

Comparative Analysis of Leading Vendor Spatial Databases

Prashant Nadagoud

Senior Software Engineer, Workday, California, USA

Abstract This paper provides a comprehensive comparative analysis of leading vendor spatial databases, including Microsoft SQL Server 2022, IBM DB2, Oracle Spatial, and PostGIS. The study delves into the architecture, spatial data types, indexing techniques, and query models employed by each database. It evaluates the performance and capabilities of these databases in managing spatial data, highlighting their strengths and weaknesses in various scenarios. Additionally, the paper explores the practical aspects of spatial data storage, retrieval, and manipulation, offering insights into the ease of use, licensing costs, and community support for each platform. Through detailed comparisons, the research aims to guide practitioners and decision-makers in selecting the most suitable spatial database solution based on their specific requirements.

Keywords Spatial Databases

1. Introduction

Spatial Databases are specialized database systems designed to store, query, and manage spatial data, which includes geographic and geometric information such as points, lines, and polygons. These databases are integral to various applications, including Geographic Information Systems (GIS), location-based services, urban planning, environmental monitoring, and more. As the importance of spatial data continues to grow, so does the need for robust and efficient spatial database systems.

The primary focus of this paper is to provide a comprehensive comparison of the leading vendor spatial databases: Microsoft SQL Server 2022, IBM DB2, Oracle Spatial, and PostGIS. These databases are widely used in the management of spatial data, each offering unique features, performance characteristics, and levels of support for spatial operations.

The paper begins with an overview of spatial databases, highlighting their fundamental concepts, including spatial data types, spatial indexing, and the query models used to perform spatial operations. It then delves into the specific features of each database, comparing their strengths and weaknesses in various aspects such as indexing techniques, data types, query performance, and support for spatial relationships.

By examining these leading spatial databases, the paper aims to provide valuable insights for practitioners,

developers, and decision-makers, helping them to choose the most appropriate spatial database technology based on their specific needs and the unique requirements of their spatial data management tasks.

2. Spatial Database

A Spatial Database [1] is a specialized type of database designed to efficiently store and query data related to spatial objects, such as points, lines, and polygons. Standard databases can handle various numeric and character data types, but additional functionality is required to manage spatial data types, commonly referred to as geometry or features. The Open Geospatial Consortium (OGC) established the Simple Features Specification and sets the standards for integrating spatial capabilities into database systems.

2.1. Features of Spatial Databases

Database systems use indexes to quickly look up values and the manner that most databases index data is not optimal for spatial queries. So, spatial databases use a spatial index to improve database operations.

In Addition to typical SQL queries such as SELECT statements, spatial databases can perform a wide variety of spatial operations. Open Geospatial Consortium (OGC) supports the following query types and many more:

- **Spatial Measurements:** Finds the distance between points, polygon area, etc.
- **Spatial Functions:** Modify existing features to create new ones, for example by providing a buffer around them, intersecting features, etc.
- **Spatial Predicates:** Allows true/false queries such as 'is

* Corresponding author:

nadagoud.prashant@gmail.com (Prashant Nadagoud)

Received: Jul. 23, 2024; Accepted: Aug. 25, 2024; Published: Sep. 10, 2024

Published online at <http://journal.sapub.org/computer>

there a residence located within a mile of the area we are planning to build the landfill?'

- **Constructor Functions:** Creates new features with an SQL query specifying the vertices (points of nodes), which can make up lines. If the first and last vertices of a line are identical the feature can also be of the type polygon (a closed line).
- **Observer Functions:** Queries, which return specific information about a feature such as the location of the center of a circle.

2.2. Spatial Data

Spatial data describes the location, shape, and orientation of objects in space. These objects can be tangible, like an office building, a valley, or a street, or they can be abstract, such as an imaginary line delineating a border between countries.

2.3. Uses of Spatial Data

- Examining regional, national, or international sales trends.
- Taking up a decision such as where to place a new store based on vicinity to customers and competitors.
- Navigating using a Global Positioning System (GPS) device.
- Tracking the delivery of a parcel by the customers.

2.4. Representing Features on the Earth

In the real world, objects on the earth often have complex, uneven shapes. It would be very hard for any item of spatial data to define the exact shape of these objects. So, spatial data represents these objects by using simple, geometrical shapes that approximate their actual shape and position. These shapes are called geometries.

Figure 2.1 shows the different types of geometries that can be used to represent the features on the earth.

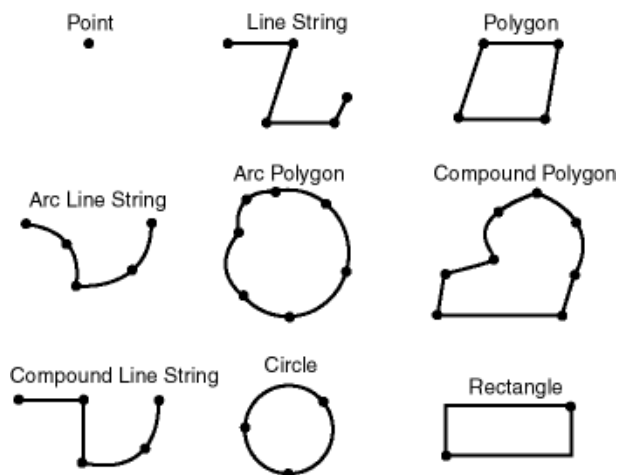


Figure 2.1. Geometry types

2.5. Spatial Data Model

It is a hierarchical structure consisting of elements, geometries, and layers, which correspond to representations

of spatial data. Layers are composed of geometries, which in turn are made up of elements.

For example, a point might represent a building location, a line string might represent a road or flight path, and a polygon might represent a state, city, zoning district, or city block.

2.5.1. Element

An element is the basic building block of geometry. The supported spatial element types are points, line strings, and polygons. For example, elements might model star constellations (point clusters), roads (line strings), and county boundaries (polygons). Each coordinate in an element is stored as an X, Y pair.

- Point data consists of one coordinate.
- Line data consists of two coordinates representing a line segment of the element.
- Polygon data consists of coordinate pair values, one vertex pair for each line segment of the polygon.

2.5.2. Geometry

A geometry (or geometry object) is the demonstration of a spatial feature, modeled as a structured set of primitive elements. It can consist of a single element, which is an instance of one of the supported primitive types, or a homogeneous or heterogeneous collection of elements.

2.5.3. Layer

A layer is a collection of geometries having identical feature sets. For example, one layer in a GIS might include topographical features, while another describes population density, and a third describes the network of roads and bridges in the area (lines and points). Each layer's geometries and associated spatial index are stored in the database in standard tables.

2.5.4. Coordinate System

A coordinate system (also called a spatial reference system) is a way of assigning coordinates to a location and creating relationships between sets of such coordinates. It enables the interpretation of a set of coordinates as a representation of a position in a real world space.

Each spatial data has a coordinate system associated with it. The coordinate system can be of type georeferenced (related to a specific representation of the Earth) or not georeferenced (that is, Cartesian, and not related to a specific representation of the Earth).

Spatial data can be associated with a Cartesian, geodetic (geographical), projected, or local coordinate system:

- Cartesian coordinates are coordinates that measure the position of a point from a defined origin along axes that are perpendicular in the represented two-dimensional or three-dimensional space.
- Geodetic coordinates (sometimes called geographic coordinates) are angular coordinates (longitude and latitude), closely related to spherical polar coordinates,

and are defined relative to a particular Earth geodetic datum. (A geodetic datum is a means of representing the figure of the Earth and is the reference for the system of geodetic coordinates.)

- Projected coordinates are planar Cartesian coordinates that result from performing a mathematical mapping from a point on the Earth's surface to a plane.
- Local coordinates are Cartesian coordinates in a non-Earth (non-georeferenced) coordinate system. Local coordinate systems are often used for CAD applications and local surveys.

2.6. Types of Spatial Data Model

In GIS, there are two basic spatial data models representing the real world:

2.6.1. Raster Model

In the raster data model [2], land cover is represented as single square cells. Figure 2.2 depicts the raster data model.

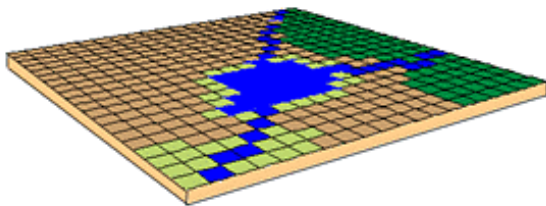


Figure 2.2. Raster data model

Raster data are good at:

- Representing continuous data (e.g., slope, elevation)
- Representing multiple feature types (e.g., points, lines, and polygons) as single feature types (cells).
- Rapid computations ("map algebra") in which raster layers are treated as elements in mathematical expressions.
- Analysis of multi-layer or multivariate data (e.g., satellite image processing and analysis).

2.6.2. Vector Data

In the vector data model, features of the earth are represented as

- Points
- Lines/routes
- Polygons/regions
- TINs (Triangulated Irregular Networks)

Figure 2.3. depicts the vector data model.

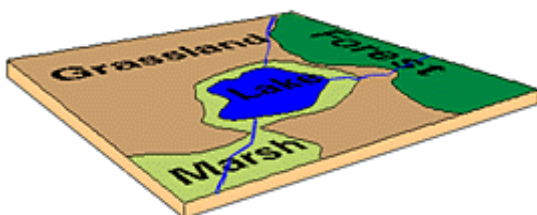


Figure 2.3. Vector data model

Vector data are good at:

- Accurately representing true shape and size
- Representing non-continuous data (e.g., rivers, political boundaries, road lines, mountain peaks)

2.7. Query Model

Spatial databases use a two-tier query model to determine spatial queries and spatial joins. The term two-tier is used to indicate that two separate operations are performed to resolve queries. The output of the two combined operations returns the exact result set.

The two operations are referred to as primary and secondary filter operations.

- The primary filter permits fast selection of candidate records to pass along to the secondary filter. The primary filter compares geometry approximations to reduce computation complexity and is considered a lower-cost filter. Because the primary filter compares geometric approximations, it returns a superset of the exact result set.
- The secondary filter applies exact computations to geometries that result from the primary filter. The secondary filter produces a correct answer to a spatial query. The secondary filter operation is computationally expensive, but it is only applied to the primary filter results, not the entire data set. Figure 2.4 depicts the two-tier query model used in the spatial queries.

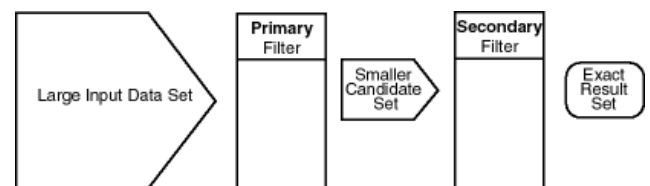


Figure 2.4. Query model

3. Adding Spatial Support to IBM DB2 and Microsoft SQL Server 2022

3.1. IBM DB2 Spatial Extender

DB2 spatial extender provides the ability to generate, analyze & exploit spatial information about geographic features, such as the locations of office buildings or the size of an earthquake zone.

- DB2 Spatial Extender extends the function of DB2 Universal Database with a set of advanced spatial data types that represent geometries such as points, lines, and polygons and many functions and features that interoperate with those new data types. These capabilities make it possible to integrate spatial information with business data, adding another element of intelligence to the database.
- DB2 Spatial Extender implements types and functions defined by ISO SQL/MM and Open GeoSpatial Consortium (OGC) specifications, allowing customers

to leverage the power of SQL for spatial data analysis.

DB2 database from IBM can be downloaded from [3], and the spatial extender for the database can be downloaded from [4]. It is a free download.

3.2. SQL Server 2022 Spatial

Microsoft SQL Server Express Edition, which is a free download of SQL Server 2022, can be downloaded from [5].

4. Spatial Data Types

4.1. IBM DB2 Spatial Data Types

When you enable a database for spatial operations, DB2 Spatial Extender supplies the database with a hierarchy of structured data types. Figure 4.1 presents this hierarchy.

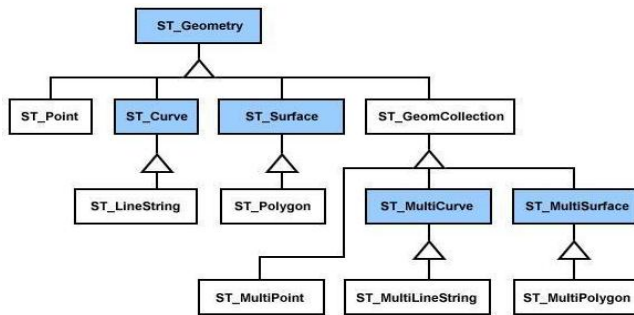


Figure 4.1. Hierarchy of spatial data types of DB2

The hierarchy in the Figure 4.1 includes:

- Data types for geographic features that can be perceived as forming a single unit; for example, individual residences and isolated lakes.
- Data types for geographic features that are made up of multiple units or components; for example, canal systems and groups of islands in a lake.
- A data type for geographic features of all kinds.

4.2. SQL Server 2022 Data Types

SQL Server 2022 [6] introduces two new data types specifically intended to hold spatial data: geography and geometry.

- Geography: used for geodetic vector spatial data.
- Geometry: Planar vector spatial data.

Although both data types can be used to store spatial data, they are distinct from each other and are used in different ways. Whenever you define an item of spatial data in SQL Server 2022, you must also choose whether to store that information using the geometry data type or the geography data type.

There are several similarities between the two spatial data types:

- They can both represent spatial information using a range of geometries – Points, LineStrings, and Polygons.
- Internally, both data types store spatial data as a stream of binary data in the same format.

- When working with items of data from either type, you must use object-oriented methods based on the .NET framework.
- They both implement many of the same standard spatial methods to analyze and perform calculations on data of that type.

However, there are also a number of important differences between the two spatial data types as outlined in the below table. You must choose the appropriate data type to reflect how you plan to use spatial data in your database. See Table 4.1 for the major differences between Geometry and Geography data type.

Table 4.1. Differences between Geometry and Geography Data Type

Property	Geometry Data Type	Geography Data Type
Shape of the earth	Flat	Round
Coordinate system	Projected (or natural planar)	Geographic
Coordinate values	Cartesian (x and y)	Latitude and Longitude
Unit of Measurement	Same as coordinate values	Defined in sys.spatial_reference_systems
Spatial reference identifier	Not enforced	Enforced
Default SRID	0	4326(WGS84)
Size limitations	None	No object may occupy more than one hemisphere
Ring orientation	Not significant	Significant

4.3. Creating a Table in SQL Server 2022

There are no special attributes or features required to enable spatial data to be stored in a SQL Server database – all that is required is a table that contains at least one geography or geometry column. Since both the geography and geometry column types are already registered, system defined data types; you can use a normal T-SQL CREATE TABLE statement to create a table containing a field of data type geography or geometry as follows.

Example 4.3:

```

CREATE TABLE [dbo].[Cities] (
    [CityName] [varchar] (255) NOT NULL,
    [CityLocation] [geometry] NOT NULL,
    [StateName] [varchar] (255) NOT NULL,
    [CountryName] [varchar] (255) NOT NULL,
)
GO
  
```

Example 4.3 creates a table containing four columns—CityName, which can hold a 255-character variable-length string, and CityLocation, which can be used to hold the spatial data relating to that city, using the geometry data type, StateName, which can hold a 255-character variable-length string and CountryName which can hold a 255-character variable-length string.

4.4. Creating a Table in DB2

In order to perform spatial operations in DB2 [7], we must first enable the database for spatial operations.

To enable a database for spatial operations from the Control Center:

- From the Control Center window, expand the object tree until you find the Databases folder under the server where you want Spatial Extender to run.
- Click the Databases folder. The databases are displayed in the contents pane on the left side of the window.
- Right-click the database that you want, and click Spatial Extender —Enable in the pop-up menu. Spatial Extender supplies the database with the resources that allow you to create and work with spatial columns and data.

4.5. Inserting Spatial Data into SQL Server 2022 Table

Example 4.4:

```
INSERT into Cities.dbo.Customer VALUES (
    'Bob',
    'Hillary',
    'Miramar',
    97778,
    'USA',
    Geometry::STGeomFromText('POLYGON((0 1, 1 7,
    7 6, 0 1))', 4326))
```

4.6. Inserting Spatial Data into IBM DB2 Table

Example 4.5:

```
INSERT INTO GEO_SPAT.Customer VALUES(
    'Bob',
    'Henry',
    'Miramar',
    'USA',
    9212,
    db2gse.ST_Geometry('polygon ((52 28,58 28,58 23,52
    23,52 28))', 1))
```

5. Methods Provided by Databases to Modify Spatial Objects

5.1. DB2

DB2 provides a whole set of methods that can be used to modify the spatial objects.

Table 5.1 summarizes the different methods provided by DB2. The entire detailed set of examples and syntaxes of different methods can be referred to at the DB2 tutorials provided by IBM [8].

5.2. SQL Server 2022

Similar to DB2, SQL Server 2022 provides a large set of methods that can be used to modify the spatial objects. Table 5.2 provides a brief summary of different methods provided by SQL Server 2022 to modify spatial objects. For further details of these methods refer [9].

6. Methods for Testing Spatial Relationships

When examining spatial information, frequently there is a need to understand the relationship between two or more features on earth – for example, how far is the point a from point b? Does the route between x and y pass through z? Does m and n share a common border? Spatial databases provide methods that can be used to answer these questions, by comparing different aspects of the relationship between two items of spatial data. The general syntax of all of these methods is that the instance to which a comparison is being made is provided as a parameter to a method acting upon the first instance; for example:

```
Instance1.Method (Instance2)
```

Table 5.1. Methods Provided by DB2 to Modify Spatial Objects

SI No	Function	Syntax	Return Type
1	ST_IsValid	db2gse.ST_IsValid(geometry)	Integer
2	ST_Union	db2gse.ST_Union (geometry1, geometry2)	db2gse.ST_geometry
3	ST_Intersection	db2gse.ST_Intersection(geometry1, geometry2)	db2gse.ST_geometry
4	ST_Difference	db2gse.ST_Difference (geometry1, geometry2)	db2gse.ST_geometry

Table 5.2. SQL Server 2022 Methods to Modify Spatial Objects

SI No	Function	Syntax	Return Type
1	STIsValid	.STIsValid()	Integer
2	MakeValid	.MakeValid()	Geography
3	STUnion	.STUnion (other_geometry)	Geography
4	STIntersection	.STIntersection (other_geometry)	Geography
5	STDifference	.STDifference (other_geomtery)	Geography

Table 6.1. DB2 Methods to Test Relationships between Different Spatial Objects

SI No	Function	Syntax	Return Type
1	ST_Equals()	db2gse.ST_Equals(geometry1, geometry2)	Integer
2	ST_Distance()	db2gse.ST_Distance(geometry1, geometry2, unit)	Double
3	ST_Intersects()	db2gse.ST_Intersects(geometry1, geometry2)	Integer
4	ST_Disjoint()	db2gse.ST_disjoint(geometry1, geometry2)	Integer
5	ST_Contains()	db2gse.ST_Contains (geometry1, geometry2)	Integer

Table 6.2. SQL Server 2022 Methods to Test Relationships between Different Spatial Objects

SI No	Function	Syntax	Return Type
1	STEquals()	.STEquals (other_geometry)	bit
2	STDistance()	.STDistance (other_geometry)	float
3	STIntersects()	.STIntersects (other_geometry)	bit
4	Filter()	.Filter (other_geometry)	bit
5	STDisjoint()	.STDisjoint (other_geometry)	bit
6	STCrosses()	.STCrosses (other_geometry)	bit

6.1. DB2

DB2 provides different methods that can be used to test the relationship between different spatial objects. Table 6.1 provides a brief summary of the methods provided by DB2, which can be used to test relationships between different spatial objects. For further details about these methods, including examples and syntaxes refer [8].

6.2. SQL Server 2022

On similar lines of DB2, SQL Server 2022 also provides a set of methods, which can be used to test relationships between different spatial objects. Table 6.2 summarizes the different methods provided by SQL Server 2022 to test relationships between different spatial objects. For further details of these methods including the examples and syntax refer [9].

7. Methods to Examine Properties of Spatial Objects

There are many different questions that arise about any individual item of spatial data: Where is it? How big is it? What sort of object is it? Where does it start and end? Where does its center lie? For all these questions and more, spatial databases provide different methods that can be used to get answers for these questions. The general syntax that is required to access any property of an instance is:

Instance.Property

Here, Instance is the name of the column or variable containing the information that can be queried, and Property

is the name of the particular property in question.

In some cases, certain aspects of information about an item are not strictly properties of that item, but rather the result of a method, in which case the appropriate syntax is

Instance.Method()

7.1. DB2

DB2 provides different methods that can be used to examine the properties of spatial objects. Table 7.1 provides a brief summary of different methods, which can be used to examine the properties of spatial objects. For further details of these methods including their syntax and examples refer [8].

DB2 also provides a set of separate methods that can be used to return the coordinate values of a point. Table 7.2 summarizes the different methods, which can be used to return the coordinate values of a point. For further details of these methods refer [8].

7.2. SQL Server 2022

Similar to DB2, SQL Server 2022 provides different methods, which can be used to examine properties of spatial objects. Table 7.3 summarizes the different methods provided by SQL Server 2022 to examine the properties of spatial objects. For further details of these methods, including examples and syntax refer [9].

Similar to DB2, SQL Server 2022 provides methods that can be used to return the coordinate values of a point. Table 7.4 summarizes the methods that can be used to return the coordinate values of the points. For further details of these methods refer [9].

Table 7.1. DB2 Methods to Examine Properties of Spatial Objects

SI No	Method	Description
1	ST_GeometryType()	This method takes a geometry as input parameter and returns the fully qualified type name of the geometry
2	ST_Dimension()	This method takes a geometry as an input parameter and returns its dimension
3	ST_IsSimple()	This method takes a geometry as an input parameter and returns 1 if the given geometry is simple. Otherwise, 0 is returned
4	ST_IsClosed()	This method takes a curve or multcurve as an input parameter and returns 1 if the given curve or multcurve is closed. Otherwise, 0 is returned
5	ST_IsRing()	This method takes a curve as an input parameter and returns 1 if it is a ring. Otherwise, 0 is returned.
6	ST_NumPoints()	This method takes geometry as an input parameter and returns the number of points that were used to define that geometry.

Table 7.2. DB2 Methods to Return the Coordinate Values of a Point

SI No	Method	Description
1	ST_X ()	This method takes a point as an input parameter and returns its X coordinate
2	ST_Y ()	This method takes a point as an input parameter and returns its Y coordinate
3	ST_Z ()	This method takes a point as an input parameter and returns its Z coordinate
4	ST_M ()	If a measure is stored with a given point, then this method can take the point as an input parameter and return the measure.

Table 7.3. SQL Server 2022 Methods to Examine Properties of Spatial Objects

SI No	Method	Description
1	STGeometryType()	Returns the name of the type of the geometry (e.g., Point, LineString)
2	InstanceOf()	Tests whether an instance is of a particular geometry type.
3	STDimension()	Returns the number of dimensions an instance occupies
4	STIsSimple()	Determines whether an instance is simple
5	STIsClosed()	Determines whether an instance is closed
6	STIsRing()	Determines whether an instance is a ring
7	STNumPoints()	Returns the number of points in an instance

Table 7.4. SQL Server 2022 Methods to Return Coordinate Values of a Point

SI No	Property	Description
1	STX	Returns the x coordinate of a geometry Point instance
2	STY	Returns the y coordinate of a geometry Point instance
3	Lat	Returns the latitude of a geography Point instance
4	Long	Returns the longitude of a geography Point instance
5	M	Returns the m(measure) coordinate of a geometry instance
6	Z	Returns the z(elevation) coordinate of a Point instance

8. Spatial Indexes

Spatial indexing is a technique used in database systems and Geographic Information Systems (GIS) to efficiently manage and query spatial data. Spatial data refers to information about the location, shape, and relationships between objects in space, such as points, lines, and polygons.

Good query performance is related to having efficient indexes defined on the columns of the base tables in a database. The performance of the query is directly related to how quickly values in the column can be found during the query. Queries that use an index can execute more quickly and can provide a significant performance improvement.

Spatial queries are typically queries that involve two or more dimensions. For example, in a spatial query it might be worthy to know if a point is included within an area (polygon). Due to the multidimensional nature of spatial queries, spatial indices are used by spatial databases to optimize spatial queries. Indexes used by non-spatial databases cannot effectively handle features such as how far two points differ and whether points fall within a spatial area of interest.

8.1. DB2

DB2 Spatial Extender's indexing technology utilizes grid indexing [10], which is designed to index multi-dimensional

spatial data, to index spatial columns. DB2 Spatial Extender provides a grid index that is optimized for two-dimensional data on a flat projection of the Earth.

Spatial Extender generates a spatial grid index using the Minimum Bounding Rectangle (MBR) of geometry. For most geometries, the MBR is a rectangle that surrounds the geometry.

A spatial grid index divides a region into logical square grids with a fixed size that you specify when you create the index. The spatial index is constructed on a spatial column by making one or more entries for the intersections of each geometry's MBR with the grid cells. An index entry consists of the grid cell identifier, the geometry MBR, and the internal identifier of the row that contains the geometry.

You can define up to three spatial index levels (grid levels). Using several grid levels is beneficial because it allows you to optimize the index for different sizes of spatial data.

8.2. SQL Server 2022

SQL Server 2022 uses the B-Tree indexing technique, to create the spatial indexes.

8.2.1. B-Tree

A B-tree [11] is a tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions in logarithmic time. The B-tree is a generalization of a binary search tree in that a node can have more than two children. Unlike self-balancing binary search trees, the B-tree is optimized for systems that read and write large blocks of data. It is commonly used in databases and file systems. Figure 8.1 [11] depicts an example of B-Tree.

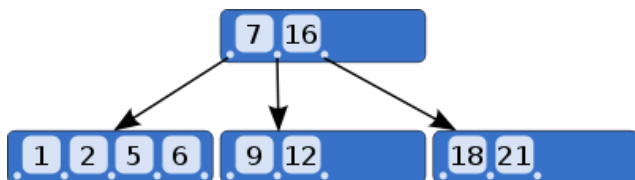


Figure 8.1. B-tree. Source: Wikipedia. B-Tree
<http://en.wikipedia.org/wiki/B-tree>

Since the indexes are created using B-trees [12], the indexes must represent the 2-dimensional spatial data in the linear order of B-trees. Therefore, before reading data into a spatial index, SQL Server 2022 implements a hierarchical uniform decomposition of space. The index-creation process decomposes the space into a four-level grid hierarchy. These levels are referred to as level 1 (the top level), level 2, level 3, and level 4.

Each successive level further decomposes the level above it, so each upper-level cell contains a complete grid at the next level. On a given level, all the grids have the same number of cells along both axes (for example, 4x4 or 8x8), and the cells are all one size.

Figure 8.2 [12] shows the decomposition of the upper-right cell at each level of the grid hierarchy into a 4x4 grid. In reality, all the cells are decomposed in this way. Thus, for example, decomposing a space into four levels of 4x4

grids actually produces a total of 65,536 level-four cells.

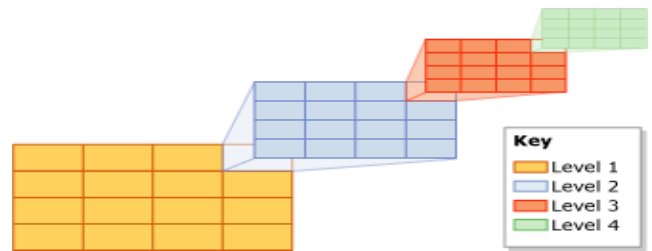


Figure 8.2. <http://technet.microsoft.com/en-us/library/bb964712.aspx>

The cells of a grid hierarchy are numbered in a linear fashion by using a variation of the Hilbert space-filling curve. For the purpose of illustration, however, this discussion uses a simple row-wise numbering, instead of the numbering that is actually produced by the Hilbert curve.

Figure 8.3 [12], shows several polygons that represent buildings and lines that represent streets have already been placed into a 4x4, level-1 grid. The level-1 cells are numbered from 1 through 16, starting with the upper-left cell.

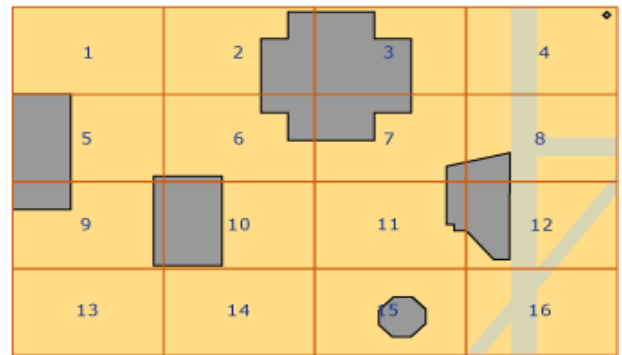


Figure 8.3. <http://technet.microsoft.com/en-us/library/bb964712.aspx>

8.2.2. Tessellation

Tessellation [12] is the process of creating a two-dimensional plane using the repetition of a geometric shape with no overlaps and no gaps.

After decomposition of an indexed space into a grid hierarchy, the spatial index reads the data from the spatial column, row by row. After reading the data for a spatial object (or instance), the spatial index performs a tessellation process for that object. The tessellation process fits the object into the grid hierarchy by associating the object with a set of grid cells that it touches (touched cells). Starting at level 1 of the grid hierarchy, the tessellation process proceeds breadth first across the level. Potentially, the process can continue through all four levels, one level at a time.

The output of the tessellation process is a set of touched cells that are recorded in the spatial index for the object. By referring to these recorded cells, the spatial index can locate the object in space relative to other objects in the spatial column that are also stored in the index.

8.2.3. Tessellation Rules

To limit the number of touched cells that are recorded for an object, the tessellation process applies several tessellation

rules. These rules determine the depth of the tessellation process and which of the touched cells are recorded in the index.

These rules are as follows:

- The covering rule--If the object completely covers a cell, that cell is said to be covered by the object. A covered cell is counted and is not tessellated. This rule applies at all levels of the grid hierarchy. The covering rule simplifies the tessellation process and reduces the amount of data that a spatial index records.
- The cells-per-object rule--This rule enforces the cells-per-object limit, which determines the maximum number of cells that can be counted for each object, except on level 1. At lower levels, the cells-per-object rule controls the amount of information that can be recorded about the object.
- The deepest-cell rule--The deepest-cell rule generates the best approximation of an object by recording only the bottom-most cells that have been tessellated for the object. Parent cells do not contribute to the cells-per-object count, and they are not recorded in the index.

8.3. Viewing Spatial Data in SQL Server 2022

Spatial data loaded into a database can be viewed in the SQL Server Management Object Explorer. Figure 8.4 provides the view of the Canada Province. The shapefile of Canada Province can be downloaded from [15].

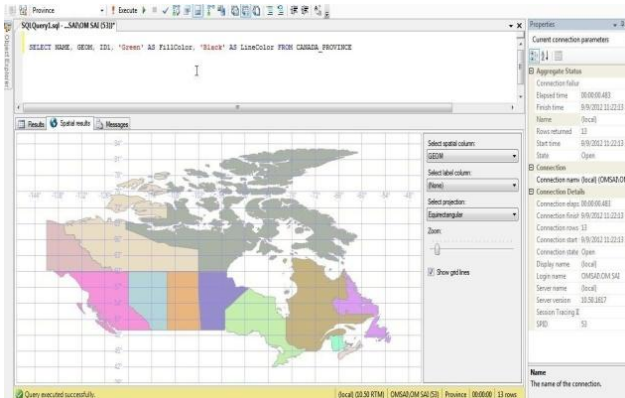


Figure 8.4. Canada Province



Figure 8.5. GUI of SQL spatial query visualizer

SQL Server 2022 also provides a free tool called SQL Spatial Query Visualizer, which can be used to view spatial results. This tool can be downloaded from [14]. Figure 8.5 provides the view of the Canada Province using the SQL Spatial Query Visualizer tool.

8.4. Viewing Spatial Data in DB2

DB2 provides a map viewer tool called GeoBrowser, which can be downloaded for free from [16]. The imported spatial data can be viewed using this tool. Figure 8.6 provides the view of American states along with the world countries using GeoBrowser.

8.5. Viewing Spatial Data in Oracle

Oracle MapViewer is a programmable tool for rendering maps using spatial data managed by Oracle Spatial. The latest version of MapViewer can be installed from [17].

8.6. Viewing Spatial Data in PostGIS

Desktop viewers for PostGIS data cannot render curved geometries. There are a number of open source options for desktop viewers / editors of PostGIS data:

1. QGIS, a C++ / Qt program
2. uDig, a Java / Eclipse program
3. gvSIG, a Java / Swing program

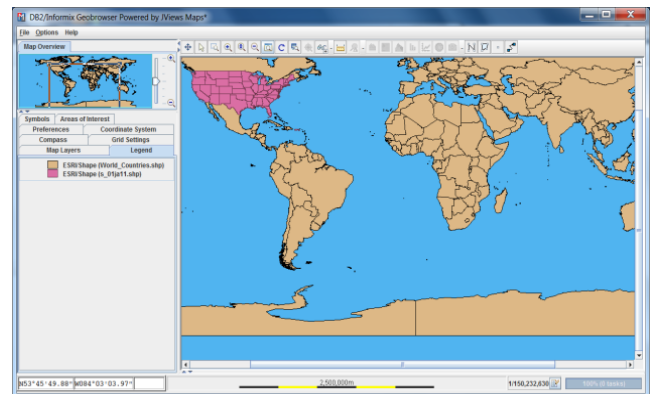


Figure 8.6. View of American states along with the world countries

8.7. Hybrid Spatial Indexing with Adaptive Grid-B-Tree (AGBT)

The Hybrid Spatial Indexing with Adaptive Grid-B-Tree (AGBT) technique is an innovative approach that combines the strengths of grid-based indexing, as used in DB2, and hierarchical B-tree indexing, as implemented in SQL Server 2022. This hybrid method is designed to dynamically adapt based on the characteristics of the spatial data, optimizing query performance and scalability.

8.7.1. Dynamic Grid-B-Tree Hybridization

- Grid Indexing: Initially employs grid-based indexing for uniformly distributed spatial data. The spatial grid is divided into cells, each associated with a B-tree structure for finer granularity.

- **Adaptive Switching:** When spatial data exhibits significant variability in distribution, the system dynamically switches to a hierarchical B-tree structure within specific grid cells, enhancing the precision and efficiency of the indexing.

8.7.2. Multi-Level Grid Adaptation

- The grid system dynamically adjusts its cell size and level depth according to the density of spatial features. Sparse areas are managed with fewer, larger cells, while dense areas are managed with more, smaller cells with deeper B-tree indexing.

8.7.3. Context-Aware Index Optimization

- AGBT continuously monitors query patterns and spatial data changes, optimizing the balance between grid-based and B-tree indexing. For frequently queried dense areas, the system may increase grid granularity and deepen the B-tree structure to improve performance.

8.7.4. Enhanced Query Performance

- By leveraging both grid and B-tree indexing methods, the AGBT technique aims to reduce query times for simple and complex spatial operations, such as nearest neighbor searches or spatial joins.

8.7.5. Scalability and Flexibility

- The hybrid approach of AGBT offers scalability across various types of spatial datasets, whether they are highly clustered, sparse, or exhibit varying levels of complexity.

8.7.6. Implementation Considerations

- **Integration with Existing Systems:** The AGBT technique can be implemented as an extension to existing spatial databases like SQL Server or DB2. This approach leverages current indexing mechanisms and augments them with this hybrid model for enhanced performance.
- **Cost-Benefit Analysis:** While the AGBT approach may introduce additional complexity in index management, the resulting gains in query performance and scalability make it particularly beneficial for large-scale Geographic Information System (GIS) applications.

9. Comparison of Features

Table 9.1 briefly summarizes the features of the major vendors of spatial databases.

Table 9.1. Comparison of Different Vendor Spatial Databases

Feature	SQL Server 2022	Oracle Spatial	IBM DB2	PostGIS
OS Supported	Windows	Cross-platform	Windows	Cross-platform
Licensing	Proprietary – closed Source, various levels of features based on version, Express version has full spatial support but limitation on database size and only use one processor	Proprietary – forms a separately licensed option component of the Oracle Database. Spatial is an option for Oracle Enterprise Edition, and must be licensed separately. It is not included in the Standard Edition and Standard Edition One	Proprietary – various levels of features based on version, Express Edition can be downloaded for free and also can be integrated with Spatial Extender can be integrated with Spatial Extender provided for free from IBM.	Open Source- GNU General Public License
Spatial Indexing	B-Tree	R-Tree/QuadTree Indexing	Grid Index	R-Tree indexes, implemented on top of the general GiST (generalized Search Tree) indexing schema.
Importing Data from Shape Files	Available	Available	Available	Available
Exporting data to Shape Files	Not Available	Not Available	Available	Available
Map Viewer Tool	Available as default	Oracle Map Viewer	IBM GeoBrowser, ESRI ArcViewer	QGIS, uDig, gvSIG
Relationship Functions	STRelate(), STEquals(), STIntersects(), STDisjoint(), STCrosses()	SDO_Distance(), SDO_Relate()	ST_Equals(), ST_Intersects(), ST_Disjoint(), ST_Contains()	ST_DWithin(), ST_Equals(), ST_Disjoint(), ST_Intersects(), ST_Relate()

10. Conclusions

SQL Server 2022 Spatial, IBM DB2 Spatial, PostGIS and Oracle Spatial are mature implementations of a spatial type system and are known for their relevant host databases. However, the research done as a part of this thesis suggests the following advantages and disadvantages over each other based on the following criteria listed below:

- **Cost:** PostGIS is the only open source spatial database of the four. SQL Server 2022 Spatial and IBM DB2 do provide free versions of spatially enabled databases; there are limitations on the size of the database and limited to the number of users and also on the number of processors. Oracle Spatial is commercial software, which involves licensing costs.
- **Ease of use:** Creation and usage of spatial objects is quite simple and straight forward in PostGIS and SQL Server 2022, whereas it is a bit complicated with DB2 Spatial and Oracle Spatial.
- **Tech Support & Developer community:** SQL Server 2022, Oracle Spatial has a strong tech support and a solid set of the developer community resources, where as DB2 Spatial and PostGIS tech support and developer community resources are very limited, which in a sense is a kind of hindrance for the developers or any enterprise for the usage of these spatial database sets.

REFERENCES

- [1] http://en.wikipedia.org/wiki/Spatial_database
- [2] <http://gis.washington.edu/phurvitz/professional/SSI/datatype.html>
- [3] <http://www-01.ibm.com/support/docview.wss?uid=swg24023357>
- [4] <https://www.ibm.com/docs/en/cfm/2.0.0.3?topic=ucfd-installing-db2-spatial-extender>
- [5] <https://www.microsoft.com/en-us/download/details.aspx?id=104781>
- [6] Alastair Aitchison. *Beginning Spatial with SQL Server 2008*. Apress, Berkeley, CA
- [7] <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/opt/csbsp1003.htm>
- [8] <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=%2Fcom.ibm.db2.udb.spatial.doc%2Frsbp4087.html>
- [9] <http://msdn.microsoft.com/en-us/library/bb933960>
- [10] <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/opt/csbsp3020.htm>
- [11] <http://en.wikipedia.org/wiki/B-tree>
- [12] <http://technet.microsoft.com/en-us/library/bb964712>
- [13] <http://www.sharpgis.net/page/SQL-Server-2008-Spatial-Tools.aspx>
- [14] <https://github.com/microsoft/SQLServerSpatialTools>
- [15] <https://www.sciencebase.gov/catalog/item/5ab555c6e4b081f61ab78093>
- [16] <http://www.ibm.com/developerworks/apps/download/index.jsp?contentid=642280&filename=DB2JviewsMapsDemo.zip&method=http&locale=>
- [17] <http://www.oracle.com/technetwork/java/javase/downloads/index.html>