

# ENGINEERING SPATIAL AND GRAPH ANALYSIS FOR BIG DATA

International Journal of Scientific & Engineering  
Research Volume 10, Issue 3, March-2019

Folami Gbenga Obokhai

19

Spatial and Graph analysis is basically about understanding relationships. The evolution of applications and infrastructure allows new platforms and technologies to find distinct ways to incorporate and exploit locational and social information into the business and analytics workflows. The emergence of Cloud services, Internet of things, mobile tracking, social media and real time systems creates new challenges which allow the management of data and more predominantly Pattern Recognition in data types, connections and the relationships.

We are including a wide range of spatial analysis functions and services to allow us evaluate data based on Nearest Neighbour Analysis, the congruence of the fact whether something falls within a boundary or region, and the visualizations of Geospatial pattern recognition on maps and imagery. The Property feature allows us the facility of graph storage within a database (PL/SQL), a SQL graph query language, and powerful built in built-in social graph comments for making recommendations, finding communities and influencers, pattern matching, and identification of fraud and other anomalies. This Property graph also has interoperability with the RDF graph pattern that provides link-data applications with W3C standards based-triple store query language and native inferencing.

The Property graph feature is unique to allow both a powerful in-memory analyst with built-in analytics and a scalable graph database. It also includes PGQL which makes it easier to perform graph reversals, and find graph instances and patterns of sub-graphs. Property graph support is a feature to handle the Big data generated these days which also contains inherent relationships between collected data entities. On due observation, these relationships can be structured as a property graph which express a set of connected entities , and analyzed to allow finding of opportunities and better decision processes. The property graph models entities as vertices, relationships as edges, and stores attributes as key-value pairs.

These new property graph capabilities for PGQL includes the querying of the graph data stored in the database and the location of in-memory subgraph instances that match a stipulated query pattern. These New property graph analytics are supported for SQL-based collaborative filtering, that could adequate management of recommendations. The referencing and collaborative work integrated and allowed more in-memory analytics to be added, including the variants of Pagerank, a personalized SALSA for making recommendations, a K-core for locating subgraphs by properties, Diameter, Radius, and Eccentricity to analyze distances in a graph, and PRIM for finding the minimum spanning tree of edges connected in all vertices of an undirected graph. The Property graph capabilities also include new support for undirected graphs, Node.js client, Apaches Zeppelin and an execution and scheduling manager to better control in-memory analyst tasks and resources.

It should be noticed that the geospatial data features are designed to support the more complex requirements that exist in Geographic Information Systems (GIS), enterprise applications, and location-enabled businesses and web applications. These features allowed include native support for geocoding, a routing engine, and spatial web services that are synchronously compliant with the Open Geospatial Consortium (OGC) and ISO standards. Support for advanced spatial models and types include a network data model , georaster( for geo-referenced imagery and gridded data), topology, 3D, including triangulated irregular networks (TINs) and point clouds (supporting LIDRA data), and linear referencing. These features provide a complete platform for geospatial applications disparaging domains, including defense, land management retail, insurance and finance.

This optimized database is spatial operations are faster, cloud-ready, and more developer friendly. The presence of GeoJSON support, a location tracking service that allows tracking of millions of objects against thousands of regions, the database allows more spatial functions, and an interactive HTML 5 map visualization component.

The new spatial features include spatial support for distributed transactions for cloud applications with distributed architectures; expanded spatial JSON support; some data services support for spatial operations in database for modern RESTful development; support for shared databases with spatial data types; expanded spatial web services features for scalability and usability; performance enhancements for operations on spatial point data.

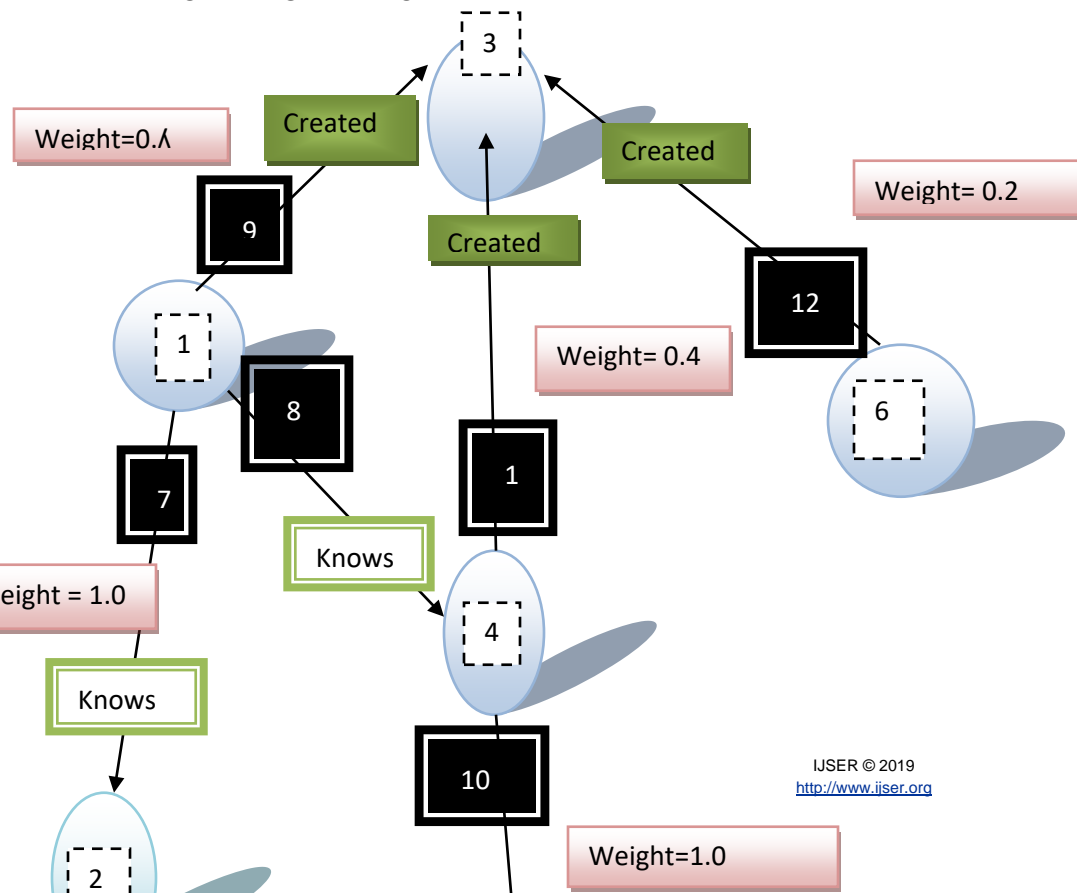
The RDF semantic graph feature with spatial and graph is a mature, special purpose graph conforming to the World Wide Web consortium standards. It provides a parallelized data storage, querying and inferencing that are used in a semantic integration of data and linked open data applications. This spatial and RDF support provides an open and secure RDF database with scalability over a trillion triples. These new capabilities include faster operations with database support for in-memory, better SPARQL performance with list-lash composite partitioning, a faster loading of RDF quads having literals greater than 2800 bytes, and native support for Turtle and Tig formats for loading data for creating RDF views.

The database kernel has capabilities of both spatial and Graph spatial nature, and geospatial and graph deployments to harness these database features for scalability, security, partitioning and parallelism. They help reduce application logic and support real world analysis by moving complex spatial and graph logic into the database. The processing power and bandwidth of extradata Database machine is exploited, realizing extreme performance capabilities.

### Property-Graph Description and Outline:

When Big data is generated, it should be noticed that much of these Big data contains inherent relationships between the collected data entities. We could easily structure these relationships as a Property graph- which is a set of connected entities. In the property graph we allow vertices to denote entities, the edges to denote relationships, and the associated properties or attributes are stored as key-value pairs for both categorizations mentioned.

THE PROPERTY GRAPH MODEL



**A set of Vertices (or Nodes)**

**Each vertex has a unique identifier**

**Each vertex has a set of IN/OUT edges**

**Each vertex has a collection of key-value properties.**

**A set of edges (or Links)**

**Each edge has a unique identifier**

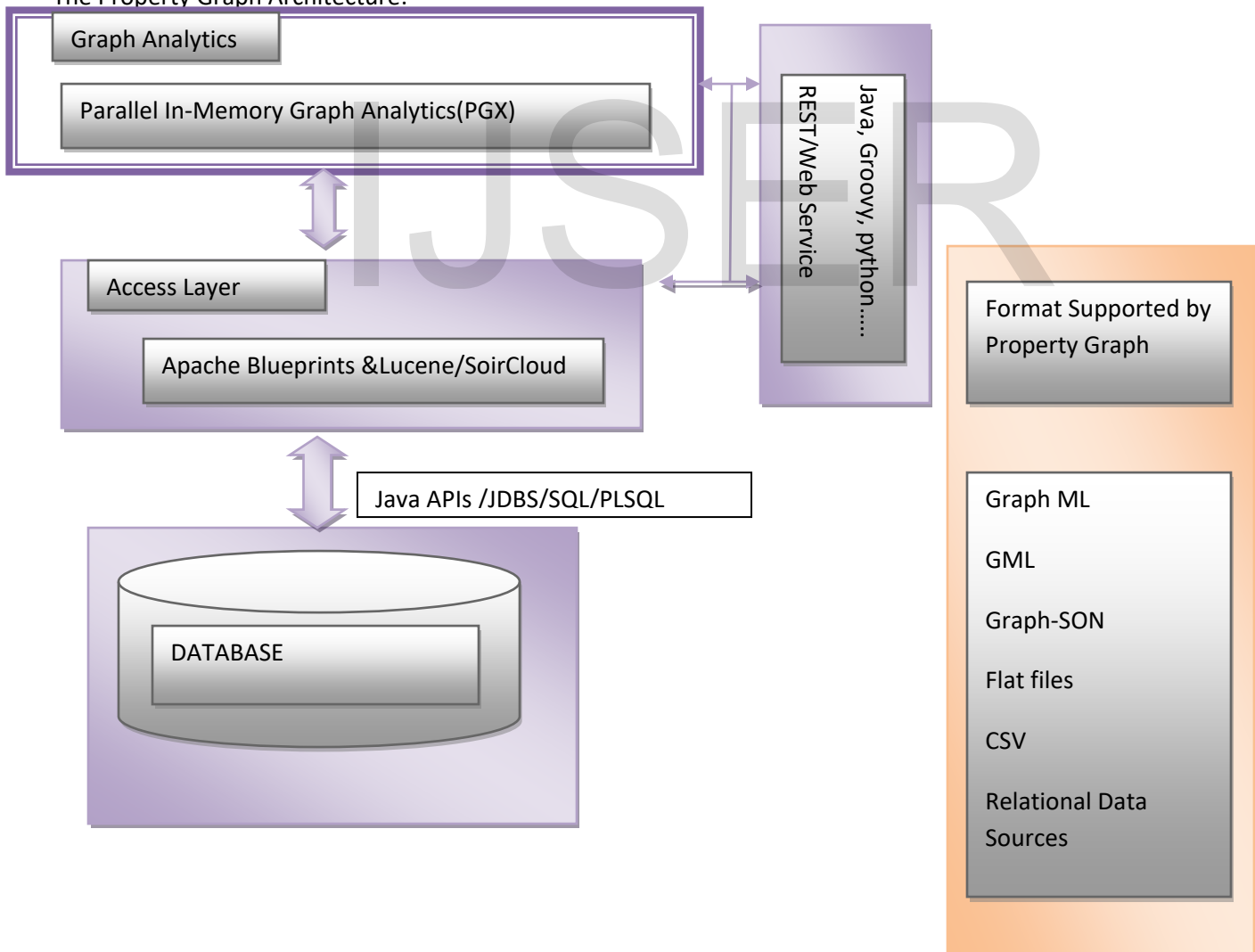
**Each edge has a head/tail vertex**

**Each edge has a label denoting type of relationship between two vertices**

**Each edge has a collection of key-value properties.**

We must notice that the major capabilities are the in-memory analyst and the data access layer. The in-memory analyst (PGX) is the engine for 35 built-in, powerful, parallel graph analytics. The graph database data access layer includes a Groovy-based console, Java APIs, a fast searching through text indexing, fast, parallel bulk loading, spatial filtering graph queries and multi-level security.

The Property Graph Architecture:



### *Explanation of Graph Analytics:*

---

The in-memory analyst is basically performs the graph analytics using the over 35 built-in , powerful, parallel, in-memory analytics, including ranking, centralizing, recommendation, community detection, and path finding. It should be noticed as inferred in earlier pages that most of the commonly used graph analytics can be also executed using SQL. SQL- based analytics are helpful for large graphs, reducing network traffic and obtaining more up-to-date results.

The modern server architecture is the infrastructure that the in-memory analyst takes advantage of, this parallelizes computation using multiple cores and sizeable memory configurations, allowing fast non-sequential data access across a larger portion of a graph read into memory.

The execution of the in-memory should be within a java application, or executed in multi-user, multi-graph in-memory analyst server environment on a WebLogic Server. The output of graph analysis can be another graph, such as bipartite, filtered, undirected, sorted or simplified edges graph.

### *Graph Analysis in Business: Application Layer display advantages*

---

<b>Recommended the most similar item bought by similar persons.</b>
<b>Finding out persons that are central in the given network like influencer marketing</b>
<b>Identify group of persons that are close to each other like segmentation group marketing</b>
<b>Finding out the sets of entities that match the given pattern like fraud detection.</b>

### *Querying using SQL and PGSQL query language*

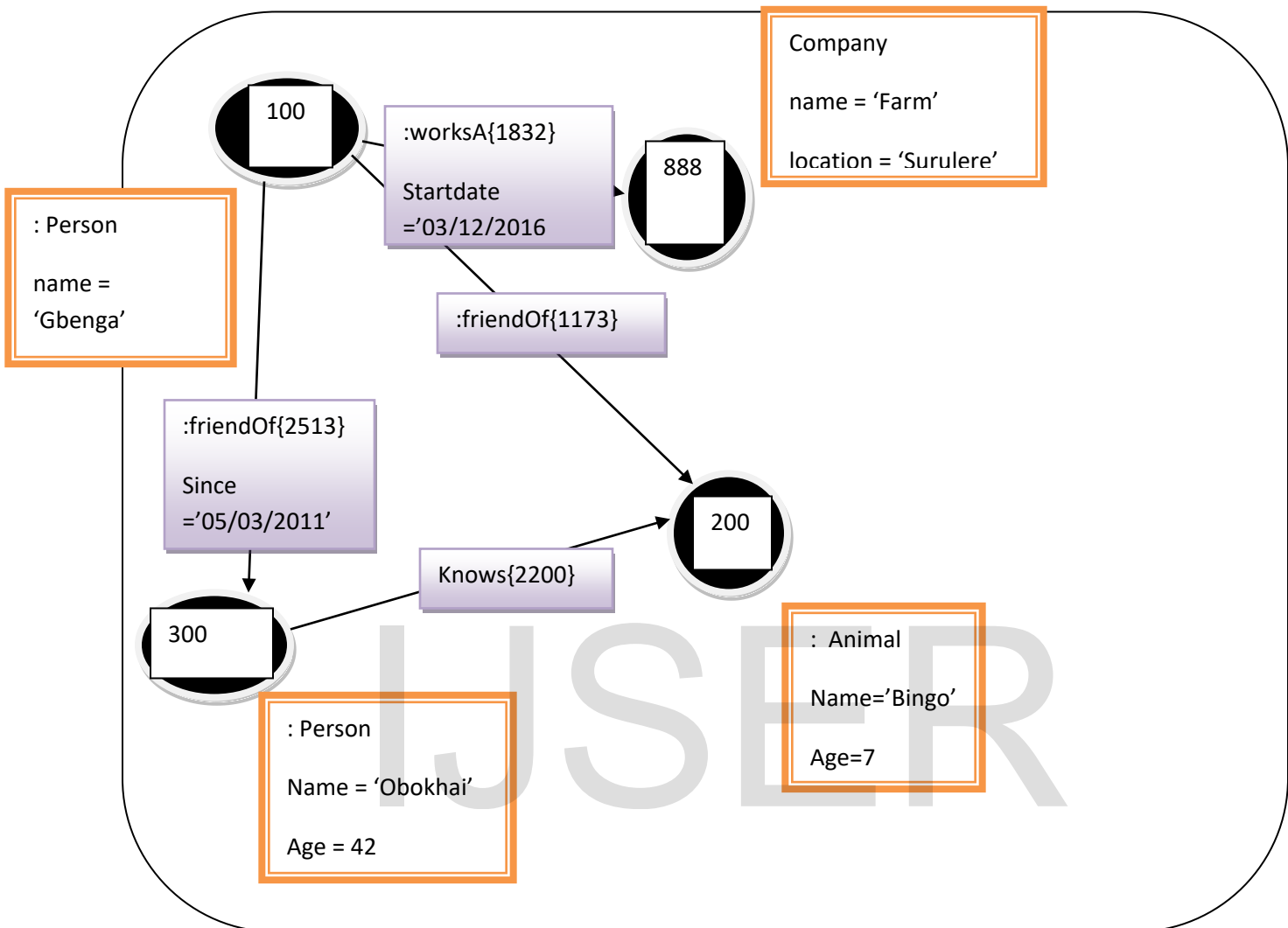
---

The described PGSQL query describes a pattern of graph with vertices, edges, properties, and their respective relationships. This query , when evaluated against a property graph, the query engine locates all subgraph instances that match to the specified query pattern. The query engine inturn returns the selected data entities from each of the matched subgraph instances.

```
SELECT v3.name, v3.age
```

```
WHERE
```

```
(v1:person WITH name = 'Randy')-[:friendOf]-> (v2: animal) -[:knows]-> (v3: person)
```



PGQL includes support for grouping (GROUP BY), aggregation,(e.g MIN,MAX,SD), sorting (ORDER BY) and some other familiar SQL constructs.

The PGQL also supports regular path queries (recursion) for the case of applications such as reachability analysis. A PGQL regular path query is simpler and more construct than a comparable SQL query would look like.

The following illustration matches a pattern repeatedly by defining a PATH pattern at the top of the query, referring to it using the WHERE clause, and using the star symbol (\*) for repeated matching.

```
PATH has a parent :m(child)-[:has father | has mother ]-> (parent)

SELECT x.id(), y.id(), ancestor .id()

WHERE

    (x;person WITH name = 'Gbenga') -/:has_parent*/-> (ancestor),

    (y) -/:has_parent*/-> (ancestor),

    x !=ancestor AND y != ancestor AND x != y
```

Result Set

x.id()	y.id()	ancestor.id()
600	400	300
600	500	300
600	500	400

The property graph can be queried using Java APIs, these Java APIs perform parallel scans on vertices and edges. Parallel retrieval takes advantage of the distribution of the data across table partitions, so that each partition is queried using a distinct or separate database connection.

Let us look at querying for a vertex using a vertex ID

Find the vertex with the vertex ID in the connections graph.

```
SQL> select v id, k, v, vn, vt.
```

```
From connection V7%
```

```
Where v id=1 ;
```

The output might be as follows:

1 country	Nigeria
1 name	Muhammad Buhari
1 occupation	President of the federal republic of Nigeria

### *Addressing Ease of development and Management:*

---

This is facilitated by a group of Java APIs. The built-in shell, provides database access and in-memory analyst operations. This command line shell interface, allowing exploration of the Java APIs to create and drop property graphs add and remove vertices and edges, search for vertices and edges using key-value pairs, create text indexes, and perform other manipulations. The Java APIs allow for inclusion of the Apache TinkerPop interfaces. The database structure and infrastructure allows supports for scripting languages such as Groovy and Python is also included. Groovy scripts allow developers to test Java code snippets without defining objects or compilations.

### *Fast Search with text indexing*

---

The features Apache Lucene and optionally Apache SolrCloud provide text indexing on properties for fast retrieval of vertices and edges. Native SQL text indexing is still supported, but text queries are automatically translated into SQL SELECT statements with a “contains” clause.

These features show the following:

### *Apache Lucene*

---

This feature allows text searching and indexing for queries on property graph elements. It must be noticed that the limitation of this feature is in the premise that the indexes cannot be shared directly amongst users and applications.



The Auto index characteristics are easy to use. It is a b-tree index that is created by specifying the index name and what to index, vertices or edges. It allows automatic updates as the graph elements change.

There is a Manual (selective) index which is a more flexible manual process. The index content is determined and loaded by the developers. This allows more flexibility to choose what graph elements will be indexed, increasing the focus of the index.

### *Apache SoirCloud*

---

This feature of the database system has a capability called faceted search. The SoirCloud is implemented with the Lucene. It allows multiple users and applications to share a SoirCloud index. The Documents are sent via the HTTP interface, shared, and then the index is replicated once the feature is enabled. The query coordinator transparently queries the shards and assembles the final results.

### *Text Indexing and Search*

---

Text uses SQL standards to index, search, and analyze text values stored in columns of the vertices and edges tables. All the existing K/V pairs can be text indexed in the property graph.

### *Fast, Parallel bulk Loading:*

---

The data can be loaded into multiple database partitions, because of the data type-rich flat file format and SQL \* loader, a utility is provided to easily convert tables and comma separated values (CSV) files into the flat file format. This supports other open source graph file formats, GraphSON, GraphML, and GML.

### *Spatial filtering to enhance graph analysis*

---

A spatial geometry, such as coordinates for an address can be stored as a property and analyzed. Spatial filtering in graph queries allows the enhancements of graph analysis. Support for point, line and polygon geometries and function – based spatial indexing, and access to spatial analytic functions.

### *An Overview of some Special Features of the Database System*

#### *Vector Performance and Acceleration*

---

Vectors can either be 2-dimensional and 3-dimensional sets of vertices, like in latitude, longitude and height that describe geometries of objects, such as points, lines, polygons, surfaces, and solids. These Geometries of objects more often represent the real world objects. Vector operations evaluate some or all of the relationships between geometries, like within distance, nearest neighbour, and also geometric interactions like touches, overlaps, contains, covers, distance, and buffer zone generation over geometries.

Spatial and Graph vector acceleration capabilities improve the performance of vector operations substantially. Vector acceleration also produces enhanced computational algorithms along with CPU and memory enhancements that allow the spatial index creation, geometric computations in functions and spatial operator secondary filter operation to all drastically improve in performance. Spatial and Graph vector acceleration builds on the improvements in Locator operator and function improvement.

### *Geometric Functions Vector*

---

Spatial and Graph provides over 350 functions that allow the performance of calculations on geometries, length or perimeter, area of polygon. These functions are used to length and area dependencies' like the length of an interstate highway, or the provincial boarder, or to determine the total area of all the settlements around a particular settlement.

Some other functions are used to generate new geometries, such as buffers, unions, intersection, and much more. The description allows us to notice that these functions can be used for e.g. to find the intersection between two sales regions, identification of the new geometry representing the union of two sales regions, to define market territories by creating a 5 mile buffer around all market boundaries.

These other functions include interior point, concave hull, and generation of triangulated irregular network through Delaunay triangulation. These other functions also allow for support of Cross-endian operations for transportation tablespaces.

### *Projections and Coordinate Systems*

---

Spatial and Graph integrate represent spatial information effectively and accurately, and also provide comprehensive tools for managing coordinate systems. Spatial and Graph also provides support for implicitly and explicitly transforming data between different coordinate systems, this enables explicit map projection transformation of vector objects from one coordinate to another.

The European Petroleum Survey Group (EPSG) uses a model that provides benefits of standardization, expanded support, and flexibility for all industries, georaster data vendors, and GIS user in general, this in turn allows a data model and data set based Coordinate system support.

These infrastructures allow the Spatial and Graph to support 3D coordinate systems, which include the "z" coordinate, that denotes the height of an object, in addition to the "x" coordinate, that denotes the longitude, and "y" coordinates, that denotes the latitude; the reprojection of raster's is also supported.

### *Spatial Aggregates*

---

Spatial and Graph performs aggregate operation on a set of input geometries, and return a single geometry object. The following statements returns the state boundaries of Lagos generated from all of the settlements around Lagos:

```
Select sdo_aggr_union(sdoaggrtype(geom, 0.6)) state
```

```
From geod_settlements
```

```
Where state_abrv='LAG';
```

The use of spatial aggregates improves performance and simplifies coding; users are allowed to define other aggregate functions.

### *Supporting Linear Referencing*

---

Linear referencing is always used in transportation, utilities, and telecommunications industries. Spatial and Graph database system structures can store and associate attributes and events with a specified segment on a linear geometry. The attributes and events are stored in tables separately from the geometry, and the geometry does not have to be duplicated in the attributes tables.

We allow functions to manipulate linear referenced geometries and they are also included such as snapping the closest point to the linear feature of a given point, and clipping a piece of a linear feature which is dynamic segmentation, and the conversions between standard and linear referenced geometries.

### *Topology Data Model:*

---

Spatial and Graph Database systems include a data model and schema that persistently stores topology in the database. In situations where there is a high degree of feature editing, and a strong requirement for data integrity across maps and map layers. Land management systems (cadastral) and spatial data providers benefit from these capabilities. Topology based queries typically perform faster than the alternative for relationships such as adjacency, connectivity, and containment.

### *The Issue of Georaster Support:*

---

Geometries could be represented by either Vectors or Rasters, or both.

Image processing systems usually refer to raster as data images, eg satellite imagery, or in airborne photograph.

Gridded Data: are raster data when they are used in GIS. The Spatial and Graph database system structure Georaster can store, index, query, analyze, and deliver raster image, and gridded data and the associated metadata.

Georaster is used with remote sensing, photogrammetry, and geospatial thematic mapping.

It has uses in a disparaging range of areas including environmental monitoring and assessment, geological engineering, and geological exploration, provision of image processing with simple innovation of PL/SQL procedures.

Georaster is the ability to perform raster analysis, extremely large images and data sets and is designed to deliver data management capabilities to large image processing and GIS solutions. Developers and Engineers are allowed to integrate this powerful data management technology with the leading image processing and raster grid analysis tools.

Georaster are cost-effective, loading and native storage flexible, and performant in nature. The following file formats are supported for loading and exporting Georaster objects, GeoTiff, JPEG 2000, and Digital Globe RPC. Locations are associated with Geometries in a raster, by assigning location values to a matrix of cells, that cover storing the cells as an array, and the raster. There is presence of transparent lossless compression, even though native JPEG 2000

provides a high compression ratio and image quality.

To allow for third party compression techniques, standard compression techniques including JPEG baseline and DEFLATE (lossless) ,with an open plug-in architecture.

Spatial and Graph database System allows the support for relational data tables (RDTs) to allow users to specify default alpha channel, and pyramid level in its metadata structure, thereby allowing a resampling algorithm that supports specifications of resolution unit and parallel processing in many operations, adding additional loading and export capabilities.

Georaster supports raster image and raster data sizes and degrees of resolution through pyramiding and very large image tilting. Automatic blocking size optimization is the resource that chooses a block size for georaster and also minimizes storage, while optimizing retrieval and processing.

### *Raster Algebra and Analytics*

---

The Spatial and Graph database system structure supports raster algebra operations that work on individual raster cells, or pixels in order to generate new maps from two or more raster layers.

Raster Algebra operations allow applications to robustly implement sophisticated analytical algorithms such as, Tasseled Cap Transformation (TST), Normal Difference Vegetation Index (NDVI).

Raster operation performance can be parallelized to scale up to 100 times faster for larger sets of data; statistical analysis functions dynamically computes, statistical values for a Georaster object, or producing individual statistical values. The explanation above allows for the support of image classification, time series analysis, and raster GIS modeling with capabilities to merge multiple bands or layers of different Georaster objects into a single Georaster object.

### *Sophisticated Image Processing*

---

Spatial and Graph database system architecture Georaster provides advanced image processing and serving Capabilities. These include Ground Control Point georeferencing, rectification, orthorectification, image scaling Stretching, masking, filtering, image segmentation, NDVI computation, Tasseled Cap transformation, image appending, Bands merging, large-scale advanced image mosaicking, and virtual mosaic. These Georaster uses industry standard

resampling and interpolation methods for image and raster transformations and operations. Grid point interpolations infer values at spatial points and positions between or within cells. Irregularly shaped regions inside an image can be defined with bitmap masks. Transformations between 2D or 3D ground coordinates, and 2D cell coordinates, and vice-versa are supported. Customizable memory control further improves performance.

### *Manageable Administration:*

---

Georaster templates are supported to develop Georaster applications. Georaster DML triggers are created and monitored by the system automatically. The users can monitor resource-intensive operations on the Georaster system data. Partial raster updates are also supported.

### *THE JAVA API:*

---

A Java API supports query, raster management, and manipulation. It also allows and supports the development of ETL tools, Web applications, and raster processing applications; thereby simplifying the development of Java Applications that use, access, and manipulate raster and gridded data sets stored in the Database. It supports features such as ground control point (GCP) storage and manipulation, grid interpolations, reprojection, GCP georeferencing, and getcellvalue.

### *Loading and Exporting Raster Data*

---

It is necessary to notice that the Georaster data type is supported by the leading third party GIS and image processing vendors. The Geospatial Data Abstraction Layer (GDAL), natively supports importing and exporting of over 45 raster file formats to and from SDO\_GEOASTER. GDAL is a high performance tool that supports large file sizes, and it includes c/c++, Java, Python APIs to access the Georaster; and the utilities to translate the different raster formats, wrap rasters, generate contours from DEM rasters, and a host of other raster operations. The concept of concurrent batch loading and exporting of various image and raster files using GDAL is included or facilitated in the Georaster by an ETL wizard tool.

### *THE SPATIAL ANALYTICAL FUNCTIONS*

---

Spatial database architecture supports spatial analysis and mining. This process of data mining allows automatic Discovery of knowledge from a database, such as classification of data as a result of some samples, and discovering

of hidden associations between various data attributes, and clustering to identify intrinsic patterns.

Data at a certain location could be influenced by data of the neighbourhood elements; these correlations are exploited in the following ways:

- **Colocation Mining:** used to determine associatively of distinct patterns, like determining the clause whether locating a pizza restaurant franchise with a video store results in higher returns in sales.  
**Location prospecting:** this involves using spatial information to determine the prospective derivatives of a particular location, like identifying the best neighborhood for opening new hospitals by an evaluation of the population of the patients within the neighborhood.  
**Spatial clustering:** used to determine where particular elements cluster, in ratio to the spatial distribution of the elements determining the spatial description of the region or area. It is like determining the regions where crime rates are high in order to know where exactly to deploy additional police.  
**Materializing Spatial Correlation (constructing or observing Neighborhood Influence):** this accesses the value of an entity by correlating it with the value of such similar entities within the neighborhood. It is like examining the values of similar houses in a neighborhood when accessing the value of a house within the same neighborhood.  
**Combining and binning data into regions:** thus it is used to predict the response of a particular region of data set to a geospatial adjustment. It is like determining if southern African customers of a certain age range, would likely prefer regular or diet coke.

### *The Iterative Process of Geocoding:*

---

Geocoding is the process of associating geographical references to location coordinates; geographical references are addresses, postal codes, while location coordinates refer to longitude and latitude. This database system architecture provides international address standardization, geocoding points of interest, and mapping them with geocoded data stored with the database system architecture, it also allows capabilities of reverse geocoding, batch geocoding, as well as other geocoding capabilities. The APIs provided for geocoding are SQL, Java, and XML, and it could be deployed at either the database server tier or at the middle tier, but preferably at the database server tier because its unique unparsed address adds great flexibility and convenience to customer and third party applications. This architecture being described supports standard address geocoding based on interpolation and a point-based geocoding where the data sets include exact points of interest, intersections and addresses, this capabilities drive the utility of spatial and graph geocoding to include point address geocoding support for countries that don't have address ranges and language support for countries that have addresses in multiple languages; thus you can perform reverse geocoding without specifying a country code in the procedural pattern.

### *Routing Engine and Database Architecture*

---

The routing engine in this database architecture is easily deployed in servlet containers, and provided as a Java client library, aiding its giving preference to either fastest or shortest routes, provides a detailed summary of directions from a location to multiple destinations with information containing detailed driving directions, and times and distances along the route or street. This infrastructure supports restrictions and conditions required

for advanced routing applications; it could handle a logical turn restriction that involves more than two way points in the route geometry. These enhancements yield more accurate results for logistics and truck routing applications.

The Database architecture allows for the incorporation of some capabilities; the route engine servlet supports light-weighted location based queries related to speed limit. Special services to enhance features and capabilities are described below:

### *Spatial Web Services:*

---

Spatial and Graph database system analysis architecture provides a web services platform that accesses, incorporates, publishes, and deploys geospatial services, which include services for geocoding, routing, mapping, and geospatial feature transactions. The description above provides a transactional service-oriented architecture platform with enterprise-class security. The security provided includes authorization, authentication, and transport confidentiality and integrity.

These Web feature services (WFS) help enable query and retrieval of geographic feature information in vector format, encoded in GML. Database transactions are supported on WFS –T feature tables through SQL without any restrictions. Java and PL/SQL client APIs are also provided. Web Feature Services (WFS) allows for the support retrieval of coverage, or raster data such as DEMs.

This provides a unified framework and Web –based administrative console for WFS, WCS, and CSW, for easy deployment, administration, and diagnosing. The registration of spatial layers is simplified by this menu –driven user interface. It eliminates the need for an administrator to run PL/SQL scripts by allowing users to browse through spatial layers in order to publish these spatial layers. The user interfaces are allowed to be used as client to WFS servers.

### *Some Web Services Enhancements*

---

It has become easy to scale web services applications because reading and publishing of data from multiple data sources in the same weblogic server instance is supported by WFS, WCS,[JSI2] and CSW. The definition of a metadata application profile which, enables easy exchange of spatial, and metadata among applications. Spatial and Graph CSW service can interoperate with other services and query languages that implement the ISO profile. It is noticed that the performance of the CSW is considerably improved; this is because XML full texts can be created on metadata profile data, enabling XQuery full text queries that search on XML documents efficiently by combining text and structured search.

### *An Overview of Network Data Model Graph Features*

---

The Network Data Model has a PL/SQL API for managing and maintaining network data in the database, and a Java API for performing network analysis and creating and applying network constraints. The Network Data Model (NDM) stores physical and logical network data structures commonly used in utilities, transportation, oil and gas. It explicitly stores and maintains network connectivity and provides network analysis capability, including shortest path, nearest neighbor, within cost, and reacability. NDM supports partitioning large networks into smaller sub-networks that are manageable and can automatically load network partitions into memory as needed for efficient in-memory analysis.

NDM integrates with Spatial and Graph geocoding and the routing engine; the applications that are using these features can perform their analysis using NDM functions. NDM has modeling and analysis features that meet the requirements for utility networks, logistics, transportation, and other network –based applications.

### *Modeling Features*

---

Customized link and node properties

Model of partial link paths, partial link paths are also called subpaths.

Perform partitioning of logical networks, eg social and biomedical pathways networks, based on the metrics appropriate to the application.

Model and represent any point along a link of all analysis functions, such as specific addresses in street networks with any number of properties on the nodes and anti-nodes or links.

### *Network Analysis Features*

---

- To Generate a polygon which represents a region that can be reached with a specific cost from a given node with a specific cost. It is shown in application in the generation of drive-time and drive-distance polygons.
- Computation of the shortest route connecting a given set of nodes.
- Compute K shortest paths between two nodes.
- Compute a buffer based on network cost; the buffer representation contains coverage and cost information.

### *Feature Modeling, Analysis, and Editing*

---

Feature Modeling bridges the gap between concrete objects of interest in a real world and abstract world network elements.

Feature Modeling simplifies application development by connecting real world objects with elements. It maintains relationships between application features and network elements (nodes and links); this can be illustrated by a scenario of a utility network application that needs to find affected households when a substation experiences a power failure, through feature metadata, simplifying application development and maintenance.

Network Feature Editing allows you create and manage an NFE model; this model extends the feature modeling capabilities by using Java Swing objects to enable visualization and manipulation of features; determine features on the top of an existing network and PL/SQL APIs.

### *Network Modeling with Time, Multimodal Transportation Routing*

---

Spatial and Graphs adds support for modeling networks, that have time dimensions. Sometimes time properties are associated with nodes and links, and specifications of temporal inputs in network analysis queries. Utility networks



experience different load ranges based on seasonal demand and times of the day. Travel times on road segments vary with the different times of the day. NDM supports modeling and analysis of multimodal transport networks, and computing the fastest paths on multimodal transportation networks.

### *GeoJSON Support*

---

Spatial and Graph support the use of GeoJSON objects for storage, indexing, and managing of geographic data that is JSON (Java Script Object Notation) format, directly into the database. The GeoJSON is an open data format designed for representing simple geographical features, with their non-spatial attributes. This is based on JSON; a lightweight data interchange format that gradually became a standard for reading and publishing data in web, in Big data, and in the Internet of things. You can use spatial operators, spatial functions, and a special SD\_GEOMETRY method to work with GeoJSON data; and conversely, if the application requires a JSON data store, GeoJSON data can be embedded into that JSON store, thus allowing these JSON documents to be spatially indexed and used in spatial queries.

This allows Developers and Engineers to access JSON through REST services, or other APIs, and use SQL to query JSON documents, this gives a flexible application development and it also releases powerful SQL analytics for development environments.

### *The Enrichment of Location Data*

---

The data set functions include commonly used textual location data such as names, places, addresses, partial addresses, longitude, and latitude information. This features of the spatial and graph database system architecture enables you to process less structured geographic and location data to facilitate the information to be categorized, compared, filtered, and associated with other data; like the case of enriching data with only partial names to include, city, CDA, local government, state, and country, allowing the enriched data to become joined or analyzed with other data sets which may have state level information. This very useful in the comparism of Big data results with structured information in operational systems and data warehouses.

### *Improvements made on the Spatial Indexing*

---

Spatial indexes can be system-managed. This new index type eliminates the need for most or all index type partitioning management operations. This demonstrates the main benefit of spatial index management.

### *Point Data Optimized Index Type*

---

Point Data sets are only commonly used for location services, point of interest, moving objects, and applications often require fast update, and query performance since data like yellow page data can be really massive. When a large number of concurrent DMLs are performed on spatial tables, performance using R-tree indexes can be very challenging; thus an alternative index type, the composite B-tree index can therefore significantly improve the performance of spatial index creation and DML operations for updates of only Point Data sets, with a relatively 20-30 times faster index creation, and about 10 times faster for updates.

### *Geographical Data the Map Visualization*

---

HTML-5 has a spatial visualization feature mashed up component, that allows developers familiar with SQL and JavaScript to incorporate a wide variety of map styles and spatial analysis into business applications; the mapping engine of these map styles visualizes data in the Spatial and Graph database system architecture, thus allowing developers to combine this data extrinsically with external web services such as WMS, WFS, GeorSS streams and WMTS. Dynamic tile layers enable large data sets to be visualized in an interactive manner, with the ability to explore individual features.

### *Location Tracking Server*

---

This enables you to define regions of interest, track the movement of objects into or out of those regions and receiving of notifications when certain movements occur. There is an increasing demand for applications to be able to monitor subscriber location data continuously, since location is becoming an increasingly important aspect of daily lives, location-sensing devices are actually becoming more ubiquitous. An API is provided for the continuous location monitoring, of objects within a tracking network. The use of a queue mechanism in the management and handling of incoming location updates and tracking of requests and outgoing relevant locations thus this delivers efficient, continuous location for monitoring thousands of objects within the database.

#### New Spatial Features in the Architecture

- Spatial support for Distributed Transactions
- REST Data services for Spatial Operations
- Spatial index for Spatial Operations.

### *RDF Semantic Graph Data Overview*

---

This is a special purpose graph for linked data and web applications that conform to World Wide Web consortium standards. This represent standard for defining complex, semantically related data and SPARQL is a query language designed for graph analysis specifically. The scalable graph data platform is integrated with the database for scalability, security, performance and high availability.

### *Data Manipulation*

---

It supports all standard database loading, storing, and data manipulation operations on RDF models. Each RDF model contains a set of subject-object-relationship triples. The edge is a link (or relationship) that connects a subject node to an object node and is labeled by a predicate (property). Space-efficient storage saves up to 60% disk space for scalable and performant loading, querying, and inferencing.

### *Native Inferencing*

---

It is a native, forward-chaining, persistent inferencing using any combinations of RDF, RDFS, and EL profiles, as well as user-defined rules for specialized inference capabilities. This allows the provision of a plug-in framework to support specialty reasoners. There is support for ladder-based inferencing that ensures newly inferred triples are labeled with the proper security.

### *Database Querying of RDF Graphs*

---

RDF data can be queried using SQL; the database SQL SEM\_MATCH tables function embeds SPARQL graph pattern queries in an SQL query. A view-like feature to combine queries for models is provided by a virtual model capability. The database architecture supports GeoSPARQL for storing, and querying spatial data in an RDF graph.

RDF views are allowed to be created on relational tables, and views; the W3C standards for Direct Mapping, custom mapping and RDF views are allowed to present relational data in RDF triple format so it can be queried using SPARQL and connected with other link data and facilitation of enterprise data integration.

SQL queries can extract more semantically complete results from the table data by associating relational data with ontologies that organize the domain knowledge of data. A SPARQL gateway feature is provided that presents SPARQL query results in XML formats for visualization tools that support XML data sources.

Sensitivity labels can be defined on individual triples. Semantically indexed documents can be searched using the SEM\_CONTAINS operator with standard SQL query. The search criteria for these documents are expressed using SPARQL query patterns that operate on the information extracted from the documents.

The described allows the RDF semantic graphs to support parallelism, compression, partitioning, Real Application Clusters (RAC), and exadata database machine enterprise-level performance and scalability.

Integration with property graph is enabled by RDF views on the property graph data for SPARQL queries and property graph views on RDF data for property graph social network analysis.

### *Exadata Database Machine*

---

Engineered systems provide high performance, high bandwidth, and massive parallelism with enormous capacity to address the challenges faced by high volume workloads; the combining of Spatial and Graph advanced analysis with exadata Database machine performance and scalability delivers an ideal platform for the most demanding applications. Spatial and Graph fully uses the balanced hardware and highly parallelized architecture of Exadata. This achieves results faster and features are engineered to natively leverage the parallelism, partitioning, indexing, and the scalability features of Exadata without application changes. The fully parallelized joins and aggregations of Exadata coupled with the extreme I/O bandwidth and high performance of Exadata storage server provide Spatial and graph analysis with the processing power need for server -based geo-processing and graph applications. LTP index compression improves query performance by compressing and increasing memory resilience for spatial and graph indexes; the Exadata hybrid columnar compression increases memory residence for large spatial and graph data sets, and the rule sets for inferencing.

### *Enterprise Features of A Database Architecture*

---

These are features which enrich the Spatial and graph analysis capabilities through a flexible Internet deployment architecture, object capabilities, and robust management utilities that ensure data integrity, data recovery, and data security. Some features are described below:

### *Spatial Indexes Partitioning Support*

---

The Database architecture includes partitioning, where a single logical table and its indexes are broken up into one or more physical tables, each with its own index. The Spatial Indexes associated with partitioned tables can be partitioned; range partitioning is the partitioning scheme supported for spatial indexes.

Partitioning offers significant performance, scalability, and manageability benefits which include:

- Partitions can be split, merged, and exchanged.
- Ability to build indexes on partitions without affecting the queries on other partitions.
- Reduced response times for long-running queries; partitioning can reduce disk I/O operations.
- Reduced response times for concurrent queries; I/O operations run concurrently on each partition.

### *Parallel Index Creation*

---

Geospatial R-tree index and graph B-tree index creation can be subdivided into smaller tasks that can be performed in parallel, making use of unused hardware (CPU) resources. Large non-print datasets (commonly used in GIS applications) can show dramatic performance improvements.

### *Parallel Load, Query and Inference*

---

This helps location services and land management applications, which need to execute high volumes of spatial queries quickly. Spatial queries can run in parallel on partitioned spatial indexes, improving the performance of “within distance”, “nearest neighbor”, and “relate” queries. RDF graph data loading, graph queries and inferencing operations are also fully parallelized.

### *Replication*

---

Distributed Systems that involve geographically dispersed yet logically replicated web sites, can take advantage of synchronized replication of data objects across multiple databases.

### *Database Workspaces and Long Transactions*

---

Workspaces are virtual environments that allow current, proposed, historical spatial data values, to be managed in the same database. Workspaces can be shared and they used to isolate a collection of changes to production data until they are approved and merged into production; kept long term history of changes to data; and the creation of multiple data scenarios', based on a common data set for the “what if” analysis.

### *CONCLUSION*

---

Spatial and Graph analysis understands the relationship between the Internet of things, Cloud services, mobile tracking, social media, and real time systems creates new challenges to manage the volume of data, but more importantly, to discover patterns, and connections. The Database includes a wide range of spatial analysis functions and services, to evaluate data based on distance, proximity, and to visualize geospatial patterns on maps and imagery. The property graph feature also enables analysis of inherent relationships between collected data entities for a variety of graph use

cases, including making recommendations, finding communities and influencers, pattern matching, and identifying fraud and other anomalies. Features from native 3D point cloud, raster, topology, and network models to geocoding, routing, web services, JSON, and REST interface, map visualization makes this a complete, advanced geospatial platform. The RDF semantic graph model enables creation of a unified metadata layer for disparate applications that facilitates identification, integration and discovery; this framework has been adapted for linked data and social network applications that by leading organizations worldwide.

IJSER