

# 数据库第一次课程大作业

姓名：\*\*\*      学号：\*\*\*

## 一、 分析与说明

### 1.1 OceanBase 和 TiDB 的两大国产数据库对比

OceanBase 以及 TiDB 是当前在国内最受欢迎的两大数据库。这两个数据库在 2023 年 1 月的 DB-Engine 世界排名中分别达到了 145 名和 108 名。表 1-1 给出了两个数据库具体的差异点。

表 1-1：OceanBase 和 TiDB 的对比表

对比项		OceanBase	TiDB
基本信息	开发团队	阿里	PingCAP
	数据库类型	关系型数据库	分布式数据库
	操作系统支持	CentOS、RedHat、SUSE、Debian/Ubuntu、AliOS、NeoKylin、Kylin	CentOS、Ubuntu LTS、Red Hat（以 Linux 为主）
	对标产品架构	Oracle	MySQL
	许可证类型	商业许可证	开源许可证
	开源时间	2021 年 6 月 1 日	2019 年 9 月
工具与语言	GUI 工具	控制台	TiDB Dashboard
	实现语言	C++	Go
	编程语言支持	Java、C++、Python、Go	Java、Go
存储与分布式	存储引擎	OBStore	TiKV
	存储架构	分布式共享架构	分布式事务架构
	分布式协议	Paxos	Raft
	分布式扩展	水平、垂直	仅水平方向扩缩容
数据	特殊数据类型	空间数据类型等	JSON 类型等
	索引	B-tree、Hash、Bitmap、R-tree，索引结构多样	B-tree，索引结构较为单一
	数据查询优化	Cost-based、Rule-based	Rule-based
	数据分区	Range、Hash、Key、List	主要支持 Range、Hash 的数据分区

其他	安全机制	SSL、Kerberos 等多种安全机制	主要支持 SSL 安全机制
	数据备份	物理备份、逻辑备份	逻辑备份
	兼容性	Oracle、MySQL	MySQL
	部署架构	本地部署	云原生
	部署便捷程度	部署复杂、组件较多（约需一天时间）	简单、架构清晰（最快半天时间）
社区与运维	文档完善度	文档少，完善度低	文档齐全，公司仍在持续更新。
	专利论文	专利 151 项，论文 18 篇，创新创造价值高	专利 26 项，论文 44 篇，学术价值较高
	社区情况	氛围一般	社区成熟氛围好
	企业案例	蚂蚁集团内部，应用案例目前受限。	和包支付、福米科技、光大银行、浦发银行等，应用范围广泛
	评测认可度	信通院可信数据库、电信行业数据库	信通院可信数据库
	培训认证体系	PCTA 与 PCTP 两级认证	OBCA、OBCP、OBCE 三级认证

## 1.2 解决基础系统软件“卡脖子”问题认识

基础系统软件的“卡脖子”问题是指在信息化建设中，由于操作系统、数据库、网络通信等基础系统软件的薄弱和不稳定，导致整个信息化建设的进程和效率受到了极大的影响，甚至出现了系统崩溃、数据丢失等严重问题。这是一个长期存在的问题，也是当前信息技术发展中的一大瓶颈。

基础系统软件是支撑软件应用正常运行的核心技术，若要解决卡脖子问题，我们需要了解系统软件具体容易在哪些地方卡脖子？笔者归结为三类——技术水平、稳定性、安全性。

技术水平是限制行业发展的主要因素，从 0 到 1 也许只差一点技术的融会贯通。良好的研发生产能力才能匹配上当前日新月异的互联网浪潮。我国的技术水平不及一些发达国家，许多著名的公司常常以模仿剽窃他人研究成果进行产品革新，也就常常受制于他人的新技术；反观华为之类的创新型公司，主动攻关实践，突破技术封锁，即将走出硬件的卡脖子阶段。

稳定性方面，随着信息化建设的不断深入，基础系统软件所承载的应用越来越多，系统的负荷也越来越大。而中国的基础系统软件在面对大负荷、高并发等情况时，往往容易出现崩溃、死机等问题，给用户带来了很大不便。疫情期间，

四川省核酸信息平台在 2022 年 9 月在面对全市全员大核酸时数据库系统严重宕机，导致信息录入效率极慢，严重影响了检测效率。特别是在一些关键领域，系统的稳定性尤为重要，一旦出现问题，将会给社会带来严重的后果。

安全性方面，信息安全是当前全球关注的热点问题，而基础系统软件是信息安全的重要组成部分。如图 1 所示，以存储密码为例将会涉及到操作系统、数据库、计算机网络三大基础性软件系统的安全性防护

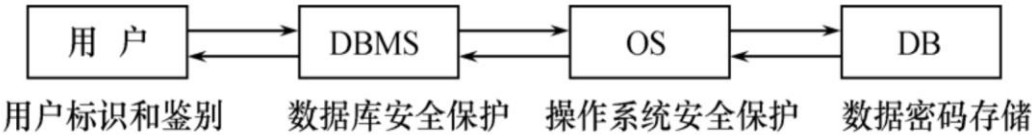


图 1：计算机安全层级模型

然而，一旦基础系统软件在安全性方面存在漏洞和问题，则极其容易受到黑客攻击和病毒侵袭，给国家的信息安全带来了很大的威胁。考虑到基础系统软件常常处于较为底层，涉及面广，一旦入侵基础系统软件，则可轻易盗取用户信息；亦可通过网络连接开展攻击……我们常常听闻新闻爆料某某公司数据库泄露，某某网络发生入侵，可见基础系统软件的安全保证迫在眉睫。

此外，缺乏规范的约束和监管、没有形成良好的竞争力和占有率也是基础软件被卡脖子的部分原因之一。

总之，中国的基础系统软件的“卡脖子”问题是一个比较严重的问题。要解决这个问题，任重而道远。为此，可从以下几个大方向努力：

- 📖 **加强基础系统软件研发：**要想解决基础系统软件“卡脖子”问题，首先要加强基础系统软件的研发。需要加大对基础软件研发的投入，提高研发团队的技术水平，推动技术创新和成果转化，提高基础软件的稳定性和性能。
- 📖 **加强基础系统软件的推广和普及：**除了加强基础系统软件的研发之外，还需要加强基础系统软件的推广和普及。目前，我国基础系统软件的普及率相对较低，需要加大对基础系统软件的宣传和推广力度，提高用户的认识和使用率，提高基础系统软件的市场占有率。
- 📖 **加强对基础系统软件的管理和监控：**为了保障基础系统软件的安全性和稳定性，需要加强对基础系统软件的管理和监控。建立全面的管理和监控机制，对基础系统软件的开发、测试、运行、维护等各个环节进行监督和管理，及时解决基础系统软件中出现的问题和漏洞，提高系统的稳定性和安全性。
- 📖 **加强人才培养和引进：**基础系统软件的研发和推广需要大量的高素质人才，为了培养和引进这些人才，需要加强基础系统软件领域的人才培养

和引进工作。加强对基础系统软件相关专业的教育和培训，鼓励人才创新和成果转化，同时加强对优秀人才的引进和留用工作，为基础系统软件的发展提供强有力的人才支持。

📖 **推动公共服务信息化建设：**基础系统软件是信息化建设的核心技术，要想解决基础系统软件“卡脖子”问题，需要推动公共服务信息化建设，加快数字中国建设进程。推动政务信息化、医疗信息化、教育信息化、公共安全信息化等领域的建设，提高基础系统软件的使用率和市场占有率，为基础系统软件的发展提供更广阔的空间和市场。

解决基础系统软件“卡脖子”问题需要全社会的共同努力，唯有齐心攻坚克难，才能最终柳暗花明。

### 1.3 发展国产数据库技术的建议

随着信息化时代的到来，数据库技术已经成为现代企业信息化建设的核心技术之一。然而，我国在数据库技术领域的发展相对滞后，很难与国际先进水平相比。在基础系统软件“卡脖子”问题日益严峻的背景下，发展国产数据库技术已经成为解决问题的重要途径之一。为此，笔者给出如下的建议：

- 1 完善数据库技术人才培养体系：**数据库技术是一门高度专业化的技术，需要具备一定的理论基础和实践经验。因此，完善数据库技术人才培养体系是发展国产数据库技术的重要保障。应该加强高校数据库技术人才培养，推动教学改革，提高学生的实战能力。同时，政府和企业应该加强数据库技术人才的培养和引进，提高数据库技术人才的水平和数量。
- 2 加强国家级数据库研发和应用平台中心建设：**在国家级数据库研发和应用平台建设方面，国际先进水平仍然比较遥远。我国应该加强国家级数据库研发和应用平台建设，推动数据库技术的创新和发展，加强大数据中心、数据服务处理中心等基础数据中心的建设。加大对国家级数据库研发和应用平台的支持力度，鼓励企业参与其中，提高国家级数据库研发和应用平台的质量和水平。
- 3 推动数据库技术的标准化：**数据库技术的标准化是发展国产数据库技术的必要条件。应该加强对数据库技术标准的制定和推广，鼓励企业积极参与标准的制定和实施，淘汰更新老旧的技术标准，加强数据库技术的交流和合作。同时，政府应该制定相关政策，提高数据库技术的标准化水平，逐步与国际接轨。

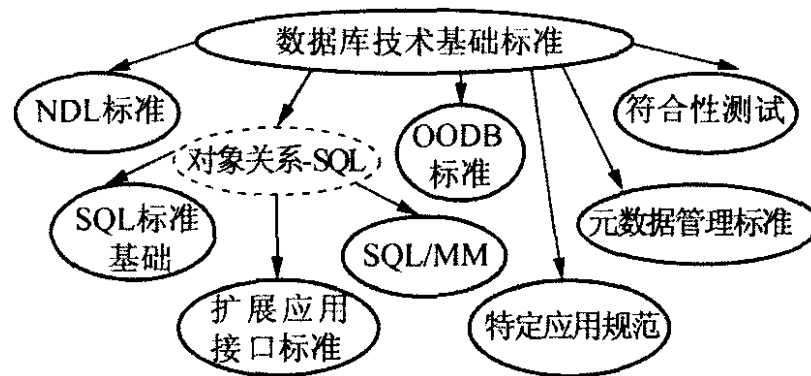


图 2：常见的数据库设计标准

- 4 **加强数据库技术的安全性：**数据库技术的安全性是信息化建设中不可忽视的问题。应该加强数据库技术的安全性研究和防范，提高数据库技术的安全性和可靠性。政府应该加强对数据库技术的监管，制定相关政策和法规，鼓励企业加强数据库技术的安全性保障，提高数据库技术的安全性和可靠性。规避数据泄露、撞库事件的发生
- 5 **加强数据库技术的创新和应用：**鼓励企业加强数据库技术的研发和创新，推动数据库技术的应用和推广，加大对数据库技术的支持力度，提高数据库技术在信息化建设中的应用水平和质量。同时，扩大宣传，将国产化数据库推广到国内使用，形成品牌效应，扩大市场知名度，促进形成良性循环与再创新。

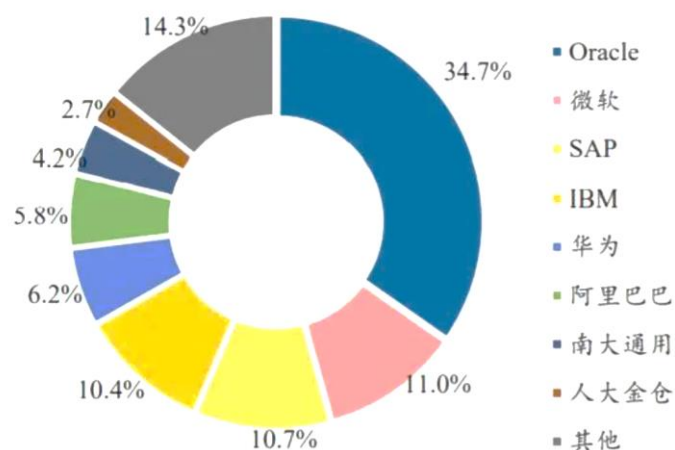


图 3：2019 年主流数据库国内市场占比

总之，发展国产数据库技术之路漫漫，需要共同努力。这些举措将有助于推动我国数据库技术的发展和 innovation，提高我国在基础系统软件领域的国际竞争力，有力应对未来潜在的科技风险。

## 二、 实践

### 2.1 下载与安装 PostgreSQL、pgAdmin4

PostgreSQL 是一款免费开源数据库软件，是世界上一款较为开放的数据库管理系统。其配套的数据库管理软件为 pgAdmin4。

首先，从 PostgreSQL 官网找到与系统匹配的下载路径和版本，下载。

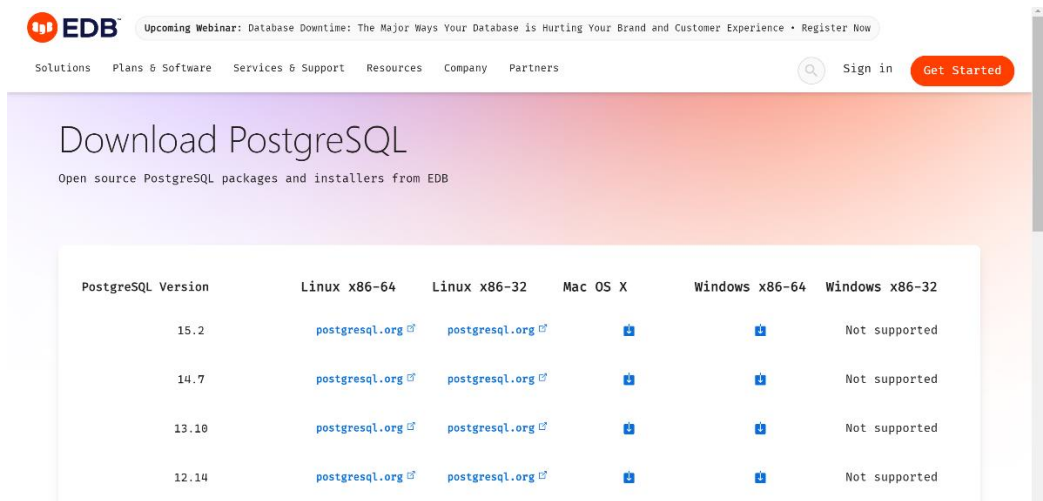


图 4：PostgreSQL 官网

选择安装目录

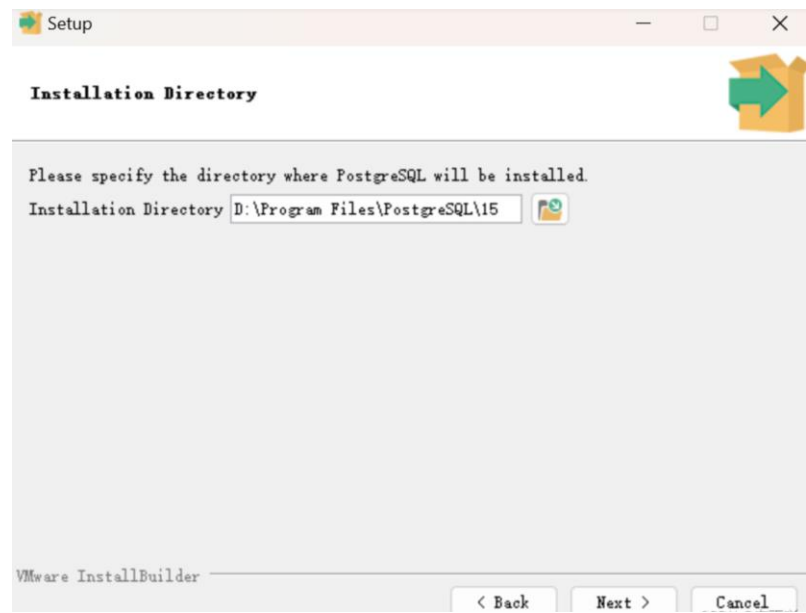


图 5：选择安装目录

之后继续“next”安装，进行账户密码设置

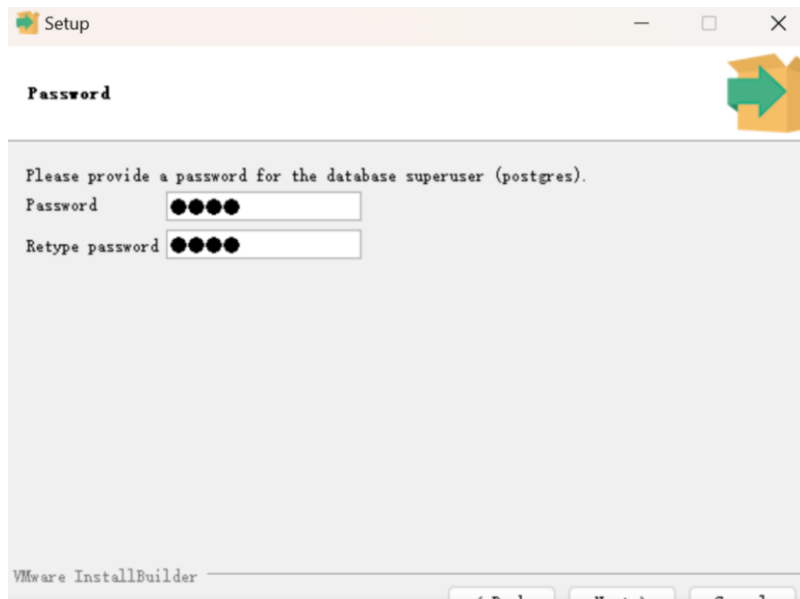


图 6：设置账户密码

设置服务器监听端口（笔者用的默认端口）

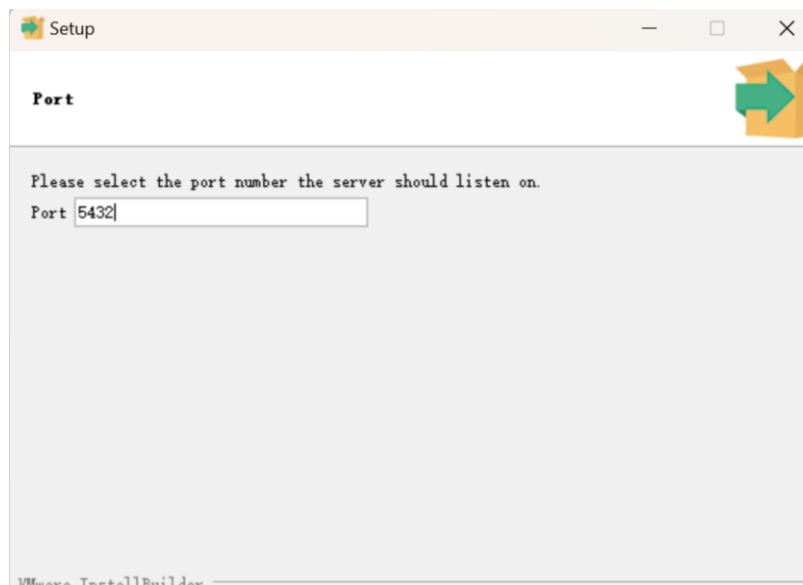


图 7：设置监听端口

安装 PostgreSQL 的同时程序也会安装 pgAdmin4。当安装完成后打开 pgAdmin4。

PgAdmin4 是 PostgreSQL 配套的 GUI 图形化数据库管理系统（软件），界面简洁干净。打开后会要求输入密码连接数据库，当左侧出现“server”时表明数据库已经连接成功。可以在 server 上右键点击进行数据库的创建等操作。当然，也可以点击页面中部的添加数据库来进行手动添加。

界面中部也提供了使用文档等快捷入口，对于新手数据库使用者十分友好。



图 8: pgAdmin4 主界面

## 2.2 PostgreSQL 连接数据库以及 PostgreSQL 的各类对象

打开 pgAdmin4，输入密码后即自动化连接数据库。连接成功后左侧出现 server。点击即可看到更加详细的信息。亦可点击添加服务器进行手动连接。

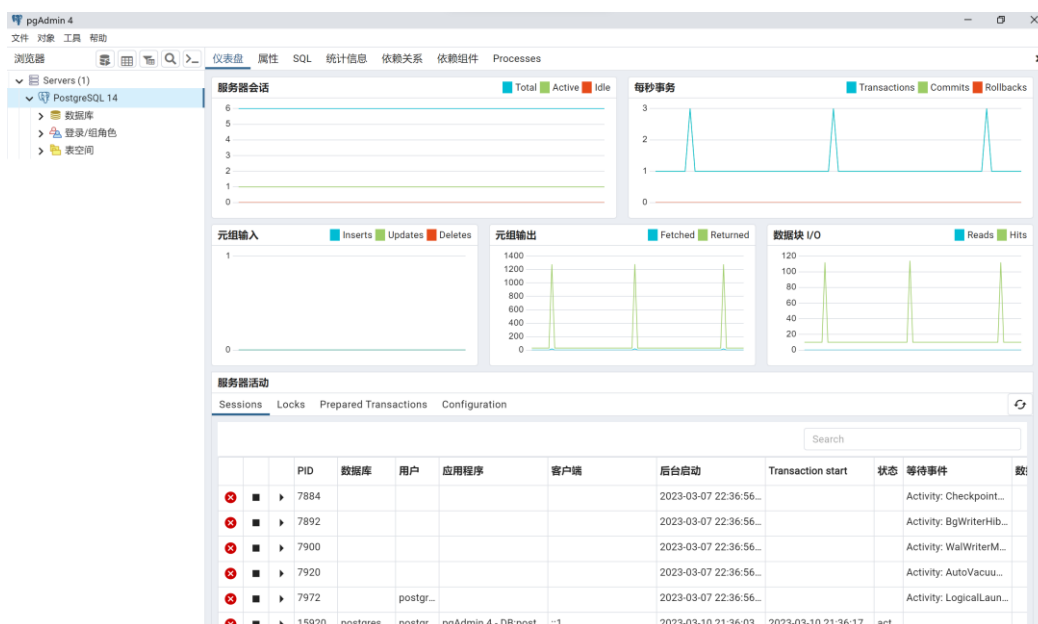


图 9: 数据库连接后的仪表盘信息

在左侧，可以看到我们连接到了 PostgreSQL 14，其下分数据库、组角色、表空间三个区块。其中，数据库是用以存储我们每一个数据库的对象内容，包括扩展、模式架构（类、表、视图、函数等）、事件触发器等；登录/组角色用以数据库用户角色权限的管理控制；表空间即存放我们的各种表数据。

打开数据库，其下有一个自带的默认数据库“Postgres”



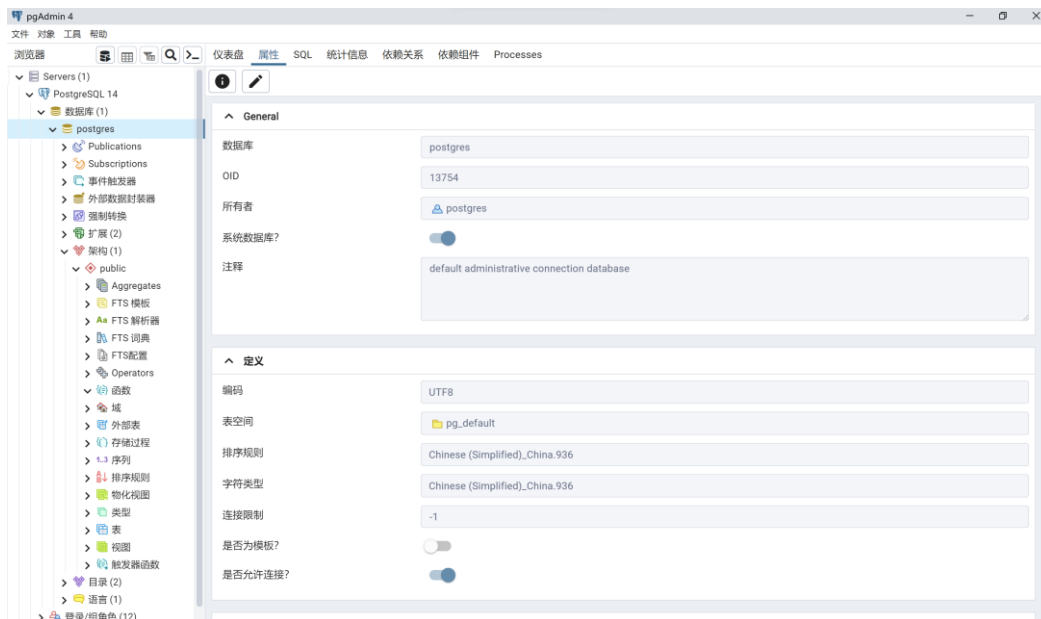


图 10：自带的数据库 Postgres

关于 Postgres 的配置项很多,其中架构(模式)是我们主要关心的。PostgreSQL 模式(也叫做架构, SCHEMA)可以看着是一个表的集合,可以看作操作系统层的目录,一个模式可以包含视图、索引、数据类型、函数和操作符等。

使用模式的优势在于:

- 允许多个用户使用一个数据库并且不会互相干扰。
- 将数据库对象组织成逻辑组以便更容易管理,每个模式对象可互相调用。
- 第三方应用的对象可以放在独立的模式中,这样它们就不会与其他对象的名称发生冲突。

相同的数据库对象可以存在于不同的模式中不会冲突。



图 11：架构下的其他子项

下面详细介绍各个子项：

**表：**数据库的表是数据库的一个重要组成部分，关系型数据库以二维表组建，Postgres 中的表即关联了各个数据之间的关系。行、列构成了二维表的基础，数据库以表存储数据。

可以右键点击创建来创建一个新表，对表的默认值、列、约束、分区等进行设置。也可以直接通过 SQL 语句直接创建表。

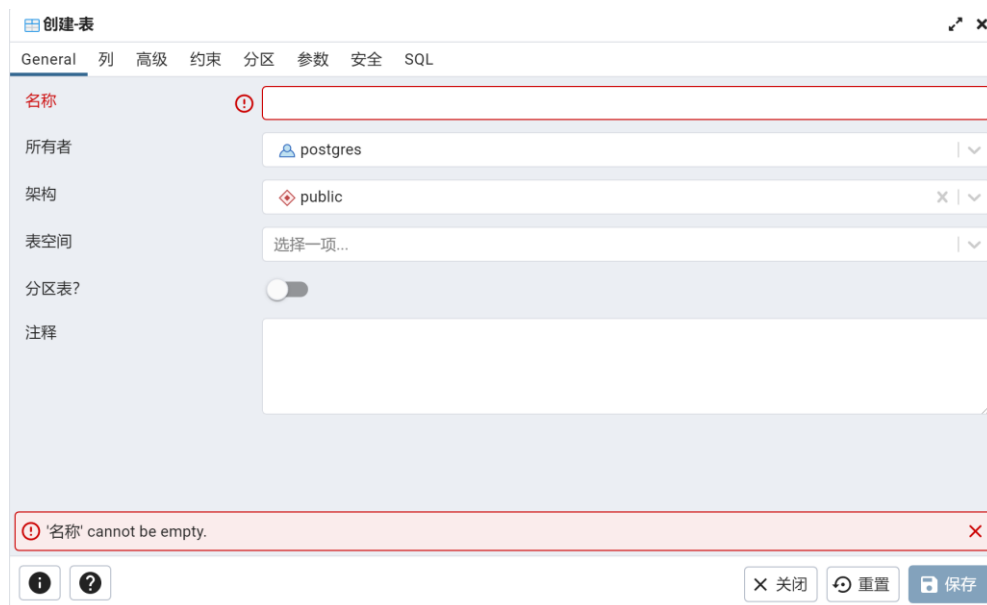


图 12：创建表对话框

如下图我们创建了一个 test\_student 表：

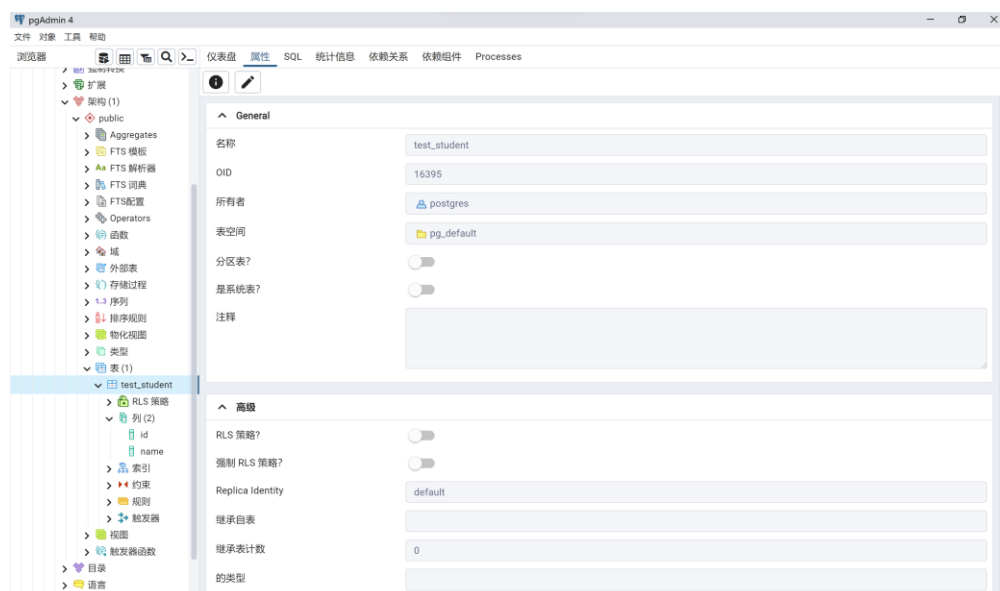


图 13：创建表示例

**视图：**数据的视图也可以看作一个“表”，他实质是是一个以预定义的 PostgreSQL 查询形式存在的表的组合。可以包含某一个表的所有行或者多个表

的选定行。

这一种虚拟表的实现，可以为用户提供如下的帮助：

- 用户或用户组认为更自然或直观查找结构数据的方式。
- 限制数据访问，用户只能看到有限的数​​据，而不是完整的表。
- 汇总各种表中的数据，用于生成报告。

视图是只读的，这可以极大程度减少他人对表的破坏，也就无法在视图上执行删除、插入、修改，可以在视图上创建一个触发器。



图 14：创建视图对话框

**函数：**PostgreSQL 中的函数和编程的函数有着极为相似之处，可以执行数据库特定的操作任务，并返回一个值。例如查询张三的信息，返回他的学号。

通过函数可以实现应用程序的快速封装等。

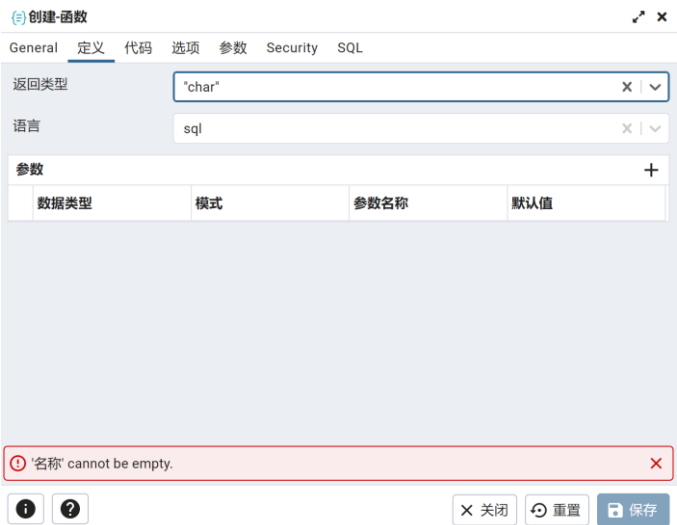


图 15：创建函数对话框

**存储过程：**存储过程则是一种更便捷的函数，是一种块结构的指令任务。在存储前/存储时/存储后执行这一组指令。

**触发器：**PostgreSQL 触发器是数据库的回调函数，它会在指定的数据库事件发生时自动执行/调用。常常用于执行结束前后或者更新操作（包含增删）时，其实也可以理解成一种自动化执行的函数机制。

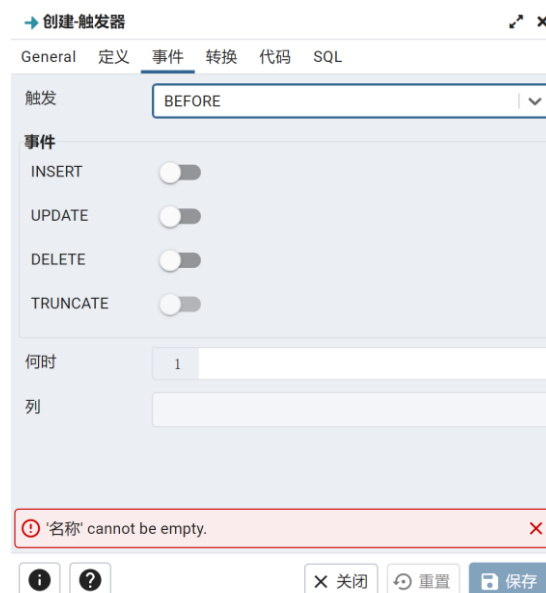


图 16：创建触发器对话框

## 2.3 PostgreSQL 如何具体存储元数据

元数据是关于数据的组织、数据域及其关系的描述性信息，是描述数据的数据，元数据常常存储在系统表（亦称元数据表）之中，PostgreSQL 系统表是由系统自动创建的。

这些系统表以 `pg_` 开头，位于系统模式 `pg_catalog` 中，由全局数据目录（global data directory）下的 `PG_INTERNAL_DIRECTORY` 目录管理的。

元数据表分为系统元数据表和私有元数据表。系统元数据表是 PostgreSQL 内部使用的元数据表，每个数据库都有，全局的，它们的数据比较稳定，一般是启动时候初始化就不变化了；私有元数据表是每个数据库自己管理的，它们的数据会随着操作的变化而变化，比如创建表，建立索引等等。

下面是各个对象具体的存储方式：

- 1 **表空间：**表空间的元数据存储在 `pg_tablespace` 系统表中。该表的每一行表示一个表空间的元数据信息，包括表空间的名称、所属文件系统的路径等信息。
- 2 **数据库：**数据库的元数据存储在 `pg_database` 系统表中。该表的每一行表

示一个数据库的元数据信息，包括数据库的名称、所有者、默认表空间等信息。

- 3 **表：**表的元数据存储在 `pg_class` 系统表中。该表的每一行表示一个表的元数据信息，包括表的名称、所属的命名空间、表的类型（普通表、视图、物化视图等）、表的大小等信息。
- 4 **索引：**索引的元数据存储在 `pg_index` 系统表中。该表的每一行表示一个索引的元数据信息，包括索引的名称、所属的表、索引的类型（B 树索引、哈希索引等）等信息。

除了以上系统表外，还有一些其他的系统表用于存储元数据信息，例如 `pg_attribute` 表用于存储表的列信息，`pg_constraint` 表用于存储表的约束信息等。这些系统表共同构成了 PostgreSQL 中存储对象元数据的基础。