Jul 29, 2025

Getting started working with LiDAR data (Part 1)

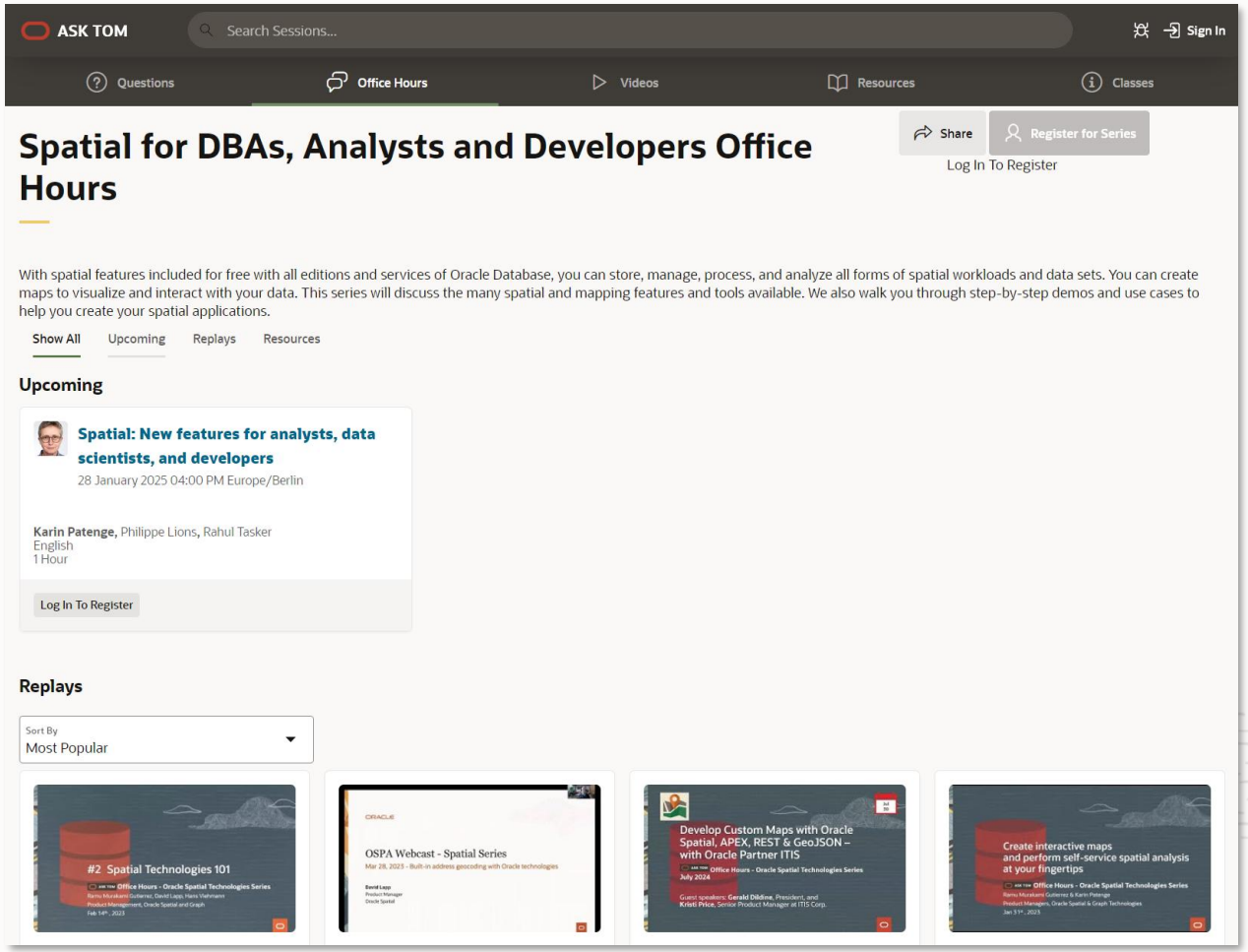
ASK TOM **Office Hours - Oracle Spatial Technologies Series**

Host: Denise Myrick

Speaker: Karin Patenge

Oracle Database Product Management | Spatial and Graph Technologies

Previous sessions of the Spatial Series on ASK TOM



Agenda for today

1. Basics about LiDAR
2. Tools to work with LiDAR data
3. Managing LiDAR data in the Oracle Database
4. A LiDAR data workflow
5. Where to find datasets

Tbc. in the August session

Basics about LiDAR

What is LiDAR?

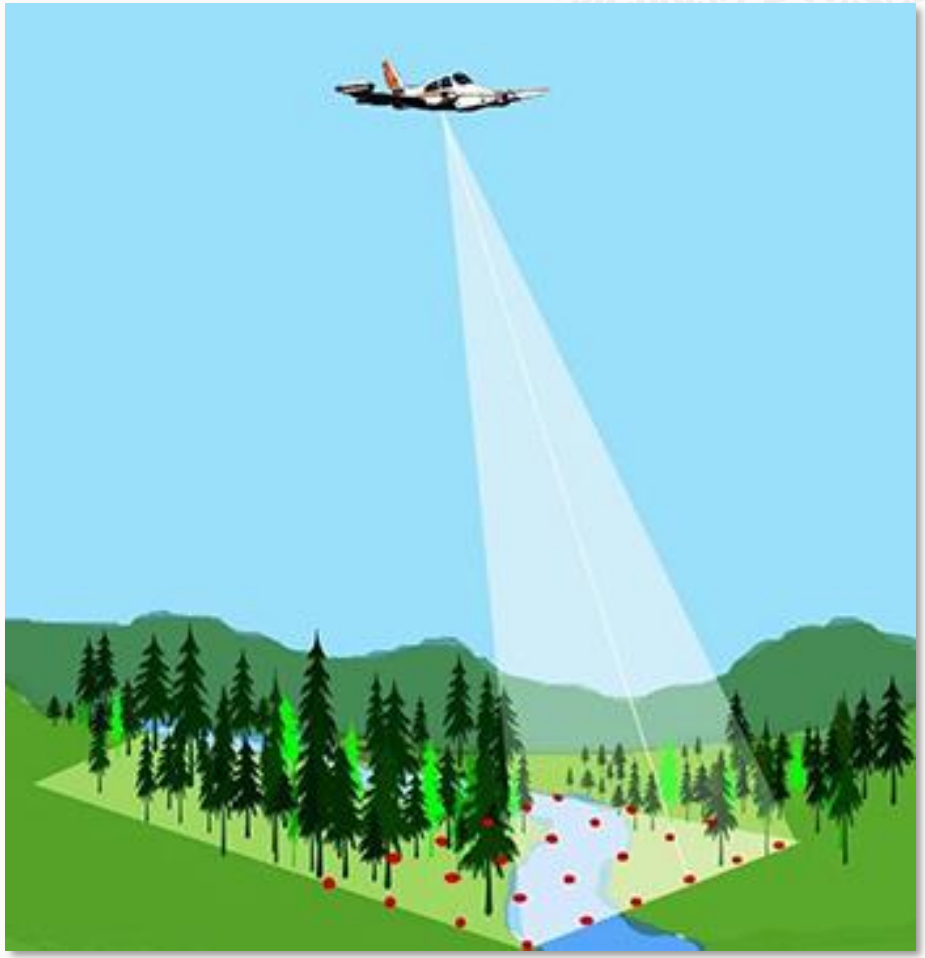
Light **D**etection **a**nd **R**anging …

- … is an active remote sensing system
- … that uses laser light (which is reflected by the surface it encounters)
- … to create 3-dimensional representations of real-world objects, environments, or surfaces
- … or to estimate the range between two objects

Sensors record the reflected light to measure a range.

LiDAR data are also known as **Point Clouds**.

Point Clouds are often used to generate products, such as digital elevation models, canopy models, biomass estimation, building models, contours, or digital twins.



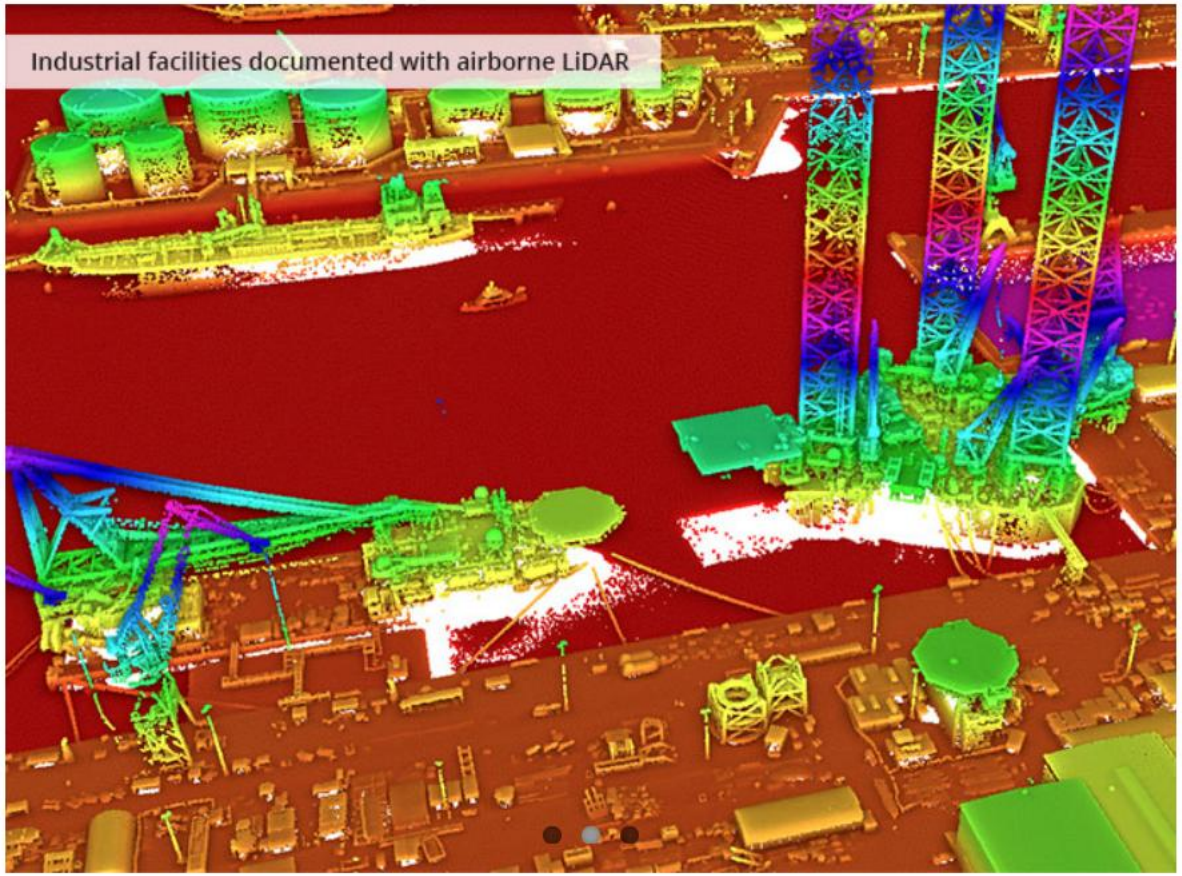
The Data Format

- The standard format for Point Clouds is the LAS file format
- There are other formats, such as E57
- The compressed version of LAS is LAZ
- LAS/LAZ files contain a collection of points with attributes
- Attributes are useful for analysis and visualization
  - XYZ coordinates
  - Point classification
  - Intensity of the returned pulse
  - RGB color values
  - Return number and the total number of returns from the pulse
  - Scan angle
  - Overlap points
  - ...

```
[opc@db-spatial-tools-instance LAStools]$ lasinfo64 ./data/zurich.laz
lasinfo (241210) report for './data/zurich.laz'
reporting all LAS header entries:
  file signature:             'LASF'
  file source ID:             0
  global_encoding:            1
  project ID GUID data 1-4:   00000000-0000-0000-0000-000000000000
  version major.minor:        1.2
  system identifier:          'LAStools (c) rapidlasso'
  generating software:        'LAStools'
  file creation day/year:     16/2015
  header size:                227
  offset to point data:       229
  number var. length records: 0
  point data format:          1
  point data record length:   28
  number of point records:    656837
  number of points by return: 467357 107076 47254 21904 8935
  scale factor x y z:         0.01 0.01 0.01
  offset x y z:               -0 -0 -0
  min x y z:                  676750.00 246000.00 547.29
  max x y z:                  676849.99 246099.99 573.87
the header is followed by 2 user-defined bytes
LASzip compression (version 2.4r0 c2 50000): POINT10 2 GPSTIME11 2
reporting minimum and maximum for all LAS point record entries ...
  X            67675000   67684999
  Y            24600000   24609999
  Z               50623      57390
  intensity           1       7669
  return_number       1          7
  number_of_returns   1          7
  edge_of_flight_line 0          0
  scan_direction_flag 0          0
  classification      2         17
  scan_angle_rank   -20         29
  user_data           0          0
  point_source_ID  2404      10102
  gps_time 78474513.065274 80531980.003439
WARNING: 228 points outside of header bounding box
number of first returns:        467357
number of intermediate returns: 82506
number of last returns:         466716
number of single returns:       359742
WARNING: there are 3160 points with return number 6
WARNING: there are 1151 points with return number 7
overview over number of returns of given pulse: 359742 119359 76362 52315 28914 13431 6714
histogram of classification of points:
          171354  ground (2)
           20228  low vegetation (3)
           37072  medium vegetation (4)
           98061  high vegetation (5)
          156178  building (6)
             579  noise (7)
          172798  overlap (12)
             567  bridge deck (17)
WARNING: real max z larger than header max z by 0.030000
WARNING: real min z smaller than header min z by 41.060000
```

Visualizing Point Cloud (1/2)

POINT CLOUD TECHNOLOGY



THE POWER OF POINT CLOUDS

3D point clouds serve as precise models of reality, available at low costs, and perfectly suited for high-performance processing.

Capture assets, sites, and entire environments 1:1 every year, every month, every day, every hour, every minute, or every second.

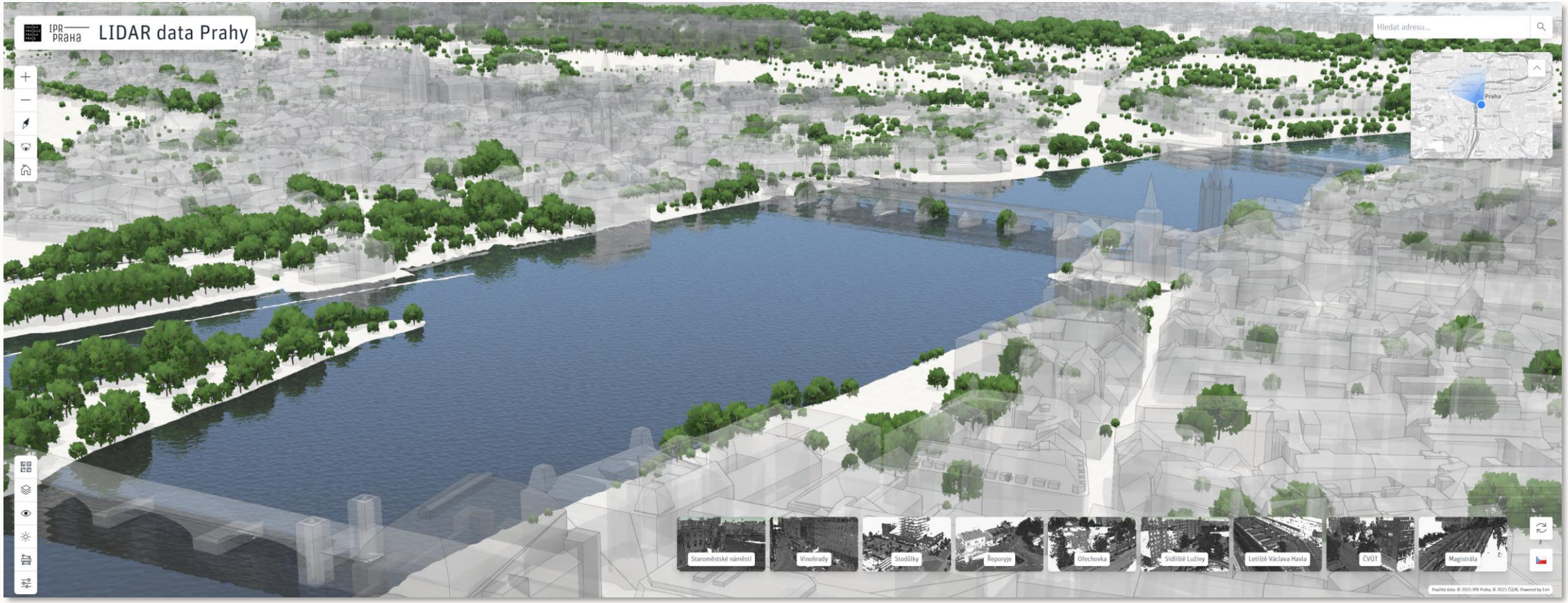
Run spatial analytics based on 3D point cloud streams that continuously deliver billions of data points.

Perform feature detection and differential analysis as core algorithms operating on 3D point clouds.

Build a unique spatio-temporal model of reality with 4D point clouds, combining 3D point clouds captured at different points in time.

Source: www.pointcloudtechnology.com/en/

Visualizing Point Cloud (2/2)



Source: app.iprpraha.cz/apl/app/CAMP/lidar/

Challenges with Point Clouds

Data Volume

- Can contain hundreds of millions of points
- Increasing densities with enhanced sensors and technology
- Overlapping and noisy data
- Versioning and metadata
- …

Storage and Computation

- Where do I put all this data?
- What about security?
- What are effective compression methods?
- How is the data indexed?
- Optimal format for analysis?
- Memory bottlenecks when processing or analyzing the data
- Parallel processing
- …

Analysis

- Derivative product generation (TINs, Contours, DEMs)
- Correlate with other data (vector, raster, networks, addresses …)
- …

Tools to work with LiDAR data?

A Selection of Tools

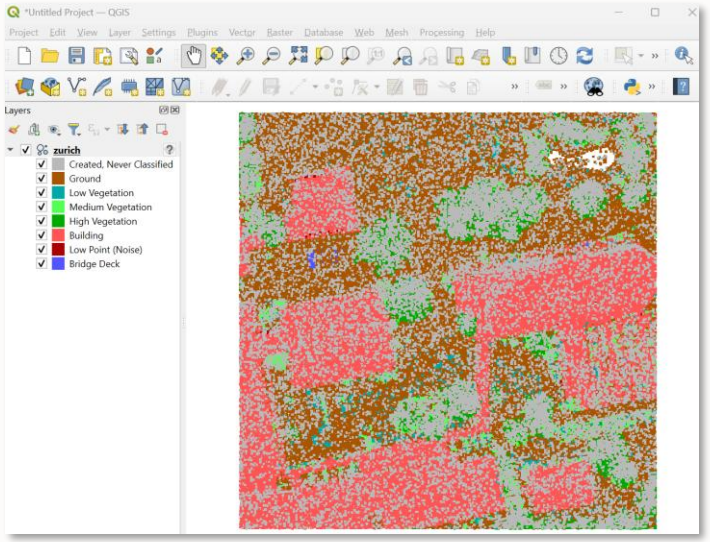
LAStools by rapidlasso

- lastools.github.io
- rapidlasso.de

- A set of command-line tools to load, process, transform, and visualize LiDAR datasets
- Tools divided into
  - Open-source
  - Free
  - Closed source
- New: GUI and workbench for LAStools
  - rapidlasso.de/laslook

QGIS

- Open-source software
- Visualizes LAS and LAZ files



Python libraries

- Visualize point clouds
  - PDAL
    pdal.io
  - laspy
    laspy.readthedocs.io

Setting up LAStools on a Compute machine

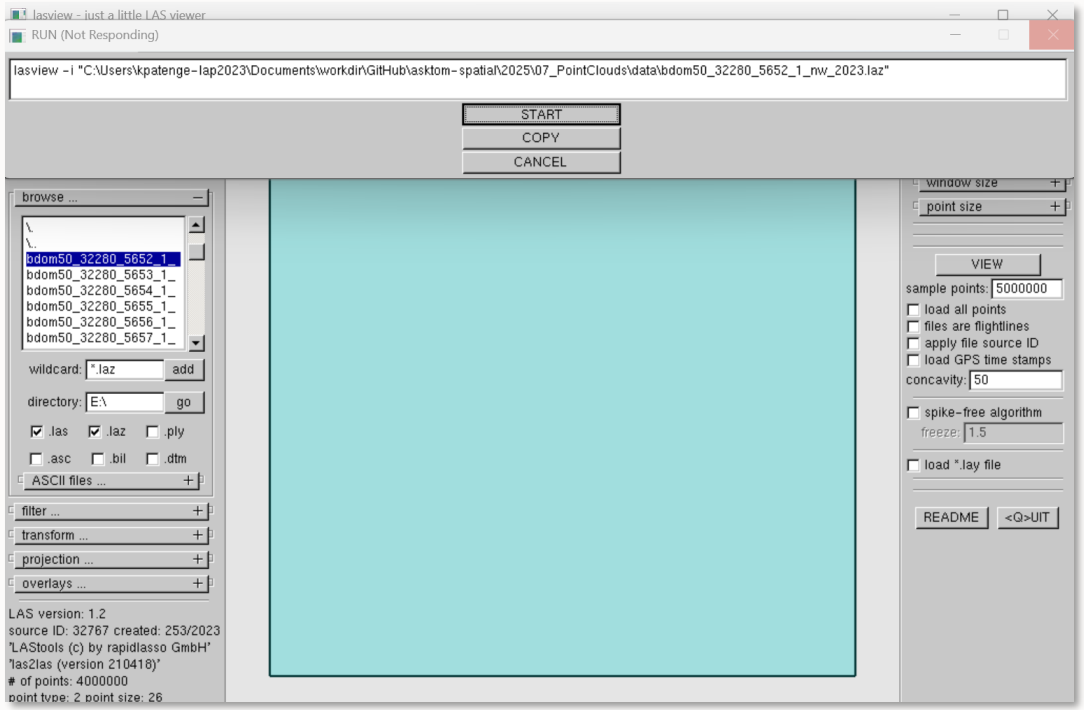
- Install SQL*Loader from Oracle Instant Client 23_8 for Linux (tools) (https://www.oracle.com/database/technologies/instant-client/downloads.html)
- Upload and extract the Database wallet. Set the `TNS_ADMIN` environment variable and modify `sqlnet.ora` to match the wallet location.
- Install LAStools (example for Linux):

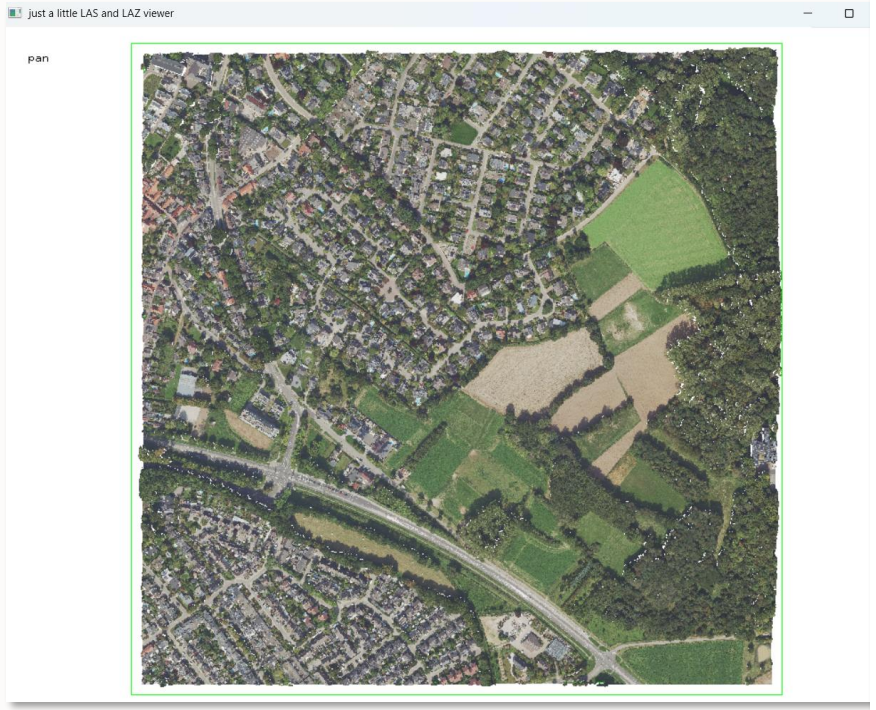
```
cd ~
mkdir lastools
cd lastools
wget https://downloads.rapidlasso.de/LAStools.tar.gz
tar -xvzf LAStools.tar.gz
```

- Add environment variables:
  - PATH (Oracle client path & lastools path) & LD_LIBRARY_PATH (Oracle client library path & lastools library path)
- Create a table in your DB schema before loading .las/.laz/.txt files

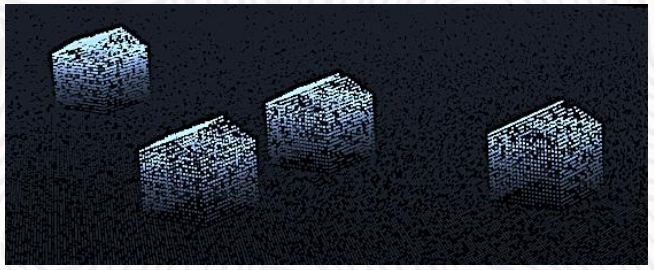
Using LAStools

View .laz data using “lasview”





Managing LiDAR data in the Oracle Database



Oracle Spatial

Two complementary approaches

- **Flat tables** containing X, Y, Z, and all other attributes
- Especially recommended if you can leverage intelligent Exadata storage servers (Autonomous Database, ExaCC, …)
  - High compression rates with Hybrid Columnar Compression (HCC)
  - Parallel enabled smart scan for best performance
- No indexes needed
- Direct load from .las/.laz files or from flat ASCII files
- Best for volatile datasets (no reblocking needed)

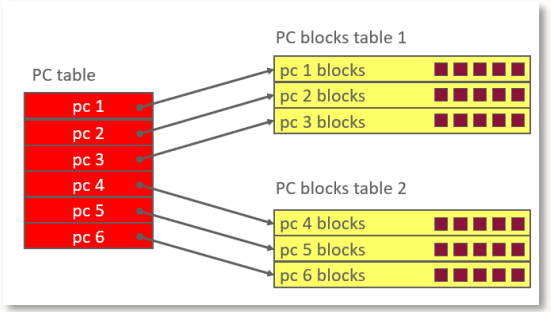
- **SDO_PC** object type

```
BASE_TABLE            VARCHAR2(70)
BASE_TABLE_COL        VARCHAR2(1024)

PC_ID                 NUMBER
BLK_TABLE             VARCHAR2(70)

PTN_PARAMS            VARCHAR2(1024)
PC_EXTENT             MDSYS.SDO_GEOMETRY
PC_TOL                NUMBER
PC_TOT_DIMENSIONS     NUMBER
PC_DOMAIN             MDSYS.SDO_ORGSCL_TYPE
PC_VAL_ATTR_TABLES    MDSYS.SDO_STRING_ARRAY
PC_OTHER_ATTRS        SYS.XMLTYPE
```

- Needs a spatial index for spatial searches
- The points are stored as arrays in a BLOB using SECUREFILES
  - Allows compression, encryption, and de-duplication



PL/SQL Package SDO_PC_PKG

SDO_PC_PKG.CLIP_PC
SDO_PC_PKG.CLIP_PC_FLAT
SDO_PC_PKG.CLIP_PC_FLAT_STRING
SDO_PC_PKG.CLIP_PC_INTO_TABLE
SDO_PC_PKG.CREATE_CONTOUR_GEOMETRIES
SDO_PC_PKG.CREATE_PC
SDO_PC_PKG.CREATE_PC_UNIFIED
SDO_PC_PKG.DROP_DEPENDENCIES
SDO_PC_PKG.GENERATE_CROSS_SECTION_AS_GEOMS
SDO_PC_PKG.GET_PT_IDS
SDO_PC_PKG.HAS_PYRAMID
SDO_PC_PKG.INIT
SDO_PC_PKG.PC_DIFFERENCE
SDO_PC_PKG.PC2DEM
SDO_PC_PKG.PRESERVES_LEVEL1
SDO_PC_PKG.SDO_PC_NN
SDO_PC_PKG.SDO_PC_NN_FOR_EACH
SDO_PC_PKG.TO_GEOMETRY

- Documentation
  - Oracle Spatial Developer´s Guide docs.oracle.com/en/database/oracle/oracle-database/23/spatl/spatial-developers-guide.pdf
- Code examples can be found in
  - $ORACLE_HOME/md/demo/PointCloud/examples
  - Requires the installation of the Oracle Database Examples
    - https://docs.oracle.com/en/database/oracle/oracle-database/21/exmpl/index.html
    - https://docs.oracle.com/en/database/oracle/oracle-database/19/exmpl/index.html

Processing Point Clouds

Two selected processing types of flat tables

CLIP_PC

- Clip Point Cloud
- 2D or 3D query window
- Returns points for any block whose extent intersects the query window
- Only points that intersect the query window are returned
- Creates a set of PC blocks, which can be stored or used in queries

CREATE_CONTOUR_GEOMETRIES

- Create the contour lines for a point cloud
- Processing is grid-based. The points within a grid cell get averaged.
- Set of elevations as input parameters

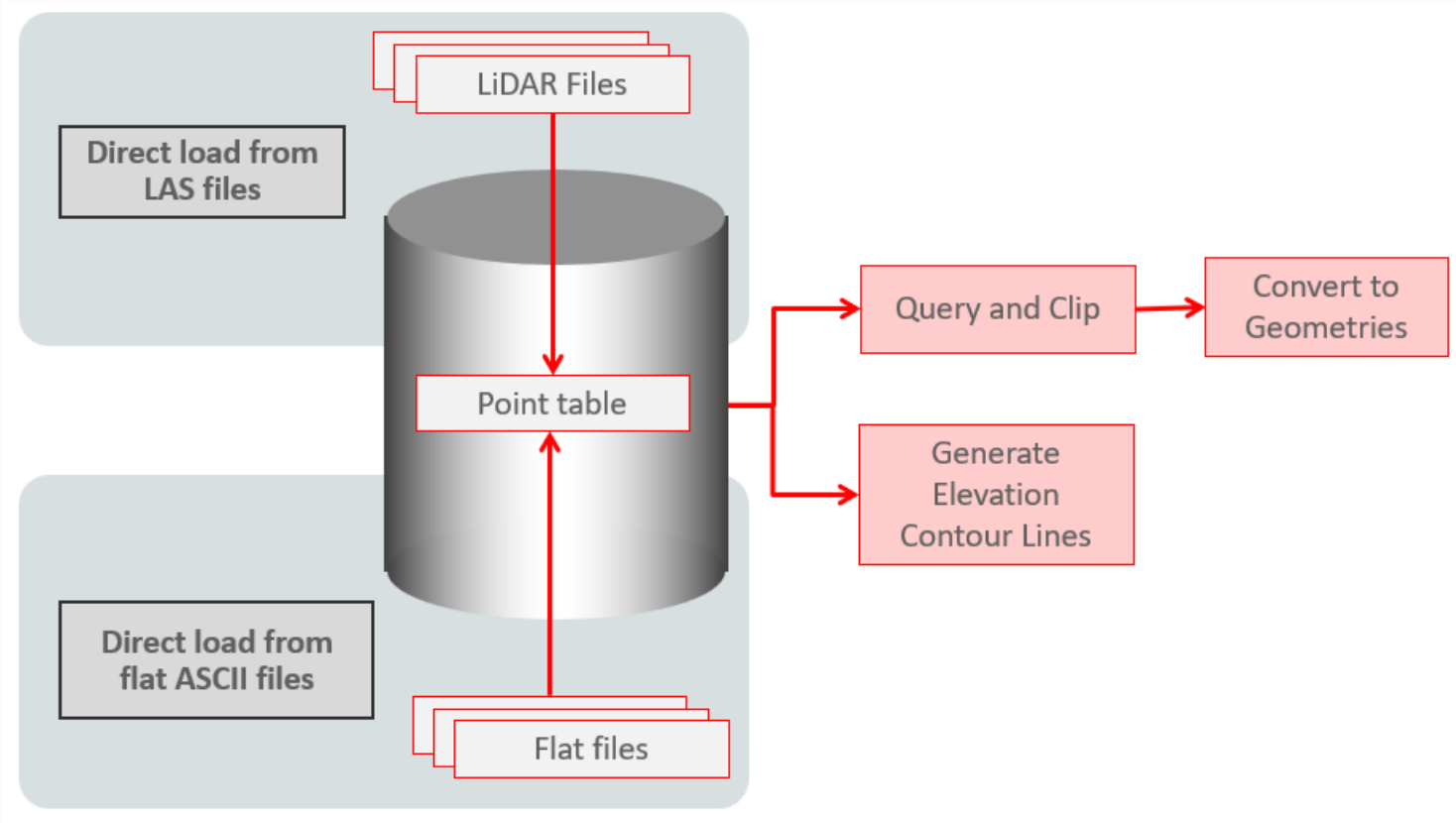
```
SDO_PC_PKG.CLIP_PC_FLAT(
       geometry IN SDO_GEOMETRY,
       table_name IN VARCHAR2,
       tolerance IN NUMBER,
       other_dim_qry IN SDO_MBR,
       mask IN VARCHAR2 DEFAULT NULL
) RETURN REF CURSOR;
```

```
SDO_PC_PKG.CREATE_CONTOUR_GEOMETRIES(
       pc_flat_table IN VARCHAR2,
       srid IN NUMBER,
       sampling_resolution IN NUMBER,
       elevations IN SDO_ORDINATE_ARRAY,
       region IN SDO_GEOMETRY
) RETURN SDO_GEOMETRY_ARRAY;
```

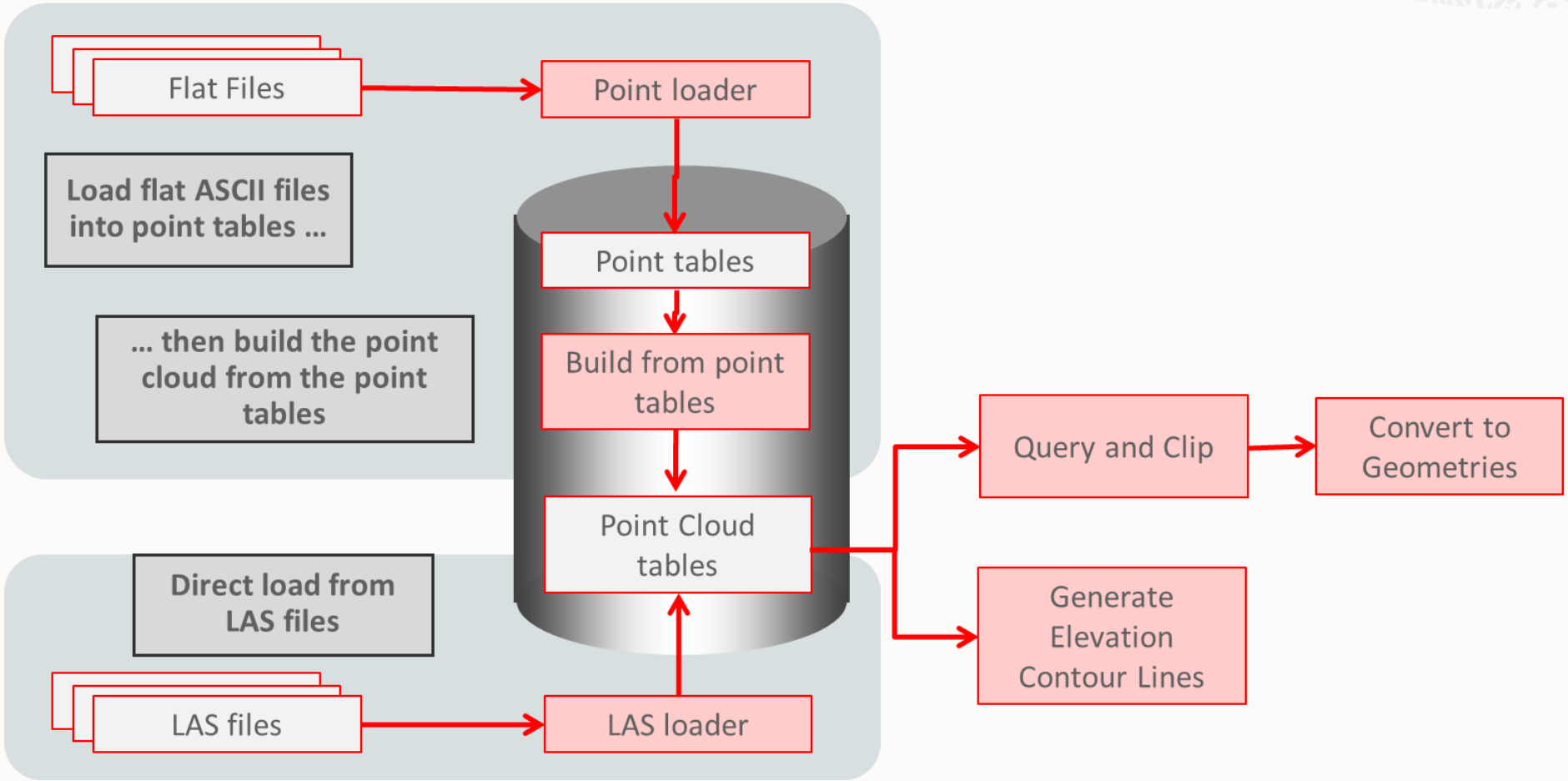
A LiDAR data workflow

Using datasets published on
www.opengeodata.nrw.de/produkte/geobasis/hm/bdom50_las/bdom50_las/
under the license "Datenlizenz Deutschland – Zero – Version 2.0"

Using Flat Tables



Using SDO_PC



Example 1:
Loading a larger set of .las/.laz files

The LiDAR datasets used in this example were downloaded from the [OpenData.NRW portal](https://www.opengeodata.nrw.de/produkte/geobasis/hm/bdom50_las/bdom50_las/).

License information: [Datenlizenz Deutschland - Zero - Version 2.0](https://www.govdata.de/dl-de/zero-2-0)

Check LAS/LAZ files using LAStools

```
#!/bin/bash
NUM_FILES=0
TOTAL_POINTS=0
TOTAL_AREA=0

LAS_FILES=~/data/pc
ls -lh $LAS_FILES

echo Filename, X_min, X_max, Y_min, Y_max, Z_min, Z_max, points, area, density

find $LAS_FILES -name *.laz | while read file
do
  NUM_POINTS=`lasinfo64 $file -nc -stdout | grep 'number of points by return' |
awk '{print $6}'`
  MIN=`lasinfo64 $file -nc -stdout | grep 'min x y z:'`
  MAX=`lasinfo64 $file -nc -stdout | grep 'max x y z:'`
  MIN_X=`echo $MIN | awk '{print int($5+0.5)}'`
  MIN_Y=`echo $MIN | awk '{print int($6+0.5)}'`
  MIN_Z=`echo $MIN | awk '{print int($7+0.5)}'`
  MAX_X=`echo $MAX | awk '{print int($5+0.5)}'`
  MAX_Y=`echo $MAX | awk '{print int($6+0.5)}'`
  MAX_Z=`echo $MAX | awk '{print int($7+0.5)}'`
  #((AREA=(MAX_X-MIN_X)*(MAX_Y-MIN_Y)))
  ((DENSITY=NUM_POINTS/AREA))
  NAME="$(basename $file)"
  echo
$NAME,$MIN_X,$MAX_X,$MIN_Y,$MAX_Y,$MIN_Z,$MAX_Z,$NUM_POINTS,$AREA,$DENSITY
  ((NUM_FILES++))
  ((TOTAL_POINTS+=NUM_POINTS))
  ((TOTAL_AREA+=AREA))
done
```

```
Filename, X_min, X_max, Y_min, Y_max, Z_min, Z_max, points, area, density
bdom50_32280_5660_1_nw_2023.laz,280000,281000,5660000,5661000,30,73,4000000,1000000,4
bdom50_32280_5659_1_nw_2023.laz,280000,281000,5659000,5660000,30,65,4000000,1000000,4
bdom50_32280_5658_1_nw_2023.laz,280000,281000,5658000,5659000,31,72,4000000,1000000,4
bdom50_32280_5657_1_nw_2023.laz,280000,281000,5657000,5658000,32,75,4000000,1000000,4
bdom50_32280_5656_1_nw_2023.laz,280000,281000,5656000,5657000,32,77,4000000,1000000,4
bdom50_32280_5655_1_nw_2023.laz,280000,281000,5655000,5656000,34,82,4000000,1000000,4
bdom50_32280_5654_1_nw_2023.laz,280000,281000,5654000,5655000,37,119,4000000,1000000,4
bdom50_32280_5653_1_nw_2023.laz,280000,281000,5653000,5654000,43,116,4000000,1000000,4
bdom50_32280_5652_1_nw_2023.laz,280000,281000,5652000,5653000,50,131,4000000,1000000,4
bdom50_32281_5661_1_nw_2023.laz,281000,282000,5661000,5662000,23,69,4000000,1000000,4
bdom50_32281_5660_1_nw_2023.laz,281000,282000,5660000,5661000,28,74,4000000,1000000,4
bdom50_32281_5659_1_nw_2023.laz,281000,282000,5659000,5660000,31,71,4000000,1000000,4
bdom50_32281_5658_1_nw_2023.laz,281000,282000,5658000,5659000,33,72,4000000,1000000,4
bdom50_32281_5657_1_nw_2023.laz,281000,282000,5657000,5658000,35,66,4000000,1000000,4
bdom50_32281_5656_1_nw_2023.laz,281000,282000,5656000,5657000,35,74,4000000,1000000,4
bdom50_32281_5655_1_nw_2023.laz,281000,282000,5655000,5656000,32,78,4000000,1000000,4
bdom50_32281_5654_1_nw_2023.laz,281000,282000,5654000,5655000,39,84,4000000,1000000,4
bdom50_32281_5653_1_nw_2023.laz,281000,282000,5653000,5654000,52,110,4000000,1000000,4
bdom50_32281_5652_1_nw_2023.laz,281000,282000,5652000,5653000,60,130,4000000,1000000,4
bdom50_32280_5661_1_nw_2023.laz,280000,281000,5661000,5662000,26,69,4000000,1000000,4
```

Convert LAS/LAZ files using LAStools

Input: A set of 20 .laz files

```
#!/bin/bash
cd "$(dirname "$0")"

LAS_FILES=~/data/pc
ls -lh $LAS_FILES

TILE_ID=1
echo Converting LAS/LAZ files to TXT in $LAS_FILES

find $LAS_FILES -name *.laz | while read file
do
  echo reading information from file $file
  lasinfo64 $file
  text=${file/%.laz/.txt}
  base=${file##*/}
  echo Converting file $base to $text ...
  echo ... las2txt64 -i "$file" -o $text -parse xyzirnedcaRGB -sep
comma
  time las2txt64 -i "$file" -o $text -parse xyzirnedcaRGB -sep comma
-verbose
  awk -F',' -v OFS=',' -v col1=$TILE_ID -v col2=$base '{print $0,
col1, col2}' $text > temp && mv temp $text
  echo show header of converted file
  head -n5 $text
  ((TILE_ID++))
done

ls -lh $LAS_FILES
```

```
head bdom50_32281_5660_1_nw_2023.txt
281000.25,5660199.75,46.61,38912,1,1,0,0,0,0,14848,14848,16128,11,bdom50
_32281_5660_1_nw_2023.laz
281000.75,5660199.75,46.49,37632,1,1,0,0,0,0,14336,14336,16128,11,bdom50
_32281_5660_1_nw_2023.laz
281001.25,5660199.75,45.99,37632,1,1,0,0,0,0,14592,14336,16384,11,bdom50
_32281_5660_1_nw_2023.laz
281001.75,5660199.75,45.77,37120,1,1,0,0,0,0,13824,13824,16128,11,bdom50
_32281_5660_1_nw_2023.laz
281002.25,5660199.75,59.60,36096,1,1,0,0,0,0,13824,13568,14080,11,bdom50
_32281_5660_1_nw_2023.laz
281002.75,5660199.75,60.25,46848,1,1,0,0,0,0,26624,26880,22016,11,bdom50
_32281_5660_1_nw_2023.laz
281003.25,5660199.75,60.37,44800,1,1,0,0,0,0,25600,25344,20736,11,bdom50
_32281_5660_1_nw_2023.laz
281003.75,5660199.75,61.21,46080,1,1,0,0,0,0,24832,25344,19200,11,bdom50
_32281_5660_1_nw_2023.laz
281004.25,5660199.75,61.54,44800,1,1,0,0,0,0,24320,24832,18944,11,bdom50
_32281_5660_1_nw_2023.laz
281004.75,5660199.75,61.79,45568,1,1,0,0,0,0,22016,22016,17152,11,bdom50
_32281_5660_1_nw_2023.laz
```

Create tables to store the points

```
-- Flat table
DROP TABLE IF EXISTS lidar_points CASCADE CONSTRAINTS PURGE;

CREATE TABLE IF NOT EXISTS lidar_points (
  x                   NUMBER,     -- X
  y                   NUMBER,     -- Y
  z                   NUMBER,     -- Z
  intensity           NUMBER,     -- i => Intensity
  return_number       NUMBER,     -- r => Return Number
  number_of_returns   NUMBER,     -- n => Number of Returns
  edge_of_flight_line NUMBER,     -- e => Flightline Edge
  scan_direction_flag NUMBER,     -- d => Scan Direction Flag
  classification      NUMBER,     -- c => Classification
  scan_angle_rank     NUMBER,     -- a => Scan Angle Rank
  r                   NUMBER,     -- R => Color red (2 bytes [0-65536])
  g                   NUMBER,     -- G => Color green (2 bytes [0-65536])
  b                   NUMBER,     -- B => Color blue (2 bytes [0-65536])
  tile_id             NUMBER,       -- tile identifier
  sourcefile          VARCHAR2(200)  -- File source
)
NOLOGGING;
```

Load converted data as points into a flat table

SQL*Loader

```
#!/bin/bash
cd "$(dirname "$0")"

# Change the following to match your environment:
DB_SERVICE=<service>
DB_USER=<user>
DB_PASS=<pwd>

TXT_FILES=../../data/pc

echo Loading text files from $TXT_FILES

# The following works for all file names (including those with
spaces)
find $TXT_FILES -name *.txt | while read file
do
  echo Loading $file
  sqlldr $DB_USER/$DB_PASS@$DB_SERVICE \
    control=04a_load_point_table.ctl \
    data=$file \
    direct=yes \
    rows=100000
done

# ~ 3 mins for 80 mio points
```

```
-- SQL*Loader control file to load point cloud data into a staging table
LOAD DATA
  APPEND
  INTO TABLE lidar_points
  FIELDS TERMINATED BY ',' (
  x,
  y,
  z,
  intensity,
  return_number,
  number_of_returns,
  edge_of_flight_line,
  scan_direction_flag,
  classification,
  scan_angle_rank,
  r,
  g,
  b,
  tile_id,
  sourcefile
)
```

Show stored points

Worksheet | Query Builder

```
1  SELECT * FROM lidar_points;
```

Script Output | Query Result

SQL | Fetched 50 rows in 0,039 seconds

	X	Y	Z	INTENSITY	RETURN...	NUMBE...	EDGE_...	SCAN_D...	CLASSIFI...	SCAN_...	R	G	B	TILE_ID	SOURCEFILE
1	80333.25	5660726.25	32.72	54272	1	1	0	0	0	0	27136	31488	22784	1	bdom50_32280_5660_1_nw_2023.1
2	80333.75	5660726.25	32.66	51968	1	1	0	0	0	0	25344	28672	22016	1	bdom50_32280_5660_1_nw_2023.1
3	80334.25	5660726.25	32.46	45568	1	1	0	0	0	0	20480	20736	18176	1	bdom50_32280_5660_1_nw_2023.1
4	80334.75	5660726.25	32.27	53248	1	1	0	0	0	0	26112	29952	23040	1	bdom50_32280_5660_1_nw_2023.1
5	80335.25	5660726.25	32.23	55808	1	1	0	0	0	0	28928	32000	24064	1	bdom50_32280_5660_1_nw_2023.1
6	80335.75	5660726.25	32.25	41984	1	1	0	0	0	0	17408	19200	14080	1	bdom50_32280_5660_1_nw_2023.1
7	80336.25	5660726.25	32.35	56576	1	1	0	0	0	0	29184	33280	24576	1	bdom50_32280_5660_1_nw_2023.1
8	80336.75	5660726.25	32.52	54528	1	1	0	0	0	0	25856	29440	22528	1	bdom50_32280_5660_1_nw_2023.1
9	80337.25	5660726.25	32.58	51456	1	1	0	0	0	0	23808	26880	21248	1	bdom50_32280_5660_1_nw_2023.1
10	80337.75	5660726.25	32.58	52224	1	1	0	0	0	0	21760	24832	18432	1	bdom50_32280_5660_1_nw_2023.1
11	80338.25	5660726.25	32.63	58112	1	1	0	0	0	0	29952	34560	25856	1	bdom50_32280_5660_1_nw_2023.1
12	80338.75	5660726.25	32.64	48640	1	1	0	0	0	0	19968	22016	17920	1	bdom50_32280_5660_1_nw_2023.1
13	80339.25	5660726.25	32.52	52480	1	1	0	0	0	0	22528	26112	20480	1	bdom50_32280_5660_1_nw_2023.1
14	80339.75	5660726.25	32.23	40192	1	1	0	0	0	0	14848	15872	13056	1	bdom50_32280_5660_1_nw_2023.1
15	80340.25	5660726.25	32.09	50432	1	1	0	0	0	0	26112	28160	21248	1	bdom50_32280_5660_1_nw_2023.1
16	80340.75	5660726.25	32.13	51456	1	1	0	0	0	0	33536	34048	25088	1	bdom50_32280_5660_1_nw_2023.1
17	80341.25	5660726.25	32.2	53248	1	1	0	0	0	0	36096	35584	26624	1	bdom50_32280_5660_1_nw_2023.1
18	80341.75	5660726.25	32.27	50944	1	1	0	0	0	0	34048	33792	26112	1	bdom50_32280_5660_1_nw_2023.1
19	80342.25	5660726.25	32.32	49664	1	1	0	0	0	0	35584	33792	25856	1	bdom50_32280_5660_1_nw_2023.1
20	80342.75	5660726.25	32.33	48384	1	1	0	0	0	0	35840	33536	25600	1	bdom50_32280_5660_1_nw_2023.1
21	80343.25	5660726.25	32.32	46592	1	1	0	0	0	0	34816	32256	24320	1	bdom50_32280_5660_1_nw_2023.1

Process points

Clipping

```
CREATE TABLE IF NOT EXISTS lidar_points_clip AS
SELECT * FROM lidar_points WHERE 0 = 1;

-- Clip
DECLARE
  points_cursor sys_refcursor;
  TYPE points_list IS TABLE OF lidar_points%rowtype;
  points points_list;
BEGIN
  points_cursor :=
    SDO_PC_PKG.CLIP_PC_FLAT (
      geometry =>
        SDO_GEOMETRY (
          2003, 25832, NULL,
          SDO_ELEM_INFO_ARRAY (1, 1003, 3),
          SDO_ORDINATE_ARRAY (281200, 5654100, 281300,
5654200 )
        ),
      table_name    => 'LIDAR_POINTS',
      tolerance     => 0.000005,
      other_dim_qry => null,
      mask          => null   -- Default: mask=ANYINTERACT
   );
  LOOP
    FETCH points_cursor
      BULK COLLECT INTO points
      LIMIT 10000;
    FORALL I in 1 .. points.COUNT
      INSERT INTO lidar_points_clip VALUES points(i);
    EXIT WHEN points_cursor%NOTFOUND;
  END LOOP;
  CLOSE points_cursor;
END;
/

SELECT count(*) FROM lidar_points_clip;
```

Save the date!
LiDAR Part 2

Processing, Querying, and Visualizing LiDAR dataset

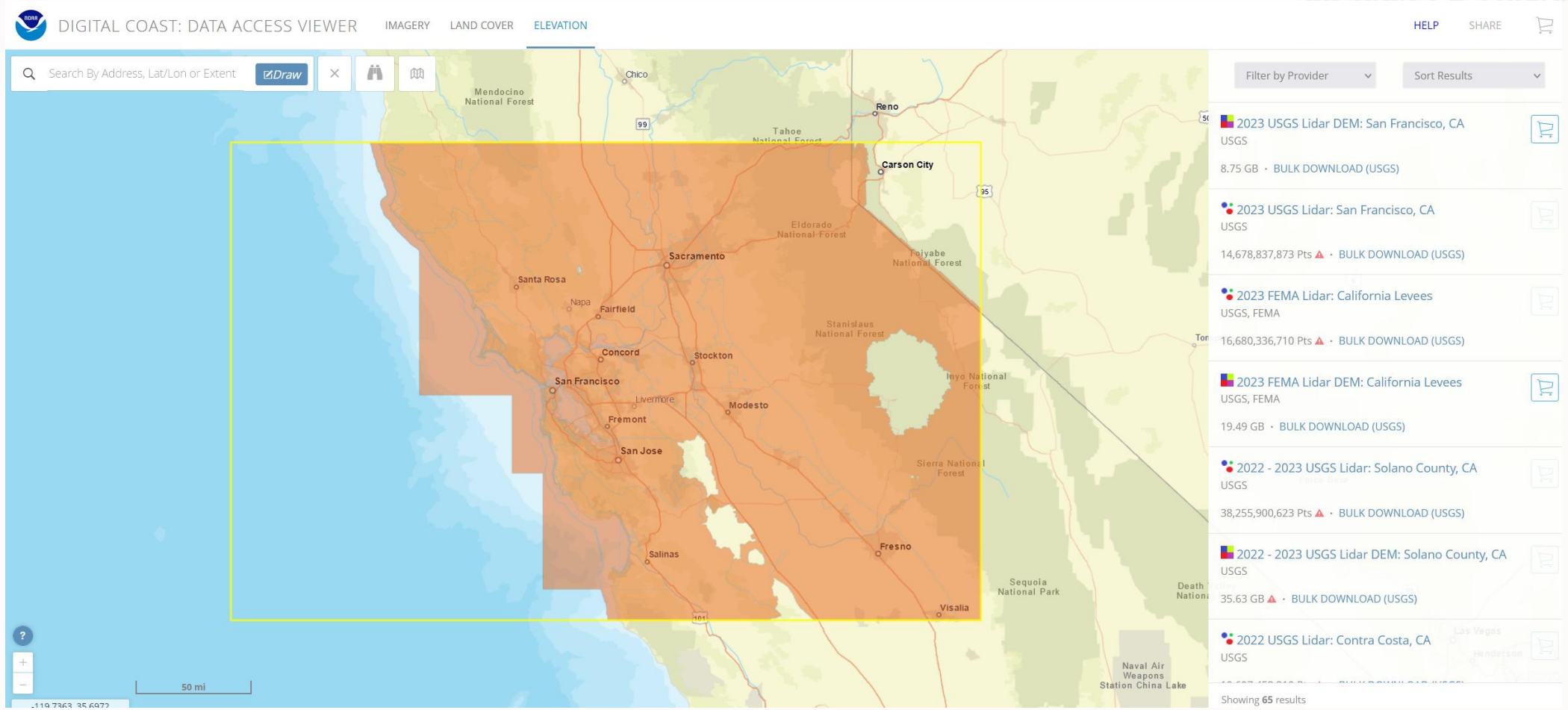
Where to find datasets

Some Resources

- gisgeography.com/top-6-free-lidar-data-sources/
- Demodata for a *Forest and agricultural area* and *Sandviken urban area* (Sweden) www.lantmateriet.se/sv/geodata/vara-produkter/produktlista/laserdata-nedladdning-nh/#anchor-2
- Geoportal Praha geoportalpraha.cz/en/data-and-services/articles-and-projects/3d-model/3d-model-lidar
- European LiDAR and Aerial Maps lidarandaerialarchaeology.com/europe/
- Collection of open nation-scale LiDAR datasets registry.opendata.aws/open-lidar-data/
- National LiDAR datasets
  - http://vterrain.org/Locations/eu/
  - https://en.wikipedia.org/wiki/National_lidar_dataset publications.jrc.ec.europa.eu/repository/bitstream/JRC126223/jrc126223_jrc126223_lidaropensourcedata.pdf

Digital Coast Access Viewer

coast.noaa.gov/dataviewer/#/lidar/search/



Channels to post questions at any time

stackoverflow

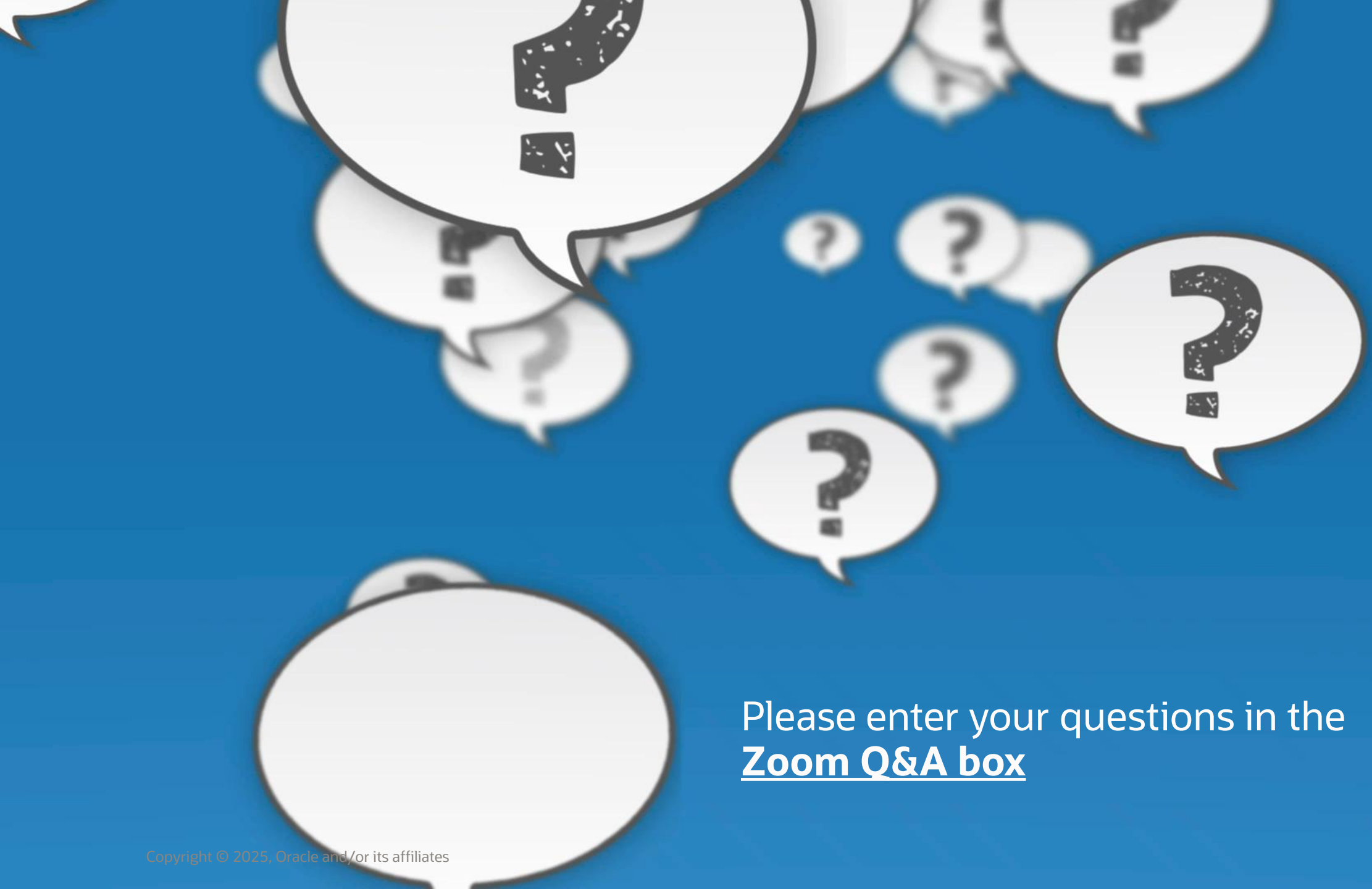
stackoverflow.com/questions/tagged/oracle-spatial

Oracle Forums
Posts tagged as ‘spatial’

forums.oracle.com/ords/apexds/domain/dev-community/category/database-software?tags=spatial

Resources on Oracle Spatial Technologies

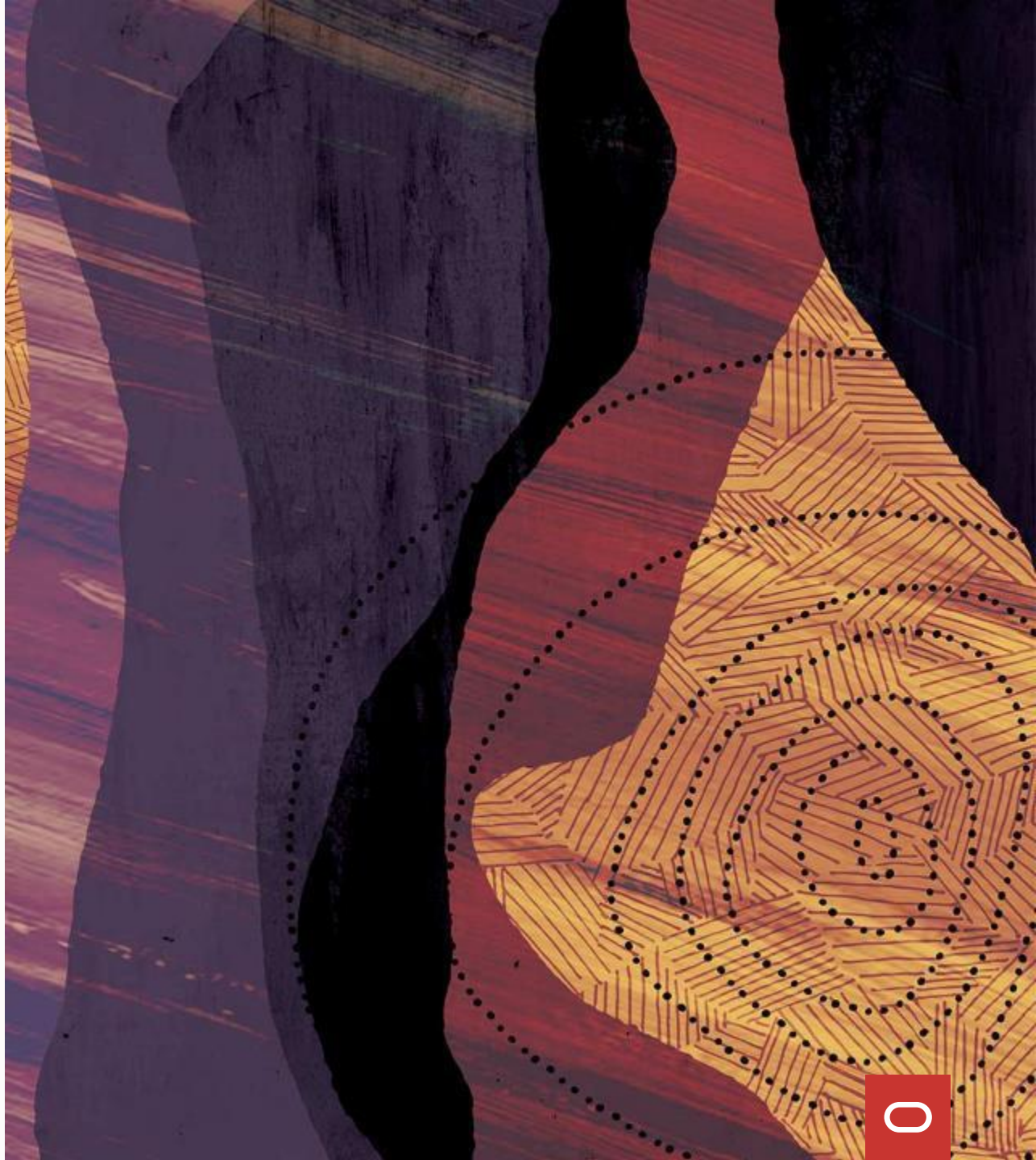
- Vector tiles & H3 blog post & Video: bit.ly/OracleSpatialVT
- Oracle Spatial technologies: www.oracle.com/database/spatial/
- Oracle LiveLabs: bit.ly/golivelabs-spatial
- Blog: blogs.oracle.com/database/category/db-spatial
- Slack (Please join #spatial channel): bit.ly/Join-ANDOUC-Slack
- YouTube: bit.ly/Spatial-Graph-YouTube
- AskTOM Spatial Series videos: bit.ly/AskTOMSpatial
- LinkedIn: bit.ly/Spatial-Graph-LinkedIn



Please enter your questions in the **Zoom Q&A box**

Thank you for your attention!

karin.patenge@oracle.com
hans.viehmann@oracle.com




ORACLE