

SQL - SELECT

- Hemos visto cómo obtener datos de una tabla utilizando los comandos `SELECT` y `WHERE`.
- Con frecuencia, necesitamos mostrar el resultado en un orden particular. Esto podría ser en orden ascendente, en orden descendente, o podría basarse en valores numéricos o de texto. En tales casos, podemos utilizar el comando `ORDER BY`.


SQL - SELECT

- Hemos visto cómo obtener datos de una tabla utilizando los comandos SELECT y WHERE.
- Con frecuencia, necesitamos mostrar el resultado en un orden particular. Esto podría ser en orden ascendente, en orden descendente, o podría basarse en valores numéricos o de texto. En tales casos, podemos utilizar el comando ORDER BY.
- La cláusula ORDER BY ASC significa que los resultados se mostrarán en orden ascendente, y DESC significa que los resultados se mostrarán en orden descendente. Si no se especifica ninguno, la configuración predeterminada es ASC.
- Es posible ordenar por más de una columna.

SQL - SELECT - Ordenamiento

- Muestre el primer apellido, segundo apellido y nombres de todos los empleados ordenados de forma alfabética según el primer apellido.

```
1 • USE empresa;  
2 • SELECT PrimerApellido 'Primer Apellido', SegundoApellido 'Segundo Apellido', Nombres  
3 FROM empleado  
4 ORDER BY PrimerApellido
```

Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: <input type="checkbox"/>		
Primer Apellido	Segundo Apellido	Nombres
Acco	Olvarez	Daniela
Jimenez	Portugal	Alice
Lima	Maldonado	Pedro
Linares	Gomez	Francisco
Perez	Rodriguez	Juan
Santos	Ferrel	Luisa
Vela	Marruecos	Mateo
Velazquez	Flores	Frank

SQL - SELECT - Yunción

- En SQL la yunción es representada por el comando JOIN el cual se utiliza para combinar filas de dos o más tablas, en función a una columna relacionada entre ella.
- Tenemos diferentes tipos:
 - Cross Join (FULL JOIN)
 - Inner Join (Más utilizado)
 - Left Join
 - Right Join

SQL - SELECT - Yunción

- Para demostrar el funcionamiento de los métodos o comandos de yunción (joins) vamos a crear dos tablas entre las cuales se debe realizar algún relacionamiento para "cruzar" datos.
- Veremos algunos ejemplos.

SQL - SELECT - Cross Join

- 1 • **USE** empresa;
- 2 • **SELECT** * **FROM** departamento
- 3 **CROSS JOIN**
- 4 proyecto;

Result Grid								
		Filter Rows:		Export:		Wrap Cell Content:		
	CodDepto	NombreDepto	CodGerente	FechaInicioGerencia	CodProyecto	NombreProyecto	Localizacion	CodDepto
	5	Investigacion	333445555	1998-05-22	1	Proyecto X	Oficina 317	5
	4	Administracion	987654321	1995-01-01	1	Proyecto X	Oficina 317	5
	1	Direccion	888665555	2001-06-19	1	Proyecto X	Oficina 317	5
	5	Investigacion	333445555	1998-05-22	2	Proyecto Y	Oficina 22	5
	4	Administracion	987654321	1995-01-01	2	Proyecto Y	Oficina 22	5
	1	Direccion	888665555	2001-06-19	2	Proyecto Y	Oficina 22	5
	5	Investigacion	333445555	1998-05-22	3	Proyecto Z	Oficina 544	5
	4	Administracion	987654321	1995-01-01	3	Proyecto Z	Oficina 544	5
	1	Direccion	888665555	2001-06-19	3	Proyecto Z	Oficina 544	5
	5	Investigacion	333445555	1998-05-22	10	Proyecto A	Oficina 51	4
	4	Administracion	987654321	1995-01-01	10	Proyecto A	Oficina 51	4
	1	Direccion	888665555	2001-06-19	10	Proyecto A	Oficina 51	4
	5	Investigacion	333445555	1998-05-22	20	Proyecto B	Oficina 67	1
	4	Administracion	987654321	1995-01-01	20	Proyecto B	Oficina 67	1
	1	Direccion	888665555	2001-06-19	20	Proyecto B	Oficina 67	1
	5	Investigacion	333445555	1998-05-22	30	Proyecto C	Oficina 385	4
	4	Administracion	987654321	1995-01-01	30	Proyecto C	Oficina 385	4
	1	Direccion	888665555	2001-06-19	30	Proyecto C	Oficina 385	4

SQL - SELECT - INNER JOIN

- El comando Inner Join (Yunción Interna) realiza la yunción de tablas basándose en un condición de yunción (o punto en común).
- Veremos varios ejemplos de este comando.

SQL - SELECT - INNER JOIN

- 1 • **USE** empresa;
- 2 • **SELECT** * **FROM** departamento d
- 3 **INNER JOIN** proyecto p
- 4 **ON** d.CodDepto = p.CodDepto;

Result Grid							
		Filter Rows:		Exports:	Wrap Cell Content:		
CodDepto	NombreDepto	CodGerente	FechaInicioGerencia	CodProyecto	NombreProyecto	Localizacion	CodDepto
1	Direccion	888665555	2001-06-19	20	Proyecto B	Oficina 67	1
4	Administracion	987654321	1995-01-01	10	Proyecto A	Oficina 51	4
4	Administracion	987654321	1995-01-01	30	Proyecto C	Oficina 385	4
5	Investigacion	333445555	1998-05-22	1	Proyecto X	Oficina 317	5
5	Investigacion	333445555	1998-05-22	2	Proyecto Y	Oficina 22	5
5	Investigacion	333445555	1998-05-22	3	Proyecto Z	Oficina 544	5

SQL - SELECT - INNER JOIN

- Muestre, para todos los proyectos, su código, nombre y departamento al cual pertenece.

- 1 • **USE** empresa;
- 2 • **SELECT** p.CodProyecto, p.NombreProyecto, d.NombreDepto **FROM** departamento d
- 3 **INNER JOIN** proyecto p
- 4 **ON** d.CodDepto = p.CodDepto;

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
CodProyecto	NombreProyecto	NombreDepto			
20	Proyecto B	Direccion			
10	Proyecto A	Administracion			
30	Proyecto C	Administracion			
1	Proyecto X	Investigacion			
2	Proyecto Y	Investigacion			
3	Proyecto Z	Investigacion			

SQL - SELECT - INNER JOIN

- Muestre para todas las empleadas mujeres, su código, primer apellido, segundo apellido y nombres, así como el departamento al cual pertenecen. Ordenar el resultado por el primer apellido.

```
1 • USE empresa;  
2 • SELECT e.CodEmpleado 'Código',  
3     e.PrimerApellido 'Primer Apellido',  
4     e.SegundoApellido 'Segundo Apellido',  
5     e.Nombres,  
6     d.NombreDepto Departamento  
7 FROM empleado e  
8 INNER JOIN departamento d  
9 ON e.CodDepto = d.CodDepto  
10 WHERE e.Sexo = 'F'  
11 ORDER BY e.PrimerApellido
```

Result Grid				
Filter Rows:				
Export:				
Wrap Cell Content:				
Código	Primer Apellido	Segundo Apellido	Nombres	Departamento
453453453	Acco	Olvarez	Daniela	Investigacion
999887777	Jimenez	Portugal	Alice	Administracion
987654321	Santos	Ferrel	Luisa	Administracion

SQL - SELECT - INNER JOIN

- Muestre el código, primer apellido, segundo apellido y nombres, de todos los empleados hombres, así como cuántas hora dedica, al Proyecto Z (Código 3). También mostrar el departamento al cual pertenece el proyecto y ordenar el resultado por el primer apellido.

SQL - SELECT - INNER JOIN

```

1 • USE empresa;
2 • SELECT e.CodEmpleado 'Código',
3     e.PrimerApellido 'Primer Apellido',
4     e.SegundoApellido 'Segundo Apellido',
5     e.Nombres,
6     p.NombreProyecto Proyecto, te.Horas,
7     d.NombreDepto
8 FROM empleado e
9 INNER JOIN trabaja_en te
10 INNER JOIN departamento d
11 INNER JOIN proyecto p
12 ON e.CodEmpleado = te.CodEmpleado
13 AND p.CodProyecto = te.CodProyecto
14 AND p.CodDepto = d.CodDepto
15 WHERE e.Sexo = 'M' AND p.CodProyecto = 3
16 ORDER BY e.PrimerApellido

```

Result Grid						
Filter Rows:		Export:		Wrap Cell Content:		
Código	Primer Apellido	Segundo Apellido	Nombres	Proyecto	Horas	NombreDepto
666884444	Lima	Maldonado	Pedro	Proyecto Z	40	Investigacion
333445555	Velazquez	Flores	Frank	Proyecto Z	10	Investigacion