

- a. Instale el motor de base de datos PostgreSQL sobre una máquina virtual con sistema operativo Linux Slackware en Virtual Box

- Para iniciar creamos un usuario llamado “postgres” que será nuestro administrador de la base de datos

```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@Host:~# adduser postgres
```

- Ahora creamos un directorio en el cual vamos a guardar las fuentes que descargaremos del postgresql, descomprimiremos y compilaremos, después haremos dueño de este directorio a nuestro usuario postgres

```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@Host:~# mkdir /usr/src/pgsql
root@Host:~# chown postgres:users /usr/src/pgsql
root@Host:~#
```

- Ahora crearemos un directorio en el cual instalaremos nuestro postgresql y al igual que en el paso anterior haremos dueño de este directorio a nuestro usuario postgres.

```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@Host:~# mkdir /usr/local/pgsql
root@Host:~# chown postgres:users /usr/local/pgsql
root@Host:~#
```

- Saldremos de nuestro usuario root y accederemos a nuestro usuario postgres

```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@Host:~# exit
logout

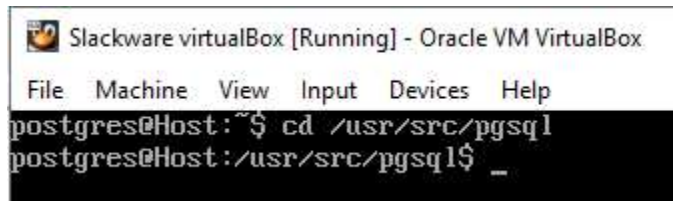
Welcome to Linux 4.4.14 (tty1)

Host login: postgres
Password:
Linux 4.4.14.
No mail.

Do you have lysdexia?

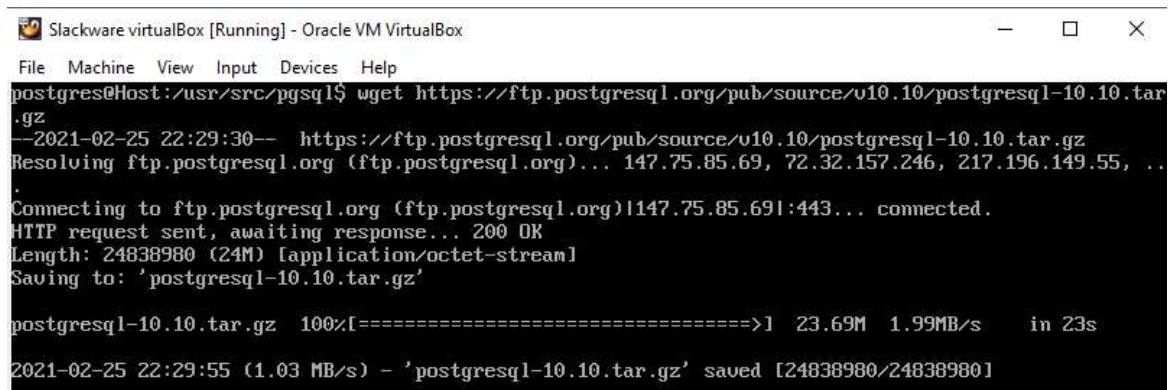
postgres@Host:~$
```

- Estando dentro de nuestro usuario postgres nos dirigimos a la carpeta que creamos con el fin de encargarse de la descarga



```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:~$ cd /usr/src/pgsql
postgres@Host:/usr/src/pgsql$ _
```

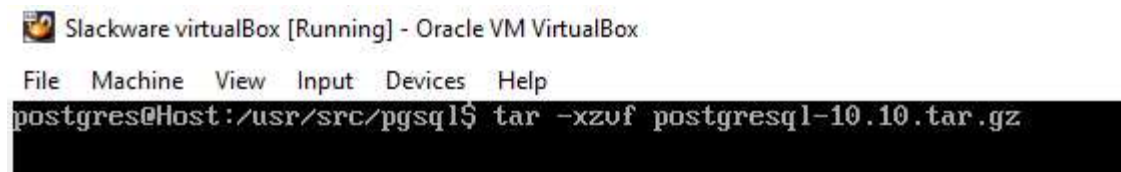
- Al entrar obtendremos las fuentes del postgresql v 10.10 (recomendada para nuestra versión de slackware) de la dirección <https://ftp.postgresql.org/pub/source/v10.10/postgresql-10.10.tar.gz>



```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:/usr/src/pgsql$ wget https://ftp.postgresql.org/pub/source/v10.10/postgresql-10.10.tar.gz
--2021-02-25 22:29:30-- https://ftp.postgresql.org/pub/source/v10.10/postgresql-10.10.tar.gz
Resolving ftp.postgresql.org (ftp.postgresql.org)... 147.75.85.69, 72.32.157.246, 217.196.149.55, ..
Connecting to ftp.postgresql.org (ftp.postgresql.org)|147.75.85.69|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24838980 (24M) [application/octet-stream]
Saving to: 'postgresql-10.10.tar.gz'

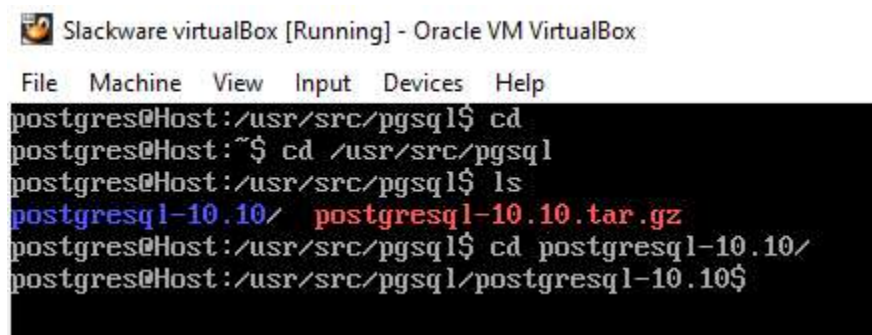
postgresql-10.10.tar.gz 100%[=====>] 23.69M 1.99MB/s in 23s
2021-02-25 22:29:55 (1.03 MB/s) - 'postgresql-10.10.tar.gz' saved [24838980/24838980]
```

- Ahora descomprimos nuestro archivo .tar



```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:/usr/src/pgsql$ tar -xzf postgresql-10.10.tar.gz
```

- Y después accedemos al directorio "postgresql-10.10/" que se ha creado



```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:/usr/src/pgsql$ cd
postgres@Host:~$ cd /usr/src/pgsql
postgres@Host:/usr/src/pgsql$ ls
postgresql-10.10/ postgresql-10.10.tar.gz
postgres@Host:/usr/src/pgsql$ cd postgresql-10.10/
postgres@Host:/usr/src/pgsql/postgresql-10.10$
```

- Ahora corremos el script para realiza la configuración del postgresql, lo que colocamos después del “prefix=” será la ruta donde queremos realizar nuestra instalación, en nuestro caso el segundo directorio que creamos

```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:/usr/src/pgsql/postgresql-10.10$ ./configure --prefix=/usr/local/pgsql --with-perl --w
ith-tcl --with-python --with-openssl_
```

- Ahora utilizamos el comando “make” para compilar y construir nuestro postgresql, sabremos que el proceso salió bien si al final nos sale el mensaje “All of PostgreSQL successfully made. Ready to install.”

```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:/usr/src/pgsql/postgresql-10.10$ make_
```

- Y ahora realizamos la instalación digitando el comando “make install”, al final sabremos que la instalación fue exitosa si sale el mensaje “PostgreSQL installation complete.”

```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:/usr/src/pgsql/postgresql-10.10$ make install_
```

- Ahora iniciamos el cluster de las bases de datos para poder comenzar a utilizarlas, al final sabremos que todo va bien si nos sale el mensaje “./postgres -D /usr/local/pgsql/data” o el mensaje “./pg\_ctl -D /usr/local/pgsql/data -l logfile start”

```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:/usr/src/pgsql/postgresql-10.10$ /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
-E LATIN1 --locale=es_CO
```

- Y ahora con el siguiente comando inicializamos el servidor del postgresql y quedara listo para usarse



```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:/usr/src/pgsql/postgresql-10.10$ /usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l error_postgres.log start_
```

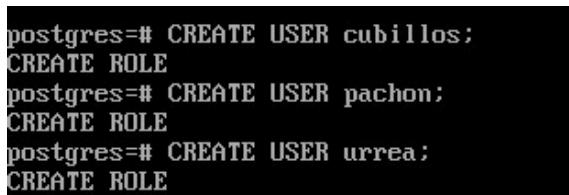
- b. Cree un usuario por cada estudiante del grupo. Use los apellidos de los estudiantes como nombre de usuario

- Para manipular el postgresql directamente , es decir crear usuarios, bases de datos u otras operaciones similares accedemos desde nuestro usuario postgresql (al cual podemos acceder mas fácilmente con el comando sudo -i -u user) a la terminal “dirección de la carpeta en que lo instalamos/psql”



```
Slackware virtualBox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@Host:~$ /usr/local/pgsql/bin/psql
```

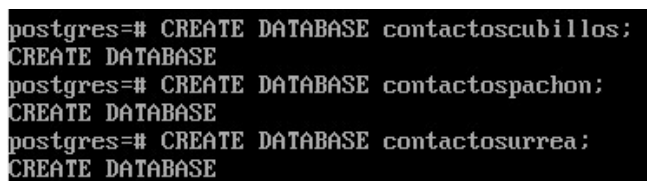
- Ahora creamos nuestros 3 usuarios



```
postgres=# CREATE USER cubillos;
CREATE ROLE
postgres=# CREATE USER pachon;
CREATE ROLE
postgres=# CREATE USER urrea;
CREATE ROLE
```

- c. Cree una base de datos para manejar la información de contactos de cada estudiante. En dicha base de datos se debe indicar nombre, teléfono, ciudad y país de residencia, fecha de cumpleaños, parentesco (Ej: familiar, compañero de estudio, compañero de trabajo, vecino, etc) y gustos comunes (Ej: Montar bici, jugar fútbol, ir al cine, escuchar música, jugar videojuegos, ir al gimnasio, etc). La base de datos debe tener al menos 3 tablas. Cada estudiante debe tener acceso sólo a la base de datos de cada uno.

- Ahora creamos nuestras bases de datos desde la misma terminal donde creamos los usuarios

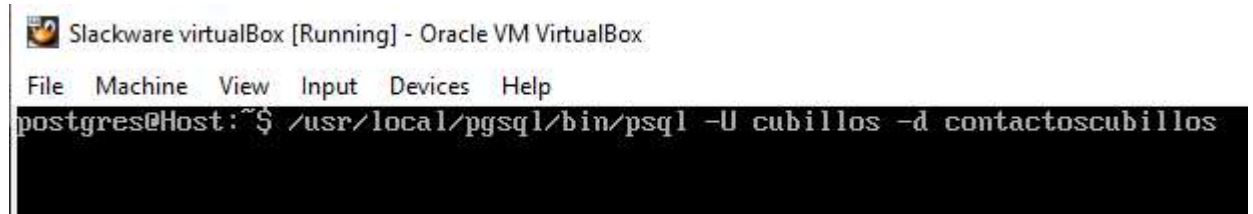


```
postgres=# CREATE DATABASE contactoscubillos;
CREATE DATABASE
postgres=# CREATE DATABASE contactospachon;
CREATE DATABASE
postgres=# CREATE DATABASE contactosurrea;
CREATE DATABASE
```

- Después revocamos los permisos públicos de nuestras bases de datos para que nadie pueda acceder y le brindamos todos los permisos a cada usuario específico a quien pertenece su base de datos

```
postgres=# REVOKE ALL ON DATABASE contactoscubillos FROM PUBLIC;
REVOKE
postgres=# GRANT ALL PRIVILEGES ON DATABASE contactoscubillos TO cubillos;
GRANT
postgres=# REVOKE ALL ON DATABASE contactospachon FROM PUBLIC;
REVOKE
postgres=# GRANT ALL PRIVILEGES ON DATABASE contactospachon TO pachon;
GRANT
postgres=# REVOKE ALL ON DATABASE contactosurrea FROM PUBLIC;
REVOKE
postgres=# GRANT ALL PRIVILEGES ON DATABASE contactosurrea TO urrea;
GRANT
```

- Y desde nuestro usuario postgres accedemos a alguno de los usuarios que creamos, el -U indica el usuario con el que nos conectaremos y el -d la base de datos a la que nos conectaremos



Slackware virtualBox [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
postgres@Host:~$ /usr/local/pgsql/bin/psql -U cubillos -d contactoscubillos
```

- Ahora creamos las tablas de nuestra base de datos, este mismo proceso se realizó para las 3 bases de datos

```
contactoscubillos=> CREATE TABLE Contactos(telefono NUMERIC(15) NOT NULL,PrimerNombre VARCHAR(15) NOT NULL, SegundoNombre VARCHAR(15), PrimerApellido VARCHAR(15) NOT NULL, SegundoApellido VARCHAR(15), fechaDeNacimiento DATE NOT NULL, codParentesco NUMERIC(5) NOT NULL, codCiudad NUMERIC(5) NOT NULL);
CREATE TABLE
contactoscubillos=> CREATE TABLE Parentescos(codigo NUMERIC(5) NOT NULL, parentesco VARCHAR(20) NOT NULL);
CREATE TABLE
contactoscubillos=> CREATE TABLE Paises(codPais NUMERIC(10) NOT NULL, nombrePais VARCHAR(30) NOT NULL);
CREATE TABLE
contactoscubillos=> CREATE TABLE Ciudades(codCiudad NUMERIC(10), codPais NUMERIC(10), nombreCiudad NUMERIC(30));
CREATE TABLE
contactoscubillos=> CREATE TABLE GustosPropios(codGustoP NUMERIC(5) NOT NULL, gusto VARCHAR(30) NOT NULL);
CREATE TABLE
contactoscubillos=> CREATE TABLE GustosAjenos(telefono NUMERIC(15) NOT NULL, gusto VARCHAR(30) NOT NULL);
CREATE TABLE
contactoscubillos=> CREATE TABLE GustosComunes(codGustoP NUMERIC(5) NOT NULL, telefono NUMERIC(15) NOT NULL, gusto VARCHAR(30) NOT NULL);
CREATE TABLE
contactoscubillos=> _
```

- Ahora le asignamos las llaves primarias a nuestras tablas

```
contactoscubillos=> ALTER TABLE Contactos ADD CONSTRAINT Contactos_PK PRIMARY KEY(telefono);
ALTER TABLE
contactoscubillos=> ALTER TABLE Parentescos ADD CONSTRAINT Parentescos_PK PRIMARY KEY(codigo);
ALTER TABLE
contactoscubillos=> ALTER TABLE Paises ADD CONSTRAINT Paises_PK PRIMARY KEY(codPais);
ALTER TABLE
contactoscubillos=> ALTER TABLE Ciudades ADD CONSTRAINT Ciudades_PK PRIMARY KEY(codCiudad);
ALTER TABLE
contactoscubillos=> ALTER TABLE GustosPropios ADD CONSTRAINT GustosPropios_PK PRIMARY KEY(codGustoP);
ALTER TABLE
contactoscubillos=> ALTER TABLE GustosAjenos ADD CONSTRAINT GustosAjenos_PK PRIMARY KEY(telefono,gusto);
ALTER TABLE
contactoscubillos=> ALTER TABLE GustosComunes ADD CONSTRAINT GustosComunes_PK PRIMARY KEY(codGustoP,telefono,gusto);
ALTER TABLE
contactoscubillos=> _
```

- Y después asignamos las llaves foráneas

```
contactoscubillos=> ALTER TABLE Contactos ADD CONSTRAINT ContactosParentesco_FK FOREIGN KEY(codParentesco) REFERENCES PARENTESCOS(codigo);
ERROR: syntax error at or near "FOREING"
LINE 1: ...E Contactos ADD CONSTRAINT ContactosParentesco_FK FOREIGN KE...
^

contactoscubillos=> ALTER TABLE Contactos ADD CONSTRAINT ContactosParentesco_FK FOREIGN KEY(codParentesco) REFERENCES PARENTESCOS(codigo);
ALTER TABLE
contactoscubillos=> ALTER TABLE Contactos ADD CONSTRAINT ContactosUbicacion_FK FOREIGN KEY(codCiudad) REFERENCES Ciudades(codCiudad);
ALTER TABLE
contactoscubillos=> ALTER TABLE Ciudades ADD CONSTRAINT CiudadPais_FK FOREIGN KEY(codPais) REFERENCES Paises(codPais);
ALTER TABLE
```

#### d. Inserte datos en las bases de datos

- Finalmente realizamos el poblado de nuestras bases de datos

```
ALTER TABLE
contactoscubillos=> INSERT INTO Ciudades VALUES(1,1,'bogota');
INSERT 0 1
contactoscubillos=> INSERT INTO Ciudades VALUES(1,1,'ubaate');
ERROR: duplicate key value violates unique constraint "ciudades_pk"
DETAIL: Key (codciudad)=(1) already exists.
contactoscubillos=> INSERT INTO Ciudades VALUES(2,1,'ubate');
INSERT 0 1
contactoscubillos=> INSERT INTO Ciudades VALUES(3,2,'londres');
INSERT 0 1
contactoscubillos=> INSERT INTO Contactos VALUES(1,'Richard','Santiaago','Urrea','Garcia',TO_DATE('31/01/2001','DD/MM/YYYY'),3,1);
ERROR: syntax error at or near ")"
LINE 1: ...aago','Urrea','Garcia',TO_DATE('31/01/2001','DD/MM/YYYY'),);
^

contactoscubillos=> SELECT * FROM Parentesco;
ERROR: relation "parentesco" does not exist
LINE 1: SELECT * FROM Parentesco;
^

contactoscubillos=> SELECT * FROM Parentescos;
codigo | parentesco
-----+-----
1 | papa
2 | mama
3 | conocido
(3 rows)

contactoscubillos=> INSERT INTO Contactos VALUES(1,'Richard','Santiaago','Urrea','Garcia',TO_DATE('31/01/2001','DD/MM/YYYY'),3,1);
INSERT 0 1
contactoscubillos=> INSERT INTO Contactos VALUES(1,'Laura','Daniela','Pachon','Cuan',TO_DATE('15/03/2000','DD/MM/YYYY'),3,1);
ERROR: duplicate key value violates unique constraint "contactos_pk"
DETAIL: Key (telefono)=(1) already exists.
contactoscubillos=> INSERT INTO Contactos VALUES(2,'Laura','Daniela','Pachon','Cuan',TO_DATE('15/03/2000','DD/MM/YYYY'),3,1);
```