



UNIVERSIDAD
PANAMERICANA

Introducción a las Bases de Datos

Dr. Leon Felipe Palafox Novack
lpalafox@up.edu.mx

0

Anuncios parroquiales

Examen 2



■ Octubre 22 y 24:

- ▷ Para ir acorde al calendario de nuestra clase:
 - ▷ Algebra Relacional
- ▷ Va a ser en dos partes:
 - ▷ 1º Examen Individual (22 Octubre)
 - ▷ 2ª Evaluación Grupal (24 Octubre)
 - ▷ Calif = 1er Parcial + 2º Parcial

2

Algebra Relacional

Operaciones

■ $R1 := \sigma_c(R2)$

▷ C es una condición

■ $R1 := \pi_L(R2)$

▷ L es una lista

■ $R1 := \pi_{[A+B \rightarrow C, A, A]}(R2)$

■ $R3 : R1 \times R2$

■ $R3 := R1 \bowtie_c R2$

Natural Join

■ $R3 := R1 \bowtie R2$

- ▷ Una variante del Theta-Join
- ▷ Conecta dos relaciones:
 - ▷ Iguala atributos con el mismo nombre
 - ▷ Proyecta una copia de cada par de atributos igualados.

Natural - Join

R1

Plaza	Tienda	Número
Parque Delta	Starbucks	2
Parque Delta	Marti	1
Galerías Insurgentes	Starbucks	1
Galerías Insurgentes	Marti	1

R2

Plaza	Colonia
Parque Delta	Roma
Galerías Insurgentes	Del Valle
Perisur	Pedregal

Natural-Join

R3

Plaza	Tienda	Número	Colonia
Parque Delta	Starbucks	2	Roma
Parque Delta	Marti	1	Roma
Galerías Insurgentes	Starbucks	1	Del Valle
Galerías Insurgentes	Marti	1	Del Valle

3

Algebra Relacional / SQL

Arboles de Expresión



- Una forma de visualizar los “queries” de algebra relacional es crear árboles de expresión.
- Son abstracciones de las operaciones.
- Nos permiten tener una visión estructurada de como se van filtrando las tablas.

- Dada la relación Clases(nombre, profesor, dificultad), cuales son las clases que imparte Palafox, que son fáciles.

- Usando las relaciones, Bares(nombre, dirección) y Ventas(bar, ceveza, precio), encontrar los nombres de todos los bares que estén en Insurgentes o vendan Tecate por menos de \$20.

Joins

- Inner Join
- Outer Join
- Left Inner Join
- Left Outer Join
- Right Inner Join
- Right Outer Join

Tablas

Table A

id	Nombre
1	Frodo
2	Gandalf
3	Harry Potter
4	Darth Vader

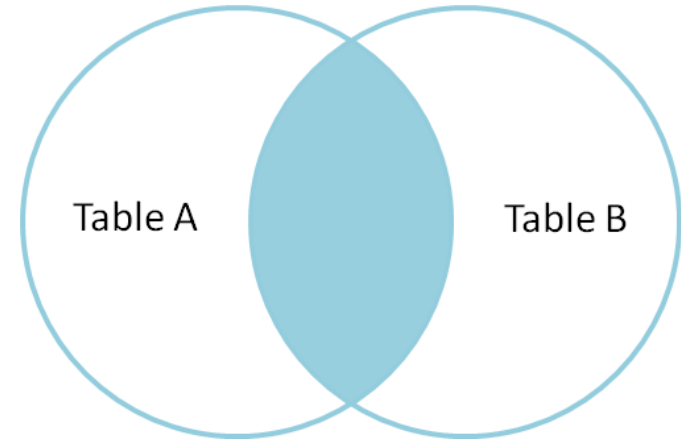
Table B

id	Nombre
1	Kermit
2	Frodo
3	Darth Vader
4	Jon Snow

Inner Join

```
SELECT * FROM TableA  
INNER JOIN TableB  
ON TableA.name = TableB.name
```

id	name	id	name
1	Frodo	2	Frodo
4	Darth Vader	3	Darth Vader

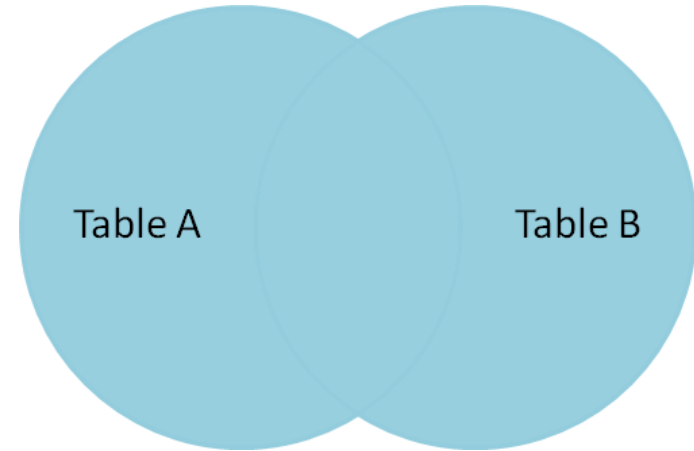


Sólo los matches entre A y B

Outer Join

```
SELECT * FROM TableA  
FULL OUTER JOIN TableB  
ON TableA.name = TableB.name
```

Produce todos los elementos de A y B, y donde no existe un match, coloca un NULL.

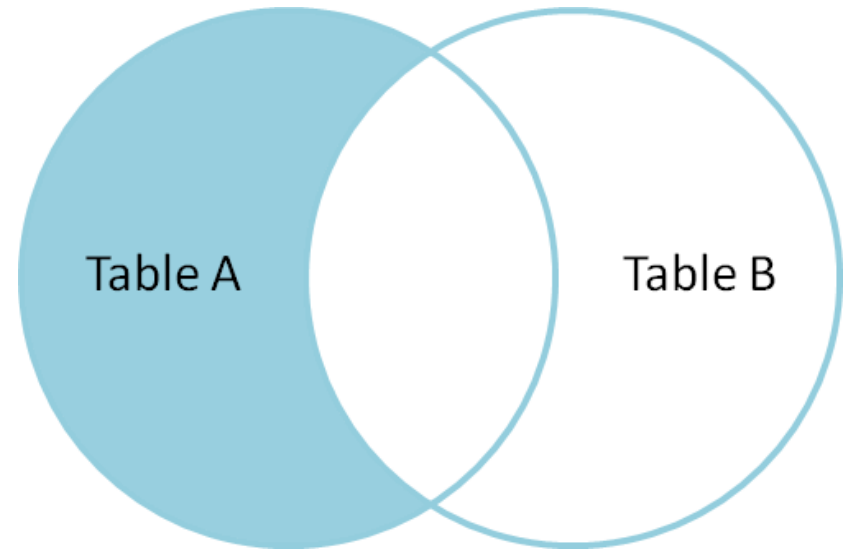


Outer Join

id	name	id	name
1	Frodo	2	Frodo
2	Gandalf	NULL	NULL
3	Harry Potter	NULL	NULL
4	Darth Vader	3	Darth Vader
NULL	NULL	1	Kermit
NULL	NULL	4	Jon Snow

Left Outer Join

```
SELECT * FROM TableA  
LEFT OUTER JOIN TableB  
ON TableA.name = TableB.name  
WHERE TableB.id IS null
```



Left Outer Join

id	name	id	name
2	Gandalf	NULL	NULL
3	Harry Potter	NULL	NULL

Terminamos Álgebra Relacional!

SQL

SELECT

- Select **A1, A2, A3,, AN** Que regresar
- From **R1** Que relaciones
- WHERE Condition

- Las instrucciones WHERE pueden tener condiciones donde la comparamos con un string:
 - ▷ <Atributo> LIKE <Patron> o
 - ▷ <Atributo> NOT LIKE <Patron>
- Patron es un string.

Patrones

```
SELECT
    customer_id,
    first_name,
    last_name
FROM
    sales.customers
WHERE
    last_name LIKE 'z%'
ORDER BY
    first_name;
```

```
1 SELECT
2     customer_id,
3     first_name,
4     last_name
5 FROM
6     sales.customers
7 WHERE
8     last_name LIKE '%er'
9 ORDER BY
10    first_name;
```



```
1 SELECT
2     customer_id,
3     first_name,
4     last_name
5 FROM
6     sales.customers
7 WHERE
8     last_name LIKE '_u%'
9 ORDER BY
10    first_name;
```

```
1 SELECT
2     customer_id,
3     first_name,
4     last_name
5 FROM
6     sales.customers
7 WHERE
8     last_name LIKE 't%s'
9 ORDER BY
10    first_name;
```

QUESTIONS?