数据库系统概论

Introduction to Database Systems

第3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

❖SQL（Structured Query Language）

结构化查询语言，是关系数据库的标准语言

- 包括数据查询、数据库模式创建、数据库数据的增删改、数据库安全性和完整性定义与控制等

SQL概述（续）

3.1.1 SQL 的产生与发展

3.1.2 SQL的特点

3.1.3 SQL的基本概念

3.1.1 SQL的产生与发展

标准	篇幅（约）/页	发布日期/年	标准	大致页数	发布日期/年
SQL 86		1986年	SQL 2003	3 600	2003
SQL 89(FIPS 127-1)	120页	1989年	SQL 2008	3 777	2008
SQL 92	622页	1992年	SQL 2011	3817	2011
SQL 99（SQL 3）	1700页	1999年	SQL2016	4035	2016

表3.1 SQL标准的发展过程

SQL的产生与发展（续）

- SQL 86和SQL 89是单个文档。
- SQL 92和SQL 99扩展为一系列开放的部分。例如，SQL 92增加了SQL调用接口、SQL永久存储模块；
- SQL 99扩展为框架、SQL基础部分、SQL调用接口、SQL永久存储模块、SQL宿主语言绑定、SQL外部数据的管理和SQL对象语言绑定等
- SQL2016扩展到了12个部分，引入XML类型、Window函数、TRUNCATE操作、时序数据以及JSON（JavaScript Object Notation）类型等

目前，没有一个关系数据库管理系统能够支持SQL标准的所有概念和特性

3.1 SQL概述

3.1.1 SQL 的产生与发展

3.1.2 SQL的特点

3.1.3 SQL的基本概念

3.1.2 SQL的特点

1.功能综合且风格统一

- 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：
  - 创建和删除数据库模式
  - 创建基本表，创建视图
  - 使用数据库，包括查询和增删改数据、事务处理等
  - 数据库控制，包括安全性控制、完整性控制和并发控制等
  - 数据库维护和重构，如修改和删除基本表、数据库备份与恢复等
- 用户在数据库投入运行后，可根据需要随时或逐步创建模式
- 数据操作符统一

SQL的特点（续）

2. 数据操纵高度非过程化

- 层次、网状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（续）

3. 面向集合的操作方式

- 层次、网状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、更新操作的对象也可以是元组的集合

SQL的特点（续）

4. 以统一的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、Python）程序中，

  供程序员设计程序时使用

SQL的特点（续）

5.语言简洁且易学易用

- SQL功能极强，完成核心功能只用9个动词

SQL 功能	动词
数据定义	CREATE，DROP，ALTER
数据查询	SELECT
数据操纵	INSERT，UPDATE，DELETE
数据控制	GRANT，REVOKE

3.1 SQL概述

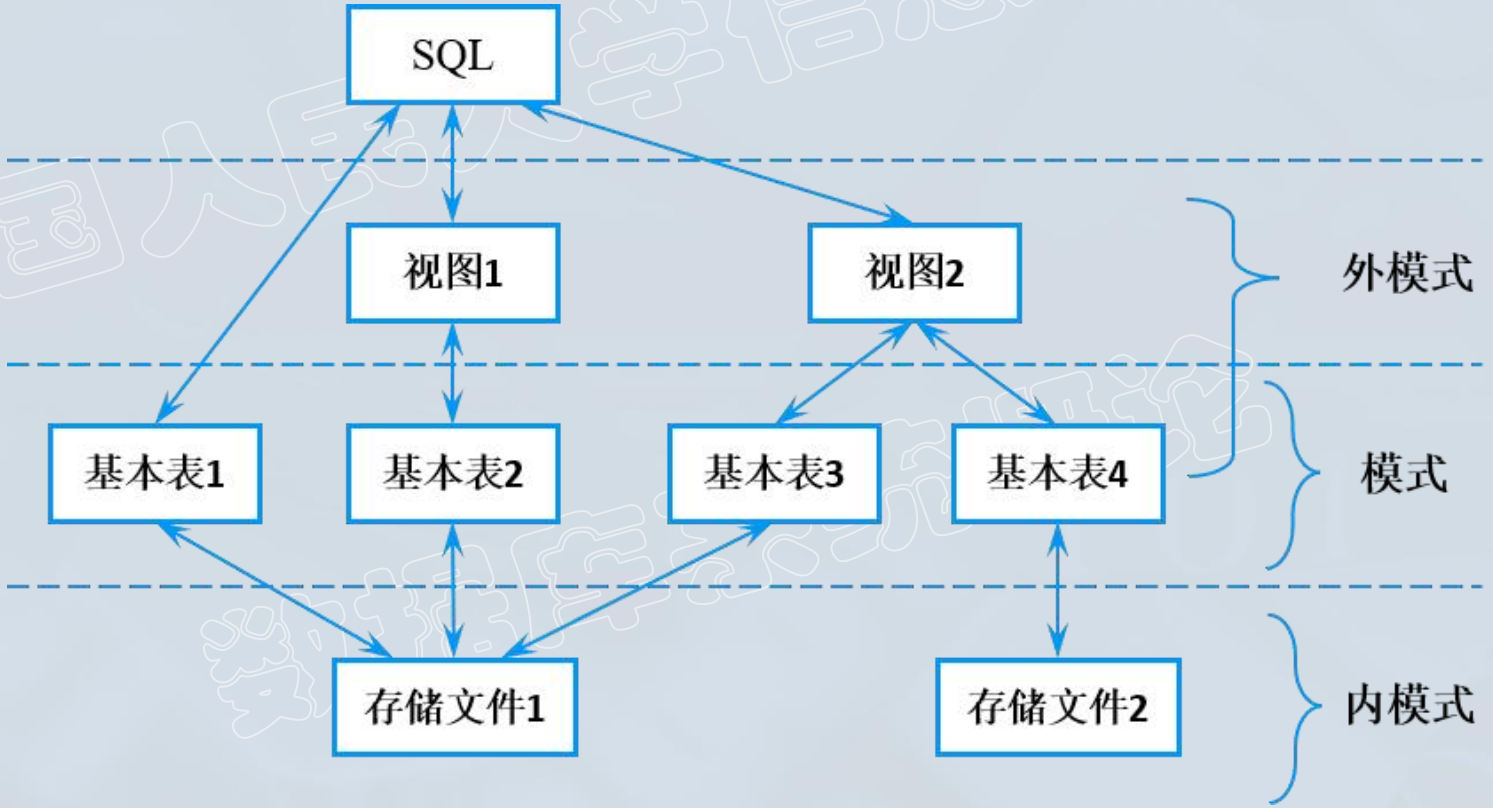
3.1.1 SQL 的产生与发展

3.1.2 SQL的特点

3.1.3 SQL的基本概念

3.1.3 SQL的基本概念

SQL对关系数据库三级模式的支持



SQL的基本概念（续）

- 基本表
  - 本身独立存在的表
  - 关系数据库管理系统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
  - 逻辑结构和物理结构组成了关系数据库的内模式
  - 物理文件结构是由数据库管理系统设计确定的

SQL的基本概念（续）

- 视图
  - 从基本表或其他视图中导出的表
  - 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

第三章 关系数据库标准语言SQL

3.2 数据定义

❖SQL的数据定义功能:

- 数据库模式定义
- 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
数据库模式	CREATE SCHEMA	DROP SCHEMA	*SQL标准无修改语句
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	*CREATE INDEX	*DROP INDEX	*ALTER INDEX

模式

- 关系数据库管理系统提供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
  - 一个数据库中可以建立多个模式
  - 一个模式下通常包括多个表、视图和索引等数据库对象

3.2 数据定义

3.2.1 模式的定义与删除

3.2.2 基本表的定义、删除与修改

3.2.3 索引的建立与删除

3.2.4 数据字典

1. 定义模式

[例3.1] 为用户WANG定义一个“学生选课”模式S-C-SC

CREATE SCHEMA “S-C-SC” 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WANG

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3.3］ 为用户ZHANG创建一个模式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)
                 );
```

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 Test CASCADE;

删除模式**Test**

该模式中定义的表**Tab1**也被删除

3.2 数据定义

3.2.1 模式的定义与删除

3.2.2 基本表的定义、删除与修改

3.2.3 索引的建立与删除

3.2.4 数据字典

定义基本表

❖1.定义基本表

CREATE TABLE <表名>

(<列名> <数据类型>[<列级完整性约束]

[,<列名> <数据类型>[<列级完整性约束>]]

…

[,<表级完整性约束>]);

- <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>：涉及相应属性列的完整性约束
- <表级完整性约束>：涉及一个或多个属性列的完整性约束
- 如果完整性约束涉及该表的多个属性列，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（续）

❖以学生选课数据库为例

定义一个“学生选课”模式S-C-SC，包括以下三个表：

- “学生”表：Student(Sno, Sname, Ssex, Sbirthdate, Smajor)
- “课程”表：Course(Cno,Cname,Ccredit,Cpno)
- “学生选课”表：SC(Sno,Cno,Grade,Semester,Teachingclass)

主码

二维码3.1

学生选课模式

3个表的示例数据

学生表Student

[例3.5] 建立“学生”表Student

```
CREATE TABLE Student
    (Sno   CHAR(8) PRIMARY KEY,
                          /* 列级完整性约束条件,Sno是主码*/
     Sname VARCHAR(20) UNIQUE,        /* Sname取唯一值*/
     Ssex    CHAR(6),
     Sbirthdate Date,
     Smajor  VARCHAR(40)
    );
```

主码

UNIQUE约束

课程表Course

[例3.6] 建立一个“课程”表Course

```
CREATE TABLE  Course
    (Cno      CHAR(5) PRIMARY KEY,
     Cname  VARCHAR(40) NOT NULL,
     Ccredit  SMALLINT,
     Cpno     CHAR(5),
     FOREIGN KEY (Cpno) REFERENCES  Course(Cno)
    );
```

直接先修课

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“学生选课”表SC

```
CREATE TABLE SC
  (Sno   CHAR(8),
   Cno  CHAR(5),
   Grade  SMALLINT,                /*成绩*/
   Semester     CHAR(5),           /*开课学期*/
   Teachingclass   CHAR(8),       /*学生选修某一门课所在的教学班*/
   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，Sno是外码，被参照表是Student */
   FOREIGN KEY (Cno) REFERENCES Course(Cno)
              /*表级完整性约束，Cno是外码，被参照表是Course*/
);
```

2.数据类型

❖2.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

数据类型（续）

数据类型	含义
CHAR(*n*), CHARACTER(*n*)	长度为*n*的定长字符串
VARCHAR(*n*), CHARACTERVARYING(*n*)	最大长度为*n*的变长字符串
CLOB	字符串大对象
BLOB	二进制大对象
INT，INTEGER	整数（4字节），取值范围是[-2147483648，2147483647]
SMALLINT	短整数（2字节），取值范围是[-32768，32767]
BIGINT	大整数（8字节），取值范围是[-2^63，2^63-1]

数据类型

数据类型（续）

数据类型	含义
NUMERIC(*p*, *d*)	定点数，由*p*位数字（不包括符号、小数点）组成，小数点后面有*d*位数字
DECIMAL(*p*, *d*), DEC(*p*, *d*)	同NUMERIC类似，但数值精度不受p和d的限制
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(*n*)	可选精度的浮点数，精度至少为*n*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

3.模式与表

❖3.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-C-SC”.Student(…);  /*Student所属的模式是S-C-SC*/
Create table "S-C-SC".Course(…);   /*Course所属的模式是S-C-SC*/
Create table "S-C-SC".SC(…);       /*SC所属的模式是S-C-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）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）

- 数据库管理员用户可以设置搜索路径，然后定义基本表
  - SET SEARCH_PATH TO "S-C-SC", PUBLIC;
  - 定义基本表：

    Create table Student(.....);
  - 建立S-C-SC.Student基本表
  - 关系数据库管理系统发现搜索路径中第一个模式名S-C-SC，就把该模式作为基本表Student所属的模式

4.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]]

[RENAME COLUMN <列名> TO <新列名>]

[ALTER COLUMN <列名> TYPE <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和新的表级完整性约束
- 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
- RENAME COLUMN子句用于修改列名
- ALTER COLUMN子句用于修改列的数据类型

修改基本表（续）

[例3.8] 向Student表增加“邮箱地址”列Semail，其数据类型为字符型

```
ALTER TABLE Student ADD Semail VARCHAR(30);
```

不论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Student表中出生日期Sbirthdate的数据类型由DATE型改为字符型

```
ALTER TABLE Student ALTER COLUMN Sbirthdate TYPE
VARCHAR(20);
```

/*注意，DATE类型占用19字节，所以修改为VARCHAR时长度要大于等于19*/

[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);
```

5.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，选择CASCADE

```
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3.12]删除Student表，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 CS_Student        /* Student表上建立计算机科学与技术专业学生视图*/
AS
SELECT Sno,Sname,Ssex,Sbirthdate,Smajor
FROM Student
WHERE Smajor='计算机科学与技术';

DROP TABLE Student RESTRICT;              /*删除Student表*/
--ERROR: cannot drop table Student because other objects depend on it
/* 系统返回错误信息，存在依赖该表的对象，此表不能被删除*/
```

删除基本表（续）

[例3.12续] 选择CASCADE时可以删除表，视图也自动被删除

```
DROP TABLE Student CASCADE;                    /*删除Student表*/

--NOTICE: drop cascades to view CS_Student

                                   /*系统返回提示，此表上的视图也被删除*/


SELECT * FROM CS_Student;                      /* CS_Student视图不存在*/

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

二维码3.2

不同产品DROP TABLE

处理策略比较

3.2 数据定义

3.2.1 模式的定义与删除

3.2.2 基本表的定义、删除与修改

3.2.3 索引的建立与删除

3.2.4 数据字典

3.2.3 索引的建立与删除

- 建立索引目的：加快查询速度
- 数据库常见索引：
  - 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》）
  - 哈希（hash）索引
  - 位图索引
- 特点：
  - B+树索引具有动态平衡的优点
  - 哈希索引具有查找速度快的特点

索 引

- 谁可以建立与删除索引
  - 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，用户不必自主地选择索引

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：可以建立在该表的一列或多列上，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。默认值：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“学生选课”数据库中的Student、Course和SC三个表建立索引。其中Student表按学生姓名升序建唯一索引，Course表按课程名升序建唯一索引，SC表按学号升序和课程号降序建唯一索引（即先按照学号升序，对同一个学号再按课程号降序）

```
CREATE UNIQUE INDEX Idx_StuSname ON Student(Sname);
                                        /*保证了Sname取唯一值的约束*/
CREATE UNIQUE INDEX Idx_CouCname ON Course(Cname);
                                        /*加上Cname取唯一值的约束*/
CREATE UNIQUE INDEX Idx_SCCno ON SC(Sno ASC,Cno DESC);
```

2.修改索引

❖2.修改索引

❖ ALTER INDEX <旧索引名> RENAME TO <新索引名>

[例3.14] 将SC表的Idx_SCCno索引名改为Idx_SCSnoCno

```
ALTER INDEX Idx_SCCno RENAME TO Idx_SCSnoCno;
```

3.删除索引

❖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Idx_StuSname索引

```
DROP INDEX Idx_StuSname;
```

3.2 数据定义

3.2.1 模式的定义与删除

3.2.2 基本表的定义、删除与修改

3.2.3 索引的建立与删除

3.2.4 数据字典

3.2.4 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，就是在更新数据字典表中的相应信息
- 查询优化和查询处理时，关系数据库管理系统要根据数据字典中的信息执行处理算法和优化算法