CARRERA DE INGENIERÍA DE SISTEMAS

Ingeniería de Datos

Introducción a operaciones formales de Algebra Relacional.

UNIVERSIDAD DE LIMA

Agenda

- Introducción Algebra Relacional
- Algebra Relacional
- Operadores del Algebra Relacional
- Ejercicios

Introducción

Los lenguajes relacionales son lenguajes formales y se basan en el carácter conjuntista de una relación.

Introducción

El álgebra relacional es un conjunto de operaciones que describen paso a paso como calcular una respuesta sobre las relaciones, como son definidas en el modelo relacional. Denominada de tipo procedimental, a diferencia del Cálculo relacional que es de tipo declarativo.

Describe el aspecto de la manipulación de datos. Estas operaciones se usan como una representación intermedia de una consulta a una base de datos y, debido a sus propiedades algebraicas, sirven para obtener una versión más optimizada y eficiente de dicha consulta.

Algebra relacional

El álgebra relacional ***es un lenguaje procedimental*** para la manipulación de relaciones:

- En esta se especifica paso por paso la respuesta a una consulta de los datos contenidos en una relación.

El cálculo relacional es un ***lenguaje aprocedimental :***

- En el cálculo relacional una consulta se resuelve en un solo paso.

El Algebra Relacional fue desarrollada en 1970-72 y el cálculo relacional en 1971 por Codd.

Lenguajes relacionales

Tanto el álgebra y el cálculo relacional proveen una forma teórica de manipular una base de datos relacional.

Codd demostró que el álgebra relacional y el cálculo relacional son ***lógicamente equivalentes***.

El álgebra relacional es importante por que contribuyó a establecer un vocabulario común que encontramos en los lenguajes de bases de datos comerciales, tales como SELECT, PROJECT, etc

El cálculo relacional es importante porque está basado en lógica de predicados.

Operadores del Algebra relacional

Colección de **operadores** :

- **Operadores tradicionales sobre conjuntos**
  - unión
  - intersección
  - diferencia
  - producto cartesiano

  - Los operandos son relaciones, y NO conjuntos arbitrarios
  - operaciones adaptadas a relaciones (tipo especial de conjuntos)

- **Operadores relacionales especiales**
  - restricción
  - proyección
  - reunión (join)
  - división

Clausura relacional

“ El resultado de cualquier operación del álgebra relacional es otra relación ”

** la salida de una operación puede ser entrada (operando) de otra

Unión compatible

- En matemáticas, $\mathbf{A \cup B = \{ e \ / \ e \in A}$ y-o $\mathbf{e \in B \}}$
- Relación = conjunto de tuplas
  ⇨ es posible hacer la unión de dos relaciones R y S
- $\mathbf{R \cup S = \{ t \ / \ t \in R}$ y-o $\mathbf{t \in S \}}$
  - Conjunto de todas las tuplas que están en R y/o en S
  - Sin embargo...
    PELICULA ∪ DIRECTOR es un conjunto, pero **no es una relación**

Las relaciones deben ser homogéneas: no pueden contener mezcla de tuplas de distintos tipos

- Ha de mantenerse la Propiedad de Clausura:
  el resultado de la operación DEBE ser una relación
- Las **relaciones** de entrada deben ser **de tipos compatibles**

Compatibilidad de tipos

Sean R (r_1, r_2,..., r_n), S (s_1, s_2, ..., s_n) **Relaciones** R y S **compatibles en tipo** si tienen el "mismo" esquema, es decir:

- ***Igual número de atributos***:
  grado(R) = grado(S) = n
- ***Atributos correspondientes definidos sobre el mismo dominio***:
  $dom(r_i) = dom(s_i)$,, i = 1, 2, ..., n

Ejemplo: DIRECTOR y DIR_FOTOGRAFICO son de tipos compatibles

- Los operadores de ***unión, intersección, diferencia*** necesitan operandos compatibles en tipo.
- El producto cartesiano no necesita compatibilidad de tipo en sus operandos.

Algebra relacional

Unión de relaciones

R∪S, con R y S compatibles en tipo, es una **relación** tal que:

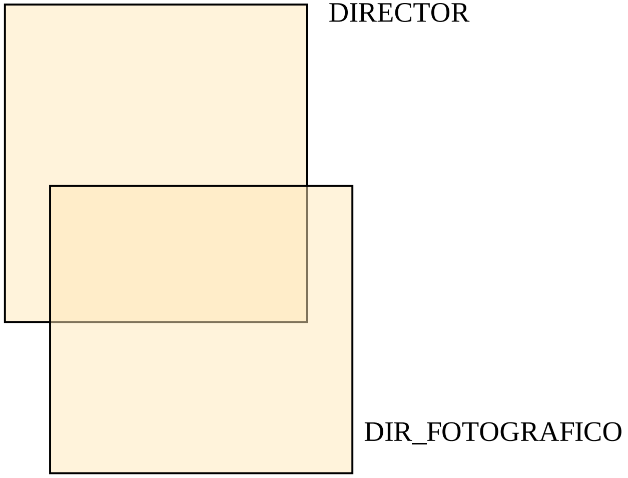
Esquema: **el de R (o S)**

Estado: **conjunto de tuplas que están en R, en S o en ambas**

Las tuplas repetidas se eliminan (por definición)

Ejemplo:

DIRECTOR ∪ DIR_FOTOGRAFICO



Unión

$$R = E \cup P$$

E	código	Nombre
	9.644.667	Pedro Pérez
	10.133.212	Gabriel Mendoza
	11.332.334	Luis Colina
	12.562.884	Andrés Rojas
	20.126.112	Gilberto Zapata

P	código	Nombre
	8.347.223	Hector Redondo
	9.644.667	Pedro Pérez
	10.133.212	Gabriel Mendoza
	11.332.334	Luis Colina
	12.123.231	Diego Dávila

R	código	Nombre
	8.347.223	Hector Redondo
	9.644.667	Pedro Pérez
	10.133.212	Gabriel Mendoza
	11.332.334	Luis Colina
	12.123.231	Diego Dávila
	12.562.884	Andrés Rojas
	20.126.112	Gilberto Zapata

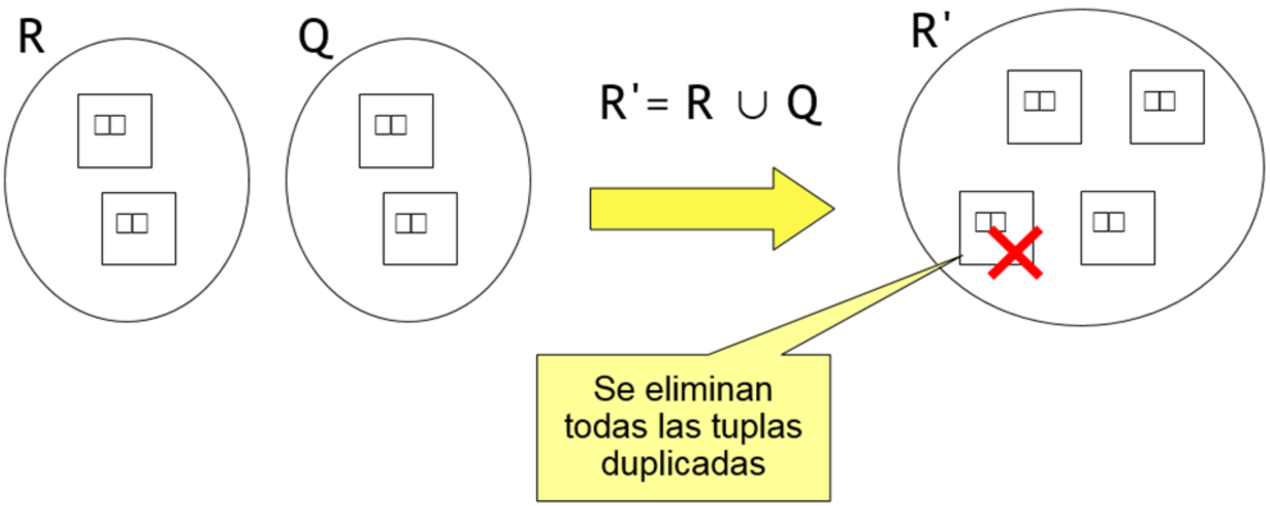
Se realiza una operación de unión de conjuntos. Se eliminan las tuplas repetidas

Las relaciones usadas como operandos deben ser compatibles entre si

* Se eliminan todas las tuplas duplicadas

Union

El resultad es todas las tuplas de ambas relaciones



Algebra relacional

Intersección de relaciones

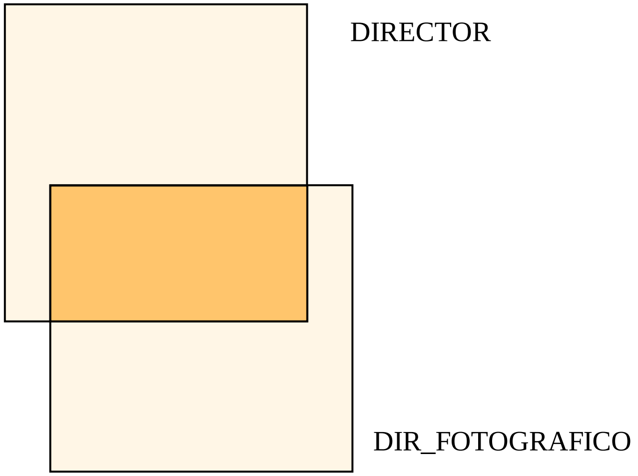
R∩S, con R y S compatibles en tipo, es una **relación** tal que:

Esquema: **el de R (o S)**

Estado: **conjunto de tuplas que están a la vez en R y en S**

Ejemplo:

DIRECTOR ∩ DIR_FOTOGRAFICO



Intersección

$$R = E \cap P$$

E	código	Nombre
	9.644.667	Pedro Pérez
	10.133.212	Gabriel Mendoza
	11.332.334	Luis Colina
	12.562.884	Andrés Rojas
	20.126.112	Gilberto Zapata

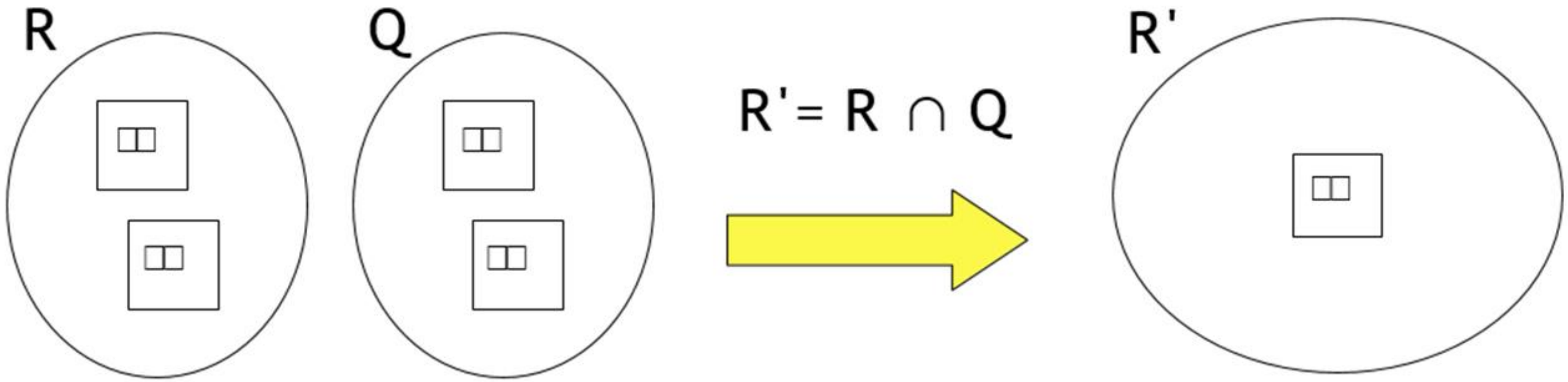
P	código	Nombre
	8.347.223	Hector Redondo
	9.644.667	Pedro Pérez
	10.133.212	Gabriel Mendoza
	11.332.334	Luis Colina
	12.123.231	Diego Dávila

R	código	Nombre
	9.644.667	Pedro Pérez
	10.133.212	Gabriel Mendoza
	11.332.334	Luis Colina

Las relaciones usadas como operandos deben ser compatibles entre si

Intersección

Sólo se consideran las tuplas comunes en ambas relaciones.



Algebra relacional

Diferencia entre relaciones

R—S, con R y S compatibles en tipo, es una **relación** tal que:

Esquema: **el de R (o S)**

Estado: **conjunto de tuplas que están en R, pero NO en S**

Ejemplo:

DIRECTOR — DIR_FOTOGRAFICO

Secuencias de operaciones: La propiedad de clausura relacional permite aplicar una operación tras otra .

$R \cap (S \cup T)$

$A \leftarrow S \cup T$

$B \leftarrow R \cap A$

Diferencia

$$R = E - P$$

E	código	Nombre
	9.644.667	Pedro Pérez
	10.133.212	Gabriel Mendoza
	11.332.334	Luis Colina
	12.562.884	Andrés Rojas
	20.126.112	Gilberto Zapata

P	código	Nombre
	8.347.223	Hector Redondo
	9.644.667	Pedro Pérez
	10.133.212	Gabriel Mendoza
	11.332.334	Luis Colina
	12.123.231	Diego Dávila

R	código	Nombre
	12.562.884	Andrés Rojas
	20.126.112	Gilberto Zapata

Las relaciones usadas como operandos deben ser compatibles entre si

Algebra relacional

Asignación de relación

No sólo permite darle nombre a una relación sino también cambiar el nombre de sus atributos :

R2 (nom_nuevo1, nom_nuevo2) ← R1

Donde:

- R1 tiene atributos nombre1 y nombre 2
- nombre1 y nom_nuevo1 están definidos sobre el mismo dominio
- nombre2 y nom_nuevo2 están definidos sobre el mismo

Algebra relacional

Asignación de relación

- Por defecto, los atributos de la relación resultado de una operación **heredan** los **nombres** de los del operando más a la izquierda

$$DIR \leftarrow DIRECTOR \cup DIR\_FOTOGRAFICO$$

* Los atributos de DIR tienen los mismos nombres que los de DIRECTOR

- Se puede indicar una **lista con nuevos nombres para los atributos de la relación resultado**:

DIR(codDir,nomDir,apeDir,nacDir,fechaNac,pelic) ← DIRECTOR ∪ DIR_FOTOG

Algebra relacional

Renombrar relación

No sólo permite renombrar el nombre de la relación sino también el nombre de sus atributos:

$\rho_{S(B1,B2,\ \ldots,\ Bn)}\ (R)$ → Renombra la relación R y sus atributos

$\rho_{S}\ (R)$ o → Renombra únicamente la relación

$\rho_{(B1,B2,\ \ldots,\ Bn)}\ (R)$ → Renombra sólo los atributos

Donde:

- R es el nombre de la relación original.
- S es el nuevo nombre de la relación.
- B1, B2, …, Bn son los nuevos nombres de los atributos. Si los atributos de R son (A1 ,A2, …, An) en ese orden, entonces cada Ai se renombra como Bi .

Algebra relacional

Producto cartesiano

En matemáticas, $\mathbf{A \times B = \{ (a,b) \ / \ a \in A}$ y $\mathbf{b \in B \}}$

Relación = conjunto de tuplas,

- es posible el producto cartesiano entre relaciones R y S

$$\mathbf{R \times S = \{ (t_R, t_S) \ / \ t_R \in R \text{ y } t_S \in S \}}$$

- Conjunto de pares ordenados de tuplas de R y S
- Debe de conservarse la Propiedad de Clausura

Algebra relacional

Producto cartesiano entre Relaciones

R X S, con R y S cualesquiera, es una **relación** tal que:

Esquema: **combinación (unión) de los esquemas de R y S**

Estado: **conjunto de todas las tuplas formadas por las posibles combinaciones de cada tupla de R con cada tupla de S**

Ejemplo: PELICULA X DIRECTOR

Obtiene un conjunto de tuplas tales que cada una es la combinación de una tupla de PELICULA y otra de DIRECTOR

Es una Operación sin demasiada importancia práctica:

- No se tiene más información a la salida que a la entrada
- pero es **necesaria para definir** la operación **Reunión** (JOIN)

Algebra relacional

Producto cartesiano entre Relaciones

El **esquema** de la relación resultante de **R X S** debe estar **bien formado** (nombres de atributos únicos)

Si R y S tienen atributos con igual nombre, R X S tendría ¡dos atributos nombrados igual!

ACTOR X AGENCIA ⇨ “colisión” de nombres en atributo “nombre”

Soluciones posibles:

1. **Renombrar atributos** de una relación, antes del producto

   AGENCIA_2(codAge, nomAge, direccion, telefono) $\leftarrow$ AGENCIA
   RESULTADO $\leftarrow$ ACTOR X AGENCIA_2

2. **Prefijar** atributos con el nombre de su tabla, en la tabla resultado

   RESULTADO(codA, ACTOR.nombre, nomreal,..., codAg, AGENCIA.nombre, ...) $\leftarrow$ ACTOR X AGENCIA

Producto Cartesiano

Carro = R x Q

R	placa	*marca*
	ADA89A	Fiat
	LBF78G	Toyota
	XSA67D	Ford

Q	*marca*	modelo	color
	Fiat	siena	gris
	Toyota	corollaXL	blanco
	Ford	Ka	rojo

Carro	placa	marcaR	marcaQ	modelo	color
	ADA89A	Fiat	Fiat	siena	gris
	ADA89A	Fiat	Toyota	corollaXL	blanco
	ADA89A	Fiat	Ford	Ka	rojo
	LBF78G	Toyota	Fiat	siena	gris
	LBF78G	Toyota	Toyota	corollaXL	blanco
	LBF78G	Toyota	Ford	Ka	rojo
	XSA67D	Ford	Fiat	siena	gris
	XSA67D	Ford	Toyota	corollaXL	blanco
	XSA67D	Ford	Ford	Ka	rojo

No es relevante para el producto cartesiano, pero es notable, que en algunas tuplas marcaR = marcaQ

Propiedades de los operadores relacionales

R, S, T relaciones de tipos compatibles

- **Asociativa**

$$(R \cup S) \cup T \equiv R \cup (S \cup T) \equiv R \cup S \cup T$$
$$(R \cap S) \cap T \equiv R \cap (S \cap T) \equiv R \cap S \cap T$$
$$(R \times S) \times T \equiv R \times (S \times T) \equiv R \times S \times T$$

- **Conmutativa**

$$R \cup S \equiv S \cup R$$
$$R \cap S \equiv S \cap R$$
$$R \times S \equiv S \times R$$

* La diferencia **no** cumple ninguna de estas propiedades

Algebra relacional

Selección σ

Obtiene un subconjunto de las tuplas de una relación para las cuales se satisface una condición de selección

$$\sigma_{\text{<condición>}} (\text{<relación>})$$

- Resultado: **Relación** (conjunto de tuplas) con **atributos** de **<**relación**>**
- **<condición>** es una expresión booleana…
    - Especificada en términos de atributos de <relación>
    - Compuesta por una o más **cláusulas**, del tipo:

<nomAtrib> <opComp> <cte> o bien **<nomAtrib> <opComp> <nomAtrib>**

- <opComp> operador de comparación $\in \{=, <, \leq, >, \geq, \neq\}$
- <cte> valor constante $\in$ dominio del atributo <nomAtrib>
- Cláusulas conectadas por operadores booleanos **AND**, **OR**, **NOT**

Algebra relacional

Selección σ

Ejemplos:

* Tuplas de actores representados por la agencia número 2

$$\sigma_{agencia=2}\ (ACTOR)$$

* Actores cuyo sueldo se mayor a 30000

$$\sigma_{sueldo>30000}\ (ACTOR)$$

* Actores representados por la agencia número 2, cuyo sueldo no llega a los 25000, o bien por la agencia 4 y con sueldo superior a 35000

$$\boldsymbol{\sigma}_{\mathbf{(agencia=2\ AND\ sueldo<25000)\ OR\ (agencia=4\ AND\ sueldo>35000)}}\ (ACTOR)$$

Algebra relacional

Selección σ

La operación de selección es **conmutativa**

$$\sigma_{cond\mathbf{1}}\left(\sigma_{cond\mathbf{2}}(R)\right) \equiv \sigma_{cond\mathbf{2}}\left(\sigma_{cond\mathbf{1}}(R)\right)$$

Esto permite …

- Secuencia de restricciones (selecciones) en **cualquier orden**
- **Combinación** de una **secuencia de restricciones** en una **única restricción** con una condición conjuntiva:

$$\sigma_{cond\mathbf{1}}(\sigma_{cond\mathbf{2}}(\ldots(\sigma_{cond\mathbf{n}}(R))\ldots)) \equiv \sigma_{cond\mathbf{1}\ \mathbf{AND}\ cond\mathbf{2}\ \mathbf{AND}\ldots\mathbf{AND}\ cond\mathbf{n}}(R)$$

Selección σ

$$R = \sigma_{marca=\text{'Ford'}}(Carro)$$

Carro	placa	marca	color
	MBO34L	Ford	verde
	LDA75K	Toyota	blanco
	ADA89A	Fiat	gris
	LBF78G	Toyota	blanco
	XSA67D	Ford	rojo

R	placa	marca	color
	MBO34L	Ford	verde
	XSA67D	Ford	rojo

$$R = \sigma_{marca=\text{'Ford'} \wedge color=\text{'rojo'}}(Carro)$$

Carro	placa	marca	color
	MBO34L	Ford	verde
	LDA75K	Toyota	blanco
	ADA89A	Fiat	gris
	LBF78G	Toyota	blanco
	XSA67D	Ford	rojo

R	placa	marca	color
	XSA67D	Ford	rojo

Algebra relacional

Proyección Π

Sólo interesan algunos atributos de una relación, Se **proyecta** la relación sobre esos atributos.

$$\Pi_{\textbf{<listAtrib>}}\textbf{(}\text{<relación>}\textbf{)}$$

- Resultado: **Relación** (conjunto de tuplas) cuyos **atributos** son **sólo** los de **<listAtrib>** y **en** ese **orden**
- **<listAtrib>** lista de nombres de atributos de <relación>

Selección vs. Proyección :

- σ selecciona algunas **tuplas** de la relación y desecha otras
- Π selecciona ciertos **atributos** y desecha los demás

Ejemplo :Obtener el código, nombre y el sueldo de todos los actores

$$\Pi_{\text{codA, nombre, sueldo}}(\text{ACTOR})$$

Algebra relacional

Proyección Π

Si <listAtrib> **no** contiene atributos clave produce tuplas repetidas.

* Obtener la agencia y la nacionalidad de todos los actores

$$\Pi_{\text{agencia, nacionalidad}}(\text{ACTOR})$$

Grado(Relación Resultado) **=** Nº atributos(<listAtrib>)

Nº Tuplas(Relación Resultado) ≤ Nº Tuplas(Relación Origen)
y es igual (**=**) si <listAtrib> contiene una clave candidata

Algebra relacional

Proyección Π

- La operación proyección **no** es **conmutativa**

$$\Pi_{lista1}\left(\Pi_{lista2}(R)\right) \neq \Pi_{lista2}\left(\Pi_{lista1}(R)\right)$$

- Además, siempre que lista1 $\subseteq$ lista2, entonces...

$$\Pi_{lista1}\left(\Pi_{lista2}(R)\right) = \Pi_{lista1}(R)$$

Algebra relacional

Reunión o Join

- <**condición de reunión**>
  - Expresión booleana especificada en términos de atributos de A y B
  - **Evaluada para cada** combinación (**par**) **de tuplas**:

    Si la cumplen, forman **una** nueva tupla de la relación resultado
  - Es de la forma:

    <condición> AND <condición> AND... AND <condición>

    donde:

    <condición> tiene la forma **ai θ bj** (condición de reunión general), y

    - ai es un atributo de A; bj es un atributo de B,
    - Dominio(ai) = Dominio(bj),
    - θ (theta) cumple que $\theta \in \{=, <, \leq, >, \geq, \neq\}$

Algebra relacional

Reunión o Join

La reunión más común es la que implica **comparación de igualdad** ($\theta \equiv =$) ⇨ **EQUI-REUNIÓN** (o REUNIÓN)

Ejemplo : Actores y agencias que los representan

$$\text{ACTOR} \bowtie_{\text{agencia=codAg}} \text{AGENCIA}$$

Problema de **colisión de nombres** de atributos

- Existen atributos nombrados igual en ACTOR y AGENCIA
- Resultado con varios atributos de igual nombre

Soluciones alternativas posibles:

1. Previo **renombramiento** de atributos de una relación

   AGENC(codAg, nomAg, dirAg, tel) $\leftarrow$ AGENCIA

   $R \leftarrow \Pi_{\text{nombre, nomAg}}(\text{ACTOR} \bowtie_{\text{agencia=codAg}} \text{AGENC})$

2. **Prefijar** atributos con el nombre de su tabla

   $R \leftarrow \Pi_{\text{ACTOR.nombre,AGENCIA.nombre}}(\text{ACTOR} \bowtie_{\text{agencia=codAg}} \text{AGENCIA})$

Algebra relacional

Reunión o Join

- Las **tuplas** cuyos **atributos** de **reunión** son **nulos, NO aparecen** en la relación **resultado**
  - Los actores que se auto-representan tienen NULL en atributo agencia
  - Sus tuplas no aparecen en ACTOR $\bowtie_{\text{agencia=codAg}}$ AGENC

- Las **tuplas** de una relación que **no** encuentran **correspondencia** en la otra, **tampoco** aparecen en la relación resultado
  - Los actores que no han actuado en ninguna película, no aparecen en ninguna tupla de la tabla ACTUA_EN
  - Sus tuplas no aparecen en ACTOR $\bowtie_{\text{codA=actor}}$ ACTUA_EN

Algebra relacional

Reunión o Join

- En general, sea A con n_A tuplas y B con n_B tuplas, entonces

$$\mathbf{R} \leftarrow \mathbf{A} \bowtie_{\textbf{<condición de reunión>}} \mathbf{B} \quad \text{cumple que} \quad \mathbf{0 \leq n_R \leq n_A * n_B}$$

- Si **ninguna combinación de tuplas** de A y B **cumple** la **<condición de reunión>**, entonces
  - Relación Resultado = **Relación vacía** (cero tuplas)

- Si **NO se especifica <condición de reunión>**, entonces
  - la <condición de reunión> es TRUE **para todas las tuplas**, y
  - $\bowtie \equiv \mathrm{X}$ (REUNIÓN $\equiv$ PROD. CARTESIANO $\equiv$ REUNIÓN CRUZADA)

Algebra relacional

Reunión o Join

- **Combina** las **tuplas relacionadas** de dos relaciones **en una sola tupla**
- Permite procesar **vínculos entre relaciones**

Ejemplo : Datos de **películas** junto con los de **su director** correspondiente

- Es necesario combinar cada tupla de PELÍCULA, p, con la tupla DIRECTOR, d, tal que el valor de codDir en d coincida con el de director en p
- Se consigue aplicando la operación REUNIÓN a las dos relaciones

$$R1 \leftarrow PELICULA \bowtie_{director=codDir} DIRECTOR$$

Algebra relacional

Reunión o Join

PELICULA (codP, título, año, genero, guión, director, directorFotog, distrib, nacio, estreno, numOscar, taquilla)

DIRECTOR (codDir, nombre, apellido, nacio, fechaNacim, óperaPrima)

Ejercicio: Obtener los Títulos de películas junto con nombre y apellido de su director

- Se consigue aplicando la operación REUNIÓN a las dos relaciones
- Y proyectando el resultado sobre los atributos requeridos

$$R2 \leftarrow \Pi_{titulo,nombre,apellido} (PELICULA \bowtie_{director=codDir} DIRECTOR)$$

R2	titulo	nombre	apellido
	La caja 507	Enrique	Urbizu
	Mensaka	Salvador	Gª Ruiz
	El viaje de Carol	Imanol	Uribe
	Airbag	Juanma	Bajo Ulloa

Reunión o Join

$$R = \text{Prof} \bowtie_{dni=\,dni\_jefe} \text{Dpto}$$

Profesor	Dni	NombreP	CodigoDpto
	6274445	José Mendez	01
	7422114	Juán Zapata	01
	8347223	Hector Redondo	02
	9644667	Pedro Pérez	02
	11332334	Luis Colina	
	12123231	Diego Dávila	03

Dpto	Código	NombreD	Dni_jefe
	01	Computación	6274445
	02	Investigación	
	03	Control	12123231

R	Dni	NombreP	CódigoDpto	Código	NombreD	Dmi_Jefe
	6274445	José Mendez	01	01	Computación	6274445
	12123231	Diego Dávila	03	03	Control	12123231

¿Dónde están los demás profesores?
¿Dónde está el departamento de Investigación?

Algebra relacional

Reunión natural *

A⁕B

- Caso particular de reunión, quizá el más importante
- No «necesita» especificar condición de reunión,
- Igual **todos** los pares de **atributos** con **igual nombre** en **A** y **B**
  - Es una **EQUI-REUNIÓN** + eliminación de atributos superfluos
    - Sólo **conserva un atributo** de **reunión**
- La **definición estándar de reunión natural** exige que los:
  - **atributos de reunión** deben tener **nombre idéntico** en ambas relaciones operando, si no es así, aplicar antes un **renombramiento** de atributos.
  - **atributos** deben tener el **mismo dominio**

Reunión natural

R(a, b, c)

S(b, d)

R	a	b	c
	10	1	100
	20	3	100
	30	5	300

S	b	d
	3	-4
	1	-5

$T1 \leftarrow R \bowtie_{R.b=S.b} S$, tiene el esquema T1 (a, R.b, c, S.b, d)

T1	a	R.b	c	S.b	d
	10	1	100	1	-5
	20	3	100	3	-4

$T2 \leftarrow R \ast S$, tiene el esquema T2 (a, b, c, d)

T2	a	b	c	d
	10	1	100	-5
	20	3	100	-4

Algebra relacional

Reunión natural *

Ejemplos:

1. Título de todas las películas junto con el título y resumen de su guión

GUIO(**guion**, **titGuion**, resumen, nomAutorPpal, fechaFin, fechaEntrega) $\leftarrow$ GUION
RESUMEN $\leftarrow \prod_{\text{titulo, titGuion, resumen}}$ (PELICULA ✱ GUIO)

2. Títulos de películas junto con el nombre y apellidos de su director

DIREC(**director**, nombre, apellidos, nacio, fechaNacim, operaPrima) $\leftarrow$ DIRECTOR
PELI_DIRE $\leftarrow \prod_{\text{titulo, nombre, apellidos}}$ (PELICULA ✱ DIREC)

3. Nombre de actores y de las agencias que los representan

AGENC(**agencia**, **nomAg**, direccion, telefono) $\leftarrow$ AGENCIA
ACT_AGEN $\leftarrow \prod_{\text{nombre, nomAg}}$ (ACTOR ✱ AGENC)

Algebra vs Cálculo

Cálculo Relacional	vs. Álgebra Relacional
- Expresiones **Declarativas** (lenguaje no procedimental)	- **Secuencias** de Operaciones
» No se indica CÓMO evaluar la consulta, sino **QUÉ** se desea obtener » **Describe** la información deseada sin dar un procedimiento específico para obtenerla	» Aunque se anidan para formar una sola expresión, **siempre** se indica explícitamente cierto **orden** de las operaciones » **Estrategia parcial de evaluación** de la consulta (≈ lenguaje procedimental de alto nivel)

Funciones de Agregado $\mathcal{F}$

$$R = {}_{\text{nacionalidad}}\mathcal{F}_{\text{avg(edad), count}}(P)$$

P	Nombre	Nacionalidad	Edad
	Pedro	Venezuela	45
	Gabriel	Venezuela	20
	Luis	Argentina	33
	Andrés	Colombia	20
	Miguel	Argentina	23
	Luis	Peru	34
	Gilberto	Colombia	15

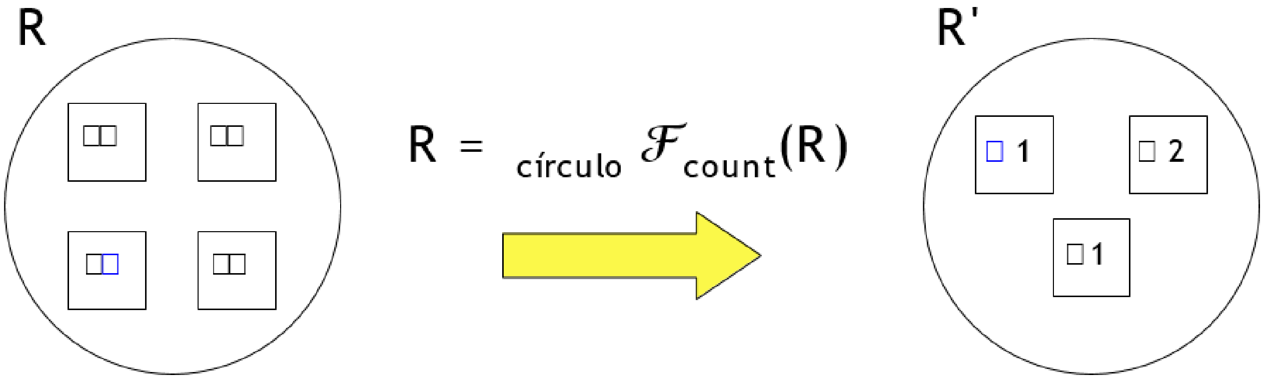
Posibles funciones de agregación: avg, sum, max, min, count, entre otras

R	Nacionalidad	AVG(Edad)	Count
	Venezuela	32,5	2
	Argentina	28	2
	Colombia	17,5	2
	Peru	34	1

Se agrupan los datos usando algún tipo de criterio y luego se calculan funciones sobre los datos agrupados

Funciones de Agregado $\mathcal{F}$

Se agrupan las tuplas en base al valor de cierto atributo y luego se pueden calcular funciones sobre atributos de las tuplas agrupadas



Ordenar

- **ASC**<atributos de orden separados por coma>(RELACIÓN_A_ORDENAR)

  DSC<atributos de orden separados por coma>(RELACIÓN_A_ORDENAR)

Ejercicio:

sucursal (nombre-sucursal, ciudad-sucursal, capital)

cliente (nombre-cliente, calle-cliente, ciudad-cliente)

cuenta (numero-cuenta, nombre-sucursal, saldo)

prestamo (numero-prestamo, nombre-sucursal, cantidad)

cliente-cuenta (nombre-cliente, número-cuenta)

cliente-prestamo (nombre-cliente, numero-prestamo)

Ejercicio:

- Encontrar todos los prestamos de más de 1200

$$\sigma_{\textit{cantidad} > 1200}\ (\textit{prestamo})$$

- Encontrar el numero-préstamo para todos los prestamos de una cantidad superior a 1200

$$\Pi_{\textit{numero-prestamo}}\ (\sigma_{\textit{cantidad} > 1200}\ (\textit{prestamo}))$$

Ejercicio:

- Cuáles son los nombres de los clientes que tiene un préstamo, una cuenta (o ambos)

$$\Pi_{\textit{nombre\_cliente}}\ (\textit{cliente-prestamo}) \cup \Pi_{\textit{nombre\_cliente}}\ (\textit{cliente-cuenta})$$

- Cuales son los nombres de los clientes que tienen una cuenta y un préstamo

$$\Pi_{\textit{nombre-cliente}}\ (\textit{cliente-prestamo}) \cap \Pi_{\textit{nombre-cliente}}\ (\textit{cliente-cuenta})$$

Ejercicio:

- Encontrar los nombres de todos los clientes que tienen un préstamo en la sucursal molina.

$$\Pi_{\textit{nombre-cliente}} (\sigma_{\textit{nombre-sucursal="molina"}} (\sigma_{\textit{c-prestamo.numero-prestamo= prestamo.numero-prestamo}} (\textit{cliente-prestamo x prestamo})))$$

- Nombres de los clientes que tienen un préstamo en la sucursal Molina pero no tienen una cuenta en dicha sucursal.

$$\Pi_{\textit{nombre-cliente}} (\sigma_{\textit{nombre-sucursal = "molina"}} (\sigma_{\textit{c-prestamo.numero-prestamo = prestamo.numero-prestamo}} (\text{cliente-prestamo x prestamo}))) - \Pi_{\textit{nombre-cliente}} (\sigma_{\textit{nombre-sucursal = "Molina"}} (\sigma_{\textit{c-cuenta.numero-cuenta = cuenta.numero-cuenta}} (\text{cliente-cuenta x cuenta})))$$

Ejercicio:

- Nombre de todos los clientes que tienen un préstamo en la sucursal Molina.

Solución 1

$\Pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal = “Molina”}}(\sigma_{\text{cliente-prestamo.numero-prestamo = prestamo.numero-prestamo}}(\text{cliente-prestamo x prestamo})))$

Solución 2

$\Pi_{\text{cliente-nombre}}(\sigma_{\text{prestamo.numero-prestamo = c-prestamo.numero-prestamo}}((\sigma_{\text{nombre-sucursal = “molina”}}(\text{prestamo})) \text{ x } \text{cliente-prestamo}))$