JOURNAL OF DATA WAREHOUSING VOLUME 6 NUMBER 3 SUMMER 2001

Picture This! A Spatially Aware Data Warehouse

Rafael Coss, Michael Gonzales, and Kathryn Zeidenstein

Abstract

Existing business processes can be enhanced by spatial analysis. Most businesses already have huge amounts of implicit location-based data in their data warehouses. With minimal investment in tools and a spatially enabled RDBMS, spatial analysis becomes a reality. When geography is added as an analysis option, you not only discover more information with common tools such as SQL or OLAP, but you now can use tools to visualize your information and understand the relationships that exist just below conventional numeric data.

The modern spatial information management (SIM) industry was born from the geographic information systems (GIS) of the early 1980s. And, as part of its evolution, spatial analysis is no longer the exclusive domain of application niches such as petroleum, utilities, and government agencies. Instead, SIM has integrated GIS components into leading DBMS products such that related capabilities are now available natively. For individuals skilled in a data warehouse that is considered non-GIS, it is now much easier to exploit the technology and include GIS-type processing as a natural part of the data warehouse. Although the integration of spatial analysis into general business processes is still in its infancy, advanced solutions such as logistics, marketing, and planning are finding acceptance because of their richer informational content and use as an incomparable competitive weapon. As tool vendors continue to integrate spatial components into their products, spatial data will take its place, over time, as a core data dimension not unlike customer, products, and time.

Introduction

What do we mean by spatial awareness? The answer starts with an understanding of spatial data. Any location-based data is considered spatial, for example: customer address, store or branch locations, sales zones, delivery routes, the scene of accidents, etc. It is estimated that 80 percent of all data stored in computers has some sort of geographic reference (Daratech, 2000). In other words, businesses already collect mountains of spatial data and use it frequently in their day-to-day data processing, such as recording information about customer transactions. As a matter of fact, although never referred to as spatial data, the level of attention spent by organizations to capture, collect, and maintain customer information makes customer relationship management (CRM) one area in which spatial awareness makes a real difference.

By transforming the implicit location data we record to explicit geographic locations, the typical analysis already done for analyzing customer habits and buying patterns is now enhanced to exploit the link to real, physical locations, not to mention the link to the rest of the people via demographic data (Sonnen and Morris, 2000). This transformation from implicit to explicit location information requires the assignment of a geographic location (usually an encoded latitude and longitude value). This simple geographical assignment to your existing customer address allows you to start analyzing and mining the spatial relationships between your customers, stores, product movements, and so on. Moreover, once you have coded your addresses, you can bind demographic data available from third-party suppliers such as Dunn & Bradstreet or Urban Data Systems, and start exploiting the rich analytical landscape of relationships, patterns, and trends not seen otherwise.

There are many industries that rely heavily on geo-spatial data and on traditional geographic information systems (GIS). For instance, the utilities industry uses spatial data and visualization tools for surveying, pipeline maintenance, line routing, and preventative maintenance. Unfortunately, members of the IT community and business users often perceive GIS as being extremely complex. This misconception is born from the history of GIS and its traditional niche applications. More often than not, a narrow definition that is applied to GIS is that it is simply an automated map management system, which only propagates the attitude that GIS is a niche application. It is true to say that GIS does involve expanding your knowledge in terms of visualization tools, data formats, and

terminology; however "Business GIS is much simpler than other applications of the technology for two reasons:

1. GIS is now an integral part of established RDBMS into the relational database, an area of familiarity for business IT professionals.

2. The precision requirements of general business applications are less stringent.

Business GIS is therefore easier to implement than in the past, but the analytical worth is not compromised.

Traditional GIS goes beyond the scope of this article. Moreover, organizations that currently rely on GIS already understand the importance of integrating their spatial data with their major business processes (Sonnen, 1999). Instead, this article focuses on IT professionals building data warehouses and more importantly, those of IT particularly interested in delivering the promise of warehousing, which is to make information content and a rich analytical environment available to the enterprise.

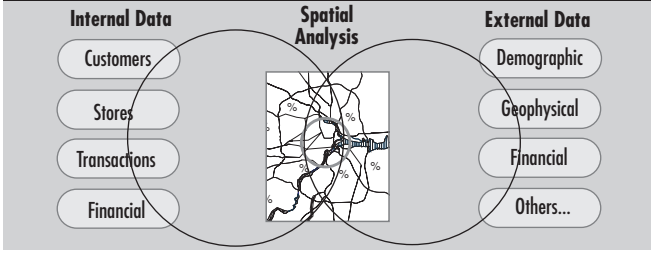
Specifically, this article addresses the issues related to integrating spatial data into existing business processes in terms understandable to the new or casual user of geo-spatial data and its applications. We describe in more detail the benefits of spatial analysis, give a brief history of GIS, and discuss the technical challenges involved in integrating spatial into the data warehouse.

Why Go Spatial?

The effect of "going spatial" can be astonishing, especially if you have never viewed your data in that context. Ralph Kimball makes a good case for the advantages of picking the "low hanging fruit that our GIS colleagues have generously provided for us (Kimball, 2001 [2])." Spatial visualization provides a new way of analyzing data. You can see on a map where your most valuable customers live in relation to where your stores are located. More importantly, by importing demographic data, you can find out where other people with a similar profile to your valuable customers live. The windfall for spatial analysis is discovering potential business and understanding the relationships, patterns, and trends that often go unnoticed without visual cues.

Spatially enabled demographic data creates a broad analytical landscape. For example, it can tell you the percent of the population that is 65 or older, the percent of population that never married, the average household size or the percent of population with a college education. But the value doesn't stop there. You can also import and blend nontraditional data sources into the warehouse, including: government boundaries such as tax records, census data, municipal data, and geophysical data, or meteorological data among others. This type of data allows analysis such as finding crime data or diaster data, such as the 100-year and 500-year flood plain areas and fault lines.[1]

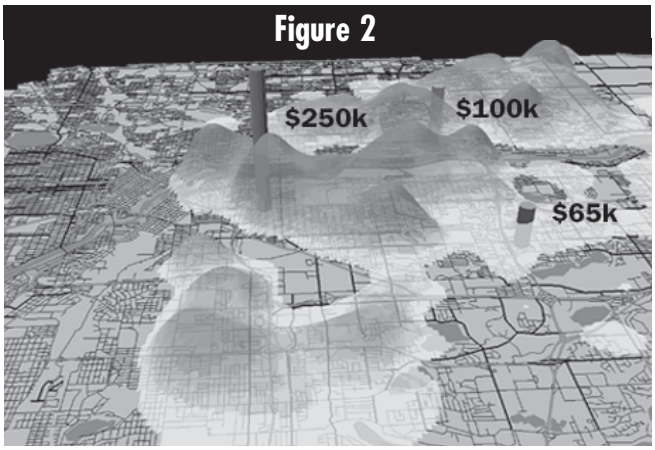
Figure 1



Sources of Spatial Data

When you enable your business data for spatial analysis, the imported data and your business data are like the equation: **1 + 1 = 4**. Information about your sales revenue is far more interesting when visualized in context of the purchaser's demographics. Supply chain management is more interesting when viewed in context of the routes taken from manufacturing to the location of the customer. Spatial awareness can actually help you make more intelligent business decisions, such as store or branch locations so that profitability is maximized. For illustration, let's examine the spatial analysis output shown in Figure 2. The chart identifies three stores using cylinders, each shaded to show sales: Medium gray represents the store experiencing $250k in sales, light gray is for the store achieving $100k in sales and dark gray for the store generating only $65k. Also shown on the chart is the surrounding population density of customers for each of the stores. What is made obvious visually is that the store that generates the most revenue is surrounded by the highest customer density, and the store with $100k in sales has somewhat less surrounding customer density. But the real discovery is that the store with the least sales is also the one whose surrounding population density is painfully low.

[1] *http://www.geographynetwork.com is a portal to all sorts of geographic content. You can search on data by location or topic and the search results includes relevant information about the various data sets that meet your search, including the coverage area, contact information, cost information, and other relevant information to make the data usable.*



Visualizing Population Density and Store Sales

So, why go spatial? Well, if you were the district manager and did not have this spatial data and analysis capability, you might think that the store manager is to blame for the suffering sales. But, in fact, the best manager in the world may not be able to make a difference in this situation because there is simply not enough customer population to support more sales. Given the spatial insight, a district manager may decide to relocate the store to an area that will support greater sales as opposed to having a revolving door for managers coming in and out, trying to squeeze blood out of a turnip. This type of insight would certainly not be visible given traditional warehouse data and access tools such as OLAP.

With a spatially enabled DBMS, like DB2® Spatial Extender, Informix® Spatial DataBlade®, and the spatial option for Oracle® 8i, you can perform spatial analysis directly using SQL, and return the results in a tabular form.

Spatial analysis can help you answer questions such as the following:

- What is the closest retail outlet for all of the customers who have spent more than $3,000 dollars during 2000 within the Chicago area.
- Within the Los Angeles key market, identify the under-performing locations within a 10-minute driving time of a competitor's location.
- How many accidents that occurred within .5 kilometers of a highway exit damaged the front bumper of red cars?
- Identify customers with home insurance policies living within 1,000 yards of a creek who do not have a flood insurance option.

By joining your own business data with spatial data, you can produce visual displays to answer these questions. Spatial analysis provides a new, human-friendly way of understanding relationships. In addition, your existing business reports can be enhanced with spatial SQL predicates.

What is Spatial Data?

Conceptually, spatial data is any location-based data: addresses, zones, parcels, roads, census blocks, and so on. To enable spatial awareness, use a process called "geocoding" to convert your textual, location-based data into another form that is recognizable by spatial visualization and analysis tools and by a spatially enabled RDBMS. Geocoding will generally represent spatial data as a latitude/longitude coordinate or as a series of coordinates, although it is possible to represent spatial data using coordinate systems that are not based on latitude and longitude, such as kilometers. Depending on the feature being modeled, locations can be represented by points, line strings (for representing rivers, roads, or networks), or polygons (for representing parcels, lakes, zones, or anything that takes up area).

GIS Moves into the Mainstream

History shows that GIS functionality is moving from being a stand-alone system for GIS specialists toward being fully integrated into the mainstream RDBMS world. This section briefly describes the architectural transformation that is enabling the promotion of spatial analysis to a fundamental part of basic business processes.

First-Generation GIS

Traditional GIS systems have their own special software and use conventional file systems as their data store. Access to the GIS system is through a proprietary API and is generally restricted to specialized GIS applications.[2] If you wanted to perform spatial analysis on your business data, you have to extract data from your RDBMS and move it into the GIS data store. Alternatively, your application could connect to a table and set up a manual join, which you have to maintain.[3]

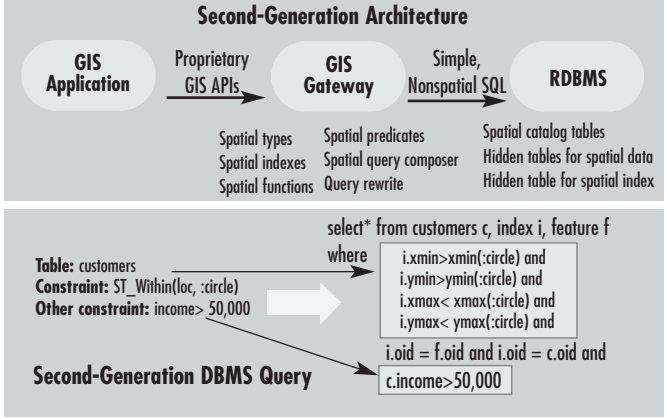
Second-Generation GIS

In recent years, GIS solutions have started to use relational databases as their data store to take advantage of the client/server architecture, high availability and data concurrency, and other data management features. One example of these "second-generation" systems is IBM's GeoManager and the Spatial Database Engine (SDE®) from ESRI®.

[2] *Existing GIS applications will still work, but through a much smaller, simpler GIS gateway.*
[3] *See http://www.opengis.org*

In these cases, however, knowledge of spatial data and operations is maintained in a gateway that is accessible only through the proprietary interface. Applications that want to include location-based information must use the proprietary API and cannot be developed using the wealth of development tools, frameworks, and environments available for relational database systems, making the mixing of spatial analysis and traditional numeric analysis very difficult.

Figure 3



Second-Generation GIS

The traditional RDBMSs used in this architecture do not have support for complex types such as geometries. For this reason, the spatial data must be modeled in the relational model by a series of columns that represent a geometry (integers, floats, blobs, etc.). To enhance performance, a series of indexes on primitive relational types are created and are interpreted by the gateway to simulate a spatial index. Because the database is not spatially aware, it cannot accept spatial predicates and it cannot optimize spatial queries.

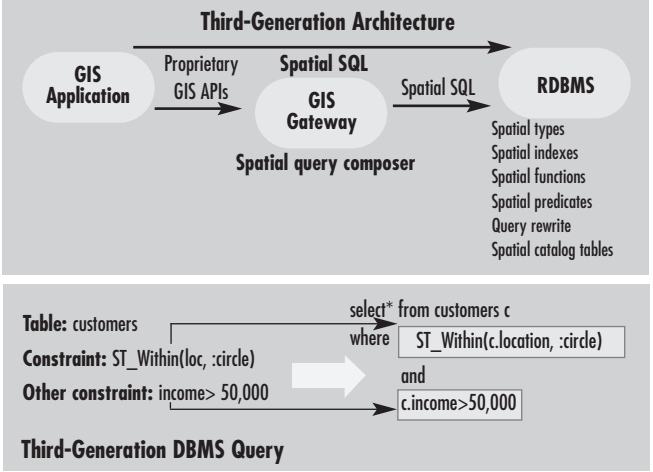
In addition, there can be integrity problems that arise, as the GIS must do the work of keeping spatial side tables in sync with the business data. In addition, because the database does not have semantic understanding of the spatial data and cannot perform semantic checks, it is possible for data integrity problems to be an issue.

Third-Generation GIS

The next generation of spatial data is the integration of knowledge about location within the database engine. The ability to move spatial functionality into the database engine is made possible by object-relational technology, such as user-defined types and user-defined functions and methods. Object-relational DBMSs (ORDBMSs) have the modeling capabilities that let you avoid complex mapping between objects and structures, thus providing the opportunity for better performance on complex data (Barry, 2000). Other advantages of this "third-generation" architecture include:

- Applications can use spatially aware SQL directly against the database.
- The optimization features of the database can be used to improve the performance of spatial queries, and native spatial indexes can also be exploited by the optimizer.
- Location information is encapsulated in a single column rather than being spread across multiple columns or in additional side tables. This simplifies data management significantly and removes one source of integrity problems.
- The database understands the semantics of the spatial data and can thus perform semantic checking, reducing the chance of introducing data integrity problems from ill-behaved applications.

Figure 4



Third-Generation GIS

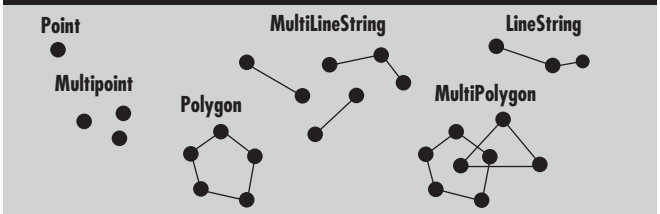
For the easiest and best performing integration into a data warehouse, the third-generation, object-relational DBMS (ORDBMS) is the best choice in most cases. Examples of the latest generation spatial data systems include DB2 Spatial Extender, Informix Spatial DataBlade, and Oracle Spatial 8i.

Characteristics of a Spatially Enabled ORDBMS

The database upon which these third-generation systems support the following:

Spatial data types. Spatial data types are usually pretty easy to understand because they are based on geometry. They include point, line, polygon, and so on. The spatial data types and functions are standardized by the Open GIS Consortium (OGC)[3] and in the SQL standard (SQLMM 1999).

Figure 5



Some Spatial Types

Spatial functions. The set of standardized functions includes distance, overlaps, intersects within, and many others.

A spatial index technique. Traditional B+ trees do not work well for point, lines, and polygons because they are multidimensional values and B+ trees work best for scalar values against which ranges can be defined. To address the problem of spatial indexing, different database vendors exploit different indexing techniques. DB2's Spatial Extender uses a grid index. Oracle Spatial 8i uses a regular region quad tree index and, as a separate feature, also supports R-trees (region trees). Informix also supports R-trees.

Naturally, the database optimizer must be aware of spatial as well, because these functions, types, and indexes will have different performance characteristics, and different statistics, than normal SQL scalar expressions.

Integration with DBA tools. The database administration tools must also know about spatial and provide spatial utilities—such as the ability to import and export to and from special file types—files where the spatial data is encoded.[4] Ideally, there should be administrative support for geocoders and spatial reference systems, tuning guidance for spatial indexes, and replication support.

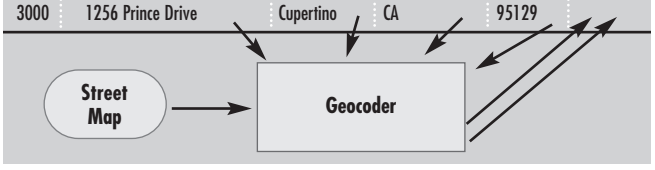
Geocoding support. Geocoding is the process of taking location-based data, most likely an address, and calculating a point. Ideally, the ORDBMS should support an open architecture that allows different geocoders to be "plugged in." The next section describes why you may need different geocoders and other issues surrounding geocoding. Third-generation GIS systems offer advantages for geocoding so you can do automatic, trigger-based geocoding, which in turn can be optimized using the ORDBMS's optimizer.

Turning Addresses into Spatial Data

To generate explicit locations from the business data's implicit locations, you must use a process called geocoding. Geocoding takes a text address as input and translates that into a pair of coordinates, such as latitude/longitude. In the third-generation system, those coordinates are stored in a column of the table that contains the business data. For example, as shown in Figure 6, the customer's table contains a column named location. The values in that column are derived from geocoding the address columns in the table. This location data for customers is considered a "layer" from a mapping perspective. To translate from the database perspective, a layer is any spatial column in a relational table that can be visualized by a geobrowser. For example, another layer might be the location data for stores. Another layer might include major roads. This concept of layers is important to understand, because when you build a map using tools like ESRI's ArcView® GIS from ESRI and MapInfo Professional® from MapInfo® that map consists of layers that are stacked on top of each other. This is how you can actually see on a map which major roads (one layer) connect your customers (another layer) and your stores (another layer).

Figure 6

Cid	Address	City	State	Zip	Location
1	4335 Queen Anne Drive	San Jose	CA	95129	
2	555 Bailey Avenue	San Jose	CA	95120	
3000	1256 Prince Drive	Cupertino	CA	95129	



Geocoding Customer Addresses

Of course, you can also use SQL functions such as ST_Touches and ST_Within to analyze and return the information in a tabular format.

[4] *"Shapefile" is a defacto industry standard from ESRI for such an interchange file form but others exist as well.*

Getting the Most from Geocoding

When the geocoder reads a record of source data, it tries to match that record with a counterpart in the reference data shipped with the geocoder. For example, if you are geocoding U.S. addresses, the geocoder contains a comprehensive list of U.S. addresses (based on a U.S. postal service standard), which it cross-references with the input data that you provide. This cross-reference list is sometimes called a base map. The match between your data and the base map must be accurate to a certain degree in order for the geocoder to process the record. For example, a precision of 85 means that the match between a source record and its counterpart in the reference data must be at least 85 percent accurate in order for the source record to be processed. You might need to adjust the precision for various reasons. For example, if you specify 100 but your input addresses are from a recently built subdivision that is not included in the reference data, your geocoder will reject what may be perfectly good addresses.

The quality of the geocoded point depends on two factors: (1) the quality of the base map, and (2) the quality of your address. Be sure that your address data is clean before inputting it to the geocoder. In a typical data warehouse, name and address cleansing is already a normal step in the data cleansing process. Luckily, many name and address cleansing products also have associated geocoders, making it easy to add a step to geocode the cleansed data.

Choosing a Geocoder

Geocoders are provided with GIS software, with a spatially aware ORDBMS, and as stand-alone products, often associated with a name and address cleansing product. Depending on what country you are in or to what level of precision you need, you may have to use a different geocoder than the one that comes with your RDBMS or GIS tools. A geocoder that only includes a U.S. base map won't do you any good if your location-based data refers to European addresses. In addition, different base maps have different levels of precision. For example, there are street base maps and parcel base maps. A parcel base map is more accurate, but if the scale of your analysis is regional or national, you probably don't need that level of detail. On the other hand, if you are an insurance company analyzing whether a particular property is in a flood zone, you might need the greater level of precision provided by the parcel base map. In either case, choose the right geocoder for the type of analysis you want to do.

Considerations and Challenges

The fact that integrating spatial processes into the data warehouse is not yet a common warehouse practice means that everyone who does the work to add spatial awareness to their data warehouse is ahead of the curve, technologically, from their counterparts. No doubt, there will be some initial outlay of work to integrate the geocoding process and the analysis tools. Although vendors are working to integrate spatial more and more into the tools they ship, users that continue to work ahead of the curve will gain the competitive advantage from their efforts. This section describes some things to think about when considering how to enable your warehouse for spatial analysis.

Data Type Complexity

Although not required by the OGC specification, the ability to model the complete set of spatial data types can require object-relational extensions in the database. These object-relational extensions are required to conform to the SQL99 spatial standard. Different vendors use different ways of representing the geometry data types. For example, DB2 uses the SQL99 structured type and Informix uses their opaque data types. These data types offer the encapsulation and type safety that helps to guard against data integrity problems, but because these complex types are not yet mature in the market, there can be challenges integrating the data seamlessly with other more traditional tools and for exchanging spatial data.

To replicate spatial data, one workaround is to convert spatial data types to the OGC well known text or well known binary representation. This is currently the solution offered by IBM DB2. Oracle has published a solution that is based on using read only replicas (ORA 2000). Informix has said that an upcoming release of their product will include built-in replication of spatial data.

Because of these technical issues, there may be situations where it is best to keep the spatial data (the geocoded data, that is) in a separate table. For example, you may want customers as a dimension in your data warehouse, but you may not be able to load the geocoded data into a multidimensional cube. (Not to mention the fact that the problem of how to aggregate spatial data has not yet been solved.) If you need to drill down to a specific customer and visualize that customer location on a map, you can write some glue code to join by a customer ID field with the geocoded data before passing to the visualizing tool.

Integrating the Geocoding Step

Geocoding is the key to spatial enablement and can be treated in some ways like other data transformation steps. Geocoders can work in batch mode or as a row of data is inserted. If you are geocoding addresses in a large table of existing customer data, you will probably need to:

1. Add a spatial column to the customer table to contain the spatial data.
2. Register the spatial column for geocoding, indicating which columns (e.g., number, street, city, state) are to be input to the geocoder.
3. Run the geocoder in batch mode to update the location columns with the geocoded data.

After the initial geocoding, you can then indicate that you want geocoding to occur only on insert or update. This process is controlled by insert and update triggers on the data.

Think about how and when you want geocoding to take place. For example, if you are planning on launching a new CRM application that lets your service personnel track a location on a map while on the phone with the customer, you very well might want to geocode on the operational side in real time and carry the geocoded data forward into the data warehouse, which may require some conversion to the OGC well known text or binary format.

Otherwise, you probably want to do batch geocoding as a separate transformation step, either in a staging area or after the data is loaded into the warehouse. You may also need to build iterative steps into the geocoding process to allow for the possibility of investigating any problems in generating points, which might be

of either spatial types or spatial queries. However, the major GIS vendors are providing analysis tools that that do such things as:

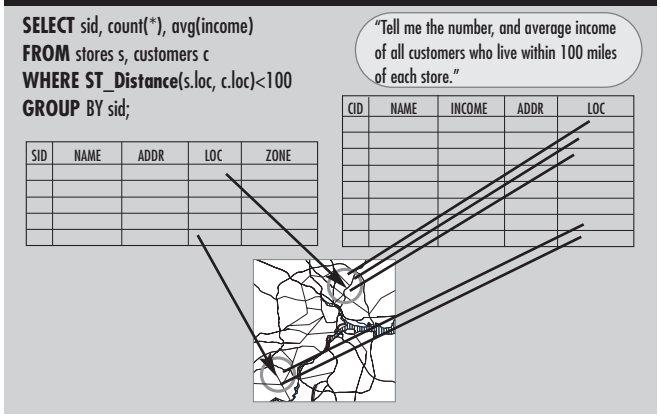
- Analyze market areas and customer profiles to determine market potential and to create prospect lists
- Create ring analysis for locations
- Generate desire lines from your locations
- Conduct drive time analysis
- Produce detailed reports

Examples of such tools are ArcView™ Business Analyst from ESRI, and MapInsight™, from Mapinfo®.

Right now, although these spatial analysis tools can understand basic SQL, it is not possible to pass complex SQL (subqueries, recursions, the latest SQL OLAP functions, and so forth) to these tools. To work around this problem, you can create views in the DBMS using the complex SQL and then run the analysis tools on the views.

Also, as we mentioned before, you can write spatially oriented queries yourself, and return the results in a tabular format. Not only are spatial functions available, but you can also combine the spatial functions with other SQL analytic capabilities. Some examples of SQL queries:

Figure 7



Simple Spatial Query

Example 1: Finding Zones

The following query finds the average customer distance from each department store. The spatial functions used in this example are ST_Distance and ST_Within (the SQL spatial standard names for these functions):

```
SELECT s.id, AVG(ST_Distance(c.location,s.location))
FROM customers c, stores s
WHERE ST_Within(c.location,s.zone)=1
GROUP BY s.id
```

Example 2: Converting Spatial Data to Text Format

The following query finds the customer locations for those who live in the San Francisco Bay Area and converts it to the OGC well known text representation. The spatial functions used in this example are ST_AsText and ST_Within:

```
SELECT ST_AsText(c.location,cordref(1))
FROM customers c
WHERE ST_Within(c.location,:BayArea)=1
```

Example 3: Creating Searchable Zones

This query finds the customers who live within the flood zone or within two miles from the boundary of the flood zone. The spatial functions used in this example are ST_Buffer and ST_Within:

```
SELECT c.name,c.phoneNo,c.address
FROM customers c
WHERE ST_Within(c.location,ST_Buffer(:floodzone,2))=1
```

In the future, as spatial technology continues to move into mainstream business processes, there will be more integration of analytical tools and spatial awareness. This integration will happen both ways; in other words, spatial analysis tools will most likely pass on the SQL they receive to the DBMS and visualize the results, and tools that understand business data will integrate spatial visualization capabilities.

Summary

Although not without technical challenges, the capability of performing spatial analysis and visualization on business data is a huge opportunity to gather new business insights. Combining spatial analysis with the historical perspective provided by the data warehouse is a winning

combination by enabling you to model your business better (by including spatial data), so that you can understand your current business better, and so that you can plan your future business opportunities.

REFERENCES

Barry and Associates. **Lack of Impedance Mismatch**, Barry and Associates, 2000.

Dana, Peter H. **Coordinate Systems Overview**. The Geographer's Craft Project, Department of Geography, The University of Colorado at Boulder, 1999. http://www.colorado.edu/geography/gcraft/notes/coordsys/coordsys_f.html.

Daratech. **Geographic Information Systems Markets and Opportunities**. Daratech, Inc., 2000.

Gonzales, Michael L. "Spatial OLAP: Conquering Geography," **DB2 Magazine**, Vol. 4, No. 1 (Spring, 1999).

Gonzales, Michael L. "Seeking Spatial Intelligence," **Intelligent Enterprise**, Vol. 3, No. 2 (January 20, 2000).

ISO International Standard (IS), Information Technology - Database Languages - SQL Multimedia and Application Packages - Part 3: Spatial, ISO/IEC 13249-3: 1999, December 1999.

Kimball, Ralph. "Spatial Enabling Your Data Warehouse," **Intelligent Enterprise Magazine—Data Webhouse**, Vol. 4, No. 1 (January 1, 2001).

Kimball, Ralph. "Address Space." **Intelligent Enterprise Magazine—Data Webhouse**, Vol. 4, No. 2 (January 30, 2001).

Mattos, Nelson M. and Kathryn Zeidenstein. "Integrating Spatial Data with Business Data," **DB2 Magazine**, Vol. 4, No. 1 (Spring 1999).

OpenGIS Consortium, Inc. OpenGIS® Simple Features Specification for SQL, Revision 1.1, OpenGIS Project Document 99-049, May 1999.

Oracle Corporation. "Read-Only Replication of Tables Containing Objects." **Oracle Technical White Paper**. Oracle Corp, 2000.

Sonnen, David, Henry Morris, and Jacqueline Sweeney. **IDC Report: 1999 Worldwide Spatial Information Management Markets and Trends**, International Data Corporation, 1999.

Sonnen, David and Henry Morris. "Linking Virtual Information to the Real World." **International Data Corporation**, 2000.

BIOGRAPHY

Rafael Anselmo Coss *is the DB2 Spatial Extender Architect from IBM's Silicon Valley Lab. He has worked in the GIS arena since 1992, and in the AM/FM and public sector arenas. His background includes both civil and software engineering. He provides a unique insight into spatial information systems by combining his GIS knowledge gained while working for the City of San Luis Obispo, CA with his IBM DB2 object-relational database expertise.*

Rafael Anselmo Coss
408.463.5219
rcoss@us.ibm.com

Michael L. Gonzales *manages a consulting firm called The Focus Group, Ltd., specializing in business intelligence related techniques and technologies. He offers practical knowledge involved in building and managing data warehouses across a variety of industries and business applications. He has a BBA, MA, and MSSE (2002), is a successful author, and conducts data warehouse seminars nationally.*

Michael L. Gonzales
915.542.2604
mlg@starfocus.com

Kathryn Zeidenstein *has been with IBM's Silicon Valley Lab since 1987. She has worked in DB2 for OS/390, database standards, business intelligence, and, most recently, information integration strategy and standards.*

Kathryn Zeidenstein
408.463.3697
krzeide@us.ibm.com