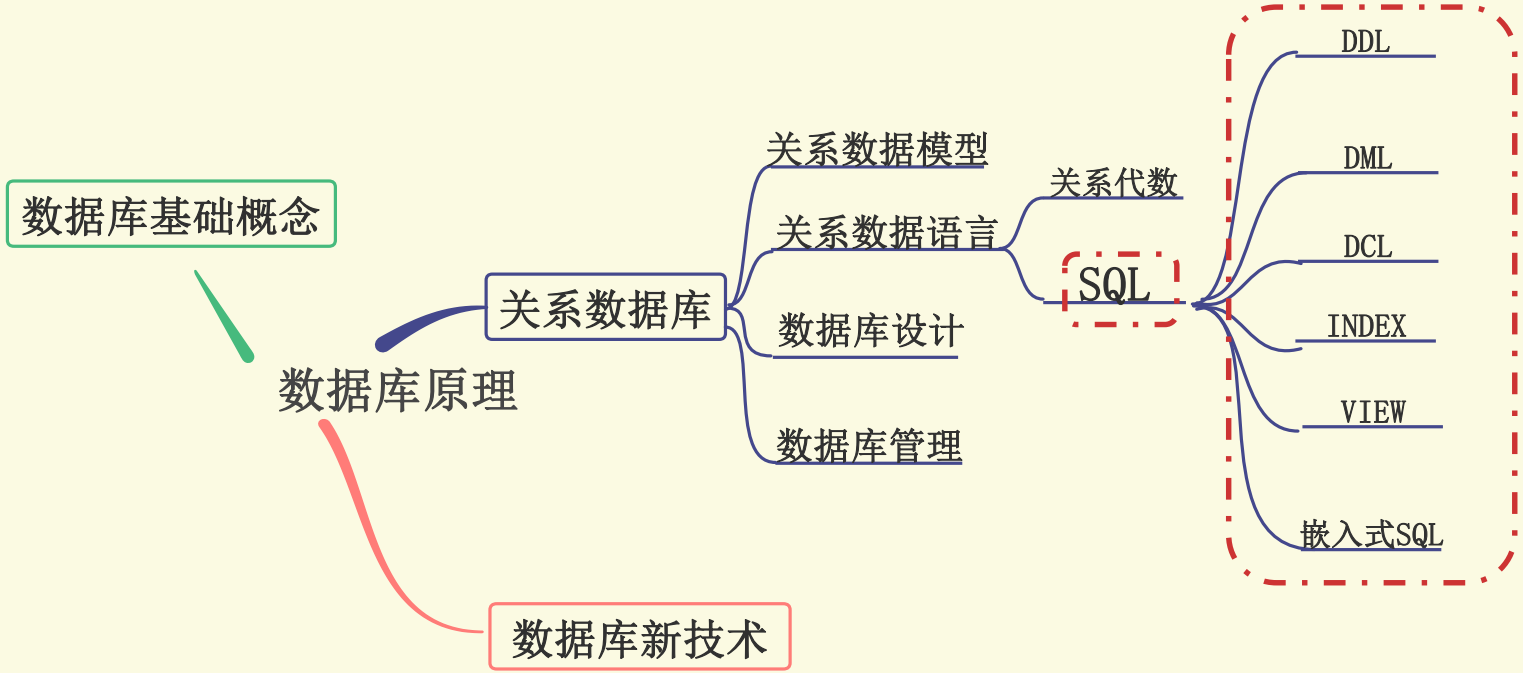
数据库原理

第4章 SQL

辽东学院 鲁 琴

本章要点


数据库基础概念
数据库原理
关系数据库
关系数据模型
关系数据语言
关系代数
SQL
DDL
DML
DCL
INDEX
VIEW
嵌入式SQL
数据库设计
数据库管理
数据库新技术

SQL概述

- SQL语言(Structured Query Language)
- 1974年由Boyce和Chamberlin提出
- 1975年～1979年IBM公司在System R原型系统上实现，SEQUEL语言
- 是关系数据库的标准语言，是数据库领域中一个主流语言

SQL标准

- SQL-86
  - 第一个SQL标准
- SQL-89
- SQL-92 （SQL2）
- 1999年， SQL3
  - 当前应用的
- 大部分商用DBMS的SQL实现类似，但是又与标准SQL不完全相同

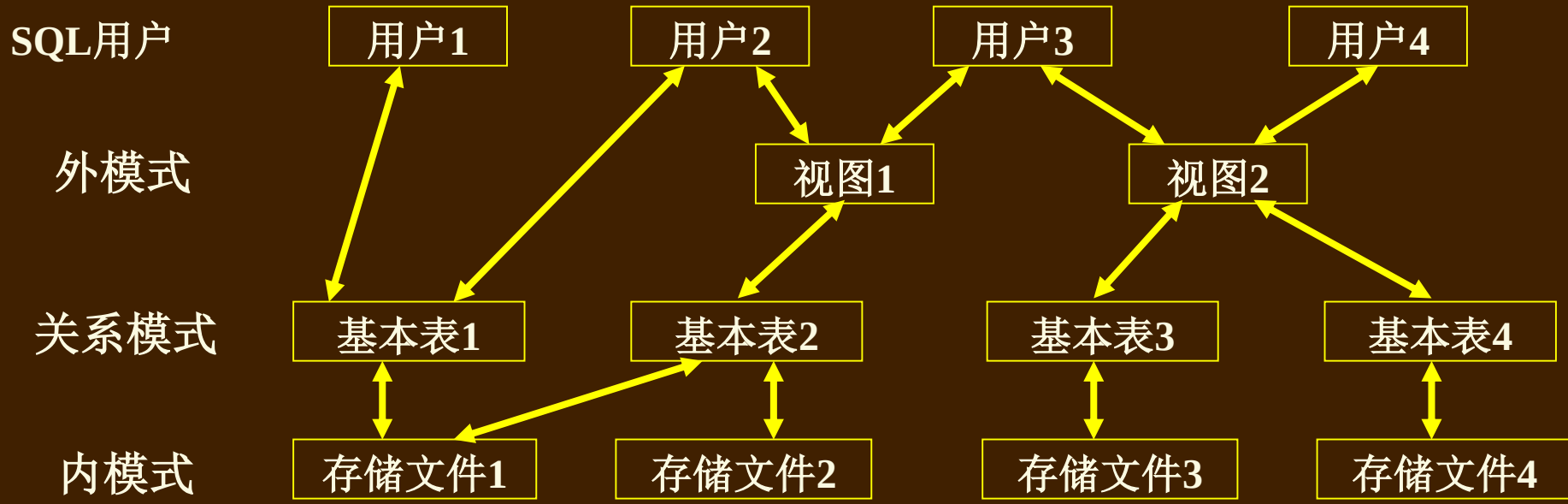
SQL的特点

1. 综合统一
   - DDL,DML,DCL
2. 高度非过程化
   - 用户只需提出“做什么”，而不必指明“怎么做”
   - 存取路径的选择以及SQL语句的操作过程由系统自动完成。
3. 面向集合的操作方式
   - 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. 同一种语法结构提供两种使用方式
   - 自含式语言
   - 嵌入式语言
5. 语言简捷，易学易用
   - 3大类，11个命令词

SQL 3大类11个命令词

SQL功能	动词
数据定义DDL	CREATE DROP ALTER
数据操纵DML	SELECT INSERT UPDATE DELETE
数据控制DCL	GRANT REVOKE

SQL支持数据库的三级模式结构



SQL数据库的体系结构

SQL语言的基本概念

用户用SQL语言对基本表、视图、索引等进行操作

- 基本表（模式）
  - 本身独立存在的表，一个关系对应一个表
  - 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，索引也存放在存储文件中
- 存储文件（内模式）
  - 存储文件的逻辑结构组成了关系数据库的内模式
  - 存储文件的物理结构是任意的，对用户是透明的
- 视图（外模式）
  - 从一个或几个基本表或视图导出的表
  - 是虚表，只存放视图的定义而不存放对应数据

SQL数据定义

SQL数据定义功能

- 定义表(模式)
  - 创建表
  - 删除表
  - 修改表定义
- 定义视图(外模式)
  - 创建视图
  - 删除视图
  - 间接修改视图定义：删除+创建
- 定义索引(内模式)
  - 创建索引
  - 删除索引
  - 间接修改索引定义：删除+创建

SQL数据定义语句

操作对象	操作方式		
	创建	删除	修改
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	CREATE INDEX	DROP INDEX	

定义、删除与修改基本表

1. 定义基本表
2. 修改基本表
3. 删除基本表

1. 定义基本表

- 关系名（表名）
- 属性名（列名）
- 属性数据类型
- 完整性约束

学号	姓名	性别	专业号	出生日期
08150101	张三	女	1	1996－12－12
08150102	李四	男	1	1998－01－20
08150103	王五	男	1	1997－09－09
08150201	赵六	女	2	1998－09－11
08150202	钱七	男	2	1997－10－23
08150301	王国	男	3	1997－08－12

定义基本表语句格式

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

表级完整性约束与列级完整性约束

- 常用完整性约束
  - 主码约束：　**PRIMARY KEY**
  - 参照完整性约束: **FOREIGN KEY…REFERENCES…**
  - 唯一性约束：**UNIQUE**
  - 非空值约束：**NOT NULL**
  - 取值约束：**CHECK**

SQL支持的数据类型

- 第一大类:整数数据
  - **bigint:** 以8个字节来存储正负数, 范围：-2^{63} 到 $2^{63}-1$
  - **int:** 以4个字节来存储正负数，范围：-2^{31} 到 $2^{31}-1$
  - **smallint:** 以2个字节来存储正负数.，范围：-2^{15} 到 $2^{15}-1$
  - **tinyint:** 是最小的整数类型,存储正整数，仅用1字节,范围:0至$2^{8}-1$
  - **bit:** 值只能是0或1，当输入0以外的其他值时，系统均认为是1

    常用来表示真假、男女等二值选择。

SQL支持的数据类型

- 第二大类:精确数值数据
  - **decimal**:用来存储从$-10^{38}+1$到$10^{38}-1$的固定精度和范围的数值型数据
    - 必须指定范围和精度：**decimal (p[,q]) 例：decimal (10,2)**
  - **numeric**:和**decimal**相同

SQL支持的数据类型

- 第三大类:浮点数值数据
  - **float:** 用8个字节来存储数据.最多可为53位.

    范围为:-1.79E+308至1.79E+308.
  - **real:** 位数为24,用4个字节

    数字范围:-3.04E+38至3.04E+38

SQL支持的数据类型

- 第四大类:字符串数据
  - **char: char(n)固定的长度为 n个字符的字符串, 不足的长度会用空格补上.**
  - **varchar: varchar(n)可变的最长长度为n个字符的字符串，尾部的空格会去掉.**

SQL支持的数据类型

- 第五大类:日期时间数据
  - **date:** 日期类型
    - **DATE 'yyyy-mm-dd'**
    - **Example: DATE '2004-09-30'**
  - **time:**时间类型
    - **TIME 'hh:mm:ss'**
    - **Example: TIME '15:30:02.5‘**
  - **datetime:**日期时间类型

例题

[例1] 建立一个“学生”表Student，它由学号Sno、姓名Sname、性别Ssex、年龄Sage、所在系Sdept五个属性组成。其中学号不能为空，值是唯一的，并且姓名取值也唯一。

Sno	Sname	Ssex	Sage	Sdept

Sno	Sname	Ssex	Sage	Sdept
↑	↑	↑	↑	↑
字符型	字符型	字符型	整数	字符型
长度为5	长度为20	长度为1		长度为15
不能为空值				

例题（续）

```
CREATE TABLE Student
        (Sno  CHAR(5) PRIMARY KEY,
         Sname  VARCHAR(20) UNIQUE,
         Ssex  CHAR(1) ,
         Sage  INT,
         Sdept  VARCHAR(15));
```

例题 （续）

[例2] 建立一个“学生选课”表SC，它由学号Sno、课程号Cno，修课成绩Grade组成，其中(Sno, Cno)为主码。

```
CREATE TABLE SC(
        Sno CHAR(5) ,
        Cno  INT ,
        Grade   INT,
        PRIMARY KEY (Sno, Cno));
```

T(TID,TNAME,TITLE)
C(CID,CNAME,TID)
S(SID,SNAME,AGE,SEX)
SC(SID,CID,SCORE)

学数据库中四个基本表

```
CREATE  TABLE  T
(TID        CHAR(4) ,
 TNAME  VARCHAR(8) NOT NULL,
 TITLE  VARCHAR(10),
 PRIMARY  KEY(TID));
```

```
CREATE  TABLE  C
(CID       CHAR(4),
  CNAME  VARCHAR(10) NOT NULL,
  TID       CHAR(4),
  PRIMARY  KEY(CID),
  FOREIGN KEY(TID) REFERENCES T(TID))
```

```
CREATE  TABLE  SC
(SID  CHAR(4) NOT NULL,
 CID  CHAR(4) NOT NULL,
 SCORE  REAL,
 PRIMARY KEY(SID,CID),
 FOREIGN KEY(SID)REFERENCES  S(SID),
 FOREIGN KEY(CID)REFERENCES  C(CID));
```

```
CREATE  TABLE  S
(SID  CHAR(4)  PRIMARY  KEY,
 SNAME  VARCHAR(8)  NOT NULL,
  AGE      SMALLINT,
  SEX      CHAR(1));
```

2. 修改基本表

ALTER TABLE <表名>

[ADD <新列名> <数据类型> | 完整性约束]

[DROP <列名>|<完整性约束名>]

[MODIFY <列名> <数据类型>];

- <表名>：要修改的基本表
- ADD子句：增加新列和新的完整性约束条件
- DROP子句：删除指定列或完整性约束条件
- MODIFY子句：用于修改列名和数据类型

例题

[例] 向Student表增加“入学时间”列，其数据类型为日期型。

```
ALTER TABLE Student ADD Scome 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。
- 如果基本表中原来已有数据，新增列不可有NOT NULL约束

例题

[例] 将年龄的数据类型改为SMALLINT 。

```
ALTER TABLE Student MODIFY Sage SMALLINT;
```

- 注：修改原有的列定义有可能会破坏已有数据。

例题

[例] 删除学生姓名必须取唯一值的约束。

```
ALTER TABLE Student DROP UNIQUE(Sname);
```

SQL Server:

- 增加一列：

ALTER TABLE Student ADD address VARCHAR(30);

- 删除一列

ALTER TABLE Student DROP COLUMN score;

- 删除主键约束

ALTER TABLE Student DROP 主键约束的名字;

- 修改列数据类型

ALTER TABLE SC ALTER COLUMN score INT;

3. 删除基本表

DROP TABLE <表名>;

- 系统从数据字典中删去:
  - 该基本表的描述
  - 该基本表上的所有索引的描述
  - 该基本表表中的数据
- 表上的视图往往仍然保留，但无法引用

例题

[例] 删除Student表。

```
DROP TABLE    Student;
```

小结：数据定义语句

操作对象	操作方式		
	创建	删除	修改
表	CREATE TABLE	DROP TABLE	ALTER TABLE