

CSCI 403 Database Management

4 – SQL Functions and Operators

CS@Mines

BOOLEAN(ISH) OPERATORS

CS@Mines

2

Comparison Operators

Comparison operators:

Operator	Meaning
=	equals
<> or !=	not equals
<	less than
>	greater than
<=	less than or equal
>=	greater than or equal
BETWEEN	a BETWEEN x AND y ≡ a >= x AND a <= y

CS@Mines

3

Logical Operators

AND, OR, NOT

SQL has three-valued logic because of NULL. Here are the truth tables:

a	b	a AND b	a OR b
true	true	true	true
true	false	false	true
true	NULL	NULL	true
false	false	false	false
false	NULL	false	NULL
NULL	NULL	NULL	NULL

a	NOT a
true	false
false	true
NULL	NULL

If you think of NULL as meaning "unknown", or "cannot be determined", then these truth tables are pretty easy to reproduce.

CS@Mines

4

Operators To Use With NULL

NULL cannot be (usefully) compared:

a = NULL gives NULL

NULL <> b gives NULL

NULL = NULL gives NULL!

Instead, use IS NULL or IS NOT NULL:

a IS NULL

b IS NOT NULL

Also available: IS [NOT] DISTINCT FROM

Acts like <> or =, except treating NULL as a distinct value.

CS@Mines

5

String Pattern Matching

string [NOT] LIKE *pattern*

pattern is a string literal containing:

Regular characters – these just match themselves

% - matches zero or more of any character

_ - matches exactly one of any character

\% - matches %

_ - matches _

e.g.

'foobar' LIKE 'foo%' → true

'foobar' LIKE '%bar' → true

'foobar' LIKE 'f%r' → true

'foobar' LIKE 'foo_ar' → true

'foobar' LIKE 'foobar' → true

PostgreSQL also supports POSIX regular expression matching (this is not standard SQL):

<https://www.postgresql.org/docs/9.5/static/functions-matching.html#FUNCTIONS-POSIX-REGEXP>

CS@Mines

6

Usage

- The above operators are most used in WHERE


```
SELECT * FROM mines_cs_faculty
WHERE office IS NOT NULL;

SELECT * FROM mines_cs_faculty
WHERE name LIKE '%Christopher';
```
- Can also use in other clauses if t/f value is desired:


```
SELECT 'foobar' LIKE 'f_o_a_';

?column?
-----
t
(1 row)
```

CS@Mines

7

OTHER FUNCTIONS AND OPERATORS

CS@Mines

8

Constants

Not really a SQL function, but any literal acts like a constant function:

```
SELECT name, 'yay' FROM mines_cs_faculty;
```

name	?column?
Han, Qi	yay
Painter-Wakefield, Christopher	yay
Paone, Jeffrey	yay
Wang, Hua	yay
Yang, Dejun	yay
Yue, Chuan	yay
Camp, Tracy	yay
Dantam, Neil	yay
Fisher, Wendy	yay
...	

CS@Mines

9

Math

A small sampling of math operators/functions:

Operator	Meaning
+, -, *, /	Addition, etc.
%	Modulo
^	Exponentiation
!	Factorial
	Square root

Function	Meaning
ceil(n)	Ceiling function
log(n)	Base 10 logarithm
pi()	π
sin(n)	Sine function
round(n,s)	Round to s places

See

<https://www.postgresql.org/docs/9.5/static/functions-math.html>

for a complete list!

CS@Mines

10

Strings

Lots to choose from here, too. A sampling:

Operator or Function	Meaning	Examples
	concatenation	'foo' 'bar' → 'foobar' 'answer:' 42 → 'answer:42'
lower(s)	convert to all lowercase	lower('SQL Yay') → 'sql yay'
substring(s from x for y)	substring	substring('hello' from 4) → 'lo'
rtrim(s, chars)	remove chars from right	rtrim('joe2', '123') → 'joe'
Translate(s, from, to)	replace characters	translate('hello', 'eo', 'au') → 'hallu'

See

<https://www.postgresql.org/docs/9.5/static/functions-string.html>

CS@Mines

11

...and More

- Date/Time functions:
 - extract() – extract hour, minute, etc.
 - date_trunc() – truncate to start of day, hour, etc.
 - current_date, current_time, now() – get current date/time/timestamp
- Geometric functions
- XML, JSON functions
- coalesce(), case(), ...

CS@Mines

12

Usage

- Pretty much anywhere...

```
SELECT lastname || ', ' || firstname  
FROM person;
```

```
SELECT * FROM sales  
WHERE amt + tax > 100.0;
```

```
UPDATE employee SET salary = salary * 1.04;
```

```
SELECT * FROM foo  
GROUP BY substring(bar from 1 for 3);
```

Up Next

- Next lecture:

Intro to SQL: Schemas and aliasing, joins.