一、语法与分类

```
#进阶2：条件查询
/*
语法：select 查询列表 from 表名 where 筛选条件；

分类：
1.按条件表达式筛选
条件运算符：> < = != <> >= <=

2.按逻辑表达式筛选
作用：用于连接条件表达式
逻辑运算符：&&/||/!/and/or/not
&&和and：两个条件都为true，结果为true，反之为false
||或 or：只要有一个为true，结果为true，反之为false
! 或not：如果连接的条件本身为false，结果为true，反之为false

3.模糊查询
like/between and/in/is null
```

二、具体分类之例

```
#1.按条件表达式筛选
#案例一：查询工资>12000的员工信息
SELECT * FROM employees WHERE salary > 12000;
#案例二：查询部门编号不等于90号的员工名和部门编号
SELECT `last_name`,`department_id` FROM employees WHERE department_id <>90;

#2.按逻辑表达式筛选
#案例一：查询工资在10000到20000之间的员工名、工资以及奖金
SELECT last_name,salary,commission_pct
FROM
employees
WHERE
salary >= 10000 AND salary <=20000;
```

```
/*3.模糊查询like/between and/in/is null
⊙like特点：
>一般和通配符搭配使用
>通配符：%任意多个字符，包含0个字符
        _任意单个字符

⊙between and注意事项：
>提高语句的简洁度
>包含临界值
>两个临界值不要调换顺序

⊙in
>含义：用于判断某字段的值是否属于in列表中的某一项
>特点：使用in提高语句简洁度：
        in列表的值类型必须统一或兼容：
        不支持通配符

⊙is null
>符号=或<>不能用于判断null值
>is null或is not null可以用于判断null值
*/
```

```
#⊙like
#案例一：查询员工名中包含字符a的员工信息
SELECT * FROM employees WHERE last_name LIKE '%a%';
#案例二：查询员工名中第三个字符为n，第五个字符为l的员工名和工资
SELECT last_name,salary FROM employees WHERE last_name LIKE '__n_l%';
#案例三：查询员工名中第二个字符为_的员工名
#加转义字符\或者任意指定转义字符
#SELECT last_name FROM employees WHERE last_name LIKE '_\_%';
SELECT last_name FROM employees WHERE last_name LIKE '_$_%' ESCAPE '$';

#⊙between and
#案例1：查询员工编号在100到120之间的员工信息
#select * from employees where employee_id >= 100 and employee_id <=120;
SELECT * FROM employees WHERE employee_id BETWEEN 100 AND 200;

#⊙in
#案例：查询员工的工种编号是IT_PROG\AD_VP\AD_PRES中一个的员工名和工种编号
SELECT last_name,job_id FROM employees WHERE job_id IN ('IT_PROG','AD_VP','AD_PRES');

#⊙is null
#案例一：查询没有奖金的员工名和奖金率
SELECT last_name,commission_pct FROM employees WHERE commission_pct IS NOT NULL;
```

```
#安全等于<=>
#案例一：查询没有奖金的员工名和奖金率
SELECT last_name,commission_pct FROM employees WHERE commission_pct <=> NULL;

#案例二：查询工资为12000的员工名和奖金率
SELECT last_name,commission_pct,salary FROM employees WHERE salary <=> 12000;

/*is null和<=>对比
is null：仅仅可以判断null值，可读性较高，建议使用
<=>：既可以判断null值，又可以判断普通的数值，但可读性较低
*/
```