

En este taller utilizaremos PostgreSQL, un DBMS de código abierto. Éste se encuentra instalado en Linux en la sala Ada Lovelace y tiene creado un usuario *lovelace* y una base de datos por defecto del mismo nombre. También utilizamos pgAdmin4, una aplicación web para administrar bases de datos en PostgreSQL.

## 1. Realizando consultas

1. Inicie pgAdmin4, cree una nueva base de datos taller3db. Con el Query Tool abra el archivo `taller3_base.sql` y ejecute todas las instrucciones que allí se encuentran.
2. Elimine todos los cursos de la relación `curso2` cuyo nombre termine en '4'.
3. Cree una nueva relación `curso_all` con el mismo esquema de la relación `curso` (sin datos).
  - a) Llene la nueva relación con los registros en la relación `curso`.
  - b) Agregue a la nueva relación los registros en la relación `curso2` que no aparecen en la relación `curso` (dado que ya los agregó).
4.
  - a) Aumente el presupuesto de todas las unidades en 3 %.
  - b) Aumente el presupuesto de todas las unidades en una cantidad igual al presupuesto de la entidad con el presupuesto más pequeño.
5. Cree una vista que resuma el presupuesto promedio, mínimo, máximo y su desviación estándar. Su relación resultado debe tener un solo registro con los atributos mencionados. Asigne nombres claros a los atributos.
6. Cree una vista `resumenGruposUnidad` que tenga una lista de todos los grupos. Cada registro debe incluir el nombre de la unidad que lo ofrece, el nombre del curso, el código del grupo, el año y el semestre (en ese orden). Si una unidad no tiene grupos asociados, también debe incluirse en la lista. (Utilice la relación `curso`). Verifique que la vista se haya creado en pgAdmin4.
  - a) Agregue dos registros a la relación `grupo` correspondientes al curso con código 003, grupos 1 y 2, abiertos en el primer semestre de 2018.
  - b) Verifique que la vista `resumenGruposUnidad` se haya actualizado con los nuevos registros.
7. Cree una vista materializada `instructorUbic` con los nombres y apellidos de todos los instructores, así como con el nombre del edificio en el que trabajan.
  - a) Agregue un registro a la relación `instructor` que corresponda a un nuevo instructor con nombre Camilo, apellido Ramirez, id 5, asociado a la unidad ECON, con un salario de 200.
  - b) Verifique que la vista materializada NO se haya actualizado.

- c) Actualice la vista materializada con el comando  

```
refresh materialized view instructorUbic;
```
  - d) Verifique que la vista materializada se haya actualizado.
8. Realice las siguientes actualizaciones como una transacción:
- a) Incremente el presupuesto de la unidad con menor presupuesto en una cantidad igual al presupuesto promedio de todas las unidades.
  - b) Incremente el presupuesto de la unidad con mayor presupuesto en una cantidad igual al presupuesto de la unidad con el presupuesto más pequeño.
9. Crearemos una nueva relación examen.
- a) Modifique la relación `curso` para que el nombre sea único. Utilice el comando  

```
alter table nombre_tabla  
add constraint nombre_restriccion unique (nombre_atributo);
```
  - b) Cree una nueva relación `examen` con atributos `examen_cod` (varchar(7) llave primaria), `curso_nombre` (varchar(20)), `fecha` (fecha no nula), `hora_inicio` (tiempo con valor por defecto '00:00:00'), `hora_fin` (tiempo con valor por defecto '00:00:00'). Además `curso_nombre` debe ser una llave foránea que referencie al atributo `nombre` de la relación `curso`. Esta llave debe actualizarse y borrarse en cascada. Verifique las restricciones sobre esta relación en pgAdmin4.
  - c) Agregue al menos 3 registros a la tabla `examen`. Visualice los registros de las relaciones `curso` y `examen`.
  - d) En la relación `curso` cambie el nombre del curso 'programacion' por 'prog computadores'. Visualice los cambios en las relaciones `curso` y `examen`.
  - e) Cree una vista `duracionExamenes` que tenga el nombre del curso, la hora de inicio (solamente el número del 0 al 23), y la duración.
10. Cree un dominio de nombre `credito` de tipo entero que garantice que el valor siempre sea no-negativo. Modifique la relación `curso` para cambiar el tipo de la columna `creditos` a ser de tipo `credito`.
11. Cree algunos usuarios y roles, asígneles y quíteles algunos privilegios.
- a) Cree un nuevo usuario `ramon` con clave 'ramon', y permítale leer la información de la tabla `unid_acad`.
  - b) En pgAdmin4 cree un nuevo servidor con nombre `servidor_ramon`. La dirección debe ser `localhost` y el usuario debe ser `ramon` con su clave. Conéctese a la base de datos `taller3db` e intente visualizar la relación `unid_acad`. Ahora intente insertar un registro en esta relación.
  - c) Desde su usuario, cree un nuevo rol `planeacion`. Asígnele a este rol permisos de inserción en la relación `unid_acad`. Asigne este rol al usuario `ramon`. Vuelva al servidor de ramon e intente agregar un registro a la relación `unid_acad`.

- d)* Desde su usuario, quítele el permiso de inserción al rol **planeación** sobre la relación **unid\_acad**. Vuelva al servidor de ramon e intente agregar un registro a la relación **unid\_acad**. Intente visualizar la relación.
- e)* Desde su usuario, quítele el permiso de lectura al usuario **ramon** sobre la relación **unid\_acad**. Vuelva al servidor de ramon e intente visualizar la relación.