

TEMA 1. BASES DE DATOS RELACIONALES

1. El modelo relacional
2. Lenguaje SQL

1. El modelo relacional

Se debe a E.F. Codd y data de 1970.

Basado en dos ramas de las **matemáticas**:

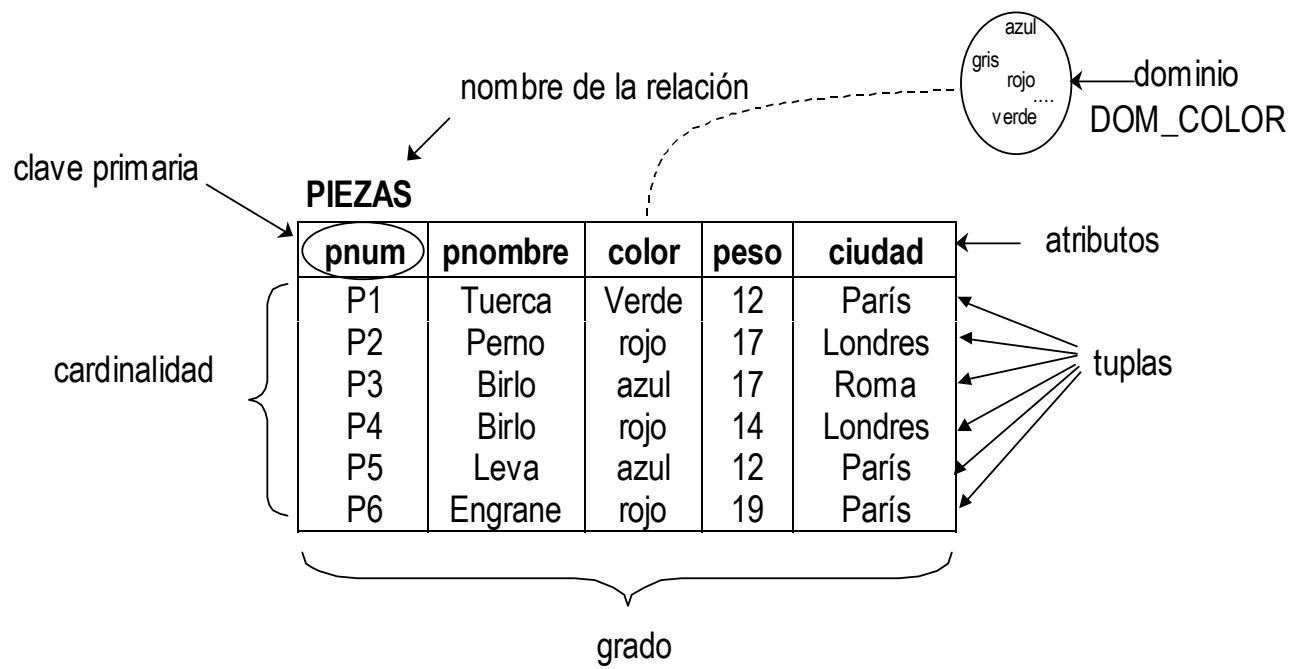
- Teoría de conjuntos y
- Lógica de predicados de primer orden.

□□ → Predecible, fiable y seguro ¿qué más se puede pedir?

Todo modelo de datos tiene que ver con tres aspectos de los datos:

- *Estructura de datos.*
- *Integridad de datos.*
- *Manejo de datos.*

Estructura de Datos Relacional



PIEZAS(pnum, pnombre, color, peso, ciudad)

Base de datos de proveedores y piezas

s (suministradores, proveedores)

snum	snombre	estado	ciudad
S1	Salazar	20	Londres
S2	Jaimes	10	París
S3	Bernal	30	París
S4	Corona	20	Londres
S5	Aldana	30	Atenas

p (piezas)

pnum	pnombre	color	peso	ciudad
P1	Tuerca	verde	12	París
P2	Perno	rojo	17	Londres
P3	Birlo	azul	17	Roma
P4	Birlo	rojo	14	Londres
P5	leva	azul	12	París
P6	Engrane	rojo	19	París

sp (envíos)

snum	pnum	cant
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

S

snum
snombre
estado
ciudad

SP

snum
pnum
cant

P

pnum
pnombre
color
peso
ciudad

proveedor que
envía la pieza

pieza que se envía

Propiedades de las Relaciones

- Cada **relación** tiene un **nombre distinto**.
- Los valores de los atributos son **atómicos**: las relaciones están normalizadas (1FN).

Antes (grado 2)

snum	detalle		
	línea	pnum	cant
S2	1	P1	300
	2	P3	200
	3	P4	400
S4	1	P2	300
	2	P4	400
S5	1	P6	200

Relación no normalizada
(con grupos repetitivos)

Después (grado 3)

snum	línea	pnum	cant
S2	1	P1	300
S2	2	P3	200
S2	3	P4	400
S4	1	P2	300
S4	2	P4	400
S5	1	P6	200

Relación normalizada
(1ª forma normal)

los valores del atributo 'detalle'
no son atómicos, son relaciones

- En una relación cada **atributo** tiene un **nombre distinto**.
- El orden de los atributos no importa: los **atributos no están ordenados**.
- Cada tupla es distinta de las demás: **no hay tuplas duplicadas**.
- El orden de las tuplas no importa: **las tuplas no están ordenadas**.

Tipos de relaciones

- **Relaciones base:**

```
CREATE TABLE piezas
( pnum    VARCHAR2(2),
  pnombre VARCHAR2(20), ...,
  CONSTRAINT cp_piezas PRIMARY KEY (pnum) );
```

- **Vistas:**

```
CREATE VIEW piezas_azules AS
SELECT pnum, pnombre, peso, ciudad
FROM   piezas
WHERE  color = 'azul';
```

- **Instantáneas:**

```
CREATE SNAPSHOT piezas_bcn
STORAGE INITIAL 50K NEXT 50K
REFRESH FAST NEXT NEXT_DAY(TRUNC(SYSDATE), 'MONDAY')
AS
SELECT * FROM bcn.piezas;
```

- **Resultados de consultas, resultados intermedios, resultados temporales**

Claves

➤ **Clave única**

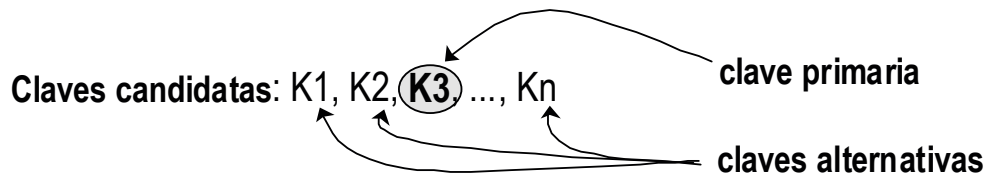
➤ **Clave candidata**

$K \in R$ es una clave candidata para R si, y sólo si, satisface:

① **Unicidad**

② **Irreducibilidad (minimalidad)**

➤ **Clave primaria**



Importancia de las claves primarias:

mecanismo de direccionamiento
de tuplas de un sistema relacional.

Reglas de integridad

Regla de integridad :

Restricción sobre una base de datos a cumplir en todos sus estados.

Tipos de reglas de integridad:

Reglas de integridad específicas : **reglas de negocio**.

Reglas de integridad generales :

➤ **Regla de integridad de entidades**

➤ **Regla de integridad referencial**

➤ **Restricciones de dominios**

Nulo : no es un valor, implica ausencia de información.

Regla de integridad de entidades

"Ninguno de los atributos que componen la clave primaria puede ser nulo."

¡Una base de datos relacional nunca almacena información de algo que no se puede identificar!

Observaciones:

- La regla se aplica a las relaciones base.
- La regla se aplica sólo a la clave primaria.

Claves

➤ **Claves ajenas**  relaciones entre datos
Valor: referencia a tupla  tupla referenciada

Diagramas referenciales:

FACTURAS  CLIENTES Cliente al que corresponde la factura
 codcli

Observaciones

- Clave ajena y clave primaria : **mismo dominio**.
- $R_n \rightarrow \dots \rightarrow R_2 \rightarrow R_1$: **camino referencial**.
- $R_1 \rightarrow R_1$: **auto-referencia**.
- $R_n \rightarrow \dots \rightarrow R_2 \rightarrow R_1 \rightarrow R_n$: **ciclo referencial**.
- En ocasiones pueden admitir **nulos**.
- Son el “**pegamento**” que mantiene unida la base de datos.

Regla de integridad referencial

"Si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser todos nulos."

Se enmarca en términos de **estados** de la base de datos:

indica lo que es un estado ilegal, pero no cómo se puede evitar.

¿Qué hacer si en un estado legal, una operación conduce a un estado ilegal?

- **Rechazar** la operación.
- **Aceptar** la operación y realizar **operaciones adicionales** compensatorias que conduzcan a un estado legal.

Reglas para las claves ajenas

Regla de los nulos

¿Tiene sentido que la clave ajena acepte nulos?

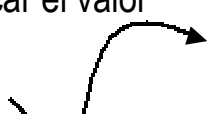
Regla de borrado:

¿Qué hacer si se intenta borrar la tupla referenciada por la clave ajena?

- 
- Restringir
 - Propagar
 - Anular

Regla de modificación

¿Qué hacer si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave ajena?

- 
- Restringir
 - Propagar
 - Anular

Ejemplos

			<u>NULOS</u>	<u>BORRADO</u>	<u>MODIFICACION</u>
PUEBLOS	codpro →	PROVINCIAS	No	Propagar	Propagar
CLIENTES	codpue →	PUEBLOS	No	Restringir	Propagar
FACTURAS	codcli →	CLIENTES	Sí	Anular	Propagar
LINEAS_FAC	codfac →	FACTURAS	No	Propagar	Propagar
LINEAS_FAC	codart →	ARTICULOS	Sí	Anular	Propagar

Pregunta: ¿se puede borrar la provincia de Castellón ?

PIEZA(codpieza,descrip)
COMPONENTE(codpieza,codcomponente,cant)

			<u>NULOS</u>	<u>BORRADO</u>	<u>MODIFICACION</u>
COMPONENTE	codpieza →	PIEZA			
COMPONENTE	codcomponente →	PIEZA			

Lenguajes Relacionales

Álgebra relacional → procedural

Cálculo relacional → no procedural

Ambos lenguajes son **equivalentes**.

Lenguaje **relacionalmente completo** → SQL