

数据类型、运算符与常用 函数

本章知识点

- 知识点1: 数据类型
- 知识点2: 运算符
- 知识点3: 运算符的优先级
- 知识点4: 字符串函数
- 知识点5: 数值函数
- 知识点6: 日期和时间函数
- 知识点7: 流程函数
- 知识点8: JSON函数
- 知识点9: 窗口函数
- 知识点10: 其它常用函数

知识点1 【数据类型】

- MySQL提供了多种数据类型，主要包括：
 - 数值型
 - 字符串类型
 - 日期和时间类型

知识点1 【数据类型】 -数值类型

- MySQL中的数值类型:

整型类型	字节	最小值	最大值
TINYINT	1	有符号 -128 无符号 0	有符号 127 无符号 255
SMALLINT	2	有符号 -32768 无符号 0	有符号 32767 无符号 65535
MEDIUMINT	3	有符号 -8388608 无符号 0	有符号 8388607 无符号 16777215
INT、INTEGER	4	有符号 -2147483648 无符号 0	有符号 2147483647 无符号 4294967295
BIGINT	8	有符号 -9223372036854775808 无符号 0	有符号 9223372036854775807 无符号 18446744073709551615
浮点数类型	字节	最小值	最大值
FLOAT	4	$\pm 1.175494351E-38$	$\pm 3.402823466E+38$
DOUBLE	8	$\pm 2.2250738585072014E-308$	$\pm 1.7976931348623157E+308$
定点数类型	字节	描述	
DEC(M,D), DECIMAL(M,D)	M+2	最大取值范围与DOUBLE相同, 给定DECIMAL的有效取值范围由M和D决定	
位类型	字节	最小值	最大值
BIT(M)	1~8	BIT(1)	BIT(64)

知识点1 【数据类型】 - 日期时间类型

- MySQL中的日期和时间类型：

日期和时间类型	字节	最小值	最大值	零值表示
DATE	4	1000-01-01	9999-12-31	0000-00-00 00:00:00
DATETIME	8	1000-01-01 00:00:00	9999-12-31 23:59:59	0000-00-00
TIMESTAMP	4	19700101080001	2038年的某个时刻	0000000000000000
TIME	3	-838:59:59	838:59:59	00:00:00
YEAR	1	1901	2155	0000

- 如果要用来表示年月日，通常用DATE来表示；
- 如果要用来表示年月日时分秒，通常用DATETIME或者TIMESTAMP表示；
- 如果只用来表示时分秒，通常用TIME来表示；
- 如果只是表示年份，可以用YEAR来表示，它比DATE占用更少的空间。YEAR有2位或4位格式的年。默认是4位格式中，允许的值是1901~2155和0000。在2位格式中，允许的值是70~69，表示从1970~2069年。MySQL以YYYY格式显示YEAR值（从5.5.27开始，2位格式的year已经不被支持）。

注意：每种日期时间类型都有一个有效值范围，如果超出这个范围，在默认的SQLMode下，系统会进行错误提示，并将以零值来进行存储。

知识点1 【数据类型】-字符串类型1

- MySQL提供了多种对字符数据的存储类型，不同的版本可能有所差异，以5.7版本为例，MySQL包括了CHAR、VARCHAR、BINARY、VARBINARY、BLOB、TEXT、ENUM和SET等多种字符串类型，如下表所示：

日期和时间类型	字节	描述及存储需求
CHAR(M)	M	M为0~255之间的整数
VARCHAR(M)		M为0~65535之间的整数，值的长度+1个字节
TINYBLOB		允许长度0~255字节，值的长度+1个字节
BLOB		允许长度0~65535字节，值的长度+2个字节
MEDIUMBLOB		允许长度0~167772150字节，值的长度+3个字节
OLONGBLOB		允许长度0~4294967295字节，值的长度+4个字节
TINYTEXT		允许长度0~255字节，值的长度+2个字节
TEXT		允许长度0~65535字节，值的长度+2个字节
MEDIUMTEXT		允许长度0~167772150字节，值的长度+3个字节
LONGTEXT		允许长度0~4294967295字节，值的长度+4个字节
VARBINARY(M)		允许长度0~M个字节的变长字节字符串，值的长度+1个字节
BINARY(M)	M	允许长度0~M个字节的定长字节字符串

知识点1 【数据类型】-字符串类型2

- CHAR和VARCHAR类型：
 - 两者的主要区别在于存储方式不同：
 - CHAR列的长度固定为创建表时声明的长度，而VARCHAR列中的值为可变长字符串；
 - 在检查的时候，CHAR列删除尾部的空格，而VARCHAR则保留这些空格
 - 课堂案例：
 - 创建测试表vc，并定义两个字段“v VARCHAR(4)”和“c CHAR(4)”；
 - v列和c列中同时插入字符串“ab ”；
 - 显示查询结果并分析上述区别；

案例

- 完成以下案例，选择使用什么数据类型？

- (1) 年龄
- (2) 价格
- (3) 手机号
- (4) 身份证号
- (5) 出生日期
- (6) 订单时间
- (7) 籍贯
- (8) 性别
- (9) 图片:通常存储图片的路径
如果存储图片的二进制
- (10) 文章
- (11) 状态/角色

知识点1 【数据类型】 -JSON类型

- JSON是JavaScript Object Notation的缩写，它是一种数据交换格式。
- JSON类型比字符类型有如下优点：
 - JSON数据类型会自动校验数据是否为JSON格式，如果不是JSON格式数据，则会报错；
 - MySQL提供了一组操作JSON数据的内置函数，可以方便地提取各类数据，可以修改特定的键值；
 - 优化的存储格式，存储在JSON列中的JSON数据被转换成内部的存储格式，允许快速读取；
 - 简单地说，JSON实际就是Javascript的一个子集，支持的数据类型包括NUMBER、STRING、BOOLEAN、NULL、ARRAY、OBJECT共6种，一个JSON中的元素可以是这6种类型元素的任意组合；
 - BOOLEAN: true/false
 - NULL: null
 - 字符串和日期类型：用双引号引起来
 - ARRAY: 括号引起来
 - OBJECT: KV，要用大括号引起来
- 注意：JSON，数据类型对于大小写是敏感的，常见的null、true、false必须是小写才合适。

知识点2 【运算符】-算术运算符

- MySQL支持的算术运算符：

- +：加法
- -：减法
- *：乘法
- /、DIV：除法，返回商
- %、MOD：除法，返回余数

- 描述：

- +：运算符用于获得一个或多个值的和；
- -：运算符用于从一个值中减去另一个值；
- *：运算符使数字相乘，得到两个或多个值的乘积；
- /：运算符用一个值除以另一个值得到商；
- %：运算符用一个值除以另外一个值得到余数；
- 注意：在除法运算和模运算中，如果除数为0，将是非法除数，返回结果为NULL。

知识点2 【运算符】-比较运算符

- 比较运算符:

- "="运算符: 用于比较运算符两侧的操作数是否相等, 如果两侧操作数相等, 则返回值为1, 否则为0。注意NULL不能用于"="比较。
- "<>"运算符: 和"="相反, 如果两侧操作数不等, 则值为1, 否则为0。NULL不能用于"<>"比较。
- "<=>"运算符: 和"="类似, 在操作数相等时值为1, 不同之处在于即使操作的值为NULL也可以正确比较。
- "<"运算符: 当左侧操作数小于右侧操作数时, 其返回值为1, 否则其值为0。
- "<="运算符: 当左侧操作数小于等于右侧操作数时, 其返回值为1, 否则返回值为0。
- ">"运算符: 当左侧操作数大于右侧操作数时, 其返回值为1, 否则返回值为0。
- ">="运算符: 当左侧操作数大于等于右侧操作数时, 其返回值为1, 否则返回值为0。
- "BETWEEN"运算符的使用格式为"a BETWEEN min AND max", 当a大于等于min并且小于等于max, 则返回值为1, 否则返回0。
- "IN"运算符使用的格式为"a IN(value1,value2...)", 当a的值存在于列表中时, 则整个比较表达式返回的值为1, 否则返回0。
- "IS NULL"运算符的使用格式为"a IS NULL", 当a的值为NULL, 则返回值为1, 否则返回值为0。
- "IS NOT NULL"运算符的使用格式为"a IS NOT NULL", 当a的值为不为NULL, 则返回值为1, 否则返回值为0。
- "LIKE"运算符的使用格式为"a LIKE %123%", 当a中含有字符串"123"时, 则返回值为1, 否则返回0。
- "REGEXP"运算符的使用格式为"str REGEXP str_pat", 当字符串中含有str_pat相匹配的字符串, 则返回值为1, 否则返回0。

知识点2 【运算符】-逻辑运算符

- MySQL支持的逻辑运算符：
 - “NOT” 或 “!” 表示逻辑非。
 - “AND” 或 “&&” 表示逻辑与运算。
 - “OR” 或 “||” 表示逻辑或运算。
 - “XOR” 表示逻辑异或。

知识点2 【运算符】-位运算符

- MySQL支持的位运算符：
 - "&" 位与：对多个操作数的二进制位做逻辑与操作。
 - "|" 位或：对多个操作数的二进制位做逻辑或操作。
 - "^" 位异或：对操作数的二进制位做异或操作。
 - "~" 位取反：对操作数的二进制位做NOT操作。
 - ">>" 位右移：对左操作数向右移动右操作数指定的位数。
 - "<<" 位左移：对左操作数向左移动右操作数指定的位数。

知识点3 【运算符的优先级】

- 优先级由低到高排列，同一行中的运算符具有相同的优先级：

优先级顺序	描述及存储需求
1	:=
2	、OR、XOR
3	&&、AND
4	NOT
5	BETWEEN、CASE、WHEN、THEN、ELSE
6	=、<=>、>=、>、<=、<、<>、!=、IS、LIKE、REGEXP、IN
7	
8	&
9	<<、>>
10	-、+
11	*/、DIV、%、MOD
12	^
13	-（一元减号）、~（一元比特反转）
14	!

知识点4 【字符串函数】 -1

- MySQL支持的字符串函数：

函数	功能
concat(s1,s2,.....sn)	连接s1,s2,.....,sn为一个字符串
CONCAT_ws(x,s1,s2)	将字符串s1,s2使用指定分隔符x进行连接
lower(str)	将字符串str中字符变为小写
upper(str)	将字符串str中字符变为大写
left(str,x)	返回字符串str最左边的x个字符
right(str,x)	返回字符串str最右边的x个字符
LENGTH()	获取字符串长度
rpad(str,n,pad)	用字符串pad对str最右边进行填充，直到长度为n个字符长度
ltrim(str)	去掉字符串str左侧的空格
rtrim(str)	去掉字符串str右侧的空格
repeat(str,x)	返回str重复x次的结果
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
strcmp(s1,s2)	比较字符串s1和s2
trim(str)	去掉字符串行尾和行头的空格
substring(str,x,y)	返回从字符串str x位置起y个字符长度的字串

知识点4 【字符串函数】 -2

- 完成以下案例：
 - 自动生成昵称,empname加上手机号后四位。
 - 自动生成昵称,empname_手机号后四位
 - 显示工资，并将工资格式化。数字格式化 FORMAT()
 - 查询员工名以及获取员工名的字符串长度
 - 查询每个员工手机的后四位数
 - 查询显示每个员工的手机号为132*****789
 - 查询显示员工名为L.Lei

知识点5 【数值函数】 -1

- MySQL支持的数值函数：

函数	功能
abs(x)	返回x的绝对值
ceil(x)	返回大于x的最小整数值
floor(x)	返回小于x的最大整数值
mod(x,y)	返回x/y的模
rand()	返回0~1内的随机值
round(x,y)	返回参数x的四舍五入的有y位小数的值
truncate(x,y)	返回数字x截断为y位小数的结果

知识点6 【日期和时间函数】 -1

- MySQL支持的日期和时间函数：

函数	功能
curdate()	返回当前日期
curtime()	返回当前时间
now()	返回当前的日期和时间
unix_timestamp(date)	返回日期date的unix时间戳
from_unixtime	返回unix时间戳的日期值
week(date)	返回日期date为一年中的第几周
year(date)	返回日期date的年份
hour(time)	返回time的小时值
minute(time)	返回time的分钟值
monthname(date)	返回日期date的月份名
date_format(date,fmt)	返回按字符串fmt格式化日期date值
date_add(date,interval expr type)	返回一个日期或时间值加上一个时间间隔的时间值
datediff(expr,expr2)	返回起始时间expr和结束时间expr2之间的天数

知识点6 【日期和时间函数】 -2

- MySQL支持的日期和时间格式:

格式符	格式说明
%S和%s	两位数字形式的秒（00，01，...，59）
%i	两位数字形式的分（00，01，...，59）
%H	两位数字形式的小时，24小时（00，01，...，23）
%h和%l	两位数字形式的小时，12小时（00，01，...，11）
%k	数字形式的小时，24小时（0，1，...，23）
%l	数字形式的小时，12小时（0，1，...，12）
%T	24小时的时间形式（hh:mm:ss）
%r	12小时的时间形式（hh:mm:ssAM或hh:mm:ssPM）
%p	AM或PM
%W	一周中每一天的名称（Sunday，Monday，...，Saturday）
%a	一周中每一天名称的缩写（Sun，Mon，...，Sat）

知识点6 【日期和时间函数】 -3

- MySQL支持的日期和时间格式:

格式符	格式说明
%d	两位数字表示月中的天数（00， 01， ...， 31）
%e	数字形式表示月中的天数（00， 01， ...， 31）
%D	英文后缀表示月中的天数（1st， 2nd， 3rd， ...）
%w	以数字形式表示周中的天数（0=Sunday， 1=Monday， ...， 6=Saturday）
%j	以3位数字表示年中的天数（001， 002， ...， 366）
%U	周（0， 1， 52）， 其中Sunday为周中的第一天
%u	周（0， 1， 52）， 其中Monday为周中的第一天
%M	月名（January， February， ...， December）
%b	缩写的月名
%m	两位数字表示的月份（01， 02， ...， 12）
%c	数字表示的月份（1， 2， ...， 12）
%Y	4位数字表示的年份
%y	两位数字表示的年份
%%	直接值“%”

知识点6 【日期和时间函数】 -3



- MySQL支持的日期间隔类型:

表达式类型	描述	格式
hour	小时	hh
minute	分	mm
second	秒	ss
year	年	YY
month	月	MM
day	日	DD
year_month	年和月	YY-MM
day_hour	日和小时	DD hh
day_minute	日和分钟	DD hh:mm
day_second	日和秒	DD hh:mm:ss
hour_minute	小时和分	hh:mm
hour_second	小时和秒	hh:ss
minute_second	分钟和秒	mm:ss

知识点7 【流程函数】 -1

- MySQL支持流程函数：

函数	功能
if(value,t,f)	如果value是真，返回t；否则返回f
ifnull(value1,value2)	如果value1不为空，返回value1,否则value2
case when[value1] then [result]...else [default] end	如果value1是真，返回result，否则返回default
case [expr] when [value1] then [result1] ... else [default] end	如果expr等于value1，返回result1，否则返回default

案例1:查询员工信息，性别如果是1就显示男，否则就显示女

案例2： 查询员工信息，如果手机号为null就显示没添加号码，不为null就正常显

示

知识点8 【JSON函数】 -1

- 创建JSON函数：

函数	功能
json_array()	创建json数组
json_object()	创建json对象
json_quote()/json_unquote()	加上/去掉json文档两边的双引号

- json_array([val[,val]...])
- json_object[key,val[,key,val]...])
- json_quote(string)

知识点8 【JSON函数】 -2



- 查询JSON函数：

函数	功能
json_contains()	查询文档中是否包含指定的元素
json_contains_path()	查询文档中是否包含指定的路径
json_extract()	根据条件提取文档中的数据
json_keys()	提取所有key的集合
json_search()	返回所有符合条件的路径集合

- json_contains(target,candidate[,path])
- json_contains_path(json_doc,one_or_all,path[,path]...)
- json_extract(json_doc,path[,path]...)
- json_keys(json_doc[,path])
- json_search(json_doc,one_or_all,search_str[,escape_char[,path]...])

知识点8 【JSON函数】 -3



- 修改JSON函数:

函数	功能
json_merge()(deprecated 5.7.22) json_merge_preserve	将两个文档合并
json_array_append()	数组尾部追加元素
json_array_insert()	在数组的指定位置插入元素
json_remove()	删除文档中指定位置的元素
json_replace()	替换文档中指定位置的元素
json_set()	给文档中指定位置的元素设置新值，如果元素不存在，则进行插入

- json_array_append(json_doc,path,val[,path,val]...)
- json_array_insert(json_doc,path,val[,path,val]...)
- json_replace(json_doc,path,val[,path,val]...)
- json_set(json_doc,path,val[,path,val]...)
- json_merge_preserve(json_doc,json_doc[,json_doc]...)
- json_remove(json_doc,path[,path]...)

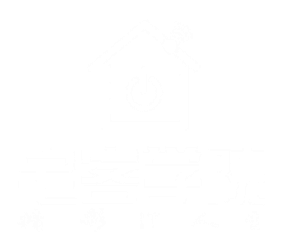
知识点8 【JSON函数】 -4

- 查询JSON元数据的函数：

函数	功能
json_depth()	JSON文档的深度（元素最大嵌套层数）
json_length()	JSON文档的长度（元素个数）
json_type()	JSON文档类型（数组、对象、标量类型）
json_valid()	JSON格式是否合法

- json_depth(json_doc)
- json_length(json_doc[,path])
- json_type(json_val)
- json_valid(val)

知识点8 【JSON函数】 -5



- JSON工具函数：

函数	功能
json_pretty()	美化JSON格式
json_storage_size()	JSON文档占用的存储空间
json_storage_free()	JSON文档更新操作后剩余的空间，MySQL8.0新增
json_table()	将JSON文档转换为表格，MySQL8.0新增
json_arrayagg()	将聚合后参数中的多个值转换为JSON数组
json_objectagg()	将两个列或者是表达式解释为一个key和一个value，返回一个JSON对象

- json_pretty(json_val)
- json_storage_size(json_val) / json_storage_free(json_val)
- json_table(expr,path columns(column_list) [as] alias)
- json_arrayagg(col_or_expr)
- json_objectagg(key,value)

知识点9 【窗口函数】 -1

- ROW_NUMBER():

- row_number()后面的over是关键字，用来指定函数执行的窗口范围，如果后面什么都不写，则意味着窗口包含所有行，窗口：函数在所有行上进行计算；如果不为空，则支持以下4种语法：
 - window_name：给窗口指定一个别名，如果SQL中涉及的窗口较多，采用别名则更清晰易读；
 - partition子句：窗口按照哪些字段进行分组，窗口函数在不同的分组上分别执行；
 - order by子句：按照哪些字段进行排序，窗口函数将按照排序后的记录顺序进行编号，既可以和partition子句配合使用，也可以单独使用；
 - frame子句：frame是当前分区的一个子集，子句用来定义子集的规则，通常用来作为滑动窗口使用。
 - 基于行
 - 基于范围

知识点9 【窗口函数】 -2

- RANK()/DENSE_RANK():
 - rank()/dense_rank()这两个函数与row_number()非常类似，只是在出现重复值时处理逻辑有所不同。

知识点9 【窗口函数】 -3

- PERCENT_RANK() / CUME_DIST() :
 - percent_rank()和cume_dist()这两个函数都是计算数据分布的函数，percent_rank()和之前的rank()函数相关，每行按照以下公式进行计算：
 - $(\text{rank}-1)/(\text{rows}-1)$ ，其中，rank为rank()函数产生的序号，rows为当前窗口的记录总行数。
 - 相比percent_rank()，cume_dist()函数的应用场景更多，它的作用是分组内小于等于rank值的行数/分组内的总行数（统计大于等于当前订单金额的订单数，占总订单数的比例）。

知识点9 【窗口函数】 -4

- NTILE(N) :
 - ntile()函数的功能是对一个数据分区中的有序结果集进行划分，将其分为N个组，并为每个小组分配一个唯一的组编号。
 - 此函数在数据分析中应用较多，比如由于数据最大，需要将数据分配到N个并行的进程分别计算，此时就可以用NTILE(N)对数据进行分组，由于记录数不一定被N整除，所以每组记录数不一定完全一致，然后将不同组号的数据再分配。

知识点9 【窗口函数】 -5

- NTH_VALUE(expr, N) :
 - NTH_VALUE(expr, N)函数可以返回窗口中第N个expr的值，expr既可以是表达式，也可以是列名。

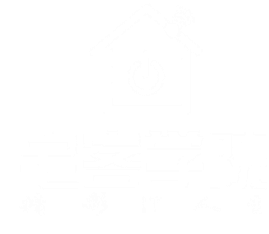
知识点9 【窗口函数】 -6

- $\text{LAG}(\text{expr}, N) / \text{LEAD}(\text{expr}, N)$:
 - $\text{LAG}(\text{expr}, N)$ 和 $\text{LEAD}(\text{expr}, N)$ 这两个函数的功能是获取当前数据行按照某种排序规则上的N行（LAG） / 下N行（LEAD） 数据的某个字段。

知识点9 【窗口函数】 -7

- FIRST_VALUE(expr)/LAST_VALUE(expr) :
 - FIRST_VALUE(expr)和LAST_VALUE(expr)这两个函数的功能分别是获得滑动窗口范围内参数字段中第一个和最后一个的值。

知识点10 【其他常用函数】 -1



- MySQL中的其他常用函数：

函数	功能
DATABASE()	返回当前数据库名
VERSION()	返回当前数据库版本
USER()	返回当前登录用户名
INET_ATON(IP)	返回IP地址的数字表示
INET_NTOA(num)	返回数字代表的IP地址
PASSWORD(str)	返回字符串str的加密版本
MD5()	返回字符串str的MD5值

Thanks

感谢