

数据库系统原理

陈岭

浙江大学计算机学院

3

SQL语言 (1)

- SQL语言概览
- SQL数据定义
- SQL基本模式定义

❑ SQL已在数据库领域中广泛使用

- SQL (Structured Query Language, 结构化查询语言), SQL最早的版本是由IBM开发的, 它最初叫做Sequel (Structured English QUery Language), 在20世纪70年代早期作为System R项目的一部分。

❑ SQL-86: 1986年ANSI和ISO发布了第一个SQL标准

❑ SQL-89: ANSI发布的一个SQL扩充版本

❑ SQL-92: 目前大多数商业RDBMS支持此标准

❑ SQL-99: SQL-92的扩充版本, 现在部分商业数据库系统支持此标准

- ❑ SQL标准符合度，分为四个等级：
 - Entry level SQL（入门级）
 - Transitional SQL（过渡级）
 - Intermediate SQL（中间级）
 - Full SQL（完全级）
- ❑ SQL-92：支持SQL标准符合度的四个等级
- ❑ SQL-99：支持SQL标准符合度的四个等级
 - 现在，没有DBMS产品全部实现了SQL标准符合度的四个等级

□ SQL语言有以下几个部分：

■ DDL (Data-definition Language) 数据定义语言

- create table, alter table, drop table
- create index, drop index
- create view, drop view
- create trigger, drop trigger
- ...

■ DML (Data-manipulation Language) 数据操纵语言

- select ... from
- insert, delete, update

■ DCL (Data-control Language) 数据控制语言

- grant, revoke

SQL 数据定义

❑ 数据库中的关系集合必须由数据定义语言 (DDL) 指定给系统。

❑ 例, CREATE TABLE instructor (
 ID char(5),
 name varchar(20) not null,
 dept_name varchar(20),
 salary numeric(8, 2),
 primary key (ID)) ;

❑ 参见:

- super-key
- candidate key
- primary key

- ❑ SQL的DDL不仅能够定义一组关系，还能够定义每个关系的信息，包括：
 - 每个关系模式
 - 每个属性的取值类型
 - 完整性约束
 - 每个关系维护的索引集合
 - 每个关系的安全性和权限信息
 - 每个关系在磁盘上的物理存储结构

基本类型

- ❑ `char (n)`：固定长度字符串，用户指定长度`n`
- ❑ `varchar (n)`：可变长度的字符串，用户指定最大长度`n`
- ❑ `int`：整数类型（和机器相关的整数类型的子集），等价于全程 `integer`
- ❑ `smallint`：小整数类型（和机器相关的整数类型的子集）
- ❑ `numeric (p, d)`：定点数，精度由用户指定。这个数有`p`位数字，其中`d`位数字在小数点右边
- ❑ `real, double precision`：浮点数与双精度浮点数，精度与机器相关
- ❑ `float (n)`：精度至少为`n`位的浮点数
- ❑ `null`：每种类型都可以包含一个特殊值，即空值。可以申明属性值不为空，禁止加入空值

基本类型

- ❑ date: 日期, 含年、月、日, 如 '2015-3-20'
- ❑ time: 时间, 含小时、分钟、秒, 如 '08:15:30' 或 '08:15:30.75'
- ❑ timestamp: 日期 + 时间, 如 '2015-3-20 08:15:30.75'

注: 日期和时间类型在SQL Server 2000中用“datetime”表示

❑ Sql中有许多函数，用于处理各种类型的数据及其类型转换，但各数据库系统中函数的标准化程度不高

■ 不同的函数：

— SqlServer: substring(s, start, length), getdate(), datalength('abc'),
.....

— Oracle: substr(s, start, length), sysdate, length('abc'),
to_char(sysdate, 'yyyy/mm/dd')得: 2015/03/20, to_date('15/03/20' ,
'yy/mm/dd'),

■ 相同的函数：

— abs() (绝对值)

— exp() (指数)

— round() (四舍五入)

—

基本模式定义

□ 我们用 `create table` 命令定义SQL关系:

`CREATE TABLE r`

`(A1 D1 , A2 D2 , ..., An Dn ,`

`<完整性约束1> ,`

`...,`

`<完整性约束k>);`

其中, r 是关系名, 每个 A_i 是关系 r 模式中的一个属性名, D_i 是属性 A_i 的域

基本模式定义

- ❑ SQL支持许多不同的完整性约束。在本节我们只讨论其中少数几个：
 - not null, 一个属性上的not null约束表明在该属性上不允许空值
 - primary key (A_1, \dots, A_n), 声明表示属性 A_1, \dots, A_n 构成关系的主码。主码属性必须非空且唯一。主码属性在SQL_92标准中无需指明 not null, 但是在SQL_89中需要明确指明
 - foreign key (A_1, \dots, A_m) references, 声明表示关系中任意元组在属性上(A_1, \dots, A_m)上的取值必须对应于关系s中某元组的主码属性上的取值
 - check (P), P是谓词条件

基本模式定义

- 例，声名 *ID* 为 *instructor* 的主码，并确保 *salary* 的值非负

```
CREATE TABLE instructor
  (ID char(5),
   name varchar(20) not null,
   dept_name varchar(20),
   salary numeric(8,2),
   primary key (ID),
   check (salary >= 0));
```

方法2:

```
CREATE TABLE instructor2
  ( ID char(5)
    primary key ,
    name varchar(20),
    dept_name varchar(20),
    salary numeric(8,2),
    check (salary >= 0));
```

- *salary* 可能为空？在 SQL Server 2000 中，与 WHERE 子句不同的是：它能为空

基本模式定义

- 我们用 **drop table** 命令，从数据库中删除关于被去掉关系的所有信息
 - `DROP TABLE r;`
 - 如，`DROP TABLE instructor2;`
- 我们用 **alter table** 命令，为已有关系增加属性
 - `ALTER TABLE r ADD A D;`
 - 其中，*A* 是待添加属性的名字，*D* 是待添加属性的域。关系中所有元组在新属性上的取值将被设为 *null*
 - 如，`ALTER TABLE instructor add birthday date;`
- 我们也可以用 **alter table** 命令，从关系中去掉属性
 - `ALTER TABLE r DROP A`
 - 但是很多数据库系统并不支持此操作
- 我们还可以用 **alter table** 命令，修改关系中的属性
 - 如，`ALTER TABLE instructor MODIFY (ID char(10));`

- ❑ SQL是具有影响力的商用市场化的关系查询语言。SQL语言包括几个部分：
 - 数据定义语言（DDL）
 - 数据操纵语言（DML）
 - 数据控制语言（DCL）
- ❑ SQL的数据定义语言用于创建具有特定模式的关系。除了声明关系属性的名称和类型之外，SQL还允许声明完整性约束，例如主码约束和外码约束

谢谢！