

Álgebra Relacional

Gabriel Chaves Benítez (gchaves@us.es)

Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

15-10-2025, Bases de Datos

Índice

- 1 Introducción
- 2 Operaciones básicas
- 3 Operaciones de la Teoría de Conjuntos
- 4 Operaciones derivadas
- 5 Otros
- 6 Bibliografía

Índice

- 1 **Introducción**
- 2 Operaciones básicas
- 3 Operaciones de la Teoría de Conjuntos
- 4 Operaciones derivadas
- 5 Otros
- 6 Bibliografía

Álgebra

Un álgebra es un sistema matemático constituido por

- **Operandos:** objetos (valores o variables) del sistema, a partir de los cuales se pueden construir nuevos objetos.
- **Operadores:** símbolos del sistema, que producen (denotan, identifican) nuevos objetos a partir de otros.

Álgebra relacional: *Álgebra en la cual...*

- Sus operandos son **relaciones** (instancias) o variables que representan relaciones.
- Sus operadores están diseñados para hacer la tareas más comunes que se necesitan para **manipular relaciones** en una base de datos.

Nota: el AR se puede utilizar como lenguaje de consulta, aunque en la práctica debe ser extendida para abarcar la mayoría de tareas reales con datos.

Motivación

- Un modelo de datos **debe incluir** un conjunto de **operaciones para manipular la base de datos**.
- El conjunto de operaciones básicas del modelo relacional es el **álgebra relacional**, que permite al usuario especificar las **peticiones fundamentales de recuperación**.
- El resultado de una recuperación es una **nueva relación**.
- Una *secuencia de operaciones* de álgebra relacional conforma una **expresión** de álgebra relacional, cuyo resultado será también una **nueva relación** que representa el producto obtenido de una consulta a la base de datos (o una petición de recuperación).

Relevancia

El álgebra relacional es importante por varias razones:

- Proporciona un fundamento formal para las operaciones del modelo relacional.
- Se utiliza como **base para la implementación** y optimización **de consultas** en los SGBD relacionales.
- Algunos de sus **conceptos** se han **incorporado** al **lenguaje** estándar de consultas **SQL**.

Índice

- 1 Introducción
- 2 Operaciones básicas**
- 3 Operaciones de la Teoría de Conjuntos
- 4 Operaciones derivadas
- 5 Otros
- 6 Bibliografía

Operación SELECCIÓN/RESTRICCIÓN

- Se emplea para seleccionar un **subconjunto de las tuplas** de una relación que **satisfacen una condición** de selección.
- Se puede considerar esta operación como un *filtro* que mantiene sólo las tuplas que satisfacen la condición establecida.
- Puede visualizarse también como una ***partición horizontal*** de la relación.

$\sigma_{\langle \text{CondicionDeSeleccion} \rangle}(R)$ o bien `R WHERE condicion`

donde R es una expresión del álgebra relacional (nombre de una relación, variable o resultado de operación previa).

Operación SELECCIÓN/RESTRICCIÓN

Expresión de Condiciones

Dentro de la condición de selección podemos incorporar distintos tipos de operadores:

- Operadores relacionales ($>$, $<$, \leq , \geq , $=$): $x > 3$
- Operadores lógicos (AND, OR, NOT): $x > 3$ AND $y = \text{'Hola'}$
- Paréntesis:
($x > 3$ AND $y = \text{'Hola'}$) OR ($x \leq 3$ AND $y = \text{'Adiós'}$)

Ejemplo: Relación EMPLEADO

Relación EMPLEADO

Nombre	Apell1	Apell2	Dni	Dirección	Sex	Sueldo	NRP	Dep
José	Pérez	Pérez	123456789	Eloy 1,98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	Avda. Ríos, 9	H	40000	888665555	5
Alicia	Jiménez	Celaya	999887777	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	Antón, 6	M	25000	333445555	5
Luis	Pajares	Morera	987987987	Enebros, 90	H	25000	987654321	4
Eduardo	Ochoa	Paredes	888665555	Las Peñas, 1	H	55000	NULL	1

Ejercicio: Recuperar los empleados del departamento 4 que ganen más de 25.000 euros y los del 5 que ganen más de 30.000.

Ejemplo: Relación EMPLEADO

Relación EMPLEADO

Nombre	Apell1	Apell2	Dni	Dirección	Sex	Sueldo	NRP	Dep
José	Pérez	Pérez	123456789	Eloy 1,98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	Avda. Ríos, 9	H	40000	888665555	5
Alicia	Jiménez	Celaya	999887777	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	Antón, 6	M	25000	333445555	5
Luis	Pajares	Morera	987987987	Enebras, 90	H	25000	987654321	4
Eduardo	Ochoa	Paredes	888665555	Las Peñas, 1	H	55000	NULL	1

Ejercicio: Recuperar los empleados del departamento 4 que ganen más de 25.000 euros y los del 5 que ganen más de 30.000.

$\sigma_{(Dep=4 \text{ AND } Sueldo > 25000) \text{ OR } (Dep=5 \text{ AND } sueldo > 30000)}(EMPLEADO)$

EMPLEADO WHERE (Dep=4 AND Sueldo>25000) OR (Dep=5 AND sueldo>30000)

Nombre	Apell1	Apell2	Dni	Dirección	Sex	Sueldo	NRP	Dep
Alberto	Campos	Sastre	333445555	Avda. Ríos, 9	H	40000	888665555	5
Juana	Sainz	Oreja	987654321	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	Portillo, s/n	H	38000	333445555	5

Operación PROYECCIÓN

- Si pensamos en una relación como en una tabla, la operación SELECCIÓN elige algunas de las filas de la tabla.
- PROYECCIÓN **devuelve** los datos de **ciertas columnas** de la tabla y **descarta otras**. Además, **elimina** resultados **duplicados** en la relación resultante.
- Si sólo estamos interesados en algunos atributos de una relación, usamos la operación PROYECCIÓN.
- Por consiguiente, el resultado de esta operación puede visualizarse como una partición vertical de la relación.

$\pi_{\langle \text{ListaDeAtributos} \rangle}(R)$ o bien $R[\text{ListaDeAtributos}]$

donde R es una expresión del álgebra relacional.

Ejemplo: Relación EMPLEADO

Relación EMPLEADO

Nombre	Apell1	Apell2	Dni	Dirección	Sex	Sueldo	NRP	Dep
José	Pérez	Pérez	123456789	Eloy 1,98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	Avda. Ríos, 9	H	40000	888665555	5
Alicia	Jiménez	Celaya	999887777	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	Antón, 6	M	25000	333445555	5
Luis	Pajares	Morera	987987987	Enebros, 90	H	25000	987654321	4
Eduardo	Ochoa	Paredes	888665555	Las Peñas, 1	H	55000	NULL	1

Ejercicio: Obtener el DNI y el sueldo de los empleados.

Ejemplo: Relación EMPLEADO

Relación EMPLEADO

Nombre	Apell1	Apell2	Dni	Dirección	Sex	Sueldo	NRP	Dep
José	Pérez	Pérez	123456789	Eloy 1,98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	Avda. Ríos, 9	H	40000	888665555	5
Alicia	Jiménez	Celaya	999887777	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	Antón, 6	M	25000	333445555	5
Luis	Pajares	Morera	987987987	Enebros, 90	H	25000	987654321	4
Eduardo	Ochoa	Paredes	888665555	Las Peñas, 1	H	55000	NULL	1

Ejercicio: Obtener el DNI y el sueldo de los empleados.

$$\pi_{Dni, Sueldo}(EMPLEADO)$$

EMPLEADO[Dni, Sueldo]

Dni	Sueldo
123456789	30000
333445555	40000
999887777	25000
987654321	43000
666884444	38000
453453453	25000
987987987	25000
888665555	55000

Operación RENOMBRADO

- En general, podemos aplicar varias operaciones de AR una tras otra. Para ello, tenemos varias opciones:
 - Escribir las operaciones como **una única expresión** de álgebra relacional **anidando** dichas **operaciones**
 - Aplicar expresiones más simples, usando **relaciones intermedias**. Para ello, debemos **asignar a una variable** cada una de las relaciones intermedias:

TEMP ← *expresion_relacional*

- Además, podemos **renombrar** campos A_1, \dots, A_n de una relación R de grado n a nuevos nombres B_1, \dots, B_n mediante el operador ρ :

$$\rho_{(B_1, \dots, B_n)}(R)$$

Operación de asignación y RENOMBRADO

Ejemplo

Podemos escribir la expresión anidada

$\pi_{Nombre, Apell1, Sueldo}(\sigma_{Dep=5}(EMPLEADO))$, y posteriormente renombrar sus campos, haciendo:

$$TEMP_1 \leftarrow \sigma_{Dep=5}(EMPLEADO)$$

$$TEMP_2 \leftarrow \pi_{Nombre, Apell1, Sueldo}(TEMP_1)$$

$$RESULTADO \leftarrow \rho_{(Nombre, Apellido, Salario)}(TEMP_2)$$

Índice

- 1 Introducción
- 2 Operaciones básicas
- 3 Operaciones de la Teoría de Conjuntos**
- 4 Operaciones derivadas
- 5 Otros
- 6 Bibliografía

Operaciones de la Teoría de Conjuntos

- Son las operaciones UNIÓN, INTERSECCIÓN y DIFERENCIA DE CONJUNTOS (UNION, INTERSECTION y MINUS).
- Son operaciones binarias, *i.e.*, reciben como entradas dos relaciones. Para poder aplicarlas, las relaciones de entrada deben ser de **unión compatible**.

Dos relaciones $R(A_1, A_2, \dots, A_n)$ y $S(B_1, B_2, \dots, B_n)$ se dice que son de unión compatible si tienen el mismo grado n y si $dom(A_i) = dom(B_i)$ para $1 \leq i \leq n$. En definitiva, deben tener el mismo número de atributos sobre los mismos dominios.

Nota: obsérvese que, aunque en el modelo relacional el orden no importa, en las operaciones de álgebra relacional es fundamental.

Unión, Intersección y Diferencia

Podemos definir estas operaciones sobre relaciones R y S como:

- **UNIÓN**, $R \cup S$ (o R UNION S): incluye **todas las tuplas** que están en R , en S o en ambas, eliminando duplicados.
- **INTERSECCIÓN**, $R \cap S$ (o R INTERSECTION S): incluye **aquellas tuplas que están a la vez en ambas** R y en S .
- **DIFERENCIA DE CONJUNTOS**, $R - S$ (o R MINUS S): incluye todas las tuplas que **están en R pero no en S** .

R		S	
X	N	X	N
a	1	a	2
a	2	b	4
b	3		

R UNION S		R INTERSECTION S		R MINUS S		S MINUS R	
X	N	X	N	X	N	X	N
a	1	a	2	a	1		
a	2					b	4
b	3						
b	4						

Producto Cartesiano

- Conocido también como PRODUCTO CRUZADO (CROSS PRODUCT) o CONCATENACIÓN CRUZADA (CROSS JOIN), que se identifica por el símbolo \times (o bien como R TIMES S).
- Se trata también de una operación de conjuntos binarios, aunque **no es necesario** que las relaciones en las que se aplica sean de **unión compatible**.
- Combina cada tupla de una relación con los de la otra.
 $R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$ es una relación con $n + m$ atributos $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$.

Producto Cartesiano

Resulta por tanto en la relación $Q(A_1, \dots, A_n, B_1, \dots, B_m)$, tal que:

- Q tiene 1 tupla por cada combinación de tuplas de R con S . Así, de R con n_R tuplas ($|R| = n_R$), y S con n_S tuplas, $R \times S$ tendrá $n_R \times n_S$ tuplas.
- Si hay dos atributos distintos con el mismo nombre x en R y S , se puede nombrar en Q como $R.x$ y $S.x$

R

X	N
a	1
a	2
b	3

S

X	M
a	2
b	4

T ← R TIMES S

R.X	N	S.X	M
a	1	a	2
a	1	b	4
a	2	a	2
a	2	b	4
b	3	a	2
b	3	b	4

Possible selección posterior:

T WHERE R.X = S.x

R.X	N	S.X	M
a	1	a	2
a	2	a	2
b	3	b	4

Ejemplo de Producto Cartesiano

CLIENTES

codcli	nombre	dirección	codpostal	codpue
333	Sos Carretero, Jesús	Mosen Compte, 14	12964	53596
336	Miguel Archilés, Ramón	Bernardo Mundina, 132-5	12652	07766
342	Pinel Huerta, Vicente	Francisco Sempere, 37-10	12112	07766
345	López Botella, Mauro	Avenida del Puerto, 20-1	12010	12309
348	Palau Martínez, Jorge	Raval de Sant Josep, 97-2	12003	12309
354	Murría Vinaiza, José	Ciudadela, 90-18	12003	12309
357	Huguet Peris, Juan Ángel	Calle Mestre Rodrigo, 7	12100	12309

PUEBLOS

codpue	nombre	codpro
07766	Burriana	12
12309	Castellón	12
17859	Enramona	12
46332	Soneja	12
53596	Vila-real	12

Ej: *obtener los nombres de los pueblos en los que hay clientes*

Ejemplo de Producto Cartesiano

$$TEMP_1 \leftarrow \pi_{codpue}(CLIENTES)$$

$$TEMP_2 \leftarrow TEMP_1 \times PUEBLOS$$

$$TEMP_3 \leftarrow \sigma_{TEMP_1.codpue=PUEBLOS.codpue}(TEMP_2)$$

$$RESULTADO \leftarrow \pi_{nombre}(TEMP_3)$$

*TEMP*₁

codpue
53596
07766
12309

PUEBLOS

codpue	nombre	codpro
07766	Burriana	12
12309	Castellón	12
17859	Enramona	12
46332	Soneja	12
53596	Vila-real	12

*TEMP*₂

<i>TEMP</i> ₁ .codpue	PUEBLOS.codpue	nombre	codpro
53596	07766	Burriana	12
53596	12309	Castellón	12
...
53596	53596	Vila-real	12
...

*TEMP*₃

<i>TEMP</i> ₁ .codpue	PUEBLOS.codpue	nombre	codpro
53596	53596	Vila-real	12
07766	07766	Burriana	12
12309	12309	Castellón	12

RESULTADO

nombre
Vila-real
Burriana
Castellón

Operaciones básicas

A las operaciones SELECCIÓN, PROYECCIÓN, PRODUCTO CARTESIANO, UNIÓN Y DIFERENCIA se les llama **operaciones básicas** (les podemos añadir RENOMBRADO e INTERSECCION).

A partir de aquí, veremos **operaciones derivadas**, que pueden ser por tanto obtenidas a partir de las anteriores, pero que conviene conocer para expresar de forma más ágil determinadas acciones anidadas.

Índice

- 1 Introducción
- 2 Operaciones básicas
- 3 Operaciones de la Teoría de Conjuntos
- 4 Operaciones derivadas**
- 5 Otros
- 6 Bibliografía

Operación CONCATENACIÓN (JOIN)

La CONCATENACIÓN de dos relaciones $R(A_1, A_2, \dots, A_n)$ y $S(B_1, B_2, \dots, B_m)$, denotada por $R \bowtie_{\text{CondicionDeConexion}} S$, es una relación de $n + m$ atributos $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$, que contiene **una tupla por cada combinación** de tuplas (una de R , otra de S) **que satisfaga una condición de conexión.**

La principal diferencia entre el PRODUCTO CARTESIANO y la CONCATENACIÓN es que en esta última sólo aparecen en el resultado las combinaciones de tuplas que satisfacen la **condición de conexión**, mientras que en el PRODUCTO CARTESIANO se incluyen todas.

Operación CONCATENACIÓN (JOIN)

- La **condición de conexión** está especificada **sobre los atributos de las dos relaciones** R y S y es evaluada para cada combinación de tuplas, incluyéndose en la relación Q resultante en forma de una única tupla combinada sólo aquellas cuya condición de conexión se evalúe como VERDADERO.
- Dicha conexión puede incluir cualesquiera operadores de comparación entre atributos de ambas relaciones (*i.e.* $\{=, <, \leq, >, \geq, \neq\}$), combinados en expresiones lógicas que agrupen varias condiciones mediante AND y OR y empleando paréntesis, como vimos con los operadores aplicados e la operación de selección.

EQUIJOIN y CONCATENACIÓN NATURAL

- El uso más habitual de CONCATENACIÓN supone el uso de condiciones de conexión sólo con comparaciones de igualdad, en cuyo caso recibe el nombre de EQUIJOIN.
- Generalmente en el resultado de una EQUIJOIN tenemos uno o más pares de atributos con valores idénticos en cada tupla.
- Ya que uno de estos valores idénticos es innecesario, y que es habitual que la condición de comparación por igualdad se refiera a los campos de ambas relaciones cuyo nombre coincida, se creó una nueva operación llamada CONCATENACIÓN NATURAL (identificada generalmente por $*$, pero como $R \bowtie S$ en ReLaX sin condición) para **evitar la necesidad de hacer explícita la condición**, y además **deshacerse en el resultado de los campos duplicados** (tipo $R.x$ y $S.x$), dejando simplemente un campo x .

Ejemplo de CONCATENACIÓN

CLIENTES

codcli	cliente	dirección	codpostal	codpue
333	Sos, Jesús	Mosen Compte, 14	12964	53596
336	Miguel, Ramón	Bernardo Mundina, 132-5	12652	07766
342	Pinel, Vicente	Francisco Sempere, 37-10	12112	07766
345	López, Mauro	Avenida del Puerto, 20-1	12010	12309
348	Palau, Jorge	Raval de Sant Josep, 97-2	12003	12309
354	Murría, José	Ciudadela, 90-18	12003	12309
357	Huguet, Juan	Calle Mestre Rodrigo, 7	12100	12309

PUEBLOS

codpue	nombre	codpro
07766	Burriana	12
12309	Castellón	12
17859	Enramona	12
46332	Soneja	12
53596	Vila-real	12

Obtener los nombres de las poblaciones en la que hay clientes ¹

¹Nótese que este ejemplo está modificado respecto al archivo del tema para Relax que tenemos en la web, en el que el nombre del cliente se almacena en el campo *nombre* en lugar de *cliente*, lo que hace que tengamos que hacer algún paso adicional en el caso del fichero de clase.

Ejemplo de Concatenación

Operación:

$$TEMP \leftarrow CLIENTES * PUEBLOS$$

$$RESULTADO \leftarrow \pi_{nombre}(TEMP)$$

Abreviando:

$$\pi_{nombre}(CLIENTES * PUEBLOS)$$

TEMP

codcli	cliente	dirección	codpostal	codpue	nombre	codpro
333	Sos, Jesús	Mosen Compte, 14	12964	53596	Vila-real	12
336	Miguel, Ramón	Bernardo Mundina, 132-5	12652	07766	Burriana	12
342	Pinel, Vicente	Francisco Sempere, 37-10	12112	07766	Burriana	12
345	López, Mauro	Avenida del Puerto, 20-1	12010	12309	Castellón	12
348	Palau, Jorge	Raval de Sant Josep, 97-2	12003	12309	Castellón	12
354	Murría, José	Ciudadela, 90-18	12003	12309	Castellón	12
357	Huguet, Juan	Calle Mestre Rodrigo, 7	12100	12309	Castellón	12

RESULTADO

nombre
Vila-real
Burriana
Castellón

Otros ejemplos de CONCATENACIÓN

- Si la condición de conexión es la igualdad de los atributos con el mismo nombre en ambas relaciones, tenemos la **CONCATENACIÓN NATURAL**, $R * S$ ($R \bowtie S$ en Re1aX).
- Si en su lugar queremos conectarlos por algún/os campo/s con el mismo nombre, pero no todos, se puede emplear la notación: $R \bowtie_{\text{NombresDeCampos}} S$.
- Si no tienen el mismo nombre, basta especificar qué campos deben ser iguales: $R \bowtie_{R.Campo1=S.Campo2} S$.
Por ejemplo: $R \bowtie_{\text{LugarDeNacimiento=LugarDeResidencia}} S$, o bien $R \bowtie_{R.LugarDeNacimiento=S.LugarDeResidencia} S$
- Aparte de equijoins, valdría cualquier condición booleana.
Por ejemplo: $R \bowtie_{\text{Sueldo2003}<\text{Sueldo2004}} S$

²Notación no aceptada en Re1aX; o hacemos join natural o ponemos condición, nunca secuencia de campos

Operación DIVISIÓN

$R \div S$, o bien R DIVIDE BY S

Sean R y S relaciones, y X , Y y Z conjuntos de atributos.

- La operación DIVISIÓN $R \div S$ se aplica a dos relaciones $R(Z)$ (grado n) y $S(X)$ (grado m), con $X \subsetneq Z$ (así, $m < n$).
- Si $Y = Z - X$ (Y serían los atributos de R no en S , grado $p = n - m$), el resultado de la DIVISIÓN es una relación $T(Y)$ que **incluye una tupla** $t = (y_1, \dots, y_p)$ si para toda tupla $(x_1, \dots, x_m) \in S$, existe la tupla $(x_1, \dots, x_m, y_1, \dots, y_p) \in R$.
- En definitiva, para que t aparezca en T , los valores de t deben aparecer en R en combinación con **cada** tupla de S .

R(Z)			S(X)	=	T(Y)
A	B		B		A
a1	b1	÷	b1		a1
a1	b2		b2		
a2	b1				
a3	b2				

Operación DIVISIÓN

- La DIVISIÓN $R(Z) \div S(X)$ puede expresarse como una secuencia de operaciones π , \times y $-$ del siguiente modo:

$$T_1 \leftarrow \pi_Y(R) \quad \text{donde } Y = Z - X$$

$$T_2 \leftarrow \pi_Y((T_1 \times S) - R)$$

$$\text{RESULTADO} \leftarrow T_1 - T_2$$

Ejemplo de División

VENDEDORES

codven	nombre	dirección	codpostal	codpue	codjefe
5	Guillén, Natalia	Sant Josep, 110	12597	53596	105
105	Poy, Paloma	Tarazona, 103-1	12257	46332	NULL
155	Rubert, Diego	Sol, 154	12425	17859	5
455	Agost, Jorge	Pasaje, 21-19	12914	53596	5

FACTURAS

codfac	fecha	codcli	codven	iva	dto
6643	16/07/2010	333	105	18	10
6645	16/07/2010	336	105	0	20
6654	31/07/2010	333	155	8	0
6659	08/08/2010	333	5	0	0
6680	10/09/2010	333	455	8	0
6723	06/11/2010	342	5	18	0
6742	17/12/2010	312	105	8	20

Obtener la lista de los códigos de los clientes que han realizado compras a todos los vendedores

Ejemplo de división

$$TEMP_1 \leftarrow \pi_{codcli, codven}(FACTURAS)$$
$$TEMP_2 \leftarrow \pi_{codven}(VENDEDORES)$$
$$RESULTADO \leftarrow TEMP_1 \div TEMP_2$$
 $TEMP_1$

codcli	codven
333	105
336	105
333	155
333	5
333	455
342	5
312	105

 $TEMP_2$

codven
5
105
155
455

 $TEMP_1 \div TEMP_2$

codcli
333

Índice

- 1 Introducción
- 2 Operaciones básicas
- 3 Operaciones de la Teoría de Conjuntos
- 4 Operaciones derivadas
- 5 Otros**
- 6 Bibliografía

Otros operadores

- Concatenación externa por la izquierda (**LEFT OUTER JOIN**).
- Concatenación externa por la derecha (**RIGHT OUTER JOIN**).
- Concatenación externa completa (**FULL OUTER JOIN**).
- Unión externa **OUTER UNION**
- Álgebra Relacional Extendida:
 - Agrupación o resumen **SUMMARIZE / GROUP BY**
 - Proyección generalizada/extendida
 - Cierre recursivo
 - Unicidad
- ...

Agrupación

$\Gamma_{\langle \text{CamposAgrupacion} \rangle} \langle \text{Agregaciones} \rangle (R)$ o bien

SUMMARIZE R GROUP BY <Campos> ADD <Agregaciones>

- Permite realizar operaciones de agregación de datos, proporcionando una serie de campos por los que agrupar y un conjunto de funciones de agregación sobre los atributos de la relación.
- Las *agregaciones* pueden contener tanto campos de R como aplicación de funciones de agregación tipo *count*, *sum*, *min*, *max*, *avg*, etc.

Ejemplo de agrupación

Ejercicio: Contar el número de ventas por cada cliente y el promedio de IVA aplicado, de la relación FACTURAS.

$\Gamma_{\text{codcli}}^{\text{count}(\text{codfac}), \text{avg}(\text{iva})}(\text{CLIENTE})$ o bien

SUMMARIZE CLIENTE GROUP BY codcli ADD count(codfac), avg(iva)

CLIENTE			Orden:			Agregación:		
codcli	codfac	iva	codcli	codfac	iva	codcli	count(codfac)	avg(iva)
333	6643	18	312	6742	8	312	1	8
336	6645	0	333	6643	18	333	4	8.5
333	6654	8	333	6654	8	336	1	0
333	6659	0	333	6659	0	342	1	18
333	6680	8	333	6680	8			
342	6723	18	336	6645	0			
312	6742	8	342	6723	18			

Nota: en realidad la agregación no ordena, pero lo hacemos así a mano por facilitar la traza.

Índice

- 1 Introducción
- 2 Operaciones básicas
- 3 Operaciones de la Teoría de Conjuntos
- 4 Operaciones derivadas
- 5 Otros
- 6 Bibliografía**

Bibliografía



Ramez Elmasri, Shamkant B. Navathe

Fundamentos de Sistemas de Bases de Datos.

Pearson Educación S.A. - Addison Wesley Quinta Ed. (2007)

ISBN: 978-84-7829-085-7



Mercedes Marqués

Bases de Datos.

Universitat Jaume I (2011)

ISBN: 978-84-693-0146-3



Dolores Cuadra, Elena Castro, Ana M^a Iglesias, Paloma Martínez, Fco.

Javier Calle, César de Pablo, Harith Al-Jumaily, Lourdes Moreno, Sonia

García Manzano, José Luis Martínez, Jesica Rivero, Isabel Segura

Desarrollo De Bases De Datos: Casos Prácticos Desde El Análisis A La Implementación.

RA-MA Editorial - 2^a Edición (2013)

ISBN: 978-84-996-4124-9