

Trabajo Practico N 5

Ejercicio N 1

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

```
SELECT empleados.apellido FROM sistema.empleados;
```

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

```
SELECT empleados.apellido FROM sistema.empleados group by empleados.apellido;
```

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

```
Select * from sistema.empleados;
```

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

```
SELECT nombre, apellido FROM sistema.empleados;
```

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

```
SELECT cuil_cuit FROM sistema.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 ucase(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 Lcase(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 d.nombre FROM sistema.departamentos as d order by presupuesto asc;
```

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

```
SELECT d.nombre FROM sistema.departamentos as d order by nombre asc;
```

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

```
SELECT d.nombre FROM sistema.departamentos as d 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 e.nombre as Nombre, e.apellido as Apellido from sistema.empleados as e order by apellido asc;
```

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

```
SELECT d.nombre, d.presupuesto FROM sistema.departamentos as d 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 d.nombre, d.presupuesto FROM sistema.departamentos as d 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 d.nombre, d.presupuesto FROM sistema.departamentos as d where d.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 d.nombre, d.presupuesto FROM sistema.departamentos as d where d.presupuesto >100000 and d.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 d.nombre, d.presupuesto FROM sistema.departamentos as d where d.presupuesto <100000 or d.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 d.nombre, d.presupuesto FROM sistema.departamentos as d where d.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 d.nombre, d.presupuesto FROM sistema.departamentos as d where d.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 as e
inner join sistema.departamentos as 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 as e
inner join sistema.departamentos as d where e.id_departamento = d.iddepartamento order by
d.nombre asc, e.apellido asc;

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

select d.iddepartamento as ID, d.Nombre from sistema.departamentos as d
join sistema.empleados as e where e.id_departamento = d.iddepartamento group by d.nombre;

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

select d.Nombre Departamento, e.Apellido, e.Nombre from sistema.departamentos as d
join sistema.empleados as e where e.id_departamento = d.iddepartamento and e.cuil_cuit like
'27-3838%';

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

select d.Nombre Departamento, e.Apellido, e.Nombre from sistema.departamentos as d
join sistema.empleados as e where e.id_departamento = d.iddepartamento and e.nombre like
'pepe%';

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

select e.idempleado as ID, e.Apellido, e.Nombre, e.Cuil_Cuit, d.nombre as Departamento from
sistema.empleados as e
join sistema.departamentos as d where e.id_departamento = d.iddepartamento and
d.iddepartamento=5 order by e.apellido 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 e.idempleado as ID, e.Apellido, e.Nombre, e.Cuil_Cuit, d.nombre as Departamento from
sistema.empleados as e
join sistema.departamentos as d where e.id_departamento = d.iddepartamento and d.iddepartamento
in ('2', '4', '5') order by e.apellido 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 e.Apellido, e.Nombre, d.nombre as Departamento from sistema.empleados as e
join sistema.departamentos as d where e.id_departamento = d.iddepartamento and d.presupuesto
not between 100000 and 200000 order by e.apellido asc;
```

Ejercicio N 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 fecha desc;

-- 2. Devuelve todos los datos de los dos pedidos de mayor valor.
select * from ventas.pedidos order by 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 id_cliente from ventas.pedidos group by 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 as p where p.cantidad >500 and p.fecha like '2022%';

-- 5. Devuelve un listado con el nombre y apellido de los vendedores que tienen una comisión
entre 0.05 y 0.11.
select concat_ws(' ', nombre, apellido) as Nombre, Comisión from ventas.vendedores as v where
v.comisión between 0.05 and 0.11;

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

```
SELECT Comisión FROM ventas.vendedores order by comisión desc limit 1;
```

-- 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 Id, concat_ws(' ', nombre, apellido) as Nombre_Cliente from ventas.clientes c where c.cuitcuil is not null;
```

-- 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 concat_ws(' ', nombre, apellido) as Nombre_Cliente from ventas.clientes where nombre like 'A%n' or nombre like 'p%' order by nombre asc;
```

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

```
select concat_ws(' ', nombre, apellido) as Nombre_Cliente from ventas.clientes where nombre not like 'A%' order by 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 concat_ws(' ', nombre, apellido) as Nombre_Vendedor from ventas.vendedores where nombre like '%el' or nombre like '%o' group by nombre;
```

-- 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 eliminar los elementos repetidos.

```
select c.Id, concat_ws(' ', nombre, apellido) as Nombre_Cliente from ventas.clientes as c inner join ventas.pedidos as p where c.id=p.id_cliente group by nombre order by nombre 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 * from ventas.clientes as c inner join ventas.pedidos as p where c.id=p.id_cliente order by c.nombre, p.fecha;
```

-- 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 as v
inner join ventas.pedidos as p where v.id=p.id_cliente order by v.nombre, p.fecha;
```

-- 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 * from ventas.pedidos as p
join ventas.vendedores as v, ventas.clientes as c where p.id_vendedor=v.id and
c.id=p.id_cliente order by c.apellido;
```

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

```
select c.Id, concat_ws(' ', nombre, apellido) as Nombre_Cliente, p.cantidad as Monto, p.Fecha
from ventas.clientes as c
join ventas.pedidos as p on c.id=p.id_cliente where p.fecha like '2022%' and p.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 concat_ws(' ', v.nombre, v.apellido) as Nombre_Vendedor from ventas.vendedores as v
join ventas.pedidos as p on p.id_vendedor=v.id, ventas.clientes as c where p.id_cliente=c.id
and c.nombre like 'Mar%' and c.apellido like 'Santana';
```

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

```
select concat_ws(' ', c.nombre, c.apellido) as Nombre_Cliente from ventas.vendedores as v
join ventas.pedidos as p on p.id_vendedor=v.id, ventas.clientes as c where p.id_cliente=c.id
and v.nombre like 'D%' and v.apellido like 'S%' group by c.nombre;
```

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

-- 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 * from ventas.vendedores as v
left join ventas.pedidos as p on v.id=p.id_vendedor order by v.apellido, v.nombre;
```

```
-- 20.Devuelve un listado que solamente muestre los clientes que no han realizado ningún
pedido.
select c.Id as Id_Cliente, c.Nombre, c.Apellido from ventas.pedidos as p
right join ventas.clientes as c 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 v.Id as Id_Vendedor, v.Nombre, v.Apellido from ventas.pedidos as p
right join ventas.vendedores as v on v.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 c.Id as Id_Cliente, ' ' as Id_Vendedor, c.Nombre, c.Apellido from ventas.pedidos as p
right join ventas.clientes as c on c.id=p.id_cliente where p.id_cliente is null
union
select ' ' as Id_Cliente, v.Id as Id_Vendedor, v.Nombre, v.Apellido from ventas.pedidos as p
right join ventas.vendedores as v on v.id=p.id_vendedor where p.id_vendedor is null;
```

```
-- 23.Calcula la cantidad total que suman todos los pedidos que aparecen en la tabla pedido.
select sum(cantidad) from ventas.pedidos;
```

```
-- 24.Calcula la cantidad media de todos los pedidos que aparecen en la tabla pedido.
select avg(cantidad) from ventas.pedidos;
```

```
-- 25.Calcula el número total de vendedores distintos que aparecen en la tabla pedido.
select count(distinct id_vendedor) as Vendedores from ventas.pedidos;
```

```
-- 26.Calcula el número total de clientes que aparecen en la tabla cliente.
select count(id) Clientes from ventas.clientes;
```

```
-- 27.Calcula cuál es la mayor cantidad que aparece en la tabla pedido.
select max(cantidad) Max_Valor from ventas.pedidos;
```

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

```
select min(cantidad) Max_Valor 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 distinct Ciudad, max(Categoría) from ventas.clientes group by ciudad order by 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 c.Id, concat_ws(' ', c.Nombre, c.Apellido) as Nombre, p.Fecha, max(p.Cantidad)
```

```
Mayor_Valor from ventas.clientes as c
```

```
join ventas.pedidos as p on c.id=p.id_cliente group by id_cliente, p.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 c.Id, concat_ws(' ', c.Nombre, c.Apellido) Nombre, p.Fecha, max(p.Cantidad) Mayor_valor
```

```
from ventas.clientes as c
```

```
join ventas.pedidos as p on c.id=p.id_cliente where p.cantidad >2000 group by id_cliente,
```

```
p.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 v.Id, concat_ws(' ', v.Nombre, v.Apellido) Nombre_Vendedor, p.Fecha, max(p.Cantidad)
```

```
Mayor_valor from ventas.vendedores as v
```

```
join ventas.pedidos as p on v.id=p.id_vendedor where p.fecha like '2021-08-17' group by
```

```
id_vendedor;
```

-- 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 c.Id, concat_ws(' ', c.Nombre, c.Apellido) Nombre_Cliente, count(p.id) Cantidad_Pedidos
from ventas.clientes c
left join ventas.pedidos p on c.id=p.id_cliente group by c.id order by Cantidad_Pedidos;
```

-- 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 c.Id, concat_ws(' ', c.Nombre, c.Apellido) Nombre_Cliente, count(p.id) Cantidad_Pedidos
from ventas.clientes c
left join ventas.pedidos p on c.id=p.id_cliente where p.fecha like '2020%' group by c.id;
```

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

```
select distinct year(p.fecha) Año, max(cantidad) Mayor_Valor from ventas.pedidos p group by año
order by cantidad;
```

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

```
select distinct year(p.fecha) Año, count(id) Pedidos from ventas.pedidos p group by año order
by cantidad;
```

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

```
select concat_ws(' ', c.Nombre, c.Apellido) Nombre, p.Fecha, p.Id Pedido, p.Cantidad from
ventas.clientes c
left join ventas.pedidos p on c.id=p.id_cliente where nombre like 'adela';
```

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

```
select count(p.id) Pedidos, concat_ws(' ', v.Nombre, v.Apellido) Nombre from ventas.pedidos p
left join ventas.vendedores v on v.id=p.id_vendedor where nombre like 'Daniel';
```

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

```
select c.Id, concat_ws(' ', c.apellido, c.nombre) Cliente, max(p.cantidad) Monto from
ventas.clientes as c
right join ventas.pedidos p on c.id=p.id_cliente where p.fecha like '2020%';
```

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

```
select c.Id, concat_ws(' ', c.apellido, c.nombre) Cliente, min(p.cantidad) Monto from
ventas.clientes as c
right join ventas.pedidos p on c.id=p.id_cliente where nombre like 'pepe';
```

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

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

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

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