指定查询列,例如id,姓名,语文成绩 格式: SELECT (column\_name, column\_name,...) FROM table\_name [WHERE Clause] [LIMIT N][ OFFSET M] 虽然mysql提供有distinct这个关键字来过滤掉多余的重复记录只保留一条,但往往只用它来返回不重复记录的条数,而不是用它来返回不重复记录的所有值。其原因是distinct只能返回它的目标字段,而无法返回其它字段。 当我们需要在多个字段中消重复的时候,可以配合group by 和count (\*)使用 例如: 查询得到name不重复的所有数据包括id值 此时使用select distinct name from table得到的仅仅是name a b c而不包括id值,因为distinct只作用于name字段 用select \*, count(distinct name) from table group by name, 利用count来绕过distinct的局限性 可得到 id name count(distinct name) 1 a 1 2 b 1 3 c 1 distinct可去除结果中完全相同的行 查询: as起别名: 在select语句中可以使用as给查看出的信息赋予别名 🗨 格式 select 字段名 as 别名 from 表名 where条件过滤 在select语句查询过程中,可以使用where进行条件的限定 大于、小于、大于(小于)等于、不等于 显示在某一区间的值 显示在1种列表中的值,例: in(100,200) 連報查询 select 字段名 from 表名 where 字段名 between xx and xxx between and  $\Theta$ xx和xxx是一个范围 order by 排序子句: 在进行模糊查询时要分清'\_'和'%'的使用场景(下划线适用于某一位随意匹配) 🖨 示例: select 字段名 from 表名 where 字段名 like '社%' 在使用中可以用于统计满足该条件的有多少行;例如统计有多少个学生数学成绩超过了50: count (列名):返回某一列行的总数 select count(\*) from student where math > 50 总结: where在group by之前执行,having 在group by 之后执行。where和having 可以同时存在一个SQL语句中。 sum函数返回满足where条件限定的行的和; 在查询过程中聚合语句(sum, min, max, avg, count)要比having子句优先执行.而where子句在查询过程中执行优先级别优先于聚合语句 avg函数返回满足where条件的一列的平均值; 聚合函数 max/min返回满足where条件的一列的最大/最小值; 例如找出一个班中的数学的最高和最低分的学生: select max(math), min(math) from student; where在sq1中是一个约束声明,使用where来约束数据库的数据。 select 查询总分大于100, 且名字第二个字为神的学生: select \*from student where chinese+math+english > 100 and name like '\_神'; Having和where的区别 having是一个过滤声明,是在查询返回结果集以后对查询结果进行的过滤操作。 by asc表示升序 (默认) having是要配合group by 使用,在分组之后过滤数据。 select的结尾处添加order by指定排序的顺序; group by子句 by desc表示降序 对于group by往往和having配合使用,对group by结果进行过滤。 例如找出数学平均分大于40分的班级: select class, avg(math) as avgmath from student group by class having avgmath > 40; class | avgmath 1 | 63.000000 | 2 | 57.000000 | +----+ 显示每个班数学的平均分和最高分: select class, max(math), avg(math) from student group by class; class | max(math) | avg(math) | 1 | 73.00 | 63.000000 |

2 | 66.00 | 57.000000 |