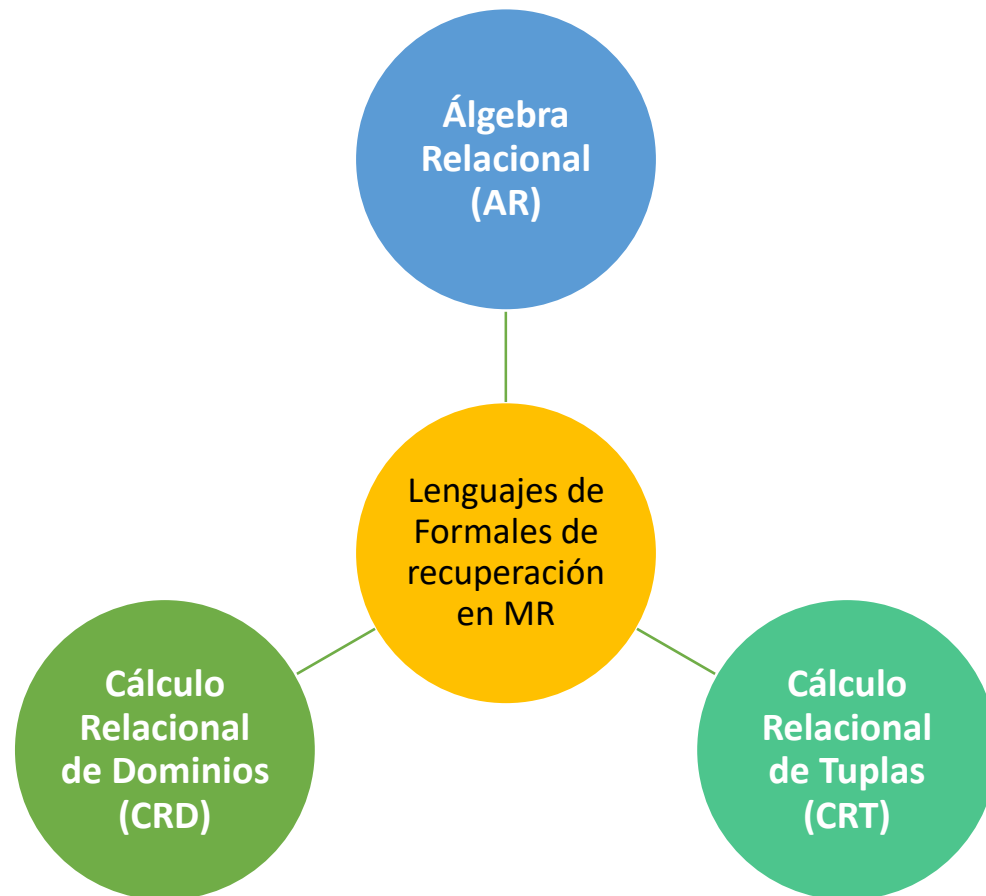


Lenguajes de Consulta

Autor: Sergio D'Arrigo

Lenguajes de consulta

- Todo modelo de base de datos debe incluir un conjunto de operaciones para manipular la base.
- Un conjunto de operaciones fundamentales son las de **recuperación de datos**.



Lenguajes de consulta

- **Álgebra** relacional es un lenguaje **procedural**
- **Cálculo** relacional es un lenguaje **declarativo**.
- Los 3 lenguajes formales tienen igual poder expresivo si se consideran las siguientes restricciones:
 - AR sin las operaciones extendidas de agregación y proyección generalizada (no las veremos en la materia)
 - CRT y CRD restringido a expresiones seguras (devuelven un número finito de resultados)
- A nivel práctico, el lenguaje más difundido es el **SQL**, basado tanto en el AR como en el CRT.

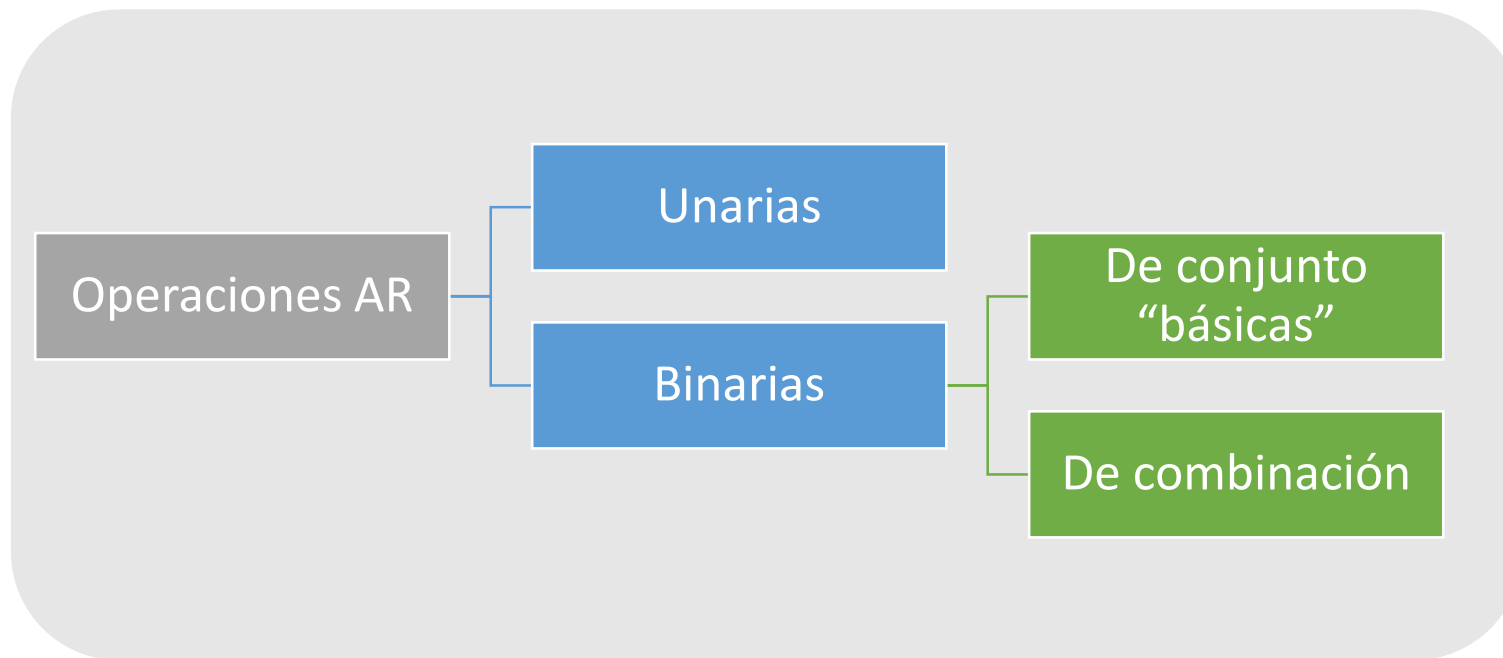
Álgebra Relacional

- Es un lenguaje formal de consultas para recuperar datos almacenados en un modelo relacional.
- Es un lenguaje **procedural**, se detallan las operaciones que deben realizarse para obtener los resultados deseados.
- Las consultas de AR:
 - ☐ Reciben como entrada una o dos instancias de relación
 - ☐ Devuelven una instancia de relación (las operaciones son cerradas)
- El resultado de una consulta AR es un conjunto de elementos, **no tiene tuplas repetidas**.
- Las operaciones de AR **pueden componerse**. El input de una operación puede ser el resultado de una operación.

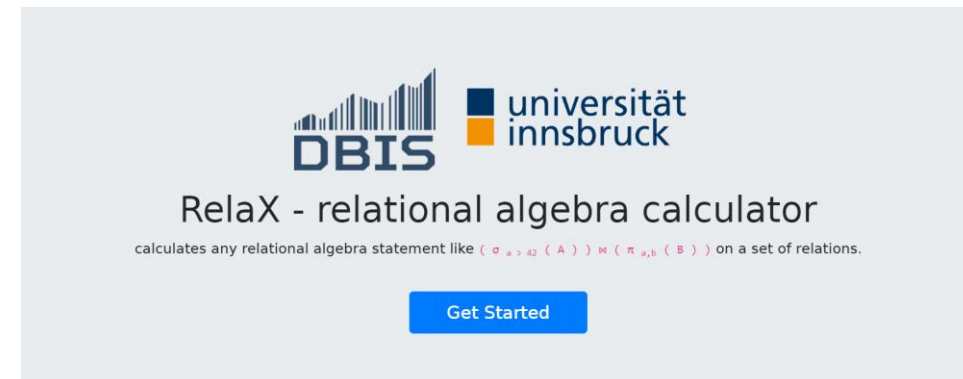
✍ AR es la base para la ejecución y optimización de consultas en los SGBD Relacionales.

Álgebra Relacional

- Los tipos de operaciones podrían clasificarse así:



El siguiente link corresponde a un calculador de álgebra relacional desarrollado por la Universidad de Innsbruck.
<https://dbis-uibk.github.io/relax/landing>



Álgebra Relacional

Operaciones Unarias


☐ Proyección

☐ Selección

☐ Renombre

Proyección (π): permite recuperar algunos atributos de una relación. Mantiene todas las tuplas origen.

La operación $\pi_{Id}(Materia)$ retorna

Id
1
2
3
4
5

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

La operación $\pi_{Id_Carrera}(Materia)$ retorna

Id_Carrera
1
2

Álgebra Relacional

Operaciones Unarias



☐ Proyección

☐ **Selección**

☐ Renombre

Selección (σ): permite recuperar algunas filas de una relación.
Mantiene todos los atributos origen.

La operación $\sigma_{\text{Nombre}='Materia 1'}(\text{Materia})$ retorna

Id	Nombre	Id_Carrera
1	Materia 1	1

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Álgebra Relacional

Operaciones Unarias



☐ Proyección

☐ Selección

☒ Renombre

Renombre (ρ): permite renombrar nombres de atributos o de relaciones.

La operación ρ ($id \rightarrow Id_Materia, Materia$) retorna

Id_Materia	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

La operación ρ ($MateriaFacultad, Materia$) retorna

MateriaFacultad

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Álgebra Relacional

Binarias de conjunto “Básicas”



- ☐ Unión
- ☐ Intersección
- ☐ Diferencia

*Sólo son posibles si las relaciones son “**Unión Compatibles**”.
Esto es, tienen la misma aridad, y los dominios de los i -ésimos atributos de cada relación son iguales para todo i*

Álgebra Relacional

Binarias de conjunto “Básicas”


☐ Unión

☐ Intersección

☐ Diferencia

Unión (U): Es la unión matemática de conjuntos.

El resultado incluye a las tuplas que están en la primera relación, en la segunda o en ambas, eliminando tuplas duplicadas.

La operación $\text{Materia_1C} \cup \text{Materia_2C}$ retorna

Materia_1C	
Id	Nombre
1	Materia 1
2	Materia 2
3	Materia 3

Materia_2C	
Id	Nombre
1	Materia 1
3	Materia 3
4	Materia 4
5	Materia 5

Id	Nombre
1	Materia 1
2	Materia 2
3	Materia 3
4	Materia 4
5	Materia 5

Álgebra Relacional

Binarias de conjunto “Básicas”



- ☐ Unión
- ☒ Intersección
- ☐ Diferencia

Intersección (\cap): Es la intersección matemática de conjuntos. El resultado incluye a las tuplas que están en ambas relaciones.

La operación $\text{Materia_1C} \cap \text{Materia_2C}$ retorna

Materia_1C	
Id	Nombre
1	Materia 1
2	Materia 2
3	Materia 3

Materia_2C	
Id	Nombre
1	Materia 1
3	Materia 3
4	Materia 4
5	Materia 5

Id	Nombre
1	Materia 1
3	Materia 3

Álgebra Relacional

Binarias de conjunto “Básicas”



- ☐ Unión
- ☐ Intersección
- ☒ **Diferencia**

Diferencia (-): Es la diferencia matemática de conjuntos. El resultado incluye a las tuplas que están en la primera relación y no están en la segunda.

La operación $\text{Materia_1C} - \text{Materia_2C}$ retorna

Materia_1C	
Id	Nombre
1	Materia 1
2	Materia 2
3	Materia 3

Materia_2C	
Id	Nombre
1	Materia 1
3	Materia 3
4	Materia 4
5	Materia 5

Id	Nombre
2	Materia 2

Álgebra Relacional

Binarias de combinación



☐ Producto Cartesiano

☐ Tetha Join

☐ Equijoin

☐ Natural Join

☐ Left Outer Join

☐ Right Outer Join

☐ Full Outer Join

☐ División

*Para estas operaciones **NO ES NECESARIO** que las relaciones sean “**Unión Compatibles**”.*

Álgebra Relacional

Binarias de combinación


☐ **Producto Cartesiano**
☐ Theta Join

☐ Equijoin

☐ Natural Join

☐ Left Outer Join

☐ Right Outer Join

☐ Full Outer Join

☐ División

Producto Cartesiano (\times): Devuelve una relación donde cada elemento es la combinación de cada tupla de la primera relación con cada tupla de la segunda. Notar que no exige ninguna relación entre las tuplas para combinarlas.

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

Título

Id	Nombre
1	Licenciatura
2	Profesorado

La operación $\text{Carrera} \times \text{Título}$ retorna

Id_Carrera	Carrera.Nombre	Id	Título.Nombre
1	Carrera 1	1	Licenciatura
1	Carrera 1	2	Profesorado
2	Carrera 2	1	Licenciatura
2	Carrera 2	2	Profesorado
3	Carrera 3	1	Licenciatura
3	Carrera 3	2	Profesorado

Álgebra Relacional

Binarias de combinación


☐ Producto Cartesiano

☐ **Theta Join**
☐ Equijoin

☐ Natural Join

☐ Left Outer Join

☐ Right Outer Join

☐ Full Outer Join

☐ División

Theta Join (\bowtie_{Cond}): Combina las tuplas de dos relaciones que cumplen una cierta condición lógica de junta.

La operación $Carrera \bowtie_{Id_Carrera > Id} Titulo$ retorna

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

Título

Id	Nombre
1	Licenciatura
2	Profesorado

Id_Carrera	Carrera.Nombre	Id	Titulo.Nombre
2	Carrera 2	1	Licenciatura
3	Carrera 3	1	Licenciatura
3	Carrera 3	2	Profesorado

Álgebra Relacional

Binarias de combinación


☐ Producto Cartesiano

☐ Tetha Join

☒ **Equijoin**
☐ Natural Join

☐ Left Join

☐ Right Join

☐ Full Join

☐ División

Equijoin($\bowtie_{Cond=}$): Combina las tuplas de dos relaciones que cumplen una cierta condición lógica de junta que sólo tiene comparaciones de igualdad.

La operación $Materia \bowtie_{Id_Carrera=Id_Carrera} Carrera$ retorna

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

Id	Materia.Nombre	Materia.Id_Carrera	Carrera.Id_Carrera	Carrera.Nombre
1	Materia 1	1	1	Carrera 1
2	Materia 2	1	1	Carrera 1
3	Materia 3	1	1	Carrera 1
4	Materia 4	2	2	Carrera 2
5	Materia 5	2	2	Carrera 2

Álgebra Relacional

Binarias de combinación


☐ Producto Cartesiano

☐ Tetha Join

☐ Equijoin

☒ **Natural Join**
☐ Left Join

☐ Right Join

☐ Full Join

☐ División

Natural Join (\bowtie): Combina las tuplas de dos relaciones que cumplen que todos los atributos de nombre idéntico tienen igual valor, removiendo los atributos innecesarios (repeticiones de atributos utilizados para la junta).

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

La operación $\text{Materia} \bowtie \text{Carrera}$ retorna

Id	Materia.Nombre	Id_Carrera	Carrera.Nombre
1	Materia 1	1	Carrera 1
2	Materia 2	1	Carrera 1
3	Materia 3	1	Carrera 1
4	Materia 4	2	Carrera 2
5	Materia 5	2	Carrera 2

Álgebra Relacional

Binarias de combinación


☐ Producto Cartesiano

☐ Tetha Join

☐ Equijoin

☐ Natural Join

☒ Left Outer Join

☐ Right Outer Join

☐ Full Outer Join

☐ División

Left Outer Join (\bowtie): Similar a los joins que vimos, con la diferencia que devuelve todas la tuplas de la primera relación, aún cuando no combinen con ninguna tupla de la segunda.

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2
6	Materia 6	3

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

La operación $\text{Materia} \bowtie \text{Carrera}$ retorna

Id	Materia.Nombre	Id_Carrera	Carrera.Nombre
1	Materia 1	1	Carrera 1
2	Materia 2	1	Carrera 1
3	Materia 3	1	Carrera 1
4	Materia 4	2	Carrera 2
5	Materia 5	2	Carrera 2
6	Materia 6	4	NULL

Álgebra Relacional

Binarias de combinación


☐ Producto Cartesiano

☐ Tetha Join

☐ Equijoin

☐ Natural Join

☐ Left Outer Join

☒ Right Outer Join

☐ Full Outer Join

☐ División

Right Outer Join (\bowtie): Similar a los joins que vimos, con la diferencia que devuelve todas la tuplas de la segunda relación, aún cuando no combinen con ninguna tupla de la primera.

La operación $\text{Materia} \bowtie_{\text{Id_Carrera}=\text{Id_Carrera}} \text{Carrera}$ retorna

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2
6	Materia 6	3

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

Id	Materia.Nombre	Materia.Id_Carrera	Carrera.Id_Carrera	Carrera.Nombre
1	Materia 1	1	1	Carrera 1
2	Materia 2	1	1	Carrera 1
3	Materia 3	1	1	Carrera 1
4	Materia 4	2	2	Carrera 2
5	Materia 5	2	2	Carrera 2
NULL	NULL	NULL	3	Carrera 3

Álgebra Relacional

Binarias de combinación



☐ Producto Cartesiano

☐ Tetha Join

☐ Equijoin

☐ Natural Join

☐ Left Outer Join

☐ Right JOuter oin

☐ Full Outer Join

☐ División

Full Outer Join (\bowtie): Similar a los joins que vimos, con la diferencia que devuelve todas la tuplas de la primera relación y de la segunda relación, aún cuando no combinen con ninguna tupla de la otra relación.

La operación $\text{Materia} \bowtie_{\text{Id_Carrera}=\text{Id_Carrera}} \text{Carrera}$ retorna

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2
6	Materia 6	3

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

Id	Materia.Nombre	Materia.Id_Carrera	Carrera.Id_Carrera	Carrera.Nombre
1	Materia 1	1	1	Carrera 1
2	Materia 2	1	1	Carrera 1
3	Materia 3	1	1	Carrera 1
4	Materia 4	2	2	Carrera 2
5	Materia 5	2	2	Carrera 2
6	Materia 6	4	NULL	NULL
NULL	NULL	NULL	3	Carrera 3

Álgebra Relacional

Binarias de combinación


☐ Producto Cartesiano

☐ Tetha Join

☐ Equijoin

☐ Natural Join

☐ Left Join

☐ Right Join

☐ Full Join

☒ División

División (\div): Dadas dos relaciones $R(Z)$ y $S(X)$, con $X \subseteq Z$, y $Y=Z-X$. La división proyecta los atributos Y de la relación R para aquellas tuplas que aparecen en tuplas asociadas a todas las combinaciones de atributos X de la relación Z .

Sirve para recuperar datos de consultas del estilo:

- Devolver los datos de los marineros que reservaron **todos** los botes.
- Devolver los nombres de los empleados que trabajan en **todos** los proyectos en los que trabaja Juan Pérez

Division

Sean R y S con cardinalidad r y s respectivamente

$$R/S = \{t/t(a_1, a_2, \dots, a_{r-s}) \text{ tq } \forall t' \in S, t' = (a'_1, a'_2, \dots, a'_s), \\ \exists t'' \in R, t'' = (a''_1, a''_2, \dots, a''_r) \text{ y } \forall_{i=1}^{r-s} a_i = a''_i \text{ y } \forall_{i=1}^s a'_i = a''_{r-s+i}, \}$$

Álgebra Relacional

Binarias de combinación


☐ Producto Cartesiano

☐ Tetha Join

☐ Equijoin

☐ Natural Join

☐ Left Join

☐ Right Join

☐ Full Join

☒ División

División (\div): Dadas dos relaciones $R(Z)$ y $S(X)$, con $X \subseteq Z$, y $Y=Z-X$. La división proyecta los atributos Y de la relación R para aquellas tuplas que aparecen en tuplas asociadas a todas las combinaciones de atributos X de la relación Z .

La operación $\text{Ofrece_Titulo} \div \pi_{\text{Id_Titulo}}(\text{Titulo})$ retorna

Titulo	
Id_Titulo	Nombre
1	Licenciatura
2	Profesorado

Ofrece_Titulo	
Id_Carrera	Id_Titulo
1	1
2	1
2	2
3	2

Id_Carrera
2

Álgebra Relacional

Composición de operaciones: Las operaciones de algebra relacional pueden componerse, para lograr expresiones más complejas.

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

La operación $\pi_{\text{NomMateria, NomCarrera}} (\rho_{(\text{Nombre} \rightarrow \text{NomMateria}, \text{Materia})} \bowtie \rho_{(\text{Nombre} \rightarrow \text{NomCarrera}, \text{Carrera})})$ retorna

NomMateria	NomCarrera
Materia 1	Carrera 1
Materia 2	Carrera 1
Materia 3	Carrera 1
Materia 4	Carrera 2
Materia 5	Carrera 2

Álgebra Relacional

Secuenciación de operaciones: Las operaciones de algebra relacional pueden secuenciarse, debiendo para esto dar un nombre a los resultados intermedios mediante el operador de asignación “←”

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

La secuencia de operaciones

$MC \leftarrow \text{Materia} \bowtie_{\text{Materia.Id_Carrera}=\text{Carrera.Id_Carrera}} \text{Carrera}$

$\pi_{\text{NomMateria, NomCarrera}} (\rho_{(\text{Materia.Nombre} \rightarrow \text{NomMateria, Carrera.Nombre} \rightarrow \text{NomCarrera, MC})}$

retorna

NomMateria	NomCarrera
Materia 1	Carrera 1
Materia 2	Carrera 1
Materia 3	Carrera 1
Materia 4	Carrera 2
Materia 5	Carrera 2
NULL	Carrera 3

Álgebra Relacional

Árbol de Consultas: Las expresiones de álgebra relacional pueden ser vistas también como un árbol. Esta representación es usada por los SGBD en el compilador o en el optimizador de consultas

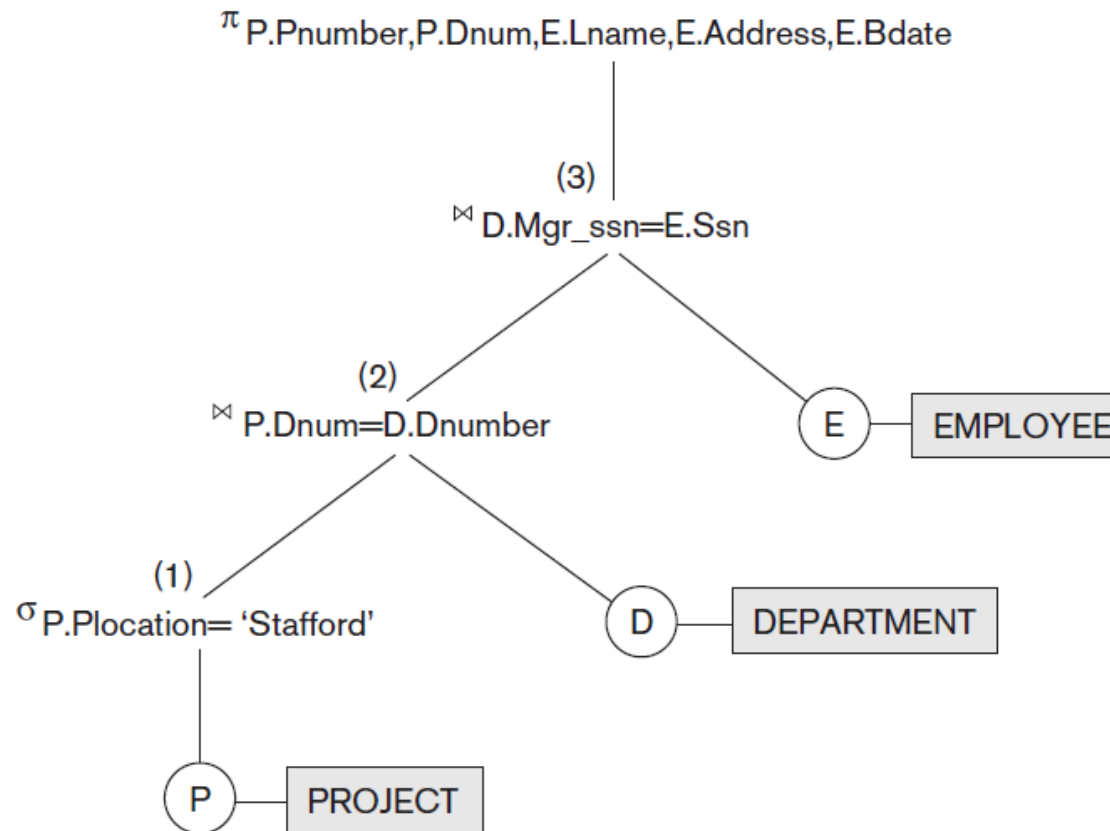


Figura de: Elmasri, Navathe (2016)
Fundamentals of Database Systems

AR - Ejercicio

- Dado el siguiente esquema de relación

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;

Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;

Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;

Guest contains guest details and guestNo is the primary key.

Realizar las consultas en AR que retornen

- a) El nombre y ciudad de todos los hoteles que tienen alguna habitación tipo “Suite”.
- b) Los nombres de los huéspedes que están actualmente en algún hotel denominado “Blue Star”.
- c) Los nombres de los huéspedes que se reservaron alguna vez en Madrid y en Roma.
- d) Los nombres de los huéspedes que se reservaron alguna vez en Madrid y nunca en Roma.
- e) Los nombres de los huéspedes con reservas realizadas en todas las ciudades en que hay hoteles.

(*) Asumimos que no hay huéspedes con nombres repetidos

Pensémoslos entre todos...

AR - Ejercicio – Una posible solución

- Dado el siguiente esquema de relación

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;

Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;

Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;

Guest contains guest details and guestNo is the primary key.

- a) El nombre y ciudad de todos los hoteles que tienen alguna habitación tipo “Suite”.

$$\pi_{\text{hotelName, city}} (\sigma_{\text{type='Suite'}} (\text{Hotel} \bowtie \text{Room}))$$

AR - Ejercicio – Una posible solución

- Dado el siguiente esquema de relación

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;

Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;

Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;

Guest contains guest details and guestNo is the primary key.

b) Los nombres de los huéspedes que están actualmente en algún hotel denominado “Blue Star”.

$$GAct \leftarrow \pi_{\text{guestNo}} (\sigma_{\text{hotelName}='Blue Star'} (\text{Hotel}) \bowtie \sigma_{\text{dateFrom} \leq \text{Now}() \wedge \text{dateTo} \geq \text{Now}()} (\text{Booking}))$$

$$\pi_{\text{guestName}} (\text{Guest} \bowtie GAct)$$

AR - Ejercicio – Una posible solución

- Dado el siguiente esquema de relación

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;

Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;

Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;

Guest contains guest details and guestNo is the primary key.

c) Los nombres de los huéspedes que se reservaron alguna vez en Madrid y en Roma.

$$G_{\text{Madrid}} \leftarrow \pi_{\text{guestName}} ((\sigma_{\text{city}='Madrid'} (\text{Hotel}) \bowtie (\text{Booking})) \bowtie (\text{Guest})))$$

$$G_{\text{Roma}} \leftarrow \pi_{\text{guestName}} ((\sigma_{\text{city}='Roma'} (\text{Hotel}) \bowtie (\text{Booking})) \bowtie (\text{Guest})))$$

$$G_{\text{Madrid}} \cap G_{\text{Roma}}$$

AR - Ejercicio – Una posible solución

- Dado el siguiente esquema de relación

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;

Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;

Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;

Guest contains guest details and guestNo is the primary key.

d) Los nombres de los huéspedes que se reservaron alguna vez en Madrid y nunca en Roma.

$$G_{\text{Madrid}} \leftarrow \pi_{\text{guestName}} ((\sigma_{\text{city}='Madrid'} (\text{Hotel}) \bowtie (\text{Booking})) \bowtie (\text{Guest})))$$

$$G_{\text{Roma}} \leftarrow \pi_{\text{guestName}} ((\sigma_{\text{city}='Roma'} (\text{Hotel}) \bowtie (\text{Booking})) \bowtie (\text{Guest})))$$

$G_{\text{Madrid}} - G_{\text{Roma}}$

AR - Ejercicio – Una posible solución

- Dado el siguiente esquema de relación

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;

Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;

Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;

Guest contains guest details and guestNo is the primary key.

e) Los nombres de los huéspedes con reservas realizadas en todas las ciudades en que hay hoteles.

$$\text{ResGC} \leftarrow \pi_{\text{guestNo, city}} (\text{Hotel} \bowtie \text{Booking})$$

$$\pi_{\text{guestName}} (\text{ResGC} \div \pi_{\text{city}} (\text{Hotel})) \bowtie \text{Guest})$$

Cálculo Relacional de Tuplas

- Es un lenguaje formal de consultas para **recuperar** datos almacenados en un modelo relacional.
- Es un lenguaje **declarativo**
- Una consulta de CRT es de la forma $\{ t \mid \text{COND}(t) \}$
 - Donde **t** es una variable tupla que evalúa en Verdadero o Falso sobre la condición **COND**
 - La consulta devuelve el **conjunto de tuplas que satisfacen COND(t)**, es decir, que evalúan en Verdadero
- En una expresión de CRT hay que especificar:
 - Una **relación rango** para cada variable tupla (a qué relación pertenece la tupla). Si no se establece, el rango son todas las posibles tuplas “ en el universo”.
 - Una **condición** (o fórmula) **de selección** sobre las tuplas
 - Un **conjunto de atributos** a ser recuperados

Cálculo Relacional de Tuplas

- Una expresión general es de la forma

$$\{t_1.A_j, t_2.A_k, \dots, t_n.A_m \mid \text{COND}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$$

- Fórmulas atómicas

- $R(t_i)$ *(especifica el rango de t_i)*
- $t_i.A \text{ op } t_j.B$ *(donde op es un operador de comparación: $<$, $>$, $=$, $>=$, $<=$, $<>$)*
- $t_i.A \text{ op } c \text{ o } c \text{ op } t_j.B$ *(donde c es una constante y op es un operador de comparación)*

- Fórmula

- Toda fórmula **atómica** es una fórmula
- Cualquier fórmula conectada por los operadores **NOT**, **AND** y **OR** es también una fórmula
- Si F es una fórmula, entonces $\exists (t) (F)$ también es una fórmula
- Si F es una fórmula, entonces $\forall (t) (F)$ también es una fórmula

Cálculo Relacional de Tuplas

- Ejemplos
- Recuperar tuplas completas de una relación, con alguna condición
- Materias de la carrera con id 1
- $\{ t \mid t \in \text{Materia y } t.\text{id_carrera}=1 \}$
- Recuperar algunos atributos de tuplas, con alguna condición
- Nombre de las materias de la carrera con id 1
- $\{ r \mid \exists t \in \text{Materia y } t.\text{id_carrera}=1 \text{ y } r.\text{Nombre}=t.\text{nombre} \}$
- Recuperar atributos de varias relaciones vinculadas
- Nombre de las materias que pertenecen a una carrera y de las carreras a las que pertenecen
- $\{ r \mid \exists t \in \text{Materia y } \exists s \in \text{Carrera y } t.\text{id_carrera}=s.\text{id_carrera} \text{ y } r.\text{nombre_materia}=t.\text{nombre y } r.\text{nombre_carrera}=s.\text{nombre} \}$

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

Cálculo Relacional de Tuplas

- Ejemplos
- Nombre de las Carreras que tienen alguna materia asociada
- $\{ r \mid \exists t \in \text{Materia y not } \exists s \in \text{Carrera y } t.\text{id_carrera}=s.\text{id_carrera} \text{ y } r.\text{nombre_materia}=t.\text{nombre} \}$

Materia

Id	Nombre	Id_Carrera
1	Materia 1	1
2	Materia 2	1
3	Materia 3	1
4	Materia 4	2
5	Materia 5	2

Carrera

Id_Carrera	Nombre
1	Carrera 1
2	Carrera 2
3	Carrera 3

Cálculo Relacional de Tuplas

• Ejemplos

ALUMNO

Legajo	NomyApe	CodCarrera
1	Juan	A
2	Pedro	B
3	Maria	A
4	Jose	B

MATERIA

CodMateria	DMateria	CodCarrera
M1	Ingeniería del Software I	A
M2	Base de Datos	A
M3	Ingeniería del Software II	A
M4	Biología I	B
M5	Biología II	B

APROBADA

Legajo	CodMateria
1	M1
1	M2
1	M3
2	M4

- Nombre de las alumnos que tienen todas las materias aprobadas
- $\{ r \mid \exists t \in \text{Alumno} \text{ y } (\forall s \in \text{Materia}, \exists u \in \text{Aprobada} / u.\text{legajo}=t.\text{legajo} \text{ y } u.\text{codMateria}=s.\text{codMateria}) \text{ y } r.\text{nombre}=t.\text{nombre} \}$

Bibliografía del tema

- *Elmasri, Navathe (2016) Fundamentals of Database Systems, 7th Edition. Pearson. Cap. 8*

Dudas



Muchas gracias!