

# 数据库

## 关系代数：

### 一. 关系代数五个基本操作

- 1.选择 ( $\sigma$ ) :把满足条件的元组找出来(元组对应行)
- 2.投影 ( $\Pi$ ) : 把不需要的列去掉 (属性对应列, 对某些或几列属性进行查询)
- 3.笛卡尔乘积 ( $\times$ ) :把2个关系拼接
- 4.集合差 ( $-$ ) :集合中减法
- 5.集合并 ( $\cup$ ) :2个关系中模式相同的元组并起来

### 二. 常用操作

- 1.集合交( $\cap$ ):和数字中类似

#### 2.连接

- (1) 条件连接:表1 表2= $\sigma$ 条件 (表1 $\times$ 表2)
- (2) 等值连接:若 (1) 中条件都是相等判断, 则是等值连接
- (3) 自然连接:2张表在它们所有的公共属性上做等值连接, 并把结果中重复的

列去掉

- (4) 外连接:

①左外连接:把连接运算符左边的所有元组都保留下来, 不匹配的元组在右边补空值

②右外连接:把连接运算符右边的所有元组都保留下来, 不匹配的元组在右边补空值

③外连接: 把所有元组都保留下来

- 3.除运算:查找所有 (全部) (至少) ...的时候用除法操作(所查信息不能在一个表中查)


首先两表相除需要有公共属性，除法作用是我们找出被除数中是否有一个或几个元组（除了公共属性以外的元组）与除数中所有相关属性都有关联，往往全部...作为除数，而所查的信息（属性）作为被除数。

SQL查询:

基本查询

### 1.格式

select[distinct](去重) 所查内容

from 所查询的表(可多个)

where 条件

### 2.常用条件

(1)between...and..., not between...and...,在...之间或不在...之

(2)in ,not 确定集合

(3)like 模糊查询

①%表示任意长度 ②\_表示任意单个字符

③若你想查的字符为\_,则在\_前加\

(4) and 查询结果满足and两边条件

or 查询结果满足两边一个条件就行

not 否定

(5) is null, is not null

### 3. 聚集函数

(1) count(\*): 统计在一个关系中有多少元组

(2) count([distinct]A): 计算属性A有多少个值(加distinct表示有多少不同的值)

(3) sum([distinct]A): A的属性上所有元组加一起求和

(4) avg([distinct]A): 对A的属性上所有元组求平均值

(5) max([distinct]A): 求属性A的最大值

(6) min([distinct]A): 求属性A的最小值

### 4. 查询完整格式

select [distinct] 所查内容 (注: having可以和聚集函数

from 所查询的表 (可多个) 一起使用, where不可以)

where 条件或连接或嵌套

group by 属性 对属性相同的值进行分组

having 条件 对group by得到的组进行筛选

order by \_\_\_\_ (desc) 对查询结果的一个或多个属性升序或降序(desc)

### 5. 连接查询

(1) 用法: 所查信息不能在一个表中查出来

(2) 自然连接: 在等值连接的基础上把重复属性列去掉

(3) 自身连接: 自己与自己做笛卡尔积, 再连接条件做选择操作

(4) 外连接: 不是重点

(5) 多表连接

### 6. 嵌套查询

(1)分类:

①不相关子查询:子查询的查询条件不依赖于父查询

②相关子查询:子查询的查询条件依赖于父查询

(2)in

(3)比较运算符

(4)any.all

(5)exists (存在), not exists(不存在)

7集合查询

(1)intersect(交)

(2)union(并)

(3)except(差)

8派生表

9插入数据

(1)插入元组:

insert

into 表名 (属性1, 属性2, ...)

values (值1,值2,...)

(2)插入子查询结果:

(3)insert

into 表名(属性1,属性2,....)

子查询

10.修改数据

update 表名

set 属性=修改的值

where

条件

## 11. 删除数据

delete

from 表名 (修改和删除的条件可以是子查询)

where 条件

## 12. 视图

### (1) 创建视图

create view 视图名 (属性1, 属性2,...) //属性可省

as 查询语句

[with check option] //可省, 加上表示对视图进行更新, 插入, 删除操作

时要保证更新, 插入或删除的行满足子查询中的条件表达式

### (2) 删除视图

drop view 视图名 [cascade] //cascade可省, 表示级联删除该视图和它导出的所有视图

### (3) 查询视图, 更新视图和基本表类似

完整性与安全性

## 1. 三大完整性约束

基本概念: (1) 候选码: 一个或一组属性能唯一地标识一个元组, 而其子集不能

(2) 主码: 若有多个候选码, 则选其中一个为主码

(3) 外键(外码): 一个表中一个(多个)属性, 不是自身的主码, 且这个属性和另一个表中的主码属性是同一个属性, 则这个(组)属性称为外码

(1) 实体完整性: 主码不能重复, 主码不能为空

(2) 参照完整性: 外键不能为空(或者均为空值) 外键等于被参照表的主码值

(3) 用户定义完整性: 用户自定义的完整性约束条件

2. 可能大题: 创建一个表, 给出属性以及属性的条件, 定义主键和外键, 而且当你更新或删除与外键有关的另一个表时, 这个表是否要级联, 拒绝更新, 删除

(1) 基本概念: ①数据类型: a.char(n): 长度为n的定长字符串 b.varchar(n): 最大长度为n的变长字符串 c.int: 长整数(4字节) d.smallint: 短整数(2字节)

②定义主键，外键

a.主键: primary key (属性1, 属性2, )

相同

b.外键: foreign key 外键 reference 被参照表 (主键)

③级联更新/删除: on delete/update cascade

④拒绝更新/删除: on delete/update no action

一般③④跟在外键后

(2)结构

create table 表名

(属性 数据类型 定义的约束条件 (包括①非空: not null ②唯一 日唯 unique ③check: 自定义的约束条件 ④定义为主码))

属性 数据类型 定义的约束条件

.....

primary key (属性1, 属性2)//定义主键

foreign key (属性)外键 references 被参照表 (属性) 主键 on delete/update cascade/no action

画√的地为可以加constraint 完整性约束条件名

(4)授权

grant 权限(如delete)

on 对象类型 (如table, view) 对象名

to 用户

[with grant option]//可省, 加上表示可以将权限授予其他用户

(5)对权限收回

revoke 权限(如select)

on 对象类型(如table, view)

from 用户

范式

1.1NF: 关系中的每个属性都是原子的, 不可再分的, 每个关系都要满足1NF

2.2NF: 满足1NF且这张表中不存在属性对主键的部分函数依赖(部分函数依赖指的是依赖于主键的一部分)

如: S(sno, sname, age, cno, grade)

主键(sno, cno)

成绩由sno, cno共同决定, 但sname, age仅由sno就可以决定, 即sname, age只依赖于主键的一部分, 所以不满足2NF

解决方法:一张表只管一件事.

把学生表分成学生基本信息表和选课表

S(sno, sname,age) SC (sno, cno, grade)

3.3NF:满足2NF且不存在属性对主键的传递依赖

如Teacher(T#, sal\_level,salary)

T#(决定)→sal\_level sal\_level→salary

如将2级工资水平评为3000改为2级工资水平工资为3000。虽然学号没变,但工资额已经分为  
level (sal\_level,salary)

Teacher (T#, sal\_level)

解决方案:一张表管一件事


4.BCNF:不存在以下关系


R (C, S, P)




## E-R图

### 1.基本概念

(1)实体: 

(2)属性: 

(3)联系: 

### 2.画E-R图

### 3.转换为关系模型

(1)E-R图的实体转换为关系(表名)

(2)E-R图的属性转换为关系的属性

### (3)联系转换

①联系种类:1:1, 1:n(n:1), n:m

②联系转换方法:

a.1:1两种方法

第一种将联系转换成一个关系(表), 这个关系的属性是两个关系的主码加联系的属性.

第二种是将一个关系的主码加联系的属性写到另一个关系条中

b.1:n转换两种方法

第一种

第二种此时是将联系的1:n中1端关系的主码写到n端上

C.n:m

第一种