

Álgebra Relacional

Bibliografía recomendada: Capítulo 4, Libro Ramakrishnan

Repaso

- *Álgebra Relacional* (AR) es un lenguaje de consulta asociado al Modelo Relacional (MR).
- Una consulta de AR recibe como input *instancias de relación*.
- A su vez, una consulta de AR devuelve también una *instancia de relación*.
- Recordar que las instancias de relación son conjuntos de tuplas → las consultas de AR no tendrán repetidos.
- Una consulta de AR describe paso a paso (componiendo operadores) cómo se llega a calcular el resultado.

Operadores

Básicos

- Proyección $\pi_{r1}(R)$
- Selección $\sigma_{r1='abc'}(R)$

Conjuntos

- Unión $R \cup S$
- Intersección $R \cap S$
- Resta $R - S$
 - Estos 3 operadores exigen que las relaciones sean unión-compatibles → misma cantidad de atributos y cada atributo (de izquierda a derecha) tiene mismo dominio
 - Los nombres de los atributos de la relación resultante son los de la 1° relación
- Producto cartesiano $R \times S$

Joins

- Join con condición $R \bowtie_{\text{cond}} S$
- Equijoin $R \bowtie_{R.i=S.sj \wedge \dots} S$
 - Formado por conjunción de igualdades entre atributos de ambas relaciones
 - Remueve el atributo de la derecha en la relación resultado
- Natural join $R \bowtie S$
 - Junta por atributos con el mismo nombre
 - Deja un solo atributo de los repetidos

Avanzados

- Renombre ρ

Ejemplo:

$A(id)$

$\rho(1 \rightarrow id1, 2 \rightarrow id2, AxA)$

$\rho(B, AxA)$

$\rho(B(1 \rightarrow id1, 2 \rightarrow id2), AxA)$

- División $R \div S$

➤ Conviene siempre trabajar con R de 2 atributos y S de 1 solo atributo

Ejemplo:

R		S		R \div S
a1 b1		b1		
a1 b2		b2		
a1 b3	\div	b3	=	a1
a2 b2				
a3 b3				

Ejercicios

ACTOR(idActor, nombreActor, edad)
SERIE(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
GÉNERO(idGénero, nombreGénero)
CANAL(idCanal, nombreCanal)
PARTICIPA_EN(idActor, idSerie)
TRANSMITE(idCanal, idSerie)

1. Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends”.

Primero, nos vamos a parar en la relación PARTICIPA_EN y, mediante un join, vamos a “añadirle” toda la información de la serie y del actor.

$$\rho(\mathbf{PAS}, \mathbf{PARTICIPA_EN} \bowtie \mathbf{ACTOR} \bowtie \mathbf{SERIE})$$

Luego, filtramos las tuplas que nos interesan.

$$\rho(\mathbf{PAS}', \sigma_{\text{nombreSerie} = \text{"Friends"} \wedge \text{edad} > 30}(\mathbf{PAS}))$$

Finalmente, proyectamos sólo el nombre.

$$\rho(\mathbf{RESULTADO}, \pi_{\text{nombreActor}}(\mathbf{PAS}'))$$

2. Listar los nombres de los canales que transmiten todas las series de comedia.

Analizando el enunciado, nos damos cuenta de que la consulta puede resolverse con una división. Por lo tanto, la idea será llegar a tener una relación con 2 atributos, otra de 1 atributo para luego aplicar la división entre ellas.

Comenzamos obteniendo las series de comedia y le proyectamos el id (relación de 1 atributo).

$$\rho(\mathbf{ID_SERIE_COMEDIA}, \pi_{\text{idSerie}}(\sigma_{\text{nombreGénero} = \text{"Comedia"}}(\mathbf{SERIE} \bowtie \mathbf{GÉNERO})))$$

Ya tenemos la relación de 2 atributos que buscábamos: TRANSMITE. Entonces, aplicamos la división.

$$\rho(\mathbf{ID_CANAL_TRANSMITE_TODAS_COMEDIAS}, \mathbf{TRANSMITE} \% \mathbf{ID_SERIE_COMEDIA})$$

Una vez hecha la división, obtenemos los nombres de los canales que cumplen todo lo anterior.

$$\rho(\mathbf{RESULTADO}, \pi_{\text{nombreCanal}}(\mathbf{ID_CANAL_TRANSMITE_TODAS_COMEDIAS} \bowtie \mathbf{CANAL}))$$

3. Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además participaron alguna vez en series que comenzaron luego del año 2000.

Debemos conseguir a actores que cumplan lo mismo que en el ejercicio 1 y que además cumplan otra condición.

La primera parte ya es conocida.

$$\rho(\text{ID_ACTOR_FR_}>_30, \\ \pi_{\text{idActor}}(\sigma_{\text{nombreSerie} = \text{“Friends”} \wedge \text{edad} > 30} (\text{PARTICIPA_EN} \bowtie \text{ACTOR} \bowtie \text{SERIE})))$$

Queremos obtener los actores que participaron al menos una vez en series que comenzaron luego del año 2000.

$$\rho(\text{ID_ACTOR_DESP_2000}, \pi_{\text{idActor}}(\sigma_{\text{añoInicio} > 2000} (\text{PARTICIPA_EN} \bowtie \text{SERIE})))$$

Ahora, simplemente intersecamos ambos conjuntos y hacemos un join para poder devolver los nombres.

$$\rho(\text{ID_ACTOR_1_Y_2}, \text{ID_ACTOR_FR_}>_30 \cap \text{ID_ACTOR_DESP_2000}) \\ \rho(\text{RESULTADO}, \pi_{\text{nombreActor}}(\text{ID_ACTOR_1_Y_2} \bowtie \text{ACTOR}))$$

4. Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además **nunca** participaron en series que comenzaron luego del año 2000.

El ejercicio es similar al anterior, solamente que ahora queremos los que nunca participaron en ciertas series.

Para la primera parte, reutilizamos la relación **ID_ACTOR_FR_>_30** del ejercicio anterior.

Luego, buscamos obtener los actores que nunca participaron en series que comenzaron luego del año 2000. Atacaremos el problema *negando* lo que nos piden. Es decir, obtendremos un conjunto con los actores que sí participaron en series que se iniciaron luego del 2000 y luego al conjunto de todos los actores le restaremos ese conjunto.

$$\rho(\text{ID_ACTOR_DESP_2000}, \pi_{\text{idActor}}(\sigma_{\text{añoInicio} > 2000} (\text{PARTICIPA_EN} \bowtie \text{SERIE}))) \\ \rho(\text{ID_ACTOR_NO_DESP_2000}, \pi_{\text{idActor}}(\text{ACTOR}) - \text{ID_ACTOR_DESP_2000})$$

Finalmente, intersecamos ambos conjuntos y hacemos un join para poder devolver los nombres.

$$\rho(\text{ID_ACTOR_1_Y_2}, \text{ID_ACTOR_FR_}>_30 \cap \text{ID_ACTOR_NO_DESP_2000}) \\ \rho(\text{RESULTADO}, \pi_{\text{nombreActor}}(\text{ID_ACTOR_1_Y_2} \bowtie \text{ACTOR}))$$

5. Listar los ids de las series que comenzaron más recientemente.

Este ejercicio nos está pidiendo que obtengamos las series con mayor fecha de comienzo (es como calcular un *máximo*, pero en AR¹ no contamos con funciones de agregación que nos lo calculen). Para resolver esto vamos a usar una estrategia similar a la del ejercicio anterior (*negar* lo que nos piden, llevando todo a un *existencia*). Vamos a obtener las series que tienen alguna que comenzó después (o sea, no son las más recientes) y le restaremos todas las series a ese conjunto.

$$\begin{aligned} & \rho(\text{SERIE_1}, \text{SERIE}) \\ & \rho(\text{SERIE_2}, \text{SERIE}) \\ & \rho(\text{ID_SERIE_INICIO_NO_MÁS_RECIENTE}, \\ & \pi_{\text{SERIE_1.idSerie}}(\sigma_{\text{SERIE_1.añoInicio} < \text{SERIE_2.añoInicio}} (\text{SERIE_1} \times \text{SERIE_2})) \end{aligned}$$

Finalmente, a todas las series les restamos lo anterior.

$$\rho(\text{RESULTADO}, \pi_{\text{idSerie}}(\text{SERIE}) - \text{ID_SERIE_INICIO_NO_MÁS_RECIENTE})$$

6. Listar los ids de los actores que participaron en al menos 2 series.

Como antes, en AR no contamos con funciones de agregación que nos ayuden a contar, por lo tanto debemos utilizar otra alternativa. Una estrategia posible es hacer un join de la tabla PARTICIPA_EN consigo misma y pedir que el actor sea el mismo pero que la serie no. De esta forma, obtendremos actores que participan en 2 o más series.

$$\begin{aligned} & \rho(\text{P_1}, \text{PARTICIPA_EN}) \\ & \rho(\text{P_2}, \text{PARTICIPA_EN}) \\ & \rho(\text{RESULTADO}, \pi_{\text{P_1.idActor}}(\sigma_{\text{P_1.idActor} = \text{P_2.idActor} \wedge \text{P_1.idSerie} \neq \text{P_2.idSerie}} (\text{P_1} \times \text{P_2})) \end{aligned}$$

Ejercicios pendientes

(Para que cada uno lo piense solo)

- Resolver el último ejercicio suponiendo que piden devolver los que cumplen “exactamente 2”
(Hint 1: Al menos 2 = $x \geq 2$)
(Hint 2: $x = 2 \Leftrightarrow x \geq 2 \wedge x < 3 \Leftrightarrow x \geq 2 \wedge \neg x \geq 3$)
- Reescribir $R \% S$ con los demás operadores (asuma que R es de 2 atributos y S de 1)

¹Existe bibliografía donde se extiende al AR clásico para agregar funciones de agregación, pero en la materia no lo utilizaremos.