Outline

1. What is a Query? Query Language?
2. Example Database Tables
3. SQL Overview: 3 Components
4. SELECT statement with 1 table
5. Multi-table SELECT statements
6. Why spatial extensions are needed?
7. 1-table spatial queries
8. Multi-table spatial queries
9. Trends

Learning Objectives

- After this segment, students will able to
  - Determine output of a SQL/OGIS query with spatial join
  - Compose a SQL/OGIS query with spatial join

Simple SQL SELECT_FROM_WHERE Examples

- Last Video: Spatial analysis operations
  - Unary operator: Area
  - Binary operator: Distance
- This Video
  - Spatial-Join using Topological operations
  - Touch, Cross
  - Using both spatial analysis and topological operations
    - Buffer, within

Spatial Join with Cross()

Query: For all the rivers listed in the River table, find the countries through which they pass.

```
SELECT  R.Name, C.Name
FROM River R, Country C
WHERE Cross(R.Shape,C.Shape) = 1
```

Note: Spatial operation “Cross” is used to join River and Country tables. This query represents a spatial join operation.

Spatial Self-Join with Touch()

Query: Find the names of all countries which are neighbors of the United States (USA) in the Country table.

```
SELECT  C1.Name AS "Neighbors of USA"
FROM Country C1,Country C2
WHERE Touch(C1.Shape,C2.Shape)=1
AND C2.Name ='USA '
```

Note: Spatial operator Touch() is used in WHERE clause to join Country table with itself. This query is an example of spatial self-join operation.

Spatial Join with Within()

Query: The St. Lawrence River can supply water to cities that are within 300 km. List the cities that can use water from the St. Lawrence.

```
SELECT  Ci.Name
FROM City Ci, River R
WHERE Within (Ci.Shape, Buffer (R.Shape, 300)) = 1
AND R.Name = 'St.Lawrence'
```


Spatial Join & Aggregation

Query: List all countries, ordered by number of neighboring countries.

```
SELECT  Co.Name, Count (Co1.Name)
FROM Country Co, Country Co1
WHERE Touch(Co.Shape,Co1.Shape)
GROUP BY Co.Name
ORDER BY Count(Co1.Name)
```

Note: This query is difficult to answer in point-and-click GIS software (e.g. Arc/View) without support for programming languages, e.g., SQL.

Spatial Join with Nesting

Query: For each river, identify the closest city.

```
SELECT C1.Name, R1.Name
FROM City C1, River R1
WHERE Distance (C1.Shape,R1.Shape)
                    <= ALL (
                            SELECT Distance(C2.Shape, R1.Shape)
                            FROM City C2
                            WHERE C1.Name <> C2.Name
                            )
```