

030609

CHRISTIAN MILLÁN SORIA

1º DAW TARDE

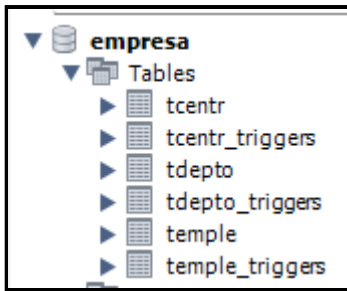
Crear en cada una de las tablas un disparador que registre los movimientos en la tabla para cada una de las siguientes circunstancias:

- Antes de un INSERT.
- Antes de un UPDATE.
- Antes de un DELETE.

Para ello, se deberá insertar, actualizar y borrar datos para demostrar su correcto funcionamiento.

Creo una tabla secundaria para cada una de las tablas de la base de datos, donde se almacenarán los movimientos de su tabla correspondiente.

```
create table `tcentr_triggers`(  
  `id` int not null auto_increment,  
  `tipo` varchar(45) null,  
  `fecha` datetime null default current_timestamp,  
  primary key (`id`)  
);  
  
/*****/  
  
create table `tdepto_triggers`(  
  `id` int not null auto_increment,  
  `tipo` varchar(45) null,  
  `fecha` datetime null default current_timestamp,  
  primary key (`id`)  
);  
  
/*****/  
  
create table `temple_triggers`(  
  `id` int not null auto_increment,  
  `tipo` varchar(45) null,  
  `fecha` datetime null default current_timestamp,  
  primary key (`id`)  
);
```



Ahora creo los triggers de cada tipo para cada tabla.

```

delimiter //
create trigger registro1_tcentr after insert on tcentr
for each row
begin
    insert into tcentr_triggers(tipo) value('after insert');
end//
delimiter ;

/*****/

delimiter //
create trigger registro2_tcentr before insert on tcentr
for each row
begin
    insert into tcentr_triggers(tipo) value('before insert');
end//
delimiter ;

/*****/

delimiter //
create trigger registro3_tcentr after update on tcentr
for each row
begin
    insert into tcentr_triggers(tipo) value('after update');
end//
delimiter ;

/*****/

delimiter //
create trigger registro4_tcentr before update on tcentr
for each row
begin
    insert into tcentr_triggers(tipo) value('before update');
end//
delimiter ;

/*****/

delimiter //
create trigger registro5_tcentr after delete on tcentr

```

```

for each row
begin
    insert into tcentr_triggers(tipo) value('after delete');
end//
delimiter ;

/*****

delimiter //
create trigger registro6_tcentr before delete on tcentr
for each row
begin
    insert into tcentr_triggers(tipo) value('before delete');
end//
delimiter ;

/*****

-- igual para las tablas "tdepto" y "temple"...

```

Prueba de comportamiento correcto:

```
insert into tcentr values(30, 'test', 'test');
```

	id	tipo	fecha
▶	1	before insert	2023-05-09 02:04:56
	2	after insert	2023-05-09 02:04:56
*	NULL	NULL	NULL

```
update tcentr set numce=40 where numce=30;
```

	id	tipo	fecha
▶	1	before insert	2023-05-09 02:04:56
	2	after insert	2023-05-09 02:04:56
	3	before update	2023-05-09 02:07:18
	4	after update	2023-05-09 02:07:18
*	NULL	NULL	NULL

```
delete from tcentr where numce=40;
```

	id	tipo	fecha
▶	1	before insert	2023-05-09 02:04:56
	2	after insert	2023-05-09 02:04:56
	3	before update	2023-05-09 02:07:18
	4	after update	2023-05-09 02:07:18
	5	before delete	2023-05-09 02:09:18
	6	after delete	2023-05-09 02:09:18
*	NULL	NULL	NULL

Además, para cada una de las preguntas se deberá hacer lo siguiente (siempre que sea posible):

- Crear una vista.
- Crear un procedimiento.
- Crear una función.

*Utilizar cursores y bucles siempre que sea posible.

BLOQUE 1

1. Hallar, por orden alfabético, los nombres de los departamentos cuyo director lo es en funciones y no en propiedad.

```
create view ej1b1 as
select nomde from tdepto where tidir='f' order by 1;

/*****/

select * from empresa.ej1b1;
```

	nomde
▶	ORGANIZACION
	SECTOR INDUSTRIAL

```
delimiter //
create procedure procedure_ej1b1()
begin
    select nomde from tdepto where tidir='f' order by 1;
end //
delimiter ;

/*****/

call empresa.procedure_ej1b1();
```

	nomde
▶	ORGANIZACION
	SECTOR INDUSTRIAL

2. Obtener un listín telefónico de los empleados del departamento 121, incluyendo nombre del empleado, número del empleado y extensión telefónica. Por orden alfabético descendente.

```
create view ej2b1 as
select nomem, numem, extel from temple where numde=121 order by 1 desc;

/*****/

select * from empresa.ej2b1;
```

	nomem	numem	extel
▶	RUIZ, FABIOLA	370	360
	PONS, CESAR	110	350
	PEREZ, JULIO	150	340

```
delimiter //
create procedure procedure_ej2b1(in numde_param int)
begin
    select nomem, numem, extel from temple where numde=numde_param order by nomem
    desc;
end //
delimiter ;

/*****/

call empresa.procedure_ej2b1(121);
```

	nomem	numem	extel
▶	RUIZ, FABIOLA	370	360
	PONS, CESAR	110	350
	PEREZ, JULIO	150	340

3. Obtener por orden creciente una relación de todos los números de extensiones telefónicas de los empleados. Elimina las repeticiones.

```
create view ej3b1 as
select distinct extel from temple order by 1;

/*****/

select * from empresa.ej3b1;
```

	extel
▶	220
	340
	350
	360
	410
	508
	750
	760
	800
	810
	850
	880
	910

```

delimiter //
create procedure procedure_ej3b1()
begin
  declare extel_value varchar(255);
  declare done int default false;

  declare temple_cursor cursor for select extel from temple;

  declare extel_cursor cursor for select distinct extel from temple order by 1;

  create temporary table if not exists unique_extel (extel_value varchar(255));

  open temple_cursor;

  repeat
    fetch temple_cursor into extel_value;

    if not exists(select * from unique_extel where extel_value=extel_value) then
      insert into unique_extel values(extel_value);
    end if;

    if(extel_value is null) then
      set done=true;
    end if;
  until done end repeat;

  close temple_cursor;

  open extel_cursor;

  repeat
    fetch extel_cursor into extel_value;

    if(extel_value is not null) then
      select extel_value;
    end if;
  until extel_value is null end repeat;

```

```

close extel_cursor;

drop table if exists unique_extel;
end //
delimiter ;

/*****/

call empresa.procedure_ej3b1();

```



4. Hallar la comisión, nombre y salario de los empleados con más de un hijo, clasificados por comisión, y dentro de la comisión por orden alfabético. El listado debe incluir también los empleados con más de un hijo aunque no tengan comisión.

```

create view ej4b1 as
select ifnull(comis, 0), nomem, salar from temple where numhi>1 order by 1, 2;

/*****/

select * from empresa.ej4b1;

```

	ifnull(comis, 0)	nomem	salar
▶	0.00	FLOR, DOROTEA	1500.00
	0.00	GIL, GLORIA	2200.00
	0.00	LOPEZ, ANTONIO	2200.00
	0.00	PONS, CESAR	1000.00
	0.00	VEIGA, JULIANA	1500.00
	500.00	PEREZ, MARCOS	1800.00
	800.00	GARCIA, OCTAVIO	2200.00
	1000.00	LARA, LUCRECIA	1800.00
	1000.00	SANZ, LAVINIA	2200.00
	1100.00	TEROL, LUCIANO	1200.00

```

delimiter //
create procedure procedure_ej4b1()
begin
declare finalizado int default false;
declare comision decimal(10, 2);
declare nombre_empleado varchar(50);
declare salario decimal(10, 2);

declare cursor_datos_temple cursor for
select ifnull(comis, 0), nomem, salar
from temple
where numhi>1
order by 1, 2;

```

```

declare continue handler for not found set finalizado = true;

open cursor_datos_temple;

while not finalizado do
    fetch cursor_datos_temple into comision, nombre_empleado, salario;
    if not finalizado then
        select comision, nombre_empleado, salario;
    end if;
end while;


close cursor_datos_temple;
end //
delimiter ;

/*****

call empresa.procedure_ej4b1();

```

	comision	nombre_empleado	salario
▶	1100.00	TEROL, LUCIANO	1200.00



Result 1
Result 2
Result 3
Result 4
Result 5
Result 6
Result 7
Result 8
Result 9
Result 10 x

5. Obtener salario y nombre de los empleados con dos hijos por orden decreciente de salario y por orden alfabético dentro del salario.

```

create view ej5b1 as
select salar, nomem from temple where numhi=2 order by 1 desc, 2;

/*****

select * from empresa.ej5b1;

```


	salar	nomem
▶	1800.00	LARA, LUCRECIA
	1800.00	PEREZ, MARCOS
	1200.00	TEROL, LUCIANO

```

delimiter //
create procedure procedure_ej5b1()
begin
    declare salario decimal(10,2);
    declare nombre_empleado varchar(50);
    declare var_count int;

    declare cur_datos cursor for
        select salar, nomem
        from temple
        where numhi=2
        order by salar desc, nomem;

    select count(*) into var_count from temple where numhi=2;

    open cur_datos;
    fetch cur_datos into salario, nombre_empleado;

    while var_count>0 do
        select salario, nombre_empleado;
        fetch cur_datos into salario, nombre_empleado;
        set var_count=var_count-1;
    end while;

    close cur_datos;
end //
delimiter ;


/*****/

call empresa.procedure_ej5b1();

```

	salario	nombre_empleado
▶	1200.00	TEROL, LUCIANO

Result 1 Result 2 **Result 3** x



6. Obtener el nombre de los empleados cuya comisión es superior o igual al 50% de su salario, por orden alfabético.

```
create view ej6b1 as
select nomem from temple where comis>=salar*0.5 order by 1;

/*****/

select * from empresa.ej6b1;
```

	nomem
▶	DIEZ, AMELIA
	LARA, LUCRECIA
	TEROL, LUCIANO

```
delimiter //
create procedure procedure_ej6b1()
begin
  declare nomem_var varchar(50);
  declare salar_var decimal(10,2);
  declare comis_var decimal(10,2);

  declare temple_cursor cursor for select nomem, salar, comis from temple;

  create temporary table if not exists temple_reporte_tabla (
    nomem varchar(50)
  );

  open temple_cursor;
  fetch temple_cursor into nomem_var, salar_var, comis_var;
```

```

while (nomem_var is not null) do
  if(comis_va>=(salar_var * 0.5)) then
    insert into temple_reporte_tabla (nomem) values (nomem_var);
  end if;
  fetch temple_cursor into nomem_var, salar_var, comis_var;
end while;

close temple_cursor;

select nomem from temple_reporte_tabla order by nomem;

drop temporary table temple_reporte_tabla;
end //
delimiter ;

/*****/

call empresa.procedure_ej6b1();

```



7. En una campaña de ayuda familiar se ha decidido dar a los empleados una paga extra de 30 euros por hijo a partir del tercero inclusive. Obtener por orden alfabético para estos empleados: nombre y salario total que van a cobrar incluyendo esta paga extra.

```

create view ej7b1 as
select nomem, salar+30*(numhi-2) from temple where numhi>2 order by 1;

/*****/

select * from empresa.ej7b1;

```

	nomem	salar+30*(numhi-2)
►	FLOR, DOROTEA	1590.00
	GARCIA, OCTAVIO	2230.00
	GIL, GLORIA	2230.00
	LOPEZ, ANTONIO	2320.00
	PONS, CESAR	1030.00
	SANZ, LAVINIA	2230.00
	VEIGA, JULIANA	1560.00

```

/*****/

call empresa.procedure_ej7b1();

```



7.1. En una campaña de ayuda familiar se ha decidido dar a los empleados una paga extra de 30 euros por hijo a partir del tercero inclusive. Obtener por orden alfabético para estos empleados: nombre y salario total que van a cobrar incluyendo esta paga extra. Para el resto de los empleados se debe mostrar el nombre y el salario que reciben siempre. Hacer este ejercicio de dos formas diferentes: con el operador "union" y con la expresión "case".

```
create view ej71b1 as
select nomem, salar+30*(numhi-2) from temple where numhi>2 union select nomem,
salar from temple where numhi<=2 order by 1;
```

```
/*****
```

```
select * from empresa.ej71b1;
```

	nomem	salar+30*(numhi-2)
►	DIEZ, AMELIA	1500.00
	FLOR, DOROTEA	1590.00
	GARCIA, OCTAVIO	2230.00
	GIL, GLORIA	2230.00
	LARA, LUCRECIA	1800.00
	LOPEZ, ANTONIO	2320.00
	MARTÍN, MICAELA	1800.00
	PEREZ, JULIO	2200.00
	PEREZ, MARCOS	1800.00
	PONS, CESAR	1030.00
	RUIZ, FABIOLA	1800.00
	SANZ, LAVINIA	2230.00
	TEROL, LUCIANO	1200.00
	VEIGA, JULIANA	1560.00

```
delimiter //
create procedure procedure_ej71b1()
begin
  declare nom_em varchar(50);
  declare salario decimal(10,2);
  declare num_em int;

  declare cur_empleados cursor for select nomem, salar, numhi from temple;

  declare continue handler for not found set @fin=1;

  set @fin=0;

  create temporary table empleados_temp(
    nom_empleado varchar(50),
    salario_nuevo decimal(10,2)
  );

  open cur_empleados;
```

```

fetch cur_empleados into nom_em, salario, num_em;

while(@fin=0) do
  if(num_em>2) then
    set salario=salario+(30*(num_em-2));
  end if;

  insert into empleados_temp (nom_empleado, salario_nuevo)
  values (nom_em, salario);

  fetch cur_empleados into nom_em, salario, num_em;
end while;

close cur_empleados;

select nom_empleado, salario_nuevo
from empleados_temp
where salario_nuevo is not null
order by 1;

drop temporary table if exists empleados_temp;
end //
delimiter ;

/*****/

call empresa.procedure_ej71b1();

```

	nom_empleado	salario_nuevo
►	DIEZ, AMELIA	1500.00
	FLOR, DOROTEA	1590.00
	GARCIA, OCTAVIO	2230.00
	GIL, GLORIA	2230.00
	LARA, LUCRECIA	1800.00
	LOPEZ, ANTONIO	2320.00
	MARTÍN, MICAELA	1800.00
	PEREZ, JULIO	2200.00
	PEREZ, MARCOS	1800.00
	PONS, CESAR	1030.00
	RUIZ, FABIOLA	1800.00
	SANZ, LAVINIA	2230.00
	TEROL, LUCIANO	1200.00
	VEIGA, JULIANA	1560.00

```

create view ej72b1 as
select nomem, case when numhi>2 then salar+30*(numhi-2) else salar end as
nuevo_salario from temple order by nomem;

/*****/

select * from empresa.ej72b1;

```

	nomem	nuevo_salario
►	DIEZ, AMELIA	1500.00
	FLOR, DOROTEA	1590.00
	GARCIA, OCTAVIO	2230.00
	GIL, GLORIA	2230.00
	LARA, LUCRECIA	1800.00
	LOPEZ, ANTONIO	2320.00
	MARTÍN, MICAELA	1800.00
	PEREZ, JULIO	2200.00
	PEREZ, MARCOS	1800.00
	PONS, CESAR	1030.00
	RUIZ, FABIOLA	1800.00
	SANZ, LAVINIA	2230.00
	TEROL, LUCIANO	1200.00
	VEIGA, JULIANA	1560.00

```

delimiter //
create procedure procedure_ej72b1()
begin
    declare nombre_empleado varchar(50);
    declare salario_actual float;
    declare horas_extras int;
    declare nuevo_salario float;

    declare cur_empleados cursor for select nomem, salar, numhi from temple;

    open cur_empleados;

    fetch cur_empleados into nombre_empleado, salario_actual, horas_extras;

    while (select count(*)) do
        if horas_extras>2 then
            set nuevo_salario=salario_actual+(30*(horas_extras-2));
        else
            set nuevo_salario=salario_actual;
        end if;

        select nombre_empleado, nuevo_salario;

        fetch cur_empleados into nombre_empleado, salario_actual, horas_extras;
    end while;

    close cur_empleados;
end //
delimiter ;

/*****/

call empresa.procedure_ej72b1();

```

nombre_empleado	nuevo_salario
MARTÍN, MICAELA	1800

Result 1 Result 2 Result 3 Result 4 Result 5 Result 6 Result 7 Result 8 Result 9 Result 10 Result 11 Result 12 Result 13 Result 14 x

8. Hallar por orden alfabético los nombres de los empleados tales que, si se les da una gratificación de 60 euros por hijo, el total de esta gratificación no supera a la décima parte de su salario.

```
create view ej8b1 as
select nomem from temple where 60*numhi<=salar/10 order by 1;

/*****/

select * from empresa.ej8b1;
```

nomem
DIEZ, AMELIA
GARCIA, OCTAVIO
GIL, GLORIA
LARA, LUCRECIA
MARTÍN, MICAELA
PEREZ, JULIO
PEREZ, MARCOS
RUIZ, FABIOLA
SANZ, LAVINIA
TEROL, LUCIANO

```
delimiter //
create procedure procedure_ej8b1()
begin
  declare nom_empleado varchar(50);
  declare sal_empleado int;
  declare num_horas int;

  declare cur_empleados cursor for select nomem, salar, numhi from temple;

  create temporary table tmp_nombres(
    nom_empleado varchar(50)
  );

  open cur_empleados;

  fetch cur_empleados into nom_empleado, sal_empleado, num_horas;
```

```

while (select count(*)>0) do
  if (60 * num_horas<=sal_empleado/10) then
    insert into tmp_nombres values(nom_empleado);
  end if;

  fetch cur_empleados into nom_empleado, sal_empleado, num_horas;
end while;

close cur_empleados;

select nom_empleado from tmp_nombres order by 1;

drop temporary table if exists tmp_nombres;
end //
delimiter ;

/*****/

call empresa.procedure_ej8b1();

```



9. Obtener para los departamentos con un presupuesto superior a 5000 euros, su nombre junto con el nombre del centro donde está ubicado.

```

create view ej91b1 as
select d.nomde, c.nomce from tcentr c, tdepto d where c.numce=d.numce and
d.presu>5000;

/*****/

select * from empresa.ej91b1;

```

	nomde	nomce
▶	DIRECCIÓN GENERAL	SEDE CENTRAL
	SECTOR INDUSTRIAL	RELACION CON CLIENTES
	SECTOR SERVICIOS	RELACION CON CLIENTES



```

create view ej92b1 as
select d.nomde, c.nomce from tcentr c, tdepto d where c.numce=d.numce and
d.presu>5000

```



```
union
```

```
select d.nomde, c.nomce from tcentr c join tdepto d on c.numce=d.numce where
d.presu>5000 and c.nomce not in (select c.nomce from tcentr c, tdepto d where
c.numce=d.numce and d.presu>5000)
```

```
/*****
```

```
select * from empresa.ej92b1;
```

	nomde	nomce
►	DIRECCIÓN GENERAL	SEDE CENTRAL
	SECTOR INDUSTRIAL	RELACION CON CLIENTES
	SECTOR SERVICIOS	RELACION CON CLIENTES

10. Para los empleados del departamento de Nóminas, obtener el nombre, salario y número de hijos. Hacer el ejercicio de dos formas: utilizando un producto cartesiano y con la cláusula "join".

```
create view ej101b1 as
select e.nomem, e.salar, e.numhi from temple e, tdepto d where e.numde=d.numde
and d.nomde='nominas';
```

```
/*****
```

```
select * from empresa.ej101b1;
```

	nomem	salar	numhi
►	PEREZ, MARCOS	1800.00	2
	VEIGA, JULIANA	1500.00	4

```
create view ej102b1 as
select e.nomem, e.salar, e.numhi from temple e join tdepto d on e.numde=d.numde
where d.nomde='nominas';
```

```
/*****
```

```
select * from empresa.ej102b1;
```

	nomem	salar	numhi
►	PEREZ, MARCOS	1800.00	2
	VEIGA, JULIANA	1500.00	4

BLOQUE 2

1.