

QBFunctions

Property Summary

QBSelect	parent Get query builder parent table clause, this may be a query or a join clause.
QBSelect	root Get query builder parent.

Method Summary

QBFunction	abs (value) Create abs(column) expression
QBFunction	bit_length (value) Create bit_length(column) expression
QBFunction	cast (value, type) Cast using type name.
QBFunction	ceil (arg) Create ceil(column) expression
QBFunction	coalesce (args) Create year(date) expression
QBFunction	concat (arg1, arg2) Create concat(args, arg2) expression
QBFunction	day (arg) Create day(date) expression
QBFunction	divide (arg1, arg2) Create divide(args, arg2) expression
QBFunction	floor (arg) Create floor(column) expression
QBFunction	hour (arg) Create hour(date) expression
QBFunction	len (value) Create length(column) expression
QBFunction	locate (string1, string2) Create locate(str1, str2) expression
QBFunction	locate (string1, string2, start) Create locate(str1, str2, start) expression
QBFunction	lower (value) Create lower(column) expression
QBFunction	minus (arg1, arg2) Create minus(args, arg2) expression
QBFunction	minute (arg) Create minute(date) expression
QBFunction	mod (dividend, divisor) Create mod(dividend, divisor) expression
QBFunction	month (arg) Create month(date) expression
QBFunction	multiply (arg1, arg2) Create multiply(args, arg2) expression
QBFunction	nullif (arg1) Create nullif(arg1, arg2) expression
QBFunction	plus (arg1, arg2) Create plus(args, arg2) expression
QBFunction	round (arg) Create round(column) expression
QBFunction	second (arg) Create second(date) expression
QBFunction	sqrt (value) Create sqrt(column) expression
QBFunction	substring (arg, pos) Create substring(column, pos) expression
QBFunction	substring (arg, pos, len) Create substring(column, pos, len) expression
QBFunction	trim (value) Create trim(column) expression
QBFunction	upper (value) Create upper(column) expression

[QBFunction](#) [year](#)(arg)
Create year(date) expression

Property Details

parent

Get query builder parent table clause, this may be a query or a join clause.

Returns

[QBSelect](#)

Sample

```
/** @type {QBSelect<db:/example_data/person>} */
var query = databaseManager.createSelect('db:/example_data/person')
query.where.add(query.joins.person_to_parent.joins.person_to_parent.columns.name.eq('john'))
foundset.loadRecords(query)
```

root

Get query builder parent.

Returns

[QBSelect](#)

Sample

```
/** @type {QBSelect<db:/example_data/order_details>} */
var subquery = databaseManager.createSelect('db:/example_data/order_details')

/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders')
query.where.add(query
    .or
        .add(query.columns.order_id.not.isin([1, 2, 3]))
        .add(query.exists(
            subquery.where.add(subquery.columns.orderid.eq(query.columns.
order_id)).root
        ))
    )
foundset.loadRecords(query)
```

Method Details

abs

[QBFunction](#) [abs](#)(value)
Create abs(column) expression

Parameters

[Object](#) value

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') // $NON-NLS-1$
query.where.add(query.columns.mynum.abs.eq(query.functions.abs(myval)))
foundset.loadRecords(query);
```

bit_length**QBFunction** **bit_length** (value)

Create bit_length(column) expression

Parameters

{Object} value

Returns**QBFunction****Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.shipname.bit_length.eq(query.functions.bit_length('Sample')));
foundset.loadRecords(query);
```

cast**QBFunction** **cast** (value, type)

Cast using type name.

Parameters

{Object} value - object to cast

{String} type - type see QUERY_COLUMN_TYPES

Returns**QBFunction****Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.result.add(query.functions.cast("22", QUERY_COLUMN_TYPES.TYPE_INTEGER)).add(query.columns.amt_discount.
cast(QUERY_COLUMN_TYPES.TYPE_STRING));
application.output(databaseManager.getDataSetByQuery(query,1).getAsHTML())
```

ceil**QBFunction** **ceil** (arg)

Create ceil(column) expression

Parameters

{Object} arg - number object

Returns**QBFunction****Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mynumcol.ceil.eq(query.functions.ceil(myvar)));
foundset.loadRecords(query);
```

coalesce**QBFunction** **coalesce** (args)

Create year(date) expression

Parameters

{Object...} args - arguments to coalesce

Returns**QBFunction**

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mycol.coalesce('defval').eq(query.functions.coalesce(myvar, 'defval'))
foundset.loadRecords(query);
```

concat

[QBFunction](#) **concat** (arg1, arg2)

Create concat(args, arg2) expression

Parameters

[Object](#) arg1
[Object](#) arg2

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/udm/contacts>} */
var query = databaseManager.createSelect('db:/udm/contacts') //NON-NLS-1$
query.result.add(query.columns.name_first.concat(' ').concat(query.columns.name_last))
var ds = databaseManager.getDataSetByQuery(query, -1)
```

day

[QBFunction](#) **day** (arg)

Create day(date) expression

Parameters

[Object](#) arg - date object

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mydatecol.day.eq(query.functions.day(mydatevar))
foundset.loadRecords(query);
```

divide

[QBFunction](#) **divide** (arg1, arg2)

Create divide(args, arg2) expression

Parameters

[Object](#) arg1
[Object](#) arg2

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mynumcol.divide(2).eq(query.functions.divide(myvar, 2))
foundset.loadRecords(query);
```

floor

[QBFunction](#) **floor** (arg)

Create floor(column) expression

Parameters

[Object](#) arg - number object

Returns[QBFunction](#)**Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mynumcol.floor.eq(query.functions.floor(myvar)))
foundset.loadRecords(query);
```

hour[QBFunction](#) **hour** (arg)

Create hour(date) expression

Parameters[Object](#) arg - date object**Returns**[QBFunction](#)**Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mydatecol.hour.eq(query.functions.hour(mydatevar)))
foundset.loadRecords(query);
```

len[QBFunction](#) **len** (value)

Create length(column) expression

Parameters[Object](#) value**Returns**[QBFunction](#)**Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.shipname.len.eq(query.functions.len('Sample')))
foundset.loadRecords(query);
```

locate[QBFunction](#) **locate** (string1, string2)

Create locate(str1, str2) expression

Parameters[Object](#) string1 - string to locate[Object](#) string2 - string to search in**Returns**[QBFunction](#)**Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.shipname.locate('amp').eq(query.functions.locate('Sample', 'amp')))
foundset.loadRecords(query);
```

locate[QBFunction](#) **locate** (string1, string2, start)

Create locate(str1, str2, start) expression

Parameters

{Object} string1 - string to locate
 {Object} string2 - string to search in
 {Number} start - start pos

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //$NON-NLS-1$
query.where.add(query.columns.shipname.locate('amp', 1).eq(query.functions.locate('Sample', 'amp', 1)))
foundset.loadRecords(query);
```

lower

[QBFunction](#) **lower** (value)

Create lower(column) expression

Parameters

{Object} value

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //$NON-NLS-1$
query.where.add(query.columns.shipname.lower.eq(query.functions.lower('Sample')))
foundset.loadRecords(query);
```

minus

[QBFunction](#) **minus** (arg1, arg2)

Create minus(args, arg2) expression

Parameters

{Object} arg1
 {Object} arg2

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //$NON-NLS-1$
query.where.add(query.columns.mynumcol.minus(2).eq(query.functions.minus(myvar, 2)))
foundset.loadRecords(query);
```

minute

[QBFunction](#) **minute** (arg)

Create minute(date) expression

Parameters

{Object} arg - date object

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //$NON-NLS-1$
query.where.add(query.columns.mydatecol.minute.eq(query.functions.minute(mydatevar)))
foundset.loadRecords(query);
```

mod

[QBFunction](#) **mod** (dividend, divisor)

Create mod(dividend, divisor) expression

Parameters

[Object](#) dividend
[Object](#) divisor

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */  
var query = databaseManager.createSelect('db:/example_data/orders') //$NON-NLS-1$  
query.where.add(query.columns.mynumcol.mod(2).eq(query.functions.mod(myvar, 2))  
foundset.loadRecords(query);
```

month

[QBFunction](#) **month** (arg)

Create month(date) expression

Parameters

[Object](#) arg - date object

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */  
var query = databaseManager.createSelect('db:/example_data/orders') //$NON-NLS-1$  
query.where.add(query.columns.mydatecol.month.eq(query.functions.month(mydatevar))  
foundset.loadRecords(query);
```

multiply

[QBFunction](#) **multiply** (arg1, arg2)

Create multiply(args, arg2) expression

Parameters

[Object](#) arg1
[Object](#) arg2

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */  
var query = databaseManager.createSelect('db:/example_data/orders') //$NON-NLS-1$  
query.where.add(query.columns.mynumcol.multiply(2).eq(query.functions.multiply(myvar, 2))  
foundset.loadRecords(query);
```

nullif

[QBFunction](#) **nullif** (arg1)

Create nullif(arg1, arg2) expression

Parameters

[Object](#) arg1

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.shipname.nullif('none').eq(query.functions.nullif('Sample', 'none')))
foundset.loadRecords(query);
```

plus

[QBFunction](#) **plus** (arg1, arg2)

Create plus(args, arg2) expression

Parameters

[Object](#) arg1
[Object](#) arg2

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mynumcol.plus(2).eq(query.functions.plus(myvar, 2)))
foundset.loadRecords(query);
```

round

[QBFunction](#) **round** (arg)

Create round(column) expression

Parameters

[Object](#) arg - number object

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mynumcol.round.eq(query.functions.round(myvar)))
foundset.loadRecords(query);
```

second

[QBFunction](#) **second** (arg)

Create second(date) expression

Parameters

[Object](#) arg - date object

Returns

[QBFunction](#)

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mydatecol.second.eq(query.functions.second(mydatevar)))
foundset.loadRecords(query);
```

sqrt

[QBFunction](#) **sqrt** (value)

Create sqrt(column) expression

Parameters

[Object](#) value

Returns[QBFunction](#)**Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mynum.sqrt.eq(query.functions.sqrt(myval)))
foundset.loadRecords(query);
```

substring[QBFunction](#) **substring** (arg, pos)

Create substring(column, pos) expression

Parameters

[Object](#) arg - column name
[Number](#) pos - position

Returns[QBFunction](#)**Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.shipname.substring(3).eq(query.functions.substring('Sample', 3)))
foundset.loadRecords(query);
```

substring[QBFunction](#) **substring** (arg, pos, len)

Create substring(column, pos, len) expression

Parameters

[Object](#) arg - column name
[Number](#) pos - position
[Number](#) len - length

Returns[QBFunction](#)**Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.shipname.substring(3, 2).eq(query.functions.substring('Sample', 3, 2)))
foundset.loadRecords(query);
```

trim[QBFunction](#) **trim** (value)

Create trim(column) expression

Parameters[Object](#) value**Returns**[QBFunction](#)**Sample**

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.shipname.trim.eq(query.functions.trim('Sample')))
foundset.loadRecords(query);
```

upper[QBFunction](#) **upper** (value)

Create upper(column) expression

Parameters

{Object} value

Returns

QBFunction

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.shipname.upper.eq(query.functions.upper('Sample')))
foundset.loadRecords(query);
```

year

QBFunction year (arg)

Create year(date) expression

Parameters

{Object} arg - date object

Returns

QBFunction

Sample

```
/** @type {QBSelect<db:/example_data/orders>} */
var query = databaseManager.createSelect('db:/example_data/orders') //NON-NLS-1$
query.where.add(query.columns.mydatecol.year.eq(query.functions.year(mydatevar))
foundset.loadRecords(query);
```