

完成功能添加 clickhouse 函数

目录

完成功能添加 clickhouse 函数.....	1
1. 测试有有关 clickhosue 的所有函数 。目前测试的函数包含 ：	1
2.把有关正则的和 sql 函数的代码移动到了新的 Functions 类中。	1

1. 测试有有关 clickhosue 的所有函数 。目前测试的函数包含：

- * clickhouse json 函数
- * clickhosue 字符串函数
- * clickhouse 类型转换函数
- * clickhouse hash 函数
- * clickhouse ip 地址函数
- * clickhouse Nullable 处理函数
- * clickhouse UUID 函数
- * clickhouse 系统函数
- * clickhouse 数学函数

添加的函数包含的是一些常用的函数，没有添加的函数：

- * 一些特别偏的函数，就没有添加了
- * 格式不支持的函数，比如 clickhosue 有数组类型，APIJSON 目前不支持解析数组类型。

2.把有关正则的和 sql 函数的代码移动到了新的 Functions 类中。

因为加上 clickhosue 的函数之后，AbstractSqlConf 的代码多了很多行。而且是 final 修饰的固定的代码。单独放一个类。

添加的函数如下：

1. //clickhouse 字符串函数 注释的函数表示返回的格式暂时不支持，如：返回数组，同时包含因版本不同 clickhosue 不支持的函数，版本

```
2.      SQL_FUNCTION_MAP.put("empty", ""); // empty(s) 对于空字符串 s 返回 1, 对于非空字符串返回 0
3.      SQL_FUNCTION_MAP.put("notEmpty", ""); //notEmpty(s) 对于空字符串返回 0, 对于非空字符串返回 1。
4.      SQL_FUNCTION_MAP.put("lengthUTF8", ""); //假定字符串以 UTF-8 编码组成的文本, 返回此字符串的 Unicode 字符长度。如果传入的字符串不是 UTF-8 编码, 则函数可能返回一个预期外的值
5.      SQL_FUNCTION_MAP.put("lcase", ""); //将字符串中的 ASCII 转换为小写
6.      SQL_FUNCTION_MAP.put("ucase", ""); //将字符串中的 ASCII 转换为大写。
7.      SQL_FUNCTION_MAP.put("lowerUTF8", ""); //将字符串转换为小写, 函数假设字符串是以 UTF-8 编码文本的字符集。
8.      SQL_FUNCTION_MAP.put("upperUTF8", ""); //将字符串转换为大写, 函数假设字符串是以 UTF-8 编码文本的字符集。
9.      SQL_FUNCTION_MAP.put("isValidUTF8", ""); // 检查字符串是否为有效的 UTF-8 编码, 是则返回 1, 否则返回 0。
10.     SQL_FUNCTION_MAP.put("toValidUTF8", "");//用 U+FFFD 字符替换无效的 UTF-8 字符。所有连续的无效字符都会被替换为一个替换字符。
11.     SQL_FUNCTION_MAP.put("reverseUTF8", "");//以 Unicode 字符为单位反转 UTF-8 编码的字符串。
12.     SQL_FUNCTION_MAP.put("concatAssumeInjective", ""); // concatAssumeInjective(s1, s2, ...) 与 concat 相同, 区别在于, 你需要保证 concat(s1, s2, s3) -> s4 是单射的, 它将用于 GROUP BY 的优化。
13.     SQL_FUNCTION_MAP.put("substringUTF8", ""); // substringUTF8(s,offset,length) 与 'substring' 相同, 但其操作单位为 Unicode 字符, 函数假设字符串是以 UTF-8 进行编码的文本。如果不是则可能返回一个预期外的结果 (不会抛出异常)。
14.     SQL_FUNCTION_MAP.put("appendTrailingCharIfAbsent", ""); // appendTrailingCharIfAbsent(s,c) 如果 's' 字符串非空并且末尾不包含 'c' 字符, 则将 'c' 字符附加到末尾
15.     SQL_FUNCTION_MAP.put("convertCharset", ""); // convertCharset(s,from,to) 返回从 'from' 中的编码转换为 'to' 中的编码的字符串 's'。
16.     SQL_FUNCTION_MAP.put("base64Encode", ""); // base64Encode(s) 将字符串 's' 编码成 base64
17.     SQL_FUNCTION_MAP.put("base64Decode", ""); //base64Decode(s) 使用 base64 将字符串解码成原始字符串。如果失败则抛出异常。
18.     SQL_FUNCTION_MAP.put("tryBase64Decode", ""); //tryBase64Decode(s) 使用 base64 将字符串解码成原始字符串。但如果出现错误, 将返回空字符串。
19.     SQL_FUNCTION_MAP.put("endsWith", ""); //endsWith(s,后缀) 返回是否以指定的后缀结尾。如果字符串以指定的后缀结束, 则返回 1, 否则返回 0。
20.     SQL_FUNCTION_MAP.put("startsWith", ""); //startsWith (s, 前缀) 返回是否以指定的前缀开头。如果字符串以指定的前缀开头, 则返回 1, 否则返回 0。
21.     SQL_FUNCTION_MAP.put("trimLeft", ""); //trimLeft(s)返回一个字符串, 用于删除左侧的空白字符。
22.     SQL_FUNCTION_MAP.put("trimRight", ""); //trimRight(s) 返回一个字符串, 用于删除右侧的空白字符。
23.     SQL_FUNCTION_MAP.put("trimBoth", ""); //trimBoth(s), 用于删除任一侧的空白字符
24.     //SQL_FUNCTION_MAP.put("splitByChar", ""); //splitByChar (分隔符, s) 将字符串以 'separator' 拆分成多个子串。'separator' 必须为仅包含一个字符的字符串常量。
25.     // SQL_FUNCTION_MAP.put("splitByString", ""); // splitByString(分隔符, s) 与上面相同, 但它使用多个字符的字符串作为分隔符。 该字符串必须为非空。
26.     SQL_FUNCTION_MAP.put("arrayStringConcat", ""); //arrayStringConcat(arr[,分隔符])使用 separator 将数组中列出的字符串拼接起来。'separator' 是一个可选参数: 一个常量字符串, 默认情况下设置为空字符串。
27.     SQL_FUNCTION_MAP.put("alphaTokens", ""); //alphaTokens(s) 从范围 a-z 和 A-Z 中选择连续字节的子字符串。返回子字符串数组。
28.     //SQL_FUNCTION_MAP.put("splitByRegexp", ""); //splitByRegexp(regexp, s) 将字符串分割为由正则表达式分隔的子字符串。它使用正则表达式字符串 regexp 作为分隔符。
29.     //SQL_FUNCTION_MAP.put("splitByWhitespace", ""); //splitByWhitespace(s) 将字符串分割为由空格字符分隔的子字符串。返回选定子字符串的数组
```

```
30.      //SQL_FUNCTION_MAP.put("splitByNonAlpha", ""); //splitByNonAlpha(s) 将字符串分割为由空格和标点字符分隔的子
      字符串
31.      SQL_FUNCTION_MAP.put("extractAllGroups", ""); //extractAllGroups(text, regexp) 从正则表达式匹配的非重叠子
      字符串中提取所有组
32.      // SQL_FUNCTION_MAP.put("leftPad", ""); //leftPad('string', 'length'[, 'pad_string']) 用空格或指定的字
      符串从左边填充当前字符串(如果需要, 可以多次), 直到得到的字符串达到给定的长度
33.      // SQL_FUNCTION_MAP.put("leftPadUTF8", ""); //leftPadUTF8('string', 'length'[, 'pad_string']) 用空格或指
      定的字符串从左边填充当前字符串(如果需要, 可以多次), 直到得到的字符串达到给定的长度
34.      // SQL_FUNCTION_MAP.put("rightPad", ""); // rightPad('string', 'length'[, 'pad_string']) 用空格或指定的
      字符串(如果需要, 可以多次)从右边填充当前字符串, 直到得到的字符串达到给定的长度
35.      // SQL_FUNCTION_MAP.put("rightPadUTF8", ""); // rightPadUTF8('string', 'length'[, 'pad_string']) 用空格
      或指定的字符串(如果需要, 可以多次)从右边填充当前字符串, 直到得到的字符串达到给定的长度。
36.      SQL_FUNCTION_MAP.put("CRC32", ""); // 使用 CRC-32-IEEE 802.3 多项式和初始值 0xffffffff (zlib 实现)返回字符串
      的 CRC32 校验和
37.      SQL_FUNCTION_MAP.put("CRC32IEEE", ""); // 使用 CRC-32-IEEE 802.3 多项式返回字符串的 CRC32 校验和
38.      SQL_FUNCTION_MAP.put("CRC64", ""); // 使用 CRC-64-ECMA 多项式返回字符串的 CRC64 校验和
39.      SQL_FUNCTION_MAP.put("normalizeQuery", ""); //normalizeQuery(x) 用占位符替换文字、文字序列和复杂的别名。
40.      SQL_FUNCTION_MAP.put("normalizedQueryHash", ""); //normalizedQueryHash(x) 为类似查询返回相同的 64 位散列
      值, 但不包含文字值。有助于对查询日志进行分析
41.      SQL_FUNCTION_MAP.put("encodeXMLComponent", ""); //encodeXMLComponent(x) 转义字符以将字符串放入 XML 文本节
      点或属性中
42.      SQL_FUNCTION_MAP.put("decodeXMLComponent", ""); // 用字符替换 XML 预定义的实体
43.      SQL_FUNCTION_MAP.put("extractTextFromHTML", ""); //extractTextFromHTML(X) 从 HTML 或 XHTML 中提取文本的函
      数
44.      SQL_FUNCTION_MAP.put("positionCaseInsensitive", ""); //positionCaseInsensitive(s, needle[, start_pos])
      与 position 相同, 返回在字符串中找到的子字符串的位置(以字节为单位), 从 1 开始。使用该函数进行不区分大小写的搜索。
45.      SQL_FUNCTION_MAP.put("positionUTF8", ""); // positionUTF8(s, needle[, start_pos]) 返回在字符串中找到的子
      字符串的位置(以 Unicode 点表示), 从 1 开始。
46.      SQL_FUNCTION_MAP.put("positionCaseInsensitiveUTF8", ""); //positionCaseInsensitiveUTF8(s, needle[, start
      _pos]) 与 positionUTF8 相同, 但不区分大小写。返回在字符串中找到的子字符串的位置(以 Unicode 点表示), 从 1 开始
47.      SQL_FUNCTION_MAP.put("multiSearchAllPositions", ""); // 与 position 相同, 但返回字符串中找到的相应子字符串的
      位置数组(以字节为单位)。位置从 1 开始索引。
48.      SQL_FUNCTION_MAP.put("multiSearchAllPositionsUTF8", ""); //See multiSearchAllPositions.
49.      SQL_FUNCTION_MAP.put("multiSearchFirstPosition", ""); // multiSearchFirstPosition(s, [needle1, needle2,
      ..., needlen])
50.      SQL_FUNCTION_MAP.put("multiSearchFirstIndex", ""); //multiSearchFirstIndex(s, [needle1, needle2, ..., nee
      dlen]) 返回字符串 s 中最左边的 needle1 的索引 i(从 1 开始), 否则返回 0
51.      SQL_FUNCTION_MAP.put("multiSearchAny", ""); // multiSearchAny(s, [needle1, needle2, ..., needlen])如果至
      少有一个字符串 needle1 匹配字符串 s, 则返回 1, 否则返回 0。
52.      SQL_FUNCTION_MAP.put("match", ""); //match(s, pattern) 检查字符串是否与模式正则表达式匹配。re2 正则表达式。
      re2 正则表达式的语法比 Perl 正则表达式的语法更有局限性。
53.      SQL_FUNCTION_MAP.put("multiMatchAny", ""); //multiMatchAny(s, [pattern1, pattern2, ..., patternn]) 与
      match 相同, 但是如果没有匹配的正则表达式返回 0, 如果有匹配的模式返回 1
```

```
54.      SQL_FUNCTION_MAP.put("multiMatchAnyIndex", ""); //multiMatchAnyIndex(s, [pattern1, pattern2, ..., pattern
      n]) 与 multiMatchAny 相同，但返回与干堆匹配的任何索引
55.      SQL_FUNCTION_MAP.put("multiMatchAllIndices", ""); //multiMatchAllIndices(s, [pattern1, pattern2, ..., pat
      ternn]) 与 multiMatchAny 相同，但返回以任意顺序匹配干堆的所有索引的数组。
56.      SQL_FUNCTION_MAP.put("multiFuzzyMatchAny", ""); //multiFuzzyMatchAny(s, distance, [pattern1, pattern2,
      ..., patternnn])与 multimchany 相同，但如果在固定的编辑距离内有匹配的模式则返回 1
57.      SQL_FUNCTION_MAP.put("multiFuzzyMatchAnyIndex", ""); //multiFuzzyMatchAnyIndex(s, distance, [pattern1,
      pattern2, ..., patternnn])与 multiFuzzyMatchAny 相同，但返回在固定编辑距离内匹配干草堆的任何索引。
58.      SQL_FUNCTION_MAP.put("multiFuzzyMatchAllIndices", ""); // multiFuzzyMatchAllIndices(s, distance, [patt
      ern1, pattern2, ..., patternnn])
59.      SQL_FUNCTION_MAP.put("extract", ""); // extract(s, pattern) 使用正则表达式提取字符串的片段
60.      SQL_FUNCTION_MAP.put("extractAll", ""); //extractAll(s, pattern) 使用正则表达式提取字符串的所有片段
61.      SQL_FUNCTION_MAP.put("extractAllGroupsHorizontal", ""); //extractAllGroupsHorizontal(s, pattern)使用模
      式正则表达式匹配 s 字符串的所有组
62.      SQL_FUNCTION_MAP.put("extractAllGroupsVertical", ""); //extractAllGroupsVertical(s, pattern) 使用模式正
      则表达式匹配 s 字符串的所有组
63.      SQL_FUNCTION_MAP.put("like", ""); //like(s, pattern) 检查字符串是否与简单正则表达式匹配
64.      SQL_FUNCTION_MAP.put("notLike", ""); // 和 'like' 是一样的，但是是否定的
65.      SQL_FUNCTION_MAP.put("countSubstrings", ""); //countSubstrings(s, needle[, start_pos])返回子字符串出现的
      次数
66.      SQL_FUNCTION_MAP.put("countSubstringsCaseInsensitive", ""); //countSubstringsCaseInsensitive(s, needle[,
      start_pos])返回不区分大小写的子字符串出现次数。
67.      SQL_FUNCTION_MAP.put("countMatches", ""); //返回于 s 中的正则表达式匹配数。countMatches(s, pattern)
68.      SQL_FUNCTION_MAP.put("replaceOne", ""); //replaceOne(s, pattern, replacement)将 ' s '中的 ' pattern '子串
      的第一个出现替换为 ' replacement '子串。
69.
70.      SQL_FUNCTION_MAP.put("replaceAll", ""); //replaceAll(s, pattern, replacement)/用 ' replacement '子串替换
      ' s '中所有出现的 ' pattern '子串
71.      SQL_FUNCTION_MAP.put("replaceRegexpOne", ""); //replaceRegexpOne(s, pattern, replacement)使用
      ' pattern '正则表达式进行替换
72.      SQL_FUNCTION_MAP.put("replaceRegexpAll", ""); //replaceRegexpAll(s, pattern, replacement)
73.      SQL_FUNCTION_MAP.put("regexpQuoteMeta", ""); //regexpQuoteMeta(s)该函数在字符串中某些预定义字符之前添加一
      个反斜杠
74.
75.      //clickhouse 日期函数
76.      SQL_FUNCTION_MAP.put("toYear", ""); //将 Date 或 DateTime 转换为包含年份编号（AD）的 UInt16 类型的数字。
77.      SQL_FUNCTION_MAP.put("toQuarter", ""); //将 Date 或 DateTime 转换为包含季度编号的 UInt8 类型的数字。
78.      SQL_FUNCTION_MAP.put("toMonth", ""); //Date 或 DateTime 转换为包含月份编号（1-12）的 UInt8 类型的数字。
79.      SQL_FUNCTION_MAP.put("toDayOfYear", ""); //将 Date 或 DateTime 转换为包含一年中的某一天的编号的 UInt16（1-
      366）类型的数字。
80.      SQL_FUNCTION_MAP.put("toDayOfMonth", ""); //将 Date 或 DateTime 转换为包含一月中的某一天的编号的 UInt8（1-31）
      类型的数字。
81.      SQL_FUNCTION_MAP.put("toDayOfWeek", ""); //将 Date 或 DateTime 转换为包含一周中的某一天的编号的 UInt8（周一
      是 1，周日是 7）类型的数字。
```

82.	SQL_FUNCTION_MAP.put("toHour", ""); //将 DateTime 转换为包含 24 小时制（0-23）小时数的 UInt8 数字。
83.	SQL_FUNCTION_MAP.put("toMinute", ""); //将 DateTime 转换为包含一小时中分钟数（0-59）的 UInt8 数字。
84.	SQL_FUNCTION_MAP.put("toSecond", ""); //将 DateTime 转换为包含一分钟中秒数（0-59）的 UInt8 数字。
85.	SQL_FUNCTION_MAP.put("toUnixTimestamp", ""); // 对于 DateTime 参数：将值转换为 UInt32 类型的数字-Unix 时间戳
86.	SQL_FUNCTION_MAP.put("toStartOfYear", ""); //将 Date 或 DateTime 向前取整到本年的第一天。
87.	SQL_FUNCTION_MAP.put("toStartOfISOYear", ""); // 将 Date 或 DateTime 向前取整到 ISO 本年的第一天。
88.	SQL_FUNCTION_MAP.put("toStartOfQuarter", ""); //将 Date 或 DateTime 向前取整到本季度的第一天。
89.	SQL_FUNCTION_MAP.put("toStartOfMonth", ""); //将 Date 或 DateTime 向前取整到本月的第一天。
90.	SQL_FUNCTION_MAP.put("toMonday", ""); //将 Date 或 DateTime 向前取整到本周的星期
91.	SQL_FUNCTION_MAP.put("toStartOfWeek", ""); //按 mode 将 Date 或 DateTime 向前取整到最近的星期日或星期一。
92.	SQL_FUNCTION_MAP.put("toStartOfDay", ""); //将 DateTime 向前取整到今天的开始。
93.	SQL_FUNCTION_MAP.put("toStartOfHour", ""); //将 DateTime 向前取整到当前小时的开始。
94.	SQL_FUNCTION_MAP.put("toStartOfMinute", ""); //将 DateTime 向前取整到当前分钟的开始。
95.	SQL_FUNCTION_MAP.put("toStartOfSecond", ""); //将 DateTime 向前取整到当前秒数的开始。
96.	SQL_FUNCTION_MAP.put("toStartOfFiveMinute", ""); //将 DateTime 以五分钟为单位向前取整到最近的时间点。
97.	SQL_FUNCTION_MAP.put("toStartOfTenMinutes", ""); //将 DateTime 以十分钟为单位向前取整到最近的时间点。
98.	SQL_FUNCTION_MAP.put("toStartOfFifteenMinutes", ""); //将 DateTime 以十五分钟为单位向前取整到最近的时间点。
99.	SQL_FUNCTION_MAP.put("toStartOfInterval", ""); //
100.	SQL_FUNCTION_MAP.put("toTime", ""); //将 DateTime 中的日期转换为一个固定的日期，同时保留时间部分。
101.	SQL_FUNCTION_MAP.put("toRelativeYearNum", ""); //将 Date 或 DateTime 转换为年份的编号，从过去的某个固定时间点开始。
102.	SQL_FUNCTION_MAP.put("toRelativeQuarterNum", ""); //同上
103.	SQL_FUNCTION_MAP.put("toRelativeMonthNum", ""); //同上
104.	SQL_FUNCTION_MAP.put("toRelativeWeekNum", ""); //同上
105.	SQL_FUNCTION_MAP.put("toRelativeDayNum", ""); //同上
106.	SQL_FUNCTION_MAP.put("toRelativeHourNum", ""); //同上
107.	SQL_FUNCTION_MAP.put("toRelativeMinuteNum", ""); //同上
108.	SQL_FUNCTION_MAP.put("toRelativeSecondNum", ""); //同上
109.	SQL_FUNCTION_MAP.put("toISOYear", ""); //将 Date 或 DateTime 转换为包含 ISO 年份的 UInt16 类型的编号。
110.	SQL_FUNCTION_MAP.put("toISOWeek", ""); //
111.	SQL_FUNCTION_MAP.put("toWeek", ""); // 返回 Date 或 DateTime 的周数。
112.	SQL_FUNCTION_MAP.put("toYearWeek", ""); //返回年和周的日期
113.	SQL_FUNCTION_MAP.put("date_trunc", ""); //截断日期和时间数据到日期的指定部分
114.	SQL_FUNCTION_MAP.put("date_diff", ""); //回两个日期或带有时间值的日期之间的差值。
115.	
116.	SQL_FUNCTION_MAP.put("yesterday", ""); //不接受任何参数并在请求执行时的某一刻返回昨天的日期(Date)。
117.	SQL_FUNCTION_MAP.put("today", ""); //不接受任何参数并在请求执行时的某一刻返回当前日期(Date)。
118.	SQL_FUNCTION_MAP.put("timeSlot", ""); //将时间向前取整半小时。
119.	SQL_FUNCTION_MAP.put("toYYYYMM", ""); //
120.	SQL_FUNCTION_MAP.put("toYYYYMMDD", ""); //
121.	SQL_FUNCTION_MAP.put("toYYYYMMDDhhmmss", ""); //

```

122.      SQL_FUNCTION_MAP.put("addYears", ""); // Function adds a Date/DateTime interval to a Date/DateTime and
      then return the Date/DateTime
123.      SQL_FUNCTION_MAP.put("addMonths", ""); //同上
124.      SQL_FUNCTION_MAP.put("addWeeks", ""); //同上
125.      SQL_FUNCTION_MAP.put("addDays", ""); //同上
126.      SQL_FUNCTION_MAP.put("addHours", ""); //同上
127.      SQL_FUNCTION_MAP.put("addMinutes", ""); //同上
128.      SQL_FUNCTION_MAP.put("addSeconds", ""); //同上
129.      SQL_FUNCTION_MAP.put("addQuarters", ""); //同上
130.      SQL_FUNCTION_MAP.put("subtractYears", ""); //Function subtract a Date/DateTime interval to a Date/DateT
      ime and then return the Date/DateTime
131.      SQL_FUNCTION_MAP.put("subtractMonths", ""); //同上
132.      SQL_FUNCTION_MAP.put("subtractWeeks", ""); //同上
133.      SQL_FUNCTION_MAP.put("subtractDays", ""); //同上
134.      SQL_FUNCTION_MAP.put("subtractours", ""); //同上
135.      SQL_FUNCTION_MAP.put("subtractMinutes", ""); //同上
136.      SQL_FUNCTION_MAP.put("subtractSeconds", ""); //同上
137.      SQL_FUNCTION_MAP.put("subtractQuarters", ""); //同上
138.      SQL_FUNCTION_MAP.put("formatDateTime", ""); //函数根据给定的格式字符串来格式化时间
139.      SQL_FUNCTION_MAP.put("timestamp_add", ""); //使用提供的日期或日期时间值添加指定的时间值。
140.      SQL_FUNCTION_MAP.put("timestamp_sub", ""); //从提供的日期或带时间的日期中减去时间间隔。
141.
142.      //clickhouse json 函数
143.      SQL_FUNCTION_MAP.put("visitParamHas", ""); //visitParamHas(params, name)检查是否存在«name»名称的字段
144.      SQL_FUNCTION_MAP.put("visitParamExtractUInt", ""); //visitParamExtractUInt(params, name)将名为«name»的字
      段的值解析成 UInt64。
145.      SQL_FUNCTION_MAP.put("visitParamExtractInt", ""); //与 visitParamExtractUInt 相同，但返回 Int64。
146.      SQL_FUNCTION_MAP.put("visitParamExtractFloat", ""); //与 visitParamExtractUInt 相同，但返回 Float64。
147.      SQL_FUNCTION_MAP.put("visitParamExtractBool", ""); //解析 true/false 值。其结果是 UInt8 类型的。
148.      SQL_FUNCTION_MAP.put("visitParamExtractRaw", ""); //返回字段的值，包含空格符。
149.      SQL_FUNCTION_MAP.put("visitParamExtractString", ""); //使用双引号解析字符串。这个值没有进行转义。如果转义失
      败，它将返回一个空白字符串。
150.      SQL_FUNCTION_MAP.put("JSONHas", ""); //如果 JSON 中存在该值，则返回 1。
151.      SQL_FUNCTION_MAP.put("JSONLength", ""); //返回 JSON 数组或 JSON 对象的长度。
152.      SQL_FUNCTION_MAP.put("JSONType", ""); //返回 JSON 值的类型。
153.      SQL_FUNCTION_MAP.put("JSONExtractUInt", ""); //解析 JSON 并提取值。这些函数类似于 visitParam*函数。
154.      SQL_FUNCTION_MAP.put("JSONExtractInt", ""); //
155.      SQL_FUNCTION_MAP.put("JSONExtractFloat", ""); //
156.      SQL_FUNCTION_MAP.put("JSONExtractBool", ""); //
157.      SQL_FUNCTION_MAP.put("JSONExtractString", ""); //解析 JSON 并提取字符串。此函数类似于
      visitParamExtractString 函数。
158.      SQL_FUNCTION_MAP.put("JSONExtract", ""); //解析 JSON 并提取给定 ClickHouse 数据类型的值。
159.      SQL_FUNCTION_MAP.put("JSONExtractKeysAndValues", ""); //从 JSON 中解析键值对，其中值是给定的 ClickHouse 数据
      类型

```

```

160.     SQL_FUNCTION_MAP.put("JSONExtractRaw", ""); //返回 JSON 的部分。
161.     SQL_FUNCTION_MAP.put("toJSONString", ""); //
162.
163.     //clickhouse 类型转换函数
164.     SQL_FUNCTION_MAP.put("toInt8", ""); //toInt8(expr) 转换一个输入值为 Int 类型
165.     SQL_FUNCTION_MAP.put("toInt16", "");
166.     SQL_FUNCTION_MAP.put("toInt32", "");
167.     SQL_FUNCTION_MAP.put("toInt64", "");
168.     SQL_FUNCTION_MAP.put("toInt8OrNull", ""); //toInt(8|16|32|64)OrNull 这个函数需要一个字符类型的入参，然后尝
    试把它转为 Int (8 | 16 | 32 | 64)，如果转换失败直接返回 0。
169.     SQL_FUNCTION_MAP.put("toInt16OrNull", "");
170.     SQL_FUNCTION_MAP.put("toInt32OrNull", "");
171.     SQL_FUNCTION_MAP.put("toInt64OrNull", "");
172.     SQL_FUNCTION_MAP.put("toInt8OrNull", ""); //toInt(8|16|32|64)OrNull 这个函数需要一个字符类型的入参，然后尝
    试把它转为 Int (8 | 16 | 32 | 64)，如果转换失败直接返回 NULL
173.     SQL_FUNCTION_MAP.put("toInt16OrNull", "");
174.     SQL_FUNCTION_MAP.put("toInt32OrNull", "");
175.     SQL_FUNCTION_MAP.put("toInt64OrNull", "");
176.     SQL_FUNCTION_MAP.put("toUInt8", ""); //toInt8(expr) 转换一个输入值为 Int 类型
177.     SQL_FUNCTION_MAP.put("toUInt16", "");
178.     SQL_FUNCTION_MAP.put("toUInt32", "");
179.     SQL_FUNCTION_MAP.put("toUInt64", "");
180.     SQL_FUNCTION_MAP.put("toUInt8OrNull", ""); //toInt(8|16|32|64)OrNull 这个函数需要一个字符类型的入参，然后
    尝试把它转为 Int (8 | 16 | 32 | 64)，如果转换失败直接返回 0。
181.     SQL_FUNCTION_MAP.put("toUInt16OrNull", "");
182.     SQL_FUNCTION_MAP.put("toUInt32OrNull", "");
183.     SQL_FUNCTION_MAP.put("toUInt64OrNull", "");
184.     SQL_FUNCTION_MAP.put("toUInt8OrNull", ""); //toInt(8|16|32|64)OrNull 这个函数需要一个字符类型的入参，然后尝
    试把它转为 Int (8 | 16 | 32 | 64)，如果转换失败直接返回 NULL
185.     SQL_FUNCTION_MAP.put("toUInt16OrNull", "");
186.     SQL_FUNCTION_MAP.put("toUInt32OrNull", "");
187.     SQL_FUNCTION_MAP.put("toUInt64OrNull", "");
188.
189.     SQL_FUNCTION_MAP.put("toFloat32", "");
190.     SQL_FUNCTION_MAP.put("toFloat64", "");
191.     SQL_FUNCTION_MAP.put("toFloat32OrNull", "");
192.     SQL_FUNCTION_MAP.put("toFloat64OrNull", "");
193.     SQL_FUNCTION_MAP.put("toFloat32OrZero", "");
194.     SQL_FUNCTION_MAP.put("toFloat64OrZero", "");
195.
196.     SQL_FUNCTION_MAP.put("toDate", ""); //
197.     SQL_FUNCTION_MAP.put("toDateOrZero", ""); //toInt16(expr)
198.     SQL_FUNCTION_MAP.put("toDateOrNull", ""); //toInt32(expr)
199.     SQL_FUNCTION_MAP.put("toDateTimeOrZero", ""); //toInt64(expr)

```

```

200.         SQL_FUNCTION_MAP.put("toDateTimeOrNull", ""); //toInt(8|16|32|64)OrZero 这个函数需要一个字符类型的入参，然
            后尝试把它转为 Int (8 | 16 | 32 | 64)，如果转换失败直接返回 0。

201.

202.         SQL_FUNCTION_MAP.put("toDecimal32", "");

203.         SQL_FUNCTION_MAP.put("toFixedString", ""); // 将 String 类型的参数转换为 FixedString(N)类型的值

204.         SQL_FUNCTION_MAP.put("toStringCutToZero", ""); // 接受 String 或 FixedString 参数,返回 String，其内容在找到
            的第一个零字节处被截断。

205.         SQL_FUNCTION_MAP.put("toDecimal1256", "");

206.         SQL_FUNCTION_MAP.put("toDecimal32OrNull", "");

207.         SQL_FUNCTION_MAP.put("toDecimal64OrNull", "");

208.         SQL_FUNCTION_MAP.put("toDecimal1280OrNull", "");

209.         SQL_FUNCTION_MAP.put("toDecimal2560OrNull", "");

210.         SQL_FUNCTION_MAP.put("toDecimal32OrZero", "");

211.         SQL_FUNCTION_MAP.put("toDecimal64OrZero", "");

212.         SQL_FUNCTION_MAP.put("toDecimal1280OrZero", "");

213.         SQL_FUNCTION_MAP.put("toDecimal2560OrZero", "");

214.

215.

216.         SQL_FUNCTION_MAP.put("toIntervalSecond", ""); //把一个数值类型的值转换为 Interval 类型的数据。

217.         SQL_FUNCTION_MAP.put("toIntervalMinute", "");

218.         SQL_FUNCTION_MAP.put("toIntervalHour", "");

219.         SQL_FUNCTION_MAP.put("toIntervalDay", "");

220.         SQL_FUNCTION_MAP.put("toIntervalWeek", "");

221.         SQL_FUNCTION_MAP.put("toIntervalMonth", "");

222.         SQL_FUNCTION_MAP.put("toIntervalQuarter", "");

223.         SQL_FUNCTION_MAP.put("toIntervalYear", "");

224.         SQL_FUNCTION_MAP.put("parseDateTimeBestEffort", ""); //把 String 类型的时间日期转换为 DateTime 数据类型。

225.         SQL_FUNCTION_MAP.put("parseDateTimeBestEffortOrNull", "");

226.         SQL_FUNCTION_MAP.put("parseDateTimeBestEffortOrZero", "");

227.         SQL_FUNCTION_MAP.put("toLowCardinality", "");

228.         SQL_FUNCTION_MAP.put("toUnixTimestamp64Milli", "");

229.         SQL_FUNCTION_MAP.put("toUnixTimestamp64Micro", "");

230.         SQL_FUNCTION_MAP.put("toUnixTimestamp64Nano", "");

231.         SQL_FUNCTION_MAP.put("fromUnixTimestamp64Milli", "");

232.         SQL_FUNCTION_MAP.put("fromUnixTimestamp64Micro", "");

233.         SQL_FUNCTION_MAP.put("fromUnixTimestamp64Nano", "");

234.

235.

236.         ///clickhouse hash 函数

237.         SQL_FUNCTION_MAP.put("halfMD5", ""); //计算字符串的 MD5。然后获取结果的前 8 个字节并将它们作为 UInt64（大端）
            返回

238.         SQL_FUNCTION_MAP.put("MD5", ""); //计算字符串的 MD5 并将结果放入 FixedString(16)中返回

239.

240.         //clickhouse ip 地址函数

```



```

241.         SQL_FUNCTION_MAP.put("IPv4NumToString", ""); //接受一个 UInt32（大端）表示的 IPv4 的地址，返回相应 IPv4 的字符串表现形式，格式为 A.B.C.D（以点分割的十进制数字）。

242.         SQL_FUNCTION_MAP.put("IPv4StringToNum", ""); //与 IPv4NumToString 函数相反。如果 IPv4 地址格式无效，则返回 0。

243.         SQL_FUNCTION_MAP.put("IPv4NumToStringClassC", ""); // 与 IPv4NumToString 类似，但使用 xxx 替换最后一节。

244.         SQL_FUNCTION_MAP.put("IPv6NumToString", ""); //接受 FixedString(16)类型的二进制格式的 IPv6 地址。以文本格式返回此地址的字符串。

245.         SQL_FUNCTION_MAP.put("IPv6StringToNum", ""); //与 IPv6NumToString 的相反。如果 IPv6 地址格式无效，则返回空字节字符串。

246.         SQL_FUNCTION_MAP.put("IPv4ToIPv6", ""); // 接受一个 UInt32 类型的 IPv4 地址，返回 FixedString(16)类型的 IPv6 地址

247.         SQL_FUNCTION_MAP.put("cutIPv6", ""); //接受一个 FixedString(16)类型的 IPv6 地址，返回一个 String，这个 String 中包含了删除指定位之后的地址的文本格式

248.         SQL_FUNCTION_MAP.put("toIPv4", ""); //IPv4StringToNum()的别名。

249.         SQL_FUNCTION_MAP.put("toIPv6", ""); //IPv6StringToNum()的别名

250.         SQL_FUNCTION_MAP.put("isIPAddressInRange", ""); //确定一个 IP 地址是否包含在以 CIDR 符号表示的网络中

251.

252.         //clickhouse Nullable 处理函数

253.         SQL_FUNCTION_MAP.put("isNull", ""); //检查参数是否为 NULL。

254.         SQL_FUNCTION_MAP.put("isNotNull", ""); //检查参数是否不为 NULL。

255.         SQL_FUNCTION_MAP.put("ifNull", ""); //如果第一个参数为«NULL»，则返回第二个参数的值。

256.         SQL_FUNCTION_MAP.put("assumeNotNull", ""); //将可为空类型的值转换为非 Nullable 类型的值。

257.         SQL_FUNCTION_MAP.put("toNullable", ""); //将参数的类型转换为 Nullable。

258.

259.         //clickhouse UUID 函数

260.         SQL_FUNCTION_MAP.put("generateUUIDv4", ""); // 生成一个 UUID

261.         SQL_FUNCTION_MAP.put("toUUID", ""); //toUUID(x) 将 String 类型的值转换为 UUID 类型的值。

262.

263.         //clickhouse 系统函数

264.         SQL_FUNCTION_MAP.put("hostName", ""); //hostName() 回一个字符串，其中包含执行此函数的主机的名称。

265.         SQL_FUNCTION_MAP.put("getMacro", ""); //从服务器配置的宏部分获取指定值。

266.         SQL_FUNCTION_MAP.put("FQDN", ""); //返回完全限定的域名。

267.         SQL_FUNCTION_MAP.put("basename", ""); //提取字符串最后一个斜杠或反斜杠之后的尾随部分

268.         SQL_FUNCTION_MAP.put("currentUser", ""); //返回当前用户的登录。在分布式查询的情况下，将返回用户的登录，即发起的查询

269.         SQL_FUNCTION_MAP.put("version", ""); //以字符串形式返回服务器版本。

270.         SQL_FUNCTION_MAP.put("uptime", ""); //以秒为单位返回服务器的正常运行时间。

271.

272.         //

273.         SQL_FUNCTION_MAP.put("least", ""); //返回 a 和 b 中最小的值。

274.         SQL_FUNCTION_MAP.put("greatest", ""); //返回 a 和 b 的最大值。

275.         SQL_FUNCTION_MAP.put("plus", ""); //plus(a, b), a + b operator 计算数值的总和。

276.         SQL_FUNCTION_MAP.put("minus", ""); //minus(a, b), a - b operator 计算数值之间的差，结果总是有符号的。

```

```
277.      SQL_FUNCTION_MAP.put("multiply", ""); //multiply(a, b), a * b operator 计算数值的乘积
278.      SQL_FUNCTION_MAP.put("divide", ""); //divide(a, b), a / b operator 计算数值的商。结果类型始终是浮点类型
279.      SQL_FUNCTION_MAP.put("intDiv", ""); //intDiv(a,b)计算数值的商，向下舍入取整（按绝对值）。
280.      SQL_FUNCTION_MAP.put("intDivOrZero", ""); // intDivOrZero(a,b)与'intDiv'的不同之处在于它在除以零或将最小
      负数除以-1 时返回零。
281.      SQL_FUNCTION_MAP.put("modulo", ""); //modulo(a, b), a % b operator 计算除法后的余数。
282.      SQL_FUNCTION_MAP.put("moduloOrZero", ""); //和 modulo 不同之处在于，除以 0 时结果返回 0
283.      SQL_FUNCTION_MAP.put("negate", ""); //通过改变数值的符号位对数值取反，结果总是有符号
284.      SQL_FUNCTION_MAP.put("gcd", ""); //gcd(a,b) 返回数值的最大公约数。
285.      SQL_FUNCTION_MAP.put("lcm", ""); //lcm(a,b) 返回数值的最小公倍数
286.      SQL_FUNCTION_MAP.put("e", ""); //e() 返回一个接近数学常量 e 的 Float64 数字。
287.      SQL_FUNCTION_MAP.put("pi", ""); //pi() 返回一个接近数学常量  $\pi$  的 Float64 数字。
288.      SQL_FUNCTION_MAP.put("exp2", ""); //exp2(x)接受一个数值类型的参数并返回它的 2 的 x 次幂。
289.      SQL_FUNCTION_MAP.put("exp10", ""); //exp10(x)接受一个数值类型的参数并返回它的 10 的 x 次幂。
290.      SQL_FUNCTION_MAP.put("cbrt", ""); //cbrt(x) 接受一个数值类型的参数并返回它的立方根。
291.      SQL_FUNCTION_MAP.put("erf", ""); //erf(x) 如果 'x' 是非负数，那么  $\text{erf}(x / \sigma\sqrt{2})$  是具有正态分布且标准偏差为  $\sigma$  的
      随机变量的值与预期值之间的距离大于  $\sigma x$ 
292.      SQL_FUNCTION_MAP.put("erfc", ""); //erfc(x) 接受一个数值参数并返回一个接近  $1 - \text{erf}(x)$  的 Float64 数字，但不会
      丢失大  $\sigma x$  值的精度。
293.      SQL_FUNCTION_MAP.put("lgamma", ""); //lgamma(x) 返回 x 的绝对值的自然对数的伽玛函数。
294.      SQL_FUNCTION_MAP.put("tgamma", ""); //tgamma(x)返回 x 的伽玛函数。
295.      SQL_FUNCTION_MAP.put("intExp2", ""); //intExp2 接受一个数值类型的参数并返回它的 2 的 x 次幂 (UInt64)
296.      SQL_FUNCTION_MAP.put("intExp10", ""); //intExp10 接受一个数值类型的参数并返回它的 10 的 x 次幂 (UInt64)。
297.      SQL_FUNCTION_MAP.put("cosh", ""); // cosh(x)
298.      SQL_FUNCTION_MAP.put("sinh", ""); //sinh(x)
299.      SQL_FUNCTION_MAP.put("asinh", ""); //asinh(x)
300.      SQL_FUNCTION_MAP.put("atanh", ""); //atanh(x)
301.      SQL_FUNCTION_MAP.put("atan2", ""); //atan2(y, x)
302.      SQL_FUNCTION_MAP.put("hypot", ""); //hypot(x, y)
303.      SQL_FUNCTION_MAP.put("log1p", ""); //log1p(x)
304.      SQL_FUNCTION_MAP.put("trunc", ""); //和 truncate 一样
305.      SQL_FUNCTION_MAP.put("roundToExp2", ""); //接受一个数字。如果数字小于 1，它返回 0。
306.      SQL_FUNCTION_MAP.put("roundDuration", ""); //接受一个数字。如果数字小于 1，它返回 0。
307.      SQL_FUNCTION_MAP.put("roundAge", ""); // 接受一个数字。如果数字小于 18，它返回 0。
308.      SQL_FUNCTION_MAP.put("roundDown", ""); //接受一个数字并将其舍入到指定数组中的一个元素
309.      SQL_FUNCTION_MAP.put("bitAnd", ""); //bitAnd(a,b)
310.      SQL_FUNCTION_MAP.put("bitOr", ""); //bitOr(a,b)
```