

Álgebra Relacional



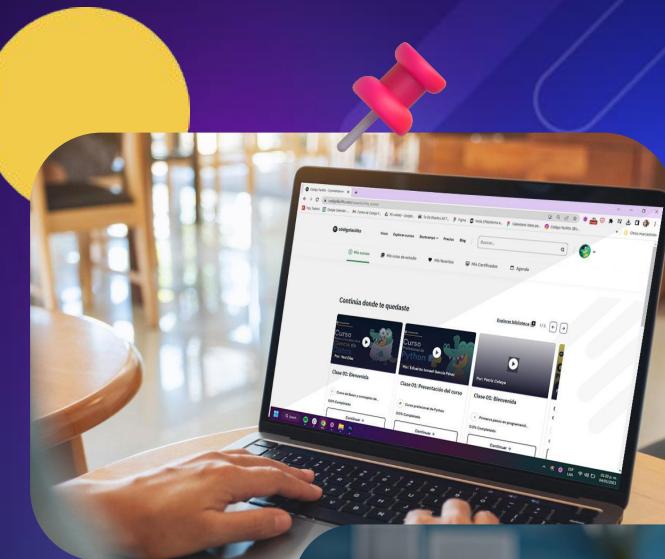
¿De qué se trata esta charla?

- Qué es el álgebra relacional
- Por qué es importante, y qué tiene que ver con SQL
- Operadores básicos, de conjuntos, y derivados



/01

Por qué esta clase es importante

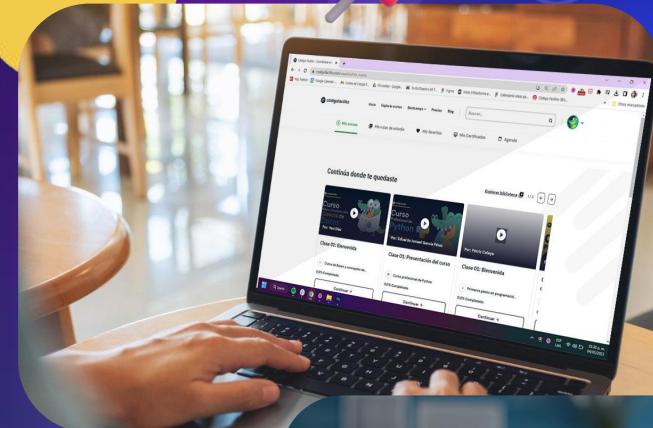




Aprendamos más que sólo la sintaxis

/02

Qué es el álgebra relacional



Álgebra relacional

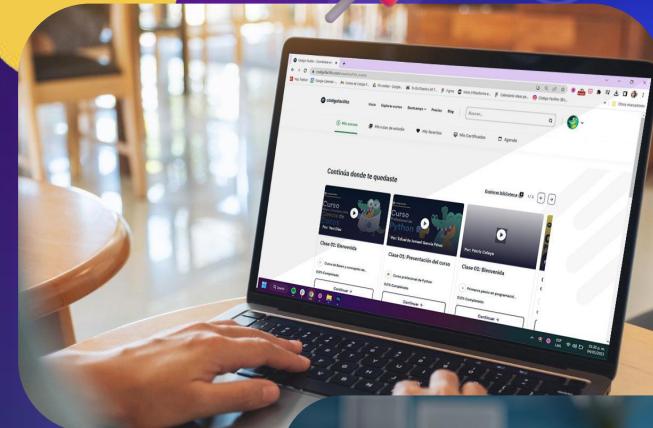
- Fundamento de SQL
- Manipulación de Datos
- Consultas Eficientes

Diferencias entre SQL y álgebra relacional

Aspecto	Álgebra Relacional	SQL
Estructura de Datos	Relaciones (conjuntos de tuplas)	Tablas (filas y columnas)
Duplicados	No permite duplicados (conjuntos)	Permite duplicados (multisets/bags)
Expresión de Datos	Relación de tuplas con nombres de atributos y valores	Filas de datos con atributos definidos en columnas
Operaciones	Basado en operadores matemáticos como selección (σ), proyección (Π), unión (U), etc.	Lenguaje declarativo con comandos como SELECT, JOIN, GROUP BY, etc.
Orden de Resultados	No garantiza un orden específico	Puede ordenar los resultados con ORDER BY
Definición Teórica	Formalismo matemático basado en la teoría de conjuntos	Lenguaje de consulta utilizado en bases de datos

/03

Prepara tu entorno para la clase



Schema referencia

Usuario

uID	nombre	rol	edad

Curso

cID	titulo	duracion

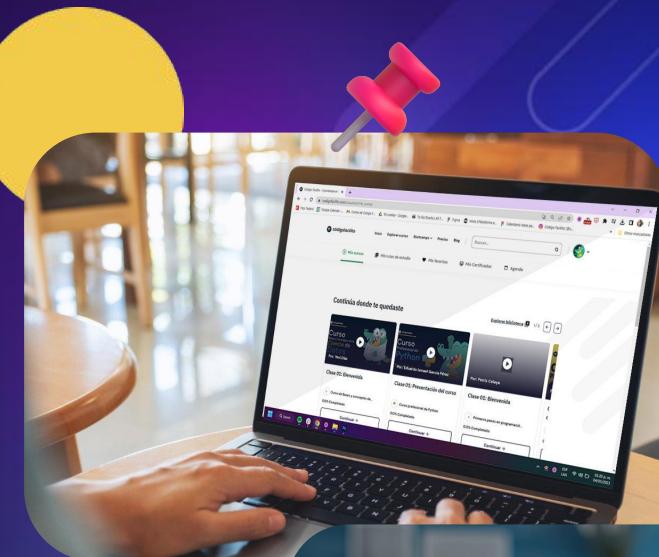
Progreso_Curso

pclID	cID	uID	progreso

Herramienta de visualización

/04

Operadores



**Los operadores aceptan una
o dos relaciones y retornan
una relación como resultado**

Select

El operador select σ , representado por el operador sigma, cuya sintaxis incluye una o varias condiciones, y una relación:

$$\sigma c1..cn(R)$$

Select

Ejercicio 1: Muestra todos los cursos con duración en segundos mayor a 1200

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Select

Ejercicio 2: Usuarios mayores de 30 que sean estudiantes

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Select

Ejercicio 3: Cursos que cuyo ID es menor que 100, o que duren menos de 100 segundos

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Ahora ponlo en práctica

- Ejercicio 1: Usuarios menores de 25 años que sean desarrolladores
- Ejercicio 2: Progresos de curso donde el usuario ha completado al menos el 50% del curso
- Ejercicio 3: Cursos con título 'Curso de SQL' con duración mayor a 100, y ID menor que 100

Proyección

El operador de proyección (Π) representado por la pi mayúscula. Permite extraer columnas (atributos) de una relación, dando como resultado un subconjunto vertical de atributos de la relación.

$$\Pi A_1, A_2 \dots A_n (\text{relación})$$

Select

Ejercicio 1: Obtener solo los títulos de todos los cursos:

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Select

Ejercicio 2: Obtener los nombres y roles de todos los usuarios:

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Select

Ejercicio 3: Obtén los nombres de los usuarios cuya edad sea mayor a 18

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

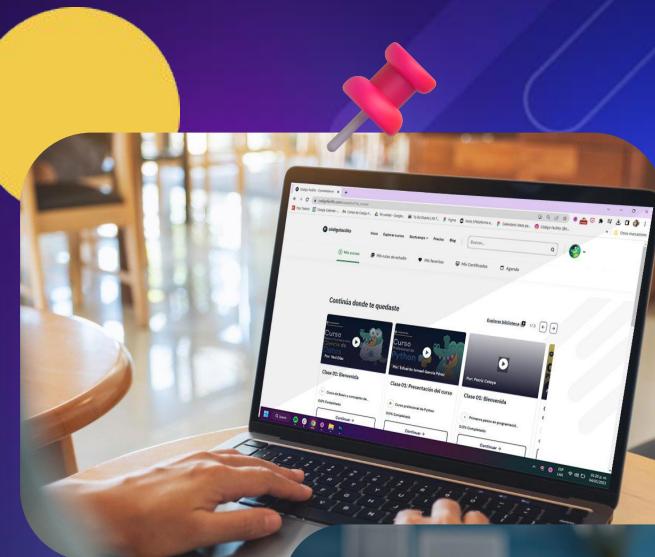
pclID	cID	uID	progreso

Ahora ponlo en práctica

- Ejercicio 1: Obtener los títulos de los cursos y sus duraciones, sin incluir el ID:
- Ejercicio 2: Obtener el cID de los cursos cuyo progreso sea mayor que 50
- Ejercicio 3: Obtén una lista con los roles únicos de usuarios

/05

Operadores conjuntos



**El álgebra relacional está
basada en la teoría de
conjuntos**

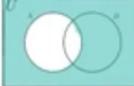
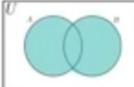
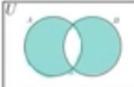
File Home Comment View Form Protect Foxit eSign Share Help Tell me... Convert PDF to JPG Images

Hand Select Note File Typewriter Textbox Callout Drawing Pencil Eraser Area Highlight Search & Highlight Measure Stamp Create Summarize Import Export FDF via Email Manage Comments Keep Tool Selected

Lógica y Conjuntos - Caso ... Lógica y Conjuntos - Cheatsheet (DIY).pdf - Foxit PDF Reader

Bookmarks

File:///C:/Users/dell/Downloads/d65d82a0caab7a49c7c890a59a1d99.pdf

Tipo	Español	Notación Matemática	Álgebra Booleana	Conjuntos ¹	Diagrama de Venn	Circuitos ¹	Python	Tabla de Verdad															
NOT	Negación	$\neg p$	β	$A^c = A'$			not P	<table border="1"><tr><td>p</td><td>$\neg p$</td></tr><tr><td>T</td><td>F</td></tr><tr><td>F</td><td>T</td></tr></table>	p	$\neg p$	T	F	F	T									
p	$\neg p$																						
T	F																						
F	T																						
AND	Conjunción	$p \wedge q$	$P \cdot Q$	$A \cap B$			P and Q	<table border="1"><tr><td>p</td><td>q</td><td>$p \wedge q$</td></tr><tr><td>T</td><td>T</td><td>T</td></tr><tr><td>T</td><td>F</td><td>F</td></tr><tr><td>F</td><td>T</td><td>F</td></tr><tr><td>F</td><td>F</td><td>F</td></tr></table>	p	q	$p \wedge q$	T	T	T	T	F	F	F	T	F	F	F	F
p	q	$p \wedge q$																					
T	T	T																					
T	F	F																					
F	T	F																					
F	F	F																					
OR	Disyunción	$p \vee q$	$P + Q$	$A \cup B$			P or Q	<table border="1"><tr><td>p</td><td>q</td><td>$p \vee q$</td></tr><tr><td>T</td><td>T</td><td>T</td></tr><tr><td>T</td><td>F</td><td>T</td></tr><tr><td>F</td><td>T</td><td>T</td></tr><tr><td>F</td><td>F</td><td>F</td></tr></table>	p	q	$p \vee q$	T	T	T	T	F	T	F	T	T	F	F	F
p	q	$p \vee q$																					
T	T	T																					
T	F	T																					
F	T	T																					
F	F	F																					
XOR	Disyunción exclusiva	$p \veebar q$	$P \oplus Q$	$A \triangle B$			$P \wedgebar Q$																

¹ Recuerda que los conjuntos tienen la propiedad de ser disjuntos, es decir, no tienen elementos en común.

Navigation icons: back, forward, search, etc. at the bottom left.

Page number: 1/2

Page footer: Cynthia Castillo

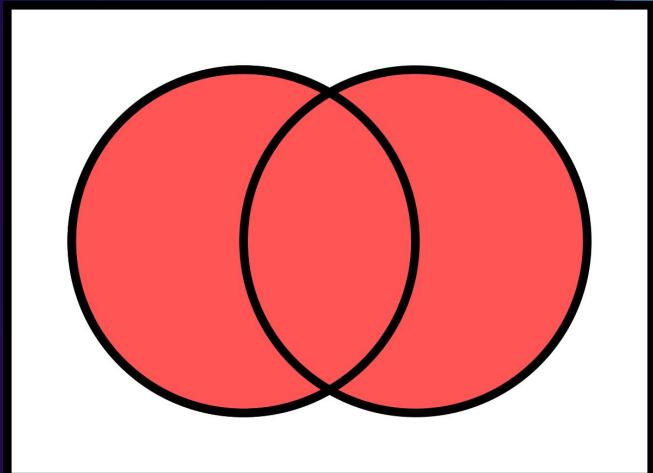
Page footer: 7:11 1x

Esquemas compatibles

- Algunos de los operadores de conjuntos en álgebra relacional requieren que las relaciones tengan esquemas compatibles entre sí.
- Esto quiere decir que las columnas de los datos sean iguales o del mismo tamaño, dependiendo el contexto en el que estemos.

Union

- En teoría de conjuntos el operador unión (\cup) “combina” todas las tuplas de un set en una lista vertical.



Union

Obtén la lista de nombres de usuario y títulos de un curso

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

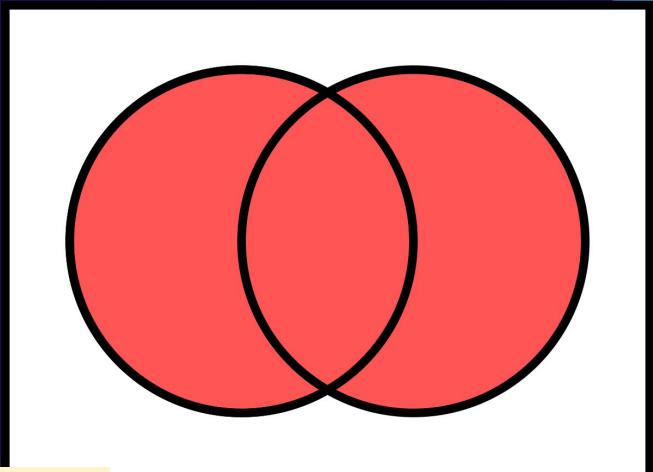
Progreso_Curso

pclID	cID	uID	progreso

Union

- En teoría de conjuntos el operador unión (\cup) representa todos las tuplas de los sets

Sólo se puede aplicar a esquemas iguales o compatibles.

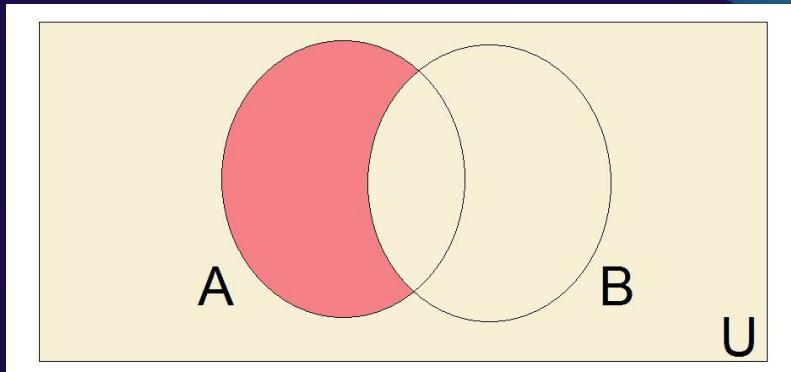


Diferencia

El operador de diferencia (-) permite obtener la diferencia de dos relaciones, la sintaxis:

A-B

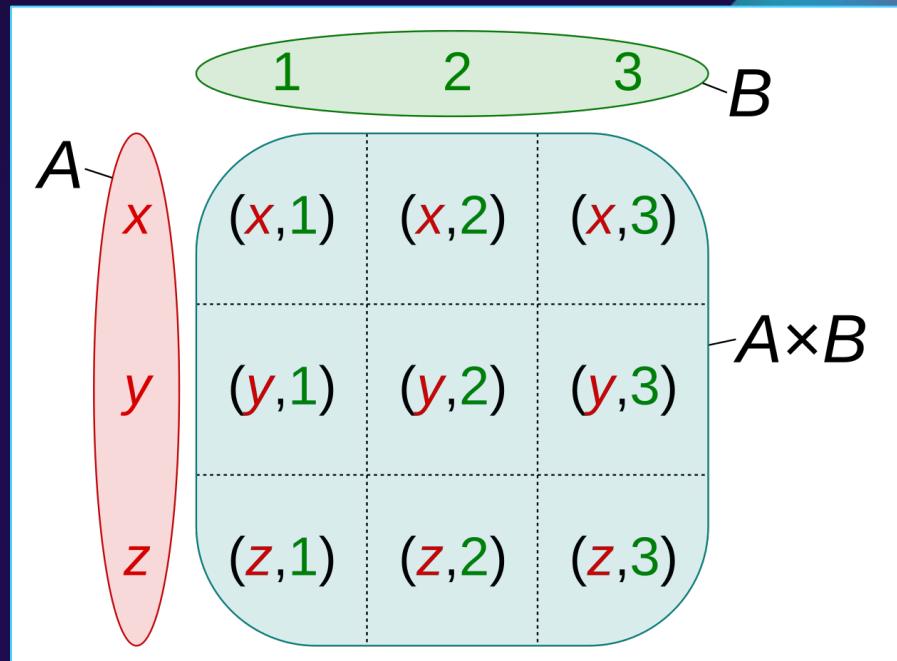
El resultado son las tuplas de A, menos las tuplas de B.



Producto cartesiano

La operación de producto cartesiano (\times) permite obtener el producto cartesiano de dos relaciones.

El producto cartesiano son todas las combinaciones posibles de las tuplas de una relación con otra.



Producto cartesiano

¿Cuál sería el producto cartesiano para esta operación?

Curso

cID	titulo	duracion
1	SQL	200
2	Álgebra	500

Usuario

uID	nombre	rol	edad
1	Uriel	dev	31

Progreso_Curso

pclID	cID	uID	progreso

Producto cartesiano

¿Cuál sería el producto cartesiano para esta operación?

cID	Título	Duración	uID	Nombre	Rol	Edad
1	SQL	200	1	Uriel	dev	31
2	Álgebra	500	1	Uriel	dev	31

Curso

Usuario

Progreso_Curso

cID	titulo	duracion
1	SQL	200
2	Álgebra	500

uID	nombre	rol	edad
1	Uriel	dev	31

pclID	cID	uID	progreso

Producto cartesiano

Ejercicio 1: Obtener el producto cartesiano entre curso y progreso_curso

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Producto cartesiano

Ejercicio 2: Obtén el nombre del usuario y el id del curso tomado para cada combinación posible

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Producto cartesiano

Ejercicio 3: Obtener el nombre del usuario, y el progreso de todos los cursos que está tomando un usuario

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

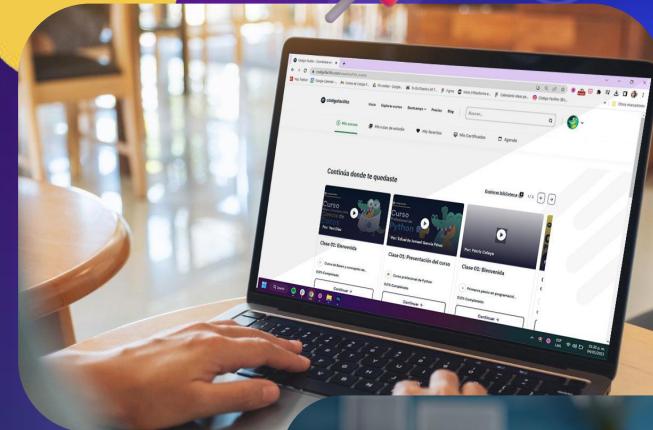
pclID	cID	uID	progreso

Ahora ponlo en práctica

- Ejercicio 1: Despliega título del curso y nombre de usuario de todos los usuarios con los cursos que están tomando

/06

Operadores derivados



Operadores derivados

Pueden considerarse una abreviación de operaciones que se pueden hacer con los anteriores vistos.

Natural Join

El operador Natural Join(\bowtie), combina las tuplas cuyo valor es igual, en los campos con nombres iguales, además de usar proyección para quitar los campos duplicados.

Natural Join

El operador natural join entonces:

1. Ejecuta un producto cartesiano
2. Valida igualdad en los atributos con el mismo nombre
3. Proyecta eliminando columnas con nombre repetido

Natural Join

La relación entre el usuario y el progreso_curso

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Natural Join

La relación entre el usuario y el curso

Curso

cID	titulo	duracion

Usuario

uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

Theta Join

El operador join puede incluir también condiciones personalizadas de combinación, a esta operación la llamamos un join o un theta join ($\bowtie\Theta$), dependiendo el recurso que consulten.

Natural Join

El operador natural join entonces:

1. Ejecuta un producto cartesiano
2. ~~Valida igualdad en los atributos con el mismo nombre.~~ Ejecuta las condiciones de la operación join
3. Proyecta eliminando columnas con nombre repetido

Theta Join

Obtén los usuarios y el progreso de los cursos en que están inscritos

Curso

cID	titulo	duracion

Usuario

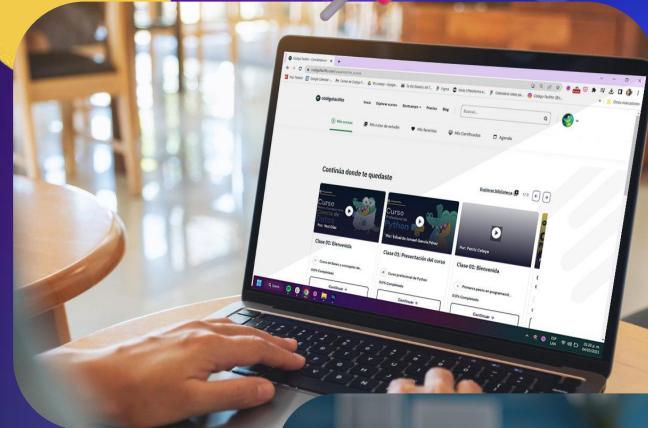
uID	nombre	rol	edad

Progreso_Curso

pclID	cID	uID	progreso

/07

Recapitulando





¿Qué es lo más importante
que aprendimos hoy?

```
1 SELECT Usuario.nombre, Curso.titulo, Progreso_Curso.progreso  
2 FROM Usuario  
3 INNER JOIN Progreso_Curso ON Usuario.uID = Progreso_Curso.uID  
4 INNER JOIN Curso ON Progreso_Curso.cID = Curso.cID;
```



```
1 SELECT U.nombre, C.titulo, PC.progreso
2 FROM Usuario U
3 JOIN (
4     SELECT uID, cID, progreso
5     FROM Progreso_Curso
6     WHERE progreso = (SELECT MAX(progreso) FROM Progreso_Curso WHERE
7       Progreso_Curso.uID = U.uID)
8   ) PC ON U.uID = PC.uID
9   JOIN Curso C ON PC.cID = C.cID;
```



```
1 SELECT titulo AS categoria, 'Curso' AS tipo FROM Curso  
2 UNION  
3 SELECT rol AS categoria, 'Rol de Usuario' AS tipo FROM Usuario;
```



Muchas gracias