

CURSO  
SEMESTRAL



PROTECO

# MODELO RELACIONAL

DISEÑO LÓGICO BD

DISEÑO BASES DE DATOS

21/ 09/ 18

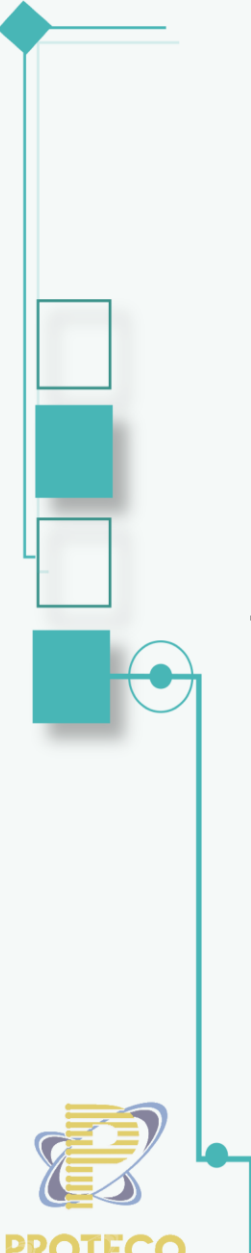
# DISEÑO LÓGICO DE LA BASE DE DATOS

- Se crea a partir del modelo conceptual.
- El esquema lógico global, es un esquema que presenta la estructura general de una base de datos.
- La descripción de la estructura de la BD esta basada en el modelo de datos lógico elegido para Implementarla.



- ▣ Actualmente, para la mayoría de las aplicaciones de gestión que utilizan bases de datos, el modelo más empleado es el modelo relacional, por su gran versatilidad, potencia y por los formalismos matemáticos sobre los que se basa.
- ▣ Este modelo fue propuesto por Edgar Frank Codd. El modelo relacional es un modelo lógico que establece una estructura sobre los datos.



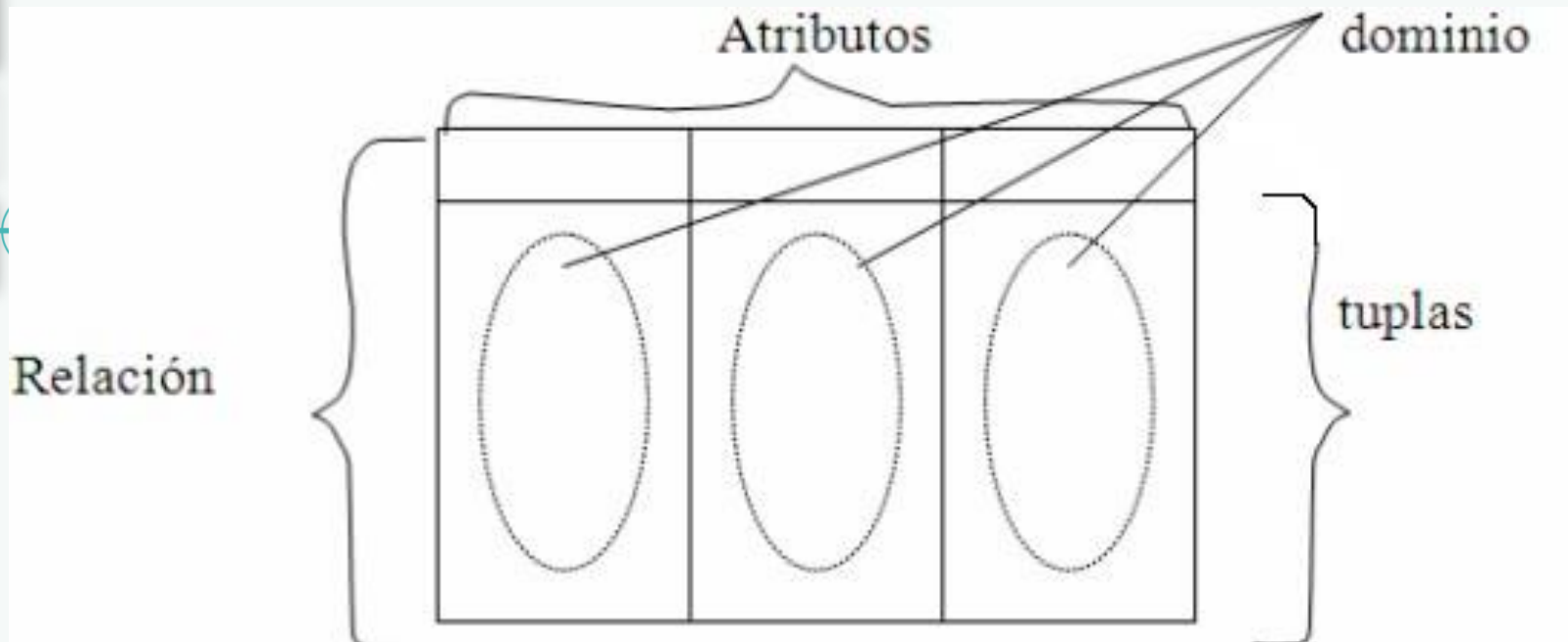








El modelo relacional permite representar la información del mundo real de una manera intuitiva, introduciendo conceptos cotidianos y fáciles de entender por cualquier persona.



# Modelo Relacional

Modelo lógico de datos, donde los datos se representan en forma de tablas





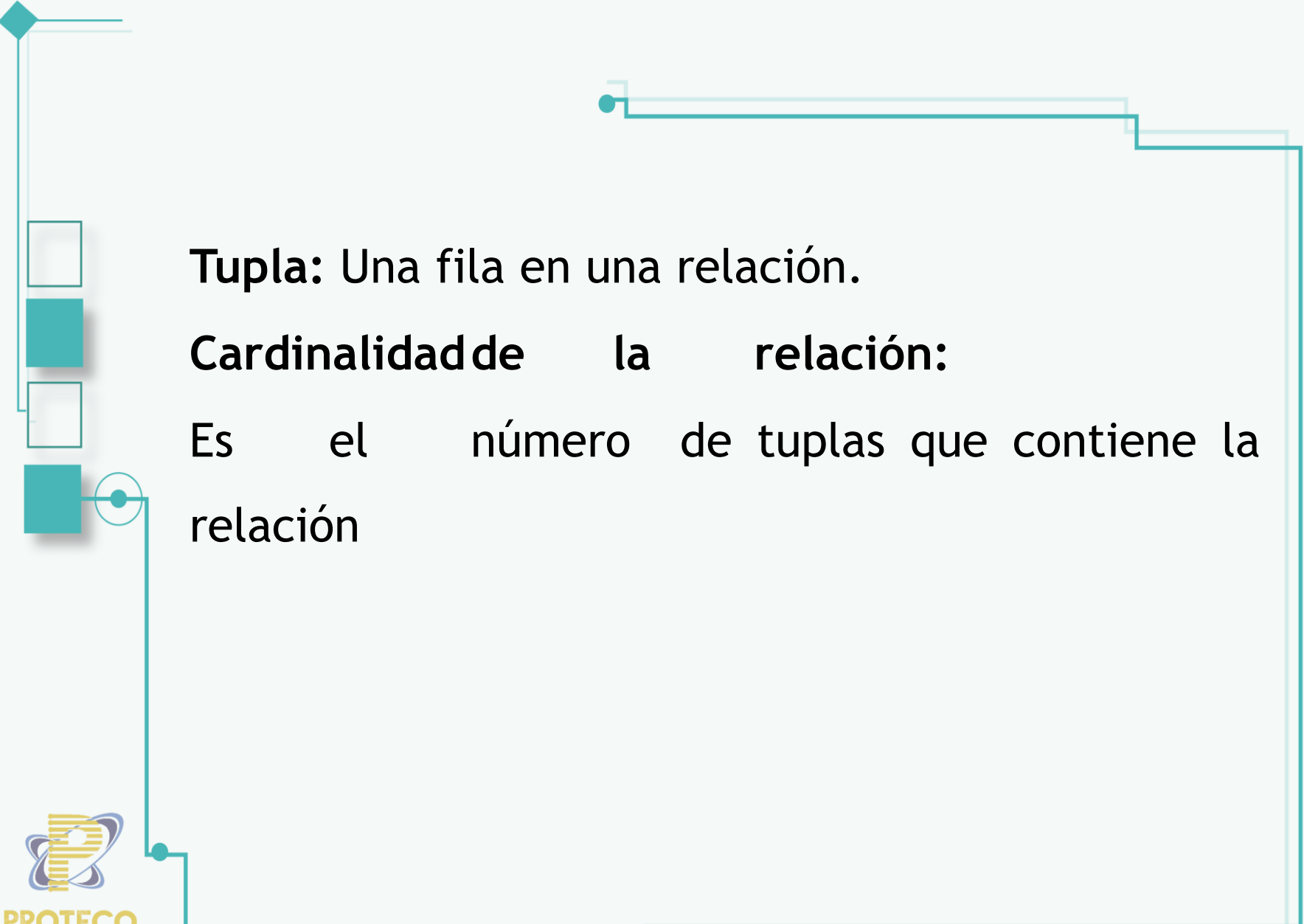
**Relación:** Es la estructura básica del modelo relacional, se representa como una tabla de dos dimensiones que contiene filas y columnas de datos.

**Atributo:** Es una columna en una relación.

**Grado de la relación:** Número de atributos en una relación.

**Dominio:** El conjunto de valores válidos que toma un atributo.





**Tupla:** Una fila en una relación.

**Cardinalidad de la relación:**

Es el número de tuplas que contiene la relación



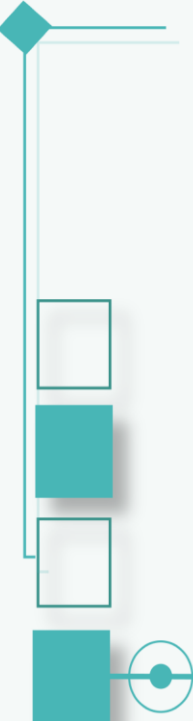
# Tipos de llaves

**Llave candidata:** Cualquier conjunto de atributos que identifican unívoca y mínimamente cada tupla de una relación

**Llave primaria:** Es la llave candidata elegida por el diseñador como llave de la relación.



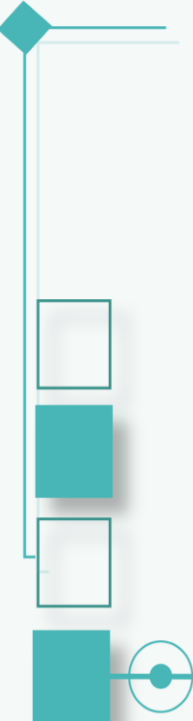




**Llave alternativa:** Es una llave candidata que no fue elegida como llave primaria pero que es importante considerarla dentro del modelo.

**Llave foránea:** Un atributo o conjunto de atributos en una relación que constituyen una llave primaria en alguna otra.





**Restricción:** Una regla que restringe los valores en una base de datos.

**Valor por defecto:** Un valor que se inserta automáticamente si el usuario no especifica la entrada.

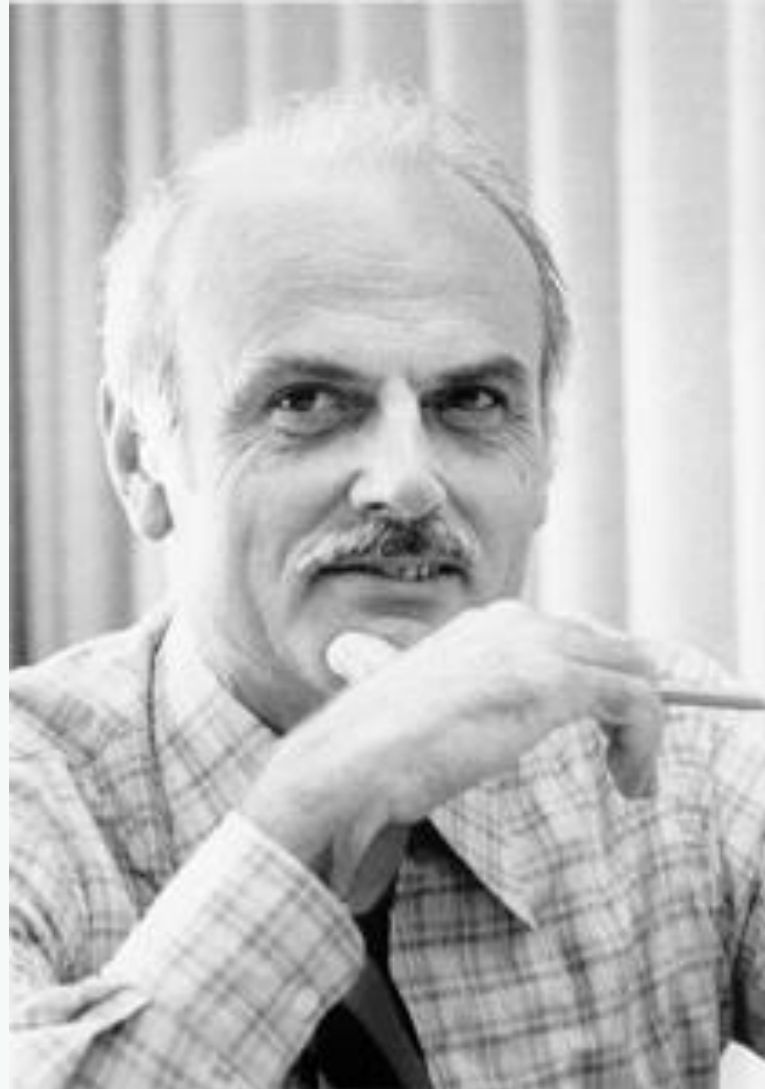
**Valor nulo:** El valor en un atributo en una tupla es inaplicable, ausencia de valor

# Reglas de Codd

Codd definió un conjunto de reglas que un SGBD debe satisfacer para que sea considerado **relacional**  
Creadas a principios de los 70's por Edgar Frank "Ted" Codd



# Reglas de Codd



Edgar Frank Codd

# Reglas de Codd

Codd definió un conjunto de reglas que un SGBD debe satisfacer para que sea considerado **relacional**

## Regla 1. Representación de la información

Toda información almacenada en una base de datos relacional debe representarse explícitamente a nivel lógico, y de manera única, por medio de valores en tablas. **Principio básico del modelo relacional.**



## Regla 2. Acceso garantizado

Todo dato debe ser accesible mediante una combinación de un nombre de tabla, un valor de su clave y el nombre de una columna. Es una forma de insistir en la obligatoriedad de la clave primaria.

## Regla 3. Tratamiento sistemático de valores nulos.

Los valores nulos, información desconocida o inaplicable, han de ser tratados sistemáticamente por el sistema, el cual ha de ofrecer las facilidades necesarias para su tratamiento.



## Regla 4. Catálogo activo en línea basado en el modelo Relacional

La representación de la meta información (descripción de la base de datos) debe ser igual a la de los otros datos y su acceso debe poder realizarse por medio del mismo lenguaje relacional que se utiliza para los demás datos; es decir, el modelo de datos para la meta información debe ser también el relacional

## Regla 5. Sublenguaje de datos completo

Debe existir un lenguaje que permita un completo manejo de la base de datos (definición de datos, definición de vistas, manipulación de datos, restricciones de integridad, autorizaciones y gestión de transacciones).

## Regla 6. Actualización de vistas

Toda vista teóricamente actualizable debe poder ser actualizada por el sistema.

## Regla 7. Inserciones, modificaciones y eliminaciones de alto nivel

Todas las operaciones de manipulación de datos (consulta, inserción, modificación y borrado) deben operar sobre conjuntos de filas (lenguaje no navegacional).





## Regla 8. Independencia física de los datos

El acceso lógico a los datos debe mantenerse incluso cuando cambien los métodos de acceso o la forma de almacenamiento.

## Regla 9. Independencia lógica de los datos

Los programas de aplicación no deben verse afectados por cambios realizados en las tablas que estén permitidos teóricamente y que preserven la información



## **Regla 10. Independencia de la integridad**

Las reglas de integridad de una base de datos deben ser definibles por medio del sublenguaje de datos relacional y habrán de almacenarse en el catálogo de la base de datos (meta base), independiente de los programas de aplicación.

## **Regla 11. Independencia de la distribución**

Debe existir un sublenguaje de datos que pueda soportar bases de datos distribuidas sin alterar los programas de aplicación cuando se distribuyan los datos por primera vez o se redistribuyan éstos posteriormente.



## Regla 12. Regla de la no subversión

Si un SGBD soporta un lenguaje de bajo nivel que permite el acceso fila a fila, éste no puede utilizarse para saltarse las reglas de integridad expresadas por medio del lenguaje de más alto nivel.

## Regla 0. Regla básica

Cualquier sistema que se anuncie como sistema gestor de bases de datos relacionales debe ser capaz de gestionar por completo las bases de datos utilizando sus capacidades relacionales.  
(DDL, DML)



# Tipos de Relaciones

## Relaciones Persistentes

- ▶ **Base:** Son las relaciones obtenidas como transformación de los diferentes elementos del modelo ER y forman parte del esquema conceptual
- ▶ **Vista:** Son relaciones virtuales creadas dependiendo del tipo de usuario que utiliza la base de datos. Son los esquemas externos
- ▶ **Instantáneas:** Son relaciones creadas con cierta regularidad pero que no pertenecen al diseño de la base de datos.



## Relaciones No persistentes

- ▶ **Resultados Intermedios:** Son los resultados que surgen de las subconsultas.
- ▶ **Temporales o auxiliares:** Son creadas al inicio del sistema y eliminada al terminar éste.



# Propiedades de las relaciones

- ▶ No pueden existir tuplas iguales
- ▶ No hay orden en las tuplas ni en los atributos
- ▶ Los valores que pueden tomar los atributos son atómicos, es decir, en el cruce de un atributo y una tupla sólo puede haber un valor.



# Reglas de integridad

## Entidad

- ▶ Ningún atributo que forme parte de la llave primaria puede tomar un valor nulo.

## Referencial

- ▶ No deben existir llaves foráneas sin concordancia o deben tener valores nulos.

BANCOS	
ENTIDAD	NOMBRE
0893	Santander
0059	Popular
3428	Bilbao Vizcaya
5632	Banesto

OFICINAS			
ENTIDAD	CODIGO_OFICINA	POBLACION	DIRECCION
0893	001	Madrid	Castellana, 73
3428	022	Las Palmas	Triana, 21
0893	022	Gáldar	R. Moreno, 3
5632	213	Oviedo	Uria, 43
0893	300	Barcelona	Diagonal, 435



# Restricciones Semánticas

## Declarativas

- ▶ Clave primaria declarar un atributo o conjunto de atributos como clave primaria que identifica unívocamente cada tupla de una relación. No toma valores nulos. Es decir, siempre será un valor NOT NULL.



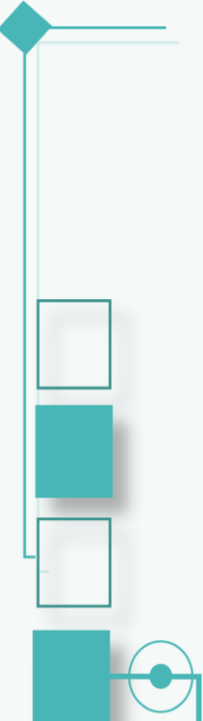


- ▶ Unicidad (UNIQUE) los valores de un atributo o conjunto de atributos no pueden repetirse. Permite la definición de claves alternativas
- ▶ Obligatoriedad (NOT NULL) permite declarar si uno o varios atributos de una relación deben tomar siempre un valor, es decir, no pueden tomar valores nulos.



- ▶ Clave Foránea (FOREIGN KEY) también denominada integridad referencial, se utiliza para enlazar relaciones en una base de datos.
- ▶ La integridad referencial indica que los valores de la clave ajena en la relación que referencia deben corresponderse con los valores existentes de la clave primaria definida.



- 
- ▶ Verificación (CHECK). Comprueba en toda operación de actualización, si el predicado es cierto o falso. La restricción se define sobre un único elemento.

## Procedimental

- ▶ Disparador (TRIGGER). Restricciones en las que el usuario puede especificar la respuesta o acción ante una determinada condición

# Reglas de Integridad Referencial (borrado y actualización)

- ▶ **Restringida(No Action):** No permite borrar o actualizar una tupla de la tabla que tiene llave primaria referenciada
- ▶ **Cascada (Cascade):** Al eliminar o actualizar la tupla principal se eliminarán o modificarán todas las tuplas que estén relacionadas con ella en otras relaciones
- ▶ **Puesta a nullos (Set Null):** Permite poner el valor de la clave ajena referenciada en nulo cuando se produce un borrado o actualización



# Esquema de una relación

Es la parte invariante de una relación y esta constituida por:

$$R=\{A:D,I\}$$

- ▶ R el nombre de la relación
- ▶ A el nombre de los atributos que la componen
- ▶ D los dominios de los que toman valores los atributos
- ▶ I restricciones de integridad, llaves que contiene



# Obtención del Modelo Relacional a partir del modelo Entidad Relación

- ▶ Para obtener el modelo Relacional a partir del modelo Entidad Relación existen una serie de reglas a seguir...



## Simbología a utilizar

UNIQUE	(U)
FOREIGN KEY	(FK)
CHECK	(CK)
Calculados	(C)
Discriminantes	(D)
No obligatorio	(N)



Ejemplo:

PROFESOR={cveProf(PK),nomProf,apPat,apMat(N),rfcProf(U),edad(C)\*CS1}

EMAILPROF={cveProf(PK,FK),email(PK)}

\*CS1 Se obtiene del rfc del profesor

## Entidades Débiles

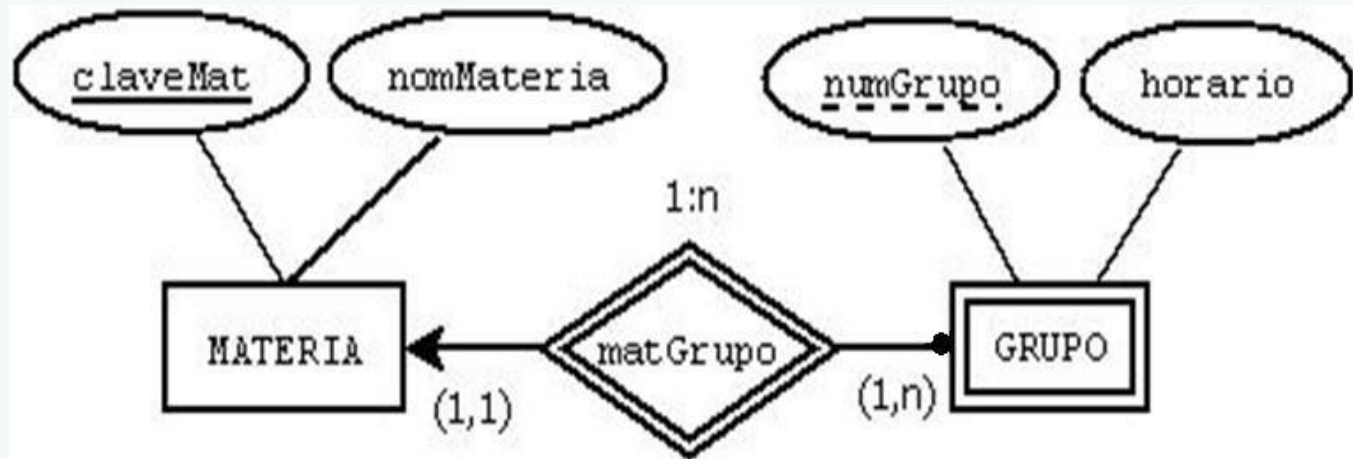
Se crea la relación conservando todos sus atributos y se propaga la llave principal de la entidad fuerte de la que depende.

La llave principal se formará con la llave primaria de la entidad fuerte y el discriminante de la entidad débil





# Ejemplo



MATERIA={claveMat(PK), nomMateria}



GRUPO={[claveMat(FK), numGrupo(D)](PK), horario}



# Relaciones

- **1:1** La llave primaria de una entidad se propaga a la otra entidad como llave foránea con la restricción de unicidad, dependiendo de la cardinalidad mínima, si la cardinalidad mínima es cero, se pasa el atributo de la entidad de cardinalidad uno hacia la de cardinalidad cero, si en ambas es la misma cardinalidad, dependerá del contexto.

**1:1**

# Relaciones

- **m:1 ó 1:m** La llave primaria de la entidad que tiene la cardinalidad uno se propaga a la relación de muchos como llave foránea.

**m:1 / 1:m**

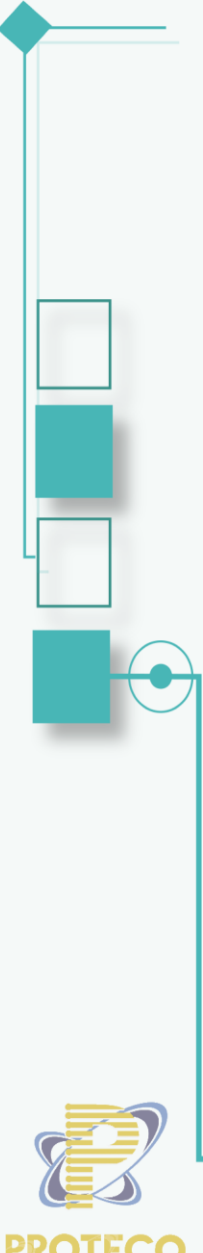


# Relaciones

- M:M Se genera una nueva relación que contendrá la llave primaria de cada una de las entidades que une, pero como llaves foráneas y juntas como llave primaria, además de sus atributos propios (descriptivos) de la relación.

**m:m**





Las relaciones (1:1, 1:m, m:1) que contengan atributos descriptivos, se tienen dos opciones.

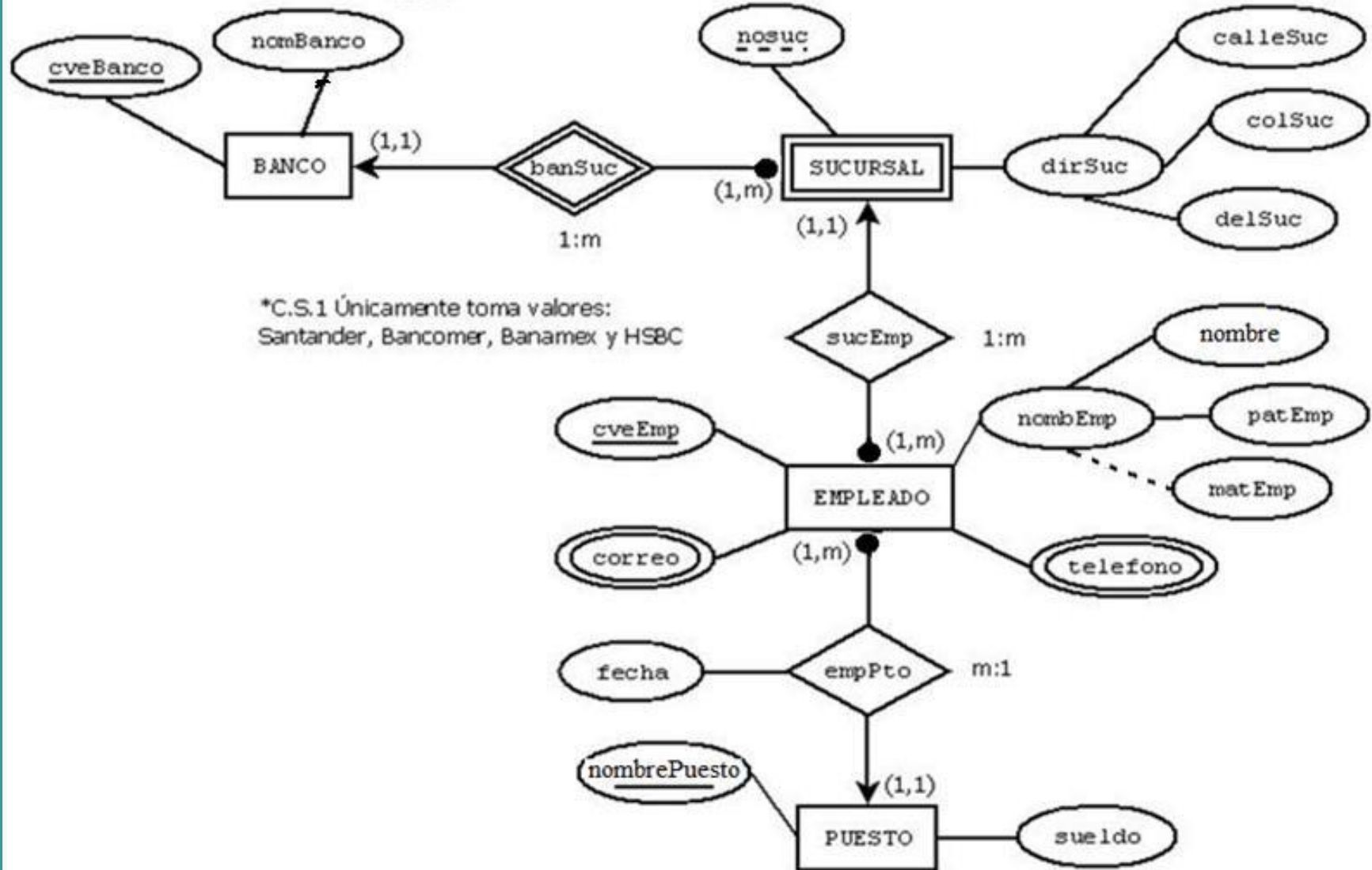
- Generar una nueva relación que contendrá la llave primaria de cada una de las entidades que une como llaves foráneas, además de sus atributos descriptivos, conservando la unicidad en la nueva relación.
- Propagar la llave primaria de la entidad uno junto con todos los atributos de la relación a la entidad m.



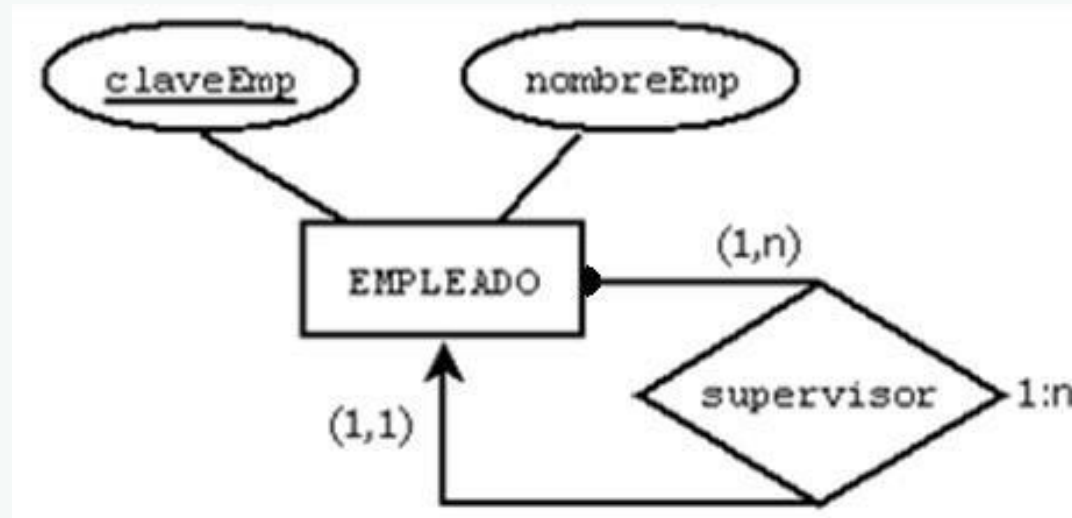
# EJEMPLOS

## SUCURSALES BANCARIAS

\*C.S.1



# Transformación de relaciones recursivas



**EMPLEADO={claveEmp(PK), nombreEmp, claveSup(FK)}**



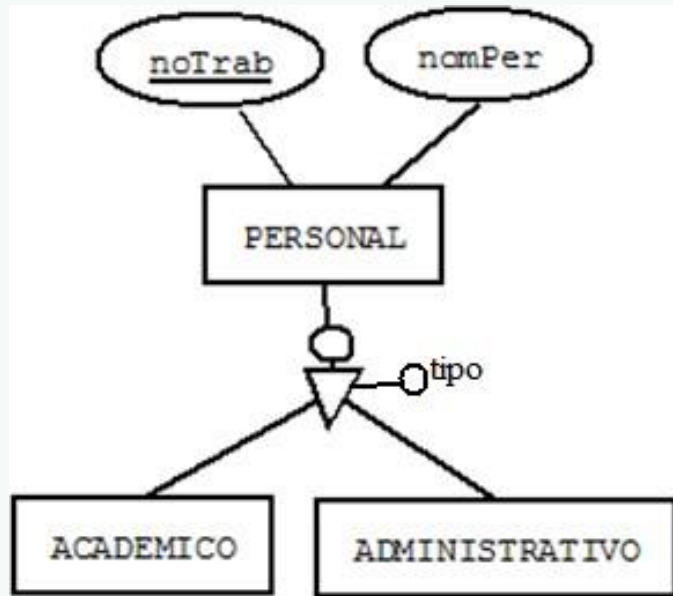


# Transformación de Jerarquía de tipos

Se tienen varias opciones según el contexto...



# Generalización

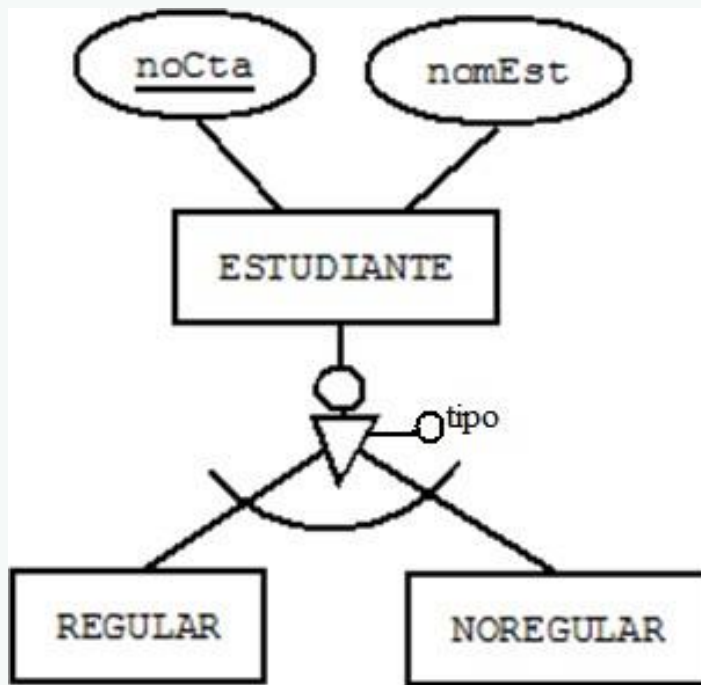


- ▶ No excluyente y sin atributos en los subtipos

PERSONAL={noTrab(PK), nomPer}

TIPOPERSONAL={noTrab(FK), tipoPersonal(CK)}

Ck verifica los valores de ACADÉMICO o ADMINISTRATIVO



- ▶ Excluyente y sin atributos en los subtipos

ESTUDIANTE={noCta(PK), nomEst, tipoStud(CK)}

ck verifica los valores de REGULAR o NOREGULAR

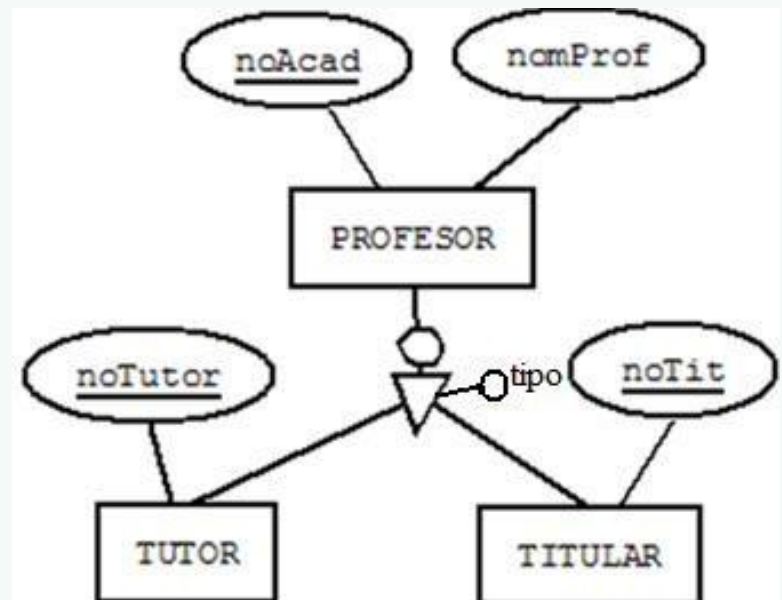
- ▶ No excluyente, con atributos y llave primaria en los subtipos

PROFESOR={noAcad(PK), nomProf,...}

TIPOPROFESOR={noAcad(FK), tipoProfesor(CK)}

TUTOR={noAcad(Pk,FK ),noTutor(U),...}

TITULAR={noAcad(Pk,Fk),noTit(U),...}



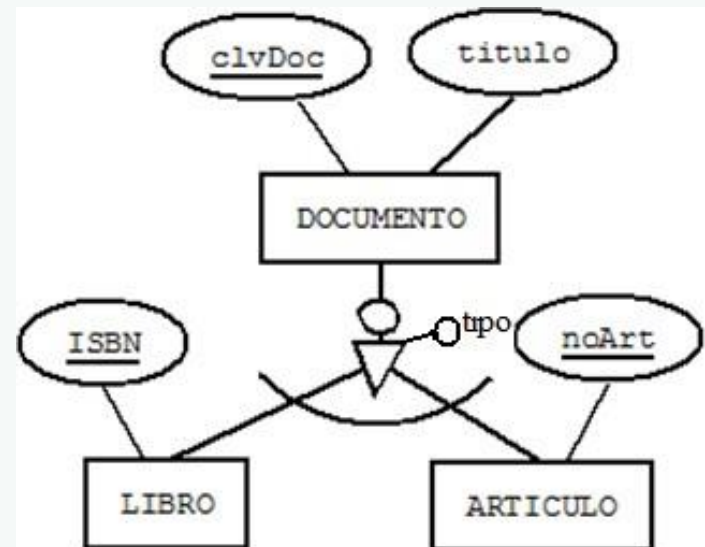
► Excluyente, con atributos y llave primaria en los subtipos

DOCUMENTO={clvDoc(pk), titulo,...,tipoDocumento(ck)}

LIBRO={clvDoc(Pk,FK),ISBN(U),...}

ARTICULO= {clvDoc(PK,Fk)),noArt(U,...)}

Asegúrese que las FK no se encuentren en ambos subtipos.



# Especialización

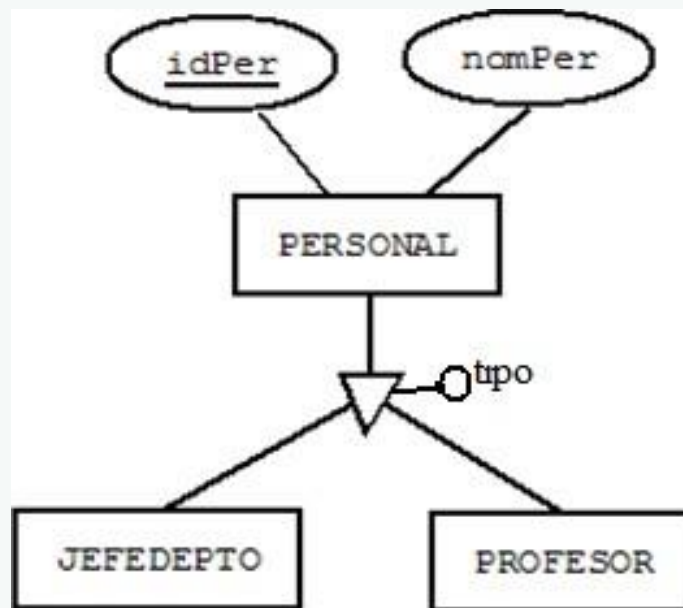
- ▶ No excluyente y sin atributos en los subtipos

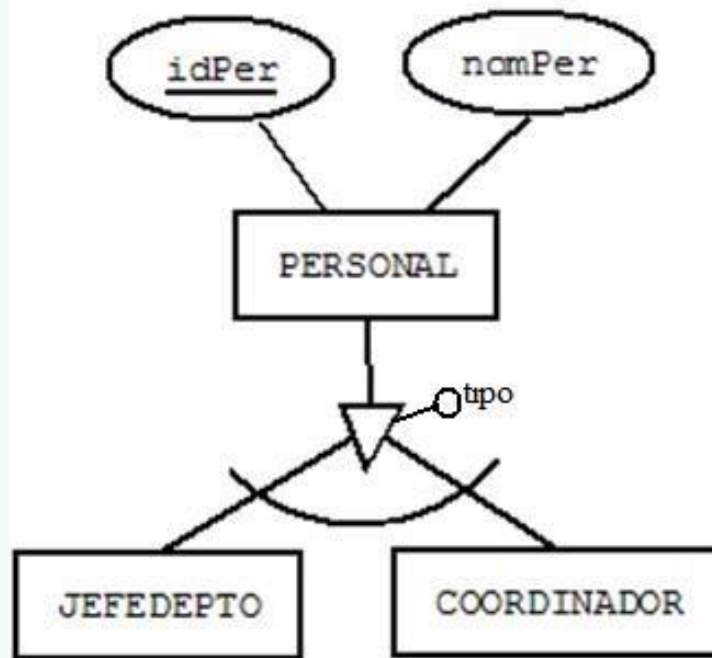
PERSONAL={idPer(PK),nomPer,...}

TIPO={idTipo(PK), descripTipo}

TIPOPERSONAL={idPer(FK), idTipo(FK)}

Habr  PK del supertipo que no tengan asignado ning n tipo.





- ▶ Excluyente y sin atributos en los subtipos

PERSONAL={idPer(PK), nomPer, idTipo(FK)}

TIPOPERSONAL={idTipo(PK), descripTipo}

La FK en el supertipo podrá tener valores nulos

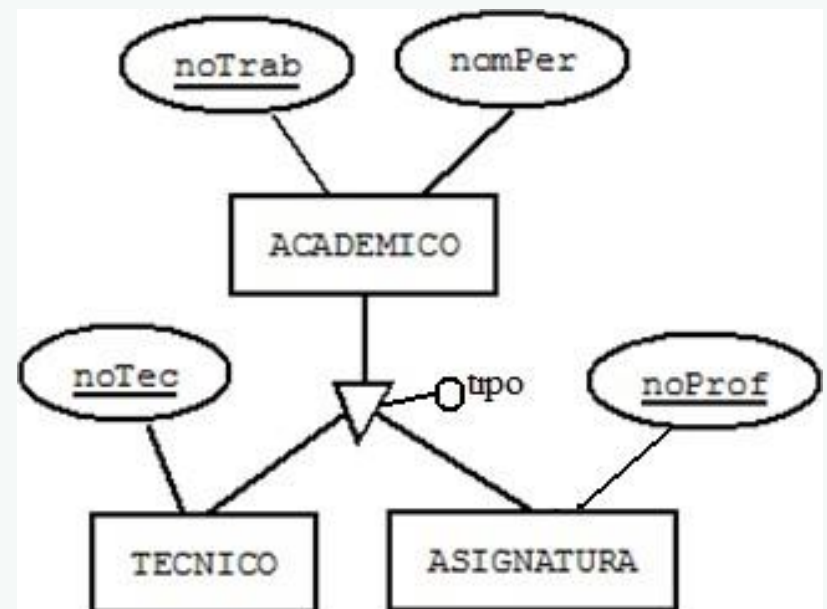
- ▶ No excluyente, con atributos y llave primaria en los subtipos

ACADEMICO={noTrab(PK),nomPer,...}

TIPOACADEMICO={(noTrab(FK),tipoAcademico)(pk)}

TECNICO={noTrab(PK, FK),noTec(U),...}

ASIGNATURA={noTrab(Pk,FK),noProf(U),...}





► Excluyente , con atributos y llave primaria en los subtipos

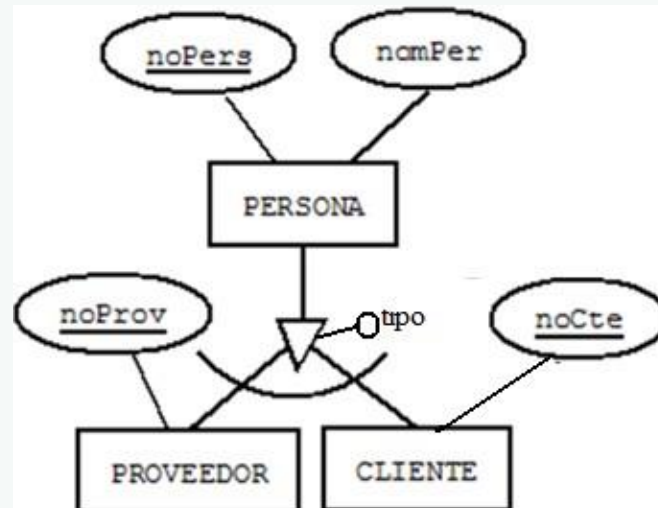
PERSONA={noPer(PK),nomPer,...}

TIPOPERSONA={noPer(PK,FK),tipoPersona}

PROVEEDOR={noPer(Pk,FK),noProv(U),...}

CLIENTE={noPer(Pk,FK),noCte(U),...}

Asegúrese que las FK no se encuentren en varios subtipos a la vez.



# ADMINISTRACIÓN DE UNA ESCUELA

