DQL

数据查询语言。用来查询数据库中表的记录（数据）。

语法介绍

```
# 语法介绍
SELECT
    字段列表
FROM
    表名
WHERE
    条件列表
GROUP BY
    分组字段
HAVING
    分组后的过滤条件
ORDER BY
    排序
LIMIT
    分页
```

查询全部

```
## 查询全部数据
SELECT * FROM 表名;
SELECT * FROM product;

## 查询指定字段的表数据
SELECT 列名1,列名2,... FROM 表名;
SELECT NAME,price,brand FROM product;


## 去掉重复查询 distinct;
SELECT DISTINCT 列名1,列名2,...FROM 表名;
SELECT brand FROM product;
SELECT DISTINCT brand FROM product;


## 计算列的值(四则运算)
SELECT 列名1 运算符(+-*/)列名2 FROM 表名;
SELECT NAME , price + 1000 AS price , brand FROM product;
## 如果某一列的值为null，可以进行替换
# ifnull(表达式1,表达式2) 函数
# 表达式1: 想替换的列
# 表达式2: 想替换的值
SELECT NAME , IFNULL(price,0) + 1000 AS price , brand FROM product;
```

```
## 起别名查询 as
SELECT 列名 AS 别名 FROM 表名;
SELECT p.name AS p_name ,p.price AS p_price ,p.brand AS p_brand FROM product p;
```

条件查询

符合	功能
>	大于
<	小于
>=	大于等于
<=	小于等于
=	等于
<> 或 !=	不等于
between ... and ...	在某个范围之内（都包含）
IN(...)	多选一
LIKE 占位符	模糊查询 _单个任意字符 % 多个任意字符
IS NULL	是null
IS NOT NULL	不是null
AND 或 &&	并且
OR 或 ||	或者
NOT 或 !	非， 不是

```
# 条件查询
SELECT 列名列表 FROM 表名 WHERE 条件;

## 查询 价格大于5000 的商品信息
SELECT * FROM product WHERE price > 5000;

## 查询品牌为华为的商品信息
SELECT * FROM product WHERE brand = '华为';

## 查询价格在2999 ~ 5999 之间的商品信息
SELECT * FROM product WHERE price BETWEEN 2999 AND 5999;
SELECT * FROM product WHERE price >=2999 AND price <=5999;


## 查询价格为3999,4999 , 5999 的商品信息
SELECT * FROM product WHERE price IN (3999,4999,5999);
```

```
## 查询价格不为null 的商品信息
SELECT * FROM product WHERE price IS NOT NULL;
SELECT * FROM product WHERE price IS NULL;


## 查询品牌以 '小' 为开头的商品信息    模糊查询 like
SELECT * FROM product WHERE brand LIKE '小%';

## 查询品牌第二个字是'为'的商品信息
SELECT * FROM product WHERE brand LIKE '_为%';

## 查询名称为3个字符的商品信息
SELECT * FROM product WHERE NAME LIKE '___';

## 查询名称中包含手机的商品信息
SELECT * FROM product WHERE NAME LIKE '%手机%';
```

聚合函数查询

函数名	功能
count(列名)	统计数量（一般选用不为null的列）
max(列名)	最大值
min(列名)	最小值
sum(列名)	求和
avg(列名)	平均值

```
## 聚合查询的语法
SELECT 函数名称(列名) FROM [WHERE 条件]

## 计算procduct 表中总记录数据
SELECT COUNT(*) FROM product;

## 获取最高价格
SELECT MAX(price) FROM product;

## 获取最低价格
SELECT MIN(price) FROM product;

## 获取总库存数量
SELECT SUM(stock) FROM product;

## 获取品牌为苹果的总库存数量
SELECT SUM(stock) FROM product WHERE brand = '苹果';
```

```
## 获取品牌为小米的平均商品价格
SELECT AVG(price) FROM product WHERE brand = '小米';
```

排序查询

```
# 排序查询

## 语法
SELECT 列名列表 FROM 表名 [WHERE 条件] ORDER BY 列名 排序方式, 列名 排序方式...;

## 排序方式 ASC: 升序(默认), DESC : 降序
## 如果有多个排序条件，只有当前面的条件值一样时，才会判断第二个条件;

SELECT * FROM product ORDER BY price DESC;

## 查询品牌包含华为的商品信息，并且 降序排序
SELECT * FROM product WHERE brand LIKE '%华为%' ORDER BY price DESC;

## 按照库存升序排序，如果库存一样，按照价格降序排序
SELECT * FROM product ORDER BY stock , price DESC;
```

分组查询

```
# 分组查询
## 语法
SELECT 列名 FROM 表名
    [WHERE 条件]
    GROUP BY 分组列名
    [HAVING 分组后条件过滤]
    [ORDER BY 排序列 排序方式]


## 按照品牌分组, 获取每组商品的总金额
SELECT brand , SUM(price) AS getSum FROM product GROUP BY brand;

## 对price大于3000的商品 按照品牌分组, 获取每组商品的总金额
SELECT brand, SUM(price) AS getSu FROM product WHERE price > 3000 GROUP BY
brand;


## 对price大于3000的商品 按照品牌分组, 获取每组商品的总金额 , 只显示总金额>5000的商品
SELECT brand , SUM(price) AS getSum FROM product
    WHERE price > 3000
    GROUP BY brand
    HAVING getSum > 5000;

## 对price大于3000的商品 按照品牌分组, 获取每组商品的总金额 , 只显示总金额>8000的商品
SELECT brand , SUM(price) AS getSum FROM product
    WHERE price > 3000
    GROUP BY brand
```

```
    HAVING getSum > 5000
    ORDER BY getSum DESC;
```

分页查询

```
# 分页查询
## 语法
    SELECT 列表 FROM 表名
        [WHERE 条件]
        [GROUP BY 分组列名]
        [HAVING 分组后的条件过滤]
        [ORDER BY 排序列名 排序方式]
        LIMIT 开始数据,每页显示的条数;

        开始数据 = (当前页数 - 1) * 每页显示的条数

# 每页显示3条数据

# 第1页   开始数据 = (1-1) * 3 = 0
SELECT * FROM product LIMIT 0,3;

# 第2页   开始数据 = (2-1) * 3 = 3
SELECT * FROM product LIMIT 3,3;

# 第3页   开始数据 = (3-1) * 3 = 6
SELECT * FROM product LIMIT 6,3;
```