

第六章（2）：视图

- 视图是从一个或者多个表及其他视图中通过select语句导出的虚拟表，数据库中只存放了视图的定义，而并没有存放视图中的数据。浏览视图时所对应数据的行和列数据来自定义视图查询所引用的表，并且在引用视图时动态生成。通过视图可以实现对基表数据的查询与修改。
- 视图为数据库用户提供了很多的便利，主要包括以下几个方面。
 - (1) 简化数据查询和处理。视图可以为用户集中多个表中的数据，简化用户对数据的查询和处理。
 - (2) 屏蔽数据库的复杂性。数据库表的更改不影响用户对数据库的使用，用户也不必了解复杂的数据库中的表结构。例如，那些定义了若干张表连接的视图，就将表与表之间的连接操作对用户隐蔽起来了。
 - (3) 安全性。如果想要使用户只能查询或修改用户有权限访问的数据，也可以只授予用户访问视图的权限，而不授予访问表的权限，这样就提高了数据库的安全性。

6.1：创建视图

1. 创建视图

- 创建视图是指在指定的数据库表上建立视图。视图可以建立在一张表上，也可以建立在多张表或既有视图上。要求创建用户具有针对视图的create view权限，以及针对由select语句选择的每一列上的某些权限。创建视图的语法形式。语法形式如下：

```
create [or replace][algorithm = { undefined|merge|temptable }  
view view_name [(column_list)]  
as select_statement  
[ with [ cascaded|local ] check option ] ;
```

- 创建视图有如下限制：
 - 在视图的from子句中不能使用子查询。
 - 在视图的select语句不能引用系统或用户变量。
 - 在视图的select语句不能引用预处理语句参数。
 - 在视图定义中允许使用order by，但是，如果从特定视图进行了选择，而该视图使用了具有自己order by的语句，它将被忽略。

■在定义中引用的表或视图必须存在。但是，创建了视图后，能够舍弃定义引用的表或视图。要想检查视图定义是否存在这类问题，可使用check table语句。
■在定义中不能引用temporary表，不能创建temporary视图。
■不能将触发程序与视图关联在一起。

- 在单表上创建视图。MySQL中可以在单个表上创建视图。
【例1】在teacher表上创建一个简单的视图，视图名称为teach_view1。

```
mysql> create view teach_view1  
-> as select * from teacher;
```

可以利用select语句查询视图teach_view1的数据如下。

```
mysql> select * from teach_view1;
```

- 在多表上创建视图。MySQL数据库中也可以在两个或两个以上的表上创建视图。

【例2】在student表、course表和score表上创建一个名为stu_score1的视图。视图中保留18级的女生的学号、姓名、电话、课程名和期末成绩。

```
mysql> create view stu_score1
-> as select student.studentno, sname, phone, cname, final
-> from score join student on student.studentno=score.studentno
-> join course on course.courseno=score.courseno
-> where sex='女' and left(student.studentno,2)= '18';
```

此视图保存在3个表的数据，可以利用select语句查询视图stu_score1的数据如下。

```
mysql> select * from stu_score1;
```

- 在已存在的视图上创建视图

【例3】创建视图teach_view2，统计计算机学院的教师中的教授和副教授的教师号、教师名和专业。

```
mysql> create view teach_view2
-> as select teacherno, tname, major
-> from teach_view1
-> where prof like '%教授' and department='计算机学院';
```

说明：

(1) 定义视图时基本表可以是当前数据库的表，也可以来自于另外一数据库的基本表，只要在表名前添加数据库名称即可，如mysql.student02。

(2) 定义视图时可在视图名后面指明视图列的名称，名称之间用逗号分隔，但列数要与select语句检索的列数相等。例如，定义视图teach_view2可以写成如下方式：

```
create view teach_view2(教师号,教师名,专业)
as select teacherno, tname, major .....
```

(3) 使用视图查询时，若其基本表中添加了新字段，则该视图将不包含新字段。

(4) 如果与视图相关联的表或视图被删除，则该视图将不能再使用。

6.2：查看视图的定义

- 查看视图是指查看数据库中已存在的视图的定义。查看视图必须要有show view的权限，MySQL数据库下的user表中保存着这个信息。
- 查看视图的方法包括describe语句、show table status语句、show create view语句和查询information_schema数据库下的views表等。

(1) describe语句查看视图基本信息。可以使用describe语句可以查看表的基本定义。同样可以使用describe语句可以用来查看视图的基本定义。describe语句查看视图的基本形式与查看表的形式是一样的。

(2) 利用show table status语句查看视图基本信息。MySQL中，可以使用show table status语句来查看视图的信息。其语法形式如下：

```
show table status like 'view_name';
```

(3) 利用show create view语句查看视图详细信息。MySQL中，show create view语句可以查看视图的详细定义。语法形式如下

```
show create view view_name
```

(4) 在views表中查看视图详细信息。MySQL数据库中，所有视图的定义都存在information_schema数据库下的views表中。例如，查询information_schema.views表，可以查看到数据库中所有视图的详细信息。代码如下：

- select * from information_schema.views;

其中，*表示查询所有的列的信息；information_schema.views表示information_schema数据库下面的views表。

6.3：修改视图

- 修改视图是指修改数据库中已存在的表的定义。当基本表的某些字段发生改变时，可以通过修改视图来保持视图和基本表之间一致。MySQL中通过create or replace view语句和alter语句来修改视图。
- MySQL中，create or replace view语句可以用来修改视图。该语句的使用非常灵活。在视图已经存在的情况下，对视图进行修改；视图不存在时，可以创建视图。

create or replace view语句的语法形式如下：

在MySQL中，alter语句可以修改表的定义，可以创建索引。不仅如此，alter语句还可以用来修改视图。

alter语句修改视图的语法格式如下：

```
alter [algorithm = {undefined|merge|temptable}]  
view view_name [(column_list)]  
as select语句  
[with [cascaded|local]check option];
```

【例4】修改视图teach_view2，统计计算机学院和材料学院的教师中教授和副教授的教师号、教师名、和专业，并在视图名后面指明视图列名称。

```
mysql> alter view teach_view2(教师号,教师名,专业)  
-> as select teacherno, tname, major  
-> from teach_view1  
-> where prof like '%教授'  
-> and (department='计算机学院' or department='材料学院');
```

可以通过用select语句查看视图teach_view2的数据如下。

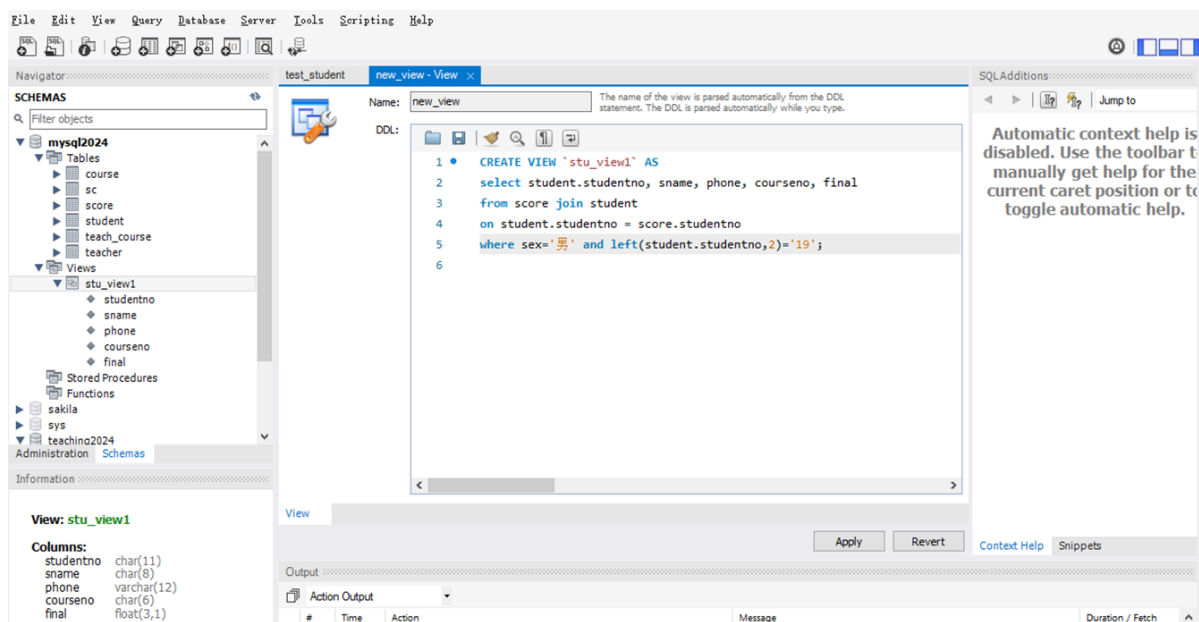
```
mysql> select * from teach_view2;
```

6.4：利用MySQL Workbench工具创建和管理视图

4. 利用MySQL Workbench工具创建和管理视图

- 利用MySQL Workbench创建视图

- (1) 启动MySQL Workbench工具，单击选择当前数据库mysql2024。
- (2) 在mysql2024数据库中选择views，展开views选项，可以看到已经创建的视图，执行create view命令。
- (3) 在文本编辑区，按照如图所示输入创建视图stu_view1的内容。实现以student表和score表为基表的视图中保留19级的男生的学号、姓名、电话、课程号和期末成绩的功能。



- (4) 自己检查无误后，单击Apply按钮，进入如图所示的代码对话框中，这是要向数据库Mysql2024中存储的脚本。

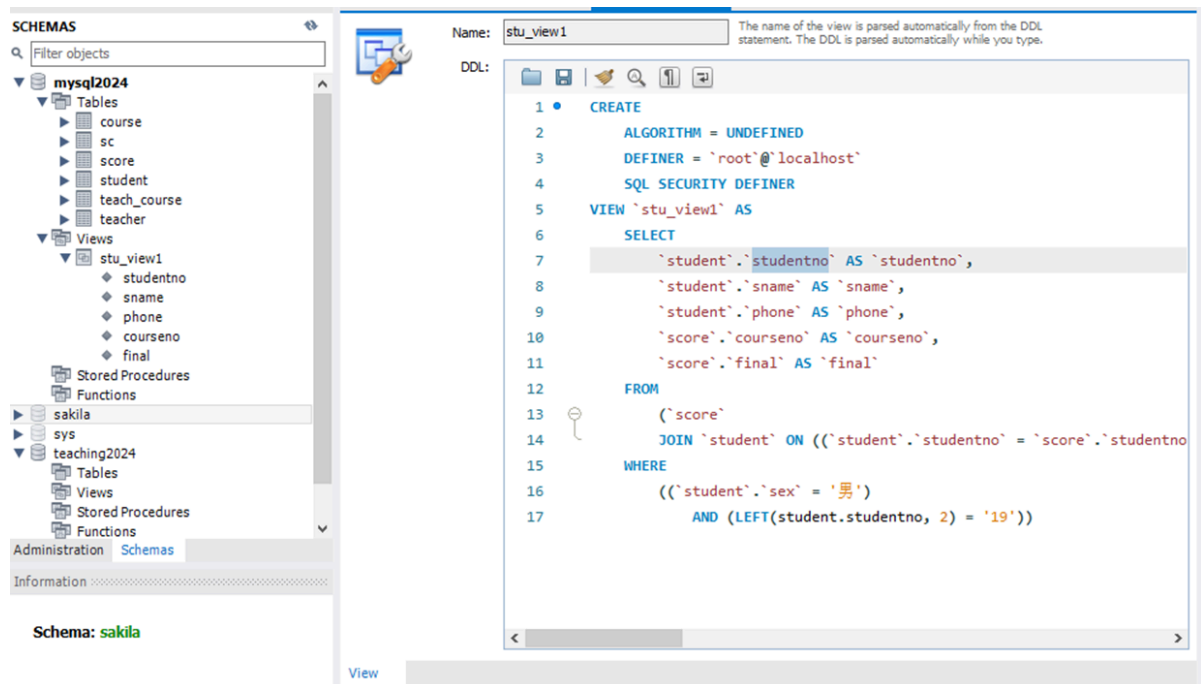
- (5) 单击Apply按钮，可以通过Show logs (Hide logs) 转换按钮查看信息记录 (Message Log) 窗口中的信息。

可以查看到成功创建视图的提示：SQL script was successfully applied to the database。单击finish按钮完成视图创建过程。

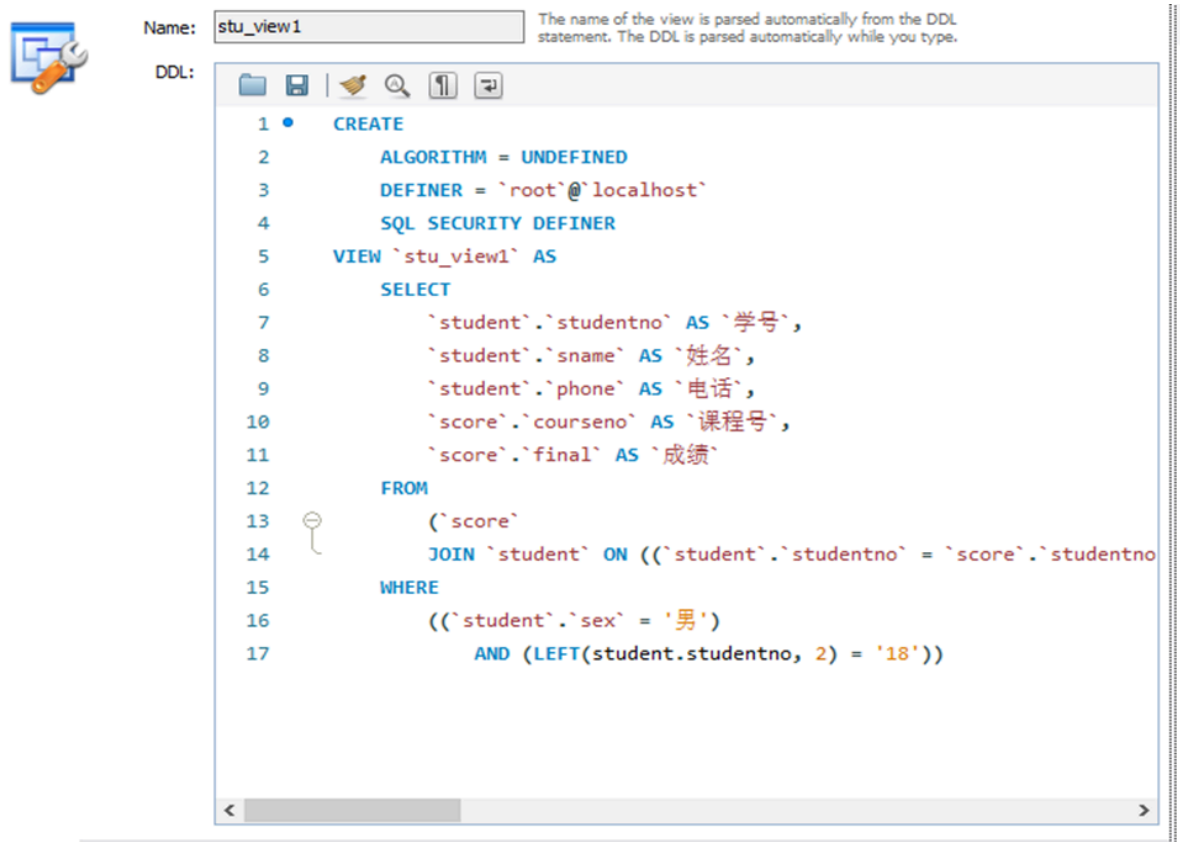
- (6) 在数据库Mysql2024中展开view文件夹，找到视图tru_view1，执行Select Rows-Limit 10命令，即可看到视图tru_view1的查询结果。即含有19级的男生的学号、姓名、电话、课程号和期末成绩的结果集。

- 利用MySQL Workbench修改视图

- (1) 在数据库mysql2024中展开文件夹，右击视图tru_view1，执行Alter view...命令，进入如图所示的修改对话框中。



(2) 如图所示。输入修改项，例如，视图名改为stu_view2、各个视图列的输出名称改为汉语标识，过滤条件改为18级



(3) 依次对话框中的单击Apply按钮和finish按钮即可完成视图修改过程。

(4) 在数据库Mysql2024中展开view文件夹，找到视图stu_view2，执行Select Rows-Limit 10命令，即可看到视图stu_view2的查询结果。即含有18级的男生的学号、姓名、电话、课程号和期末成绩的结果集。

test_student new_view - View stu_view1 - View stu_view1 stu_view2 - View **stu_view2** x

Limit to 1000 rows

1 • `SELECT * FROM mysql2024.stu_view2;`

学号	姓名	电话	课程号	成绩
18125111109	敬横江	15678945623	c05138	80.0
18137221508	赵临江	12367823453	3	99.0

6.5：删除视图

5.删除视图

- 删除视图是指删除数据库中已存在的视图。删除视图时，只能删除视图的定义，不会删除数据。MySQL数据库中，用户必须拥有drop权限才能使用drop view语句来删除视图。
- 对需要删除的视图，使用drop view语句进行删除。drop view命令可以删除多个视图，各视图名之间用逗号分隔。基本格式如下：
`drop view [if exists]视图名列表 [restrict|cascaded]`
 例如，删除视图V1_student命令如下：
`drop view V1_student;`
 如果在MySQL Workbench中删除视图，只要右击要删除的视图，执行Drop view命令，按照操作提示，就可以完成。
- 视图的使用主要包括视图的检索，以及通过视图对基表进行插入、修改、删除操作。视图的检索几乎没有什么限制，但是对通过视图实现表的插入、修改、删除操作则有一定的限制条件。

1.使用视图管理表数据

- 使用视图进行查询。使用视图进行查询实际上就是把视图作为数据源，实现查询功能。

【例5】通过视图stu_view2，查询选修课程号为c08123、且成绩在80分以上的18级男生学生的学号、课程号和成绩。

```
mysql> select 学号, 姓名,课程号,成绩
-> from stu_view2
-> where 课程号='c08123' and 成绩 > 80;
```

- 使用视图进行统计计算

【例6】创建视图course_avg, 统计各门课程平均成绩, 并按课程名称降序排列。

```
mysql> create view course_avg  
-> as select cname 课程名, avg(final) 平均成绩  
-> from score join course on score.courseno= course.courseno  
-> group by cname desc;  
mysql> select * from course_avg;
```

- 使用视图修改基本表数据。使用视图修改表数据是指在视图中进行insert、update和delete等操作而修改基表的数据。通过视图修改表数据时, 要有执行相关操作的权限。

【例7】通过视图teach_view1, 对基表teacher进行插入、更新和删除数据的操作。

```
mysql> insert into teach_view1(teacherno,tname,major,prof,department)  
-> values ('t06027', '陶期年', '纳米技术', '教授', '材料学院');
```

```
mysql> update teach_view1 set prof = '副教授' where teacherno = 't07019';  
mysql> delete from teach_view1 where teacherno = 't08017';  
使用select语句查询teacher表, 可以看到基表中的数据也相应地进行了修改。  
mysql> select * from teacher;
```

【例8】视图stu_score1依赖于源表student、course和score等3张表, 包括studentno、sname、phone、cname和final等5个字段, 通过stu_score1修改基本表student中的字号为18125121107的电话号码。

```
mysql> update stu_score1 set phone='132123456777'  
-> where studentno = '18125121107';
```

通过查看student 表, 可以看到相应成绩已做了更改。

```
mysql> select studentno,sname, phone from student  
-> where studentno = '18125121107';
```

说明:

- (1) 视图若只依赖于一个基表, 则可以直接通过更新视图来更新基本表的数据。
- (2) 若一个视图依赖于多张基表, 则一次只能修改一个基表的数据, 不能同时修改多个基本表的数据。
- (3) 如果视图包含下述结构中的任何一种, 都是不可修改的:
 - 视图的列含有聚合函数 (avg、count、sum、min、max) 。
 - 视图的列是通过表达式并使用列计算出其他列。
 - 含有distinct关键字。
 - 含有group by子句、order by子句、having子句。
 - 含有union运算符。
 - 视图的列位于选择列表中的子查询。
 - from子句中包含多个表。
 - select语句中引用了不可更新视图。
 - where子句中的子查询, 引用from子句中的表。