第5章 使用 DML 语言操作数据表

所谓 DML (Data Manipulation Language)语言就是数据操纵语言,主要包括对数据表数据的添加、删除、修改以及查询的操作。对于查询操作也可被称作数据查询语言(DQL)。对数据表的操作是数据库的核心内容,在所有现有的使用数据库的软件中都会用到对数据表数据的操作。

本章的主要知识点如下:

- □ 向数据表中添加数据;
- □ 修改数据表中的数据;
- □ 删除数据表中的数据;
- □ 数据表操作实例。

5.1 使用 INSERT 语句向数据表中添加数据

向数据表中添加数据,记住 INSERT 这个关键字就可以了。在实际的应用中,在网站 上注册用户名、注册邮箱等操作都是对数据表中的数据进行添加操作。在本节中将讲解插 入一条完整的记录、给指定的字段插入记录、复制其他表中的记录、将查询结果插入到表 中等操作。

5.1.1 INSERT 语句的基本语法形式

不论向数据表插入什么样的数据,都要使用 INSERT 关键字来完成。INSERT 语句分为如下 3 种语法格式。

1. INSERT...VALUES形式

INSERT...VALUES 形式用于不指定列直接向数据表中添加数据或者是向指定列中添加数据。具体的语法格式如下所示:

```
INSERT [INTO] table_name[(col_name,...)]
VALUES (col_value1,col_value2,...)
```

其中:

□ INSERT [INTO]: [INTO]是可以省略的,但是一般情况下是加上的。

□ table_name: 要插入数据的表名。

□ col_name: 要插入数据的列名。如果想向表中所有的字段插入值就可以省略列名。

省略列名后插入数据时就要按表中列的顺序插入值。

- □ col_value1:要插入指定列的值。这里,需要注意的是列的个数一定要与插入值的 个数一致,并且数据类型也要兼容。
- 2. INSERT...SET形式

使用 INSERT...SET 形式通常都是通过 SET 对指定字段插入值。具体的语法格式如下 所示:

```
INSERT
 [INTO] table_name
 SET col name= col value, ...
```

其中:

□ table_name: 是数据表的名称。

□ col_name: 是数据表中列的名称。

□ col value: 是数据表中列的值。

这里的[INTO]关键字也是可以省略的。在 SET 后面分别给表中列的设置值,每个列之间用逗号隔开。

3. INSERT...SELECT形式

使用 INSERT...SELECT 可以快速地从一个或多个表中向一个表中插入多个行。这种 插入的方式也被称为数据的复制。在本章的 5.1.3 小节中将给出详细的例子。INSERT... SELECT 的具体语法格式如下:

INSERT [INTO] table_name SELECT ...

这里,SELECT 是查询语句,用来将其他表的查询结果填充到数据表中。但是,将查询出的数据添加到表中,数据也要与表中字段的数据类型和个数匹配。其他的关键字在前面的语法中已经解释,这里就不再赘述。

通过上面给出的3种语法形式,就可以很容易完成数据的添加操作。

5.1.2 给表中指定字段添加数据

给表中指定字段添加数据,可以采用 INSERT 语句的 INSERT...VALUES 形式和 INSERT...SET 形式来完成。下面就分别举例来讲解每种形式语句的使用方法。

【示例 1】 创建一张账目信息表,记录每天的花销情况,表结构如表 5.1 所示。并按下列要求完成数据的添加:

(1)分别使用 INSERT...VALUES 形式和 INSERT...SET 形式向表中添加一条完整的记录。

(2)分别使用 INSERT...VALUES 形式和 INSERT...SET 形式向表中账目编号、消费内容、消费金额字段添加数据。

• 67 •

| 编号 | 列 名 | 数 据 类 型 | 中文释义 |
|----|--------------|--------------|------|
| 1 | id | int | 账目编号 |
| 2 | name | varhcar(20) | 消费内容 |
| 3 | account | decimal(7,2) | 消费金额 |
| 4 | account type | varchar(20) | 消费类型 |
| 5 | accountdate | varchar(20) | 消费时间 |
| 6 | Remark | varchar(200) | 备注 |

表 5.1 账目信息表 (AccountInfo)

创建该数据表的语句如下所示:

CREATE TABLE AccountInfo
(
 id INT PRIMARY KEY,
 name VARCHAR(50),
 account decimal(7,2),
 accounttype VARCHAR(20),
 accountdate VARCHAR(20),
 remarks VARCHAR(200)
);

可以将该数据表建立在 mysql 数据库中。

(1) 向表中添加的2条数据记录如表 5.2 所示。

表 5.2 向账目信息表添加的数据

| 编号 | id | name | account | accounttype | accountdate | remarks | |
|----|----|------|---------|-------------|-------------|---------|--|
| 1 | 1 | 购书 | 50 | 购物 | 20120201 | 2 本书 | |
| 2 | 2 | 吃饭 | 120 | 餐饮 | 20120501 | 吃火锅 | |

下面首先使用 INSERT...VALUES 格式向数据表中添加记录如下:

```
INSERT INTO accountinfo
VALUES(1,'购书',50,'购物','20120201','2本书');
```

执行添加账目的信息,效果如图 5.1 所示。

| Hy | me BySQL 5.5 Command Line Client | | | | | | | | |
|------|---|--------|-------------|------------------|---------------|---------|---|--|--|
| n: | mysql> INSERT INTO accountinfo | | | | | | | | |
| Q | -> UHLUES(1, %小力, 50, %小水, 20120201, 224-寸, 2; Query OK, 1 row affected (0.03 sec) | | | | | | | | |
| L | | N OFFE | T w PROM | | | | | | |
| - mg | ysd1 | + | 1 * FRUM at | ;countinfo; + | + | ++ | | | |
| E | id | l name | l account | accounttype | l accountdate | remarks | | | |
| | 1 | :购书 | : 50.00 | | 20120201 | 2本书 | | | |
| 1 | row | in set | (0.03 sec) | + > | + | + | + | | |
| | Insu | > | | | | | | | |
| Ľ | , | | | | | | • | | |

图 5.1 使用 INSERT...VALUES 插入整行记录

在图 5.1 中通过 SELECT 语句查看了添加的数据。这里,需要注意的是插入数据的顺序是按照数据表中字段的顺序添加的。如果不想按顺序添加数据,可以在表名后面指定添加的顺序。

• 68 •

再使用 INSERT...SET 形式添加表 5.2 中的第2条记录,语句如下所示:

```
INSERT INTO accountinfo
SET id=2,name='吃饭',account=120,accounttype='餐饮',
remarks='吃火锅',accountdate='20120501';
```

执行效果如图 5.2 所示。

| mysgl> INSERT INTO accountinfo -> SET id=2,name="吃饭",account=120,accounttype="餐饮",remarks="吃火锅",acco untdate="20120501"; | | | | | | | | | |
|--|------------|-----------|-------------------|--------------------------|------------------------------|--------------------------------|---------------------|-------------|---|
| նա այ | ery sql | , ' L> | OK, 1 1 SELECI | ∾ow affect [¥ FROM a | ed (0.05 sec) ccountinfo; | | | | 1 |
| | id | 1 | name | account | + accounttype | accountdate | + remarks | -+ | |
| + ! | 1 | - | 购书 吃饭 | 50.00 120.00 | ▪--------- ▪ 购物 ▪ 餐饮 | -+ 20120201 20120501 | ↓ ↓ 2本书 ↓ 吃火锅 | -+ | |

图 5.2 使用 INSERT...SET 插入整行记录

在图 5.2 的执行效果中可以看出,使用 INSERT...SET 语句添加数据时,在 SET 后面可以不按表中字段的顺序添加数据,这样就可以避免由于记不住表中字段的顺序而添加错误的情况。

(2) 向指定的字段添加如表 5.3 所示的值。

表 5.3 向账目信息表添加的数据

| 编号 | id | name | account |
|----|----|------|---------|
| 1 | 3 | 看电影 | 50 |
| 2 | 4 | 爬山 | 210 |

使用 INSERT...VALUES 形式向数据表中添加表 5.3 的第1条记录,语句如下所示:

```
INSERT INTO accoutinfo(id,name,account)
VALUES(3,'看电影',50);
```

执行效果如图 5.3 所示。

| my MySQL 5.5 Command Line Client | | | | | | | |
|--|---|------------------------|------------------------------|--------------------|-----------------|---|--|
| mysql〉INSERT INTO accountinfo(id,name,account) -> VALUES(3,'看电影',50); Query OK, 1 row affected (0.44 sec) mysgl> select * from accountinfo; | | | | | | | |
| ++ | + | + | ۰، | · | + | | |
| id na | me l'account | accounttype | accountdate | remarks | 1 | | |
| 1 购 2 吃 3 看 | 书 ¦ 50.00 饭 ¦ 120.00 电影 ¦ 50.00 | : 购物 : 餐饮 : NULL | 20120201 20120501 NULL | 2本书 吃火锅 NULL | - | | |
| ++ 3 rows in mysql> _ | set (0.00 sec) | • | | | + | T | |

图 5.3 使用 INSERT...VALUES 向指定列插入值

在上一章中的约束里已经讲过设置主键约束的列和非空约束的列是不允许为空的,因

• 69 •

此对于账目信息表来说,账目编号列是主键列是必须要添加的,否则就会出现错误。此外, 从图 5.3 的添加结果可以看出,没有添加的列全部都是 NULL 而不是空值。

再使用 INSERT...SET 语句向账目信息表中插入表 5.3 中的第 2 条记录,语句如下 所示:

```
INSERT INTO accountinfo
SET id=4,name='爬山',account=210;
```

效果如图 5.4 所示。

| mysql> −> Query mysql> | L 5.5 Com INSERT SET id= OK, 1 roo SELECT | and Line Cli INTO accour 4,name='爬 w affected * FROM acco | ient htinfo ',account=21 (0.02 sec) puntinfo; | 0; | | | |
|------------------------------------|---|---|---|---|------------------------------|-----------------|---|
| ++ id ++ 1 2 | name 购书 吃饭 | account 50.00 120.00 | accounttype 购物 餐饮 | + accountdate + 20120201 20120501 | remarks 2本书 吃火锅 | + + | |
| 3 4 ++ 4 rows | 有电影 爬山 in set | 50.00 210.00 + | NULL NULL | : NULL : NULL + | NULL NULL | 1 + | |
| mysql> | | | | | | | T |

图 5.4 使用 INSERT...SET 向表中指定列添加数据

从图 5.4 的结果中可以看出,使用 INSERT...VALUES 和 INSERT...SET 可以完成同样的操作。但是从书写方式来看,使用 INSERT...SET 的方式更不容易出现错误也更方便一些。

5.1.3 复制其他表中的数据

所谓复制其他表中的数据,就是把一个表中的数据直接添加到另一个表中,并且添加 的数据可以是表中的全部数据,这样就能够减少添加数据的工作量。这种复制表中数据的 做法,可以理解为文件夹内容的复制粘贴操作。

下面使用示例 2 来完成表数据的复制操作,表数据的复制操作使用的是 5.1.1 小节中 INSERT...SELECT 语句来完成。

【示例2】 将表 5.4 中的数据复制到账目信息表(accountinfo)中。

| 编号 | 列名 | 数 据 类 型 | 中文释义 | | | | | |
|----|-------------|--------------|------|--|--|--|--|--|
| 1 | accountid | int | 账目编号 | | | | | |
| 2 | accountname | varhcar(20) | 消费内容 | | | | | |
| 3 | account | decimal(7,2) | 消费金额 | | | | | |

表 5.4 accountinfo1

创建该数据表的语句如下所示:

CREATE TABLE accountInfol

(id INT PRIMARY KEY,

```
accountname VARCHAR(50),
account decimal(7,2)
);
```

并向该表中添加如下5条数据,如表5.5所示。

表 5.5 向accountinfo1 中添加的数据

| 序号 | 编号 | 内容 | 金额 |
|----|----|------|-----|
| 1 | 11 | 练瑜伽 | 100 |
| 2 | 12 | 订蛋糕 | 367 |
| 3 | 13 | 买裙子 | 320 |
| 4 | 14 | 去游乐场 | 200 |
| 5 | 15 | 雇钟点工 | 100 |

添加的语句如下所示:

INSERT INTO accountinfol VALUES(11,'练瑜伽',100); INSERT INTO accountinfol VALUES(12,'订蛋糕',367); INSERT INTO accountinfol VALUES(13,'买裙子',320); INSERT INTO accountinfol VALUES(14,'去游乐场',200); INSERT INTO accountinfol VALUES(15,'雇钟点工',100);

现要将表 5.5 中的数据全部复制到表 accountinfo 中,具体语句如下所示:

```
INSERT INTO accountinfo(id,name,account)
SELECT * FROM accountinfol
```

执行效果如图 5.5 所示。

| ∎s ∎yS | QI | L 5.5 Comman | d | Line Clie | n I | 1 | | | | | | 1× |
|------------------------------|---|--------------|-----|-----------|-----|-------------|---|-------------|---|---------|---|----|
| mysq] - Query Recor | mysql> INSERI INTO accountinfo(id,name,account) -> SELECI * FROM accountinfo1; Query OK, 5 rows affected (0.03 sec) Recorde: 5 Dualicates: 0 Amanians: 0 | | | | | | | | | | | |
| mysq] | > | SELECT * 1 | FF | ROM accou | IN | tinfo; | | | | | | |
| i id | ļ | пате | 1 | account | ļ | accounttype | 1 | accountdate | 1 | remarks | 1 | |
| 1 | 1 | 购书 | 1 | 50.00 | ł | 购物 | 1 | 20120201 | 1 | 2本书 | 1 | |
| 1 2 | ł | 吃饭 | 1 | 120.00 | ł | 餐饮 | 1 | 20120501 | 1 | 吃火锅 | 1 | |
| 1 3 | ł | 有电影 | 1 | 50.00 | 1 | NULL | 4 | NULL | 4 | NULL | 1 | |
| i 4. | 1 | 城瑜伽 | 1 | 100.00 | 1 | NULL | 1 | NULL | 1 | NULL | | |
| 1 12 | i | 江星桜 | i | 367 00 | ÷ | NULL | ÷ | NULL | ÷ | NULL | 1 | |
| 13 | i | 实裙子 | i. | 320.00 | ÷ | NULL | ÷ | NULL | ÷ | NULL | i | |
| 14 | i | 去游东场 | i. | 200.00 | i | NULL | i | NULL | i | NULL | 1 | |
| 15 | ł | 雇钟点工 | I. | 100.00 | ł | NULL | ÷ | NULL | ÷ | NULL | ł | |
| + | + | | +- | | + | | + | | + | | + | |
| 9 PO% | s | in set (Ø | . e | NO Sec) | | | | | | | | |
| mysa] | > | _ | | | | | | | | | | |
| | | - | | | | | | | | | | |

图 5.5 复制 accountinfo1 中的数据到 accountinfo 中

从图 5.5 的复制结果可以看出,从一个表中将数据复制到另一个表中,要注意复制的列是否是同一类型并且个数也要一致。accountinfo1 中有 3 列,就需要在复制时指定 accountinfo 的 3 列来接收值。

• 71 •

5.1.4 为表添加多条数据

向表中一次插入多条记录是实际操作中经常会用到的,就像在示例 2 中为表 accountinfo1 添加 5 条记录一样,读者会发现添加语句都是重复的,只是添加的值有所改变。 那么,为了简化添加时的语句,使用 INSERT...VALUES 添加多条记录的语法如下所示:

INSERT INTO table_name
VALUES(value_list1)...(value_listn);

下面就使用示例3来演示如何向表中一次插入多条记录。

【示例3】 向表 accountinfo1 中一次添加如表 5.6 所示的记录。

| 序号 | 编号 | 内容 | 金额 |
|----|----|------|------|
| 1 | 21 | 买报纸 | 5 |
| 2 | 22 | 交电费 | 200 |
| 3 | 23 | 交物业费 | 1200 |

表 5.6 向表中添加的记录

将表 5.6 所示的记录添加到 accountinfol 中的语句如下所示:

```
INSERT INTO accountinfo1
VALUES(21,'买报纸',5),(22,'交电费',200),(23,'交物业费',1200);
```

执行效果如图 5.6 所示。

| ∎s. ∎ySQ | L 5.5 Command L | ine Client | | _ 🗆 × |
|---|---|---|---|-------|
| mysq1) -) Query Record mysq1) | → INSERT INTO → UALUES(21,/ OK, 3 rows af: ls: 3 Duplica → SELECT * FRO | accountinf 行扱紙',5), fected (0. tes: 0 Wa M accounti | o1 <22,'交电费',200>,<23,'交物业费',1200>; 95 sec> rnings: 0 nfo1; | |
| + | accountname | + account | + | |
| 1 11 | | + 1 100.00 1 367.00 | * | |
| 13 14 15 | 去游乐场 | ; 320.00 ; 200.00 ; 100.00 | | |
| 21 22 | : 买报纸 : 交电费 - 充物业费 | 5.00 200.00 | 1 | |
| ++ 8 rows | · 入10)正页 | + sec) | * | |
| mysql) | • | | | - |

图 5.6 一次插入多条记录

从图 5.6 中可以看出,通过上面的语句就可以将 3 条记录一次全部插入到数据表中。

5.2 使用 UPDATE 语句修改表中的数据

数据表中的数据并不是一次添加后就不再更改的,修改数据表中的数据使用 UPDATE 关键字开始的语句完成。在现实生活中,哪些对数据库的操作会使用修改呢? 当你要修改

• 72 •

邮箱的密码时,当你修改注册的用户名时,实际上都是对数据表的修改操作。在本节中将 学习如何来修改数据表中的数据。

5.2.1 UPDATE 语句的基本语法形式

UPDATE 语句的语法形式没有 INSERT 的语法形式多,对于单表的修改只有一种形式, 具体语法如下所示:

```
UPDATE table_name
SET col_name1=value1,col_name2=value2,...
[WHERE 条件]
[ORDER BY ...]
[LIMIT row count]
```

其中:

- □ table_name: 表名。
- □ col_name: 列名。
- □ [WHERE 条件]: 该语句是可选的,代表修改数据时的条件。如果不选择该语句, 代表的是修改表中的全部数据。
- □ [ORDER BY ...]: 该语句是可选的,代表的是修改数据的顺序。
- □ [LIMIT row_count]: 该语句是可选的,用来限制可以被更新的行的数目。

➡注意:如果是多个表连接来修改数据,不能够使用 ORDER BY 和 LIMIT 子句。

5.2.2 修改表中的全部数据

修改表中的全部数据是一种不太常用的操作,但是有的时候也会用到,比如:当需要 将所有的商品价格增长10%时,就需要修改商品信息表中的所有商品价格。修改全部数据 的语句是不带任何可选项的语句,具体的语句如下所示:

```
UPDATE table_name
```

SET col_name1=value1,col_name2=value2,...

下面就通过示例4来演示如何修改表中的全部数据。

【示例 4】 修改账目信息表中的备注字段 remarks,将其全部修改成"消费"。 根据 UPDATE...SET 的语法形式,具体的修改语句如下所示:

```
UPDATE accountinfo
SET remarks='消费';
```

执行效果如图 5.7 所示。

从图 5.7 中显示的修改结果可以看出,所有的 remarks 值全部修改成了"消费"。

5.2.3 根据条件修改表中的数据

根据条件修改表中的数据是一种比较常用的操作。比如:对于商品信息表中的商品调

• 73 •

价的问题,要将某一类商品的价格上涨或者下降,都是要将商品类型作为条件进行调价的。 根据条件修改表中的数据,要使用的是 UPDATE...SET...WHERE...的语句来完成的。下 面通过示例 5 来演示如何根据条件来修改表中的数据。

| WSUL 5.5 Comman mysql> UPDATE act -> SET remark Query OK, 9 rows Rows matched: 9 mysql> SELECT * 1 | <mark>d Line Clien</mark> countinfo ks='消费'; affected (Changed: S FROM accour | t 10.47 sec) 9 Warnings: 0 1tinfo; | | | | - |
|--|---|--|--|---------------------------------------|--------------------------------------|---|
| id name | account | accounttype | accountdate | ¦ remarks | i i | |
| 1 : 1: : 购吃 2 : 6 3 : 7 4 : 6 11 : 5 12 : 7 13 : 7 13 : 7 14 : 5 15 : 16 | 50.00 120.00 50.00 100.00 367.00 320.00 100.00 | 购物 帮饮 NULL NULL NULL NULL NULL NULL | 20120201 20120501 NULL NULL NULL NULL NULL NULL NULL | · · · · · · · · · · · · · · · · · · · | • • • • • • • • | |
| 9 rows in set (Ø mysql) | .00 sec) | | | | | |

图 5.7 修改 accountinfo 中的 remarks 字段

【示例 5】 将账目信息表(accountinfo)中所有的餐饮类项目的备注(remarks)内容 修改为"吃大餐"。

在账目信息表中要修改餐饮类项目的值,实际上就是要修改消费类型(accounttype) 字段是餐饮的数据值。具体的语句如下所示:

```
UPDATE accountinfo
SET remarks='吃大餐'
WHERE accounttype='餐饮';
```

执行效果如图 5.8 所示。

| 🔤 BySQL 5.5 Command Li | ne Client | | | |
|---|--|-------------|---------|--------|
| mysql> UPDATE accoun -> SET remarks=' -> WHERE account Query OK, 1 row affe Rows matched: 1 Cha | tinfo 吃大餐, type " 餐饮 ; icted <0.05 sec> inged: 1 Warnings: | Ø | | * |
| mysql> SELECT * FROM | accountinfo; | | | _ |
| lid I name I ac | count accounttype | accountdate | renarks | • |
| 111 购书 1 | 50.00 : 购物 | 20120201 | 消费 | • ! |
| 2 吃饭」 1 | 20.00 餐饮 | 20120501 | 吃太餐 | ! |
| | 50.00 i NULL | I NULL | 伯贺 | |
| | .00.00 NULL | I NULL | 温器 | |
| 12 订蛋糕 3 | 67.00 NULL | INULL | 消费 | i |
| 13 买裙子 3 | 20.00 NULL | I NULL | 逍嚢 | 1 |
| 14 丟游乐场 2 | 00.00 NULL | I NULL | 逍靈 | |
| 15 准计点上 1 | .00.00 ! NOTT | 1 NULL | 伯贷 | • |
| 9 rows in set (0.02 | sec) | | | |
| nysq1> | | | | - |

图 5.8 根据条件修改 remarks 字段

在图 5.8 中可以看到,在没有修改之前,所有的 remarks 字段的值全部都是"消费",执行了上面的语句后,已经将消费类型是餐饮的记录中的 remarks 的字段值修改成了"吃大餐"。

5.2.4 根据顺序修改数据表中的数据

在修改数据表中的数据时,也可以指定修改数据的顺序,但是这种方法也不太常用。

• 74 •

如果要指定修改的顺序,要在 UPDATE...SET 语句中加上可选子句 ORDER BY 来完成。 具体的语法格式分为如下两种。

(1) 有 WHERE 条件的排序修改

如果要加上修改的条件,就要把 ORDER BY 子句放在 WHERE 子句之后。具体语法 如下所示:

```
UPDATE table_name
SET col_name1=value1,col_name2=value2,...
WHERE 条件
ORDER BY 列名 1 DESC/ASC, 列名 2 DESC/ASC...
```

这里, ORDER BY 后面可以按多个列进行排序。DESC 是降序排列, ASC 是升序排列, 如果不加排序的方式, 默认的是升序排列。

(2) 没有 WHERE 条件的排序修改

没有 WHERE 条件的排序修改就比较容易了,直接在 SET 后面加上 ORDER BY 就可以了。具体的语法如下所示:

```
UPDATE table_name
SET col_name1=value1,col_name2=value2,...
ORDER BY 列名1 DESC/ASC,列名2 DESC/ASC...
```

下面就通过示例 6 来演示如何根据顺序来修改数据。

【示例6】 对账目信息表(accountinfo)中的消费类型做如下修改:

(1) 按照消费金额从高到低的顺序来修改消费类型,将消费类型全部修改成"购物"。

(2) 按照消费金额从高到低的顺序来修改消费类型,将消费金额大于 200 的消费类型 修改成"购大件商品"。

根据按顺序修改数据的语法规则,这两个题的答案如下:

(1) 属于无条件的按顺序修改的方式,具体的语句如下所示:

UPDATE accountinfo SET accounttype='购物' ORDER BY account desc

执行效果如图 5.9 所示。

| Bo, ByS | QL 5.5 Comman | d Line Clie | nt | | | | . 🗆 🗙 |
|--|--|---|---|--|--------------------------------------|---------|---------|
| mysql Query Rows mysql | > UPDATE ac OK, 8 rows matched: 9 | countinfo affected Changed: FROM accou | SET accounttyr (0.05 sec) 8 Warnings: Ø ntinfo; | pe='购物'ORDER | BY account | t desc; | |
| l id | l name | l account | l accounttype | accountdate | l remarks | • | |
| 1 2 3 4 11 12 13 14 15 | ₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩ ₩₩₩₩₩₩₩₩₩₩₩₩ | 50.00 120.00 210.00 100.00 367.00 200.00 100.00 100.00 | · 购物 · 购物物 · 购物物 · 购物物 · 购物物 · 购物物 · 购物物 · 购物物 · 购物物 · 购物物 | : 20120201 : 20120501 : NULL : NULL : NULL : NULL : NULL : NULL : NULL | ₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩ | | |
| , 9 row mysql | us in set (0 | .00 sec) | | | | | ¥ |

图 5.9 无条件的顺序修改

(2) 有条件的按顺序修改的方式,具体语句如下所示。

UPDATE accountinfo SET accounttype=''购大件商品' WHERE account>=200 ORDER BY account desc

执行效果如图 5.10 所示。

| Hys. | yS | QL 5.5 Comma | nd Line Clien | ıt | | | |
|--------------------------|--|---|--|--|---|-----------|----------|
| mys Que Rou mys | sq1 - - ery #s 1 | > UPDATE ac > WHERE acc > ORDER BY > ; OK, 4 rows matched: 4 > SELECT * | countinfo S ount>=200 account des affected (Changed: 4 FROM accour | ET accounttyp cc (0.47 sec) I Warnings: Ø ntinfo; | ≥='购大件商品, | | <u> </u> |
| + : | id | + name | + | accounttype | + accountdate | remarks | + |
| | 1 2 3 4 11 12 13 14 | ▶ | 50.00 120.00 50.00 100.00 367.00 320.00 200.00 100.00 | 购购物 购购物大物 购购为为为的 购购为为大物 并件件 的 购购 物 为 为 为 为 为 为 为 为 为 为 为 为 为 为 为 为 为 | 20120201 20120501 NULL NULL NULL NULL NULL NULL NULL NUL | 消吃消消消消消消消 | * |
| , 9 1 my: | row sql | s in set (@ | .00 sec) | | | | • |

图 5.10 有条件的顺序修改

从图 5.10 中,就可以看出已经将消费金额大于 200 的消费类型,修改成了"购大件商品"。

5.2.5 限制行数的修改

限制修改行数的修改就是能够限制修改一个表中的多少行,比如:修改价格在前5名的商品。就可以先将商品的价格按从高到低的顺序排列,然后限制修改数据表的5行数据就可以了。限制修改行数的可选子句是 LIMIT,它在所有的子句后面来使用。也就说,可以直接放在 UPDATE...SET 之后,也可以放在 UPDATE...SET...WHERE 之后,还可以放在 UPDATE...SET...WHERE...ORDER BY 之后。下面就通过示例7来演示如何修改限制的行数。

【示例7】 根据下面的要求来修改账目信息表 (accountinfo):

(1) 修改账目信息表前2条记录,将其备注修改成"消费1"。

(2) 修改消费金额大于 100 的前 3 条记录,将其备注修改成"消费 2"。

(3)修改消费类型是"购大件商品"的消费金额在前2名的记录,将其备注修改成"消费3"。

根据前面的语法格式分别讲解(1)~(3)题。

(1) 使用 UPDATE...SET...LIMIT 的语法形式来完成,具体语句如下所示:

```
UPDATE accountinfo
SET remarks='消费1'
LIMIT 2
```

• 76 •

执行效果如图 5.11 所示。

| ■ BySQL 5.5 Comman | nd Line Clien | ıt | | | | -OX |
|---|--|---|--|-------------|---|-----|
| mysql> UPDATE ac -> SET remar -> LIMIT 2; Query OK, 2 rows Rows matched: 2 mysql> SELECT * | countinfo ks='消费1' affected (Changed: 2 FROM accour | (0.06 sec) ? Warnings: O htinfo; | | | | |
| ++ | + | accounttype | + accountdate | remarks | + | |
| 1 1 Pp吃了 2 1 Pp吃看 9 0 10 11 15 14 | 50.00 120.00 50.00 100.00 367.00 320.00 200.00 100.00 | 购购物 购购物大物 购购大大大大的 购为人大大大物 购购为大大大物 | 20120201 20120501 NULL NULL NULL NULL NULL NULL NULL NULL | 消消消消消消消消消消消 | • • • • • • • • • | |
| 9 rows in set <0 mysql> _ | .02 sec) | | | | | T |

图 5.11 使用 UPDATE...SET...LIMIT 修改数据

从图 5.11 中可以看出已经将表 accountinfo 中前 2 条记录的 remarks 修改成了"消费 1"。 (2) 使用 UPDATE...SET...WHERE...LIMIT 的语法形式来完成,具体语句如下所示:

UPDATE accountinfo SET remarks='消费2' WHERE account>100 LIMIT 3

执行效果如图 5.12 所示。

| 🔤 BySQL 5.5 Command | l Line Clien | t | | | |
|---------------------|--------------|-------------|---------------|-------|----------|
| mysql> UPDATE acc | ountinfo | | | | <u> </u> |
| -> SET remark | s='消费2' | | | | |
| -> WHERE acco | unt>100 | | | | |
| -> LIMIT 3 | | | | | |
| ->; | | | | | |
| Query OK, 3 rows | affected < | 0.11 sec) | | | |
| Rows matched: 3 | Changed: 3 | Warnings: Ø | ļ — | | |
| 1. 0.51.507 | DOM | | | | |
| mysq1> SELECT * F | ROM accoun | tinfo; | | | |
| id name | account ! | accounttune | ! accountdate | | + |
| tt | + | | 1 accountuate | | + |
| :1:购书 : | 50.00 | 购物 | 20120201 | ; 消费1 | |
| 121吃饭 1 | 120.00 | 购物 | 20120501 | ι 淄慸2 | |
| 3 看电影 | 50.00 ¦ | 购物 | I NULL | ▪ 消费 | 1 |
| 4 : 爬山 : | 210.00 ¦ | 购大件商品 | I NULL | : 消费2 | 1 |
| 11 练瑜伽 | 100.00 ¦ | 购物 | I NULL | : 消费 | 1 |
| 12 订蛋糕 | 367.00 | 购大件商品 | I NULL | : 消费2 | 1 |
| 13 买裙子 | 320.00 ¦ | 购大件商品 | I NULL | : 消费 | : |
| 14 去游乐场 | 200.00 ¦ | 购大件商品 | I NULL | ▪ 消费 | 1 |
| 15 雇钟点工 | 100.00 ¦ | 购物 | I NULL | ▪ 消费 | 1 |
| ++ | + | | + | + | + |
| 9 rows in set (0. | 00 sec) | | | | |
| | | | | | |
| mysql> | | | | | <u> </u> |

图 5.12 使用 UPDATE ... SET ... WHERE ... LIMIT 修改数据

在图 5.12 中可以看出, 该语句修改的是消费金额是 120、210 以及 367 的 3 条记录, 都是按数据表中原有的顺序修改的。

• 77 •

(3) 使用 UPDATE...SET...WHERE...ORDER BY...LIMIT 的语句形式来完成,具体的语句如下所示:

```
UPDATE accountinfo
SET remarks='消费3'
WHERE accounttype='购大件商品'
ORDER BY account DESC
LIMIT 2;
```

执行效果如图 5.13 所示。

| BR BySQL 5.5 Comman | d Line Clier | ıt | | | |
|--|--|---|--|---------------|---------------------------------------|
| mysql> UPDATE ac -> SET remar -> WHERE acc -> ORDER BY -> LIMIT 2; Query OK, 2 rows Rows matched: 2 mysql> SELECT * | countinfo ks='消费3' ounttype='J account DES affected Changed: 2 FROM accour | 购大件商品, SC (0.03 sec) 2 Warnings: 0 ntinfo; | | | |
| ++ | l account | + accounttype | accountdate | remarks | + |
| 1 1: 购吃用 2 : 吃吃 3 : 看爬山 11 : 练强。 11 : 练强。 12 : 订编和 11 : 标理 13 : 买招游标 14 : 去霍钟 15 : 雇 | 50.00 120.00 50.00 1210.00 100.00 367.00 320.00 200.00 100.00 | 购购物物 购购物大件 商品 购购为大件件 商品品 品 的 为 为 为 为 为 为 为 为 为 为 为 为 为 为 为 为 为 | 20120201 20120501 NULL NULL NULL NULL NULL NULL NULL | 消消消消消消消消消消消消消 | * * * * * * * * * * * * * * * * * * * |
| 9 rows in set (0 mysql> | .00 sec) | | • | | • |

图 5.13 使用 UPDATE...SET...WHERE...ORDER BY...LIMIT 修改数据

在图 5.13 中可以看出,该语句修改的是购大件商品类型中价格为 367 和 320 的 2 条价格最高的记录。

5.3 使用 DELETE 语句删除表中的数据

删除数据表中多余的数据也是数据表中必不可少的操作之一。那么,在我们使用软件时什么时候是对数据表进行删除操作呢?比如:在网站进行购物时,要去掉订单中的商品或取消订单的操作都是对数据表的删除操作。在本节中将详细讲解删除语句 DELETE 的各种使用方法。

5.3.1 DELETE 语句的基本语法形式

数据表的删除操作使用的是 DELETE 关键字开头的语句完成的。使用 DELETE 语句可以完成对数据表中全部数据的删除、按条件删除数据以及按顺序删除等操作。具体的语法形式如下所示:

```
DELETE FROM table_name
[WHERE 条件]
[ORDER BY ...]
```

• 78 •

[LIMIT row count]

其中:

- □ table_name: 表名。
- □ [WHERE 条件]: 该语句是可选的,代表删除数据时的条件。如果不选择该语句, 代表的是删除表中的全部数据。
- □ [ORDER BY ...]: 该语句是可选的,代表的是删除数据的顺序。
- □ [LIMIT row_count]: 该语句是可选的,用来限制可以被删除的行的数目。

➡注意:如果是多个表连接来删除数据时,不能够使用 ORDER BY 和 LIMIT 子句。

5.3.2 删除表中的全部数据

删除表中的全部数据是很简单的操作,但也是一个危险的操作。一旦删除了所有记录, 就无法恢复了。因此,在删除操作之前一定要对现有数据进行备份,以避免不必要的麻烦。

删除表中全部数据使用 DELETE FROM...语句就可以完成,下面就使用示例 8 来演示 删除数据的操作。

【示例8】 删除表 accountinfo1 中的全部数据。

删除语句如下所示:

```
DELETE FROM accountinfol;
```

执行效果如图 5.14 所示。从图中可以看出,执行删除语句后,再查询 accountinfo1 表 得到的结果是 Empty set (无数据)。



图 5.14 删除表中的全部数据

除了 DELETE FROM 语句可以删除数据表中的数据之外,还可以通过一个更快的方法 ——使用 TRUNCATE TABLE 来删除数据。与 DELETE FROM 语句不同的是,使用 TRUNCATE TABLE 的方式删除数据,不会返回删除数据行数。下面使用 TRUNCATE TABLE 语句来完成示例 8 的操作,具体的语句如下所示:

TRUNCATE TABLE accountinto1;

由于在示例 8 中已经将其记录全部删除了,因此先为 accountinfol 表添加一条记录, 然后再执行删除操作。执行效果如图 5.15 所示,从中可以看出执行了 TRUNCATE TABLE 后并没有返回删除的行数。但是,在查询数据表 accountinfol 时,已经将其记录删除了。

5.3.3 根据条件删除表中的数据

大多数对数据表的删除操作都是有条件的删除操作,比如:将一段时间内没有使用的

• 79 •

账号删除或者将没有交费的报名信息删除等操作。根据条件删除数据表中的数据使用的是 DELETE FROM ...WHERE 语句来完成的。下面使用示例 9 来演示该语句的使用。

| By BySQL 5.5 Command Line Client | _ 🗆 🗵 |
|---|---------|
| mysql> INSERT INTO accountinfo1 VALUES(1,2,3); Query OK, 1 row affected (0.06 sec) | |
| mysql> SELECT * FROM accountinfo1; | |
| id accountname account | |
| 1 1 2 3.00 | |
| 1 row in set (0.00 sec) | |
| mysql> TRUNCATE TABLE accountinfo1; Query OK, Ø rows affected (0.05 sec) | |
| mysql> SELECT * FROM accountinfo1; Empty set (0.00 sec) | |
| | ▼ ▼ |

图 5.15 使用 TRUNCATE TABLE 删除数据

【示例9】 按下列条件删除账目信息表(accountinfo)中的数据。

(1) 删除编号(id) 是1的账目信息。

(2) 删除消费金额(account) 在 50~100 之间的账目信息。

这两个小题都是很简单的操作,只要将 WHERE 语句后面的条件写对就可以。具体答案如下:

(1) 删除编号是1的账目信息, 是一个单一条件的删除语句。具体语句如下所示:

DELETE FROM accountinfo WHERE id=1

执行效果如图 5.16 所示。

| mm MySQL 5.5 Command Line Client | <u> </u> |
|--|----------|
| nyzql> DELETE FROM accountinfo UHERE id=1; Query OK, 1 row affected (0.03 sec) nysql> SELECT = FROM accountinfo; | * |
| id name account accounttype accountdate remarks | |
| 2 1 P 匹先 1 2 2 8 4 9 0 1 (約) 1 2 9 1 2 8 5 1 0 1 (約) 1 4 4 1 8 5 1 5 0.0 0 1 5 0.0 0 1 5 0.0 0 1 1 0.0 1 1 0.0 1 0.0 0 1 4 1 1 1 5 5 0.0 0 1 5 0.0 0 1 5 0.0 0 1 0.0 1 1 0.0 1 0.0 0 1 1 1 1 5 5 0.0 0 1 1 0.0 0.0 0 1 5 0.0 0 1 5 0.0 0 1 0.0 1 0.0 0 1 2 1 1 1 5 5 0.0 0 1 3 0.7 0.0 1 1 5 0.7 0.0 1 1 5 0.7 0.0 1 1 5 0.7 0.0 1 1 5 0.7 0.0 1 1 0.0 1.0 0 1 3 1 5 7.6 7 1 3 2 0.0 0 1 5 0.7 0.0 1 1 | |
| 8 rows in set (8.80 sec) nysql>_ | - |

图 5.16 删除编号是1的账目信息

从图 5.16 中可以看到, id 是 1 的账目信息已经被删除了。

(2) 消费金额在 50~100 之间, 是一个多条件的删除语句。删除语句如下所示:

DELETE FROM accountinfo WHERE account>=50 and account<=100;

执行效果如图 5.17 所示,从中可以看出已经不存在消费金额在 50~100 之间的数据了。

5.3.4 按指定顺序删除数据

按指定顺序删除数据与修改数据时的语句是类似的,使用的语句是 DELETE FROM ...

• 80 •

[WHERE]...ORDER BY。不管是否有 WHERE 条件语句,都要将 ORDER BY 放在语句的 后面。下面就使用示例 10 来演示如何按指定顺序删除数据。

| 1 | <mark>aa∎y</mark> S | Q | L 5.5 Comman | d Line Clier | ıt | | | | | | _ 🗆 🗵 |
|---|--------------------------|--------|---------------------------------------|--|--|----|--|-----|-------------------------|----|-------|
| | iysql luery iysql | > > | DELETE FR OK, 3 rows SELECT * 1 | DM account: affected ' FROM accour | info WHERE acco (0.05 sec) ntinfo; | •• | unt>=50 and a | ¢ (| count<=106 |); | |
| | id | 1 | name | l account | accounttype | i | accountdate | ļ | remarks | | |
| | 2 4 12 13 14 | | 吃爬山 爬山 玉裙子 去游乐场 | 120.00 210.00 367.00 320.00 200.00 | 购物 购为大件商品 购大件件商品 购大件商品 购大件商品 | | 20120501 NULL NULL NULL NULL NULL | | 消费2 費费3 消费3 消费 | | |
| - | i row nysql | s | in set (Ø | .00 sec) | - | | | - | | | T |

图 5.17 删除消费金额在 50~100 之间的数据

【示例 10】 按下列要求删除账目信息表(accountinfo)的数据:

(1) 按账目编号从大到小删除账目信息表中的数据。

(2) 按账目编号从大到小的顺序, 删除消费金额是 200 的数据。

按顺序删除数据只是在之前的删除语句上加上 ORDER BY 子句,编写语句是很简单的。具体的操作如下:

(1) 无条件删除账目信息表的信息,实际上也是删除了全部信息。具体语句如下所示:

```
DELETE FROM accountinfo
ORDER BY id DESC;
```

执行上面的语句后, accountinfo 表中将没有数据了。为了后面要使用 accountinfo 中的数据,这里就不演示其执行效果,请读者自行测试。

(2) 有条件并按顺序删除账目信息表中的数据,具体语句如下所示:

```
DELETE FROM accountinfo
WHERE account=200
ORDER BY id DESC;
```

执行效果如图 5.18 所示,从中可以看出消费金额是 200 的账目信息已经删除了。

| mysq12 Query | L 5.5 Comm DELETE 1 OK, 1 rot | and Line Cl: FROM accour w affected | ient ntinfo WHERE ad (0.05 sec) | ccount=200 ORD | ER BY id D | ESC; |
|--|-------------------------------------|--|--|---|--|---------------------------|
| Hysq1, + id + 2 4 12 13 | | account account 120.00 210.00 367.00 320.00 | accounttype accounttype 购物 购大件商品 购大件商品 | accountdate 20120501 NULL NULL NULL | ↓ remarks ↓ · · · · · · · · · · · · · · · · · · · | + |
| 4 rows mysql2 | s in set | + (0.00 sec) | • | ŧ | • | • |

图 5.18 有条件的按顺序删除数据

5.3.5 限制行数的删除

在修改数据时可以限制一次修改的行数,在删除操作时也可以同样限制删除的行数。

• 81 •

限制一次删除的行数,可以让删除操作更便捷。比如:当表中有上万条记录时,要删除其中的几千条记录,一次删除这么多行数据可能会造成执行速度慢的问题。那么,就可以在每次删除时限制一下删除记录的行数,多执行几次删除操作就可以完成同样的效果,并提高了每次的删除效率。

限制删除行数也是通过 LIMIT 关键字来限制的, 它放在所有的删除语句之后。下面通过示例 11 来演示如何在删除时限制删除的行数。

【示例11】 按下列要求删除账目信息表(accountinfo)中的数据:

(1) 删除消费金额最高的账目信息。

(2) 删除按账目编号从大到小排序后的前2条账目信息。

使用 LIMIT 关键字限制删除行数时,一定看清题目的要求是删除几条记录。其他的写 法都按照 DELETE 语句的规则写就可以。

(1) 消费金额最高的账目信息,可以通过把数据按照消费金额从高到低排序,然后删除1条数据。具体语句如下所示:

```
DELETE FROM accountinfo
ORDER BY account DESC
LIMIT 1;
```

执行效果如图 5.19 所示,从中可以看出,消费金额是 367 的记录已经被删除了。



图 5.19 删除消费金额最高的数据

(2) 先按编号进行降序排列,再限制删除2条记录。具体语句如下所示:

```
DELETE FROM accountinfo
ORDER BY id DESC
LIMIT 2;
```

执行效果如图 5.20 所示,从中可以看出,已经将编号是4 和 13 的 2 条记录删除了。

| ny: | 991 - - | > DELET > ORDER > LIMIT | E FROM acco BY id DES 2; | ountinfo C | | | - |
|----------|------------|-------------------------------|--------------------------------|----------------|---------------|------------|-------|
| 2u | ery | OK, 2 | rows affect | ted (0.03 sec) | | | |
| ny: + | sq1 id | > SELEC. + name | l * FROM an + account | accounting; | | remarks | + |
| • | 2 | ▪ ▪ 吃饭 | 120.00 | ↓ ↓ 购物 | + 20120501 | + Ⅰ 消费2 | + |

图 5.20 按编号删除 2 条记录

• 82 •

5.4 使用图形界面操作数据表

在前面的3节中已经详细地讲解了如何使用DML语句来对数据表中的数据进行操作。 实际上,对于 MySQL 中的数据表操作也可以借助其他的图形界面来完成。前面讲过的 MySQL WORKBENCH 或者是 SQLyog 都可以完成。SQLyog 的主界面如图 5.21 所示。



图 5.21 SQLyog 的主界面

在本节中使用 SQLyog 来演示如何使用图形界面操作账目信息表。不论对数据表的数据做添加、修改、删除哪一个操作,首先都需要打开要操作的数据表。在图 5.21 所示的界面中右击 accountinfo 表,在图 5.22 中弹出的右键菜单中选择"打开表"选项,显示数据表中的数据如图 5.23 所示。

| aysql - root@localhost | | | | | | | _ 🗆 × |
|---|--------------------|-------------|-----------------|----------------|---------------|---------------|----------------|
| 🖬 root@localhost | 🚳 Query 强 Qur | ery Builder | 🔡 架构设计器 | 1 | | | () × |
| + = information_schema | Autocomplete: [Tal | o]->Next Ta | ig. [Ctrl+Space |]->List Matchi | ng Tags. [C | Ctrl+Enter]-> | List All Tags. |
| mysql Jables | 1 | | | | | | |
| | | • | | | | | |
| 田 columns 自复制表到其它主机。 | /数据库(C) | | | | | | |
| ⊕ db | | | | | | | Þ |
| 田 event 田 打开表 ① □ | F | 11 | | . ata Milan | | (14) | 1 |
| 王 🛄 func 🔍 创建表 (1) | | ler | ♥ 3 信息 ■ | 4 表数据 4 | <u>5</u> Into | ◎ 6 历史 | |
| 🛨 📃 general 📑 更改表 (A) | i | F6 | Y | | 圜. | | |
| 🗄 🛄 help_cs ҧ 管理索引 (M) | ; | F7 | | | -w | | |
| 田 help_ke - 关联/外键(L) | F | 10 | | | | | |
| Theip_re Th | | • | | | | | |
| H host | | | | | | | |
| 田 ndb bin 备份/导出 (B) | | • | | | | | |
| 王 plugin 导入(I) | | • | | | | | |
| + procs_p | | _ | | | | | |
| 🛨 🔳 product 🎋 创建触发器 @) | | | | | | | |
| 🛨 📕 proxies_priv | | | | | | | |
| + servers | | | | | | | |
| E slow_log | | | | | | | |
| + tables priv | | | | | | | |

图 5.22 表操作的右键菜单

下面分别讲解添加数据、修改数据以及删除数据的操作。

| ∎ysql – root@localhost | | _ 🗆 × |
|---------------------------|--|--------------|
| 🖢 root@localhost | A Query 温 Query Builder 器架构设计器 | () × |
| 🗄 📗 information_schema | Autocompletes [Tab] - Next Tag. [Ctels Second - List Matching Tags. [Ctels Enter] - List | t All Tags |
| H 🗐 hotel | Addition piece: [Tab]*/wext Tag, [Correspace]*/List matching Tags, [Correcter]*/List | ic All Tays. |
| E 📗 mysql | | |
| 🖃 📕 Tables | | |
| + accountinfo | | |
| accountinfol | | |
| 📧 🛄 columns_priv | | |
| 🕀 🛄 db | | <u> </u> |
| event | | |
| 🛨 🛄 func | ▲ 1 结末 1 個 2 Profiler ♥ 3 信息 ■ 4 枚数数 ● 5 Info ■ 5 ///文 | |
| 🕀 🛄 general_log | 🗧 🔛 🧮 🦉 🔍 C All Row 📀 Rows in a Bang First 🛛 🗐 💿 🕒 No. of | 50 |
| 🗄 📕 help_category | Row: Rows: | 1 |
| help_keyword | id name account accounttype accountdate remarks | |
| 🗄 🛄 help_relation | 2 吃饭 120.00 购物 20120501 消费2 | |
| help_topic | * OMIT) OMIT) OMIT) OMIT) OMIT) | |
| 🛨 🛄 host | | |
| Indb_binlog_index | | |
| 🛨 🛄 plugin | | |
| procs_priv | | |
| 🗄 📕 productinfo | | |
| 🛨 🔳 proxies_priv | | |
| 🛨 🛄 servers | | |
| 🕀 🛄 slow_log | | |
| 🕂 🔚 tables priv | | |

图 5.23 accountinfo 表的操作界面

1. 在accountinfo表中添加1条记录

添加记录是很简单的,只要在图 5.23 中最后 1 条记录下面接着填入数据就可以了。下面就向 accountinfo 表中添加数据,如图 5.24 所示。

| Query 器 Query Builder 器 架构设计器 Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->List All T Ⅰ | × ags. | | | | | |
|---|-----------|--|--|--|--|--|
| Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->List All T I 保存 | ags. | | | | | |
| 保存 | | | | | | |
| 1 保存 | | | | | | |
| | Þ | | | | | |
| 🚡 1结果 🧑 2 Profiler 🙂 3 信息 🔳 4 表数据 🖊 5 Info 🛅 6 历史 | | | | | | |
| Image: Solution of the second seco | | | | | | |
| id name account accounttype accountdate remarks | | | | | | |
| 2 吃饭 120.00 购物 20120501 消费2 | | | | | | |
| * 3 看电影 50 娱乐 20120201 2张票 | | | | | | |
| * OUTT) OUTT) OUTT) OUTT) OUTT) | | | | | | |
| Database: mysql Table: accountinfo | | | | | | |

图 5.24 向 accountinfo 表中添加 1 条记录

图 5.24 中 id 是 3 的记录就是新添加的记录。添加好数据之后,当把光标移动到下一 行或者单击图中标注的"保存"按钮即可保存当前的添加数据。

2. 修改accountinfo表中的数据

修改表中的数据操作更简单,直接在需要修改的数据项上改就可以了。修改完记录后, 只需要像添加数据时一样,将光标移动到下一行或者单击图 5.24 中标注的"保存"按钮即 可保存当前修改的数据。请读者试着将看电影这个消费项目的金额改成 100。

3. 删除accountinfo表中的数据

删除表中的数据,要先选中要删除的数据,这里,选中第1行记录,如图 5.25 所示。

• 84 •

| 🗞 Query | 🔡 Query Bu | ilder 🔡 势 | 喂构设计器 | | | | () × |
|-----------|-----------------|-----------------|------------|------------------|-----------------|--------------|------------------|
| Autocomp | lete: [Tab]->Ne | ext Tag. [Ct | rl+Space]- | >List Mat | ching Tags. | [Ctrl+Enter] | ->List All Tags. |
| 1 一册原 | | | | | | | |
| • | | | | | | | Þ |
| 1 结果 | Profile | er 🙂 <u>3</u> (| 言息 🔲 4 | 表数据 | 🍋 <u>5</u> Info | 11 6 历史 | |
| 📲 🛃 🗖 | L 🔊 🖏 🗔 | .l Row 🖲 Ro | ws in a Ra | ng First Row: | •0 | No. Rows | f 50 |
| id | name | account | accounttyp | e acc | ountdate | remarks | |
| | 2 吃饭 | 120.00 | 购物 | 2012 | 20501 | 消费2 | |
| | 3 看电影 | 50.00 | 娱乐 | 2012 | 20201 | 2张票 | |
| * (N | NTT) (MNTT) | (NULL) | (NULL) | (MU) | T) | (NULL) | |
| | | | | | | | |
| Database: | mysql Table: | accountinfo | | | | | |

图 5.25 删除选中记录的操作界面

在图 5.25 中单击标注的"删除"标记,弹出如图 5.26 所示的删除提示界面。其中选择"是"按钮,即可删除选中的记录。

| SQLyog : | 企业版 - ∎ySQL GUI 🔀 |
|----------|--|
| ? | Do you really want to delete the currently selected row? |
| | 是(1) 【 古(1)] |

图 5.26 删除数据提示

5.5 数据表数据操作综合实例

通过前面4节的讲解,相信读者已经迫不及待地想试试所学的知识了。在本节中将通 过一个综合实例来考核一下对数据表的基本操作。

【示例 12】 在 MySQL 中创建天气预报信息表(WeatherInfo),并完成对该表的如下 操作。表结构如表 5.7 所示。

| 编号 | 列 名 | 数 据 类 型 | 中文释义 | | | | | |
|----|-------------|--------------|--------|--|--|--|--|--|
| 1 | id | int | 编号, 主键 | | | | | |
| 2 | cityname | varchar(20) | 城市名称 | | | | | |
| 3 | weather | varchar(30) | 天气状况 | | | | | |
| 4 | weathertime | varhcar(20) | 时间 | | | | | |
| 5 | remarks | varchar(200) | 备注 | | | | | |

表 5.7 天气预报信息表(WeatherInfo)

建表的语句如下所示:

CREATE TABLE WeatherInfo

```
id int PRIMARY KEY,
cityname varchar(20),
weather varchar(30),
weathertime varchar(20),
remarks varchar(200));
```

(1)向 WeatherInfo 表中添加如表 5.8 所示的数据。

| 表 | 5.8 | 向天气预报 | 信息表添加的 | 数据 |
|---|-----|-------|--------|----|
| | | | | |

| 编号 | id | cityname | weather | weathertime | remarks |
|----|----|----------|-----------|-------------|---------|
| 1 | 1 | 北京 | 晴 20~30 | 20120601 | 注意防晒 |
| 2 | 2 | 上海 | 阴转晴 20~32 | 20120601 | 带伞 |
| 3 | 3 | 南京 | 晴 25~34 | 20120601 | 注意防晒 |
| 4 | 4 | 天津 | 小雨 18~27 | 20120601 | 带伞 |
| 5 | 5 | 大连 | 阴 19~23 | 20120601 | 带伞 |

(2) 假设有一个城市信息表,表结构如表 5.9 所示。将城市信息表中的城市编号和城市名称添加到天气预报信息表中。

表 5.9 城市信息表 (cityinfo)

| 编号 | 列 名 | 数 据 类 型 | 中文释义 |
|----|----------|-------------|--------|
| 1 | id | int | 编号, 主键 |
| 2 | cityname | varchar(20) | 城市名称 |

(3) 修改天气预报信息表中编号是1的天气日期,将其日期修改成"20120702"。

(4) 修改天气预报信息表中 2 条记录的天气状况,将其修改成"多云有阵雨"。

(5) 删除城市信息表中编号是11的城市信息。

(6) 按天气预报信息表中编号的顺序,降序排列,删除前2条数据。

(7) 删除城市信息表中的全部数据。

下面对本题一一解答如下。

(1)应用 INSERT...VALUES 语句一次添加多条记录,添加语句如下所示。执行效果 如图 5.27 所示。

INSERT INTO weatherinfo
VALUES(1,'北京','晴 20~30','20120601','注意防晒'),
(2,'上海','阴转晴 20~32','20120601','带伞'),
(3,'南京','晴 25~34','20120601','注意防晒'),
(4,'天津','小雨 18~27','20120601','带伞'),
(5,'大连','阴 19~23','20120601','带伞');

(2)将 cityinfo 表中的数据插入到 weatherinfo 中,使用的语句是 INSERT...SELECT。 具体的语句如下所示:

INSERT INTO weatherinfo(id,cityname)
SELECT id,cityname FROM cityinfo;

这里,需要注意的是在 weatherinfo 中 id 是主键,不能够重复。因此,要确保城市信息表(cityinfo)中的 id 不能够重复,才能添加成功。执行效果如图 5.28 所示,从图的运行结果可以看出,已经把城市信息表中的3条记录插入到了天气预报信息表中,对于天气

• 86 •

预报信息表中没有添加的字段全部都是 NULL 值。

| 😎 BySQL 5.5 Comman | nd Line Client | | | | |
|---|--|---|----------------|-------|--|
| nysql> INSERT INTO weatherinfo -> UALUES<1.'北京'.'晴 20~30'.'20120601'.'注意防晒'>. -> (2.'上海'.'阴转晴 20~32'.'20120601'.'带伞'>. -> (3.'南京'.'晴 25~34'.'20120601'.' 注意防晒'>. -> (4.'天津'.'小雨 18~27'.'20120601'.' 带伞'>. -> (5.'大连'.'阴 19~23'.'20120601'.' 带伞'); Query OK, 5 rows affected <0.05 sec> Records: 5 Duplicates: 0 Warnings: 0 nysql> SELECT * FROM weatherinfo; | | | | | |
| mysql> SELECT * ++ | FROM weatherinfo + | o; + weathertime | + remarks | + | |
| ↓ 1 : 北京 ↓ 2 : 上京 ↓ 3 : 南京 ↓ 4 : 天连 ↓ 5 : 大连 | + 晴 20~30 ! 阴转晴 20~32 ! 晴 25~34 ! 小雨 18~27 ! 阴 19~23 | + 20120601 20120601 20120601 20120601 20120601 | → | • | |
| 5 rows in set (0 | 1.00 sec) | • | + | • | |

图 5.27 批量向 weatherinfo 表中插入数据

| Bas ∎ySQ | L 5.5 Comman | d Line Client | | | | _ 🗆 × |
|--------------|--------------|------------------------------|-------------|-----------|---------|---------|
| mysql> | SELECT * I | FROM cityinfo; | | | | |
| ++ | | + | | | | |
| 1 10 1 tt | cityname | • | | | | |
| 11 | 沈阳 | l | | | | |
| 12 | 本溪 | I | | | | |
| 13 | 杭州 | I | | | | |
| 3 2040 | in set (0 | + 00 sec) | | | | |
| 5 FOWS | III 360 (0) | .00 3007 | | | | |
| mysql> | INSERT IN | [O weatherinfo(i | d,cityname) | | | |
| -> | SELECT id, | cityname FROM c | ityinfo; | | | |
| Query | OK, 3 rows | affected (0.05 | sec) | | | |
| necoru | s. 5 Dupi. | ICates. 6 warni | | | | |
| mysql> | SELECT * I | FROM weatherinfo | ; | | | |
| tt | | •• | | • | + | |
| 1 14 1 | cityname | Weather | weathertime | i remarks | 1 •• | |
| 1 1 1 | 北京 | , 于于10~30 ; | 20120601 | : 注意防晒 | 1 | |
| 1 2 1 | 上海 | : 阴转晴 20~32 : | 20120601 | : 带尘 | 1 | |
| 1 3 1 | 南京 | 啃_25~34 | 20120601 | : 涯意防晒 | | |
| 1 4 1 | 茶徒 | :/▶附 18**27 : :阳 19**23 : | 20120601 | 事業 | 1 | |
| 1 11 1 | 涗銆 | INULL I | NULL | NULL | | |
| 12 | 本溪 | I NULL I | NULL | NULL | 1 | |
| 13 | 杭州 | I NULL I | NULL | NULL | 1 | |
| ++ | /B | ++ 00> | | • | •+ | |
| o rows | III SEC (0. | .00 3667 | | | | |
| mysql> | | | | | | - |

图 5.28 复制 cityinfo 表中的数据到 weatherinfo 中

(3) 将天气预报编号作为条件来修改数据表的数据,具体语句如下所示:

UPDATE weatherinfo SET weathertime='20120702' WHERE id=1;

执行效果如图 5.29 所示。

从图 5.29 中可以看出,编号是 1 的记录中 weathertime 已经修改成了"20120702"。

(4) 要修改表中的 2 条记录, 要使用 LIMIT 关键来限制修改的记录数。具体的语句如 下所示:

UPDATE weatherinfo SET weather='多云有阵雨' LIMIT 2;

执行效果如图 5.30 所示,从图中可以看出,前两条记录中的 weather 已经都修改成了 "多云有阵雨"。

| Ba BySQ | L 5.5 Comman | d Line Client | | | | - U X |
|--|--------------|--|--|--|-------------|----------|
| mysql> UPDATE weatherinfo SET weathertime='20120702' WHERE id=1; Query OK, 1 row affected (0.05 sec) Rows matched: 1 Changed: 1 Warnings: 0 mysql> SELECT * FROM weatherinfo; | | | | | | |
| l id | cityname | weather | weathertime | i remarks | • - | |
| + 1 + 2 + 3 + 4 + 5 + 11 + 12 + 13 + | 北上南天大沈本杭 | : 晴 20~30 : 閉转晴 20~32 : 晴 25~34 : 小雨 18~27 : 阴 19~23 : NULL : NULL : NULL | 20120702 20120601 20120601 20120601 20120601 20120601 NULL NULL NULL | : 注意 · 注意 · 注意 · 带意 · 带令 · 带小 · NULL · NULL · NULL | | |
| 8 rows | s in set (Ø | .00 sec) | | • | • | • |

图 5.29 修改编号是1的数据

| BySQL 5.5 Command Line Client | | | | | |
|---|---------------|--|--|--|---|
| mysql〉UPDATE weatherinfo SET weather='多云有阵雨' LIMIT 2; Query OK, 2 rows affected (0.03 sec) Rows matched: 2 Changed: 2 Warnings: 0 | | | | | |
| mysql> SELECT * FROM weatherinfo; | | | | | |
| id cityname | weather | weathertime | remarks | l. | |
| 1 1 北京京 1 1 北京京 1 2 1 3 1 市京洋 3 1 五京洋 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 13 1 1 13 1 | + | 20120702 20120601 20120601 20120601 20120601 20120601 20120601 NULL NULL NULL | :: :: :: :: :: :: :: :: :: :: :: :: :: | * * * * * * * * * * * * * * | |
| ++ | + .00 sec> | + | + | + | _ |

图 5.30 使用 LIMIT 限制修改 2 条记录

(5) 由编号作为条件删除城市信息表中的数据, 删除语句如下所示:

DELETE FROM cityinfo WHERE id=11;

执行效果如图 5.31 所示,从图中可以看出, id 为 11 的记录已经被删除掉了。

| BySQL 5.5 Command Line Client | | | | |
|---|-----------|--|--|--|
| mysql> DELETE FROM cityinfo WHERE id=11; Query OK, 1 row affected (0.06 sec) | | | | |
| nysql> SELECT * FROM cityinfo; ++ | | | | |
| id cityname + | | | | |
| 1 12 本溪 13 杭州 + | | | | |
| 2 rows in set (0.00 sec) | | | | |
| nysql> 1 | ۲ // ۱ | | | |

图 5.31 删除 id 为 11 的城市信息

(6) 在删除时使用排序和限制删除的记录数,具体语句如下所示:

• 88 •

DELETE FROM weatherinfo ORDER BY id DESC LIMIT 2;

执行效果如图 5.32 所示,从图中可以看出,已经将编号是 12 和 13 的记录被删除掉了。

| | <mark>⊛a</mark> ∎ySi | QI | . 5.5 Comman | d Line Client | | | | | |
|---|---|-------------|--------------|--|--|---|--|-------|---|
| | mysql> DELETE FROM weatherinfo ORDER BY id DESC LIMIT 2; Query OK, 2 rows affected (0.05 sec) mysql> SELECT * FROM weatherinfo; | | | | | | 2; | | |
| ľ | id | 1 | cityname | weather | weathertime | ļ | remarks | + | |
| | 1 2 3 4 5 11 | | 北上南天大沈 | 多云有阵雨 多云有阵雨 晴 25~34 小雨 18~27 阴 19~23 NULL | 20120702 20120601 20120601 20120601 20120601 20120601 NULL | | 注带注意 中 市 市 市 市 市 市 市 市 市 市 市 市 市 | * | |
| | i row nysql | * s > | in set (Ø | .00 sec) | | | | * | • |

图 5.32 删除排序后的前 2 条记录

(7) 删除一张表中的全部记录, 是最简单的操作了。具体语句如下所示:

DELETE FROM cityinfo;

执行上面的语句就可以将 cityinfo 中的数据全部删除掉了。除了用 DELETE FROM 语 句删除外,还可以用 TRUNCATE TABLE cityinfo 语句完成数据的删除。

通过示例 12,相信读者已经能够基本掌握 DML 语句是如何操作数据表的了,那么, 请读者再把这些例子在图形界面里操作一遍,以熟悉图形界面的使用。

5.6 本章小结

在本章中主要讲解了使用 INSERT 语句向数据表中插入数据、使用 UPDATE 语句修改 数据表中的数据、使用 DELETE 语句删除数据表中的数据。此外,还学习了在图形界面 SQLyog 企业版中如何操作数据表中的数据。另外,在应用 UPDATE 和 DELETE 语句时还 要特别的注意,只有针对单表的操作时才能够使用 ORDER BY 和 LIMIT 子句。

5.7 本章习题

一、填空题

- (1) 向数据表中添加数据使用的基本语句是____。
- (2) 修改数据表中的数据使用的基本语句是。
- (3) 删除数据表中的数据使用的基本语句是____。
- (4) 按照顺序修改数据表中的数据使用的语句是_____。

二、选择题

对数据表的添加、修改、删除操作属于 SQL 语句中的_____。
 A. DDL
 B. DCL

• 89 •

C. DQL

- D. DML
- (2)下面对添加数据的描述哪个是正确的?__
 - A. 给主键列添加值时, 值是可以重复的
 - B. 一次不能向表中添加多条数据
 - C. 向数据表添加数据时,可以将其他表的数据复制到当前表中
 - D. 以上都正确
- (3) 限制修改行数的关键字是____。
 - A. ORDER C. LIMIT

B. DISTINCT

D. WHERE

三、上机题

(1) 假设有如表 5.10 所示的表结构, 向数据表中添加如表 5.11 所示的数据。

| 表 5.10 图书信息表(BookrInfo) | | | | | | |
|-------------------------|---------|--------------|------|--|--|--|
| 编号 | 列 名 | 数 据 类 型 | 中文释义 | | | |
| 1 | id | integer | 图书编号 | | | |
| 2 | name | varhcar(12) | 图书名称 | | | |
| 3 | price | decimal(5,2) | 图书价格 | | | |
| 4 | author | varchar(4) | 图书作者 | | | |
| 5 | pub | varchar(15) | 出版社 | | | |
| 6 | remarks | varchar(200) | 备注 | | | |

表 5.11 向图书信息表添加的数据

| 编号 | id | name | price | author | pub | remarks |
|----|----|------|-------|--------|------|---------|
| 1 | 1 | 数据库 | 30 | 张三 | 北京大学 | 畅销书 |
| 2 | 2 | 会计实务 | 35 | 李四 | 南京大学 | 教材 |
| 3 | 3 | 大学物理 | 28 | 王五 | 大连大学 | 教材 |
| 4 | 4 | 数据结构 | 36 | 赵四 | 沈阳大学 | 教材 |
| 5 | 5 | 英语口语 | 25 | 刘六 | 上海大学 | 应试 |

(2) 修改图书信息表中编号是1的图书信息,将其价格修改成32.5。

(3) 使用限制修改行数的方法,将表中前2行数据中的作者修改成"未知"。

(4) 将价格高于 30 的图书,降低 5 元。

(5) 删除表中编号是1的图书信息。

(6) 删除表中前2条图书信息。