### 一、一些基本语法

1. 一些常用的查询条件：

|  |  |
| --- | --- |
| 查询条件 | 谓词 |
| 比较 | =、>、<、>=、<=、!=（<>）、!>、!< |
| 确定范围 | BETWEEN AND、NOT BETWEEN AND |
| 确定集合 | IN、NOT IN |
| 字符匹配 | LIKE、NOT LIKE |
| 空值 | IS NULL、IS NOT NULL |
| 逻辑运算 | AND、OR、NOT |

注意： IN表示的是 或关系 ；NOT IN表示的是 且关系；

使用 LIKE 时可能会使用通配符 %和\_ ，%代表任意长度的字符串；\_代表任意单字符；

1. ORDER BY子句（默认升序排列）

可以使用 ORDER BY 子句对查询结果按照一个或多个属性列的升序或降序排序

eg：查询学生情况，查询结果按所在系的系号升序排列，同一系中的学生按年龄降序排列。

select \* from student order by sdept,sage desc

注意：ORDER BY 3 DESC 表示按表的第三列降序排列

1. GROUP BY子句

可以使用 GROUP BY 子句对查询结果按某一列或多列的值分组，值相等的为一组

eg：查询平均成绩大于等于90分的学生学号和平均成绩

select Sno, avg(grade) from sv group by sno having avg(grade) >= 90

1. LIMIT [offest] rows

offest表示从offest+1行开始，查询rows行

eg：select \* from user limit 0,10 表示查询1-10行的记录；

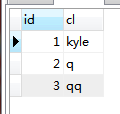
limit rows表示查询数据库表信息的前rows行

eg：select \* from user limit 5 表示查询数据库表的前5行记录

### 流程控制类函数

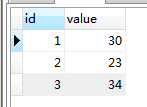
1. IF(v,t,f)：如果v为true，则返回t，否则返回f

eg：select id, IF(2>1,name,money) as cl from account



1. IFNULL(expression,value) 判断表达式是否为null，若为null则返回value，否则返回表达式的值

eg：select id, IFNULL(null,money) as value from account



1. case when 函数

eg：select id,

case

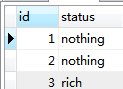
when money > 30 then "rich"

when money < 20 then "poor"

ELSE "nothing"

end "status"

from account



### 字符串类函数

1. CONCAT(str1,str2...)拼接字符串（常用语LIKE后面）

eg：select concat(‘a’,’b’,’c’) str;

select \* from account where name like concat(‘%’,’k’,’%’);

1. LOWER(s) 将字符串全部变成小写

eg：select lower(‘ABC’) lower;



1. UPPER(s) 将字符串全部变成大写

eg：select upper(‘abc’) upper;



### 聚集函数（只能用于select子句和having子句中）

1. count(\*)，count(1)，count(column\_name)

count(\*)与count(1)都是统计所有行，并且包括列为NULL的值；注：count(1)并不是表示统计第一列

count(column\_name)统计列不为NULL的行数

1. SUM(column\_name)

表示计算一列值的的综合，不包含列为NULL的，并且此列必须为数值型，否则结果会返回为0，并不是NULL

1. AVG(column\_name)

表示计算一列值的平均值，不包含列为NULL的，并且此列必须为数值型，否则结果会返回为0，并不是NULL

1. MAX(column\_name)

表示求一列中的最大值

1. MIN(column\_name)

表示求一列中的最小值