**mysql视图（虚拟的表，假表）**

视图概述

视图介绍

什么是视图

虚拟表

内容与真实的表相似，有字段有记录

视图并不在数据库中以存储的数据形式存在

行和列的数据来自定义视图时查询所引用的基表，并且在具体引用视图时动态生成

更新视图的数据，就是更新基表的数据

更新基表的数据，视图的数据也会跟着改变

视图优点

简单

用户不需关心视图中的数据如何查询获得

视图中的数据已经是过滤好的符合条件的结果集

安全

用户只能看到视图中的数据

数据独立

一旦视图结构确定，可以屏蔽表结构对用户的影响

视图使用限制

不能在视图上创建索引

在视图的from字句中不能使用子查询

以下情形中的视图是不可更新的

包含以下关键字的sql语句：聚合函数（SUM、MIN、MAX、COUNT等）、DISTINCT、GROUP BY、HAVING、UNION或UNION ALL

常量视图、JOIN、FROM一个不能更新的视图

where字句的子查询引用了FROM字句中的表

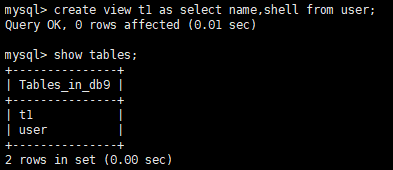
使用了临时表

视图基本使用

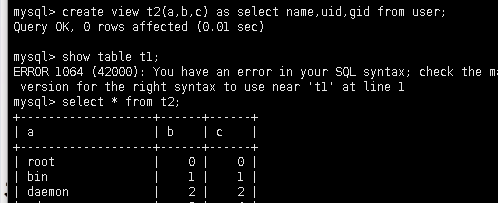
创建视图

语法格式

create view 视图名称 as sql查询;



create view 视图名称(字段列表) as sql查询;

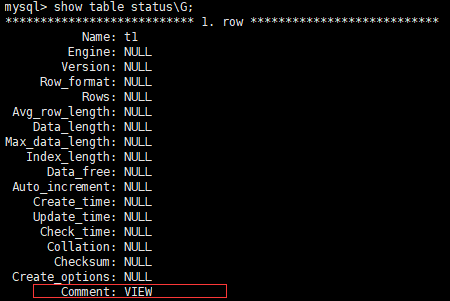


在视图中不定义字段名的话，默认使用基表的字段名，若定义字段名的话，视图中的字段必须和基表的字段个数相等。

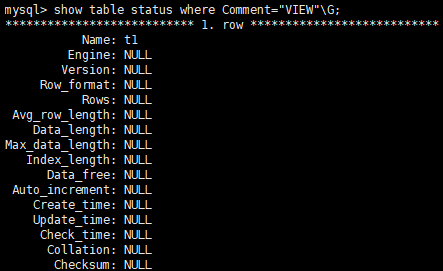
查看视图

查看当前库下所有表的状态（查看是否为视图）

show table status;

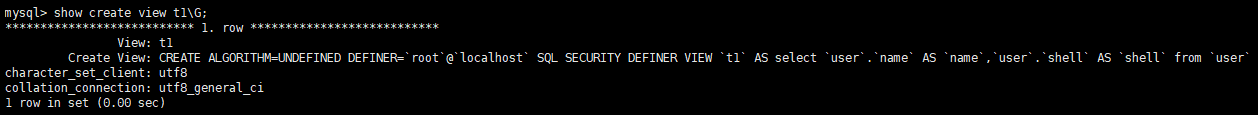


show table status where comment=”view”\G;



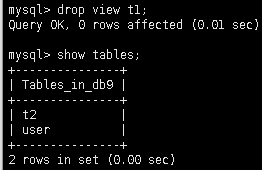
查看创建视图具体命令（查看基于哪个基表）

show create view 视图名;



删除视图

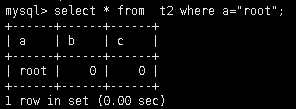
drop view 视图名;



使用视图

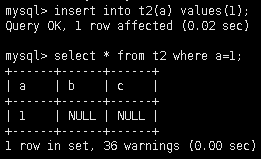
查询记录

select 字段名列表 from 视图名 where条件;



插入记录

insert into 视图名(字段名列表) values(字段值列表);



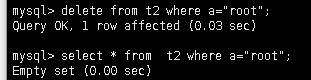
更新记录

update 视图名set 字段名=值 where 条件;



删除记录

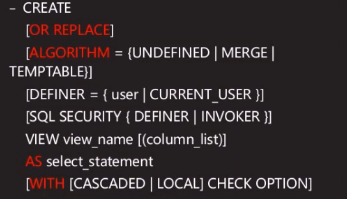
delete from 视图名 where 条件;



视图表修改记录，基表的记录也会更改，基表记录更改，视图表的记录也会更改

视图进阶

创建视图的完整格式



设置字段别名

视图中的字段名不可以重复 所以要定义别名

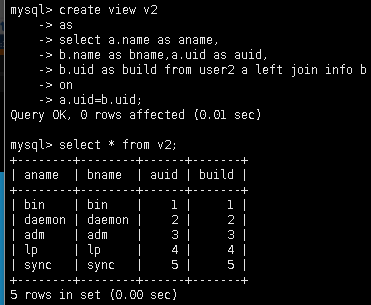
create view 视图名

as

select 表别名.源字段名 as 字段别名

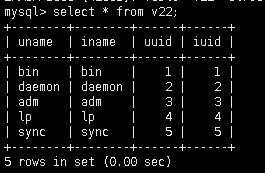
from 源表名 表别名 left join 源别名 表别名

on条件;

 给表和字段起别名



只给字段起别名



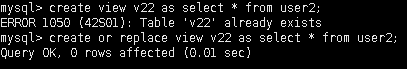
关联视图查询建的视图 默认不允许修改视图字段的值

重要选项说明

or replace

create or replace view 视图名 as select 查询;

创建时，若视图已经存在，会替换已有的视图



algorithm（视图的执行方式）

algorithm = {undefined | merge | temptable}

merage,替换方式

视图名直接使用视图的公式替换掉，把视图公式合并到了select中。



temptable,具体化方式

先得到视图的执行结果，该结果形成一个中间结果暂时存在内存中之后，外面的select语句就调用了这些中间结果

undefined，未定义

algorithm选项的值是undefined表示使用的是merage替换的方式。

definer={user|current\_user}//指定视图的创建者

sql security（指定视图查询数据时的安全验证方式）

definer表示按定义者拥有的权限来执行

invoker表示用调用者的权限来执行。默认情况下，系统指定为definer

任意用户访问view时，能否成功取决于是否有调用该view的权限，以及definer是否拥有view中的select的权限，只需要修改创建同名用户或者修改definer即可。

with [] check option（作用是定义对视图表里的数据做操作时的限制方式）

当视图是根据另一个视图定义时，对视图更新/删除/插入

local和cascaded关键字决定检查的范围

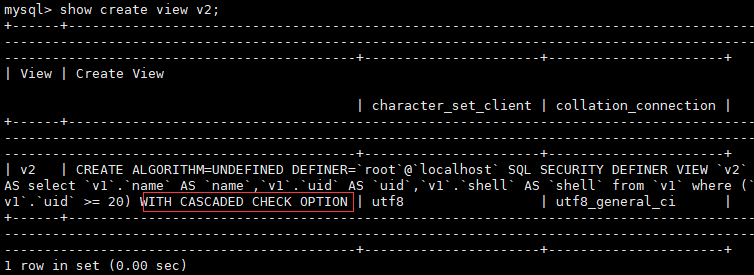
local仅检查当前视图的限制（满足视图本身的限制即可）

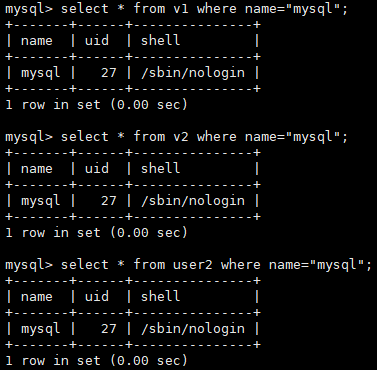
cascaded同时要满足基表的限制（默认值，要同时满足基表的限制，让改但是数据没有了，改的是最原始的基表）

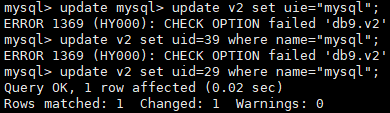
create view v2 select \* from user2 limit 3;不能修改数据，因为修改了系统就不认为是数据是前3行了。

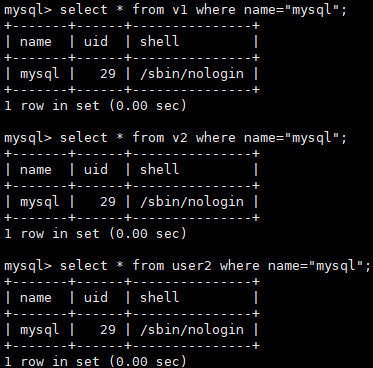




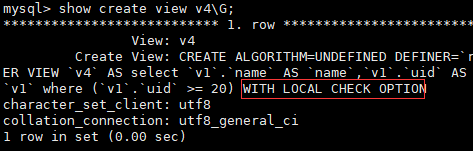


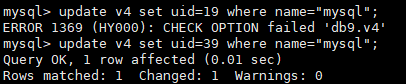


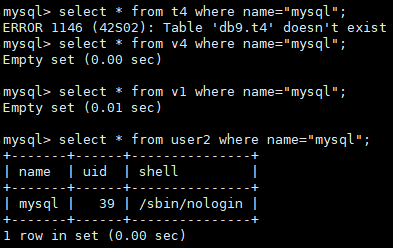












因为视图v4表是基于v1表创建的，39满足v4的建表条件，所以同意修改，但是不满足v1视图表的建表条件，所以在v1中删除此数据，v4也不记录，因为user2是v1的基表，满足条件，所以user2物理表的值被修改了。