

Álgebra relacional

Introducción y operaciones básicas

Introducción a las bases de datos (IBD)

UNQ

Apunte complementario sobre álgebra relacional y sus operaciones básicas. Se introducen definiciones y ejemplos sobre su uso.

Índice

0. El ejemplo	3
1. Definición - ¿qué es el álgebra relacional?	4
¿Qué quiere decir "expresar", cómo se expresan consultas?	4
Ya lo sabemos con números, ahora aprendámoslo con instancias de BD.	5
2. Selección - σ	6
Definición	6
Un ejemplo: actos poco peligrosos	6
Ejemplo 2: actos largos	7
3. Proyección - π	8
Definición	8
Un ejemplo: nombre y población de ciudades	8
Otro ejemplo: actos y duraciones	9
4. Combinación de operaciones	10
Qué es combinar – composición	10
Un ejemplo: ciudades y público en junio	10
Ejemplo 2: ciudades grandes	12
5. De nuevo: ¿qué es el álgebra relacional?	12
¿Quién hace las consultas?	12
OK, quiero describir una consulta. ¿Qué hago?	13
Muy importante (Resultado de toda operación)	13
6. Join natural – combinación de tablas	14
Lo nuevo - combinar tablas	14
En pocas palabras	14
Un ejemplo: cada componente de una función con su peligrosidad	15
Ejemplo combinado: público vs población	16

0. El ejemplo

Vamos a trabajar con estos esquemas de tabla de la BD que modela parte de la información de un circo.

```

funcion <fn-dia, fn-hora, ciudad-nombre, precioEntrada, publicoEsperado>
acto <acto-nombre, gradoPeligrosidad>
formaParte <fn-dia, fn-hora, acto-nombre, nroOrden, duracion>
ciudad <ciudad-nombre, provincia, poblacion>
textoActo <acto-nombre, idioma, texto, fRedaccion, calidad>
artista <nombreArt, fNac, cachet, patenteTrailer, nombreArt-maestro>
puedeParticipar <acto-nombre, nombreArt>
trailer <patente, maxOcupantes, marca>

```

y una instancia de este esquema que incluye las siguientes tablas, o sea, instancias de tabla.

función

fn-dia	fn-hora	ciudad-nombre	precioEntrada	publicoEsperado
28/05/2008	19	Azul	30	2000
02/06/2008	18	Tandil	20	320
02/06/2008	21	Tandil	25	700
05/06/2008	20	Tornquist	28	75
09/06/2008	20	Bahía Blanca	16	1400
02/07/2008	17	Río Gallegos	35	470

formaParte

fn-dia	fn-hora	acto-nombre	nroOrden	duracion
28/05/2008	19	payasos	1	40
28/05/2008	19	mago	2	25
28/05/2008	19	domadores	3	38
02/06/2008	18	sogas	1	38
02/06/2008	18	payasos	2	15
02/06/2008	18	domadores	3	22
02/06/2008	18	cuchillos	4	31

Acto

acto-nombre	gradoPeligrosidad
mago	0
domadores	7
sogas	3
payasos	1
cuchillos	7
mono	2

Ciudad

ciudad-nombre	provincia	poblacion
Azul	Buenos Aires	70000
Tandil	Buenos Aires	120000
Tornquist	Buenos Aires	12000
Bahía Blanca	Buenos Aires	600000
Rawson	Chubut	140000
Río Gallegos	Santa Cruz	80000

1. Definición - ¿qué es el álgebra relacional?

Es una forma de expresar consultas a partir de una instancia de BD.

Consulta = operación de extracción de datos, parte de una instancia, da como resultado una tabla que por lo general no es una de las tablas de la instancia de BD, pero sí sus datos surgen de los datos que están en la BD.

Ejemplos de consultas que puedo expresar

- qué funciones va a haber en Tandil en junio.
- en qué provincias va a haber funciones que incluyan sogas en agosto.
- en qué actos puede participar PepeMax.
- qué artistas pueden participar en actos con grado de peligrosidad mayor a 5.

¿Qué quiere decir "expresar", cómo se expresan consultas?

Mediante cuentas, que pueden incluir muchas operaciones encadenadas. El mecanismo de expresar cosas mediante cuentas lo conocemos desde la primaria, haciendo cuentas no con tablas, pero sí con números. Un ejemplo: tengo dos tupper, uno con 18 fetas de queso, otro con 12 más. Tengo que armar 8 sandwiches, sé que 3 fetas me las voy a comer yo y 11 las tengo que guardar para armar empanadas mañana.

¿Cuántas fetas le tengo que poner a cada sandwich? Sí, dos.

... ¿por qué dos? ... o sea, ¿cómo llegamos al resultado 2? Ahora nos resulta fácil porque somos capaces de traducir el problema a cuenta enseguida, y obtener el resultado.

La cuenta es

$$((18+12) - (3+11)) / 8$$

y el "dos" es el **resultado final** de la cuenta.

La expresión aritmética que está acá arriba es una forma de expresar el problema que teníamos (saber cuántas fetas le tengo que poner a cada sandwich).

Los operandos son números, que son la forma de representar lo que me interesa de las fetas, que es la cantidad. En otro problema, puede que me interese otra cosa, p.ej. el volumen, el peso, la hora a la que pasa algo, muchas cosas distintas. Todas estas las podemos representar mediante números.

La ventaja de representar problemas usando números es que hay varias operaciones sobre números que aprendimos de chicos: para empezar, suma, resta, multiplicación y división.

El resultado de cada operación es un número, que a su vez es operando de la operación de más afuera. P.ej. $18 + 12$ da 30, que es el minuendo (el de adelante) de la resta. El resultado de la resta es 16, que es el dividendo de la división.

El resultado final, si planteamos bien el problema, si lo expresamos en forma correcta, es la solución que estábamos buscando.

Ya lo sabemos con números, ahora aprendámoslo con instancias de BD.

¿Qué tienen de común un número y una tabla?

Que usamos a ambos para representar cosas.

Una tabla representa información (p.ej. las funciones incluidas en la instancia de BD) mediante datos que están organizados según los principios del modelo relacional:

1. es un conjunto de filas, y
2. cada fila tiene un valor por cada atributo.

Entonces, si quiero obtener cierta información (los precios de las entradas de las funciones de junio) hago operaciones. El álgebra relacional define esas operaciones, como la aritmética define qué quiere decir +, qué quiere decir -, etc. .

Al igual que cuando usamos números,

- el resultado de todas las operaciones va a ser una tabla, que no es casi siempre ninguna de las que tengo, sino una nueva que se arma a partir de los resultados de las que tengo. Ninguno de los números en la expresión de arriba es 30, el 30 se obtiene como resultado de una cuenta parcial.
- pueden encadenarse operaciones, el resultado de una es operando de la que está afuera.
- el resultado final es la solución que estaba buscando.

En el álgebra relacional, en lugar de tener suma, resta, etc., vamos a tener otras operaciones. Ahora vamos a ver la primera.

1. 2. Selección - σ

Definición

La selección es una operación cuyos operandos son: una tabla y una condición. El resultado de una operación de selección es el conjunto de filas de la tabla que cumplen la condición, tiene las mismas columnas que la tabla.

Notación:

$\sigma_{\langle \text{criterio} \rangle} \text{tabla}$

Por ejemplo el resultado de

$\sigma_{\langle \text{precioEntrada} = 25 \rangle} \text{function}$

son las filas de `function` que representan funciones cuya entrada vale 25 pesos.

Un ejemplo: actos poco peligrosos

Queremos obtener los actos poco peligrosos, o sea, cuyo grado de peligrosidad no supere 2. Esa información está en la instancia de BD, en la tabla `acto`.

Siempre el resultado que quiero va a ser una tabla, entonces, pensemos qué columnas y qué filas va a tener.

- **Columnas:** las mismas que la tabla `acto`.
- **Filas:** algunas de las filas de la tabla `acto`, las que cumplen con una condición, que el valor de la columna `gradoPeligrosidad` sea menor o igual a 2.

Si copiamos la tabla `acto`, podemos pintar qué filas queremos en el resultado deseado.

acto-nombre	gradoPeligrosidad
mago	0
Domadores	7
sogas	3
payasos	1
cuchillos	7
mono	2

y el resultado es

acto-nombre	gradoPeligrosidad
mago	0
payasos	1
mono	2

La operación de selección nos permite exactamente esto: a partir de una tabla, obtener otra con las mismas columnas y algunas de sus filas, las que cumplen una condición que es el otro operando de la selección.

Recordamos que la notación es

$\sigma<\text{criterio}> \text{tabla}$

en este caso la tabla es `acto`, el criterio es `gradoPeligrosidad <= 2`. Por lo tanto la forma de expresar esta consulta en álgebra relacional es

$\sigma<\text{gradoPeligrosidad} \leq 2> \text{acto}$

Ejemplo 2: actos largos

Ahora queremos las realizaciones de actos en funciones que estén en tercer lugar (o sea, que sean el tercer acto de la función), y que además duren al menos media hora.

Razonemos de una forma análoga al ejemplo anterior
¿Dónde está esa información? En la tabla `formaParte`.

1. Entonces, el resultado de lo que quiero va a tener las mismas columnas que `formaParte` ...
2. ... y algunas de sus filas, ahora la condición es doble: que la duración sea ≥ 30 , y que además el `nroDeOrden` sea exactamente 3.

La operación que describe esta consulta es

$\sigma<\text{nroOrden} = 3 \wedge \text{duracion} \geq 30> \text{formaParte}$

y el resultado es una tabla con una sola fila

fn-dia	fn-hora	acto-nombre	nroOrden	duracion
28/05/2008	19	Domadores	3	38

Tomemos ahora otra condición: queremos los actos que vayan, o bien primeros o bien terceros, y que además su duración esté entre 15 y 25 minutos.

Esta condición es más compleja que las anteriores, pero sigue siendo una condición, o sea, que esto lo podemos resolver con una selección

$\sigma<\dots> \text{formaParte}$

La parte compleja va a estar entre los corchetes. Para armarla y obtener el resultado que queremos, tenemos que tener presentes los operadores lógicos \wedge (conjunción - y) y \vee (disyunción - o), y usarlos bien. Esta consulta se puede expresar en álgebra relacional así:

$\sigma<(\text{nroOrden} = 1 \vee \text{nroOrden} = 3) \wedge (\text{duracion} \geq 15 \wedge \text{duracion} \leq 25)> \text{formaParte}$

3. Proyección - π

Definición

La proyección es una operación cuyos operandos son: una tabla y una secuencia de nombres de columna.

El resultado de una operación de proyección es una tabla que tiene

- Las mismas filas que la tabla operando.
- Las columnas que se eligieron, que deben ser todas columnas de la tabla operando.

Dicho de otra forma, la proyección nos permite quedarnos con algunas de las columnas de una tabla, descartando las otras.

Notación:

$\pi\langle\text{columnas}\rangle \text{ tabla}$

Por ejemplo el resultado de

$\pi\langle\text{ciudad-nombre, precioEntrada}\rangle \text{ funcion}$

es una tabla

- cuyas columnas son ciudad-nombre y precioEntrada (las que pedí explícitamente), y
- que tiene las mismas filas que *funcion*.

Un ejemplo: nombre y población de ciudades

Queremos el nombre y la población de cada ciudad incluida en la BD.

Esa información está en la instancia de BD, en la tabla *ciudad*.

Pensando en tablas, lo que estamos eligiendo es qué **columnas** queremos en el resultado: queremos las columnas ciudad-nombre y poblacion, la columna provincia no nos interesa. Filas, las queremos todas.

Si copiamos la tabla *ciudad*, podemos pintar qué columnas queremos en el resultado deseado.

ciudad-nombre	provincia	poblacion
Azul	Buenos Aires	70000
Tandil	Buenos Aires	120000
Tornquist	Buenos Aires	12000
Bahía Blanca	Buenos Aires	600000
Rawson	Chubut	140000
Río Gallegos	Santa Cruz	80000

y el resultado es

ciudad-nombre	poblacion
Azul	70000
Tandil	120000
Tornquist	12000
Bahía Blanca	600000
Rawson	140000
Río Gallegos	80000

Entonces, pensando en la tabla resultado de la operación, tengo

- **Columnas:** las que yo digo explícitamente.
- **Filas:** todas las filas de la tabla *ciudad*.

Esta operación, la de quedarnos con algunas columnas, es exactamente la que podemos expresar usando el operador de proyección.

Dado que la notación es

$\pi\langle\text{columnas}\rangle \text{ tabla}$

la forma de expresar esta consulta en álgebra relacional es
 $\pi\langle\text{ciudad-nombre, poblacion}\rangle \text{ ciudad}$

Otro ejemplo: actos y duraciones

Queremos ahora saber, de la composición de las funciones, los nombres de los actos y la duración de cada uno. No nos interesa de qué función, ni el número de orden.
¿Para qué? Para tener una idea de qué actos se repiten más, cuánto duran las distintas veces que se incluyen en funciones, etc..

La tabla que queremos como resultado es esta:

acto-nombre	duracion
payasos	40
mago	25
domadores	38
sogas	38
payasos	15
domadores	22
cuchillos	31

Otra vez la operación de proyección nos permite expresar lo que queremos, en este caso sobre *formaParte*. O sea:

$\pi\langle\text{acto-nombre, duracion}\rangle \text{ formaParte}$

4. Combinación de operaciones

Qué es combinar – composición

Así como en aritmética se pueden armar cuentas complejas combinando $+$ $-$ $*$ $/$, en álgebra relacional también se pueden armar consultas complejas combinando sus operadores.

Hasta ahora conocemos dos operadores: selección y proyección. Uno nos permite quedarnos con algunas filas, el otro con algunas columnas. Entonces, cuando partiendo de una tabla quiero solamente algunas columnas de algunas filas, tengo que hacer las dos cosas, una selección y una proyección.

Los operadores se combinan como la composición de funciones¹: el resultado de uno es el argumento del otro. Cuando en matemática escribo

$$\begin{array}{l} \text{o} \qquad \qquad \qquad f(g(3)) \\ \qquad \qquad \qquad f \circ g (3) \end{array}$$

lo que va a recibir f es el resultado de g . Por ejemplo, si

$$\begin{array}{l} f(x) := x^2 \\ g(x) := x + 2 \end{array}$$

entonces

$$f(g(3)) = f(5) = 25$$

En álgebra relacional lo mismo, si escribimos

$$\pi\langle\text{columna1, columna2, ...}\rangle (\sigma \langle\text{condicion}\rangle \text{tabla})$$

la tabla a la que se aplica la proyección es el resultado de la selección. Como el resultado de todas las operaciones es siempre una tabla, la combinación es fácil.

Miremos fijo esta consulta

$$\pi\langle\text{columna1, columna2, ...}\rangle (\sigma \langle\text{condicion}\rangle \text{tabla})$$

el resultado de la selección son algunas filas de la tabla original, esa tabla con menos filas le llega a la proyección, que de la tabla reducida deja solamente algunas columnas.

¡Excelente! El resultado es el que queríamos: algunas columnas de algunas filas. Ahora vamos a ver un par de ejemplos.

Cuando veamos más operadores, se van a ampliar mucho las posibilidades de combinación. Practiquemos ahora con los que tenemos.

¹ que deberían tener del polimodal o secundaria

Un ejemplo: ciudades y público en junio

Queremos saber, para las funciones que sean a las 20 horas, ciudad y público esperado. Esa información está en la instancia de BD, en la tabla `funcion`.

Para obtenerla tengo que hacer

- una selección, para quedarme con las funciones cuya hora sea 20, y
- una proyección, para quedarme solamente con las columnas ciudad-nombre y publicoEsperado.

En la instancia del ejemplo, el resultado esperado es

ciudad-nombre	publicoEsperado
Tornquist	75
Bahía Blanca	1400

Tengo dos opciones, analicémoslas

1. Selección primero, y proyección sobre el resultado de la selección:
 $\pi\langle \text{ciudad-nombre, publicoEsperado} \rangle (\sigma \langle \text{fn-hora}=20 \rangle \text{funcion})$

¿cómo se entiende esto?

Como una composición: la operación de afuera toma el resultado de la de adentro. La "de adentro" es la selección, cuyo resultado es

fn-dia	fn-hora	ciudad-nombre	precioEntrada	publicoEsperado
05/06/2008	20	Tornquist	28	75
09/06/2008	20	Bahía Blanca	16	1400

sobre **esta tabla** se va a hacer la proyección, que nos va a dar el resultado esperado

ciudad-nombre	publicoEsperado
Tornquist	75
Bahía Blanca	1400

Acá se ve la ventaja de pensar al resultado de cada operación como una tabla. Como el resultado de la selección es una tabla (que contiene las filas de `funcion` que cumplen la condición), sobre esa tabla resultado puedo hacer cualquier otra operación, en este caso la proyección que quiero.

Veamos la otra opción:

2. Al revés, proyección primero, y selección sobre el resultado de la proyección:

$$\sigma \langle \text{fn-hora}=20 \rangle (\pi \langle \text{ciudad-nombre, publicoEsperado} \rangle \text{funcion})$$

en este caso, la selección se va a hacer sobre el resultado de la proyección...

ciudad-nombre	publicoEsperado
Azul	2000
Tandil	320
Tandil	700
Tornquist	75
Bahía Blanca	1400
Río Gallegos	470

... que no tiene la columna fn-hora, por lo tanto la selección no se puede hacer.

O sea, esta opción es incorrecta. La forma de describir la operación que queremos hacer en álgebra relacional es (repetimos)

$\pi < \text{ciudad-nombre, publicoEsperado} > (\sigma < \text{fn-hora}=20 > \text{funcion})$

Ejemplo 2: ciudades grandes

Queremos saber ahora los nombres de las ciudades de más de 200000 habitantes. Otra vez, tenemos que hacer una selección (quedarnos con las ciudades grandes) y también una proyección (quedarnos con la columna ciudad-nombre). Lo resolvemos en forma similar al ejemplo anterior, proyectamos el resultado de la selección:

$\pi < \text{ciudad-nombre} > (\sigma < \text{poblacion}=200000 > \text{funcion})$

Lo que se proyecta es el resultado de la selección, que va a tener una fila sola

Ciudad-nombre	provincia	poblacion
Bahía Blanca	Buenos Aires	600000

Proyectar esta tabla nos da el resultado deseado

ciudad-nombre
Bahía Blanca

5. De nuevo: ¿qué es el álgebra relacional?

¿Quién hace las consultas?

Ahora que ya vimos dos de las operaciones más comunes del álgebra relacional, y las combinamos en consultas un poco más complejas, podemos volver un poco a pensar qué estamos haciendo.

En una instancia de BD podemos tener muchos datos. Lo bueno de tener una BD organizada según un modelo es que sabemos cómo están organizados esos datos, y esa organización no depende de la cantidad, por más que tengamos 1000 millones de filas, sabemos que van a ser filas de una tabla, qué valores va a incluir cada tabla, el dominio de cada valor, y podemos ubicar cada valor por su nombre de atributo.

En muchas circunstancias, de todos esos datos queremos extraer algunos, que son los que sirven para un objetivo particular: las funciones que incluyen payasos y domadores, los artistas que pueden hacer actos peligrosos, las funciones cuyo precio de entrada está entre 25 y 32 pesos, etc..

Eso es una **consulta**: el hecho de extraer, de todos los datos que hay, los que quiero en un momento. Este es un buen momento para recordar que el relacional está parado, en la línea conceptual – lógico – físico, en el nivel **lógico**, en particular, los detalles físicos no se incluyen. Veamos un poco qué quiere decir esto a la hora de resolver una consulta.

Para obtener el resultado de una consulta en un DBMS hay muchos detalles físicos que deben tenerse en cuenta: cómo están organizados físicamente los datos, si están ordenados, qué mecanismos de búsqueda usar, etc. .

Además de los aspectos físicos, hay un aspecto lógico muy importante: **qué datos quiero obtener, cuál es el resultado deseado de mi consulta**. Si no sé qué estoy buscando, difícil que lo encuentre.

En principio, todos los detalles físicos de la resolución de una consulta los puede resolver el DBMS, eso nos permite como usuarios focalizarnos en pensar qué queremos obtener. Ahora, los DBMS no son (tal vez todavía, pero creo que por un largo tiempo) tan capos como para que uno les hable en castellano (o en inglés) y entiendan. Hay que buscar otras formas, más matemáticas si se quiere, para poder expresar una consulta.

El álgebra relacional es exactamente eso: una forma de expresar consultas con tablas, así como la aritmética (como dijimos al principio) es una forma de expresar cuentas.

Después vamos a ver otra forma de expresar consultas, llamada SQL², que vamos a usar en un DBMS, y que va a andar en máquina y todo. El SQL está basado en el álgebra relacional, entonces todo lo que aprendamos ahora nos va a ser útil para cuando vayamos al laboratorio.

OK, quiero describir una consulta. ¿Qué hago?

Siempre conviene pensar lo que quiero obtener como una tabla, o sea, pensar en qué columnas y qué filas va a tener. Esta tabla resultado no va a ser una de las que están en la instancia de BD, va a ser otra distinta, que se va a generar a partir de los datos de las tablas. Para armar la consulta, también va a ser útil pensar en: qué tabla o tablas están los datos que quiero, y qué operación o combinación de operaciones me acerca al resultado.

Vamos con otro ejemplo: quiero número de orden y duración de las apariciones de los payasos en junio.

- La tabla resultado va a tener dos columnas: nroOrden y duracion.
- Los datos están en la tabla `formaParte`.
- Tengo que hacer una selección (para quedarme con las filas que cumplen una condición) y también una proyección (para quedarme con las columnas que quiero).
- La condición es un "y" entre: el día tiene que estar en junio, y el nombre del acto tiene que ser "payasos".
- ¿Cómo combinar? Tiene que ser proyección (selección), porque si hago adentro la proyección, la selección no tiene las columnas que necesita para seleccionar.

La consulta es

² El SQL nos permite expresar todas las operaciones que podemos hacer con una BD, tanto a nivel instancia como a nivel esquema. Más sobre esto ... cuando veamos SQL.

$\pi\langle \text{nroOrden}, \text{duracion} \rangle$
 $(\sigma \langle \text{acto-nombre} = \text{"payasos"} \wedge \text{fn-dia} \geq 01/06/2008 \wedge \text{fn-dia} \leq 30/06/2008 \rangle$
 $\text{formaParte})$

Recordemos: si pusiera

$\sigma \langle \text{acto-nombre} = \text{"payasos"} \wedge \text{fn-dia} \geq 01/06/2008 \wedge \text{fn-dia} \leq 30/06/2008 \rangle$
 $(\pi\langle \text{nroOrden}, \text{duracion} \rangle \text{ formaParte})$

estaría mal, porque el resultado de la proyección no tiene las columnas acto-nombre ni fn-dia.

Muy importante (Resultado de toda operación)

Para combinar operadores y armar una consulta compleja, es importantísimo tener siempre en cuenta la regla de oro del álgebra relacional

el resultado de toda operación es una tabla

eso hace fácil componer operaciones, como ya vimos.

6. Join natural – combinación de tablas

Lo nuevo - combinar tablas

Los dos operadores que conocemos toman una sola tabla como operando. Por lo tanto, hasta ahora solamente podemos hacer consultas que tomen sus datos de una sola de las tablas de mi instancia de BD. Por más que combine operadores, si todos toman una sola tabla, el de más adentro va a tomar una tabla, y cada uno de los siguientes el resultado del anterior.

Todavía no conocemos ninguna forma de combinar datos que estén en distintas tablas. El operador que más se usa para eso es el join natural, que vamos a ver ahora.

En pocas palabras

El join natural junta dos tablas, uniéndolas por las columnas que tengan el mismo nombre en las dos. En general, cuando pasa esto, es porque hay una relación fk-pk entre las dos tablas.

La sintaxis es bien sencilla

$\text{tabla1} * \text{tabla2}$

El resultado de este join natural es una tabla

- cuyas **columnas** son la unión de las columnas de tabla1 y tabla2 , quitando las repetidas.
- cuyas **filas** son todas las combinaciones de de las filas de las dos tablas que son correspondientes entre sí.
Dos filas en distintas tablas se dicen **correspondientes** entre sí si tienen los mismos valores en todas las columnas que se llaman igual en las dos tablas.
Esto tiene una consecuencia: las filas que no tienen correspondiente, no van a quedar incluidas en el join.

Por ejemplo: si la única columna de mismo nombre en tabla1 y tabla2 es hora, entonces en el join van a estar las combinaciones de filas de tabla1 y tabla2 que tengan el mismo valor en el atributo hora. Supongamos que las tablas tienen estos datos

tabla1

hora	otras columnas 1...
15	datos1...
18	datos2...
20	datos3...
22	datos4...

tabla2

Hora	otras columnas 2...
10	datos5...
15	datos6...
18	datos7...
18	datos8...

¿Qué pasa con cada fila de `tabla1` en `tabla1 * tabla2`?

- la primera se corresponde con la segunda de `tabla2`, esa combinación va a ir al join.
- la segunda tiene dos correspondientes: la tercera y la cuarta de `tabla2`. Entonces el join va a tener dos filas, una para cada combinación.
- la tercera y la cuarta no tienen correspondientes, por lo tanto, no van a aparecer en el join. Observamos que lo mismo pasa con la primer columna de `tabla2`.

El resultado queda así:

hora	otras columnas 1...	otras columnas 2...
15	datos1...	datos6...
18	datos2...	datos7...
18	datos2...	datos8...

Combinando join natural, selección y proyección se pueden describir muchas consultas.

Un ejemplo: cada componente de una función con su peligrosidad

Queremos una tabla igual que `formaParte`, con el agregado del grado de peligrosidad de cada acto. Pensando en columnas, va a tener entonces: todas las de `formaParte` más `gradoPeligrosidad`.

¿Cómo podemos describir "el grado de peligrosidad de cada acto" con los conceptos del modelo relacional? Veamos con una fila de `formaParte`:

fn-dia	fn-hora	acto-nombre	nroOrden	duracion
28/05/2008	19	payasos	1	40

El grado de peligrosidad del acto es 1.

¿Por qué?

Como lo diría una persona:

porque es el grado de peligrosidad de los payasos, y esta parte de la función son payasos.

Para el modelo relacional:

porque es el valor de la columna `gradoPeligrosidad` en la fila de `acto` que está "enganchada" con esta fila de `formaParte` por la relación `fk-pk`; la columna `acto-nombre` en `formaParte` es una `fk` a `acto`.

Para pensar en las operaciones que puede describir la consulta que quiero hacer, me conviene pensar en los términos del modelo: filas, tablas, columnas, pk, fk. Observo que las dos tablas tienen una columna acto-nombre, y es el único nombre que coincide. O sea, el valor que quiero es el valor de la columna gradoPeligrosidad en la fila de acto que es correspondiente con esta fila de formaParte. Como acto-nombre es la pk de acto, ninguna fila en formaParte puede tener más de una correspondiente³.

Enganchar las filas correspondientes de dos tablas ... exactamente lo que hace el join natural. La forma de expresar esta consulta es sencillamente

`formaParte * acto`

Y el resultado es

fn-dia	fn-hora	acto-nombre	nroOrden	duracion	gradoPeligrosidad
28/05/2008	19	payasos	1	40	1
28/05/2008	19	mago	2	25	0
28/05/2008	19	domadores	3	38	7
02/06/2008	18	sogas	1	38	3
02/06/2008	18	payasos	2	15	1
02/06/2008	18	domadores	3	22	7
02/06/2008	18	cuchillos	4	31	7

Todas las filas de formaParte aparecen en el join. Las de acto no: como el acto del mono no está programado en ninguna función, esa fila de acto no tiene correspondiente en formaParte, y por lo tanto no aparece en el join.

Ejemplo combinado: público vs población

Lo que queremos obtener ahora es una tabla que me muestre el público esperado y la población de la ciudad en cada función de junio.

Queremos: día, hora, ciudad, público esperado, y población de la ciudad.

La información está repartida entre funcion y ciudad, que se enganchan naturalmente por el atributo ciudad-nombre, que es el único que tiene el mismo nombre en las dos tablas. Perfecto, se da la condición para el join.

Ahora, de este join no quiero

- ni todas las filas: sólo las funciones de junio y sus correspondientes en ciudad.
- ni todas las columnas: quiero solamente las cuatro que se especificaron.

Epa, vamos a tener que hacer: un join natural, una selección, y una proyección, las tres cosas.

La proyección conviene hacerla sobre el resultado del resto. P.ej. esta operación

`funcion * (π<poblacion> ciudad)`

está mal definida, porque la proyección (o sea, la tabla que es el resultado de la proyección) solamente tiene la columna poblacion, que no está en funcion, y por lo tanto el join natural no

³ Si no se violaría la restricción de clave en acto ... es importante que se entienda, si no lo ves, detenete un poco en esta parte.

tiene sentido⁴.

Entonces, va a ser algo así como

$\pi_{\langle \text{fn-dia, fn-hora, publicoEsperado, poblacion} \rangle} (\dots \text{algo} \dots)$

Si el "algo" fuera

$\text{funcion} * \text{ciudad}$

nos estarían sobrando filas, porque solamente queremos las funciones de junio.

Entonces, en lugar de joinear todas las funciones, podemos joinear solamente las de junio, que es el resultado de una selección, o sea que el "algo" puede ser

$(\sigma_{\langle \text{fn-dia} \geq 01/06/2008 \wedge \text{fn-dia} \leq 30/06/2008 \rangle} \text{funcion}) * \text{ciudad}$

Juntar todo se hace muy largo. Podemos partir la expresión en partes, y darle un nombre a una tabla que es el resultado de una operación, y la vamos a usar en otra.

En este caso puede quedar así:

$\text{funcionesJunio} \quad (\sigma_{\langle \text{fn-dia} \geq 01/06/2008 \wedge \text{fn-dia} \leq 30/06/2008 \rangle} \text{funcion})$

$\pi_{\langle \text{fn-dia, fn-hora, publicoEsperado, poblacion} \rangle} (\text{funcionesJunio} * \text{ciudad})$

Cuando le damos un nombre a una tabla que interviene en una consulta compleja, se entiende mucho más si se le pone un nombre claro, y también si la tabla que separamos tiene un sentido que se puede explicar con un nombre.

En este caso funcionesJunio es ... exactamente eso, la tabla con las funciones de junio.

Otra vez aprovechamos que el resultado de cada operación es una tabla, que a su vez puede entrar en otra operación; componiendo podemos describir consultas complejas.

Veamos cómo son las tablas intermedias:

funcionesJunio

fn-dia	fn-hora	ciudad-nombre	precioEntrada	publicoEsperado
02/06/2008	18	Tandil	20	320
02/06/2008	21	Tandil	25	700
05/06/2008	20	Tornquist	28	75
09/06/2008	20	Bahía Blanca	16	1400

funcionesJunio * ciudad

fn-dia	fn-hora	ciudad-nombre	precioEnt rada	publicoEs perado	provincia	poblacion
02/06/2008	18	Tandil	20	320	Buenos Aires	120000
02/06/2008	21	Tandil	25	700	Buenos Aires	120000
05/06/2008	20	Tornquist	28	75	Buenos Aires	12000
09/06/2008	20	Bahía Blanca	16	1400	Buenos Aires	600000

Vemos que las filas de ciudad que no tienen correspondencia en funcionesJunio no entran en el join.

⁴ En realidad sí está definido el resultado de un join natural entre tablas que no comparten ningún nombre de atributo, pero lo vamos a contar un poco más adelante, cuando veamos otras operaciones que también sirven para juntar tablas.

El resultado final es

fn-dia	fn-hora	publicoEsperado	poblacion
02/06/2008	18	320	120000
02/06/2008	21	700	120000
05/06/2008	20	75	12000
09/06/2008	20	1400	600000