

## LABORATORIO 4

### Enunciado del problema

Tenemos una gran emergencia en Ucasoft: a la gerencia le encanta estar “a la moda” y sobre todo hacer \$\$\$, así que se le ocurrió ofrecer a los clientes del ERP el software pero esta vez, en lugar de ser instalado en los servidores de cada empresa (“on premise”) se va a ofrecer como servicio en la nube... se lee bonito, pero sin duda tiene consecuencias graves para quienes administran la infraestructura de Ucasoft, así que han tenido que ponerse las pilas, esperando que valga la pena el esfuerzo.

El primer punto crítico que hay que resolver es el hospedaje de las bases de datos de dicho sistema para cada cliente, las cuales se prevén que crezcan enormemente según se van migrando usuarios, por lo que va a ser necesario realizar las siguientes actividades:

### Ejercicios

1. Crear un segundo tablespace denominado SAAS para albergar las bases de datos de cada empresa.
  - a) Haga que este tablespace sea el que funcione por defecto para efectuar todas las instrucciones a partir de la presente.
  - b) Considere que el punto de montaje del almacenamiento SAN dedicado a tal fin está en /mnt/saas. Si desea simularlo en su computadora puede convertirse en root y ejecutar el siguiente comando:

```
mkdir /mnt/saas; chown postgres /mnt/saas
```
2. Cree usuarios en el SGBD para cada una de las empresas con permiso de creación de BD. Nuestros clientes actuales son: powerrepuestos, laatencao, simon y moonbucks.

3. Cada uno de los usuarios anteriores deberá crear una base de datos con su mismo nombre. No olvide convertirse en el usuario (\c - nombre en psql) antes de ejecutar los comandos.
4. Para hacer pruebas de funcionamiento se va a replicar la BD de Ucasoft (laboratorios 2 y 3). Reemplace en el script de estructura las tablas simples de la especialización de Proyecto (web, ERP y venta\_almacen) por tablas que utilicen herencia.
5. Ejecute en cada BD y con cada usuario administrador los scripts de creación de tablas e inserción de datos de la BD Ucasoft (en psql con el comando \i). Compruebe que todo funcionó como se esperaba.
6. Conceda permiso de conexión y creación de cada BD al usuario uca.
7. Conceda permiso de uso al esquema public de cada base de datos al usuario uca.
8. Conceda en cada BD todos los privilegios sobre todas las tablas al usuario uca. Esto debe ejecutarse en un solo comando GRANT.
9. Con el usuario uca cree un esquema denominado gestion en cada base de datos.
10. Cree en dicho esquema un tipo de datos compuesto denominado tipo\_sentencia de dos campos: el primero será un enumerado llamado clase con los valores select, insert, delete y update, y el segundo será un texto denominado cadena.
11. Ejecute en el esquema la sentencia de creación de una tabla para futura auditoría que tenga una columna fechahora de tipo timestamp como clave primaria, una columna sentencia del tipo compuesto tipo\_sentencia, así como un array de textos que servirá para almacenar la pantalla en la que se ejecutó la sentencia y cualquier otra información relevante. Por ser de prueba, no establezca la restricción NOT NULL a ningún campo exceptuando el campo fechahora.
12. Particione la tabla de auditoría anteriormente creada por rango de fechas, cree una partición para cada año 2019 y 2020. También agregue una tabla para los casos no incluidos en las particiones definidas (default).

13. Utilice la función `generate_series` para insertar valores en la tabla particionada. Por ejemplo, para generar un valor cada dos horas:

```
generate_series('2019-01-01 00:00'::timestamp, '2021-12-31 12:00', '2 hours')
```

Compruebe que todo funcionó correctamente. Posteriormente invítese a su bebida favorita: ¡se la merece!.

**Instrucciones:**

1. Elabore un script de texto plano con todos los comandos necesarios para efectuar los ejercicios.
2. Adjunte los scripts de estructura y datos utilizados.
3. Comprímalos en un solo archivo y súbalo al moodle.