

数据管理基础

第3章 关系数据库标准语言SQL

(SQL概述 & 数据定义)

智能软件与工程学院



- ❑ 3.1 SQL概述
- ❑ 3.2 学生-课程数据库
- ❑ 3.3 SQL数据定义
- ❑ 3.4 SQL数据查询
- ❑ 3.5 SQL数据更新
- ❑ 3.6 空值的处理
- ❑ 3.7 视图

❑ 3.1 SQL概述

- SQL的产生与发展
- SQL的特点
- SQL中的基本概念

❑ 3.2 学生-课程数据库

❑ 3.3 SQL数据定义

- 模式的定义与删除
- 基本表的定义、删除与修改
- 索引的建立与删除
- 数据字典

数据管理基础

3.1 SQL概述

智能软件与工程学院

❑ SQL (Structured Query Language) 结构化查询语言，是关系数据库的标准语言

❑ SQL是一个通用的、功能极强的关系数据库语言

- 综合统一
- 高度非过程化
- 面向集合的操作方式
- 以同一种语法结构提供两种使用方法
- 语言简洁，易学易用

□综合统一

- 集数据定义语言（DDL），数据操纵语言（DML），数据控制语言（DCL）功能于一体。
- 可以独立完成数据库生命周期中的全部活动：
 - 定义和修改、删除关系模式，定义和删除视图，插入数据，建立数据库；
 - 对数据库中的数据进行查询和更新；
 - 数据库重构和维护
 - 数据库安全性、完整性控制，以及事务控制
 - 嵌入式SQL和动态SQL定义
- 用户数据库投入运行后，可根据需要随时逐步修改模式，不影响数据库的运行。
- 数据操作符统一

□高度非过程化

- 非关系数据模型的数据操纵语言“面向过程”，必须指定存取路径。
- SQL只要提出“做什么”，无须了解存取路径。
- 存取路径的选择以及SQL的操作过程由系统自动完成。

□面向集合的操作方式

- 非关系数据模型采用面向记录的操作方式，操作对象是一条记录
- SQL采用集合操作方式
 - 操作对象、查找结果可以是元组的集合
 - 一次插入、删除、更新操作的对象可以是元组的集合

SQL的特点 3

□以同一种语法结构提供多种使用方式

- SQL是独立的语言，能够独立地用于联机交互的使用方式
- SQL又是嵌入式语言，能够嵌入到高级语言（例如C，C++，Java）程序中，供程序员设计程序时使用

□语言简洁，易学易用

- SQL功能极强，完成核心功能只用了9个动词：
 - 数据定义：CREATE，DROP，ALTER
 - 数据查询：SELECT
 - 数据操作：INSERT，UPDATE，DELETE
 - 数据控制：GRANT，REVOKE

从关系模型 到 SQL语言

□ 基本概念的变化

关系模型	关系	子模式	属性	元组
SQL	基本表 (base table)	视图 (view)	列 (column)	行 (row)

视图也被称为‘虚表’
(virtual table)

两者统称为‘表’
(table)

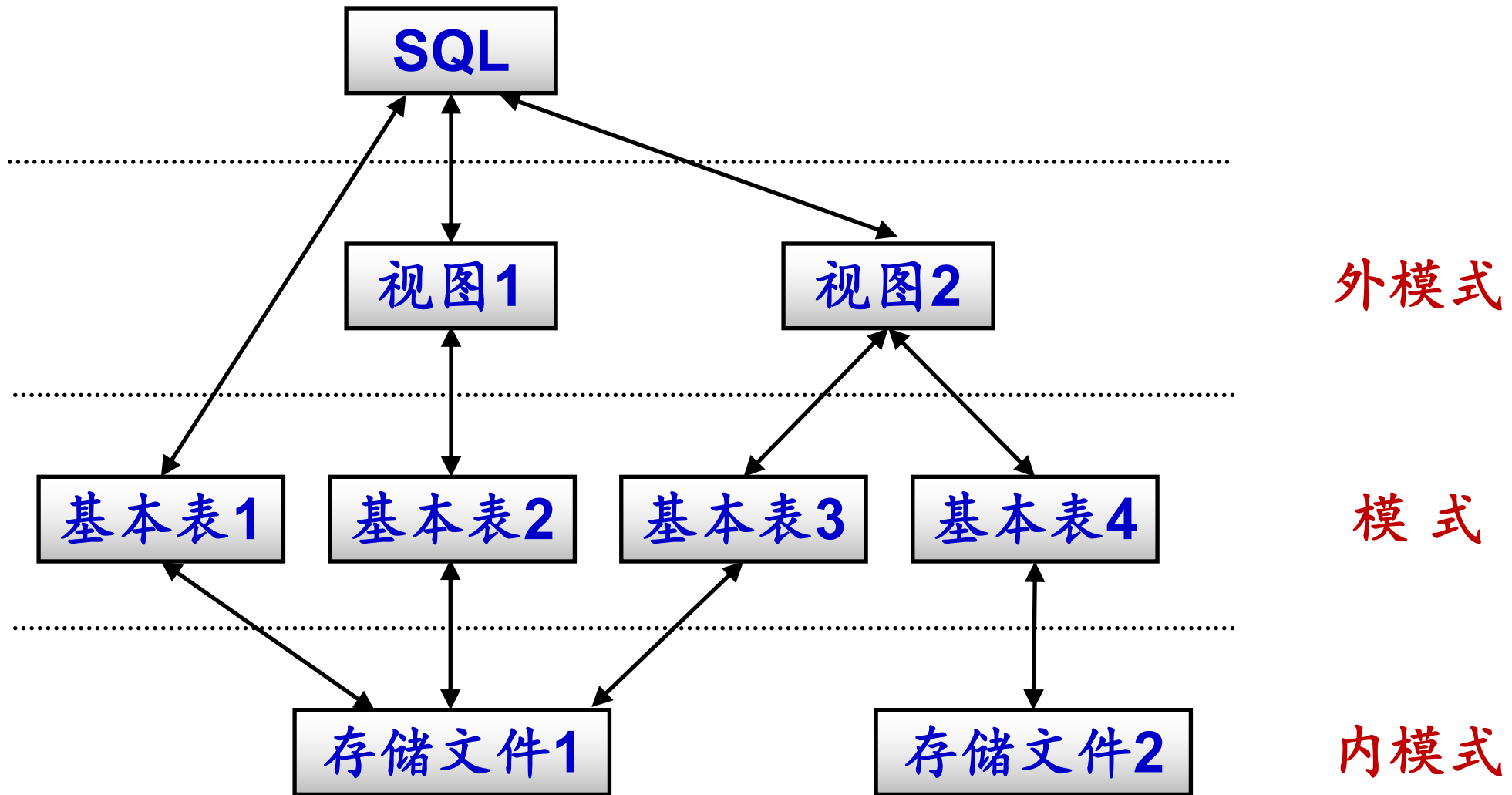
SQL语言的基本语言成分

- **符号**：26个英文字母，阿拉伯数字，括号，四则运算，.....
- **保留字**：在SQL语言中具有特定含义的语言成分，通常是一个英文单词（或缩写），通常用于标识一条SQL语句的组成成分，也包括系统内置的系统表、存储过程/函数、SQL函数等等，例如：
 - 用于标识语句类型：create, select, insert, alter,
 - 用于标识对象类型：table, view, procedure, trigger,
 - 用于标识语句成分：unique, primary key, with check option,
 - 数据类型，内置函数，.....
 - 可编程SQL中的流程控制命令：if...then...else, while..., for ...
- **标识符**：用于对由用户创建的数据库对象进行命名，包括表、视图、属性、存储过程/函数、触发器、变量.....
- **常量**：数值常量、字符（串）常量、日期/时间常量

SQL语言的基本表示规范（交互式SQL）

- ❑ 一条完整SQL语句，通常以命令动词开始，以**分号** ';' 作为结束符
 - 在交互式SQL执行窗口中，可以一次只执行一条SQL语句，也可以一次执行多条SQL语句（批处理）
 - 在批处理执行方式下，分号既作为前一条SQL语句的结束符，也可以看做是不同SQL语句之间的分隔符
- ❑ 除常量外，SQL语言中的其他语言成分仅支持西文字符，且（字母）不区分大小写
 - 保留字、表名、列名等都不区分大小写（在个别数据库系统中有例外）
- ❑ 数值类型的常量不需要定界符，但字符（串）或日期/时间类型的常量需要使用 **‘单引号’** 作为定界符，可支持不同的日期显示格式。

SQL与关系数据库三级模式 1



□基本表

- 本身独立存在的表
- 一个关系对应一个基本表
- 一个（或多个）基本表对应一个存储文件
- 一个表可以带若干索引

□存储文件

- 逻辑结构组成了关系数据库的内模式
- 物理结构对用户是隐蔽的

□视图

- 从一个或几个基本表导出的表
- 数据库中只存放视图的定义而不存放视图对应的数据
- 视图是一个虚表
- 用户可以在视图上再定义视图

数据管理基础

3.2 学生-课程数据库

智能软件与工程学院

❑ 学生-课程数据库:

学生关系Student、课程关系Course、选修关系SC

学生关系 Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

基础关系 2

课程关系 **Course**

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL语言	6	4

基础关系 3

选修关系 SC

学号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

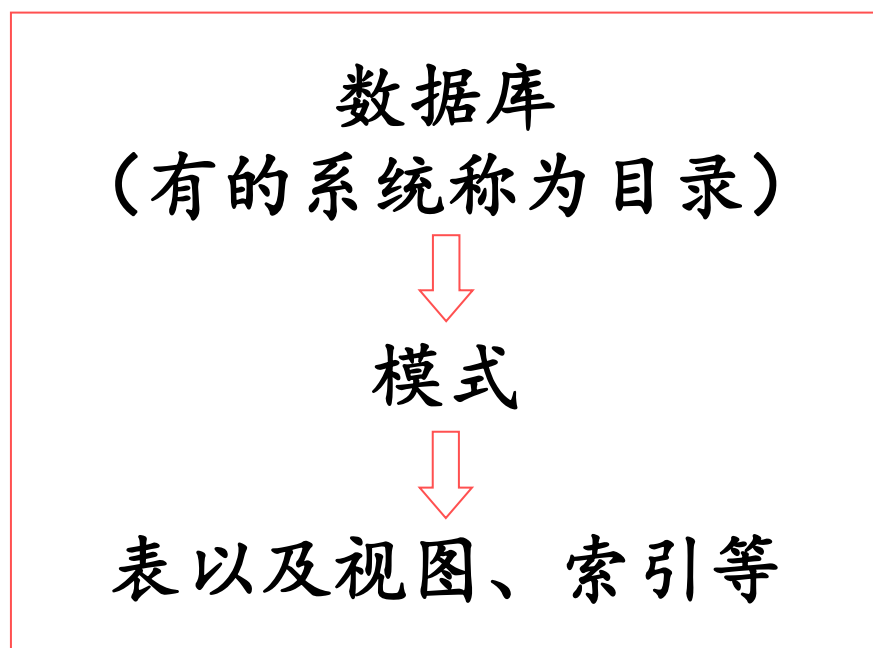
数据管理基础

ch3.3 SQL数据定义

智能软件与工程学院

层次化的数据库对象命名机制

- ❑ 一个关系数据库管理系统的实例（Instance）中可以建立多个数据库
- ❑ 一个数据库中 can 建立多个模式
- ❑ 一个模式下通常包括多个表、视图和索引等数据库对象



数据定义

□SQL的数据定义功能:

- 模式定义
- 表定义
- 视图和索引的定义

操 作 对 象	操 作 方 式		
	创 建	删 除	修 改
模式	CREATE SCHEMA	DROP SCHEMA	
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	CREATE INDEX	DROP INDEX	ALTER INDEX

定义模式 1

❑ 定义模式实际上定义了一个命名空间。

➤ 在这个空间中可以定义该模式包含的数据库对象，例如基本表、视图、索引等。

❑ 在CREATE SCHEMA中可以接受CREATE TABLE, CREATE VIEW和GRANT子句。

➤ CREATE SCHEMA <模式名> AUTHORIZATION <用户名> [
<表定义子句>| <视图定义子句>|<授权定义子句>]

定义模式 2

[例3.1] 为用户WANG定义一个学生-课程模式S-T

CREATE SCHEMA “S_T” AUTHORIZATION WANG;

[例3.2] CREATE SCHEMA AUTHORIZATION WANG;

该语句没有指定<模式名>，<模式名>隐含为<用户名>

[例3.3]为用户ZHANG创建了一个模式TEST，并且在其中定义一个表TAB1

CREATE SCHEMA TEST AUTHORIZATION ZHANG

**CREATE TABLE TAB1 (COL1 SMALLINT,
COL2 INT,
COL3 CHAR(20),
COL4 NUMERIC(10,3),
COL5 DECIMAL(5,2));**

删除模式

❑ **DROP SCHEMA <模式名> <CASCADE|RESTRICT>**

➤ **CASCADE（级联）**

- 删除模式的同时把该模式中所有的数据库对象全部删除

➤ **RESTRICT（限制）**

- 如果该模式中定义了下属的数据库对象（如表、视图等），则拒绝该删除语句的执行。
- 仅当该模式中没有任何下属的对象时才能执行。

❑ **[例3.4] DROP SCHEMA ZHANG CASCADE;**

➤ 删除模式ZHANG

➤ 同时该模式中定义的表TAB1也被删除

定义基本表

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>]);

- <表名>: 所要定义的基本表的名字
- <列名>: 组成该表的各个属性 (列)
- <列级完整性约束条件>: 涉及相应属性列的完整性约束条件
- <表级完整性约束条件>: 涉及一个或多个属性列的完整性约束条件
- 如果完整性约束条件涉及到该表的多个属性列, 则必须定义在表级上, 否则既可以定义在列级也可以定义在表级。

定义基本表-学生表Student

❑[例3.5] 建立“学生”表Student。学号是主码，姓名取值唯一。

```
CREATE TABLE Student
```

```
(Sno CHAR(9) PRIMARY KEY,
```

主码

/* 列级完整性约束条件,Sno是主码*/

```
Sname CHAR(20) UNIQUE,
```

/* Sname取唯一值*/

```
Ssex CHAR(2),
```

```
Sage SMALLINT,
```

```
Sdept CHAR(20)
```

```
);
```

UNIQUE
约束

定义基本表-课程表Course

❑ [例3.6] 建立一个“课程”表Course

```
CREATE TABLE Course (  
    Cno          CHAR(4) PRIMARY KEY,  
    Cname       CHAR(40) NOT NULL,  
    Cpno        CHAR(4),  
    Ccredit     SMALLINT,  
    FOREIGN KEY (Cpno) REFERENCES Course(Cno)  
);
```

先修课

Cpno是外码
被参照表是**Course**
被参照列是**Cno**

❑[例3.7] 建立一个学生选课表SC

```
CREATE TABLE SC (  
    Sno CHAR(9),  
    Cno CHAR(4),  
    Grade SMALLINT,  
    PRIMARY KEY (Sno,Cno),  
    /* 主码由两个属性构成，必须作为表级完整性进行定义*/  
    FOREIGN KEY (Sno) REFERENCES Student(Sno),  
    /* 表级完整性约束条件，Sno是外码，被参照表是Student */  
    FOREIGN KEY (Cno)REFERENCES Course(Cno)  
    /* 表级完整性约束条件， Cno是外码，被参照表是Course*/  
);
```

数据类型 1

- ❑ SQL 中域的概念用数据类型来实现
- ❑ 定义表的属性时需要指明其数据类型及长度
- ❑ 选用哪种数据类型
 - 取值范围
 - 要做哪些运算

数据类型 2

数据类型	含义
CHAR(<i>n</i>), CHARACTER(<i>n</i>)	长度为 <i>n</i> 的定长字符串
VARCHAR(<i>n</i>), CHARACTERVARYING(<i>n</i>)	最大长度为 <i>n</i> 的变长字符串
CLOB	字符串大对象
BLOB	二进制大对象
INT, INTEGER	长整数（4字节）
SMALLINT	短整数（2字节）
BIGINT	大整数（8字节）
NUMERIC(<i>p</i> , <i>d</i>)	定点数，由 <i>p</i> 位数字（不包括符号、小数点）组成，小数后面有 <i>d</i> 位数字
DECIMAL(<i>p</i> , <i>d</i>), DEC(<i>p</i> , <i>d</i>)	同NUMERIC
REAL	取决于机器精度的单精度浮点数
DOUBLE PRECISION	取决于机器精度的双精度浮点数
FLOAT(<i>n</i>)	可选精度的浮点数，精度至少为 <i>n</i> 位数字
BOOLEAN	逻辑布尔量
DATE	日期，包含年、月、日，格式为YYYY-MM-DD
TIME	时间，包含一日的时、分、秒，格式为HH:MM:SS
TIMESTAMP	时间戳类型
INTERVAL	时间间隔类型

ALTER TABLE <表名>

[ADD [COLUMN] <新列名> <数据类型> [完整性约束]]

[ADD <表级完整性约束>]

[DROP [COLUMN] <列名> [CASCADE | RESTRICT]]

[DROP CONSTRAINT <完整性约束名> [RESTRICT | CASCADE]]

[ALTER COLUMN <列名> <数据类型>] ;

修改基本表 2

- ❑ <表名>是要修改的基本表
- ❑ **ADD**子句用于增加新列、新的列级完整性约束条件和新的表级完整性约束条件
- ❑ **DROP COLUMN**子句用于删除表中的列
 - 如果指定了**CASCADE**短语，则自动删除引用了该列的其他对象
 - 如果指定了**RESTRICT**短语，则如果该列被其他对象引用，关系数据库管理系统将拒绝删除该列
- ❑ **DROP CONSTRAINT**子句用于删除指定的完整性约束条件
- ❑ **ALTER COLUMN**子句用于修改原有的列定义，包括修改列名和数据类型

修改基本表 3

[例3.8] 向Student表增加“入学时间”列，其数据类型为日期型

```
ALTER TABLE Student ADD S_entrance DATE;
```

不管基本表中原来是否已有数据，新增加的列一律为空值

[例3.9] 将年龄的数据类型由字符型（假设原来的数据类型是字符型）改为整数。

```
ALTER TABLE Student ALTER COLUMN Sage INT;
```

[例3.10] 增加课程名称必须取唯一值的约束条件。

```
ALTER TABLE Course ADD UNIQUE(Cname);
```


DROP TABLE <表名> [RESTRICT| CASCADE] ;

➤ **RESTRICT**: 删除表是有限制的。

- 欲删除的基本表不能被其他表的约束所引用
- 如果存在依赖该表的对象，则此表不能被删除

➤ **CASCADE**: 删除该表没有限制。

- 在删除基本表的同时，相关的依赖对象一起删除

删除基本表 2

❑ [例3.11] 删除Student表

DROP TABLE Student CASCADE;

- 基本表定义被删除，数据被删除
- 表上建立的索引、视图、触发器等一般也将被删除

❑ [例3.12] 若表上建有视图，选择RESTRICT时表不能删除；选择CASCADE时可以删除表，视图也自动删除。

CREATE VIEW IS_Student

AS

SELECT Sno, Sname, Sage

FROM Student

WHERE Sdept='IS';

DROP TABLE Student RESTRICT;

--ERROR: cannot drop table Student because other objects depend on it

DROP TABLE Student CASCADE;

--NOTICE: drop cascades to view IS_Student

❑ 建立索引的目的：加快查询速度

- 由数据库管理员或表的拥有者建立
- 由关系数据库管理系统自动完成维护
- 关系数据库管理系统自动使用合适的索引作为存取路径，用户不必也不能显式地选择索引

❑ 关系数据库管理系统中常见索引：

- 顺序文件上的索引
- B+树索引
- 散列（hash）索引
- 位图索引

□ 语句格式

```
CREATE [UNIQUE] [CLUSTER] INDEX <索引名>  
ON <表名>(<列名>[<次序>][,<列名>[<次序>] ]...);
```

- **<表名>**: 要建索引的基本表的名字
- **索引**: 可以建立在表的一列或多列上，各列名之间用逗号分隔
- **<次序>**: 指定索引值的排列次序，升序：ASC，降序：DESC。缺省值：ASC
- **UNIQUE**: 此索引的每一个索引值只对应唯一的数据记录
- **CLUSTER**: 表示要建立的索引是聚簇索引

- ❑ [例3.13] 为学生-课程数据库中的Student, Course, SC三个表建立索引。Student表按学号升序建唯一索引, Course表按课程号升序建唯一索引, SC表按学号升序和课程号降序建唯一索引

```
CREATE UNIQUE INDEX Stusno ON Student(Sno);
```

```
CREATE UNIQUE INDEX Coucno ON Course(Cno);
```

```
CREATE UNIQUE INDEX SCno ON SC(Sno ASC, Cno DESC);
```

修改/删除索引

❑ **ALTER INDEX** <旧索引名> **RENAME TO** <新索引名>

➤ [例3.14] 将SC表的SCno索引名改为SCSno

```
ALTER INDEX SCno RENAME TO SCSno;
```

❑ **DROP INDEX** <索引名>;

➤ 删除索引时，系统会从数据字典中删去有关该索引的描述。

➤ [例3.15] 删除Student表的Stusname索引

```
DROP INDEX Stusname;
```

数据字典

- ❑ 数据字典是关系数据库管理系统内部的一组系统表，它记录了数据库中所有定义信息：
 - 关系模式定义
 - 视图定义
 - 索引定义
 - 完整性约束定义
 - 各类用户对数据库的操作权限
 - 统计信息等

- ❑ 关系数据库管理系统在执行SQL的数据定义语句时，实际上就是在更新数据字典表中的相应信息。