

1, 数据库的查询

①单表查询

a, select **【all | distinct】** <目标列名列表> from <数据源>

【where <检索条件表达式>】 【group by<分组依据列>having <分组提取条件表达式>】

【order by<排序依据列> 【asc | desc】】

select 姓名, 年龄+10 as age from S 使得每个信息的年龄增加10岁, 并且给年龄取一个新的属性名age

select 姓名, '籍贯' as jg, 学号 from S 在表中添加一个新的列, 并且是无列明, 列值都是籍贯, 其中as关键字可以省略

sql提供了为关系表和属性重新命名的机制, 列更名: 旧列名 | 表达式 as 新列名 其中as关键字可以省略 或者: 新列名 = 旧列名 | 表达式

如果在结果中重复的数据不想显示出来, 可以使用distinct关键字

b, select *from表示查询表中的全部信息

②嵌套字查询

③多表连接查询

④复杂查询

2, 条件查询

select 列名列表 from 表名 where 条件表达式

select *from S where 性别 = '女', 在查询过程中, 如果查询列的类型是字符型, 该值需要使用单引号引起来

a, 比较运算符: =, >=, <, <=, !=, not

b, 确定范围: between and (包含上下限), not between and

c, 确定集合: in, not in (如果选择的東西有多个, 则需要将这多个选择的東西用单引号包含并用小括号括起来)

d, 模糊字符匹配: like, not like

列名 like <字符串>, 在使用字符串中可以使用通配符: _代表任意一个字符, %表示任意多个字符

若字符串为“王%”表示查询数据库中姓王的所有人的基本信息

若字符串为“王_”表示查询数据库中姓王且名字只有俩个字的所有人的信息

e, 空值: is null, is not null

select *from S WHERE 学号 is null, 查询学号是空的所有学生信息

f, 多重条件: and , or

嵌套在比较运算符符合其他查询条件之间

3, 排序

order by 列名 asc / desc 表示对某一列进行排序, asc表示从小到大进行排序, desc表示从大到小进行排序, 默认是升序排列

注意: 对于空值, 若按照升序排列, 含有空值的元组将最后显示, 若按照降序排列, 空值将最先显示