Sybase iq 数据库函数总结

建数据库脚本：

CREATE DATABASE 'e:\Program Files\Sybase\ASIQ-12\_7\myiq\myiq.db'

TRANSACTION LOG ON

COLLATION '936ZHO'

PAGE SIZE 4096

JAVA ON

JCONNECT ON

CASE RESPECT

PASSWORD CASE RESPECT

IQ PATH 'myiq.iq'

IQ SIZE 100

IQ PAGE SIZE 131072

TEMPORARY PATH 'myiq.iqtmp'

TEMPORARY SIZE 10

SET OPTION public.MINIMIZE\_STORAGE = ON ;

这个可以在数据库中设置一下，这样减轻字段存储空间的压力

--select \* from all\_ind\_columns where table\_name='TB\_CM\_SERV' and table\_owner='IODATA\_GZ';

数据文件的使用率：

Create procedure myspace()

Begin

Declare mt unsigned bigint;

Declare mu unsigned bigint;

Declare tt unsigned bigint;

Declare tu unsigned bigint;

Call sp\_iqspaceused (mt,mu,tt,tu);

Select cast(mt/1024 as unsigned bigint) as mainMB,

Cast(mu/1024 as unsigned bigint) as mainusedMB,

Mu\*100/mt as mainPerCent,

Cast(tt/1024 as unsigned bigint) as tempMB,

Cast(tu/1024 as unsigned bigint) as tempusedMB,

Tu\*100/tt as tempPerCent;

End

# 数据类型

## 日期及时间数据类型

**DATE** 日历日期，如年、月、日。年可以从 0001 到 9999。日必须是非零值，以便最早日期为 0001-01-01。DATE 值需要 4 个字节的存储空间。

**DATETIME** 一个以 TIMESTAMP 的形式实现的域。之所以提供 DATETIME，主要是为了与 Adaptive Server Enterprise 兼容。有关例外，请参见“字符串到日期时间转换的兼容性”。

**SMALLDATETIME** 一个以 TIMESTAMP 的形式实现的域。

**TIME** 一天中的时间，包含小时、分钟、秒，以及秒的小数部分。分数存储为 6 位小数。TIME 值需要 8 个字节的存储空间。（ODBC 标准将 TIME数据类型限制为精确到秒。因此，不要在要求的精度比秒更高的 WHERE子句比较中使用 TIME 数据类型。）

**TIMESTAMP** 时间点，包含年、月、日、小时、分钟、秒，以及秒的小数部分。分数存储为 6 位小数。日必须为非零值。TIMESTAMP 值需要 8 个字节的存储空间。

## 日期和时间格式

以下是用于指定时间的有效而明确的字符串：

21:35 -- 24 小时制（如果未指定 am 或 pm）

10:00pm -- 已指定 pm，因此为 12 小时制

10:00 -- 不带 pm，因此是指 10:00am

10:23:32.234 -- 包括秒以及秒的小数部分

在向数据库发送时间是，小时、分钟和秒必须用冒号分隔开（格式为 *hh*:*mm*:*ss*:*sss*），但可以显示在字符串中的任意位置。可以选择用句点将秒与秒的小数部分相互分隔开，如*hh*:*mm*:*ss*.*sss*。

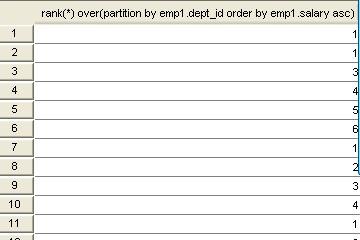
将日期以字符串形式发送到数据库时，其到日期数据类型的转换是自动发生的。您可以通过下面两种方式之一提供字符串：

• 以 *yyyy/mm/dd* 或 *yyyy-mm-dd* 格式的字符串形式，其格式由数据库明确解释

# 函数

## Rank 窗口化集合函数 分区函数的用法

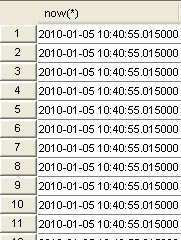
select rank() over(partition by dept\_id order by salary) from emp1;



Partition by 后一定要加order by

## Now 函数

select now(\*) from emp1



## ARGN 函数 [Miscellaneous]

从参数列表中返回所选参数。

**语法： ARGN** ( *integer-expression*, *expression* [ , ...] )

SELECT ARGN( 4, 1,2,3,7,5,6 ) FROM iq\_dummy;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\6U66UZ9MSK]T)]Q29%P[7_M.jpg

SELECT ARGN( 4, 'sdf','asd','sdfsdrg','asrewryry','345sdfs') FROM iq\_dummy;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\WM3AL6ZB8B8V09NPISUKG{K.jpg

## ARGN 函数 [Miscellaneous]

返回字符串表达式中第一个字节的整数 ASCII 值。

语法：**ASCII** ( *string-expression* )

SELECT ASCII( 'Z' ) FROM iq\_dummy

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\DNRZ@XQSDN_I4VF@7`78%MF.jpg

## AVG 函数 [Aggregate]

为一组行计算数值表达式的平均值，或计算一组唯一值的平均值。

语法：AVG ( *numeric-expression* | DISTINCT *column-name* )

## ABIGNTTOHEX 函数[Data type conversion]

在 VARCHAR(16) 中返回十进制整数的十六进制等效数字。

语法：**BIGINTTOHEX** ( *integer-expression* )

下面的语句返回值 0000000000000009：

SELECT BIGINTTOHEX(9) FROM iq\_dummy;

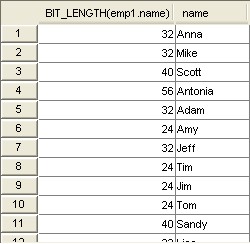
下面的语句返回值 FFFFFFFFFFFFFFF7：

SELECT BIGINTTOHEX(-9) FROM iq\_dummy;

## BIT\_LENGTH 函数 [String]

返回字符串中的位数，即字符数\*8

select BIT\_LENGTH(name),name from emp1;



## BYTE\_LENGH 函数 [String]

返回字符串中的字节数。

语法：BYTE\_LENGTH ( *string-expression* )

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\ILBRSYLD@G]RN3D36%90AJ6.jpg

## CAST 函数 [Data type conversion]

返回转换为提供的数据类型的表达式的值。

语法：**CAST** ( *expression* **AS** *data type* )

下面的函数可确保使用字符串作为日期：

CAST( '2000-10-31' AS DATE )

计算表达式 1 + 2 的值，并将结果转换为单字符字符串（即数据服务器分配的长度）：

CAST( 1 + 2 AS CHAR )

可以使用 CAST 函数缩短字符串：

SELECT CAST( lname AS CHAR(5) ) FROM customer

用法如果未指定字符串类型的长度，Sybase IQ 将选择适当的长度。如果没有为十进制转换指定精度和标度，则数据库服务器将选择适当的值。如果针对 NULL 到 NUMERIC 的显式转换既未指定精度也未指定标度，则缺省值为 NUMERIC(1,0)。例如，

SELECT CAST( NULL AS NUMERIC ) A,

CAST( NULL AS NUMERIC(15,2) ) B

说明为：

A NUMERIC(1,0)

B NUMERIC(15,2)

## CEIL 函数 [Numeric]

返回大于或等于指定表达式的最小整数。

## ceiling 函数 [Numeric]

返回一个数字的上限（不小于的最小整数）。 与ceil为同义词。

语法：**CEILING** ( *numeric-expression* )

下面的语句返回值 60.00000：

SELECT CEILING( 59.84567 ) FROM iq\_dummy

下面的语句返回值 123：

SELECT CEILING( 123 ) FROM iq\_dummy

下面的语句返回值 124.00：

SELECT CEILING( 123.45 ) FROM iq\_dummy

下面的语句返回值 -123.00：

SELECT CEILING( -123.45 ) FROM iq\_dummy

## CHAR 函数 [String]

返回含数字所表示的 ASCII 值的字符。

语法：**CHAR** ( *integer-expression* )

下面的语句返回值“Y”：

SELECT CHAR( 89 ) FROM iq\_dummy

下面的语句返回值“S”：

SELECT CHAR( 83 ) FROM iq\_dummy

对于值大于 255 或小于 0 的整数表达式， CHAR 返回 NULL。

## CHAR\_LENGTH 函数 [String]

返回字符串中的字符数。

语法：**CHAR\_LENGTH** ( *string-expression* )

下面的语句返回值 8：

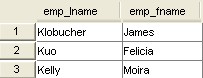
SELECT CHAR\_LENGTH( 'Chemical' ) FROM iq\_dummy;

## CHARINDEX 函数 [String]

返回一个字符串在另一个字符串中第一次出现的位置。

语法：**CHARINDEX** ( *string-expression1, string-expression2* )

SELECT emp\_lname, emp\_fname FROM employee WHERE CHARINDEX('K', emp\_lname ) = 1



select charindex(upper('a'),name),name from emp1;

## COALESCE 函数 [Miscellaneous]

返回列表中的第一个非空表达式。

语法：**COALESCE** ( *expression*, *expression* [ , ...] )

## COL\_NAME 函数 [System]

返回列名。

语法：**COL\_NAME** ( *table-id*, *column-id* [, *database-id* ] )

下面的语句返回值“Y”：

SELECT CHAR( 89 ) FROM iq\_dummy

下面的语句返回值“S”：

SELECT CHAR( 83 ) FROM iq\_dummy

对于值大于 255 或小于 0 的整数表达式， CHAR 返回 NULL。

## CONNECTION\_PROPERTY 函数 [System]

以字符串形式返回给定连接属性的值。

语法：**CONNECTION\_PROPERTY** ( { *integer-expression1* | *string-expression* }

... [ , *integer-expression2* ] )

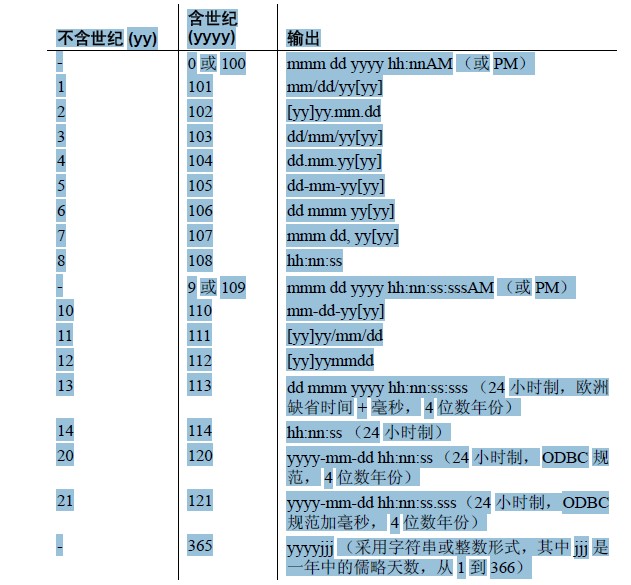
SELECT connection\_property( 'PrepStmt' ) FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\N}SE7620K9}]7{`XN$~9Q80.jpg

## CONVERT 函数 [Data type conversion]

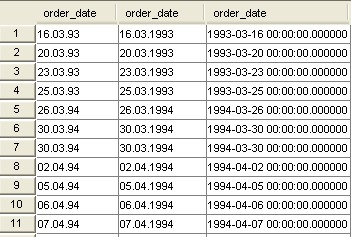
返回转换成提供的数据类型的表达式。

语法：**CONVERT** ( *data-type*, *expression* [, *format-style* ] )



0、9 表示都是含世纪的，其他如1、2都是可以不含世纪的。

SELECT CONVERT( CHAR( 20 ), order\_date, 4),CONVERT( CHAR( 20 ), order\_date, 104),order\_date FROM sales\_order；



## DATALENGTH 函数 [System]

返回表达式的长度（以字节为单位）。

语法：**DATALENGTH** ( *expression* )

下面的语句返回值 35，即 company\_name 列中最长的字符串：

SELECT MAX( DATALENGTH( company\_name ) ) FROM customer

## DATE 函数 [Date and time]

将表达式转换为日期，并删除任何小时、分钟或秒。

语法：**DATE** ( *expression* )

下面的语句返回值 1988-11-26 作为日期。

SELECT DATE( '1988-11-26 21:20:53' ) FROM iq\_dummy

## DATEADD 函数 [Date and time]

返回通过将指定数量的指定日期分量添加到日期中而生成的日期。

语法：**DATEADD** ( *date-part*, *numeric-expression*, *date-expression* )

**date part** 要添加到日期中的日期分量。

**numeric-expression** 要添加到日期中的日期分量的数量。

*numeric-expression* 可以是任何数值类型；值截断为整数。

**date-expression** 要修改的日期。

下面的语句返回值 1995-11-02 00:00:00.000：

SELECT DATEADD( month, 102, '1987/05/02' ) FROM iq\_dummy

## DATEDIFF 函数 [Date and time]

返回两个日期之间的时间间隔。

语法：**DATEDIFF** ( *date-part*, *date-expression1*, *date-expression2* )

**date-expression1** 间隔的开始日期。从 *date-expression-2* 中减去该值，以返回两个参数之间日期分量的数量。

**date-expression2** 时间间隔的结束日期。从此值中减去 *date-expression1*以返回两个参数间的日期分量的数量。

下面的语句返回 1：

SELECT DATEDIFF( hour, '4:00AM', '5:50AM' ) FROM iq\_dummy;

下面的语句返回 102：

SELECT DATEDIFF( month, '1987/05/02', '1995/11/15' ) FROM iq\_dummy;

下面的语句返回 0：

SELECT DATEDIFF( day, '00:00', '23:59' ) FROM iq\_dummy ;

下面的语句返回 4：

SELECT DATEDIFF( day, '1999/07/19 00:00', '1999/07/23:23:59' ) FROM iq\_dummy;

下面的语句返回 0：

SELECT DATEDIFF( month, '1999/07/19', '1999/07/23' ) FROM iq\_dummy;

下面的语句返回 1：

SELECT DATEDIFF( month, '1999/07/19', '1999/08/23' ) FROM iq\_dummy;

## DATEFORMAT 函数 [Date and time]

以指定的格式返回表示日期表达式的字符串。

语法：**DATEFORMAT** ( *datetime-expression*, *string-expression* )

**datetime-expression** 要转换的日期/时间。必须是日期、时间、时间戳或字符串。

**string-expression** 转换后的日期格式。

下面的语句返回类似“Jan 01, 1989”的字符串值：

SELECT DATEFORMAT( start\_date, 'Mmm dd, yyyy' ) from employee;

下面的语句返回字符串“Feb 19, 1987”。

SELECT DATEFORMAT( CAST ( ‘1987/02/19’ AS DATE ), ‘MmmDd, yyyy’ ) FROM iq\_dummy

## DATEPART 函数 [Date and time]

返回日期/时间值的指定部分的整数值。

语法：**date-part** 要返回的日期分量。

有关允许的日期分量的完整列表，请参见“日期分量”（第 235 页）。

**date-expression** 要返回其日期分量的日期。日期必须包含 *date-part*字段。

SELECT DATEPART( month , '1987/05/02' ) FROM iq\_dummy;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\U]UK_N~884B@%PI[%K8)686.jpg

## DATENAME 函数 [Date and time]

以字符串形式返回日期/时间值中指定部分的名称（如月份“June”）

语法：**DATENAME** ( *date-part*, *date-expression* )

**date-part** 要指定的日期分量。

**date-expression** 要返回其日期分量名称的日期。日期必须包含请求的*date-part*。

下面的语句返回值 May：

SELECT datename( month , '1987/05/02' ) FROM iq\_dummy

## DATETIME 函数 [Date and time]

将表达式转换成时间戳。

语法：**DATETIME** ( *expression* )

**expression** 要转换的表达式。表达式通常是字符串。可能会报告转换错误。

SELECT DATETIME( '1998-09-09 12:12:12.000' ) FROMiq\_dummy

返回含有值 1998-09-09 12:12:12.000 的时间戳：

## DAY 函数 [Date and time]

返回 1 到 31 之间的一个整数，该整数对应于指定日期中的每月几号。

语法：**DAY** ( *date-expression* )

下面的语句返回值 12：

SELECT DAY( '2001-09-12' ) FROM iq\_dummy

## DAYNAME 函数 [Date and time]

返回指定日期中的星期几的名称。

语法：**COALESCE** ( *expression*, *expression* [ , ...] )

下面的语句返回值 Saturday：

SELECT DAYNAME ( '1987/05/02' ) FROM iq\_dummy

## DAYS 函数 [Date and time]

返回自任意开始日期以来的天数，返回两个指定日期之间的天数，或者将指定的 *integer-expression* 数量的天添加到给定日期中。

语法

**DAYS** ( *datetime-expression* )| ( *datetime-expression*, *datetime-expression* )| ( *datetime-expression*, *integer-expression* )

参数**datetime-expression** 日期和时间。

**integer-expression** 要添加到 *datetime-expression* 中的天数。如果*integer-expression* 是负数，则从日期/时间中减去相应的天数。如果提供整数表达式，必须将 *datetime-expression* 显式转换为日期。

DAYS 忽略小时、分钟和秒。

示例下面的语句返回整数值 729948：

SELECT DAYS( '1998-07-13 06:07:12' ) FROM iq\_dummy

下面的语句返回整数值 -366，这是两个日期之间的差值：

SELECT DAYS( '1998-07-13 06:07:12','1997-07-12 10:07:12' ) FROM iq\_dummy

下面的语句返回值 1999-07-14：

SELECT days(cast('2009-01-03' as date),3) ,days(date('2009-01-03'),3)from iq\_dummy

## DB\_ID 函数 [System]

返回数据库 ID 号。

语法：**DB\_ID** ( [ *database-name* ] )

**database-name** 包含数据库名的字符串表达式。如果 *database-name* 是字符串常量，则它必须用引号引起来。如果未提供 *database-name*，则返回当前数据库的 ID 号。

如果 asiqdemo 是唯一正在运行的数据库，则下面的语句返回值 0：

SELECT DB\_ID( 'asiqdemo' ), DB\_ID( ) FROM iq\_dummy

## DB\_NAME 函数 [System]

返回数据库名。

语法：**DB\_NAME** ( [ *database-id* ] )

**database-id** 数据库的 ID。*database-id* 必须是数值表达式。

示例下面的语句在对样本数据库执行时返回数据库名 asiqdemo。

SELECT DB\_NAME( 0 ) FROM iq\_dummy

## DB\_PROPERTY 函数 [System]

返回给定属性的值。

语法：**COALESCE** ( *expression*, *expression* [ , ...] )

**DB\_PROPERTY** ( { *property-id* | *property-name* }[, { *database-id* | *database-name* } ] )

**property-id** 数据库属性 ID。

**property-name** 数据库属性名称。

**database-id** 数据库 ID 号，由 DB\_ID 返回。通常情况下，使用数据库名称。

**database-name** 数据库的名称，由 DB\_NAME 返回。

下面的语句返回当前数据库的页大小（以字节为单位）。

SELECT DB\_PROPERTY( 'PAGESIZE' ) FROM iq\_dummy

## DEGREES 函数 [Numeric]

将数字从弧度转换为度数

语法：**DEGREES** ( *numeric-expression* )

参数**numeric-expression** 以弧度表示的角度。

示例下面的语句返回值 29.793805：

SELECT DEGREES( 0.52 ) FROM iq\_dummy

## DIFFERENCE 函数 [String]

比较两个字符串，评估它们之间的相似性，并返回 0 到 4 之间的一个值。最佳匹配值是 4。

**语法：DIFFERENCE** ( *string-expression1*, *string-expression2* )

参数string-expression1 要比较的第一个字符串。string-expression2 要比较的第二个字符串。

Select difference('huyong','huyong'), difference('huyong','jiayong'),difference('huyong','jiahen'),difference('huyong','jiafen') ,difference('huyong','adfcvdast') from TEST;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\7WI79~R4Y0]60(W6H3E6ARJ.jpg

## DOW 函数 [Date and time]

返回 1 到 7 之间的一个数字，表示指定日期中的星期几，星期日=1、星期一=2，依此类推。

语法：**DOW** ( *date-expression* )

SELECT DOW( '1998-07-09' ) FROM test;

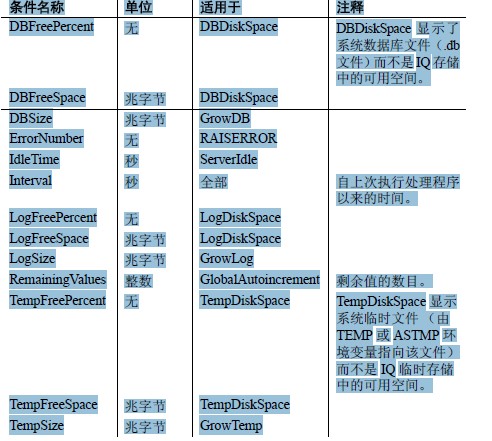
C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\H7~Q~{{I`NB76SPEEFP`VB7.jpg

## EVENT\_CONDITION 函数 [System]

指定何时触发事件处理程序。

语法**EVENT\_CONDITION** ( *condition-name* )

参数**condition-name** 触发事件的条件。可能的值已经在数据库中预设，并且不区分大小写。每个条件仅对于某些事件类型有效。



create event LogNotifier

type LogDiskSpace

where event\_condition( 'LogFreePercent' ) < 50

handler

begin

message 'LogNotifier message'

end

## EVENT\_CONDITION\_NAME 函数 [System]

可用于列出 EVENT\_CONDITION 的可能参数。

语法:**EVENT\_CONDITION\_NAME** ( *integer* )

参数**integer** 必须大于或等于零。

用法可以使用 EVENT\_CONDITION\_NAME 获得所有 EVENT\_CONDITION 参数的列表，方法是：在整数中循环，直到函数返回 NULL。

## EVENT\_PARAMETER 函数 [System]

提供事件处理程序的上下文信息。

语法：**COALESCE** ( *expression*, *expression* [ , ...] )

*context-name*:

**'ConnectionID'**

| **'User'**

| **'EventName'**

| **'Executions'**

| **'NumActive'**

| **'TableName'**

**| *condition-name***

参数**context-name** 其中一个预设字符串。这些字符串不区分大小写，它们

传递下列信息：

**• ConnectionId** 连接 ID，由以下语句返回connection\_property( 'id' )

**• User** 导致事件被触发的用户的用户 ID。

**• EventName** 已触发的事件的名称。

**• Executions** 事件处理程序执行的次数。

**• NumActive** 事件处理程序的活动实例数。如果想限制事件处理程序以便在任何给定的时间仅执行一个实例，则这很有用。

**• TableName** 表名，与 RemainingValues 一起使用。

此外，可以从 EVENT\_PARAMETER 函数访问 EVENT\_CONDITION 函数的任何有效的 *condition-name* 参数。

## EXP 函数 [Numeric]

函数返回指数函数，即 e 的指定数字次乘方。

语法**EXP** ( *numeric-expression* )

示例下面的语句返回值 3269017.372472109：

SELECT EXP( 15 ) FROM iq\_dummy

## FLOOR 函数 [Numeric]

函数返回一个数字的下限（不大于的最大整数）。

语法**FLOOR** ( *numeric-expression* )

参数**numeric-expression** 数字，通常是浮点数。

下面的语句返回值 123：

SELECT FLOOR ( 123 ) FROM iq\_dummy

示例下面的语句返回值 123.00：

SELECT FLOOR ( 123.45 ) FROM iq\_dummy

下面的语句返回值 -124.00。

SELECT FLOOR ( -123.45 ) FROM iq\_dummy

## GETDATE 函数 [Date and time]

函数返回当前的日期和时间。

语法**GETDATE** ()

示例下面的语句返回系统日期和时间。

SELECT GETDATE( ) FROM test;

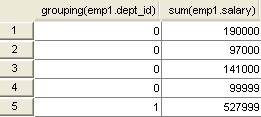
## GROUPING 函数 [Aggregate]

标识 ROLLUP 或 CUBE 操作结果集中的某一列是否由于是小计行的一部分或由于基础数据而为 NULL。

语法**GROUPING (** *group-by-expression* **)**

参数**group-by-expression** 表达式（作为分组列出现在使用带有 ROLLUP 或CUBE 关键字的 GROUP BY 子句的查询结果集中） 。该函数标识通过ROLLUP 或 CUBE 操作添加到结果集中的小计行。当前， Sybase IQ 不支持使用 PERCENTILE\_CONT 或 PERCENTILE\_DISC函数来执行 GROUP BY CUBE 操作。

select grouping(dept\_id),sum(salary)from emp1 group by cube(dept\_id);



## HEXTOBIGINT 函数 [Data type conversion]

返回十六进制字符串的 BIGINT 等效项。

语法**HEXTOBIGINT** ( *hexadecimal-string* )参数**hexadecimal-string** 要转换为大整数 (BIGINT) 的十六进制值。输入可以采用下列形式，前缀中包括小写的或大写的“0x”，或者不包括前缀：

0x*hex-string* 0X*hex-string hex-string*

示例以下语句返回值 4294967287：

SELECT HEXTOBIGINT ( '0xfffffff7' ) FROM iq\_dummy

SELECT HEXTOBIGINT ( '0Xfffffff7' ) FROM iq\_dummy

SELECT HEXTOBIGINT ( 'fffffff7' ) FROM iq\_dummy

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\T}UR2YU%(~GEW[7$[L@@%NJ.jpg

## HEXTOINT 函数 [Data type conversion]

函数返回十六进制字符串的不带符号的 BIGINT 等效项。

语法**HEXTOINT** ( *hexadecimal-string* ) 参数**hexadecimal-string** 要转换为整数的字符串。输入可以采用下列形式，前缀中包括小写的或大写的“x”，或者不包括前缀：

0x*hex-string*

0X*hex-string*

*hex-string*

示例以下语句返回值 420：

SELECT HEXTOINT ( '0x1A4' ) FROM iq\_dummy

SELECT HEXTOINT ( '0X1A4' ) FROM iq\_dummy

SELECT HEXTOINT ( '1A4' ) FROM iq\_dummy

**HEXTOINT 函数 [Data type conversion]**

函数返回十六进制字符串的不带符号的 BIGINT 等效项。

语法**HEXTOINT** ( *hexadecimal-string* ) 参数**hexadecimal-string** 要转换为整数的字符串。输入可以采用下列形式，前缀中包括小写的或大写的“x”，或者不包括前缀：

0x*hex-string*

0X*hex-string*

*hex-string*

示例以下语句返回值 420：

SELECT HEXTOINT ( '0x1A4' ) FROM iq\_dummy

SELECT HEXTOINT ( '0X1A4' ) FROM iq\_dummy

SELECT HEXTOINT ( '1A4' ) FROM iq\_dummy

**HEXTOINT** 函数 **[Data type conversion]**

函数返回十六进制字符串的不带符号的 BIGINT 等效项。

## HOUR 函数 [Date and time]

返回 0 到 23 之间的一个数字，该数字对应于指定日期/时间的小时部分。

SELECT HOUR( '1998-07-09 21:12:13' ) FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\A3`$VZGTR@AV6~B8JM2PN_A.jpg

## HOURS 函数 [Date and time]

返回自任意开始日期和时间以来的小时数，返回两个指定时间之间的完整小时数，或者将指定的 integer-expression 数量的小时添加到某个时间中。

注： 可以有3种方式，

SELECT HOURS( '1998-07-13 06:07:12' ), HOURS( '1999-07-13 06:07:12','1999-07-13 10:07:12' ),HOURS( CAST( '1999-05-12 21:05:07'AS DATETIME ), 5 ) FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\QE{M473%_88)489TE]VH{JW.jpg

## HTML\_DECODE 函数 [HTTP]

对出现在 HTML 文字字符串中的特殊字符实体进行解码。

select HTML\_DECODE('&#39;'), HTML\_DECODE('&quot;') from test;

**HTML\_ENCODE** 函数 **[HTTP]**

对要插入到 HTML 文档中的字符串中的特殊字符进行编码。

**HTTP\_ENCODE** 函数 **[HTTP]**

对与 HTTP 一起使用的字符串中的特殊字符进行编码。

**HTML\_ENCODE** ( *string* )

**HTTP\_VARIABLE** 函数 **[HTTP]**

## IFNULL 函数 [Miscellaneous]

如果第一个表达式为 NULL 值，则返回第二个表达式的值。如果第一个表达式不为 NULL，则返回第三个表达式的值。如果第一个表达式不为NULL，并且没有第三个表达式，则返回 NULL 值。

select ifnull('df','af','sdf') ,ifnull('','af','sdf') ,ifnull('af','sdf'),ifnull(null,'af','sdf') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\O~18B~C{@3V)J4L$)]B1R1I.jpg

语法**HEXTOINT** ( *hexadecimal-string* ) 参数**hexadecimal-string** 要转换为整数的字符串。输入可以采用下列形式，前缀中包括小写的或大写的“x”，或者不包括前缀：

0x*hex-string*

0X*hex-string*

*hex-string*

示例以下语句返回值 420：

SELECT HEXTOINT ( '0x1A4' ) FROM iq\_dummy

SELECT HEXTOINT ( '0X1A4' ) FROM iq\_dummy

SELECT HEXTOINT ( '1A4' ) FROM iq\_dummy

## INSERTSTR 函数 [String]

将一个字符串插入到另一个字符串中的指定位置。

**INSERTSTR** ( *numeric-expression*, *string-expression1*, *stringexpression2*

)

SELECT INSERTSTR( 0, 'jia ', 'yong' ),INSERTSTR(2,'jia','yong') FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\]1ZM)A7UYUUCS6X`}(3ZPKV.jpg

## ISDATE 函数 [Date and time]

测试字符串参数是否可以转换为日期。如果转换能够进行，该函数返回1；否则返回 0。如果参数为空，则返回 0。

SELECT iSDATE ('dfs'),ISDATE ('20090909') FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\TVGN$)@G%{C4OO0AQXQ8F8X.jpg

语法**HEXTOINT** ( *hexadecimal-string* ) 参数**hexadecimal-string** 要转换为整数的字符串。输入可以采用下列形式，前缀中包括小写的或大写的“x”，或者不包括前缀：

0x*hex-string*

0X*hex-string*

*hex-string*

示例以下语句返回值 420：

SELECT HEXTOINT ( '0x1A4' ) FROM iq\_dummy

SELECT HEXTOINT ( '0X1A4' ) FROM iq\_dummy

SELECT HEXTOINT ( '1A4' ) FROM iq\_dummy

## ISNULL 函数 [Miscellaneous]

返回参数列表中的第一个非 NULL 表达式的值。

## ISNUMERIC 函数 [Miscellaneous]

测试字符串参数是否可以转换为数字。如果转换能够进行，该函数返回1；否则返回 0。如果参数为空，则返回 0。

select case

when isnumeric(height\_in\_cms)=0

then NULL

else cast(height\_in\_cms as int)

end

from MyData

## LCASE 函数 [String]

将字符串中的所有字符转换成小写形式。

SELECT LCASE ('AFDA'),lower('AFDA') FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\(V%Y6TV$ZZXE@`F8]OI9AZS.jpg

## LEFT 函数 [String]

从字符串开头返回指定数量的字符。

SELECT LEFT( 'chocolate', 5 ) FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\GO$Z0A8GQW]8O9%1ZTDX5PL.jpg

## LEN 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入并返回由数据库的归类序列定义的、指定字符串表达式的字符数（不包括尾随空白）。结果可能不同于多字节字符集的字符串字节长度。此外，也允许使用 BINARY 和 VARBINARY，在这种情况下， LEN() 返回输入的字节数。LEN 是 LENGTH 函数的别名

select len('aaa'),length('aaa') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\{8SA2`(ZH)AZ61N71QRJOUC.jpg

## LN 函数 [Numeric]

返回指定表达式的自然对数。

## LOCATE 函数 [String]

返回一个字符串在另一个字符串中的位置。

Select locate('hujiayong1985','jia'), locate('hujiayong1985','jia',1),locate('hujiayong1985','jia',0),locate('hujiayong1985','jia',-3) from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\H[H9__)$`)2UD{IM4YBZ}DO.jpg

## LOWER 函数 [String]

将字符串中的所有字符转换成小写形式。

## LTRIM 函数 [String]

删除字符串中的前导空白

## MINUTE 函数 [Date and time]

返回 0 到 59 之间的一个数字，该数字对应于指定日期/时间值的分钟部分。

## MINUTES 函数 [Date and time]

返回自任意开始日期和时间以来的分钟数，返回两个指定时间之间的完整分钟数，或者将指定的 integer-expression 数量的分钟添加到某个时间中。

## MOD 函数 [Numeric]

返回一个整数除以另一个整数之后产生的余数。

SELECT MOD( 5, 3 ) FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\HJ%VVCLWVB3O7$~VH_Q6Y{2.jpg

## MONTH 函数 [Date and time]

返回 1 到 12 之间的一个数字，该数字对应于给定日期的月份。

## MONTHNAME 函数 [Date and time]

返回指定日期表达式中的月份名称。

SELECT MONTHNAME( '1998-09-05' ) FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~A2[(YJ51Y9O[1Y(2Q1U@7F.jpg

## PATINDEX 函数 [String]

返回指定模式在第一次出现时的起始位置。**PATINDEX** ( **'%***pattern***%'**, *string-expression* )

\_ （下划线） 任意一个字符

% （百分号） 包含零个或多个字符的任意字符串

[] 指定范围或集合中的任意单个字符

[^] 不在指定范围或集合中的任意单个字符

SELECT PATINDEX( '%hoco%', 'chocolate' ) FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\T}5XO`YQC[UN{ZG72U4WB{U.jpg

## PERCENTILE\_CONT 函数 [Analytical]

没搞懂~~~

**PERCENTILE\_DISC 函数 [Analytical]**

**没搞懂**

## POWER 函数 [Numeric]

以一个数字为底数另一个数字为指数计算乘方值。

## PROPERTY 函数 [System]

以字符串形式返回指定的服务器级属性值。

SELECT PROPERTY( 'Name' ) FROM emp1;

## QUARTER 函数 [Date and time]

返回一个表示所提供的日期表达式中年的季度的数字。

## RADIANS 函数 [Numeric]

将数字由角度转换成弧度。

SELECT RADIANS( 30 ) FROM test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\5%$VRQKLY9B`LFNZ3_9XRKP.jpg

## RAND 函数 [Numeric]

返回一个双精度的随机数字 x，其中 0 <= x <1，使用可选种子。

## REPEAT 函数 [String]

以指定的次数连接字符串。

下面的语句返回值“repeatrepeatrepeat”：

SELECT REPEAT( 'repeat', 3 ) FROM test;

## REPLICATE 函数 [String]

以指定的次数连接字符串。

select replicate('ab',2) from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\ZL91{N_A4BX9]}KY0OG1A05.jpg

## RIGHT 函数 [String]

返回字符串中最右边的字符。

Select right(‘acb123’,3) from test ;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\ZZPTCZF@K}]364MK[SWTD1A.jpg

## ROUND 函数 [Numeric]

将 *numeric-expression* 舍入到小数点后指定的 *integer-expression* 位数。

select round(12.3123,2) from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\Z05M9G{43V[[PW0@X@}H(]H.jpg

## ROWID 函数 [Miscellaneous]

返回表的每一行的内部行 ID 值。

select rowid("emp1"),\* from emp1 where rowid(emp1)<4;



DELETE FROM emp1 WHERE ROWID ( emp1 ) > 50

## RTRIM 函数 [String]

返回尾随空白已删除的字符串。

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## SIMILAR 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg

## REVERSE 函数 [String]

采用一个参数作为 BINARY 或 STRING 类型的输入，并返回字符以反向顺序列出的指定字符串。

select reverse('abcd') from test;

C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~0VU2@6UX(S`82DJQ9S9U2E.jpg