

Instrucciones:

- Fecha de publicación: 27 de agosto de 2019.
- Fecha de entrega: 2 de septiembre de 2019 a las 10:00 a.m.
- Medio de entrega: <https://e-aulas.urosario.edu.co> (no se reciben tareas por correo electrónico u otros medios).
- La tarea **debe** realizarse en grupos de dos o máximo tres personas.
- Formato de entrega: un solo archivo comprimido (.zip, .rar, .tgz) cuyo nombre debe tener el formato: APELLIDOS_tarea1.xxx. Por cada punto debe haber un archivo cuyo nombre tenga el formato APELLIDOS_tarea1_puntoX.xxx.
- No deje espacios en los nombres de los archivos.

1. [10 ptos.] Cree una nueva base de datos **tarea1**. De **e-aulas** descargue el archivo **world.sql** y utilícelo para definir y llenar su base de datos **desde la consola de Linux** (este archivo no se puede ejecutar desde pgAdmin). Escriba los comandos usados en su archivo solución para este punto. Si definitivamente no sabe cómo hacer consulte con el profesor pues es necesario para realizar los demás puntos.
2. [10 ptos.] Cree una consulta que genere la lista de países de la región South America que se independizaron antes de 1900.
3. [5 ptos.] Liste todas las ciudades cuyo nombre empiece por “Saint” y cuente su número.
4. [5 ptos.] Liste todos los continentes con su respectiva población total en orden del continente con mayor población al continente con menor población.
5. [10 ptos.] Cree una consulta que genere la lista de continentes y regiones con su respectivo número de países. El resultado de su consulta debe estar ordenado descendientemente por el número de países en cada tupla (continente, región).
6. [10 ptos.] Cree una consulta que genere una lista de idiomas con la población total que los hablan. Tenga en cuenta que cada país tiene una población total y en cada país se pueden hablar varios idiomas. Para cada país y cada idioma se conoce el porcentaje de la población que lo habla. Para ilustrar esto ejecute la siguiente consulta

```
select * from countrylanguage where countrycode = 'BEL';
```

Note que el campo percentage está entre 0 y 100.

7. [10 ptos.] Aunque todas las ciudades corresponden a un país, hay países que no tienen ciudades asociadas. Cree una consulta que liste los países que no tienen ciudades asociadas.

8. [10 ptos.] Note que la tabla `city` no tiene definidas llaves primarias ni foráneas. ¿Cómo modificaría el comando de creación de esta relación para incorporar una llave primaria y una foránea? El comando original lo encuentra al inicio del archivo `world.sql`.
9. [10 ptos.] Se quiere mantener la población en las relaciones `city` y `country` sincronizadas. Para este fin cree un `trigger` que automáticamente aumente la población de un país en la tabla `country` en caso de que se cree una ciudad asociada a ese país en la tabla `city`. Cree la función asociada al trigger y el trigger. Pruebe su trigger verificando la población del país con código '`CHL`' antes y después de ejecutar la siguiente consulta

```
insert into city
values(4080, 'Springfield', 'CHL', 'Springfield', 10000000);
```

10. [10 ptos.] Construya una vista que contenga la densidad poblacional de cada país. La vista debe incluir el nombre del país, su densidad poblacional y su posición en el ranquin de países de acuerdo con su densidad poblacional. Es decir, el país con la mayor densidad poblacional debe aparecer al comienzo de la tabla y debe tener el ranquin 1.

Realice una consulta sobre la vista creada que muestre el top 5 de los países con mayor densidad poblacional.

11. [10 ptos.] Agregue una restricción a la tabla `city` que garantice que el atributo población siempre sea no-negativo. Verifique su restricción intentando insertar un registro con población negativa.