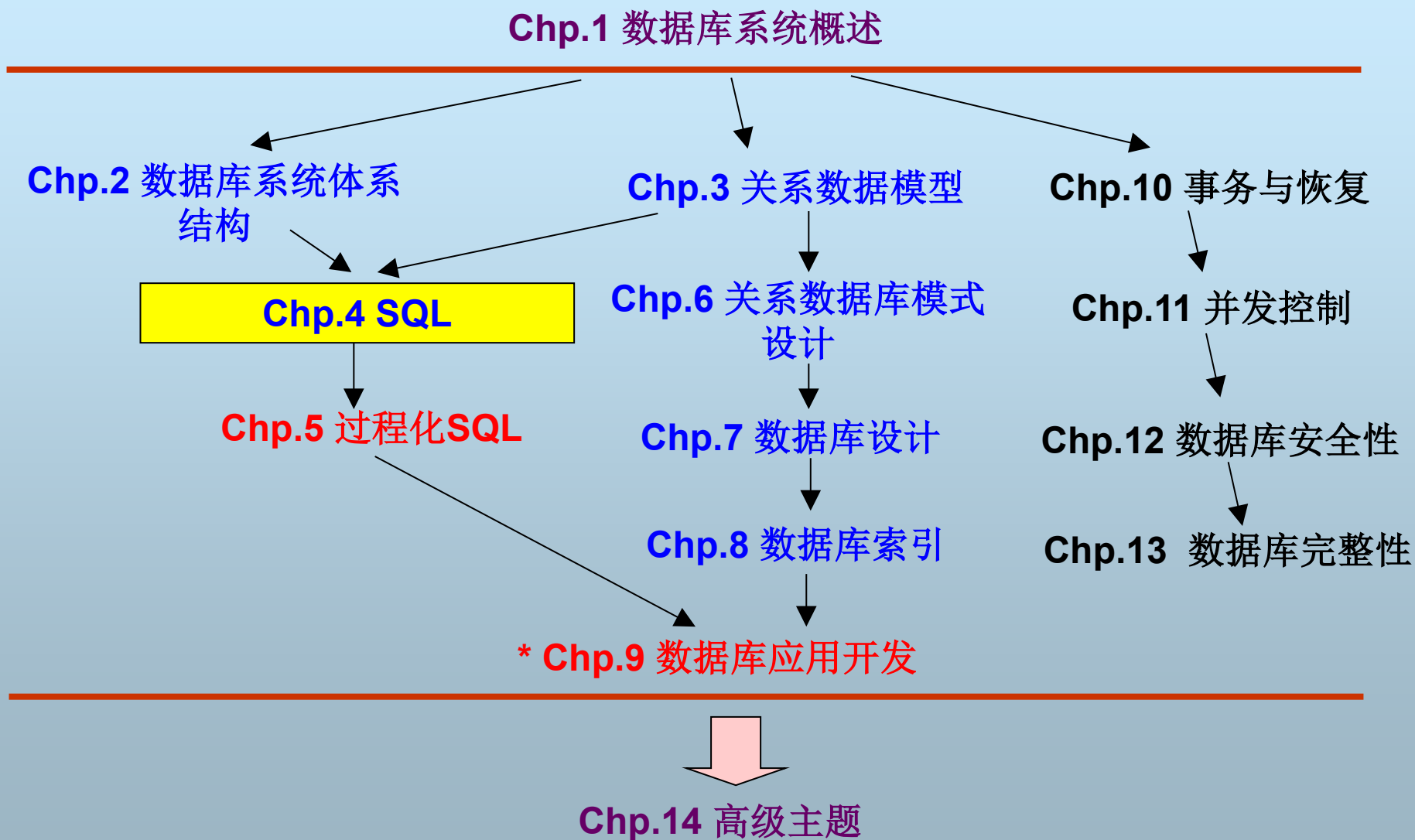


第4章 关系数据库语言SQL

课程知识结构

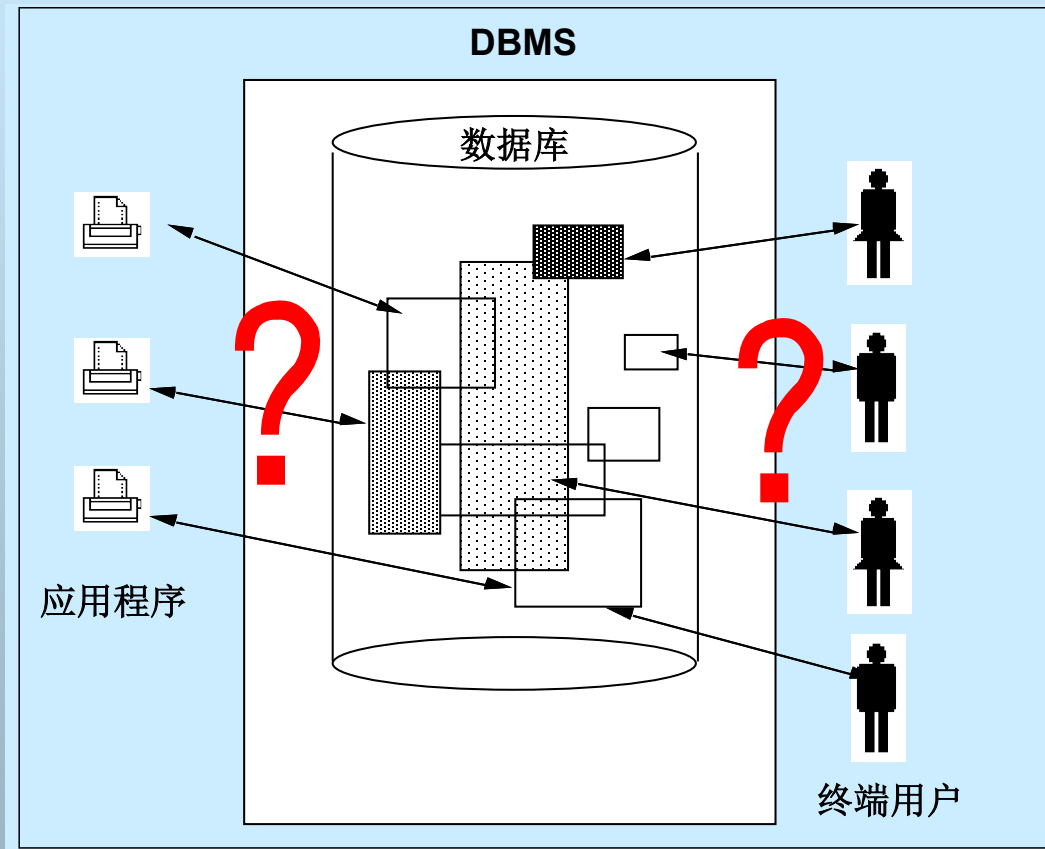


本章主要内容

- 数据库语言
- SQL概述
- SQL DDL
- SQL DML
- 视图

一、数据库语言

■ 用户如何存取数据库中的数据？需要存取哪些数据？

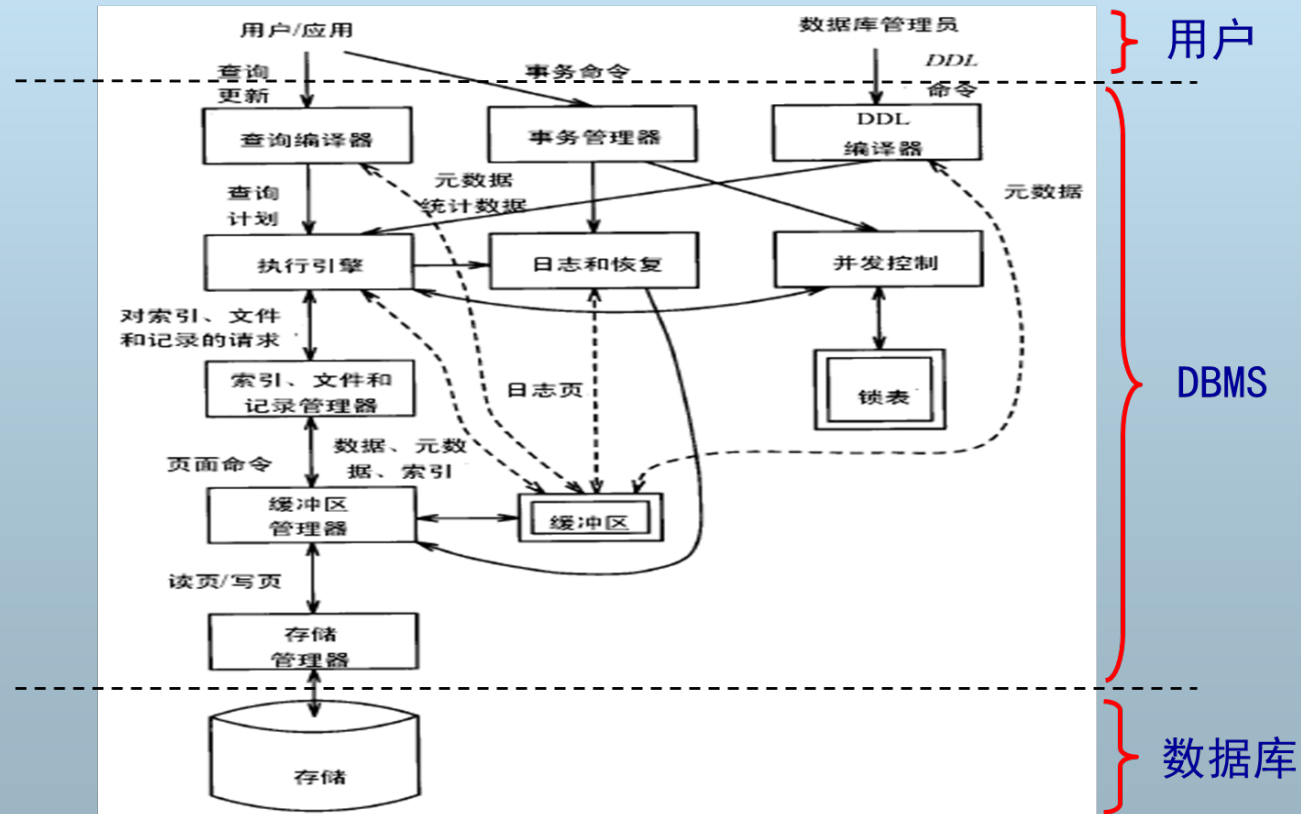


■ 需存取三类数据

- 数据库的存取？
- 数据库模式的存取？
- 数据库访问控制信息的存取？

一、数据库语言

- 用户与数据库的唯一接口——数据库语言
- 用户通过数据库语言进行数据存取



一、数据库语言

■ 数据库语言包括三类子语言

- **数据定义语言** (Data Definition Language, DDL)
——存取数据库模式
- **数据操纵语言** (Data Manipulation Language, DML)
——存取数据库数据
- **数据库控制语言** (Data Control Language, DCL)
——存取访问控制信息

二、SQL概述

- SQL的发展历程
- SQL数据库中的术语
- SQL数据库的三级体系结构
- SQL的组成

1、SQL的发展历程

- 1972: IBM开始研究System R系统, 配置了数据库语言SQUARE
 - SQUARE (Specifying Queries As Relational Expressions)
- 1974: Boyce和Chamberlin将SQUARE修改为SEQUEL
 - SEQUEL (Structured English QUery Language)
 - 去掉了数学符号, 以英语单词和结构式语法代替查询
 - 后简称为SQL (Structured Query Language)

SQL主流发音/'si:kw(ə)l/, 很少发音Ess-cue-ell, 不能发音成circle/'s3:k(ə)l/

1、SQL的发展历程

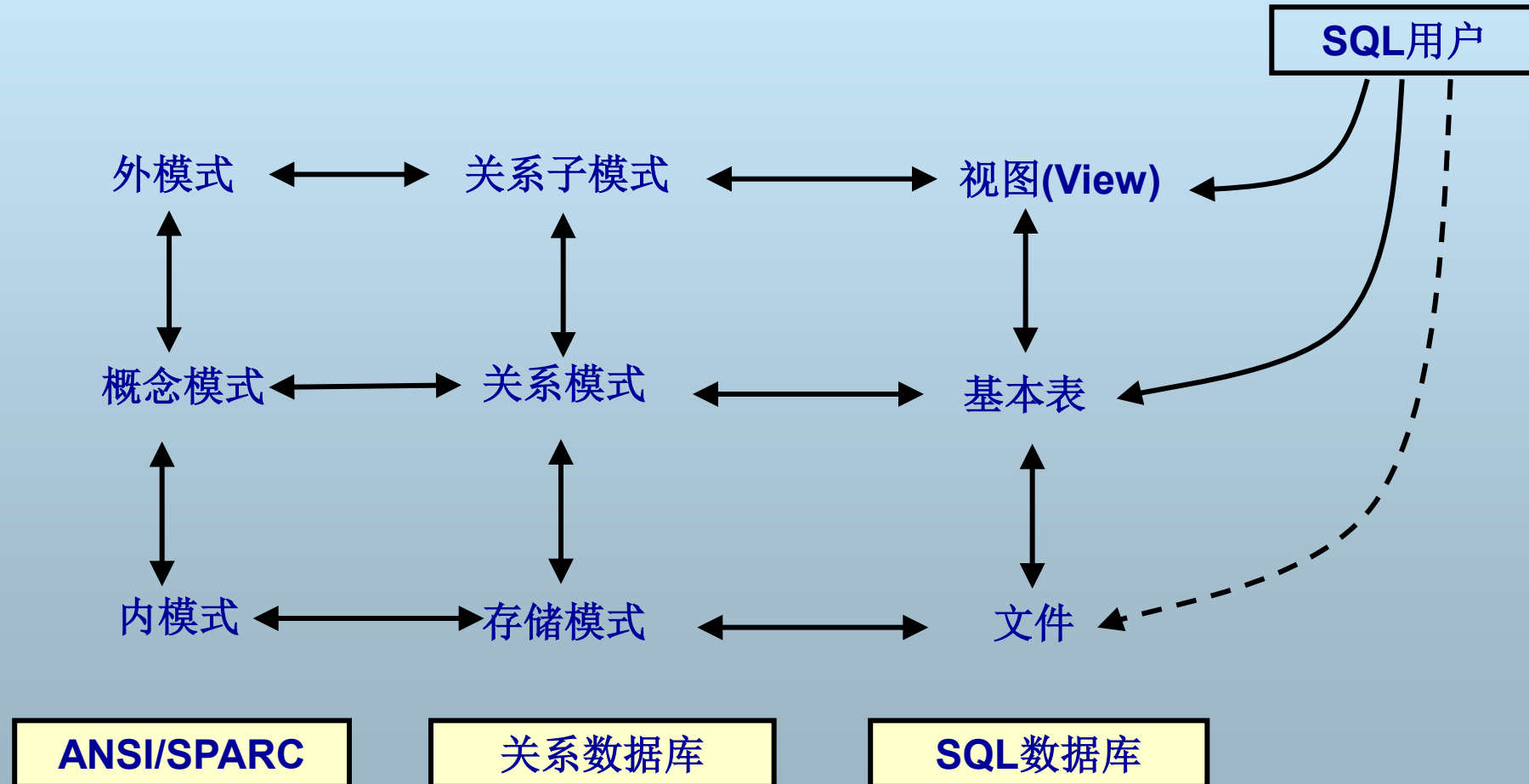
- **1970s末起**：主流的数据库厂商纷纷在其产品中支持SQL
 - **Oracle、DB2、Sybase等**
- **1986年**，ANSI X3.135-1986，ISO/IEC 9075:1986，**SQL-86**
- **1989年**，ANSI X3.135-1989，ISO/IEC 9075:1989，**SQL-89**
- **1992年**，ANSI X3.135-1992，ISO/IEC 9075:1992，**SQL-92 (SQL2)**
- **1999年**，ISO/IEC 9075:1999，**SQL:1999 (SQL3)**
- **2003年**，ISO/IEC 9075:2003，**SQL:2003**
- **2008年**，ISO/IEC 9075:2008，**SQL:2008**
- **2011年**，ISO/IEC 9075:2011，**SQL:2011**

2、SQL数据库中的术语

- 基本表 (Table) ———— 关系
 - 简称“表”。 表结构 ———— 关系模式
- 记录 (Record) ———— 元组
- 字段 (列) (Field/Column) ———— 属性
- 字段值 ———— 属性值
- 字段类型 (列类型) ———— 域
- 键 (Key) ———— 码
- 主键 (Primary Key) ———— 主码
- 外键 (Foreign Key) ———— 外码

3、SQL数据库的三级体系结构

■ SQL数据库：支持SQL语言的关系数据库



4、SQL的组成



三、SQL的数据定义——DDL

- 基本表的结构
- 创建基本表: **Create Table**
- 修改基本表: **Alter Table**
- 删除基本表: **Drop Table**

1、基本表的结构

■ 一个基本表的结构包括：

- 表名 —— 对应关系模式名
- 列 —— 对应关系模式的属性
- 完整性约束 —— 对应关系模式的三类完整性

对应关系模式的形式化定义：

$R(U, D, \text{dom}, F)$

(1) 列

■ 列名

- 字母开头，可含字母、数字、_等
- **<=30**字符

■ 列类型

- **Char(n)** **【定长字符串类型】**
- **Varchar(n)** **【可变长字符串类型】**
- **Int/Decimal** **【数值型】**
- **Date/Time** **【日期时间型】**
-

常用的列类型

	ANSI/ISO	MySQL	Oracle
字符型	Char(n)	Char(n)	Char(n)
	Character(n)		
	Varchar(n)	Varchar(n)	Varchar(n)
	Char Varying(n)		Varchar2(n)
数值型	Numeric	Numeric	Number
	Decimal	Decimal	
	Integer	Integer	
	Int	Int	
	Float	Float	
	Double	Double	
	Real	Real	
日期型	Date	Date	Date
	Time	Time	
		Datetime	

(2) 完整性约束

■ 主键约束 (Primary Key)

- 定义主码, 实体完整性

■ 唯一键约束 (Unique)

- 定义候选码

■ 外键约束 (Foreign Key)

- 定义外码, 参照完整性

■ 检查约束 (Check)

- 用户自定义完整性

这些约束既可以定义在列上，也可以定义在基本表之上

列约束：在每列后定义，只对当前列有效

表约束：在全部列定义后定义，可定义多个列上的约束

2、创建基本表

- 基本表构成：表名，列和约束
- **Create Table** <基本表名> (
 列名1 列类型1 [列约束1],
 列名2 列类型2 [列约束2],

 [表约束]
)
 - 定义列
 - 定义约束

在SQL数据库中，不一定必须定义主键，这与关系模型有差别

```
Create Table Student(  
    Sno Varchar(10) Constraint PK Primary Key,  
    Sname Varchar(20),  
    Age Int,  
    Sex Char(1)  
)
```

(1) 定义列

■ 完整格式

- <列名> <列类型> [**DEFAULT** <默认值>] [[**NOT**] **NULL**] [<列约束>]

```
Create Table Student(  
    Sno Varchar(10) Constraint PK Primary Key,  
    Sname Varchar(20) NOT NULL,  
    Age Int,  
    Sex Char(1) DEFAULT 'F'  
)
```

NOT NULL表示不允许空值，实际上是**Check**约束的简化

A) 默认值

- 当往表中插入一条新记录时，如果某列上有默认值，并且新记录中未指定该列的值，则自动以默认值填充

```
Insert Into Student(sno,sname,age)  
Values('001','John',20)
```

插入一条
新记录

Sno	Sname	Age	Sex
001	John	20	F

自动以默认
值填充

B) 列约束

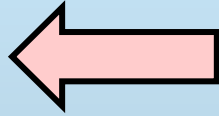
- 必须在每个列定义后定义
- 只对当前列有效
- 可以使用四种类型的约束
- 格式
 - **[Constraint <约束名>] <约束类型>**
- 例
 - **Sno char(n) Constraint **PK_Student** Primary Key**
 - **Sno char(n) Primary Key**

Where are we?

■ 创建基本表

- 定义列

- 定义约束



(2) 定义约束

- 列约束：在每个列后定义，可以有多个约束子句
 - 但不能定义多个列上的约束
- 表约束：在全部列定义完成后定义，可以有多个约束子句
 - 多个列上的约束必须使用表约束
 - 单列上的约束可以用列约束，也可用表约束
- 四种约束都可以作为列约束或表约束

A) 列约束和表约束举例

```
Create Table Student(  
    Sno Varchar(10) Constraint PK_S Primary Key,  
    Sname Varchar(20),  
    Age Int Constraint CK_S Check (age>14 and age<100),  
    Sex Char(1),  
    Constraint UQ_S Unique(Sname),  
    Constraint CK_SS Check (Sex IN ('M','F'))  
)
```


B)Primary Key约束

- 定义主键：不许有空值，也不可重复

```
Create Table Student(
```

```
  Sno Varchar(10) Constraint PK_S Primary Key,
```

```
  Sname Varchar(20),
```

```
  Age Int ,
```

```
  Sex Char(1))
```



一个表只能
有一个主键
!!!

```
Create Table SC( ——选课表
```

```
  Sno Varchar(10)
```

```
  Cno Varchar(20),
```

```
  Score Int,
```

```
  Constraint PK_SC Primary Key(Sno,Cno))
```

C)Unique约束

- 唯一性约束：值不可重复，但可以为空

```
Create Table Department(  
    NO Varchar(10),  
    NAME Varchar(20),  
    SCHOOL Char(20),  
    Constraint UQ_D Unique(NAME, SCHOOL)  
)  
)
```

多个列上的约束只能
能用表约束来实现

D)Unqiue约束对空值的处理

若约束列中有一列不为空，就实施约束；若约束列都为空，则不实施约束

NO	NAME	SCHOOL		
1	管理系	商学院		
2	管理系	管理学院	OK	值唯一
3	管理系	管理学院	Error!	值重复
4	管理系		OK	值唯一
5		管理学院	OK	值唯一
6		管理学院	Error!	实施约束
7			OK	约束列都空，
8			OK	不实施约束

E)Foreign Key约束

- 外键约束：表中某列值引用其它表的主键列或Unique列，
 - 实施参照完整性

```
Create Table Student(  
    Sno Varchar(10) Constraint PK_S Primary Key,  
    Sname Varchar(20),  
    Age Int)
```

```
Create Table SC( ——选课表  
    Sno Varchar(10) Constraint FK_SC References Student(Sno),  
    Cno Varchar(20),  
    Score Float ,  
    Constraint FK_SC Foreign Key(Sno) References Student(Sno))  
)
```

F)Foreign Key约束示例

被参照表（主表）：Student表

Sno	SNAME	AGE
001	John	20
002	Rose	21

参照表（子表）：SC表

Sno	Cno	Score
001	c001	90

Insert Into SC values('003','c001',85); -- Error!!

Delete From Student where Sno='001'; -- Error!!

在子表中（如SC）插入记录时，若主表中对应的列值不存在，则插入出错；

删除主表中的记录时，若有子表中的相应记录存在，也出错。
——若设置了级联删除则不会出错

G) Foreign Key约束的选项

- **级联删除**：删除主表中的记录时，同时删除子表中相关联的记录：
On Delete Cascade
- **级联设空**：删除主表中的记录时，同时将子表中相应记录的外键列值设为空：
On Delete Set NULL

Create Table SC(——选课表

Sno Varchar(10) ,

Cno Varchar(20),

Score Number(3) ,

Constraint FK_SC Foreign Key(Sno) References Student(Sno) On Delete Cascade

)

H)Check约束

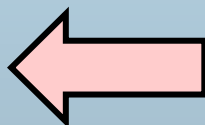
■ 检查约束：自定义某些列上的约束

- **Constraint CK_S1 Check (Age>15)**
- **Constraint CK_S2 Check (Sex In ('M','F'))**
- **Constraint CK_SC Check (Score>=0 and Score<=100)**
- **Constraint CK_S3 Check (Sname Is Not NULL)**

Where are we?

■ SQL的数据定义——DDL

- 基本表的组成
- 创建基本表
 - ◆ 定义列
 - ◆ 定义约束
- 修改基本表
- 删除基本表



2、修改基本表

- **Alter Table <表名> -- 表名不能修改**
 - [Add <列定义>] |**
 - [Modify <列定义>] |**
 - [Rename Column <old> To <new>] |**
 - [Drop Column <列名>] |**
 - [Add <表约束>] |**
 - [Drop Constraint <约束名>] |**
 - [Rename To <new_table_name>]**

(1) 增加列

- **Alter Table <表名>
Add <列定义>**
- **<列定义>与Create Table中相同**

Alter Table Student

Add Class Varchar (10)

Alter Table Student

Add Dept Varchar (10) Constraint UQ_S3 UNIQUE

(2) 删除列

- **Alter Table <表名>
Drop Column <列名>**

Alter Table Student

Drop Column age

(3) 修改列

- **Alter Table** <表名>
Modify <列定义>
- <列定义>与**Create Table**中相同
 - 但列名不能用**Modify**修改

```
Alter Table Student
```

```
Modify age Integer NOT NULL
```

(4) 重命名列

- **Alter Table <表名>
Rename Column <old> To <new>**

Alter Table Student

Rename Column sex To gender

(5) 增加约束

- **Alter Table <表名>
Add <表约束>**
- 只能增加表约束
- 表约束格式与创建表时相同

Alter Table Student

Add Constraint PK_Student Primary Key(Sno)

(6) 删除约束

■ Alter Table <表名> Drop Constraint <约束名>

Create Table SC(——选课表

Sno Varchar(10) ,

Cno Varchar(20),

Score Float ,

Constraint FK_SC Foreign Key(Sno) References Student(Sno) On Delete Cascade

)

Alter Table SC

Drop Constraint FK_SC

(7) 重命名表

■ Alter Table <表名> Rename To <新的表名>

```
Create Table SC( --选课表
  Sno Varchar(10) ,
  Cno Varchar(20),
  Score Float ,
  Constraint FK_SC Foreign Key(Sno) References Student(Sno) On Delete Cascade
)
```

```
Alter Table SC
  Rename To course_selection
```


3、删除基本表

- **Drop Table <表名> [Cascade Constraints]**
- **Cascade Constraints**表示删除表时同时删除该表的所有约束

```
Drop Table Student;  
Drop Table Student Cascade Constraints
```

Next

- 数据库语言
- SQL概述
- SQL DDL
- SQL DML
- 视图

