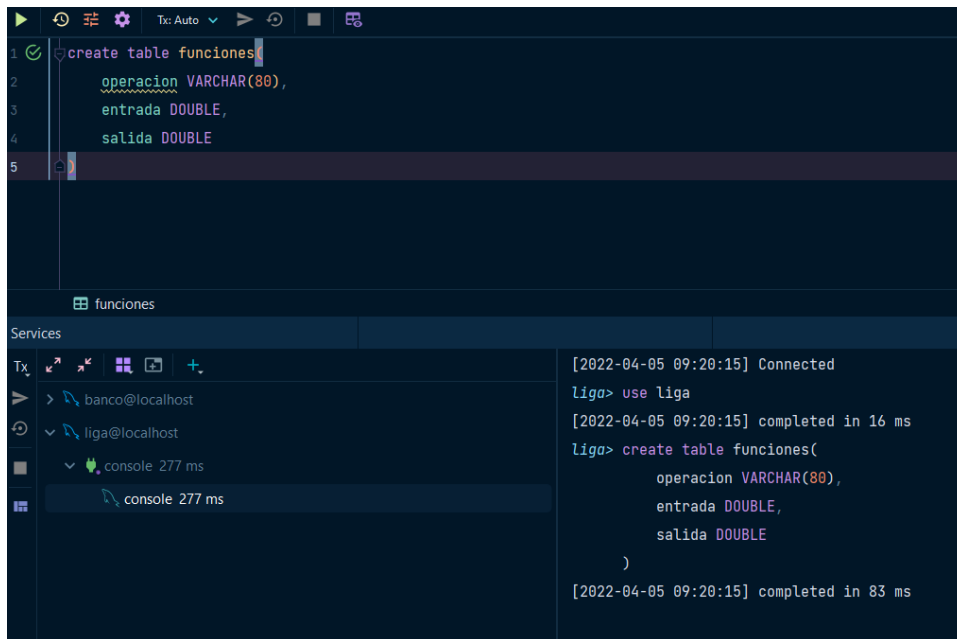


Unidad 6 Construcción de scripts
Daniel Rodríguez Fernández

PRACTICA 1

Sobre la BBDD 'Liga', crea una tabla llamada funciones formada por tres campos:

- Operación de tipo VARCHAR.
- Entrada de tipo Double.
- Salida de tipo Double.



```
1 create table funciones
2     operacion VARCHAR(80),
3     entrada DOUBLE,
4     salida DOUBLE
5
```

funciones

Services

Tx: Auto

banco@localhost

liga@localhost

console 277 ms

console 277 ms

[2022-04-05 09:20:15] Connected

liga> use liga

[2022-04-05 09:20:15] completed in 16 ms

liga> create table funciones(

operacion VARCHAR(80),

entrada DOUBLE,

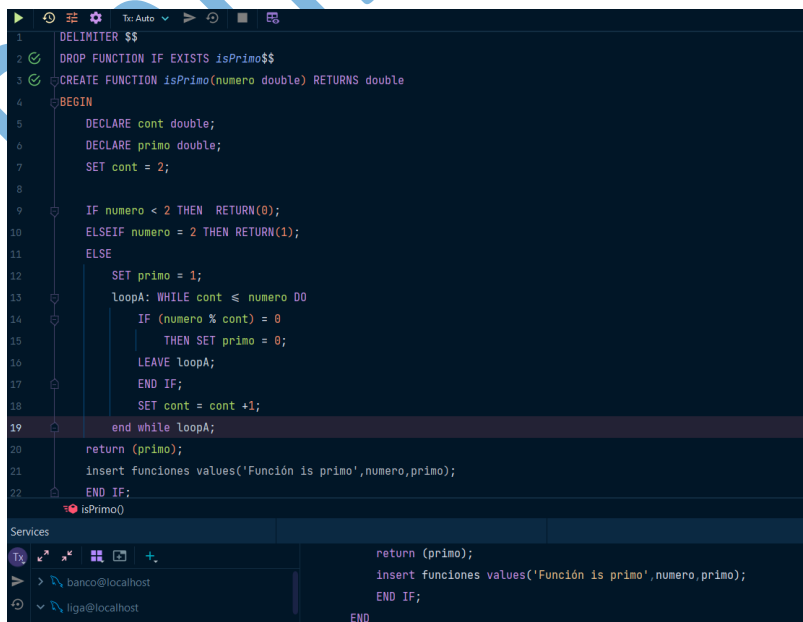
salida DOUBLE

)

[2022-04-05 09:20:15] completed in 83 ms

A continuación construye los siguientes scripts:

1. Crea una función que determine si un número es primo devolviendo 0 o 1. La función también deberá insertar los valores adecuados en la tabla funciones antes de finalizar.



```
1 DELIMITER $$
2 DROP FUNCTION IF EXISTS isPrimo$$
3 CREATE FUNCTION isPrimo(numero double) RETURNS double
4 BEGIN
5     DECLARE cont double;
6     DECLARE primo double;
7     SET cont = 2;
8
9     IF numero < 2 THEN RETURN(0);
10    ELSEIF numero = 2 THEN RETURN(1);
11    ELSE
12        SET primo = 1;
13        loopA: WHILE cont <= numero DO
14            IF (numero % cont) = 0
15            THEN SET primo = 0;
16            LEAVE loopA;
17            END IF;
18            SET cont = cont +1;
19        end while loopA;
20        return (primo);
21        insert funciones values('Función is primo',numero,primo);
22    END IF;
23    isPrimo0
24
25    return (primo);
26    insert funciones values('Función is primo',numero,primo);
27    END IF;
28    END
```

Services

banco@localhost

liga@localhost

console 277 ms

console 277 ms

[2022-04-05 09:20:15] Connected

liga> use liga

[2022-04-05 09:20:15] completed in 16 ms

liga> create table funciones(

operacion VARCHAR(80),

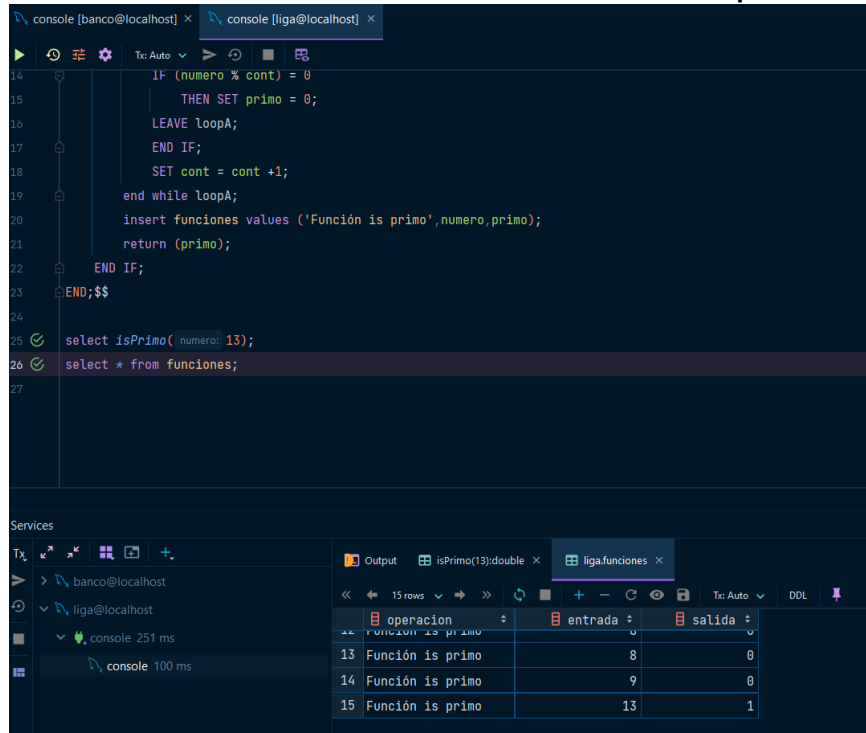
entrada DOUBLE,

salida DOUBLE

)

[2022-04-05 09:20:15] completed in 83 ms

Unidad 6 Construcción de scripts



DELIMITER \$\$

DROP FUNCTION IF EXISTS isPrimo\$\$

CREATE FUNCTION isPrimo(numero double) RETURNS double
BEGIN

DECLARE cont double;

DECLARE primo double;

SET cont = 2;

IF numero < 2 THEN RETURN(0);

ELSEIF numero = 2 THEN RETURN(1);

ELSE

SET primo = 1;

loopA: WHILE numero > cont DO

IF (numero % cont) = 0

THEN SET primo = 0;

LEAVE loopA;

END IF;

SET cont = cont +1;

end while loopA;

insert funciones values ('Función is primo',numero,primo);

return (primo);

END IF;

END;\$\$

select isPrimo(13);

select * from funciones;

Unidad 6 Construcción de scripts

2. Crea una función, llamada sucesión, que calcule la suma de los términos $1/n$ con n entre 1 y n , siendo n el parámetro de entrada. Ejemplo: $n=5$

$$1/1+1/2+1/3+1/4+1/5$$

La función también deberá insertar los valores adecuados en la tabla operaciones antes de finalizar.

DELIMITER \$\$

DROP FUNCTION IF EXISTS sumaTerminos \$\$

CREATE FUNCTION sumaTerminos(numero double) RETURNS int

BEGIN

DECLARE total int;

DECLARE contador int;

SET total = 1;

SET contador = 1;

WHILE contador<=numero DO

SET total=total+contador;

SET contador=contador+1;

END WHILE;

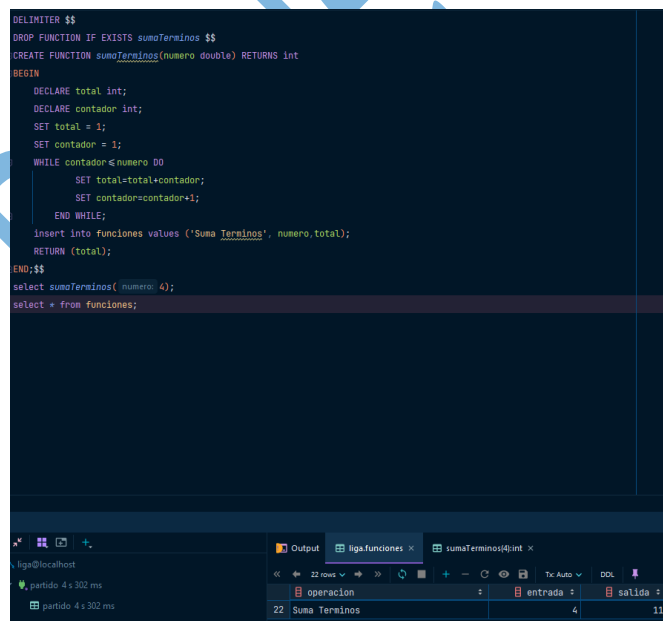
insert into funciones values ('Suma Terminos', numero,total);

RETURN (total);

END;\$\$

select sumaTerminos(4);

select * from funciones;



The screenshot shows a MySQL command line interface with a dark background. The script is being executed line by line. The output shows the function being created successfully. The final output shows the result of the function call and the contents of the 'funciones' table.

```
DELIMITER $$
DROP FUNCTION IF EXISTS sumaTerminos $$
CREATE FUNCTION sumaTerminos(numero double) RETURNS int
BEGIN
  DECLARE total int;
  DECLARE contador int;
  SET total = 1;
  SET contador = 1;
  WHILE contador<=numero DO
    SET total=total+contador;
    SET contador=contador+1;
  END WHILE;
  insert into funciones values ('Suma Terminos', numero,total);
  RETURN (total);
END;$$
select sumaTerminos(4);
select * from funciones;
```

The output shows the function being created successfully. The final output shows the result of the function call and the contents of the 'funciones' table.

operacion	entrada	salida
Suma Terminos	4	11

Unidad 6 Construcción de scripts

Sobre la BBDD 'Liga' construye los siguientes scripts:

1. Crea una rutina que devuelva el número de partidos jugados como local y el número de partidos jugados como visitante, de un equipo pasado como parámetro.

```
DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS visitanteLocal $$
```

```
CREATE PROCEDURE contarVisitante(inout equipo varchar(12))
```

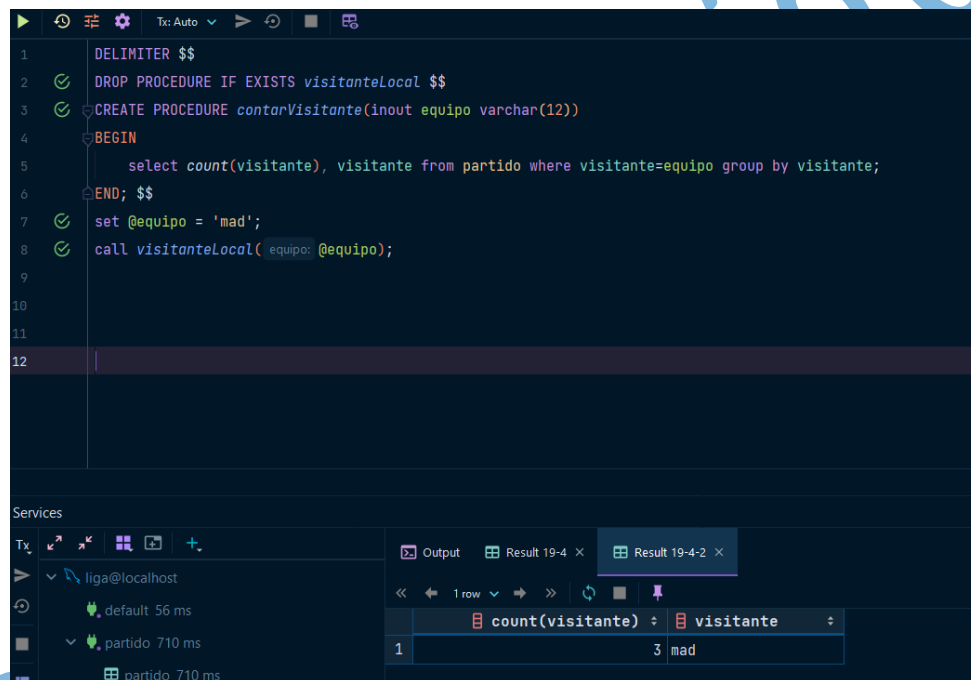
```
BEGIN
```

```
    select count(visitante), visitante from partido where visitante=equipo  
    group by visitante;
```

```
END; $$
```

```
set @equipo = 'mad';
```

```
call visitanteLocal(@equipo);
```



```
DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS contarLocal $$
```

```
CREATE PROCEDURE contarLocal(in equipo varchar(12))
```

```
BEGIN
```

```
    select count(local), local from partido where local=equipo group by  
    local;
```

```
END; $$
```

```
set @equipo = 'sev';
```

```
call contarLocal(@equipo);
```

Unidad 6 Construcción de scripts

```

1 DELIMITER $$
2 DROP PROCEDURE IF EXISTS contarLocal $$
3 CREATE PROCEDURE contarLocal(inout equipo varchar(12))
4 BEGIN
5     select count(local), local from partido where local=equipo group by local;
6 END; $$
7 set @equipo = 'sev';
8 call contarLocal( equipo: @equipo);

```

Services

liga@localhost

default 56 ms

partido 710 ms

partido 710 ms

Output

Result 22-4

Result 19-4-2

1 row

count(local)	local
1	sev

Otra opción:

```

DELIMITER $$
DROP FUNCTION IF EXISTS contarEquipoGanador$$
CREATE FUNCTION contarEquipoGanador(nomEquipo char(10)) RETURNS VARCHAR(50)
BEGIN
    DECLARE equipoLocal double;
    DECLARE equipoVisitante double;
    DECLARE equipo char(8);
    DECLARE solucion varchar(50);
    SELECT count(*) into equipo FROM equipo WHERE nombre = nomEquipo;
    IF equipo <= 1 THEN
        set solucion = 'No existe equipo con puntuación';
    ELSE
        SELECT count(*) INTO equipoLocal FROM partido WHERE local = nomEquipo;
        SELECT count(*) INTO equipoVisitante FROM partido WHERE visitante = nomEquipo;
        SELECT concat('Nº Partidos Local: ', equipoLocal, ' | Nº partidos visitante: ', equipoVisitante) into solucion;
    END IF;
    RETURN solucion;
END; $$
select contarEquipoGanador( nomEquipo: 'sev');

```

liga@localhost

console_1 76 ms

console_1 76 ms

partido

Output

contarEquipoGanador('sev');varchar(50)

1 row

1	Nº Partidos Local: 1 Nº partidos visitante: 1
---	---

DELIMITER \$\$**DROP FUNCTION IF EXISTS contarEquipoGanador\$\$****CREATE FUNCTION contarEquipoGanador(nomEquipo char(10)) RETURNS VARCHAR(50)****BEGIN****DECLARE equipoLocal double;****DECLARE equipoVisitante double;****DECLARE equipo char(8);**

Unidad 6 Construcción de scripts

```

DECLARE solucion varchar (50);
SELECT count(*) into equipo FROM equipo WHERE nombre = nomEquipo;
IF equipo !=1 THEN
    set solucion = 'No existe equipo con puntuación';
ELSE
    SELECT count(*) INTO equipoLocal FROM partido WHERE local = nomEquipo;
    SELECT count(*) INTO equipoVisitante FROM partido WHERE visitante =
nomEquipo;
    SELECT concat('Nº Partidos Local: ', equipoLocal, ' | Nº partidos
visitante: ', equipoVisitante) into solucion;
END IF;
RETURN solucion;
END; $$
select contarEquipoGanador('sev');

```

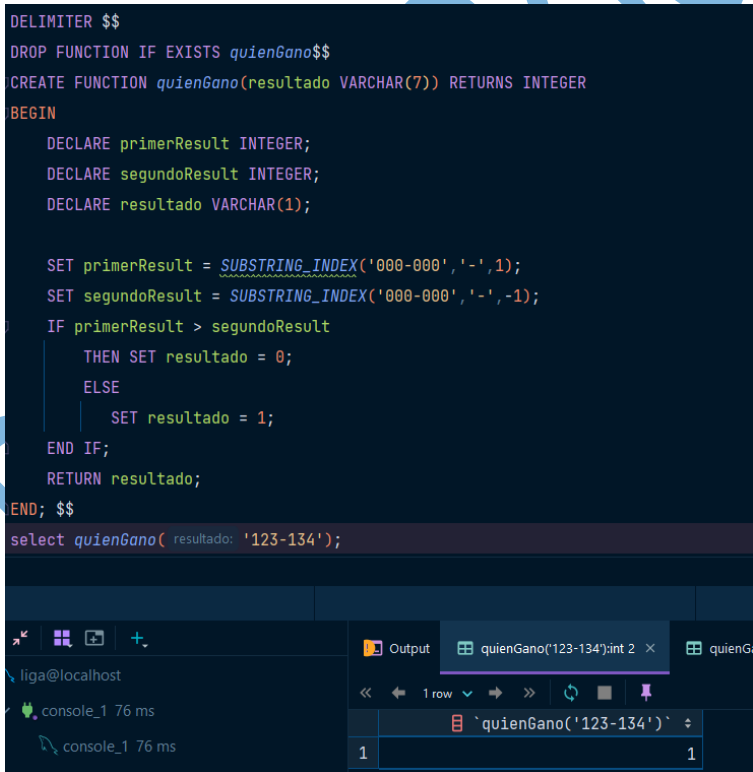
2. Crea una rutina que devuelva 1 si gana el equipo visitante y 0 en caso contrario. El parámetro de entrada es el resultado con el formato xxx-xxx.

```

DELIMITER $$
DROP FUNCTION IF EXISTS quienGano$$
CREATE FUNCTION quienGano(resultado VARCHAR(7)) RETURNS INTEGER
BEGIN
    DECLARE primerResult INTEGER;
    DECLARE segundoResult INTEGER;
    DECLARE resultado VARCHAR(1);

    SET primerResult = SUBSTRING_INDEX('000-000', '-', 1);
    SET segundoResult = SUBSTRING_INDEX('000-000', '-', -1);
    IF primerResult > segundoResult
    THEN SET resultado = 0;
    ELSE
        SET resultado = 1;
    END IF;
    RETURN resultado;
END; $$
select quienGano( resultado: '123-134');

```



The screenshot shows a SQL IDE interface. The top pane contains the SQL script for the 'quienGano' function. The bottom pane shows the execution results. The 'Output' window displays the result of the function call: '1'. The 'console' window shows the execution time: '76 ms'.

quienGano('123-134')
1

Unidad 6 Construcción de scripts

```
DELIMITER $$
DROP FUNCTION IF EXISTS quienGano$$
CREATE FUNCTION quienGano(resultado VARCHAR(7)) RETURNS INTEGER
BEGIN
    DECLARE primerResult INTEGER;
    DECLARE segundoResult INTEGER;
    DECLARE resultado VARCHAR(1);
    SET primerResult = SUBSTRING_INDEX('000-000','- ',1);
    SET segundoResult = SUBSTRING_INDEX('000-000','- ',-1);
    IF primerResult > segundoResult
        THEN SET resultado = 0;
    ELSE
        SET resultado = 1;
    END IF;
    RETURN resultado;
END; $$
select quienGano('123-134');
```

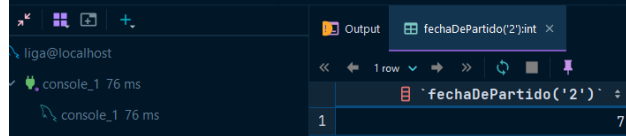
3. Crea una rutina que compruebe si los partidos ganados por un equipo coinciden con el campo pg en la tabla equipo. En caso correcto la rutina devolverá "correcto" y en el caso contrario devolverá "error".

He intentado utilizar varios cursores y diferentes métodos para resolver el ejercicio pero no lo he conseguido. He conseguido sacar total de local + visitante de un equipo, pero de todos, sin comprobar si ha ganado o no.

4. Número de partidos jugados en un mes dado. Usa el mes como parámetro

Unidad 6 Construcción de scripts

```
DELIMITER $$
DROP FUNCTION IF EXISTS fechaDePartido$$
CREATE FUNCTION fechaDePartido(month VARCHAR(10)) RETURNS INT
BEGIN
    DECLARE resultado INT(2);
    SELECT COUNT(*) INTO resultado FROM partido WHERE month(fecha) = month;
    RETURN resultado;
END; $$
select fechaDePartido( month: '2');
```



fechaDePartido('2')
7

DELIMITER \$\$

DROP FUNCTION IF EXISTS fechaDePartido\$\$

CREATE FUNCTION fechaDePartido(month VARCHAR(10)) RETURNS INT

BEGIN

DECLARE resultado INT(2);

SELECT COUNT(*) INTO resultado FROM partido WHERE month(fecha) = month;

RETURN resultado;

END; \$\$

select fechaDePartido('2');