

EL MODELO RELACIONAL

1. SGBD relacionales

Hay muchos tipos de SGBD, pero la mayor parte de los utilizados comercialmente en la actualidad son relacionales, es decir, se basan en una cierta teoría o forma de representar los datos para implementar sus herramientas e interfaces, en este caso el Modelo Relacional. Entendemos como representación de los datos, la forma en que se presentan al usuario y que permiten ciertas operaciones para poder manejarlos.

De hecho, en estos SGBD, la información se presenta en forma de tablas, con columnas para las características de los objetos o conceptos que se pretende representar en la tabla, y filas para cada caso concreto o instancia de objeto. Existe un lenguaje considerado como estándar para manejar esas tablas, SQL, que permite crear y modificar tablas, consultarlas, introducir nuevos datos, modificar datos o borrarlos.

1.1. Estructura de una base de datos relacional

En general se puede decir que una base de datos consta de los siguientes elementos:

- **Tabla o relación:** es una estructura bidimensional formada por filas y columnas. En cada tabla se almacena un tipo de **entidad**, es decir, un objeto o elemento que se va a almacenar en la base de datos. Así en una base de datos académica podrá haber información de las siguientes entidades: alumno, profesor, asignatura, centro, curso, etc.

Campos (grado=5)

Filas (cardinalidad=4)

	DNI	Nombre	Apellido1	Apellido2	FechaNacimiento
	48352698Q	Jose	Pérez	Reverte	19/11/1993
	12684259Z	Manuel	Martínez	Juan	15/06/1992
	69834521O	María	Valero	Pérez	01/05/1992
	58741965A	Mercedes	Mendoza	Martínez	25/02/1993

- **Registros:** son cada una de las filas que se almacenan en la tabla. Es decir diferentes instancias de una entidad. (Ej: cada uno de los alumnos es un registro)
- **Campos:** son los atributos que se pueden almacenar de cada entidad. Puede ser un atributo, cualquier característica o propiedad de una entidad. Por ejemplo, posibles atributos de una entidad alumno son: DNI, nombre, apellidos, fecha de nacimiento etc. En cada campo solo se aceptarán valores de un determinado tipo de datos o dominio (texto, numérico, fecha, etc)



- **Relación:** en una base de datos integrada por más de una tabla, probablemente existirán relaciones entre las tablas de la misma. Las relaciones entre tablas también disponen de cardinalidad. Por ejemplo, dada una tabla de Alumnos y otra de Provincias, entre ambas existirá una relación con cardinalidad de uno a varios, ya que un alumno solo puede pertenecer a una provincia pero dada una provincia puede haber varios alumnos que pertenezcan a ella. Desarrollaremos los tipos de relaciones más adelante.

Otros conceptos relacionados con la tabla son el grado y la cardinalidad. El **grado** de una **tabla** es igual al número de columnas que la forman, y es fijo mientras no se cambie el diseño de la base de datos. La **cardinalidad** se corresponde con el número de filas de la tabla, por lo cual es variable siempre que la tabla permita realizar altas y bajas.

Esta organización de la información, permite recuperar de forma flexible los datos de una o varias tablas, así como combinar registros de diferentes tablas para formar otras nuevas.

1.2. Integridad referencial

La integridad referencial es una propiedad deseable en las bases de datos. Gracias a la integridad referencial se garantiza que una entidad (fila o registro) siempre se relaciona con otras entidades válidas, es decir, que existen en la base de datos. Implica que en todo momento dichos datos sean correctos, sin repeticiones innecesarias, datos perdidos y relaciones mal resueltas.

Todas las bases de datos relacionales gozan de esta propiedad gracias a que el software gestor de base de datos vela por su cumplimiento.

Para garantizar esa integridad referencial entran en juego los siguientes elementos:

- **Clave primaria:** se conoce como clave primaria al campo o conjunto de campos que identifican unívocamente un registro de una tabla. Solo puede haber una clave primaria por tabla. Por ejemplo: para la tabla alumnos del ejemplo anterior, el campo que identificaría un alumno del resto sería el campo NIF, por tanto el campo NIF sería la clave primaria de la tabla Alumnos.
- **Clave ajena:** se conoce como clave ajena de una tabla a un campo o conjunto de campos que se utilizan para relacionar esa tabla con otra. El valor de la clave ajena debe concordar con el valor de la clave primaria de otra tabla. Ejemplo: el campo IdProvincia es una clave ajena que relaciona el alumno con una provincia.

DNI	Nombre	Apellido1	Apellido2	FechaNacimiento	IdProvincia
12684259Z	Manuel	Martínez	Juan	15/06/1992	1
48352698Q	Jose	Pérez	Reverte	19/11/1993	1
58741965A	Mercedes	Mendoza	Martínez	25/02/1993	1
69834521O	María	Valero	Pérez	01/05/1992	2

IdProvincia	Provincia
1	Alicante
2	Valencia
3	Castellón
4	Murcia
0	

- **Clave candidata:** se conoce como clave candidata de una tabla al campo o conjunto de campos que cumplen las siguientes propiedades:
 - **Unicidad.** No existen dos registros en una tabla que tengan el mismo valor para ese campo o campos.
 - **Minimalidad.** Si la clave candidata consta de más de un campo, no es posible eliminar de la clave ningún campo sin que se deje de cumplir la propiedad de Unicidad.

De las claves candidatas que haya en una tabla, una se elegirá una como clave primaria de esa tabla, el resto se denominarán **claves alternativas**.

Las claves alternativas y las claves primarias, por tanto, no pueden tener valores repetidos y siempre tienen que tener valor (no pueden ser nulas).

2. Transformar el Modelo EER al Modelo Relacional

Para transformar un **modelo entidad-relación** a **modelo relacional** seguiremos las siguientes reglas:

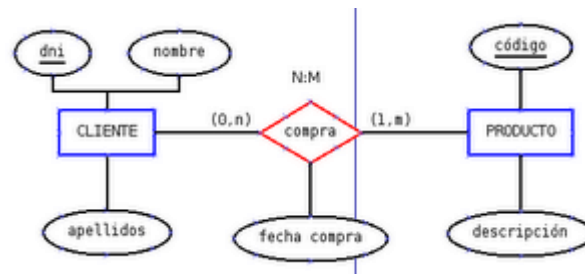
- Toda **entidad** del modelo entidad-relación se transforma en una tabla.
- Cualquier **atributo** de una entidad se transforma en un campo dentro la tabla, manteniendo las claves primarias (campos identificadores)
- Las **relaciones muchos a muchos** (N:M) se transforman en una nueva tabla que tendrá como clave primaria la concatenación de los atributos clave de las entidades que relaciona.
- En las **relaciones uno a muchos** (1:N) se pueden tener dos casos:
 - Si la entidad que participa con cardinalidad máxima uno lo hace también con cardinalidad mínima uno, entonces se propaga el atributo de la entidad que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N, desapareciendo el nombre de la relación. Si existen atributos en la relación éstos también se propagarán.
 - Si la entidad que participa con cardinalidad máxima uno lo hace también cardinalidad mínima cero, entonces se crea una nueva tabla formada por las claves de cada entidad y los atributos de la relación. La clave primaria de la nueva tabla será el identificador de la entidad que participa con cardinalidad máxima N.
- En el caso de las **relaciones uno a uno** (1:1) también pueden darse dos casos:
 - Si las entidades poseen cardinalidades (0,1), la relación se convierte en una tabla.
 - Si una de las entidades posee cardinalidad (0,1) y la otra (1,1), conviene propagar la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad con cardinalidad (0,1). Si ambas entidades poseen cardinalidades (1,1) se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra.

- En el caso de las **relaciones N-arias** se aplica la misma regla que para las relaciones N:M
- En el caso 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 puntos anteriores.

Veamos algunos ejemplos.

Relaciones N:M

Supongamos el siguiente **modelo entidad-relación**.



En este caso la relación “compra” se transforma en una nueva tabla cuya clave primaria estará formada por los atributos dni, que es la clave primaria de cliente, y código, que es la clave primaria de producto. Además tendrá como campo fecha compra, ya que este atributo forma parte de la relación.

El **modelo relacional** quedaría de la siguiente forma:

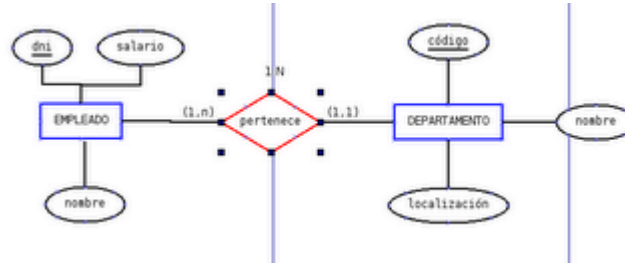
CLIENTE
 CP(dni)
 Nombre
 Apellidos

PRODUCTO
 CP(codigo)
 descripción

COMPRAS
 CP(dni_cliente,codigo_producto)
 Fecha_compra
 C.Aj(dni_cliente)=> Cliente
 C.Aj(codigo_producto)=>Producto

Relaciones 1:N

Veamos ahora el caso de una relación 1:N. En el siguiente **modelo entidad-relación** un empleado pertenece a un único departamento (debe pertenecer a uno obligatoriamente), y un departamento tiene 1 o más empleados.



En este caso se propaga el atributo código de departamento a la tabla EMPLEADO. El **modelo relacional** quedaría de la siguiente manera:

EMPLEADO

CP(dni)

nombre

salario

C.Aj(codigo_departamento)=>departamento

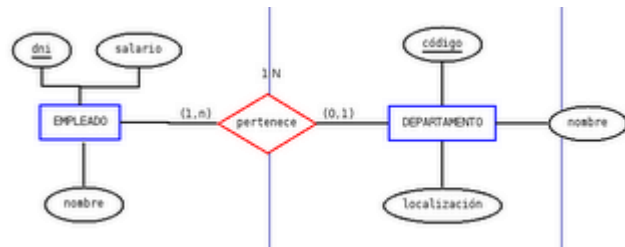
DEPARTAMENTO

CP(codigo)

nombre

localizació

Imaginemos ahora que pudiera darse el caso de que hubiera empleados que no pertenecieran a ningún departamento.

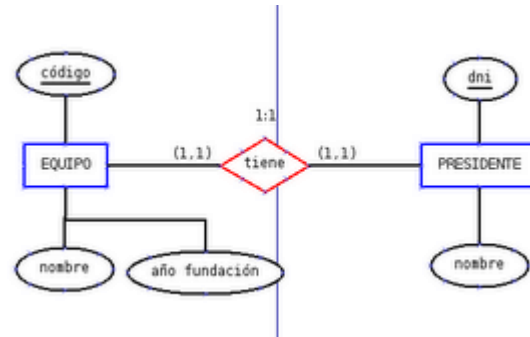


En este caso la entidad que participa con cardinalidad máxima 1, DEPARTAMENTO, también lo hace con cardinalidad mínima 0, ya que puede haber empleados que no pertenezcan a ningún departamento. Así pues, se crea una nueva tabla formada por dni de EMPLEADO y código de DEPARTAMENTO. En esta nueva tabla dni de EMPLEADO será la clave primaria. El **modelo relacional** quedaría de la siguiente forma:

- EMPLEADO(CP(dni),nombre,salario)
- DEPARTAMENTO(CP(código),nombre,localización)
- PERTENECE(CP(dni_empleado),C.Aj(código_departamento)=>departamento)

Relaciones 1:1

Veamos ahora el caso de una relación 1:1 a través del siguiente ejemplo. En el siguiente **modelo entidad-relación** un equipo de fútbol tiene a un único presidente y un presidente preside a un único club de fútbol.



En este ejemplo, tal y como dicen las reglas, podemos propagar la clave de cualquier tabla a la tabla resultante de la otra. Es decir, tenemos dos opciones, o mover la clave de PRESIDENTE a EQUIPO o mover la clave de EQUIPO a PRESIDENTE. El **modelo relacional** podría quedar de cualquiera de las dos formas siguientes:

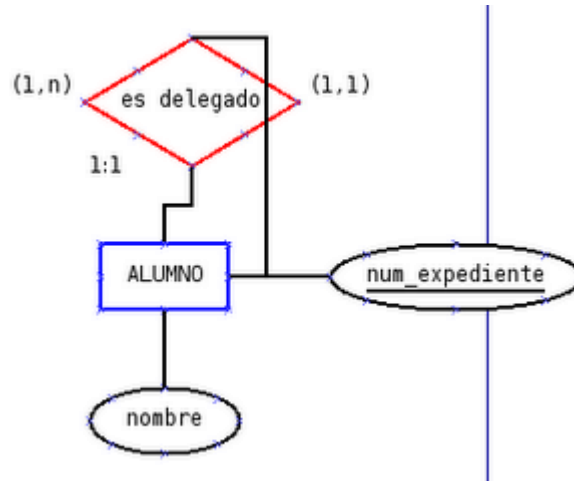
- EQUIPO(CP(código), nombre, año_fundación)
- PRESIDENTE(CP(dni), nombre, CAj(código_equipo)=>equipo)

ó

- EQUIPO(CP(código, nombre), año_fundación, CAj(dni_presidente)=>presidente)
- PRESIDENTE(CP(dni), nombre)

Relaciones reflexivas

Veamos ahora como quedaría en el modelo relacional la siguiente relación reflexiva. En el siguiente modelo entidad-relación un ALUMNO es delegado de varios ALUMNOS y un ALUMNO tiene obligatoriamente un delegado y sólo a uno.



Como podemos observar en las reglas de transformación, en este caso la relación reflexiva se trata como si fuera una relación binaria con la particularidad de que las dos entidades son iguales. Al tratarse de una relación 1:N se propagará la clave de la entidad **ALUMNO** a la entidad **ALUMNO**, quedando el modelo relacional de la siguiente forma:

- $ALUMNO(CP(num_expediente), nombre, C.Aj(num_expediente_delegado) \Rightarrow alumno)$