

Tema III: El enfoque relacional



**Departamento de
Informática e Ingeniería
de Sistemas**

Universidad Zaragoza

5 MODELO RELACIONAL

- 5.1 Conceptos básicos. Estática del modelo Relacional
- 5.2 Transformación del modelo E/R en el modelo Relacional
- 5.3 Dinámica del modelo Relacional: Álgebra relacional.
- 5.4 Interrogación de una B.D. Relacional utilizando álgebra relacional.

5.1 Conceptos básicos: estática del modelo Relacional

M.R. propuesto por Codd (1970) → *Objetivo: independencia de la Estructura Física*

{ de la ordenación
de la indización
de los caminos de accesos

características
(ventajas)

- Independencia física
- Independencia lógica
- Uniformidad (tablas)
- Sencillez → *accesibilidad y facilidad de uso*
- Flexibilidad

• soporte formal

{ ↑ integridad
optimización (consultas y representación)

T^a Normalización

☛ ↓ redundancias

☛ ayuda diseño

dinámica →

- ✓ álgebra relacional
- ✓ cálculo relacional (de tuplas y de dominios)

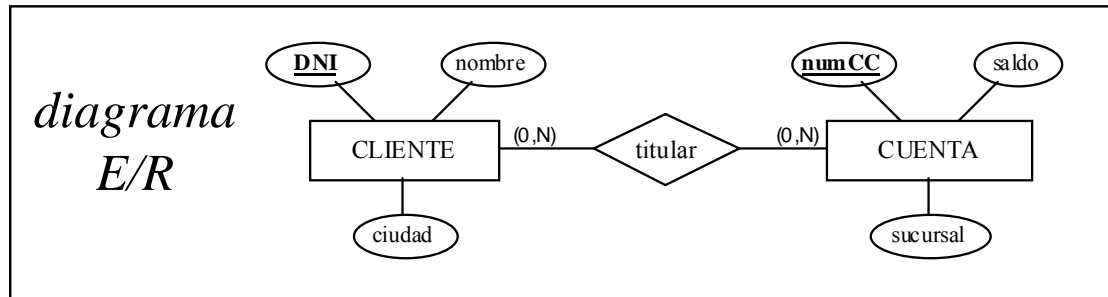
evolución histórica del modelo Relacional.

inconvenientes {
(*pasado reciente*) { • implementaciones poco eficientes (lentas y voluminosas)
• no soporta muchos de los conceptos

muy superados

evolución	≈ 1970	surge el modelo	}	prerrelacional
	1973-1978	prototipos (Ingres, Sistema R, ..)		
	1978	QBE		
	<u>1979</u>	ORACLE	}	relacional
	1980	INGRES		
	1981	SQL		
	1982	DB2		
	1986	SQL/ANSI		
			
	1992	SQL2	}	postrelacional
			
	≈ 2000	SQL3, MROO,		

aproximación intuitiva al modelo relacional.



*Esquema relacional (simplificado)
de la Base de Datos bancaria*

Cuenta

<u>numCC</u>	Sucursal	Saldo
123	23	2567
194	23	125874
237	14	654875
257	18	25984
100	22	0

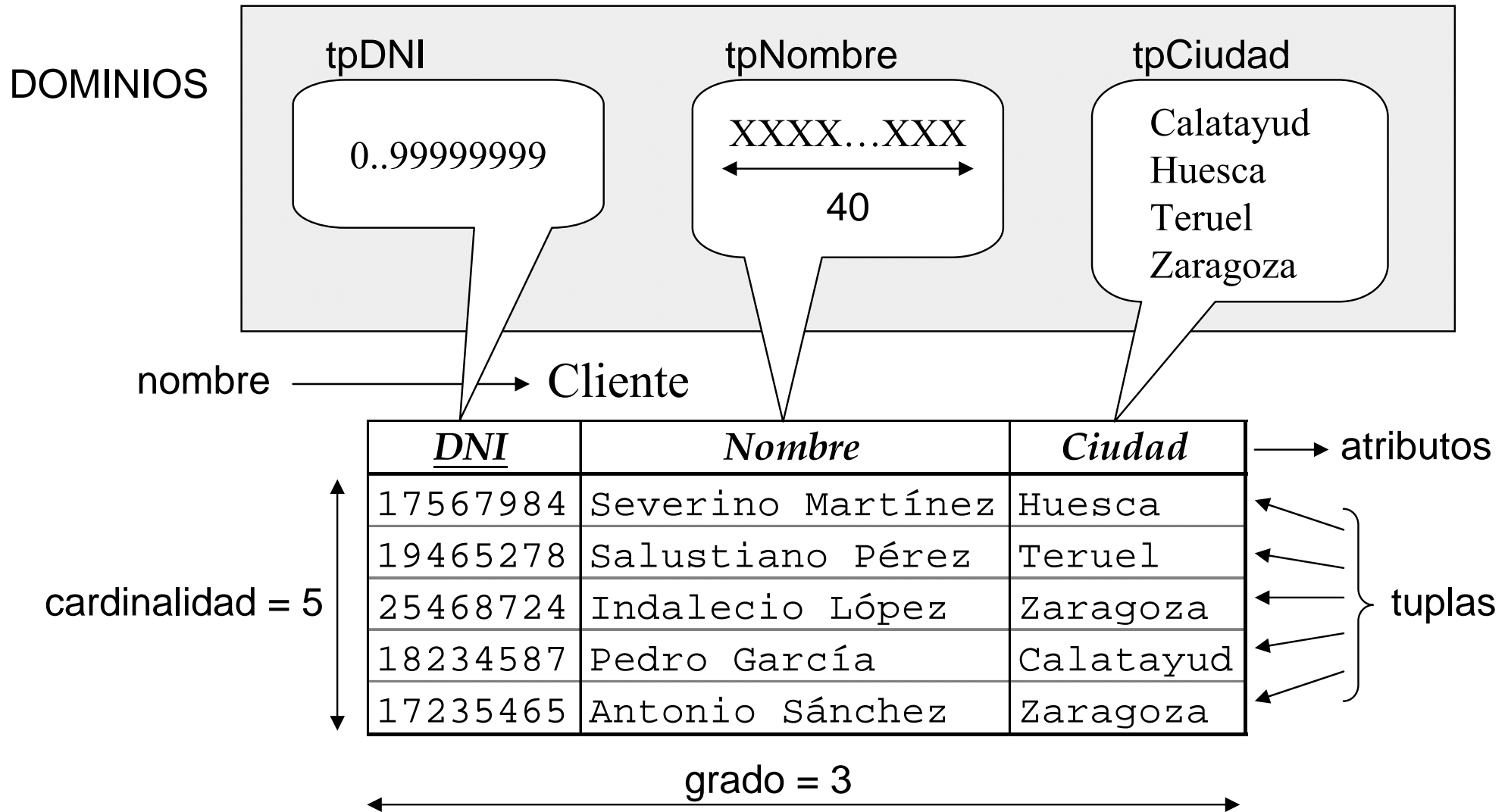
Cliente

<u>DNI</u>	Nombre	Ciudad
17567984	Severino Martínez	Huesca
19465278	Salustiano Pérez	Teruel
25468724	Indalecio López	Zaragoza
18234587	Pedro García	Calatayud
17235465	Antonio Sánchez	Zaragoza

Titular

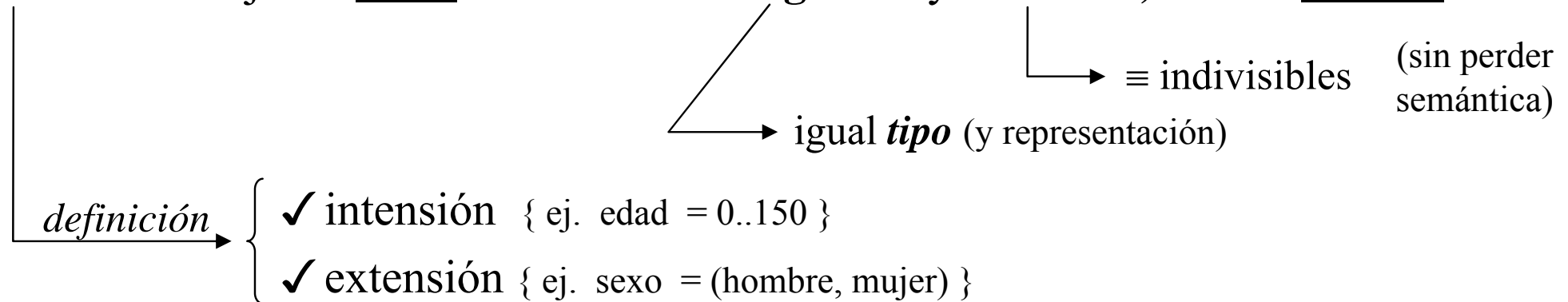
<u>DNI</u>	<u>numCC</u>
17567984	123
18234587	123
25468724	257
25468724	194
17235465	237

introducción al concepto de relación



modelo relacional: Dominios y Atributos

Dominio \equiv conjunto finito de valores **homogéneos** y **atómicos**, con un nombre asociado



Dominio compuesto \equiv combinación de dominios simples (+ restricciones integridad)

Atributo \equiv papel de un dominio en una relación

posibilidad de valores nulos

☛ el Universo Discurso de una base de datos relacional está formado por un conjunto *finito y no vacío* de *atributos* estructurados en *relaciones*

relaciones y esquemas de relación

Relación \approx subconjunto del *producto cartesiano* de los *dominios* sobre los que se define
└─┬─> suele tener un *nombre* asociado

cabecera de relación \equiv conjunto de n pares *atributo - dominio subyacente* $\{(A_i : D_i), \forall i 1.. n\}$
└─┬─> **grado**

cuerpo de relación \equiv conjunto de m tuplas $\{t_1, t_2, \dots, t_m\}$
└─┬─> **cardinalidad** └─┬─> conjunto pares *atributo - valor* $\{(A_i : V_{ij}), \forall i 1.. n\}$

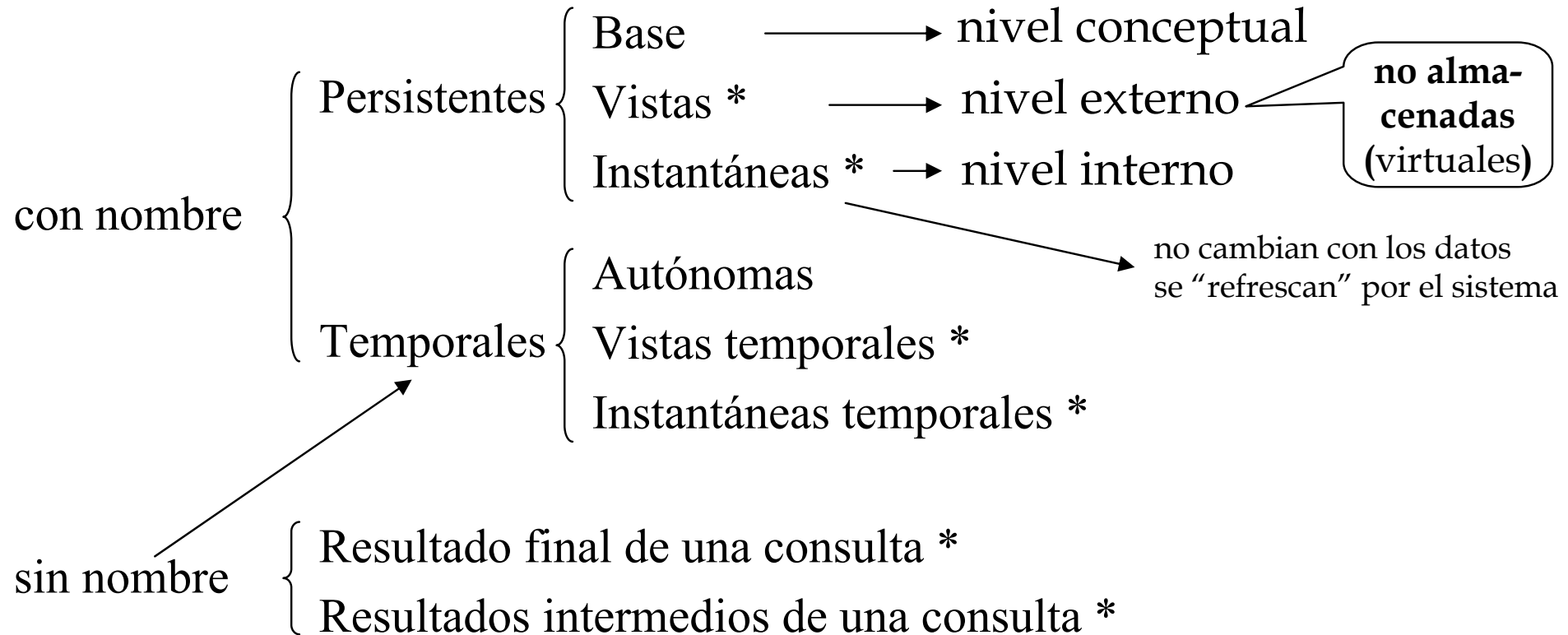
intensión o esquema de relación $R \equiv R (\{A_i : D_i \}, \forall i 1.. n)$

extensión o estado de relación $r(R) \equiv \langle \text{esquema}, \text{cuerpo} \rangle \longrightarrow$ **ocurrencia**

Base Datos Relacional \equiv conjunto de *variables de relación* + . . .

clasificación de las relaciones (ISO 92)

clases de relación:



* significa derivada

concepto de clave en el modelo relacional.

claves de una relación:

✓ **superclave** \equiv cualquier conjunto de atributos que identifica cada tupla de la relación

✓ **clave candidata** \equiv cualquier superclave mínima

✓ **clave primaria** \equiv clave candidata elegida por el diseñador. El resto \rightarrow *alternativas*

✓ **clave ajena** \equiv conjunto de atributos cuyos valores deben coincidir (o ser nulos)
con los de la clave primaria de la relación especificada

\rightarrow en SQL92 puede ser clave candidata

\rightarrow muy importantes para estudio de la integridad

restricciones en el modelo relacional (1)

Restricciones:

✓ inherentes

- no hay dos tuplas iguales
- el orden de las tuplas es irrelevante
- el orden de los atributos no es significativo
- cada atributo (en cada tupla) sólo puede tener un valor del dominio

→ 1ª Forma Normal

+ **integridad de entidad** ≡ ningún atributo que forme parte de la clave primaria puede tomar el valor nulo

el modelo no lo exige para claves alternativas

✓ de usuario (semánticas) básicas

- especificación de clave primaria, (PRIMARY KEY)
- unicidad (UNIQUE)
- obligatoriedad (NOT NULL)
- **integridad referencial** ⇒

los valores de los atributos de una clave ajena (FOREIGN KEY), deben existir, o ser nulos

pb. Valores NULOS

restricciones en el modelo relacional (2)

especificación de clave ajena \Rightarrow indicar acción en operaciones de borrado y modificación

- operación restringida (RESTRICT)
- operación transmisión en cascada (CASCADE)
- operación con puesta a nulos (SET NULL)
- operación con puesta a valor por defecto (SET DEFAULT)

- especificación de restricciones basadas en predicados
 - { Intrarrelación (sobre atributos y tuplas)
 - { Interrelación de Dominio
 - { CHECK y ASSERTION
 - asociadas a la estática o a la dinámica (TRIGGERS) con posible activación de un procedimiento asociado
 - indicación del momento en que se verifica dentro de la transacción

☛ *están en el DICCIONARIO, como el resto de las relaciones*

concepto de Base de Datos Relacional y notación

esquema de relación $\equiv R (\{A_i:D_i\}, S)$

→ lista de restricciones (intrarelación)
→ lista de pares atributo-dominio

esquema de B.D. relacional $\equiv \langle \{R_i\}, \{V_i\} \rangle$

→ lista de restricciones (interrelación)
→ lista de esquemas de relación

Base de Datos relacional \equiv *esquema relacional* + *ocurrencia válida del esquema*

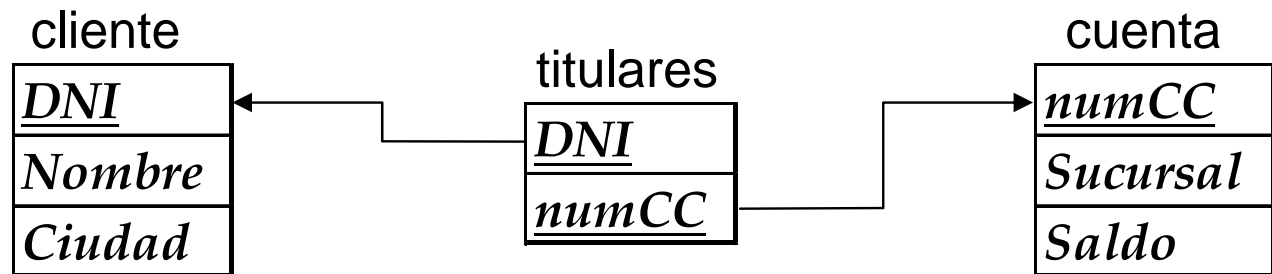
notación: {

- gráfica \equiv diagrama relacional → hay que completarlo con notación textual
- textual (en lenguaje natural, o utilizando un estándar como SQL92)

ejemplo de esquema relacional simple

ejemplo de B.D. Bancaria:

diagrama relacional



Dominios:

tpDNI = 0..99999999;

tpNombre = cadena(50);

esquemas de relación:

cliente = (DNI : tpDNI; Nombre : tpNombre, *UNICO*; Ciudad : cadena(20));

cuenta = (numCC : entero, *clave primaria*; Sucursal : cadena(20) *NO NULO*; saldo : real);

verificar que saldo > 0;

titulares = (DNI : tpDNI; numCC : entero);

clave primaria (DNI, numCC);

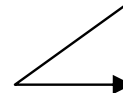
clave ajena (DNI), *referencia a cliente; borrado en cascada*

clave ajena (numCC), *referencia a cuenta; borrado en cascada*

+ restricciones

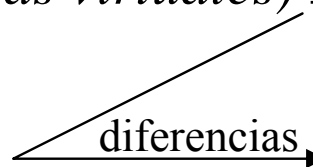
modelo relacional y arquitectura ANSI (1)

el modelo relacional es un modelo lógico \Rightarrow no contempla el nivel interno de la B.D.

 parcialmente en SQL (CREATE INDEX, ...)
y en implementaciones concretas

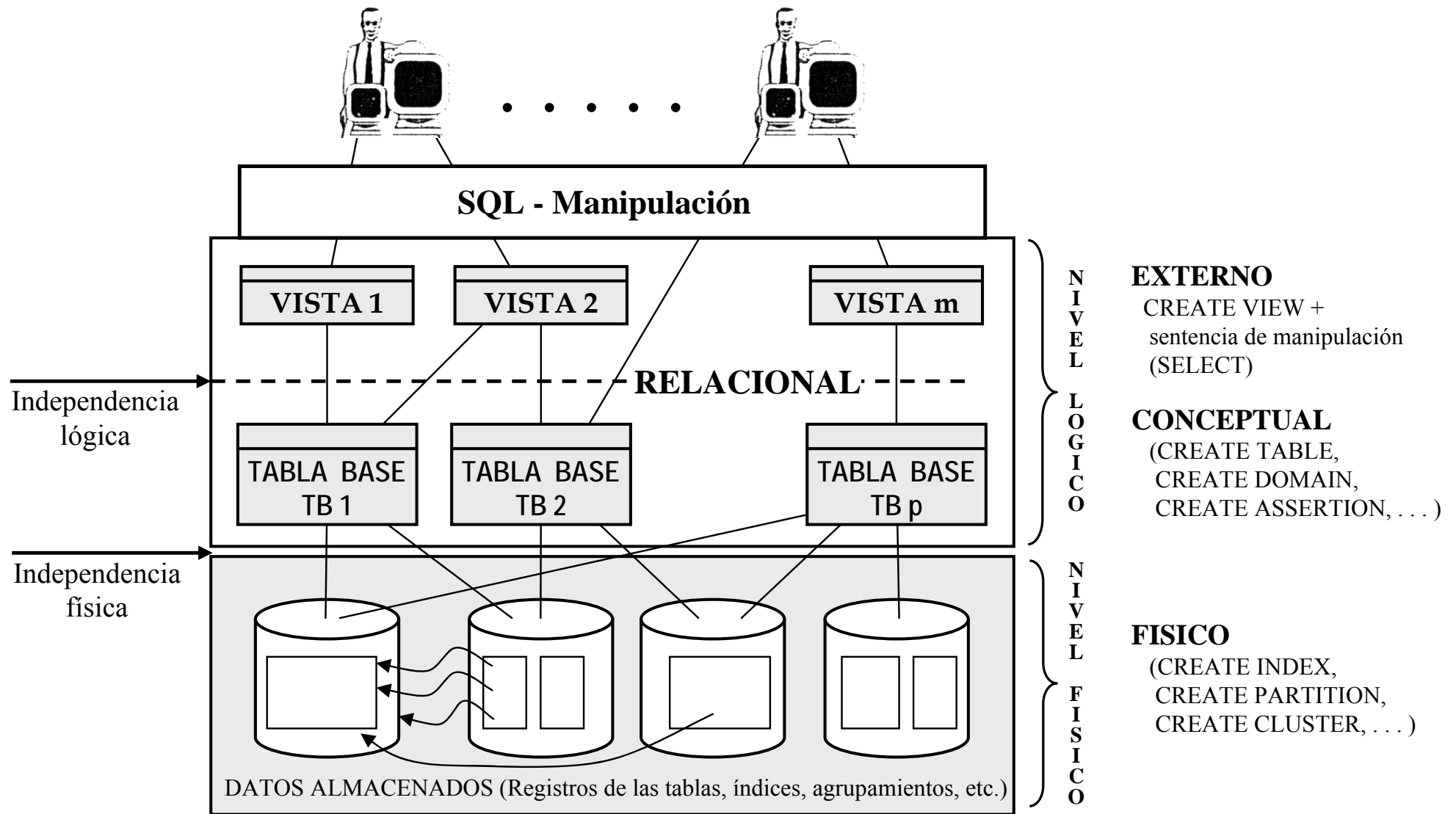
Relaciones base \rightarrow *nivel conceptual* de la B.D. (esquema lógico global)

Vistas (\equiv *tablas virtuales*) \cong *nivel externo* de la B.D.

 diferencias {
• el usuario tiene acceso a más información (autorizaciones, tablas, etc.)
• no todas las vistas son actualizables

Las tablas temporales no tienen correspondencia directa en la arquitectura ANSI

modelo relacional y arquitectura ANSI (2)



5.2 Transformación del modelo E/R en modelo Relacional

las reglas de transformación son bastante simples y fáciles de deducir

✓ *dominios ER* → dominios relacionales

✓ *atributos E/R* → atributos de relaciones

- └─ AIP → CLAVE PRIMARIA
- └─ AIA → UNICO (o clave alternativa)
- └─ obligatoriedad → NO NULO (por defecto opcionales)

✓ *tipos de entidad* → relaciones (tablas)

✓ *tipos de interrelación*

caso general (**N : M**) {

- relación con clave primaria ≡ concatenación de las C.P. de los tipos de entidad que participan
- + especificación de claves ajenas

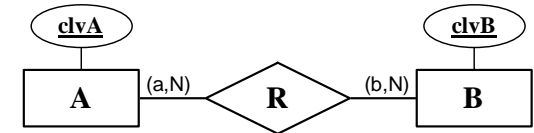
casos (**1 : N**) y (**1 : 1**) {

- propagación de clave
- + especificación de clave ajena
- o como el caso general

al tipo correspondiente a la N,
o a cualquiera de ellos si es 1:1

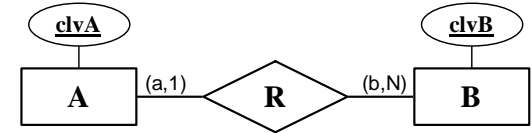
transformación de tipo de correspondencia N:M

las cardinalidades mínima y máxima \Rightarrow casos particulares



- ① $R(A(0,N), B(0,N)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, \dots); R(\underline{clvA}, \underline{clvB})$
 - \rightarrow clave ajena de B
 - \rightarrow clave ajena de A
- ② $R(A(0,N), B(1,N)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, \dots); R(\underline{clvA}, \underline{clvB})$
 - \rightarrow clave ajena de B
 - \rightarrow clave ajena de A
 - $\rightarrow \forall$ ocurrencia de **clvA** en A, \exists en R
 - $\rightarrow \neq (\Pi_{clvA}(A) - \Pi_{clvA}(R)), \rightarrow \Pi_{clvA}(R) = \Pi_{clvA}(A)$
 - \rightarrow **clvA referencia a R(clvA)**
- ③ $R(A(1,N), B(0,N)) \rightarrow$ similar al anterior
- ④ $R(A(1,N), B(1,N)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, \dots); R(\underline{clvA}, \underline{clvB})$
 - \rightarrow clave ajena de B
 - \rightarrow clave ajena de A
 - \rightarrow **clvB referencia a R(clvB)**
 - \rightarrow **clvA referencia a R(clvA)**

transformación de tipo de correspondencia 1:N



- ① $R(A(0,1), B(0,N)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, clvA, \dots);$
↘ clave ajena de A
- ② $R(A(1,1), B(0,N)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, clvA, \dots);$
↘ clave ajena de A, NO NULO
- ③ $R(A(0,1), B(1,N)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, clvA, \dots);$
↘ clave ajena de A
↘ clvA referencia a B(clvA)
- ④ $R(A(1,1), B(1,N)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, clvA, \dots);$
↘ clave ajena de A, NO NULO
↘ clvA referencia a B(clvA)

con 3 tablas:

- ①b $R(A(0,1), B(0,N)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, \dots); R(\underline{clvA}, \underline{clvB})$
↘ clave ajena de B
↘ clave ajena de A, NO NULO
↖ **sólo clave primaria**

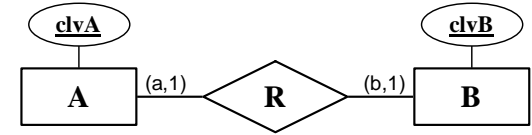
el resto de casos, mezcla del caso general y el anterior

transformación de tipo de correspondencia 1:1

si ambas cardinalidades máximas son = 1

⇒ se puede propagar la clave a cualquiera de las relaciones

→ hacia la tabla más pequeña



① $R(A(0,1), B(0,1)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, clvA, \dots);$

→ clave ajena de A, UNICO

② $R(A(1,1), B(0,1)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, clvA, \dots);$

→ clave ajena de A, UNICO, NO NULO

③ $R(A(0,1), B(1,1)) \rightarrow A(\underline{clvA}, \dots); B(\underline{clvB}, clvA, \dots);$

→ clave ajena de A, UNICO

→ clvA referencia a B(clvA)

④ $R(A(1,1), B(1,1)) \rightarrow$ mejor fusionar las tablas

↪ no interesa la propagación de clave si

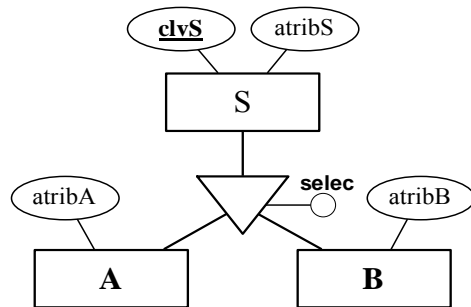
- hay muy pocas ocurrencias de la interrelación
→ daría lugar a muchos "nulos"
- en el futuro será una interrelación N:M
- hay atributos en la interrelación

otras transformaciones de atributos e interrelaciones

- ✓ los *atributos de los tipos de interrelación* se añaden a la traducción de la interrelación
- ✓ *tipos de interrelación ternaria* → similar, pero puede haber casos particulares (p.e. 1:N:N)
- ✓ *tipos de entidad débiles* → como los fuertes, pero añadiendo a su clave primaria la clave primaria de los tipos de entidad de los que depende + especificación de clave ajena
- ✓ *tipos de interrelación en exclusión* → restricciones adicionales (predicados)
+ posibilidad de “trucos” de implementación
- ✓ *atributos multivaluados* → nueva relación cuya clave primaria está formada por la clave primaria y el atributo multivaluado
+ especificación de clave ajena → *casos particulares*

transformación de la generalización / especialización (1)

✓ la *generalización/especialización* plantea varias posibilidades



AtribA	A...	<u>clvS</u>	selec	AtribS...	AtribB	B...

AtribA	A...	<u>clvS</u>	selec	AtribS...	AtribB	B...

①

<u>clvS</u>	selec	AtribS...

AtribA	A...	<u>clvS</u>

<u>clvS</u>	AtribB	B...

②

AtribA	A...	<u>clvS</u>	selec	AtribS...

<u>clvS</u>	selec	AtribS...	AtribB	B...

③ + tabla con resto

transformación de la generalización / especialización (2)

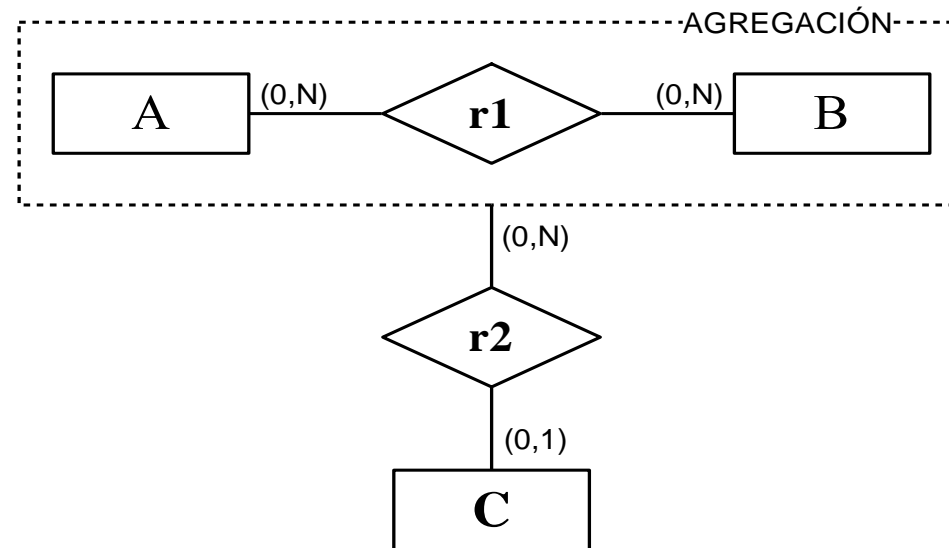
- ① *una única tabla con toda la información* (supertipo y subtipos)
S(clvS, atribS, selec..., atribA, ..., atribB, ...);
→ Acceso más eficiente
- ② *una tabla para cada uno de los tipos de entidad* (supertipo y subtipos)
S(clvS, atribS, selec, ...);
A(clvS, atribA, ...);
B(clvS, atribB, ...);
→ Mejor la semántica
- ③ *una tabla para cada uno de los subtipos* (incluyen la del supertipo)
A(clvS, atribS, selec, ..., atribA, ...);
B(clvS, atribS, selec, ..., atribB, ...);
S(clvS, atribS, selec, ...);
→ Acceso eficiente peor la semántica

sólo para los que no están en ningún subtipo

habrá que añadir *restricciones adicionales* para implementar la *semántica* y los *casos particulares*

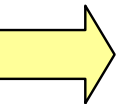
transformación de la agregación

✓ la *agregación* queda representada por la interrelación



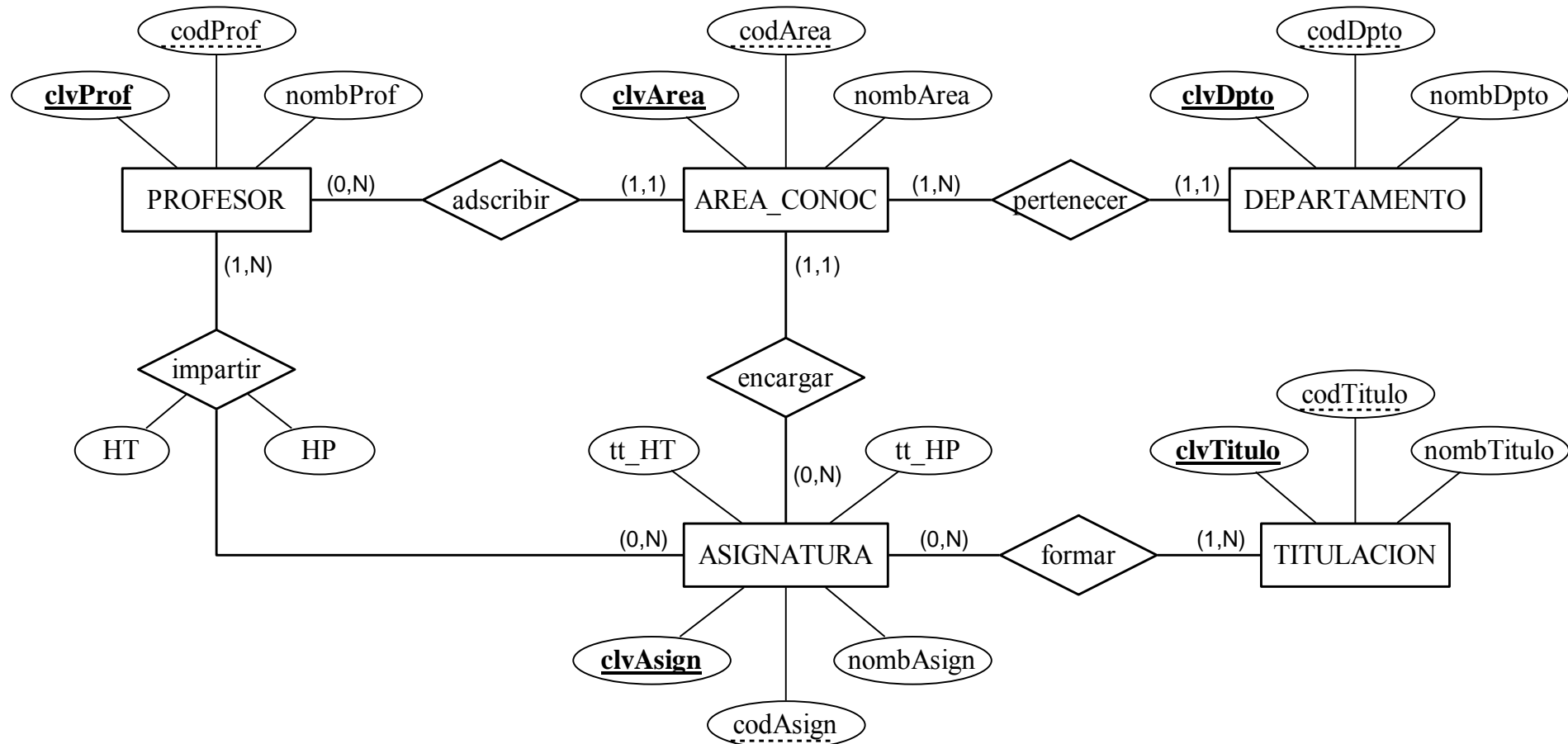
puede haber muchos casos particulares, pero en la práctica sólo algunos y simples

Ejemplos: . . .



transformación de la B.D. de la Universidad (1)

diagrama E/R:



transformación de la B.D. de la Universidad (2)

Atributos de tipos de entidad:

clvProf: tpClave; **AIP**
codProf: tpCódigo; **AIA**
nombProf: tpNombre;

clvArea: tpClave; **AIP**
codArea: tpCódigo; **AIA**
nombArea: tpNombre;

clvDpto: tpClave; **AIP**
codDpto: tpCódigo; **AIA**
nombDpto: tpNombre;

clvAsign: tpClave; **AIP**
codAsign: tpCódigo; **AIA**
nombAsign: tpNombre;
tt_HT, tt_HP: tpHoras;

clvTitulo: tpClave; **AIP**
codTitulo: tpCódigo; **AIA**
nombTitulo: tpNombre;

Atributos de interrelaciones:

impartir.HT, impartir.HP: tpHoras;

Dominios:

tpClave = entero;
tpCódigo = 0..99999;
tpNombre = cadena(40);
tpHoras = 0..400;

Restricciones:

- 1) Ningún profesor puede impartir docencia en una asignatura que no esté encargada a su área de conocimiento
- 2) El total de horas impartidas de una asignatura debe ser menor o igual que el correspondiente a la asignatura.

• • •

transformación de la B.D. de la Universidad (3)

Dominios:

tpClave = entero; tpNombre = cadena(40);
tpCódigo = 0..99999; tpHoras = 0..400;

Esquemas de relación:

Departamento = (clvDpto: tpClave; codDpto: tpCódigo, *UNICO, NO NULO*; nombDpto: tpNombre, *NO NULO*);

eliminar pertenecer y propagar clvDpto a relación AreaConoc

AreaConoc = (clvArea: tpClave; codArea: tpCódigo, *UNICO, NO NULO*; nombArea: tpNombre, *NO NULO*
clvDpto: tpClave, *NO NULO, clave ajena de Departamento*);

eliminar adscribir y propagar clvArea a relación Profesor

Profesor = (clvProf: tpClave; codProf: tpCódigo, *UNICO, NO NULO*; nombProf: tpNombre, *NO NULO*);
clvArea: tpClave, *NO NULO, clave ajena de AreaConoc*);

eliminar encargar y propagar clvArea a relación Asignatura

Asignatura = (clvAsign: tpClave; codAsign: tpCódigo, *UNICO, NO NULO*; nombAsign: tpNombre, *NO NULO*);
clvArea: tpClave, *NO NULO, clave ajena de AreaConoc*; tt_HT, tt_HP: tpHoras, *NO NULOS*);

transformación de la B.D. de la Universidad (4)

Titulacion = (clvTitulo: tpClave; codTitulo: tpCódigo, *UNICO, NO NULO*; nombTitulo: tpNombre, *NO NULO*);

ImparteAsign = (clvProf: tpClave; clvAsign: tpClave; HT, HP: tpHoras, *NO NULOS*);
clave ajena (clvProf), referencia a Profesor; borrado en cascada
clave ajena (clvAsign), referencia a Asignatura; borrado en cascada

AsignTitulo = (clvAsign: tpClave; clvTitulo: tpClave);
clave ajena (clvAsign), referencia a Asignatura; borrado en cascada
clave ajena (clvTitulo), referencia a Titulacion; borrado en cascada

restricciones:

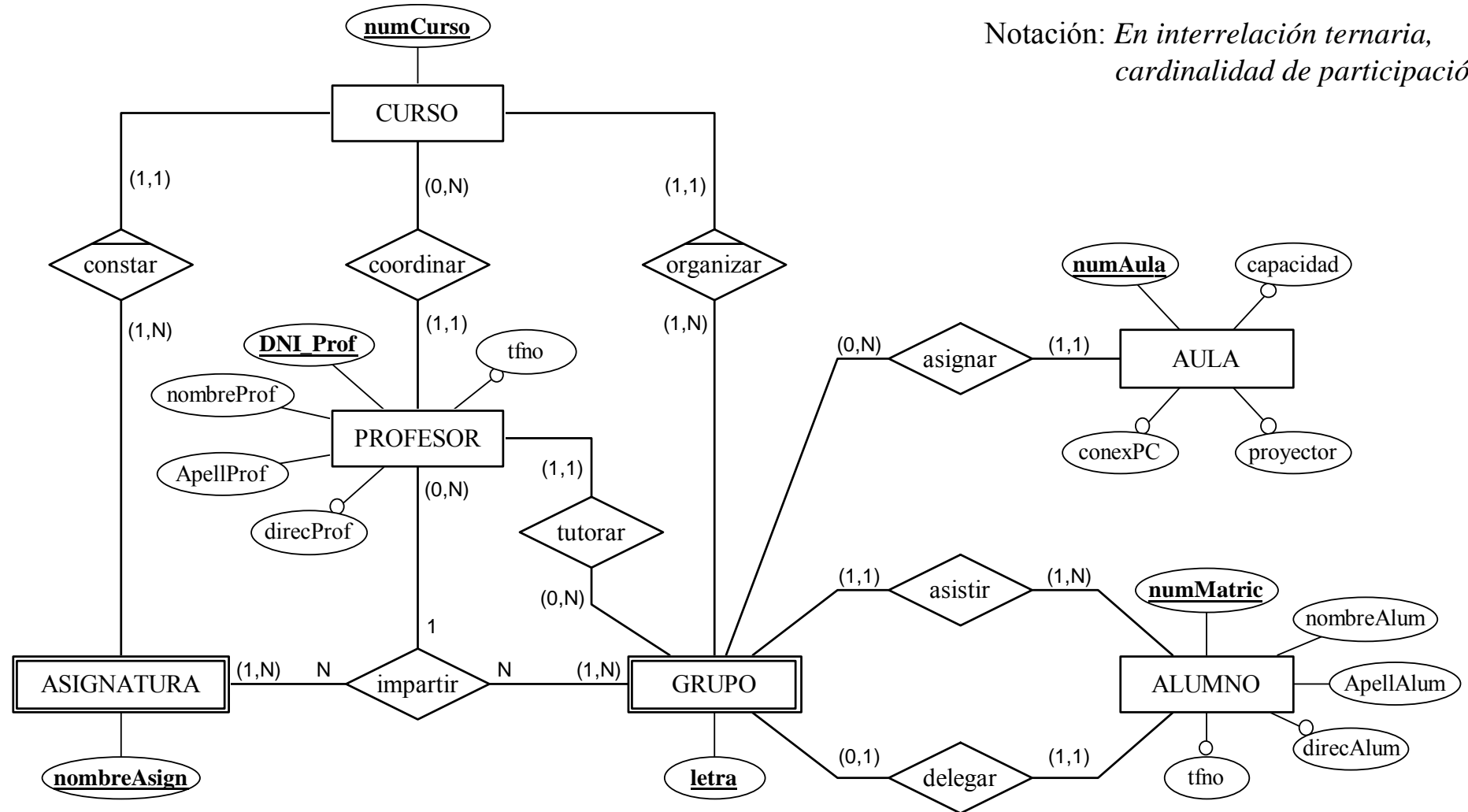
verificar que \forall *ocurrencia de* clvDpto *en* Departamento, \exists *en* AreaConoc
verificar que \forall *ocurrencia de* clvAsign *en* Asignatura, \exists *en* AsignTitulo
verificar que \forall *ocurrencia de* clvAsign *en* Asignatura, \exists *en* ImparteAsign

verificar que \forall *ocurrencia en* ImparteAsign, *la* clvArea *del* Profesor *coincide con la de la* Asignatura

verificar que *la suma de* HT, *y la de* HP *en* ImparteAsign, *agrupando por* asignatura
es menor o igual que el valor de tt_HT *y* tt_HP *correspondiente a la* Asignatura

transformación de la B.D. del colegio (1)

diagrama E/R:



Notación: *En interrelación ternaria, cardinalidad de participación*

transformación de la B.D. del colegio (2)

Dominios:

tpNombre = cadena(40);
tpDirección = cadena(40);
tpTfno = 0..999999999;

Atributos de tipos de entidad:

DNI_Prof: cadena(9); **AIP**
nombreProf: tpNombre;
apellProf: tpNombre;
direcProf: tpDireccion;
profesor.tfno: tpTfno;

numMatric: 0..9999; **AIP**
nombreAlum: tpNombre;
apellAlum: tpNombre;
direcAlum: tpDireccion;
alumno.tfno: tpTfno;

Atributos de tipos de entidad (cont.):

numCurso: 0..8; **AIP**
letra: 'A'..'G'; **AIP**
nombreAsign: tpNombre; **AIP**
numAula: 0..99; **AIP**
capacidad: 0..150;
conex_PC: booleano;
proyector: booleano;

Restricciones:

1) Un alumno sólo puede ser delegado del grupo al que asiste.

• • •

transformación de la B.D. del colegio (3)

Dominios:

tpDNI = cadena(9); tpNombre = cadena(40); tpNombAsign = cadena(16);
tpCurso = 1..8; tpTfno = 0..999999999; tpNumMatric = 0..99999;
tpIdAula = 1..69; tpIdGrupo = 'A'..'G'; tpCapacAula = 10..100;

Esquemas de relación:

Profesor = (DNI_Prof: tpDNI; tfno: tpTfno; nombreProf: tpNombre, *NO NULO*);
ApellProf: tpNombre, *NO NULO*; direcProf: tpNombre);

eliminar coordinar y propagar DNI_Prof a relación Curso

Curso = (numCurso: tpCurso; ProfCoord: tpDNI, *NO NULO*, *clave ajena de Profesor*);

añadir a entidad débil asignatura la clave de entidad fuerte numCurso

eliminar constar y propagar numCurso a relación Asignatura (ya hecho)

Asignatura = (nombreAsign: tpNombre; numCurso: tpCurso, *clave ajena de Curso*);
verificar que \exists numCurso *en* Asignatura \forall *ocurrencia de* Curso

Aula = (numAula: tpIdAula; capacidad: tpCapacAula; conexPC, proyector: booleano);

transformación de la B.D. del colegio (4)

eliminar asignar y propagar numAula a relación Grupo
eliminar tutorar y propagar DNI_Prof a relación Grupo
eliminar delegar y propagar numMatric a relación Grupo
añadir a entidad débil Grupo la clave de entidad fuerte numCurso
eliminar organizar y propagar numCurso a relación Grupo (ya hecho)

Grupo = (letra: tpIdGrupo; tfno: numCurso: tpCurso; numAula: tpIdAula, *NO NULO*;
profTutor: tpDNI, *NO NULO*; deleGrupo: tpNumMatric, *NO NULO, UNICO*);
clave ajena (numCurso), referencia a Curso; borrado en cascada
clave ajena (numAula), referencia a Aula; borrado en cascada
clave ajena (profTutor), referencia a Profesor; borrado en cascada
clave ajena (deleGrupo), referencia a Alumno; borrado en cascada
verificar que \exists numCurso en Grupo \forall *ocurrencia de curso*
verificar que \exists (deleGrupo, numcurso, letra) en Alumno

eliminar asistir y propagar (numCurso, letra) a relación Alumno

Alumno = (numMatric: tpNumMatric; nombreAlum, ApellAlum: tpNombre, *NO NULOS*;
direcAlum: tpNombre; tfno: tpTfno; numCurso: tpCurso, *NO NULO*; letra: tpIdGrupo, *NO NULO*);
clave ajena (numCurso, letra), referencia a Grupo; borrado en cascada
verificar que \exists (letra , numCurso) en Alumno \forall *ocurrencia de Grupo*

transformación de la B.D. del colegio (5)

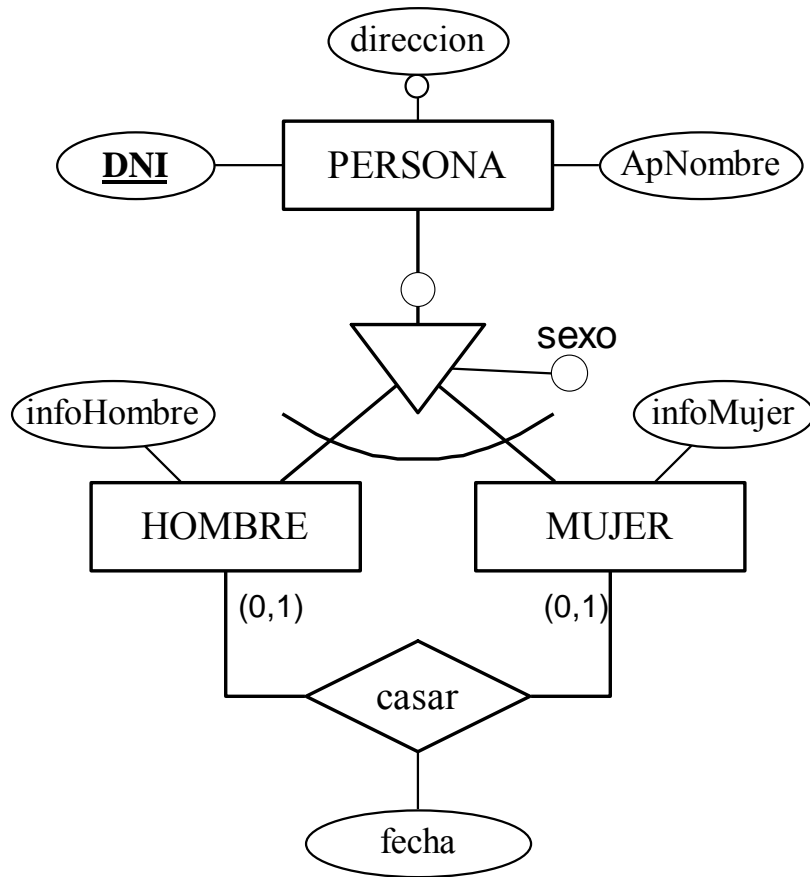
impartir = (DNI_Prof: tpDNI , **NO NULO**; numCurso: tpCurso; letra: tpIdGrupo; nombreAsign: tpNombre);
clave ajena (numCurso, letra), *referencia a* Grupo; *borrado en cascada*
clave ajena (DNI_Prof), *referencia a* Profesor; *borrado en cascada*
clave ajena (nombreAsign, numCurso), *referencia a* Asignatura; *borrado en cascada*

verificar que \exists (letra , numCurso) en impartir \forall *ocurrencia de* Grupo

verificar que \exists (nombreAsign, numCurso) en impartir \forall *ocurrencia de* Asignatura

- *Verificar el esquema relacional obtenido y añadir las restricciones que faltan*
- *Haga pequeñas modificaciones en el enunciado y estudie cómo afectan al esquema relacional, así como posibles mejoras a la solución propuesta*
- *Transforme el resto de los esquemas E/R diseñados*

otros ejemplos de transformación



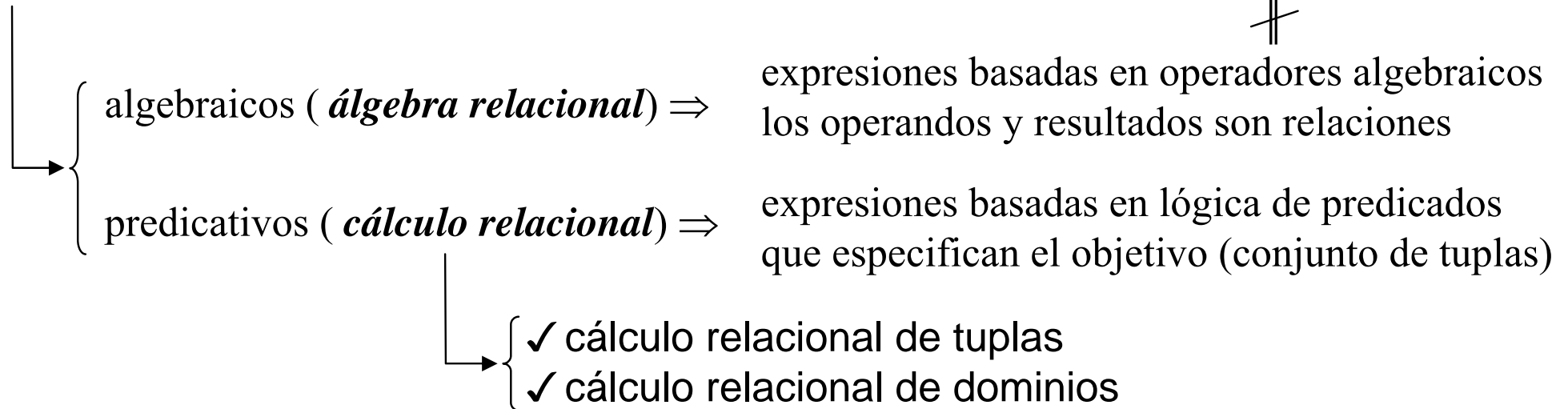
¿y como multivaluado ?

5.3 Dinámica del modelo Relacional: Álgebra relacional

la componente dinámica de un modelo \equiv reglas de transformación de estado aplicables a las ocurrencias de la B.D. (operadores y restricciones)

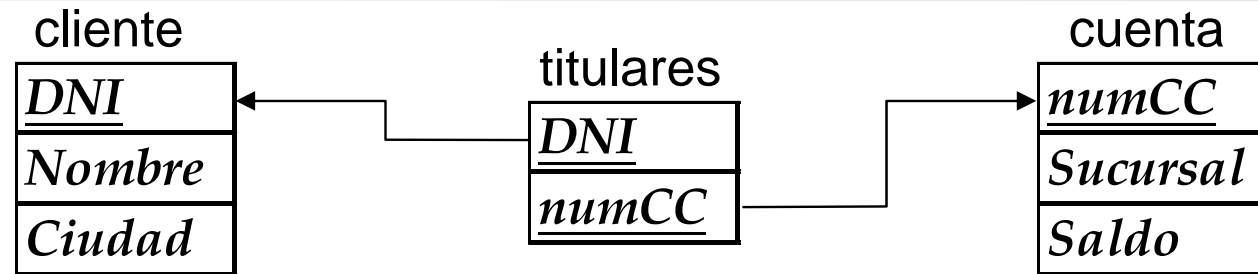
operadores básicos $\left\{ \begin{array}{l} \text{Selección } \langle \text{condición} \rangle \\ \text{Acción } \langle \text{objetivo} \rangle \end{array} \right. \longrightarrow \text{ se expresan mediante LMD}$

LMD relacionales $\left\{ \begin{array}{l} \bullet \text{ operan sobre conjuntos de relaciones (tuplas)} \\ \bullet \text{ suelen ser lenguajes de especificación} \end{array} \right. \equiv \begin{array}{l} \text{no-navegacionales y} \\ \text{no-procedurales} \end{array}$



ejemplos de consultas en álgebra y cálculo relacional (1)

Ejemplos:



1) *numCC, Sucursal y saldo* de todas las cuentas con $\text{saldo} > 100000$:

Álgebra $\longrightarrow \sigma_{\text{saldo} > 100000} (\text{cuenta})$

cálculo de tuplas $\longrightarrow \{ t \mid t \in \text{cuenta} \wedge t[\text{saldo}] > 100000 \}$

cálculo de dominios $\longrightarrow \{ \langle a, b, c \rangle \mid \langle a, b, c \rangle \in \text{cuenta} \wedge c > 100000 \}$

2) *nombre* de todos los clientes del banco:

Álgebra $\longrightarrow \Pi_{\text{nombre}} (\text{cliente})$

cálculo de tuplas $\longrightarrow \{ t \mid \exists s \in \text{cliente} (t[\text{nombre}] = s[\text{nombre}]) \}$

cálculo de dominios $\longrightarrow \{ \langle n \rangle \mid \exists d, c (\langle d, n, c \rangle \in \text{cliente}) \}$

ejemplos de consultas en álgebra y cálculo relacional (2)

3) *DNI* de todos los clientes del banco que tienen cuenta:

Álgebra $\longrightarrow \Pi_{\text{DNI}}(\text{cliente} \bowtie \text{titulares})$

cálculo de tuplas $\longrightarrow \{ d \mid \exists t \in \text{titulares} (d [\text{DNI}] = t [\text{DNI}]) \}$

cálculo de dominios $\longrightarrow \{ \langle d \rangle \mid \exists c (\langle d, c \rangle \in \text{titulares}) \}$

4) *nombre y numCC* de todos los clientes :

Álgebra $\longrightarrow \Pi_{\text{nombre, numCC}}(\text{cliente} \bowtie \text{titulares})$

cálculo de tuplas $\longrightarrow \{ s \mid \exists c \in \text{clientes} (s [\text{nombre}] = c [\text{nombre}] \wedge \exists t \in \text{titulares} (c [\text{DNI}] = t [\text{DNI}] \wedge s [\text{numCC}] = t [\text{numCC}])) \}$

cálculo de dominios $\longrightarrow \{ \langle a, b \rangle \mid \exists d, c (\langle d, a, c \rangle \in \text{cliente} \wedge \langle d, b \rangle \in \text{titulares}) \}$

introducción al Álgebra Relacional

operandos y resultados son *relaciones* R_i y los *operadores* $O_j / O_j (R_i) \rightarrow R_k$

operadores básicos	<i>unión</i>	\cup	$R1 \cup R2$
	<i>diferencia</i>	$-$	$R1 - R2$
	<i>producto cartesiano</i>	\times	$R1 \times R2$
	<i>selección</i>	σ	$\sigma_{\text{condición}} (R)$
	<i>proyección</i>	Π	$\Pi_{\text{atributos}} (R)$
operadores derivados	<i>intersección</i>	\cap	$R1 \cap R2$
	<i>cociente</i>	\div	$R1 \div R2$
	<i>join</i>	\bowtie	$R1 \bowtie R2$

- además de añaden otros operadores (*asignación, tratamientos aritméticos, agrupación, etc.*) para facilitar la descripción de consultas más complejas \rightarrow álgebra relacional extendida
- normalmente sólo se usan para obtener información (otros operadores para inserción, eliminación, etc.)

álgebra relacional: operador selección

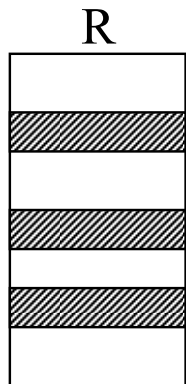
Sean:

$R(A) = (A_1 : D_1, \dots, A_n : D_n)$ la *intensión* (esquema de relación) de una relación R de grado n , y

$r(R) = \{ t_i \}$ con $1 \leq i \leq m$, y $t_i = \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle / v_{ij} \in D_j$, la *extensión* de R .

operadores unarios

Ⓐ *Selección, o restricción* σ sobre una relación R , según un predicado p



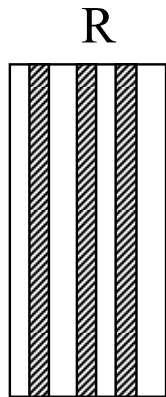
$\sigma_p(R) \equiv$ selección de las tuplas de la relación R que verifican el predicado p

$\sigma_p(R) = R'$, con $r'(R') = \{ t_i \in r(R) \mid p(t_i) \}$, donde p es un predicado de selección formado por una expresión lógica de cláusulas de la forma $A_i \theta A_j$, o $A_i \theta \text{cte.}$, siendo θ operador de comparación. Los operadores lógicos serán AND, OR y NOT

☞ el grado de R' , n' , es igual a n , y su cardinalidad, $m' \leq m$

álgebra relacional: operador proyección

ⓑ *Proyección* Π de una relación R sobre un conjunto de atributos X



$\Pi_X(R) \equiv$ selección de los atributos X de la relación R

$\Pi_X(R) = R'(X)$, con $r'(R') = \{t_i(X) \mid X \subset A\}$, es decir, las tuplas de la relación original definidas sobre los atributos X , eliminando las duplicadas.

➔ el grado de R' es $n' < n$, y su cardinalidad, $m' \leq m$

operadores binarios

\Rightarrow tienen como operandos dos relaciones, y el resultado es otra relación

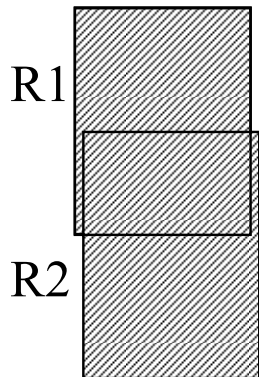
➔ en algunos operadores (\cup , $-$, \cap) los operandos deben ser **compatibles**; es decir, deben estar definidos sobre el mismo conjunto de dominios (**semánticamente equivalentes**)

para cambiar los nombres de los atributos y su orden, se puede utilizar el operador unario: **RENOMBRAR** ($R; A'$), donde A' describe la transformación de atributos de R

álgebra relacional: operadores unión y diferencia

(c) Unión

\cup de dos relaciones *compatibles* R1 y R2



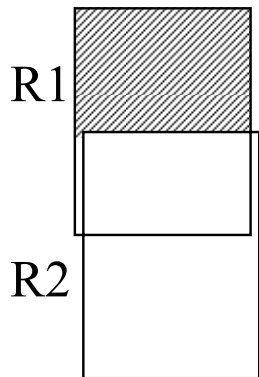
$R1 \cup R2 \equiv$ relación formada por las tuplas de ambas relaciones R1 y R2

$R1 \cup R2 = R'$, con $r'(R') = \{ t_i \mid t_i \in r1(R1) \text{ or } t_i \in r2(R2) \}$

➡ el grado de R' es $n' = n1 = n2$, y su cardinalidad es $m' \geq m1$, $m' \geq m2$, y $m' \leq m1 + m2$

(d) Diferencia

$-$ de dos relaciones *compatibles* R1 y R2



$R1 - R2 \equiv$ relación formada por las tuplas de R1 que no aparecen en R2

$R1 - R2 = R'$, con $r'(R') = \{ t_i \mid t_i \in r1(R1) \text{ and } t_i \notin r2(R2) \}$

➡ el grado de R' es, $n' = n1 = n2$, y su cardinalidad, $m' \leq m1$

álgebra relacional: operador producto cartesiano

④ *Producto cartesiano generalizado* \times *de dos relaciones R1 y R2* (*cross JOIN*)

$R1 \times R2 \equiv$ relación formada por todas las tuplas obtenidas concatenando una tupla de R1 con otra de R2

$R1 \times R2 = R'$, con $r'(R') = \{ \langle v_{i1}, \dots, v_{in1}, v_{j1}, \dots, v_{jn2} \rangle \mid \langle v_{i1}, \dots, v_{in1} \rangle \in r1(R1), \langle v_{j1}, \dots, v_{jn2} \rangle \in r2(R2) \}$

➡ el grado de R' es $n' = n1 + n2$, y su cardinalidad $m' = m1 * m2$

➡ *Para distinguir los atributos que tienen el mismo nombre en las dos relaciones, se antepone el nombre de la relación: R1.X, R2.X*

álgebra relacional: ejemplo de operaciones básicas

Sean: $R1(A1, A2)$; $R2(A3, A4)$; $R3 = \text{RENOMBRAR}(R2; A1=A3, A2=A4)$

$$r(R1) =$$

A1	A2
a	1
b	2
b	1
c	3

$$r(R2) =$$

A3	A4
x	1
b	2

$$\sigma_{A1=b}(R1) =$$

A1	A2
b	1
b	2

$$\Pi_{A1}(R1) =$$

A1
a
b
c

$$R1 \cup R3 =$$

A1	A2
a	1
b	2
b	1
c	3
x	1

$$R1 - R3 =$$

A1	A2
a	1
b	1
c	3

$$R1 \cap R3 =$$

A1	A2
b	2

$$R1 \times R2 =$$

A1	A2	A3	A4
a	1	x	1
a	1	b	2
b	2	x	1
b	2	b	2
b	1	x	1
b	1	b	2
c	3	x	1
c	3	b	2

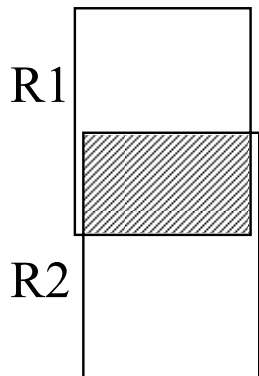
álgebra relacional: operador intersección

operadores derivados

→ se introducen para simplificar las expresiones, ya que pueden ser descritos con los operadores básicos

Ⓣ Intersección

\cap de dos relaciones **compatibles** $R1$ y $R2$



$R1 \cap R2 \equiv$ relación formada por las tuplas que pertenecen a $R1$ y a $R2$

$R1 \cap R2 = R'$, con $r'(R') = \{ t_i \mid t_i \in r1(R1) \text{ and } t_i \in r2(R2) \}$

→ el grado de R' es $n' = n1 = n2$, y su cardinalidad $m' \leq m1$, $m' \leq m2$

Obsérvese que: $R' = (R1 \cup R2) - ((R1 - R2) \cup (R2 - R1))$, o bien

$R' = R1 - (R1 - R2)$, o $R' = R2 - (R2 - R1)$

álgebra relacional: operador división

ⓐ *División* \div de dos relaciones $R1$ y $R2$

$R1 \div R2 \equiv$ relación formada por todas las tuplas (*cociente*) tales que su producto cartesiano por $R2$ (*divisor*) está incluida en $R1$ (*dividendo*)

supóngase, para simplificar, que los k últimos atributos de $R1$ coinciden con los de $R2$

$R1 \div R2 = R'$, con $R'(A') = (A_1 : D_1, \dots, A_{n1-k} : D_{n1-k})$, y

$$r'(R') = \{ \langle v_{i1}, \dots, v_{i(n1-k)} \rangle \mid \forall \langle v_{j(n1-k+1)}, \dots, v_{jn1} \rangle \in r_2(R2), \\ \exists \langle v_{i1}, \dots, v_{i(n1-k)}, v_{j(n1-k+1)}, \dots, v_{jn1} \rangle \in r(R1) \}$$

➡ el grado de R' es $n' = n1 - n2$, y su cardinalidad $m' \leq m1 / m2$

Obsérvese que: $R1 \div R2 = \underbrace{\prod_{A'}(R1)}_{\text{las que hay}} - \underbrace{\prod_{A'}((\underbrace{\prod_{A'}(R1) \times R2}_{\text{tuplas que debería haber}}) - R1)}_{\text{las que no pueden ser}}$

ej. de interpretación: Obtener los *alumnos* que están *matriculados* en ***todas*** las *asignaturas*

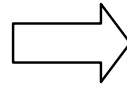
álgebra relacional: ejemplo de operación división

$$r(R4) =$$

A1	A2	A3	A4
a	1	x	1
a	1	b	2
b	2	x	1
b	2	b	2
b	1	x	1
c	3	b	2

$$r(R2) =$$

A3	A4
x	1
b	2



$$R4 \div R2 =$$

A1	A2
a	1
b	2

$$R4 \div R2 = \underbrace{\prod_{A1,A2}(R4)}_{R1} - \underbrace{\prod_{A1,A2} \left(\underbrace{\prod_{A1,A2}(R4)}_{R1} \times R2 \right)}_{R5} - R4$$

$\underbrace{\hspace{10em}}_{R6}$
 $\underbrace{\hspace{15em}}_{R7}$

$$\prod_{A1,A2}(R4) =$$

$$= R1 =$$

A1	A2
a	1
b	2
b	1
c	3

$$R1 \times R2 =$$

$$= R5 =$$

A1	A2	A3	A4
a	1	x	1
a	1	b	2
b	2	x	1
b	2	b	2
b	1	x	1
b	1	b	2
c	3	x	1
c	3	b	2

$$R5 - R4 =$$

$$= R6 =$$

A1	A2	A3	A4
b	1	b	2
c	3	x	1

$$\prod_{A1,A2}(R6) =$$

$$= R7 =$$

A1	A2
b	1
c	3

$$R1 - R7 =$$

A1	A2
a	1
b	2

álgebra relacional: operadores JOIN (1)

(h) θ -Combinación, o θ -JOIN \bowtie_{θ} de dos relaciones R1 y R2

donde θ es un predicado de selección que relaciona atributos de R1 y R2

$R1 \bowtie_{\theta} R2 \equiv$ relación formada por todas las tuplas obtenidas concatenando (*combinando*) a una tupla de R1 otra de R2, y que verifican el predicado θ

si k y l representan dos atributos cualquiera de R1 y R2, respectivamente, sobre los que se define θ ,

$R1 \bowtie_{\theta} R2 = R'$, con $r'(R') = \{ \langle v_{i1}, \dots, v_{in1}, v_{j1}, \dots, v_{jn2} \rangle \mid \forall i, j \langle v_{i1}, \dots, v_{in1} \rangle \in r1(R1), \langle v_{j1}, \dots, v_{jn2} \rangle \in r2(R2), \text{ y se verifica } v_{ik} \theta v_{jl} \}$

➡ el grado de R' es $n' = n1 + n2$, y su cardinalidad $m' \leq m1 * m2$

Obsérvese que: $R1 \bowtie_{\theta} R2 = \sigma_{\theta}(R1 \times R2)$

álgebra relacional: operadores JOIN (1)

Equi-JOIN $\bowtie_{=}$ de dos relaciones R1 y R2

$R1 \bowtie_{=} R2 \equiv R1 \bowtie_{\theta} R2$, con θ el operador $=$, definido sobre los atributos comunes de R1 y R2

JOIN Natural \bowtie de dos relaciones R1 y R2

$R1 \bowtie R2 \equiv R1 \bowtie_{=} R2$, seguido de la eliminación de atributos repetidos
 \equiv *equi-JOIN*, seguido de proyección sobre la unión de los atributos de R1 y R2

➡ *Es uno de los operadores más usados, ya que permite completar la información de una tabla con la de otra, referenciada en la primera mediante una clave ajena*

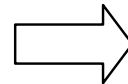
álgebra relacional: ejemplo de operaciones JOIN

$$r(R8) =$$

A1	A2	A3
a	1	x
a	1	y
b	2	x
b	2	z
b	1	s
c	3	y

$$r(R9) =$$

A3	A4
x	7
y	9



$$R8 \bowtie R9 =$$

A1	A2	A3	A4
a	1	x	7
a	1	y	9
b	2	x	7
c	3	y	9

$$R8 \bowtie R9 = \prod_{A1, A2, A3, A4} (\sigma_{R8.A3 = R9.A3} (R8 \times R9))$$

$$R8 \bowtie_{4*A2 > A4} R9 =$$

A1	A2	A3	A3	A4
b	2	x	x	7
b	2	z	x	7
c	3	y	x	7
c	3	y	y	9

$$R8 \bowtie_{4*A2 > A4} R9 = \sigma_{4*A2 > A4} (R8 \times R9)$$

$$R8 \bowtie R9 =$$

A1	A2	A3	A3	A4
a	1	x	x	7
a	1	y	y	9
b	2	x	x	7
c	3	y	y	9

$$R8 \bowtie R9 = \sigma_{R8.A3 = R9.A3} (R8 \times R9)$$

álgebra relacional: tratamiento de los valores nulos

el *valor nulo* (o *ausente*) es consecuencia de

- ✓ *valores desconocidos de atributos*
- ✓ *nuevos atributos añadidos a una relación*
- ✓ *atributos inaplicables a una tupla concreta*

➔ Su tratamiento complica las operaciones y obliga a la **redefinición de los operadores** (aritméticos, comparación, lógicos, relacionales, estadísticos, etc.)

↳ *No existe una total unidad de criterio en las implementaciones de SGBD*
 ⇒ *hay que asegurarse al implementar sobre un SGBD concreto*

➔ **los valores nulos no son comparables** (*un valor nulo no es ni igual ni distinto de otro valor*)

➔ el operador **ES-NULO** (v) devuelve verdad si v es el valor nulo, y falso en caso contrario

lógica trivaluada

AND	V	Q	F
V	V	Q	F
Q	Q	Q	F
F	F	F	F

OR	V	Q	F
V	V	V	V
Q	V	Q	Q
F	V	Q	F

	NOT
V	F
Q	Q
F	V

la lógica cuatrivaluada usa:
verdad (V), falso (F),
desconocido aplicable (A),
desconocido no-aplicable (I)

5.4 Interrogación de B.D. con álgebra relacional extendida

para diseñar consultas a una B.D. R. se añaden otros operadores (*álgebra relacional extendida*)

- *simplificar expresiones complejas*
- *permitir tratamientos especiales (aritméticos, estadísticos, etc.)*

Asignación

← ò :=

$R1 := R2 \equiv$ asocia a la relación R1 la ocurrencia correspondiente a la expresión relacional R2

se puede utilizar para el renombrado de atributos

Agrupación

AGRUPAR

una relación R respecto a un subconjunto de atributos X

AGRUPAR $F(R, X) \equiv$ agrupa las tuplas de R según los valores de los atributos X, y aplica a cada uno de los grupos la lista de funciones de agregación F

el resultado consta de los atributos X, más uno por cada una de las funciones F, y tantas tuplas como “grupos” se hayan obtenido (= cardinalidad de $\prod_X(R)$)

*funciones de agregación típicas: **CONTAR, SUMA, MEDIA, MAX, MIN, etc.***

ejemplo de operaciones de agrupamiento

$r(R) =$

A1	A2	A3
a	1	x
a	1	y
b	2	x
b	2	z
b	1	s
c	3	y

$R'(A2, N) := \text{AGRUPAR}_{\text{contar}}(R, A2) \longrightarrow R' =$

A2	N
1	3
2	2
3	1

$R'(A3, S) := \text{AGRUPAR}_{\text{suma}(A2)}(R, A3) \longrightarrow R' =$

A3	S
x	3
y	4
s	1
z	2

☛ *no se suelen eliminar los duplicados antes de aplicar la función de agregación*

☛ *¡¡ Pb. tratamiento de **nulos** !! \longrightarrow no se suelen contar*

extensión de atributos y operadores de JOIN externo

Extensión de atributos **EXTENDER** *una relación R con un conjunto de atributos X*

EXTENDER (R, X) \equiv añade a R los atributos especificados por X (pares atributo-dominio)

JOIN externo izquierdo, derecho, o completo $\overset{+}{\bowtie}$ \bowtie^+ $\overset{+}{\bowtie}^+$ *left, right, o full outer*

$R1 \overset{+}{\bowtie} R2$ \equiv JOIN natural de R1 y R2 completado con las tuplas de R1 que no tienen correspondencia en R2 (dichos atributos de R2 tomarán el valor nulo)

$R1 \bowtie^+ R2$ \equiv JOIN natural de R1 y R2 completado con las tuplas de R2 que no tienen correspondencia en R1 (dichos atributos de R1 tomarán el valor nulo)

$R1 \overset{+}{\bowtie}^+ R2$ \equiv JOIN natural de R1 y R2 completado con las tuplas de R1 que no tienen correspondencia en R2 (dichos atributos de R2 tomarán el valor nulo), y con las tuplas de R2 que no tienen correspondencia en R1 (serán nulos)

ejemplo de operaciones de JOIN externo

$$r(R1) =$$

A1	A2	A3
a	1	x
a	1	y
b	2	x
b	2	z
b	1	s
c	3	y

$$r(R2) =$$

A3	A4
x	7
y	9

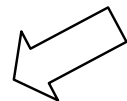
$$r(R3) =$$

A3	A4
x	7
y	9
t	6



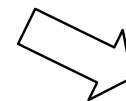
$$R1 \overset{+}{\bowtie} R2 =$$

A1	A2	A3	A4
a	1	x	7
a	1	y	9
b	2	x	7
b	2	z	
b	1	s	
c	3	y	9



$$R1 \overset{+}{\bowtie} R3 =$$

A1	A2	A3	A4
a	1	x	7
a	1	y	9
b	2	x	7
c	3	y	9
		t	6



$$R1 \overset{+}{\bowtie} R3 =$$

A1	A2	A3	A4
a	1	x	7
a	1	y	9
b	2	x	7
b	2	z	
b	1	s	
c	3	y	9
		t	6

otros operadores de álgebra relacional extendida. Ejemplos

cierre transitivo **CT** de R sobre un subconjunto de atributos compatibles X

CT (R, X) \equiv relación obtenida aplicando la transitividad sobre X hasta la saturación

unión externa \cup^+ devuelve relación con todos los atributos . . .

para el diseño de *consultas complejas* \rightarrow *descomponer* en pasos intermedios

ejemplo: Considérese el siguiente esquema (simplificado) de una B.D. relacional

Pieza (clvPieza: entero; nombPieza, color: tpNombre);

Proveedor (clvProv: entero; nombProv: tpNombre);

suministrar (clvProv: entero; clvPieza: entero)

clvProv clave ajena de Proveedor; clvPieza clave ajena de Pieza;

ejemplos de consulta a una B.D. con álgebra relacional (1)

① *Piezas de color 'verde'*

$$R1 = \sigma_{color = 'verde'} (Pieza)$$

② *Nombre de los proveedores que suministran al menos una pieza de color 'verde'*

$$R2 = \Pi_{nombProv} (Proveedor \bowtie Suministrar \bowtie R1)$$

③ *Nombre de los proveedores que no suministran "tuercas"*

$$R3 = \sigma_{nombPieza = 'tuerca'} (Pieza) \equiv \text{piezas que son tuercas}$$

$$R4 = \Pi_{clvProv} (Suministrar \bowtie R3) \equiv \text{proveedores de tuercas}$$

$$R5 = \Pi_{clvProv} (Proveedor) - R4 \equiv \text{proveedores que no suministran tuercas}$$

$$R6 = \Pi_{nombProv} (Proveedor \bowtie R5) \equiv \text{nombre de proveedores que no suministran tuercas}$$

o también

$$R5 = \Pi_{nombProv} (Proveedor \bowtie (\Pi_{clvProv} (Proveedor) - \Pi_{clvProv} (Suministrar \bowtie R3)))$$

ejemplos de consulta a una B.D. con álgebra relacional (2)

④ Clave, nombre, y suministradores de las piezas existentes

a) si todas tienen algún suministrador (faltaría la restricción en el esquema relacional):

$$R6 = \prod_{clvPieza, nombPieza, clvProv} (Pieza \bowtie Suministrar)$$

b) si puede haber piezas que no tienen suministrador:

$$R7 = \prod_{clvPieza, nombPieza, clvProv} (Pieza \overset{\dagger}{\bowtie} Suministrar)$$

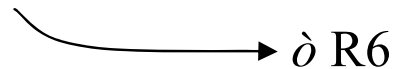
o también

$$R8 = \prod_{clvPieza, nombPieza} (Pieza \bowtie Suministrar) \equiv \text{piezas que tienen proveedor}$$

$$R9 = \prod_{clvPieza, nombPieza} (Pieza) - R8 \equiv \text{piezas que no son suministradas por nadie}$$

$$R7 = (R8 \bowtie Suministrar) \cup \mathbf{EXTENDER} (R9, clvProv: entero)$$

o también

 $\rightarrow \text{ò } R6$

$$R9 = \prod_{clvPieza, nombPieza} (Pieza) - \prod_{clvPieza, nombPieza} (R6) \equiv \text{piezas que no son suministradas}$$

$$R7 = R6 \cup \mathbf{EXTENDER} (R9, clvProv: entero)$$

ejemplos de consulta a una B.D. con álgebra relacional (3)

⑤ *Nombre de los proveedores que suministran al menos todas las piezas suministradas por el proveedor de clave 25*

$R1 = \prod_{clvPieza} (\sigma_{clvProv = 25} (\text{Suministrar})) \equiv \text{piezas suministradas por el proveedor 25}$

$R2 = \prod_{clvProv} (\text{Proveedor}) \times R1 \equiv \text{suministros si todos los proveedores suministrasen R1}$

$R3 = R2 - \text{Suministrar} \equiv \text{suministros de R1 que no se realizan}$

$R4 = \prod_{clvProv} (R3) \equiv \text{proveedores que no suministran todas las piezas de R1}$

$R5 = \prod_{clvProv} (\text{Suministrar}) - R4 \equiv \text{suministradores de todas las piezas de R1}$

$R6 = \prod_{nombProv} (\text{Proveedor} \bowtie R5)$

o también

ejemplos de consulta a una B.D. con álgebra relacional (4)

- ⑤ *Nombre de los proveedores que suministran al menos todas las piezas suministradas por el proveedor de clave 25*

$$R1 = \prod_{clvPieza} (\sigma_{clvProv = 25} (\text{Suministrar})) \equiv \text{piezas suministradas por el proveedor 25}$$

$$R5 = \text{Suministrar} \div R1 \equiv \text{suministradores de todas las piezas de R1}$$

$$R6 = \prod_{nombProv} (\text{Proveedor} \bowtie R5)$$

- ⑥ *Nombre de los proveedores que suministran 'tuercas' y 'tornillos'*

$$R1 = \prod_{clvPieza} (\sigma_{nombPieza = 'tuerca'} (\text{Pieza})) \equiv \text{piezas que son tuercas}$$

$$R2 = \prod_{clvPieza} (\sigma_{nombPieza = 'tornillo'} (\text{Pieza})) \equiv \text{piezas que son tornillos}$$

$$R3 = \prod_{clvProv} (\text{Suministrar} \bowtie R1) \equiv \text{proveedores de tuercas}$$

$$R4 = \prod_{clvProv} (\text{Suministrar} \bowtie R2) \equiv \text{proveedores de tornillos}$$

$$R5 = \prod_{nomProv} (\text{Proveedor} \bowtie (R3 \cap R4))$$

ejemplos de consulta a una B.D. con álgebra relacional (5)

⑥ *Nombre de los proveedores que suministran 'tuercas' y 'tornillos'*

$$R1 = R2 = \text{Suministrar} \bowtie \text{Pieza} \equiv \text{piezas y sus suministradores}$$

$$R3(\text{clvProv}) = \prod_{R1.\text{clvProv}} (\sigma_p (R1 \times R2)) \equiv \text{proveedores que suministran tuercas y tornillos}$$

Siendo $p = (R1.\text{clvProv} = R2.\text{clvProv}) \text{ AND } (R1.\text{nombPieza} = \text{'tuerca'}) \text{ AND } (R2.\text{nombPieza} = \text{'tornillo'})$

$$R5 = \prod_{\text{nomProv}} (\text{Proveedor} \bowtie R3)$$

⑦ *Clave, nombre, y n° de piezas que suministra cada proveedor*

$$R1(\text{clvProv}, \text{numPiezas}) := \text{AGRUPAR}_{\text{contar}(\text{clvPieza})}(\text{Suministrar}, \text{clvProv}) \equiv$$

└───────────┬───────────> *de los proveedores que suministran piezas*

$$R2 = \prod_{\text{clvProv}, \text{nombProv}, \text{numPiezas}} (\text{Proveedor} \overset{+}{\bowtie})$$

ejemplos de consulta a una B.D. con álgebra relacional (6)

- ⑧ *Clave de las piezas que son suministradas por más de un proveedor*
- ⑨ *Clave de las piezas que a lo sumo tienen un proveedor*
- ⑩ *Clave de los proveedores que sólo suministran un tipo de piezas*
- ⑪ *Clave de los proveedores que suministran alguna pieza en exclusiva (sólo ellos)*
- ⑫ *Clave de los proveedores “prescindibles”, pues no suministran ninguna pieza en exclusiva*
- ⑬ *Parejas (sin repetir ninguna) de proveedores que comparten el suministro de alguna pieza*
- ⑭ *Proveedor(es) que más piezas suministra del mismo tipo (tuercas, tornillos, etc.)*
- ⑮ *Parejas (sin repetir ninguna) de proveedores suministran exactamente las mismas piezas*
- ⑯ *Clave de los proveedores que suministran piezas de todos los colores*
- ⑰ *Clave de los proveedores que suministran más de dos piezas*
- ⑱ *Pares de suministradores (S_1, S_2) tales que S_1 suministra todas las piezas que suministra S_2*
- ⑲ *•••••*