TEMA II

El Modelo Relacional de Datos

El Modelo Relacional de Datos

Objetivos:

- conocer las estructuras de datos del modelo: la tupla y la relación, así como sus operadores asociados,
- conocer básicamente la forma de modelizar la realidad utilizando el modelo relacional,
- conocer los mecanismos del modelo relacional para expresar restricciones de integridad: definición de dominios y definición de claves, y
- conocer los lenguajes de manipulación propuestos para este modelo de datos: álgebra relacional y cálculo relacional de tuplas

El Modelo Relacional de Datos

- Temario:

2.1 Modelo relacional de datos (aproximación algebraica).

2.1.1 Estructuras: tupla y relación

2.1.2 Operadores asociados a la estructura relación: Álgebra Relacional.

2.2 Esquema relacional: representación de la realidad

2.3 Modelo relacional de datos (aproximación lógica).

2.3.1 La lógica de 1er orden

2.3.2 Interpretación lógica de una base de datos relacional

El Modelo Relacional de Datos

- Temario (cont.):

2.4 Restricciones de integridad

2.4.1 Restricciones sobre atributos: de *dominio* y de *valor no nulo*.

2.4.2 Restricciones de unicidad

2.4.3 Concepto de clave primaria. Integridad de clave primaria

2.4.4 Concepto de clave ajena. Integridad referencial

2.4.5 Restauración de la integridad referencial: directrices al SGBD

2.4.6 Otros mecanismos para representar restricciones de integridad

El Modelo Relacional de Datos

- <u>Temario (cont.):</u>

2.5 El lenguaje estándar SQL.

2.5.1 SQL como lenguaje de definición de datos (DDL).

2.5.2 SQL como lenguaje de manipulación de datos.

2.5.2.1 INSERT, DELETE Y UPDATE.

2.5.2.2 Aproximación lógica de la cláusula SELECT.

2.5.2.3 Aproximación algebraica de la cláusula SELECT.

2.6 Información derivada: vistas.

2.6.1 Concepto de vista.

2.6.2 Aplicaciones de las vistas.

2.6.3 Vistas en SQL

El Modelo Relacional de Datos

- <u>Temario (cont.):</u>

2.7 Mecanismos de actividad: disparadores.

2.7.1 Concepto de disparador.

2.7.2 Reglas Evento-Condición-Acción.

2.7.3 Aplicaciones de los disparadores

2.7.4 Disparadores en SQL

2.8 Limitaciones del modelo relacional de datos

2.- Introducción al Modelo Relacional de Datos

Reseñas históricas del Modelo Relacional de Datos (MRD):

70's: Propuesto por E. Codd en 1970

80's: Se populariza en la práctica (Oracle, ...). ANSI define el estándar SQL.

90's: Generalización y estandarización (SQL'92) y extensiones

Razones del éxito:

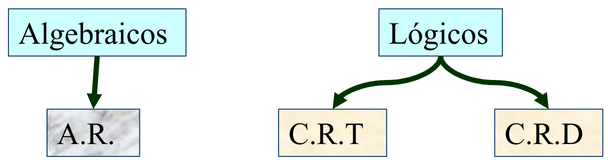
Sencillez: una base de datos es "un conjunto de tablas".

2.- El MRD: Componentes y Perspectivas

MRD = Estructuras de datos + Operadores asociados

Estructuras de datos comunes:
- *dominios*
- *atributos*
- la *tupla*
- la *relación*.

Dos Grandes Familias de Operadores:



2.- El MRD: Terminología

Estructuras de datos:

Terminología Corriente (informática)		Terminología MRD (matemática)
• *tipos de datos*	⟷	• *dominios*
• *campos / columnas*	⟷	• *atributos*
• *registro / filas*	⟷	• *tupla*
• *tabla*	⟷	• *relación*

No son *exactamente* equivalentes

2.1.1.- Concepto de tupla

Esquema de tupla:

Un esquema de tupla, τ, es un conjunto de pares de la forma:

$$\tau = \{(A_1, D_1), (A_2, D_2), \ldots, (A_n, D_n)\}$$

donde:

$\{A_1, A_2, \ldots, A_n\}$ $(n > 0)$ es el conjunto de nombres de **atributos** del esquema, necesariamente distintos.

$D_1, D_2 \ldots, D_n$ son los **dominios** asociados a dichos atributos que no tienen que ser necesariamente distintos.

2.1.1.- Concepto de tupla

Ejemplo de ***Esquema de tupla***:

Persona = {(dni, entero), (nombre, cadena), (dirección, cadena)}

donde:

{ dni, nombre, dirección } es el conjunto de nombres de atributos del esquema.

entero, cadena, cadena son los dominios asociados a dichos atributos.

2.1.1.- Concepto de tupla

Tupla:

Una tupla, *t*, de esquema de tupla τ donde

$$\tau = \{(A_1, D_1), (A_2, D_2), \ldots, (A_n, D_n)\}$$

es un conjunto de pares de la forma siguiente:

$$t = \{(A_1, v_1), (A_2, v_2), \ldots, (A_n, v_n)\}$$

tal que $\forall i \; v_i \in D_i$.

2.1.1.- Concepto de tupla

Ejemplo de ***Tupla***:

Del esquema de tupla siguiente:

Persona = {(dni, entero), (nombre, cadena), (dirección, cadena)}

Tenemos

t_1 = {(dni, 25442525), (nombre, “Joan Roig”), (dirección, “Sueca 15”)}

t_2 = {(dni, “28442879F”), (nombre, “R3PO”), (dirección, “46022”)}

t_3 = { (nombre, “Pep Blau”), (dni, 9525869), (dirección, “No se sap”)}

2.1.1.- Dominios

PROBLEMA: ¿Y si no se sabe el valor que toma una tupla en alguno de los atributos del esquema?

Solución en Lenguajes de Programación: uso de valores significativos o extremos (-1, “Vacío”, “ ”, “No se sabe”, 0, “Sin dirección”, “---”, ...)

Solución en el Modelo Relacional: **VALOR NULO (?)**

Un **Dominio** es algo más que un tipo de datos:

Un dominio es un conjunto de elementos que siempre incluye además el valor NULO.

2.1.1.- Operadores de Tupla

Dada una tupla: $t = \{(A_1, v_1), \ldots, (A_i, v_i), \ldots, (A_n, v_n)\}$

CONSULTAR:

- Consultar(t, A_i) = v_i

ASIGNAR:

- Asignar(t, A_i, w_i) = $\{(A_1, v_1), \ldots, (A_i, \mathbf{w_i}), \ldots, (A_n, v_n)\}$

Notación usual para los operadores

- Consultar(t, A_i): $\boldsymbol{t.A_i}$ $\boldsymbol{t(A_i)}$
- Asignar(t, A_i, w_i): $\boldsymbol{t.A_i \leftarrow w_i}$ $\boldsymbol{t(A_i) \leftarrow w_i}$

2.1.1.- Ejemplo

Dados los dominios: dom_dni: entero
dom_nom, dom_dir: cadena(20)

Esquema de tupla:

Persona = {(dni, dom_dni), (nombre, dom_nom), (dirección, dom_dir)}

Tuplas:

t_1 = {(dni, 12345678), (nombre, “Pepa Gómez”), (dirección, “Paz 10”)}

t_2 = { (nombre, “Pep Blau”), (dni, 9525869), (dirección, ?) }

Operaciones:

Consultar (t_1, nombre) = “Pepa Gómez”

Asignar (t_1, dirección, “Colón 15”) = {(dni, 12.345.678), (nombre, “Pepa Gómez”), (dirección, “Colón 15”)}

Consultar (t_2, dirección) = ?

***Diremos* que t_2.dirección es nulo, no que t_2.dirección = nulo.**

2.1.1.- Concepto de relación (algebraico)

Relación:

Una relación es un conjunto de tuplas del mismo esquema.

Esquema de relación

El esquema de una relación es el esquema de las tuplas que la forman.

Notación

$$R(A_1: D_1, A_2: D_2, \ldots, A_n: D_n)$$

define una relación *R* de esquema

$$\{(A_1, D_1), (A_2, D_2), \ldots, (A_n, D_n)\}$$

2.1.1.- Propiedades de una Relación

Propiedades de una relación

- *Grado de una relación:* número de atributos de su esquema
- *Cardinalidad de una relación:* número de tuplas que la forman.
- *Compatibilidad:* dos relaciones *R* y *S* son compatibles si sus esquemas son idénticos.

2.1.1.- Ejemplo de relación

Ejemplo:

Una relación del esquema *PERSONA* podría ser la siguiente:

{{(dni, 12.345.678), (nombre, “Pepa Gómez”), (dirección, “Colón 15”)},
{ (dni, 20.450.120), (nombre, “Juan Pérez”), (dirección, “Cuenca 20”) },
{ (nombre, “José Abad”), (dni, 12.904.569), (dirección, “Blasco Ibáñez 35)},
{ (nombre, “María Gutiérrez”), (dni, 35.784.843) (dirección, “Reina 7”)}}

Grado:

Cardinalidad:

Compatible con:

2.1.1.- Representación de relación

Representación de una relación → TABLA

- las tuplas se representan por filas
- las atributos dan nombre a la cabecera de las columnas

Ejemplo: Relación PERSONA

Columna ≈ Atributo

Fila ≈ Tupla

dni	Nombre	dirección
20.450.120	Juan Pérez	Cuenca 20
12.904.569	José Abad	Blasco Ibáñez 35
35.784.843	María Gutiérrez	Reina 7
12.345.678	Pepa Gómez	Colón 15

2.1.1.- Diferencia Relación - Tabla

La Tabla es sólo una Representación Matricial de una Relación

CARACTERÍSTICAS DIFERENCIADORAS DE RELACIÓN:
(Derivadas de la definición de relación como conjunto de conjuntos)

- No pueden existir tuplas repetidas en una relación (una relación es un conjunto).
- No existe un orden arriba-abajo entre las tuplas de una relación por el mismo motivo (una relación es un conjunto).
- No existe un orden izquierda-derecha de los atributos de una relación (una tupla es un conjunto). Se utiliza el nombre del atributo para seleccionar.

2.1.1.- Diferencia Extensión - Esquema

EXTENSIÓN (datos)	**ESQUEMA**
Tupla ←	*Esquema de tupla = Def. de relación*
(Extensión de la) relación : conjunto de tuplas de la relación ←	
Base de datos: conjunto de relaciones ←	*Esquema Relacional* : conjunto de *definiciones de relación* que representa un sistema de información

*OJO: en los SGBD se conoce a una tabla como la definición de la relación y no su contenido (cambia con el tiempo) aplicando **operadores**.*

2.1.1.- Operadores de la Estructura Relación

Operadores asociados a la estructura relación:

- INSERCIÓN
- BORRADO
- SELECCIÓN
- PROYECCIÓN
- UNIÓN
- INTERSECCIÓN
- DIFERENCIA
- PRODUCTO CARTESIANO
- CONCATENACIÓN

(SELECCIÓN … CONCATENACIÓN) También A.R.

2.1.1.- Inserción

$\boxed{\text{Insertar}(R, t) = R \cup \{ t \}}$ R y t han de tener el mismo esquema

Ejemplo:

Insertar({ {(dni, 12.345.678), (nombre, "Pepa Gómez"), (dirección, "Colón 15")},
{ (dni, 20.450.120), (nombre, "Juan Pérez"), (dirección, "Cuenca 20") },
{ (nombre, "María Gutiérrez"), (dni, 35.784.843) (dirección, "Reina 7")} },

{ (nombre, "José Abad"), (dni, 12.904.569), (dirección, "Blasco Ibáñez 35)})

=

{ {(dni, 12.345.678), (nombre, "Pepa Gómez"), (dirección, "Colón 15")},
{(dni, 20.450.120), (nombre, "Juan Pérez"), (dirección, "Cuenca 20") },
{ (dni, 12.904.569), (nombre, "José Abad"), (dirección, "Blasco Ibáñez 35)},
{(nombre, "María Gutiérrez"), (dni, 35.784.843) (dirección, "Reina 7")}}

Cuestión: Cómo afecta la inserción a:
El grado:
La cardinalidad:

2.1.1.- Borrado

$Borrar(R, t) = R - \{ t \}$ *R* y *t* han de tener el mismo esquema

Ejemplo:

Borrar({ {(dni, 12.345.678), (nombre, "Pepa Gómez"), (dirección, "Colón 15")},
{(dni, 20.450.120), (nombre, "Juan Pérez"), (dirección, "Cuenca 20") },
{ (dni, 12.904.569), (nombre, "José Abad"), (dirección, "Blasco Ibáñez 35)},
{(nombre, "María Gutiérrez"), (dni, 35.784.843) (dirección, "Reina 7")}}

{ (nombre, "José Abad"), (dni, 12.904.569), (dirección, "Blasco Ibáñez 35)})

=

{ {(dni, 12.345.678), (nombre, "Pepa Gómez"), (dirección, "Colón 15")},
{ (dni, 20.450.120), (nombre, "Juan Pérez"), (dirección, "Cuenca 20") },
{ (nombre, "María Gutiérrez"), (dni, 35.784.843) (dirección, "Reina 7")} },

Cuestión: Cómo afecta el borrado a:
El grado:
La cardinalidad:

2.1.2.- Álgebra Relacional (A.R.)

A.R.: Conjunto de operadores unarios o binarios que actúan sobre relaciones

Son operadores *cerrados*: el resultado de aplicar cualquier operador del A.R. sobre una o dos relaciones es una relación.

- *operadores conjuntistas*:
  - unión,
  - intersección,
  - diferencia, y
  - producto cartesiano
- *operadores propiamente relacionales*:
  - selección,
  - proyección,
  - división, y
  - concatenación.
- *operador especial*: renombrar

2.1.2.- A.R. (Operador Renombrar)

$R((A_i, B_i),\ldots, (A_j, B_j))$

Sea *R* una relación de esquema $\{(A_1, D_1), (A_2, D_2),\ldots, (A_n, D_n)\}$. *Renombrar en R los atributos* $A_i,\ldots, A_j$ *por* $B_i,\ldots, Bj$, denotado de la forma $R((A_i, B_i),\ldots, (A_j, B_j))$, produce una relación que contiene cada una de las tuplas de *R*, cambiando adecuadamente los nombres de atributo.

$$R((A_i, B_i),\ldots, (A_j, B_j)) = \{\{(A_1, v_1),\ldots, (B_i, v_i),\ldots, (B_j, v_j),\ldots, (A_n, v_n)\} \mid \{(A_1, v_1),\ldots, (A_i, v_i),\ldots, (A_j, v_j),\ldots, (A_n, v_n)\} \in R\}$$

El esquema de la relación resultado es el siguiente:

$$\{(A_1, D_1),\ldots, (B_i, D_i),\ldots, (B_j, D_j),\ldots, (A_n, D_n)\}.$$

2.1.2.- A.R. (Operador Renombrar)

Ejemplo:

Sea el siguiente esquema de una base de datos relacional:

Río (rcod: dom_rcod, nombre: dom_nom)
Otros_Ríos (rcod: dom_rcod, nombre: dom_nom)
Provincia (pcod: dom_pcod, nombre: dom_nom)
Pasa_por (pcod: dom_pcod, rcod: dom_rcod)

Cuestión:

¿Cómo renombraríamos la relación *Pasa_por* para que el atributo *pcod* pase a llamarse *codProv* y *rcod* pase a llamarse *codRio*?

2.1.2.- A.R. (Operador Renombrar)

El operador renombrar se aplica sobre relaciones.
NO sobre esquemas de relaciones

Ejemplo:

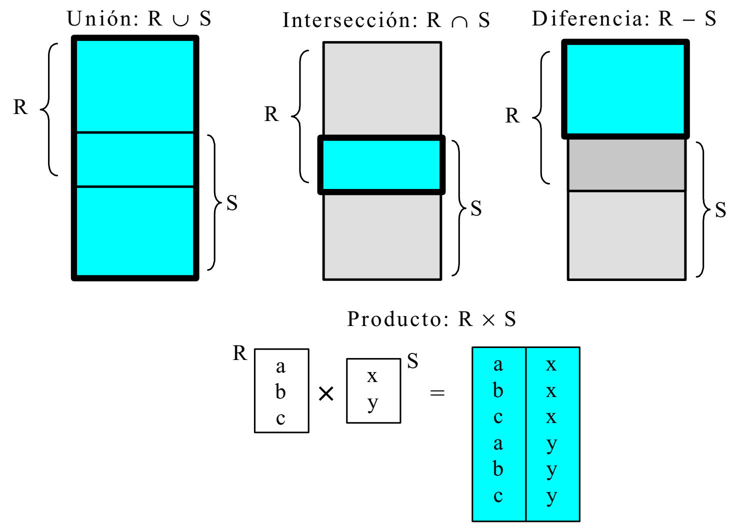
Sea *Pasa_por* una relación representada por la siguiente tabla:

Pasa_por	
pcod	**rcod**
44	r2
46	r2
45	r1
28	r1
16	r1

Pasa_por((**pcod**, **codProv**), (**rcod**, **codRio**)) =

Pasa_por	
codProv	**codRio**
44	r2
46	r2
45	r1
28	r1
16	r1

2.1.2.- A.R. (Operadores Conjuntistas)



2.1.2.- A.R. (Operador Unión)

$$\boldsymbol{R \cup S}$$

R y *S* han de tener el mismo esquema

Sean *R* y *S* dos relaciones compatibles cuyo esquema es $\{(A_1, D_1),\ldots, (A_n, D_n)\}$. *La unión de R y* S, denotada como $R \cup S$, es una relación cuyo esquema es el mismo que el de *R* y *S*, y que está formada por todas las tuplas pertenecientes a *R*, a *S*, o a ambas relaciones.

$$R \cup S = \{\, t \mid t \in R \ \vee \ t \in S \}$$

La unión es asociativa y conmutativa

2.1.2.- A.R. (Operador Unión)

Ejemplo:

pcod	**Nombre**
44	Teruel
46	Valencia
16	Cuenca
12	Castellón

∪

pcod	**Nombre**
16	Cuenca
45	Toledo
28	Madrid
12	Castelló

=

pcod	**Nombre**
44	Teruel
46	Valencia
16	Cuenca
12	Castellón
45	Toledo
28	Madrid
12	Castelló

2.1.2.- A.R. (Operador Diferencia)

R – S

R y *S* han de tener el mismo esquema

Sean *R* y *S* dos relaciones compatibles cuyo esquema es $\{(A_1, D_1),\ldots,(A_n, D_n)\}$. *La diferencia entre R y S*, denotada como $R - S$, es una relación cuyo esquema es el de *R* y *S*, y que está formada por todas las tuplas que pertenecen a *R* y no pertenecen a *S*.

$$R - S = \{\, t \mid t \in R \wedge t \notin S \}$$

La diferencia no es ni asociativa ni conmutativa.

2.1.2.- A.R. (Operador Diferencia)

Ejemplo:

pcod	Nombre
44	Teruel
46	Valencia
16	Cuenca
12	Castellón
45	Toledo
28	Madrid
12	Castelló

–

pcod	Nombre
16	Cuenca
45	Toledo
28	Madrid
12	Castelló

=

pcod	Nombre
44	Teruel
46	Valencia
12	Castellón

2.1.2.- A.R. (Operador Intersección)

R ∩ S

R y *S* han de tener el mismo esquema

Sean *R* y *S* dos relaciones compatibles cuyo esquema es $\{(A_1, D_1),\ldots,(A_n, D_n)\}$. *La intersección de R y* S, denotada como $R \cap S$, es una relación cuyo esquema es el mismo que el de *R* y *S*, y que está formada por todas las tuplas pertenecientes a *R* y a *S*.

$$R \cap S = \{\, t \mid t \in R \wedge t \in S \}$$

La intersección es asociativa y conmutativa.

2.1.2.- A.R. (Operador Intersección)

Ejemplo:

pcod	Nombre
44	Teruel
46	Valencia
16	Cuenca
12	Castellón

∩

pcod	Nombre
16	Cuenca
45	Toledo
28	Madrid
12	Castelló

=

pcod	Nombre
16	Cuenca

2.1.2.- A.R. (Operador Producto Cartesiano)

$\boldsymbol{R \times S}$

R y *S* no pueden tener nombres de atributo en común

Sean *R* y *S* dos relaciones cuyos respectivos esquemas $\{(A_1, D_1), \ldots, (A_n, D_n)\}$ y $\{(B_1, E_1), \ldots, (B_m, E_m)\}$ cumplen que no tienen ningún nombre de atributo en común. *El producto cartesiano de R y S*, denotado como $R \times S$, es una relación cuyo esquema es la unión de los esquemas de *R* y *S*, y que está formada por todas las tuplas que se pueden construir uniendo una de *R* y una de *S*.

$$R \times S = \{ \{(A_1, v_1), \ldots, (A_n, v_n), (B_1, w_1), \ldots, (B_m, w_m)\} \mid \{(A_1, v_1), \ldots, (A_n, v_n)\} \in R \text{ y } \{(B_1, w_1), \ldots, (B_m, w_m)\} \in S\}$$

El esquema de la relación resultante de $R \times S$ es $\{(A_1, D_1), \ldots, (A_n, D_n), (B_1, E_1), \ldots, (B_m, E_m)\}$. El producto cartesiano cumple las propiedades asociativa y conmutativa.

2.1.2.- A.R. (Operador Producto Cartesiano)

Ejemplo:

pcod	nomprov
44	Teruel
46	Valencia
16	Cuenca
12	Castellón

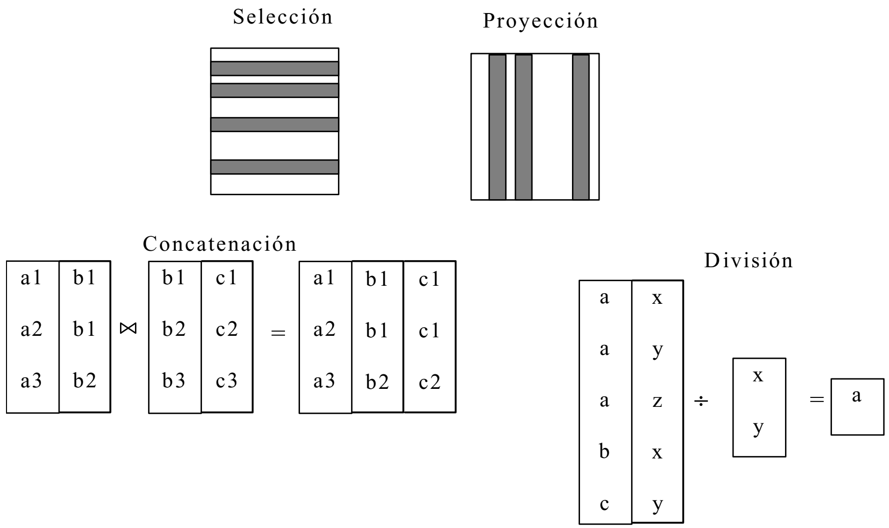
×

rcod	nombre
r1	Sénia
r2	Túria
r3	Xúquer

=

pcod	nomprov	rcod	nombre
44	Teruel	r1	Sénia
44	Teruel	r2	Túria
44	Teruel	r3	Xúquer
46	Valencia	r1	Sénia
46	Valencia	r2	Túria
46	Valencia	r3	Xúquer
16	Cuenca	r1	Sénia
16	Cuenca	r2	Túria
16	Cuenca	r3	Xúquer
12	Castellón	r1	Sénia
12	Castellón	r2	Túria
12	Castellón	r3	Xúquer

2.1.2.- A.R. (Operadores Relacionales)



2.1.2.- A.R. (Operador Proyección)

$\boldsymbol{R[A_i, A_j, \ldots, A_k]}$

Sea *R* una relación cuyo esquema es $\{(A_1, D_1), \ldots, (A_n, D_n)\}$ y sea $\{A_i, A_j, \ldots, A_k\}$ un subconjunto de los nombres de atributo de *R* con *m* elementos ($1 \leq m \leq n$). *La proyección de R sobre* $\{A_i, A_j, \ldots, A_k\}$, denotada como $R[A_i, A_j, \ldots, A_k]$, es una relación que se define como sigue:

$$R[A_i, A_j, \ldots, A_k] = \{\{(A_i, v_i), (A_j, v_j), \ldots, (A_k, v_k)\} \mid \exists t \in R \text{ tal que } \{(A_i, v_i), (A_j, v_j), \ldots, (A_k, v_k)\} \subseteq t\}$$

El esquema de relación de $R[A_i, A_j, \ldots, A_k]$ es

$\{(A_i, D_i), (A_j, D_j), \ldots, (A_k, D_k)\}$.

2.1.2.- A.R. (Operador Proyección)

Ejemplo:

Sea R =

dni	Nombre	dirección
20.450.120	Juan Pérez	Cuenca 20
12.904.569	José Abad	Blasco Ibáñez 35
35.784.843	María Gutiérrez	Reina 7
12.345.678	Pepa Gómez	Colón 15

R[dni, dirección] =

dni	dirección
20.450.120	Cuenca 20
12.904.569	Blasco Ibáñez 35
35.784.843	Reina 7
12.345.678	Colón 15

2.1.2.- A.R. (Operador Concatenación)

$\boldsymbol{R \triangleright\triangleleft S}$

Sean R y S dos relaciones cuyos esquemas son $\{(A_1, D_1),\ldots, (A_n, D_n), (B_1, E_1),\ldots,(B_m,E_m)\}$ y $\{(B_1,E_1),\ldots,(B_m,E_m), (C_1,F_1),\ldots,(C_p,F_p)\}$, resp., de forma que $B_1,\ldots, B_m$ son los atributos comunes de los dos esquemas. *La concatenación de R y S*, denotada como $R \triangleright\triangleleft S$, es una relación que contiene todas las tuplas que se pueden construir combinando una tupla de R con otra de S que tengan para cada nombre de atributo común el mismo valor asociado.

$$R \triangleright\triangleleft S=\{\{(A_1, v_1),\ldots,(A_n, v_n),(B_1, w_1),\ldots,(B_m, w_m), (C_1, y_1),\ldots,(C_p, y_p)\} \mid \{(A_1, v_1),\ldots, (A_n, v_n), (B_1, w_1),\ldots, (B_m, w_m)\} \in R \wedge \{(B_1, w_1),\ldots, (B_m, w_m), (C_1, y_1),\ldots, (C_p, y_p)\} \in S \}$$

La concatenación es asociativa y conmutativa. El esquema de la relación resultado de la concatenación es $\{(A_1, D_1),\ldots, (A_n, D_n), (B_1, E_1),\ldots, (B_m, E_m), (C_1, F_1),\ldots, (C_p, F_p)\}$.

2.1.2.- A.R. (Operador Concatenación)

Ejemplo:

pcod	nombre
44	Teruel
46	Valencia
16	Cuenca
12	Castellón

$\bowtie$

pcod	rcod
44	r1
46	r2
30	r2
20	r1
44	r3
12	r1

=

pcod	nombre	rcod
44	Teruel	r1
46	Valencia	r2
44	Teruel	r3
12	Castellón	r1

2.1.2.- A.R. (Operador Concatenación)

Más ejemplos:

pcod	Nom
44	Terol
46	València
16	Conca
12	Castelló

$\bowtie$

pcod	rcod
43	r1
50	r2
30	r2

=

pcod	Nom	rcod

pcod	Nom
44	Terol
46	València
16	Conca

$\bowtie$

scod	rcod
44	r1
50	r2

=

pcod	Nom	scod	rcod
44	Terol	44	r1
44	Terol	50	r2
46	València	44	r1
46	València	50	r2
16	Conca	44	r1
16	Conca	50	r2

2.1.2.- A.R. (Operador División)

R ÷ S

Sean R y S dos relaciones cuyos esquemas son $\{(A_1, D_1),\ldots,(A_n, D_n), (B_1, E_1),\ldots, (B_m, E_m)\}$ y $\{(B_1, E_1),\ldots, (B_m, E_m)\}$ respectivamente. *La división de R entre S,* denotada como $R \div S$, es una relación que se define como sigue:

$$
\begin{aligned}
R \div S = \{ & \{(A_1, v_1),\ldots, (A_n, v_n)\} \mid \\
& \forall s \in S\ (s = \{(B_1, w_1),\ldots,(B_m, w_m)\} \rightarrow \\
& \exists\, t \in R \text{ y } t = \{(A_1, v_1),\ldots,(A_n, v_n), (B_1, w_1),\ldots,(B_m, w_m)\}) \}
\end{aligned}
$$

El esquema de $R \div S$ es $\{(A_1, D_1),\ldots, (A_n, D_n)\}$. La división no es asociativa ni conmutativa.

2.1.2.- A.R. (Operador Selección)

R* DONDE *F

Sea R una relación de esquema $\{(A_1, D_1),\ldots, (A_n, D_n)\}$. *La selección en R respecto a la condición F*, denotada como *R donde F*, es una relación del mismo esquema que R y que está formada por todas las tuplas de R que cumplen la condición F.

$$\text{R DONDE F} = \{ t \mid t \in R \text{ y } F(t) \text{ se evalúa al valor } \textit{cierto} \}$$

¿Cómo es la condición $F(t)$?

¿Cómo se evalúa $F(t)$?

2.1.2.- A.R. (Operador Selección)

¿Cómo es la condición *F*?

Tipos de <u>Comparación</u>:

- Nulo(A_i)
- $A_i \,\alpha\, A_j$
- $A_i \,\alpha\, a$

donde α es un operador de comparación ($<, >, \leq, \geq, =, \neq$), A_i y A_j son nombres de atributo y a es un valor del dominio asociado al atributo A_i distinto del valor nulo.

Las <u>Condiciones</u> se construyen a partir de comparaciones, usando los paréntesis y los operadores lógicos ($\vee, \wedge, \neg$).

2.1.2.- A.R. (Operador Selección)

¿Cómo se evalúa la condición *F*(*t*)?

Valor Nulo $\Rightarrow$ Necesidad de una <u>Lógica Trivaluada</u> {V, F, indefinido}:

- si F es de la forma $A_i \,\alpha\, A_j$ entonces $F(t)$ se evalúa a indefinido si al menos uno de los atributos, A_i o A_j tiene valor nulo en t, en caso contrario se evalúa al valor de certeza de la comparación $t(A_i) \,\alpha\, t(A_j)$;
- si F es de la forma $A_i \,\alpha\, a$ entonces $F(t)$ se evalúa a indefinido si A_i tiene valor nulo en t, en caso contrario se evalúa al valor de certeza de la comparación $t(A_i) \,\alpha\, a$; y
- si F es de la forma $nulo(A_i)$ entonces $F(t)$ se evalúa a cierto si A_i tiene valor nulo en t, en caso contrario se evalúa a falso.

2.1.2.- A.R. (Operador Selección)

Lógica Trivaluada: (Tablas de verdad de las conectivas lógicas ∧, ∨ y ¬)

G	H	F = G ∧ H	F = G ∨ H
falso	falso	falso	falso
indefinido	falso	falso	indefinido
cierto	falso	falso	cierto
falso	indefinido	falso	indefinido
indefinido	indefinido	indefinido	indefinido
cierto	indefinido	indefinido	cierto
falso	cierto	falso	cierto
indefinido	cierto	indefinido	cierto
cierto	cierto	cierto	cierto

G	F = ¬ G
falso	cierto
indefinido	indefinido
cierto	falso

2.1.2.- A.R. (Operador Selección)

Ejemplo:

Sea R =

dni	Nombre	dirección
20.450.120	Juan Pérez	Cuenca 20
12.904.569	José Abad	Blasco Ibáñez 35
?	María Gutiérrez	Reina 7
12.345.678	Pepa Gómez	Colón 15

Operaciones:

R **donde** ¬ (nombre = “Juan Pérez”) ∧ (dni > 12.500.500) =

dni	Nombre	dirección
12.904.569	José Abad	Blasco Ibáñez 35

R **donde** ¬ (dni > 12.500.500) =

dni	Nombre	dirección
12.345.678	Pepa Gómez	Colón 15

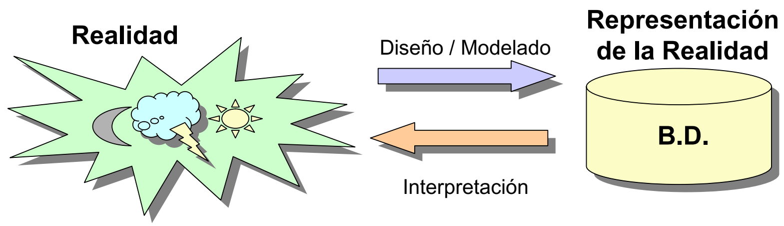
¿ R **donde** (dni <= 12.500.500) ∨ (dni > 12.500.500) = R ?

2.1.2.- Resumen de Operadores

- INSERCIÓN
- BORRADO
- RENOMBRAR
- SELECCIÓN
- PROYECCIÓN
- UNIÓN
- INTERSECCIÓN
- DIFERENCIA
- PRODUCTO CARTESIANO
- CONCATENACIÓN
- DIVISIÓN

Álgebra Relacional

2.2.- Representación de la Realidad



- Para cada objeto de la realidad del que se quiere tener información se define una relación cuyos atributos denotan las propiedades de interés de esos objetos (código, nombre, …) de manera que cada tupla presente en esa relación debe interpretarse como una instancia particular de ese objeto;

- Para representar las asociaciones entre objetos se utilizan referencias explícitas mediante atributos que identifican cada objeto.

2.2.- Representación de la Realidad

EJEMPLO 1:

- Realidad: *Los platos y menúes de un restaurante.*
- Esquema de Base de Datos:

Menú(nom_menú: d4, precio: d2)

Consta_de(nom_plato: d5, nom_menú: d4)

Plato(nom_plato: d5, calorías: d6, cod_vino: d8, nom_cocinero:d7)

Vino(cod_vino: d8, nom_vino:d11, añada: d13, color:d14)

Cocinero(nombre:d7, edad: d9, país:d10)

Interviene(nom_ing: d1, nom_pla:d5,cantidad:d15)

Ingrediente(nom_ing: d1, precio: d2, descripción:d3)

2.2.- Representación de la Realidad

CARDINALIDAD/MULTIPLICIDAD entre dos *objetos* A y B:

Notación Genérica (Capítulo 2 del libro)

$$R\ (A(\min_A,\ \max_A),\ B(\min_B,\ \max_B))$$

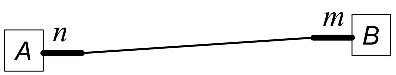
- Cada tupla de A requiere un $\min_A$ de tuplas correspondientes en B, pero como mucho $\max_A$.
- Cada tupla de B requiere un $\min_B$ de tuplas correspondientes en A, pero como mucho $\max_B$.

Ejemplo:

Un vino puede aparecer en muchos platos pero un plato debe tener uno y sólo un vino.

2.2.- Representación de la Realidad

REPRESENTACIÓN INTUITIVA (Access)



Si esta cardinalidad es ∞, entonces corresponde a (0, ∞)
Si esta cardinalidad es 1, entonces corresponde a (0, 1)

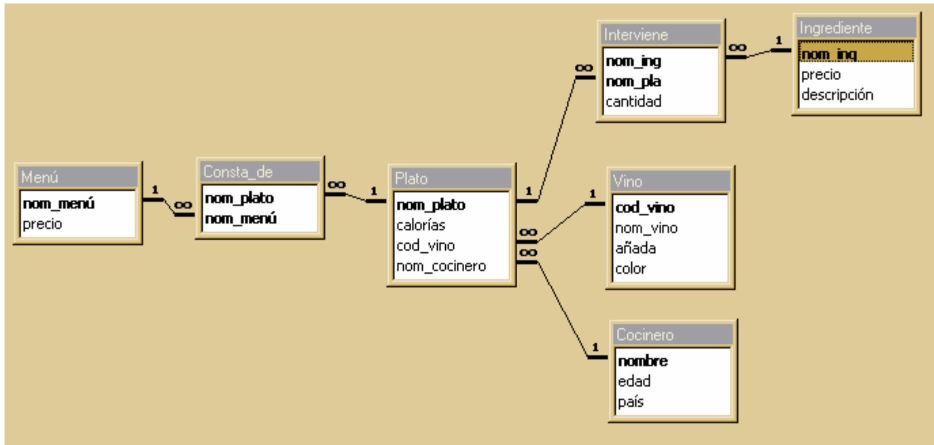
OJO: Esto implica que:

> En el modelo relacional, las cardinalidades varios a varios (muchos a muchos) sólo se pueden hacer a través de una tabla intermedia.

Ejemplo: un plato puede tener muchos ingredientes. A su vez, un ingrediente puede aparecer en muchos platos. *Se necesita la tabla: interviene.*

2.2.- Representación de la Realidad

EJEMPLO 1:



2.1.- Ejercicios de A.R. Consultas

EJEMPLO 2. RESTAURANT:

- Obtener el nombre de los platos de menos de 2.000 calorías:
- Obtener el nombre del cocinero de los platos donde el vino sea blanco.
- Obtener toda la información de los platos de los cocineros rusos:

2.1.- Ejercicios de A.R. Consultas

EJEMPLO 2. RESTAURANT (Cont.):

- Obtener la edad de los cocineros cuyos platos tengan sólo vinos añejos (añada < 1982).
- Obtener el nombre de los menúes que no llevan huevo:
- Obtener el nombre del ingrediente más caro: