Preparing for the Oracle Spatial Essentials Exam
Session Overview

OVERVIEW
• The Oracle “Spatial Essentials Certification” measures skills in a number of technical areas
• Organizations interested in obtaining Oracle PartnerNetwork “Oracle Spatial 11g Specialized” status must have at least one individual who holds this certification

CHALLENGES / OPPORTUNITIES
• The exam covers a wide range of topics—many of which users might not have had experience with
• Preparation can appear difficult

SOLUTIONS
• Learn about how the exam is structured and what types of questions are included
• Review the topics included and the general level of detail covered for each
• See demonstrations and use cases on the specialized concepts and topics on the exam
• Learn about classroom courses and online resources

RESULTS
• Gain a compact, comprehensive overview of all key Oracle Spatial technical capabilities, even if you don’t plan to register for the exam
• Learn how to prepare for the certification exam efficiently and effectively
• Prepare yourself to pass the exam and become a recognized expert in Oracle Spatial implementations
May 2014
Oracle Spatial Summit
May 21, 2014
Walter E. Washington Convention Center
Washington, DC  USA
Olivier Dubois
CEO, OSCARS

Albert Godfrind
Spatial Solutions Architect, Oracle

Steve Pierce
CEO, Think Huddle
Preparing for the Oracle Spatial Essentials Exam: Topic Review and Strategies
Overview

Certification vs Specialization

• **Individual Certification**

“The Oracle Spatial 11g Certified Implementation Specialist certification is designed for individuals who possess a strong technical background and exposure to Oracle Spatial 11g implementation. This certification exam covers topics such as: Oracle Spatial Concepts and Data Structures; Loading and Validating Spatial Data; Indexing Spatial Data; Performing Spatial Queries; Performing Spatial Processing; Oracle Fusion Middleware MapViewer; Geocoding, Routing, Spatial Analysis and Mining; Web Services; Advanced Indexing (Partitioning, Parallelism, Function-Based Indexes); Using Linear Referencing and Network and Topology Models; GeoRaster; Managing 3D and Lidar Data; Performance and Tuning / Exadata; and Workspace Manager. This certification differentiates candidates in the marketplace by providing a competitive edge through proven expertise. Up-to-date training and field experience are recommended.

This certification is available to all candidates but is geared toward members of the Oracle PartnerNetwork. OPN members earning this certification will be recognized as OPN Certified Specialists. This certification qualifies as competency criteria for the Oracle Spatial 11g Specialization.”
Overview
Certification vs Specialization

★ Company Specialization

“The Oracle Spatial 11g Specialization recognizes partner organizations that are proficient in selling, implementing and/or developing Oracle Spatial 11g solutions. Topics covered in this Specialization include: Oracle Spatial Concepts and Data Structures; Loading and Validating Spatial Data; Indexing Spatial Data; Performing Spatial Queries; Performing Spatial Processing; Oracle Fusion Middleware MapViewer; Geocoding, Routing, Spatial Analysis and Mining; Web Services; Advanced Indexing (Partitioning, Parallelism, Function-Based Indexes); Using Linear Referencing and Network and Topology Models; GeoRaster; Managing 3D and Lidar Data; Performance and Tuning / Exadata; and Workspace Manager.”
Overview
How to Get Certified

• 2-Step Certification Process
  – Preparation
    • Oracle Spatial: Essentials
    • Oracle Spatial: Advanced
    • Experience
  – Exam
    • Oracle Spatial 11g Essentials 1Z0-595
There are a number of challenges when preparing for the exam:

- The exam covers dozens of topics
- Many of those topics aren’t encountered frequently by spatial users
- Varying levels of detail are covered for each topic

Excerpt of topic list from Oracle University site
Overview

Included in this Presentation

• Overview of exam topics

• For each topic:
  – Subtopics covered
  – Examples of the type of knowledge to be tested, to include:
    • Demonstrations
    • Sample questions

• Strategies for preparing and taking the exam
Section 1:

Exam Topics
Overview

Exam Topics

1. Oracle Spatial Concepts and Data Structures
2. Loading and Validating Spatial Data
3. Indexing Spatial Data
4. Performing Spatial Queries
5. Performing Spatial Processing
6. Oracle Fusion Middleware MapViewer
7. Geocoding, Routing, Spatial Analysis and Mining
Overview

Exam Topics (continued)

8. Web Services
9. Advanced Indexing
10. Using Linear Referencing / Network & Topology Models
11. GeoRaster
12. Managing 3D and Lidar Data
13. Performance and Tuning / Exadata
14. Workspace Manager
Spatial Concepts & Data Structures
Topic 1

• What you need to know:
  – Describe the data types, data models, coordinate systems, indexing structure, query model, and types of spatial queries
  – Describe the Oracle Spatial schema and how spatial data is represented in the database
  – Explain spatial metadata structures and views
  – Explain coordinate systems and unit support, including Oracle data structures
Geometric Primitive Types

- Point
- Line string
- Arc line string
- Compound line string
- Polygon
- Polygon with one or more holes
- Arc polygon
- Compound polygon
- Optimized polygons
- Self-crossing line strings valid
- Self-crossing polygons not valid
Spatial Data Model

Spatial layer

Geometry

Element

Point

Geometry

Element

Line string

Geometry

Element

Polygon
Spatial Metadata & Coordinate Systems

- The spatial routines require you to populate a view that contains metadata about the SDO_GEOMETRY columns.
- For every SDO_GEOMETRY column, insert a row in the USER_SDO_GEOM_METADATA view.
- A coordinate system (CS) is a means of assigning coordinates to a location.
  - It establishes relationships between sets of coordinates.
- All spatial data has an associated coordinate system.
Spatial Query Model

Anatomy of a spatial query

Layer Data → Primary Filter
Spatial Index

Reduced Data Set → Secondary Filter
Spatial Functions

Exact Result

Spatial Column → Index retrieves area of interest
Spatial function determines exact result
Loading and Validating Spatial Data

Topic 2

• What you need to know:
  – Load spatial data from text files using SQL Loader and external tables
  – Load spatial data from GIS files using Oracle Map Builder, Oracle shapefile loader and GDAL/OGR
  – Perform transactional inserts
  – Describe data validation and correction routines in Oracle Spatial
  – Use export and import utilities to move spatial data between databases
  – Move spatial data between databases by using transportable tablespaces
LOAD DATA
CONTINUEIF NEXT(1:1) = '#'
INTO TABLE us_counties
FIELDS TERMINATED BY ';'
(county,
state,
poppsqmi,
geom COLUMN OBJECT (  
sdo_gtype INTEGER EXTERNAL,
sdo_srid CONSTANT 8307,
sdo_elem_info VARRAY TERMINATED BY '/' (e FLOAT EXTERNAL),
sdo_ordinates VARRAY TERMINATED BY '/' (o FLOAT EXTERNAL) )
)

Autauga;Alabama;57.428300000;
#2003;1;1003;1/
#-86.916969000;32.664028000;-86.816589000;32.659988000;-86.713409000;....;
#-87.765160000;31.297176000;-86.916969000;32.664028000/
Baldwin;Alabama;61.569000000;
#2003;1;1003;1/
#-87.765160000;31.297176000;-87.760429000;31.297289000;-87.759232000;....;
Using GDAL

- To setup
  - Add $<…>\texttt{gdal<xxxx>\bin}$ to your path
  - Set GDAL_DATA to $<…>\texttt{gdal<xxxx>\data}$ (optional)

- Main commands for vector data
  - ogrinfo = get information about a file (or spatial table)
  - ogr2ogr = copy from one format to another (import/export)

- Documentation
  - [http://www.gdal.org/ogr/ogr_formats.html](http://www.gdal.org/ogr/ogr_formats.html)
Using GDAL

```
ogr2ogr -f OCI OCI:scott/tiger@orcl112 world_countries.shp
 -lco DIM=2 -lco SRID=8307 -lco GEOMETRY_NAME=geometry -lco INDEX=NO
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f OCI</code></td>
<td>Use Oracle as output</td>
</tr>
<tr>
<td><code>OCI:scott/tiger@orcl112</code></td>
<td>Database connection</td>
</tr>
<tr>
<td><code>world_countries.shp</code></td>
<td>Input shape file</td>
</tr>
<tr>
<td><code>-lco DIM=2</code></td>
<td>Dimension (2D)</td>
</tr>
<tr>
<td><code>-lco SRID=8307</code></td>
<td>SRID</td>
</tr>
<tr>
<td><code>-lco GEOMETRY_NAME=geometry</code></td>
<td>Name of geometry column</td>
</tr>
<tr>
<td><code>-lco INDEX=NO</code></td>
<td>Do not create a spatial index</td>
</tr>
</tbody>
</table>
Validating Geometries

- Oracle Spatial validation routines ensure that spatial data in Oracle Spatial is valid.
  - SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT
    - Determines whether a single geometry is valid
  - SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT
    - Determines whether all geometries in a layer are valid
- If data is invalid, both routines return why and where the geometry is invalid.
Indexing Spatial Data

Topic 3

• What you need to know:
  – Explain R-tree indexing concepts, how R-tree indexes are built, and index structures
  – Create indexes, set index parameters, and create indexes in parallel
  – Describe index metadata and find index size
R-tree Indexing

• R-tree indexing is used to index spatial data.
  – Requires almost no configuration
  – Indexes two or three dimensions
• Primary filter (index only query) can operate on two or three dimensions
• Secondary filters are two dimensional or three dimensional
• Each index entry approximates geometry using Minimum Bounding Rectangle (MBR) for 2-D and Minimum Bounding Volume (MBV) for 3-D.
• MBRs and MBVs are indexed internally using a tree structure.
R-tree Indexing Concept

Leaf nodes of R-tree store <MBR, geometry pointer>
Building the R-Tree

- Fanout is the number of branches that comes out of each node.
- Oracle Spatial R-tree has the same fanout for all nodes.
A Look at R-tree Index Structures

```
CREATE INDEX us_states_sx
  ON us_states (geom)
  INDEXTYPE IS mdsys.spatial_index;
```

Index information

- Index tables are “opaque” structures
- **Do not change them in any way**

Table MDRT_7B50$
CREATE INDEX <index-name>
ON <table-name> (<column-name>)
INDEXTYPE IS MDSYS.Spatial_INDEX
[PARAMETERS ( '<parameter> = <value> ... <parameter> = <value>' ) ]
[PARALLEL [<parallel_degree>]];

• Parameters
  – LAYER_GTYPE
  – PARALLEL
  – SDO_INDEX_DIMS
  – SDO_RTR_PCTFREE
  – ...

29
Parallel Index Creation

- PARALLEL [<parallel_degree>]  
  - This parameter breaks the work of index creation into smaller pieces that can be performed in parallel. 
  - <parallel_degree> is an optional parameter that specifies the degree of parallelism. 
  - If the degree of parallelism is not specified, Oracle chooses a default based on the number of CPUs.

```
CREATE INDEX us_counties_sx
    ON us_counties(geom)
    INDEXTYPE IS MDSYS.SPATIAL_INDEX
    PARALLEL 4;
```
Spatial Index Dictionary Views

- **USER_SDO_INDEX_INFO**
  - Summary information

- **USER_SDO_INDEX_METADATA**
  - Detailed information

- Also “ALL” variants
USER_SDO_INDEX_INFO

<table>
<thead>
<tr>
<th>INDEX_NAME</th>
<th>VARCHAR2(32)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_NAME</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>VARCHAR2(2048)</td>
</tr>
<tr>
<td>SDO_INDEX_TYPE</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_INDEX_TABLE</td>
<td>VARCHAR2(32)</td>
</tr>
</tbody>
</table>

- Useful to relate a spatial index to its physical MDRT table
- Allows you to find out the size of the index
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDO_INDEX_OWNER</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_INDEX_TYPE</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_INDEX_NAME</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_INDEX_TABLE</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_INDEX_PRIMARY</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_INDEX_PARTITION</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_PARTITIONED</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_TNAME</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_COLUMN_NAME</td>
<td>VARCHAR2(2048)</td>
</tr>
<tr>
<td>SDO_INDEX_DIMS</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_RTREE_HEIGHT</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_RTREE_NUMNODES</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_RTREE_DIMENSIONALITY</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_RTREE_FANOUT</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_RTREE_SEQ_NAME</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_RTREE_XPND</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_ROUT_MBR</td>
<td>SDO_GEOMETRY</td>
</tr>
<tr>
<td>SDO_INDEX_STATUS</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_LAYOUT_GTYPE</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_LAYER_GTYPE</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_LEVEL</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_NUMTILES</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_MAXLEVEL</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_COMMIT_INTERVAL</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_FIXED_META</td>
<td>RAW(255)</td>
</tr>
<tr>
<td>SDO_TABLESPACE</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_INDEX_VERSION</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_INDEX_GEODETIC</td>
<td>VARCHAR2(8)</td>
</tr>
<tr>
<td>SDO_INDEX_NL_INDEX_TABLE</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_DML_BATCH_SIZE</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_INDEX_GEODIMENSIONALITY</td>
<td>NUMBER</td>
</tr>
<tr>
<td>SDO_INDEX_STATUS</td>
<td>VARCHAR2(32)</td>
</tr>
<tr>
<td>SDO_INDEX_GEOMETRY</td>
<td>SDO_GEOMETRY</td>
</tr>
</tbody>
</table>
Find Index Size

```
SELECT i.table_name, i.index_name, si.column_name,
       si.sdo_index_table, s.bytes
FROM   user_indexes i,
       user_sdo_index_info si,
       user_segments s
WHERE  i.index_type = 'DOMAIN'
AND    i.ityp_name = 'SPATIAL_INDEX'
AND    i.index_name = si.index_name
AND    s.segment_name = si.sdo_index_table
ORDER BY i.table_name, i.index_name;
```

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>INDEX_NAME</th>
<th>COLUMN_NAME</th>
<th>SDO_INDEX_TABLE</th>
<th>BYTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>US_CITIES</td>
<td>US_CITIES_SX</td>
<td>LOCATION</td>
<td>MDRT_1ABA5$</td>
<td>65536</td>
</tr>
<tr>
<td>US_COUNTIES</td>
<td>US_COUNTIES_SX</td>
<td>GEOM</td>
<td>MDRT_1ABAE$</td>
<td>327680</td>
</tr>
<tr>
<td>US_INTERSTATES</td>
<td>US_INTERSTATES_SX</td>
<td>GEOM</td>
<td>MDRT_1ABC1$</td>
<td>65536</td>
</tr>
<tr>
<td>US_PARKS</td>
<td>US_PARKS_SX</td>
<td>GEOM</td>
<td>MDRT_1ABCA$</td>
<td>65536</td>
</tr>
<tr>
<td>US_RIVERS</td>
<td>US_RIVERS_SX</td>
<td>GEOM</td>
<td>MDRT_1ABD4$</td>
<td>65536</td>
</tr>
<tr>
<td>US_STATES</td>
<td>US_STATES_SX</td>
<td>GEOM</td>
<td>MDRT_1ABB8$</td>
<td>65536</td>
</tr>
<tr>
<td>WORLD_CONTINENTS</td>
<td>WORLD_CONTINENTS_SX</td>
<td>GEOM</td>
<td>MDRT_1ABDD$</td>
<td>65536</td>
</tr>
<tr>
<td>WORLD_COUNTRIES</td>
<td>WORLD_COUNTRIES_SX</td>
<td>GEOM</td>
<td>MDRT_1ABE6$</td>
<td>65536</td>
</tr>
</tbody>
</table>
Demo
Performing Spatial Queries

Topic 4

• What you need to know:
  – Explain the differences between spatial operators and spatial functions
  – Write queries to determine spatial relationships and return geometries using SDO_RELATE and SDO_FILTER operators
  – Write queries returning results within a specific distance and nearest neighbors
  – Write queries finding correlations between two spatial layers
  – Write queries combining spatial and non-spatial criteria
Spatial Operators vs. Functions

- **Spatial operators:**
  - Require a spatial index on the first geometry specified in the operator
  - Take advantage of spatial indexes
  - Appear only in the WHERE clause
  - Implicitly transform the coordinate system of the window, if required

- **Spatial functions:**
  - Do not take advantage of spatial indexes!
  - Can be used on small tables that are not spatially indexed
  - Can be used in the SELECT list and the WHERE clause
  - Both geometries must exist in the same coordinate system.
Which parks fully inside the state of Wyoming?

```sql
SELECT p.id, p.name
FROM us_parks p, us_states s
WHERE s.state = 'Wyoming'
AND SDO_INSIDE (p.geom, s.geom) = 'TRUE';
```
SDO_WITHIN_DISTANCE example

How many customers are within 10 km of a sales region?

SELECT count(*)
FROM sales_regions r, customers c
WHERE r.region_id = 'R1'
AND sdo_within_distance (c.location, r.geom,
'distance=10 unit=km')
= 'TRUE';

SELECT count(*)
FROM sales_regions r, customers c
WHERE r.region_id = 'R1'
AND sdo_within_distance (c.location, r.geom,
'distance=10 unit=km')
= 'TRUE'
AND SDO_GEOM.SDO_DISTANCE (c.location, r.geom, 0.5) > 0;
Which cities are within 10 miles of an interstate?

```
SELECT c.city, i.interstate
FROM us_cities c,
     us_interstates i,
     TABLE(SDO_JOIN(
         'US_CITIES', 'LOCATION',
         'US_INTERSTATES', 'GEOM',
         'DISTANCE=10 UNIT=MILE')) j
WHERE j.rowid1 = c.rowid
  AND j.rowid2 = i.rowid
ORDER BY c.city;
```
Performing Spatial Processing

Topic 5

- What you need to know:
  - Perform area, length, and distance calculations using spatial functions
  - Create buffers, combine geometries, and derive geometries
  - Use spatial aggregate functions
  - Describe the functions for arc densification/coordinate transformation
  - Use spatial utility functions
  - Describe the functions for conversion from/to OGC formats
  - Generate GML documents from spatial objects
  - Modify geometries using PL/SQL
  - Use the Java API to work with geometries
The SDO_AREA Function

- **<geometry>**
  - SDO_GEOMETRY that defines a polygon
    - Can be a variable or table column
- **<tolerance>**
  - Number used as the tolerance
- **<unit>**
  - A quoted string with the units for the result
- Returned value
  - A number that is the area of the input polygon

```sql
area := SDO_GEOM.SDO_AREA
      (<geometry>, <tolerance> [, <unit>])
```
The SDO_LENGTH Function

- `<geometry>`: SDO_GEOMETRY that defines a polygon or line
  - Can be a variable or table column
- `<tolerance>`: Number used as the tolerance
- `<unit>`: Quoted string with the units for the result
- `<returned_value>`: Numeric length

```sql
length := SDO_GEOM.SDO_LENGTH
         ( <geometry>, <tolerance> [, <unit>] )
```
The SDO_DISTANCE Function

- `<geometry-1> <geometry-2>`: SDO_GEOMETRY objects
  - Can be variables or table columns
- `<tolerance>`: Numeric tolerance for the function
- `<unit>`: Quoted string with the units for the result
- `<returned_value>`: A number (the minimum distance between the geometries)

```
distance := SDO_GEOM.SDO_DISTANCE
            ( <geometry-1>, <geometry-2>,
              <tolerance> [, <unit>])
```
SDO_BUFFER Function

SDO_GEOM.SDO_BUFFER

• Generates a buffer polygon around a geometry
• Takes an SDO_GEOMETRY object as input
  – Any kind (point, line, polygon, compound)
  – Can buffer geodetic geometries
• Returns an SDO_GEOMETRY object containing the buffer (polygon)
Buffer Examples

• Simple geometries

• Collection geometries
The SDO_BUFFER Function

- `<geometry>`: SDO_GEOMETRY object to buffer
  - Can be a variable or table column
- `<distance>`: The buffer distance
- `<tolerance>`: A number used as the tolerance

```
SDO_GEOMETRY := SDO_GEOM.SDO_BUFFER
    ( <geometry>, <distance>,
      <tolerance> [, '<params>'] )
```

```
SELECT SDO_GEOM.SDO_BUFFER(location, 100, .05, 'unit=km') FROM cities
```
Geometric Intersection

- SDO_GEOM.SDO_UNION
- SDO_GEOM.SDO_INTERSECTION
- SDO_GEOM.SDO_DIFFERENCE
- SDO_GEOM.SDO_XOR
• UNION all the county boundaries for the states of New York and New Jersey, and generate a geometry for each state:

```sql
SELECT sdo_aggr_union(sdoaggrtype(geom, 0.5)),
       state
FROM us_counties
WHERE state_abrv in ('NY', 'NJ')
GROUP by state;
```

• Note: SDO_AGGR_SET_UNION
Coordinate Transformation Functions

- **SDO_CS.TRANSFORM**: Transforms a geometry from one coordinate system to another
- **SDO_CS.TRANSFORM_LAYER**: Transforms a layer from one coordinate system to another
The SDO_CS.TRANSFORM Function

• <geom>
  – Geometry of type SDO_GEOMETRY
  – Can be a variable or a table column

• <to_srid>
  – Spatial reference system ID to transform to

• Return value
  – Geometry of type SDO_GEOMETRY

```
SDO_GEOMETRY := SDO_CS.TRANSFORM (<geom>, <to_srid>)
```
SDO_CS.TRANSFORM: Example

- Single geometry transformed, for example, Hillsborough County in New Hampshire:

  ```sql
  SELECT sdo_cs.transform (geom, 82151)
  FROM us_counties_p
  WHERE county = 'Hillsborough'
  AND state = 'New Hampshire';
  ```

- Note: All transformations require valid SDO_SRID field set in source geometry.
Package SDO_UTIL

- simplify()
- circle_polygon()
- ellipse_polygon()
- polygonToLine()
- remove_duplicate_vertices
- append()
- concat_lines
- reverseLinestring()

- getNumElem
- getNumVertices
- convert_unit
- point_at_bearing
- AffineTransforms
OGC Methods

• GET_WKT
  – Returns Well Known Text format of SDO_GEOMETRY

• GET_WKB
  – Returns Well Known Binary format of SDO_GEOMETRY

• Extract Geometries to WKT

```sql
SELECT c.geom.get_wkt()
FROM us_counties c
WHERE county = 'Denver';
```

```
POLYGON (  
  (-105.052597 39.791199, -105.064606 39.789928, ...
  ... -105.024757 39.790947, -105.052597 39.791199),
  (-104.933578 39.698139, -104.936104 39.698299, ...
  ... -104.9338 39.696701, -104.933578 39.698139))
```
SELECT sdo_util.to_gmlgeometry(location)
FROM us_cities
WHERE state_abrv = 'CO';

<gml:Point srsName="SDO:8307"
  xmlns:gml="http://www.opengis.net/gml">
  <gml:coordinates decimal="." cs="," ts=" ">
    -104.872655,39.768035
  </gml:coordinates>
</gml:Point>
Oracle Fusion Middleware MapViewer

Topic 6

• What you need to know:
  – Explain the use of Oracle Fusion Middleware MapViewer
  – Describe the architecture of MapViewer
  – Install, configure, and administer MapViewer using OC4J and WebLogic Server
  – Define maps using styles, themes, and maps

  – Define and manage tile caches
  – Build applications using the Oracle Maps tutorial
  – Integrate external data
  – Define and access OGC Web Mapping Service and Web Feature Services

*Resource:  MapViewer Primer
Oracle Application Server MapViewer

- A standard feature of all versions of Fusion Middleware
  - Oracle Application Server
  - Weblogic Server
- XML, Java and JSP APIs
- Map definition tool
- Map definitions and symbology stored in the database
- Thematic mapping
- Produces PNG, GIF, JPEG, SVG
- AJAX API for rich user interfaces
- High performance map cache
- Integration with Business Intelligence tools
MapViewer "AJAX" Architecture: Oracle Maps
Map Definitions

• **Style**: defines how shapes should be rendered
  - Areas: color, transparency, boundary thickness and color, …
  - Lines: color, thickness, center lines, wing lines, hash marks, …
  - Points: symbols, graphics, …
  - Texts: font, styling, color, size, halo, …

• **Theme**: associates a spatial table and a style
  - Graphic style and text style (for labels)

• **Map**: groups a number of themes
  - Visibility of themes based on scale

• **Map Cache**: makes a base map available to web applications
Updating Map Definitions

• Definitions are stored in the database
  – USER_SDO_STYLES
  – USER_SDO_THEMES
  – USER_SDO_MAPS
  – USER_SDO_CACHED_MAPS

• Updated using MapBuilder
USER_SDO_STYLES

- Style information stored in the database
  - Text, colors, line styles, area and fill information, markers
  - Advanced styles for thematic mapping based on a column value
- Includes XML definition of each style
- Can add styles easily using Mapbuilder

```sql
SQL> describe user_sdo_styles
Name        Type
-----------  --------
NAME         VARCHAR2(32)
TYPE         VARCHAR2(32)
DESCRIPTION  VARCHAR2(4000)
DEFINITION   CLOB
IMAGE        BLOB
GEOMETRY     MDSYS.SDO_GEOMETRY
```
• Stores user-defined themes
• User specifies:
  – Theme name and description
  – Table name
  – Geometry column name
  – Styling rules in XML based on values in USER_SDO/styles

SQL> describe user_sdo_themes
Name               Type
------------------ ---------------
NAME               VARCHAR2(32)
DESCRIPTION        VARCHAR2(4000)
BASE_TABLE         VARCHAR2(32)
GEOMETRY_COLUMN    VARCHAR2(2048)
STYLING_RULES      CLOB
What Are Themes?

• A theme applies one or more styles to a geometry column
  – Feature drawing information (color, fill, marker, and so on)
  – Text and label font, color, size
• Thematic mapping is supported by associating an advanced style to a theme.
• Themes can be based on any kind of spatial data:
  – Vector (SDO_GEOMETRY)
  – Raster (SDO_GEORASTER)
  – Network, topology ,…
  – WMS or WFS servers
• Possible to preview the results …
USER_SDO_MAPS

- A map is a collection of themes
- User specifies a map name and (optionally) a description
- User specifies an XML definition:
  - Theme names to display
  - Zoom scale information to specify when themes display

SQL> describe user_sdo_maps
Name                  Type
--------------------- -------------------
NAME                  VARCHAR2(32)
DESCRIPTION           VARCHAR2(4000)
DEFINITION            CLOB
What Are Maps?

- A map groups one or more themes
- Can specify “min scale” and/or “max scale” values to control the visibility of each theme depending on map scale.
- Can also control label visibility
- Use ratio-scales (not “mapviewer native”)
- Can control the order in which themes are rendered
- A theme can be used in many different maps
Updating Definitions

- Changes are only applied to the database tables
  - `USER_SDO_STYLES`, `_THEMES` and `_MAPS`
- Existing applications will not see the changes
  - Definitions are cached in memory
  - Clear the cache:

- Map Tile Caches are also invalid now!
  - Rebuild the cache!
Integrating External Data Sources

• From the browser
  – Use an eLocation background (« Oracle Maps »)
    • Only for Oracle tools and applications!
  – Use a Google Maps or Microsoft Bing Maps background
  – Access any other tile server
    • Custom map tile layer

• From the server
  – Use WMS or WFS servers
  – Data will be cached in the map cache
Using an Eolocation Background

- Saves you all the effort of buying, loading and maintaining your own spatial data for the base maps.
- Also saves you the effort of designing a base map from scratch.
- Free for use by applications that use Oracle.
- Check out terms and conditions on [http://elocation.oracle.com/elocation/legal.html](http://elocation.oracle.com/elocation/legal.html)

```javascript
mapview.addMapTileLayer(new MVMMapTileLayer("elocation_mercator.world_map", "http://elocation.oracle.com/mapviewer/mcserver") );
```
Using a Google or Bing Maps

Background

- Saves you all the effort of buying, loading and maintaining your own spatial data for the base maps.
- Also saves you the effort of designing a base map from scratch.
- Google or Bing APIs automatically loaded
- Free for private use, not for commercial use!
  - Check out access restrictions

```javascript
var basemap = new MVBingTileLayer({key:<your_bing_maps_key> });
mapview.addMapTileLayer(basemap);
```
WMS Themes

Choose layers

Choose coordinate system
Oracle Maps: the Javascript API

```html
<html>
<head>
<script language="Javascript" src="/mapviewer/fsmc/jslib/oraclemaps.js"></script>
<script language=JavaScript>
var baseURL = "http://"+document.location.host+"/mapviewer";
function display_map() {
  var mapview = new MVMapView (document.getElementById("MAP_DIV"), baseURL);
  mapview.addMapTileLayer(new MVMapTileLayer("mvdemo.demo_map"));
  mapview.setCenter(-122.45, 37.75);
  mapview.setZoomLevel(5);
  mapview.addNavigationPanel("east");
  var foiLayer = new MVThemeBasedFOI ('customers', 'mvdemo.customers');
  mapview.addThemeBasedFOI (foiLayer);
  mapview.display();
}
</script>
</head>
<body onload=display_map() >
  <div id="MAP_DIV" style="left:10; top:60;width: 600px; height: 500px"></div>
</body>
</html>
```
Geocoding, Routing, Spatial Analysis and Mining

Topic 7

• What you need to know:
  – Describe geocoding concepts, process, and functionality provided in Oracle Spatial
  – Describe the geocoding data model
  – Describe geocoding functions and structure of address results
  – Explain point addressing, structured address geocoding, and reverse geocoding
  – Install, configure, and use geocoding services with the XML API
  – Describe where to obtain data to use with the geocoding and routing engines (…)

73
What you need to know (continued):

– Describe the capabilities and architecture of the routing engine
– Describe the routing engine tables
– Install and configure the routing engine
– Formulate XML route requests and describe the structure of route responses
– Explain the concept of network partitioning
– Partition the network used by the routing engine
– Describe the uses and capabilities of the spatial analysis and mining functions
The Geocoding Process

1. Address Parsing
2. Searching and Cleansing
3. Coordinate Generation

Reference Data for Geocoding:
- Address Structure
- Street and place names
- Geometries

Address to Match
Coordinates + Corrected Address
Geocoding Tables

Metadata tables
- Describe the organization of the geocoding data
- Define address structure and parsing rules

Data tables
- Can have multiple sets of data
- One per country, or multiple countries per set
- Typically one set per country, with a country suffix
PL/SQL API Example

```sql
SELECT SDO_GCDR.GEOCODE('SCOTT',
   SDO_KEYWORDARRAY('Clay Street', 'San Francisco, CA'),
   'US', 'DEFAULT') GEO_ADDR
FROM DUAL;
```

MATCHCODE 1 = exact match with provided input
Deploy & use Geocoding web service

• Deploy **GEOCODER.EAR** in your application server
  – File is in $ORACLE_HOME/md/jlib
  – The initial startup fails because the default configuration has no database connection
  – [Geocoder demonstration](#)

• Update geocoder configuration
  – Using geocoder console: [http://<server>/geocoder/admin.jsp](http://<server>/geocoder/admin.jsp)
  – Or manually update file $WEB-INF/config/geocodercfg.xml
  – Set proper database connection to database user that owns the geocoding tables
  – Restart the geocoder application
Routing

- Architecture
- Routing tables
- Installation and configuration
- Partitioning the network
- How to formulate route requests
- Structure of the route responses
Routing Web Service Architecture

Client

Application

XML/HTTP

Java Environment

Routing Engine

JDBC

Database

Routing Tables
Routing tables
A route request

An XML request for a single route includes:

• A numeric route ID
• One start and one end location
• Optionally, one or more attributes:
  – Route preference: fastest or shortest
  – Road preference: highway or local
  – Vehicle type and parameters (Oracle Database 11g Release 2)
  – Whether to return the geometry of the route
  – Time and distance units
  – Language for driving directions
  – And more …
Spatial Analysis and Mining

Correlate data based on location (spatial correlation).

*Neighborhood analysis*
  - to determine specific information about an area of interest
  - Proportion of theme layer geometries overlapping geometry of interest is applied to aggregate analysis

*Spatial binning*
  - to classify data based on location

*Spatial clustering*
  - to determine patterns based on location

*Co-location analysis*
  - to determine how the location of one thing correlates to the location of something else

All functions are in the SDO_SAM package.
Spatial Analysis and Mining

- Find how large a population lives in a radius of 3 miles around a point

```sql
SELECT sdo_sam.aggregates_for_geometry(
    'US_COUNTIES', 'GEOM','sum', 'totpop',
    SDO_GEOMETRY(2001, 8307,
        SDO_POINT_TYPE(-73.943849, 40.6698,NULL),NULL,NULL),
    'distance=3 unit=mile')
FROM DUAL;
```

724464.319
Spatial Analysis and Mining

- The area covers counties in NY

<table>
<thead>
<tr>
<th>COUNTY</th>
<th>ST</th>
<th>TOTPOP</th>
<th>COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queens</td>
<td>NY</td>
<td>1951598</td>
<td>0.5437756</td>
</tr>
<tr>
<td>Kings</td>
<td>NY</td>
<td>2300664</td>
<td>31.0430579</td>
</tr>
<tr>
<td>New York</td>
<td>NY</td>
<td>1487536</td>
<td>0.0188785</td>
</tr>
</tbody>
</table>
Web Services

Topic 8

• What you need to know:
  – Describe the four major Open Geospatial Consortium standards for spatial web services and the services they enable
  – Describe the architecture, configuration steps, and example queries for Web Mapping Service for Oracle Database
  – Describe the architecture, configuration steps, and example queries for Oracle Database for Web Feature Service
  – Use WMS/WFS themes in MapViewer applications (…)

86
Web Services

Topic 8

• What you need to know (continued):
  – Describe the architecture, configuration steps, and example queries for Oracle Database for Open Location Services
  – Describe the architecture, configuration steps, and example queries for Oracle Database for Catalog Service
  – Explain the Oracle licensing requirements
Open Geospatial Consortium Standards

• **Web Map Service (WMS)**
  - Request/Provide maps
  - Request/Provide information about content of a map

• **Web Feature Service (WFS)**
  - Request/Provide data
  - Access/search/update/delete geospatial feature instances
  - Based on spatial/non-spatial search criteria
  - Using a standard interface over the web

• **Catalog Service for Web (CSW)**
  - Defines a common interface that enables diverse but conformant applications to perform discovery, browse and query operations against catalog servers.

• **Location Services (OpenLS)**
  - Geocoding Service
  - Routing Service
  - Mapping Service
  - Directory Service
MapViewer WMS

Client

WMS Requests/responses

HTTP

Application
Server

WMS Client

WMS Server API

Map Rendering Engine

JDBC

Database

Spatial Tables

Map Definitions

Mapbuilder

89
Web Feature Service

• OGC standard (and ISO 19142)
  • current version 1.1

• A WFS is a Web interface that allows one to publish and deploy geographic feature data locally or across the Internet.

• Two types of services:
  • “standard” WFS
  • “transactional” WFS (WFS-T)

• WFS-T operations include the ability to insert, delete, update, get and query features on spatial and non-spatial constraints
Advanced Indexing

Topic 9

• What you need to know:
  – Explain the concept of table and index partitioning
  – Describe the benefits of partitioning for spatial tables
  – Partition a spatial index using range partitioning
  – Partition spatial data based on location
  – Explain why spatial queries achieve performance benefits from partitioned indexes
  – Create indexes in parallel
  – Explain where parallel query is supported
  – Describe and create a spatial function-based index
Partitioning

- Table partitioning is a key feature of the Oracle database.
  - An option of the Enterprise Edition database
- One table decomposes to multiple “physical tables”, called partitions
- The decomposition is based on the value of a partitioning key (single or multiple columns)
- Indexes can also be partitioned
- Partitions (table or index) can be mapped to different tablespaces
- Transparent to applications

Table T1

| P1 | P2 | ... | ... | ... | Pn |
Benefits of Partitioning

• Maintenance:
  – Partition-level index rebuilds
  – Partition-level table and index archives
  – Partition-level table and index exchanges, splits, merges

• Performance:
  – Reduce response times for long-running queries: local index scan
  – Single queries that search multiple index partitions can be parallelized
Creating a Partitioned Table

- A table is partitioned at creation time
- Can add/remove/split/merge partitions later.

```sql
CREATE TABLE yellow_pages_part (
    id       NUMBER, name VARCHAR2(50),
    category NUMBER, location SDO_GEOMETRY
) PARTITION BY RANGE (category)(
    PARTITION p1 VALUES LESS THAN (2),
    PARTITION p2 VALUES LESS THAN (3),
    PARTITION p3 VALUES LESS THAN (4),
    PARTITION p4 VALUES LESS THAN (5),
    PARTITION p5 VALUES LESS THAN (6),
    PARTITION p6 VALUES LESS THAN (MAXVALUE)
);
```
Create a Partitioned Spatial Index

- Use the LOCAL keyword

```sql
CREATE INDEX yp_part_sidx ON yellow_pages_part (location)
  INDEXTYPE IS MDSYS.SPATIAL_INDEX
LOCAL;
```

- Can specify partition-specific parameters

```sql
CREATE INDEX yp_part_sidx ON yellow_pages_part (location)
  INDEXTYPE IS MDSYS.SPATIAL_INDEX
LOCAL (  
  PARTITION p1 PARAMETERS ('TABLESPACE=TBS_1'),
  ...
  PARTITION p6 PARAMETERS ('TABLESPACE=TBS_6')
);
```
Partitioned Index Tip

- Create the index without actually building it

```sql
CREATE INDEX yp_part_sidx ON yellow_pages_part (location)
  INDEXTYPE IS MDSYS.Spatial_Index
LOCAL (
  PARTITION p1 PARAMETERS ('TABLESPACE=TBS_1'),
  PARTITION p6 PARAMETERS ('TABLESPACE=TBS_6')
) UNUSABLE ;
```

- Build individual index partitions

```sql
ALTER INDEX yp_part_sidx REBUILD PARTITION p1;
ALTER INDEX yp_part_sidx REBUILD PARTITION p6;
```
Spatially Partitioned Table and Index
Spatial Partition Pruning

- Used with these Spatial operators
  - SDO_WITHIN_DISTANCE
  - SDO_FILTER
  - SDO_RELATE
- Completely transparent and automatic
- Requires no partition key in the WHERE clause
  - Including a spatial partition key may be incorrect
- Occurs at run time
Parallel Spatial Index Creation

CREATE INDEX .... PARALLEL 4;

Four partitions, local indexes with parallel degree = 4

Index Partition 1  
R-tree Index 1

Index Partition 2  
R-tree Index 2

Index Partition 3  
R-tree Index 3

Index Partition 4  
R-tree Index 4
Function-Based Indexes

• Build an index on the results of a function.
  – If a search by last name was often done in uppercase index can be built using UPPER(LAST_NAME)

• Build a *spatial* index on the result of a function that returns a geometry:
  – Offers a powerful mechanism to enable Oracle Spatial functionality without an SDO_GEOMETRY column in the table.
Function-Based Indexes: Example

- This is a table that does not contain any geometry column
- Only two columns that contain the coordinates of the cities:

```sql
CREATE TABLE us_cities_xy (  
  city varchar2(20),  
  state_abrv char(2),  
  pop90 number,  
  rank90 number,  
  longitude number,  
  latitude number
);
```
Function-Based Indexes: Example

- The following function transforms the longitude and latitude columns into a geometry object:

```sql
CREATE OR REPLACE FUNCTION get_point (longitude NUMBER, latitude NUMBER) RETURN sdo_geometry DETERMINISTIC IS BEGIN RETURN sdo_geometry (2001, 8307, sdo_point_type (longitude, latitude, NULL), NULL, NULL); END; /
```
Function-Based Indexes: Example

• Setup spatial metadata:

```sql
INSERT INTO user_sdo_geom_metadata values ('US_CITIES_XY', 'SCOTT.GET_POINT(LONGITUDE,LATITUDE)',
  sdo_dim_array(
    sdo_dim_element('LONG', -180, 180, 0.5),
    sdo_dim_element('LAT', -90, 90, 0.5)),
  8307);
```

• You MUST specify explicitly the name of the owner of the function!
Function-Based Indexes: Example

• Create the spatial index on the function

```
CREATE INDEX us_cities_xy_sx ON
  us_cities_xy( get_point(Tongitude,latitude) )
INDEXTYPE IS mdsys.spatial_index;
```
Using Linear Referencing and Network and Topology Models

Topic 10

• What you need to know:
  – Explain linear referencing concepts, use cases, and data structures
  – Convert geometries to LRS structures and use the main dynamic segmentation, point, and validation functions
  – Describe network modeling concepts, use cases, and data structures
  – Define, load, and manage networks in the Oracle Spatial network data model using the PL/SQL API
  – Explain the concept of load-on-demand and network partitioning (…)
Using Linear Referencing and Network and Topology Models

Topic 10

• What you need to know (continued):
  – Describe network analysis functions available with the load-on-demand Java API for analysis
  – Use the network tutorial to build an application using network analysis
  – Describe topology data model concepts, advantages, use cases, and storage model
  – Define and load topologies, define features, and perform topology queries and editing using the PL/SQL and Java APIs
What Is LRS?

- LRS is a mechanism to associate a **measure** value with a 2-D or 3-D **point** along a **line string**, multiline string, or polygon.
- Measure value is typically **proportional** to the distance from the start measure of the geometry.
- Measure values must be either **increasing** or **decreasing** along the linear-referenced segment.

![Diagram showing LRS points](Image)
Network Data Model

Oracle Spatial Network Data Model Demo

Shortest Path Analysis
Left click for start point, right click for end point, or manually enter node ID, link ID, percentage, or address.
Start 159486717
End 159486717

Network Constraints
Network Edge/Node select/deselect
custom NetworkFeatureConstraint
custom ProhibitedZoneConstraint
custom spatial reider ndm True/HeightConstraint
custom spatial reider ndm True/HazardConstraint

Prohibited Zone

Link Cost Calculators
Default/Link Cost Calculator

Keep Previous Results
Reverse Direction
Find Shortest Path

Analytic Result:
159486717 (344.4 km)
(time 0.019 s)
Time to analyze the network: 0.019 s
Time to compute geometries: 0.019 s.
Network Data Model
Isochrones
Network Data Structures

- Metadata
- Nodes
- Links
- (Paths)
- (Path Links)
- (Partitions)
- (Partition BLOBS)
Network Partitioning

Problem

• Network analysis uses a memory-resident copy of the network.
• Large networks (>1000000 nodes) hard to use
  • Memory limitation of the Java runtime
  • Long initialization times

Solution

• Divide the network into subsections called “partitions”
• Load partitions as needed
• Limit the number of partitions in memory
Fundamental Analysis Functions

• `shortestPathDijkstra()` and `shortestPathAStar()`
  • Find the shortest path between two nodes
• `nearestNeighbors()` and `nearestReachingNeighbors()`
  • Find the N nearest nodes from a node
• `withinCost()` and `withinReachingCost()`
  • Find all nodes within a chosen distance (=cost) from a node
Geometry vs. Topology

Geometry Storage

Topology Storage
Creating and Populating Topologies

Create the Topology

Create the Feature Tables

Register the Feature Tables with the Topology

Generate Topology Primitives and Features from existing Geometries

Populate the Topology Primitives

Define the Features using the Topology Primitives

Define Hierarchical Features

Perform Queries

Perform Edits
Editing Topologies

1. Create TopoMap object (CREATE_TOPO_MAP)
2. Load TopoMap object for update (LOAD_TOPO_MAP)
3. Perform editing operations (for example, add 1000 nodes)
   - Validate cache (VALIDATE_TOPO_MAP)
   - Rebuild indexes (CREATE_EDGE_INDEX and CREATE_FACE_INDEX) (for example, after each 100 added nodes)
4. Update topology (UPDATE_TOPO_MAP) (for example, after each 1000 added nodes)
5. Commit changes (COMMIT_TOPO_MAP)
6. Remove TopoMap object (DROP_TOPO_MAP)
7. Clear cache (CLEAR_TOPO_MAP)
What you need to know:

- Describe raster data concepts
- Describe the Oracle Spatial GeoRaster storage model
- Explain the compression techniques available/benefits
- Load, export, use, and manage rasters in Oracle Spatial GeoRaster
- Use the open source GDAL/OGR library and tools to load and export rasters between other data formats and GeoRaster
- View/manage raster data by using the features of the GeoRaster viewing tool (…)
GeoRaster

Topic 11

• What you need to know (continued):
  – Describe the GeoRaster functions for managing and manipulating rasters, and the Java API features and functions
  – Describe available 3rd party and open source GeoRaster processing and analysis tools available
  – Use MapViewer to display rasters and overlay them with vectors
## Raster/Vector Data Differences

<table>
<thead>
<tr>
<th>Vector Data</th>
<th>Vector Coordinates</th>
<th>Raster Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Vector Data" /></td>
<td>-74.1651749, 41.339141</td>
<td><img src="image" alt="Raster Data" /></td>
</tr>
<tr>
<td><img src="image" alt="Vector Data" /></td>
<td>-74.1651749, 41.339141, -73.4284481, 40.678193, -72.9792214, 41.686228</td>
<td><img src="image" alt="Raster Data" /></td>
</tr>
<tr>
<td><img src="image" alt="Vector Data" /></td>
<td>-74.1651749, 41.339141, -74.1651749, 39.559004, -72.9792214, 39.559004, -72.9792214, 41.339141</td>
<td><img src="image" alt="Raster Data" /></td>
</tr>
</tbody>
</table>
Raster Data and Cell Size

Coarser resolution

Finer resolution
What is a Raster?

- Two dimensional array of regularly spaced elements (*pixels or cells*)
  - Orthophotos
  - Remote Sensing
  - Gridded data (raster GIS)
- Image data is collected by a variety of technologies
  - Satellite remote sensing
  - Airborne photogrammetry
  - Sonar
- Digital images can be composed of one or more bands
  - Bands often represent an interval of wavelengths along the electromagnetic spectrum
  - Band data can be simultaneously recorded
Cells and Bands

- Cell data is the value associated with the cells/pixels
  - Each cell has a value in a raster
  - Each value is associated with a cell/pixel
  - In an RGB image, each cell has 3 values: one each to represent the intensity of red, green, and blue

- In GeoRaster, all cells associated with a 2-D set of rows and columns in a plane is a band
  - An RGB image will have 3 bands
  - A multispectral image with 4 channels will have 4 bands in GeoRaster
  - Grid data with 4 bands will have 4 bands in GeoRaster
Some bands accentuate different features

Multi band Images
Resolution Pyramid

Typically increases storage cost by 33% to 40%.
Blocking

- A GeoRaster image can be composed of an extremely large number of cells
- It is more efficient in terms of storage and retrieval to break large images into smaller blocks
- In GeoRaster, users/applications can determine how data is blocked
  - Specify rows, columns, and optionally bands

4 x 4 blocks
Compression

JPEG Compression

- Lossy compression
- For rasters with cellDepth=8BIT_U and no more than 4 bands per block
- JPEG-B or JPEG-F mode
- Control the compression level using the quality parameter
  - 0 (max compression) to 100 (no compression)

DEFLATE Compression

- Lossless compression
- Uses the ZLIB format

JPEG2000 and MrSid Compression

- Via a plugin from LizardTech
- Also loader for JP2/SID files

You can also use securefiles (blob) compression. This is part of the Advanced Compression option.
The Impact of Compression

- Consider one TIFF file of 53 MB (uncompressed)
- When compressed as JPEG-F it is reduced to 4.4 MB
- This a compression ratio of 1:12
- Add 50% to accommodate the pyramid (approx 2.1 MB)
- Total storage will be approx 6.5 MB
- This is now a compression ratio of 1:8
Using GDAL

- To setup
  - Add `<...>\gdal<xxxx>\bin` to your path
  - Set `GDAL_DATA` to `<...>\gdal<xxxx>\data` (optional)

- Main commands for raster data
  - `gdalinfo` = get information about a file (or spatial table)
  - `gdal_translate` = copy from one format to another (import/export)

- Documentation
  - [http://www.gdal.org/formats_list.html](http://www.gdal.org/formats_list.html)
  - [http://www.gdal.org/frmt_georaster.html](http://www.gdal.org/frmt_georaster.html)
GDAL supported Formats

- See complete list here: [http://www.gdal.org/formats_list.html](http://www.gdal.org/formats_list.html)
- Some formats are “built-in”, some use plugins
- Find details about one format:
  - D:\> gdalinfo --format georaster
  - Useful to find out the syntax of parameters specific to this format

D:\> gdalinfo --formats

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAIGrid (rw)</td>
<td>Arc/Info ASCII Grid</td>
</tr>
<tr>
<td>AIG (ro)</td>
<td>Arc/Info Binary Grid</td>
</tr>
<tr>
<td>EIR (ro)</td>
<td>Erdas Imagine Raw</td>
</tr>
<tr>
<td>ELAS (rw+)</td>
<td>ELAS</td>
</tr>
<tr>
<td>ERS (rw+)</td>
<td>ERMapper .ers Labelled</td>
</tr>
<tr>
<td>ESAT (ro)</td>
<td>Envisat Image Format</td>
</tr>
<tr>
<td>GTiff (rw+)</td>
<td>GeoTIFF</td>
</tr>
<tr>
<td>GXF (ro)</td>
<td>GeoSoft Grid Exchange Format</td>
</tr>
<tr>
<td>GeoRaster (rw+)</td>
<td>Oracle Spatial GeoRaster</td>
</tr>
<tr>
<td>HFA (rw+)</td>
<td>Erdas Imagine Images (.img)</td>
</tr>
<tr>
<td>JPEG (rw)</td>
<td>JPEG JFIF</td>
</tr>
<tr>
<td>VRT (rw+)</td>
<td>Virtual Raster</td>
</tr>
<tr>
<td>XPM (rw)</td>
<td>X11 PixMap Format</td>
</tr>
</tbody>
</table>

Oracle Georaster plugin is installed
Main GDAL Utilities

• **gdal_translate**
  - Main tool to convert a raster from one format to another (like TIFF to JPEG)
  - **Will be used to load rasters into Oracle as well as to export from Oracle**
  - Can be used to change the structure of a raster from BSQ to BIP (so that the JAI classes can read it)
  - Can also be used to extract the georeferencing information from a GeoTIFF raster into a TFW file
  - Many other conversions are possible, such as breaking a very large file into a set of smaller files or fusing many into a single large raster (mosaicing)
Other GDAL Utilities

- gdalinfo
  - Prints out information about a raster
  - Useful to find out any georeferencing and projection information as well as the structure of a raster
- gdaladdo
  - Adds overviews to a raster, i.e. generate a resolution pyramid
- gdalwarp
  - Reprojects a raster to a different coordinate system

Many more available. See: [http://www.gdal.org/gdal_utilities.html](http://www.gdal.org/gdal_utilities.html)
Supported formats: see: [http://www.gdal.org/formats_list.html](http://www.gdal.org/formats_list.html)
Oracle Georaster: see: [http://www.gdal.org/frmt_georaster.html](http://www.gdal.org/frmt_georaster.html)
Importing a raster from a file

```
gdal_translate -of georaster sf1.tif
   georaster:scott/tiger@orcl111,us_rasters,georaster
   -co blockxsize=512 -co blockysize=512 -co blockbsize=3
   -co interleave=bip -co srid=26943
   -co "insert=values (1, 'sf1.tif', 'Aerial photo San Francisco 1',
       sdo_geor.init('us_rasters_rdt_01', 1),null,null)"
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-of georaster</code></td>
<td>Use Oracle Georaster as output</td>
</tr>
<tr>
<td><code>sf1.tif</code></td>
<td>Name of input TIFF file</td>
</tr>
<tr>
<td><code>georaster:scott/tiger@orcl112</code></td>
<td>Database connection</td>
</tr>
<tr>
<td><code>us_rasters,georaster</code></td>
<td>Destination table and column</td>
</tr>
<tr>
<td><code>-co blocksize</code></td>
<td>Block size in x, y and b (band)</td>
</tr>
<tr>
<td><code>-co SRID=26943</code></td>
<td>Coordinate system of the raster</td>
</tr>
<tr>
<td><code>-co interleave</code></td>
<td>Interleaving (default is band sequential)</td>
</tr>
<tr>
<td><code>-co insert</code></td>
<td>The VALUES of an insert statement to insert a row in the raster table</td>
</tr>
</tbody>
</table>
Managing 3D and Lidar Data

Topic 12

• What you need to know:
  – Describe 3D data concepts, objects/models supported, coordinate systems, and data structures for modeling surfaces/solids
  – Explain 3D operations available and how to transform 2D to 3D
  – Convert data into CityGML and KML for publishing/viewing, load CityGML data, and query 3D data using SQL

  – Describe point clouds and their storage model
  – Load point clouds and describe point cloud processing functions
  – Describe triangulated irregular network concepts, storage model, and available functions for data management/extraction/conversion
Spatial 3D Data Model

Hierarchical Decomposition

- Composite Solid
- Simple Solid
- Composite Surface
- Simple Surface
- Line
- Point
Operations on 3D Data

- Spatial indexing in 3D
  - 3D Rtrees
- Spatial queries
  - SDO_FILTER
  - SDO_ANYINTERACT
  - SDO_WITHIN_DISTANCE
  - SDO_NN
- Measurements
  - Distance
  - Length
  - Area
  - Volume
- Generate 3D shapes
  - Convert 2D polygons to 3D solids (by extrusion)
- Publish
  - Extract 3D objects to KML and GML
- Load
  - Load 3D objects from GML
- Process
  - Extract components of a 3D geometry.
  - Solids, composite surfaces in each solid, simple surfaces, etc
- Validate
  - Verify the validity of a 3D object
Point Clouds Storage Model

PC table

<table>
<thead>
<tr>
<th>pc 1</th>
<th>pc 2</th>
<th>pc 3</th>
<th>pc 4</th>
<th>pc 5</th>
<th>pc 6</th>
</tr>
</thead>
</table>

PC blocks table 1

<table>
<thead>
<tr>
<th>pc 1 blocks</th>
<th>pc 2 blocks</th>
<th>pc 3 blocks</th>
</tr>
</thead>
</table>

PC blocks table 2

<table>
<thead>
<tr>
<th>pc 4 blocks</th>
<th>pc 5 blocks</th>
<th>pc 6 blocks</th>
</tr>
</thead>
</table>
Loading a TIN

• Generate the TIN from the point table

```
DECLARE
    TIN SDO_TIN;
BEGIN
    SELECT TIN INTO TIN
    FROM TINS WHERE ID = 1;
    SDO_TIN_PKG.CREATE_TIN (TIN, 'INPUT_POINTS');
END;
/
```

• Structure of the input table

<table>
<thead>
<tr>
<th>RID</th>
<th>VARCHAR2 (40)</th>
<th>Unique point identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAL_D1</td>
<td>NUMBER</td>
<td>Point coordinates</td>
</tr>
<tr>
<td>VAL_D2</td>
<td>NUMBER</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAL_Dn</td>
<td>NUMBER</td>
<td></td>
</tr>
</tbody>
</table>
• What you need to know:
  – Describe key tips on spatial indexing
  – Optimize your spatial queries
  – Explain best practices of the usage of spatial operators
  – Explain best practices of the usage of spatial functions
  – Describe how to best define your spatial data model
  – Describe best practices for spatial metadata, tolerance and coordinate systems usage
  – Explain best practices for spatial data loading and validation (…)

Performance and Tuning / Exadata
Topic 13
What you need to know (continued):

- Describe key spatial application considerations
- Explain how to optimize Spatial on Exadata
- Explain how Oracle Spatial is architected to exploit the processing power, bandwidth and parallelism of Exadata
- Explain how Oracle Spatial on Exadata can ingest massive amounts of data from sensors, weather data, satellites and other streams
Spatial Index Creation Options

WORK_TABLESPACE

- During spatial index creation, the process creates intermediate tables that get dropped when the index is complete.
- The intermediate tables can take up to 2 times the size of the final index. If WORK_TABLESPACE is not specified, the intermediate tables are created in the same tablespace as the final index, causing fragmentation, and possible performance degradation.
- You can use SDO_TUNE.ESTIMATE_RTREE_INDEX_SIZE, and multiply the result by 2 to provide guidance on sizing the work tablespace. The work tablespace can be re-used to create other spatial indexes.
Spatial Index Creation Options

**LAYERTYPE**

- This parameter is needed especially when working with point-only layers. If a point-only layer stores its points in the SDO_ORDINATE_ARRAY, you can still specify `LAYERTYPE=POINT` on spatial index creation.
- This can help query performance when performing spatial analysis.
Spatial Index Creation Options
SDO_NON_LEAF_TBL

- This parameter is useful for very large spatial indexes.
- Splits the index in two tables instead of one.
- The smaller spatial index table is the **non-leaf table**, which is traversed most often during spatial analysis.
- It can be beneficial to pin the non-leaf table into the buffer pool, since it is accessed most often.
Spatial Index Creation Parameters
SDO_NON_LEAF_TBL

• Create the index, specifying a “non-leaf” table

```
CREATE INDEX geod_counties_sidx ON geod_counties(geom)
INDEXTYPE IS MDSYS.SPATIAL_INDEX
PARAMETERS ('sdo_non_leaf_tbl=TRUE');
```

• Find the name of the “non-leaf” table

```
SELECT sdo_nl_index_table
FROM user_sdo_index_metadata
WHERE sdo_index_name='GEOD_COUNTIES_SIDX';
```

• “Pin” it in memory

```
ALTER TABLE MDNT_A930$ STORAGE(BUFFER_POOL KEEP)
```
Workspace Manager

Topic 14

• What you need to know:
  – Explain Workspace Manager concepts, features, and architecture
  – Describe the concepts of versioning on a table and create, merge, refresh tables
  – Describe the concepts of administering privileges and locking, and detecting and resolving conflicts
  – Describe some 3rd party GIS tools that use Workspace Manager
Workspace Manager

- Enables Web and application-based collaboration on database-backed projects
- Provides shareable workspaces within the database to version data
- Example application: managing parcels in a housing development project
Multi-Version a Table

```sql
DBMS_WM.EnableVersioning('PARCELS');
```

- PARCELS: base table
- PARCELS_LT: Renamed base table with four new columns
- PARCELS: View with “instead of” trigger

UPDATE parcels SET ...
Section 2:

Test Preparation Strategies
Exam Structure

Preparation Strategies

• Exam duration: 120 minutes
• Number of questions: 76
• Question Structure: Multiple Choice
• Multiple Choice: 1-N Answers (Given # Correct)
• Passing score: 62%
For a spatial table, what two things are stored in USER_SDO_GEOM_METADATA?

- A. The name of the table’s spatial index
- B. The table’s name
- C. WKTEXT of the spatial extent of data stored in the table
- D. Spatial geometry data
- E. The table’s Spatial Reference ID (SRID)
Approaching the Exam
Preparation Strategies

• Read the Book
  – Work through Examples

• Read the Docs
  – 11.2 Documentation
  – Complex Queries in Appendix
More Resources
Resources
Oracle Technology Network

- www.oracle.com/technetwork/database/options/spatialandgraph
- www.oracle.com/technetwork/middleware/mapviewer
- blogs.oracle.com ➔ oraclespatial ➔ oracle_maps_blog
Oracle Spatial & Graph Special Interest Group
Connect and exchange knowledge with the community of Spatial & Graph users

• Talk with the Board this week
  – Wednesday lunch – SIG Board presentation (150AB)
  – Stop by the SIG User Group roundtable at Meet the Experts, 4:30pm Wednesday in 150AB
  – Visit Oracle’s exhibitor table at breaks & sign up

• Join us
  – Online communities: LinkedIn, Google+, IOUG SIG (free membership)
  – Visit OTN Spatial Community page www.oracle.com/technetwork/database/options/spatialandgraph/community (or search online for “Oracle Spatial and Graph Community”)
  – Email oraclespatialsig@gmail.com
Spatial Certification & Partner Specialization
Get valuable credentials – differentiate your skills

• **Learn more at the Summit**
  – Wed, Track C 3:30 – Exam preparation session
  – Talk to us at Oracle’s exhibitor table & “Meet the Experts” Certification table (Wed 4:30-5:00)

• **Take the next steps**
  – Schedule an exam, access topic lists / online training, learn about Partner Specialization requirements
  – Online training materials for Certified Implementation Specialist exam
MapViewer in Action

Oracle eLocation Services

http://maps.oracle.com/elocation
Courses and Other Resources
Preparation Strategies

• Oracle courses
  – Oracle Spatial: Essentials
  – Oracle Spatial: Advanced

• Online resources
  – Spatial Forum
  – Spatial SIG Site

• Talk to us during the “meet the experts” session up next
Q&A