



湘潭理工学院
Xiangtan Institute of Technology

数据库原理及应用

3 关系数据库标准语言SQL

3.4 数据更新



第3章 关系数据库标准语言SQL

- 1 SQL概述
- 2 数据定义
- 3 数据查询
- 4 数据更新
- 5 视图
- 6 小结

3.5 数据更新



数据更新

- 1 插入数据
- 2 修改数据
- 3 删除数据

3.5 数据更新



注意的问题

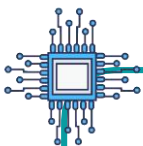
修改数据与修改基本表的区别

update与alter

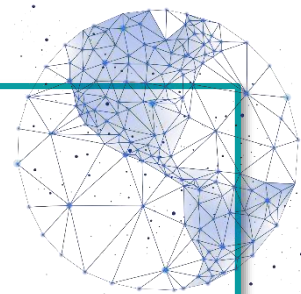
删除数据与删除基本表的区别

delete与drop

3.5 数据更新



插入数据



3.5 数据更新

3.5.1 插入数据



3.4.1 插入数据

三种插入数据方式

插入单个元组

插入多个元组

插入子查询结果

3.5 数据更新

3.5.1 插入数据



1. 插入单个元组

语句格式

```
insert | replace [into] 表名[(列名,...)]  
values({常量|表达式|default},...), (...),...
```

3.5 数据更新

3.5.1 插入数据



1. 插入单个元组

into子句：指定要插入数据的表名及属性列名。

- **全部**指定属性列名：属性列的顺序可与表定义中的顺序**不一致**。
- **没有**指定任何属性列：表示要插入的是一条**完整的元组**，且插入数据顺序与表定义中属性列的顺序**一致**。
- 指定**部分**属性列：插入的元组在**其余属性列**上**取空值**。

values子句：提供的值必须与into子句匹配。

- 值的个数
- 值的类型

3.5 数据更新

3.5.1 插入数据

1. 插入单个元组

例：将下面新的学生记录插入到S表中：

(04050, 陈冬, 男, 信息, 1997/10/01)

字段	索引	外键	触发器
名		类型	
sno		char	
sname		char	
ssex		char	
sbirth		date	
▶scollege		varchar	

3.5 数据更新

3.5.1 插入数据



1. 插入单个元组

例：将下面新的学生记录插入到S表中：

(04050, 陈冬, 男, 信息, 1997/10/01)

```
insert  
into s  
values ('04050', '陈冬', '男', '1997/10/01', '信息');
```

字段	索引	外键	触发器
名			类型
sno			char
sname			char
ssex			char
sbirth			date
▶scollege			varchar

3.5 数据更新

3.5.1 插入数据



1. 插入单个元组

例：将下面新的学生记录插入到S表中：

(04050, 陈冬, 男, 信息, 1997/10/01)

```
insert  
into s(sno,scollege,sname,ssex,sbirth)  
values ('04050','信息','陈冬','男','1997/10/01');
```

字段	索引	外键	触发器
名		类型	
sno		char	
sname		char	
ssex		char	
sbirth		date	
▶scollege		varchar	

3.5 数据更新

3.5.1 插入数据

1. 插入单个元组

例：插入一条选课记录('04020', '001')。

字段	索引	外键
名		
▶ sno		c
cno		c
grade		

```
insert  
into sc(sno,cno)  
values ('04020','001');
```

■此题要考虑**参照完整性规则**，所插入的sno、cno值必须来自s.sno和c.cno，否则会插入失败。

3.5 数据更新

3.5.1 插入数据

2. 插入多个元组

❖ 使用insert into | replace语句添加数据——说明

- 使用insert语句可以向表中插入一行数据，也可以插入多行数据，最好一次插入多行数据，各行数据之间用“，”分隔。

```
insert into c values  
( '024', '计算机导论', '002', 3),  
( '025', '计算机基础', '002', 1.5);
```

3.5 数据更新

3.5.1 插入数据

2. 插入多个元组

❖ 使用insert into | replace语句添加数据——说明

- replace into向表中插入数据时，首先尝试插入数据到表中，如果发现表中已经有此行数据（根据主键或者唯一索引判断），则先删除此行数据，然后插入新的数据，否则，直接插入新数据。

```
replace into c values  
( '024', '计算机导论', '002', 3 ),  
( '025', '计算机基础', '002', 1.5 );
```

3.5 数据更新

3.5.1 插入数据



3. 插入子查询结果

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

3.5 数据更新

3.5.1 插入数据



3. 插入子查询结果

- 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,
值的个数和值的类型都一致

3.5 数据更新

3.5.1 插入数据

3. 插入子查询结果

例：对每一个学院，求学生的平均年龄，并把结果存入数据库。

第一步：建表

```
create table college_avg_age  
(college varchar(40),  
avgage smallint);
```

3.5 数据更新

3.5.1 插入数据



3. 插入子查询结果

例：对每一个学院，求学生的平均年龄，并把结果存入数据库。

college	avgage
人文	28
管理	30
统计与信息	31
信息	26

第二步：插入数据

```
insert  
into college_avg_age(college,avgage)  
select scollege,avg(year(CURDATE())-year(sbirth))  
from s  
group by scollege;
```

要先学好select
from where

3.5 数据更新

3.5.1 插入数据

特别注意： DBMS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性规则。

- **实体完整性**

主码不能重复

- **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



- 1 插入数据
- 2 修改数据
- 3 删除数据

3.5 数据更新

3.5.2 修改数据



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

3.5 数据更新

3.5.2 修改数据

SET子句

1. 指定修改方式
2. 要修改的列
3. 修改后取值

WHERE子句

1. 指定要修改的元组
2. 缺省表示要修改表中的所有元组

• 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

3.5 数据更新

3.5.2 修改数据

例： 将管理学院所有学生的成绩置100。

sno	sname	ssex	sbirth	scollege	smajor
04001	董顺	男	1995-09-29	人文	英语
04002	覃祥	男	1995-08-23	人文	英语
04003	陈旭	男	1994-12-12	人文	英语
04004	王钦	男	1995-05-04	人文	动画
04005	韩振兴	男	1995-04-04	人文	动画
04006	田野	男	1995-09-01	人文	法学
04007	赵永亮	男	1994-07-23	人文	法学
04008	侯玉珏	男	1995-05-15	管理	财务管理
04009	宋传奎	男	1995-04-16	管理	财务管理
04010	白晓鹏	男	1994-03-08	管理	财务管理
04011	刘阳阳	男	1995-02-01	管理	财务管理
04012	冯帅	男	1994-05-09	管理	财务管理
04013	马慧刚	男	1992-07-07	管理	会计学
04014	李阳	男	1991-01-15	管理	会计学
04015	刘锦运	男	1990-10-14	管理	会计学

sno	cno	grade
04001	002	81
04001	004	87
04001	005	99
04001	007	74
04002	003	94
04002	004	93
04002	005	98
04002	006	90
04003	002	96
04003	003	55
04003	004	87
04003	005	75
04003	006	88
04003	007	85
04003	008	96

3.5 数据更新

3.5.2 修改数据

例： 将管理学院所有学生的成绩置100。

```
update sc
set grade=100
where sno in
(select sno
from s
where scollege='管理');
```


3.5 数据更新

3.5.2 修改数据

例： 将管理学院所有学生的成绩置100。

```
update sc,s  
set grade=100  
where s.sno=sc.sno and scollege='管理';
```

这样写对吗？

Mysql 中可以

3.5 数据更新

3.5.2 修改数据

特别注意： DBMS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

- **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



- 1 插入数据
- 2 修改数据
- 3 删除数据

3.5 数据更新

3.5.3 删除数据

语句格式

```
DELETE    FROM    <表名>  
[WHERE <条件>]  
[order by子句]  
[limit row_count] ;
```

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全部元组，表的定义仍在字典中。

3.5 数据更新

3.5.3 删除数据

例：删除学号为04009的学生记录。

```
delete from s  
where sno='04009';
```

这条语句能否顺利执行呢？

3.5 数据更新

3.5.3 删除数据

例： 删除管理学院所有学生的选课记录。

sno	sname	ssex	sbirth	scollege	smajor
04001	董顺	男	1995-09-29	人文	英语
04002	覃祥	男	1995-08-23	人文	英语
04003	陈旭	男	1994-12-12	人文	英语
04004	王钦	男	1995-05-04	人文	动画
04005	韩振兴	男	1995-04-04	人文	动画
04006	田野	男	1995-09-01	人文	法学
04007	赵永亮	男	1994-07-23	人文	法学
04008	侯玉珏	男	1995-05-15	管理	财务管理
04009	宋传奎	男	1995-04-16	管理	财务管理
04010	白晓鹏	男	1994-03-08	管理	财务管理
04011	刘阳阳	男	1995-02-01	管理	财务管理
04012	冯帅	男	1994-05-09	管理	财务管理
04013	马慧刚	男	1992-07-07	管理	会计学
04014	李阳	男	1991-01-15	管理	会计学
04015	刘锦运	男	1990-10-14	管理	会计学

sno	cno	grade
04001	002	81
04001	004	87
04001	005	99
04001	007	74
04002	003	94
04002	004	93
04002	005	98
04002	006	90
04003	002	96
04003	003	55
04003	004	87
04003	005	75
04003	006	88
04003	007	85
04003	008	96

3.5 数据更新

3.5.3 删除数据

例：

删除管理学院所有学生的选课记录。

```
delete from sc
where sno in
(select sno
  from s
 where scollege='管理');
```

3.5 数据更新

3.5.3 删除数据

例： 删除SC表中成绩最低的2行记录。

```
delete from sc  
order by grade  
limit 2;
```


~~→→~~

-

练一练



2、设有一个SPJ数据库，包括S，P，J，SPJ四个关系模式：

S(SNO, SNAME, STATUS, CITY);

P(PNO, PNAME, COLOR, WEIGHT);

J(JNO, JNAME, CITY);

SPJ(SNO, PNO, JNO, QTY);

- 供应商表S由供应商代码（SNO）、供应商姓名（SNAME）、供应商状态（STATUS）、供应商所在城市（CITY）组成；
- 零件表P由零件代码（PNO）、零件名（PNAME）、颜色（COLOR）、重量（WEIGHT）组成；
- 工程项目表J由工程项目代码（JNO）、工程项目名（JNAME）、工程项目所在城市（CITY）组成；
- 供应情况表SPJ由供应商代码（SNO）、零件代码（PNO）、工程项目代码（JNO）、供应数量（QTY）组成，表示某供应商 供应某种零件 给某工程项目的数量为QTY。今有若干数据如下：

S 表

SNO	SNAM E	STATU S	CITY
S1	精 益	20	天津
S2		10	
S3	盛 锡	30	北京
S4		20	
S5	东方红	30	北京
	丰泰盛		天津
	为 民		上海

P 表

PNO	PNAM E	COLO R	WEIG HT
P1	螺 母	红	12
P2	螺 栓	绿	17
P3	螺丝刀	蓝	14
P4	螺丝刀	红	14
P5	凸 轮	蓝	40
P6	齿 轮	红	30

J 表

JNO	JNAME	CITY
J1	三 建	北京
J2	一 汽	长春
J3	弹 簧 厂	天津
J4	造 船 厂	天津
J5	机 车 厂	唐山
J6	无线电厂	常州
J7	半导体厂	南京

SPJ 表

SNO	PNO	JNO	QTY
S1	P1	J1	200
S1	P1	J3	100
S1	P1	J4	700
S1	P2	J2	100
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J4	500
S2	P3	J5	400
S2	P5	J1	400
S2	P5	J2	100
S3	P1	J1	200
S3	P3	J1	200
S4	P5	J1	100
S4	P6	J3	300
S4	P6	J4	200
S5	P2	J4	100
S5	P3	J1	200
S5	P6	J2	200
S5	P6	J4	500

练一练



要求:

2、试用SQL语句完成以下操作

- (8) 把全部红色零件的颜色改成蓝色。**
- (9) 由S5供给J4的零件P6改为由S3供应，请作必要的修改。**
- (10) 从供应商关系中删除S2的记录，并从供应情况关系中删除相应的记录。**
- (11) 请将 (S2, J6, P4, 200) 插入供应情况关系。**

练一练



要求:

2、试用SQL语句完成以下操作

(8) 把全部红色零件的颜色改成蓝色。

```
update P set color=' 蓝' where color=' 红' ;
```

(9) 由S5供给J4的零件P6改为由S3供应，请作必要的修改。

```
update SPJ set sno=' s3' where sno=' s5' and jno=' j4'  
and pno=' p6' ;
```

(10) 从供应商关系中删除S2的记录，并从供应情况关系中删除相应的记录。


```
delete from SPJ where sno=' s2' ;
```

```
delete from S where sno=' s2' ;    要注意删除的顺序
```

(11) 请将 (S2, J6, P4, 200) 插入供应情况关系。

```
insert into SPJ (sno,jno,pno,qty) values(s2,j6,p4,200);
```

```
或 insert into SPJ values(s2,p4,j6,200);
```

The background is a solid blue color with abstract geometric shapes. A large, light blue arrow points from the left towards the right, containing the text. There are also some darker blue lines and shapes, including a circuit-like line on the right side.

谢谢!

T h a n k s