SQL 必知必会 11 - 使用子查询

11.1 子查询

简单查询，即从单个数据库表中检索数据的单条语句。

> **查询（ query ）**
> 任何 SQL 语句都是查询。但此术语一般指 SELECT 语句。

SQL 还允许创建子查询（ subquery ），即**嵌套在其他查询中的查询**

> MySQL 支持
> 如果使用 MySQL ，应该知道对子查询的支持是从 4.1 版本引入的。
> MySQL 的早期版本不支持子查询。

11.2 利用子查询进行过滤

现在，假如需要列出订购物品 RGAN01 的所有顾客：

1. 检索包含物品 RGAN01 的所有订单号。
2. 检索具有前一步骤列出的订单编号的所有顾客的 ID 。
3. 检索前一步骤返回的所有顾客 ID 的顾客信息。

两种处理方式：

- 可以把一条 SELECT 语句返回的结果用于另一条 SELECT 语句的 WHERE 子句。
- 也可以使用子查询来把 3 个查询组合成一条语句。

第一步：

```
SELECT order_num
FROM OrderItems
WHERE prod_id = 'RGAN01';
```

得到输出：

```
order_num
-----------
20007
20008
```

第二步：

```
SELECT cust_id
FROM Orders
WHERE order_num IN (20007,20008);
```

第一步结合第二步，变成子查询：

```
SELECT cust_id
FROM Orders
WHERE order_num IN (SELECT order_num
                    FROM OrderItems
                    WHERE prod_id = 'RGAN01');
```

得到的客户 `ID` 结果是：

```
cust_id
----------
1000000004
1000000005
```

在 `SELECT` 语句中，子查询总是从内向外处理。

> 包含子查询的 `SELECT` 语句难以阅读和调试。
> 把子查询分解为多行并进行适当的缩进，能极大地简化子查询的使用。
> 好的 `DBMS` 客户端正是出于这个原因使用了颜色代码 `SQL` 。

第三步，根据 `ID` 查询客户信息：

```
SELECT cust_name, cust_contact
FROM Customers
WHERE cust_id IN ('1000000004','1000000005');
```

把三个步骤全部结合起来：

```
SELECT cust_name, cust_contact
FROM Customers
WHERE cust_id IN (SELECT cust_id
                  FROM Orders
                  WHERE order_num IN (SELECT order_num
                                      FROM OrderItems
                                      WHERE prod_id = 'RGAN01'));
```

对于能嵌套的子查询的数目没有限制，不过在实际使用时由于性能的限制，不能嵌套太多的子查询。

> 作为子查询的 `SELECT` 语句只能查询单个列。企图检索多个列将返回错误。

11.3 子查询用作计算字段

```
SELECT cust_name,
    cust_state,
    (SELECT COUNT(*)
     FROM Orders
     WHERE Orders.cust_id = Customers.cust_id) AS orders
FROM Customers
ORDER BY cust_name;
```

子查询中的 `WHERE` 子句与前面使用的 `WHERE` 子句稍有不同，因为它使用了完全限定列名。
它指定表名和列名。
用一个句点分隔表名和列名，在有可能混淆列名时必须使用这种语法。