

Integrantes:

- Juan Pablo Cardona Bedoya
- Felipe Villa Jaramillo

EJERCICIOS:

1. Tabla city:

- Agregar el campo "last_user", tipo de dato texto, opcional
- Agregar el campo "last_modified", tipo de dato fecha y hora, opcional

SENTENCIA: ALTER TABLE city ADD COLUMN last_user VARCHAR(30), ADD COLUMN last_modified DATETIME;

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Name	char(35)	NO			
CountryCode	char(3)	NO	MUL		
District	char(20)	NO			
Population	int(11)	NO		0	
last_user	varchar(40)	YES		NULL	
last_modified	datetime	YES		NULL	

2. Crear la tabla "country_history" con los siguientes campos:

- id, country_id, whois, when, population_before, population_now
- Los tipos de dato resaltados en este formato deben coincidir con el código de país y la población en la tabla "country"
- El campo "id" es clave primaria y autoincremental

SENTENCIA: CREATE TABLE country_history(id INT PRIMARY KEY AUTO_INCREMENT, country_id VARCHAR(3), whois VARCHAR(30), `when` DATETIME, population_before INT(11), population_now INT(11));

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
country_id	varchar(3)	YES		NULL	
whois	varchar(30)	YES		NULL	
when	datetime	YES		NULL	
population_before	int(11)	YES		NULL	
population_now	int(11)	YES		NULL	

Escribir una función con las siguientes características:

Nombre: fn_country_size

Parámetros de entrada: área del país

Retorno: Texto con las siguientes reglas:

- "Pequeño", si el área es menor a 1'000.000 de km2
- "Mediano", si el área está entre 1'000.000 y 1'999.999 de km2
- "Grande", si el área es mayor a 2'000.000 km2

Al ejecutar una sentencia sql sobre la tabla "country" y llamar la función sobre esta sentencia, deberá indicar para cada registro el tamaño del país

SENTENCIA PARA CREAR LA FUNCION:

DELIMITER \$\$

CREATE FUNCTION fn_country_size(

 p_surface_area TYPE OF country.SurfaceArea

)

RETURNS VARCHAR(10)

DETERMINISTIC

BEGIN

 IF (p_surface_area < 1000000) THEN

 RETURN 'Pequeño';

 ELSEIF (p_surface_area BETWEEN 1000000 AND 1999999) THEN

 RETURN 'Mediano';

 ELSE

 RETURN 'Grande';

 END IF;

END;

\$\$

DELIMITER ;

SENTENCIA PARA PROBAR: SELECT Name, SurfaceArea , fn_country_size(SurfaceArea)
AS Country_Size FROM country;

Escribir un trigger con las siguientes características:

Nombre: tr_city_population_01

Lógica: Al momento de actualizar la población de una ciudad, se debe llenar automáticamente el campo "last_user" con el usuario que actualmente está conectado, y el campo "last_modified" con la fecha en la cual se realizó la modificación

Ejemplo:

update city set Population = 1000000 where Name = 'Medellín'

Al ejecutar esta sentencia, explícitamente se está actualizando únicamente la población. El trigger debe encargarse de llenar los otros dos campos.

SENTENCIA PARA CREAR TRIGGER

DELIMITER \$\$

CREATE TRIGGER tr_city_population_01

BEFORE UPDATE ON city

FOR EACH ROW

BEGIN

IF (OLD.Population != NEW.Population) THEN

SET NEW.last_user = USER();

SET NEW.last_modified = NOW();

END IF;

END

\$\$

DELIMITER ;

SENTENCIA PARA PROBAR

SELECT * FROM city WHERE Name = 'Medellín';

UPDATE city SET Population = 1000050 WHERE Name = 'Medellín';

SELECT * FROM city WHERE Name = 'Medellín';

Escribir un trigger con las siguientes características:

Nombre: tr_country_audit_01

Lógica: Al momento de actualizar la población de un país, se debe insertar en la tabla "country_history" id del país afectado, quien lo hizo, cuando se hizo, población anterior y población actual

SENTENCIA DEL TRIGGER

DELIMITER \$\$

CREATE TRIGGER tr_country_audit_01

AFTER UPDATE ON country

FOR EACH ROW

BEGIN

IF (OLD.Population != NEW.Population) THEN

INSERT INTO country_history(country_id, whois, `when`, population_before,
population_now)

VALUES (OLD.Code, USER(), NOW(), OLD.Population, NEW.Population);

END IF;

END\$\$

DELIMITER ;

SENTENCIA PARA PROBAR

SELECT * FROM country_history;

UPDATE country SET Population = 10219 WHERE Code = 'COL';

SELECT * FROM country_history;

Escribir un procedimiento almacenado con las siguientes características:

Nombre: pr_get_country_languages

Parámetros de entrada: código de país

Parámetros de salida: Lista de idiomas

Lógica: El procedimiento debe buscar que idiomas se hablan en un país y retornarlos separados por coma, indicando entre paréntesis si el idioma es oficial o no. Un ejemplo es el siguiente, para el país "COL":

Arawakan 0.1% (no oficial), Caribbean 0.1% (no oficial), Chibcha 0.4% (no oficial), Creole English 0.1% (no oficial),
Spanish 99% (oficial)

CONSULTA PROCEDIMIENTO:

DELIMITER \$\$

```
CREATE PROCEDURE pr_get_country_languages(  
    IN p_country_code TYPE OF country.Code,  
    OUT p_languages_list TEXT  
)  
BEGIN  
    SELECT  
        GROUP_CONCAT(CONCAT(Language, ' ', Percentage, '% (' , IF(IsOfficial = 'T',  
'oficial', 'no oficial'), ')') SEPARATOR ', ' )  
    INTO  
        p_languages_list  
    FROM  
        countrylanguage  
    WHERE  
        CountryCode = p_country_code;  
END $$
```

DELIMITER ;

CONSULTA PARA PROBAR EL PROCEDIMIENTO:

```
CALL pr_get_country_languages('COL', @p_languages_list);  
SELECT @p_languages_list;
```