# DQL

基础查询

条件查询

模糊查询

排序查询

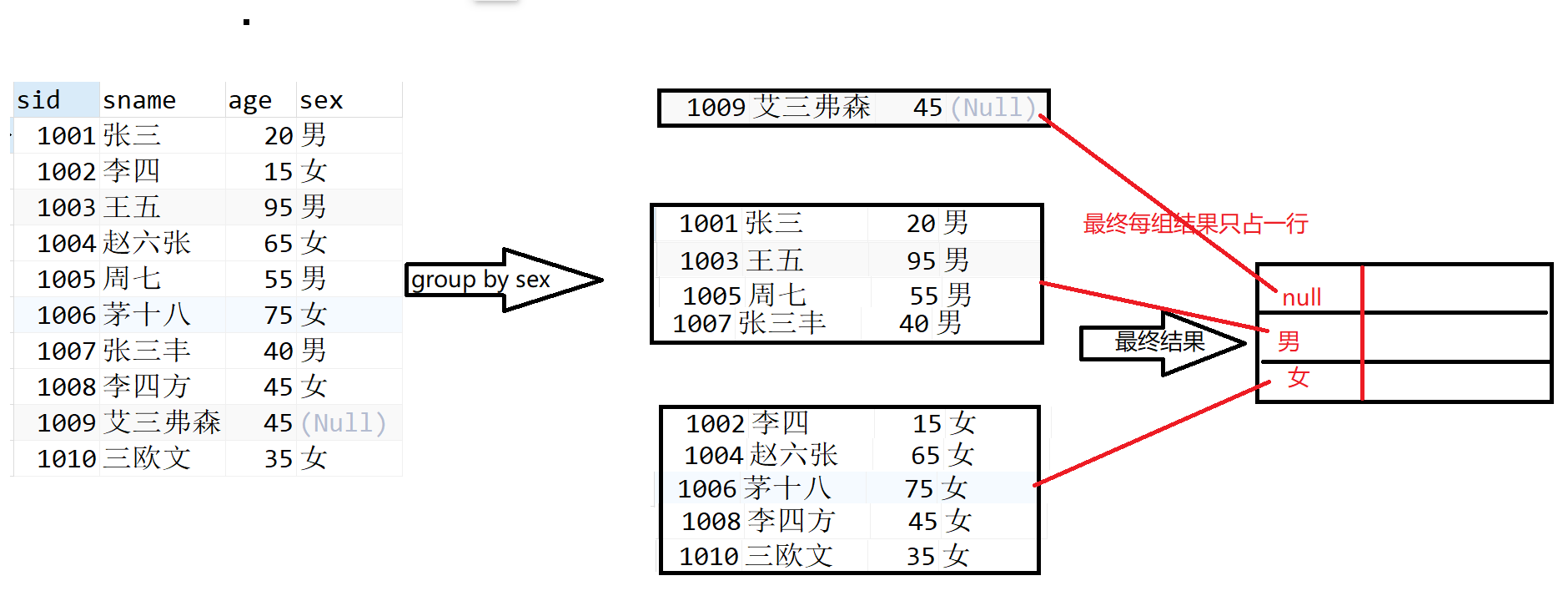
## 聚合函数

|  |
| --- |
| /\*  聚合函数  count(列名) 统计指定列不为null的行数  max(列名) 获得指定列中最大值,如果是字符串,按照字符顺序  min(列名) 获得最小  sum(列名) 计算指定列的和值,计算非数字,结果是0  avg(列名) 计算指定列的平均值,计算非数字,结果是0  ---------------  语法:  select 聚合函数 from 表名 [where.. ]  \*/  -- 查询stu表中记录数：  select count(sid) from stu;  update stu set score = null where sid = 1001;  select count(sex) from stu;  -- 在列名后使用as 给列取别名,as可以省略  select count(sid) as '记录数' from stu;  select count(sid) '记录数' from stu;  -- 查询stu表中有成绩的人数：  select count(score) '人数' from stu;  -- 查询stu表中成绩大于60的人数：  select count(\*) from stu where score > 60;  -- 查询所有学生成绩和：  select sum(score) from stu;  select sum(score) '总成绩' from stu;  -- 统计所有学生平均成绩  select avg(score) from stu;  -- 统计stu表中成绩大于60的平均分  select avg(score) from stu where score > 60  -- 查询最高成绩和最低成绩：  select max(score) '最高分',min(score) '最低分' from stu;  -- 统计stu表中成绩大于60的最高成绩和最低成绩：  select max(score) '最高分',min(score) '最低分' from stu where score > 60;  -- 查询最高成绩,以及学生姓名  /\*  与聚合函数一同出现的列名,必须出现在group by后  反之,如果出现了一个没有在groupby后出现的字段,那么查询结果不正常!  \*/  select max(score) '最高分',sname from stu;  select sum(age) from stu; -- 490  select sum(score) from stu; -- 742  -- 计算总年龄与总成绩的和  select sum(age)+sum(score) from stu;  select sum(age + score) from stu; |

## 去重

|  |
| --- |
| /\*  去重函数 distinct(列)  将指定列的数据去重  \*/  -- 不重复的年龄  select distinct(age) from stu;  -- 查询年龄不重复的共有多少人  select count(distinct age) from stu; |

## 分组查询



|  |
| --- |
| -- ================ 分组查询 ==================  /\*  当需要分组查询时,需要使用group by语句.  语法:  select \* from stu [where ...] group by 字段 [order by..]  字段值相同的分到一组  注意:  1 一般分组之后的操作都是聚合操作  2 分成几组,虚拟表结果就是几行数据  3 聚合函数,是对每组进行单独聚合操作  --------------------------------  分组后再过滤,使用having子句  与where的区别:  where是分组前过滤,having是分组后过滤  where后不能使用聚合函数,having可以使用聚合函数    \*/  -- 查询男生多少人,女生多少人  select sex,count(\*) '人数' from stu group by sex;  -- 查询每个班级的班级编号和每个班级的成绩和：  select cid,sum(score) from stu group by cid;  -- 查询每个班级的班级编号以及每个班级的人数：  select cid '班级',count(sid) '人数' from stu group by cid;  -- 查询成绩总和大于200的班级编号以及成绩和  select cid,sum(score) from stu group by cid having sum(score) > 200  -- having可以使用前面定义的别名  select cid,sum(score) 'sum' from stu group by cid having sum > 200  -- 查询成绩总和大于200的班级编号以及成绩和并根据成绩总和降序  select cid,sum(score) 'sum' from stu group by cid having sum > 200 order by sum asc; |

## 限制输出

|  |
| --- |
| -- =============== 限制输出 ================  /\*  可以将查询好的结果,限制输出.  可以限制输出几条,从哪里输出.  语法:  在sql语句最后写 limit offset,row\_count  limit 开始下标,行数;  例如: limit 0,2; ==> 从第一条输出,输出2条  \*/  -- 输出前两条  select \* from stu limit 0,2;  -- 从第四条开始,输出三条  select \* from stu limit 3,3;  /\*  分页: 每页3条数据  \*/  -- 第一页  select \* from stu limit 0,3;  -- 第二页  select \* from stu limit 3,3; |

# 函数

## 流程函数

|  |
| --- |
| /\*  IF(expr1,expr2,expr3)  如果expr1为真，则返回expr2，否则返回expr3  -- isnull()函数,判断是否为空  -- isnull(字段) 如果是null,返回1 不是null返回0  \*/  -- 查询学生id,姓名,成绩,如果成绩为null,显示缺考  select sid,sname,if(isnull(score)=1,'缺考',score) from stu;  -- is null 是判断条件  -- isnull() 是函数  update stu set score = 72 where score is null;  -- IFNULL(expr1,expr2)如果 expr1不是NULL,则返回expr1，否则返回expr2；  -- 查询学生id,姓名,成绩,如果成绩为null,显示缺考  select sid,sname,ifnull(score,'缺考') '成绩' from stu  /\*  CASE WHEN [expr1] THEN [result1]… ELSE [default] END 如果expr是真, 返回result1,否则返回default  case  when 条件 then 执行语句  when 条件 then 执行语句  ...  else 执行语句  end  执行第一个when后的条件,如果为true,执行then后的语句,  如果when后的条件为false,执行第二when后的条件  如果都是false,执行else后语句  \*/  -- 查询学生id,姓名,成绩,以及等级  -- (0-59 不及格,60-69中,70-89良,90-100优 )  select sid,sname,score,  case  when score < 60 then '不及格'  when score < 70 then '中'  when score < 90 then '良'  when isnull(score)=1 then '缺考'  else '优'  end as '等级'  from stu |
| -- 类似于switch case结构  CASE  [VALUE]  WHEN [ value1 ] THEN [ result1 ]  …  ELSE  [ DEFAULT ]  END  如果value等于value1,返回result1,  否则返回default  select case score  when 72 then '七十二'  when 78 then '七十八'  else '其他'  end  from stu |

# 总结

书写语法:

select

selection\_list --要查询的列

from

table\_name -- 要查询的表名

where condition -- 过滤行条件

group by grouping\_clumns -- 对结果按照列进行分组

having condition -- 分组后再过滤

order by sort\_column -- 排序

limit offset,row\_count -- 对结果限制

# 多表查询

## 1 联合查询-合并结果集

将两表的查询结果纵向拼接在一起

union / union all

拼接的两个表的字段个数,类型,顺序保证一致.

union 在拼接时会去重

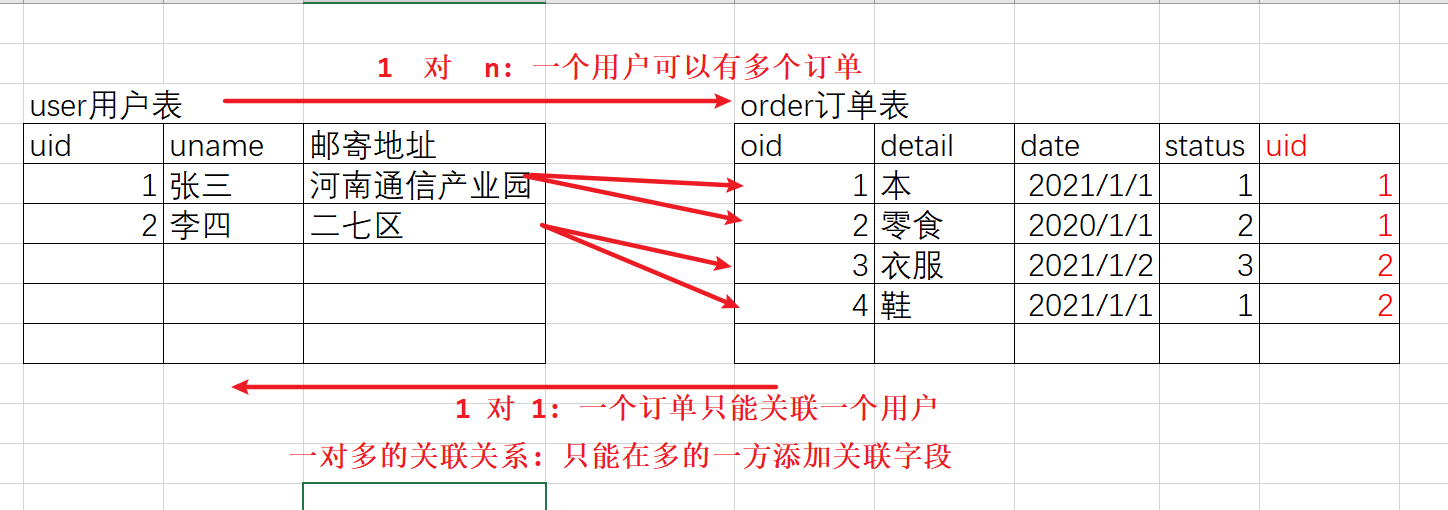
union all将两表的全部数据都拼接

|  |
| --- |
| select sid,sname from stu  union all  select cid,cname from class; |

## 2 连接查询

### 表和表之间的关系







连接查询就是将多个表多行数据相乘. 🡪 笛卡尔积

### 内连接

|  |
| --- |
| -- =========== 内连接 ==========  /\*  内连接语法:  select \* from 表1 inner join 表2 on 关联条件  内连接只会保留完全符合关联条件的数据  \*/  -- 没有关联条件,会产生笛卡尔积  select \* from stu inner join class  select \* from stu inner join class on stu.cid = class.cid  -- 查询每个学生的信息,包括班级的全部信息  -- 查询学生名称、学生成绩、班级名称、班级地址  select sname,score,cname,caddress from stu inner join class on stu.cid = class.cid  -- 查询学号为1007的学生名称、学生成绩、班级名称、班级地址  -- 给表取别名  SELECT  sname,  score,  cname,  caddress  FROM  stu s  INNER JOIN class c ON s.cid = c.cid  WHERE  s.sid = 1007  -- 内连接可以简写  select \* from stu inner join class on stu.cid = class.cid  -- 简写成  select \* from stu,class where stu.cid = class.cid |

### 外连接

|  |
| --- |
| /\*  外连接: 会保留不满足条件的数据  左外连接-left outer join on  会保留左表中不符合的条件的数据  右外连接 - right outer join on  会保留右表中不符合的条件的数据  一般outer关键字可以省略  \*/  -- 查询全部学生信息,如果有班级信息,一并查出.  select \* from stu s left outer join class c on s.cid = c.cid  -- 查询全部班级信息,如果有学生,将学生信息查出  select \* from stu s right outer join class c on s.cid = c.cid |

### 子查询

子查询就是嵌套查询.

一般子查询出现在:

from后, 当做一张表使用

where后, 当做条件使用

select后

|  |
| --- |
| -- ============= 子查询 =============  /\*  子查询就是嵌套查询.  一般子查询出现在:  from后, 当做一张表使用  where后, 当做条件使用  select后  \*/  -- 查询与张三同一个班级的学生。  -- 1查出张三的班级  select cid from stu where sname = '张三'  -- 2查出1班的学生  select \* from stu where cid = 1  -- 合并成子查询  select \* from stu where cid = (select cid from stu where sname = '张三')  -- 成绩高于3号班级所有人的学生信息  -- 1 找到3班最高分  select max(score) from stu where cid = 3  -- 2 找成绩比96高的学生  select \* from stu where score > 96  -- 合并  select \* from stu where score > (select max(score) from stu where cid = 3)  -- 有2个以上直接组员的学生信息  -- 1 根据组长编号分组,计算分组人数,保留>2的组  select groupLeaderId,count(sid) from stu group by groupLeaderId having count(sid) > 2  -- 2 查询学号1007,1010学生的信息  select \* from stu where sid in(1007,1010)  -- 合并  SELECT  \*  FROM  stu  WHERE  sid IN (  SELECT  groupLeaderId  FROM  stu  GROUP BY  groupLeaderId  HAVING  count(sid) > 2  )  -- 自连接,自己连接自己,s1当做组长,s2当做组员  select s1.sid,s1.sname from stu s1,stu s2 where s1.sid = s2.groupLeaderId group by s1.sid having count(s1.sid) > 2  -- stu s1当做组员,stu s2 组长  select s2.sid,s2.sname from stu s1,stu s2 where s1.groupLeaderId = s2.sid group by s2.sid having count(s2.sid) > 2  -- 求1008学生编号、姓名、组长编号和组长姓名  -- 1 找到1008学生的组长的编号  select groupLeaderId from stu where sid = 1008  -- 2 找到1010的编号和姓名  select sid,sname from stu where sid = 1010  -- 3 找1008编号和姓名  select sid,sname from stu where sid = 1008  -- 合并  SELECT  s.sid,  s.sname,  b.sid,  b.sname  FROM  stu s,  (  SELECT  sid,  sname  FROM  stu  WHERE  sid = (  SELECT  groupLeaderId  FROM  stu  WHERE  sid = 1008  )  ) b  WHERE  s.sid = 1008  -- 自连接,s1做学生表 s2做组长表  -- s1的组长是s2中的学生  select s1.sid,s1.sname,s2.sid,s2.sname from stu s1,stu s2 where s1.groupLeaderId = s2.sid and s1.sid = 1008  -- 自连接,s1做组长 s2做学生  select s2.sid,s2.sname,s1.sid,s1.sname from stu s1,stu s2 where s2.groupLeaderId = s1.sid and s2.sid = 1008 |

### 自连接

自己连接自己

# 作业

1查询每个学生成绩大于60且成绩总和小于200的班级编号以及成绩和并根据成绩和降序

2创建一个表：student sid sname sage sex score sclass sbirthday

》 插入20条数据: 其中sgae随机 sex随机 score随机

》 获取所有年龄小于18 并且 名字中有妹的 所有女生的信息

》 获取所有学生的总人数 平均分 最低分 最高分

》 获取所有参加考试的学生的总人数和参加考试学生的平均分 和 所有学生的平均分

》 获取所有学生的自我介绍信息：我叫xxx,今年xxx岁，明年xx岁，考了xx分,我是成年人/未成年人

》 获取5月份生日的学生的个数

》 获取本月生日的学生的个数

》 获取年龄在18到22岁之间的学生的信息

》 获取第6-10条的所有学生的信息

》 把所有女生 并且年龄<20的学生分数+1

》 把所有成绩为null的学生 分数设置为100

》 删除sclass列

》 添加sgrade列 列类型固定长度4的字符串

》 修改列名sgrade为grade

》 修改grade列类型为char(4)

》 修改表名为stu

# 预习

mysql 函数,字符串函数,日期函数,数字函数

mysql 事务

JDBC