



# 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](mailto:qiliuql@ustc.edu.cn)

课程主页:

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



## 3.4.3 嵌套查询

2

- 一、带有**IN**谓词的子查询
- 二、带有比较运算符的子查询
- 三、带有**ANY (SOME)** 或**ALL**谓词的子查询
- 四、带有**EXISTS**谓词的子查询



### 三、带有ANY (SOME) 或ALL谓词的子查询

3

#### 谓词语义

- ⑩ **ANY:** 任意一个值
- ⑩ **ALL:** 所有值



## 带有ANY (SOME) 或ALL谓词的子查询 (续)

4

### 需要配合使用比较运算符

> ANY	大于子查询结果中的某个值
> ALL	大于子查询结果中的所有值
< ANY	小于子查询结果中的某个值
< ALL	小于子查询结果中的所有值
>= ANY	大于等于子查询结果中的某个值
>= ALL	大于等于子查询结果中的所有值
<= ANY	小于等于子查询结果中的某个值
<= ALL	小于等于子查询结果中的所有值
= ANY	等于子查询结果中的某个值
= ALL	等于子查询结果中的所有值 (通常没有实际意义)
!= (或<>) ANY	不等于子查询结果中的某个值
!= (或<>) ALL	不等于子查询结果中的任何一个值



## 带有ANY (SOME) 或ALL谓词的子查询 (续)

5

**[例42]** 查询其他系中比计算机科学**某一**学生年龄小的学生姓名和年龄

```
SELECT Sname, Sage
```

```
FROM Student
```

```
WHERE Sage < ANY (SELECT Sage
```

```
FROM Student
```

```
WHERE Sdept= ' CS ')
```

```
AND Sdept <> 'CS ' ;      /*父查询块中的条件 */
```



## 带有ANY (SOME) 或ALL谓词的子查询 (续)

结果:

Sname	Sage
王敏	18
张立	19

执行过程:

1. **RDBMS**执行此查询时, 首先处理子查询, 找出  
**CS**系中所有学生的年龄, 构成一个集合(**20**, **19**)
2. 处理父查询, 找所有不是**CS**系且年龄小于  
**20 或 19**的学生



## 带有ANY (SOME) 或ALL谓词的子查询 (续)

7

**[例42]** 查询其他系中比计算机科学**某一**学生年龄小的学生姓名和年龄

用聚集函数实现[例42]

```
SELECT Sname, Sage
FROM Student
WHERE Sage <
      (SELECT MAX(Sage)
       FROM Student
       WHERE Sdept= 'CS ')
AND Sdept <> 'CS ';
```



## 带有ANY (SOME) 或ALL谓词的子查询 (续)

8

**[例43]** 查询其他系中比计算机科学系**所有**学生年龄都小的学生姓名及年龄。

方法一：用ALL谓词

```
SELECT Sname, Sage
FROM Student
WHERE Sage < ALL
      (SELECT Sage
       FROM Student
       WHERE Sdept= ' CS ')
AND Sdept <> ' CS ';
```





## 帶有ANY (SOME) 或ALL谓词的子查询 (续)

9

方法二：用聚集函数

```
SELECT Sname, Sage
FROM Student
WHERE Sage <
      (SELECT MIN(Sage)
       FROM Student
       WHERE Sdept= ' CS ')
AND Sdept <> ' CS ';
```



## 带有ANY (SOME) 或ALL谓词的子查询 (续)

10

表3.5 ANY (或SOME), ALL谓词与聚集函数、IN谓词的等价转换关系(例如, =ANY等价于IN)

	=	<>或!=	<	<=	>	>=
ANY	IN	--	<MAX	<=MAX	>MIN	>= MIN
ALL	--	NOT IN	<MIN	<= MIN	>MAX	>= MAX



## 练习

11

- 嵌套查询命令中的IN，相当于（）
  - A. 等号=      B. 集合运算符  $\in$
  - C. 加号+                  D. 减号-
- 下面关于SELECT嵌套语句的叙述中，错误的是（）。
  - A) 首先应对子查询求值
  - B) 外部查询依赖于子查询的求值结果
  - C) 子查询必须被括在圆括号中
  - D) 子查询的结果会被显示出来



## 作业-8 用SQL实现简单查询

12

- 学生选课数据库由以下三个关系模式组成：
  - 学生关系 S (Sno, SN, Sdept, Age)
  - 课程关系 C (Cno, CN, Teacher)
  - 学生选课关系 SC (Sno, Cno, Grade)
- 关系模式中各属性含义是：Sno 学生号，SN 学生名，Sdept 学生所在系，Age 年龄，Cno 课程号，CN 课程名，Teacher 授课教师，Grade 成绩。
- 用SQL语言实现下列的操作：
  - (1) 查询“计算机系”年龄18岁以下的学生名单；
  - (2) 查询选修‘数据库’课程且分数在90到100分之间的学生的学号、姓名、成绩，要求按成绩降序排序；
  - (3) 按课程号统计各门课程的选课人数、最高分、最低分、平均分；
  - (4) 将所有选修“数据库”课程的成绩增加10分（假设原成绩无超过90分的）。



### 3.4.3 嵌套查询

13

- 一、带有**IN**谓词的子查询
- 二、带有比较运算符的子查询
- 三、带有**ANY** (**SOME**) 或**ALL**谓词的子查询
- 四、带有**EXISTS**谓词的子查询



# 带有EXISTS谓词的子查询(续)

14

## □ 1. EXISTS谓词

- 存在量词 $\exists$
- 带有EXISTS谓词的子查询不返回任何数据，只产生逻辑真值“true”或逻辑假值“false”。
  - 若内层查询结果非空，则外层的WHERE子句返回真值
  - 若内层查询结果为空，则外层的WHERE子句返回假值
- 由EXISTS引出的子查询，其目标列表表达式通常都用\*，因为带EXISTS的子查询只返回真值或假值，给出列名无实际意义

## □ 2. NOT EXISTS谓词

- 若内层查询结果非空，则外层的WHERE子句返回假值
- 若内层查询结果为空，则外层的WHERE子句返回真值



## 带有EXISTS谓词的子查询(续)

15

[例44]查询所有选修了1号课程的学生姓名。

思路分析:

- 本查询涉及**Student**和**SC**关系
- 在**Student**中依次取每个元组的**Sno**值，用此值去检查**SC**关系
- 若**SC**中存在这样的元组，其**Sno**值等于此**Student.Sno**值，并且其**Cno**= '1'，则取此**Student.Sname**送入结果关系



## 带有EXISTS谓词的子查询(续)

16

[例44]查询所有选修了1号课程的学生姓名。

■ 用连接运算

**SELECT Sname**

**FROM Student, SC**

**WHERE Student.Sno=SC.Sno AND SC.Cno= '1';**





## 带有EXISTS谓词的子查询(续)

17

[例44]查询所有选修了1号课程的学生姓名。

- 用嵌套查询

**SELECT Sname**

**FROM Student**

**WHERE EXISTS**

**(SELECT \***

**FROM SC**

**WHERE Sno=Student.Sno AND Cno= ' 1 ' );**



## 带有EXISTS谓词的子查询(续)

18

[例45] 查询没有选修1号课程的学生姓名。

```
SELECT Sname
FROM Student
WHERE NOT EXISTS
    (SELECT *
     FROM SC
     WHERE Sno = Student.Sno AND Cno='1');
```



## 带有EXISTS谓词的子查询(续)

19

- 不同形式的查询间的替换
  - 一些带**EXISTS**或**NOT EXISTS**谓词的子查询不能被其他形式的子查询等价替换
  - 所有带**IN**谓词、比较运算符、**ANY**和**ALL**谓词的子查询都能用带**EXISTS**谓词的子查询等价替换

- 用**EXISTS/NOT EXISTS**实现全称量词(难点)

SQL语言中没有全称量词 $\forall$  (For all)

可以把带有全称量词的谓词转换为等价的带有存在量词的谓词:

$$(\forall x)P \equiv \neg (\exists x(\neg P))$$



## 带有EXISTS谓词的子查询(续)

20

例: [例39]查询与“刘晨”在同一个系学习的学生。

可以用带EXISTS谓词的子查询替换:

```
SELECT Sno, Sname, Sdept
FROM Student S1
WHERE EXISTS
    (SELECT *
     FROM Student S2
     WHERE S2.Sdept = S1.Sdept AND
           S2.Sname = '刘晨' );
```



## 带有EXISTS谓词的子查询(续)

21

[例46] 查询选修了全部课程的学生姓名。

SELECT Sname

FROM Student

WHERE **NOT EXISTS**

(SELECT \*

FROM Course

WHERE **NOT EXISTS**

(SELECT \*

FROM SC

WHERE Sno= Student.Sno

AND Cno= Course.Cno

)

);

1.任意一个学生A

2.不存在一个课程a

3.课程a没有被学生A选修  
(即, 在选课记录表里  
不存在A对a的选课记录)

不存在一个课程, 这个课程没有被选修  
(双重否定)



## 带有EXISTS谓词的子查询(续)

22

### 用EXISTS/NOT EXISTS实现逻辑蕴涵(难点)

- SQL语言中没有蕴涵(Implication)逻辑运算
- 可以利用谓词演算将逻辑蕴涵谓词等价转换为:

$$p \rightarrow q \equiv \neg p \vee q$$

p	q	$p \rightarrow q$	$\neg p \vee q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

Why?



# 命题逻辑

23

- 逻辑(**Logic**):表示知识并进行推理
  - 命题逻辑、谓词逻辑
- 命题: 能判断真假的陈述句。
- 真值: 一个命题表达的判断结果称为命题的**真值**。命题的真值有“真”和“假”两种, 分别用True、T、1(真)和False、F、0(假)来表示。真值为真的命题称为真命题, 真值为假的命题称为假命题。任何命题的真值是惟一的。
- 注: 一切没有判断内容的句子, 无所谓是非的句子, 如感叹句、疑问句、祈使句等都不是命题。



# 命题逻辑

24

1. 2是素数。
2. 雪是黑色的。
3.  $2+3=5$  。
4. 明年十月一日是晴天。
5. 这朵花多好看呀！
6. 3能被2整除.
7. 明天下午有会吗？
8. 请关上门！
9.  $x+y>5$  。

命题判断的关键：  
1. 是否是陈述句；  
2. 真值是否是唯一的。





# 命题逻辑

25

## 表示法:

	表示法	例子	有关概念
简单命题	$p, q, r, \dots,$ $p_i, q_i, r_i, \dots$	$p$ : 2是素数 $q$ : 雪是黑色的	命题符号化: 将命题的符号放在该命题的前面
命题常项 (常元)	同上	同上	真值确定的简单命题
命题变项 (变元)	同上	$p$ : $x+y>5$	真值可以变化的简单陈述句
复合命题	$p \wedge q$	2是素数和偶数	

注: 一个符号表示的是命题常项还是命题变项由上下文决定。



# 命题逻辑

26

## •命题联结词

常用的逻辑联结词有五种：否定联结词、合取联结词、析取联结词、蕴涵联结词和等价联结词。

### 1. 否定联结词

定义1.1 设 $p$ 为命题，则 $p$ 的否定是一个复合命题，记作： $\neg p$ ，读作“非 $p$ ”或“ $p$ 的否定”。 $\neg$ 为**否定联结词**。 $\neg p$ 为真当且仅当 $p$ 为假。

表1.1

$p$	$\neg p$
0	1
1	0

【例】否定下列命题。

$p$ : 王强是一名大学生。

$\neg p$ : 王强不是一名大学生。



# 命题逻辑

27

## 2. 合取联结词

**定义1.2** 设 $p$ 和 $q$ 均为命题，则 $p$ 和 $q$ 的合取是一个复合命题，记作 $p \wedge q$ ，读作“ $p$ 与 $q$ ”或“ $p$ 合取 $q$ ”。 $\wedge$ 为**合取联结词**。 $p \wedge q$ 为真当且仅当 $p$ 和 $q$ 同时为真。

表1.2

$p$	$q$	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

【例】设  $p$ : 北京成功举办了第29届夏季奥运会。

$q$ : 今年10月1日是我国国庆60周年。

则 $p \wedge q$ : 北京成功举办了第29届夏季奥运会并且今年10月1日是我国国庆60周年。



# 命题逻辑

28

## 3. 析取联结词

**定义1.3** 设 $p$ 和 $q$ 均为命题，则 $p$ 和 $q$ 的析取是一个复合命题，记作 $p \vee q$ ，读作“ $p$ 或 $q$ ”或者“ $p$ 析取 $q$ ”。 $\vee$ 为**析取联结词**。 $p \vee q$ 为真当且仅当 $p$ 与 $q$ 中至少一个为真。

表1.3

$p$	$q$	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

“ $\vee$ ”与汉语中的“或”相似，但又不相同。汉语中的或有可兼或与不可兼或(排斥或)的区分。

**【例】**下列两个命题中的“或”，哪个是可兼或？哪个是不可兼或？

- (1)在家里看奥运会或在现场看奥运会。(不可兼或)
- (2)灯泡有故障或开关有故障。(可兼或)



## 4. 蕴涵联结词

**定义1.4** 设 $p$ 和 $q$ 均为命题， $p$ 与 $q$ 的蕴涵式是个复合命题，记为： $p \rightarrow q$ 。读作“如果 $p$ ，那么 $q$ ”或“若 $p$ ，则 $q$ ”。 $\rightarrow$ 为**蕴涵联结词**。 $p \rightarrow q$ 为假当且仅当 $p$ 为真且 $q$ 为假。 $p$ 称为条件命题 $p \rightarrow q$ 的前件， $q$ 称为条件命题 $p \rightarrow q$ 的后件。

表1.4

$p$	$q$	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

**【例】**  $p$ ：小王努力学习。 $q$ ：小王学习成绩优秀。

$p \rightarrow q$ ：如果小王努力学习，那么他的学习成绩就优秀。

联结词“ $\rightarrow$ ”与汉语中的“如果…，那么…”或“若…，则…”相似，但又是不同的。



# 命题逻辑

30

将下列各命题符号化（并判断3-6的真值情况）

1. 只要不下雨，我就骑自行车上班.
2. 只有不下雨，我才骑自行车上班.
3. 若 $2+2=4$ ，则太阳从东方升起.
4. 若 $2+2\neq 4$ ，则太阳从东方升起.
5. 若 $2+2=4$ ，则太阳从西方升起.
6. 若 $2+2\neq 4$ ，则太阳从西方升起.



# 命题逻辑

31

## 5. 等价联结词

定义1.5 设 $p$ 和 $q$ 均为命题，其复合命题 $p \leftrightarrow q$ 称为等价式， $p \leftrightarrow q$ 读作：“ $p$ 当且仅当 $q$ ”。 $\leftrightarrow$ 为**等价联结词**。 $p \leftrightarrow q$ 为真当且仅当 $p$ 和 $q$ 的真值相同。

表1.5

$p$	$q$	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

【例】  $p$ : 张华是三好学生。

$q$ : 张华德、智、体全优秀。

$p \leftrightarrow q$ : 张华是三好学生当且仅当德、智、体全优秀。



# 命题逻辑

32

## • 联结词的比较表

$p, q$  为两个命题

复合命题	…式	记作	联结词	为真的条件 iff	优先级	其他
否定	非 $p$ ( $p$ 的否定)	$\neg p$	$\neg$	$p$ 为假	1	
合取	$p$ 并且 $q$ ( $p$ 和 $q$ )	$p \wedge q$	$\wedge$	$p$ 与 $q$ 同时为真	2	既…又…, 不仅…而且
析取	$P$ 或 $q$	$p \vee q$	$\vee$	$p$ 与 $q$ 至少一个为真	3	相容或
蕴含	如果 $p$ 则 $q$	$p \rightarrow q$	$\rightarrow$	$\neg(P \text{ 为真且 } q \text{ 为假})$	4	只要 $p$ 就 $q$ , $p$ 仅当 $q$
等价	$P$ 当且仅当 $q$	$p \leftrightarrow q$	$\leftrightarrow$	$p, q$ 真值相同	5	

注：以上5种联结词也称真值联结词或逻辑联结词或逻辑运算符。





# 等值演算—逻辑等价

- 设 $A, B$ 为两个命题公式，若等价式 $A \leftrightarrow B$ 是重言式，则称 $A$ 与 $B$ 是等值的，记作 $A \Leftrightarrow B$ .
- $A \Leftrightarrow B$ 不是命题公式
- 可通过判断 $A$ 与 $B$ 的真值表是否相同，来判断 $A$ 与 $B$ 是否等值。



# 等值演算—逻辑等价

□ 判断下列命题公式是否等值

(1)  $\neg(p \vee q)$  与  $\neg p \vee \neg q$  ;

(2)  $\neg(p \vee q)$  与  $\neg p \wedge \neg q$  ;

$p$	$q$	$\neg p$	$\neg q$	$p \vee q$	$\neg(p \vee q)$	$\neg p \vee \neg q$	$\neg p \wedge \neg q$
0	0	1	1	0	1	1	1
0	1	1	0	1	0	1	0
1	0	0	1	1	0	1	0
1	1	0	0	1	0	0	0



24 个重要的等值式：  $\neg \wedge \vee \rightarrow \leftrightarrow \Leftrightarrow$

编号	表达式	中文名称
1	$A \Leftrightarrow \neg \neg A$	双重否定律
2	$A \Leftrightarrow A \vee A$	等幂律 (或幂等律)
3	$A \Leftrightarrow A \wedge A$	
4	$A \vee B \Leftrightarrow B \vee A$	交换律
5	$A \wedge B \Leftrightarrow B \wedge A$	
6	$(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C)$	结合律
7	$(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$	
8	$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$	分配律
9	$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$	
10	$\neg (A \vee B) \Leftrightarrow \neg A \wedge \neg B$	德 ● 摩根律
11	$\neg (A \wedge B) \Leftrightarrow \neg A \vee \neg B$	
12	$A \vee (A \wedge B) \Leftrightarrow A$	吸收律
13	$A \wedge (A \vee B) \Leftrightarrow A$	



14	$A \vee 1 \Leftrightarrow 1$	零律
15	$A \wedge 0 \Leftrightarrow 0$	
16	$A \vee 0 \Leftrightarrow A$	同一律
17	$A \wedge 1 \Leftrightarrow A$	
18	$A \vee \neg A \Leftrightarrow 1$	排中律
19	$A \wedge \neg A \Leftrightarrow 0$	矛盾律
20	$A \rightarrow B \Leftrightarrow \neg A \vee B$	蕴涵等值式
21	$A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$	等价等值式
22	$A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$	假言易位
23	$A \leftrightarrow B \Leftrightarrow \neg A \leftrightarrow \neg B$	等价否定等值律
24	$(A \rightarrow B) \wedge (A \rightarrow \neg B) \Leftrightarrow \neg A$	归缪论

●根据已知的等值式推演出另外一些等值式的过程称为等值演算。



# 谓词逻辑

37

## □ 命题逻辑的问题

### □ 例2： 苏格拉底三段论

□ P: 所有人必死 Q: 苏格拉底是人 R: 苏格拉底必死

□ 应该有  $(P \wedge Q) \Rightarrow R$ , 也就是公式  $(P \wedge Q) \rightarrow R$  应该是恒真的。

□ 显然该公式不是恒真的, 解释  $\{P, Q, \neg R\}$  就能弄假该公式。

因为苏格拉底不等于所有人

□ 原因: 命题R和命题P, Q是有内在关系的, 只是这种关系在命题逻辑中无法表示。

□ 因此, 需要对命题的成分、结构和命题间的共同特性等作进一步的分析, 分析出对象(苏格拉底)、谓词(是人)和量词(所有), 以期达到表达出个体与总体的内在联系和数量关系, 这正是谓词逻辑所要研究的问题。



## 谓词逻辑

38

- 引入谓词后，命题P就可确切地符号化如下：

$$\forall x(H(x) \rightarrow M(x))$$

命题P的否定命题为：

$$\begin{aligned}\neg P &= \neg(\forall x(H(x) \rightarrow M(x))) \\ &= \exists x(H(x) \wedge \neg M(x))\end{aligned}$$

亦即“至少有一个人是不死的”。这个命题才是“所有人都要死”的否定。

- 三段论的三个命题，在谓词逻辑中可以如下表示：

$$P: \forall x(H(x) \rightarrow M(x))$$

$$Q: H(\text{苏格拉底})$$

$$R: M(\text{苏格拉底})$$



# 谓词逻辑

39

## □ 我们只关注于量词

(1) 所有的老虎都会吃人。

(1) 令  $P(x)$ :  $x$ 会吃人     $U(x)$ :  $x$ 是老虎

应该使用全称量词规则，将特性谓词作为前件加入  
则符号化的正确形式应该是

$$(\forall x)(U(x) \rightarrow P(x))$$

“任意 $x$ , 若 $x$ 是老虎, 则 $x$ 会吃人”, 符合原命题的逻辑含义



# 谓词逻辑

40

## □ 我们只关注于量词

(2) 有些人登上过月球。

(2) 令  $S(x)$ :  $x$ 登上过月球

$U(x)$ :  $x$ 是人

应该使用存在量词规则，将特性谓词作为合取式前项加入  
则符号化正确形式为

$$(\exists x)(U(x) \wedge S(x))$$

- “存在 $x$ ,  $x$ 是人并且 $x$ 登上过月球”，符合原命题的逻辑含义





## 谓词逻辑

41

### (2) 天下乌鸦一般黑

设  $F(x)$ :  $x$  是乌鸦;  $G(x, y)$ :  $x$  与  $y$  一般黑, 则:

$$(\forall x)(\forall y)(F(x) \wedge F(y) \rightarrow G(x, y))$$

或者  $\neg (\exists x)(\exists y)(F(x) \wedge F(y) \wedge \neg G(x, y))$ ;

### (3) 没有人登上过木星

设  $H(x)$ :  $x$  是人;  $M(x)$ :  $x$  登上过木星, 则:

$$\neg (\exists x)(H(x) \wedge M(x)) \Leftrightarrow (\forall x)(\neg H(x) \vee (\neg M(x)))$$

或者  $(\forall x)(H(x) \rightarrow \neg M(x)) \Leftrightarrow (\forall x)(\neg H(x) \vee (\neg M(x)))$ ;



## 带有EXISTS谓词的子查询(续)

42

**[例47]**查询至少选修了学生200215122选修的全部课程的学生号码。

解题思路:

- 用逻辑蕴涵表达: 查询学号为x的学生, 对所有的课程y, 只要200215122学生选修了课程y, 则x也选修了y。

- 形式化表示:

用P(y)表示谓词 “学生200215122选修了课程y”

用q(y)表示谓词 “学生x选修了课程y”

则上述查询为:  $(\forall y) p \rightarrow q$



## 带有EXISTS谓词的子查询(续)

43

- 等价变换:

$$\begin{aligned}(\forall y) p \rightarrow q &\equiv \neg (\exists y (\neg(p \rightarrow q))) \\ &\equiv \neg (\exists y (\neg(\neg p \vee q))) \\ &\equiv \neg \exists y (p \wedge \neg q)\end{aligned}$$

- 变换后语义: 不存在这样的课程y, 学生200215122选修了y, 而学生x没有选。



## 带有EXISTS谓词的子查询(续)

44

- 不存在这样的课程y，学生200215122选修了y，而学生x没选。
- 用NOT EXISTS谓词表示：

```
SELECT DISTINCT Sno
FROM SC SCX
WHERE NOT EXISTS
    (SELECT *
     FROM SC SCY
     WHERE SCY.Sno = ' 200215122 ' AND
          NOT EXISTS
            (SELECT *
             FROM SC SCZ
             WHERE SCZ.Sno=SCX.Sno AND
                   SCZ.Cno=SCY.Cno));
```



## 3.4 数据查询

45

- 3.4.1 单表查询
- 3.4.2 连接查询
- 3.4.3 嵌套查询
- 3.4.4 集合查询
- 3.4.5 Select语句的一般形式



## 3.4.4 集合查询

46

- 集合操作的种类
  - 并操作**UNION**
  - 交操作**INTERSECT**
  - 差操作**EXCEPT**
- 参加集合操作的各查询结果的列数必须相同；对应项的数据类型也必须相同



## 集合查询（续）

47

[例48] 查询计算机科学系的学生及年龄不大于19岁的学生。

方法一：

```
SELECT *  
FROM Student  
WHERE Sdept= 'CS'  
UNION  
SELECT *  
FROM Student  
WHERE Sage<=19;
```

- **UNION**: 将多个查询结果合并起来时，系统自动去掉重复元组。
- **UNION ALL**: 将多个查询结果合并起来时，保留重复元组



## 集合查询（续）

48

方法二：

```
SELECT DISTINCT *  
FROM Student  
WHERE Sdept= 'CS' OR Sage<=19;
```





## 集合查询（续）

49

【例49】 查询选修了课程1或者选修了课程2的学生。

```
SELECT Sno  
FROM SC  
WHERE Cno=' 1 '  
UNION  
SELECT Sno  
FROM SC  
WHERE Cno= ' 2 ';
```



## 集合查询（续）

50

**[例50]** 查询计算机科学系的学生与年龄不大于19岁的学生的交集

```
SELECT *  
FROM Student  
WHERE Sdept='CS'  
INTERSECT  
SELECT *  
FROM Student  
WHERE Sage<=19
```



## 集合查询（续）

51

- [例50] 实际上就是查询计算机科学系中年龄不大于19岁的学生

```
SELECT *  
FROM Student  
WHERE Sdept= 'CS' AND Sage<=19;
```



## 集合查询（续）

52

[例51] 查询选修课程1的学生集合与选修课程2的学生集合的交集

```
SELECT Sno  
FROM SC  
WHERE Cno=' 1 '  
INTERSECT  
SELECT Sno  
FROM SC  
WHERE Cno='2 ';
```



## 集合查询（续）

53

[例51]实际上是查询既选修了课程1又选修了课程2 的学生

```
SELECT Sno
FROM SC
WHERE Cno=' 1 ' AND Sno IN
        (SELECT Sno
         FROM SC
         WHERE Cno=' 2 ');
```



## 集合查询（续）

54

[例52] 查询计算机科学系的学生与年龄不大于19岁的学生的差集。

```
SELECT *  
FROM Student  
WHERE Sdept='CS'
```

**EXCEPT**

```
SELECT *  
FROM Student  
WHERE Sage <=19;
```

**$R - S$**

仍为 $n$ 目关系，由属于 $R$ 而不属于 $S$ 的所有元组组成

$$R - S = \{ t | t \in R \wedge t \notin S \}$$



## 集合查询（续）

55

[例52]实际上是查询计算机科学系中年龄大于19岁的学生

```
SELECT *  
FROM Student  
WHERE Sdept= 'CS' AND Sage>19;
```



## 3.4 数据查询

56

- 3.4.1 单表查询
- 3.4.2 连接查询
- 3.4.3 嵌套查询
- 3.4.4 集合查询
- 3.4.5 Select语句的一般形式





## 3.4.5 SELECT语句的一般格式

57

**SELECT** [ALL|DISTINCT]

<目标列表表达式> [别名] [ , <目标列表表达式> [别名]] ...

**FROM** <表名或视图名> [别名]

[ , <表名或视图名> [别名]] ...

[**WHERE** <条件表达式>]

[**GROUP BY** <列名1>

[**HAVING** <条件表达式>]

[**ORDER BY** <列名2> [ASC|DESC]



## 第三章 关系数据库标准语言SQL

58

### 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



## 3.5 数据更新

59

### 3.5.1 插入数据

### 3.5.2 修改数据

### 3.5.3 删除数据



## 3.5.1 插入数据

60

- 两种插入数据方式
  1. 插入元组
  2. 插入子查询结果
- 可以一次插入多个元组



# 一、插入元组

61

## □ 语句格式

**INSERT**

**INTO <表名> [( <属性列1>[, <属性列2>...])]**

**VALUES ( <常量1> [, <常量2>] ... )**

## □ 功能

- 将新元组插入指定表中



## 插入元组（续）

62

- **INTO**子句
  - 属性列的顺序可与表定义中的顺序不一致
  - 没有指定属性列
  - 指定部分属性列
- **VALUES**子句
  - 提供的值必须与**INTO**子句匹配
    - 值的个数
    - 值的类型

**Into子句没有出现的属性列，新元组在这些列上取Null;  
在表定义时说明了Not Null的属性列不能取空，否则报错;**



## 插入元组（续）

63

[例1] 将一个新学生元组（学号：200215128；姓名：陈冬；性别：男；所在系：IS；年龄：18岁）插入到 Student 表中。

```
1 • INSERT
2     INTO Student (Sno, Sname, Ssex, Sdept, Sage)
3     VALUES ("200215129", "陈春", "男", "IS", 18);
4 • SELECT * FROM student:
```

result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
Sno	Sname	Ssex	Sage	Sdept
200215128	陈冬	男	18	IS
200215129	陈春	男	18	IS

注意单引号或者双引号的使用



## 插入元组（续）

64

[例2] 将学生张成民的信息插入到Student表中。

INSERT

INTO Student

VALUES ('200215126', '张成民', '男', 18, 'CS');

```
1 • INSERT
2 INTO Student
3 VALUES ("200215126", "张成民", "男", 18, "CS");
4 • SELECT * FROM student;
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
Sno	Sname	Ssex	Sage	Sdept
200215126	张成民	男	18	CS
200215128	陈冬	男	18	IS





## 插入元组（续）

65

[例3] 插入一条选课记录( '200215128', '1 ' )。

**INSERT**

**INTO SC(Sno, Cno)**

**VALUES ( ' 200215128 ', ' 1 ' );**

**RDBMS**将在新插入记录的**Grade**列上自动地赋空值。

或者：

**INSERT**

**INTO SC**

**VALUES ( ' 200215128 ', ' 1 ', NULL);**



## 查询表中有哪些属性列

66

- SHOW COLUMNS FROM 表名;

2 • SHOW COLUMNS FROM student;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
	Sno	char(9)	NO	PRI	NULL	
▶	Sname	char(20)	YES	UNI	NULL	
	Ssex	char(2)	YES		NULL	
	Sage	int(11)	NO		NULL	
	Sdept	char(20)	YES		NULL	



## 二、插入子查询结果

67

### □ 语句格式

**INSERT**

**INTO** <表名> [(<属性列1> [, <属性列2>... ])

子查询;

### □ 功能

将子查询结果插入指定表中



## 插入子查询结果（续）

68

- INTO子句(与插入元组类似)
- 子查询
  - SELECT子句目标列必须与INTO子句匹配
    - 值的个数
    - 值的类型



## 插入子查询结果（续）

69

**[例4]** 对每一个系，求学生的平均年龄，并把结果存入数据库。

第一步：建表

```
CREATE TABLE Dept_age  
  (Sdept CHAR(15)          /* 系名*/  
   Avg_age SMALLINT);     /*学生平均年龄*/
```



## 插入子查询结果（续）

70

第二步：插入数据

```
INSERT  
INTO Dept_age(Sdept, Avg_age)  
SELECT Sdept, AVG(Sage)  
FROM Student  
GROUP BY Sdept;
```



## 插入子查询结果（续）

71

**RDBMS**在执行插入语句时会检查所插元组是否破坏表上已定义的完整性规则

- 实体完整性
- 参照完整性
- 用户定义的完整性
  - **NOT NULL**约束
  - **UNIQUE**约束
  - 值域约束



## 3.5 数据更新

72

### 3.5.1 插入数据

### 3.5.2 修改数据

### 3.5.3 删除数据





## 3.4.2 修改数据

73

### □ 语句格式

**UPDATE** <表名>

**SET** <列名>=<表达式>[, <列名>=<表达式>]...

**[WHERE** <条件>];

### □ 功能

- 修改指定表中满足**WHERE**子句条件的元组



# 修改数据（续）

74

## ■ SET子句

- 指定修改方式
- 要修改的列
- 修改后取值

## ■ WHERE子句

- 指定要修改的元组
- 缺省表示要修改表中的所有元组



## 修改数据（续）

75

### □ 三种修改方式

1. 修改某一个元组的值
2. 修改多个元组的值
3. 带子查询的修改语句



# 1. 修改某一个元组的值

76

**[例5]** 将学生200215121的年龄改为22岁

```
1 • UPDATE Student
2   SET Sage=22
3   WHERE Sno="200215126";
4 • SELECT * FROM student;
```

ult Grid    Filter Rows: <input type="text"/>   Edit:      Export/Import:				
Sno	Sname	Ssex	Sage	Sdept
200215126	张成民	男	22	CS
200215128	陈冬	男	18	IS
200215129	陈春	男	18	IS



## 2. 修改多个元组的值

77

[例6] 将某个学生的年龄增加1岁

1 •  
2  
3  
4 •

```
UPDATE Student  
SET Sage= Sage+1  
WHERE Sno="200215126";  
SELECT * FROM student;
```

ult Grid | Filter Rows:  | Edit: | Export/Import: | Wr

Sno	Sname	Ssex	Sage	Sdept
200215126	张成民	男	23	CS



### 3. 带子查询的修改语句

78

**[例7]** 将计算机科学系全体学生的成绩置零。

```
UPDATE SC  
SET Grade=0  
WHERE 'CS'=  
    (SELETE Sdept  
    FROM Student  
    WHERE Student.Sno = SC.Sno);
```



## 修改数据（续）

79

**RDBMS**在执行修改语句时会检查修改操作是否破坏表上已定义的完整性规则

- 实体完整性
- 主码不允许修改
- 用户定义的完整性
  - **NOT NULL**约束
  - **UNIQUE**约束
  - 值域约束



## 3.5 数据更新

80

3.5.1 插入数据

3.5.2 修改数据

3.5.3 删除数据





## 3.5.3 删除数据

81

- 语句格式

**DELETE**

**FROM** <表名>

**[WHERE <条件>];**

- 功能

- 删除指定表中满足**WHERE**子句条件的元组

- **WHERE**子句

- 指定要删除的元组
- 缺省表示要删除表中的所有元组，表的定义仍在字典中



## 删除数据（续）

82

- 三种删除方式
  - 1. 删除某一个元组的值
  - 2. 删除多个元组的值
  - 3. 带子查询的删除语句



## 1. 删除某一个元组的值

83

[例8] 删除学号为200215128的学生记录。

**DELETE**

**FROM Student**

**WHERE Sno= '200215128 ';**

```
1 • DELETE
2 FROM Student
3 WHERE Sno="200215128";
4 • SELECT * FROM student:
```

Sno	Sname	Ssex	Sage	Sdept
200215126	张成民	男	23	CS
200215129	陈春	男	18	IS



## 2. 删除多个元组的值

84

[例9] 删除所有的学生选课记录。

```
DELETE  
FROM SC;
```



### 3. 带子查询的删除语句

85

**[例10]** 删除计算机科学系所有学生的选课记录。

```
DELETE  
FROM SC  
WHERE 'CS' =  
    (SELETE Sdept  
        FROM Student  
        WHERE Student.Sno=SC.Sno);
```



# 空值的处理

86

- 空值NULL
  - 属性应当有值，但是现在还不清楚是多少（如，年龄）
  - 属性不应当有值，如缺考学生的成绩
  - 因某种原因不便于填写，如涉及隐私的电话号码等属性
- 空值的产生，例：

插入一条选课记录( '200215128', '1 ' )。

**INSERT**

**INTO SC(Sno, Cno)**

**VALUES ( ' 200215128 ', ' 1 ' );**

**RDBMS**将在新插入记录的**Grade**列上自动地赋空值。或者：

**INSERT**

**INTO SC**

**VALUES ( ' 200215128 ', ' 1 ', NULL );**



# 空值的处理

87

## □ 空值的判断

- 判断一个属性的值是否为空值，用IS NULL或IS NOT NULL来表示。

[例] 从Student表中找出漏填了数据的学生信息。

```
SELECT *  
FROM Student  
WHERE Sname IS NULL OR Ssex IS NULL  
OR Sage IS NULL OR Sdept IS NULL;
```

属性定义中有Not NULL约束条件的不能取空值，加了UNIQUE限制的属性不能取空，码（主）属性不能取空



# 空值的处理

88

- 空值的算术运算、比较运算和逻辑运算
  - 空值与另外值（含空值）的算术运算（如加减）结果还为空值
  - 空值与另外值（含空值）的比较运算结果为**UNKNOWN**
    - 三值逻辑：TRUE、FALSE、UNKNOWN
  - 在查询语句中，只有使WHERE和HAVING子句中的选择条件为**TRUE的元组**才被选出作为输出结果





## 空值的处理

89

- 空值的算术运算、比较运算和逻辑运算

[例] 找出选修了1号课程的不及格的学生

```
SELECT Sno  
FROM SC  
WHERE Grade < 60 AND Cno='1 ';
```

只有参加了考试（**Grade 属性不为空**）且不及格的学生会被选出，若学生缺考（**Grade 属性为空**），则WHERE的条件为**UNKNOWN**，不会有返回值



## 作业9 - 用SQL实现较为复杂的查询

90

2、设学生课程数据库中有三个关系：

学生关系S (S#, SNAME, AGE, SEX)

学习关系SC (S#, C#, GRADE)

课程关系C (C#, CNAME)

其中S#、C#、SNAME、AGE、SEX、GRADE、CNAME分别表示学号、课程号、姓名、年龄、性别、成绩和课程名。用SQL语句表达下列操作

- (1) 检索选修课程名称为“MATHS”的学生的学号与姓名。
- (2) 检索至少学习了课程号为“C1”和“C2”的学生的学号
- (3) 检索年龄在18到20之间（含18和20）的女生的学号、姓名和年龄
- (4) 检索平均成绩超过80分的学生学号和平均成绩
- (5) 检索选修了全部课程的学生姓名
- (6) 检索选修了三门课以上的学生的姓名



## 第三章 关系数据库标准语言SQL

91

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



## 3.6 视图

92

### 视图(View)的特点

- 虚表，是从一个或几个基本表（或视图）导出的表
- 只存放视图的定义，不存放视图对应的数据
- 基表中的数据发生变化，从视图中查询出的数据也随之改变



## 3.6 视图

93

### □ 多视图





## 3.6 视图

94

- 什么时候使用视图
  - 经常用到的查询，或较复杂的联合查询应当创立视图，以优化查询效率
    - 查询逻辑复杂，或别的查询需要查询这个查询的结果时
      - 如果你的查询需要连接几个表的数据的时候你可以做成视图，那么你在业务端只需要查询这张视图就可以了
  - 涉及到权限管理方面，用来保护敏感数据
    - 比如某表中的部分字段含有机密信息，不应当让低权限的用户访问，这时候给这些用户提供一个适合他们权限的视图，供他们阅读自己的数据即可



## 3.6 视图

95

### 基于视图的操作

- 查询
- 删除
- 受限更新
- 定义基于该视图的新视图



## 3.6 视图

96

### 3.6.1 定义视图

### 3.6.2 查询视图

### 3.6.3 更新视图

### 3.6.4 视图的作用





## 3.6.1 定义视图

97

- 建立视图
- 删除视图



## 一、建立视图

98

### □ 语句格式

**CREATE VIEW**

<视图名> [(<列名> [, <列名>]...)]

**AS** <子查询>

**[WITH CHECK OPTION];**

- 组成视图的属性列名：全部省略或全部指定
- **WITH CHECK OPTION**表示对视图进行**UPDATE**、**INSERT**和**DELETE**等操作时要保证满足视图定义中的谓词条件（即，子查询中的条件表达式）
- 子查询不允许含有**ORDER BY**子句和**DISTINCT**短语（取决于系统实现）



## 建立视图（续）

99

- ❑ RDBMS执行**CREATE VIEW**语句时只是把视图定义存入数据字典，并不执行其中的**SELECT**语句。
- ❑ 在对视图查询时，按视图的定义从基本表中将数据查出。



## 建立视图（续）

100

[例1] 建立信息系学生的视图。

```
CREATE VIEW IS_Student  
AS  
SELECT Sno, Sname, Sage  
FROM Student  
WHERE Sdept= 'IS';
```



## 建立视图（续）

101

[例2]建立信息系学生的视图，并要求进行修改和插入操作时仍需保证该视图只有信息系的学生。

```
CREATE VIEW IS_Student
```

```
AS
```

```
SELECT Sno, Sname, Sage
```

```
FROM Student
```

```
WHERE Sdept= 'IS'
```

```
WITH CHECK OPTION;
```



## 建立视图（续）

102

对IS\_Student视图的更新操作：

- 修改操作：自动加上Sdept= 'IS'的条件
- 删除操作：自动加上Sdept= 'IS'的条件
- 插入操作：自动检查Sdept属性值是否为'IS'
  - 如果不是，则拒绝该插入操作
  - 如果没有提供Sdept属性值，则自动定义Sdept为'IS'



## 建立视图（续）

103

- 行列子集视图
  - 若一个视图是从单个基本表导出，且只是去掉了基本表的某些行和某些列，但是保留了主码，则称这类视图为行列子集视图，如，**IS\_Student**

```
CREATE VIEW IS_Student
AS
SELECT Sno, Sname, Sage
FROM Student
WHERE Sdept= 'IS'
WITH CHECK OPTION;
```



## 建立视图（续）

104

### □ 基于多个基表的视图

[例3] 建立信息系选修了1号课程的学生视图。（成绩单）

```
CREATE VIEW IS_S1(Sno, Sname, Grade)
```

```
AS
```

```
SELECT Student.Sno, Sname, Grade
```

```
FROM Student, SC
```

```
WHERE Sdept= 'IS' AND
```

```
Student.Sno=SC.Sno AND
```

```
SC.Cno= '1';
```





## 建立视图（续）

105

### □ 基于视图的视图

**[例4]** 建立信息系选修了1号课程且成绩在90分以上的学生的视图。

```
CREATE VIEW IS_S2  
AS  
SELECT Sno, Sname, Grade  
FROM IS_S1  
WHERE Grade>=90;
```



## 建立视图（续）

106

### □ 带表达式的视图

[例5] 定义一个反映学生出生年份的视图。

```
CREATE VIEW BT_S(Sno, Sname, Sbirth)
AS
SELECT Sno, Sname, 2000-Sage
FROM Student;
```



## 建立视图（续）

107

### □ 分组视图

**[例6]** 将学生的学号及他的平均成绩定义为一个视图

假设SC表中“成绩”列**Grade**为数字型

```
CREAT VIEW S_G(Sno, Gavg)
```

```
AS
```

```
SELECT Sno, AVG(Grade)
```

```
FROM SC
```

```
GROUP BY Sno;
```



## 建立视图（续）

108

### □ 不指定属性列

[例7]将Student表中所有女生记录定义为一个视图

```
CREATE VIEW F_Student(F_Sno, name, sex, age, dept)
AS
SELECT * #未指定具体的属性列
FROM Student
WHERE Ssex='女' ;
```

缺点:

如果以后修改了基表Student的结构，Student表与F\_Student视图的映象关系被破坏，导致该视图不能正确工作。所以，（1）最好指定属性列；或者（2）在修改基本表以后，删除由该表导出的视图，然后重建这个视图



## 二、删除视图

109

### □ 语句的格式:

**DROP VIEW** <视图名>;

- 该语句从数据字典中删除指定的视图定义
- 如果该视图上还导出了其他视图，使用**CASCADE**级联删除语句，把该视图和由它导出的所有视图一起删除
- 删除基表时，由该基表导出的所有视图定义都必须显式地使用**DROP VIEW**语句删除



## 删除视图(续)

110

[例8] 删除视图BT\_S: **DROP VIEW BT\_S;**

删除视图IS\_S1: **DROP VIEW IS\_S1;**

- 拒绝执行
- 级联删除:

**DROP VIEW IS\_S1 CASCADE;**



## 3.6 视图

111

### 3.6.1 定义视图

### 3.6.2 查询视图

### 3.6.3 更新视图

### 3.6.4 视图的作用



## 3.6.2 查询视图

112

- 用户角度：查询视图与查询基本表相同
- RDBMS实现视图查询的方法
  - 视图消解法（**View Resolution**），步骤：
    - 进行有效性检查
    - 转换成等价的对基本表的查询
    - 执行修正后的查询





## 查询视图（续）

113

**[例9]** 在信息系学生的视图中找出年龄小于20岁的学生。

```
SELECT Sno, Sage  
FROM IS_Student  
WHERE Sage<20;
```

**IS\_Student视图的定义 (参见视图定义例1)**

```
CREATE VIEW IS_Student  
AS  
SELECT Sno, Sname, Sage  
FROM Student  
WHERE Sdept= 'IS'  
WITH CHECK OPTION;
```



## 查询视图（续）

114

视图消解转换后的查询语句为：

```
SELECT Sno, Sage  
FROM Student  
WHERE Sdept= 'IS' AND Sage<20;
```



## 查询视图（续）

115

[例10] 查询选修了1号课程的信息系学生

```
SELECT IS_Student.Sno, Sname  
FROM   IS_Student, SC  
WHERE  IS_Student.Sno = SC.Sno AND SC.Cno= '1';
```



## 查询视图（续）

116

- 视图消解法的局限
  - 有些情况下，视图消解法不能生成正确查询。



## 查询视图（续）

117

[例11]在S\_G视图中查询平均成绩在90分以上的学生学号和平均成绩

```
SELECT *  
FROM S_G  
WHERE Gavg>=90;
```

S\_G视图的子查询定义:

```
CREATE VIEW S_G (Sno, Gavg)  
AS  
SELECT Sno, AVG(Grade)  
FROM SC  
GROUP BY Sno;
```



# 查询转换

118

错误:

```
SELECT Sno, AVG(Grade)
FROM SC
WHERE AVG(Grade)>=90
GROUP BY Sno;
```

**Where子句不能用聚集函数  
作为条件表达式**

正确:

```
SELECT Sno, AVG(Grade)
FROM SC
GROUP BY Sno
HAVING AVG(Grade)>=90;
```

**目前多数关系数据库管理系统对行列子集视图的查询均能正确转换，但是对非行列子集视图的查询不一定能转换，因此这类查询应当直接对基本表进行**



## 3.6 视图

119

### 3.6.1 定义视图

### 3.6.2 查询视图

### 3.6.3 更新视图

### 3.6.4 视图的作用



## 更新视图（续）

120

[例12] 将信息系学生视图IS\_Student中学号200215122的学生姓名改为“刘辰”。

```
UPDATE IS_Student  
SET Sname= '刘辰'  
WHERE Sno= ' 200215122 ';
```

转换后的语句:

```
UPDATE Student  
SET Sname= '刘辰'  
WHERE Sno= ' 200215122 ' AND Sdept= 'IS';
```





## 更新视图（续）

121

[例13] 向信息系学生视图IS\_S中插入一个新的学生记录:

200215129, 赵新, 20岁

INSERT

INTO IS\_Student

VALUES('95029', '赵新', 20);

转换为对基本表的更新:

INSERT

INTO Student(Sno, Sname, Sage, Sdept)

VALUES('200215129 ', '赵新', 20, 'IS' );



## 更新视图（续）

122

**[例14]**删除信息系学生视图IS\_Student中学号为200215129的记录。

**DELETE**

**FROM IS\_Student**

**WHERE Sno= ' 200215129 ';**

转换为对基本表的更新：

**DELETE**

**FROM Student**

**WHERE Sno= ' 200215129 ' AND Sdept= 'IS';**



## 更新视图（续）

123

- 更新视图的限制：一些视图是不可更新的，因为对这些视图的更新不能唯一地有意义地转换成对相应基本表的更新

例：视图S\_G为不可更新视图。

```
UPDATE S_G  
SET      Gavg=90  
WHERE Sno= '200215121';
```

这个对视图的更新无法转换成对基本表SC的更新

**系统无法修改各科成绩，以使平均成绩成为90**



## 更新视图（续）

124

- 允许对行列子集视图进行更新
  - 行列子集视图是满足以下条件的视图
    - 从单个基本表导出，且只是去掉了基本表的某些行或某些列，但是保留了基本表的主码
- 对其他类型视图的更新不同系统有不同限制
  - 参考教材P127



## 3.6 视图

125

### 3.6.1 定义视图

### 3.6.2 查询视图

### 3.6.3 更新视图

### 3.6.4 视图的作用



## 3.6.4 视图的作用

126

- 1. 视图能够简化用户的操作
  - 简化查询等操作
- 2. 视图使用户能以多种角度看待同一数据
  - 便于多用户共享同一数据库时
- 3. 视图对重构数据库提供了一定程度的逻辑独立性
  - 有时数据库的逻辑结构改变了，但是应用程序不用变
- 4. 视图能够对机密数据提供安全保护
  - 机密数据不出现在低权限用户的视图中
- 5. 适当的利用视图可以更清晰的表达查询



## 练习

127

- 视图是数据库系统三层模式中的（ ）
  - A.外模式 B.模式 C.内模式 D.模式映象
- 关于视图的正确描述是（ ）
  - A)视图与数据库表相同，用来存储数据    B)视图不能同数据库表进行连接操作
  - C)在视图上不能进行更新操作    D)视图是从一个或多个表或视图导出的虚拟表
- 下列说法不正确的是（ ）
  - A.基本表和视图一样，都是关系
  - B.可以使用SQL对基本表和视图进行操作
  - C.可以从基本表或视图上定义视图
  - D.基本表和视图中都存储数据
- 删除视图的命令是（ ）
  - A.Drop SQL View XXX;    B. Drop View XXX;
  - C.Delete SQL View XXX;    D.Delete View XXX;



## 练习

128

- 在视图上不能完成的操作是（ ）
  - A 更新视图 B 查询 C 在视图上定义新的表 D 在视图上定义新的视图
- 下面关于SELECT嵌套语句的叙述中，错误的是（ ）。
  - A) 首先应对子查询求值 B) 外部查询依赖于子查询的求值结果  
C) 子查询必须被括在圆括号中 D) 子查询的结果会被显示出来
- 如果要查询借阅了两本及两本以上图书的读者姓名和单位，请对下面的SQL语句填空。
  - SELECT 姓名，单位 FROM 图书管理.读者;  
WHERE 借阅证号 IN  
(SELECT \_\_\_\_\_ FROM 图书管理.借阅  
GROUP BY 借阅证号  
\_\_\_\_\_ COUNT(\*)>=2)





下课了。。。

