第 5 章
单 表 操 作

学习目标：

- 掌握表结构和数据的复制，能够复制现有数据表的表结构和数据。
- 掌握主键冲突的解决方法，能够解决主键冲突问题。
- 掌握清空数据操作，能够利用 TRUNCATE 语句清空数据。
- 掌握如何去除查询结果中的重复记录，能够利用 DISTINCT 实现去重查询。
- 掌握排序查询操作，能够利用 ORDER BY 对返回的查询结果进行排序。
- 掌握限量查询操作，能够利用 LIMIT 对返回的数据进行限量。
- 掌握分组查询操作，能够利用 GROUP BY 对返回的查询结果进行分组。
- 掌握聚合函数的使用，能够根据不同场景对查询数据进行统计。
- 熟悉算术运算符的用法，能够说明每个算术运算符的含义。
- 熟悉比较运算符的用法，能够说明每个比较运算符的含义。
- 熟悉逻辑运算符的用法，能够说明每个逻辑运算符的含义。
- 熟悉赋值运算符的用法，能够说明每个赋值运算符的含义。
- 熟悉位运算符的用法，能够说明每个位运算符的含义。
- 熟悉运算符的优先级，能够说明常用运算符的优先级。

在前面的章节中已经学习了数据表和数据的基本操作。但是，实际的需求可能会更加复杂，仅仅通过前面学习的知识并不能完全满足开发的需要。例如，对查询到的数据进行排序、限量和分组，以及连接多张表进行查询等。因此，需要学习更多关于数据操作的知识，这些知识主要分为单表操作和多表操作，将在第 5 章和第 6 章中进行详细讲解。本章讲解数据库中的单表操作。

5.1 数据进阶操作

在实际开发中，除了需要对数据进行添加、修改、查询和删除外，有时还需要进行一些进阶操作，例如复制表结构和数据、解决主键冲突、清空数据和去除查询结果中的重复记录。本节对数据进阶操作进行详细讲解。

5.1.1 复制表结构和数据

MySQL 中提供了专门的 SQL 语句，用于创建表并复制已有数据表的表结构和数据。

下面分别进行讲解。

1. 复制已有的表结构

在开发时，若需要创建一个与已有数据表相同结构的数据表，基本语法格式如下。

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] 新数据表名称 {LIKE 源表 | (LIKE 源表)}
```

上述语法格式中，在复制已有的数据表结构时，使用“LIKE 源表”与使用“(LIKE 源表)”语法效果相同，任选其一即可。通过这种方式复制的数据表为一个空表，该表包括源表中定义的任何字段属性和索引，但不会复制源表中保存的数据。

为了帮助读者更好地理解如何复制已有的表结构，接下来将以 4.4 节动手实践中设计的电子商务网站数据库为例进行演示。读者可通过本书配套源代码获取该数据库的 SQL 文件，导入 MySQL 中，将数据库命名为 shop。下面演示如何复制 shop 数据库中 sh_goods 数据表的表结构，将复制出来的数据表命名为 my_goods，并存放于 mydb 数据库中，具体 SQL 语句及执行结果如下。

```
mysql>USE shop;
Database changed
mysql>CREATE TABLE mydb.my_goods (LIKE sh_goods);
Query OK, 0 rows affected (0.05 sec)
```

按以上步骤创建完成后，通过 SHOW CREATE TABLE 语句查看 my_goods 数据表的结构，具体 SQL 语句及执行结果如下。

```
mysql>SHOW CREATE TABLE mydb.my_goods;
+----------+-------------------------------------------------------+
| Table    | Create Table                                          |
+----------+-------------------------------------------------------+
| my_goods | CREATE TABLE `my_goods` (                             |
|          |  `id` int unsigned NOT NULL AUTO_INCREMENT            |
|          |  COMMENT '商品 id',                                    |
|          |  `category_id` int unsigned NOT NULL DEFAULT '0'      |
|          |  COMMENT '分类 id',                                    |
|          |  `spu_id` int unsigned NOT NULL DEFAULT '0'           |
|          |  COMMENT 'SPU id ',                                   |
|          |  ……(此处省略部分字段)                                  |
|          |  PRIMARY KEY (`id`)                                   |
|          | ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4               |
|          | COLLATE=utf8mb4_0900_ai_ci COMMENT='商品表'           |
+----------+-------------------------------------------------------+
1 row in set (0.00 sec)
```

从上述执行结果可以看出，当前已经成功依据已有的数据表创建出与其结构相同的数据表。

2. 复制已有的表数据

复制已有的表数据是新增数据的一种方式，它是从已有的数据中获取数据，并且将获取

的数据添加到对应的数据表中。需要注意的是,此种方式获取数据与添加数据的表结构要相同,否则可能会遇到添加不成功的情况,基本语法格式如下。

```
INSERT [INTO] 新数据表名称 [(字段名[, …])]
SELECT * | {字段名[, …]} FROM 源表;
```

在上述语法格式中,如果将新数据表名称和源表设为同一个数据表,可在短期内快速增加该数据表的数据量。

下面演示如何将 sh_goods 表中的数据复制到数据表 my_goods 中,具体 SQL 语句及执行结果如下。

```
mysql> INSERT INTO mydb.my_goods SELECT * FROM sh_goods;
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

执行完上述 SQL 语句后,使用 SELECT 语句查看商品数据的添加情况,会看到已将数据表 sh_goods 中的数据完全复制到了数据表 my_goods 中。由于查询结果很长,比较占用篇幅,这里不再演示。

需要注意的是,在向一张数据表中复制数据时,如果该数据表中含有主键,可能会遇到主键重复的问题。例如,再次将 sh_goods 表中的数据复制到 my_goods 表中,系统会报主键重复的错误,具体 SQL 语句如下。

```
mysql> INSERT INTO mydb.my_goods SELECT * FROM sh_goods;
ERROR 1062 (23000): Duplicate entry '1' for key 'my_goods.PRIMARY'
```

对于上述问题,可以通过指定主键 id 字段以外的字段来完成数据复制,具体 SQL 语句如下。

```
mysql>  INSERT INTO mydb.my_goods (category_id, name, keyword, content,
    ->  price, stock, score, comment_count)
    ->  SELECT category_id, name, keyword, content, price, stock,
    ->  score, comment_count FROM sh_goods;
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

上述 SQL 语句中,在添加数据时,指定除主键 id 字段之外的字段,避免了主键重复。

多学一招:临时表

临时表是一种在当前会话中可见,并在当前会话关闭时自动删除的数据表,主要用于临时存储数据。若要创建临时表,只需要在 CREATE 与 TABLE 关键字中间添加 TEMPORARY 即可,示例如下。

```
#方式 1: 创建临时表
CREATE TEMPORARY TABLE mydb.tmp_table1 (id int);
Query OK, 0 rows affected (0.00 sec)
#方式 2: 创建临时表并复制数据
CREATE TEMPORARY TABLE mydb.tmp_table2 SELECT id, name FROM shop.sh_goods;
Query OK, 10 rows affected (0.00 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

上述示例中，方式 1 表示在 mydb 数据库下创建一个 tmp_table1 临时表；方式 2 表示将 shop 数据库下的 sh_goods 数据表中的数据复制到 mydb 数据库下的 tmp_table2 临时表中。临时表中数据的操作与普通表相同，都可以进行 SELECT、INSERT、UPDATE 和 DELETE 操作，这里不再演示。

需要注意的是，使用 SHOW TABLES 语句不能查看当前数据库下有哪些临时表。

若要修改临时表的表名必须使用 ALTER TABLE，而不能使用 RENAME TABLE。下面演示如何将 tmp_table2 的表名改为 tmp_table3，具体 SQL 语句及示例结果如下。

```
mysql>ALTER TABLE mydb.tmp_table2 RENAME TO mydb.tmp_table3;
Query OK, 10 rows affected (0.00 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

使用 DESC 语句查看临时表的结构，具体 SQL 语句及执行结果如下。

```
mysql>DESC mydb.tmp_table3;
+--------+---------------+------+-----+----------+--------+
| Field  | Type          | Null | Key | Default  | Extra  |
+--------+---------------+------+-----+----------+--------+
| id     | int unsigned  | NO   |     | 0        | NULL   |
| name   | varchar(120)  | NO   |     |          | NULL   |
+--------+---------------+------+-----+----------+--------+
2 rows in set (0.00 sec)
```

上述 SQL 语句中，使用 DESC 语句可以查看 tmp_table3 临时表的表结构。

5.1.2　解决主键冲突

在向数据表中添加一条记录时，如果添加的记录的主键值在现有的数据中已经存在，会产生主键冲突的情况。

下面演示主键冲突的情况。例如，当 my_goods 表经过数据复制以后，再添加一条会引起主键冲突的数据，具体 SQL 语句及执行结果如下。

```
mysql>INSERT INTO mydb.my_goods (id, name, content, keyword)
    -> VALUES (20, '橡皮', '修正书写错误', '文具');
ERROR 1062 (23000): Duplicate entry '2' for key 'my_goods.PRIMARY'
```

从上述执行结果可以看出，系统提示添加数据的主键发生冲突。若要解决这类问题，MySQL 中提供了两种方式，分别为主键冲突更新和主键冲突替换，下面分别进行讲解。

1. 主键冲突更新

主键冲突更新是指在添加数据的过程中若发生主键冲突，则添加数据操作利用更新的方式实现，语法格式如下。

```
INSERT [INTO] 数据表名称 [(字段名[, …])]
{VALUES | VALUE} (值[, …])
ON DUPLICATE KEY UPDATE 字段名 1=新值 1[, 字段名 2=新值 2, …];
```

上述语法格式中,在 INSERT 语句后添加 ON DUPLICATE KEY UPDATE,以便在主键冲突时,通过“字段名 1=新值 1[, 字段名 2=新值 2, …]”更新此条记录中设置的字段名对应的新值。

例如,修改以上发生主键冲突的添加语句,具体 SQL 语句及执行结果如下。

```
mysql>INSERT INTO mydb.my_goods (id, name, content, keyword)
    ->VALUES (20, '橡皮', '修正书写错误', '文具')
    ->ON DUPLICATE KEY UPDATE name='橡皮', content='修正书写错误',
    ->keyword='文具';
Query OK, 2 rows affected (0.01 sec)
```

上述 SQL 语句中,添加数据时使用 ON DUPLICATE KEY UPDATE 更新 name 字段、content 字段和 keyword 字段的值。由执行结果可知,当添加的记录与数据表中已存在的记录主键冲突时,返回的结果为 2 rows affected,表示影响了两条记录。

修改完成后,查看数据是否添加成功,具体 SQL 语句及执行结果如下。

```
mysql>SELECT name, content, keyword FROM mydb.my_goods WHERE id=20;
+------+--------------+----------+
| name | content      | keyword  |
+------+--------------+----------+
| 橡皮 | 修正书写错误 | 文具     |
+------+--------------+----------+
1 row in set (0.00 sec)
```

从上述执行结果可以看出,成功查询出 id 字段值为 20 的记录,说明数据添加成功。

2. 主键冲突替换

主键冲突替换是指在添加数据的过程中若发生主键冲突,则先删除原有记录,再新增记录,语法格式如下。

```
REPLACE [INTO] 数据表名称 [(字段名[, …])]
{VALUES|VALUE} (值[, …]);
```

上述语法中,REPLACE 语句与 INSERT 语句的使用类似,区别在于前者每执行一次就会发生两个操作,即删除记录和添加记录。

例如,修改发生主键冲突的添加语句,具体 SQL 语句及执行结果如下。

```
mysql>REPLACE INTO mydb.my_goods (id, name, content, keyword)
    ->VALUES (20, '橡皮', '修正书写错误', '文具');
Query OK, 2 rows affected (0.00 sec)
```

从上述执行结果可以看出,返回的结果为 2 rows affected,表示影响了两条记录。

修改完成后,查看数据是否添加成功,具体 SQL 语句及执行结果如下。

```
mysql>SELECT name, content, keyword FROM mydb.my_goods WHERE id=20;
```

```
+------+---------------+----------+
| name   | content          | keyword    |
+------+---------------+----------+
| 橡皮    | 修正书写错误       | 文具        |
+------+---------------+----------+
1 row in set (0.00 sec)
```

从上述执行结果可以看出，成功查询出 id 字段值为 20 的记录，说明数据添加成功。

综上所述，主键冲突更新和主键冲突替换这两种方式都可以解决添加数据时主键冲突的问题。主键冲突替换遇到主键重复会先删除、后新增，适用于添加数据字段特别多的情况。

5.1.3　清空数据

在 MySQL 中，除了可以使用 DELETE 语句删除数据表中的部分数据或全部数据外，还可以通过 TRUNCATE 语句删除数据表中的全部数据，其基本语法如下。

```
TRUNCATE [TABLE] 数据表名称;
```

上述语法中，“数据表名称”用于指定要删除的数据表的名称。

使用 TRUNCATE 语句与使用 DELETE 语句删除数据的操作类似，但是两者存在本质的区别，具体如下。

（1）实现方式不同。TRUNCATE 语句相当于先执行删除数据表（DROP TABLE）的操作，再根据有效的表结构文件（.frm）重新创建数据表，实现数据清空操作；而 DELETE 语句则是逐条地删除数据表中保存的数据。

（2）执行效率不同。在针对大型数据表（如千万级的数据记录）时，从实现方式角度考虑，TRUNCATE 语句比 DELETE 语句删除数据的方式执行效率更好。而当删除的数据量很小时，DELETE 语句的执行效率高于 TRUNCATE 语句。

（3）对设置了 AUTO_INCREMENT 字段的影响不同。使用 TRUNCATE 语句删除数据后，如果字段值设置了 AUTO_INCREMENT，那么再次添加数据时，该字段的值会从默认的初始值重新开始；而使用 DELETE 语句删除数据时，字段值会保持原有的自动增长值。

（4）删除数据的范围不同。TRUNCATE 语句只能用于清空数据表中的全部数据；而 DELETE 语句可以通过 WHERE 子句指定删除满足条件的部分数据。

（5）返回值含义不同。TRUNCATE 语句的返回值一般是无意义的；而 DELETE 语句则会返回符合条件被删除的数据数量。

（6）所属 SQL 语言的组成部分不同。TRUNATE 语句通常被认为是数据定义语言；而 DELETE 语句属于数据操作语言。

下面通过实际操作演示 TRUNCATE 语句和 DELETE 语句的区别。使用 TRUNCATE 语句删除全部数据并重新添加一条数据，具体 SQL 语句及执行结果如下。

```
#删除全部数据
mysql>TRUNCATE TABLE mydb.my_goods;
```

```
Query OK, 0 rows affected (0.06 sec)
#添加数据
mysql>INSERT INTO mydb.my_goods (name, content, keyword)
    ->VALUES ('香蕉', '一种富含钾元素的水果', '水果');
Query OK, 1 row affected (0.01 sec)
#查看数据
mysql>SELECT id, name, content, keyword FROM mydb.my_goods;
+----+------+----------------------+----------+
| id  | name   | content                    | keyword    |
+----+------+----------------------+----------+
| 1    | 香蕉   | 一种富含钾元素的水果       | 水果       |
+----+------+----------------------+----------+
1 row in set (0.00 sec)
```

从上述执行结果可以看出,执行 TRUNCATE 语句后,返回值为 0 rows affected,明显无实际意义。删除数据后,再次新增一条数据,查询到的商品 id 值为 1。

使用 DELETE 语句删除全部数据并重新添加一条数据,查询添加后的结果,具体 SQL 语句及执行结果如下。

```
#删除全部数据
mysql>DELETE FROM mydb.my_goods;
Query OK, 1 row affected (0.01 sec)
#添加数据
mysql>INSERT INTO mydb.my_goods (name, content, keyword)
    ->VALUES ('苹果', '一种很有营养的水果', '水果');
Query OK, 1 row affected (0.01 sec)
#查看数据
mysql>SELECT id, name, content, keyword FROM mydb.my_goods;
+----+------+--------------------+----------+
| id  | name   | content                  | keyword    |
+----+------+--------------------+----------+
| 2    | 苹果   | 一种很有营养的水果       | 水果       |
+----+------+--------------------+----------+
1 row in set (0.00 sec)
```

从上述执行结果可以看出,执行 DELETE 语句后,返回值为 1 row affected,表示有一条记录受影响。删除数据后,再次新增一条数据,查询到的商品 id 值为 2。

5.1.4 去除查询结果中的重复记录

数据表的字段如果没有设置唯一约束或主键约束,那么该字段就有可能存储了重复的值。在实际应用中,有时需要从查询记录中去除重复数据,这时可以使用 SELECT 语句的查询选项 DISTINCT 实现去重查询。带有查询选项的 SELECT 语句的语法格式如下。

```
SELECT [查询选项] 字段名[, …] FROM 数据表名称;
```

上述语法中,查询选项为可选项,取值为 ALL 或 DISTINCT,其中 ALL 为默认值,表示保存所有查询到的记录;当设置为 DISTINCT 时,表示去除重复记录,只保留一条记录。

需要注意的是，当查询的字段有多个时，只有所有字段的值完全相同，才会被认为是重复数据。

下面通过具体操作演示查询选项 DISTINCT 的使用。先查看 sh_goods 表中所有 keyword 字段的值，具体 SQL 语句及执行结果如下。

```
mysql>SELECT keyword FROM sh_goods;
+-----------+
| keyword   |
+-----------+
| 办公      |
| 办公      |
| 办公      |
| 电子产品  |
| 电子产品  |
| 电子产品  |
| 电子产品  |
| 电子产品  |
| 服装      |
| 服装      |
+-----------+
10 rows in set (0.00 sec)
```

从上述执行结果可知，查询出的 keyword 字段值有 3 条为“办公”，5 条为“电子产品”，2 条为“服装”。即使存在重复的数据，默认情况下也会保存所有查询到的记录。

接下来，查看 sh_goods 表中去除重复记录的 keyword 字段值，具体 SQL 语句如下。

```
mysql>SELECT DISTINCT keyword FROM sh_goods;
+-----------+
| keyword   |
+-----------+
| 办公      |
| 电子产品  |
| 服装      |
+-----------+
3 rows in set (0.01 sec)
```

从上述执行结果可以看出，查询结果中仅包含 3 条记录，分别为办公、电子产品和服装，不再包含重复的记录。

5.2　排序和限量

随着电子商务网站的迅速发展，商品的数量越来越多，商品的种类越来越丰富，人们在查看商品列表时，常常会对其进行排序，以便将符合要求的数据显示在前面，方便进一步操作。同时，为了提高执行效率，经常需要对操作的数据进行限量。例如，在查看商品时，只显示 10 条符合要求的记录。本节详细讲解 MySQL 中的排序和限量操作。

5.2.1 排序

在查询数据表中的数据时,如果需要进行排序,可以通过 ORDER BY 来实现。排序可以使数据更有组织性、更容易查找,经过排序整理后的数据便于观察,易于从中发现规律。同样,在我们的学习和工作中,也需要做到有组织、有计划,以便更高效地完成任务。

ORDER BY 排序查询的基本语法格式如下。

```
SELECT * | {字段名[, …]} FROM 数据表名称
ORDER BY 字段名 1 [ASC | DESC][, 字段名 2 [ASC | DESC]]…;
```

上述语法格式中,使用 ORDER BY 进行排序时,如果不指定排序方式,默认按照 ASC(ascending,升序)方式进行排序。排序意味着数据与数据发生比较,需要遵循一定的比较规则,具体规则取决于当前使用的校对集。默认情况下,数字和日期的顺序为从小到大;英文字母的顺序按 ASCII 码的次序,即从 A 到 Z。如果想要降序排序,将 ASC 改为 DESC(descending,降序)即可。

ORDER BY 可以对多个字段的值进行排序,首先按照字段名 1 进行排序,当字段名 1 的值相同时,再按照字段名 2 进行排序,以此类推。ORDER BY 后面也可以跟表达式。

需要说明的是,按照指定字段进行排序时,如果指定字段中包含 NULL,NULL 会被当作最小值进行排序。

下面演示查询 sh_goods 表中的数据,让数据在显示时首先按商品分类(category_id 字段)升序排序,然后再按商品价格(price 字段)降序排序,具体 SQL 语句及执行结果如下。

```
mysql>SELECT category_id, keyword, name, price FROM sh_goods
    ->ORDER BY category_id, price DESC;
+-------------+----------+------------------------+----------+
| category_id | keyword  | name                   | price    |
+-------------+----------+------------------------+----------+
|           3 | 办公     | 钢笔 T1616             |    15.00 |
|           3 | 办公     | 碳素笔 GP1008          |     1.00 |
|           3 | 办公     | 2H 铅笔 S30804         |     0.50 |
|           6 | 电子产品 | 华为 P50 智能手机      |  1999.00 |
|           8 | 电子产品 | 桌面音箱 BMS10         |    69.00 |
|           9 | 电子产品 | 头戴耳机 Star Y360     |   109.00 |
|          11 | 电子产品 | 办公计算机 天逸 510Pro |  2000.00 |
|          12 | 电子产品 | 超薄笔记本 Pro12       |  5999.00 |
|          15 | 服装     | 收腰风衣中长款         |   299.00 |
|          16 | 服装     | 薄毛衣联名款           |    48.00 |
+-------------+----------+------------------------+----------+
10 rows in set (0.00 sec)
```

从上述执行结果可以看出,查询的所有数据先按 category_id 字段升序排序,相同 category_id 值的记录再按照 price 字段降序排序。此外,由于 sh_goods 数据表的字符集是 utf8mb4,当排序的字段为中文时,默认不会按照中文拼音的顺序排序。在不改变数据表结构的情况下,若要强制字段按中文首字母排序,可以使用“CONVERT(字段名 USING gbk)”函数将字段的字符集指定为 gbk。

下面演示如何查询 sh_goods 表中的数据,按照 keyword 关键词字段进行降序排序,具

体 SQL 语句及执行结果如下。

```
mysql>SELECT category_id, keyword, name, price FROM sh_goods
    ->ORDER BY CONVERT(keyword USING gbk) DESC;
+-------------+-----------+------------------------+----------+
| category_id | keyword   | name                   | price    |
+-------------+-----------+------------------------+----------+
| 15          | 服装      | 收腰风衣中长款         |   299.00 |
| 16          | 服装      | 薄毛衣联名款           |    48.00 |
| 12          | 电子产品  | 超薄笔记本 Pro12       |  5999.00 |
| 6           | 电子产品  | 华为 P50 智能手机      |  1999.00 |
| 8           | 电子产品  | 桌面音箱 BMS10         |    69.00 |
| 9           | 电子产品  | 头戴耳机 Star Y360     |   109.00 |
| 11          | 电子产品  | 办公计算机 天逸 510Pro |  2000.00 |
| 3           | 办公      | 2H 铅笔 S30804         |     0.50 |
| 3           | 办公      | 钢笔 T1616             |    15.00 |
| 3           | 办公      | 碳素笔 GP1008          |     1.00 |
+-------------+-----------+------------------------+----------+
10 rows in set (0.00 sec)
```

上述 SELECT 语句中，按照 keyword 字段值的中文首字母进行降序排序。

多学一招：按指定顺序排序

前面使用 ORDER BY 实现了对字段的升序和降序排序，如果想要对 sh_goods 表中 keyword 字段的查询结果集进行指定顺序排序，则可以借助 FIELD()函数来实现。使用 FIELD()函数查询排序结果的语法格式如下。

```
SELECT * |{字段名[, …]} FROM 数据表名称
ORDER BY FIELD(value, str1, str2, str3, …);
```

上述语法格式表示将获取到的 value 字段，按照“str1，str2，str3”的顺序进行排序，其中 str1、str2、str3 属于查询字段 value 的结果集中的内容。value 参数后面的参数可自定义，不限制参数个数。

下面演示如何查询 sh_goods 表中的数据，将关键词 keyword 字段按照“办公，服装，电子产品”排序，具体 SQL 语句及执行结果如下。

```
mysql>SELECT category_id, keyword, name, price FROM sh_goods
    ->ORDER BY FIELD(keyword, '办公', '服装', '电子产品');
+-------------+-----------+------------------------+----------+
| category_id | keyword   | name                   | price    |
+-------------+-----------+------------------------+----------+
| 3           | 办公      | 2H 铅笔 S30804         |     0.50 |
| 3           | 办公      | 钢笔 T1616             |    15.00 |
| 3           | 办公      | 碳素笔 GP1008          |     1.00 |
| 15          | 服装      | 收腰风衣中长款         |   299.00 |
| 16          | 服装      | 薄毛衣联名款           |    48.00 |
| 12          | 电子产品  | 超薄笔记本 Pro12       |  5999.00 |
| 6           | 电子产品  | 华为 P50 智能手机      |  1999.00 |
| 8           | 电子产品  | 桌面音箱 BMS10         |    69.00 |
| 9           | 电子产品  | 头戴耳机 Star Y360     |   109.00 |
| 11          | 电子产品  | 办公计算机 天逸 510Pro |  2000.00 |
```