQUE,
"tipo_DI" VARCHAR(4) ,
Direccion VARCHAR(40)',
CONSTRAINT "CK_tipo_DI"
CHECK ( "tipo_DI" IN ('CC', 'TI', 'NA' ) )
);
```

```
CREATE TABLE telefono
(cod_per INTEGER,
numero VARCHAR(14),
CONSTRAINT "PK_telefono"
PRIMARY KEY(cod_per, numero),
CONSTRAINT FK_telefono_persona
FOREIGN KEY (universidad)
REFERENCES "persona"(cod_per)
);
```

Reglas detalladas de Transformación - Ejemplo

ENTIDAD	ATRIBUTOS	DOMINIOS	DESCRIPCION	VALORES		RESTRICCIONES
				DEFECTO	VALORES	
EMPLEADO	<DESCRIPCION DE LA ENTIDAD EMPLEADO>					
	Cod_EMP	Numerico	{Digito}8	Números		PK , AUTOINCREMENT O

ENTIDAD	ATRIBUTOS	DOMINIOS	DESCRIPCION	VALORES		RESTRICCIONES
				DEFECTO	VALORES	
PROYECTO	<DESCRIPCION DE LA ENTIDAD EMPLEADO>					
	Cod_proyec	Numerico	{Digito}8	Números		PK , AUTOINCREMENTO

ENTIDAD	ATRIBUTOS	DOMINIOS	DESCRIPCION	VALORES		RESTRICCIONES
				DEFECTO	VALORES	
ADJUNTO	<DESCRIPCION DE LA ENTIDAD EMPLEADO>					
	Cod_proyec	Numerico	{Digito}8	Números		PK ,FK
	Cod_EMP	Numerico	{Digito}8	Números		PK ,FK
[<EL ATRIBUTO Cod_proyec ES UNA CLAVE FORANEA CON LA ENTIDAD PROYECTO]						
[<EL ATRIBUTO Cod_EMP ES UNA CLAVE FORANEA CON LA ENTIDAD EMPLEADO]						
[<LOS ATRIBUTOS Cod_EMP y Cod_proyec SON LA CLAVE PRIMARIA COMPUESTA]						

Reglas detalladas de Transformación

Atributos Multivaluados

Agenda

<u>nombre</u>	<u>fecha_nacimiento</u>	<u>estado_civil</u>
Mengano	15/12/1985	soltero
Fulano	13/02/1960	casado
Tulana	24/06/1975	soltera

Teléfonos

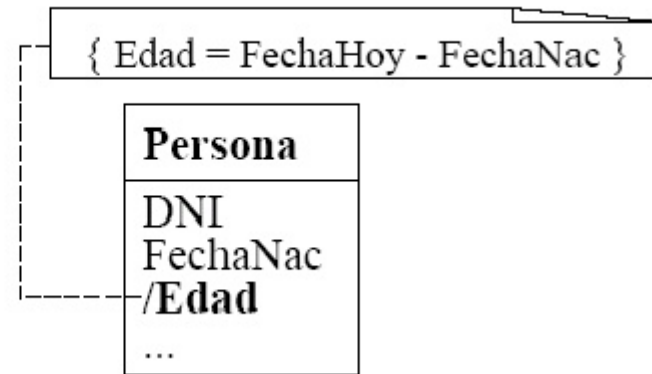
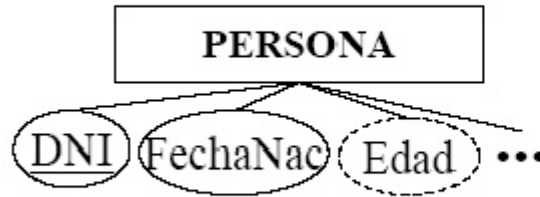
<u>nombre</u>	<u>teléfono</u>
Mengano	12322132
Fulano	13321232
Fulano	25565445
Fulano	36635363
Tulana	45665456



Reglas detalladas de Transformación

Atributos Derivados

Calculus



Gracias - ¿Preguntas?