

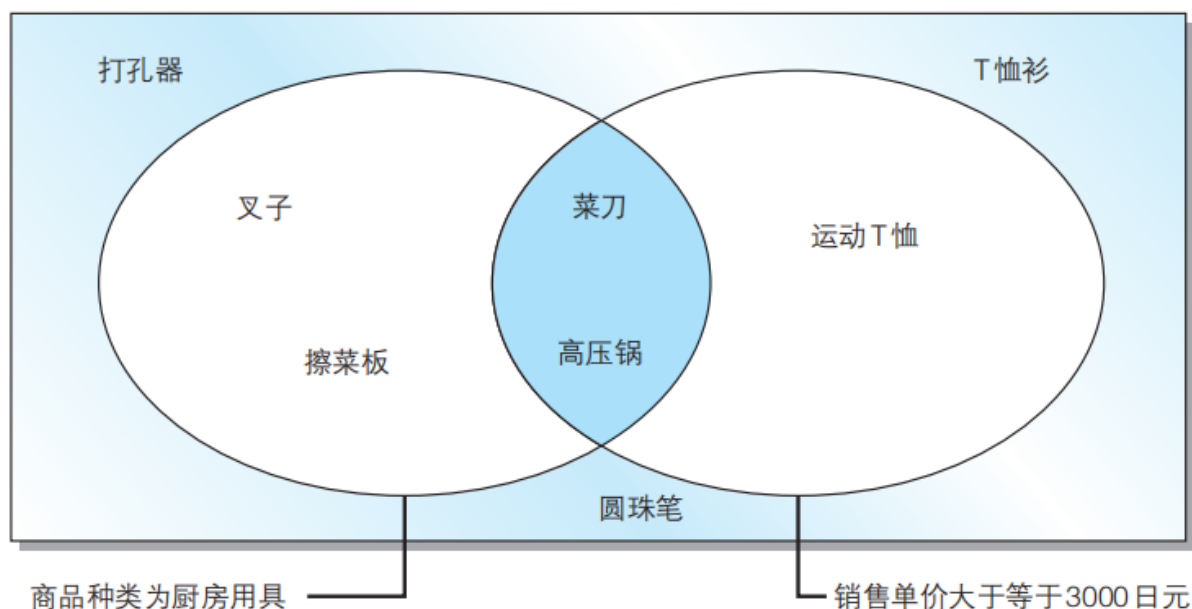
第三章 基础查询与排序

1. SELECT语句基础

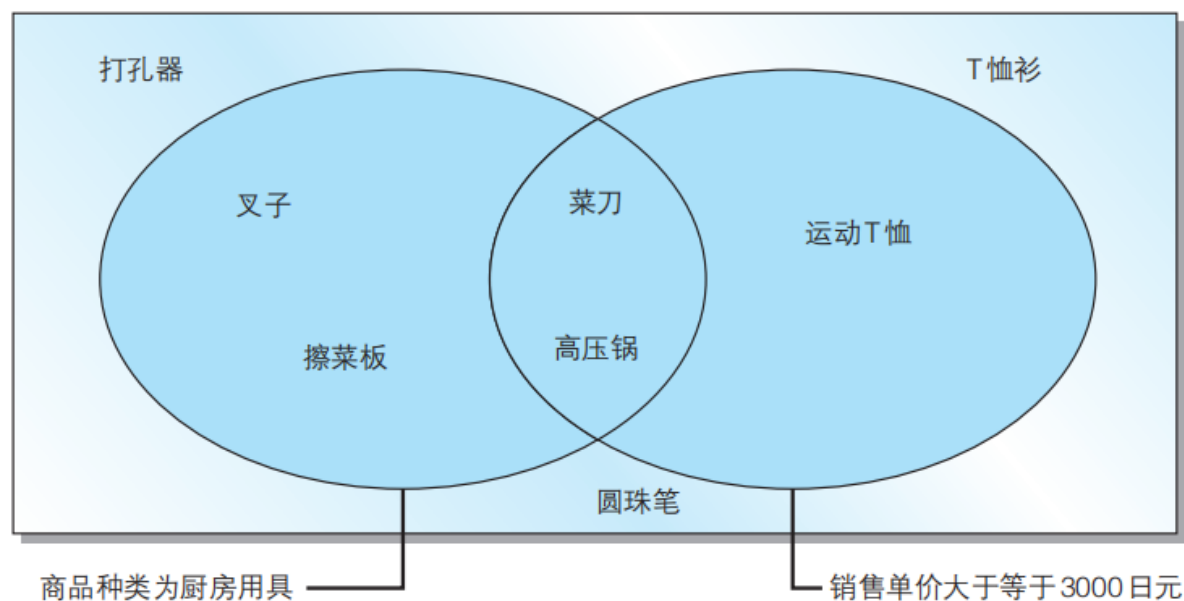
基本语法

```
SELECT <列名>, .....  
FROM <表名>  
WHERE <条件表达式>;
```

- DISTINCT 去重复
- as 设置别名
- NOT 不能单独使用，必须和其他查询条件组合起来使用
- AND 相当于“并且”，类似数学中的取交集；（AND 运算符优先于 OR 运算符）



- OR 相当于“或者”，类似数学中的取并集。



- 真值表
 - AND 运算符两侧的真值都为真时返回真，除此之外都返回假。
 - OR 运算符两侧的真值只要有一个不为假就返回真，只有当其两侧的真值都为假时才返回假。
 - NOT运算符只是单纯的将真转换为假，将假转换为真。

| AND | | |
|-----|---|---------|
| P | Q | P AND Q |
| 真 | 真 | 真 |
| 真 | 假 | 假 |
| 假 | 真 | 假 |
| 假 | 假 | 假 |

| OR | | |
|----|---|--------|
| P | Q | P OR Q |
| 真 | 真 | 真 |
| 真 | 假 | 真 |
| 假 | 真 | 真 |
| 假 | 假 | 假 |

| NOT | |
|-----|-------|
| P | NOT P |
| 真 | 假 |
| 假 | 真 |

P AND (Q OR R)

| P | Q | R | Q OR R | P AND (Q OR R) |
|---|---|---|--------|----------------|
| 真 | 真 | 真 | 真 | 真 |
| 真 | 真 | 假 | 真 | 真 |
| 真 | 假 | 真 | 真 | 真 |
| 真 | 假 | 假 | 假 | 假 |
| 假 | 真 | 真 | 真 | 假 |
| 假 | 真 | 假 | 真 | 假 |
| 假 | 假 | 真 | 真 | 假 |
| 假 | 假 | 假 | 假 | 假 |

P: 商品种类为办公用品
 Q: 登记日期是2009年9月11日
 R: 登记日期是2009年9月20日
 Q OR R: 登记日期是2009年9月11日或者2009年9月20日
 P AND (Q OR R): 商品种类为办公用品, 并且, 登记日期是2009年9月11日或者2009年9月20日

- 含有NULL时的真值处理（真假值排序为：真、不确定、假）：可以理解为 and 取低值，or 取高值

| P | Q | P AND Q |
|-----|-----|---------|
| 真 | 真 | 真 |
| 真 | 假 | 假 |
| 真 | 不确定 | 不确定 |
| 假 | 真 | 假 |
| 假 | 假 | 假 |
| 假 | 不确定 | 假 |
| 不确定 | 真 | 不确定 |
| 不确定 | 假 | 假 |
| 不确定 | 不确定 | 不确定 |

| P | Q | P OR Q |
|-----|-----|--------|
| 真 | 真 | 真 |
| 真 | 假 | 真 |
| 真 | 不确定 | 真 |
| 假 | 真 | 真 |
| 假 | 假 | 假 |
| 假 | 不确定 | 不确定 |
| 不确定 | 真 | 真 |
| 不确定 | 假 | 不确定 |
| 不确定 | 不确定 | 不确定 |

SQL中用于汇总的函数叫做聚合函数。以下五个是最常用的聚合函数：

- ### 3.分组 (GROUP BY)

- 聚合键中包含NULL时会将NULL作为一组特殊数据进行聚合运算

- GROUP BY的位置

其中前三项用于筛选数据，GROUP BY对筛选出的数据进行处理

- ## 4.排序 (ORDER BY)

基本语法：

```
SELECT <列名1>, <列名2>, <列名3>, .....  
FROM <表名>  
ORDER BY <排序基准列1> [ASC, DESC], <排序基准列2> [ASC, DESC], .....
```

- ASC 表示升序排列，默认为升序
- DESC 表示降序排列，
- 由于 NULL 无法使用比较运算符进行比较，也就是说，无法与文本类型，数字类型，日期类型等进行比较，当排序列存在 NULL 值时，NULL 结果会展示在查询结果的开头或者末尾。
- 在MySQL中，NULL 值被认为比任何非NULL 值低，因此，当顺序为 ASC（升序）时，NULL 值出现在第一位，而当顺序为 DESC（降序）时，则排序在最后。
- GROUP BY 子句中不能使用SELECT 子句中定义的别名，但是在 ORDER BY 子句中却可以使用别名原因是 SQL 在使用 HAVING 子句时 SELECT 语句的执行顺序为：

```
FROM → WHERE → GROUP BY → SELECT → HAVING → ORDER BY
```

其中 SELECT 的执行顺序在 GROUP BY 子句之后，ORDER BY 子句之前。

当在 ORDER BY 子句中使用别名时，已经知道了 SELECT 子句设置的别名，但是在 GROUP BY 子句执行时还不知道别名的存在，所以在 ORDER BY 子句中可以使用别名，但是在GROUP BY中不能使用别名。

练习1：编写一条SQL语句，从 product(商品) 表中选取出“登记日期(regist_date)在2009年4月28日之后”的商品，查询结果要包含 product name 和 regist_date 两列。

```
select product_name,regist_date  
from product  
where Date(regist_date)>'2009-04-28';
```

日期比较大小特别需要注意日期格式，参考文档：https://blog.csdn.net/qq_28869233/article/details/88831824

练习2: SELECT语句能够从 product 表中取出“销售单价（sale_price）比进货单价（purchase_price）高出500日元以上”的商品。请写出两条可以得到相同结果的SELECT语句。执行结果如下所示：

| product_name | sale_price | purchase_price |
|--------------|------------|----------------|
| T恤衫 | 1000 | 500 |
| 运动T恤 | 4000 | 2800 |
| 高压锅 | 6800 | 5000 |

参考答案：

```
select product_name,sale_price,purchase_price
from product
where sale_price - purchase_price >= 500;
```

练习3: 请写出一条SELECT语句, 从 product 表中选取出满足“销售单价打九折之后利润高于 100 日元的办公用品和厨房用具”条件的记录。查询结果要包括 product_name列、product_type 列以及销售单价打九折之后的利润 (别名设定为 profit) 。

提示: 销售单价打九折, 可以通过 sale_price 列的值乘以0.9获得, 利润可以通过该值减去 purchase_price 列的值获得。

```
select
    product_name,
    product_type,
    sale_price*0.9-purchase_price as profit
from product
where sale_price*0.9-purchase_price >= 100 and (product_type in ('办公用品','厨房用具'));
```

练习4:请编写一条SELECT语句, 求出销售单价 (sale_price 列) 合计值大于进货单价 (purchase_price 列) 合计值1.5倍的商品种类。执行结果如下所示。

| product_type | sum | sum |
|--------------|------|------|
| 衣服 | 5000 | 3300 |
| 办公用品 | 600 | 320 |

```
select
    product_type,
    sum(sale_price) as ssp,
    sum(purchase_price) as spp
from product
group by product_type
having ssp > spp * 1.5;
```

练习6:此前我们曾经使用SELECT语句选取出product (商品) 表中的全部记录。当时我们使用了ORDER BY 子句来指定排列顺序, 但现在已经无法记起当时如何指定的了。请根据下列执行结果, 思考ORDER BY 子句的内容。

```
SELECT *
FROM product
ORDER BY COALESCE(regist_date, 'zzzz') DESC , sale_price;
```