

### 04\_数据查询语言DQL(单表)

2024全新MySQL企业开发版





- 1 DQL概述和作用
- **基本SELECT查询语句**
- 3 运算符使用
- 单行和多行函数
- 5 高级查询处理



## 1 DQL概述和作用

### 1 DQL概述和作用



#### 数据查询语句(DQL)作用

#### 查询

#### 数据查询语句(DQL)介绍

数据查询语句(Data Query Language)。

DQL 用于查询数据库数据操作。

DQL 不影响库表结构,也不会影响原表的数据。

DQL 会基于**原表数据**查询出一个**虚拟表**。

DQL关键

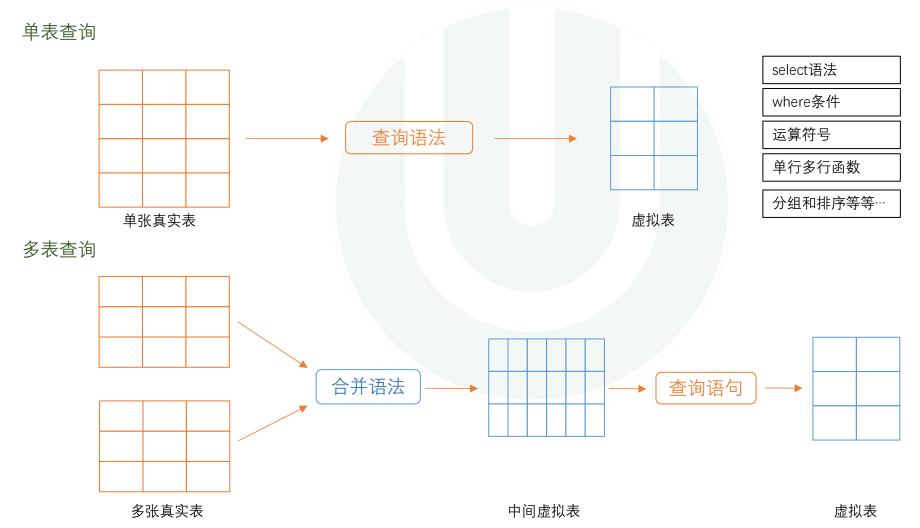
SELECT: 用于查询数据的关键字。

字

### 1 DQL概述和作用



#### 数据操作语句(DQL)语法分类



select语法

where条件

运算符号

单行多行函数

分组和排序等等···

多表数据合并语法





- 1 DQL概述和作用
- **基本SELECT查询语句**
- 3 运算符使用
- 单行和多行函数
- 5 高级查询处理





#### 2.1 基础SELECT语法(不指定条件)

情况1: 非表查询

SELECT 1; SELECT 9/2; SELECT VERSION();

类似Java控制台输出,直接输出结果。

情况2: 指定表

SELECT 列名1, 列名2, 列名3 FROM 表名;

或者

SELECT 表名.列名,表名.\* FROM 表名;

指定表,查询表中的全部或者某些列。

列和列之间使用[,]分割,如果是全部列可以使用\*替代

情况3: 查询列起别名

SELECT 列名1 as 别名, 列名2, 列名3 as 别名 FROM 表名;

或者

SELECT 列名1 别名, 列名2, 列名3 别名 FROM 表名;

查询列可以起别名, as 可以省略。

起别名的意义主要是简化列名或者对应后期Java数据属性等如果别名想要区分大小写,可以添加双引号 例如:"Name"

情况4: 去除重复行

SELECT DISTINCT 列名 [,列名,列名] FROM 表名;

指定列值去重复行,可以指定单列或者多列,但是DISTINCT关键字只写一次且在前面。

情况5: 查询常数

SELECT '尚硅谷' as corporation, 列名, 列名… FROM 表名;

SELECT 查询还可以对常数进行查询。就是在 SELECT 查询结果中增加固定的常数列。这列的取值是我们指定的,而不是从数据表中动态取出的。



#### 2.2 基础SELECT练习

1.准备表

代码和答案位置:资料/代码/04\_数据库DQL.txt文件

#### 2.练习题

- #1. 查询所有员工信息
- # 2. 查询所有员工信息,并且添加一列 etype,值固定为`总部`
- #3. 查询所有员工姓名和工资以及工作地址
- # 4. 查询所有员工姓名,月薪和年薪(年薪等于月薪\*12,结果列字段为 姓名, 月薪, 年薪)
- # 5. 查询所有员工姓名, 月薪, 每月奖金, 每月总收入(结果列字段为姓名, 月薪, 奖金, 月总)
- #6. 查询所有员工一共有几种薪资



#### 2.3 显示表结构

情况1: 使用命令查看表结构

**DESCRIBE** employees;

或

**DESC** employees;

在没有可视化工具场景下可以查看表结构。 其中,各个字段的含义分别解释如下:

Field:表示字段名称。 Type:表示字段类型。

Null:表示该列是否可以存储NULL值。

Key:表示该列是否已编制索引。

PRI表示该列是表主键的一部分;

UNI表示该列是UNIQUE索引的一部分; MUL表示在列中某个给定值允许出现多次。

Default: 表示该列是否有默认值,如果有,那么值是多少。

Extra: 表示可以获取的与给定列有关的附加信息,例如AUTO\_INCREMENT等

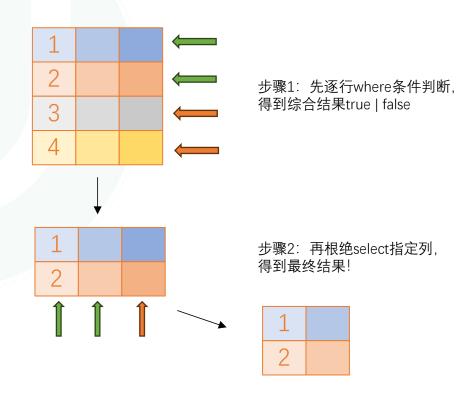
#### 2.4 过滤数据(条件查询)

情况1: 非全表,添加过滤条件

SELECT 字段1,字段2 FROM 表名

WHERE 过滤条件;

where添加以后,就不是全表查询,先过滤条件,符合,再返回指定列。





#### 2.5 基础SELECT练习

1.准备表

代码和答案位置:资料/代码/04\_数据库DQL.txt文件

#### 2.练习题

#1. 查询员工表结构, 并分析

# 2. 查询工资高于9000的员工信息 # 3. 查询年薪高于200000的员工信息

# 4. 查询工资高于8000且性别为女的员工信息





- 1 DQL概述和作用
- **基本SELECT查询语句**
- 3 运算符使用
- 单行和多行函数
- 5 高级查询处理



### U 尚硅谷 www.atguigu.com

#### 3.1 算数运算符

算术运算符主要用于数学运算,其可以连接运算符前后的两个数值或表达式,对数值或表达式进行加(+)、减(-)、乘(\*)、除(/)和取模(%)运算。

运算符	描述
%, MOD	模运算符
*	乘法运算符
+	加法运算符
-	减号运算符
/	浮点除法
DIV	整数除法

注意: 运算表达式可以应用在select列位置或者where条件后!

#### 情况1:基本的加减乘除使用

SELECT 100, 100 + 0, 100 - 0, 100 + 50, 100 + 50 -30, 100 + 35.5, 100 - 35.5, 3\*5, 3\*5.0, 3/5, 100/0,5 DIV 2, -5 DIV 2, 100 /(1-1);

结果-> 100, 100, 100, 150, 120, 135.5 ,64.5, 15, 15.0, 0.6 , null , 2 , -2 , null

注意: 运算符优先级与我们之前学习一致,提高优先级可以使用()\*如果有浮点,会保留浮点类型,参数都是整数结果整数/DIV的区别,/浮点除法,DIV整除除法如果出现除以零的情况,通常会返回 NULL,而不是抛出错误

#### 情况2: 模运算符号使用

SELECT 5 % 2 , 5 MOD 2 , -5 % 2 , -5 MOD 2 ; 结果->1, 1 , -1, -1

注意: 两种取模结果是等效的

%, MOD都是关系型数据库的标准用法, 所以喜欢哪个用哪个!



#### 3.2 算数运算符练习

1.准备表

代码和答案位置:资料/代码/04\_数据库DQL.txt文件

#### 2.练习题

# 1. 查询薪资奖金和大于20000的员工信息 # 2. 查询薪资减去奖金的差小于8000的员工信息 # 3. 查询所有员工的姓名,工资,奖金数额 # 4. 查询员工编号是偶数的员工信息

#### 3.3 比较运算符

名字	描述
>	大于运算符
>=	大于或等于运算符
<	小于操作符号
<=	小于或等于运算符
<>, !=	不相等的运算符,(!= 非标准)
=	相等运算符
<=>	NULL 安全等于运算符,( <mark>非标准)</mark>
IS NULL	NULL 值测试
IS NOT NULL	NOT NULL 值测试
BETWEEN AND	值是否在值范围内
NOT BETWEEN AND	值是否不在值范围内
IN()	值是否在一组值中
NOT IN()	值是否不在一组值中
LIKE	简单的模式匹配 (模糊等于)
NOT LIKE	否定简单的模式匹配

注意: 比较运算符的结果为 1、0、null, 1 代表true, 0和null代表false 操作适用于数字和字符串。根据需要,字符串会自动转换为数字,数字会自动转换为字符串,不需要考虑类型 '1' = 1 为true



#### 情况1: 等于对比

```
mysql> SELECT 1 = 0; -> 0
mysql> SELECT '0' = 0; -> 1
mysql> SELECT '0.0' = 0; -> 1
mysql> SELECT '.01' = 0.01; -> 1
mysql> SELECT 1 <=> 1, NULL <=> NULL, 1 <=> NULL; -> 1, 1, 0
mysql> SELECT 1 <=> 1, NULL IS NULL, 1 IS NULL; -> 1, 1, 0
mysql> SELECT 1 = 1, NULL = NULL, 1 = NULL; -> 1, NULL, NULL
```

注意: <=>进行null在=的基础上,添加了null判断,但是他是`方言`推荐使用is null 或者 is not null。

#### 情况2: 不等于运算使用

```
mysql> SELECT '.01' <> '0.01'; -> 1
mysql> SELECT .01 <> '0.01'; -> 0
mysql> SELECT 'zapp' <> 'zappp'; -> 1
```

注意: 如果双方都是字符串,就按照字符串比较,不会转成数字比较值

#### 情况3: 区间运算使用

```
# 表达式 expr BETWEEN min AND max
# 等效于表达式 (min <= expr AND expr <= max)
mysql> SELECT 2 BETWEEN 1 AND 3, 2 BETWEEN 3 and 1; -> 1, 0
mysql> SELECT 1 BETWEEN 2 AND 3; -> 0
mysql> SELECT 'b' BETWEEN 'a' AND 'c'; -> 1
mysql> SELECT 2 BETWEEN 2 AND '3'; -> 1
```

注意: 字符串就是按照ascii编码排序 固定第一个是min值,第二个是max值,不会自动转化 有字符串和数字,会自动值转化

#### 3.3 比较运算符

名字	描述
>	大于运算符
>=	大于或等于运算符
<	小于操作符号
<=	小于或等于运算符
<>, !=	不相等的运算符,(!= 非标准)
=	相等运算符
<=>	NULL 安全等于运算符,( <mark>非标准)</mark>
IS NULL	NULL 值测试
IS NOT NULL	NOT NULL 值测试
BETWEEN AND	值是否在值范围内
NOT BETWEEN AND	值是否不在值范围内
IN()	值是否在一组值中
NOT IN()	值是否不在一组值中
LIKE	简单的模式匹配 (模糊等于)
NOT LIKE	否定简单的模式匹配

注意: 比较运算符的结果为 1、0 、null, 1 代表true, 0和null代表false 操作适用于数字和字符串。根据需要,字符串会自动转换为数字,数字会自动转换为字符串,不需要考虑类型 '1' = 1 为true



#### 情况4: 组范围对比

```
# expr IN (value,...)
mysql> SELECT 2 IN (0,3,5,7); -> 0
mysql> SELECT 'wefwf' IN ('wee','wefwf','weg'); -> 1
mysql> SELECT (3,4) IN ((1,2), (3,4)); -> 1
mysql> SELECT (3,4) IN ((1,2), (3,5)); -> 0
mysql> SELECT 'a' NOT IN ('a','b','c'), 1 NOT IN (2,3); -> 0,1
```

注意: not in用法和in一样,他只是不等于范围内

#### 情况5: 简单匹配模式

LIKE运算符主要用来<mark>匹配字符串</mark>,通常用于<mark>模糊匹配</mark>,如果满足条件则返回 1,否则返回0。如果给定的值或者匹配条件为NULL,则返回结果为NULL。

#### LIKE运算符通常使用如下通配符:

"%": 匹配0个或多个字符。 "\_": 只能匹配一个字符。

#### 演示

```
# 第三个是___
mysql> SELECT 'aaa' LIKE '%', 'aaa' LIKE '_', 'aaa' LIKE '__'; -> 1,0,1
mysql> SELECT 'abc' LIKE '_a%', 'abc' LIKE 'a__', 'abc' LIKE '%c'; -> 0,1,1
```

#### 情况6: 特殊情况扩展

#### #多列一次对比

(a, b) = (x, y) 等同于 (a = x) AND (b = y)

(a, b) <=> (x, y) 等同于 (a <=> x) AND (b <=> y)

(a, b) <> (x, y) 等同于 (a <> x) OR (b <> y)

#### U) 尚硅谷 www.atguigu.com

#### 3.4 比较运算符练习

#### 1.准备表

代码和答案位置:资料/代码/04\_数据库DQL.txt文件

#### 2.练习题

- #1.查询员工编号为1的员工信息。
- # 2. 查询薪资大于5000的员工信息。
- #3. 查询有奖金的员工信息。
- # 4. 查询出生日期在 '1990-01-01' 和 '1995-01-01' 之间的员工信息。
- #5. 查询性别为女性的员工信息。
- #6. 查询手机号码以 '138' 开头的员工信息。
- #7. 查询邮箱以 '@company.com' 结尾的员工信息。
- #8. 查询地址为NULL的员工信息。
- # 9. 查询工作地点在 '北京'、'上海' 或 '深圳' 的员工信息。
- #10. 查询员工姓名以 '张' 开头的员工信息。
- # 11. 查询出生日期不在 '1980-01-01' 和 '2000-01-01' 之间的员工信息。
- #12. 查询性别不是'男'的员工信息。
- #13. 查询员工编号为奇数的员工信息。



#### 3.5 逻辑运算符

逻辑运算符主要用来判断表达式的真假,在MySQL中,逻辑运算符的返回结果为1、O或者NULL。

逻辑运算符	描述
AND, &&	逻辑且
NOT, !	否定值
OR,	逻辑或
XOR	逻辑异或

注意: MySQL将任何非零, 非 NULL 值计算为 TRUE!

#### 示例:

```
mysql> SELECT NOT 10; -> 0
mysql> SELECT NOT 0; -> 1
mysql> SELECT 1 AND 1; -> 1
mysql> SELECT 1 AND 0; -> 0
mysql> SELECT 1 OR 1; -> 1
mysql> SELECT 1 OR 0; -> 1
mysql> SELECT 0 OR 0; -> 0
mysql> SELECT 1 XOR 1; -> 0
mysql> SELECT 1 XOR 1; -> 0
mysql> SELECT 1 XOR 0; -> 1
mysql> SELECT 1 XOR 0; -> 1
```

注意: 逻辑运算符主要做多条件结果运算!

### U) 尚硅谷 www.atguigu.com

#### 3.6 逻辑运算符练习

1.准备表

代码和答案位置:资料/代码/04\_数据库DQL.txt文件

#### 2.练习题

- #1. 查询薪资大于5000并且工作地点为'北京'的员工信息。
- #2. 查询奖金比例为NULL或者地址为NULL的员工信息。
- # 3. 查询出生日期在 '1985-01-01' 之前或者薪资小于4000的员工信息。
- # 4. 查询性别为'男'并且工作地点不为'上海'的员工信息。
- # 5. 查询薪资大于6000或者邮箱以 '@gmail.com' 结尾的员工信息。
- #6.查询工作地点为'上海'或者薪资小于4500的员工信息。
- #7. 查询员工编号为偶数并且地址不为NULL的员工信息。
- #8. 查询性别为'女'或者薪资小于等于5500的员工信息。
- #9. 查询薪资大于5000且工作地点为'北京'或者'上海'的员工信息。
- #10. 查询邮箱中包含字母'b'并且地址为NULL的员工信息。

#### 3.7 运算符优先级

运算符优先级显示在以下列表中,编号越大优先级越高。优先级高先运算!

优先级	运 算 符
1	:=, = (赋值)
2	∥, OR, XOR
3	&&, AND
4	NOT
5	BETWEEN, CASE, WHEN, THEN 和 ELSE
6	= (比较运算符), <=>, >=, >, <=, <, <>, !=, IS, LIKE, REGEXP和IN
7	
8	&
9	<<与>>>
10	-和+
11	*, /, DIV, %和MOD
12	^
13	- (负号) 和~ (按位取反)
14	!

注意: 这是一般情况下的运算符优先级。在实际使用中,如果不确定优先级,可以使用<mark>括号</mark>来明确运算的顺序! ()内优先级最高!





#### 知识点小结

1. 掌握算数运算符和使用!

2. 掌握比较运算符和使用!

3. 掌握逻辑运算符和使用!

4. 理解运算符优先级和掌握()定义优先级!







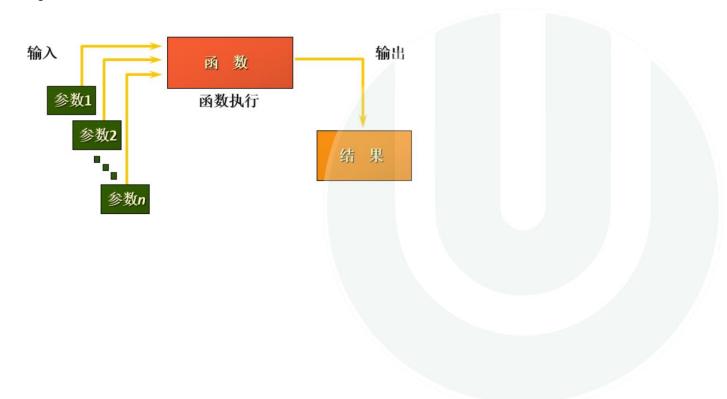
- 1 DQL概述和作用
- **基本SELECT查询语句**
- 3 运算符使用
- 单行和多行函数
- 5 高级查询处理



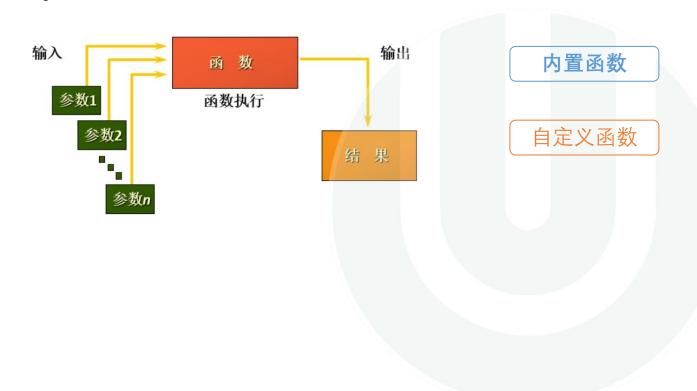




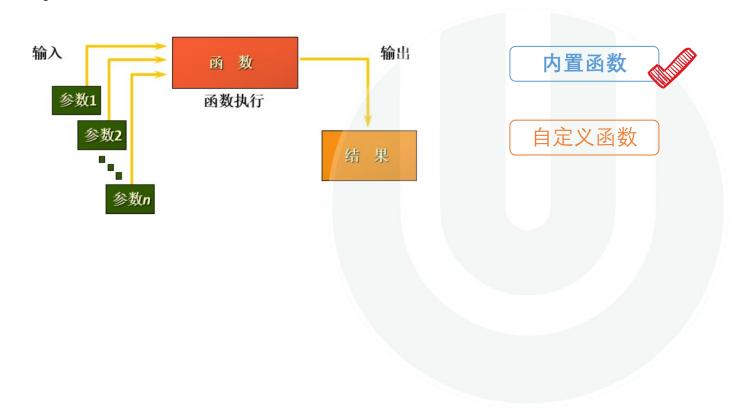




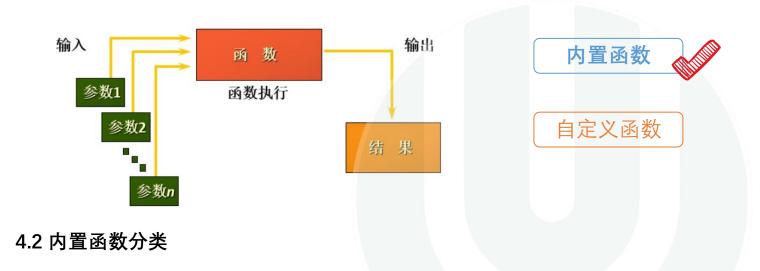






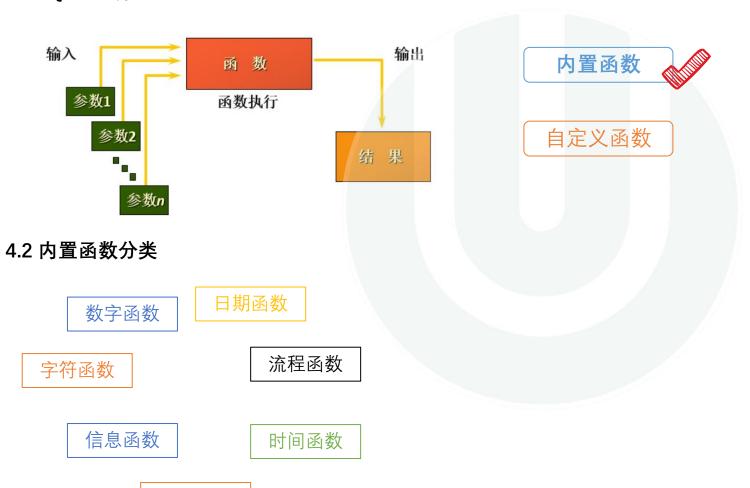






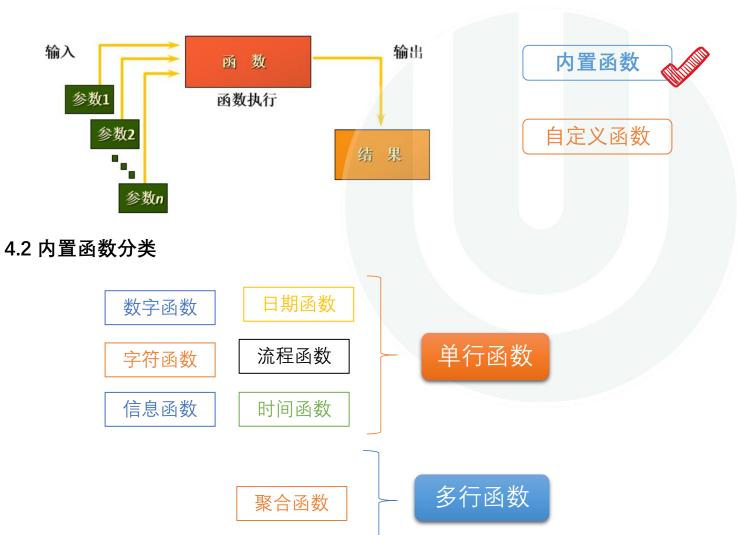


#### 4.1 SQL函数介绍

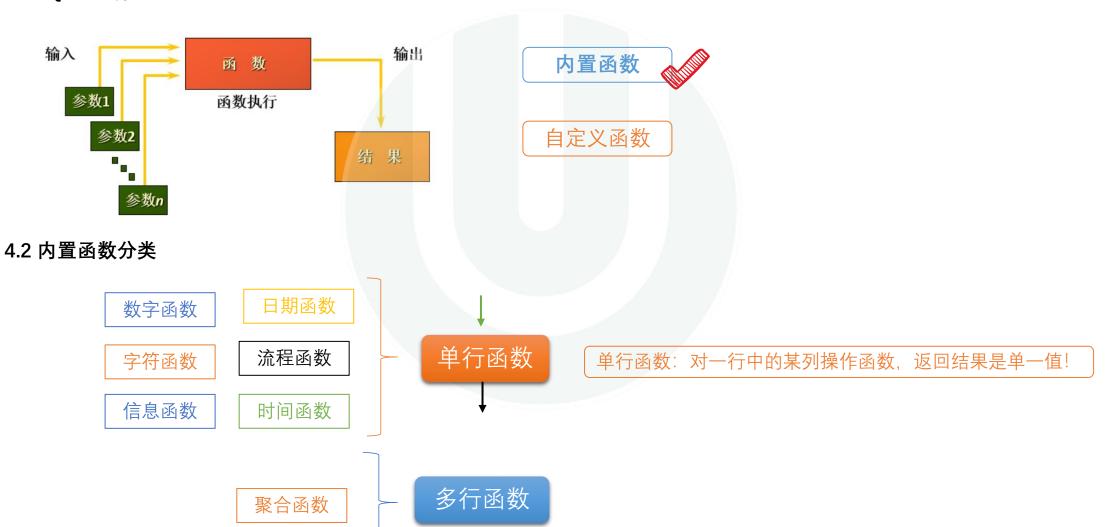


聚合函数

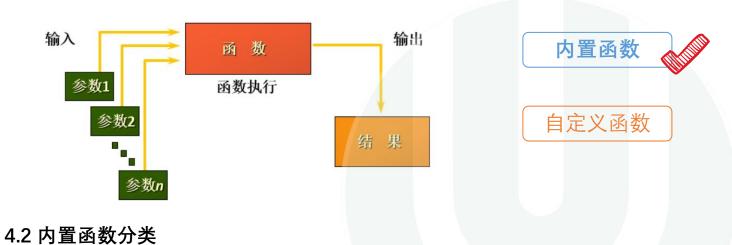
















#### 4.3 单行函数/数值函数

函数	用法
ABS(x)	返回x的绝对值
SIGN(X)	返回X的符号。正数返回1,负数返回-1,0返 回0
PI()	返回圆周率的值
CEIL(x), CEILING(x)	返回大于或等于某个值的最小整数
FLOOR(x)	返回小于或等于某个值的最大整数
LEAST(e1,e2,e3···)	返回列表中的最小值
GREATEST(e1,e2,e3···)	返回列表中的最大值
MOD(x,y)	返回X除以Y后的余数
RAND()	返回0~1的随机值
RAND(x)	返回0~1的随机值,其中x的值用作种子值,相同的X值会产生相同的随机数
ROUND(x)	返回一个对x的值进行四舍五入后,最接近于 X的整数
ROUND(x,y)	返回一个对x的值进行四舍五入后最接近X的值,并保留到小数点后面Y位
TRUNCATE(x,y)	返回数字x截断为y位小数的结果
SQRT(x)	返回x的平方根。当X的值为负数时,返回 NULL

#### 示例:

mysql>SELECT ABS(-123),ABS(32),SIGN(-23),SIGN(43),PI(),CEIL(32.32),CEILING(-43.23),FLOOR(32.32),FLOOR(-43.23),MOD(12,5); -> 123 32 -1 1 3.141593 33 -43 32 -44 2



#### 4.4 单行函数/字符串函数

函数	用法
CHAR_LENGTH(s)	返回字符串s的字符数。作用与 CHARACTER_LENGTH(s)相同
LENGTH(s)	返回字符串s的字节数,和字符集有关
CONCAT(s1,s2,,sn)	连接s1,s2,,sn为一个字符串
INSERT(str, idx, len, replacestr)	将字符串str从第idx位置开始,len个字符长的子串替换为字符串replacestr
REPLACE(str, a, b)	用字符串b替换字符串str中所有出现的字符串a
UPPER(s) 或 UCASE(s)	将字符串s的所有字母转成大写字母
LOWER(s) 或LCASE(s)	将字符串s的所有字母转成小写字母
LEFT(str,n)	返回字符串str最左边的n个字符
RIGHT(str,n)	返回字符串str最右边的n个字符
TRIM(s)	去掉字符串s开始与结尾的空格
SUBSTR(s,index,len)	返回从字符串s的index位置取len个字符,作用与 SUBSTRING(s,n,len)、MID(s,n,len)相同
FIND_IN_SET(s1,s2)	返回字符串s1在字符串s2中出现的位置。其中,字符串s2是一个以逗号分隔的字符串
REVERSE(s)	返回s反转后的字符串
NULLIF(value1,value2)	比较两个字符串,如果value1与value2相等,则返回NULL,否则返回value1

#### 示例:

mysql> SELECT FIND\_IN\_SET('mm', 'hello,mm,amma'); -> 2 mysql> SELECT CONCAT('%', 'hello', '%') -> '%hello%' mysql> SELECT SUBSTR('abcdef',1,2) -> 'ab'

注意: MySQL中, 字符串的位置是从1开始的。



#### 4.5 单行函数/时间函数

函数	用法
CURDATE(), CURRENT_DATE()	返回当前日期,只包含年、月、日
CURTIME(), CURRENT_TIME()	返回当前时间,只包含时、分、秒
NOW() / SYSDATE()	返回当前系统日期和时间
UTC_DATE()	返回UTC(世界标准时间)日期
UTC_TIME()	返回UTC(世界标准时间)时间

#### 示例:

mysql> SELECT CURDATE(),CURTIME(),NOW(),UTC\_DATE(),UTC\_TIME()

注意: UTC即不考虑时区的日期。

函数	用法
YEAR(date) / MONTH(date) / DAY(date)	返回具体的日期值
HOUR(time) / MINUTE(time) / SECOND(time)	返回具体的时间值
MONTHNAME(date)	返回月份:January,
DAYNAME(date)	返回星期几: MONDAY, TUESDAYSUNDAY
WEEKDAY(date)	返回周几,注意,周1是0,周2是 1,。。。周日是6
QUARTER(date)	返回日期对应的季度,范围为1~4
WEEK(date), WEEKOFYEAR(date)	返回一年中的第几周
DAYOFYEAR(date)	返回日期是一年中的第几天
DAYOFMONTH(date)	返回日期位于所在月份的第几天
DAYOFWEEK(date)	返回周几,注意:周日是1,周一是 2,。。。周六是7



#### 4.5 单行函数/时间函数

函数	用法
DATE_ADD(datetime, INTERVAL expr type), ADDDATE(date,INTERVAL expr type)	返回与给定日期时间相差 INTERVAL时间段的日期时间
DATE_SUB(date,INTERVAL expr type), SUBDATE(date,INTERVAL expr type)	返回与date相差INTERVAL时 间间隔的日期

#### 示例:

#### #时间+-运算

mysql>SELECT DATE\_SUB('2021-01-21',INTERVAL 31 DAY) AS col1, SUBDATE('2021-01-21',INTERVAL 31 DAY) AS col2, DATE\_SUB('2021-01-21 02:01:01',INTERVAL '1 1' DAY\_HOUR) AS col3

#### #间隔天数计算

mysql>SELECT

ADDTIME(NOW(),20),SUBTIME(NOW(),30),SUBTIME(NOW(), $^1:1:3$ ),DATEDIF F(NOW(), $^2:021-10-01$ ),

TIMEDIFF(NOW(),'2021-10-25

22:10:10'),FROM\_DAYS(366),TO\_DAYS('0000-12-25'),

LAST\_DAY(NOW()),MAKEDATE(YEAR(NOW()),12),MAKETIME(10,21,23)

函数	用法		
ADDTIME(time1,time2)	返回time1加上time2的时间。当time2 为一个数字时,代表的是秒,可以为 负数		
SUBTIME(time1,time2)	返回time1减去time2后的时间。当 time2为一个数字时,代表的是秒,可 以为负数		
DATEDIFF(date1,date2)	返回date1 - date2的日期间隔天数		
TIMEDIFF(time1, time2)	返回time1 - time2的时间间隔		
FROM_DAYS(N)	返回从0000年1月1日起,N天以后的日期		
TO_DAYS(date)	返回日期date距离0000年1月1日的天 数		
LAST_DAY(date)	返回date所在月份的最后一天的日期		
MAKEDATE(year,n)	针对给定年份与所在年份中的天数返 回一个日期		
MAKETIME(hour,minute,second)	将给定的小时、分钟和秒组合成时间 并返回		

#### 4.5 单行函数/时间函数

函数	用法
DATE_FORMAT(date,fmt)	按照字符串fmt格式化日期date值
TIME_FORMAT(time,fmt)	按照字符串fmt格式化时间time值
STR_TO_DATE(str, fmt)	按照字符串fmt对str进行解析,解析为一 个日期

#### 示例:

```
mysql> SELECT DATE_FORMAT(NOW(), '%H:%i:%s');
mysql> SELECT STR_TO_DATE('09/01/2009','%m/%d/%Y');
mysql> SELECT STR_TO_DATE('2020-01-01 00:00:00','%Y-%m-%d');
```



格式符	说明	格式符	说明
%Y	4位数字表示年份	%y	表示两位数字表示年份
%M	月名表示月份(January,)	<mark>%m</mark>	两位数字表示月份 (01,02,03。。。)
%b	缩写的月名(Jan., Feb.,)	%с	数字表示月份(1,2,3,)
%D	英文后缀表示月中的天数 (1st,2nd,3rd,)	<mark>%d</mark>	两位数字表示月中的天数 (01,02)
%e	数字形式表示月中的天数 (1,2,3,4,5)		
<mark>%H</mark>	两位数字表示小数,24小时 制(01,02)	%h和%l	两位数字表示小时, 12小时 制 (01,02)
%k	数字形式的小时,24小时制 (1,2,3)	%I	数字形式表示小时, 12小时 制(1,2,3,4)
<mark>%i</mark>	两位数字表示分钟(00,01,02)	%S和%s	两位数字表示秒(00,01,02)
%W	一周中的星期名称(Sunday)	%a	一周中的星期缩写(Sun., Mon.,Tues.,)
%W	以数字表示周中的天数 (0=Sunday,1=Monday)		
%j	以3位数字表示年中的天数 (001,002)	%U	以数字表示年中的第几周, (1,2,3。。)其中Sunday为 周中第一天
%u	以数字表示年中的第几周, (1,2,3。。)其中Monday为 周中第一天		
%T	24小时制	%r	12小时制
%р	AM或PM	%%	表示%



#### 4.6 时间函数练习

1.准备表

代码和答案位置:资料/代码/04\_数据库DQL.txt文件

#### 2.练习题

- #1. 查询今天过生日的员工信息
- # 2. 查询本月过生日的员工信息
- # 3. 查询下月过程日的员工信息 # 4. 查询员工姓名,工资,年龄(保留1位小数点)信息 # 5. 查询年龄在25-35之间的员工信息



#### 4.7 单行函数/流程函数

流程处理函数可以根据不同的条件,执行不同的处理流程,可以在SQL语句中实现不同的条件选择。MySQL中的流程处理函数主要包括IF()、IFNULL()和CASE()函数。

IF 函数是一种条件函数,用于在 SQL 查询中执行基本的条件判断。它的语法如下:

IF(condition, true\_value, false\_value)

解释: 当 condition 成立时,返回 true\_value,否则返回 false\_value。

IFNULL函数是 MySQL 中的一个函数,用于处理 NULL 值。它的语法如下:

IFNULL(column, null\_value)

解释: 当指定列column值为null, 取null\_value的值作为结果。

#### IF和IFNULL练习:

# 1. 此查询将根据员工的生日,如果生日1990年之前,则薪资涨幅为当前薪资的 10%,否则为 5%。

#2. 查询员工编号和姓名,以及生成一个type列,更具性别显示男员工or女员工。

#3. 查询员工的姓名和工资以及奖金数额(奖金额=salary\*commission\_pct)。



#### 4.7 单行函数/流程函数

CASE 表达式用于实现多条件判断,并根据条件的结果返回不同的值。它的基本语法如下:

```
# 格式1
CASE
WHEN condition1 THEN result1
WHEN condition2 THEN result2
...
ELSE default_result
END [AS alias_name]
# 这种形式中,WHEN 子句后面跟着一个条件,而不是一个具体的值。

# 格式2
CASE expr
WHEN value1 THEN result1
WHEN value2 THEN result2
...
ELSE default_result
END [AS alias_name]
# 这种形式中,expr 是要比较的表达式或列名,然后逐个与 WHEN 子句中的值进行比较。
```

#### CASE表达式练习:

# 4. 查询姓名,性别,以及补助金额(补助按照性别基准值\*commission\_pct),男性基础 2000,女性基准3000,其他0,commission\_pct为null算0.1比例!



#### 4.8 多行函数/聚合函数

聚合函数作用于全部或者分组数据进行统计和计算,最终返回一条结果。例如:统计员工数量、员工平均工资,每个部门的人数等!

eid	ename	salary
1	孙洪亮	28000
8	李晨熙	9000
22	刘烨	13099
14	谢吉娜	18978
7	贾宝玉	15700
2	何进	7001
18	舒淇格	16788
20	章嘉怡	15099
3	邓超远	8000
4	黄熙萌	9456
5	陈浩	8567
6	韩庚年	12000
9	李易峰	7897
10	陆风	8789
11	黄冰茹	15678
12	孙红梅	9000
13	李冰冰	18760
15	萱吉祥	8978
16	彭超越	9878
17	李诗雨	9000
19	周旭飞	7876
21	白露	9787
23	陈纲	13090
24	吉日格勒	10289
25	额日古那	9087
26	李红	5000
27	周洲	8000

计算平均工资 计算总工资 计算最大工资 计算最小工资

函数	用法
AVG(列名)	计算某一列的平均值(数值类型)
SUM(列名)	计算某一列的和(数值类型)
MIN(列名)	计算某一列的最小值(任意类型)
MAX(列名)	计算某一列的最大值(任意类型)
COUNT(列名/*/1)	计算某一列或者行的记录数(任意类型)

注意:聚合函数不能嵌套调用。比如不能出现类似"AVG(SUM(字段名称))"形式的调用。



函数	用法
AVG(列名)	计算某一列的平均值(数值类型)
SUM(列名)	计算某一列的和(数值类型)
MIN(列名)	计算某一列的最小值(任意类型)
MAX(列名)	计算某一列的最大值(任意类型)
COUNT(列名 /*/1)	计算某一列或者行的记录数(任意类型)

注意。聚合函数不能嵌套调用。比如不能出现类似



#### 示例1: AVG和SUM函数

# 求平均工资,最小和最大工资以及总工资 SELECT AVG(salary) ,MIN(salary),MAX(salary),SUM(salary) FROM t\_employee;

#### 示例2: MIN和MAX函数

# 求最大年龄和最小年龄的员工生日 SELECT MIN(birthday),MAX(birthday),MIN(ename) FROM t\_employee;

注意:对于日期/时间类型,它会考虑日期/时间的顺序;对于字符串类型, MySQL 默认使用的字符集是 utf8mb4,字符编码是 Unicode。在 Unicode 中,字符按照其对应的 Unicode 编码值进行排序

#### 示例3: COUNT函数

# 求总员工数和有奖金的员工数 SELECT COUNT(\*),COUNT(1),COUNT(commission\_pct) FROM t\_employee;

注意: count(\*)和count(1)都是一行有任意列有数据就会统计,count(列名)统计指定列出现的次数,为null的不统计。

<sup>&</sup>quot;AVG(SUM(字段名称))"形式的调用。



#### 4.8 多行函数/聚合函数

函数	用法			
AVG(列名)	计算某一列的平均值(数值类型)			
SUM(列名)	计算某一列的和(数值类型)			
MIN(列名)	计算某一列的最小值(任意类型)			
MAX(列名)	计算某一列的最大值(任意类型)			
COUNT(列名 /*/1)	计算某一列或者行的记录数(任意类型)			

<del>|注意:聚合函数不能嵌套调用。比如不能出现类似</del>

问题1: avg/sum/min/max/count碰到NULL的态度?

问题2: 能不能使用count(列名)替换count(\*)?

<sup>&</sup>quot;AVG(SUM(字段名称))"形式的调用。



#### 知识点小结

1. 理解函数作用!

2. 理解多行和单行函数的区别!

3. 掌握重点单行和多行函数的使用!





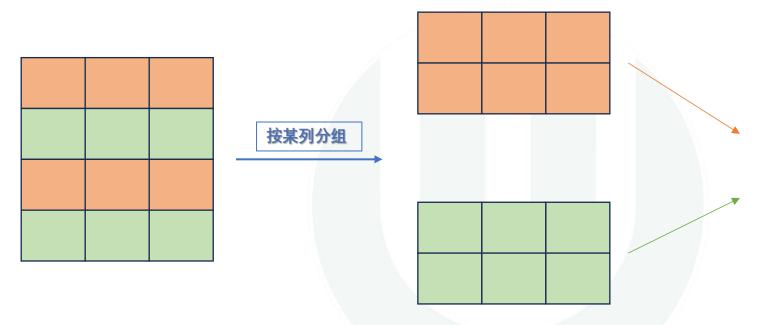


- 1 DQL概述和作用
- **基本SELECT查询语句**
- 3 运算符使用
- 单行和多行函数
- 5 高级查询处理





#### 5.1 分组查询



1.当前组数据统计(分组列,聚合函数)

2.当前组数据统计(分组列,聚合函数)

#### 分组查询概念

先将数据行,按照某一或者多特性列进行分组,最后查询<mark>每组的特性</mark>,分组查询的结果只能是分组特性列或者聚合函数!

SELECT 分组列, 分组列, 聚合函数

**FROM** table

[WHERE condition]

[GROUP BY 分组列,分组列··· HAVING 分组后条件]



#### 5.1 分组查询

- 1. 查询每种性别的员工数量以及性别平均工资
- 2. 查询生日年份、性别相同的人数和平均工资
- 3. 查询工资高于5000, 每种性别的员工数量以及性别平均工资
- 4. 查询平均工资高于5000的性别和性别人数

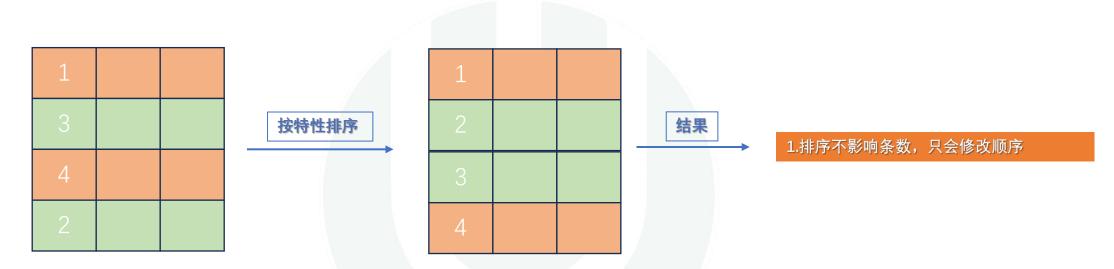
#### 分组过滤条件HAVING:

- 1.分组后条件过滤。
- 2.一般使用聚合函数。
- 3.满足HAVING 子句中条件的分组将被显示。
- 4.HAVING 不能单独使用,必须要跟 GROUP BY 一起使用。

问题: WHERE和HAVING有什么区别?



#### 5.2 排序查询



#### 排序查询概念

按照某一或者多特性列进行数据排序,不会影响结果条数,只是改变结果排序!例如:商品按照价格排序等等

SELECT 列, 列, 函数

**FROM** table

[WHERE condition]

[ORDER BY 排序列 ASC|DESC, 排序列 ASC|DESC ·····]

ASC为正序(默认值),DESC为倒序 多列排序,只有第一列相同,第二列才会生效以此类推······

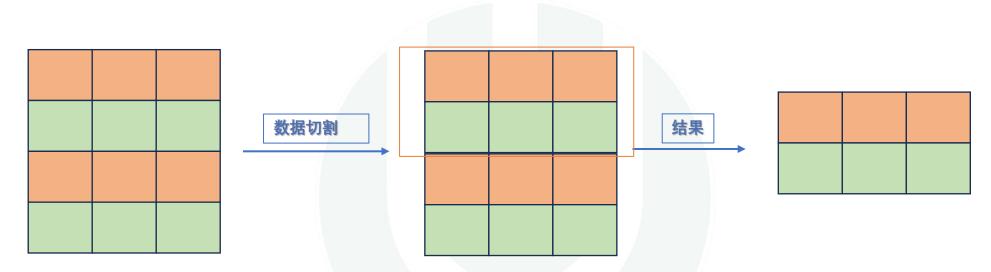


#### 5.2 排序查询

- 1. 按照年龄正序排序,查询员工信息
- 2. 按照工资倒序,查询员工信息
- 3. 按照工资倒序,如果工资相同,按照年龄正序排序查询员工信息
- 4. 查询有奖金的员工,最终按照工资倒序显示员工信息



#### 5.3 数据切割(分页查询)



#### 数据切割(分页查询)概念

将结果,进行分页切割,按照指定的区域一段一段的进行展示!

例如: 商品分页展示、查询工资前三的员工等

#### SELECT 列, 列, 函数

**FROM** table

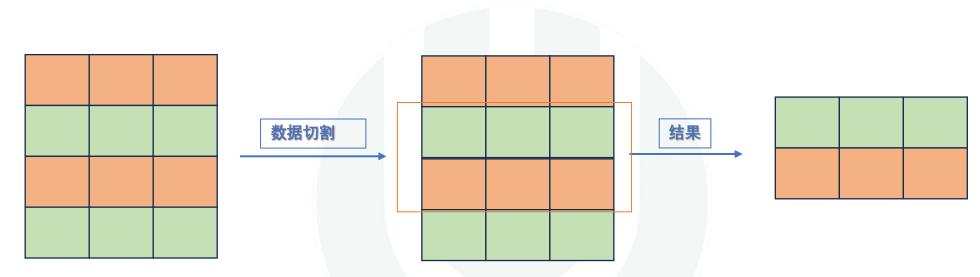
[WHERE condition]

[LIMIT [位置偏移量,] 行数]

位置偏移量:可选参数,位置偏移量,不写默认是O,代表不偏移; 行数:指示返回的记录条数。



#### 5.3 数据切割(分页查询)



#### 数据切割(分页查询)概念

将结果,进行分页切割,按照指定的区域一段一段的进行展示!

例如: 商品分页展示、查询工资前三的员工等

#### SELECT 列, 列, 函数

**FROM** table

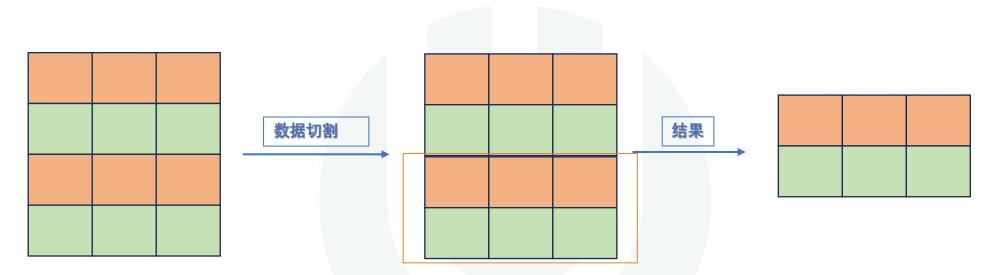
[WHERE condition]

[LIMIT [位置偏移量,] 行数]

位置偏移量:可选参数,位置偏移量,不写默认是O,代表不偏移; 行数:指示返回的记录条数。



#### 5.3 数据切割(分页查询)



#### 数据切割(分页查询)概念

将结果, 进行分页切割, 按照指定的区域一段一段的进行展示!

例如: 商品分页展示、查询工资前三的员工等

#### SELECT 列, 列, 函数

**FROM** table

[WHERE condition]

[LIMIT [位置偏移量,] 行数]

位置偏移量:可选参数,位置偏移量,不写默认是0,代表不偏移; 行数:指示返回的记录条数。

LIMIT必须放在整个SELECT语句的最后



#### 5.3 数据切割(分页查询)

1. 查询工资最高的员工信息

2. 查询工资第二高的员工信息

3. 查询工资最高的女性员工信息

分页显式公式: (**当前页数-1**)\*每页条数,每页条数

SELECT \* FROM table
LIMIT(PageNo - 1)\*PageSize,PageSize;



#### 5.4 SELECT语句执行过程

#### 单表查询结构

SELECT ...,...,....
FROM ...,...,...
WHERE 多表的连接条件
AND 不包含组函数的过滤条件
GROUP BY ...,...
HAVING 包含组函数的过滤条件
ORDER BY ... ASC/DESC
LIMIT ...,...

#### 关键字的顺序是不能颠倒的:

SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ... ORDER BY ... LIMIT...

#### SELECT 语句的执行顺序:

FROM -> WHERE -> GROUP BY -> HAVING -> SELECT 的字段 -> ORDER BY -> LIMIT



#### 5.4 SELECT语句执行过程

#### 示例1:

SELECT gender, YEAR (birthday) AS br, COUNT(\*)
FROM t\_employee
WHERE br > 2000;

#### 示例2:

SELECT gender, YEAR (birthday) AS **br**, COUNT(\*) **ct**FROM t\_employee
GROUP BY gender, **br**HAVING **ct** > 1;

标准 SQL 不允许在 WHERE 子句中引用列别名。这个限制是由于在评估 WHERE 子句时,列的值可能尚未确定。 别名可以在查询的选择列表中为列指定不同的名称。您可以在 GROUP BY、ORDER BY 或 HAVING 子句中使用别名来引用列

https://dev.mysql.com/doc/refman/8.0/en/problems-with-alias.html



#### 5.5 单表综合练习

- #1、创建数据库test04 lib
- #2、创建表 books. 表结构如下
- #3、向books表中插入记录
- # 1) 不指定字段名称, 插入第一条记录
- # 2) 指定所有字段名称,插入第二记录
- # 3) 同时插入多条记录(剩下的所有记录)
- #4 将小说类型(novel)的书的价格都增加5。
- #5、将名称为EmmaT的书的价格改为40,并将说明改为drama。
- #6、删除库存为0的记录。
- #7、统计书名中包含a字母的书
- #8、统计书名中包含a字母的书的数量和库存总量
- #9、找出"novel"类型的书、按照价格降序排列
- #10、查询图书信息,按照库存量降序排列,如果库存量相同的按照note升序排列
- #11、按照note分类统计书的数量
- #12、按照note分类统计书的库存量,显示库存量超过30本的
- #13、查询所有图书,每页显示5本,显示第二页
- #14、按照note分类统计书的库存量,显示库存量最多的
- #15、查询书名达到10个字符的书。不包括里面的空格
- #16、查询书名和类型,其中note值为novel显示小说,law显示法律,medicine显示
- 医药, cartoon显示卡通, joke显示笑话
- # 17、查询书名、库存,其中num值超过30本的,显示滞销,大于0并低于10的,显示畅销,为0的显示需要无货
- #18、统计每一种note的库存量,并合计总量
- #19、统计每一种note的数量,并合计总量
- #20、统计库存量前三名的图书
- #21、找出最早出版的一本书
- #22、找出novel中价格最高的一本书
- #23、找出书名中字数最多的一本书,不含空格

字段名	字段说明	数据类型
id	书编号	INT
name	书名	VARCHAR(50)
authors	作者	VARCHAR(100)
price	价格	FLOAT
pubdate	出版日期	YEAR
note	说明	VARCHAR(100)
num	库存	INT

id	name	authors	price	pubdate	note	num
1	Tal of AAA	Dickes	23	1995	novel	11
2	EmmaT	Jane lura	35	1993	joke	22
3	Story of Jane	Jane Tim	40	2001	novel	0
4	Lovey Day	George Byron	20	2005	novel	30
5	Old land	Honore Blade	30	2010	law	0
6	The Battle	Upton Sara	30	1999	medicine	40
7	Rose Hood	Richard haggard	28	2008	cartoon	28

## 感谢观看

