

2. Introducción

Ya hemos visto que técnica (Diagramas Entidad/Relación) hay que utilizar para realizar el Modelo conceptual de datos. Ese Modelo conceptual se realiza en la fase de **Análisis** (dónde la pregunta sería "**Qué**" tenemos que hacer). En la siguiente fase, el **Diseño**, tendremos que realizar el modelo lógico de datos. En esta fase nos centraremos en "**Cómo**" tenemos que hacerlo.

FASES \ MODELOS	Modelo de datos	Modelo de procesos	Modelo de eventos
Análisis	Modelo <u>conceptual</u> de datos	Análisis de procesos	Análisis de eventos
Diseño	Modelo <u>lógico</u> de datos	Diseño de procesos	Diseño de <u>interfaces de usuario</u>
Implementación	Modelo <u>físico</u> de datos	Codificación de procesos	Codificación de las interfaces de usuario

3. Modelo Lógico de Datos. Objetivos y Modelos

- **Base de Datos (BBDD):** Colección de datos interrelacionados almacenados sin redundancias perjudiciales o innecesarias.
- **Sistema de Gestión de Bases de Datos (SGBD):** Software para describir, recuperar y manipular los datos almacenados en una BBDD.

El **modelo lógico de datos** está orientado a describir los datos teniendo en cuenta "Cómo" se tienen que implementar los datos en un SGBD. Por tanto el modelo lógico depende del tipo de SGBD elegido.

Los **SGBD** se apoyan en modelos teóricos. Los principales modelos teóricos en los que se basan los SGBD son:

- Modelo Jerárquico (En desuso)
- Modelo En red (En desuso)
- **Modelo Relacional**
- Modelo Orientado a Objetos

El **objetivo** principal del modelo relacional **es proteger al usuario de la obligación de conocer las estructuras de datos físicas con las que se representa la información de una base de datos**. Desvincular estas estructuras de datos, que son complejas por naturaleza, del diseño lógico (Modelo Relacional), permite que la BD se pueda implementar en cualquier gestor de base de datos relacional (Oracle, MySQL, DB2...).

El modelo que siguen la gran mayoría de los SGBD comerciales es el Modelo Relacional, por tanto, este será el modelo que vamos a estudiar.

Los SGBD que se apoyan en el modelo relacional se denominan Sistemas de Gestión de Bases de Datos Relacionales (SGDBR) y las Bases de Datos que se crean con estos sistemas se denominarán Bases de Datos Relacionales.

Algunos de los SGBDR que existen en el mercado son: Access, SQL Server, Oracle ó MySQL Server.

En resumidas cuentas, como el modelo lógico de datos depende del SGBD elegido, y los SGBDR son los más utilizados, nosotros estudiaremos el **Modelo Lógico Relacional de Datos**.

4. Modelo lógico de datos. Modelo Relacional

El **Modelo Relacional**: Modelo de datos usado en la gestión de bases de datos basado en la *lógica de predicados*¹ y la *teoría de conjuntos*². Este modelo considera la base de datos como una colección de relaciones.

Componentes del Modelo Relacional:

- Una **estructura lógica de datos** representada por relaciones.
- Un **conjunto de reglas de integridad** para hacer cumplir que los datos sean y sigan siendo consistentes a lo largo del tiempo.
- Un **conjunto de operaciones** que define la forma en que los datos se manipulan.

Base de Datos Relacional: Conjunto de tablas con registros y atributos, que tienen un campo común, el cual es igual en ambas tablas y recibe el nombre llave principal, ID o clave primaria.

Este modelo supone una serie de ventajas:

- ✓ **Independencia física**: el modo en que se almacenan los datos no influye en la definición de los mismos, que se produce en esta fase.
- ✓ **Independencia lógica**: Las operaciones sobre los datos no afectan a cómo los datos son tratados.
- ✓ **Flexibilidad**: es flexible porque puede ofrecer diversos esquemas externos.
- ✓ **Uniformidad**: en cuanto a la representación de los datos es una técnica uniforme, facilitando de este modo el empleo de los mismos.
- ✓ **Sencillez**: Es fácil de comprender y utilizar.

¹ Basado en relaciones entre objetos, como también cualidades y atributos de tales objetos. Los objetos pueden ser personas, objetos físicos o conceptos. Relaciones o atributos se denominan predicado.

² Es la rama de las matemáticas que estudia los conjuntos, que son colecciones de objetos.

4.1. Conceptos del Modelo Relacional

4.1.1. Tablas

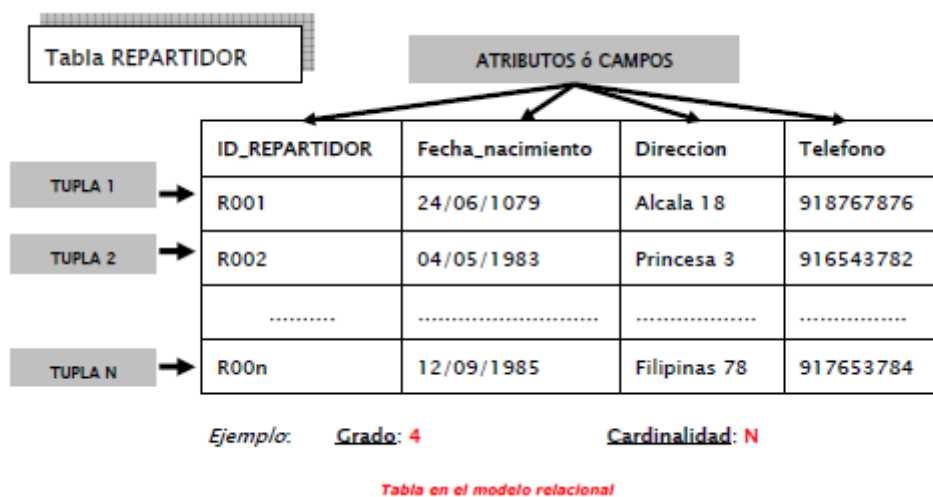
Una **tabla** es percibida como una estructura en dos dimensiones compuesta de tuplas o registros y de columnas o campos. Una tabla también se conoce como una **relación** porque el creador del modelo relacional, Codd, utilizó el término **relación** como sinónimo de tabla. Podemos considerar una **tabla** como una representación persistente de una relación lógica, es decir, una relación cuyo contenido se puede guardar de manera permanente para un uso futuro.

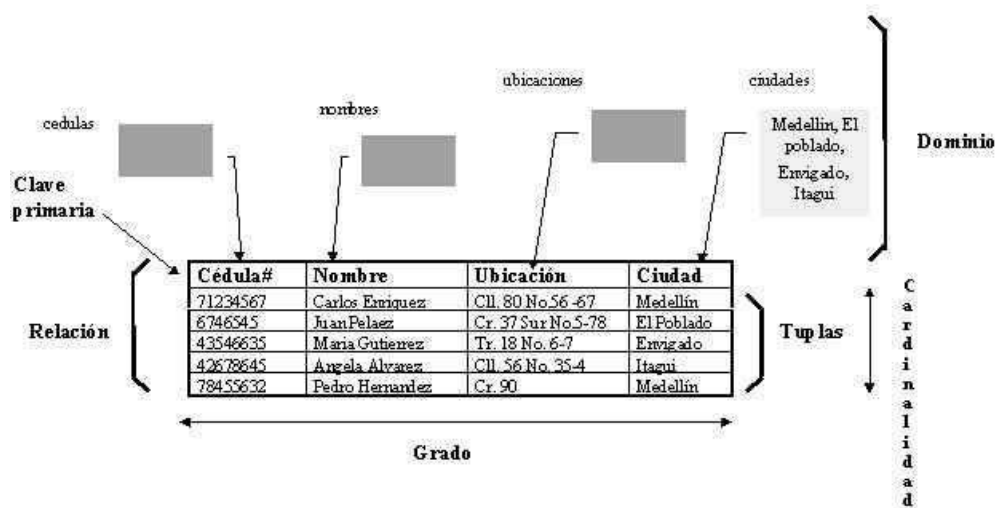
Estructura de una tabla o relación:

- ✓ **Campos o atributos** (columnas).
- ✓ **Tuplas o registros** (filas).
- ✓ **Dominio** será el conjunto de valores que puede tomar un atributo.
- ✓ **Grado** el número de atributos de una tabla.
- ✓ **Cardinalidad**, es el número de tuplas de la misma.

NOTA: No confundir la cardinalidad y el grado del modelo relacional con la cardinalidad y el grado en los diagramas E/R. Simplemente tienen el mismo nombre pero en el Diagrama E/R significan cosas distintas.

Ejemplo: en el caso de la pizzería, las tablas del modelo relacional se corresponden con las entidades del modelo entidad-relación, en este caso una tabla podría ser *repartidor*.





Características de una tabla relacional

- 1 Una *tabla* es percibida como una estructura *bidimensional* compuesta por *filas* y *columnas*.
- 2 Cada *fila* representa una *ocurrencia única* de entidad dentro del conjunto de entidades.
- 3 Cada *columna* representa un *atributo* y cada columna tiene un *nombre distinto*.
- 4 Cada *intersección* de *fila/columna* representa un *valor único de datos*.
- 5 Todos los *valores de una columna* deben tener el *mismo formato*.
- 6 Cada *columna* tiene un *intervalo específico* de valores conocido como dominio de atributos.
- 7 El *orden de las columnas y filas* no tiene importancia en el SGBD.
- 8 Cada tabla debe tener un *atributo o combinación de atributos que identifique de manera unívoca cada fila*.

4.1.2. Claves

En el modelo relacional, las claves son importantes porque:

- ✓ Se usan para asegurar que cada fila de la tabla es identificable de manera única.
- ✓ Se usan para establecer relaciones entre tablas y asegurar la integridad entre los datos.

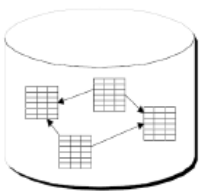
Claves de bases de datos relacional

Superclave	Un atributo o combinación de atributos de una que de modo único identifica cada fila de la tabla.
Clave Candidata	Una <i>Superclave mínima</i> . Una superclave que no contiene atributos innecesarios.
Clave Primaria (Primary Key)	Una clave candidata seleccionada para identificar de modo único todos los valores de atributo en cualquier fila. No puede contener entradas nulas.
Clave Alternativa (Alternative Key)	Claves candidatas que no han sido seleccionadas como clave primaria.
Clave Secundaria (Secondary Key)	Un atributo o combinación de atributos de una tabla que se usa estrictamente para fines de recuperación de datos.
Clave Foránea, Ajena o Extranjera (Foreign Key)	Un atributo o combinación de atributos de una tabla cuyos valores deben corresponderse con la clave primaria en otra tabla o ser nulos.

Comparativa entre Modelo E-R y Modelo Relacional

Modelo Entidad Relación	Modelo Relacional
Ocurrencia	Fila ó Tupla
Atributo	Columna
Entidad	Tabla ó Relación
Grado	Número de columnas
Cardinalidad	Número de Filas

4.1.3. Interrelaciones entre tablas

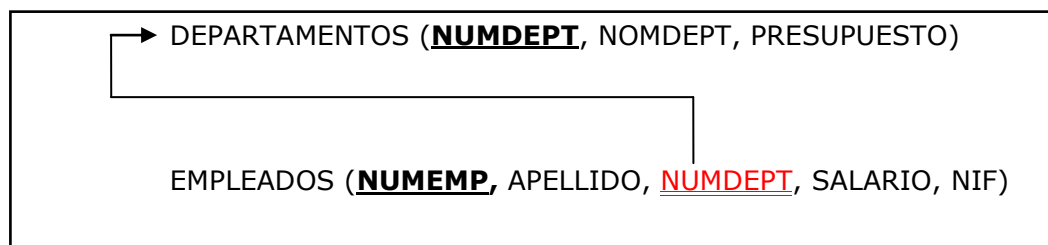


Una Base de Datos (relacional o de otro tipo) es una colección de datos (en nuestro caso relaciones o tablas) *interrelacionados*. Por tanto necesitamos asociar una relación (tabla) con otras mediante las *claves ajenas*.

Clave ajena (Foreign Key): Se denomina clave ajena de una relación **R2** a uno o varios atributos cuyos valores han de coincidir con los valores de la clave primaria de una relación **R1**.

- ✓ R1 y R2 pueden ser la misma relación (o Tabla).
- ✓ La clave ajena y la clave primaria deben estar definidas sobre los mismos dominios.

Clave primaria			Clave primaria		Clave ajena	Clave alternativa	
NUMDEPT	NOMDEPT	PRESUPUESTO	NUMEMP	APELLIDO	NUMDEP	SALARIO	NIF
D1	Marketing	1000	E1	López	D1	40	52.123.456-X
D2	Desarrollo	1200	E2	Fernández	D2	42	40.123.456-Y
D3	Investigación	500	E3	Martínez	D3	50	30.123.456-Z
			E4	Sánchez	D2	34	20.123456-A



Según este ejemplo. ¿Cuál sería el nombre del departamento al que pertenece el empleado Martínez?

INVESTIGACION

4.1.4. Reglas o Restricciones de Integridad

Las *reglas de integridad* de una base de datos relacional son muy importantes para un buen diseño de base de datos. Muchos de los SGBD hacen cumplir automáticamente las reglas de integridad. No obstante, es mucho más seguro verificar que el diseño cumple con dichas reglas.

NOTA: El concepto NULO (NULL) no es ningún valor. No significa un CERO o un ESPACIO.

Restricciones Inherentes al Modelo ó Integridad de Entidad (impuestas por el propio modelo relacional)

Todas las entradas de clave primaria son únicas y ninguna parte de una clave primaria puede ser nula.

Cada fila tendrá una identidad única y los valores de clave foránea pueden referirse debidamente a valores de clave primaria.

Una clave foránea puede tener una entrada nula, mientras que no forme parte de la clave primaria.

NO se admiten atributos multivaluados.

El orden de los campos (atributos) no es significativo.

El orden de las tuplas (filas) no es significativo.

Restricciones de Usuario (permiten recoger con la mayor fidelidad posible el mundo real)

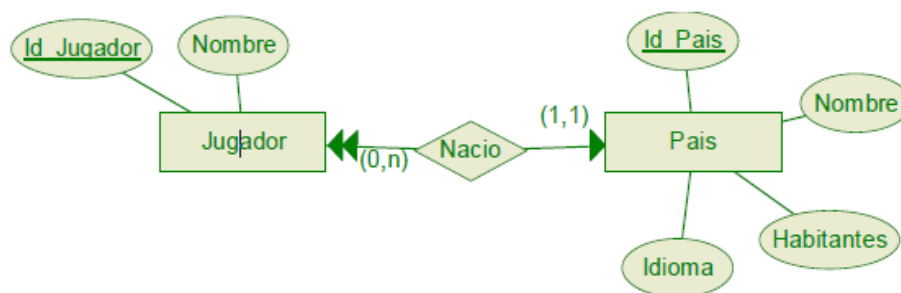
Unicidad (UNIQUE)

Obliga a que los valores de un conjunto de atributos (uno o más) no puedan repetirse en una relación. Esta restricción permite la definición de claves alternativas.

Obligatoriedad (NOT NULL):

No se admiten nulos, es decir, el valor de un atributo será siempre un valor que esté dentro del dominio al que pertenece el atributo.

EJEMPLO



Las tablas (relaciones) que se corresponden con este diagrama entidad relación en el modelo relacional serán las siguientes:

Tabla (Relación): PAISES				Tabla (Relación): JUGADORES		
<u>IdPais</u>	Nombre	Habitantes	Idioma	<u>IdJugador</u>	Nombre	<u>IdPais</u>
ESP	ESPAÑA	9.000.000	ESPAÑOL	J001	Raul Lopez	ESP
ING	INGLATERRA	12.000.000	INGLES	J002	Robinho	BRA
FRA	FRANCIA	11.345.655	FRANCÉS	J003	Zinedine Zidane	FRA
BRA	BRASIL	23.098.476	PORTUGUÉS	J004	Beckam	ING
HOL	HOLANDA	5.000.765	HOLANDES			

En esta tabla, **IdPais**, actúa como clave ajena. Relacionando a la tabla con JUGADOR.

Ejemplo1: Sí interpretamos la relación de que un jugador pueda poseer sólo una nacionalidad, tendríamos:

PAISES (idPais, nombre, habitantes, idioma)

JUGADORES (idJugador, nombre, idPais)

Ejemplo2: Sí interpretamos la relación de que un jugador pueda poseer varias nacionalidades tendríamos un atributo multivaluado 'idPais'.

PAISES (idPais, nombre, habitantes, idioma)

JUGADORES (idJugador, nombre)

POSEE_NACIONALIDAD (idPais, idJugador)

La **existencia de integridad referencial** tiene consecuencias sobre las **operaciones de modificación** de la clave y **borrado** de las tuplas, por lo que se debe de escoger entre una de las siguientes acciones que controlan la consistencia de los datos cuando se produce una de estas operaciones de borrado o modificación:

Notación respecto al Borrado y Modificación en el cuadro siguiente:

B/M: Borrado/Modificación (Delete/Cascade).

R1: Relación o tabla a la que se referencia.

R2: Relación o tabla que contiene una clave ajena que referencia a R1.

BR, MR: Borrado Restringido, Modificación Restringida (No Action)

BC, MC: Borrado en Cascada, Modificación en Cascada (Cascade)

BN, MN: borrado a Nulo (Set Null)

BD, MD: Borrado por defecto (Set Default)

Tabla (Relación): PAISES				Tabla (Relación): JUGADORES		
<u>IdPais</u>	Nombre	Habitantes	Idioma	<u>IdJugador</u>	Nombre	<u>IdPais</u>
ESP	ESPAÑA	9.000.000	ESPAÑOL	J001	Raul Lopez	ESP
ING	INGLATERRA	12.000.000	INGLES	J002	Robinho	BRA
FRA	FRANCIA	11.345.655	FRANCÉS	J003	Zinedine Zidane	FRA
BRA	BRASIL	23.098.476	PORTUGUÉS	J004	Beckam	ING
HOL	HOLANDA	5.000.765	HOLANDES			

- **NO ACTION (operación restringida):** solo se permite borrar (BR) tuplas o modificar (MR) la clave primaria si no existen claves ajenas que la referencien.
- **CASCADE (transmisión en cascada):** el borrado de una tupla o la modificación de la clave primaria en R1 lleva consigo el BC/MC en cascada de las tuplas de R2. Por ejemplo, sí borramos la fila de INGLATERRA en la tabla de PAISES, tendríamos que eliminar todas las filas de la tabla JUGADORES, que contengan como idPais ING, las tablas del ejemplo quedarían del siguiente modo:

Tabla (Relación): PAISES				Tabla (Relación): JUGADORES		
<u>IdPais</u>	Nombre	Habitantes	Idioma	<u>IdJugador</u>	Nombre	<u>IdPais</u>
ESP	ESPAÑA	9.000.000	ESPAÑOL	J001	Raul Lopez	ESP
BORRADA				J002	Robinho	BRA
FRA	FRANCIA	11.345.655	FRANCÉS	J003	Zinedine Zidane	FRA
BRA	BRASIL	23.098.476	PORTUGUÉS			
HOL	HOLANDA	5.000.765	HOLANDES			

- **SET NULL (puesta a nulos):** el borrado o modificación de tuplas de R1 pone a nulo los valores de las tuplas de R2. En el ejemplo, si borrásemos el país con idPais ING, las tablas quedarían del siguiente modo:

Tabla (Relación): PAISES				Tabla (Relación): JUGADORES		
<u>IdPais</u>	Nombre	Habitantes	Idioma	<u>IdJugador</u>	Nombre	<u>IdPais</u>
ESP	ESPAÑA	9.000.000	ESPAÑOL	J001	Raul Lopez	ESP
BORRADA				J002	Robinho	BRA
FRA	FRANCIA	11.345.655	FRANCÉS	J003	Zinedine Zidane	FRA
BRA	BRASIL	23.098.476	PORTUGUÉS	J004	Beckam	NULL
HOL	HOLANDA	5.000.765	HOLANDES			

- **SET DEFAULT (puesta a valor por defecto):** el borrado o modificación de tuplas de R1 pone a un valor por defecto definido los valores de la clave ajena R2. En las tablas del ejemplo, si consideramos que eliminamos el país INGLATERRA, la tabla quedaría del siguiente modo, considerando que tomamos como valor por defecto NGN, que significa ninguno:

Tabla (Relación): PAISES				Tabla (Relación): JUGADORES		
<u>IdPais</u>	Nombre	Habitantes	Idioma	<u>IdJugador</u>	Nombre	<u>IdPais</u>
ESP	ESPAÑA	9.000.000	ESPAÑOL	J001	Raul Lopez	ESP
BORRADA				J002	Robinho	BRA
FRA	FRANCIA	11.345.655	FRANCÉS	J003	Zinedine Zidane	FRA
BRA	BRASIL	23.098.476	PORTUGUÉS	J004	Beckam	NGN
HOL	HOLANDA	5.000.765	HOLANDES			

5. Transformación del modelo E/R al modelo lógico relacional

La transformación del Modelo entidad-relación al Modelo relacional produce cierta pérdida de semántica ya que las entidades y las interrelaciones se convierten en tablas, pero la pérdida de semántica no implica, sin embargo, un peligro para la integridad de la base de datos, ya que se definen las restricciones de integridad referencial que aseguran la conservación de la misma.

5.1. Tres reglas básicas en las transformaciones del modelo E-R

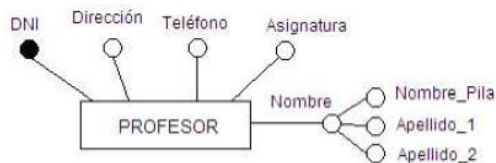
Existen **tres reglas básicas** que se deben cumplir en la transformación del modelo entidad relación al modelo relacional, reglas que se explicarán más detalladamente en los apartados que siguen a éste, son:

- Toda entidad se convierte en una relación (o tabla).
- Toda interrelación N:M se transforma en una relación (también llamada tabla).
- Toda interrelación 1:N se traduce en el fenómeno “propagación de claves”.

5.2. Transformación de entidades

- Cada entidad del esquema E/R dará lugar a una nueva **relación ó tabla** cuya clave primaria es el identificador principal de la entidad.
- Cada atributo de una entidad se transforma en un atributo de la relación creada para la entidad teniendo en cuenta los distintos tipos de restricciones semánticas.
 - **Atributos univaluados:** dan lugar a un atributo de la relación.
 - **Atributos multivaluados:** dan lugar a una nueva relación (tabla) cuya clave primaria es la concatenación de la clave primaria de la entidad en la que se sitúa el atributo multivaluado más el nombre del atributo multivaluado.
 - **Atributos obligatorios:** atributos con la restricción de NOT NULL.
 - **Atributos opcionales:** atributos que pueden tomar ‘valores’ NULL.
 - **Identificador principal:** atributos que forma la clave primaria.
 - **Identificador alternativo:** atributos con la restricción de UNIQUE.

Modelo E/R



Modelo relacional

PROFESOR						
DNI	Nombre_Pila	Apellido_1	Apellido_2	Asignatura	Dirección	Teléfono
32320000	Manuel	Leal	Blanco	Lengua	Real 22	986111111
36101010	Juan	Gómez	Ríos	Inglés	Badajoz 11	650777777
33999999	María	Rodríguez	Sánchez	Física	Pez 77	567888888

Nosotros lo representaremos así:

PROFESORES (DNI, nombrePila, apellido1, apellido2, asignatura, dirección, telefono)

5.3. Transformación de entidades débiles

Cada atributo de una entidad débil se transforma en una columna de la relación, igual que en el caso anterior pero, además, hay que incluir como clave foránea el/los atributos que formen la clave primaria en la tabla principal. Además, en el caso de que la entidad sea débil en identificación, la clave principal de esta relación o tabla estará formada por los atributos de la clave foránea más los atributos que formen la clave parcial de la entidad débil.

❖ DEPENDENCIA EN IDENTIFICACION

Suelen ser siempre **1:N** por lo que no generan tablas nuevas. **La clave primaria de la entidad fuerte tiene que propagarse a la tabla de la entidad débil y formar parte de esta** (además de ser clave ajena).



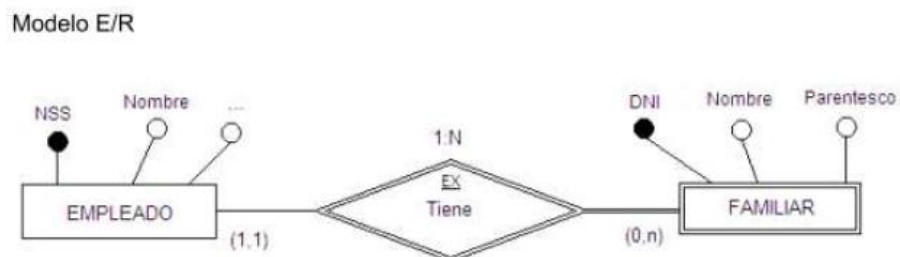
Modelo Relacional:

PRESTAMOS (numPrestamo, importe)

PAGOS (numPrestamo, numPago, fecha, importePago)

❖ DEPENDENCIA EN EXISTENCIA

Una interrelación **1:N** de dependencia en existencia **se transforma propagando la clave primaria como clave ajena de la entidad fuerte a la entidad débil**. Además debe tener la opción de **DELETE ON CASCADE**, es decir, no puede existir ninguna ocurrencia de la entidad hijo que no esté relacionada con una ocurrencia de la entidad padre.



Modelo Relacional:

EMPLEADOS (NSS, nombre)

FAMILIARES (dni, parentesco, nombre)

5.4. Transformación de relaciones binarias

5.4.1. INTERRELACIÓN N:M

Siempre dan lugar a **NUEVA TABLA o RELACIÓN** cuya **clave primaria será la concatenación de las claves primarias de las entidades que interrelacionan**. De esta forma los atributos que forma la clave primaria en esta nueva relación son claves ajenas respecto a las relaciones en las que estos atributos son clave primaria.

Por ejemplo:

Modelo E/R:



Modelo relacional:

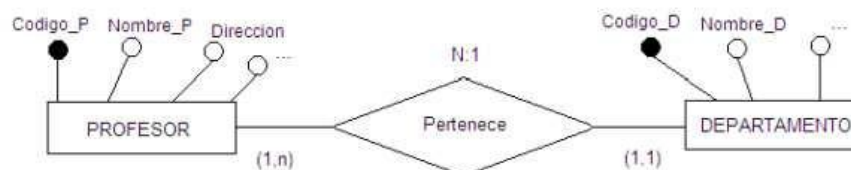
PROFESOR (Clodigo_P, Nombre_P, Direccion...)
CURSO (Clodigo_C, Nombre_C, NumHoras...)
IMPARTE (Clodigo_P, Clodigo_C)

5.4.2. INTERRELACIÓN 1:N

Existen dos posibilidades:

- 1) **Propagación de clave:** Propagar los atributos que forman la clave principal del tipo de entidad de cardinalidad 1 a la que tiene cardinalidad N, desapareciendo el nombre de la interrelación. Si la interrelación tuviese atributos propios, estos se propagarían en el mismo sentido que la clave.

Modelo E/R:



Modelo relacional:

PROFESOR (Clodigo_P, Nombre_P, Dirección..., Clodigo_D)
DEPARTAMENTO (Clodigo_D, Nombre_D...)

- 2) **Transformar la interrelación 1:N como si fuese una interrelación N:M.** Es decir, se crea una nueva relación (tabla) cuya clave primaria es, en este caso, sólo la clave primaria de la tabla a la que le corresponde la cardinalidad N. Por ejemplo, para el modelo E/R anterior, la transformación al modelo relacional daría lugar a las relaciones siguientes:

PROFESOR (Codigo_P, Nombre_P, Dirección...)
DEPARTAMENTO (Codigo_D, Nombre_D...)
PERTENECE (Codigo_P, Codigo_D)

5.4.3. INTERRELACIONES 1:1

Una interrelación de tipo 1:1 es un caso particular de una N:M o también de una 1:N, por lo que no hay regla fija para la transformación: puede crearse una nueva relación o bien efectuar una propagación de clave. El siguiente ejemplo se ha resuelto mediante la creación de una nueva relación.



Modelo relacional:

PROFESOR (Codigo_P, Nombre_P, Dirección...)
CURSO (Codigo_C, Nombre_C, Num_alumnos...)
ES_TUTOR_DE (Codigo_P, Codigo_C)

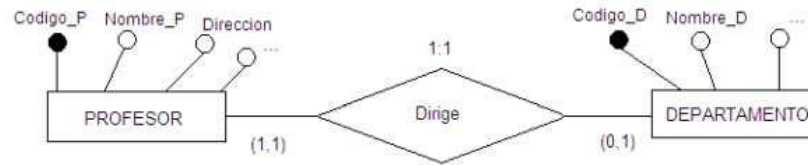
La clave de la nueva relación (ES_TUTOR_DE) puede ser cualquiera de sus dos atributos.

En el caso de la propagación de clave, dicha propagación podría realizarse en ambos sentidos, aunque debemos tener en cuenta dos posibilidades:

- a) **Sí una entidad de las que participan en la interrelación posee cardinalidad (0,1) mientras que la otra posee (1,1)**, es mejor propagar la clave de la entidad con cardinalidad (1,1) para evitar los nulos.

Por ejemplo:

Modelo E/R



Modelo relacional

PROFESOR (Codigo_P, Nombre_P, Dirección...)
DEPARTAMENTO (Codigo_D, Nombre_D..., Codigo_P)

Se ha propagado la clave de la entidad PROFESOR ("Codigo_P") porque es la entidad que tiene cardinalidad (1,1). Esta solución evita los valores nulos que se producirían de haber hecho la propagación en sentido contrario. Es decir, si el paso a modelo relacional se hubiera hecho propagando la clave de la entidad DEPARTAMENTO a la relación PROFESOR, el resultado hubiera sido:

PROFESOR (Codigo_P, Nombre_P, Dirección..., Codigo_D)
DEPARTAMENTO (Codigo_D, Nombre_D...)

Como hay profesores que no dirigen ningún departamento, lo que viene indicado por la cardinalidad (0,1), habrá tuplas en la relación PROFESOR en las que el campo "Codigo_D" no tenga ningún valor, o lo que es igual, tenga valor nulo.

- b) **Sí las dos entidades que participan en la interrelación poseen cardinalidad (0,1)**, puede ser más adecuado realizar la transformación en **una nueva tabla o relación**. Por ejemplo:

Modelo E/R



Modelo relacional

HOMBRE (DNI_H, Nombre_H, Dirección_H...)
MUJER (DNI_M, Nombre_M, Dirección_M...)
CONTRAE_MAT. (DNI_H, DNI_M)

La clave de la nueva relación (CONTRAE_MATRIMONIO) puede ser cualquiera de sus dos atributos.

Al crear una nueva relación, se evitan los valores nulos que se producirían de haber propagado la clave de la relación MUJER a la relación HOMBRE o viceversa, ya que, según refleja la cardinalidad, no todos los hombres ni todas las mujeres están casados.

5.5. Transformación de las relaciones reflexivas

Supondremos que se trata de una relación binaria con la particularidad que las dos entidades son iguales y aplicaremos las reglas vistas en los apartados anteriores.

5.6. Transformación de relaciones ternarias

Se crea una nueva tabla para la relación. Esta tabla tendrá como **claves ajenas** las claves primarias de las relaciones que relaciona, y su **clave primaria** estará **formada por las claves ajenas de aquellas tablas en las que su cardinalidad es (x, N).**

5.7. Transformación de atributos multivaluados

Para los **atributos multivaluados** se creará una relación o tabla que incluya:

- ❖ La **clave primaria** (PK) de la relación correspondiente al tipo de entidad (o al tipo de interrelación) que contiene el atributo multivaluado.
- ❖ Un atributo (A) que corresponda al propio atributo multivaluado.

La **clave primaria de la nueva relación** estará formada por K y A.

Por ejemplo, supongamos en el siguiente esquema el atributo multivaluado "Ubicación", referente a la entidad DEPARTAMENTO:



Al transformarlo al modelo relacional, obtenemos las relaciones que se indican a continuación:

PROFESOR	(<u>Código_P</u> , Nombre_P, Dirección...)
DEPARTAMENTO	(<u>Código_D</u> , Nombre_D..., <u>Código_P</u>)
UBICACIÓN	(<u>Código_D</u> , Ubicación)

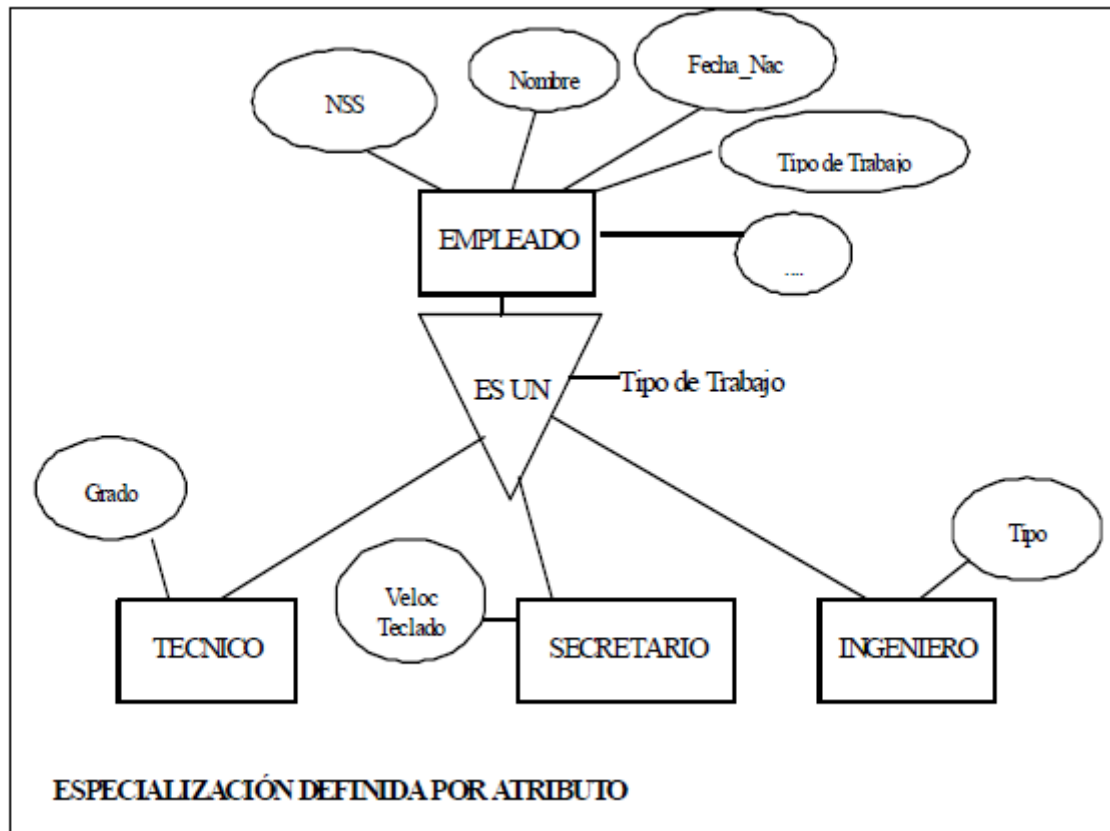
Se ha creado una nueva relación llamada UBICACIÓN, que tiene como atributos:

- a) La clave primaria de la relación DEPARTAMENTO, correspondiente a la entidad del mismo nombre, la cual contiene el atributo multivaluado.
- b) Un atributo llamado "Ubicación", correspondiente al propio atributo multivaluado.

Ambos atributos conforman la clave primaria.

5.8. Transformación de relaciones jerárquicas

Suponiendo el siguiente diagrama ER, para convertir cada superclase C con m subclases {s1,s2,...,Sn} a relaciones ...



... podríamos seguir una de las siguientes opciones:

1) Crear una tabla para la superclase, siguiendo las reglas de un tipo de entidad, y crear tantas tablas como subclases existan y que contengan:

- Los atributos propios de la subclase.
- Los atributos que conformen la clave primaria de la superclase, que constituirán la clave primaria de la relación.

EMPLEADO (NSS_EMP, NOMBRE_EMP, FNAC_EMP, TIPOTRABAJO)

SECRETARIO (NSS_EMP, VELO_SEC)

TECNICO (NSS_EMP, GRADO_TEC)

INGENIERO (NSS_EMP, TIPO_ING)

2) No crear una tabla para la superclase y crear tantas tablas como subclases existan y que contengan:

- Los atributos propios de la subclase que no formen parte de la clave primaria.
- Los atributos propios de la superclase que no formen parte de la clave primaria.
- Los atributos que conformen la clave primaria de la superclase, que constituirán la clave primaria de la relación.

SECRETARIO (NSS_EMP, NOMBRE_SEC, ,FNAC_SEC, TIPOTRABAJO, VELO_SEC)

TECNICO (NSS_EMP, NOMBRE_TEC, FNAC_TEC, TIPOTRABAJO, GRADO_TEC)

INGENIERO (NSS_EMP, NOMBRE_ING,APE1,FNAC_ING, TIPOTRABAJO, TIPO_ING)

Esta opción es válida si las subclases son **disjuntas y total**. Si es solapada una entidad que pertenezca a más de una subclase tendrá los atributos heredados de la superclase almacenados en más de una relación. Si no es total una entidad que no pertenezca a ninguna de las subclases se perdería.

3) Crear una única tabla que contenga:

- Los atributos propios de la superclase que no formen parte de la clave primaria.
- Los atributos propios de cada subclase que no formen parte de la clave primaria.
- Un atributo 't' que indica a que subclase pertenece cada tupla.
- Los atributos que conformen la clave primaria de la superclase, que constituirán la clave primaria de la relación.

- ❖ Esta opción es válida cuando las subclases son **disjuntas** y existe un atributo que indica la subclase a la cual cada tupla pertenece.
- ❖ Esta opción tiene un potencial alto de generar valores nulos si el número de atributos específicos es alto.
- ❖ Si la especialización es parcial 't' puede tener valor nulo y en consecuencia todos los atributos de todas las subclases tendrán valores nulos.
- ❖ Si la especialización es definida por atributo no es necesario el atributo t.

EMPLEADOS (NSS_EMP, NOMBRE_EMP, FNAC_EMP, TIPOTRABAJO, VELO_SEC, GRADO_TEC, TIPO_ING)

- 4) Se puede crear una sola tabla para la superclase como en la opción anterior, pero en lugar de añadir un solo campo "tipo", se añaden varios campos que indiquen si se cumple un perfil, de este modo se soportan las especializaciones inclusivas o solapadas.

EMPLEADOS (NSS_EMP, NOMBRE_EMP, FNAC_EMP, VELO_SEC, GRADO_TEC, TIPO_ING esSecretaria, esTecnico, esIngeniero)

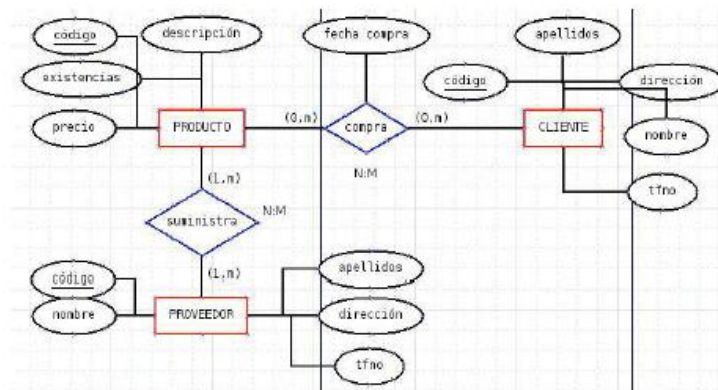
6. Notación

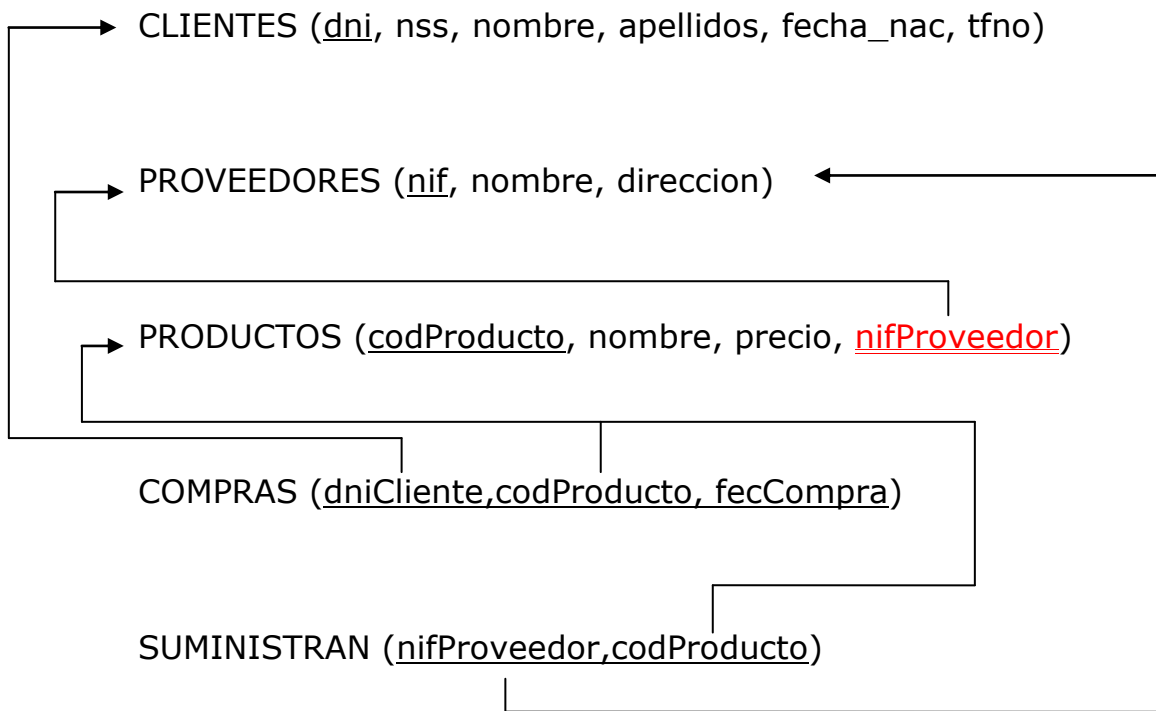
Un esquema relacional se representará mediante un grafo, conocido como grafo relacional; básicamente, se trata de un grafo dirigido cuyos nodos son las relaciones de la BD y los arcos representan las restricciones de clave ajena, y en el aparecerán además de las distintas relaciones con sus atributos y las restricciones de clave ajenas las restricciones de clave primaria, unicidad y obligatoriedad. Las convecciones empleadas para la representación del grafo son:

- El **nombre de las tablas** en mayúsculas y plural seguido de los atributos encerrados entre paréntesis.
- Las claves primarias subrayadas con trazo continuo.
- Las claves alternativas subrayadas con trazo discontinuo.

- d) Las **claves ajenas o foráneas** con doble subrayado y referencian a la relación en la que son clave primaria mediante una flecha. (ver en ejemplo).
- e) Los atributos que pueden tomar valores nulos aparecen con un asterisco (*).
- f) Las opciones para la integridad referencial son:
- ❖ BC: Borrado en Cascada.
 - ❖ BD: Borrado con puesta a valor por Defecto.
 - ❖ BN: Borrado con puesta a Nulos.
 - ❖ BR: Borrado Restringido.
 - ❖ MC: Modificación en Cascada.
 - ❖ MD: Modificación con puesta a valor por Defecto.
 - ❖ MN: Modificación con puesta a Nulos.
 - ❖ MR: Modificación Restringido.

7. Ejemplo





NOTA: En todos los casos se aplicará borrado restringido y modificación en cascada.

8. Documentación

La metodología de la administración española es Métrica v3. En esta metodología la fase de análisis se denomina "ASI = Análisis de Sistemas de Información", en esta fase se indica que:

- 1) Se debe **realizar la especificación de requisitos** (que se comenzó en la fase anterior EVS). En la especificación de requisitos se recogen todos los requisitos definidos por el usuario y que serán necesarios para poder realizar el análisis conceptual de datos.
- 2) Después se tiene que realizar el **modelo conceptual de datos**: se parte de la especificación de requisitos (donde los datos que se necesitan guardar están expresados en lenguaje natural) y se genera el modelo conceptual de datos utilizando alguna TÉCNICA descriptiva como puede ser el Diagrama de E/R.
- 3) Después se tiene que obtener el **modelo lógico de datos** a partir del modelo conceptual de datos.
- 4) Y se debe aplicar las normas de normalización para obtener el modelo lógico de datos normalizado.

Además de incluir el diagrama de DED del modelo lógico de datos normalizado en la fase de ASI, toda la información sobre el modelo lógico de datos queda recogida en el diccionario de datos:

8.1. Diccionario de datos

En el diccionario de datos se recoge la siguiente información sobre el modelo conceptual de datos:

- ❖ Nombre de las entidades y descripción.
- ❖ Nombre de las relaciones y descripción.
- ❖ Atributos de las entidades, descripción de cada atributo y definición (tipo de dato, longitud, restricciones, ...)

Se utiliza la Notación **BNF**

=	Compuesto de
+	y
()	Opcionalidad
{ }	Iteración
[]	Selección de alternativas
* *	Comentario
@	Clave

Se pueden indicar los límites de las iteraciones

Palabra = {Letra}	Sin límite (indeterminado)
Palabra = 1{Letra}	Mínimo 1, máximo indeterminado
Palabra = {Letra}10	Mínimo indeterminado, máximo 10
Palabra = 1{Letra}10	Mínimo 1, máximo 10
Palabra = 10{Letra}10	Exactamente 10

Ejemplos:

Alumno = Nombre + Apellido

Alumno = { Carácter }

Apellido = { Carácter }

Carácter = ["A"- "Z" | "a"- "z"]