

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. Instalando PostgreSQL y pgAdmin en Ubuntu 18.04

### 1.1. Instalando PostgreSQL

Para instalar PostgreSQL es suficiente con ejecutar los siguientes comandos en la terminal para actualizar la información sobre las actualizaciones a paquetes e instalar el core de PostgreSQL y algunos paquetes adicionales.

```
sudo apt-get update
sudo apt-get install postgresql postgresql-contrib
```

Por defecto se crea el rol postgres y el usuario postgres. Para definir la contraseña de este usuario, cambie su usuario actual a postgres en Ubuntu desde la terminal

```
sudo -i -u postgres
```

Ahora conéctese a la base de datos por defecto de este usuario (también llamada postgres)

```
psql
```

Ahora se encuentra en la interfaz de línea de comandos de PostgreSQL. Cambie el password usando el siguiente comando

```
\password
```

Una vez haya ingresado y confirmado el nuevo password del usuario postgres, desconéctese de la base de datos.

```
\q
```

Este puede ser un buen momento para crear otro usuario para usted, diferente a postgres. Desde la terminal de Ubuntu, pero aún con el usuario postgres, ejecute el comando

```
createuser --interactive
```

Defina el nombre y si el nuevo usuario debe ser superusuario

```
Enter name of role to add: juanpg
Shall the new role be a superuser? (y/n) y
```

Para definir o cambiar el password del nuevo usuario puede realizar el mismo procedimiento que con el usuario postgres decrito arriba. Todo usuario se conecta por defecto a una base de datos con el mismo nombre. Usted puede en este momento crear la base de datos asociada al usuario que acaba de crear. Para esto, desde la terminal de Ubuntu con el usuario postgres, ejecute el comando

```
createdb juanpg
```

Si el nuevo usuario de PostgreSQL ya existe en Ubuntu (con el mismo nombre), puede cambiar a ese usuario y conectarse a su base de datos por defecto, o a otra si lo prefiere. Si el usuario no existe en Ubuntu puede crearlo desde su la terminal de Ubuntu en la cuenta de superusuario con el comando

```
sudo adduser juanpg
```

## 1.2. Instalando pgAdmin4

Para instalar pgAdmin4 primero es necesario instalar curl y ca-certificates

```
sudo apt-get install curl ca-certificates
```

Utilice ahora curl con el comando

```
curl https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key  
add -
```

Agregue ahora el repositorio de pgAdmin4 con el comando

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/$(  
lsb_release -cs)-pgdg-main" > /etc/apt/sources.list.d/pgdg.list'
```

Actualice el listado de paquetes e instale la versión 11 de PostgreSQL y pgAdmin4

```
sudo apt-get update  
sudo apt-get install postgresql-11 pgadmin4
```

Ahora puede iniciar pgAdmin4 con el comando

```
pgadmin4
```

## 2. Instalando PostgreSQL y pgAdmin en Windows 10

Para instalar PostgreSQL en Windows 10 vaya a la página de descargas <https://www.postgresql.org/download/> Allí seleccione el enlace de Windows, que lo llevará a la página <https://www.postgresql.org/download/windows/> Dando click en 'Download the Installer' lo llevará a una página de EDB <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads> Una vez allí un click en la versión 11 de PostgreSQL para Windows 64 inicia la descarga del instalador.

Una vez descargado, ejecute el instalador. En un cuadro de diálogo se le preguntará qué componentes quiere instalar. Mantenga señaladas las cuatro componentes: el servidor (server), pgAdmin4, Stack Builder, y la línea de comandos.

Durante la instalación se le pedirá que defina la contraseña del usuario postgres. Introduzca la contraseña y asegúrese de recordarla pues le será necesaria para conectarse, crear nuevos usuarios, etc.

Al terminar la instalación de PostgreSQL, se lanzará Stack Builder, que le preguntará si quiere instalar paquetes adicionales. En este momento no es necesario, así que puede simplemente cancelar la operación.

Ahora puede iniciar pgAdmin4 buscándola como una aplicación de Windows: presione la tecla Windows y escriba pgAdmin4. En caso de que no aparezca, puede encontrarla en un directorio similar a

```
C:\Program Files\PostgreSQL\11\pgAdmin 4\bin
```

### 3. Realizando consultas

1. Ahora 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.