PROCEDIMIENTOS CON EXPRESIONES REPETITIVAS

CHRISTIAN MILLÁN SORIA

1.

a. Procedimiento "numeros" que muestre los 20 primeros números enteros.

```
drop procedure if exists numeros;
delimiter //
create procedure numeros()
begin
declare i int default 1;
while i<=20 do
select i;
set i=i+1;
end while;
end//
delimiter ;
call numeros();</pre>
```

b. Realiza un procedimiento "numeros" que muestre los 20 primeros números enteros pero con la opción "repeat".

```
drop procedure if exists numeros;
delimiter //
create procedure numeros()
begin
declare i int default 1;
repeat
select i;
set i=i+1;
until i>20 end repeat;
end//
delimiter ;
call numeros();
```

```
Result 61 Result 62 Result 63 Result 64 Result 65 Result 66 Result 67 Result 68 Result 69 Result 70 Result 71 Result 72 Result 73 Result 76 Result 76 Result 77 Result 78 Result 79 Result
```

c. Explica qué está realizando este código.

```
drop procedure if exists numeros;
delimiter //
create procedure numeros()
begin
declare i integer default 1;
declare nums varchar(100) default "";
while i<=20 DO
set nums=CONCAT(nums, " ", i);
set i=i+1;
end while;
select nums as numeros;
end//
delimiter;</pre>
```

Crea un procedimiento llamado "numeros" que genera una cadena de texto que contiene los números del 1 al 20 separados por un espacio.

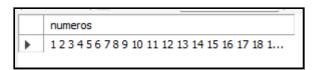
El procedimiento comienza eliminando cualquier versión anterior del procedimiento "numeros" si existe. Luego, establece el delimitador en "//" para indicar el final del procedimiento y permitir el uso de múltiples sentencias SQL.

Después, se define una variable i con un valor inicial de 1 y otra variable llamada "nums" que se utilizará para almacenar los números. A continuación, se inicia un bucle WHILE que se ejecutará hasta que i sea mayor que 20.

Dentro del bucle, se concatena el valor actual de i a la variable "nums" utilizando la función "concat" y se establece el valor de i en i+1.

Una vez que el bucle se ha completado y se han agregado todos los números a la variable "nums", se selecciona la variable "nums" y se le da un alias de "numeros".

Por último, se restablece el delimitador a su valor predeterminado ";" para indicar el final de la sentencia "sql create procedure".



a. Opción de "repeat".

```
drop procedure if exists impares;
delimiter //
create procedure impares()
begin
declare i integer; -- opción alternativa: declare i integer default 0;
set i=0;
repeat
set i=i+1;
if mod(i,2)<>0 then
select i;
end if;
until i>=20
end repeat;
end//
delimiter ;
```

```
call impares();
```



b. Opción de "while".

```
drop procedure if exists impares;
delimiter //
create procedure impares()
begin
declare i integer default 1;
declare count integer default 0;
while count<10 do
if mod(i, 2) <> 0 then
select i;
set count=count+1;
end if;
set i=i+1;
end while;
end//
delimiter;
call impares();
```



c. Usando cadena. Explica el siguiente código:

```
drop procedure if exists impares;
delimiter //
create procedure impares()
begin
declare i integer;
declare impar varchar(100) default "";
set i=1;
while i<=20 do
if mod(i,2) < 0 then
set impar=concat(impar, " ",i);
end if;
set i=i+1;
end while;
select impar as numeros_impares;
end//
delimiter;
```

```
call impares();
```

Crea un procedimiento llamado "impares" que utiliza una cadena de texto para almacenar los números impares del 1 al 20 y luego muestra esta cadena como resultado.

Primero, se elimina cualquier versión anterior del procedimiento "impares" si existe y se establece el delimitador en "//" para permitir el uso de múltiples sentencias SQL.

Luego, se declaran dos variables: i, que se inicializa en 1, y "impar", que se utiliza para almacenar los números impares.

Después, se inicia un bucle "while" que se ejecutará hasta que i sea mayor que 20.

Dentro del bucle, se verifica si i es un número impar utilizando la función MOD. Si es impar, se agrega a la cadena "impar" utilizando la función "concat" para concatenar el número al final de la cadena.

Después, se incrementa el valor de i en 1 y se continúa con la siguiente iteración del bucle "while".

Finalmente, se selecciona la variable "impar" y se le da un alias de "numeros_impares" como resultado de la ejecución del procedimiento.

