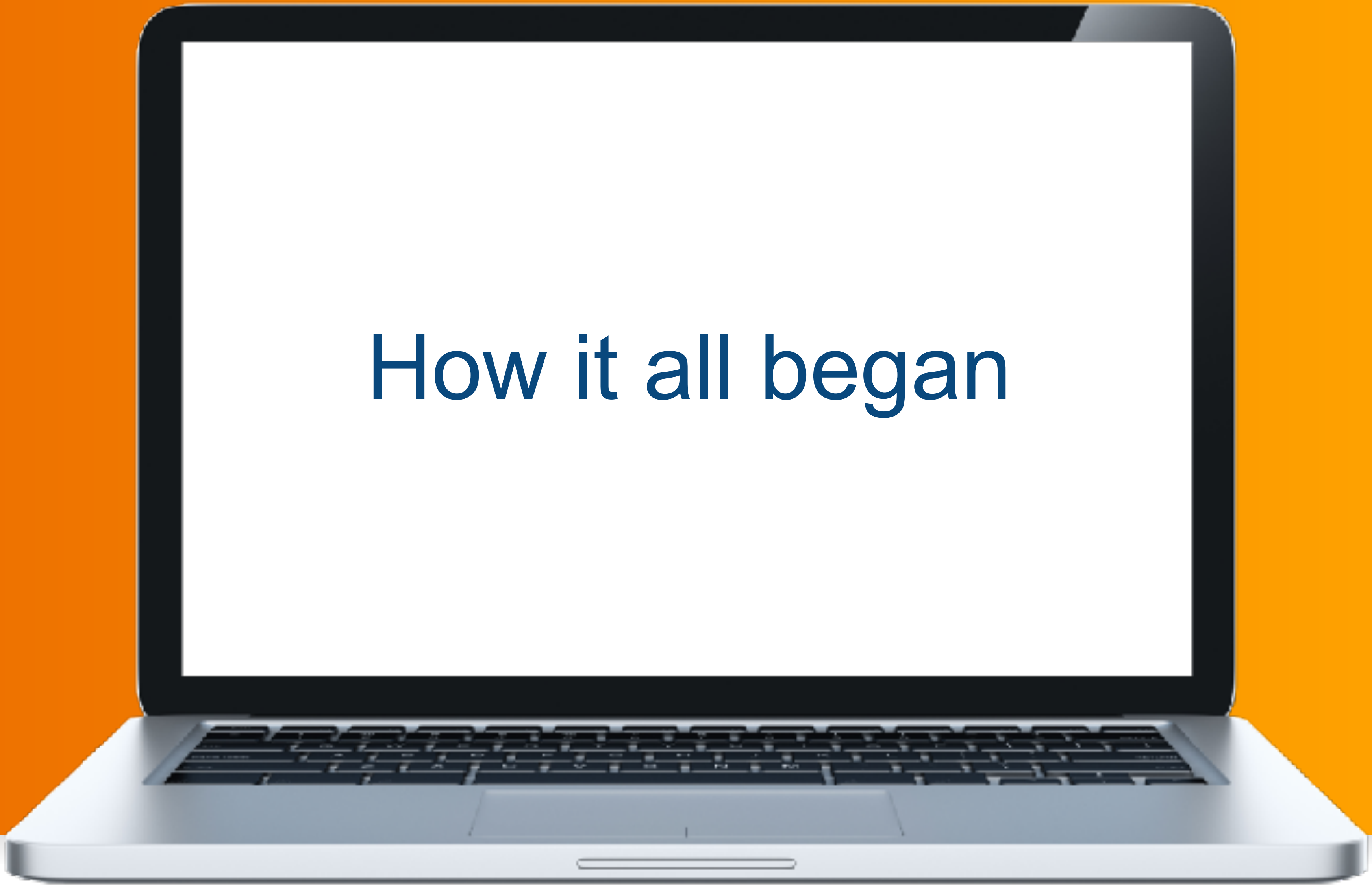
pgAdmin: How the project works

Dave Page

Introduction

- Dave Page
- @pgsnake
- dpage@pgadmin.org

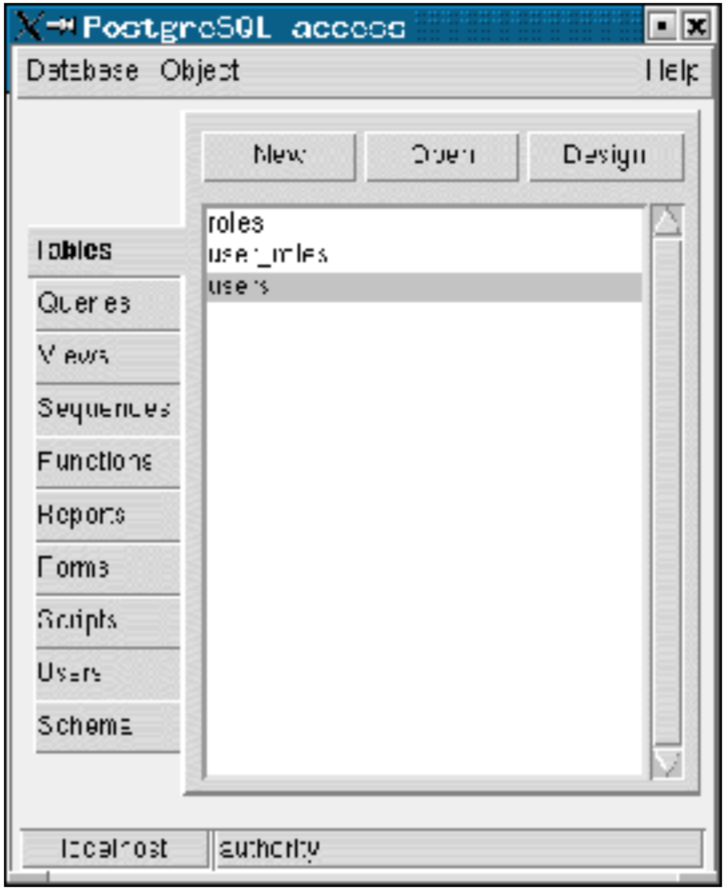
- PostgreSQL:
  - Core team member
  - pgAdmin lead developer
  - Web/sysadmin teams
  - PGCAC/PGEU board member
- EDB:
  - VP & Chief Architect:
    - Tools
    - Cloud products
    - Configuration management
    - Packaging/distribution



How it all began

In the beginning…

- Well, 1996ish
- The only GUI admin tool for PostgreSQL was pgAccess
- Written using TCL/TK
- Hard to install on Windows
- Urgh. Just urgh.



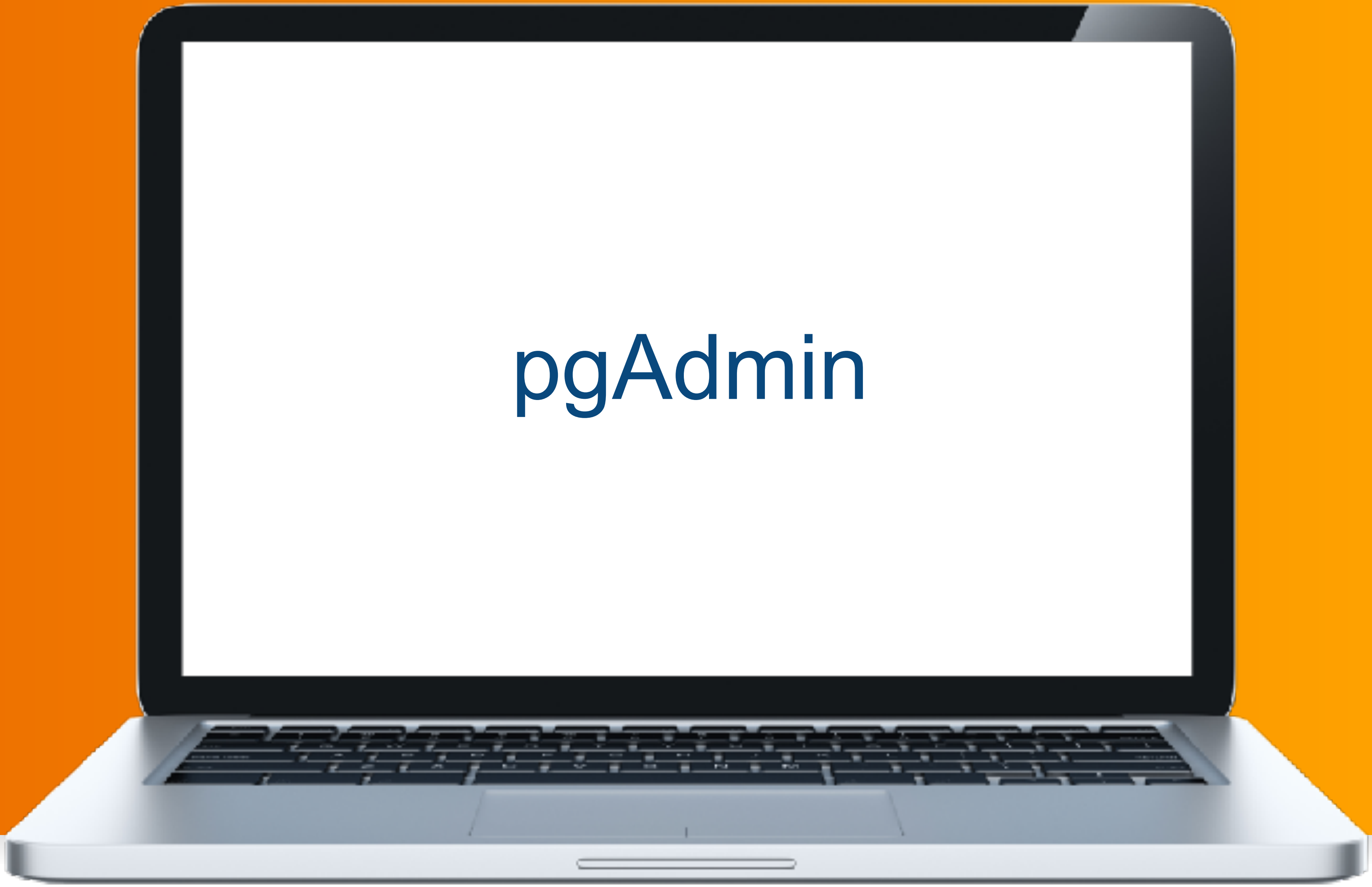
The solution

- Write my own!
- Hey, this is Open Source right?
- How hard can it be?
- This won't take much of my time!

pgManager

- Circa 1997
- Prototype in VB4
- Called pgManager
- Looked just like pgAccess (urgh)
- Never released

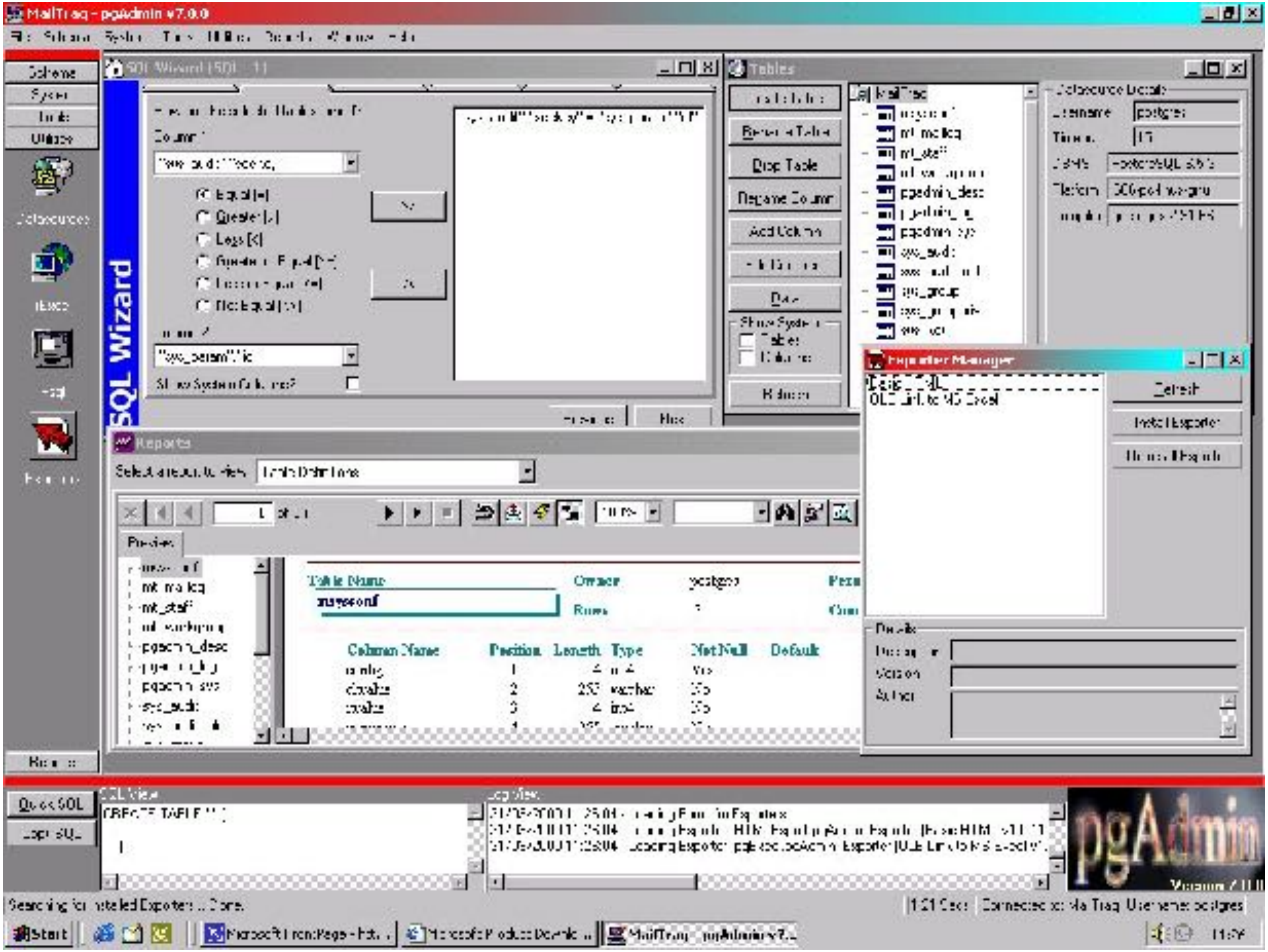
(thank $DEITY)



pgAdmin

pgAdmin

- Circa 1997/98
- Written in Visual Basic 5
- (and therefore only supported Windows)
- Very little UI consistency
- Terrible marketing



MailTraq - pgAdmin v7.0.0

SQL Wizard

Reports

Table Name	Owner	postgres
[illegible]	Rows	[illegible]

Column Name	Position	Length	Type	Not Null	Default
[illegible]	1	4	int4	Yes	
[illegible]	2	255	varchar	No	
[illegible]	3	4	int4	No	

SQL View

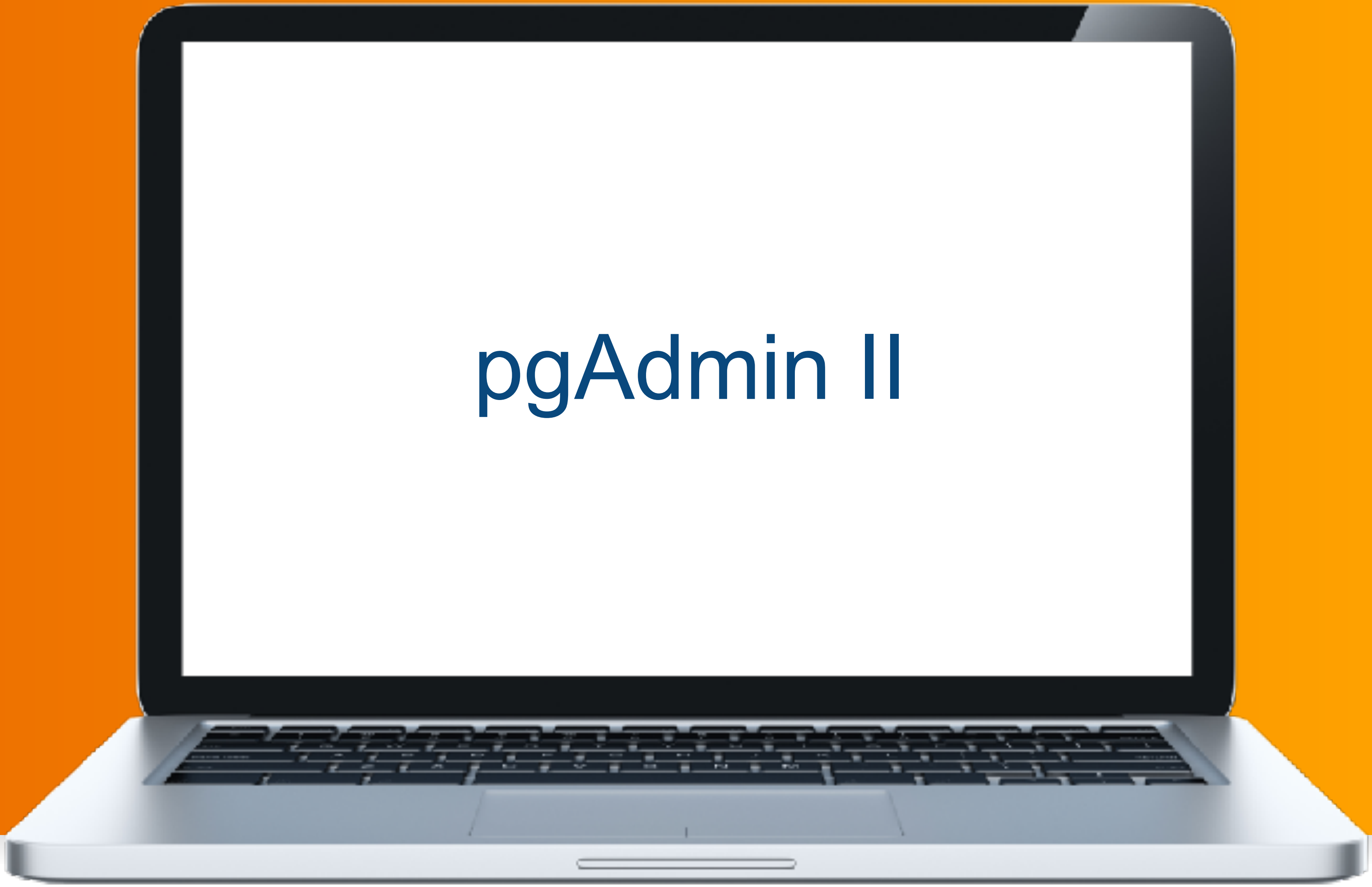
```
CREATE TABLE [illegible]
```

Searching for installed Exporters ... Done.

Connected to: MailTraq Username: postgres

Development

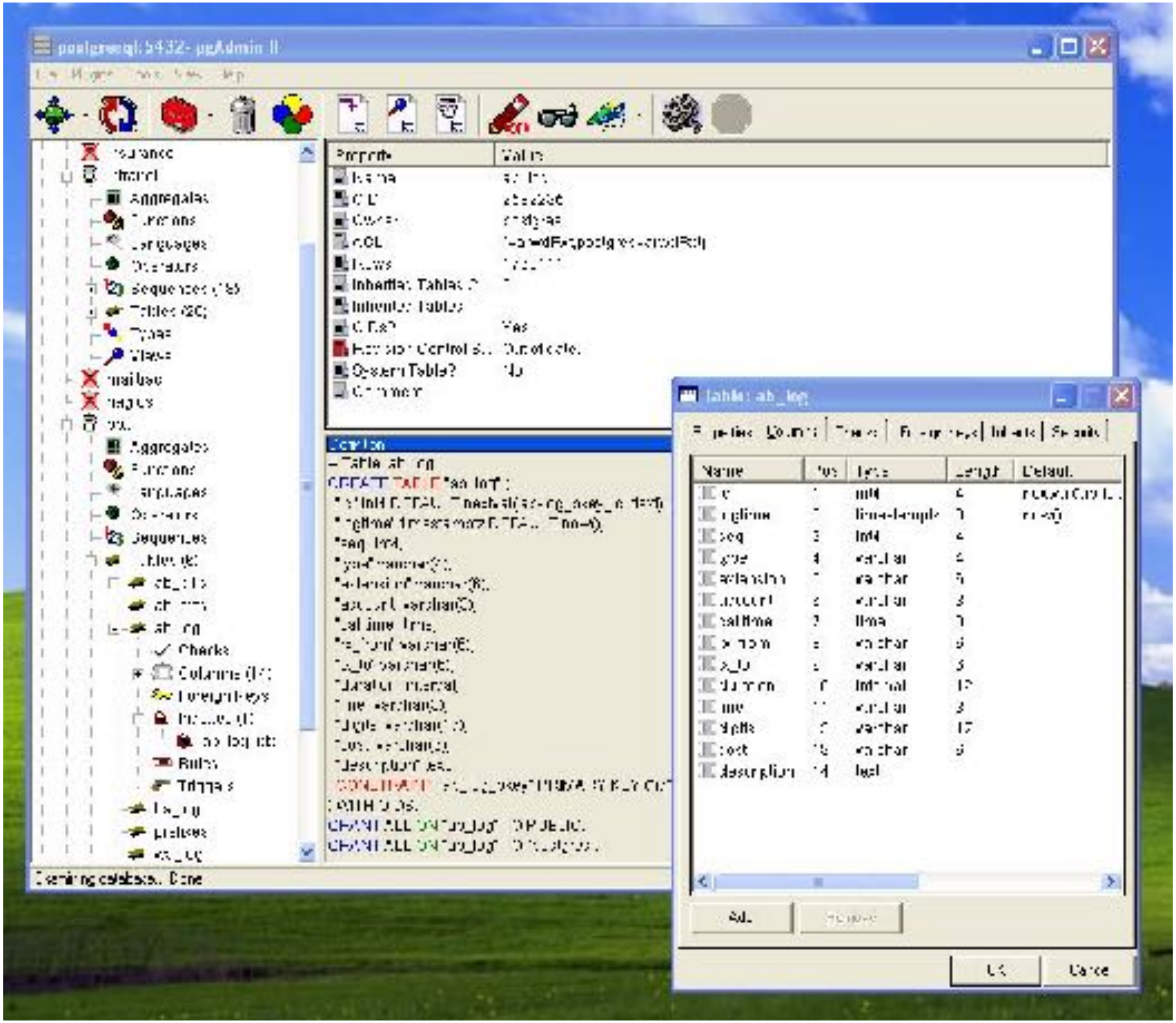
- Source code not released initially
- Virtually all work done in private
- Eventually released the code
- Attracted a small number of contributors, most notably:
  - Jean-Michel POURE
  - Hiroshi Saito
- No processes of note; patches were submitted and applied with only basic thought to maintainability



pgAdmin II

pgAdmin II

- Circa 2000
- Written in VB6
- 😀 Clean-room implementation
- 😀 Tightly controlled
- 😀 Well structured code
- 😕 Windows only
- 😕 No Localisation/Unicode



Property	Value
Name	ab_log
OID	[illegible]
Owner	[illegible]
ACL	[illegible]
Rows	[illegible]
Inherited Tables?	[illegible]
Inherited Tables	
Has OIDs?	Yes
Revision Control Status	Out of date
System Table?	No
Comment	

```
-- Table ab_log
CREATE TABLE "ab_log" (
 [illegible]
 [illegible]
 "seq" int4,
 [illegible]
 [illegible]
 "account" varchar(3),
 "calltime" time,
 [illegible]
 [illegible]
 "duration" interval,
 "line" varchar(3),
 "digits" varchar(12),
 "cost" varchar(6),
 "description" text,
 CONSTRAINT [illegible] PRIMARY KEY [illegible]
) WITH OIDS;
GRANT ALL ON "ab_log" TO PUBLIC;
GRANT ALL ON "ab_log" TO [illegible];
```

Name	Pos	Type	Length	Default
id	1	int4	4	nextval('[illegible]
logtime	2	timestamp	8	now()
seq	3	int4	4	
type	4	varchar	4	
extension	5	varchar	5	
account	6	varchar	3	
calltime	7	time	8	
[illegible]	8	varchar	6	
[illegible]	9	varchar	3	
duration	10	interval	12	
line	11	varchar	3	
digits	12	varchar	12	
cost	13	varchar	6	
description	14	text		

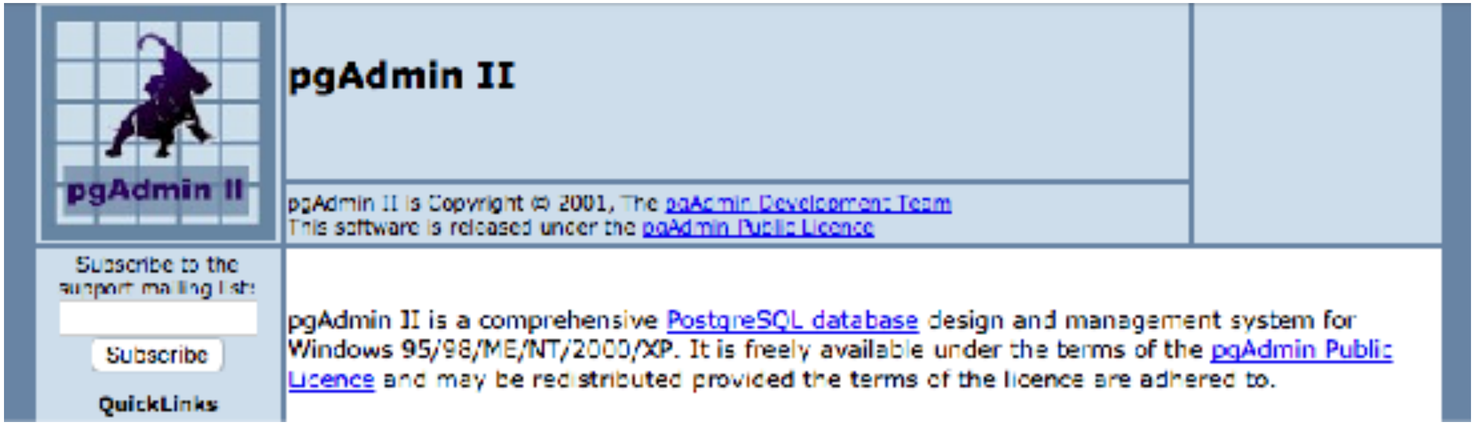
Opening database... Done

Development

- Source code released once the first version was ready
- Patches continued to come in from previous contributors
- Still little thought given to process and project organisation
- Releases were ad-hoc, sometimes roughly coordinated with PostgreSQL

Online

- PHP website
- Hard to maintain
- Ugly, fixed-width design



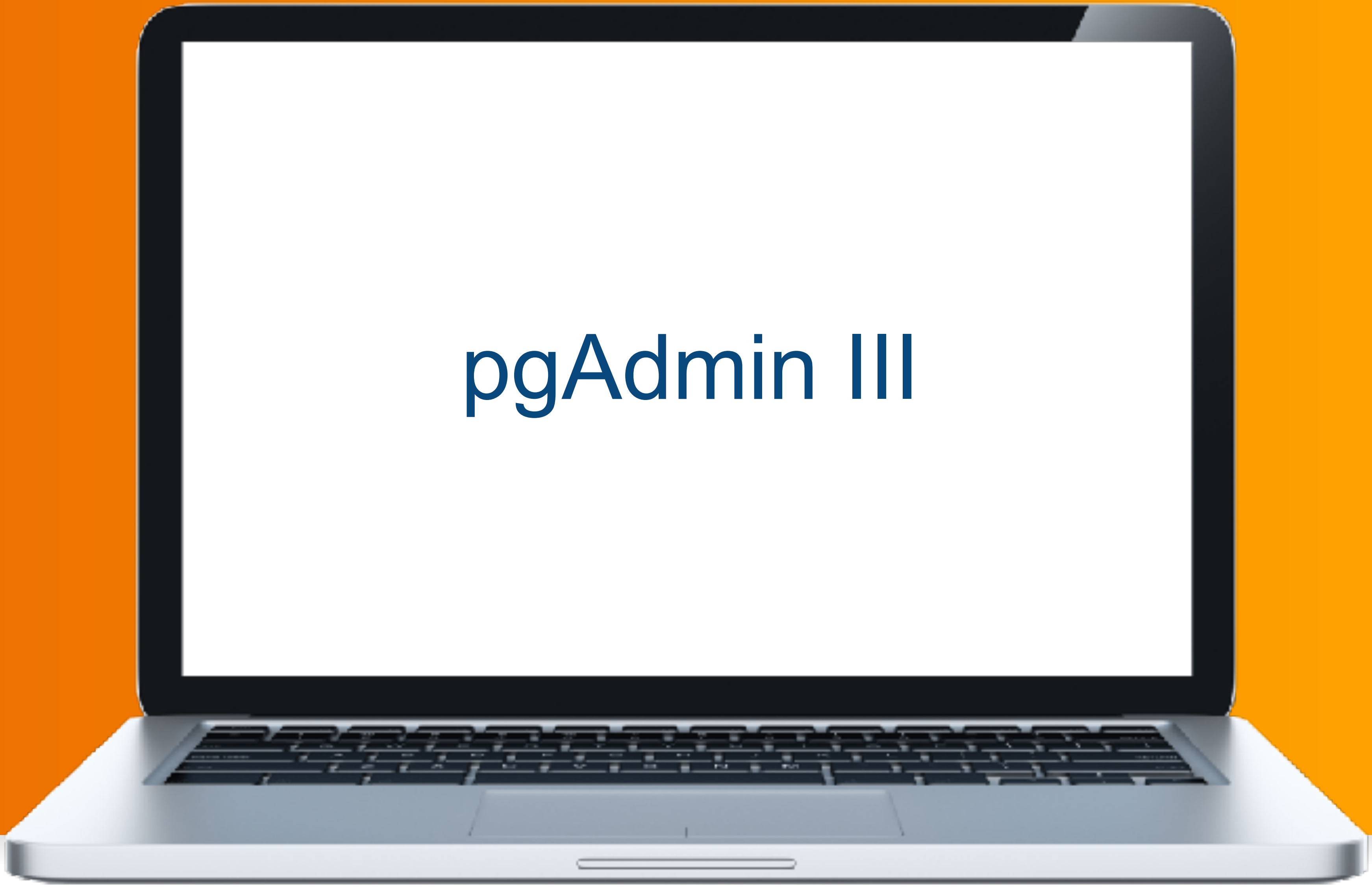
pgAdmin II

pgAdmin II is Copyright (c) 2001, The pgAdmin Development Team
This software is released under the pgAdmin Public Licence

Subscribe to the support mailing list:

QuickLinks

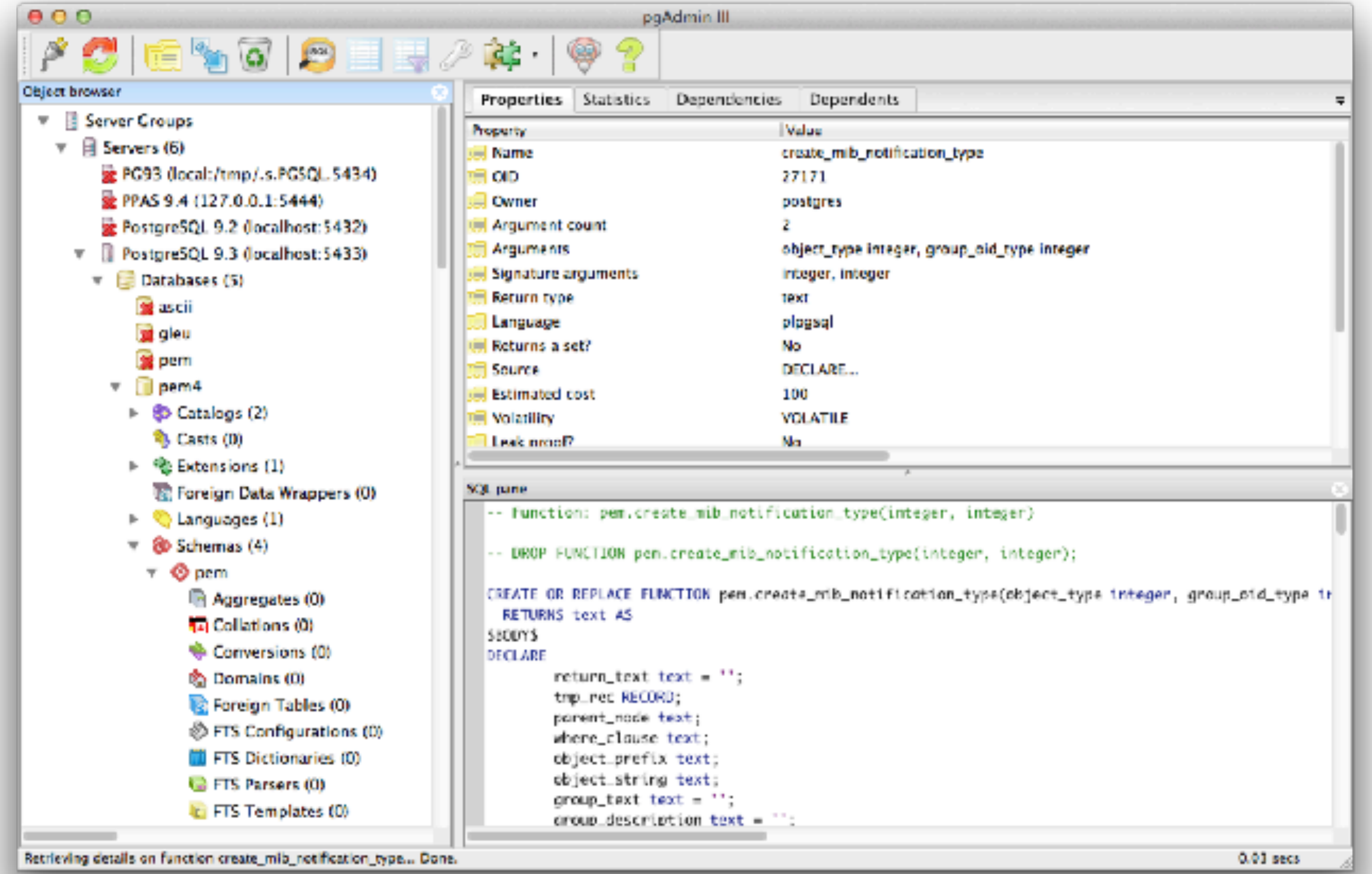
pgAdmin II is a comprehensive PostgreSQL database design and management system for Windows 95/98/ME/NT/2000/XP. It is freely available under the terms of the pgAdmin Public Licence and may be redistributed provided the terms of the licence are adhered to.



pgAdmin III

pgAdmin III

- 2002 - 2016
- 😀 Written in C++/wxWidgets (Multiplatform)
- 😀 Supports Localisation & Unicode
- 😕 Limited C++ developer pool
- 😕 Poor support from wxwidgets.org



Property	Value
Name	create_mib_notification_type
OID	27171
Owner	postgres
Argument count	2
Arguments	object_type integer, group_oid_type integer
Signature arguments	integer, integer
Return type	text
Language	plpgsql
Returns a set?	No
Source	DECLARE...
Estimated cost	100
Volatility	VOLATILE
Leak proof?	No

```
-- Function: pem.create_mib_notification_type(integer, integer)

-- DROP FUNCTION pem.create_mib_notification_type(integer, integer);

CREATE OR REPLACE FUNCTION pem.create_mib_notification_type(object_type integer, group_oid_type in
  RETURNS text AS
$BODY$
DECLARE
        return_text text = '';
        tmp_rec RECORD;
        parent_node text;
        where_clause text;
        object_prefix text;
        object_string text;
        group_text text = '';
        group_description text = '';
```

Retrieving details on function create_mib_notification_type... Done.　　0.03 secs

Development

- Source code public from the first commit
- pgadmin-hackers@postgresql.org mailing list used for development coordination
- More stringent requirements on contributors:
  - New features should be discussed to gain agreement on direction before work begins
  - Documentation updates expected where appropriate
  - Contributors expected to fix bugs in their code

Testing

- Reliant primarily on manual community testing of beta releases
- Completely ad-hoc process, with no guarantee of success

Translations

- Translation team organised by Guillaume Lelarge
- Translations updated prior to releases for greatest efficiency
- Translators (and users) could track translation status on the project website
- Over 20 translations produced

Releases

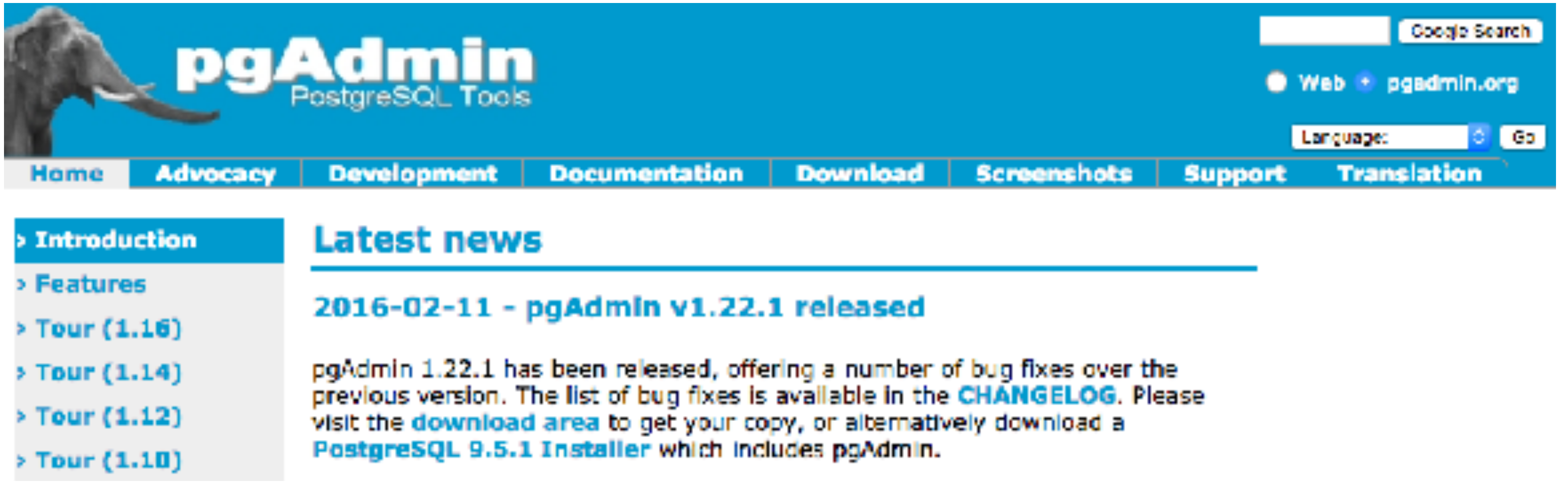
- Major and minor releases
- Code branched for each major release
- Major releases synchronised with new releases of PostgreSQL (to ship with the EDB PostgreSQL installers)
- Each branch maintained until the release of the next major version

Packages

- Source tarballs produced for each release
- Windows and macOS packages provided by the project
- RPMs, DEBs and other packages provided independently of the project by various volunteers

Online

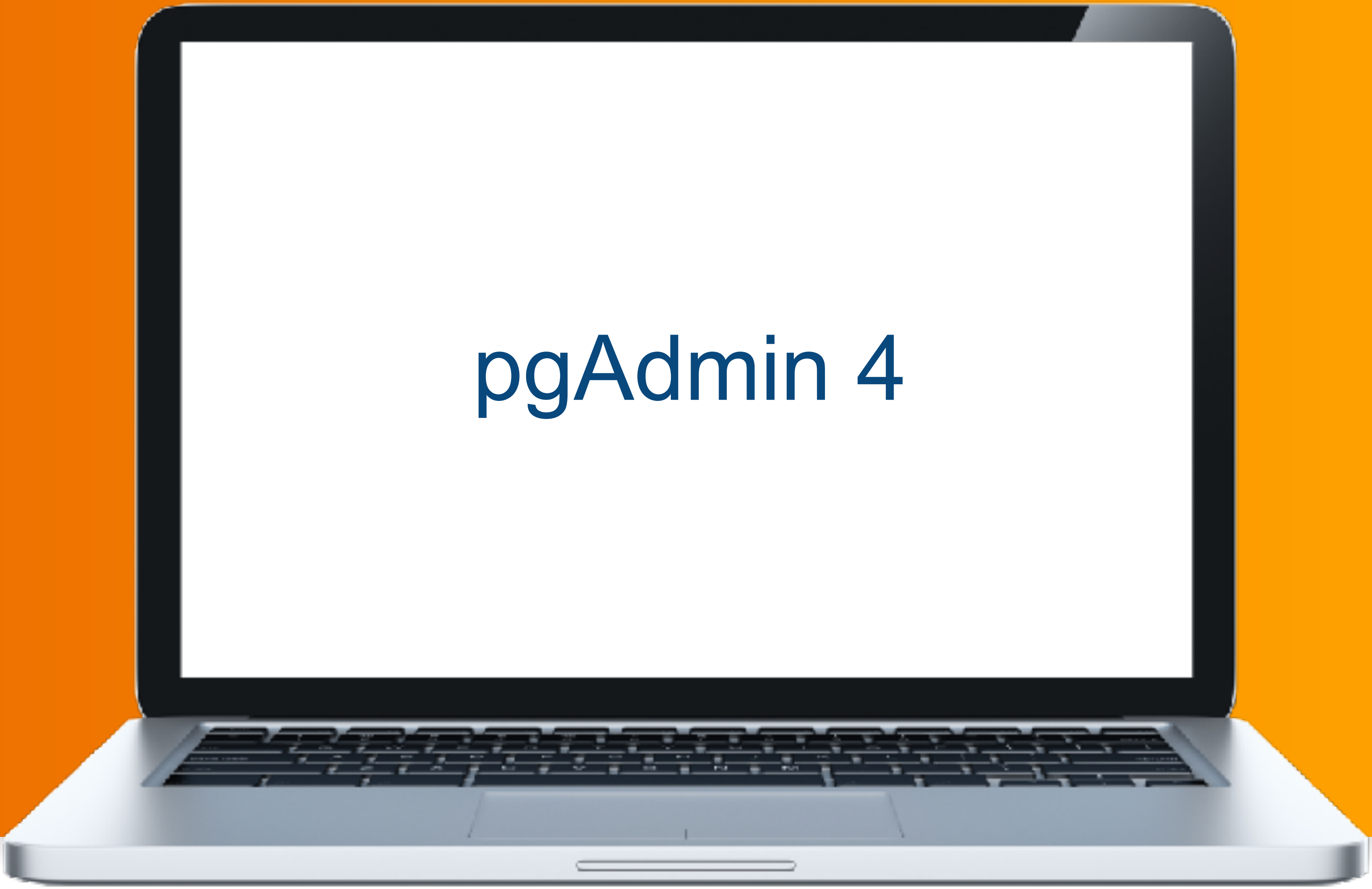
- A whole new PHP based website!
- Still hard to maintain!
- Still ugly!



Latest news

2016-02-11 - pgAdmin v1.22.1 released

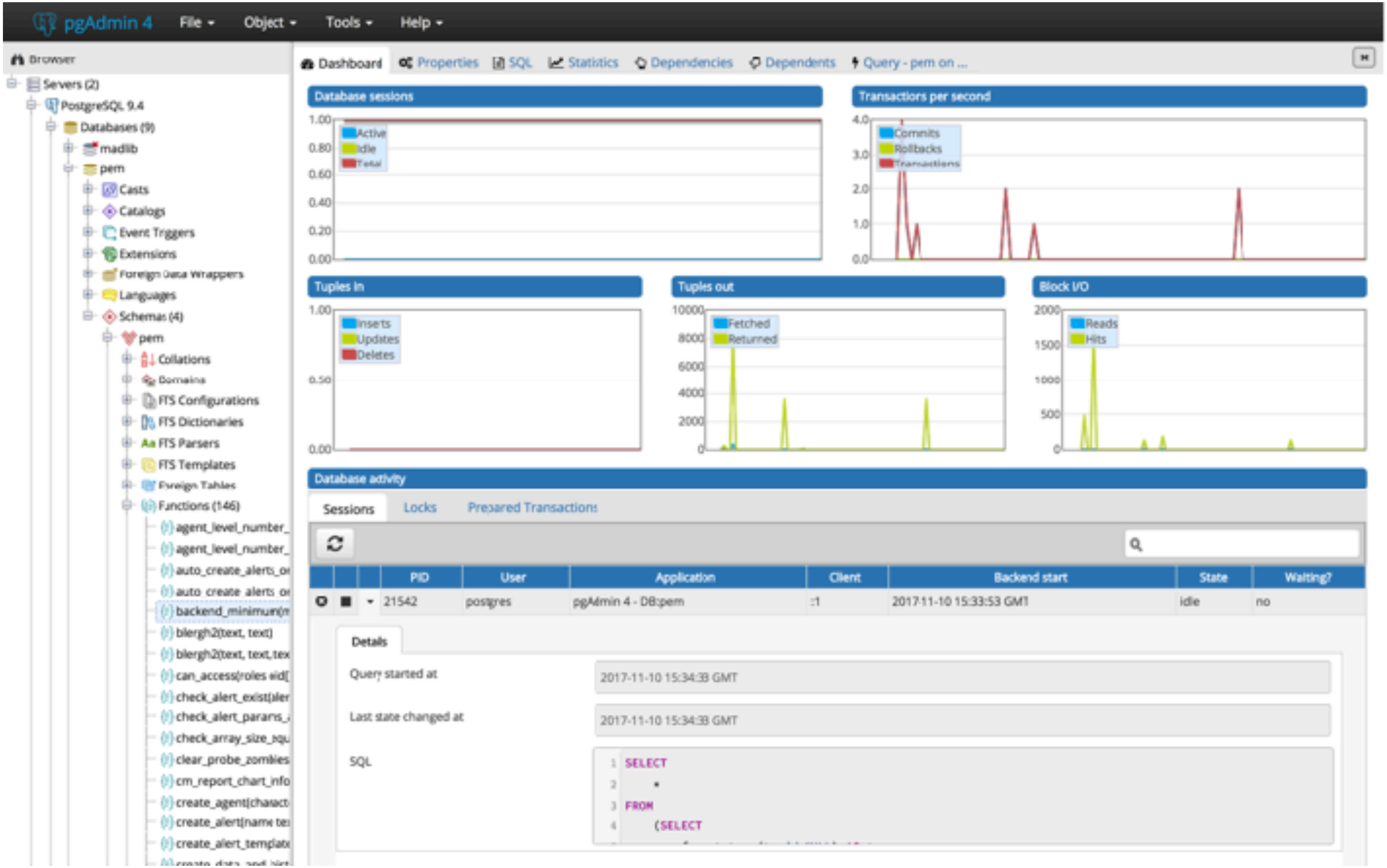
pgAdmin 1.22.1 has been released, offering a number of bug fixes over the previous version. The list of bug fixes is available in the CHANGELOG. Please visit the download area to get your copy, or alternatively download a PostgreSQL 9.5.1 Installer which includes pgAdmin.



pgAdmin 4

pgAdmin 4

- 2016 - ?
- Written in Python with a little C++/Qt
- Flask + jQuery + Backbone.js + Bootstrap + React
- Runs on web server or desktop
- Contributions from EDB, Pivotal, Dalibo, Postgres Pro, OpenSCG and more



			PID	User	Application	Client	Backend start	State	Waiting?
●	■	▾	21542	postgres	pgAdmin 4 - DB:pem	::1	2017-11-10 15:33:53 GMT	idle	no

Query started at: 2017-11-10 15:34:38 GMT

Last state changed at: 2017-11-10 15:34:38 GMT

SQL:

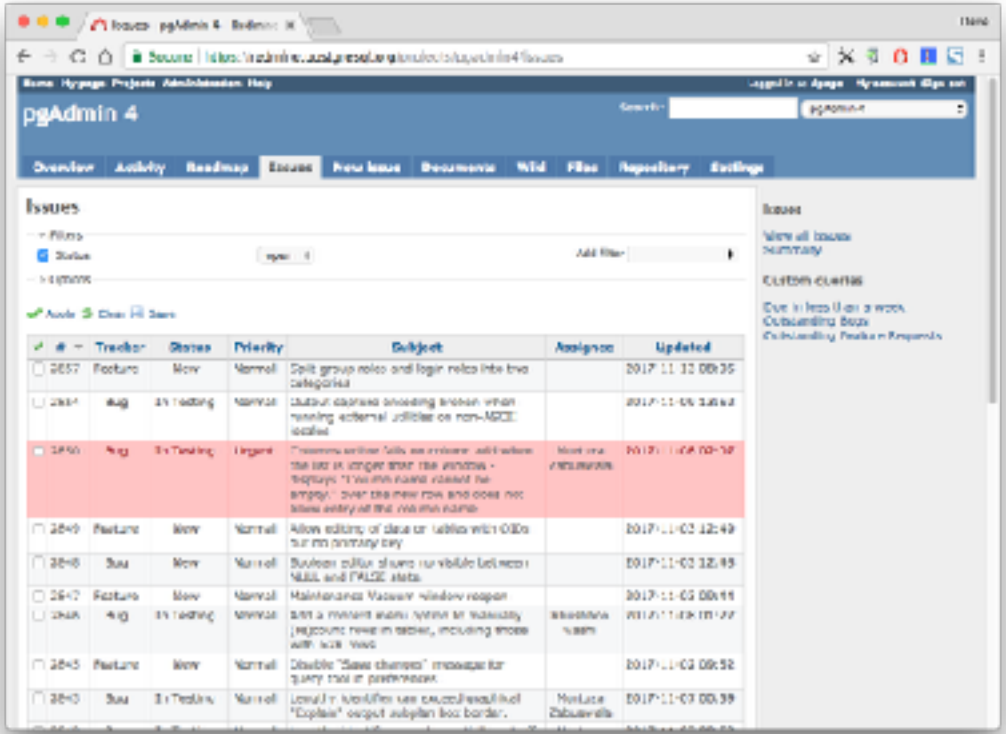
```
SELECT
    *
FROM
    (SELECT
```

Development

- `#include <pgadmin3_development_practices.h>`
- Bug/feature tracking
- More stringent requirements on contributors:
  - Coding standards
  - Style guide
  - New features should include tests
- Automation
- Agile

Bug/feature tracking

- redmine.postgresql.org used to track bugs and feature requests
- No more reliance on email or ad-hoc lists etc.
- Workflow includes QA/ verification phase
- Fixes and changes tracked against release versions

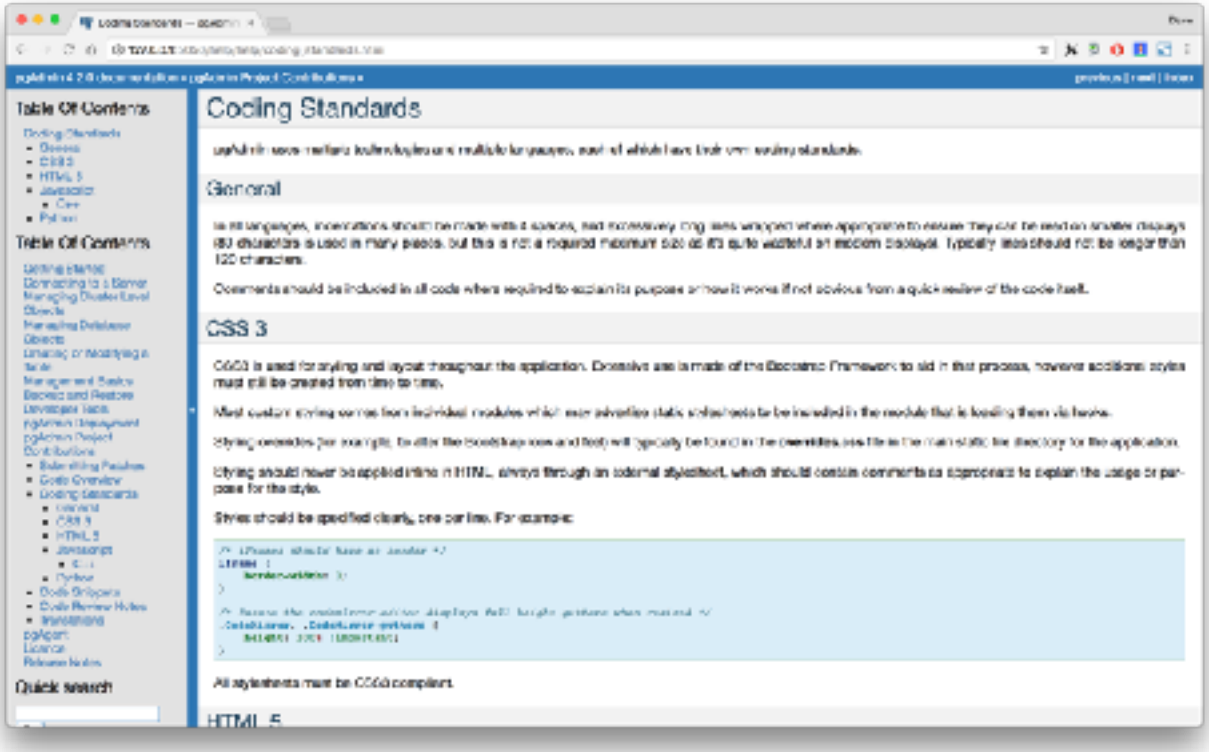


Issues

#	Tracker	Status	Priority	Subject	Assignee	Updated
2857	Feature	New	Normal	Split group roles and login roles into two categories		2017-11-12 08:35
2854	Bug	In Testing	Normal	Output charset encoding broken when running external utilities on non-ASCII locales		2017-11-06 [illegible]
2850	Bug	In Testing	Urgent	[illegible] the list is longer than the window - displays "Trigger/column name cannot be empty." over the new row and does not allow entry of the column name	[illegible]	2017-11-06 [illegible]
2849	Feature	New	Normal	Allow editing of data on tables with OIDs but no primary key		2017-11-03 12:49
2848	Bug	New	Normal	Boolean editor shows no visible difference between NULL and FALSE state		2017-11-03 12:45
2847	Feature	New	Normal	Maintenance Vacuum window reopen		2017-11-03 08:44
2846	Bug	In Testing	Normal	[illegible] rows in tables, including those with OIDs	Khushboo Vashi	2017-11-03 [illegible]
2845	Feature	New	Normal	Disable "Save changes" message for query tool in preferences.		2017-11-03 08:52
2843	Bug	In Testing	Normal	[illegible] "Explain" output subplan box border.	Murtuza Zabuawala	2017-11-03 08:39

Coding standards

- Documented coding standards for the various languages in use within the product
- Intended to ensure code remains consistent in both style and design for future maintainability



pgAdmin 4 2.0 documentation » pgAdmin Project Contributions » previous | next | index

Table Of Contents

- Coding Standards
  - General
  - CSS 3
  - HTML 5
  - Javascript
    - C++
  - Python

Table Of Contents

Getting started
Connecting to a Server
Managing Cluster Level Objects
Managing Database Objects
Creating or Modifying a Table
Management Basics
Backup and Restore
Developer Tools
pgAdmin Deployment
pgAdmin Project Contributions
- Submitting Patches
- Code Overview
- Coding Standards
  - General
  - CSS 3
  - HTML 5
  - Javascript
    - C++
  - Python
- Code Snippets
- Code Review Notes
- Translations

pgAgent
Licence
Release Notes

Quick search

Coding Standards

pgAdmin uses multiple technologies and multiple languages, each of which have their own coding standards.

General

In all languages, indentations should be made with 4 spaces, and excessively long lines wrapped where appropriate to ensure they can be read on smaller displays (80 characters is used in many places, but this is not a required maximum size as it's quite wasteful on modern displays). Typically lines should not be longer than 120 characters.

Comments should be included in all code where required to explain its purpose or how it works if not obvious from a quick review of the code itself.

CSS 3

CSS3 is used for styling and layout throughout the application. Extensive use is made of the Bootstrap Framework to aid in that process, however additional styles must still be created from time to time.

Most custom styling comes from individual modules which may advertise static stylesheets to be included in the module that is loading them via hooks.

Styling overrides (for example, to alter the Bootstrap look and feel) will typically be found in the overrides.css file in the main static file directory for the application.

Styling should never be applied inline in HTML, always through an external stylesheet, which should contain comments as appropriate to explain the usage or purpose for the style.

Styles should be specified clearly, one per line. For example:

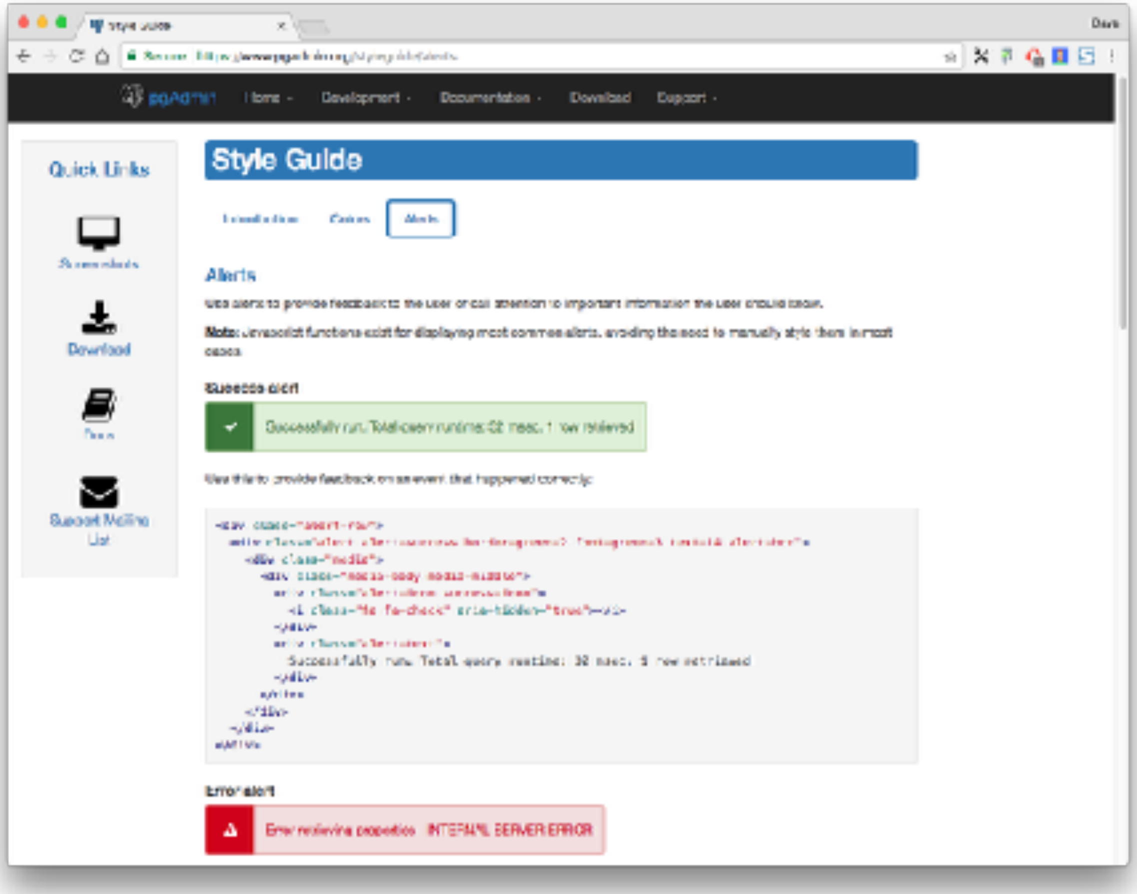
```
/* iFrames should have no border */
iframe {
    border-width: 0;
}

/* Ensure the codemirror editor displays full height gutters when resized */
.CodeMirror, .CodeMirror-gutters {
    height: 100% !important;
}
```

All stylesheets must be CSS3 compliant.

HTML 5

Style guide

- The style guide defines standard colours, fonts, message presentation and other elements of the UI design
- Intended to ensure the UI has a consistent look and feel throughout to help give the optimal user experience



Style Guide

Alerts

Use alerts to provide feedback to the user or call attention to important information the user should see.

Note: Javascript functions exist for displaying most common alerts, avoiding the need to manually style them in most cases.

Success alert

✓ Successfully run. Total query runtime: 32 msec. 1 row retrieved

Use this to provide feedback on an event that happened correctly.

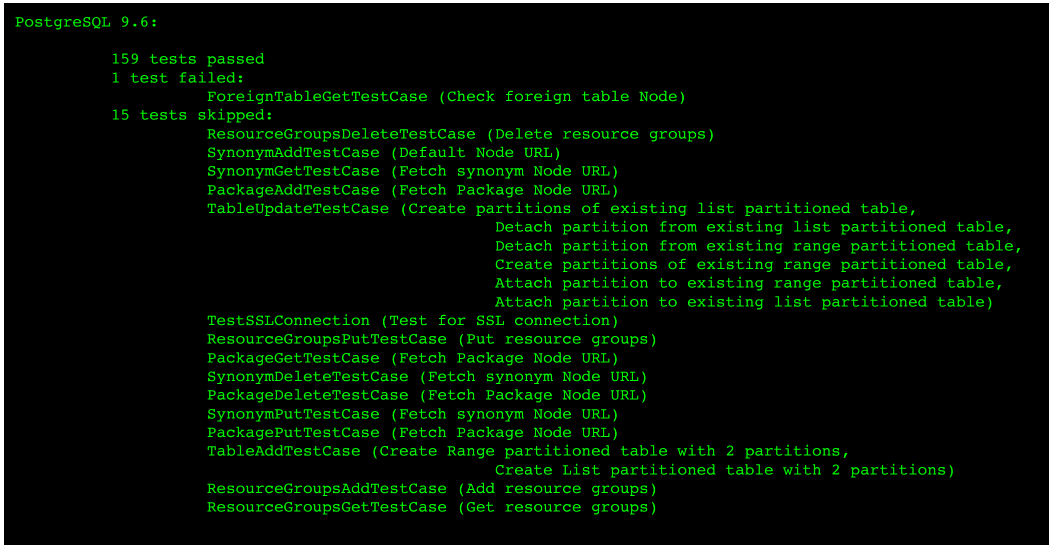
```
<div class="alert-row">
  <div class="alert alert-success [illegible]">
    <div class="media">
      <div class="media-body media-middle">
        <div class="alert-icon success-icon">
          <i class="fa fa-check" aria-hidden="true"></i>
        </div>
        <div class="alert-text">
          Successfully run. Total query runtime: 32 msec. 1 row retrieved
        </div>
      </div>
    </div>
  </div>
</div>
```

Error alert

⚠ Error retrieving properties - INTERNAL SERVER ERROR

Testing

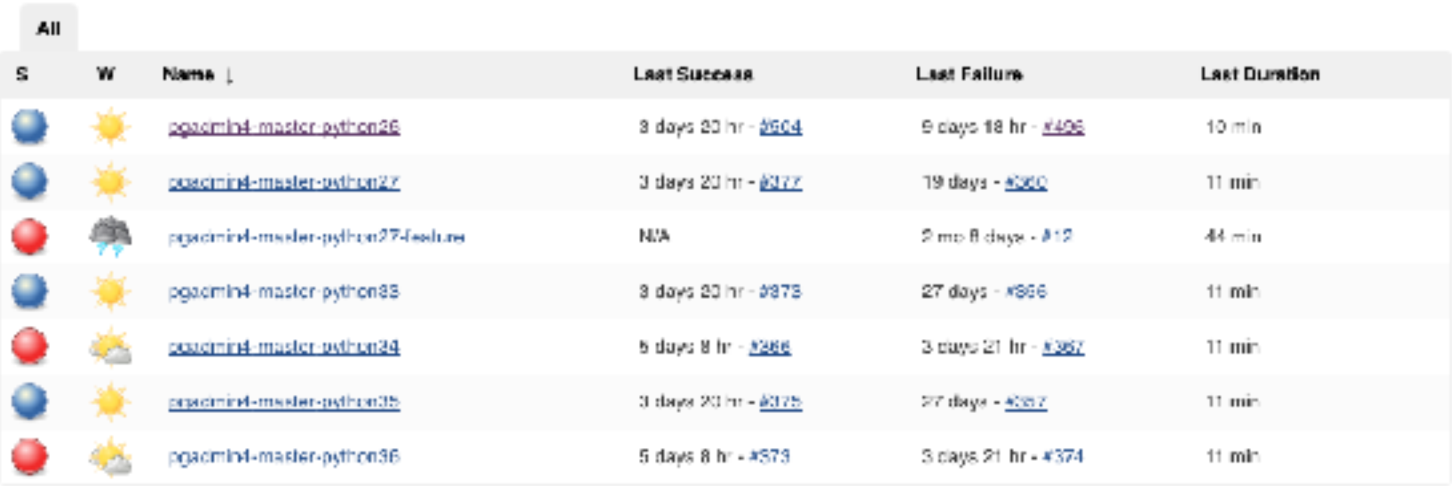
- Ad-hoc community testing
- Manual bug/feature verification by EDB’s QA Team
- Multiple automated test suites:
  - Jasmine unit tests for Javascript code
  - API tests for Python REST APIs
  - Selenium-based feature tests for end-to-end verification



```
PostgreSQL 9.6:

          159 tests passed
          1 test failed:
                    ForeignTableGetTestCase (Check foreign table Node)
          15 tests skipped:
                    ResourceGroupsDeleteTestCase (Delete resource groups)
                    SynonymAddTestCase (Default Node URL)
                    SynonymGetTestCase (Fetch synonym Node URL)
                    PackageAddTestCase (Fetch Package Node URL)
                    TableUpdateTestCase (Create partitions of existing list partitioned table,
                                                  Detach partition from existing list partitioned table,
                                                  Detach partition from existing range partitioned table,
                                                  Create partitions of existing range partitioned table,
                                                  Attach partition to existing range partitioned table,
                                                  Attach partition to existing list partitioned table)
                    TestSSLConnection (Test for SSL connection)
                    ResourceGroupsPutTestCase (Put resource groups)
                    PackageGetTestCase (Fetch Package Node URL)
                    SynonymDeleteTestCase (Fetch synonym Node URL)
                    PackageDeleteTestCase (Fetch Package Node URL)
                    SynonymPutTestCase (Fetch synonym Node URL)
                    PackagePutTestCase (Fetch Package Node URL)
                    TableAddTestCase (Create Range partitioned table with 2 partitions,
                                                  Create List partitioned table with 2 partitions)
                    ResourceGroupsAddTestCase (Add resource groups)
                    ResourceGroupsGetTestCase (Get resource groups)
```

Automation

- Jenkins-based Continuous Integration automatically runs builds and tests against different Postgres versions under each supported Python version.
- Production build packages/tarballs etc. are in the process of being moved to dedicated automated build systems

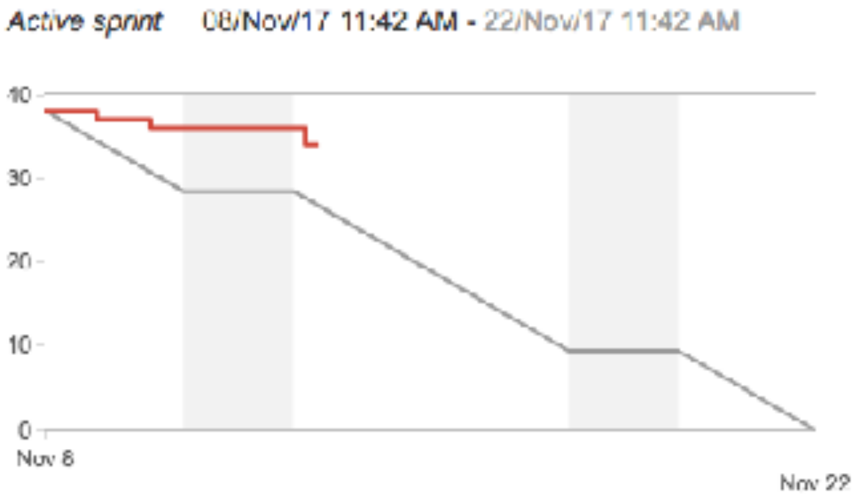


All

S	W	Name ↓	Last Success	Last Failure	Last Duration
🔵	☀	pgadmin4-master-python26	3 days 20 hr - #504	9 days 18 hr - #496	10 min
🔵	☀	pgadmin4-master-python27	3 days 20 hr - #377	19 days - #360	11 min
🔴	⛈	pgadmin4-master-python27-feature	N/A	2 mo 8 days - #12	44 min
🔵	☀	pgadmin4-master-python33	3 days 20 hr - #378	27 days - #366	11 min
🔴	⛅	pgadmin4-master-python34	5 days 8 hr - #368	3 days 21 hr - #367	11 min
🔵	☀	pgadmin4-master-python35	3 days 20 hr - #375	27 days - #367	11 min
🔴	⛅	pgadmin4-master-python36	5 days 8 hr - #373	3 days 21 hr - #374	11 min

Agile

- Automation and CI are key to agile methodologies
- The community as a whole cannot follow defined processes such as SCRUM - it's hard to herd cats!



- However, teams at both EDB and Pivotal organise their work on pgAdmin following defined agile processes, creating backlogs from their own priorities and items from the pgAdmin bug tracker (Redmine)

Translations

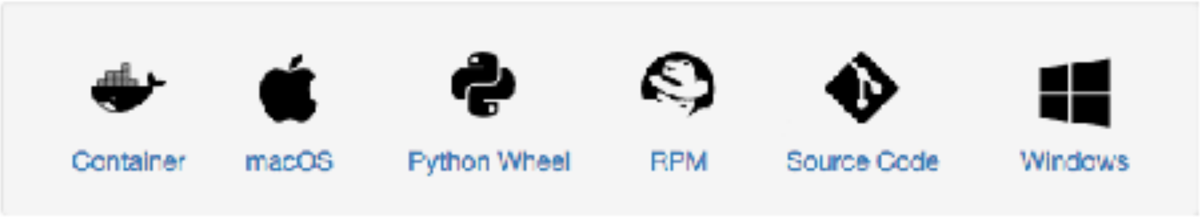
- Translations currently being managed on an ad-hoc basis
- No translation status monitoring on the website (yet)
- 5 translations at present:
  - German
  - Japanese
  - Polish
  - Russian
  - Chinese
- **We need a translation manager/coordinator!**

Releases

- Major and minor releases
- Minor releases include new features
- Major releases may be incompatible with previous releases in some way (e.g. config file changes)
- **No branches** - continuous release stream
- Higher release cadence than previous versions (~60 days on average)
- Some releases synchronised with new releases of PostgreSQL

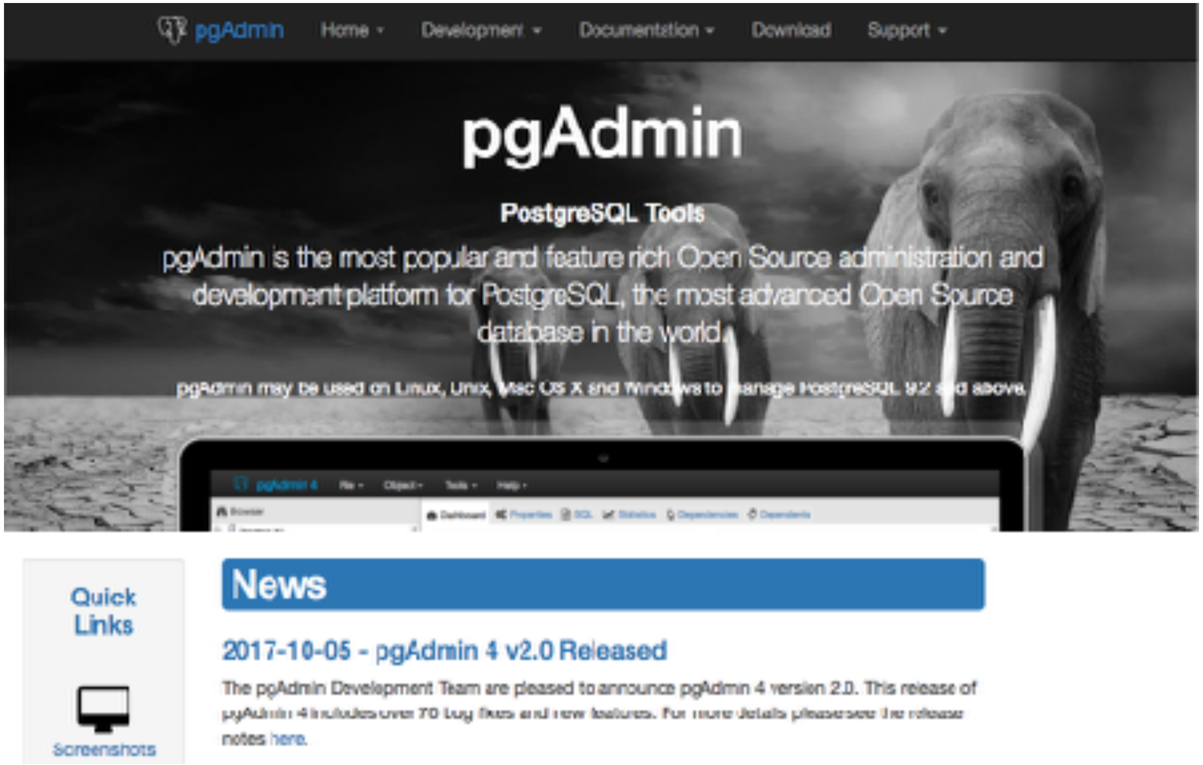
Packages

- Source and doc tarballs produced for each release
- Windows and macOS packages
- Python Wheel
- Docker container
- RPMs and other packages provided independently of the project by various volunteers
- Debian/Ubuntu packages are a work in progress by the PostgreSQL APT Team.



Online

- A shiny new website
- Written in Python and Django with PostgreSQL
- Looks almost… professional!



pgAdmin

PostgreSQL Tools

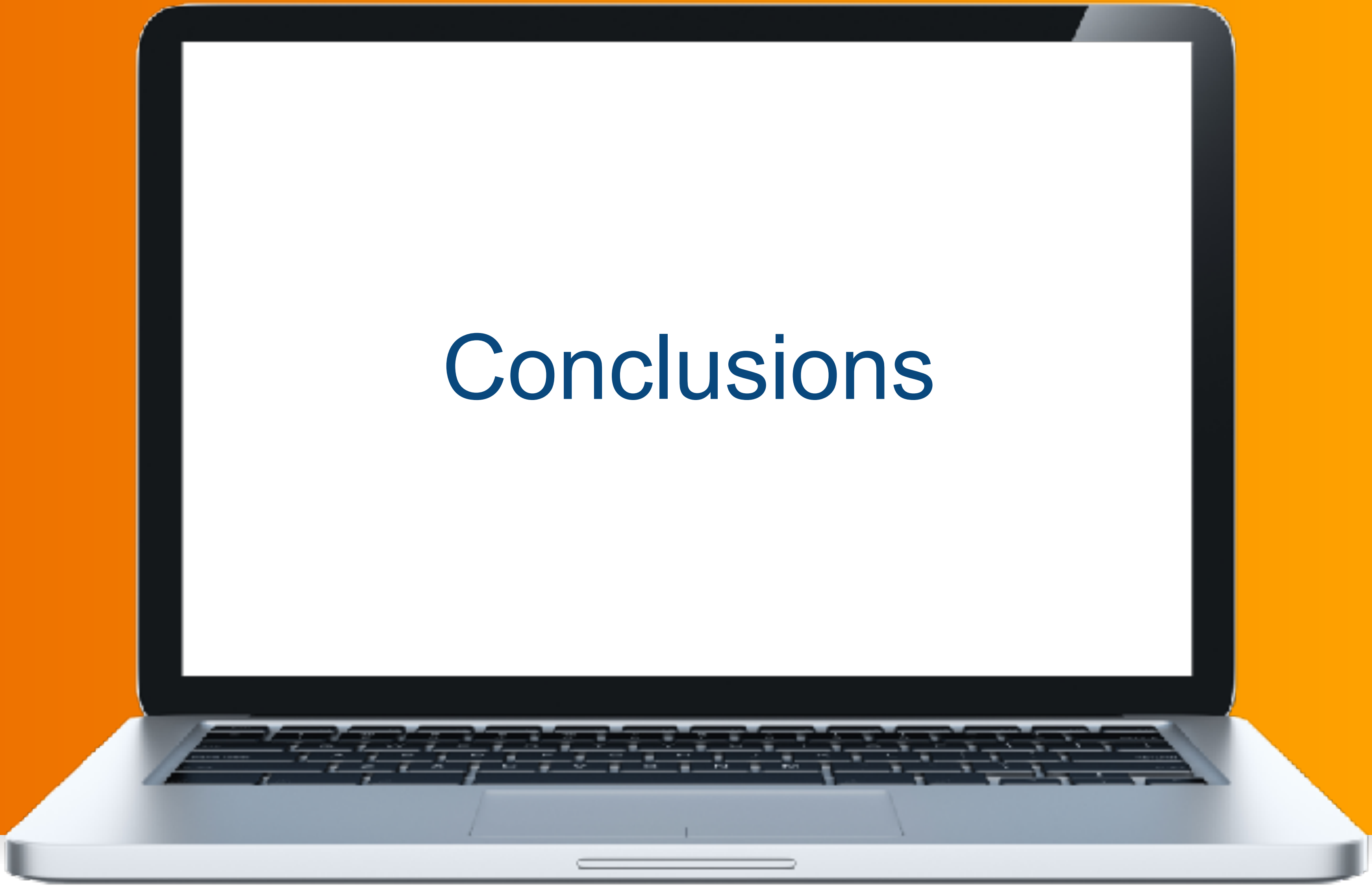
pgAdmin is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world.

pgAdmin may be used on Linux, Unix, Mac OS X and Windows to manage PostgreSQL 9.2 and above.

News

2017-10-05 - pgAdmin 4 v2.0 Released

The pgAdmin Development Team are pleased to announce pgAdmin 4 version 2.0. This release of pgAdmin 4 includes over 70 bug fixes and new features. For more details please see the release notes here.



Conclusions

Summary

- pgAdmin has grown as a project with every iteration over the last 20 years
- What started as a hobby has become much bigger, utilising modern agile processes to continually deliver improvements and enhancements
- Processes and standards have evolved, helping to minimise bugs, ensure code maintainability and consistent user experience.

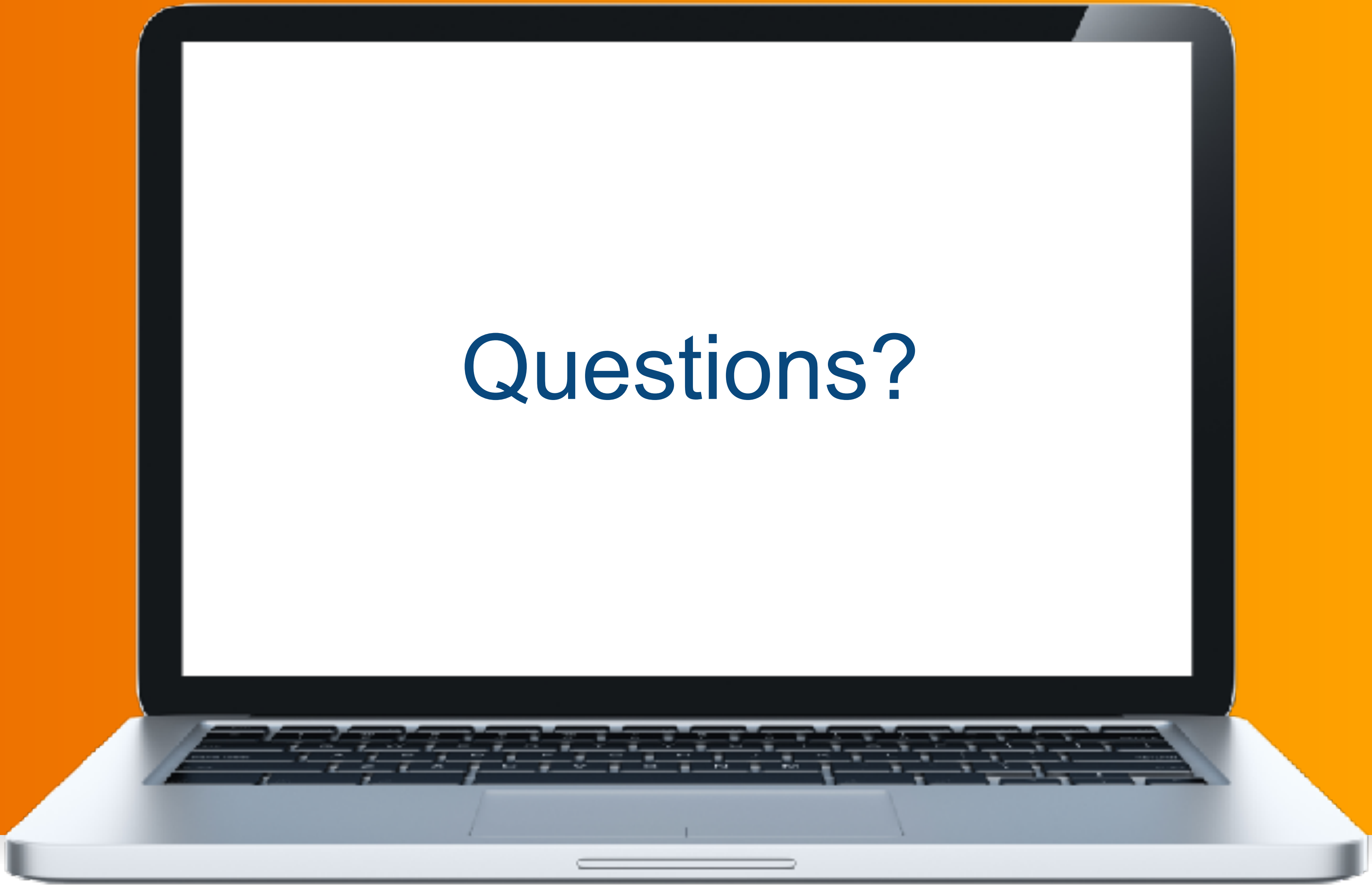
Join the project!

- Develop new code, expand the documentation, improve translations, build packages for a new platform, write an extension…
- There are many ways to contribute!

www.pgadmin.org

git.postgresql.org/gitweb/?p=pgadmin4.git;a=summary

pgadmin-hackers@postgresql.org



Questions?

THANK YOU
grazie
merci
spasiba
kam ouen
tak
gratzias
manana
mahalo
kitos
hvala
welalin
cheers
toda
gracias
grassie
thank you
danki
mahalo
danki
gracias
talofa
miigwetch
thanks
takk
domo arrigato
grattitude
merci
dankon
danke
dziekuje
thanks
na gode
mesi
modupe
kitos
takk