

CSEG601 & CSE5601: Spatial Data Management & Application

Sungwon Jung

Bigdata Processing & DB Lab
Dept. of Computer Science and Engineering
Sogang University
Seoul, Korea
Tel: +82-2-705-8930
Email : jungsung@sogang.ac.kr



What is a query?

➊ What is a Query ?

- A query is a “question” posed to a database
- Queries are expressed in a high-level declarative manner
 - Algorithms needed to answer the query are not specified in the query

➋ Examples:

- Mouse click on a map symbol (e.g. road) may mean
 - What is the name of road pointed to by mouse cursor ?
- Typing a keyword in a search engine (e.g. google, yahoo) means
 - Which documents on web contain given keywords?
- SELECT S.name FROM Senator S WHERE S.gender = 'F' means
 - Which senators are female?



What is a query language?

❖ What is a query language?

- ❖ A language to express interesting questions about data
- ❖ A query language restricts the set of possible queries

❖ Examples:

- ❖ Natural language, e.g. English, can express almost all queries
- ❖ Computer programming languages, e.g. Java,
 - can express computable queries
 - however algorithms to answer the query is needed
- ❖ Structured Query Language(SQL)
 - Can express common data intensive queries
 - Not suitable for recursive queries
- ❖ Graphical interfaces, e.g. web-search, mouse clicks on a map
 - can express few different kinds of queries

3



An Example World Database

❖ Purpose: Use an example database to learn query language SQL

❖ Conceptual Model

- ❖ 3 Entities: Country, City, River
- ❖ 2 Relationships: capital-of, originates-in
- ❖ Attributes listed in Figure 3.1

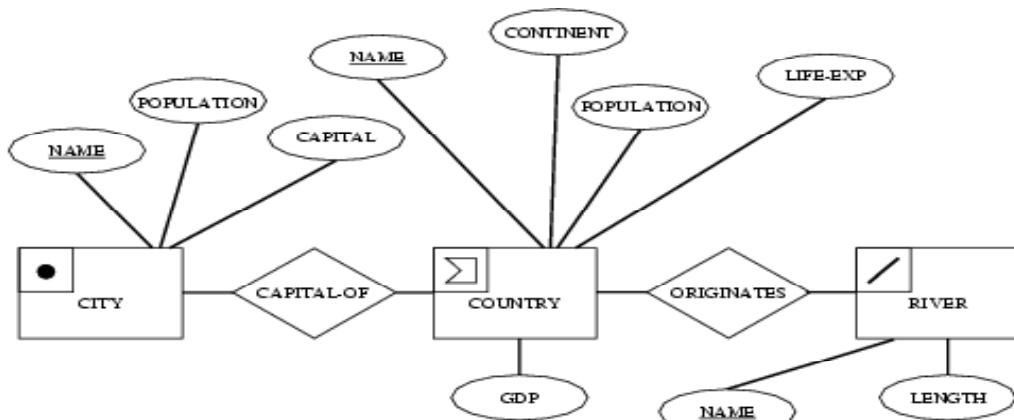
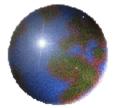


Figure 3.1. The ER diagram of the World database.

4



An Example Database - Logical Model

- 3 Relations

Country(*Name, Cont, Pop, GDP, Life-Exp, Shape*)

City(*Name, Country, Pop, Capital, Shape*)

River(*Name, Origin, Length, Shape*)

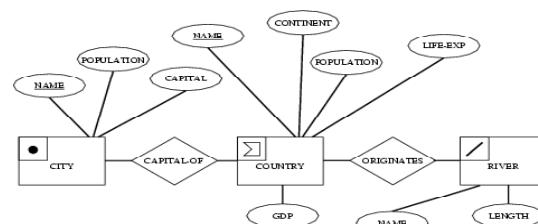
- Keys

- Primary keys are **Country.Name**, **City.Name**, **River.Name**

- Foreign keys are **River.Origin**, **City.Country**

- Data for 3 tables

- Shown on next slide



5



World database data tables

COUNTRY	Name	Cont	Pop (millions)	GDP (billions)	Life-Exp	Shape
Canada	NAM		30.1	658.0	77.08	Polygonid-1
Mexico	NAM		107.5	694.3	69.36	Polygonid-2
Brazil	SAM		183.3	1004.0	65.60	Polygonid-3
Cuba	NAM		11.7	16.9	75.95	Polygonid-4
USA	NAM		270.0	8003.0	75.75	Polygonid-5
Argentina	SAM		36.3	348.2	70.75	Polygonid-6

(a) Country

CITY	Name	Country	Pop (millions)	Capital	Shape
Havana	Cuba		2.1	Y	Pointid-1
Washington, D.C.	USA		3.2	Y	Pointid-2
Monterrey	Mexico		2.0	N	Pointid-3
Toronto	Canada		3.4	N	Pointid-4
Brasilia	Brazil		1.5	Y	Pointid-5
Rosario	Argentina		1.1	N	Pointid-6
Ottawa	Canada		0.8	Y	Pointid-7
Mexico City	Mexico		14.1	Y	Pointid-8
Buenos Aires	Argentina		10.75	Y	Pointid-9

(b) City

RIVER	Name	Origin	Length (kilometers)	Shape
Rio Parana	Brazil		2600	LineStringid-1
St. Lawrence	USA		1200	LineStringid-2
Rio Grande	USA		3000	LineStringid-3
Mississippi	USA		6000	LineStringid-4

(c) River

6



What is SQL?

⊕ SQL - General Information

- is a standard query language for relational databases
- It supports logical data model concepts, such as relations, keys, ...
- Supported by major brands, e.g. IBM DB2, Oracle, MS SQL Server, Sybase, ...
- 3 versions: SQL1 (1986), SQL2 (1992), SQL 3 (1999)
- Can express common data intensive queries
- SQL 1 and SQL 2 are not suitable for recursive queries

⊕ SQL and spatial data management

- ESRI Arc/Info included a custom relational DBMS named Info
- Other GIS software can interact with DBMS using SQL
 - using open database connectivity (ODBC) or other protocols
- In fact, many software use SQL to manage data in back-end DBMS
- And a vast majority of SQL queries are generated by other software
- Although we will be writing SQL queries manually!

7



Three Components of SQL?

⊕ Data Definition Language (DDL)

- Creation and modification of relational schema
- Schema objects include relations, indexes, etc.

⊕ Data Manipulation Language (DML)

- Insert, delete, update rows in tables
- Query data in tables

⊕ Data Control Language (DCL)

- Concurrency control, transactions
- Administrative tasks, e.g. set up database users, security permissions

⊕ Focus for now

- A little bit of table creation (DDL) and population (DML)
- Primarily Querying (DML)

8



Creating Tables in SQL

- Table definition
 - “CREATE TABLE” statement
 - Specifies table name, attribute names and data types
 - Create a table with no rows.
 - See an example at the bottom
- Related statements
 - ALTER TABLE statement modifies table schema if needed
 - DROP TABLE statement removes an empty table

```
CREATE      TABLE      River(  
                      Name       varchar(30),  
                      Origin     varchar(30),  
                      Length     number,  
                      Shape      LineString );
```

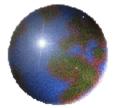
9



Populating Tables in SQL

- Adding a row to an existing table
 - “INSERT INTO” statement
 - Specifies table name, attribute names and values
 - Example:
`INSERT INTO River(Name, Origin, Length) VALUES('Mississippi', 'USA', 6000)`
- Related statements
 - SELECT statement with INTO clause can insert multiple rows in a table
 - Bulk load, import commands also add multiple rows
 - DELETE statement removes rows
 - UPDATE statement can change values within selected rows

10



Querying populated Tables in SQL

- SELECT statement
 - The commonly used statement to query data in one or more tables
 - Returns a relation (table) as result
 - Has many clauses
 - Can refer to many operators and functions
 - Allows nested queries which can be hard to understand
- Scope of our discussion
 - Learn enough SQL to appreciate spatial extensions
 - Observe example queries
 - Read and write simple SELECT statement
 - Understand frequently used clauses, e.g. SELECT, FROM, WHERE
 - Understand a few operators and function

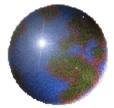
11



SELECT Statement- General Information

- Clauses
 - SELECT specifies desired columns
 - FROM specifies relevant tables
 - WHERE specifies qualifying conditions for rows
 - ORDER BY specifies sorting columns for results
 - GROUP BY, HAVING specifies aggregation and statistics
- Operators and functions
 - arithmetic operators, e.g. +, -, ...
 - comparison operators, e.g. =, <, >, BETWEEN, LIKE...
 - logical operators, e.g. AND, OR, NOT, EXISTS,
 - set operators, e.g. UNION, IN, ALL, ANY, ...
 - statistical functions, e.g. SUM, COUNT, ...
 - many other operators on strings, date, currency, ...

12



SELECT Example 1.

- Simplest Query has **SELECT** and **FROM** clauses

- **Query:** List all the cities and the country they belong to.

SELECT Name, Country FROM CITY

* Result



Name	Country
Havana	Cuba
Washington, D.C.	USA
Monterrey	Mexico
Toronto	Canada
Brasilia	Brazil
Rosario	Argentina
Ottawa	Canada
Mexico City	Mexico
Buenos Aires	Argentina

13



SELECT Example 2.

- Commonly 3 clauses (**SELECT**, **FROM**, **WHERE**) are used

- **Query:** List the names of the capital cities in the CITY table.

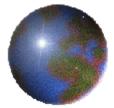
SELECT * FROM CITY WHERE CAPITAL='Y'

* Result



Name	Country	Pop(millions)	Capital	Shape
Havana	Cuba	2.1	Y	Point
Washington, D.C.	USA	3.2	Y	Point
Brasilia	Brazil	1.5	Y	Point
Ottawa	Canada	0.8	Y	Point
Mexico City	Mexico	14.1	Y	Point
Buenos Aires	Argentina	10.75	Y	Point

14



Query Example...Where clause

Query: List the attributes of countries in the Country relation where the life-expectancy is less than seventy years.

```
SELECT Co.Name, Co.Life-Exp  
FROM Country Co  
WHERE Co.Life-Exp <70
```

* Result ↗

Name	Life-exp
Mexico	69.36
Brazil	65.60

Note: use of alias ‘Co’ for Table ‘Country’

15



Multi-table Query Examples

Query: List the capital cities and populations of countries whose GDP exceeds one trillion dollars.

Note: Tables City and Country are joined by matching “City.Country = Country.Name”. This simulates relational operator “join” discussed in 3.2

```
SELECT Ci.Name, Co.Pop  
FROM City Ci, Country Co  
WHERE Ci.Country = Co.Name  
AND Co.GDP >1000.0 AND Ci.Capital='Y'
```

* Result ↗

Ci.Name	Co.Pop
Brasilia	183.3
Washington, D.C.	270.0

16



Multi-table Query Example

Query: What is the name and population of the capital city in the country where the St. Lawrence River originates?

```
SELECT Ci.Name, Ci.Pop  
FROM City Ci, Country Co, River R  
WHERE R.Origin =Co.Name  
AND Co.Name = Ci.Country AND R.Name = 'St.Lawrence '  
AND Ci.Capital='Y '
```

Note: Three tables are joined together pair at a time. River.Origin is matched with Country.Name and City.Country is matched with Country.Name. The order of join is decided by query optimizer and does not affect the result.

17



Query Examples...Aggregate Statistics

Query: What is the average population of the noncapital cities listed in the City table?

```
SELECT AVG(Ci.Pop)  
FROM City Ci  
WHERE Ci.Capital='N '
```

Query: For each continent, find the average GDP.

```
SELECT Co.Cont, Avg(Co.GDP) AS Continent-GDP  
FROM Country Co  
GROUP BY Co.Cont
```

18



Query Example..Having clause, Nested queries

Query: For each country in which at least two rivers originate, find the length of the smallest river.

```
SELECT R.Origin, MIN(R.length) AS Min-length  
FROM River  
GROUP BY R.Origin  
HAVING COUNT(*) > 1
```

Query: List the countries whose GDP is greater than that of Canada.

```
SELECT Co.Name  
FROM Country Co  
WHERE Co.GDP > ANY(SELECT Co1.GDP  
                 FROM Country Co1  
                 WHERE Co1.Name = 'Canada' )
```

19



3.4 Extending SQL for Spatial Data

- Motivation
 - SQL has simple atomic data-types, like integer, dates and string
 - Not convenient for spatial data and queries
 - Spatial data (e.g. polygons) is complex
 - Spatial operation: topological, euclidean, directional, metric
- SQL3 allows user defined data types and operations
 - Spatial data types and operations can be added to SQL3
- Open Geodata Interchange Standard (OGIS)
 - Half a dozen spatial data types
 - Several spatial operations
 - Supported by major vendors, e.g. ESRI, Intergraph, Oracle, IBM,...

20



OGIS Spatial Data Model

- Consists of base-class Geometry and four sub-classes:
 - Point, Curve, Surface and GeometryCollection
 - Figure 2.2 (pp. 27) lists the spatial data types in OGIS
- Operations fall into three categories:
 - Apply to all geometry types
 - SpatialReference, Envelope, Export, IsSimple, Boundary
 - Predicates for Topological relationships
 - Equal, Disjoint, Intersect, Touch, Cross, Within, Contains
 - Spatial Data Analysis
 - Distance, Buffer, Union, Intersection, ConvexHull, SymDiff
 - Table 3.9 (pp. 66) details spatial operations

21



```
CREATE TABLE Country(
    Name    varchar(30),
    Cont    varchar(30),
    Pop     Integer,
    GDP     Number,
    Shape   Polygon);
```

(a)

```
CREATE TABLE River(
    Name    varchar(30),
    Origin  varchar(30),
    Length  Number,
    Shape   LineString);
```

(b)

```
CREATE TABLE City (
    Name      varchar(30),
    Country   varchar(30),
    Pop       integer,
    Shape     Point );
```

(c)

22

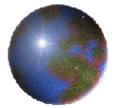


TABLE 3.9: A Sample of Operations Listed in the OGIS Standard for SQL [OGIS, 1999]

Basic Functions	SpatialReference()	Returns the underlying coordinate system of the geometry
	Envelope()	Returns the minimum orthogonal bounding rectangle of the geometry
	Export()	Returns the geometry in a different representation
	IsEmpty()	Returns true if the geometry is a null set
	IsSimple()	Returns true if the geometry is simple (no self-intersection)
Topological/ Set Operators	Boundary()	Returns the boundary of the geometry
	Equal	Returns true if the interior and boundary of two geometries are spatially equal
	Disjoint	Returns true if the boundaries and interior do not intersect
	Intersect	Returns true if the geometries are not disjoint
	Touch	Returns true if the boundaries of two surfaces intersect but the interiors do not
	Cross	Returns true if the interior of a surface intersects with a curve
	Within	Returns true if the interior of the given geometry does not intersect with the exterior of another geometry
	Contains	Tests if the given geometry contains another given geometry
	Overlap	Returns true if the interiors of two geometries have nonempty intersection
Spatial Analysis	Distance	Returns the shortest distance between two geometries
	Buffer	Returns a geometry that consists of all points whose distance from the given geometry is less than or equal to the specified distance
	ConvexHull	Returns the smallest convex geometric set enclosing the geometry
	Intersection	Returns the geometric intersection of two geometries
	Union	Returns the geometric union of two geometries
Difference	Difference	Returns the portion of a geometry that does not intersect with another given geometry
	SymmDiff	Returns the portions of two geometries that do not intersect with each other

23



Spatial Queries with SQL/OGIS

- SQL/OGIS - General Information
 - Both standard are being adopted by many vendors
 - The choice of spatial data types and operations is similar
 - Syntax differs from vendor to vendor
 - Readers may need to alter SQL/OGIS queries given in text to make them run on specific commercial products
- Using OGIS with SQL
 - Spatial data types can be used in DML to type columns
 - Spatial operations can be used in DML
- Scope of discussion
 - Illustrate use of spatial data types with SQL
 - Via a set of examples

24



List of Spatial Query Examples

- Simple SQL SELECT_FROM_WHERE examples
 - Spatial analysis operations
 - Unary operator: Area
 - Binary operator: Distance
 - Boolean Topological spatial operations - WHERE clause
 - Touch
 - Cross
 - Using spatial analysis and topological operations
 - Buffer, overlap
- Complex SQL examples
 - Aggregate SQL queries
 - Nested queries

25



Using spatial operation in SELECT clause

Query: List the name, population, and area of each country listed in the Country table.

```
SELECT C.Name, C.Pop, Area(C.Shape) AS "Area"  
FROM Country C
```

Note: This query uses spatial operation, Area(). Note the use of spatial operation in place of a column in SELECT clause.

26



Using spatial operator Distance

Query: List the GDP and the distance of a country's capital city to the equator for all countries.

```
SELECT Co.GDP, Distance(Point(Ci.Shape.x, 0), Ci.Shape) AS "Distance"  
FROM Country Co, City Ci  
WHERE Co.Name = Ci.Country  
AND Ci.Capital = 'Y'
```

* Result →

Co. Name	Co. GDP	Dist-to-Eq (in Km).
Havana	16.9	2562
Washington, D.C.	8003	4324
Brasilia	1004	1756
Ottawa	658	5005
Mexico City	694.3	2161
Buenos Aires	348.2	3854

27



Using Spatial Operation in WHERE clause

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.

28



Spatial Query with multiple tables

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.

Exercise: Modify above query to report length of river in each country.

29



Example Spatial Query...Buffer and Overlap

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 Overlap(Ci.Shape, Buffer(R.Shape,300)) = 1  
AND R.Name = 'St.Lawrence'
```

Note: This query uses spatial operation of Buffer, which is illustrated in Figure 3.2

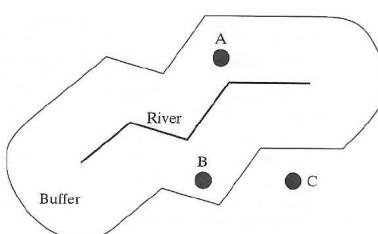
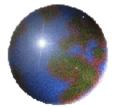


FIGURE 3.2. The buffer of a river and points within and outside.

30



Recall List of Spatial Query Examples

- Simple SQL SELECT_FROM_WHERE examples
 - Spatial analysis operations
 - Unary operator: Area
 - Binary operator: Distance
 - Boolean Topological spatial operations - WHERE clause
 - Touch
 - Cross
 - Using spatial analysis and topological operations
 - Buffer, overlap
- Complex SQL examples
 - Aggregate SQL queries
 - Nested queries

31



Using spatial operation in an aggregate query

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)
```

Notes: This query can be used to differentiate querying capabilities of simple GIS software (e.g. ArcView) and a spatial database. It is quite tedious to carry out this query in GIS.

Earlier version of OGIS did not provide spatial aggregate operation to support GIS operations like reclassify.

32



Using Spatial Operation in Nested Queries

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 )
```

Note: Spatial operation Distance used in context of a nested query.

Exercise: It is interesting to note that SQL query expression to find smallest distance from each river to nearest city is much simpler and does not require nested query. Audience is encouraged to write a SQL expression for this query.

33



Answer to the preceding exercise

Query: Find the smallest distance from each river to nearest city

```
SELECT R1.Name, MIN(Distance(R1.Shape, C1.Shape))  
FROM River R1, City C1  
GROUP BY R1.Name
```

34



Nested Spatial Query

Query: List the countries with only one neighboring country. A country is a neighbor of another country if their land masses share a boundary. According to this definition, island countries, like Iceland, have no neighbors.

```
SELECT Co.Name  
FROM Country Co  
WHERE Co.Name IN (SELECT Co.Name  
                  FROM Country Co, Country Co1  
                  WHERE Touch(Co.Shape, Co1.Shape)  
                  GROUP BY Co.Name  
                  HAVING Count(*)=1)
```

```
SELECT Co.Name  
FROM Country Co, Country Co1  
WHERE Touch(Co.Shape, Co1.Shape)  
GROUP BY Co.Name  
HAVING Count(Co1.name) = 1
```

Note: It shows a complex nested query with aggregate operations. Such queries can be written into two expression, namely a view definition, and a query on the view. The inner query becomes a view and outer query is run on the view. This is illustrated in the next slide.

35

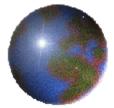


Rewriting nested queries using Views

- Views are like tables
 - Represent derived data or result of a query
 - Can be used to simplify complex nested queries
 - Example follows: Which country has the maximum number of neighbors ?

```
CREATE VIEW Neighbor AS  
  SELECT Co.Name, Count(Co1.Name) AS num_neighbors  
  FROM Country Co, Country Co1  
  WHERE Touch(Co.Shape, Co1.Shape)  
  GROUP BY Co.Name  
  
  SELECT Co.Name, num_neighbors  
  FROM Neighbor  
  WHERE num_neighbor = (SELECT Max(num_neighbors)  
                        FROM Neighbor )
```

36



Summary

- ◆ Queries to databases are posed in high level declarative manner
- ◆ SQL is the “lingua-franca” in the commercial database world
- ◆ Standard SQL operates on relatively simple data types
- ◆ SQL3/OGIS supports several spatial data types and operations