Toward ADQL 2.2

Grégory Mantelet





Reformat list of functions

#107

ADQL 2.1

2.3 Mathematical and Trigonometrical Functions

ADQL declares a list of reserved keywords (see Section 2.1.2) which include the mathematical and trigonometrical function names. Their syntax, usage and description are detailed in tables 1 and 2.

Name	Argument datatype	Return datatype	Description
abs(x)	*x* double	double	Returns the absolute value of *x*.
ceiling(x)	*x* double	double	Returns the smallest integer that is not less than *x*.
degrees(x)	*x* double	double	Converts the angle *x* from radians to degrees.
exp(x)	*x* double	double	Returns Euler's number *e* raised to the power of *x*.
floor(x)	*x* double	double	Returns the largest integer that is not greater than *x*.
log(x)	*x* double	double	Returns the natural logarithm (base *e*) of *x*. The value of *x* must be greater than zero.
log10(x)	*x* double	double	Returns the base 10 logarithm of *x*. The value of *x* must be greater than zero.
mod(x,y)	*x* double, *y* double	double	Returns the remainder *r* of *x*/*y* as a double, where: • *r* has the same sign as *x* • - *r* - is less than - *y* - • *x* = (*f* * *y*) + *r* for a given integer *f*
pi()		double	The numeric constant π.
power(x,y)	*x* double, *y* double	double	Returns the value of *x* raised to the power of *y*.
radians(x)	*x* double	double	Converts the angle *x* from degrees to radians.
sqrt(x)	*x* double	double	Returns the positive square root of *x*.
rand(x)	*x* double	double	Returns a random value between 0.0 and 1.0. The optional argument, *x*, originally intended to provide a random seed, has undefined semantics. Query writers are advised to omit this argument.
round(x,n)	*x* double, *n* integer	double	Rounds *x* to *n* decimal places. The integer *n* is optional and defaults to 0 if not specified. A negative value of *n* will round to the left of the decimal point.
truncate(x, n)	*x* double *n* integer	double	Truncates *x* to *n* decimal places. The integer *n* is optional and defaults to 0 if not specified.

Table 1: Mathematical functions

ADQL 2.2

2.3 Mathematical and Trigonometrical Functions

ADQL declares a list of reserved keywords (see Section 2.1.2) which include the mathematical and trigonometrical function names. Their syntax, usage and description are detailed in below.

2.3.1 Mathematical Functions

`abs(double x) → double`
Returns the absolute value of x.

`ceiling(double x) → double`
Returns the smallest integer that is not less than *x*.

`degrees(double x) → double`
Converts the angle *x* from radians to degrees.

`exp(double x) → double`
Returns Euler's number *e* raised to the power of *x*.

`floor(double x) → double`
Returns the largest integer that is not greater than *x*.

`log(double x) → double`
Returns the natural logarithm (base *e*) of *x*. The value of *x* must be greater than zero.

`log10(double x) → double`
Returns the base 10 logarithm of *x*. The value of *x* must be greater than zero.

`mod(double x, double y) → double`
Returns the remainder *r* of *x*/*y* as a double, where:
- *r* has the same sign as *x*
- |*r*| is less than |*y*|
- *x* = (*f* * *y*) + *r* for a given integer *f*

`pi() → double`
The numeric constant π.

`power(double x, double y) → double`
Returns the value of *x* raised to the power of *y*.

`radians(double x) → double`
Converts the angle *x* from degrees to radians.

`sqrt(double x) → double`
Returns the positive square root of *x*.

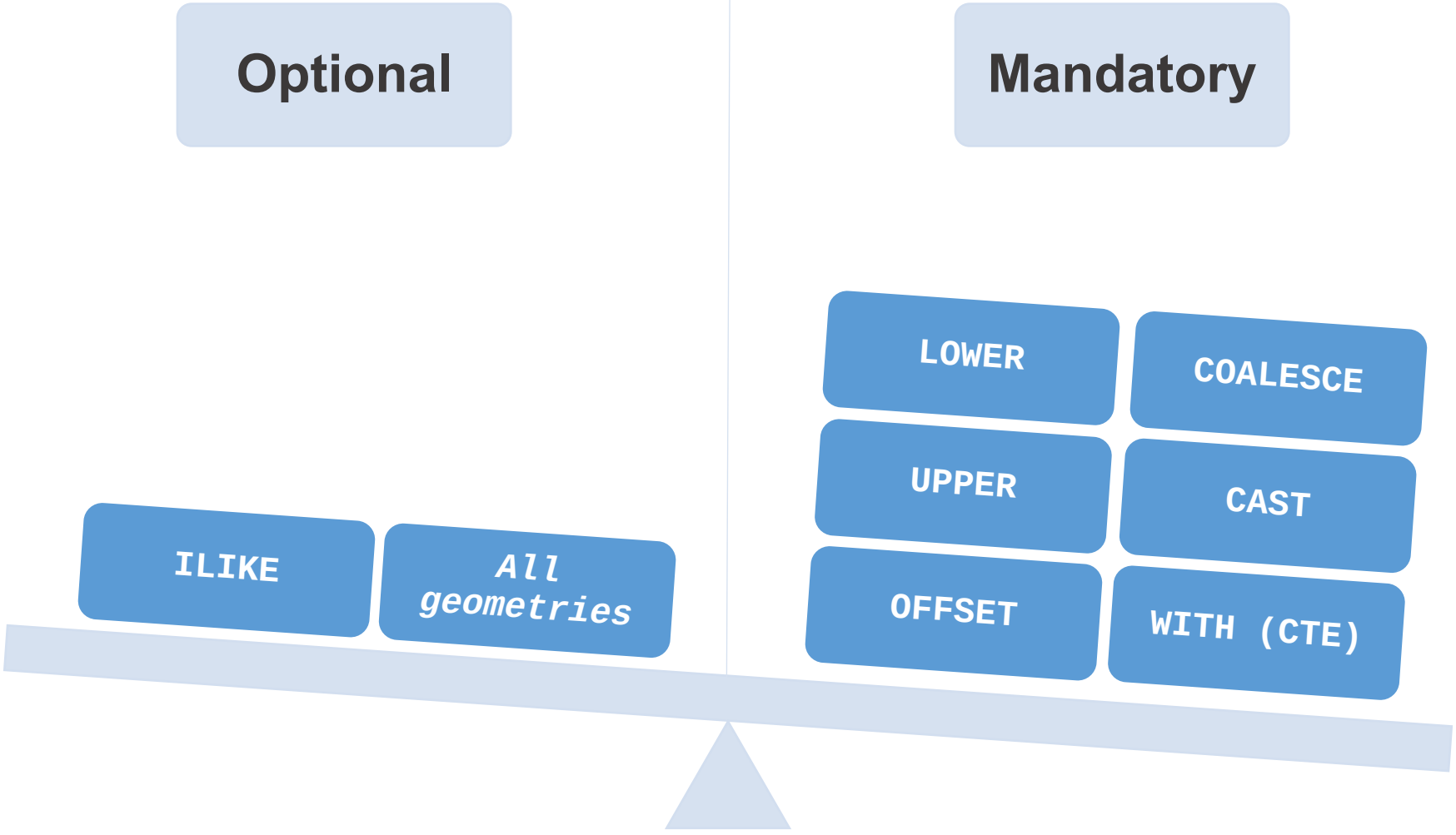
`rand(double x) → double`
Returns a random value between 0.0 and 1.0. The optional argument, *x*, originally intended to provide a random seed, has undefined semantics. Query writers are advised to omit this argument.

`round(double x, integer n) → double`
Rounds *x* to *n* decimal places. The integer *n* is optional and defaults to 0 if not specified. A negative value of *n* will round to the left of the decimal point.

`truncate(double x, integer n) → double`
Truncates *x* to *n* decimal places. The integer *n* is optional and defaults to 0 if not specified.

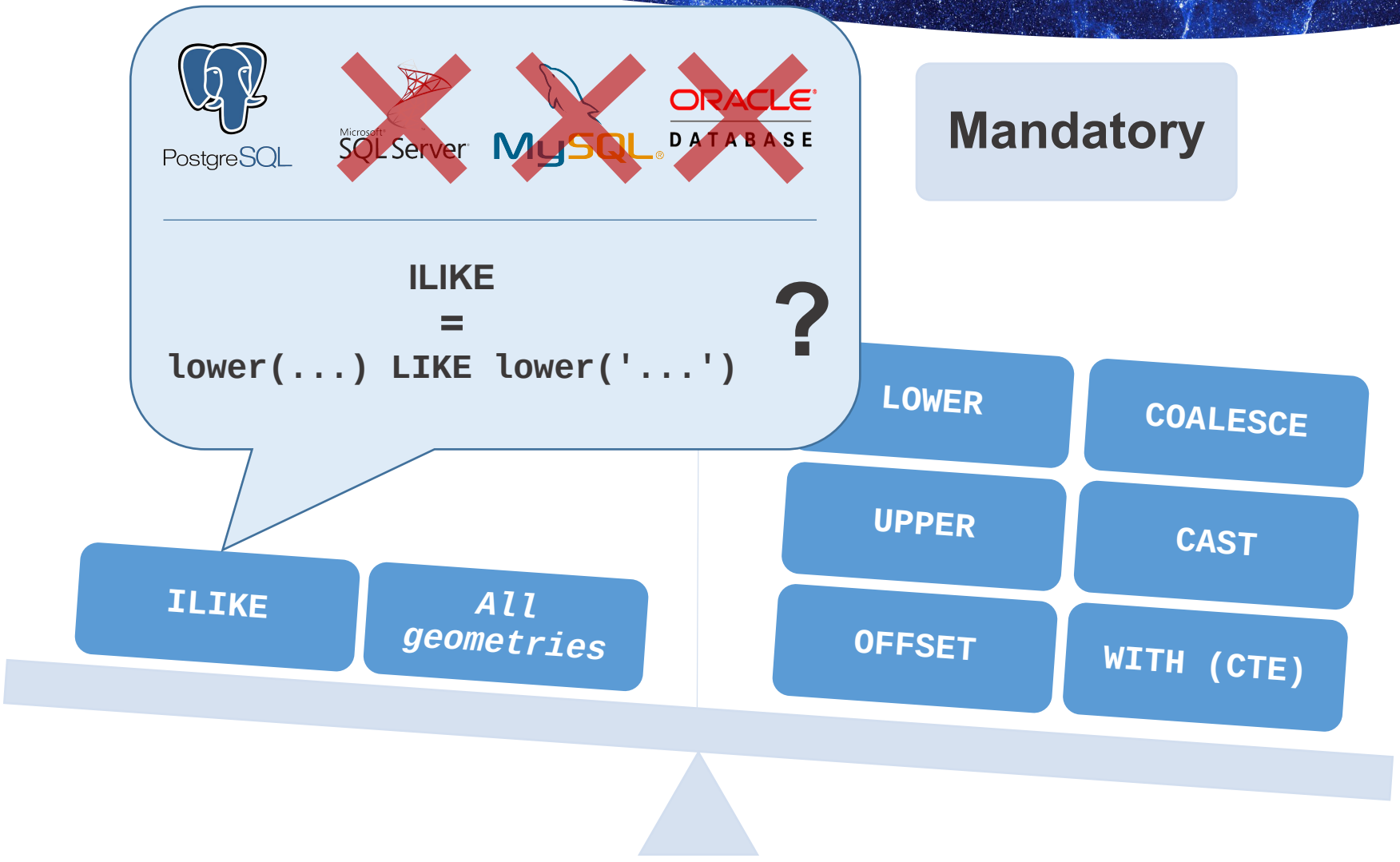
Less optional features

#108



Less optional features

#108



Remove deprecated features?

```
DELETE FROM "ADQL-2.2"
 WHERE deprecated = true;
```
?

- CooSys argument (from `POINT`, `CIRCLE`, ...)
- Implicit conversion (in `DISTANCE`, `INTERSECTS` and `CONTAINS`)
- `COOSYS(...)`
- `BOX(...)`

REGION uses DALI syntax

#109

Type REGION

```
> SELECT my_region
     my_region
- - - - - - - - - - - -
CIRCLE 1.0 2.0 0.33
```

- - Use a UDF for STC-s:

```
> SELECT
region_to_stcs('CIRCLE
1.0 2.0 0.33')
          stcs
- - - - - - - - - - - - - -
Circle ICRS 1.0 2.0 0.33
```

Function REGION()

- - Syntax: `shape` serialization

```
> REGION('CIRCLE 1.0 2.0
0.33')
```

- - Syntax: STC-s MAY be supported

```
> REGION('Circle ICRS
1.0 2.0 0.33')
```

- - Prefer using a UDF:

```
> stcs_to_region('Circle
ICRS 1.0 2.0 0.33')
```

To be defined in UDF Catalogue

Still a lot to do

1. Interval #44
2. Array/Vector
3. MOC #9
4. Enumerations #104
5. INTERSECTION #52

6. Binary data
7. Boolean data

ADQL-v2.3 ?