# 查询语句

## 基本语法

MySQL从数据表中查询数据的语句是SELECT语句。在SELECT语句中，可以根据自 己 对数据的需求，使用不同的查询条件。SELECT语句的基本语法格式为：

**SELECT [DISTINCT] \*|字段名1,字段名2,… FROM 表名**

**[WHERE 条件表达式]**

**[GROUP BY 字段名 [HAVING 条件表达式] ]**

**[ORDER BY 字段名 [ASC|DESC] ]**

**[LIMIT [OFFSET] n]**

1. DEITINCT：是可选参数，功能是剔除查询结果中重复的数据。
2. \*|字段名1,字段名2,…：表示从表中查询指定字段，\*表示表中所有字段，二者为 互斥关系，任选其一。
3. FROM 表名：表示从指定的表中查询数据。
4. WHERE 条件表达式：可选参数，用于指定查询条件。
5. GROUP BY 字段名 [HAVING 条件表达式]：可选参数，用于将查询结果按照指定 字段进行分组，“HAVING”也是可选参数，用于对分组后的结果进行过滤。
6. ORDER BY 字段名 [ASC|DESC]：可选参数，用于将查询结果按照指定字段进行排序 。排序方式有参数ASC或DESC控制；ASC表示按升序进行排序，DESC表示按降序进 行排序。如果没指定，默认为按升序进行排序。
7. LIMIT [OFFSET] , n 记录数：可选参数，用于限制查询结果的数量。OFFSET是可选参数，表示偏移量，如果偏移量为0则从查询结果的第一条记录开始，偏移量为1，则从 查询结果的第二条记录开始，以此类推，如果没有指定其默认值为0。n表示返回查询记录的条数。

## 简单查询

简单查询指的是在没有使用任何条件语句或可选参数的情况下对数据表进行查询。它包括两种，即查询所有字段的数据和查询指定字段的数据。

1. 查询所有字段

查询所有字段是指查询表中所有字段的数据。其语法格式为：

**SELECT \* FROM 表名;**

查询结果会按照字段在表中定义的顺序显示。

1. 查询指定字段

查询指定字段是指在SELECT语句的字段列表中指定要查询的字段。其语法格式为：

**SELECT 字段名1,字段名2,… FROM 表名;**

查询结果中字段的显示顺序会和在SELECT语句中指定的字段顺序一致。

# 使用where子句

## 基本语法

在SELECT语句中，最常见的是使用WHERE子句指定查询条件对查询结果进行过滤，其语法格式为：

**SELECT \*|字段名1,字段名2,… FROM 表名**

**WHERE 条件表达式 ;**

其中，根据条件表达式的不同可以将WHERE子句分为五种：关系表达式查询、IN关键 字查询、BETWEEN AND关键字查询、空值查询、LIKE关键字查询。

## 关系表达式查询

关系表达式查询用于查询满足关系表达式的记录。其语法格式为：

**SELECT \*|字段1,字段2,… FROM 表名**

**WHERE 字段名 关系运算符 字段值 ;**

关系运算符有>、>=、<、<=、==、！=（在MySQL中建议使用<>代替！=，表示不等于）六种。

## IN关键字查询

IN关键字查询用于查询指定字段的值等于指定值集合中的记录。其语法格式为：

**SELECT \*|字段1,字段2… FROM 表名**

**WHERE 字段名 [NOT] IN (元素1,元素2,…) ;**

其中，NOT为可选参数。使用NOT表示查询指定字段值不等于指定值集合中的记录。

例如WHERE id IN ( 1 , 2, 3 )表示查询id字段值等于1,2,3的记录。如果使用了NOT，则 表示查询id字段值不等于1,2,3的记录。

## BETWWEEN AND关键字查询

BETWEEN AND关键字用于查询指定字段的某值到某值之间的记录。其语法格式为：

**SELECT \*|字段1,字段2,… FROM 表名**

**WHERE 字段名 [NOT] BETWWEN 值1 AND 值2 ;**

其中，值1表示范围条件的起始值，值2表示范围条件的结束值，且值1必须小于值2；

NOT为可选参数，使用了NOT则表示查询指定字段某值到某值之间以外的记录。

## 空值查询

在数据表中，某些列的值可能为空值（NULL）。在MySQL中，可通过IS NULL关键字 查询指定字段值为空的记录。其语法格式为：

**SELECT \*|字段1,字段2,… FROM 表名**

**WHERE 字段名 IS [NOT] NULL ;**

NOT为可选参数，使用了NOT关键字则表示查询指定字段不为空的记录。

## LIKE关键字查询

LIKE关键字查询表示字符模糊查询，用于查询匹配指定字符串的记录。其语法格式 为：

**SELECT \*|字段名1,字段名2,… FROM 表名**

**WHERE 字段名 [NOT] LIKE‘匹配字符串’;**

其中，‘匹配字符串’指定用来匹配规则，可以是包含百分号（%）和下划线（\_） 的 字符串。百分号和下划线统称为通配符。

1. 使用百分号可以代表一个字符串或者空字符。例如：

·‘%c’表示匹配以字符c结束的字符串

·‘a%’表示匹配以字符串a开始的字符串

·‘a%c’表示匹配以字符a开始、c结束的字符串

1. 使用下划线可以代表一个字符，但一个下划线只能匹配一个字符，如果要匹配多个 字符，则需要使用多个下划线。例如：

·‘c\_ \_1’表示匹配长度为4，以c开头1结束，中间包含两个字符的字符串

·‘cu\_’表示匹配长度为3，以cu开头，任意单个字符结尾的字符串

·‘\_a’表示匹配长度为2，以任意单个字符开头，a结尾的字符串

当匹配字符串与指定字段的值匹配成功，得到其记录。

NOT关键字为可选参数，使用了NOT关键字则表示查询匹配成功的记录以外的记录。

如果需要匹配包含“%”或“\_”的字符串，则需要使用“\”进行转义。

# 多条件查询

多条件查询指的是在WHERE子句后跟上多个条件表达式，各个条件表达式之间使用AND关键字或OR关键字连接。

## AND关键字

使用AND关键字连接多个条件表达式，只有满足所有条件表达式的记录才会被返回。 其语法格式为：

**SELECT \*|字段1,字段2,... FROM 表名**

**WHERE 条件表达式1 AND 条件表达式2 ...... ;**

AND代表与。

## OR关键字

使用OR关键字连接多个条件表达式时，只要满足其中任意一个条件表达式的记录就可 以被返回。其语法格式为：

**SELECT \*|字段1,字段2,... FROM 表名**

**WHERE 条件表达式1 OR 条件表达式2...... ;**

OR代表或者。

# 聚合函数

## 认识聚合函数

聚合函数用于统计；

使用聚合函数可以求一张表的有多少条记录、某个字段字的所有总和、某个字段所有值 的平均值、某个字段的最大值或某个字段的最小值等。

## 常用的聚合函数

常用的聚合函数有：

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名称 | 作用 | 函数名称 | 作用 |
| COUNT( ) | 返回表中有多少条记录 | MAX( ) | 返回某最大值 |
| SUM( ) | 返回某列值的和 | MIN( ) | 返回某列最小值 |
| AVG( ) | 返回某列值的平均值 |  |  |

1）COUNT( )函数：

COUNT( )函数用于统计记录的条数，其语法格式为：

**SELECT COUNT(\*) FROM 表名 ;**

2）SUM( )函数：

SUM( )函数用于求出表中某个字段的所有值的总和，其语法格式为：

**SELECT SUM(字段名) FROM 表名 ;**

1. AVG( )函数：

AVG( )函数用于求出表中某个字段所有值的平均值，其语法格式为：

**SELECT AVG(字段名) FROM 表名 ;**

1. MAX( )函数：

MAX( )函数用于求某个字段的最大值，其语法格式为：

**SELECT MAX(字段名) FROM 表名 ;**

1. MIN( )函数：

MIN( )函数用于求出某个字段的最小值，其语法格式为：

**SELECT MIN(字段名) FROM 表名 ;**

# 结果处理

## DISTINCT关键字

DISTINCT关键字用于过滤查询结果中重复的记录。其语法格式为：

**SELECT DISTINCT \*|字段1,字段2,… FROM 表名 ;**

只有DISTINCT关键字后指定的多个字段的值都相同时，才会被认为是重复记录。

重复记录指的是在指定字段的基础上两个记录的每个字段值都相同时，才会被认为是重 复记录。

## 对查询结果进行排序

可以使用ORDER BY对查询结果按照某个字段或某几个字段进行排序。其语法格式为：

**SELECT \*|字段名1,字段名2,... FROM 表名**

**ORDER BY 字段名1 [ASC|DESC] , 字段名2 [ASC|DESC],...... ;**

1）ASC表示表示按照升序进行排序、DESC表示按照降序进行排序；默认情况下， 按照ASC方式排序。

2）可以指顶按照多个字段进行排序，按照多个字段进行排序时，首先会使用按照第一 个字段进行排序，如果两个值都相同，则会按照第二个字段进行排序。

3）排序规则中NULL值被认为是最小值。

## 分组查询

分组查询可以将查询结果按照某个字段或多个字段中的值进行分组，字段中值相同的为 一组，其语法格式为：

**SELECT \*|字段名1,... FROM 表名**

**GROUP BY 字段名1,... [HAVING 条件表达式];**

1）分组查询一般和聚合函数一起使用，如果是只使用分组查询，则查询结果只会显示 分组后每组中的第一条记录，这样做的意义不大。

2）HAVING用于查询的结果进一步过滤。HAVING关键字和WHERE关键字相同，都用于 设置条件表达式对查询结果进行过滤，两者的区别在于，HAVING关键字后可以跟聚 合函数，而WHERE关键字不能，通常情况下HAVING关键字和GROUP BY一起使用。

## 限制查询结果的数量

使用LIMIT关键字可以指定查询结果从哪一条记录开始显示并指定要显示多少条记录。 其语法格式为：  
 **SELECT \*|字段名1,字段名2,… FROM 表名**

**LIMIT [OFFSET, ] n ;**

1. OFFSET为可选参数，表示偏移量，用于控制记录从哪一条开始显示。如果偏移量为 0，则从查询结果的第一条记录开始显示，如果偏移量为1，则从查询结果的第二条 记录开始显示，以此类推。默认情况下为0。
2. n表示查询结果中要显示多少条记录数。
3. LIMIT一般与ORDER BY结合使用。

# 使用别名

## 为表取别名

在查询操作中，如果表名很长则使用起来就不太方便，此时可以为表取一个别名，用别 名来代替表的名称。其语法格式为：

**SELECT \* FROM 表名 [AS] 别名 ;**

AS可写可不写。

## 为字段取别名

在查询操作中，为了让显示查询结果更加直观，可以在SELECT后指定字段时为字段取 一个别名，以更加凸显此字段所表示的含义。其语法格式为：

**SELECT 字段1 [AS] 别名 , 字段2 [AS] 别名,… FROM 表名;**

AS可写可不写。

为字段取别名后，在查询结果中将以别名代替原先字段名进行显示。