

## 第三章 MySQL 常用函数

### 课前回顾

现有员工表 emp，包含字段员工编号 (no)，类型为整数，长度为20，是主键，自增长，非空；姓名 (name)，类型为字符串，长度为20，非空；性别 (sex)，类型为字符串，长度为2，默认值为"男"；年龄 (age)，类型为整数，长度为3，非空；所属部门 (dept)，类型为字符串，长度为20，非空；薪资 (salary)，类型为浮点数，长度为10，小数点后面保留2位有效数字，非空。

```
CREATE TABLE IF NOT EXISTS emp(  
  `no` BIGINT(20) AUTO_INCREMENT PRIMARY KEY NOT NULL COMMENT '员工编号',  
  name VARCHAR(20) NOT NULL COMMENT '姓名',  
  sex VARCHAR(2) DEFAULT '男' COMMENT '性别',  
  age TINYINT(3) UNSIGNED NOT NULL COMMENT '年龄',  
  dept VARCHAR(20) NOT NULL COMMENT '所属部门',  
  salary DOUBLE(10, 2) NOT NULL COMMENT '薪资'  
)ENGINE=InnoDB CHARSET=utf8 COMMENT '员工表';
```

#### 1. 向员工表插入如下数据：

姓名	性别	年龄	部门	薪资
张三	男	22	研发部	13000
李刚	男	24	研发部	14000
金凤	女	23	财务部	8000
肖青	女	26	财务部	9000
张华	男	28	研发部	15000
董钰	女	24	研发部	12000
吴梅	女	24	测试部	9000
王玲	女	26	测试部	9500

```
INSERT INTO emp(`no`, name ,sex, age, dept, salary) VALUES(DEFAULT, '张三', '男',  
22, '研发部', 13000);  
INSERT INTO emp(name ,sex, age, dept, salary) VALUES('李刚', '男', 24, '研发部',  
14000);  
INSERT INTO emp VALUES(DEFAULT, '金凤', '女', 23, '财务部', 8000);  
INSERT INTO emp(name ,sex, age, dept, salary) VALUES('肖青', '女', 26, '财务部',  
9000), ('张华', '男', 28, '研发部', 15000), ('董钰', '女', 24, '研发部', 12000);  
INSERT INTO emp VALUES(DEFAULT, '吴梅', '女', 24, '测试部', 9000), (DEFAULT, '王玲',  
'女', 26, '测试部', 9500);
```

#### 2. 吴梅因工作出色而被提升为测试主管，薪资调整为11000

```
UPDATE emp SET salary = 11000 WHERE name='吴梅';
```

### 3. 研发部金凤离职

```
DELETE FROM emp WHERE name='金凤';
```

### 4. 从员工表中查询出平均年龄小于25的部门

```
SELECT dept FROM emp GROUP BY dept HAVING AVG(age)<25;
```

### 5. 从员工表中统计研发部的最高薪资、最低薪资、平均薪资和总薪资

```
SELECT MAX(salary), MIN(salary), AVG(salary), SUM(salary) FROM emp WHERE dept='研发部';
```

### 6. 从员工表中统计各个部门的员工数量

```
SELECT dept, COUNT(*) FROM emp GROUP BY dept;
```

### 7. 从员工表中查询薪资在10000以上的员工信息并按薪资从高到低排列

```
SELECT * FROM emp WHERE salary > 10000 ORDER BY salary DESC;
```

### 8. 从员工表中分页查询员工信息，每页显示5条员工信息，按薪资从高到低排列，查询第2页员工信息

```
SELECT * FROM emp ORDER BY salary DESC LIMIT 5, 5;
```

## 章节内容

- 常用数学函数 熟悉
- 常用字符串函数 重点
- 常用日期和时间函数 重点
- 条件判断函数 重点
- 系统信息函数 熟悉

## 章节目标

- 熟悉常用数学函数
- 掌握常用字符串函数
- 掌握常用日期和时间函数
- 掌握条件判断函数
- 熟悉系统信息函数

# 第一节 常用数学函数

函数	说明	示例
ABS(X)	返回X的绝对值。	SELECT ABS(-8);
FLOOR(X)	返回不大于X的最大整数。	SELECT FLOOR(1.3);
CEIL(X)	返回不小于X的最小整数。	SELECT CEIL(1.3);
TRUNCATE(X,D)	返回数值X保留到小数点后D位的值，截断 时不进行四舍五入。	SELECT TRUNCATE(1.2328,3);
ROUND(X)	返回离X最近的整数，截断时要进行四舍五入	SELECT ROUND(1.8);
ROUND(X,D)	保留X小数点后D位的值，截断时要进行四舍五入。	SELECT ROUND(1.2323,3);
RAND()	返回0~1的随机数。	SELECT RAND();
MOD(N,M)	返回N除以M以后的余数。	SELECT MOD(9,2);

# 第二节 常用字符串函数

函数	说明	示例
CHAR_LENGTH(str)	计算字符串字符个数。	SELECT CHAR_LENGTH('中国');
LENGTH(str)	返回值为字符串str的长度，单位为字节。	SELECT LENGTH('中国');
CONCAT(s1,s2, ...)	将多个字符串拼接在一起，其中任意一个 为NULL则返回值为NULL。	SELECT CONCAT('ad','min');
LOWER(str) LCASE(str)	将str中的字母全部转换成小写。	SELECT LOWER('ABC'); SELECT LCASE('ABC');
UPPER(str) UCASE(str)	将字符串中的字母全部转换成大写。	SELECT UPPER('abc'); SELECT UCASE('abc');
LEFT(s,n) RIGHT(s,n)	前者返回字符串s从最左边开始的n个字符 后者返回字符串s从最右边开始的n个字符。	SELECT LEFT('abcdefg', 5); SELECT RIGHT('abcdefg', 5);
LTRIM(s) RTRIM(s)	前者返回字符串s，其左边所有空格被删除 后者返回字符串s，其右边所有空格被删除。	SELECT LTRIM(' abcde '); SELECT RTRIM(' abcde ');
TRIM(s)	返回字符串s删除了两边空格之后的字符串。	SELECT TRIM(' abcde ');
REPLACE(s,s1,s2)	返回一个字符串，用字符串s2替代字符串s中 所有的字符串s1。	SELECT REPLACE('ababac', 'ab', 'd');
SUBSTRING(s,n,len)	从字符串s中返回一个第n个字符开始 长度为len的字符串。	SELECT SUBSTRING('abcdef', 2, 3);
CHAR_LENGTH(str)	计算字符串字符个数。	SELECT CHAR_LENGTH('中国');
LENGTH(str)	返回值为字符串str的长度，单位为字节。	SELECT LENGTH('中国');

## 练习

查询计科和软工各有多少人

```
SELECT LEFT(class, 2), COUNT(*) FROM stu GROUP BY LEFT(class, 2);
```

查询名字有4个字的学生信息

```
SELECT * FROM stu WHERE CHAR_LENGTH(`name`)=4;
```

查询成绩能够被10整除的考试信息

```
SELECT * FROM score WHERE MOD(score, 10)=0;
```

# 第三节 日期和时间函数

函数	说明	示例
CURDATE() CURRENT_DATE()	返回当前日期	SELECT CURDATE();
CURTIME() CURRENT_TIME()	返回当前时间	SELECT CURTIME();
NOW() CURRENT_TIMESTAMP() SYSDATE()	返回当前日期和时间	SELECT NOW();
YEAR(d)	返回日期d中的月份值。范围是1~12	SELECT YEAR(NOW());
MONTH(d)	返回日期d是星期几	SELECT MONTH(NOW());
DAYOFMONTH(d)	返回给定日期d是当月的第几天	SELECT DAYOFMONTH(NOW());
HOUR(d)	返回给定日期d的小时数	SELECT HOUR(NOW());
MINUTE(d)	返回给定日期d的分钟数	SELECT MINUTE(NOW());
SECOND(d)	返回给定日期d的秒数	SELECT SECOND(NOW());
ADDDATE(d, n)	返回起始日期d加上n天的日期	SELECT ADDDATE(NOW(), 3);
TIMESTAMPDIFF(INTERVAL expr type, d1, d2)	返回给定日期 d1 和 d2 的时间差	SELECT TIMESTAMPDIFF(YEAR,'2019-10-10','2021-10-1');
DATE_FORMAT(d,f)	返回给定日期格式的字符串	SELECT DATE_FORMAT(NOW(), '%Y-%m-%d %H:%i:%s');

## 练习

查询年龄在20岁以上的学生信息

```
SELECT * FROM stu WHERE TIMESTAMPDIFF(YEAR, birthday, NOW()) > 20;
```

查询今天过生日的学生信息

```
SELECT * FROM stu WHERE MONTH(birthday)=MONTH(NOW()) AND
DAYOFMONTH(birthday)=DAYOFMONTH(NOW());
```

查询本周过生日的学生信息

```
SELECT * FROM stu WHERE RIGHT(birthday, 5) > RIGHT(DATE_FORMAT(ADDDATE(NOW(), -
DAYOFWEEK(NOW())), '%Y-%m-%d'), 5) AND RIGHT(birthday, 5) <=
RIGHT(DATE_FORMAT(ADDDATE(NOW(), 7-DAYOFMONTH(NOW())), '%Y-%m-%d'), 5);
```

# 第四节 条件判断函数

## 1. IF函数

### 1.1 IF(条件, 表达式1, 表达式2)

如果条件满足，则使用表达式1,否则使用表达式2

示例：将学生成绩展示为及格和不及格

```
SELECT id,stu_name,course, IF(score>=60, '及格','不及格') score FROM score;
```

## 1.2 IFNULL(字段, 表达式)

如果字段值为空，则使用表达式，否则，使用字段值

示例：将未参加考试的学生成绩展示为缺考

```
SELECT id,stu_name,course, IFNULL(score, '缺考') score FROM score;
```

## 2. CASE...WHEN 语句

### 2.1 CASE WHEN

语法

```
CASE WHEN 条件1 THEN 表达式1 [WHEN 条件2 THEN 表达式2 ...] ELSE 表达式n END
```

如果条件1满足，则使用表达式1；【如果条件2满足，则使用表达式2， ...】否则，使用表达式n。相当于Java中的多重if..else语句。

示例

```
SELECT (CASE WHEN (course = 'Java') THEN score ELSE 0 END) Java FROM score;
```

行转列：查询每位学生的各课程成绩

```
SELECT
    stu_name,
    course,
    MAX(CASE WHEN (course = 'Java') THEN score ELSE 0 END) Java,
    MAX(CASE WHEN (course = 'Html') THEN score ELSE 0 END) Html,
    MAX(CASE WHEN (course = 'Jsp') THEN score ELSE 0 END) Jsp,
    MAX(CASE WHEN (course = 'Spring') THEN score ELSE 0 END) Spring
FROM score
GROUP BY stu_name;
```

### 2.2 CASE ... WHEN语句

语法

```
CASE 表达式 WHEN 值1 THEN 表达式1 [WHEN 值2 THEN 表达式2 ...] ELSE 表达式n END
```

如果表达式的执行结果为值1，则使用表达式1；【执行结果为值2，则使用表达式2， ...】否则，使用表达式n。相当于Java中的switch语句。

示例

```
SELECT (CASE course WHEN 'Java' THEN score ELSE 0 END) Java FROM score;
```

行转列：查询每位学生的各课程成绩

```
SELECT
    stu_name,
    course,
    MAX(CASE course WHEN 'Java' THEN score ELSE 0 END) Java,
    MAX(CASE course WHEN 'Html' THEN score ELSE 0 END) Html,
    MAX(CASE course WHEN 'Jsp' THEN score ELSE 0 END) Jsp,
    MAX(CASE course WHEN 'Spring' THEN score ELSE 0 END) Spring
FROM score
GROUP BY stu_name;
```

## 练习

查询各班级性别人数，查询结果格式如图

班级	男	女	其他
计科2班	811	822	845
软工1班	825	868	809
软工2班	833	841	841
计科1班	798	862	845

```
SELECT
    class,
    SUM(CASE sex WHEN 0 THEN 1 ELSE 0 END) '男',
    SUM(CASE sex WHEN 1 THEN 1 ELSE 0 END) '女',
    SUM(CASE sex WHEN 2 THEN 1 ELSE 0 END) '其他'
FROM stu
GROUP BY class
```

## 第五节 其他函数

### 1. 数字格式化函数

FORMAT(X,D)，将数字X格式化，将X保留到小数点后D位，截断时要进行四舍五入。

#### 示例

```
SELECT FORMAT(1.2353,2);
```

### 2. 系统信息函数

函数	说明	示例
VERSION()	获取数据库的版本号	SELECT VERSION();
CONNECTION_ID()	获取服务器的连接数	SELECT CONNECTION_ID();
DATABASE() SCHEMA()	获取当前数据库名	SELECT DATABASE(); SELECT SCHEMA();
USER() SYSTEM_USER() SESSION_USER()	获取当前用户名	SELECT USER();
CURRENT_USER() CURRENT_USER	获取当前用户名	SELECT CURRENT_USER;
VERSION()	获取数据库的版本号	SELECT VERSION();
CONNECTION_ID()	获取服务器的连接数	SELECT CONNECTION_ID();
DATABASE() SCHEMA()	获取当前数据库名	SELECT DATABASE(); SELECT SCHEMA();