



第5节

数据处理之增删改

讲师：李玉婷

目 标

通过本章学习，您将可以：

- 使用 DML 语句
- 向表中插入数据
- 更新表中数据
- 从表中删除数据

数据操纵语言

- DML (Data Manipulation Language - 数据操纵语言) 可以在下列条件下执行：
 - 向表中插入数据
 - 修改现存数据
 - 删除现存数据
- 事务是由完成若干项工作的DML语句组成的

插入数据

DEPARTMENTS

70	Public Relations	100	1700
----	------------------	-----	------

新行

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

向 DEPARTMENTS
表中插入新的记录

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

70	Public Relations	100	1700
----	------------------	-----	------

INSERT 语句语法

- 使用 INSERT 语句向表中插入数据。

```
INSERT INTO    table [(column [, column...])]  
VALUES         (value [, value...]);
```

- 使用这种语法一次只能向表中插入一条数据。

放屁，可以插入多条记录，直接在values后面添加多个括号即可，如

insert into xx

values(...), (...), (...)这就是插入三条记录

还支持子查询，将values替换为一个子查询结果，可以为一行或多行记录

还有一种语法：

insert into table

set xx=xx, xx=xx

这种语法无法插入多条记录，且不支持子查询

插入数据

- 为每一列添加一个新值。
- 按列的默认顺序列出各个列的值。
- 在 INSERT 子句中随意列出列名和他们的值。
- 字符和日期型数据应包含在单引号中。

只要指定了列名和对应的数值，列的顺序无所谓，如果没有指定，则按照表中默认顺序

```
INSERT INTO departments(department_id, department_name,  
                        manager_id, location_id)  
VALUES      (70, 'Public Relations', 100, 1700);  
1 row created.
```

```
INSERT INTO  
employees(employee_id,last_name,email,hire_date,job_id)  
VALUES      (300,'Tom','tom@126.com',to_date('2012-3-  
21','yyyy-mm-dd'),'SA_RAP');  
1 row created.
```

向表中插入空值

- 隐式方式：在列名表中省略该列的值。

```
INSERT INTO departments (department_id,  
                          department_name  )  
VALUES (30, 'Purchasing');  
1 row created.
```

- 显示方式：在VALUES子句中指定空值。

```
INSERT INTO departments  
VALUES (100, 'Finance', , );  
1 row created.
```

插入指定的值

NOW () 函数：记录当前系统的日期和时间。

```
INSERT INTO employees (employee_id,  
                        first_name, last_name,  
                        email, phone_number,  
                        hire_date, job_id, salary,  
                        commission_pct, manager_id,  
                        department_id)  
VALUES  
    (113,  
     'Louis', 'Popp',  
     'LPOPP', '515.124.4567',  
     NOW(), 'AC_ACCOUNT', 6900,  
     NULL, 205, 100);
```

1 row created.

从其它表中拷贝数据

- 在 INSERT 语句中加入子查询。

```
INSERT INTO emp2
```

```
SELECT *
```

```
FROM employees
```

```
WHERE department id = 90;
```

3 rows created.

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
```

```
SELECT employee_id, last_name, salary, commission_pct
```

```
FROM employees
```

```
WHERE job_id LIKE '%REP%';
```

4 rows created.

- 不必书写 VALUES 子句。
- 子查询中的值列表应与 INSERT 子句中的列名对应

更新数据

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_P
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	60	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	60	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	60	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

更新 EMPLOYEES 表

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_P
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	30	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	30	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	30	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

UPDATE 语句语法

- 使用 UPDATE 语句更新数据。

```
UPDATE          table  
SET             column = value [, column = value, ...]  
[WHERE          condition];
```

- 可以一次更新多条数据。
- 如果需要回滚数据，需要保证在DML前，进行设置：**SET AUTOCOMMIT = FALSE;**

还可以更新多个表：
两个表先连接，再从结果表中更新记录

更新数据

- 使用 **WHERE** 子句指定需要更新的数据。

```
UPDATE employees  
SET    department_id = 70  
WHERE  employee_id = 113;  
1 row updated.
```

- 如果省略 WHERE 子句，则表中的所有数据都将被更新

```
UPDATE    copy_emp  
SET       department_id = 110;  
22 rows updated.
```


更新中的数据完整性错误

```
UPDATE employees  
SET     department_id = 55  
WHERE   department_id = 110;
```

错误代码: 1452

```
Cannot add or update a child row: a foreign key  
constraint fails (`myemployees`.`employees`,  
CONSTRAINT `dept_id_fk` FOREIGN KEY (`department_id`)  
REFERENCES `departments` (`department_id`))
```

不存在 55 号部门

另例:

```
update employees set manager_id = 299 where employee_id = 203;
```

删除数据

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
100	Finance		
50	Shipping	124	1500
60	IT	103	1400

从表DEPARTMENTS 中删除一条记录。

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
50	Shipping	124	1500
60	IT	103	1400

DELETE 语句

使用 **DELETE 语句** 从表中删除数据。

```
DELETE FROM      table  
[WHERE           condition];
```

truncate子句：
直接删除一个表，不允许添加
where子句：
truncate from 表；

删除多表的记录：
先连接，再删除，删除哪个表的记录，就在delete 后面添加
该表的别名
delete a
from boy a
inner join girls b
on a.id = b.boyfriend_id
where a.age > 15;

删除数据

- 使用 WHERE 子句删除指定的记录。

```
DELETE FROM departments  
WHERE department_name = 'Finance';  
1 row deleted.
```

- 如果省略 WHERE 子句，则表中的全部数据将被删除

```
DELETE FROM copy_emp;  
22 rows deleted.
```


删除中的数据完整性错误

```
DELETE FROM departments  
WHERE      department_id = 60;
```

错误代码: 1451

```
Cannot delete or update a parent row: a foreign key  
constraint fails (`myemployees`.`employees`,  
CONSTRAINT `dept_id_fk` FOREIGN KEY (`department_id`)  
REFERENCES `departments` (`department_id`))
```

You cannot delete a row that contains a primary key that is used as a foreign key in another table.

总 结

通过本章学习，您应学会如何使用DML语句改变数据和事务控制

语句	功能
INSERT	插入
UPDATE	修正
DELETE	删除

