

☐ **PARTE II. MODELO RELACIONAL. ESTÁTICA**

- III.4 INTRODUCCIÓN AL MODELO RELACIONAL
- III.5 ESTRUCTURA DEL MODELO
- III.6 RESTRICCIONES
- III.7 EL MODELO RELACIONAL Y LA ARQUITECTURA ANSI
- III.8 LAS 12 REGLAS DE CODD
- III.9 EJEMPLO DE MODELADO RELACIONAL

**Evolución del Modelo Relacional**

- 1970** E.F. Codd, "*Relational Model of Data for Large Shared Data Banks,*" Communications of the ACM , 13, 1970, pp. 377--387.  
Codd (IBM) propone los modelos de datos y introduce el modelo relacional
- 1975-77 **SEQUEL** (*Structured English QUery Language*) sobre SEQUEL-XRM de IBM
- 1979** **ORACLE**: primer SGBDR (basado en SQL);
- 1980 le siguen DB2, Sybase, Informix... o copian su interfaz (Adabas, Ingres...)
- 1982 **ANSI** presenta una propuesta basada en SQL de IBM
- 1985** Codd critica la *debilidad* del estándar (poco fiel al modelo relacional)
- 1986 primera estandarización: el **ANSI-SQL**
- 1987 estandarización **ISO**: el SQL-ISO (9075)
- 1990** versión 2 del Modelo Relacional: **DRDA** (Distributed Relational Database Access) de IBM en su arquitectura SAA (System Application Architecture)
- 1992 nuevo estándar: **SQL2 (SQL92)** de **ISO**, con tres niveles (entry, intermediate, full)
- 1999 **revisión** del estándar SQL (SQL3)

### Origen

**E.F. Codd**

"A RELATIONAL MODEL OF DATA FOR LARGE SHARED DATA BANKS"  
-CACM (1970) -

- No sólo presenta un "nuevo" modelo de datos
- Introduce una teoría de los modelos de datos
- El artículo sigue siendo actual
- Ha constituido una fuente fundamental de la investigación en Bases

### Objetivos del Modelo Relacional

- Independencia físico / lógico
- Eliminación de redundancias
- Flexibilidad
- Uniformidad
- Sólido fundamento teórico

#### PROBLEMAS

- Dificultades de instrumentación inicialmente (la comercialización de sistemas relacionales se hizo esperar más de 10 años)
- Escaso rendimiento (ha mejorado sustancialmente)
- Falta de capacidad semántica (se va añadiendo)

**“La vista relacional de los datos (...) parece ser superior al modelo de grafos o en red (...) Proporciona un medio de describir datos con su estructura natural únicamente, es decir sin superponer ninguna estructura adicional con el propósito de su representación en la máquina”**

CODD (1970)

**INDEPENDENCIA FÍSICO/LÓGICO**

Independencia de la representación lógica de los datos respecto al almacenamiento interno.

○ **Independencia de la ordenación**

○ **Independencia de la indexación**

○ **Independencia de los camino de acceso**

**INDEPENDENCIA FÍSICO/LÓGICO**

- ❑ El modelo relacional se basa en la noción matemática de relación:

$$R: N \times N \text{ ,, } \quad R \{(x, y) / y = x^2\}$$

$$R \{(1, 1), (2, 4), (3, 9), \dots\}$$

**Dominio (D<sub>i</sub>):** totalidad de valores de la misma naturaleza en el universo de discurso

**Atributo (A<sub>i</sub>):** propiedad común a los elementos de una generalidad definida sobre un dominio

**Relación:** Subconjunto del producto cartesiano de un conjunto de  $n$  dominios (no necesariamente distintos). Sobre cada uno de esos dominios toman valores los  $n$  atributos de la relación. ( $n$  es el **grado** de la relación)

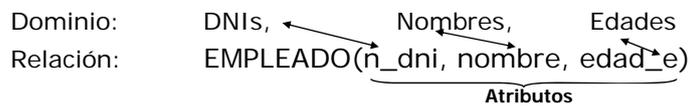
$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

- ❑ Una **tabla** es la estructura empleada para representar una relación:
- Las columnas de la tabla son los atributos de la relación.
  - Cada fila de la tabla se denomina **tupla** y representa un elemento del conjunto definido por la relación.

**Un atributo y un dominio pueden recibir el mismo nombre, pero es preciso distinguirlos:**

- Un atributo está siempre asociado a una relación, mientras que un dominio tiene existencia con independencia de las relaciones
- Un atributo representa una propiedad (*que puede ser la identificación*) de una relación
- Un atributo toma valores de un dominio
- Varios atributos distintos (*de la misma o de diferentes relaciones*) pueden tomar sus valores del mismo dominio

EJEMPLO:



**INTENSIÓN Y EXTENSIÓN DE UNA RELACIÓN**

□ **Intensión:** Definición invariante en el tiempo de una relación; también denominada *esquema de relación*.

EMPLEADO(DNI\_E: Dnis, Nombre: Nombres, Edad\_e: Edades)

$$R(\{A_i: D_i\}_{i=1 \dots n})$$

□ **Extensión:** Definición de la relación a través del conjunto de tuplas que, en un momento determinado, cumplen el esquema de relación.

EMPLEADO	DNI_E	Nombre	Edad_e
	5553344	Juan García	60
	5554545	Eladio Pérez	40
	5556767	José López	25
	5558899	Luis Rodríguez	30

COMPARACIÓN DE TERMINOLOGÍA

<div style="border: 1px solid black; padding: 2px; display: inline-block;">RELACIÓN</div>	~	<div style="border: 1px solid black; padding: 2px; display: inline-block;">TABLA</div>	~	<div style="border: 1px solid black; padding: 2px; display: inline-block;">FICHERO</div>
TUPLA ATRIBUTO GRADO CARDINALIDAD		FILA COLUMNA N. DE COLUMNAS N. DE FILAS		REGISTRO CAMPO N. DE CAMPOS N. DE REGISTROS

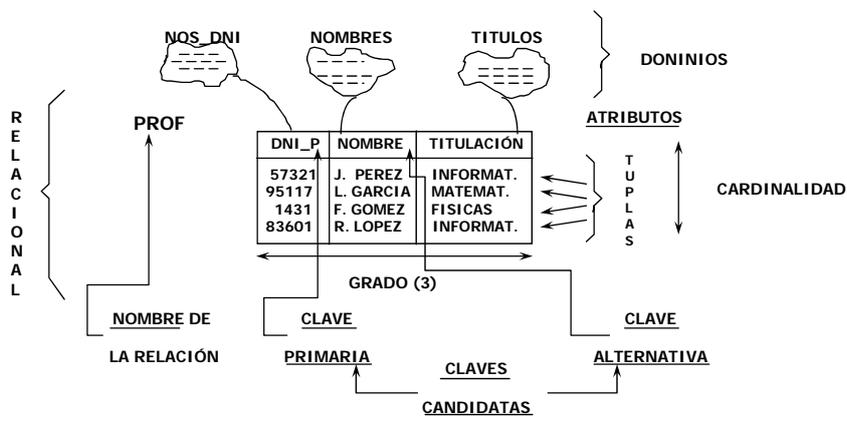
CLAVES

- Clave candidata:** atributo o conjunto de atributos mínimo que identifican unívocamente cada tupla de una relación.
  - *unívoco:* a cada valor de la clave le corresponde a lo sumo una tupla
  - *mínimo:* no existe ningún subconjunto suyo que también sea clave candidata
- Clave primaria:** clave candidata privilegiada; la que selecciona el usuario para identificar las tuplas de la relación. Se representa subrayada.
- Clave secundaria:** resto de claves candidatas. Si se usa alguna, se subraya de modo discontinuo.

**INTERRELACIONES**

- ❑ Asociaciones definidas entre los esquemas de relación. No existen vinculaciones físicas, sólo lógicas.
- ❑ Se distinguen distintos tipos según el número máximo de tuplas que intervienen por parte de cada relación. A este número se le denomina *cardinalidad de la relación*. Puede ser:
  - **1 a 1**: correspondencia biunívoca  
Un elemento de la primera relación está asociado a un único elemento de la segunda relación
  - **1 a n**: correspondencia múltiple  
Un esquema (1) es padre del otro (n). Un elemento de la primera relación se asocia a varios elementos de la segunda relación. Y, aun elemento de la segunda relación le corresponde un único elemento (máximo) de la primera.
  - **n a n**: Asociación varios a varios

**ESTRUCTURA RELACIONAL**



- Estructuras u ocurrencias no permitidas en el modelo relacional. Pueden ser de dos tipos: inherentes y semánticas (o de usuario)

**RESTRICCIONES INHERENTES**

- No hay dos tuplas iguales -> Es obligatorio que exista una clave primaria
- El orden de las tuplas no es significativo
- El orden de los atributos no es significativo
- Cada atributo sólo puede tomar un único valor en el dominio sobre el que está definido (es decir, no existen grupos repetitivos).

**En cada casilla (*cruce fila y columna*) sólo existe un valor - *no se admiten grupos repetitivos* -**

DNI_P	DNI_A	CALIF.
57321	13251	4,5
	5682	8,2
	13007	9

**NO**

### RESTRICCIÓN DE INTEGRIDAD DE ENTIDAD

**“Ningún atributo** que forme parte de la **clave primaria** de una relación puede tomar un **valor nulo**”

### RESTRICCIONES SEMÁNTICAS (O DE USUARIO)

Utilidades ofrecidas por el modelo de datos con el fin de contemplar, de forma sencilla, la semántica del mundo real.

Las restricciones semánticas del modelo relacional, expresadas en el estándar SQL92 son:

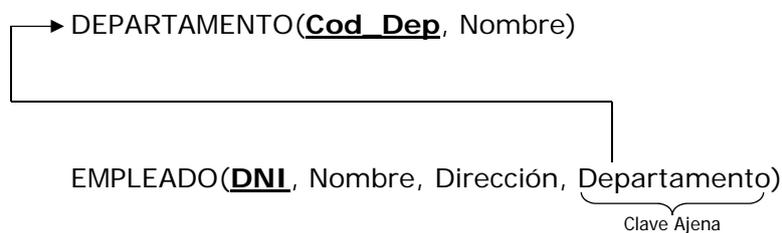
- **Clave Primaria (PRIMARY KEY)**. Permite declarar un atributo o conjunto de atributos como clave primaria de una relación. La obligatoriedad de la existencia de una clave primaria es una restricción inherente; en cambio, la elección de los atributos que la componen es una restricción semántica.

**RESTRICCIONES SEMÁNTICAS (O DE USUARIO)**

- **Unicidad (UNIQUE)**. Se emplea para indicar que los valores de un atributo o conjunto de atributos no pueden repetirse en la relación. Define una clave alternativa.
- **Obligatoriedad (NOT NULL)**. Aplicable a uno o más atributos para indicar deben tomar un valor distinto de NULO.
- **Integridad Referencial (FOREIGN KEY)**. Toda relación (*relación que referencia*) que tenga un atributo (o conjunto de atributos) asociado con una clave primaria de otra relación (*relación referenciada*), debe tomar valor de entre los definidos para la clave primaria de la relación referenciada o bien tomar valor nulo. Al descriptor de la relación que referencia se le denomina **clave ajena**.

**RESTRICCIONES SEMÁNTICAS (O DE USUARIO)**

Ejemplo:



**RESTRICCIONES SEMÁNTICAS (O DE USUARIO)**

¿Qué sucede cuando se produce una operación de borrado o modificación sobre las tuplas de la base de datos?

- ❑ Junto con las claves ajenas es necesario definir las *reglas de integridad referencial*, es decir, el comportamiento deseado cuando se produce una eliminación o modificación de tuplas relacionadas a través de una restricción de clave ajena

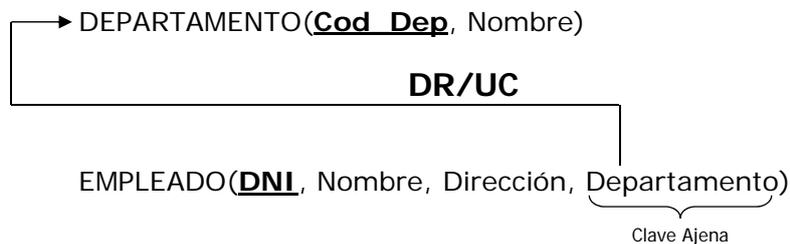
**RESTRICCIONES SEMÁNTICAS (O DE USUARIO)**

- ❑ Las reglas u opciones de integridad referencial definidas en el estándar SQL92 son:
  - **Operación restringida (NO ACTION)**. Si la operación rompe la restricción referencial, no se lleva a cabo. Se abrevia **NA o R (Restrict)**.
  - **Operación con transmisión en cascada (CASCADE)**. Los valores afectados serán actualizados también en la clave ajena. Se abrevia **C**.
  - **Operación con puesta a nulo (SET NULL)**. Los valores afectados serán sustituidos por el valor nulo. Se abrevia **SN**.
  - **Operación con puesta a valor por defecto (SET DEFAULT)**. Los valores afectados serán sustituidos por un valor por defecto. Se abrevia **SD**.

**RESTRICCIONES SEMÁNTICAS (O DE USUARIO)**

- ❑ Para toda interrelación se debe especificar que regla ha de aplicarse en cada acción que haga peligrar la integridad (*on Delete / on Update*)

Ejemplo:



**INTERRELACIONES**

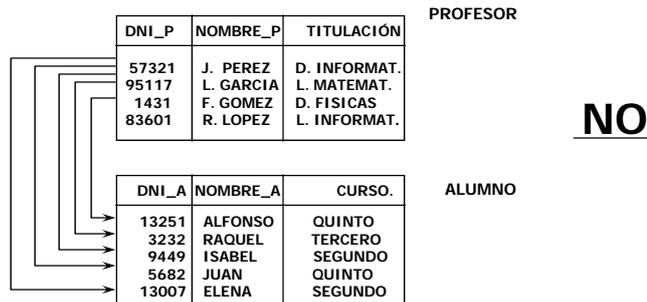
- ❑ Según la relación en la que se sitúa la clave ajena de la interrelación se representa una cardinalidad diferente. Así:
  - **1 a 1**: correspondencia biunívoca. Tres opciones:
    - Se podría fusionar ambos esquemas en uno sólo
    - La clave ajena puede localizarse en cualquiera de los dos
    - Compartir clave primaria
  - **1 a n**: correspondencia múltiple. Una opción:
 

Un esquema (1) es padre del otro (n). La clave ajena se sitúa en la relación que participa con múltiples tuplas (n).
  - **n a n**: correspondencia varios a varios. Una opción:
 

Se crea una nueva relación con una clave primaria formada por la combinación de las claves primarias de las relaciones que intervienen

**ADVERTENCIA**

No existen vinculaciones entre datos visibles por los usuarios, es decir, todos los datos (*tanto entidades como asociaciones*) están representados explícitamente por valores y nada más que valores.



**RESTRICCIONES SEMÁNTICAS (O DE USUARIO)**

- ❑ El modelo relacional permite también definir otras restricciones semánticas, dirigidas a impedir ocurrencias que no cumplan determinados predicados definidos por el usuario:
  - **Verificación (CHECK):** Comprueba si las operaciones de actualización o modificación sobre una relación cumplen una determinada condición. Si no se verifica la condición, se aborta la operación. Su definición va unida al esquema y puede tener o no asignado un nombre.
  - **Aserción (ASSERTION):** Equivalente a la verificación cuando intervienen varias relaciones en la definición de la condición. No va unida a un esquema concreto y debe estar nominada.

**RESTRICCIONES SEMÁNTICAS (O DE USUARIO)**

- **Disparadores (TRIGGERS):** Permiten al usuario indicar la acción a llevar a cabo ante una determinada condición. No están contemplados en SQL92 (sí en SQL3) pero muchos productos comerciales los incorporan

**ESQUEMA DE RELACIÓN vs. ESQUEMA RELACIONAL**

- **Esquema de Relación:** Compuesto por los atributos de la relación, los dominios sobre los que toman valor y las restricciones de integridad que afectan a cada uno de los elementos del esquema (restricciones intraelementos).

$$R \langle A: D, S \rangle \quad \left\{ \begin{array}{l} \mathbf{A: Atributos} \\ \mathbf{D: Dominios} \\ \mathbf{S: Restricciones intraelementos} \end{array} \right.$$

- **Esquema relacional:** Colección de esquemas de relación junto con las restricciones de integridad definidas sobre más de un esquema de relación (restricciones interelementos).

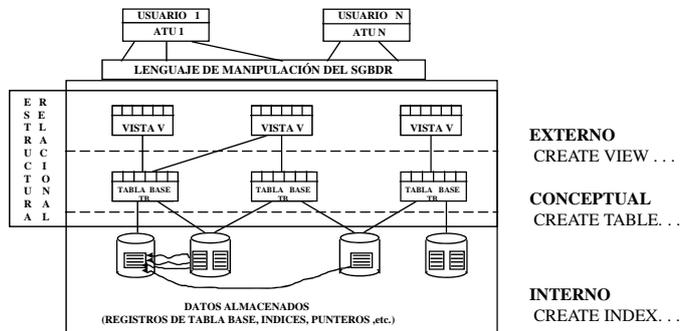
$$E \langle \{R_i\}, \{I_i\} \rangle \quad \left\{ \begin{array}{l} \mathbf{E: Nombre del esquema relacional} \\ \mathbf{\{R_i\}: Conjunto de esquemas de relación} \\ \mathbf{\{I_i\}: Restricciones interelementos} \end{array} \right.$$

**DEPENDENCIAS FUNCIONALES**

- ❑ Propiedades inherentes al contenido semántico de los datos
- ❑ Han de cumplirse para cualquier extensión de una relación
- ❑ Representan asociaciones o relaciones existentes entre los atributos de una misma relación
- ❑ Así, dada una relación R definida sobre un conjunto de atributos A, donde **X e Y son subconjuntos de A** (llamados **descriptores**), **Y depende funcionalmente de X** si, y solo si, *a todo valor de X le corresponde en todo momento un único valor de Y*, representándose:

$$X \rightarrow Y$$

**Arquitectura a 3 niveles de una BDR**



<u>ANSI</u>	<u>RELACIONAL</u>
ESQUEMA CONCEPTUAL	RELACIONES BASE <i>(Son tablas que tienen una representación directa en el almacenamiento interno)</i>
ESQUEMA EXTERNO	VISTAS <i>(Son tablas virtuales definidas sobre una o más tabla base. Son "ventanas" sobre tablas reales, de las que solo se almacena su definición, y no tienen por tanto, representación directa en el almacenamiento). También se suele permitir la visión directa de las RELACIONES BASE.</i>
ESQUEMA INTERNO	REGISTROS ALMACENADOS <i>(Corresponden a las filas de las tablas base, pero no existe correspondencia biunívoca, ya que un registro almacenado puede estar constituido por varias filas de distintas tablas, o viceversa una fila se puede dividir en varios registros)</i> INDICES, PUNTEROS, etc. <i>(necesarios por motivos de rendimiento)</i>

Para valorar qué SGBDR seguían el Modelo Relacional y cuáles no, en 1985 Codd propone 12 reglas que todos ellos deben seguir:

- 1. Representación de la Información:** representación única y a nivel lógico
- 2. Acceso garantizado:** a una relación (por su nombre), a sus tuplas (por la clave), o a sus atributos (a través de los nombres de atributo).
- 3. Tratamiento sistemático de valores nulos**
- 4. Catálogo Relacional:** la metainformación se almacena en la BD
- 5. Sublenguaje de Datos completo:** permite describir relaciones, vistas, restricciones, permisos (acceso), manejar datos, y gestionar transacciones.

## Modelo Relacional. Estática

## 8. Las 12 Normas de Codd

6. **Actualización de Vistas:** toda vista actualizable de poder ser actualizada
7. **Actualizaciones y Recuperaciones de alto nivel:** a nivel de conjuntos
8. **Independencia física de los datos:** datos independientes de los caminos físicos
9. **Independencia lógica de los datos:** datos independientes de los procesos
10. **Independencia de la integridad:** la semántica no depende del SGBD
11. **Independencia de la distribución:** soportará BD distribuidas
12. **Regla de No Subversión:** si el SGBD soporta un lenguaje de bajo nivel, este no podrá usarse nunca para ignorar las reglas de integridad

TEMA III. PARTE II

FBD3 III.33

© Grupo de Bases de Datos Avanzadas – Univ. Carlos III de Madrid

## Modelo Relacional. Estática

## Bibliografía

1. Miguel, A. De, Piattini, M. **Fundamentos y modelos de Bases de Datos**, Ed. Rama 1999
2. Miguel, A. De, Piattini, M. y Marcos, E. **Diseño de Bases de Datos Relacionales**, Ed. Rama 1999
3. Miguel, A. De, Martínez, P., Castro, E., Caveró, J.M., Cuadra, D., Iglesias, A.M. y Nieto, C. **Diseño de Bases de Datos. Problemas Resueltos**, Ed. Rama, 2001
4. Oszu, M.T. y Valduriez, P., **Principles of Distributed database systems**, 2ª Edición, Prentice Hall, 1999

Tema I

FBD3 III.34

© Grupo de Bases de Datos Avanzadas – Univ. Carlos III de Madrid