

05 修改数据

关系数据库的基本操作就是增删改查，即CRUD：Create、Retrieve、Update、Delete。其中，对于查询，我们已经详细讲述了 **SELECT** 语句的详细用法。

而对于增、删、改，对应的SQL语句分别是：

- **INSERT**：插入新记录；
- **UPDATE**：更新已有记录；
- **DELETE**：删除已有记录。

我们将分别讨论这三种修改数据的语句的使用方法。

INSERT

当我们需要向数据库表中插入一条新记录时，就必须使用 **INSERT** 语句。



INSERT 语句的基本语法是：

```
INSERT INTO <表名> (字段1, 字段2, ...) VALUES (值1, 值2, ...);
```

例如，我们向 **students** 表插入一条新记录，先列举出需要插入的字段名称，然后在 **VALUES** 子句中依次写出对应字段的值：

```
-- 添加一条新记录
INSERT INTO students (class_id, name, gender, score)
VALUES (2, '大牛', 'M', 80);
-- 查询并观察结果:
SELECT * FROM students;
```

ID	CLASS_ID	NAME	GENDER	SCORE
1	1	小明	M	90
2	1	小红	F	95
3	1	小军	M	88
4	1	小米	F	73
5	2	小白	F	81
6	2	小兵	M	55
7	2	小林	M	85
8	3	小新	F	91
9	3	小王	M	89
10	3	小丽	F	88
11	2	大牛	M	80

注意到我们并没有列出 `id` 字段，也没有列出 `id` 字段对应的值，这是因为 `id` 字段是一个自增主键，它的值可以由数据库自己推算出来。此外，如果一个字段有默认值，那么在 `INSERT` 语句中也可以不出现。

要注意，字段顺序不必和数据库表的字段顺序一致，但值的顺序必须和字段顺序一致。也就是说，可以写 `INSERT INTO students (score, gender, name, class_id) ...`，但是对应的 `VALUES` 就得变成 `(80, 'M', '大牛', 2)`。

还可以一次性添加多条记录，只需要在 `VALUES` 子句中指定多个记录值，每个记录是由 `(...)` 包含的一组值：

```
-- 一次性添加多条新记录
INSERT INTO students (class_id, name, gender, score)
VALUES
  (1, '大宝', 'M', 87),
  (2, '二宝', 'M', 81);

SELECT * FROM students;
```

ID	CLASS_ID	NAME	GENDER	SCORE
1	1	小明	M	90
2	1	小红	F	95
3	1	小军	M	88
4	1	小米	F	73
5	2	小白	F	81
6	2	小兵	M	55
7	2	小林	M	85

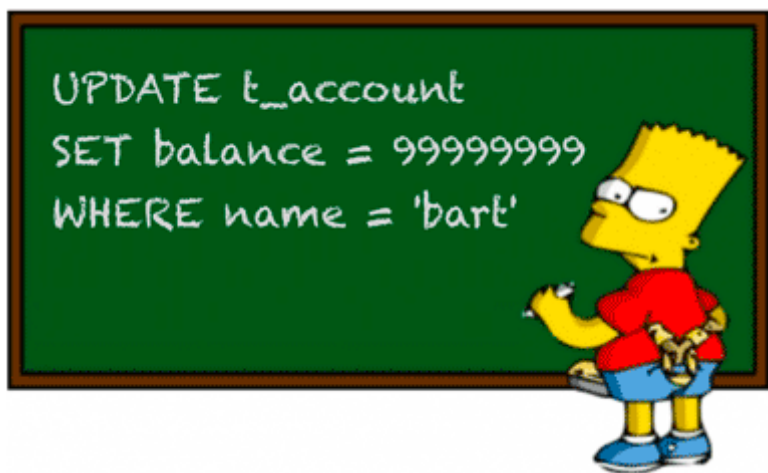
ID	CLASS_ID	NAME	GENDER	SCORE
9	3	小王	M	89
10	3	小丽	F	88
11	2	大牛	M	80
12	1	大宝	M	87
13	2	二宝	M	81

小结

使用 **INSERT**，我们就可以一次向一个表中插入一条或多条记录。

UPDATE

如果要更新数据库表中的记录，我们就必须使用 **UPDATE** 语句。



UPDATE 语句的基本语法是：

```
UPDATE <表名> SET 字段1=值1, 字段2=值2, ... WHERE ...;
```

例如，我们想更新 **students** 表 **id=1** 的记录的 **name** 和 **score** 这两个字段，先写出 **UPDATE students SET name='大牛', score=66**，然后在 **WHERE** 子句中写出需要更新的行的筛选条件 **id=1**：

```
-- 更新id=1的记录
UPDATE students SET name='大牛', score=66 WHERE id=1;
-- 查询并观察结果：
SELECT * FROM students WHERE id=1;
```

ID	CLASS_ID	NAME	GENDER	SCORE
1	1	大牛	M	66

注意到 **UPDATE** 语句的 **WHERE** 条件和 **SELECT** 语句的 **WHERE** 条件其实是一样的，因此完全可以一次更新多条记录：

```
-- 更新id=5,6,7的记录
UPDATE students SET name='小牛', score=77 WHERE id>=5 AND
id<=7;
-- 查询并观察结果:
SELECT * FROM students;
```

ID	CLASS_ID	NAME	GENDER	SCORE
1	1	大牛	M	66
2	1	小红	F	95
3	1	小军	M	88
4	1	小米	F	73
5	2	小牛	F	77
6	2	小牛	M	77
7	2	小牛	M	77
8	3	小新	F	91
9	3	小王	M	89
10	3	小丽	F	88

在 **UPDATE** 语句中，更新字段时可以使用表达式。例如，把所有80分以下的同学的成绩加10分：

```
-- 更新score<80的记录
UPDATE students SET score=score+10 WHERE score<80;
-- 查询并观察结果:
SELECT * FROM students;
```

ID	CLASS_ID	NAME	GENDER	SCORE
1	1	大牛	M	76
2	1	小红	F	95
3	1	小军	M	88
4	1	小米	F	83
5	2	小牛	F	87
6	2	小牛	M	87
7	2	小牛	M	87
8	3	小新	F	91
9	3	小王	M	89
10	3	小丽	F	88

其中，**SET score=score+10** 就是给当前行的 **score** 字段的值加上了10。

如果 **WHERE** 条件没有匹配到任何记录，**UPDATE** 语句不会报错，也不会有任何记录被更新。例如：

```
-- 更新id=999的记录
UPDATE students SET score=100 WHERE id=999;
-- 查询并观察结果：
SELECT * FROM students;
```

ID	CLASS_ID	NAME	GENDER	SCORE
1	1	大牛	M	76
2	1	小红	F	95
3	1	小军	M	88
4	1	小米	F	83
5	2	小牛	F	87
6	2	小牛	M	87
7	2	小牛	M	87
8	3	小新	F	91
9	3	小王	M	89
10	3	小丽	F	88

最后，要特别小心的是，`UPDATE` 语句可以没有 `WHERE` 条件，例如：

```
UPDATE students SET score=60;
```

这时，整个表的所有记录都会被更新。所以，在执行 `UPDATE` 语句时要非常小心，最好先用 `SELECT` 语句来测试 `WHERE` 条件是否筛选出了期望的记录集，然后再用 `UPDATE` 更新。

MySQL

在使用MySQL这类真正的关系数据库时，`UPDATE` 语句会返回更新的行数以及 `WHERE` 条件匹配的行数。

例如，更新 `id=1` 的记录时：

```
mysql> UPDATE students SET name='大宝' WHERE id=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

MySQL会返回1，可以从打印的结果 `Rows matched: 1 Changed: 1` 看到。

当更新 `id=999` 的记录时：

```
mysql> UPDATE students SET name='大宝' WHERE id=999;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0
```

MySQL会返回0，可以从打印的结果 `Rows matched: 0 Changed: 0` 看到。

小结

使用 `UPDATE`，我们就可以一次更新表中的一条或多条记录。

DELETE

如果要删除数据库表中的记录，我们可以使用 `DELETE` 语句。



`DELETE` 语句的基本语法是：

```
DELETE FROM <表名> WHERE ...;
```

例如，我们想删除 `students` 表中 `id=1` 的记录，就需要这么写：

```
-- 删除id=1的记录
DELETE FROM students WHERE id=1;
-- 查询并观察结果：
SELECT * FROM students;
```

ID	CLASS_ID	NAME	GENDER	SCORE
2	1	小红	F	95
3	1	小军	M	88
4	1	小米	F	73
5	2	小白	F	81
6	2	小兵	M	55
7	2	小林	M	85
8	3	小新	F	91
9	3	小王	M	89
10	3	小丽	F	88

注意到 `DELETE` 语句的 `WHERE` 条件也是用来筛选需要删除的行，因此和 `UPDATE` 类似，`DELETE` 语句也可以一次删除多条记录：

```
-- 删除id=5,6,7的记录
DELETE FROM students WHERE id>=5 AND id<=7;
-- 查询并观察结果：
SELECT * FROM students;
```

ID	CLASS_ID	NAME	GENDER	SCORE
2	1	小红	F	95
3	1	小军	M	88
4	1	小米	F	73
8	3	小新	F	91
9	3	小王	M	89
10	3	小丽	F	88

如果 `WHERE` 条件没有匹配到任何记录，`DELETE` 语句不会报错，也不会有任何记录被删除。例如：

```
-- 删除id=999的记录
DELETE FROM students WHERE id=999;
-- 查询并观察结果：
SELECT * FROM students;
```

ID	CLASS_ID	NAME	GENDER	SCORE
2	1	小红	F	95
3	1	小军	M	88
4	1	小米	F	73
8	3	小新	F	91
9	3	小王	M	89
10	3	小丽	F	88

最后，要特别小心的是，和 `UPDATE` 类似，不带 `WHERE` 条件的 `DELETE` 语句会删除整个表的数据：

```
DELETE FROM students;
```

这时，整个表的所有记录都会被删除。所以，在执行 `DELETE` 语句时也要非常小心，最好先用 `SELECT` 语句来测试 `WHERE` 条件是否筛选出了期望的记录集，然后再用 `DELETE` 删除。

MySQL

在使用MySQL这类真正的关系数据库时，`DELETE` 语句也会返回删除的行数以及 `WHERE` 条件匹配的行数。

例如，分别执行删除 `id=1` 和 `id=999` 的记录：

```
mysql> DELETE FROM students WHERE id=1;  
Query OK, 1 row affected (0.01 sec)  
  
mysql> DELETE FROM students WHERE id=999;  
Query OK, 0 rows affected (0.01 sec)
```

小结

使用 `DELETE`，我们就可以一次删除表中的一条或多条记录。