**中山大学数据科学与计算机学院**

**移动信息工程专业-数据库系统**

**本科生实验报告**

**（2017-2018学年秋季学期）**

课程名称：**数据库系统实验**

|  |  |  |  |
| --- | --- | --- | --- |
| 教学班级 | **15M1** | 专业（方向） | **移动互联网** |
| 学号 | **15352408** | 姓名 | **张镓伟** |

# 实验目的

1. 熟悉SQL的有关视图的操作。

2. 能够使用SQL语句创建需要的视图，对视图进行查询和取消视图。

# 实验内容

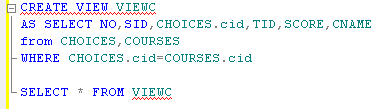
1. 定义常见的视图形式，包括：
   * 行列子集视图
   * WITH CHECK OPTION的视图
   * 基于多个基表的视图
   * 基于视图的视图
   * 带表达式的视图
   * 分组视图
2. 考察 WITH CHECK OPTION这一语句在视图定义后产生的影响，包括  
   对修改操作、删除操作、插入操作的影响
3. 讨论视图的数据更新情况，对子行列视图进行数据更新。
4. 使用DROP语句删除一个视图，由该视图导出的其他视图定义仍在数据字典中，但已不能使用，必须显式删除。同样的原因，删除基表时，由该基表导出的所有视图定义都必须显式删除。

。

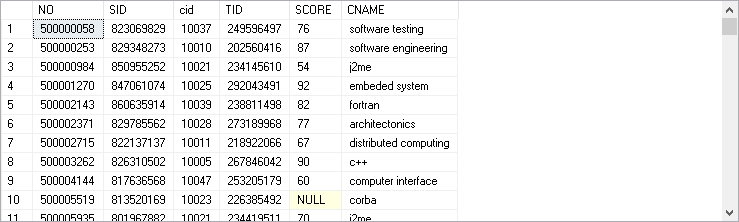
# 实验过程及结果

(1) 定义选课信息和课程名称的视图 VIEWC。

Sql语句：

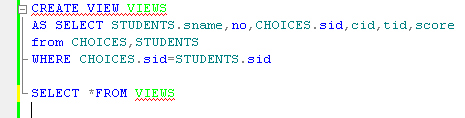


查询结果：

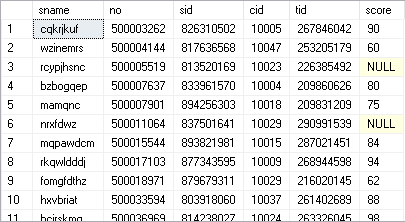


(2) 定义学生姓名与选课信息的视图 VIEWS。

Sql语句：

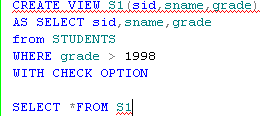


查询结果如图：

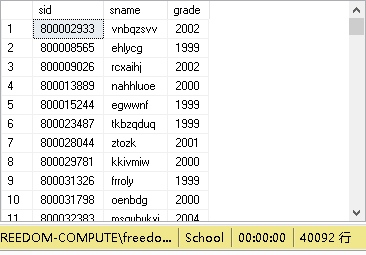


(3) 定义年级低于1998的学生的视图S1(SID, SNAME, GRADE。

Sql语句：

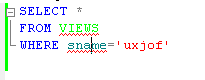


查询结果如图：

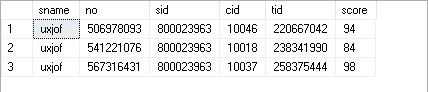


(4) 查询学生为“ uxjof”的学生的选课信息。

Sql：语句：

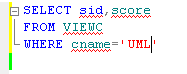


查询结果如下：

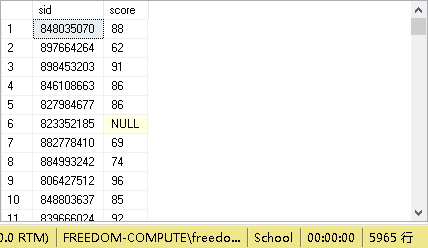


(5) 查询选修课程“UML”的学生的编号和成绩。

Sql语句：

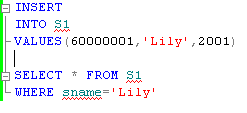


查询结果如下：



(6) 向视图S1插入记录(60000001,Lily,2001)。

Sql语句：



查询结果如下：



(7) 定义包括更新和插入约束的视图S1,尝试向视图插入记录(60000001,Lily,1997),删除所有年级为1999的学生记录,讨论更新和插入约束带来的影响。

之前已经创建了含WITH CHECK OPTION 的S1视图，先删除原先的Lily记录，再进

行下面的操作。

Sql：

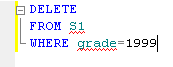


结果：



WITH CHECK OPTION会对插入或更新操作进行检查。由于插入的数据不符合视图定义grade>1998，所以数据插入失败。

Sql:



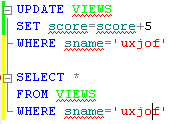
结果：



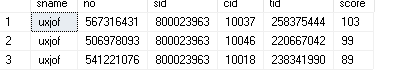
在视图中删除学生信息会在STUDENTS表中一起删掉，而我由于CHOICES表中含有指向STUDENTS表的外键，所以删除操作失败。

(8) 在视图 VIEWS中将姓名为“ uxjof”的学生的选课成绩都加上5分。

Sql语句：



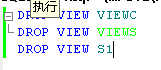
查询结果如下：

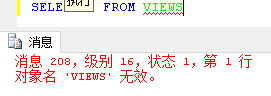
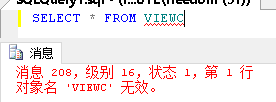


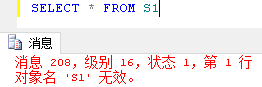
原分数在第4题，对比可知分数都增加了5分。

(9) 取消以上建立的所有视图。

Sql 语句如下：







# 实验感想

这次实验是熟悉sql语句中的跟视图操作相关的部分。创建视图优势在于其实就是把本来要联合几个表的数据放在一张表里，后面只要直接对这张表查询就可以，这样如果查询次数多的话，sql语句书写也很简单，可以节省不少时间。