

视图

前几个小节介绍了SQL的增删改查功能，其中查询是SQL最核心的功能。其实，使用SQL语言来书写查询语句可以类比为使用C语言来编写程序。在编写程序时，为了使程序的整体逻辑更加清晰和提升程序开发效率，程序员会使用函数（Function）来实现程序的模块化设计。通常，重复使用的计算功能会被封装成一个函数，然后在程序的其他部分反复被调用。同样地，SQL也提供了类似函数的功能，它将重复使用的查询结果封装成视图（View），然后在SQL的其他部分直接调用视图。

视图是从一个或几个基本表（或视图）导出的表。但是，它是一张虚表。数据库中只存放视图的定义，而不存放视图对应的数据。当使用视图做查询时，数据库管理系统首先执行视图的SQL查询语句，获得视图的结果之后，在视图的结果之上再执行查询。这和函数的使用是一样的，当调用函数时，程序需要重新执行一遍函数。值得注意的是，一旦基本表中的数据发生变化后，从视图中查询出的数据也将随之改变。

视图一经定义，就可以和基本表一样被查询、被更新、被删除。但是，对视图的更新（增、删、改）操作有一定的限制。

视图的创建

SQL语言用CREATE VIEW命令来创建视图，其基本格式如下：

```
CREATE VIEW <视图名> [ (<列名1> [, <列名2>] ...) ]  
AS <子查询>  
[WITH CHECK OPTION];
```

其中子查询是任意的SELECT查询语句；WITH CHECK OPTION表示对视图进行增、删、改操作时要进行校验，保证更新的元组满足视图的定义。

[例5.31] 创建男生信息的视图。

```
CREATE VIEW M_Student  
AS  
SELECT Sno, Sname, Age  
FROM Student  
WHERE Gender = '男';
```

本例省略了视图M_Student的属性列名，其隐含意思是视图的属性列由子查询中SELECT子句的三个列名组成。关系数据库管理系统执行CREATE VIEW语句时只是把视图的定义存入数据字典，并不执行其中的SELECT语句。只有在对视图查询时，才按视图的定义从基本表中将数据查出。

视图不仅可以建立在一个或者多个基本表上，也可以建立在一个或多个已定义的视图上，或建立在基本表与视图上。

视图的删除

SQL语言用DROP VIEW命令来删除视图，其基本格式为：

```
DROP VIEW <视图名> [CASCADE] ;
```

删除视图其实是将视图的定义从数据字典中删除。如果视图之上还建立了其他视图，那么删除时则需要使用关键词CASCADE，表示将视图和由它导出的所有视图一起删除。

当基本表被删除后，基于基本表创建的所有视图都无法使用，但视图的定义仍存在于数据字典中，此时就需要使用DROP VIEW语句来删除视图。

[例5.31] 删除视图M_Student。
DROP VIEW M_Student CASCADE;

视图的查询

视图创建之后，就可以对视图进行查询，其基本格式与基本表的格式相同。

[例5.32] 查询年龄小于19岁的男生学号和姓名。
SELECT Sno,Sname
FROM M_Student
WHERE Age < 19;

该语句的执行过程是：首先进行有效性检查，检查查询中涉及的视图是否存在。如果存在，则从数据字典中取出视图的定义，把视图定义中的子查询和此查询结合起来，转换成等价的对基本表的新查询，然后执行新查询。

例5.32转换之后的查询语句为：

```
SELECT Sno,Sname  
FROM Student  
WHERE Gender = '男' AND Age < 19;
```

视图的更新

更新视图其实是通过视图来增、删、改数据。由于视图是一张虚表，不存储任何数据，因此对视图的更新最终是转换成对基本表的增、删、改。如果定义视图时加上了WITH CHECK OPTION子句，那么执行更新视图SQL时需要检查更新数据是否满足视图的定义。

修改视图中的元组

[例5.33] 将男生视图M_Student中学号为2022123的学生姓名改为宇轩。
UPDATE M_Student
SET Sname = '宇轩'
WHERE Sno = '2022123';

转换为对基本表更新的SQL为：

```
UPDATE Student  
SET Sname = '宇轩'  
WHERE Sno = '2022123' AND Gender = '男';
```

向视图中插入元组

[例5.34] 向男生视图M_Student中插入一个新的学生信息（学号：2022013，姓名：宇轩，年龄：20）。

```
INSERT  
INTO M_Student  
VALUES('2022013','宇轩', 20);
```

转换为对基本表的插入：

```
INSERT  
INTO M_Student(Sno,Sname,Age,Gender)  
VALUES('2022013','宇轩', 20,'男');
```

删除视图中的元组

[例5.35] 删除男生视图M_Student中学号为2022013的学生信息。

```
DELETE  
FROM M_Student  
WHERE Sno = '2022013';
```

转换为对基本表的删除：

```
DELETE  
FROM M_Student  
WHERE Sno = '2022013' AND Gender = '男';
```

在关系数据库中，不是所有的视图都能被更新。如果对某个视图的增、删、改操作不能转换成对应基本表的增、删、改，那么该视图就不能被更新。

视图除了使查询语句SQL的逻辑更加清晰，简化用户的SQL编写之外，还能实现用户的权限访问功能。