**Sadiq S.**

# Practical 4

## AIM: Spatial Database

**Create a spatial database table that stores the number, name and location, which consists of four different areas say cola_a , cola_b , cola_c , cola_d.**

## What is Spatial Database?

- Spatial data is associated with geographic locations such as cities, towns etc.
- A spatial database is optimized to store and query data representing objects.
- These are the objects which are defined in a geometric space.

## Characteristics of Spatial Database:

A spatial database system has the following characteristics:

- It is a database system
- It offers spatial data types (SDTs) in its data model and query language.
- It supports spatial data types in its implementation, providing at least spatial indexing and efficient algorithms for spatial join.

## Example:

A road map is a visualization of geographic information. A road map is a 2-dimensional object which contains points, lines, and polygons that can represent cities, roads, and political boundaries such as states or provinces.

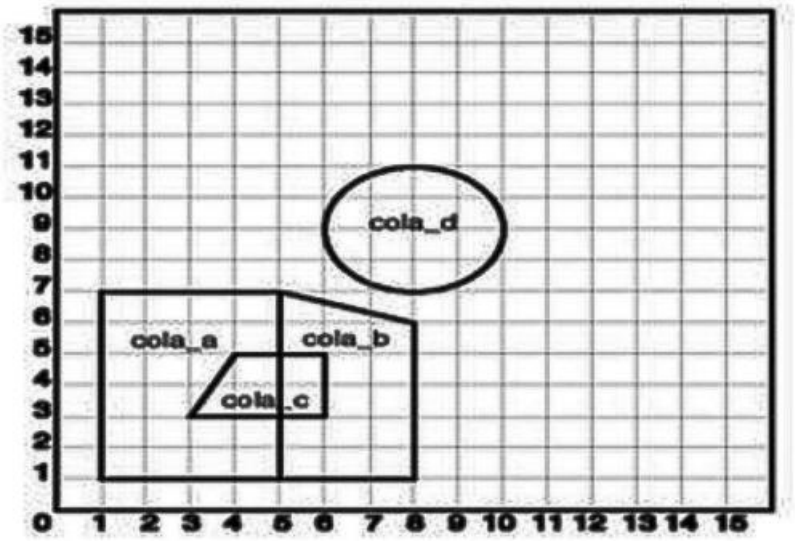
## In general, spatial data can be of two types:

**Vector data:** This data is represented as discrete points, lines and polygons
**Rastor data:** This data is represented as a matrix of square cells.

The spatial data in the form of points, lines, polygons etc. is used by many different databases

## Queries:

1. Find the topological intersection of two geometries
2. Find whether two geometric figures are equivalent to each other
3. Find the areas of all different locations
4. Find the area of only one location
5. Find the distance between two geometries




**Step 1: Create Table:**

SQL> create table areas(
no number(5) primary key, name varchar2(20), location MDSYS.SDO_GEOMETRY);

Table created.

```
SQL> spool 'D:\SADIQ\MSc\SEM 1\ADT\PRAC\sp.txt'
SQL> create table areas(
  2  no number(5) primary key, name varchar2(20), location MDSYS.SDO_GEOMETRY);

Table created.
```

**Step 2: Insert Values into Table:**

SQL> insert into areas
values(1,'rect',MDSYS.SDO_GEOMETRY(2003,null,null,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),MDSYS.SDO_ORDINATE_ARRAY(1,1,5,7)));

1 row created.

SQL> insert into areas
values(2,'poly1',MDSYS.SDO_GEOMETRY(2003,null,null,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),MDSYS.SDO_ORDINATE_ARRAY(5,1,8,1,8,6,5,7,5,1)));

1 row created.

SQL> insert into areas
values(3,'poly2',MDSYS.SDO_GEOMETRY(2003,null,null,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),MDSYS.SDO_ORDINATE_ARRAY(3,3,6,3,6,5,4,5,3,3)));

1 row created.

SQL> insert into areas
values(4,'circle',MDSYS.SDO_GEOMETRY(2003,null,null,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,4),MDSYS.SDO_ORDINATE_ARRAY(8,7,10,9,8,11)));

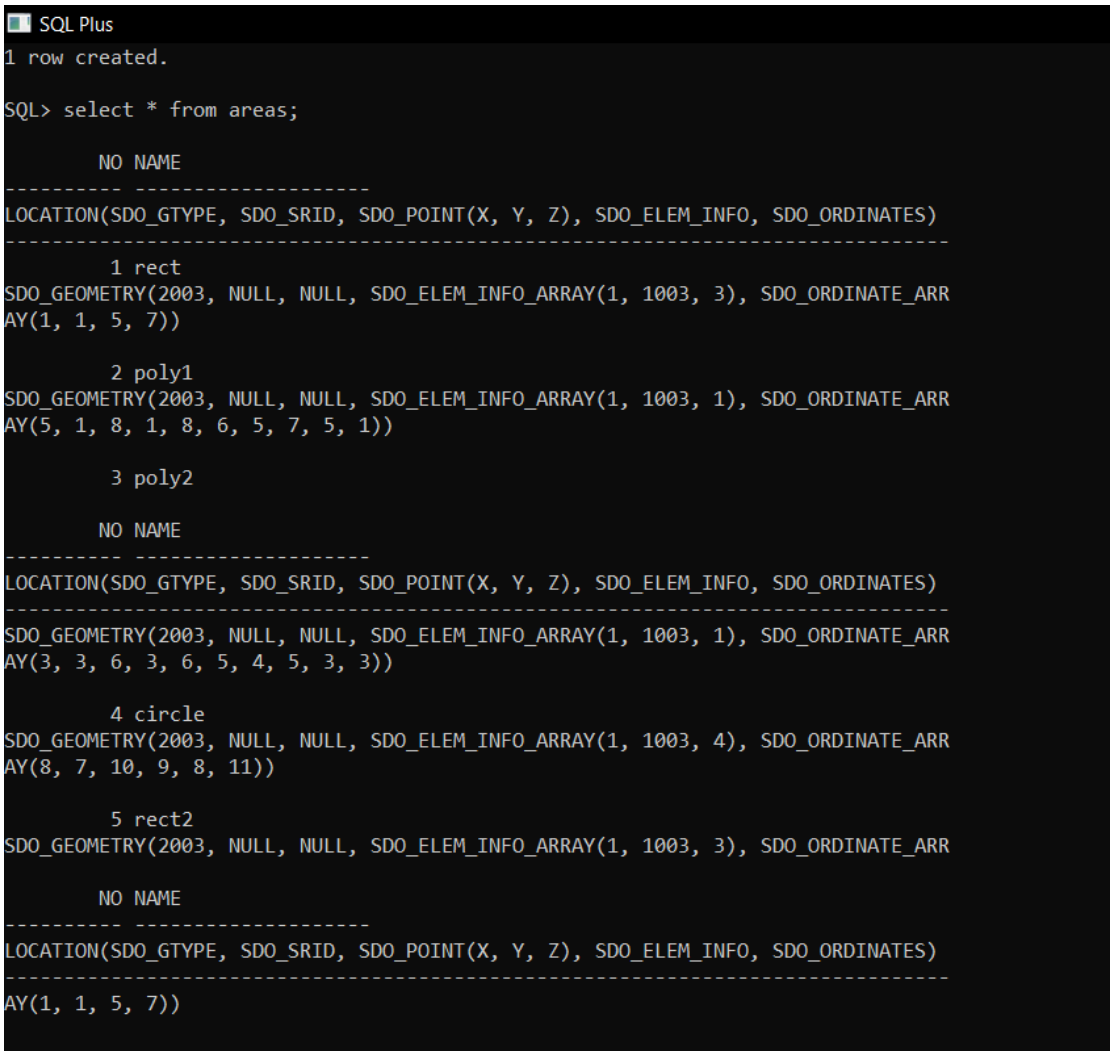
1 row created.

SQL> insert into areas
values(5,'rect2',MDSYS.SDO_GEOMETRY(2003,null,null,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),MDSYS.SDO_ORDINATE_ARRAY(1,1,5,7)));

1 row created.

**Step 3: Display the Values:**

SQL> select * from areas;



## Step 4: Fire the queries:

### 1. Find the topological intersection of two geometries

SQL> select SDO_GEOM.SDO_INTERSECTION (a1.location,a2.location,0.005) from areas a1, areas a2 where a1.name='rect' and a2.name='poly2';

```
SQL> select SDO_GEOM.SDO_INTERSECTION(a1.location,a2.location,0.005) from areas a1, areas a2 where a1.name='rect' and a2.name='poly2';

SDO_GEOM.SDO_INTERSECTION(A1.LOCATION,A2.LOCATION,0.005)(SDO_GTYPE, SDO_SRID, SD
--------------------------------------------------------------------------------
SDO_GEOMETRY(2003, NULL, NULL, SDO_ELEM_INFO_ARRAY(1, 1003, 1), SDO_ORDINATE_ARR
AY(5, 3, 5, 5, 4, 5, 3, 3, 5, 3))
```

### 2. Find whether two geometric figures are equivalent to each other

SQL> select SDO_GEOM.RELATE (a1.location,'anyinteract',a2.location,0.005) from areas a1, areas a2 where a1.name='rect' and a2.name='rect2';

```
SQL>  select SDO_GEOM.RELATE(a1.location,'anyinteract',a2.location,0.005) from areas a1, areas a2 where a1.name='rect' and a2.name='rect2';

SDO_GEOM.RELATE(A1.LOCATION,'ANYINTERACT',A2.LOCATION,0.005)
--------------------------------------------------------------------------------
TRUE
```

### 3. Find the areas of all different locations

SQL> select name,SDO_GEOM.SDO_AREA(location,0.005) from areas;

```
SQL> select name,SDO_GEOM.SDO_AREA(location,0.005) from areas;

NAME                 SDO_GEOM.SDO_AREA(LOCATION,0.005)
-------------------- ---------------------------------
rect                                              24
poly1                                           16.5
poly2                                              5
circle                                     12.5663706
rect2                                             24
```

### 4. Find the area of only one location

SQL> select name,SDO_GEOM.SDO_AREA(a1.location,0.005) from areas a1 where a1.name='rect2';

```
SQL> select name,SDO_GEOM.SDO_AREA(a1.location,0.005) from areas a1 where a1.name='rect2';

NAME                 SDO_GEOM.SDO_AREA(A1.LOCATION,0.005)
-------------------- ------------------------------------
rect2                                                 24
```

## 5. Find the distance between two geometries

SQL> select SDO_GEOM.SDO_DISTANCE(a1.location,a2.location,0.005) from areas a1, areas a2 where a1.name='poly1' and a2.name='circle';

```
SQL> select SDO_GEOM.SDO_DISTANCE(a1.location,a2.location,0.005) from areas a1, areas a2 where a1.name='poly1' and a2.name='circle';

SDO_GEOM.SDO_DISTANCE(A1.LOCATION,A2.LOCATION,0.005)
---------------------------------------------------
                                        .846049894

SQL> spool end;
SQL> commit;

Commit complete.

SQL>
```