

IV ÁLGEBRA RELACIONAL

Codd propuso tres lenguajes de especificación para el modelo relacional como base teórica de cualquier lenguaje que quisiera cumplir con los requisitos formales del modelo. Estos lenguajes no pueden ser explotados comercialmente, al menos tal y como los definió Codd, porque adolecen de falta de operadores: carecen de operadores aritméticos simples (sumas, restas, etc.), o de manipulación de cadenas de caracteres, por poner dos ejemplos "escandalosos". El propósito de estos lenguajes no es el de definir y manejar bases de datos relaciones, tan sólo constituyen una declaración de los mínimos requeridos para cualquier lenguaje de manipulación de datos que se quiera etiquetar a sí mismo como "relacional". En otras palabras, cualquier lenguaje de manipulación y definición de datos en bases de datos relacionales ha de poseer la potencia suficiente como para "hacer", como mínimo, lo que pueden "hacer" los lenguajes de Codd.

El primero de ellos, al menos por el orden en el que se van a introducir en esta asignatura, es el *álgebra relacional*. Recibe este nombre precisamente por su carácter algebraico: incluye un conjunto de operadores (ocho, concretamente) cuyos operandos son relaciones y el resultado de la operación es otra relación, del mismo modo que cuando sumamos dos enteros obtenemos otro número entero.

En primer lugar se definirán una serie de conceptos necesarios para definir los operadores del álgebra relacional y para, finalmente, trabajar con ellos. A continuación se describirán, uno por uno, los ocho operadores propuestos por Codd:

Unión	\cup	Selección	donde
Intersección	\cap	Proyección	[]
Diferencia	$-$	Concatenación Natural	∞
Producto Cartesiano	\times	División	\div

IV1. Conceptos previos.

Al describir las propiedades de cada operador se van a utilizar una serie de términos que debemos definir previamente.

En primer lugar se presentará una adaptación del concepto de relación matemática en la que se vuelve a hacer uso de la ordenación de las componentes de una tupla. El resto, son expresiones o reformulaciones de conceptos ya presentes en la definición del modelo.

Los conceptos a definir son:

- Relación
- Esquema de relación
- Nombres Cualificados de Atributo
- Alias de una relación
- Relación Nominada
- Relación Derivada
- Relaciones Compatibles
- Operación conmutativa
- Operación asociativa

relación

El AR hace uso del orden de las componentes de las tuplas para definir operadores y propiedades de los operadores. En realidad, se trata de retomar la definición original de la relación matemática como el subconjunto de un producto cartesiano de n dominios, de tal forma que las tuplas resultado de ese producto cumplan y cumplen que

Las tuplas son listas de valores (conjunto ordenado) tal que el i -ésimo valor pertenece al i -ésimo dominio.

Vamos a combinar la definición anterior de tupla con la adaptación que en su momento introdujimos a la relación matemática para adecuarla al objetivo final que es una base de datos. Utilizaremos al mismo tiempo los nombres de atributos y el orden de las componentes en una tupla:

El conjunto de nombres de atributos es un conjunto ordenado.

Las tuplas son listas de valores (conjunto ordenado) tal que el i -ésimo valor pertenece al i -ésimo dominio asociado al i -ésimo nombre de atributo.

A partir de ahora, los operadores pueden utilizar tanto el nombre simbólico de un atributo como su orden dentro de la tupla.

esquema de relación

Es la descripción formal de la relación con sus atributos y dominios asociados. En realidad se aplica únicamente a las relaciones nominadas, aquellas descritas en el esquema lógico relacional.

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

donde: R es el nombre de la relación

A_i es el nombre del atributo

D_i es el nombre del dominio asociado a A_i

nombres cualificados de atributo

Es, por decirlo así, el nombre completo de un atributo, por ejemplo $R.A_i$, el atributo A_i de la relación R . Su uso evita la ambigüedad de dos atributos en dos tablas distintas con el mismo nombre.

En general, nos referimos a los atributos por su nombre sin especificar la relación a la que pertenecen. No obstante, es habitual que en distintas relaciones, y sobre todo en las relaciones derivadas (los resultados de operar con relaciones nominadas), nos podamos encontrar nombres de atributo coincidentes en relaciones distintas. La forma de diferenciar unos de otros es utilizar los nombres cualificados: “*alumno.nombre*”, “*asignatura.nombre*”.

En definitiva, se pueden utilizar indistintamente, siempre y cuando no se produzcan ambigüedades, las dos formas ya conocidas de referirse a un atributo:

- nombre cualificado: $R.A_i$
- nombre no cualificado: A_i

alias de una relación

Nombre alternativo para una relación. Dada una relación R se define un alias mediante la declaración:

definir alias S para R

Entonces la relación puede referenciarse tanto por R como por S , y los nombres cualificados de atributos $R.A_i$ o $S.A_i$.

relación nominada

Es toda relación definida en el esquema lógico relacional. En otras palabras, las que constituyen nuestra base de datos.

relación derivada

Es aquella que se obtiene como resultado de una expresión del Álgebra Relacional.

Una relación derivada no tiene nombre ni alias. Así pues, los nombres de los atributos de ésta se obtendrán a partir de los nombres cualificados de atributos de las relaciones operando, y si existe ambigüedad se utilizarán los alias.

Las reglas que rigen en los operadores para la asignación de nombres a los atributos de relaciones derivadas se verán con cada uno de ellos.

relaciones compatibles

Dos relaciones son compatibles si el grado de ambas es el mismo y los dominios asociados a los i-ésimos atributos de cada una son iguales.

$$R(A1:D1, A2:D2, \dots, An:Dn)$$

$$S(B1:E1, B2:E2, \dots, Bm:Em)$$

R y S son compatibles si y sólo si:

- 1) $n = m$
- 2) $\forall i \, Di = Ei \, (1 \leq i \leq n)$

Dicho de otra forma, el número de atributos ha de ser el mismo en ambas relaciones y, además, los dominios han de ser los mismos para atributos de la misma posición.

operación conmutativa

Un operación es conmutativa¹⁵ si se cumple que

$$A \otimes B = B \otimes A$$

operación asociativa

Una operación es asociativa si se cumple que

$$(A \otimes B) \otimes C = A \otimes (B \otimes C)$$

operadores

Los operadores del Álgebra Relacional (bajo el punto de vista de Codd) se dividen en dos grupos:

de la teoría de conjuntos

UNIÓN
INTERSECCIÓN
DIFERENCIA
PRODUCTO CARTESIANO

relacionales

SELECCIÓN
PROYECCIÓN
DIVISIÓN
CONCATENACIÓN NATURAL

¹⁵En la descripción de los operadores, algunos se dice que cumplen la propiedad conmutativa, refiriéndonos exclusivamente al conjunto de tuplas de la relación derivada, y no al conjunto de atributos de dicha relación resultado, cuyos nombres cualificados dependen del orden de las relaciones operando.

Como el resultado de una expresión en Álgebra Relacional es otra relación, ésta puede participar como operando a su vez, permitiendo la construcción de expresiones anidadas.

$R \otimes S = D_1$ $D_1 \otimes O = D_2$	$R \otimes S \otimes O = D_2$
--	-------------------------------

La única regla de precedencia entre operadores es la utilización de paréntesis, en cualquier caso las expresiones se evalúan de izquierda a derecha de tal forma que el resultado de una operación es el operando de la siguiente operación a su derecha:

$R \otimes S = D_1$ $O \otimes D_1 = D_2$	$O \otimes (R \otimes S) = D_2$
--	---------------------------------

Otra clasificación se basa en la propia definición de los operadores: Son **operadores básicos** (o primitivos) la *unión*, *diferencia*, *producto cartesiano*, *selección* y *proyección*, y **operadores derivados** la *intersección*, la *división* y la *concatenación natural* ya se definen a partir de la combinación de varios operadores básicos.

Dicho de otra forma, los cinco operadores básicos definen la potencia expresiva del lenguaje, con ellos se pueden realizar todas las operaciones que permite el álgebra relacional, mientras que los tres derivados se pueden ver como “atajos” o “resúmenes” de varias operaciones básicas reunidas en un único operador.

IV2. definición informal de los operadores

En primer lugar, se pretende dar una visión de cómo operan y que resultado obtienen los operadores antes mencionados. Posteriormente, se dará la definición formal de todos ellos como referencia del lenguaje.

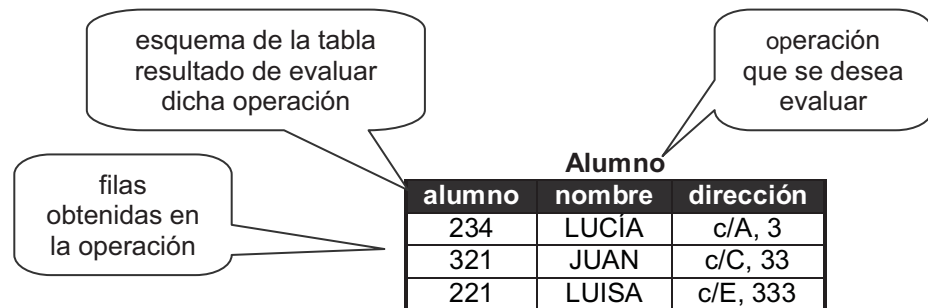
Cuando trabajamos con una base de datos relacional, si de manipulación de datos estamos hablando, la recuperación de información desde las tablas es un proceso habitual y necesario.

En álgebra relacional, si queremos recuperar el contenido completo de una relación, por ejemplo todos los alumnos almacenados en mi base de datos, una vez identificada la tabla de la que extraer los datos, la tabla alumno, la evaluación del nombre de la relación nos devuelve todas sus filas:

Alumno		
alumno	nombre	dirección
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333

```
select *
from alumno
```

Si a un intérprete de AR teórico le ordenamos simplemente “alumno”, el resultado es la tabla de 3 filas anterior. Por razones de claridad los ejemplos incluyen la operación, el esquema de la relación resultado y las filas obtenidas, según se detalla en el gráfico siguiente.



A partir de ahora, los ejemplos muestran el contenido de las tablas utilizadas y el resultado de cada uno de los operadores. También se darán las expresiones correspondientes en SQL como ayuda para la comprensión de cada operación.

selección

En muchas ocasiones sólo nos interesan algunos individuos de una relación, aquellos que poseen unas determinadas características: “clientes de la provincia de Alicante”, “artículos que cuestan mas de 500 euros”.

El operador selección (“**donde condición**”) recupera tuplas de una relación que cumplen una determinada condición.

Alumno		
alumno	nombre	dirección
577	YÉNIFER	c/C, 33
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333

Alumno donde dirección = 'c/C, 33'

alumno	nombre	dirección
577	YÉNIFER	c/C, 33
321	JUAN	c/C, 33

```
select *
from alumno
where dirección = 'c/C, 33'
```

proyección

Si la selección recupera filas, la proyección ([*columna(s)*]) recupera atributos.

Alumno		
alumno	nombre	dirección
577	YÉNIFER	c/C, 33

234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333

Alumno [alumno, nombre]

alumno	nombre
577	YÉNIFER
234	LUCÍA
321	JUAN
221	LUISA

```
select alumno, nombre
from alumno
```

La selección me permite recuperar aquellos individuos que me interesan y la proyección elimina aquellos datos de los individuos que no necesito. Como ya se ha dicho, se pueden realizar varias operaciones consecutivas, como se muestra a continuación:

Alumno donde dirección = 'c/C, 33' [alumno, nombre]

alumno	nombre
577	YÉNIFER
321	JUAN

```
select alumno, nombre
from alumno
where dirección = 'c/C, 33'
```

La proyección podría eliminar la o las columnas definidas como clave candidata y podría dar lugar a duplicados entre las filas (o más bien nuestra percepción de la relación nos puede inducir a pensar que se puedan mostrar filas iguales). Como el resultado de esta operación es también una tabla, ésta está sujeta a las mismas restricciones del modelo que cualquier otra. Se puede pensar que el AR realiza la proyección en dos pasos: la proyección como tal y una eliminación de duplicados, si es que se producen.

Si proyectáramos sobre la columna dirección, la tabla alumno contiene dos filas con los mismos valores. Sin embargo, el resultado de tal operación es el siguiente:

Alumno [dirección]

dirección
c/C, 33
c/A, 3
c/E, 333

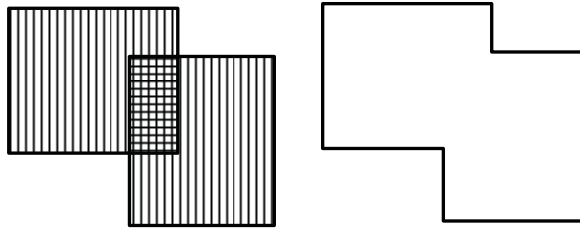
```
select dirección16
from alumno
```

¹⁶ En realidad, es habitual que los SGBD relacionales y su implementación de SQL, ante expresiones como esta devuelvan filas duplicadas. El modificador *distinct* realiza ese segundo paso al que hemos hecho referencia en el álgebra relacional. A partir de ahora, aunque no aparezca en los ejemplos, se entenderá que todas las órdenes *select* incluyen este modificador: "select **distinct** dirección from alumno where dirección = 'c/C, 33' "

Obviamente, esto es aplicable a cualquier operación que hagamos puesto que el resultado siempre va a ser una tabla derivada.

unión

La unión de dos relaciones da como resultado otra relación que contiene las tuplas de las dos. Como en una relación no existen tuplas duplicadas (por ser un conjunto, por estar definida por un producto cartesiano), sería más exacto describir la unión de dos relaciones A y B como otra relación C que contiene las tuplas de A que no están en B, las tuplas de B que no están en A, y las tuplas que están en A y B a la vez.



Alumno		
alumno	nombre	dirección
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333

Profesor		
profesor	nombre	dirección
522	JOSÉ	c/F, 32
778	EVA	c/F, 51
221	LUISA	c/E, 333

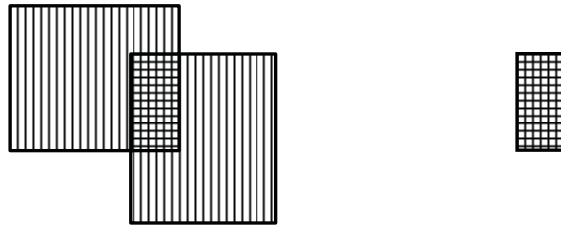
Alumno \cup Profesor		
alumno	nombre	dirección
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333
522	JOSÉ	c/F, 32
778	EVA	c/F, 51

```
select * from alumno
union
select * from profesor
```

La unión sólo se puede realizar si A y B son compatibles y, evidentemente, es conmutativa y asociativa (ver las definiciones anteriores)

intersección

La intersección de dos relaciones da como resultado otra relación que contiene las tuplas comunes a las dos primeras.

**Alumno**

alumno	nombre	dirección
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333

Profesor

profesor	nombre	dirección
522	JOSÉ	c/F, 32
778	EVA	c/F, 51
221	LUISA	c/E, 333

Alumno \cap Profesor

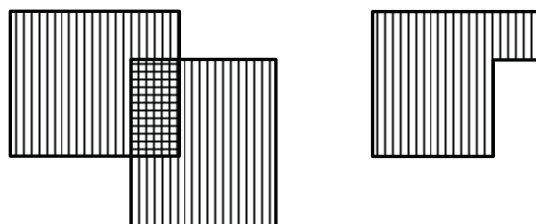
alumno	nombre	dirección
221	LUISA	c/E, 333

```
select * from alumno
intersect
select * from profesor
```

Al igual que la unión, la intersección sólo se puede realizar si A y B son compatibles y es conmutativa y asociativa.

diferencia

La diferencia de dos relaciones se define como aquellas tuplas que están en la primera pero no en la segunda. Nótese que este operador ya no es conmutativo, importa el orden de los operandos.



Alumno

alumno	nombre	dirección
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333

Profesor

profesor	nombre	dirección
522	JOSÉ	c/F, 32
778	EVA	c/F, 51
221	LUISA	c/E, 333

Alumno - Profesor

alumno	nombre	dirección
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33

```
select * from alumno
minus
select * from profesor
```

La diferencia sólo se puede realizar si A y B son compatibles.

producto cartesiano

Este operador se utiliza para generar todas las posibles combinaciones de las tuplas de una relación con todas y cada una de las tuplas de otra relación; se obtienen todas las combinaciones posibles de filas entre dos tablas.

Alumno

alumno	nombre	dirección
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333

Profesor

profesor	nombre	dirección
522	JOSÉ	c/F, 32
778	EVA	c/F, 51
221	LUISA	c/E, 333

Alumno × Profesor

alumno	nombre	dirección	profesor	nombre	dirección
234	LUCÍA	c/A, 3	522	JOSÉ	c/F, 32
234	LUCÍA	c/A, 3	778	EVA	c/F, 51
234	LUCÍA	c/A, 3	221	LUISA	c/E, 333
321	JUAN	c/C, 33	522	JOSÉ	c/F, 32
321	JUAN	c/C, 33	778	EVA	c/F, 51
321	JUAN	c/C, 33	221	LUISA	c/E, 333
221	LUISA	c/E, 333	522	JOSÉ	c/F, 32
221	LUISA	c/E, 333	778	EVA	c/F, 51
221	LUISA	c/E, 333	221	LUISA	c/E, 333

```
select * from alumno, profesor
```

concatenación natural

Este operador se utiliza para combinar información de dos tablas que comparten un atributo común. En este caso quedará más claro un ejemplo en el que la base de datos refleje una relación entre, por ejemplo, profesores y departamentos a los que están adscritos. La relación entre los dos se representa por una columna en la tabla profesor que indica el código de departamento en el que trabaja.

La concatenación natural utiliza los atributos que tienen el mismo nombre y, por supuesto, están definidos sobre el mismo dominio.

Profesor				Departamento	
profesor	nombre	dirección	cod	cod	departamento
522	JOSÉ	c/F, 32	LSI	LSI	LENGUAJES
778	EVA	c/F, 51	LSI	CCIA	CIENCIAS
221	LUISA	c/E, 333	FI	FI	FILOLOGÍA

Profesor ∞ Departamento

profesor	nombre	dirección	cod	departamento
522	JOSÉ	c/F, 32	LSI	LENGUAJES
778	EVA	c/F, 51	LSI	LENGUAJES
221	LUISA	c/E, 333	FI	FILOLOGÍA

```
select p.*, d.departamento
from profesor p, departamento d
where p.cod = d.cod17
```

Si no existen 2 filas, una en cada tabla con ese valor idéntico que las une, el resultado de la concatenación es vacío (no devuelve ninguna tupla). En nuestro ejemplo sería el caso de que ningún profesor trabajara en ningún departamento (si la columna Profesor.cod almacenara nulos para todas las filas en un determinado estado de la base de datos).

El problema se presenta cuando más de una columna comparte tanto nombre como dominio: el operador intentará unir las filas que comparten los mismos valores en todos esos atributos. Cambiando un poco el ejemplo anterior,

¹⁷ El operador concatenación natural lleva implícita la condición de la cláusula *where*. En realidad, esta expresión en SQL es la descomposición en primitivas de la concatenación natural que se verá más adelante: un producto cartesiano más una selección y una proyección.

podemos pensar que el esquema de *Departamento* tiene como nombre de la columna de descripción *nombre* en vez de *departamento*. La concatenación natural sólo devolverá aquellos profesores y departamentos que cumplan que el profesor trabaja en él y que ambos tienen el mismo nombre.

Profesor				Departamento	
profesor	nombre	dirección	cod	cod	nombre
522	LENGUAJES	c/F, 32	LSI	LSI	LENGUAJES
778	EVA	c/F, 51	LSI	CCIA	CIENCIAS
221	LUISA	c/E, 333	FI	FI	FILOLOGÍA

Profesor \bowtie Departamento			
profesor	nombre	dirección	cod
522	LENGUAJES	c/F, 32	LSI

```
select p.*,d.departamento
from profesor p, departamento d
where p.cod = d.cod and p.nombre =
d.nombre
```

Cuando no se desea este comportamiento, si queremos que la unión de filas se realice únicamente por las columnas etiquetadas como *cod*, estamos obligados a utilizar una combinación de otros operadores. De hecho, la concatenación natural es uno de los operadores derivados, y se puede expresar en función de los operadores básicos selección, proyección y producto cartesiano.

división

Este operador es el de aplicación menos habitual, se utiliza para consultas del tipo “alumnos matriculados en **todas** las asignaturas”. Es, además, el operador con más restricciones a la hora de construir una expresión en AR. El dividendo debe tener más atributos que el divisor, dividendo y divisor han de compartir uno o varios atributos, éstos han de ser los últimos del dividendo y los únicos del divisor y estar en el mismo orden. Los atributos no comunes son la información que se obtiene de la división. En la práctica, sea necesario o no, se suele utilizar la proyección para asegurar que se cumplen todas estas reglas.

En nuestro ejemplo, “alumnos matriculados en **todas** las asignaturas”, usamos la tabla matriculado que es la que representa qué alumnos están matriculados en qué asignaturas. Todas las asignaturas son, evidentemente, las asignaturas almacenadas en la tabla asignatura¹⁸: el atributo común a ambas relaciones es “asignatura”, y la información que queremos obtener es “alumno” (que después se puede concatenar con la tabla Alumno para obtener su “nombre” y “dirección”).

¹⁸ En realidad, debemos identificar ese “todos” al qué se refiere, a qué tabla y a qué datos, puede no ser trivial: “alumnos matriculados en **todas las asignaturas de menos de 6 créditos**”, por ejemplo.

Alumno		
alumno	nombre	dirección
234	LUCÍA	c/A, 3
321	JUAN	c/C, 33
221	LUISA	c/E, 333

Asignatura	
asignatura	nombre
BD1	Bases1
BD2	Bases2
FP2	Prog2

Matriculado	
alumno	asignatura
234	BD1
234	BD2
234	FP2
321	BD1
321	FP2

Matriculado [alumno, asignatura] % (Asignatura [asignatura])

alumno

234

```
select alumno
from matriculado m
where not exists
(select * from asignatura a
 where not exists
 (select * from matriculado m2
  where m2.alumno = m.alumno
    and m2.asignatura =
      a.asignatura))19
```

La razón de que la división tenga unas reglas de composición tan estrictas está en que éste es otro de los operadores derivados, ya que se puede realizar la misma operación con una combinación de diferencias y proyecciones.

IV2.1.el problema de los nombres de los atributos en las relaciones derivadas

Cuando operamos en AR de dos tablas obtenemos una tercera tabla resultado. Esta tabla cumple todas las restricciones del modelo relacional y, por tanto, tiene un esquema de relación que es generado automáticamente. Dicho de otra forma, las relaciones derivadas también tienen nombre para sus atributos pero estos nombres se asignan directamente de los operandos y según el operador utilizado y el orden de tales operandos.

Por ejemplo, la unión, intersección y diferencia dan como resultado una tabla que tiene los mismos nombres de columna que la primera tabla operando.

Alumno (alumno, nombre, dirección)

Profesor (profesor, nombre, dirección)

Alumno \cup Profesor

T(Alumno.alumno, Alumno.nombre, Alumno.dirección)

¹⁹ La expresión "seleccionar aquellos alumnos que están matriculados en **todas** las asignaturas" se puede resolver en SQL si cambiamos "un poco" el enunciado: "seleccionar aquellos alumnos que cumplen que **no existe** ninguna asignatura en la que **no** estén matriculados"

Pero para la concatenación natural los nombres se asignan de diferente forma: todos los nombres de atributo de la primera relación más los no “comunes” de la segunda.

Profesor (profesor, nombre, dirección, cod)

Departamento (cod, descripción)

Profesor \bowtie Departamento

T(**Profesor**.profesor, **Profesor**.nombre,
Profesor.dirección, **Profesor**.cod,
Departamento.descripción)

El producto cartesiano produce tablas con todos los atributos de las dos relaciones operando, pero respetando el orden en que se ha operado:

Alumno (alumno, nombre, dirección)

Profesor (profesor, nombre, dirección)

Alumno \times Profesor

T(**Alumno**.alumno, **Alumno**.nombre,
Alumno.dirección, **Profesor**.profesor,
Profesor.nombre, **Profesor**.dirección)

La división produce una tabla con los atributos no comunes de la primera relación:

Matriculado [alumno, asignatura]

% (Asignatura [asignatura])

T(**Matriculado**.alumno)

Finalmente, la selección y la proyección, al ser operadores con un único operando, la tabla resultado tiene todos (si es una selección) o algunos (si es una proyección) de los atributos de la relación operando.

Alumno donde dirección='c/C, 33'

T(**Alumno**.alumno, **Alumno**.nombre, **Alumno**.dirección)

Alumno [dirección]

T(**Alumno**.dirección)

Es fundamental tener claro cuales son los atributos que resultan de cada operación ya que las expresiones del AR son una secuencia de operaciones cuyo resultado es el operando de la siguiente. La evaluación de las expresiones se realiza de izquierda a derecha salvo si se utilizan paréntesis que alteran el orden de evaluación. Las siguientes expresiones son correctas:

Alumno \times Profesor [Alumno.nombre]

Matriculado [alumno, asignatura]

% (Asignatura [asignatura])

Alumno donde dirección = 'c/C, 33' [alumno, nombre]

Pero las mostradas a continuación son incorrectas:

Alumno \times Profesor [nombre]

hay dos columnas “nombre” en la relación derivada.

Matriculado [alumno, asignatura] % Asignatura [asignatura]

al evaluar de izquierda a derecha la aplicación de los operadores sería proyección, división y, finalmente, proyección. Los atributos del divisor serían todos los de Asignatura, y no son consistentes con las restricciones del operador. Antes de dividir, se debería proyectar sobre Asignatura.

Alumno [alumno, nombre] donde dirección = 'c/C, 33'

si proyectamos antes de seleccionar estamos eliminando precisamente la columna dirección que es la que utiliza la selección en la condición de filtro.

IV2.2.operaciones primitivas.

Las únicas operaciones consideradas como primitivas son la *selección*, *proyección*, *unión*, *producto cartesiano*, y *diferencia*. Así, *concatenación natural*, *intersección* y *división* se pueden expresar en función de las primitivas mencionadas antes:

$$R \cap S = R - (R - S)$$

$$R \infty S = ((R \times S) \text{ DONDE } R.B_1=S.B_1 \text{ y... y } R.B_m=S.B_m) \\ [R.A_1, \dots, R.A_n, R.B_1, \dots, R.B_m, S.C_1, \dots, S.C_p]$$

donde los B_i son los atributos comunes a las dos relaciones, los A_i los no comunes de R y los C_i los no comunes de S.

$$R \div S = R[B] - ((R[B] \times S) - R)[B]$$

donde B es el conjunto de atributos no comunes de R.

IV2.3.uso del alias de una relación

Se puede definir un alias para una relación en cualquier ocasión pero su uso está más justificado por operaciones como la siguiente

Matriculado \times Matriculado

Un producto de una tabla por si misma es a veces necesario para ciertas consultas de conteo simple ("alumnos que se han matriculado de al menos 2 asignaturas"). El problema reside en que el esquema de la relación derivada sería:

$T(\text{Matriculado.alumno}, \text{Matriculado.asignatura},$
 $\text{Matriculado.alumno}, \text{Matriculado.asignatura}$)

Evidentemente, en una tabla no pueden existir dos columnas distintas con el mismo nombre (aunque sí en dos tablas distintas). La solución consiste en utilizar uno o dos alias para la relación Matriculado:

definir alias M para Matriculado
 Matriculado \times M

$T(\text{Matriculado.alumno}, \text{Matriculado.asignatura},$

IV3. Resumen de los operadores del Álgebra Relacional

Supongamos que R y S son dos relaciones.

selección

Selecciona aquellas tuplas que cumplen una determinada condición lógica.

Los nombres de los atributos de la relación derivada son los mismos que los de la relación operando.

proyección

Elimina los atributos no especificados.

Los nombres de atributo son los de la lista de parámetros

producto cartesiano

Es la combinación de todas y cada una de las tuplas de la primera relación con todas y cada una de las de la segunda.

Es asociativo y conmutativo, si no tenemos en cuenta el nombre y el orden de los atributos.

Los nombres de atributo son, en este orden, todos los de R y todos los de S.

concatenación natural

Se combinan aquellas tuplas de R y S cuyos valores coinciden para un grupo de atributos comunes. Por atributos comunes entendemos aquellos que tienen el mismo nombre y están definidos en el mismo dominio.

Es asociativo y conmutativo, si no tenemos en cuenta el nombre y el orden de los atributos.

Los nombres de atributo son, en este orden, todos los de R y todos los no comunes de S.

diferencia

Todas las tuplas que están en R pero no en S.

R y S han de ser compatibles.

Los nombres de atributo son todos los de R.

unión

Todas las tuplas que están en R o en S (o en ambas).

R y S han de ser compatibles.

Los nombres de atributo son todos los de R.

intersección

Todas las tuplas que están en R y en S a la vez.

R y S han de ser compatibles.

Los nombres de atributo son todos los de R.

división

Son las tuplas de R que están combinadas que están combinadas con todas las de S. En este caso se da que los atributos de S se encuentran también en R pero ordenados de una manera especial:

- a) el orden de los atributos comunes es igual en R y en S, y además,
- b) en R son los últimos (si "leemos" el esquema de R de izquierda a derecha)

Los nombres de atributo son todos los no comunes de R

IV4. ejemplos

El esquema que se muestra a continuación es el que se va a utilizar para resolver todas las consultas que se plantean a modo de ejemplo. Se especifican los dominios asociados a cada atributo puesto que las comparaciones entre atributos sólo se pueden realizar si sus dominios son idénticos.

ASIGNATURAS (cod_asg: *dCod*, nombre: *dNom*, curso: *dCur*, t:*dHora*, p:*dHora*, l:*dHora*)

ALUMNOS (exp: *dExp*, nombre: *dNom*, dir: *dDir*, ciudad: *dCiudad*, estudios: *dEstudios*)

NOTAS (exp: *dExp*, cod_asg: *dCod*, nota: *dNota*)²⁰

asignaturas

cod_asg	nombre	curso
BD	bases de datos	3
EIN	estructuras	2
AFO	análisis	2
FP	programación	1

alumnos

exp	nombre	estudios
1	pepe	cou
2	ana	cou
3	luisa	fp
4	juan	cou
5	pedro	cou
6	pilar	cou

notas

exp	cod_asg	nota
1	BD	7
1	EIN	5
1	FP	5
1	AFO	6
2	FP	5
2	EIN	6
3	FP	1

²⁰ Nótese que no se ha definido ninguna clave candidata ni ninguna clave ajena. El AR, no hace uso de las definiciones de clave candidata, clave ajena o valor no nulo. Los operadores de AR se basan en comparaciones de valores y, por tanto, la información necesaria y suficiente es la mostrada en este esquema. Por eso, claves ajenas, si se definen, y claves candidatas, definidas obligatoriamente, son aquí meramente una información útil a la hora de escribir una expresión concreta.

En condiciones normales el esquema de base de datos será conforme al modelo relacional, y esquemas de este tipo, incompletos y erróneos según el modelo relacional, no se volverán a proponer.