

Bases de Datos I



Álgebra Relacional y de Bolsas



ÁLGEBRA RELACIONAL

Lenguajes de consulta en BD Relacionales:

- **Álgebra Relacional: consulta procedimental**
 - Conjunto de operaciones que permiten describir cómo obtener una respuesta sobre relaciones definidas en el modelo relacional
 - Se utilizan como representación para traducir y optimizar consultas SQL
- **Cálculo Relacional: consulta no procedimental**
 - Describe la información deseada sin dar un procedimiento específico para obtenerla.
 - Poder expresivo similar al álgebra relacional, pero declarativo.
- **Lenguajes de Usuario:** SQL (Structured Query Language), basado en **álgebra relacional** pero su interpretación es mediante **álgebra de bolsas**)

OPERADORES

El Algebra Relacional define un conjunto de **operadores estándar** para consultas en BD relacionales. Son **símbolos** que denotan procedimientos para construir nuevas relaciones a partir de otras relaciones (propiedad de clausura)

- **Operadores tradicionales sobre conjuntos de tuplas:**

Unión , Diferencia , Producto Cartesiano, Intersección

- **Operadores específicos para BD relacionales:**

Selección, Proyección, Ensamble o reunión (Join), División

Las relaciones no admiten tuplas repetidas, **NO** hay operación del álgebra relacional que genere **tuplas repetidas**

TABLAS EN SQL

- SQL no trata a las relaciones como conjuntos sino que, en general, permite tuplas repetidas en los resultados de las consultas (*a menos que se indique explícitamente que deben ser removidos*)
 - La eliminación de duplicados es una operación costosa (requiere ordenar y luego eliminar repeticiones)
 - Es posible que se requiera conocer todos los resultados de una consulta, incluso si están duplicados
- **Los motores de BD relacionales no trabajan con relaciones, trabajan con bolsas**
- Las operaciones del álgebra relacional se redefinen para el **álgebra de bolsas**

OPERADORES ALGEBRAICOS

Cada operador del algebraico **acepta una o dos relaciones** (o expresiones relacionales) y retorna una relación como resultado

Operadores Unarios: (interviene una sola relación)

Selección y Proyección (σ π)

Operadores Binarios: (intervienen dos relaciones)

Unión, Diferencia, Producto Cartesiano

\cup $-$ \times

Intersección, Ensamble, División

\cap \bowtie \div

OPERADORES ALGEBRAICOS

Algunas operaciones son **fundamentales** y otras pueden ser expresadas como una combinación o **derivación** de las anteriores

- **Operadores básicos o fundamentales:**

Unión, Diferencia, Producto Cartesiano, Selección, Proyección

- **Operadores derivados:** se pueden expresar en función de operadores básicos; expresan operaciones importantes y se usan habitualmente

Intersección, Ensamble, División

SECUENCIA DE OPERACIONES y RENOMBRADO

Para resolver algunas consultas sobre la BD frecuentemente se necesita aplicar varios operadores relacionales. Es necesario utilizar **expresiones relacionales**. **Opciones:**

Anidar expresiones

- Las operaciones unarias tienen prioridad sobre las binarias
- Las expresiones se evalúan de izquierda a derecha
- Para alterar el orden implícito de las expresiones se usan paréntesis

Aplicar los operadores individualmente y generar resultados intermedios

Se pueden re-nombrar las relaciones intermedias: $T \leftarrow \text{Expresión algebr.}$

También: **Renombrado de atributos:** $\rho_{Atrtb1 \rightarrow Alias1, \dots, Atrtbn \rightarrow Aliasn}$
(R)

ALGEBRA RELACIONAL: SELECCIÓN

Permite obtener las tuplas de una relación R que cumplen una condición

$$\sigma_{\text{condición}}(R)$$

condición es una expresión booleana que puede utilizar operadores de comparación ($<$, $>$, $=$, \leq , \geq , \neq), en la que aparece al menos un atributo de R

Puede ser una combinación booleana de varias de estas comparaciones mediante OR o AND (\vee ó \wedge) o una negación (\neg)

Ejemplo

ALUMNO

Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura

$\sigma_{\text{id_alum} > 200}(\text{ALUMNO})$

Id_alum	Apellido	Nombre
250	García	Juan
438	Alonso	Laura

AR - PROYECCIÓN

Permite obtener las tuplas con un sub-conjunto de atributos de R

$$\pi_{\text{lista_atributos}}(R)$$

lista_atributos es una lista de atributos de R que aparecen en el resultado

Se **eliminan tuplas repetidas** en el resultado (podrían quedar menos tuplas que en R)

Ejemplo

ALUMNO
(ALUMNO)

Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura

π

Apellido

Apellido

Alonso

García

TABLAS EN SQL

SQL no trata a las relaciones como conjuntos sino que, en general, permite tuplas repetidas en los resultados de las consultas (*a menos que se indique explícitamente que deben ser removidos*)

- La eliminación de duplicados es una **operación costosa** (requiere ordenar y luego eliminar repeticiones)
- Es posible que se requiera conocer todos los resultados de una consulta, incluso si están duplicados

Los motores de BD relacionales no trabajan con relaciones, trabajan con BOLSAS

Las operaciones del álgebra relacional se redefinen para el **álgebra de bolsas**

ÁLGEBRA DE BOLSAS - OPERACIONES

Selección: $\sigma_{\text{cond}}(\mathbf{B})$: filtra las tuplas **preservando duplicados**

Proyección: $\pi_A(\mathbf{B})$: proyecta B sobre las columnas del conjunto de atributos A, **preservando los duplicados**

Eliminación de duplicados: $\varepsilon(\mathbf{B})$ → indicación explícita para eliminar los duplicados en B

Ejemplo

(ALUMNO)

Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura

π
Apellido

Apellido
Alonso
García
Alonso

ALGEBRA RELACIONAL: UNIÓN

- Genera la unión de dos relaciones tomadas como conjuntos de tuplas

$$R_1 \cup R_2$$

- Las tuplas en el resultado son las que se encuentran en R_1 , o en R_2 , o en ambas relaciones a la vez (pero sin repeticiones)
- Para poder realizar esta operación: R_1 y R_2 deben ser **compatibles para la unión** (poseer el mismo esquema y dominio de definición: cada atributo de R_1 debe tener igual nombre y dominio que el correspondiente atributo de R_2)

Ejemplo

Listar los alumnos y los ayudantes

ALUMNO

Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura

AYUDANTE

Id_alum	Apellido	Nombre
108	Pérez	José
250	García	Juan

ALUMNO U AYUDANTE

Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura
108	Pérez	José

ALGEBRA RELACIONAL: DIFERENCIA

- Obtiene la diferencia de dos relaciones tomadas como conjuntos de tuplas

$$R_1 - R_2$$

- Las tuplas en el resultado son las que se están en R_1 y que no están en R_2
- el esquema resultante es igual al de R_1 (y de R_2)
- Para poder realizar esta operación: R_1 y R_2 deben ser **compatibles para la unión**

Ejemplo

Listar los alumnos que no son ayudantes

ALUMNO		
Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura

AYUDANTE		
Id_alum	Apellido	Nombre
108	Pérez	José
250	García	Juan

ALUMNO – AYUDANTE		
Id_alum	Apellido	Nombre
158	Alonso	Martin
438	Alonso	Laura

ALGEBRA RELACIONAL: INTERSECCIÓN

- Obtiene la intersección de 2 relaciones tomadas como conjuntos de tuplas

$$R_1 \cap R_2$$

- Las tuplas en el resultado son las que se encuentran en R_1 y también en R_2
- Para poder realizar esta operación: R_1 y R_2 deben ser **compatibles para la unión**
- Se puede expresar como: $R_1 \cap R_2 = R_1 - (R_1 - R_2)$

Ejemplo

Listar los alumnos que son ayudantes

ALUMNO		
Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura

AYUDANTE		
Id_alum	Apellido	Nombre
108	Pérez	José
250	García	Juan

ALUMNO \cap AYUDANTE		
Id_alum	Apellido	Nombre
250	García	Juan

ALGEBRA RELACIONAL: PRODUCTO CARTESIANO

- Obtiene una relación cuyas tuplas están formadas por la concatenación de todas las tuplas de R_1 con todas las tuplas de R_2

$$R_1 \times R_2$$

→ Las tuplas en el resultado son todas las combinaciones posibles de las tuplas de R_1 y de R_2 *(no suele usarse como operación significativa por sí sola)*

Ejemplo

ALUMNO		
Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura

MATERIA	
Id_m	Nom_mat
C1	Materia 1
C2	Materia 2

ALUMNO x MATERIA

Id_alum	Apellido	Nombre	Id_m	Nom_mat
158	Alonso	Martin	C1	Materia 1
250	García	Juan	C1	Materia 1
438	Alonso	Laura	C1	Materia 1
158	Alonso	Martin	C2	Materia 2
250	García	Juan	C2	Materia 2
438	Alonso	Laura	C2	Materia 2

ÁLGEBRA DE BOLSAS: UNIÓN

- **Unión máxima:** $B_1 \cup_{\max} B_2 \rightarrow$ devuelve las tuplas que aparecen en B_1 o en B_2 , incluyendo tantas ocurrencias como la cantidad máxima de repeticiones entre las bolsas

$$\text{contar}(x, B_1 \cup_{\max} B_2) = \max(\text{contar}(x, B_1), \text{contar}(x, B_2))$$

- **Unión aditiva:** $B_1 \uplus B_2 \rightarrow$ devuelve las tuplas que aparecen en B_1 o en B_2 , incluyendo tantas ocurrencias como la suma de apariciones en ambas bolsas

$$\text{contar}(x, B_1 \uplus B_2) = \text{contar}(x, B_1) + \text{contar}(x, B_2)$$

ÁLGEBRA DE BOLSAS: OTRAS OPERACIONES

- **Diferencia (minus):** $B_1 \dot{-} B_2 \rightarrow$ un elemento duplicado aparece tantas veces como la diferencia de apariciones de B_1 menos las de B_2 , pero no menos de cero

$$\text{contar}(x, B_1 \dot{-} B_2) = \max((\text{contar}(x, B_1) - \text{contar}(x, B_2)), 0)$$

- **Intersección mínima:** $B_1 \cap_{\min} B_2 \rightarrow$ un elemento duplicado aparece tantas veces como el mínimo de apariciones entre las bolsas B_1 y B_2

$$\text{contar}(x, B_1 \cap_{\min} B_2) = \min(\text{contar}(x, B_1), \text{contar}(x, B_2))$$

- **Producto cartesiano:** $B_1 \times B_2 \rightarrow$ obtiene todos los posibles pares de tuplas entre B_1 y B_2 , preservando las repeticiones

$$\text{contar}((x, s), B_1 \times B_2) = \text{contar}(x, B_1) * \text{contar}(s, B_2)$$

ALGEBRA: EJEMPLOS

Ejemplos

Considerando la *BD de Voluntarios* (TP2):

- Listar apellido y e-mail de los voluntarios que llevan aportadas más de 1000 horas

- Listar los distintos códigos de tareas que realizan los voluntarios

- Listar nombre y teléfono de todos los voluntarios de las instituciones 20 y 50

VOLUNTARIO		
nro_voluntario	int	PK
nombre	varchar(20)	N
apellido	varchar(25)	
e_mail	varchar(25)	
telefono	varchar(20)	N
fecha_nacimiento	date	
id_tarea	varchar(10)	FK
horas_aptadas	decimal(8,2)	N
porcentaje	decimal(2,2)	N
id_institucion	int	N FK
id_coordinador	int	N FK

ALGEBRA: EJEMPLOS

Ejemplos

Considerando la *BD de Voluntarios* (TP2):

- Listar apellido y e-mail de los voluntarios que llevan aportadas más de 1000 horas

$\pi_{\text{nombre, e_mail}} (\sigma_{\text{horas_aportadas} > 1000} (\text{VOLUNTARIO}))$

- Listar los distintos códigos de tareas que realizan los voluntarios

$\pi_{\text{id_tarea}} (\text{VOLUNTARIO})$

- Listar nombre y teléfono de todos los voluntarios de las instituciones 20 y 50

$\pi_{\text{nombre, telefono}} (\sigma_{\text{id_institucion} = 20 \text{ OR } \text{id_institucion} = 50} (\text{VOLUNTARIO}))$

o también:

- $\pi_{\text{nombre, telefono}} (\sigma_{\text{id_institucion} = 20} (\text{VOLUNTARIO})) \cup$
- $\pi_{\text{nombre, telefono}} (\sigma_{\text{id_institucion} = 50} (\text{VOLUNTARIO}))$

VOLUNTARIO		
nro_voluntario	int	PK
nombre	varchar(20)	N
apellido	varchar(25)	
e_mail	varchar(25)	
telefono	varchar(20)	N
fecha_nacimiento	date	
id_tarea	varchar(10)	FK
horas_aportadas	decimal(8,2)	N
porcentaje	decimal(2,2)	N
id_institucion	int	NFK
id_coordinador	int	NFK

PostgreSQL y ALGEBRA DE BOLSAS

Ejemplos

Considerando las siguientes tablas y sus tuplas:

```
CREATE TABLE R ( a int , b int );
```

```
CREATE TABLE S ( a int, b int );
```

```
INSERT INTO R (a, b) VALUES
```

```
(1,2), (1,2),
```

```
(1,2), (3,4);
```

```
INSERT INTO S (a, b) VALUES
```

```
(1,2), (3,4),
```

```
(3,4), (5,6);
```

Si una tupla t aparece n veces en R y m veces en S , entonces aparecerá:

- $n + m$ veces en $R \cup S$
- $\min(n, m)$ veces en $R \cap S$
- $\max(0, n-m)$ veces en $R - S$

PostgreSQL y ALGEBRA DE BOLSAS

Ejemplos

Cuál es el resultado de las siguientes consultas:

```
SELECT * FROM R  
UNION [ALL]  
SELECT * FROM S  
ORDER BY a;
```

```
SELECT * FROM R  
INTERSECT [ALL ]  
SELECT * FROM S  
ORDER BY a;
```

```
SELECT * FROM R  
EXCEPT [ALL ]  
SELECT * FROM S  
ORDER BY a;
```

ALGEBRA RELACIONAL: JOIN

Join General

$$R_1 \bowtie_{\text{cond}} R_2$$

Equijoin

$$R_1 \bowtie_{R_1.A=R_2.B} R_2 = \sigma_{R_1.A=R_2.B} (R_1 \times R_2)$$

Ensamble Natural

$$R_1 \bowtie R_2 = \pi_{\text{esq}(R_1) \cup \text{esq}(R_2) - (\text{esq}(R_1) \cap \text{esq}(R_2))} (\sigma_{R_1.A=R_2.A} (R_1 \times R_2))$$

→La selección chequea la igualdad de los atributos comunes y la proyección elimina los duplicados

LEFT OUTER JOIN - $R_1 \ltimes R_2$

RIGHT OUTER JOIN - $R_1 \rimes R_2$

FULL OUTER JOIN - $R_1 \Join R_2$

ALGEBRA RELACIONAL: DIVISIÓN

Si R_1 es una relación con esquema $(A_1, \dots, A_n, \mathbf{B_1}, \dots, \mathbf{B_m})$ y R_2 es una relación con esquema $(\mathbf{B_1}, \dots, \mathbf{B_m})$, la operación

$$\mathbf{R_1 \div R_2}$$

da como resultado otra relación con esquema (A_1, \dots, A_n) que contiene las correspondientes ocurrencias de las tuplas de R_1 que están acompañadas por **todas** las tuplas de R_2

NOTA: no existe un operador explícito para esta operación en el standard SQL, sino que hay que "simularlo" (*en algunos sistemas relacionales aparece el operador contains que es similar a la operación división del álgebra relacional*)

Ejemplo

Listar los identificadores de los alumnos inscritos en todas las materias

INSCRIPCIONES	
Id_alum	Id_m
158	C1
158	C2
250	C2

MATERIAS	
Id_m	Nom_mat
C1	Materia 1
C2	Materia 2

$\mathbf{INSCRIPCIONES \div \pi_{id_m}(MATERIAS)}$	
Id_alum	
158	

AR – FUNCIONES DE AGREGACIÓN y AGRUPAMIENTO

- Algunas consultas no se pueden responder con las operaciones anteriores
- **Funciones de agregación:** amplían el poder expresivo del AR
Toman un conjunto de valores y devuelven un valor único:

avg: valor promedio

min: valor mínimo

max: valor máximo

sum: suma de valores

count: número de valores

- Extensión del álgebra relacional para incorporar operadores de **agrupamiento de tuplas:**

$$\delta_{ATR, AGR} (R)$$

ATR → atributo o conjunto de atributos de agrupamiento

AGR → función/es de agrupamiento

OPERACIONES del ÁLGEBRA y SQL

- **Eliminación de duplicados:** se usa **DISTINCT**
 - **Operaciones del álgebra relacional:** algunas operaciones de conjuntos se incorporaron directamente en SQL
 - Unión de conjuntos: R_1 **UNION** R_2
 - Intersección de conjuntos: R_1 **INTERSECT** R_2
 - Diferencia entre conjuntos: R_1 **EXCEPT** R_2
- R_1 y R_2 son resultado de expresiones SELECT
- Las relaciones resultantes son conjuntos de tuplas → las repetidas son eliminadas del resultado, a menos que se anexe **ALL** a la operación
- Estas operaciones se aplican sobre relaciones **compatibles con la unión** (*puede ser necesario renombrar atributos*)

OPERACIONES del ÁLGEBRA y SQL

Operaciones en conjuntos

ÁLG. RELACIONAL	SQL
\cup	UNION
\cap	INTERSECT
$-$	EXCEPT

Operaciones en bolsas

ÁLG. BOLSAS	SQL
\uplus	UNION ALL
\cap_{min}	INTERSECT ALL
\div	EXCEPT ALL
$\varepsilon(B)$	DISTINCT
$\pi_{A1, \dots, An}(B)$	<i>SELECT A1,..., An FROM B</i>

ALGEBRA RELACIONAL: SELECCIÓN

Algunas **propiedades** de la selección:

- **Conmutativa:**

$$\sigma_{C_2} (\sigma_{C_1} (R)) = \sigma_{C_1} (\sigma_{C_2} (R))$$

- **Idempotente:**

$$\sigma_C (\sigma_C (R)) = \sigma_C (R)$$

- Se pueden **combinar** múltiples selecciones en una única selección:

$$\sigma_{C_1} (\sigma_{C_2} (\dots \sigma_{C_n} (R) \dots)) = \sigma_{C_1 \text{ AND } C_2 \text{ AND } \dots C_n} (R)$$

ALGEBRA RELACIONAL: OTRAS PROPIEDADES

- La Unión y la Intersección son operaciones conmutativas:

$$R_1 \cup R_2 = R_2 \cup R_1 \quad \text{y} \quad R_1 \cap R_2 = R_2 \cap R_1$$

- también son asociativas:

$$R_1 \cup (R_2 \cup R_3) = (R_1 \cup R_2) \cup R_3$$
$$\text{y } R_1 \cap (R_2 \cap R_3) = (R_1 \cap R_2) \cap R_3$$

- la Diferencia no es conmutativa, en general:

$$R_1 - R_2 \neq R_2 - R_1$$

OPERACIONES del ÁLGEBRA y SQL: DIVISIÓN

Ejemplo

Listar los identificadores de los alumnos inscriptos en todas las materias

ALUMNO		
Id_alum	Apellido	Nombre
158	Alonso	Martin
250	García	Juan
438	Alonso	Laura



```
SELECT A.Id_alum
FROM alumno A
WHERE NOT EXISTS
    ((SELECT M.Id_m
      FROM materia M)
  EXCEPT
  (SELECT I.Id_m
   FROM inscripcion I
   WHERE I.Id_alum=A.Id_alum));
```

INSCRIPCION	
Id_alum	Id_m
158	C1
158	C2
250	C2

```
SELECT A.Id_alum
FROM alumno A
WHERE NOT EXISTS
    ( SELECT M.Id_m
      FROM materia M
    WHERE NOT EXISTS
        ( SELECT I.Id_m
          FROM inscripcion I
          WHERE I.Id_alum=A.Id_alum
            AND I.id_m=M.Id_m ) );
```

→ Otra opción utilizando COUNT (en este caso chequeando que la cantidad de materias en las que se inscribió cada alumno (Inscripciones) coincida con la cantidad de materias (en Materias))

BIBLIOGRAFÍA

Date, C., "An Introduction to Database Systems". 7º ed., Addison Wesley, 2000

Elmasri, R., Navathe, S., "Fundamentals of Database Systems", Addison Wesley, 2011

Ramakrishnan R., Gehrke J., "Database Management Systems", 3º ed. , McGraw-Hill, 2003

Silberschatz, A., Korth, H, Sudarshan, S., "Database System Concepts", McGraw Hill, 2001