Cómo trabajar con PostgreSQL

Para conectarse al servidor

Es necesario logearse a anakena.dcc.uchile.cl por SSH a través de una ventana terminal. En la ventana, ejecutar:

```
psql -U cc42aXX -d cc42aXXdb
```

En el caso de la conexión a dichato (postgrado), ejecutar:

```
psql -U cc55aXX -d cc55aXXdb
```

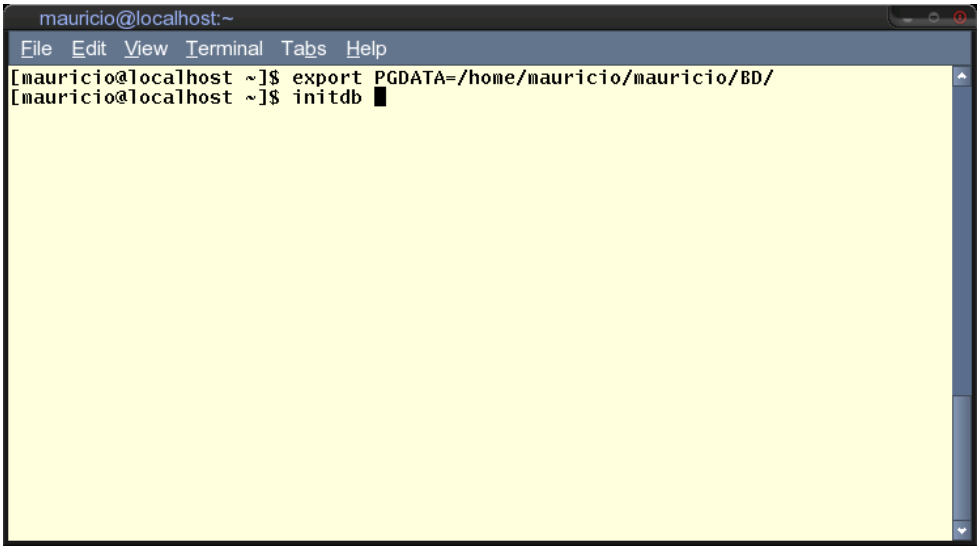
Para avanzar en una base de datos PostgreSQL personal

Si se trabaja en la casa, o el demonio Postmaster no está en ejecución, seguir los siguientes pasos:

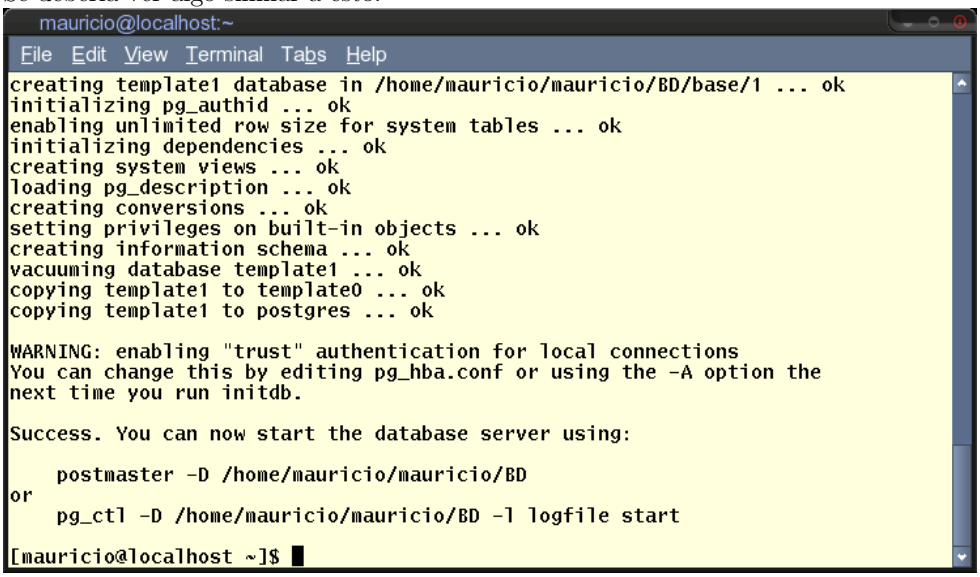
Construir la base de datos:

Setear la variable de ambiente **PGDATA** hacia la dirección donde la BD será creada.

Crear la BD con el comando **initdb**.

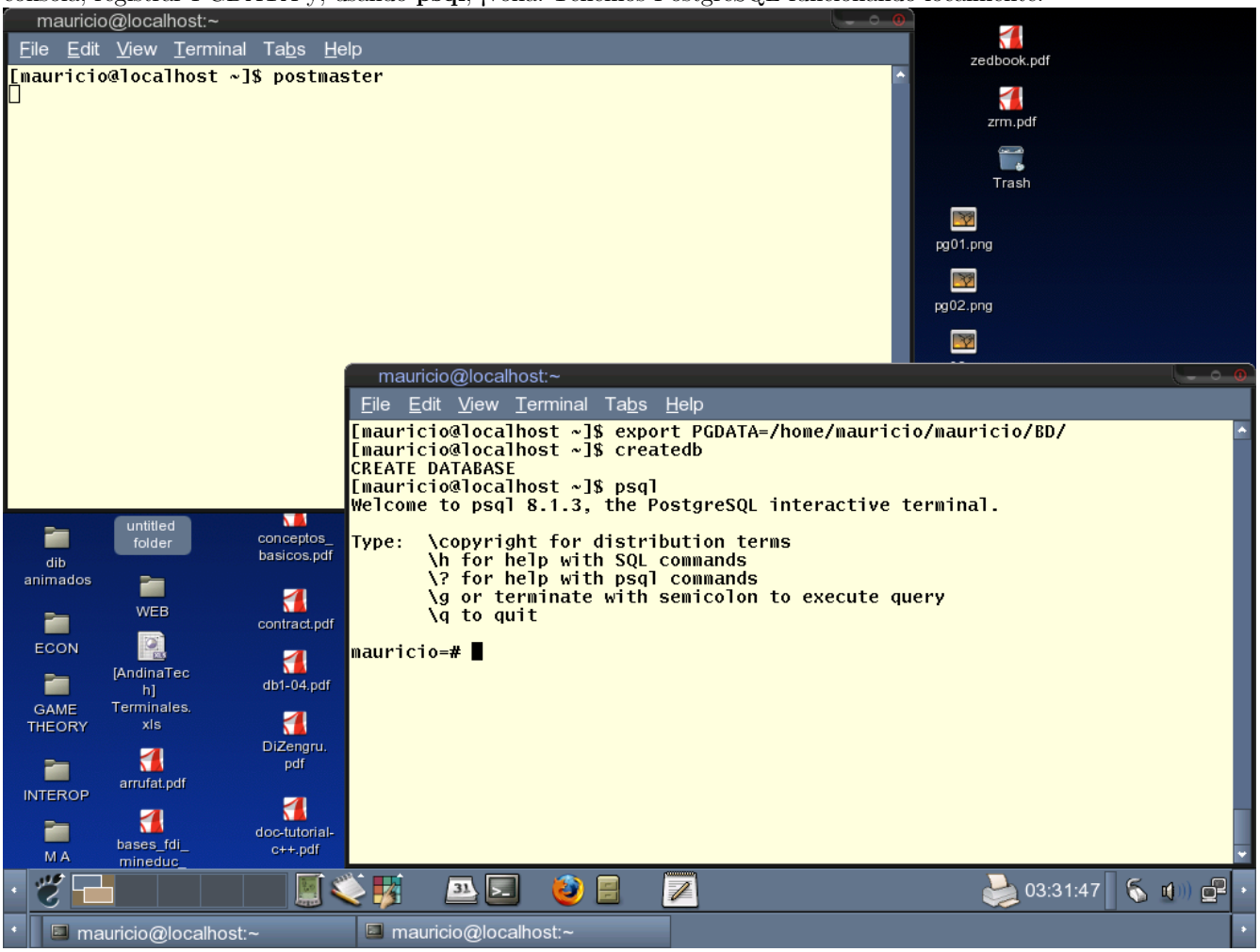


Se debería ver algo similar a esto:



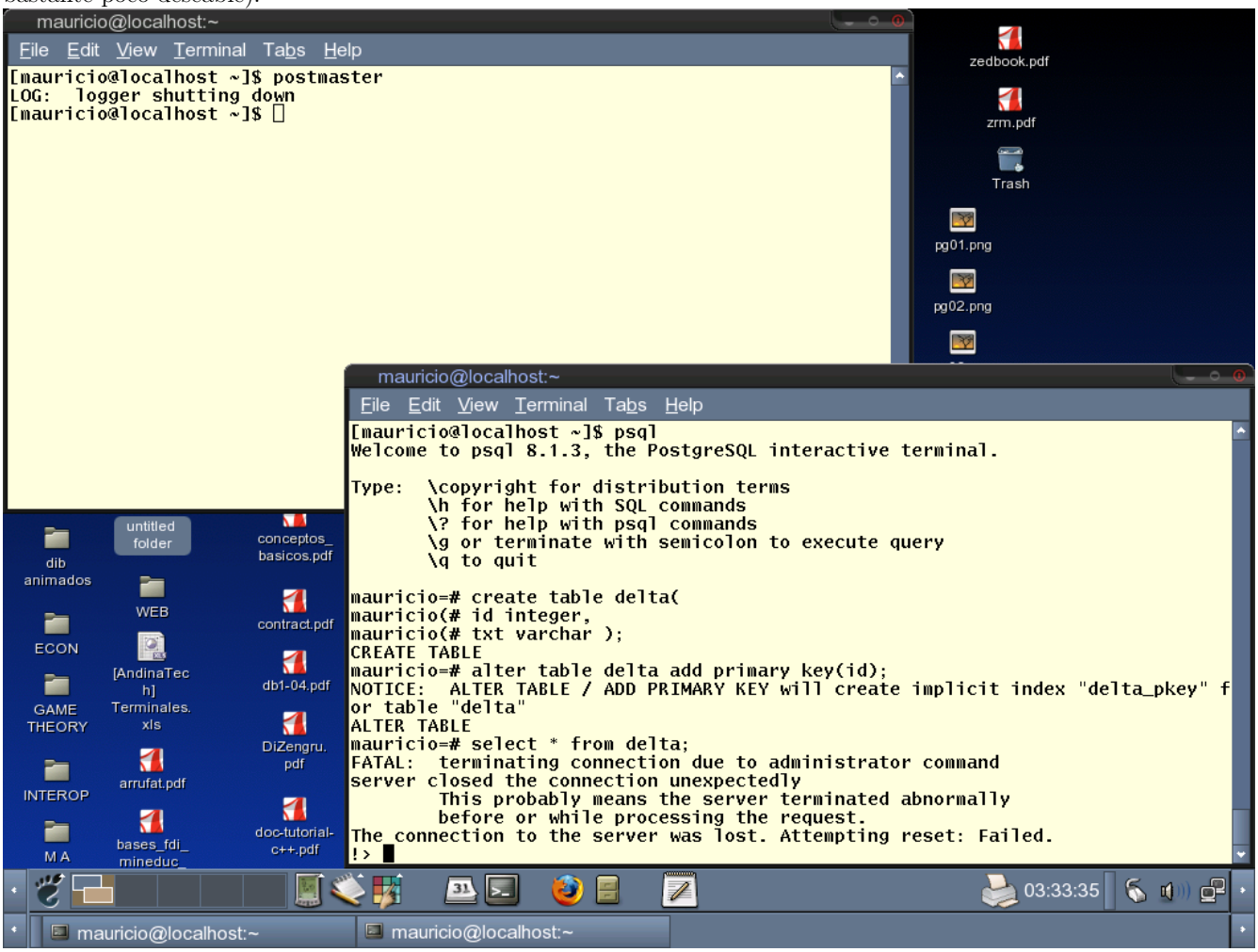
Luego iniciamos la base de datos PostgreSQL:

Ahora está en ejecución una BD. Localmente podemos acceder desde otro terminal. En este caso se puede abrir otra consola, registrar PGDATA y, usando **psql**, ¡voilá! Tenemos PostgreSQL funcionando localmente.

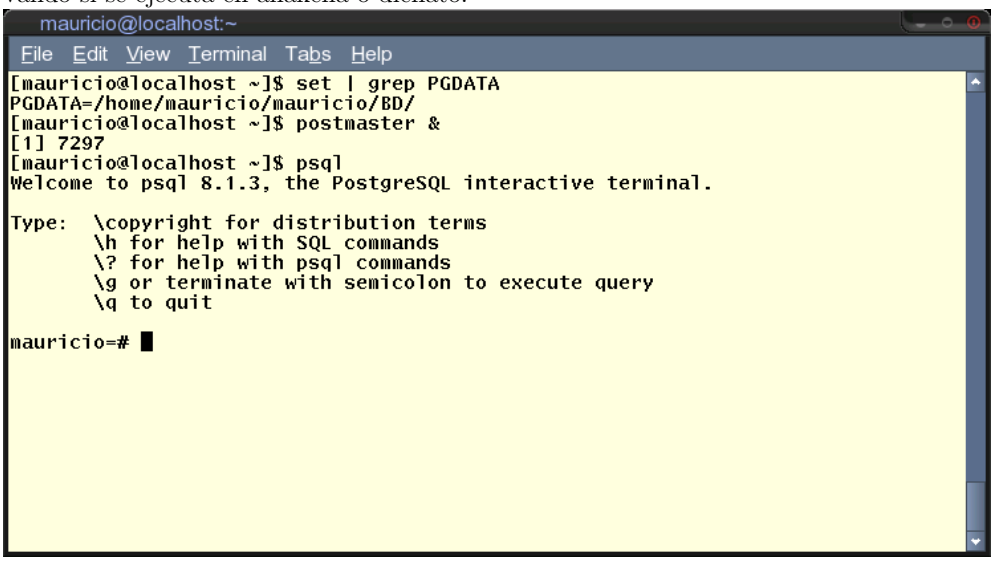


(No vea mi escritorio)

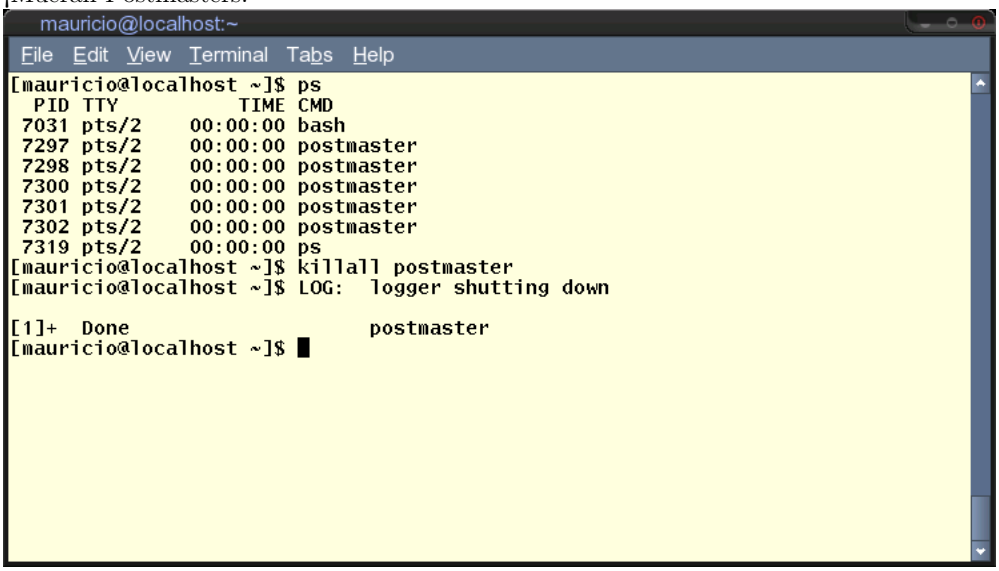
Pero es necesario que Postmaster siga en ejecución. Si el proceso es eliminado, psql dejará de funcionar (lo que es bastante poco deseable).



Otra alternativa, más cómoda -a mí parecer-, es ejecutar Postmaster en el mismo terminal y dejarlo en el fondo usando &. Cuando se desee destruir el proceso, aniquilarlo con kill. O bien, **killall postmaster**. Esto también es válido si se ejecuta en anakena o dichato.



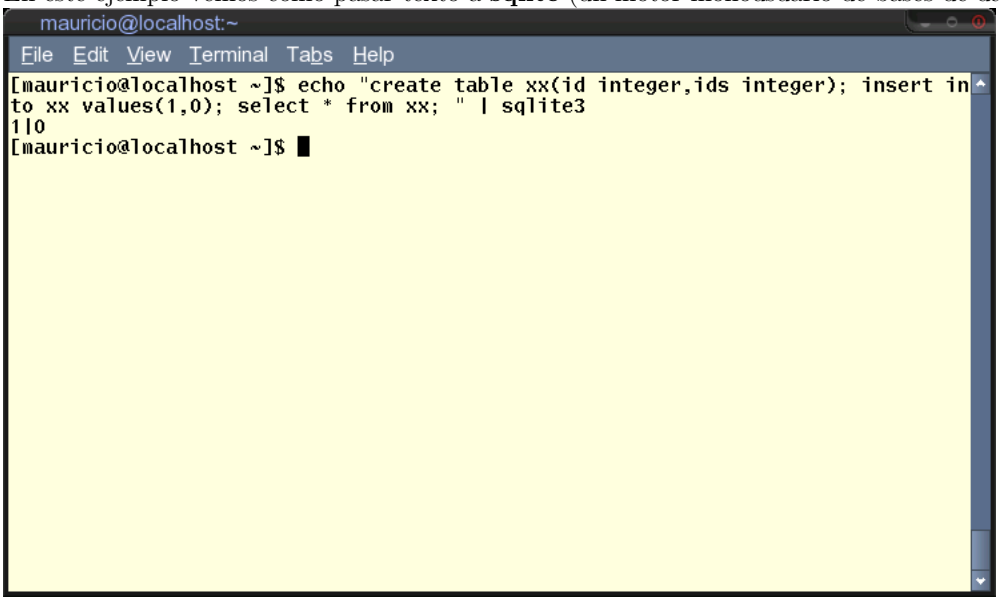
¡Mueran Postmasters!



Una estrategia medianamente práctica para ejecutar comandos en bases de datos desde un terminal

Generalmente los sistemas administradores de bases de datos permiten el acceso desde terminal. Eso implica que se les pueden pasar comandos desde un **pipe** (`|`). El texto emitido por un proceso es pasado a otro proceso en el vuelo.[1]

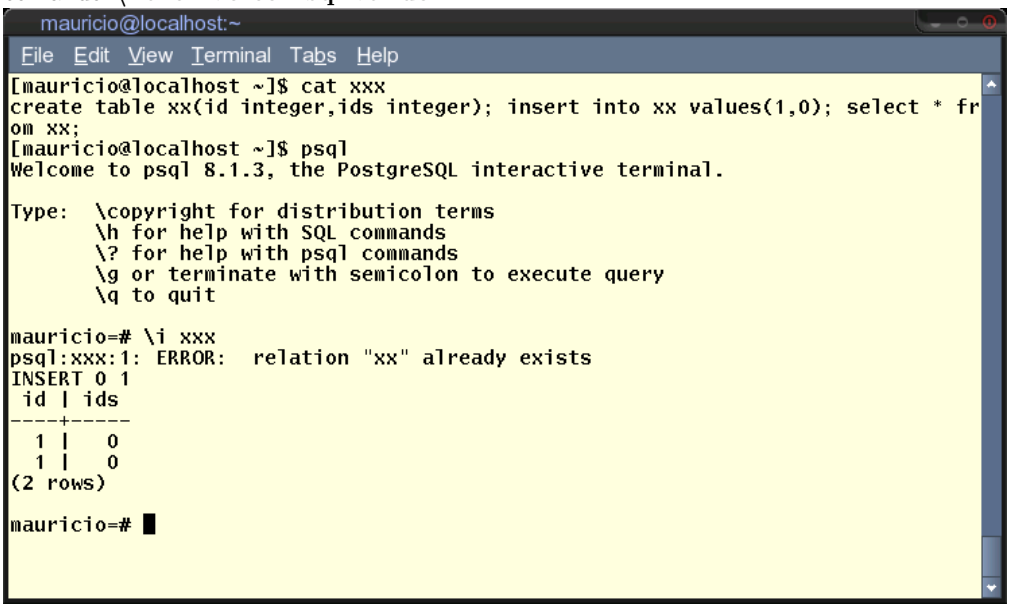
En este ejemplo vemos cómo pasar texto a **sqlite** (un motor monousuario de bases de datos):



Luego del pipe, la interfaz del sistema de base de datos cesará de funcionar (típicamente).

1 Esto se asegura en sistemas operativos que cumplen con POSIX. Probablemente esto no sea válido en Windows u otro sistema operativo no POSIX.

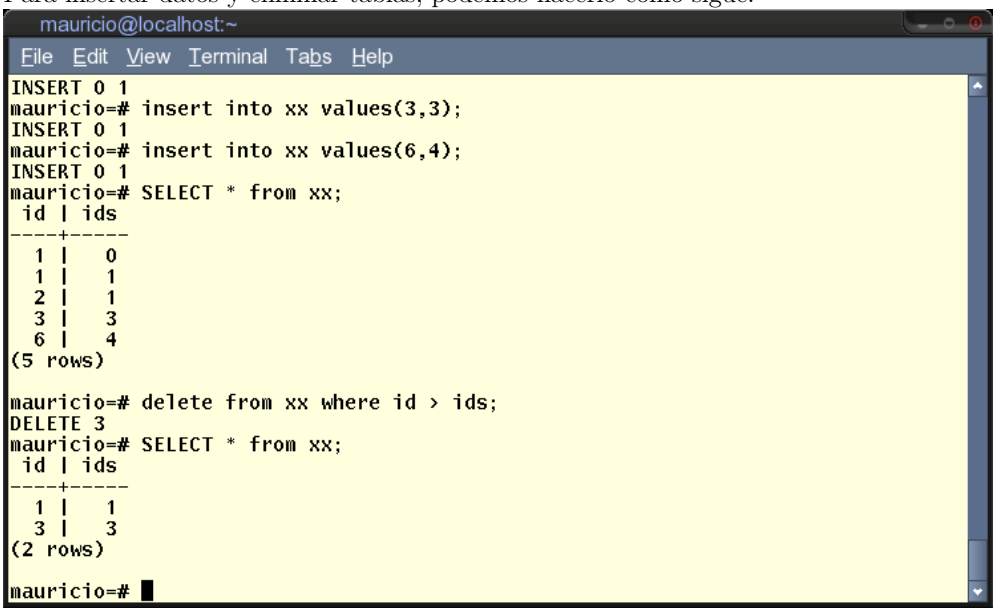
Esto mismo es válido con **psql**. Sin embargo, es más fácil dejar todo en un archivo separado y luego ejecutar el comando **\i archivo-con-sql-válido**.



Otras ayudas prácticas para el proyecto

Para listar todo lo que hay en la base de datos, **\d** es la solución.

Para insertar datos y eliminar tablas, podemos hacerlo como sigue:



Espero que esto les sea de ayuda, ¡saludos!

PD: Disculpen el enfoque de bajo nivel, pero yo soy 100% linuxero :P.
Mauricio