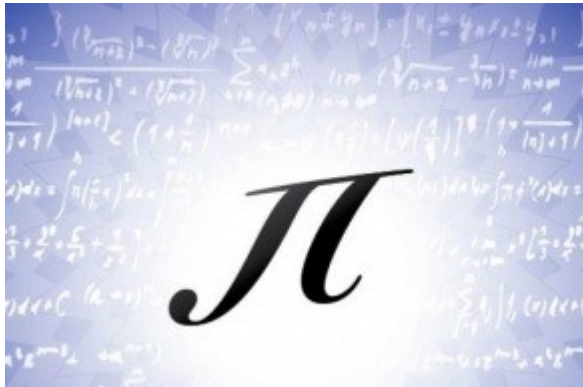


REPASO ALGEBRA RELACIONAL



Ejemplo de proyección

Tabla de insumos

nombre	precio
arena	20
cemento	80
cal	50
hierro	120

π_{nombre} insumos

nombre
arena
cemento
cal
hierro

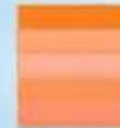
SELECCIÓN

FORMATO

σ

(condición)

(R)



Unión

R

Nombre	Dirección	Fecha
Juan Díaz	C/Pez,10	9-9-62
Ana Gómez	C/Luna,3	8-3-58

S

Nombre	Dirección	Fecha
Juan Díaz	C/Pez,10	9-9-62
Luis Díaz	C/ Sol, 5	8-5-67

RUS

Nombre	Dirección	Fecha
Juan Díaz	C/Pez,10	9-9-62
Ana Gómez	C/Luna,3	8-3-58
Luis Díaz	C/ Sol, 5	8-5-67

Diferencia (-)

- Se utiliza la diferencia entre la tabla **R** y la tabla **S** para determinar las personas que no están en las dos tablas.

R-S	Nombre	Dirección	Fecha
	Ana Gómez	C/Luna,3	8-3-58

Intersección (n)

La operación de intersección permite identificar filas que son comunes en dos relaciones. Al igual que la operación Union, para poder realizar esta operación es necesario e imprescindible que las tablas a unir tengan las mismas estructuras, que sus campos sean iguales.

- Equivalencia

- $R \cap S \equiv (R - (R - S))$

$R \cap S$

Juan Díaz	C/Pez,10	9-9-62
-----------	----------	--------

Producto (x)

- La operación producto consiste en la realización de un producto cartesiano entre dos tablas dando como resultado todas las posibles combinaciones entre los registros de la primera y los registros de la segunda.
- Sean R y S dos relaciones de grado m y n, respectivamente. El producto cartesiano, $R \times S$, es una relación de grado $m + n$ formada por todas las posibles tuplas en las que los m primeros elementos constituyen una tupla de R y los n últimos una tupla de S:

R	A	B
	1	2
	3	4

S	B	C	D
	2	5	6
	4	7	8
	9	10	11

RXS	A	R.B	S.B	C	D
	1	2	2	5	6
	1	2	4	7	8
	1	2	9	10	11
	3	4	2	5	6
	3	4	4	7	8
	3	4	9	10	11

Selección (σ)

- Por medio de esta operación se posibilita la selección de un subconjunto de tuplas de una relación que corresponden a una **condición: (columnaOPERADORvalor)** determinada.
- El grado (total de columnas de la Relación), se conserva

- Formato de Uso: $\sigma_{\text{condición}}(\text{RELACION})$
 - $\sigma_F(R)$
- Esta operación es la que normalmente se conoce como consulta.
- En este tipo de consulta se emplean los diferentes operadores de comparación
 - ($=, >, <, \geq, \leq, <>$)
- Y los operadores lógicos:
 - \wedge (and), \vee (or), \neg (not)

Operación Selección (σ)

R

Nombre	Dirección	Fecha
Juan Díaz	C/Pez, 10	9-9-62
Ana Gómez	C/Luna, 3	8-3-58
Luis Díaz	C/ Sol, 5	8-5-67

La fórmula es $F = \text{Nombre} > \text{"B"}$

$\sigma_F(R)$

Nombre	Dirección	Fecha
Juan Díaz	C/Pez, 10	9-9-62
Luis Díaz	C/ Sol, 5	8-5-67

Ejemplo Selección (σ)

PERSONA

Cedula	Nombre	Primer_Apellido	Segundo_Apellido	Sexo	Dirección	Telefono	Salario
71134534	Juan	Mesa	Uribe	M	Cra 25 22-1	2567532	1,600,000
	Ana						
23423445	María	Betancur	Bermudez	F	Cra 45 11-13	3433444	1,300,000
12453535	Gloria	Betancur	Garces	F	Tr. 12 43-5	2756533	1,700,000
75556743	Pedro	Ochoa	Pelaez	M	Cll.6ta 14-45	2686885	1,200,000
43533322	Patricia	Angel	Guzmán	F	Cll. 45 23-1	2674563	1,350,000
78900456	Carlos	Betancur	Agudelo	M	Cir. 5 12-5	4445775	1,500,000

Ejemplo Selección (σ)

- 1.- Mostrar las personas con cedula=71134534
- 2.- Mostrar las personas de sexo Femenino
- 3.- Mostrar las personas que cumplan las dos condiciones anteriores
- 4.- Mostrar las personas de sexo Masculino o las que tengan un salario igual o superior a 1.350.000

- Muestra el resultado de las siguientes selecciones:

- $\sigma_{\text{cedula} = 71134534}(\text{PERSONA})$
- $\sigma_{\text{sexo} = "F"}(\text{PERSONA})$
- $\sigma_{(\text{primer_apellido} = "Bentancur") \wedge (\text{sexo} = "F")}(\text{PERSONA})$
- $\sigma_{\text{sexo} = "M" \vee (\text{salarios} \geq 1,350,000)}(\text{PERSONA})$

Ejemplo Selección (σ)

$\sigma_{\text{cedula} = 71134534}(\text{PERSONA})$

Resultado:

Cedula	Nombre	Primer_Apellido	Segundo_Apellido	Sexo	Dirección	Telefono	Salario
71134534	Juan	Mesa	Uribe	M	Cra 25 22-1	2567532	1,600,000

$\sigma_{\text{sexo} = "F"}(\text{PERSONA})$

Resultado:

Cedula	Nombre	Primer_Apellido	Segundo_Apellido	Sexo	Dirección	Telefono	Salario
	Ana						
23423445	María	Betancur	Bermudez	F	Cra 45 11-13	3433444	1,300,000
12453535	Gloria	Betancur	Garces	F	Tr. 12 43-5	2756533	1,700,000

Ejemplo Selección (σ)

$\sigma_{(\text{primer_apellido} = \text{"Betancur"}) \wedge (\text{sexo} = \text{"F"})}$ (PERSONA)

Resultado:

Cedula	Nombre	Primer_Apellido	Segundo_Apellido	Sexo	Dirección	Telefono	Salario
23423445	Ana	Betancur	Bermudez	F	Cra 45 11-13	3433444	1,300,000
12453535	Gloria	Betancur	Garces	F	Tr. 12 43-5	2756533	1,700,000

$\sigma_{\text{sexo} = \text{"M"}} \vee (\text{salarios} \geq 1,350,000)$ (PERSONA)

Resultado:

Cedula	Nombre	Primer_Apellido	Segundo_Apellido	Sexo	Dirección	Telefono	Salario
71134534	Juan	Mesa	Uribe	M	Cra 25 22-1	2567532	1,600,000
12453535	Gloria	Betancur	Garces	F	Tr. 12 43-5	2756533	1,700,000
75556743	Pedro	Ochoa	Pelaez	M	Cll.6ta 14-45	2686885	1,200,000
43533322	Patricia	Angel	Guzmán	F	Cll. 45 23-1	2674563	1,350,000
78900456	Carlos	Betancur	Agudelo	M	Cir. 5 12-5	4445775	1,500,000

Proyección (Π)

- Una proyección es una operación en la que seleccionamos aquellos campos que queremos recuperar.

- Formato de Uso: $\pi_{\langle \text{lista de atributos} \rangle}(\text{Relación})$
 - $\pi_{(x)}(R)$
- Ejemplos:
 - La proyección, $\pi_X(R)$, donde R es una relación definida sobre el conjunto de atributos T y $X \subseteq T$, es una relación constituida por las columnas de R correspondientes a los atributos de X

Proyección (Π)

R

Nombre	Dirección	Fecha
Juan Díaz	C/Pez,10	9-9-62
Ana Gómez	C/Luna,3	8-3-58
Luis Díaz	C/ Sol, 5	8-5-67

$\pi_{\text{nombre, fecha de Nacimiento}}(R)$

Nombre	Fecha
Juan Díaz	9-9-62
Ana Gómez	8-3-58
Luis Díaz	8-5-67

Ejemplos Proyección (Π)

- Muestre el resultado de las siguientes proyecciones:

- $\pi_{\text{cedula, nombre, primer_apellido, segundo_apellido}} (\text{PERSONA})$

- $\pi_{\text{cedula, salario}} (\text{PERSONA})$

- $\pi_{\text{cedula, nombre, salario}} (\sigma_{(\text{sexo} = \text{"M"}) \vee (\text{salario} \geq 1,350,000)} (\text{PERSONA}))$

Ejemplos Proyección (Π)

$\Pi_{\text{cedula, nombre, primer_apellido, segundo_apellido}}$ (PERSONA)

Resultado

Cedula	Nombre	Primer_Apellido	Segundo_Apellido
71134534	Juan	Mesa	Uribe
23423445	Ana María	Betancur	Bermudez
12453535	Gloria	Betancur	Garces
75556743	Pedro	Ochoa	Pelaez
43533322	Patricia	Angel	Guzmán
78900456	Carlos	Betancur	Agudelo

$\Pi_{\text{cedula, salario}}$ (PERSONA)

Resultado:

Cedula	Salario
71134534	1,600,000
23423445	1,300,000
12453535	1,700,000
75556743	1,200,000
43533322	1,350,000
78900456	1,500,000

Ejemplos Proyección (Π)

$\pi_{\text{cedula, nombre, salario}} \left(\sigma_{(\text{sexo} = \text{"M"}) \vee (\text{salario} \geq 1,350,000)} (\text{PERSONA}) \right)$

Resultado:

Cedula	Nombre	Salario
71134534	Juan	1,600,000
12453535	Gloria	1,700,000
75556743	Pedro	1,200,000
43533322	Patricia	1,350,000
78900456	Carlos	1,500,000

Reunión (Join) θ

- La reunión se utiliza para recuperar datos a través de varias tablas conectadas unas con otras mediante cláusulas JOIN. La operación reunión se puede combinar con las operaciones selección y proyección.
- Se forma el producto cartesiano R y S. Se selecciona, en el producto, solo la tupla que cumplan la condición C

R				S		
A	B	C	D	A	C	E
1	3	5	7	1	5	2
3	2	9	1	1	5	9
2	3	5	4	3	9	2
				2	3	7

Ejemplo Reunión (Join) θ

- La condición es: $C = A \geq E$

$R > _C S$	A	B	C	D	S.A	S.C	E
	3	2	9	1	1	5	2
	3	2	9	1	3	9	2
	2	3	5	4	1	5	2
	2	3	5	4	3	9	2

Reunión Natural (Join Natural)

- Sean R y S dos relaciones con uno o mas atributos en común. La reunión natural, $R \bowtie S$, se calcula del modo siguiente:
- Se calcula el producto cartesiano $R \times S$ Para cada atributo A_i común, se seleccionan las filas en las que el valor $R.A_i$ coinciden con el valor de $S.A_i$ Realizada la selección, eliminar la columna $S.A_i$

R	A	B	C	D
	1	3	5	7
	3	2	9	1
	2	3	5	4

S	A	C	E
	1	5	2
	1	5	9
	3	9	2
	2	3	7

$R \bowtie S$	A	B	C	D	E
	1	3	5	7	2
	1	3	5	7	9
	3	2	9	1	2

División ÷

- Operación del álgebra relacional que crea una nueva relación, seleccionando las filas en una relación que se corresponden con todas las filas en otra relación.

Sean R y S relaciones de grado r y s , respectivamente, donde $r > s$ y $S \neq \emptyset$. Entonces, el cociente, $R \div S$, es el conjunto de tuplas t de grado $(r-s)$, tales que para toda tupla u de S , la tupla (t, u) esta en R .

- En términos de operaciones básicas:

$$R \div S \equiv \pi_{1,2,\dots,r-s}(R) - \pi_{1,2,\dots,r-s}((\pi_{1,2,\dots,r-s}(R) \times S) - R)$$

División ÷

R	A	B	C	D
	1	2	3	5
	4	3	5	9
	3	2	8	1
	1	2	2	7
	1	3	2	7

S	C	D
	3	5
	2	7

$R \div S$	A	B
	1	2

Estructura del Lenguaje SQL

La estructura del lenguaje SQL contiene un limitado número de verbos o palabras clave, distribuidos en tres grandes grupos funcionales:

- DDL – Data Definition Language (lenguaje de descripción de datos) Permite la descripción de la estructura de la BD (tablas, vistas, índices,...)
- DML – Data Manipulation Language (lenguaje de manipulación de datos) Permite el manejo de las tablas y las vistas mediante sus cuatro verbos, correspondientes a las cuatro operaciones fundamentales sobre los datos.
- DCL – Data Control Language (lenguaje de control de datos) Contiene los operadores para la gestión de transacciones (COMMIT y ROLLBACK) y prioridades de acceso a los datos (GRANT y REVOKE)

■ Los Comandos mas importantes son:

DDL	DML	DCL
CREATE	SELECT	GRANT
DROP	INSERT	REVOKE
ALTER	DELETE	COMMIT
	UPDATE	ROLLBACK

Consulta de Datos con SQL

- El DQL - Data Query Language (lenguaje de consulta de datos) forma parte del lenguaje DML de SQL. El único comando que pertenece a este lenguaje es el comando SELECT.
- Este comando permite:
 - Obtener datos de ciertas columnas de una tabla (proyección).
 - Obtener registros (filas) de una tabla de acuerdo con ciertos criterios (selección)
 - Mezclar datos de tablas diferentes (union, join)
 - Realizar cálculos sobre los datos
 - Agrupar datos

Sintaxis sencilla del Select

```
SELECT * | { [DISTINCT] columna | expresión [AS] alias} , }  
FROM tabla;
```

- * => significa que se seleccionan todas las columnas
- DISTINCT. Hace que no se muestren los valores duplicados
- Columna. Es el nombre de la columna de la tabla que se desea mostrar
- expresión. Una expresión válida SQL
- Alias. Es el nombre que se le da a la cabecera de la columna en el resultado de esta instrucción

Ejemplos Select

- Selección de todos los registros de la tabla clientes:

```
SELECT * FROM Clientes;
```

- Selección de algunos campos:

```
SELECT nombre, apellido1,  
apellido2 FROM Clientes
```

Relación con el álgebra relacional

- El comando SELECT permite implementar de forma exacta todas las consultas del álgebra relacional.

- La **proyección** (π) se implementa así:

π nombre, apellido (Cliente)

En SQL se escribe como:

```
SELECT nombre, apellido FROM Cliente;
```

- La **selección** (σ) se implementa usando la cláusula WHERE

σ nombre="Pepe" \wedge edad > 25(Cliente)

En SQL se escribiría así:

```
SELECT * FROM Cliente WHERE nombre='Pepe' and edad > 25
```

- En la cláusula WHERE se pueden utilizar los siguientes operadores de comparación: >, <, >=, <=, =, <>
- Y los operadores lógicos: AND, OR, NOT

Ejemplos

/* Obtener las personas entre 25 y 50 años de edad */

❑ SELECT nombre, apellido FROM Personas
WHERE edad >= 25 AND edad <= 50;

/* Obtener a las personas de más de 60 años o de menos de 20 años */

❑ SELECT nombre, apellido FROM Personas
WHERE edad > 60 or edad < 20;

Relación con el álgebra relacional

- El **Producto Cartesiano** ($R \times S$) se implementa en SQL así:
`SELECT * FROM R, S`
- Ejemplo: Teniendo dos tablas una llamada Películas y otra Estudio. El Producto Cartesiano se escribiría así:
Álgebra relacional \rightarrow Película \times Estudio
SQL \rightarrow `SELECT * FROM Película, Estudio`

Ejemplo Producto Cartesiano

Película

ID_Película	Nombre	Año	ID_Estudio
1	La guerra de las galaxias	1977	3
2	La comunidad del anillo	2001	2
3	Mar adentro	2004	4
4	El viaje de Chihiro	2001	1

Estudio

ID_Estudio	Nombre
1	Ghibli
2	New Line Cinema
3	Lucasfilms
4	Sogecine

Ejemplo Producto Cartesiano

Película x Estudio

ID_Película	Nombre	Año	ID_Estudio	ID_Estudio	Nombre
1	La guerra de las galaxias	1977	3	1	Ghibli
1	La guerra de las galaxias	1977	3	2	New Line Cinema
1	La guerra de las galaxias	1977	3	3	Lucasfilms
1	La guerra de las galaxias	1977	3	4	Sogecine
2	La comunidad del anillo	2001	2	1	Ghibli
2	La comunidad del anillo	2001	2	2	New Line Cinema
2	La comunidad del anillo	2001	2	3	Lucasfilms
2	La comunidad del anillo	2001	2	4	Sogecine
3	Mar adentro	2004	4	1	Ghibli
3	Mar adentro	2004	4	2	New Line Cinema
3	Mar adentro	2004	4	3	Lucasfilms
3	Mar adentro	2004	4	4	Sogecine
4	El viaje de Chihiro	2001	1	1	Ghibli
4	El viaje de Chihiro	2001	1	2	New Line Cinema
4	El viaje de Chihiro	2001	1	3	Lucasfilms
4	El viaje de Chihiro	2001	1	4	Sogecine

Relación con el álgebra relacional

- La operación **Reunion** (Join) $R \bowtie S$

En SQL se escribe:

```
SELECT * FROM R, S WHERE R.c = S.c
```

- Usando las tablas anteriores
- ```
SELECT * Película, Estudio WHERE Pelicula.ID_Estudio = Estudio.ID_Estudio
```

# Ejemplo Join

| Película >< Estudio |                           |      |            |            |                 |
|---------------------|---------------------------|------|------------|------------|-----------------|
| ID_Película         | Nombre                    | Año  | ID_Estudio | ID_Estudio | Nombre          |
| 1                   | La guerra de las galaxias | 1977 | 3          | 3          | Lucasfilms      |
| 2                   | La comunidad del anillo   | 2001 | 2          | 2          | New Line Cinema |
| 3                   | Mar adentro               | 2004 | 4          | 4          | Sogecine        |
| 4                   | El viaje de Chihiro       | 2001 | 1          | 1          | Ghibli          |



# Relación con el álgebra relacional

- El operador **Renombrar** ( $\rho$ )  
Permite cambiar el nombre de la relación, para diferenciar atributos del mismo nombre.
- SQL proporciona un mecanismo para renombrar tanto relaciones como atributos. Para ello utiliza la cláusula AS, que tiene la forma siguiente:  
`SELECT * nombre-antiguo AS nombre-nuevo`

# Relación con el álgebra relacional

- Las operaciones de SQL UNION, INTERSECT y EXCEPT operan sobre relaciones y corresponden a las operaciones del álgebra relacional,  $\cup$ ,  $\cap$  y  $-$
- Al igual que la unión, intersección y diferencia de conjuntos en el álgebra relacional, las relaciones que participan en las operaciones han de ser compatibles; esto es, deben tener el mismo conjunto de atributos

## Ejemplo SELECT - UNIÓN

- Para encontrar todas las mascotas que están en una de las dos tablas o en las dos, se escribirá:
- (SELECT nombre FROM mascotas) UNION (SELECT nombre FROM copiamascotas3)

```
mysql> select nombre from mascotas UNION select nombre FROM copiamascotas3;
+-----+
| nombre |
+-----+
| horis |
| dante |
| donna |
| kira |
| laica |
| lorato |
| terry |
+-----+
7 rows in set (0.00 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_veterinario |
+-----+
| copiamascotas |
| copiamascotas2 |
| copiamascotas3 |
| mascotas |
| propietarios |
+-----+
5 rows in set (0.00 sec)
```

Inicialmente ambas tablas contienen los mismos datos. Vamos a añadir algunos elementos a la tabla mascotas:

```
mysql> insert into mascotas values (null,'don',default,'dogo',false,'1998-09-08',
'm','222333444a'),(null,'fox',default,'basset hound',true,'2002-09-07','m','222
333444a');
Query OK, 2 rows affected (0.04 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> select nombre from mascotas UNION select nombre FROM copiamascotas3;
+-----+
| nombre |
+-----+
| horis |
| dante |
| don |
| donna |
| fox |
| kira |
| laica |
| lorato |
| terry |
+-----+
9 rows in set (0.00 sec)
```

## Ejemplo SELECT - INTERSECCIÓN

- Para encontrar todas las mascotas que están en las dos tablas simultáneamente:
- `(SELECT nombre FROM mascotas)INTERSECT (SELECT nombre FROM copiamascotas3)`
- Para encontrar todos los clientes que tienen tanto un préstamo como una cuenta en el banco se escribiría:
- `(SELECT nom_clien FROM prestatario)INTERSECT (SELECT nom_clien FROMimpositor)`
- No mysql

# Ejemplo SELECT – DIFERENCIA

- Para encontrar todos los clientes que tienen cuenta pero no tienen ningún préstamo en el banco, se escribiría:

```
(SELECT nom_clien FROM prestatario)EXCEPT
(SELECT nom_clien FROM impositor)
```

No mysql

# BIBLIOGRAFÍA

- <http://es.scribd.com/doc/2450884/Algebra-Relacional#scribd>