

一、常见函数

1. 功能: 类似于java中的方法

2. 优点

1. 概念: 类似于java的方法, 将一组逻辑语句封装在方法体中, 对外暴露方法名.

2. 优点: ① 隐藏了实现细节.
② 提高代码的重用性

3. 调用: select 函数名(实参列表) from 表

4. 特点: ① 叫什么(名称)
② 干什么(函数功能)

3. 分类

5. 分类: ① 单行函数

如 concat / length() / ifnull() 等.

② 多行函数.

功能: 做统计使用, 又称统计函数. 聚合函数. 组函数.

二、单行函数

1. 字符函数

```
#1.length(): 获取参数值的字节个数
SELECT LENGTH('john');

#2.concat(): 拼接字符串
SELECT CONCAT(last_name, '_', first_name) FROM employees;

#3.upper() \ lower()
SELECT UPPER('john');
SELECT LOWER('john');

#4.substr / substring()
#注意: 索引从1开始
#6截取从指定索引处后面的所有字符
SELECT SUBSTR('李奥慈爱上了陆展元', 7) out_put;
#6截取从指定索引处指定长度的字符
SELECT SUBSTR('李奥慈爱上了陆展元', 1, 3) out_put;
#案例: 姓名中首字母大写, 其他字母小写, 然后用_拼接, 显示出来
SELECT CONCAT(SUBSTR(last_name, 1, 1), '_', LOWER(SUBSTR(last_name, 2))) out_put FROM employees;

#5.instr(): 返回子串第一次出现的索引, 如果找不到返回0
SELECT INSTR('杨不悔爱上了股六侠', '股六侠') AS out_put;

#6.trim(): 去除前后的指定字符
SELECT TRIM(' 张翠山 ') AS out_put;
SELECT TRIM('a' FROM 'aaaa张aaaa翠山aaaaa') AS out_put;

#7.lpad(): 用指定的字符左填充指定长度
SELECT LPAD('股素素', 10, '**') AS out_put;

#8.rpad(): 用指定的字符右填充指定长度
SELECT RPAD('股素素', 10, 'ab') AS out_put;

#9.replace():
SELECT REPLACE('张无忌爱上了周芷若', '周芷若', '赵敏');
```

2. 数学函数

```
#round 四舍五入
SELECT ROUND(-1.55);
SELECT ROUND(1.567, 2);

#ceil 向上取整, 返回>=该参数的最小整数
SELECT CEIL(-1.02);

#floor 向下取整, 返回<=该参数的最大整数
SELECT FLOOR(-9.99);
```

```
#truncate 截断
SELECT TRUNCATE(1.69999, 1);

#mod 取余 I
Q/*
mod(a, b) : a-a/b*b
mod(-10, -3) : -10 - (-10)/(-3) * (-3) = -1
*/
SELECT MOD(10, -3);
SELECT 10%3;
```

3. 日期函数

```
#now 返回当前系统日期+时间
SELECT NOW();

#curdate 返回当前系统日期, 不包含时间
SELECT CURDATE();

#curtime 返回当前时间, 不包含日期
SELECT CURTIME();
```

```
#可以获取指定的部分, 年、月、日、小时、分钟、秒
SELECT YEAR(NOW()) 年;
SELECT YEAR('1998-1-1') 年;

SELECT YEAR(hiredate) 年 FROM employees;

SELECT MONTH(NOW()) 月;
SELECT MONTHNAME(NOW()) 月;
```

```
#查询入职日期为1992--4-3的员工信息
SELECT * FROM employees WHERE hiredate = '1992-4-3';

SELECT * FROM employees WHERE hiredate = STR_TO_DATE('4-3 1992', '%c-%d %Y');
```

```
#date_format 将日期转换成字符串

SELECT DATE_FORMAT(NOW(), '%y年%m月%d日') AS out_put;

#查询有奖金的员工名和入职日期(xx月/xx日 xx年)
SELECT last_name, DATE_FORMAT(hiredate, '%m月/%d %y年') 入职日期
FROM employees
WHERE commission_pct IS NOT NULL;
```

序号	格式符	功能
1	%Y	四位的年份
2	%y	2位的年份
3	%m	月份 (01,02...11,12)
4	%c	月份 (1,2...11,12)
5	%d	日 (01,02...)
6	%h	小时 (24小时制)
7	%h	小时 (12小时制)
8	%i	分钟 (00,01...59)
9	%s	秒 (00,01...59)

补充 datediff: 求两日期

之间的时间差.

datediff('1997-1-12', '1995-2-17')

4. 其他函数

```
SELECT VERSION();
SELECT DATABASE();
SELECT USER();
```

5. 流程控制函数

① if函数: if else 的效果

```
/*if函数: if else 的效果
SELECT IF(10<5, '大', '小');

SELECT last_name, commission_pct, IF(commission_pct IS NULL, '没奖金, 呵呵', '有奖金, 嘻嘻') 备注
FROM employees;
```

② case函数使用一: switch case 的效果.

• java中: switch (变量或表达式) {

case 常量1: 语句1; break;

default: 语句n; break;

• MySQL中: case 要判断的字段或表达式

when 常量1 then 要显示值1或语句1;

when 常量2 then 要显示值2或语句2;

...

else 要显示值n或语句n;

end

• 案例:

```
/*案例: 查询员工的工资, 要求
部门号=30, 显示的工资为1.1倍
部门号=40, 显示的工资为1.2倍
部门号=50, 显示的工资为1.3倍
其他部门, 显示的原工资
*/
```

```
SELECT salary 原始工资, department_id,
CASE department_id
WHEN 30 THEN 1.1*salary
WHEN 40 THEN 1.2*salary
WHEN 50 THEN 1.3*salary
ELSE salary
END AS 新工资
FROM employees;
```

③ case函数使用二: 类似于多重if.

• java中: if (条件1) {

语句1;

else if (条件2) {

• MySQL中: case

when 条件1 then 要显示值1或语句1;

when 条件2 then ... 2 ... 2;

```

} 结束;
else {
} when;

```

```

else 字符串或数字n.
end

```

· 案例:

```

/*
案例: 查询员工的工资情况
如果工资>20000, 显示A级别
如果工资>15000, 显示B级别
如果工资>10000, 显示C级别
否则, 显示D级别
*/
SELECT salary,
CASE
WHEN salary>20000 THEN 'A'
WHEN salary>15000 THEN 'B'
WHEN salary>10000 THEN 'C'
ELSE 'D'
END AS 工资级别
FROM employees;

```

4. 单行函数总结

三. 单行函数总结

常见函数:

① 字符串函数

① 字符串函数: length / concat / substr / instr / trim /
upper / lower / lpad / rpad / replace

② 数学函数

② 数学函数: round / ceil / floor / truncate / mod / rand

③ 日期函数

③ 日期函数: now / curdate / curtime / year / month /
monthname / day / hour / minute / second /
str_to_date / date_format / datediff

④ 其他函数

④ 其他函数: version / database / user / password / md5

⑤ 流程控制函数

⑤ 控制函数: if / case

5. 分组函数的功能

四. 分组函数的介绍

6. 语法

1. 功能: 用作统计使用, 又称为聚合函数或统计函数或组函数

7. 特点

2. 分类: sum 求和 / avg 平均值 / max / min / count 计算个数

3. 特点: ① sum / avg 一般用于处理数值型

max / min / count 可处理任意类型

② 以上分组函数均忽略 null 值.

③ 可以和 distinct 关键字配合使用 (去重)

④ count函数的详细介绍

- `Select count(*) from employees;` 和
`Select count(1) from employees;` 均是求行数
- 一般使用 `count(*)` 统计结果集的行数

⑤ 和分组函数同查询的字段要求是 `group by` 后的字段

五. 分组函数的简单使用

```
#1. 简单 的使用
SELECT SUM(salary) FROM employees;
SELECT AVG(salary) FROM employees;
SELECT MIN(salary) FROM employees;
SELECT MAX(salary) FROM employees;
SELECT COUNT(salary) FROM employees;

SELECT SUM(salary) 和, AVG(salary) 平均, MAX(salary) 最高, MIN(salary) 最低, COUNT(salary) 个数
FROM employees;

SELECT SUM(salary) 和, ROUND(AVG(salary), 2) 平均, MAX(salary) 最高, MIN(salary) 最低, COUNT(salary) 个数
FROM employees;
```

