

IN predicate

[See also](#)

Indicates whether a value exists in a set of values.

```
value [NOT] IN (value_set)
```

Description

Use the IN comparison predicate to filter a table based on the existence of a column value in a specified set of comparison values. The set of comparison values may be either static using a comma-separated list of literals or dynamic using the result set from a subquery.

The value to compare with the values set can be any or a combination of: a column value, a literal value, or a calculated value.

The comparison set can be a static comma-separated list of literal values.

```
SELECT C.Company, C.State
FROM Customer C
WHERE (C.State IN ("CA", "HI"))
```

The comparison set can also be the result set from a subquery. The subquery may return multiple rows, but must only return a single column for comparison.

```
SELECT C.Company, C.State
FROM Customer C
WHERE (C.State IN
      (SELECT R.State
       FROM Regions R
       WHERE (R.Region = "Pacific")))
```

Use NOT to return the converse of an IN comparison.

IN can be used with all non-BLOB data types, but all values compared must be of the same or a compatible data type. If one value is of an incompatible data type, convert that value with the [CAST function](#) to a compatible data type.

Hint IN is useful when filtering to retrieve rows with noncontiguous values. For filtering to retrieve rows with contiguous values that fall within a specified range, use the [BETWEEN predicate](#).