

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

SQL概述

❖ SQL (Structured Query Language)

结构化查询语言，是关系数据库的标准语言



SQL历史

- ❖ 1970年Codd提出了关系模型之后，由于关系代数或者关系都太数学了，难以被普通用户接受，于是1973年IBM开展了System R的研制工作
- ❖ System R 以关系模型为基础，但是摒弃了数学语言，以自然语言为方向，结果诞生了结构化英语查询语言（Structured English Query Language, SEQUEL），负责人为Don Chamberlin博士。
- ❖ 后来更名为SQL，发音不变。
- ❖ System R获得1988年度ACM “软件系统奖”



SQL标准的进展过程

标准	大致页数	发布日期
SQL/86		1986.10
SQL/89 (FIPS 127-1)	120页	1989年
SQL/92	622页	1992年
SQL99 (SQL 3)	1700页	1999年
SQL2003	3600页	2003年
SQL2008	3777页	2006年
SQL2011		2010年

目前，没有一个数据库系统能够支持SQL标准的所有概念和特性

SQL的特点

❖ SQL示例

```
SELECT Sno, Grade  
FROM SC  
WHERE Cno= '3'  
ORDER BY Grade DESC;
```

从表 SC中，
选取 学号Sno和成绩Grade
满足条件 课程号Cno= “3”
按照 成绩Grade降序排列



SQL的特点

❖ 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



SQL的特点

❖ 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。



SQL的特点

❖ 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

❖ 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

SQL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



SQL的特点

❖ 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。

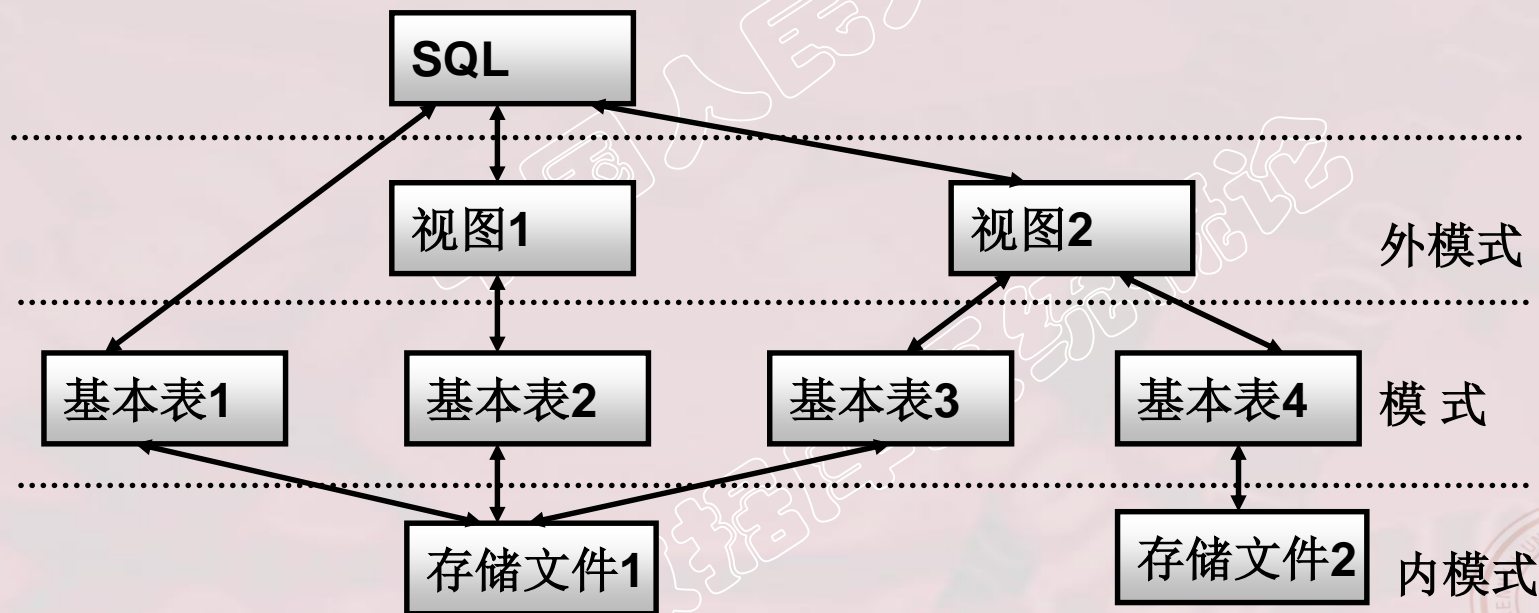
表 3.2 SQL 的动词

SQL 功 能	动词
数 据 查 询	SELECT
数 据 定 义	CREATE, DROP, ALTER
数 据 操 纵	INSERT, UPDATE, DELETE
数 据 控 制	GRANT, REVOKE



SQL的基本概念（续）

SQL支持关系数据库三级模式结构



SQL的基本概念（续）

❖ 基本表

- 本身独立存在的表
- **SQL**中一个关系就对应一个基本表
- 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



SQL的基本概念（续）

❖ 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



SQL的基本概念（续）

❖ 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



学生-课程 数据库

❖ 示例数据库

❖ 学生-课程模式 S-T :

学生表: **Student(Sno,Sname,Ssex,Sage,Sdept)**

课程表: **Course(Cno,Cname,Cpno,Ccredit)**

学生选课表: **SC(Sno,Cno,Grade)**



Student表

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



Course表

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

SC表

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



第三章 关系数据库标准语言SQL

- 1 SQL概述（1个视频）
- 2 数据定义（1个视频）
- 3 数据查询（6个视频）
- 4 数据更新（1个视频）
- 5 空值的处理（1个视频）
- 6 视图（2个视频）





数据定义



数据定义

❖ **SQL的数据定义功能: 定义各种数据库的“对象”**

- 模式定义
- 表定义
- 视图定义
- 索引定义

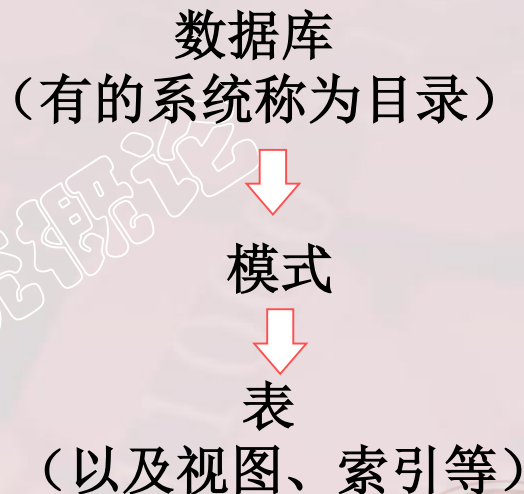
表 3.3 SQL 的数据定义语句

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	ALTER INDEX

各种数据库“对象”

❖ 现代关系数据库管理系统 提供了一个层次化的数据库 对象命名机制

- 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



```
CREATE TABLE Course
(Cno    CHAR(4) PRIMARY KEY,
Cname  CHAR(40),
Cpno   CHAR(4),
Ccredit SMALLINT,
FOREIGN KEY (Cpno)
REFERENCES Course(Cno)
);
```

SQL语言
翻译与处理
程序

数据字典
(以表的方式存储)



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对象的定义信息以及一些统计信息：
 - 关系模式、表、视图、索引的定义
 - 完整性约束的定义
 - 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。



数据定义

1 模式定义

2 表定义

3 索引定义



1. 定义模式

[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>



定义模式（续）

- ❖ 定义模式实际上定义了一个命名空间（或者说目录）。
- ❖ 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3.3]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。



删除模式（续）

[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



2 基本表的定义

❖ 定义基本表

```
CREATE TABLE <表名>                                /* 基本表的名称 */  
(<列名> <数据类型>[ <列级完整性约束条件> ] /*组成该表的列*/  
[,<列名> <数据类型>[<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。

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



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),
```

```
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

- ❖ 关系模型中“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（续）

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

模式与表

❖ 每一个基本表需属于某个模式

❖ 定义基本表所属模式的方式:

■ 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：在创建模式的同时创建表

■ 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
- ❖ 关系数据库管理系统会以“搜索路径”指向的模式作为数据库对象的模式名
- ❖ 设置搜索路径的方式：

SET search_path TO "S-T",PUBLIC;



修改基本表

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（续）

- <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（续）

[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



修改基本表（续）

[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);
```



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



删除基本表（续）

DROP TABLE时，SQL2011 与 3个RDBMS的处理策略比较

序号	标准及主流数据库 依赖基本表 的处理方式 的对象	SQL2011		Kingbase ES		Oracle 12c		MS SQL Server 2012
		R	C	R	C		C	
1	索引	无规定		√	√	√	√	√
2	视图	×	√	×	√	√ 保留	√ 保留	√ 保留
3	DEFAULT, PRIMARY KEY, CHECK (只含该表的列) NOT NULL 等约束	√	√	√	√	√	√	√
4	外码FOREIGN KEY	×	√	×	√	×	√	×
5	触发器TRIGGER	×	√	×	√	√	√	√
6	函数或存储过程	×	√	√ 保留	√ 保留	√ 保留	√ 保留	√ 保留

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 索引的定义

- ❖ 建立索引的目的：加快查询速度
- ❖ 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（参见爱课程网3.2节动画《B+树的增删改》）
 - 散列（hash）索引
 - 位图索引
- ❖ 特点：
 - 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



索引

❖ 谁可以建立索引

- 数据库管理员 或 表的属主（即建立表的人）

❖ 谁维护索引

- 关系数据库管理系统自动完成

❖ 使用索引

- 关系数据库管理系统自动选择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



(1) . 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>]]...);

- **<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



建立索引（续）

[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);
```



(2) 修改索引

❖ **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;



(3) 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3.15] 删除Student表的Stusname索引

DROP INDEX Stusname;



