SQL Conditional Expressions:
Part 2

BETWEEN

Logical SQL operator with no direct analogue in Python

Operates on a column using given upper and lower bound

Evaluates as TRUE if column's value is between the two bounds

Bounds are **inclusive**

Python equivalent:

```python
def BETWEEN(value, lower, upper):
    return value >= lower and value <= upper
```

BETWEEN

id [PK] integer	title text	author text	genre text	year integer
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

```sql
SELECT b.title as twentieth_century_books
FROM books b
WHERE b.year BETWEEN 1900 AND 2000;
```

	twentieth_century_books text
1	The Great Gatsby
2	Big Fish

BETWEEN

	id [PK] integer	title text	author text	genre text	year integer
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

Question: What is the bug here?

```
SELECT b.title as twentieth_century_books
FROM books b
WHERE b.year BETWEEN 1900 AND 2000;
```

BETWEEN

	id [PK] integer ✏	title text ✏	author text ✏	genre text ✏	year integer ✏
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

Answer: the upper bound includes the year 2000

```
SELECT b.title as twentieth_century_books
FROM books b
WHERE b.year BETWEEN 1900 AND 2000;
```

id [PK] integer	title text	author text	genre text	year integer
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

Answer: the upper bound includes the year 2000

```sql
SELECT b.title as twentieth_century_books
FROM books b
WHERE b.year BETWEEN 1900 AND 1999;
```

LIKE

Evaluates as TRUE if string is contained in set of strings represented by given pattern

Character	Meaning
%	Matches any sequence of zero or more characters
_	Matches any single character

Value		Pattern	Result
'Big Fish'	LIKE	'Big Fish'	TRUE
'Big Fish'	LIKE	'Big%'	TRUE
'Big Fish'	LIKE	'B_g F_sh'	TRUE
'Big Fish'	LIKE	'Big_Fish'	TRUE
'Big Fish'	LIKE	'Big_Fish_'	FALSE
'Big Fish'	LIKE	'Small %'	FALSE
'Big Fish'	LIKE	'Big%Dog'	FALSE

LIKE and ILIKE

ILIKE uses a case-insensitive match

id [PK] integer	title text	author text	genre text	year integer
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

"Book titles that begin with T"

```
SELECT b.title FROM books b
WHERE b.title LIKE 'T%';
```

LIKE and ILIKE

ILIKE uses a case-insensitive match

id [PK] integer	title text	author text	genre text	year integer
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

"Book titles that begin with T"

```
SELECT b.title FROM books b
WHERE b.title LIKE 'T%';
```

title text
The Great Gatsby

1 The Great Gatsby

LIKE and ILIKE

ILIKE uses a case-insensitive match

id [PK] integer	title text	author text	genre text	year integer
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

"Book titles that contain T"

```
SELECT b.title FROM books b
WHERE b.title LIKE '%T%';
```

LIKE and ILIKE

ILIKE uses a case-insensitive match

id [PK] integer	title text	author text	genre text	year integer
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

"Book titles that contain T"

```sql
SELECT b.title FROM books b
WHERE b.title LIKE '%T%';
```

title text	
1	The Great Gatsby

LIKE and ILIKE

ILIKE uses a case-insensitive match

id [PK] integer	title text	author text	genre text	year integer
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

"Book titles that contain T or t"

```sql
SELECT b.title FROM books b
WHERE b.title ILIKE '%T%';
```

LIKE and ILIKE

ILIKE uses a case-insensitive match

id [PK] integer	title text	author text	genre text	year integer
1	Frankenstein	Mary Shelley	Novel	1818
2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	Big Fish	Daniel Wallace	Magical Realism	1998
4	Don Quixote	Miguel de Cervantes	Novel	1605

"Book titles that contain T or t"

```
SELECT b.title FROM books b
WHERE b.title ILIKE '%T%';
```

title text
1 Frankenstein
2 The Great Gatsby
3 Don Quixote

IN

Evaluates as TRUE if list of
options contains specified value

Similar to **in** keyword in Python:

```python
def IN(value, options):
    return value in options
```

```python
options = [1, 2, 3]
1 in options          # True
4 in options          # False
```

SQL with Python

nucamp

id [PK] integer	title text	author text	genre text	year integer
1	1 Frankenstein	Mary Shelley	Novel	1818
2	2 The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3 Big Fish	Daniel Wallace	Magical Realism	1998
4	4 Don Quixote	Miguel de Cervantes	Novel	1605

```
SELECT b.title, b.year FROM books b
WHERE b.year IN (1990, 1986, 1996, 1998);
```

id [PK] integer ✏	title text ✏	author text ✏	genre text ✏	year integer ✏
1	1	Frankenstein	Mary Shelley	Novel	1818
2	2	The Great Gatsby	F. Scott Fitzgerald	Novel	1925
3	3	Big Fish	Daniel Wallace	Magical Realism	1998
4	4	Don Quixote	Miguel de Cervantes	Novel	1605

```
SELECT b.title, b.year FROM books b
WHERE b.year IN (1990, 1986, 1996, 1998);
```

title text 🔒	year integer 🔒
1	Big Fish	1998