

【MySQL】视图(十)

🚗 MySQL学习·第十站~
📌 本文已收录至专栏：[MySQL通关路](#)
❤️ 文末附全文思维导图，感谢各位点赞收藏支持~
★ 学习汇总贴，超详细思维导图：[【MySQL】学习汇总\(完整思维导图\)](#)

一.引入

视图 (View) 是一种虚拟存在的表。视图中的数据并不在数据库中实际存在，行和列数据来自创建视图的查询中使用的表(基表)，并且是在使用视图时动态生成的。也就是说,视图其本身只是一段查询的SQL逻辑。

库表原始数据
(基表)

id	name	phone	email	profession	age	gender	status	createtime
1	吕布	17799990000	lvbu666@163.com	软件工程	23	1	6	2001-02-02 00:00:00
2	吕布	17799990000	lvbu666@163.com	软件工程	23	1	6	2001-02-02 00:00:00
3	曹操	17799990001	caocao666@qq.com	通讯工程	33	1	0	2001-03-05 00:00:00
4	赵云	17799990002	17799990@139.com	英语	34	1	2	2002-03-02 00:00:00
5	孙悟空	17799990003	17799990@sina.com	工程造价	54	1	0	2001-07-02 00:00:00
6	花木兰	17799990004	19980729@sina.com	软件工程	23	2	1	2001-04-22 00:00:00
7	大乔	17799990005	daqiao666@sina.com	舞蹈	22	2	0	2001-02-07 00:00:00
8	露娜	17799990006	luna love@sina.com	应用数学	24	2	0	2001-02-08 00:00:00

数据为创建视图的SQL语句映射，
本身并不存在，随基表动态变化

视图

id	name	age	gender	status
3	曹操	33	1	0
4	赵云	34	1	2
5	孙悟空	54	1	0

基于
SELECT id,name,age,gender,status From user WHERE age > 30;
所生成的视图

CSDN @观止study

视图也是一张数据库表，当我们创建以后，也可以像操作正常数据库一样操作它。其用途：

- 可以简化用户对数据的理解 and 操作。那些被经常使用的复杂多表查询可以被定义为视图，从而避免以后的操作每次都需指定全部的条件。
- 数据库可以授权，但不能授权到数据库特定行和特定的列上。配合视图使用可以限制用户只能查询和修改他们所能见到的数据
- 视图可帮助用户屏蔽真实表结构变化带来的影响。

二.相关操作

(1) 创建视图

- 语法

```
CREATE [OR REPLACE] VIEW 视图名称[(列名列表)] AS SELECT语句 [ WITH [ CASCADED | LOCAL ] CHECK OPTION ]
```

```

studentmessage
study
  > 表
  > 视图
    > show_user
  > 函数
  > 事件
12 -- 切换数据库
13 use study;
14
15 -- 创建视图
16 CREATE VIEW show_user AS
17 SELECT id,name,age,gender,status From user WHERE age > 30
18

```

(2) 操作数据

视图也是一张表，我们可以像操作正常表一样操作视图。

- 例如查询视图所有数据

```

19 -- 查询数据
20 select * from show_user;
21
22
23

```

信息	结果1	概况	状态	
id	name	age	gender	status
3	曹操	33	1	0
4	赵云	34	1	2
5	孙悟空	54	1	0

(3) 修改视图

- 语法

```

-- 方式一：
CREATE OR REPLACE VIEW 视图名称[(列名列表)] AS SELECT语句 [ WITH [ CASCADED | LOCAL ] CHECK OPTION ]
-- CREATE OR REPLACE 不存在则创建， 存在则替换

-- 方式二：
ALTER VIEW 视图名称[(列名列表)] AS SELECT语句 [ WITH [ CASCADED | LOCAL ] CHECK OPTION ]

```

- 方式一

```

-- 方式一
CREATE OR REPLACE VIEW show_user AS
SELECT id,name,age,gender,status From user WHERE gender = 1

select * from show_user;

```

结果1	概况	状态	
name	age	gender	status
1 吕布	23	1	6
2 吕布	23	1	6
3 曹操	33	1	0
4 赵云	34	1	2
5 孙悟空	54	1	0

- 方式二

```

5 -- 方式二
6 ALTER VIEW show_user AS
7 SELECT id,name,age,gender,status From user WHERE status < 3
8
9 select * from show_user;
10
11
12
13
14
15
16

```

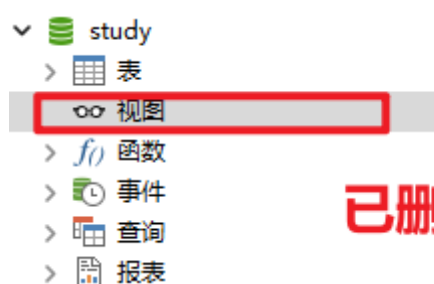
结果1	概况	状态
name	age	gender
3 曹操	33	1
4 赵云	34	1
5 孙悟空	54	1
6 花木兰	23	2
7 大乔	22	2
8 露娜	24	2

CSDN @观止study

(4) 删除视图

- 语法

DROP VIEW [**IF EXISTS**] 视图名称 [,视图名称]
 -- [,视图名称] 若同时删除多张视图，以逗号分割



```

20 select * from show_user;
21
22 -- 删除视图
23 drop view show_user
24
25
26
27

```

CSDN @观止study

(5) 查看创建视图语句

- 语法

SHOW CREATE VIEW 视图名称

- 除了我们所指定的参数，还有一堆默认配置

```

mysql> SHOW CREATE VIEW show_user;
+-----+-----+-----+-----+-----+
View | Create View | character_set_client | collation_connection |
+-----+-----+-----+-----+-----+
show_user | CREATE ALGORITHM=UNDEFINED DEFINER='root'@'localhost' SQL SECURITY DEFINER VIEW `show_user` AS select `user`.`id` AS `id`,`user`.`name` AS `name`,`user`.`age` AS `age`,`user`.`gender` AS `gender`,`user`.`status` AS `status` from `user` where (`user`.`status` < 3) | utf8mb4 | utf8mb4_0900_ai_ci |
+-----+-----+-----+-----+-----+
row in set (0.00 sec)

```

CSDN @观止study

三.检查选项

(1) 概述

由于视图是虚拟存在的表，我们对视图的操作都会反应到基表当中，假设我们对视图表插入不符合视图where条件的数据，那么这条数据只会存在于基表当中，而我们在视图表无法获悉，这岂不是插入了一条无效数据？

```

36 -- 创建视图
37 CREATE VIEW show_user AS
38 SELECT id,name,age,gender,status From user WHERE age > 30
39
40 -- 插入数据
41 insert into show_user (name,age,gender,status) VALUES('王煊',28,1,0);
42
43 -- 查询比对
44 SELECT * from show_user;
45
46 SELECT * from user;
47
48

```

插入数据

我们插入的垃圾数据，由于where条件的限制，在视图无法展示，而在基表却存在

id	name	phone	email	profession	age	gender	status
1	吕布	17799990000	lvbu666@163.com	软件工程	23	1	6
2	吕布	17799990000	lvbu666@163.com	软件工程	23	1	6
3	曹操	17799990001	caocao666@qq.com	通讯工程	33	1	0
4	赵云	17799990002	17799990@139.com	英语	34	1	2
5	孙悟空	17799990003	17799990@sina.com	工程造价	54	1	0
6	花木兰	17799990004	19980729@sina.com	软件工程	23	2	1
7	大乔	17799990005	daqiao666@sina.com	舞蹈	22	2	0
8	露娜	17799990006	luna_love@sina.com	应用数学	24	2	0
10	王煊	(Null)	(Null)	(Null)	28	1	0

在我们创建或修改视图的语句中可以看到 **WITH CHECK OPTION** 选项，MySQL会通过视图检查正在更改的每个行，例如插入，更新，删除，对于满足条件（例如where条件）的操作，我们允许，否则禁止，以使其符合视图的定义。由于MySQL**允许基于另一个视图创建视图**，因此它还会检查依赖视图中的规则以保持一致性。为了确定检查的范围，mysql提供了两个选项：**CASCADED** 和 **LOCAL**，默认值为 **CASCADED**。

(2) CASCADED级联

假设v2视图是基于v1视图的，如果在v2视图创建的时候指定了检查选项为 **cascaded**，v1视图创建时未指定检查选项（如果指定了则继续检查上一级，以此往复）。则在对v2进行操作执行检查时，不仅会检查v2，还会级联检查v2的关联视图v1，如果不满足条件则无法进行相关操作。

如果指定了需要继续检查上一级

```
create view v1 as select id,name from student where id <= 20 with cascaded check option ;
```



必须同时满足两个限制条件

```
create view v2 as select id , name from v1 where id >= 10 with cascaded check option ;
```

- u2 插入失败示例

```

50 -- 创建视图1
51 CREATE VIEW u1 AS
52 SELECT id,name,age From user WHERE age < 30
53
54
55 -- 创建视图2
56 CREATE VIEW u2 AS
57 SELECT id,name,age From u1 WHERE age > 20 with CASCADED CHECK OPTION;
58
59 -- 插入
60 insert into u2 (name,age) values('楚风',31)
61

```

虽然插入的31满足u2的where,但是由于
cascaded检查, 该数据不满足u1的where
插入失败 因此被拒绝插入

信息 概况 状态

[SQL]insert into u2 (name,age) values('楚风',31)

[Err] 1369 - CHECK OPTION failed 'study.u2'

CSDN @观止study

- u2 插入成功

```

50 -- 创建视图1
51 CREATE VIEW u1 AS
52 SELECT id,name,age From user WHERE age < 30
53
54
55 -- 创建视图2
56 CREATE VIEW u2 AS
57 SELECT id,name,age From u1 WHERE age > 20 with CASCADED CHECK OPTION;
58
59 -- 插入
60 insert into u2 (name,age) values('楚风',21)
61

```

21同时满足u1与u2

信息 概况 状态

[SQL]insert into u2 (name,age) values('楚风',21)

受影响的行: 1
时间: 0.002s

插入成功

CSDN @观止study

- u1 插入情况

```

50 -- 创建视图1
51 CREATE VIEW u1 AS
52 SELECT id,name,age From user WHERE age < 30
53
54
55 -- 创建视图2
56 CREATE VIEW u2 AS
57 SELECT id,name,age From u1 WHERE age > 20 with CASCADED CHECK OPTION;
58
59 -- 插入
60 insert into u1 (name,age) values('楚风',11)
61

```

由于u1没有检查, 因此可以随意插入数据

信息 概况 状态

[SQL]insert into u1 (name,age) values('楚风',11)

受影响的行: 1
时间: 0.002s

不需要检查u2

```

59 -- 插入
60 insert into u1 (name,age) values('楚风',31)
61

```

信息 概况 状态

[SQL]insert into u1 (name,age) values('楚风',31)

受影响的行: 1
时间: 0.004s

也不需要检查自身条件

CSDN @观止study

(3) LOCAL本地

与上述类似，LOCAL也会递归的去寻找当前视图所依赖的视图。不同的是，如果当前视图或者依赖的视图后面定义了 `with` `check option` 检查选项才会校验我们操作的数据是否满足该视图的条件（where），否则不会校验。

未添加检查选择，不检查

```
create view v1 as select id,name from student where id <= 15
```

检查当前条件

```
create view v2 as select id , name from v1 where id >= 10
```

`with local check option ;`

CSDN @观止study

- 依赖视图未定义检查选项

```
-- 创建视图1
51 CREATE VIEW u1 AS
52 SELECT id,name,age From user WHERE age < 30
53
54
55 -- 创建视图2
56 CREATE VIEW u2 AS
57 SELECT id,name,age From u1 WHERE age > 20 with LOCAL CHECK OPTION;
58
59 -- 插入
60 insert into u2 (name,age) values ('楚风',31)
61
```

只检查添加了检查选择的视图的条件

信息 概况 状态

[SQL]insert into u2 (name,age) values('楚风',31)

受影响的行: 1
时间: 0.003s

插入成功

```
58
59 -- 插入
60 insert into u2 (name,age) values ('楚风',11)
61
```

信息 概况 状态

[SQL]insert into u2 (name,age) values('楚风',11)

[Err] 1369 - CHECK OPTION failed 'study.u2'

插入失败

CSDN @观止study

- 依赖视图定义了检查选项

```
48 -- 创建视图1
49 CREATE VIEW u1 AS
50 SELECT id,name,age From user WHERE age < 30 with LOCAL CHECK OPTION;
51
52
53 -- 创建视图2
54 CREATE VIEW u2 AS
55 SELECT id,name,age From u1 WHERE age > 20 with LOCAL CHECK OPTION;
56
57 -- 插入
58 insert into u2 (name,age) values ('楚风',31)
59
60 -- 查询视图
61 SELECT * from u1;
```

检查添加检查选项的视图的条件

信息 概况 状态

[SQL]insert into u2 (name,age) values('楚风',31)

[Err] 1369 - CHECK OPTION failed 'study.u2'

插入失败

```
57 -- 插入
58 insert into u2 (name,age) values ('楚风',21)
59
60 -- 查询视图
61 SELECT * from u1;
```

信息 概况 状态

[SQL]insert into u2 (name,age) values('楚风',21)

受影响的行: 1
时间: 0.003s

同时满足，插入成功 CSDN @观止study

四.视图更新条件

要使视图可更新，**视图中的行与基础表中的行之间必须存在一对一的关系**。如果视图包含以下任何一项，则该视图不可更新：

- 聚合函数或窗口函数（SUM()、MIN()、MAX()、COUNT()等）
- DISTINCT
- GROUP BY
- HAVING
- UNION 或者 UNION ALL

例如我们创建视图时使用了聚合函数，破坏了一对一关系

```
create view stu_v_count as select count(*) from student;
```

如果我们对这个视图进行更新或插入的，将会报错。

```
insert into stu_v_count values(10);
```

```
[HY000][1471] The target table stu_v_count of the INSERT is not insertable-into
```

五.全文概览

