UNIVERSIDAD NACIONAL DE JUJUY

FACULTAD DE INGENIERIA

BASE DE DATOS I

ÁLGEBRA RELACIONAL

Profesor Adjunto: **Mg. Ing. Héctor Pedro Liberatori**

AÑO 2023

La parte de manipulación del Modelo Relacional ha evolucionado considerablemente desde la publicación de los documentos originales de Codd sobre el tema. Sin embargo el álgebra relacional es el componente principal de esa parte de manipulación.

4.1. OPERADORES RELACIONALES

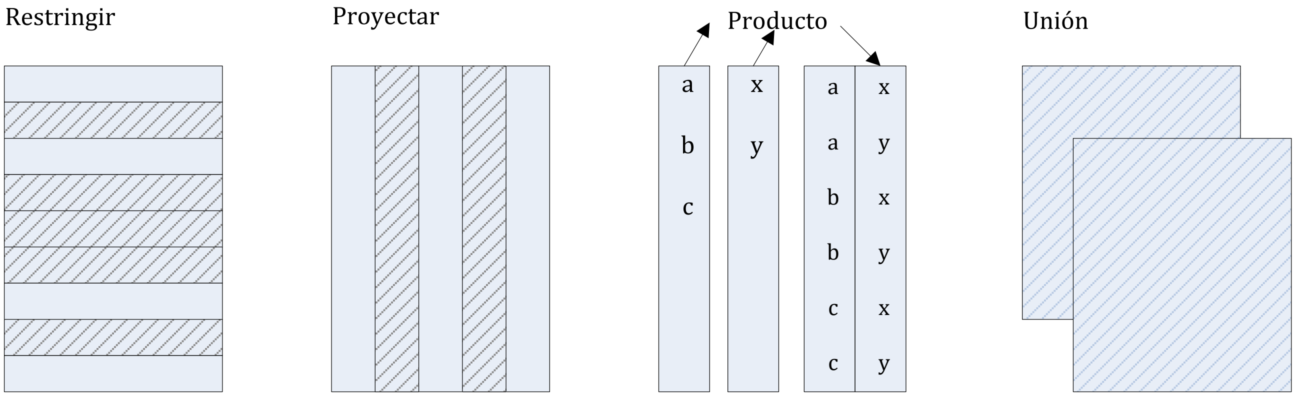
El componente principal de la parte de manipulación es lo que se denomina *algebra relacional*, que básicamente es sólo el conjunto de operadores que toman relaciones como sus operandos y regresan una relación como su resultado. Los 3 operadores del algebra relacional estudiados en el capítulo tres son:

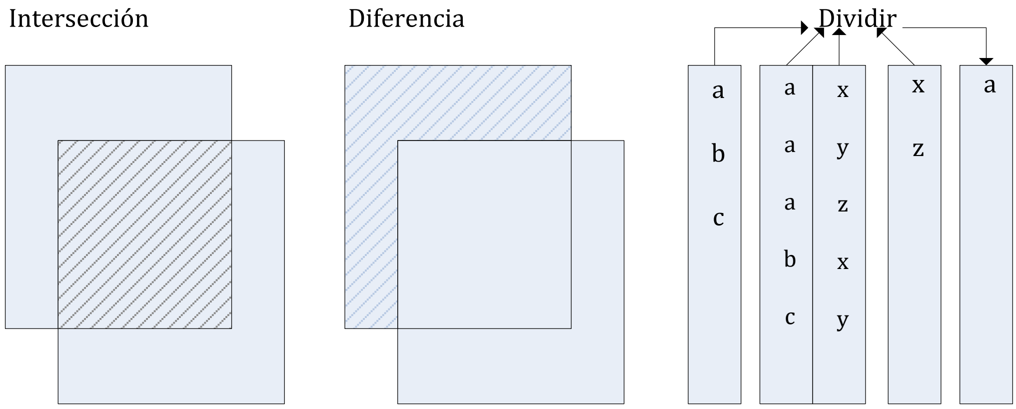
- **Restringir** (seleccionar): extrae las filas especificadas de una tabla.
- **Proyectar:** extrae las columnas especificadas de una tabla.
- **Juntar:** reúne dos tablas con base en valores comunes de una columna común.

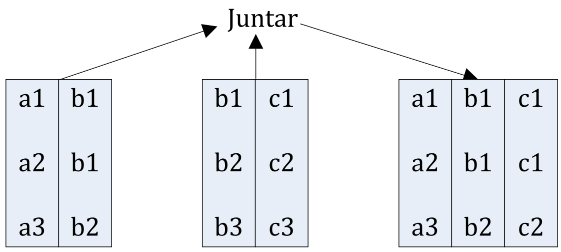
E. F. Codd definió lo que generalmente se conoce como “algebra original”, es decir, un conjunto de 8 operadores divididos en dos grupos.

1. El conjunto tradicional de operadores: unión, intersección, diferencia y producto cartesiano.
2. Los operadores relacionales especiales: restringir, proyectar, juntar y dividir.

- **Restringir:** regresa una relación que contiene todas las tuplas de una relación especificada, que satisfacen una condición especificada.
- **Proyectar:** regresa una relación que contiene todas las tuplas o subtuplas que quedan en una relación especificada después de quitar los atributos especificados.
- **Producto:** regresa una relación que contiene todas las tuplas posibles que son una combinación de dos tuplas, una de cada una de dos relaciones especificadas.
- **Unión:** regresa una relación que contiene todas las tuplas que aparecen en una o en las dos relaciones especificadas.
- **Intersección:** regresa una relación que contiene todas las tuplas que aparecen en las dos relaciones especificadas (en ambas, no en una u otra).
- **Diferencia:** regresa una relación que contiene todas las tuplas que aparecen en la primera pero no en la segunda de las dos relaciones especificadas.
- **Juntar:** regresa una relación que contiene todas las tuplas posibles que son una combinación de dos tuplas de cada una de dos relaciones especificadas, tales que la dos tuplas que contribuyen a cualquier combinación dada tengan un valor común para los atributos comunes de las dos relaciones (el valor común aparecerá sólo una vez en la tupla resultante).
- **Dividir:** toma dos relaciones unarias y una relación binaria y regresa una relación que contiene todas las tuplas de una relación unaria que aparecen en la relación binaria y que a la vez coinciden con todas las tuplas de la otra relación unaria.







4.2. REVISIÓN DE LA PROPIEDAD DE CIERRE

Al hecho de que la salida de cualquier operación relacional dada sea otra relación, se conoce como la propiedad relacional de cierre.

Cierre significa que se pueden escribir expresiones relacionales anidadas; es decir, expresiones relacionales en la que los propios *operandos* están representados a su vez por expresiones relacionales de una complejidad arbitraria.

El encabezado de una varrel base es bien entendido y conocido por el sistema, ya que está especificado como parte de la definición de la varrel base relacionada. Pero como interpreta el sistema el encabezado de las relaciones derivadas? , por ejemplo:

V Join P (representa la junta de proveedores y partes con base en ciudades coincidentes, en donde CIUDAD es el único atributo común a las 2 relaciones)

Es obvio como quedará el cuerpo de la varrel resultado, pero el encabezado del resultado debe ser un *tipo de relación* bien definido, en particular con nombre de atributo apropiado. De esta forma se podrá hacer referencia a esos nombres de atributo en operaciones posteriores. Por ejemplo no se podría escribir la siguiente expresión:

(V Join P) WHERE CIUDAD = “Atenas”

Si el sistema no conociera que el resultado de evaluar la expresión (V Join) tiene un atributo común llamado ciudad.

Por lo tanto se necesita de un conjunto de *reglas de inferencia de tipo de relación* integradas dentro del algebra. De esta forma se podrá inferir el tipo de relación de salida en cada operación y en particular un conjunto bien definido de nombres de atributo.

Para lograr este objetivo se utiliza el operador RENAME, este operador toma una relación dada y regresa otra que es idéntica a la primera con excepción de que por lo menos uno de los atributos tiene un nombre diferente. Por ejemplo:

V RENAME ciudad AS CIUDADV, provoca el siguiente cambio en el encabezado de la relación de proveedores:

V#	PROVEEDOR	STATUS	CIUDADV

P RENAME PARTE AS NOMP, PESO AS PS

produce los siguientes cambios en el encabezado de la relación de partes

P#	NOMP	COLOR	PS	CIUDAD

De manera explícita la disponibilidad del operador RENAME significa (a diferencia de SQL) que el algebra relacional no necesita de nombres de atributo cualificados como por ejemplo V.V#.

4.3. SINTAXIS (utilizar la Base de Datos de Proveedores y Partes en página 11)

Para representar las expresiones del Algebra Relacional se utilizará la sintaxis de Tutorial D.

<expresión relacional>
::= RELATION {<lista de expresiones de tupla>}
<nombre de varrel>
<operación relacional>
(expresión relacional)

A

V#	PROVEEDOR	STATUS	CIUDAD
V1	Smith	20	Londres
V4	Clark	20	Londres

B

V#	PROVEEDOR	STATUS	CIUDAD
V1	Smith	20	Londres
V2	Jones	10	París

Unión (A unión B)			
V#	**PROVEEDOR**	**STATUS**	**CIUDAD**
V1	Smith	20	Londres
V4	Clark	20	Londres
V2	Jones	10	París

Intersección (A intersect B)			
V#	**PROVEEDOR**	**STATUS**	**CIUDAD**
V1	Smith	20	Londres

Diferencia (A minus B)			
V#	**PROVEEDOR**	**STATUS**	**CIUDAD**
V4	Clark	20	Londres

Diferencia (B minus A)			
V#	**PROVEEDOR**	**STATUS**	**CIUDAD**
V2	Jones	10	París

4.3.1. Unión

En matemáticas la unión de 2 conjuntos es el conjunto de elementos que pertenecen ya sea a uno o a ambos conjuntos originales. Como una relación contiene un conjunto de tuplas, el resultado será un conjunto consistente en todas las tuplas que aparecen en cualquiera o en ambas relaciones originales.

Por ejemplo la unión del conjunto de tuplas de proveedores que aparecen actualmente en la varrel V y el conjunto de tuplas de partes que aparecen actualmente en la varrel P, es en realidad un conjunto. Sin embargo este resultado no es una relación, debido a que las relaciones no pueden contener una mezcla de diferentes clases de tuplas, deben ser tuplas homogéneas.

Por lo tanto, la unión en el algebra relacional no es la unión matemática común; es una clase especial de unión en la que se requiere que las dos relaciones de entrada sean del mismo tipo. (A unión B).

Si las dos relaciones que intervienen en la unión son del mismo tipo, su resultado será también una relación del mismo tipo, en otras palabras se conservará la propiedad la propiedad de cierre.

4.3.2. Intersección

Al igual que la unión y básicamente por la misma razón, el operador relacional de intersección requiere que sus operandos sean del mismo tipo.

Dadas 2 relaciones A y B del mismo tipo, la intersección de esas dos relaciones (A intersect B) es una relación del mismo tipo, con un cuerpo que consiste en todas las tuplas t, tal que t aparece tanto en A como en B.

4.3.3. Diferencia

Al igual que la unión y la intersección, el operador relacional de diferencia requiere también que sus operandos sean del mismo tipo.

Dadas 2 relaciones A y B del mismo tipo, la diferencia entre esas dos relaciones (A minus B) es una relación del mismo tipo, con un cuerpo que consiste en todas las tuplas t, tal que t aparece tanto en A y no en B.

4.3.4. Producto

En matemáticas, el producto cartesiano de 2 conjuntos es el conjunto de todos los pares ordenados tales que en cada par, el primer elemento viene del primer conjunto y el segundo elemento viene del segundo conjunto. Por lo tanto, el producto cartesiano de dos relaciones será un conjunto ordenado de pares de tuplas.

Para conservar la propiedad de cierre, o sea que el resultado contenga las tuplas como tales; en la versión relacional del producto cartesiano cada par ordenado de tuplas es sustituido por una sola tupla que es la unión de las tuplas en cuestión (usando “unión” en el sentido normal de la teoría de conjuntos, NO en el sentido relacional). Esto es, dadas las tuplas

$$\{A_1 : a_1, A_2 : a_2, \ldots, A_n : a_n\}\ y\ \{B_1 : b_1, B_2 : b_2, \ldots, B_n : b_n\}$$

la unión de ambas es la tupla:

$$\{A_1 : a_1, A_2 : a_2, \ldots, A_n : a_n, B_1 : b_1, B_2 : b_2, \ldots, B_n : b_n\}$$

Otro problema que se presenta es que la relación resultante tenga un encabezado bien formado, es decir, que sea de un tipo apropiado de relación.

Surgirá un problema si los 2 encabezados tienen un nombre de atributo común, debido a que el resultado tendría 2 atributos con el mismo nombre y por lo tanto no estaría “bien formado”. Entonces para generar el producto cartesiano de 2 relaciones que tienen nombres de atributo comunes, primero se debe utilizar el operador RENAME.

Por lo expuesto, el producto cartesiano (relacional) de 2 relaciones A y B (A times B) (donde A y B no tienen atributos comunes) es una relación con un encabezado que es la unión (en la teoría de conjuntos) de los encabezados de A y B y con un cuerpo que consiste en el conjunto de todas las tuplas t, tal que t es la unión (en la teoría de conjuntos) de una tupla que aparece en A y una tupla que aparece en B. La cardinalidad del resultado es el producto de las cardinalidades de A y B. el grado del resultado es la suma del grado de A y B.

A

V#
V1
V2
V3

B

P#
P1
P2
P3

Producto Cartesiano (A TIMES A)

V#	P#
V1	P1
V2	P2
V3	P3

V1	P1
V2	P2
V3	P3

V1	P1
V2	P2
V3	P3

4.3.5. Restringir

Si la relación A tiene los atributos X e Y, y sea Θ un operador (por lo general “=”, “<”, “>”, etc.) tal que la condición XΘY esté bien definida y dé como resultado valores (verdadero o falso) para valores *particulares* de X e Y. entonces la restricción Θ de la relación A sobre los atributos X e Y, es una relación con el mismo encabezado que A y con un cuerpo que consiste en todas las tuplas t de A tal que la condición XΘY dé como resultado verdadero para esa tupla t.

P WHERE PESO < PESO(14)

P#	Parte	Color	Peso	Ciudad
P1	Tuerca	Rojo	12	Londres
P5	Lera	Azul	12	París

El operador restricción produce un subconjunto horizontal de una relación. Es decir un subconjunto de tuplas de la relación dada para el cual se satisface una condición de restricción especifica.

4.3.6. Proyectar

Dada la relación A y los atributos $X, Y, \ldots, Z$; entonces la proyección de la relación A sobre $X, Y, \ldots, Z$ es: $A\{X, Y, \ldots, Z\}$ que es una relación con:

- Un encabezado derivado del encabezado de A al quitar todos los atributos no mencionados en el conjunto $\{X, Y, \ldots, Z\}$
- Un cuerpo consistente en todas las tuplas $\{X : x, Y : y, \ldots, Z : z\}$, tal que una tupla aparece en A con el valor x para X, el valor y para Y y el valor z para Z.

Por lo tanto, el operador de proyección produce un subconjunto "vertical" de una relación dada. Es decir un subconjunto obtenido al quitar todos los atributos no mencionados en la lista de atributos especificada y después eliminar las tuplas duplicadas.

Si la lista de nombres de atributo menciona todos los atributos de A, la proyección es una proyección identidad. Es válida una proyección de la forma A{} (lista de nombres de atributos vacía). Representa una proyección de carácter nulo.

V{Ciudad}

CIUDAD
Londres
París
Atenas

(V WHERE CIUDAD = "París") {V#}

V#
V2
V3

4.3.7. Juntar

Dada las relaciones A y B con los encabezados

$$\{X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n\}$$

y

$$\{Y_1, Y_2, \ldots, Y_n, Z_1, Z_2, \ldots, Z_p\}$$

respectivamente. Es decir, con los atributos $Y_1, Y_2, \ldots, Y_n$ comunes a las 2 relaciones.

Entonces la *junta natural* de A y B (A JOIN B) es una relación con el encabezado $\{X, Y, \ldots, Z\}$ y un cuerpo que consiste en el conjunto de todas las tuplas $\{X : x, Y : y, \ldots, Z : z\}$.

V JOIN P
Junta natural
sobre el atributo
común CIUDAD

V#	Proveedor	Status	Ciudad	P#	Parte	Color	Peso
V1	Smith	20	Londres	P1	Tuerca	Rojo	12
V1	Smith	20	Londres	P4	Tornillo	Rojo	14
V1	Smith	20	Londres	P6	Engrane	Rojo	19
V2	Jones	10	París	P2	Perno	Verde	17
V2	Jones	10	París	P5	Leva	Azul	12
V3	Blake	30	París	P2	Perno	Verde	17
V3	Blake	30	París	P5	Leva	Azul	12
V4	Clark	20	Londres	P1	Tuerca	Rojo	12
V4	Clark	20	Londres	P4	Tornillo	Rojo	14
V4	Clark	20	Londres	P6	Engrane	Rojo	19

Las JUNTAS no siempre son entre una clave externa y una clave primaria coincidente (aunque son las más comunes).

La junta Θ es una operación que está diseñada para juntar 2 relaciones con base en algún operado de comparación distinto al de igualdad. Sean las relaciones A y B tales que satisfagan los requerimientos del producto cartesiano (que no tengan atributos comunes). X es un atributo de A e Y es un atributo de B. Entonces la junta Θ de la relación A sobre el atributo X con la relación B sobre el atributo Y, se define como el resultado de evaluar la expresión:

(A TIMES B) where X Θ Y

en otras palabras, es una relación con el mismo encabezado que el producto cartesiano de A y B, con un cuerpo que consiste en el conjunto de todas las tuplas t tal que t aparece en ese producto cartesiano y la condición "X Θ Y" resulta verdadera para esa tupla t.

Ejemplo: calcular la junta Θ "mayor que" (>) de la relación V sobre CIUDAD con la relación P sobre CIUDAD.

Una expresión relacional apropiada sería la siguiente:

((V RENAME CIUDAD AS CIUDADV) TIMES
(P RENAME CIUDAD AS CIUDAD P))
WHERE CIUDADV > CIUDADP

Con cambiar el nombre de un solo atributo hubiera sido suficiente.

V#	Proveedor	Status	Ciudad	P#	Parte	Color	Peso	Ciudad P
V2	Jones	10	París	P1	Tuerca	Rojo	12	Londres
V2	Jones	10	París	P4	Tornillo	Rojo	14	Londres
V2	Jones	10	París	P6	Engrane	Rojo	19	Londres
V3	Blake	30	París	P1	Tuerca	Rojo	12	Londres
V3	Blake	30	París	P4	Tornillo	Rojo	14	Londres
V3	Blake	30	París	P6	Engrane	Rojo	19	Londres

4.3.8. Dividir

Dadas las relaciones A y B con los encabezados $\{X_1, X_2, \ldots, X_n\}$ y $\{Y_1, Y_2, \ldots, Y_n\}$ respectivamente. Y la relación C con el encabezado $\{X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_n\}$ (unión de los encabezados de A y B). Entonces la división de A entre B por C (A->dividendo, B->divisor y C es el "mediador")

Es; A DIVIDE B y B PER C

El resultado consiste en aquellos valores X de A cuyos correspondientes valores Y en C incluyen a todos los valores Y de B.

En el último ejemplo, el divisor es una relación que contiene los números de todas las partes.

Entonces como resultado de la división se obtienen los proveedores que suministran todas las partes.

A (Dividendo)

V#
V1
V2
V3
V4
V5

C (Mediados)

V#	P#
V1	P1
V1	P2
V1	P3
V1	P4
V1	P5
V1	P6
...	...
...	...
V2	P1
V2	P2
V3	P2
V4	P2
V4	P4
V4	P5

B (divisor) B<- (divisor) -> B

P#
P1

P#
P2
P4

P#
P1
P2
P3
P4
P5
P6

RESULTADOS

V#
V1
V2

V#
V2
V4

V#
V1

4.3.9. Asociatividad y Conmutatividad

Es fácil verificar que la operación UNION es asociativa. Si A y B son expresiones relacionales:

(A UNION B) UNION C
y
A UNION (B UNION C)

son equivalentes; por lo tanto se podría escribir sin los paréntesis.

Las operaciones INTERSECT, TIMES y JOIN, también son asociativas.

La propiedad conmutativa también se verifica en la unión.

A UNION B
B UNION A

En la teoría de conjuntos, la operación de producto cartesiano no es asociativa ni conmutativa. En cambio en la versión relacional si presenta ambas propiedades.

Si A y B no tienen nombres de atributos comunes entonces: A JOIN B es equivalente a : A TIMES B; es decir, en este caso la *junta natural* degenera en un producto cartesiano.

4.4. EJEMPLOS

Los ejemplos que se muestran tratan el uso de expresiones del álgebra relacional en la formulación de consultas.

- Obtener los nombres de los proveedores que suministran la parte P2.

  ((VP JOIN V) WHERE P# = P# (“P2”)) {PROVEEDOR}

- Obtener los nombres de los proveedores que suministran por lo menos una parte roja

  (((P WHERE COLOR = COLOR (“ROJO”)
  JOIN VP) {V#}) JOIN V {PROVEEDOR}

- Obtener los nombres de los proveedores que suministran todas las partes

  ((V {V#} DIVIDE BY P {P#} PER VP {V#, P#})
  JOIN V) {PROVEEDOR}

- Obtener los números de los proveedores que suministran al menos todas las partes que suministra el proveedor V2.

  V {V#} DIVIDE BY (VP WHERE V# = V# (“V2”)) {P#}
  PER VP {V#, P#}

- Obtener todos los pares de números de proveedor, tal que los pares siempre sean de la misma ciudad.

  (((V RENAME V# AS VA) {VA, CIUDAD} JOIN
  (V RENAME V# AS VB) {VB, CIUDAD})
  WHERE VA < VB) {VA, VB}

  la finalidad d la condición VA < VB es doble:

  - Elimina los pares de números de proveedor de la forma (X,X)
  - Garantiza que no aparezcan ambos pares: (X,Y) y (Y,X).

- Obtener los nombres de los proveedores que no suministran la parte P2

  ((V {V#} MINUS (VP WHERE P# = P# (“P2”)) {V#})
  JOIN V) {PROVEEDOR}

  Nota: no siempre es fácil ver de inmediato como formular una consulta dada como una sola expresión anidada. Otra forma de realizar el ejercicio anterior sería:

  WITH (V RENAME V# AS VA) {VA, CIUDAD} AS T1,
  (V RENAME V# AS VB) {VB, CIUDAD} AS T2,
  T1 JOIN VA < VB AS T4:
  T4 {VA, VB}

Las expresiones que anteceden a los 2 puntos no requieren de una evaluación inmediata por parte del sistema, todo lo que éste tiene que hacer es recordarlas junto con los nombres introducidos por las cláusulas **AS** correspondientes.

La expresión que sigue a los dos puntos denota el resultado final de la consulta. Al llegar a este punto, el sistema no puede demorar más la evaluación y deberá calcular el valor buscado.

4.5. PARA QUE SIRVE EL ÁLGEBRA?

Hasta ahora en el presente capítulo se han utilizado las operaciones del álgebra relacional para la recuperación de datos.

La intención fundamental del álgebra es permitir la escritura de expresiones relacionales para atender diversos fines. La siguiente lista muestra algunas de las aplicaciones posibles:

- Definir un alcance para la *recuperación*, es decir, definir los datos a obtener en las operaciones de recuperación.
- Definir un alcance para la *actualización*, es decir, definir los datos a insertar, modificar o eliminar en alguna operación de actualización.
- Definir *restricciones de integridad*, es decir, definir ciertas restricciones que la base de datos debe satisfacer.
- Definir *varrels derivadas*, es decir, definir los datos a incluir en una vista.
- Definir *requerimientos de estabilidad*, es decir, definir los datos que serán el alcance de cierta operación de control de concurrencia.
- Definir *restricciones de seguridad*, es decir, definir los datos sobre los cuales se va a conceder autorización de alguna clase.
- El álgebra también sirve como una base conveniente para la optimización; por ejemplo la expresión:

  ((VP JOIN V) WHERE P# = P# (“P2”)) {PROVEEDOR}

  que produce una consulta de los operadores que suministran la parte P2, también puede ser escrita de la siguiente forma:

  ((VP WHERE P# = P# (“P2”)) JOIN V) {PROVEEDOR}

  esta expresión es más eficiente que la anterior. Si un usuario ingresara la primera expresión, el OPTIMIZADOR se encarga de convertirla a su forma más conveniente antes de ejecutarla.
- Debido a su naturaleza fundamental, también se usa el álgebra como una clase de norma contra la cual se puede medir el poder expresivo de un lenguaje de programación. Se dice que un lenguaje está *relacionalmente completo* si es por lo menos tan poderoso como el álgebra, es decir si sus expresiones permiten la definición de cada relación que pueda ser definida por medio de expresiones del álgebra.

4.6. AGRUPAMIENTO Y DESAGRUPAMIENTO

El hecho de que se puedan tener relaciones con atributos cuyos valores son a su vez relaciones conduce a la necesidad de ciertos operadores relacionales adicionales denominados agrupar y desagrupar.

VP GROUP (P#, CANT) AS PC

La expresión puede ser leída de la siguiente manera: agrupa VP por V# (V# es el único atributo de VP que no se menciona en la cláusula GROUP).

VPC		
V#	**PC**	
V1	**P#**	**Cant**
	P1	300
	P2	200
	P3	400
	P4	200
	P5	100
	P6	100
V2	**P#**	**Cant**
	P1	300
	P2	200
V3	**P#**	**Cant**
	P2	200
V4	**P#**	**Cant.**
	P2	200
	P4	300
	P5	400

Agrupar

La relación **VPC** presenta los atributos **V#** y **PC**. Este último tiene valor de relación y tiene a su vez los atributos **P#** y **CANT**. El cuerpo de la relación **VPC**, contiene exactamente una tupla para cada valor distinto de **V#** en **VP**.

Desagrupar

La expresión **VPC** UNGROUP **PC** conduce nuevamente a la relación **VP**.

El encabezado consiste en los atributos **P#** y **CANT** (derivados del atributo **PC**) junto con los demás atributos de **VPC** (**V#**).

El cuerpo contiene exactamente una tupla para cada combinación de una tupla de **VPC** con una tupla del valor de **PC** dentro de esa tupla.

4.7. COMPARACIONES RELACIONALES

El álgebra relacional, tal como se definió originalmente, no incluía ninguna forma directa de comparar 2 relaciones.

Una consecuencia de esta omisión fue que ciertas consultas eran extremadamente difíciles de expresar. Para corregir este inconveniente se define una nueva clase de condición:

<expresión relacional> θ <expresión relacional>

las relaciones denotadas por las dos <expresiones relacionales> deben ser del mismo tipo. El operador de comparación puede ser cualquiera de los siguientes:

= (igual a)

≠ (desigual a)

≤ (subconjunto de)

< (subconjunto propio de)

≥ (superconjunto de)

> (superconjunto propio de)

Ejemplos

- Verificar si la proyección de *proveedores* sobre ciudad es la misma que la proyección de *partes* sobre *CIUDAD*.

  V {CIUDAD} = P {CIUDAD}

- Existe algún proveedor que no suministre parte alguna?

  V{V#} > VP{V#}

- Obtener una relación cuyas tuplas identifiquen a los proveedores que suministran todas las partes.

  V WHERE ((VP RENAME V# AS X) WHERE X = V#) {P#}

- Obtener los números de parte que suministra un proveedor dado.

  ((VP RENAME V# AS X) WHERE X = V#) {P#}

  esta consulta también se puede formular en términos de DIVIDE BY:

V JOIN (V{V#} DIVIDE BY P {P#} PER VP {V#, P#}

Importante

Las comparaciones relacionales no son condiciones de restricción.

A WHERE C1 AND C2 = (A WHERE C1) INTERSECT (A WHERE C2)

A WHERE C1 OR C2 = (A WHERE C1) UNION (A WHERE C2)

A WHERE NOT C = A MINUS (A WHERE C)

4.8. LA BASE DE DATOS DE PROVEEDORES Y PARTES

Debido a que en unidades posteriores se tomará como ejemplo la base de datos de Proveedores y Partes, en esta sección se hará un análisis de la misma:

V (Proveedores)			
V#	**Proveedor**	**Status**	**Ciudad**
V1	Smith	20	Londres
V2	Jones	10	París
V3	Blake	30	París
V4	Clark	20	Londres
V5	Adams	30	Atenas

P (Partes)				
P#	**Parte**	**Color**	**Peso**	**Ciudad**
P1	Tuerca	Rojo	12	Londres
P2	Perno	Verde	17	París
P3	Tornillo	Azul	17	Roma
P4	Tornillo	Rojo	14	Londres
P5	Leva	Azul	12	París
P6	Engrane	Rojo	19	Londres

VP (Envíos)		
V#	**P#**	**Cant**
V1	P1	300
V1	P2	200
V1	P3	400
V1	P4	200
V1	P5	100
V1	P6	100
V2	P1	300
V2	P2	400
V3	P2	200
V4	P2	200
V4	P4	300
V4	P5	400