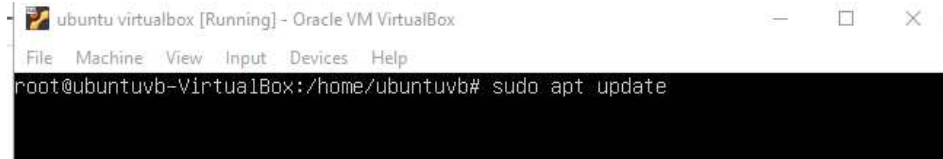


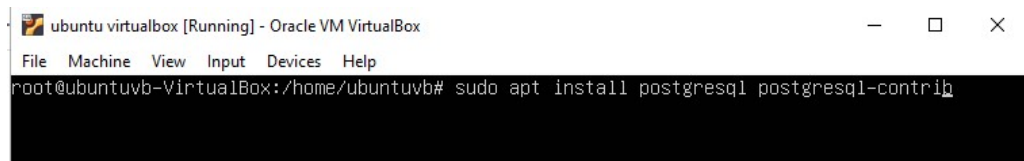
- a. Instale el motor de base de datos PostgreSQL sobre una máquina virtual con sistema operativo Linux Ubuntu en Virtual Box.

- Realizamos el update a nuestro Ubuntu



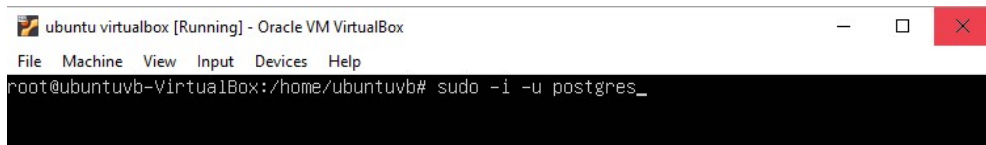
```
ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@ubuntuvb-VirtualBox:/home/ubuntuvb# sudo apt update
```

- Ahora con la función Sudo apt installamos el postgre



```
ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@ubuntuvb-VirtualBox:/home/ubuntuvb# sudo apt install postgresql postgresql-contrib
```

- Esta instalación viene por defecto con el usuario postgres, así que nos dirigiremos a el



```
ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@ubuntuvb-VirtualBox:/home/ubuntuvb# sudo -i -u postgres_
```

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

- Ahora creamos un usuario, el -P es para dejarlo sin contraseña, el -S para que no sea super usuario, el -D para que no pueda crear ni borrar bases de datos, el -R para que no pueda crear ni borrar roles(usuarios) y el -e para el nombre

```
ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@ubuntuvb-VirtualBox:~$ createuser -P -S -D -R -e cubillos
Enter password for new role:
Enter it again:
SELECT pg_catalog.set_config('search_path', '', false);
CREATE ROLE cubillos PASSWORD 'md509551d217a8a60cb57981a978b636bd7' NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN;

ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@ubuntuvb-VirtualBox:~$ createuser -P -S -D -R -e pachon
Enter password for new role:
Enter it again:
SELECT pg_catalog.set_config('search_path', '', false);
CREATE ROLE pachon PASSWORD 'md5485c8a930e6c6b75b6db636408ca4005' NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN;
postgres@ubuntuvb-VirtualBox:~$ _

ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
postgres@ubuntuvb-VirtualBox:~$ createuser -P -S -D -R -e urrea
Enter password for new role:
Enter it again:
SELECT pg_catalog.set_config('search_path', '', false);
CREATE ROLE urrea PASSWORD 'md5d23adc8d4127f4ff6ac4fe3520398da1' NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN;
postgres@ubuntuvb-VirtualBox:~$
```


- Por recomendación de instalación debemos crear un usuario real en nuestra maquina con el mismo nombre del usuario del usuario creado en postgresql, asi que volvemos a nuestro usuario root y hacemos la creación de nuestros usuarios

```
ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@ubuntuvb-VirtualBox:/home/ubuntuvb# sudo adduser cubillos
Adding user `cubillos'
Adding new group `cubillos' with GID 1001
Adding new user `cubillos' with UID 1001 and shell /bin/bash

ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@ubuntuvb-VirtualBox:/home/ubuntuvb# sudo adduser pachon
Adding user `pachon'
Adding new group `pachon' with GID 1002
Adding new user `pachon' with UID 1002 and shell /bin/bash

ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@ubuntuvb-VirtualBox:/home/ubuntuvb# sudo adduser urrea
Adding user `urrea'
Adding new group `urrea' with GID 1003
Adding new user `urrea' with UID 1003 and shell /bin/bash
```


- 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 nuevamente desde el usuario postgres

 ubuntu virtualbox [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
postgres@ubuntuvb-VirtualBox:~$ createdb destinoscubillos
postgres@ubuntuvb-VirtualBox:~$ createdb destinospachon
postgres@ubuntuvb-VirtualBox:~$ createdb destinosurrea
postgres@ubuntuvb-VirtualBox:~$
```

- Ahora revocamos los permisos public de cada una de nuestras bases de datos para que nadie se pueda conectar, después le asignamos los permisos a cada usuario específico al que le pertenezca esa base de datos

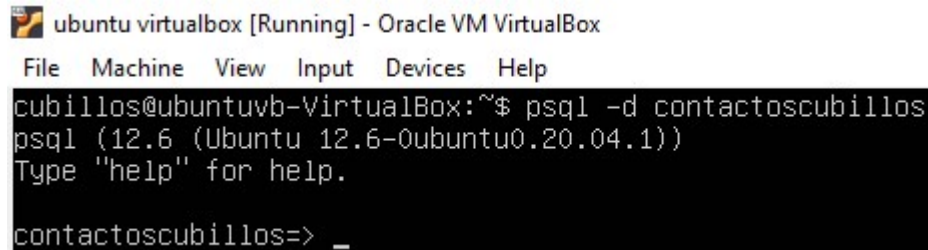
 ubuntu virtualbox [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
root@ubuntuvb-VirtualBox:/home/ubuntuvb# sudo -i -u postgres
postgres@ubuntuvb-VirtualBox:~$ psql
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

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
postgres=# _
```

- Ahora accedemos con alguno de los usuarios que creamos



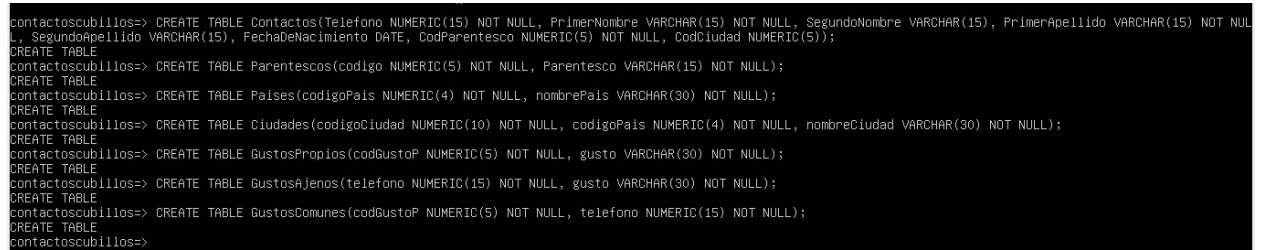
```

ubuntu virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
cubillos@ubuntuvb-VirtualBox:~$ psql -d contactoscubillos
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

contactoscubillos=> _

```

- Después creamos cada una de las tablas de nuestras bases de datos, en las 3 bases de datos se realiza el mismo proceso

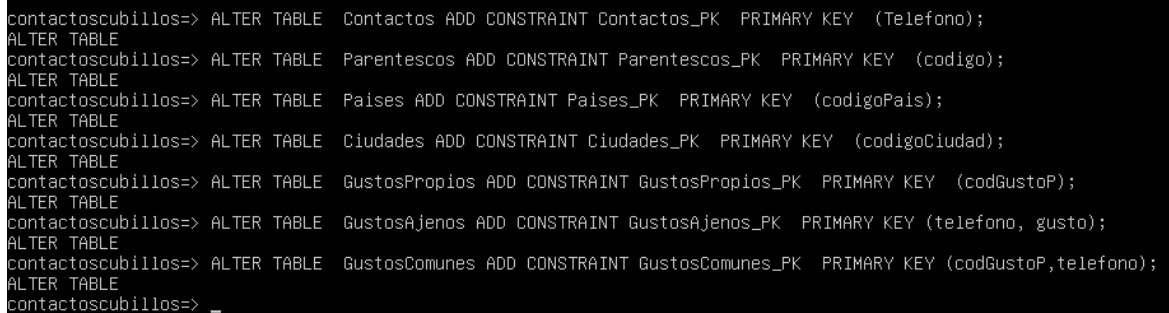


```

contactoscubillos=> CREATE TABLE Contactos(Telefono NUMERIC(15) NOT NULL, PrimerNombre VARCHAR(15) NOT NULL, SegundoNombre VARCHAR(15) NOT NULL, PrimerApellido VARCHAR(15) NOT NULL, SegundoApellido VARCHAR(15), FechaDeNacimiento DATE, CodParentesco NUMERIC(5) NOT NULL, CodCiudad NUMERIC(5));
CREATE TABLE
contactoscubillos=> CREATE TABLE Parentescos(codigo NUMERIC(5) NOT NULL, Parentesco VARCHAR(15) NOT NULL);
CREATE TABLE
contactoscubillos=> CREATE TABLE Paises(codigoPais NUMERIC(4) NOT NULL, nombrePais VARCHAR(30) NOT NULL);
CREATE TABLE
contactoscubillos=> CREATE TABLE Ciudades(codigoCiudad NUMERIC(10) NOT NULL, codigoPais NUMERIC(4) NOT NULL, nombreCiudad VARCHAR(30) NOT NULL);
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);
CREATE TABLE
contactoscubillos=>

```

- Luego asignamos nuestras llaves primarias a cada tabla

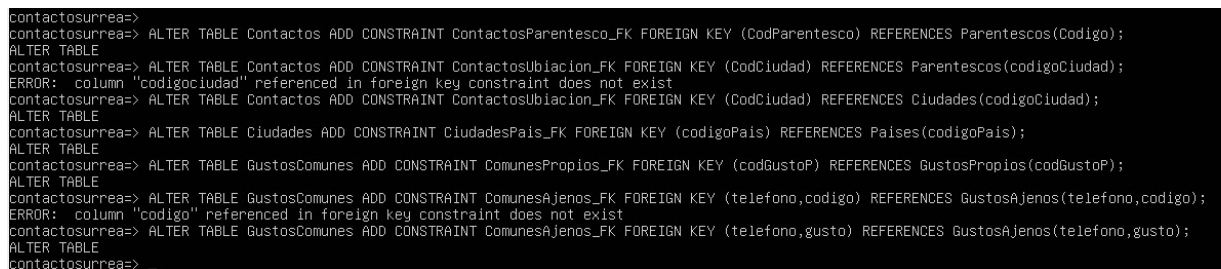


```

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 (codigoPais);
ALTER TABLE
contactoscubillos=> ALTER TABLE Ciudades ADD CONSTRAINT Ciudades_PK PRIMARY KEY (codigoCiudad);
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);
ALTER TABLE
contactoscubillos=> _

```

- Y finalmente asignamos las llaves foráneas correspondientes a nuestras tablas



```

contactosurrea=>
contactosurrea=> ALTER TABLE Contactos ADD CONSTRAINT ContactosParentesco_FK FOREIGN KEY (CodParentesco) REFERENCES Parentescos(Codigo);
ALTER TABLE
contactosurrea=> ALTER TABLE Contactos ADD CONSTRAINT ContactosUbiacion_FK FOREIGN KEY (CodCiudad) REFERENCES Parentescos(codigoCiudad);
ERROR: column "codigoCiudad" referenced in foreign key constraint does not exist
contactosurrea=> ALTER TABLE Contactos ADD CONSTRAINT ContactosUbiacion_FK FOREIGN KEY (CodCiudad) REFERENCES Ciudades(codigoCiudad);
ALTER TABLE
contactosurrea=> ALTER TABLE Ciudades ADD CONSTRAINT CiudadesPais_FK FOREIGN KEY (codigoPais) REFERENCES Paises(codigoPais);
ALTER TABLE
contactosurrea=> ALTER TABLE GustosComunes ADD CONSTRAINT ComunesPropios_FK FOREIGN KEY (codGustoP) REFERENCES GustosPropios(codGustoP);
ALTER TABLE
contactosurrea=> ALTER TABLE GustosComunes ADD CONSTRAINT ComunesAjenos_FK FOREIGN KEY (telefono, codigo) REFERENCES GustosAjenos(telefono, codigo);
ERROR: column "codigo" referenced in foreign key constraint does not exist
contactosurrea=> ALTER TABLE GustosComunes ADD CONSTRAINT ComunesAjenos_FK FOREIGN KEY (telefono, gusto) REFERENCES GustosAjenos(telefono, gusto);
ALTER TABLE
contactosurrea=> _

```

d. Inserte datos en las bases de datos

- Poblamos la base de datos

```
contactoscubillos=> INSERT INTO GustosPropios VALUES (1,'Escuchaar musica');
INSERT 0 1
contactoscubillos=> INSERT INTO GustosPropios VALUES (1,'Ver peliculas');
ERROR:  duplicate key value violates unique constraint "gustospropios_pk"
DETAIL:  Key (codgustop)=(1) already exists.
contactoscubillos=> INSERT INTO GustosPropios VALUES (2,'Ver peliculas');
INSERT 0 1
contactoscubillos=> INSERT INTO GustosAjenos VALUES (3123123,'Ver peliculas');
INSERT 0 1
contactoscubillos=> INSERT INTO GustosAjenos VALUES (3123123,'Ir al gimnasio');
INSERT 0 1
contactoscubillos=> INSERT INTO GustosAjenos VALUES (3213213,'Escuchar musica');
INSERT 0 1
contactoscubillos=> INSERT INTO GustosAjenos VALUES (3213213,'Escuchar musica');
ERROR:  duplicate key value violates unique constraint "gustosajenos_pk"
DETAIL:  Key (telefono, gusto)=(3213213, Escuchar musica) already exists.
contactoscubillos=> INSERT INTO Paises VALUES (57, 'Colombia');
INSERT 0 1
contactoscubillos=> INSERT INTO Ciudades VALUES (1, 'Bogota');
ERROR:  invalid input syntax for type numeric: "Bogota"
LINE 1: INSERT INTO Ciudades VALUES (1, 'Bogota');
                                     ^
contactoscubillos=> INSERT INTO Ciudades VALUES (1,57, 'Bogota');
INSERT 0 1
contactoscubillos=> INSERT INTO Ciudades VALUES (2,57, 'Ubate');
INSERT 0 1
contactoscubillos=> INSERT INTO Parentescos VALUES (1,'papá');
ERROR:  syntax error at or near "INSERT"
LINE 1: INSERT INTO Parentescos VALUES (1,'papá');
                                     ^
contactoscubillos=> INSERT INTO Parentescos VALUES (1,'papá');
INSERT 0 1
contactoscubillos=> INSERT INTO Parentescos VALUES (2,'mamá');
INSERT 0 1
contactoscubillos=> INSERT INTO Parentescos VALUES (3,'Hermano');
INSERT 0 1
contactoscubillos=> INSERT INTO Parentescos VALUES (4,'Conocido');
INSERT 0 1
contactoscubillos=> INSERT INTO Contactos(3123123,'Laura','Daniela','Pachon','Cuan',TO_DATE('15/03/2000','DD/MM/YYYY'),4,2)
contactoscubillos=>
```