

《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 4/项目：任务一 数据库的创建与管理

第 3.1 单元（节）， 2 学时

授课时间 2018 年 3 月 21 日 授课地点： 实训-1602

二、学情分析

在数据库安装任务学习的基础上，学生对数据库有了初步的认识，这就为本节课的学习打下了良好的基础；通过本节的学习，提高学生的学习数据库技术的兴趣，找到数据库设计的规律和方法。并对本节课的前期知识结构有所了解。

三、教学目标

（一）知识目标

- 1.掌握关系数据库标准语言的结构化查询语言 Select 语句的使用；
- 2.掌握数据定义语言 CREATE，DROP，ALTER 的使用；
- 3.掌握数据操作语言 INSERT，UPDATE，DELETE 的使用。
- 4.掌握数据控制语言 GRANT， REVOKE 的使用。

（二）能力目标

熟练使用关系数据库标准语言的结构化查询语言。

（三）素质目标

培养学生项目合作、团队精神以及沟通能力；

培养学生逻辑思维能力和分析解决问题的能力。

四、教学重点、难点及解决方案

（一）教学重点

1. 数据定义（表、视图、索引的定义）。表定义与完成性约束定义，数据定义与数据字典，索引能加快查询速度的原因，对视图操作与对基本表操作，视图及

外模式。

2. 数据更新操作与数据的完整性维护，数据更新操作与数据库的一致性维护问题。

3. SQL查询（单表查询、连接查询、嵌套查询、集合查询）及相关的关系代数表达。

4. 嵌入式SQL。

（二）教学难点

1. 带有EXISTS谓词的子查询。

（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](#) [项目四 数据表的操作](#) [基本表的创建](#)

导入新课 5min

本周开始我们学习专用于关系数据库的结构化查询语言——SQL。

讲授新课 30min

3-1 学习标准 SQL 语言核心功能的 9 个动词 （10min）

各个 DBMS 产品在实现标准 SQL 语言时各有差别，一般都作了某些扩充。

SQL 功能	动词
数据定义	CREATE,DROP,ALTER
数据查询	SELECT
数据操纵	INSERT,UPDATE,DELETE
数据控制	GRANT,REVOKE

3-2 数据定义 (30min)

数据定义的基本对象为表、索引、视图（导出表）

操 作 对 象	操作方式		
	创建	删除	修改
表	CREATE	DROP	ALTER

	TABLE	TABLE	TABLE
索引	CREATE INDEX	DROP INDEX	
视图	CREATE VIEW	DROP VIEW	

****基本表独立存在，不同的 RDBMS 产品，表的物理存储会有不同的方式：若干基本表存储于一个文件；一个表存储与一个文件。（小型 DBMS 如 DBASE，一个基本表对应一个存储文件）**

****索引依附于基本表。其作用是为了加快数据的访问速度。两类：聚簇索引与非聚簇索引**

****视图为导出表（基于基本表的虚表），只存储视图定义，不存放视图对应的数据，数据存放在导出视图的基本表中。（对视图操作最终通过视图的定义变为对基本表操作）**

****SQL 数据定义语句**

3.2.1 定义、删除、修改基本表

1、 举例

例 1：从以下 3 方面理清定义表的条件

a 名、属性： student(no,name,sex,age,dept)

b 属性的域（即属性的数据类型）

c 约束

1) 主键（no）

2) 参照（无）

3) 自定义（由实际需要定）

name（ unique not null）

sex（not null）

age (10<age<150)

dept(无约束条件)

定义的实现

```
create table student
( no      char(6) primary key, // 主键 即实体完整性
  name    char(6) unique not null ,
  sex     char(1) not null ,
  age     int(eget) check(age >10 and age<150),
  dept    char(8)
// primary key(no) //表级约束
);
```

2、一般格式

1) 表定义内容(关系模式、完整性约束)

*关系模式 R (U, D, DOM, F)。F 没有直接体现。

*完整性约束

结合前例进一步说明。

2) 定义基本表的一般格式

CREATE TABLE <表名>

(

<列名> <数据类型> [列级完整性约束条件]

[, <列名> <数据类型> [列级完整性约束条件]...]

[, <表级完整性约束条件>]

);

*格式说明：create table 为关键字，表名由用户自定义，属性由逗号分割，
结束符为；

3) 最常用的数据类型（域）

不同数据库系统支持的数据类型不完全相同（实际使用时需要查阅相关资料）。最常用的数据类型有：

integer,float,date,char(n)等

SQL SERVER 支持以下数据类型

二进制 binary[(n)] varbinary[(n)]

字符型 char(n),varchar(n)

日期及时间型 data datetime,

精确数值型 decimal(p,s), numeric(p,s) p 表示整数位与小数位之和,
s 表示小数位

近似数值型 float,real

整形 int (integer) 4 字节 smallint 2 字节 tinyint 1 字节

货币 money

文本和图象 text image

4) 完整性规则

(1)实体完整性不可少（NOT NULL UNIQUE）

(2)列级约束与表级约束的区别

如果完整性约束涉及到该表的多个属性列，必须定义在表级上，否则
既可以定义在列级也可以定义在表级

用 SC 表说明

create table student_course

```
( sno char(6) ,
  cno char(6),
  grade int,
  primary key(sno,cno)
);
```

(3) 自定义

- *非空约束 (not null),
- *唯一约束(unique), (可 NULL) (与 primary key 的区别)
- *检查约束(check)
- *缺省约束 DEFAULT。

(4) 约束的定义的形式

在不同的数据库管理系统产品上某些地方有所不同,需要查相关产品的资料。

(5) 外键约束

student(no, name, sex , age , dept) //外码 dept

department(dept, dept_name, location) //主码

create table student

(no char(6) primary key,

name char(6),

sex char(1),

age integer,

dept char(8) references department,/(dept),

// 列级约束 // constraint f_k foreign key (dept) references


```
department, //(dept_code), // 表级约束

);

create table department

(

    dept      char(8)  primary key,

    dept_name char(20),

    location  char(20),

);
```

(6) 完整性检查过程

*建表时完整性约束条件被存入系统的数据字典中

*用户操作表中的数据时，由 DBMS 自动检查该操作是否违背所定义的完整性约束条件。

5) 表（数据定义）内容的存储

执行建表语句后，关系模式与完整性约束条件被存入系统的数据字典中

6) 修改基本表

*增加新列

*删除完整性约束

*修改列

其一般格式为：

ALTER TABLE <表名>

[ADD <新列名><数据类型>[完整性约束]]

[DROP <完整性约束名>]

[MODIFY 〈列名〉〈数据类型〉];

举例:

```
drop table student;
```

```
create table student
```

```
(
```

```
dept_code char(8) primary key,
```

```
name char(20) unique,
```

```
sex char(1)
```

```
);//增加新列
```

```
alter table student add age integer not null;
```

```
alter table student add age integer not null unique;
```

//删除约束

```
alter table student drop unique(name);
```

//修改已经存在的列

```
alter table student modify sex char(2) not null;//char(1)改为 char(2),并加
```

上约束

7) 删除基本表（一般用得不多）

```
drop table<表名>;
```

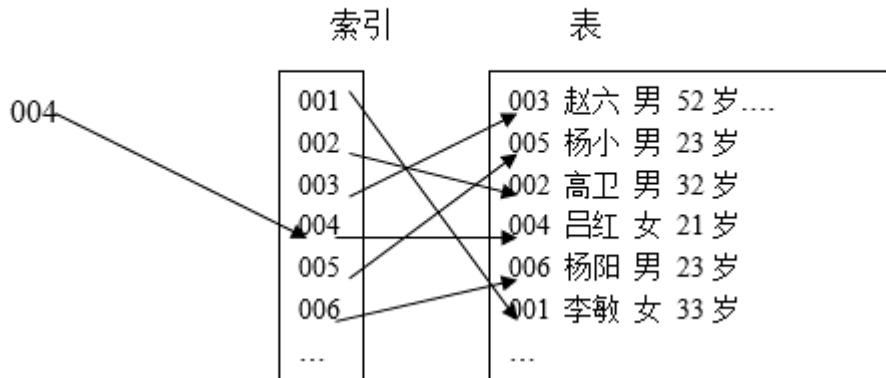
```
drop table student;
```

3.2.2 建立与删除索引（11_29）

1、 索引的作用

*建立索引是加快表查询速度的有效手段。

*原理示例图： student(no,name,sex,age)



数据更新部分功能举例

1、 表中插数据

1) 插入整个元组:

```
insert into b values('97006','mm','f',21,'信电系');
```

```
insert into b values('97007','nn','m',22,'信电系');
```

2) 插入部分属性

```
insert into b(no,name) values('97006','mm','f',21,'信电系');
```

// 出错（与前面元组主码同）

3) 插入操作时的完整性检查

(1) 主键约束: 不能插入主键属性值相同的元组，主键属性值不能为空。

```
insert into b values('97006','mm','f',21,'信电系');
```

```
insert into b values('97007','nn','m',22,'信电系');
```

```
insert into b(no,dept) values('97006','信电系');//因为主键值相同不能插入
```

```
insert into b(name,sex,age,dept) values('nn','f',22,'信电系');
```

//主键属性为空，不能插入。

(2) 外键约束

```
student(no, name, sex , age , dept)      //外码 dept
```

department(dept, dept_name, location) //主码

*被引用关系元组

insert into department values('1','信电系','教二楼');

insert into department values('2','化工系','教二楼');

select * from department;

*外键取值

insert into s.. values('97001','aa','f',21,null);//空值

insert into s.. values('97002','aa','f',21,'1');

//被引用关系元组主属性值

insert into s.. values('97003','aa','f',21,'2');

insert into s.. values('97004','aa','f',21,'5');//出错

(3) 检查约束

drop table student;

create table student

(no char(6) primary key,

name char(6),

sex char(1),

age integer check(age<28 and age>10),

dept char(8) check(dept like '%系'),

);

insert into b values('97002','aa','m',27,'信电系');//正确

insert into b values('97002','aa','m',30,'信电系');//错

insert into b values('97002','aa','m',26,'信电');//错

4) 插入操作中的索引维护

2、 删除元组

`delete from b where ;`

3、 查看表中数据

`select * from b;`

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。

(3) 基于榜样教学法，老师是榜样，是老程序员，学生是新程序员，要向老程序员学习。学生模拟教师讲授过的项目，完成知识点相同或类似的项目，以巩固学生对相关知识的掌握，并树立学生信心和成就感。

（二）教学手段

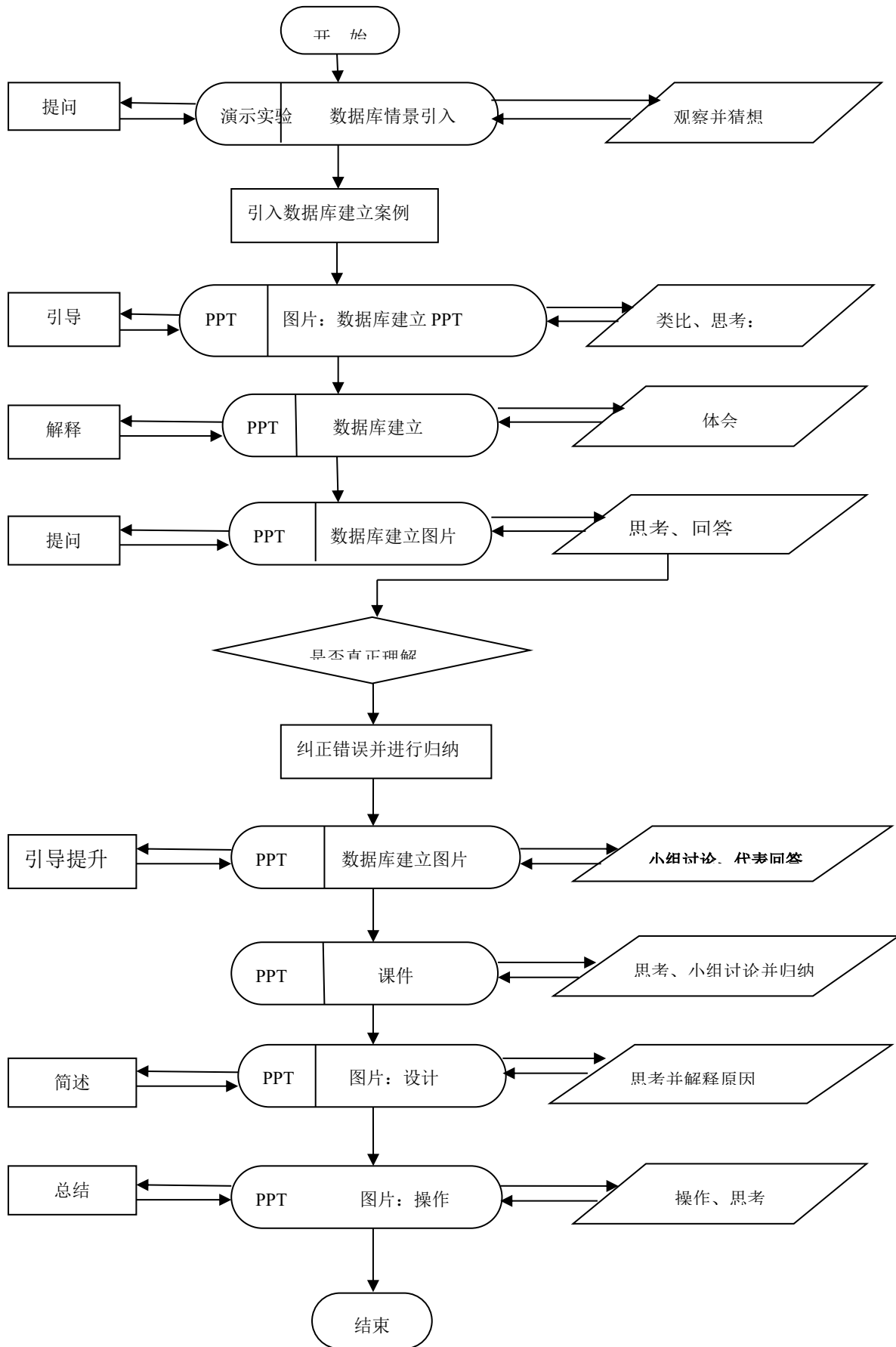
采取的教学手段有：

- （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、教学流程图

教学流程图如下所示。



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。一般由数据库、数据库管理系统、应用系统、数据库管理员和用户构成。数据库的建立是使用数据库的第一步，也是很关键的一步。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。