

第九讲 函数

教学内容

- 9.1 系统内置函数
 - 9.1.1 系统函数
 - 9.1.2 字符串函数
 - 9.1.3 日期函数
 - 9.1.4 数学函数
 - 9.1.5 **CASE**函数
 - 9.1.6 系统内置函数应用
- 9.2 自定义函数
 - 9.2.1 自定义函数
 - 9.2.2 创建和使用自定义函数
 - 9.2.3 自定义函数示例

9.1.1 系统函数

- 系统函数用于获取有关计算机系统、用户、数据库和数据库对象的信息

函数名称	功能	函数名称	功能
VERSION()	返回数据库的版本号	SESSION_USER()	返回当前用户
DATABASE()	返回当前数据库名	USER()/SYSTEM_USER()	返回当前用户

- 示例
 - **Select DATABASE() , SESSION_USER()**

9.1.2 字符串函数

- 使用字符串函数对字符串输入值执行操作，以获得所需的字符串或数字值。
- 字符串函数分类
 - 字符转换函数
 - 去空格函数
 - 取子串函数
 - 字符串比较函数

字符转换函数

- **ASCII**

- 功能：返回字符表达式最左端字符的**ASCII**码值。
- 语法： **ASCII** <character_expression>
- 返回类型： **INT**
- 示例： **select ascii(123),ascii('A');**

- **CHAR**

- 功能：用于将 **ASCII** 码转换为字符。
- 语法： **CHAR** <integer_expression>
- 返回类型： **CHAR**
- 示例： **select char(65),Char(123);**

字符串连接

- **CONCAT(s1,s2...sn)**
 - 字符串 **s1,s2** 等多个字符串合并为一个字符串
- **CONCAT_WS(x, s1,s2...sn)**
 - 以**x**为分隔符将 **s1,s2** 等多个字符串合并为一个字符串
- 示例
 - **select concat('I ','love ','MySQL.');**
 - **select concat_ws('-', 'I ','love ','MySQL.');**

字符串长度

- **CHAR_LENGTH(s) / CHARACTER_LENGTH(s)**
 - 返回字符串 **s** 的字符数
- 示例
 - **select CHAR_LENGTH('西林SWFU');**
 - **select CHARACTER_LENGTH('西林SWFU');**

字符位置

- **FIELD(s,s1,s2...)**
 - 返回第一个字符串 **s** 在字符串列表(**s1,s2...**)中的位置
- **FIND_IN_SET(s1,s2)**
 - 返回在字符串**s2**中与**s1**匹配的字符串的位置
- 示例
 - **select FIELD('S','西','林','S','WF','U');**
 - **select FIND_IN_SET('S','西,林,S,WF,U');**
 - **select FIND_IN_SET('S','西林SWFU');**

字符转换函数

- **LOWER/LCASE**

- 功能：把字符串全部转换为小写。
- 语法： **LOWER <character _expression>**
- 返回类型： **VARCHAR**
- 示例： **select LOWER ('Abc'), LOWER ('A李C');**

- **UPPER/UCASE**

- 功能：把字符串全部转换为大写。
- 语法： **UPPER<character _expression>**
- 返回类型： **VARCHAR**
- 示例： **select upper('Abc'),upper('a李c');**

去空格函数

- **LTRIM**

- 功能：把字符串头部的空格去掉。
- 语法： **LTRIM** <character_expression>
- 返回类型： **VARCHAR**
- 示例： **Select Ltrim(' A'),rtrim(' 李 ');**

- **RTRIM**

- 功能：把字符串尾部的空格去掉。
- 语法： **RTRIM** <character_expression>
- 返回类型： **VARCHAR**
- 示例： **Select Ltrim('A '),rtrim(' 李 ');**



显示A~Z字母

- delimiter \$\$
- drop procedure if exists while_loop;
- create procedure while_loop()
- begin
- declare i int default 65;
- declare R varchar(100);
- while i<65+26 do
- if R is null then
- set R=CHAR(i);
- else
- set R=CONCAT_WS(',', R,char(i));
- end if;
- set i=i+1;
- end while;
- select R;
- end \$\$
- delimiter;
- call while_loop;

取子串函数——LEFT

- 功能
 - 返回从字符串左边开始指定个数的字符
- 语法
 - **LEFT(character_expression,integer_expression)**
- 返回类型
 - **VARCHAR**
- 示例
 - **Select LEFT('ABC',2),LEFT('A西林C',2);**

RIGHT

- 功能
 - 返回从字符串右边开始指定个数的字符
- 语法
 - **RIGHT(character_expression,integer_expression)**
- 返回类型
 - **VARCHAR**
- 示例
 - **Select RIGHT('ABC',2),RIGHT('A西林C',2);**

SUBSTRING / SUBSTR / MID

- 功能
 - 返回字符串、**binary**、**text**表达式的一部分
- 语法
 - **SUBSTRING (expression,start,length)**
- 返回类型
 - **VARCHAR, NVARCHAR, VARBINARY**
- 示例:
 - **Select SubString('ABC',2,1),
SubString('ABC',2,4),
SubString('A西F',1,2);**

SUBSTRING_INDEX

- **SUBSTRING_INDEX(s, delimiter, number)**
 - 返回从字符串 **s** 的第 **number** 个出现的分隔符 **delimiter** 之后的子串。
 - **number** 正数, 返回第 **number** 个字符左边的字符串。
 - **number** 负数, 返回第(**number** 的绝对值(从右边数))个字符右边的字符串。
- 示例
 - **SELECT SUBSTRING_INDEX('a*b','*',1) -- a**
 - **SELECT SUBSTRING_INDEX('a*b','*','-1) -- b**
 - **SELECT SUBSTRING_INDEX(SUBSTRING_INDEX('a*b*c*d*e','*',3),'*','-1) -**
- c

LOCATE

- 功能
 - 返回字符串中某个指定子串出现的起始位置
- 语法
 - **LOCATE(substring_expression,
expression[,start_location])**
- 返回类型: **INT**
- 示例:
 - **Select LOCATE ('B','ABC'),
LOCATE ('AD','ABCD'),
LOCATE ('B','ABCBDE',3);**

POSITION

- **POSITION(s1 IN s)**
 - 从字符串 **s** 中获取 **s1** 的开始位置
- 示例
 - **SELECT POSITION('b' in 'abc')**

REPEAT & REVERSE

- **REPEAT(s,n)**
 - 将字符串 **s** 重复 **n** 次
 - 示例: **SELECT REPEAT('runoob',3);**
- **REVERSE(s)**
 - 将字符串**s**的顺序反过来
 - **SELECT REVERSE('abc')**
- **SPACE(s)**
 - 返回 **n** 个空格
 - **Select space(10);**

REPLACE

- 功能
 - 用第三个表达式替换第一个字符串表达式中出现的所有第二个给定字符串表达式。
- 语法
 - REPLACE('string_expression1',
 'string_expression2',
 'string_expression3')
- 返回类型：与表达式类型一致
- 示例
 - Select Replace('ABC','B','123'),
 Replace('ABCD','BD','ERR');

9.1.3 日期函数

- 显示日期和时间的信息
- 数据类型为**datetime**和**smalldatetime**值
- 可以执行算术运算，返回一个字符串、数字值或日期和时间值
- 可以在**SELECT**语句的**SELECT**和**WHERE**子句以及表达式中使用日期函数。

日期函数

- **CURDATE() / CURRENT_DATE()**
 - 返回当前系统日期
- **CURRENT_TIMESTAMP() / NOW()**
 - 返回当前日期和时间
- **CURTIME()**
 - 返回当前时间
- 示例
 - **select CURDATE()**

日期函数

- **DAY(d)**
 - 返回日期值 **d** 的日期部分
 - 示例: **Select Day('2019-05-12'), Day('05/12/2019');**
- **MONTH(d)**
 - 返回日期**d**中的月份值, 1 到 12
 - 示例: **Select Month('2019-05-12'),Month('05/12/2019');**
- **YEAR(d)**
 - 返回年份
 - 示例: **Select YEAR('2019-05-12'),YEAR('05/12/2019')**

日期函数

- **DATE_ADD(d, INTERVAL expr type)**
 - 计算起始日期 d 加上一个时间段后的日期
 - 示例: **SELECT ADDDATE('2021-04-11 11:11:11',1)**
 - **SELECT ADDDATE('2021-04-11 11:11:11', INTERVAL 5 MINUTE)**
- **DATEDIFF(d1,d2)**
 - 计算日期 d1->d2 之间相隔的天数
 - 示例: **SELECT DATEDIFF('2021-01-01','2021-02-02')**
- **DATE_SUB(date,INTERVAL expr type)**
 - 函数从日期减去指定的时间间隔。
 - **SELECT DATE_SUB('2021-01-01',INTERVAL 2 DAY)**

9.1.4 数学函数

- 用于执行各种算术运算或函数运算
- **POWER / POW**
 - 功能：返回给定表达式乘指定次方的值
 - 语法：**POWER(numeric_expression,y)**
 - 返回类型：与numeric_expression相同
 - 示例：**Select POWER(2,3),POWER(2,0)**
- **ROUND**
 - 功能：返回数字表达式并四舍五入为指定的长度或精度
 - 语法：**ROUND(numeric_expression,length[,function])**
 - 返回类型：与numeric_expression相同。
 - 示例：**Select ROUND(24.567,2),ROUND(24.25,0)**

数学函数

- **MOD(x,y)**
 - 返回 **x** 除以 **y** 以后的余数
 - **SELECT MOD(5,2)**
- **RAND()**
 - 返回 **0** 到 **1** 的随机数
 - **SELECT RAND()**
- **LEAST(expr1, expr2, expr3, ...)**
 - 返回列表中的最小值
 - **SELECT LEAST(3, 12, 34, 8, 25);**
- **GREATEST(expr1, expr2, expr3, ...)**
 - 返回列表中的最大值
 - **SELECT GREATEST(3, 12, 34, 8, 25);**

9.1.5 CASE函数——简单CASE函数

- 功能
 - 将某个表达式与一组简单表达式比较以确定结果
- 语法
 - **CASE input_expression**
 - **WHEN when_expression THEN result_expression**
 - **[...n]**
 - **[ELSE else_result_expression]**
 - **END;**

查询学生信息，将性别以英文显示

- **Select StudNo,StudName,
Case StudGender
When '男' then 'Male'
When '女' then 'Female'
Else '性别不详'
End AS 学生性别**
- **From StudInfo;**

CASE搜索函数

- 功能
 - **CASE**搜索函数计算一组布尔表达式以确定结果
- 语法
 - **CASE**
 - **WHEN Boolean_expression THEN result_expression**
 - **[...n]**
 - **[ELSE else_result_expression]**
 - **END;**

统计各学生平均分，并按等级显示

- **Select StudNo,Avg(StudScore) AvgScore,
Case When Avg(StudScore)>=90 then '优秀'
When Avg(StudScore)>=80 And
Avg(StudScore)<90 then '良好'
When Avg(StudScore)>=70 then '中等'
When Avg(StudScore)>=60 then '及格'
Else '不及格'
End As ScoreLevel**
- **From StudScoreInfo**
- **Group by StudNo;**

9.1.6 系统内置函数应用

- 查询姓“李”的学生信息
 - **Select * From StudInfo**
 - **Where Left(StudName,1)='李'**
- 查询姓名中包含“文”的学生信息
 - **Select * From StudInfo**
 - **Where LOCATE('文',StudName)>0;**
- 查询姓名中第二个字为“文”的学生信息
 - **Select * From StudInfo**
 - **Where substring(StudName,2,1)='文';**

系统内置函数应用

- 将姓名中的“云”字改为“芸”。
 - Update StudInfo
 - Set **StudName=Replace(StudName,'云','芸');**
- 统计同姓学生人数为3人以上的信息
 - Select Left(StudName,1),Count(*)
 - From StudInfo
 - **Group By Left(StudName,1)**
 - Having Count(*)>=3;

统计某学号各分数段课程门数

- **Select Case When StudScore>=90 then '优秀'**
When StudScore>=80 then '良好'
When StudScore>=70 then '中等'
When StudScore>=60 then '及格'
Else '不及格'
End As ScoreLevel,
Count(*) CourseCount
- **From StudScoreInfo**
- **Where StudNo='20180712001'**
- **Group By Case When StudScore>=90 then '优秀'**
When StudScore>=80 then '良好'
When StudScore>=70 then '中等'
When StudScore>=60 then '及格'
Else '不及格'
End;

9.2 自定义函数

- 功能
 - 用户自定义函数可以像系统函数一样在查询或存储过程等的程序段中使用
- 语法
 - **delimiter** 自定义符号
 - **create function** 函数名(形参列表) **returns** 返回类型
 - **begin**
 - 函数体;
 - 返回值;
 - **end;**
 - **delimiter ;**

创建一个计算阶乘的函数

- `set GLOBAL log_bin_trust_function_creators=1`
- `delimiter $$`
- `drop function if exists get_jc;`
- `create function get_jc(n int) returns int`
- `begin`
- `declare i int default 1;`
- `declare k bigint default 1;`
- `while i <=n DO`
- `set k=k*i;`
- `set i=i+1;`
- `end while;`
- `return k;`
- `end $$`
- `delimiter ;`
- `select get_jc(5);`

指定数据之间的和

- **delimiter \$\$**
- **create function my_sum(x int) returns int**
- **begin**
- **set @i = 1;**
- **set @sum = 0;**
- **while @i <= x do**
- **set @sum = @sum + @i;**
- **set @i = @i + 1;**
- **end while;**
- **return @sum;**
- **End \$\$**
- **delimiter ;**

管理自定义函数

- 查看自定义函数
 - **show function status [like 'pattern'];**
 - **show create function 函数名;-- 查看函数创建语句**
- 删除自定义函数
 - 语法
 - **DROP FUNCTION 自定义函数名**
 - 示例
 - **DROP FUNCTION Get_JC;**

自定义函数示例

- **Select Length('ABC') → 3**
- **Select LOCATE('B','ABC') → 2**
- **Select LOCATE('BD','ABCD') → 0**
- **Select LOCATE('B','ABCDBC',3) → 5**
- **Select SubString('ABC',2,1) → B**
- **Select SubString('ABC',2,3) → BC**
- **Select LOCATE(SubString('ABC',2,1),'BCD') → 1**

学生答案信息表 (StudAnswer)

Stand_Ans	Custor_Ans	StudNo
B	B	20180404002
ACD	ACD	20180404002
A	C	20180404002
C	C	20180404002
AD	D	20180404002
B	B	20190316003
ABC	ABCD	20190316003
AB	A	20190316003
ACD	AD	20190316003

学生成绩信息表 (StudScore)

StudNo	StudName	StudScore
20180404002	李泽伟	
20190316003	陈莉	
20190505002	沈映璧	
20190505003	段练	
20190505004	张璐	
20190505005	胡靖萱	
20190505006	包云鹏	
20190505007	王平	
20190505008	吴边	
20190505009	史波梅	

评分标准：错选一个全错

学生答案	标准答案	标准分	得分
A	A	1	1
BCD	BC	2	0
B	BD	2	1
ABC	ABD	3	0
BD	ABCD	4	2
	标准总分	12	4
	百分制： $4 \times 100 / 12 = 33$		

学生统分函数

- **CREATE Function GetItemScore(Stand_Ans varchar(10),Custor_Ans varchar(10))**
- **returns int**
- **begin**
- **declare i int default 1;**
- **if CHAR_LENGTH(Custor_Ans)>CHAR_LENGTH(Stand_Ans) or Custor_Ans IS NULL then**
- **return 0;**
- **end if;**
- **while i<=CHAR_LENGTH(Custor_Ans) do**
- **if locate(SubString(Custor_Ans,i,1),Stand_Ans)<1 then**
- **return 0;**
- **end if;**
- **Set i=i+1;**
- **end while;**
- **return CHAR_LENGTH(Custor_ans);**
- **end ;**



Select GetItemScore('ABC','AC')

```
CREATE Function GetItemScore(Stand_Ans varchar(10),Custor_Ans varchar(10))
returns int
begin
  declare i int default 1;
  if CHAR_LENGTH(Custor_Ans)>CHAR_LENGTH(Stand_Ans) or Custor_Ans IS NULL then
    return 0;
  end if;
  while i<=CHAR_LENGTH(Custor_Ans) do
    if locate(SubString(Custor_Ans,i,1),Stand_Ans)<1 then
      return 0;
    end if;
    Set i=i+1;
  end while;
  return CHAR_LENGTH(Custor_ans);
end ;
```

Diagram illustrating the execution of the `GetItemScore` function with inputs `'ABC'` and `'AC'`:

- `Stand_Ans` is `'ABC'` (length 3).
- `Custor_Ans` is `'AC'` (length 2).
- The function checks if `CHAR_LENGTH(Custor_Ans) > CHAR_LENGTH(Stand_Ans)` (2 > 3) or `Custor_Ans IS NULL`. This condition is false.
- The function enters a `while` loop where `i` ranges from 1 to `CHAR_LENGTH(Custor_Ans)` (2).
- For `i=1`, it checks `locate(SubString(Custor_Ans,1,1),Stand_Ans)`. `SubString(Custor_Ans,1,1)` is `'A'`, and `locate('A','ABC')` returns 1, which is not less than 1. The condition is false.
- For `i=2`, it checks `locate(SubString(Custor_Ans,2,1),Stand_Ans)`. `SubString(Custor_Ans,2,1)` is `'C'`, and `locate('C','ABC')` returns 3, which is not less than 1. The condition is false.
- The loop ends, and the function returns `CHAR_LENGTH(Custor_ans)`, which is 2.

Select GetItemScore('ABC','AC')

CREATE Function GetItemScore(Stand_Ans varchar(10),Custor_Ans varchar(10))

returns int

begin

declare i int default 1;

if CHAR_LENGTH(Custor_Ans)>CHAR_LENGTH(Stand_Ans) or Custor_Ans IS NULL then

return 0;

end if;

while i<=CHAR_LENGTH(Custor_Ans) do

if locate(SubString(Custor_Ans,i,1),Stand_Ans)<1 then

return 0;

end if;

Set i=i+1;

end while;

return CHAR_LENGTH(Custor_ans);

end ;

↑
ABC

↑
AC

↓
AC

↓
ABC

2

3

2

2

2

↓
C

↓
ABC

3

3

自定义函数调用

- **Select GetItemScore('AB','ACD')**
- **Select GetItemScore('A','B')**
- **Select GetItemScore('ACD','ABD')**
- **Select GetItemScore('ABC','AC')**

统计各题正确个数

- **Select StudNo,Stand_Ans,Custor_Ans,
GetItemScore(Stand_Ans,
Custor_Ans) As CorrectCount**
- **From TestAnswer;**

统计学生所有题正确数

- **Select StudNo,
Sum(GetItemScore(Stand_Ans,
Custor_Ans)) As TotalCount**
- **From TestAnswer**
- **Group by StudNo;**

统计各学生所有题目正确得分和标准分

- **Select StudNo,
Sum(GetItemScore(Stand_Ans,
Custor_Ans)) As TotalCount,
Sum(char_Length(Stand_Ans)) As
StandCount**
- **From TestAnswer**
- **Group by StudNo;**

百分制转换

- **Select StudNo,
Sum(GetItemScore(Stand_Ans,
Custor_Ans))*100.0/
Sum(char_Length(Stand_Ans)) As
StudScore**
- **From TestAnswer**
- **Group by StudNo;**

利用视图统计结果更新到表

- 创建视图
 - **Create View V_Calc_StudScore**
 - **As**
 - **Select StudNo,**
(Sum(GetItemScore(Stand_Ans,Custor_Ans))*100.0/
Sum(char_Length(Stand_Ans))) As StudScore
 - **From TestAnswer Group By StudNo;**
- 利用视图更新表数据
 - **Update StudScore, V_Calc_StudScore S**
 - **Set StudScore=S.StudScore**
 - **Where StudNo=S.StudNo;**

下次课内容

- 存储过程
 - 存储过程介绍
 - 创建存储过程
 - 使用存储过程
 - 管理存储过程
- 触发器
 - 触发器介绍
 - 创建触发器
 - 管理触发器
- 游标
 - 游标介绍
 - 创建游标
 - 使用游标