

第10章 MySQL视图

人工智能学院
古 晶

西安电子科技大学



第10章 MySQL视图



- 10.1 视图
- 10.2 视图定义和管理
- 10.3 更新视图数据
- 10.4 对视图的进一步说明
- 10.5 知识点小结
- 本章实验



第10章 MySQL视图



- 视图是从一个或多个表中导出的表，是一种**虚拟**存在的表。
- 视图就像一个窗口，通过这个窗口可以看到系统专门提供的数据。这样，用户可以只关心对自己有用的数据。
- 视图可以使用户的**操作更方便**，并且可以保障数据库**系统安全性**。
- 本章将介绍视图的含义和作用、视图定义的原则和创建视图的方法以及修改视图、查看视图和删除视图的方法。



第10章 MySQL视图



□10.1 视图

□10.2 视图定义和管理

□10.3 更新视图数据

□10.4 对视图的进一步说明

□10.5 知识点小结

□本章实验



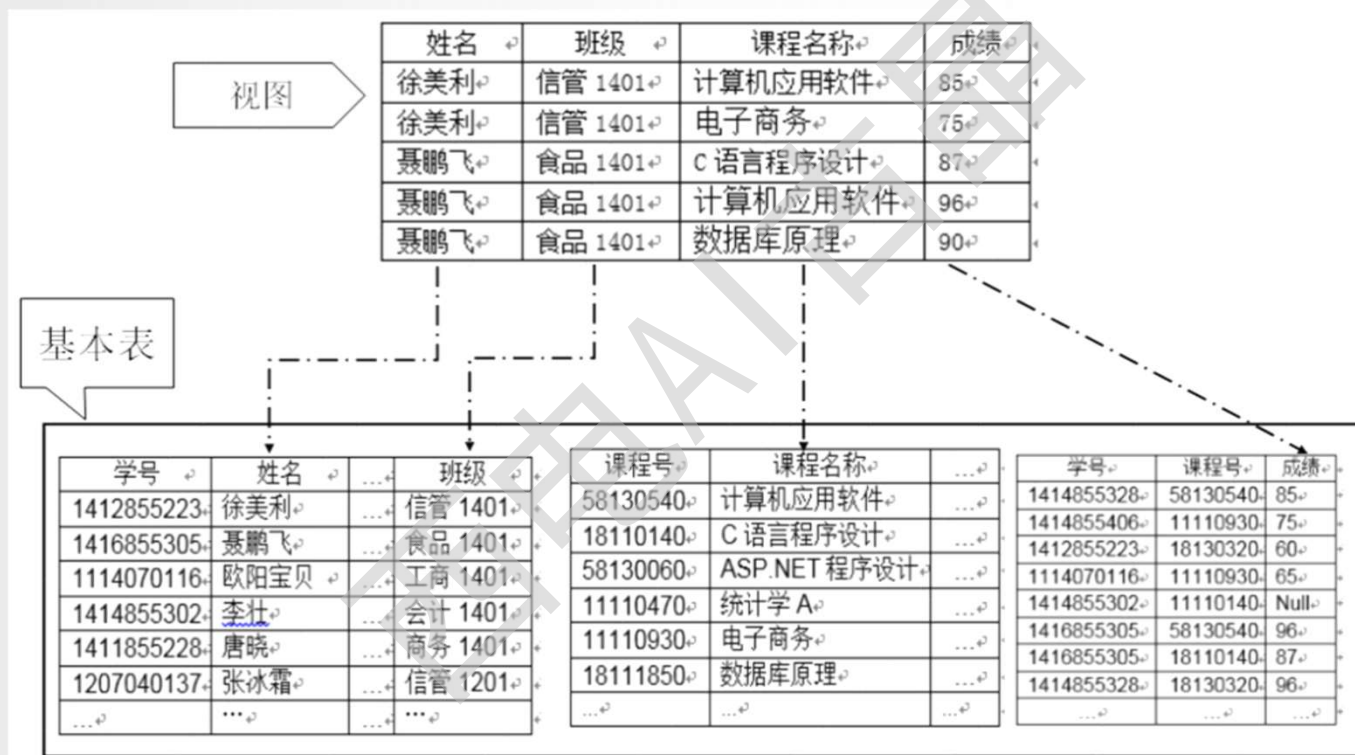
10.1 视图概述



- 视图(view)为数据查询提供了一条捷径；视图是一个**虚拟表**，其内容由查询定义，即视图中的数据并不像表、索引那样需要占用存储空间，视图中保存的仅仅是一条**select语句**，其数据源来自于数据库表，或者其他视图。
- 它同真实的表一样，视图包含一系列带有名称的列和行数据。
- 视图并不在数据库中以**存储的数据**的形式存在。行和列数据来自于定义视图的查询所引用的表，并且在引用视图时动态生成。当基本表发生变化是，视图的数据也会随之变化。
- 视图是存储在数据库中的**查询的SQL语句**，使用它主要出于两种原因：第一是**安全**原因，视图可以隐藏一些数据；第二是可使复杂的查询**易于理解和使用**。



10.1 视图与基本表之间的对应关系:





10.1 视图的优势



□视图的**优势**:

➤ **第一**, 增强**数据安全性**

- 同一个数据库表可以创建不同的视图, 为不同的用户分配不同的视图, 这样就可以实现不同的用户只能查询或修改与之对应的数据, 继而增强了数据的安全访问控制。

➤ **第二**, 提高**灵活性**, 操作变简单

- 有灵活性的功能需求后, 需要改动表的结构而导致工作量比较大。那么可以使用虚拟表的形式达到少修改的效果。
- 使用视图可以简化数据查询操作, 对于经常使用, 但结构复杂的select语句, 建议将其封装为一个视图。



10.1 视图的优势



□视图的**优势**:

➤ **第三**, 提高数据的**逻辑独立性**

- 使应用程序和数据库表结构在一定程度上逻辑分离
(1)使用视图可以向应用程序屏蔽表结构, 此时即便表结构发生变化, 只需重新定义视图或者修改视图的定义, 无须修改应用程序即可使应用程序正常运行。
(2)使用视图可以向数据库表屏蔽应用程序, 此时即便应用程序发生变化, 只需重新定义视图或者修改视图的定义, 无须修改数据库表结构即可使应用程序正常运行。



10.1 视图的工作机制



□视图的**工作机制**:

- 当调用视图的时候，才会**执行视图中的SQL**，进行取数据操作。
- 视图的内容没有存储，而是在视图被引用的时候才派生出数据。
- 这样**不会占用空间**，由于是即时引用，视图的内容总是与真实表的内容是一致的。



第10章 MySQL视图



- 10.1 视图
- **10.2 视图定义和管理**
- 10.3 更新视图数据
- 10.4 对视图的进一步说明
- 10.5 知识点小结
- 本章实验



10.2 视图定义和管理



□视图定义和管理

- 创建视图
- 删除视图
- 查看视图定义
- 修改视图定义



10.2 创建视图

- ❑ 创建视图需要具有create view的**权限**，同时应该具有查询涉及的列的select权限。
- ❑ 在MySQL数据库下面的user表中保存这些权限信息，可以使用select语句查询。
- ❑ 视图**语法格式**:

```
create [algorithm = {undefined | merge | temptable}]  
view 视图名 [(视图列表)]  
as 查询语句  
[with [cascaded | local] check option]
```

其中：(1)“**视图名**” 参数表示要创建的视图名称。



10.2 创建视图



(2) **algorithm**是可选参数，表示**视图选择的算法**；undefined选项表示MySQL自动选择要使用的算法；merge选项表示将使用视图的语句与视图的定义合起来，使得视图定义的某部分取代语句的对应部分；temptable选择表示将视图的结果存入临时表，然后使用临时表执行语句。

(3)“**查询语句**”参数是一个**完整的查询语句**，表示从某个表中查出某些满足条件的记录，将这些记录导入视图中。

(4)**cascaded**是可选参数，表示更新视图时要满足所有相关视图和表的条件，该参数为默认值；local表示更新视图时，要满足该视图本身的定义条件即可。

(5)**with check option**是可选参数，表示更新视图时要保证在该视图的权限范围之内。



10.2 创建视图



- 视图定义后，就可以如同查询基本表那样对视图进行查询。
- 创建视图的**注意事项**:
 - 1.运行创建视图的语句需要用户具有**创建视图(create view)的权限**，若加了[or replace]，则还需要用户具有**删除视图(drop view)的权限**。
 - 2.select语句不能包含**from子句中的子查询**。
 - 3.select语句不能引用**系统或用户变量**。
 - 4.select语句不能引用**预处理语句参数**。
 - 5.在存储子程序内，定义不能引用**子程序参数或局部变量**。
 - 6.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。



10.2 创建视图



- 7.在定义中不能引用**temporary表**，不能创建temporary视图。
 - 8.在视图定义中命名的表必须已存在。
 - 9.不能将**触发程序**与视图关联在一起。
 - 10.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。
- **注意：**使用视图查询时，若其关联的基本表中添加了新字段，则该视图将**不包括新字段**。如果与视图相关联的表或视图被删除，则该视图将不能使用。



10.2 删除视图



- 删除视图时，只能删除视图的**定义**，**不会删除数据**。
- 用户必须拥有**drop**权限。

□语法格式：

```
drop view [if exists]  
view_name[, view_name2]...  
restrict | cascade
```

- 其中，view_name是视图名，声明了if exists，若视图不存在的话，也不会出现错误信息。也可以声明restrict和cascade，但他们没什么影响。使用drop view 可以一次删除**多个视图**。



10.2 查看视图定义



□查看视图是指查看数据库中已经存在的视图的定义。查看视图必须要**有show view的权限**。

□查看视图的几种方法：

- 1) **describe** 语句，语法格式：describe 视图名称； 或者desc视图名称；
- 2) **show table status** 语句,语法格式: show table status like '视图名'
- 3) **show create view** 语句，语法格式: show create view '视图名'
- 4) 查询information_schem数据库下的views表
语法格式：select * from information_schema.views where table_name = '视图名'



10.2 修改视图定义



□修改视图是指修改数据库中已经存在表的定义。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的一致。

□MySQL中通过**create or replace view** 语句或者**alter**语句来修改视图。

(1) create or replace view **语句格式**

create or replace [algorithm = {undefined | merge | temptable}]

view 视图名[{ 属性清单 }]

as select 语句

[with [cascaded | local] check option];



10.2 修改视图定义

(2) alter **语句格式**

create or replace [algorithm = {undefined | merge | temptable}]

view 视图名[{ 属性清单 }]

as select 语句

[with [cascaded | local] check option];

这里的所有参数都与创建视图的参数一样。



第10章 MySQL视图



- 10.1 视图
- 10.2 视图定义和管理
- 10.3 更新视图数据**
- 10.4 对视图的进一步说明
- 10.5 知识点小结
- 本章实验



10.3 更新视图数据



- 对视图的更新其实就是**对表的更新**，更新视图是指通过视图来插入（insert）、更新（update）和删除（delete）表中的数据。因为视图是一个虚拟表，其中没有数据。
- 通过视图更新时，都是转换到**基本表**来更新。
- 更新视图时，只能**更新权限范围内**的数据。超出了范围，就不能更新。
- 对检查视图进行更新操作时，只有满足**检查条件**的更新操作才能顺利执行。



10.3 更新视图数据



- 检查视图分为**local检查视图**与**cascade检查视图**。
 - **with_check_option**的值为1时表示local (**local视图**)，通过检查视图对表进行更新操作时，只有满足了视图检查条件的更新语句才能够顺利执行；
 - 值为2时表示**cascade** (**级联视图**，在视图的基础上再次创建另一个视图)，通过级联视图对表进行更新操作时，只有满足所有针对该视图的所有视图的检查条件的更新语句才能够顺利执行。
- **原则**：尽量不要更新视图，并且更新视图的语法update语法一样。



10.3 视图无法更新的情况



□ 视图**无法更新**的情况:

- 1.视图中包含sum(), count()等**聚集函数**的;
- 2.视图中包含**union、union all、distinct、group by、having**等关键字的;
- 3.**常量视图**, 比如: `create view view_now as select now();`
- 4.视图中包含**子查询**;
- 5.由**不可更新**的视图导出的视图;
- 6.创建视图时algorithm为**temptable**类型;



10.3 视图无法更新的情况



□视图无法更新的情况:

7.视图对应的表上存在**没有默认值**的列, 而且该列没有包含在视图里;

8.with [cascaded|local] check option也将决定视图是否可以更新:

- local参数表示更新视图时要满足该视图本身定义的条件即可;
- cascaded参数表示更新视图时要满足所有相关视图和表的条件, 默认值。



第10章 MySQL视图



- 10.1 视图
- 10.2 视图定义和管理
- 10.3 更新视图数据
- 10.4 对视图的进一步说明**
- 10.5 知识点小结
- 本章实验



10.4 对视图的进一步说明



- 视图是在原有的表或者视图的基础上重新定义的**虚拟表**，这可以从原有的表上选取对用户有用的信息。那些对用户没有用，或者用户没有权限了解的信息，都可以直接屏蔽掉。
- 这样做既可以使应用简单化，也保证了系统的安全。
- 视图起着类似于**筛选**的作用。



10.4 对视图的进一步说明



□ 视图的**作用**:

1. 使**操作简单化**

视图需要达到的目的就是所见即所需。也就是说，从视图看到的信息就是所需要了解的信息。视图可以简化对数据的操作。

2. 增加**数据的安全性**

通过视图，用户只能查询和修改指定的数据。数据库授权命令可以限制用户的操作权限，但不能限制到特定行和列上。这样可以保证敏感信息不会被没有权限的人看到，可以保证一些机密信息的安全。

3. 提高表的**逻辑独立性**

视图可以屏蔽原有表结构变化带来的影响。



第10章 MySQL视图

- 10.1 视图
- 10.2 视图定义和管理
- 10.3 更新视图数据
- 10.4 对视图的进一步说明
- 10.5 知识点小结**
- 本章实验



10.5 知识点小结



本章知识小结:

- MySQL数据库中视图的含义和作用
- 创建视图、修改视图和删除视图的方法



第10章 MySQL视图



- 10.1 视图
- 10.2 视图定义和管理
- 10.3 更新视图数据
- 10.4 对视图的进一步说明
- 10.5 知识点小结
- 本章实验



本章实验

□ 实验内容:

(8) 删除视图alexview。



Thanks!

See you