[Skip Headers](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm" \l "BEGIN)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Oracle® Spatial User's Guide and Reference 10*g* Release 1 (10.1)** Part Number B10826-01 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | [[Go to Documentation Home](http://stanford.edu/dept/itss/docs/oracle/10g/index.htm) Home](http://stanford.edu/dept/itss/docs/oracle/10g/index.htm) | [[Go to Book List](http://stanford.edu/dept/itss/docs/oracle/10g/nav/portal_3.htm) Book List](http://stanford.edu/dept/itss/docs/oracle/10g/nav/portal_3.htm) | [[Go to Table of Contents](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/toc.htm) Contents](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/toc.htm) | [[Go to Index](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/index.htm) Index](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/index.htm) | [[Go to Master Index](http://stanford.edu/dept/itss/docs/oracle/10g/mix.101/b12039/toc.htm) Master Index](http://stanford.edu/dept/itss/docs/oracle/10g/mix.101/b12039/toc.htm) | [[Go to Feedback page](http://stanford.edu/dept/itss/docs/oracle/10g/dcommon/html/feedback.htm) Feedback](http://stanford.edu/dept/itss/docs/oracle/10g/dcommon/html/feedback.htm) | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | [[Go to previous page](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_meth.htm) Previous](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_meth.htm) | [[Go to next page](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objgeom.htm) Next](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objgeom.htm) | | [View PDF](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826.pdf) |

**12** **Spatial Operators**

This chapter describes the operators that you can use when working with the spatial object data type. For an overview of spatial operators, including how they differ from spatial procedures and functions, see [Section 1.9](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#BAJDEGDH). [Table 12-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#g996892) lists the main operators.

***Table 12-1 Main Spatial Operators***

| **Operator** | **Description** |
| --- | --- |
| [SDO\_FILTER](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BJAFBCFC) | Specifies which geometries may interact with a given geometry. |
| [SDO\_JOIN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEDJIBF) | Performs a spatial join based on one or more topological relationships. |
| [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) | Determines the nearest neighbor geometries to a geometry. |
| [SDO\_NN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i80529) | Returns the distance of an object returned by the [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) operator. |
| [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) | Determines whether or not two geometries interact in a specified way. (See also [Table 12-2](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEBCFBE) for convenient alternative operators for performing specific mask value operations.) |
| [SDO\_WITHIN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i77653) | Determines if two geometries are within a specified distance from one another. |

[Table 12-2](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEBCFBE) lists operators, provided for convenience, that perform an [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operation of a specific mask type.

***Table 12-2 Convenience Operators for SDO\_RELATE Operations***

| **Operator** | **Description** |
| --- | --- |
| [SDO\_ANYINTERACT](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEJHDGD) | Checks if any geometries in a table have the ANYINTERACT topological relationship with a specified geometry. |
| [SDO\_CONTAINS](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEHCFDH) | Checks if any geometries in a table have the CONTAINS topological relationship with a specified geometry. |
| [SDO\_COVEREDBY](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEHEAEJ) | Checks if any geometries in a table have the COVEREDBY topological relationship with a specified geometry. |
| [SDO\_COVERS](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEGIJFB) | Checks if any geometries in a table have the COVERS topological relationship with a specified geometry. |
| [SDO\_EQUAL](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEBCEJE) | Checks if any geometries in a table have the EQUAL topological relationship with a specified geometry. |
| [SDO\_INSIDE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEFABDH) | Checks if any geometries in a table have the INSIDE topological relationship with a specified geometry. |
| [SDO\_ON](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGECFCGB) | Checks if any geometries in a table have the ON topological relationship with a specified geometry. |
| [SDO\_OVERLAPBDYDISJOINT](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEGDCEI) | Checks if any geometries in a table have the OVERLAPBDYDISJOINT topological relationship with a specified geometry. |
| [SDO\_OVERLAPBDYINTERSECT](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEGBEAI) | Checks if any geometries in a table have the OVERLAPBDYINTERSECT topological relationship with a specified geometry. |
| [SDO\_OVERLAPS](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEDACIF) | Checks if any geometries in a table overlap (that is, have the OVERLAPBDYDISJOINT or OVERLAPBDYINTERSECT topological relationship with) a specified geometry. |
| [SDO\_TOUCH](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEHHIGF) | Checks if any geometries in a table have the TOUCH topological relationship with a specified geometry. |

The rest of this chapter provides reference information on the operators, listed in alphabetical order.

**SDO\_ANYINTERACT**

**Format**

SDO\_ANYINTERACT(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the ANYINTERACT topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=ANYINTERACT'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_ANYINTERACT(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the ANYINTERACT topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the ANYINTERACT relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 4,6, 8,8). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).)

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_ANYINTERACT(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(4,6, 8,8))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

2 cola\_b

1 cola\_a

4 cola\_d

**SDO\_CONTAINS**

**Format**

SDO\_CONTAINS(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the CONTAINS topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=CONTAINS'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_CONTAINS(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the CONTAINS topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the CONTAINS relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 2,2, 4,6). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, only cola\_a contains the query window geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_CONTAINS(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(2,2, 4,6))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

1 cola\_a

**SDO\_COVEREDBY**

**Format**

SDO\_COVEREDBY(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the COVEREDBY topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=COVEREDBY'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_COVEREDBY(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the COVEREDBY topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the COVEREDBY relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 1,1, 5,8). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, only cola\_a is covered by the query window geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_COVEREDBY(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(1,1, 5,8))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

1 cola\_a

**SDO\_COVERS**

**Format**

SDO\_COVERS(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the COVERS topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=COVERS'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_COVERS(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the COVERS topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the COVERS relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 1,1, 4,6). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, only cola\_a covers the query window geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_COVERS(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(1,1, 4,6))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

1 cola\_a

**SDO\_EQUAL**

**Format**

SDO\_EQUAL(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the EQUAL topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=EQUAL'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_EQUAL(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the EQUAL topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the EQUAL relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 1,1, 5,7). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, cola\_a (and only cola\_a) has the same boundary and interior as the query window geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_EQUAL(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(1,1, 5,7))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

1 cola\_a

**SDO\_FILTER**

**Format**

SDO\_FILTER(geometry1, geometry2);

**Description**

Uses the spatial index to identify either the set of spatial objects that are likely to interact spatially with a given object (such as an area of interest), or pairs of spatial objects that are likely to interact spatially. Objects interact spatially if they are not disjoint.

This operator performs only a primary filter operation. The secondary filtering operation, performed by the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator, can be used to determine with certainty if objects interact spatially.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_FILTER(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that are non-disjoint, and FALSE otherwise.

**Usage Notes**

SDO\_FILTER is the only operator that can be used with data that is indexed using more than two dimensions. The operator considers all dimensions specified in the spatial index.

The operator must always be used in a WHERE clause and the condition that includes the operator should be an expression of the form SDO\_FILTER(arg1, arg2) = 'TRUE'.

geometry2 can come from a table or be a transient SDO\_GEOMETRY object (such as a bind variable or SDO\_GEOMETRY constructor).

* If the geometry2 column is not spatially indexed, the operator indexes the query window in memory and performance is very good.
* If the geometry2 column is spatially indexed with the same SDO\_LEVEL value as the geometry1 column, the operator reuses the existing index, and performance is very good or better.
* If the geometry2 column is spatially indexed with a different SDO\_LEVEL value than the geometry1 column, the operator reindexes geometry2 in the same way as if there were no index on the column originally, and then performance is very good.
* If two or more geometries from geometry2 are passed to the operator, the ORDERED optimizer hint must be specified, and the table in geometry2 must be specified first in the FROM clause.

If geometry1 and geometry2 are based on different coordinate systems, geometry2 is temporarily transformed to the coordinate system of geometry1 for the operation to be performed, as described in [Section 6.7.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_cs_concepts.htm#CIHBJCAG).

In previous releases, the SDO\_FILTER operator required a third parameter. Effective with Oracle Spatial release 10.1, the operator has only two parameters. For backward compatibility, any keywords for the third parameter that were supported in the previous release will still work; however, the use of those keywords is discouraged and is not supported for new uses of the operator.

**Examples**

The following example selects the geometries that are likely to interact with a query window (here, a rectangle with lower-left, upper-right coordinates 4,6, 8,8). (The example uses the definitions and data from [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750).)

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_FILTER(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(4,6, 8,8))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

2 cola\_b

1 cola\_a

4 cola\_d

The following example selects the GID values from the POLYGONS table where the GEOMETRY column objects are likely to interact spatially with the GEOMETRY column object in the QUERY\_POLYS table that has a GID value of 1.

SELECT A.gid

FROM Polygons A, query\_polys B

WHERE B.gid = 1

AND SDO\_FILTER(A.Geometry, B.Geometry) = 'TRUE';

The following example selects the GID values from the POLYGONS table where the GEOMETRY column object is likely to interact spatially with the geometry stored in the aGeom variable.

Select A.Gid

FROM Polygons A

WHERE SDO\_FILTER(A.Geometry, :aGeom) = 'TRUE';

The following example selects the GID values from the POLYGONS table where the GEOMETRY column object is likely to interact spatially with the specified rectangle having the lower-left coordinates (x1,y1) and the upper-right coordinates (x2, y2).

Select A.Gid

FROM Polygons A

WHERE SDO\_FILTER(A.Geometry, sdo\_geometry(2003,NULL,NULL,

sdo\_elem\_info\_array(1,1003,3),

sdo\_ordinate\_array(x1,y1,x2,y2))

) = 'TRUE';

The following example selects the GID values from the POLYGONS table where the GEOMETRY column object is likely to interact spatially with any GEOMETRY column object in the QUERY\_POLYS table. In this example, the ORDERED optimizer hint is used and the QUERY\_POLYS (geometry2) table is specified first in the FROM clause, because multiple geometries from geometry2 are involved (see the Usage Notes).

SELECT /\*+ ORDERED \*/

A.gid

FROM query\_polys B, polygons A

WHERE SDO\_FILTER(A.Geometry, B.Geometry) = 'TRUE';

**Related Topics**

* [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531)

**SDO\_INSIDE**

**Format**

SDO\_INSIDE(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the INSIDE topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=INSIDE'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_INSIDE(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the INSIDE topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the INSIDE relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 5,6, 12,12). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, only cola\_d (the circle) is inside the query window geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_INSIDE(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(5,6, 12,12))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

4 cola\_d

**SDO\_JOIN**

**Format**

SDO\_JOIN(table\_name1, column\_name1, table\_name2, column\_name2, params,  
preserve\_join\_order) RETURN SDO\_ROWIDSET;

**Description**

Performs a spatial join based on one or more topological relationships.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| table\_name1 | Name of the first table to be used in the spatial join operation. The table must have a column of type SDO\_GEOMETRY. Data type is VARCHAR2. |
| column\_name1 | Name of the spatial column of type SDO\_GEOMETRY in table\_name1. A spatial R-tree index must be defined on this column. Data type is VARCHAR2. |
| table\_name2 | Name of the second table to be used in the spatial join operation. (It can be the same as or different from table\_name1.) The table must have a column of type SDO\_GEOMETRY. Data type is VARCHAR2. |
| column\_name2 | Name of the spatial column of type SDO\_GEOMETRY in table\_name2. A spatial R-tree index must be defined on this column. Data type is VARCHAR2. |
| params | Optional parameter string of keywords and values; available only if mask=ANYINTERACT. Determines the behavior of the operator. See [Table 12-3](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEEACJD) in the Usage Notes for information about the available keywords. Data type is VARCHAR2. Default is NULL. |
| preserve\_join\_order | Optional parameter to specify if the join order is guaranteed to be preserved during processing of the operator. If the value is 0 (the default), the order of the tables might be changed; if the value is 1, the order of the tables is not changed. Data type is NUMBER. Default is 0. |

**Returns**

SDO\_JOIN returns an object of SDO\_ROWIDSET, which consists of a table of objects of SDO\_ROWIDPAIR. Oracle Spatial defines the type SDO\_ROWIDSET as:

CREATE TYPE sdo\_rowidset as TABLE OF sdo\_rowidpair;

Oracle Spatial defines the object type SDO\_ROWIDPAIR as:

CREATE TYPE sdo\_rowidpair AS OBJECT

(rowid1 VARCHAR2(24),

rowid2 VARCHAR2(24));

In the SDO\_ROWIDPAIR definition, rowid1 refers to a rowid from table\_name1, and rowid2 refers to a rowid from table\_name2.

**Usage Notes**

SDO\_JOIN is technically not an operator, but a table function. (For an explanation of table functions, see [*PL/SQL User's Guide and Reference*](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10807/toc.htm).) However, it is presented in the chapter with Spatial operators because its usage is similar to that of the operators, and because it is not part of a package with other functions and procedures.

This function is recommended when you need to perform full table joins.

The geometries in column\_name1 and column\_name2 must have the same SRID (coordinate system) value and the same number of dimensions.

[Table 12-3](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEEACJD) shows the keywords for the params parameter.

***Table 12-3 params Keywords for the SDO\_JOIN Operator***

| **Keyword** | **Description** |
| --- | --- |
| mask | The topological relationship of interest.Valid values are 'mask=*<value>*' where *<value>* is one or more of the mask values valid for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator (TOUCH, OVERLAPBDYDISJOINT, OVERLAPBDYINTERSECT, EQUAL, INSIDE, COVEREDBY, CONTAINS, COVERS, ANYINTERACT, ON), or FILTER, which checks if the MBRs (the filter-level approximations) intersect. Multiple masks are combined with the logical Boolean operator OR (for example, 'mask=inside+touch'); however, FILTER cannot be combined with any other mask.  If this parameter is null or contains an empty string, mask=FILTER is assumed. |
| distance | Specifies a numeric distance value that is added to the tolerance value (explained in [Section 1.5.5](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i884589)) before the relationship checks are performed. For example, if the tolerance is 10 meters and you specify 'distance=100 unit=meter', two objects are considered to have spatial interaction if they are within 110 meters of each other.If you specify distance but not unit, the unit of measurement associated with the data is assumed. |
| unit | Specifies a unit of measurement to be associated with the distance value (for example, 'distance=100 unit=meter'). See [Section 2.6](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1010145) for more information about unit of measurement specification. If you specify unit, you must also specify distance.Data type is VARCHAR2. Default = unit of measurement associated with the data. For geodetic data, the default is meters. |

**Examples**

The following example joins the COLA\_MARKETS table with itself to find, for each geometry, all other geometries that have any spatial interaction with it. (The example uses the definitions and data from [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750).) In this example, rowid1 and rowid2 correspond to the names of the attributes in the SDO\_ROWIDPAIR type definition. Note that in the output, cola\_d (the circle in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467)) interacts only with itself, and not with any of the other geometries.

SELECT a.name, b.name FROM cola\_markets a, cola\_markets b,

TABLE(SDO\_JOIN('COLA\_MARKETS', 'SHAPE', 'COLA\_MARKETS', 'SHAPE',

'mask=ANYINTERACT')) c

WHERE c.rowid1 = a.rowid AND c.rowid2 = b.rowid ORDER BY a.name;

NAME NAME

-------------------------------- --------------------------------

cola\_a cola\_a

cola\_a cola\_b

cola\_a cola\_c

cola\_b cola\_a

cola\_b cola\_b

cola\_b cola\_c

cola\_c cola\_a

cola\_c cola\_b

cola\_c cola\_c

cola\_d cola\_d

10 rows selected.

**Related Topics**

* [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531)

**SDO\_NN**

**Format**

SDO\_NN(geometry1, geometry2, param [, number]);

**Description**

Uses the spatial index to identify the nearest neighbors for a geometry.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. The nearest neighbor or neighbors to geometry2 will be returned from geometry1. (geometry2 is specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |
| param | Determines the behavior of the operator. The available keywords are listed in [Table 12-4](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#g996876). If you do not specify this parameter, the operator returns all rows in increasing distance order from geometry2. Data type is VARCHAR2. |
| number | If the [SDO\_NN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i80529) ancillary operator is included in the call to SDO\_NN, specifies the same number used in the call to [SDO\_NN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i80529). Data type is NUMBER. |

[Table 12-4](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#g996876) lists the keywords for the param parameter.

***Table 12-4 Keywords for the SDO\_NN Param Parameter***

| **Keyword** | **Description** |
| --- | --- |
| sdo\_batch\_size | Specifies the number of rows to be evaluated at a time when the SDO\_NN expression may need to be evaluated multiple times in order to return the desired number of results that satisfy the WHERE clause. Available only when an R-tree index is used. If you specify sdo\_batch\_size=0 (or if you omit the param parameter completely), Spatial calculates a batch size suited to the result set size. See the Usage Notes and Examples for more information. Data type is NUMBER.  For example: 'sdo\_batch\_size=10' |
| sdo\_num\_res | If sdo\_batch\_size is not specified, specifies the number of results (nearest neighbors) to be returned. If sdo\_batch\_size is specified, this keyword is ignored; instead, use the ROWNUM pseudocolumn to limit the number of results. See the Usage Notes and Examples for more information. Data type is NUMBER. Default = 1.  For example: 'sdo\_num\_res=5' |
| unit | If the [SDO\_NN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i80529) ancillary operator is included in the call to SDO\_NN, specifies the unit of measurement: a quoted string with unit= and an SDO\_UNIT value from the MDSYS.SDO\_DIST\_UNITS table. See [Section 2.6](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1010145) for more information about unit of measurement specification. Data type is VARCHAR2. Default = unit of measurement associated with the data. For geodetic data, the default is meters.  For example: 'unit=KM' |

**Returns**

This operator returns the sdo\_num\_res number of objects from geometry1 that are nearest to geometry2 in the query. In determining how near two geometry objects are, the shortest possible distance between any two points on the surface of each object is used.

**Usage Notes**

The operator is disabled if the table does not have a spatial index or if the index has been built on more than two dimensions.

The operator must always be used in a WHERE clause, and the condition that includes the operator should be an expression of the form SDO\_NN(arg1, arg2, '<some\_parameter>') = 'TRUE'.

The operator can be used in two ways:

* If all geometries in the layer are candidates, use the sdo\_num\_res keyword to specify the number of geometries returned.
* If any geometries in the table might be nearer than the geometries specified in the WHERE clause, use the sdo\_batch\_size keyword and use the WHERE clause (including the ROWNUM pseudocolumn) to limit the number of geometries returned.

Specify the sdo\_batch\_size keyword if any geometries in the table might be nearer than the geometries specified in the WHERE clause. For example, assume that a RESTAURANTS table contains different types of restaurants, and you want to find the two nearest Italian restaurants to your hotel. The query might look like the following:

SELECT r.name FROM restaurants r WHERE

SDO\_NN(r.geometry, :my\_hotel, 'sdo\_batch\_size=10') = 'TRUE'

AND r.cuisine = 'Italian' AND ROWNUM <=2;

If the sdo\_batch\_size keyword is not specified in this example, only the two nearest restaurants are returned, regardless of their CUISINE value; and if the CUISINE value of these two rows is not Italian, the query may return no rows. The ROWNUM <=2 clause is necessary to limit the number of results returned to no more than 2 where CUISINE is Italian.

The sdo\_batch\_size value can affect the performance of nearest neighbor queries. A good general guideline is to specify the number of candidate rows likely to satisfy the WHERE clause. Using the preceding example of a query for Italian restaurants, if approximately 20 percent of the restaurants nearest to the hotel are Italian and if you want 2 restaurants, an sdo\_batch\_size value of 10 will probably result in the best performance. On the other hand, if only approximately 5 percent of the restaurants nearest to the hotel are Italian and if you want 2 restaurants, an sdo\_batch\_size value of 40 would be better.

You can specify sdo\_batch\_size=0, which causes Spatial to calculate a batch size that is suitable for the result set size. However, the calculated batch size may not be optimal, and the calculation incurs some processing overhead; if you can determine a good sdo\_batch\_size value for a query, the performance will probably be better than if you specify sdo\_batch\_size=0.

If the sdo\_batch\_size keyword is specified, any sdo\_num\_res value is ignored. Do not specify both keywords.

Specify the number parameter only if you are using the [SDO\_NN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i80529) ancillary operator in the call to SDO\_NN. See the information about the [SDO\_NN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i80529) operator in this chapter.

If this operator is used with geodetic data, the data must be indexed with an R-tree spatial index. If this operator is used with geodetic data and if the R-tree spatial index is created with 'geodetic=false' specified, you cannot use the unit parameter.

If two or more objects from geometry1 are an equal distance from geometry2, any of the objects can be returned on any call to the function. For example, if item\_a, item\_b, and item\_c are nearest to and equally distant from geometry2, and if SDO\_NUM\_RES=2, two of those three objects are returned, but they can be any two of the three.

If the SDO\_NN operator uses a partitioned spatial index (see [Section 4.1.6](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_index_query.htm#i1005901)), the requested number of geometries is returned for *each* partition that contains candidate rows based on the query criteria. For example, if you request the 5 nearest restaurants to a point and the spatial index has 4 partitions, the operator returns up to 20 (5\*4) geometries. In this case, you must use the ROWNUM pseudocolumn (here, WHERE ROWNUM <=5) to return the 5 nearest restaurants.

If geometry1 and geometry2 are based on different coordinate systems, geometry2 is temporarily transformed to the coordinate system of geometry1 for the operation to be performed, as described in [Section 6.7.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_cs_concepts.htm#CIHBJCAG).

SDO\_NN is not supported for spatial joins.

In some situations the SDO\_NN operator will not use the spatial index unless an optimizer hint forces the index to be used. This can occur when a query involves a join; and if the optimizer hint is not used in such situations, an internal error occurs. To prevent such errors, you should always specify an optimizer hint to use the spatial index with the SDO\_NN operator, regardless of how simple or complex the query is. For example, the following excerpt from a query specifies to use the COLA\_SPATIAL\_IDX index that is defined on the COLA\_MARKETS table:

SELECT /\*+ INDEX(c cola\_spatial\_idx) \*/

c.mkt\_id, c.name, ... FROM cola\_markets c, ...;

However, if there is an index associated with the column predicate in the WHERE clause, be sure that this index is not used by specifying the NO\_INDEX hint for that index. For example, if there was an index named COLA\_NAME\_IDX defined on the NAME column, you would need to specify the hints in the preceding example as follows:

SELECT /\*+ INDEX(c cola\_spatial\_idx) NO\_INDEX(c cola\_name\_idx) \*/

c.mkt\_id, c.name, ... FROM cola\_markets c, ...;

(Note, however, that there is no index named COLA\_NAME\_IDX in the example in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750).)

For detailed information about using optimizer hints, see [*Oracle Database Performance Tuning Guide*](http://stanford.edu/dept/itss/docs/oracle/10g/server.101/b10752/toc.htm).

**Examples**

The following example finds the two objects from the SHAPE column in the COLA\_MARKETS table that are nearest to a specified point (10,7). (The example uses the definitions and data from [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750).)

SELECT /\*+ INDEX(c cola\_spatial\_idx) \*/

c.mkt\_id, c.name FROM cola\_markets c WHERE SDO\_NN(c.shape,

sdo\_geometry(2001, NULL, sdo\_point\_type(10,7,NULL), NULL,

NULL), 'sdo\_num\_res=2') = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

2 cola\_b

4 cola\_d

The following example uses the sdo\_batch\_size keyword to find the two objects (ROWNUM <=2), with a NAME value less than 'cola\_d', from the SHAPE column in the COLA\_MARKETS table that are nearest to a specified point (10,7). The value of 3 for sdo\_batch\_size represents a best guess at the number of nearest geometries that need to be evaluated before the WHERE clause condition is satisfied. (The example uses the definitions and data from [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750).)

SELECT /\*+ INDEX(c cola\_spatial\_idx) \*/ c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_NN(c.shape, sdo\_geometry(2001, NULL,

sdo\_point\_type(10,7,NULL), NULL, NULL),

'sdo\_batch\_size=3') = 'TRUE'

AND c.name < 'cola\_d' AND ROWNUM <= 2;

MKT\_ID NAME

---------- --------------------------------

2 cola\_b

3 cola\_c

See also the more complex SDO\_NN examples in [Section C.3](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_complex_queries.htm#CIHHIIJE).

**Related Topics**

* [SDO\_NN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i80529)

**SDO\_NN\_DISTANCE**

**Format**

SDO\_NN\_DISTANCE(number);

**Description**

Returns the distance of an object returned by the [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) operator. Valid only within a call to the [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) operator.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| number | Specifies a number that must be the same as the last parameter passed to the [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) operator. Data type is NUMBER. |

**Returns**

This operator returns the distance of an object returned by the [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) operator. In determining how near two geometry objects are, the shortest possible distance between any two points on the surface of each object is used.

**Usage Notes**

SDO\_NN\_DISTANCE is an ancillary operator to the [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) operator. It returns the distance between the specified geometry and a nearest neighbor object. This distance is passed as ancillary data to the [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) operator. (For an explanation of how operators can use ancillary data, see the section on ancillary data in the chapter on domain indexes in [*Oracle Data Cartridge Developer's Guide*](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10800/toc.htm).)

You can choose any arbitrary number for the number parameter. The only requirement is that it must match the last parameter in the call to the [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) operator.

Use a bind variable to store and operate on the distance value.

**Examples**

The following example finds the two objects from the SHAPE column in the COLA\_MARKETS table that are nearest to a specified point (10,7), and it finds the distance between each object and the point. (The example uses the definitions and data from [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750).)

SELECT /\*+ INDEX(c cola\_spatial\_idx) \*/

c.mkt\_id, c.name, SDO\_NN\_DISTANCE(1) dist

FROM cola\_markets c

WHERE SDO\_NN(c.shape, sdo\_geometry(2001, NULL,

sdo\_point\_type(10,7,NULL), NULL, NULL),

'sdo\_num\_res=2', 1) = 'TRUE' ORDER BY dist;

MKT\_ID NAME DIST

---------- -------------------------------- ----------

4 cola\_d .828427125

2 cola\_b 2.23606798

Note the following about this example:

* 1 is used as the number parameter for SDO\_NN\_DISTANCE, and 1 is also specified as the last parameter to [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067) (after 'sdo\_num\_res=2').
* The column alias dist holds the distance between the object and the point. (For geodetic data, the distance unit is meters; for non-geodetic data, the distance unit is the unit associated with the data.)

**Related Topics**

* [SDO\_NN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78067)

**SDO\_ON**

**Format**

SDO\_ON(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the ON topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=ON'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_ON(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the ON topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the ON relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 4,6, 8,8). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) This example returns no rows because there are no line string geometries in the SHAPE column.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_ON(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(4,6, 8,8))

) = 'TRUE';

no rows selected

**SDO\_OVERLAPBDYDISJOINT**

**Format**

SDO\_OVERLAPBDYDISJOINT(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the OVERLAPBDYDISJOINT topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=OVERLAPBDYDISJOINT'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_OVERLAPBDYDISJOINT(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the OVERLAPBDYDISJOINT topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the OVERLAPBDYDISJOINT relationship with a line string geometry (here, a horizontal line from 0,6 to 2,6). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, only cola\_a has the OVERLAPBDYDISJOINT relationship with the line string geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_OVERLAPBDYDISJOINT(c.shape,

SDO\_GEOMETRY(2002, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,2,1),

SDO\_ORDINATE\_ARRAY(0,6, 2,6))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

1 cola\_a

**SDO\_OVERLAPBDYINTERSECT**

**Format**

SDO\_OVERLAPBDYINTERSECT(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the OVERLAPBDYINTERSECT topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=OVERLAPBDYINTERSECT'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_OVERLAPBDYINTERSECT(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the OVERLAPBDYINTERSECT topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the OVERLAPBDYINTERSECT relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 4,6, 8,8). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, cola\_a, cola\_b, and cola\_d have the OVERLAPBDYINTERSECT relationship with the query window geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_OVERLAPBDYINTERSECT(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(4,6, 8,8))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

2 cola\_b

1 cola\_a

4 cola\_d

**SDO\_OVERLAPS**

**Format**

SDO\_OVERLAPS(geometry1, geometry2);

**Description**

Checks if any geometries in a table overlap (that is, have the OVERLAPBDYDISJOINT or OVERLAPBDYINTERSECT topological relationship with) a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=OVERLAPBDYDISJOINT+OVERLAPBDYINTERSECT'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_OVERLAPS(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the OVERLAPBDYDISJOINT or OVERLAPBDYINTERSECT topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that overlap a query window (here, a rectangle with lower-left, upper-right coordinates 4,6, 8,8). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, three of the geometries in the SHAPE column overlap the query window geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_OVERLAPS(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(4,6, 8,8))

) = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

2 cola\_b

1 cola\_a

4 cola\_d

**SDO\_RELATE**

**Format**

SDO\_RELATE(geometry1, geometry2, param);

**Description**

Uses the spatial index to identify either the spatial objects that have a particular spatial interaction with a given object such as an area of interest, or pairs of spatial objects that have a particular spatial interaction.

This operator performs both primary and secondary filter operations.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |
| param | Uses the mask keyword to specify the topological relationship of interest. This is a required parameter. Data type is VARCHAR2.  Valid mask keyword values are one or more of the following in the nine-intersection pattern: TOUCH, OVERLAPBDYDISJOINT, OVERLAPBDYINTERSECT, EQUAL, INSIDE, COVEREDBY, CONTAINS, COVERS, ANYINTERACT, ON. Multiple masks are combined with the logical Boolean operator OR, for example, 'mask=inside+touch'; however, see the Usage Notes for an alternative syntax using UNION ALL that may result in better performance. See [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253) for an explanation of the nine-intersection relationship pattern.  For backward compatibility, any additional keywords for the param parameter that were supported in the previous release will still work; however, the use of those keywords is discouraged and is not supported for new uses of the operator. |

**Returns**

The expression SDO\_RELATE(geometry1,geometry2, 'mask = <some\_mask\_val>') = 'TRUE' evaluates to TRUE for object pairs that have the topological relationship specified by <some\_mask\_val>, and FALSE otherwise.

**Usage Notes**

The operator is disabled if the table does not have a spatial index or if the index has been built on more than two dimensions.

The operator must always be used in a WHERE clause, and the condition that includes the operator should be an expression of the form SDO\_RELATE(arg1, arg2, 'mask = <some\_mask\_val>') = 'TRUE'.

geometry2 can come from a table or be a transient SDO\_GEOMETRY object (such as a bind variable or SDO\_GEOMETRY constructor).

* If the geometry2 column is not spatially indexed, the operator indexes the query window in memory and performance is very good.
* If the geometry2 column is spatially indexed with the same SDO\_LEVEL value as the geometry1 column, the operator reuses the existing index, and performance is very good or better.
* If the geometry2 column is spatially indexed with a different SDO\_LEVEL value than the geometry1 column, the operator reindexes geometry2 in the same way as if there were no index on the column originally, and then performance is very good.
* If two or more geometries from geometry2 are passed to the operator, the ORDERED optimizer hint must be specified, and the table in geometry2 must be specified first in the FROM clause.

If geometry1 and geometry2 are based on different coordinate systems, geometry2 is temporarily transformed to the coordinate system of geometry1 for the operation to be performed, as described in [Section 6.7.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_cs_concepts.htm#CIHBJCAG).

Unlike with the [SDO\_GEOM.RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objgeom.htm#BGHCDIDG) function, DISJOINT and DETERMINE masks are not allowed in the relationship mask with the SDO\_RELATE operator. This is because SDO\_RELATE uses the spatial index to find candidates that may interact, and the information to satisfy DISJOINT or DETERMINE is not present in the index.

Although multiple masks can be combined using the logical Boolean operator OR, for example, 'mask=inside+coveredby', better performance may result if the spatial query specifies each mask individually and uses the UNION ALL syntax to combine the results. This is due to internal optimizations that Spatial can apply under certain conditions when masks are specified singly rather than grouped within the same SDO\_RELATE operator call. For example, consider the following query using the logical Boolean operator OR to group multiple masks:

SELECT a.gid

FROM polygons a, query\_polys B

WHERE B.gid = 1

AND SDO\_RELATE(A.Geometry, B.Geometry,

'mask=inside+coveredby') = 'TRUE';

The preceding query may result in better performance if it is expressed as follows, using UNION ALL to combine results of multiple SDO\_RELATE operator calls, each with a single mask:

SELECT a.gid

FROM polygons a, query\_polys B

WHERE B.gid = 1

AND SDO\_RELATE(A.Geometry, B.Geometry,

'mask=inside') = 'TRUE'

UNION ALL

SELECT a.gid

FROM polygons a, query\_polys B

WHERE B.gid = 1

AND SDO\_RELATE(A.Geometry, B.Geometry,

'mask=coveredby') = 'TRUE';

**Examples**

The following examples are similar to those for the [SDO\_FILTER](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BJAFBCFC) operator; however, they identify a specific type of interaction (using the mask keyword), and they determine with certainty (not mere likelihood) if the spatial interaction occurs.

The following example selects the geometries that have any interaction with a query window (here, a rectangle with lower-left, upper-right coordinates 4,6, 8,8). (The example uses the definitions and data from [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750).)

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_RELATE(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(4,6, 8,8)),

'mask=anyinteract') = 'TRUE';

MKT\_ID NAME

---------- --------------------------------

2 cola\_b

1 cola\_a

4 cola\_d

The following example selects the GID values from the POLYGONS table where the GEOMETRY column objects have any spatial interaction with the GEOMETRY column object in the QUERY\_POLYS table that has a GID value of 1.

SELECT A.gid

FROM Polygons A, query\_polys B

WHERE B.gid = 1

AND SDO\_RELATE(A.Geometry, B.Geometry,

'mask=ANYINTERACT') = 'TRUE';

The following example selects the GID values from the POLYGONS table where a GEOMETRY column object has any spatial interaction with the geometry stored in the aGeom variable.

Select A.Gid

FROM Polygons A

WHERE SDO\_RELATE(A.Geometry, :aGeom, 'mask=ANYINTERACT') = 'TRUE';

The following example selects the GID values from the POLYGONS table where a GEOMETRY column object has any spatial interaction with the specified rectangle having the lower-left coordinates (x1,y1) and the upper-right coordinates (x2, y2).

Select A.Gid

FROM Polygons A

WHERE SDO\_RELATE(A.Geometry, sdo\_geometry(2003,NULL,NULL,

sdo\_elem\_info\_array(1,1003,3),

sdo\_ordinate\_array(x1,y1,x2,y2)),

'mask=ANYINTERACT') = 'TRUE';

The following example selects the GID values from the POLYGONS table where the GEOMETRY column object has any spatial interaction with any GEOMETRY column object in the QUERY\_POLYS table. In this example, the ORDERED optimizer hint is used and QUERY\_POLYS (geometry2) table is specified first in the FROM clause, because multiple geometries from geometry2 are involved (see the Usage Notes).

SELECT /\*+ ORDERED \*/

A.gid

FROM query\_polys B, polygons A

WHERE SDO\_RELATE(A.Geometry, B.Geometry, 'mask=ANYINTERACT') = 'TRUE';

**Related Topics**

* [SDO\_FILTER](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BJAFBCFC)
* [SDO\_JOIN](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BGEDJIBF)
* [SDO\_WITHIN\_DISTANCE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i77653)
* [SDO\_GEOM.RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objgeom.htm#BGHCDIDG) function

**SDO\_TOUCH**

**Format**

SDO\_TOUCH(geometry1, geometry2);

**Description**

Checks if any geometries in a table have the TOUCH topological relationship with a specified geometry. Equivalent to specifying the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator with 'mask=TOUCH'.

See the section on the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter for information about the operations performed by this operator and for usage requirements.

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| geometry2 | Specifies either a geometry from a table or a transient instance of a geometry. (Specified using a bind variable or SDO\_GEOMETRY constructor.) Data type is SDO\_GEOMETRY. |

**Returns**

The expression SDO\_TOUCH(geometry1,geometry2) = 'TRUE' evaluates to TRUE for object pairs that have the TOUCH topological relationship, and FALSE otherwise.

**Usage Notes**

See the Usage Notes for the [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531) operator in this chapter.

For an explanation of the topological relationships and the nine-intersection model used by Spatial, see [Section 1.8](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_intro.htm#i880253).

**Examples**

The following example finds geometries that have the TOUCH relationship with a query window (here, a rectangle with lower-left, upper-right coordinates 1,1, 5,7). (The example uses the definitions and data described in [Section 2.1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004750) and illustrated in [Figure 2-1](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1004467).) In this example, only cola\_b has the TOUCH relationship with the query window geometry.

SELECT c.mkt\_id, c.name

FROM cola\_markets c

WHERE SDO\_TOUCH(c.shape,

SDO\_GEOMETRY(2003, NULL, NULL,

SDO\_ELEM\_INFO\_ARRAY(1,1003,3),

SDO\_ORDINATE\_ARRAY(1,1, 5,7))

) = 'TRUE';

FROM cola\_markets c

MKT\_ID NAME

---------- --------------------------------

2 cola\_b

**SDO\_WITHIN\_DISTANCE**

**Format**

SDO\_WITHIN\_DISTANCE(geometry1, aGeom, params);

**Description**

Uses the spatial index to identify the set of spatial objects that are within some specified distance of a given object (such as an area of interest or point of interest).

**Keywords and Parameters**

| **Value** | **Description** |
| --- | --- |
| geometry1 | Specifies a geometry column in a table. The column has the set of geometry objects that will be operated on to determine if they are within the specified distance of the given object (aGeom). The column must be spatially indexed. Data type is SDO\_GEOMETRY. |
| aGeom | Specifies the object to be checked for distance against the geometry objects in geometry1. Specify either a geometry from a table (using a bind variable) or a transient instance of a geometry (using the SDO\_GEOMETRY constructor). Data type is SDO\_GEOMETRY. |
| params | A quoted string containing one or more keywords (with values) that determine the behavior of the operator. The remaining items (distance, querytype, and unit) are potential keywords for the params parameter. Data type is VARCHAR2. |
| distance | Specifies the distance value. If a coordinate system is associated with the geometry, the distance unit is assumed to be the unit associated with the coordinate system. This is a required keyword. Data type is NUMBER. |
| querytype | Set 'querytype=FILTER' to perform only a primary filter operation. If querytype is not specified, both primary and secondary filter operations are performed (default). Data type is VARCHAR2. |
| unit | Specifies the unit of measurement: a quoted string with unit= and an SDO\_UNIT value from the MDSYS.SDO\_DIST\_UNITS table (for example, 'unit=KM'). See [Section 2.6](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objrelschema.htm#i1010145) for more information about unit of measurement specification. Data type is NUMBER. Default = unit of measurement associated with the data. For geodetic data, the default is meters. |

**Returns**

The expression SDO\_WITHIN\_DISTANCE(arg1, arg2, arg3) = 'TRUE' evaluates to TRUE for object pairs that are within the specified distance, and FALSE otherwise.

**Usage Notes**

Distance between two extended objects (nonpoint objects such as lines and polygons) is defined as the minimum distance between these two objects. The distance between two adjacent polygons is zero.

If this operator is used with geodetic data, the data must be indexed with an R-tree spatial index. If this operator is used with geodetic data and if the R-tree spatial index is created with 'geodetic=false' specified, you cannot use the unit parameter.

The operator is disabled if the table does not have a spatial index or if the index has been built on more than two dimensions.

The operator must always be used in a WHERE clause and the condition that includes the operator should be an expression of the form:

SDO\_WITHIN\_DISTANCE(arg1, arg2, 'distance = <some\_dist\_val>') = 'TRUE'

The geometry column must have a spatial index built on it. If the data is geodetic, the spatial index must be an R-tree index.

SDO\_WITHIN\_DISTANCE is not supported for spatial joins. See [Section 4.2.1.3](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_index_query.htm#i1005386) for a discussion on how to perform a spatial join within-distance operation.

**Examples**

The following example selects the GID values from the POLYGONS table where the GEOMETRY column object is within 10 distance units of the geometry stored in the aGeom variable.

SELECT A.GID

FROM POLYGONS A

WHERE

SDO\_WITHIN\_DISTANCE(A.Geometry, :aGeom, 'distance = 10') = 'TRUE';

The following example selects the GID values from the POLYGONS table where the GEOMETRY column object is within 10 distance units of the specified rectangle having the lower-left coordinates (x1,y1) and the upper-right coordinates (x2, y2).

SELECT A.GID

FROM POLYGONS A

WHERE

SDO\_WITHIN\_DISTANCE(A.Geometry, sdo\_geometry(2003,NULL,NULL,

sdo\_elem\_info\_array(1,1003,3),

sdo\_ordinate\_array(x1,y1,x2,y2)),

'distance = 10') = 'TRUE';

The following example selects the GID values from the POLYGONS table where the GID value in the QUERY\_POINTS table is 1 and a POLYGONS.GEOMETRY object is within 10 distance units of the QUERY\_POINTS.GEOMETRY object.

SELECT A.GID

FROM POLYGONS A, Query\_Points B

WHERE B.GID = 1 AND

SDO\_WITHIN\_DISTANCE(A.Geometry, B.Geometry, 'distance = 10') = 'TRUE';

See also the more complex SDO\_WITHIN\_DISTANCE examples in [Section C.2](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_complex_queries.htm#CIHIAIDD).

**Related Topics**

* [SDO\_FILTER](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#BJAFBCFC)
* [SDO\_RELATE](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_operat.htm#i78531)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | [[Go to previous page](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_meth.htm) Previous](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_meth.htm) | [[Go to next page](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objgeom.htm) Next](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/sdo_objgeom.htm) | | Oracle [Copyright © 1999, 2003 Oracle Corporation](http://stanford.edu/dept/itss/docs/oracle/10g/dcommon/html/cpyr.htm) All Rights Reserved. | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | [[Go to Documentation Home](http://stanford.edu/dept/itss/docs/oracle/10g/index.htm) Home](http://stanford.edu/dept/itss/docs/oracle/10g/index.htm) | [[Go to Book List](http://stanford.edu/dept/itss/docs/oracle/10g/nav/portal_3.htm) Book List](http://stanford.edu/dept/itss/docs/oracle/10g/nav/portal_3.htm) | [[Go to Table of Contents](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/toc.htm) Contents](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/toc.htm) | [[Go to Index](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/index.htm) Index](http://stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10826/index.htm) | [[Go to Master Index](http://stanford.edu/dept/itss/docs/oracle/10g/mix.101/b12039/toc.htm) Master Index](http://stanford.edu/dept/itss/docs/oracle/10g/mix.101/b12039/toc.htm) | [[Go to Feedback page](http://stanford.edu/dept/itss/docs/oracle/10g/dcommon/html/feedback.htm) Feedback](http://stanford.edu/dept/itss/docs/oracle/10g/dcommon/html/feedback.htm) | |