หลักสูตรอบรม

#1 DBMS of the Year 2023

BASIC

PostgreSQL for Administration

V.16

อาจารย์สามิตร
สถาบันไอทีจีเนียส

Canon
Canon Hi-Tech (Thailand) Ltd.

วิทยากร

อ.สามิตร โกยม

ปริญญาโทคณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

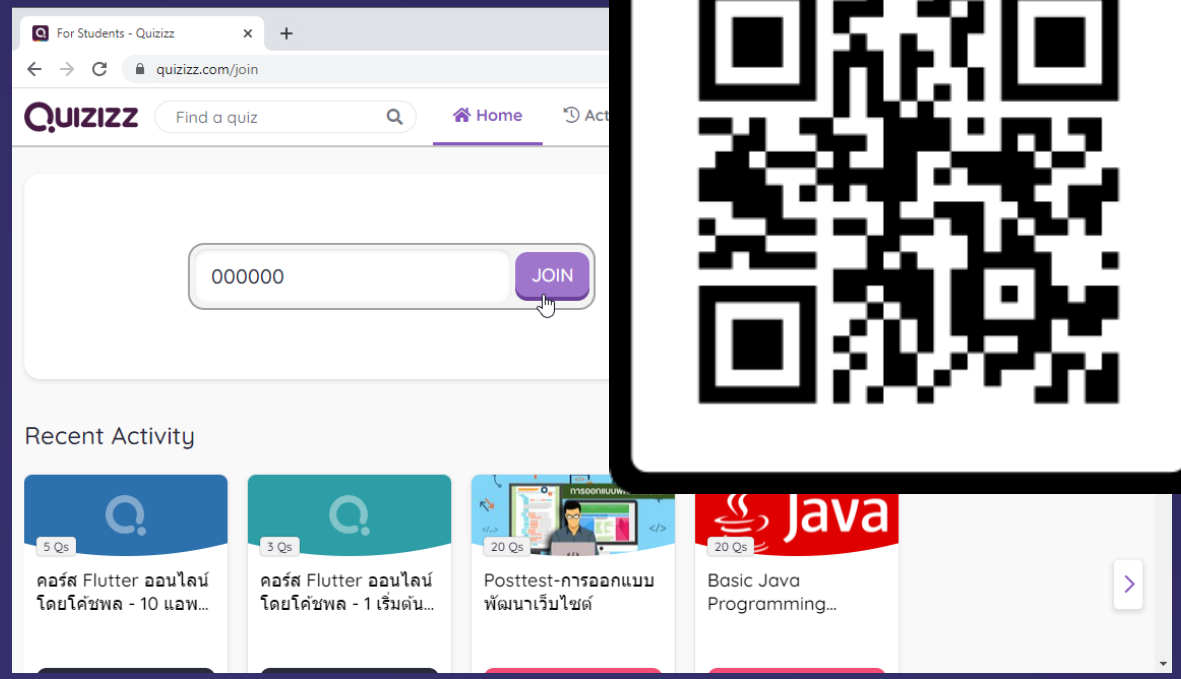
- **Frontend**
  Angular, React, NextJS, VueJS, Bootstrap,TailwindCSS
- **Backend**
  PHP, Python, Java, NodeJS, ASP.net
- **Database**
  MySQL, PostgreSQL, MS SQL Server, MongoDB
- **Mobile**
  Java, Kotlin, Objective C, Swift, Cordova, ionic, Flutter
- **DevOps**
  Git, Github, Gitlab, Docker, Kubernetes

แบบทดสอบก่อนอบรม

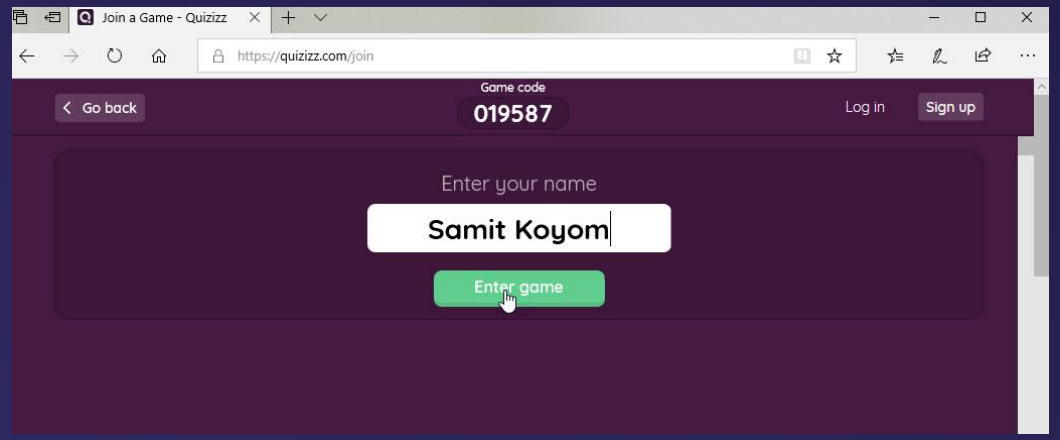
Pretest ทำแบบทดสอบก่อนเรียน

STEP 1: เข้าทำแบบทดสอบที่ลิงก์ ป้อนรหัสเข้าห้องสอบ

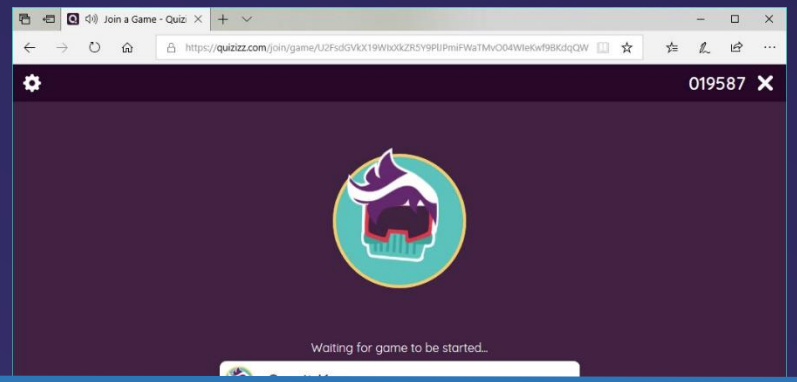
quizizz.com/join



STEP 2: ป้อนชื่อ



STEP 3: รอผู้สอน Start ข้อสอบ



ดาวน์โหลดเอกสารประกอบการอบรม

bit.ly/pgcanon

Course Outline

PostgreSQL for Administration (3 Days)

Section 1: บทนำ PostgreSQL คืออะไร เหตุใดจึงเป็นระบบฐานข้อมูลอันดับ 1
Section 2: ติดตั้ง PostgreSQL 15.x บน Windows & macOS
Section 3: เรียนรู้วิธีใช้ pgAdmin 4 ในการจัดการ PostgreSQL
Section 4: เรียนรู้คำสั่ง psql พื้นฐานสำหรับการใช้งาน PostgreSQL
Section 5: เรียนรู้วิธีเขียนคิวรี่ SQL พื้นฐาน
Section 6: เรียนรู้การปรับแต่งประสิทธิภาพ PostgreSQL
Section 7: PostgreSQL กับ Docker
Section 8: การใช้งาน PostgreSQL บน Vercel Cloud

DAY 1

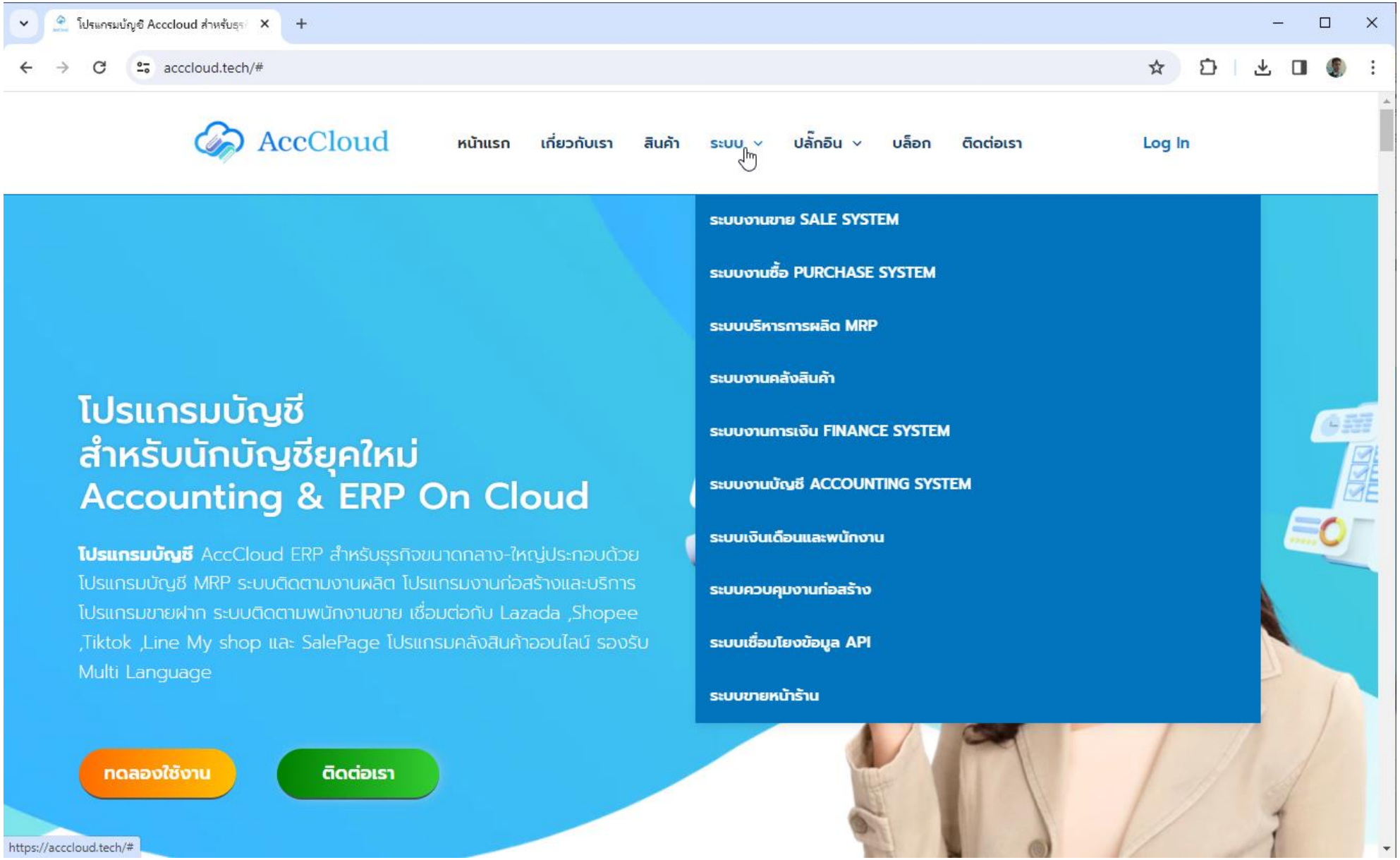
Section 1: PostgreSQL คืออะไร เหตุใดจึงเป็นระบบฐานข้อมูลอันดับ 1
Section 2: ติดตั้ง PostgreSQL 16.x บน Windows & macOS
Section 3: เรียนรู้วิธีใช้ pgAdmin 4 ในการจัดการ PostgreSQL

Section 1

PostgreSQL คืออะไร เหตุใดจึงเป็นระบบฐานข้อมูลอันดับ 1

- PostgreSQL คืออะไร?
- สถาปัตยกรรมของ PostgreSQL
- คุณสมบัติเด่นของ PostgreSQL
- เหตุใด PostgreSQL จึงเป็นระบบฐานข้อมูลอันดับ 1
- ตัวอย่างการใช้งาน PostgreSQL ในโลกจริง

โปรแกรมบัญชี Acccloud สำหรับธุร
acccloud.tech
AccCloud
หน้าแรก
เกี่ยวกับเรา
สินค้า
ระบบ
ปลั๊กอิน
บล็อก
ติดต่อเรา
Log In
โปรแกรมบัญชี
สำหรับนักบัญชียุคใหม่
Accounting & ERP On Cloud
โปรแกรมบัญชี AccCloud ERP สำหรับธุรกิจขนาดกลาง-ใหญ่ประกอบด้วย โปรแกรมบัญชี MRP ระบบติดตามงานผลิต โปรแกรมงานก่อสร้างและบริการ โปรแกรมขายฝาก ระบบติดตามพนักงานขาย เชื่อมต่อกับ Lazada ,Shopee ,Tiktok ,Line My shop และ SalePage โปรแกรมคลังสินค้าออนไลน์ รองรับ Multi Language
ทดลองใช้งาน
ติดต่อเรา
https://acccloud.me/login


โปรแกรมบัญชี Acccloud สำหรับธุร
acccloud.tech/#
AccCloud
หน้าแรก
เกี่ยวกับเรา
สินค้า
ระบบ
ปลั๊กอิน
บล็อก
ติดต่อเรา
Log In
ระบบงานขาย SALE SYSTEM
ระบบงานซื้อ PURCHASE SYSTEM
ระบบบริหารการผลิต MRP
ระบบงานคลังสินค้า
ระบบงานการเงิน FINANCE SYSTEM
ระบบงานบัญชี ACCOUNTING SYSTEM
ระบบเงินเดือนและพนักงาน
ระบบควบคุมงานก่อสร้าง
ระบบเชื่อมโยงข้อมูล API
ระบบขายหน้าร้าน
โปรแกรมบัญชี
สำหรับนักบัญชียุคใหม่
Accounting & ERP On Cloud
โปรแกรมบัญชี AccCloud ERP สำหรับธุรกิจขนาดกลาง-ใหญ่ประกอบด้วย โปรแกรมบัญชี MRP ระบบติดตามงานผลิต โปรแกรมงานก่อสร้างและบริการ โปรแกรมขายฝาก ระบบติดตามพนักงานขาย เชื่อมต่อกับ Lazada ,Shopee ,Tiktok ,Line My shop และ SalePage โปรแกรมคลังสินค้าออนไลน์ รองรับ Multi Language
ทดลองใช้งาน
ติดต่อเรา
https://acccloud.tech/#

STR Accountonline.com

BRD2020_RAC20_result.pdf 40 / 6103 100%

36

37

38

รายงานบัญชีแยกประเภท (General Ledger Report)

Period 2022-01-01 to 2022-10-31

Div. 00 - 01
Sec. 00 - 01
Acc.Code 111110 - 990050

Date	Voucher Code	Description	Debit	Credit	Balance
20/09/2565	CB-00-2209/0071	; ; บันทึกนำฝากเงินสด RV-0-2209/0327 คุณ ไพศาล ;	0.00	800.00	(14,815.00)
21/09/2565	CB-00-2209/0051	; ; บันทึกนำฝากเงินสด RV-00-2209/0257 บ ฯ (คุณคนึงนุช โอวรัฐตระกูล) ;	0.00	800.00	(15,615.00)
21/09/2565	CB-00-2209/0055	; ; บันทึกนำฝากเงินสด RV-00-2209/0261 บ สปา ;	0.00	800.00	(16,415.00)
22/09/2565	CB-00-2209/0053	; ; บันทึกนำฝากเงินสด RV-00-2209/0260 คุ ปัญญา ;	0.00	800.00	(17,215.00)
22/09/2565	CB-00-2209/0058	; ; บันทึกนำฝากเงินสด RV-00-2209/0307 คุ ชิต ;	0.00	800.00	(18,015.00)
22/09/2565	CB-00-2209/0060	; ; บันทึกนำฝากเงินสด RV-00-2209/0308 คุ วรประเสริฐ ;	0.00	3,700.00	(21,715.00)
22/09/2565	CB-00-2209/0062	; ; บันทึกนำฝากเงินสด RV-00-2209/0309 คุ ณารักษ์ ;	0.00	2,000.00	(23,715.00)
22/09/2565	CB-00-2209/0063	; ; บันทึกนำฝากเงินสด RV-00-2209/0309 คุ ณารักษ์ ;	0.00	1,000.00	(24,715.00)
23/09/2565	CB-00-2209/0092	; ; บันทึกนำฝากเงินสด RV-00-2209/0359 ค ภาคกุล ;	0.00	3,000.00	(27,715.00)
23/09/2565	CB-00-2209/0094	; ; บันทึกนำฝากเงินสด RV-00-2209/0360	0.00	3,520.00	(31,235.00)

File Edit Selection View Go Run Terminal Help — Get Started - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[redacted]ud=# \dn+
                          List of schemas
  Name  |  Owner   |  Access privileges   |      Description
--------+----------+----------------------+------------------------
 PAYROL | postgres |                      |
 STRDBA | postgres |                      |
 public | postgres | postgres=UC/postgres+| standard public schema
        |          | =UC/postgres         |
(3 rows)

[redacted]ud=# SELECT COUNT(1) FROM "STRDBA"."VOUCHERDETAIL";
  count
---------
 6139631
(1 row)

[redacted]ud=#
```

```
a[redacted]d=# \dn+
                          List of schemas
  Name  |  Owner   |  Access privileges   |      Description
--------+----------+----------------------+------------------------
 PAYROL | postgres |                      |
 STRDBA | postgres |                      |
 public | postgres | postgres=UC/postgres+| standard public schema
        |          | =UC/postgres         |
(3 rows)

a[redacted]d=# SELECT COUNT(1) FROM "STRDBA"."VOUCHERDETAIL";
  count
---------
 6139631
(1 row)

a[redacted]d=#
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL — 1: bash, bash

```
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------+----------+----------+------------+------------+-----------------------
 [redacted]d | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 demodb    | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(5 rows)

[redacted]s=# SELECT pg_size_pretty( pg_database_size('a[redacted]ud'));
 pg_size_pretty
----------------
 8810 MB
(1 row)

postgres=#
```

```
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------+----------+----------+------------+------------+-----------------------
 [redacted]d | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 demodb    | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(5 rows)

[redacted]s=# SELECT pg_size_pretty( pg_database_size('a[redacted]ud'));
 pg_size_pretty
----------------
 8810 MB
(1 row)
```

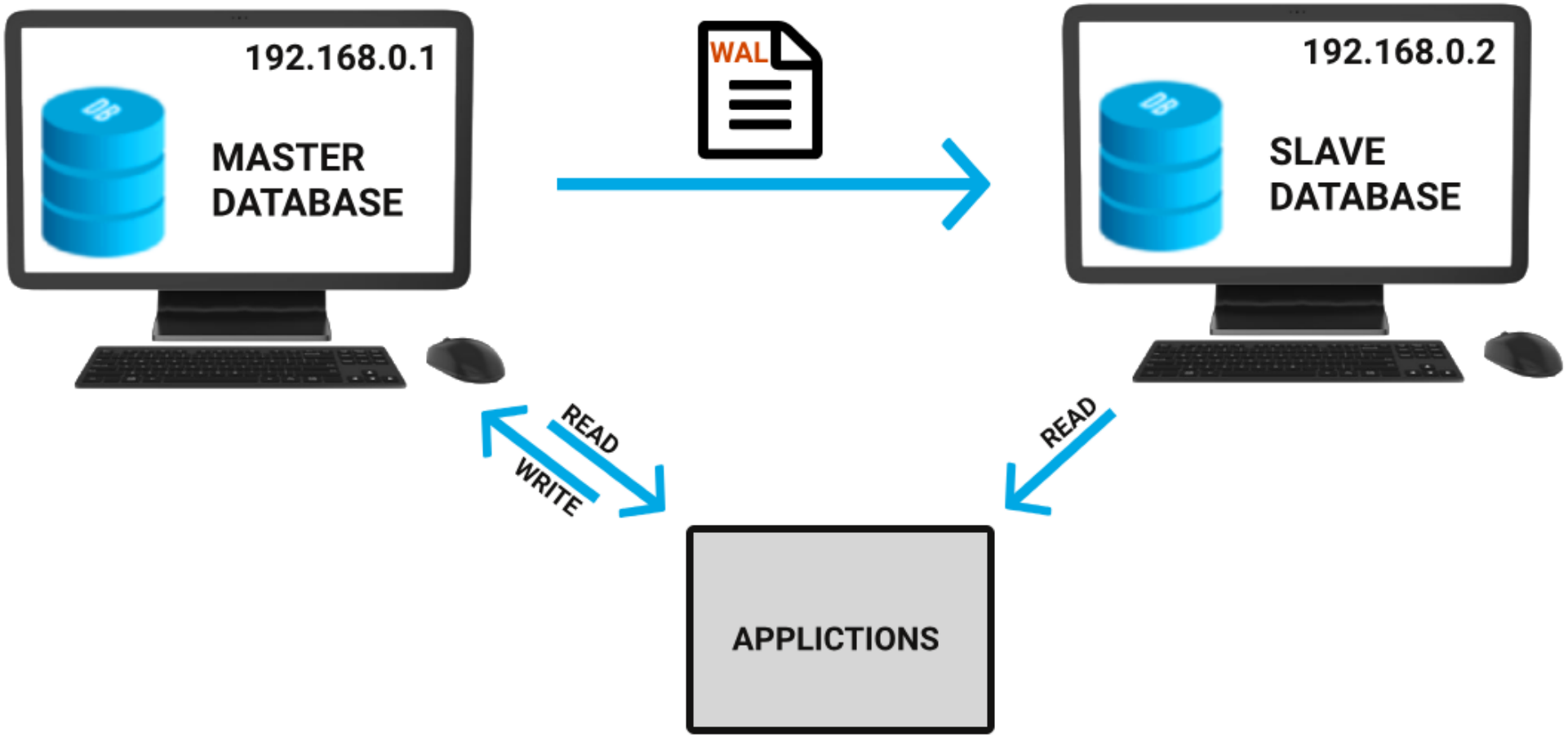
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash, bash

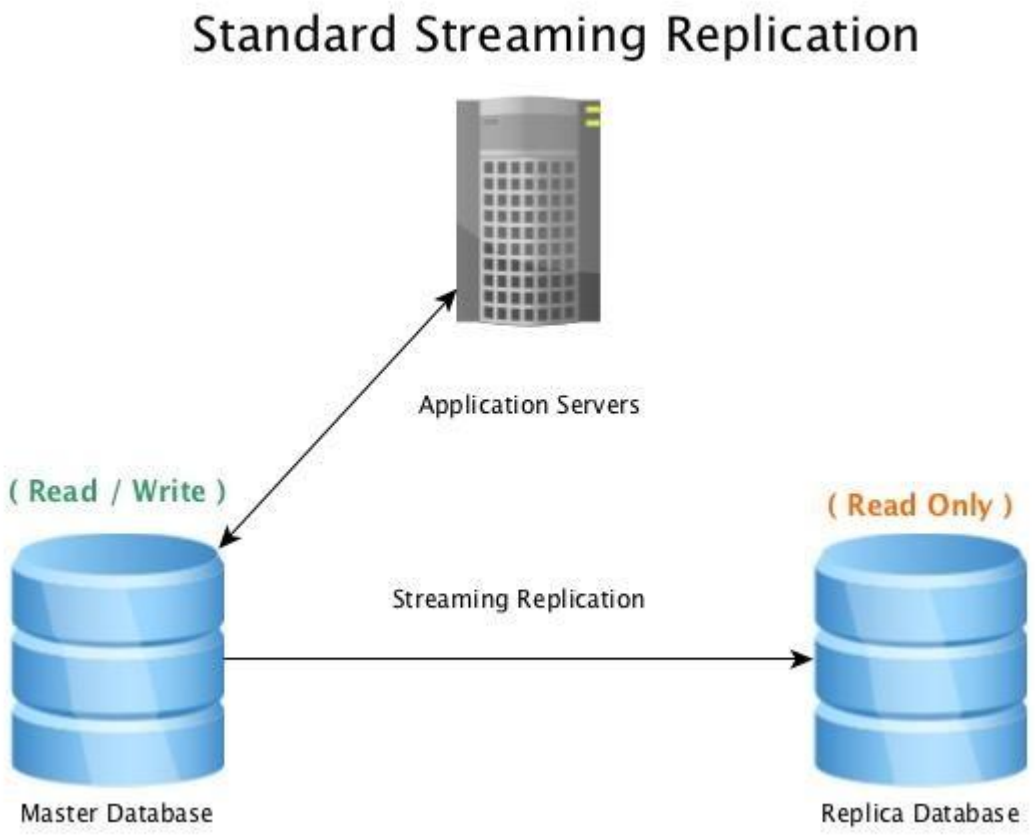
STRDBA	SCREEN	0
STRDBA	COMMIS	0
STRDBA	SETUPM	0
STRDBA	SUPPLI	0
STRDBA	PRODRE	0
STRDBA	PRODUC	0
STRDBA	SERVIC	0
STRDBA	CUSTOM	0
STRDBA	DELIVE	0
STRDBA	PURCHA	0
STRDBA	INTERN	0
STRDBA	PROFOR	0
STRDBA	JOURNA	0
STRDBA	SERVIC	0
STRDBA	BOARDR	0
STRDBA	PRODRE	0
STRDBA	SERVMA	0
STRDBA	PRODUC	0
STRDBA	MENUMA	0
STRDBA	CUSTOM	0
STRDBA	DELIVE	0
STRDBA	PRODUC	0
STRDBA	CLAIMD	0

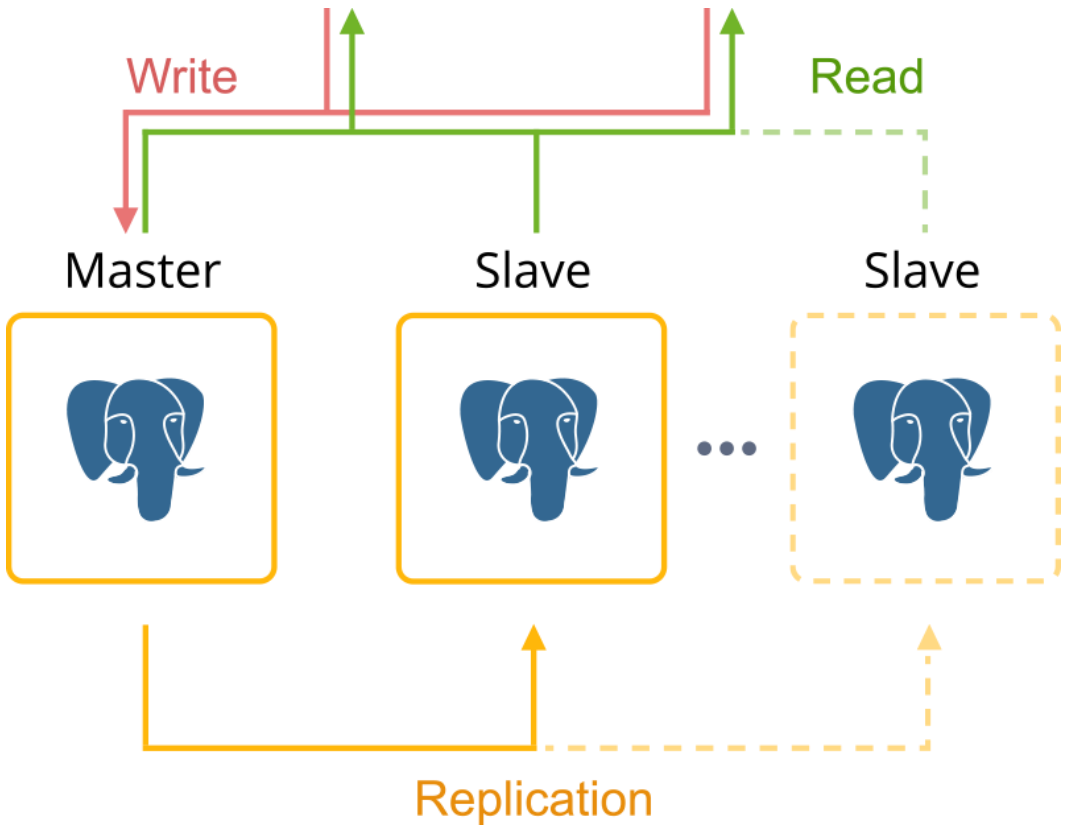
(593 rows)

STRDBA	DELIV	0
STRDBA	SCREE	0
STRDBA	COMPA	0
STRDBA	ASSET	0
STRDBA	TRACK	0
PAYROL	EMPLO	0
STRDBA	SECTI	0
STRDBA	PRODU	0
STRDBA	PRODR	0
STRDBA	DELIV	0
PAYROL	LEAVE	0
STRDBA	CREDI	0
STRDBA	PRODU	0
STRDBA	MENUM	0
STRDBA	CUSTO	0
STRDBA	ASSET	0
STRDBA	DELIV	0
STRDBA	JOBRM	0
STRDBA	DEFAU	0
STRDBA	QCINS	0
STRDBA	PRODU	0
STRDBA	PRODU	0
STRDBA	DEPOS	0

(593 rows)


192.168.0.1
DB
MASTER
DATABASE
WAL
192.168.0.2
DB
SLAVE
DATABASE
READ
WRITE
READ
APPLICTIONS


Standard Streaming Replication
Application Servers
(Read / Write)
(Read Only)
Streaming Replication
Master Database
Replica Database


Write
Read
Master
Slave
Slave
Replication

PostgreSQL คืออะไร

RDBMS

PostgreSQL คือ ระบบจัดการฐานข้อมูลเชิงวัตถุ-สัมพันธ์ (Object-Relational Database Management System หรือ ORDBMS) ซึ่งปรับปรุงจากต้นแบบระบบฐานข้อมูล POSTGRES 4.2 ของมหาวิทยาลัยแคลิฟอร์เนีย วิทยาลัยเขตเบอร์กเลย์ (UC Berkeley) ภายใต้ความควบคุมของ Professor Michael Stonebraker โดยได้รับเงินวิจัยสนับสนุนจาก the Defense Advanced Research Project Agency (DARPA) , the Army Research Office (ARO), the National Science Foundation (NSF) และ ESL, Inc

PostgreSQL บน Supabase

Supabase เปิดตัว Postgres Language Server เขียนโค้ดใน editor ได้ง่ายขึ้นในอนาคต

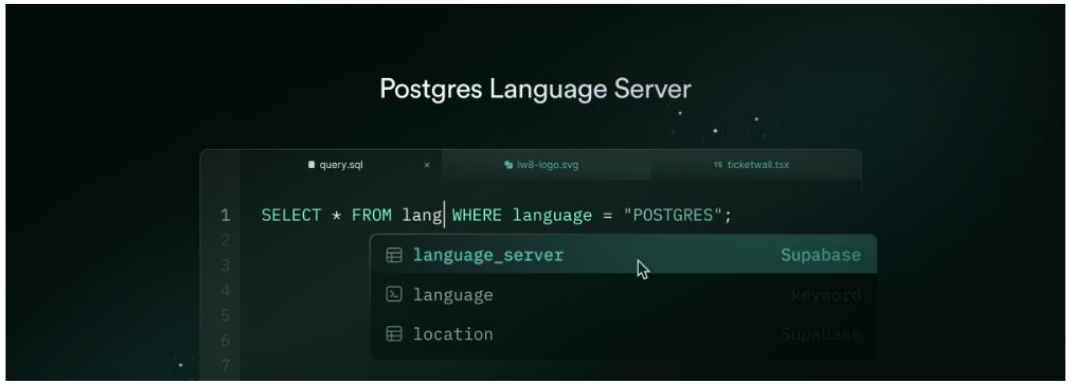
By: lew on 6 August 2023 - 23:22 Tags: PostgreSQL Supabase

Supabase โครงการ back-end-as-a-service แบบโอเพนซอร์สเปิดโครงการ Postgres Language Server เปิดทางให้ editor ภาษาต่างๆ สามารถเขียนโค้ด SQL แบบ PostgreSQL ได้ง่ายขึ้น

Language Server นั้นเป็นโปรโตคอลที่สื่อสารกับ editor เพื่อให้ editor รองรับภาษาใหม่ๆ ได้เต็มรูปแบบ โดยรองรับฟีเจอร์เช่น syntax highlighting, auto-completion, แสดงจุดผิดพลาดของโค้ด, ทำ code formatting

โครงการนี้จะมุ่งรองรับ PostgreSQL เท่านั้นไม่สนใจจะรองรับภาษา SQL ของโปรแกรมอื่นๆ แต่อย่างใด เนื่องจากหัวใจของโครงการจะใช้ `libpg_query` มาพัฒนาเพิ่มให้กลายเป็น Language Server โดยทาง Supabase ดูโครงการต้นแบบ คือ `rust-analyzer` และตอนนี้เริ่มใช้งานกับ Neovim ได้แล้ว แม้จะยังไม่พร้อมอย่างมาก แนะนำสำหรับนักพัฒนาที่อยากช่วยโครงการอย่างเดียว

ที่มา - GitHub: supabase/postgres_lsp



Post LINE it! FB Share

Get latest news from Blognone Follow @blognone Follow 267K

PostgreSQL บน AWS

Amazon RDS รองรับ PostgreSQL แล้ว

By: mk blog on 15/11/13 9:24 Tags: AWS Database PostgreSQL

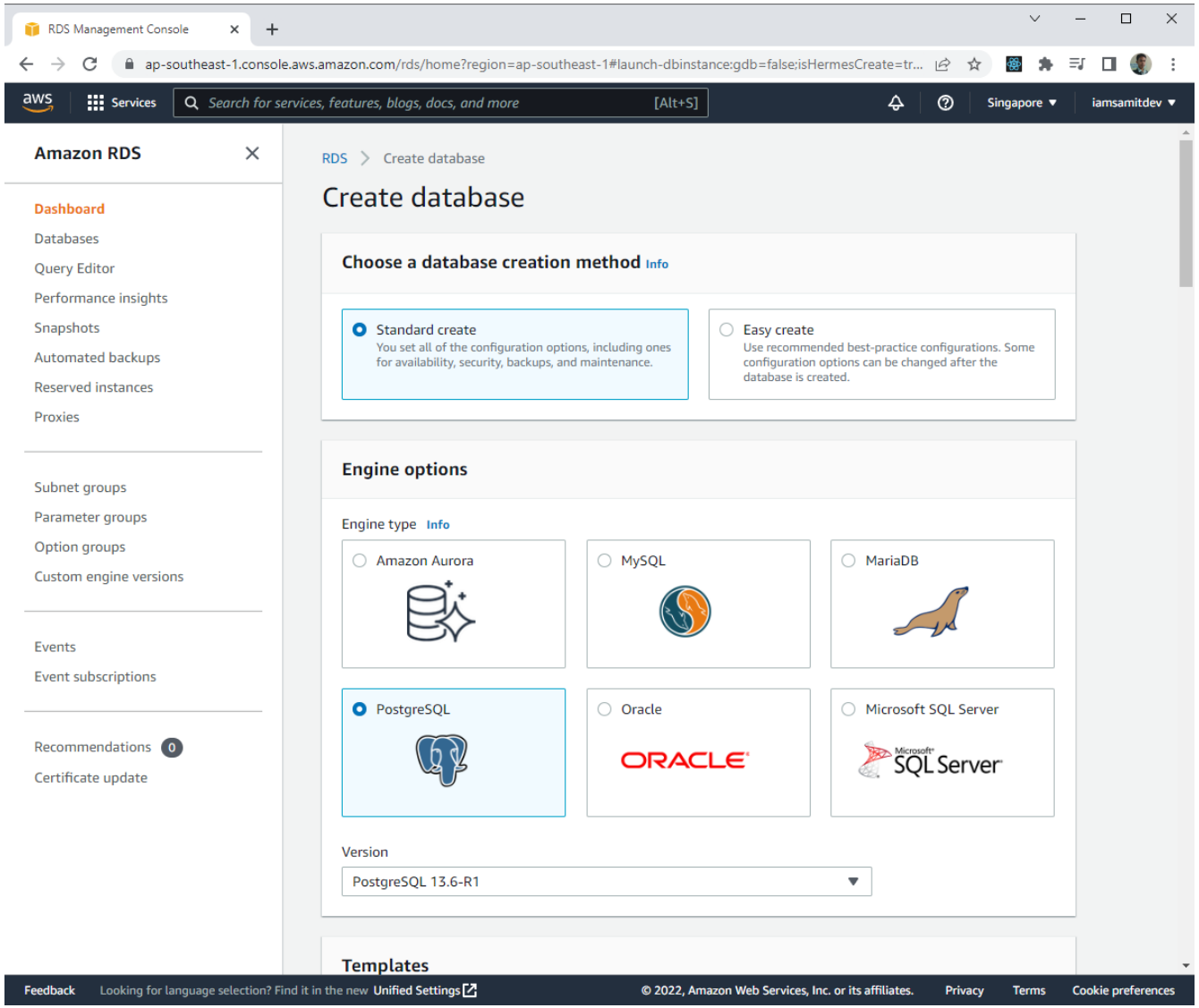
บริการฐานข้อมูลบนกลุ่มเมฆ Amazon Relational Database Service (ตัวย่อ RDS) เปิดบริการตั้งแต่ปี 2009 โดยเริ่มจาก MySQL จากนั้นก็ขยายมายังฐานข้อมูลตัวอื่นๆ คือ Oracle และ Microsoft SQL Server

วันนี้ Amazon ประกาศรองรับฐานข้อมูลโอเพนซอร์สอีกตัวคือ PostgreSQL แล้ว เบื้องต้นยังมีเฉพาะเวอร์ชัน 9.3.1 แต่ Amazon ก็ประกาศว่าจะรองรับเวอร์ชันใหม่ๆ ในอนาคตด้วย ส่วนฟีเจอร์ด้านฐานข้อมูลบนกลุ่มเมฆก็ไม่ต่างอะไรจากฐานข้อมูลยี่ห้ออื่นๆ มากนัก มีทั้งการแบ็คอัพอัตโนมัติ, การสร้าง instance ของฐานข้อมูลใหม่โคลนจาก instance เดิมตามช่วงเวลาที่ต้องการ, การสร้างฐานข้อมูลข้ามโซนกลุ่มเมฆ, รองรับ VPS เป็นต้น

ที่มา - Amazon Web Services Blog

Tweet 4 g+1 7 Like Share 16 people like this. Be the first of your friends.

PostgreSQL บน AWS



PostgreSQL บน IBM

PostgreSQL - IBM Cloud

cloud.ibm.com/catalog/content/postgresql-QmI0bmFtaS1wb3N0Z3Jlc3Fs-global

IBM Cloud | Search resources and offerings... | Catalog | Docs | Support | Manage | Samit Koyom's ...

Catalog / Software /

PostgreSQL

Third party • **Version:** 13.2.0-12 • **Date of last update:** 03/05/2021 • Get help?

Create | Readme

Select your target

IBM Kubernetes Service

VMware vCenter Server

Select a method

OVA Image
Version 13.2.0-12

View the existing installations

Select your cluster

Select a Kubernetes cluster in your account to copy the public image across the private network.

Link your PostgreSQL instance to a Kubernetes cluster with a billable pricing plan. If you don't have one, create a cluster and then return to this page after it's active.

Summary

PostgreSQL
Target: VMware vCenter Server
Method: OVA Image
VMware vCenter Server: No vCenter server available
Workspace: postgresql-03-08-2021
Resource group: Default

FEEDBACK

I have read and agree to the following third party terms:
Apache 2.0
VMware

Install

PostgreSQL บน IBM

Databases for PostgreSQL - IBM

cloud.ibm.com/catalog/services/databases-for-postgresql

IBM Cloud | Search resources and offerings... | Catalog | Docs | Support | Manage | Samit Koyom's Account

Catalog / Services /

Pg Databases for PostgreSQL

IBM • Date of last update: 03/08/2021 • Docs • API docs

Looking for Compose for PostgreSQL?

Compare Versions

Create | About

Select a location

Select a location

Dallas (us-south)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: United States

Plan	Features	Pricing
Standard	**Serverless scaling** Automatic backup orchestration Bring your own encryption key	$0.58 USD/GB-Disk $5.00 USD/GB-RAM $0.03 USD/GB-Backups $30.00 USD/Virtual Processor Core

One highly available PostgreSQL instance with two data members. The Standard plan is priced based on disk storage, RAM, backup storage, and vCPUs allocated to deployments, prorated hourly. Databases for PostgreSQL instances have a minimum of 5GB of Disk and 1GB of RAM per data member. Dedicated cores can optionally be provisioned at a minimum of 3 cores per member.

Summary

Databases for PostgreSQL Estimate costs

Location: Dallas

Plan: Standard

Service name: Databases for PostgreSQL-a3

Resource group: Default

FEEDBACK

Upgrade

Add to estimate

PostgreSQL บน IBM

Databases for EDB - IBM Cloud

cloud.ibm.com/catalog/services/databases-for-enterprisedb

IBM Cloud | Search resources and offerings... | Catalog | Docs | Support | Manage | Samit Koyom's ...

Catalog / Services /

En Databases for EDB

IBM • **Date of last update:** 03/08/2021 • Docs • API docs

Create | About

Select a location

Select a location

Dallas (us-south)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: United States

Plan	Features	Pricing
Standard	**Serverless scaling** Automatic backup orchestration Bring your own encryption key	$0.58 USD/GB-Disk $0.03 USD/GB-Backups $11.50 USD/GB-RAM $100.00 USD/Virtual Processor Core

One highly available EDB instance with three data members. The Standard plan is priced based on disk storage, RAM, backup storage, and vCPUs allocated to deployments, prorated hourly. Databases for EDB instances have a minimum of 20GB of Disk, 1GB of RAM, and 3 cores per data member.

Summary

Databases for EDB Estimate costs

Location: Dallas
Plan: Standard
Service name: Databases for EDB-lt
Resource group: Default

FEEDBACK

Upgrade

Add to estimate

View terms

Enterprise Ready

Fully Managed

Highly Customizable

Object-Relational Database

Open Source

Language Independent

Fast Developent and Easily Scalable

PostgreSQL คืออะไร

- PostgreSQL is pronouced "post-gres“
- ANSI-Standard SQL
- General purpose
- Object-relational database management system (RDBMS)
- Free and Open source RDBMS
- Platform independent
- Very stable and requires minimum maintenance
- Large storage
- Fast performance
- PostgreSQL is extensible: You can defend your own data type, index types , plugin etc.
- Active community support

Facts about PostgreSQL

- The world's most advanced open sources databases
- Designed for extensibility and customization
- ANSI/ISO compliant SQL Support
- Actively developed for more than 30 years
  - University Postgres (1986-1993)
  - Postgres95 (1994-1995)
  - PostgreSQL (1996 – Current)

ความเป็นมาของ PostgreSQL

กำเนิด Postgres

PostgreSQL มีต้นตอมาจากโครงการ University Ingres ตั้งแต่ปี ค.ศ. 1977 ภายใต้การควบคุมการวิจัยโดย Professor Michael Stonebraker ซึ่งต้นแบบของ Ingres ได้นำไปปรับปรุงเป็นเชิงพาณิชย์โดย Relational Technologies/Ingres Corporation

ต่อมาในปี ค.ศ. 1986 Professor Michael Stonebraker เล็งเห็นว่าระบบฐานข้อมูลเชิงสัมพันธ์ในขณะนั้น ไม่เพียงพอในการรองรับระบบงานด้านฐานข้อมูลที่ซับซ้อนในอนาคตได้ ซึ่งต้องการความสามารถพิเศษเพิ่มเติมแบ่งออกเป็น 3 หัวข้อใหญ่ๆ คือ

1. ระบบจัดการฐานข้อมูล (Database Management System) ต้องการองค์ความรู้และสถาปัตยกรรมโครงสร้างใหม่ในการจัดการฐานข้อมูลให้มีประสิทธิภาพยิ่งขึ้น

2. ระบบจัดการฐานความรู้ (knowledge-based Management System) เป็นโครงสร้างใหม่เพื่อสร้างฐานความรู้ ซึ่งเห็นได้ทั่วไปในการจัดการกฎเกณฑ์และข้อกำหนดทางธุรกิจ (Business Rules and Policy) องค์ความรู้ใหม่ที่นำมาเพิ่มเติมลงในระบบฐานข้อมูล เพื่อใช้ในการสนับสนุนความสามารถดังกล่าว ได้แก่ database constraints, triggers, rules และ transaction integrity เป็นต้น

3. ระบบจัดการวัตถุ (Object Management System) เป็นโมเดลใหม่ที่จำเป็นต้องขยายต่อเพื่อช่วยเสริมให้ระบบฐานข้อมูลเชิงสัมพันธ์สามารถรองรับระบบงานที่ต้องการประเภทข้อมูล (data type) และ โมเดลเชิงวัตถุ (object-oriented model) ระบบงานที่ต้องการใช้โมเดลเชิงวัตถุและระบบจัดการฐานข้อมูล ได้แก่ งานประเภท CAD-CAM หรือ multimedia เป็นต้น องค์ความรู้ที่นำมาเพิ่มเติมเพื่อใช้ในการสนับสนุนความสามารถดังกล่าว ได้แก่ inheritance, user-defined data types และ functions เป็นต้น

ต้นแบบ Postgres

ต้นแบบระบบฐานข้อมูลตัวแรกในโครงการนี้ใช้ชื่อเริ่มแรกว่า POSTGRES ต้นแบบตัวแรกถูกเขียนด้วยภาษา LISP ซึ่งทำงานได้ช้ามาก หลังจากนั้นจึงเขียนใหม่ด้วยภาษา C ต้นแบบเริ่มใช้งานได้เมื่อปี ค.ศ. 1987 และได้ถูกเปิดตัวครั้งแรกในงานประชุมวิจัย ACM-SIGMOD ในปีเดียวกัน นับจากนั้นมาแนวความคิดใหม่ต่างๆ ได้ถูกเพิ่มเติม พร้อมทั้งสิ่งที่ล้าสมัยถูกรื้อทิ้งและได้รับการออกแบบพัฒนาใหม่มาตลอดเวลา

ระหว่างนั้น POSTGRES ถูกนำไปใช้เป็นระบบจัดการฐานข้อมูลในระบบงานที่ใช้งานจริงไม่ว่าจะเป็นระบบวิเคราะห์การเงิน ระบบตรวจวัดสมรรถภาพเครื่องยนต์อากาศยาน ระบบติดตามทางดาราศาสตร์ ระบบฐานข้อมูลการแพทย์ หรือระบบงานเชิงภูมิศาสตร์ นอกจากถูกนำไปใช้ในมหาวิทยาลัยในงานการศึกษาในปี ค.ศ. 1992 ยังถูกนำไปใช้เป็นระบบจัดการฐานข้อมูลหลักของโครงการ Sequoia 2000 ของ NASA ซึ่งใช้เก็บข้อมูลเกี่ยวกับความเปลี่ยนแปลงของโลกไม่ว่าจะเป็นภูมิอากาศ ระดับน้ำ รังสี และอื่นๆ โดยมีจำนวนข้อมูลประมาณ 2 terabytes/วัน การที่ถูกนำไปใช้อย่างมากมายนี่เอง ทำให้จำนวนผู้ใช้เพิ่มขึ้นอย่างรวดเร็ว

ต้นแบบ Postgres

โดยจุดประสงค์ของโครงการนั้น เพียงเพื่อใช้เป็นต้นแบบในการทดสอบหลักการและทฤษฎีที่คิดค้นแต่การที่ได้รับความนิยมอย่างมากทำให้ต้องให้การสนับสนุนแก่ผู้ใช้จนเกิดเป็นภาระแก่ผู้พัฒนา ทำให้จุดประสงค์ของโครงการถูกเบี่ยงเบนไป อีกทั้งการเพิ่มเติมความสามารถต่างๆ ลงไปในซอฟต์แวร์ ทำให้ POSTGRES มีขนาดใหญ่มากขึ้นจนยากที่จะควบคุม ด้วยเหตุผลต่างๆ ดังกล่าว ทำให้โครงการ POSTGRES ได้สิ้นสุดอย่างเป็นทางการเมื่อปี ค.ศ. 1993 ที่เวอร์ชัน 4.2 อย่างน่าเสียดาย

Postgres95

ในปี ค.ศ. 1994 นักศึกษาสองคนของโครงการ POSTGRES คือ Andrew Yu และ Jolly Chen ได้นำ POSTGRES เวอร์ชัน 4.2 มาทำการรื้อใหม่หมด ซึ่งทั้งสองได้แก้ไขข้อบกพร่อง และเพิ่มเติมข้อดีต่างๆ ให้แก่ซอฟต์แวร์ดังนี้

- ตัดทอน source code ส่วนที่ซ้ำซ้อนออกไป โดยใช้ ANSI C ทั้งหมดเพื่อประโยชน์ในการพอร์ตข้ามระบบ
- เปลี่ยนภาษาในการสืบค้นข้อมูลมาตรฐาน SQL แทนที่ภาษาในการสืบค้นเดิม Postquel
- ปรับปรุงสมรรถภาพให้สามารถทำงานได้เร็วขึ้น 30-50%
- เพิ่มเติมเครื่องมือต่างๆ เพื่อสนับสนุนในการใช้งาน เช่น Tcl/Tk Interface

รวมทั้งปรับแต่งระบบต่างๆ และเปิดให้ download ผ่านทาง Web site ในรูปแบบของ open-source ซอฟต์แวร์ภายใต้ชื่อรหัสโครงการใหม่ Postgres95

ในปี ค.ศ. 1996 ชื่อ Postgres95 ถูกเปลี่ยนใหม่เป็น PostgreSQL โดยเริ่มต้นที่เวอร์ชั่น 6.0 ด้วยเหตุผลหลังจากการเพิ่มความสามารถในภาษาสืบค้นข้อมูล SQL เพื่อให้เทียบเท่ากับมาตรฐาน SQL-92 ลงในระบบฐานข้อมูล Postgres95

ในระยะเริ่มต้นโครงการนั้นต้องการอาสาสมัคร (ที่เป็นมืออาชีพ) โดยไม่เน้นที่จำนวนแต่ต้องมีเวลาทุ่มเทมากกว่าจำนวนมากแต่ไม่ค่อยมีเวลาทำงาน เริ่มแรกนอกจาก Jolly Chen ยังมีผู้พัฒนาที่ทุ่มเทให้กับโครงการอีก 4 คน คือ Marc Fournier ชาว Canada, Vadim Mikheev ชาว Russia, Thomas Lockhart และ Bruce Momjian ชาว American ในเริ่มต้นเป็นการแก้ไขข้อผิดพลาดต่างๆ ที่มีอยู่ตามที่ได้รับการแจ้งมา มีการจำแนกประเภทของข้อผิดพลาดเพื่อจัดลำดับในการแก้ไข บางอย่างสามารถแก้ไขได้ง่าย บางอย่างจำเป็นต้องใช้ความรู้เพิ่มเติมในการวิจัยเพิ่มเติม อย่างไรก็ตาม ในการปรับปรุงนั้นเน้นอยู่ที่ความน่าเชื่อถือของระบบเนื่องจากงานฐานข้อมูลเป็นงานที่ละเอียดอ่อนระบบงานที่ทำงานภายใต้ระบบฐานข้อมูลไม่เหมือนงานประเภทอื่น เช่น โปรแกรมจัดการเอกสารหรือเกมส์ ที่ระบบหยุดทำงานแล้วเริ่มใหม่ได้โดยไม่สนใจงานที่ทำมา

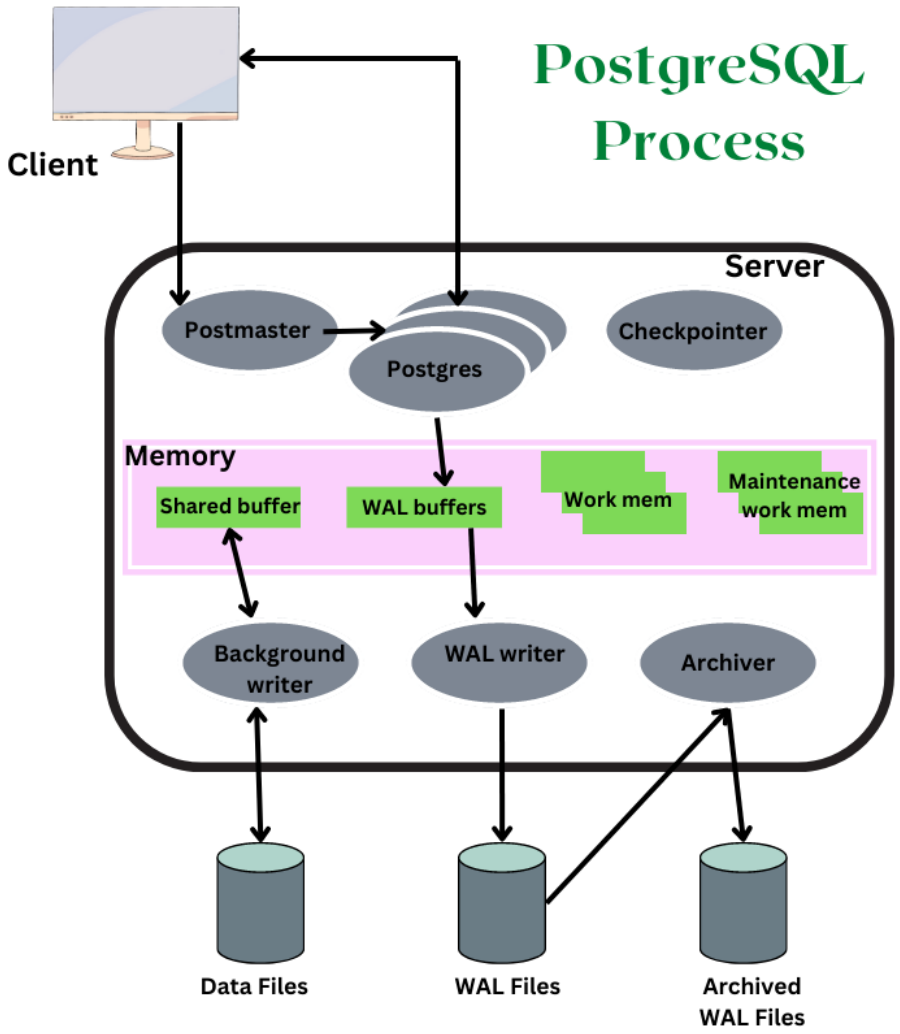
ในปัจจุบัน มีนักพัฒนาจำนวนมากร่วมกันพัฒนาเพิ่มเติมความสามารถให้กับ PostgreSQL ส่วนของฐานผู้ใช้เองก็ขยายมากขึ้น อีกทั้งได้มีการตั้งบริษัทเพื่อให้บริการสนับสนุนการใช้งาน และให้คำปรึกษาทางเทคนิคอีกด้วย

ตั้งแต่ PostgreSQL version 8 เป็นต้นมาได้ออก version สำหรับ Windows Platform อย่างจริงจัง ไม่ต้องใช้งานบน Cygwin เพื่อจำลอง Unix บน Windows อีกต่อไป ทำให้ได้รับความสนใจเพิ่มขึ้น เนื่องจากได้ฐานผู้ใช้กลุ่มใหญ่จาก Platform นี้

อาจกล่าวได้ว่า PostgreSQL ได้มีการพัฒนามาอย่างต่อเนื่องตลอดเวลา ความสามารถหลักที่เพิ่มเติมลงในระบบฐานข้อมูลนับจาก Postgres95 ศึกษาเพิ่มเติมได้จาก http://en.wikipedia.org/wiki/PostgreSQL

สถาปัตยกรรมของ PostgreSQL

PostgreSQL เป็นระบบจัดการฐานข้อมูลแบบ relational database management system (RDBMS) ที่มีความซับซ้อนและมีประสิทธิภาพสูง สถาปัตยกรรมของ PostgreSQL นั้นออกแบบมาเพื่อรองรับการใช้งานที่หลากหลาย ตั้งแต่การใช้งานขนาดเล็กไปจนถึงการใช้งานขนาดใหญ่



องค์ประกอบหลักของ PostgreSQL

1. เซิร์ฟเวอร์ (Server): เซิร์ฟเวอร์ PostgreSQL เป็นโปรแกรมหลักที่รับผิดชอบในการจัดการฐานข้อมูล เซิร์ฟเวอร์จะทำงานบนระบบปฏิบัติการ Linux, Windows, macOS และอื่นๆ

2. ไคลเอนต์ (Client): ไคลเอนต์ PostgreSQL เป็นโปรแกรมที่ใช้สำหรับเชื่อมต่อและโต้ตอบกับฐานข้อมูล ไคลเอนต์มีหลายประเภท เช่น psql, pgAdmin, DBeaver, และอื่นๆ

3. ฐานข้อมูล (Database): ฐานข้อมูล PostgreSQL ประกอบไปด้วยชุดของตาราง (Tables) มุมมอง (Views) ฟังก์ชัน (Functions) และข้อมูลอื่นๆ

4. ตาราง (Table): ตาราง PostgreSQL เป็นโครงสร้างที่ใช้สำหรับจัดเก็บข้อมูล ตารางประกอบไปด้วยคอลัมน์ (Columns) และแถว (Rows)

5. คอลัมน์ (Column): คอลัมน์ PostgreSQL คือชุดของข้อมูลที่มีประเภทเดียวกัน เช่น ตัวเลข ข้อความ วันที่ และอื่นๆ

6. แถว (Row): แถว PostgreSQL คือชุดของข้อมูลที่เกี่ยวข้องกัน แถวหนึ่งแถวจะประกอบไปด้วยข้อมูลในคอลัมน์ทั้งหมด

คุณสมบัติเด่นของ PostgreSQL

- **ความเสถียร: PostgreSQL** เป็นระบบที่มีความเสถียรสูง เหมาะสำหรับการใช้งานที่ต้องการความน่าเชื่อถือ
- **ความปลอดภัย: PostgreSQL** มีระบบความปลอดภัยที่แข็งแกร่ง ช่วยป้องกันข้อมูลจากการถูกโจรกรรม
- **ความยืดหยุ่น: PostgreSQL** สามารถปรับแต่งให้เหมาะกับการใช้งานที่หลากหลาย
- **ความสามารถในการปรับขนาด: PostgreSQL** สามารถรองรับการใช้งานขนาดใหญ่
- **ฟีเจอร์: PostgreSQL** มีฟีเจอร์มากมาย เช่น การรองรับ ACID transactions, การสำรองข้อมูล, การกระจายข้อมูล

เหตุใด PostgreSQL จึงเป็นระบบฐานข้อมูลอันดับ 1

DB-ENGINES

"Pipelining Vector-Based Statistical Functions for In-Memory Analytics" Get the free white paper

English
Deutsch

Knowledge Base of Relational and NoSQL Database Management Systems

provided by solid IT

Home | DB-Engines Ranking | Systems | Encyclopedia | Blog | Sponsors | Search | Vendor Login

Featured Products: Milvus Datastax Astra Neo4j SingleStore Ontotext

Blog > Post

RSS Feed Tags
Want to publish an article here?

PostgreSQL is the DBMS of the Year 2023

by Matthias Gelbmann, Paul Andlinger, 2 January 2024
Tags: Databricks, DBMS of the year, Google BigQuery, PostgreSQL

> *PostgreSQL is the database management system that gained more popularity in our DB-Engines Ranking within the last year than any of the other 417 monitored systems.*
> *We thus declare* **PostgreSQL** *as the* **DBMS of the Year 2023**.

For determining the DBMS of the year, we subtracted the popularity scores of January 2023 from the latest scores of January 2024. We use the difference of these numbers, rather than a percentage, because that would favor systems with a tiny popularity at the beginning of the year. The result is a list of DBMSs sorted by how much they managed to increase their popularity in 2023, or in other words, how many additional people started to communicate about it in one of the ways we measure in our methodology, for example job offers, professional profile entries and citations on the web.

DBMS of the Year: PostgreSQL

Featured Products

ontotext
GraphDB allows you to link diverse data, index it for semantic search and enrich it via text analysis to build big knowledge graphs. Get it free.

milvus
The open source vector database for GenAI.
Try Managed Milvus Free

SingleStore
Build AI apps with Vectors on SQL and JSON with milliseconds response times.
Try it today.

DATASTAX

เหตุใด PostgreSQL จึงเป็นระบบฐานข้อมูลอันดับ 1

Previous winners of the DB-Engines DBMS of the Year Award:

PostgreSQL	2023
Snowflake	2022
Snowflake	2021
PostgreSQL	2020
MySQL	2019
PostgreSQL	2018
PostgreSQL	2017
Microsoft SQL Server	2016
Oracle	2015
MongoDB	2014
MongoDB	2013

ความนิยมในตัว PostgreSQL

Select a ranking

- Complete ranking
- Relational DBMS
- Key-value stores
- Document stores
- Time Series DBMS
- Graph DBMS
- Search engines
- RDF stores
- Object oriented DBMS
- Vector DBMS
- Wide column stores
- Multivalue DBMS
- Spatial DBMS
- Native XML DBMS
- Event Stores
- Content stores
- Navigational DBMS

Special reports

- Ranking by database model
- Open source vs. commercial

Featured Products

ontotext

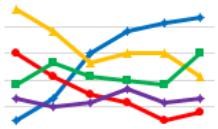
GraphDB allows you to link diverse data, index it for semantic search and enrich it via text analysis to build big knowledge graphs. Get it free.

Ranking > Complete Ranking https://db-engines.com/en/ranking RSS Feed

DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the method of calculating the scores.



trend chart

417 systems in ranking, February 2024

Rank			DBMS	Database Model	Score		
Feb 2024	Jan 2024	Feb 2023			Feb 2024	Jan 2024	Feb 2023
1.	1.	1.	Oracle	Relational, Multi-model	1241.45	-6.05	-6.08
2.	2.	2.	MySQL	Relational, Multi-model	1106.67	-16.79	-88.78
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	853.57	-23.03	-75.52
4.	4.	4.	PostgreSQL	Relational, Multi-model	629.41	-19.55	+12.90
5.	5.	5.	MongoDB	Document, Multi-model	420.36	+2.88	-32.41
6.	6.	6.	Redis	Key-value, Multi-model	160.71	+1.33	-13.12
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	135.74	-0.33	-2.86
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model	132.23	-0.18	-10.74
9.	9.	↑ 12.	Snowflake	Relational	127.45	+1.53	+11.80
10.	↑ 11.	↓ 9.	SQLite	Relational	117.28	+2.08	-15.38
11.	↓ 10.	↓ 10.	Microsoft Access	Relational	113.17	-4.50	-17.86
12.	12.	↓ 11.	Cassandra	Wide column, Multi-model	109.27	-1.77	-6.95
13.	13.	13.	MariaDB	Relational, Multi-model	97.23	-2.00	+0.42
14.	14.	14.	Splunk	Search engine	91.65	-1.07	+4.57
15.	↑ 16.	15.	Amazon DynamoDB	Multi-model	82.90	+1.96	+3.21
16.	↓ 15.	16.	Microsoft Azure SQL Database	Relational, Multi-model	79.56	-1.51	+0.81
17.	17.	↑ 19.	Databricks	Multi-model	76.91	-3.62	+16.58
18.	18.	↓ 17.	Hive	Relational	65.81	-1.15	-6.31
19.	19.	↑ 22.	Google BigQuery	Relational	63.63	+0.15	+11.17
20.	20.	↓ 18.	Teradata	Relational, Multi-model	51.24	-1.94	-11.79

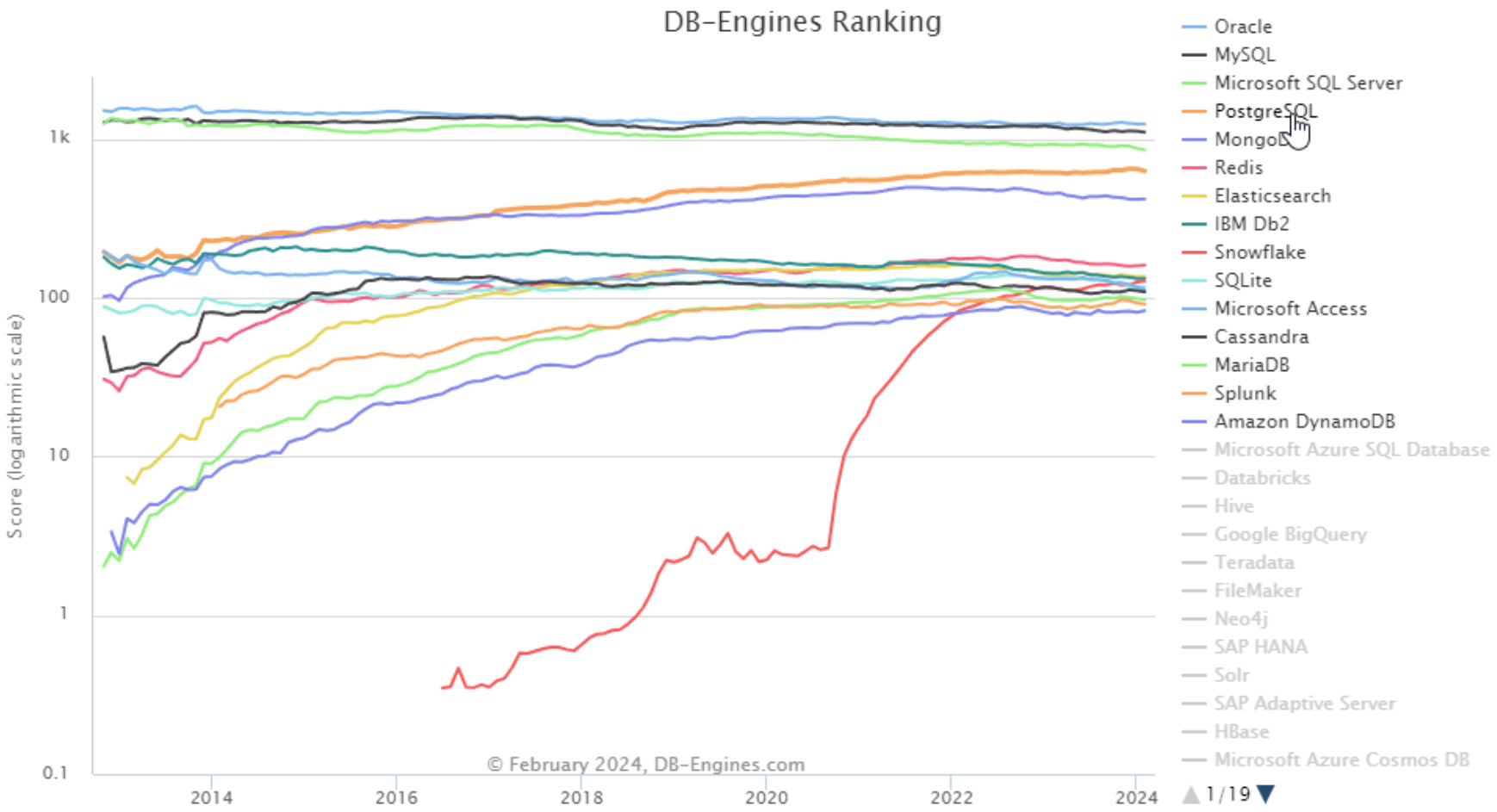
DB-Engines Ranking - Trend Popularity

The DB-Engines Ranking ranks database management systems according to their popularity.

Read more about the method of calculating the scores.

Rank	Trend	System	Score	Change
1		Oracle	1560	+ 27
2	↑	MySQL	1342	+ 47
3	↓	SQL Server	1278	- 40
4		PostgreSQL	174	- 3
5		MS Access	161	- 8
6		DB2	155	- 4

ranking table
February 2024



n/ranking

Click at a system in the legend
to hide or show its trend line

PostgreSQL System Properties

Please select another system to compare it with PostgreSQL.

Our visitors often compare PostgreSQL with MySQL, Oracle and MongoDB.

Editorial information provided by DB-Engines

Name	**PostgreSQL**
Description	Widely used open source RDBMS
Primary database model	Relational DBMS
Secondary database models	Document store Graph DBMS Spatial DBMS Vector DBMS
DB-Engines Ranking Trend Chart	Score 629.41 Rank #4 Overall #4 Relational DBMS
Website	www.postgresql.org
Technical documentation	www.postgresql.org/docs
Developer	PostgreSQL Global Development Group
Initial release	1989
Current release	16.2, February 2024
License	Open Source
Cloud-based only	no
DBaaS offerings (sponsored links)	Aiven for PostgreSQL: Fully managed PostgreSQL for developers with 70+ extensions and flexible orchestration tools.
Implementation language	C
Server operating systems	FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows

Data scheme	yes
Typing	yes
XML support	yes
Secondary indexes	yes
SQL	yes
APIs and other access methods	ADO.NET JDBC native C library ODBC streaming API for large objects
Supported programming languages	.Net C C++ Delphi Java JavaScript (Node.js) Perl PHP Python Tcl
Server-side scripts	user defined functions
Triggers	yes
Partitioning methods	partitioning by range, list and (since PostgreSQL 11) by hash
Replication methods	Source-replica replication
MapReduce	no
Consistency concepts	Immediate Consistency
Foreign keys	yes
Transaction concepts	ACID
Concurrency	yes
Durability	yes
In-memory capabilities	no
User concepts	fine grained access rights according to SQL-standard

PostgreSQL vs MySQL

PostgreSQL

Initial: 1989 (35 Years)
License: Open source by PostgreSQL Global Development Group
Implementation language: C
Supported programming languages: .Net, C, C++, Delphi, Java, Perl, PHP, Python, Tcl
Cloud Hosting: Amazon, Google, Microsoft, IBM รวมถึงเจ้าหลัก ๆ อื่น ๆ
Partitioning: declarative partitioning (by range or by list) since PostgreSQL 10.0
Replication: PostgreSQL ทำ master-slave ได้
Single master to one standby, Single master to multiple standbys, Hot Standby/Streaming Replication, Bi-Directional replication, Logical log streaming replication, Cascading replication

MySQL

Initial: 1995 (29 Years)
License: Open source ที่ถูก Oracle ซื้อไปแล้ว (ยังฟรีอยู่)
Implementation language: C and C++
Supported programming languages: Ada, C, C#, C++, D, Delphi, Eiffel, Erlang, Haskell, Java, JavaScript (Node.js), Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme, Tcl
Cloud Hosting: Amazon, Google, Microsoft, IBM รวมถึงเจ้าหลัก ๆ อื่น ๆ
Partitioning: horizontal partitioning, sharding with MySQL Cluster or MySQL Fabric
Replication: MySQL ทำ master-master และ master-slave ได้
Single master to one standby, Single master to multiple standbys, Single master to one standby to one or more standbys, Circular replication (A to B to C and back to A), Master to master

Concurrency

การรองรับ concurrency เป็นหนึ่งในคุณสมบัติที่เราจำเป็นต้องพิจารณา เมื่อเราจะสร้างระบบขึ้นมาเพื่อให้รองรับ user ที่จะเข้ามา access ข้อมูลออกไปพร้อม ๆ กันจำนวนมาก

“PostgreSQL มี Multi-version concurrency control (MVCC) ในขณะที่ MySQL จะใช้ MVCC ได้เฉพาะใน innodb เท่านั้น”

NoSQL Feature / JSON support

PostgreSQL รองรับ JSON และ สามารถจัดการพวก NoSQL ได้ รวมถึง รองรับ native xml สามารถใช้ key-value เทียบค่า กับ HSTORE แล้วยัง สามารถทำ index ของ JSON ในการเข้าถึงข้อมูลได้

MySQL รองรับข้อมูล JSON ได้ แต่ไม่รองรับประเภท NoSQL และ ไม่สามารถ ทำ index บน JSON ได้

Performance

PostgreSQL ใช้งานกันอย่างแพร่หลายในระบบขนาดใหญ่ โดยมีความเร็วทั้งในการอ่าน และ เขียน ซึ่งจะเทพกว่านี้อีกหากใช้แบบเชิงพาณิชย์ เช่น การใช้งานพร้อม ๆ กันมาก ๆ โดยที่เขาเคลมว่า “จะไม่มีอาการ LOCK ของ database...” ซึ่งเขายังเคลมต่อไปอีกว่า ถ้าจะรีดประสิทธิภาพโดยรวมของ PostgreSQL เนี่ยต้องใช้การค้นหาที่ซับซ้อน ยังไม่พอเขาเคลมอีกหน่อยว่า PostgreSQL จะทำงานได้ดีในระบบ OLTP/OLAP เมื่อจำเป็นต้องอ่าน หรือ เขียน และ เมื่อมีการวิเคราะห์ข้อมูลอย่างละเอียด สุดท้าย PostgreSQL ทำงานได้ดีกับ application ที่เป็นพวก Business Intelligence เหมาะกับพวก Data Warehousing และ application การวิเคราะห์ข้อมูลที่ต้องการความเร็วในการ อ่าน และ เขียน

Performance

MySQL ใช้งานกันอย่างแพร่หลายเช่นกัน สำหรับเว็บไซต์ที่ต้องการฐานข้อมูลเพียงเพื่อการทำธุรกรรม ข้อมูลที่ไม่ซับซ้อนมาก ซึ่งเมื่อต้องการจะค้นหาข้อมูลที่ซับซ้อนประสิทธิภาพจะลดลงไป... นอกจากนี้ MySQL ก็ยังโดนบลัฟไปอีกว่า ในระบบ OLAP / OLTP MySQL จะทำงานได้ดีก็ต่อเมื่อต้องการ “READ” เท่านั้น...
เมื่อรวมร่างระหว่าง MySQL กับ InnoDB จะทำให้มีความเร็วในการอ่าน / เขียนที่ดีมากสำหรับ OLTP
โดยรวมแล้ว MySQL สามารถทำงานได้ดีกับสถานการณ์ที่มี concurrency สูง
อย่างไรก็ดี MySQL ก็มีความน่าเชื่อถือ และ ทำงานได้ดีกับ application Business Intelligence เนื่องจาก application Business Intelligence มักจะใช้ “READ”

Security

PostgreSQL ใช้ SSL ในการเชื่อมต่อกับ client/server นอกจากนั้นยังมีการเพิ่มประสิทธิภาพความปลอดภัยในตัวเรียกว่า SE-PostgreSQL ซึ่งเป็นการควบคุมการเข้าถึงตามหลักการความปลอดภัยของ SELinux

MySQL ใช้ Access Control Lists (ACLs) สำหรับการเชื่อมต่อ และ ตอนนี้ MySQL ก็รองรับการเชื่อมต่อแบบ SSL แล้ว

ตัวอย่างการใช้งาน PostgreSQL ในโลกจริง

เวอร์ชั่นของ PostgreSQL และการ Support

Version	Current minor	Supported	First Release	Final Release
17	17.2	Yes	September 26, 2024	November 8, 2029
16	16.6	Yes	September 14, 2023	November 9, 2028
15	15.10	Yes	October 13, 2022	November 11, 2027
14	14.15	Yes	September 30, 2021	November 12, 2026
13	13.18	Yes	September 24, 2020	November 13, 2025
12	12.22	No	October 3, 2019	November 21, 2024
11	11.22	No	October 18, 2018	November 9, 2023
10	10.23	No	October 5, 2017	November 10, 2022
9.6	9.6.24	No	September 29, 2016	November 11, 2021
9.5	9.5.25	No	January 7, 2016	February 11, 2021
9.4	9.4.26	No	December 18, 2014	February 13, 2020

https://www.postgresql.org/support/versioning/

Section 2

ติดตั้ง PostgreSQL 16.x บน Windows & macOS

- ดาวน์โหลดและติดตั้ง PostgreSQL
- ตั้งค่าการใช้งาน PostgreSQL
- เชื่อมต่อกับ PostgreSQL
- จัดการผู้ใช้และฐานข้อมูล

การเตรียมเครื่องมือ

โปรแกรม (Tool and Editor) ที่ใช้อบรม

1. PostgreSQL 16.x
2. Visual Studio Code
3. Docker Desktop

PostgreSQL

ติดตั้ง PostgreSQL 16.x

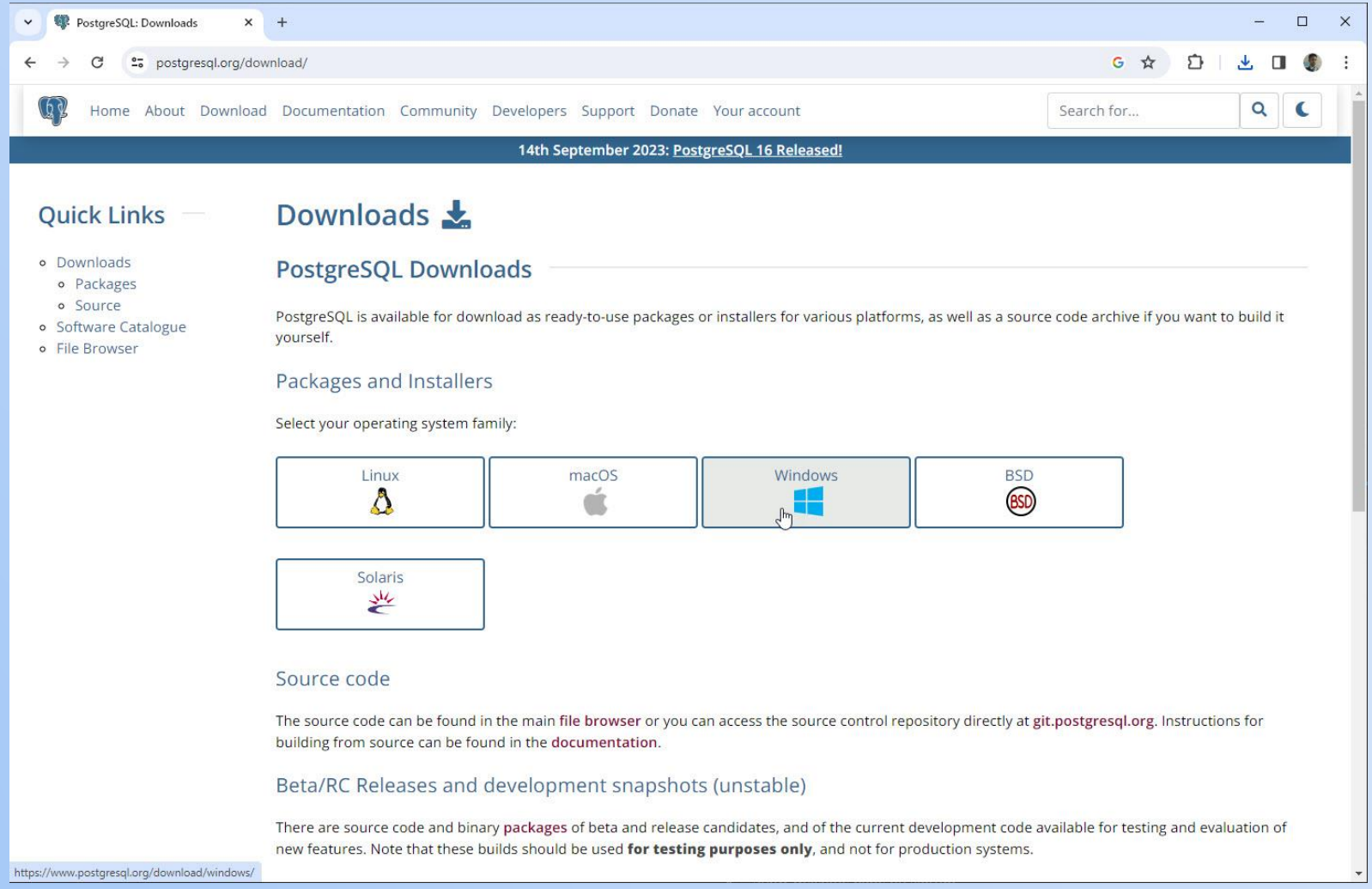
ติดตั้ง PostgreSQL 16.x บน Windows

- Download PostgreSQL 16.x installer for Windows
- Install PostgreSQL
- Verify the installation

Download PostgreSQL for Windows

Windows installers

https://www.postgresql.org/download/windows/



Windows installers

https://www.postgresql.org/download/windows/

PostgreSQL: Windows installer

postgresql.org/download/windows/

Home About Download Documentation Community Developers Support Donate Your account

Search for...

14th September 2023: PostgreSQL 16 Released!

Quick Links

- Downloads
  - Packages
  - Source
- Software Catalogue
- File Browser

Windows installers

Interactive installer by EDB

Download the installer certified by EDB for all supported PostgreSQL versions.

Note! This installer is hosted by EDB and not on the PostgreSQL community servers. If you have issues with the website it's hosted on, please contact webmaster@enterprisedb.com.

This installer includes the PostgreSQL server, pgAdmin; a graphical tool for managing and developing your databases, and StackBuilder; a package manager that can be used to download and install additional PostgreSQL tools and drivers. Stackbuilder includes management, integration, migration, replication, geospatial, connectors and other tools.

This installer can run in graphical or silent install modes.

The installer is designed to be a straightforward, fast way to get up and running with PostgreSQL on Windows.

Advanced users can also download a zip archive of the binaries, without the installer. This download is intended for users who wish to include PostgreSQL as part of another application installer.

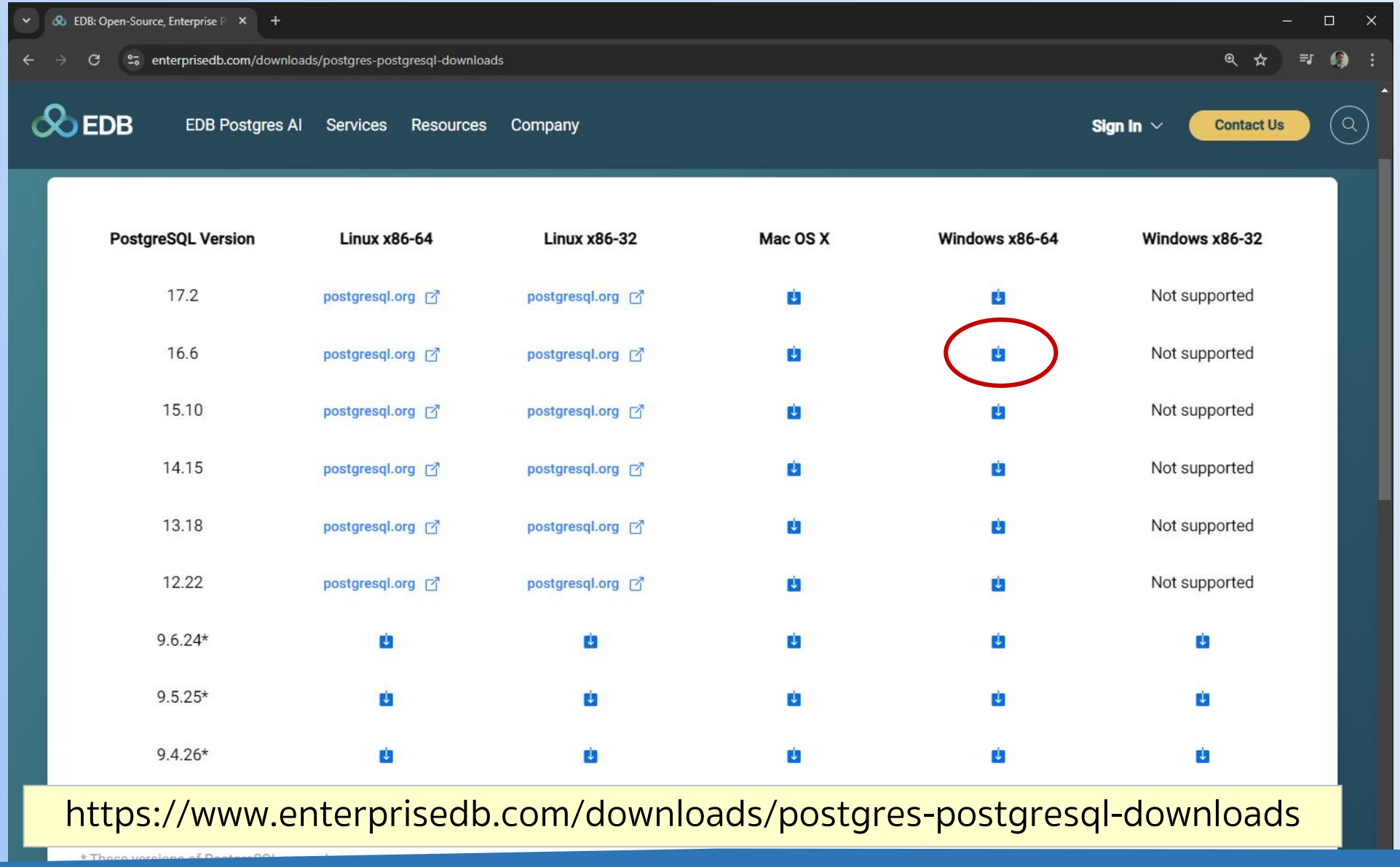
Platform support

The installers are tested by EDB on the following platforms. They can generally be expected to run on other comparable versions, for example, desktop releases of Windows:

PostgreSQL Version	64 Bit Windows Platforms	32 Bit Windows Platforms
16	2022, 2019	
15	2019, 2016	
14	2019, 2016	
13	2019, 2016	

https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

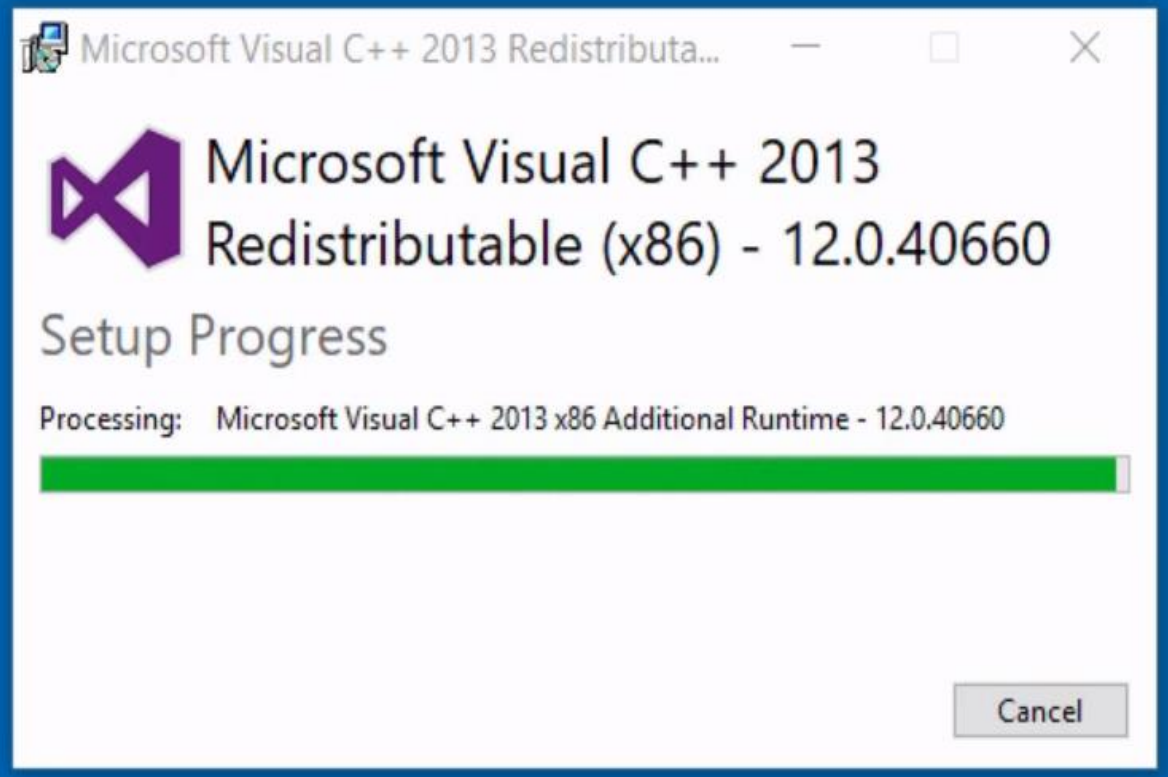
ดาวน์โหลดชุดติดตั้ง PostgreSQL 16.x สำหรับ Windows

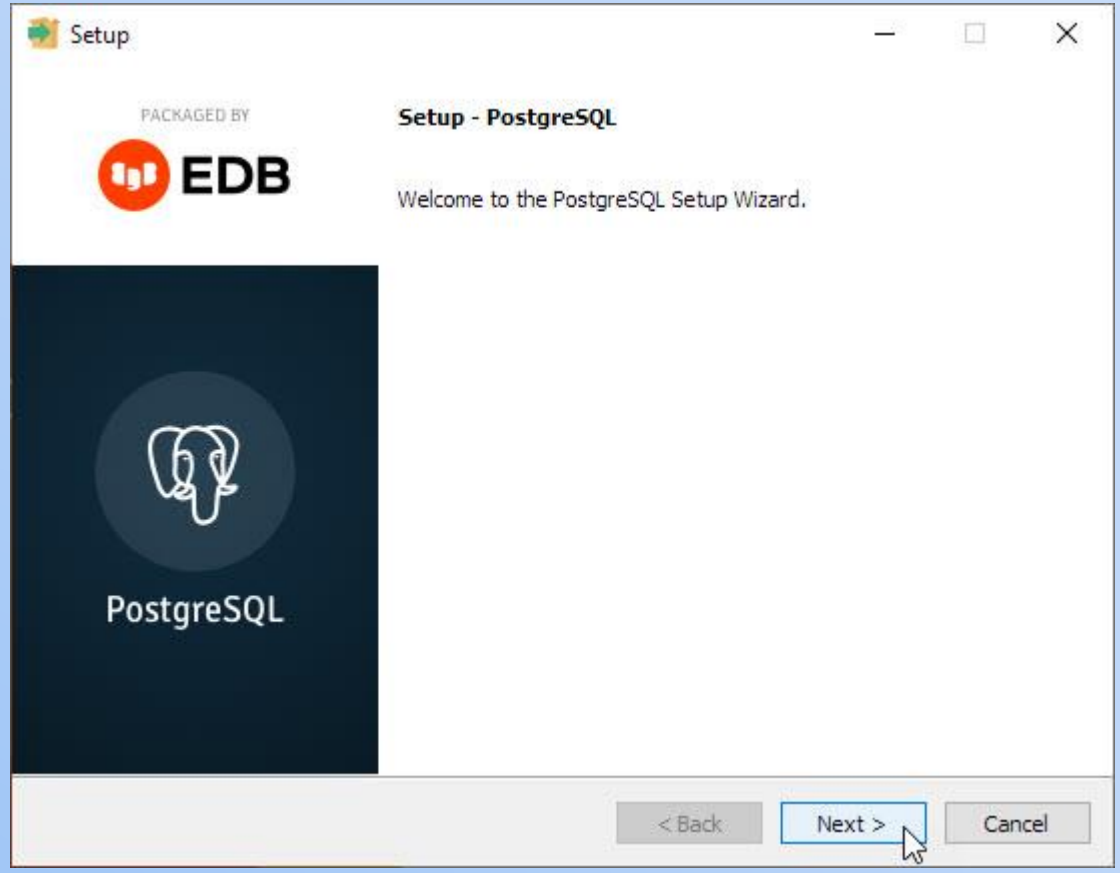


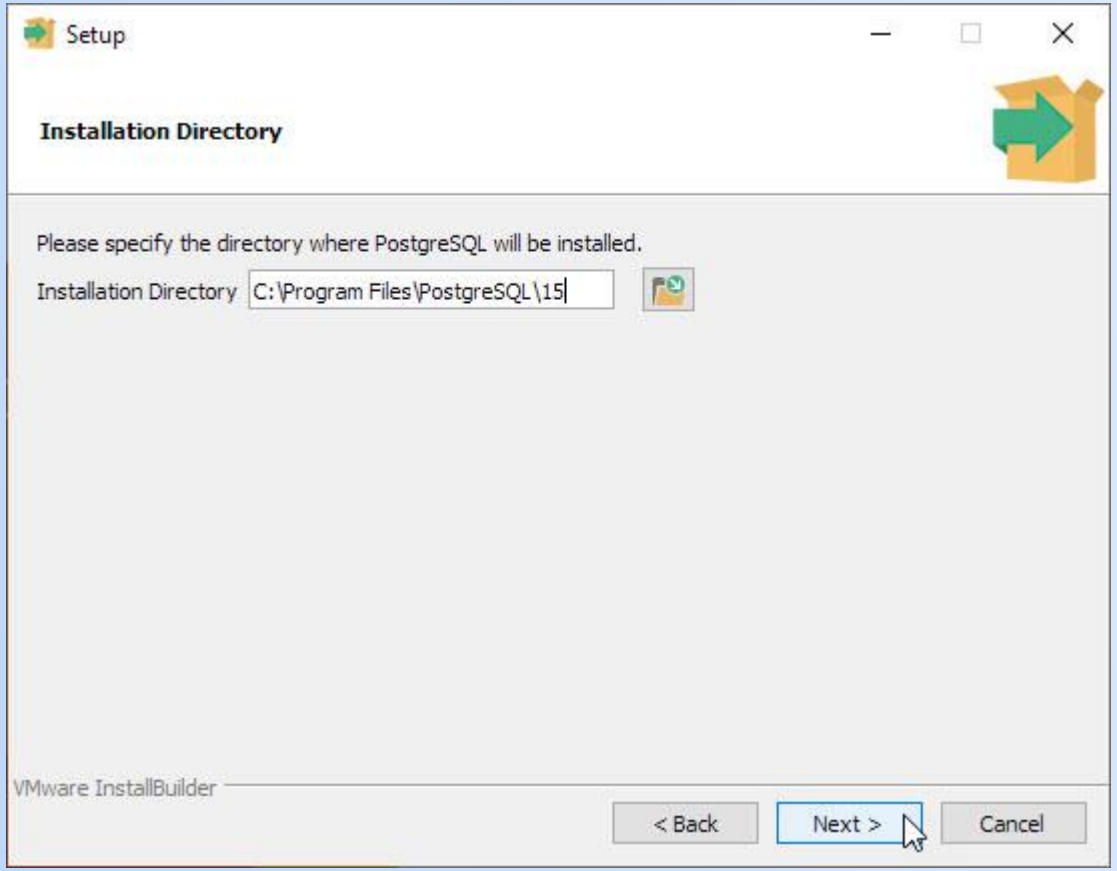
PostgreSQL Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
17.2	postgresql.org	postgresql.org			Not supported
16.6	postgresql.org	postgresql.org			Not supported
15.10	postgresql.org	postgresql.org			Not supported
14.15	postgresql.org	postgresql.org			Not supported
13.18	postgresql.org	postgresql.org			Not supported
12.22	postgresql.org	postgresql.org			Not supported
9.6.24*					
9.5.25*					
9.4.26*					

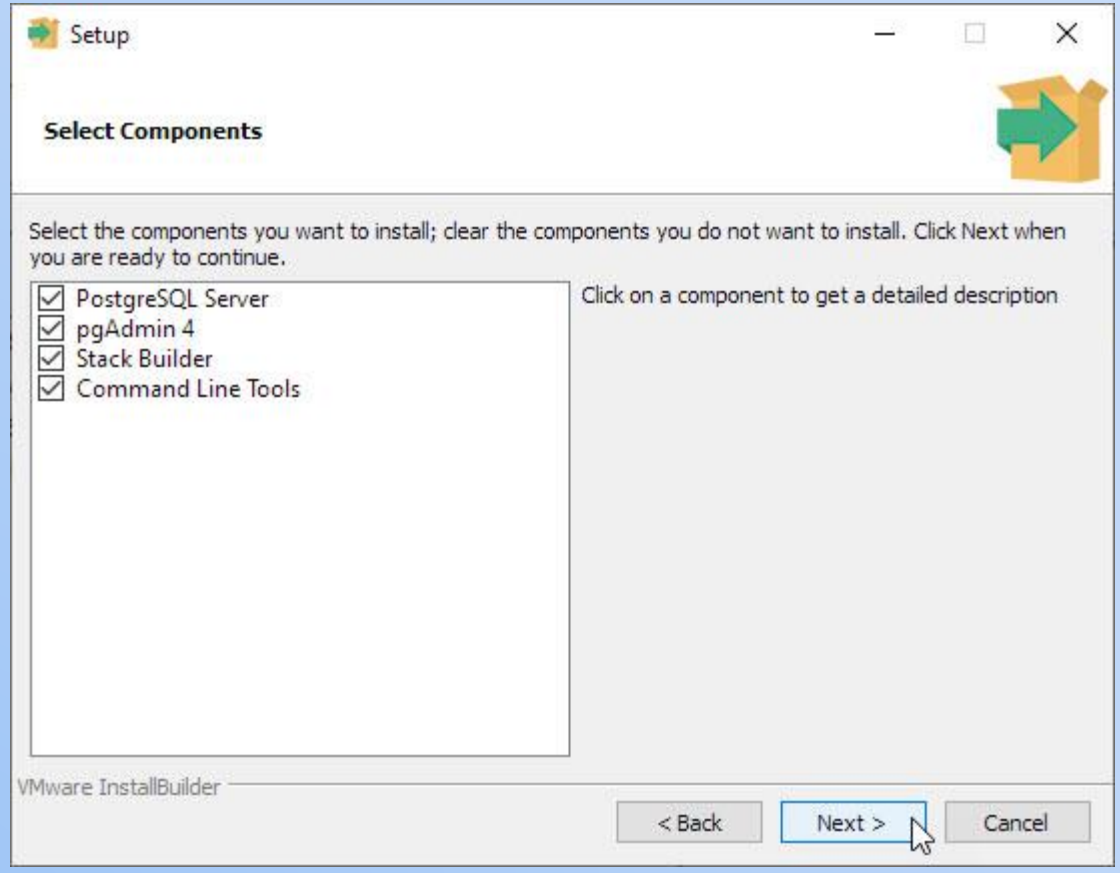
https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

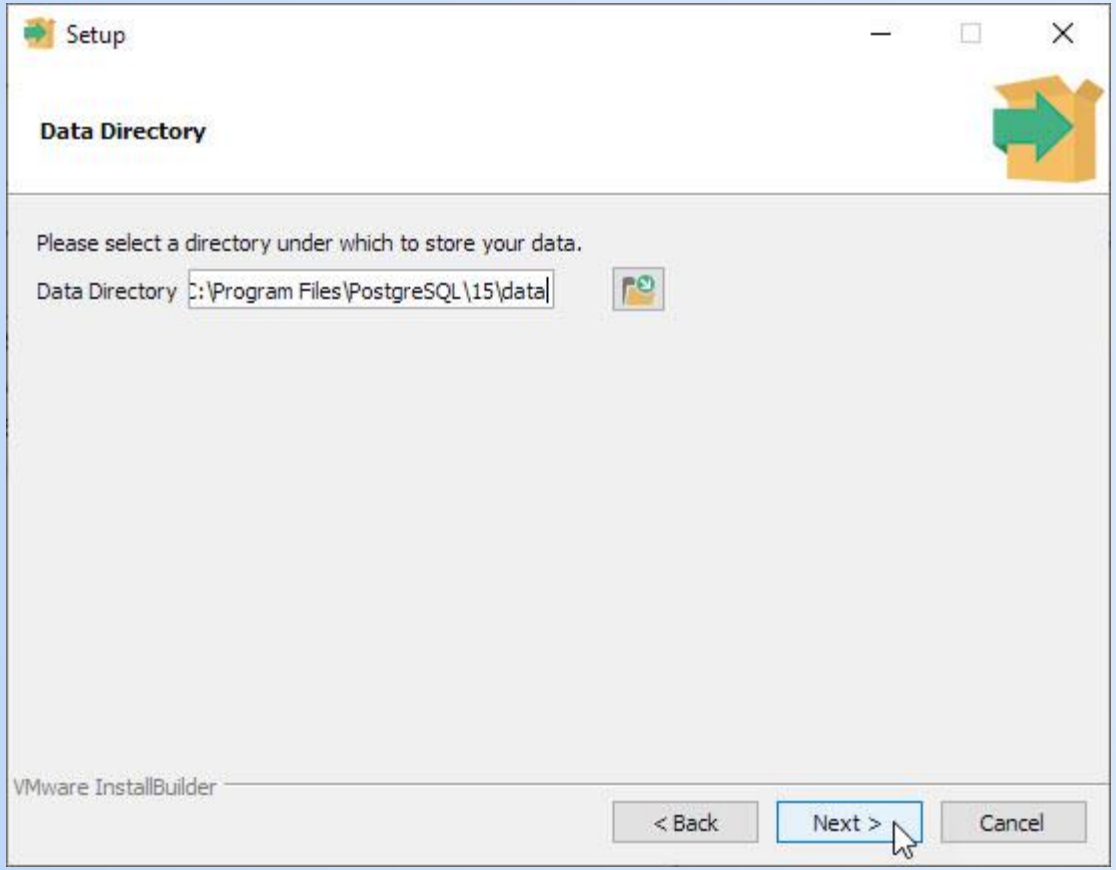
Installing PostgreSQL 16.x

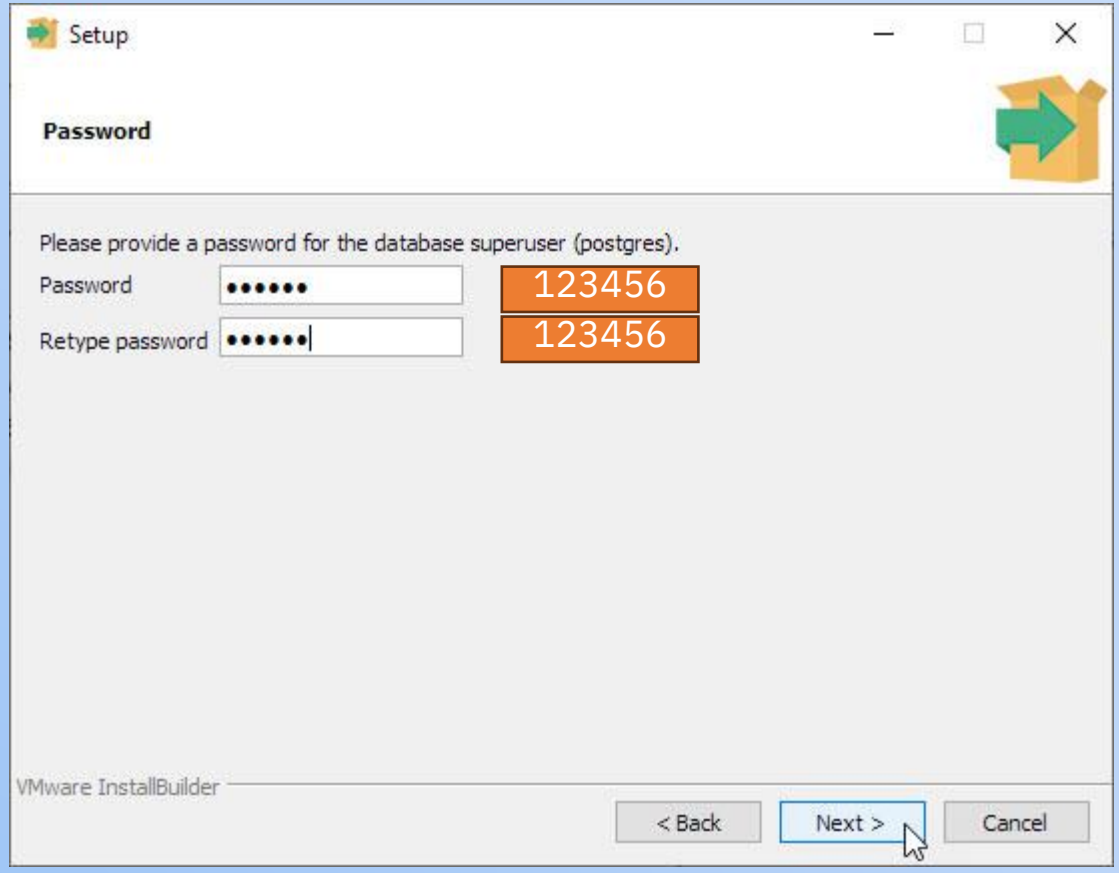

Microsoft Visual C++ 2013 Redistributa...
Microsoft Visual C++ 2013 Redistributable (x86) - 12.0.40660
Setup Progress
Processing: Microsoft Visual C++ 2013 x86 Additional Runtime - 12.0.40660
Cancel

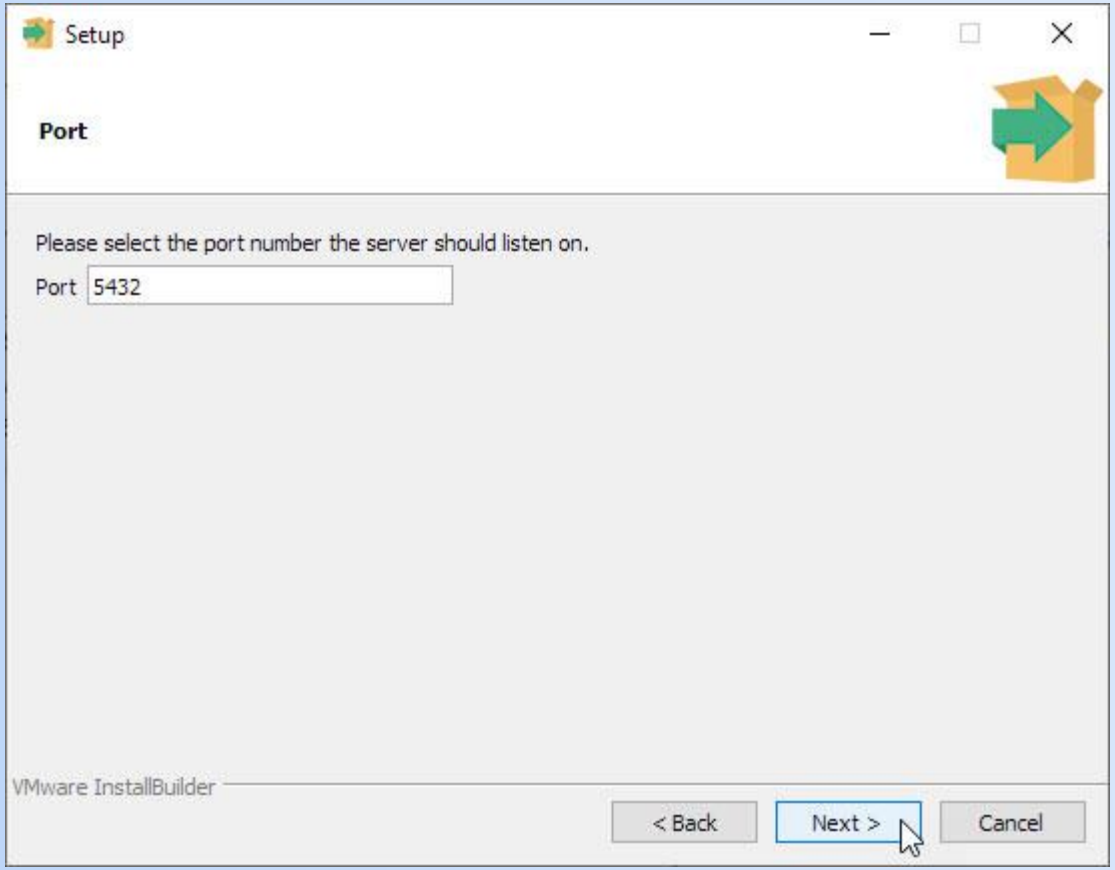

Setup
PACKAGED BY
EDB
Setup - PostgreSQL
Welcome to the PostgreSQL Setup Wizard.
PostgreSQL
< Back
Next >
Cancel

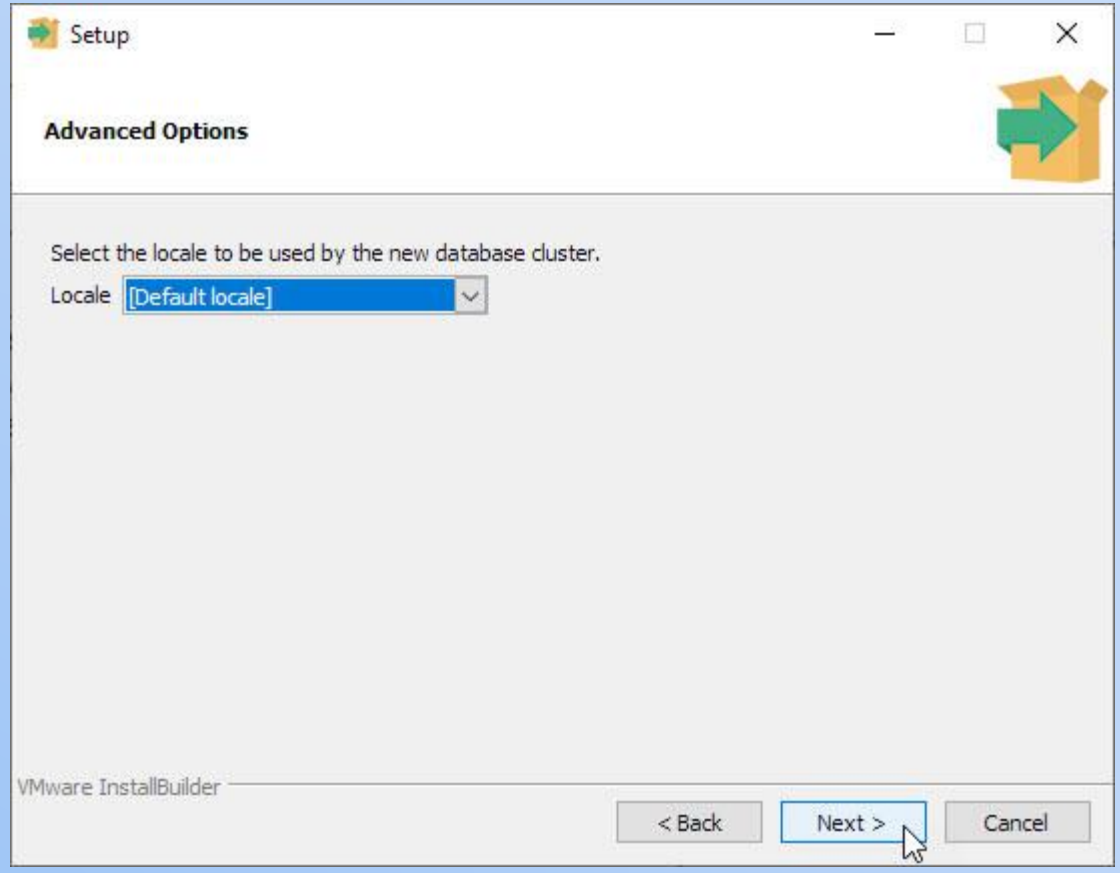

Setup
Installation Directory
Please specify the directory where PostgreSQL will be installed.
Installation Directory
C:\Program Files\PostgreSQL\15
VMware InstallBuilder
< Back
Next >
Cancel

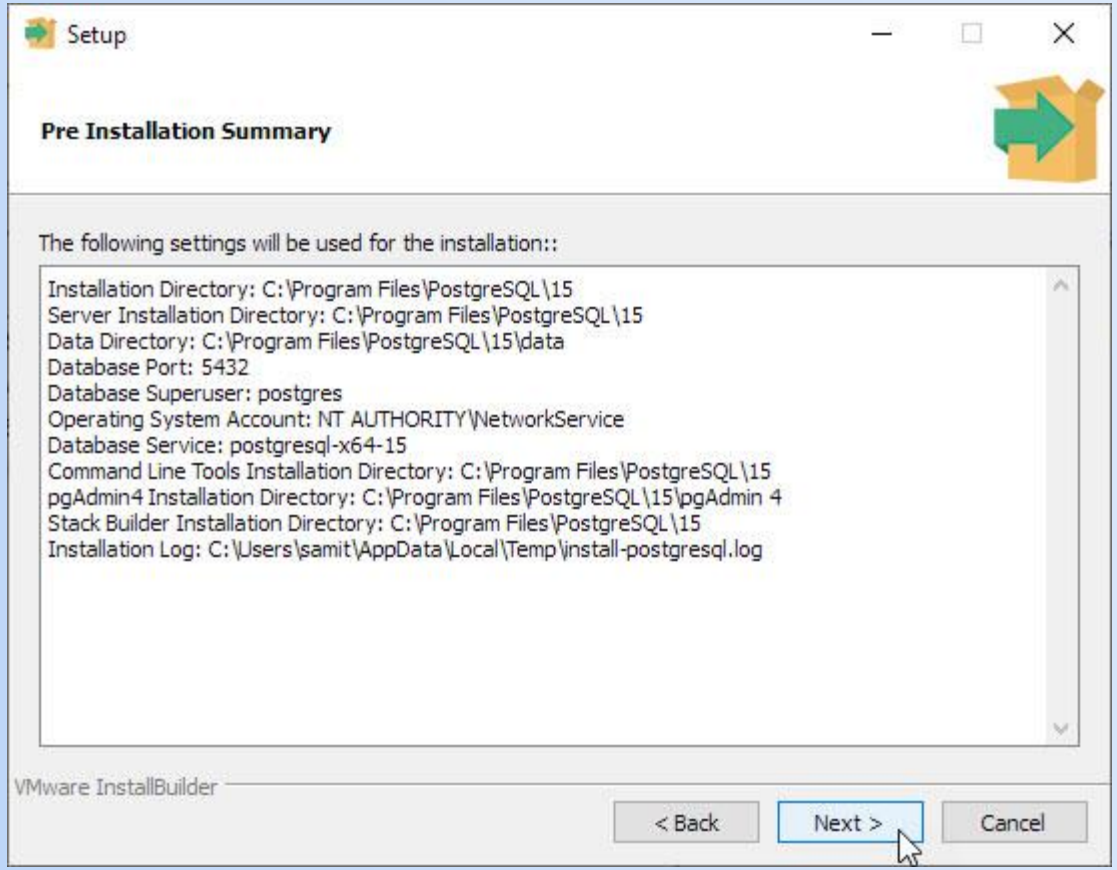

Setup
Select Components
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
PostgreSQL Server
pgAdmin 4
Stack Builder
Command Line Tools
Click on a component to get a detailed description
VMware InstallBuilder
< Back
Next >
Cancel

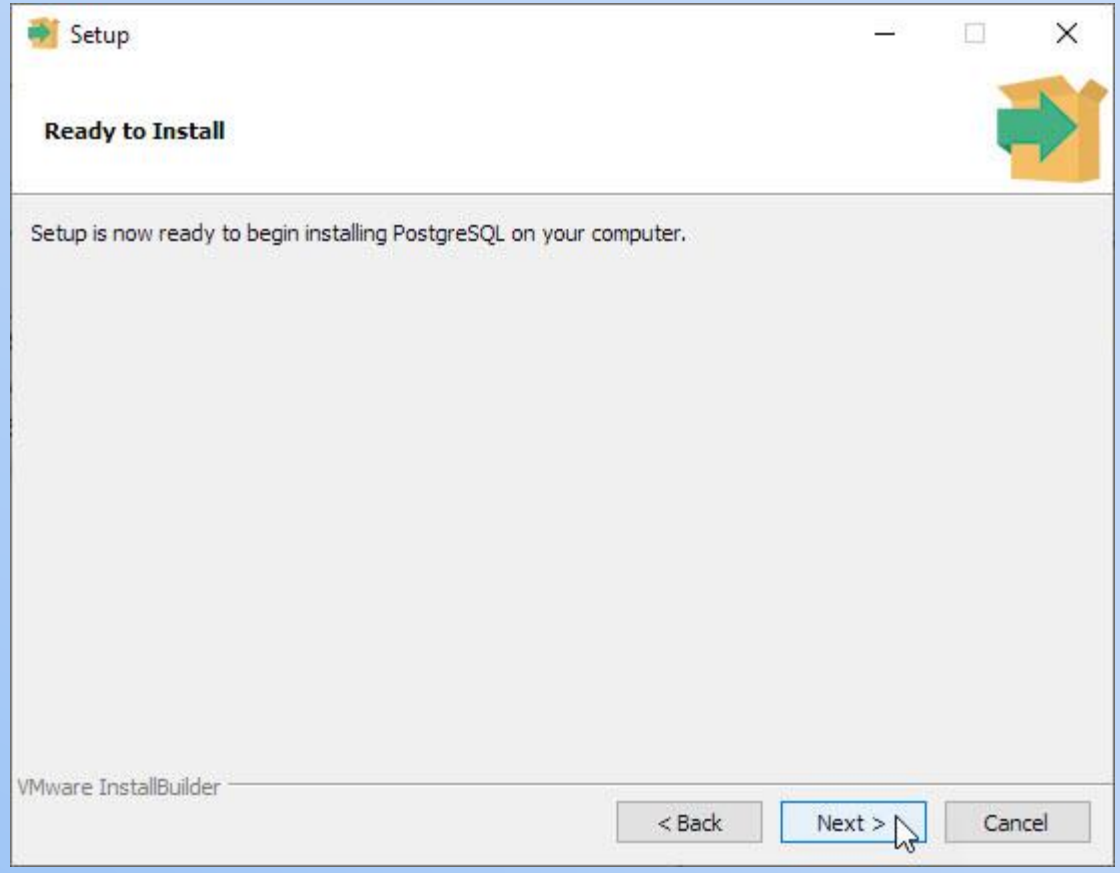

Setup
Data Directory
Please select a directory under which to store your data.
Data Directory
C:\Program Files\PostgreSQL\15\data
VMware InstallBuilder
< Back
Next >
Cancel

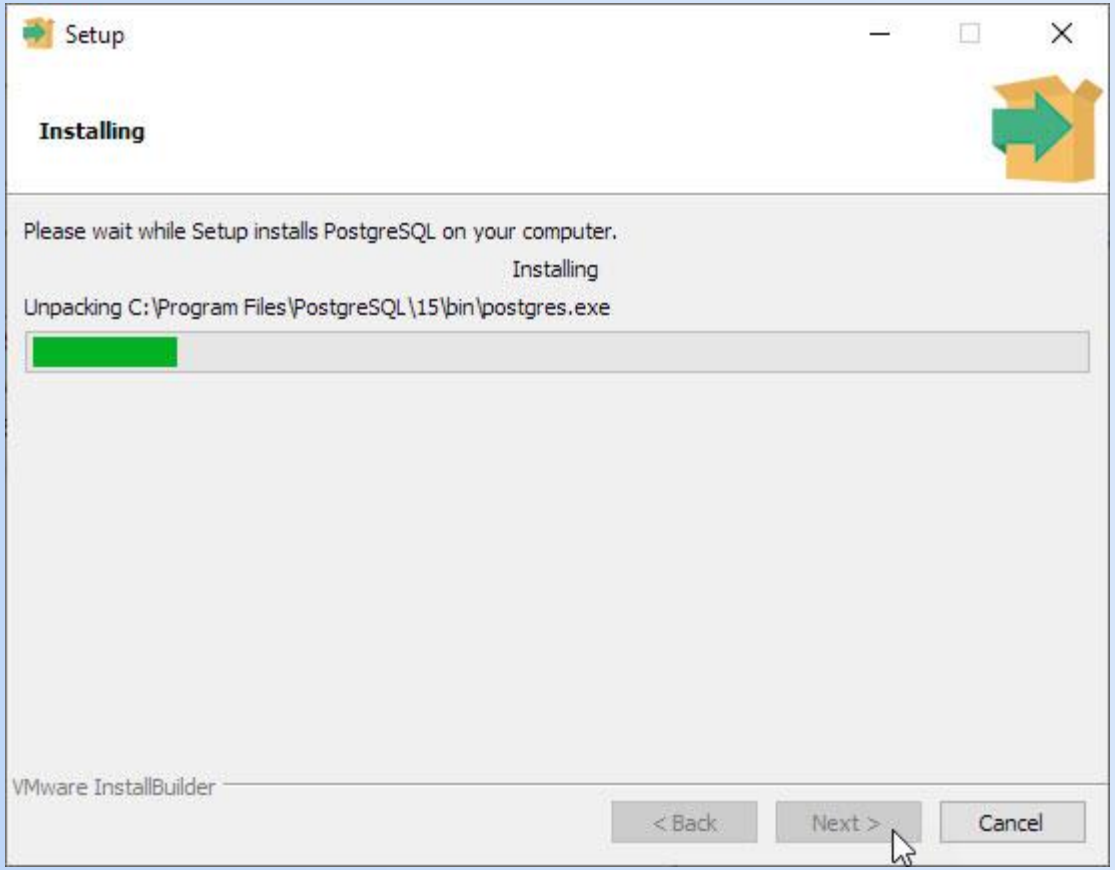

Setup
Password
Please provide a password for the database superuser (postgres).
Password
Retype password
123456
123456
VMware InstallBuilder
< Back
Next >
Cancel

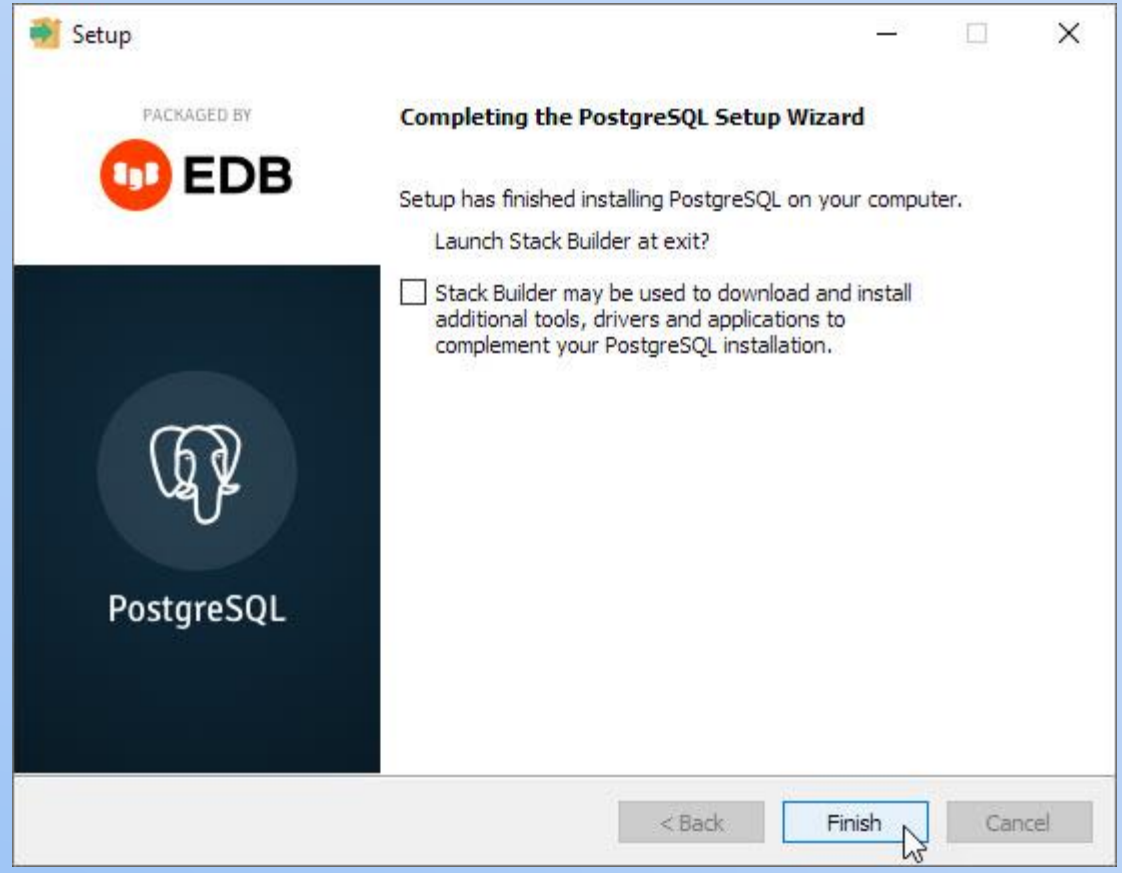

Setup
Port
Please select the port number the server should listen on.
Port
5432
VMware InstallBuilder
< Back
Next >
Cancel

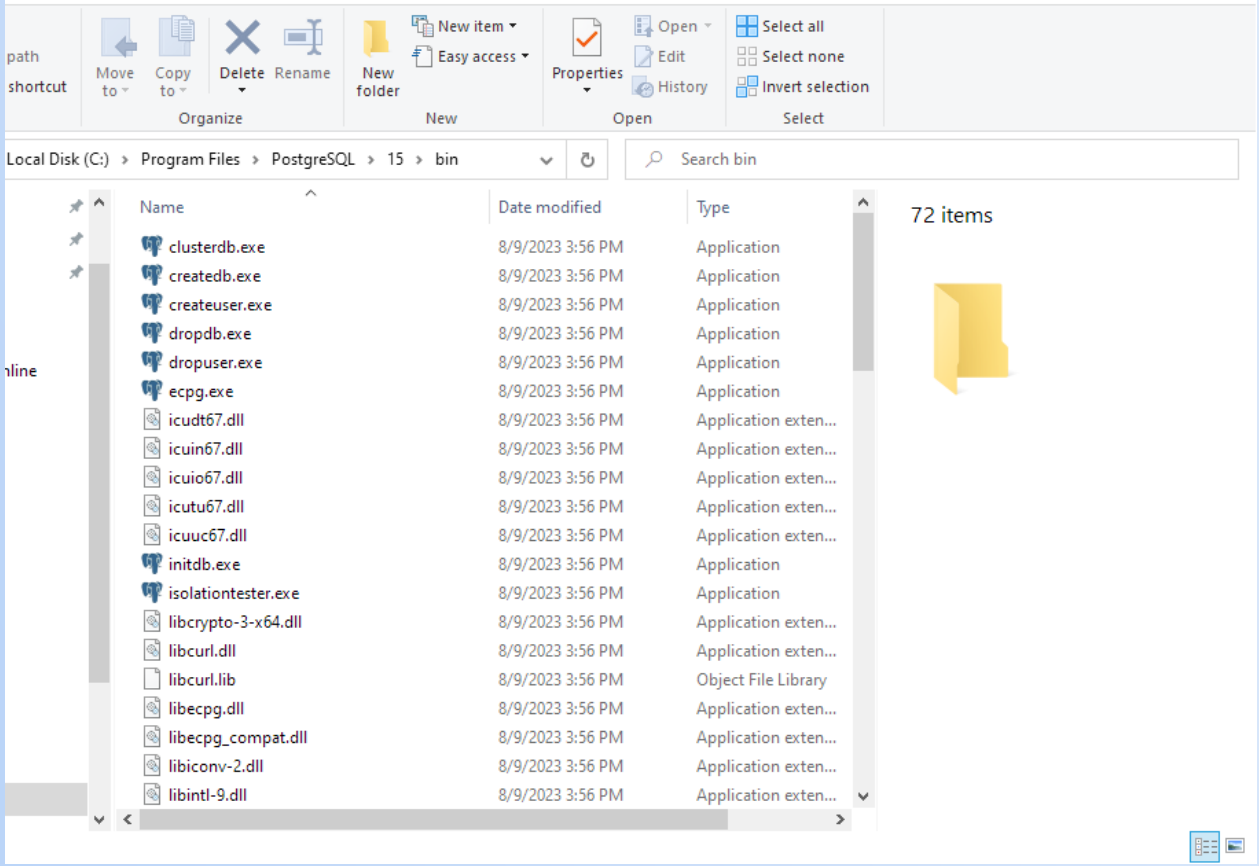

Setup
Advanced Options
Select the locale to be used by the new database cluster.
Locale [Default locale]
VMware InstallBuilder
< Back
Next >
Cancel


Setup
Pre Installation Summary
The following settings will be used for the installation::
Installation Directory: C:\Program Files\PostgreSQL\15
Server Installation Directory: C:\Program Files\PostgreSQL\15
Data Directory: C:\Program Files\PostgreSQL\15\data
Database Port: 5432
Database Superuser: postgres
Operating System Account: NT AUTHORITY\NetworkService
Database Service: postgresql-x64-15
Command Line Tools Installation Directory: C:\Program Files\PostgreSQL\15
pgAdmin4 Installation Directory: C:\Program Files\PostgreSQL\15\pgAdmin 4
Stack Builder Installation Directory: C:\Program Files\PostgreSQL\15
Installation Log: C:\Users\samit\AppData\Local\Temp\install-postgresql.log
VMware InstallBuilder
< Back
Next >
Cancel


Setup
Ready to Install
Setup is now ready to begin installing PostgreSQL on your computer.
VMware InstallBuilder
< Back
Next >
Cancel

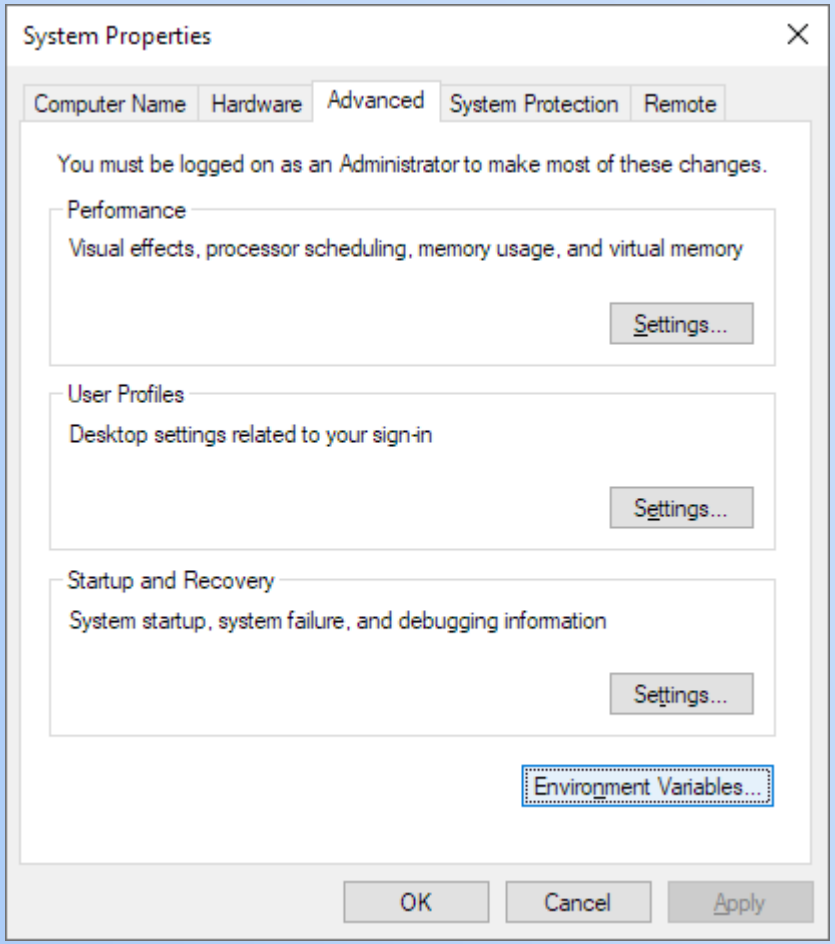

Setup
Installing
Please wait while Setup installs PostgreSQL on your computer.
Installing
Unpacking C:\Program Files\PostgreSQL\15\bin\postgres.exe
VMware InstallBuilder
< Back
Next >
Cancel


Setup
PACKAGED BY
EDB
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.
PostgreSQL
< Back
Finish
Cancel


Move to
Copy to
Delete
Rename
New folder
New item
Easy access
Properties
Open
Edit
History
Select all
Select none
Invert selection
Organize
New
Open
Select
Local Disk (C:) › Program Files › PostgreSQL › 15 › bin
Search bin
Name
Date modified
Type
clusterdb.exe 8/9/2023 3:56 PM Application
createdb.exe 8/9/2023 3:56 PM Application
createuser.exe 8/9/2023 3:56 PM Application
dropdb.exe 8/9/2023 3:56 PM Application
dropuser.exe 8/9/2023 3:56 PM Application
ecpg.exe 8/9/2023 3:56 PM Application
icudt67.dll 8/9/2023 3:56 PM Application exten...
icuin67.dll 8/9/2023 3:56 PM Application exten...
icuio67.dll 8/9/2023 3:56 PM Application exten...
icutu67.dll 8/9/2023 3:56 PM Application exten...
icuuc67.dll 8/9/2023 3:56 PM Application exten...
initdb.exe 8/9/2023 3:56 PM Application
isolationtester.exe 8/9/2023 3:56 PM Application
libcrypto-3-x64.dll 8/9/2023 3:56 PM Application exten...
libcurl.dll 8/9/2023 3:56 PM Application exten...
libcurl.lib 8/9/2023 3:56 PM Object File Library
libecpg.dll 8/9/2023 3:56 PM Application exten...
libecpg_compat.dll 8/9/2023 3:56 PM Application exten...
libiconv-2.dll 8/9/2023 3:56 PM Application exten...
libintl-9.dll 8/9/2023 3:56 PM Application exten...
72 items

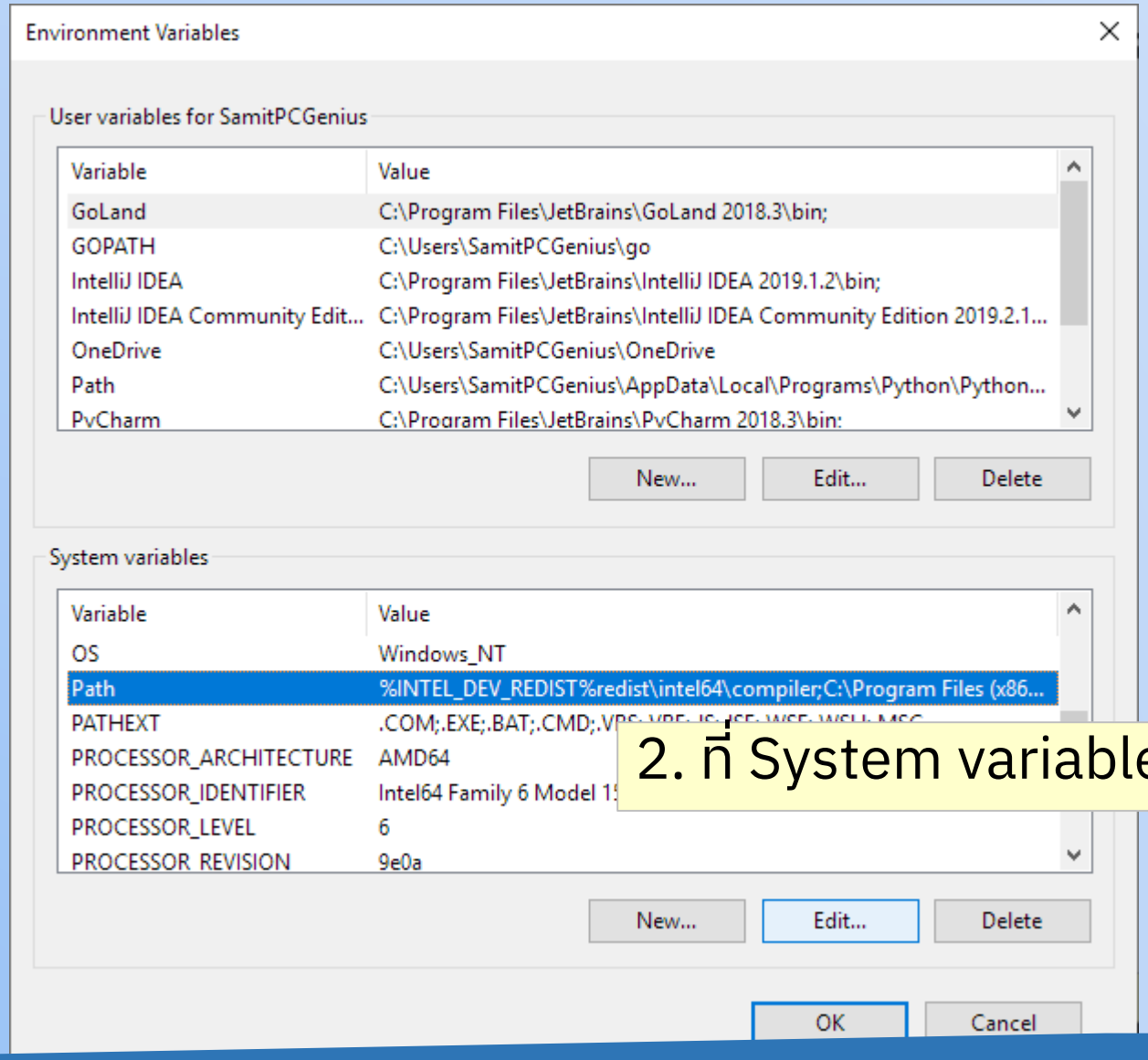
การกำหนด Path ของ PostgreSQL ให้กับ Windows

การกำหนด Path ของ PostgreSQL ให้กับ Windows



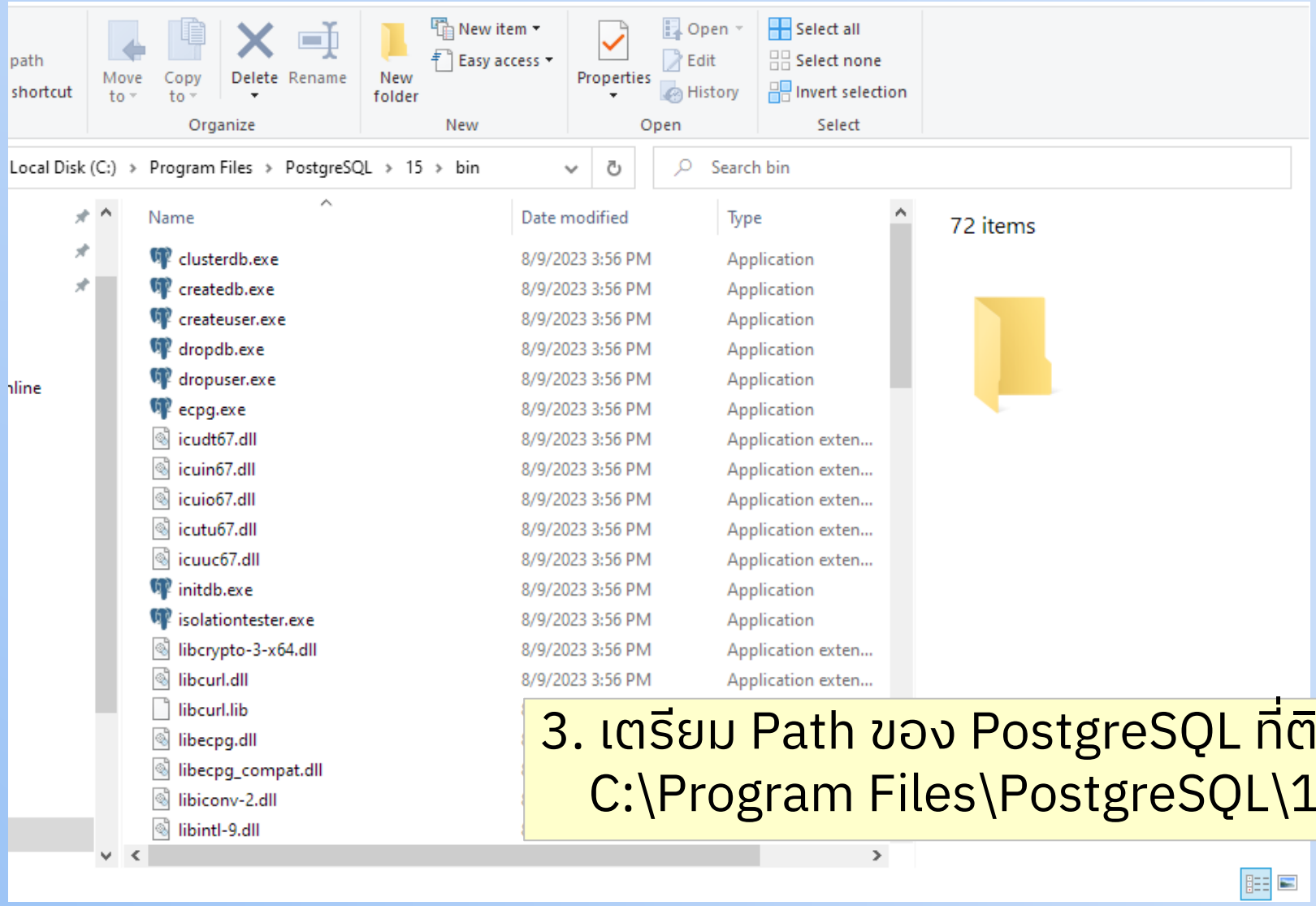
1. เข้าไปที่ System เลือก Advanced system setting ดังภาพ

การกำหนด Path ของ PostgreSQL ให้กับ Windows



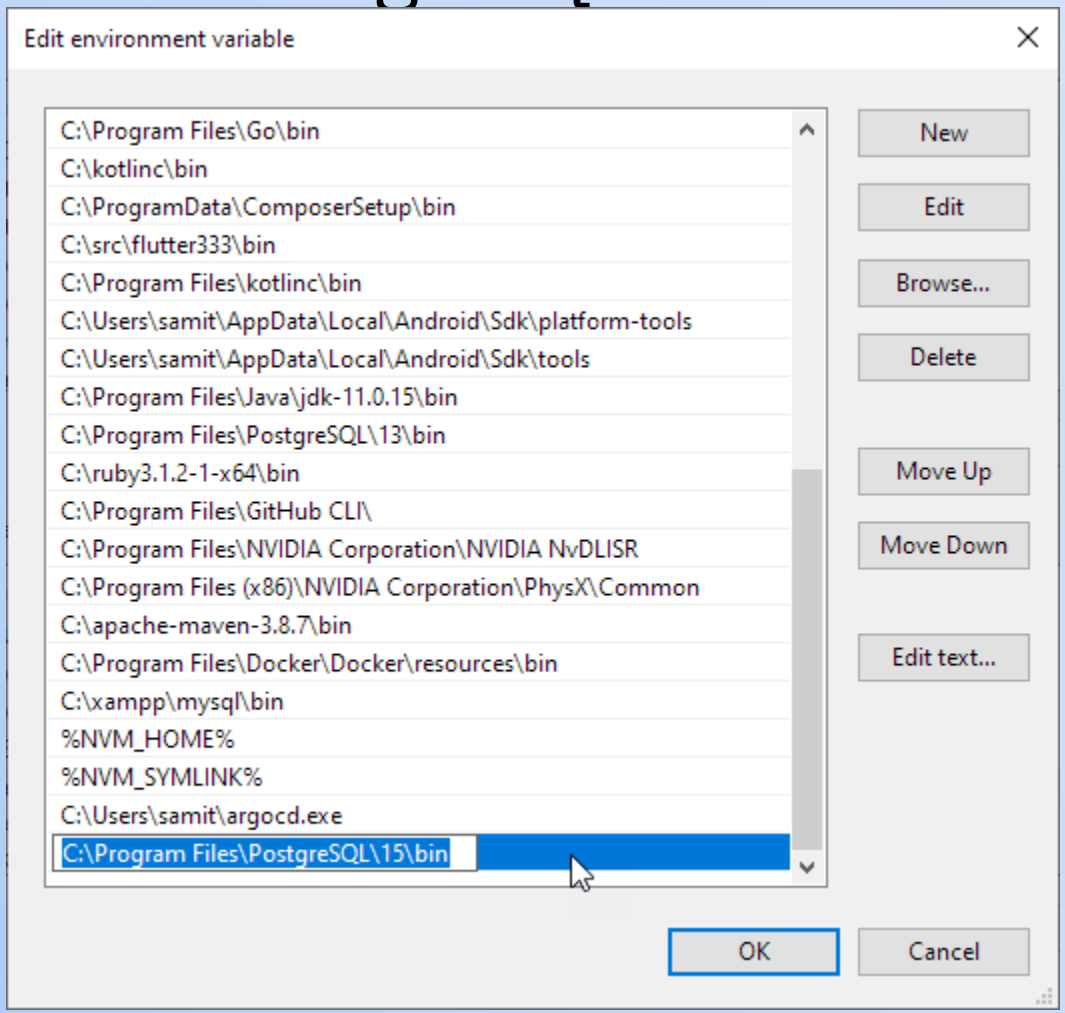
2. ที่ System variable ด้านล่างเลือก Path แล้วคลิ๊กปุ่ม Edit

การกำหนด Path ของ PostgreSQL ให้กับ Windows



3. เตรียม Path ของ PostgreSQL ที่ติดตั้งไว้ ในตัวอย่างคือ C:\Program Files\PostgreSQL\15\bin

การกำหนด Path ของ PostgreSQL ให้กับ Windows

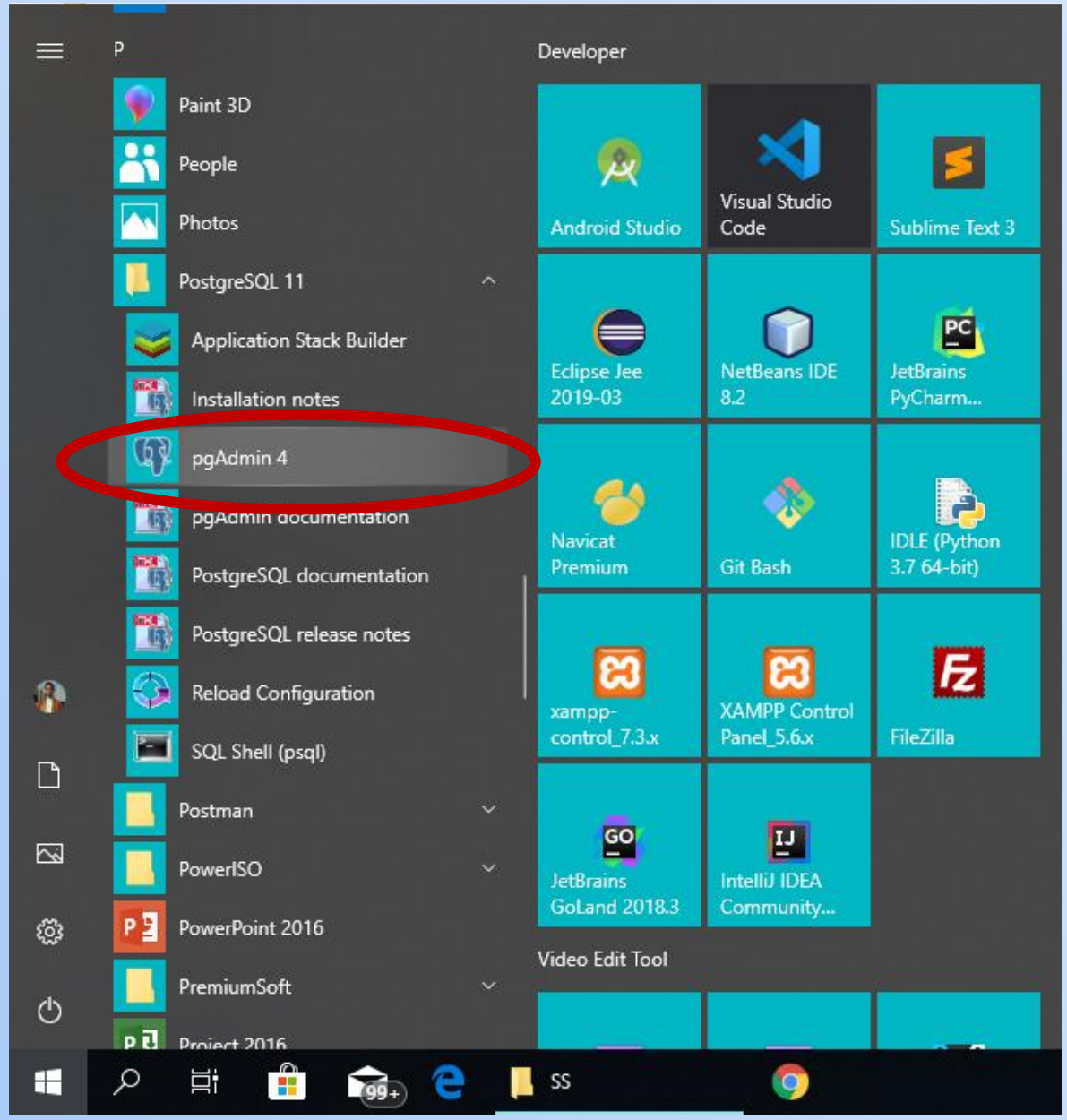


4. วาง Path ที่ได้ลงไป แล้วคลิ๊ก OK

Verify the Installation

Select Command Prompt

```
C:\Users\samit>psql --version
psql (PostgreSQL) 15.4

C:\Users\samit>
```


P
Paint 3D
People
Photos
PostgreSQL 11
Application Stack Builder
Installation notes
pgAdmin 4
pgAdmin documentation
PostgreSQL documentation
PostgreSQL release notes
Reload Configuration
SQL Shell (psql)
Postman
PowerISO
PowerPoint 2016
PremiumSoft
Project 2016
Developer
Android Studio
Visual Studio Code
Sublime Text 3
Eclipse Jee 2019-03
NetBeans IDE 8.2
JetBrains PyCharm...
Navicat Premium
Git Bash
IDLE (Python 3.7 64-bit)
xampp-control_7.3.x
XAMPP Control Panel_5.6.x
FileZilla
JetBrains GoLand 2018.3
IntelliJ IDEA Community...
Video Edit Tool
SS

pgAdmin 4

127.0.0.1:53306/browser/

pgAdmin 4 File Object Tools Help

Browser

Servers (1)

Dashboard Properties SQL Statistics Dependencies Dependents

Welcome

pgAdmin

Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links

Add New Server

Configure pgAdmin

Getting Started

PostgreSQL Documentation

pgAdmin Website

Planet PostgreSQL

Community Support

pgAdmin 4

127.0.0.1:53306/browser/

pgAdmin 4 File Object Tools Help

Browser | Dashboard | Properties | SQL | Statistics | Dependencies | Dependents

Servers (1)

PostgreSQL 11

Please connect to the selected server to view the dashboard.

Connect to Server

Please enter the password for the user 'postgres' to connect the server - "PostgreSQL 11"

Password ••••••

Save Password

Cancel OK

pgAdmin 4

File Object Tools Help

Object Explorer

- PostgreSQL 15
  - Databases (1)
    - postgres
      - Casts
      - Catalogs
      - Event Triggers
      - Extensions
      - Foreign Data Wrappers
      - Languages
      - Publications
      - Schemas
      - Subscriptions
  - Login/Group Roles (13)
    - pg_checkpoint
    - pg_database_owner
    - pg_execute_server_program
    - pg_monitor
    - pg_read_all_data
    - pg_read_all_settings
    - pg_read_all_stats
    - pg_read_server_files
    - pg_signal_backend
    - pg_stat_scan_tables
    - pg_write_all_data
    - pg_write_server_files
    - postgres
  - Tablespaces (2)
    - pg_default
    - pg_global

Dashboard | Properties | SQL | Statistics | Dependencies | Dependents

Server sessions — Total, Active, Idle

Transactions per second — Transactions, Commits, Rollbacks

You are currently running version 7.5 of pgAdmin 4, however the current version is 7.6.

Please click **here** for more information.

Tuples in — Inserts, Updates, Deletes

Tuples out — Fetched, Returned

Block I/O — Reads, Hits

Server activity

Sessions | Locks | Prepared Transactions | Configuration

Active sessions only

Search

			PID	Database	User	Application	Client	Backend start	Transaction start	State	Wait event
			1756		postgr...			2023-10-25 01:01:02...			Activity: Logi
			4400	postgres	postgr...	pgAdmin 4 - DB:post...	::1	2023-10-25 01:08:50...	2023-10-25 01:09:17...	act...	
			5056					2023-10-25 01:01:02...			Activity: Walv
			12752					2023-10-25 01:01:02...			Activity: BgW
			15428					2023-10-25 01:01:02...			Activity: Chec
			16532					2023-10-25 01:01:02...			Activity: Auto

PostgreSQL

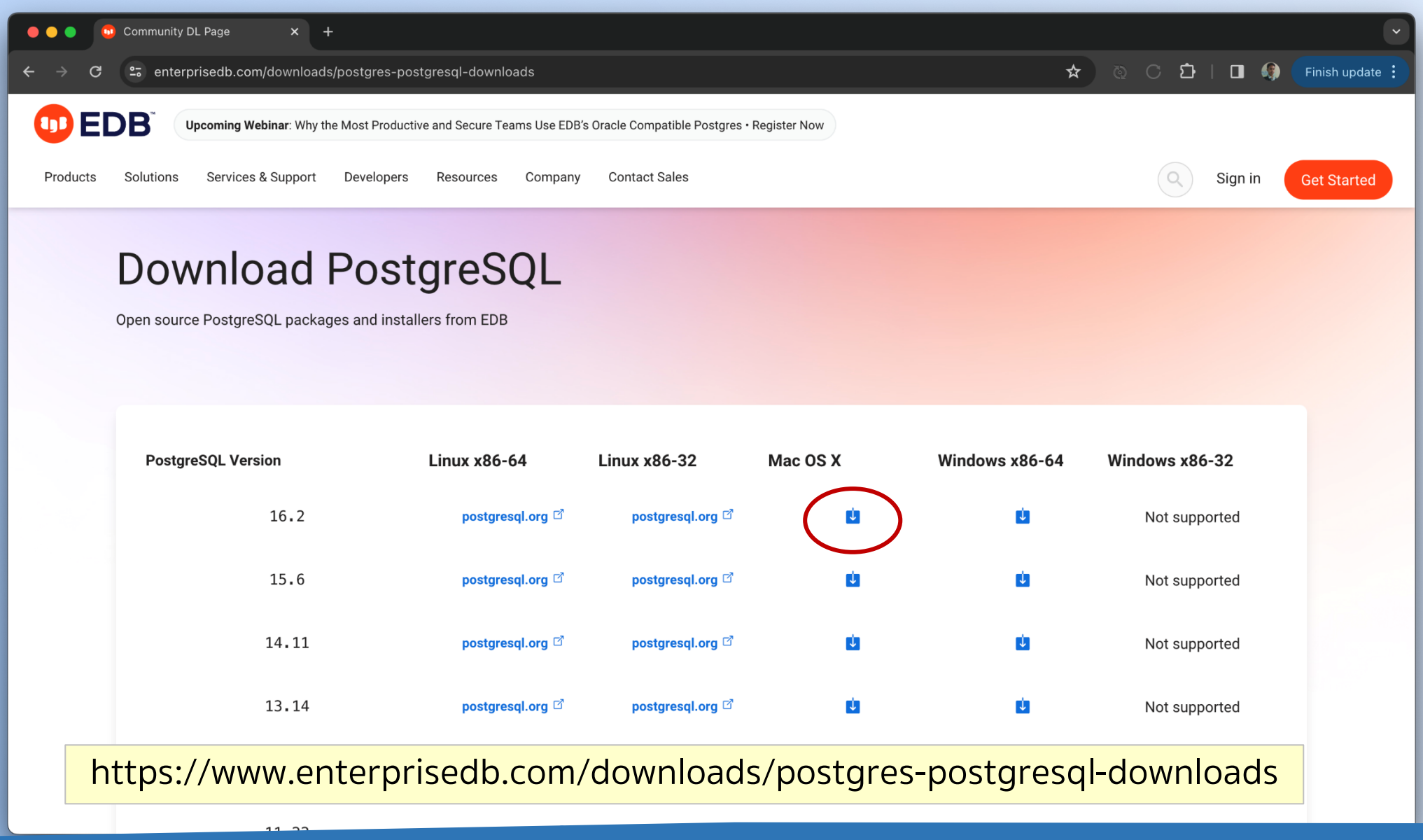
ติดตั้ง PostgreSQL 16.x

ติดตั้ง PostgreSQL 16.x บน MacOS

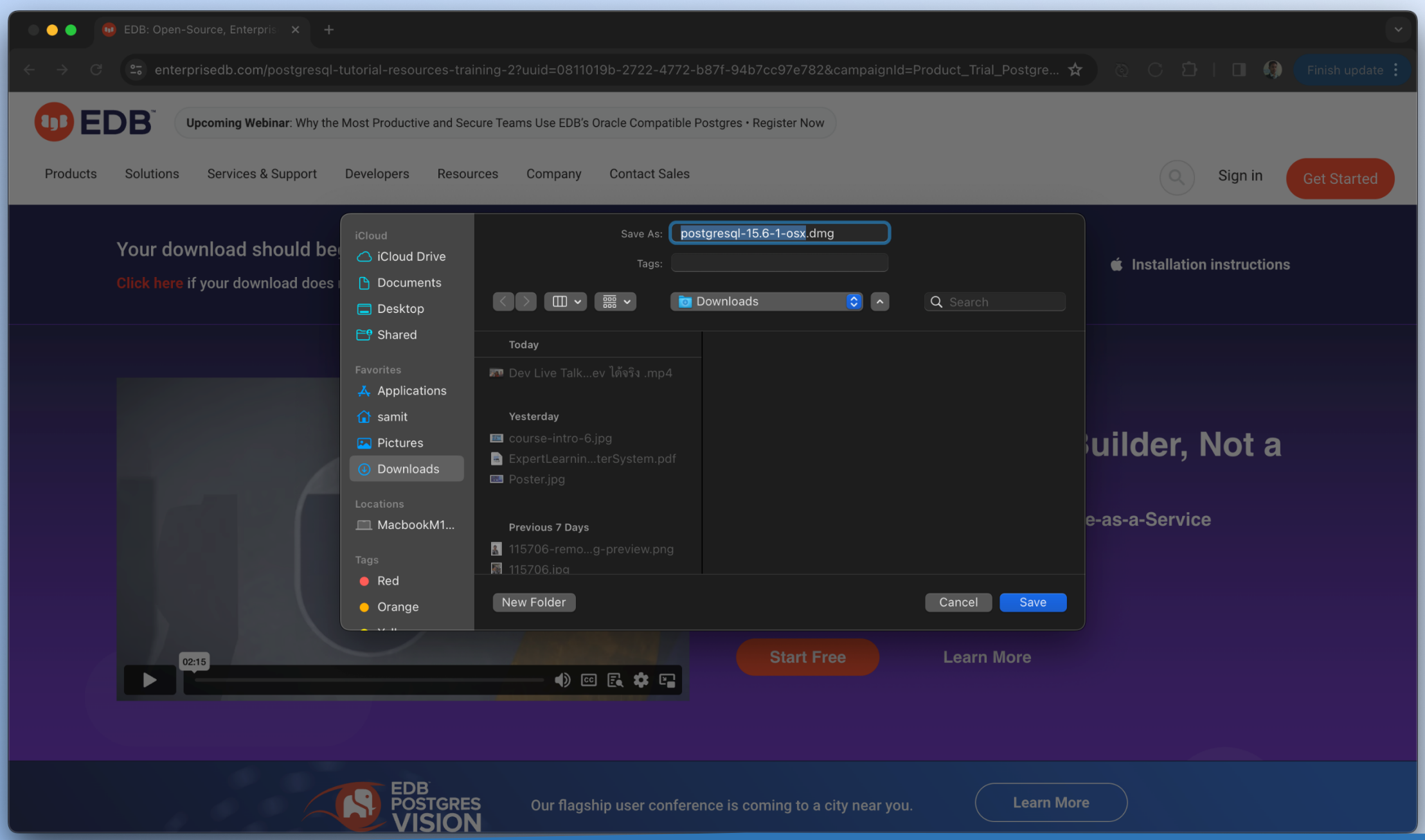
- Download PostgreSQL 16.x installer for MacOS
- Install PostgreSQL
- Verify the installation

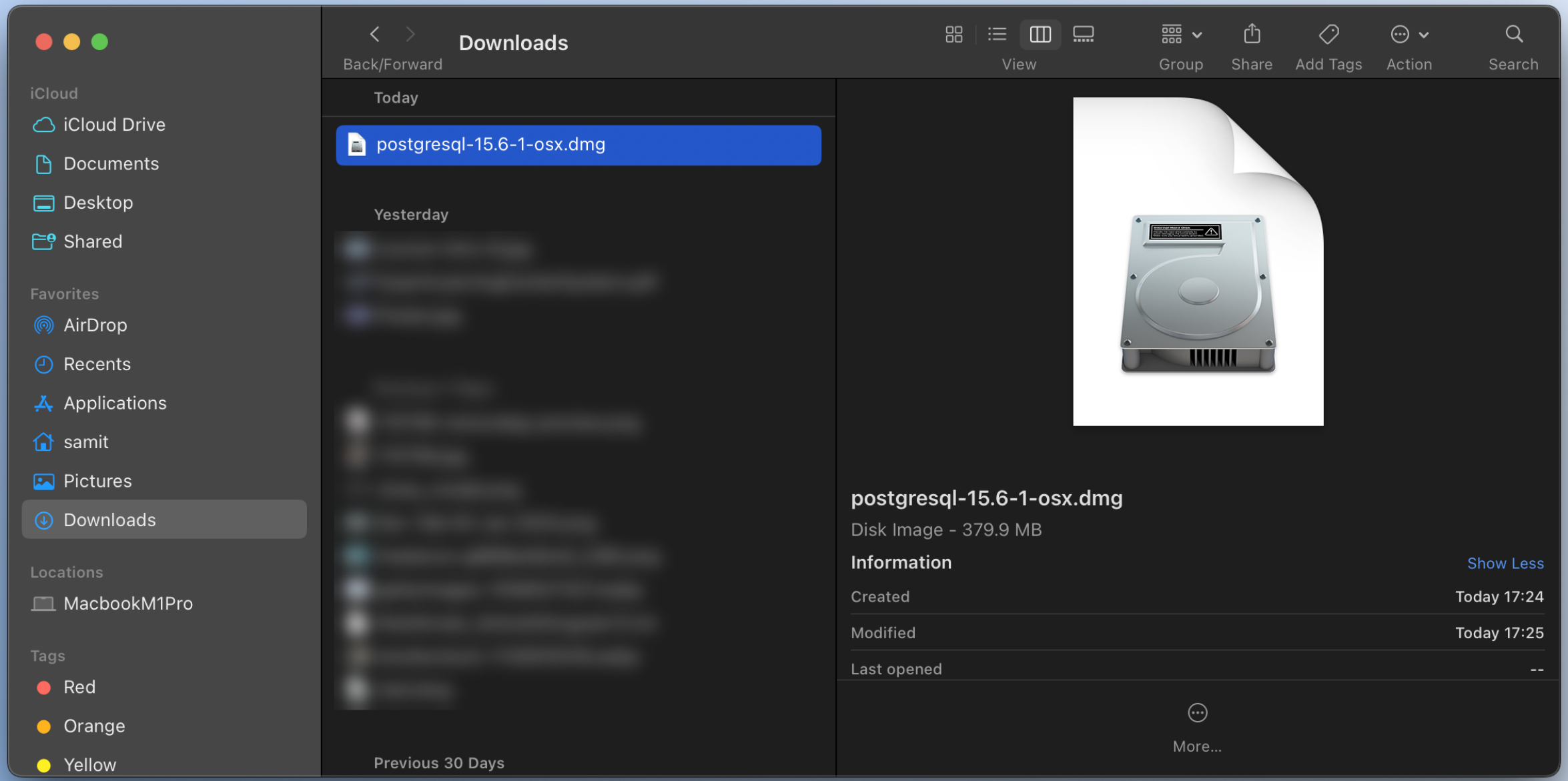
Download PostgreSQL for MacOS

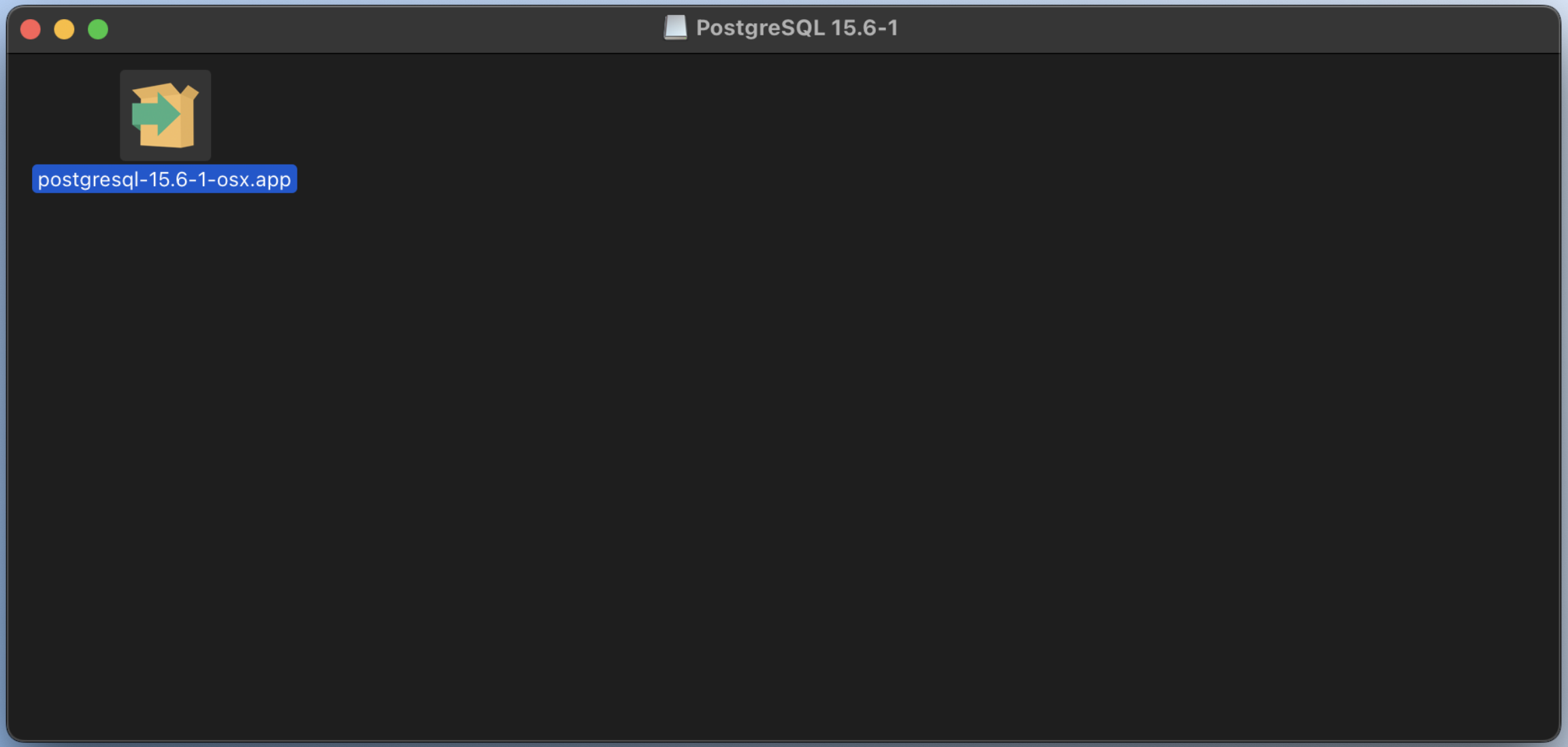
ดาวน์โหลดชุดติดตั้ง PostgreSQL 16.x สำหรับ MacOS

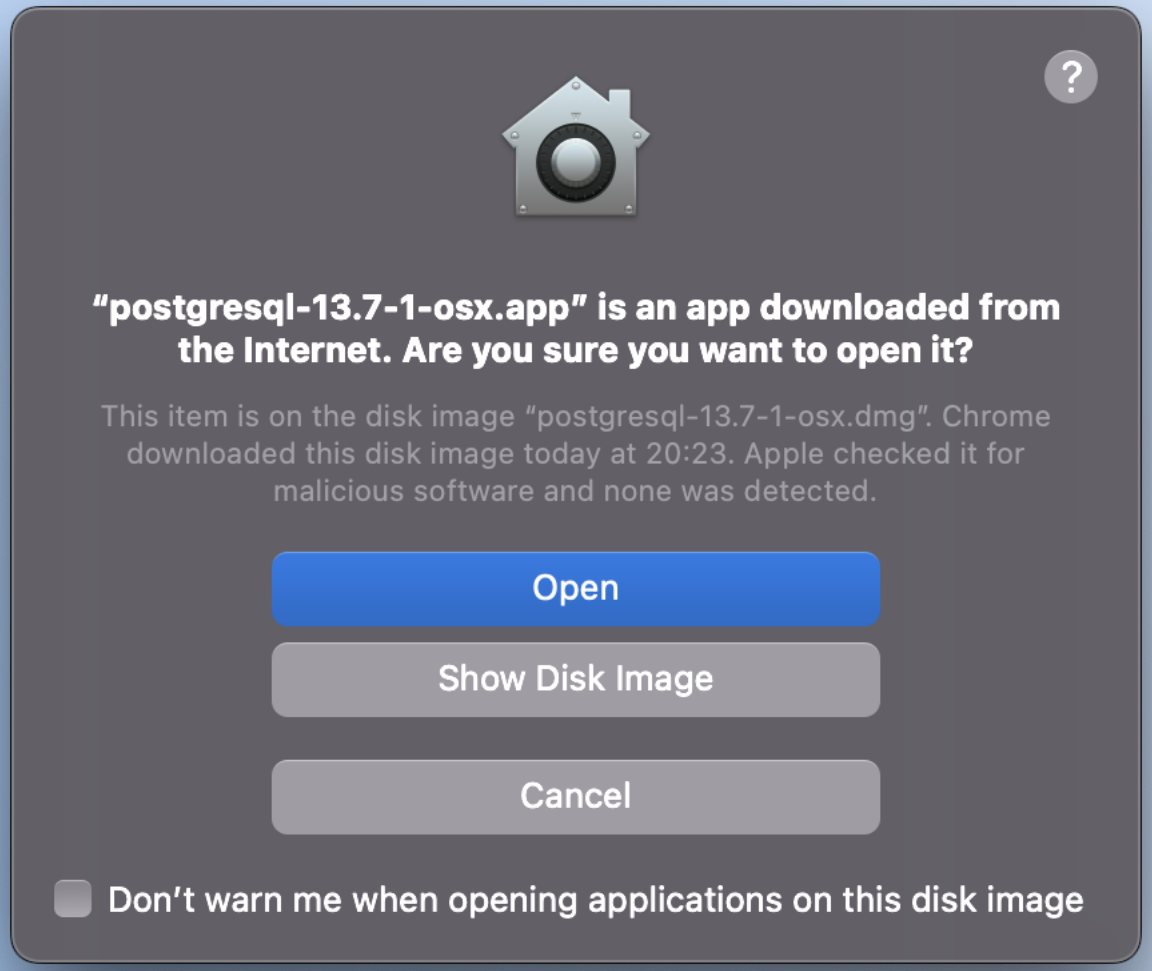


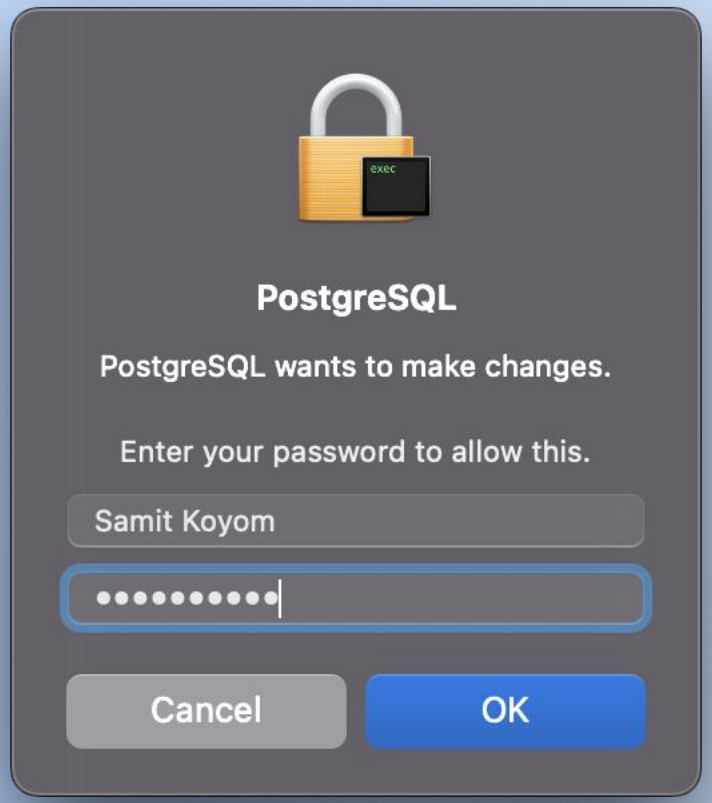
https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

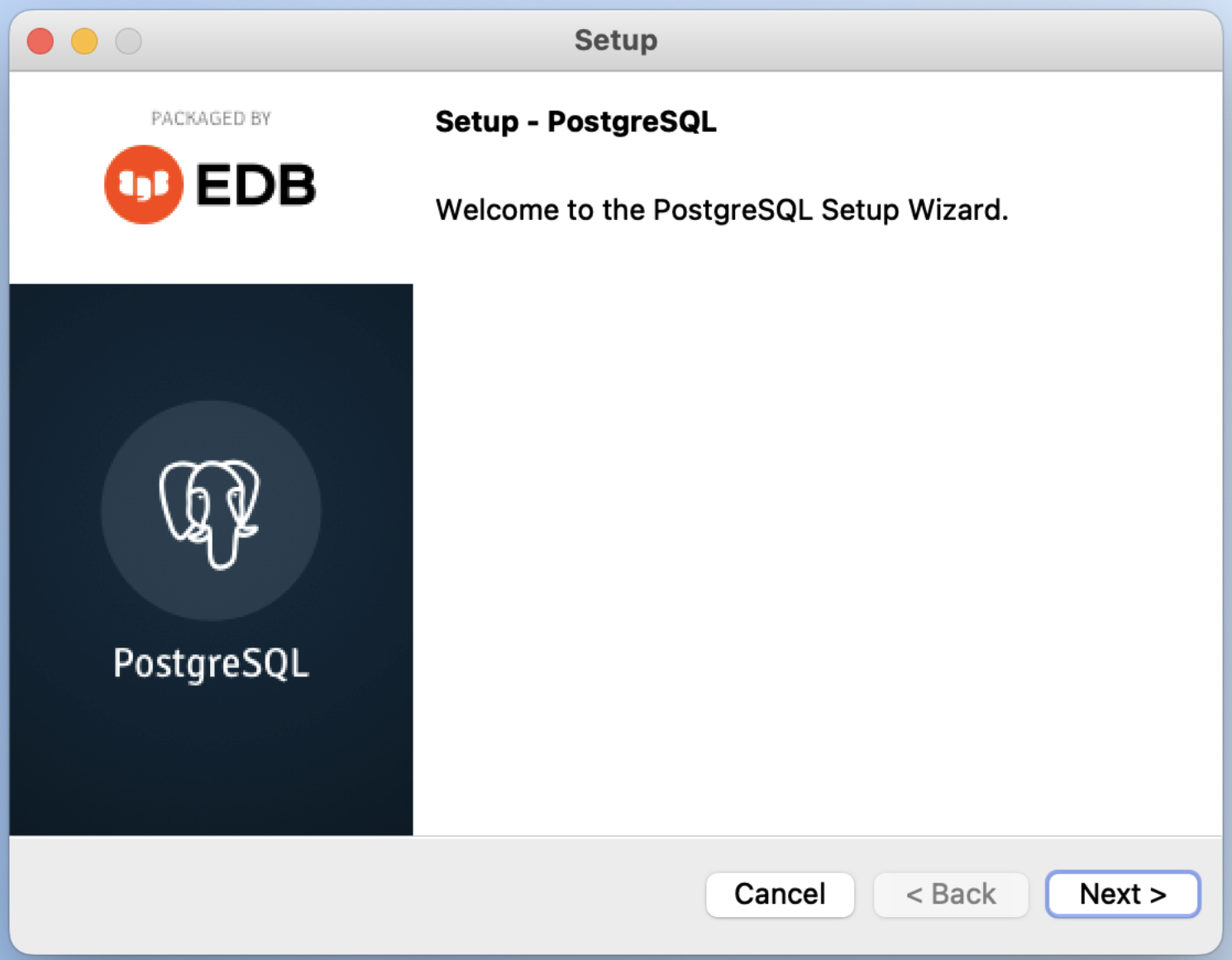

EDB: Open-Source, Enterpris
enterprisedb.com/postgresql-tutorial-resources-training-2?uuid=0811019b-2722-4772-b87f-94b7cc97e782&campaignId=Product_Trial_Postgre...
Finish update
EDB
Upcoming Webinar: Why the Most Productive and Secure Teams Use EDB's Oracle Compatible Postgres • Register Now
Products
Solutions
Services & Support
Developers
Resources
Company
Contact Sales
Sign in
Get Started
Your download should be
Click here if your download does
Installation instructions
Save As: postgresql-15.6-1-osx.dmg
Tags:
Downloads
Search
iCloud
iCloud Drive
Documents
Desktop
Shared
Favorites
Applications
samit
Pictures
Downloads
Locations
MacbookM1...
Tags
Red
Orange
Today
Dev Live Talk...ev ได้จริง .mp4
Yesterday
course-intro-6.jpg
ExpertLearnin...terSystem.pdf
Poster.jpg
Previous 7 Days
115706-remo...g-preview.png
115706.jpg
New Folder
Cancel
Save
uilder, Not a
e-as-a-Service
02:15
Start Free
Learn More
EDB POSTGRES VISION
Our flagship user conference is coming to a city near you.
Learn More

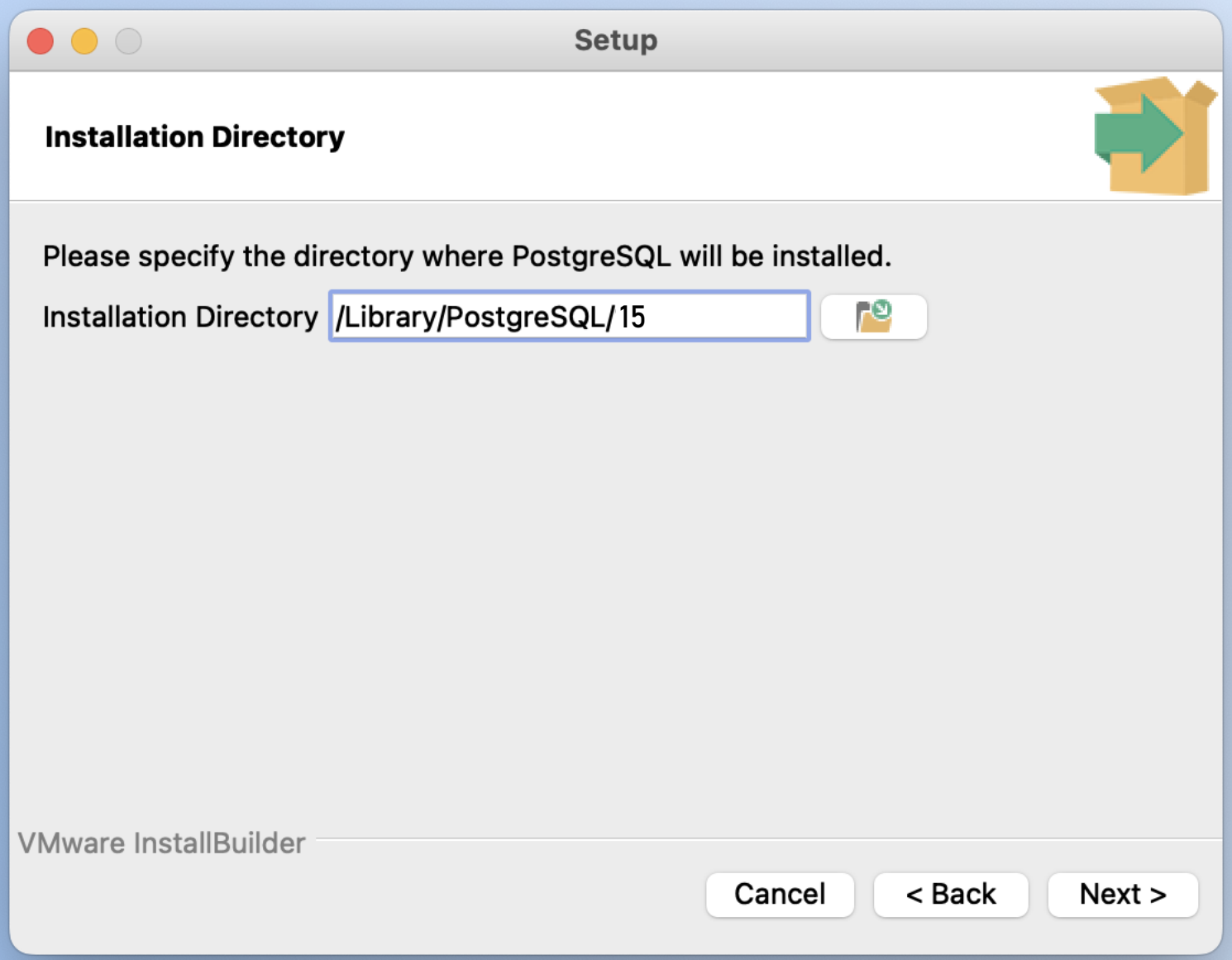

Downloads
Back/Forward
View
Group
Share
Add Tags
Action
Search
iCloud
iCloud Drive
Documents
Desktop
Shared
Favorites
AirDrop
Recents
Applications
samit
Pictures
Downloads
Locations
MacbookM1Pro
Tags
Red
Orange
Yellow
Today
postgresql-15.6-1-osx.dmg
Yesterday
Previous 30 Days
postgresql-15.6-1-osx.dmg
Disk Image - 379.9 MB
Information
Show Less
Created
Today 17:24
Modified
Today 17:25
Last opened
--
More...

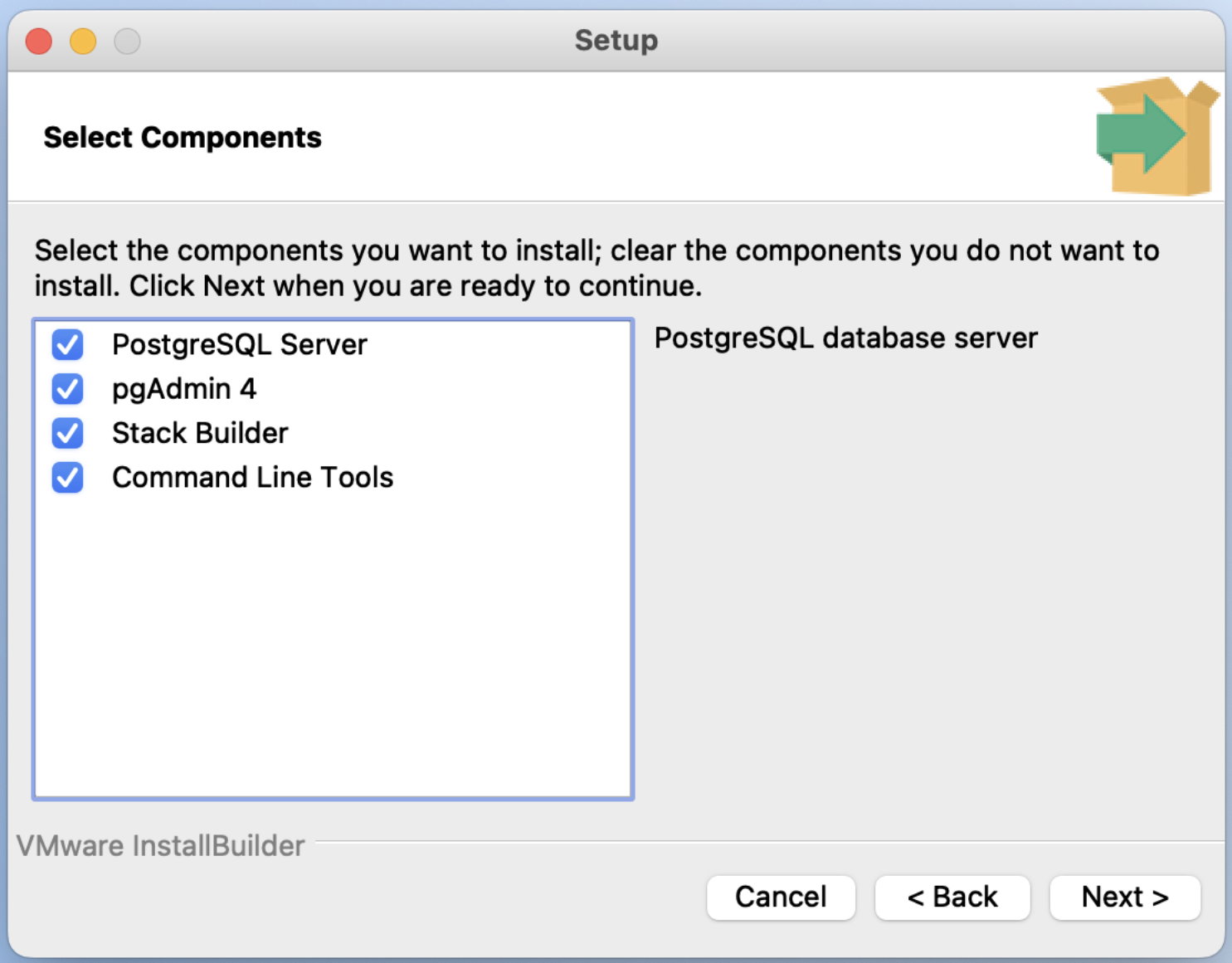

PostgreSQL 15.6-1
postgresql-15.6-1-osx.app

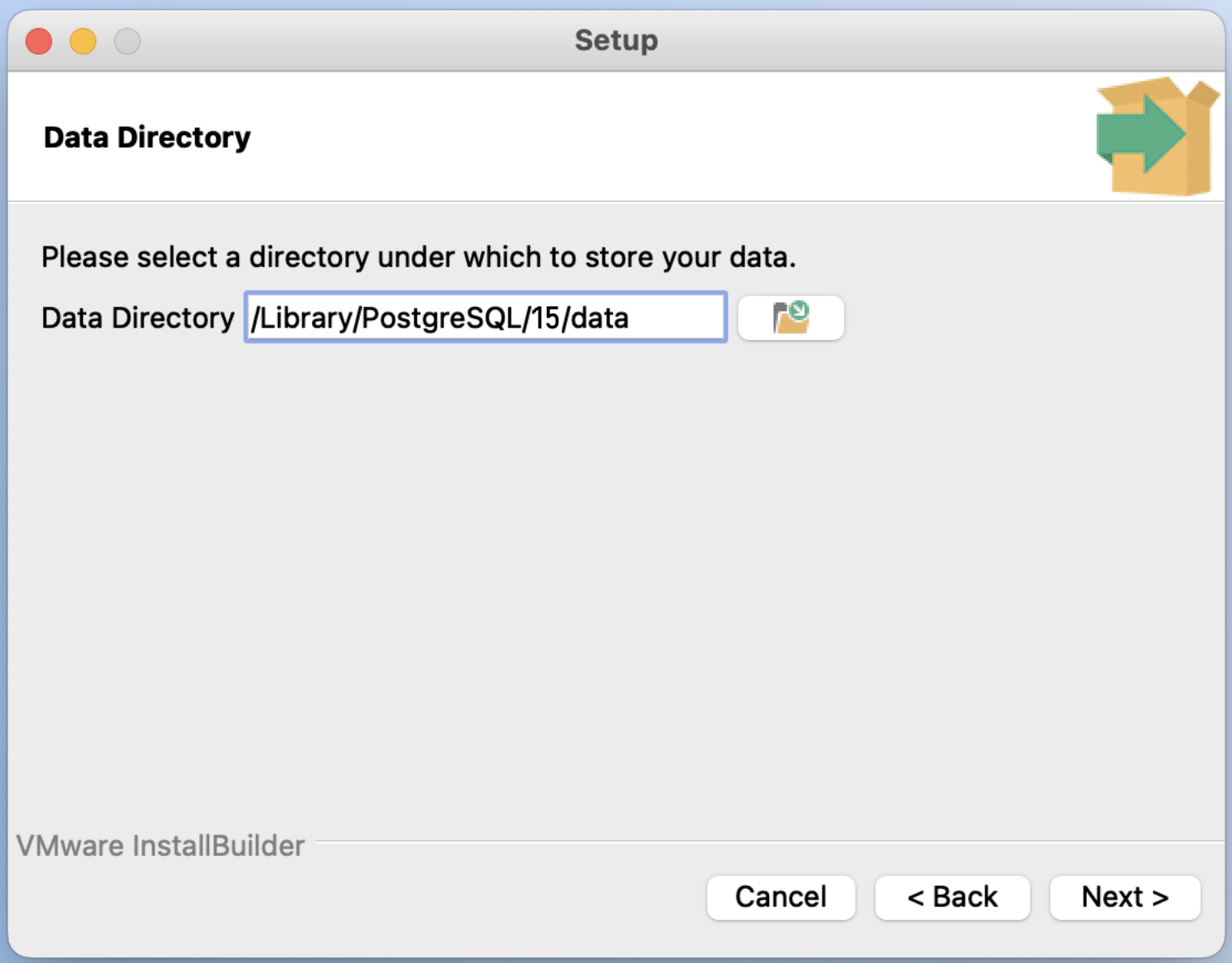

"postgresql-13.7-1-osx.app" is an app downloaded from the Internet. Are you sure you want to open it?
This item is on the disk image "postgresql-13.7-1-osx.dmg". Chrome downloaded this disk image today at 20:23. Apple checked it for malicious software and none was detected.
Open
Show Disk Image
Cancel
Don't warn me when opening applications on this disk image

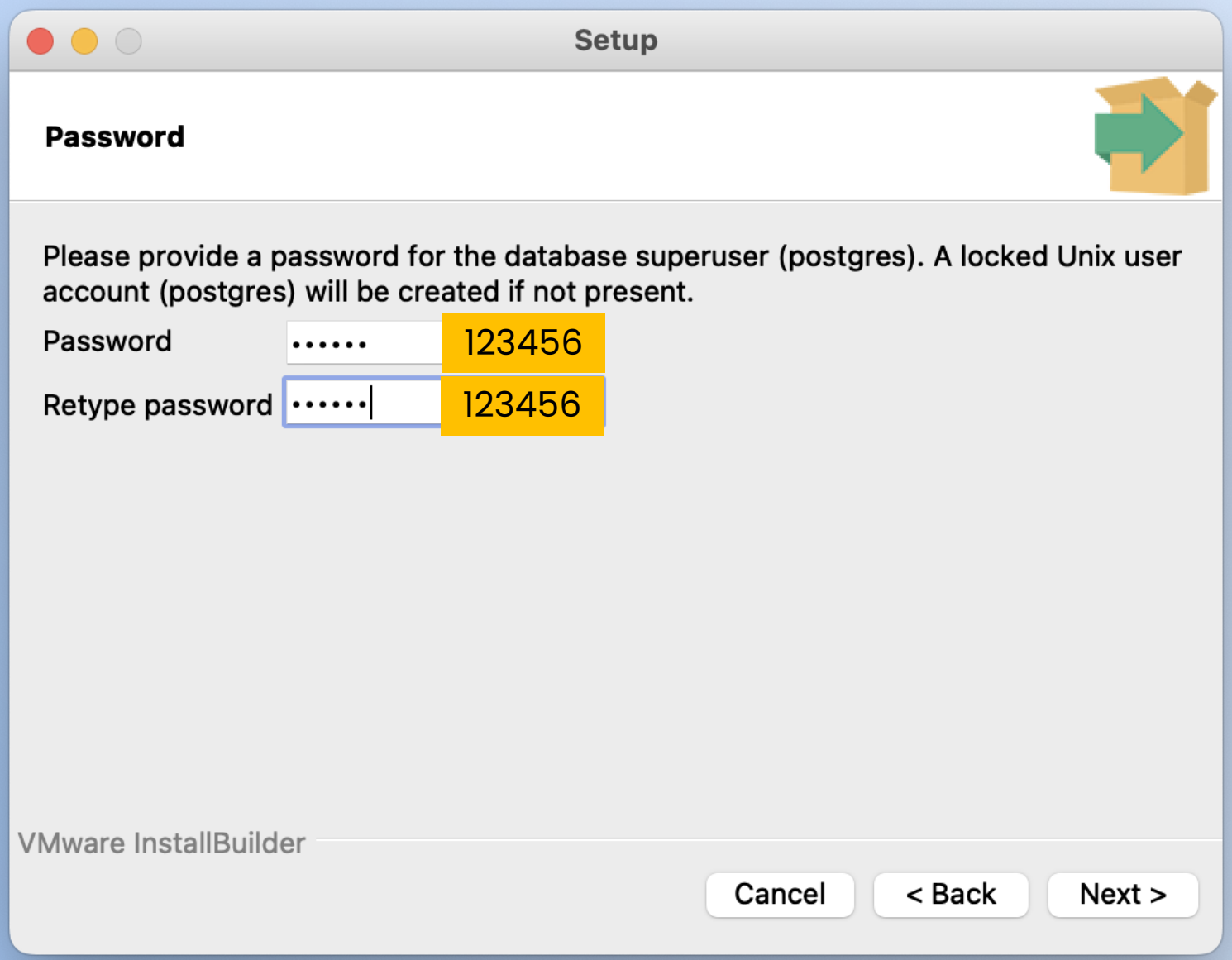

exec
PostgreSQL
PostgreSQL wants to make changes.
Enter your password to allow this.
Samit Koyom
Cancel
OK

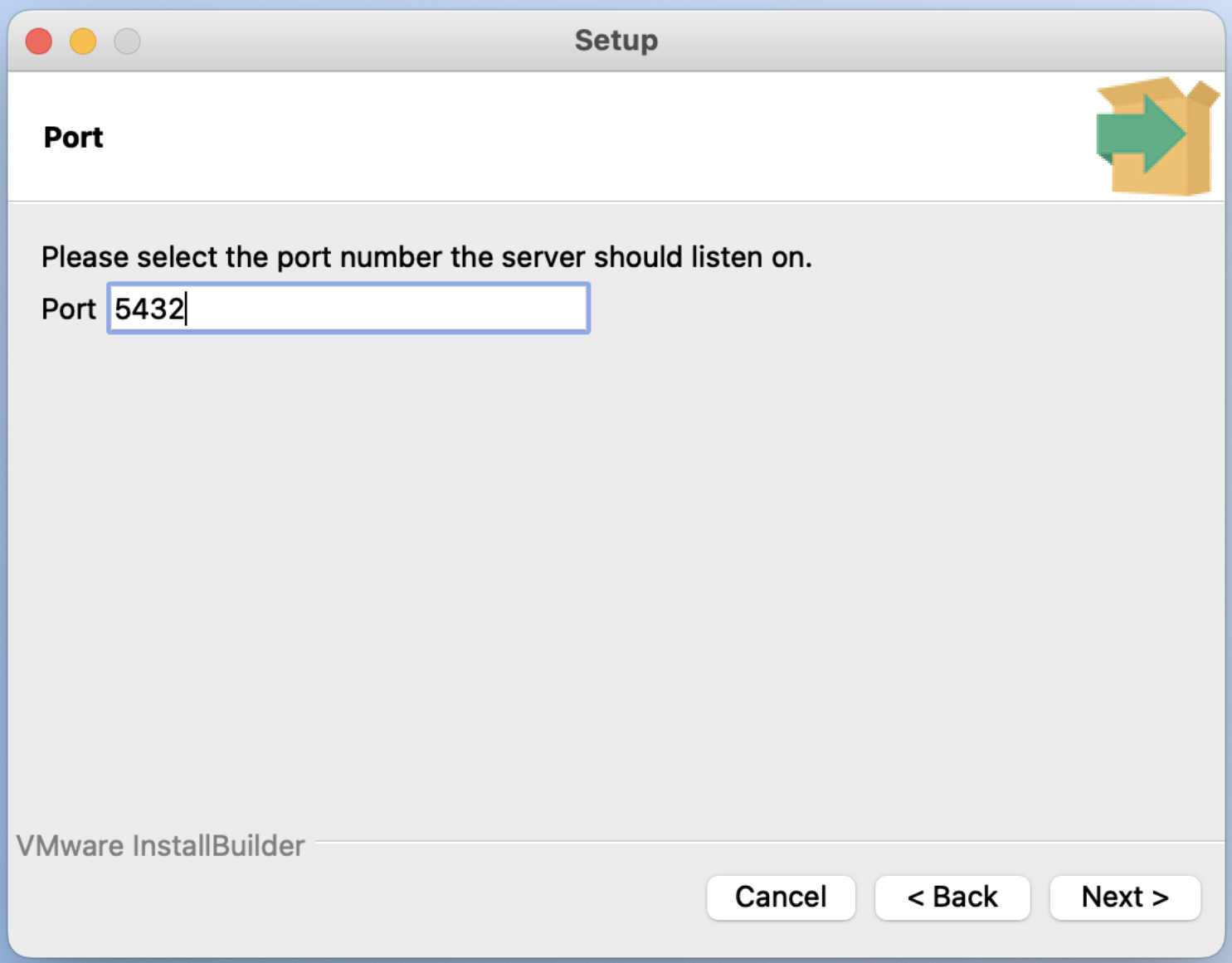

Setup
PACKAGED BY
EDB
Setup - PostgreSQL
Welcome to the PostgreSQL Setup Wizard.
PostgreSQL
Cancel
< Back
Next >

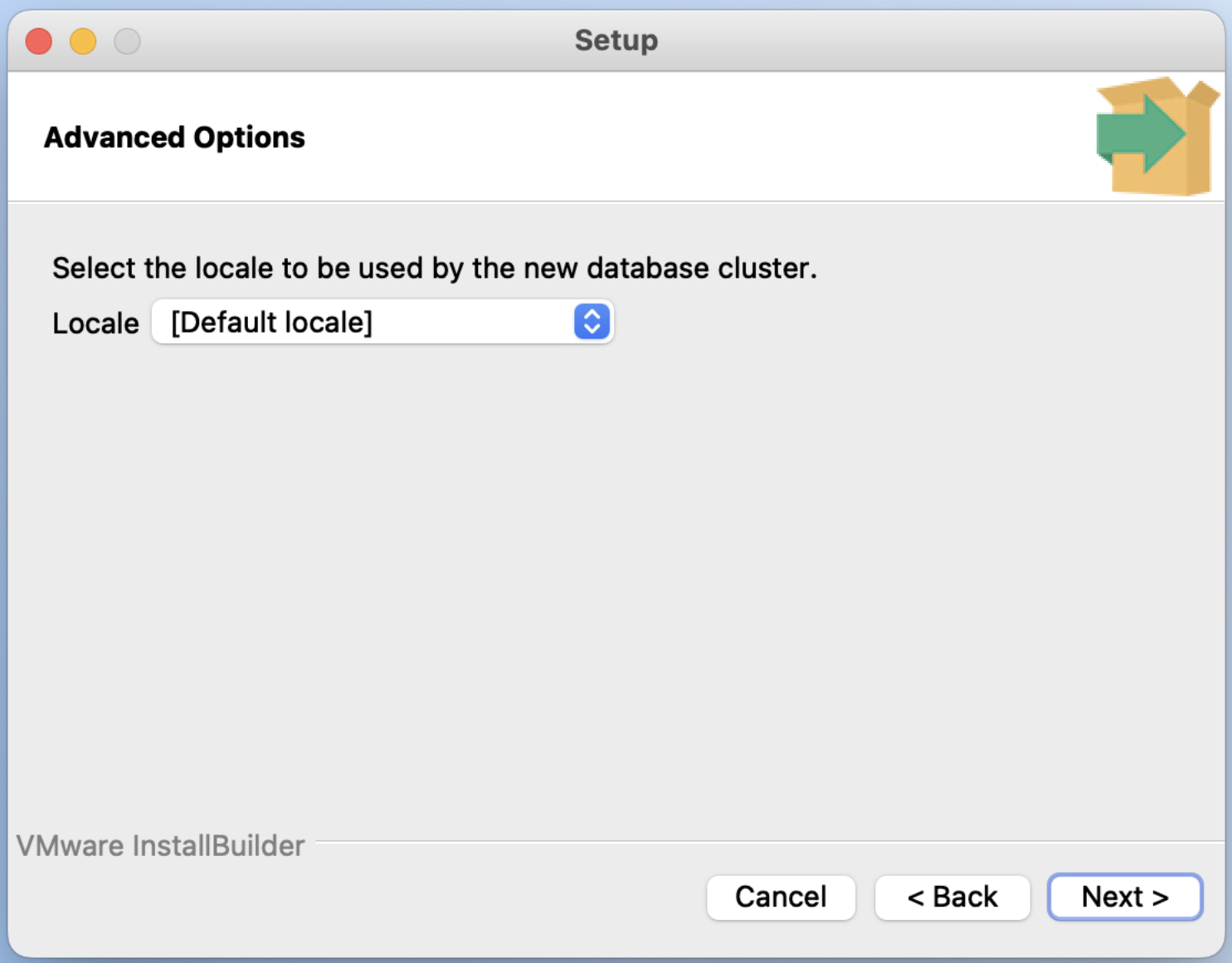

Setup
Installation Directory
Please specify the directory where PostgreSQL will be installed.
Installation Directory
/Library/PostgreSQL/15
VMware InstallBuilder
Cancel
< Back
Next >

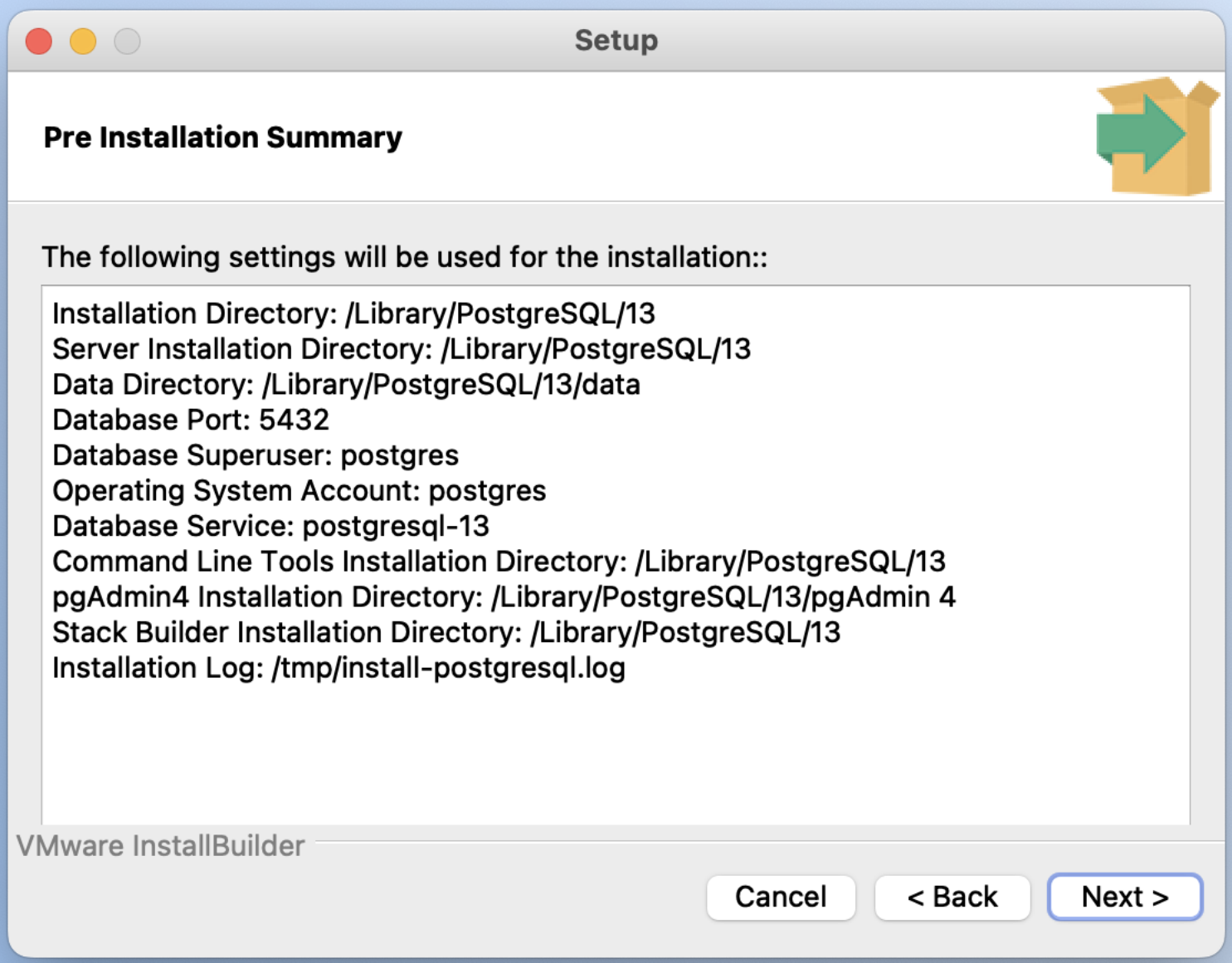

Setup
Select Components
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
PostgreSQL Server
pgAdmin 4
Stack Builder
Command Line Tools
PostgreSQL database server
VMware InstallBuilder
Cancel
< Back
Next >

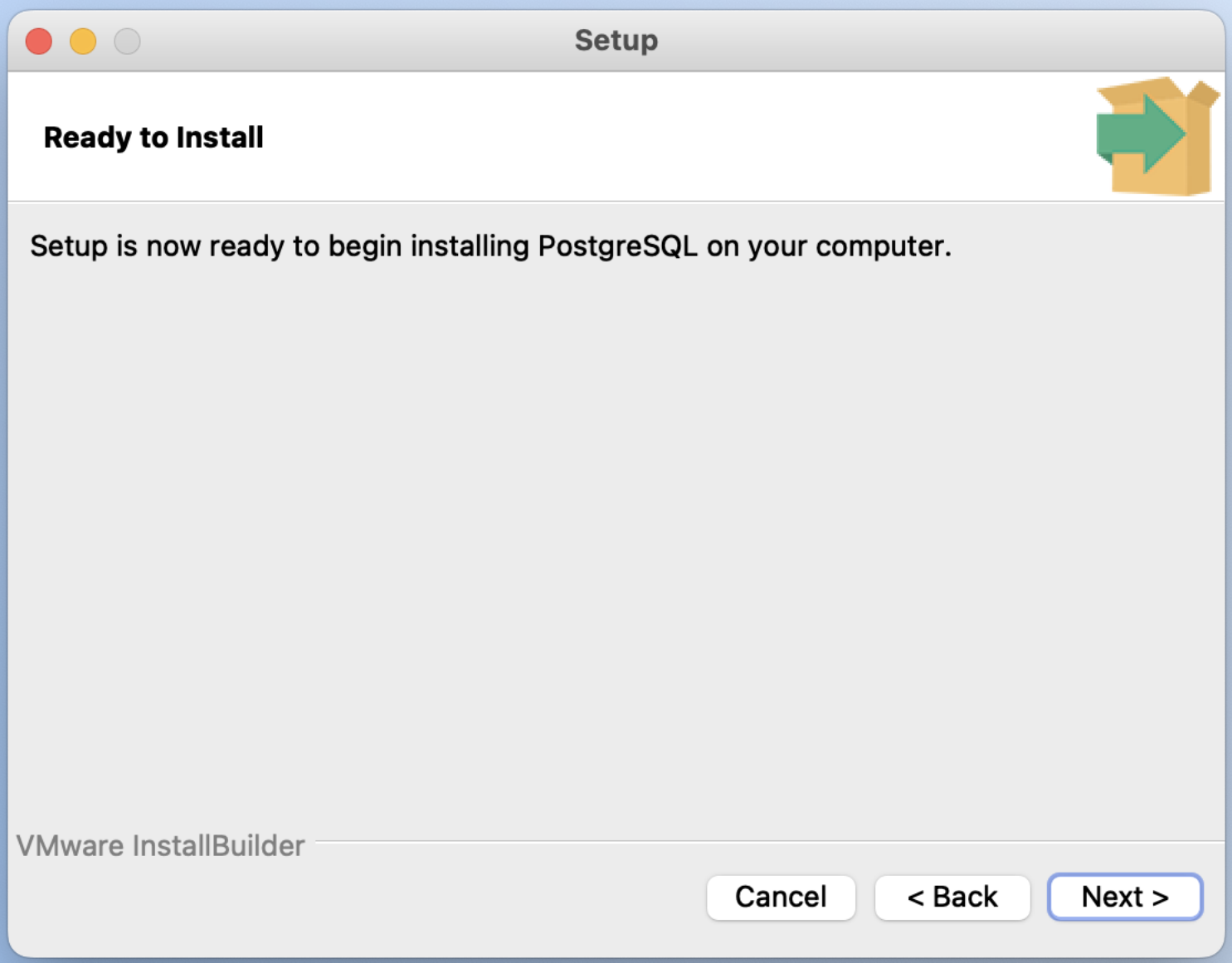

Setup
Data Directory
Please select a directory under which to store your data.
Data Directory
/Library/PostgreSQL/15/data
VMware InstallBuilder
Cancel
< Back
Next >

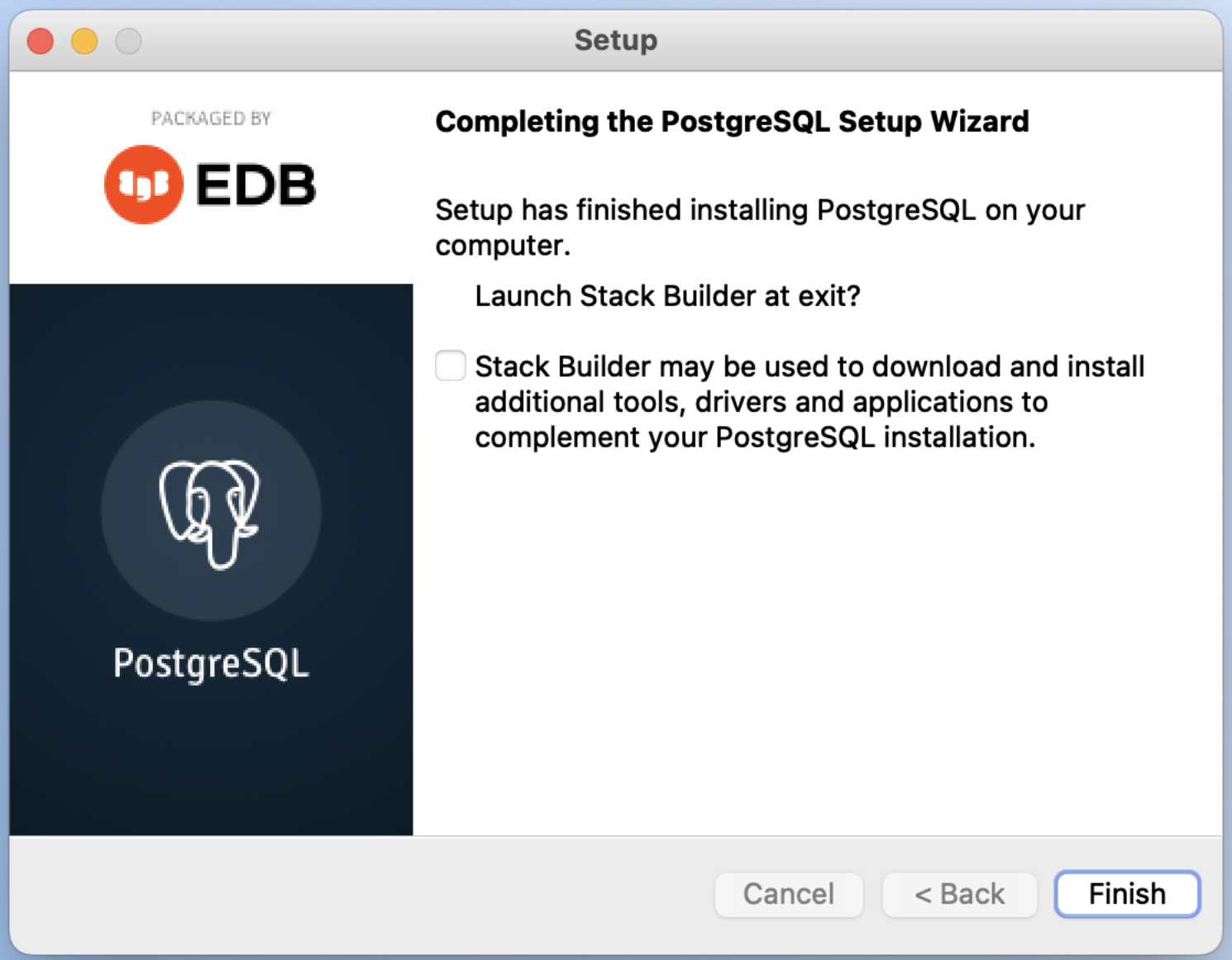

Setup
Password
Please provide a password for the database superuser (postgres). A locked Unix user account (postgres) will be created if not present.
Password
123456
Retype password
123456
VMware InstallBuilder
Cancel
< Back
Next >

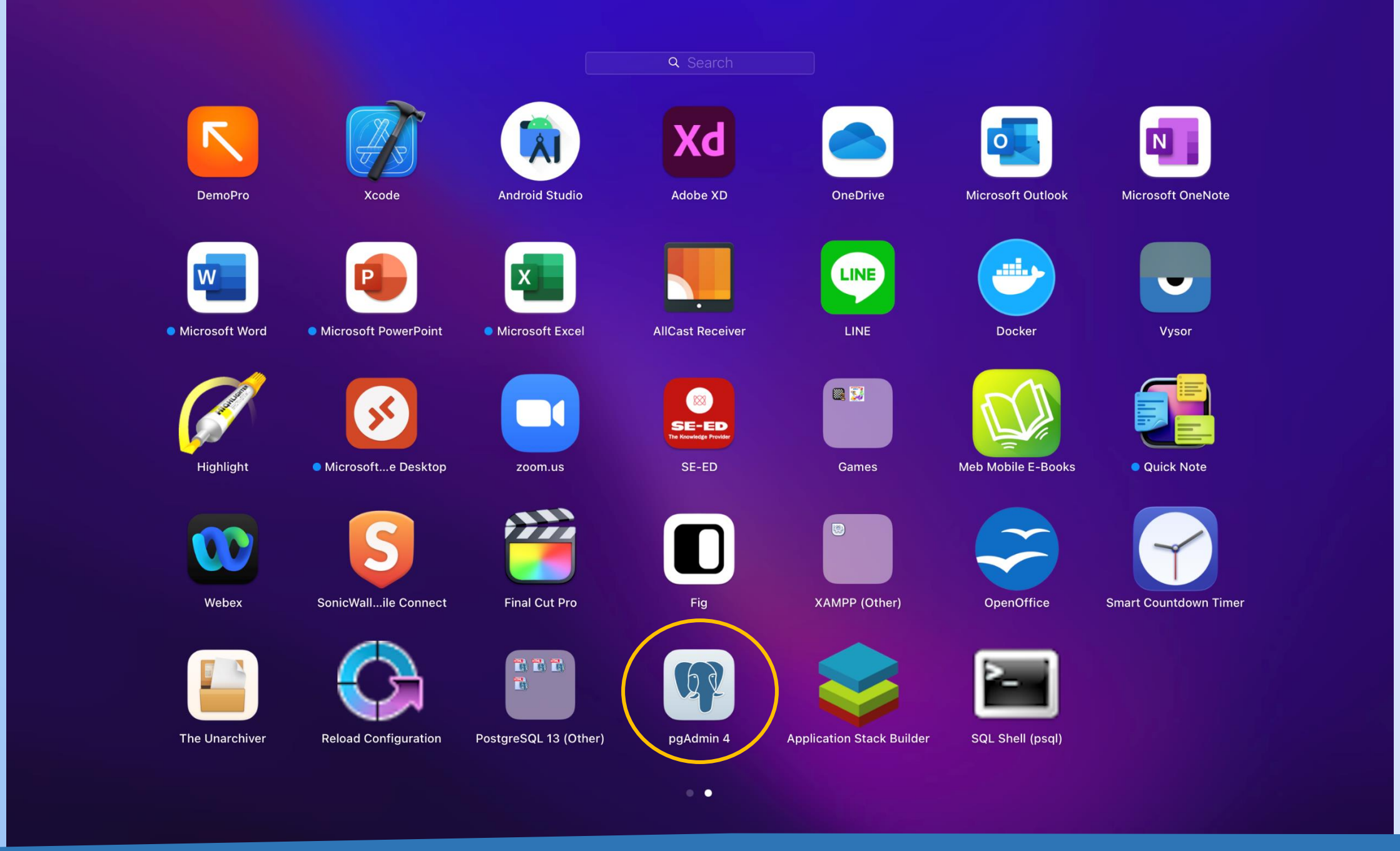

Setup
Port
Please select the port number the server should listen on.
Port 5432
VMware InstallBuilder
Cancel
< Back
Next >

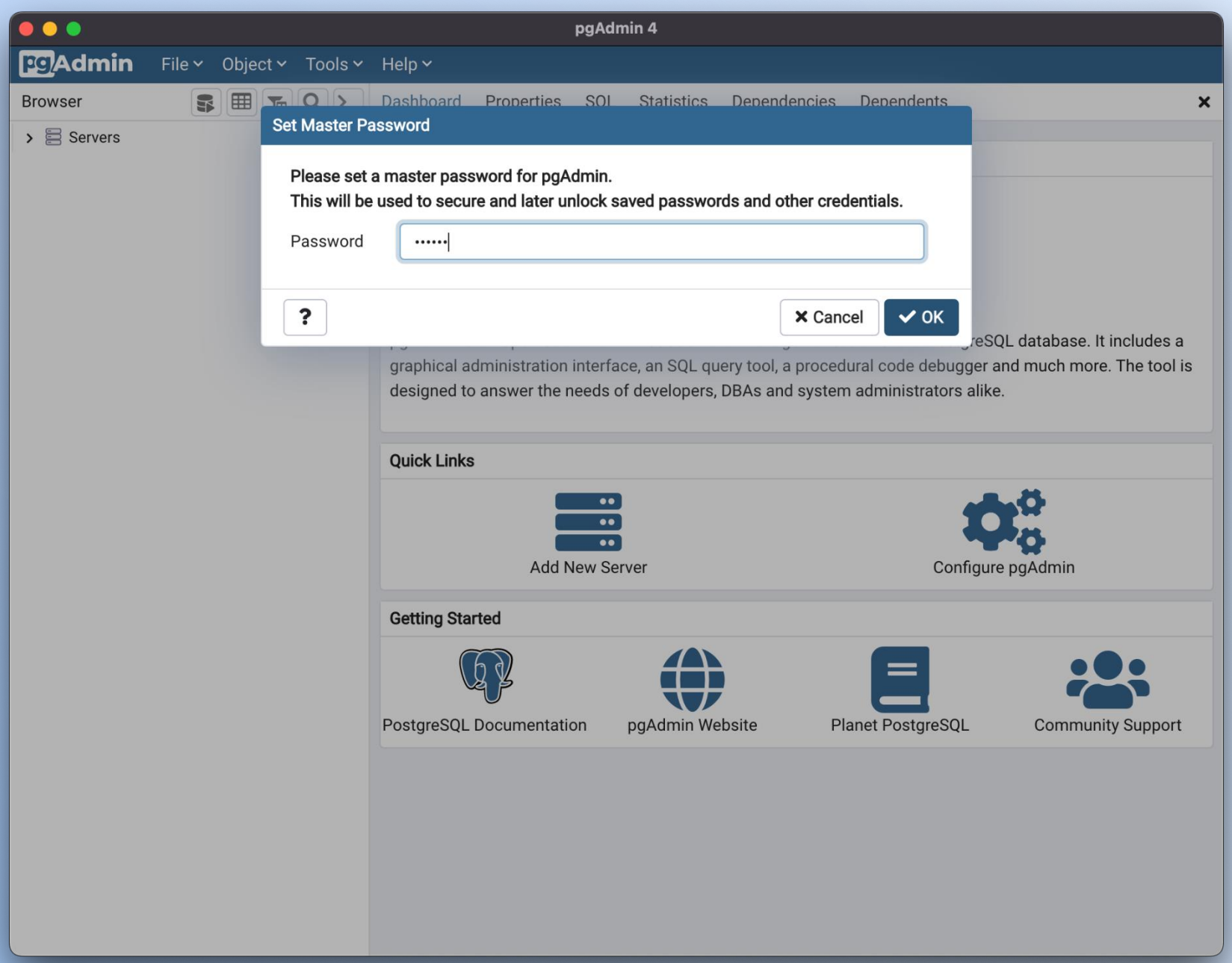

Setup
Advanced Options
Select the locale to be used by the new database cluster.
Locale
[Default locale]
VMware InstallBuilder
Cancel
< Back
Next >


Setup
Pre Installation Summary
The following settings will be used for the installation::
Installation Directory: /Library/PostgreSQL/13
Server Installation Directory: /Library/PostgreSQL/13
Data Directory: /Library/PostgreSQL/13/data
Database Port: 5432
Database Superuser: postgres
Operating System Account: postgres
Database Service: postgresql-13
Command Line Tools Installation Directory: /Library/PostgreSQL/13
pgAdmin4 Installation Directory: /Library/PostgreSQL/13/pgAdmin 4
Stack Builder Installation Directory: /Library/PostgreSQL/13
Installation Log: /tmp/install-postgresql.log
VMware InstallBuilder
Cancel
< Back
Next >


Setup
Ready to Install
Setup is now ready to begin installing PostgreSQL on your computer.
VMware InstallBuilder
Cancel
< Back
Next >


Setup
PACKAGED BY
EDB
PostgreSQL
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.
Cancel
< Back
Finish


Search
DemoPro
Xcode
Android Studio
Adobe XD
OneDrive
Microsoft Outlook
Microsoft OneNote
Microsoft Word
Microsoft PowerPoint
Microsoft Excel
AllCast Receiver
LINE
Docker
Vysor
Highlight
Microsoft...e Desktop
zoom.us
SE-ED
Games
Meb Mobile E-Books
Quick Note
Webex
SonicWall...ile Connect
Final Cut Pro
Fig
XAMPP (Other)
OpenOffice
Smart Countdown Timer
The Unarchiver
Reload Configuration
PostgreSQL 13 (Other)
pgAdmin 4
Application Stack Builder
SQL Shell (psql)


pgAdmin 4
pgAdmin
File
Object
Tools
Help
Browser
Servers
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Set Master Password
Please set a master password for pgAdmin.
This will be used to secure and later unlock saved passwords and other credentials.
Password
Cancel
OK
graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.
Quick Links
Add New Server
Configure pgAdmin
Getting Started
PostgreSQL Documentation
pgAdmin Website
Planet PostgreSQL
Community Support

pgAdmin 4

File Object Tools Help

Browser

Dashboard Properties SQL Statistics Dependencies Dependents

- Servers (1)
  - PostgreSQL 13
    - Databases
    - Login/Group Roles
    - Tablespaces

Server sessions — Total, Active, Idle

Transactions per second — Transactions, Commits, Rollbacks

Tuples in

ock I/O — Reads, Hits

Server ac

Sessions

Search

About pgAdmin 4

Version	6.8
Application Mode	Desktop
Current User	pgadmin4@pgadmin.org
NW.js Version	0.55.0
Browser	Chromium 92.0.4515.107
Operating System	macOS-12.4-x86_64-i386-64bit
pgAdmin Database File	/Users/samit/.pgadmin/pgadmin4.db
Log File	/Users/samit/.pgadmin/pgadmin4.log

Server Configuration

Copy

```
ALLOW_SAVE_PASSWORD = True
ALLOW_SAVE_TUNNEL_PASSWORD = False
APP_COPYRIGHT = "Copyright (C) 2013 - 2022, The pgAdmin Development Team"
APP_ICON = "pg-icon"
APP_NAME = "pgAdmin 4"
APP_RELEASE = 6
APP_REVISION = 8
```

OK

									State	Wait event	Blocking PIDs
										Activity: Checkpointe...	
93131						2022-07-15 21:27:47 ...				Activity: BgWriterMain	
93132						2022-07-15 21:27:47 ...				Activity: WalWriterM...	
93133						2022-07-15 21:27:47 ...				Activity: AutoVacuu...	
93135		postgres				2022-07-15 21:27:47 ...				Activity: LogicalLaun...	
93488	postgres	postgres	pgAdmin 4 - DB:post...	::1		2022-07-15 21:29:12 ...	2022-07-15 21:29:12 ...	active			

pgAdmin 4

pgAdmin File Object Tools Help

Browser

- Servers (1)
  - PostgreSQL 13
    - Databases (1)
      - postgres
        - Casts
        - Catalogs
        - Event Triggers
        - Extensions
        - Foreign Data Wrappers
        - Languages
        - Publications
        - Schemas (1)
        - Subscriptions
    - Login/Group Roles (9)
      - pg_execute_server_program
      - pg_monitor
      - pg_read_all_settings
      - pg_read_all_stats
      - pg_read_server_files
      - pg_signal_backend
      - pg_stat_scan_tables
      - pg_write_server_files
      - postgres
    - Tablespaces (2)
      - pg_default
      - pg_global

Dashboard Properties SQL Statistics Dependencies Dependents

Database sessions — Total, Active, Idle

Transactions per second — Transactions, Commits, Rollbacks

Tuples in — Inserts, Updates, Delete

Tuples out — Fetched, Returned

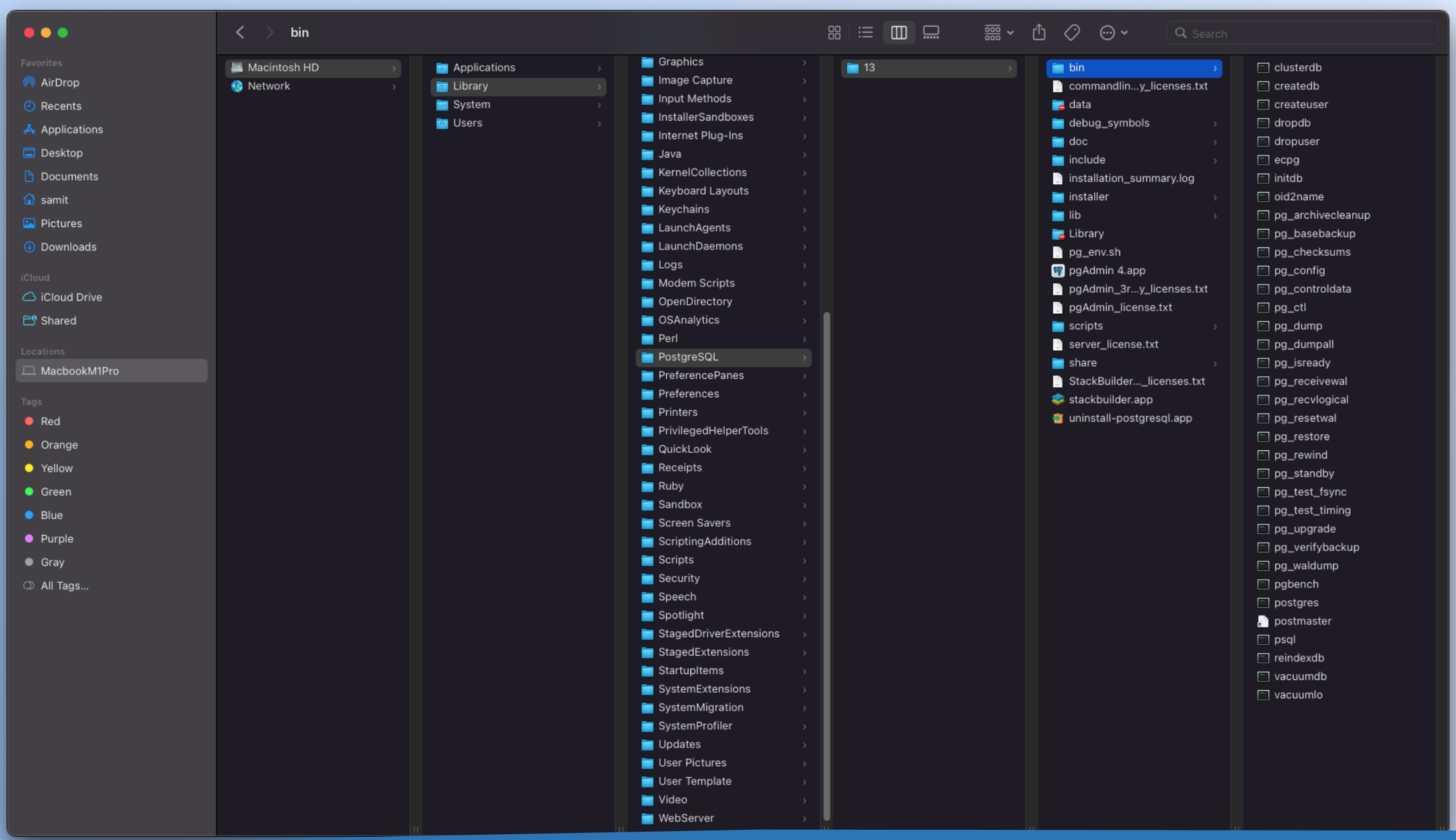
Block I/O — Reads, Hits

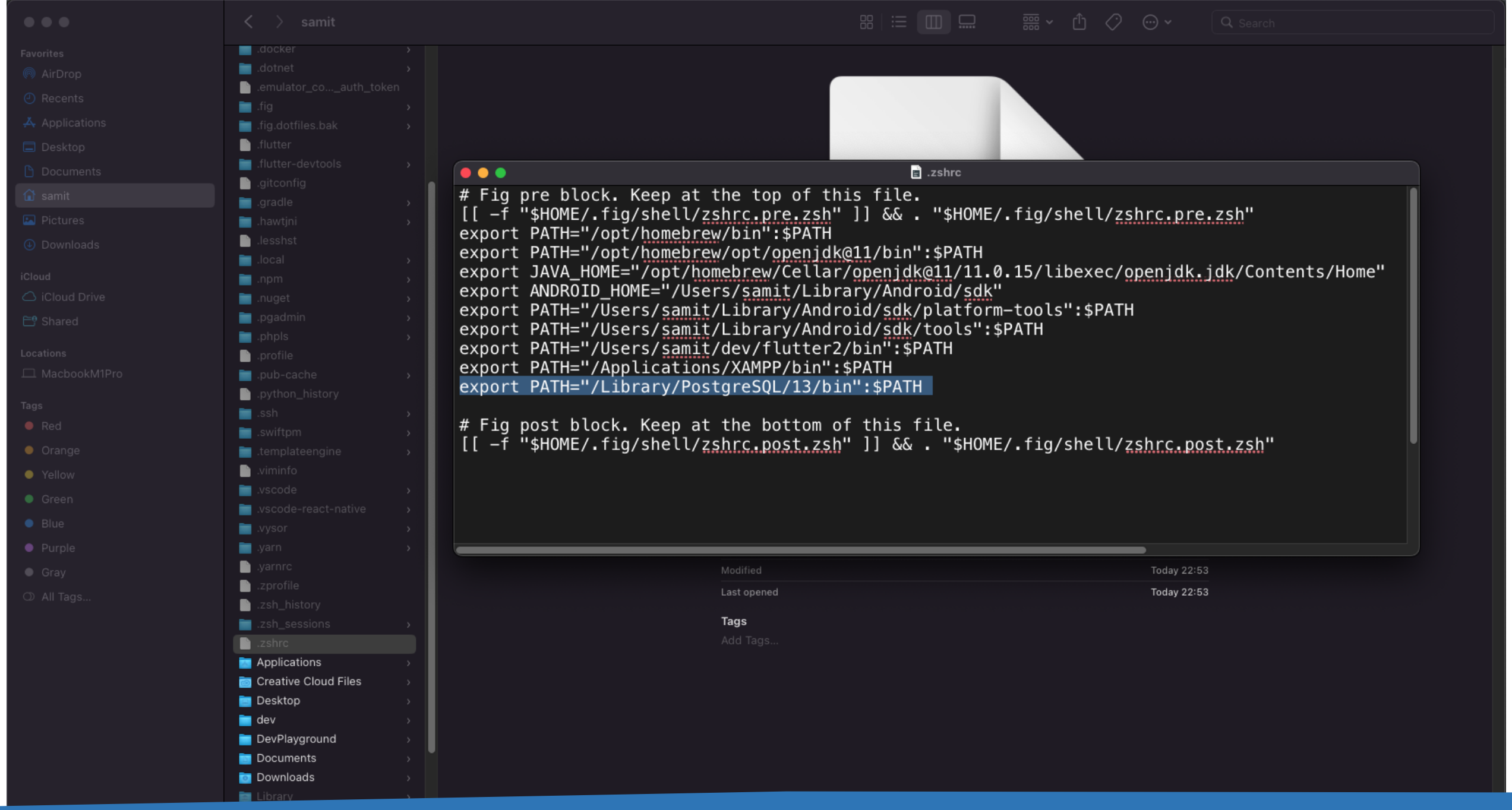
Server activity

Sessions Locks Prepared Transactions

Search

			PID	User	Application	Client	Backend start	Transaction start	State	Wait event	Blocking PIDs
			93488	postgres	pgAdmin 4 - DB:post...	::1	2022-07-15 21:29:12 ...	2022-07-15 21:35:47 ...	active		
			93628				2022-07-15 21:35:47 ...				


bin
Search
Favorites
AirDrop
Recents
Applications
Desktop
Documents
samit
Pictures
Downloads
iCloud
iCloud Drive
Shared
Locations
MacbookM1Pro
Tags
Red
Orange
Yellow
Green
Blue
Purple
Gray
All Tags...
Macintosh HD
Network
Applications
Library
System
Users
Graphics
Image Capture
Input Methods
InstallerSandboxes
Internet Plug-Ins
Java
KernelCollections
Keyboard Layouts
Keychains
LaunchAgents
LaunchDaemons
Logs
Modem Scripts
OpenDirectory
OSAnalytics
Perl
PostgreSQL
PreferencePanes
Preferences
Printers
PrivilegedHelperTools
QuickLook
Receipts
Ruby
Sandbox
Screen Savers
ScriptingAdditions
Scripts
Security
Speech
Spotlight
StagedDriverExtensions
StagedExtensions
StartupItems
SystemExtensions
SystemMigration
SystemProfiler
Updates
User Pictures
User Template
Video
WebServer
13
bin
commandlin...y_licenses.txt
data
debug_symbols
doc
include
installation_summary.log
installer
lib
Library
pg_env.sh
pgAdmin 4.app
pgAdmin_3r...y_licenses.txt
pgAdmin_license.txt
scripts
server_license.txt
share
StackBuilder..._licenses.txt
stackbuilder.app
uninstall-postgresql.app
clusterdb
createdb
createuser
dropdb
dropuser
ecpg
initdb
oid2name
pg_archivecleanup
pg_basebackup
pg_checksums
pg_config
pg_controldata
pg_ctl
pg_dump
pg_dumpall
pg_isready
pg_receivewal
pg_recvlogical
pg_resetwal
pg_restore
pg_rewind
pg_standby
pg_test_fsync
pg_test_timing
pg_upgrade
pg_verifybackup
pg_waldump
pgbench
postgres
postmaster
psql
reindexdb
vacuumdb
vacuumlo


samit
Favorites
AirDrop
Recents
Applications
Desktop
Documents
samit
Pictures
Downloads
iCloud
iCloud Drive
Shared
Locations
MacbookM1Pro
Tags
Red
Orange
Yellow
Green
Blue
Purple
Gray
All Tags...
.docker
.dotnet
.emulator_co..._auth_token
.fig
.fig.dotfiles.bak
.flutter
.flutter-devtools
.gitconfig
.gradle
.hawtjni
.lesshst
.local
.npm
.nuget
.pgadmin
.phpls
.profile
.pub-cache
.python_history
.ssh
.swiftpm
.templateengine
.viminfo
.vscode
.vscode-react-native
.vysor
.yarn
.yarnrc
.zprofile
.zsh_history
.zsh_sessions
.zshrc
Applications
Creative Cloud Files
Desktop
dev
DevPlayground
Documents
Downloads
Library
.zshrc
Fig pre block. Keep at the top of this file.
[[-f "$HOME/.fig/shell/zshrc.pre.zsh"]] && . "$HOME/.fig/shell/zshrc.pre.zsh"
export PATH="/opt/homebrew/bin":$PATH
export PATH="/opt/homebrew/opt/openjdk@11/bin":$PATH
export JAVA_HOME="/opt/homebrew/Cellar/openjdk@11/11.0.15/libexec/openjdk.jdk/Contents/Home"
export ANDROID_HOME="/Users/samit/Library/Android/sdk"
export PATH="/Users/samit/Library/Android/sdk/platform-tools":$PATH
export PATH="/Users/samit/Library/Android/sdk/tools":$PATH
export PATH="/Users/samit/dev/flutter2/bin":$PATH
export PATH="/Applications/XAMPP/bin":$PATH
export PATH="/Library/PostgreSQL/13/bin":$PATH
Fig post block. Keep at the bottom of this file.
[[-f "$HOME/.fig/shell/zshrc.post.zsh"]] && . "$HOME/.fig/shell/zshrc.post.zsh"
Modified
Today 22:53
Last opened
Today 22:53
Tags
Add Tags...

samit — zsh (figterm) ▸ psql — 80×24

```
samit@MacbookM1Pro ~ % psql -U postgres
Password for user postgres:
psql (13.7)
Type "help" for help.

postgres=#
```

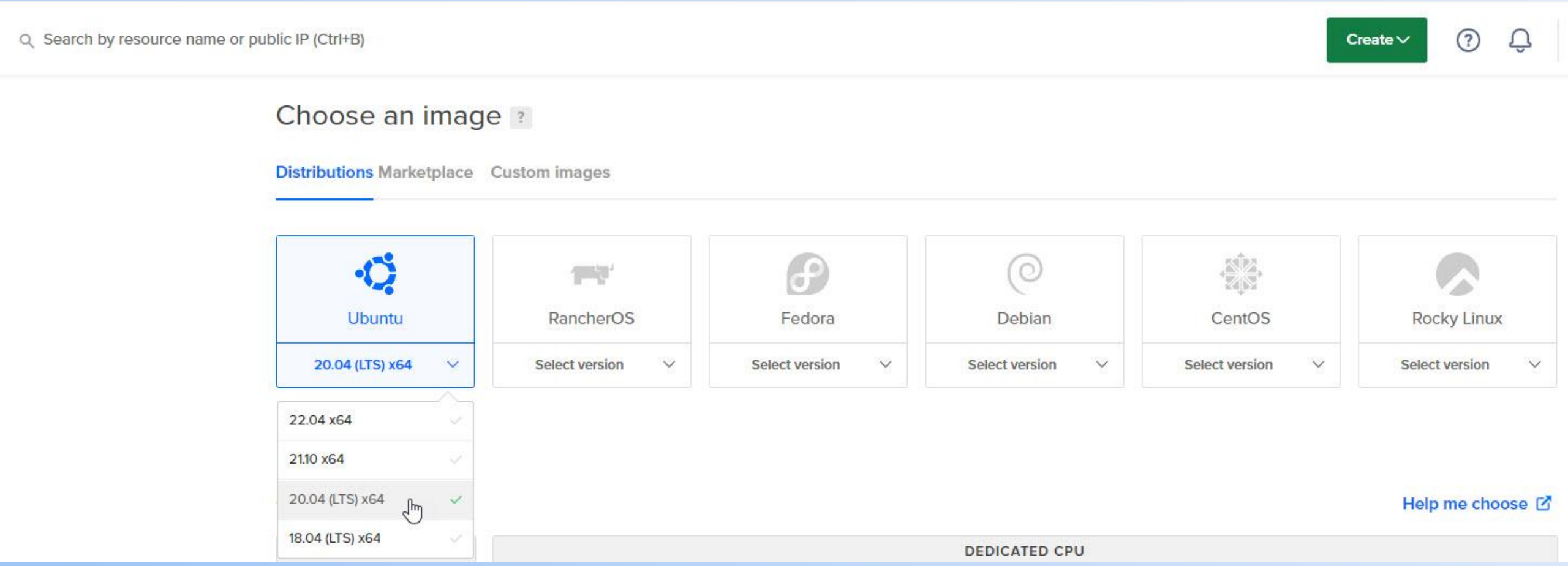
samit — zsh (figterm) ▸ psql — 92×30

```
samit@MacbookM1Pro ~ % psql -U postgres
Password for user postgres:
psql (13.7)
Type "help" for help.

postgres=# \l
                              List of databases
   Name    |  Owner   | Encoding | Collate | Ctype |   Access privileges
-----------+----------+----------+---------+-------+-----------------------
 postgres  | postgres | UTF8     | C       | C     |
 template0 | postgres | UTF8     | C       | C     | =c/postgres          +
           |          |          |         |       | postgres=CTc/postgres
 template1 | postgres | UTF8     | C       | C     | =c/postgres          +
           |          |          |         |       | postgres=CTc/postgres
(3 rows)

postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

postgres=# \dp
                             Access privileges
 Schema | Name | Type | Access privileges | Column privileges | Policies
--------+------+------+-------------------+-------------------+----------
(0 rows)

postgres=#
```

PostgreSQL

ติดตั้ง PostgreSQL 16.x/17.x

ubuntu


Search by resource name or public IP (Ctrl+B)
Create
Choose an image
Distributions Marketplace Custom images
Ubuntu
20.04 (LTS) x64
RancherOS
Select version
Fedora
Select version
Debian
Select version
CentOS
Select version
Rocky Linux
Select version
22.04 x64
21.10 x64
20.04 (LTS) x64
18.04 (LTS) x64
Help me choose
DEDICATED CPU

Choose a plan

Help me choose

SHARED CPU	DEDICATED CPU			
Basic	General Purpose	CPU-Optimized	Memory-Optimized	Storage-Optimized

Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

CPU options: Regular with SSD | Premium Intel with NVMe SSD NEW | Premium AMD with NVMe SSD NEW

$4/mo $0.006/hour	$6/mo $0.009/hour	$12/mo $0.018/hour	$18/mo $0.027/hour	$24/mo $0.036/hour	$48/mo $0.071/hour
512 MB / 1 CPU	1 GB / 1 CPU	2 GB / 1 CPU	2 GB / 2 CPUs	4 GB / 2 CPUs	8 GB / 4 CPUs
10 GB SSD Disk	25 GB SSD Disk	50 GB SSD Disk	60 GB SSD Disk	80 GB SSD Disk	160 GB SSD Disk
500 GB transfer	1000 GB transfer	2 TB transfer	3 TB transfer	4 TB transfer	5 TB transfer

Show all plans

Choose a datacenter region

New York	San Francisco	Amsterdam	Singapore	London	Frankfurt
1, 2, 3	1, 2, 3	2, 3	1	1	1

Toronto	Bangalore
1	1

VPC Network

default-sgp1 DEFAULT

All resources created in this datacenter will be members of the same VPC network. They can communicate securely over their Private IP addresses. What does this mean?

Authentication ?

SSH keys
A more secure authentication method

Password
Create a root password to access Droplet (less secure)

Create root password *

PASSWORD REQUIREMENTS

- ✓ Must be at least 8 characters long
- ✓ Must contain 1 uppercase letter (cannot be first or last character)
- ✓ Must contain 1 number
- ✓ Cannot end in a number or special character

⚠ Please store your password securely. You will not be sent an email containing the Droplet's details or password.

Finalize and create

How many Droplets?

Deploy multiple Droplets with the same configuration.

— 1 Droplet +

Choose a hostname

Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

ubuntu-docker-lab

Add tags

Use tags to organize and relate resources. Tags may contain letters, numbers, colons, dashes, and underscores.

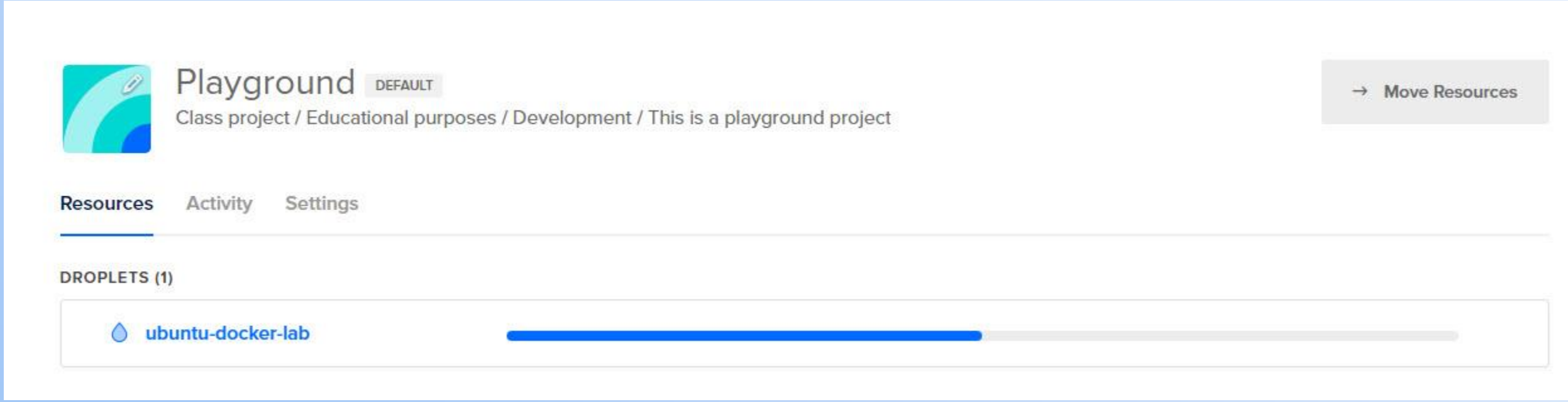
ubuntu-docker-lab Type tags here

Select Project

Assign Droplets to a project

Playground

Create Droplet


Playground
DEFAULT
Class project / Educational purposes / Development / This is a playground project
Move Resources
Resources
Activity
Settings
DROPLETS (1)
ubuntu-docker-lab

Playground DEFAULT

Class project / Educational purposes / Development / This is a playground project

→ Move Resources

Resources Activity Settings

DROPLETS (1)

ubuntu-docker-lab
SGP1 / 1GB / 25GB Disk

174.138.16.96 Copy

ubuntu-docker-lab

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: cmd

```
$ ssh root@174.138.16.96
The authenticity of host '174.138.16.96 (174.138.16.96)' can't be established.
ECDSA key fingerprint is SHA256:+8HzIjqbs8KERRa/4Ut6w1zed1vnagOn+dDdhCPryFk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: cmd

```
Microsoft Windows [Version 10.0.19044.1826]

$ ssh root@174.138.16.96
The authenticity of host '174.138.16.96 (174.138.16.96)' can't be established.
ECDSA key fingerprint is SHA256:+8HzIjqbs8KERRa/4Ut6w1zed1vnagOn+dDdhCPryFk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '174.138.16.96' (ECDSA) to the list of known hosts.
root@174.138.16.96's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-122-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Jul 19 17:03:52 UTC 2022

  System load:  0.0               Users logged in:       0
  Usage of /:   6.3% of 24.05GB   IPv4 address for eth0: 174.138.16.96
  Memory usage: 21%               IPv4 address for eth0: 10.15.0.5
  Swap usage:   0%                IPv4 address for eth1: 10.104.0.2
  Processes:    104

8 updates can be applied immediately.
7 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# sudo apt update
Hit:1 http://mirrors.digitalocean.com/ubuntu focal InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu focal-updates InRelease
Hit:3 https://download.docker.com/linux/ubuntu focal InRelease
Hit:4 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:5 http://mirrors.digitalocean.com/ubuntu focal-backports InRelease
Hit:6 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# sudo apt install curl gpg gnupg2 software-properties-common apt-transport-https lsb-release ca-certi
ficates
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu2).
lsb-release set to manually installed.
ca-certificates is already the newest version (20211016~20.04.1).
curl is already the newest version (7.68.0-1ubuntu2.12).
gpg is already the newest version (2.2.19-3ubuntu2.2).
gpg set to manually installed.
software-properties-common is already the newest version (0.99.9.8).
software-properties-common set to manually installed.
apt-transport-https is already the newest version (2.0.9).
gnupg2 is already the newest version (2.2.19-3ubuntu2.2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" |sudo tee  /etc/apt/sources.list.d/pgdg.list
deb http://apt.postgresql.org/pub/repos/apt/ focal-pgdg main
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu focal InRelease
Hit:3 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:4 http://mirrors.digitalocean.com/ubuntu focal-updates InRelease
Hit:5 http://mirrors.digitalocean.com/ubuntu focal-backports InRelease
Get:6 http://apt.postgresql.org/pub/repos/apt focal-pgdg InRelease [91.7 kB]
Hit:7 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:8 http://apt.postgresql.org/pub/repos/apt focal-pgdg/main amd64 Packages [240 kB]
Fetched 331 kB in 1s (264 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# systemctl status postgresql@13-main.service
● postgresql@13-main.service - PostgreSQL Cluster 13-main
     Loaded: loaded (/lib/systemd/system/postgresql@.service; enabled-runtime; vendor preset: enabled)
     Active: active (running) since Tue 2022-07-19 19:17:22 UTC; 34s ago
   Main PID: 4059 (postgres)
      Tasks: 7 (limit: 1131)
     Memory: 17.8M
     CGroup: /system.slice/system-postgresql.slice/postgresql@13-main.service
             ├─4059 /usr/lib/postgresql/13/bin/postgres -D /var/lib/postgresql/13/main -c config_file=/etc/postgresql/13/main>
             ├─4061 postgres: 13/main: checkpointer
             ├─4062 postgres: 13/main: background writer
             ├─4063 postgres: 13/main: walwriter
             ├─4064 postgres: 13/main: autovacuum launcher
             ├─4065 postgres: 13/main: stats collector
             └─4066 postgres: 13/main: logical replication launcher

Jul 19 19:17:20 ubuntu-docker-lab systemd[1]: Starting PostgreSQL Cluster 13-main...
Jul 19 19:17:22 ubuntu-docker-lab systemd[1]: Started PostgreSQL Cluster 13-main.
lines 1-17/17 (END)
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# systemctl status postgresql@13-main.service
● postgresql@13-main.service - PostgreSQL Cluster 13-main
     Loaded: loaded (/lib/systemd/system/postgresql@.service; enabled-runtime; vendor preset: enabled)
     Active: active (running) since Tue 2022-07-19 19:17:22 UTC; 34s ago
   Main PID: 4059 (postgres)
      Tasks: 7 (limit: 1131)
     Memory: 17.8M
     CGroup: /system.slice/system-postgresql.slice/postgresql@13-main.service
             ├─4059 /usr/lib/postgresql/13/bin/postgres -D /var/lib/postgresql/13/main -c config_file=/etc/postgresql/13/main
             ├─4061 postgres: 13/main: checkpointer
             ├─4062 postgres: 13/main: background writer
             ├─4063 postgres: 13/main: walwriter
             ├─4064 postgres: 13/main: autovacuum launcher
             ├─4065 postgres: 13/main: stats collector
             └─4066 postgres: 13/main: logical replication launcher

Jul 19 19:17:20 ubuntu-docker-lab systemd[1]: Starting PostgreSQL Cluster 13-main...
Jul 19 19:17:22 ubuntu-docker-lab systemd[1]: Started PostgreSQL Cluster 13-main.
root@ubuntu-docker-lab:~# sudo su - postgres
postgres@ubuntu-docker-lab:~$
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
postgres@ubuntu-docker-lab:~$ psql -c "alter user postgres with password '123456'"
ALTER ROLE
postgres@ubuntu-docker-lab:~$
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE PORTS COMMENTS TERMINAL ssh

```
postgres@ubuntu-postgres-sample:~$ psql
psql (15.4 (Ubuntu 15.4-2.pgdg20.04+1))
Type "help" for help.

postgres=#
```

```
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql" at port "5432".
postgres=#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
postgres=# CREATE DATABASE mytestdb;
CREATE DATABASE
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding | Collate | Ctype |   Access privileges
-----------+----------+----------+---------+-------+-----------------------
 mytestdb  | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
 postgres  | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
 template0 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
           |          |          |         |         | postgres=CTc/postgres
 template1 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
           |          |          |         |         | postgres=CTc/postgres
(4 rows)
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
postgres=# CREATE USER mytestuser WITH ENCRYPTED PASSWORD '1234';
CREATE ROLE
postgres=#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE mytestdb to mytestuser;
GRANT
postgres=#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
postgres=# \du
                                   List of roles
 Role name  |                         Attributes                         | Member of
------------+------------------------------------------------------------+-----------
 mytestuser |                                                            | {}
 postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

postgres=#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
  GNU nano 4.8                    /etc/postgresql/13/main/postgresql.conf                    Modified
hba_file = '/etc/postgresql/13/main/pg_hba.conf'        # host-based authentication file
                                        # (change requires restart)
ident_file = '/etc/postgresql/13/main/pg_ident.conf'    # ident configuration file
                                        # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/13-main.pid'                   # write an extra PID file
                                        # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
  GNU nano 4.8                    /etc/postgresql/13/main/pg_hba.conf                    Modified
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.




# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             0.0.0.0/0               md5
# IPv6 local connections:
host    all             all             ::/0                    md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            md5
host    replication     all             ::1/128                 md5


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text
```

main Go 1.17.3 ⊗ 14 ⚠ 0 Select Postgres Server Go Live ⚠ Go Update Available

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# sudo systemctl restart postgresql
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# netstat  -tunelp | grep 5432
tcp        0      0 0.0.0.0:5432            0.0.0.0:*               LISTEN      112        37708      5440/postgres
tcp6       0      0 :::5432                 :::*                    LISTEN      112        37709      5440/postgres
root@ubuntu-docker-lab:~#
```

Command Prompt - psql -U p

```
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\samit>psql -U postgres -h 174.138.16.96 -p 5432 postgres
Password for user postgres:
psql (13.4, server 13.7 (Ubuntu 13.7-1.pgdg20.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=# \l
                              List of databases
   Name    |  Owner   | Encoding | Collate |  Ctype  |    Access privileges
-----------+----------+----------+---------+---------+-----------------------
 mytestdb  | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =Tc/postgres          +
           |          |          |         |         | postgres=CTc/postgres  +
           |          |          |         |         | mytestuser=CTc/postgres
 postgres  | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
 template0 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres           +
           |          |          |         |         | postgres=CTc/postgres
 template1 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres           +
           |          |          |         |         | postgres=CTc/postgres
(4 rows)
```

คำสั่งแสดงหมายเลขเวอร์ชั่นของ PostgreSQL

```
C:\Users\SamitPCGenius
λ postgres --version
postgres (PostgreSQL) 13.1
```

```
C:\Users\SamitPCGenius
λ psql --version
psql (PostgreSQL) 13.1
```

```
C:\Users\SamitPCGenius
λ psql -U postgres
Password for user postgres:
psql (13.1)
Type "help" for help.

postgres=# show server_version;
 server_version
----------------
 13.1
(1 row)
```

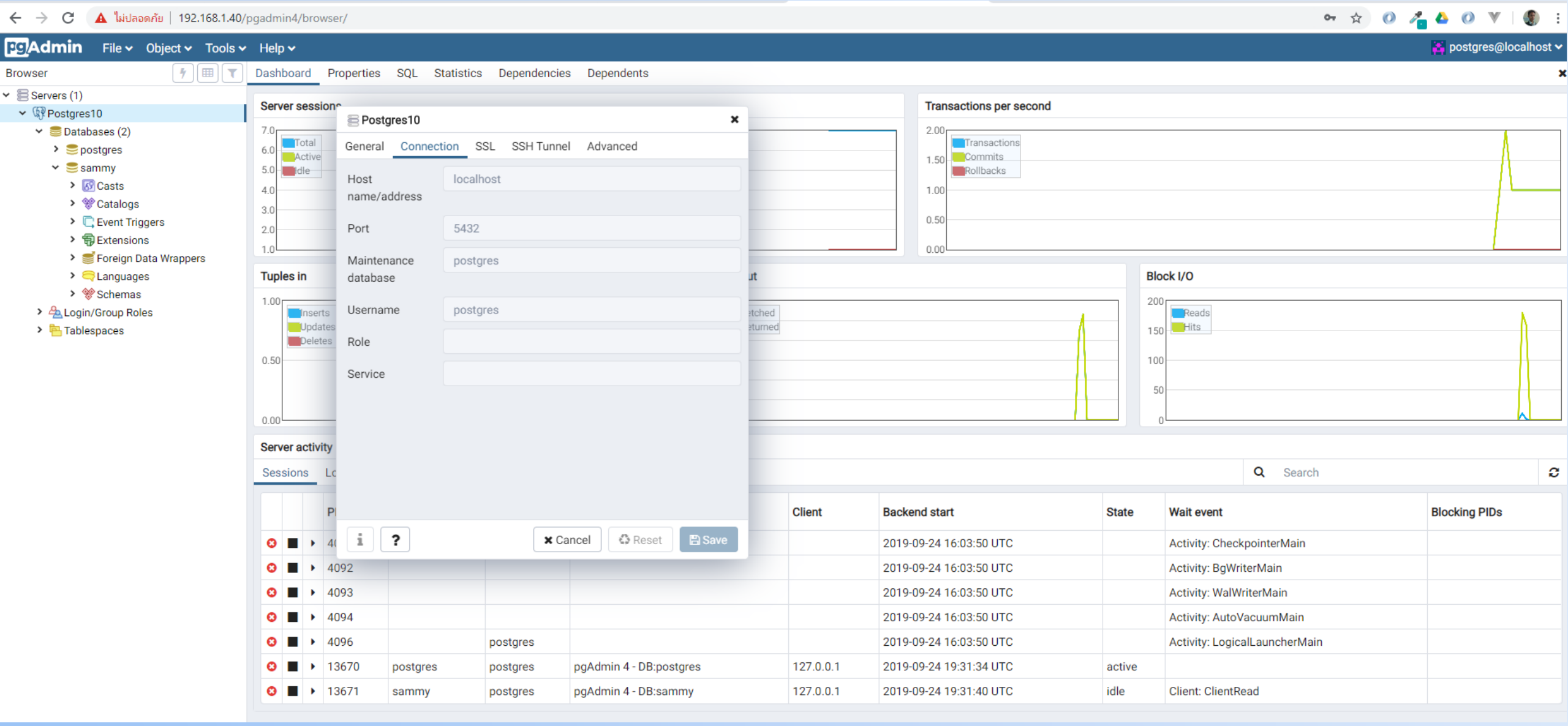
```
postgres=# SELECT version();
                              version
-----------------------------------------------------------------
 PostgreSQL 13.1, compiled by Visual C++ build 1914, 64-bit
(1 row)
```

การติดตั้ง pgAdmin4 on Ubuntu 20.04

```
sudo apt update -y && sudo apt upgrade -y
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" >>
/etc/apt/sources.list.d/pgdg.list'

sudo apt update
sudo apt install pgadmin4 pgadmin4-apache2 -y
```

กำหนดอีเมล์ postgres@localhost และรหัสผ่าน 377040

เข้าใช้งาน pgAdmin4
http://192.168.1.40/pgadmin4/


ไม่ปลอดภัย | 192.168.1.40/pgadmin4/browser/
pgAdmin
File
Object
Tools
Help
postgres@localhost
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Servers (1)
Postgres10
Databases (2)
postgres
sammy
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas
Login/Group Roles
Tablespaces
Server sessions
Total
Active
Idle
Transactions per second
Transactions
Commits
Rollbacks
Tuples in
Inserts
Updates
Deletes
Block I/O
Reads
Hits
Postgres10
General
Connection
SSL
SSH Tunnel
Advanced
Host name/address
localhost
Port
5432
Maintenance database
postgres
Username
postgres
Role
Service
Cancel
Reset
Save
Server activity
Sessions
Search
Client
Backend start
State
Wait event
Blocking PIDs
2019-09-24 16:03:50 UTC
Activity: CheckpointerMain
4092
Activity: BgWriterMain
4093
Activity: WalWriterMain
4094
Activity: AutoVacuumMain
4096
postgres
Activity: LogicalLauncherMain
13670
postgres
postgres
pgAdmin 4 - DB:postgres
127.0.0.1
2019-09-24 19:31:34 UTC
active
13671
sammy
postgres
pgAdmin 4 - DB:sammy
127.0.0.1
2019-09-24 19:31:40 UTC
idle
Client: ClientRead

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# curl  -fsSL https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo gpg --dearmor -o /etc/apt
/trusted.gpg.d/pgadmin.gpg
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# curl  -fsSL https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo gpg --dearmor -o /etc/apt
/trusted.gpg.d/pgadmin.gpg
root@ubuntu-docker-lab:~# sudo sh -c 'echo "deb https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin
4 main" > /etc/apt/sources.list.d/pgadmin4.list'
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# curl  -fsSL https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo gpg --dearmor -o /etc/apt
/trusted.gpg.d/pgadmin.gpg
root@ubuntu-docker-lab:~# sudo sh -c 'echo "deb https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin
4 main" > /etc/apt/sources.list.d/pgadmin4.list'
root@ubuntu-docker-lab:~# cat /etc/apt/sources.list.d/pgadmin4.list
deb https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/focal pgadmin4 main
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu focal InRelease
Hit:3 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:4 http://mirrors.digitalocean.com/ubuntu focal-updates InRelease
Hit:5 http://mirrors.digitalocean.com/ubuntu focal-backports InRelease
Get:6 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:7 http://apt.postgresql.org/pub/repos/apt focal-pgdg InRelease
Get:8 https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/focal pgadmin4 InRelease [4217 B]
Get:9 https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/focal pgadmin4/main amd64 Packages [4500 B]
Get:10 https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/focal pgadmin4/main all Packages [3093 B]
Fetched 126 kB in 3s (39.4 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@ubuntu-docker-lab:~#
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# sudo apt install pgadmin4
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils cpp cpp-9 fontconfig-config fonts-dejavu-core gcc-9-base
  libapache2-mod-wsgi-py3 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libatomic1 libauthen-sasl-perl
  libdata-dump-perl libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libencode-locale-perl libfile-basedir-perl
  libfile-desktopentry-perl libfile-listing-perl libfile-mimeinfo-perl libfont-afm-perl libfontconfig1 libfontenc1 libgl1
  libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libhtml-form-perl libhtml-format-perl libhtml-parser-perl
  libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl
  libhttp-negotiate-perl libice6 libio-html-perl libio-socket-ssl-perl libio-stringy-perl libipc-system-simple-perl libisl22
  libjansson4 libllvm12 liblua5.2-0 liblwp-mediatypes-perl liblwp-protocol-https-perl libmailtools-perl libmpc3
  libnet-dbus-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libpciaccess0 libsm6 libtie-ixhash-perl
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2022-07-19 19:44:54 UTC; 21s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 8231 (apache2)
      Tasks: 55 (limit: 1131)
     Memory: 15.7M
     CGroup: /system.slice/apache2.service
             ├─8231 /usr/sbin/apache2 -k start
             ├─8233 /usr/sbin/apache2 -k start
             └─8234 /usr/sbin/apache2 -k start

Jul 19 19:44:54 ubuntu-docker-lab systemd[1]: Starting The Apache HTTP Server...
Jul 19 19:44:54 ubuntu-docker-lab apachectl[8220]: AH00558: apache2: Could not reliably determine the server's fully qualifie>
Jul 19 19:44:54 ubuntu-docker-lab systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

Apache2 Ubuntu Default Page: It

Not secure | 174.138.16.96

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual

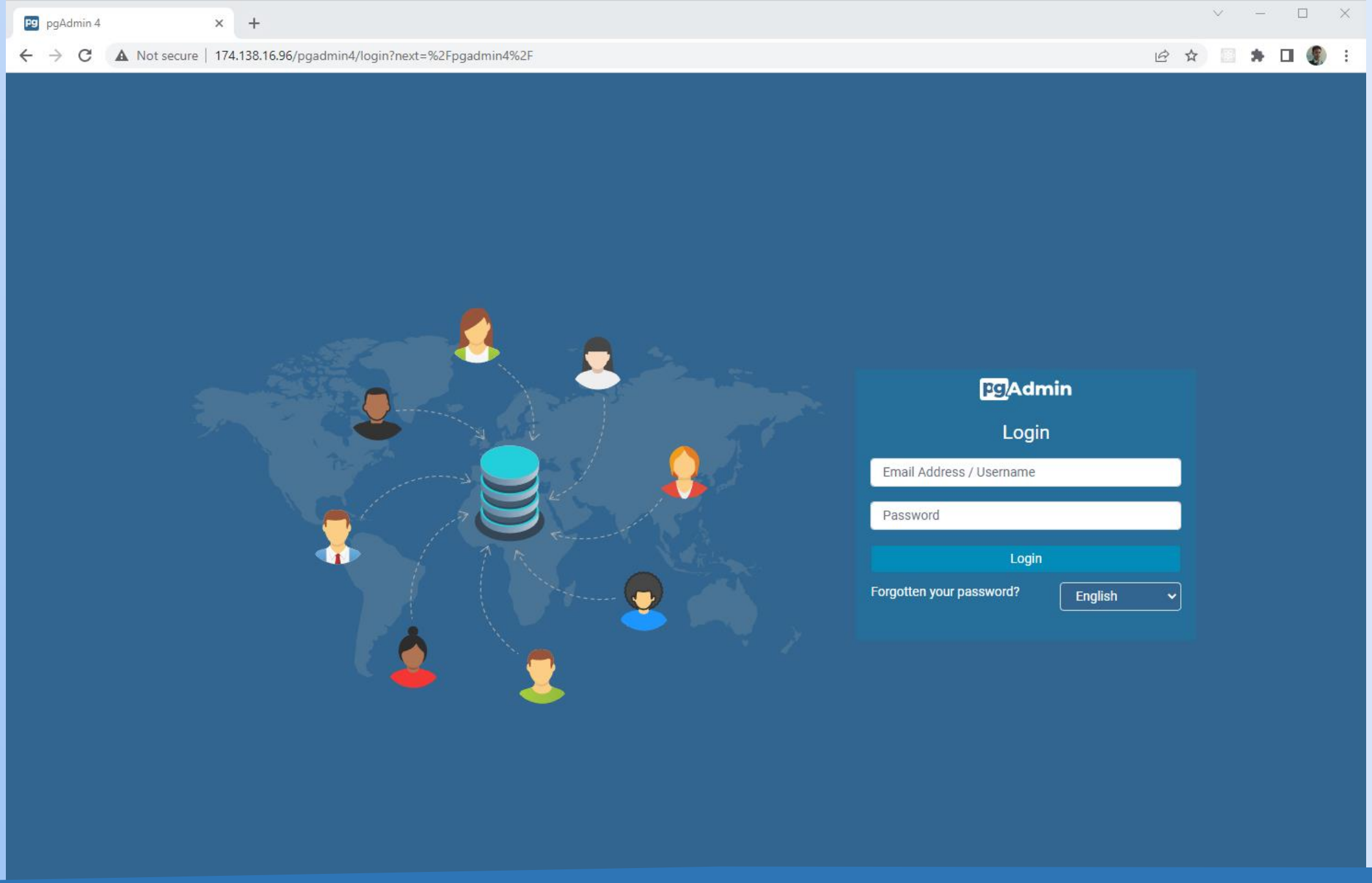
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# sudo /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Debian based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

Enter the email address and password to use for the initial pgAdmin user account:

Email address:
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 1: ssh

```
root@ubuntu-docker-lab:~# sudo /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Debian based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

Enter the email address and password to use for the initial pgAdmin user account:

Email address: samit@email.com
Password:
Retype password:
pgAdmin 4 - Application Initialisation
======================================

Creating storage and log directories...
We can now configure the Apache Web server for you. This involves enabling the wsgi module and configuring the pgAdmin 4 appli
cation to mount at /pgadmin4. Do you wish to continue (y/n)?
```

```
Creating storage and log directories...
We can now configure the Apache Web server for you. This involves enabling the wsgi module and configuring the pgAdmin 4 appli
cation to mount at /pgadmin4. Do you wish to continue (y/n)? y
The Apache web server is running and must be restarted for the pgAdmin 4 installation to complete. Continue (y/n)? y
Apache successfully restarted. You can now start using pgAdmin 4 in web mode at http://127.0.0.1/pgadmin4
root@ubuntu-docker-lab:~#
```


pgAdmin 4
Not secure | 174.138.16.96/pgadmin4/login?next=%2Fpgadmin4%2F
pgAdmin
Login
Email Address / Username
Password
Login
Forgotten your password?
English

pgAdmin 4

Not secure | 174.138.16.96/pgadmin4/browser/

pgAdmin File Object Tools Help samit@email.com (internal)

Browser Dashboard Properties SQL Statistics Dependencies Dependents

Servers

Register
- Server...
- Deploy Cloud Instance...

Create

Refresh...

Remove Server Group

Properties...

pgAdmin

Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Quick Links

Add New Server

Configure pgAdmin

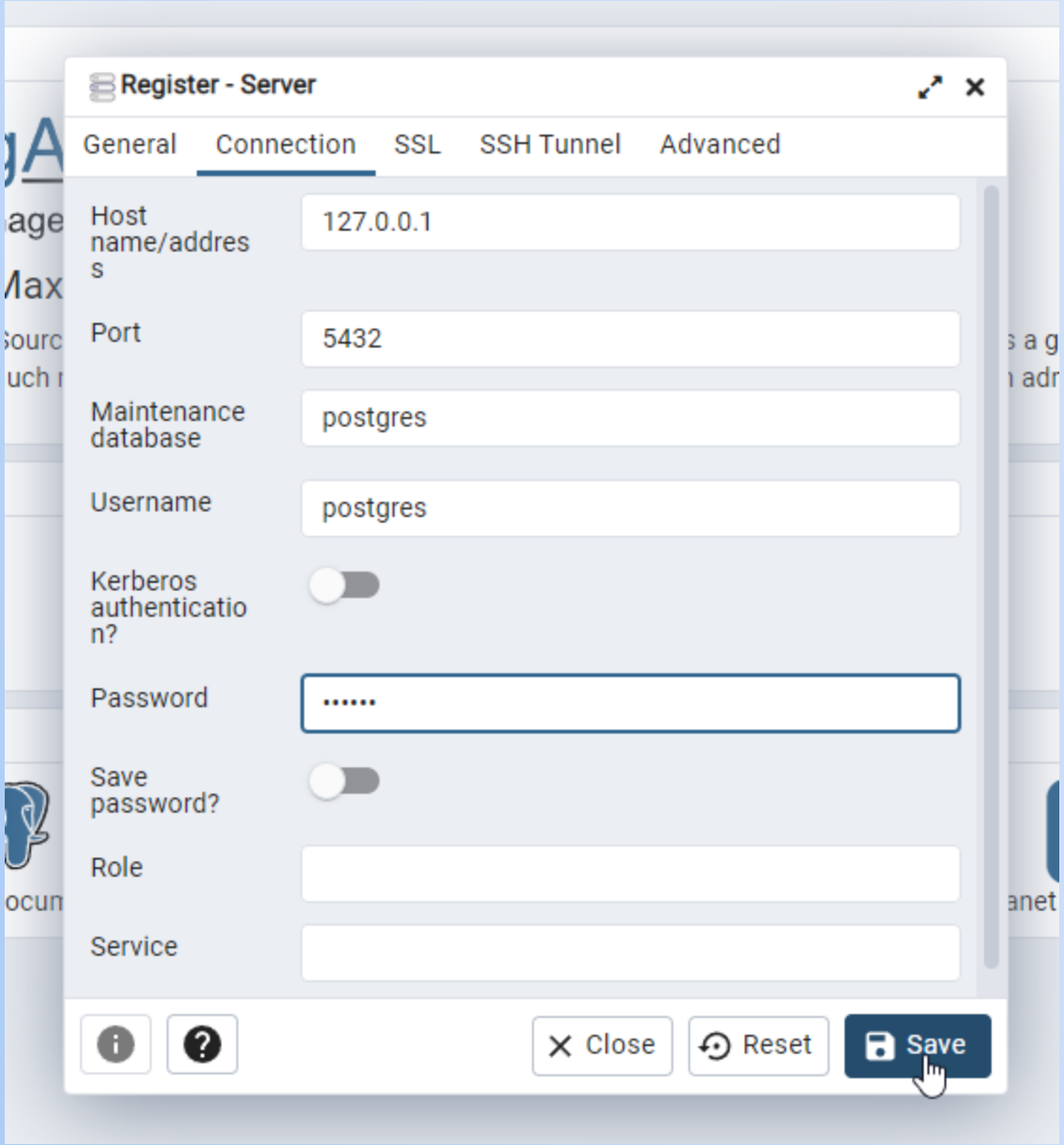
Getting Started

PostgreSQL Documentation

pgAdmin Website

Planet PostgreSQL

Community Support


Register - Server
General
Connection
SSL
SSH Tunnel
Advanced
Host name/address
127.0.0.1
Port
5432
Maintenance database
postgres
Username
postgres
Kerberos authentication?
Password
Save password?
Role
Service
Close
Reset
Save

pgAdmin 4

Not secure | 174.138.16.96/pgadmin4/browser/

pgAdmin File Object Tools Help samit@email.com (internal)

Browser

- Servers (1)
  - PostgreSQL13 Ubuntu
    - Databases (2)
      - mytestdb
      - postgres
        - Casts
        - Catalogs
        - Event Triggers
        - Extensions
        - Foreign Data Wrappers
        - Languages
        - Publications
        - Schemas
        - Subscriptions
    - Login/Group Roles
    - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents

Database sessions — Total, Active, Idle

Transactions per second — Transactions, Commits, Rollbacks

Tuples in — Inserts, Updates, Delete

Tuples out — Fetched, Returned

Block I/O — Reads, Hits

Database activity

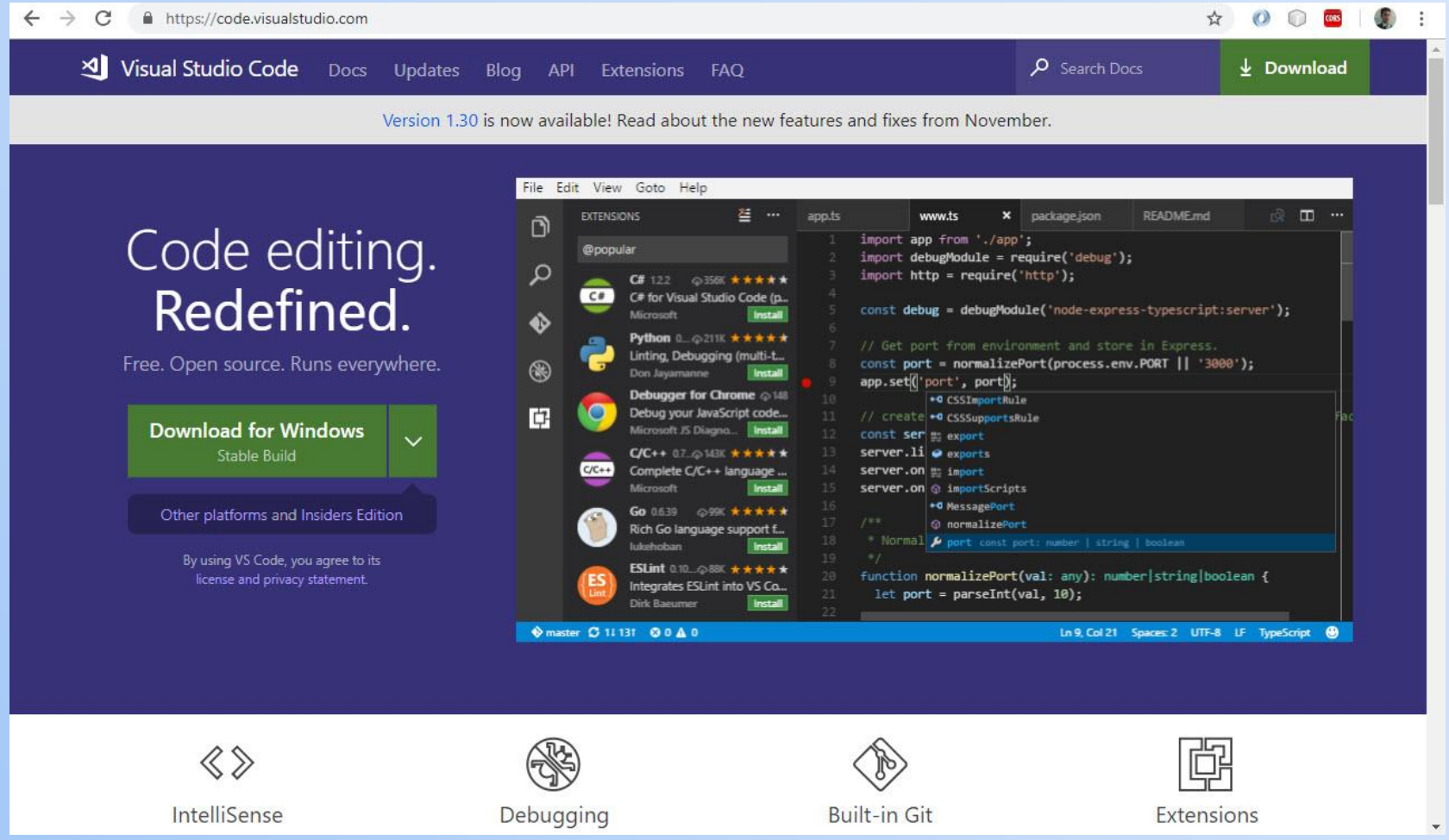
Sessions Locks Prepared Transactions

Search

			PID	User	Application	Client	Backend start	Transaction start	State	Wait event	Blocking PIDs
			5747	postgr...	psql	119.76.30.228	2022-07-19 19:39:44...		idle	Client: ClientRead	
			9711	postgr...	pgAdmin 4 - DB:post...	127.0.0.1	2022-07-19 19:52:39...	2022-07-19 19:52:47...	act...		

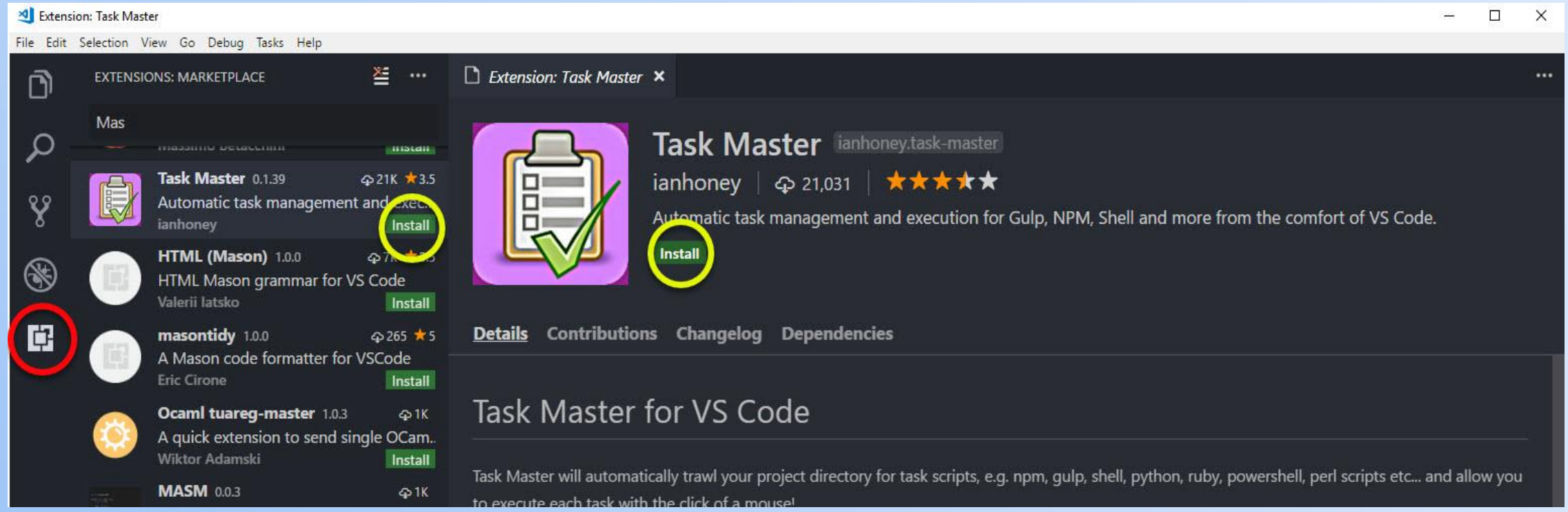
ติดตั้ง Visual Studio Code

ติดตั้ง Visual Studio Code พร้อมส่วนเสริมที่จำเป็น



เข้าไปดาวน์โหลด Visual Studio Code ได้ที่ https://code.visualstudio.com

การติดตั้งส่วนเสริม (Extension) ของ Visual Studio Code



รายชื่อ Extensions ที่แนะนำสำหรับ VS Code

1. **Color Picker** by anseki
2. **AutoFileName** by JerryHong
3. **Material Icon Theme** by pkief.com
4. **MySQL** by Weijen Chen
5. **One Dark Pro** by binaryify

ติดตั้ง Docker Desktop

System Requirements Windows

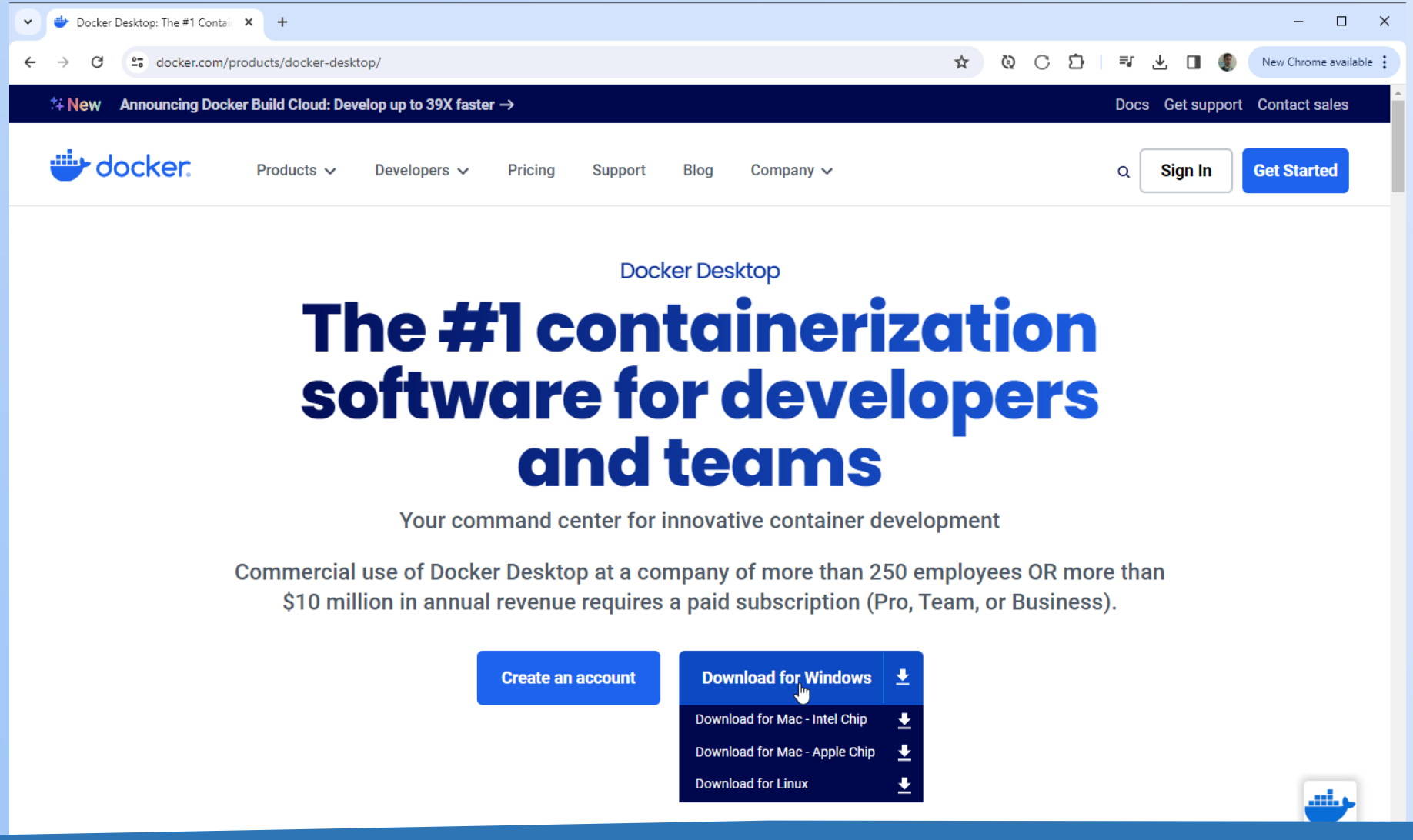
Docker Engine 24.x
Docker Desktop 4.27.x

- Windows 10 64-bit: Pro 2004 (build 19041) or higher, or Enterprise or Education 1909 (build 18363) or higher.
- For Windows 10 Home, see System requirements for WSL 2 backend.
- Hyper-V and Containers Windows features must be enabled.
- The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
- 64 bit processor with Second Level Address Translation (SLAT)
- 4GB system RAM
- BIOS-level hardware virtualization support must be enabled in the BIOS settings

https://docs.docker.com/desktop/windows/install/

Download Docker Desktop for Windows

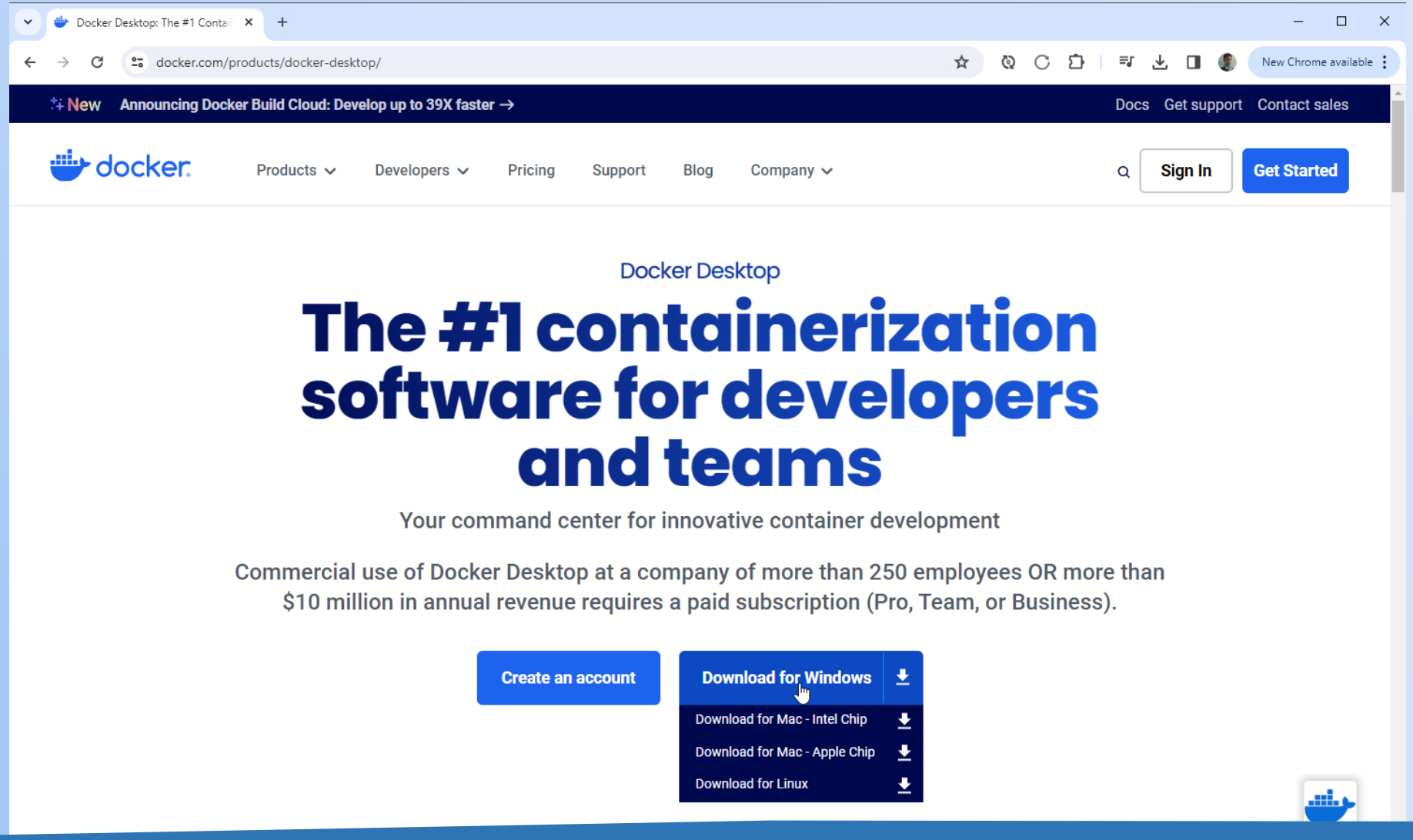
https://www.docker.com/products/docker-desktop



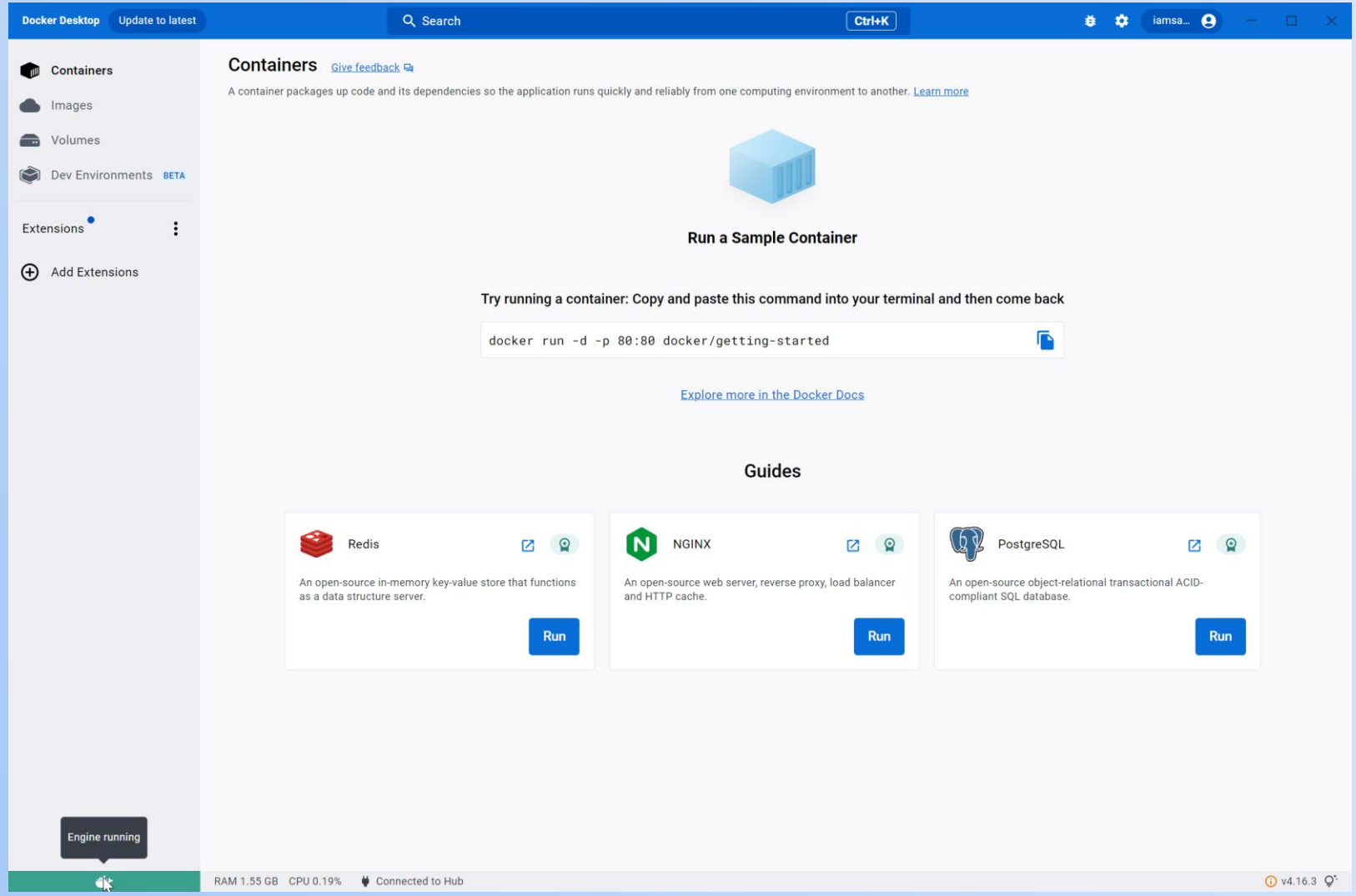
ติดตั้ง Docker Desktop

Download Docker Desktop for Mac

https://www.docker.com/products/docker-desktop



ตัวอย่าง Docker Desktop หลังจากติดตั้งเรียบร้อยแล้ว



Section 3

เรียนรู้วิธีใช้ pgAdmin 4 ในการจัดการ PostgreSQL

- บทนำ pgAdmin 4
- จัดการผู้ใช้และฐานข้อมูล
- ทำงานกับตารางและข้อมูล
- สำรองข้อมูลและกู้คืนข้อมูล

การใช้คำสั่งผ่าน
Command line เบื้องต้น

การล็อกอินเข้า PostgreSQL

```
psql -h <host> -p <port> -U <user> -d <database_name>
```

-h hostname: hostname เซิร์ฟเวอร์ PostgreSQL (default: localhost)
-p port: ระบุ port ของเซิร์ฟเวอร์ PostgreSQL (default: 5432)
-U username: ระบุชื่อผู้ใช้
-d database_name: ระบุชื่อฐานข้อมูล

ตัวอย่าง

```
psql -h localhost -p 5432 -U postgres -d postgres
```

แบบย่อ

```
psql -U postgres
```

คำสั่งพื้นฐานกับ psql

```
\l ดูรายการฐานข้อมูล
\du ดู user
\dn, \dn+ ดูรายการ schema
\c เลือกฐานข้อมูล  ตัวอย่าง \c postgres
\d ดูรายการข้อมูลทั้งหมดใน schema
\dt, \dt+ ดูเฉพาะตาราง
\ds, \ds+ ดูเฉพาะ schema
\d, \d+ <ชื่อตาราง> ดูโครงสร้างตาราง
\! cls , \! clear เคลียร์สกรีน
\q ออกจาก psql
```

การสร้างฐานข้อมูล

ทำได้ 2 วิธี

1. ผ่าน cmd

```
createdb -U postgres -E unicode sampledb
```

2. ผ่าน psql

```
CREATE DATABASE sampledb ENCODING unicode;
```

การสร้างตาราง

```
CREATE TABLE member(
  id serial PRIMARY KEY,
  fullname VARCHAR(50)
);
```

การแก้ไขชื่อตาราง

```
ALTER TABLE member
RENAME TO customer;
```

เพิ่ม column ในตางราง customer

```
ALTER TABLE customer
ADD COLUMN gender VARCHAR(16);
```

เพิ่มคอลัมน์ age เป็นแบบ NOT NULL

```
ALTER TABLE customer
ADD COLUMN age INT NOT NULL;
```

เปลี่ยน Collection ให้ฐานข้อมูล sampledb

```
UPDATE pg_database set encoding =
pg_char_to_encoding('UTF8') WHERE datname =
'sampledb';
```

เปลี่ยน collate กับ ctype

```
UPDATE pg_database SET
datcollate='Thai_Thailand.874',
datctype='Thai_Thailand.874'
WHERE datname='sampledb';
```

collate และ **ctype** เป็นสองตัวเลือกที่ใช้ควบคุมวิธีการเปรียบเทียบและจัดการสตริงใน PostgreSQL ซึ่งมีความแตกต่างกันดังนี้:

collate: กำหนดกฎเกณฑ์สำหรับการเปรียบเทียบตัวอักษร มีผลต่อการเรียงลำดับ การค้นหา และฟังก์ชันการเปรียบเทียบสตริงอื่นๆ ตัวอย่าง collate ทั่วไป:

- **C:** เปรียบเทียบตัวอักษรตามลำดับไบต์ เหมาะสำหรับภาษาที่มีตัวอักษรน้อย
- **en_US.UTF-8:** เปรียบเทียบตัวอักษรตามมาตรฐาน Unicode เหมาะสำหรับภาษาที่มีตัวอักษรมาก

ctype: กำหนดประเภทของตัวอักษร มีผลต่อฟังก์ชันที่ตรวจสอบประเภทตัวอักษร เช่น `isupper()` ตัวอย่าง ctype ทั่วไป:

- **C:** แบ่งตัวอักษรเป็นตัวอักษรพิมพ์ใหญ่ ตัวอักษรพิมพ์เล็ก ตัวเลข และเครื่องหมายวรรคตอน
- **en_US.UTF-8:** แบ่งตัวอักษรตามประเภทภาษา Unicode

ข้อควรระวัง

- ไม่สามารถเปลี่ยน collate และ ctype ของฐานข้อมูลที่มีอยู่ได้
- การเลือก collate และ ctype ที่เหมาะสมจะช่วยให้แอปพลิเคชันทำงานได้อย่างถูกต้อง

เพิ่มข้อมูลใหม่เข้า table customer

```
INSERT INTO customer(fullname,gender,age) VALUES('Samit Koyom','Male', 35);
```

เรียกดูข้อมูล

```
SELECT * FROM customer;
```

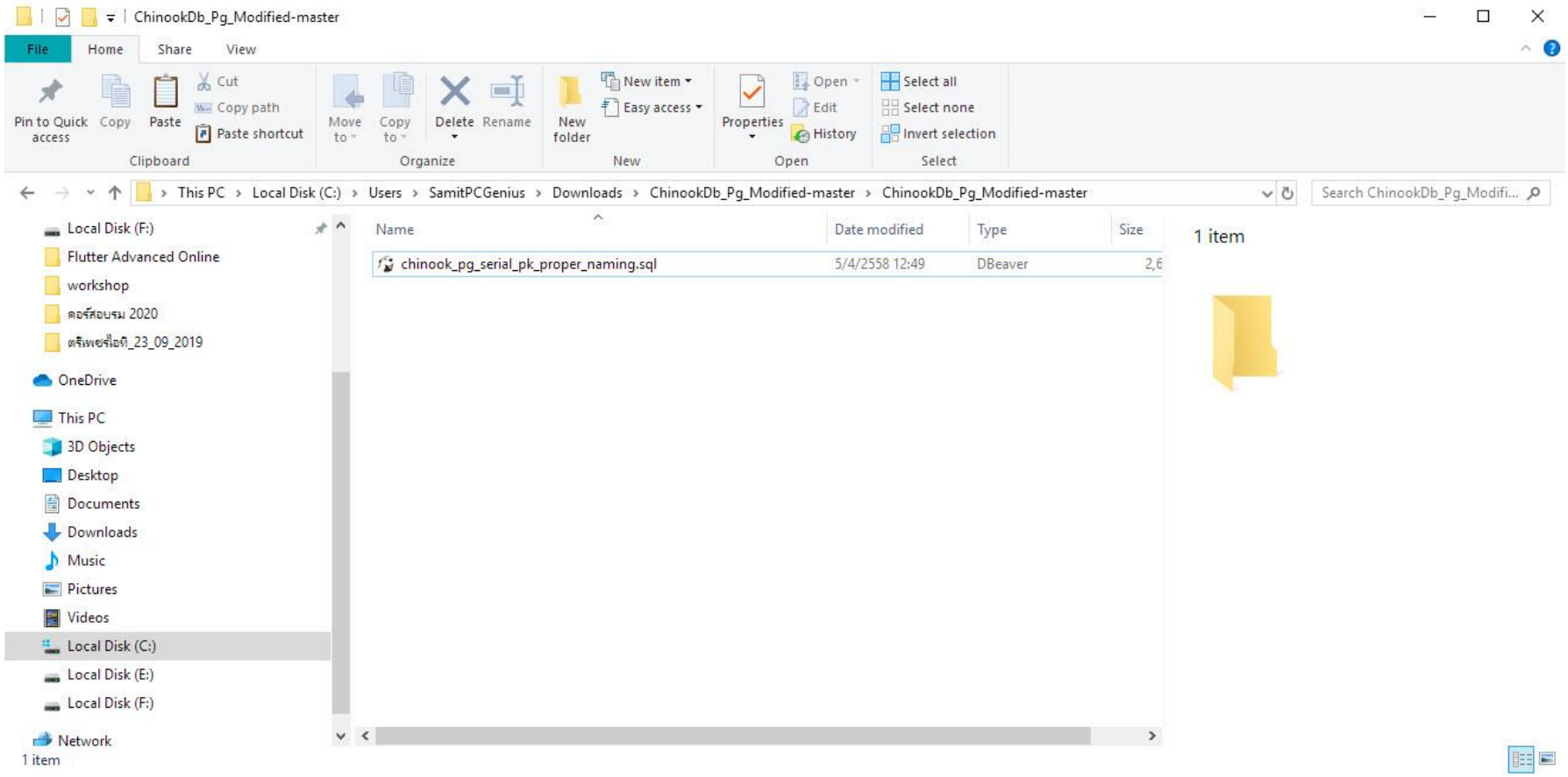
เรียกดูข้อมูล

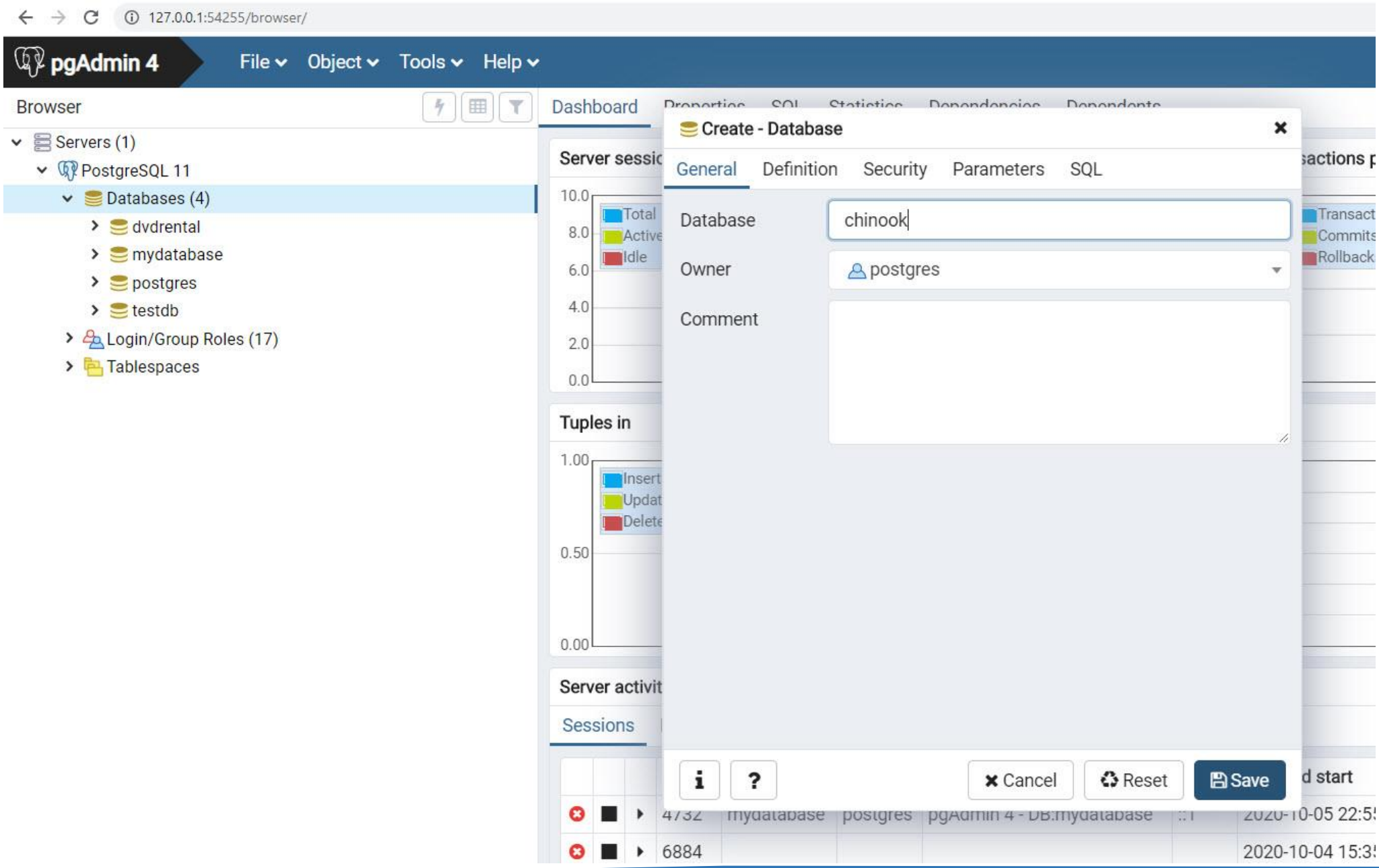
```
SELECT * FROM customer;
```

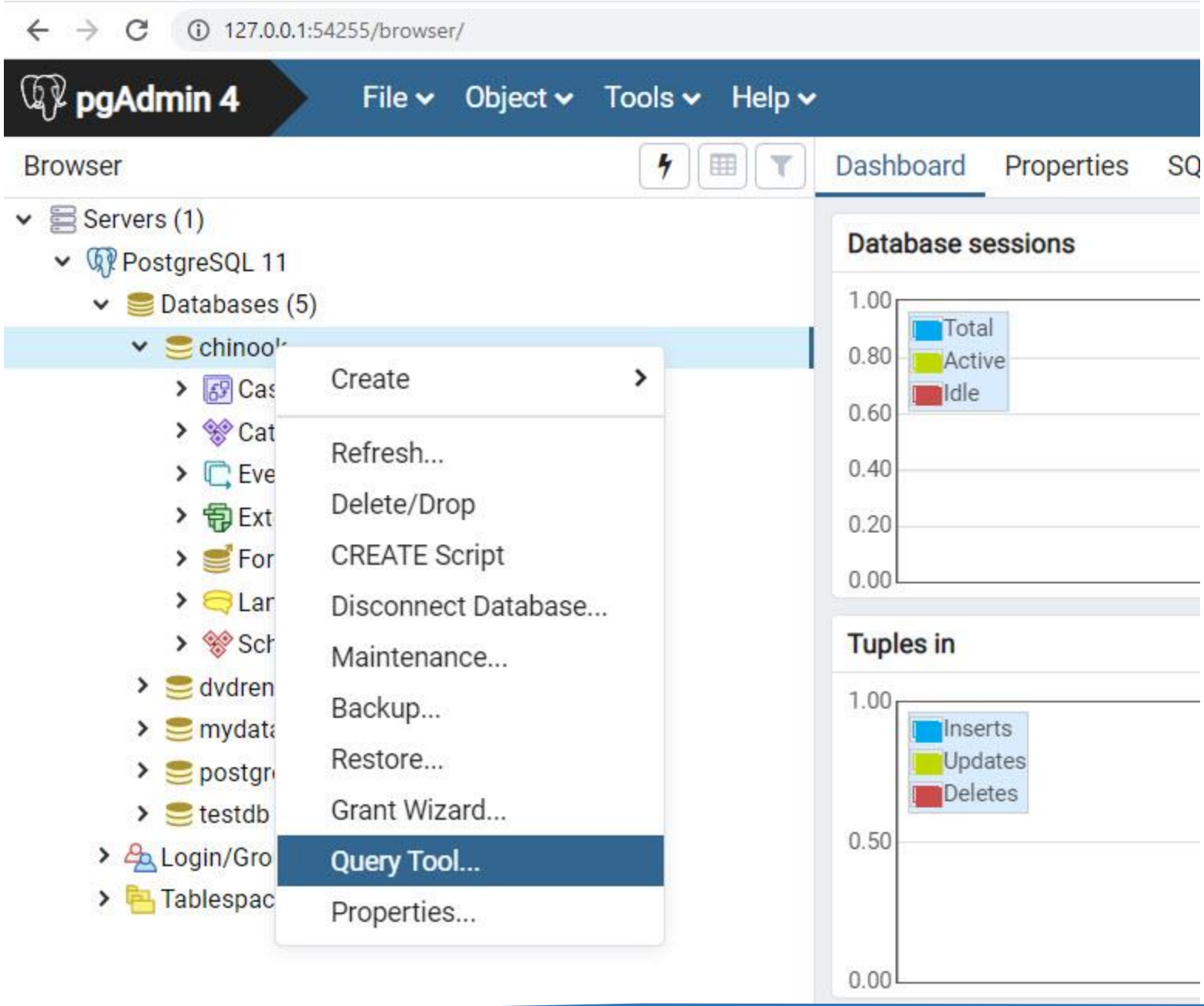
การ Import ข้อมูลผ่าน pgAdmin 4

ฐานข้อมูลตัวอย่าง ChinookDB

ดาวน์โหลด
https://github.com/xivSolutions/ChinookDb_Pg_Modified




127.0.0.1:54255/browser/
pgAdmin 4
File
Object
Tools
Help
Browser
Servers (1)
PostgreSQL 11
Databases (4)
dvdrental
mydatabase
postgres
testdb
Login/Group Roles (17)
Tablespaces
Dashboard
Create - Database
General
Definition
Security
Parameters
SQL
Database
chinook
Owner
postgres
Comment
Cancel
Reset
Save
Tuples in
Server activit
Sessions


127.0.0.1:54255/browser/
pgAdmin 4
File
Object
Tools
Help
Browser
Dashboard
Properties
Servers (1)
PostgreSQL 11
Databases (5)
Create
Refresh...
Delete/Drop
CREATE Script
Disconnect Database...
Maintenance...
Backup...
Restore...
Grant Wizard...
Query Tool...
Properties...
Database sessions
Total
Active
Idle
Tuples in
Inserts
Updates
Deletes

Dashboard Properties SQL Statistics Dependencies Dependents Query - chinook on postgres@PostgreSQL 11

No limit

chinook on postgres@PostgreSQL 11

Query Editor Query History

1

Data Output Explain Messa

Select file

C:\Users\SamitPCGenius\Downloads\ChinookDb_Pg_Mo...

Name	Size	Modified
chinook_pg_serial_pk_proper_naming.sql	2.5 MB	Sun Apr 5 12:49:54 2015

Show hidden files and folders? ☐

Format sql

✖ Cancel Select

Dashboard Properties SQL Statistics Dependencies Dependents chinook_pg_serial_pk_proper_naming.sql

No limit

chinook on postgres@PostgreSQL 11

Query Editor Query History

```sql
-- ----------------------------
-- Sequence structure for album_album_id_seq
-- ----------------------------
CREATE SEQUENCE "album_album_id_seq"
 INCREMENT 1
 MINVALUE 1
 MAXVALUE 2147483647
 START 347
 CACHE 1;
SELECT setval('"public"."album_album_id_seq"', 347, true);


-- ----------------------------
```

Data Output Explain Messages Notifications

127.0.0.1:54255/browser/

pgAdmin 4 File Object Tools Help

Browser

- Extensions
- Foreign Data Wrappers
- Languages
- Schemas (1)
  - public
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Procedures
    - Sequences
    - Tables (16)
      - actor
      - album
      - artist
      - category
      - customer
      - employee
      - film
      - film_actor
      - film_category
      - genre
      - invoice
      - invoice_line

Dashboard Properties SQL Statistics Dependencies Dependents

No limit

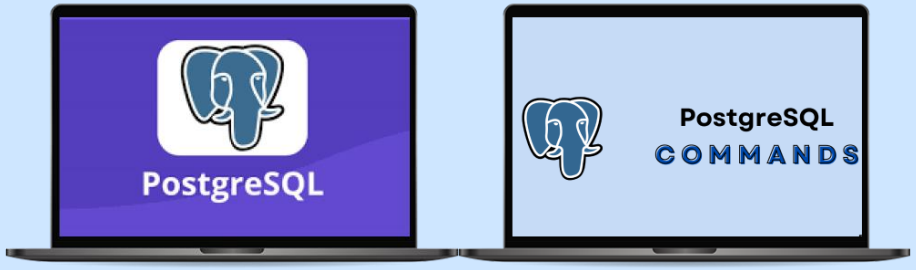
chinook on postgres@PostgreSQL 11

Query Editor Query History

```
-- ----------------------------
-- Sequence structure for album_album_id_seq
-- ----------------------------
CREATE SEQUENCE "album_album_id_seq"
 INCREMENT 1
 MINVALUE 1
 MAXVALUE 2147483647
 START 347
 CACHE 1;
SELECT setval('"public"."album_album_id_seq"', 347, tr


-- ----------------------------
```

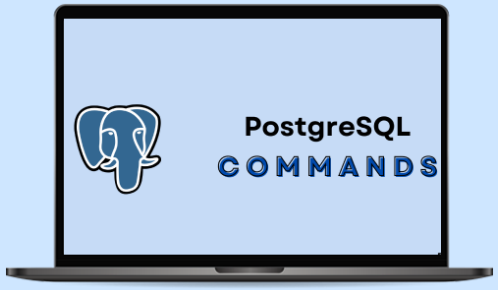
Data Output Explain Messages Notifications

```
ALTER TABLE

Query returned successfully in 1 secs 135 msec.
```



DAY 2

Section 4: คำสั่ง psql พื้นฐานสำหรับการใช้งาน PostgreSQL
Section 5: เรียนรู้วิธีเขียนคิวรี่ SQL พื้นฐาน

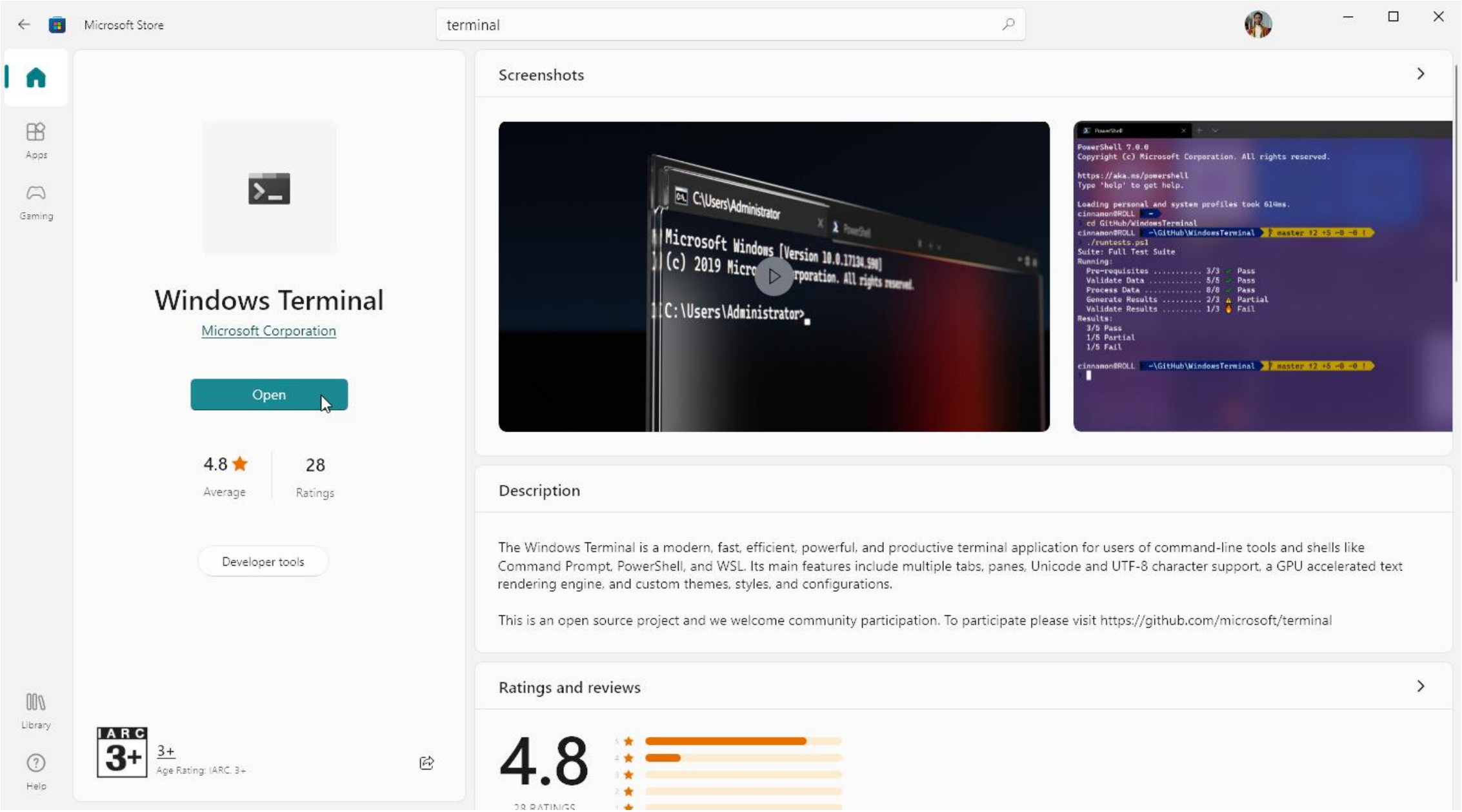


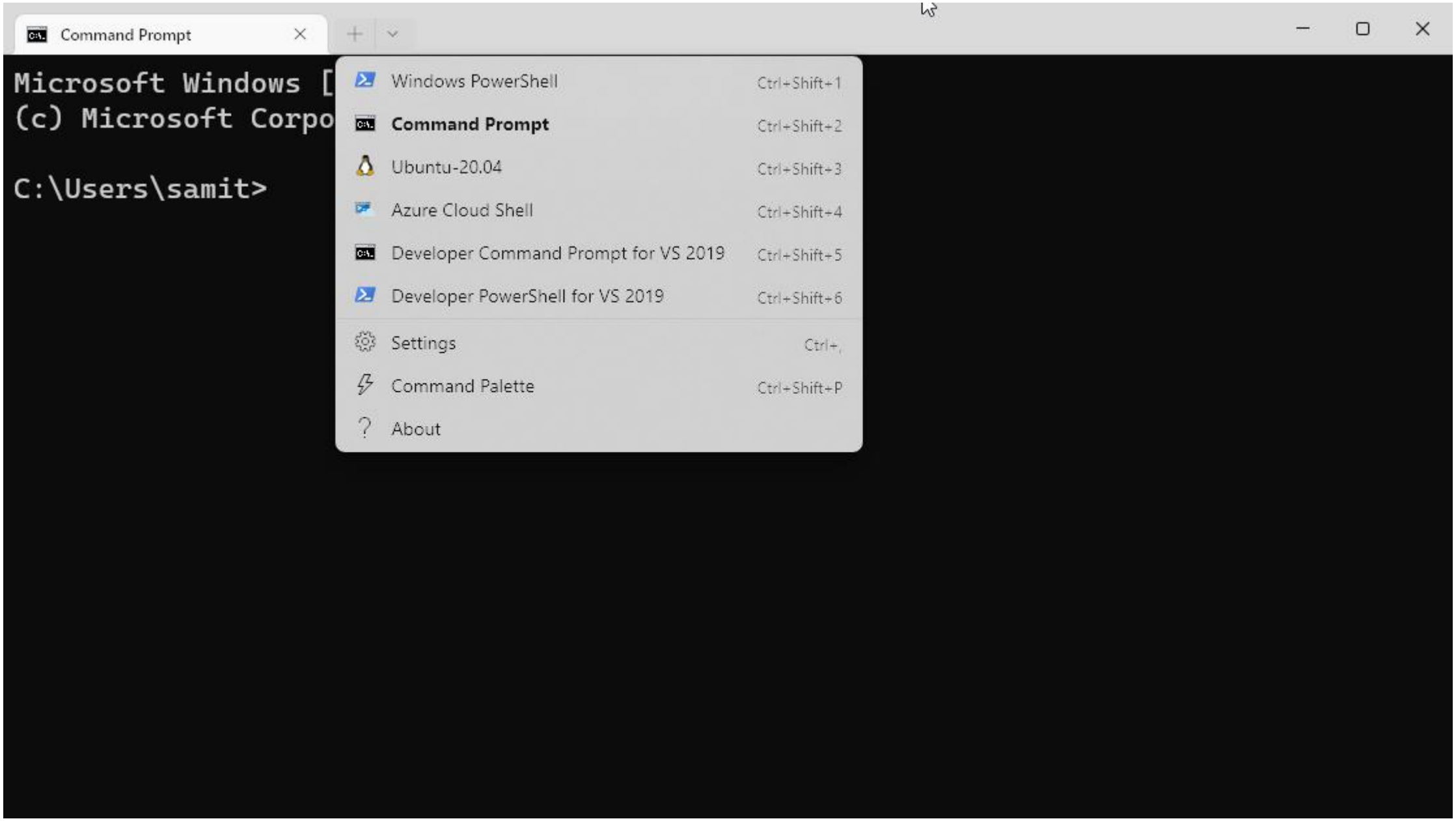
Section 4

เรียนรู้คำสั่ง psql พื้นฐานสำหรับการใช้งาน PostgreSQL

- เชื่อมต่อกับ PostgreSQL ผ่าน psql
- ทำงานกับตารางและข้อมูล
- เขียนคิวรี่ SQL พื้นฐาน
- จัดการสิทธิ์การเข้าถึง (role)
- ตัวอย่างการใช้งาน psql

การใช้ภาษาไทยใน psql.exe command line ผ่าน Windows Terminal


Microsoft Store
terminal
Apps
Gaming
Library
Help
Windows Terminal
Microsoft Corporation
Open
4.8
Average
28
Ratings
Developer tools
3+
Age Rating: IARC 3+
Screenshots
Description
The Windows Terminal is a modern, fast, efficient, powerful, and productive terminal application for users of command-line tools and shells like Command Prompt, PowerShell, and WSL. Its main features include multiple tabs, panes, Unicode and UTF-8 character support, a GPU accelerated text rendering engine, and custom themes, styles, and configurations.
This is an open source project and we welcome community participation. To participate please visit https://github.com/microsoft/terminal
Ratings and reviews
4.8

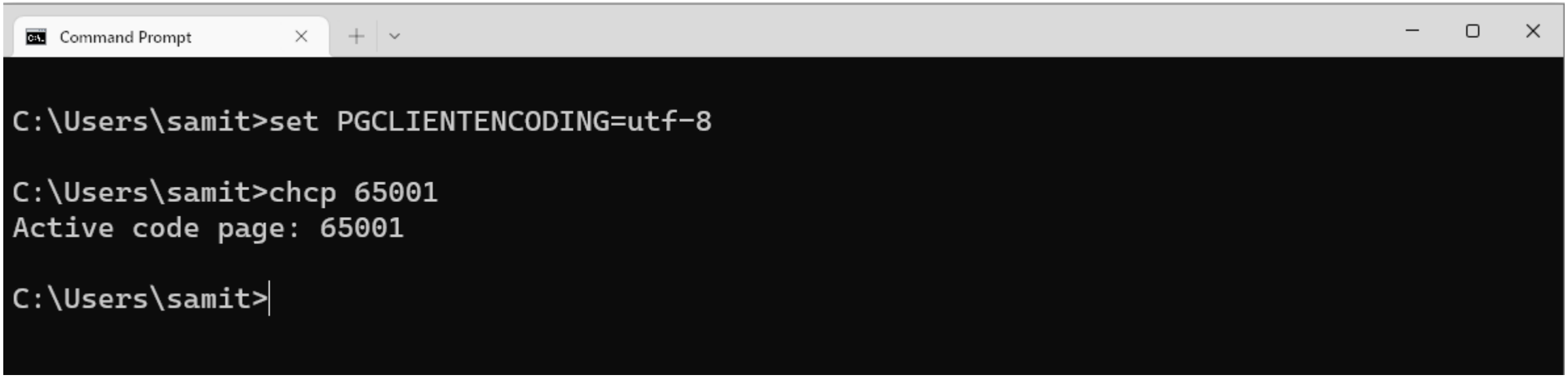

Command Prompt
Microsoft Windows [
(c) Microsoft Corpo
C:\Users\samit>
Windows PowerShell Ctrl+Shift+1
Command Prompt Ctrl+Shift+2
Ubuntu-20.04 Ctrl+Shift+3
Azure Cloud Shell Ctrl+Shift+4
Developer Command Prompt for VS 2019 Ctrl+Shift+5
Developer PowerShell for VS 2019 Ctrl+Shift+6
Settings Ctrl+,
Command Palette Ctrl+Shift+P
About

```
Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\prasert>psql -U postgres -h 192.168.211.150 -d covid19
Password for user postgres:
psql (12.2, server 12.4 (Debian 12.4-1.pgdg100+1))
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.
```

set PGCLIENTENCODING=utf-8

chcp 874

chcp 65001



ชื่อฐานข้อมูลใน PostgreSQL เป็น case sensitive หรือไม่

ใช่ค่ะ **ชื่อฐานข้อมูลใน PostgreSQL เป็น case sensitive** หมายความว่าชื่อฐานข้อมูลที่มีตัวอักษรตัวพิมพ์ใหญ่และตัวพิมพ์เล็กจะถือเป็นชื่อที่แตกต่างกัน

ตัวอย่างเช่น:

- `mydb` กับ `MyDB` จะเป็นฐานข้อมูลสองฐานที่แยกจากกัน
- `test_database` กับ `TEST_DATABASE` จะเป็นฐานข้อมูลสองฐานที่แยกจากกัน

ข้อควรระวัง

- ตรวจสอบให้แน่ใจว่าได้ใช้ชื่อฐานข้อมูลที่ถูกต้องเมื่อเชื่อมต่อกับฐานข้อมูล
- เมื่อใช้คำสั่ง SQL กับฐานข้อมูล ตรวจสอบให้แน่ใจว่าได้ใช้ชื่อฐานข้อมูลที่ถูกต้อง
- การตั้งชื่อฐานข้อมูลควรใช้อักษรตัวพิมพ์เล็กทั้งหมดเพื่อหลีกเลี่ยงปัญหาเรื่อง case sensitivity

เพิ่มเติม

- PostgreSQL ยังมีความละเอียดอ่อนต่อ **collation** ของตัวอักษรด้วย หมายความว่าตัวอักษรที่มีสำเนียง (เช่น á, é, í, ó, ú) จะถือเป็นตัวอักษรที่แตกต่างกัน

ชื่อฐานข้อมูลใน PostgreSQL เป็น case sensitive หรือไม่

SQL

```sql
-- สร้างฐานข้อมูลชื่อ "mydb"
CREATE DATABASE mydb;

-- เชื่อมต่อกับฐานข้อมูล "mydb"
\c mydb

-- ตรวจสอบว่าฐานข้อมูล "MyDB" ไม่ได้ถูกสร้าง
SELECT datname FROM pg_catalog.pg_database WHERE datname = 'MyDB';

-- ผลลัพธ์:

-- (ไม่มี)
```

Postgres Schema

Postgres Data Types

Postgres Data Types

- Numeric Types
  - SMALLINT - 2 bytes
  - INTEGER - 4 bytes
  - BIGNIT - 8 bytes
  - DECIMAL or NUMERIC
  - REAL
  - DOUBLE PRECISION
  - SERIAL
  - BIGSERIAL

Postgres Data Types

- Character Types
  - VARCHAR(n)
  - CHAR(n)
  - TEXT
- Date and Time Types
  - TIMESTAMP
  - DATE
  - TIME
  - INTERVAL

Postgres Data Types

- Binary Types
  - Packed binary data
- Boolean Types
  - One byte of storage
- Enum Types

ชนิดข้อมูล	ช่วงข้อมูล	ตัวอย่าง	รายละเอียด
Boolean	`TRUE` หรือ `FALSE`	`TRUE`, `FALSE`	เก็บค่าตรรกะ
Character	ข้อความ	`'Hello, world!'`, `'สวัสดี'`	เก็บข้อความที่มีความยาวไม่เกิน 255 ตัวอักษร
Character varying	ข้อความที่มีความยาวแปรผัน	`'This is a long sentence.'`, `'ประโยคนี้ยาวมาก'`	เก็บข้อความที่มีความยาวสูงสุด 1 GB
Numeric	ตัวเลขทศนิยม	`123.456`, `-789.012`	เก็บตัวเลขทศนิยมที่มีความแม่นยำสูงสุด 38 หลัก
Integer	จำนวนเต็ม	`123`, `-456`	เก็บจำนวนเต็มที่มีค่าระหว่าง -2,147,483,648 ถึง 2,147,483,647
Bigint	จำนวนเต็มขนาดใหญ่	`1234567890123456789`, `-9876543210987654321`	เก็บจำนวนเต็มที่มีค่าระหว่าง -9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807
Date	วันที่	`'2023-12-01'`, `'2566-11-16'`	เก็บวันที่ในรูปแบบ YYYY-MM-DD
Time	เวลา	`'12:34:56'`, `'18:59:00'`	เก็บเวลาในรูปแบบ HH:MM:SS
Timestamp	วันที่และเวลา	`'2023-12-01 12:34:56'`, `'2566-11-16 18:59:00'`	เก็บวันที่และเวลาในรูปแบบ YYYY-MM-DD HH:MM:SS
Interval	ช่วงเวลา	`'1 day'`, `'1 hour 30 minutes'`	เก็บช่วงเวลาในรูปแบบ DAY, HOUR, MINUTE, SECOND

SQL

```sql
-- สร้างตาราง
CREATE TABLE my_table (
  id INT,
  name VARCHAR(255),
  age INT,
  created_at TIMESTAMP
);

-- แทรกข้อมูล
INSERT INTO my_table (id, name, age, created_at) VALUES
  (1, 'John Doe', 30, '2023-12-01 12:34:56'),
  (2, 'Jane Doe', 25, '2023-12-02 13:56:00');

-- เลือกข้อมูล
SELECT * FROM my_table;

-- ผลลัพธ์:

-- id | name       | age | created_at
-- -- | --         | -- | --
-- 1  | John Doe   | 30 | 2023-12-01 12:34:56
-- 2  | Jane Doe   | 25 | 2023-12-02 13:56:00
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

Serial

ใน pgAdmin 4 ตัวเลือก smallserial, serial และ bigserial เป็นตัวเลือกสำหรับชนิดข้อมูลของ sequences ซึ่งใช้เก็บค่า ID ของตารางโดยอัตโนมัติ แต่ละชนิดมีความแตกต่างกัน ดังนี้:

1. smallserial:

- เก็บค่า ID ได้ตั้งแต่ 1 ถึง 32,767 (2^15-1)
- เหมาะสำหรับตารางที่มีขนาดเล็ก
- ใช้พื้นที่เก็บข้อมูลน้อย

2. serial:

- เก็บค่า ID ได้ตั้งแต่ 1 ถึง 2,147,483,647 (2^31-1)
- เหมาะสำหรับตารางที่มีขนาดปานกลาง
- ใช้พื้นที่เก็บข้อมูลปานกลาง

3. bigserial:

- เก็บค่า ID ได้ตั้งแต่ 1 ถึง 9,223,372,036,854,775,807 (2^63-1)
- เหมาะสำหรับตารางที่มีขนาดใหญ่
- ใช้พื้นที่เก็บข้อมูลมาก

สรุป:

- เลือก smallserial สำหรับตารางที่มีขนาดเล็ก
- เลือก serial สำหรับตารางที่มีขนาดปานกลาง
- เลือก bigserial สำหรับตารางที่มีขนาดใหญ่

ตัวอย่าง:

- ตารางเก็บข้อมูลลูกค้า: เลือก serial
- ตารางเก็บข้อมูลสินค้า: เลือก bigserial
- ตารางเก็บข้อมูลประวัติการสั่งซื้อ: เลือก smallserial

หมายเหตุ:

- การเลือกชนิดข้อมูลของ sequences ขึ้นอยู่กับขนาดของตารางและจำนวนข้อมูลที่จะเก็บ
- สามารถเปลี่ยนชนิดข้อมูลของ sequences ได้ภายหลัง
- sequences จะสร้างค่า ID ใหม่โดยอัตโนมัติเมื่อมีการเพิ่มข้อมูลใหม่ในตาราง

คำสั่งสร้างตารางผ่าน SQL โดยใช้ชนิดข้อมูล smallserial, serial และ bigserial นั้น เขียนได้ดังนี้:

1. smallserial:

```sql
CREATE TABLE my_table (
  id smallserial NOT NULL PRIMARY KEY,
  name VARCHAR(255)
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

2. serial:

```sql
CREATE TABLE my_table (
  id serial NOT NULL PRIMARY KEY,
  name VARCHAR(255)
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

3. bigserial:

```sql
CREATE TABLE my_table (
  id bigserial NOT NULL PRIMARY KEY,
  name VARCHAR(255)
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

คำอธิบาย:

- `NOT NULL` : ระบุว่า field `id` ไม่สามารถเก็บค่า null ได้
- `PRIMARY KEY` : ระบุว่า field `id` เป็น primary key ของตาราง

Serial

ตัวอย่างเพิ่มเติม:

SQL

```sql
-- สร้างตารางเก็บข้อมูลลูกค้า

CREATE TABLE customers (
  id serial NOT NULL PRIMARY KEY,
  name VARCHAR(255),
  email VARCHAR(255),
  phone_number VARCHAR(20)
);

-- สร้างตารางเก็บข้อมูลสินค้า

CREATE TABLE products (
  id bigserial NOT NULL PRIMARY KEY,
  name VARCHAR(255),
  description TEXT,
  price DECIMAL(10,2)
);

-- สร้างตารางเก็บข้อมูลประวัติการสั่งซื้อ

CREATE TABLE orders (
  id smallserial NOT NULL PRIMARY KEY,
  customer_id INT NOT NULL REFERENCES customers(id),
  product_id INT NOT NULL REFERENCES products(id),
  quantity INT,
  order_date TIMESTAMP
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

UUID

UUID คืออะไร?

UUID ย่อมาจาก Universally Unique Identifier เป็นค่า 128 บิตที่ใช้ระบุเอกลักษณ์เฉพาะของวัตถุ UUIDs มักใช้ในระบบฐานข้อมูลเพื่อระบุแถวข้อมูล UUIDs มีคุณสมบัติพิเศษคือ มีโอกาสน้อยมากที่จะซ้ำกัน

UUID ทำงานอย่างไร?

UUIDs ถูกสร้างขึ้นโดยอัลกอริธึมที่ซับซ้อน อัลกอริธึมนี้ใช้ข้อมูลต่างๆ เช่น ที่อยู่ MAC ของคอมพิวเตอร์ เวลาปัจจุบัน และค่าสุ่ม เพื่อสร้าง UUID ที่ไม่ซ้ำกัน

UUID ใช้กับอะไร?

UUIDs สามารถใช้กับงานต่างๆ เช่น:

- ระบุแถวข้อมูลในฐานข้อมูล
- ติดตามเอกสารในระบบจัดการเอกสาร
- ระบุอุปกรณ์ในเครือข่าย

UUID

เขียนคำสั่งแบบไหนตอนสร้างตาราง?

ในการสร้างตารางที่มีคอลัมน์ UUID ใน PostgreSQL คุณสามารถใช้คำสั่ง SQL ดังนี้:

```sql
CREATE TABLE my_table (
  id UUID NOT NULL PRIMARY KEY,
  name VARCHAR(255)
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

ในตัวอย่างนี้:

- `id` เป็นชื่อคอลัมน์
- `UUID` เป็นชนิดข้อมูลของคอลัมน์
- `NOT NULL` ระบุว่าคอลัมน์นี้ไม่สามารถเก็บค่า null ได้
- `PRIMARY KEY` ระบุว่าคอลัมน์นี้เป็น primary key ของตาราง

ตัวอย่างเพิ่มเติม

```sql
-- สร้างตารางเก็บข้อมูลลูกค้า

CREATE TABLE customers (
  id UUID NOT NULL PRIMARY KEY,
  name VARCHAR(255),
  email VARCHAR(255),
  phone_number VARCHAR(20)
);

-- สร้างตารางเก็บข้อมูลสินค้า

CREATE TABLE products (
  id UUID NOT NULL PRIMARY KEY,
  name VARCHAR(255),
  description TEXT,
  price DECIMAL(10,2)
);

-- สร้างตารางเก็บข้อมูลประวัติการสั่งซื้อ

CREATE TABLE orders (
  id UUID NOT NULL PRIMARY KEY,
  customer_id UUID NOT NULL REFERENCES customers(id),
  product_id UUID NOT NULL REFERENCES products(id),
  quantity INT,
  order_date TIMESTAMP
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

หมายเหตุ

- UUIDs มีความยาว 128 บิต ซึ่งยาวกว่าชนิดข้อมูล integer และ bigint
- UUIDs สามารถใช้แทน primary key ของตารางได้
- UUIDs สามารถใช้เพื่อสร้างความสัมพันธ์ระหว่างตารางได้

UUID

ข้อเสียของการใช้ UUID

การใช้ UUIDs มีข้อเสียดังนี้:

- **ใช้พื้นที่เก็บข้อมูลมากกว่า**: UUIDs มีความยาว 128 บิต ซึ่งยาวกว่าชนิดข้อมูล integer (32 บิต) และ bigint (64 บิต)
- **ประสิทธิภาพ**: การใช้ UUIDs ในคิวรี่อาจทำให้ประสิทธิภาพช้าลง
- **ความซับซ้อน**: การจัดการ UUIDs อาจมีความซับซ้อนมากกว่าการใช้ชนิดข้อมูลอื่น

กรณีที่ควรใช้ UUID

ควรใช้ UUIDs ในกรณีดังต่อไปนี้:

- **ต้องการความเป็นเอกลักษณ์ที่แท้จริง**: UUIDs มีโอกาสน้อยมากที่จะซ้ำกัน
- **ต้องการกระจายข้อมูล**: UUIDs สามารถกระจายข้อมูลไปยังเซิร์ฟเวอร์ต่างๆ ในคลัสเตอร์ได้อย่างสม่ำเสมอ
- **ต้องการความปลอดภัย**: UUIDs สามารถใช้สร้างรหัสผ่านและโทเค็นความปลอดภัย

กรณีที่ไม่ควรใช้ UUID

ไม่ควรใช้ UUIDs ในกรณีดังต่อไปนี้:

- **ต้องการประสิทธิภาพ**: การใช้ UUIDs ในคิวรี่อาจทำให้ประสิทธิภาพช้าลง
- **ต้องการความเรียบง่าย**: การจัดการ UUIDs อาจมีความซับซ้อนมากกว่าการใช้ชนิดข้อมูลอื่น
- **ไม่ต้องการความเป็นเอกลักษณ์ที่แท้จริง**:

สรุป

การใช้ UUIDs มีทั้งข้อดีและข้อเสีย ควรพิจารณาข้อดีและข้อเสียก่อนตัดสินใจใช้ UUIDs ในระบบของคุณ

UUID

ใน pgAdmin 4, UUID และ UUID[] เป็นชนิดข้อมูลสำหรับเก็บค่า UUID

UUID:

- เก็บค่า UUID เดียว
- เหมาะสำหรับใช้เป็น primary key ของตาราง
- ตัวอย่าง:

SQL

```sql
CREATE TABLE my_table (
  id UUID NOT NULL PRIMARY KEY,
  name VARCHAR(255)
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

UUID[]:

- เก็บค่า UUID หลายค่า
- เหมาะสำหรับใช้เก็บรายการ UUID
- ตัวอย่าง:

SQL

```sql
CREATE TABLE my_table (
  id INT NOT NULL PRIMARY KEY,
  uuids UUID[]
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

สรุป:

- UUID เก็บค่า UUID เดียว
- UUID[] เก็บค่า UUID หลายค่า
- เลือกใช้ชนิดข้อมูลให้เหมาะสมกับการใช้งาน

ตัวอย่างเพิ่มเติม:

- ตารางเก็บข้อมูลลูกค้า: ใช้ UUID เก็บค่า ID ของลูกค้า
- ตารางเก็บข้อมูลสินค้า: ใช้ UUID เก็บค่า ID ของสินค้า
- ตารางเก็บข้อมูลสิทธิ์การเข้าถึง: ใช้ UUID[] เก็บค่า UUID ของสิทธิ์การเข้าถึง

หมายเหตุ:

- UUIDs มีความยาว 128 บิต
- UUIDs สามารถใช้แทน primary key ของตารางได้
- UUIDs สามารถใช้เพื่อสร้างความสัมพันธ์ระหว่างตารางได้

Postgres Role

การ Import ข้อมูลผ่าน Command Line

ฐานข้อมูลตัวอย่าง Airlines

การ Import ข้อมูลผ่าน Command Line

```
psql -U postgres -f C:\Users\samit\Downloads\airlines.sql
```

```
C:\Users\samit>psql -U postgres -f C:\Users\samit\Downloads\airlines.sql
Password for user postgres:
SET
SET
SET
SET
SET
SET
SET
SET
psql:C:/Users/samit/Downloads/airlines.sql:17: ERROR:  database "demo" does not exist
CREATE DATABASE
You are now connected to database "demo" as user "postgres".
SET
SET
SET
SET
SET
SET
SET
SET
CREATE SCHEMA
COMMENT
CREATE EXTENSION
COMMENT
SET
CREATE FUNCTION
CREATE FUNCTION
COMMENT
SET
SET
CREATE TABLE
```

การ Import ข้อมูลผ่าน Command Line

```
C:\Users\samit>psql -U postgres
Password for user postgres:
psql (15.4)
Type "help" for help.

postgres=# \l
                                                    List of databases
      Name      |  Owner   | Encoding |          Collate           |           Ctype            | ICU Locale | Locale Provider |   Access pri
vileges
----------------+----------+----------+----------------------------+----------------------------+------------+-----------------+-------------
----------
 chinook        | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 demo           | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 postgres       | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 sampleairlines | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 sampledb       | postgres | UTF8     | Thai_Thailand.874          | Thai_Thailand.874          |            | libc            |
 sampleprismadb | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 template0      | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            | =c/postgres
         +
                |          |          |                            |                            |            |                 | postgres=CTc
/postgres
 template1      | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            | =c/postgres
         +
                |          |          |                            |                            |            |                 | postgres=CTc
/postgres
(8 rows)
```

เปลี่ยนชื่อฐานข้อมูล

ALTER DATABASE demo RENAME TO airlines;

```
postgres=# ALTER DATABASE demo RENAME TO airlines;
ALTER DATABASE
postgres=# \l
                                                    List of databases
      Name      |  Owner   | Encoding |          Collate           |           Ctype            | ICU Locale | Locale Provider |   Access pri
vileges
----------------+----------+----------+----------------------------+----------------------------+------------+-----------------+-------------
----------
 airlines       | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 chinook        | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 postgres       | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 sampleairlines | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 sampledb       | postgres | UTF8     | Thai_Thailand.874          | Thai_Thailand.874          |            | libc            |
 sampleprismadb | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            |
 template0      | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            | =c/postgres
        +
                |          |          |                            |                            |            |                 | postgres=CTc
/postgres
 template1      | postgres | UTF8     | English_United States.874  | English_United States.874  |            | libc            | =c/postgres
        +
                |          |          |                            |                            |            |                 | postgres=CTc
/postgres
(8 rows)
```

สร้างฐานข้อมูล dvdrental ก่อน

Command Prompt - psql -U p

```
C:\Users\samit>psql -U postgres
Password for user postgres:
psql (13.4)
Type "help" for help.

postgres=# create database dvdrental;
CREATE DATABASE
postgres=# \l
                                   List of databases
      Name        |  Owner   | Encoding |      Collate      |       Ctype       |   Access privileges
------------------+----------+----------+-------------------+-------------------+-----------------------
 djangoinventory  | postgres | UTF8     | Thai_Thailand.874 | Thai_Thailand.874 |
 dotnetinvdb      | postgres | UTF8     | Thai_Thailand.874 | Thai_Thailand.874 |
 dvdrental        | postgres | UTF8     | Thai_Thailand.874 | Thai_Thailand.874 |
 postgres         | postgres | UTF8     | Thai_Thailand.874 | Thai_Thailand.874 |
 saturn           | postgres | UTF8     | Thai_Thailand.874 | Thai_Thailand.874 |
 template0        | postgres | UTF8     | Thai_Thailand.874 | Thai_Thailand.874 | =c/postgres          +
                  |          |          |                   |                   | postgres=CTc/postgres
 template1        | postgres | UTF8     | Thai_Thailand.874 | Thai_Thailand.874 | =c/postgres          +
                  |          |          |                   |                   | postgres=CTc/postgres
 testlab          | postgres | UTF8     | Thai_Thailand.874 | Thai_Thailand.874 |
(8 rows)
```

ใช้คำสั่ง pg_restore

pg_restore -U postgres -d dvdrental C:\Users\samit\Downloads\sampledb\dvdrental.tar

Command Prompt

```
C:\Users\samit>pg_restore -U postgres -d dvdrental C:\Users\samit\Downloads\sampledb\dvdrental.tar
Password:

C:\Users\samit>
```

Command Prompt - psql -U p

```
postgres=# \c dvdrental
You are now connected to database "dvdrental" as user "postgres".
dvdrental=# \dt
             List of relations
 Schema |     Name      | Type  |  Owner
--------+---------------+-------+----------
 public | actor         | table | postgres
 public | address       | table | postgres
 public | category      | table | postgres
 public | city          | table | postgres
 public | country       | table | postgres
 public | customer      | table | postgres
 public | film          | table | postgres
 public | film_actor    | table | postgres
 public | film_category | table | postgres
 public | inventory     | table | postgres
 public | language      | table | postgres
 public | payment       | table | postgres
 public | rental        | table | postgres
 public | staff         | table | postgres
 public | store         | table | postgres
(15 rows)


dvdrental=#
```

pgAdmin 4

pgAdmin File Object Tools Help

Browser

- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
  - public
    - Collations
    - Domains (1)
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions (8)
    - Materialized Views
    - Procedures
    - Sequences (13)
    - Tables (15)
      - actor
      - address
      - category
      - city
      - country
      - customer
      - film
      - film_actor
      - film_category
      - inventory
      - language
      - payment
      - rental
      - staff
      - store
    - Trigger Functions (1)
    - Types (1)
    - Views (7)
- Subscriptions
- postgres
- saturn
- testlab
- Login/Group Roles
- Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents public.customer/dvdrental/postgres@PostgreSQL 13

No limit

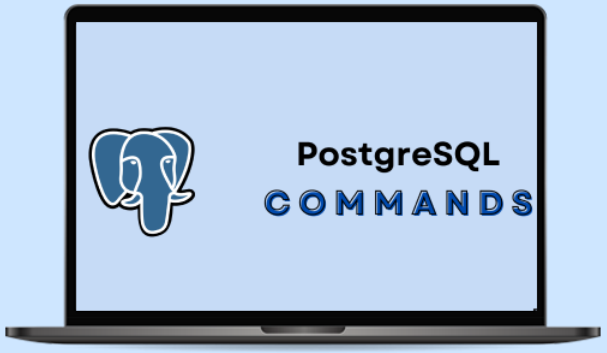
dvdrental/postgres@PostgreSQL 13

Query Editor Query History Scratch Pad

```
1 SELECT * FROM public.customer
2 ORDER BY customer_id ASC
```

Data Output Explain Messages Notifications

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	address_id smallint	activebool boolean	create_date date	last_update timestamp withou
1	1	1	Mary	Smith	mary.smith@sakilacustomer.org	5	true	2006-02-14	2013-05-26 14:
2	2	1	Patricia	Johnson	patricia.johnson@sakilacustomer.org	6	true	2006-02-14	2013-05-26 14:
3	3	1	Linda	Williams	linda.williams@sakilacustomer.org	7	true	2006-02-14	2013-05-26 14:
4	4	2	Barbara	Jones	barbara.jones@sakilacustomer.org	8	true	2006-02-14	2013-05-26 14:
5	5	1	Elizabeth	Brown	elizabeth.brown@sakilacustomer.org	9	true	2006-02-14	2013-05-26 14:
6	6	2	Jennifer	Davis	jennifer.davis@sakilacustomer.org	10	true	2006-02-14	2013-05-26 14:
7	7	1	Maria	Miller	maria.miller@sakilacustomer.org	11	true	2006-02-14	2013-05-26 14:
8	8	2	Susan	Wilson	susan.wilson@sakilacustomer.org	12	true	2006-02-14	2013-05-26 14:
9	9	2	Margaret	Moore	margaret.moore@sakilacustomer.org	13	true	2006-02-14	2013-05-26 14:
10	10	1	Dorothy	Taylor	dorothy.taylor@sakilacustomer.org	14	true	2006-02-14	2013-05-26 14:
11	11	2	Lisa	Anderson	lisa.anderson@sakilacustomer.org	15	true	2006-02-14	2013-05-26 14:
12	12	1	Nancy	Thomas	nancy.thomas@sakilacustomer.org	16	true	2006-02-14	2013-05-26 14:
13	13	2	Karen	Jackson	karen.jackson@sakilacustomer.org	17	true	2006-02-14	2013-05-26 14:
14	14	2	Betty	White	betty.white@sakilacustomer.org	18	true	2006-02-14	2013-05-26 14:
15	15	1	Helen	Harris	helen.harris@sakilacustomer.org	19	true	2006-02-14	2013-05-26 14:
16	16	2	Sandra	Martin	sandra.martin@sakilacustomer.org	20	true	2006-02-14	2013-05-26 14:
17	17	1	Donna	Thompson	donna.thompson@sakilacustomer.org	21	true	2006-02-14	2013-05-26 14:
18	18	2	Carol	Garcia	carol.garcia@sakilacustomer.org	22	true	2006-02-14	2013-05-26 14:
19	19	1	Ruth	Martinez	ruth.martinez@sakilacustomer.org	23	true	2006-02-14	2013-05-26 14:
20	20	2	Sharon	Robinson	sharon.robinson@sakilacustomer.org	24	true	2006-02-14	2013-05-26 14:
21	21	1	Michelle	Clark	michelle.clark@sakilacustomer.org	25	true	2006-02-14	2013-05-26 14:
22	22	1	Laura	Rodriguez	laura.rodriguez@sakilacustomer.org	26	true	2006-02-14	2013-05-26 14:
23	23	2	Sarah	Lewis	sarah.lewis@sakilacustomer.org	27	true	2006-02-14	2013-05-26 14:
24	24	2	Kimberly	Lee	kimberly.lee@sakilacustomer.org	28	true	2006-02-14	2013-05-26 14:
25	25	1	Deborah	Walker	deborah.walker@sakilacustomer.org	29	true	2006-02-14	2013-05-26 14:



DAY 3

Section 5: เรียนรู้วิธีเขียนคิวรี่ SQL พื้นฐาน
Section 6: เรียนรู้การปรับแต่งประสิทธิภาพ PostgreSQL

Section 5

เรียนรู้วิธีเขียนคิวรี่ SQL พื้นฐาน

- คิวรี่ SQL พื้นฐาน
- โครงสร้างของ SQL
- คิวรี่ SELECT
- คิวรี่ INSERT, UPDATE, DELETE
- ฟังก์ชัน SQL พื้นฐาน
- การ Join ตาราง
- ตัวอย่างการเขียนคิวรี่ SQL

การติดตั้งฐานข้อมูลตัวอย่าง “dvdrental” ผ่าน pg_restore

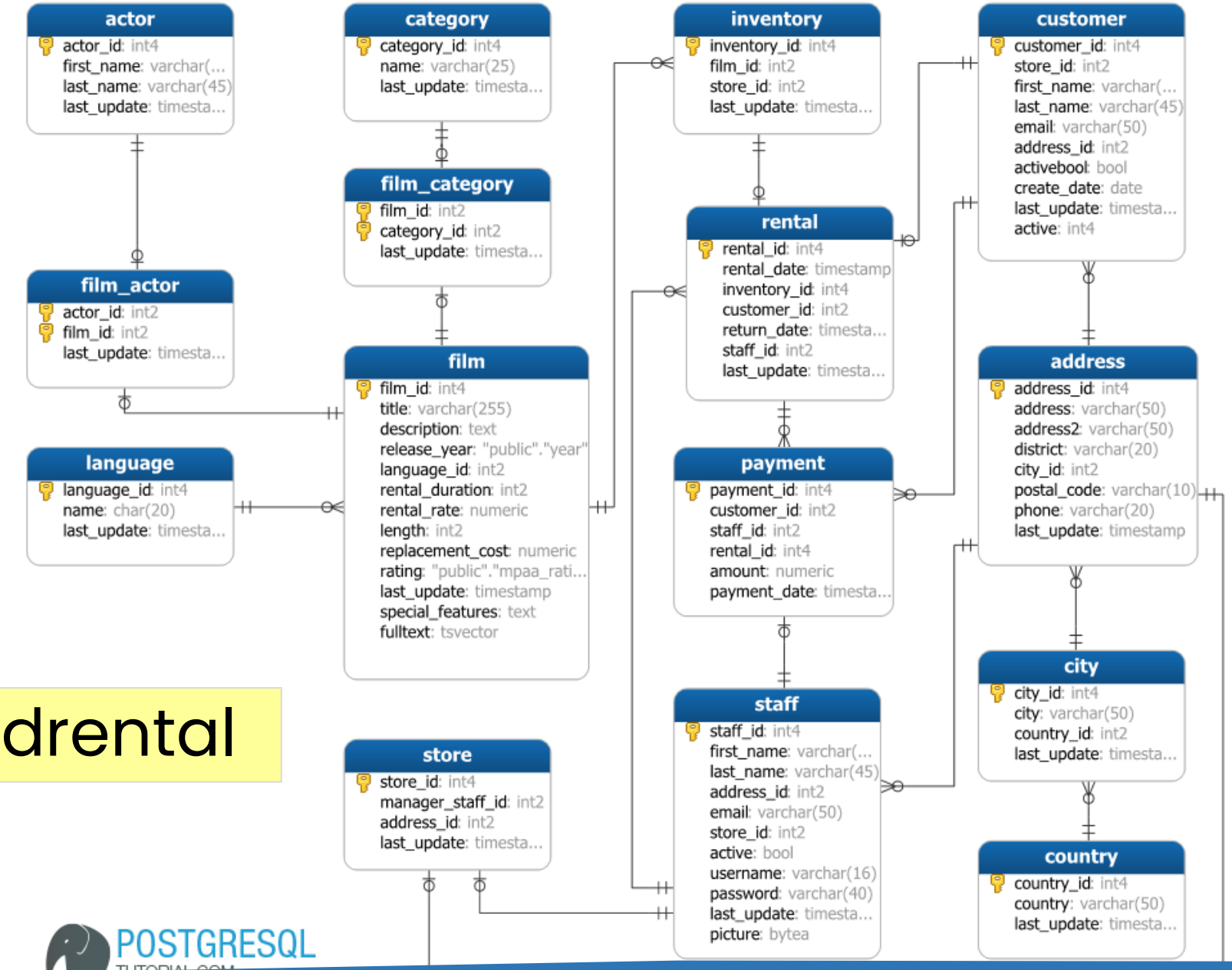
(restore database)

ฐานข้อมูลตัวอย่าง dvdrental

dvdrental | Sample Database

The DVD rental database was ported from the sakila sample database for PostgreSQL with some adjustments. The DVD rental database represents business processes of a DVD rental store. The DVD rental database has many objects including:

- 15 tables
- 1 trigger
- 7 views
- 8 functions
- 1 domain
- 13 sequences



dvdrental

POSTGRESQL

ตัวอย่างพื้นฐานการเขียน
Query PostgreSQL
กับฐานข้อมูล dvdrental

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

SELECT

1_QueryBasic.sql

```sql
-- SELECT
-------------------------------------------
-- 1. SELECT ข้อมูลจากตาราง film
-------------------------------------------
-- 1.1 เลือกคอลัมน์ทั้งหมด
SELECT * FROM film;
```

1_QueryBasic.sql

```sql
-- 1.2 เลือกคอลัมน์บางคอลัมน์ film_id, title, release_year
SELECT film_id, title, release_year FROM film;
```

1_QueryBasic.sql

```sql
-- 1.3 ใช้เงื่อนไข WHERE ในการเลือกข้อมูล
SELECT * FROM film WHERE rating = 'G';
```

1_QueryBasic.sql

```sql
-- 1.4 ใช้เงื่อนไข WHERE ในการเลือกข้อมูล และใช้ ORDER BY เรียงข้อมูล
SELECT film_id, title, rating, length
FROM film
WHERE
    rating = 'G'
ORDER BY length DESC;
```

1_QueryBasic.sql

```sql
-- 1.5 ใช้เงื่อนไข WHERE ในการเลือกข้อมูล และใช้ ORDER BY เรียงข้อมูล และใช้ LIMIT จำกัดจำนวนข้อมูล
SELECT film_id, title, rating, length
FROM film
WHERE
    rating = 'G'
ORDER BY length DESC
LIMIT 5;
```

1_QueryBasic.sql

```sql
-- 1.6 ใช้เงื่อนไข WHERE ในการเลือกข้อมูล และใช้ ORDER BY เรียงข้อมูล และใช้ LIMIT จำกัดจำนวนข้อมูล และใช้ OFFSET ข้ามข้อมูล
SELECT film_id, title, rating, length
FROM film
WHERE
    rating = 'G'
ORDER BY length DESC
LIMIT 5
OFFSET 5;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

SELECT

1_QueryBasic.sql

```sql
-- 1.7 ใช้เงื่อนไข WHERE ในการเลือกข้อมูล และใช้ ORDER BY เรียงข้อมูล และใช้ LIMIT จำกัดจำนวนข้อมูล และใช้ OFFSET ข้ามข้อมูล และใช้ FETCH FIRST และ ROWS ONLY
SELECT film_id, title, rating, length
FROM film
WHERE
    rating = 'G'
ORDER BY length DESC
OFFSET
    5
FETCH FIRST
    5 ROWS ONLY;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

SUBQuery

1_QueryBasic.sql

```sql
---------------------------------------------
-- 2. Subquery
---------------------------------------------
-- 2.1 ใช้ Subquery ในการเลือกข้อมูล ค้นหาภาพยนตร์ที่มี length มากที่สุด
SELECT film_id, title, length
FROM film
WHERE
    length = (
        SELECT MAX(length)
        FROM film
    )
LIMIT 3;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

SUBQuery

1_QueryBasic.sql

```sql
-- 2.2 ใช้ Subquery ในการเลือกข้อมูล ค้นหาภาพยนตร์ที่มี length น้อยที่สุด
SELECT film_id, title, length
FROM film
WHERE
    length = (
        SELECT MIN(length)
        FROM film
    )
LIMIT 3;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

SUBQuery

1_QueryBasic.sql

```sql
-- 2.3 ใช้ Subquery ในการเลือกข้อมูล ค้นหาภาพยนตร์ที่มี length มากกว่าเฉลี่ย
SELECT film_id, title, length
FROM film
WHERE
    length > (
        SELECT AVG(length)
        FROM film
    )
LIMIT 10;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

INSERT

1_QueryBasic.sql

```sql
---------------------------------------------
-- 3. Insert ข้อมูลเพื่อทดสอบ
---------------------------------------------
INSERT INTO
    film (
        title,
        description,
        release_year,
        language_id,
        rental_duration,
        rental_rate,
        length,
        replacement_cost,
        rating,
        last_update,
        special_features,
        fulltext
    )
VALUES (
        'Test',
        'Test',
        2022,
        1,
        3,
        4.99,
        120,
        19.99,
        'G',
        now(),
        '{"Trailers"}',
        to_tsvector('Test')
    );
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

INSERT

1_QueryBasic.sql

```sql
- 3.1 ใช้ RETURNING ในการเพิ่มข้อมูลและเลือกข้อมูลที่เพิ่มเข้าไป
INSERT INTO
    film (
        title,
        description,
        release_year,
        language_id,
        rental_duration,
        rental_rate,
        length,
        replacement_cost,
        rating,
        last_update,
        special_features,
        fulltext
    )
VALUES (
        'Test for new film',
        'Test for new film description',
        2023,
        1,
        3,
        4.99,
        115,
        29.99,
        'R',
        now(),
        '{"Trailers"}',
        to_tsvector('new')
    )
RETURNING film_id, title, release_year;

```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

UPDATE

1_QueryBasic.sql

```sql
---------------------------------------------
-- 4. Update ข้อมูล
---------------------------------------------
-- 4.1 ใช้ UPDATE ในการแก้ไขข้อมูล
UPDATE
    film
SET
    title = 'Test for update film'
WHERE
    film_id = 1003;

-- 4.2 ใช้ RETURNING ในการแก้ไขข้อมูลและเลือกข้อมูลที่แก้ไข
UPDATE
    film
SET
    title = 'Test for update film 2'
WHERE
    film_id = 1003
RETURNING film_id, title;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

DELETE

1_QueryBasic.sql

```sql
-----------------------------------------
-- 5. Delete ข้อมูล
-----------------------------------------
-- 5.1 ใช้ DELETE ในการลบข้อมูล
DELETE FROM
    film
WHERE
    film_id = 1003;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

INNER JOIN

1_QueryBasic.sql

```sql
-----------------------------------------------
-- 6. การใช้คำสั่ง JOIN
-----------------------------------------------
-- 6.1 ใช้ INNER JOIN ในการเลือกข้อมูล ค้นหารายชื่อผู้เช่าภาพยนตร์ customer joint กับ rental และ film
SELECT
    c.first_name,
    c.last_name,
    f.title,
    r.rental_date
FROM
    customer c
INNER JOIN rental r ON c.customer_id = r.customer_id
INNER JOIN inventory i ON r.inventory_id = i.inventory_id
INNER JOIN film f ON i.film_id = f.film_id
ORDER BY
    r.rental_date DESC;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

LEFT JOIN

1_QueryBasic.sql

```sql
-- 6.2 ใช้ LEFT JOIN ในการเลือกข้อมูล ค้นหารายชื่อผู้เช่าภาพยนตร์ customer joint กับ rental และ film
SELECT
    c.first_name,
    c.last_name,
    f.title,
    r.rental_date
FROM
    customer c
LEFT JOIN rental r ON c.customer_id = r.customer_id
LEFT JOIN inventory i ON r.inventory_id = i.inventory_id
LEFT JOIN film f ON i.film_id = f.film_id
ORDER BY
    r.rental_date DESC;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

RIGHT JOIN

1_QueryBasic.sql

```sql
customer joint กับ rental และ film
SELECT
    c.first_name,
    c.last_name,
    f.title,
    r.rental_date
FROM
    customer c
RIGHT JOIN rental r ON c.customer_id = r.customer_id
RIGHT JOIN inventory i ON r.inventory_id = i.inventory_id
RIGHT JOIN film f ON i.film_id = f.film_id
ORDER BY
    r.rental_date DESC;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

LEFT JOIN

1_QueryBasic.sql

```sql
-- 6.4 ค้นหาภาพยนตร์ที่ยังไม่ถูกเช่า
SELECT
    f.title
FROM
    film f
LEFT JOIN inventory i ON f.film_id = i.film_id
LEFT JOIN rental r ON i.inventory_id = r.inventory_id
WHERE
    r.rental_id IS NULL;
```

ตัวอย่างพื้นฐานการเขียน Query PostgreSQL กับฐานข้อมูล dvdrental

หัวข้อที่น่าสนใจเพิ่มเติมสำหรับการเขียน Query PostgreSQL

- การกรองข้อมูลด้วย LIKE: ค้นหาข้อมูลที่มีรูปแบบคล้ายคลึง
- การใช้ IN: ค้นหาข้อมูลที่อยู่ในรายการ
- การใช้ BETWEEN: ค้นหาข้อมูลอยู่ในช่วง
- การใช้ GROUP BY: จัดกลุ่มข้อมูลและรวมผลลัพธ์
- การใช้ DISTINCT: ค้นหาเฉพาะข้อมูลที่ไม่ซ้ำกัน
- การใช้ Aggregate Functions: คำนวณค่ารวม เช่น SUM, AVG, COUNT
- การใช้ JOIN: รวมข้อมูลจากตารางต่างๆ
- การใช้ CTE (Common Table Expressions): เขียน Query ที่ซับซ้อนให้ง่ายขึ้น

การกรองข้อมูลด้วย LIKE: ค้นหาข้อมูลที่มีรูปแบบคล้ายคลึง

1_QueryBasic.sql

```sql
-- ค้นหาภาพยนตร์ที่มีชื่อคล้ายกับ "Queen":
SELECT
    title
FROM
    film
WHERE
    title LIKE 'Queen%';
```

การใช้ IN: ค้นหาข้อมูลที่อยู่ในรายการ

1_QueryBasic.sql

```sql
-- ค้นหาภาพยนตร์ที่มี rating อยู่ใน G และ PG:
SELECT
    title,
    rating
FROM
    film
WHERE
    rating IN ('G', 'PG');
```

การใช้ BETWEEN: ค้นหาข้อมูลอยู่ในช่วง

1_QueryBasic.sql

```sql
-- ค้นหาภาพยนตร์ที่มี length อยู่ในช่วง 60 ถึง 90:
SELECT
    title,
    length
FROM
    film
WHERE
    length BETWEEN 60 AND 90;
```

การใช้ DISTINCT: ค้นหาเฉพาะข้อมูลที่ไม่ซ้ำกัน

1_QueryBasic.sql

```sql
-- การใช้ DISTINCT: ค้นหาเฉพาะข้อมูลที่ไม่ซ้ำกัน
SELECT DISTINCT rating FROM film;
```

การใช้ Aggregate Functions: คำนวณค่ารวม เช่น SUM, AVG, COUNT

1_QueryBasic.sql

```sql
-- การใช้ Aggregate Functions: คำนวณค่ารวม เช่น SUM, AVG, COUNT
SELECT
    rating,
    COUNT(*) AS total_films,
    AVG(length) AS avg_length,
    SUM(length) AS total_length
FROM
    film
GROUP BY
    rating;
```

การใช้ GROUP BY: จัดกลุ่มข้อมูลและรวมผลลัพธ์

1_QueryBasic.sql

```sql
-- จัดกลุ่มข้อมูลและรวมผลลัพธ์ จำนวนผู้เช่าแต่ละ rating เรียงลำดับจากมากไปน้อย:
SELECT rating, COUNT(*) AS total_rentals
FROM film
INNER JOIN rental ON film.film_id = rental.inventory_id
INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id
GROUP BY rating
ORDER BY total_rentals DESC;
```

การใช้ CTE (Common Table Expressions): เขียน Query ที่ซับซ้อนให้ง่ายขึ้น

1_QueryBasic.sql

```sql
-- การใช้ CTE (Common Table Expressions): เขียน Query ที่ซับซ้อนให้ง่ายขึ้น
-- ค้นหารายชื่อผู้เช่าภาพยนตร์ 5 อันดับแรก
WITH top_rentals AS (
    SELECT
        c.first_name,
        c.last_name,
        COUNT(*) AS total_rentals
    FROM
        customer c
    INNER JOIN rental r ON c.customer_id = r.customer_id
    GROUP BY
        c.first_name,
        c.last_name
    ORDER BY
        total_rentals DESC
    LIMIT 5
)
SELECT * FROM top_rentals;
-- 1.CTE top_rentals ค้นหา 5 อันดับภาพยนตร์ที่มีผู้เช่ามากที่สุด
-- 2.SELECT statement ดึงข้อมูลชื่อผู้เช่าและชื่อภาพยนตร์จากตาราง customer, rental
```

การใช้ CTE (Common Table Expressions): เขียน Query ที่ซับซ้อนให้ง่ายขึ้น

1_QueryBasic.sql

```sql
-- ค้นหาภาพยนตร์ที่มี length มากกว่าเฉลี่ย
WITH avg_length AS (
    SELECT AVG(length) AS avg_length
    FROM film
)
SELECT
    title,
    length
FROM
    film
WHERE
    length > (SELECT avg_length FROM avg_length)
ORDER BY
    length DESC;
```

การใช้ CTE (Common Table Expressions): เขียน Query ที่ซับซ้อนให้ง่ายขึ้น

1_QueryBasic.sql

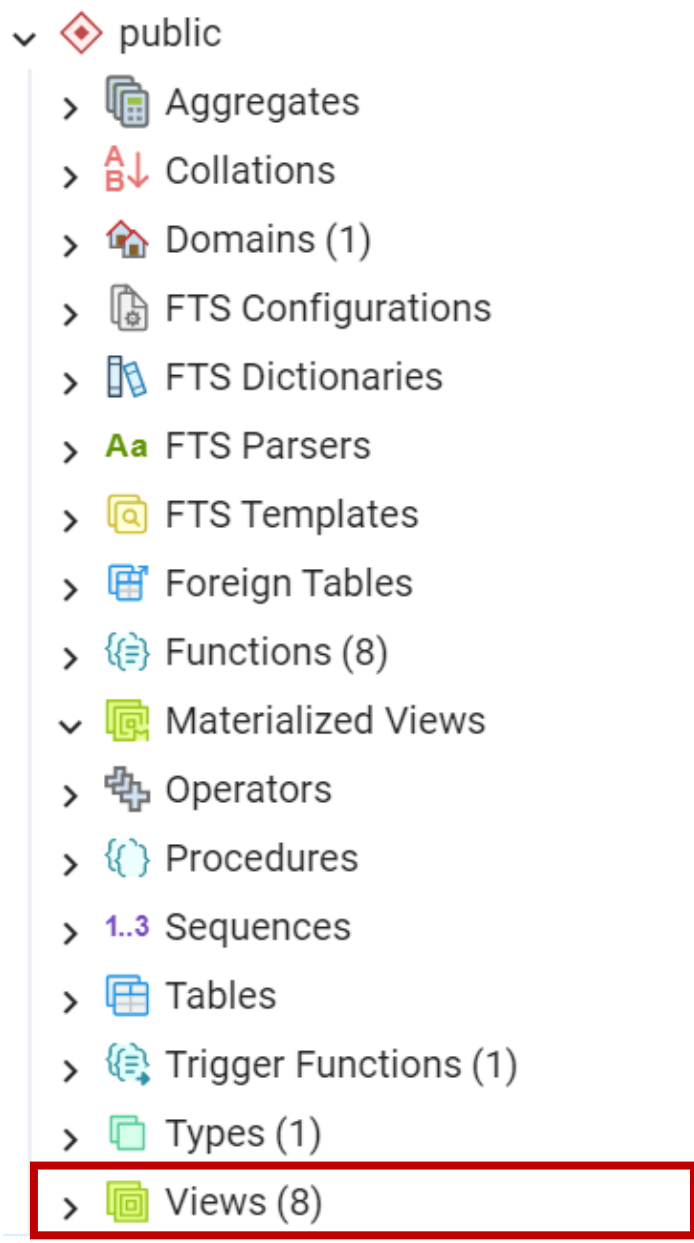
```sql
-- ค้นหารายชื่อผู้เช่าภาพยนตร์ประเภท R
WITH rental_r AS (
    SELECT
        c.first_name,
        c.last_name,
        f.title,
        r.rental_date
    FROM
        customer c
    INNER JOIN rental r ON c.customer_id = r.customer_id
    INNER JOIN inventory i ON r.inventory_id = i.inventory_id
    INNER JOIN film f ON i.film_id = f.film_id
    WHERE
        f.rating = 'R'
)
SELECT * FROM rental_r;
```

ตัวอย่างพื้นฐานการเขียน
View
กับฐานข้อมูล dvdrental

View คืออะไร ?

View คือ วัตถุเสมือนในฐานข้อมูลที่เก็บผลลัพธ์ของ SELECT statement โดยไม่ต้องสร้างตารางใหม่ ช่วยให้ผู้ใช้สามารถ

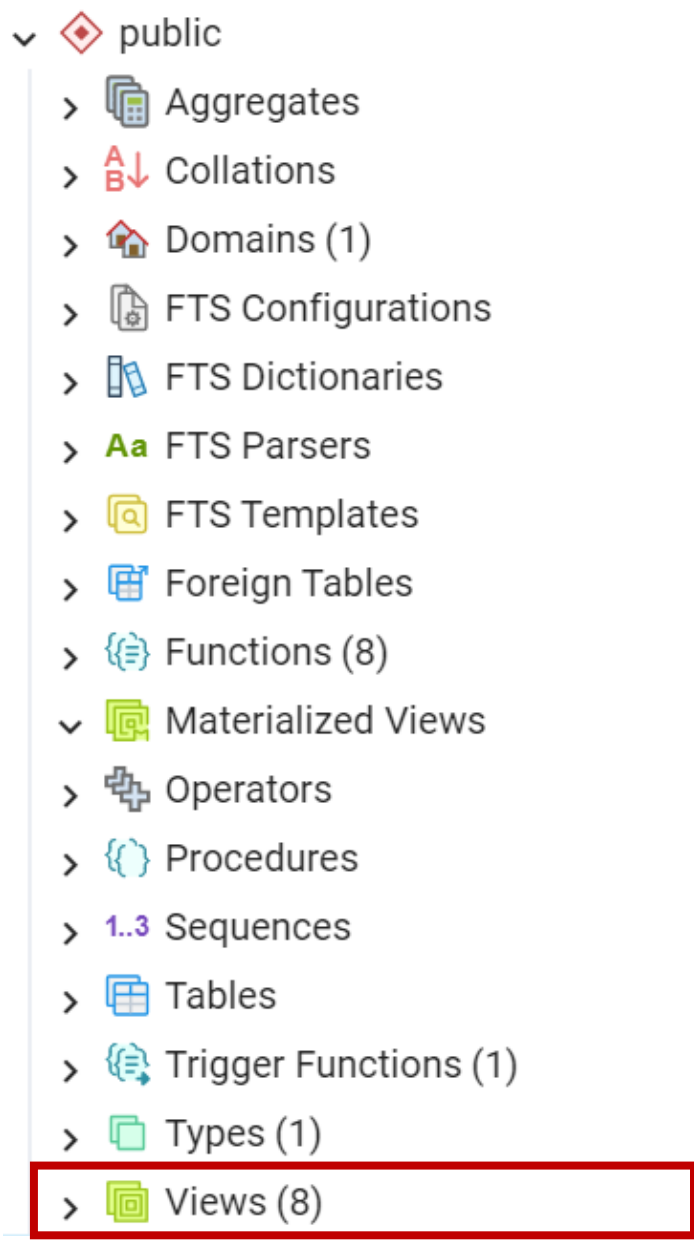
- เข้าถึงข้อมูลซับซ้อนได้ง่ายขึ้น
- กรองข้อมูลเฉพาะที่ต้องการ
- เปลี่ยนชื่อคอลัมน์
- เพิ่มคอลัมน์ใหม่
- กำหนดสิทธิ์การเข้าถึง



View สำหรับแสดงรายชื่อภาพยนตร์ที่มี rating สูง

1_QueryBasic.sql

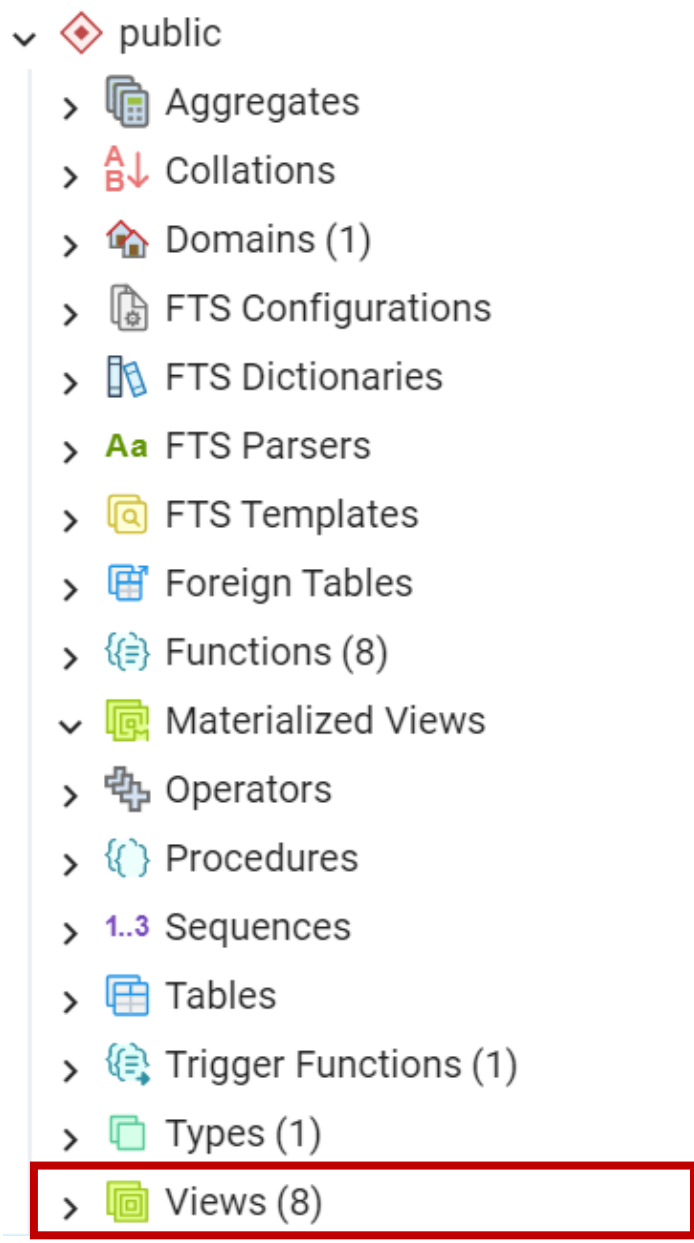
```sql
-- View สำหรับแสดงรายชื่อภาพยนตร์ที่มี rating สูง:
CREATE OR REPLACE VIEW high_rated_films AS
SELECT
    title,
    rating
FROM
    film
WHERE
    rating = 'R';

-- ดึงข้อมูลจาก View:
SELECT * FROM high_rated_films;

-- ลบ View:
DROP VIEW high_rated_films;
```



View สำหรับแสดงรายชื่อผู้เช่าภาพยนตร์ 5 อันดับแรก

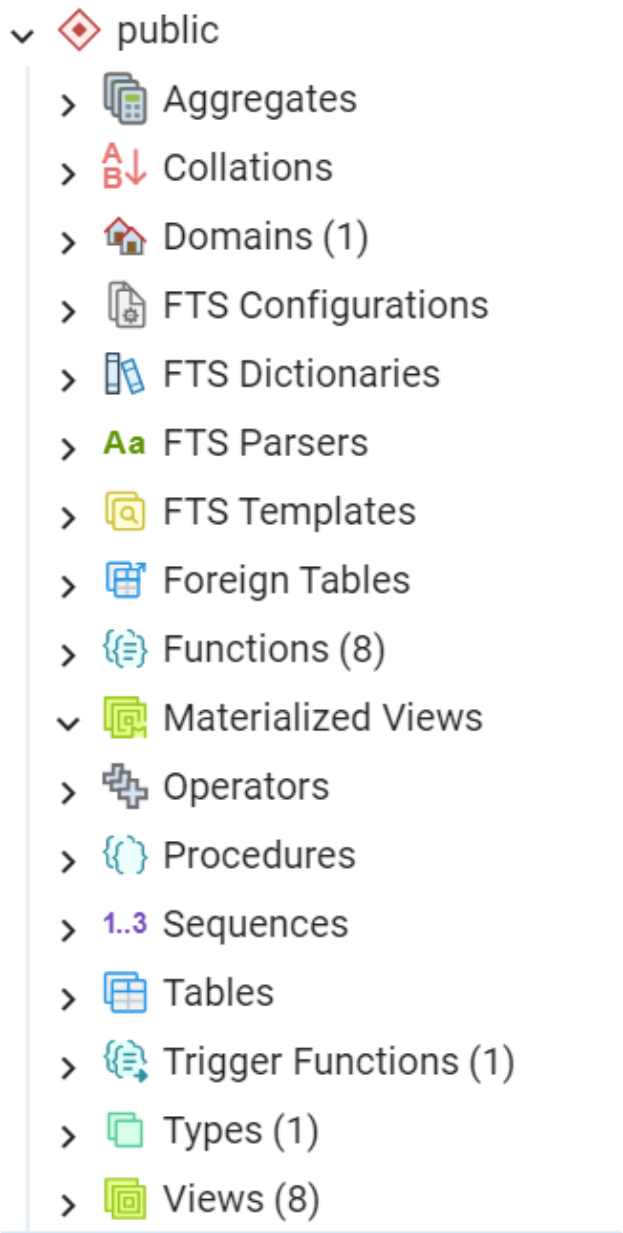
1_QueryBasic.sql

```sql
-- View สำหรับแสดงรายชื่อผู้เช่าภาพยนตร์ 5 อันดับแรก:
CREATE OR REPLACE VIEW top_rentals AS
WITH top_rentals AS (
    SELECT
        c.first_name,
        c.last_name,
        COUNT(*) AS total_rentals
    FROM
        customer c
    INNER JOIN rental r ON c.customer_id = r.customer_id
    GROUP BY
        c.first_name,
        c.last_name
    ORDER BY
        total_rentals DESC
    LIMIT 5
)
SELECT * FROM top_rentals;

-- ดึงข้อมูลจาก View:
SELECT * FROM top_rentals;

-- ดึงข้อมูลจาก View และเพิ่มเงื่อนไข:
SELECT * FROM top_rentals WHERE total_rentals >= 45;

-- ข้อควรระวัง:
-- View ไม่ได้เก็บข้อมูลจริง
-- การเปลี่ยนแปลงข้อมูลในตารางต้นทางจะส่งผลต่อ view
```



ตัวอย่างพื้นฐานการเขียน

- Function
- Procedures
- Trigger
- Domain



1. ฟังก์ชัน (Functions)

ฟังก์ชันเป็นบล็อกโค้ด SQL ที่สามารถรับอินพุต คำนวณผลลัพธ์ และส่งคืนค่า ฟังก์ชันมักใช้เพื่อ:

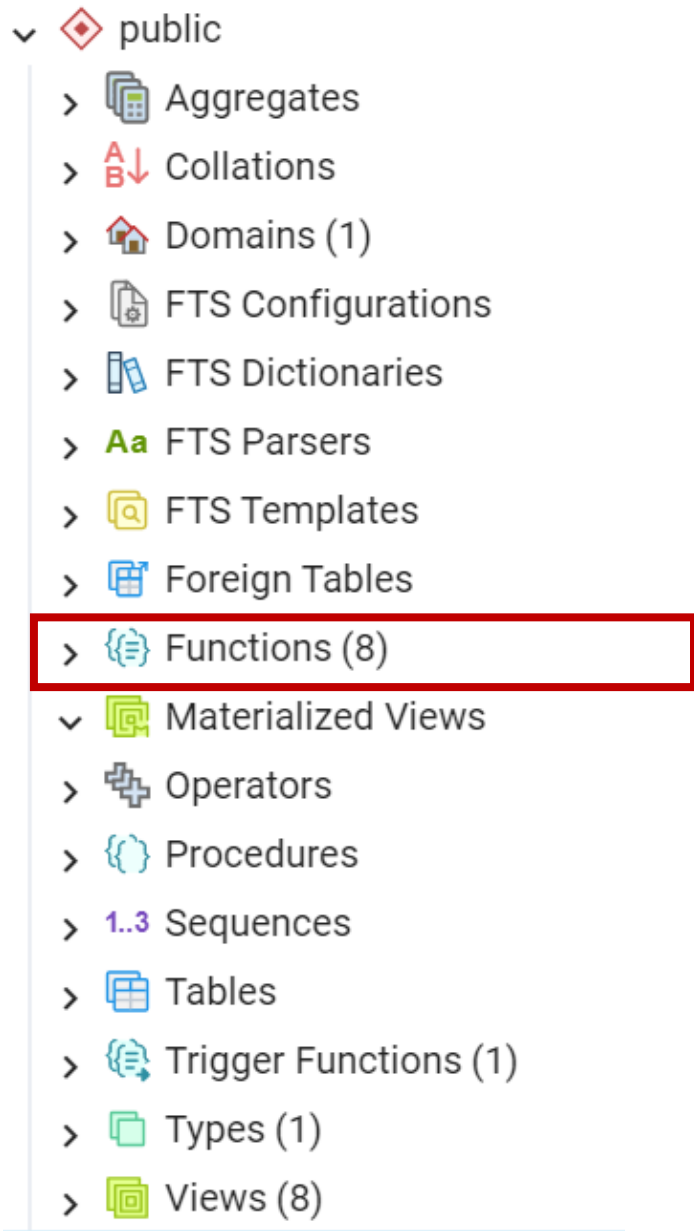
- เพิ่มขีดความสามารถในการใช้งานซ้ำของโค้ด
- แยกโค้ดออกเป็นส่วนย่อยๆ เพื่อจัดการง่ายขึ้น
- คำนวณค่าที่ซับซ้อน

ตัวอย่าง:

SQL

```sql
CREATE FUNCTION get_product_price(product_id integer) RETURNS numeric
AS $$
SELECT price
FROM product
WHERE product_id = $1;
$$ LANGUAGE sql;

SELECT get_product_price(10); -- ผลลัพธ์: 1000
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม



2. โพรซีเจอร์ (Procedures)

โพรซีเจอร์คล้ายกับฟังก์ชัน แต่สามารถ:

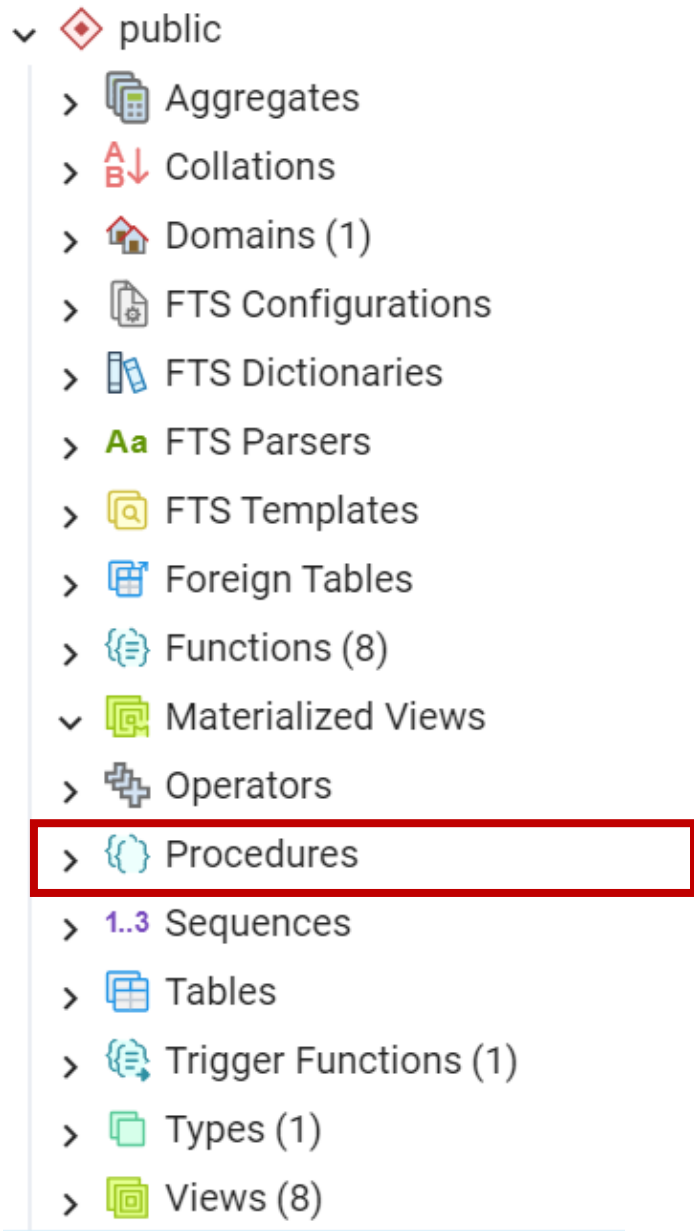
- ประกอบไปด้วยหลาย statements
- เปลี่ยนแปลงข้อมูลในฐานข้อมูล
- ส่งคืนค่าหลายค่า

ตัวอย่าง:

SQL

```sql
CREATE PROCEDURE update_product_price(product_id integer, new_price nume
AS $$
BEGIN
UPDATE product
SET price = $2
WHERE product_id = $1;
END;
$$ LANGUAGE plpgsql;

CALL update_product_price(10, 1200);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม



3. ทริกเกอร์ (Triggers)

ทริกเกอร์เป็นโค้ด SQL ที่ถูกเรียกโดยอัตโนมัติเมื่อมีการดำเนินการกับตาราง ทริกเกอร์มักใช้เพื่อ:

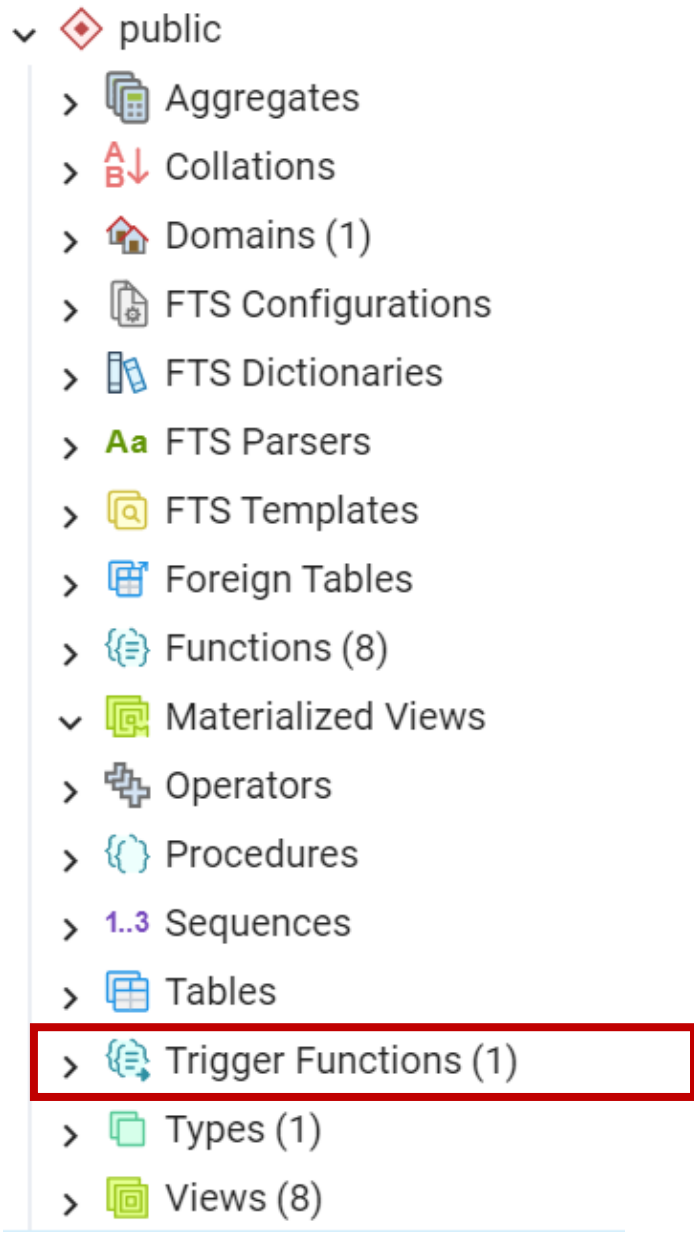
- ตรวจสอบการเปลี่ยนแปลงข้อมูล
- บังคับใช้กฎธุรกิจ
- ดำเนินการอัตโนมัติ

ตัวอย่าง:

SQL

```sql
CREATE TRIGGER log_product_update
AFTER UPDATE ON product
FOR EACH ROW
AS $$
INSERT INTO product_log (product_id, old_price, new_price)
VALUES (OLD.product_id, OLD.price, NEW.price);
$$ LANGUAGE plpgsql;

UPDATE product
SET price = 1200
WHERE product_id = 10;

-- ตรวจสอบ product_log
SELECT * FROM product_log;
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม



4. โดเมน (Domains)

โดเมนเป็นการกำหนดชนิดข้อมูลและข้อจำกัดสำหรับคอลัมน์ในตาราง โดเมนมีประโยชน์สำหรับ:

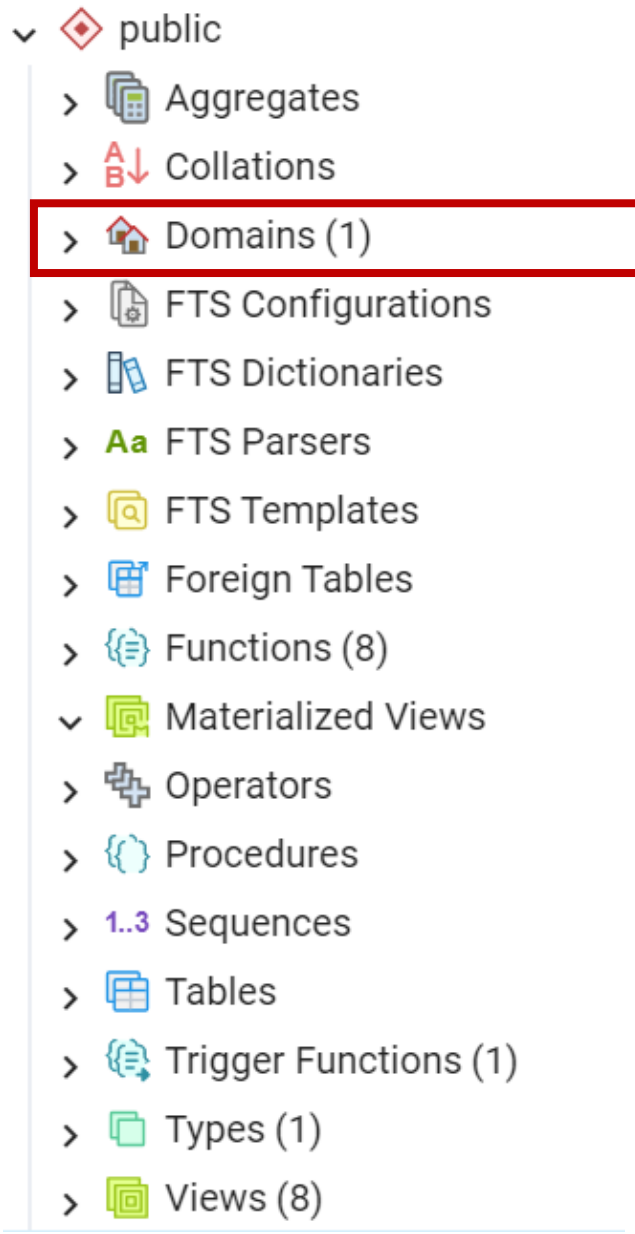
- เพิ่มความถูกต้องของข้อมูล
- เพิ่มความชัดเจนของโครงสร้างตาราง
- ป้องกันการป้อนข้อมูลที่ไม่ถูกต้อง

ตัวอย่าง:

SQL

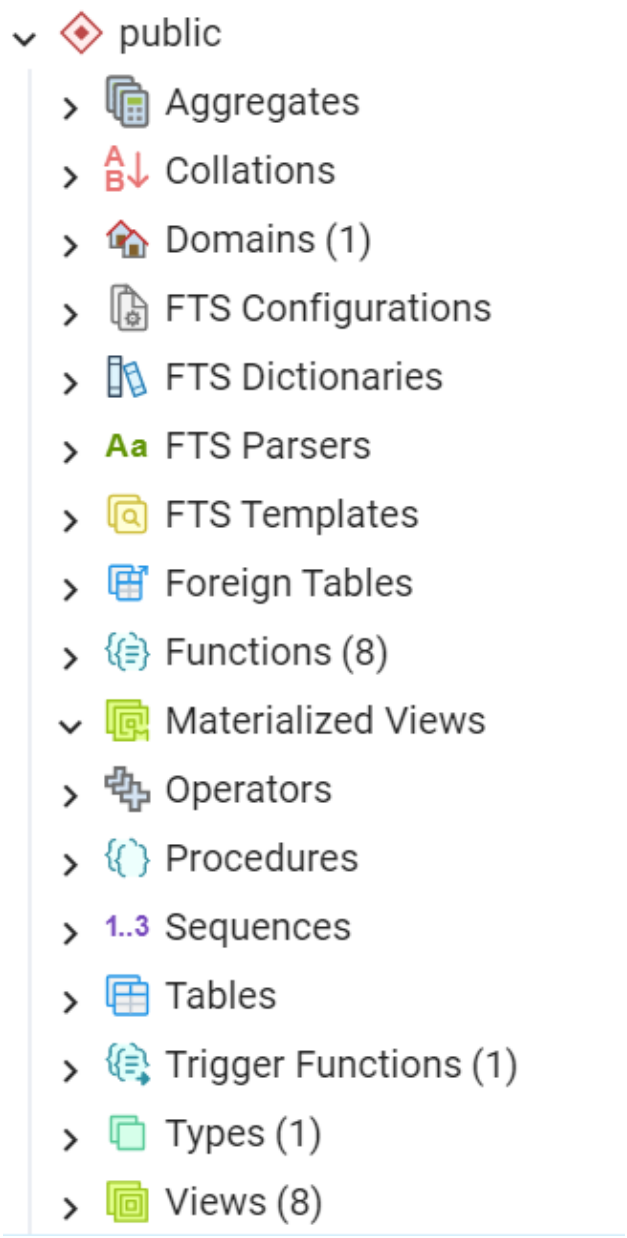
```sql
CREATE DOMAIN product_category
AS TEXT
CHECK (VALUE IN ('electronics', 'clothing', 'furniture'));

CREATE TABLE product (
  product_id serial PRIMARY KEY,
  name text NOT NULL,
  category product_category NOT NULL,
  price numeric NOT NULL
);
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม



สรุป

- ฟังก์ชัน: ใช้สำหรับคำนวณค่าและส่งคืนค่า
- โพรซีเจอร์: ใช้สำหรับดำเนินการที่ซับซ้อน
- ทริกเกอร์: ใช้สำหรับตรวจสอบและดำเนินการเมื่อมีการเปลี่ยนแปลงข้อมูล
- โดเมน: ใช้สำหรับกำหนดชนิดข้อมูลและข้อจำกัดสำหรับคอลัมน์



การเรียกใช้งาน Functions

- เรียกจาก SQL โดยตรง:

SQL

```sql
SELECT get_product_price(10);
```

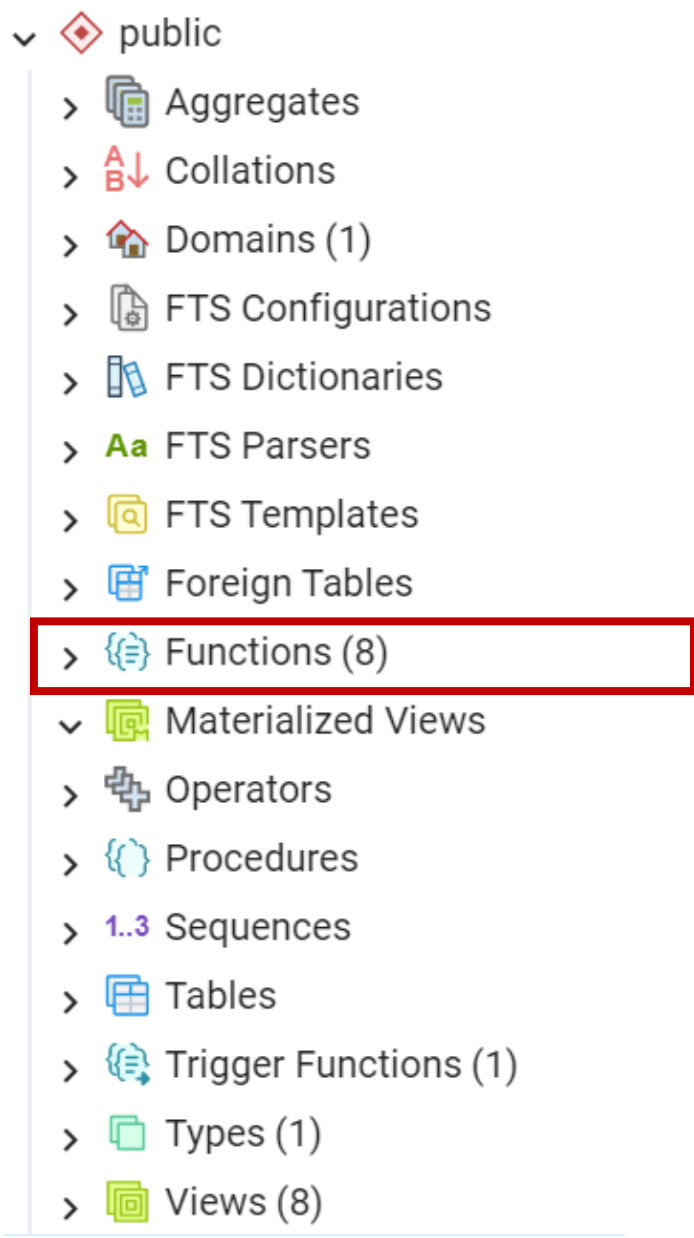
ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

- เรียกจากภาษาโปรแกรมอื่น เช่น Python:

Python

```python
import psycopg2

conn = psycopg2.connect(...)
cursor = conn.cursor()

cursor.execute("SELECT get_product_price(10)")
price = cursor.fetchone()[0]

cursor.close()
conn.close()
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม



การเรียกใช้งานโพรซีเจอร์ (Procedures)

- เรียกจาก SQL โดยตรง:

SQL

```sql
CALL update_product_price(10, 1200);
```

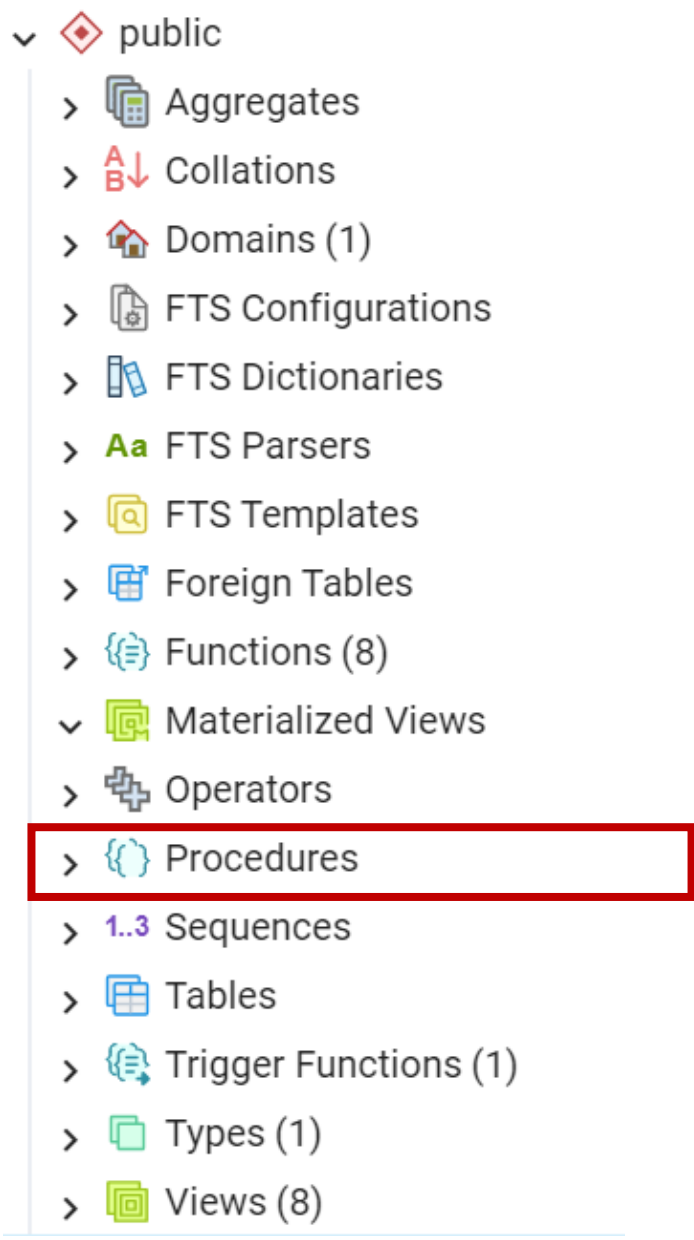
ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

- เรียกจากภาษาโปรแกรมอื่น เช่น Python:

Python

```python
import psycopg2

conn = psycopg2.connect(...)
cursor = conn.cursor()

cursor.callproc("update_product_price", (10, 1200))

cursor.close()
conn.close()
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม



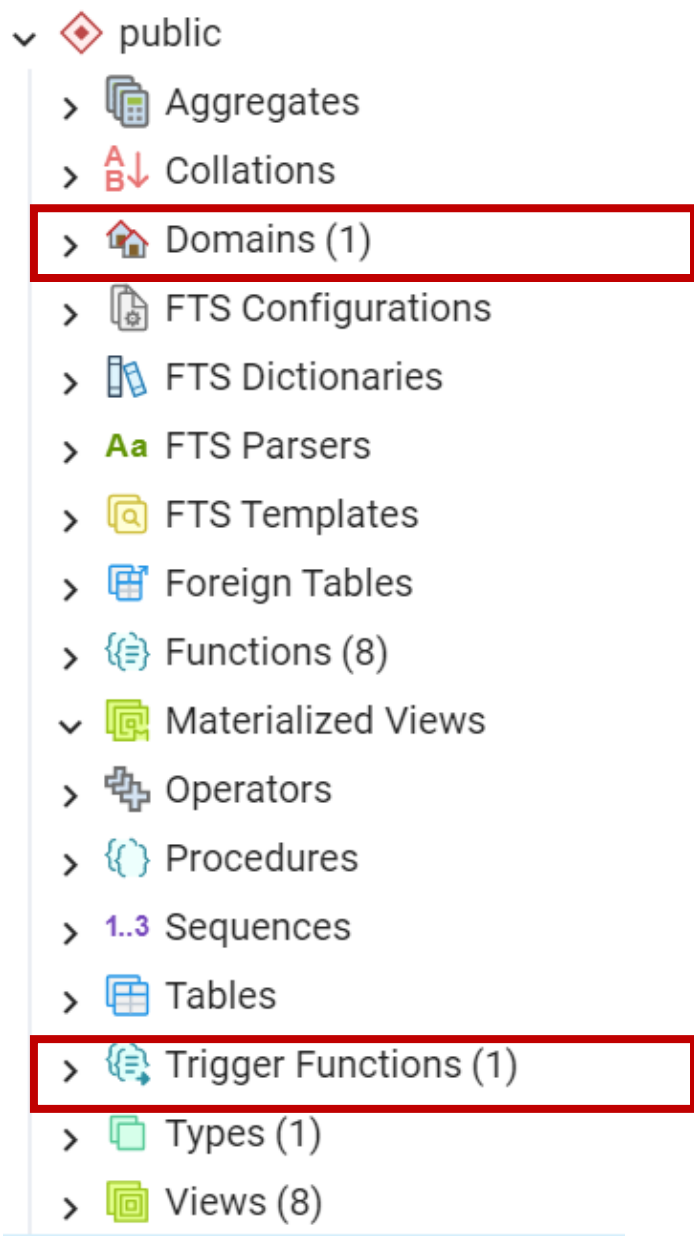
การเรียกใช้งานทริกเกอร์ (Triggers) และ โดเมน (Domains)

3. ทริกเกอร์ (Triggers)

- ทริกเกอร์จะถูกเรียกโดยอัตโนมัติเมื่อมีการดำเนินการกับตารางที่เกี่ยวข้อง

4. โดเมน (Domains)

- โดเมนจะถูกใช้โดยอัตโนมัติเมื่อมีการกำหนดคอลัมน์ในตาราง



ตัวอย่างการเขียนฟังก์ชัน (Functions)

SQL

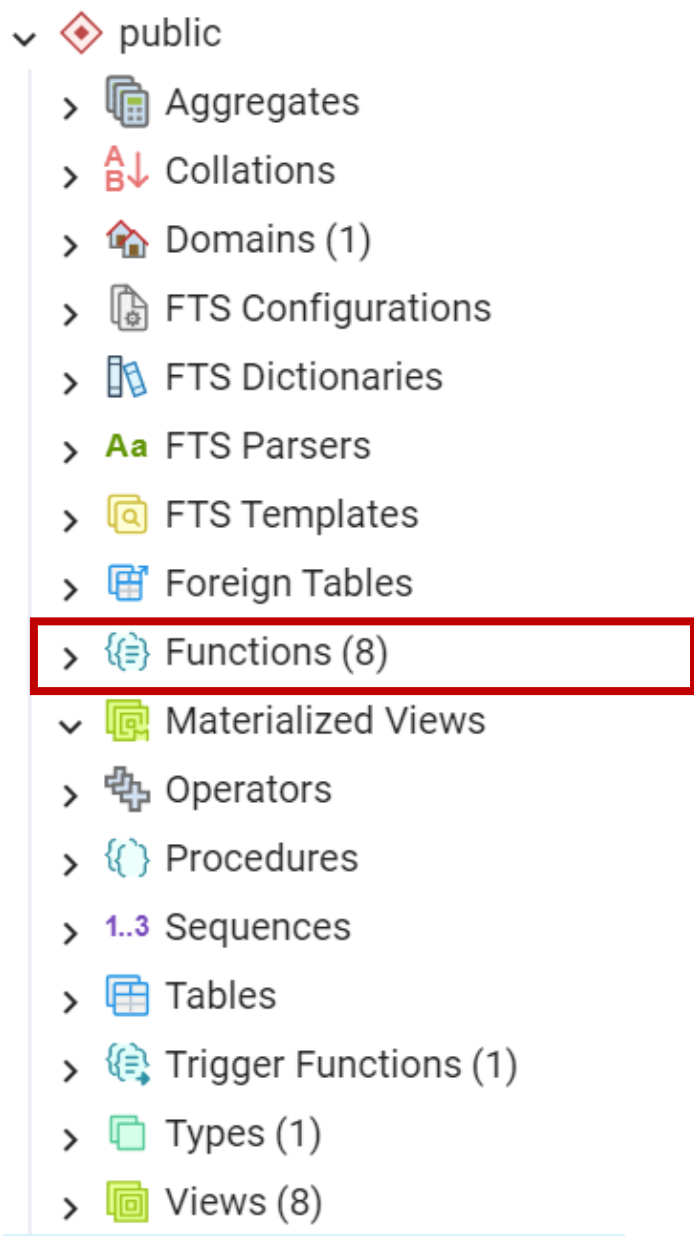
```sql
-- ตัวอย่าง 1: ฟังก์ชัน

CREATE FUNCTION get_product_price(product_id integer) RETURNS numeric
AS $$
DO
  SELECT price
  FROM product
  WHERE product_id = $1;
END;
$$ LANGUAGE sql;
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

- `DO` เป็นคำหลัก SQL มาตรฐาน
- `$$` เป็น delimiter เฉพาะของ PostgreSQL

สรุป:

- `DO` และ `$$` กำหนดขอบเขตและแยกโค้ด SQL
- ช่วยให้โค้ดอ่านง่ายขึ้น
- ป้องกันข้อผิดพลาดจาก parser



ตัวอย่างการเขียนโพรซีเจอร์ (Procedures)

SQL

```sql
-- ตัวอย่าง 2: โพรซีเจอร์

CREATE PROCEDURE update_product_price(product_id integer, new_price nume
AS $$
DO
BEGIN
  UPDATE product
  SET price = $2
  WHERE product_id = $1;
END;
$$ LANGUAGE plpgsql;
```

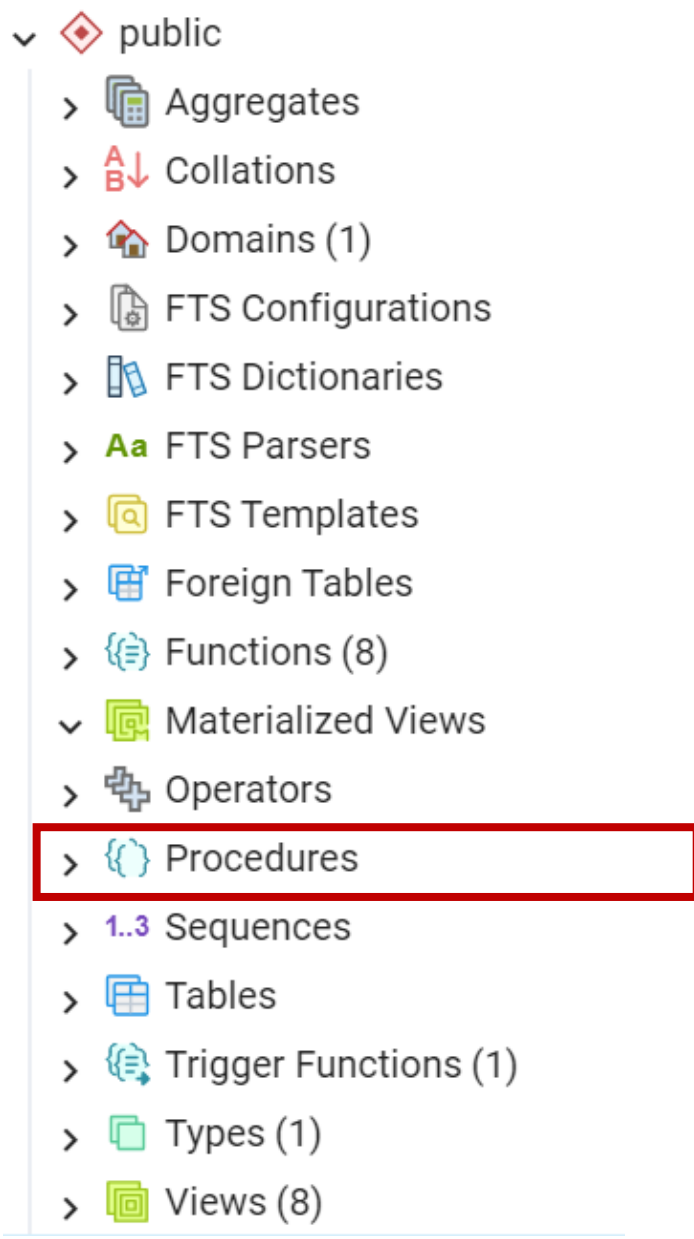
ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

หมายเหตุ:

- `DO` เป็นคำหลัก SQL มาตรฐาน
- `$$` เป็น delimiter เฉพาะของ PostgreSQL

สรุป:

- `DO` และ `$$` กำหนดขอบเขตและแยกโค้ด SQL
- ช่วยให้โค้ดอ่านง่ายขึ้น
- ป้องกันข้อผิดพลาดจาก parser



ตัวอย่างการเขียนฟังก์ชัน (Functions)

3_Function.sql

```sql
-- ฟังก์ชันรับชื่อภาพยนตร์ (title) และส่งคืนจำนวนดีวีดีที่มีอยู่ในสต็อก
DROP FUNCTION get_dvd_stock;

CREATE OR REPLACE FUNCTION public.get_dvd_stock(p_title
character varying) RETURNS integer
LANGUAGE sql
AS
$$
    SELECT COUNT(*)
    FROM inventory
    WHERE
        film_id = (
            SELECT film_id
            FROM film
            WHERE
                title = p_title
        )
        AND inventory_in_stock (inventory_id);
$$

-- เรียกใช้งานฟังก์ชัน
SELECT * FROM get_dvd_stock ('Chamber Italian');

-- ลบฟังก์ชัน
DROP FUNCTION get_dvd_stock;
```

ตัวอย่างการเขียนโพรซีเจอร์ (Procedure)

3_Function.sql

```sql
-- โพรซีเจอร์ add_rental นี้รับรหัสลูกค้า (customer_id) และวันที่เช่า (rental_date) และเพิ่มการเช่าใหม่
CREATE OR REPLACE PROCEDURE public.add_rental(p_customer_id integer, p_rental_date date)
LANGUAGE plpgsql
AS
$$
DECLARE
    v_rental_id integer;
BEGIN
    INSERT INTO rental (rental_date, inventory_id, customer_id, staff_id)
    VALUES (p_rental_date, 1525, p_customer_id, 1)
    RETURNING rental_id INTO v_rental_id;
    INSERT INTO payment (customer_id, staff_id, rental_id, amount, payment_date)
    VALUES (p_customer_id, 1, v_rental_id, 4.99, p_rental_date);
END;
$$

-- เรียกใช้งานโพรซีเจอร์ payment_date with timezone
CALL add_rental(6, '2022-10-20');

-- ลบโพรซีเจอร์
DROP PROCEDURE add_rental;
```

ตัวอย่างการเขียนทริกเกอร์ (Triggers)

3_Function.sql

```sql
-- ทริกเกอร์นี้จะอัปเดตคอลัมน์ last_update ของตาราง inventory เมื่อมีการเพิ่มการเช่าใหม่
CREATE OR REPLACE FUNCTION public.update_dvd_stock()
RETURNS trigger
LANGUAGE plpgsql
AS
$$
BEGIN
    UPDATE inventory
    SET last_update = CURRENT_TIMESTAMP
    WHERE inventory_id = NEW.inventory_id;
END;
$$

-- เรียกใช้งานทริกเกอร์
INSERT INTO rental (rental_date, inventory_id, customer_id, staff_id)
VALUES ('2022-10-20', 1526, 6, 1);

-- ตรวจสอบการทำงานของทริกเกอร์
SELECT * FROM inventory WHERE inventory_id = 1526;
```

ตัวอย่างการเขียนโดเมน (Domains)

3_Function.sql

```sql
-- โดเมนนี้กำหนดชนิดข้อมูลและข้อจำกัดสำหรับคอลัมน์ rating
CREATE DOMAIN rating
AS text
CHECK (VALUE IN ('G', 'PG', 'PG-13', 'R', 'NC-17'));

-- ตัวอย่างการเปลี่ยนชนิดข้อมูลของคอลัมน์ rating ในตาราง film
ALTER TABLE film
ALTER COLUMN rating SET DATA TYPE rating;
-- ลบโดเมน
DROP DOMAIN rating;
```

ตัวอย่างการเรียกใช้ Procedure create_product ภายนอก

ภาษา PHP:

PHP

```php
<?php

$conn = pg_connect("host=localhost port=5432 dbname=mydb user=postgres p

$company_id = 33;
$product_type_name = 'สินค้าใหม่';
$name = 'ชื่อสินค้าใหม่';
$description = 'รายละเอียดสินค้าใหม่';
$price = 100;
$unit = 'ชิ้น';

$query = "CALL create_product($company_id, $product_type_name, $name, $d

pg_query($conn, $query);

pg_close($conn);

?>
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

ตัวอย่างการเรียกใช้ Procedure create_product ภายนอก

ภาษา JavaScript:

JavaScript

```javascript
const { Pool } = require('pg');

const pool = new Pool({
  host: 'localhost',
  port: 5432,
  database: 'mydb',
  user: 'postgres',
  password: 'postgres',
});

const company_id = 33;
const product_type_name = 'สินค้าใหม่';
const name = 'ชื่อสินค้าใหม่';
const description = 'รายละเอียดสินค้าใหม่';
const price = 100;
const unit = 'ชิ้น';

const query = `CALL create_product(${company_id}, ${product_type_name},

pool.query(query, (err, res) => {
  if (err) {
    console.error(err);
  } else {
    console.log(res);
  }
});

pool.end();
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

ตัวอย่างการเรียกใช้ Procedure create_product ภายนอก

ภาษา Python:

Python

```python
import psycopg2

conn = psycopg2.connect(
  host="localhost",
  port="5432",
  database="mydb",
  user="postgres",
  password="postgres",
)

cur = conn.cursor()

company_id = 33
product_type_name = 'สินค้าใหม่'
name = 'ชื่อสินค้าใหม่'
description = 'รายละเอียดสินค้าใหม่'
price = 100
unit = 'ชิ้น'

query = f"CALL create_product({company_id}, {product_type_name}, {name},

cur.execute(query)

conn.commit()

cur.close()
conn.close()
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

ตัวอย่างการเรียกใช้ Procedure create_product ภายนอก

ภาษา C#:

C#

```
using Npgsql;

using (var conn = new NpgsqlConnection("Host=localhost;Port=5432;Databas
{
  conn.Open();

  var cmd = new NpgsqlCommand("CALL create_product(@company_id, @product

  cmd.Parameters.AddWithValue("@company_id", 33);
  cmd.Parameters.AddWithValue("@product_type_name", "สินค้าใหม่");
  cmd.Parameters.AddWithValue("@name", "ชื่อสินค้าใหม่");
  cmd.Parameters.AddWithValue("@description", "รายละเอียดสินค้าใหม่");
  cmd.Parameters.AddWithValue("@price", 100);
  cmd.Parameters.AddWithValue("@unit", "ชิ้น");

  cmd.ExecuteNonQuery();

  conn.Close();
}
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

Section 6

เรียนรู้การปรับแต่งประสิทธิภาพ PostgreSQL

- ตรวจสอบประสิทธิภาพ PostgreSQL
- ปรับแต่งการตั้งค่า PostgreSQL
- วิเคราะห์คิวรี่ SQL
- เทคนิคการ Optimize ฐานข้อมูล
- ตัวอย่างการปรับแต่งประสิทธิภาพ PostgreSQL

Database Tunning

Performance Tuning สำหรับ PostgreSQL

1. Hardware Configuration
2. OS Configuration
3. SQL Tuning
4. Query Plan
5. Configuration (postgresql.conf)

1. Hardware Configuration

- **CPU:**
  - เลือก CPU ที่มีความเร็วสูง
  - เลือก CPU ที่มีจำนวนคอร์ที่เหมาะสมกับ workload ของระบบ
  - ตัวอย่าง:
    - สำหรับ workload ทั่วไป: เลือก CPU ที่มีความเร็ว 2 GHz ขึ้นไป
    - สำหรับ workload ที่หนัก: เลือก CPU ที่มีความเร็ว 4 GHz ขึ้นไป
    - สำหรับ workload ที่ต้องการใช้ multithreading: เลือก CPU ที่มีจำนวนคอร์ 4 คอร์ขึ้นไป
- **Memory:**
  - เพิ่ม RAM ให้เพียงพอสำหรับการใช้งาน PostgreSQL
  - โดยทั่วไปแล้ว PostgreSQL ต้องการ RAM ประมาณ 1GB ต่อ 1GB ของ database size
  - ตัวอย่าง:
    - สำหรับ database ขนาด 10GB: เลือก RAM อย่างน้อย 10GB
    - สำหรับ database ขนาด 100GB: เลือก RAM อย่างน้อย 100GB
- **Storage:**
  - เลือกใช้ SSD แทน HDD เพื่อเพิ่มความเร็วในการอ่านเขียนข้อมูล
  - SSD มีความเร็วในการอ่านเขียนข้อมูลเร็วกว่า HDD ประมาณ 10 เท่า
  - ตัวอย่าง:
    - ใช้ SSD สำหรับ storing database files
    - ใช้ SSD สำหรับ storing temporary files
- **Network:**
  - ตรวจสอบให้แน่ใจว่ามี bandwidth ที่เพียงพอสำหรับการใช้งาน PostgreSQL
  - Bandwidth ที่ต้องการขึ้นอยู่กับ workload ของระบบ
  - ตัวอย่าง:
    - สำหรับ workload ทั่วไป: Bandwidth 100 Mbps น่าจะเพียงพอ
    - สำหรับ workload ที่หนัก: Bandwidth 1 Gbps น่าจะเพียงพอ

2. OS Configuration

- **Kernel:**
  - ปรับแต่ง kernel parameters ให้เหมาะสมกับ PostgreSQL
  - ตัวอย่าง kernel parameters ที่สามารถปรับแต่งได้:
    - `vm.dirty_background_ratio` : ตั้งค่า percentage ของ dirty memory ที่จะเขียนไปยัง disk
    - `vm.dirty_ratio` : ตั้งค่า percentage ของ memory ที่จะใช้สำหรับ dirty buffers
    - `vm.swappiness` : ตั้งค่า tendency ของ kernel ในการ swap memory pages ไปยัง disk
- **Memory Management:**
  - ตั้งค่า memory allocation ให้เหมาะสม
  - ตัวอย่าง:
    - ใช้ memory allocator ที่ออกแบบมาสำหรับ PostgreSQL เช่น jemalloc
    - ตั้งค่า huge pages ให้เหมาะสม
- **File System:**
  - เลือก file system ที่เหมาะสมกับ PostgreSQL
  - ตัวอย่าง file system ที่เหมาะสมกับ PostgreSQL:
    - ext4
    - XFS
    - ZFS
- **Networking:**
  - ปรับแต่ง network parameters ให้เหมาะสม
  - ตัวอย่าง network parameters ที่สามารถปรับแต่งได้:
    - `tcp_keepalive_time` : ตั้งค่า keepalive time สำหรับ TCP connections
    - `tcp_keepalive_probes` : ตั้งค่าจำนวน probes ที่จะส่งก่อนที่จะถือว่า TCP connection ตาย
    - `tcp_rmem` : ตั้งค่า receive buffer size สำหรับ TCP connections

3. SQL Tuning

- **Analyze queries:**
  - วิเคราะห์ queries หาจุดที่ช้าและปรับแก้ให้มีประสิทธิภาพ
  - เครื่องมือที่ใช้สำหรับวิเคราะห์ queries:
    - EXPLAIN
    - pg_stat_statements
    - pg_profile
- **Use appropriate indexes:**
  - สร้าง indexes บน columns ที่ใช้ในการค้นหา
  - Indexes ช่วยให้ PostgreSQL ค้นหาข้อมูลได้เร็วขึ้น
- **Use EXPLAIN:**
  - ใช้ EXPLAIN เพื่อดู query plan และหาจุดที่ช้า
  - EXPLAIN แสดงแผนการทำงานของ queries
- **Use prepared statements:**
  - ใช้ prepared statements เพื่อลด overhead ของการ compile queries
  - Prepared statements ช่วยให้ PostgreSQL

ตัวอย่างการใช้คำสั่ง EXPLAIN เพื่อดูประสิทธิภาพของการเขียน SQL Query

```
EXPLAIN ANALYZE
SELECT
  c.first_name AS customer_first_name,
  c.last_name AS customer_last_name,
  f.title AS film_title,
  r.rental_date
FROM customer c
JOIN rental r ON c.customer_id = r.customer_id
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film f ON i.film_id = f.film_id;
```

4. Query Plan

- **Understand query plans:**
  - เข้าใจวิธีการทำงานของ query plan
  - Query plan แสดงขั้นตอนที่ PostgreSQL ใช้ในการดึงข้อมูล
- **Optimize query plans:**
  - ปรับแต่ง query plan ให้มีประสิทธิภาพ
  - เทคนิคการ optimize query plans: *

    *
- **Use materialized views:**
  - ใช้ materialized views เพื่อ pre-compute results ของ queries ที่ใช้บ่อย
  - Materialized views ช่วยให้ PostgreSQL

5. Configuration (postgresql.conf)

- **Tuning parameters:**
  - ปรับแต่ง parameters ใน postgresql.conf ให้เหมาะสมกับ workload ของระบบ
  - ตัวอย่าง parameters ที่สามารถปรับแต่งได้:

ตัวอย่าง parameters ที่สามารถปรับแต่งได้ใน postgresql.conf

Parameters ทั่วไป:

- `max_connections` : ตั้งค่าจำนวน concurrent connections สูงสุด
- `shared_buffers` : ตั้งค่าจำนวน shared buffers ที่ใช้สำหรับ caching data
- `work_mem` : ตั้งค่า memory ที่ใช้สำหรับ sorting and hashing operations
- `maintenance_work_mem` : ตั้งค่า memory ที่ใช้สำหรับ maintenance operations
- `checkpoint_segments` : ตั้งค่าจำนวน segments ที่จะเขียนไปยัง disk ใน checkpoint
- `wal_buffers` : ตั้งค่าจำนวน buffers ที่ใช้สำหรับ storing write-ahead log (WAL) data
- `fsync` : ตั้งค่าวิธีการ sync data ไปยัง disk

Parameters สำหรับการ tuning queries:

- `enable_nestloop` : เปิดใช้งาน nestloop join algorithm
- `join_cost_factor` : ตั้งค่า cost factor สำหรับ join operations
- `seq_scan_cost_factor` : ตั้งค่า cost factor สำหรับ sequential scans
- `random_page_cost` : ตั้งค่า cost factor สำหรับ accessing random pages on disk

Parameters สำหรับการ tuning performance:

- `effective_cache_size` : ตั้งค่า effective size ของ cache
- `vacuum_cost_delay` : ตั้งค่า delay
- `bgwriter_delay` : ตั้งค่า delay
- `wal_writer_delay` : ตั้งค่า delay

Parameters สำหรับการ tuning security:

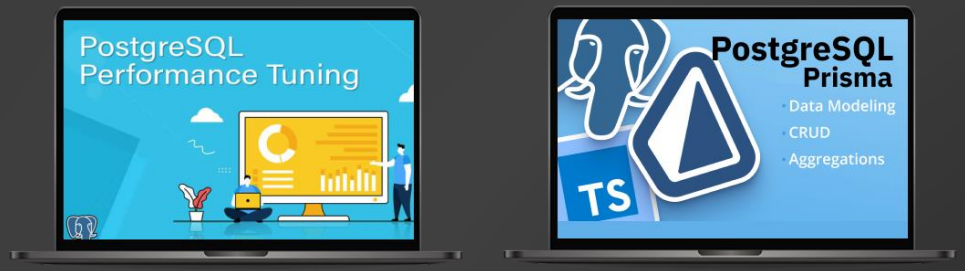
- `password_encryption` : ตั้งค่า encryption algorithm สำหรับ passwords
- `superuser_reserved_roles` : ตั้งค่า roles
- `client_authentication` : ตั้งค่า authentication method
- `ssl` : ตั้งค่า SSL encryption

หมายเหตุ:

- ตัวอย่าง parameters เหล่านี้เป็นเพียงส่วนหนึ่งของ parameters ที่สามารถปรับแต่งได้ใน postgresql.conf
- การปรับแต่ง parameters เหล่านี้อาจส่งผลต่อ performance ของ PostgreSQL
- ควรทดสอบการปรับแต่ง parameters เหล่านี้ก่อนใช้งานจริง

เว็บตัวอย่างในการกำหนดค่า tunning ที่เหมาะสม

https://pgtune.leopard.in.ua/#/
https://pgconfigurator.cybertec-postgresql.com/
https://www.pgconfig.org/



DAY 4

Section 6: เรียนรู้การปรับแต่งประสิทธิภาพ PostgreSQL

Section 6 (ต่อ)

เรียนรู้การปรับแต่งประสิทธิภาพ PostgreSQL

- ตรวจสอบประสิทธิภาพ PostgreSQL
- ปรับแต่งการตั้งค่า PostgreSQL
- วิเคราะห์คิวรี่ SQL
- เทคนิคการ Optimize ฐานข้อมูล
- ตัวอย่างการปรับแต่งประสิทธิภาพ PostgreSQL

Database Tunning

Performance Tuning สำหรับ PostgreSQL

1. Hardware Configuration
2. OS Configuration
3. SQL Tuning
4. Query Plan
5. Configuration (postgresql.conf)

เว็บตัวอย่างในการกำหนดค่า tunning ที่เหมาะสม

https://pgtune.leopard.in.ua/#/
https://pgconfigurator.cybertec-postgresql.com/
https://www.pgconfig.org/

This PC › Local Disk (C:) › Program Files › PostgreSQL › 15 › data ›

Name	Date modified	Type	Size
base	14/2/2567 23:42	File folder	
global	16/2/2567 14:31	File folder	
log	16/2/2567 0:00	File folder	
pg_commit_ts	25/10/2566 1:00	File folder	
pg_dynshmem	25/10/2566 1:00	File folder	
pg_logical	16/2/2567 0:27	File folder	
pg_multixact	25/10/2566 1:00	File folder	
pg_notify	25/10/2566 1:00	File folder	
pg_replslot	25/10/2566 1:00	File folder	
pg_serial	25/10/2566 1:00	File folder	
pg_snapshots	25/10/2566 1:00	File folder	
pg_stat	14/1/2567 18:26	File folder	
pg_stat_tmp	25/10/2566 1:00	File folder	
pg_subtrans	25/10/2566 1:00	File folder	
pg_tblspc	25/10/2566 1:00	File folder	
pg_twophase	25/10/2566 1:00	File folder	
pg_wal	14/2/2567 23:54	File folder	
pg_xact	25/10/2566 1:00	File folder	
.postgresql.auto.conf.swp	16/2/2567 14:41	SWP File	1 KB
current_logfiles	16/2/2567 14:31	File	1 KB
pg_hba.conf	25/10/2566 13:31	CONF File	5 KB
pg_ident.conf	25/10/2566 1:00	CONF File	2 KB
PG_VERSION	25/10/2566 1:00	File	1 KB
postgresql.auto.conf	16/2/2567 14:41	CONF File	1 KB
postgresql.conf	16/2/2567 14:26	CONF File	30 KB
postmaster.opts	14/1/2567 18:26	OPTS File	1 KB
postmaster.pid	14/1/2567 18:26	PID File	1 KB

PostgreSQL File กำหนดค่า tunning ที่เหมาะสม

- postgresql.conf
- postgresql.auto.conf

ความแตกต่างระหว่าง file postgresql.auto.conf และ postgresql.conf

postgresql.conf และ **postgresql.auto.conf** เป็นไฟล์การตั้งค่าสำหรับ PostgreSQL แต่มีความแตกต่างกันดังนี้:

1. วัตถุประสงค์:

- **postgresql.conf:** ใช้สำหรับตั้งค่าการกำหนดค่าเริ่มต้นของเซิร์ฟเวอร์ PostgreSQL ไฟล์นี้สามารถแก้ไขด้วยตนเองได้
- **postgresql.auto.conf:** ใช้สำหรับเก็บค่าการตั้งค่าที่เปลี่ยนแปลงโดยใช้คำสั่ง ALTER SYSTEM ไฟล์นี้ไม่ควรแก้ไขด้วยตนเอง

2. ตำแหน่ง:

- **postgresql.conf:** อยู่ในไดเร็กทอรีหลักของ PostgreSQL
- **postgresql.auto.conf:** อยู่ในไดเร็กทอรีข้อมูล PostgreSQL

postgresql.conf

postgresql.auto.conf

3. การจัดการ:

- **postgresql.conf:**
  - สามารถแก้ไขด้วยตนเอง
  - การเปลี่ยนแปลงจะมีผลเมื่อเซิร์ฟเวอร์ PostgreSQL เริ่มต้นใหม่
- **postgresql.auto.conf:**
  - ไม่ควรแก้ไขด้วยตนเอง
  - การเปลี่ยนแปลงจะมีผลทันทีโดยไม่ต้องรีสตาร์ทเซิร์ฟเวอร์

4. ตัวอย่างการใช้งาน:

- **postgresql.conf:**
  - ตั้งค่าจำนวนการเชื่อมต่อพร้อมกันสูงสุด
  - ตั้งค่าขนาดหน่วยความจำที่ใช้โดย PostgreSQL
  - ตั้งค่าพฤติกรรมการบันทึก
- **postgresql.auto.conf:**
  - เปลี่ยนค่า max_connections ชั่วคราว
  - เปิดใช้งานการบันทึกเพิ่มเติมสำหรับการแก้ไขปัญหา

สรุป:

- **postgresql.conf:** ใช้สำหรับตั้งค่าการกำหนดค่าเริ่มต้น
- **postgresql.auto.conf:** ใช้สำหรับเก็บค่าการตั้งค่าที่เปลี่ยนแปลงแบบไดนามิก

ข้อควรระวัง:

- ไม่ควรแก้ไขไฟล์ postgresql.auto.conf ด้วยตนเอง
- การเปลี่ยนแปลงที่ไม่ถูกต้องอาจทำให้เซิร์ฟเวอร์ PostgreSQL ทำงานไม่เสถียร

PostgreSQL File กำหนดค่า tunning ที่เหมาะสม

postgresql.conf

postgresql.conf

C: > Program Files > PostgreSQL > 15 > data > postgresql.conf

```
# - Connection Settings -
listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for all
                                # (change requires restart)
port = 5432                     # (change requires restart)
max_connections = 100           # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directories = ''   # comma-separated list of directories
                        # (change requires restart)
#unix_socket_group = ''         # (change requires restart)
#unix_socket_permissions = 0777     # begin with 0 to use octal notation
                        # (change requires restart)
#bonjour = off                  # advertise server via Bonjour
                        # (change requires restart)
#bonjour_name = ''              # defaults to the computer name
                        # (change requires restart)

# - TCP settings -
# see "man tcp" for details

#tcp_keepalives_idle = 0        # TCP_KEEPIDLE, in seconds;
                        # 0 selects the system default
#tcp_keepalives_interval = 0        # TCP_KEEPINTVL, in seconds;
                        # 0 selects the system default
#tcp_keepalives_count = 0       # TCP_KEEPCNT;
```

postgresql.auto.conf

postgresql.auto.conf

C: > Program Files > PostgreSQL > 15 > data > postgresql.auto.conf

```
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
```

ตัวอย่างการเปลี่ยน max_connections ใน postgresql.auto.conf

```
ALTER SYSTEM SET max_connections = '300';
```

pwsh in samit

```
postgres=# ALTER SYSTEM SET max_connections = '300';
ALTER SYSTEM
postgres=#
```

postgresql.auto.conf

postgresql.auto.conf

C: > Program Files > PostgreSQL > 15 > data > postgresql.auto.conf

```
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
max_connections = '300'
```

ถ้าฉันแก้ไข ค่า max_connections ทั้ง 2 ไฟล์ postgresql จะเอาค่าจากไฟล์ไหนมาใช้ ?

postgresql.auto.conf

postgresql.auto.conf

C: > Program Files > PostgreSQL > 15 > data > postgresql.auto.conf

```
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM co
max_connections = '300'
```

postgresql.conf

postgresql.conf

C: > Program Files > PostgreSQL > 15 > data > postgresql.conf

```
# - Connection Settings -
listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for all
                                # (change requires restart)
port = 5432                     # (change requires restart)
max_connections = 100           # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
```

ถ้าฉันแก้ไข ค่า max_connections ทั้ง 2 ไฟล์ postgresql จะเอาค่าจากไฟล์ไหนมาใช้ ?

PostgreSQL จะใช้ค่า **max_connections** จาก **postgresql.auto.conf ก่อน postgresql.conf**

เหตุผล:

- postgresql.auto.conf มีลำดับความสำคัญสูงกว่า postgresql.conf
- postgresql.auto.conf ถูกออกแบบมาเพื่อเก็บค่าการตั้งค่าที่เปลี่ยนแปลงแบบไดนามิก

ตัวอย่าง:

- สมมติว่า postgresql.conf ตั้งค่า max_connections เป็น 100
- สมมติว่า postgresql.auto.conf ตั้งค่า max_connections เป็น 50

ผลลัพธ์:

- PostgreSQL จะใช้ค่า max_connections 50 **จาก postgresql.auto.conf**

ข้อควรระวัง:

- ไม่ควรแก้ไขไฟล์ postgresql.auto.conf ด้วยตนเอง
- การเปลี่ยนแปลงที่ไม่ถูกต้องอาจทำให้เซิร์ฟเวอร์ PostgreSQL ทำงานไม่เสถียร

ฉันจะลบค่าที่เขียนไว้ใน postgresql.auto.conf อย่างไร ?

วิธีลบค่าที่เขียนไว้ใน postgresql.auto.conf

มีสองวิธีในการลบค่าที่เขียนไว้ใน postgresql.auto.conf:

1. ใช้คำสั่ง ALTER SYSTEM:

- วิธีนี้ปลอดภัยและสะดวกที่สุด
- ตัวอย่าง:

```sql
ALTER SYSTEM RESET max_connections;
```

ใช้โค้ดอย่างระมัดระวัง ดูข้อมูลเพิ่มเติม

- คำสั่งนี้จะลบค่า max_connections ที่เขียนไว้ใน postgresql.auto.conf
- ค่า max_connections จะกลับไปใช้ค่าเริ่มต้นจาก postgresql.conf

2. แก้ไขไฟล์ postgresql.auto.conf:

- วิธีนี้ **ไม่** แนะนำ
- การแก้ไขไฟล์ postgresql.auto.conf ด้วยตนเองอาจทำให้เกิดข้อผิดพลาด
- **ขั้นตอน:**
  - เปิดไฟล์ postgresql.auto.conf ด้วยโปรแกรมแก้ไขข้อความ
  - ค้นหาบรรทัดที่มีค่าที่ต้องการลบ
  - ลบค่าที่ต้องการ
  - บันทึกไฟล์

หมายเหตุ:

- การ restart PostgreSQL **ไม่** ลบค่าที่เขียนไว้ใน postgresql.auto.conf

ฉันจะลบค่าที่เขียนไว้ใน postgresql.auto.conf อย่างไร ?

pwsh in samit

```
postgres=# ALTER SYSTEM SET max_connections = '300';
ALTER SYSTEM
postgres=# ALTER SYSTEM RESET max_connections;
ALTER SYSTEM
postgres=#
```

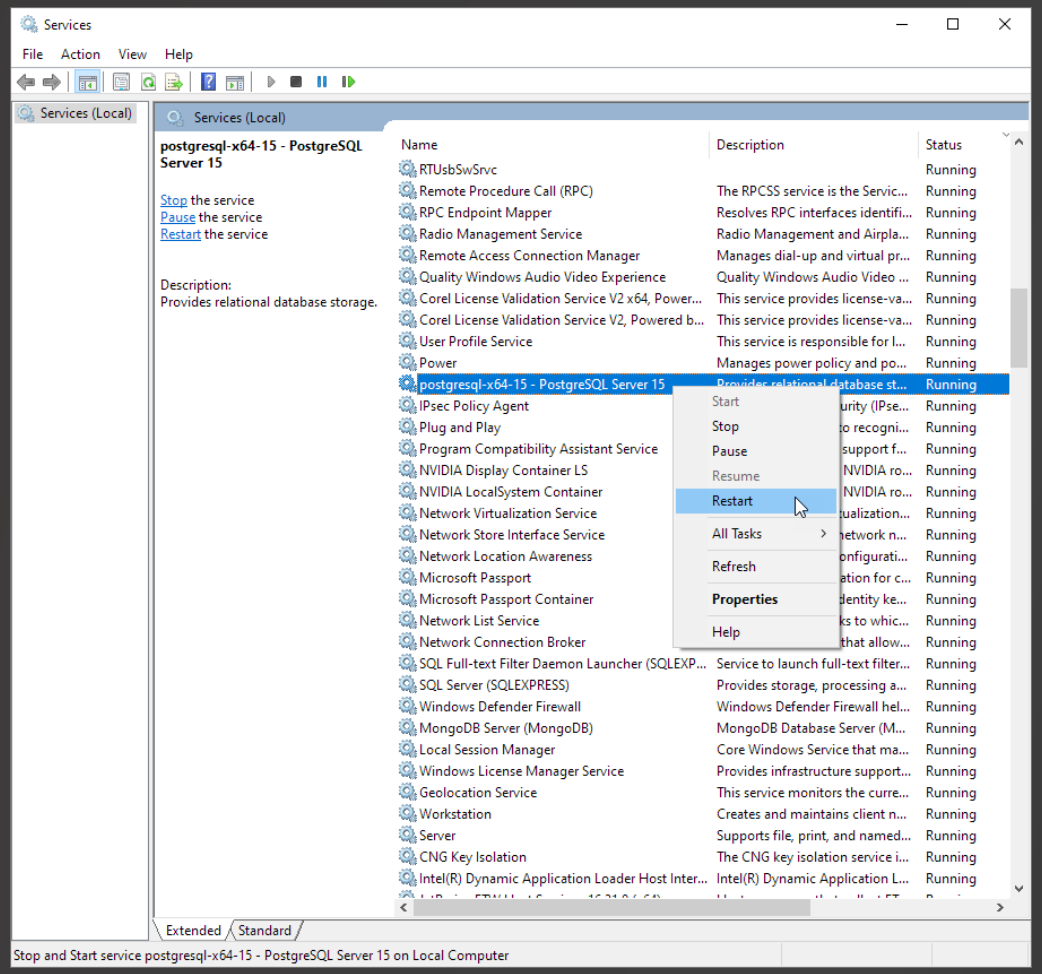
stgreSQL - Visual Studio Code

postgresql.auto.conf

C: > Program Files > PostgreSQL > 15 > data > postgresql.auto.conf

```
# Do not edit this file manually!
# It will be overwritten by the
ALTER SYSTEM command.
```

ฉันจะ restart postgresql บน windows , macos และ linux ทำอย่างไร ?



วิธีเริ่มต้น PostgreSQL ใหม่บน Windows, macOS และ Linux

Windows:

1. เปิด **Run Window** โดยกดปุ่ม **Win + R**
2. พิมพ์ **services.msc** และกด Enter
3. ค้นหาบริการ **PostgreSQL**
4. คลิกขวาที่บริการ PostgreSQL และเลือก **Restart**

macOS:

1. เปิด **Terminal**
2. พิมพ์คำสั่งต่อไปนี้:

```
sudo launchctl restart postgres
```

Linux:

วิธีที่ 1:

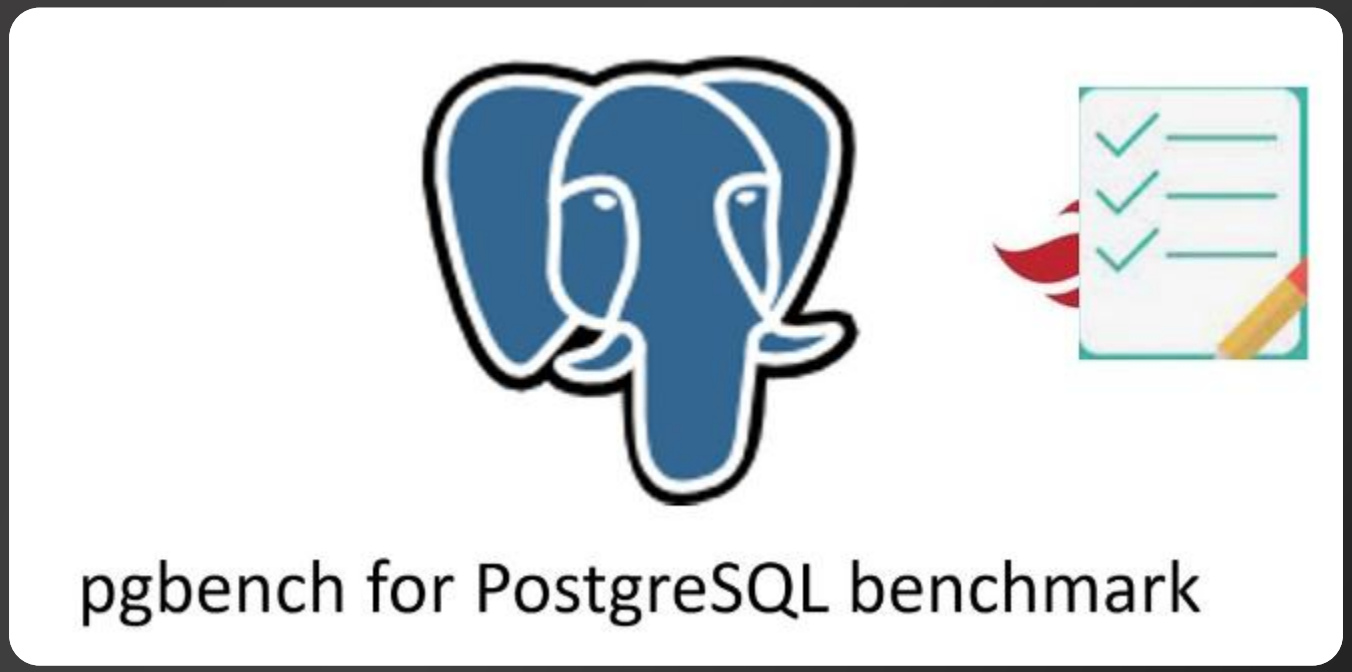
1. เปิด **Terminal**
2. พิมพ์คำสั่งต่อไปนี้:

```
sudo systemctl restart postgresql.service
```

วิธีที่ 2:

1. พิมพ์คำสั่งต่อไปนี้:

```
sudo service postgresql restart
```


pgbench for PostgreSQL benchmark

Step 1: สร้างฐานข้อมูลตัวอย่างไว้ทดสอบ bench mark

```
CREATE DATABASE example WITH ENCODING unicode
LC_COLLATE="en_US.UTF-8" LC_CTYPE="en_US.UTF-8"
TEMPLATE  template0;
```

จากนั้นออกจาก psql

```
\q
```

Step 2: คำสั่งทดสอบ dummpy ฐานข้อมูลด้วย pgbench

```
pgbench -U postgres -i -s 50 example
```

ในคำสั่งด้านบน เราสั่งใช้งาน pgbench ด้วยตัวเลือก -i และ -s ตามด้วยชื่อฐานข้อมูล (example)

ตัวเลือก -i (initialize) จะบอก pgbench ให้ทำการปรับแต่งฐานข้อมูลที่ระบุ ซึ่งหมายความว่า pgbench จะสร้างตารางต่างๆ ภายในฐานข้อมูล example ดังนี้:

- pgbench_accounts: ตารางนี้จะเก็บข้อมูลบัญชีผู้ใช้
- pgbench_tellers: ตารางนี้จะเก็บข้อมูลพนักงานธนาคาร
- pgbench_branches: ตารางนี้จะเก็บข้อมูลสาขา
- pgbench_history: ตารางนี้จะเก็บประวัติการทำธุรกรรม

Step 2: คำสั่งทดสอบ dummpy ฐานข้อมูลด้วย pgbench

```
pgbench -U postgres -i -s 50 example
```

```
table                    # of rows
---------------------------------
pgbench_branches         1
pgbench_tellers          10
pgbench_accounts         100000
pgbench_history          0
```

ตามค่าเริ่มต้น pgbench จะสร้างตารางตามที่กล่าวไปข้างต้น พร้อมจำนวนแถวที่ระบุไว้ด้านบน การกระทำนี้จะสร้างฐานข้อมูลขนาด 16 เมกะไบต์

เนื่องจากเราจะใช้ pgbench เพื่อวัดการเปลี่ยนแปลงประสิทธิภาพ ฐานข้อมูลขนาด 16 เมกะไบต์จึงไม่เพียงพอที่จะทดสอบความสามารถของระบบ นี่คือที่มาของตัวเลือก -s (scaling)

ตัวเลือก -s ใช้เพื่อเพิ่มจำนวนแถวที่ป้อนลงในแต่ละตาราง ในคำสั่งด้านบน เราใส่ตัวเลือก "scaling" ที่ 50 ซึ่งบอก pgbench ให้สร้างฐานข้อมูลที่มีขนาดใหญ่กว่าค่าเริ่มต้น 50 เท่า

นั่นหมายความว่าตาราง pgbench_accounts ของเรามีบันทึกจำนวน 5,000,000 บันทึก นอกจากนี้ ขนาดฐานข้อมูลของเรายังกลายเป็น 800 เมกะไบต์ (50 x 16 เมกะไบต์)

Step 3: ชุดคำสั่งทดสอบ perfomance ของฐานข้อมูล

```
pgbench -U postgres -c 10 -j 2 -t 10000 example
```

ตัวเลือกสำหรับคำสั่ง pgbench:

-c (clients) เป็นตัวเลือกแรก ใช้กำหนดจำนวนลูกค้า (client) ที่จะเชื่อมต่อ ในตัวอย่างนี้ ใช้ 10 บอกให้ pgbench รันด้วยลูกค้า 10 คน

pgbench จะทำการทดสอบโดยเปิดเซสชันที่แตกต่างกัน 10 เซสชัน

-j (threads) ตัวเลือกนี้ใช้กำหนดจำนวนกระบวนการ worker สำหรับ pgbench ในคำสั่งด้านบน กำหนดค่าเป็น 2 ซึ่งจะบอกให้ pgbench เริ่มกระบวนการ worker สองกระบวนการระหว่างการวัดประสิทธิภาพ

-t (transactions) ตัวเลือกนี้ใช้กำหนดจำนวนธุรกรรมที่จะดำเนินการ ในคำสั่งด้านบน มีค่า 10,000 ซึ่งหมายความว่าแต่ละเซสชันของลูกค้าจะดำเนินการธุรกรรม 10,000 รายการ

สรุป: การทดสอบพื้นฐาน รัน pgbench worker processes สองกระบวนการ จำลองธุรกรรม 10,000 รายการ จาก 10 ลูกค้า รวมเป็น 100,000 ธุรกรรม

Step 3: ชุดคำสั่งทดสอบ perfomance ของฐานข้อมูล

```
pgbench -U postgres -c 10 -j 2 -t 10000 example
```

```
pgbench (15.4)
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 50
query mode: simple
number of clients: 10
number of threads: 2
maximum number of tries: 1
number of transactions per client: 10000
number of transactions actually processed: 100000/100000
number of failed transactions: 0 (0.000%)
latency average = 1.442 ms
initial connection time = 479.672 ms
tps = 6936.327977 (without initial connection time)
```

Step 3: ชุดคำสั่งทดสอบ perfomance ของฐานข้อมูล

```
pgbench -U postgres -c 10 -j 2 -t 10000 example
```

```
tps = 2394.718707 (including connections establishing)
tps = 2394.874350 (excluding connections establishing)
```

- **tps = 2394.718707:** รวมเวลาที่ใช้ในการสร้างการเชื่อมต่อ (connection establishing)
- **tps = 2394.874350:** ไม่รวมเวลาที่ใช้ในการสร้างการเชื่อมต่อ

กล่าวคือ:

- **tps แรก** นั้นวัดประสิทธิภาพโดยรวมของระบบ ซึ่งรวมถึงเวลาที่ใช้ในการสร้างการเชื่อมต่อระหว่างไคลเอนต์และฐานข้อมูล
- **tps ที่สอง** นั้นวัดประสิทธิภาพเฉพาะของระบบโดยไม่รวมเวลาที่ใช้ในการสร้างการเชื่อมต่อ ซึ่งจะแสดงประสิทธิภาพการทำงานจริงของระบบเมื่อมีการเชื่อมต่อที่พร้อมใช้งาน

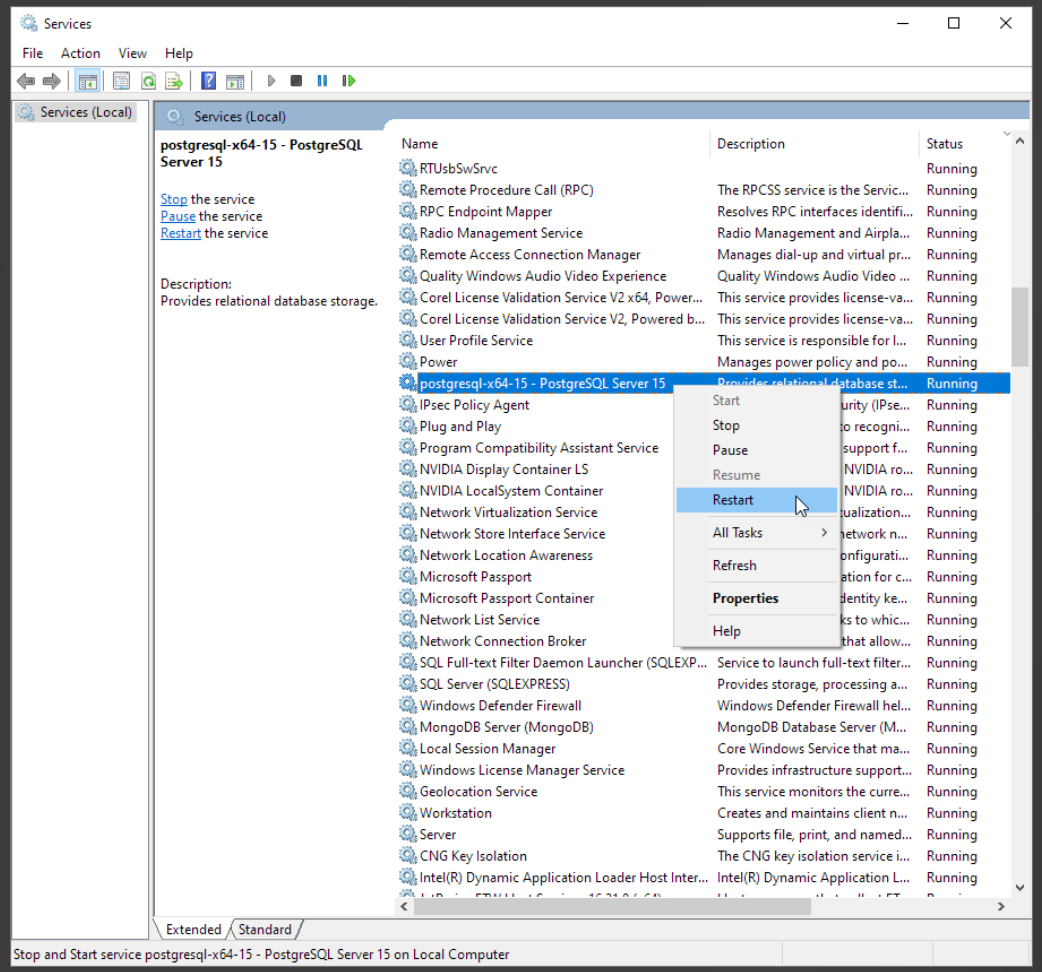
"จากผลลัพธ์เหล่านี้ ดูเหมือนว่าประสิทธิภาพพื้นฐานของเราอยู่ที่ 2,394 ทรานแซคชั่นต่อวินาที ลองดูว่าเราสามารถเพิ่มจำนวนนี้ได้หรือไม่ โดยการปรับแต่งพารามิเตอร์การกำหนดค่าอย่างง่ายภายใน PostgreSQL"

มาแก้ไขค่า postgresql.conf เพื่อปรับปรุง performance กัน

มาแก้ไขค่า postgresql.conf เพื่อปรับปรุง performance กัน

```
# Network Related Configuration
max_connections = 300                    # (change requires restart)

# Memory Configuration
shared_buffers = 8GB                     # min 128kB
effective_cache_size = 24GB
work_mem = 32MB                          # min 64kB
maintenance_work_mem = 2GB               # min 1MB

# Storage Configuration
effective_io_concurrency = 200           # 1-1000; 0 disables prefetching

# Checkpoint Related Configuration
checkpoint_timeout = 15min               # range 30s-1d
checkpoint_completion_target = 0.9       # checkpoint target duration, 0.0 - 1.0
max_wal_size = 4GB
min_wal_size = 1GB
```

Restart the PostgreSQL service



วิธีเริ่มต้น PostgreSQL ใหม่บน Windows, macOS และ Linux

Windows:

1. เปิด **Run Window** โดยกดปุ่ม **Win + R**
2. พิมพ์ **services.msc** และกด Enter
3. ค้นหาบริการ **PostgreSQL**
4. คลิกขวาที่บริการ PostgreSQL และเลือก **Restart**

macOS:

1. เปิด **Terminal**
2. พิมพ์คำสั่งต่อไปนี้:

```
sudo launchctl restart postgres
```

Linux:

วิธีที่ 1:

1. เปิด **Terminal**
2. พิมพ์คำสั่งต่อไปนี้:

```
sudo systemctl restart postgresql.service
```

วิธีที่ 2:

1. พิมพ์คำสั่งต่อไปนี้:

```
sudo service postgresql restart
```

ทดสอบ perfomance ของฐานข้อมูลอีกครั้ง

```
pgbench -U postgres -c 10 -j 2 -t 10000 example
```

```
pgbench (15.4)
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 50
query mode: simple
number of clients: 10
number of threads: 2
maximum number of tries: 1
number of transactions per client: 10000
number of transactions actually processed: 100000/100000
number of failed transactions: 0 (0.000%)
latency average = 1.260 ms
initial connection time = 380.986 ms
tps = 7937.268276 (without initial connection time)
```

ทดสอบ perfomance ของฐานข้อมูลอีกครั้ง

Before tuning

```
pgbench (15.4)
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 50
query mode: simple
number of clients: 10
number of threads: 2
maximum number of tries: 1
number of transactions per client: 10000
number of transactions actually processed: 100000/100000
number of failed transactions: 0 (0.000%)
latency average = 1.442 ms
initial connection time = 479.672 ms
tps = 6936.327977 (without initial connection time)
```

After tuning

```
pgbench (15.4)
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 50
query mode: simple
number of clients: 10
number of threads: 2
maximum number of tries: 1
number of transactions per client: 10000
number of transactions actually processed: 100000/100000
number of failed transactions: 0 (0.000%)
latency average = 1.260 ms
initial connection time = 380.986 ms
tps = 7937.268276 (without initial connection time)
```

Backup And Restore

Backup

รูปแบบการสำรองข้อมูล PostgreSQL ยอดนิยม

รูปแบบการสำรองข้อมูล PostgreSQL ยอดนิยม

1. Plain Text:

- เหมาะสำหรับการสำรองข้อมูลขนาดเล็ก
- อ่านและแก้ไขด้วย Notepad ได้
- ตัวอย่าง: `pg_dump -Fc my_database > my_database.sql`

2. Compressed Binary Format (.bak):

- บีบอัดข้อมูล ช่วยลดขนาดไฟล์
- เหมาะสำหรับการสำรองข้อมูลขนาดใหญ่
- ตัวอย่าง: `pg_dump -Fc -Z gzip my_database > my_database.bak`

3. Custom Format:

- รูปแบบเฉพาะที่กำหนดเอง
- รวมข้อมูลเพิ่มเติม เช่น ACL และ comment
- ตัวอย่าง: `pg_dump -Fc -a -C my_database > my_database.custom`

4. Archive Format:

- เก็บถาวร เหมาะสำหรับการกู้คืนข้อมูลทั้งหมด
- ตัวอย่าง: `pg_dumpall -Fc > my_database.tar`

5. Point-in-Time Recovery (PITR):

- สำรองข้อมูลแบบต่อเนื่อง
- กู้คืนข้อมูลไปยังจุดใดๆ ในช่วงเวลา
- ตัวอย่าง: การใช้ WAL (Write-Ahead Logging)

คำแนะนำ:

- เลือกรูปแบบการสำรองข้อมูลตามความต้องการของคุณ
- พิจารณาขนาดของฐานข้อมูล ประเภทของข้อมูล และความถี่ในการสำรองข้อมูล
- ทดสอบการสำรองข้อมูลและกู้คืนข้อมูลเป็นประจำ

เครื่องมือสำรองข้อมูล PostgreSQL:

- pg_dump: เครื่องมือสำรองข้อมูลมาตรฐาน
- pgBackRest: เครื่องมือสำรองข้อมูลแบบอัตโนมัติ
- Barman: เครื่องมือสำรองข้อมูลแบบ point-in-time

คำสั่ง pg_dump และ pg_dumpall

ความแตกต่างของ pg_dump และ pg_dumpall

pg_dump และ **pg_dumpall** เป็นเครื่องมือที่ใช้สำหรับการสำรองข้อมูล PostgreSQL แต่มีความแตกต่างกันดังนี้:

pg_dump:

- สำรองข้อมูลเฉพาะฐานข้อมูลที่ระบุ
- ตัวเลือกเพิ่มเติมสำหรับการสำรองข้อมูลเฉพาะส่วน
- ตัวอย่าง: `pg_dump -Fc my_database > my_database.sql`

pg_dumpall:

- สำรองข้อมูลทั้งหมดใน PostgreSQL cluster
- ตัวเลือกน้อยกว่า pg_dump
- ตัวอย่าง: `pg_dumpall -Fc > all_databases.sql`

สรุป:

คุณสมบัติ	pg_dump	pg_dumpall
ประเภทของข้อมูล	ฐานข้อมูลที่ระบุ	ทั้งหมด
ตัวเลือก	เพิ่มเติม	น้อย
ตัวอย่าง	`pg_dump -Fc my_database > my_database.sql`	`pg_dumpall -Fc > all_databases.sql`

ส่งออกไปยังชีต

ควรใช้ pg_dump เมื่อ:

- ต้องการสำรองข้อมูลเฉพาะฐานข้อมูล
- ต้องการใช้ตัวเลือกเพิ่มเติมสำหรับการสำรองข้อมูลเฉพาะส่วน

ควรใช้ pg_dumpall เมื่อ:

- ต้องการสำรองข้อมูลทั้งหมดใน PostgreSQL cluster
- ต้องการความเรียบง่าย ไม่ต้องการใช้ตัวเลือกเพิ่มเติม

หมายเหตุ:

- ทั้งสองคำสั่งสามารถใช้สำรองข้อมูล PostgreSQL ไปยังไฟล์ plain text หรือ compressed binary format
- ควรทดสอบการสำรองข้อมูลและกู้คืนเป็นประจำ

ตอน backup ข้อมูลทุกอย่างเช่น view, procedure, function , trigger จะไปด้วยหรือไม่ ?

ใช่ค่ะ การ backup ข้อมูล PostgreSQL โดยใช้ pg_dump หรือ pg_dumpall จะรวมทุกอย่างไปด้วย

- **View**
- **Procedure**
- **Function**
- **Trigger**

ตัวอย่าง:

- `pg_dump -Fc my_database > my_database.sql`
- `pg_dumpall -Fc > all_databases.sql`

ทั้งสองคำสั่งนี้จะ:

- เก็บโครงสร้างและข้อมูลของ view, procedure, function, และ trigger
- เก็บข้อมูลในรูปแบบ plain text หรือ compressed binary format

หมายเหตุ:

- คุณสามารถเลือกเฉพาะบางส่วนของ view, procedure, function, และ trigger ที่ต้องการ backup โดยใช้ตัวเลือกเพิ่มเติม
- คุณสามารถใช้เครื่องมืออื่น ๆ เช่น pgBackRest หรือ Barman สำหรับการ backup ข้อมูลแบบอัตโนมัติ

ตัวอย่างการ backup ฐานข้อมูล dvdrental

```
pg_dump -U postgres -h localhost -p 5432 dvdrental >
G:\TrainingWorkshop\BasicSQLforPostgreSQL\BackupDB\dvdrental.bak
```

PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL powershell

```
BasicSQLforPostgreSQL pg_dump -U postgres -h localhost -p 5432 dvdrental > G:\TrainingWorkshop\BasicSQLforPostgreSQL\BackupDB\dvdrental.bak
Password:
samit BasicSQLforPostgreSQL ✓                                   in pwsh at 17:28:34
```

This PC › Local Disk (G:) › TrainingWorkshop › BasicSQLforPostgreSQL › BackupDB

Search BackupDB

Name	Date modified	Type	Size
dvdrental.bak	16/2/2567 17:28	BAK File	5,498 KB

dvdrental.bak
BAK File

Date modified: 16/2/2567 17:28
Size: 5.36 MB
Date created: 16/2/2567 17:28

Restore

ตัวอย่างการ restore ฐานข้อมูล dvdrental ใน server เดียวกัน

คำสั่งสำหรับ restore ฐานข้อมูลที่ backup ไว้

1. pg_restore:

```
pg_restore -d dvdrental dvdrental.bak
```

คำอธิบาย:

- `-d dvdrental` : ระบุชื่อฐานข้อมูลที่จะ restore
- `dvdrental.bak` : ระบุไฟล์ backup

ตัวอย่างเพิ่มเติม:

- **Restore เฉพาะ schema public:**

```
pg_restore -d dvdrental -n public dvdrental_public.bak
```

- **Restore เฉพาะ table 'film':**

```
pg_restore -d dvdrental -t film dvdrental_film.bak
```

คำสั่ง restore ฐานข้อมูลไปยัง server อื่น

1. pg_restore:

```
pg_restore -h <host> -p <port> -d <database> <backup_file>
```

คำอธิบาย:

- `-h <host>` : ระบุ hostname ของ server ใหม่
- `-p <port>` : ระบุ port ของ server ใหม่
- `-d <database>` : ระบุชื่อฐานข้อมูลที่จะ restore
- `<backup_file>` : ระบุไฟล์ backup

ตัวอย่าง:

```
pg_restore -h 192.168.1.100 -p 5432 -d dvdrental dvdrental.bak
```

การ restore เราไม่จำเป็นต้องลบฐานข้อมูลเดิมออกก่อนใช่มั้ย

ใช่ค่ะ คุณไม่จำเป็นต้องลบฐานข้อมูลเดิมออกก่อน การ restore ฐานข้อมูลจะเขียนทับข้อมูลในฐานข้อมูลเดิม

กรณีที่คุณต้องการ restore ฐานข้อมูลไปยังชื่อเดิม:

- ตรวจสอบให้แน่ใจว่าฐานข้อมูลเดิมไม่ใช้งานอยู่
- ปิด PostgreSQL server
- Restore ฐานข้อมูล
- เปิด PostgreSQL server

กรณีที่คุณต้องการ restore ฐานข้อมูลไปยังชื่ออื่น:

- ไม่จำเป็นต้องปิด PostgreSQL server
- Restore ฐานข้อมูลโดยระบุชื่อใหม่

หมายเหตุ:

- การ restore ฐานข้อมูลจะลบข้อมูลทั้งหมดในฐานข้อมูลเดิม
- คุณสามารถ restore ฐานข้อมูลไปยัง PostgreSQL เวอร์ชั่นอื่นได้
- คุณสามารถ restore ฐานข้อมูลไปยัง server อื่นได้

ตัวอย่างการ restore ฐานข้อมูล dvdrental

Step 1: ลบฐานข้อมูลออกก่อน และสร้างฐานข้อมูล dvdrental ขึ้นมาใหม่

```
DROP DATABASE dvdrental;
```

ลบฐานข้อมูลปกติจะ**ไม่ได้**เพราะมีการเชื่อมต่อที่อื่นอยู่

```
postgres=# DROP DATABASE dvdrental;
ERROR:  database "dvdrental" is being accessed by other users
DETAIL:  There are 4 other sessions using the database.
postgres=#
```

ใช้ FORCE จะลบฐานข้อมูล โดยไม่ต้อง ตรวจสอบการเชื่อมต่อ

```
DROP DATABASE dvdrental WITH (FORCE);
```

ตัวอย่างการ restore ฐานข้อมูล dvdrental

Step 2:สร้างฐานข้อมูล dvdrental ขึ้นมาใหม่

```
CREATE DATABASE  dvdrental  WITH  ENCODING unicode LC_COLLATE="en_US.UTF-8"
LC_CTYPE="en_US.UTF-8" TEMPLATE  template0;
```

```
pwsh in samit

postgres=# CREATE DATABASE dvdrental;
CREATE DATABASE
postgres=# \l
                                                    List of databases
      Name      |  Owner   | Encoding |          Collate          |           Ctype           | ICU Locale | Locale Provider |   Access priv
ileges
----------------+----------+----------+---------------------------+---------------------------+------------+-----------------+--------------
---------
 airlines       | postgres | UTF8     | English_United States.874 | English_United States.874 |            | libc            |
 chinook        | postgres | UTF8     | English_United States.874 | English_United States.874 |            | libc            |
 dvdrental      | postgres | UTF8     | English_United States.874 | English_United States.874 |            | libc            |
 postgres       | postgres | UTF8     | English_United States.874 | English_United States.874 |            | libc            |
 roledb         | postgres | UTF8     | en_US.UTF-8               | en_US.UTF-8               |            | libc            |
 sampleairlines | postgres | UTF8     | English_United States.874 | English_United States.874 |            | libc            |
 sampledb       | postgres | UTF8     | Thai_Thailand.874         | Thai_Thailand.874         |            | libc            |
 sampleprismadb | postgres | UTF8     | English_United States.874 | English_United States.874 |            | libc            |
 template0      | postgres | UTF8     | English_United States.874 | English_United States.874 |            | libc            | =c/postgres
```

ตัวอย่างการ restore ฐานข้อมูล dvdrental

Step 3: ออกจาก psql

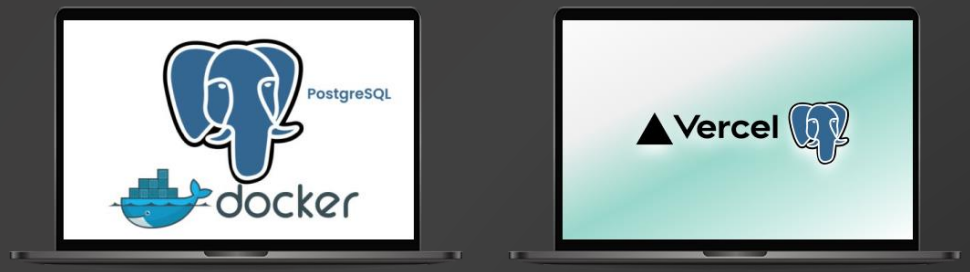
```
\q
```

Step 4: จากนั้นใช้คำสั่ง restore

```
psql -U postgres -h localhost -p 5432 -d dvdrental  <
G:\TrainingWorkshop\BasicSQLforPostgreSQL\BackupDB\dvdrental.bak
```

```
----------------------------------------------
Backup Database
----------------------------------------------
กำหนด permission ให้ directory
----
sudo chown -R postgres:postgres /home/backupdb/

คำสั่ง dump ฐานข้อมูล
---
su - postgres
pg_dump -U postgres -p 7732 demodb > /home/backupdb/demodb.bak

Remote Database
---
pg_dump -h 198.51.100.0 -p 5432 dbname > dbname.bak

pg_dump -F c -h 192.168.0.166 -p 7732 -U postgres -f /home/backupdb/acccloud/demodb01.bak demodb
```

```
----------------------------------------------
Restore Database form .bak file
----------------------------------------------
psql -U postgres demodb < /home/backupdb/acccloud/20221022095902_db.bak
```



DAY 6

Section 7: PostgreSQL กับ Docker
Section 8: การใช้งาน PostgreSQL บน Vercel Cloud

Section 7

PostgreSQL กับ Docker

- ติดตั้ง PostgreSQL บน Docker
- การเขียน Dockerfile & Docker compose file สำหรับ postgresql
- การ Build Image และ Container PostgreSQL ไปใช้งาน

Docker Desktop | Update to latest | Search for images, containers, volumes, extensions and more... Ctrl+K | iamsa...

- Containers
- Images
- Volumes
- Dev Environments BETA
- Docker Scout
- Learning center

Extensions

Add Extensions

Containers

Give feedback

Container CPU usage
0.00% / 1000% (10 cores available)

Container memory usage
53.59MB / 15.22GB

Show charts

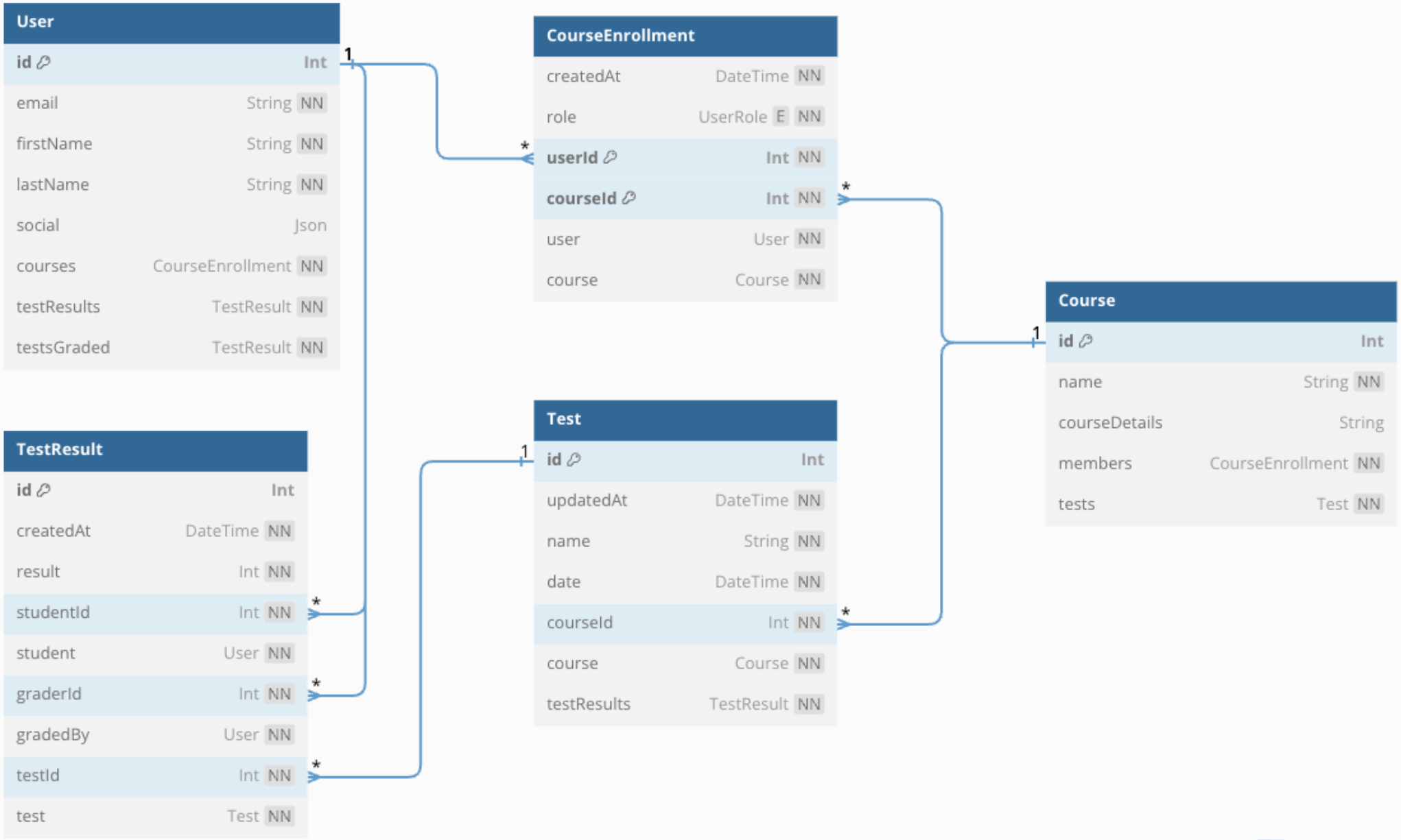
Search

Only show running containers

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
postgres-docker-prisma-typescr		Running (1/1)	0%		1 hour ago	
postgresdb f3e4d2fc02a3	postgres:15.6	Running	0%	6432:5432	1 hour ago	

Showing 2 items

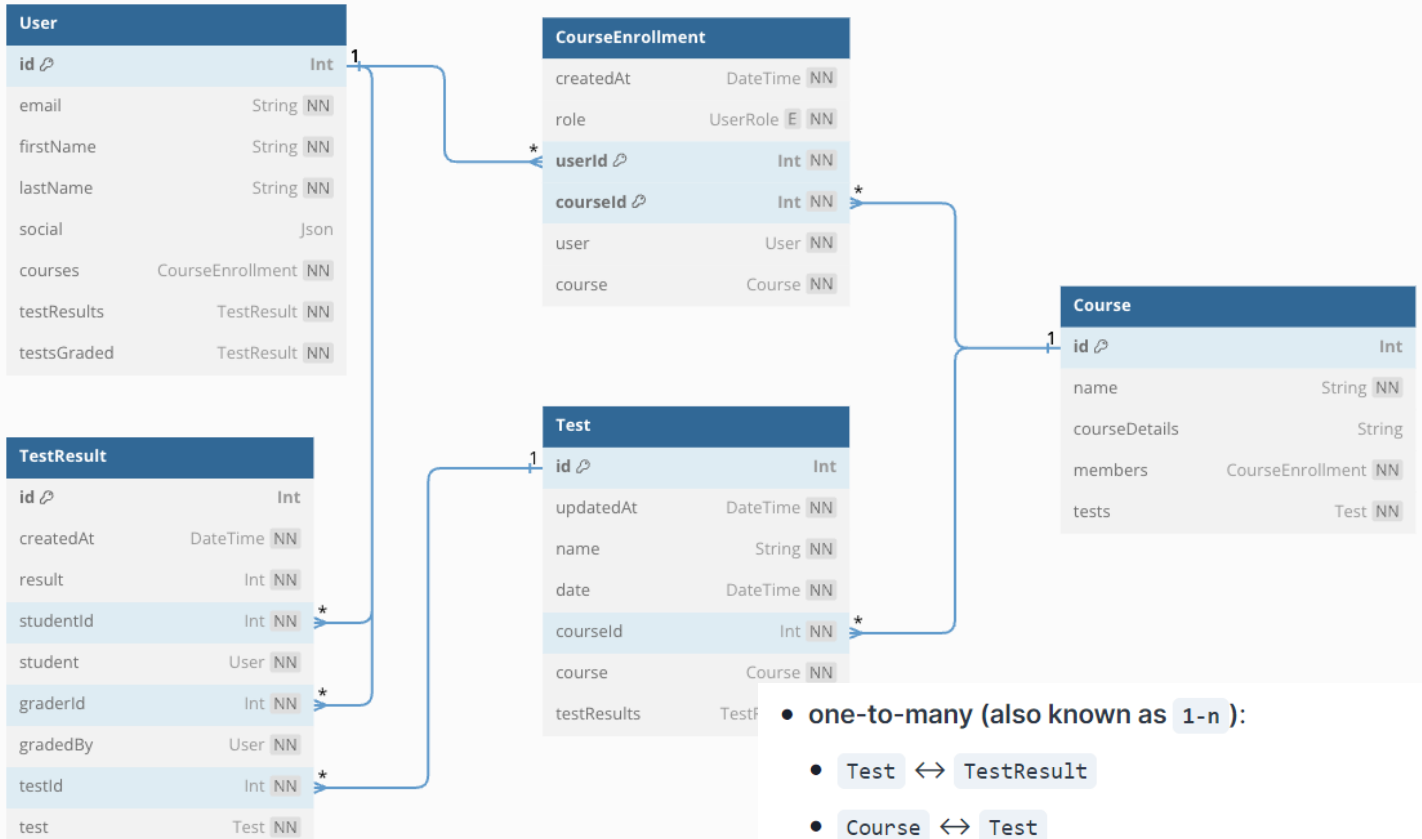
Engine running | RAM 2.66 GB CPU 0.21% | Signed in | v4.25.1 | 1



dbdiagram.io

DATABASE

- PostgresDocker 15.6 (D...
  - prismatypedb
    - public
      - Tables (6)
        - CourseEnrollment
          - columns
            - userId FK integer
            - courseId FK integer
          - index
        - Test
          - columns
            - id integer
            - updatedAt timestamp without tim...
            - name text
            - date timestamp without time zone
            - courseId FK integer
          - index
        - TestResult
          - columns
            - id integer
            - createdAt timestamp without time ...
            - result integer
            - studentId FK integer
            - graderId FK integer
            - testId FK integer
          - index
        - User
          - columns
            - id integer
            - email text
            - firstName text
            - lastName text
            - social jsonb
          - index



- one-to-many (also known as `1-n`):
  - `Test` ↔ `TestResult`
  - `Course` ↔ `Test`
  - `User` ↔ `TestResult` (via `graderId`)
  - `User` ↔ `TestResult` (via `student`)
- many-to-many (also known as `m-n`):
  - `User` ↔ `Course` (via the `CourseEnrollment` relation table with two *foreign keys*: `userId` and `courseId`). Many-to-many relations typically require an additional table. This is necessary so that the grading system can have the following properties:
    - A single course can have many associated users (as students or teachers)
    - A single user can be associated with many courses.

Section 8

การใช้งาน PostgreSQL บน Vercel Cloud

- Vercel Cloud คืออะไร
- สมัครสมาชิกใช้งาน Vercel Cloud
- สร้างฐานข้อมูล PostgreSQL บน Vercel
- เชื่อมต่อกับฐานข้อมูลบน Vercel

laravel-store-api – Postgres – V

vercel.com/samit-koyoms-projects/laravel-store-api/stores/postgres/store_6UEfM2L0ZXLH7QJy/data

Samit Koyom's projects Hobby / laravel-store-api

Feedback Changelog Help Docs

Project Deployments Analytics Speed Insights Logs Observability Firewall Storage Settings

laravelstore

← All Databases / Postgres Database Beta

Status	Created	Region	Endpoint
Available	6mo ago	Singapore sin1	ep-still-credit-a1tcxfc1-pooler

Storage Size	Compute Time	Data Transfer	Written Data
32 MB/256 MB	0.09 hrs/60 hrs	0 B	102 kB

Quickstart

psql @vercel/postgres-kysely drizzle-orm/vercel-postgres @vercel/postgres Prisma pg .env.local Show secret Copy Snippet

```
1 psql "postgres://default:************@ep-still-credit-a1tcxfc1.ap-southeast-1.aws.neon.tech:5432/verceldb?sslmode=require"
```

Data

Getting Started

Projects

Data

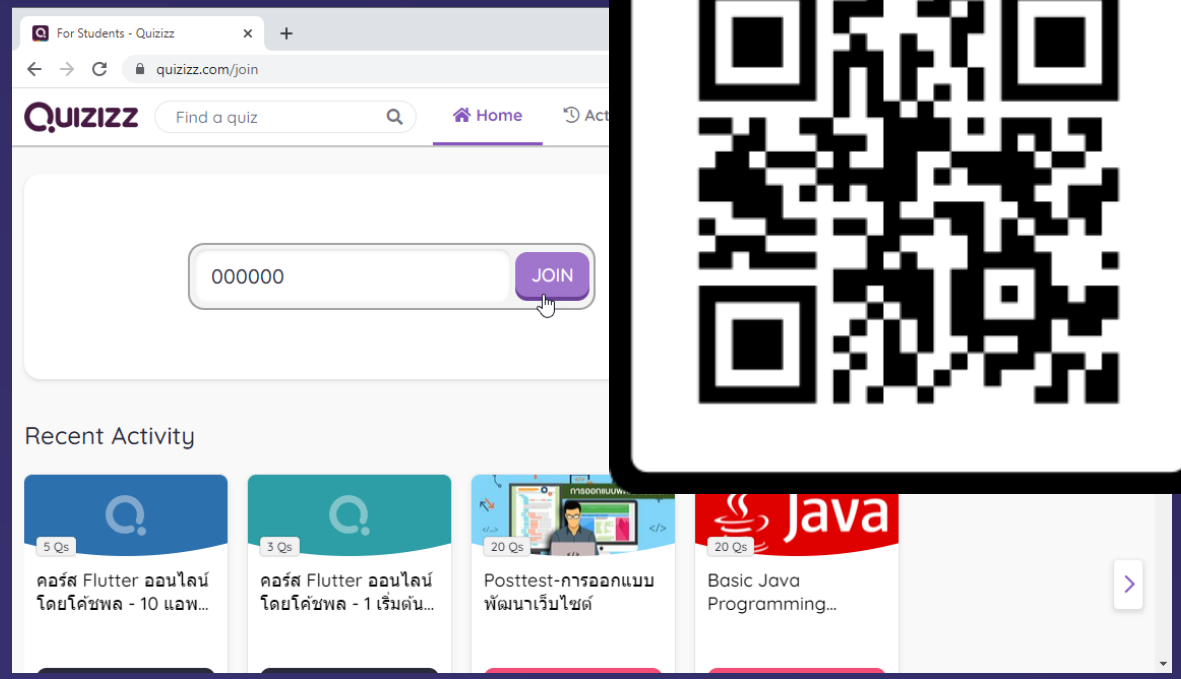
Browse the tables in this database and run SQL queries against the data.

แบบทดสอบหลังอบรม

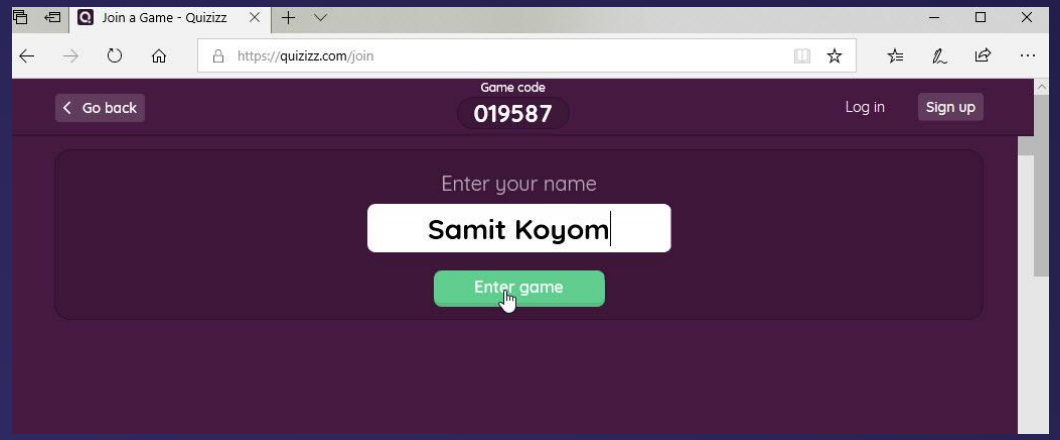
Pretest ทำแบบทดสอบก่อนเรียน

STEP 1: เข้าทำแบบทดสอบที่ลิงก์ ป้อนรหัสเข้าห้องสอบ

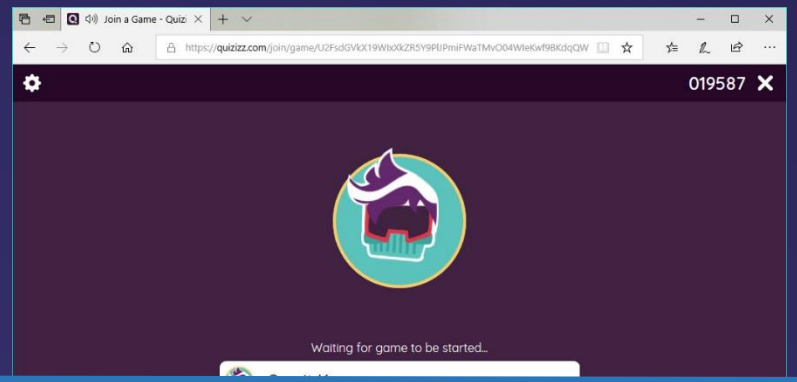
quizizz.com/join



STEP 2: ป้อนชื่อ



STEP 3: รอผู้สอน Start ข้อสอบ



อาจารย์สามิตร โกยม

samitkoyom@gmail.com

iamsamit

faecebook.com/iamsamit

twitter.com/iamsamit

github.com/iamsamitdev

linkedin.com/in/samit-koyom-72173348