

Guardería GaghieI - Parcial SQL Regularidad

301

301.1 inner join

Enunciado

301.1 - **Contratos activos en sectores manuales.** Listar las embarcaciones con contratos activos (sin fecha de baja) que se estén guardando en camas de sectores de tipo de operación manual. Indicar hin y nombre de la embarcación, fecha y hora del contrato, número de cama, código y nombre del sector donde se almacenan, código y nombre del tipo de embarcación. Ordenar los registros por fecha y hora de contrato descendente y nombre ascendente.

Resolución

```
select em.hin, em.nombre
      , ec.fecha_hora_contrato, ec.numero_cama
      , se.codigo, se.nombre
from embarcacion em
inner join embarcacion_cama ec
      on em.hin=ec.hin
inner join sector se
      on se.codigo = ec.codigo_sector
Inner join tipo_embarcacion ti
      On ti.codigo = em.codigo_tipo_embarcacion
where ec.fecha_hora_baja_contrato is null
      and se.tipo_operacion = 'Manual'
order by ec.fecha_hora_contrato desc, em.nombre asc;
```

301.2 left join

Enunciado

301.2 **Socios que se retrasan.** Listar los socios que son personas físicas (con tipo de documento dni) y sus embarcaciones y, en caso que en alguna de las salidas hayan regresado tarde, listar

dichas salidas. Indicar, numero y nombre del socio; hin, nombre y descripción de la embarcacion; fechas y horas de salida, regreso real y tentativo.

Resolución

```
select soc.numero, soc.nombre
      , em.hin, em.nombre,em.descripcion
      ,sal.fecha_hora_salida, sal.fecha_hora_regreso_real,
sal.fecha_hora_regreso_tentativo
from socio soc
inner join embarcacion em
      on em.numero_socio=soc.numero
left join salida sal
      on em.hin=sal.hin
      and sal.fecha_hora_regreso_real > sal.fecha_hora_regreso_tentativo
where soc.tipo_doc='dni'
```

301.3 group by

Enunciado

301.3 - **Socios con pocos cursos recientes.** Listar todos los socios que realizaron menos de 5 cursos el año pasado según la fecha de inscripción al curso. Indicar número y nombre del socio; cantidad de cursos de 2024, (si no realizó ningún curso este año indicar 0) y la fecha en la que se inscribió por última vez a un curso (si no se inscribió a ninguno mostrar 'sin inscripciones'). Ordenar por cantidad de cursos descendente.

Resolución

```
select s.numero, s.nombre
      , count(i.numero_socio) cant_cursos
      , coalesce(max(i.fecha_hora_inscripcion),'sin inscripciones')
ultima_insc
from socio s
left join inscripcion i
      on s.numero=i.numero_socio
and i.fecha_hora_inscripcion ≥ '20240101'

group by s.numero
having cant_cursos < 5
order by cant_cursos desc;
```

301.4 not in

Enunciado

301.4 - **Actividades recientes para realizar.** Listar las actividades de las que se vayan a realizar cursos a futuro pero no se haya realizado ninguno antes de este año. Indicar número y nombre de la actividad. Ordenar por número de actividad ascendente.

Resolución

```
#####opción A
select distinct act.numero, act.nombre
from actividad act
inner join curso cur
    on act.numero=cur.numero_actividad
where cur.fecha_inicio >= now()
    and act.numero not in (
        select c.numero_actividad
        from curso c
        where c.fecha_inicio < '20240101'
    )
order by act.numero;
```

```
#####opción B
select act.numero, act.nombre
from actividad act
where act.numero in (
    select cur.numero_actividad
    from curso cur
    where cur.fecha_inicio >= now()
)
and act.numero not in (
    select c.numero_actividad
    from curso c
    where c.fecha_inicio < '20240101'
)
order by act.numero;
```

```
#####opcion C
select act.numero, act.nombre
from actividad act
where exists (
    select 1
    from curso cur
```

```

        where cur.numero_actividad=act.numero
              and cur.fecha_inicio >= now()
    )
    and not exists (
        select 1
        from curso c
        where c.numero_actividad=act.numero
              and c.fecha_inicio < '20240101'
    )
    order by act.numero;

####opcion D
select distinct act.numero, act.nombre
from actividad act
inner join curso cur
    on act.numero=cur.numero_actividad
left join curso cur_no
    on act.numero=cur_no.numero_actividad
    and cur_no.fecha_inicio < '20240101'
where cur.fecha_inicio >= now()
    and cur_no.numero is null
order by act.numero;

```

301.5 subquery/temp/variable/cte

Enunciado

301.5 - Socios con más cursos. Listar los socios que se inscribieron a más cursos en 2024 que el promedio de la cantidad de cursos a la que se inscribió cada socio durante el 2024. Se debe considerar los socios que no se inscribieron a ningún curso en 2024 dentro del cálculo del promedio. Indicar número, nombre del socio y cantidad de cursos que realiza en 2024. Ordenar por cantidad de cursos descendente.

Resolución

```

with cur_soc as (
    select soc.numero, soc.nombre, count(ins.numero_socio) cant_cursos
    from socio soc
    left join inscripcion ins
        on soc.numero=ins.numero_socio
        and ins.fecha_hora_inscripcion > '20240101'
    group by soc.numero, soc.nombre
)

```

```

select *
from cur_soc
where cur_soc.cant_cursos > (
    select avg(cs.cant_cursos)
    from cur_soc cs
)
order by cant_cursos desc;

```

301.6 transactions

Enunciado

301.6- Jet Ski espacial Debido al poco éxito de la actividad “Carrera de Asteroides” la empresa ha decidido modificarla para que la misma se realice con “Jet Ski”. Dar de alta este nuevo tipo de embarcación que requiere operatoria manual y modificar la actividad Carrera de Asteroides para que la utilice.

La empresa además decidió organizar para este año nuevos cursos de carrera de asteroides iguales a los que se realizaron el año 2023 (sólo que 1 año más tarde). No se cargarán horarios por ahora ya que los definirá la empresa luego

Resolución

```

begin;

select numero into @ca
from actividad
where nombre='Carrera de Asteroides';

insert into tipo_embarcacion(nombre, operacion_requerida)
values ('Jet Ski', 'Manual');

select LAST_INSERT_ID() into @js;

update actividad
set codigo_tipo_embarcacion=@js
where numero=@ca;

insert into curso(fecha_inicio, fecha_fin, cupo, legajo_instructor,
numero_actividad)
select adddate(fecha_inicio, interval 1 year), adddate(fecha_fin,
interval 1 year), cupo, legajo_instructor, numero_actividad
from curso
where numero_actividad=@ca;

```

```
commit;
```

301.7 routines

Enunciado

301.7- Ordenando el closet. La empresa quiere optimizar el uso de los sectores más cercanos a la costa, se necesita crear una rutina "embarcaciones_posibles" que reciba un código de un sector como parámetro y devuelva la lista de embarcaciones registradas en el sistema que podrían almacenarse en dicho sector (según los tipos de embarcación que dicho sector puede almacenar) pero que no estén actualmente almacenadas en él. Indicar hin y nombre de la embarcación.

Utilizar el código de sector 2 para probar la rutina (debe incluir esto en la entrega).

Resolución

```
delimiter ~-~

drop procedure if exists embarcaciones_posibles ~-~

create procedure embarcaciones_posibles(in cod_sec int unsigned)
READS SQL DATA
begin
select distinct e.hin, e.nombre
from sector_tipo_embarcacion ste
inner join embarcacion e
    on ste.codigo_tipo_embarcacion=e.codigo_tipo_embarcacion
where ste.codigo_sector=cod_sec
    and e.hin not in (
        select ec.hin
        from embarcacion_cama ec
        where ec.codigo_sector=cod_sec
            and (ec.fecha_hora_baja_contrato is null
                or
                ec.fecha_hora_baja_contrato ≥ current_date)
    );
end ~-~

delimiter ;
call embarcaciones_posibles(2);
```

302

302.1 inner join

Enunciado

302.1 - **Veleros y Lanchas en almacenamiento.** Socios con las embarcaciones de tipo Velero o Lancha que actualmente se estén guardando (sin fecha y hora de baja de contrato). Indicar numero y nombre del socio, hin y nombre de la embarcación, código y nombre de tipo de embarcación. Ordenar alfabéticamente por nombre de tipo de embarcación y nombre de la embarcación.

Resolución

```
select soc.numero, soc.nombre
      , em.hin, em.nombre
      , te.codigo, te.nombre
from socio soc
inner join embarcacion em
      on soc.numero=em.numero_socio
inner join embarcacion_cama ec
      on em.hin=ec.hin
inner join tipo_embarcacion te
      on em.codigo_tipo_embarcacion=te.codigo
where ec.fecha_hora_baja_contrato is null
      and te.nombre in ('Velero', 'Lancha')
order by te.nombre, em.nombre;
```

302.2 left join

Enunciado

Listar las actividades para los tipos de embarcaciones 'Lancha' y 'Tabla Wind Surf' y, si se realizaron cursos desde abril al presente mostrarlos. Indicar numero y nombre de la actividad, nombre del tipo de embarcacion, numero, fecha de inicio y fecha de fin del curso.

Resolución

```
select act.numero, act.nombre
      , te.nombre
      , cur.numero, cur.fecha_inicio, cur.fecha_fin
```

```

from actividad act
inner join tipo_embarcacion te
    on act.codigo_tipo_embarcacion=te.codigo
left join curso cur
    on act.numero=cur.numero_actividad
    and cur.fecha_inicio > '20240401'
where te.nombre in ('Tabla Wind Surf','Lancha');

```

302.3 group by

Enunciado

302.3 - Personas físicas con pocos contratos activos. Listar las personas físicas (propietarios con tipo de documento dni) que tengan menos de 2 contratos activos actualmente. Indicar número y nombre del socio; cantidad de contratos activos actualmente (si no tienen ninguno debe mostrar 0) y fecha de inicio de su último contrato (si no tuvo contratos debe decir 'sin contratos'). Ordenar por cantidad de contratos activos descendente.

Resolución

```

select soc.numero, soc.nombre
    , count(ec.hin) cant_contratos_activos
    , coalesce(max(ec.fecha_hora_contrato),'sin contratos') ult_contrato
from socio soc
left join embarcacion emb # inner no incluye socios que no contratan
    on soc.numero=emb.numero_socio # camas (se perdona en regularidad)
left join embarcacion_cama ec
    on emb.hin=ec.hin
    and ec.fecha_hora_contrato ≤ now() #esto puede darse por sentado
    and ec.fecha_hora_baja_contrato is null
where soc.tipo_doc='dni'
group by soc.numero
having cant_contratos_activos < 2
order by cant_contratos_activos desc;

```

Otra forma de resolverlo.

```

select soc.numero numero_socio, soc.nombre nombre_socio,
count(eca.fecha_hora_contrato)-
count(eca.fecha_hora_baja_contrato) cant_contratos_activos,
coalesce(max(eca.fecha_hora_contrato), 'Sin contratos')
ultimo_contrato

```



```

from socio soc
left join embarcacion emb
on soc.numero=emb.numero_socio
left join embarcacion_cama eca
on emb.hin=eca.hin
where soc.tipo_doc='dni'
group by soc.numero, soc.nombre
having cant_contratos_activos < 2
order by cant_contratos_activos desc
;

```

302.4 not in

Enunciado

302.4 - Embarcaciones sin salidas recientes. Listar las embarcaciones que hayan tenido salidas el último año pero no en los últimos 6 meses. Indicar hin, nombre y descripción de la embarcación. Ordenar por nombre.

Resolución

```

#####opción B
select emb.hin, emb.nombre, emb.descripcion
from embarcacion emb
where emb.hin in (
    select sal.hin
    from salida sal
    where sal.fecha_hora_salida ≥ date_add(now(), interval -1 year)
)
and emb.hin not in (
    select sal.hin
    from salida sal
    where sal.fecha_hora_salida ≥ date_add(now(), interval -6 month)
)
order by emb.nombre;

```

302.5 subquery/temp/variable/cte

Enunciado

302.5 - Embarcaciones con pocas salidas. Listar las embarcaciones que tuvieron menos salidas en 2024 que el promedio de salidas de cada socio en 2024. Se deben tener en cuenta en el promedio los

socios sin salidas. Indicar hin, nombre y descripción de la embarcación y cantidad de salidas. Ordenar por cantidad de salidas descendente.

Resolución

```
with emb_sal as (  
    select emb.hin, emb.nombre, emb.descripcion  
        , count(sal.hin) cant_salidas  
    from embarcacion emb  
    left join salida sal  
        on emb.hin=sal.hin  
        and sal.fecha_hora_salida > '20240101'  
    group by emb.hin  
)  
select *  
from emb_sal  
where emb_sal.cant_salidas < (  
    select avg(cant_salidas)  
    from emb_sal  
)  
order by cant_salidas desc;
```

302.6 Transactions

Enunciado

302.6- Reemplazo por sábitico El instructor Yang Wen-li va a tomarse un año sabático a partir del 10 de octubre de 2024, la empresa ha decidido contratar a una nueva instructora:

Legajo: 20

Cuil: 20-01234567-9

Nombre y apellido: Frederica Greenhill

Teléfono: 555-0120

Ella puede dictar las mismas actividades que Yang Wen-li y debe reemplazarlo como instructor en sus cursos que inicien posteriormente a la fecha indicada.

Resolución

```
begin;  
  
select i.legajo into @wl  
from instructor i
```

```

where i.nombre='Yang' and i.apellido='Wen-li';

insert into instructor values (20, '20-01234567-9', 'Frederica',
'Greenhill', '555-0120');

insert into instructor_actividad
select 20, ia.numero_actividad
from instructor_actividad ia
where ia.legajo_instructor=@wl;

update curso
set legajo_instructor=20
where legajo_instructor=@wl
and fecha_inicio >= '20241010';

commit;

```

302.7 routines

Enunciado

302.7- Muestrame las actividades. La empresa desea empezar a fomentar los cursos para algún tipo de embarcación a la vez. Por ello se necesita crear una "actividades_para_tipos" que reciba el código de tipo de embarcación como parámetro y devuelva la lista de todas las actividades que se pueden realizar para dicho tipo de embarcación, junto con todos los posibles instructores para cada actividad pero que no se esté dictando actualmente ningún curso.

Indicar número y nombre de actividad, legajo, nombre y apellido del instructor.

Invocar la rutina con el código de tipo de embarcación 6 para probarla (incluirlo en la entrega)

Resolución

```

delimiter [-°-°]-

drop procedure if exists actividades_para_tipos [-°-°]-

create procedure actividades_para_tipos(in cod_tipo int unsigned)
reads sql data
begin

```

```

select a.numero, a.nombre actividad
      ,i.legajo, i.nombre, i.apellido
from actividad a
inner join instructor_actividad ia
      on a.numero=ia.numero_actividad
inner join instructor i
      on ia.legajo_instructor=i.legajo
where a.codigo_tipo_embarcacion=cod_tipo
      and a.numero not in (
          select c.numero_actividad
          from curso c
          where current_date between c.fecha_inicio and c.fecha_fin
      );
end [-º-º]-

delimiter ;

call actividades_para_tipos(6);

```

303

303.1 inner join

Enunciado

Listar las salidas realizadas el mes pasado. Indicando hin y nombre de la embarcación; fecha y hora de salida, regreso tentativo y regreso real; código y nombre del sector donde se la almacena. Ordenar alfabéticamente por nombre de sector y descendente por fecha y hora de salida.

Resolución

```

select em.hin, em.nombre
      , sal.fecha_hora_salida, sal.fecha_hora_regreso_tentativo
      , sal.fecha_hora_regreso_real
      , sec.codigo, sec.nombre
from salida sal
inner join embarcacion em
      on sal.hin=em.hin
inner join embarcacion_cama ec
      on sal.hin=ec.hin
inner join sector sec
      on ec.codigo_sector=sec.codigo
where ec.fecha_hora_baja_contrato is null

```

```
and sal.fecha_hora_salida between '20240701' and '20240731T235959'
order by sec.nombre, sal.fecha_hora_salida desc;
```

303.2 left join

Enunciado

Listar las actividades que se realizan para los tipos de embarcaciones canoa, kayak, velero y lancha y, si se realizarán cursos que empiecen más adelante indicarlos. Indicar código y nombre del tipo de embarcación; nombre y descripción de la actividad; numero, fecha de inicio y fecha de fin del curso.

Resolución

```
select te.codigo, te.nombre
      , act.nombre, act.descripcion
      , cur.numero, cur.fecha_inicio, cur.fecha_fin
from tipo_embarcacion te
inner join actividad act
      on te.codigo=act.codigo_tipo_embarcacion
left join curso cur
      on act.numero=cur.numero_actividad
      and cur.fecha_inicio > now()
where te.nombre in ('Canoa','Kayak','Lancha', 'Velero')
```

303.3 group by

Enunciado

303.3 - Embarcaciones con pocas salidas Listar las embarcaciones de empresas (el tipo de documento del propietario debe ser cuit) que tengan menos de 4 salidas durante el 2024. Indicar el número y nombre del propietario; el hin, nombre y descripción de la embarcación; la cantidad de salidas durante 2024 (si no tuvo salidas en 2024 debe indicarse 0), y la fecha de su última salida (si no tuvo salidas debe indicar 'sin salidas registradas'). Ordenar por fecha de última salida descendente.

Resolución

```
select soc.numero, soc.nombre
      , emb.hin, emb.nombre, emb.descripcion
```

```
, count(sal.hin) cant_salidas,
coalesce(max(sal.fecha_hora_salida), 'sin salidas registradas')
ult_salida
from embarcacion emb
inner join socio soc
    on emb.numero_socio=soc.numero
left join salida sal
    on emb.hin=sal.hin
    and sal.fecha_hora_salida > '20240101'
where tipo_doc='cuit'
group by emb.hin
having cant_salidas < 4
order by ult_salida desc;
```

303.4 not in

Enunciado

303.4 - Embarcaciones que no renovaron contrato. Listar embarcaciones que hayan estado almacenadas el año pasado (según la fecha de contrato) pero no tengan contratos activos actualmente. Indicar hin, nombre y descripción de la embarcación. Ordenar por nombre.

Resolución

```
#####opción A
select distinct emb.hin, emb.nombre, emb.descripcion
from embarcacion emb
inner join embarcacion_cama ec
    on emb.hin=ec.hin
where ec.fecha_hora_contrato between '20230101' and '20231231T235959'
and emb.hin not in (
    select emca.hin
    from embarcacion_cama emca
    where emca.fecha_hora_baja_contrato is null
        # or emca.fecha_hora_baja_contrato > now() #se perdona si no lo
    hacen
)
order by emb.nombre;
```

303.5 subquery/temp/variable/cte

Enunciado

303.5 - Sectores con muchas embarcaciones. Listar los sectores que tienen más embarcaciones almacenadas actualmente que el promedio de la cantidad de embarcaciones actualmente almacenadas en cada sector. Se deben tener en cuenta para el promedio los sectores que podrían no tener embarcaciones actualmente almacenadas. Indicar código, nombre, cantidad de embarcaciones actualmente almacenadas.

Resolución

```
drop temporary table if exists emb_alm;
create temporary table emb_alm
  select sec.codigo, sec.nombre
        , count(ec.fecha_hora_contrato) cant_emb_almac
  from sector sec
 left join embarcacion_cama ec
    on sec.codigo=ec.codigo_sector
    and ec.fecha_hora_baja_contrato is null
 group by sec.codigo, sec.nombre
;

select avg(cant_emb_almac) into @prom
from emb_alm;

select *
from emb_alm
where cant_emb_almac > @prom;

drop temporary table if exists emb_alm;
```

303.6 transactions

Enunciado

303.6- Transferencia por quiebra La empresa socio Dread Team ha quebrado y transferido todas sus embarcaciones a otra empresa llamada Casval Rem Deikun. Se requiere dar de alta la nueva empresa como socio con los siguientes datos:

Tipo de documento: cuit

Numero de documento: 134134134

Nombre: Casval Rem Deikun

Luego debe reemplazarse como propietaria de las embarcaciones por la nueva empresa. Finalmente al cambiar el propietario deben

darse de baja los contratos aún activos de guardado de las embarcaciones transferidas con fecha de hoy.

Resolución

```
begin;

select numero into @dread_team
from socio
where nombre='Dread Team';

insert into socio (tipo_doc, nro_doc, nombre)
values ('cuit', '134134134', 'Casval Rem Deikun');

select LAST_INSERT_ID() into @casval;

update embarcacion
set numero_socio=@casval
where numero_socio=@dread_team;

update embarcacion_cama ec
inner join embarcacion e
on ec.hin=e.hin
set fecha_hora_baja_contrato = now()
where fecha_hora_baja_contrato is null
and e.numero_socio = @casval;

commit;
```

303.7 routines

Enunciado

303.7- A trabajar. La empresa quiere aprovechar al máximo el potencial de cada instructor. Para ello se necesita crear una rutina llamada "cursos_posibles" que reciba como parámetro el legajo de un instructor y devuelva una lista de cursos pasados (ya finalizados) que podría haber dictado, en base a las actividades que puede dictar. Excluir los cursos de aquellas actividades que sí ha dictado en algún curso.

Indicar número, cupo, duración días del curso, nombre y descripción de la actividad

Utilizar el legajo 1 para probar la rutina (incluir esto en la entrega).

Resolución

```
delimiter (⌘⌘⌘)

drop procedure if exists cursos_posibles (⌘⌘⌘)

create procedure cursos_posibles(in leg_ins int unsigned)
reads sql data
begin
select c.numero, c.cupo, DATEDIFF(c.fecha_fin, c.fecha_inicio),
a.nombre, a.descripcion
from instructor_actividad ia
inner join curso c
    on c.numero_actividad=ia.numero_actividad
inner join actividad a
    on c.numero_actividad=a.numero
where ia.legajo_instructor=leg_ins
    and CURRENT_DATE ≥ c.fecha_fin
    and c.numero_actividad not in (
        select cu.numero_actividad
        from curso cu
        where cu.legajo_instructor =leg_ins
    );
end (⌘⌘⌘)

delimiter ;

call cursos_posibles(1);
```

304

304.1 inner join

Enunciado

Listar los cursos que se dictarán en el futuro para los tipos de embarcaciones de tabla (contengan la palabra tabla en el nombre). Indicar numero y nombre de actividad; número, fecha de inicio y fin del curso, código y nombre del tipo de embarcación; días y horarios de inicio y fin de dictado. Ordenar alfabéticamente por

nombre de tipo de embarcación y ascendente por fecha de inicio del curso.

Resolución

```
select act.numero, act.nombre
      , cur.numero, cur.fecha_inicio, cur.fecha_fin
      , te.codigo, te.nombre
      , dc.dia_semana, dc.hora_inicio, dc.hora_fin
from tipo_embarcacion te
inner join actividad act
      on te.codigo=act.codigo_tipo_embarcacion
inner join curso cur
      on cur.numero_actividad=act.numero
inner join dictado_curso dc
      on cur.numero=dc.numero_curso
where cur.fecha_inicio > now()
      and te.nombre like '%tabla%'
order by te.nombre, cur.fecha_inicio;
```

304.2 left join

Enunciado

Listar los socios con embarcaciones de tipo 'No convencional' y, si salieron con dicha embarcación este año (2024), mostrar datos de dichas salidas. Indicar número y nombre del socio; hin, nombre y descripción de la embarcación, fecha y hora de salida, regreso tentativo y regreso real.

Resolución

```
select s.numero, s.nombre
      , emb.hin, emb.nombre, emb.descripcion
      , sal.fecha_hora_salida, sal.fecha_hora_regreso_tentativo,
      sal.fecha_hora_regreso_real
from socio s
inner join embarcacion emb
      on emb.numero_socio=s.numero
inner join tipo_embarcacion te
      on emb.codigo_tipo_embarcacion=te.codigo
left join salida sal
      on emb.hin=sal.hin
```

```
and sal.fecha_hora_salida > '20240101'  
where te.nombre='No convencional'
```

304.3 group by

Enunciado

304.3 - **Instructores con baja convocatoria.** Listar todos los instructores que hayan tenido menos de 4 socios inscriptos en total a los cursos que dictaron y ya terminaron. Indicar legajo, nombre y apellido del instructor, cantidad de alumnos inscriptos (debe mostrar 0 si no hubo) y última fecha que dictó un curso (debe mostrar 'sin cursos' en caso que no haya dictado ninguno ya finalizado). Ordenar por cantidad de inscriptos descendente.

Resolución

```
select inst.legajo, inst.nombre, inst.apellido  
      , count(insc.numero_curso) cant_insc  
      , coalesce(max(cur.fecha_inicio), 'sin cursos') ult_curso  
from instructor inst  
left join curso cur  
      on inst.legajo=cur.legajo_instructor  
      and cur.fecha_fin ≤ now()  
left join inscripcion insc  
      on cur.numero=insc.numero_curso  
group by inst.legajo, inst.nombre, inst.apellido  
having cant_insc < 4  
order by cant_insc desc;
```

304.4 not in

Enunciado

304.4 - **Socios que dejaron de hacer cursos.** Listar los socios que se hayan inscripto a cursos el año pasado pero no este. Indicar número y nombre del socio. Ordenar por nombre.

Resolución

```
#####opción C  
select soc.numero, soc.nombre  
from socio soc  
where exists (
```

```

select 1
from inscripcion ins
where ins.numero_socio=soc.numero
and ins.fecha_hora_inscripcion between '20230101' and
'20231231T235959'
) and not exists (
select 1
from inscripcion i
where i.numero_socio=soc.numero
and i.fecha_hora_inscripcion > '20240101'
)
order by soc.nombre;

```

304.5 subquery/temp/variable/cte

Enunciado

304.5 - **Instructores haraganes**. Listar los instructores que dictan menos cursos durante 2024 que el promedio de la cantidad de cursos que dicta cada instructor en 2024. Se deben tener en cuenta para el promedio los instructores que no dictan cursos. Indicar legajo, nombre y apellido del instructor y la cantidad de cursos que dicta en 2024. Ordenar por cantidad de cursos descendente.

Resolución

```

drop temporary table if exists cur_inst;

create temporary table cur_inst
select ins.legajo, ins.nombre, ins.apellido
, count(cur.numero) cant_cursos
from instructor ins
left join curso cur
on ins.legajo=cur.legajo_instructor
and cur.fecha_inicio > '20240101'
group by ins.legajo, ins.nombre, ins.apellido
;

select avg(cant_cursos) into @prom
from cur_inst;

select *
from cur_inst
where cant_cursos < @prom

```

```
order by cant_cursos desc;
```

```
drop temporary table cur_inst;
```

304.6 transactions

Enunciado

304.5- Que empiece yá, que el público se va. Debido al éxito del curso 35, los socios inscriptos han pedido hacerlo antes. Pero la empresa ya realizaba publicidad, por eso ha decidido crear un nuevo curso idéntico al 35 pero que comience 10 días antes con el mismo instructor y actividad, dure la misma cantidad de días, el mismo cupo y mismos horarios. Luego reasignar todos los socios actualmente inscriptos al nuevo curso.

Resolución

```
begin;
```

```
insert into curso (fecha_inicio, fecha_fin, cupo, legajo_instructor,  
numero_actividad)
```

```
select adddate(fecha_inicio,-10), adddate(fecha_fin,-10), cupo,  
legajo_instructor, numero_actividad
```

```
from curso
```

```
where numero=35;
```

```
select LAST_INSERT_ID() into @nuevo_curso;
```

```
insert into dictado_curso
```

```
select @nuevo_curso, dia_semana, hora_inicio, hora_fin
```

```
from dictado_curso
```

```
where numero_curso=35;
```

```
update inscripcion
```

```
set numero_curso=@nuevo_curso
```

```
where numero_curso=35;
```

```
commit;
```

304.7 routines

Enunciado

304.7- Promo para socio. La empresa desea incrementar la cantidad de socios que asisten a cursos. Se debe crear una rutina llamada "actividades_para_socio" que reciba un número de socio como parámetro y liste las actividades que podría realizar con alguna de sus embarcaciones, en base al tipo de embarcación, pero que aún no haya realizado ningún curso de dicha actividad. Indicar hin y nombre de la embarcación, número, nombre y descripción de la actividad. Utilizar el socio número 5 para invocar la rutina (incluirlo en la entrega.)

Resolución

```
delimiter (-o_o)

drop procedure if exists actividades_para_socio (-o_o)

create procedure actividades_para_socio(in num_soc int unsigned)
reads sql data
begin
select e.hin, e.nombre
      , a.numero, a.nombre, a.descripcion
from embarcacion e
inner join actividad a
      on a.codigo_tipo_embarcacion=e.codigo_tipo_embarcacion
where e.numero_socio=num_soc
      and a.numero not in (
      select c.numero_actividad
      from curso c
      inner join inscripcion i
            on i.numero_curso=c.numero
      where i.numero_socio=num_soc
      );
end (-o_o)

call actividades_para_socio(5);
```

305

305.1 inner join

Enunciado

305.1 - Proximas actividades. Actividades de cursos que no hayan comenzado aún, tengan cupo superior a 5 y tengan inscriptos. Indicar el número y nombre de la actividad; número, fecha de inicio y cupo del curso, número y nombre del socio. Ordenar por fecha de inicio del curso descendente y por nombre alfabéticamente.

Resolución

```
select act.numero, act.nombre
       , cur.numero, cur.fecha_inicio, cur.cupo
       , soc.numero, soc.nombre
from   curso cur
inner join actividad act
       on cur.numero_actividad=act.numero
inner join inscripcion ins
       on ins.numero_curso=cur.numero
inner join socio soc
       on ins.numero_socio=soc.numero
where  cur.fecha_inicio > now()
       and cur.cupo > 5
order by fecha_inicio desc, soc.nombre;
```

305.2 left join

Enunciado

305.2. Veleros y lanchas almacenados. Listar las embarcaciones de tipo Lancha o Velero que están registradas en el sistema y, si aún tienen un contrato activo (sin fecha y hora de baja) mostrar los datos del contrato y almacenamiento. Indicar código y nombre del tipo de embarcación; hin y nombre de la embarcación; fecha y hora de contrato, código del sector y número de cama correspondiente al contrato.

Resolución

```
select te.codigo, te.nombre
       , em.hin, em.nombre
       , ec.fecha_hora_contrato, ec.codigo_sector, ec.numero_cama
from   tipo_embarcacion te
inner join embarcacion em
       on te.codigo=em.codigo_tipo_embarcacion
left join embarcacion_cama ec
```

```
on em.hin=ec.hin
and ec.fecha_hora_baja_contrato is null
where te.nombre in ('lancha','velero');
```

305.3 group by

Enunciado

305.3 - Actividades para embarcaciones manuales con pocos cursos. Para los tipos de embarcación que requieren operación manual listar las actividades de las mismas para las que se hayan realizado menos de 4 cursos este año (según la fecha de inicio). Indicar el nombre del tipo de embarcación; número, nombre y descripción de la actividad; cantidad de cursos (en caso de no haberse realizado ninguno este año indicar 0) y la fecha del primer curso de este año (si no hay ninguno debe indicar 'sin cursos'). Ordenar por cantidad de cursos descendente.

Resolución

```
select te.nombre, act.numero, act.nombre, act.descripcion
      , count(cur.numero) cant_cursos
      , coalesce(min(cur.fecha_inicio), 'sin cursos')
from tipo_embarcacion te
inner join actividad act
      on act.codigo_tipo_embarcacion=te.codigo
left join curso cur
      on act.numero=cur.numero_actividad
      and cur.fecha_inicio >= '20240101'
where te.operacion_requerida = 'Manual'
group by act.numero
having cant_cursos < 4
order by cant_cursos desc;
```

305.4 not in

Enunciado

305.4 - Embarcaciones puntuales. Listar las embarcaciones que tengan regresos puntuales (la fecha y hora de regreso real sea menor o igual a la tentativa) pero no tengan ningún regreso impuntual (fecha y hora de regreso real posterior a la

tentativa). Indicar hin, nombre y descripción. Ordenar por nombre.

Resolución

```
select distinct emb.hin, emb.nombre, emb.descripcion
from embarcacion emb
inner join salida sal
    on emb.hin=sal.hin
left join salida s
    on emb.hin=s.hin
    and s.fecha_hora_regreso_real > s.fecha_hora_regreso_tentativo
where sal.fecha_hora_regreso_real ≤ sal.fecha_hora_regreso_tentativo
and s.fecha_hora_regreso_real is null
order by emb.nombre;
```

305.5 subquery/temps/variable/cte

Enunciado

305.5 - **Prioridad de mantenimiento de sector.** Listar los sectores con más cantidad de camas en mantenimiento que el promedio de la cantidad de camas en mantenimiento de cada sector. En el cálculo del promedio se deben tener en cuenta los sectores que podrían tener 0 camas en mantenimiento. Indicar código y nombre del sector y cantidad de camas en mantenimiento. Ordenar por cantidad de camas en mantenimiento descendente.

Resolución

```
##opcion a
with mant_sec as (
    select sec.codigo, sec.nombre
        , count(cama.numero) cant_camias_mant
    from sector sec
    left join cama
        on sec.codigo=cama.codigo_sector
        and cama.estado='mantenimiento'
    group by sec.codigo
)
select *
from mant_sec
where cant_camias_mant > (
```

```

        select avg(cant_camias_mant)
        from mant_sec
    )
order by cant_camias_mant desc;

##opcion b

select count(cama.numero) into @cant_camias
from sector sec
left join cama
    on sec.codigo=cama.codigo_sector
    and cama.estado='mantenimiento';

select count(*) into @cant
from sector;

select @cant_camias/@cant;

select sec.codigo, sec.nombre
    , count(cama.numero) cant_camias_mant
from sector sec
left join cama
    on sec.codigo=cama.codigo_sector
    and cama.estado='mantenimiento'
group by sec.codigo
having count(cama.numero)> @cant_camias/@cant;

```

305.6 transactions

Enunciado

305.6- ¿Dónde guardamos los Gundam? La empresa ha notado que muchos socios tienen canoas modelo RX, las mismas requieren un cuidado diferente y ha decidido tratarlas como un nuevo tipo de embarcación. Se requiere registrar el nuevo tipo con los siguientes datos:

Nombre: Canoa Gundam

Operación requerida: Manual

Luego cambiar a este tipo de embarcación las canoas cuyo nombre comience con 'RX-' y asignar a este nuevo tipo de embarcación los mismos sectores de almacenamiento que las canoas.

Resolución

```

begin;

select codigo into @canoa
from tipo_embarcacion
where nombre='canoa';

insert into tipo_embarcacion(nombre, operacion_requerida)
values ('Canoa Gundam', 'Manual');

select LAST_INSERT_ID() into @gundam;

update embarcacion
set codigo_tipo_embarcacion=@gundam
where codigo_tipo_embarcacion=@canoa
and nombre like 'rx-%';

insert into sector_tipo_embarcacion
select @gundam, codigo_sector
from sector_tipo_embarcacion
where codigo_tipo_embarcacion=@canoa;

commit;

```

305.7 routines

Enunciado

305.7- Recuperar contratos. La empresa quiere recuperar los contratos de almacenamiento que alguna vez tuvo. Se requiere una rutina "contratos_posibles" que reciba un número de socio como parámetro y liste las embarcaciones de dicho socio que no están actualmente almacenadas en la guardería.

Indicar hin y nombre de la embarcación y nombre del tipo de embarcación.

Utilizar el socio número 31 para probar la rutina (incluirlo en la respuesta)

Resolución

```

delimiter o|'|__

drop procedure if exists contratos_posibles o|'|__

create procedure contratos_posibles(in num_soc int unsigned)

```

```

reads sql data
begin
select e.hin, e.nombre
      , te.nombre
from embarcacion e
inner join tipo_embarcacion te
      on e.codigo_tipo_embarcacion=te.codigo
where e.numero_socio=num_soc
      and e.hin not in (
          select hin
          from embarcacion_cama ec
          where ec.fecha_hora_baja_contrato is null
      );
end o|_|_

call contratos_posibles(31);

```

Recuperatorio

REC.1 inner join

Enunciado

REC.1- Embarcaciones en sectores a reacondicionar. El sector “Mars” debe ser reacondicionado y para esto deberán moverse las embarcaciones actualmente almacenadas. Listar las embarcaciones que estén actualmente almacenadas en el sector “Mars”. Indicar número de cama, hin y nombre de la embarcación, número y nombre del propietario. Ordenar por número de cama ascendente.

Resolución

```

select ec.numero_cama
      ,e.hin, e.nombre
      ,soc.numero,soc.nombre
from sector s
inner join cama c
      on s.codigo=c.codigo_sector
inner join embarcacion_cama ec
      on c.codigo_sector=ec.codigo_sector
      and c.numero=ec.numero_cama
inner join embarcacion e
      on ec.hin=e.hin
inner join socio soc
      on e.numero_socio=soc.numero

```

```
where s.nombre='Mars'  
and (ec.fecha_hora_baja_contrato is null or  
ec.fecha_hora_baja_contrato > now())  
order by ec.numero_cama;
```

REC.2 left join

Enunciado

REC.2- Personas físicas que regresaron tarde en 2024. Listar las personas físicas (con tipo de documento dni) y sus embarcaciones aunque no hayan registrado salidas, y si realizaron salidas en las que regresaron tarde (el regreso real fue después del horario que habían estimado) este año indicarlas. Indicar número de documento, y nombre del socio; hin y nombre de la embarcación y para las salidas donde regresaron tarde la fecha y hora de salida, fecha y hora de regreso real y tentativo.

Resolución

```
select soc.nro_doc, soc.nombre  
      , e.hin, e.nombre  
      , sal.fecha_hora_salida, sal.fecha_hora_regreso_real,  
      sal.fecha_hora_regreso_tentativo  
from socio soc  
inner join embarcacion e  
      on soc.numero=e.numero_socio  
left join salida sal  
      on e.hin=sal.hin  
      and sal.fecha_hora_regreso_real > sal.fecha_hora_regreso_tentativo  
      and sal.fecha_hora_regreso_real ≥ '20240101'  
where soc.tipo_doc='dni';
```

REC.3 group by

Enunciado

REC.3- Canoas y Kayas con pocas salidas Listar las embarcaciones de tipo canoa o kayak que realizaron 10 o menos salidas. Indicar código y nombre del tipo de embarcación, hin y nombre de la embarcación, cantidad de salidas realizadas y fecha de su última salida. Si no realizó salidas debe indicarse cantidad 0 y en la fecha y hora de la última salida debe decir “sin salidas”.

Ordenar el resultado por cantidad de salidas descendente y hin ascendente.

Resolución

```
select te.codigo, te.nombre
      , e.hin, e.nombre
      , count(s.fecha_hora_salida) cantidad_salidas
      , COALESCE(max(fecha_hora_salida), 'sin salidas')
from embarcacion e
inner join tipo_embarcacion te
      on e.codigo_tipo_embarcacion=te.codigo
left join salida s
      on e.hin=s.hin
where te.nombre in ('canoa', 'kayak')
group by te.codigo, te.nombre,
e.hin, e.nombre
having cantidad_salidas ≤ 10
order by cantidad_salidas desc, hin;
```

REC.4 not in

Enunciado

REC.4- Instructores de windsurf libres. Debido a la demanda actual de cursos para windsurf se requiere un listado de instructores que pueden dictar actividades para el tipo de embarcación windsurf y no estén actualmente dictando ningún curso. Indicar legajo, nombre y apellido del instructor; nombre y descripción de las actividades que pueden dictar para embarcaciones de windsurf.

Resolución

```
select i.legajo, i.nombre, i.apellido
      , a.nombre, a.descripcion
from instructor i
inner join instructor_actividad ia
      on i.legajo=ia.legajo_instructor
inner join actividad a
      on ia.numero_actividad=a.numero
inner join tipo_embarcacion te
      on a.codigo_tipo_embarcacion=te.codigo
where te.nombre='Tabla Wind Surf'
```

```
and i.legajo not in (  
select c.legajo_instructor  
from curso c  
where current_date between c.fecha_inicio and c.fecha_fin);
```

REC.5 temp/var/cte/subquery

Enunciado

REC.5- Variación en salidas de cada embarcación. Listar todas las embarcaciones que hayan tenido menos salidas en un año que el promedio de la cantidad de salidas de embarcaciones para ese mismo año y tipo de embarcación. Indicar código y nombre del tipo de embarcación; año, promedio y cantidad de salidas de dicho año.

Resolución

```
with salidas_anuales as (  
    select e.codigo_tipo_embarcacion, e.hin, e.nombre,  
    year(s.fecha_hora_salida) anio, count(s.hin) cant_salidas  
    from embarcacion e  
    left join salida s  
        on e.hin=s.hin  
    group by e.hin, year(s.fecha_hora_salida)  
,  
prom as (  
    select te.codigo codigo_tipo, te.nombre tipo, sa.anio,  
    avg(cant_salidas) prom_salidas  
    from tipo_embarcacion te  
    left join salidas_anuales sa  
        on te.codigo=sa.codigo_tipo_embarcacion  
    group by te.codigo, sa.anio  
)  
select prom.codigo_tipo, prom.tipo  
    , prom.anio, prom.prom_salidas promedio  
    , sa.cant_salidas salidas  
from prom  
left join salidas_anuales sa  
    on prom.codigo_tipo=sa.codigo_tipo_embarcacion  
    and prom.anio=sa.anio  
where prom.prom_salidas > sa.cant_salidas;
```

REC.6 – transaction

Enunciado

REC.6- Nuevo tipo de embarcación: Catamarán Deportivo. La empresa ha decidido brindar actividades para catamarán deportivo. Se requiere agregar este nuevo tipo de embarcación que requiere operación automática. Asignarla a todos los sectores que tienen este tipo de operación. Además generar una copia de todas las actividades para veleros pero para este tipo de embarcación, agregando al nombre de la actividad la palabras “en catamarán” al final y con la misma descripción.

Resolución

```
begin;

insert into tipo_embarcacion (nombre, operacion_requerida)
values ('Catamarán Deportivo', 'Automática');

select LAST_INSERT_ID() into @cat;

insert into sector_tipo_embarcacion
select @cat, codigo
from sector
where tipo_operacion='Automático';

insert into actividad (nombre, descripcion, codigo_tipo_embarcacion)
select concat(a.nombre, ' en catamarán'), a.descripcion, @cat
from actividad a
inner join tipo_embarcacion te
on a.codigo_tipo_embarcacion=te.codigo
where te.nombre='Velero';

commit;
```

REC.7 routines

Enunciado

REC.7- Instructores posibles. La empresa necesita una forma simple de saber que instructores podrían dictar un nuevo curso. Para ello se necesita crear una rutina llamada posibles_instructores que reciba como parámetros un número de actividad y las fechas de inicio y fin del posible curso y devuelva la lista de instructores que puede dictar dicha actividad y no estén dictando ningún curso durante las fechas ingresadas. Indicar legajo, nombre, apellido y teléfono del

instructor. Utilizar la actividad número 15, y las fechas 1/12/2024 y 31/1/2025 para probar la rutina. Incluir la invocación en la entrega.

Resolución

```
delimiter (⦿.⦿)7

drop procedure if exists posibles_instructores (⦿.⦿)7

create procedure posibles_instructores(in act int, in since date, in
until date)
reads sql data
begin

select i.legajo, i.nombre, i.apellido, i.telefono
from instructor_actividad ia
inner join instructor i
    on ia.legajo_instructor=i.legajo
where ia.numero_actividad=act
and ia.legajo_instructor not in (
    select c.legajo_instructor
    from curso c
    where (since between c.fecha_inicio and c.fecha_fin)
        or (until between c.fecha_inicio and c.fecha_fin)
        or (since < c.fecha_inicio and until > c.fecha_fin)
);

end (⦿.⦿)7

delimiter ;

call posibles_instructores(15, '20241201', '20250131');
```

Globalizador

GLB.1 inner join

Enunciado

GLB.1- Contratos cortos de Lanchas y Veleros. Listar las Lanchas y Veleros que hayan tenido contratos que hayan finalizado y que hayan durado menos de 100 días. Indicar nombre del tipo de

embarcación, hin, nombre de la embarcación, nombre del socio y duración del contrato. Ordenar por tipo de embarcación alfabéticamente y por duración descendente.

Resolución

```
select te.nombre, e.hin, e.nombre, s.nombre
      ,TIMESTAMPDIFF(day, ec.fecha_hora_contrato,
ec.fecha_hora_baja_contrato) duracion
from embarcacion e
inner join embarcacion_cama ec
      on e.hin=ec.hin
inner join tipo_embarcacion te
      on e.codigo_tipo_embarcacion=te.codigo
inner join socio s
      on e.numero_socio=s.numero
where ec.fecha_hora_baja_contrato is not null
      and te.nombre in ('Velero', 'Lancha')
      and TIMESTAMPDIFF(day, ec.fecha_hora_contrato,
ec.fecha_hora_baja_contrato) <100
order by te.nombre, duracion desc;
```

GLB.2 – left join

Enunciado

GLB.2- Socios y cursos de 2024. Listar todos los socios y para cada uno los cursos a los que se inscribieron desde el 1/10/2024. Indicando número y nombre de socio, número de curso, fecha y hora de inscripción, número y nombre de la actividad. En caso que no haya realizado cursos mostrar 'sin cursos realizados' en los datos del curso y la actividad Ordenar por nombre de la actividad y nombre del socio alfabéticamente.

Resolución

```
select s.numero nro_socio ,s.nombre socio
      , c.numero nro_numero, i.fecha_hora_inscripcion
      , a.numero nro_actividad, a.nombre actividad
from socio s
left join inscripcion i
      on s.numero=i.numero_socio
      and i.fecha_hora_inscripcion > '20241001'
left join curso c
```

```
on i.numero_curso=c.numero
left join actividad a
on c.numero_actividad=a.numero
order by a.nombre, s.nombre;
```

GLB.3 – group by

Enunciado

GLB.3- Socios con múltiples embarcaciones. Listar los socios que hayan contratado camas para 2 o más embarcaciones distintas antes de 2024. Indicar número y nombre del socio, cantidad de embarcaciones distintas que almacenó, cantidad de tipos de embarcaciones distintas, cantidad de contratos que hizo en total y fecha de inicio del primer contrato.

Resolución

```
select s.numero nro_socio, s.nombre socio
, count(distinct te.codigo) cant_tipo_emb, count(distinct e.hin)
cant_emb
, count(*) cant_contratos, min(ec.fecha_hora_contrato)
fecha_ult_contrato
from socio s
inner join embarcacion e
on s.numero=e.numero_socio
inner join embarcacion_cama ec
on e.hin=ec.hin
inner join tipo_embarcacion te
on e.codigo_tipo_embarcacion=te.codigo
where ec.fecha_hora_contrato < '20240101'
group by s.numero
having cant_emb > 1;
```

GLB.4 – not in

Enunciado

GLB.4- Reemplazantes para cursos. Se recibieron quejas sobre el instructor en los cursos 29, 33 y 36. Se necesita listar para cada curso los posibles instructores para reemplazarlos según las actividades que cada uno puede dictar. Además la empresa quiere

que los instructores reemplazantes no estén dictando actualmente ningún curso.

Resolución

```
select c.*, ia.legajo_instructor
from curso c
inner join instructor_actividad ia
  on c.numero_actividad=ia.numero_actividad
  and c.legajo_instructor <> ia.legajo_instructor
where c.numero in (29,33, 36)
  and ia.legajo_instructor not in (
  select ci.legajo_instructor
  from curso ci
  where now() between ci.fecha_inicio and ci.fecha_fin
  );
```

##además de la forma con una diferencia por instructor al final también se puede hacer joinando en la subquery de inst con inst_act que no estén haciendo cursos y haciendo diferencia y joinando con eso

```
-- Otra forma
-- en el primer in traigo todas las actividades de los cursos
encuestion
-- luego resto los instructores que estan ocupados
```

```
select distinct inst.legajo, nombre, apellido
from instructor inst
inner join instructor_actividad ia
  on inst.legajo=ia.legajo_instructor #faltaba
where ia.numero_actividad in (
select distinct c.numero_actividad
from curso c
  inner join instructor_actividad i_a #cambie alias para joinar en where
  on c.numero_actividad=i_a.numero_actividad
  where c.numero in (29,33, 36)
  and ia.numero_actividad=i_a.numero_actividad)
  and ia.legajo_instructor not in (
select cur.legajo_instructor
from curso cur
where now() between cur.fecha_inicio and cur.fecha_fin);
```

GLB.5 - subquery/tt/cte/var

Enunciado

GLB.5- Socios tardíos. Listar los socios que tengan más cantidad de regresos tardíos (hora de regreso real es de la hora tentativa que informaron) que la cantidad de regresos en horario. Indicando número, nombre del socio, cantidad de regresos tarde, cantidad de regresos a horario y la proporción de cada uno respecto del total de regresos.

Resolución

```
-- simplest
with reg_tarde as (
    select so.numero numero_socio, so.nombre socio
           , count(*) cant_tarde
    from socio so
    inner join embarcacion e
        on so.numero=e.numero_socio
    inner join salida sal
        on e.hin=sal.hin
    where sal.fecha_hora_regreso_real > sal.fecha_hora_regreso_tentativo
    group by so.numero
)
select rt.*
       , count(sa.fecha_hora_regreso_real) cant_a_tiempo
       , rt.cant_tarde/(count(sa.fecha_hora_regreso_real)+cant_tarde)*100
prop_tarde
       ,
count(sa.fecha_hora_regreso_real)/(count(sa.fecha_hora_regreso_real)+cant_tarde)*100 prop_a_tiempo
from reg_tarde rt
left join embarcacion em
    on rt.numero_socio=em.numero_socio
left join salida sa
    on em.hin=sal.hin
    and sa.fecha_hora_regreso_real ≤ sa.fecha_hora_regreso_tentativo
group by rt.numero_socio
having cant_tarde>cant_a_tiempo;
```

```
-- harder
select so.numero numero_socio, so.nombre socio
       , sum(sa.fecha_hora_regreso_real > sa.fecha_hora_regreso_tentativo)
cant_tarde
       , sum(sa.fecha_hora_regreso_real ≤ sa.fecha_hora_regreso_tentativo)
cant_a_tiempo
       , sum(sa.fecha_hora_regreso_real > sa.fecha_hora_regreso_tentativo)
/ (sum(sa.fecha_hora_regreso_real
```

```

≤ sa.fecha_hora_regreso_tentativo)+sum(sa.fecha_hora_regreso_real >
sa.fecha_hora_regreso_tentativo))*100 prop_tarde
    , sum(sa.fecha_hora_regreso_real ≤ sa.fecha_hora_regreso_tentativo)
/ (sum(sa.fecha_hora_regreso_real
≤ sa.fecha_hora_regreso_tentativo)+sum(sa.fecha_hora_regreso_real >
sa.fecha_hora_regreso_tentativo))*100 prop_a_tiempo
from socio so
inner join embarcacion e
    on so.numero=e.numero_socio
inner join salida sa
    on e.hin=sa.hin
group by so.numero
having cant_tarde > cant_a_tiempo;

```

```

-- intermediate
with cants as (
    select so.numero numero_socio, so.nombre socio
        , sum(sa.fecha_hora_regreso_real >
sa.fecha_hora_regreso_tentativo) cant_tarde
        , sum(sa.fecha_hora_regreso_real
≤ sa.fecha_hora_regreso_tentativo) cant_a_tiempo
    from socio so
    inner join embarcacion e
        on so.numero=e.numero_socio
    inner join salida sa
        on e.hin=sa.hin
    group by so.numero
    having cant_tarde > cant_a_tiempo
)
select cants.*
    , cant_tarde/(cant_tarde+cant_a_tiempo)*100 prop_tarde
    , cant_a_tiempo/(cant_tarde+cant_a_tiempo)*100 prop_a_tiempo
from cants;

```

GLB.6 – transac

Enunciado

GLB.6- Nuevas incorporaciones.La empresa ha contratado a un nuevo instructor, darlo de alta con los datos a continuación:

Legajo: 20

CUIL: 20-01234567-9

Nombre: Norrin

Apellido: Radd

Teléfono: 555-0120

Además habilitarlo para dictar todas las actividades que se realizan para embarcaciones de tipo Tabla Wind Surf, Tabla Kite y No convencional.

Asegurarse que los cambios sean consistentes.

Resolución

```
begin;
```

```
insert into instructor values(20,'20-01234567-9','Norrin', 'Radd','555-0120');
```

```
insert into instructor_actividad
select 20, a.numero
from actividad a
inner join tipo_embarcacion te
    on a.codigo_tipo_embarcacion=te.codigo
and te.nombre in ('Table Wind Surf', 'Tabla Kite', 'No convencional');

commit;
```

GLB.7 – routines

Enunciado

GLB.7- Horarios actuales. Crear un procedimiento almacenado `horarios_instructor` que recibe de parámetro el legajo de un instructor y una fecha e informe los días y horarios de dictado de los cursos que se estén dictando durante esas fechas. Indicar número y nombre de actividad; número, fecha de inicio y fin del curso; día de la semana, hora de inicio y fin de dictado. Probar los datos con el legajo 12 y la fecha 1 de Oct de 2024
Incluir la invocación del procedimiento en la resolución

Resolución

```
delimiter (#_#)_
```

```
drop procedure if exists horarios_instructor(#_#)_
```

```
create procedure horarios_instructor(in ins int unsigned, in fecha date)
reads sql data
begin
    select a.numero nro_act, a.nombre actividad
    , c.numero, c.fecha_inicio, c.fecha_fin
    , dc.dia_semana, dc.hora_inicio, dc.hora_fin
from curso c
inner join dictado_curso dc
    on c.numero=dc.numero_curso
inner join actividad a
    on c.numero_actividad=a.numero
where c.legajo_instructor=ins
    and fecha between c.fecha_inicio and c.fecha_fin;
end (#_#)#

delimiter ;

call horarios_instructor(12, '20241001')
```