Introducción a las Bases de Datos y Bases de Datos

ESQUEMAS DE TABLAS

TEORÍA 2

2017

Tecnicaturas TUPAR y TUDAI

DISEÑO LÓGICO

Modelo de Entidades y Relaciones

Reglas de Transformación

Esquema Lógico *según el* **Modelo Relacional**

Esquema Post-relacional (tablas en SQL)

LENGUAJE SQL

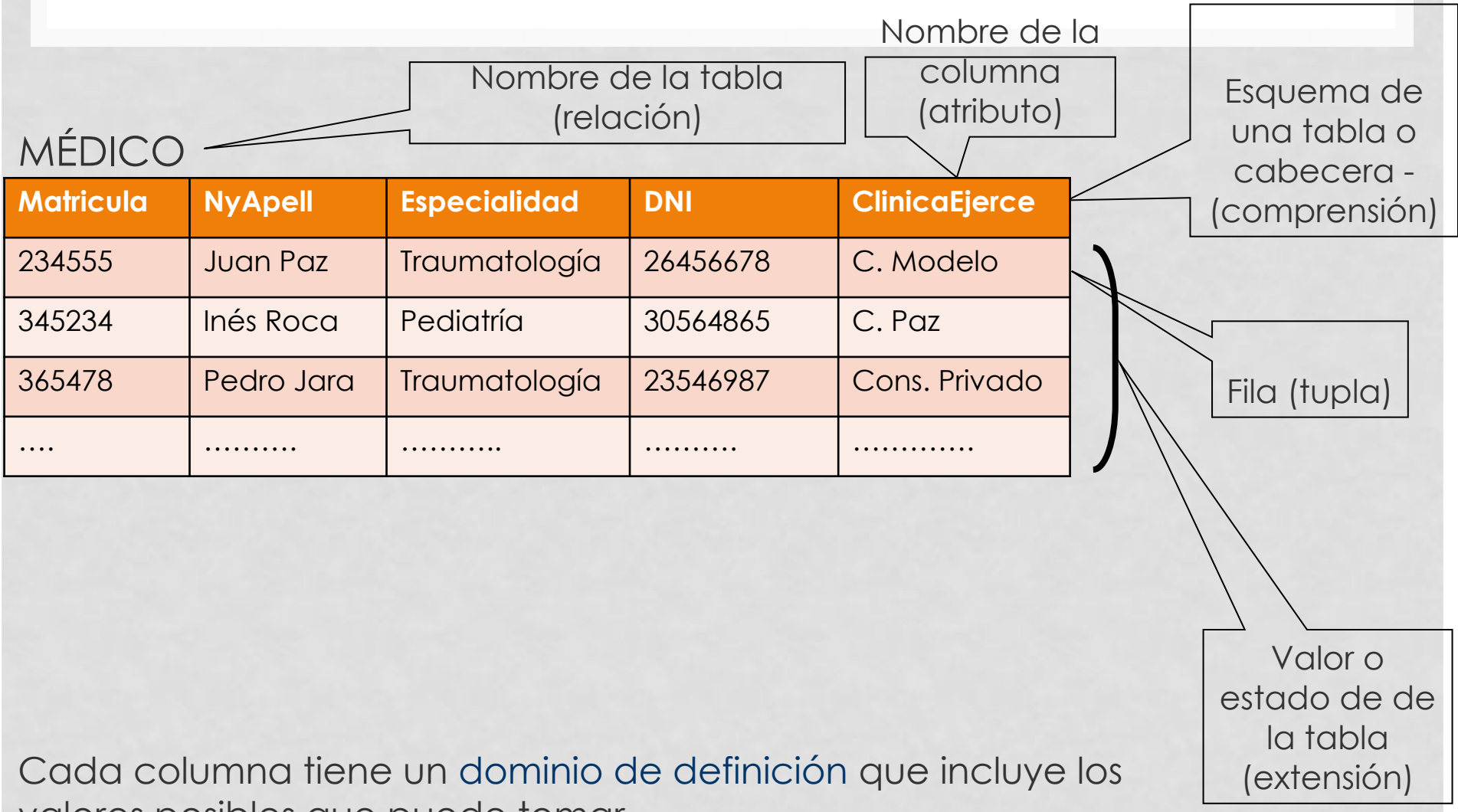
- La definición de los datos se realiza a través de sentencia de DDl
  - Sus comandos permiten definir la semántica del esquema relacional: que tablas o relaciones se establecen, sus posibles valores (dominios), asociaciones, restricciones, etc.
  - Los datos o información de dichas tablas las guarda el SGBD en tablas propias denominadas tablas de **metadatos**.
  - Él nombre de las tablas deben ser único dentro de cada esquema.
  - Una tabla en una base de datos relacional es similar a una tabla en papel, posee columnas y filas.

ESQUEMA DE BASE DE DATOS

- Una base de datos relacional consiste en un conjunto de tablas relacionales (o relaciones) cada una de las cuales contiene un conjunto de tuplas.
- La clave y clave extranjera (en adelante abreviado como F. K. foreign key).
- Una clave (o clave alternativa) en una tabla es un subconjunto de las columnas de la tabla que identifica a cada tupla.
- Un F. K. en una tabla T es un conjunto de columnas F que hace referencia a la clave de otra tabla T 'e impone una restricción (Restricción de Integridad Referencial)

EJEMPLO

Las relaciones pueden visualizarse en forma tabular



MÉDICO

Matricula	NyApell	Especialidad	DNI	ClinicaEjerce
234555	Juan Paz	Traumatología	26456678	C. Modelo
345234	Inés Roca	Pediatría	30564865	C. Paz
365478	Pedro Jara	Traumatología	23546987	Cons. Privado
….	…….	…….	…….	……

Cada columna tiene un dominio de definición que incluye los valores posibles que puede tomar

TABLAS EN SQL

- En SQL no existe un orden para las filas de una tabla. Cuando se lee una tabla, las filas aparecerán en un orden aleatorio, a menos que se especifique uno.
- Las columnas contienen la información de los campos de la tabla: nombre, tipo de dato y restricciones asociadas a la columna.
- Las filas contiene los registros o instancias.

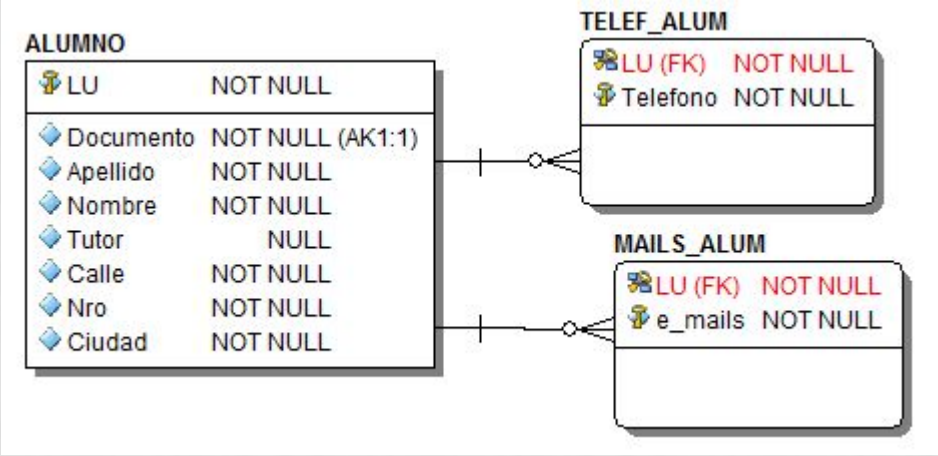
REGLAS DE TRANSFORMACIÓN DE ENTIDADES

Las reglas de transformación del DERExt al Esquema Relacional de Bases de Datos para entidades son las siguientes:

- Se crea **una tabla por cada entidad**, con el mismo nombre de la entidad.
- El **identificador de la entidad** se transforma en la **clave** de dicha tabla.
- Todo **atributo simplemente valuado** de la entidad se transforma en un atributo de dicha tabla.
- Los **atributos compuestos** se despliegan en sus partes componentes, como si fueran univaluados.
- Los **atributos obligatorios** llevan una leyenda de NOT NULL.
- Los **atributos opcionales** se indican de la misma manera que los obligatorios sin la leyenda.
- Los **atributos multivaluados** se proyectan en otra tabla conjuntamente con la clave de la entidad o de la (inter)relación.

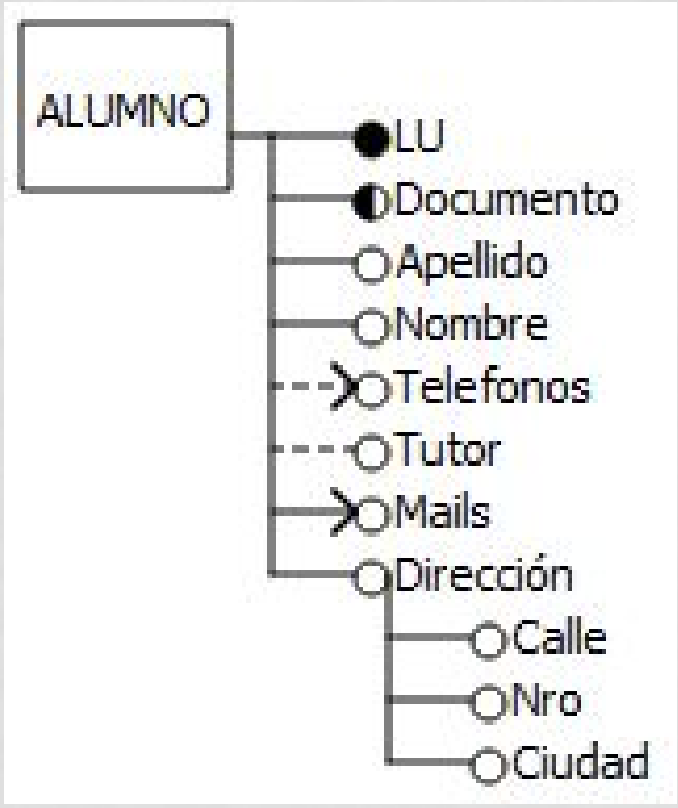
DERIVACIÓN DE ENTIDADES

Aplicando las reglas anteriores...



Relaciones entre tablas:

- ***Identificatorias***
- ***No Identificatorias***



CREACIÓN DE TABLAS

- Cada columna debe tener un determinado tipo de dato.
- El tipo de dato limita el conjunto de valores posibles que se pueden asignar a una columna


```
CREATE TABLE ALUMNO(
   LU              integer          NOT NULL,
   Documento          integer          NOT NULL,
   Apellido        varchar(30)    NOT NULL,
   Nombre          varchar(30)    NOT NULL,
   Tutor           varchar(50),
   Calle           varchar(40)    NOT NULL,
   Nro             integer          NOT NULL,
   Ciudad             varchar(6 0)    NOT NULL,
   CONSTRAINT PK_ALUMNO PRIMARY KEY (LU)
);
```

O puede colocarse la definición de la clave primaria en sentencia aparte

```
ALTER TABLE ALUMNO
ADD  CONSTRAINT PK_ALUMNO PRIMARY KEY (LU);
```

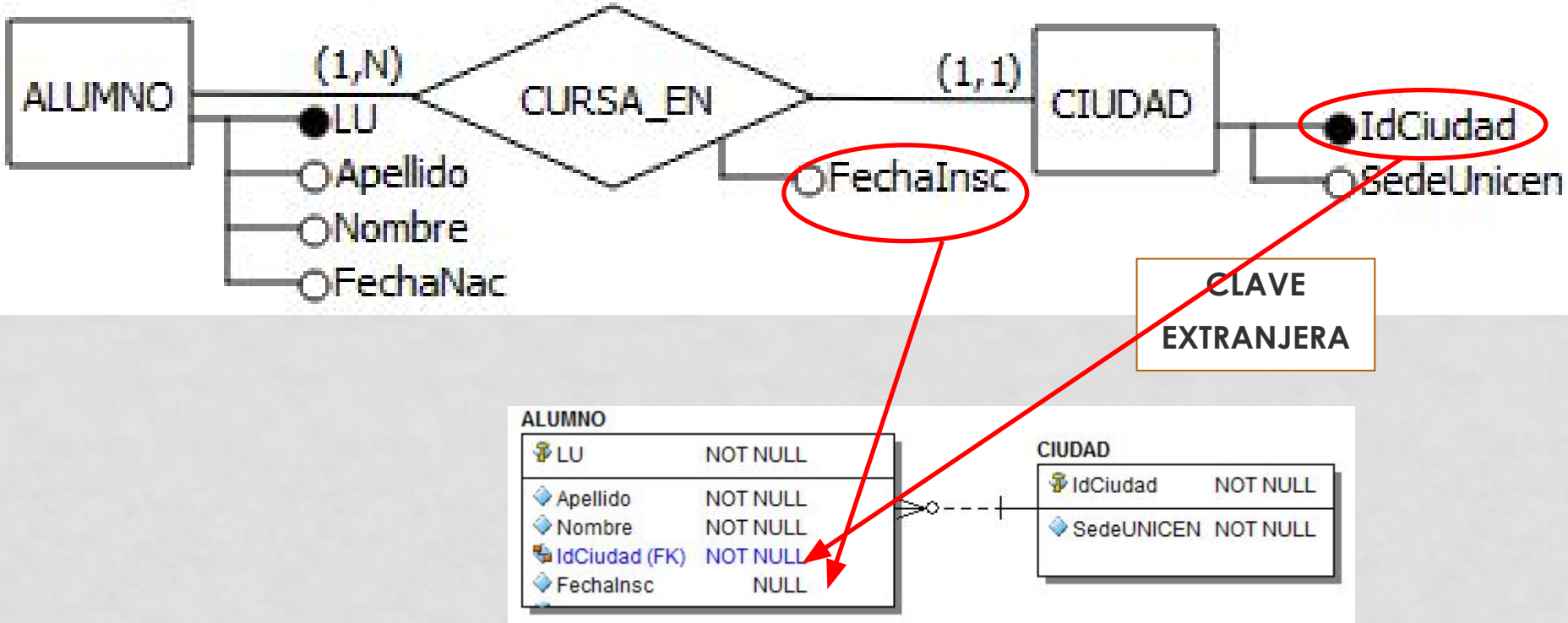
TIPOS DE DATOS

Tipos de datos Postgresql

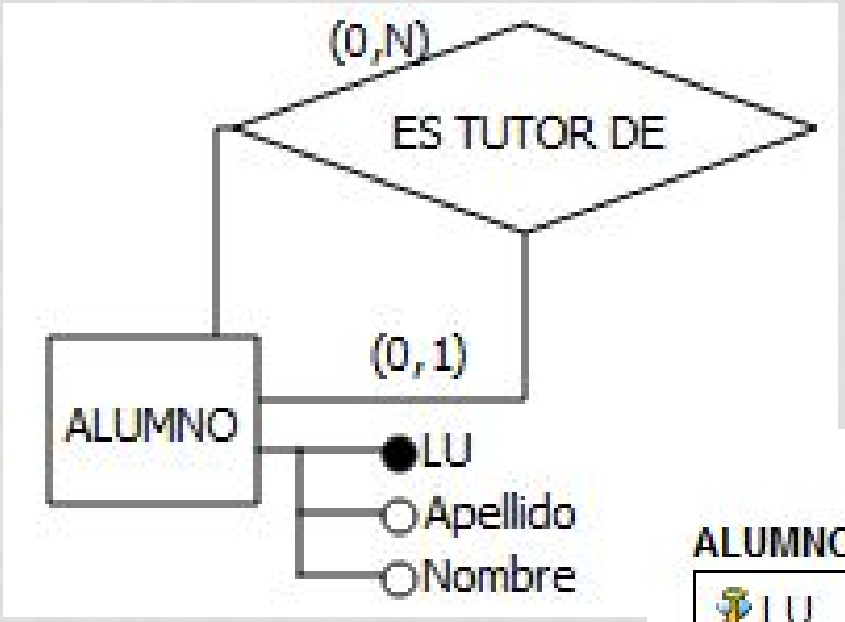
https://www.postgresql.org/docs/9.5/static/datatype.html

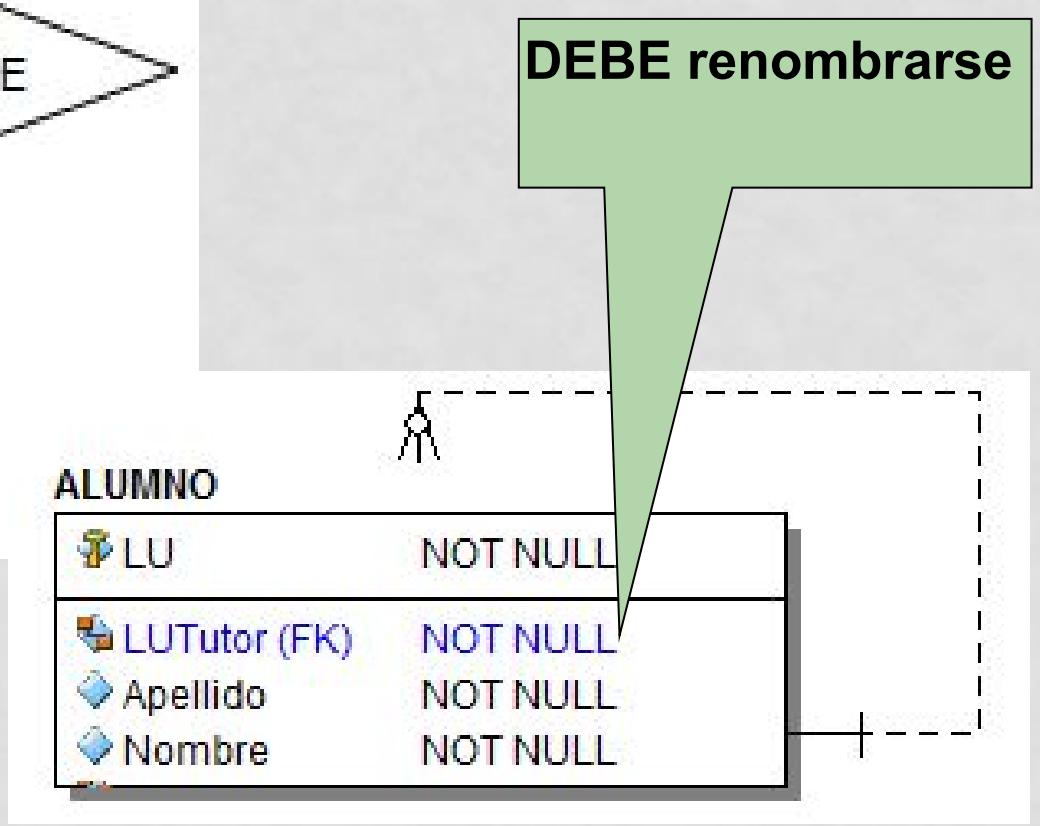
DERIVACIÓN DE RELACIONES BINARIAS 1:N

Los atributos identificadores de la entidad (clave de la relación) del 'lado 1', se agregan como atributos en la tabla correspondiente a la entidad del 'lado N' → constituyen una **clave extranjera**



DERIVACIÓN DE RELACIONES: UNARIAS 1:N (Ó N:1)

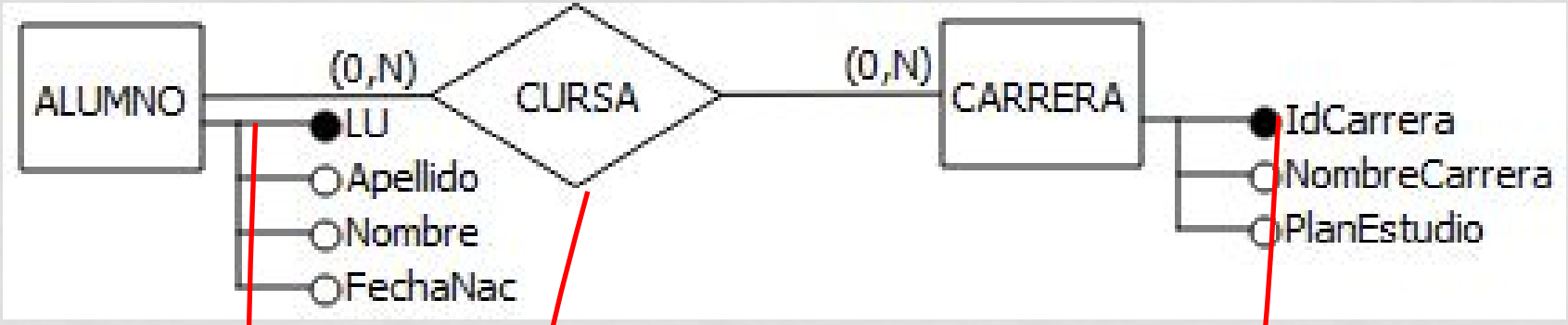


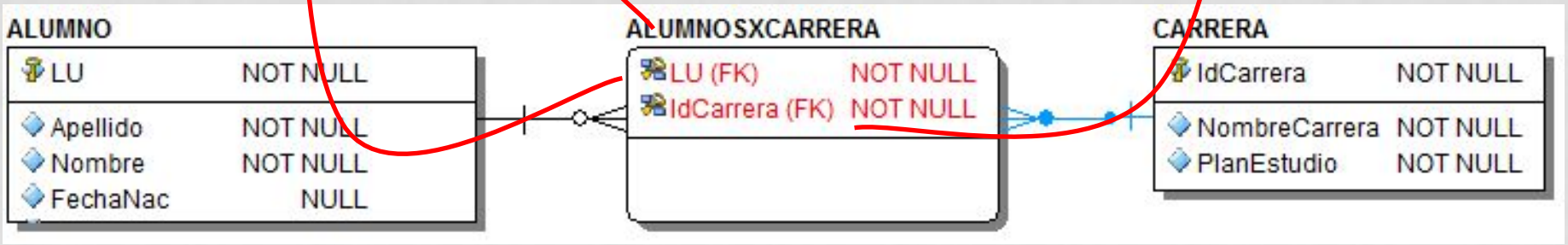


DERIVACIÓN DE RELACIONES UNARIAS Y BINARIAS N:N

- Se crea una **nueva tabla**, cuya clave es la yuxtaposición de los identificadores (claves) de cada una de las entidades participantes.
- Nombre de tabla: nombre indicado en el rombo, o puede renombrarse.
- Cada una de las claves, por separado es una clave extranjera referida a la tabla(entidad) de la cual proviene.

DERIVACIÓN DE RELACIONES BINARIAS N:N





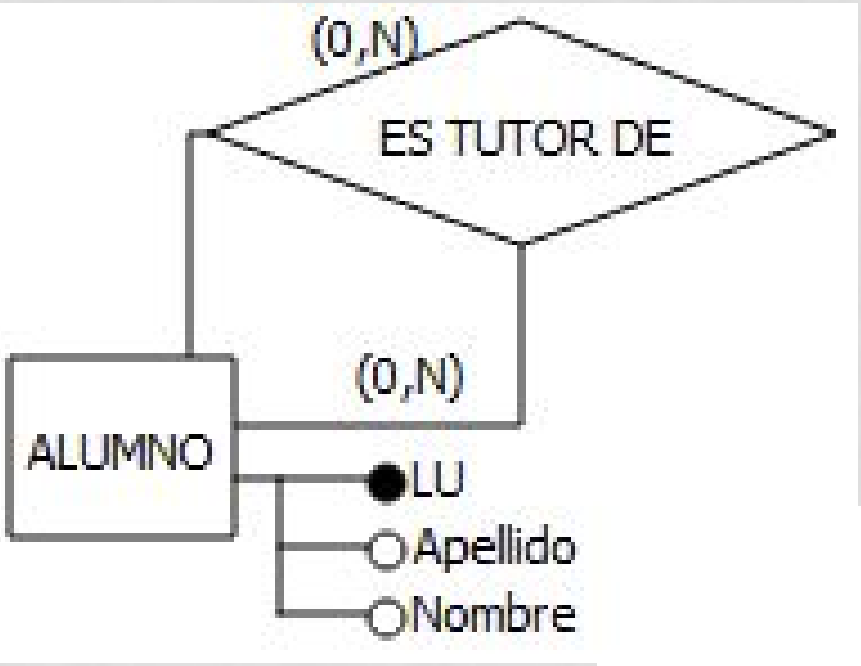
CREACIÓN DE TABLAS

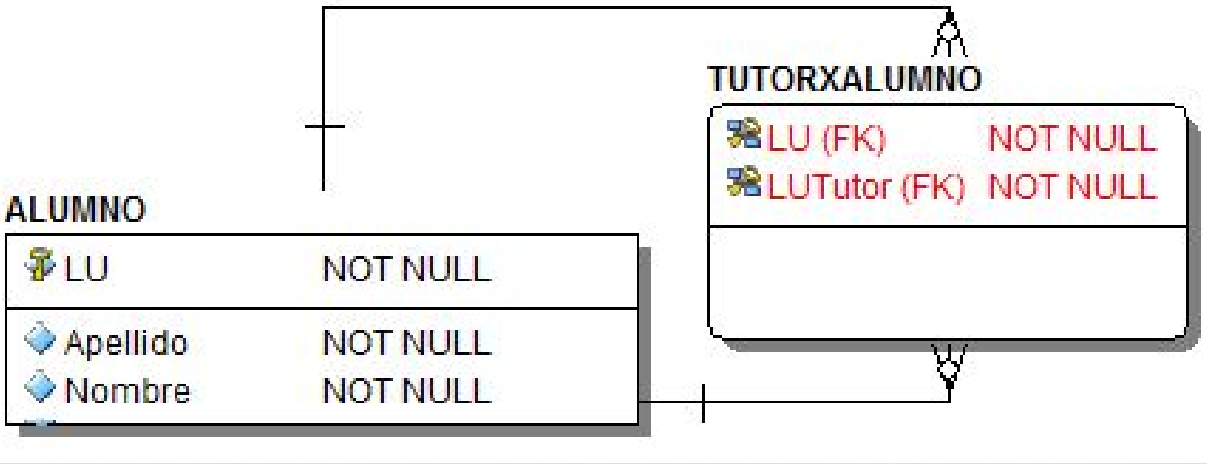

```
CREATE TABLE ALUMNO(
   LU              integer         NOT NULL,
   Apellido        varchar(30)     NOT NULL,
   Nombre          varchar(30)     NOT NULL,
   FechaNac                date,
CONSTRAINT PK_ALUMNO PRIMARY KEY (LU)
);
```

```
CREATE TABLE CARRERA(
   IdCarrera      char(5)    NOT NULL,
   NombreCarrera  varchar(100)    NOT NULL,
   PlanEstudio    char(6)    NOT NULL,
   CONSTRAINT PK_CARRERA PRIMARY KEY (IdCarrera)
);
```

```
CREATE TABLE ALUMNOSXCARRERA(
   LU        integer NOT NULL,
```

DERIVACIÓN DE RELACIONES UNARIAS N:N





DERIVACIÓN DE ATRIBUTOS EN RELACIONES

- Los atributos de una relación pueden ser del mismo tipo que los de una entidad.
- Si la relación que describen es designativa (1:N) entonces se incluyen en la tabla del lado N, derivándolos en forma análoga a los de las entidades.
- Si la relación es asociativa (binaria N:N o ternaria), se derivan en la tabla producto de la relación, también de forma análoga a los de las entidades.

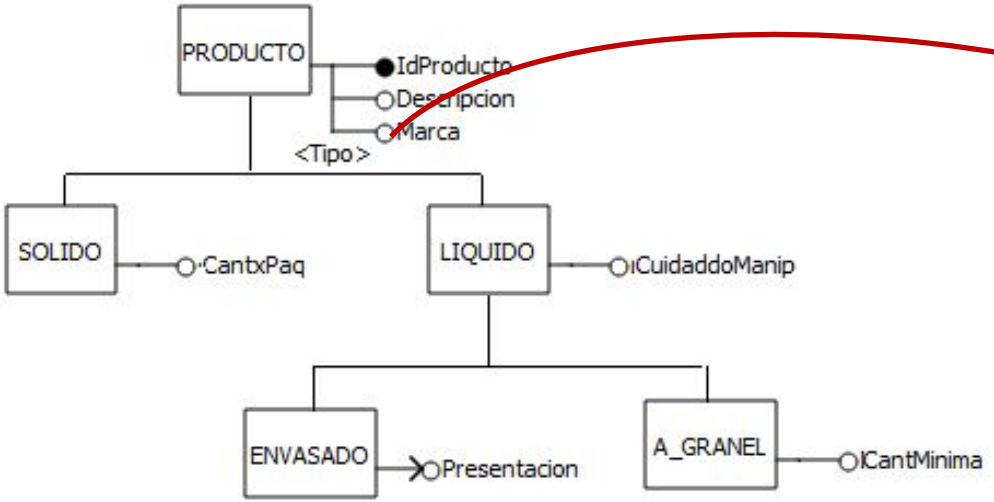
DERIVACIÓN DE JERARQUÍAS

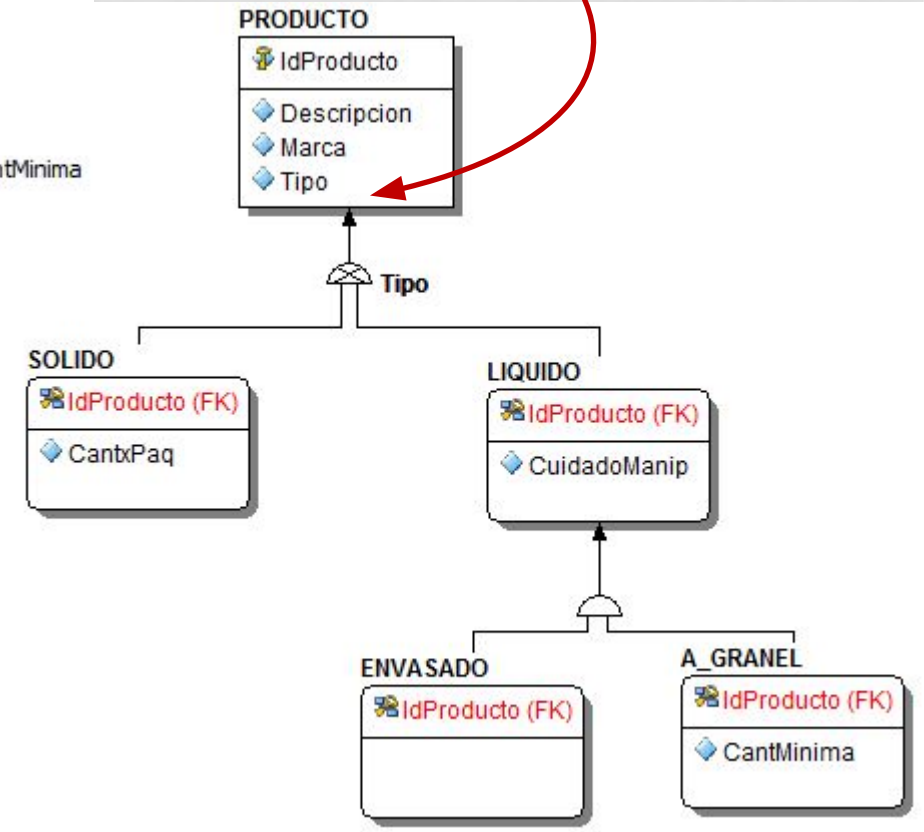
- Se crea una tabla por la **entidad supertipo** (con los atributos en común) y una tabla por cada una de las **entidades subtipo** (con los atributos propios).

- La **clave de la tabla supertipo** es la clave de cada una de las entidades **subtipo**.

- Para las jerarquías exclusivas, que deben incluir el **atributo discriminante (tipo)**, éste se debe agregar a la tabla correspondiente a la entidad supertipo

DERIVACIÓN DE JERARQUÍAS

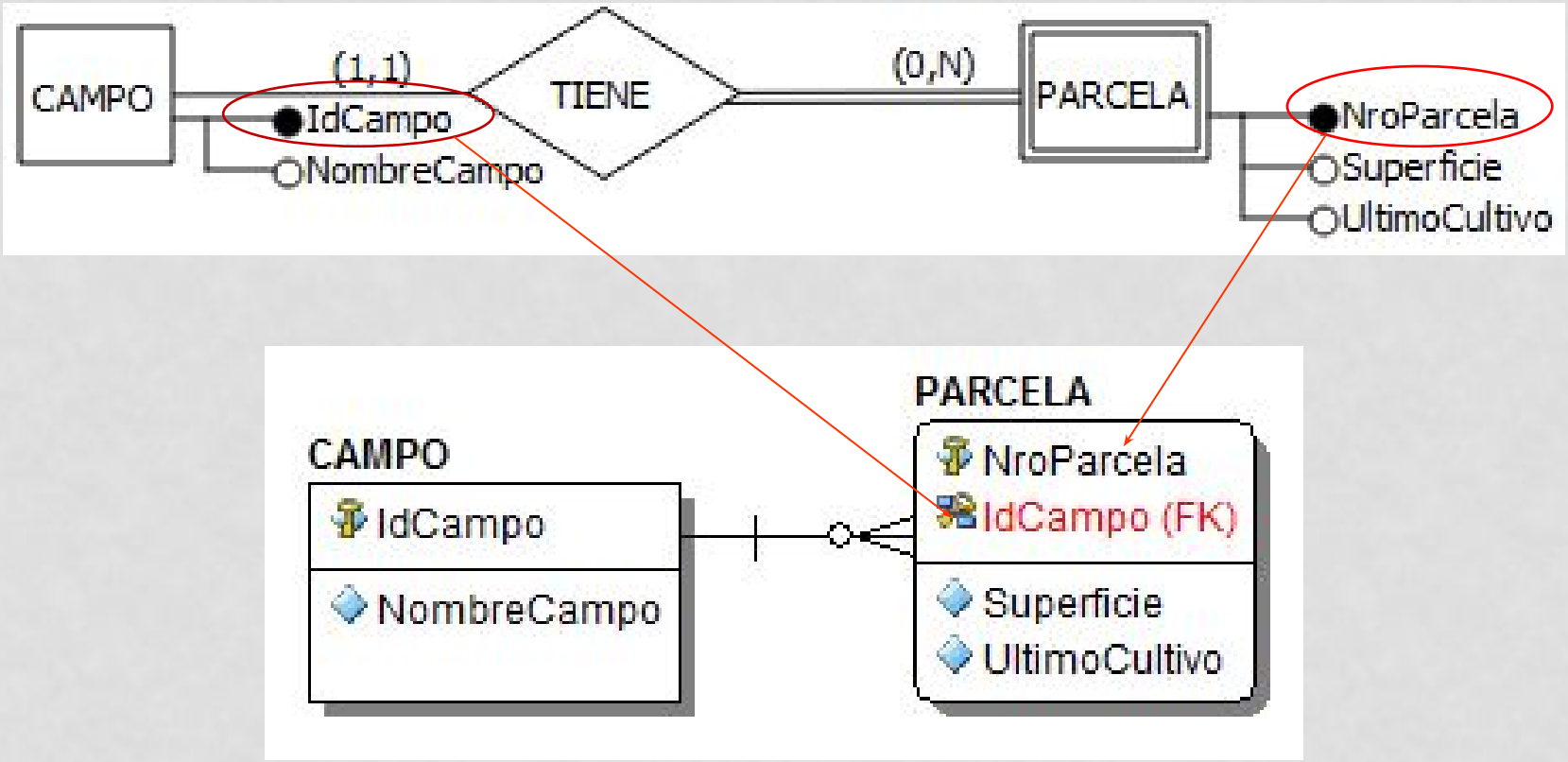




DERIVACIÓN DE ENTIDADES DÉBILES

Entidades Débiles: tienen dependencia de existencia y de identificación.

Su clave se forma con el identificador propio (clave parcial) más el identificador (clave) de la entidad fuerte,



SENTENCIA CREATE TABLE

```
CREATE [ TABLE [ IF NOT EXISTS ] nombre_tabla (
[
  { nombre_columna tipo_dato [ column_constraint [ ... ], ]
    |  table_constraint }
    [, ... ]
] );
```

donde column_constraint es:

```
[ CONSTRAINT constraint_name ]
{ NOT NULL  |
  NULL  |
  DEFAULT default_expr  |
  UNIQUE index_parameters  |
  PRIMARY KEY index_parameters  |
  REFERENCES reftable [ ( refcolumn ) ] }
```

y table_constraint es:

```
[ CONSTRAINT constraint_name ]
```