030609

CHRISTIAN MILLÁN SORIA

1° DAW TARDE

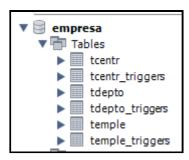
Crear en cada una de las tablas un disparador que registre los movimientos en la tabla para cada una de las siguientes circunstancias:

- 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;
create trigger registro2_tcentr before insert on tcentr
for each row
   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
```

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.

BLOQUE 1

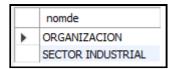
1. Hallar, por orden alfabético, los nombres de los departamentos cuyo director lo es en funciones y no en propiedad.

```
drop view if exists ej1b1;
create view ej1b1 as
select nomde from tdepto where tidir='f' order by 1;
/**********************************
select * from empresa.ej1b1;
```

```
nomde

ORGANIZACION
SECTOR INDUSTRIAL
```

^{*}Utilizar cursores y bucles siempre que sea posible.



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.

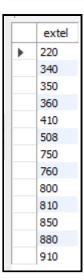
```
drop view if exists ej2b1;
create view ej2b1 as
select nomem, numem, extel from temple where numde=121 order by 1 desc;
/******************************
select * from empresa.ej2b1;
```

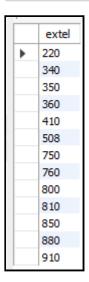
	nomem	numem	extel
•	RUIZ, FABIOLA	370	360
	PONS, CESAR	110	350
	PEREZ, JULIO	150	340

	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.

```
select * from empresa.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.

	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

```
drop procedure if exists procedure_ej4b1;
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, ∅), 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
```

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

Result 6

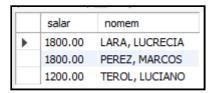
Result 7

Result 8

Result 9

Result 10 ×

Result 5



Result 2

Result 3

Result 4

Result 1

```
drop procedure if exists procedure_ej5b1;
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;

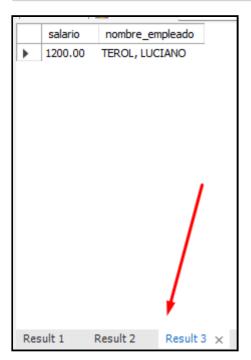
open cur_datos;
fetch cur_datos into salario, nombre_empleado;

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

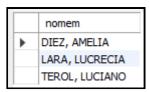
close cur_datos;
end //
delimiter;

/******************************/

call empresa.procedure_ej5b1();
```



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





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.

	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

```
drop procedure if exists procedure_ej7b1;
delimiter //
create procedure procedure_ej7b1()
```

```
begin
  select nomem, salar+30*(numhi-2) from temple where numhi>2 order by 1;
end //
delimiter;

/**************************

call empresa.procedure_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

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".

_		
	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

```
drop procedure if exists procedure_ej71b1;
delimiter //
create procedure procedure_ej71b1()
 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

```
drop view if exists ej72b1;
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

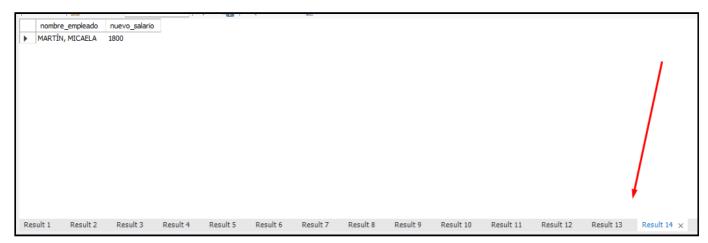
```
drop procedure if exists procedure_ej72b1;
delimiter //
create procedure procedure_ej72b1()
begin
   declare nombre_empleado varchar(50);
   declare salario_actual float;
   declare horas_extras int;
   declare nuevo_salario float;

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;

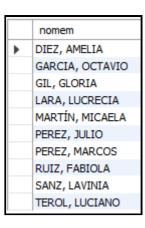
    fetch cur_empleados into nombre_empleado, salario_actual, horas_extras;
    end while;

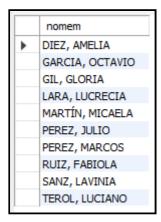
    close cur_empleados;
end //
delimiter;
/*********************************/
call empresa.procedure_ej72b1();
```



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.

```
drop view if exists ej8b1;
create view ej8b1 as
select nomem from temple where 60*numhi<=salar/10 order by 1;
/**********************************
select * from empresa.ej8b1;</pre>
```





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

```
drop view if exists ej9b1;
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

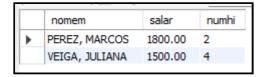
_		
	nomde	nomce
•	DIRECCIÓN GENERAL	SEDE CENTRAL
	SECTOR INDUSTRIAL	RELACION CON CLIENTES
	SECTOR SERVICIOS	RELACION CON CLIENTES

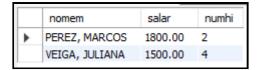
	nomde	nomce
•	DIRECCIÓN GENERAL	SEDE CENTRAL
	SECTOR INDUSTRIAL	RELACION CON CLIENTES
	SECTOR SERVICIOS	RELACION CON CLIENTES

```
drop procedure if exists procedure_ej92b1;
delimiter //
create procedure procedure_ej92b1()
begin
  select d.nomde, c.nomce from tcentr c, tdepto d where c.numce=d.numce and
```



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".





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

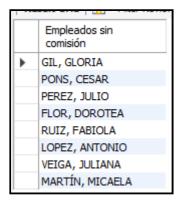
	nomem	salar	numhi
•	PEREZ, MARCOS	1800.00	2
	VEIGA, JULIANA	1500.00	4

BLOQUE 2

1. Hallar el nombre de los empleados que no tienen comisión, clasificados de manera que aparezcan primero aquellos nombres que son más cortos.

```
select * from empresa.ej1b2;
```

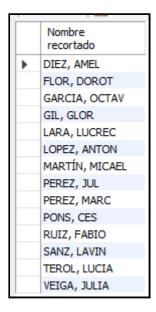




2. Hallar, por orden alfabético, los nombres de los empleados suprimiendo las dos últimas letras.

```
drop view if exists ej2b2;
create view ej2b2 as
select substring(nomem, 1, length(nomem)-2) as 'Nombre recortado' from temple
order by 1;
/******************************
select * from empresa.ej2b2;
```





3. Hallar cuántos departamentos hay y el presupuesto anual medio de ellos.

```
drop view if exists ej3b2;
create view ej3b2 as
select count(distinct nomde)as '№ de departamentos',
```

```
round(sum(presu)/count(distinct nomde), 2) as 'Presupuesto anual medio' from
tdepto;

/***********************

select * from empresa.ej3b2;
```

1		
	Nº de departamentos	Presupuesto anual medio
•	6	7000.00

	Nº de departamentos	Presupuesto anual medio
•	6	7000.00

4. Hallar la masa salarial anual (salario más comisión) de la empresa (se suponen 14 pagas anuales).

```
drop view if exists ej4b2;
create view ej4b2 as
select sum((salar+comis)*14) as 'Masa salarial' from temple;

/*************************
select * from empresa.ej4b2;
```

```
Masa salarial

• 224000.00
```

```
drop procedure if exists procedure_ej4b2;
delimiter //
```

```
create procedure procedure_ej4b2()
begin
   select sum((salar+comis)*14) as 'Masa salarial' from temple;
end //
delimiter;

/*************************

call empresa.procedure_ej4b2();
```



5. Hallar la masa salarial anual (salario más comisión) de cada empleado (se suponen 14 pagas anuales). Hacer el ejercicio de diferentes maneras: con el operador "union", con la función "ifnull", con la función "if" y con la expresión "case".

```
drop view if exists ej5b2;
create view ej5b2 as
```



```
drop procedure if exists procedure_ej5b2;
delimiter //
create procedure_ej5b2()
begin
end //
delimiter;
```



"ifnull":

```
drop view if exists ej51b2;
create view ej51b2 as
```



```
drop procedure if exists procedure_ej51b2;
delimiter //
```

```
create procedure_ej51b2()
begin
end //
delimiter;
```



"if":

```
drop view if exists ej52b2;
create view ej52b2 as
```



```
drop procedure if exists procedure_ej52b2;
delimiter //
create procedure_ej52b2()
begin
end //
delimiter;
```



"case":

```
drop view if exists ej53b2;
create view ej53b2 as
```



```
drop procedure if exists procedure_ej53b2;
delimiter //
create procedure_ej53b2()
begin
end //
delimiter;
```

