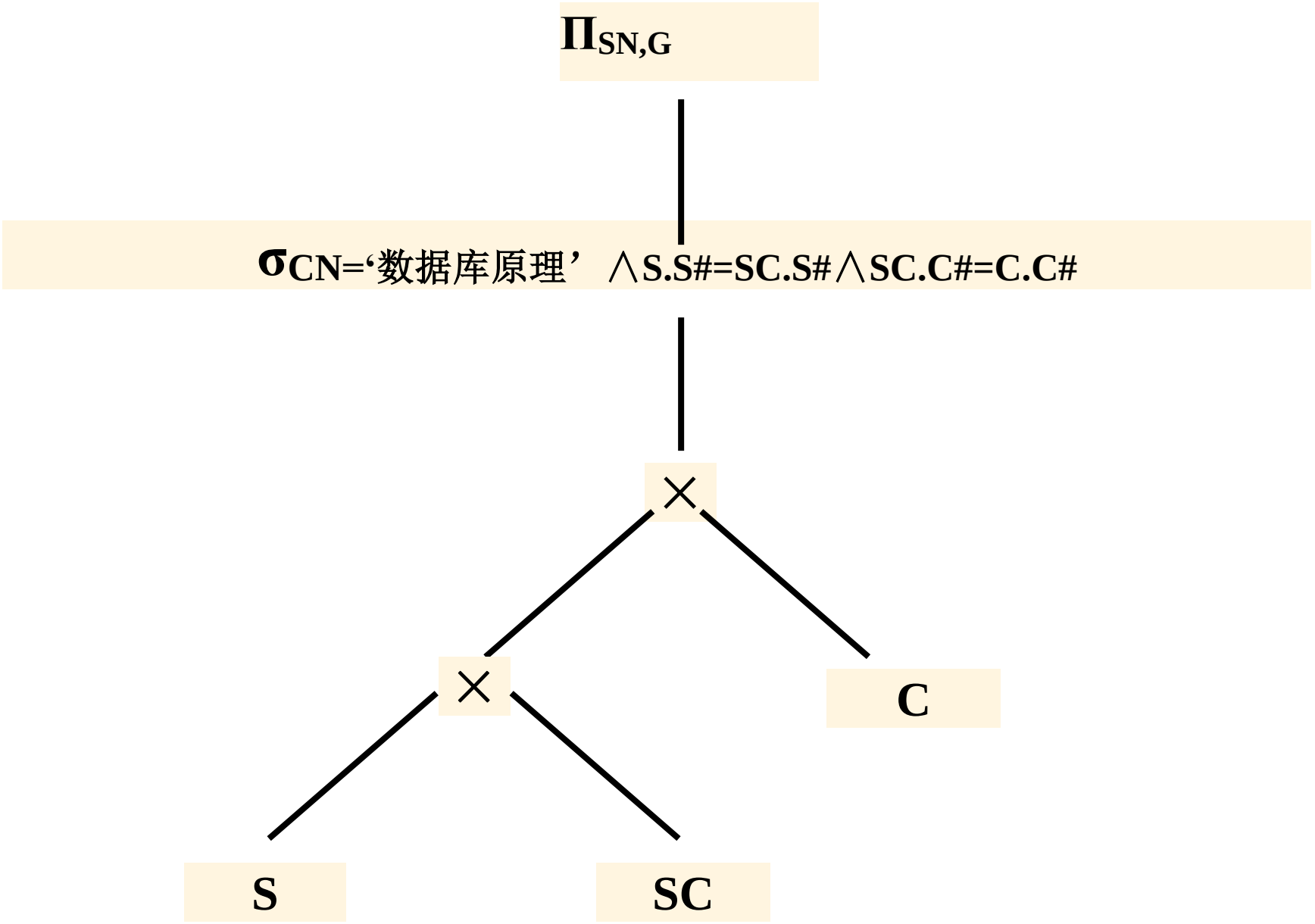
本讲主要内容

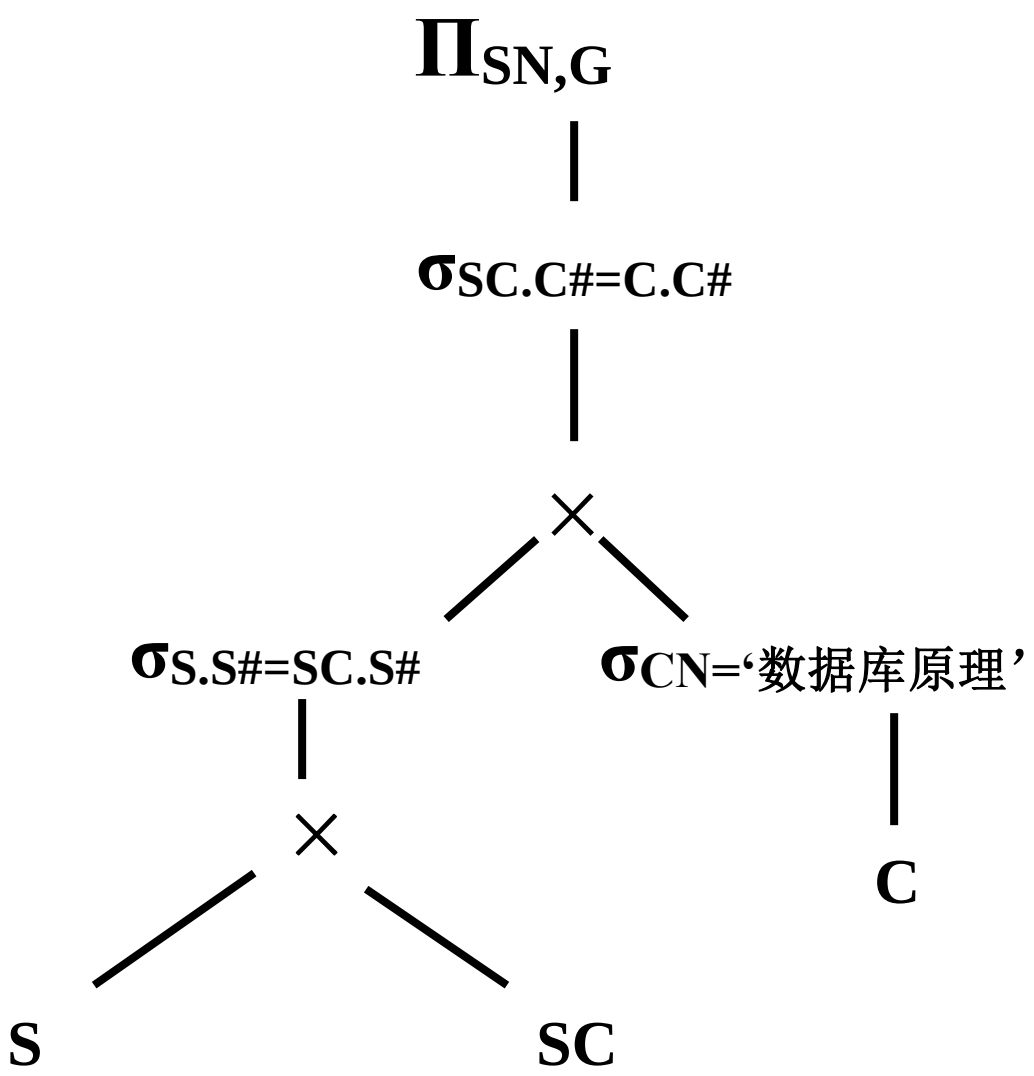
- 关系代数优化（例题）
- 关系系统定义
- SQL概述及定义功能

例：查询选修了数据库原理的学生姓名和成绩

$\pi_{SN,G}$ （$\sigma_{CN=‘数据库原理’}$ （S ⋈ SC ⋈ C））

$\pi_{SN,G}$ （$\sigma_{CN=‘数据库原理’ \wedge S.S\#=SC.S\# \wedge SC.C\#=C.C\#}$（S × SC × C））

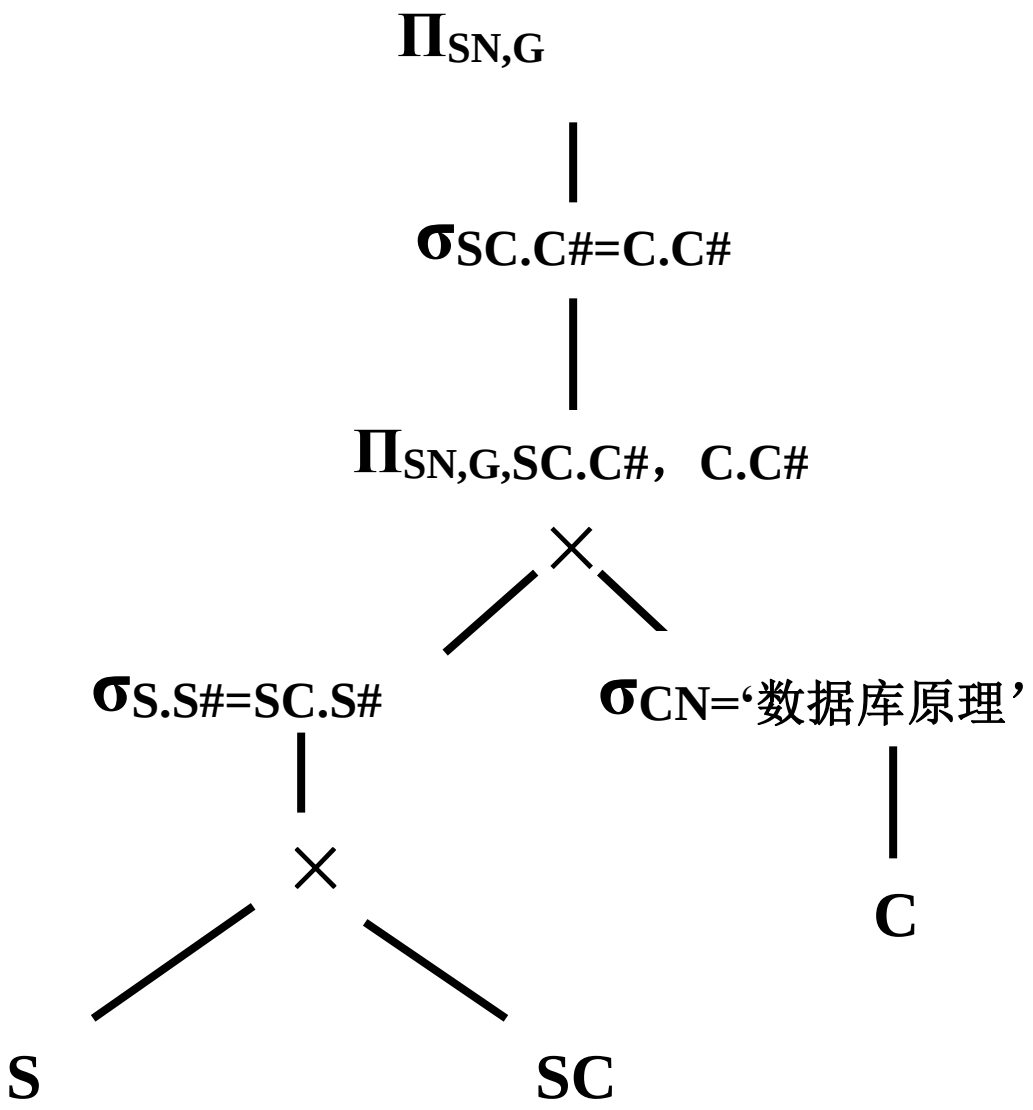

ΠSN,G
σCN=‘数据库原理’∧S.S#=SC.S#∧SC.C#=C.C#
×
×
C
S
SC



规则４、６

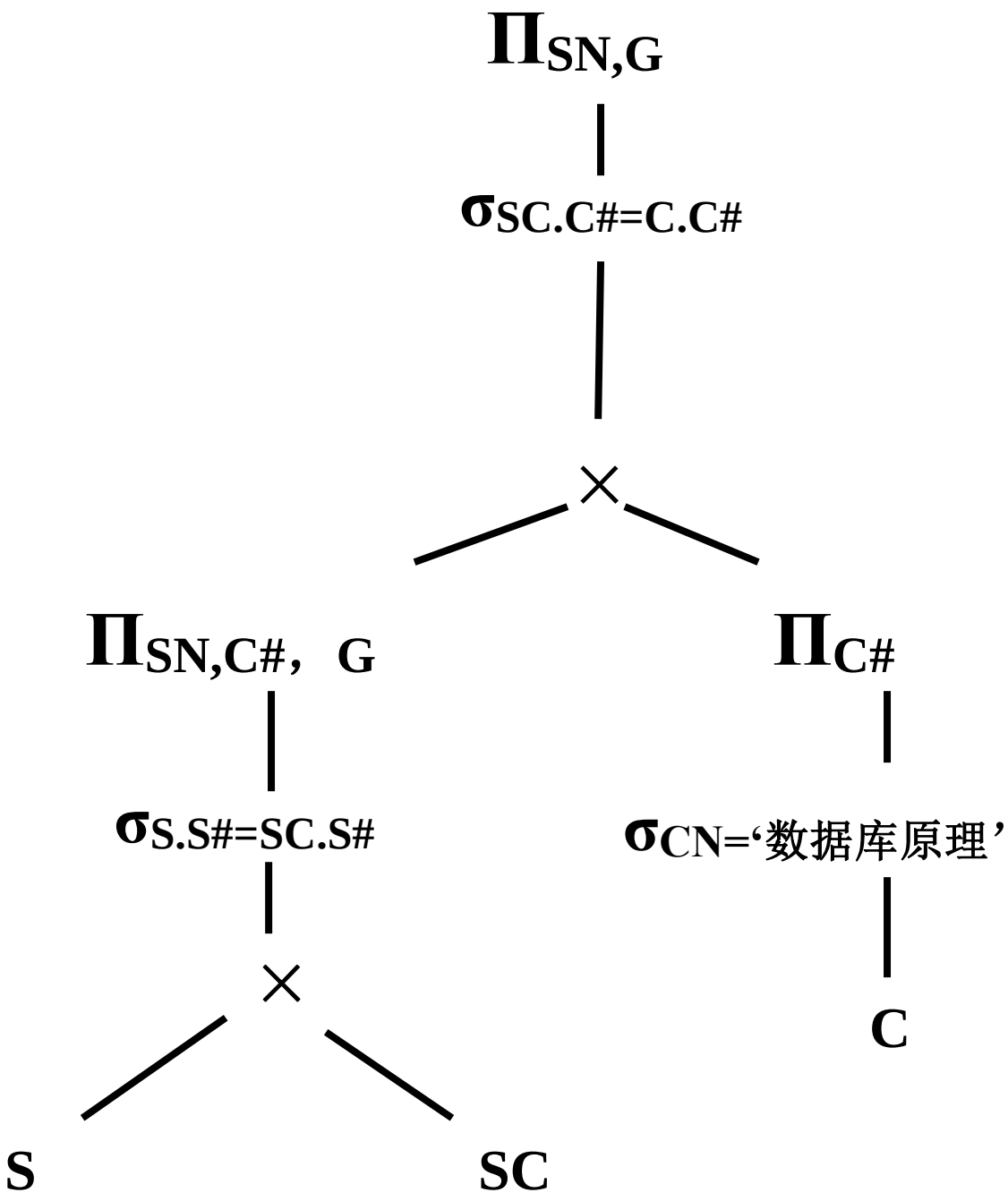
选择的串接

选择和笛卡尔积交换



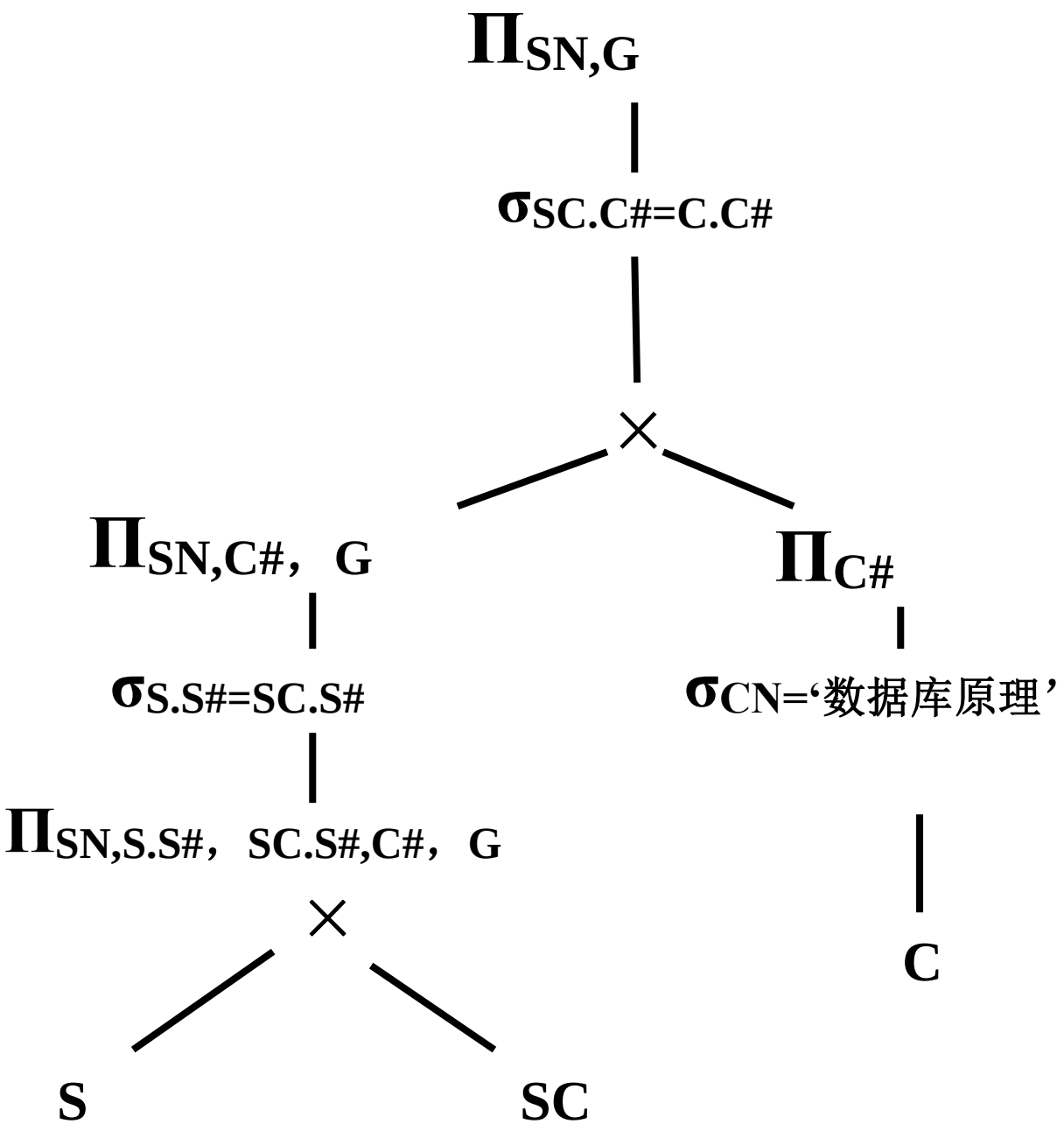
规则 5

选择和投影交换



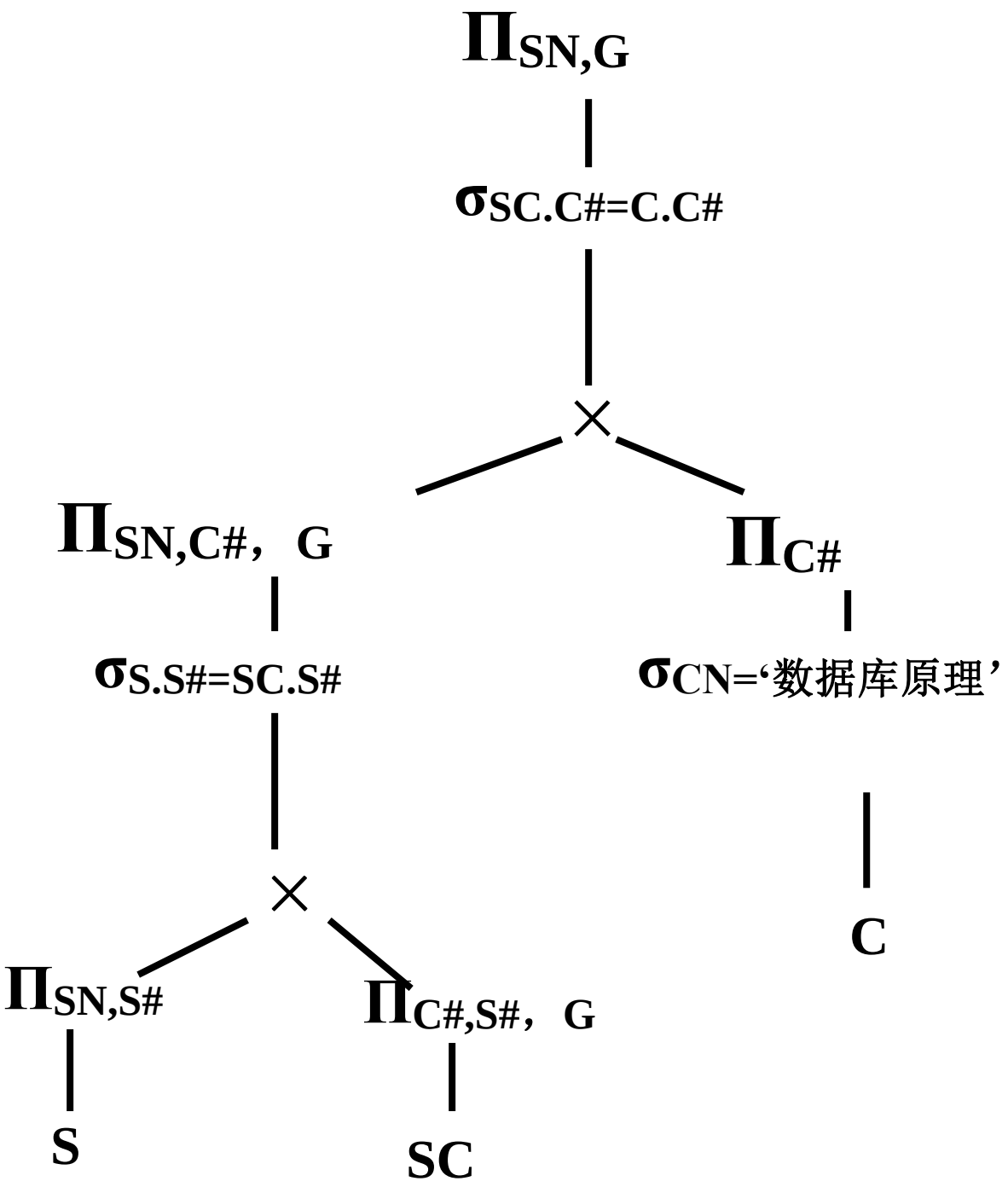
规则 9

投影和笛卡尔积交换



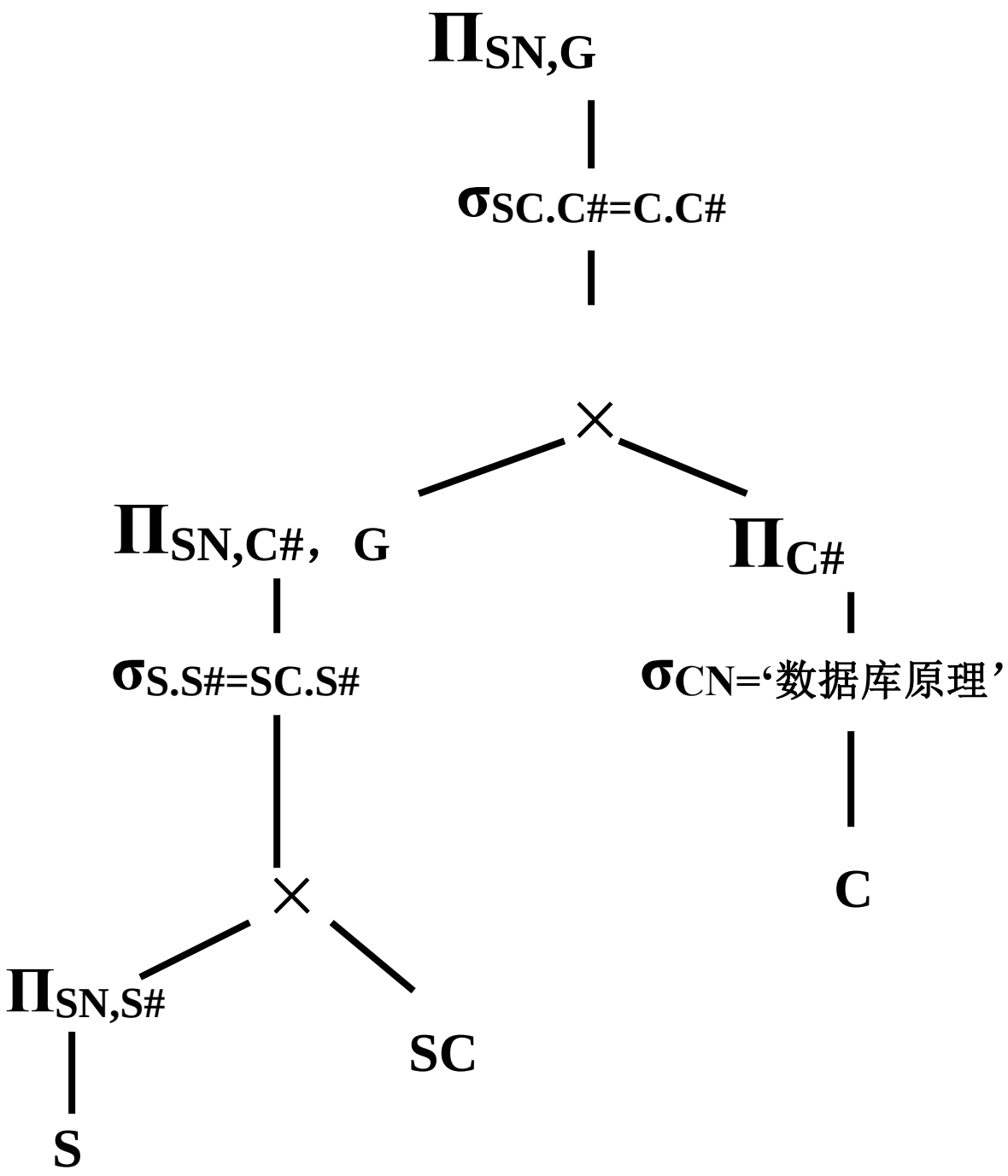
规则 5

选择和投影交换



规则 9

投影和笛卡尔积交换


ΠSN,G
σSC.C#=C.C#
×
ΠSN,C#，G
ΠC#
σS.S#=SC.S#
σCN=‘数据库原理’
C
×
ΠSN,S#
SC
S

2.5 关系系统

关系系统定义

关系系统分类

全关系系统准则

关系模型的三个基本要素?

关系系统的三个基本特征?

关系数据结构

关系操作集合

关系完整性约束

选择

投影

连接

2.5.1 关系系统的定义

一个系统可定义为关系系统，当且仅当它支持：

（1）关系数据结构。

（2）支持选择、投影和（自然）连接运算。对这些运算不必要求定义任何物理存取路径。

2.5.2、关系系统的分类

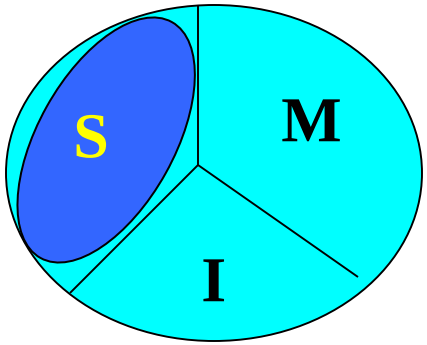
1） 表式系统：仅支持数据结构。

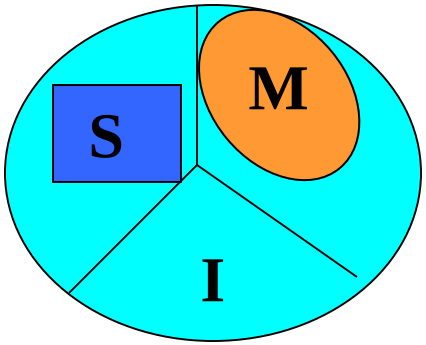
2） (最小)关系系统： 数据结构+三种关系操作。

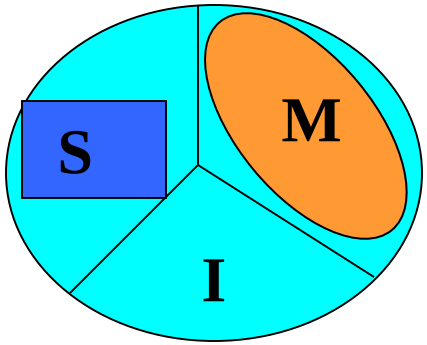
3）关系完备的系统:数据结构+所有关系代数操作。

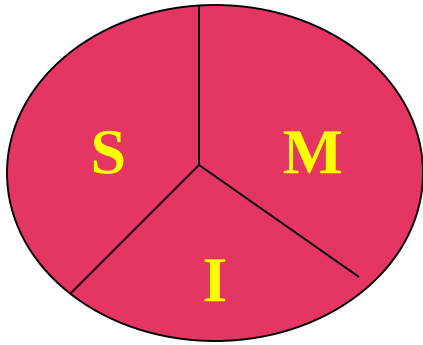
4）全关系系统：支持关系模型的所有特征。

S: 结构　M: 数据操纵　I: 完整性









第三章　关系数据库标准语言SQL

Structured Query language

3.1 SQL概述

一、SQL的发展

- 1974年提出，在SYSTEM R上实现
- 1986年10月，ANSI定为关系数据库语言的美国标准，并公布了标准SQL
- 1987年6月，ISO将其采纳为国际标准
- **SQL92，SQL99，SQL2003， SQL2008**

二、SQL的特点

1、数据定义、操纵、控制等功能一体化

2、两种使用方式，统一的语法结构

3、高度非过程化

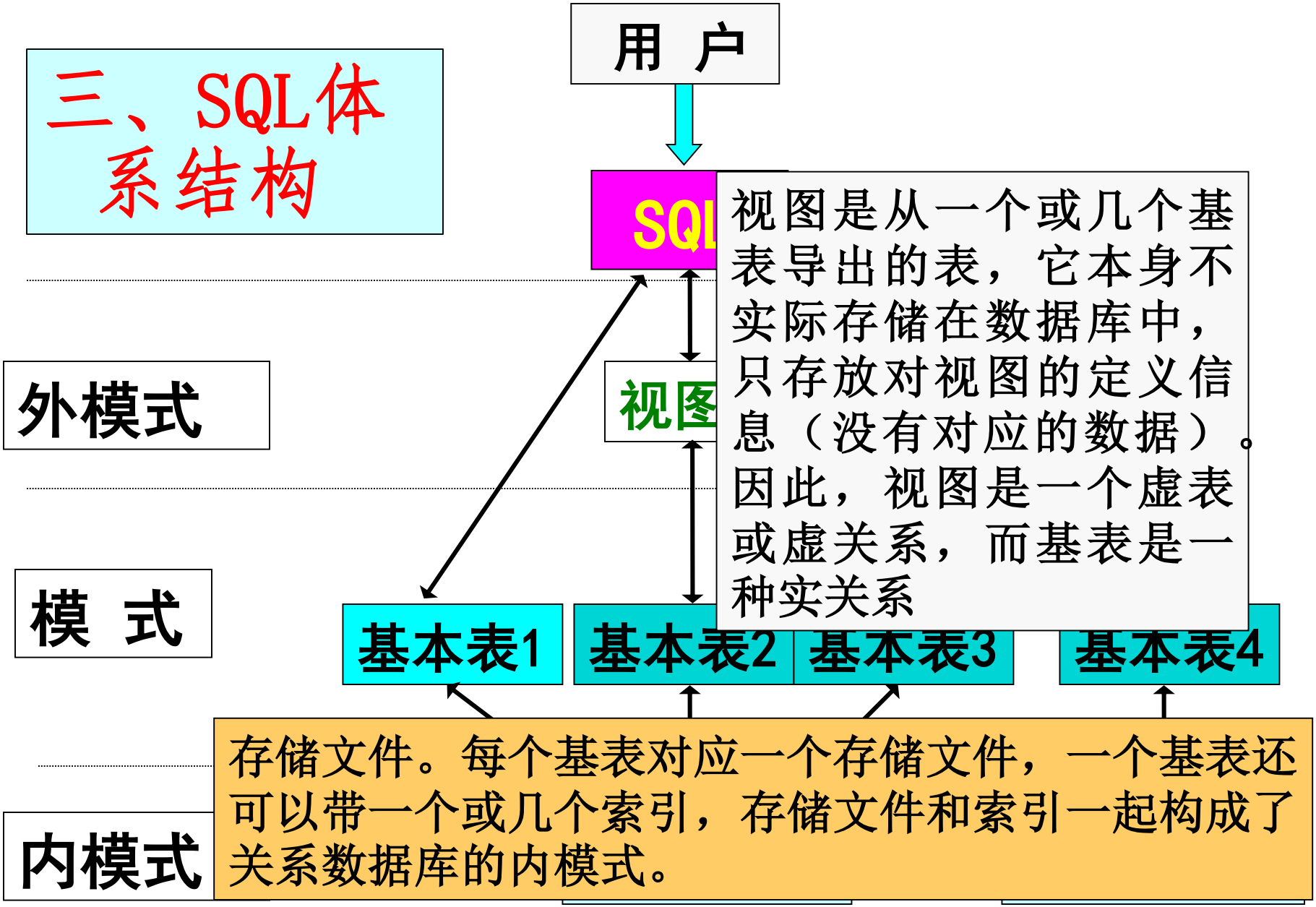
4、语言简洁，易学易用

功　能	动　词
数据库查询	SELECT
数据定义	CREATE，DROP ALTER
数据操纵	INSERT，UPDATE，DELETE
数据控制	GRANT，REVOKE

过程性语言——以关系代数为基础设计出的数据库语言。即用户不但要说明需要什么数据，而且还要说明获得这些数据的过程。

非过程性语言——用户只要说明需要的数据，而如何获得这些数据则不必由用户说明，而由系统来实现。

三、SQL体系结构


用 户
SQL
视图
基本表1
基本表2
基本表3
基本表4
外模式
模 式
内模式
视图是从一个或几个基表导出的表，它本身不实际存储在数据库中，只存放对视图的定义信息（没有对应的数据）。因此，视图是一个虚表或虚关系，而基表是一种实关系
存储文件。每个基表对应一个存储文件，一个基表还可以带一个或几个索引，存储文件和索引一起构成了关系数据库的内模式。

学生-课程数据库

学生表：

Student(Sno,Sname,Ssex,Sage,Sdept)

课程表：

Course(Cno,Cname,Cpno,Ccredit)

学习表：

SC(Sno,Cno,Grade)

3.2 SQL的定义功能

3.2.1 基本表的定义

1、数据类型

1)定长和变长字符串CHAR（n） VARCHAR(n)

2)定长和变长 位串 BIT（n） BITVARING（n）

3)整型数 INT(32位字长） SMALLINT (16位字长）

4)定点数 DECIMAL(p,d)

4)浮点数 FLOAT DOUBLE

5)日期型 DATE “MM-DD-YYYY ”

6)时间型 TIME “HH:MM:SS”

2、基本表的定义

语句格式:

CREATE TABLE <表名>

（<列名1> <类型 1 > [列级完整性约束条件]

[, <列名2> <类型2> [列级完整性约束条件]]

…

[,<表级完整性约束条件>]);

列级完整性约束：

1）“NOT NULL”的属性的值不允许为空值

2）“UNIQUE”表示该属性上的值不得重复；

```
CREATE  TABLE  S
 ( Sno   CHAR(8)   NOT NULL UNIQUE,
   Sname  CHAR(20) NOT NULL,
   Ssex     CHAR(2)  NOT NULL ,
   Sage     SMALLINT,
   Sdept    CHAR(20)
  );
```

3、主关键字的定义

1）在列出关系模式的属性时，在属性及其类型后加上保留字PRIMARY KEY，

2）在列出关系模式的所有属性后，再附加一个声明：

PRIMARY KEY (<属性1>[，<属性2>，...])

说明：如果关键字由多个属性构成，则必须使用第二种方法。

```
CREATE  TABLE  Student
 ( Sno   CHAR(8) NOT NULL UNIQUE,
   Sname  CHAR(20)  UNIQUE,
   Ssex     CHAR(2)  NOT NULL ,
   Sage     SMALLINT,
   Sdept    CHAR(20),
   PRIMARY KEY (Sno)
  );
```

```
CREATE  TABLE  Course
 ( Cno   CHAR(4)   PRIMARY KEY,
    Cname  CHAR(20),
    Cpno     CHAR(4),
    Ccredit     SMALLINT);
```

4、外部关键字的定义

1）如果外部关键字只有一个属性，可以在它的属性名和类型后面直接用“REFERENCES”说明它参照了某个表的某些属性，其格式为：

REFERENCES〈表名〉（<属性>）

2）在CREATE TABLE语句的属性列表后面增加一个或几个外部关键字说明，其格式为：

FOREIGN KEY （<属性>） REFERENCES〈表名〉（<属性>）

```
CREATE  TABLE  Course
 ( Cno   CHAR(4)   PRIMARY KEY,
    Cname  CHAR(20),
    Cpno     CHAR(4)  REFERENCES
                          Course (Cno),
    Ccredit     SMALLINT);
```

```
CREATE  TABLE  SC
  ( Sno   CHAR(8) REFERENCES
                          Student(Sno) ,
    Cno  CHAR(4) REFERENCES
                          Course(Cno) ,
    Grade SMALLINT,
    PRIMARY KEY(Sno,Cno)
   );
```

```
CREATE  TABLE  SC
  ( Sno   CHAR(8),
    Cno  CHAR(4),
    Grade SMALLINT,
    PRIMARY KEY(Sno,Cno),
    FOREIGN KEY Sno REFERENCES
                        Student(Sno),
    FOREIGN KEY Cno REFERENCES
                        Course(Cno)
   );
```

3.2.2 基本表的修改和删除

ALTER TABLE <表名>

[**ADD** <列名><类型>[完整性约束]]

[**DROP** <列名>[<完整性约束名>]]

[**ALTER COLUMN** <列名> <类型>] ;

NOT NULL ; UNIQUE ; PRIMARY KEY ; FOREIGN KEY; CHECK ; DEFAULT

删除表的列的格式为：

ALTER TABLE 表名 DROP Column 列名

Alter table student drop Column Sdept;

```
create table mydb1
(id char(4) not null,
 sname char(32),
 deptid char(32));
```

增加名称必须取唯一值的约束条件

```
ALTER  TABLE mydb1
     ADD  Constraint  UQ_name UNIQUE (name) ;
```

添加表的主键约束：

Alter table mydb1 Add Constraint PK_id primary key (id)

添加表的外约束：

Alter table mydb1 Add Constraint FK_deptid foreign key (deptid) references Dept (deptid)

```
Create table myTB1 (
id nvarchar(4) not null,
name nvarchar(32),
Sex  nvarchar(4),
deptid nvarchar(32))
```

添加表的Check约束：

Alter table myTB1 Add Constraint CK_sex

Check (sex='男'or sex='女');

```
Create table myTB1 (
id nvarchar(4) not null,
name nvarchar(32),
Sex  nvarchar(4),
deptid nvarchar(32))
```

添加表的Default约束：

Alter table myTB1 add Constraint DF_sex Default ('F')
for sex

删除表的操作语句格式为：

DROP TABLE <表名> [RESTRICT|CASCADE];

3.2.3 索引的建立和删除

1、建立索引

CREATE
[UNIQUE][CLUSTERED|NOCLUSTERED]
INDEX <索引名> ON <表名> (<列名> [<次序>]
[, <列名> [<次序>]…])

CREATE UNIQUE INDEX U_idx_cname ON Course(Cname);

CREATE UNIQUE INDEX SCno ON SC(Sno asc,Cno desc);

MySQL的索引

- 单列索引(普通索引，唯一索引，主键索引)、组合索引、全文索引引擎为MyISAM、空间索引MyISAM

explain select * from student where sno='1’;

查看是否使用索引

3.2.3 索引的建立和删除

2、删除索引

DROP INDEX <索引名> ON <表名>

例：

drop INDEX stusno on student ;