



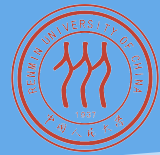
数据库系统概论

An Introduction to Database System

第三章 关系数据库标准语言 SQL

中国人民大学信息学院

第三章 关系数据库标准语言 SQL



3.1 SQL 概述

3.2 学生 - 课程数据库

3.3 数据定义

3.4 数据查询

3.5 数据更新

3.6 视图

3.7 小结

3.3 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

表 3.2 SQL 的数据定义语句

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	

3.3 数据定义



❖ 3.3.1 模式的定义与删除

❖ 3.3.2 基本表的定义、删除与修改

❖ 3.3.3 索引的建立与删除

定义模式（续）



[例 1] 定义一个学生 - 课程模式 S-T

```
CREATE SCHEMA "S-T" AUTHORIZATION
```

```
WANG;
```

为用户 WANG 定义了一个模式 S-T

[例 2] CREATE SCHEMA AUTHORIZATION WANG ;

< 模式名 > 隐含为用户名 WANG

- 如果没有指定 < 模式名 > , 那么 < 模式名 > 隐含为 < 用户名 >

定义模式（续）



- ❖ 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（续）



[例 3]

```
CREATE SCHEMA TEST AUTHORIZATION ZHANG  
CREATE TABLE TAB1(COL1 SMALLINT ,  
                   COL2 INT ,  
                   COL3 CHAR(20) ,  
                   COL4 NUMERIC(10 , 3) ,  
                   COL5 DECIMAL(5 , 2)  
                   );
```

为用户 ZHANG 创建了一个模式 TEST，并在其中定义了一个表 TAB1。

二、删除模式



■ DROP SCHEMA < 模式名 > <CASCADE|RESTRICT>

CASCADE(级联)

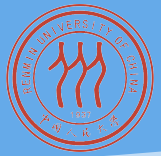
删除模式的同时把该模式中所有的数据库对象全部删除

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。

当该模式中没有任何下属的对象时 才能执行。

删除模式（续）



[例 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

3.3 数据定义



- ❖ 3.3.1 模式的定义与删除
- ❖ 3.3.2 基本表的定义、删除与修改
- ❖ 3.3.3 索引的建立与删除

3.3.2 基本表的定义、删除与修改



一、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 >]) ;

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。

学生表 Student



[例 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 */
```

```
    Sname CHAR(20) UNIQUE , /* Sname 取唯一值 */
```

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

```
    Sage SMALLINT ,
```

```
    Sdept CHAR(20)
```

```
);
```

课程表 Course



[例 6] 建立一个“课程”表 Course

```
CREATE TABLE Course
```

```
( Cno    CHAR(4) PRIMARY KEY ,
```

```
  Cname  CHAR(40) ,
```

```
  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

学生选课表 SC



[例 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 */

```
);
```

二、数据类型



- ❖ 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

二、数据类型



数据类型	长度
CHAR(n)	n 个字节
VARCHAR(n)	n 个字节
INT	INTEGER 4 字节
SMALLINT	2 字节
NUMERIC(p, d)	p 个字节
REAL	4 字节
Double Precision	8 字节
FLOAT(n)	n 个字节
DATE	YYYY-MM-DD
TIME	时间，包含一日的时、分、秒，格式为 HH:MM:SS

三、模式与表



- ❖ 每一个基本表都属于某一个模式

- ❖ 一个模式包含多个基本表

- ❖ 定义基本表所属模式

- 方法一：在表名中明显地给出模式名

- Create table “S-T”.Student (.....) ; /* 模式名为 S-T*/

- Create table “S-T”.Course (.....) ;

- Create table “S-T”.SC (.....) ;

- 方法二：在创建模式语句中同时创建表

- 方法三：设置所属的模式

模式与表（续）



- ❖ 创建基本表（其他数据库对象也一样）时，若没有指定模式，系统根据**搜索路径**来确定该对象所属的模式
- ❖ RDBMS 会使用模式列表中**第一个存在的模式**作为数据库对象的模式名
- ❖ 若搜索路径中的模式名都不存在，系统将给出错误
- ❖ 显示当前的搜索路径： `SHOW search_path;`
- ❖ 搜索路径的当前默认值是： `$user` , `PUBLIC`

模式与表（续）



❖ DBA 用户可以设置搜索路径，然后定义基本表

```
SET search_path TO "S-T", PUBLIC ;
```

```
Create table Student ( ..... ) ;
```

结果建立了 S-T.Student 基本表。

RDBMS 发现搜索路径中第一个模式名 S-T 存在，就把
该

模式作为基本表 Student 所属的模式。

四、修改基本表



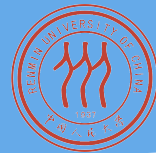
ALTER TABLE < 表名 >

[ADD < 新列名 > < 数据类型 > [完整性约束]]

[DROP < 完整性约束名 >]

[ALTER COLUMN < 列名 > < 数据类型 >] ;

修改基本表（续）



[例 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 ;
```

- 不论基本表中原来是否已有数据，新增加的列一律为空值。

[例 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 10] 增加课程名称必须取唯一值的约束条件。

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

五、删除基本表



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（续）



[例 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

删除基本表（续）



[例 12] 若表上建有视图，选择 RESTRICT 时表不能删除

```
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
```


删除基本表（续）



[例 12] 如果选择 **CASCADE** 时可以删除表，视图也自动被删除

```
DROP TABLE Student CASCADE;
```

```
--NOTICE: drop cascades to view IS_Student
```

```
SELECT * FROM IS_Student;
```

```
--ERROR: relation " IS_Student " does not exist
```

删除基本表（续）



DROP TABLE 时，SQL99 与 3 个 RDBMS 的处理策略比较

	□□□□□□□□□□ □□□□□□□□	SQL99		Kingbase ES		ORACLE 9i		MS SQL SERVER 2000
		R	C	R	C		C	
1.	□□	□□□		√	√	√	√	√
2.	□□	×	√	×	√	√ □□	√ □□	√ □□
3.	DEFAULT □ PRIMARY KEY □ CHECK □□ □□□□□ NOT NULL □□□	√	√	√	√	√	√	√
4.	Foreign Key	×	√	×	√	×	√	×
5.	TRIGGER	×	√	×	√	√	√	√
6.	□□□□□□□	×	√	√ □□	√ □□	√ □□	√ □□	√ □□

R 表示 RESTRICT, C 表示 CASCADE

'×' 表示不能删除基本表，'√' 表示能删除基本表，'保留' 表示删除基本表后，还保留依赖对象

3.3 数据定义



- ❖ 3.3.1 模式的定义与删除
- ❖ 3.3.2 基本表的定义、删除与修改
- ❖ 3.3.3 索引的建立与删除

3.3.3 索引的建立与删除



- ❖ 建立索引的目的：加快查询速度
- ❖ 谁可以建立索引
 - DBA 或 表的属主（即建立表的人）
 - DBMS 一般会自动建立以下列上的索引
 - PRIMARY KEY
 - UNIQUE
- ❖ 谁 维护索引
 - DBMS 自动完成
- ❖ 使用索引
 - DBMS 自动选择是否使用索引以及使用哪些索引

索引



- ❖ RDBMS 中索引一般采用 B+ 树、HASH 索引来实现
 - B+ 树索引具有动态平衡的优点
 - HASH 索引具有查找速度快的特点
- ❖ 采用 B+ 树，还是 HASH 索引则由具体的 RDBMS 来决定
- ❖ 索引是关系数据库的内部实现技术，属于内模式的范畴
- ❖ CREATE INDEX 语句定义索引时，可以定义索引是唯一索引、非唯一索引或聚簇索引

一、建立索引



❖ 语句格式

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

建立索引（续）



[例 13] CREATE CLUSTER INDEX Stusname
ON Student(Sname) ;

- 在 Student 表的 Sname（姓名）列上建立一个聚簇索引
- ❖ 在最经常查询的列上建立聚簇索引以提高查询效率
- ❖ 一个基本表上最多只能建立一个聚簇索引
- ❖ 经常更新的列不宜建立聚簇索引

建立索引（续）



[例 14] 为学生 - 课程数据库中的 Student , Course , SC 三个表建立索引。

```
CREATE UNIQUE INDEX Stusno ON Student(Sno) ;  
CREATE UNIQUE INDEX Coucno ON Course(Cno) ;  
CREATE UNIQUE INDEX SCno ON SC(Sno ASC , Cno  
DESC) ;
```

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

二、删除索引



❖ **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 15] 删除 Student 表的 Stusname 索引

```
DROP INDEX Stusname ;
```