**《数据库系统原理》实验报告**

**实验题目：索引管理**

**姓名： 郑德凯** **实验日期： 2024年 11月 26日**

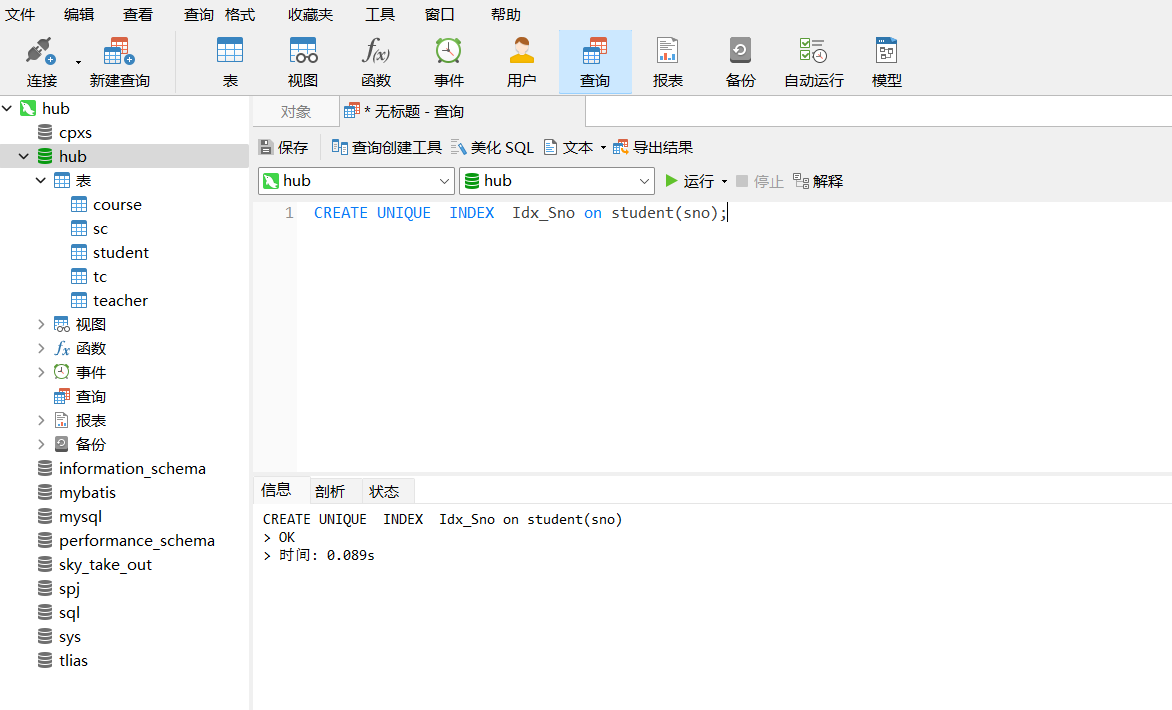
**实验内容及完成情况：**

# 一、实验目的

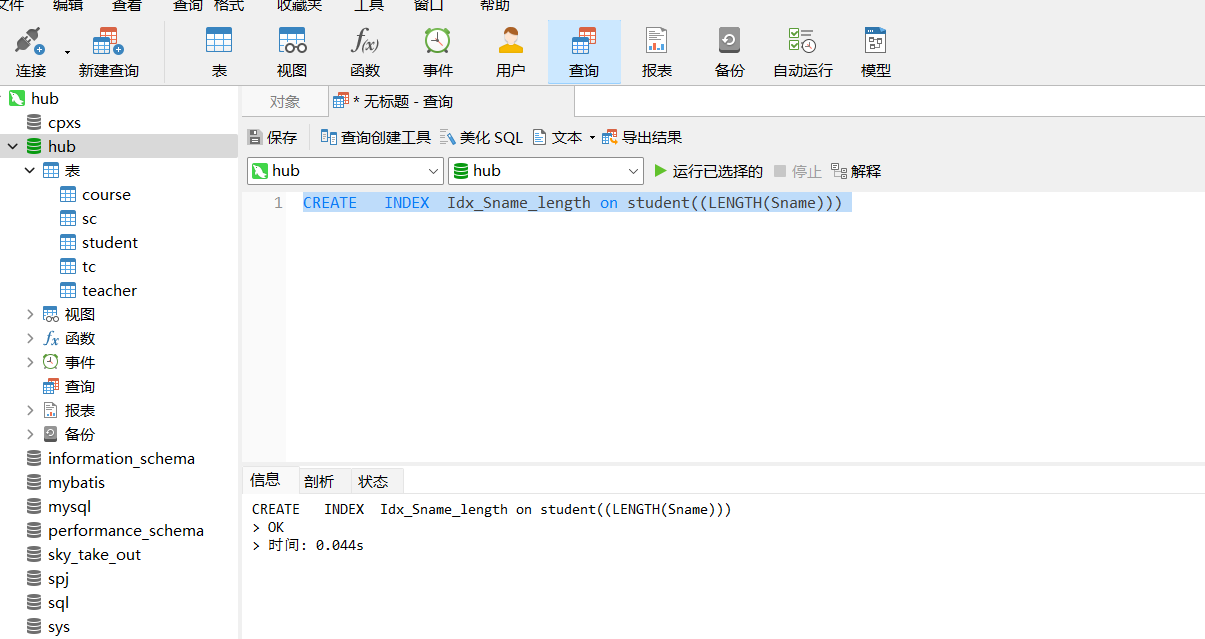
本实验的目的是使学生熟练掌握索引设计的原则和技巧，能够创建合适的索引以提高数据库查询、统计分析效率。针对给定的数据库模式和具体应用需求，创建唯一索引、函数索引、复合索引等；修改索引；删除索引。同时设计SQL查询验证索引有效性。

# 实验内容

1.创建唯一索引，在student表的学生学号字段上创建唯一索引。



2.创建函数索引（对某个属性的函数创建索引，称为函数索引），在student表的学生名字字段上创建一个学生名字长度的函数索引。



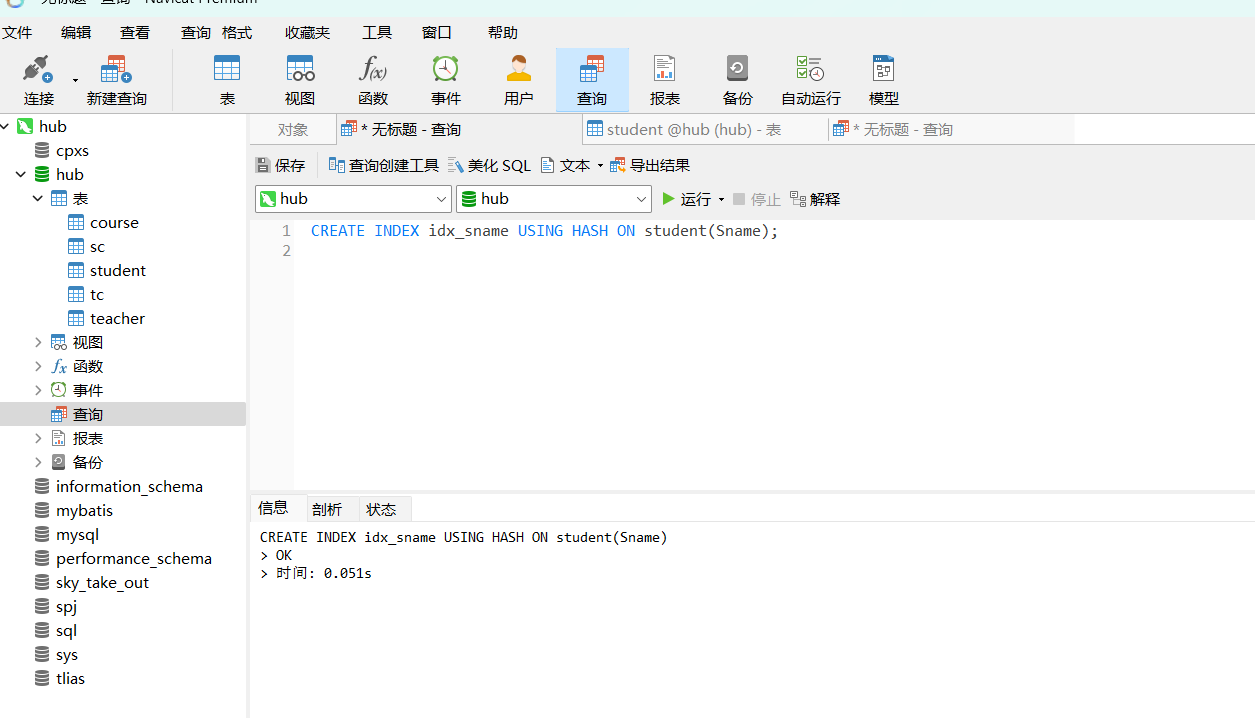
3.创建复合索引（对两个及两个以上的属性创建索引，称为复合索引）在student表的院系名称和专业名称两个字段上创建一个复合索引。



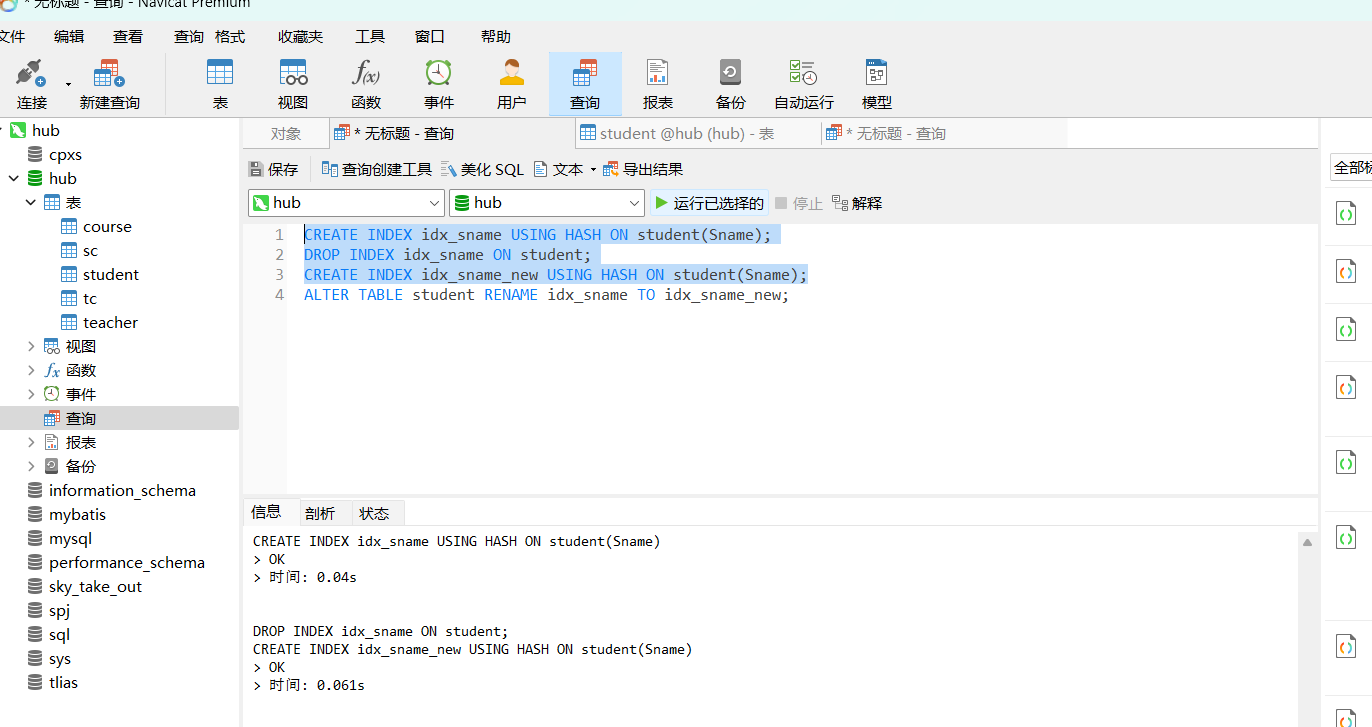
4.创建聚簇索引，在student表的院系名称字段上创建一个聚簇索引。



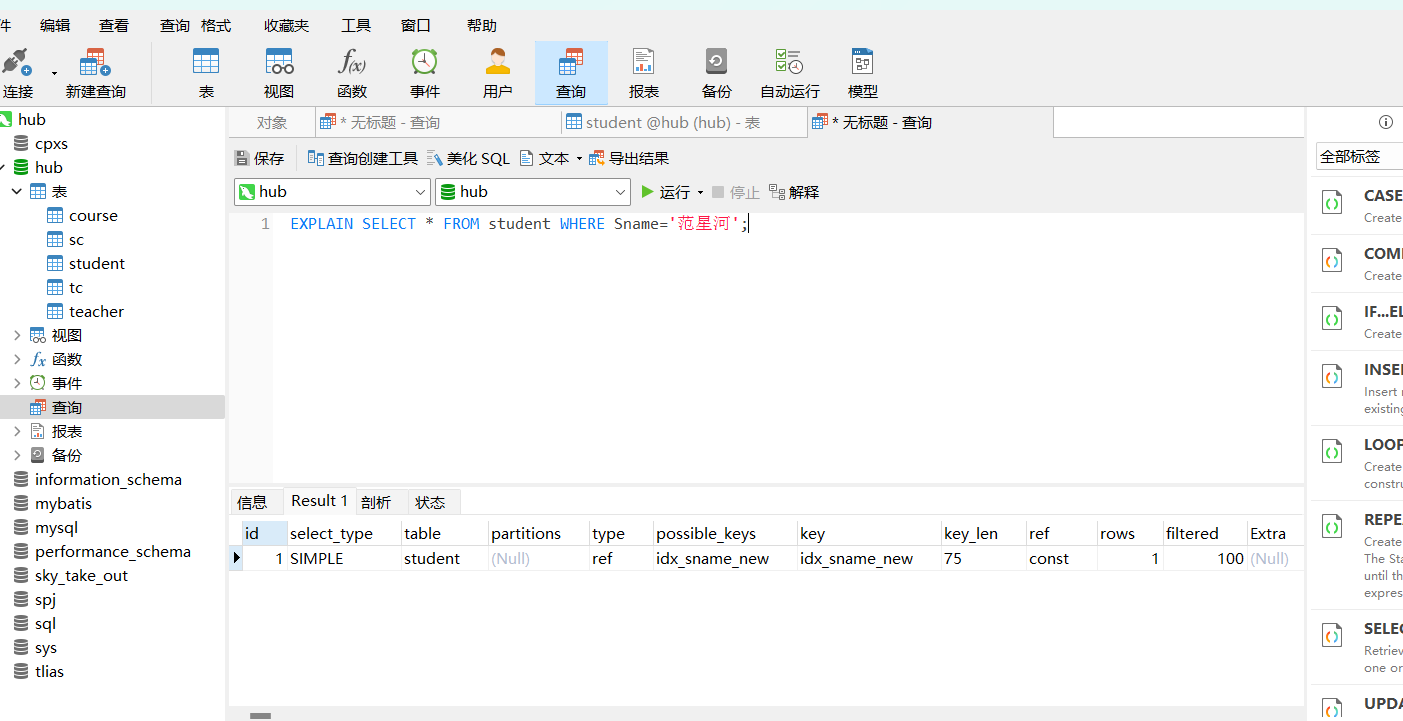
5.创建Hash索引，在student表的学生名字字段上创建一个Hash索引。



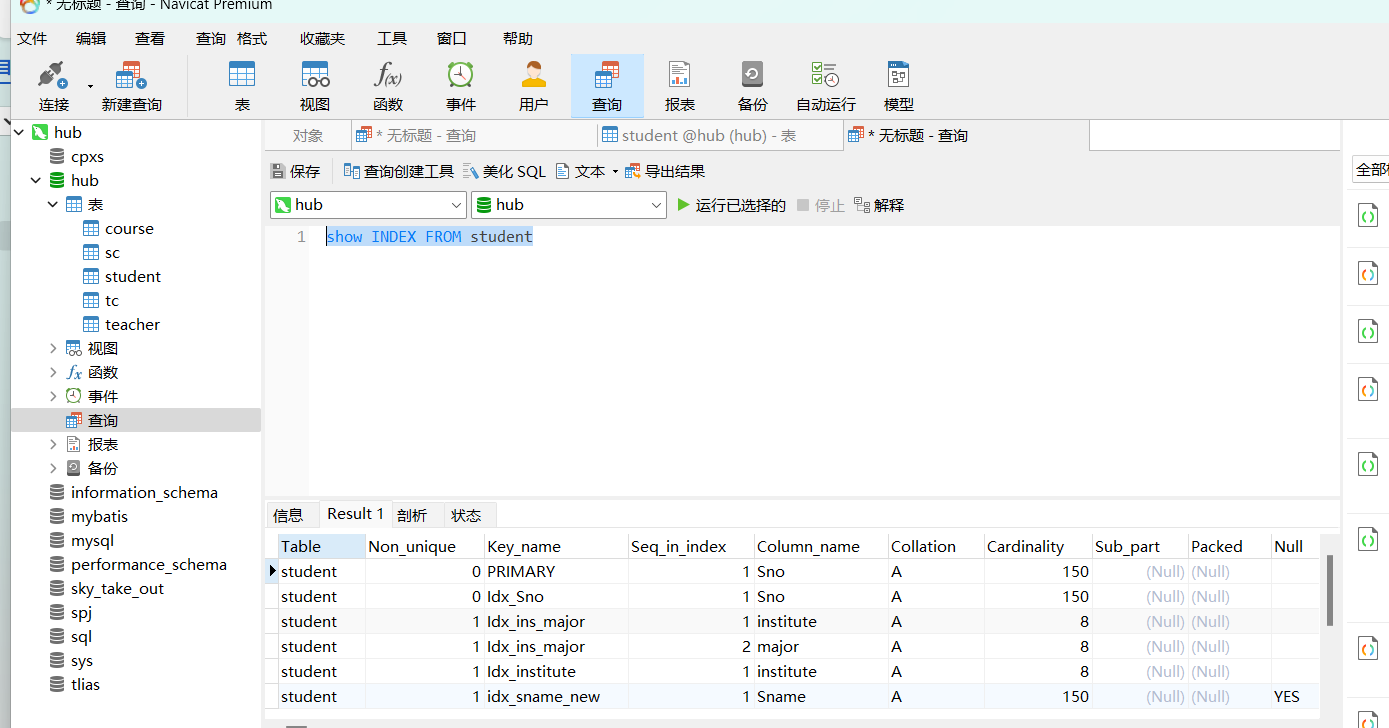
1. 修改索引名称，修改student表的学生名字字段上的Hash索引名。



1. 分析某个SQL查询语句执行时是否使用了索引。



1. 使用SHOW INDEX语句来查看表的索引信息。



# 课后练习题

1. 在一个表的多个字段上创建的复合索引，与相应的每个字段上创建的多个简单索引有何异同？请设计相应的例子加以验证。

复合索引和简单索引的主要区别在于：

查询优化范围：复合索引适合多字段组合查询，尤其是符合最左前缀原则的查询；简单索引独立适用每个字段的单字段查询。

性能：复合索引在多字段过滤时性能更优，而多个简单索引在字段独立查询时表现更好。

存储：复合索引节省存储空间，因为只需存储一次索引结构。

验证示例：

假设表 user\_info 有字段 age 和 city：

创建复合索引：

CREATE INDEX idx\_age\_city ON user\_info(age, city);

创建简单索引：

CREATE INDEX idx\_age ON user\_info(age);

CREATE INDEX idx\_city ON user\_info(city);

对比查询：

复合索引更优：

SELECT \* FROM user\_info WHERE age = 30 AND city = 'New York';

简单索引更优（单字段查询）：

SELECT \* FROM user\_info WHERE city = 'New York';

**实验总结：**

### ****实验总结****

本次实验探讨了 MySQL 中的单字段索引、复合索引和 Hash 索引。单字段索引适用于单独查询，而复合索引更适合多字段组合查询。Hash 索引在 Memory 引擎中有效，适用于精确匹配查询。通过实验验证了不同索引在查询性能上的差异及适用场景。

**教师评语及成绩**：