**MySQL查询语句大全（列举所有常用查询方式）**

**前提条件**

首先创建SQL测试文件，其中创建了一个库两个表，用于测试。

文件名称：test.sql

/\*------------员工信息库-------------\*/

create database staff;

use staff;

create table yunwei(id int not null primary key,name char(4) not null,age tinyint(3) unsigned not null,sex enum('man','woman') not null);

create table caiwu(id int not null primary key,name char(4) not null,age tinyint(3) unsigned not null,sex enum('man','woman') not null);

insert into yunwei values(1,'张三',20,'man'),(2,'张四',21,'man'),(3,'张莉',22,'woman'),(4,'张五',23,'man'),(5,'张六',24,'man'),(6,'张丽',25,'woman');

insert into caiwu values(1,'李一',24,'man'),(2,'李莉',23,'woman'),(3,'李二',22,'man'),(4,'李三',21,'man'),(5,'李四',20,'man'),(6,'李丽',19,'woman');

将SQL文件导入到数据库中

mysql < test.sql

导入完成后，在staff库下有两张表，分别是yunwei和caiwu

mysql> select \* from staff.yunwei;

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 1 | 张三 | 20 | man |

| 2 | 张四 | 21 | man |

| 3 | 张莉 | 22 | woman |

| 4 | 张五 | 23 | man |

| 5 | 张六 | 24 | man |

| 6 | 张丽 | 25 | woman |

+----+--------+-----+-------+

mysql> select \* from staff.caiwu;

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 1 | 李一 | 24 | man |

| 2 | 李莉 | 23 | woman |

| 3 | 李二 | 22 | man |

| 4 | 李三 | 21 | man |

| 5 | 李四 | 20 | man |

| 6 | 李丽 | 19 | woman |

+----+--------+-----+-------+

**简单查询**

**直接查询**

**语法：select 字段 from 表名；**

最简单的一种查询方式，可查看多个字段或整张表。

例如：查询运维表姓名及年龄

mysql> select name,age from yunwei;

+--------+-----+

| name | age |

+--------+-----+

| 张三 | 20 |

| 张四 | 21 |

| 张莉 | 22 |

| 张五 | 23 |

| 张六 | 24 |

| 张丽 | 25 |

+--------+-----+

**条件查询**

关键字为**where**，通常位于表名后面

**语法：select 字段 from 表名 where 条件；**

根据条件，查询指定条件的字段。

例如：查询财务表中年龄为20的员工

mysql> select \* from caiwu where age=20;

+----+--------+-----+-----+

| id | name | age | sex |

+----+--------+-----+-----+

| 5 | 李四 | 20 | man |

+----+--------+-----+-----+

**模糊查询**

关键字是**like**，通常位于条件字段后面

**语法：select 字段 from 表名 where 字段 like ‘%数据%’；**

通过输入具体的数据，来对记录进行查询

例如：模糊查询caiwu表中姓名字段还有三的记录

mysql> select \* from caiwu where name like '%三%';

+----+--------+-----+-----+

| id | name | age | sex |

+----+--------+-----+-----+

| 4 | 李三 | 21 | man |

+----+--------+-----+-----+

**算数运算符**

运算符一般配合逻辑运算符一起使用，可以使条件限制更加具体。

| **符号** | **作用** |
| --- | --- |
| > | 大于 |
| < | 小于 |
| = | 等于 |
| != | 不等于 |
| <> | 与!=同义，不等于 |
| >= | 大于等于 |
| <= | 小于等于 |

通过逻辑运算符可以将条件限制在一个范围内。

例如：查看caiwu表中id不等于1的记录

mysql> select \* from caiwu where id!=1;

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 2 | 李莉 | 23 | woman |

| 3 | 李二 | 22 | man |

| 4 | 李三 | 21 | man |

| 5 | 李四 | 20 | man |

| 6 | 李丽 | 19 | woman |

+----+--------+-----+-------+

**逻辑运算符**

可以将查询的单个条件改为多个条件或满足多个条件中的一个。

| **符号** | **作用** |
| --- | --- |
| and | 与，同时满足多个条件 |
| or | 或，满足多个条件中的一个即可 |
| not | 否，不满足条件 |

和算数运算符一起使用，可以将条件更加具体。

例如：查询yunwei表中性别为女，或年龄为23的记录

mysql> select \* from yunwei where sex='woman' or age=23;

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 3 | 张莉 | 22 | woman |

| 4 | 张五 | 23 | man |

| 6 | 张丽 | 25 | woman |

+----+--------+-----+-------+

**in与not in运算符**

关键字为**in**，通常位于条件字段后面

**语法：select 字段 from 表名 where 字段 in （列表）；**

| **符号** | **作用** |
| --- | --- |
| in | 在一个条件列表中 |
| not in | 不在一个条件列表中 |

例如：查询yunwei表中，年龄为22-25之间的记录

mysql> select \* from yunwei where age in(22,23,24,25);

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 3 | 张莉 | 22 | woman |

| 4 | 张五 | 23 | man |

| 5 | 张六 | 24 | man |

| 6 | 张丽 | 25 | woman |

+----+--------+-----+-------+

**排序查询**

关键字为**order by**与**asc，desc**，通常位于表名之后

排序分为两种，升序（asc）和降序（desc）

**语法：select 字段 from 表名 order by 字段 排序方式；**

例如：将caiwu表中记录按年龄从大到小查询

mysql> select \* from caiwu order by age desc;

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 1 | 李一 | 24 | man |

| 2 | 李莉 | 23 | woman |

| 3 | 李二 | 22 | man |

| 4 | 李三 | 21 | man |

| 5 | 李四 | 20 | man |

| 6 | 李丽 | 19 | woman |

+----+--------+-----+-------+

**高级查询**

**范围运算**

关键字为**between…and…**，通常位于条件字段后面。

**语法：select 字段 from 表名 where 字段 between 范围1 and 范围2；**

也是用来限制查询范围，作为算数运算符的一种替换。

例如：查询caiwu表中，年龄为21-23的记录，使用算数运算符表示为age>=21 and age<=23，使用范围运算啧表示为age between 21 and 23。

mysql> select \*from caiwu where age >=21 and age<=23;

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 2 | 李莉 | 23 | woman |

| 3 | 李二 | 22 | man |

| 4 | 李三 | 21 | man |

+----+--------+-----+-------+

mysql> select \*from caiwu where age between 21 and 23;

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 2 | 李莉 | 23 | woman |

| 3 | 李二 | 22 | man |

| 4 | 李三 | 21 | man |

+----+--------+-----+-------+

**限制查询**

关键字为**limit**，通常位于表名后面。

**语法：select 字段 from 表名 limit n，m；**

limit可以强制指定查询结果的记录条数。

n是开始记录行，0表示第一条记录，m表示显示行，从n开始，共显示几行记录。

时刻注意开始范围时从0开始的，1表示的是第二行，而非第一行。  
显示范围就是共显示几条记录，并不是结束范围。

例如：查询yunwei表中第2-4行记录

mysql> select \* from yunwei limit 1,3;

+----+--------+-----+-------+

| id | name | age | sex |

+----+--------+-----+-------+

| 2 | 张四 | 21 | man |

| 3 | 张莉 | 22 | woman |

| 4 | 张五 | 23 | man |

+----+--------+-----+-------+

命令解读：查询yunwei表中记录，从第二行开始，共显示三行。

**嵌套查询**

没有关键字，嵌套查询分为查询语句和子查询语句，在查询语句中含有子查询语句，所以叫做嵌套查询。

嵌套子查询通常位于查询语句的条件之后。

例如：在caiwu表中查询名称为张三的字段。

先在caiwu表中添加一个张三字段，并且年龄不同

mysql> insert into caiwu values(7,'张三',25,'man');

通过子查询的方式，查询caiwu和yunwei表中名称字段相同的字段

mysql> select name,age from caiwu where name=(select name from yunwei where age=20);

+--------+-----+

| name | age |

+--------+-----+

| 张三 | 25 |

+--------+-----+

前半段是正常的查询语句，name=（）中为子查询语句，查询的字段为name，查找yunwei表中age为20的name，结果为张三，将结果执行外部的查询语句，就完成了。

使用嵌套查询，两个表必须要有一个关联字段，否则无法实现。

测试完成后，将添加的字段删除即可。

**多表连查**

多表连查全称多表连接查询，和嵌套子查询一样，都需要有一个共同的字段，然后将多个表连接在一起查询，将符合条件的组成一个合集。

常见连接：内连接，外连接

**内连接**

根据两个表中共有的字段进行匹配，然后将符合条件的合集进行拼接。

关键字为**inner join…on…**，通常位于表名后面。

**语法：select 字段 from 表1 inner join 表2 on 表1.字段=表2.字段；**

on后面的是连接条件，也就是表1和表2共有的字段

例如：将yunwei和caiwu表连接在一起

mysql> select \* from yunwei inner join caiwu on caiwu.id=yunwei.id ;

+----+--------+-----+-------+----+--------+-----+-------+

| id | name | age | sex | id | name | age | sex |

+----+--------+-----+-------+----+--------+-----+-------+

| 1 | 张三 | 20 | man | 1 | 李一 | 24 | man |

| 2 | 张四 | 21 | man | 2 | 李莉 | 23 | woman |

| 3 | 张莉 | 22 | woman | 3 | 李二 | 22 | man |

| 4 | 张五 | 23 | man | 4 | 李三 | 21 | man |

| 5 | 张六 | 24 | man | 5 | 李四 | 20 | man |

| 6 | 张丽 | 25 | woman | 6 | 李丽 | 19 | woman |

+----+--------+-----+-------+----+--------+-----+-------+

**左连接**

关键字为**left join…on**，通常位于表名后面

**语法：select 字段 from 表1 left join 表2 on 连接条件；**

左连接是左表为主表，指定字段都显示，右表为从表，没内容显示null。

使用左连接将两个表连接到一起

mysql> select \* from yunwei left join caiwu on yunwei.id=caiwu.id;

+----+--------+-----+-------+------+--------+------+-------+

| id | name | age | sex | id | name | age | sex |

+----+--------+-----+-------+------+--------+------+-------+

| 1 | 张三 | 20 | man | 1 | 李一 | 24 | man |

| 2 | 张四 | 21 | man | 2 | 李莉 | 23 | woman |

| 3 | 张莉 | 22 | woman | 3 | 李二 | 22 | man |

| 4 | 张五 | 23 | man | 4 | 李三 | 21 | man |

| 5 | 张六 | 24 | man | 5 | 李四 | 20 | man |

| 6 | 张丽 | 25 | woman | 6 | 李丽 | 19 | woman |

+----+--------+-----+-------+------+--------+------+-------+

**右连接**

关键字为**right join**，通常位于表名后面

**语法：select 字段 from 表1 right join 表2 on 连接条件**

右连接是以右表为主表，指定字段都显示，左表为从表，没内容显示null。

使用右连接将两个表连接在一起

mysql> select \* from yunwei right join caiwu on yunwei.id=caiwu.id;

+------+--------+------+-------+----+--------+-----+-------+

| id | name | age | sex | id | name | age | sex |

+------+--------+------+-------+----+--------+-----+-------+

| 1 | 张三 | 20 | man | 1 | 李一 | 24 | man |

| 2 | 张四 | 21 | man | 2 | 李莉 | 23 | woman |

| 3 | 张莉 | 22 | woman | 3 | 李二 | 22 | man |

| 4 | 张五 | 23 | man | 4 | 李三 | 21 | man |

| 5 | 张六 | 24 | man | 5 | 李四 | 20 | man |

| 6 | 张丽 | 25 | woman | 6 | 李丽 | 19 | woman |

+------+--------+------+-------+----+--------+-----+-------+

**聚合函数**

**最小值min()**

关键字为**min()**，通常位于select之后

**格式：select min(字段) from 表名；**

查询字段的最小数值

例如：查询yunwei表中年龄最小的值

mysql> select min(age) from yunwei;

+----------+

| min(age) |

+----------+

| 20 |

+----------+

**最大值max()**

关键字为**max()**，通常位于select之后

**格式：select max(字段) from 表名；**

查询字段的最大数值

例如：查询yunwei表中年龄最大的值

mysql> select max(age) from yunwei;

+----------+

| max(age) |

+----------+

| 25 |

+----------+

**求和sum()**

关键字为**sum()**，通常位于select之后

**格式：select sum(字段) from 表名；**

用于统计数值和数

例如：将yunwei表中id字段求和

mysql> select sum(id) from yunwei;

+---------+

| sum(id) |

+---------+

| 21 |

+---------+

**平均值avg()**

关键字为**avg()**，通常位于select之后

**格式：select avg(字段) from 表名；**

例如：查询caiwu表中年龄字段的平均值

用于统计数值平均值

mysql> select avg(age) from caiwu;

+----------+

| avg(age) |

+----------+

| 21.5000 |

+----------+

**统计记录count()**

关键字为**count()**，通常位于select之后

**语法：select count(字段) from 表名**

用于统计记录的个数

例如：查询caiwu记录个数

mysql> select count(id) from caiwu;

+-----------+

| count(id) |

+-----------+

| 6 |

+-----------+

**as 聚合别名**

关键字为**as**，常用于聚合查询之后

**语法：select 运算函数(字段) as 别名 from 表名；**

可以给字段设置别名

例如：统计caiwu表中的记录个数，并设置别名为记录总个数

mysql> select count(id) as 记录总个数 from caiwu;

+-----------------+

| 记录总个数 |

+-----------------+

| 6 |

+-----------------+

**大小写转换**

关键字为**upper()**，常位于select之后

**格式：select upper(字段) from 表名；**

只能用于英文字母的数据

例如：将caiwu表中id为1的性别字段改为大写

mysql> select upper(sex) from caiwu where id=1;

+------------+

| upper(sex) |

+------------+

| MAN |

+------------+