

IES Virrey Morcillo. (Nombre Ciclo Formativo)

Procedimientos a funciones

Módulo: BDD

Nombre del Alumno:

Fecha:

Contenido

Criterios de Evaluación 1

Material 1

Objetivos 1

1. Introducción 1

- 2. Tareas 2
- 3. Conclusión 2
- 4. Links 2

Criterios de Evaluación

- CE.x.x) ...

Material

Haremos uso de los siguientes recursos para realizar las tareas:

- Ordenador
- Conexión de Internet
- Procesador de Textos

Objetivos

- Conocer ...
- Repasar ...

1. Introducción

Actividad) Haciendo uso de la base de datos llamada BDEMPLEADOSDNIAlumno y de sus tablas:

EMPLEADOS (DNI, Nombre, Apellidos, Departamento)

DEPARTAMENTOS (Codigo, Nombre, Presupuesto)

Crear procedimientos con **FUNCIONES para los siguientes casos, escribir su código en cada**

apartado y una captura de pantalla mostrando su solución al ser ejecutado, todos los procedimientos deben empezar por “DNIAlumno-ProcedimientoNº”:

2. Tareas

1. Introducir los datos de un empleado indicando sus valores por parámetros

DL:

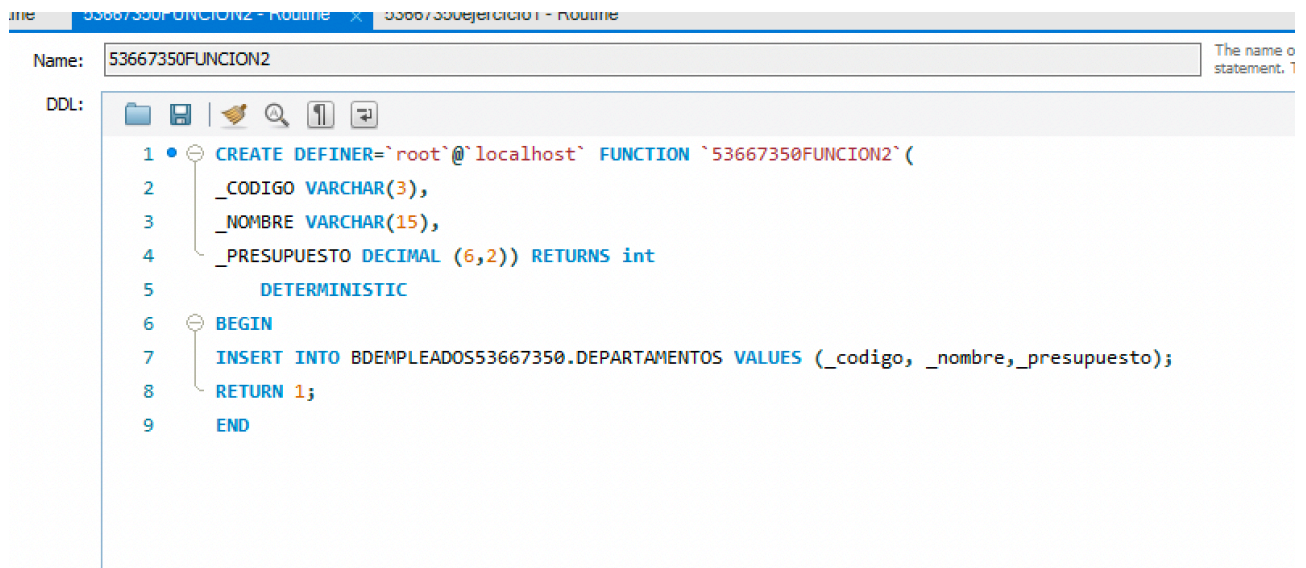
```
1 • CREATE DEFINER='root'@'localhost' FUNCTION `53667350ejercicio1`(  
2   _DNI varchar(9),  
3   _nombre varchar(10),  
4   _apellidos varchar(15),  
5   _departamento varchar(3)) RETURNS tinyint(1)  
6     DETERMINISTIC  
7   BEGIN  
8     DECLARE COMPLETADO BOOLEAN DEFAULT 0;  
9     INSERT INTO EMPLEADOS values(_DNI, _NOMBRE, _APELLIDOS, _DEPARTAMENTO);  
10    SET completado = 1;  
11    RETURN COMPLETADO;  
12  END
```

53667350ejercicio1

```
1 • CREATE DEFINER='root'@'localhost' FUNCTION `53667350ejercicio1`(  
2   _DNI varchar(9),  
3   _nombre varchar(10),  
4   _apellidos varchar(15),  
5   _departamento varchar(3)) RETURNS tinyint(1)  
6     DETERMINISTIC  
7   BEGIN  
8     DECLARE COMPLETADO tinyint(1) DEFAULT '0';  
9     INSERT INTO EMPLEADOS values(_DNI, _NOMBRE, _APELLIDOS, _DEPARTAMENTO);  
10    IF _DNI > "" THEN RETURN 1;  
11    ELSE RETURN 0;  
12  END IF;  
13  END
```

**ASÍ ME ASEGURO DE QUE
EL DNI QUE ES CLAVE PRINCIPAL
NO SE QUEDE VACÍO**

2. Introducir los datos de un departamento indicando sus valores por parámetros.



```
1 CREATE DEFINER=`root`@`localhost` FUNCTION `53667350FUNCION2` (  
2     _CODIGO VARCHAR(3),  
3     _NOMBRE VARCHAR(15),  
4     _PRESUPUESTO DECIMAL (6,2)) RETURNS int  
5     DETERMINISTIC  
6 BEGIN  
7     INSERT INTO BDEMPLEADOS53667350.DEPARTAMENTOS VALUES (_codigo, _nombre, _presupuesto);  
8     RETURN 1;  
9 END
```

3. Obtener los datos de los empleados según la Inicial del Apellido. NOTA (POSIBLEMENTE APLICA A OTROS EJERCICIOS): no todos los datos, solamente sacamos el apellido.

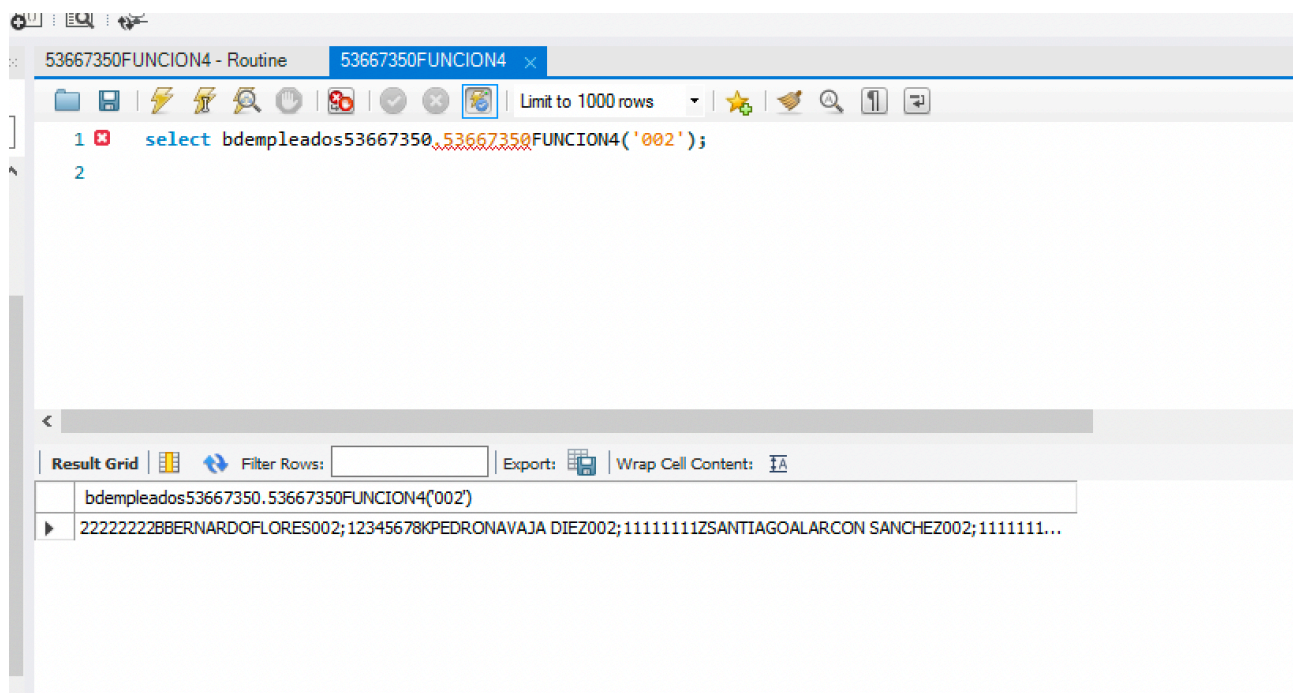
He usado tal cual el código visto en clase:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `53667350EJERCICIO3` (LETRA VARCHAR(1))  
RETURNS varchar(4000) CHARSET utf8mb4  
    READS SQL DATA  
BEGIN  
    declare datos varchar(4000) default "";  
    declare resultado varchar(4000) default "";  
    declare final integer;  
    declare apellidov varchar(15) default "";  
    declare cursor1 cursor for SELECT apellidos from empleados where apellidos like  
    concat(letra, '%');  
    declare continue handler for not found set final = 1;  
    declare continue handler for sqlexception set resultado = "Error de excepcion";  
    open cursor1;  
    bucle: loop  
        fetch cursor1 into apellidov;  
        if final = 1 then leave bucle;  
        end if;  
        set datos = concat(apellidov, "--", datos);  
    end loop bucle;  
    close cursor1;  
    select datos into resultado;  
    return resultado;  
RETURN 1;  
END
```

(Me saca los apellidos de todos los empleados que empizan por letra)

4. Obtener todos los datos de los empleados que trabajan para según el número de departamento. Es decir, querré usar un cursor para sacar todos los datos.

```
CREATE FUNCTION `53667350FUNCION4` (CODIGO_DEP VARCHAR(3)) RETURNS
VARCHAR(4000)
DETERMINISTIC
BEGIN
DECLARE _DNI VARCHAR(9);
DECLARE _NOMBRE VARCHAR(10);
DECLARE _APELLIDOS VARCHAR(15);
DECLARE _DEPARTAMENTO VARCHAR (15);
DECLARE MUCHEDUMBRE VARCHAR(4000) DEFAULT "";
DECLARE TERMINAU INTEGER DEFAULT 0;
DECLARE CURSOR4
CURSOR FOR
SELECT * FROM EMPLEADOS WHERE DEPARTAMENTO = CODIGO_DEP;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET TERMINAU = 1;
OPEN CURSOR4;
SACARDATOS: LOOP
FETCH CURSOR4 INTO _DNI, _NOMBRE, _APELLIDOS, _DEPARTAMENTO;
IF TERMINAU = 1
THEN LEAVE SACARDATOS;
END IF;
SET MUCHEDUMBRE = CONCAT(_DNI, _NOMBRE, _APELLIDOS, _DEPARTAMENTO, ";",
MUCHEDUMBRE);
END LOOP SACARDATOS;
RETURN MUCHEDUMBRE;
END
```



5. Obtener el presupuesto de los departamentos según el nombre del departamento.

```
CREATE FUNCTION `53667350FUNCION5` (_nombre varchar(15)) returns decimal(6,2)
deterministic
BEGIN
DECLARE RESULTADO DECIMAL(6,2);
```

```

SELECT PRESUPUESTO INTO RESULTADO FROM DEPARTAMENTOS WHERE NOMBRE LIKE
_NOMBRE;
RETURN RESULTADO;
END

```

The screenshot shows a database IDE with a toolbar at the top containing icons for file operations, execution, and search. Below the toolbar, a SQL query is entered in a text area:

```

1 select bdempleados53667350.53667350FUNCION5('VENTAS');
2

```

Below the query editor, a "Result Grid" is displayed. It includes a "Filter Rows" input field, an "Export" button, and a "Wrap Cell Content" checkbox. The grid contains the following data:

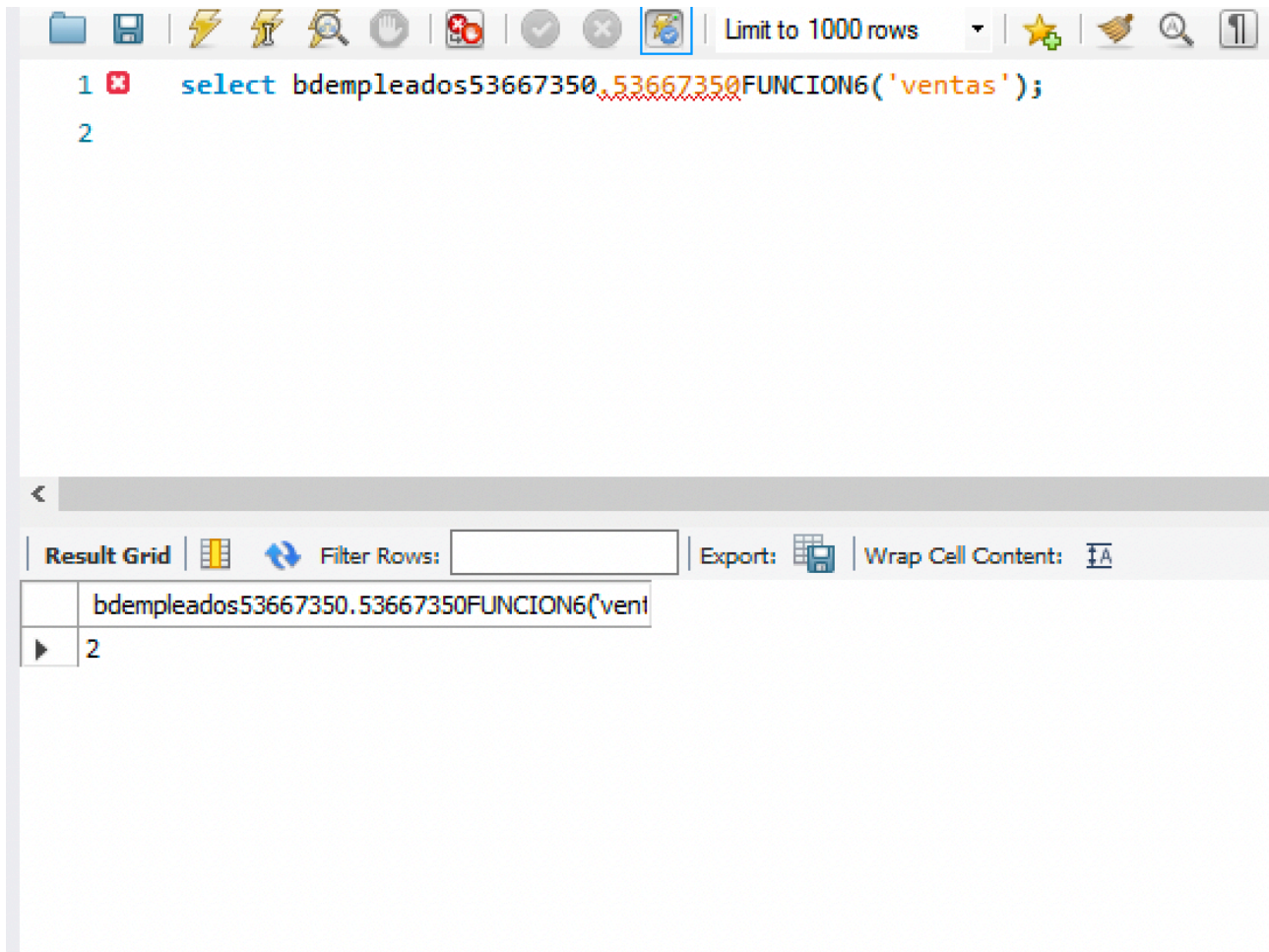
	bdempleados53667350.53667350FUNCION5('VEN'
	800.00

6. Obtener el nº de empleados según el nombre de departamento.

```

CREATE FUNCTION `53667350FUNCION6` (NOMBREDEP VARCHAR(15))
RETURNS INTEGER
deterministic
BEGIN
DECLARE RESULTADO INTEGER DEFAULT 0;
SELECT COUNT(*) INTO RESULTADO FROM EMPLEADOS WHERE DEPARTAMENTO =
(SELECT CODIGO FROM DEPARTAMENTOS WHERE NOMBRE LIKE NOMBREDEP);
RETURN RESULTADO;
END

```

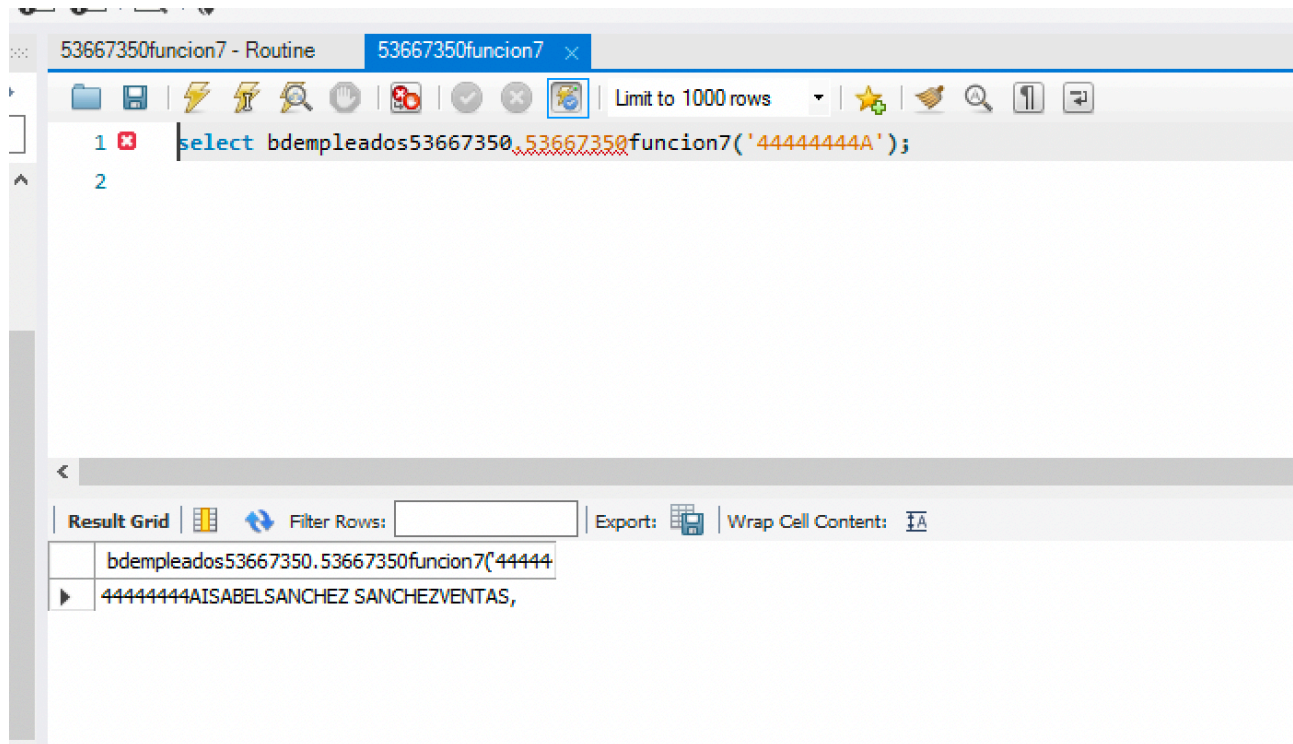
7. Obtener un listado completo de empleados, incluyendo los datos del empleado y el nombre de su departamento según su DNI. He hecho el código siguiendo como construí el procedimiento pero no me devuelve nada.

```
CREATE DEFINER='root'@'localhost' FUNCTION `53667350funcion7`(_dni varchar(9)) RETURNS
varchar(4000) CHARSET utf8mb4
  DETERMINISTIC
BEGIN
declare datos varchar(4000) default "";
declare resultado varchar(4000) default "";
declare final integer;
DECLARE DNIB VARCHAR(9);
DECLARE _NOMBRE VARCHAR(10);
DECLARE _APELLIDOS VARCHAR(15);
DECLARE NOMBREDEP VARCHAR(15);
DECLARE CURSOR7
CURSOR FOR
SELECT dni, EMPLEADOS.nombre, apellidos, departamentos.nombre FROM EMPLEADOS
RIGHT JOIN DEPARTAMENTOS
ON DEPARTAMENTOS.CODIGO = EMPLEADOS.DEPARTAMENTO
WHERE DNI LIKE _DNI;
declare continue handler for not found set final = 1;
declare continue handler for sqlexception set resultado = "Error de excepcion";
OPEN CURSOR7;
VUELTA: LOOP
```

```

FETCH CURSOR7 INTO DNIB,_NOMBRE,_APELLIDOS, NOMBREDEP;
IF FINAL = 1 THEN LEAVE VUELTA;
END IF;
SET DATOS = CONCAT(DNIB, _NOMBRE,_APELLIDOS,NOMBREDEP, ",", DATOS);
END LOOP VUELTA;
CLOSE CURSOR7;
SELECT DATOS INTO RESULTADO;
RETURN RESULTADO;
END

```



8. Obtener los nombres y apellidos de los empleados que trabajen en departamentos

cuyo valor indicado por parámetro sea mayor que su presupuesto.

```

CREATE DEFINER=`root`@`localhost` FUNCTION `53667350FUNCION8`(_presupuesto
decimal(6,2)) RETURNS varchar(4000) CHARSET utf8mb4
    DETERMINISTIC
BEGIN
    declare datos varchar(4000) default "";
    declare resultado varchar(4000) default "";
    declare final integer;
    DECLARE _NOMBRE VARCHAR(10);
    DECLARE _APELLIDOS VARCHAR(15);
    DECLARE CURSOR8
    CURSOR FOR
    SELECT NOMBRE, APELLIDOS FROM EMPLEADOS WHERE
    DEPARTAMENTO IN
    (SELECT CODIGO FROM DEPARTAMENTOS WHERE PRESUPUESTO < _PRESUPUESTO);
    declare continue handler for not found set final = 1;
    declare continue handler for sqlexception set resultado = "Error de excepcion";
    OPEN CURSOR8;
    VUELTA: LOOP
    FETCH CURSOR8 INTO _NOMBRE, _APELLIDOS;
    IF FINAL = 1 THEN LEAVE VUELTA;
    END IF;

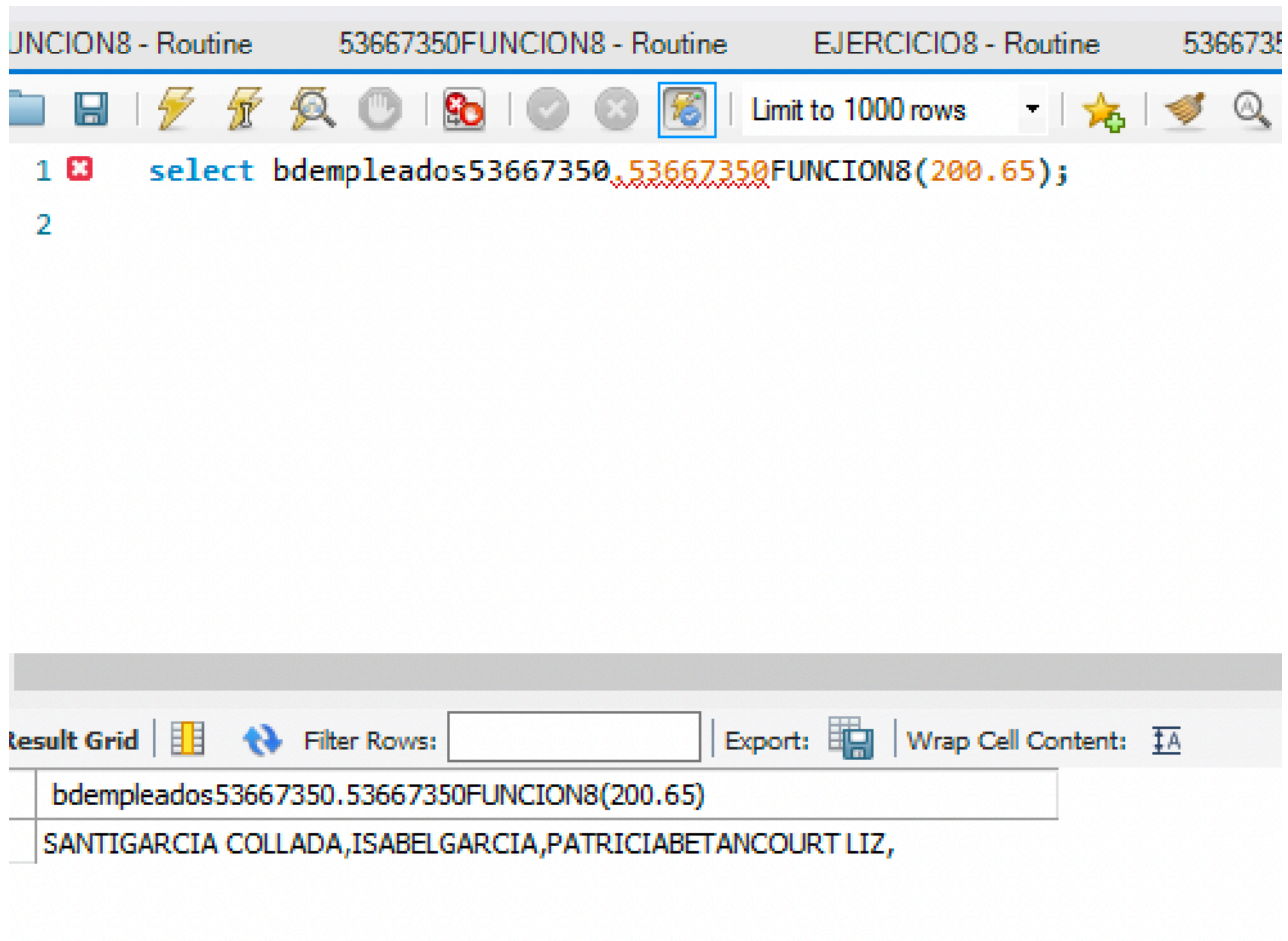
```



```

SET DATOS = CONCAT(_NOMBRE,_APELLIDOS,"", DATOS);
END LOOP VUELTA;
CLOSE CURSOR8;
SELECT DATOS INTO RESULTADO;
RETURN RESULTADO;
END

```



The screenshot shows a SQL IDE interface. At the top, there are tabs for 'FUNCION8 - Routine', '53667350FUNCION8 - Routine', 'EJERCICIO8 - Routine', and '5366735'. Below the tabs is a toolbar with various icons. The main area displays a SQL query:

```

1 select bdempleados53667350.53667350FUNCION8(200.65);
2

```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid shows the following data:

bdempleados53667350.53667350FUNCION8(200.65)
SANTIGARCIA COLLADA,ISABELGARCIA,PATRICIABETANCOURT LIZ,

9. Obtener los datos de los departamentos cuyo presupuesto dado por parámetro sea **superior al presupuesto medio de todos los departamentos.**

10. Obtener los nombres (sólo los nombres) de los departamentos que tienen el número de empleados indicado por parámetro.

Me falla la definición del cursor porque me fallaba el procedimiento en cuestión:

```

CREATE DEFINER='root'@'localhost' PROCEDURE `PROCEDIMIENTO10`(IN NEMPLEADOS
INTEGER)
BEGIN
SELECT NOMBRE FROM EMPLEADOS as e WHERE
DEPARTAMENTO IN
(SELECT DEPARTAMENTO FROM EMPLEADOS having COUNT(e.departamento) LIKE
NEMPLEADOS);
END

```

11. Aplicar un recorte presupuestario de un valor indicado por parámetro a todos los **departamentos.**

12. Reasignar a los empleados de un departamento a otro indicando los códigos de departamento como parámetros.

```
CREATE FUNCTION `53667350FUNCION11` (DEPARTAMENTO1 VARCHAR(3), DEPARTAMENTOF VARCHAR(3))
RETURNS INTEGER
DETERMINISTIC
BEGIN
declare FINAL INTEGER;
declare final integer;
DECLARE CURSOR11
CURSOR FOR
SELECT DEPARTAMENTO FROM EMPLEADOS WHERE DEPARTAMENTO = DEPARTAMENTO1;
declare continue handler for not found set final = 1;
OPEN CURSOR11;
VUELTA: LOOP
FETCH CURSOR11 INTO DEPARTAMENTO1;
IF FINAL=1 THEN LEAVE VUELTA;
END IF;
UPDATE EMPLEADOS SET DEPARTAMENTO = DEPARTAMENTOF WHERE DEPARTAMENTO
LIKE DEPARTAMENTO1;
END LOOP VUELTA;
CLOSE CURSOR11;
RETURN 1;
END
```

The screenshot shows a database grid with columns: I, nombre, apellidos, and departa. The data includes employees like Pedro Páramo (003), Patricia Betancourt Liz (001), Santiago Garcia (002), Carlos Lorens (004), Santiago Alarcon Sanchez (002), Pedro Navaia Diez (002), Isabel Garcia (001), Bernardo Flores (002), Isabel Sanchez Sanchez (003), Santi Garcia Collada (001), and Santiago Paramo (004).

Overlaid on the grid is a dialog box titled "Call stored function bdempleados53667350.53667350FUN...". It contains the instruction: "Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:". The parameters are: DEPARTAMENTO1 (003) and DEPARTAMENTOF (004), both of type VARCHAR(3). There are "Execute" and "Cancel" buttons at the bottom right.

✓	53	23:29:39	SELECT FROM bdempleados53667350.emple...	11 row(s) returned
✗	54	23:30:45	select bdempleados53667350.53667350FUNC...	Error Code: 1328. Incorrect nu
✓	55	23:31:42	Apply changes to 53667350FUNCION11	Changes applied
4	56	23:32:01	select bdempleados53667350.53667350FUNC...	Running...

Se me queda pillada la máquina virtual por lo que debo haber confundido las variables dentro del bucle.

13. Despedir a todos los empleados que trabajan para un departamento indicado por parámetro.

(De este no tengo captura porque no me responde la máquina virtual)

```
CREATE FUNCTION `53667350FUNCION12` (_DEPARTAMENTO VARCHAR(3)) RETURNS  
VARCHAR(4000)  
DETERMINISTIC  
BEGIN  
declare FINAL INTEGER;  
declare datos varchar(4000) default "";  
declare ELIMINACIONES varchar(4000) default "";  
DECLARE DNIB VARCHAR(9);  
DECLARE _NOMBRE VARCHAR(10);  
DECLARE _APELLIDOS VARCHAR(15);  
DECLARE NOMBREDEP VARCHAR(15);  
DECLARE CURSOR12  
CURSOR FOR  
SELECT * FROM EMPLEADOS WHERE DEPARTAMENTO = _DEPARTAMENTO;  
declare continue handler for not found set final = 1;  
OPEN CURSOR12;  
VUELTA: LOOP  
FETCH CURSOR12 INTO dnib, _nombre, _apellidos, _nombredep;  
IF FINAL=1 THEN LEAVE VUELTA;  
END IF;  
SET DATOS = CONCAT(DNIB, _NOMBRE, _APELLIDOS, NOMBREDEP, ",", DATOS);  
END LOOP VUELTA;  
CLOSE CURSOR12;  
DELETE EMPLEADOS WHERE DEPARTAMENTO LIKE DEPARTAMENTO1;  
SELECT DATOS INTO ELIMINACIONES;  
RETURN ELIMINACIONES;  
END
```

14. Despedir a todos los empleados que trabajen para departamentos cuyo presupuesto sea superior a un valor establecido por parámetro.

3. Conclusión

El alumno redactará sus impresiones sobre la realización de estas tareas

4. Links

Indicamos los sitios web visitados para poder realizar las actividades.