

Postgres Database: Getting Started

obtain, install, and begin using

Basil Bourque
basil.bourque@pobox.com

Why?

I ❤️ DB



Me

- Basil Bourque
- Decades of developing custom-crafted database-backed business apps.
- Looking for niche product opportunities, micro-startups.
- Email: basil.bourque@pobox.com
LinkedIn: [basil.bourque](#)
Google+ [Basil Bourque](#)

Agenda

- Overview

- Run through slides, hyper-fast

- Demo

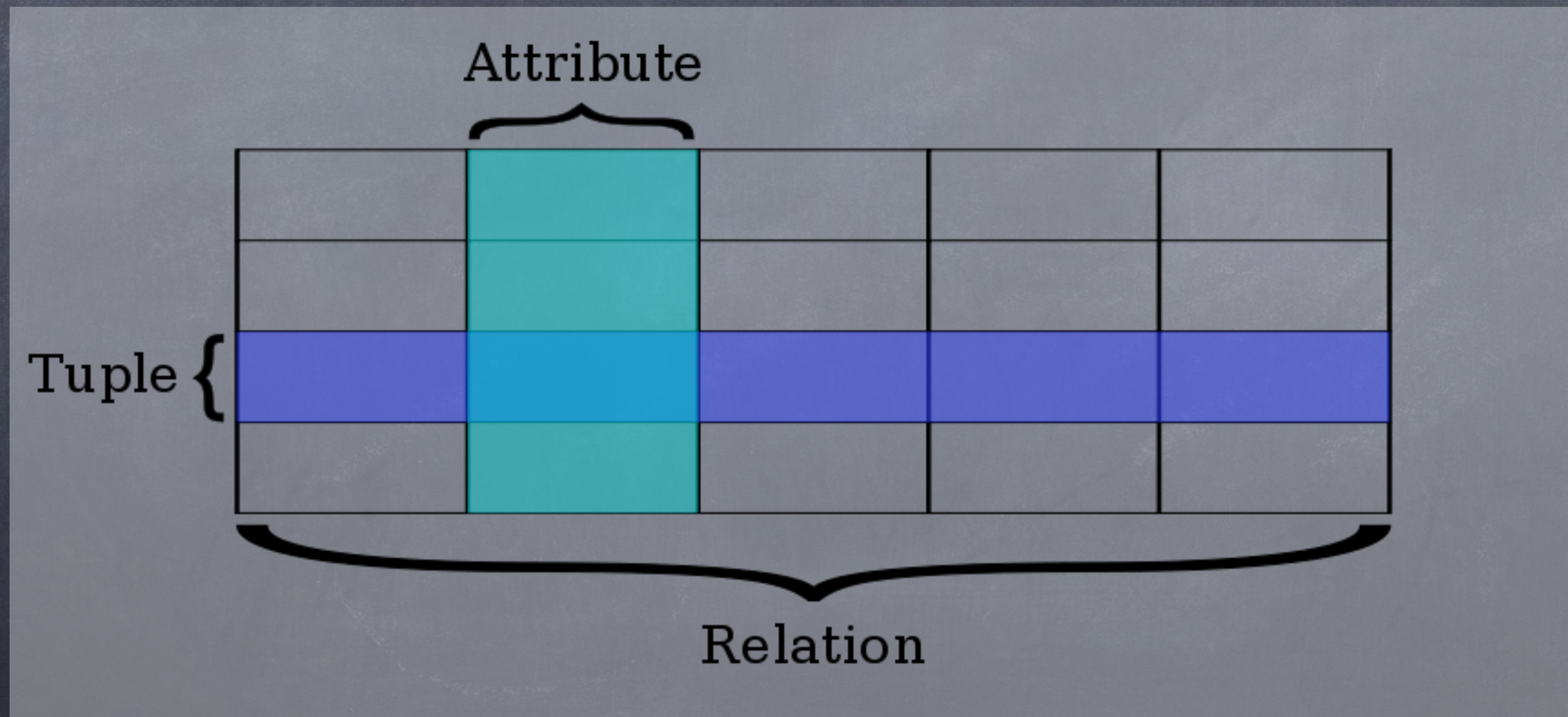
- Install Postgres in a virtual machine, live

- Review

- Go through slides in detail

What is Postgres?

- World's most advanced open source database
- A relational database server



Customer

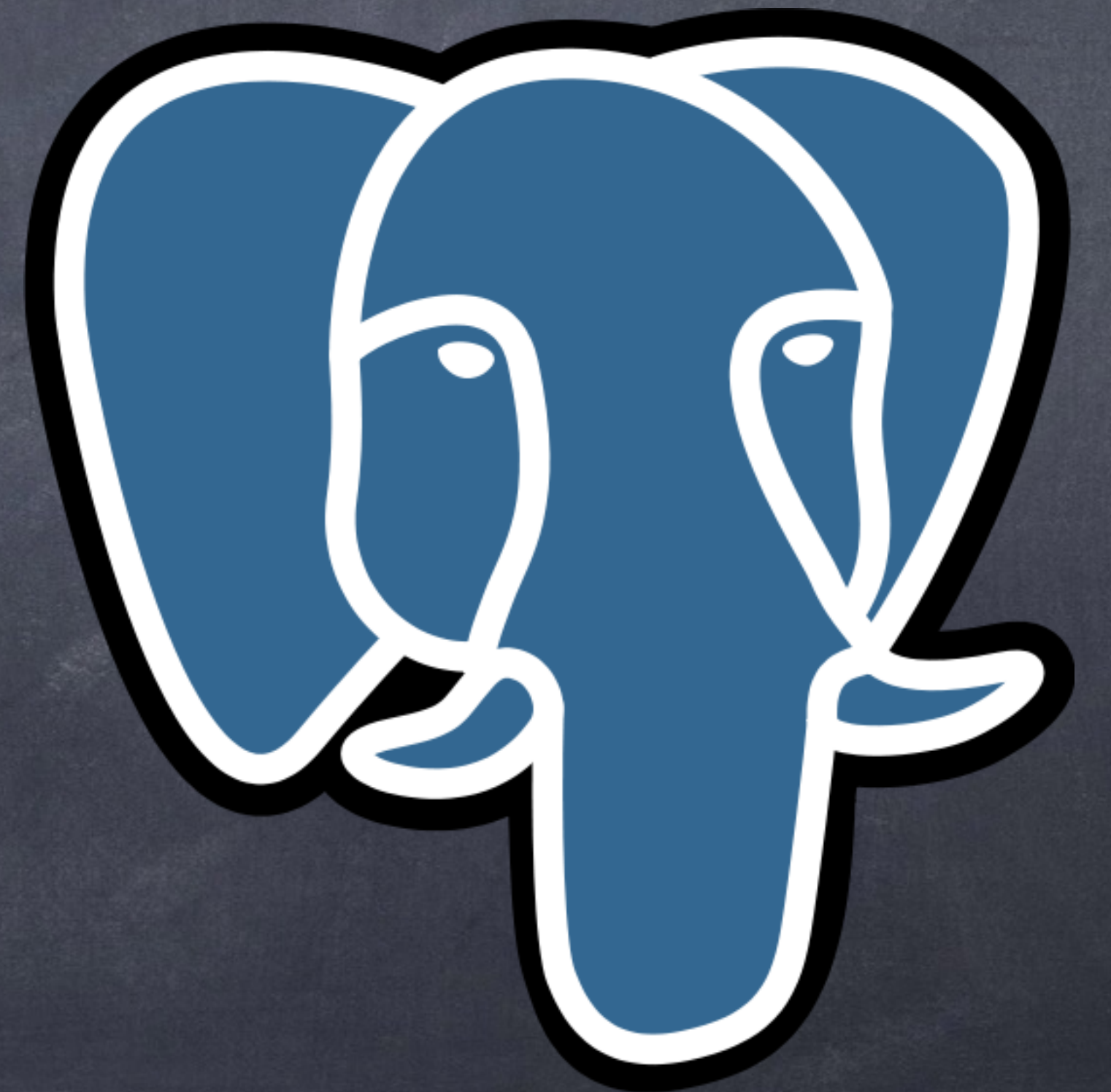
Key	First Name	Last Name	Phone	Email
311	Sally	Jones	425.555.1324	s.jones@acme.com
312	Jarrold	Barkley	206.555.3454	j.barkley@example.com
313	Wendy	Melvin	415.555.2343	wendy@wendyandlisa.com
314	Lisa	Coleman	425.555.7282	lisa@wendyandlisa.com
315	Jesse	Johnson	507.555.7865	j.j@guitar.com
316	Jean-Luc	Martin	212.555.2244	jean-luc.martin@example.com

Pet

Key	Species	Name	Description	Foreign Key
1032	Dog	Pascal	Black Lab	312
1033	Cat	Sprinkles	Tuxedo cat	315
1034	Bird	Tutti-Frutti	Cockatiel	314

Post-what?

- Inventor of "Ingres" database later created Postgres
 - Get the play-on-words?
- Related items often named "pg..."
- Formal name "PostgreSQL"
 - Informally, "Postgres"



Open Source

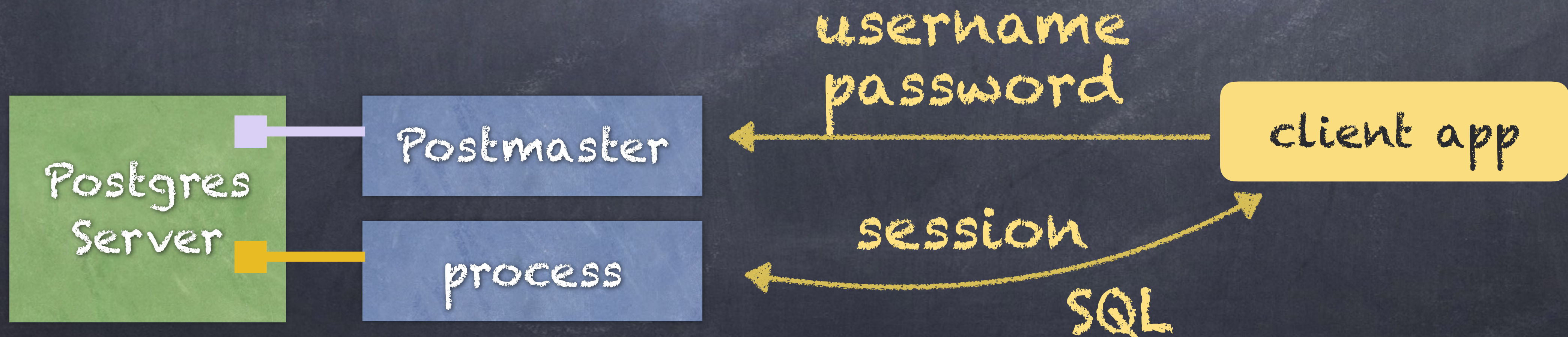
- Truly open source. BSD/MIT-style license, not GPL.
- Simplest of licenses: do-anything-but-sue-me
- "Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies."
 - ... no liability ...
 - ... no warranty ...
- Forked many times for special-purpose or commercial

Marketplace

- Competes for mind-share with MySQL
- Similar in power, features, and reliability to:
 - Microsoft SQL Server
 - Oracle (the basic db product)
 - IBM DB2
- Known for protecting your data, safe & sane development practices. Enterprise-quality.

Black Box

- Invisible, no user-interface. Cannot directly view data
- Separate client app connects to Postgres
- Background process, the "PostMaster", listens
- Unlike FileMaker, 4D, MS Access.



SQL

- SQL = Structured Query Language
- More than queries. Also defines the database.
 - DML (data manipulation), DDL (db definition)
- IBM
 - Originally intended for end-users (ha!)
 - Actually used by programmers & DB admins

Bundled Clients

- pgAdmin
- GUI
- Open-source
- Clunky, but useful
- psql

The screenshot displays the pgAdmin III interface. The top menu includes File, Edit, Plugins, View, Tools, and Help. The Object browser on the left shows a tree view of server groups, including 'Serveurs (1)' and 'Tunnels SSH (3)'. The main pane shows the 'Properties' tab for a table named 'pgbench_accounts', with details such as Name, OID (16391), Owner (guillaume), Tablespace (pg_default), and Rows (estimated) (100000). Below this, a 'Query' window is open, showing the SQL Editor with the query 'select * from pg_stat_activity'. The Output pane at the bottom displays a table with columns: datid, datname, procpid, usesysid, username, application_name, client_addr, client_port, and backtime. The table contains four rows of data.

	datid oid	datname name	procpid integer	usesysid oid	username name	application_name text	client_addr inet	client_port integer	backtime
1	11874	postgres	12482	10	guillaume		:::1	49467	2010
2	33018	b1	12483	10	guillaume		:::1	49468	2010
3	16384	benchs	12485	10	guillaume		:::1	49469	2010
4	16384	benchs	12490	10	guillaume		:::1	49470	2010

```
PostgreSQL Frontend 7.4.6 - psql -h sql.servers.home test test
C:\Documents and Settings\testuser>psql -h sql.servers.home test test
Password:
Welcome to psql 7.4.6, the PostgreSQL interactive terminal.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help on internal slash commands
      \g or terminate with semicolon to execute query
      \q to quit
```

Third-Party Admin Clients

- Many GUI clients available, akin to pgAdmin
- Some commercial. Some open-source/libre.
- Most multi-db (not just Postgres)
- Cross-platform (often Java-based)
- Platform-specific (PG Commander for Mac)
- IDEs with database features (NetBeans, IntelliJ)
- Community Guide to PostgreSQL GUI Admin Tools

Your Own App

- CRUD (Create, Read, Update, Delete)
- SQL, the native lingo, plain text
 - "SELECT name, phone FROM customer"
- App ↔ Postgres
 - C-based library provided (libpq)
 - Drivers mediate (JDBC for Java, ODBC in Windows, ...)
 - Java, Python, Go, PHP, Perl, JavaScript, Tcl, .NET, Lisp

Why db for your app?

- ◉ Persistence
 - ◉ Save your data to storage, when app is quit
- ◉ Delegate
 - ◉ Let db do work of writing, searching, sorting
- ◉ Concurrency
 - ◉ >1 user or thread → Problems!
- ◉ A.C.I.D. = Keep your data safe and intact.

Obtain - Source Code

- Source-code.
- Compile it yourself.
 - C compiler, C89-compliant, GNU make.
- Only artifact delivered by core team.
- Compatible for many OSes (BSD, Linux, Mac, Win...)
- Over a dozen chip architectures

Obtain - Cloud provisioned

- Heroku - Heroku Postgres
- Amazon AWS - RDS for Postgres
- IBM Bluemix - PostgreSQL Service

Obtain - Prebuilt

- App repositories
 - BSD 'Ports and Packages' Collection
 - apt-get , yum , rpm , and so on
- **Installer** (double-clickable, GUI)
- Mac only: **Postgres.app**
 - Double-clickable to start/stop Postgres (no install)
 - Provided as courtesy by heroku.com

My pref: Download Installer

- ◉ Consulting companies with expertise in Postgres
 - ◉ 2nd Quadrant (see booth at trade show)
 - ◉ EnterpriseDB (enterprisedb.com)
 - ◉ Provides **installers as courtesy** to community
 - ◉ **Does *not*** own Postgres, nor control Postgres
 - ◉ Builds and serves GUI installers
 - ◉ **Linux, Windows, Mac OS X.**
 - ◉ <http://www.EnterpriseDB.com/downloads>

» Downloads

Binary

Source

» Software Catalogue

» File Browser

Downloads

PostgreSQL Core Distribution

The core of the PostgreSQL object-relational database management system is available in several source and binary formats.

Binary packages

Pre-built binary packages are available for a number of different operating systems:

- BSD
 - [FreeBSD](#)
 - [OpenBSD](#)
- Linux
 - [Red Hat](#) family Linux (including CentOS/Fedora/Scientific/Oracle variants)
 - [Debian](#) GNU/Linux and derivatives
 - [Ubuntu](#) Linux and derivatives
 - [SuSE](#) and OpenSuSE
 - [Other](#) Linux
- [Mac OS X](#)
- [Solaris](#)
- [Windows](#)

Source code

The source code can be found in the main [file browser](#) or you can access the source control repository directly at git.postgresql.org. Instructions for building from source can be found in the [documentation](#).

Alpha/Beta/RC Releases and development snapshots (unstable)

There are source code and binary [packages](#) of beta and release candidates, and of the current development code available for testing and evaluation of new features. Note that these builds should be used **for testing purposes only**, and not for production systems.

If you want to
download a
local installer:

Navigate to a
page mentioning
"Graphical
installer"

Links to page at
EnterpriseDB.com



Postgres Advanced Server



PostgreSQL

xDB Replication Server






Postgres Enterprise Manager

EDB Fallover Manager

EDB Backup & Recovery Tool

Components & Other Downloads

Please make sure you upgrade your components to the latest patch / point releases using StackBuilder immediately after installation is complete.

Current Release		
PostgreSQL 9.5		
Recommended: Try EnterpriseDB tools by running Stackbuilder after installing PostgreSQL, expanding Trial Products and selecting Components under EnterpriseDB		
Windows-32		Download
Windows-64		Download
Linux x86-32		Download
Linux x86-64		Download
Mac		Download

Benefits

-  World's most advanced open source database
-  25+ years enterprise-class development by a large independent community
-  Most secure open source database
-  Fully supported by EnterpriseDB's own PostgreSQL experts
-  Active world-wide deployments in public and private sector organizations of all sizes and missions

Previous Release		
PostgreSQL 9.4		
Recommended: Try EnterpriseDB tools by running Stackbuilder after installing PostgreSQL,		

Benefits

Heavyweight Install

- New user account at OS level: 'postgres'
- **Security.** Database folders owned by this user.
- Few bits placed in various folders.
 - Removal is possible but tedious.
 - Look for uninstaller script.
 - OS user account may be permanent.

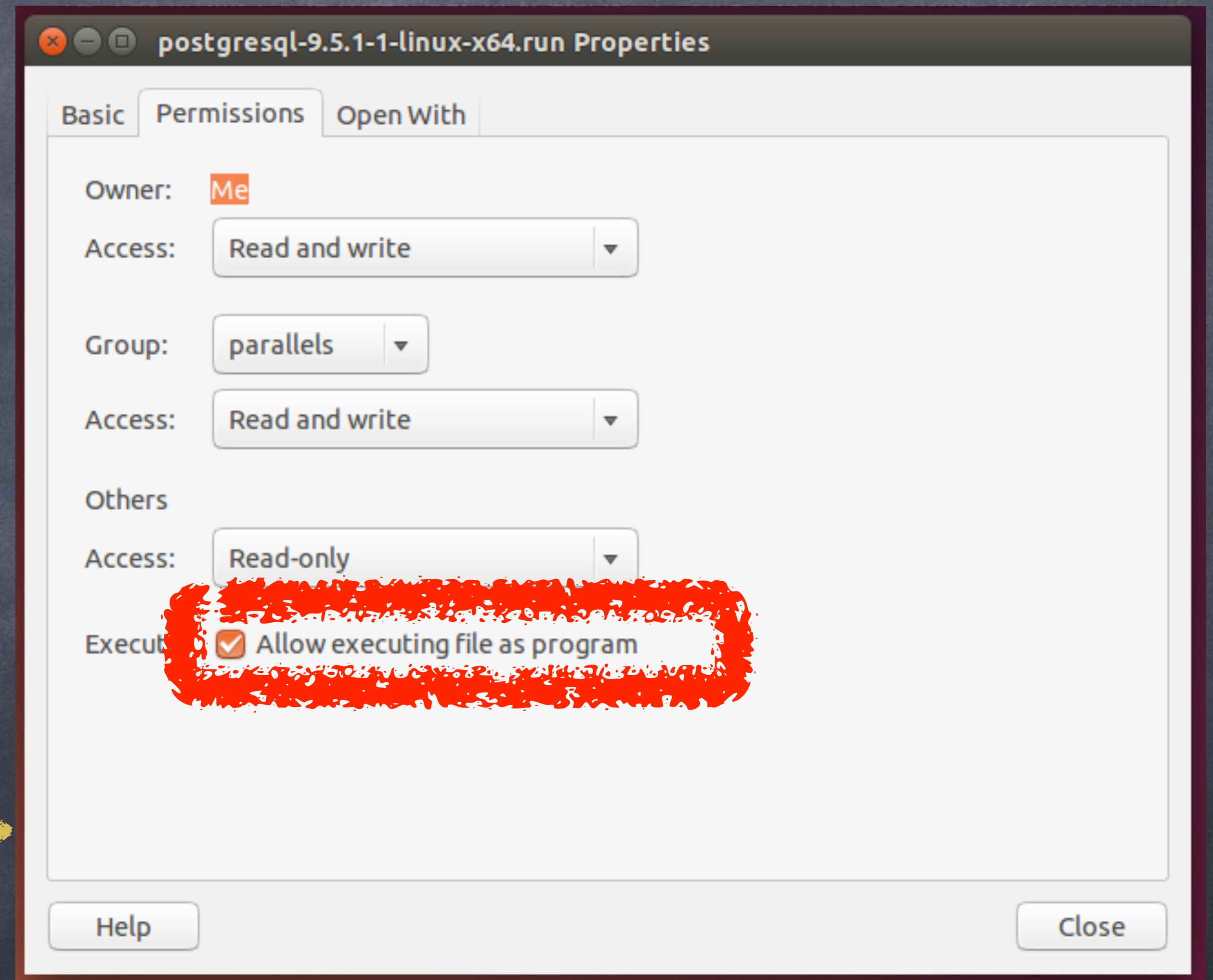
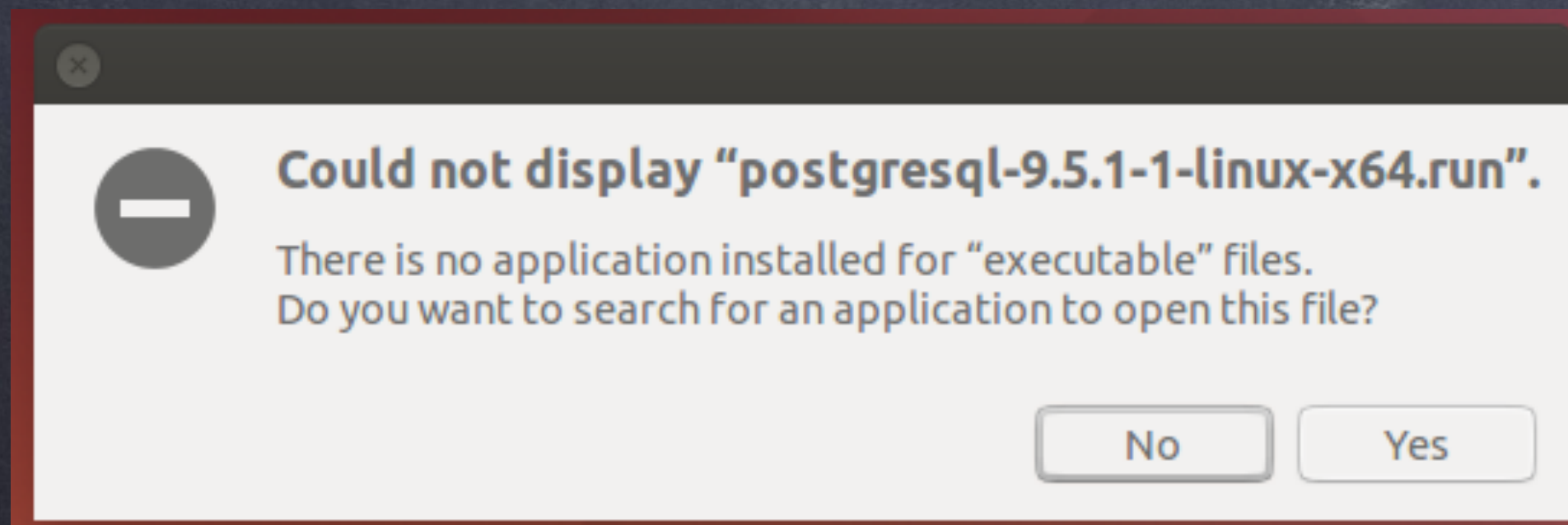
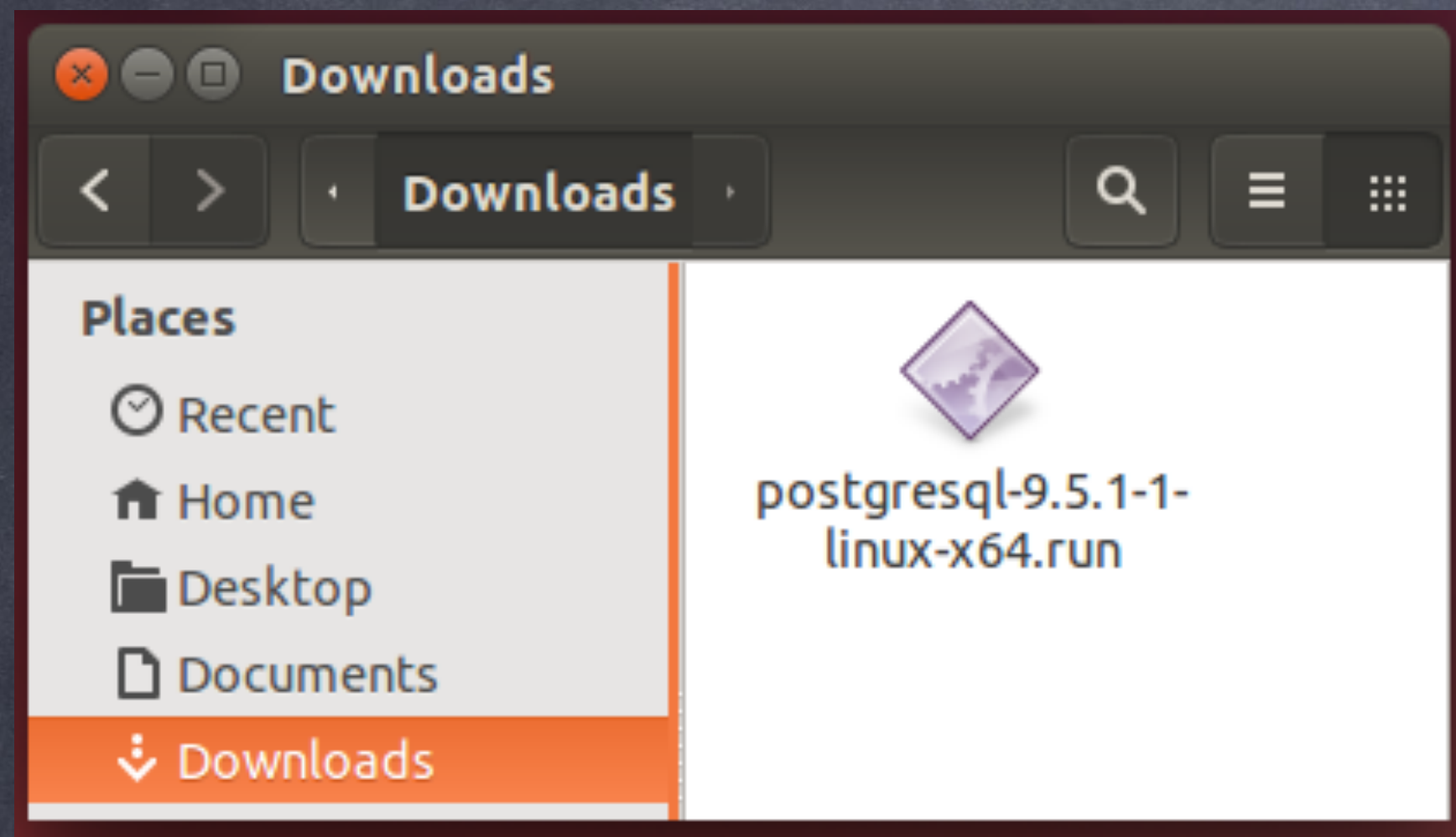
Virtual Machine

- Tip: Practice your Postgres inside a VM
- Stored in single file on real computer. Trash later.
- VM technology is surprisingly good
 - VirtualBox (open-source, Libre, Oracle/Sun)
 - Parallels (commercial, demo today)
 - Fusion (commercial, by VMware)

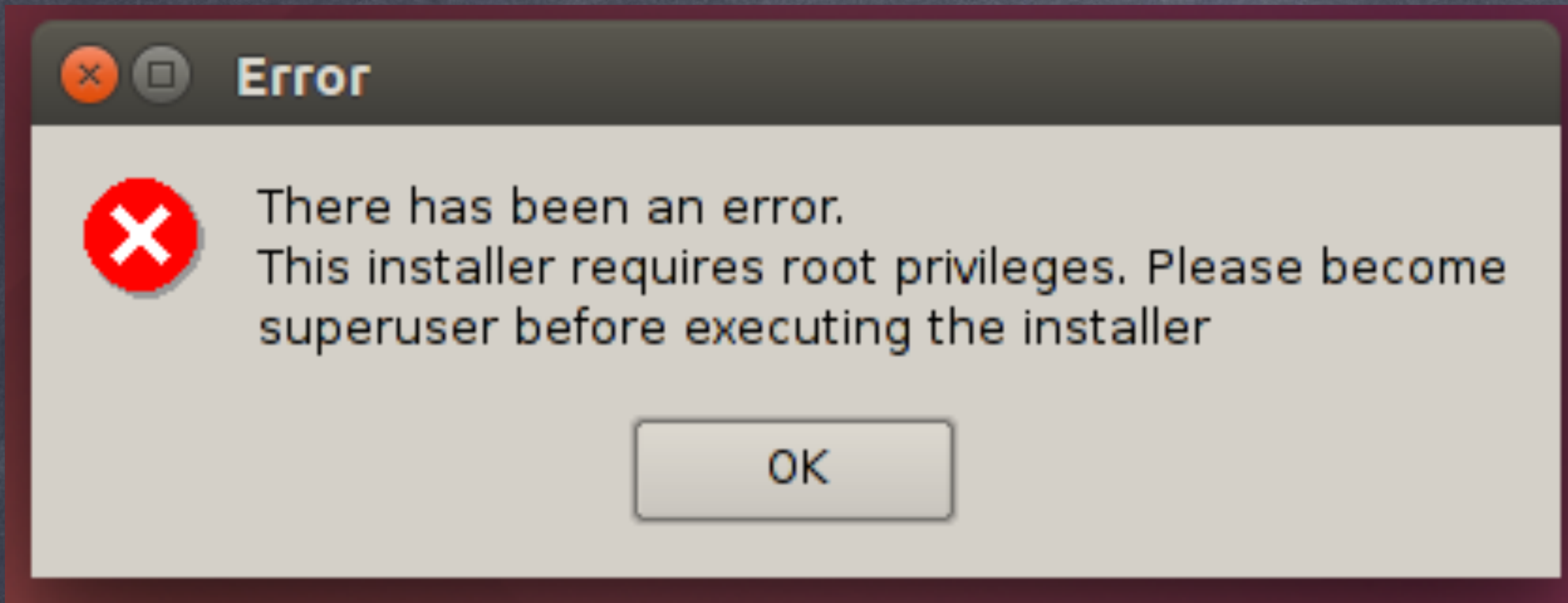
Glossary

- "oid" = internal identifier for various objects 😐
- "FTS" = Full Text Search (indexing documents) 😐
- "TableSpace" = Physical storage locations of db 😐

Enable execution



sudo



```
parallels@ubuntu: ~/Downloads
parallels@ubuntu:~$ cd ~/Downloads/
parallels@ubuntu:~/Downloads$ sudo ./postgresql-9.5.1-1-linux-x64.run
[sudo] password for parallels:
```

Wizard

Setup

Installation Directory

Please specify the directory where PostgreSQL will be installed.

Installation Directory

Setup

Data Directory

Please select a directory under which to store your data.

Data Directory

Setup

Password

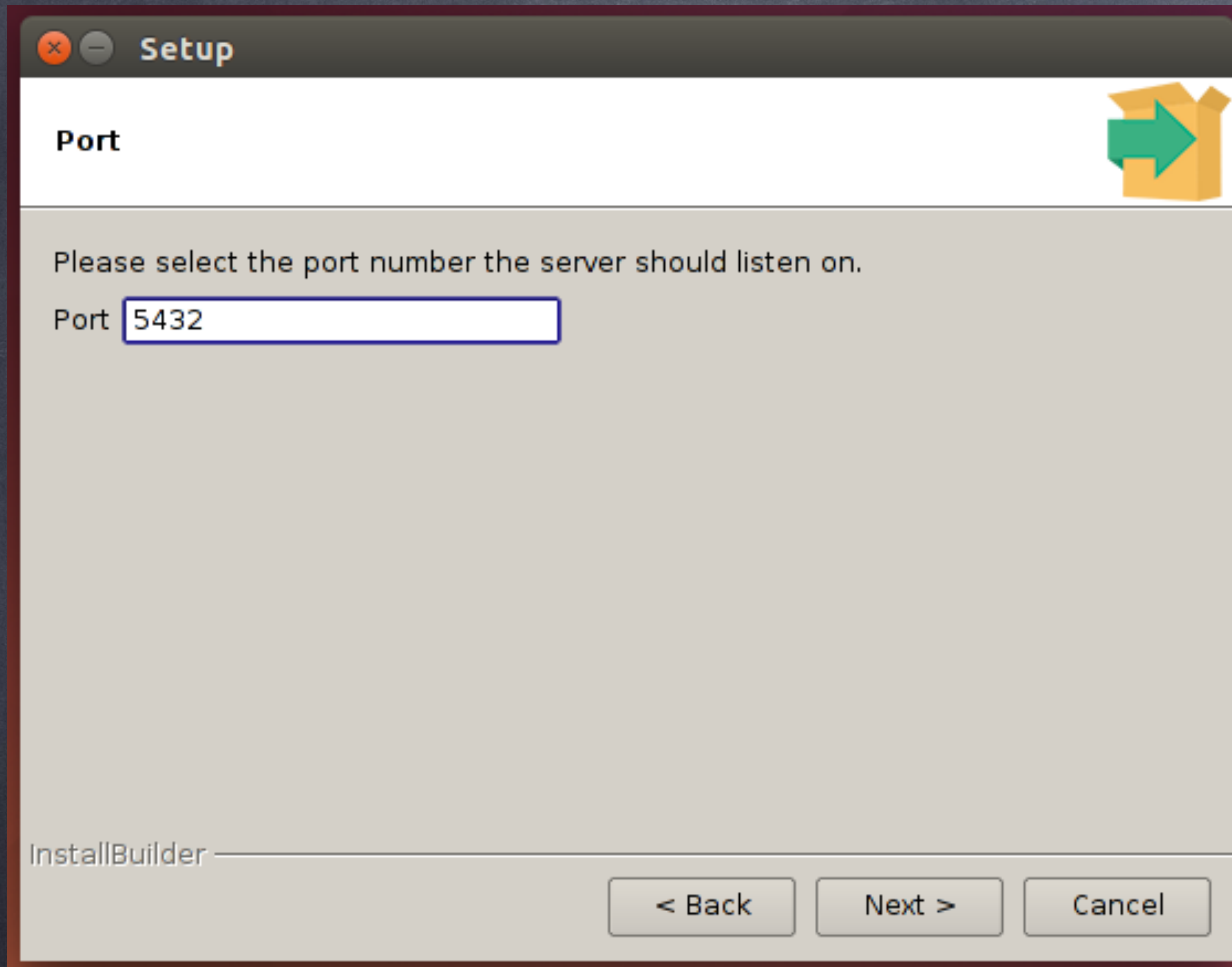
Please provide a password for the database superuser (postgres). A locked Unix user account (postgres) will be created if not present.

Password

Retype password

- Go with defaults
- Write down that password!!!

Port

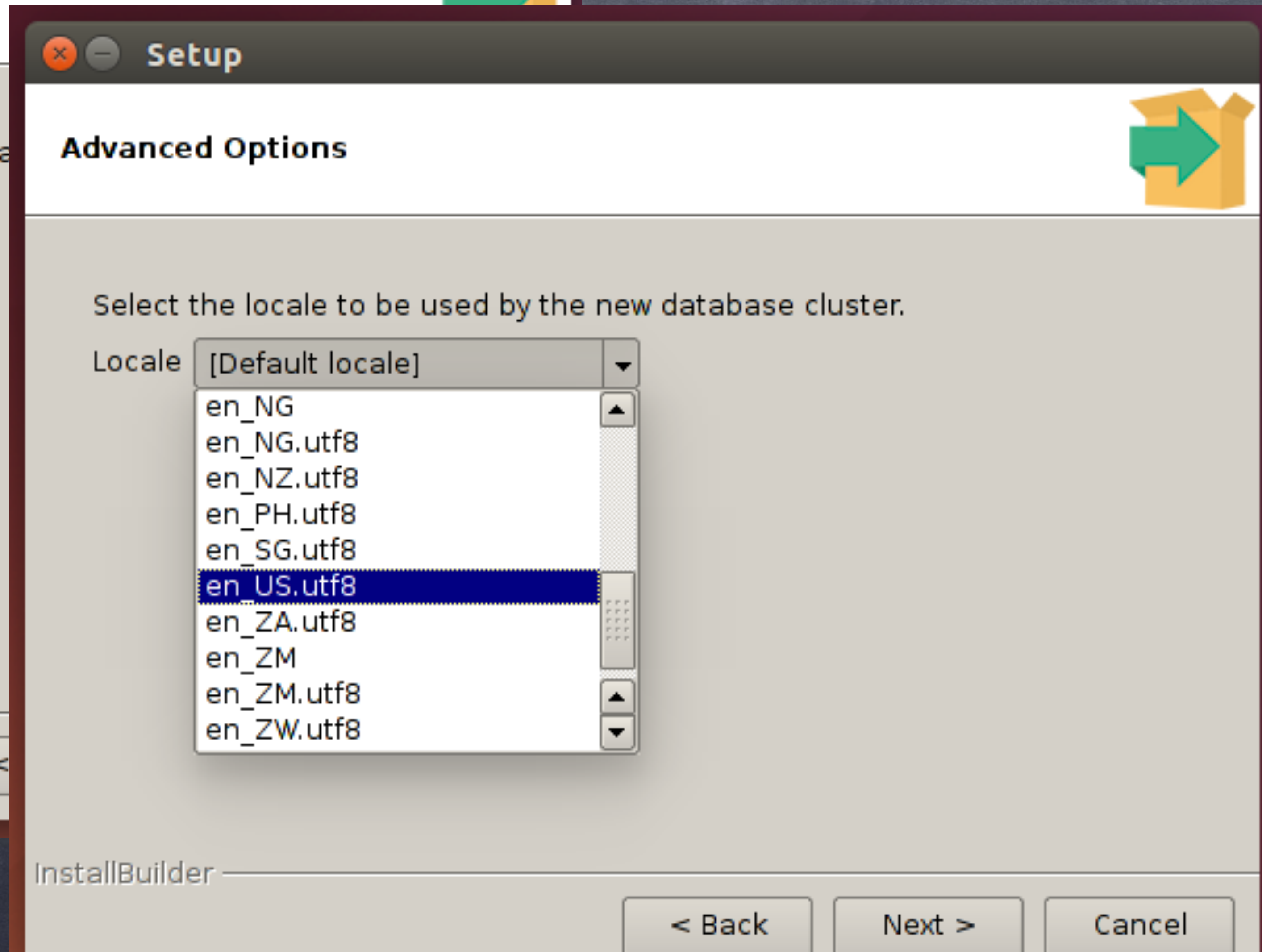
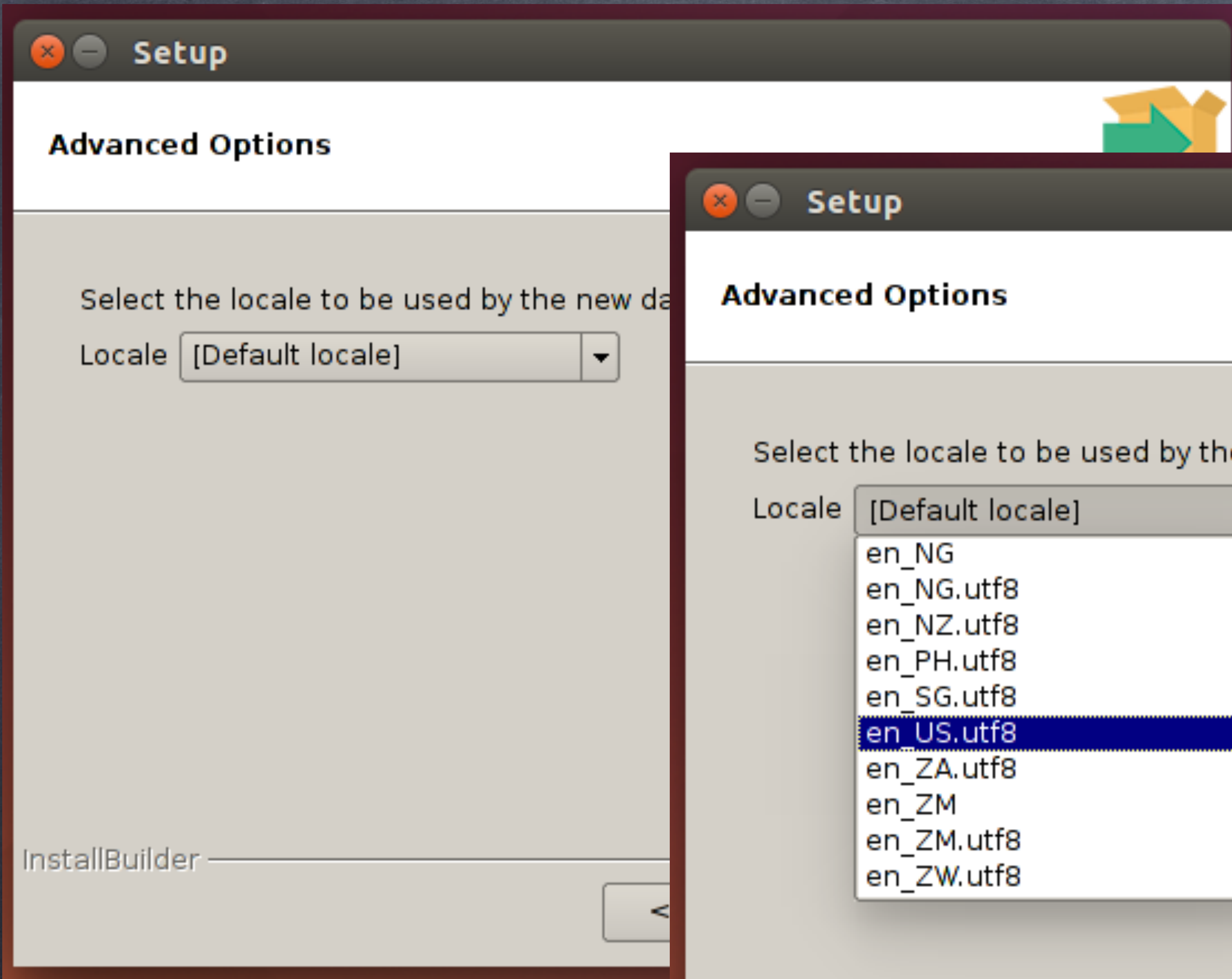


5432 = default
Use any number

Listens for
incoming
connections by
admin apps &
client apps

Locale

Override default:
en_US.utf8



Anatomy

PORT (5432)

CLUSTER (database server)
(typically a version of Postgres, 9.3 or 9.4 etc.)

CATALOG (database)

SCHEMA (namespace)

TABLE	TABLE
row	row
row	row
row	row

SCHEMA (namespace)

TABLE	TABLE
row	row
row	row
row	row

CATALOG (database)

SCHEMA (namespace)

TABLE	TABLE
row	row
row	row
row	row

SCHEMA (namespace)

TABLE	TABLE
row	row
row	row
row	row

- Defined by the SQL standard
- Compliance is a priority
- Installer is setting default for cluster.
- Used as default for each new database (catalog)

Locale

- "cultural preferences regarding alphabets, sorting, number formatting"
- "C/POSIX" = no locale
- First two fixed for each database (Catalog)
- Other four are settings in 'postgresql.conf'

LC_COLLATE	String sort order
LC_CTYPE	Character classification (What is a letter? Its upper-case equivalent?)
LC_MESSAGES	Language of messages
LC_MONETARY	Formatting of currency amounts
LC_NUMERIC	Formatting of numbers
LC_TIME	Formatting of dates and times

- en = English
- US = United States
- utf8 = UTF-8 Unicode encoding

StackBuilder

Setup



PostgreSQL

Packaged by:
EDB
ENTERPRISEDB

Completing the PostgreSQL Setup Wizard

Setup has finished installing PostgreSQL on your computer.

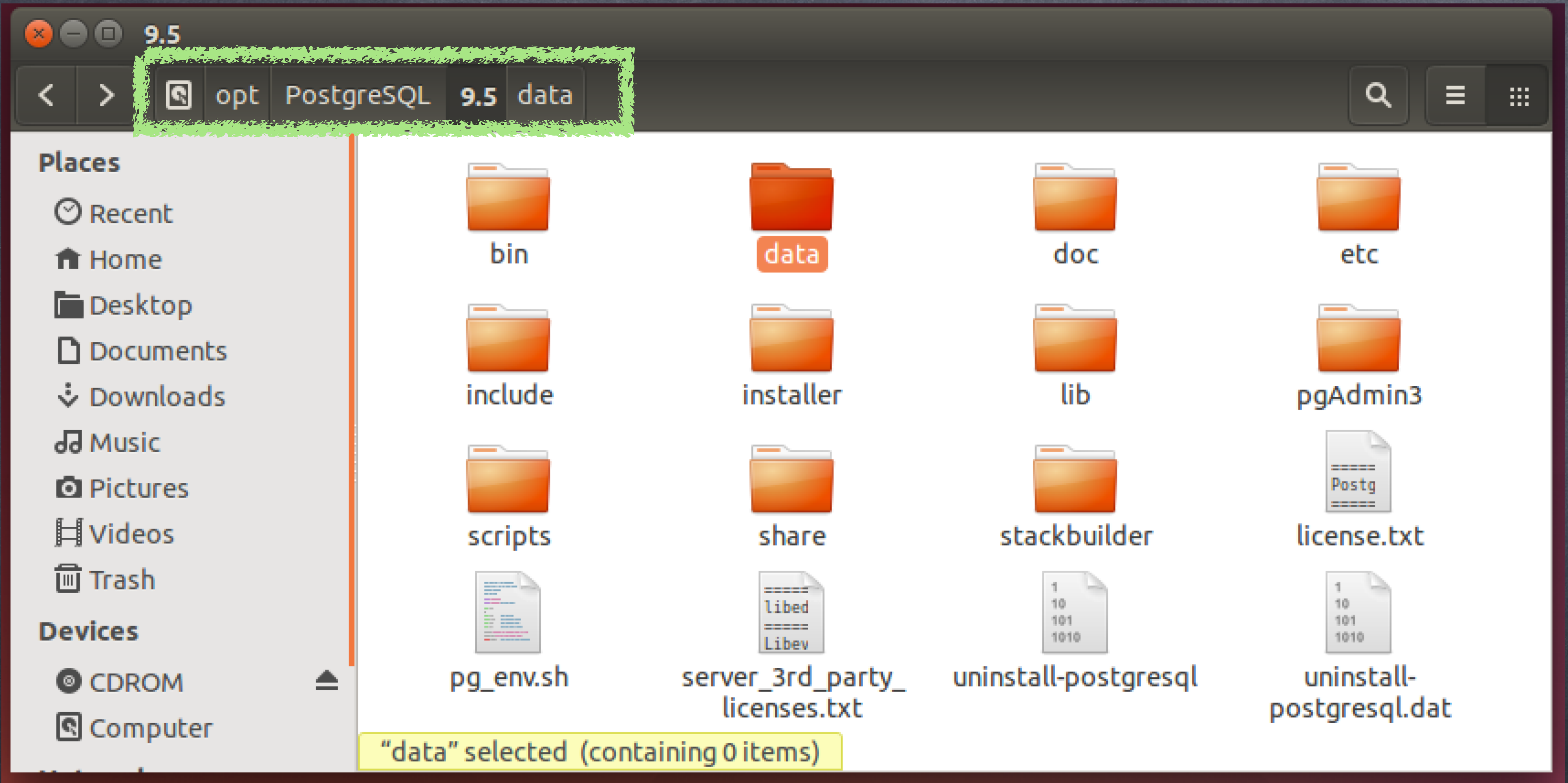
Launch Stack Builder at exit?

Stack Builder may be used to download and install additional tools, drivers and applications to complement your PostgreSQL installation.

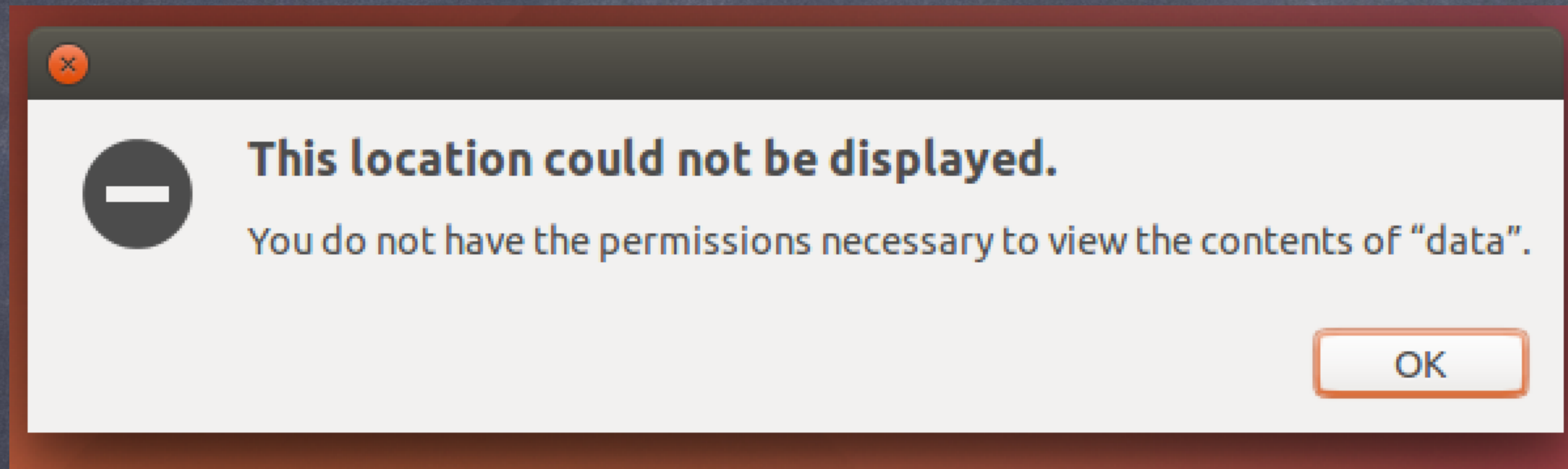
< Back Finish Cancel

- Categories
 - Add-ons, tools and utilities
 - Database Drivers
 - Database Server
 - Registration-required and trial products
 - Replication Solutions
 - Spatial Extensions
 - Web Development

Results

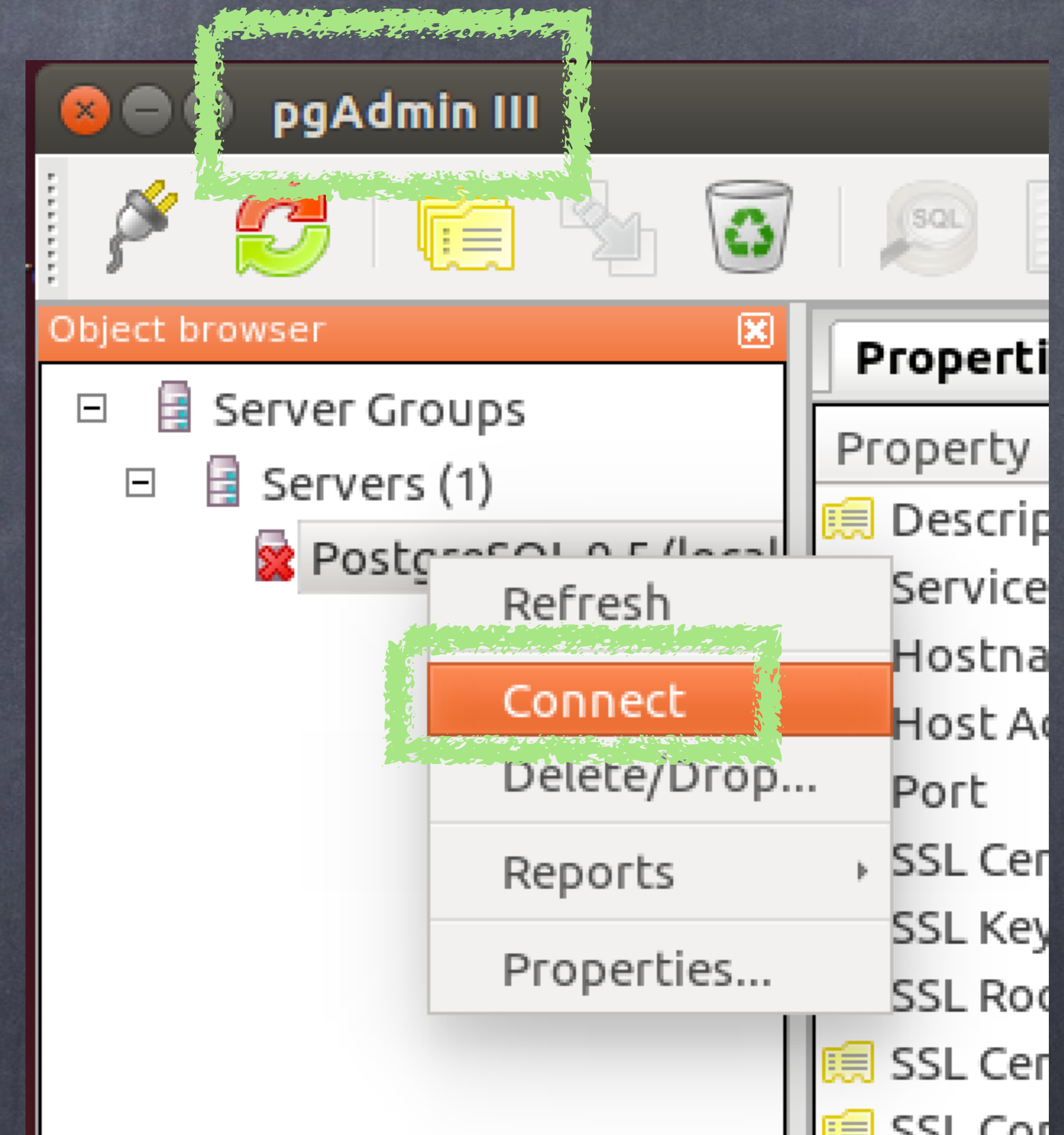


'postgres' user owns data



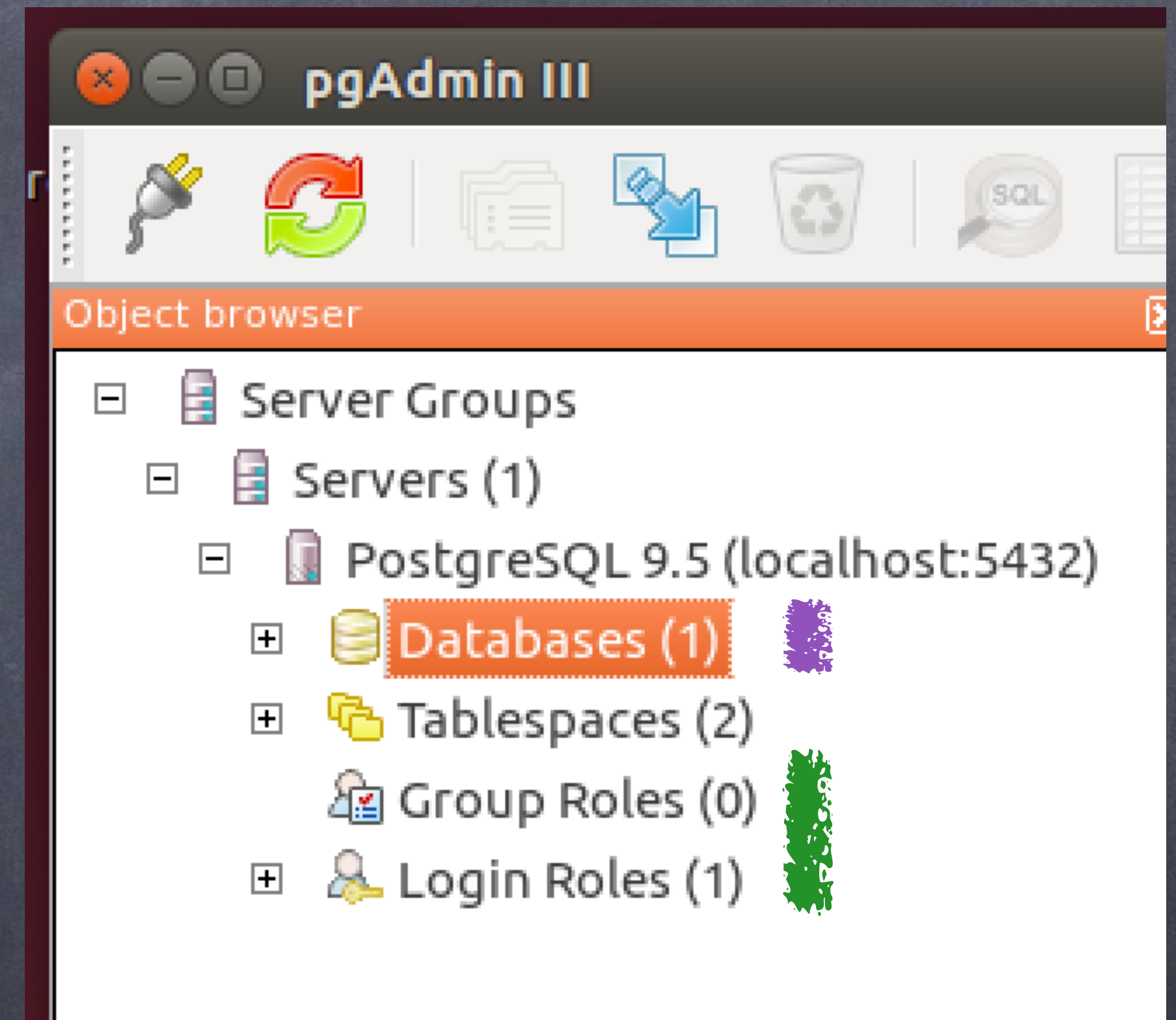
Connect to Server

- When in doubt: Context+click
- You can have more than one Postgres installation ("cluster") running.
- Connect to one Postgres.



Connected to Server

- One database already created
- Users & Groups exist **outside** the database



Create Database

- Naming
 - Use all lowercase for maximum portability
 - ASCII English chars (A-Z, a-z, 0-9) & underscore & some Unicode chars
 - Tip: append trailing underscore
- 'Comment' - Doc that won't get lost

The screenshot shows a 'New Database...' dialog box with the following fields and values:

- Name:** vet_
- OID:** (empty)
- Owner:** (empty)
- Comment:** Vetinary clinic, tracking customers and their pets.

The dialog also features a 'Help' button on the left and 'OK' and 'Cancel' buttons on the right.

SQL

The image shows a 'New Database...' dialog box with several tabs: Properties, Definition, Variables, Privileges, Security Label, and SQL. The SQL tab is selected and highlighted with a green box. Below the tabs, there is a checked checkbox for 'Read only'. The main area contains two text boxes with SQL code, both highlighted with green boxes. The first text box contains the code to create a database named 'vet_' with UTF8 encoding and a connection limit of -1. The second text box contains a comment for the database 'vet_' describing it as a 'Vetinary clinic, tracking customers and their pets.'. At the bottom of the dialog, there are buttons for 'Help', 'OK', and 'Cancel'.

New Database...

Properties Definition Variables Privileges Security Label **SQL**

Read only

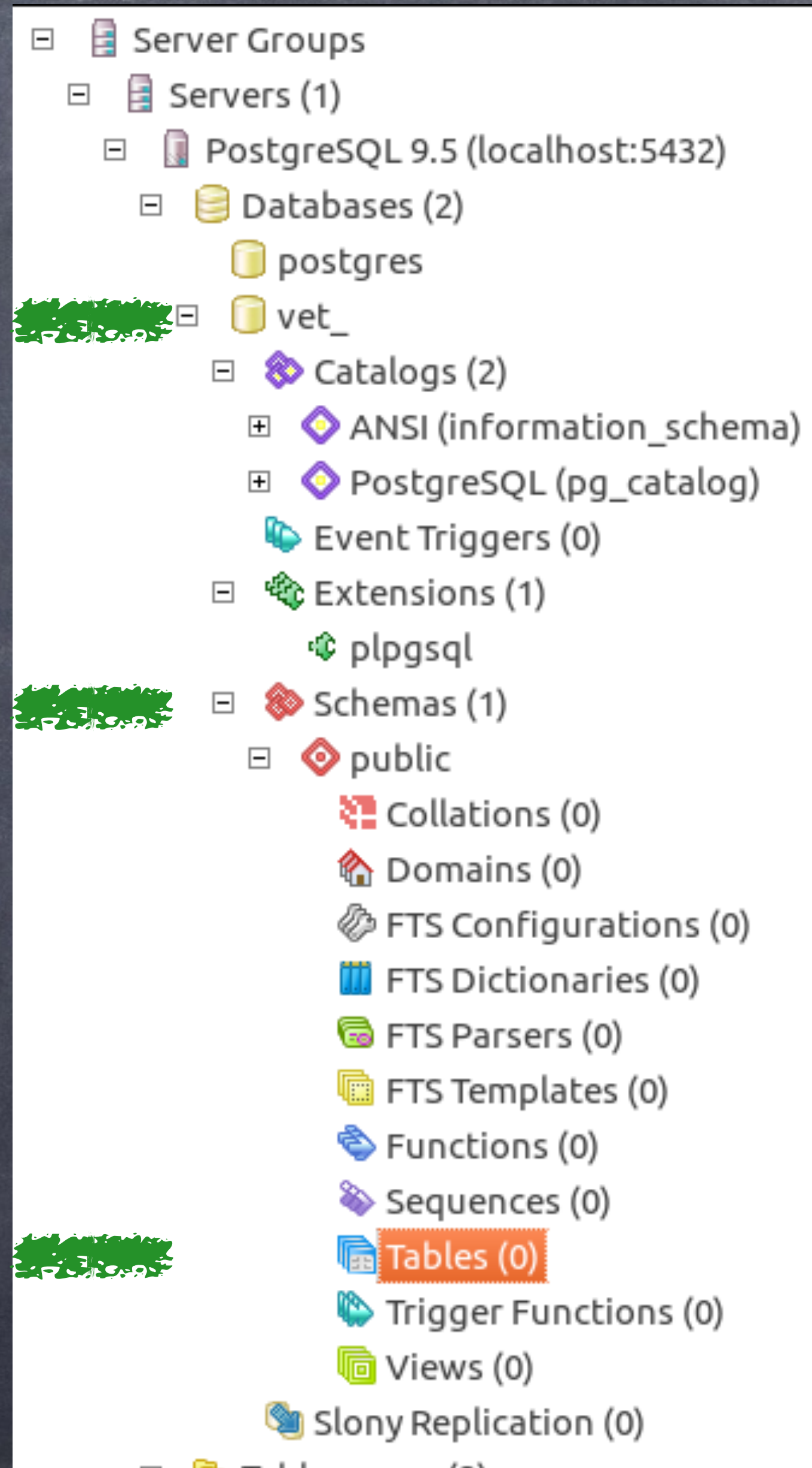
```
CREATE DATABASE vet_  
WITH ENCODING='UTF8'  
CONNECTION LIMIT=-1;
```

```
COMMENT ON DATABASE vet_  
IS 'Vetinary clinic, tracking customers and their pets.';
```

Help OK Cancel

New Database

- 'vet_' database (catalog)
- 'public' schema
 - No tables (except system tables)



New Table

New Table...

Properties Definition Inherits Like Columns Constraints Auto-vacuum Privileges Security Labels SQL

Name

OID

Owner

Schema

Comment

Use Slony

Help OK Cancel

New Column

New Column...

Properties Definition Variables Privileges Security Labels

Name

Data type

Length

Precision

Comment

Use Slony

Help

Please select

- abstime
- abstime[]
- aclitem
- aclitem[]
- bigint
- bigint[]
- bigserial
- bit
- bit[]
- bit varying
- bit varying[]
- boolean
- boolean[]
- box
- box[]
- bytea
- bytea[]
- "char"

Data Types

- Data Type = Postgres understands this data
 - Includes text, numbers, boolean (T/F), date-time
 - BLOB = Binary data not understood by Postgres
- Ignore the "[]" (array, a list of values vs single)
- Ignore synonyms (compatible with other SQL servers)


Main Data Types

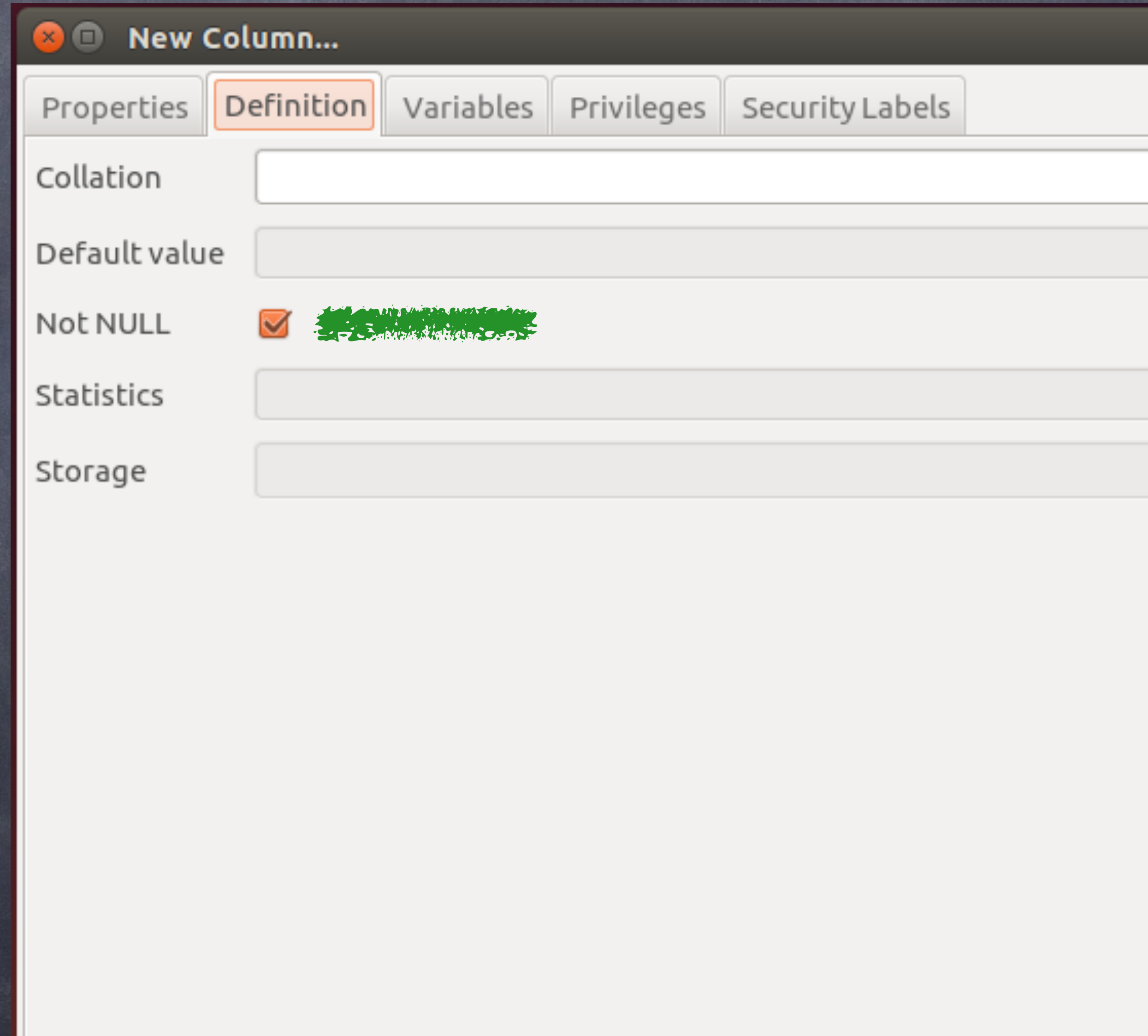
- 'TEXT'
'varchar' is old type, with width
- 'INT2', 'INT4', and 'INT8' (SMALLINT INTEGER BIGINT)
Integer. 16-bit, 32-bit, and 64-bit.
- 'NUMERIC(precision, scale)' (monetary amounts)
Ex: 23.5141 has a precision of 6 and a scale of 4.
- 'REAL', 'DOUBLE PRECISION' (floating-point, 32/64-bit)
Caution: Trades accuracy away for speed
- 'BOOLEAN' (True, False, Null)

Other Data Types

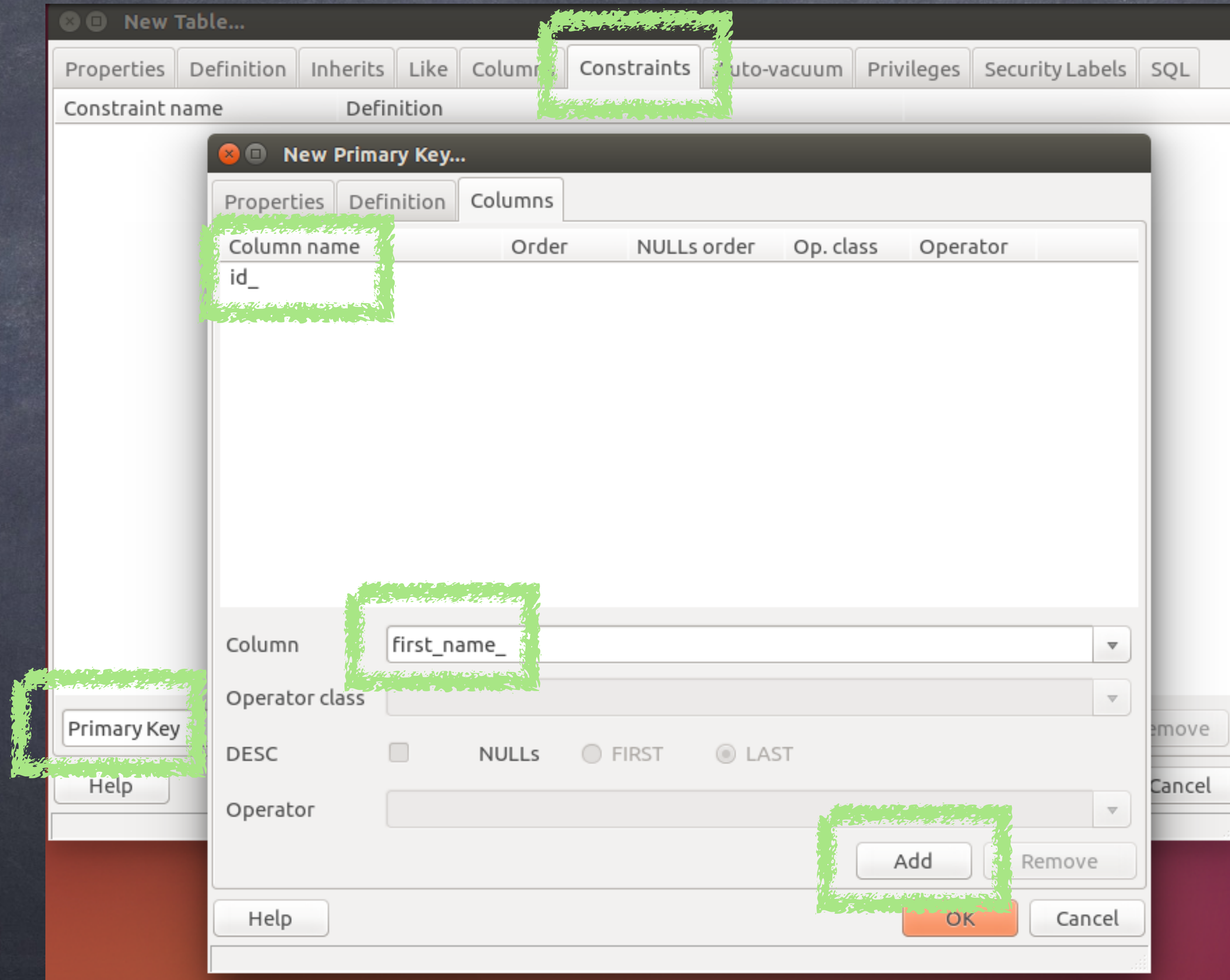
- 'SMALLSERIAL', 'SERIAL', 'BIGSERIAL' (16, 32, 64-bit) auto-incrementing integer
- 'UUID' (128-bit value)
- 'BYTEA' (serves as BLOB type, Binary Large Object)
- 'JSON' & 'XML'
- ...and more... 😐

Null

- Means "value not present"
- Complicates queries
- Complicates sorting
 - Appear at top or at bottom
- Dr. Chris Date advises against using nulls 
- I agree
- **NOT NULL** is a constraint



Primary Key Constraint



Key

• Natural key

- Value present in the data
- Ex: phone, email, emp. #
- My problem:
Always changes
(eventually)
- Change means we must update not ID on this parent but also on every child row in every child table

• Surrogate key

- Artificial value
- Add extra column
- Data Types
 - Integer (incrementing)
SERIAL, BIGSERIAL
 - UUID (MAC + datetime)
- Use DEFAULT to generate

SQL of New Table

New Table...

Properties Definition Inherits Like Columns Constraints Auto-vacuum Privileges Security Labels **SQL**

Read only

```
CREATE TABLE public.customer_  
(  
  id_ serial NOT NULL,  
  first_name_ text NOT NULL,  
  last_name_ text NOT NULL,  
  phone_ text NOT NULL DEFAULT '',  
  email_ text NOT NULL DEFAULT '',  
  CONSTRAINT pkey_employee_ PRIMARY KEY (id_)  
)  
WITH (  
  OIDS = FALSE  
)  
;  
COMMENT ON TABLE public.customer_  
  IS 'Represents a person whose pets visit our clinic.';
```


data entry

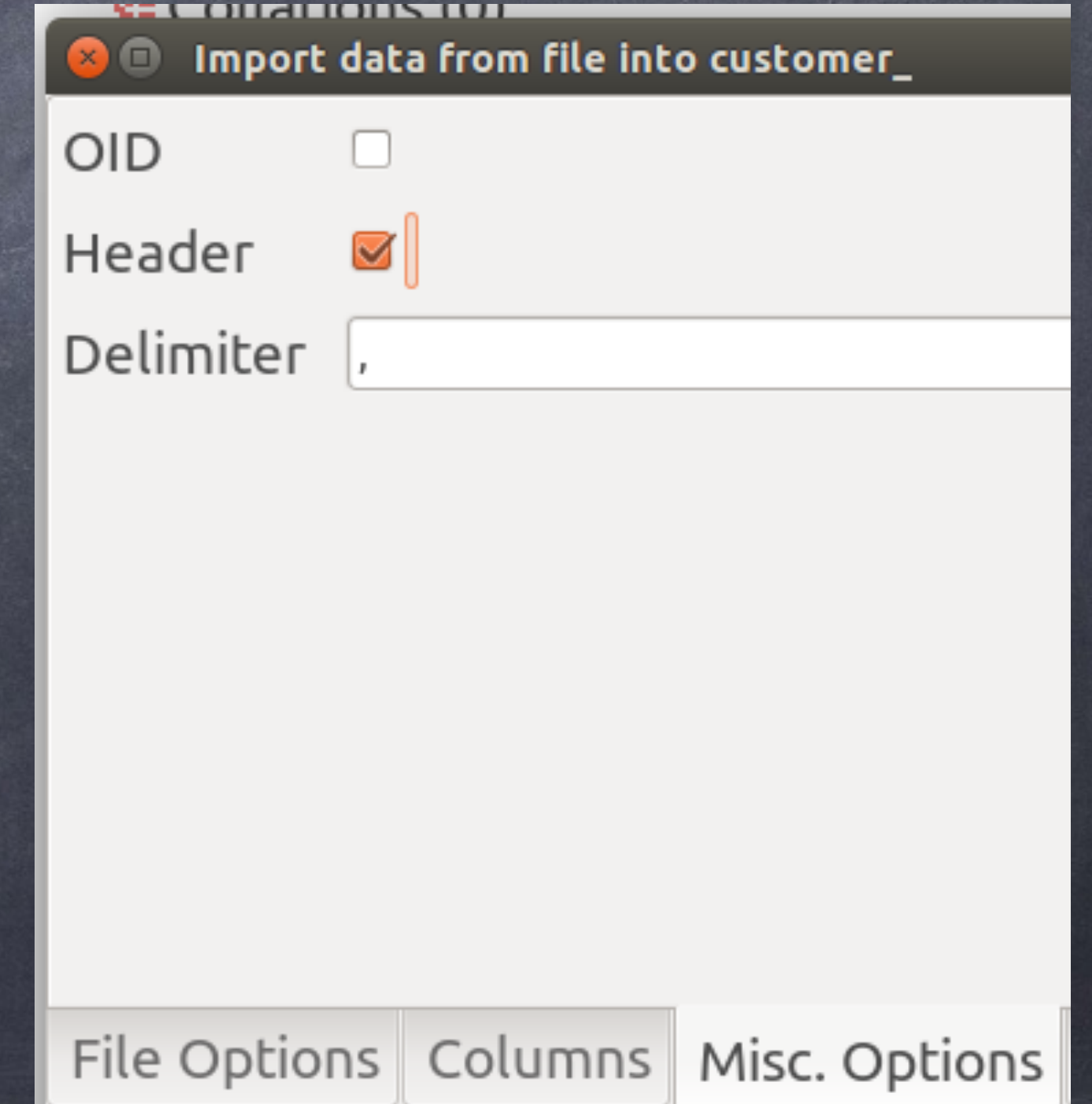
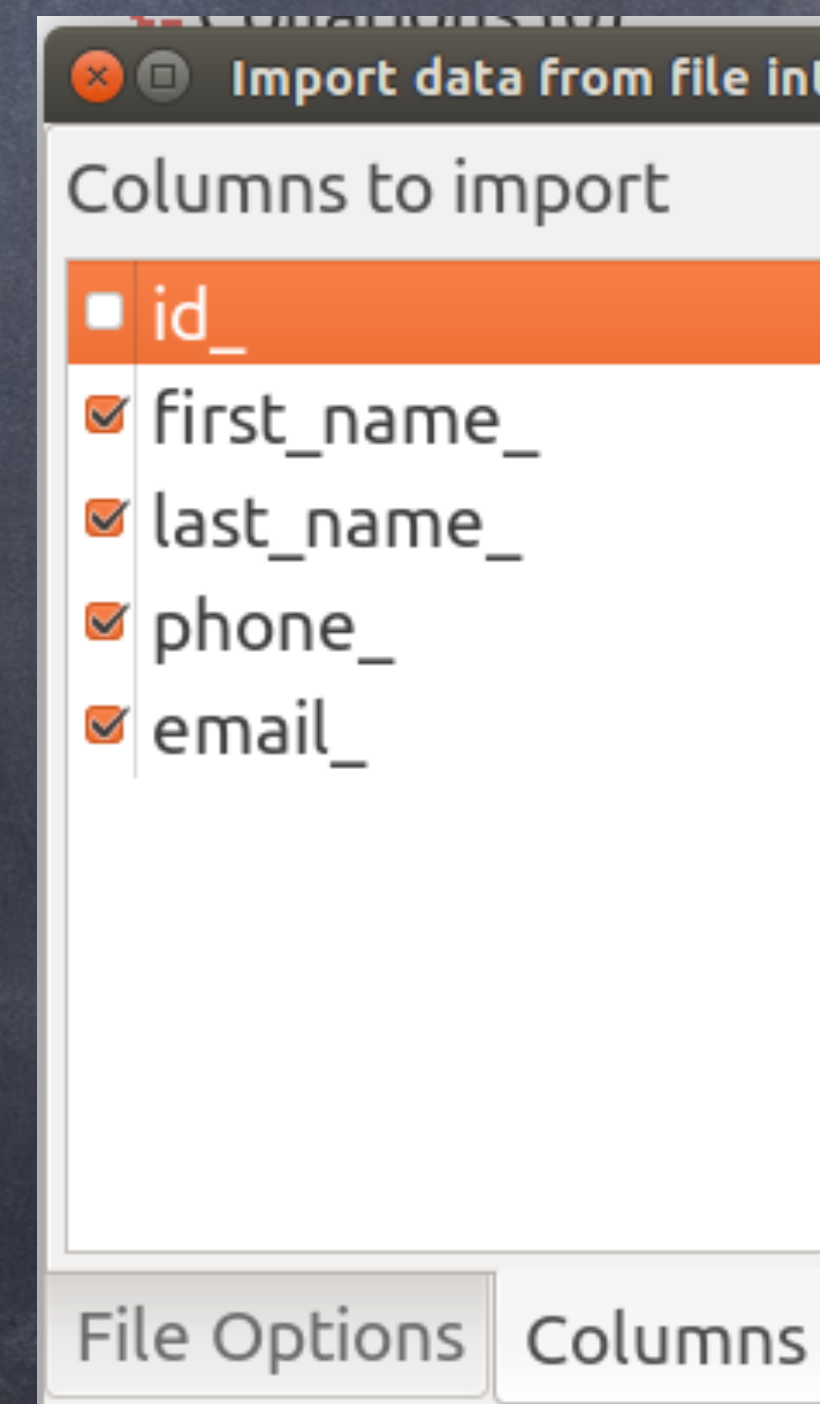
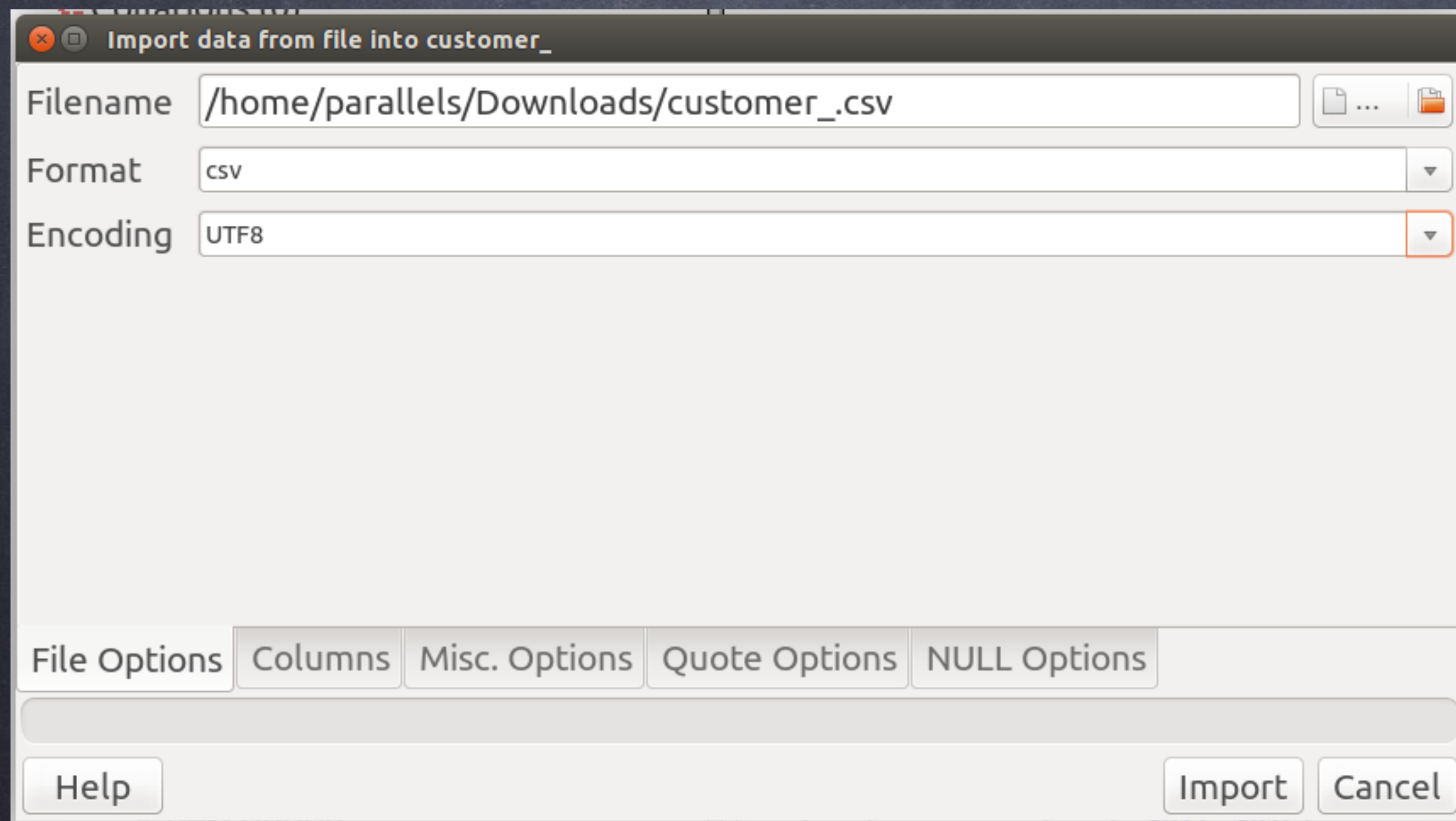
- App
 - Form in an app (fields, buttons)
 - pgAdmin (or other app)
- SQL
 - Write SQL commands that include data
- Import
 - Direct injection into table (no processing)

SQL: Insert command

- `INSERT INTO some_table_ (col1 , col2 , col3 ...)
VALUES (value1 , value2 , value3 , ...) ;`
- `TABLE some_table_ ;
SELECT * FROM some_table_ ;`

Import via 'COPY'

- COPY = mass import from file (not standard)
- Read in (or write out) formats: TSV, CSV, or binary
- Super fast, no SQL to parse and process.



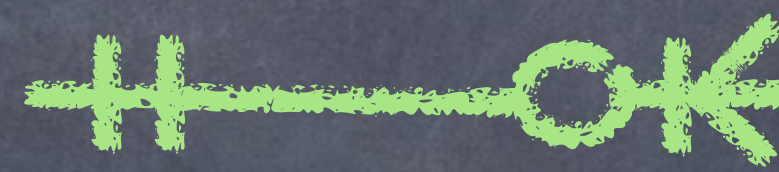
Demo

'customer...'

Child Table

customer_

Attribute	Type
id_	serial (integer)
first_name_	text
last_name_	text
phone_	text
email_	text



pet_

Attribute	Type
id_	serial
fkey_customer_	integer
species_	text
name_	text
description_	text

"ERD" Entity-Relationship Diagram

Foreign Key

- Link child table to parent
- Child knows the identifier value of its parent
- Example:
 - 'invoice_' has ID of 'customer_'
 - 'line_item_' has ID of 'invoice'

customer_
id_
...

invoice_
id_
fkey_cust_
...

line_item_
id_
fkey_inv_
...

Referential Integrity

Orphans?

The image shows a screenshot of a 'New Foreign Key...' dialog box. The dialog has four tabs: 'Properties', 'Definition', 'Columns', and 'Action'. The 'Action' tab is selected. It contains two sections: 'On Update' and 'On Delete'. Each section has a green scribble over the label. Both sections have five radio button options: 'NO ACTION' (selected), 'RESTRICT', 'CASCADE', 'SET NULL', and 'SET DEFAULT'.

New Foreign Key...

Properties Definition Columns Action

On Update

- NO ACTION
- RESTRICT
- CASCADE
- SET NULL
- SET DEFAULT

On Delete

- NO ACTION
- RESTRICT
- CASCADE
- SET NULL
- SET DEFAULT

'pet_' table

Column name	Definition
id_	serial NOT NULL
fkey_customer_	integer NOT NULL
species_	text NOT NULL
name_	text NOT NULL DEFAULT ''
description_	text NOT NULL DEFAULT ''

Constraint name	Definition
pkey_pet_	(id_)
fkey_pet-customer_	(fkey_customer_) REFERENCES public.customer_ (id_) ON UPDATE RESTRICT ON DELETE RESTRICT

```
CREATE TABLE public.pet_  
(  
    id_ serial NOT NULL,  
    fkey_customer_ integer NOT NULL,  
    species_ text NOT NULL,  
    name_ text NOT NULL DEFAULT '',  
    description_ text NOT NULL DEFAULT '',  
    CONSTRAINT pkey_pet_ PRIMARY KEY (id_),  
    CONSTRAINT "fkey_pet-customer_" FOREIGN KEY (fkey_customer_) REFERENCES public.customer_ (id_) ON UPDATE RESTRICT ON DELETE RESTRICT  
)  
WITH (
```


Tables in pgAdmin

- [-] Tables (2)
 - [-] customer_~~customer_~~
 - [-] Columns (5)
 - id_
 - first_name_
 - last_name_
 - phone_
 - email_
 - [-] Constraints (1)
 - pkey_customer_
 - [-] Indexes (0)
 - [-] Rules (0)
 - [-] Triggers (0)

- [-] pet_~~pet_~~
 - [-] Columns (5)
 - id_
 - fkey_customer_
 - species_
 - name_
 - description_
 - [-] Constraints (2)
 - pkey_pet_
 - fkey_pet-customer_ -> customer_
 - [-] Indexes (0)
 - [-] Rules (0)
 - [-] Triggers (0)

Resources

- Postgres own docs - best in the biz
 - <http://www.PostgreSQL.org/docs/> ("current" link)
- StackOverflow.com & dba.StackExchange.com
- AskUbuntu.com & unix.StackExchange.com
- Mailing lists... <http://www.PostgreSQL.org/list/>
- Blog aggregator: <http://planet.PostgreSQL.org/>