

Parte 1/3

The diagram illustrates three types of database relationships: Hierarchical, Network, and Relational.

- Hierarchical:** Shows a tree structure where one record (parent) can have one or more child records. It includes a diagram and a table with columns: ID, Nombre, and Sexo.
- Network:** Shows a structure where one record can have one or more child records, and a record can be a child to more than one parent record. It includes a diagram and a table with columns: ID, Nombre, and Sexo.
- Relational:** Shows a structure where one record can have one or more child records, and a record can be a child to more than one parent record. It includes a diagram and a table with columns: ID, Nombre, and Sexo.

Departamento de Tecnologías de la Información
Universidad de Huelva

Objetivos

- ☐ Conocer el modelo de datos relacional por ser el más utilizado en los SGBD
- ☐ Conocer los mecanismos del modelo relacional para expresar restricciones de integridad
- ☐ Conocer los principales lenguajes propuestos para el modelo relacional

Contenidos

- 3.1 Introducción
- 3.2 Estructura del modelo relacional
 - 3.2.1. Dominio y atributo
 - 3.2.2 Tupla y relación
 - 3.2.3 Características de las relaciones

Contenidos

3.3 Restricciones del modelo relacional

3.3.1 Restricciones de dominio

3.3.2 Restricciones de clave

3.3.3 Restricciones de integridad: integridad referencial y claves ajenas

3.4 Lenguajes relacionales

3.4.1 Álgebra relacional

3.4.2 Cálculo relacional

3.4.2.1 Cálculo relacional de tuplas

3.4.2.2 Cálculo relacional de dominios

Duración

□ 5 clases

Bibliografía

- ❑ Capítulos 7 y 9 de [Elmasri, 02]
- ❑ Capítulo 3 de [Silberschatz, 06]
- ❑ Capítulos 3 y 4 de [Connolly, 05]

3.1 Introducción

- Presentado por Edgar F. Codd de IBM Research en 1970
- El objetivo fundamental del modelo propuesto por Codd es mantener la independencia de la estructura lógica de la BD respecto al modo de almacenamiento y a otras características de tipo físico.
- Puedes consultar más información sobre la vida y obra de Codd en:

http://es.wikipedia.org/wiki/Edgar_Frank_Codd

- Es el modelo lógico que han adoptado, de forma estándar, los SGBD comerciales
- El modelo se basa en la teoría de las relaciones: utiliza la **relación matemática** (similar al concepto de tabla) como sistema de representación del almacenamiento
- En el modelo relacional, la base de datos se representa como un conjunto de relaciones que se denominan **tablas**

- Con idea de trabajar con un esquema relacional a lo largo del tema, supongamos que deseamos diseñar una base de datos a partir del siguiente *análisis de requisitos*:

Se desea diseñar una base de datos para gestionar la información de una Escuela de Ingeniería Informática. En dicha BD se almacenará información sobre los alumnos, los profesores y las asignaturas que se imparten en el centro. Además se mantendrá toda la información relativa a las calificaciones de los alumnos en cada asignatura de la que se matricula. Puesto que una asignatura la pueden impartir varios profesores, se desea almacenar la información del profesor que imparte una asignatura a un determinado alumno.

Por otra parte, guardaremos información sobre los ordenadores utilizados por los alumnos para realizar sus prácticas, sabiendo que un alumno usará un único ordenador durante todo el curso, pero que un ordenador será utilizado por varios alumnos.

Supongamos que para cursar ciertas asignaturas es recomendable haber cursado otras asignaturas, por lo que deberá almacenarse cierta información de “recomendaciones” entre asignaturas.

Por último, queremos saber quién es el profesor responsable de cada asignatura, sabiendo que sólo hay uno por asignatura, pero que un profesor puede ser responsable de varias asignaturas.

3.2 Estructura del modelo relacional

- El elemento básico del modelo relacional es la **relación**, que puede representarse mediante una tabla en dos dimensiones (con filas y columnas). Por tanto, una base de datos puede representarse como una colección de tablas.
- Los elementos que se distinguen en una tabla (relación) en el modelo relacional son:
 - Un **nombre** de tabla único, que la distingue de las demás.
 - Un conjunto de columnas, denominadas **atributos**, que representan propiedades de la tabla y que están caracterizadas por un nombre.
 - El **dominio** es el tipo de los valores posibles que pueden tomar cada uno de los atributos.
 - Un conjunto de filas, llamadas **tuplas**, que son los distintos elementos de la tabla.
 - El número de filas (tuplas) de una relación se denomina **cardinalidad**.
 - El número de columnas (atributos) de una relación se denomina **grado**.

nombre de la relación

ALUMNO

atributos

cardinalidad

nAI	dni	nombre	fechaNac	lugar
145	29.254.123	Manuel López González	15/03/75	Huelva
214	29.147.158	Mercedes Gómez Martín	20/06/76	Huelva
112	44.125.325	Francisco Gallego Macías	01/01/75	Cádiz
220	29.555.247	Beatriz Rico Vázquez	12/11/74	Cádiz

tuplas

grado

Las tablas o relaciones (con algunos de sus atributos) que, tras el correspondiente análisis, surgen del enunciado propuesto, son las siguientes:

ALUMNO (nAl, dni, nombre, fechaNac, ...)

PROFESOR (nPr, dni, nombre, despacho, ...)

ASIGNATURA (idAsig, nombre, créditos, ...)

ORDENADOR (idOrd, tipo, ubicación, ...)

MATRICULA (alum, idAsig, prof, año, feb_jun, sep, dic, ...)

RECOMENDACIONES (idAsig1, idAsig2, ...)

3.2.1 Dominio y atributo

- Un **dominio** D es un conjunto finito de valores *homogéneos*, es decir, del mismo tipo.

Todo dominio viene definido por:

- **Nombre.** Para referenciarlo y ayudar a interpretar sus valores
- **Tipo de datos.** Puede ser simple (carácter, entero, etc.) o definido por el usuario (números de teléfono, peso, etc.)

- Un dominio se pueden definir por:

- **Intensión.** Especificando el tipo de datos de forma explícita

Números naturales entre 6 y 16

- **Extensión.** Indicando el conjunto de valores admitidos

Capitales = { Almería, Cádiz, Huelva, ... }, Colores = { Rojo, Verde, Azul, ... }

- Un **atributo** A es el papel que desempeña un determinado dominio D en una relación.
- Se dice que D es el dominio de A, y se denota como $\text{dom}(A)=D$.
- Cada atributo toma sus valores de un único dominio, sin embargo, varios atributos pueden tener el mismo dominio.

Ejemplo:

dni_válido = cadenas de caracteres formadas por 9 dígitos

$\text{dom}(\text{dniAlumno}) = \text{dni_válido}$

$\text{dom}(\text{dniProfesor}) = \text{dni_válido}$

3.2.2 Tupla y relación

- Una **tupla** es una lista ordenada de valores $t = \langle v_1, v_2, \dots, v_n \rangle$, donde cada valor v_i es un elemento del dominio $dom(A_i)$, o bien es un **valor nulo**
- Existen básicamente dos formas de definir una relación:
 - **Intensión, esquema o cabecera de relación.**
 - $R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$
 - Conjunto de n pares atributo-dominio $(A_i:D_i)$
 - Se utiliza para describir una relación y **no suele cambiar** en el transcurso del tiempo
 - **Extensión o estado de relación.**
 - Se denota como: $r(R)$
 - Conjunto de m tuplas $\{t_1, t_2, \dots, t_m\}$
 - Cada tupla es un conjunto de n pares atributo-valor $\{(A_i:V_{ij})\}$ donde V_{ij} es el valor j del dominio D_i asociado al atributo A_i
 - **Varía** con el transcurso del tiempo

Relaciones definidas por intensión o esquemas de relaciones

ALUMNO (nAl: D_códigos, dni: D_dni, nombre: D_cadenas, fechaNac: D_fecha, ...)

PROFESOR (nPr: D_códigos, dni: D_dni, nombre: D_cadenas, despacho: D_números, ...)

ASIGNATURA (idAsig: D_idAsig, nombre: D_cadenas, créditos: D_números, ...)

ORDENADOR (idOrd: D_idOrd, tipo: D_tipos, lugar: D_lugares, ...)

MATRICULA (alum: D_dni, idAsig: D_idAsig, prof: D_dni, año: D_año, ...)

RECOMENDACIONES (idAsig1: D_idAsig, idAsig2: D_idAsig, ...)

Relación definida por extensión o estado de relación**ORDENADOR**

idOrd	tipo	lugar	...
Ord025	Servidor NT	Control	
Ord008	PC Prácticas	Aula 7	
Ord009	PC Prácticas	Aula 7	
Ord010	PC Prácticas	Aula 8	
Ord040	Servidor Multimedia	Aula 10	
Ord035	Servidor de Impresión	Control	

3.2.3 Características de las relaciones

- Cada relación tiene un **nombre distinto** al del resto de las relaciones
- **No** puede haber **tuplas repetidas** y además, entre éstas **no** existe **orden** establecido
- **No** puede haber dos atributos con **igual nombre** en una relación. Además, el orden de los atributos no es relevante
- Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor
- Puede haber tuplas con un valor desconocido en algunos de sus atributos (valor **nulo**)
- Una relación (tabla) puede representar hechos sobre entidades o puede representar la relación (interrelación) existente entre varias entidades

3.3 Restricciones del modelo relacional

- Las restricciones del modelo relacional se refieren, principalmente, a estructuras o estados no permitidos.
- Los datos almacenados en la BD han de adaptarse a las estructuras impuestas por el modelo relacional, y han de cumplir las restricciones de usuario.

3.3.1 Restricciones de dominio

- Especifican que el valor de cada atributo A debe ser un valor atómico o indivisible del dominio.
- Por ejemplo, si definimos el atributo "dirección" para almacenar el lugar de residencia de las personas, el valor de ese atributo es indivisible, es decir, no se puede separar en calle, número, piso, etc.

3.3.2 Restricciones de clave

- Todas las tuplas deben ser distintas. No puede haber dos tuplas que tengan la misma combinación de valores para **todos** sus atributos
- En general existen diferentes subconjuntos de atributos (formados por uno o más atributos) que hacen cumplir la propiedad de que dos tuplas no pueden tener los mismos valores para dicho subconjunto
- Sea SC dicho subconjunto. Para dos tuplas distintas cualesquiera t_1 y t_2 , se cumple que $t_1[SC] \neq t_2[SC]$
- Todo conjunto de atributos SC se denomina **superclave** del esquema de relación R , el cual especifica la **restricción de unicidad**
- Toda relación tiene, al menos una superclave (el conjunto de todos los atributos), pero una superclave puede tener atributos redundantes

- Una **clave** de una relación es un conjunto no vacío de atributos que identifica unívocamente cada tupla, y que además es mínimo
- **Ejemplo:** en la relación ORDENADOR podemos elegir como superclave el conjunto de atributos {idOrd, tipo}, ya que identifican a una única tupla, pero no es mínima, ya que si eliminamos el atributo "tipo", todavía podemos identificar unívocamente las distintas tuplas.
- Una clave puede ser:
 - simple (formada por un atributo)
 - compuesta (formada por varios atributos)
- Generalmente será simple, pero hay veces en que es necesario crear una clave compuesta.
- **Ejemplo:** la clave de la relación MATRICULA (alum, idAsig, ...) debe estar formada, al menos, por los atributos "alum" e "idAsig", para que un alumno pueda matricularse en más de una asignatura y una asignatura pueda tener más de un alumno matriculado.

- Una relación puede tener más de una clave
- Cada una de ellas se denomina **clave candidata**
- Se denomina **clave primaria** aquella que elige el diseñador como clave
- Al resto de las claves candidatas no elegidas, se denominan **claves alternativas**

Esquemas de relaciones del ejemplo con sus claves primarias y claves alternativas

ALUMNO (nAl, dni, nombre, fechaNac, ...)
CP: nAl
Único: dni

PROFESOR (nPr, dni, nombre, despacho, ...)
CP: nPr
Único: dni

ASIGNATURA (idAsig, nombre, créditos, ...)
CP: idAsig
Único: nombre

ORDENADOR (idOrd, tipo, lugar, ...)
CP: idOrd

MATRICULA (alum, idAsig, prof, año, feb_jun, sep, dic, ...)
CP: (alum, idAsig, año)

RECOMENDACIONES (idAsig1, idAsig2, ...)
CP: (idAsig1, idAsig2)

3.3.3 Restricciones de integridad

- Una base de datos relacional suele contener muchas relaciones o tablas, algunas de las cuales se encuentran relacionadas.
- Se define **esquema de base de datos relacional** a un conjunto de esquemas de relaciones, los cuales deben cumplir ciertas restricciones, que llamaremos **restricciones de integridad**.
- Se define **ejemplar de base de datos relacional** a un conjunto de ejemplares de relaciones que satisfacen las restricciones de integridad. También se conoce como instancia de la base de datos o estado de la base de datos.

En el fichero [ejemplar.pdf](#) * se muestra un ejemplar de la base de datos que estamos utilizando como ejemplo

* [visitar página web de la asignatura](#)

- **Restricción de integridad de entidades:** ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo.
- **Restricción de integridad referencial:** se especifica entre dos relaciones y sirve para mantener la consistencia entre las tuplas de ambas relaciones. Si en una relación hay una referencia a una tupla de otra relación, ésta deberá referenciar a una tupla existente en la relación referenciada. Se apoya en el concepto de **clave ajena** o **externa** (*foreign key*).

ALUMNO

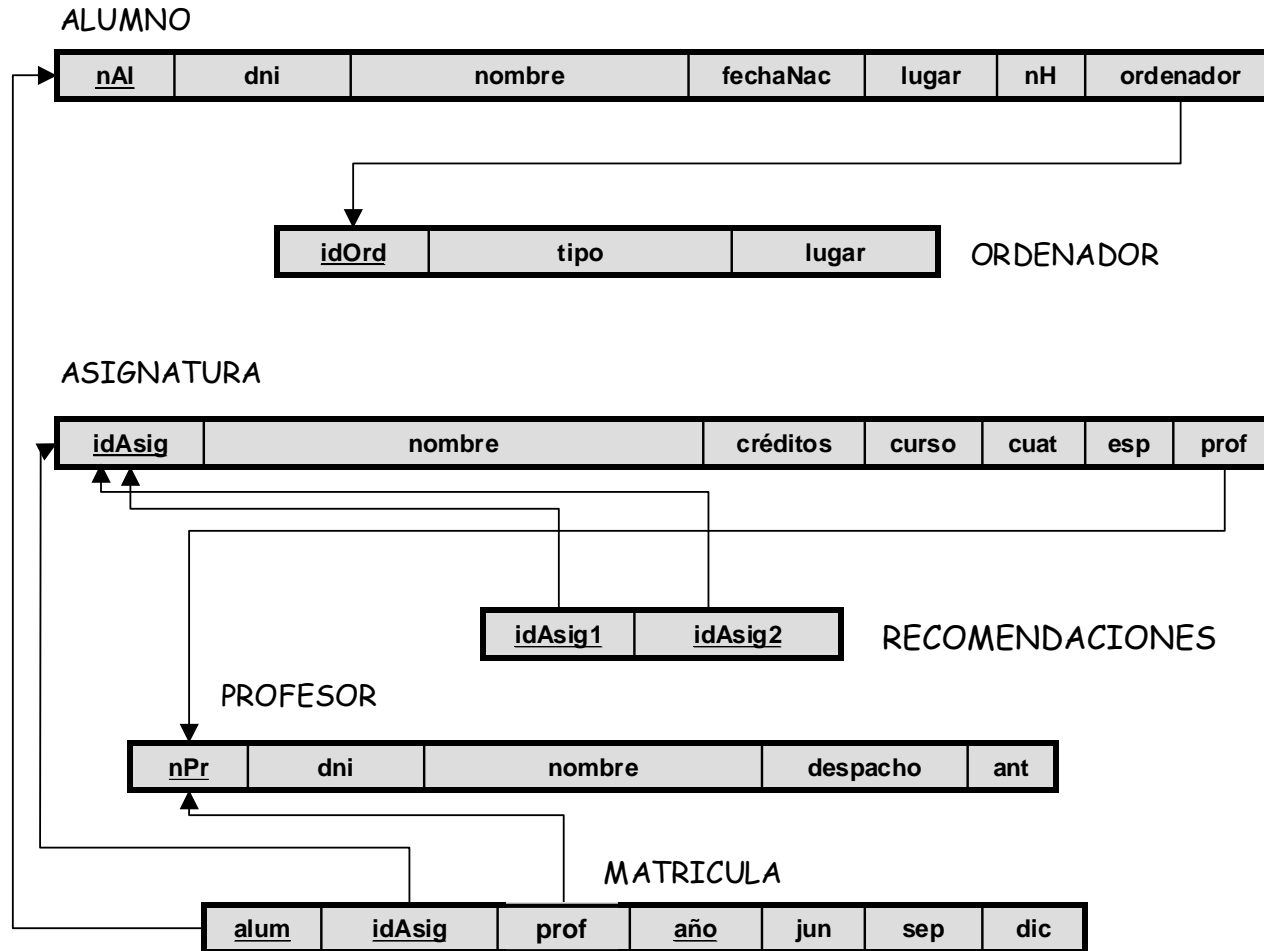
nAl	dni	nombre	fechaNac	lugar	nH	ordenador
145	29.254.123	Manuel López González	15/03/75	Huelva	0	Ord009
214	29.147.158	Mercedes Gómez Martín	20/06/76	Huelva	1	Ord010
112	44.125.325	Francisco Gallego Macías	01/01/75	Cádiz	2	Ord009
088	29.214.856	Teresa Díaz Camacho	25/10/71	Madrid	4	Ord010
220	29.555.247	Beatriz Rico Vázquez	12/11/74	Cádiz	2	Ord025

ORDENADOR

idOrd	tipo	lugar
Ord025	Servidor NT	Control
Ord008	PC Prácticas	Aula 7
Ord009	PC Prácticas	Aula 7
Ord010	PC Prácticas	Aula 8
Ord040	Servidor Multimedia	Aula 10

- Un conjunto de atributos de una relación R_2 es una **clave ajena** de la relación R_1 si satisface las dos reglas siguientes:
 1. Los atributos de la clave ajena en R_2 tienen el mismo dominio que los atributos de la clave primaria de R_1 . Se dice que los atributos de la clave ajena hacen referencia a la relación R_1
 2. Un valor de la clave ajena en una tupla t_2 de la relación R_2 debe corresponderse con un valor de la clave primaria en alguna tupla t_1 de la relación R_1 , o bien es nulo si se desconoce. En el caso de no ser nulo, tenemos que $t_2[\text{clave ajena}] = t_1[\text{clave primaria}]$, y se dice que la tupla t_2 hace referencia a la tupla t_1
- R_1 y R_2 pueden ser la misma relación
- La integridad referencial es una restricción de comportamiento y es el usuario quién la define al describir el esquema relacional.

Representación de las claves ajenas



Representación de las claves ajenas

ALUMNO (nAl, dni, nombre, fechaNac, lugar, nH, ordenador)

CP: nAl

Único: dni

CAj: ordenador → ORDENADOR (idOrd)

PROFESOR (nPr, dni, nombre, despacho, ant)

CP: nPr

Único: dni

ASIGNATURA (idAsig, nombre, créditos, curso, cuat, esp, prof)

CP: idAsig

Único: nombre

CAj: prof → PROFESOR (nPr)

ORDENADOR (idOrd, tipo, lugar)

CP: idOrd

RECOMENDACIONES (idAsig1, idAsig2)

CP: (idAsig1, idAsig2)

CAj: idAsig1 → ASIGNATURA (idAsig)

CAj: idAsig2 → ASIGNATURA (idAsig)

MATRICULA (alum, idAsig, prof, año, feb_jun, sep, dic)

CP: (alum, idAsig, año)

CAj: alum → ALUMNO (nAl)

CAj: idAsig → ASIGNATURA (idAsig)

CAj: prof → PROFESOR (nPr)

- **Definición formal de restricción de integridad referencial:**

Si una relación R_2 (relación que referencia) tiene un conjunto de atributos que se corresponde con la clave primaria de la relación R_1 (relación referenciada), todo valor de dicho conjunto debe concordar con un valor de la clave primaria de R_1 o ser nulo

Ese conjunto es, por tanto, una clave ajena de la relación R_2

En nuestro ejemplo, esto quiere decir que para una determinada tupla de MATRICULA (150, A013, 12, 2000, nulo, 5, nulo), cuyas claves ajenas son **alum**, **idAsig** y **prof**, que hacen referencia a las claves primarias de las tablas ALUMNO, ASIGNATURA y PROFESOR respectivamente, deben existir el alumno “150”, la asignatura “A013” y el profesor “12” en las citadas tablas.

Otro tipo de restricciones

- **Restricción de obligatoriedad:** se utiliza para no permitir valores nulos en ciertos atributos.
- **Restricción de unicidad:** se utiliza para indicar que el valor de un atributo o un conjunto de atributos no se puede repetir. La clave primaria posee las restricciones de unicidad y obligatoriedad.

ALUMNO (nAl, dni, nombre, fechaNac, lugar, nH, ordenador)

CP: nAl

Único: dni

VNN: dni, nombre

CAj: ordenador → ORDENADOR (idOrd)

- **Restricciones de integridad semántica:** dependen del contexto que estemos modelando.

- Ejemplos:

nota \leq 10

salario1 > salario2

si la nota en feb_jun es \geq 5, la nota en septiembre y diciembre debe ser un valor nulo

Conservación de la integridad referencial

- EL SGBD debe proporcionar un mecanismo para garantizar la integridad referencial cuando se realizan operaciones de inserción, actualización o eliminación de tuplas
- Existen dos posibilidades para manejar la integridad referencial:
 1. impedir que se introduzca información que no garantiza la integridad referencial.

Operación restringida. (RESTRICTED)

2. permitir la introducción de dicha información y realizar acciones consecuentes para que se mantenga la integridad referencial.

Operación con transmisión en cascada. (CASCADE)

Operación con puesta a nulos. (SET NULL)

Operación con puesta a valor por defecto. (SET DEFAULT)