《数据库技术实验》报告

|  |  |  |
| --- | --- | --- |
| **第 五 次实验** | **日期：2022-10-29** | **得分：** |
| **学号：** | **姓名：*Steven*** | **专业：智能科学与技术** |

一、实验目的

1. 掌握定义、查询和更新视图的SQL语句

2. 了解数据库的外模式、视图的作用

3. 了解视图消解原理、可更新视图、不可更新视图、行列子集视图

二、实验内容

（一）实验题

1. 在Student\_Course数据库上验证课本上定义、查询和更新视图的例子

2. 在TPC\_H数据库上自行设计5个以上关于视图操作的SQL语句，要求包括基于多个基本表的视图、分组视图等

（二）思考题

1. 在使用方面，视图与基本表有些什么异同

2. 修改基本表的结构会对基于它的视图产生什么影响？

三、实验过程及实验结果

1. 验证课本例

(1) 建立计算机系的学生信息视图：

|  |  |
| --- | --- |
| CREATE VIEW CS\_STU AS      SELECT Sno, Sname, Sage      FROM Student      WHERE Sdept = '计算机'      WITH CHECK  OPTION; |  |
| CREATE VIEW CS\_STU\_2(SNO, SNAME, SAGE) AS      SELECT student.Sno, Sname, Sage FROM student WHERE sdept =  '计算机'; |  |

(2) 多表联合视图：选修1号课程的学生的视图：

|  |
| --- |
| CREATE VIEW C1\_STU(SNO, SNAME, SSEX, SAGE, GRADE) AS      SELECT          student.Sno, student.Sname,          student.Ssex, student.Sage,          sc.grade      FROM student, sc      WHERE          student.sno = sc.sno AND sc.cno = 1      WITH CHECK OPTION; |
|  |

(3) 查询视图：查询选修了1号课程的计算机系学生：

|  |  |
| --- | --- |
| SELECT      CS\_Stu.Sno,      CS\_Stu.Sname  FROM CS\_Stu, sc  WHERE      CS\_Stu.Sno = sc.sno      AND sc.cno = 1; |  |

(4) 更新视图：插入一个新的学生记录，学号201215129，姓名‘赵新’，年龄20

尝试使用如下的语句进行更新，但是失败了。

|  |
| --- |
| INSERT INTO CS\_Stu SET Sno='201215129', Sname='赵新', Sage=20; |
|  |

提示“赵新”的属性不满足CHECK OPTION，但是按照课本的意思，既然是加到了含CHECK OPTION的视图中，应当按照WHERE子句的要求设置未被设置的值。

经查，[这篇博客](https://blog.csdn.net/ClearLoveQ/article/details/84285060)中表示，MySQL出于安全性考虑，不允许这种操作，想要插入赵新的信息还是得从基本表下手。

使用如下的语句后，学生表中增加了赵新的信息，同时CS\_STU视图中也能看到赵新了。

|  |  |
| --- | --- |
| INSERT INTO student  SET      Sno = '201215129',      Sname = '赵新',      Ssex = '男',      sdept = '计算机',      Sage = 20; |  |

(5) 更新视图：更新值

|  |
| --- |
| UPDATE CS\_Stu SET Sage=Sage - 1 WHERE Sname = '张三'; |
|  |

此时不论是查询视图还是查询基本表，张三的年龄都已经为18了。而之前是19（在(4)中可见）。

2. 在TPC\_H中实验视图操作

(1) 建立关于供应商的视图

|  |
| --- |
| CREATE VIEW SUPPLIER\_VIEW AS      SELECT          suppkey AS ID, name, address,          nationkey AS nation\_ID, phone      FROM supplier      WITH CHECK OPTION; |
|  |

(2) 查询在lineitem中出现的order所对应的customer.name

|  |
| --- |
| CREATE VIEW LINEITEM\_PART\_VIEW AS      SELECT          partkey AS ID, name, mfgr, type, retailprice      FROM part      WHERE partkey IN      SELECT DISTINCT partkey FROM lineitem      WITH CHECK OPTION; |
|  |

(3) 建立part和supplier详细信息的视图

|  |
| --- |
| CREATE VIEW PART\_SUPPLIER\_VIEW AS      SELECT          part.partkey AS item\_ID,          part.name AS item\_name,          supplier.suppkey AS supplier\_ID,          supplier.name AS supplier\_name,          supplier.address AS supplier\_address      FROM          supplier, part, partsupp      WHERE          supplier.suppkey = partsupp.suppkey          AND part.partkey = partsupp.partkey; |
|  |

(4) 建立关于各个region有多少个nation的视图

|  |  |
| --- | --- |
| CREATE VIEW NATION\_NUM\_VIEW AS      SELECT          regionkey AS region\_ID,          COUNT(nationkey) AS nation\_num      FROM nation      GROUP BY(regionkey); |  |

(5) 更新视图

|  |
| --- |
| UPDATE lineitem\_part\_view SET retailprice = retailprice \* 0.9 WHERE ID <1000; |
| (修改之前) |
| （修改之后） |

由于lineitem\_part\_view视图中只有简单的select子句，所有属性都被select，仅from part这一个基本表，没有group by和having，所以lineitem\_part\_view可以被更新。

四、思考题

1.

用法几乎完全一致，同样支持Select、Insert、Update、Drop等操作，只不过基本表是数据实体，是切实存在存储中的数据表，而视图是查看基本表的一种方法，没有数据实体，只是SQL语句的集合。

2.

修改基本表后，视图的查询结果会根据内容的修改产生对应的修改，但其结果仍然遵照创建视图时的定义。