



Python入门+爬虫+数据分析+办公自动化+永久学习

免费赠：大厂导师答疑+50GPython学习大礼包+5G面试礼包

[立即领取 >](#)

MySQL视图是什么？

[< 上一节](#)[下一节 >](#)

MySQL 视图 (View) 是一种虚拟存在的表，同真实表一样，视图也由列和行构成，但视图并不实际存在于数据库中。行和列的数据来自于定义视图的查询中所使用的表，并且还是在使用视图时动态生成的。

数据库中只存放了视图的定义，并没有存放视图中的数据，这些数据都存放在定义视图查询所引用的真实表中。使用视图查询数据时，数据库会从真实表中取出对应的数据。因此，视图中的数据是依赖于真实表中的数据的。一旦真实表中的数据发生改变，显示在视图中的数据也会发生改变。

视图可以从原有的表上选取对用户有用的信息，那些对用户没用，或者用户没有权限了解的信息，都可以直接屏蔽掉，作用类似于筛选。这样做既使应用简单化，也保证了系统的安全。

例如，下面的数据库中有一张公司部门表 department。表中包括部门号 (d_id)、部门名称 (d_name)、功能 (function) 和办公地址 (address)。department 表的结构如下：

```
mysql> DESC department;
```

Field	Type	Null	Key	Default	Extra
d_id	int(4)	NO	PRI	NULL	
d_name	varchar(20)	NO	UNI	NULL	
function	varchar(50)	YES		NULL	
address	varchar(50)	YES		NULL	

4 rows in set (0.02 sec)

还有一张员工表 worker。表中包含了员工的工作号 (num)、部门号 (d_id)、姓名 (name)、性别 (sex)、出生日期 (birthday) 和家庭住址 (homeaddress)。worker 表的结构如下：

```
mysql> DESC worker;
```

Field	Type	Null	Key	Default	Extra
num	int(10)	NO	PRI	NULL	
d_id	int(4)	YES	MUL	NULL	
name	varchar(20)	NO		NULL	
sex	varchar(4)	NO		NULL	
birthday	datetime	YES		NULL	
homeaddress	varchar(50)	YES		NULL	

6 rows in set (0.01 sec)

由于各部门领导的权力范围不同，因此，各部门的领导只能看到该部门的员工信息；而且，领导可能不关心员工的生日和家庭住址。为了达到这个目的，可以为各部门的领导建立一个视图，通过该视图，领导只能看到本部门员工的指定信息。

例如，为生产部门建立一个名为 product_view 的视图。通过视图 product_view，生产部门的领导只能看到生产部门员工的工作号、姓名和性别等信息。这些 department 表的信息和 worker 表的信息依然存在于各自的表中，而视图 product_view 中不保存任何数据信息。当 department 表和 worker 表的信息发生改变时，视图 product_view 显示的信息也会发生相应的变化。

技巧：如果经常需要从多个表查询指定字段的数据，可以在这些表上建立一个视图，通过这个视图显示这些字段的数据。

MySQL 的视图不支持输入参数的功能，因此交互性上还有欠缺。但对于变化不是很大的操作，使用视图可以很大程度上简化用户的操作。

视图并不同于数据表，它们的区别在于以下几点：

- 视图不是数据库中真实的表，而是一张虚拟表，其结构和数据是建立在对数据中真实表的查询基础上的。
- 存储在数据库中的查询操作 SQL 语句定义了视图的内容，列数据和行数据来自于视图查询所引用的实际表，引用视图时动态生成这些数据。
- 视图没有实际的物理记录，不是以数据集的形式存储在数据库中的，它所对应的数据实际上是在视图所引用的真实表中的。

- 视图是数据的窗口，而表是内容。表是实际数据的存放单位，而视图只是以不同的显示方式展示数据，其数据来源还是实际表。
- 视图是查看数据表的一种方法，可以查询数据表中某些字段构成的数据，只是一些 SQL 语句的集合。从安全的角度来看，视图的数据安全性更高，使用视图的用户不接触数据表，不知道表结构。
- 视图的建立和删除只影响视图本身，不影响对应的基本表。

视图的优点

视图与表在本质上虽然不相同，但视图经过定义以后，结构形式和表一样，可以进行查询、修改、更新和删除等操作。同时，视图具有如下优点：

1) 定制用户数据，聚焦特定的数据

在实际的应用过程中，不同的用户可能对不同的数据有不同的要求。

例如，当数据库同时存在时，如学生基本信息表、课程表和教师信息表等多种表同时存在时，可以根据需求让不同的用户使用各自的数据。学生查看修改自己基本信息的视图，安排课程人员查看修改课程表和教师信息的视图，教师查看学生信息和课程信息表的视图。

2) 简化数据操作

在使用查询时，很多时候要使用聚合函数，同时还要显示其他字段的信息，可能还需要关联到其他表，语句可能会很长，如果这个动作频繁发生的话，可以创建视图来简化操作。

3) 提高数据的安全性

视图是虚拟的，物理上是不存在的。可以只授予用户视图的权限，而不具体指定使用表的权限，来保护基础数据的安全。

4) 共享所需数据

通过使用视图，每个用户不必都定义和存储自己所需的数据，可以共享数据库中的数据，同样的数据只需要存储一次。

5) 更改数据格式

通过使用视图，可以重新格式化检索出的数据，并组织输出到其他应用程序中。

6) 重用 SQL 语句

视图提供的是对查询操作的封装，本身不包含数据，所呈现的数据是根据视图定义从基础表中检索出来的，如果基础表的数据新增或删除，视图呈现的也是更新后的数据。视图定义后，编写完所需的查询，可以方便地重用该视图。

要注意区别视图和数据表的本质，即视图是基于真实表的一张虚拟的表，其数据来源均建立在真实表的基础上。

使用视图的时候，还应该注意以下几点：

- 创建视图需要足够的访问权限。
- 创建视图的数目没有限制。
- 视图可以嵌套，即从其他视图中检索数据的查询来创建视图。
- 视图不能索引，也不能有关联的触发器、默认值或规则。
- 视图可以和表一起使用。
- 视图不包含数据，所以每次使用视图时，都必须执行查询中所需的任何一个检索操作。如果用多个连接和过滤条件创建了复杂的视图或嵌套了视图，可能会发现系统运行性能下降得十分严重。因此，在部署大量视图应用时，应该进行系统测试。

提示：ORDER BY 子句可以用在视图中，但若该视图检索数据的 SELECT 语句中也含有 ORDER BY 子句，则该视图中的 ORDER BY 子句将被覆盖。

关注公众号「[站长严长生](#)」，在手机上阅读所有教程，随时随地都能学习。本公众号由[C语言中文网站](#)，[长](#)亲自运营，长期更新，坚持原创。



微信扫码关注公众号

[< 上一节](#)

[下一节 >](#)

优秀文章

[C++重载++和--（自增和自减运算符）](#)

[Java面向对象：对象的概念及面向对象的三个基本特征](#)

[C语言清空（刷新）缓冲区，从根本上消除那些奇怪的行为](#)

[C语言结构体指针（指向结构体的指针）详解](#)

[Python continue的用法](#)

[Java经典视频教程汇总，从入门到精通](#)

[JSP include动作](#)

