

# 5.1 关系型数据库

---

- 数据库基本概念
- PostgreSQL 的安装
- PostgreSQL 的基本操作



# 目录

1

数据库基本概念

2

PostgreSQL 的  
安装

3

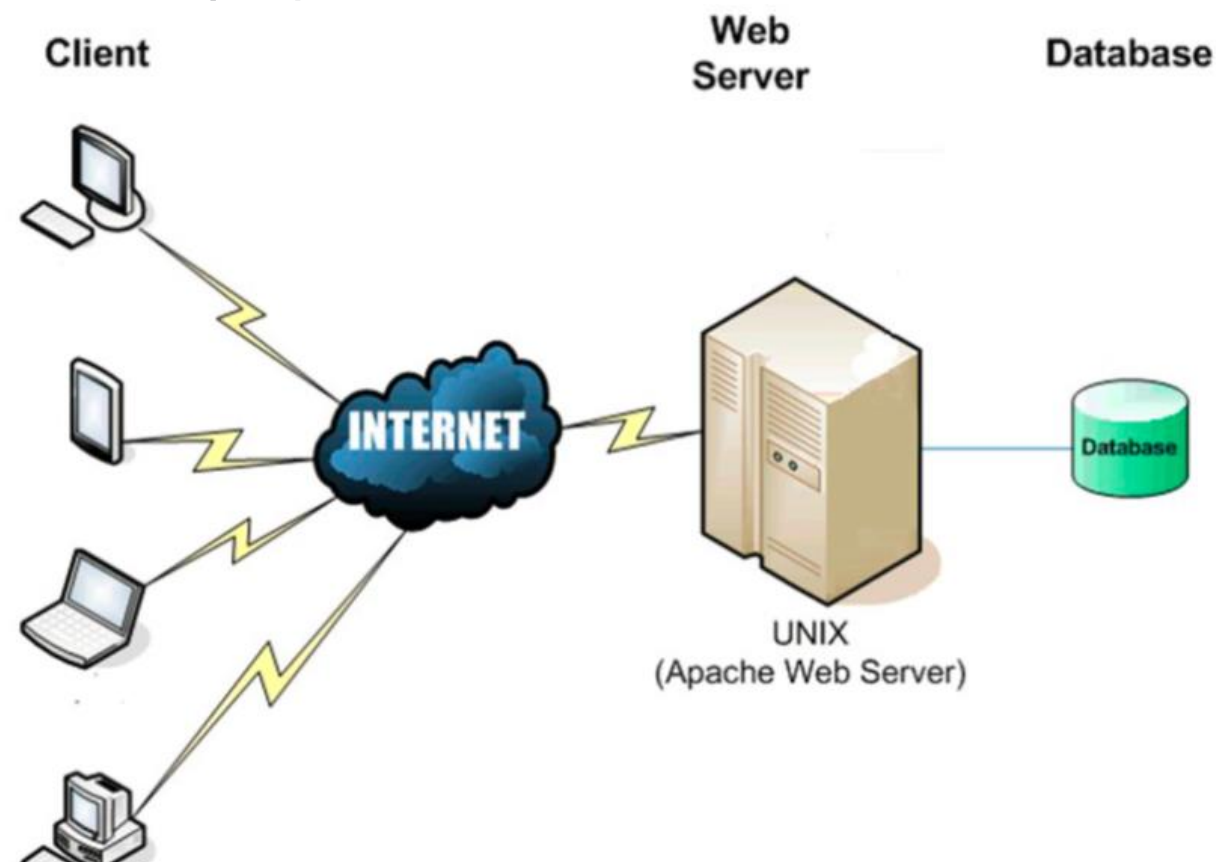
PostgreSQL 的  
基本操作

# 数据库概述

- **数据库**[データベース] (Database, DB) 是按照一定的数据结构来组织、存储、共享和管理数据的仓库。
- 特征：
  - 数据按一定的数据模型组织、描述和存储；
  - 可为各种用户共享；
  - 冗余度较小；
  - 数据独立性较高；
  - 易扩展；
  - 不同数据库可由统一的方式 (**SQL 语言**) 操作。

# 数据库在网站中的地位

- 要创建一个网站，至少需要以下部分：
  - 使用 HTML/CSS 及 JavaScript 创建网页页面。
  - 关系型数据库管理系统（Relational Database Management System, **RDBMS**），如 PostgreSQL、SQL Server、MySQL 等。
  - 使用 SQL 语言对 RDBMS 进行访问和操作，获取所需数据。
  - 网络服务器控制各个部分。

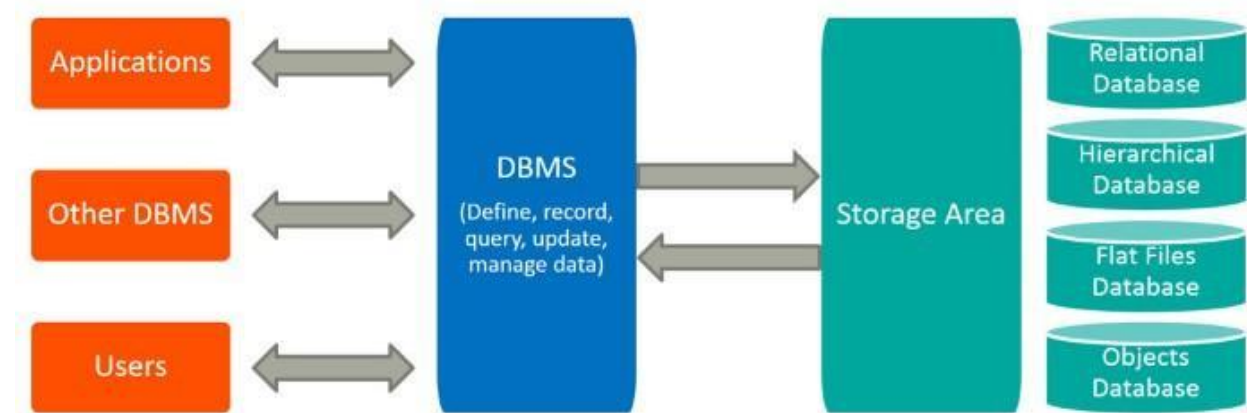




# 数据库管理系统

- **数据库管理系统** [データベース管理システム] (Database Management System, DBMS) 是对数据库进行管理的软件。

## Database Management System



- **基本功能**
  - 数据库的**建立与维护**
  - **数据定义**
    - **数据定义语言** [データ定義言語] (Data Definition Language, DDL) : 如创建数据库、删除数据库、创建表、删除表等。
  - **数据操作**
    - **数据操纵语言** [データ操作言語] (Data Manipulation Language, DML) : 如数据的插入、删除、查询、修改等。
  - **数据控制**
    - **数据控制语言** [データ制御言語] (Data Control Language, DCL) : 如分配权限。

# 关系型数据库

- 在关系型数据库中，数据库通常包含一个或多个**表**[テーブル]。每个表均由名称标识（例如“客户”或“订单”）。
- 表包含决定数据类型的**列**[カラム]以及具有数据的**记录**[レコード]（行）。每一行都有列数个**字段**[フィールド]，也就是实际的数据值。
- 每个表还可能包含一列**主键**[主キー]。主键在一个表中不能重复，可以用主键唯一确定 1 行数据。比如一般会把 ID、名字等设为主键。
- 除此之外，关系型数据库中的表还可能定义每一列数据的**类型**[タイプ]、**范围**或其它**约束**条件。

## Example ✓

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

- 上表包含 5 个**记录**（每个客户 1 个）和 7 个**列**（客户 ID、客户名、联络人、地址、城市、邮编和国家）。ID 可以被设为主**键**。

# 关系型数据库的特点

- **关系**[關係]指的就是对表中列及其约束的定义。关系数据库中的表通常具有以下特点：
  1. 关系必须规范化，属性不可分割，表中不能包含表；
  2. 同一表中没有相同的属性名；
  3. 表中不允许有完全相同的记录（冗余信息）；
  4. 同一表中记录的顺序无关紧要；
  5. 同一表中属性的顺序无关紧要；



# 基本数据操作

- 关系型数据库一般支持对数据进行以下基本操作：
  - 插入数据 (Create)：将一行新的数据插入到表中。
  - 删除数据 (Delete)：将一行数据从表中删除。
  - 修改数据 (Update)：修改表中某些字段的值。
  - 查询数据 (Read)：根据一些条件，查询表中的一行或多行数据。
- 这些操作统称为“**增删改查**” (**CRUD**)。
- 同时，我们也可以对多个表进行一些运算，比如求交集、并集、差集或者连接，从而利用其它表的数据进行增删改查操作。

# 查询操作的例子

- 以最常用到的**查询**操作为例，我们可以指定某个条件以查询满足条件的某些行的数据：

id	name	gender
1	Rose	Female
2	Jack	Male

name = "Rose"



id	name	gender
1	Rose	Female

- 也可以只选出表中某些列的值查看：

id	name	gender
1	Rose	Female
2	Jack	Male

name = "Jack"



name, gender

name	gender
Jack	Male

# SQL

- **SQL** 是用于访问和操作数据库的标准语言。
- SQL 于 1986 年成为美国国家标准协会（ANSI）和 1987 年国际标准化组织（ISO）的标准。
- SQL 的主要功能：
  - 创建新的数据库；
  - 在数据库中创建新表；
  - 对表进行增删改查操作；
  - 在数据库中创建存储过程；
  - 在数据库中创建视图；
  - 对表，过程和视图设置权限；
  - .....



# PostgreSQL

- 我们将在接下来的课程中学习如何使用 SQL 语言访问和操作 **PostgreSQL 数据库管理系统**。
- 尽管 SQL 是通用的标准，不同的数据库管理系统都会支持它的核心语言，但**不同的管理系统所使用的语法会稍有不同**。比如，PostgreSQL 提供了很多标准 SQL 中没有的高级数据类型。
- PostgreSQL 是一个免费开源的数据库管理系统，具有大型商业 RDBMS 中所具有的特性，包括事务、子选择、触发器、视图、**外键**[\[外部キ一\]](#)引用完整性和复杂锁定等功能。



Q & A

*Question and answer*

## Coffee ☕ Break

# 非关系型数据库

既然有关系型数据库，自然也存在**非关系型数据库** (NoSQL)。

和传统的关系型数据库不同，NoSQL 中不使用表，而是主要使用**键 - 值存储**的形式。每一个数据类似一个 JavaScript 中的对象，里面可以储存别的数据，如字符串、数组或者其它对象。每一个数据的结构都可能**不相同**，这是传统表结构无法实现的。

比起传统 RDBMS，NoSQL 在**大规模、高流量**的数据库中性能更佳，且更易构建**分布式系统**；但 NoSQL 在并发处理时的**数据一致性**方面不如传统 SQL。



# 目录

1

数据库基本概念

2

PostgreSQL 的  
安装

3

PostgreSQL 的  
基本操作

# PostgreSQL 安装

- 从以下地址下载安装包：

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

- 接下来的教程将分为 macOS 和 Windows 两个版本。已经安装过的同学不用再次安装，但可能需要调整一些配置。

Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
13.2	N/A	N/A	<a href="#">Download</a>	<a href="#">Download</a>	N/A
12.6	N/A	N/A	<a href="#">Download</a>	<a href="#">Download</a>	N/A
11.11	N/A	N/A	<a href="#">Download</a>	<a href="#">Download</a>	N/A
10.16	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
9.6.21	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>

# 第一步

1. 双击下载好的 .dmg 文件:



2. 双击里面的文件:

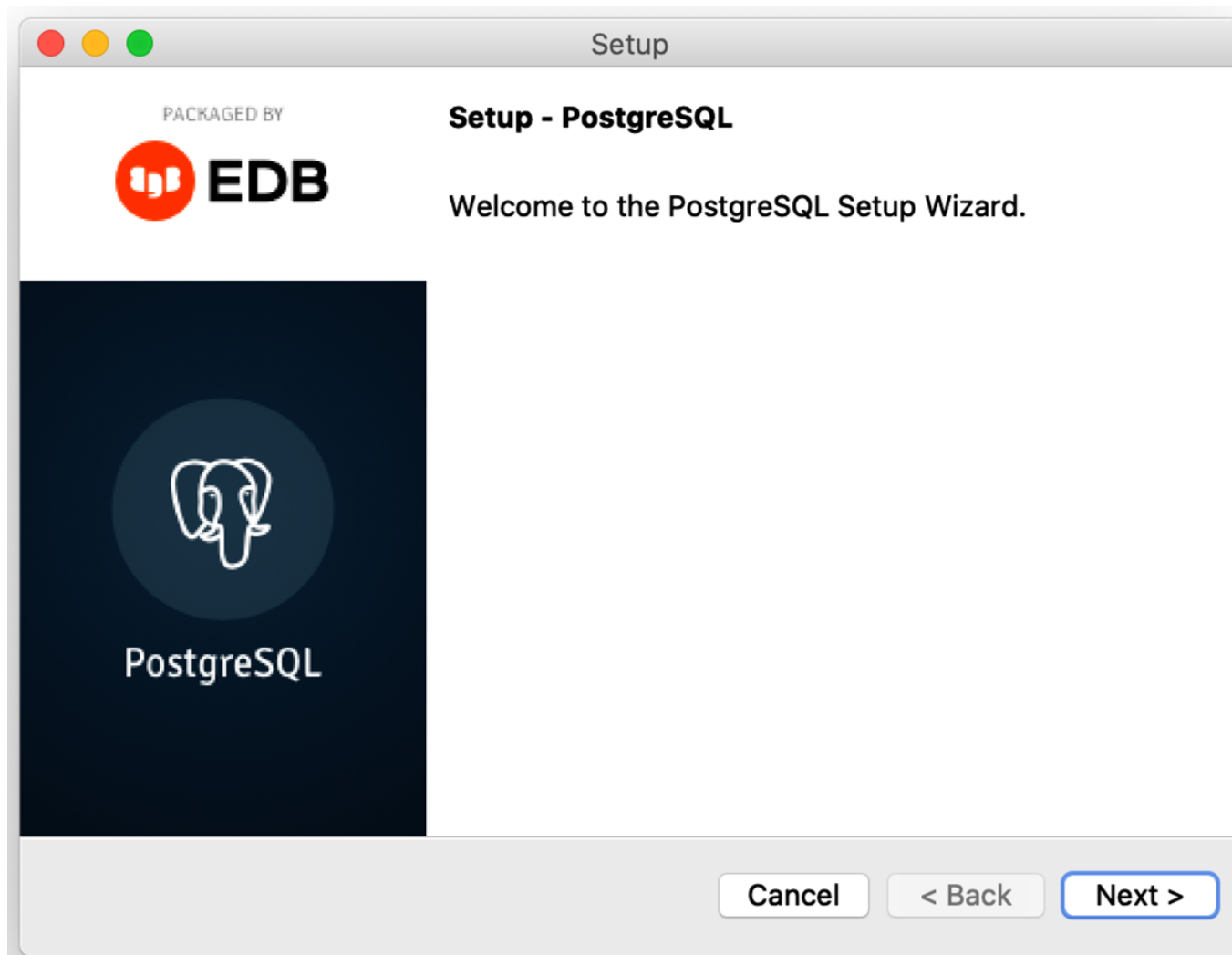


3. 可能出现权限警告, 需要确定并输入开机密码。



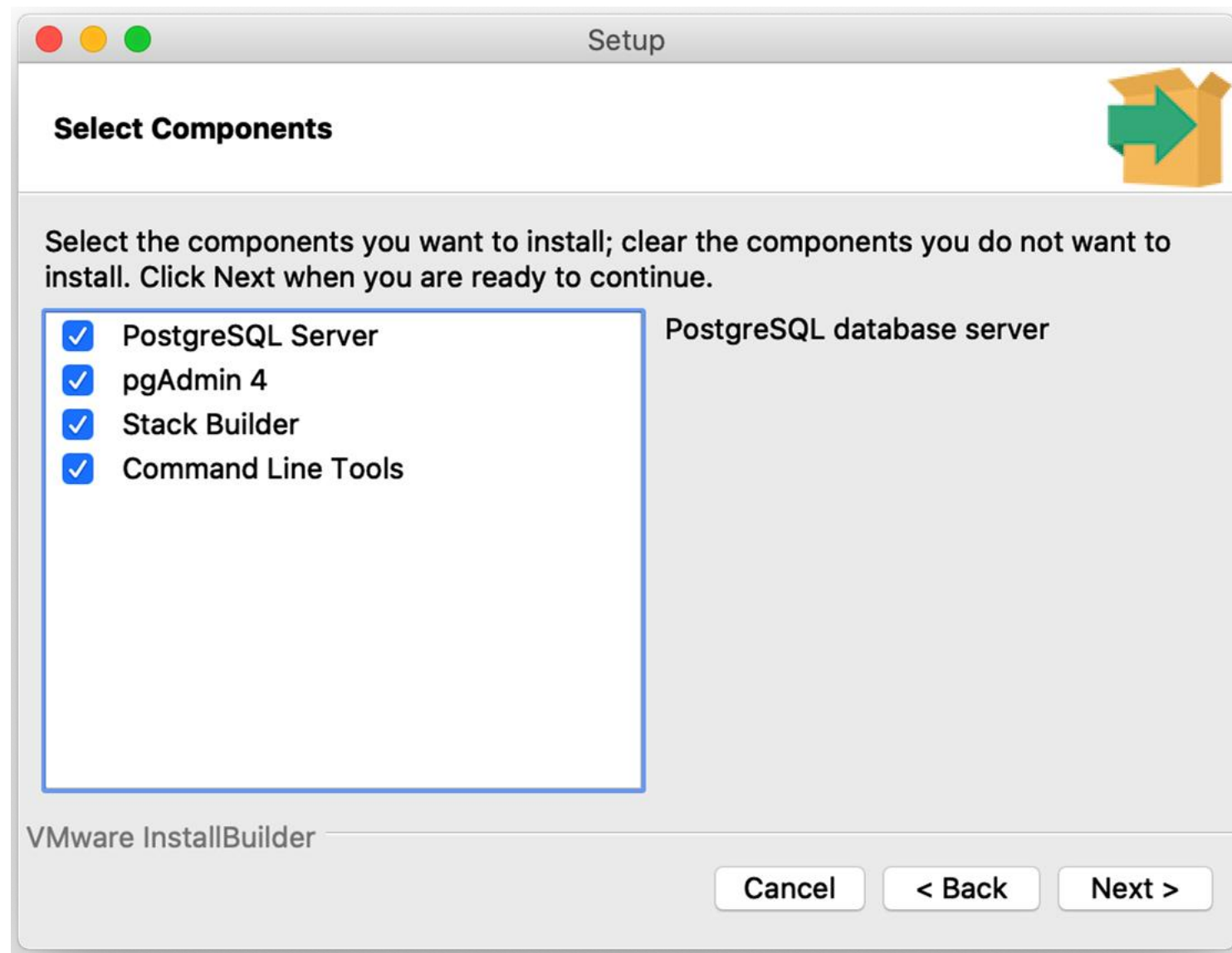
## 第二步

- 点击 Next，直到第三步的画面。
- 保存路径可以自定义。



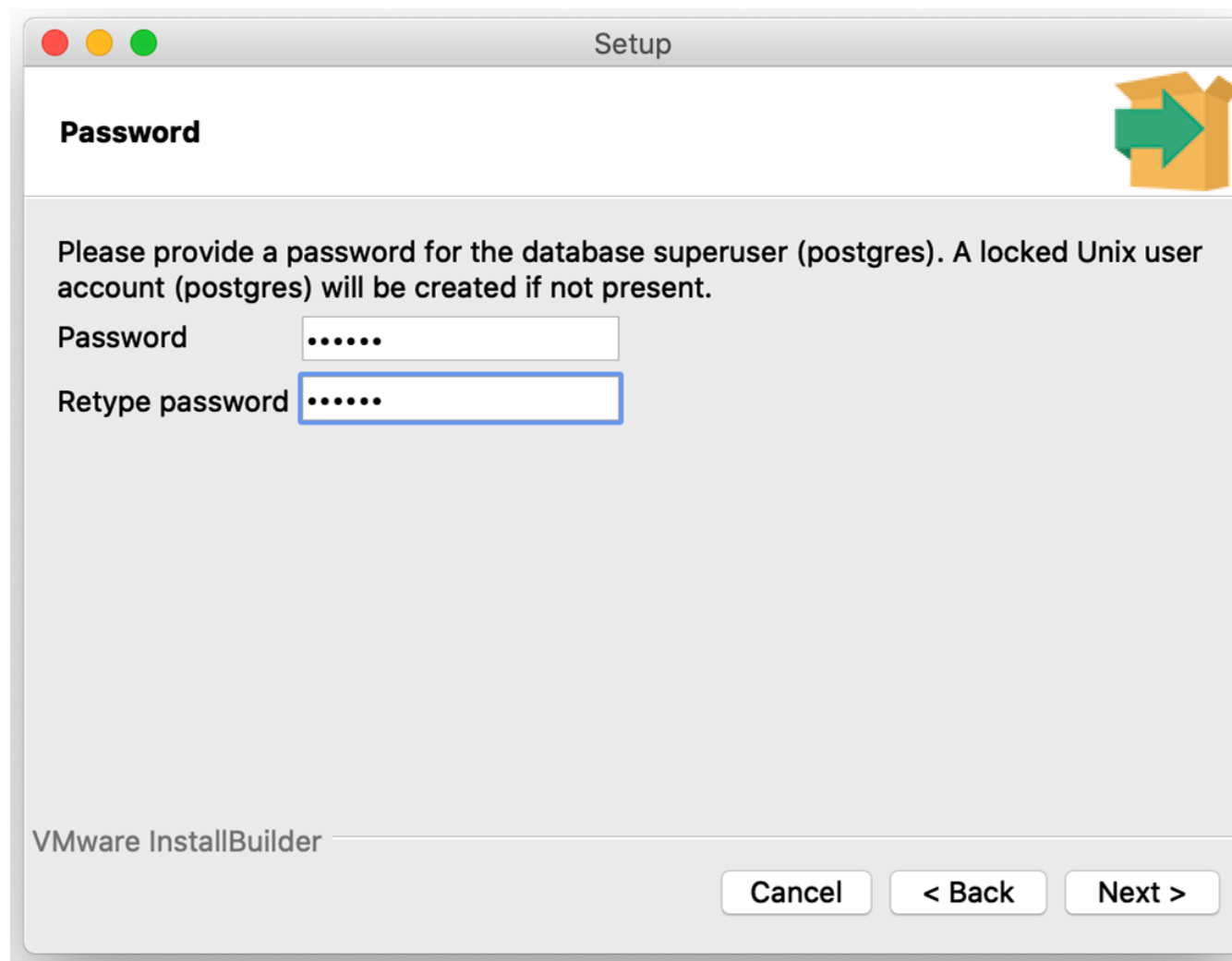
## 第三步

- 4 个组件全选， 点击 Next。



## 第四步

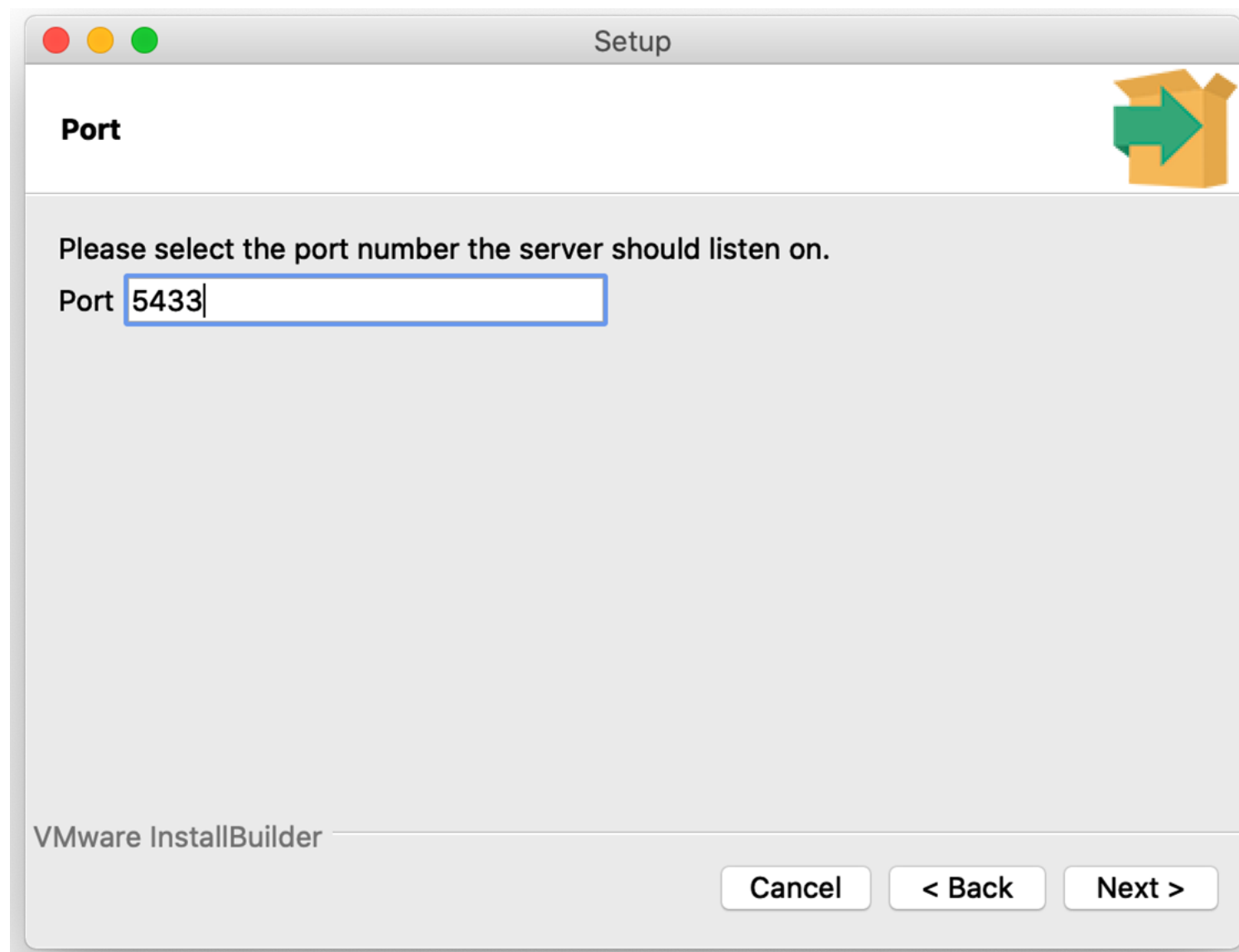
- 这个页面可以设置数据库密码
- 为了方便之后的学习，请大家统一设置为 123456。





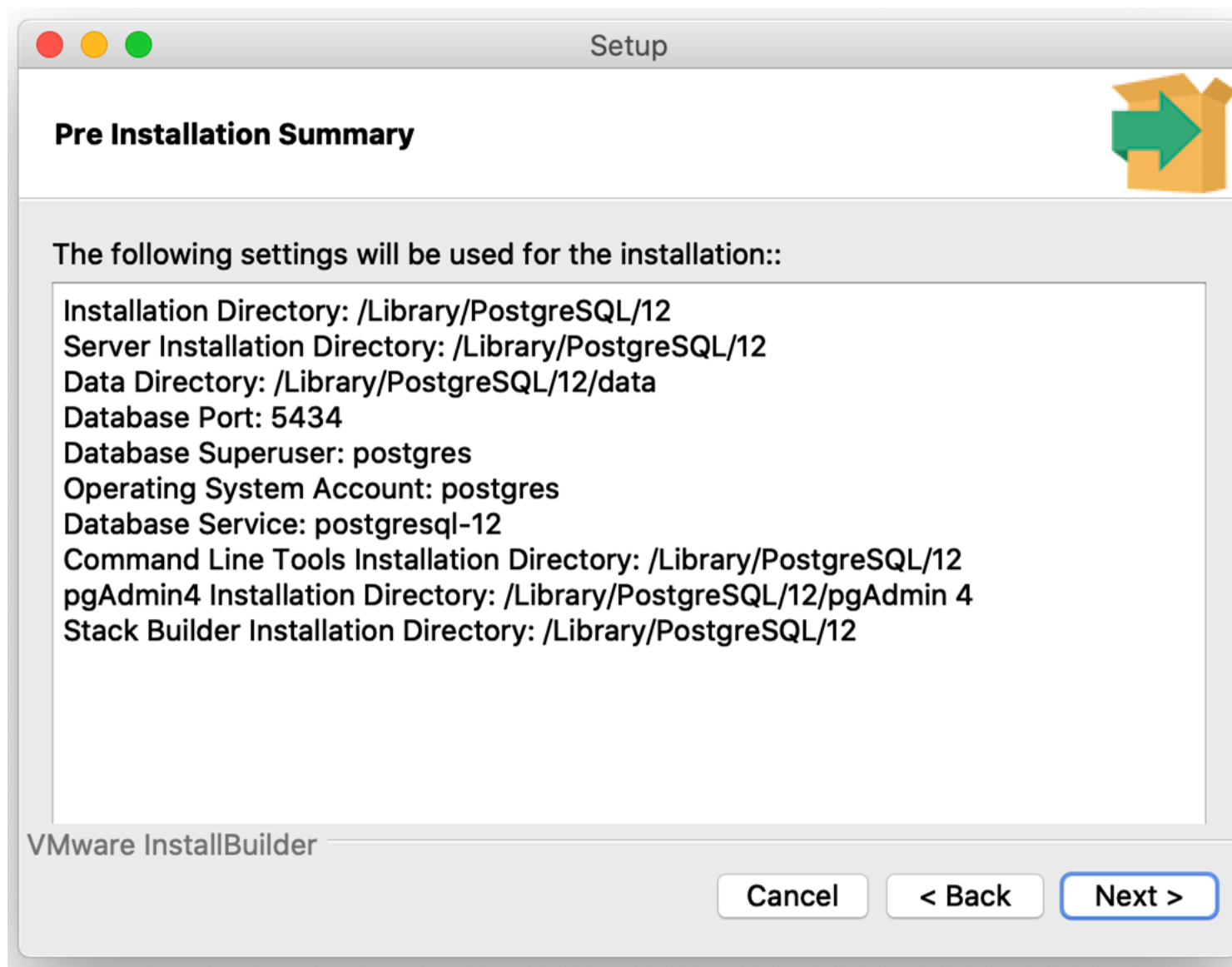
## 第五步

- 这里使用默认的端口号（5433）就可以。如果不行，则使用5434。



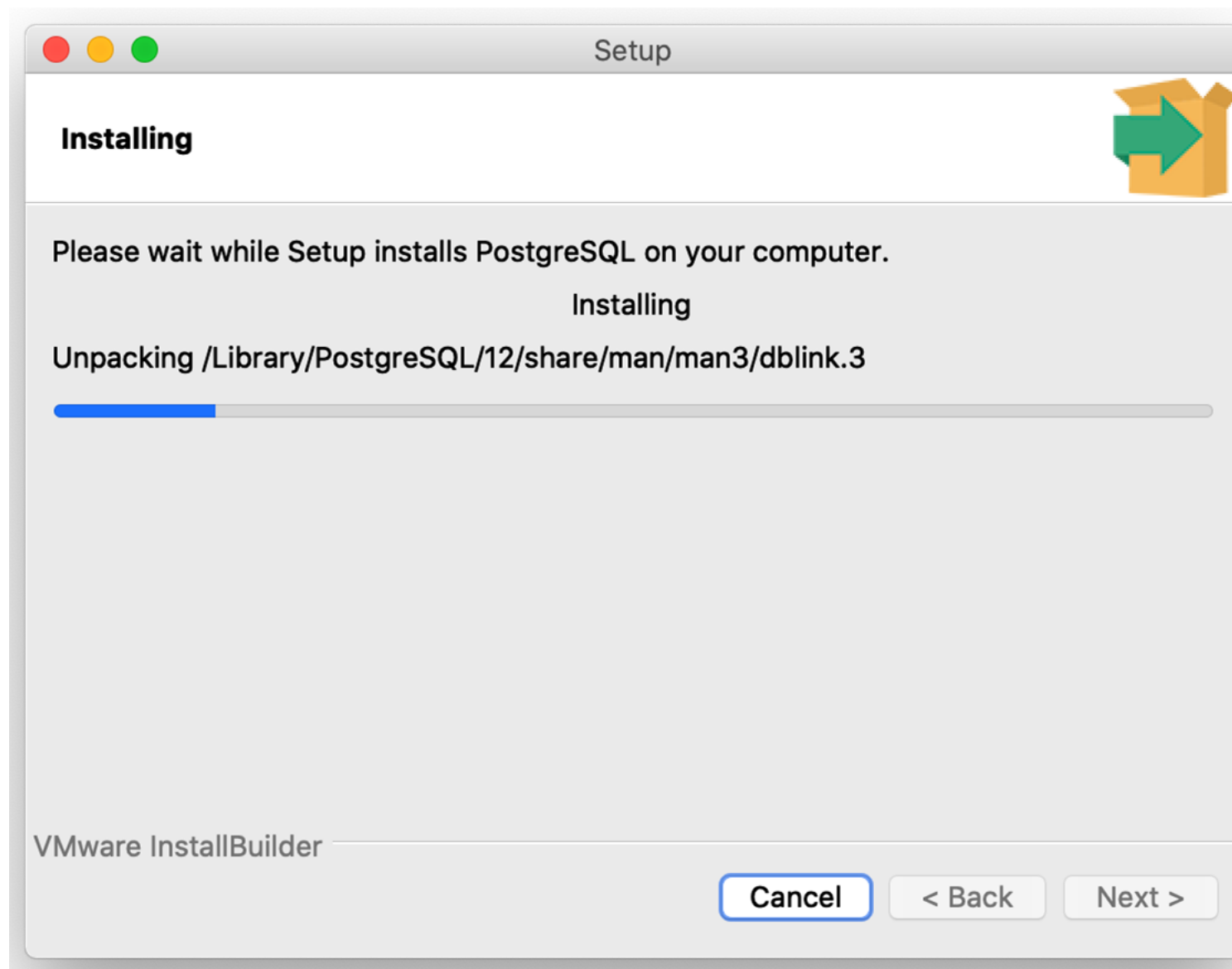
## 第六步

- 请复制或截图保存这个页面的内容:



## 第七步

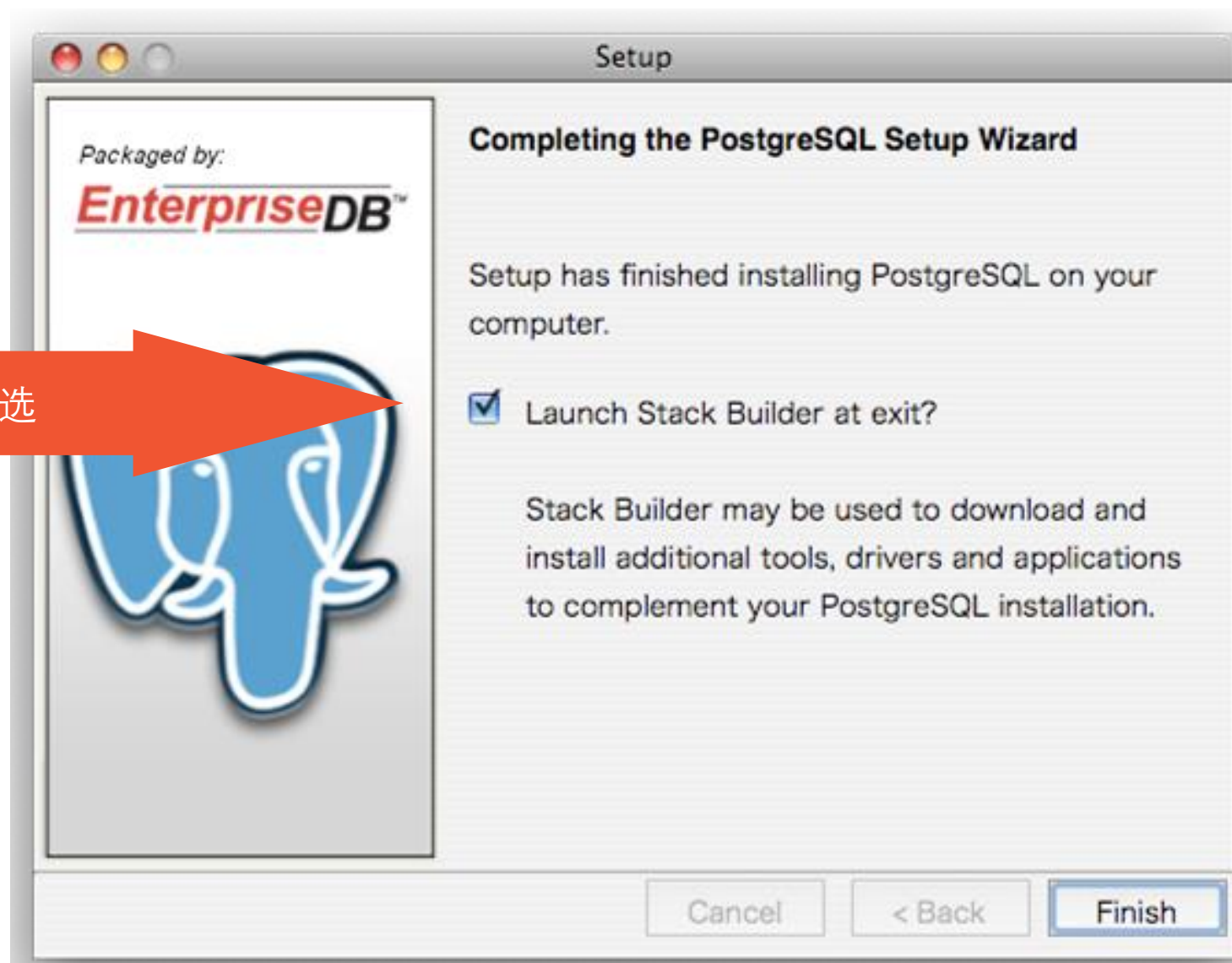
- 一直点击 Next 至到如图所示的界面，等待安装结束。





## 第八步

- 去掉勾选， 点击 Finish 完成安装。



## 确认安装完成

- 安装完成后，应用里面将会多出三个图标（可能不完全一致）：



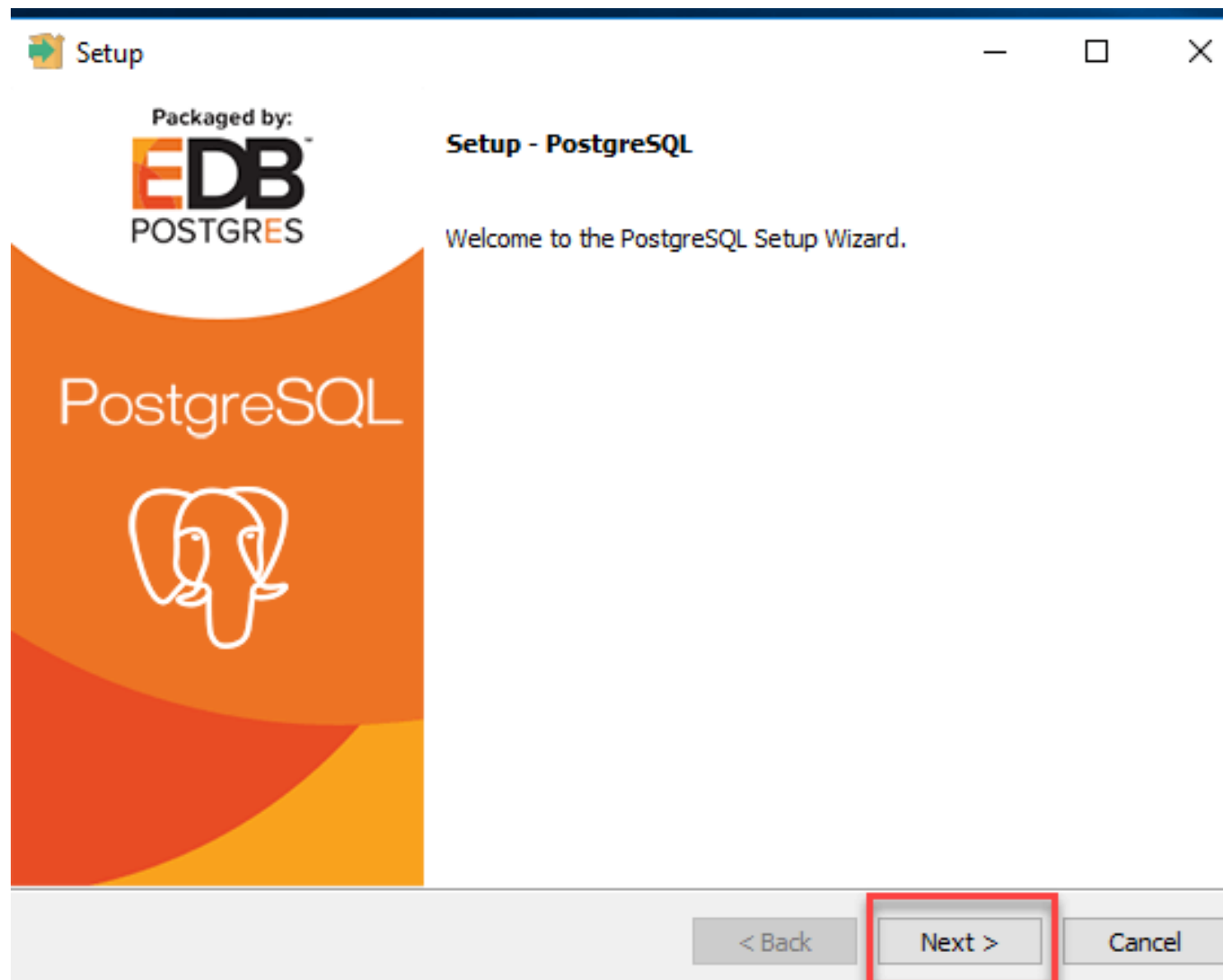
可视化交互界面 (GUI)

说明文档

命令行界面

## 第一步

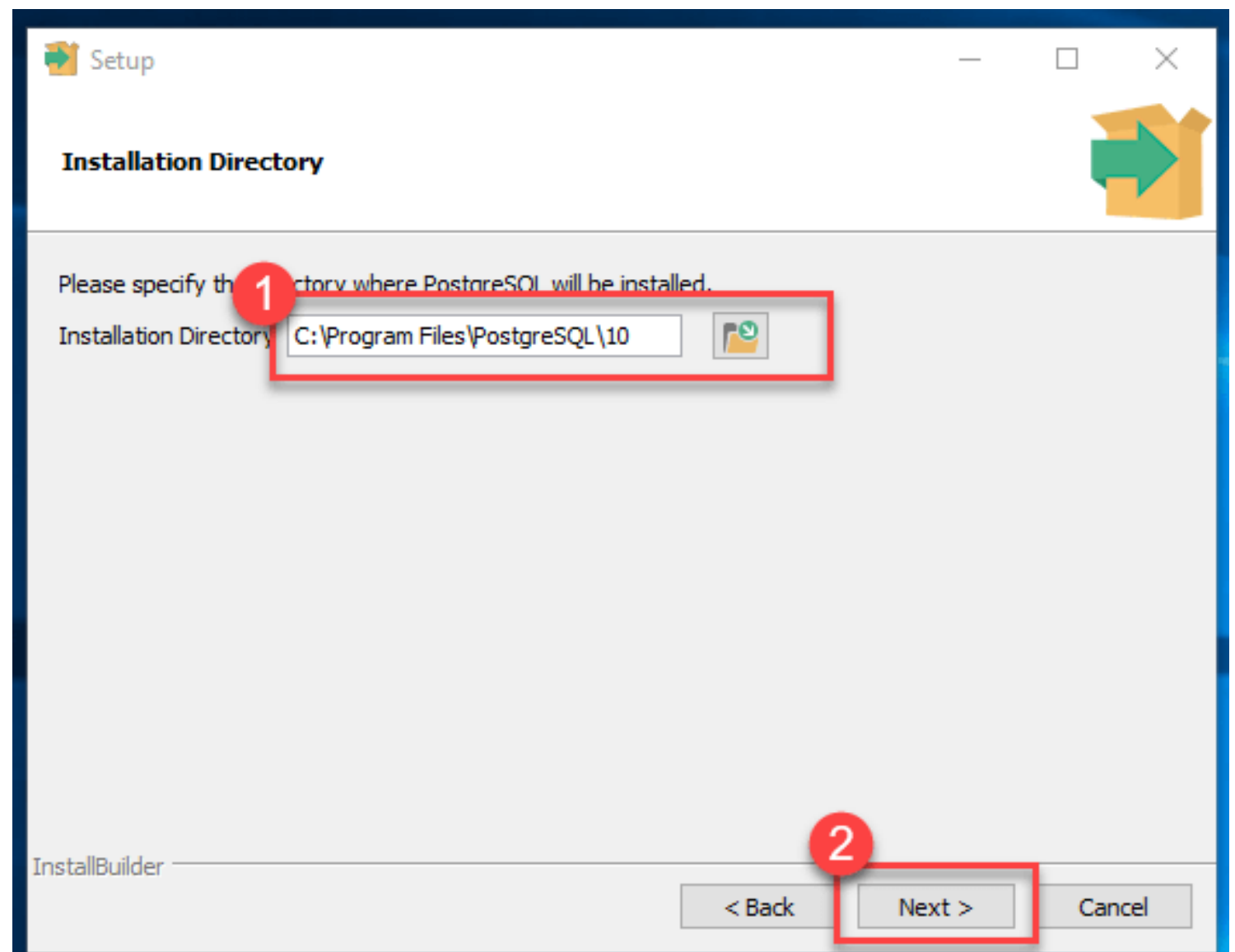
- 双击下载好的安装文件开始安装。





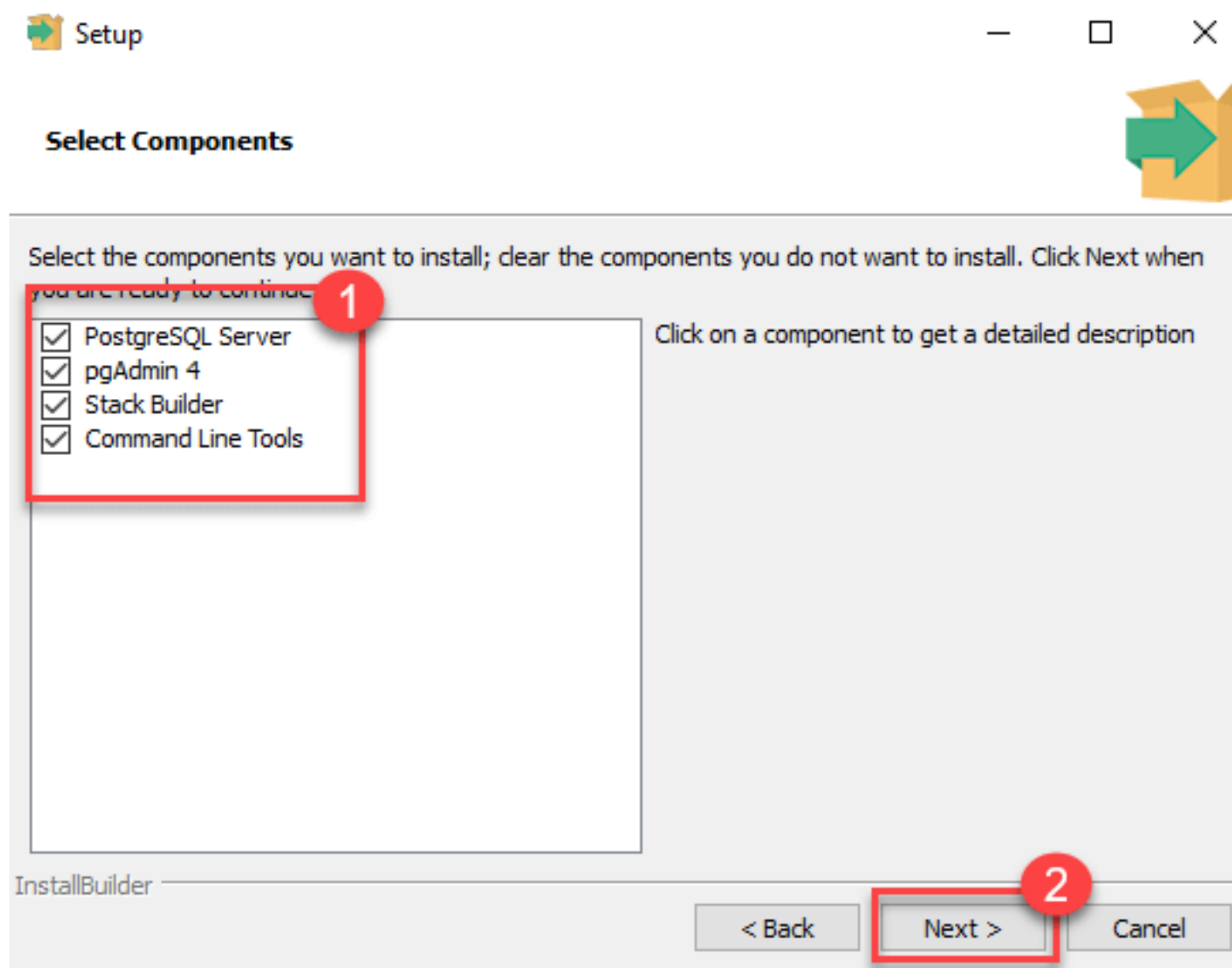
## 第二步

- 点击 Next，直到第三步的画面。
- 保存路径可以自定义。



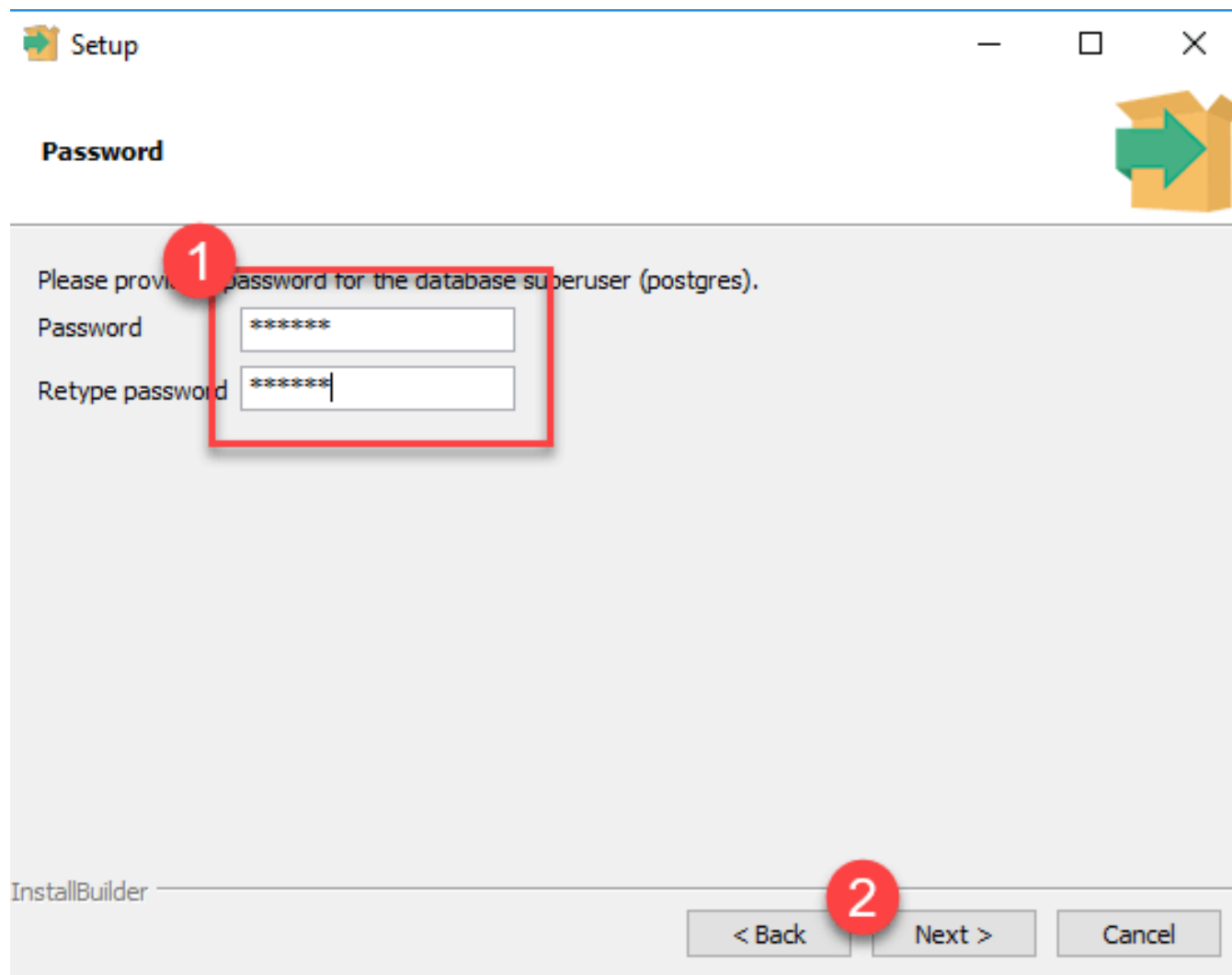
## 第三步

- 4 个组件全选， 点击 Next。



## 第四步

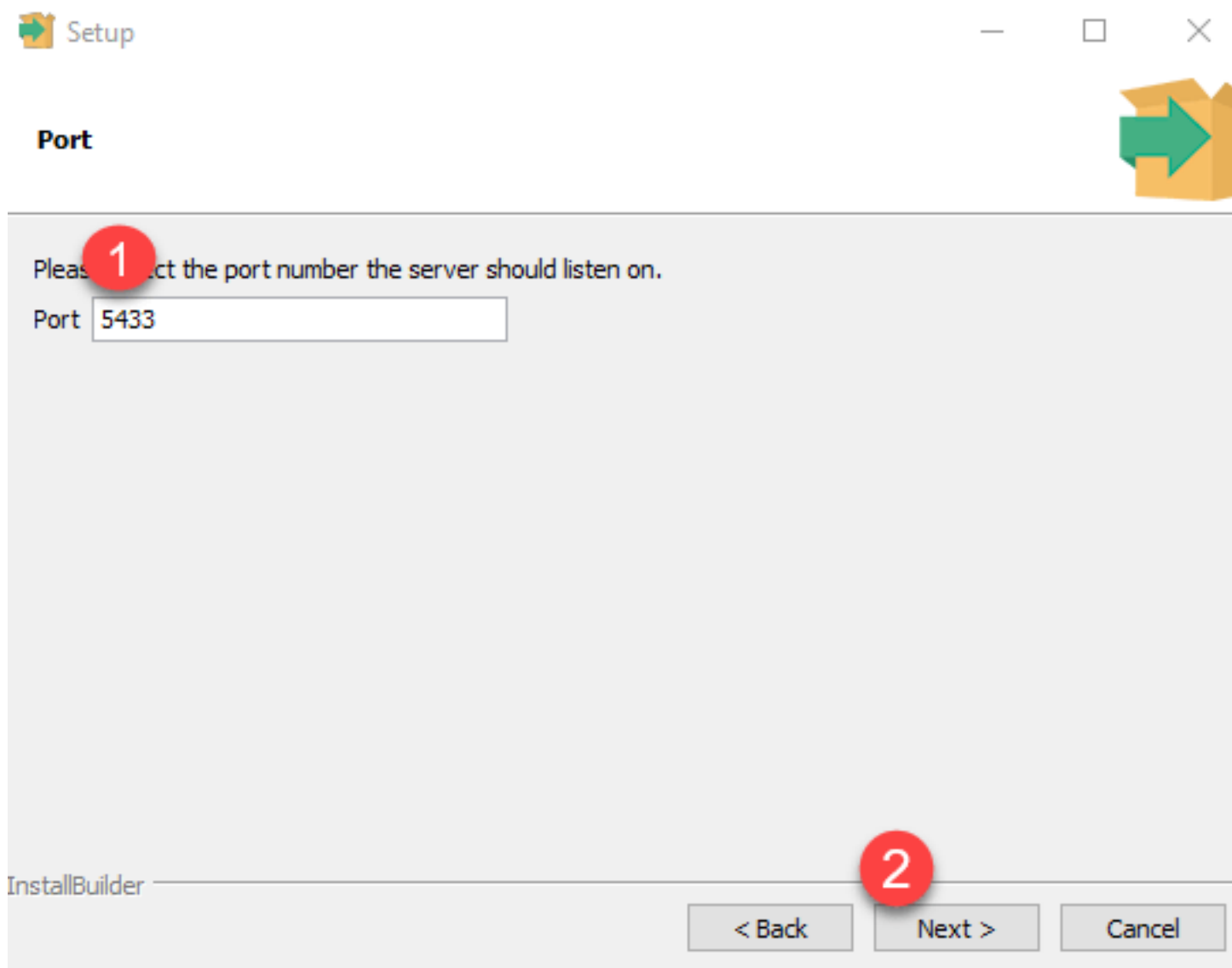
- 这个页面可以设置数据库密码
- 为了方便之后的学习，请大家统一设置为 123456。





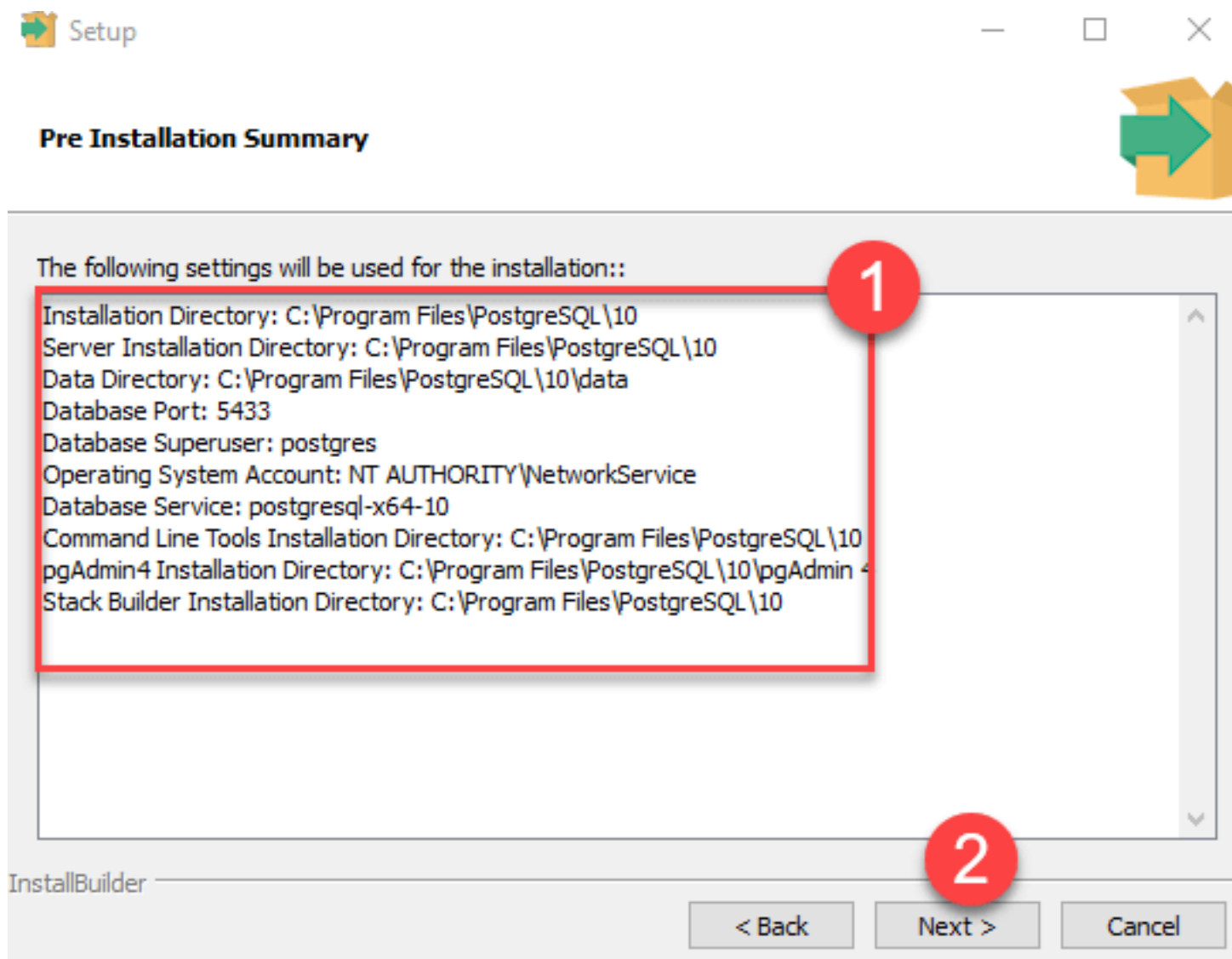
## 第五步

- 这里使用默认的端口号（5433）就可以。如果不行，则使用5434。



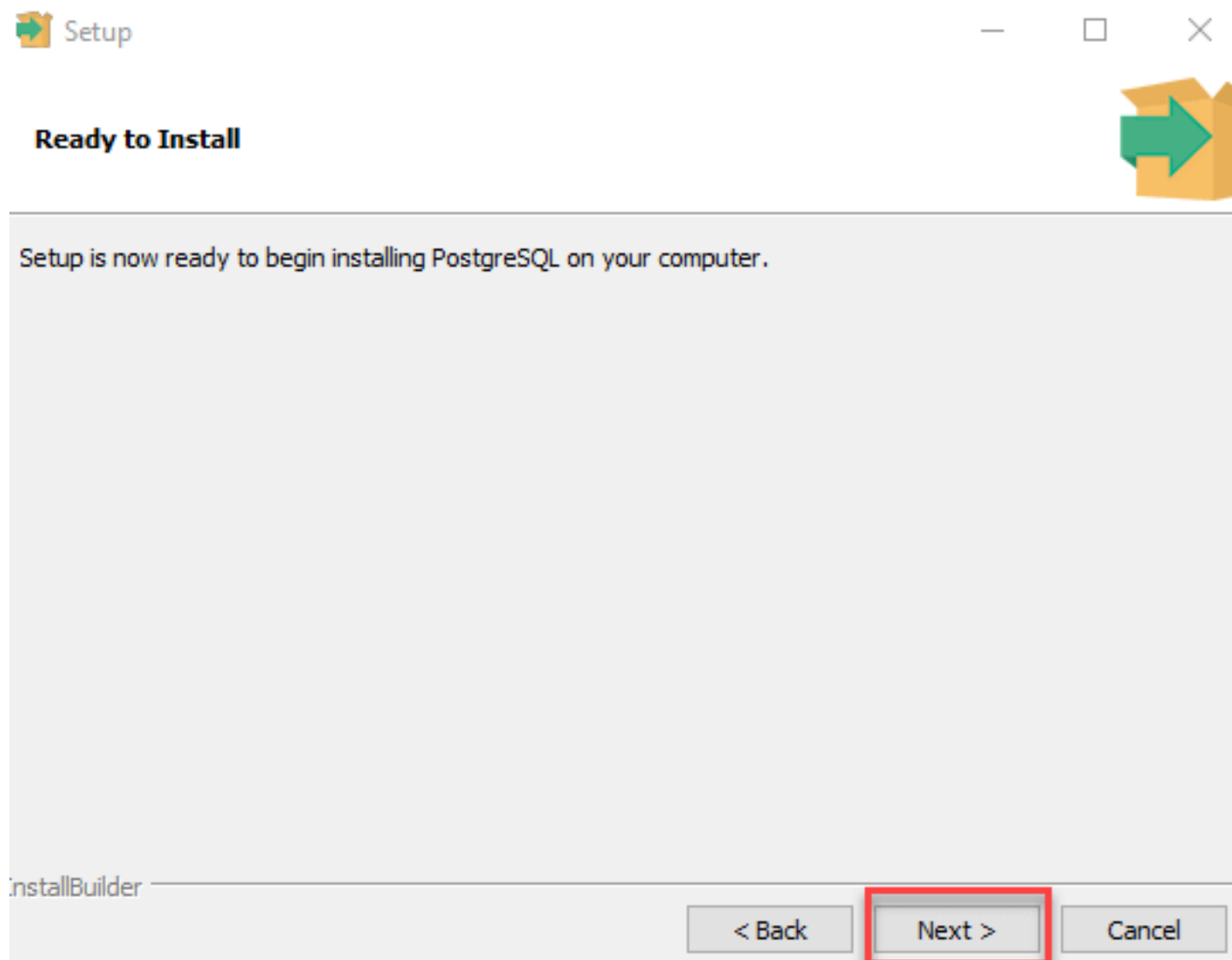
## 第六步

- 请复制或截图保存这个页面的内容：



## 第七步

- 点击 Next，等待安装结束。





## 第八步

- 去掉勾选， 点击 Finish 完成安装。



## 确认安装完成

- 安装完成后，安装路径文件夹里或开始菜单里将可以找到这些文件（可能不完全一致）：



# 目录

1

数据库基本概念

2

PostgreSQL 的  
安装

3

PostgreSQL 的  
基本操作



# SQL shell

- 点击 **SQL Shell** 打开 PostgreSQL 的命令行界面。本节课将使用它进行 SQL 语言的学习。





# 链接 PostgreSQL 数据库

- 进入到命令行窗口，一开始的几行我们不需要输入，直接回车使用默认设定。
- 在输入密码提示出现后后输入刚刚设定的密码（123456），回车。（密码可能会不显示。）

```
Server [localhost]:  
Database [postgres]:  
Port [5432]:  
Username [postgres]:  
Client Encoding [SJIS]:  
ユーザー postgres のパスワード:  
psql (14.2)  
"help"でヘルプを表示します。
```



在这里输入密码

# 创建数据库

- 输入“**CREATE DATABASE 数据库名**”语句创建一个新的数据库：

```
postgres=# CREATE DATABASE hello;
CREATE DATABASE
postgres=#
```

- 输入“**\**”命令可以查看已创建的所有数据库：

```
postgres=# \l
```

データベース一覧					
名前	所有者	エンコーディング	照合順序	Ctype (変換演算子)	アクセス権限
hello	postgres	UTF8	C	C	
jdbc1	postgres	UTF8	C	C	
lc1	postgres	UTF8	C	C	
postgres	postgres	UTF8	C	C	

# 切换数据库

- 要修改或查询数据库中的数据，我们需要先进入相应的数据库。
- “\c 数据库名” 命令可以进入对应数据库：

```
postgres=# \c hello  
データベース"hello"にユーザー"postgres"として接続しました。  
hello=#
```

# 运行外部代码

- 你可以直接在命令行界面中输入 SQL 代码，也可以使用外部代码文件。
- 要使用外部代码文件，使用 “\i '文件路径'” 命令。（注意使用单引号。）

```
hello=# \i 'D:/workspace/sql/hello.sql'  
CREATE TABLE  
hello=# _
```





# SQL 的 “Hello, world!”

- 刚刚我们运行的是一个最基本的 SQL 命令：创建了一个名为 student 的表，包括 3 列数据：id、name 和 score：

```
1 CREATE TABLE student (  
2     id          INT          PRIMARY KEY,  
3     name        VARCHAR(255) NOT NULL,  
4     score       INT  
5 );
```

# 查看表信息

- “\d” 命令可以查看已存在的表信息：

```
hello=# \d
          リレーション一覧
 スキーマ | 名前 | タイプ | 所有者
-----+-----+-----+-----
 public   | student | テーブル | postgres
(1 行)
```

- “\d 表名” 命令可以查看表的具体信息：

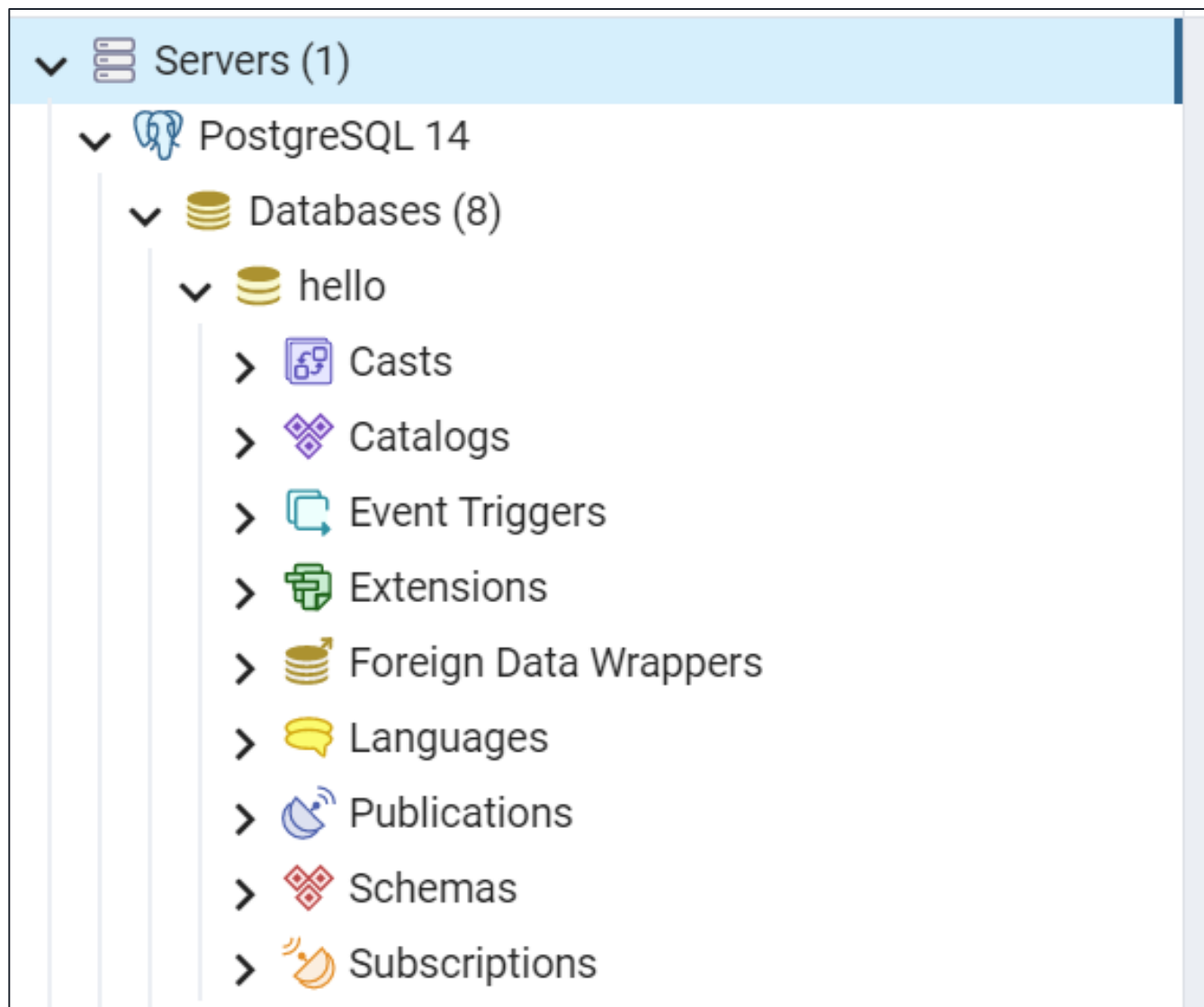
```
hello=# \d student
          テーブル "public.student"
 列   |          タイプ          | 照合順序 | Null 値を許容 | デフォルト
-----+-----+-----+-----+-----
 id    | integer                  |           | not null       |
 name  | character varying(255)   |           | not null       |
 score | integer                  |           |                 |
インデックス:
"student_pkey" PRIMARY KEY, btree (id)
```

# pgAdmin4

- **pgAdmin4** 是数据库的用户可视化操作界面。在这里我们可以更直观地查看到各个数据库的信息，也能直接运行一些 SQL 的代码。
  - 打开 pgAdmin4:
- 
- 初次打开可能会要求设置主密码。（为了方便先暂时都设成 123456 即可。）

# pgAdmin4 的使用

- 点击屏幕左侧的 Servers → PostgreSQL 14 → Database → hello, 找到 hello 数据库:

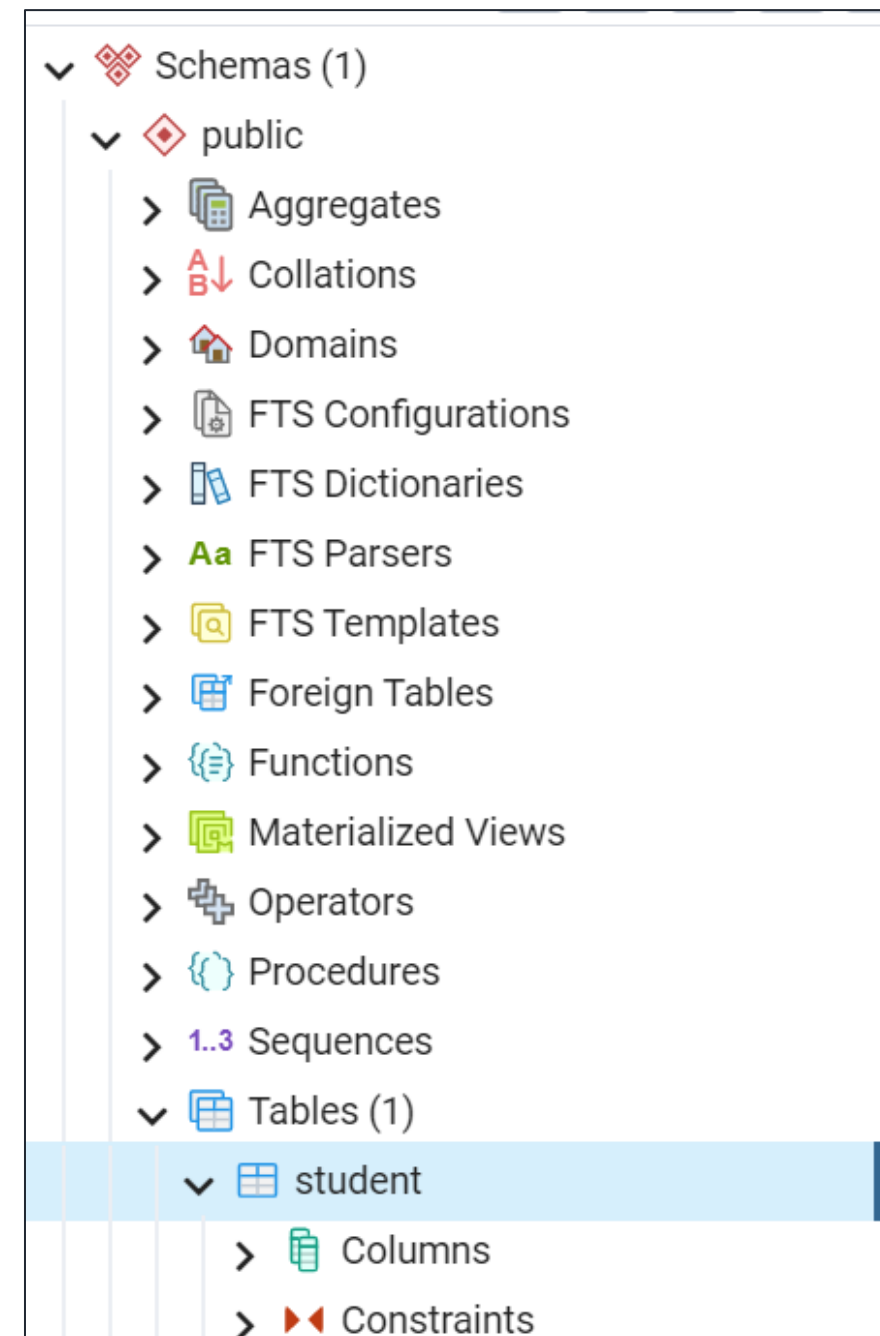


接次页 ➞



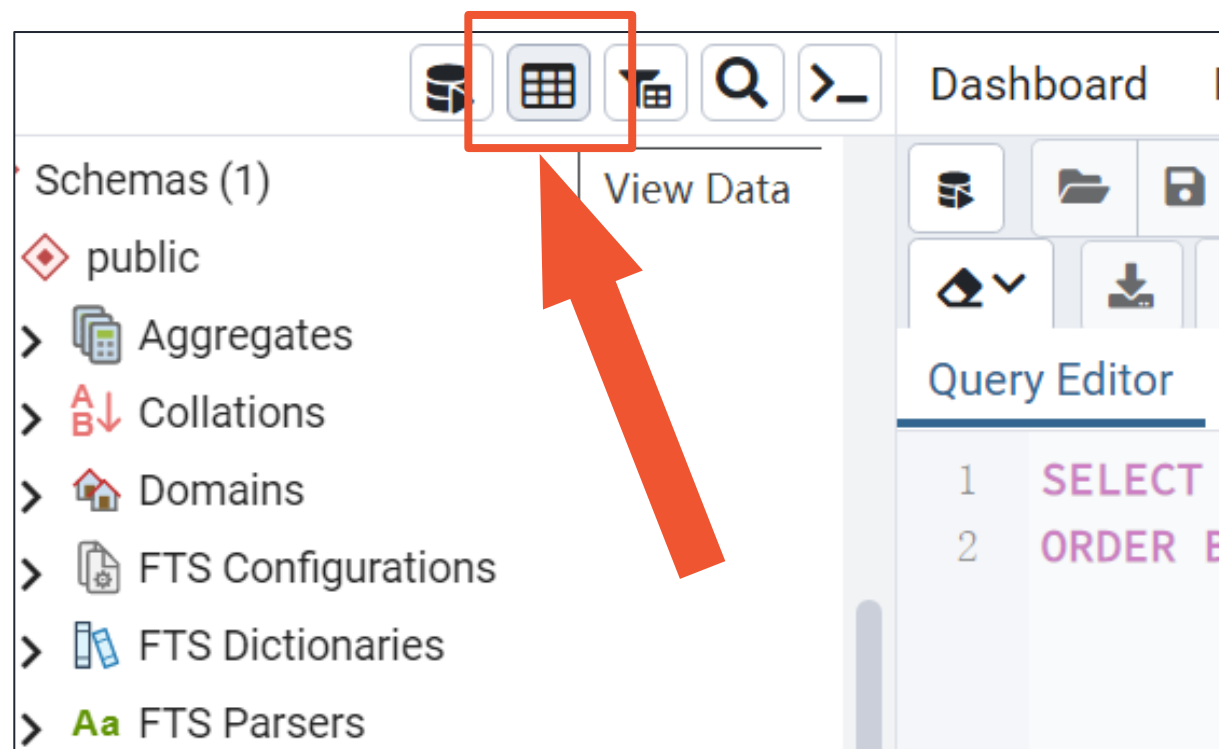


- 找到 Schemas → public → Tables 里的 student 表:



← 接前页

- 点击屏幕上方的 View Data 按钮查看表中的数据（现在是空的）：



Data Output				Explain	Messages	Notifications
	id	name	score			
	[PK] integer	character varying (255)	integer			
1						
2						

Q & A

*Question and answer*

# 总结

## Sum Up

1. 数据库的基本概念：
  - ① 数据库在网络应用开发中的地位；
  - ② 关系型数据库的基本概念；
  - ③ SQL 语言的概念。
2. PostgreSQL 的相关基本操作。



**THANK YOU!**