

简化很长的select语句。假设下面的语句经常执行，每次都写挺麻烦的，所以提出了将这条select语句保存的解决方案，即将这条select保存起来，**保存了这条select语句的虚拟表(临时表)**，就称为“视图”。这样再次查询相关内存就会得到简化。

```
select staff.sid, staff.name, staff.sex, staff.age, market.month, market.sales  
from staff  
inner join market  
on staff.sid = market.sid;
```

引出

视图是保存某条select语句的临时表(虚拟表)。

视图名就是某条select语句的别名。

在创建视图时，是将某条select语句保存起来，而非将select语句的查询结果保存起来。

定义

语法 `create view 视图名 as 查询语句;`

举例 子主题

聚焦特定数据 比如某一个表的列非常多，而我们一般只会用到某几个列的数据，此时就可以使用视图来聚焦特定的数据，为用户定制数据。

提高重用性 如果一个查询操作经常被使用，并且select语句本身又长又复杂(比如使用了很多聚合函数、关联了其他表等)，那么我们可以将其保存成一个视图。

提高安全性 对于一些不能被修改的重要字段，如果我们不希望其被用户误操作，那么可以使用视图只显示一些不重要的字段，而把那些重要的字段隐藏起来。

与表的区别

注意，视图和原表共享同一份数据，两者之间的更改会相互影响。

在**使用union、Inner join、子查询的视图**中不能执行insert\update这两种操作。

特例 而**使用普通select语句**的视图是允许执行insert和update的

update `update 视图名 set 列名 = 新值;`

```
insert into fruit_v1  
values(12, "樱桃", "核果", "夏", 84.5, '2022-07-12');
```

分类 insert

当往视图中插入数据时，即使不符合 where 条件，数据也会被直接插入原表中。

特例

解决方案 **with check option**

`create view fruit_v2`

`as select name, price from fruit where price < 10 ;`

`with check option`

delete `delete from fruit_v where name='草莓';`

包含聚合函数的视图。

包含子查询的视图。

包含 distinct、group by、having、union 等关键字的视图。

由不可更新的视图所创建的视图，

MySQL8 之视图

说明

在mysql中，并不是所有的视图都是可以修改数据的，比如以下就不行

`select 语句不能包含 from 子句中的子查询。`

`select语句不能引用系统变量或用户变量。`

不是所有的select就可以保存成视图

`select语句不能引用预处理语句参数。`

嵌套性 视图可以嵌套，也就是**基于一个视图去建立另一个视图**。

无索引性 **不能给视图建立索引，也不能有相关的触发器(因为视图本身是没有数据的)。**

费性能 视图的个数没有限制，但是过多的视图会影响 MySQL的性能。

连接性 视图可以和表一起使用，比如连接表和视图的 select语句。

--方式1
`describe 视图名;`

--方式2
`show table status like '视图名';`

-- 方式3
`show create view 视图名;`

alter view 视图名
as 查询语句;

`create or replace view 视图名
as 查询语句;`

create or replace view

较上者强大之处 存在则覆盖，不存在则创建

修改视图

语法 `drop view 视图名;`

除了单表视图之外，还有一种多表视图。多表视图本质上是**连接多个表并选取若干列来创建一个视图**，

定义 所以多表视图肯定涉及多表查询。

```
create view v1 as select staff.name, market.month, market.sales  
from staff  
inner join market
```

```
on staff.sid = market.sid;
```

多表视图