

Simple Database Learning

Part 1 了解数据库

数据库，顾名思义就是存储数据的“库”

而为什么需要数据库呢，这是因为几乎所有应用都离不开后台的“数据”，而数据该如何去“存储”，又该如何去“查找”，这都是需要经过设计的，在part1里，我想通过一些简单的设计来引入“数据库”这一概念

1 设计表

设计一个最简单的数据库，你把数据都存到数组里（这里以学号为例，比如登录网上服务大厅）

数组索引 (Id)	学号
0	2023090909009
1	2023090909010
2	2023090909011

这是存储你的学号的，那该怎么存储你的姓名或者生日呢，当然可以再多几个数组表，如姓名表，成绩表，生日表

下面以姓名表为例

数组索引 (学号)	姓名
2023090909009	tksky
2023090909010	antiO2
2023090909011	sysy.y

把学号作为数组索引来查找与学号关联的索引

但是这样设计会出现一些问题

- 当一个学生退出登录之后，所有数组表对应的存储空间就无效了，但是由于数组的特性，那部分地址空间还是需要分配的（相当于你在c语言里malloc了一块空间，用完了却不free/在java里新建了一个对象，已经没有对它的引用了却还没被回收）
- 当你登录了网上服务大厅后，网上服务大厅需要获取一大堆数据，而这些如果都是在很多很多表里面的话，那就需要反复去读取各个数组，性能会大大降低
- 你想通过姓名来搜索一个用户，那你就需要在一个数组表里不断遍历.....不断遍历.....，查找复杂度是 $O(n)$ ，这是不可接受的

2 优化表

2.1 结构优化

那么可以升级一下，用c语言里的 `结构体` 或java里的 `对象` 来存储一个用户的数据

数组索引 (Id)	姓名	成绩	...
0	tksky	100	...
1	antiO2	200	...
2	sysy.y	300	...

这样就避免了之前的第二个问题，你只需要在这张数组表里读取一次，就能获取到用户的全部数据

但是这样也会有一个问题，当你只需要其中一个或几个数据时，你读取的数组依旧会获取到该用户的全部数据，久而久之，qq号里存的数据不断增加，你要读取数据时的负载也会不断增加

2.2 分组优化

继续进行升级

用多张表来存储用户数据

只存学生身份数据的表↓

数组索引 (Id)	姓名	性别	...
0	tksky	男	...
1	antiO2	男	...
2	sysy.y	女	...

只存学生课程信息的表↓

数组索引 (Id)	班级	成绩	...
0	09	100	...
1	10	200	...
2	11	300	...

其实这种表结构的设计已经比较接近之后要学习的mysql数据库的表结构设计了

2.3 else

还可以进行其他方向的升级

- 把数组换成链表，这样就可以当用户注销身份后回收那一部分空间，当然这也会大大增加索引的难度，因为这样就不能单纯通过索引来访问一个数据了，而是要遍历链表才可以
- 把数组换成其他数据结构，例如平衡二叉树，这样的话既能减少查找所需的时间，也能够当用户在注销身份后回收那一部分空间
- 把索引换成其他值，例如为每一个qq号随机分配一个数字，这样也可以让数组表的空间腾出来
-

Part2 使用DATABASE

上面只是简单的介绍了一下数据库

实际上现在正在使用的大型可用数据库远不是设计几个数组那么简单

而接下来我们将学习使用的就是一个非常经典的数据库：MySQL

至于为什么不是让你们去造一个数据库而只是学习使用，那是因为对刚入学的学生来说还太早了。。

如果有能力，你也可以造一个（错误的）

1 安装Mysql

在你的linux虚拟机或者是linux服务器（windows也可，但是更建议前者）下载并安装一个数据库

建议以下列两种方式安装（也可以都尝试一下）

- 安装包管理器，如apt
- docker 拉取镜像

具体步骤可以用搜索引擎，实在不会可以问问学长学姐们

下面我们将用两种方式来玩玩mysql

2 命令行

此时你可以通过命令行 `mysql` 进入你的数据库辣

2.1 玩玩基本操作

下面我们玩玩数据几个常见的基本操作，建库，建表，搜索，以及他们的逆操作

1. 创建一个名为glimmer的数据库
2. 创建一个名为member的表，主键自己定义
3. 将工作室成员的基本数据写到新建的表上，如名字，性别，方向，年级，性取向（误）等
4. 编写一些查询语句，例如通过姓名查询到某个人的数据，通过性别和年级查询某些人的数据

该部分只能使用命令行

3 图形化工具

在上个过程中，什么数据都得在命令行里输入，比较麻烦

所以也会有人性化的图形化工具出现，更加直观，例如

- mysql workbench (free cash
- navicat (要钱，你可以破解
- clion, idea, pycharm 的数据库模块（这些软件本身是要收费的，但是你可以申请一个学生版
-

任意选择一个图形化工具，进行一下操作

3.1 连接你的数据库

如果你在本地安装的数据库，那么这个过程就很简单，不会特别复杂

但是如果你在虚拟机或者是服务器安装的数据库，如Ubuntu，并且你没有用docker，这个过程对初次使用者将会非常有意思

docker配置相对容易（疯狂暗示

你会遇到很多错误，但是请善用搜索引擎，也可以问问学长学姐们

这里有一点小提示：

- 你可能要配置一些网络设置，比如防火墙（太重要了，血泪教训）
- 你可能要改改mysql的配置文件
- 你可能要配置一些用户

3.2 玩玩基本操作

如果你连接成功了，再次用图形化工具完成 part 2 的 2.1 的相关操作吧

4 小拓展

4.1 questions

- 了解一下什么叫sql注入，为什么会发生这些，又该如何防范？
- 了解一下sql优化的一些策略

4.2 安全性

如果你使用服务器安装的数据库，**这里要十分注意**，说不定哪一天，你就会看到下面情况：

```
mysql> select * from RECOVER_YOUR_DATA ;
+-----+
| text |
+-----+
| All your data is a backed up. You must pay 0.26 BTC to 12KDdVSHvaB46gGTS7pDiBACyWtx5pv5Hs 48 hours for recover it. After 48 hours expiration we will sell all your data on dark markets and the database dump will be dropped from our server! |
| You can buy bitcoin here, https://localbitcoins.com or https://buy.moonpay.io/ After paying write to us in the mail with your DB IP: rambler+2hxf@onionmail.org and you will receive a link to download your database dump. |
+-----+
```

（这张图是今天某21截下来的，一些血泪教训）

大意是你需要支付一定的比特币来恢复你的数据，0.0

关于弱密码和暴露默认端口的问题可能是老生常谈了，都知道密码应该尽可能复杂，但是实际操作起来为了简单方便，又经常抱有侥幸心理地设置弱密码。
将数据库端口暴露在公网中时，默认端口是非常不安全的，但是嫌麻烦，很多时候也是懒得改，这也是非常大的安全隐患。

今天看到一则新闻，网络安全研究组织扫描发现，有近 360 万台 MySQL 服务器在使用默认的 3306 号 TCP 端口暴露在互联网上，想起自己的阿里云服务器的数据库也是默认3306端口开放了，而且用户名密码都是默认的，想起来应该改一下。

---摘自CSDN

所以你需要设置一个强密码，或者是仅能localhost访问，用其他软件访问本地数据库并向外服务，或者换一个端口以减小被攻击的可能性。

4.3 除了最简单的操作，还有什么？

- 外键是什么？试着多建几个彼此相互联系的表，试试多表查询吧！
- 了解一下什么是事务？ACID是什么？

Part 3 尝试一些编码

其实你们都可以通过使用语言来直接操作数据库，数据库本身是提供了这些功能接口的

而各种语言也都有相应的中间件来为我们提供更好的抽象接口来实现操作数据库这一功能

这里根据你们所学语言自行发挥

只需要能够完成：在运行你写的程序时能够访问数据库、能够查找和改变数据库的数据

java选手&后端选手：

后端一般也从java入门嘛，所以可以去了解下java如何直接对数据库进行操作

学习下JDBC，能够使用JDBC连接到mysql数据库并能进行操作（当然，你如果有能力，也可以用用什么mybatis辣）

Part 4 了解一下别的数据库

某一天，yewpo在网上看上一个词“NoSql数据库”，yewpo惊叹“居然还有纯纯与sql无关的数据库”！

事实上，这个世界上当然不可能只有mysql这一个数据库辣！

- 了解一下什么叫关系型数据库，什么叫非关系型数据库，NoSql究竟是什么意思？
- 思考一下什么时候适合用关系型数据库，什么适合则适合用一些非关系型数据库
- 了解一下常见的NoSql数据库的使用（增删改查等等），比如redis、mongodb.....你也可以同样试着用代码连接到你的NoSql数据库，并做一些简单的操作
- redis的事务和mysql中的事务一样吗？有什么不同？
-

数据库还有很多东西，值得你慢慢探索！