BASIC SQL III

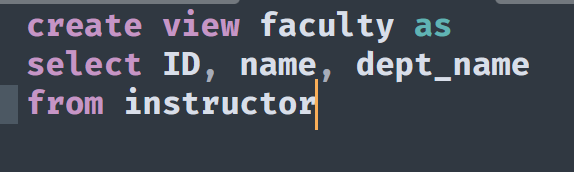
1. 实验目的

学习SQL中的view,掌握如何视图运行逻辑，进一步学习进阶数据库操作方式。

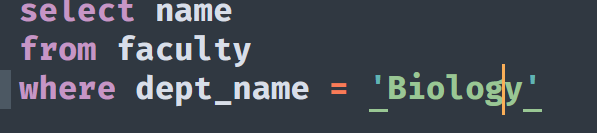
1. 实验内容
2. View

执行如下sql语句，并且得到结果

下图为创建视图



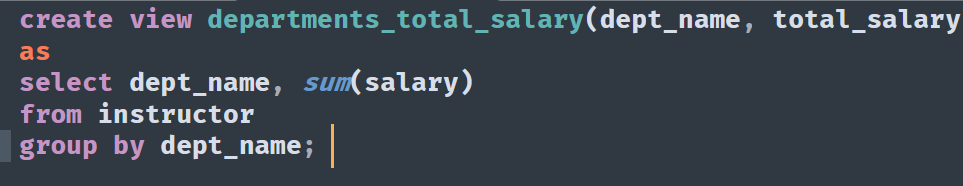
下图为使用已经创建好的视图faculty并且从中选出name字段



下图为得到的结果



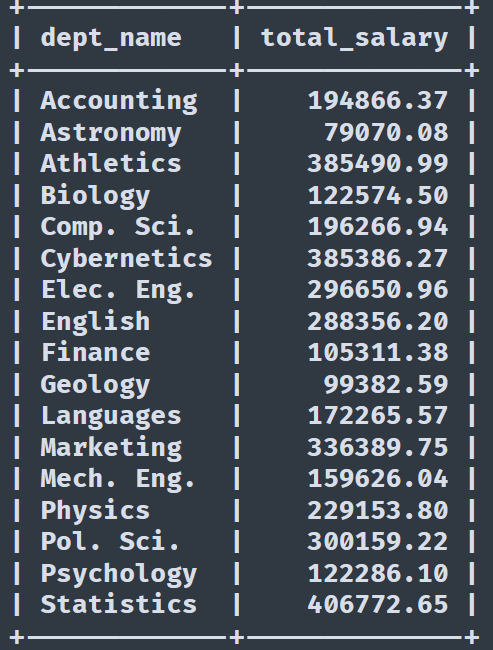
下面创建department总薪水视图



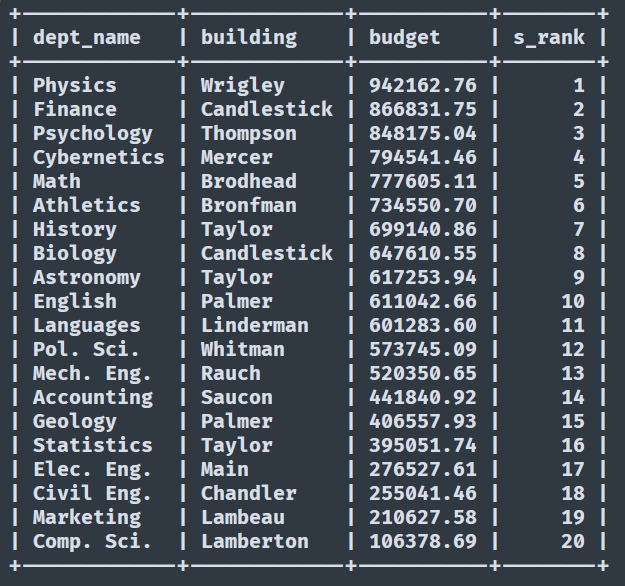
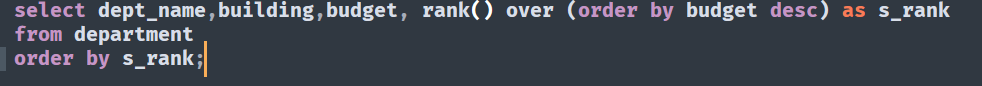
然后查询表中所有内容



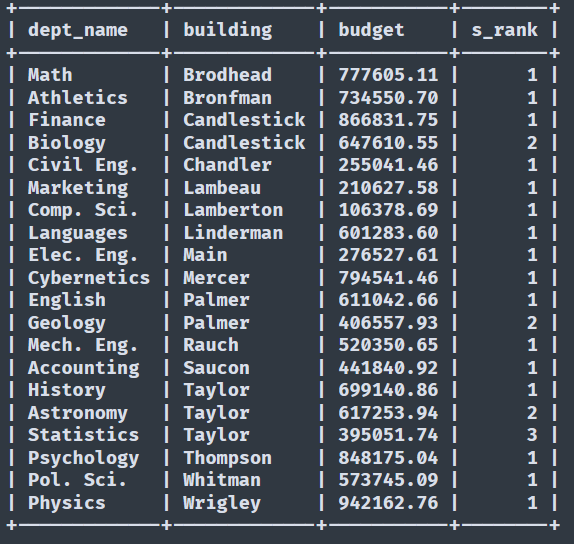
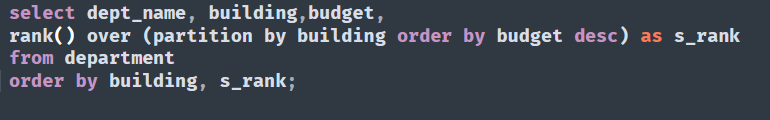
下图为查询结果



1. Advanced aggregation
2. 对学院根据预算进行排序，并给出排名

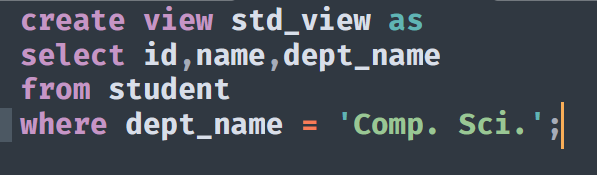


1. 对每个建筑内的学院根据预算进行排序，并给出排名

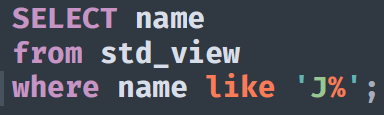


1. 实验问题

创建计算机学院学生视图



查询计算机系中以J开头的学生的名字

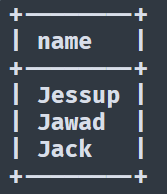
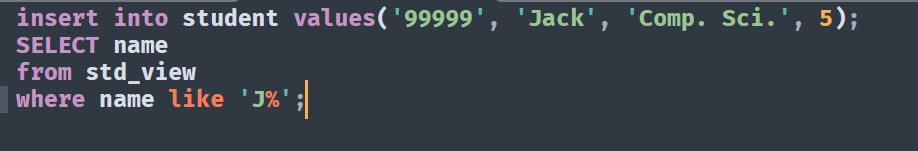


结果如下，有两位同学name以J开头

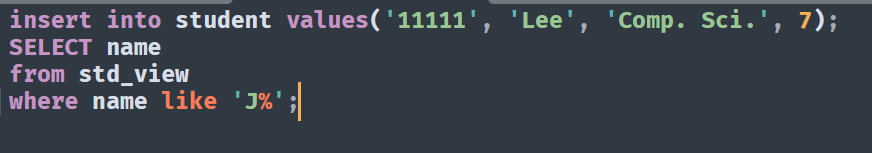


分别插入Jack和Lee并且分别执行查询

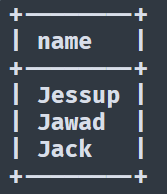
插入Jack并且查询



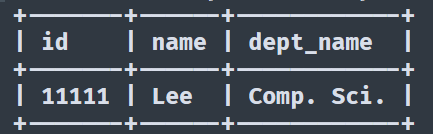
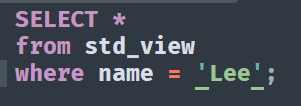
插入Lee并且查询



再次执行查询结果仍为那三位同学



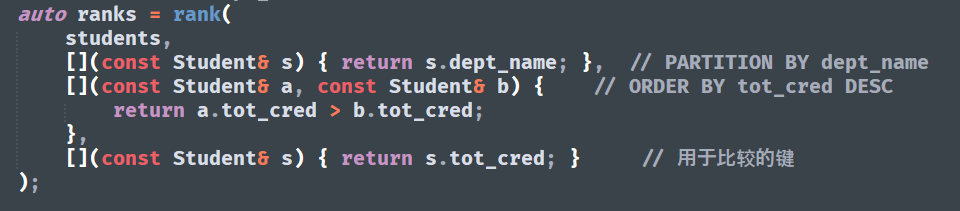
为验证Lee插入成功，查询std\_view中名字为Lee的学生



下面使用c++实现rank()，核心代码如下图



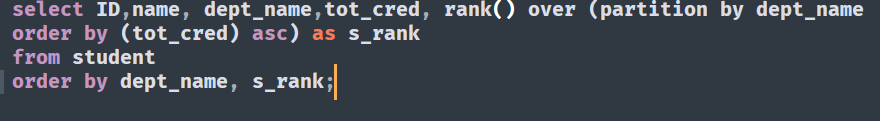
调用时体现出原有rank()的使用逻辑，假如我按照dept\_name进行分组，然后按照tot\_cred进行降序排列，函数调用如下图：



调用详细解释：

1. 第一个参数students表示学生表
2. 第二个参数为函数地址，表示分组依据，类似于partition子句
3. 第三个参数为排序依据字段，类似于order by子句
4. 第四个参数为升序或者降序逻辑，类似于desc和asc

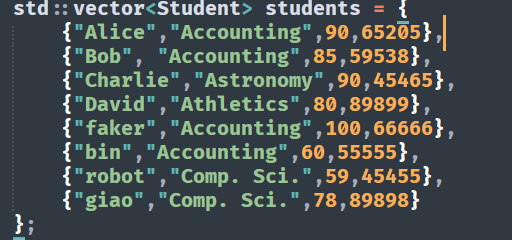
等价sql语句查询如下图



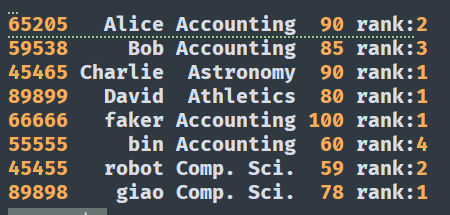
**备注：这里我写的rank函数输出时未考虑此图中的最后一行order by**

使用测试数据进行测试使用c++实现的rank()函数

测试数据集如下



输出结果如下



1. 反思总结

本次实验主要是对于视图的实验，通过这次实验，我理解了视图运行的机制以及必要性，对于不同的用户，可获取的数据库数据应有权限之分，所有视图存在是十分必要的，最后我用c++语言实现了rank函数，同时复习了之前学习的c++模板知识，这对于我对于知识的回顾有很大的启示意义，在如何实现可变参数的时候，我使用了c++中的lambda表达式，掌握一门语言各种语法，这对于我以后的编程学习有很大的帮助。

1. 代码

