Transact-SQL & simple queries

UNIT 4. Part 2

Objetives

- Go on the SELECT sentence study
- How to filter rows

The TOP clause

[TOP <expression> [PERCENT] [WITH TIES]]

- Specifies that only the first set of rows will be returned from the query result.
- The set of rows can be either a number or a percentage of rows of the query result.
- WITH TIES specifies that additional rows will be returned from the base result set with the same value as the last of the TOP *n* rows.

```
SELECT TOP 3 WITH TIES *
FROM employees ORDER BY sales;

SELECT TOP 10 ofnb, city, sales
FROM offices ORDER BY sales;
```

The DISTINCT/ALL clause

- Specifies that only unique rows can appear in the result set. Null values are considered equal for the purposes of the DISTINCT keyword.
- Including DISTINCT in the SELECT, duplicated rows are eliminated so that only one row remains.
- ALL is the default value, all rows are displayed.
- DISTINCT se aplica a la fila entera, no a la 1ª columna.

```
SELECT DISTINCT dir
FROM offices;

SELECT DISTINCT dir, zone
FROM offices;
```

The WHERE clause

Selecting rows

WHERE < search_condition >

- Specifies the search condition for the rows returned by the query.
- The search condition is a combination of one or more predicates that use the logical operators AND, OR, and NOT.
- There is no limit to the number of predicates that can be included in a search condition.
- Strongly recommended: use of () when combining predicates with AND and OR operators in the same search condition.

The WHERE clause

- Predicates. In SQL we have 5 predicates (expression that returns TRUE, FALSE, or UNKNOWN):
  - Standard comparison
  - BETWEEN test
  - IN test
  - IS NULL test
  - LIKE test

Predicates

Standard Comparison

<expression>
{=
|>
|<
|<> |!=
|>= |!<
|<= |!>
} <expression>

Ex: sales <> 0

Predicates

BETWEEN test

<expression> [NOT] BETWEEN

<expression2> AND <expression3>

EX:

sales BETWEEN 10000 AND 20000

sales NOT BETWEEN 10000 AND 20000

Predicates

IN test

<expression> [NOT] IN (<exp_value> [,...*n*])

Ex:

Office IN (2,4,8,6,9)

Office NOT IN (1,3,6)

If office is null NOT IN returns UNKNOWN so the office does not appear in the result.

Predicates

IS NULL

<expression> IS [NOT] NULL

Ex:

Office IS NULL

Office IS NOT NULL

Predicates

LIKE test

<expression> [NOT] LIKE <pattern> [ESCAPE 'char']

Determines whether a specific character string (expression) matches a specified pattern.

% : Any string of zero or more characters.

_ : Any single character.

[ ] Any single character within the specified range ([a-f]) or set ([abcdef]).

[^] Any single character not within the specified range

ESCAPE Is a character that is put in front of a wildcard character to indicate that the wildcard should be interpreted as a regular character and not as a wildcard.

Predicates

LIKE test

Examples:

Name LIKE ‘a%’

Name LIKE ‘a[_]%’

Name LIKE ‘%[a-f]’

Name LIKE ‘%[a,c,r]’

Combining predicates

NOT < predicate >

|

< predicate > {AND|OR} < predicate >

Ex:

Sales > 100 AND zone IS NULL

Sales BETWEEN 100 AND 200 OR city ='Valencia'

(office IN (1,5,8,9) OR office IS NULL) AND (sales = 0)

Combining predicates

NOT	TRUE	FALSE	NULL
	FALSE	TRUE	NULL

AND	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	NULL
FALSE	FALSE	FALSE	FALSE
NULL	NULL	FALSE	NULL

OR	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	NULL
NULL	TRUE	NULL	NULL

Exercise

- List out for each film, its name, release date, age. Use *Name, Released date and age for the column heading.*
- The same information but only for the films released this year.
- List the language ID of the films released last year.
- We want the oldest film name.
- The previous result but with this format: “The oldest film is : xxxxxxx” where xxxxxxx is the oldest film name.