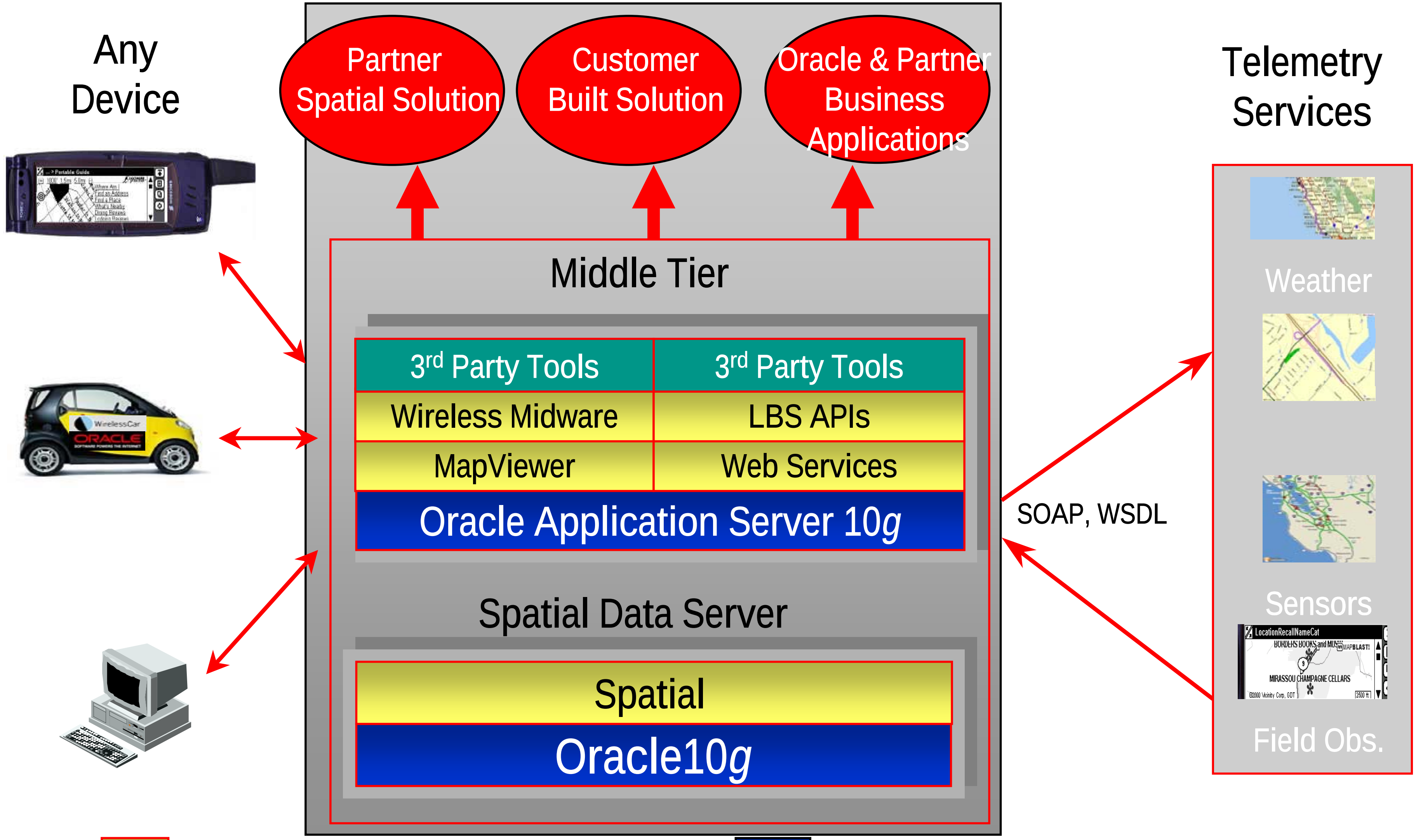
Getting Started with Oracle Spatial

Tim Armitage

Agenda

- Create database structures
- Load Spatial Data
- Index
- Issue SQL queries
- Develop simple Oracle Application Server Mapviewer application

Oracle Spatial 10*g* Platform



Any Device

Partner Spatial Solution

Customer Built Solution

Oracle & Partner Business Applications

Middle Tier

3rd Party Tools | 3rd Party Tools

Wireless Midware | LBS APIs

MapViewer | Web Services

Oracle Application Server 10*g*

Spatial Data Server

Spatial

Oracle10*g*

SOAP, WSDL

Telemetry Services

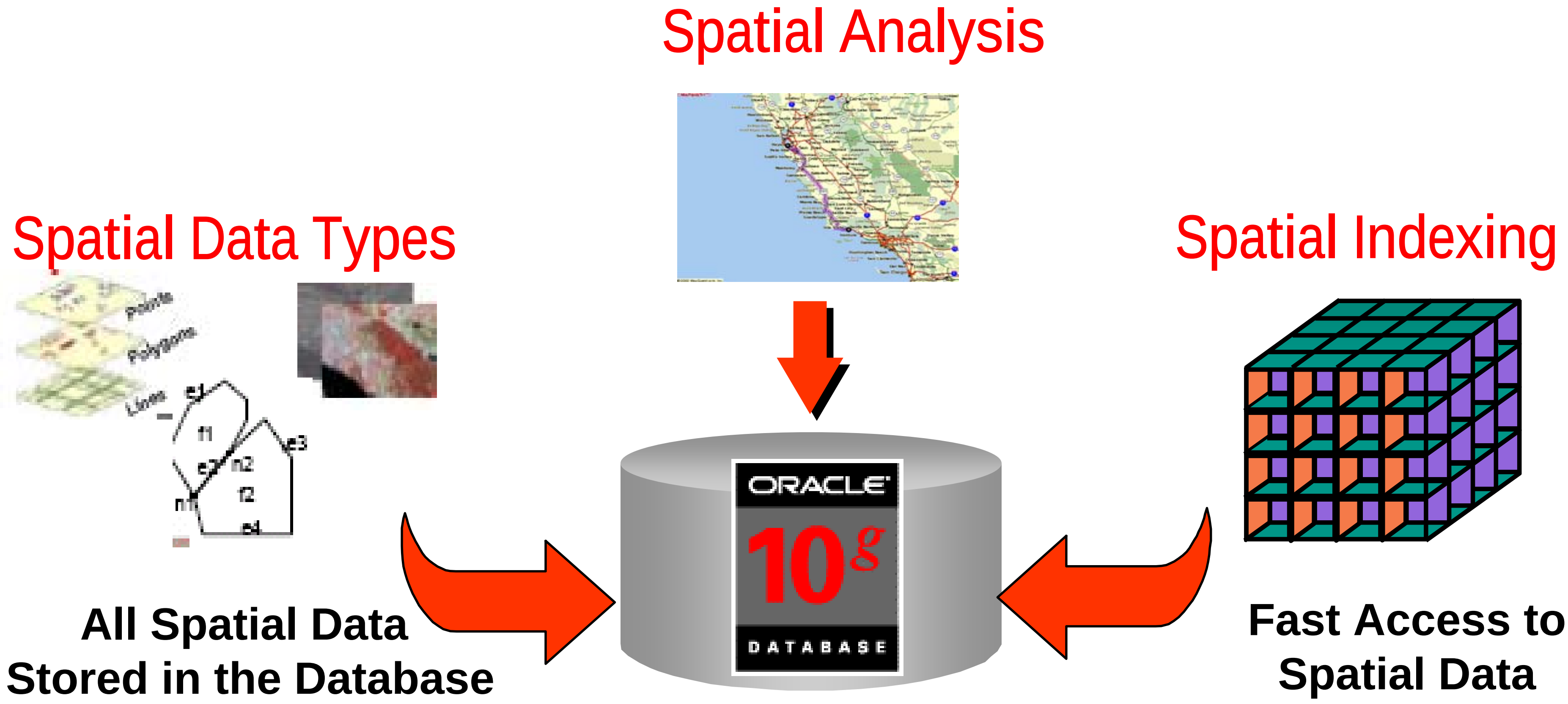
Weather

Sensors

Field Obs.

🟨 Oracle Location Technology 🟦 Oracle Core Technologies

What is a Spatial Database?



Spatial Analysis

Spatial Data Types

Spatial Indexing

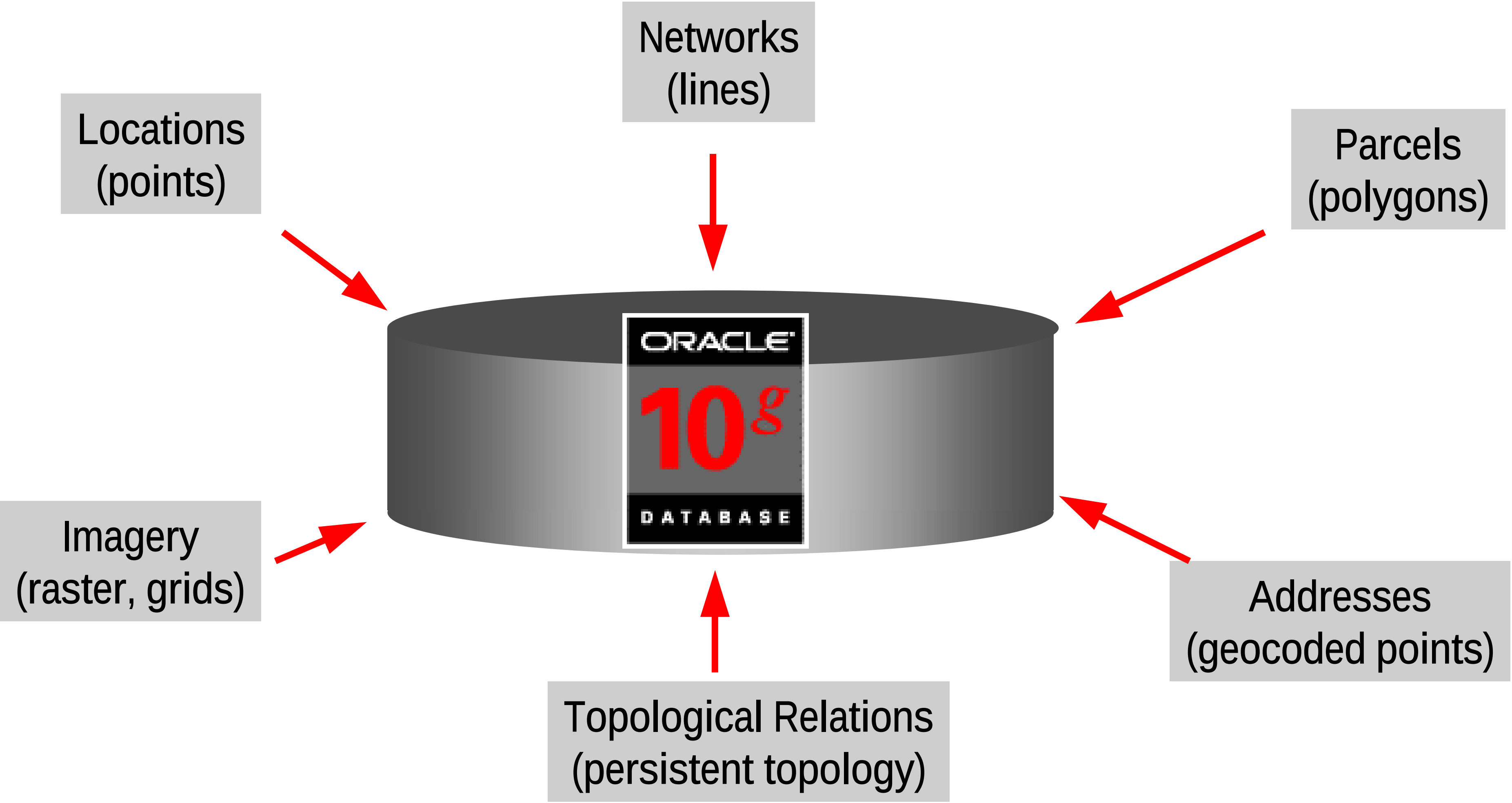
All Spatial Data Stored in the Database

Fast Access to Spatial Data

Spatial Access Through SQL

Create Required Database Structures

All Spatial Types in Oracle10*g*



Locations (points)

Networks (lines)

Parcels (polygons)

Imagery (raster, grids)

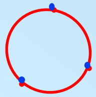
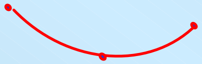
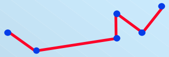
Topological Relations (persistent topology)

Addresses (geocoded points)

Vector Map Data in Oracle Tables

FILBERT ST
BROADWAY ST
PACIFIC AVE
SANSOME ST
BATTERY ST
FRONT ST
THE EMBARCADERO
KEARNY ST
CLAY ST
MONTGOMERY ST
PINE ST
BUSH ST
SUTTER ST
POST ST
GEARY ST
OFARRELL ST
ELLIS ST
MASON ST
BEALE ST
MAIN ST
SPEAR ST
1ST ST
2ND ST
3RD ST
FOLSOM ST
HARRISON ST
80
ORACLE®

Road

ROAD_ID	NAME	SURFACE	LANES	LOCATION
1	Pine Cir.	Asphalt	4	
2	2nd St.	Asphalt	2	
3	3rd St.	Asphalt	2	

The MDSYS Schema

- When Oracle Locator or Spatial is installed, the MDSYS user is created
  - Owner of Spatial types, packages, functions, procedures, metadata
  - Similar to user SYS
  - Privileged user
    - With ADMIN option
- This account is locked by default
  - Be careful with this administrative account
  - You should never need to log in as MDSYS
  - Never create any data as user MDSYS

SDO_GEOMETRY Object

- **SDO_GEOMETRY** Object

```
SDO_GTYPE           NUMBER
SDO_SRID            NUMBER
SDO_POINT           SDO_POINT_TYPE
SDO_ELEM_INFO       SDO_ELEM_INFO_ARRAY
SDO_ORDINATES       SDO_ORDINATE_ARRAY
```

- Example

```
SQL> CREATE TABLE states (
  2     state        VARCHAR2(30),
  3     totpop       NUMBER(9),
  4     geom         SDO_GEOMETRY);
```

SDO_GEOMETRY Object

- SDO_POINT_TYPE

```
x         NUMBER
y         NUMBER
z         NUMBER
```

- SDO_ELEM_INFO_ARRAY

```
VARRAY (1048576) OF NUMBER
```

- SDO_ORDINATE_ARRAY

```
VARRAY (1048576) OF NUMBER
```

SDO_GEOMETRY Object

- **SDO_GTYPE -** Defines the type of geometry stored in the object

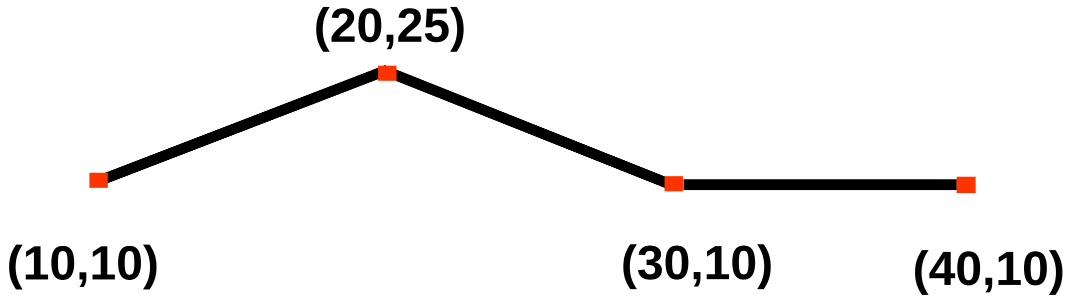
GTYPE	Explanation
1 POINT	Geometry contains one point
2 LINESTRING	Geometry contains one line string
3 POLYGON	Geometry contains one polygon
4 HETEROGENEOUS COLLECTION	Geometry is a collection of elements of different types: points, lines, polygons
5 MULTIPOINT	Geometry has multiple points
6 MULTILINESTRING	Geometry has multiple line strings
7 MULTIPOLYGON	Geometry has multiple polygons

SDO_GTYPE

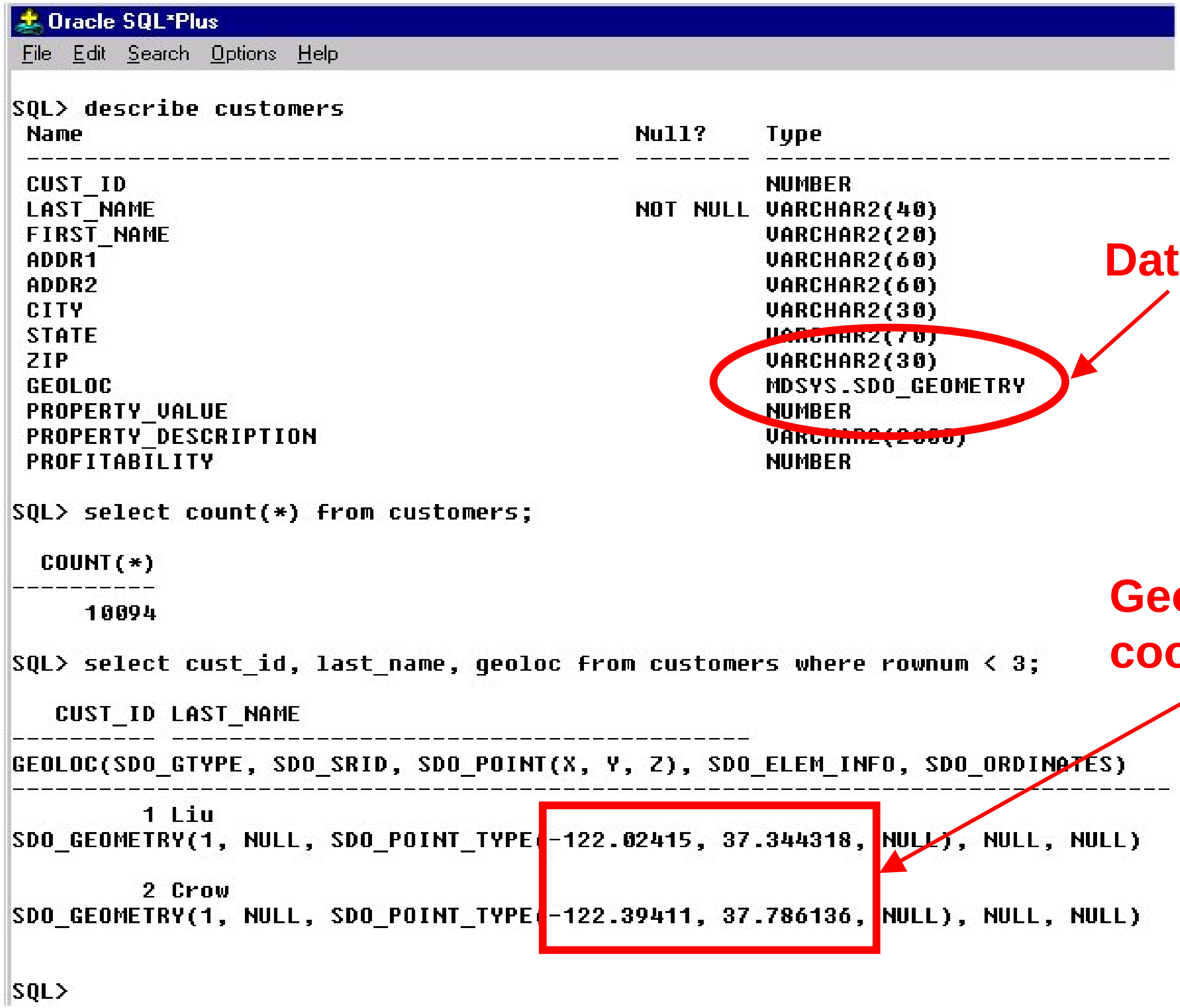
SDO_GTYPE	Four digit GTYPEs - Include dimensionality		
	2D	3D	4D
1 POINT	2001	3001	4001
2 LINESTRING	2002	3002	4002
3 POLYGON	2003	3003	4003
4 COLLECTION	2004	3004	4004
5 MULTIPOINT	2005	3005	4005
6 MULTILINESTRING	2006	3006	4006
7 MULTIPOLYGON	2007	3007	4007

Constructing Geometries

```
SQL> INSERT INTO LINES VALUES (
  2>       attribute_1, …. attribute_n,
  3>       SDO_GEOMETRY (
  4>         2002, null, null,
  5>         SDO_ELEM_INFO_ARRAY (1,2,1),
  6>         SDO_ORDINATE_ARRAY (
  7>           10,10, 20,25, 30,10, 40,10))
  8>       );
```



How Spatial Data Is Stored



```
SQL> describe customers
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUST_ID                                            NUMBER
 LAST_NAME                                 NOT NULL VARCHAR2(40)
 FIRST_NAME                                         VARCHAR2(20)
 ADDR1                                              VARCHAR2(60)
 ADDR2                                              VARCHAR2(60)
 CITY                                               VARCHAR2(30)
 STATE                                              VARCHAR2(70)
 ZIP                                                VARCHAR2(30)
 GEOLOC                                             MDSYS.SDO_GEOMETRY
 PROPERTY_VALUE                                     NUMBER
 PROPERTY_DESCRIPTION                               VARCHAR2(2000)
 PROFITABILITY                                      NUMBER

SQL> select count(*) from customers;

  COUNT(*)
----------
     10094

SQL> select cust_id, last_name, geoloc from customers where rownum < 3;

   CUST_ID LAST_NAME
---------- ----------------------------------------
GEOLOC(SDO_GTYPE, SDO_SRID, SDO_POINT(X, Y, Z), SDO_ELEM_INFO, SDO_ORDINATES)
--------------------------------------------------------------------------------
         1 Liu
SDO_GEOMETRY(1, NULL, SDO_POINT_TYPE(-122.02415, 37.344318, NULL), NULL, NULL)

         2 Crow
SDO_GEOMETRY(1, NULL, SDO_POINT_TYPE(-122.39411, 37.786136, NULL), NULL, NULL)

SQL>
```

Data type

Geographic coordinates

Spatial Metadata

- The spatial routines require you to populate a view that contains metadata about **SDO_GEOMETRY** columns
- The metadata view is created for all Oracle Spatial users when Oracle Spatial is installed
- The metadata view is called **USER_SDO_GEOM_METADATA**
- For every **SDO_GEOMETRY** column, insert a row in the **USER_SDO_GEOM_METADATA** view

USER_SDO_GEOM_METADATA

```
SQL> DESCRIBE USER_SDO_GEOM_METADATA
 Name             Null?       Type
 ---------------- ----------- ---------------------
 TABLE_NAME       NOT NULL    VARCHAR2(32)
 COLUMN_NAME      NOT NULL    VARCHAR2(1024)
 DIMINFO                      SDO_DIM_ARRAY
 SRID                         NUMBER
```

- **MDSYS.SDO_DIM_ARRAY**

```
VARRAY(4) OF SDO_DIM_ELEMENT
```

- **MDSYS.SDO_DIM_ELEMENT** object

```
 SDO_DIMNAME     VARCHAR2(64)
SDO_LB          NUMBER
SDO_UB          NUMBER
SDO_TOLERANCE   NUMBER
```

Populating the USER_SDO_GEOM_METADATA View

```
SQL> INSERT INTO USER_SDO_GEOM_METADATA
  2>  (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
  3>  VALUES (
  4>   'ROADS',
  5>   'GEOMETRY',
  6>    SDO_DIM_ARRAY (
  7>       SDO_DIM_ELEMENT('Long', -180, 180, 0.5),
  8>       SDO_DIM_ELEMENT('Lat',   -90,  90, 0.5)),
  9>    8307);
```

Note: For geodetic data, the x axis bounds must be –180 to 180, and y axis bounds –90 to 90.

Load Spatial Data into Oracle Spatial Database

Loading Spatial Data

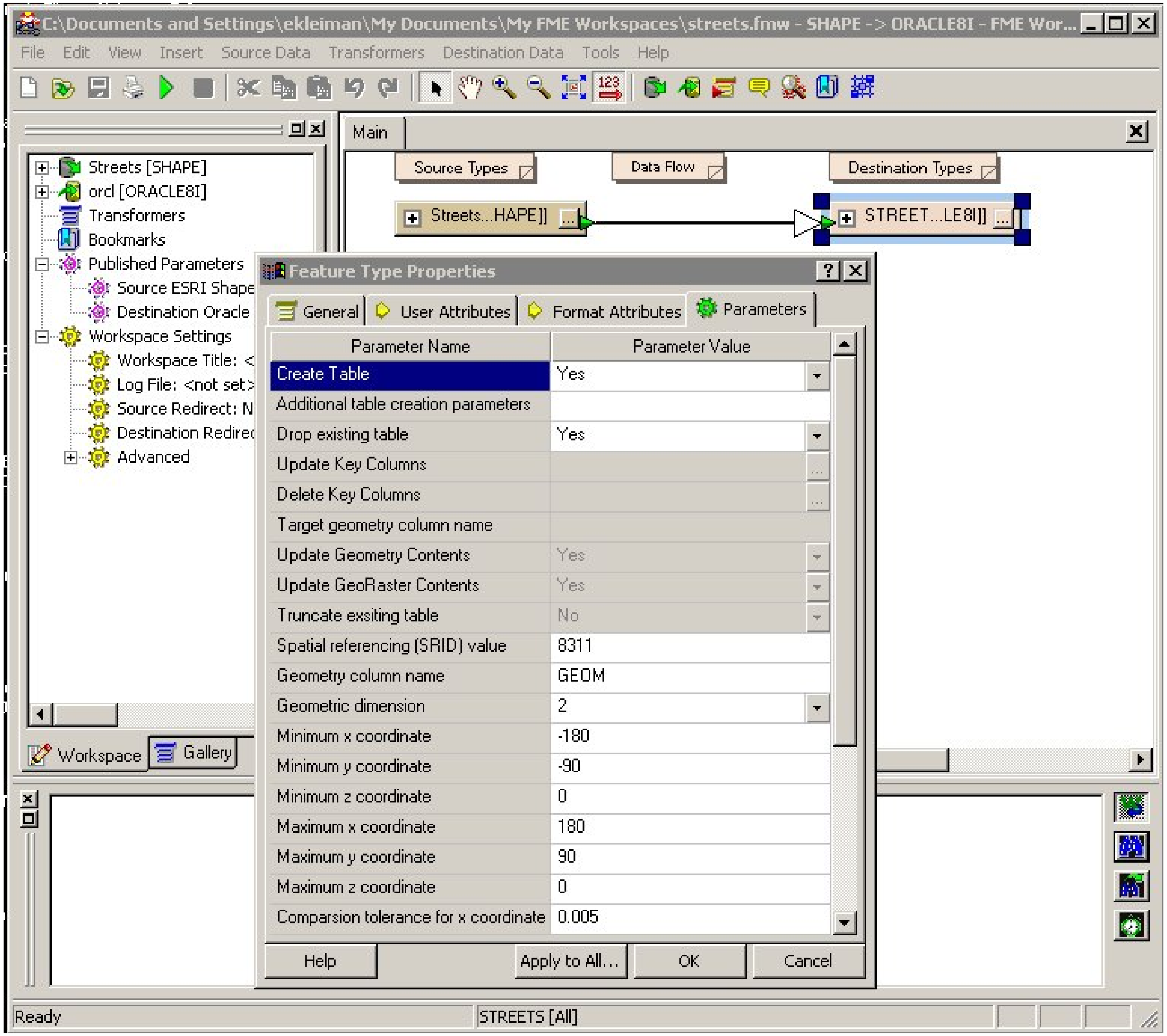
- Categories of loading:
  - Bulk loading of data
    - SQL*Loader
    - Import
  - Transactional inserts
    - INSERT statement
  - Loading using Partner Tools
    - Example – SAFE Software’s FME

Validating Geometries

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

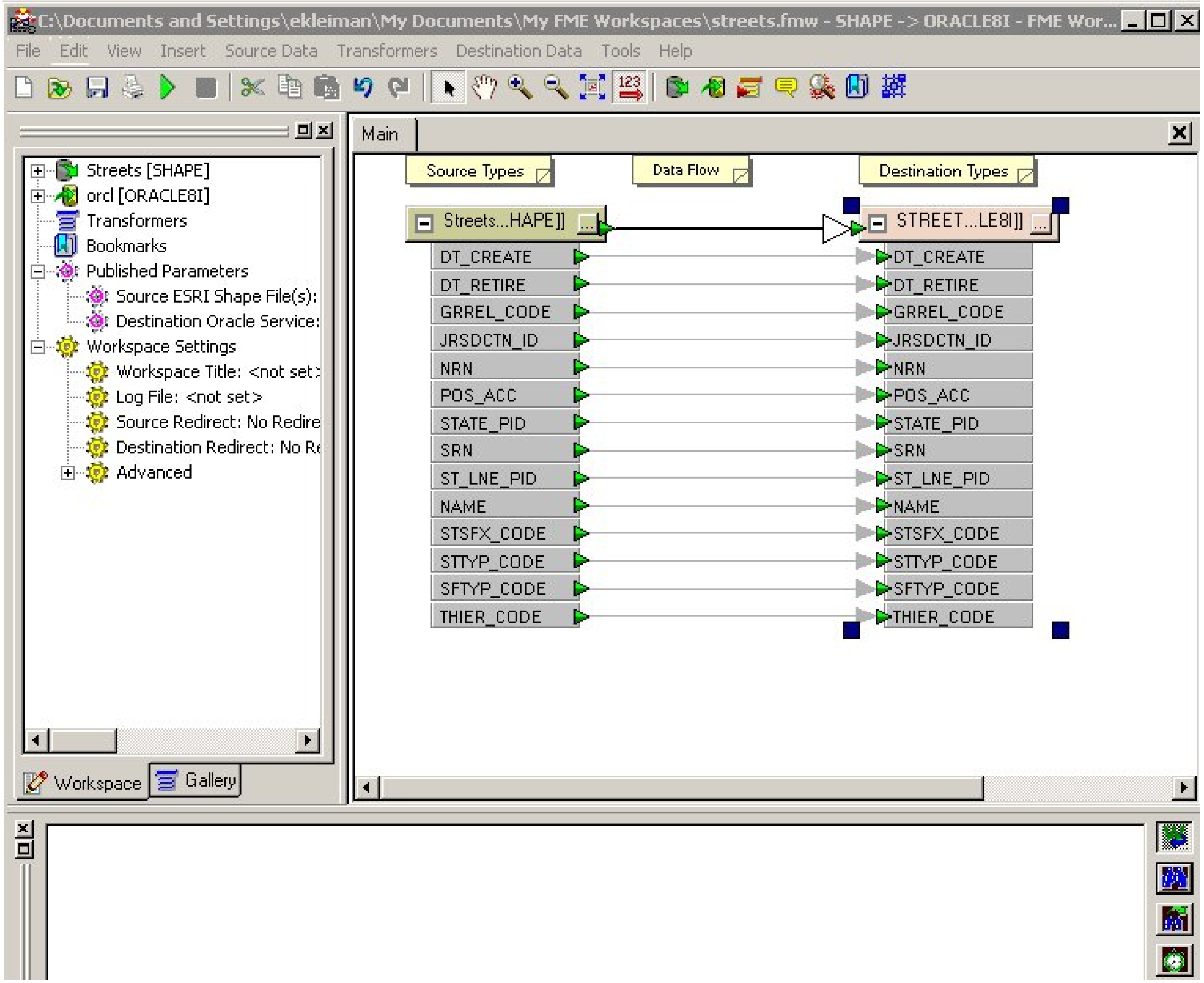
DEMO
Loading Data using FME

FME Workbench



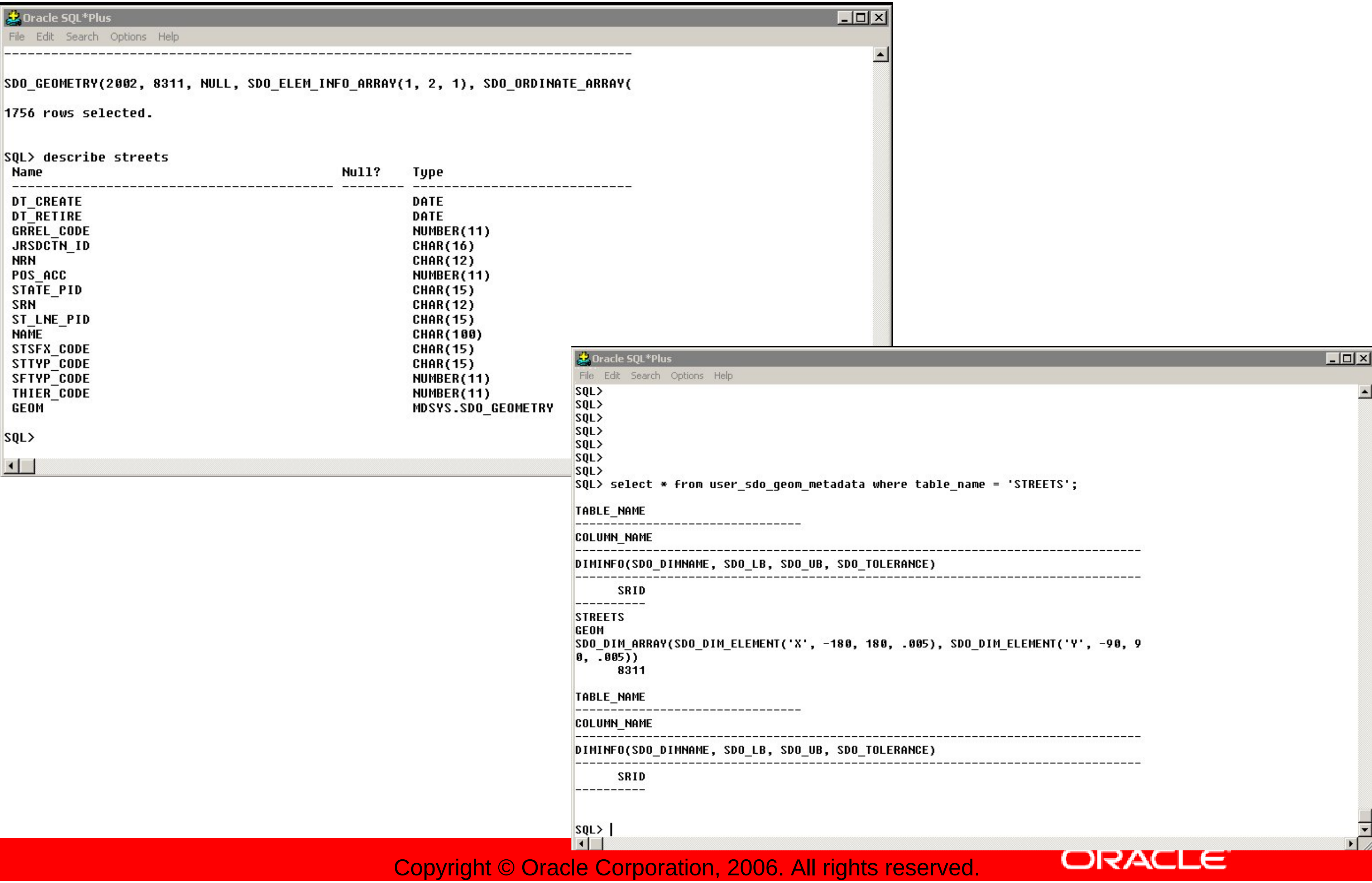
Parameter Name	Parameter Value
Create Table	Yes
Additional table creation parameters	
Drop existing table	Yes
Update Key Columns	
Delete Key Columns	
Target geometry column name	
Update Geometry Contents	Yes
Update GeoRaster Contents	Yes
Truncate exsiting table	No
Spatial referencing (SRID) value	8311
Geometry column name	GEOM
Geometric dimension	2
Minimum x coordinate	-180
Minimum y coordinate	-90
Minimum z coordinate	0
Maximum x coordinate	180
Maximum y coordinate	90
Maximum z coordinate	0
Comparsion tolerance for x coordinate	0.005

FME Mapping



C:\Documents and Settings\ekleiman\My Documents\My FME Workspaces\streets.fmw - SHAPE -> ORACLE8I - FME Wor...

Oracle Structures



```
--------------------------------------------------------------------------------

SDO_GEOMETRY(2002, 8311, NULL, SDO_ELEM_INFO_ARRAY(1, 2, 1), SDO_ORDINATE_ARRAY(

1756 rows selected.


SQL> describe streets
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DT_CREATE                                          DATE
 DT_RETIRE                                          DATE
 GRREL_CODE                                         NUMBER(11)
 JRSDCTN_ID                                         CHAR(16)
 NRN                                                CHAR(12)
 POS_ACC                                            NUMBER(11)
 STATE_PID                                          CHAR(15)
 SRN                                                CHAR(12)
 ST_LNE_PID                                         CHAR(15)
 NAME                                               CHAR(100)
 STSFX_CODE                                         CHAR(15)
 STTYP_CODE                                         CHAR(15)
 SFTYP_CODE                                         NUMBER(11)
 THIER_CODE                                         NUMBER(11)
 GEOM                                               MDSYS.SDO_GEOMETRY

SQL>
```

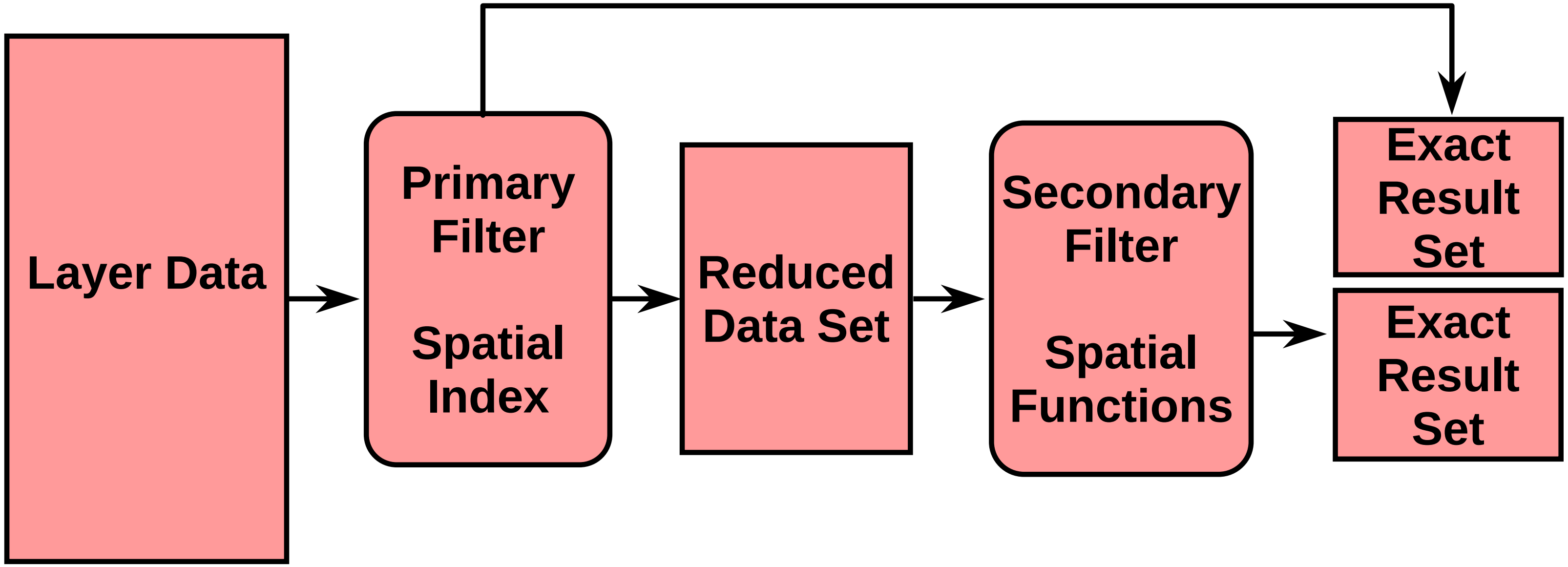
```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> select * from user_sdo_geom_metadata where table_name = 'STREETS';

TABLE_NAME
--------------------------------
COLUMN_NAME
--------------------------------------------------------------------------------
DIMINFO(SDO_DIMNAME, SDO_LB, SDO_UB, SDO_TOLERANCE)
--------------------------------------------------------------------------------
      SRID
----------
STREETS
GEOM
SDO_DIM_ARRAY(SDO_DIM_ELEMENT('X', -180, 180, .005), SDO_DIM_ELEMENT('Y', -90, 9
0, .005))
      8311

TABLE_NAME
--------------------------------
COLUMN_NAME
--------------------------------------------------------------------------------
DIMINFO(SDO_DIMNAME, SDO_LB, SDO_UB, SDO_TOLERANCE)
--------------------------------------------------------------------------------
      SRID
----------


SQL> |
```

Set up Spatial Indexes

Spatial Indexing

- Used to optimize spatial query performance
- R-tree Indexing
  - Based on minimum bounding rectangles (MBRs) for 2D data or minimum bounding volumes (MBVs) for 3D data
  - Indexes two, three, or four dimensions
- Provides an exclusive and exhaustive coverage of spatial objects
- Indexes all elements within a geometry including points, lines, and polygons

Optimized Query Model



Layer Data

Primary Filter

Spatial Index

Reduced Data Set

Secondary Filter

Spatial Functions

Exact Result Set

Exact Result Set

Column where coordinates are stored →

Index retrieves area of interest (window) →

Procedures that determine exact relationship

A Look at R-tree Index Structures

```
create index GEOD_STATES_SIDX
   on GEOD_STATES (GEOM)
 indextype is MDSYS.SPATIAL_INDEX;
```

Index Information

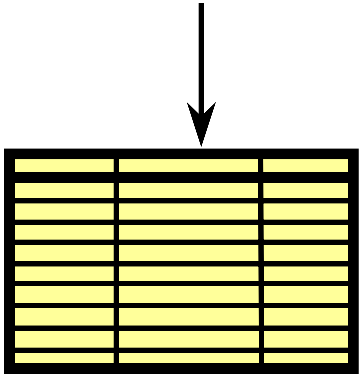
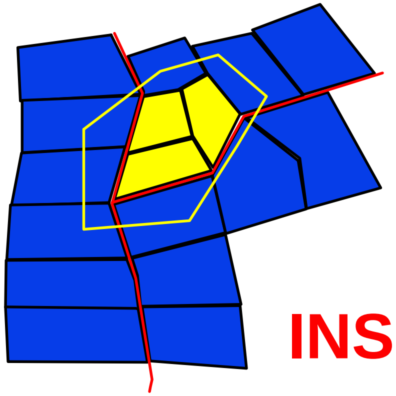


Table MDRT_7B50$

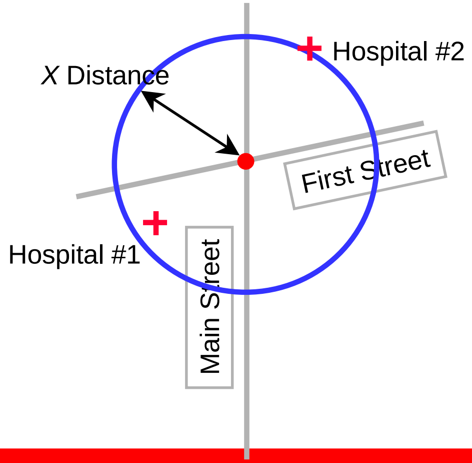
Issue SQL Queries

Spatial Operators

- Full range of spatial operators
  - Implemented as functional extensions in SQL
  - Topological Operators
    - Inside Contains
    - Touch Disjoint
    - Covers Covered By
    - Equal Overlap Boundary
  - Distance Operators
    - Within Distance
    - Nearest Neighbor



INSIDE



Spatial Operators

- Operators
  - `SDO_FILTER`
    - Performs a primary filter only
  - `SDO_RELATE` and `SDO_<relationship>`
    - Performs a primary and secondary filter
  - `SDO_WITHIN_DISTANCE`
    - Generates a buffer around a geometry and performs a primary and optionally a secondary filter
  - `SDO_NN`
    - Returns nearest neighbors

SDO_FILTER Example

- Find all the cities in a selected rectangular area
- Result is approximate

```
SELECT c.city, c.pop90
FROM proj_cities c
WHERE sdo_filter (
        c.location,
        sdo_geometry (2003, 32775, null,
          sdo_elem_info_array (1,1003,3),
          sdo_ordinate_array (1720300,1805461,
                              1831559, 2207250))
        ) = 'TRUE';
```

Hint 1: All Spatial operators return `TRUE` or `FALSE`. When writing spatial queries always test with = `'TRUE'`, never <> `'FALSE'` or = `'true'`.

SDO_RELATE Example

- Find all counties in the state of New Hampshire

```
SELECT c.county, c.state_abrv
FROM geod_counties c,
     geod_states s
WHERE s.state = 'New Hampshire'
 AND sdo_relate (c.geom,
                 s.geom,
                 'mask=INSIDE+COVEREDBY')
                    ='TRUE';
```

Note: For optimal performance, don't forget to index **`GEOD_STATES(state)`**

Relationship Operators Example

- Find all the counties around Passaic county in New Jersey:

```
SELECT /*+ ordered */ a.county
FROM geod_counties b,
     geod_counties a
WHERE b.county = 'Passaic'
    AND b.state = 'New Jersey'
    AND SDO_TOUCH(a.geom,b.geom) = 'TRUE';
```

- Previously:

```
. . .
AND SDO_RELATE(a.geom,b.geom,
     'MASK=TOUCH') = 'TRUE';
```

SDO_WITHIN_DISTANCE Examples

- Find all cities within a distance from an interstate

```
SELECT /*+ ordered */ c.city
FROM geod_interstates i, geod_cities c
WHERE i.highway = 'I170'
   AND sdo_within_distance (
        c.location, i.geom,
       'distance=15 unit=mile') = 'TRUE';
```

- Find interstates within a distance from a city

```
SELECT /*+ ordered */ i.highway
FROM geod_cities c, geod_interstates i
WHERE c.city = 'Tampa'
   AND sdo_within_distance (
        i.geom, c.location,
        'distance=15 unit=mile') = 'TRUE';
```

SDO_NN Example

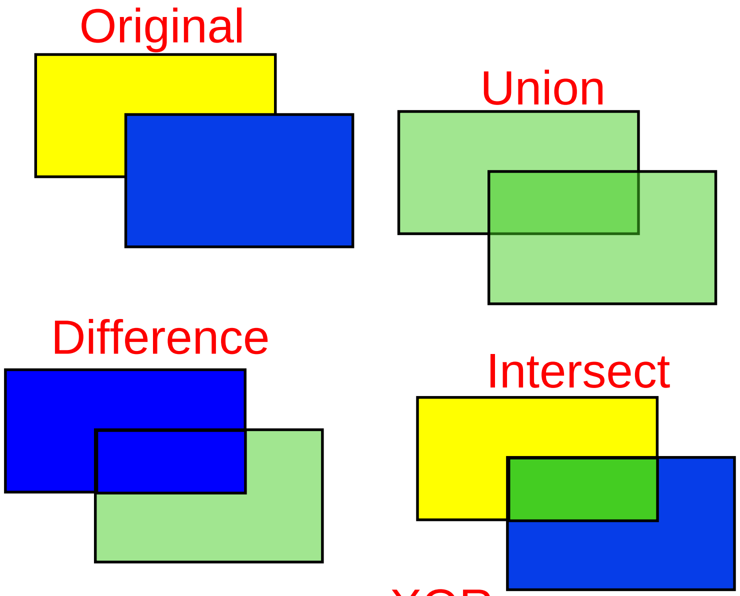
- Find the five cities nearest to Interstate I170, ordered by distance

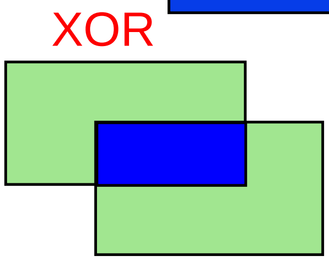
```
SELECT /*+ ordered */
       c.city, c.state_abrv,
       sdo_nn_distance (1) distance_in_miles
FROM geod_interstates i,
     geod_cities c
WHERE i.highway = 'I170'
  AND sdo_nn(c.location, i.geom,
             'sdo_num_res=5 unit=mile', 1) = 'TRUE'
ORDER by distance_in_miles;
```

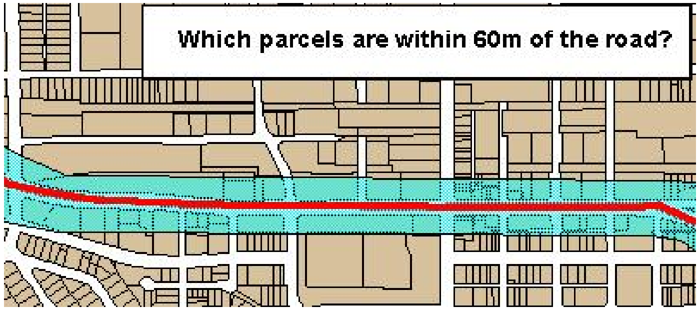
- Note: Make sure you have an index on `GEOD_INTERSTATES (HIGHWAY)`.

Spatial Functions

- Returns a geometry
  - Union
  - Difference
  - Intersect
  - XOR
  - Buffer
  - CenterPoint
  - ConvexHull
- Returns a number
  - LENGTH
  - AREA
  - Distance



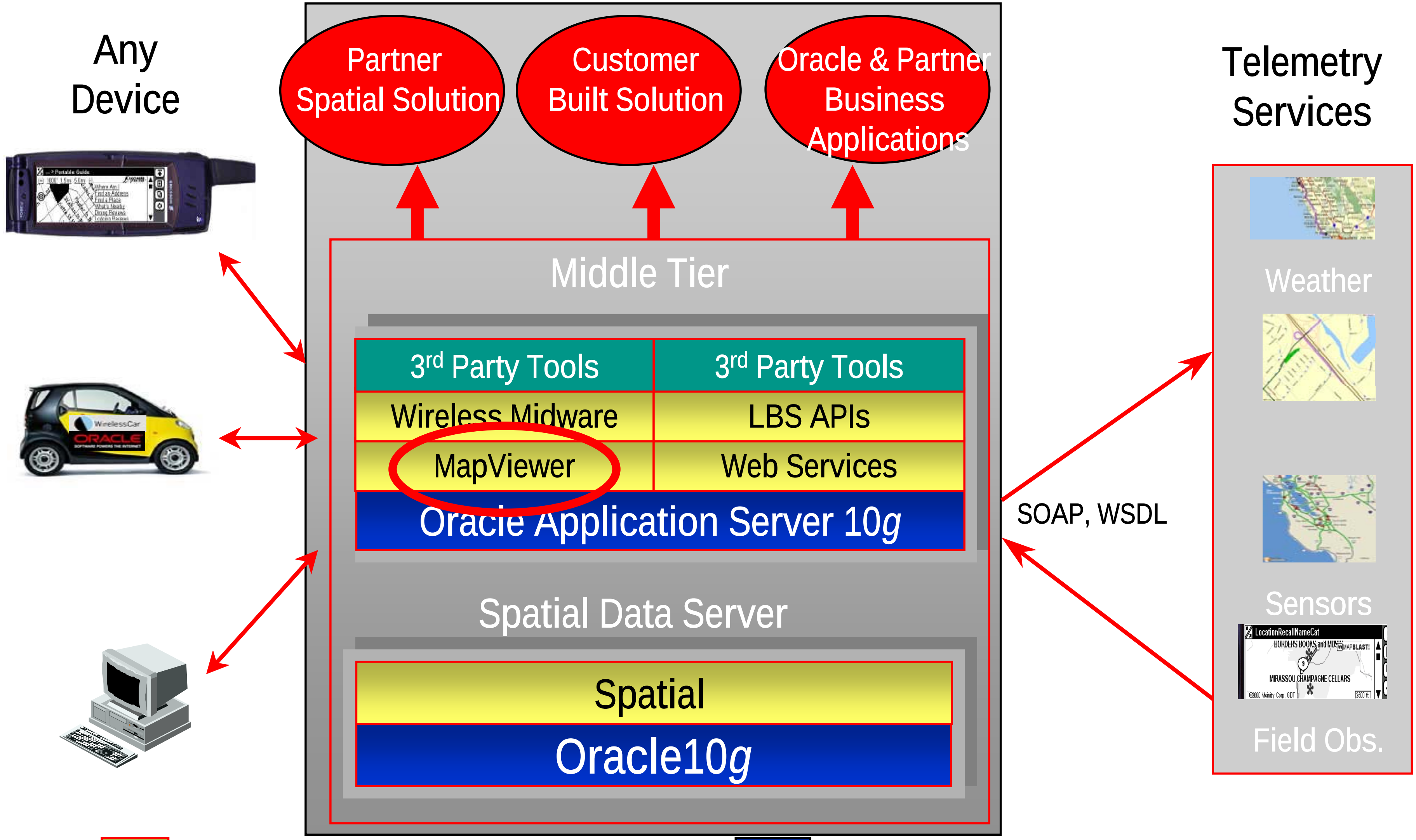




DEMO
SQL Developer

Develop Simple Oracle Application Server MapViewer Application

Oracle Spatial 10*g* Platform



Any Device

Partner Spatial Solution

Customer Built Solution

Oracle & Partner Business Applications

Middle Tier

3rd Party Tools	3rd Party Tools
Wireless Midware	LBS APIs
MapViewer	Web Services

Oracle Application Server 10*g*

Spatial Data Server

Spatial

Oracle10*g*

SOAP, WSDL

Telemetry Services

Weather

Sensors

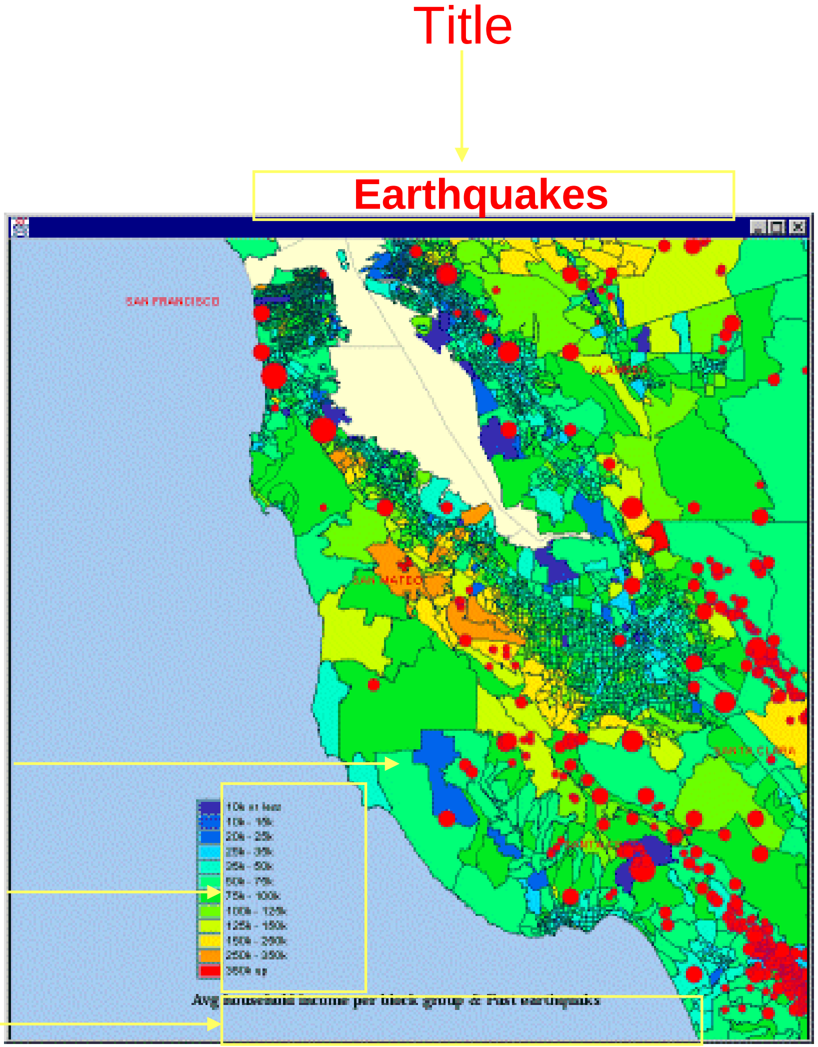
Field Obs.

🟨 Oracle Location Technology

🟦 Oracle Core Technologies

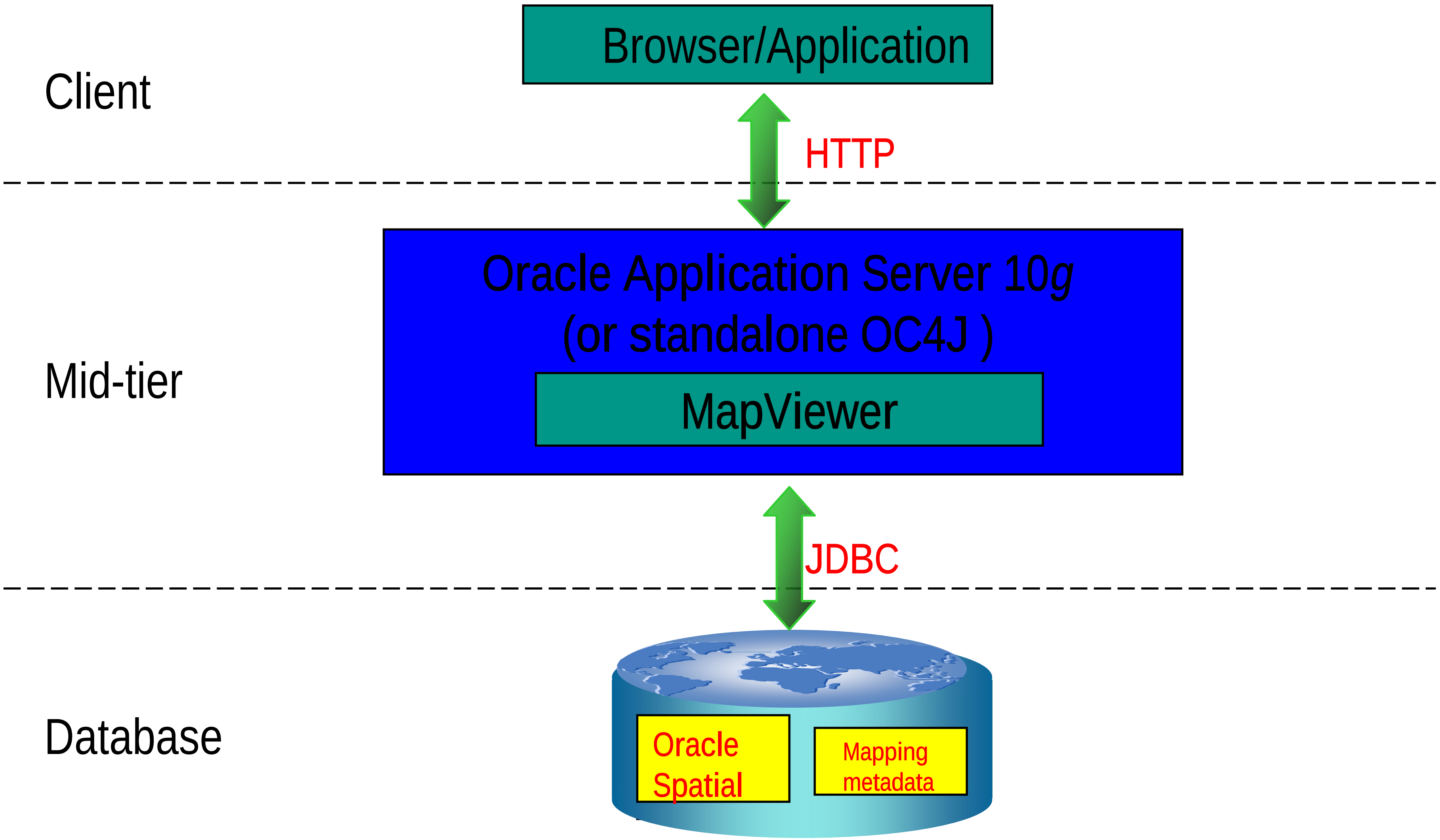
MapViewer Overview

- A map rendering service in Oracle Application Server 10*g*. It is a server component (not a client viewer!)
- It visualizes data managed by Oracle Spatial.
- Provides a comprehensive set of APIs(XML and Java-based), using which client viewers can be easily developed and OGC WMS APIs
- Provides an enterprise-level solution to mapping metadata management.



MapViewer Overview

Architecture



Client

Browser/Application

HTTP

Mid-tier

Oracle Application Server 10*g*
(or standalone OC4J)

MapViewer

JDBC

Database

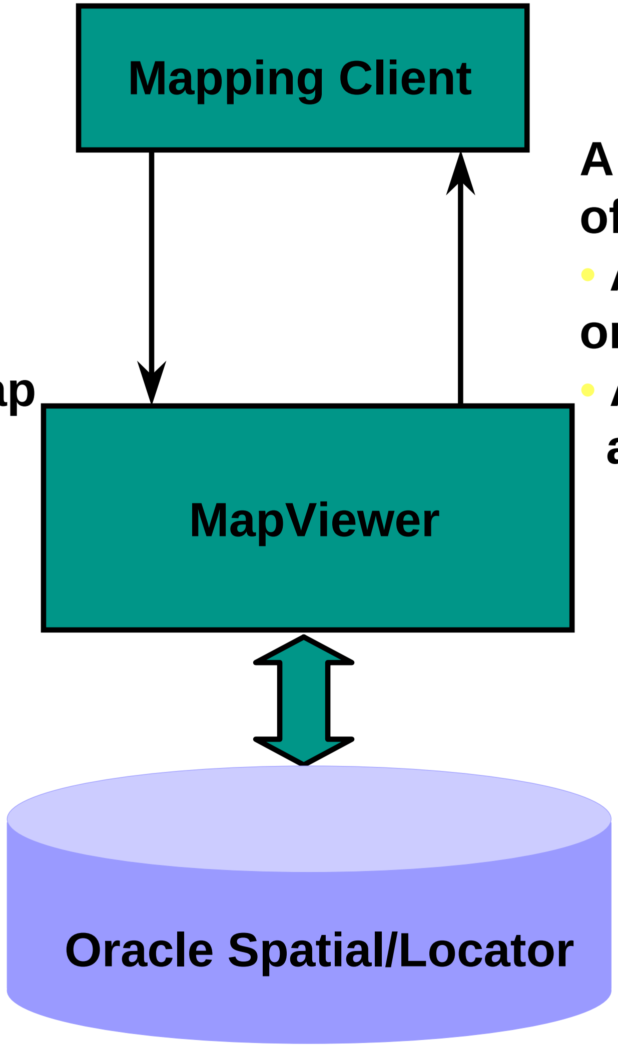
Oracle Spatial

Mapping metadata

MapViewer Query

A map request consists of:

- Base map name
- Center of map
- Width and height of map
- Optional tags
  - map name
  - jdbc_query
  - others



A map response consists of:

- A streamed map image

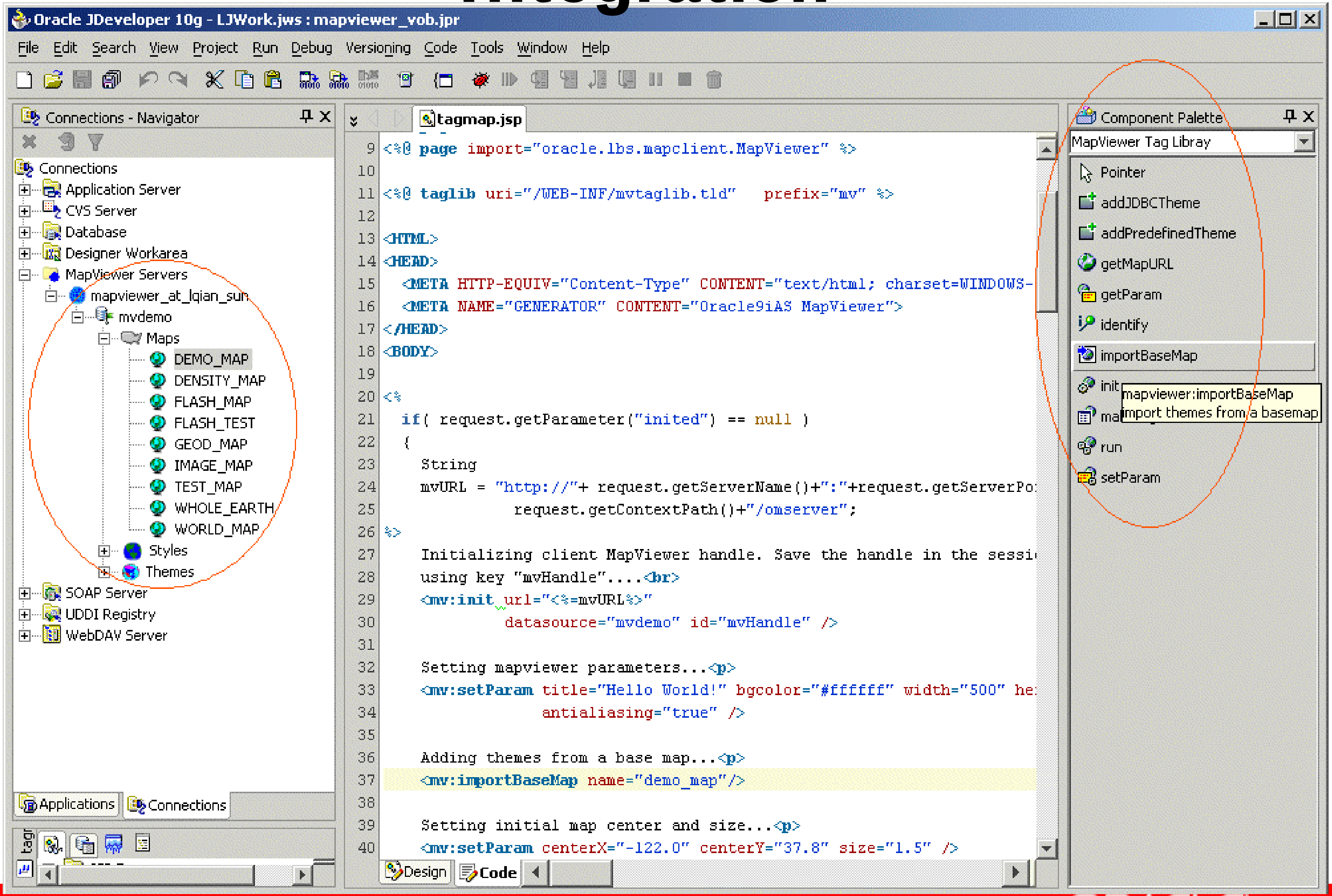
or

- A URL to the map image along with the map MBR

MapViewer APIs

- MapViewer supports 3 API flavors
  - XML-based
    - Native language to MapViewer
  - Java thin library
    - a mapping “bean” (without UI)
  - JSP custom tags
    - a subset of functions
    - To be used as a ‘fast start’ for beginners
    - The JSP taglib can be easily added to Oracle JDeveloper’s component palette
    - A JDeveloper extension that lets you browse the current list of existing maps/themes/styles in a data source

Enhanced APIs and JDeveloper Integration



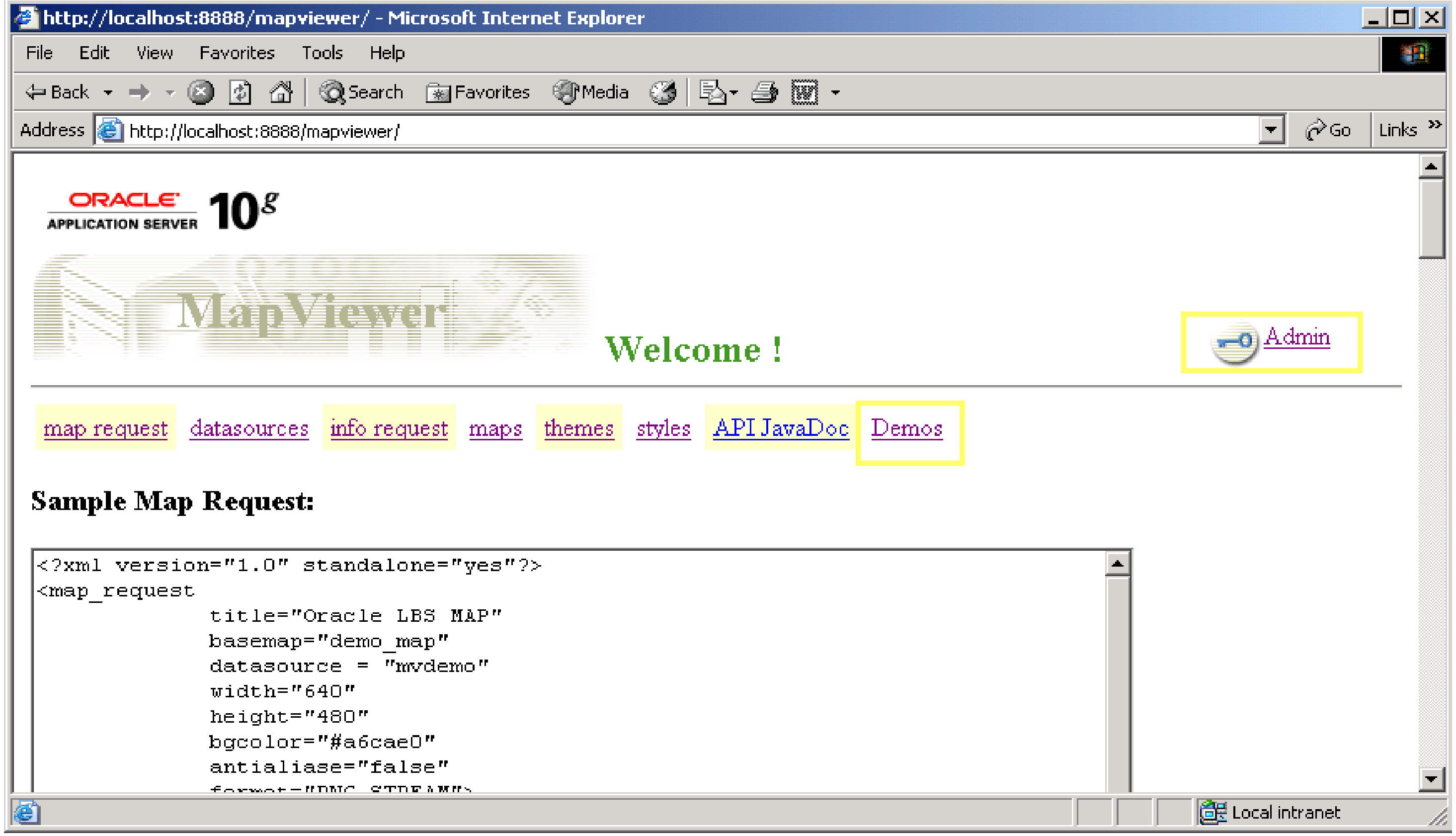
```
<%@ page import="oracle.lbs.mapclient.MapViewer" %>

<%@ taglib uri="/WEB-INF/mvtaglib.tld"   prefix="mv" %>

<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=WINDOWS-
  <META NAME="GENERATOR" CONTENT="Oracle9iAS MapViewer">
</HEAD>
<BODY>

<%
  if( request.getParameter("inited") == null )
  {
    String
    mvURL = "http://"+ request.getServerName()+":"+request.getServerPo
              request.getContextPath()+"/omserver";
%>

    Initializing client MapViewer handle. Save the handle in the sessi
    using key "mvHandle"....<br>
    <mv:init url="<%=mvURL%>"
          datasource="mvdemo" id="mvHandle" />

    Setting mapviewer parameters...<p>
    <mv:setParam title="Hello World!" bgcolor="#ffffff" width="500" he
            antialiasing="true" />

    Adding themes from a base map...<p>
    <mv:importBaseMap name="demo_map"/>

    Setting initial map center and size...<p>
    <mv:setParam centerX="-122.0" centerY="37.8" size="1.5" />
```

MapViewer Key Concepts

- Datasource
- Map
- Basemap
- Theme
- Style

MapViewer Welcome Page
http://localhost:8888/mapviewer

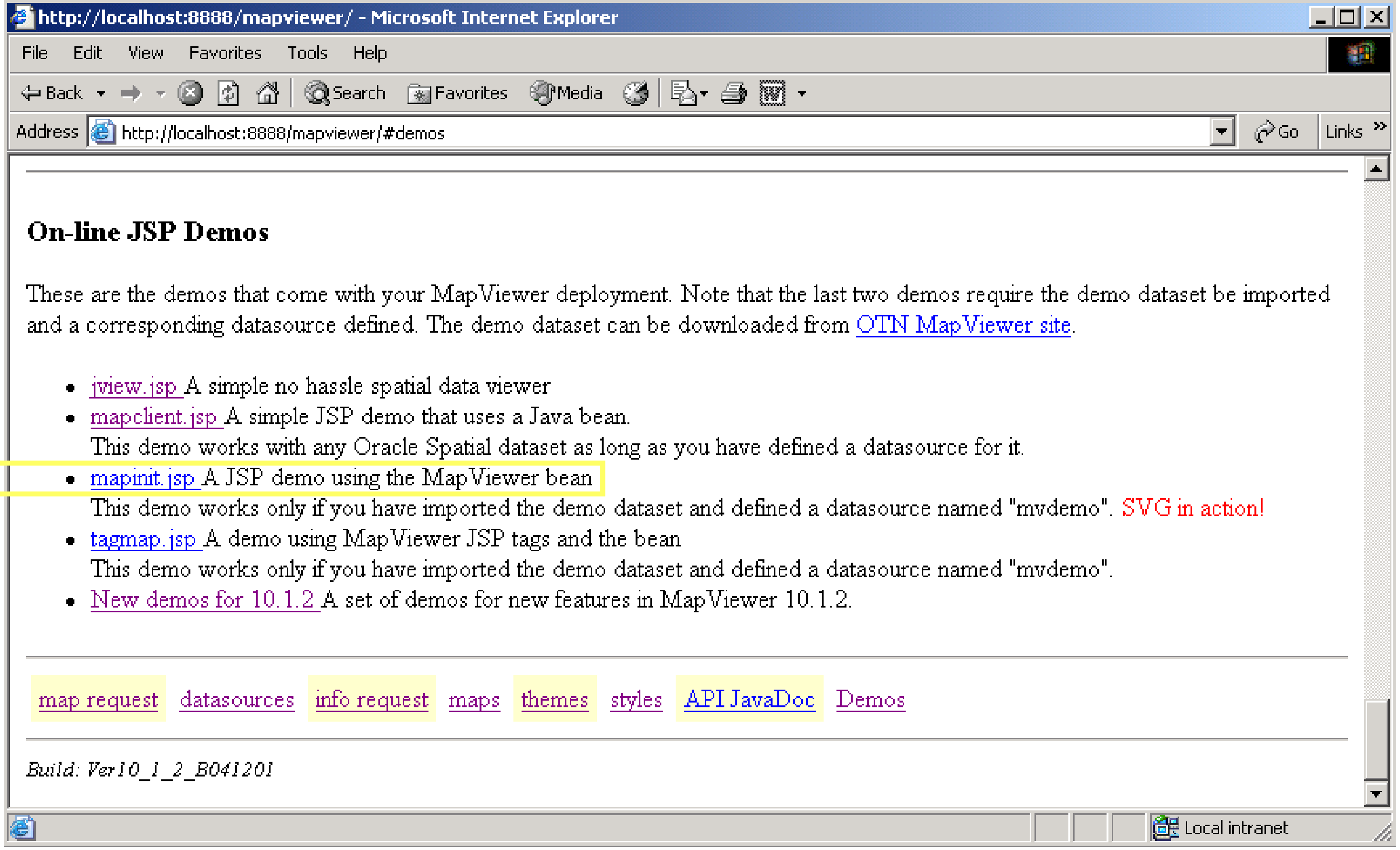
- Icon to go to/from the Admin page (see key icon in upper left)
- Several other hyperlinks, including Demos


http://localhost:8888/mapviewer/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address http://localhost:8888/mapviewer/
Go
Links
ORACLE APPLICATION SERVER 10g
MapViewer
Welcome !
Admin
map request
datasources
info request
maps
themes
styles
API JavaDoc
Demos
Local intranet

Sample Map Request:

```
<?xml version="1.0" standalone="yes"?>
<map_request
             title="Oracle LBS MAP"
             basemap="demo_map"
             datasource = "mvdemo"
             width="640"
             height="480"
             bgcolor="#a6cae0"
             antialiase="false"
             format="PNG_STREAM">
```

MapViewer Welcome Page
Demos



http://localhost:8888/mapviewer/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8888/mapviewer/#demos
Go Links
Local intranet

On-line JSP Demos

These are the demos that come with your MapViewer deployment. Note that the last two demos require the demo dataset be imported and a corresponding datasource defined. The demo dataset can be downloaded from OTN MapViewer site.

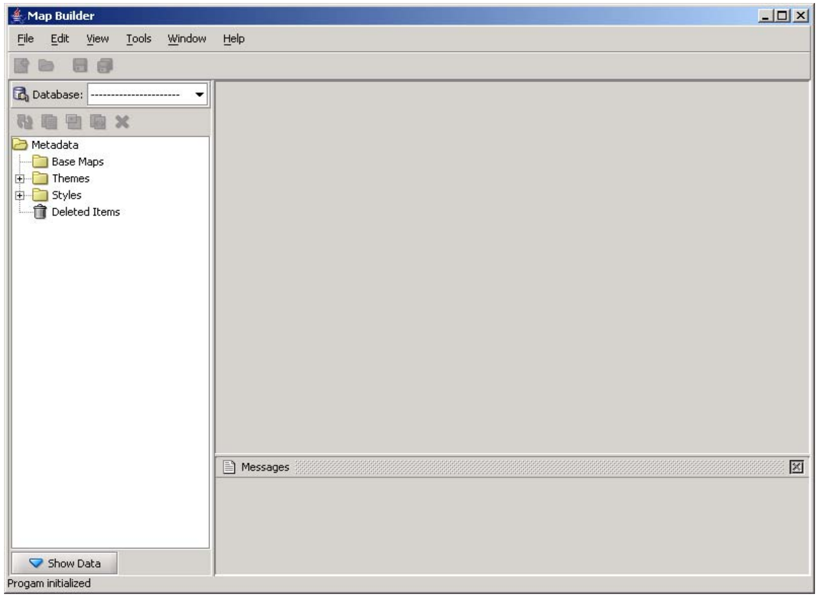
- jview.jsp A simple no hassle spatial data viewer
- mapclient.jsp A simple JSP demo that uses a Java bean.
  This demo works with any Oracle Spatial dataset as long as you have defined a datasource for it.
- mapinit.jsp A JSP demo using the MapViewer bean
  This demo works only if you have imported the demo dataset and defined a datasource named "mvdemo". SVG in action!
- tagmap.jsp A demo using MapViewer JSP tags and the bean
  This demo works only if you have imported the demo dataset and defined a datasource named "mvdemo".
- New demos for 10.1.2 A set of demos for new features in MapViewer 10.1.2.

map request datasources info request maps themes styles API JavaDoc Demos

Build: Ver10_1_2_B041201

Oracle Map Builder

- Replacement for the Map Definition tool
- Currently in Beta and available on OTN for download
  - http://www.oracle.com/technology/software/products/mapviewer



Progam initialized

DEMO
Mapviewer and Mapbuilder

ORACLE IS THE **INFORMATION** COMPANY