目标:高级查询

1. 模糊查询
2. 聚合函数和分组查询
3. 子查询
4. 多表连接查询
5. 使用Like关键字进行模糊查询
6. 用法:

Where 字段 like ‘含有通配符的值’

1. 通配符

\_ 匹配一个字符

% 匹配任意字符(0到无数个)

[可选字符] mysql暂不支持

|  |
| --- |
| 示例:  查询姓名是两个字的学生信息  SELECT \* FROM students WHERE NAME LIKE '\_\_';  查询姓名是两个字且以王开头的学生信息  SELECT \* FROM students WHERE NAME LIKE '王\_';  查询姓名中含有'明'的学生信息  SELECT \* FROM students WHERE NAME LIKE '%明%'; |

1. 使用聚合函数和分组查询
2. 聚合函数和分组查询用于”统计查询”
3. 常用聚合函数

sum(列) 统计某列总和

avg(列) 统计某列平均值

max(列) 统计某列最大值

min(列) 统计某列最小值

count(列|\*) 统计总行数

|  |
| --- |
| 示例:  统计学生的平均年龄  SELECT AVG(age) AS 平均年龄 FROM students;  统计学生年龄总和  SELECT SUM(age) AS 年龄总和 FROM students;  统计学生年龄最大  SELECT MAX(age) AS 年龄最大 FROM students;  统计学生年龄最小  SELECT MIN(age) AS 年龄最小 FROM students;  统计学生总人数  SELECT COUNT(xh) AS 总行数 FROM students;  统计所有男生的平均年龄，最大年龄，总人数  SELECT AVG(age) 平均年龄,MAX(age) 最老的,COUNT(1) AS 总人数 FROM students WHERE sex='男'; |

1. 分组查询
2. 查询语法:

select \* from 表名 [where 子句] [group by分组子句] [order by 子句]

1. 分组子句的语法

Group by 列1[，列2] [having 子句]

注意：

1. 使用分组语句后，查询的列需要出现在分组子句中或者在聚合函数
2. Having子句是针对分组以后的结果进行筛选

|  |
| --- |
| 示例:  统计男女生人数  SELECT sex,COUNT(1) FROM students GROUP BY sex  统计各年级学生人数  SELECT gid,COUNT(1)FROM students GROUP BY gid;  统计学生年龄大于20岁的各年级学生人数,并按人数降序排序  SELECT gid,COUNT(1) AS 人数 FROM students WHERE age>20 GROUP BY gid ORDER BY 人数 DESC;  统计各年级男女人数 -- 多列分组  SELECT gid,sex,COUNT(1) FROM students GROUP BY gid,sex;  统计各年级总人数大于3的年级  SELECT gid,COUNT(1) FROM students GROUP BY gid HAVING COUNT(1)>3 |

1. Where子句与having子句的区别

Where子句是表中的行进行筛选

Having子句是对分组后的结果进行筛选

1. 子查询

1.什么是子查询：

在查询语句中再嵌套另外一个查询则称为子查询.

注意: 子查询需要用小括号包起来

2.子查询的作用

一般情况下子查询作为父查询的条件出现(子查询返回单值)

子查询替换连接查询

|  |
| --- |
| 示例:  查询学生年龄大于平均年龄的学生信息  SELECT \* FROM students WHERE age>(SELECT AVG(age) FROM students);  查询“年级二”下所有的学生  SELECT \* FROM students WHERE gid=(SELECT gid FROM grade WHERE gname='年级二');  查询“年级一\年级二”下所有的学生  -- SELECT \* FROM students WHERE gid=(SELECT gid FROM grade WHERE gname='年级二')  -- or gid=(SELECT gid FROM grade WHERE gname='年级一')  SELECT \* FROM students WHERE gid IN(SELECT gid FROM grade WHERE gname='年级二' OR gname='年级一') |

1. 多表连接查询
2. 作用:连接查询多张表的综合数据

2.连接查询分类

内连接inner join、 外连接[左外连接left join、右外连接right JOIN]

内连接：表示左右表同时存在

左外连接:表示以左表为主关联右表，关联不到补空值

右外连接:表示以右表为主关联左表，关联不到补空值

3.连接查询的语法:

SELECT \* FROM 表1 INNER JOIN 表2 ON 表1.公共字段=表2.公共字段

[WHERE 子句] [分组子句] [排序子句]

|  |
| --- |
| 示例：  查询所有的学生信息及对应的年级信息  SELECT \* FROM students INNER JOIN grade ON students.gid=grade.`gid`;  SELECT students.\*,grade.\* FROM students INNER JOIN grade ON students.gid=grade.`gid`;  查询学生的学号，姓名，年级编号，年级名称  SELECT xh,NAME,s.gid,gname FROM students AS s INNER JOIN grade AS g ON s.gid=g.`gid`;  -- 理解外连接  SELECT \* FROM students;  SELECT \* FROM students RIGHT JOIN grade ON students.gid=grade.`gid`; |