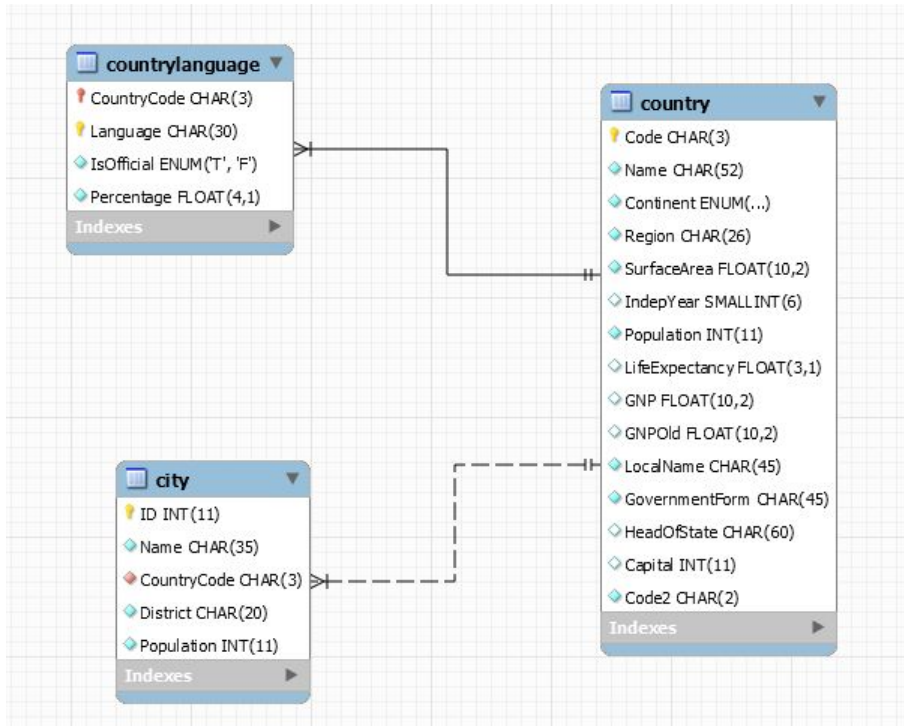


La BD WORLD es una de las BD de prueba de MySQL orientadas a la práctica con el servidor. Como véis tiene tres tablas y el esquema relacional es el siguiente:



Utilizando la BD WORLD, cuyo script tenéis en la plataforma, realizar los siguientes ejercicios:

1. Crea un procedimiento que reciba como parámetros de entrada el continente y el idioma y que **obtenga todos los países de ese continente que hablan este idioma**. Una vez realizado el procedimiento, utilízalo para obtener los países de Europa que tienen como idioma entre otros el español.- Consultar las tablas para ver qué datos introducir como parámetros.

Nota:el campo continente es de tipo enum, pero podemos pasar el nombre del continente como tipo varchar porque son compatibles. Os incluyo la cabecera del procedimiento como ayuda:

```

USE world
DELIMITER //
DROP PROCEDURE IF EXISTS EJERCICIO1_SP//
CREATE PROCEDURE EJERCICIO1_SP(IN continente char(40), IN lengua char(30))

```

Esta sería la llamada para obtener los datos que pide el ejercicio:

```

29 CALL EJERCICIO1_SP('EUROPE', 'SPANISH')
30

```

name	region	population
Andorra	Southern Europe	78000
Spain	Southern Europe	39441700
France	Western Europe	59225700
Sweden	Nordic Countries	8861400

2. Crear un procedimiento almacenado que CUENTE los países que empiezan por una determinada letra que le pasaremos por parámetro. Utilizar la tabla COUNTRY y la función CONCAT en la cláusula where del procedure (CONCAT (LETRA,'%')). Obtén el número de países que empiezan por U.

```
51 • CALL EJERCICIO2_SP('U');
```

NUMERO DE PAISES
8

3. Crea un procedimiento para añadir un nuevo registro a la tabla city haciendo el tratamiento de errores utilizando handler y mostrando los mensajes pertinentes. Ejecuta el procedimiento con los siguientes datos para comprobar el funcionamiento del procedimiento:

(1000,'PLASENCIA','ESP','CACERES',40360)

```
74 • CALL EJERCICIO3_SP(1000,'PLASENCIA','ESP','CACERES',40360);
```

Mensaje
clave de ciudad duplicada

(5000,'PLASENCIA','ESP','CACERES',40360);

```
75 • CALL EJERCICIO3_SP(5000,'PLASENCIA','ESP','CACERES',40360);
```

```
75 • SELECT * FROM CITY where id=5000;
```

ID	Name	CountryCode	District	Population
5000	PLASENCIA	ESP	CACERES	40360
NULL	NULL	NULL	NULL	NULL

4. Crear una función que calcule el volumen de una esfera cuyo radio de tipo FLOAT se pasará como parámetro. Utiliza la función para calcular el volumen de una esfera de radio 5.

VOLUMEN
523.60

Ayudas: La fórmula para calcular el volumen de una esfera es $V = \frac{4}{3} \pi R^3$.

Puedes utilizar la función **PI()**, que retorna el valor de π (pi). El número de decimales que se muestra por defecto es siete, pero MySQL usa internamente el valor de doble precisión entero y la **POW(X,Y)**, que retorna el valor de X a la potencia de Y. Las dos funciones se utilizan con SELECT. Ejemplo: select PI(), select POW(2,3);

5. Crear un procedimiento para actualizar la población de un determinado país. Se pasarán dos parámetros, la nueva población de tipo float y el nombre del país. Realiza el procedimiento primero sin hacer el tratamiento de errores y pruébalo con los siguientes valores (Angola, 1234567891234) ¿Qué ocurre y porqué? No permite porque la población introducida supera el valor permitido que es int(11)

```
DELIMITER //
CREATE PROCEDURE actualizar_poblacion( pobla float, nombre char(50) )
BEGIN
    update country set population=pobla where name=nombre;
end //
DELIMITER ;
call actualizar_poblacion (1234567891234,'Angola')
```

63 17:24:01 call actualizar_poblacion (1234567891234,'Angola')

Error Code: 1264. Out of range value for column 'Population' at row 3

Modificar el procedimiento para que haga el tratamiento de los errores, declarar un handler para capturar el error que se ha producido por introducir un valor fuera de rango. El código de este error es **1264 (columna fuera de rango)**. El control de errores hará las siguientes operaciones:

- Si se introduce un valor para la población mayor del permitido, se actualizará la población de ese país aumentándole un 10%.
- Si se introduce un país que no existe, se acabará el procedimiento con un mensaje indicando que el país no existe (para comprobar si el país existe, se hará un select a la tabla COUNTRY). Declarar un handler para capturar el error **not found select**. Devolver el mensaje 'El país no existe'.

Ejemplos de llamada al procedimiento para comprobar su funcionamiento:

```
call ejercicio4_sp (1234567891234,'Angola')
```

```
select population from country where name='Angola'
```

```
call ejercicio4_sp (1234567891234,'España')
```

TRIGGERS

Crear un trigger de modo que cada vez que se haga una operación de inserción sobre la tabla country de la BD world, automáticamente se calcule por cada continente correspondiente a ese país, el número de países y la media de la población de esos países. Estos datos se introducirán en una tabla llamada **ESTADÍSTICAS** que tendrá los siguientes campos:

ESTADÍSTICAS

(id smallint(5) primary key auto_increment ,
num_paises int,
media_poblacion int,
continente varchar(50),
fecha date);

Comprueba que el trigger funciona insertando un nuevo país:

```
insert into country values ('XXY','continent1','Asia','dddd',18,1901,11111,99,100,1000,
'hola2','dddd','ddd',1111,'A3')
```

CURSORES:

Crea un procedimiento que muestre para cada continente, los nombres y la población de los 5 países más poblados. Utiliza un cursor que recorra los distintos continentes que aparecen en la tabla country.

```
DELIMITER //
DROP PROCEDURE IF EXISTS CURSOR1_SP//
CREATE PROCEDURE CURSOR1_SP()
BEGIN
    declare FINTABLA int default 0;
    declare nombre varchar(50);
    declare c cursor for select distinct continent from country;
    declare continue handler for not found set FINTABLA=1;
    open c;
    REPEAT
        fetch c into nombre;
        if not FINTABLA then
            select name, population from country
            where continent=nombre order by population desc limit 5;
        end if;
    UNTIL FINTABLA
    END REPEAT;
    close c;
END //
```

Observa la ejecución del procedimiento:

que para cada fetch (en este caso, para cada nombre de continente), la salida se muestra en una ventana nueva.

name	population
United States	278357000
Mexico	98881000
Canada	31147000
Guatemala	11385000
Cuba	11201000

name	population
China	1277558000
India	1013662000
Indonesia	212107000
Pakistan	156483000
Banqladesh	129155000

Ahora modifica el procedimiento para que muestre, a modo de informe y en una sola ventana el resultado de todos los continentes, para ello, dentro del procedimiento debes crear una tabla para guardar los resultados. El formato de salida debe ser más o menos el siguiente:

El formato de la salida, será el siguiente:

mensaje
Países más poblados de North America
United States con una poblacion de 278357000
Mexico con una poblacion de 98881000
Canada con una poblacion de 31147000
Guatemala con una poblacion de 11385000
Cuba con una poblacion de 11201000
Países más poblados de Africa
Nigeria con una poblacion de 111506000
Egypt con una poblacion de 68470000
Ethiopia con una poblacion de 62565000
Congo, The Democratic Republic of the con una poblacion de 51654000
South Africa con una poblacion de 40377000
Países más poblados de South America
Brazil con una poblacion de 170115000

Por tanto, la tabla de resultados sólo tendrá un campo de texto a rellenar con la concatenación de los datos obtenidos de la tabla COUNTRY.

La llamada al procedimiento será:

```
284 • call CURSOR1_SP_MODIFICADO ();
285 • select * from paisemas poblados;
```

mensaje
Países más poblados de North America
United States con una poblacion de 278357000
Mexico con una poblacion de 98881000
Canada con una poblacion de 31147000
Guatemala con una poblacion de 11385000
Cuba con una poblacion de 11201000
Países más poblados de Africa
Nigeria con una poblacion de 111506000