

# UNIDAD 9

## CONSULTAS

## MULTITABLA

“Encontrarse es el comienzo de la  
separación”

Proverbio japonés

# CONSULTAS MULTITABLA

## Contenidos

- ⦿ Producto cartesiano
- ⦿ Union
- ⦿ Diferencia
- ⦿ Join
  - INNER y OUTER JOIN
- ⦿ Subconsultas

# PRODUCTO CARTESIANO

Consiste en generar todas las combinaciones posibles entre las filas de las dos tablas

Letras
L
E
O

Numeros
4
7

Letras x Numeros	
L	4
E	4
O	4
L	7
E	7
O	7

# PRODUCTO CARTESIANO

- ◉ Forma explícita

Esta es la forma correcta ANSI-SQL

```
SELECT * FROM Letras CROSS JOIN Numeros
```

- ◉ Forma implícita

Clásica. No ANSI

```
SELECT * FROM Letras, Numeros
```

- ◉ Esta última puede encontrarse en textos antiguos o poco cuidados



# UNIÓN

Une las filas no repetidas de dos conjuntos de resultados con la misma estructura

SELECT \* FROM Gente

Nombre	Altura
Enrique	175
Cristina	172
Diego	170

SELECT \* FROM Personas

Nombre	Altura
Adela	165
Diego	183
Iñigo	175
Cristina	172

SELECT \* FROM Gente  
UNION  
SELECT \* FROM Personas

Nombre	Altura
Enrique	175
Cristina	172
Diego	170
Adela	165
Diego	183
Iñigo	175

# INTERSECCIÓN

Obtiene las filas **repetidas** de dos conjuntos de resultados con la misma estructura

SELECT \* FROM Gente

Nombre	Altura
Enrique	175
Cristina	172
Diego	170

SELECT \* FROM Personas

Nombre	Altura
Adela	165
Diego	183
Iñigo	175
Cristina	172

SELECT \* FROM Gente  
INTERSECT  
SELECT \* FROM Personas

Nombre	Altura
Cristina	172

# DIFERENCIA

Obtiene las filas del primer conjunto de datos que no estén en el segundo

No es conmutativa

SELECT \* FROM Gente

Nombre	Altura
Enrique	175
Cristina	172
Diego	170

SELECT \* FROM Gente  
EXCEPT  
SELECT \* FROM Personas

Nombre	Altura
Enrique	175
Diego	170

SELECT \* FROM Personas

Nombre	Altura
Adela	165
Diego	183
Iñigo	175
Cristina	172

SELECT \* FROM Personas  
EXCEPT  
SELECT \* FROM Gente

Nombre	Altura
Adela	165
Diego	183
Iñigo	175



# JOIN

- Cada fila estará compuesta por filas de las dos tablas para las que coincida el valor de una (o mas) columna (columna de Join)
- Habitualmente se hace coincidir el valor de una columna de una tabla con la columna que referencia

```
SELECT R.Denominacion, R.GoesTo, C.ID,  
       C.Nombre
```

Tipo de JOIN Tabla Alias AS R

INNER JOIN Columnas de JOIN Tabla AS C

ON R.GoesTo = C.ID



# INNER JOIN

IDProv	Denominacion
1	Sevilla
2	Huelva
3	Soria

ID	Nombre	Provincia
10	Azucena	1
11	Rosa	2
12	Pepe	2
15	Margarita	1
16	Eloisa	
20	Mario	1

**SELECT \* FROM Provincias AS P INNER JOIN Amigos AS A  
ON P.IDProv = A.Provincia**

IDProv	Denominacion	ID	Nombre	Provincia
1	Sevilla	10	Azucena	1
2	Huelva	11	Rosa	2
2	Huelva	12	Pepe	2
1	Sevilla	15	Margarita	1
1	Sevilla	20	Mario	1

# TIPOS DE JOIN

## ◉ INNER JOIN

- Incluye únicamente las filas en las que coincida el valor de la columna de JOIN

## ◉ LEFT (OUTER) JOIN

- Incluye también las filas de la izquierda que no tengan correspondencia en la derecha

## ◉ RIGHT (OUTER) JOIN

- Incluye también las filas de la derecha que no tengan correspondencia en la izquierda

## ◉ FULL (OUTER) JOIN

- Incluye todas las filas de ambas tablas aunque no tengan correspondencia en la otra
- Todos los OUTER JOIN rellenan con nulos

# LEFT JOIN

IDProv	Denominacion
1	Sevilla
2	Huelva
3	Soria

ID	Nombre	Provincia
10	Azucena	1
11	Rosa	2
12	Pepe	2
15	Margarita	1
16	Eloisa	
20	Mario	1

SELECT \* FROM Provincias AS P LEFT JOIN Amigos AS A  
ON P.IDProv = A.Provincia

IDProv	Denominacion	ID	Nombre	Provincia
1	Sevilla	10	Azucena	1
2	Huelva	11	Rosa	2
2	Huelva	12	Pepe	2
1	Sevilla	15	Margarita	1
1	Sevilla	20	Mario	1
3	Soria	Null	Null	Null

# RIGHT JOIN

IDProv	Denominacion
1	Sevilla
2	Huelva
3	Soria

ID	Nombre	Provincia
10	Azucena	1
11	Rosa	2
12	Pepe	2
15	Margarita	1
16	Eloisa	Null
20	Mario	1

SELECT \* FROM Provincias AS P RIGHT JOIN Amigos AS A  
ON P.IDProv = A.Provincia

IDProv	Denominacion	ID	Nombre	Provincia
1	Sevilla	10	Azucena	1
2	Huelva	11	Rosa	2
2	Huelva	12	Pepe	2
1	Sevilla	15	Margarita	1
Null	Null	16	Eloisa	Null
1	Sevilla	20	Mario	1

# FULL JOIN

IDProv	Denominacion
1	Sevilla
2	Huelva
3	Soria

ID	Nombre	Provincia
10	Azucena	1
11	Rosa	2
12	Pepe	2
15	Margarita	1
16	Eloisa	Null
20	Mario	1

SELECT \* FROM Provincias AS P FULL JOIN Amigos AS A  
ON P.IDProv = A.Provincia

IDProv	Denominacion	ID	Nombre	Provincia
1	Sevilla	10	Azucena	1
2	Huelva	11	Rosa	2
2	Huelva	12	Pepe	2
1	Sevilla	15	Margarita	1
Null	Null	16	Eloisa	Null
1	Sevilla	20	Mario	1
3	Soria	Null	Null	Null

# SUBCONSULTAS

- ◉ Podemos usar una consulta para crear otra mediante la forma `SELECT FROM SELECT`
- ◉ En general es menos eficiente, ya que es más restrictiva para el optimizador
- ◉ Es útil para consultas muy complejas, porque nos permite hacerlas por partes
- ◉ Ej (NorthWind): Productos que se han vendido en Europa en más de un año y que no han sido vendidos por ningún empleado que tenga algún subordinado

# SUBCONSULTAS

- ◉ También se puede usar una consulta en el WHERE además del FROM
- ◉ La consulta se comporta como un conjunto

Las formas más usadas son

- ◉ WHERE elemento IN (SELECT...)
  - El SELECT debe devolver una sola columna
- ◉ WHERE (NOT) EXISTS (SELECT...)
  - Comprueba si el SELECT devuelve alguna fila



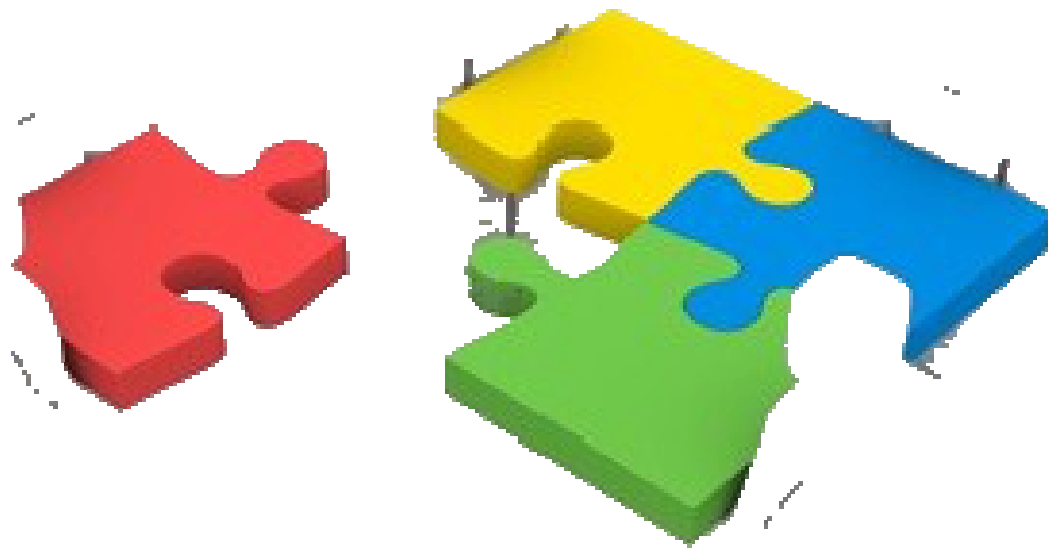
# SUBCONSULTAS

- Para simplificar la consulta se puede definir la subconsulta al principio

Opcional

WITH nombre AS (col1, col2...) (SELECT...





“Si estamos juntos no hay nada imposible. Si estamos divididos todo fallará”

Winston Churchill

