

Unidad de Trabajo 4  
Realización de consultas

# Consultas de combinación

IES Palomeras Vallecas  
Curso 2020/2021  
Profesor: Alberto Ruiz

# La cláusula UNION

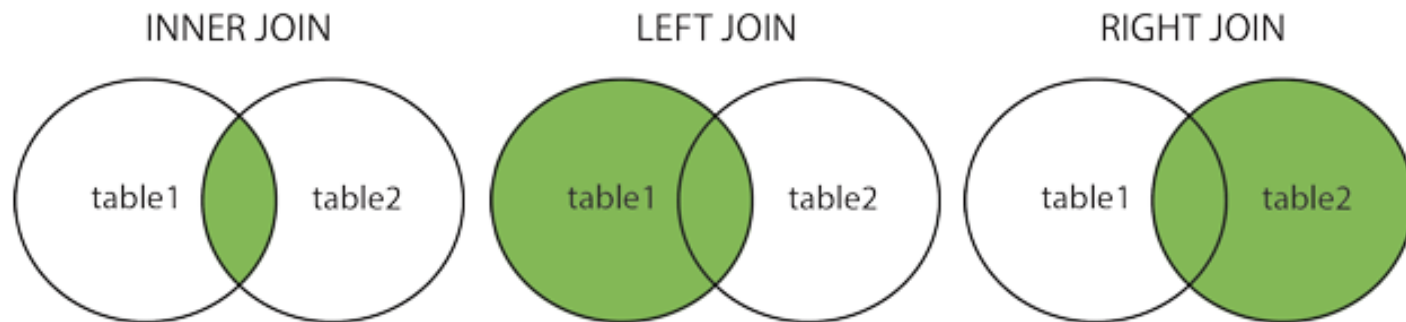
---

- Si queremos obtener los resultados de dos (o más) consultas independientes, utilizamos UNION
  - `SELECT CódigoPostal FROM Oficinas`  
`UNION`  
`SELECT CódigoPostal FROM Clientes`
- UNION elimina los duplicados que pueda haber en las dos consultas. Si quieres mantener los resultados duplicados:
  - `SELECT Ciudad FROM Profesores`  
`UNION ALL`  
`SELECT Ciudad FROM Alumnos`

# La cláusula JOIN

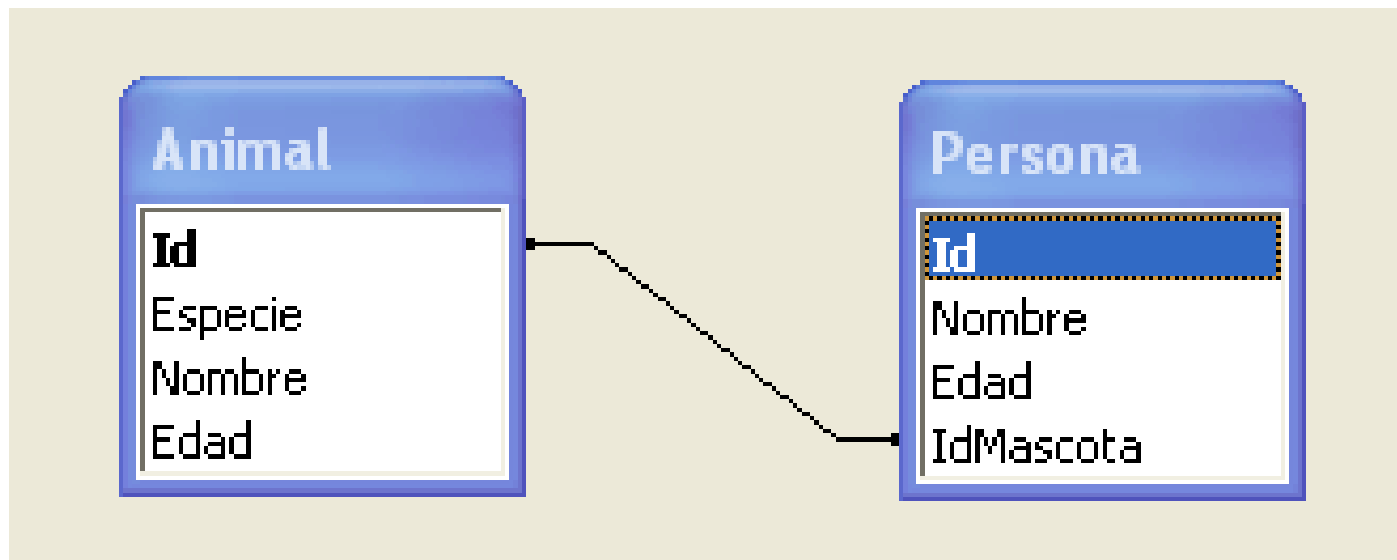
---

- Hemos estudiado consultas multitabla de la forma más sencilla para obtener datos relacionados: producto cartesiano con condición
- La sentencia JOIN (*unir, combinar*) ofrece más precisión a la hora de seleccionar los datos deseados:



# Consultas de combinación

- Considera esta base de datos sencilla:
  - Hay personas sin mascota
  - Hay animales sin dueño



# Consultas de combinación

Persona : Tabla				
	Id	Nombre	Edad	IdMascota
	1	Juan Pérez	34	3
	2	Sergio García	43	
	3	Andrés López	15	2
	4	Susana López	13	6
	5	Julio Ruiz	50	
	6	Inés Sánchez	45	5

Animal : Tabla					
		Id	Especie	Nombre	Edad
	+	1	Perro	Kity	6
	+	2	Conejo	Nejo	4
	+	3	Cobaya	Baya	1
	+	4	Tortuga	Sara	15
	+	5	Perro	Baxter	10
	+	6	Hurón	Juanito	3
	+	7	Gato	Misi	3

# Producto Cartesiano

---

```
SELECT *  
FROM Persona, Animal;
```

Combina los registros de las dos tablas en todas sus combinaciones posibles.

Si Persona tiene 6 registros y Animal tiene 7 registros, la consulta devolverá  $6 \times 7 = 42$  registros

# Producto Cartesiano

Muestra todas las combinaciones, pero sólo unas pocas están relacionadas:

cartesiano : Consulta de selección

	Persona.Id	Persona.Nombre	Persona.Edad	IdMascota	Animal.Id	Especie	Animal.Nombre	Animal.Edad
▶	1	Juan Pérez	34	3	1	Perro	Kity	6
	2	Sergio García	43		1	Perro	Kity	6
	3	Andrés López	15	2	1	Perro	Kity	6
	4	Susana López	13	6	1	Perro	Kity	6
	5	Julio Ruiz	50		1	Perro	Kity	6
	6	Inés Sánchez	45	5	1	Perro	Kity	6
	1	Juan Pérez	34	3	2	Conejo	Nejo	4
	2	Sergio García	43		2	Conejo	Nejo	4
	3	Andrés López	15	2	2	Conejo	Nejo	4
	4	Susana López	13	6	2	Conejo	Nejo	4
	5	Julio Ruiz	50		2	Conejo	Nejo	4
	6	Inés Sánchez	45	5	2	Conejo	Nejo	4
	1	Juan Pérez	34	3	3	Cobaya	Baya	1
	2	Sergio García	43		3	Cobaya	Baya	1
	3	Andrés López	15	2	3	Cobaya	Baya	1

Registro: 1 de 42

# Producto Cartesiano con condición

```
SELECT *  
FROM Persona, Animal  
WHERE Persona.IdMascota = Animal.Id;
```

- Lo estudiamos en el tema anterior:
  - Se calcula el producto cartesiano ( $6 \times 7 = 42$  registros)
  - Se filtra por la condición

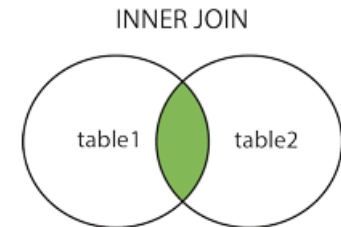
cartesiano condición : Consulta de selección								
	Persona.Id	Persona.Nombre	Persona.Edad	IdMascota	Animal.Id	Especie	Animal.Nombre	Animal.Edad
▶	1	Juan Pérez	34	3	↔ 3	Cobaya	Baya	1
	3	Andrés López	15	2	↔ 2	Conejo	Nejo	4
	4	Susana López	13	6	↔ 6	Hurón	Juanito	3
	6	Inés Sánchez	45	5	↔ 5	Perro	Baxter	10



# Inner Join

```
SELECT *  
FROM Persona, Animal  
WHERE Persona.IdMascota = Animal.Id;
```

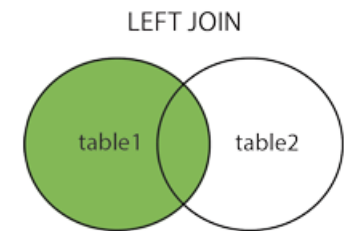
```
SELECT *  
FROM Persona INNER JOIN Animal  
ON Persona.IdMascota=Animal.Id;
```



- **El resultado es el mismo que en el producto cartesiano con condición**
- INNER JOIN es más rápido si alguno de los campos están indizados
  - Más adelante se explicará el concepto de índice

# Left Join

```
SELECT *  
FROM Persona LEFT JOIN Animal  
ON Persona.IdMascota=Animal.Id;
```

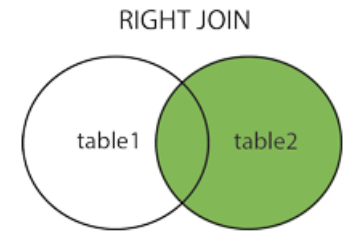


- Muestra TODOS los registros de la tabla de la izquierda, y sólo los registros de la tabla de la derecha que cumplan la condición

left : Consulta de selección								
	Persona.Id	Persona.Nombre	Persona.Edad	IdMascota	Animal.Id	Especie	Animal.Nombre	Animal.Edad
	1	Juan Pérez	34	3	↔3	Cobaya	Baya	1
	2	Sergio García	43					
	3	Andrés López	15	2	↔2	Conejo	Nejo	4
	4	Susana López	13	6	↔6	Hurón	Juanito	3
▶	5	Julio Ruiz	50					
	6	Inés Sánchez	45	5	↔5	Perro	Baxter	10

# Right Join

```
SELECT *
FROM Persona RIGHT JOIN Animal
ON Persona.IdMascota=Animal.Id;
```



- Muestra TODOS los registros de la tabla de la derecha, y sólo los registros de la tabla de la izquierda que cumplan la condición

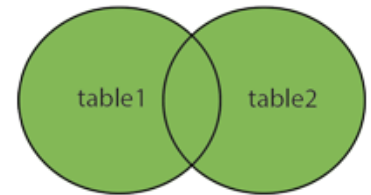
right : Consulta de selección

	Persona.Id	Persona.Nombr	Persona.Edad	IdMascota		Animal.Id	Especie	Animal.Nombre	Animal.Edad
▶						1	Perro	Kity	6
	3	Andrés López	15	2	↔	2	Conejo	Nejo	4
	1	Juan Pérez	34	3	↔	3	Cobaya	Baya	1
						4	Tortuga	Sara	15
	6	Inés Sánchez	45	5	↔	5	Perro	Baxter	10
	4	Susana López	13	6	↔	6	Hurón	Juanito	3
						7	Gato	Misi	3

# Otros tipos de JOIN

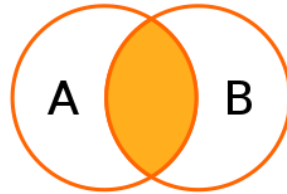
- Existen más combinaciones de JOIN. Una de ellas es FULL JOIN, a veces llamada FULL OUTER JOIN, que obtiene la combinación de LEFT y RIGHT JOIN
- En MySQL no está disponible.
  - Alternativa: combinar ambas con UNION

FULL OUTER JOIN

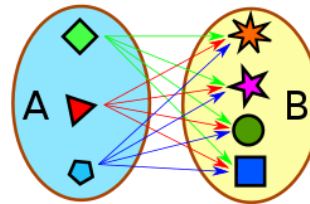


```
SELECT *  
FROM Persona LEFT JOIN Animal  
ON Persona.IdMascota=Animal.Id  
UNION  
SELECT *  
FROM Persona RIGHT JOIN Animal  
ON Persona.IdMascota=Animal.Id;
```

Select (cam pos)  
 From A Inner Join B  
 On A.Clave = B.Clave

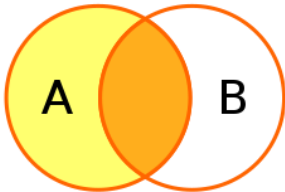


Select (cam pos)  
 From A Cross Join B

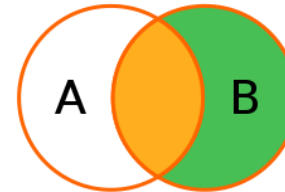


# Joins del SQL

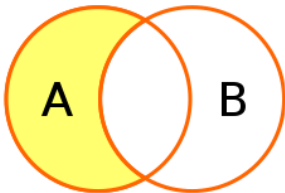
Select (cam pos)  
 From A Left Join B  
 On A.Clave = B.Clave



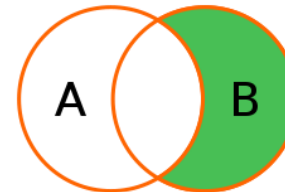
Select (cam pos)  
 From A Right Join B  
 On A.Clave = B.Clave



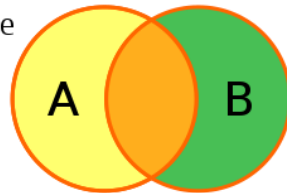
Select (cam pos)  
 From A Left Join B  
 On A.Clave = B.Clave  
 Where B.Clave is Null



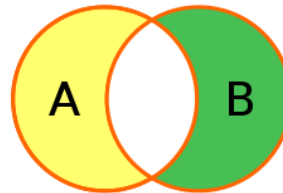
Select (cam pos)  
 From A Right Join B  
 On A.Clave = B.Clave  
 Where A.Clave is Null



Select (cam pos)  
 From A Full Join B  
 On A.Clave = B.Clave



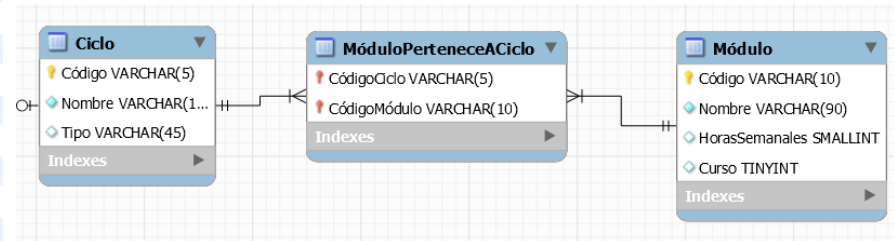
Select (cam pos)  
 From A Right Join B  
 On A.Clave = B.Clave  
 Where (A.Clave is Null) Or (B.Clave is Null)



# JOIN con más de dos tablas

- Recuerda esta consulta :
  - SELECT ciclo.Nombre, módulo.Nombre  
FROM ciclo, pertenece, módulo  
WHERE ciclo.Código=pertenece.CódigoCiclo AND pertenece.CódigoMódulo=módulo.Código  
ORDER BY ciclo.Nombre

Nombre	Nombre
Administración de Sistemas Informáti...	Administración de sistemas gestores de bases de datos
Administración de Sistemas Informáti...	Administración de Sistemas Operativos
Administración de Sistemas Informáti...	Empresa e Iniciativa Emprendedora
Administración de Sistemas Informáti...	Formación en Centros de Trabajo
Administración de Sistemas Informáti...	Fundamentos de hardware
Administración de Sistemas Informáti...	Formación y Orientación Laboral
Administración de Sistemas Informáti...	Gestión de Bases de Datos
Administración de Sistemas Informáti...	Implantación de Aplicaciones Web
Administración de Sistemas Informáti...	Implantación de Sistemas Operativos
Administración de Sistemas Informáti...	Inglés Técnico
Administración de Sistemas Informáti...	Lenguajes de Marcas y Sistemas de Gestión de Inform...
Administración de Sistemas Informáti...	Planificación y Administración de Redes
Administración de Sistemas Informáti...	Proyecto
Administración de Sistemas Informáti...	Seguridad y Alta Disponibilidad
Administración de Sistemas Informáti...	Servicios de Red e Internet
Desarrollo de Aplicaciones Multiplatafo...	Acceso a Datos
Desarrollo de Aplicaciones Multiplatafo...	Bases de Datos
Desarrollo de Aplicaciones Multiplatafo...	Desarrollo de Interfaces
Desarrollo de Aplicaciones Multiplatafo...	Entornos de Desarrollo
Desarrollo de Aplicaciones Multiplatafo...	Empresa e Iniciativa Emprendedora



# JOIN con más de dos tablas

---

- Usando producto cartesiano con condición:

- SELECT ciclo.Nombre, módulo.Nombre  
FROM ciclo, pertenece, módulo  
WHERE ciclo.Código=pertenece.CódigoCiclo  
AND pertenece.CódigoMódulo=módulo.Código  
ORDER BY ciclo.Nombre

- Usando INNER JOIN:

- SELECT ciclo.Nombre, módulo.Nombre  
FROM ciclo INNER JOIN pertenece ON ciclo.Código=pertenece.CódigoCiclo  
INNER JOIN módulo ON pertenece.CódigoMódulo=módulo.Código  
ORDER BY ciclo.Nombre