

SQL-DQL

SQL-DQL

- DQL-语法

- DQL-基本查询

 - 查询多个字段

 - 设置别名

 - 案例

 - 去除重复记录

- DQL-条件查询

 - 语法

 - 条件

 - 案例

- DQL-聚合函数

 - 介绍

 - 常见聚合函数

 - 语法

- DQL-分组查询

 - 语法

 - WHERE与HAVING的区别

 - 案例

 - 注意

- DQL-排序查询

 - 语法

 - 排序方式

 - 案例

 - 注意

- DQL-分页查询

 - 语法

 - 注意

 - 案例

- DQL-执行顺序

 - 编写顺序

DQL-语法

```
1  SELECT
2      字段列表
3  FROM
4      表名列表
5  WHERE
6      条件列表
7  GROUP BY
8      分组字段列表
9  HAVING
10     分组后条件列表
11 ORDER BY
12     排序字段列表
13 LIMIT
14     分页参数
```

- 基本查询
- 条件查询(WHERE)
- 聚合查询(count、max、min、avg、sum)
- 分组查询(GROUP BY)
- 排序查询(ORDER BY)
- 分页查询(LIMIT)

DQL-基本查询

查询多个字段

```
1  SELECT 字段1, 字段2, 字段3... FROM 表名;
```

```
1  SELECT * FROM 表名;
```

设置别名

```
1 SELECT 字段1[AS 别名1],字段2[AS 别名2] ... FROM 表名;
```

案例

```
1 SELECT workaddress [AS] '工作地址' FROM emp;
```

去除重复记录

```
1 SELECT DISTINCT 字段列表 FROM 表名;
```

DQL-条件查询

语法

```
1 SELECT 字段列表 FROM 表名 WHERE 条件列表;
```

条件

比较运算符	功能
>	大于
>=	大于等于
<	小于
<=	小于等于
<> 或 !=	不等于
BETWEEN...AND...	在某个范围之内（含最小、最大值）
IN(...)	在IN之后的列表中的值，多选一
LIKE	模糊匹配（_匹配单个字符，%匹配任意个字符）
IS NULL	是NULL

逻辑运算符	功能
AND 或 &&	并且（多个条件同时成立）
OR 或	或者（多个条件任意一个成立）
NOT 或 !	非，不是

案例

查询有身份证的员工信息

```
1 SELECT * FROM emp WHERE idcard IS NOT NULL;
```

查询年龄在15岁到20岁（包含）之间的员工信息

```
1 SELECT * FROM emp WHERE age BETWEEN 15 AND 20;
```

查询年龄等于18或20或40的员工信息

```
1 SELECT * FROM emp WHERE age IN (18, 20, 40);
```

查询姓名为两个字的员工信息

```
1 SELECT * FROM emp WHERE name LIKE '--';
```

查询身份证号最后一位是X的员工信息

```
1 SELECT * FROM emp WHERE idcard LIKE '%X';
```

DQL-聚合函数

介绍

将一系列数据作为一个整体，进行纵向计算

常见聚合函数

函数	功能
count	统计数量
max	最大值
min	最小值
avg	平均值
sum	求和

语法

```
1 SELECT 聚合函数(字段列表) FROM 表名;
```

DQL-分组查询

语法

```
1 SELECT 字段列表 FROM 表名 [WHERE 条件] GROUP BY 分组字段名 [HAVING 分组后过滤条件];
```

WHERE与HAVING的区别

- 执行时机不同：WHERE是分组之前进行过滤，不满足WHERE条件，不参与分组；而HAVING是分组之后对结果进行过滤。
- 判断条件不同：WHERE不能对聚合函数进行判断，而HAVING可以。

案例

根据性别分组，统计男性员工和女性员工数量。

```
1 SELECT gender, COUNT(*) FROM emp GROUP BY gender;
```

查询年龄小于45的员工，并根据工作地址分组，获取员工数量大于等于3的工作地址。

```
1 SELECT workaddress, COUNT(*) AS address_count FROM emp WHERE age
  < 45 GROUP BY workaddress HAVING address_count >= 3;
```

注意

- 执行顺序：WHERE > 聚合函数 > HAVING。
- 分组之后，查询的字段一般为聚合函数和分组字段，查询其他字段无任何意义。

DQL-排序查询

语法

```
1 SELECT 字段列表 FROM 表名 ORDER BY 字段1 排序方式1, 字段2 排序方式2;
```

排序方式

- ASC 升序
- DESC 降序

案例

根据年龄对公司的员工进行升序排序，年龄相同，再按照入职时间进行降序排序

```
1 SELECT * FROM emp ORDER BY age ASC, entrydate DESC;
```

注意

- 如果是多字段排序，当第一个字段值相同时，才会根据第二个字段进行排序。

DQL-分页查询

语法

```
1 SELECT 字段列表 FROM 表名 LIMIT 起始索引, 查询记录数;
```

注意

- 起始索引从0开始，起始索引 = (查询页码 - 1) * 每页显示记录数。
- 分页查询是数据库的方言，不同的数据库有不同的实现，MySQL中是LIMIT。
- 如果查询的是第一页数据，起始索引可以省略，直接简写为limit 10。

案例

查询第二页员工数据，每页展示10条记录

```
1 SELECT * FROM emp LIMIT 1, 10;
```

查询年龄为20, 21, 22, 23岁的女性员工信息

```
1 SELECT * FROM emp WHERE gender = '女' AND age IN (20, 21, 22, 23);
```

查询性别为男，并且年龄在20 - 40岁（含）以内的姓名为三个字的员工

```
1 SELECT * FROM emp WHERE gender = '男' AND (age BETWEEN 20 AND 40) AND LIKE '___';
```

统计员工表中，年龄小于60岁的，男性员工和女性员工的人数

```
1 SELECT gender, COUNT(*) FROM emp WHERE age < 60 GROUP BY gender;
```

查询所有年龄小于等于35岁员工的姓名和年龄，并对查询结果年龄升序排序，年龄相同按入职时间降序排序

```
1 SELECT name, age FROM emp WHERE age <= 35 ORDER BY age ASC,
  entrydate DESC;
```

查询性别为男，且年龄在20 - 40岁（含）以内的前5个员工信息，对查询的结果按年龄升序排序，年龄相同按入职时间升序排序

```
1 SELECT * FROM emp WHERE gender = '男' AND (age BETWEEN 20 AND 40)
  ORDER BY age ASC, entrydate ASC LIMIT 0, 5;
```

DQL-执行顺序

编写顺序

```
1 SELECT 字段列表 FROM 表名列表 WHERE 条件列表 GROUP BY 分组字段列表
  HAVING 分组后条件列表 ORDER BY 排序字段列表 LIMIT 分页参数;
```

执行顺序

```
1 FROM 表名列表 WHERE 条件列表 GROUP BY 分组字段列表 HAVING 分组后条件列表
  SELECT 字段列表 ORDER BY 排序字段列表 LIMIT 分页参数;
```

[返回文首](#)

