Back-End Development

PostgreSQL

Exploration

Purwadhika
Startup and Coding School

Database

- A **database** is *an organized collection of data*.
- The **main purpose** of database is to operate large amount of information by storing, retrieving and managing.
- There are many dynamic websites on the world wide web nowadays which are handled through databases. For example, a model to checks the availability of rooms in a hotel. It is an example of dynamic website that uses database.

Purwadhika
Startup and Coding School

RDBMS

- **RDBMS (*Relational Database Management Systems*)** is database relational model based, introduced by E.F. Codd (1970s).
- In RDBMS, data is represented in terms of tuples (rows). It contains number of tables and each table has its own primary key.
- All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL and Microsoft Access are based on RDBMS.

SQL

- **SQL (*Structured Query Language*)** is used to communicate with a database. It's the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.

- Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc.

- The standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

PostgreSQL

PostgreSQL, often simply called Postgres, is an object relational database management system (ORDBMS) with an emphasis on extensibility & standards compliance. It's free & open source, released under PostgreSQL lisence.

Postgres has been developed by PostgreSQL Global Development Group since 8th July 1996, written in C.

Purwadhika
Startup and Coding School

PostgreSQL Ranking

4th All DB-engines

341 systems in ranking, March 2018

Rank Mar 2018	Rank Feb 2018	Rank Mar 2017	DBMS	Database Model	Score Mar 2018	Score Feb 2018	Score Mar 2017
1.	1.	1.	Oracle	Relational DBMS	1289.61	-13.67	-109.89
2.	2.	2.	MySQL	Relational DBMS	1228.87	-23.60	-147.21
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1104.79	-17.25	-102.70
4.	4.	4.	PostgreSQL	Relational DBMS	399.35	+10.97	+41.71
5.	5.	5.	MongoDB	Document store	340.52	+4.10	+13.59
6.	6.	6.	DB2	Relational DBMS	186.66	-3.31	+1.75
7.	7.	7.	Microsoft Access	Relational DBMS	131.95	+1.88	-0.99
8.	8.	↑ 10.	Redis	Key-value store	131.22	+4.21	+18.22
9.	9.	↑ 11.	Elasticsearch	Search engine	128.54	+3.23	+22.32
10.	10.	↓ 8.	Cassandra	Wide column store	123.49	+0.71	-5.70

https://db-engines.com/en/ranking

Purwadhika
Startup and Coding School

PostgreSQL Ranking

4th RDBMS DB-engines

137 systems in ranking, March 2018

Rank			DBMS	Database Model	Score		
Mar 2018	Feb 2018	Mar 2017			Mar 2018	Feb 2018	Mar 2017
1.	1.	1.	Oracle	Relational DBMS	1289.61	-13.67	-109.89
2.	2.	2.	MySQL	Relational DBMS	1228.87	-23.60	-147.21
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1104.79	-17.25	-102.70
4.	4.	4.	PostgreSQL	Relational DBMS	399.35	+10.97	+41.71
5.	5.	5.	DB2	Relational DBMS	186.66	-3.31	+1.75
6.	6.	6.	Microsoft Access	Relational DBMS	131.95	+1.88	-0.99
7.	7.	7.	SQLite	Relational DBMS	114.81	-2.46	-1.37
8.	8.	8.	Teradata	Relational DBMS	72.46	-0.53	-1.07
9.	↑ 10.	↑ 12.	MariaDB	Relational DBMS	63.10	+1.45	+16.22
10.	↓ 9.	↓ 9.	SAP Adaptive Server	Relational DBMS	62.62	-0.87	-7.51

https://db-engines.com/en/ranking

Purwadhika
Startup and Coding School

PostgreSQL

Getting Started

Download & install PostgreSQL

PostgreSQL Installer

www.postgresql.org/downloads/

Purwadhika
Startup and Coding School

Getting Started

Set as super admin, so we can contacting the server easily without any password:

- Go to `C:\Program Files\PostgreSQL\10\data`
- Open "*pg_hba.conf*" then edit its content.
- Change "md5" to "trust" & save:

```
host all all 127.0.0.1/32 trust
host all all ::1/128 trust
host replication all 127.0.0.1/32 trust
host replication all ::1/128 trust
```

Purwadhika
Startup and Coding School

Activate Server

- Open terminal/command prompt

```
$ cd C:\Program Files\PostgreSQL\10\bin
$ psql -U postgres
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\usr>cd C:\Program Files\PostgreSQL\10\bin

C:\Program Files\PostgreSQL\10\bin>psql -U postgres
psql (10.1)
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.
Type "help" for help.

postgres=# \l
```

Basic Command

Keluar dari PostgreSQL:

```
$ \q
```

Daftar bantuan (help):

```
$ \h
```

Daftar perintah (command):

```
$ \?
```

Menampilkan daftar database:

```
$ \l
```

Purwadhika
Startup and Coding School

How to Work With PostgreSQL Database

Purwadhika
Startup and Coding School

Database

Menampilkan daftar database:

```
$ \l
```

Membuat database “karyawan”:

```
$ CREATE DATABASE karyawan;
```

Hapus database “karyawan”:

```
$ DROP DATABASE karyawan;
```

Terhubung ke database “karyawan”:

```
$ \c karyawan
```

Purwadhika
Startup and Coding School

Table

Membuat table “staf_IT”:

```
$ CREATE TABLE staf_IT
- (id INTEGER,
- nama VARCHAR (255),
- usia INTEGER);
```

Menampilkan daftar table:

```
$ \d
```

Lihat struktur table “staf_IT”:

```
$ \d staf_IT
```

Hapus table “staf_IT”:

```
$ DROP TABLE staf_IT;
```

Purwadhika
Startup and Coding School

Add Data Record

Insert data ke table “staf_IT”:

```
$ INSERT INTO staf_IT
- VALUES (1, 'Andi', 24);

$ INSERT INTO staf_IT
- (id, nama, usia) VALUES
- (2, 'Budi', 36);

$ INSERT INTO staf_IT
- (usia, nama, id) VALUES
- (28, 'Caca', 3);
```

Purwadhika
Startup and Coding School

Multiple Data Records

Insert multiple data ke table “staf_IT”:

```
$ INSERT INTO staf_IT VALUES
- (4, 'Dedi', 42),
- (5, 'Euis', 21),
- (6, 'Fafa', 28),
- (7, 'Gilang', 30),
- (8, 'Hani', 27),
- (9, 'Iwan', 29),
- (10, 'Janni', 28);
```

Melihat data record di table “staf_IT”:

```
$ SELECT * FROM staf_IT;
```

Purwadhika
Startup and Coding School

PostgreSQL

Multiple Data Records

```
INSERT 0 10
karyawan=# select * from staf_it;
 id |  nama   | usia
----+---------+------
  1 | Andi    |   24
  2 | Budi    |   36
  3 | Caca    |   28
  4 | Dedi    |   42
  5 | Euis    |   21
  6 | Fafa    |   28
  7 | Gilang  |   30
  8 | Hani    |   27
  9 | Iwan    |   29
 10 | Janni   |   28
(10 rows)
```

Purwadhika
Startup and Coding School

PostgreSQL

Select

Menampilkan semua data dari tabel “staf_IT”:

```
$ SELECT * FROM staf_it;
```

Menampilkan data nama & usia dari “staf_IT”:

```
$ SELECT nama, usia FROM staf_it;
```

Menampilkan hanya data nama dari “staf_IT”:

```
$ SELECT nama FROM staf_it;
```

Purwadhika
Startup and Coding School

PostgreSQL

Where

Menampilkan semua data id, nama & usia dari tabel “staf_IT”, yang memiliki nilai usia = 28:

```
$ SELECT * FROM staf_IT
- WHERE usia = 28;
```

Menampilkan semua data id, nama & usia dari “staf_IT” , yang memiliki nomor id genap:

```
$ SELECT * FROM staf_IT
- WHERE id % 2 = 0;
```

Purwadhika
Startup and Coding School

PostgreSQL

And & Or

Menampilkan semua data id, nama & usia dari tabel “staf_IT”, yang usia=28 atau nama=Andi:

```
$ SELECT * FROM staf_it
- WHERE usia = 28 OR
- nama = 'Andi';
```

Menampilkan semua data id, nama & usia dari “staf_IT” , yang memiliki usia antara 24-30:

```
$ SELECT * FROM staf_it
- WHERE usia > 24 AND
- usia < 30;
```

Purwadhika
Startup and Coding School

PostgreSQL

Order

```
$ SELECT * FROM staf_it WHERE
- usia < 29 ORDER BY usia;

$ SELECT * FROM staf_it WHERE
- usia < 29 ORDER BY usia ASC;

$ SELECT * FROM staf_it WHERE
- usia < 29 ORDER BY usia DESC;

$ SELECT * FROM staf_it WHERE
- usia < 29 ORDER BY nama, usia;
```

Purwadhika
Startup and Coding School

Update

Update semua data usia di tabel “staf_IT”:

```
$ UPDATE staf_it
- SET usia = 26;
```

Update data tertentu di tabel “staf_IT”:

```
$ UPDATE staf_it
- SET usia = 32
- WHERE nama = 'Andi';
```

Purwadhika
Startup and Coding School

PostgreSQL

Delete

Hapus semua data di tabel “staf_IT”:

```
$ DELETE FROM staf_it;
```

Hapus data tertentu di tabel “staf_IT”:

```
$ DELETE FROM staf_it
- WHERE nama = 'Andi';
```

Purwadhika
Startup and Coding School

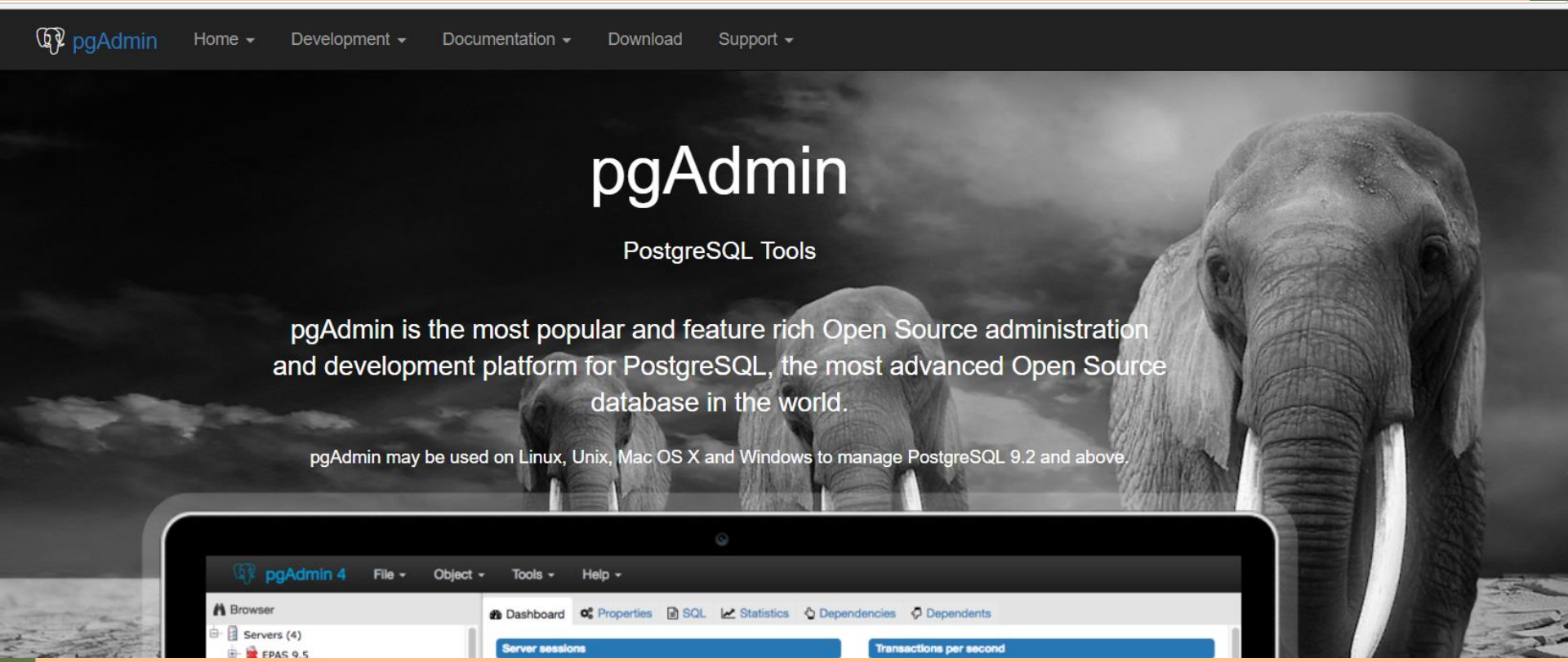
How to Work With PostgreSQL GUI Tools

Purwadhika
Startup and Coding School

PostgreSQL

Working with GUI

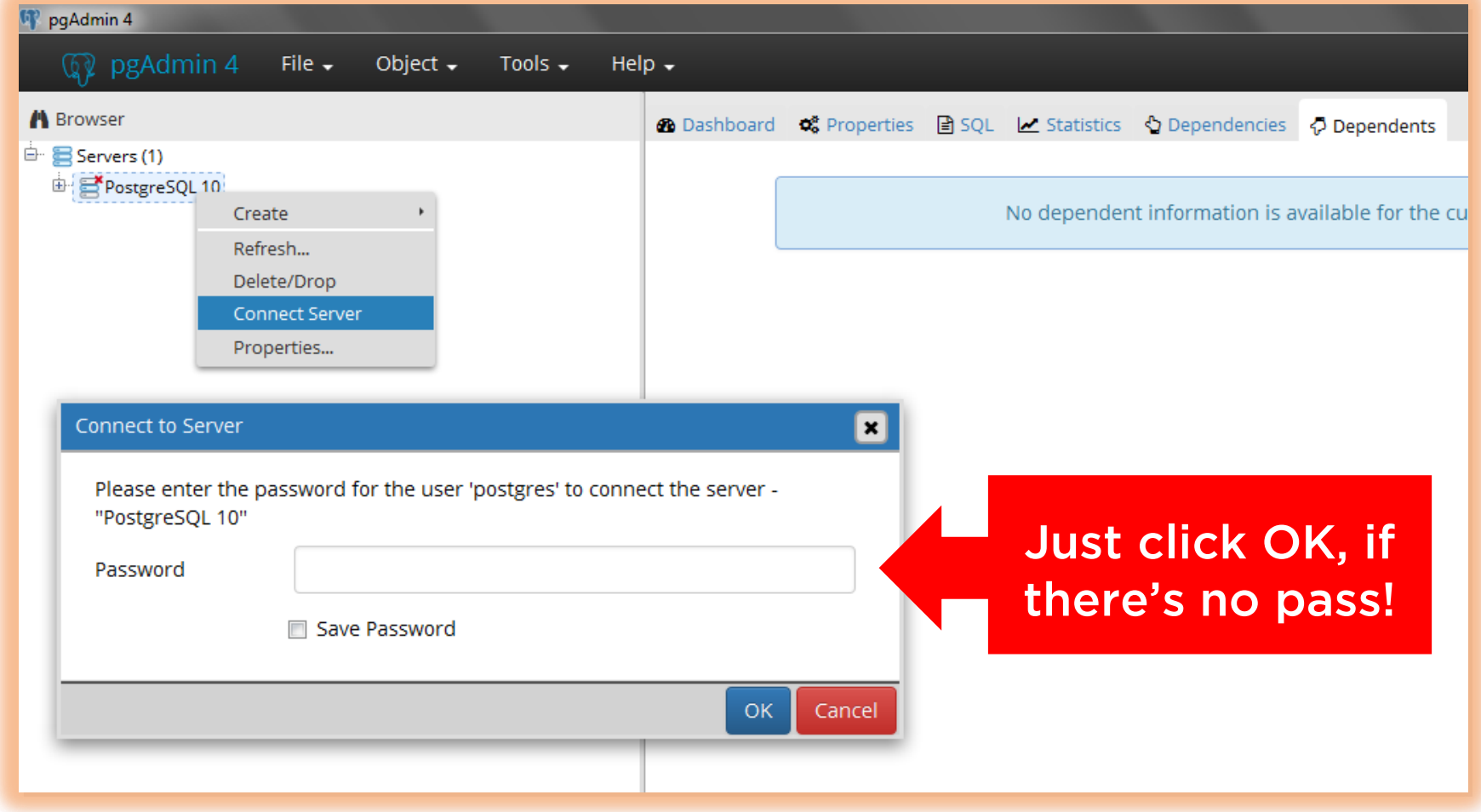
#1 Installing pgAdmin



Purwadhika
Startup and Coding School

Working with GUI

#2 Connecting Server

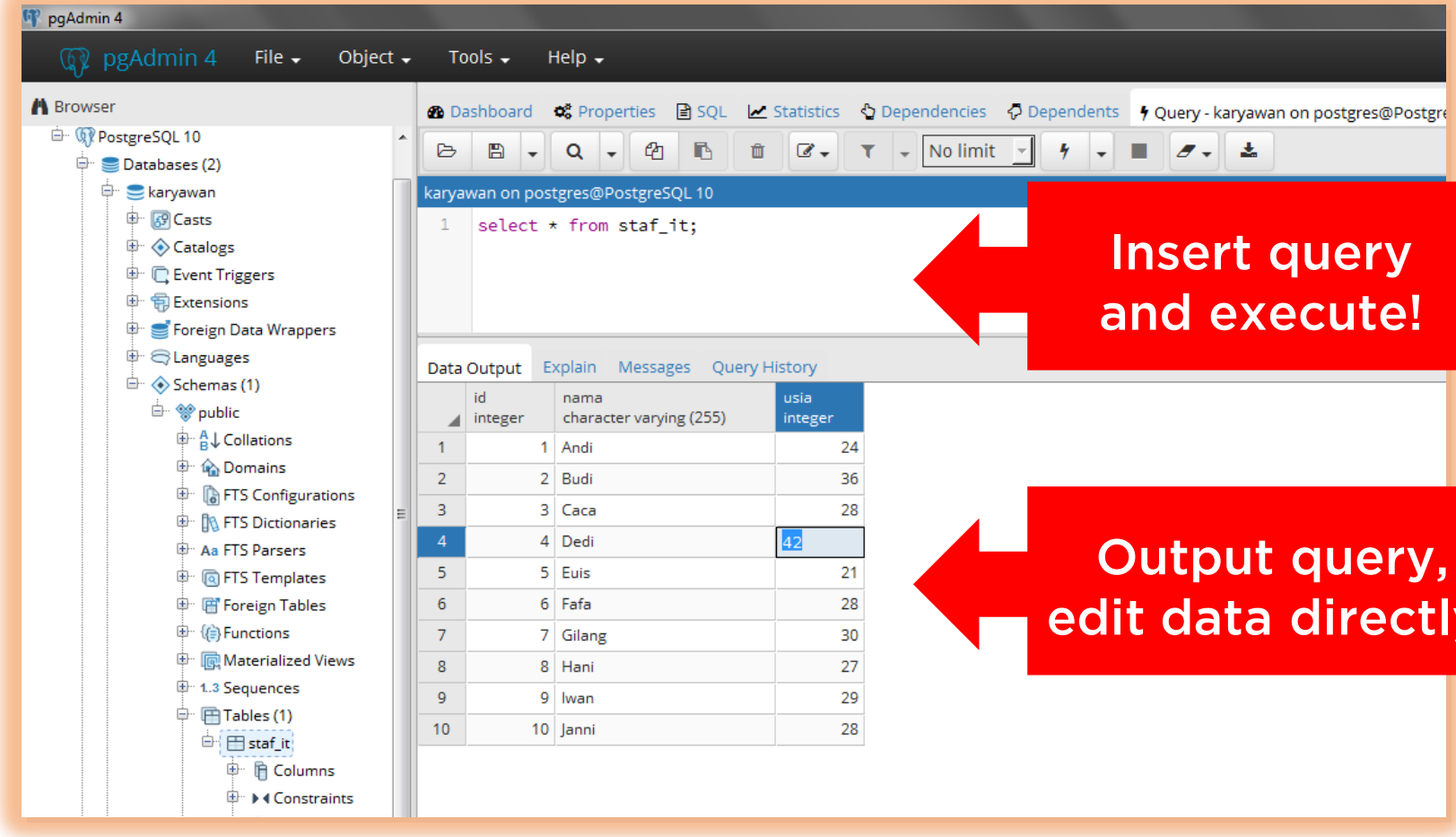


***Make sure server is activated first!**

Purwadhika
Startup and Coding School

Working with GUI

#3 Tools - Query Tool



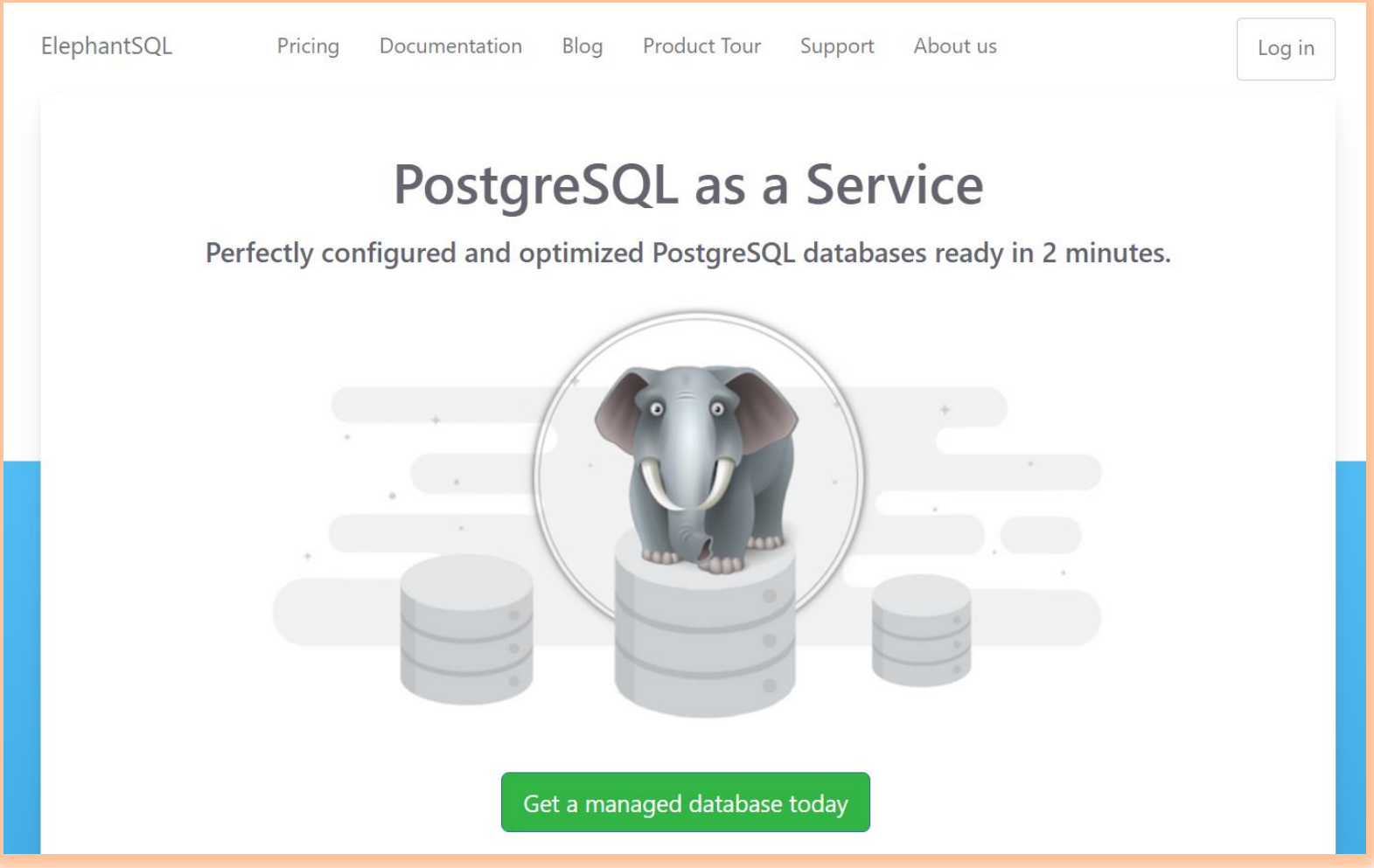
Purwadhika
Startup and Coding School

How to Work With PostgreSQL Hosting

Purwadhika
Startup and Coding School

MySQL

ElephantSQL



elephantsql.com/

Purwadhika
Startup and Coding School

MySQL

ElephantSQL

elephantsql.com/

Purwadhika
Startup and Coding School

Back-End Development

PostgreSQL

Exploration

Purwadhika
Startup and Coding School