

CURSO PROGRAMACIÓN WEB FULL STACK

NIVEL 2

Módulo 4 - Bases de datos

Ejercicio 1

-- 1. Lista el apellido de todos los empleados.

```
select apellido from empleados;
```

-- 2. Lista el apellido de los empleados eliminando los apellidos que estén repetidos.

```
select distinctrow apellido from empleados;
```

-- 3. Lista todas las columnas de la tabla empleados.

```
select * from empleados;
```

-- 4. Lista el nombre y apellido de todos los empleados.

```
select nombre,apellido from empleados;
```

-- 5. Lista el cuit/cuil de los empleados que aparecen en la tabla empleados.

```
select cuil_cuit from empleados;
```

-- 6. Lista el nombre y apellido de los empleados en una única columna.

```
select concat_ws(' ', nombre, apellido) as Empleado from empleados;
```

-- 7. Lista el nombre y apellido de los empleados en una única columna, convirtiendo todos los caracteres en mayúscula.

```
select upper(concat_ws(' ', nombre, apellido)) as Empleado from empleados;
```

-- 8. Lista el nombre y apellido de los empleados en una única columna, convirtiendo todos los caracteres en minúscula.

```
select lower(concat_ws(' ', nombre, apellido)) as Empleado from empleados;
```

-- 9. Lista el nombre de los departamentos y el valor del presupuesto actual ordenado de forma ascendente.

```
select nombre, presupuesto from departamentos order by presupuesto asc;
```

-- 10. Lista el nombre de todos los departamentos ordenados de forma ascendente.

```
select nombre from departamentos order by nombre asc;
```

-- 11. Lista el nombre de todos los departamentos ordenados de forma descendente.

```
select nombre from departamentos order by nombre desc;
```

-- 12. Lista el apellido y el nombre de todos los empleados, ordenados de forma alfabética teniendo en cuenta en primer lugar su apellido y luego su nombre.

select apellido, nombre from empleados order by apellido, nombre asc;

-- 13. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen mayor presupuesto.

select nombre, presupuesto from departamentos order by presupuesto desc limit 3;

-- 14. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen menor presupuesto.

select nombre, presupuesto from departamentos order by presupuesto asc limit 3;

-- 15. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto mayor o igual a \$150000.

select nombre, presupuesto from departamentos where presupuesto >= 150000;

-- 16. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto entre \$100000 y \$200000. Sin utilizar el operador BETWEEN.

select nombre, presupuesto from departamentos where presupuesto > 100000 and presupuesto < 200000;

-- 17. Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre \$100000 y \$200000. Sin utilizar el operador BETWEEN.

select nombre, presupuesto from departamentos where presupuesto < 100000 or presupuesto > 200000;

-- 18. Devuelve una lista con el nombre de los departamentos que tienen un presupuesto entre \$100000 y \$200000. Utilizando el operador BETWEEN.

select nombre, presupuesto from departamentos where presupuesto between 100000 and 200000;

-- 19. Devuelve una lista con el nombre de los departamentos que no tienen un presupuesto entre \$100000 y \$200000. Utilizando el operador BETWEEN.

select nombre, presupuesto from departamentos where presupuesto not between 100000 and 200000;

-- 20. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno.

select e.nombre, e.apellido, d.nombre from sistema.empleados e join sistema.departamentos d where e.id_departamento = d.iddepartamento;

-- 21. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno. Ordena el resultado, en primer lugar por el nombre del departamento (en orden alfabético) y en segundo lugar por apellido y el nombre de los empleados.

select e.nombre, e.apellido, d.nombre from sistema.empleados e join sistema.departamentos d where e.id_departamento = d.iddepartamento order by d.nombre, e.apellido, e.nombre asc;

-- 22. Devuelve un listado con el código y el nombre del departamento, solamente de aquellos departamentos que tienen empleados.

```
select distinctrow d.iddepartamento, d.nombre from sistema.departamentos d join  
sistema.empleados e where d.iddepartamento = e.id_departamento;
```

-- 23. Devuelve el nombre del departamento donde trabaja el empleado que tiene el cuit 27-38382980-3.

```
select e.cuil_cuit, d.nombre from sistema.empleados e join sistema.departamentos d where  
e.id_departamento = d.iddepartamento and e.cuil_cuit= '27-38382980-3';
```

-- 24. Devuelve el nombre del departamento donde trabaja el empleado Pepe Ruiz.

```
select concat_ws(' ', e.nombre, e.apellido) as Empleado, d.nombre as Departamento from  
sistema.empleados e join sistema.departamentos d where e.id_departamento =  
d.iddepartamento and e.nombre= 'Pepe' and e.apellido= 'Ruiz';
```

-- 25. Devuelve un listado con los datos de los empleados que trabajan en el departamento de I+D. Ordena el resultado alfabéticamente.

```
select * from sistema.empleados inner join sistema.departamentos on  
sistema.empleados.id_departamento = sistema.departamentos.iddepartamento where  
sistema.departamentos.nombre= 'I+D' order by sistema.empleados.apellido,  
sistema.empleados.nombre asc;
```

-- 26. Devuelve un listado con los datos de los empleados que trabajan en el departamento de Sistemas, Contabilidad o I+D. Ordena el resultado alfabéticamente.

```
select * from sistema.empleados inner join sistema.departamentos on  
sistema.empleados.id_departamento = sistema.departamentos.iddepartamento where  
sistema.departamentos.nombre = 'Sistemas' or sistema.departamentos.nombre =  
'Contabilidad' or sistema.departamentos.nombre = 'I+D' order by sistema.empleados.apellido,  
sistema.empleados.nombre asc;
```

-- 27. Devuelve una lista con el nombre de los empleados que tienen los departamentos que no tienen un presupuesto entre \$100000 y \$200000.

```
select * from sistema.empleados inner join sistema.departamentos on  
sistema.empleados.id_departamento = sistema.departamentos.iddepartamento where  
sistema.departamentos.presupuesto not between 100000 and 200000;
```

Ejercicio 2

-- Ejercicio 2

-- 1.Devuelve un listado con todos los pedidos que se han realizado. Los pedidos deben estar ordenados por la fecha de realización, mostrando en primer lugar los pedidos más recientes.

```
select * from ventas.pedidos order by ventas.pedidos.fecha desc;
```

-- 2.Devuelve todos los datos de los dos pedidos de mayor valor.

```
select * from ventas.pedidos order by ventas.pedidos.cantidad desc limit 2;
```

-- 3.Devuelve un listado con los identificadores de los clientes que han realizado algún pedido. Tenga en cuenta que no debe mostrar identificadores que estén repetidos.

```
select distinctrow ventas.clientes.id from ventas.clientes inner join ventas.pedidos on
ventas.clientes.id = ventas.pedidos.id_cliente;
```

-- 4.Devuelve un listado de todos los pedidos que se realizaron durante el año 2022, cuya cantidad total sea superior a \$500.

```
select * from ventas.pedidos where year(ventas.pedidos.fecha) = 2022 and
ventas.pedidos.cantidad > 500;
```

-- 5.Devuelve un listado con el nombre y apellido de los vendedores que tienen una comisión entre 0.05 y 0.11.

```
select ventas.vendedores.nombre, ventas.vendedores.apellido from ventas.vendedores
where ventas.vendedores.comisión between 0.05 and 0.11;
```

-- 6.Devuelve el valor de la comisión de mayor valor que existe en la tabla vendedores.

```
select max(ventas.vendedores.comisión) from ventas.vendedores;
```

-- 7.Devuelve el identificador, nombre y primer apellido de aquellos clientes cuyo cuitcuil no es NULL. El listado deberá estar ordenado alfabéticamente por apellido y nombre.

```
select ventas.clientes.id, ventas.clientes.nombre, ventas.clientes.apellido from ventas.clientes
where ventas.clientes.cuitcuil IS NOT NULL order by ventas.clientes.apellido,
ventas.clientes.nombre;
```

-- 8.Devuelve un listado de los nombres de los clientes que empiezan por "A" y terminan por "n" y también los nombres que empiezan por "P". El listado deberá estar ordenado alfabéticamente.

```
select ventas.clientes.nombre from ventas.clientes where ventas.clientes.nombre like
'A%n' or ventas.clientes.nombre like 'P%' order by ventas.clientes.nombre asc;
```

-- 9.Devuelve un listado de los nombres de los clientes que no empiezan por "A". El listado deberá estar ordenado alfabéticamente.

```
select ventas.clientes.nombre from ventas.clientes where ventas.clientes.nombre not like 'A%'
order by ventas.clientes.nombre asc;
```

-- 10.Devuelve un listado con los nombres de los vendedores que terminan por "el" o "o". Tenga en cuenta que se deberán eliminar los nombres repetidos.

```
select distinctrow ventas.clientes.nombre from ventas.clientes where ventas.clientes.nombre
like '%el' or ventas.clientes.nombre like '%o' order by ventas.clientes.nombre asc;
```

-- 11.Devuelve un listado con el identificador, nombre y apellido de todos los clientes que han realizado algún pedido. El listado debe estar ordenado alfabéticamente y se deben eliminarlos elementos repetidos.

```
select distinctrow ventas.clientes.id, ventas.clientes.nombre, ventas.clientes.apellido from
ventas.clientes inner join ventas.pedidos on ventas.clientes.id = ventas.pedidos.id_cliente
order by ventas.clientes.nombre, ventas.clientes.apellido asc;
```

-- 12.Devuelve un listado que muestre todos los pedidos que ha realizado cada cliente. El resultado debe mostrar todos los datos de los pedidos y del cliente. El listado debe mostrar los datos de los clientes ordenados alfabéticamente.

```
select distinctrow * from ventas.clientes inner join ventas.pedidos on ventas.clientes.id =  
ventas.pedidos.id_cliente order by ventas.clientes.nombre, ventas.clientes.apellido asc;
```

-- 13.Devuelve un listado que muestre todos los pedidos en los que ha participado un vendedor. El resultado debe mostrar todos los datos de los pedidos y de los vendedores. El listado debe mostrar los datos de los vendedores ordenados alfabéticamente.

```
select * from ventas.vendedores inner join ventas.pedidos on ventas.vendedores.id =  
ventas.pedidos.id_vendedor order by ventas.vendedores.nombre, ventas.vendedores.apellido  
asc;
```

-- 14.Devuelve un listado que muestre todos los clientes, con todos los pedidos que han realizado y con los datos de los vendedores asociados a cada pedido.

```
select ventas.c.nombre, ventas.c.apellido, ventas.p.*, ventas.vend.* from ventas.clientes c  
INNER JOIN ventas.pedidos p ON c.id = p.id_cliente INNER JOIN ventas.vendedores vend ON  
vend.id = p.id_vendedor;
```

-- 15.Devuelve un listado de todos los clientes que realizaron un pedido durante el año 2022, cuyo monto esté entre \$300 y \$1000.

```
select ventas.clientes.id, ventas.clientes.nombre, ventas.clientes.apellido from ventas.clientes  
inner join ventas.pedidos on ventas.clientes.id = ventas.pedidos.id_cliente where  
year(ventas.pedidos.fecha) = 2022 and ventas.pedidos.cantidad between 300 and 1000;
```

-- 16.Devuelve el nombre y apellido de todos los vendedores que han participado en algún pedido realizado por María Santana.

```
select ventas.vend.nombre, ventas.vend.apellido from ventas.vendedores vend inner join  
ventas.pedidos p on vend.id = p.id_vendedor inner join ventas.clientes c on c.id = p.id_cliente  
where id_cliente= 6;
```

-- 17.Devuelve el nombre de todos los clientes que han realizado algún pedido con el vendedor Daniel Sáez.

```
select ventas.c.nombre from ventas.clientes c inner join ventas.pedidos p on c.id =  
p.id_cliente inner join ventas.vendedores vend on vend.id = p.id_vendedor where  
id_vendedor= 1;
```

-- 18.Devuelve un listado con todos los clientes junto con los datos de los pedidos que han realizado. Este listado también debe incluir los clientes que no han realizado ningún pedido. El listado debe estar ordenado alfabéticamente por el apellido y nombre de los clientes.

```
select ventas.c.*, ventas.p.* from ventas.clientes c left join ventas.pedidos p on c.id =  
p.id_cliente order by c.apellido, c.nombre asc;
```

-- 19.Devuelve un listado con todos los vendedores junto con los datos de los pedidos que han realizado. Este listado también debe incluir los vendedores que no han realizado ningún pedido. El listado debe estar ordenado alfabéticamente por el apellido y nombre de los vendedores.

```
select ventas.vend.*, ventas.p.* from ventas.vendedores vend left join ventas.pedidos p on  
vend.id = p.id_vendedor order by vend.apellido, vend.nombre asc;
```

-- 20.Devuelve un listado que solamente muestre los clientes que no han realizado ningún pedido.

```
select ventas.c.* from ventas.clientes c left join ventas.pedidos p on c.id = p.id_cliente where p.id_cliente IS NULL;
```

-- 21.Devuelve un listado que solamente muestre los vendedores que no han realizado ningún pedido.

```
select ventas.vend.* from ventas.vendedores vend left join ventas.pedidos p on vend.id = p.id_vendedor where p.id_vendedor IS NULL;
```

-- 22.Devuelve un listado con los clientes que no han realizado ningún pedido y de los vendedores que no han participado en ningún pedido. Ordene el listado alfabéticamente por el apellido y el nombre. En el listado deberá diferenciar de algún modo los clientes y los vendedores.

```
select ventas.vendedores.nombre, ventas.vendedores.apellido, ' ' as cliente_id, vendedores.id as vendedor_id from ventas.pedidos right join ventas.vendedores on ventas.pedidos.id_vendedor = ventas.vendedores.id where ventas.pedidos.id_vendedor is null union select clientes.nombre, clientes.apellido, clientes.id as cliente_id, ' ' from ventas.pedidos right join ventas.clientes on ventas.pedidos.id_cliente = ventas.clientes.id where ventas.pedidos.id_cliente is null order by apellido, nombre asc;
```

-- 23.Calcula la cantidad total que suman todos los pedidos que aparecen en la tabla pedido.

```
select sum(pedidos.cantidad) as 'Suma Total' from ventas.pedidos;
```

-- 24.Calcula la cantidad media de todos los pedidos que aparecen en la tabla pedido.

```
select avg(pedidos.cantidad) as 'Suma Media' from ventas.pedidos;
```

-- 25.Calcula el número total de vendedores distintos que aparecen en la tabla pedido.

```
select count(ventas.vendedores.nombre) as 'Total de vendedores' from ventas.vendedores;
```

-- 26.Calcula el número total de clientes que aparecen en la tabla cliente.

```
select count(ventas.clientes.nombre) as 'Total de clientes' from ventas.clientes;
```

-- 27.Calcula cuál es la mayor cantidad que aparece en la tabla pedido.

```
select max(ventas.pedidos.cantidad) as 'Maximo Pedidos' from ventas.pedidos;
```

-- 28.Calcula cuál es la menor cantidad que aparece en la tabla pedido.

```
select min(ventas.pedidos.cantidad) as 'Minimo Pedidos' from ventas.pedidos;
```

-- 29.Calcula cuál es el valor máximo de categoría para cada una de las ciudades que aparece en la tabla cliente.

```
select ventas.clientes.ciudad, max(ventas.clientes.categoría) from ventas.clientes group by ventas.clientes.ciudad;
```

-- 30.Calcula cuál es el máximo valor de los pedidos realizados durante el mismo día para cada uno de los clientes. Es decir, el mismo cliente puede haber realizado varios pedidos de diferentes cantidades el mismo día. Se pide que se calcule cuál es el pedido de máximo valor

para cada uno de los días en los que un cliente ha realizado un pedido. Muestra el identificador del cliente, nombre, apellido, la fecha y el valor de la cantidad.

```
select ventas.clientes.id, ventas.clientes.nombre, ventas.clientes.apellido,  
ventas.pedidos.fecha, max(ventas.pedidos.cantidad) as 'maximo pedidos' from ventas.clientes  
left join ventas.pedidos on ventas.clientes.id = ventas.pedidos.id_cliente group by  
ventas.pedidos.fecha;
```

-- 31. Calcula cuál es el máximo valor de los pedidos realizados durante el mismo día para cada uno de los clientes, teniendo en cuenta que sólo queremos mostrar aquellos pedidos que superen la cantidad de \$2000.

```
select ventas.clientes.id, ventas.clientes.nombre, ventas.clientes.apellido,  
ventas.pedidos.fecha, max(ventas.pedidos.cantidad) as 'maximo pedidos' from ventas.clientes  
left join ventas.pedidos on ventas.clientes.id = ventas.pedidos.id_cliente where  
ventas.pedidos.cantidad > 2000 group by ventas.pedidos.fecha;
```

-- 32. Calcula el máximo valor de los pedidos realizados para cada uno de los vendedores durante la fecha 2021-08-17. Muestra el identificador del vendedor, nombre, apellido y total.

```
select vend.id, vend.nombre, vend.apellido, max(p.cantidad) as 'Total', ventas.p.fecha from  
ventas.vendedores vend left join ventas.pedidos p on vend.id = p.id_vendedor where p.fecha  
between '2021-08-17' and '2021-08-17' group by ventas.vend.id;
```

-- 33. Devuelve un listado con el identificador de cliente, nombre y apellido y el número total de pedidos que ha realizado cada uno de los clientes. Tenga en cuenta que pueden existir clientes que no han realizado ningún pedido. Estos clientes también deben aparecer en el listado indicando que el número de pedidos realizados es 0.

```
select ventas.clientes.id, ventas.clientes.nombre, ventas.clientes.apellido,  
count(ventas.pedidos.cantidad) as 'Total Pedidos' from ventas.clientes left join ventas.pedidos  
on ventas.clientes.id = ventas.pedidos.id_cliente group by ventas.clientes.id;
```

-- 34. Devuelve un listado con el identificador de cliente, nombre, apellido y el número total de pedidos que ha realizado cada uno de los clientes durante el año 2020.

```
select ventas.clientes.id, ventas.clientes.nombre, ventas.clientes.apellido,  
count(ventas.pedidos.cantidad) as 'Total Pedidos' from ventas.clientes left join ventas.pedidos  
on ventas.clientes.id = ventas.pedidos.id_cliente where year(ventas.pedidos.fecha) = 2020  
group by ventas.clientes.id;
```

-- 35. Devuelve cuál ha sido el pedido de máximo valor que se ha realizado cada año.

```
select year(ventas.pedidos.fecha) as 'Año', max(ventas.pedidos.cantidad) as 'Valor Maximo'  
from ventas.pedidos group by year(ventas.pedidos.fecha) order by ventas.pedidos.fecha asc;
```

-- 36. Devuelve el número total de pedidos que se han realizado cada año.

```
select year(ventas.pedidos.fecha) as 'Año', count(ventas.pedidos.cantidad) as 'Total Pedidos'  
from ventas.pedidos group by year(ventas.pedidos.fecha) order by ventas.pedidos.fecha asc;
```

-- 37. Devuelve un listado con todos los pedidos que ha realizado Adela Salas. (Sin utilizar INNER JOIN).

```
select * from ventas.pedidos where id_cliente = (select clientes.id from clientes where
clientes.nombre = 'Adela');
```

-- 38.Devuelve el número de pedidos en los que ha participado el vendedor Daniel Sáez. (Sin utilizar INNER JOIN).

```
select count(ventas.pedidos.cantidad) as 'Cantidad Pedidos' from ventas.pedidos where
id_vendedor = (select vendedores.id from vendedores where vendedores.nombre = 'Daniel');
```

-- 39.Devuelve los datos del cliente que realizó el pedido más caro en el año 2020. (Sin utilizar INNER JOIN).

```
select * from ventas.clientes where clientes.id = (select pedidos.id_cliente from pedidos
where year(pedidos.fecha) = 2020 order by pedidos.cantidad desc limit 1);
```

-- 40.Devuelve la fecha y la cantidad del pedido de menor valor realizado por el cliente Pepe Ruiz Santana.

```
select ventas.pedidos.fecha, min(ventas.pedidos.cantidad) as 'Minima Cantidad' from
ventas.pedidos inner join ventas.clientes on pedidos.id_cliente = clientes.id where
ventas.clientes.nombre = 'Pepe';
```

-- 41.Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando IN o NOT IN).

```
select * from ventas.clientes where clientes.id not in (select pedidos.id_cliente from
ventas.pedidos);
```

-- 42.Devuelve un listado de los vendedores que no han realizado ningún pedido. (Utilizando IN o NOT IN).

```
select * from ventas.vendedores where vendedores.id not in (select pedidos.id_vendedor
from ventas.pedidos);
```

-- 43.Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando EXISTS o NOT EXISTS).

```
select * from ventas.clientes where not exists (select pedidos.id_cliente from ventas.pedidos
where pedidos.id_cliente = clientes.id);
```

-- 44.Devuelve un listado de los vendedores que no han realizado ningún pedido. (Utilizando EXISTS o NOT EXISTS).

```
select * from ventas.vendedores where not exists (select pedidos.id_vendedor from
ventas.pedidos where pedidos.id_vendedor = vendedores.id);
```