

Convenciones Underground – Parciales SQL

links 3

301 3

1. inner join 3
2. left/right join 4
3. group by/having 5
4. difference 6
5. subcon/TT/CTE/vars 7
6. transac 8
7. routines 9

302 10

1. inner join 10
2. left/right join 11
3. group by/having 11
4. difference 12
5. subcon/TT/CTE/vars 13
6. transac 14
7. routines 15

303 17

1. inner join 17
2. left/right join 18

- 3. group by/having 18
- 4. difference 19
- 5. subcon/TT/CTE/vars 20
- 6. transac 21
- 7. routines 22

304 24

- 1. Inner join 24
- 2. left/right join 25
- 3. group by/having 25
- 4. difference 26
- 5. subcon/TT/CTE/vars 27
- 6. transac 28
- 7. routines 29

305 31

- 1. Inner join 31
- 2. left/right join 32
- 3. group by/having 32
- 4. difference 34
- 5. subcon/TT/CTE/vars 35
- 6. transac 36
- 7. routines 37

Recuperatorio 39

- 1. inner join 39

2. left/right join 40
3. group by/having 41
4. difference 41
5. subcon/TT/CTE/vars 42
6. transac 43
7. routines 44

Globalizador 45

1. inner join 45
2. left/right join 46
3. group by/having 46
4. difference 46
5. subcon/TT/CTE/vars 47
6. transac 47
7. routines 48

Anexo datos transacciones 48

links

[Enunciado MD](#)

[Resolución MD](#)

[DDL](#)

301

1. inner join

Enunciado

301.1-Presentadores que aparecieron en Orth. Listar los presentadores de alguna presentación realizada en locaciones de zona de "Orth". Indicar el id, nombre, apellido y denominación del presentador; nombre, tipo de presentación y fecha y hora de inicio. Ordenar ascendente por cuit del presentador y fecha de inicio de la presentación descendente.

Resolución

```
# no es necesario usar coalesce pero no estaría mal
select p.id, p.nombre, p.apellido, p.denominacion
      , pre.nombre, pre.tipo, pre.fecha_hora_ini
from presentador p
inner join presentador_presentacion pp
      on p.id=pp.id_presentador
inner join presentacion pre
      on pre.id_locacion=pp.id_locacion
      and pre.nro_sala=pp.nro_sala
      and pre.fecha_hora_ini=pp.fecha_hora_ini
inner join locacion loc
      on loc.id=pp.id_locacion
where loc.zona='Orth'
order by p.cuit, pre.fecha_hora_ini desc;
```

2. left/right join

Enunciado

301.2-Supervisores y si fueron coordinadores en eventos. Listar los empleados con categoría supervisor y, solo para los que hayan ocupado el rol de “coordinador” para algún evento, mostrar el/los eventos para el que lo ocuparon, el resto mostrar null. Indicar cuil, nombre y apellido del empleado; id, nombre y tipo del evento.

Resolucion

```
select emp.cuil, emp.nombre, emp.apellido
      , ev.id, ev.nombre, ev.tipo
from empleado emp
left join encargado_evento ee
```

```
on emp.cuil=ee.cuil_encargado
and ee.rol='coordinador'
left join evento ev
on ee.id_evento=ev.id
where emp.categoria='supervisor'
```

3. group by/having

Enunciado

301.3-Clientes que alquien stands dobles. Listar los clientes que hayan alquilado al menos 3 stands del tipo "doble". Indicar cuit, razón social, categoría e email del cliente; cantidad de stands dobles alquilados, promedio del valor acordado en el contrato y diferencia entre el máximo y mínimo valor acordado.

Resolución

```
select cli.cuit, cli.razon_social, cli.categoria, cli.email
      , count(*) cant_stands, avg(st.valor_acordado) prom,
max(st.valor_acordado)-min(st.valor_acordado) diferencia
from cliente cli
inner join stand st
      on cli.cuit=st.cuit_cliente
where st.tipo='doble'
group by cli.cuit, cli.razon_social, cli.categoria, cli.email
having count(*)>3
```

4. difference

Enunciado

301.4- Supervisores que no hayan coordinado convenciones Listar los empleados con categoría supervisor que nunca hayan ocupado el rol de coordinador en un evento de tipo convención. Indicar cuil, nombre, apellido, e email del empleado.

Resolución

```
select emp.cuil, emp.nombre, emp.apellido, emp.email
from empleado emp
```

```

where emp.categoria='supervisor'
and not exists (
    select 1
    from encargado_evento ee
    inner join evento ev
        on ee.id_evento=ev.id
    where ee.cuil_encargado=emp.cuil
    and ee.rol='coordinador'
    and ev.tipo='convencion'
);

```

```

select emp.cuil, emp.nombre, emp.apellido, emp.email
from empleado emp
where emp.categoria='supervisor'
and emp.cuil not in (
    select ee.cuil_encargado
    from encargado_evento ee
    inner join evento ev
        on ee.id_evento=ev.id
    where ee.rol='coordinador'
    and ev.tipo='convencion'
);

```

5. subcon/TT/CTE/vars

Enunciado

301.5- Temáticas exitosas. Listar las temáticas para las que se vendieron más entradas (en total en todos los eventos) este año (2023) que el año pasado (2022) para la misma temática. En caso que el año pasado no se hayan vendido pero este año si debe mostrarse la temática también y asumir la cantidad del año pasado como 0. Indicar temática, cantidad vendida el año pasado, cantidad vendida este año y la diferencia y cuantos compradores distintos las adquirieron este año.

Resolución

```

with tema2022 as (
    select ev.tematica, count(*) cant
    from evento ev
    inner join entrada ent
        on ev.id=ent.id_evento

```

```

where ent.fecha_hora_venta between '20220101' and '20221231 235959'
group by ev.tematica
)
select ev.tematica, count(*) cant2023
      , coalesce(tema2022.cant,0) cant2022, count(*) -
coalesce(tema2022.cant,0) diff
      , count(distinct ent.id_comprador) compradores
from evento ev
inner join entrada ent
      on ev.id=ent.id_evento
left join tema2022
      on ev.tematica=tema2022.tematica
where ent.fecha_hora_venta between '20230101' and '20231231 235959'
group by ev.tematica, tema2022.cant
having cant2022 < cant2023

```

6. transac

Enunciado

301.6- **Mantenimiento de Rouge.** Se terminó de acondicionar una nueva sala "Blanc" mientras se da mantenimiento a la sala "Rouge". Registrar la nueva sala Blanc que terminó de acondicionarse con los siguientes datos, pasar la sala Rouge a estado "en mantenimiento" y reasignar las presentaciones de noviembre de la sala Rouge a Blanc.

Sala:

nombre: Blanc

ubicación: Rimuru (11001)

Número de sala: 3

metros cuadrados: 330

capacidad: 300

estado: habilitada

Resolución

```

begin;

select id_locacion, nro into @locRouge, @nroRouge
from sala
where nombre='Rouge';

insert into sala values(11001, 3, 'Blanc', 330, 300,'habilitada');

```

```

update sala set estado='en mantenimiento'
where id_locacion=@locRouge and nro=@nroRouge;

update presentacion
set id_locacion=11001, nro_sala=3
where fecha_hora_ini between '2023-11-01' and '2023-11-30 23:59:59'
and id_locacion=@locRouge and nro_sala=@nroRouge;

commit;

```

7. routines

Enunciado

301.7- Cree una función **recaudacion_tematica** que dada una temática calcule el ingreso total por evento de esa temática (a través del costo de las entradas) y devuelva el promedio de los mismos. Luego listar los eventos cuya recaudación total supere el promedio de su temática, utilizando dicha función. Indique nombre y temática del evento, recaudación total del mismo, recaudación promedio de la temática y diferencia.

Resolución

```

## funcion
delimiter @@

drop function if exists recaudacion_tematica@@

create function recaudacion_tematica(tema varchar(255))
returns decimal(9,3)recaudacion_tematicaacc
reads sql data
begin
    declare prom decimal(9,3);
    with rec as (
        select ev.id, sum(ent.costos) total_ev
        from evento ev
        inner join entrada ent
            on ev.id=ent.id_evento
        where ev.tematica=tema
        group by ev.id
    )

```



```

        select avg(total_ev) into prom from rec ;
    return prom;
end&&

delimiter ;

## uso
select ev.nombre, ev.tematica
    , sum(ent.costos) total_ev
    , recaudacion_tematica(ev.tematica)
    , sum(ent.costos)-recaudacion_tematica(ev.tematica) diff
from evento ev
inner join entrada ent
    on ev.id=ent.id_evento
group by ev.id
having diff >0

```

302

1. inner join

Enunciado

302.1-Locaciones y salas de los eventos de marzo de 2022. Listar las distintas locaciones donde se realizaron eventos durante marzo de 2022 según la fecha desde. Indicar nombre, dirección y zona de la locación; nro y nombre de la sala. Ordenar por zona y nombre de la locación alfabéticamente y descendente por m2 de la sala.

Resolución

```

select loc.nombre, loc.direccion, loc.zona
    , sala.nro, sala.nombre
from evento ev
inner join presentacion pre
    on ev.id=pre.id_evento
inner join sala
    on sala.id_locacion=pre.id_locacion
    and sala.nro=pre.nro_sala
inner join locacion loc
    on sala.id_locacion=loc.id

```

```
where ev.fecha_desde between '20230301' and '20230401'  
order by loc.zona, loc.nombre, sala.m2 desc;
```

2. left/right join

Enunciado

302.2-Gold Sponsors y si contrataron stands. Listar los clientes de categoría "Gold Sponsor" y, si alguna vez contrataron un stand en la ubicación "área principal" en 2023, mostrar para que evento. Indicar cuit y razón social del cliente; id, nombre y temática del evento.

Resolución

```
select cli.cuit, cli.razon_social  
      , ev.id, ev.nombre, ev.tematica  
from cliente cli  
left join stand st  
      on cli.cuit=st.cuit_cliente  
      and st.ubicacion='área principal'  
      and st.fecha_contrato >= '20230101' and st.fecha_contrato <  
'20240101'  
left join evento ev  
      on st.id_evento=ev.id  
where cli.categoria='Gold Sponsor';
```

3. group by/having

Enunciado

302.3-Participación de empleados en convenciones de seguro. Listar los empleados que hayan participado en al menos 4 eventos de tipo "convención de seguros". Indicar cuil, nombre y apellido del empleado; promedio del valor base de la entrada de los eventos y diferencia entre el máximo y mínimo valor base de la entrada y cantidad de eventos del tipo "convención de seguros".

Resolución

```

select emp.cuil, emp.nombre, emp.apellido
      , avg(ev.valor_base_entrada) prom
      , max(ev.valor_base_entrada)-min(ev.valor_base_entrada) diferencia
      , count(*) cant_stands
from empleado emp
inner join encargado_evento ee
      on emp.cuil=ee.cuil_encargado
inner join evento ev
      on ev.id=ee.id_evento
where ev.tipo="convencion de seguros"
group by emp.cuil, emp.nombre, emp.apellido
having count(*)>4

```

4. difference

Enunciado

302.4- **Eventos sin un marketing adecuado.** Listar los eventos que hayan comenzado este año y no hayan tenido un encargado con categoría senior asignado con el rol marketing. Indicar id, nombre, tipo, temática y fecha desde.

Resolución

```

select ev.id, ev.nombre, ev.tipo, ev.tematica, ev.fecha_desde
from evento ev
where ev.fecha_desde BETWEEN '20230101' and '20231231 235959'
      and not exists (
          select 1
          from encargado_evento ee
          inner join empleado emp
                on ee.cuil_encargado=emp.cuil
          where ev.id=ee.id_evento
                and emp.categoria='senior'
                and ee.rol='marketing'
      )

```

```

select ev.id, ev.nombre, ev.tipo, ev.tematica, ev.fecha_desde
from evento ev
where ev.fecha_desde BETWEEN '20230101' and '20231231 235959'
      and ev.id not in (
          select ee.id_evento
          from encargado_evento ee

```

```

        inner join empleado emp
        on ee.cuil_encargado=emp.cuil
        where emp.categoria='senior'
        and ee.rol='marketing'
    )

```

5. subcon/TT/CTE/vars

Enunciado

302.5- **Zonas frecuentes.** Listar las zonas en las que se realizaron más presentaciones este mes (octubre) que el mes pasado (septiembre). En el caso que en alguna zona no se hayan realizado presentaciones el mes pasado pero este sí se deberán mostrar y considerar la cantidad del mes pasado como 0. Indicar zona, cantidad de presentaciones el mes pasado y cantidad de presentaciones de este mes, la diferencia entre ambas y la cantidad de eventos distintos de dichas presentaciones de este año.

Resolución

```

with zonaSep as (
    select loc.zona, count(*) cant
    from locacion loc
    inner join presentacion pre
        on loc.id=pre.id_locacion
    where pre.fecha_hora_ini between '20230901' and '20230930 235959'
    group by loc.zona
)
select loc.zona, count(*) cantOct
    , coalesce(zonaSep.cant,0) cantSep, count(*) -
    coalesce(zonaSep.cant,0) diff
    , count(distinct pre.id_evento) eventos
from locacion loc
inner join presentacion pre
    on loc.id=pre.id_locacion
left join zonaSep
    on loc.zona=zonaSep.zona
where pre.fecha_hora_ini between '20231001' and '20231030 235959'
group by loc.zona
having cantOct > cantSep

```

6. transac

Enunciado

302.6- **Farmeras absorbe Falmuth.** La empresa cliente Falmuth fue comprada y absorbida por Farmeras. Se debe ingresar al sistema la nueva empresa con los datos a continuación y migrar todos los stands contratados para eventos futuros por Falmuth a Farmeras.

Cliente:

cuit: 15151515151

razón social: Farmeras

teléfono: 1515151515

dirección: 44 Arwenack Street

email: farmeras@tairiku.ma

categoría: Premium

Resolución

```
select '15151515151' into @farmeras;

select cuit into @falmuth
from cliente
where razon_social='Falmuth';

insert into cliente values(@farmeras, 'Farmeras', '1515151515', '44
Arwenack Street', 'farmeras@tairiku.ma', 'Premium');

update stand set cuit_cliente=@farmeras
where cuit_cliente=@falmuth
and id_evento in (
    select id
    from evento
    where fecha_desde ≥ now() #o fecha del día
);

commit;
```

7. routines

Enunciado

302.7- Cree una función **presentaciones_zona** que dada una zona calcule la cantidad de presentaciones por sala de dicha zona y devuelva el promedio de dicha cantidad. Luego listar las salas cuya cantidad de presentaciones supere el promedio de su zona, utilizando dicha función. Indique nombre de la sala, zona a donde pertenece, cantidad de presentaciones de la sala, promedio de la zona y diferencia de los mismos.

Resolución

```
# funcion
delimiter $$

drop function if exists presentaciones_zona$$

CREATE FUNCTION `presentaciones_zona`(zone varchar(255)) RETURNS
decimal(9,3)
    READS SQL DATA
begin
    declare prom decimal(9,3);

    with rec as (
        select count(*) cant
        from locacion loc
        inner join presentacion pre
            on loc.id=pre.id_locacion
        where loc.zona=zone
        group by pre.id_locacion, pre.nro_sala
    )
    select avg(cant) into prom from rec;

    return prom;
end$$

delimiter ;

# uso
select sala.nombre, count(*) cant
    , presentaciones_zona(loc.zona), count(*) -
presentaciones_zona(loc.zona) dif
from locacion loc
inner join sala
    on loc.id=sala.id_locacion
inner join presentacion pre
    on sala.id_locacion=pre.id_locacion
```

```
and sala.nro=pre.nro_sala
group by pre.id_locacion, pre.nro_sala
having dif>0;
```

303

1. inner join

Enunciado

303.1-Asistentes 2023. Listar los asistentes a las presentaciones realizadas en 2023 (por fecha y hora de inicio). Indicar los id, nombre y apellido del asistente; nro de entrada; nombre, tipo y fecha de inicio de la presentación. Ordenar por apellido y nombre de la persona alfabéticamente y descendente por fecha y hora de venta.

Resolución

```
select per.id, per.nombre, per.apellido
      , ent.nro
      , pre.nombre, pre.tipo, pre.fecha_hora_fin
from persona per
inner join entrada ent
      on ent.id_asistente=per.id
inner join presentacion_entrada pe
      on pe.id_evento=ent.id_evento
      and pe.nro_entrada=ent.nro
inner join presentacion pre
      on pre.id_locacion=pe.id_locacion
      and pre.nro_sala=pe.nro_sala
      and pre.fecha_hora_ini=pe.fecha_hora_ini
where pre.fecha_hora_ini ≥ '20230101' and pre.fecha_hora_ini <
'20240101'
#opcion between: #where pre.fecha_hora_ini between '20230101' and
'20231231 235959'
order by per.apellido, per.nombre, ent.fecha_hora_venta desc;
```

2. left/right join

Enunciado

303.2-Salas grandes y sus presentaciones. Listar las salas de más de 100 m2 y, si tuvieron presentaciones en 2023 (según fecha y hora de inicio), mostrar los datos de la presentación y evento. Indicar nro, nombre y m2 de sala; nombre y fecha hora de inicio de la presentación; id, nombre y tipo de evento.

Resolucion

```
select sala.nro, sala.nombre, sala.m2
      , pre.nombre, pre.fecha_hora_ini
      , ev.id, ev.nombre
from sala
left join presentacion pre
      on sala.id_locacion=pre.id_locacion
      and sala.nro=pre.nro_sala
      and pre.fecha_hora_ini ≥ '20230101' and pre.fecha_hora_ini <
'20240101'
left join evento ev
      on pre.id_evento=ev.id
where sala.m2 > 100
```

3. group by/having

Enunciado

303.3-Compradores asiduos. Listar las personas que hayan comprado al menos 10 entradas desde el año 2020 para eventos. Indicar id, nombre, apellido y email de la persona; promedio del costo de las entradas y cuántos días pasaron desde la compra de la primera entrada a la última y cantidad de entradas compradas desde el 2020.

Resolución

```
select per.id, per.nombre, per.apellido, per.email
      , count(*) cant, avg(ent.costo) prom, max(ent.fecha_hora_venta) -
min(ent.fecha_hora_venta) dif
from persona per
inner join entrada ent
      on per.id=ent.id_comprador
where ent.fecha_hora_venta ≥ '20200101'
```



```
group by per.id, per.nombre, per.apellido, per.email  
having count(*) ≥ 10;
```

4. difference

Enunciado

303.4- **Gold sponsors sin stand dobles para Abyss.** Listar los clientes con categoría "Gold Sponsor" que nunca hayan alquilado un stand de tipo doble para un evento con temática "Abyss". Indicar cuit, razón social e email.

Resolución

```
select cli.cuit, cli.razon_social, cli.email  
from cliente cli  
where cli.categoria='Gold Sponsor'  
and not exists (  
    select 1  
    from stand st  
    inner join evento ev  
        on st.id_evento=ev.id  
    where cli.cuit=st.cuit_cliente  
        and ev.tematica='Abyss'  
        and st.tipo='doble'  
);
```

```
select cli.cuit, cli.razon_social, cli.email  
from cliente cli  
where cli.categoria='Gold Sponsor'  
and cli.cuit not in (  
    select st.cuit_cliente  
    from stand st  
    inner join evento ev  
        on st.id_evento=ev.id  
    where ev.tematica='Abyss'  
        and st.tipo='doble'  
);
```

5. subcon/TT/CTE/vars

Enunciado

303.5- **Especialidades más requeridas.** Listar las especialidades de los panelistas para las que se realizaron más presentaciones este año (2023) que el año pasado (2022). En caso que el año pasado no se hayan realizado pero este año si debe mostrarse la especialidad también y asumir la cantidad del año pasado como 0. Indicar la especialidad, la cantidad realizada este año, la cantidad realizada el año pasado, la diferencia entre ambas y en cuántas locaciones distintas se presentaron este año.

Resolución

```
with esp2022 as (  
    select pdor.especialidad, count(*) cant  
    from presentador pdor  
    inner join presentador_presentacion pp  
        on pdor.id=pp.id_presentador  
    where pp.fecha_hora_ini between '20220101' and '20221231 235959'  
        and pdor.especialidad is not null  
    group by pdor.especialidad  
)  
select pdor.especialidad, count(*) cant2023  
    , coalesce(esp2022.cant,0) cant2022, count(*) -  
coalesce(esp2022.cant,0) diff  
    , count(distinct pre.id_locacion) locaciones  
from presentador pdor  
inner join presentador_presentacion pp  
    on pdor.id=pp.id_presentador  
inner join presentacion pre  
    on pp.id_locacion=pre.id_locacion  
    and pp.nro_sala=pre.nro_sala  
    and pp.fecha_hora_ini=pre.fecha_hora_ini  
left join esp2022  
    on pdor.especialidad=esp2022.especialidad  
where pre.fecha_hora_ini between '20230101' and '20231231 235959'  
    and pdor.especialidad is not null  
group by pdor.especialidad, esp2022.cant  
having cant2023>cant2022
```

6. transac

Enunciado

303.6- Rimuru reemplaza a Clayman. El supervisor Clayman Clown dejará de supervisar eventos, para reemplazarlo se ha contratado un nuevo empleado: Rimuru Tempest. Darlo de alta en el sistema con los datos que figuran a continuación y asignarlo, con rol de supervisor, a todos los futuros eventos donde Clayman debería ser supervisor.

Nuevo empleado:

cuil: 24242424242

nombre: Rimuru

apellido: Tempest

telefono: 242424242

direccion: Hidden Cave 242

email: rimuru@jurashinrin.ma

fecha de nacimiento: 1992-04-24

categoria: daimaou

Resolución

```
begin;

select cuil into @clayman
from empleado
where nombre='Clayman' and apellido='Clown';

insert into empleado values(24242424242, 'Rimuru',
'Tempest','242424242','Hidden Cave 242','rimuru@jurashinrin.ma','1992-
04-24','daimaou');

update encargado_evento set cuil_encargado = 24242424242
where cuil_encargado=@clayman
and rol='supervisor'
and id_evento in (
    select id
    from evento
    where fecha_desde ≥ now()
);

commit;
```

7. routines

Enunciado

303.7- Cree una función **antig_cliente** que dado una categoría de cliente calcule el tiempo que llevan siendo cliente (diferencia entre primer y última fecha que contrató un stand en días) cada cliente de esa categoría y devuelva el promedio de la categoría. Luego calcule el tiempo que cada cliente lleva siéndolo y muestre quienes llevan más tiempo que el promedio de su categoría, utilizando dicha función. Indique la razón social y categoría del cliente, el tiempo que lleva siendo cliente, el tiempo promedio de la categoría y la diferencia entre ambas.

Resolución

```
delimiter (ツ)

drop function if exists antig_cliente (ツ)

create function antig_cliente(categ varchar(255))
returns float #decimal(9,3)
reads sql data
begin
    declare prom decimal(9,3);

    with tiempos as (
        select max(fecha_contrato)-min(fecha_contrato) dias_cliente
        from stand
        inner join cliente cli
        on stand.cuit_cliente=cli.cuit
        where cli.categoria=categ
        group by stand.cuit_cliente
    )
    select avg(dias_cliente) into prom from tiempos;

    return prom;
end(ツ)

delimiter ;

#uso
select cli.razon_social, cli.categoria
, max(fecha_contrato)-min(fecha_contrato) dias_cliente
, antig_cliente(cli.categoria)
, max(fecha_contrato)-min(fecha_contrato) -
antig_cliente(cli.categoria) dif
from stand
```

```
inner join cliente cli
on stand.cuit_cliente=cli.cuit
group by stand.cuit_cliente, cli.categoria
having dif>0;
```

304

1. Inner join

Enunciado

304.1-Locaciones de presentaciones de tipo "role play". Listar todas las locaciones donde se realizaron presentaciones de tipo "role play". Indicar nombre, descripción y fechas de la presentación; nombre y dirección de la locación; nombre de sala; id, tipo y nombre del evento. Ordenar por fecha de inicio del evento descendente y nombre del evento alfabético.

Resolución

```
select pre.nombre, pre.descripcion, pre.fecha_hora_ini,
pre.fecha_hora_fin
      , loc.nombre, loc.direccion
      , sala.nombre
      , ev.id, ev.tipo, ev.nombre
from presentacion pre
inner join sala
      on pre.id_locacion=sala.id_locacion
      and pre.nro_sala=sala.nro
inner join locacion loc
      on sala.id_locacion=loc.id
inner join evento ev
      on pre.id_evento=ev.id
where pre.tipo='role play'
order by ev.fecha_desde desc, ev.nombre;
```

2. left/right join

Enunciado

304.2-Eventos de Star Wars 2022. Listar los eventos con temática "Star Wars" y si tuvieron presentaciones en 2022 (según la fecha y hora de inicio), mostrar datos de las presentaciones y las locaciones. Indicar id y nombre del evento; nombre, y fecha y hora de inicio de la presentación; nombre y dirección de la locación.

Resolución

```
select ev.id, ev.nombre
      , pre.nombre, pre.fecha_hora_ini
      , loc.nombre, loc.direccion
from evento ev
left join presentacion pre
      on ev.id=pre.id_evento
      and pre.fecha_hora_ini ≥ '20220101' and pre.fecha_hora_ini <
'20230101'
left join locacion loc
      on pre.id_locacion=loc.id
where ev.tematica='Star Wars';
```

3. group by/having

Enunciado

304.3-Salas show room. Listar las salas en las que se hayan realizado al menos 7 presentaciones de tipo "show". Indicar nombre de la sala; promedio del costo de la entrada (de la presentación) y diferencia entre el máximo y mínimo costo y cantidad de presentaciones "show" realizadas.

Resolución

```
select sala.nombre
      , avg(pre.costo_entrada) prom, max(pre.costo_entrada)-
min(pre.costo_entrada) dif, count(*) cant
from sala
inner join presentacion pre
      on sala.id_locacion=pre.id_locacion
      and sala.nro=pre.nro_sala
where pre.tipo='show'
```

```
group by sala.id_locacion, sala.nro, sala.nombre  
having count(*) ≥ 7;
```

4. difference

Enunciado

304.4- **Eventos de Star Wars sin sponsors en área principal.**

Listar los eventos con temática "Star Wars" que no hayan tenido ningún stand ubicado en el área principal que fuera alquilado por un cliente en alguna categoría de sponsor (por ejemplo Platinum Sponsor, Gold Sponsor, etc). Indicar el id, nombre, tipo y fechas del evento.

Resolución

```
select ev.id, ev.nombre, ev.tipo, ev.fecha_desde, ev.fecha_hasta  
from evento ev  
where ev.tematica='Star Wars'  
      and not exists (  
        select 1  
        from stand st  
        inner join cliente cli  
        where ev.id=st.id_evento  
              and st.ubicacion='area principal'  
              and cli.categoria like '%sponsor%'  
      );
```

```
select ev.id, ev.nombre, ev.tipo, ev.fecha_desde, ev.fecha_hasta  
from evento ev  
where ev.tematica='Star Wars'  
      and ev.id not in (  
        select st.id_evento  
        from stand st  
        inner join cliente cli  
        on st.cuit_cliente=cli.cuit  
        where st.ubicacion='area principal'  
              and cli.categoria like '%sponsor%'  
      );
```

5. subcon/TT/CTE/vars

Enunciado

304.5- **Mejores clientes.** Listar las categorías de clientes que alquilaron más stands de tipo doble que de tipo simple. En caso que no hayan alquilado stands de tipo simple debe mostrarse la categoría y asumir la cantidad de stands simples alquilados como 0. Indicar la categoría, cantidad de stands simples alquilados, cantidad de stands dobles alquilados, la diferencia entre ambas y la cantidad de eventos diferentes donde lo hicieron.

Resolución

```
with simples as (  
    select cli.categoria, count(*) cant  
    from cliente cli  
    inner join stand  
        on cli.cuit=stand.cuit_cliente  
    where stand.tipo='simple'  
    group by cli.categoria  
)  
select cli.categoria, count(*) cantDoble  
    , coalesce(simples.cant,0) cantSimple, count(*) -  
coalesce(simples.cant,0) diff  
    , count(distinct stand.id_evento) eventos  
from cliente cli  
inner join stand  
    on cli.cuit=stand.cuit_cliente  
left join simples  
    on cli.categoria=simples.categoria  
where stand.tipo='doble'  
group by cli.categoria, simples.cant  
having cantDoble > cantSimple;
```

6. transac

Enunciado

304.6- **Reemplazo de Fuse.** El artista Fuse (denominación) sufrió una lesión en su espalda y no podrá asistir a ninguna presentación que tenía asignada durante 2023. En su lugar las realizará Shiro Masamune. Dar de alta el nuevo presentador con

los datos a continuación y reemplazar a Fuse con él para todas sus futuras presentaciones hasta fines de 2023.

Nuevo presentador:

cuit: 38383838383

telefono: 3838383838

email: masanoriota@gits.sac

denominacion: Shiro Masamune

Resolución

```
begin;

select id into @fuse
from presentador
where denominacion='Fuse';

insert into presentador set
cuit='38383838383',telefono='3838383838',email='masanoriota@gits.sac',de
nominacion='Shiro Masamune';

select last_insert_id() into @sm;

update presentador_presentacion set id_presentador=@sm
where id_presentador=@fuse
and fecha_hora_ini between now() and '2023-12-31 23:59:59'

commit;
```

7. routines

Enunciado

304.7- Cree una función **presentaciones_anuales** que dado el cuit de un presentador calcule la cantidad de presentaciones por año y devuelva el promedio de las mismas. Luego listar los presentadores cuya cantidad de presentaciones en algún año sea mayor a la cantidad de presentaciones promedio de dicho año, utilizando dicha función. Indique nombre, apellido, denominación del presentador, año, cantidad de presentaciones por año, cantidad de presentaciones promedio de dicho año y la diferencia entre ambas.

Resolución

```
# función
delimiter 完

drop function if exists presentaciones_anuales 完

create function presentaciones_anuales(presen varchar(13)) returns
decimal(9,3)
reads sql data
begin
    declare prom decimal(9,3);

    with pres_anual as (
        select count(*) cant
        from presentador_presentacion pp
        inner join presentador pres
            on pp.id_presentador=pres.id
        where pres.cuit=presen
        group by year(pp.fecha_hora_ini)
    )
    select avg(cant) into prom from pres_anual;

    return prom;
end 完

delimiter ;

#uso
select pres.nombre, pres.apellido, pres.denominacion
    , count(*) cant, presentaciones_anuales(pres.cuit)
    , count(*) - presentaciones_anuales(pres.cuit) dif
from presentador_presentacion pp
inner join presentador pres
    on pp.id_presentador=pres.id
group by pres.id, year(pp.fecha_hora_ini)
having dif >0;
```

305

1. Inner join

Enunciado

305.1-Salas y presentadores de la presentación Queen Dekim.

Indicar los presentadores y las salas que participaron en las presentaciones llamadas "Queen Dekim". Indicar nombre, apellido y denominacion del presentador; nro, nombre de sala; fecha y hora de inicio y fin de la presentación. Ordenar por nombre de sala alfabético y fecha de inicio de la presentación descendente.

Resolución

```
# no es necesario usar coalesce pero no estaría mal
select p.nombre, p.apellido, p.denominacion
      , sala.nombre, sala.nro
      , pre.fecha_hora_ini, pre.fecha_hora_fin
from sala
inner join presentador_presentacion pp
      on sala.id_locacion=pp.id_locacion
      and sala.nro=pp.nro_sala
inner join presentador p
      on pp.id_presentador=p.id
inner join presentacion pre
      on pp.id_locacion=pre.id_locacion
      and pp.nro_sala=pre.nro_sala
      and pp.fecha_hora_ini=pre.fecha_hora_ini
where pre.nombre='Queen Dekim'
order by sala.nombre, pre.fecha_hora_ini desc;
```

2. left/right join

Enunciado

305.2-Locaciones en Orth y sus salas y presentaciones. Listar todas las locaciones en la zona de "Orth" y, solo para aquellas que tienen salas con capacidad superior a 500 asistentes, informar las salas y presentaciones realizadas en las mismas. Indicar nombre y dirección de la locación; nro, nombre y capacidad de la sala; nombre y fecha y hora de inicio de la presentación.

Resolucion

```

select loc.nombre, loc.direccion
      , sala.nro, sala.nombre, sala.capacidad_maxima
      , pre.nombre, pre.fecha_hora_ini
from locacion loc
left join sala
      on loc.id=sala.id_locacion
      and sala.capacidad_maxima > 500
left join presentacion pre
      on sala.id_locacion=pre.id_locacion
      and sala.nro=pre.nro_sala
where loc.zona='Orth';

```

3. group by/having

Enunciado

305.3-Éxitos de venta del 2021. Listar las presentaciones para las que se haya adquirido acceso para al menos 50 entradas durante el 2021 (según la fecha de venta de la entrada). Indicar nombre y tipo de la presentación; cantidad de entradas vendidas en 2021, promedio del costo de la entrada (de la entrada, no de la presentación) y diferencia entre las fecha y hora de venta de la primera y última entrada a dicha presentación.

Resolución

```

select pre.nombre, pre.tipo
      , count(*) cant, avg(ent.costo) prom, max(ent.fecha_hora_venta) -
      min(ent.fecha_hora_venta) dif
from presentacion pre
inner join presentacion_entrada pe
      on pre.id_locacion = pe.id_locacion
      and pre.nro_sala = pe.nro_sala
      and pre.fecha_hora_ini=pe.fecha_hora_ini
inner join entrada ent
      on pe.id_evento=ent.id_evento
      and pe.nro_entrada=ent.nro
where ent.fecha_hora_venta >= '20210101' and ent.fecha_hora_venta <
'20220101'
group by pre.id_locacion, pre.nro_sala, pre.fecha_hora_ini, pre.nombre,
pre.tipo
having count(*) > 50;

```

4. difference

Enunciado

305.4- **Locaciones de Orth no utilizadas para concursos.** Listar las locaciones en la zona de Orth que no hayan sido utilizadas en presentaciones de tipo concurso para eventos con la temática "Abyss". Indicar id, nombre, y dirección de la locación.

Resolución

```
select loc.id, loc.nombre, loc.direccion, loc.zona
from locacion loc
where loc.zona='Orth'
      and not exists (
        select 1
        from presentacion pre
        inner join evento ev
          on pre.id_evento=ev.id
        where pre.id_locacion=loc.id
              and pre.tipo="concurso"
              and ev.tematica="Abyss"
      );
```

```
select loc.id, loc.nombre, loc.direccion, loc.zona
from locacion loc
where loc.zona='Orth'
      and loc.id not in (
        select pre.id_locacion
        from presentacion pre
        inner join evento ev
          on pre.id_evento=ev.id
        where pre.tipo="concurso"
              and ev.tematica="Abyss"
      );
```

5. subcon/TT/CTE/vars

Enunciado

305.5- **Tipos de presentaciones menguantes.** Listar los tipos de presentaciones para las que se adquirieron más entradas el año pasado (2022) que este (2023). En caso que no se hayan adquirido

entradas para algún tipo de presentación este año se debe mostrar igualmente el tipo de presentación y asumir la cantidad como 0. Indicar, el tipo de presentación, cantidad del año pasado, cantidad del año actual, diferencia entre ambas y cantidad de locaciones distintas donde se realizaron el año pasado.

Resolución

```
with tipos2023 as (  
    select pre.tipo, count(*) cant  
    from presentacion pre  
    inner join presentacion_entrada pe  
        on pre.id_locacion=pe.id_locacion  
        and pre.nro_sala=pe.nro_sala  
        and pre.fecha_hora_ini=pe.fecha_hora_ini  
    where pre.fecha_hora_ini between '20230101' and '20231231 235959'  
    group by pre.tipo  
)  
select pre.tipo, count(*) cant2022  
    , coalesce(tipos2023.cant,0) cant2023, count(*)-  
coalesce(tipos2023.cant,0) diff  
    , count(distinct pre.id_locacion) locaciones  
from presentacion pre  
inner join presentacion_entrada pe  
    on pre.id_locacion=pe.id_locacion  
    and pre.nro_sala=pe.nro_sala  
    and pre.fecha_hora_ini=pe.fecha_hora_ini  
left join tipos2023  
    on pre.tipo=tipos2023.tipo  
where pre.fecha_hora_ini between '20220101' and '20221231 235959'  
group by pre.tipo, tipos2023.cant  
having cant2022>cant2023;
```

6. transac

Enunciado

305.6- Para aumentar la audiencia se contrató a una nueva empleada para desempeñarse como "community manager". Darla de alta en el sistema con los datos a continuación y sumarla a los encargados asignados a los eventos de tipo convención que aún no han sucedido con dicho rol a todos los eventos futuros de tipo convención.

Nueva empleada
cuil: 25252525252
nombre: Shuna
apellido: Kijin
telefono: 252525252
dirección: Hidden Cave 242
email: shuna@jurashinrin.ma
fecha de nacimiento: 2002-05-25
categoria: especialista
rol en los eventos: community manager

Resolución

```
begin;  
  
insert into empleado set  
    cuil='25252525252'  
    ,nombre='Shuna'  
    ,apellido='Hime'  
    ,telefono='252525252'  
    ,direccion='Hidden Cave 242'  
    ,email='shuna@jurashinrin.ma'  
    ,fecha_nac='2002-05-25'  
    ,categoria='especialista';  
  
insert into encargado_evento  
select id, 25252525252, 'community manager'  
from evento  
where fecha_desde > now()  
and tipo='convención';  
  
commit;
```

7. routines

Enunciado

305.7- Cree una función **accesos_presentaciones** que dado un tipo de presentación calcule la cantidad de entradas con acceso para cada presentación de dicho tipo y devuelva la cantidad promedio de las mismas. Luego lista las presentaciones cuya cantidad de entradas con acceso sea menor al promedio del tipo de función al

que corresponde, utilizando dicha función. Indicar nombre y tipo de la presentación, cantidad de entradas con acceso a la presentación, promedio del tipo y diferencia entre ambas.

Resolución

```
#funcion
delimiter -.-

drop function if exists accesos_presentaciones -.-

create function accesos_presentaciones(tipo_pre varchar(255)) returns
decimal(9,3)
reads sql data
begin
    declare prom decimal(9,3);
    with accesos as (
        select count(*) cant
        from presentacion_entrada pe
        inner join presentacion pre
            on pe.id_locacion=pre.id_locacion
            and pe.nro_sala=pre.nro_sala
            and pe.fecha_hora_ini=pre.fecha_hora_ini
        where pre.tipo=tipo_pre
        group by pe.id_locacion, pe.nro_sala, pe.fecha_hora_ini
    )
    select avg(cant) into prom from accesos;
    return prom;
end -.-

delimiter ;

# uso
select pre.nombre, pre.tipo
, count(*) cant, accesos_presentaciones(pre.tipo)
, count(*) - accesos_presentaciones(pre.tipo) dif
from presentacion_entrada pe
inner join presentacion pre
    on pe.id_locacion=pre.id_locacion
    and pe.nro_sala=pre.nro_sala
    and pe.fecha_hora_ini=pre.fecha_hora_ini
group by pe.id_locacion, pe.nro_sala, pe.fecha_hora_ini
having dif >0
```


Recuperatorio

1. inner join

Enunciado

REC.1-Asistentes a "Is there a bottom?" del evento "Abyss".

Listar todos los asistentes y compradores de entradas a la presentación "Is there a bottom?" del evento "Made in Abyss".

Indicar nombre y apellido del asistente y comprador; nro, fecha y hora de venta y costo de la entrada.

Resolución

```
select asist.id, asist.nombre nombre_asistente, asist.apellido
apellido_asistente, comp.id id_comprador, comp.nombre nombre_comprador,
comp.apellido comprador_apellido
    , ent.nro, ent.fecha_hora_venta, ent.costo
from persona asist
inner join entrada ent
    on ent.id_asistente=asist.id
inner join persona comp
    on ent.id_comprador=comp.id
inner join presentacion_entrada pe
    on pe.id_evento=ent.id_evento
    and pe.nro_entrada=ent.nro
inner join presentacion pre
    on pre.id_locacion=pe.id_locacion
    and pre.nro_sala=pe.nro_sala
    and pre.fecha_hora_ini=pe.fecha_hora_ini
inner join evento eve
    on pre.id_evento=eve.id
where pre.nombre = "Is there a bottom?" and eve.nombre = "Made in Abyss"
```

2. left/right join

Enunciado

REC.2- Presentaciones de salas en reparación. Listar las locaciones con salas en estado "En reparación". Si las mismas tienen presentaciones en el próximo mes listar los datos de la presentación y evento. Indicar, nombre y nro de la sala; nombre y

fecha y hora de inicio de la presentación; id, nombre, tipo y temática del evento.

Resolución

```
select sala.nombre, sala.nro
      , pre.nombre, pre.fecha_hora_ini
      , ev.id, ev.nombre, ev.tipo, ev.tematica
from sala
left join presentacion pre
      on sala.nro=pre.nro_sala
      and sala.id_locacion=pre.id_locacion
      and pre.fecha_hora_ini between now() and TIMESTAMPADD(MONTH,1,now())
left join evento ev
      on ev.id=pre.id_evento
where sala.estado='en reparacion';
```

3. group by/having

Enunciado

REC.3- Eventos de Star Trek con pocos stands. Listar los eventos con temática "Star Trek" en los que se hayan alquilado menos de 10 stands de tipo "simple". Indicar id, nombre, tipo y temática del evento; cantidad de stands "simples" alquilados, la anticipación promedio al contratar el stand (la anticipación es cuantos días antes del evento se realizó el contrato) y el valor acordado promedio por cada alquiler.

Resolución

```
select ev.id, ev.nombre, ev.tipo, ev.tematica
      , count(stand.id_evento) cant, avg(stand.fecha_contrato-
ev.fecha_desde) prom_anticipacion
      , avg(stand.valor_acordado) prom_valor
from evento ev
left join stand
      on ev.id=stand.id_evento
      and stand.tipo='simple'
where tematica='star trek'
```

```
group by ev.id  
having count(stand.id_evento) < 10;
```

4. difference

Enunciado

REC.4- **Presentadores sin presentarse.** Listar los presentadores panelistas con especialidad "space opera" que no hayan participado en una presentación de tipo debate del 2022 (usar fecha de inicio). Indicar id, cuit, nombre y apellido del panelista.

Resolución

```
select pan.id, pan.cuit, pan.nombre, pan.apellido  
from presentador pan  
where pan.especialidad='space opera'  
      and pan.id not in (  
          select id_presentador  
          from presentador_presentacion pp  
          inner join presentacion pre  
              on pp.id_locacion=pre.id_locacion  
              and pp.nro_sala=pre.nro_sala  
              and pp.fecha_hora_ini=pre.fecha_hora_ini  
          where pp.fecha_hora_ini between '2022-01-01' and '2022-12-31  
23:59:59'  
              and pre.tipo='debate'  
      );
```

5. subcon/TT/CTE/vars

Enunciado

REC.5- **Mejores clientes.** Listar todos los clientes de los cuales el total de los alquileres de stands pagados en algún año supere al promedio del total de alquileres de stands pagados por cliente en ese mismo año. Indicar cuit, razón social, año donde sucedió, total pagado por el cliente y promedio del año.

Resolución

```

with pagado as (
    select cli.cuit, cli.razon_social
        , year(st.fecha_contrato) anio, sum(valor_acordado) total
    from cliente cli
    inner join stand st
        on cli.cuit=st.cuit_cliente
    group by cli.cuit, year(st.fecha_contrato)
),
prom as (
    select anio, avg(total) promedio
    from pagado
    group by anio
)
select *
from pagado
inner join prom
    on pagado.anio=prom.anio
where promedio < total;

```

6. transac

Enunciado

REC.6- **Copia de evento.** Dado el éxito del evento 10001 - Shitara Slime, se decidió repetirlo. Crear un nuevo evento "re:Shitara Slime" con las mismas presentaciones, salas, oferta de stands (sin ser alquilados), y empleados designados como encargados. Además la copia del evento 10001 se realiza seis meses después y con un 30% más en el valor base de la entrada en el evento, de las presentaciones y de los stands.

Resolución

```

begin;

insert into evento(fecha_desde, fecha_hasta, valor_base_entrada, tematica,
tipo, nombre)
select TIMESTAMPADD(month,6,fecha_desde),
TIMESTAMPADD(month,6,fecha_hasta), valor_base_entrada*1.3
    , tematica, tipo, 're:Shitara Slime'
from evento
where id=10001;

```

```

select LAST_INSERT_ID() into @ress;

insert into presentacion
select @ress, id_locacion,nro_sala,nombre, descripcion
      , TIMESTAMPADD(month,6,fecha_hora_ini),
TIMESTAMPADD(month,6,fecha_hora_fin), costo_entrada*1.3, tipo
from presentacion
where id_evento=10001;

insert into encargado_evento
select @ress, cuil_encargado, rol
from encargado_evento
where id_evento=10001;

insert into stand(id_evento,nro,ubicacion,valor_sugerido,tipo)
select @ress, nro, ubicacion, valor_sugerido*1.3,tipo
from stand
where id_evento=10001;

commit;

```

7. routines

Enunciado

REC.7- Cálculo de saldos. Cree un store procedure saldo_evento que dado un rango de fechas calcule la proporción de saldo (ganancia o pérdida) en alquileres (de stands) por evento entre esas fechas.

saldo total: sumatoria de (valor_acordado - valor_sugerido)

proporcion de saldo por evento: saldo total / sumatoria de valor_sugerido

El SP debe listar los eventos ordenados por mayor proporción de saldo descendente. Indique nombre del evento, temática, saldo total, y proporción de saldo.

Finalmente invocar el SP para las fechas entre 10/10/2023 y 25/10/2023

Resolución

```
#sp
```

```
delimiter お金
```

```
drop procedure if exists saldo_evento お金
```

```
create procedure saldo_evento (in desde date, in hasta date)
begin

    select ev.nombre, ev.tematica, sum(st.valor_acordado-
st.valor_sugerido) saldo
        , sum(st.valor_acordado-st.valor_sugerido)/sum(valor_sugerido)
prop
    from evento ev
    inner join stand st
        on ev.id=st.id_evento
    where st.fecha_contrato between desde and hasta
    group by ev.id
    order by prop desc;

end お金

delimiter ;
```

```
# uso
call saldo_evento('2023-10-10', '2023-10-25');
```

Globalizador

1. inner join

Enunciado

GLB.1-Entradas que alguna persona compró para alguien más en 2023. Listar las personas que le compraron entradas a alguien más en 2023. Indicar nro, fecha y hora de venta y costo de la entrada; id nombre, apellido tipo y nro documento del comprador y del asistente; id, tipo y nombre del evento al que corresponden las entradas.

Resolución

```
select e.nro, e.fecha_hora_venta, e.costo
, asis.id id_asistente, asis.nombre nombre_asistente, asis.apellido
apellido_asistente, asis.tipo_doc tipo_doc_asistente, asis.nro_doc
```

```

nro_doc_asistente
, comp.id id_comprador, comp.nombre nombre_comprador, comp.apellido
apellido_comprador, comp.tipo_doc tipo_doc_comprador, comp.nro_doc
nro_doc_comprador
, ev.id, ev.tipo, ev.nombre
from entrada e
inner join evento ev
on e.id_evento=ev.id
inner join persona comp
on comp.id=e.id_comprador
inner join persona asis
on asis.id=e.id_asistente
where e.fecha_hora_venta between '20230101' and '20231231'
and id_comprador <> id_asistente

```

2. left/right join

Enunciado

GLB.2- **Presentadores panelistas de Space Opera en 2023.** Listar los panelistas cuya especialidad sea 'space opera' y, si participaron de alguna presentación durante 2023, mostrar los datos de la presentación y evento. Indicar cuit, nombre y apellido del panelista; nombre y fecha y hora de inicio de la presentación.

Resolución

```

select pan.cuit, pan.nombre, pan.apellido
, pre.nombre, pre.fecha_hora_ini
from presentador pan
left join presentador_presentacion pp
on pan.id=pp.id_presentador
and pp.fecha_hora_ini between '20230101' and '20231231'
left join presentacion pre
on pp.id_locacion=pre.id_locacion
and pp.nro_sala=pre.nro_sala
and pp.fecha_hora_ini=pre.fecha_hora_ini
where pan.especialidad ='space opera'

```

3. group by/having

Enunciado

GLB.3- **Compradores frecuentes.** Listar las personas que compraron al menos 4 entradas desde el 2021. Indicar id, nombre y apellido del comprador; cantidad de entradas que compró, cantidad de asistentes distintos para los que compró entradas y fecha de la primera entrada que compró.

Resolución

```
select p.id, p.nombre, p.apellido
      , count(*) ent_compradas, count(distinct e.id_asistente),
      min(fecha_hora_venta) primer_compra
from entrada e
inner join persona p
      on e.id_comprador=p.id
where e.fecha_hora_venta > '20210101'
group by p.id
having count(*) ≥ 4
```

4. difference

Enunciado

GLB.4- **Asistentes autofinanciados.** Listar los asistentes que **siempre** se han comprado su propia entrada. Indicar id, nombre, apellido e email del asistente.

Resolución

```
select distinct p.id, p.nombre, p.apellido, p.email
from persona p
inner join entrada e
      on p.id=e.id_asistente
where p.id not in (
      select ent_mismo.id_asistente
      from entrada ent_mismo
      where ent_mismo.id_asistente <> ent_mismo.id_comprador
)
```


5. subcon/TT/CTE/vars

Enunciado

GLB.5- **Mejores clientes**. Indicar los eventos, en los cuales el total que se haya pagado por los stands sea mayor al promedio de totales pagados en alquiler de stands por evento. Indicar id, nombre, tipo y temática del evento, el total pagado y cuanto supera el promedio. Ordenar por importe total descendente y por nombre del evento ascendente.

Resolución

```
with pagoTotal as (  
    select ev.id, ev.nombre, ev.tipo, ev.tematica  
        , sum(valor_acordado) total_pagado  
    from evento ev  
    inner join stand  
        on ev.id=stand.id_evento  
    group by ev.id  
)  
select *  
from pagoTotal  
where total_pagado > (select avg(total_pagado) from pagoTotal);
```

6. transac

Enunciado

GLB.6- **IAs**. Se canceló la presentación "Isekais predican el uso de las IAs" y en su lugar se realizará la presentación "Habilidades de animes que parecen IAs". Dar de alta la nueva presentación para el mismo evento para el día siguiente a la misma hora, en la misma sala, con los mismos presentadores y reasignar todas las entradas que hayan adquirido acceso a la vieja presentación a la nueva y borrar la vieja presentación. Nombre de la presentación: Habilidades de animes que parecen IAs Descripción: Muchos personajes principales y de soporte parecen tener habilidades con las que pueden interactuar y les brindan información necesaria, parecen salidos de una IA avanzada entrenada para ayudarlos.

Resolución

```
begin;

select id_locacion, nro_sala, fecha_hora_ini into @loc, @sala, @fh_ini
from presentacion
where nombre='Isekais predican el uso de las IAs';

select TIMESTAMPADD(week, 1, @fh_ini) into @nuevo_ini;

insert into presentacion
select id_evento, id_locacion, nro_sala, 'Habilidades de animes que
parecen IAs'
      , 'Muchos personajes principales y de soporte parecen tener
habilidades con las que pueden interactuar y les brindan información
necesaria, parece salidos de una IA avanzada entrenada para ayudarlos.'
      , TIMESTAMPADD(week, 1, fecha_hora_ini), TIMESTAMPADD(week, 1,
fecha_hora_fin), costo_entrada, tipo
from presentacion
where id_locacion=@loc and nro_sala=@sala and fecha_hora_ini=@fh_ini;

update presentacion_entrada
set fecha_hora_ini=@nuevo_ini
where id_locacion=@loc and nro_sala=@sala and fecha_hora_ini=@fh_ini;

update presentador_presentacion
set fecha_hora_ini=@nuevo_ini
where id_locacion=@loc and nro_sala=@sala and fecha_hora_ini=@fh_ini;

delete from presentacion where id_locacion=@loc and nro_sala=@sala and
fecha_hora_ini=@fh_ini;
commit;
```

7. routines

Enunciado

GLB.7- **Eventos con altos ingresos.** Cree una función llamada ingreso_evento que dado un evento, calcule los ingresos del mismo: ingreso por stand + ingreso total de entradas, utilizando el valor acordado y costo respectivamente. Luego utilizando dicha función calcule el total por evento y liste los eventos cuyo total supere al promedio por evento. Indique id y nombre del

evento, recaudación total del evento, recaudación promedio y diferencia.

Resolución

```
delimiter $_$

drop function if exists ingreso_evento $_$

create function ingreso_evento(id integer unsigned)
returns decimal(10,3)
reads sql data
begin
    declare total_acordado decimal(10,3);
    declare total_entrada decimal(10,3);
    select coalesce(sum(valor_acordado),0) into total_acordado from
stand where id_evento=id;

    select coalesce(sum(costo),0) into total_entrada from entrada where
id_evento=id;

    select total_acordado+total_entrada;
end $_$
delimiter ;

select avg(e.id) into @prom
from evento e;

select e.id, e.nombre
    , ingreso_evento(e.id), @prom promedio
    , ingreso_evento(e.id) - @prom diff
from evento e
where ingreso_evento(e.id) > @prom;
```

Anexo datos transacciones

```
use convenciones_underground;

delete from presentacion_entrada;
delete from entrada;
delete from persona;
delete from presentador_presentacion;
delete from presentador;
```

```

delete from stand;
delete from cliente;
delete from presentacion;
delete from encargado_evento;
delete from sala;
delete from locacion;
delete from evento;
delete from empleado;

#evento
insert into evento values(10001,
'20231102','20231105',500,'Isekai','convencion','Shitara Slime');

insert into evento values(10002, '20231121','20231125',750,'Ciencia
Ficcion','premier pelicula','Space Operas');

#locacion
insert into locacion values(11001, 'Rimuru', POINT(-1.107778,
36.642778),'Jura Tempest Federation Road KM 0','Jura Forest');

#sala
insert into sala values(11001, 1, 'Rouge', 240, 200,'habilitada');
insert into sala values(11001, 2, 'Noir', 400, 350,'habilitada');

#presentacion
insert into presentacion values(10002, 11001,1,'La dualidad en la
trama','Las space operas nos llevan a poner en tela de juicio los
límites de la moralidad, la dualidad en la trama es el elemento decisivo
de éxito de la historia', '2023-11-23 17:00:00','2023-11-23
20:00:00',850,'debate');

insert into presentacion values(10001, 11001,1,'Isekais predican el uso
de las IAs','Las skills de información y soporte en los personajes de
Isekai (Raphael-sensei, Sapo-kun, Nano) se parecen a las IAs generativas
al mejor estilo SCI-FI', '2023-11-02 10:00:00','2023-11-02
12:00:00',1100,'disertación');

insert into presentacion values(10001, 11001,2,'Book Transmigrator: Un
nuevo subgenero','Recientemente ha aparecido un nuevo subgenero de
isekais donde el autor o lector transmigra como un personaje secundario
de la novela que escribe o lee. Que diferencias trae conocer la trama al
típico isekai munchkin', '2023-11-02 13:00:00','2023-11-02
16:00:00',800,'debate');

#encargado
insert into empleado values(23232323232, 'Clayman',
'Clown','232323232','Jistav

```

```
232','clayman@moderateclowntroupe.com','1992-03-23','jefe');
```

```
#evento_encargado
```

```
insert into encargado_evento values(10001,2323232323,'supervisor');
```

```
#cliente
```

```
insert into cliente values(14141414141, 'Falmuth', '1414141414','44  
Arwenack Street','falmuth@tairiku.ma','Gold');
```

```
#stand
```

```
insert into stand values(10001,1,'central',  
10000,'merchandising',14141414141,9800,'20231012');  
insert into stand values(10002,1,'entrada',  
1000,'banner',14141414141,1500,'20231022');
```

```
#presentador
```

```
insert into presentador(id,cuit,telefono,email,denominacion) values  
(30000, '30303030303', '3030303030', 'fusesensei@slime.kun', 'Fuse');  
insert into presentador(id,cuit,telefono,email,denominacion) values  
(30001, '31313131313', '3131313131', 'nitroplus@cern.ch', 'Nitro');  
insert into presentador(id,cuit,telefono,email,nombre,  
apellido,especialidad) values (30002, '32323232323', '3232323232',  
'tanakasensei@logh.jp', 'Yoshiki','Tanaka','space opera');  
insert into presentador(id,cuit,telefono,email,nombre, apellido,cv)  
values (30003, '33333333333', '33333333333', 'singshong@orv.kr',  
'Sing','Shong','Sing Shong es el seudónimo de la pareja de autores de  
Omnicient Readers Viewpoint y otras series. El nombre real de Sing Shong  
es Kim Seong-hyeon, mientras que el nombre real de Gyaon es Kim Gyeong-  
hyeon. Ambos son coreanos y nacieron en 1984.');
```

```
insert into presentador(id,cuit,telefono,email,nombre,  
apellido,especialidad) values (30004, '34343434343', '3434343434',  
'yooryeo@tcf.kr', 'Yoo Ryeo','Han','book transmigrator novels');
```

```
insert into presentador(id,cuit,telefono,email,nombre,  
apellido,especialidad) values (30005, '35353535353', '3535353535',  
'watanabesensei@sunrise.jp', 'Shinichiro','Watanabe','direccion');  
insert into presentador(id,cuit,telefono,email,nombre,  
apellido,especialidad) values (30006, '36363636363', '3636363636',  
'keino@sunrise.jp', 'Keiko','Nobumoto','space opera');  
insert into presentador(id,cuit,telefono,email,nombre,  
apellido,especialidad) values (30007, '37373737373', '3737373737',  
'toshimoto@sunrise.jp', 'Toshihiro','Kawamoto','diseño de personajes');
```

```
#presentador_presentacion
```

```
insert into presentador_presentacion values(30000,11001,1,'2023-11-02  
10:00:00');
```

```
insert into presentador_presentacion values(30003,11001,2,'2023-11-02
13:00:00');
insert into presentador_presentacion values(30004,11001,2,'2023-11-02
13:00:00');
```

```
insert into presentador_presentacion values(30002,11001,1,'2023-11-23
17:00:00');
insert into presentador_presentacion values(30005,11001,1,'2023-11-23
17:00:00');
insert into presentador_presentacion values(30006,11001,1,'2023-11-23
17:00:00');
insert into presentador_presentacion values(30007,11001,1,'2023-11-23
17:00:00');
```

#persona

```
insert into persona values(40001,'Luminous', 'Valentine', 'dni',
'41414141', 'nightmarequeen@westernholychurch.ru');
insert into persona values(40002,'Louis', 'Valentine', 'dni',
'42424242', 'pope@westernholychurch.ru');
insert into persona values(40003,'Roy', 'Valentine', 'dni', '43434343',
'archduke@westernholychurch.ru');
insert into persona values(40004,'Phantom Franklin', 'Harlock III',
'dni', '44444444', 'kaizoku@arcadia.space');
insert into persona values(40005,'Ruri', 'Hoshino', 'dni', '45454545',
'electronicfairy@nadesico.mars');
```

##entradas

```
insert into entrada values(10001,1,'2023-10-01',2400,40001,40001);
insert into entrada values(10001,2,'2023-10-01',2400,40001,40002);
insert into entrada values(10001,3,'2023-10-01',1600,40001,40003);
insert into entrada values(10001,4,'2023-09-01',1850,40004,40005);
insert into entrada values(10002,1,'2023-09-01',1600,40004,40005);
insert into entrada values(10002,2,'2023-09-01',1600,40004,40004);
insert into entrada values(10002,3,'2023-10-01',1600,40001,40002);
```

```
#presentacion_entrada
insert into presentacion_entrada values(10001, 1, 11001, 1, '2023-11-02
10:00:00');
insert into presentacion_entrada values(10001, 1, 11001, 2, '2023-11-02
13:00:00');
```

```
insert into presentacion_entrada values(10001, 2, 11001, 1, '2023-11-02
10:00:00');
insert into presentacion_entrada values(10001, 2, 11001, 2, '2023-11-02
13:00:00');
```

```
insert into presentacion_entrada values(10001, 3, 11001, 1, '2023-11-02
10:00:00');

insert into presentacion_entrada values(10001, 4, 11001, 1, '2023-11-02
10:00:00');

insert into presentacion_entrada values(10002, 1, 11001, 1, '2023-11-23
17:00:00');
insert into presentacion_entrada values(10002, 2, 11001, 1, '2023-11-23
17:00:00');
insert into presentacion_entrada values(10002, 3, 11001, 1, '2023-11-23
17:00:00');
```