



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

刘 淇

Email: qiliuql@ustc.edu.cn

课程主页:

<http://staff.ustc.edu.cn/~qiliuql/DB2020HF.html>



第三章 关系数据库标准语言SQL

2

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



3.1 SQL概述

3

□ 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



SQL概述（续）

4

- 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



SQL标准的进展过程

5

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	1700页	1999年
■ SQL2003	3600页	2003年
■ SQL2008	3777页	2006年
■ SQL2011		2010年



3.1 SQL概述

6

- 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

7

- 1.综合统一
- 2.高度非过程化
- 3.面向集合的操作方式
- 4.以同一种语法结构提供多种使用方式
- 5.语言简洁，易学易用



1.综合统一

8

- 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：
 - 定义关系模式，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、完整性控制等
- 用户数据库投入运行后，可根据需要随时逐步修改模式，不影响数据的运行。
- 数据操作符统一



2.高度非过程化

9

- 非关系数据模型的数据操纵语言“**面向过程**”，必须制定存取路径
- **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。



3.面向集合的操作方式

10

- 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



4. 以同一种语法结构提供多种使用方式

11

- **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



5.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 3.1 SQL 语言的动词

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



3.1 SQL概述

13

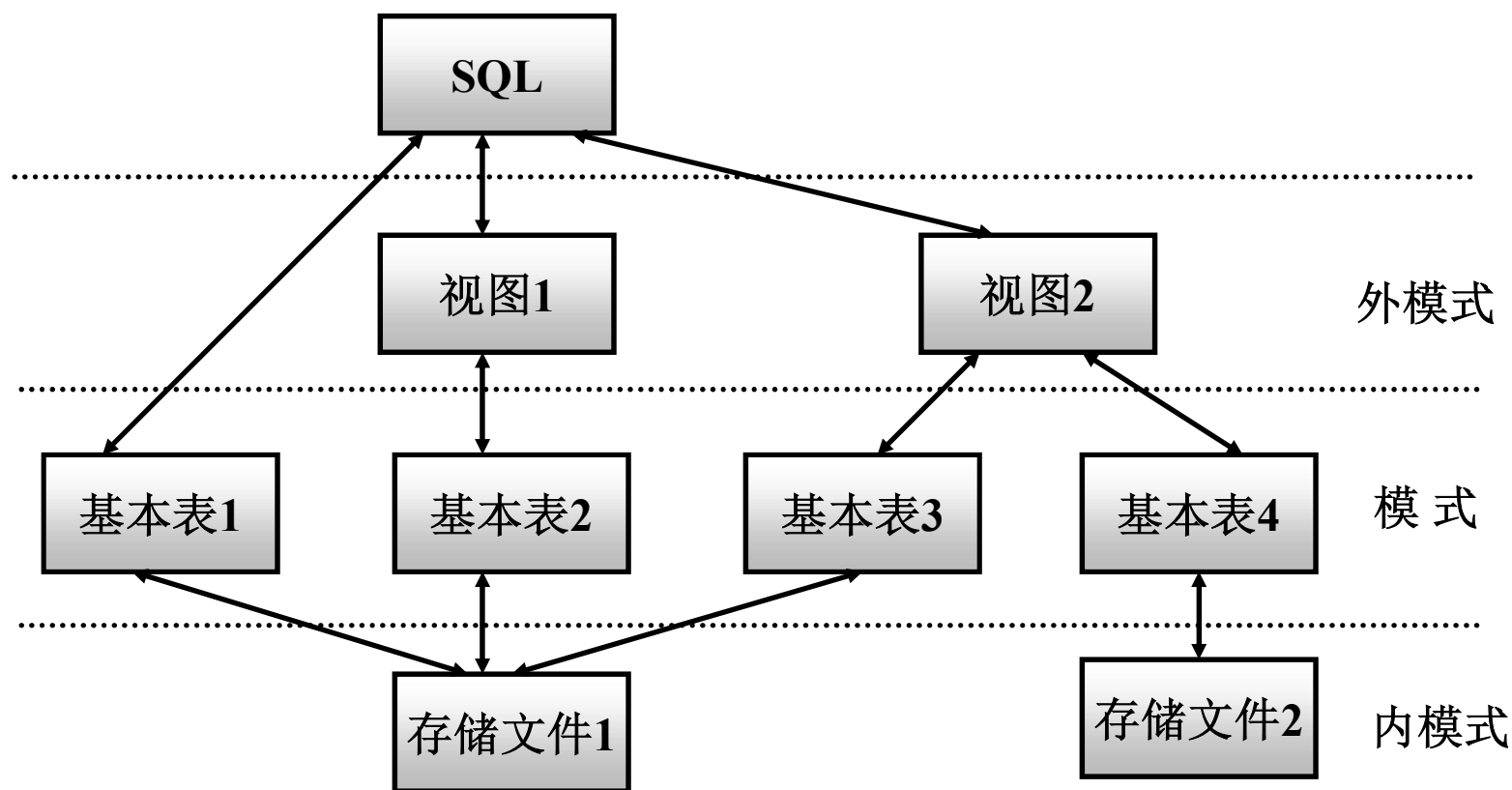
- 3.1.1 SQL 的产生与发展
- 3.1.2 SQL的特点
- 3.1.3 SQL的基本概念



SQL的基本概念（续）

14

SQL语言支持关系数据库三级模式结构





SQL的基本概念（续）

15

□ 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

□ 存储文件

- 逻辑结构组成了关系数据库的内模式
- 物理结构是任意的，对用户透明

□ 视图

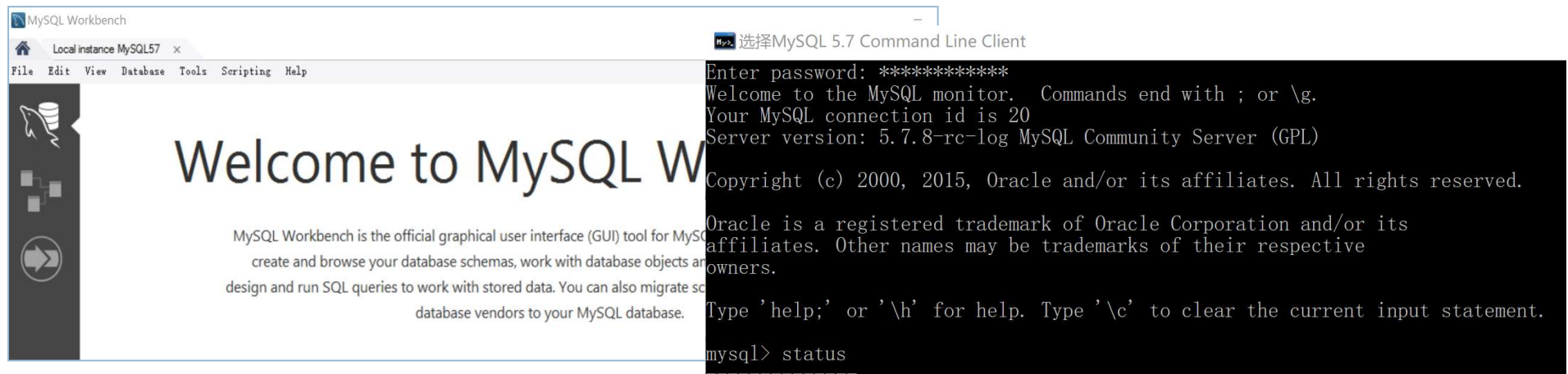
- 从一个或几个基本表导出的表
- 数据库中只存放视图的定义而不存放视图对应的数据
- 视图是一个虚表
- 用户可以在视图上再定义视图



Mysql Workbench

16

- ❑ 专为MySQL设计的ER/数据库建模工具
- ❑ 是可视化数据库设计、管理的工具，它同时有开源和商业化的两个版本
- ❑ 该软件支持Windows和Linux系统
- ❑ 安装之前请确保mysql server已经安装在本机
- ❑ <http://staff.ustc.edu.cn/~qiliuql/DB2019HF.html>





第三章 关系数据库标准语言SQL

17

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



3.2 学生-课程 数据库

18

□ 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
200215121	李勇	男	20	CS
200215122	刘晨	女	19	CS
200215123	王敏	女	18	MA
200515125	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
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80



第三章 关系数据库标准语言SQL

22

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

23

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	ALTER INDEX



3.3 数据定义

24

- 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）

25

- 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。
- 语句格式：

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）

26

[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

- 用户WANG必须是已经存在的数据库用户

[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>

注：不同的数据库系统/软件对SQL标准（命令集）的支持情况不同，也可能会有些修改或扩充，例如，在mysql中给用户授予某个schema/database的权限不是用authorization，而是用grant



定义模式（续）

27

- 创建一个名为TEST的模式，哪一个是正确的？
 - **CREATE SCHEMA "TEST"** --- **"TEST"**
 - **CREATE SCHEMA `TEST`** --- **TEST**
 - **CREATE SCHEMA TEST** --- **TEST**
 - **CREATE SCHEMA TEST;** --- **TEST**
- 创建一个名为S-T的模式呢？
 - **CREATE SCHEMA S-T**

如果你在程序里面写sql,就不要加分号，在程序里面编译器会把分号当做sql本身的一部分，所以会报错

分号是个分隔符，看到分号就标志着本条sql语句结束



定义模式（续）

28

[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**。



MySQL中的实际操作

29

□ CREATE SCHEMA `TEST`;

Use TEST;

#Otherwise Default Schema

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



二、删除模式

30

■ 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）

31

[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



MySQL中的实际操作

32

- ❑ 错误: `drop SCHEMA `TEST` cascade;`
- ❑ 正确: `drop SCHEMA `TEST` ;`
- ❑ 原因:
 - ❑ 在mysql中drop schema即可, 不支持cascade
drop schema会把这个schema下的table也一并删除



3.3 数据定义

33

- 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

34

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

35

[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, /* 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)
);

主码



课程表Course

36

[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

37

[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*/

);



实际操作-Student

38

- Use `S-T`; #we have to use `` for -
- CREATE TABLE Student
 (Sno CHAR(9) **PRIMARY KEY**, /* 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)
);



实际操作-SC

39

```
□ Use `S-T`;  
□ 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

40

- 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	含义
CHAR(n)	长度为n的定长字符串
VARCHAR(n)	最大长度为n的变长字符串
INT	长整数（也可以写作 INTEGER ，4字节）
SMALLINT	短整数（2字节）
NUMERIC(p, d)	定点数，由p位数字（不包括符号、小数点）组成，小数后面有d位数字
REAL	取决于机器精度的浮点数
Double Precision	取决于机器精度的双精度浮点数
FLOAT(n)	浮点数，精度至少为n位数字
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



三、模式与表

42

- 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

■ **Create schema XXX create table XXX**

- 方法三：设置所属的模式

■ **Use schema XXX**



实际操作

43

```
❑ create table `S-T`.CloneStudent  
  (Sno CHAR(9) PRIMARY KEY, # 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)  
  );
```



模式与表（续）

44

- 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**。其含义是首先搜索与用户名相同的模式名，如果该模式名不存在，则使用**PUBLIC**模式。



模式与表（续）

45

- 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。

有些**DBMS**，如**MYSQL**，没有搜索路径的概念