

数据库系统概论

An Introduction to Database System

第二章 关系数据库（续）

中国人民大学信息学院

第二章 关系数据库



2.1 关系模型概述

2.2 关系数据结构

2.3 关系的完整性

2.4 关系代数

2.5 关系演算

2.6 小结

2.5 关系演算



❖ 关系演算

以数理逻辑中的谓词演算为基础

❖ 按谓词变元不同 进行分类

1. 元组关系演算：

以元组变量作为谓词变元的基本对象

元组关系演算语言 ALPHA

2. 域关系演算：

以域变量作为谓词变元的基本对象

域关系演算语言 QBE

2.5.1 元组关系演算语言 ALPHA



❖ 由 E.F.Codd 提出

INGRES 所用的 QUEL 语言是参照 ALPHA 语言研制的

❖ 语句

检索语句

- GET

更新语句

- PUT , HOLD , UPDATE , DELETE , DROP

一、检索操作



❖ 语句格式:

GET 工作空间名 [(定额)] (表达式 1)

[: 操作条件] [DOWN/UP 表达式 2]

定额: 规定检索的元组个数

- 格式: 数字

表达式 1: 指定语句的操作对象

- 格式:

关系名 | 关系名 . 属性名 | 元组变量 . 属性名 | 集函数 [, ...]

操作条件: 将操作结果限定在满足条件的元组中

- 格式: 逻辑表达式

表达式 2: 指定排序方式

- 格式: 关系名 . 属性名 | 元组变量 . 属性名 [, ...]

一、检索操作



(1) 简单检索

GET 工作空间名 (表达式 1)

[例 1] 查询所有被选修的课程号码。

GET W (SC.Cno)

[例 2] 查询所有学生的数据。

GET W (Student)

(2) 限定的检索



格式

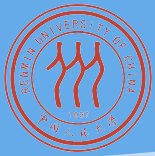
GET 工作空间名 (表达式 1) : 操作条件

[例 3] 查询信息系 (IS) 中年龄小于 20 岁的学生的学号和
年龄

GET W (Student.Sno , Student.Sage):

Student.Sdept='IS' ^ Student.Sage<20

(3) 带排序的检索



格式

GET 工作空间名 (表达式 1) [: 操作条件]
DOWN/UP 表达式 2

[例 4] 查询计算机科学系 (CS) 学生的学号、年龄，结果按年龄降

序排序

GET W (Student.Sno , Student.Sage):
Student.Sdept='CS' DOWN Student.Sage

(4) 带定额的检索



格式

GET 工作空间名 (定额) (表达式 1)
[: 操作条件] [DOWN/UP 表达式 2]

[例 5] 取出一个信息系学生的学号。

GET W (1) (Student.Sno):

Student.Sdept='IS'

[例 6] 查询信息系年龄最大的三个学生的学号及其年龄，结果按年龄降序排序。

GET W (3) (Student.Sno , Student.Sage):

Student.Sdept='IS' DOWN Student.Sage

(5) 用元组变量的检索



❖ 元组变量的含义

- 表示可以在某一关系范围内变化（也称为范围变量 Range Variable）

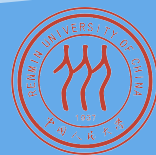
❖ 元组变量的用途

- ① 简化关系名：设一个较短名字的元组变量来代替较长的关系名。
- ② 操作条件中使用量词时必须用元组变量。

❖ 定义元组变量

- 格式： RANGE 关系名 变量名
- 一个关系可以设多个元组变量

(6) 用存在量词的检索



❖ 操作条件中使用量词时必须用元组变量

[例 8] 查询选修 2 号课程的学生名字。

RANGE SC X

GET W (Student.Sname):

$\exists X(X.Sno=Student.Sno \wedge X.Cno='2')$

[例 9] 查询选修了这样课程的学生学号，其直接先行课是 6 号课程。

RANGE Course CX

GET W (SC.Sno):

$\exists CX(CX.Cno=SC.Cno \wedge CX.Pcno='6')$

用存在量词的检索（续）



[例 10] 查询至少选修一门其先行课为 6 号课程的学生名字

RANGE Course CX

SC SCX

GET W (Student.Sname): \exists SCX (SCX.Sno=Student.Sno \wedge
 \exists CX (CX.Cno=SCX.Cno \wedge CX.Pcno='6'))

前束范式形式:

GET W (Student.Sname):

\exists SCX \exists CX (SCX.Sno=Student.Sno \wedge

CX.Cno=SCX.Cno \wedge CX.Pcno='6')

(7) 带有多个关系的表达式的检索



[例 11] 查询成绩为 90 分以上的学生名字与课程名字。

RANGE SC SCX

GET W(Student.Sname , Course.Cname):

$\exists \text{SCX} (\text{SCX.Grade} \geq 90 \wedge$

$\text{SCX.Sno} = \text{Student.Sno} \wedge$

$\text{Course.Cno} = \text{SCX.Cno})$

(8) 用全称量词的检索



[例 12] 查询不选 1 号课程的学生名字

RANGE SC SCX

GET W (Student.Sname):

\forall SCX (SCX.Sno \neq Student.Sno \vee SCX.Cno \neq '1')

用存在量词表示:

RANGE SC SCX

GET W (Student.Sname):

$\neg \exists$ SCX (SCX.Sno=Student.Sno \wedge SCX.Cno='1')

(9) 用两种量词的检索



[例 13] 查询选修了全部课程的学生姓名。

RANGE Course CX

SC SCX

GET W (Student.Sname):

$\forall CX \exists SCX (SCX.Sno=Student.Sno \wedge$
 $SCX.Cno=CX.Cno)$

(10) 用蕴函 (Implication) 的检索



[例 14] 查询最少选修了 200215122 学生所选课程
的学生学号

RANGE Couse CX
SC SCX
SC SCY

GET W (Student.Sno): $\forall CX(\exists SCX$
 $(SCX.Sno='200215122' \wedge SCX.Cno=CX.Cno)$



$\exists SCY(SCY.Sno=Student.Sno \wedge$
 $SCY.Cno= CX.Cno))$

(11) 聚集函数

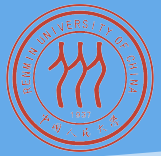


常用聚集函数 (Aggregation function) 或内部函数
(Build-in function)

函数名	功能
COUNT	对元组计数
TOTAL	求总和
MAX	求最大值
MIN	求最小值
AVG	求平均值

关系演算中的聚集函数

聚集函数（续）



[例 15] 查询学生所在系的数目。

```
GET W ( COUNT(Student.Sdept) )
```

COUNT 函数在计数时会自动排除重复值。

[例 16] 查询信息系学生的平均年龄

```
GET W (AVG(Student.Sage):
```

```
Student.Sdept='IS' )
```

二、更新操作



(1) 修改操作

(2) 插入操作

(3) 删除操作

(1) 修改操作步骤



- ① 用 HOLD 语句将要修改的元组从数据库中读到工作空间中

HOLD 工作空间名 (表达式 1) [: 操作条件]

HOLD 语句是带上并发控制的 GET 语句

- ② 用宿主语言修改工作空间中元组的属性

- ③ 用 UPDATE 语句将修改后的元组送回数据库中

UPDATE 工作空间名

修改操作（续）



[例 17] 把 200215121 学生从计算机科学系转到信息系。

HOLD W (Student.Sno , Student.Sdept):

Student.Sno='200215121'

(从 Student 关系中读出 95007 学生的数据)

MOVE 'IS' TO W.Sdept

(用宿主语言进行修改)

UPDATE W

(把修改后的元组送回 Student 关系)

(2) 插入操作



步骤

- ① 用宿主语言在工作空间中建立新元组
- ② 用 PUT 语句把该元组存入指定关系中

PUT 工作空间名 (关系名)

PUT 语句只对一个关系操作，关系演算中的聚集函数

插入操作（续）



[例 18] 学校新开设了一门 2 学分的课程“计算机组织与结构”，其课程号为 8，直接先行课为 6 号课程。插入该课程元组

MOVE '8' TO W.Cno

MOVE '计算机组织与结构' TO W.Cname

MOVE '6' TO W.Cpno

MOVE '2' TO W.Ccredit

PUT W (Course)

(3) 删除操作

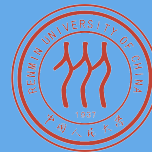


步骤

- ① 用 **HOLD** 语句把要删除的元组从数据库中读到工作空间中
- ② 用 **DELETE** 语句删除该元组

DELETE 工作空间名

删除操作（续）



[例 19] 200215125 学生因故退学，删除该学生元组

```
HOLD W (Student): Student.Sno='200215125'
```

```
DELETE W
```

删除操作（续）



[例 20] 将学号 200215121 改为 200215126

HOLD W (Student): Student.Sno='200215121'

DELETE W

MOVE '200215126' TO W.Sno

MOVE '李勇' TO W.Sname

MOVE '男' TO W.Ssex

MOVE '20' TO W.Sage

MOVE 'CS' TO W.Sdept

PUT W (Student)

删除操作（续）



[例 21] 删除全部学生

```
HOLD W (Student)
```

```
DELETE W
```

为保证参照完整性，删除 Student 中元组时相应地要删除 SC 中的元组

```
HOLD W (SC)
```

```
DELETE W
```

小结：元组关系演算语言 ALPHA



❖ 检索操作 GET

GET 工作空间名 [(定额)] (表达式 1)
[: 操作条件] [DOWN/UP 表达式 2]

❖ 插入操作

- 建立新元组 --PUT

❖ 修改操作

- HOLD-- 修改 --UPDATE

❖ 删除操作

- HOLD--DELETE

2.5 关系演算



❖ 2.5.1 元组关系演算语言 ALPHA

❖ 2.5.2 域关系演算语言 QBE

2.5.2 域关系演算语言 QBE



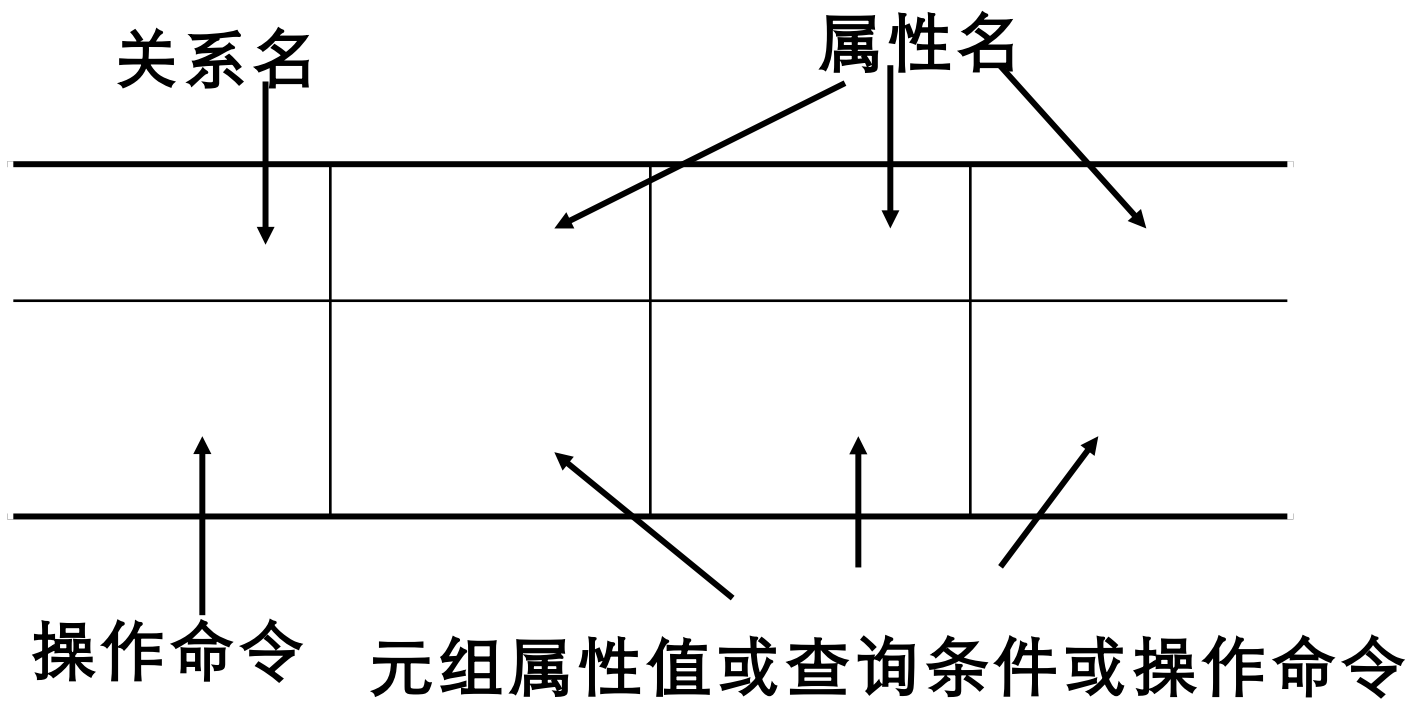
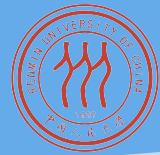
❖ 一种典型的域关系演算语言

- 由 M.M.Zloof 提出
- 以元组变量的分量即域变量作为谓词变元的基本对象

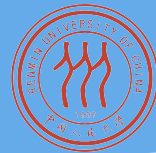
❖ QBE : Query By Example

- 基于屏幕表格的查询语言
- 查询要求：以填写表格的方式构造查询
- 用示例元素（域变量）来表示查询结果可能的情况
- 查询结果：以表格形式显示

QBE 操作框架



一、检索操作



1. 简单查询

[例 1] 求信息系全体学生的姓名

操作步骤为：

(1) 用户提出要求；

(2) 屏幕显示空白表格；

简单查询（续）



（3）用户在最左边一栏输入要查询的关系名 Student ；

Student					

（4）系统显示该关系的属性名

Student	Sno	Sname	Ssex	Sage	Sdept

简单查询（续）



（5）用户在上面构造查询要求

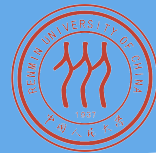
Student	Sno	Sname	Ssex	Sage	Sdept
		P. <u>李勇</u>			IS

■ 李勇是示例元素，即域变量

（6）屏幕显示查询结果

Student	Sno	Sname	Ssex	Sage	Sdept
		李勇 张立			IS

构造查询的几个要素



❖ **示例元素** 即域变量 一定要加下划线

示例元素是这个域中可能的一个值，它不必是查询结果中的元素

❖ **打印操作符 P.** 实际上是显示

❖ **查询条件**

可使用比较运算符 $>$, \geq , $<$, \leq , $=$ 和 \neq

其中 $=$ 可以省略

简单查询（续）



[例 2] 查询全体学生的全部数据

Student	Sno	Sname	Ssex	Sage	Sdept
	P. <u>200215121</u>	P. <u>李勇</u>	P. <u>男</u>	P. <u>20</u>	P. <u>CS</u>

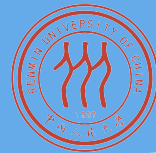
简单查询（续）



显示全部数据也可以简单地把 P. 操作符作用在关系名上。

Student	Sno	Sname	Ssex	Sage	Sdept
P.					

2. 条件查询



[例 3] 求年龄大于 19 岁的学生的学号

Student	Sno	Sname	Ssex	Sage	Sdept
	P. <u>200215121</u>			>19	

条件查询（与条件）

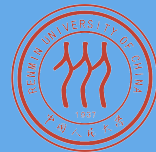


[例 4] 求计算机科学系年龄大于 19 岁的学生的学号。

方法 (1)：把两个条件写在同一行上

Student	Sno	Sname	Ssex	Sage	Sdept
	<u>P.200215121</u>			>19	CS

条件查询（与条件）



方法 (2)：把两个条件写在不同行上，但使用相同的示例元素值

Student	Sno	Sname	Ssex	Sage	Sdept
	P. <u>20021512</u> 1			>19	CS
	P. <u>20021512</u> 1				

条件查询（与条件）



[例 5] 查询既选修了 1 号课程又选修了 2 号课程的学生的学号。

Sc	Sno	Cno	Grade
	<u>P.200215121</u>	1	
	<u>P.200215121</u>	2	

条件查询（或条件）



[例 6] 查询计算机科学系或者年龄大于 19 岁的学生的

学号。

Student	Sno	Sname	Ssex	Sage	Sdept
	P. <u>200215121</u> P. <u>200215122</u>			>19	CS

条件查询（多表连接）



[例 7] 查询选修 1 号课程的学生姓名。

Student	Sno	Sname	Ssex	Sage	Sdept
	<u>200215121</u>	P. 李勇			

Sc	Sno	Cno	Grade
	<u>200215121</u>	1	

注意：示例元素 Sno 是连接属性，其值在两个表中要相同。

条件查询（非条件）



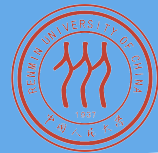
[例 8] 查询未选修 1 号课程的学生姓名

Student	Sno	Sname	Ssex	Sage	Sdept
	<u>200215121</u>	P. 李勇			

Sc	Sno	Cno	Grade
—	<u>200215121</u>	1	

思路：显示学号为 200215121 的学生名字，而该学生选修 1 号课程的情况为假

条件查询（续）



[例 9] 查询有两个人以上选修的课程号。

Sc	Sno	Cno	Grade
	<u>200215121</u>	P. <u>1</u>	
	— <u>200215121</u>	<u>1</u>	

思路：查询这样的课程 1，它不仅被 200215121 选修
而且也被另一个学生 (\neg 200215121) 选修了

3. 聚集函数

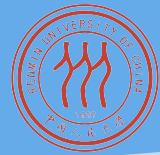


常用聚集函数：

函数名	功能
CNT	对元组计数
SUM	求总和
AVG	求平均值
MAX	求最大值
MIN	求最小值

QBE 中的聚集函数

聚集函数（续）



[例 10] 查询信息系学生的平均年龄。

Student	Sno	Sname	Ssex	Sage	Sdept
				P.AVG.ALL	IS

4. 对查询结果排序



❖ 升序排序：

- 对查询结果按某个属性值的升序排序，只需在相应列中填入 “**AO.**”

❖ 降序排序：

- 按降序排序则填 “**DO.**”

❖ 多列排序：

- 如果按多列排序，用 “**AO(i).**” 或 “**DO(i).**” 表示，其中 i 为排序的优先级， i 值越小，优先级越高

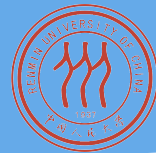
对查询结果排序（续）



[例 11] 查全体男生的姓名，要求查询结果按所在系升序排序，对相同系的学生按年龄降序排序。

Student	Sno	Sname	Ssex	Sage	Sdept
		P. <u>李勇</u>	男	DO (2) .	AO (1) .

二、更新操作



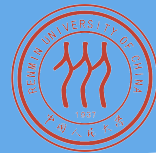
1. 修改操作

[例 12] 把 200215121 学生的年龄改为 18 岁。

方法 (1)：将操作符 “U.” 放在值上

Student	Sno	Sname	Ssex	Sage	Sdept
	200215121			U.18	

修改操作（续）



方法 (2)： 将操作符 “U.” 放在关系上

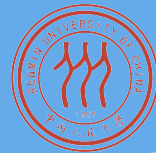
Student	Sno	Sname	Ssex	Sage	Sdept
U.	200215121			18	

码 200215121 标明要修改的元组。

“U.” 标明所在的行是修改后的新值。

由于主码是不能修改的，所以系统不会混淆要修改的属性。

修改操作（续）

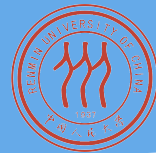


[例 13] 把 200215121 学生的年龄增加 1 岁

Student	Sno	Sname	Ssex	Sage	Sdept
U.	200215121			<u>17</u>	
	200215121			<u>17</u> +1	

操作涉及表达式，必须将操作符 “U.” 放在关系上

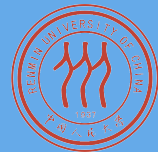
修改操作（续）



[例 14] 将计算机系所有学生的年龄都增加 1 岁

Student	Sno	Sname	Ssex	Sage	Sdept
U.	<u>200215122</u>			<u>18</u>	CS
	<u>200215122</u>			<u>18+1</u>	

2. 插入操作

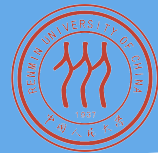


[例 15] 把信息系女生 200215701，姓名张三，年龄

17 岁存入数据库中。

Student	Sno	Sname	Ssex	Sage	Sdept
I.	200215701	张三	女	17	IS

3. 删除操作



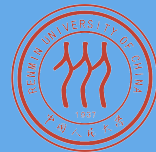
[例 17] 删除学生 200215089

Student	Sno	Sname	Ssex	Sage	Sdept
D.	200215089				

为保证参照完整性，删除 200215089 学生前，先删除 200215089 学生选修的全部课程

Sc	Sno	Cno	Grade
D.	200215089		

第二章 关系数据库



2.1 关系模型概述

2.2 关系数据结构

2.3 关系的完整性

2.4 关系代数

2.5 关系演算

2.6 小结

2.6 小结



- ❖ 关系数据库系统是目前使用最广泛的数据库系统
- ❖ 关系数据库系统与非关系数据库系统的区别：
 - 关系系统只有“表”这一种数据结构；
 - 非关系数据库系统还有其他数据结构，以及对这些数据结构的操作

小结（续）



❖ 关系数据结构

- 关系
 - 域
 - 笛卡尔积
 - 关系
 - 关系，属性，元组
 - 候选码，主码，主属性
 - 基本关系的性质
- 关系模式
- 关系数据库

小结（续）



❖ 关系操作

- 查询

- 选择、投影、连接、除、并、交、差

- 数据更新

- 插入、删除、修改

小结（续）



❖ 关系的完整性约束

- 实体完整性
- 参照完整性
 - 外码
- 用户定义的完整性

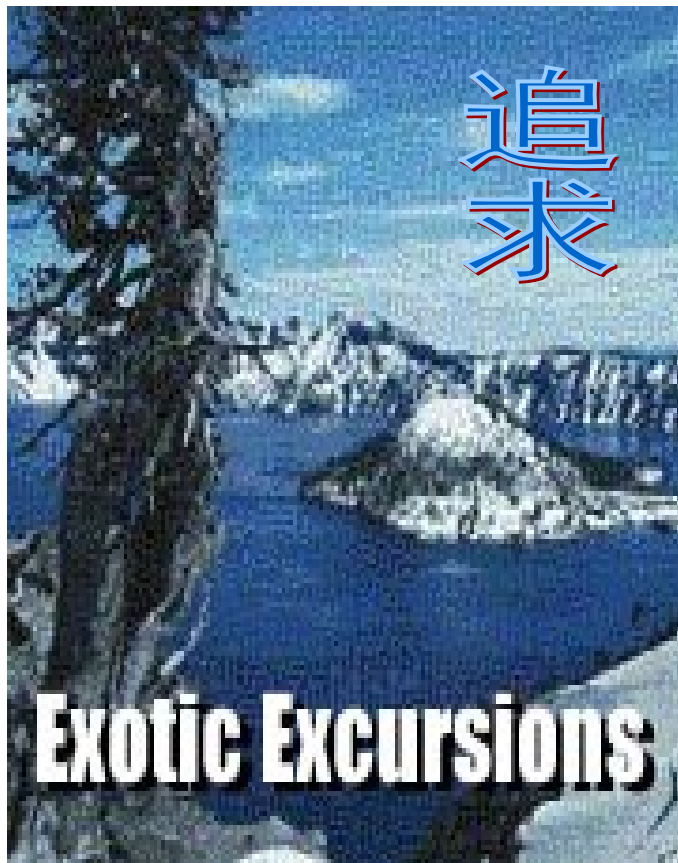
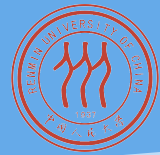
小结（续）



❖ 关系数据语言

- 关系代数语言
- 关系演算语言
 - 元组关系演算语言 ALPHA
 - 域关系演算语言 QBE

下课了。。。



休息一会儿。。。

