

《数据库技术及 SQLSERVER》

课堂教学设计

2018年6月

课堂教学设计

目 录

《数据库技术及 SQL SERVER》教学大纲中基本教学内容共 12 个项目，
此次教学设计的 10 个子项目分别节选自项目一至项目八及项目十二

1. 数据库概念与设计·····	4
选自 项目一 数据库基本概念与数据库设计 任务一 数据库概念与设计	
2. 数据模型·····	14
选自 项目一 数据库基本概念与数据库设计 任务二 数据模型	
3. 安装、配置 SQL SERVER 和数据安全性·····	26
选自 项目二 SQL Server 2008 概述 任务一 安装、配置 SQL SERVER 和数据安全性	
4. 数据库的创建与管理·····	39
选自 项目三 管理数据库 任务一 数据库的创建与管理	
5. 数据表的创建·····	54
选自 项目四 数据表的操作 任务一 数据表的创建	
6. 数据库的查询·····	64
选自 项目五 管理数据库 任务二 数据库的查询	
7. 数据表的嵌套查询·····	77
选自 项目五 数据表的操作 任务三 数据表的嵌套查询	
8. 数据表的更新操作·····	89
选自 项目五 数据表的操作 任务四 数据表的更新操作	
9. 数据视图·····	101
选自 项目六 视图及应用 任务一 数据视图	
10. 创建数据库函数·····	117
选自 项目八 流程控制与函数 任务一 创建数据库函数	

一、主题 9/项目：任务一 数据视图

第 6.1 单元（节），2 学时

授课时间 2018 年 5 月 5 日 授课地点：实训-1602

二、学情分析

学生已基本具备学习本课程必要的知识基础。在前导课程学习的基础上，学生对程序编写有了初步的认识，这就为这节课的学习打下了良好的基础；通过本节的学习，提高学生的数据视图使用的能力，找到数据库设计的规律和方法。

三、教学目标

建立(定义)视图。

四、教学重点、难点及解决方案

（一）教学重点

掌握建立视图的一般格式。

（二）教学难点

查询视图。

（三）解决办法

通过举例深入讲解，结合学生的思考，逐步深入。

五、教学内容分析

[教学资源 微课 项目五 数据库的查询 T-SQL语句的使用2](#)

导入新课 5min

本周开始我们学习建立视图及使用。

讲授新课 30min

6-1 数据视图 (20min)

6-1-1 定义视图

1、 建立(定义)视图

例 1 建立信电系学生的视图

```
drop view is_student;

create view is_student /*(sno_1,sname_2,sage_3)*/

as

select  sno,sname,sage

from student

where sdept='信电系';
```

//建立视图的一般格式:

```
create view <视图名>[(<列名>[, <列名>]...)]

as <子查询>

[with check option];
```

说明:

*子查询可以是任意复杂的 select 语句，但通常不允许含有 order by 子句和 distinct 短语。

*with check option 保证更新、插入、删除时满足视图定义中的谓词条件

***省略组成视图的各个属性列名**

视图属性列隐含由子查询中的 SELECT 子句目标列中的诸字段组成。

***下列三种情况必须明确指定组成视图的所有列名**

**其中某个列不是单纯的属性名，而是集函数或列表表达式。

**多表连接时选出了几个同名列作为视图的字段。

**需要在视图中为某个列名用新的更合适的名字。(个人喜好)

***视图是一个虚表**

******从一个或多个基本表（或视图）导出的表。

******在数据库（数据字典）中只存放视图的定义，而不存放视图所对应的数据，这些数据仍然放在原来的基本表中。

******定义视图时，并不执行其中的 SELECT 语句。

/查询视图

```
select * from is_student;
```

```
select sno,sname,sage  from is_student  
where sage<18;
```

***执行过程**

- 1) 进行有效性检查：检查涉及的表、视图等是否存在于数据库中，
- 2) 转换成对基本表的查询：从数据字典中取出查询涉及到视图的定义，将定义中的子查询和用户对视图的查询结合起来。
- 3) 执行经过修正的查询

```
select sno,sname,sage  from is_student  
where sage<18;//修正后的查询语句为：
```

```
select  sno,sname,sage  
from student  
where sdept='信电系' and sage<18;
```

***用户就可以像对基本表进行查询一样对视图查询**

1) 建立在基本表上的单表视图

***建立在基本表上的单表视图有三种形式**

行列子集视图、带表达式视图、分组视图

//行列子集视图的概念（重要）

从单个基本表中导出，只是去掉基本表的某些行和某些列，但保留了码。（前例 建立信电系学生的视图）

//带表达式视图

定义一个反映学生出生年份的视图

```
create view birth_st(sno,sname,sbirth)
as
select  sno,sname,1996-sage
from student;
select * from birth_st;
```

//分组视图

例：将学生的学号及其平均成绩定义为一个视图

```
create view s_g(sno,gavg)
as
select  sno,avg(grade)
from sc
group by sno;
select * from s_g;
```

2) 建立在基本表上的多表视图（普通多表、带表达式、分组）

例：建立信电系选修了 1 号课程的学生视图，视图属性包括 sno,sname,grade

```
drop view is_student1;
create view is_student1(sno, sname, sage)
```

as

```
select student.sno,sname,grade
from student,sc
where sdept='计算机' and student.sno=sc.sno
and sc.cno=1;
```

3) 建立在视图上的视图

```
drop view is_student2;
create view is_student2((sno, sname, sage)
```

as

```
select sno,sname,grade
from is_student
where grade>=90;
```

单视图上的视图（普通单表视图，带表达式视图，分组视图）

多视图上的视图

4) 建立在基本表与视图上的视图（普通、带表达式、分组）

建立信电系选修了 1 号课程的学生视图，视图属性包括 sno,sname

```
create view is_student3(sno, sname)
```

as

```
select sno,sname
from is_student,sc
where is_student.sno=sc.sno
and sc.cno=1;
```

5) 定义视图子查询中的 ‘*’ 的考虑

例：将 student 表中所有女生记录定义为一个视图

```
create view f_student(sno, sname,ssex,sage,sdept)
```

```
as select *
```

```
from student
```

```
where ssex='女';
```

****子查询中用 ‘*’ 的缺点：**

如果修改了基本表的结构，视图与基本表的映象关系受到破坏，视图不能正确工作。

****解决方法：**

- 1) 子查询中 “*” 明确指定属性列名
- 2) 修改基本表后删除原来的视图，重建视图

2、 删除视图

```
drop view <视图名>;
```

6-1-2 查询视图

1. 查询的执行过程为（前例已述）

- 1) 效性检查：检查涉及的表、视图等是否存在于数据库中，
- 2) 转换成对基本表的查询：从数据字典中取出查询涉及到视图的定义，将定义中的子查询和用户对视图的查询结合起来。
- 3) 执行经过修正的查询

2. 一般情况下视图查询的转换是直截了当的

例：在信息系学生的视图中，找出年龄小于 20 岁的学生

```
SELECT Sno,Ssge
```

```
From is_student
```

Where sage<20

Is_student 视图定义中的子查询

```
SELECT Sno,Sname,Ssge
```

```
From student
```

```
Where sdept='信电系'
```

修正后对基本表的查询

```
SELECT Sno,Ssge
```

```
From student
```

```
Where sdept='信电系' and sage<20
```

3. 在有些情况下转换不能直接进行

例 在 S_G 视图上查询平均成绩在 90 份以上学生的学号和成绩 P106

```
create view s_g(sno,gavg)
```

```
as
```

```
select sno,avg(grade)
```

```
from sc
```

```
group by sno;
```

以下查询:

```
select * from S_G where gavg>=90;
```

将被转换为:

```
select sno,avg(grade)
```

```
from sc
```

```
where avg(grade)>=90
```

```
group by sno; //where 子句中是不能用集函数作为条件表达式, 所以此语句
```

不能被正确执行。

正确的查询语句应该是：

```
select sno,avg(grade)
from sc
group by sno
having avg(grade)>=90;
```

//目前多数关系数据库系统不能正确转换成这种方式，但 SQL anywhere 能正确转换成这种形式，即上述查询语句能正确执行。

注意要点

一般来说 DBMS 对行列子集视图的查询均能直接进行正确转换，其它视图不一定能保证，对这类视图进行查询时应尽量避免视图中的特殊属性出现在查询中。实际应用时需要查有关资料，或直接在所选的 DBMS 产品上做尝试。

6-1-3 更新视图

目前各个关系数据库系统产品一般都只允许更新行列子集视图，而且不同的系统还有进一步的规定，使用时需要查相关的资料。

*with check option 保证更新（插入、修改、删除）时满足视图定义中的谓词条件（子查询中的条件表达式）

```
drop view is_student;
create view is_student //属性考虑
as
select sno,sname,sage//,sdept//分两种情况考虑
from student
where sdept='信电系';
```

// with check option;

1. 插入

1) 视图中包含系列 (sdept)

insert into is_student values(97008,'www',19,'信电系');//有无 with check option 都能执行

insert into is_student values(97009,'www',19,'化工系');//无——执行，有不执行

2) 视图中不包含系列 (sdept)

insert into is_student values(95029,'www',19);

在 SQL ANYWHERE 环境中 (与教材情况不同)

*无 with check option --执行，但不能自动加上 ‘信电系’

*有 with check option 不执行

书上的插入原理

insert into is_student values(97008,'www',19);

*执行过程:

转换为对基本表的更新

insert into student values(97008,'www',19,'信电系');

//在有些数据库产品中不能实现 (在 SQL Anywhere 中信电系没有自动加上)

2. 修改

sno	sname	sage	sdept
97009	www	19	化工系
95029	www	19	
2001	a	16	计算机

2002	b	19	计算机
2003	aaaaa	20	信电
2004	d	15	信电
2005	e	20	化工

update yj_is_student

set sname='aaaaa'

where sno=2001;

同前转换为对基本表的修改

*执行过程

- 1) 进行有效性检查：检查涉及的表、视图等是否存在于数据库中，
- 2) 转换成对基本表的更新：从数据字典中取出该语句所涉及视图的定义，将定义中的子查询和用户对视图的更新操作结合起来。
- 3) 执行经过修正的更新操作

前例转换后的更新语句为：

update student

set sname='aaaaa'

where sno=2001 and sdept='信电系';

不管有无 with check option，都只能修改信电系学生记录

3. 删除元组

delete from is_student

where sno=2003; //能被删除

delete from is_student

where sno=2001; //不能被删除

转换后的删除语句为

```
delete from student
```

```
where sno=2003 and sdept='信电系';//能被删除
```

//视图更新时注意事项

- 1) 目前各个关系数据库一般只允许对行列子集视图进行更新。一般对行列子集视图都可以执行修改和删除操作，如果基本表中所有不允许空值的列都出现在视图中，则也可以对其执行插入操作。
- 2) 不同系统对视图的更新有进一步的规定，不同系统实现方法的不同，具体规定也不同。

6-1-4 视图的用途

1. 能简化用户的操作

从视图的定义中可以看出。

2. 使用户能以多种角度看待同一数据

同一数据针对不同用户可以定义不同视图

3. 视图对重构数据库提供了一定程度的逻辑独立性

为何说只提供一定程度的逻辑独立性，由于对视图的更新是有条件的，因此基本表中的修改数据的语句可能仍会因基本表结构的改变而改变。

4. 视图能够对机密数据提供安全保护

不同部门只能看到本部门的数据

六、教学方法与手段

(一) 教学方法

针对课程特点、学生特点和软件企业对人才的实际需求，在“教学做一体化”模式下，积极进行多种教学方法探索，通过项目导向、协作讨论、启发引

导等多种方式，培养学生的创新能力和独立分析问题、解决问题的能力。主要采用的有以下几种：

（1）项目任务驱动教学法。解决真实或虚拟企业项目开发中的实际问题，通过精选案例、分解任务、讲练有机融合，将相关知识点融入项目，采用项目组的形式，按照软件公司运作形式和项目开发流程，通过项目实施，使学生能够积极参与整个教学过程，体现“以学生为中心，教师为主导”的课程基本理念。

（2）小组协作、角色扮演结合教学法。以项目学习小组的形式组织学生，教师是客户，每组指定项目经理，明确分工，从需求分析、总体设计、文档制作、编码、测试直到发布，全部由学生在教师的引导下独立完成，充分发挥了主观能动性、熟练了开发流程、掌握了技术细节，提高团队合作能力。

（二）教学手段

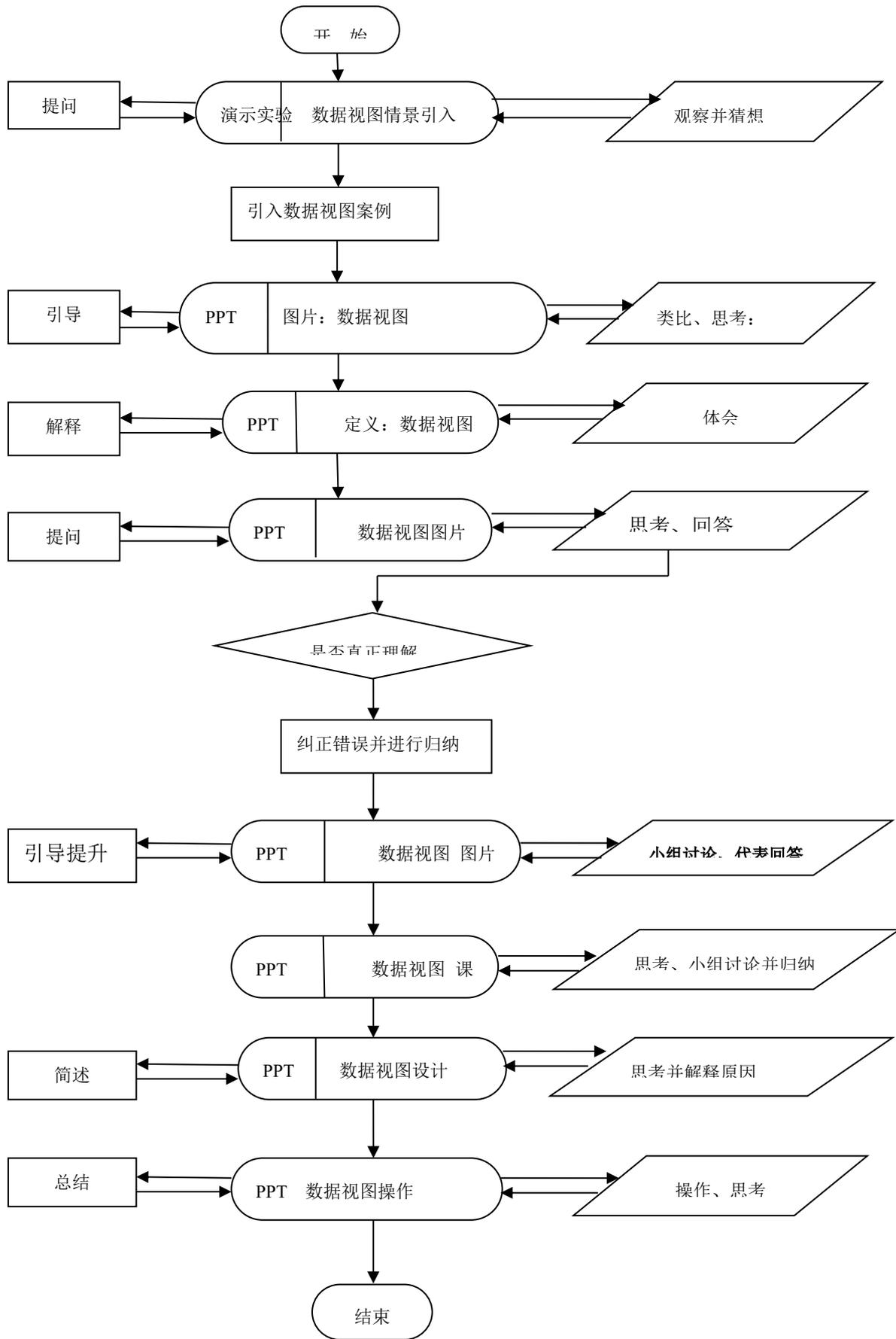
采取的教学手段有：

- （1）精心设计教学课件，通过创建问题情境，激发学生学习兴趣。
- （2）教师课堂编程示范，锻炼学生程序编制能力。
- （3）充分利用多媒体教学系统，进行课堂辅助教学。

利用网络将课堂教学延伸到课外，注重学生课后学习过程监控。

七、教学过程流程图

教学过程流程图如下所示。



八、教学评价

项目	A 级	B 级	C 级	个人评价	同学评价	教师评价
认真	完成预习作业，上课认真听讲，作业认真，参与讨论态度认真	上课能认真听讲，作业依时完成，有参与讨论	上课无心听讲，经常欠交作业，极少参与讨论			
积极	积极举手发言，积极参与讨论与交流，大量阅读课外读物	能举手发言，有参与讨论与交流，有阅读课外读物	很少举手，极少参与讨论与交流，没有阅读课外读物			
自信	大胆提出和别人不同的问题，大胆尝试并表达自己的想法	有提出自己的不同看法，并作出尝试	不敢提出和别人不同的问题，不敢尝试和表达自己的想法			
善于与人合作	善于与人合作，虚心听取别人的意见	能与人合作，能接受别人的意见。	缺乏与人合作的精神，难以听进别人的意见			
思维的条理性	能有条理表达自己的意见，解决问题的过程清楚，做事有计划	能表达自己的意见，有解决问题的能力，但条理性差些	不能准确表达自己的意思，做事缺乏计划性，条理性，不能独立解决问题			
思维的创造性	具有创造性思维，能用不同的方法解决问题，独立思考	能用老师提供的方法解决问题，有一定的思考能力和创造性	思考能力差，缺乏创造性，不能独立解决问题			
我这样评价自己：						
同伴眼里的我：						
老师的话：						

九、教学反思

数据库系统是采用了数据库技术的计算机系统。数据视图是非常重要的应用工具。改进措施是教师要不断跟踪网络技术的新发展和新应用，注意引入学科新知识、新动态，提高授课质量，加大课堂信息量。