

Enunciados SQL Cooperativa

INNER JOIN

Enunciado

Listar productos que tengan algún material con stock menor a la cantidad utilizada para componer dicho producto. Indicar código y nombre del producto y de él/los materiales con bajo stock, código, nombre, unidad de medida, stock y cantidad que componen. Ordenar por nombre de material alfabeticamente y stock en forma descendente.

Resolución

```
select prd.codigo, prd.nombre
       , mat.codigo, mat.nombre, mat.unidad_medida, mat.stock
       , comp.cantidad
from producto prd
inner join composicion comp
      on prd.codigo = comp.codigo_producto
inner join material mat
      on comp.codigo_material=mat.codigo
where mat.stock < comp.cantidad
order by mat.nombre, mat.stock;
```

LEFT/RIGHT JOIN

Enunciado

Listar los miembros y, si lo hubiera hecho, los lotes donde participó en la producción hace 2 meses atrás (julio). Indicando: cuil, nombre y apellido del miembro y de los lotes producidos, código de producto, número de lote, fecha de producción y cantidad de horas.

Resolución

```
select mi.cuil, mi.nombre, mi.apellido
       , lote.codigo_producto, lote.numero, lote.fecha_produccion,
       lote.cantidad_producida
from miembro mi
left join produce pro
      on mi.cuil=pro.cuil_miembro
left join lote
```

```
on pro.codigo_producto=lote.codigo_producto
and pro.numero_lote=lote.numero
and lote.fecha_produccion between '20210601' and '20210630';
```

GROUP BY + HAVING

Enunciado

Indicar por cada cliente la cantidad de pedidos realizados este año. Mostrar los clientes que hayan realizado menos de 5 pedidos.

Incluir los clientes que no tengan pedidos en 2021. Indicar: cuil, nombre, apellido, dirección, cantidad de pedidos y la fecha del último pedido

Resolución

```
select cuil, nombre, apellido, domicilio, count(ped.cuil_cliente),
max(ped.fecha_pedido) ult_ped
from cliente cli
left join pedido ped
on cli.cuil = ped.cuil_cliente
and ped.fecha_pedido > 20210101
group by cuil, nombre, apellido, domicilio
having count(ped.cuil_cliente) < 5
```

NOT IN

Enunciado

Listar los Grupos cuyos productos desarrollados no hayan tenido un lote fabricado este año. Indicar número y nombre del grupo.

Resolución

```
select numero, nombre
from grupo
where numero not in
(
select prod.numero_grupo
from producto prod
inner join lote
on prod.codigo=lote.codigo_producto
where lote.fecha_produccion between '20210101' and '20211231'
```

);

VARs/SUBQUERY/TEMP

Enunciado

Listar los miembros de la cooperativa y la cantidad de grupos de los que forma parte. Mostrar sólo aquellos cuya cantidad de grupos sea menor al promedio de dichas cantidades de grupos a los que pertenecen todos los miembros.

Resolución

```
drop temporary table if exists cants;
```

```
create temporary table cants
select mi.cuil, nombre, apellido, count(*) cant
from miembro mi
inner join grupo_miembro gm
      on mi.cuil=gm.cuil_miembro
group by mi.cuil, nombre, apellido;
```

```
select avg(cant) into @prom from cants;
```

```
select *
from cants
where cant>@prom;
```

alternativas: calcular el prom en una variable de movida y usarla en having hacer prom en subcons

TRANSAC

Enunciado

Realice las operaciones para aplicar los siguientes cambios dentro de una transacción. Debido a la escasez del trigo burgol se ha decidido reemplazar el material "trigo sarraceno". Se debe dar de alta el nuevo material con los datos a continuación y sustituir el trigo burgol por este nuevo en la composición de los productos actuales.

Nombre: trigo sarraceno, unidad de medida: kg, stock: 100. Precio al día de hoy: 200.

Resolución

```
begin;
```

```
select codigo into @cod_tb from material where nombre='trigo  
burgol';
```

```
insert into material (nombre, unidad_medida, stock) values ('trigo  
sarraceno','kg',100);
```

```
select last_insert_id() into @cod_ts;  
#alternativa: select codigo into @cod_ts from material where  
nombre='trigo sarraceno';
```

```
insert into valor_material values (@cod_ts, current_date, 200);
```

```
update composicion set codigo_material=@cod_ts where  
codigo_material=@cod_tb;
```

```
commit;
```

INNER JOIN

Enunciado

Listar productos que tengan un pedido con fecha de entrega convenida durante este mes (septiembre). Indicando código y nombre del producto y del pedido número, fecha de entrega convenida, cantidad a entregar y precio unitario acordado. Ordenar por nombre de producto alfabético y fecha de entrega convenida ascendente.

Resolución

```
select prd.codigo, prd.nombre  
      , ped.numero, ped.fecha_entrega_convenida  
      , sol.cantidad, sol.precio_unitario_acordado  
from producto prd  
inner join solicita sol  
      on prd.codigo=sol.codigo_producto  
inner join pedido ped  
      on sol.numero_pedido=ped.numero  
where fecha_entrega_convenida between "20210601" and "20210630"  
order by prd.nombre, ped.fecha_entrega_convenida;
```

LEFT/RIGHT JOIN

Enunciado

Listar todos los pedidos hechos por un cliente (usar el nombre y apellido de algún cliente que haya cargado en su db) y sólo las entregas realizadas durante el mes pasado (agosto) a dichos pedidos. Indicar número de pedido, fecha del pedido, código y cantidad de cada producto solicitado y si hubo alguna entrega en junio indicar número de lote, fecha de entrega y cantidad entregada

Resolución

```
select ped.numero, ped.fecha_pedido
      , sol.codigo_producto, sol.cantidad
      , ent.numero_lote, ent.fecha_entrega, ent.cantidad
from cliente cli
inner join pedido ped
  on cli.cuil=ped.cuil_cliente
inner join solicita sol
  on ped.numero=sol.numero_pedido
left join entrega ent
  on sol.numero_pedido=ent.numero_pedido
  and sol.codigo_producto=ent.codigo_producto
  and ent.fecha_entrega between '20210701' and '20210731'
where cliente.apellido='Perez' and cliente.nombre='Juan';
```

GROUP BY + HAVING

Enunciado

Indicar los miembros de la cooperativa que hayan trabajado en menos de 10 lotes para la fabricacion en el mes pasado(agosto). Indicar cuil, nombre, apellido, cantidad de lotes trabajados y fecha del ultimo lote fabricaado en agosto donde participó. Si algun miembro no participó en agosto en ningun lote, deberá mostrar 0

```
select  mi.cuil, mi.nombre, mi.apellido, count(lote.numero)
        , sum(prd.horas_trabajadas), max(lote.fecha_produccion)
from miembro mi
left join produce prd
  on mi.cuil = prd.cuil_miembro
left join lote
```

```

on prd.codigo_producto=lote.codigo_producto
and prd.numero_lote=lote.numero
and lote.fecha_produccion > 20210801 and lote.fecha_produccion
< 20210831
group by mi.cuil, mi.nombre, mi.apellido
having count(lote.numero) < 10

```

NOT IN

Enunciado

Listar los productos que no tienen entregas en los últimos 2 meses (agosto y septiembre). Indicar, código y nombre del producto.

Resolución

```

select codigo, nombre
from producto
where codigo not in
(
select lote.codigo_producto
from lote
inner join entrega ent
on lote.codigo_producto=ent.codigo_producto
and lote.numero=ent.numero_lote
where ent.fecha_entrega between '20210801' and '20210930'
);

```

VARs/SUBQUERY/TEMP

Enunciado

Listar los miembros y la cantidad total de horas que han trabajado en la fabricación de lotes. Mostrando aquellos que han trabajado menos horas totales que el promedio de dichas horas totales de los miembros. Indicar cuil, nombre, apellido y la cantidad de horas totales trabajadas.

Resolución

```

drop temporary table if exists cants;

create temporary table cants
select cuil, nombre, apellido, sum(prd.horas_trabajadas) cants

```

```

from miembro mi
inner join produce prd
    on mi.cuil=prd.cuil_miembro
group by cuil, nombre, apellido;

select avg(cant) into @prom from cants;

select *
from cants
where cant>@prom;

```

alternativas: calcular el prom en una variable de movida y usarla en having hacer prom en subcons

TRANSAC

Enunciado

Realice las operaciones para aplicar los siguientes cambios dentro de una transacción. Debido a los problemas para comercializar algunos productos alimenticios la cooperativa ha decidido probar crear un grupo especial “Gourmet” para venderlos. Se decidió a tal efecto reasignar el producto “bruscheta” al nuevo grupo y además registrar los siguientes cambios. Se llamará “bruscheta el león”, precio sugerido será \$100 y se agregarán los siguientes materiales que ya existen: “jamon crudo dma” con cantidad 50 y “aceite oliva agroecol” con cantidad 0.05

Resolución

```

begin;

select codigo into @bru from producto where nombre='bruscheta';
select codigo into @jamon from material where nombre='jamon crudo dma';
select codigo into @aceite from material where nombre='aceite oliva agroecol';

insert into grupo(nombre) values ('Gourmet');
select last_insert_id() into @gourmet;
#alternativa: select numero into @gourmet from grupo where nombre='Gourmet';

```

```
update producto set nombre='bruscheta el león', precio_sugerido=100,  
numero_grupo=@gourmet where codigo=@bru;
```

```
insert into composicion values (@bru, @jamon, 50);  
insert into composicion values (@bru, @aceite, 0.05);
```

```
commit;
```

INNER JOIN

Enunciado

Listar productos que hayan sido entregados el mes pasado (junio). Indicando código y nombre del producto, número del lote, fecha de producción y de la entrega, fecha de entrega y cantidad entregada. Ordenar por nombre del producto alfabético y fecha de entrega ascendente.

Resolución

```
select prd.codigo, prd.nombre  
      , lote.numero, lote.fecha_produccion  
      , ent.fecha_entrega, ent.cantidad  
from producto prd  
inner join lote  
      on prd.codigo=lote.codigo_producto  
inner join entrega ent  
      on lote.codigo_producto=ent.codigo_producto  
      and lote.numero=ent.numero_lote  
where ent.fecha_entrega between '20210601' and '20210630'  
order by prd.nombre, ent.fecha_entrega;
```

LEFT/RIGHT JOIN

Enunciado

Listar todos los productos de un grupo (usar el nombre de algún grupo que hayan cargado) y en caso que hayan sido fabricados lotes este mes (julio). Indicar código y nombre del producto, el número de lote, cantidad producida.

Resolución

```
select prd.codigo, prd.nombre
```



```

        , lote.numero, lote.cantidad_producida
from producto prd
inner join grupo grp
    on prd.numero_grupo=grp.numero
left join lote
    on prd.codigo=lote.codigo_producto
    and lote.fecha_produccion between '20210701' and '20210731'
where grp.nombre='grupo';

```

GROUP BY + HAVING

Enunciado

Indicar los productos cuya cantidad de lotes producidos el mes pasado (agosto) sea menor a 20. Indicar código y nombre del producto, última fecha de producción de lote en agosto, cantidad de lotes producidos y la cantidad de horas trabajadas en el mismo mes.

Si algún producto no fue fabricado ningún lote durante el mes de agosto entonces debe aparecer con cantidad 0.

```

select prod.codigo, prod.nombre, ifnull(max(lote.fecha_produccion),
0) , count(lote.numero), ifnull(sum(horas_trabajadas), 0)
from producto prod
left join lote
    on prod.codigo=lote.codigo_producto
    and lote.fecha_produccion between 20210102 and 20210102
left join produce prd
    on lote.codigo_producto=prd.codigo_producto
    and lote.numero=prd.numero_lote
group by prod.codigo, prod.nombre
having count(lote.numero) < 20

```

NOT IN

Enunciado

Listar los clientes que no realizaron pedidos con productos que se pactara un valor superior a \$5000. Indicar CUIL, nombre y apellido.

Resolución

```

select cuil, nombre, apellido
from cliente

```

```

where cuil not in
(
    select cuil_cliente
    from pedido ped
    inner join solicita sol
        on ped.numero = sol.numero_pedido
        where precio_unitario_acordado > 5000
);

```

VARs/SUBQUERY/TEMP

Enunciado

Listar los productos y la cantidad producida de cada uno. Mostrar sólo los productos cuya cantidad producida supere el promedio de dichas cantidad producidas para todos los productos. Indicar código, número y cantidad producida.

Resolución

```
drop temporary table if exists cants;
```

```

create temporary table cants
select prod.codigo, prod.nombre, sum(lote.cantidad_producida) cant
from producto prod
inner join lote
    on prod.codigo=lote.codigo_producto
group by prod.codigo, prod.nombre;

```

```
select avg(cant) into @prom from cants;
```

```

select *
from cants
where cant>@prom;

```

alternativas: calcular el prom en una variable de movida y usarla en having
hacer prom en subcons

TRANSAC

Enunciado

Realice las operaciones para aplicar los siguientes cambios dentro de una transacción. Se ha creado el nuevo grupo “Agroecológico”. El mismo será conformado por 2 miembros ya pertenecientes a la cooperativa: “Souma Yukihiro” y “Sanji Vinsmoke”. Este grupo se hará cargo de desarrollar los productos “ensalada burgol” y “ensalada tabule” los cuales deberán ser reasignados a este nuevo grupo.

Resolución

```
begin;
```

```
select cuil into @sanji from miembro where nombre='Sanji' and  
apellido='Vinsmoke';
```

```
select cuil into @souma from miembro where nombre='Souma' and  
apellido='Yukihiro';
```

```
select codigo into @ensburgol from producto where nombre='ensalada  
burgol';
```

```
select codigo into @enstabule from producto where nombre='ensalada  
tabule';
```

```
insert into grupo(nombre) values ('Agroecológico');
```

```
select last_insert_id() into @agroeco;
```

```
#alternativa: select numero into @agroeco from grupo where  
nombre='Agroecológico';
```

```
insert into grupo_miembro values(@agroeco, @sanji);
```

```
insert into grupo_miembro values(@agroeco, @souma);
```

```
update producto set numero_grupo=@agroeco where codigo in  
(@ensburgol,@enstabule);
```

```
commit;
```

INNER JOIN

Enunciado

Listar productos donde el precio unitario acordado sea menor al precio sugerido. Indicar, código, nombre y precio sugerido del producto, número de pedido, fecha de pedido, cantidad pedida, precio unitario acordado y la diferencia con el sugerido. Ordenar por nombre del producto alfabéticamente y diferencia de precios descendiente.

Resolución

```
select    prod.codigo,      prod.nombre,          prod.precio_sugerido,
sol.numero_pedido, ped.fecha_pedido,
sol.cantidad,              sol.precio_unitario_acordado,
(sol.precio_unitario_acordado - prod.precio_sugerido) dif
from solicita sol
inner join producto prod on sol.codigo_producto = prod.codigo
inner join pedido ped on ped.numero = sol.numero_pedido
where sol.precio_unitario_acordado < prod.precio_sugerido
order by prod.nombre asc, dif desc;
```

LEFT/RIGHT JOIN

Enunciado

Listar todos los productos que desarrolle un grupo (usar el nombre de algún grupo que hayan cargado) e indicar los pedidos de este mes (agosto) si los tuvo. Indicar código y nombre del producto y si hubo pedidos, número y fecha de pedido, cantidad pedida y precio convenido

Resolución

```
select prd.codigo, prd.nombre
        ,      ped.numero,      ped.fecha_pedido,      sol.cantidad,
sol.precio_unitario_acordado
from grupo grp
inner join producto prd
    on prd.numero_grupo=grp.numero
left join solicita sol
    on prd.codigo=sol.codigo_producto
```

```

left join pedido ped
  on sol.numero_pedido=ped.numero
  and ped.fecha_pedido between '20210801' and '20210831'
where grp.nombre='grupo';

```

GROUP BY + HAVING

Enunciado

Se necesita evaluar los productos menos solicitados el año pasado, para analizar su próxima fabricación. Indicar por producto la cantidad de pedidos donde los han solicitado el año pasado. Mostrar aquellos que han sido solicitados en menos de 3 pedidos. Indicando código, nombre, característica, fecha del último pedido de 2020 y cantidad de pedidos. Si no existe ningún pedido en 2020, mostrar 0.

Resolución

```

select prod.codigo, nombre, características
,      max(ped.fecha_pedido),      count(sol.codigo_producto)      #o
sum(cantidad)?
from producto prod
left join solicita sol
  on prod.codigo = sol.codigo_producto
left join pedido ped
  on sol.numero_pedido=ped.numero
  and ped.fecha_pedido between 20200101 and 20201231
group by prod.codigo, nombre, características
having count(sol.codigo_producto) < 3

```

NOT IN

Enunciado

Listar los productos que entre sus materiales no se compongan de maní, nuez o almendra. Indicar código, nombre y precio sugerido.

Resolución

```

select *
from producto
where codigo not in
(
  select comp.codigo_producto

```

```

        from composicion comp
        inner join material mat
            on comp.codigo_material=mat.codigo
        where mat.nombre in ('maní', 'nuez', 'almendra')
    );

```

alternativas, not exists, subq depend, left

VARs/SUBQUERY/TEMP

Enunciado

Listar los clientes y la cantidad de pedidos que han realizado. Incluir sólo los clientes que han realizado una cantidad mayor al promedio de dichas cantidades de todos los clientes.

Resolución

```

drop temporary table if exists cants;

create temporary table cants
select nombre, apellido, count(*) cant
from cliente cli
inner join pedido ped
    on cli.cuil=ped.cuil_cliente
group by cuil, nombre, apellido;

select avg(cant) into @prom from cants;

select *
from cants
where cant>@prom;

```

alternativas: calcular el prom en una variable de movida y usarla en having hacer prom en subcons

TRANSAC

Enunciado

Realice las operaciones para aplicar los siguientes cambios dentro de una transacción. Se ha fabricado el día de hoy un nuevo lote del producto “bruscheta” que vence en 2 días. Debe insertarse con el

número siguiente de lote y la cantidad fabricada es 50. Se debe además registrar 2 hs de Sanji Vinsmoke de trabajo de producción para dicho lote. También se deberá incrementar el precio sugerido un 20%.

Resolución

```
begin;
```

```
select codigo into @brus from producto where nombre='bruscheta';
select numero+1 into @lote from lote where codigo_producto=@brus;
select cuil into @sanji from miembro where nombre='Sanji' and
apellido='Vinsmoke';
```

```
insert into lote values (@brus, @lote, 50, current_date,
current_date+2);
```

```
insert into produce values (@brus, @lote, @sanji, 2);
```

```
update producto set precio_sugerido=precio_sugerido*1.2 where
codigo=@brus;
```

```
commit;
```

INNER JOIN

Enunciado

Listar los miembros que hayan realizado producción el mes pasado (julio) en algún lote. Indicando cuil, nombre y apellido del miembro y de lo producido en dicho mes el código y nombre del producto, número de lote, fecha de producción y cantidad de horas que trabajó dicho miembro. Ordenar por apellido y nombre alfabéticamente.

Resolución

```
select mi.cuil, mi.nombre, mi.apellido
      , prod.codigo, prod.nombre
      , lote.numero, lote.fecha_produccion
      , pro.horas_trabajadas
from miembro mi
inner join produce pro
      on mi.cuil=pro.cuil_miembro
```

```
inner join lote
    on pro.codigo_producto=lote.codigo_producto
    and pro.numero_lote=lote.numero
inner join producto prod
    on lote.codigo_producto=prod.codigo
where lote.fecha_produccion between '20210701' and '20210731'
order by mi.apellido, mi.nombre
```

LEFT/RIGHT JOIN

Enunciado

Listar todos los lotes producidos hace 2 meses (junio) y si las hubo en ese mismo mes las entregas realizadas. Indicar código y nombre del producto, número de lote y fecha de producción y de las entregas de junio (sólo si hubo en junio), fecha de entrega y cantidad entregada.

Resolución

```
select prod.codigo, prod.nombre
    , lote.numero, lote.fecha_produccion
    , ent.fecha_entrega, ent.cantidad
from lote lote
inner join producto prod
    on lote.codigo_producto=prod.codigo
left join entrega ent
    on lote.codigo_producto=ent.codigo_producto
    and lote.numero=ent.numero_lote
    and ent.fecha_entrega between '20210601' and '20210630'
where fecha_produccion between '20210601' and '20210630';
```

GROUP BY + HAVING

Enunciado

Clientes que han realizado pocos pedidos el año pasado. Indicar por cliente la cantidad de pedidos que han realizado. Mostrar aquellos clientes que han hecho menos de 2 pedidos en 2020. Si algún cliente no ha realizado ningún pedido ese año mostrarlo igualmente con

cantidad de pedidos 0. Indicar cuil, nombre, apellido, cantidad de pedidos y última fecha de pedido.

Resolución

```
select cli.cuil, cli.nombre, cli.apellido
       , count(ped.numero) cant_pedidos, max(ped.fecha_pedido)
       ult_pedido
from cliente cli
left join pedido ped
      on cli.cuil = ped.cuil_cliente
      and ped.fecha_pedido between 20200101 and 20201231
group by cli.cuil, cli.nombre, cli.apellido
having count(ped.numero) <2;
```

NOT IN

Enunciado

Listar los miembros de la cooperativa que no hayan participado en la producción de ningún lote fabricado durante el mes pasado (agosto). Indicar CUIL, Nombre y Apellido.

Resolución

```
select cuil, nombre, apellido
from miembro
where cuil not in
      (
        select cuil_miembro
        from produce prd
        inner join lote
              on prd.codigo_producto=lote.codigo_producto
              and prd.numero_lote=lote.numero
              where lote.fecha_produccion between '20210801' and
'20210831'
      );
# alternativas, not exists, subquery dep, left con is null
```

VARs/SUBQUERY/TEMP

Enunciado

Listar los Grupos que desarrollan gran cantidad de productos. Indicar para cada Grupo el número, nombre y cantidad de productos que desarrollan, pero sólo mostrar aquellos grupos cuya cantidad supere al promedio de dichas cantidades desarrolladas por todos los grupos.

Resolución

```
drop temporary table if exists cants;
create temporary table cants
select grp.numero, grp.nombre, count(*) cant
from grupo grp
inner join producto prod
    on grp.numero=prod.numero_grupo
group by grp.numero, grp.nombre;
```

```
select avg(cant) into @prom from cants;
```

```
select *
from cants
where cant>@prom;
```

alternativas: calcular el prom en una variable de movida y usarla en having
hacer prom en subcons

TRANSAC

Enunciado

Realice las operaciones para aplicar los siguientes cambios dentro de una transacción. Se ha decidido cambiar la composición de la “ensalada burgol”, se retirarán los materiales “sal” y “aceite oliva agroecol” y se agregará 0.5 de un nuevo material que debe ser dado de alta con los siguientes datos, nombre: “arroz negro”, unidad de medida: kg y stock: 10.

Resolución

```
begin;
```

```
select codigo into @sal from material where nombre='sal';
select codigo into @aceite from material where nombre='aceite oliva
agroecol';
select codigo into @ensalada from producto where nombre='ensalada
burgol';

insert into material (nombre, unidad_medida, stock) values ('arroz
negro', 'kg', 10);

select last_insert_id() into @an;
#alternativa: select codigo into @an from material where
nombre="arroz negro";

delete from composicion where codigo_producto=@ensalada and
codigo_material in (@sal, @aceite);
#alternativa: delete from composicion where
codigo_producto=@ensalada and codigo_material in (select codigo from
material where nombre in ('sal','aceite oliva agroecol'));

insert into composicion values(@ensalada, @an,0.5);

commit;
```