

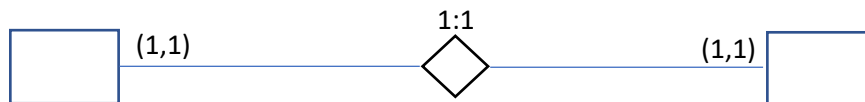
Siendo supertipo A y subtipos B y C.

GENERALIZACIONES – MODELO RELACIONAL	Total (con circulito)	Parcial (sin circulito)
Solapada (sin arquito)	ABC(...)	A BC(...)
Exclusiva (con arquito)	AB(...) AC(...)	A(...) B(...id_A) C(...id_A)

Todas las claves ajenas son UNIQUE (UQ) en relaciones 1:1.

#### ER A MODELO RELACIONAL RELACIONES 1:1

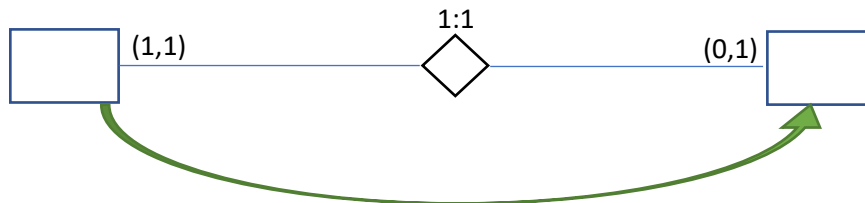
- CASO 1



Caso 1: Ambas entidades tienen cardinalidad (1,1) → SE FUSIONAN las entidades y se crea una única tabla (Entidad1Entidad2) y la PK puede ser cualquiera de las PK de esas dos entidades.

EN TOTAL SÓLO HABRÍA UNA TABLA.

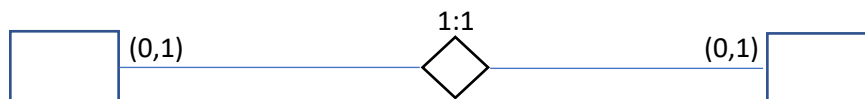
- CASO 2



Caso 2: Una de las entidades tiene cardinalidad (0,1) y la otra (1,1) → La PK de la entidad que tiene la cardinalidad (1,1) pasa a la entidad que tiene la cardinalidad (0,1) como FK de esta última.

HABRÍA DOS TABLAS.

- CASO 3



Caso 3: Las dos entidades tienen cardinalidad (0,1) → Se crea una nueva tabla para la relación tal y como se hace para las N:M, habiendo un total de 3 tablas. Una para cada entidad y la nueva tabla para la relación, que tendría como clave primaria una de las claves foráneas, y la otra clave foránea sería marcada como **unique**.

HABRÍA TRES TABLAS.

- TODA ENTIDAD SE CONVIERTE EN UNA RELACIÓN (AQUÍ RELACIÓN SIGNIFICA TABLA)
- TODA INTERRELACION N:M SE TRANSFORMA A UNA RELACIÓN (RELACIÓN SIGNIFICA TABLA)
- TODA INTERRELACION 1:N SE TRADUCE POR PROPAGACIÓN DE CLAVES O SE CREA UNA NUEVA RELACIÓN (TABLA)
- ENTIDADES DÉBILES: AL PROPAGAR LA CLAVE DEBEMOS ESPECIFICAR ELIMINACIÓN EN CASCADA (ON DELETE CASCADE)
- DEPENDENCIA EN IDENTIFICACIÓN: SE PROPAGA LA CLAVE DE LA ENTIDAD FUERTE Y FORMARÁ PARTE DE LA PK DE LA ENTIDAD DÉBIL.

---

#### ER A MODELO RELACIONAL RELACIONES TERNARIAS Y N-ARIAS

- Se resuelven como las N:M
- Una nueva tabla con las claves primarias de las tres tablas que se relacionan
- Su clave primaria está formada al menos por las claves de las entidades "N" que asocia
- Cada uno de las columnas propagadas es clave ajena de la tabla que referencia

---

#### TRANSACT SQL

##### Operadores

- Relacionales
  - =, <>, <, >, <=, >=, !=, !< y !>
- Aritméticos
  - +, -, \*, /, %
- Lógicos
  - **ALL, AND, ANY, BETWEEN, EXISTS, IN, LIKE, NOT, OR, SOME**
- Asignación
  - =
- Concatenación de cadenas
  - +
- Bit a bit
  - & (AND), | (OR) y ^ (XOR)
- Unarios
  - +, - (negativo) y ~ (NOT)
- Patrones de LIKE
  - % Cualquier cadena
  - \_ Cualquier carácter
  - [patron] Conjunto de caracteres
  - [^patron] Conjunto complementario
- Ejemplo:
  - LIKE 'A%' → Cadena que comience por A
  - LIKE '\_on' → Ion, Pon, eon, \*on, 5on, etc.
  - LIKE '[AEIOUaeiou]' → Una vocal
  - LIKE '[^0-9]%' → Cadena que no comience por un número

- FOREIGN KEY
  - (columna(s) de esta tabla) REFERENCES otraTabla[ ( columna(s) de la otra ) ]  
[ ON DELETE {NO ACTION | CASCADE | SET NULL | SET DEFAULT} ]  
[ ON UPDATE {NO ACTION | CASCADE | SET NULL | SET DEFAULT} ]
- ALTER TABLE:  
(una columna es un atributo)
  - Podemos añadir una columna o una restricción
    - ALTER TABLE MiTabla ADD COLUMN...
    - ALTER TABLE MiTabla ADD CONSTRAINT...
  - Redefinir una columna
    - ALTER TABLE MiTabla ALTER COLUMN...
  - Borrar una columna o restricción
    - ALTER TABLE MiTabla DROP COLUMN...
    - ALTER TABLE MiTabla DROP CONSTRAINT...
    - Cuando se borran objetos no puede haber otros objetos dependientes de ellos

## TIPOS DE DATOS PARA TRANSACT-SQL

### NUMERICOS EXACTOS

- bigint, int, smallint, tinyint.
- decimal y numeric.
  - –decimal[( p[, s])] y numeric[( p[, s])] – P indica el número total de cifras significativas y s el número de cifras decimales
- bit (acepta 1, 0 o NULL)
- money y smallmoney (de - 214.748,3648 a 214.748,3647)

### NUMERICOS APROXIMADOS

- float y real
- datetime, datetime2, datetimeoffset, smalldatetime, date, time

### CADENAS DE CARACTERES

- char (n) –Cadena de longitud fija
- varchar (n) –Cadena de longitud variable –n entre 1 y 8.000
- varchar (max) –Hasta 232-1 bytes –Sustituye a text

### CADENAS DE CARACTERES UNICODE

- nchar (n) –Cadena de longitud fija
  - nvarchar (n) –Cadena de longitud variable –n entre 1 y 4.000
  - nvarchar (max) –Hasta 232-1 bytes –Sustituye a text
-

## NORMALIZACIÓN

- 1FN: Atributos atómicos (no repetitivos)
- 2FN: Dependencia completa (clave completa)
- 3FN: Eliminar dependencias transitivas
- **Para cada forma normal es requisito que se cumpla la forma normal anterior.**

## DEPENDENCIA FUNCIONAL:

Un atributo Y depende funcionalmente de X si y sólo si para cada valor de X le corresponde un único valor de Y

$$\underline{X \rightarrow Y}$$

Matricula -> MarcaVehiculo (MarcaVehiculo depende funcionalmente de Matricula)

NumeroSS -> Estatura (Estatura depende funcionalmente de NumeroSS)

## DEPENDENCIA FUNCIONAL COMPLETA:

Si el atributo **X** es compuesto  $X(X_1, X_2) \rightarrow Y$  depende del conjunto de los componentes de X, no de un subconjunto de ellos

((Que si la clave primaria es compuesta, no puede haber atributos que dependan sólo de una de las partes de la clave. Que todos los atributos deben depender de la clave COMPUESTA, de todos los atributos que la formen)

$$\begin{array}{cc} X(X_1, X_2) & X_1 \nrightarrow Y \\ X \rightarrow Y & X_2 \nrightarrow Y \end{array}$$

## DEPENDENCIA TRANSITIVA:

Si Y depende de X, Z depende de Y y X no depende de Y  $\rightarrow$  decimos que Z depende de X a través de Y, ES DECIR, DECIMOS QUE Z DEPENDE TRANSITIVAMENTE DE X.

$$\begin{array}{cc} X \rightarrow Y & Y \nrightarrow X \\ Y \rightarrow Z & X \dashrightarrow Z \end{array}$$

**PRIMERA FORMA NORMAL**

- Eliminar grupos repetitivos
  - ALUMNO (NOMBRE, APELLIDOS, F\_NACIMIENTO, DIA\_SANTO, NOMBRE\_GRUPO, NOMBRE\_TUTOR, **NOMBRE\_HERMANO**)

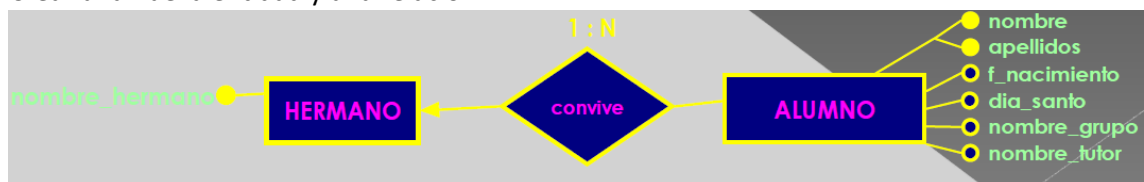
Nombre	Apellidos	F_Nac	Onomástica	Grupo	Tutor	Hermano
Pancho	Colate	12/6/1997	OCT04	1° C	Agripina	Gabriel
Olga	Rapata	21/2/1996	MAY22	3° A	Restituto	NULL
Alfredo	Loroso	8/10/1995	ENE12	3° A	Restituto	Glicería Gervasio

SOLUCIÓN PRIMERA FORMA NORMAL:

- Una ocurrencia con cada hermano, incluyéndolo en la clave

Nombr e	Apellidos	F_Nac	Onoma stica	Grupo	Tutor	Herman o
Pancho	Colate	12/6/1997	OCT04	1° C	Agripina	<b>Gabriel</b>
Olga	Rapata	21/2/1996	MAY22	3° A	Restituto	<b>NULL</b>
Alfredo	Loroso	8/10/1995	ENE12	3° A	Restituto	<b>Glicería</b>
Alfredo	Loroso	8/10/1995	ENE12	3° A	Restituto	<b>Gervasio</b>

- Crear una nueva entidad y una relación



Nombre	Apellidos	F_Nac	Onomástica	Grupo	Tutor
Pancho	Colate	12/6/1997	OCT04	1° C	Agripina
Olga	Rapata	21/2/1996	MAY22	3° A	Restituto
Alfredo	Loroso	8/10/1995	ENE12	3° A	Restituto

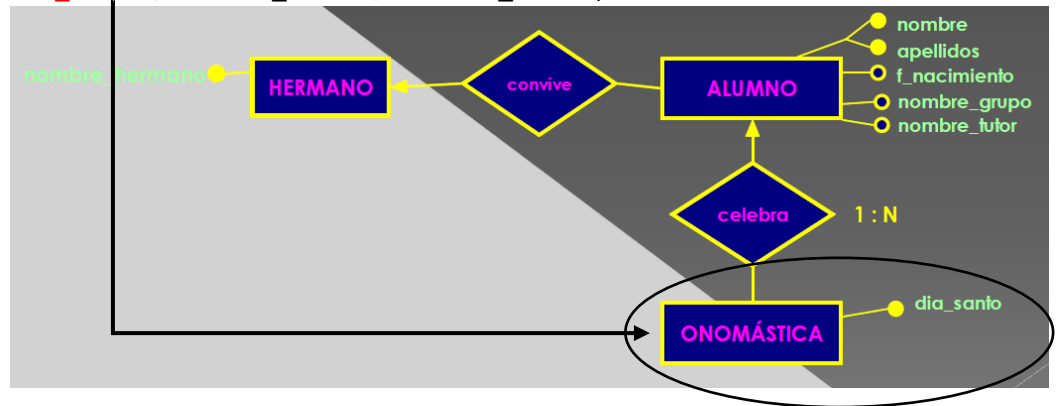
Nombre	Apellidos	Hermano
Pancho	Colate	<b>Gabriel</b>
Alfredo	Loroso	<b>Glicería</b>
Alfredo	Loroso	<b>Gervasio</b>

Si el grupo tiene un número más o menos fijo de repeticiones (si fuera chino), podemos añadir atributos.

Nombre	Apellidos	F_Nac	.....	Hermano1	Hermano 2
Pancho	Colate	12/6/1997	.....	Gabriel	NULL
Olga	Rapata	21/2/1996	...	NULL	NULL
Alfredo	Loroso	8/10/1995	.....	Glicería	Gervasio

### SEGUNDA FORMA NORMAL:

- Eliminar dependencias funcionales incompletas
  - ALUMNO (NOMBRE, APELLIDOS, F\_NACIMIENTO, **DIA\_SANTO**, NOMBRE\_GRUPO, NOMBRE\_TUTOR)



### TERCERA FORMA NORMAL:

- Eliminar dependencias funcionales transitivas
  - ALUMNO (NOMBRE, APELLIDOS, F\_NACIMIENTO, NOMBRE\_GRUPO, **NOMBRE\_TUTOR**)

(en la solución vemos que NOMBRE\_GRUPO seguirá estando en la tabla ALUMNO pero ahora como CLAVE FORÁNEA proveniente de la nueva tabla GRUPO).

