ORACLE SPATIAL DATABASE WITH MAPVIEWER AND MAP BUILDER FOR DESGINING SPATIAL APPLICATION OF THAPAR UNIVERSITY

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science & Engineering

By:
Mudit Kumar
(800832022)

Under the supervision of:
Mr. Parteek Bhatia
Assistant Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

JUNE - 2010

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Oracle Spatial Database with MapViewer and Map Builder for Designing Spatial Application of Thapar University"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Parteek Bhatia and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Mudit

(*Mudit Kumar*)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Parteek Bhatia

(Mr. PARTEEK BHATIA)
Assistant Professor,
Computer Science and Engineering Department,
Thapar University, Patiala.

Countersigned by

09/07/2010

(Dr. RAJESH BHATIA)
Head,
Computer Science & Engineering Department,
Thapar University,
Patiala.

29.7.10

(Dr. R.K.SHARMA)
Dean (Academic Affairs),
Thapar University,
Patiala.

i

Acknowledgement

No volume of words is enough to express my gratitude towards my guide **Mr. Parteek Bhatia**, Department of Computer Science & Engineering, Thapar University, Patiala, who has been very concerned and has aided for all the materials essential for the preparation of this thesis report. He has helped me to explore this vast topic in an organized manner and provided me with all the ideas on how to work towards a research-oriented venture.

I am also thankful to **Dr. Rajesh Bhatia,** Head of Department, Computer Science & Engineering Department and **Dr. Inderveer Channa**, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there at the need of the hour and provided with all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

Mudit Kumar

(800832022)

Abstract

Oracle Spatial is an integrated set of functions and procedures for handling spatial data stored in an Oracle database. Spatial data represents the essential location characteristics of real or conceptual objects as those objects relate to the real or conceptual space in which they exist. Once this spatial data is stored in an Oracle database, it can be processed, retrieved, and related to all the other data stored in the database using the Oracle Spatial.

The MapViewer (or simply, MapViewer) is a server-side component that constructs maps by reading appropriate database views and tables and returns the maps to the client applications in the appropriate formats. MapViewer provides web application developers a versatile means to integrate and visualize spatial data with maps. It uses the basic capability included with Oracle10g/11g to manage geographic mapping data. The MapViewer is supported by the Oracle Map Builder tool which is used to create and manage the MapViewer mapping metadata which includes styles, themes and base maps. Besides handling the metadata, the tool provides interfaces to preview the metadata and display the original spatial information from the spatial tables without creating MapViewer metadata.

The condition of spatial application development is not much satisfactory in India. Most of the application development works are still approaching the traditional framework of application development and avoiding the powerful features of spatial technologies. In this thesis work, we have designed and developed a spatial application of Thapar University. This application shows the map of Thapar University and is featured with various map options like navigation panel and distance tools *etc*. Our developed system has the provision to view the different locations depicted in the map. It has also a feature to find the distance between two locations in the map.

Table of Contents

List of Figures

List of Tables

ORACLE SPATIAL DATABASE WITH MAPVIEWER AND MAP BUILDER FOR DESGINING SPATIAL APPLICATION OF THAPAR UNIVERSITY

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science & Engineering

By:
Mudit Kumar
(800832022)

Under the supervision of:
Mr. Parteek Bhatia
Assistant Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

JUNE - 2010

Chapter 1

Introduction

1.1 Introduction

A picture is worth a thousand words. This is particularly true when trying to capture the complexity of interactions among people, resources, products, and business processes distributed over geographic space. For many centuries people have relied on maps to capture and simplify these complex relationships, turning them into readily consumable, powerful packages of unambiguous information. The basic Oracle platform delivers this powerful, universally understood capability to every developer.

Geographic data has traditionally been managed in proprietary formatted files and displayed using special GIS applications. Oracle Database 10g/11g provides an open and standard-based geographic data management solution via either Oracle Spatial or Locator. We can load all types of geometric data into a database, create spatial indexes, and issue spatial queries through SQL using Oracle Spatial concepts. Spatial applications can be developed using Oracle Spatial concepts and tools. Some examples of spatial applications are as follows:

- **Spatial Information for Managing the London Bus Network**: This application uses Oracle Spatial to improve the planning and management of the bus schedules and routes for the city of London.
- **P-Info: A Mobile Application for Police Forces**: The P-Info system is used to provide mobile, location-enabled access to mission-critical information for police officers operating in the field.
- **Risk Repository for Hazardous Substances**: The Risk Repository for Hazardous Substances system gives access to information on risk and possible effects of storing, processing, and transporting hazardous substances.
- **USGS National Land Cover Visualization and Analysis Tool**: The USGS spatial data warehouse tool uses Oracle Spatial to search, visualize, and analyze land-cover data for the United States.
- **U.S. Department of Defence MilitaryHOMEFRONT LBS**: The MilitaryHOMEFRONT LBS (Location Based System) application shows how

Oracle Spatial is used to locate and geocode information and how to provide street navigation to users.

Oracle Spatial uses three key terms spatial, geospatial, and geographic information system (GIS) in order to define spatial concepts. These terms are defined as follows:

- **Spatial:** Spatial data [8, 9, 22] represents things that exist in space. Examples include data representing roads, property boundaries, X-rays, machine parts, or molecules. Spatial technologies store, manage, analyze, and display spatial data. Spatial applications [11] areas include GIS, computer-aided design/manufacturing, genomics, medical imaging, molecular chemistry, and most life sciences. Figure 1.1 depicts some of the sources of spatial data existing in the real world.

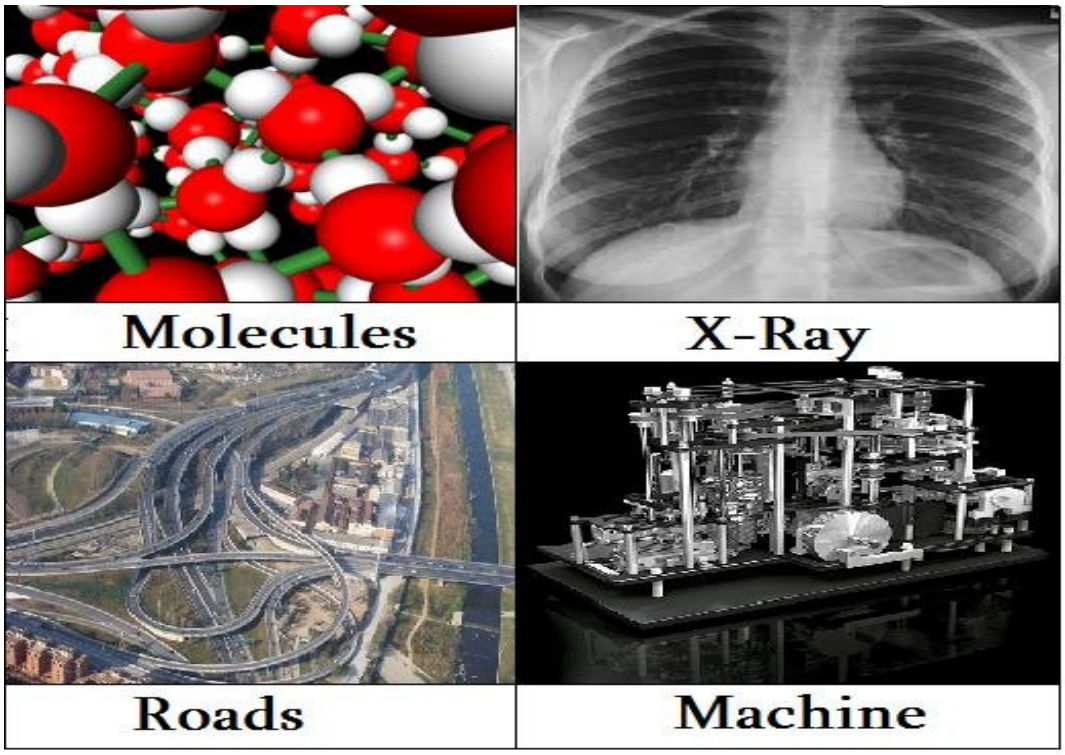

Figure 1.1: Examples of Spatial Data Sources

- **Geospatial:** Geospatial data represents things that are referenced to a location on the earth. Examples include data representing roads, rivers, property boundaries, or building footprints. Geospatial technologies are the subset of spatial technologies that store, manage, analyze, and display geospatial data. Geospatial applications areas include GIS, location-based services, seismic survey, and geospatially enabled enterprise applications. A scene is shown in figure 1.2 which can be used as a source of geospatial data.

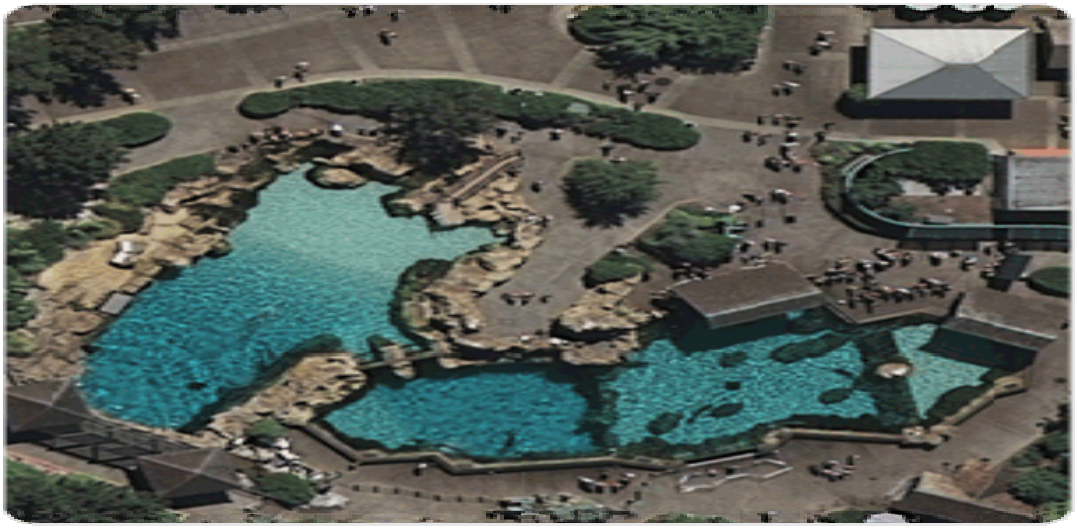

Figure 1.2: A Scene as a Source of Geospatial Data

- **GIS (Geographic Information System):** GIS is an information system or application that stores, manages, or analyzes data that represents geospatial features and that supports geocentric workflows as shown in figure 1.3. GIS systems have been used in a variety of applications, such as land use planning, geomarketing, logistics, distribution, network and utility management, and transportation [1].

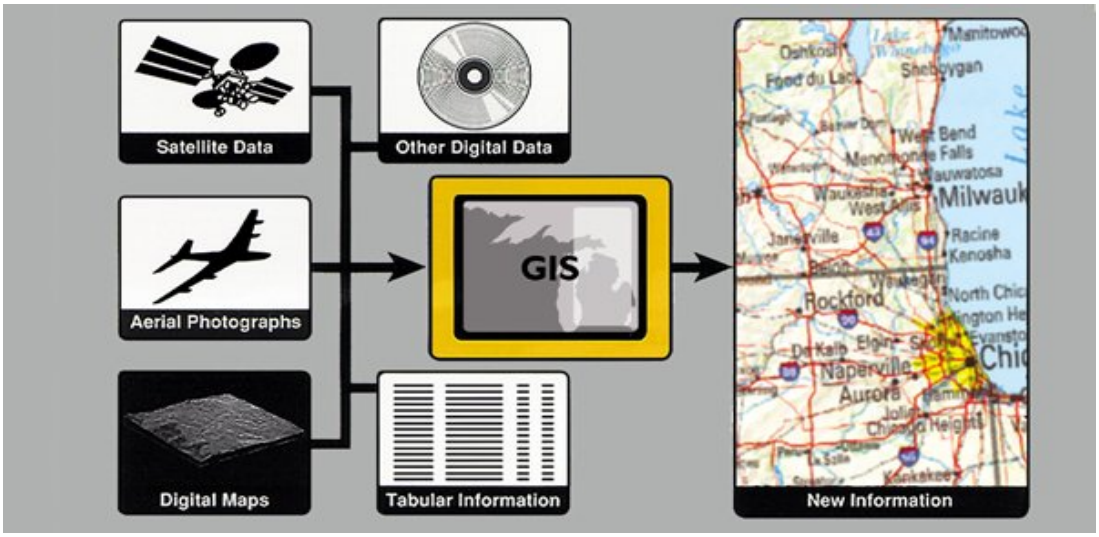

Figure 1.3: Geographical Information System (GIS) [10]

1.2 Oracle Spatial Technology

Oracle supports information systems that range from workgroup to multi enterprise systems. As shown in Figure 1.4, Oracle has added a number of capabilities that facilitate enterprise implementations.

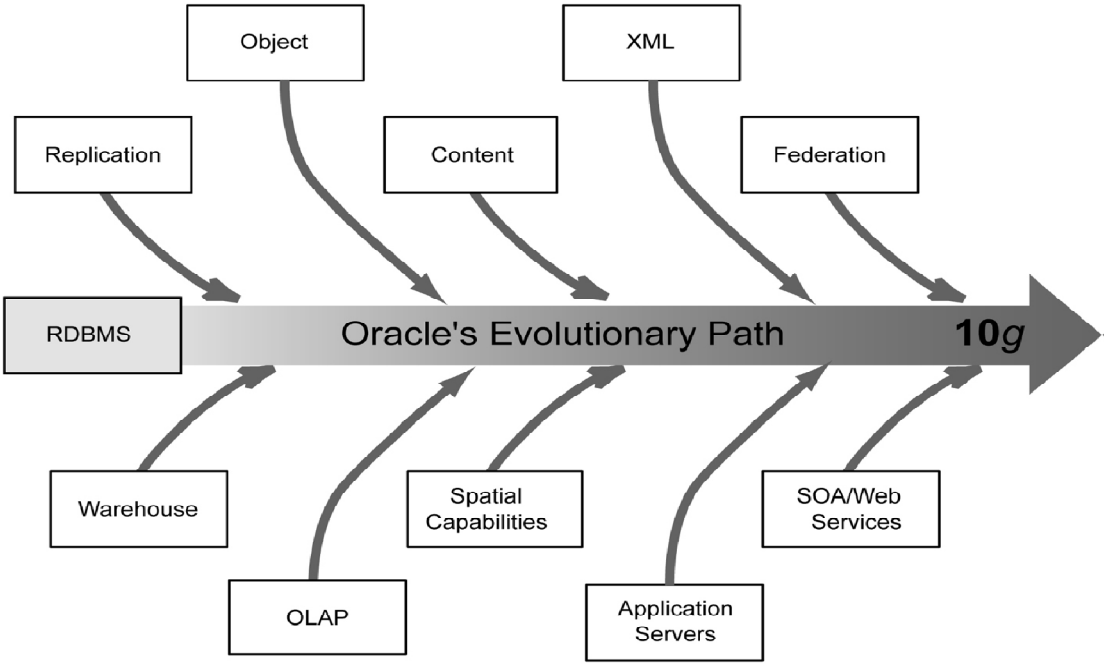

Figure 1.4: Oracle's Evolution toward Enterprise Spatial Integration [7]

One of those capabilities is "spatial." Oracle first added a spatial capability in 1996 a rudimentary point feature handler with a grand name: "helical hyperspatial" code or just "HH." HH code allowed the Canadian Hydrographic Survey to store, index, and display millions of bathyspheric data points. HH code is long gone, as are the cumbersome relational representations of earlier versions. With Oracle 10g, spatial capabilities are integrated throughout the entire technology stack. These capabilities include self-tuning R-tree indexing, scalable partitioning for multiterabyte data sets, and Real Applications Cluster (RAC) support for near-linear scalability for high-availability and high-performance systems.

1.2.1 Oracle Spatial Capabilities

Oracle has rolled most spatial features into Oracle 10g Enterprise and Oracle 10g Standard as a no-cost feature set called Oracle Locator [21]. Core spatial capabilties have been shown in figure 1.5. Many applications can be spatially enabled using standard components that are present at no additional cost in every Oracle database and application server. Specific Locator features include:

- Object types that describe and support geospatial features such as points, lines, and polygons.
- User-selectable R-tree and quadtree indexing that is integrated into Oracle's database servers.

- Spatial operators [2] that use the spatial index to perform spatial queries, Storage, management, and use of geodetic data.
- Open, standard SQL; XML; and XQuery access to spatial operations and data.

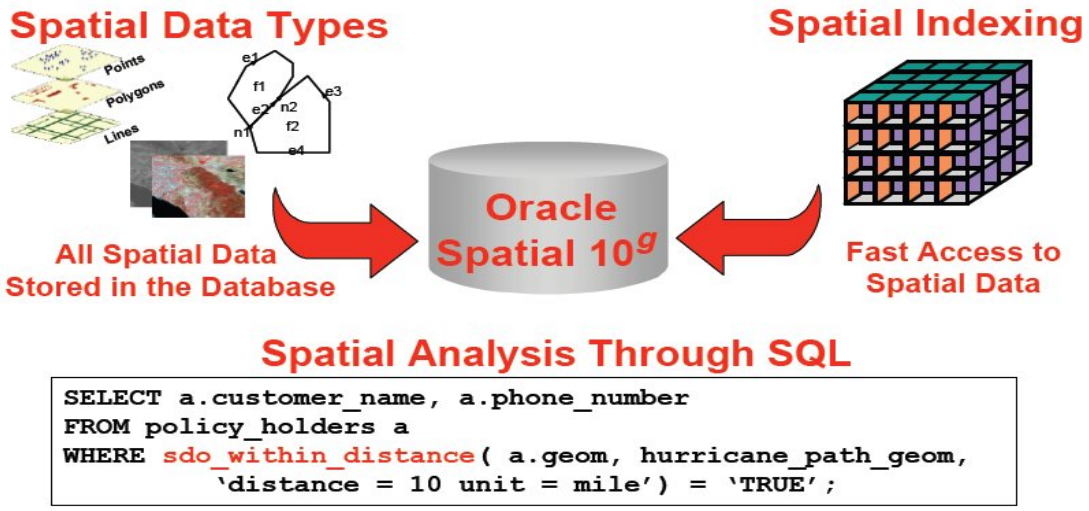

Figure 1.5: Oracle 10g Core Spatial Capabilities [6]

1.2.2 Oracle 10g Locator and Spatial

Oracle Locator is best suited for spatially enabling the majority of enterprise applications that have relatively straightforward geospatial requirements. Oracle 10g Standard Edition with Locator provides foundation-level geospatial data management.

For more complex applications, such as GIS, Oracle recommends Oracle Spatial [3], an option to Oracle 10g Enterprise Edition. Oracle Spatial Option provides advanced spatial functionality, such as spatial functions (including area, buffer, and centroid calculations), advanced coordinate systems support, and linear referencing systems; topology; aggregate functions; network data models/operations; routing; as well as geocoding. Oracle Spatial also supports direct interfaces with all major GIS systems, including those from ESRI (Environmental Systems Research Institute), Autodesk, Intergraph, and MapInfo.

The focus of the Oracle Spatial features is to address the requirements of enterprise geospatial solutions customers by providing the following six capabilities**:**

- **Network data model:** A data model is provided to store network (graph) structure in Oracle Database 10g. It explicitly stores and maintains

connectivity of link-node networks and provides network analysis capability such as shortest path and connectivity analysis.

- **Routing:** For transportation applications, the network data model also supports a routing feature. Oracle introduces a scalable routing engine that provides driving distances, times, and directions between addresses (or locations that have been geocoded in advance). It is provided as a Java client library to the network data model that can be deployed in either Oracle Application Server or standalone Oracle Application Server Containers for J2EE (OC4J) environments. Other features include preference for either fastest or shortest routes, returning summary or detailed driving directions, and returning the time and distance along a street network from a single location to multiple destinations.
- **Topology data model:** Oracle Spatial includes a data model and schema that persistently store topology in Oracle Database. This is useful when there is a high degree of feature editing and a strong requirement for data integrity across maps and map layers. Another benefit is that topology-based queries typically perform faster for queries involving relationships such as adjacency, connectivity, and containment. Land management (cadastral) systems and spatial data providers benefit from these capabilities.
- **GeoRaster:** A new data type natively manages geo referenced raster imagery (e.g., satellite imagery, remotely sensed data, and gridded data) in Oracle Database 10g and now in Oracle Database 11g also. Oracle Spatial's GeoRaster feature provides geo-referencing of imagery; XML schema for metadata management; and basic operations such as pyramiding, tiling, and interleaving. Applications in environmental management, defence/homeland security, energy exploration, and satellite image portals may benefit.
- **Spatial analytic functions:** New server-based spatial analysis capabilities include classification, binning, association, and spatial correlation — essential for business intelligence applications.

1.3 Visualization of Oracle Spatial Data

Another important aspect of Oracle Spatial is the visualization of spatial data using maps. Oracle uses the map-based visualization of spatial data using Oracle

MapViewer supported by Oracle MapBuilder tool. These tools provide a variety of features that make it possible the visualization and analysis of spatial data in various types of applications. Figure 1.6 shows a map (a visual object) generated by the MapViewer.

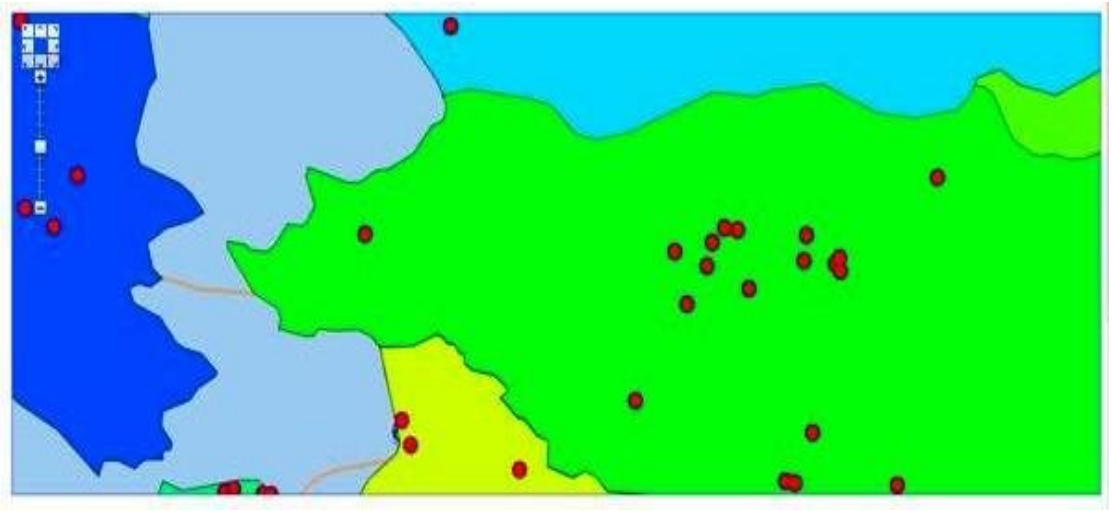

Figure 1.6: Map Generated by MapViewer

Chapter 2

Literature Review

2.1 Oracle Spatial

Oracle Spatial, referred to as spatial, is an integrated set of functions and procedures that enables spatial data to be stored, accessed, and analyzed quickly and efficiently in an Oracle database.

Oracle Spatial provides a SQL schema and functions that facilitate the storage, retrieval, update, and query of collections of spatial features in an Oracle database. Spatial consists of the following:

- A schema (MDSYS) that prescribes the storage, syntax, and semantics of supported geometric data types.
- A spatial indexing mechanism.
- Operators, functions, and procedures for performing area-of-interest queries, spatial join queries, and other spatial analysis operations.
- Functions and procedures for utility and tuning operations.
- Topology data model [4, 24] for working with data about nodes, edges, and faces in a topology.
- Network data model [4, 24] for representing capabilities or objects that are modeled as nodes and links in a network.
- GeoRaster [4, 20], a feature that make possible to store, index, query, analyze, and deliver GeoRaster data, that is, raster image and gridded data and its associated metadata.

2.1.1 Object-Relational Model

Spatial supports the object-relational model [3] for representing geometries. This model stores an entire geometry in the Oracle native spatial data type for vector data, SDO_GEOMETRY [2]. An Oracle table can contain one or more SDO_GEOMETRY columns. The object-relational model corresponds to a "SQL with Geometry Types" implementation of spatial feature tables in the Open GIS ODBC/SQL specification for geospatial features.

2.1.2 Spatial Data

Oracle Spatial [3] is designed to make spatial data management easier and more natural to users of location-enabled applications and geographic information system (GIS) applications. Once spatial data is stored in an Oracle database, it can be easily manipulated, retrieved, and related to all other data stored in the database.

A common example of spatial data can be seen in a road map. A road map is a two-dimensional object that contains points, lines, and polygons that can represent cities, roads, and political boundaries such as states or provinces. A road map is a visualization of geographic information.

The data that indicates the Earth location (such as longitude and latitude) of these rendered objects is the spatial data. When the map is rendered, this spatial data is used to project the locations of the objects on a two-dimensional piece of paper. A GIS is often used to store, retrieve, and render this Earth-relative spatial data. Types of spatial data (other than GIS data) that can be stored using Spatial include data from computer-aided design (CAD) and computer-aided manufacturing (CAM) systems. Instead of operating on objects on a geographic scale, CAD/CAM systems work on a smaller scale, such as for an automobile engine or printed circuit boards.

2.1.3 Geometry Types

Geometry is an ordered sequence of vertices that are connected by straight line segments or circular arcs. The semantics of the geometry are determined by its type. Spatial supports several primitive types, and geometries composed of collections of these types, including two-dimensional:

- Points and point clusters
- Line strings
- n-point polygons
- Arc line strings (All arcs are generated as circular arcs.)
- Arc polygons
- Compound polygons
- Compound line strings
- Circles
- Optimized rectangles

Two-dimensional points are elements composed of two ordinates, X and Y, often corresponding to longitude and latitude. Line strings are composed of one or more pairs of points that define line segments. Polygons are composed of connected line strings that form a closed ring, and the area of the polygon is implied. For example, a point might represent a building location; a line string might represent a road or flight path.

Figure 2.1, illustrates the geometric types.

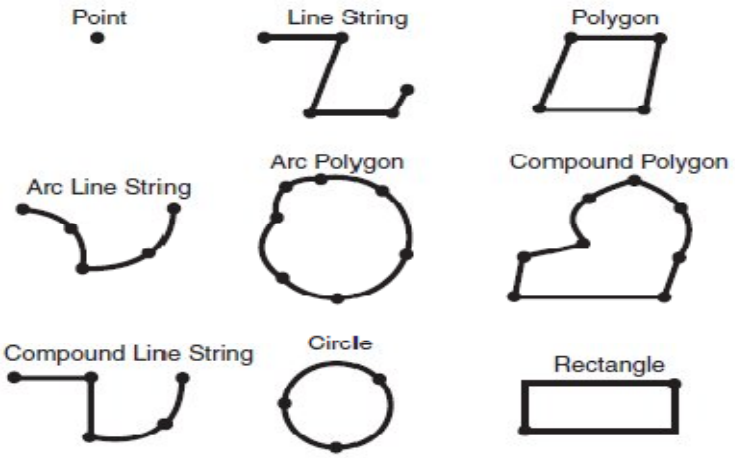

Figure 2.1: Geometric Types [3]

Spatial also supports the storage and indexing of three-dimensional and four-dimensional geometric types.

2.1.4 Data Model

The Spatial data model [3] is a hierarchical structure consisting of elements, geometries, and layers. Layers are composed of geometries, which in turn are made up of elements.

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. The exterior ring and the interior ring of a polygon with holes are considered as two distinct elements that together make up a complex polygon.

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. Coordinates are defined in order around the polygon (counterclockwise for an exterior polygon ring, clockwise for an interior polygon ring).

Geometry: A geometry (or geometry object) is the representation of a spatial feature, modeled as an ordered set of primitive elements. Geometry 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. A multipolygon, such as one used to represent a set of islands, is a homogeneous collection. A heterogeneous collection is one in which the elements are of different types, for example, a point and a polygon.

An example of geometry might describe the buildable land in a town. This could be represented as a polygon with holes where water or zoning prevents construction.

Layer: A layer is a collection of geometries having the same attribute set. 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). The geometries and associated spatial index for each layer are stored in the database in standard tables.

Coordinate System: A coordinate system (also called a spatial reference system) is a means of assigning coordinates to a location and establishing 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. Any spatial data has a coordinate system associated with it. The coordinate system can be 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). If the coordinate system is georeferenced, it has a default unit of measurement (such as meters) associated with it, but we can have Spatial automatically return results in another specified unit (such as miles).

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. If a coordinate system is not explicitly associated with geometry, a Cartesian coordinate system is assumed.

- 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. There are many such mathematical mappings, each used for a particular purpose.
- 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.

When performing operations on geometries, Spatial uses either a Cartesian or curvilinear computational model, as appropriate for the coordinate system associated with the spatial data.

Tolerance: Tolerance [3] is used to associate a level of precision with spatial data. Tolerance reflects the distance that two points can be apart and still be considered the same (for example, to accommodate rounding errors). The tolerance value must be a positive number greater than zero. The significance of the value depends on whether or not the spatial data is associated with a geodetic coordinate system.

- For geodetic data (such as data identified by longitude and latitude coordinates), the tolerance value is a number of meters. For example, a tolerance value of 100 indicates a tolerance of 100 meters. The tolerance value for geodetic data should not be smaller than 0.05 (5 centimetres), and in most cases it should be larger. Spatial uses 0.05 as the tolerance value for geodetic data if we specify a smaller value.
- For non-geodetic data, the tolerance value is a number of the units that are associated with the coordinate system associated with the data. For example, if the unit of measurement is miles, a tolerance value of 0.005 indicates a tolerance of 0.005 (that is, 1/200) mile (approximately 26 feet), and a

tolerance value of 2 indicates a tolerance of 2 miles. In both cases, the smaller the tolerance value, the more precision is to be associated with the data.

A tolerance value is specified in two cases:

- In the geometry metadata definition for a layer.
- As an input parameter to certain functions.

2.1.5 Query Model

Spatial uses a two-tier query model [3] to resolve spatial queries and spatial joins. The term is used to indicate that two distinct operations are performed to resolve queries. The output of the two combined operations yields the exact result set.

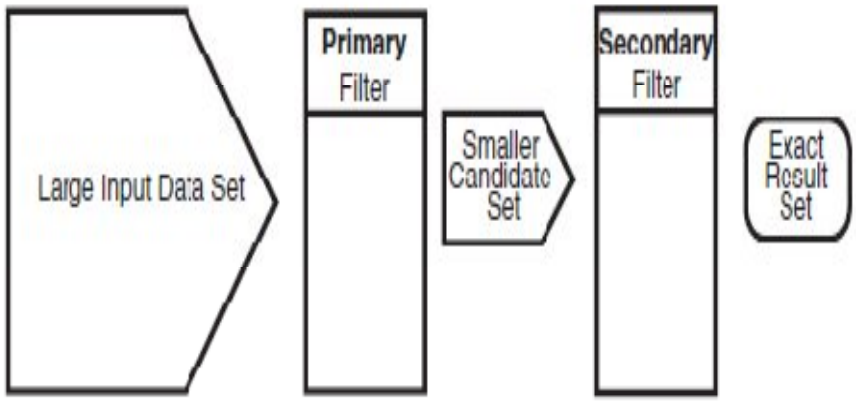

Figure 2.2: Query Model [3]

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 yields an accurate 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 2.2 shows the complete process sequentially.

The primary filter operation on a large input data set produces a smaller candidate set, which contains at least the exact result set and may contain more records. The secondary filter operation on the smaller candidate set produces the exact result set. Spatial uses a spatial index to implement the primary filter. Spatial does not require the use of both the primary and secondary filters. In some cases, just using the primary filter is sufficient. For example, a zoom feature in a mapping application queries for data that has any interaction with a rectangle representing visible boundaries. The primary filter very quickly returns a superset of the query. The mapping application can then apply clipping routines to display the target area. The purpose of the primary filter is to quickly create a subset of the data and reduce the processing burden on the secondary filter. The primary filter, therefore, should be as efficient (that is, selective yet fast) as possible. This is determined by the characteristics of the spatial index on the data.

2.1.6 Indexing of Spatial Data

The introduction of spatial indexing capabilities into the Oracle database engine is a key feature of the spatial product. A spatial index, like any other index, provides a mechanism to limit searches, but in this case the mechanism is based on spatial criteria such as intersection and containment. A spatial index is needed to:

- Find objects within an indexed data space that interact with a given point or area of interest (window query).
- Find pairs of objects from within two indexed data spaces that interact spatially with each other (spatial join).

A spatial index [2, 3] is considered a logical index. The entries in the spatial index are dependent on the location of the geometries in a coordinate space, but the index values are in a different domain. Index entries may be ordered using a linearly ordered domain, and the coordinates for geometry may be pairs of integer, floating-point, or double-precision numbers.

2.1.7 Spatial Relationships and Filtering

Spatial uses secondary filters to determine the spatial relationship between entities in the database. The spatial relationship is based on geometry locations. The most common spatial relationships are based on topology and distance. For example, the

boundary of an area consists of a set of curves that separates the area from the rest of the coordinate space. The interior of an area consists of all points in the area that are not on its boundary. Given this, two areas are said to be adjacent if they share part of a boundary but do not share any points in their interior. The distance between two spatial objects is the minimum distance between any points in them. Two objects are said to be within a given distance of one another if their distance is less than the given distance.

To determine spatial relationships, Spatial has several secondary filter methods:

- The SDO_RELATE operator evaluates topological criteria.
- The SDO_WITHIN_DISTANCE operator determines if two spatial objects are within a specified distance of each other.
- The SDO_NN [5] operator identifies the nearest neighbours for a spatial object.

The SDO_RELATE operator implements a nine-intersection model for categorizing binary topological relationships between points, lines, and polygons. Each spatial object has an interior, a boundary, and an exterior. The boundary consists of points or lines that separate the interior from the exterior. The boundary of a line string consists of its end points; however, if the end points overlap (that is, if they are the same point), the line string has no boundary. The boundaries of a multiline string are the end points of each of the component line strings; however, if the end points overlap, only the end points that overlap an odd number of times are boundaries. The boundary of a polygon is the line that describes its perimeter. The interior consists of points that are in the object but not on its boundary, and the exterior consists of those points that are not in the object.

Some of the topological relationships (as shown in figure 2.3) have been described below. Spatial uses the following names:

- DISJOINT -- The boundaries and interiors do not intersect.
- TOUCH -- The boundaries intersect but the interiors do not intersect.
- OVERLAPBDYDISJOINT -- The interior of one object intersects the boundary and interior of the other object, but the two boundaries do not intersect. This relationship occurs, for example, when a line originates outside a polygon and ends inside that polygon.

- OVERLAPBDYINTERSECT -- The boundaries and interiors of the two objects intersect.
- EQUAL -- The two objects have the same boundary and interior.
- CONTAINS -- The interior and boundary of one object is completely contained in the interior of the other object.
- COVERS -- The interior of one object is completely contained in the interior or the boundary of the other object and their boundaries intersect.
- INSIDE -- The opposite of CONTAINS. A INSIDE B implies B CONTAINS A.
- COVEREDBY -- The opposite of COVERS. A COVEREDBY B implies B COVERS A.
- ON -- The interior and boundary of one object is on the boundary of the other object (and the second object covers the first object). This relationship occurs, for example, when a line is on the boundary of a polygon.
- ANYINTERACT -- The objects are non-disjoint.

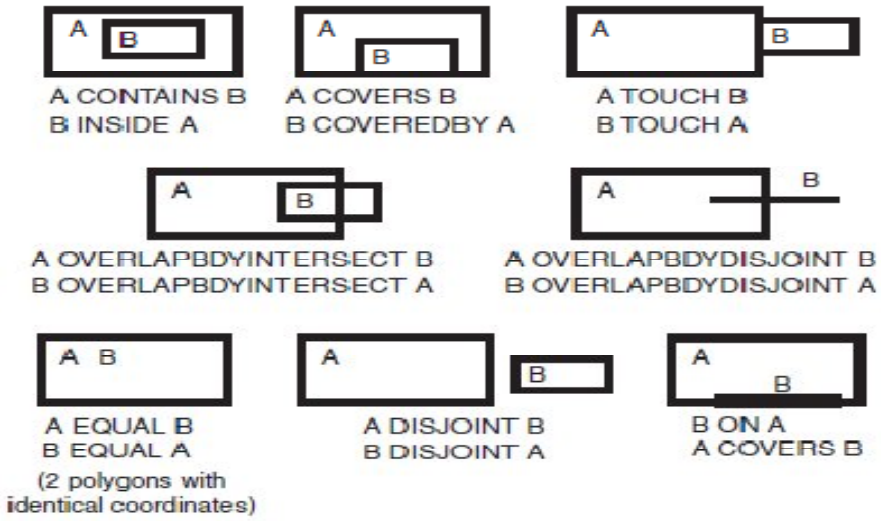

Figure 2.3: Topological Relationships [3]

The SDO_WITHIN_DISTANCE operator determines if two spatial objects, A and B, are within a specified distance of one another. This operator first constructs a distance buffer, D_b, around the reference object B. It then checks that A and D_b are non-disjoint.

The distance buffer of an object consists of all points within the given distance from that object. Figure 2.4 shows the distance buffers for a point, a line, and a polygon.

Figure 2.4: Distance Buffers for Points, Lines, and Polygons [3]

The detail of point, line, and polygon geometries as shown in Figure 2.4 is given below:

- The dashed lines represent distance buffers. The buffer is rounded near the corners of the objects.
- The geometry on the right is a polygon with a hole. The large rectangle is the exterior polygon ring and the small rectangle is the interior polygon ring (the hole). The dashed line outside the large rectangle is the buffer for the exterior ring, and the dashed line inside the small rectangle is the buffer for the interior ring.

The SDO_NN operator returns a specified number of objects from geometry columns that are closest to a specified geometry (for example, the five closest restaurants to a city park). In determining how close two geometry objects are, the shortest possible distance between any two points on the surface of each object is used.

2.1.8 Spatial Operators, Procedures and Function

The Spatial PL/SQL application programming interface (API) includes several operators and many procedures and functions. Spatial operators, such as SDO_FILTER and SDO_RELATE, provide optimum performance because they use the spatial index. Spatial procedures and functions are provided as subprograms in PL/SQL packages, such as SDO_GEOM, SDO_CS, and SDO_LRS. These subprograms do not require that a spatial index be defined, and they do not use a spatial index if it is defined.

2.1.9 Spatial Aggregate Functions

Oracle Spatial aggregate functions aggregate the results of SQL queries involving geometry objects. Spatial aggregate functions return a geometry object of type SDO_GEOMETRY. For example, the following statement returns the minimum bounding rectangle of all geometries in a table. The graph of the table is given in figure 2.5.

SELECT SDO_AGGR_MBR(shape) FROM cola_markets;

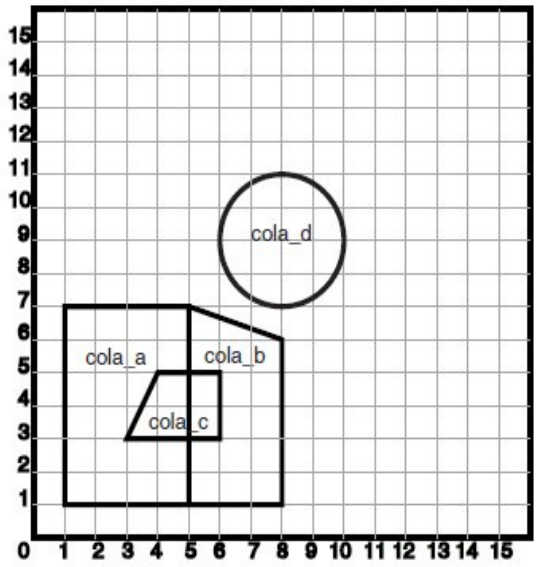

Figure 2.5: Cola Market Graph [3]

2.1.10 Geocoding

Geocoding [3, 19] is the process of converting tables of address data into standardized address, location, and possibly other data. The result of a geocoding operation includes the pair of longitude and latitude coordinates that correspond with the input address or location. For example, if the input address is 22 Monument Square, Concord, MA 01742, the longitude and latitude coordinates in the result of the geocoding operation may be (depending on the geocoding data provider) -71.34937 and 42.46101, respectively.

2.1.11 Spatial Java Application Programming Interface

Oracle Spatial provides a Java application programming interface (API) that includes the following packages:

- oracle.spatial.geometry provides support for the Spatial SQL SDO_GEOMETRY data type.
- oracle.spatial.network provides support for the Oracle Spatial network data model.
- oracle.spatial.topo provides support for the Oracle Spatial topology data model.
- oracle.spatial.util provides classes that perform miscellaneous operations.

2.1.12 MDDATA Schema

Effective with Oracle Database 10g, Spatial creates a user and schema named MDDATA, using the following internal SQL statements:

```
CREATE USER mddata IDENTIFIED BY mddata;
GRANT connect, resource TO mddata;
ALTER USER mddata ACCOUNT LOCK;
```

MDDATA schema [3] is used for storing data used by geocoding and routing applications. This is the default schema for Oracle software that accesses geocoding and routing data.

2.2 Oracle MapViewer

MapViewer [16, 18] is a Java-based visualization tool that is used to render, in map form, location based content and spatial data stored using the spatial feature of Oracle Database. It can also render map and related content from ESRI (Environmental Systems Research Institute) Shape Files, real-time XML feeds such as GeoRSS, and geographic web services including themes from OpenGeospatial Consortium Web Feature Services.

MapViewer can be used to:

- Create customized maps that show geographic features such as roads, city areas, waterways and other transportation networks.
- Display map themes such as national, state and local boundaries.

- Visualize business data (e.g. population demographics, psycho demographics, sales metrics, *etc*.), to portray and explore relationships that can often best be expressed graphically as geographic maps.
- Complement an applications workflow, providing interaction with mapped data.
- Deliver custom maps over the Internet as a component of JDeveloper or as a standalone tool [14].

2.2.1 Communication of Mapping Client with MapViewer

As shown in figure 2.6, there are two steps for communication of client with MapViewer; regardless of whether or not the client requests a map or some MapViewer administrative action.

For a map request:

- The client requests a map, passing in the map name, data source, centre location, map size, and, optionally, other data to be plotted on top of a map.
- The server returns the map image (or a URL for the image) and the minimum bounding rectangle (MBR) of the map, and the status of the request.

For a MapViewer administrative request:

- The client requests a MapViewer administrative action, passing in the specific type of request and appropriate input values.
- The server returns the status of the request and the requested information.

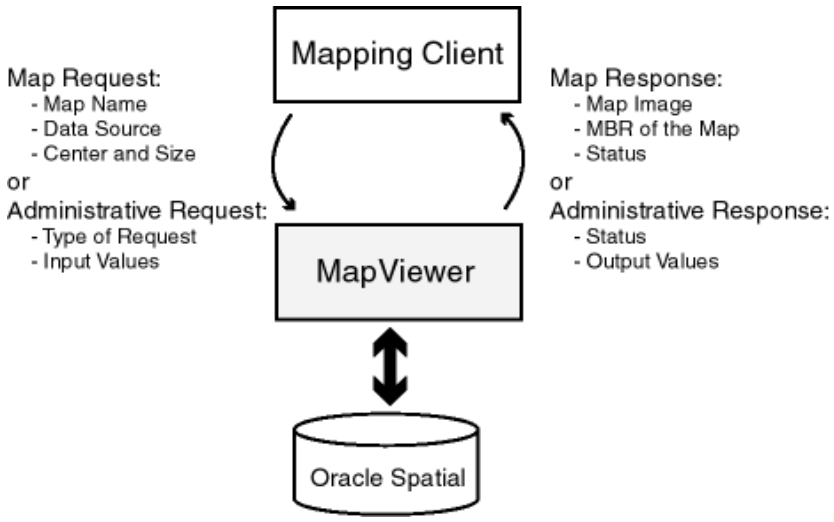

Figure 2.6: Mapping Client and MapViewer Communication [15]

2.2.2 MapViewer Architecture

The components of MapViewer have been described below on the basis of figure 2.7.

The core rendering engine connects to the Oracle database through Java Database Connectivity (JDBC). It also reads the map metadata (such as map definitions, styling rules, and symbologies created through the Map Builder tool) from the database, and applies the metadata to the retrieved spatial data during rendering operations.

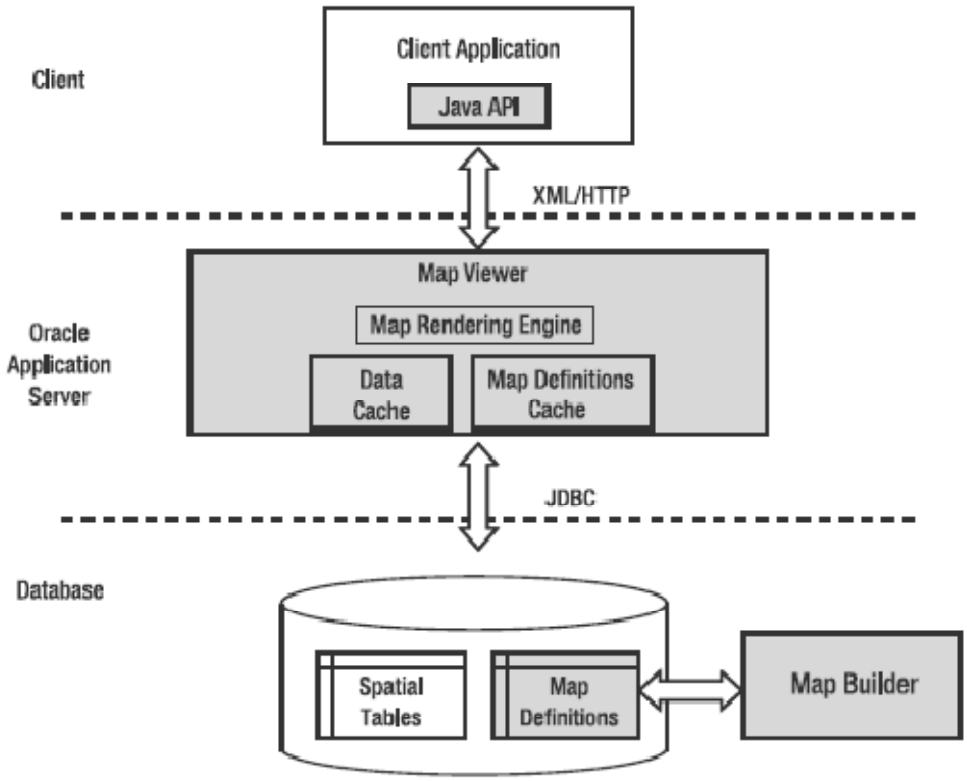

Figure 2.7: Architecture of MapViewer [16]

The XML API provides application developers with a versatile interface for submitting a map request to MapViewer and retrieving the map response. The JavaBean-based API and the PL/SQL API provide access to MapViewer's rendering capabilities. The JavaScript API enables us to create highly interactive web applications that use the Oracle Maps feature of MapViewer.

The Map Builder tool simplifies the process of creating and managing map, theme, and symbology metadata in a spatial database [13]. Oracle Maps, built on core MapViewer features, uses a map tile server that caches map image tiles, and a Feature

Of Interest (FOI) server that streams live data out of a database to be displayed as interactive features on a map. We can use the AJAX based JavaScript API with Oracle Maps to provide sophisticated mapping solutions.

Oracle Maps also allows for advanced customization and querying capabilities. The primary benefit of MapViewer is its integration with Oracle Spatial, Oracle Locator, and other Oracle Fusion Middleware components. MapViewer supports two-dimensional vector geometries stored in Oracle Spatial, as well as GeoRaster data and data in the Oracle Spatial topology and network data models. Oracle MapViewer is also an Open Geospatial Consortium (OGC)-compliant Web Map Service (WMS) server.

2.2.3 MapViewer Mapping Metadata

In MapViewer, a map conceptually consists of one or more themes. Each theme consists of a set of individual geographic features that share certain common attributes. Each feature is rendered and (optionally) labeled with specific styles. Themes can be predefined inside a database user's schema, or can be dynamically defined as part of a map request. Predefined themes can be grouped to form a predefined base map that can also be stored in a user's schema. Styles, predefined themes, and base maps are collectively called mapping metadata for MapViewer. This scheme provides a clear separation between the presentation of data and the spatial data itself. For example, any mistake made while manipulating the mapping metadata will have no effect on the corresponding spatial data, and vice versa.

Styles: A style [23] is a visual attribute that can be used to represent a spatial feature. The basic map symbols and labels for representing point, line, and area features are defined and stored as individual styles. Each style has a unique name and defines one or more graphical elements in XML syntax. Each style is of one of the following types:

- **COLOR:** a color for the fill or the stroke (border), or both.
- **MARKER:** a shape with a specified fill and stroke color, or an image. Markers are often icons for representing point features, such as airports, ski resorts, and historical attractions. When a marker style is specified for a line feature, the

rendering engine selects a suitable point on the line and applies the marker style to that point.

- **LINE:** a line style (width, color, end style, join style) and optionally a centre line, edges, and hash mark. Lines are often used for linear features such as highways, rivers, pipelines, and electrical transmission lines.
- **AREA:** a color or texture, and optionally a stroke color. Areas are often used for polygonal features such as counties and census tracts.
- **TEXT:** a font specification (size and family) and optionally highlighting (bold, italic) and a foreground color. Text is often used for annotation and labelling (such as names of cities and rivers).
- **ADVANCED:** a composite used primarily for thematic mapping. The core advanced style is BucketStyle, which defines a mapping from a set of simple styles to a set of buckets. For each feature to be plotted, a designated attribute value from that feature is used to determine which bucket it falls into, and then the style associated with that bucket is used to plot the feature.

All styles are stored in a table of the system user MDSYS, but are exposed to each user through its own USER_SDO_STYLES view.

Any geographic feature, such as a road, can be displayed differently if alternate styles are assigned or applied, even though the underlying geometric structure of the feature itself is identical. Figure 2.8 is an example of a single road being rendered using three different line styles.

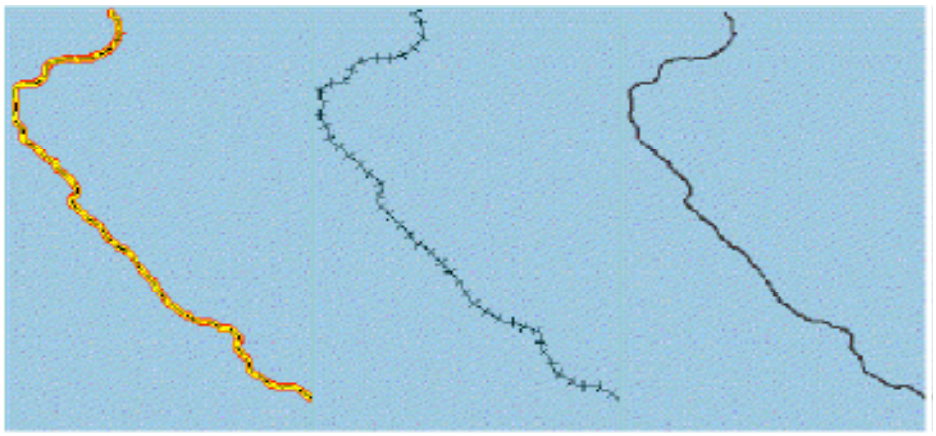

Figure 2.8: Same Geometry, Different Line Styles [17]

Themes: A theme is a visual representation of a particular data layer. Conceptually, a theme is a collection of geographic features that share similar attributes, plus the

rendering and labeling rules that tell MapViewer what styles to use to render and label the features. When a theme is defined, actually the following information is provided to MapViewer: where and how to get the data, and how to render and label the data.

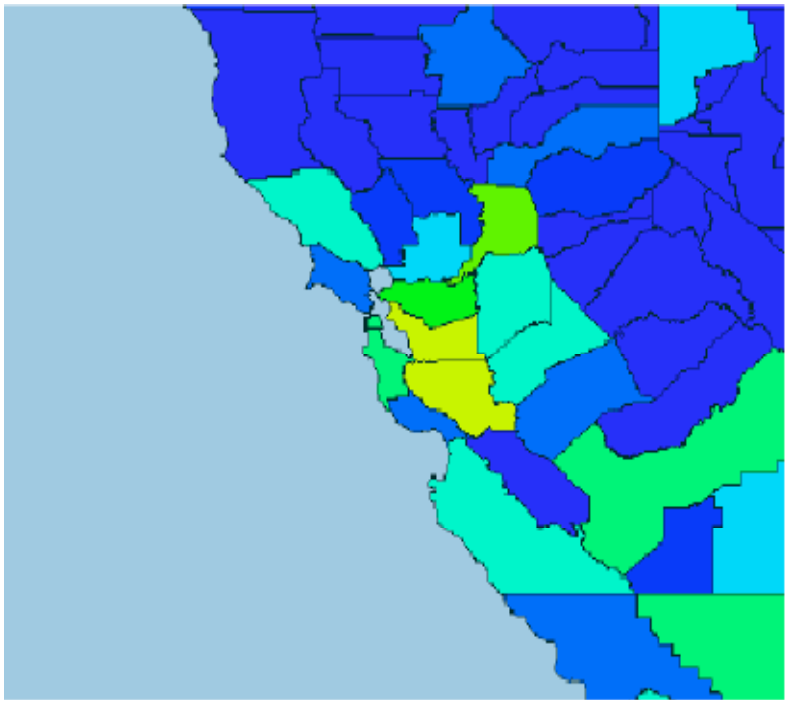

Figure 2.9: A Simple Thematic Map [17]

The figure 2.9 shows a rendered image in which each county area is scaled by color to reflect the population density value assigned to that county.

Base Maps: A map can consist of a combination of elements and attributes, such as the following:

- Background image
- Title
- Legend
- Query window
- Footnote (such as for a copyright notice)
- Base map
- Predefined themes (in addition to any in the base map)
- JDBC themes (with dynamic queries)
- Dynamically defined (temporary) styles

These elements and attributes, when specified in a map request, define the content and appearance of the generated map. A map can have a base map and a stack of themes rendered on top of each other in a window. A map has an associated coordinate system that all themes in the map must share. For example, if the maps coordinate system is 8307 (for Longitude / Latitude (WGS 84), the most common system used for GPS devices), all themes in the map must have geometries defined using that coordinate system. Themes can be added to a map by specifying a base map name or by using the programming interface to add themes. The order in which the themes are added determines the order in which they are rendered, with the last specified theme on top, so be sure we know which themes we want in the background and foreground.

Predefined themes can be grouped together to form a base map. This provides a convenient way to include multiple themes in a map request. Only predefined themes can be included in a base map. The base map definitions are stored in a user's USER_SDO_MAPS view.

A minimum and maximum map scale can be provided for each theme listed in a base map. This provides a powerful mechanism that is used to selectively reveal themes based on the current map's scale.

Tile Layers: The fourth type of mapping metadata is Map Tile Layer metadata. This metadata is primarily used by the Oracle Maps JavaScript API. It provides the JavaScript API with information about a draggable map tile layer, including its geographic boundary, coordinate system, number of discrete zoom levels and the size and format of individual map tiles at each zoom level.

A map tile layer is typically associated with a MapViewer base map. This type of map tile layer is often called "Internal Map Tile Layer". Log in to the MapViewer Admin page and open the "Manage Map Tile Layers" tab to create a new internal map tile layer. The tile layer creation wizard will guide us through a series of steps. If we make changes to any of the themes and styles used to define a base map associated with a map tile layer then the existing map tile files must be deleted from the location on disk specified when creating the tile layer and the user's browser cached files must be purged. We can create a map tile layer that gets its map tiles from an external source, such as a 3rd party web mapping service.

2.2.4 Data Sources

A data source corresponds to a database schema or user. Before we can draw any spatial data in a database schema, we must first define (create) a data source for the schema, either permanently or dynamically:

- We can define a data source permanently by specifying its connection information and user login credentials in the MapViewer configuration file named as mapViewerConfig.xml.
- We can define or modify a data source dynamically using the MapViewer administration (Admin) page.

Each map request must specify a master data source. We can, however, specify a different data source for individual themes added to the map request. This makes it easy to aggregate data stored across different database schemas. If a theme has no specified data source, it is associated with the master data source. A base map (and thus the themes included in it) is always associated with the master data source. When a theme is processed, all of its underlying data, as well as the styles referenced in its definition, must be accessible from the data source or sources associated with the theme.

Each data source has its own internal metadata cache. The metadata cache holds the definitions of all accessed styles, as well as of all predefined themes that originate from the data source. This eliminates the need to query the database repeatedly for the definition of a style or predefined theme whenever it is needed.

2.2.5 Map Generation Process

When a map request arrives at the MapViewer server, the server picks a free renderer associated with the master data source in the request. This section describes the process that the MapViewer server follows to generate a map. In brief, MapViewer performs the following steps:

- Parse and process the incoming XML map request.
- Prepare the data for each theme (executed in parallel).
- Render and label each theme.
- Generate final images or files.

Each map generated by MapViewer results from its receiving a valid XML map request. (If we use the JavaBean-based API, the request is automatically converted to an XML document and passed to the MapViewer server.) The XML map request is parsed and its content is validated. MapViewer then creates any dynamic styles specified in the XML request. It builds a theme list from all themes included in the base map (if a base map is specified), as well as any specified predefined or JDBC themes. All individual features in the request are grouped into a single temporary theme. In other words, after parsing the incoming request, all data that must be shown on the map is presented in a list of themes to the MapViewer rendering engine. The ordering of the themes in the list is important, because it determines the order in which the themes are rendered. All themes included in the base map (when present) are added to the list first, followed by all specified themes (predefined or JDBC). The theme that contains all the individual features is added as the last theme on the list. Any other requested features of a map (such as legend, map title, or footnote), are created and saved for rendering later. For each theme in the request, MapViewer then creates a separate execution thread to prepare its data, so that preparation of the themes takes place in parallel. For a predefined theme, this means formulating a query based on the theme's definition and any other information, such as the current map request window. This query is sent to the database for execution, and the result set is returned. MapViewer creates individual renderable objects based on the result set.

2.2.6 Oracle Maps

Oracle Maps is the name for a suite of technologies for developing high-performance interactive Web-based mapping applications. Oracle Maps is included with MapViewer.

Oracle Maps consists of the following main components as shown in figure 2.10:

- A map tile server that caches and serves pregenerated map image tiles.
- A FOI (feature of interest) server that renders geospatial features that are managed by Oracle Spatial.
- An Ajax-based JavaScript mapping client. (Ajax is an acronym for asynchronous JavaScript and XML.) This client provides functions for browsing and interacting with maps, as well as a flexible application programming interface (API).

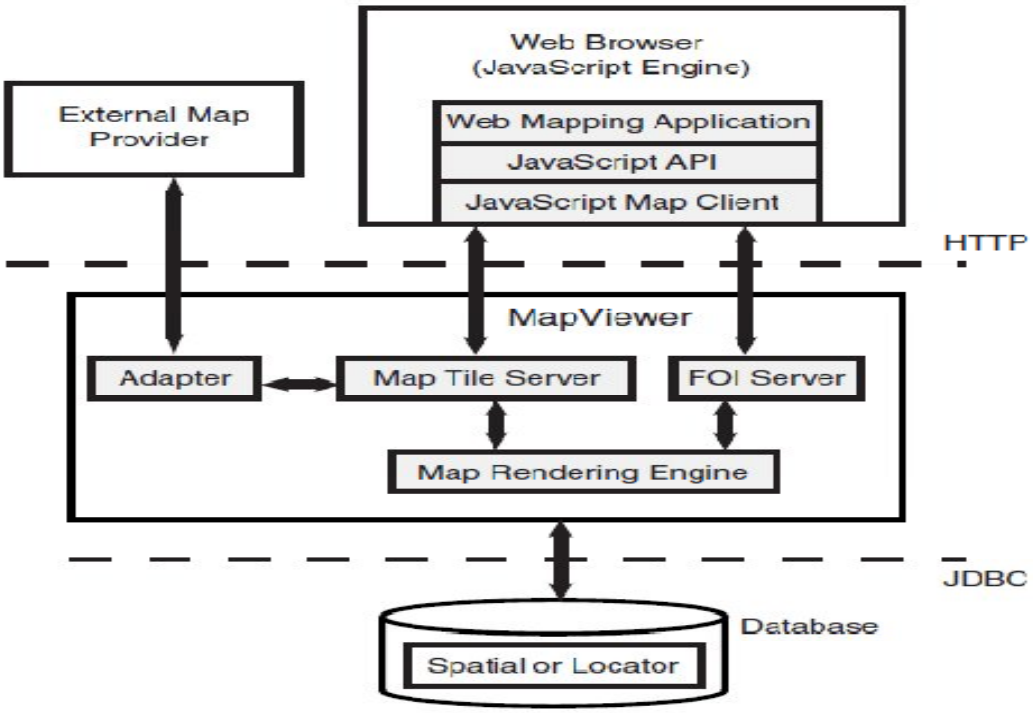

Figure 2.10: Architecture for Oracle Maps Applications [16]

Applications interact with the Oracle Maps architecture as follows:

- The application is developed using JavaScript, and it runs inside the JavaScript engine of the Web browser.
- The application invokes the JavaScript map client to fetch the map image tiles from the map tile server, and then it displays the map in the Web browser.
- The application invokes the JavaScript map client to fetch dynamic spatial features from the FOI server and display them on top of the map tiles.
- The JavaScript map client controls map-related user interaction for the application.
- When the map tile server receives a map image tile request, it first checks to see if the requested tile is already cached. If the tile is cached, the cached tile is returned to the client. If the tile is not cached, the map tile server fetches the tile into the cache and returns it to the client. Tiles can be fetched either directly from the MapViewer map rendering engine or from an external Web map services provider.
- When the FOI server receives a request, it uses the MapViewer map rendering engine to generate the feature images and to send these images, along with feature attributes, to the client.

2.3 Oracle Map Builder

Oracle Map Builder [2, 16] is a standalone application that lets us create and manage the Oracle Application Server MapViewer mapping metadata (styles, themes, and base maps) in the database. Figure 2.11 is explaining the role of Map Builder. For example, we can use this tool to create a style, theme or base map or modify its definition. Besides handling the metadata, the tool provides interfaces to preview the metadata (for example, to see how a line style will appear on a map or how is the theme rendered based on its styling rules), and display the original spatial information from the spatial tables without creating MapViewer metadata. A set of wizards is also available to create metadata based on database table contents. Existing metadata can be edited and previewed, existing spatial tables can be previewed directly, spatial table contents can be used in wizards to generate MapViewer metadata (advanced styles and themes), and import tools can be used to store metadata (True type fonts as style information) and also to generate new spatial tables (shape files generate tables with SDO_GEOMETRY type, image files generate tables with SDO_GEORASTER type).

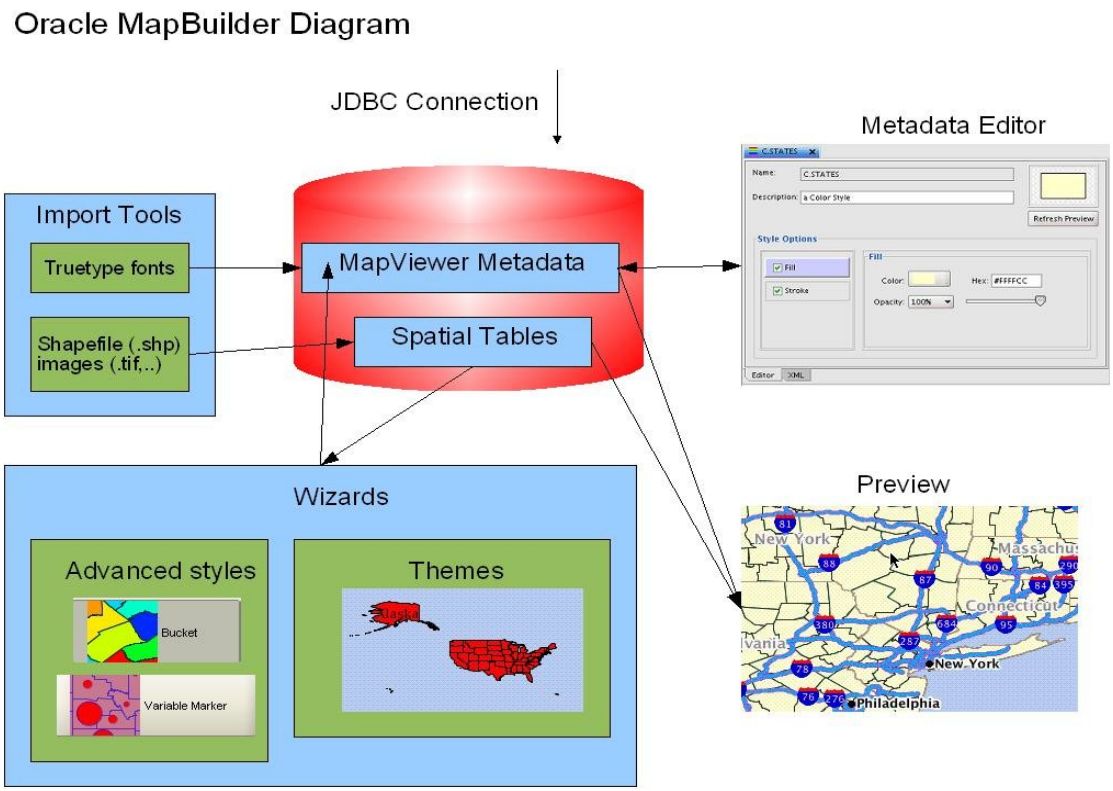

Figure 2.11: Oracle Map Builder Diagram [2]

2.3.1 The User Interface of Map Builder Tool

This section briefly describes the various elements of the MapBuilder tool's UI (Figure 2.12). These are the Menus, the toolbar, the database connection drop-down, the metadata and data navigator trees on the left, and editor/preview and message panels on the right.

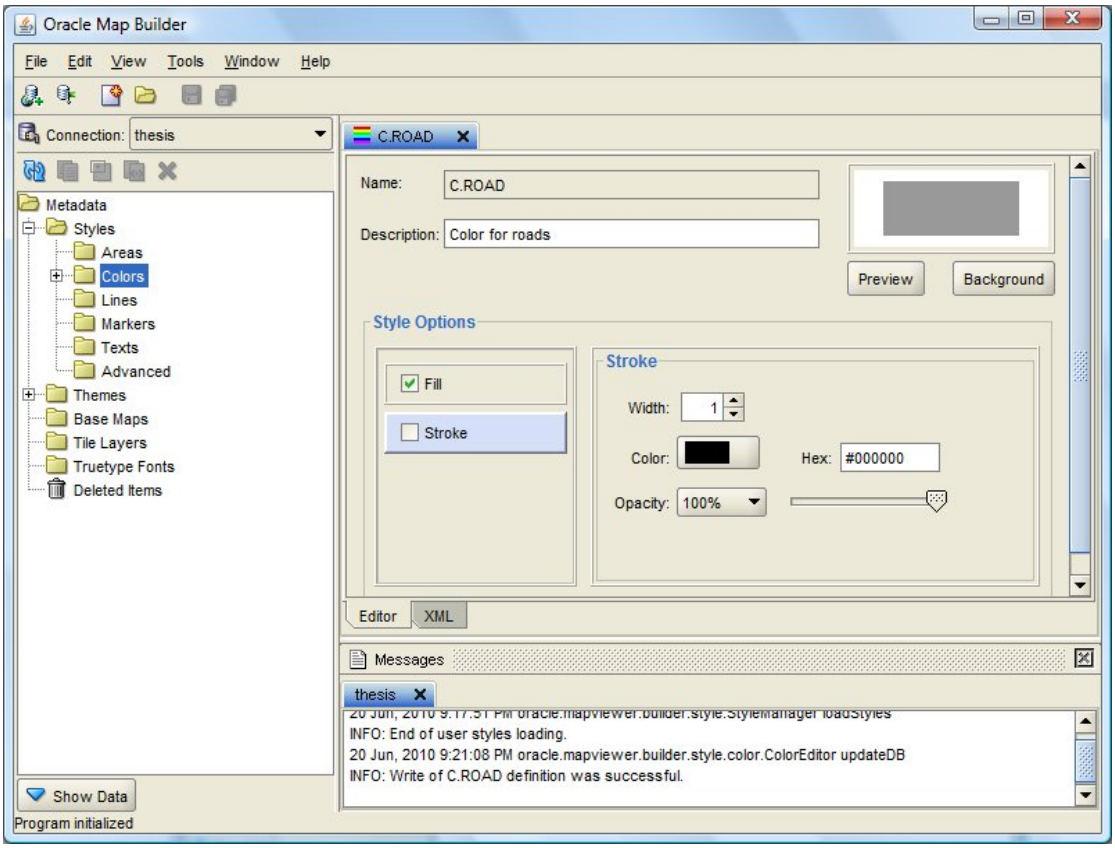

Figure 2.12: Map Builder User Interface

Menus

The Menu entries at the top are:

- **File:** contains commands to create/edit database connections and metadata.
- **Edit:** contains some edit commands such as undo/copy/paste that are reserved for future use. The only operation available is the Delete option to erase metadata.
- **View:** contains commands to preview metadata and to refresh the connection contents.
- **Tools:** contains submenus and commands to import external data (images such as tif files, or vector data such as Shapefiles), operations to import/export MapViewer metadata (styles, themes, base maps), operation to register external

spatial providers to render data in native format, and operations to update some configuration parameters on the preferences file.

- **Window:** contains submenus and commands to show/hide some user interface components.
- **Help:** submenus to access MapBuilder online help and to get the version information.

Toolbar

The Toolbar entries under the menu options are used for creating, loading or removing database connections; creating or opening metadata information; and to save edited metadata.

Connection List

The Connection drop-down lets us establish a database connection, use a different connection, create a new connection, or edit an existing connection.

Metadata Navigator

The Metadata Navigator tree is a hierarchical display of MapViewer metadata objects (styles, themes, base maps, truetype fonts) read from the currently active database connection. To select an object, expand the appropriate tree node or nodes, and then click the object. The nodes refer to:

- **Styles:** style information stored in the USER_SDO_STYLES view.
- **Themes:** theme information stored in the USER_SDO_THEMES view.
- **Base Maps:** base map information is stored in the USER_SDO_MAPS view.
- **Truetype Fonts:** true type fonts, stored in the USER_SDO_STYLES view, which can be used as marker styles.

Metadata Icons

Above the Metadata Navigator tree there are a set of icons related to this tree:

- **Refresh:** refreshes the contents of the navigator tree by loading the current database content for selected connection.
- **Duplicate:** opens a dialog to duplicate the metadata definition.

- **Preview:** opens a preview window on the right side of the application to display the theme or base map contents.
- **Copy XML:** this is currently not supported.
- **Delete:** removes the metadata definition from the database.

Spatial Data Navigators

The Spatial Tables Navigator shows the current spatial tables that are accessible for the selected connection. The nodes are organized by Oracle Spatial object types (ST_ANNOTATION_TEXT for annotation text tables, SDO_GEOMETRY for geometry tables, SDO_GEORASTER for GeoRaster tables and SDO_TOPO_GEOMETRY for topology tables). The spatial tables must be registered in corresponding metadata views (e.g. USER_SDO_GEOM_METADATA for geometry tables) in order to be shown on the tree.

The Spatial Models Navigator tree shows the current network and topology models that are accessible for the selected connection.

Editor Pages

The right side of the application may contain Editor Pages. Users can edit/preview the metadata contents for styles, themes and base maps.

Message Panel

Below the Editor page there is a Message window that displays log information for actions performed on the current connection.

Chapter 3

Problem Statement

3.1 Problem Statement

If we present directly a picture of the thing under consideration rather than presenting it in form of coordinates or textual format, it becomes more expressive and attractive way to make people know about the things because a picture or map is certainly worth 1000 words. Using the capabilities of Oracle Spatial concepts, MapViewer, and Map Builder, we can develop such spatial applications which incorporate the maps as main part of the whole. In India, the scene of spatial applications is not satisfactory; developers are still designing and developing traditional applications at very large scale. Also, the developers are not much aware of spatial applications and process of developing such applications as well. So, in this thesis work we have designed and developed a spatial application of Thapar University, in which we have clearly elaborated each step regarding spatial application development.

3.2 Objectives

We have considered the spatial application of Thapar University as target area. The tasks to design a map for Thapar University with Map Builder and to view any target location of Thapar University with navigation, zoom and distance measure options are considered as objectives to be achieved in this thesis work.

3.3 Methodology

The methodology used for the design and development of spatial application of Thapar University is as follows:

- Design a spatial database for the application with the use of Google maps tool to retrieve the coordinate information of Thapar University.
- Storage of spatial data into the database with the use of SQL Plus tool.
- Usage of Oracle Map Builder tool to define mapping metadata for Thapar University.
- Usage of MapViewer for creating data source for spatial database and creating map tile layer for the application.

- Design, Development and Deployment of the application with the use of HTML and JavaScript.

This methodology smoothly explores all the steps underlying the map-generation process (for any real world spatial object as Thapar University) like database-creation, metadata-creation, map server activation, application building *etc*.

Chapter 4

Design and Implementation of Thapar University Spatial Application

As we know that spatial data does not mere mean to be stored in databases and be queried by its users. It also includes other hidden capabilities. One of them is the visualization of spatial data by means of maps. Therefore, we have not only created a spatial database but developed a spatial application also to explore the visualization capability of spatial database. This application includes various map features in it. The process of developing such an application includes the use of various powerful applications and tools at different stages according to the requirement. As described in previous chapter, we have chosen 'Thapar University' as case study for designing and developing a spatial application. The description of tools and applications used are as follows:

- **Oracle 11g:** We have installed Oracle 11g and used SQL Plus component of this database management system for creating spatial database for the Thapar University.
- **Oracle Map Builder Tool:** This tool has been used for creating and managing the mapping metadata for the Thapar University spatial database.
- **Oracle Mapviewer:** Mapviewer is basically a map server. It handles all the map requests from the client applications.

4.1 Spatial Database Designing

We designed the spatial database for the Thapar University by identifying main locations of the university. These locations can be categorized as under:

- Academic blocks
- Hostels
- Roads
- Grounds
- Miscellaneous Locations

So, the database contains five tables corresponding to these sections. Each of these tables contains one spatial column declared with datatype SDO_GEOMETRY. Schema of each table is given below in subsequent tables:

Table 4.1: Schema of table STBL_BLOCK

S.No.	Column Name	Datatype
1.	block_id	Number (Primary Key)
2.	block_name	varchar2(20)
3.	Description	varchar2(25)
4.	block_shape	SDO_GEOMETRY

Table 4.2: Schema of table STBL_HOSTEL

S.No.	Column Name	Datatype
1.	hostel_id	Number (Primary Key)
2.	hostel_name	varchar2(20)
3.	Description	varchar2(25)
4.	hostel_shape	SDO_GEOMETRY

Table 4.3: Schema of table STBL_ROAD

S.No.	Column Name	Datatype
1.	road_id	Number (Primary Key)
2.	road_name	varchar2(20)
3.	Description	varchar2(25)
4.	road_shape	SDO_GEOMETRY

Table 4.4: Schema of table STBL_GROUND

S.No.	Column Name	Datatype
1.	ground_id	Number (Primary Key)
2.	ground_name	varchar2(20)
3.	Description	varchar2(25)
4.	ground_shape	SDO_GEOMETRY

Table 4.5: Schema of table STBL_MISC

S.No.	Column Name	Datatype
1.	misc_id	Number (Primary Key)
2.	misc_name	varchar2(20)
3.	Description	varchar2(25)
4.	misc_shape	SDO_GEOMETRY

After the designing of database, we have created a graph using the image of Thapar University (taken from Google maps tool) for the reference. The graph contains 50 blocks of length in horizontal direction and 28 blocks of width in vertical direction. Figure 4.1 shows the Thapar University on Google maps tool whereas figure 4.2 shows Thapar University on the graph.

Figure 4.1: Thapar University on Google maps tool

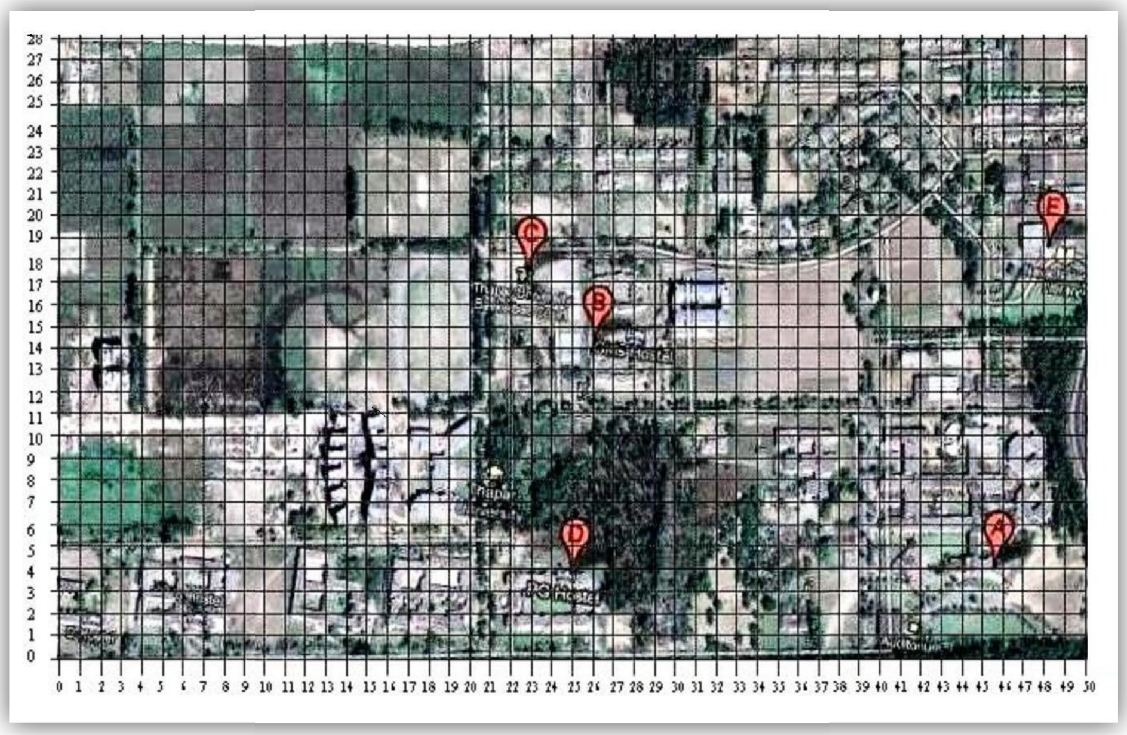

Figure 4.2: Thapar University on Graph

4.2 Working with SQL Plus: Table Creation and Record Insertion

First of all, we have created a database user named as 'thesis' and given password 'mudit' as shown in figure 4.3.

4.2.1 Creation of User and Tables

We have created database tables as shown in figure 4.4.

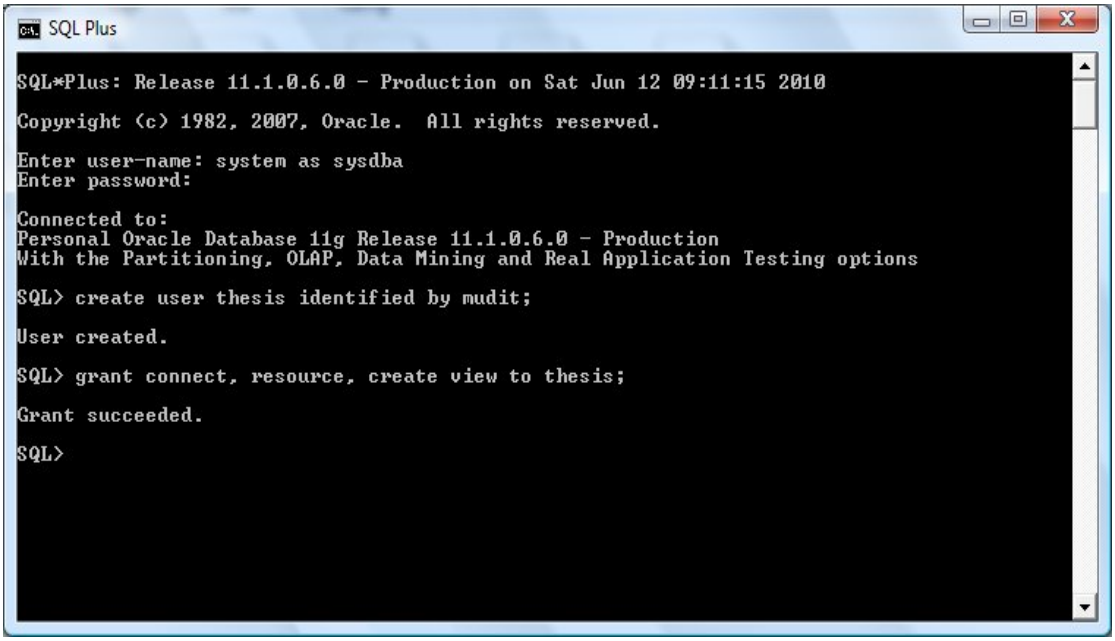

Figure 4.3: User Creation Queries Running in SQL Plus

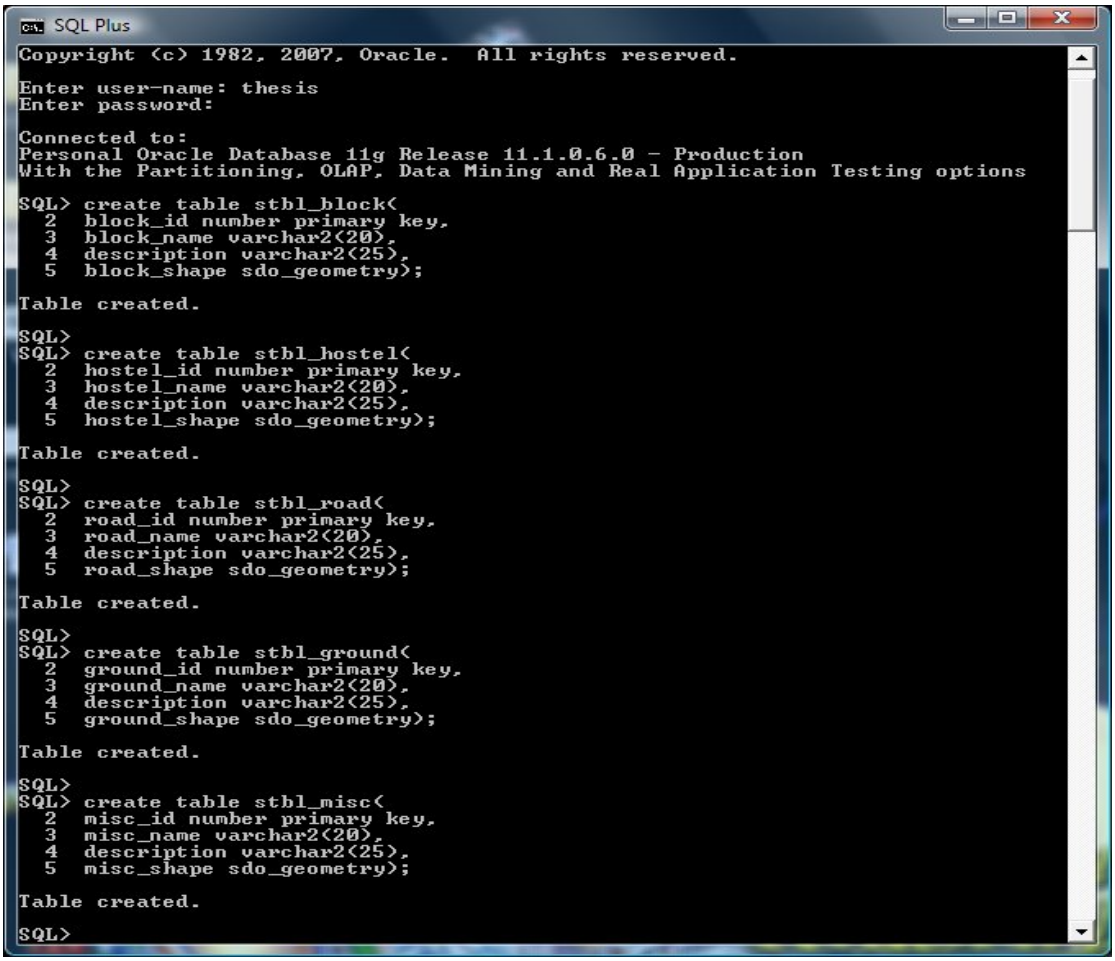

Figure 4.4: Table Creation Queries Running in SQL Plus

4.2.2 Metadata and Index Entries

We have made entries into USER_SDO_GEOM_METADATA view and created indexes for each table as shown in figure 4.5, and figure 4.6.

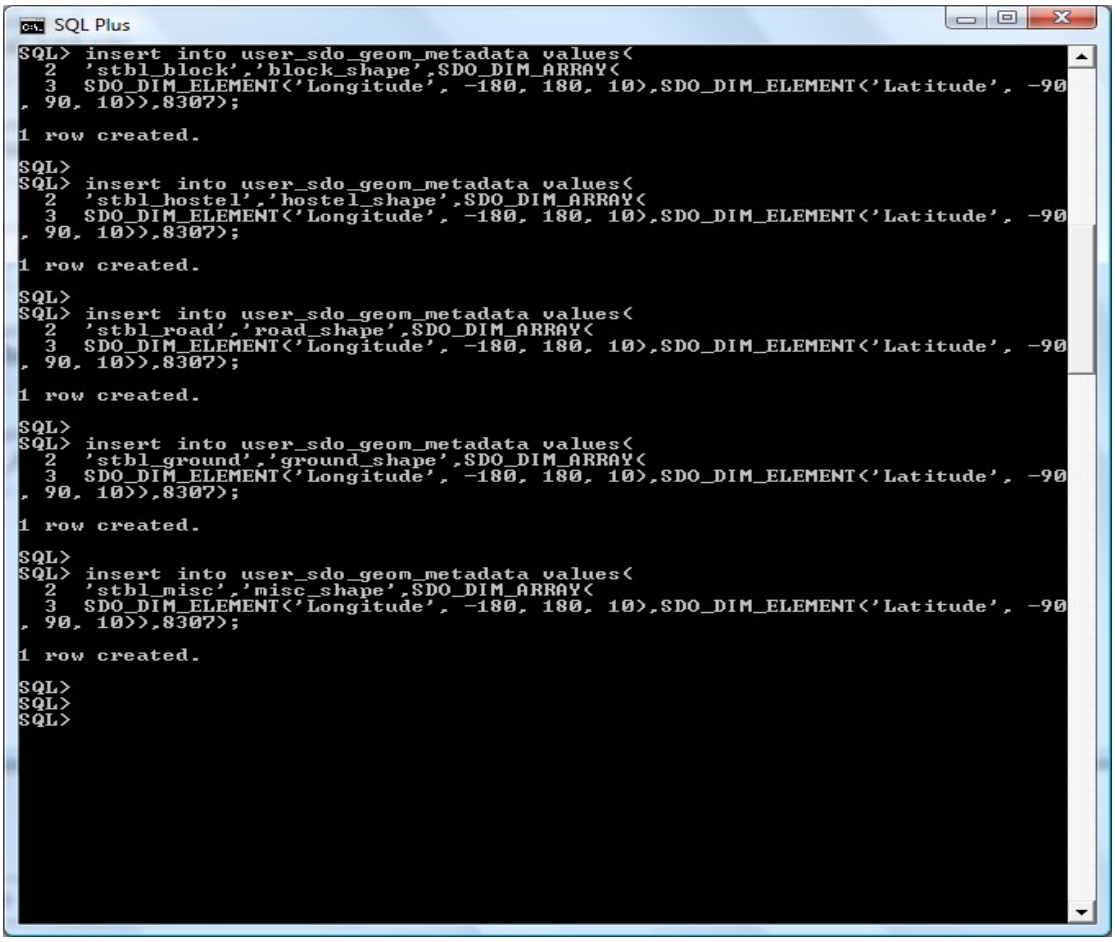

Figure 4.5: Insert Queries for USER_SDO_GEOM_METADATA View

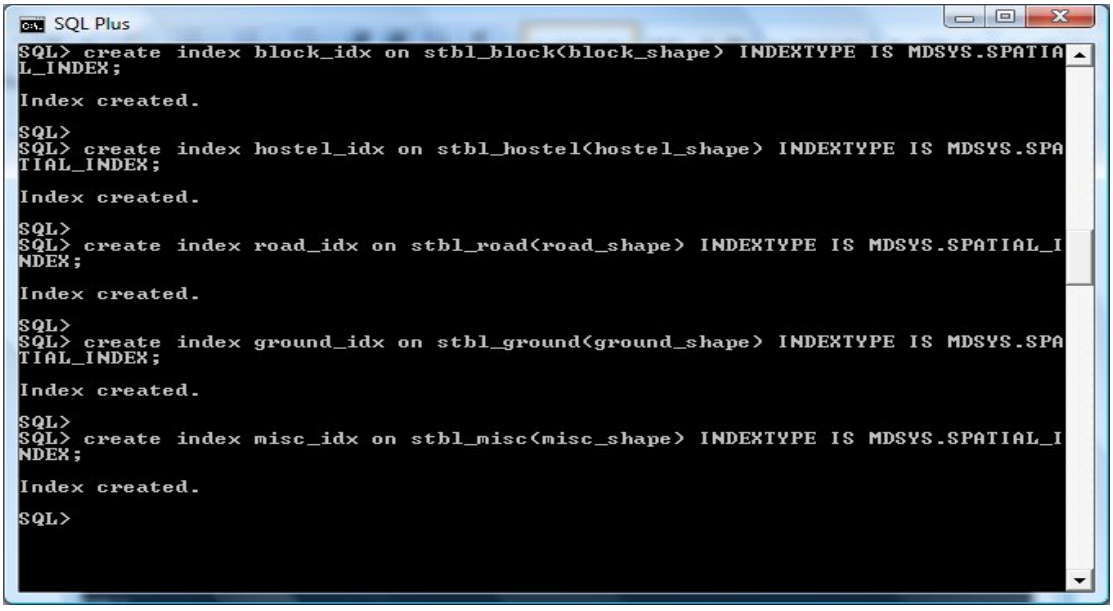

Figure 4.6 Index Creation Queries for Spatial Tables

4.2.3 Insertion into Spatial Tables

We have made entries into all the tables by inserting the coordinates of each desired location of the Thapar University. For this purpose we have referred the graph of figure 4.2. Insertion operation on each table is shown below in figure 4.7, figure 4.8, figure 4.9, figure 4.10, and figure 4.11.

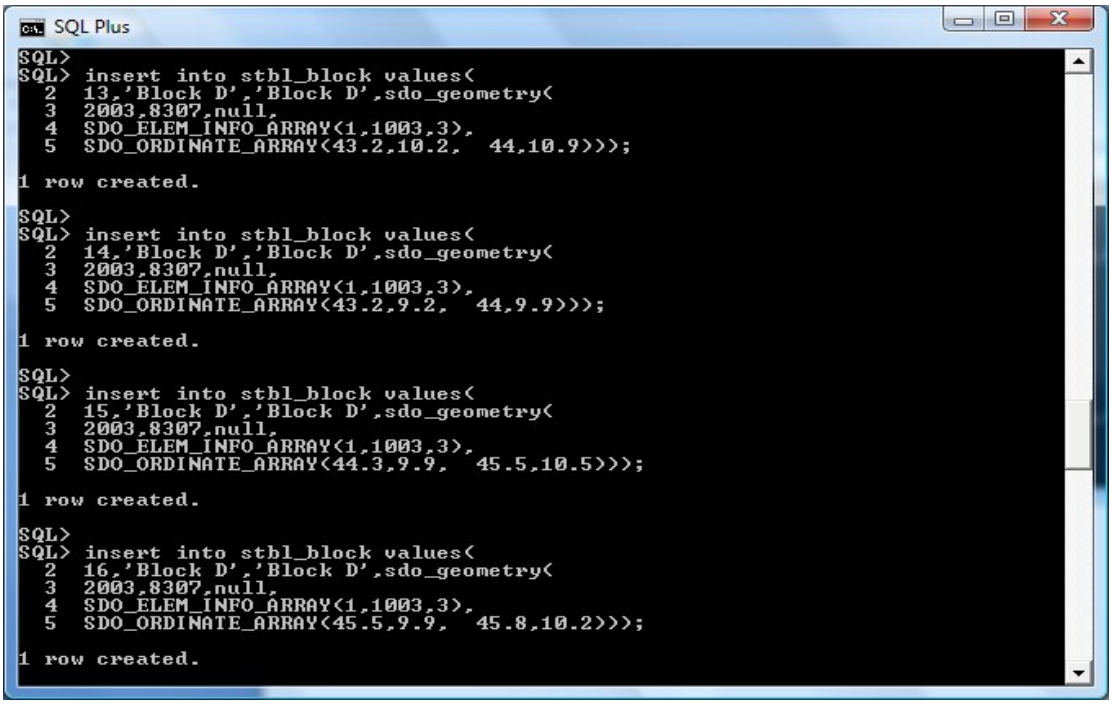

Figure 4.7: Insert Queries for Table STBL_BLOCK

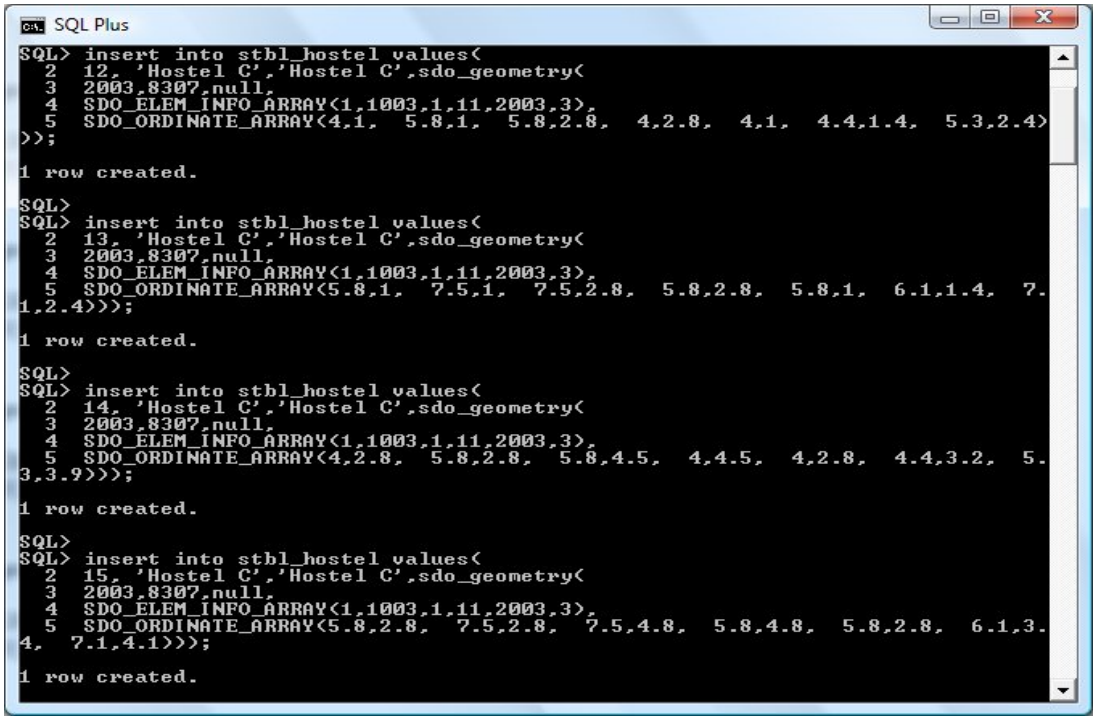

Figure 4.8: Insert Queries for Table STBL_HOSTEL

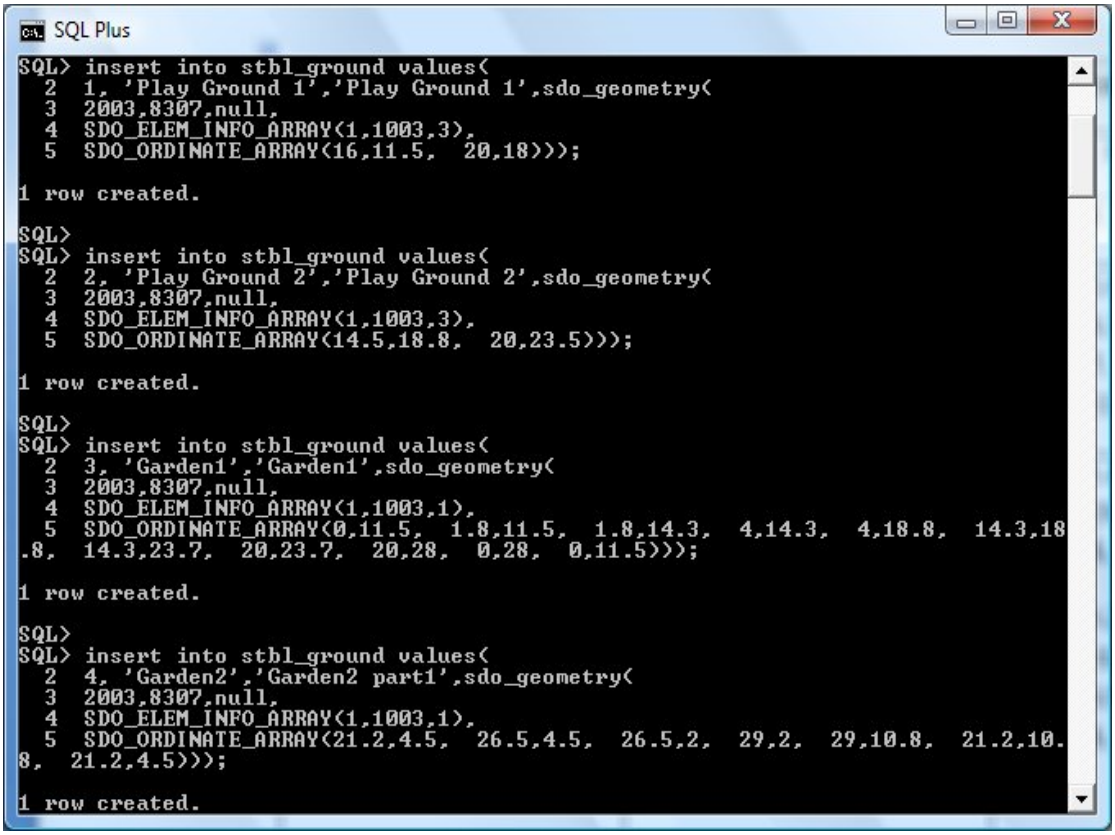

Figure 4.9: Insert Queries for Table STBL_GROUND

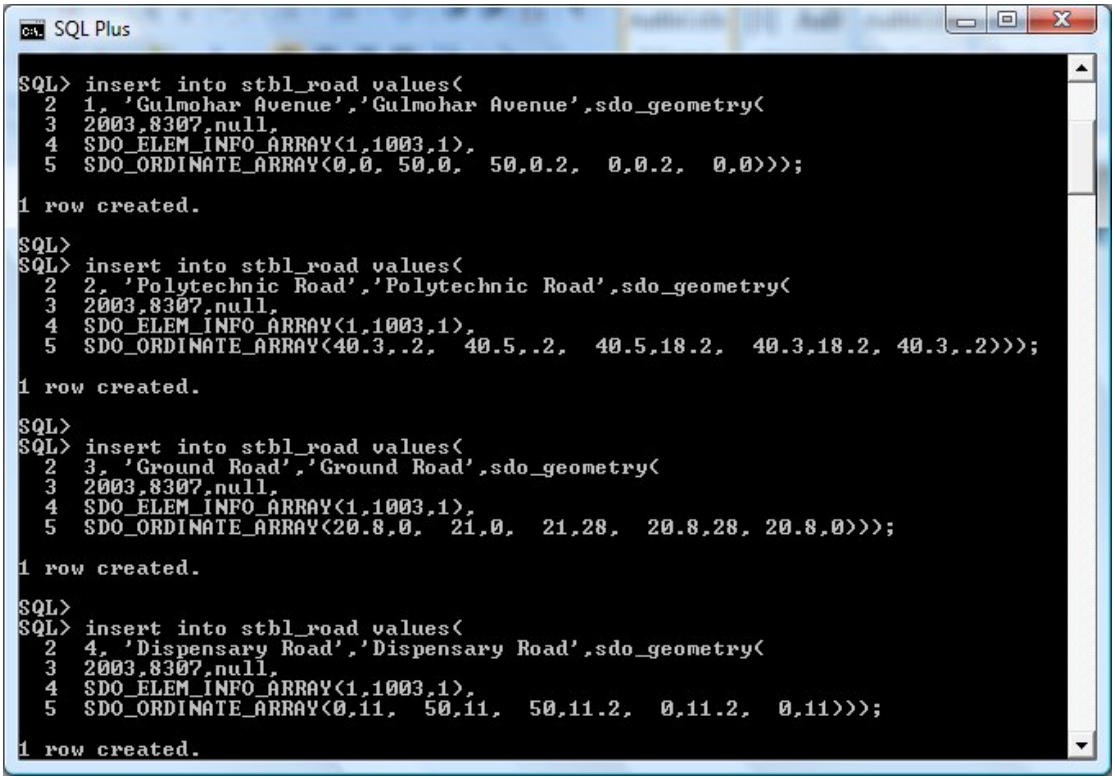

Figure 4.10: Insert Queries for Table STBL_ROAD

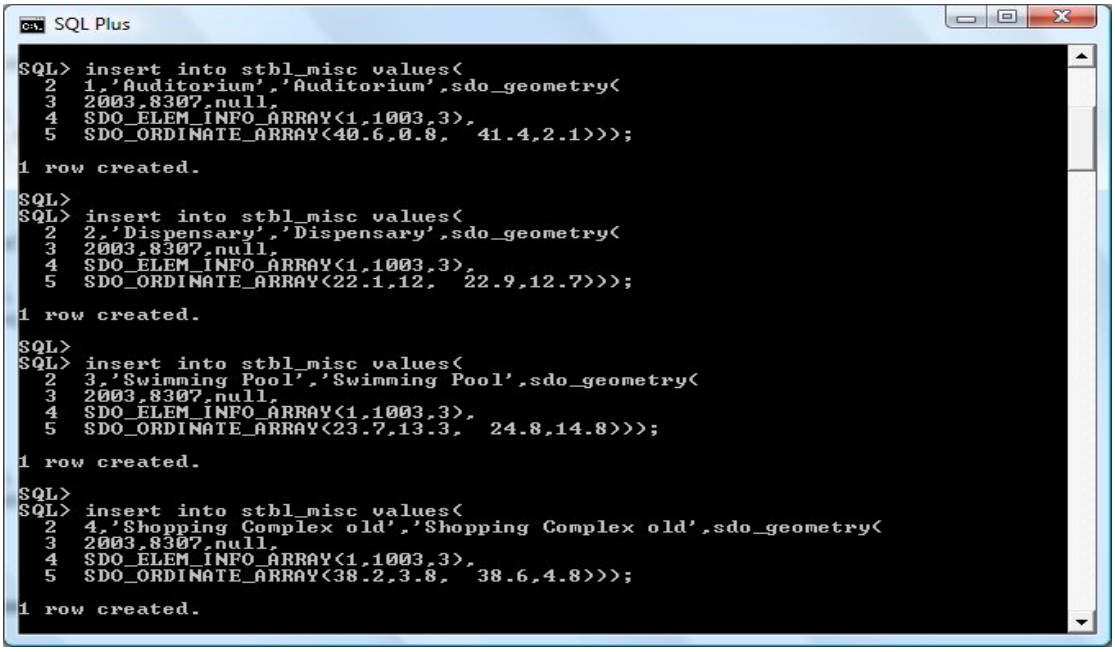

Figure 4.11: Insert Queries for Table STBL_MISC

4.3 Working with Map Builder Tool: Defining Mapping Metadata

Using Map Builder tool, we have defined styles, themes and base map for our application. Basically, this mapping metadata is used by MapViewer to generate map at run time. Also, we can preview the map for any table in Map Builder. We started the tool by establishing connection with Thapar university database by selecting new connection option from the File menu. Figure 4.12, depicts all the database connection parameters.

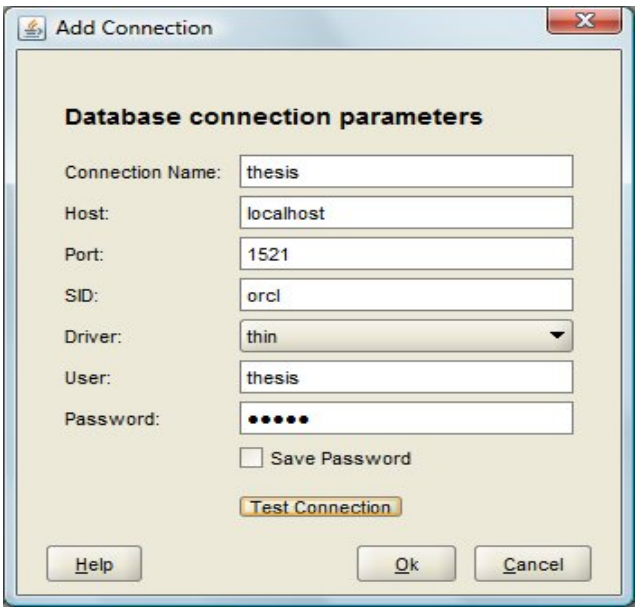

Figure 4.12: Creating Database Creation using Add Connection Option

The successful connection to database opens the window as given in figure 4.13, showing all tables that we have created during database creation.

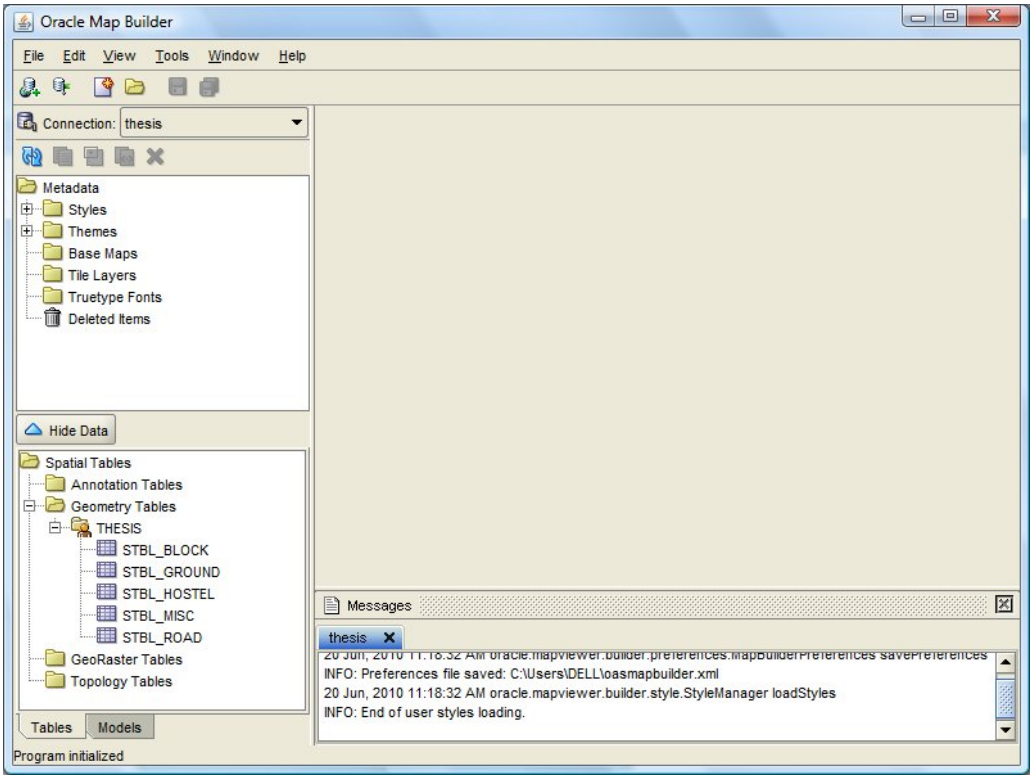

Figure 4.13: Map Builder Window with Thapar University Database

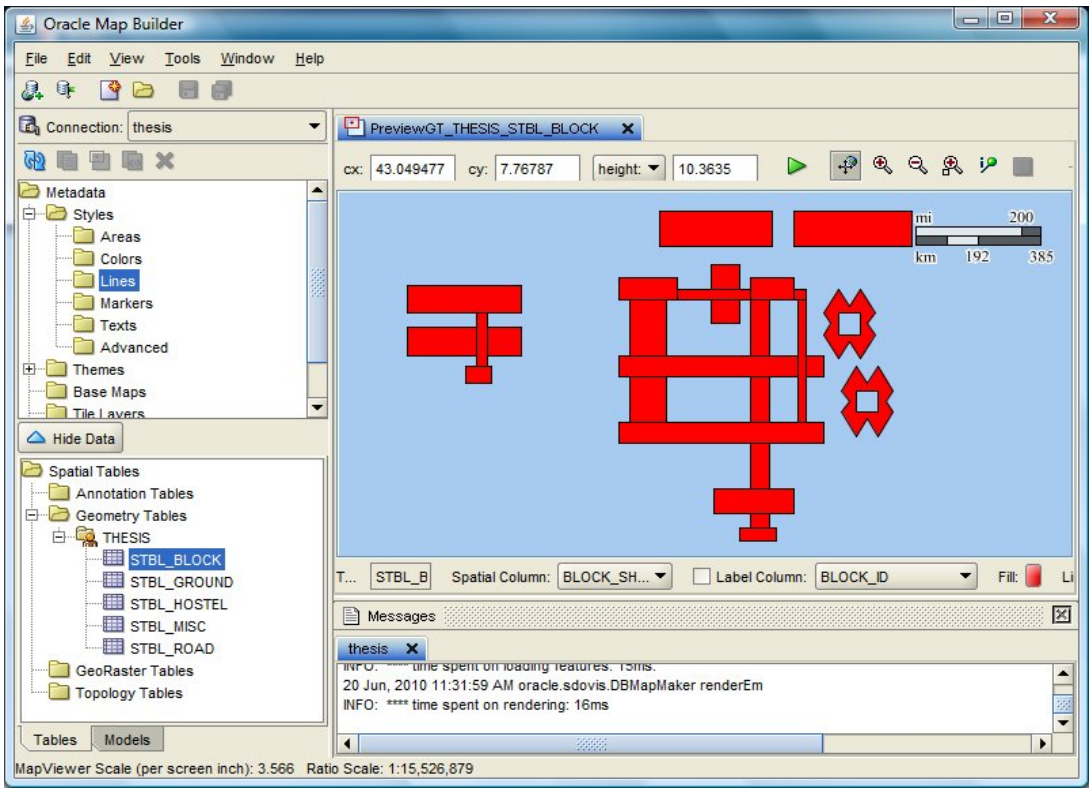

Figure 4.14: Preview of STBL_BLOCK Table with Default Styles of Map Builder

4.3.1 Style Creation

Figure 4.14 shows the preview of STBL_BLOCK table in form of map. It means all the coordinates stored in geometry column of the table are mapped into graphical shapes. Like this, we can preview maps of all other tables (or themes because all the tables represent a theme with reference to Thapar University Map) with the default styles of Map Builder. We can create our styles also. There are six types of styles that we can create and use in our themes. These styles include areas, colors, lines, markers, texts and advanced styles. We have used colors, texts and markers styles for generating themes from spatial tables. For example, we have defined color style for ground theme as shown in figure 4.15. For creating color style we have selected Create Color Style option by right clicking on Color option of metadata navigator tree.

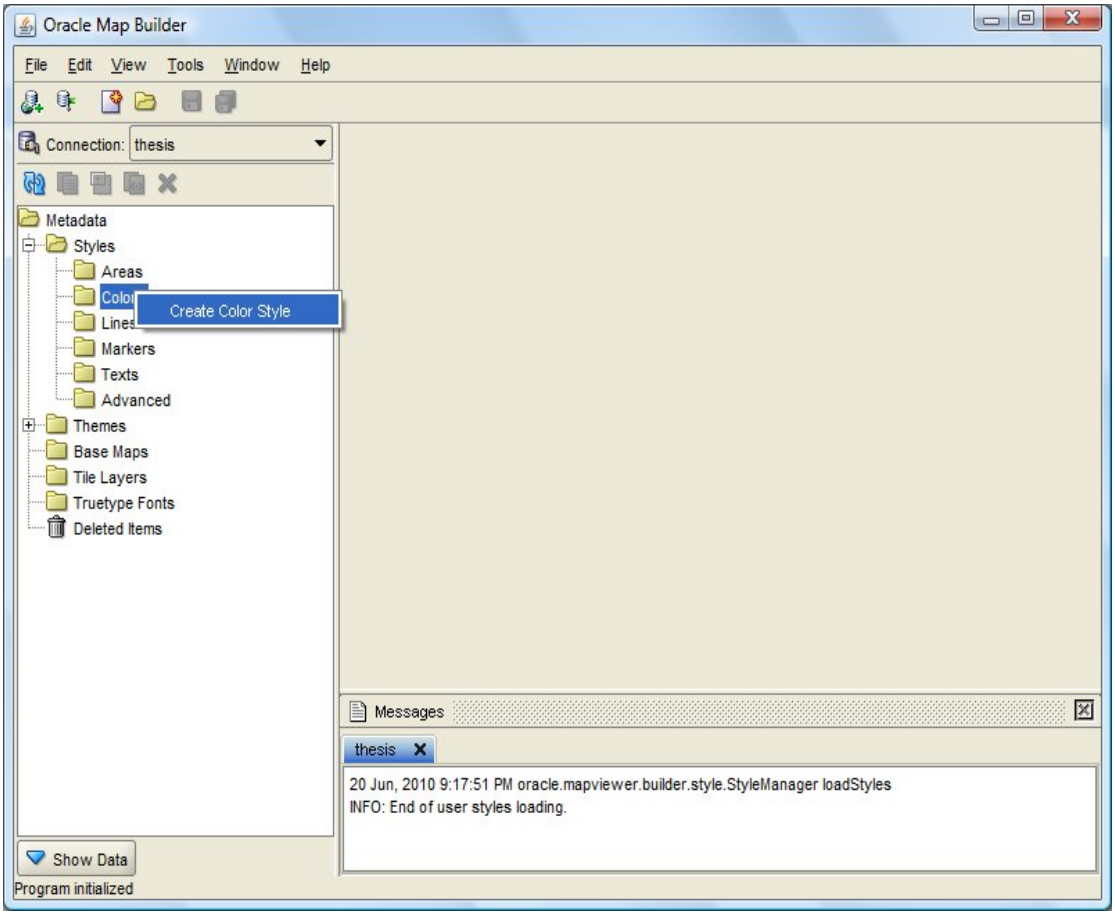

Figure 4.15: Selection of Create Color Style Option

After this, we can specify the color parameters like color name, description, and style options like Fill and Stroke and then save this color definition as shown in figure 4.16.

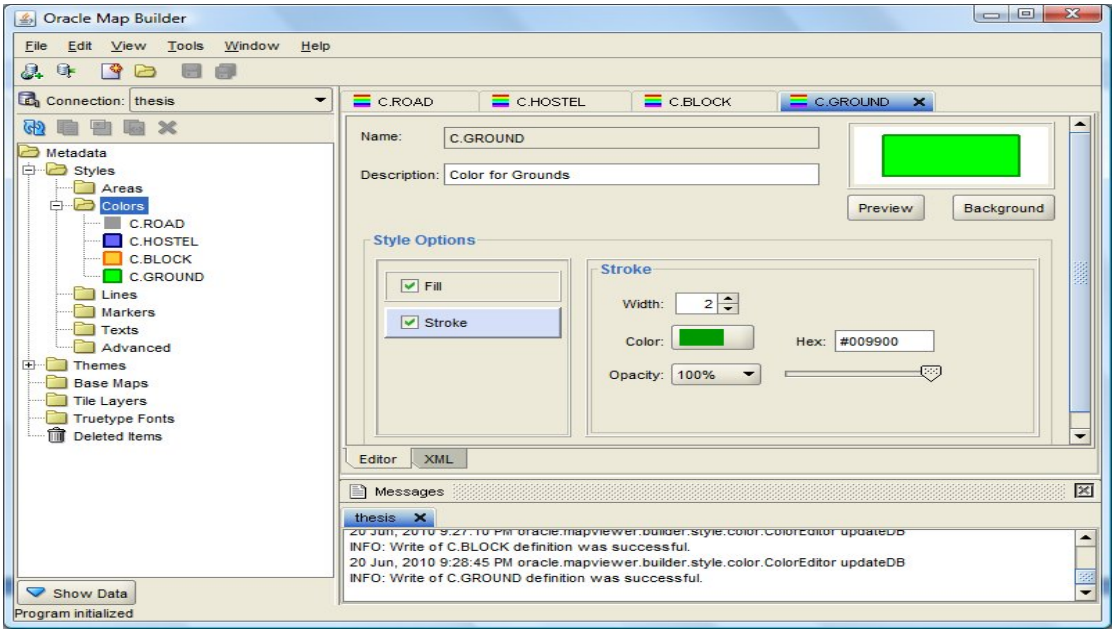

Figure 4.16: Definition of Color Style for Ground Theme

Similarly, we have defined different color styles for other tables also. In addition, we have defined one common text style for all the tables and marker styles for road theme and ground theme. In text style, we define the font, font-size, font-style, font-color and all other features for the text that is going to be a part of the map like the name of ground. In marker style, we can use any appropriate image to mark the location. For example, the car image is used for marking the roads in Thapar University. The definition of this marker style is given below in figure 4.17.

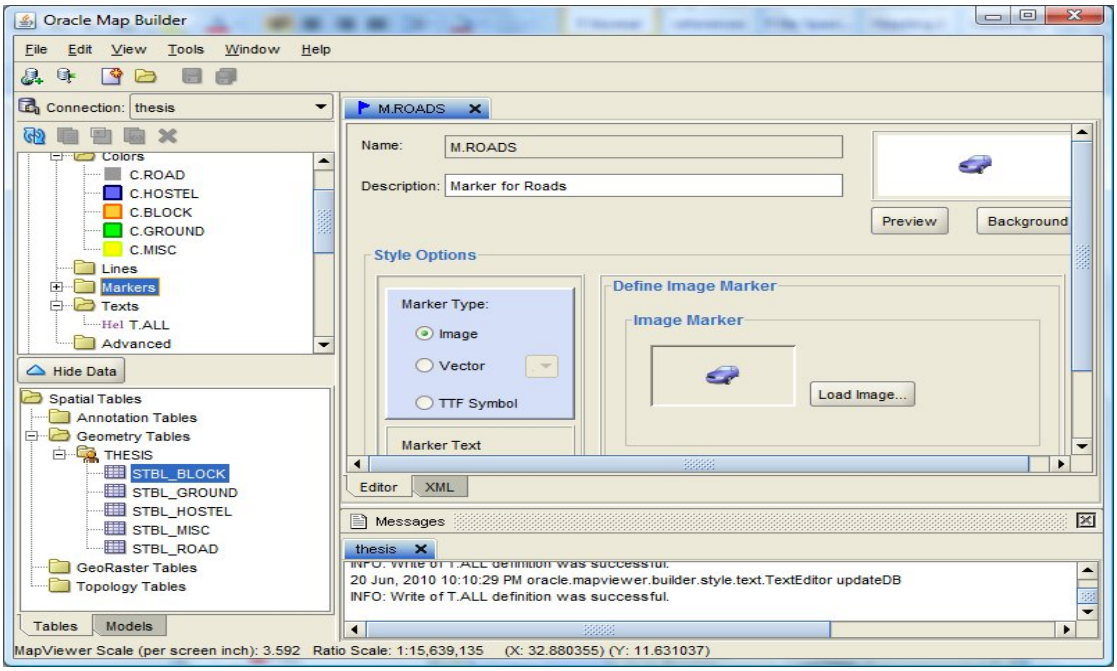

Figure 4.17: Definition of Marker Style for road

4.3.2 Theme Creation

By combining different styles, a theme is created. We have given below the procedure of creating geometry theme for table STBL_BLOCK using figures. Themes for remaining tables can be created by following the same procedure.

In order to create a block theme we have selected the required table from spatial table navigator tree and right clicked on it. Then selected the Create Geometry Theme option as shown in figure 4.18, and further, followed the steps described in figure 4.19, figure 4.20, figure 4.21, figure 4.22, and figure 4.23 respectively. Figure 4.19 shows the details of theme parameters. Figure 4.20 shows the color picking for the theme. . Figure 4.21 shows the feature style selection. Figure 4.22 shows the details of style parameters. At last, figure 4.23 shows summary box for geometry theme.

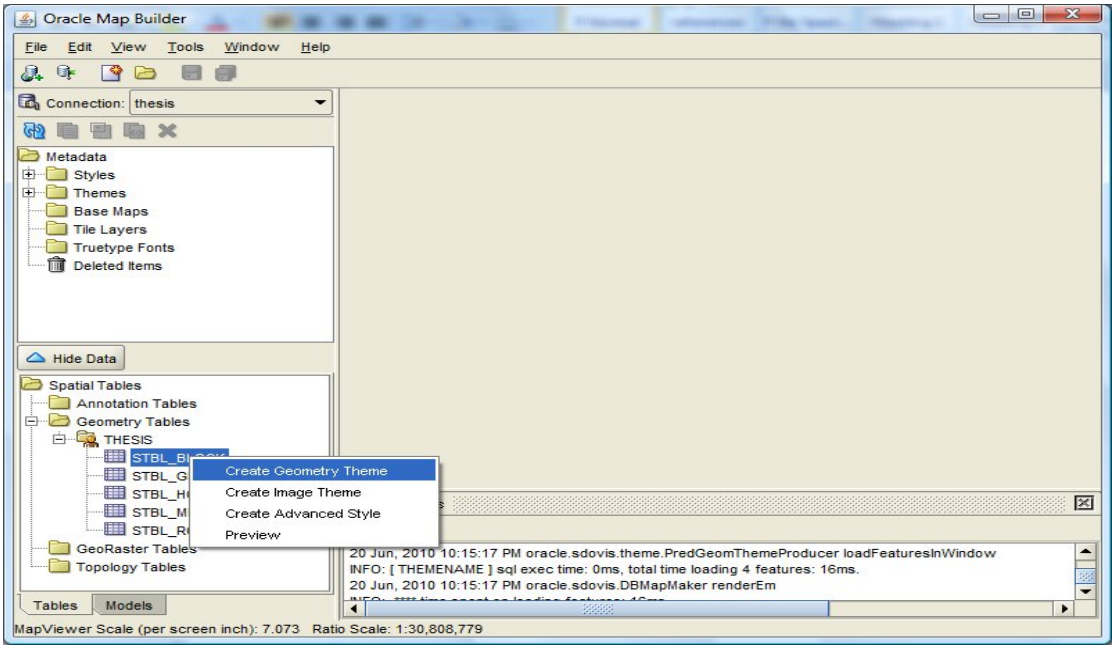

Figure 4.18: Selection of Create Geometry Option

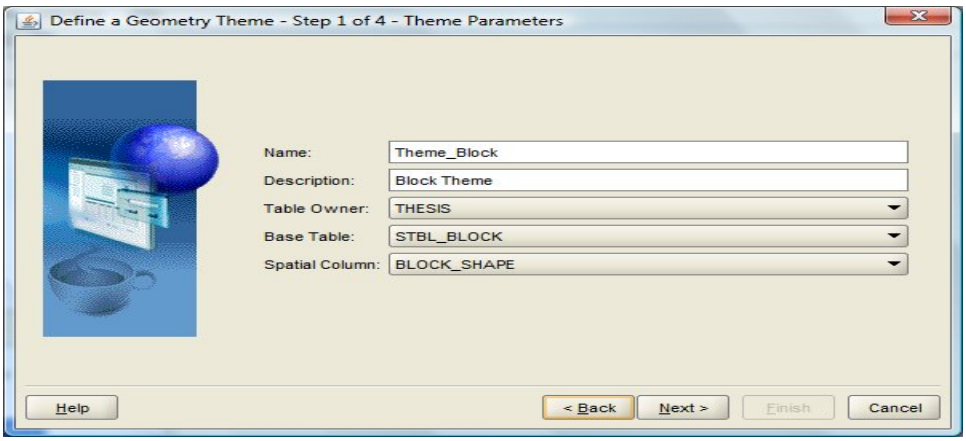

Figure 4.19: Specifying Theme Parameters

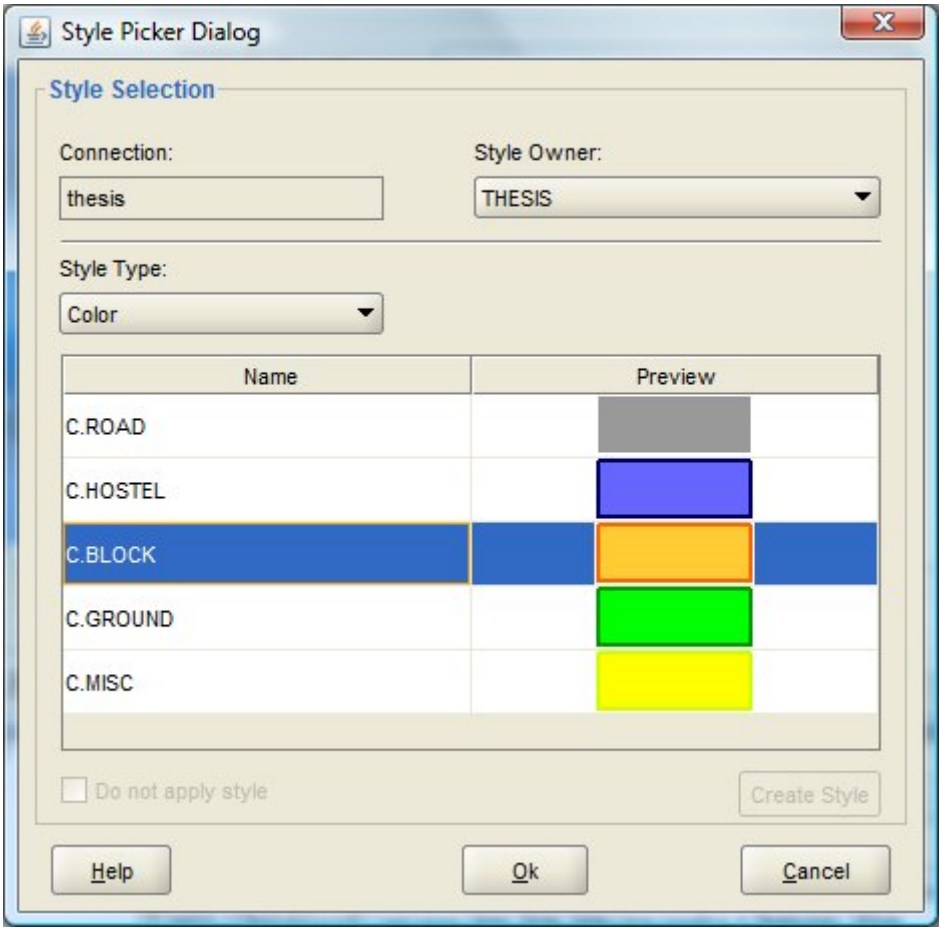

Figure 4.20: Style Picker Dialog Window

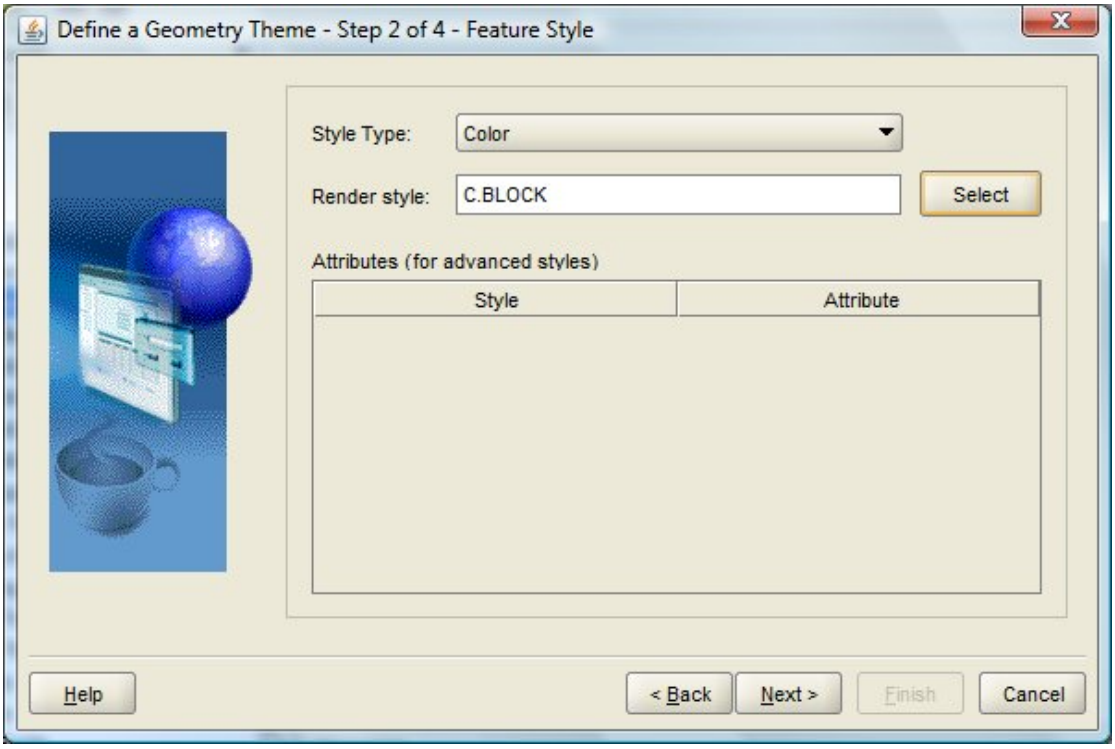

Figure 4.21: Specifying Feature Style

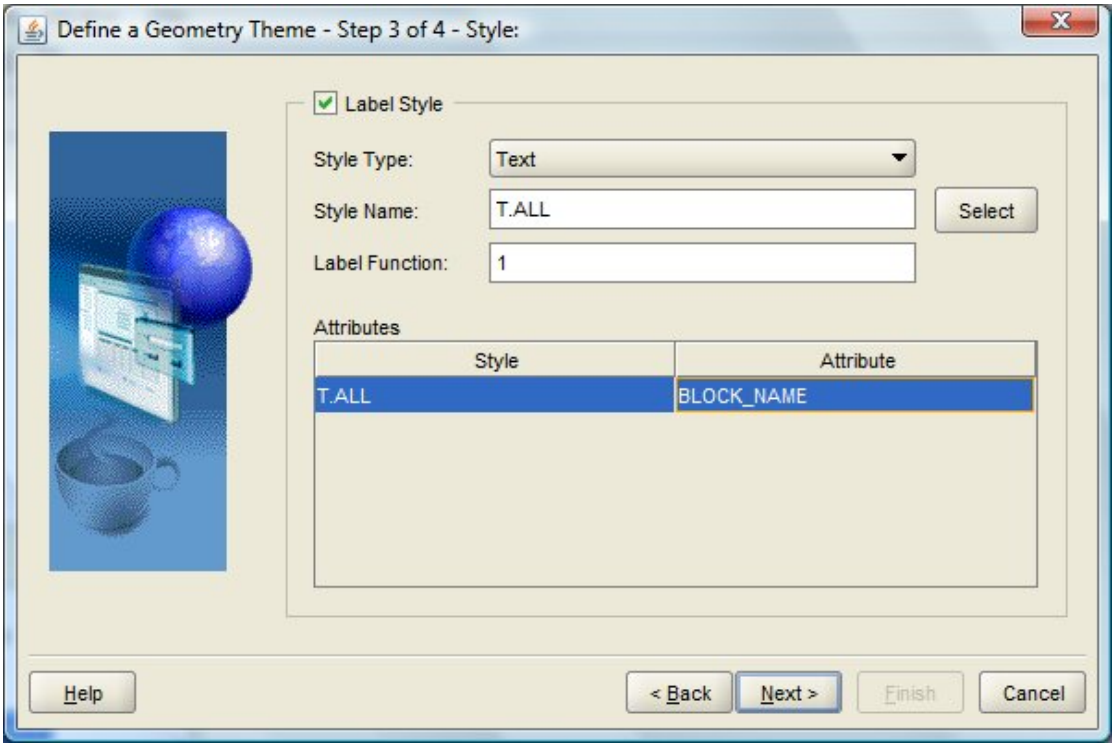

Figure 4.22: Specifying Style Parameters

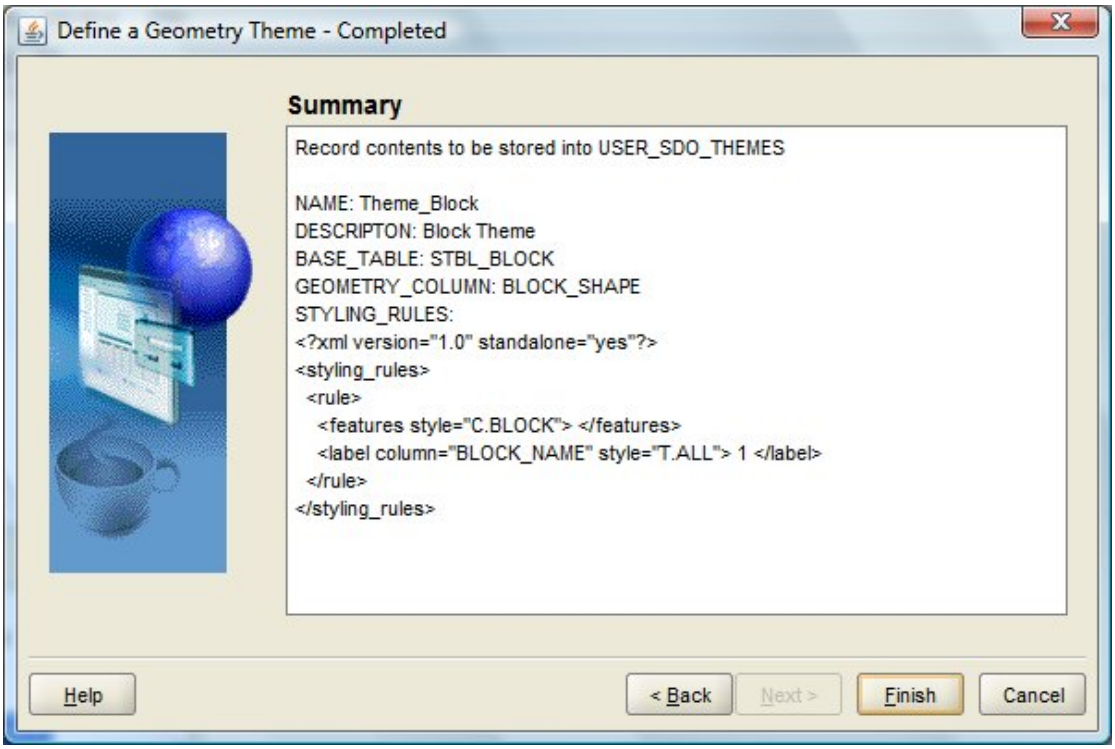

Figure 4.23: Summary Box for Geometry Theme

After the procedure has been finished, we can again preview the themes with defined styles. Figure 4.24 shows block theme with color and text styles, whereas figure 4.25, shows ground theme with color and marker styles.

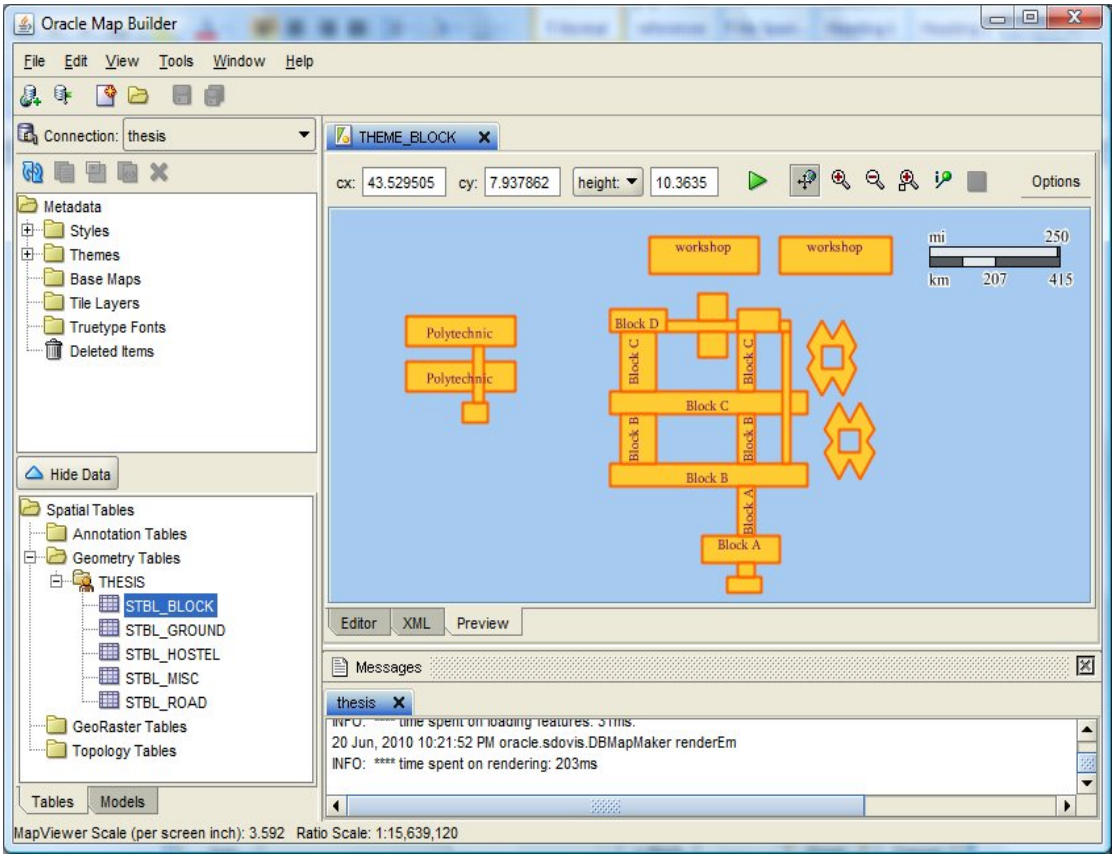

Figure 4.24: Block Theme Preview in Map Builder

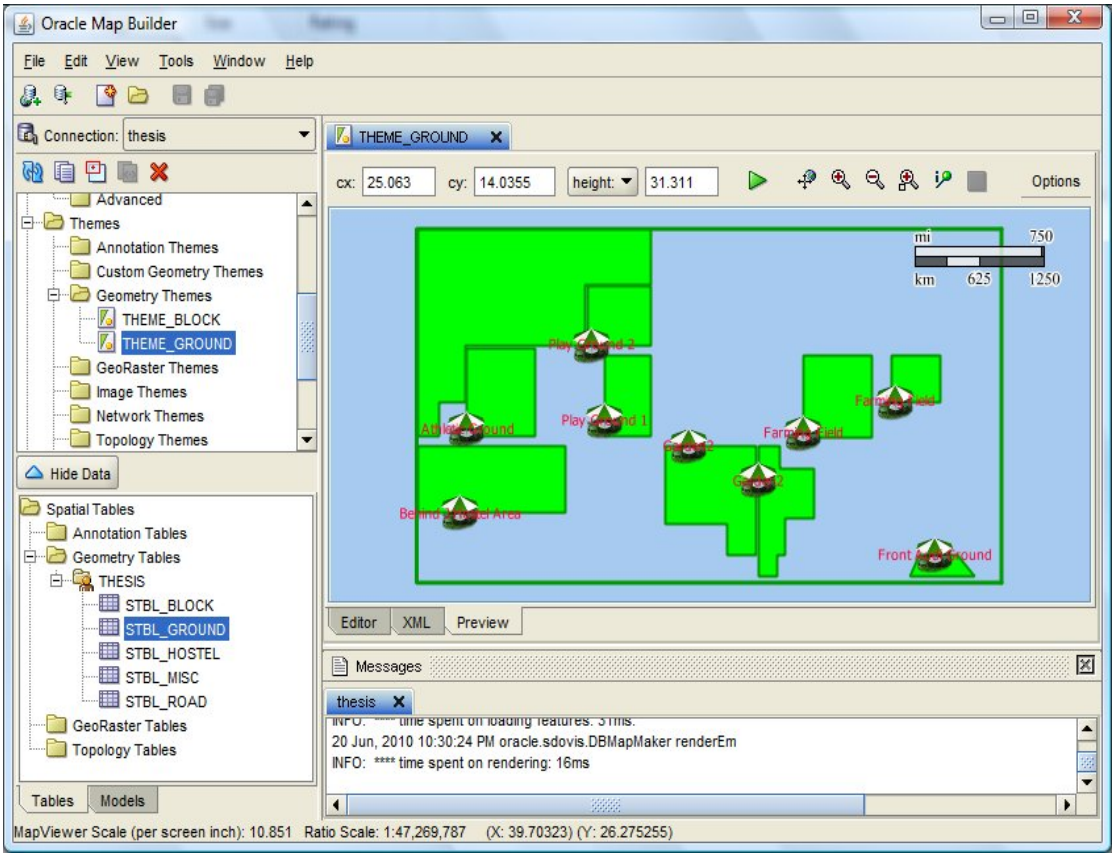

Figure 4.25: Ground Theme Preview in Map Builder

4.3.3 Base Map Generation

A base map is generated by putting one or more themes at one place. We have combined all the five themes namely road theme, ground theme, hostel theme, block theme and miscellaneous theme for making base map. This is the basic map that represents Thapar University in fully fledged form. In first step, we selected Base Map option given in metadata navigator tree and then right clicked on it. Further selected Create Base Map option and then followed the steps given in figure 4.26, figure 4.27, and figure 4.28 respectively. Figure 4.26 shows the details of base map parameters. Figure 4.27 shows the selection of base map themes. Figure 4.28 shows summary box for base map. Figure 4.29 shows the base map of Thapar University.

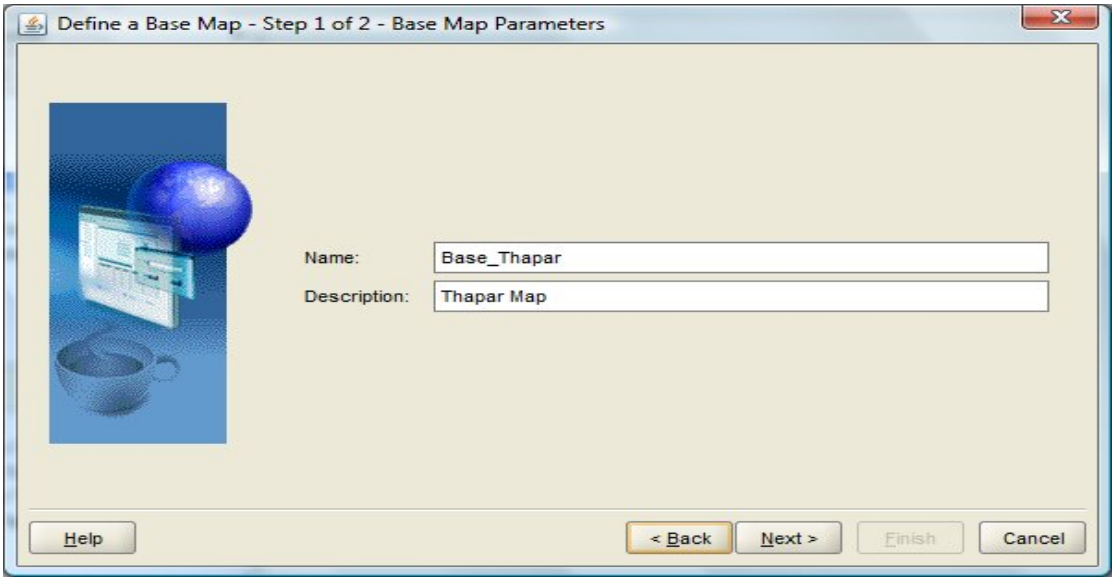

Figure 4.26: Specifying Base Map Parameters

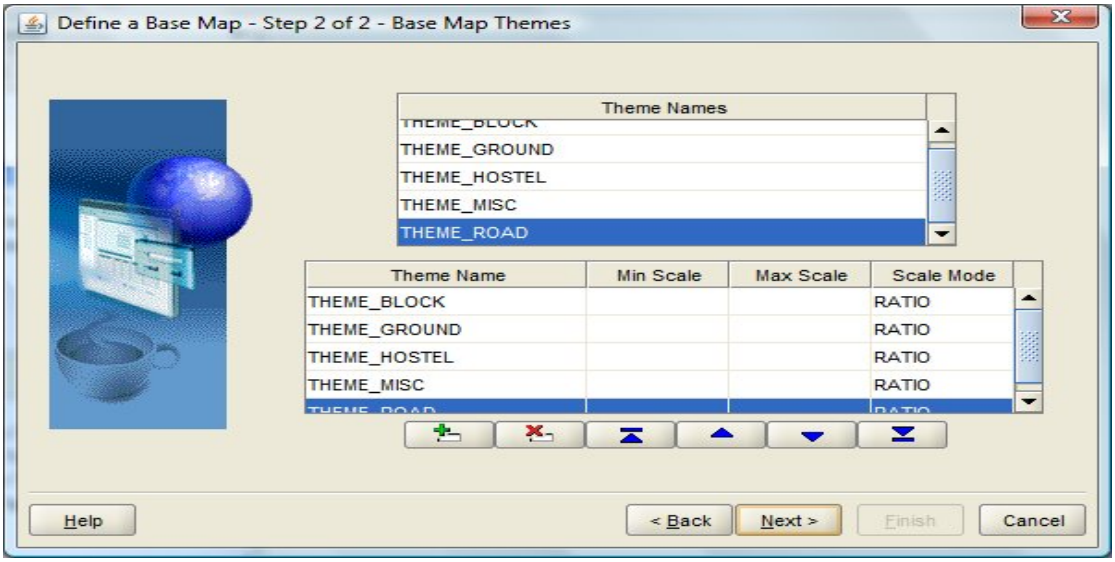

Figure 4.27: Selection of Themes Base Map

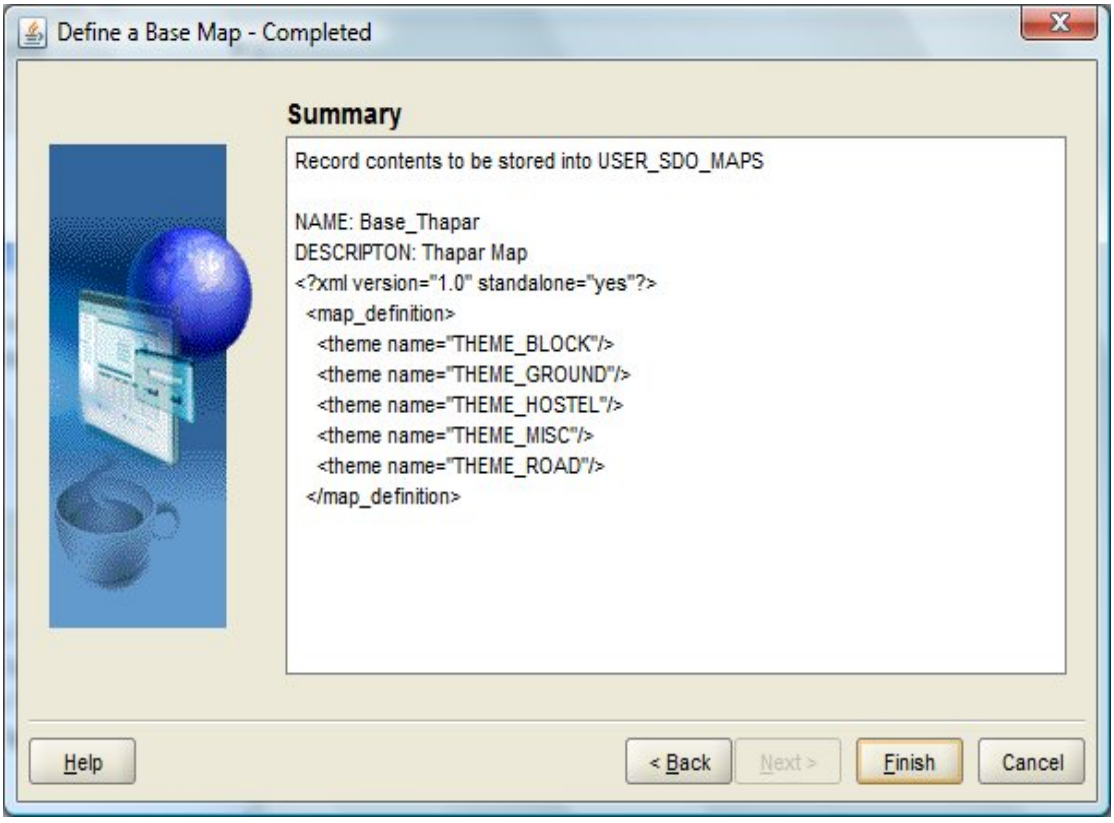

Figure 4.28: Summary Box for Base Map

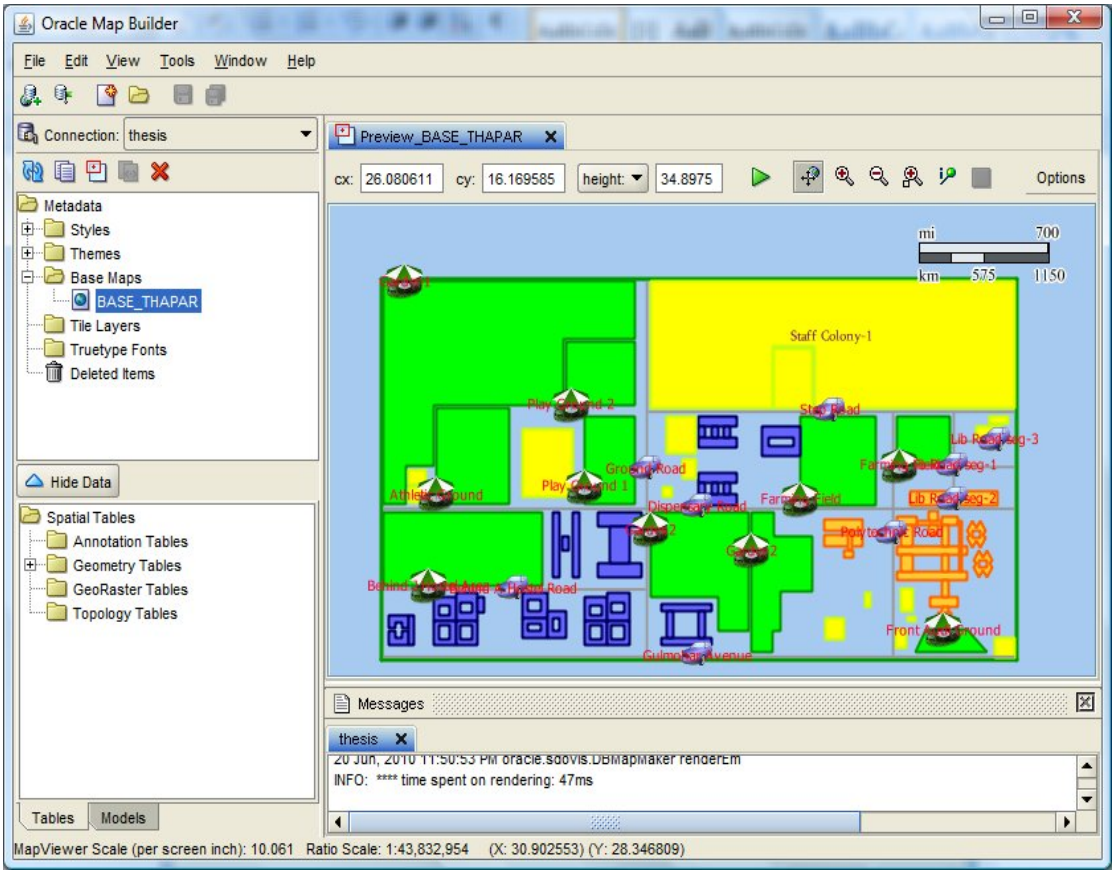

Figure 4.29: Base Map of Thapar University

4.4 Working with MapViewer: Data Source and Tile Layer Creation

Oracle MapViewer works as a map server. Whenever any application requires any activity related to map like map-generation, and map-processing *etc*., it is all done by the MapViewer. Below is given the initialization screen of MapViewer in figure 4.30.

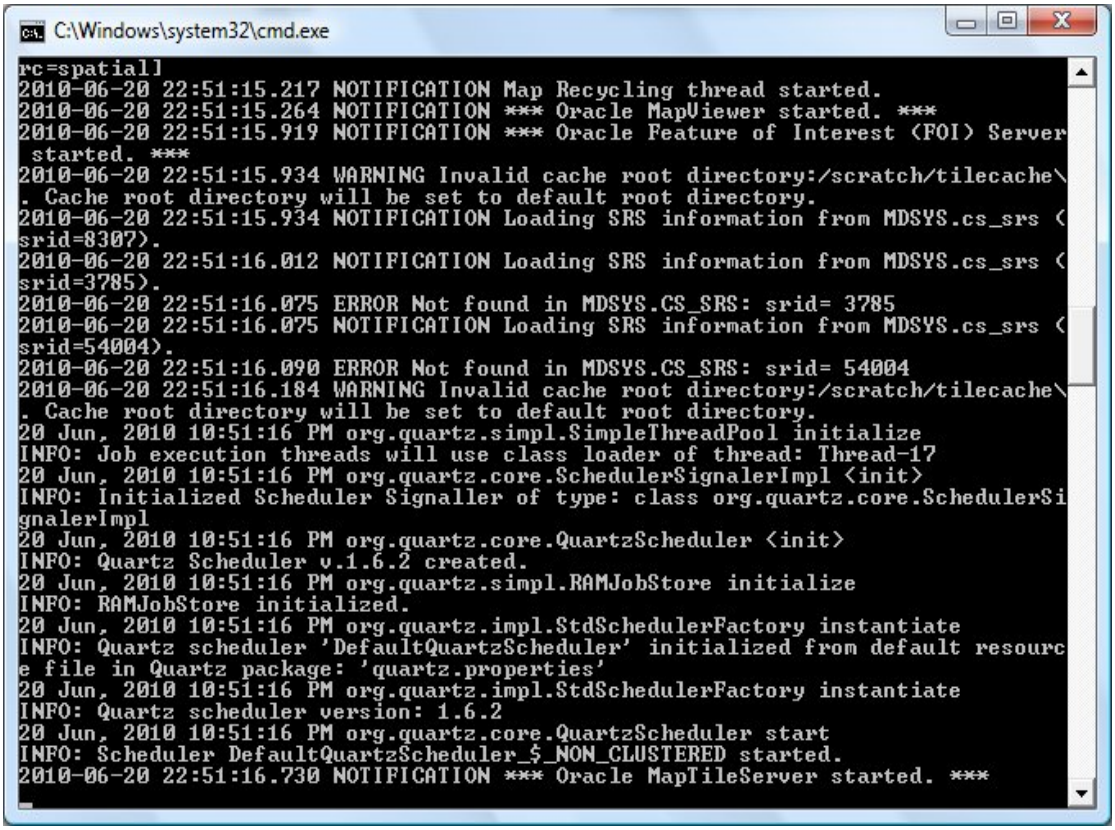

Figure 4.30: MapViewer Initialization Screen

After the initialization, MapViewer starts itself as a web application. The Home page of MapViewer is given below in figure 4.31.

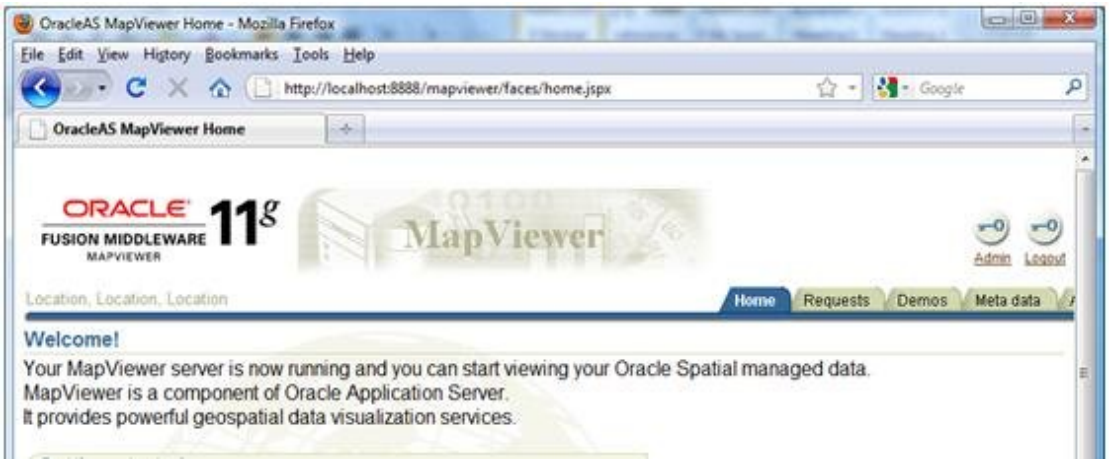

Figure 4.31: MapViewer Home Page

By clicking on the Admin link, login page is opened for the user authorization. The login page is shown in figure 4.32.

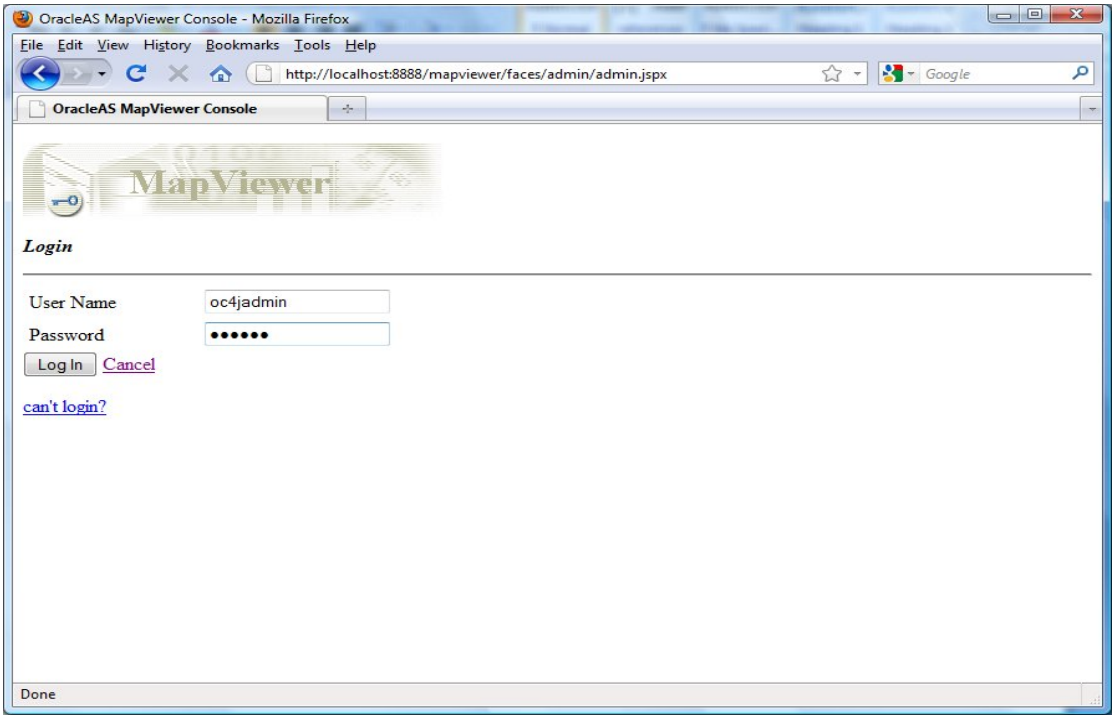

Figure 4.32: Login Screen of MapViewer

We have created a data source for Thapar University spatial database. In figure 4.33, there is given the screen of creation of data source.

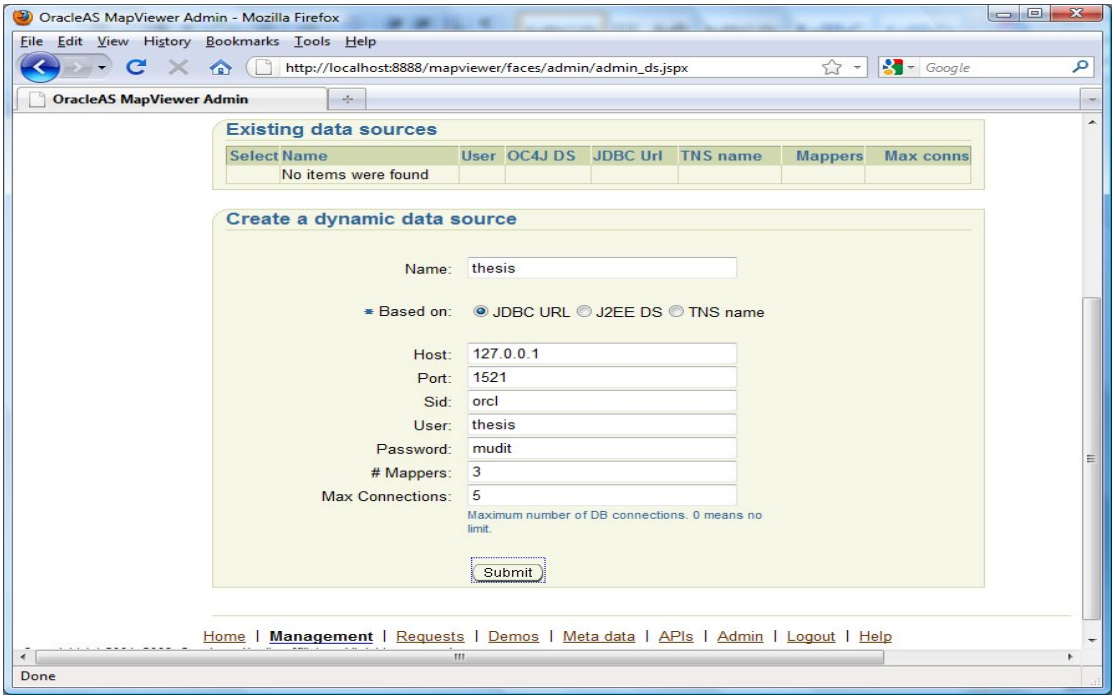

Figure 4.33: Creation of Data Source

The newly created data source is added in the existing data sources panel as shown in figure 4.34, given below.

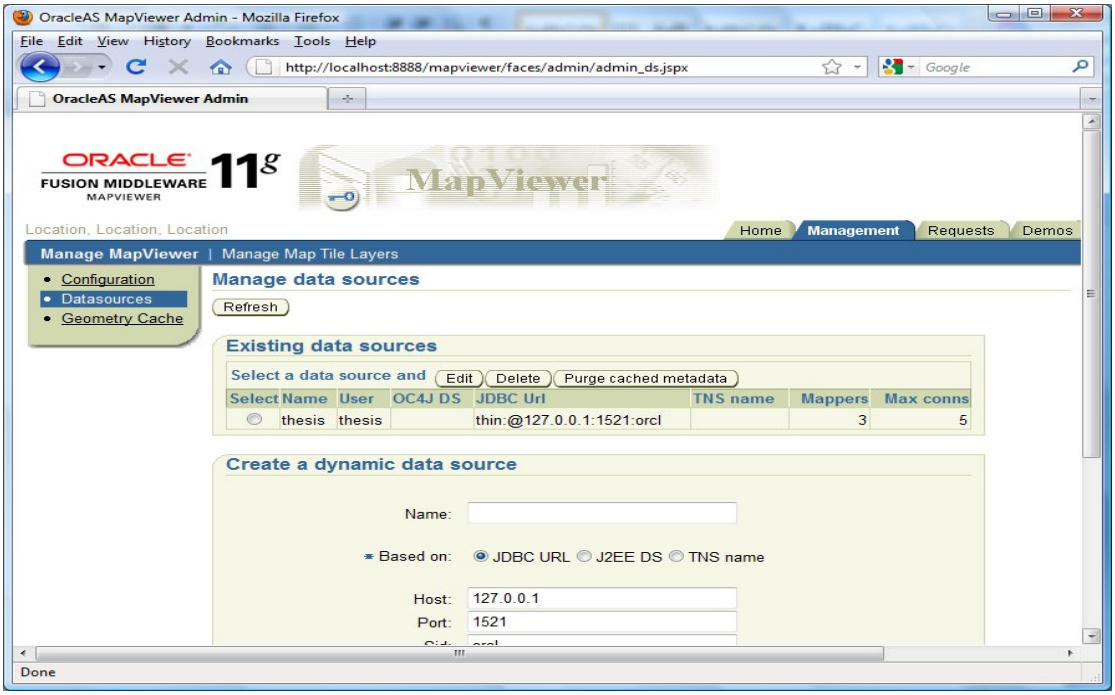

Figure 4.34: Existing Data Sources in MapViewer

Further, we have created map tile layer for our spatial application. It is very important task in order to run any application successfully. Figure 4.35, shows the creation of map tile layer.

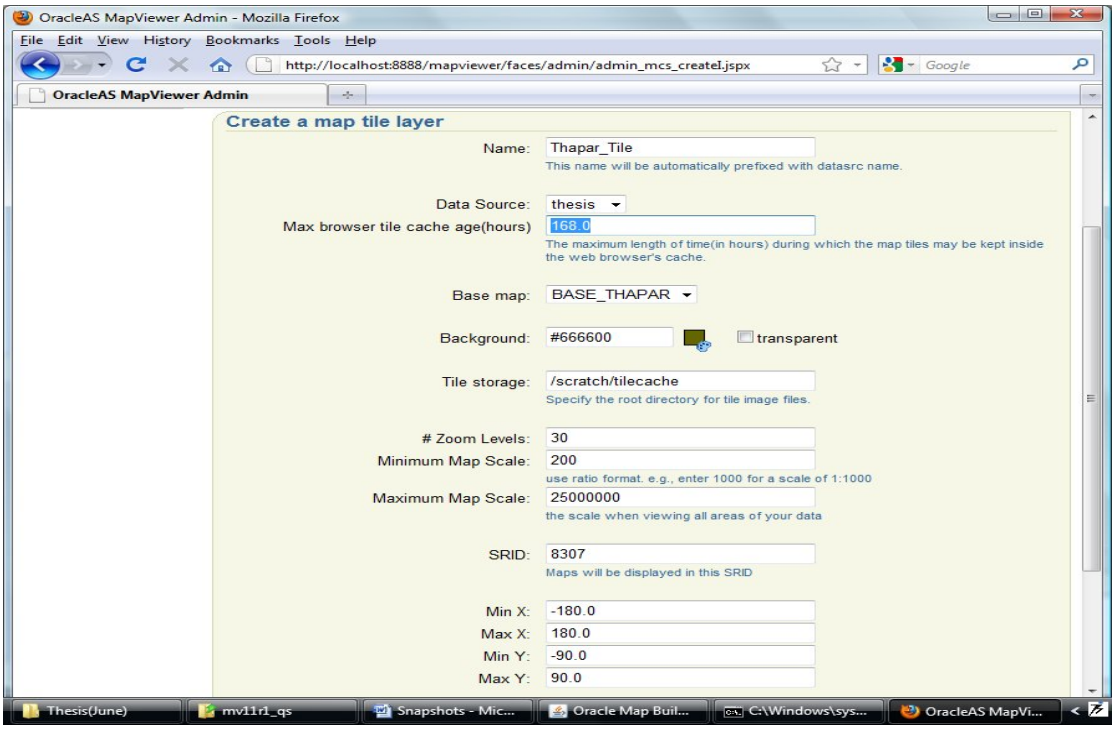

Figure 4.35: Creation of Map Tile Layer

The map tile layer that created previously is added in the existing map tile layer panel as shown in figure 4.36.

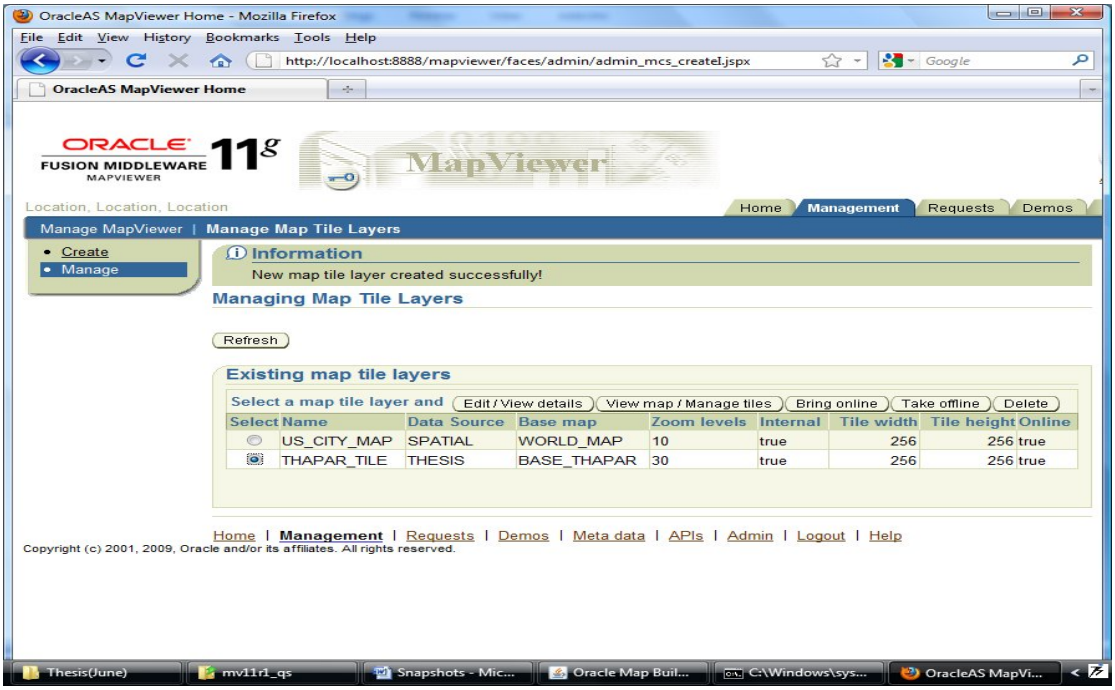

Figure 4.36: Existing Map Tile Layers in MapViewer

By selecting Thapar_Tile map tile layer and then clicking on View map/Manage Tile link, we get the map of the Thapar University as shown inn figure 4.37.

Figure 4.37: Thapar University Map in MapViewer

4.5 Designing, Developing, Deploying, and Running the Application

We have designed and developed the spatial application. The application is developed in HTML and JavaScript code with the use of MapViewer's JavaScript mapping library. We have used Mozilla Firebox web browser (in full screen mode) for running our Thapar University spatial application. Our designed application has the following provisions:

1. To show the map with the use of legend items with different styles.
2. To view any target area of Thapar University with zoom and distance measure tool options.
3. To navigate the map with navigation panel and overview map options

Figure 4.38 shows the Thapar University spatial application in running mode.

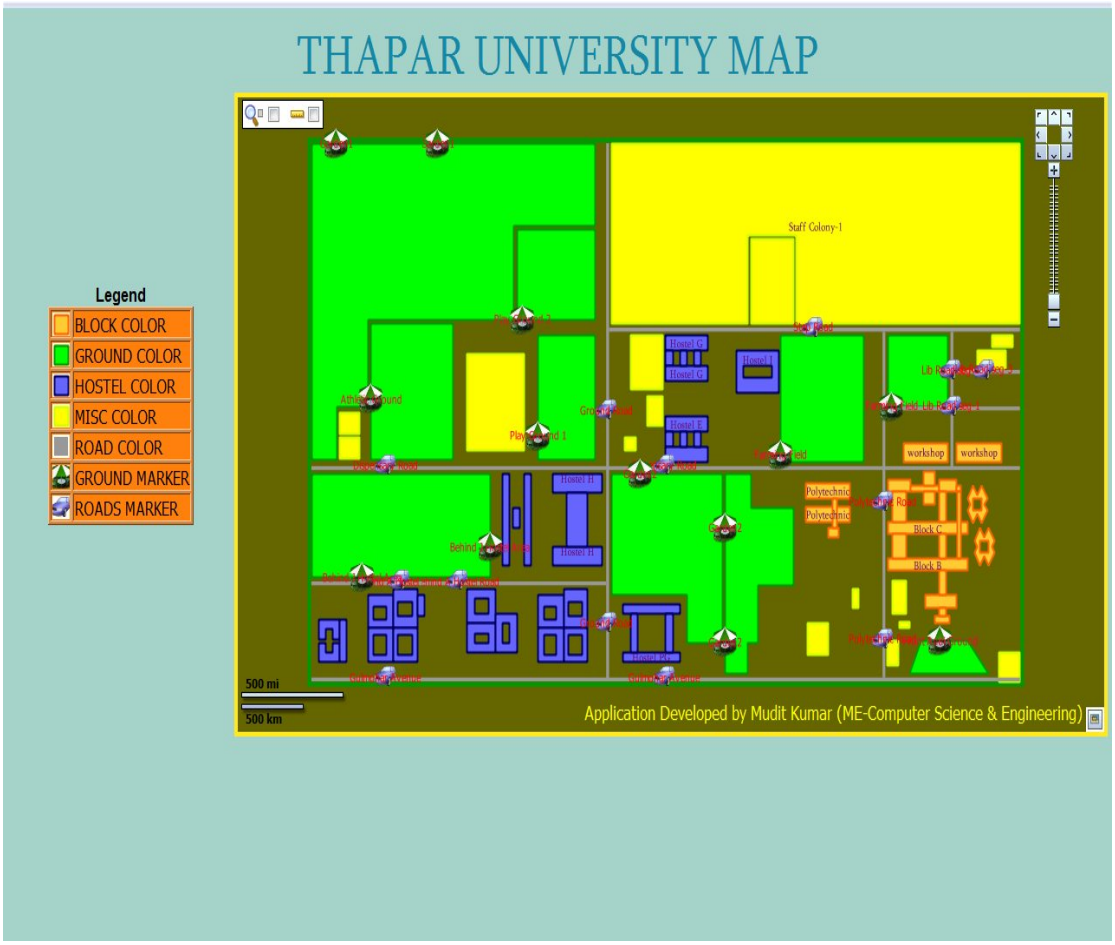

Figure 4.38: Thapar University Spatial Application in Running Mode

Figure 4.39, figure 4.40, figure 4.41, and figure 4.42 depict the map running with different options. Figure 4.39 shows the map with Overview Map at bottom-right

corner. Figure 4.40 shows the map with change zoom level option. Figure 4.41 shows the map at zoom level 3 and figure 4.42 shows the map at zoom level 5.

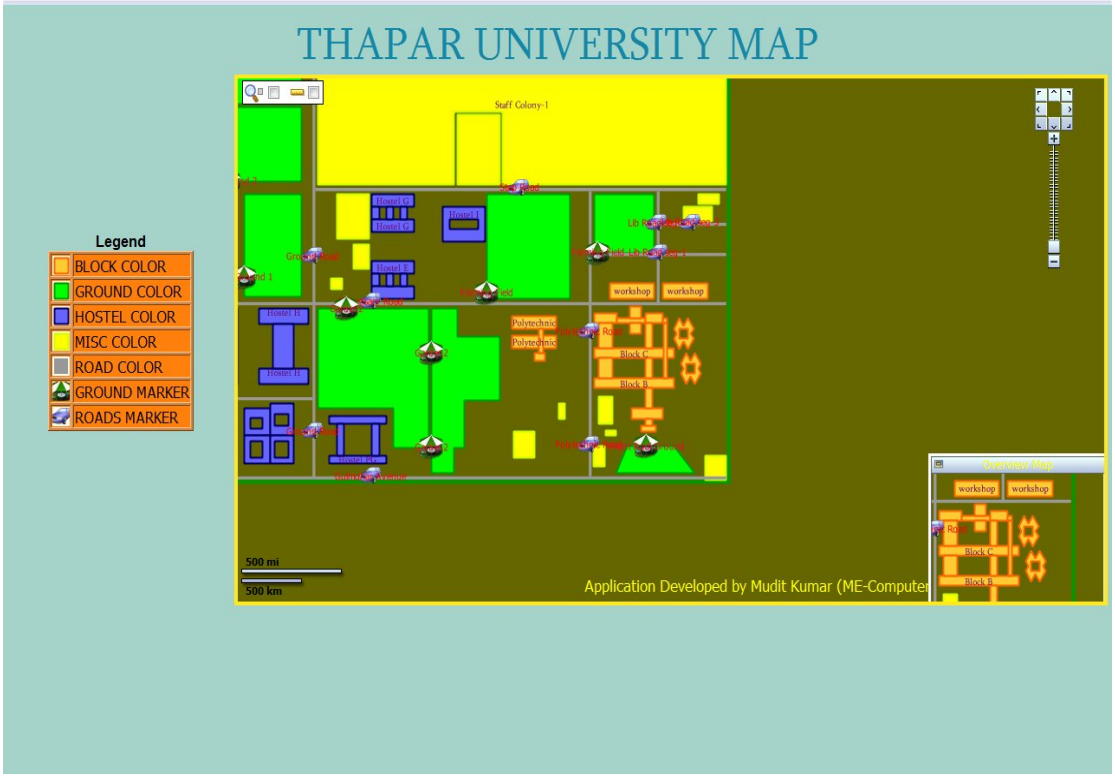

Figure 4.39: Map with Overview Map at Bottom-Right Corner

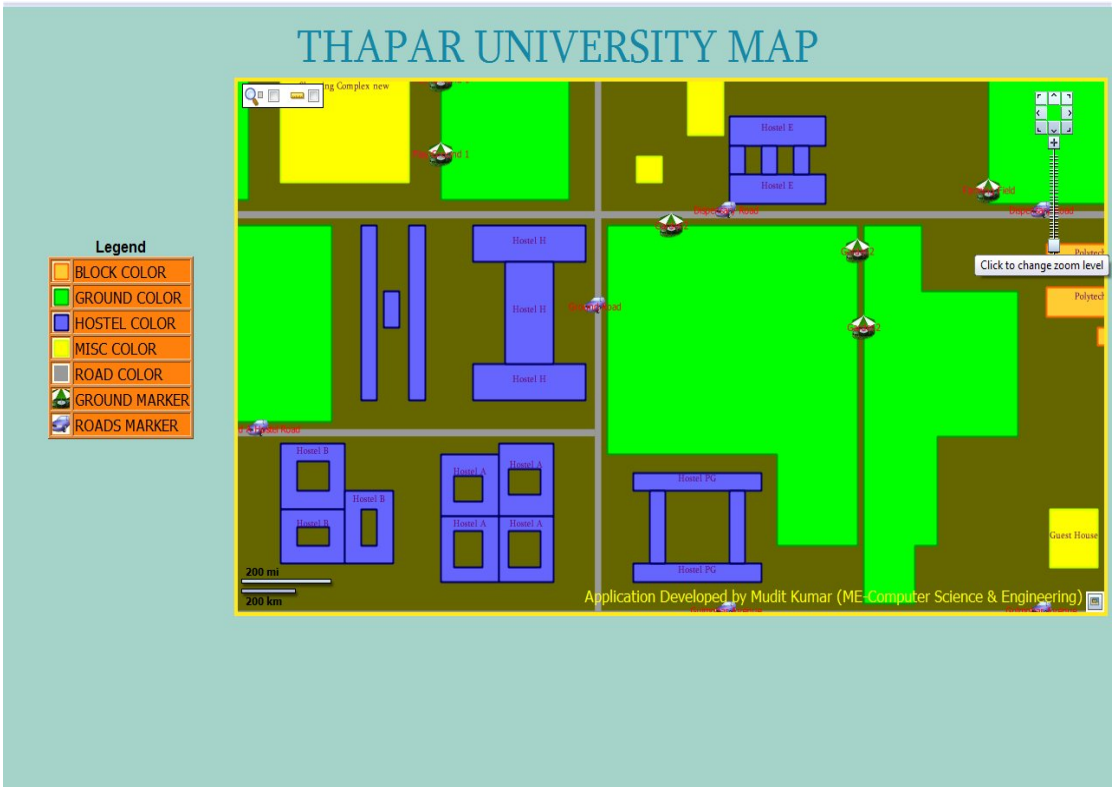

Figure 4.40: Map Showing Option to Change Zoom Level

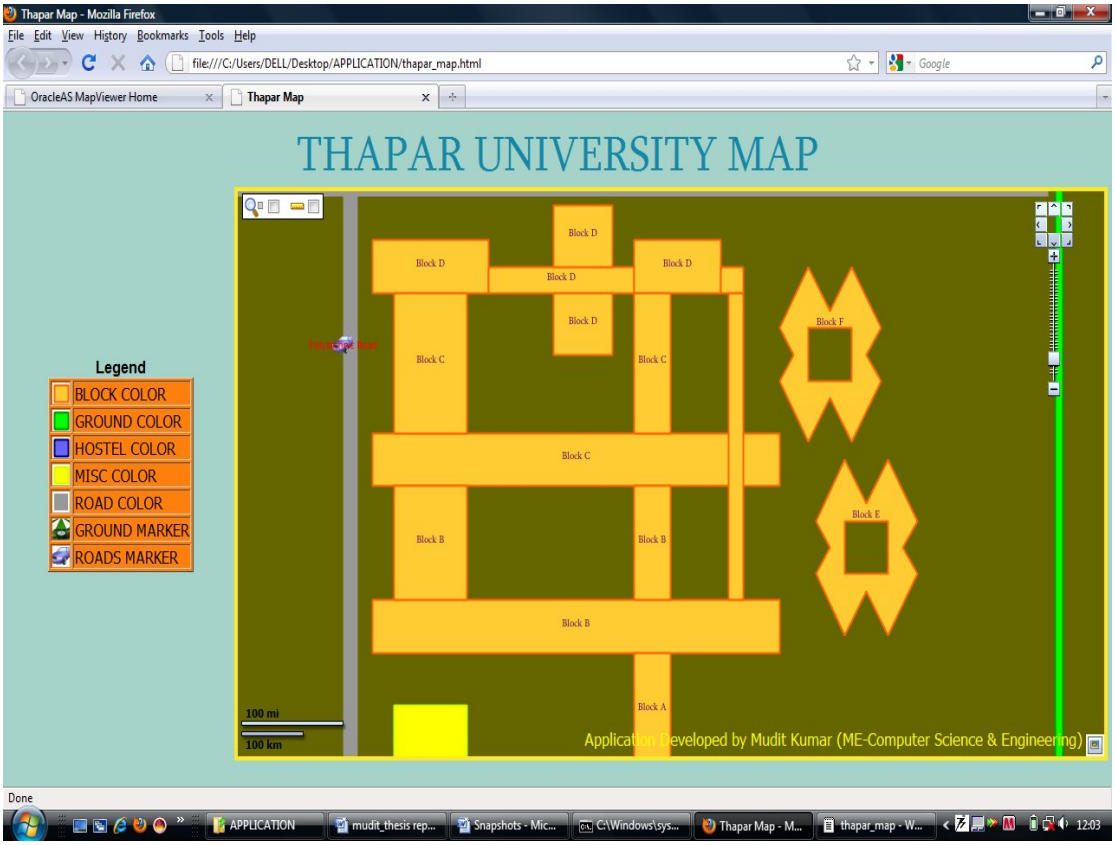

Figure 4.41: Map at Zoom Level 3

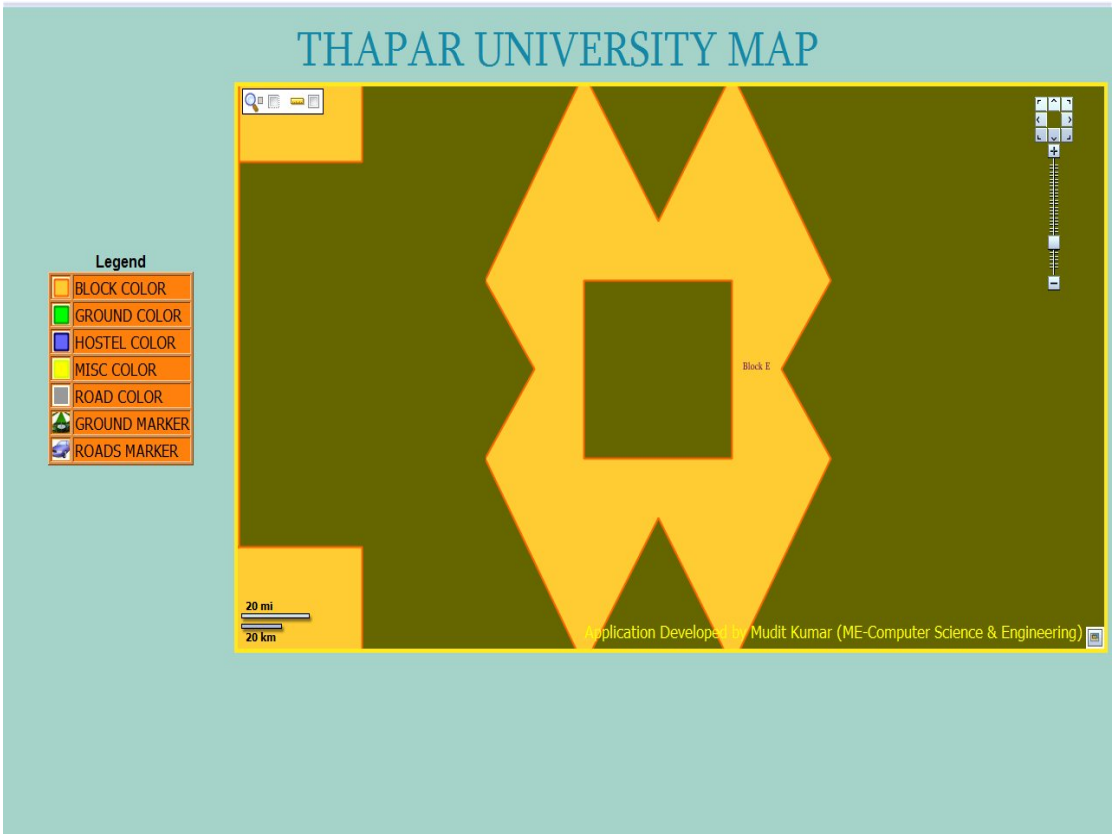

Figure 4.42: Map at Zoom Level 5

Figure 4.43 shows the map in which an area is covered by the rectangle of marquee zoom option.

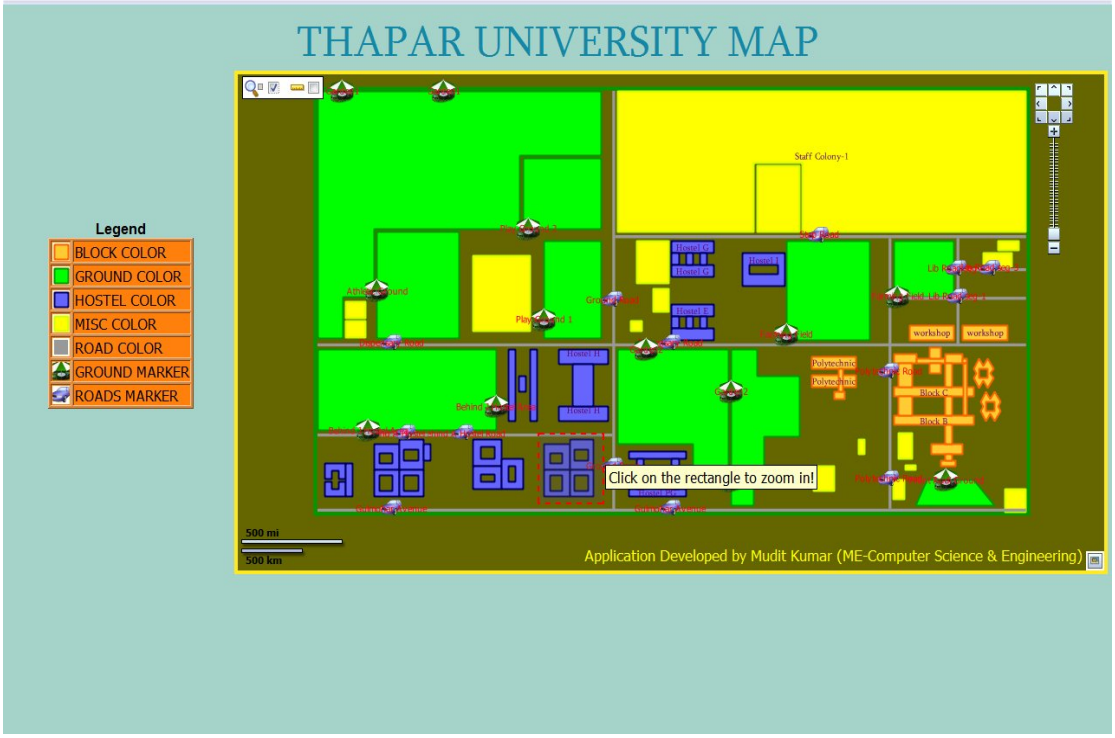

Figure 4.43: Using Marquee Zoom Option

After selecting an area, we can click on the rectangle to maximize that area. Figure 5.44 shows the area at some higher zoom level.

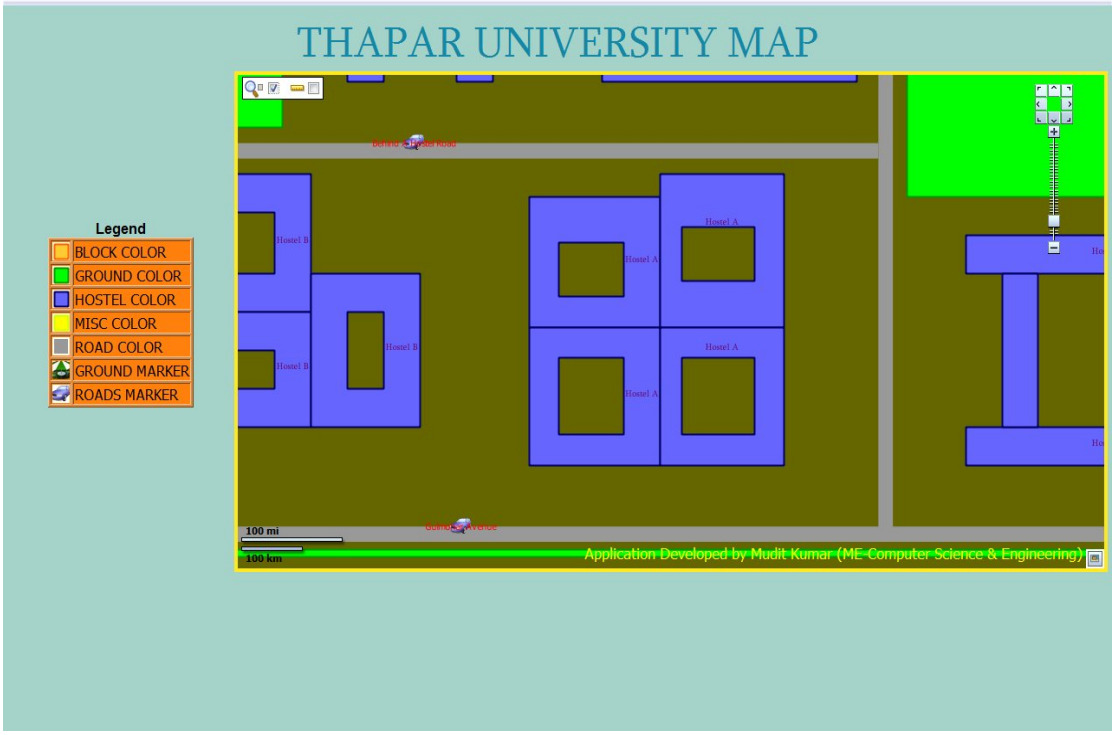

Figure 4.44: Area Zoomed with Marquee Zoom Option

Another important tool that can be used in our application is the distance tool located at the top-left corner of the map beside marquee zoom tool. In figure 5.45, we have selected two points marked by two small red colored circles and this tool is showing the distance (in meters) between these two points in the text box located at the top of the map.

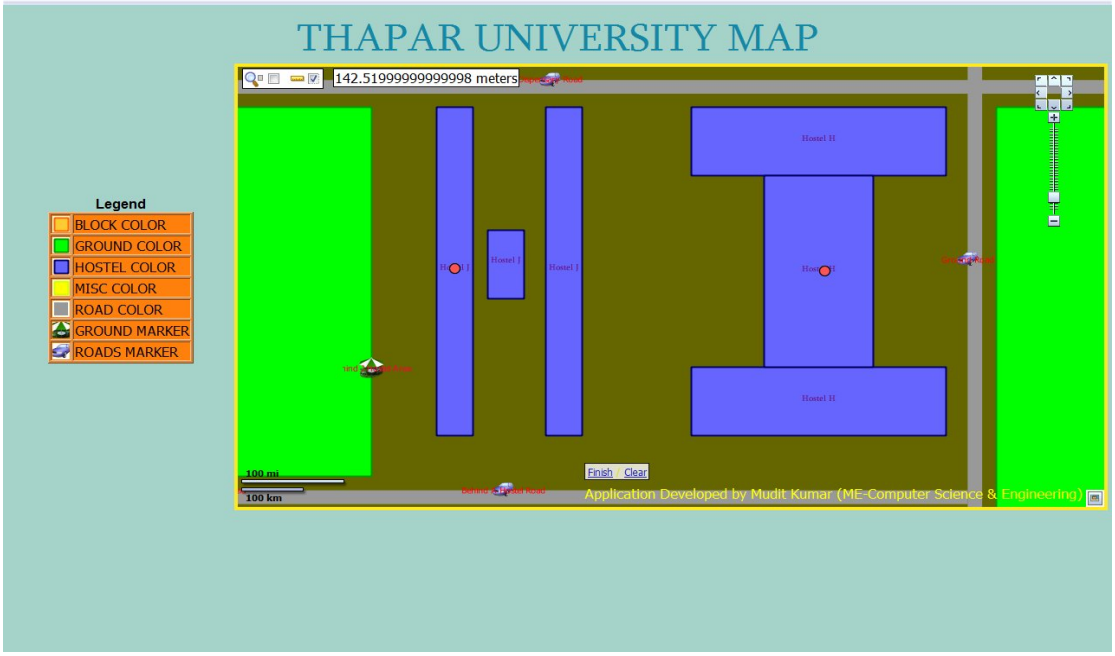

Figure 4.45: Map Showing Distance between Two Points

Figure 4.46 shows the distance travelled if we go serially through the red points marked on the map.

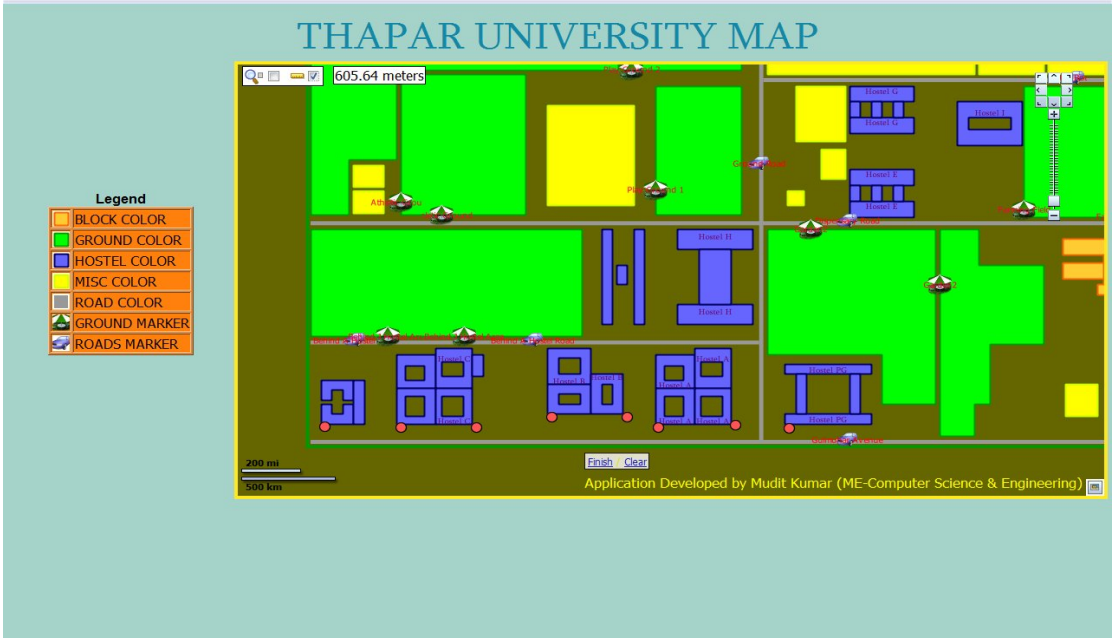

Figure 4.46: Map Showing Distance Covered through Various Points

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

Oracle Spatial can be used to design very powerful and robust spatial applications. It was not possible in case of traditional database to store, analyze, visualize, and integrate spatial data in spatial applications. Oracle Spatial is used to store and retrieve all spatial data used in the applications, and the spatial analysis performed in these applications is based on the methods and tools.

The process of designing and implementing spatial applications is not very simple because this process includes the use of various applications and tools. But, once we are familiar with the complete process of making of an application based on Oracle Spatial, it makes all the way easier.

In the design and implementation phase we have presented the overall process of making the Thapar University spatial application. In that process, we have created the spatial database for the Thapar University. This stored spatial data is used by Map Builder and MapViewer in various steps of application development. At last, the spatial application generates the map of Thapar University out of this spatial data. Thapar University spatial application possesses features like navigation panel, distance tool, marquee-zoom tool *etc.* which make the application more interactive.

Finally, we have an idea that Oracle Spatial is armed with very big concepts and suit of technologies that make it easier to draw our invisible thoughts in form of visible maps.

5.2 Future Scope

The thesis work can be extended in future on following aspects:

- The approach used in this thesis work can be used in developing applications for areas like Business GIS, Agriculture, Geology, Land Information System, Natural Hazard Management, Urban Planning, Archaeology, Environment,

Health, Military, Utility, Natural Resource Management, Corporate Case Studies *etc*.

- The application designed is using 2-d geometries in the map but further 3-d geometries can be used to draw locations in the Thapar University. In that case any real world location is considered as collection of 3-d geometries and then we can use special queries to store those 3-d geometries into the database.
- This application can be featured with more tools in order to make it more versatile. The examples of other features include theme-based FOI layer, Event Listeners for a Theme Based FOI layer, Add/Remove & Show/Hide FOI, Red Lining, Map Event Listeners, Multiple Base Maps, Circle Tool, Rectangle Tool, and many more.
- The feature of positioning and searching can be added to the application. In positioning, the map can be positioned on user given location. Searching feature searches the different types of locations around any given location and within the given range as well.

References

[1] P. Rigaux, M. Scholl and A. Voisard," Spatial Databases with applications to GIS", Morgan Kaufmann Publishers, 2002.

[2] Ravi Kothuri, Albert Godfrind and Euro Beinat,"Pro Oracle Spatial for Oracle Database 11g", Apress publishers, 2007.

[3] "Oracle spatial users' guide and reference 10g" Release 2 (10.2) B14255-01, June, 2005.
http://youngcow.net/doc/oracle10g/appdev.102/b14255.pdf

[4] Siva Ravada, Xavier Lopez, "Oracle Spatial10g".
www.nyoug.org/Presentations/2003/10gspatial.pdf

[5] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries, Proceeding of ACM SIGMOD, pp. 71–79, San Jose, CA, May 1995.

[6] Dan Geringer , Siva Ravada, "Oracle's Spatial Technologies". http://download.oracle.com/otndocs/products/spatial/pdf/spatial_features_jsirev.pdf

[7] David Sonnen, Henry D. Morris, "Oracle 10g: Spatial Capabilities for Enterprise Solutions", January, 2005.
images.autodesk.com/adsk/files/IDC-Oracle_Spatial_Capabilities_Enterprise_Solns.pdf

[8] Matt Bauer," Mapping Geometric Data with Oracle Spatial" Article. http://www.oreillynet.com/pub/a/network/2003/11/10/oracle_spatial.html

[9] Shashi Shekar and Sanjay Chawala, Spatial Database A Tour, Prentice Hall, 2003.

[10] "Oracle Location Based Services".
www.people.cis.ksu.edu/~hankley/d764/Slides08/764_23_Gudelli_OracleSpatial.ppt

[11] Liujian (LJ) Qian, "Developing spatial applications using Oracle Spatial and MapViewer", Spatial Group, Oracle.
www.ucgis.org/visualization/whitepapers/qia2.pdf

[12] Brian Klinkenberg, "Spatial Relationships in Spatial Analysis", August 30, 1997.
http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u15.html

[13] “GIS and Oracle Spatial Difference”, November 11, 2008. www.spatialdbadvisor.com/file_download/35/BADSIG_November_11th.pdf

[14] “Oracle Fusion Middleware, FMW MapViewer11g FAQ”, July 1, 2009. http://www.oracle.com/technology/products/mapviewer/index.html/mapviewer_faq_11gr1.pdf

[15] “Introduction to MapViewer”, Oracle Documentation. http://download.oracle.com/docs/cd/B10464_05/web.904/b10559/vis_star.htm

[16] Oracle Fusion Middleware User's Guide for Oracle MapViewer 11g Release 1 (11.1.1) E10145-01, May, 2009. www.oracle.com/technology/products/mapviewer/pdf/mapviewer11gr1_ug.pdf

[17] Oracle Fusion Middleware MapViewer 11g Technical Overview, July, 2009. http://www.oracle.com/technology/products/mapviewer/index.html/mapviewer11gr1_twp.pdf

[18] “Oracle Fusion Middleware, MapViewer11g Data Sheet”, July, 2009. http://www.oracle.com/technology/prpoducts/maviewer/index.html/mapviewer_datasheet_11gr1.pdf

[19] "Oracle Spatial 11g Geocoder", October, 2009. http://www.oracle.com/technology/products/spatial/pdf/11gr2_collateral/spatial11gr2_geocoder_twp.pdf

[20] "Oracle Spatial 11g GeoRaster", September, 2009. http://www.oracle.com/technology/products/spatial/pdf/11gr2_collateral/spatial11gr2_georaster_twp.pdf

[21] "Oracle Locator and Oracle Spatial 11g Best Practices", January, 2009. http://www.oracle.com/technology/products/spatial/pdf/spatial_wp09_bestprac.pdf

[22] Mudit Kumar, Parteek Bhatia, “Visualization of Oracle Spatial Data”, National Conference, NCACCET-10, Chandigarh College of Engineering and Technology, Chandigarh, January 29-30, 2010.

[23] Mudit Kumar, Parteek Bhatia, “MapViewer: A Tool for Geospatial Data Visualization on Oracle Database”, National Conference, ETIC-2010, Gurgaon Institute of Technology and Management, Gurgaon, Haryana, March 27, 2010.

[24] "Advance Spatial Data Management for Enterprise Applications", September, 2009.
http://www.oracle.com/technology/products/spatial/pdf/11gr2_collateral/spatial11gr2_wp_0922.pdf

List of Research Papers Published/Communicated

Published

[1] Mudit Kumar, Parteek Bhatia, "Visualization of Oracle Spatial Data", National Conference, NCACCET-10, Chandigarh College of Engineering and Technology, Chandigarh, January 29-30, 2010.

[2] Mudit Kumar, Parteek Bhatia, "MapViewer: A Tool for Geospatial Data Visualization on Oracle Database", National Conference on Emerging Trends in Information Technology & Computing (ETIC-2010), Gurgaon Institute of Technology and Management, Gurgaon, Haryana, March 27, 2010.

Communicated

[1] Mudit Kumar, Parteek Bhatia, "Spatial Application of a University using Oracle Spatial Database, MapViewer and Map Builder", communicated in ACST (Advances in Computational Sciences and Technology, International Research Journal).