FIRST STEPS IN SPATIAL DATA
David Kurtz
DOAG 2021
/*+Go-Faster*/ Consultancy*/

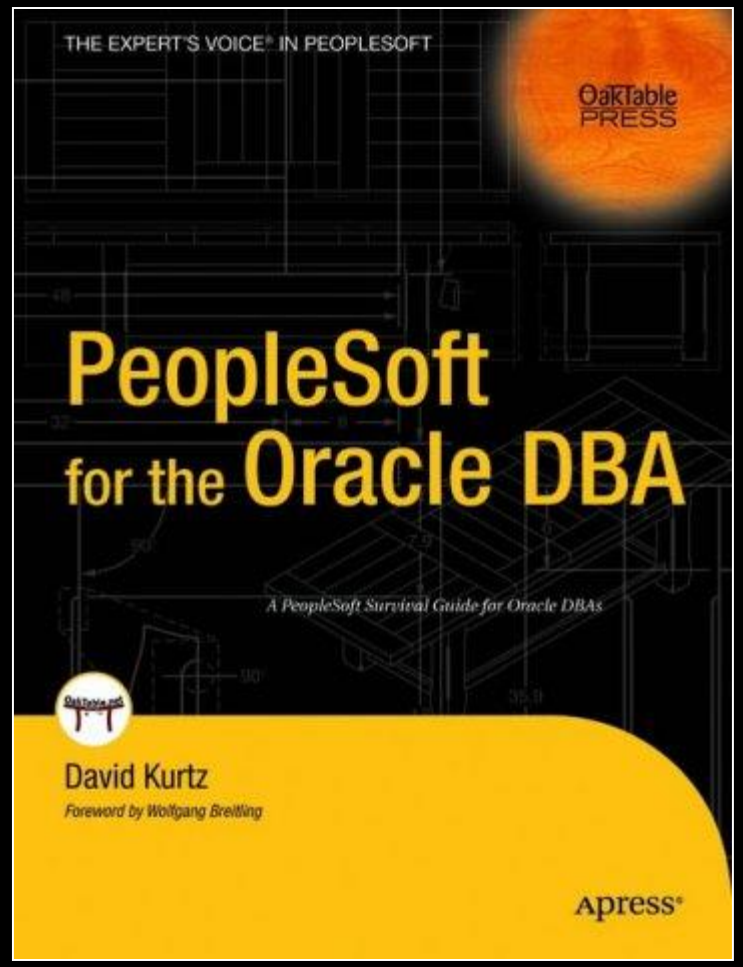
WHO AM I

Go-Faster Consultancy

- Performance tuning
  - Oracle RDBMS
  - PeopleSoft ERP
- www.go-faster.co.uk
- blog.go-faster.co.uk
- blog.psftdba.com

Oak Table

ORACLE ACE Director



SPATIAL DATA

- Loading GPS data in GPX format
- Convert GPX to Spatial Geometry
- Analyse Track in proximity to route
- Publically available sources of spatial data
- Intersecting Geometries
- Pre-processing
- Text Indexing and Searching
- Apex

BACKGROUND

Shenfield - Hatfield Broad Oak

the route (85.7km)

Map updated 2021-10-23 20:04

Download GPX

- as 1 routes (as on the map) in a GPX file
- as a single track

Show profile

SPATIAL QUERIES

I can search for routes with structured data

- Date range
- Name of leader
- Text of description
  - Start point

I ride up Swain's Lane every morning

- Am I getting faster or slower?

I want to plan a ride for this weekend

- I want to see previous routes that go to a place/area
  - E.g. routes in the Chilterns
  - E.g. routes that go to Blackmore

SPATIAL QUERIES

GPS tracks → Spatial Geometries

Load and Parse GPX files

Definition of places/areas

- E.g.
  - Parish of Blackmore
  - County of Essex
  - Chiltern Hills AONB
  - Swain's Lane

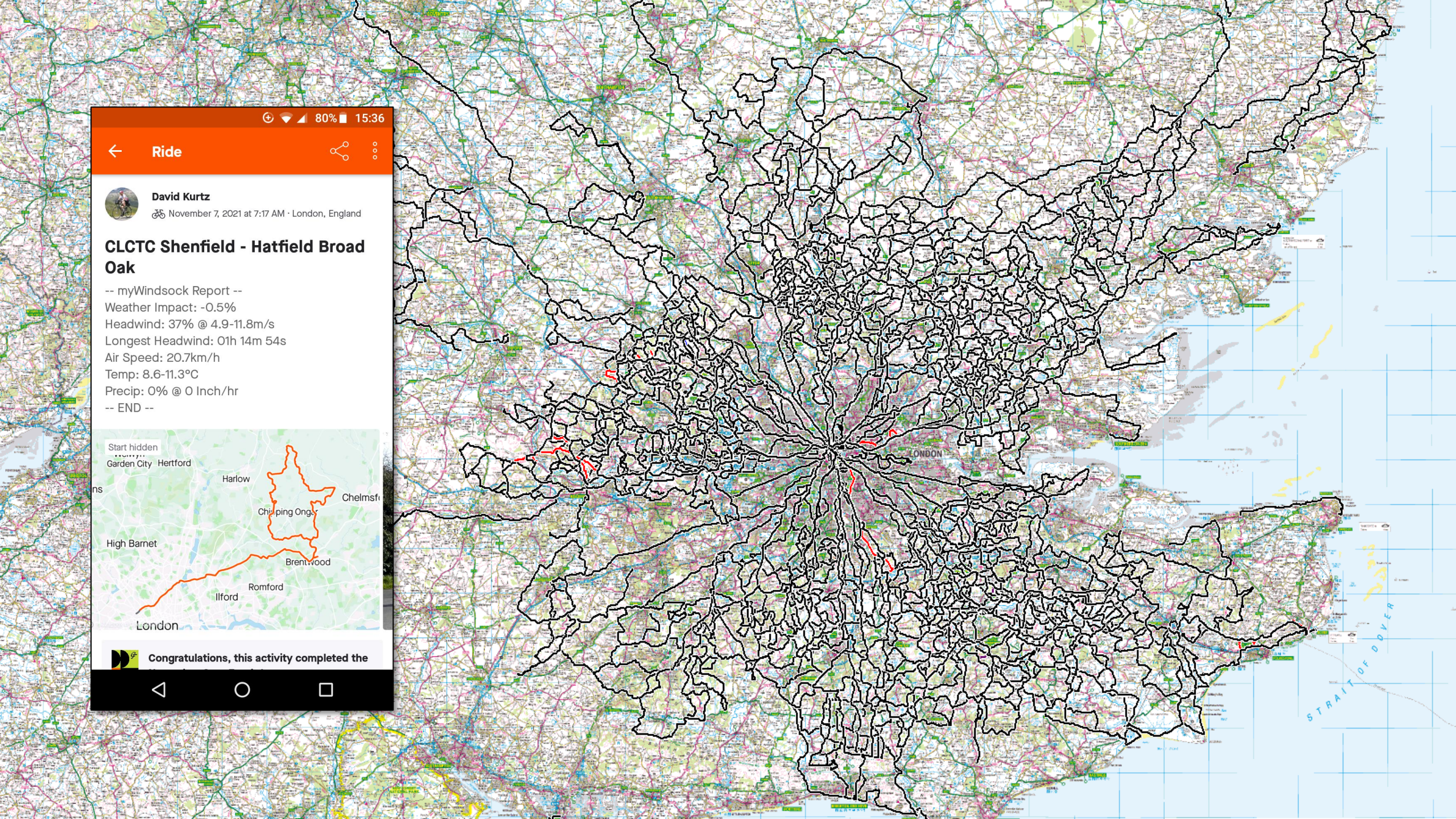
Political/Administrative boundaries

- Publically available (commercial licence)
- Industry standard format

Manual

Spatial Operations

- Track/Route intersects with an area
- One spatial geometry intersects with/overlaps/contains another spatial geometry


80% 15:36
Ride
David Kurtz
November 7, 2021 at 7:17 AM · London, England
CLCTC Shenfield - Hatfield Broad Oak
-- myWindsock Report --
Weather Impact: -0.5%
Headwind: 37% @ 4.9-11.8m/s
Longest Headwind: 01h 14m 54s
Air Speed: 20.7km/h
Temp: 8.6-11.3°C
Precip: 0% @ 0 Inch/hr
-- END --
Start hidden
Garden City
Hertford
Harlow
High Barnet
Romford
Ilford
London
Congratulations, this activity completed the
LONDON
STRAIT OF DOVER

LOADING GPX FILES

LOAD GPX FILES INTO XML DATATYPES

Strava bulk export as zip file

- Structured Data in CSV – activities.csv
- .gpx / .fit for each activity
- Some files are gzipped *.gpx.gz* / *.fit.gz*, some not.
- I prefer to zip all of them

Some applications upload .fit files to Strava

GPSBabel to convert them to *.gpx*

I unzip the Strava export.

- I can read a single zipped file in PL/SQL
- I haven't worked out how to read a file in a zipped archive

```
cd /vagrant
mkdir /tmp/strava
unzip /vagrant/export_1679301.zip activities.csv -d /tmp/strava
unzip /vagrant/export_1679301.zip "activities/*" -d /tmp/strava
gzip -9vf /tmp/strava/activities/*.gpx
```

```
for %i in (*.fit) do "C:\Program Files
(x86)\GPSBabel\GPSBabel.exe" -i garmin_fit -f %~dpni.fit -
o gpx -F %~dpni.gpx
```

CREATE SCHEMA

XDBADMIN allows user to register an XML schema

STRAVA directory from where the Strava data files are located

- Read permission
- Execute permission for another shell script

ACTIVITIES

- Directory where .gpx files are located
- Only Read Permission

EXEC_DIR is where gzip executable is located

- so I can unzip files on the fly to read them
- Read and Execute permission

```
connect / as sysdba
create user strava identified by strava;
grant connect, resource to strava;
grant create view to strava;
grant select catalog role to strava;
grant XDBADMIN to STRAVA;
grant alter session to STRAVA;
alter user strava quota unlimited on users;
alter user strava default tablespace users;

GRANT CREATE ANY DIRECTORY TO strava;
CREATE OR REPLACE DIRECTORY strava as '/tmp/strava';
CREATE OR REPLACE DIRECTORY activities as
'/tmp/strava/activities';
CREATE OR REPLACE DIRECTORY exec_dir AS '/usr/bin';

GRANT READ, EXECUTE ON DIRECTORY strava TO strava;
GRANT READ ON DIRECTORY activities TO strava;
GRANT READ, EXECUTE ON DIRECTORY exec_dir TO strava;
```

STRINGS WITH NEW LINES IN .CSV FILE

Some strings in CSV contain new line characters

External table load considers this to be the end of the record.

Pre-process the row with *gawk*

- Create simple shell script *nlfix.sh*
- Make it executable
- Note fully qualified path for *gawk*

- ASKTom: SQL loader not loading all the needed rows due to new line character and enclosement character
- How to remove new lines within double quotes?

```
#nlfix.sh
/usr/bin/gawk -v RS='"' 'NR % 2 == 0 { gsub(/\n/, "") } { printf("%s%s", $0, RT) }' $*
```

CREATE EXTERNAL TABLE TO READ .CSV

- Create External Table to Read Activities
- Invoke pre-processor to remove new line characters
- Create activities table from external table
- Delete activities where time/distance changed
- Add new activities
- Update changes to structured data (descriptions etc.)

```
create table strava.activities_ext
(…)
ORGANIZATION EXTERNAL
(TYPE ORACLE_LOADER
 DEFAULT DIRECTORY strava
 ACCESS PARAMETERS
 (RECORDS DELIMITED BY newline
  SKIP 1
  DISABLE_DIRECTORY_LINK_CHECK
  PREPROCESSOR strava:'nlfix.sh'
  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' RTRIM
  MISSING FIELD VALUES ARE NULL
  REJECT ROWS WITH ALL NULL FIELDS
  NULLIF = BLANKS
(Activity_ID,Activity_Date date "DD Mon yyyy,HH24:mi:ss"
,Activity_Name,Activity_Type,Activity_Description
…))
LOCATION ('activities.csv')
) REJECT LIMIT 5
/
```

PL/SQL PACKAGE TO READ GPX OR COMPRESSED GPX

If the GPX is zipped

- Unzip it to a BLOB
- Convert BLOB to CLOB

Otherwise

- Load the GPX to CLOB directly

```
…
 unzipped blob := empty_blob();
…
 IF v_filename LIKE '%.gz' THEN
    v_unzipped := utl_compress.lz_uncompress(v_file);
    dbms_lob.converttoclob(
      dest_lob     => v_content,    src_blob   => v_unzipped,
      amount       => DBMS_LOB.LOBMAXSIZE,
      dest_offset  => v_dst_offset, src_offset => v_src_offset,
      blob_csid    => dbms_lob.default_csid,
      lang_context => v_lang_ctx,   warning    => v_warning);

  ELSE --ELSIF v_filename LIKE '%.g__' THEN

    DBMS_LOB.LOADCLOBFROMFILE(v_Content,
      Src_bfile    => v_File,
      amount       => DBMS_LOB.LOBMAXSIZE,
      src_offset   => v_src_offset, dest_offset => v_dst_offset,
      bfile_csid   => v_charset_id,
      lang_context => v_lang_ctx,   warning     => v_warning);
  END IF;
…
```

LOAD GPX FILES INTO ACTIVITIES

I *gzip* all the GPX file

- So update the file name

```
UPDATE activities
set filename = filename||'.gz'
where filename like '%.gpx'
/
```

Simply update the GPX column

- function in the PL/SQL package reads and returns GPX content

```
UPDATE activities
SET gpx =
XMLTYPE(strava_pkg.getClobDocument('ACTIVITIES',filename))
…
/
```

QUERY XML

QUERY GPX FROM DATABASE

```
select activity_id, filename, gpx from activities
where activity_id = 4468006769;

ACTIVITY_ID FILENAME                       GPX
----------- ------------------------------ --------------------------------------------------------------------------------
 4468006769 activities/4468006769.gpx.gz   <?xml version="1.0" encoding="US-ASCII"?>
                                           <gpx creator="StravaGPX Android" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLoc
                                           ation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd http://www.garmin
                                           .com/xmlschemas/GpxExtensions/v3 http://www.garmin.com/xmlschemas/GpxExtensionsv3.xsd http://www.gar
                                           min.com/xmlschemas/TrackPointExtension/v1 http://www.garmin.com/xmlschemas/TrackPointExtensionv1.xsd
                                           " version="1.1" xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxtpx="http://www.garmin.com/xmlsch
                                           emas/TrackPointExtension/v1" xmlns:gpxx="http://www.garmin.com/xmlschemas/GpxExtensions/v3">
                                             <metadata>
                                               <time>2020-12-13T14:31:13Z</time>
                                             </metadata>
                                             <trk>
                                               <name>Loop</name>
                                               <type>1</type>
                                               <trkseg>
                                                 <trkpt lat="51.5296380" lon="-0.1875360">
                                                   <ele>30.6</ele>
                                                   <time>2020-12-13T14:31:13Z</time>
                                                   <extensions>
                                                     <gpxtpx:TrackPointExtension>
                                                       <gpxtpx:hr>57</gpxtpx:hr>
                                                     </gpxtpx:TrackPointExtension>
                                                   </extensions>
                                                 </trkpt>
                                                 <trkpt lat="51.5296350" lon="-0.1875340">
```

GPX IS A FORM OF XML

```
<?xml version="1.0" encoding="UTF-8"?>
<gpx creator="StravaGPX Android" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd
http://www.garmin.com/xmlschemas/GpxExtensions/v3 http://www.garmin.com/xmlschemas/GpxExtensionsv3.xsd
http://www.garmin.com/xmlschemas/TrackPointExtension/v1 http://www.garmin.com/xmlschemas/TrackPointExtensionv1.xsd"
version="1.1" xmlns="http://www.topografix.com/GPX/1/1"
xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1"
xmlns:gpxx="http://www.garmin.com/xmlschemas/GpxExtensions/v3">
 <metadata>
  <time>2020-12-13T14:31:13Z</time>
 </metadata>
 <trk>
  <name>Loop</name>
  <type>1</type>
  <trkseg>
   <trkpt lat="51.5296380" lon="-0.1875360">
    <ele>30.6</ele>
    <time>2020-12-13T14:31:13Z</time>
    <extensions>
     <gpxtpx:TrackPointExtension>
      <gpxtpx:hr>57</gpxtpx:hr>
     </gpxtpx:TrackPointExtension>
    </extensions>
   </trkpt>
...
```

REGISTER XML SCHEMA DEFINIITON

GPX is a form of XML

- Schema Definition
  - Topografix.com
    - GPX1.0 – 2002
    - GPX1.1 – 2004
  - Schema Extension
    - Garmin
- Heart Rate, Cadence, Temperature

```
cd /tmp/strava
wget http://www.topografix.com/GPX/1/0/gpx.xsd --output-document=gpx0.xsd
wget http://www.topografix.com/GPX/1/1/gpx.xsd
wget https://www8.garmin.com/xmlschemas/TrackPointExtensionv1.xsd
```

```
ls -l *.xsd
[oracle@oracle-database strava]$ ls -l *.xsd
-rw-r--r--. 1 oracle oinstall 12572 Jul 11  2017 gpx0.xsd
-rw-r--r--. 1 oracle oinstall 26665 Jul 11  2017 gpx.xsd
-rw-r--r--. 1 oracle oinstall  3134 Apr 25  2017 TrackPointExtensionv1.xsd
```

REGISTER XML SCHEMA DEFINIITON

```
-- Create some useful variable for use in the whole script
var schemaURL varchar2(256)
var schemaDoc varchar2(256)

begin
  :schemaURL := 'http://www.topografix.com/GPX/1/0/gpx.xsd';
  :schemaDoc := 'gpx0.xsd';
end;
/

SELECT getCLOBDocument('STRAVA',:schemaDoc) FROM dual
/

declare
  xmlSchema xmlType;
  res       boolean;
begin
  xmlSchema := XMLTYPE(getCLOBDocument('STRAVA',:schemaDoc,'AL32UTF8'));
  if (dbms_xdb.existsResource(:schemaDoc)) then
      dbms_xdb.deleteResource(:schemaDoc);
  end if;
  res := dbms_xdb.createResource(:schemaDoc,xmlSchema);
end;
/

-- May not exist, but attempt to delete anyway.
execute dbms_xmlSchema.deleteSchema(:schemaURL,DBMS_XMLSCHEMA.DELETE_CASCADE_FORCE);

-- Now reregister the schema
begin
  dbms_xmlschema.registerSchema
  (
    :schemaURL,
    XMLTYPE(getCLOBDocument('STRAVA',:schemaDoc,'AL32UTF8')),
    TRUE,TRUE,FALSE,FALSE
  );
End;
/
```

EXTRACTING POINTS FROM GPX

```
SELECT g.activity_id
,      EXTRACTVALUE(VALUE(t), 'trkpt/time') time_string
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lat')) lat
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lon')) lng
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/ele')) ele
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/extensions/gpxtpx:TrackPointExtension/gpxtpx:hr'
       ,'xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1"')) hr
 FROM activities g,
      TABLE(XMLSEQUENCE(extract(g.gpx,'/gpx/trk/trkseg/trkpt'
      ,'xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1"'
      ))) t
Where  activity_id IN(4468006769)
fetch first 10 rows only
/

ACTIVITY_ID TIME_STRING                LAT        LNG        ELE         HR
----------- -------------------- ---------- ---------- ---------- ----------
 4468006769 2020-12-13T14:31:13Z  51.529638  -0.187536       30.6         57
 4468006769 2020-12-13T14:31:14Z  51.529635  -0.187534       30.6         57
 4468006769 2020-12-13T14:31:15Z  51.529641  -0.187531       30.6         57
 4468006769 2020-12-13T14:31:16Z   51.52964  -0.187529       30.6         57
 4468006769 2020-12-13T14:31:17Z  51.529636  -0.187527       30.6         57
 4468006769 2020-12-13T14:31:18Z  51.529632  -0.187527       30.6         57
 4468006769 2020-12-13T14:31:19Z  51.529629  -0.187528       30.6         57
 4468006769 2020-12-13T14:31:20Z  51.529628  -0.187528       30.6         57
 4468006769 2020-12-13T14:31:21Z  51.529628  -0.187529       30.6         57
 4468006769 2020-12-13T14:31:22Z  51.529628   -0.18753       30.6         57
```

COORDINATE SYSTEMS

7403 Coordinate Systems

- EPSG Geodetic Parameter Dataset
  - aka. EPSG Registry
- European Petroleum Survey Group
  - Oracle 19c lists 6016 systems

- 4326 – WGS 84
  - World Geodetic System 1984
  - GPS data
- 27700 - OSGB 1936
  - British National Grid
  - Ordnance Survey Mapping
  - UK Government Data

```
select srid, coord_ref_sys_name, coord_ref_sys_kind,
legacy_cs_bounds
from SDO_COORD_REF_SYS
where srid IN(4326, 27700)
/

 SRID COORD_REF_SYS_NAME                  COORD_REF_SYS_KIND
---------- -------------------------------- ------------------
LEGACY_CS_BOUNDS
-------------------------------------------------------------------
      4326 WGS 84                              GEOGRAPHIC2D
SDO_GEOMETRY(2003, 4326, NULL, SDO_ELEM_INFO_ARRAY(1, 1003, 3),
SDO_ORDINATE_ARRAY(-180, -90, 180, 90))

     27700 OSGB 1936 / British National Grid   PROJECTED
```

FUNCTION TO MAKE A SPATIAL POINT GEOMETRY

Process individual GPS track points

Necessary if I want to process other non-spatial attributes

- Time
- Elevation
- HR

```
…
function make_point
(longitude in number, latitude  in number)
return sdo_geometry deterministic is
begin
  if longitude is not null and latitude is not null then
    return
      sdo_geometry (2001, 4326,
        sdo_point_type (longitude, latitude, null),
        null, null);
  else
    return null;
  end if;
end make_point;
…
```

EXTRACT TRACKPOINTS TO TABLE

```
INSERT /*+APPEND*/ INTO trackpoints (activity_id, seq, time, lat, lng, ele, hr, loc)
WITH a as (
SELECT g.activity_id
,      EXTRACTVALUE(VALUE(t), 'trkpt/time') time_string
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lat')) lat
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lon')) lng
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/ele')) ele
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/extensions/gpxtpx:TrackPointExtension/gpxtpx:hr'
            ,'xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1"')) hr
 FROM activities g,
      TABLE(XMLSEQUENCE(extract(g.gpx,'/gpx/trk/trkseg/trkpt'
            ,'xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1"'
      ))) t
Where  activity_date >= TO_DATE('01122020','DDMMYYYY')
)
Select activity_id
,      row_number() over (partition by activity_id order by time_string) seq
,      CASE WHEN time_string LIKE '____-__-__T__:__:__Z'
                THEN TO_DATE(time_string,'YYYY-MM-DD"T"HH24:MI:SS"Z"')
           WHEN time_string like '____-__-__T__:__:__+__:__'
                THEN cast(to_timestamp_tz(time_string,'yyyy-mm-dd"T"hh24:mi:ss"+"TZH:TZM') as date)
       END time
,      lat, lng, ele, hr
,      strava_pkg.make_point(lng,lat) loc
From   a;
```

WHICH GPX NAMESPACE

Need to determine which versions of the Topographic schema has been used to encode a GPX.

```
…
 UPDATE activities
 SET    gpx = XMLTYPE(l_gpx), geom = null, geom_27700 = null, num_pts = 0, xmlns = NULL
 WHERE  activity_id = p_activity_id
 RETURNING extractvalue(gpx,'/gpx/@creator', 'xmlns="http://www.topografix.com/GPX/1/0"')
 ,         extractvalue(gpx,'/gpx/@creator', 'xmlns="http://www.topografix.com/GPX/1/1"')
 INTO      l_xmlns0, l_xmlns1;
…
```

CONVERT TO SPATIAL GEOMETRY

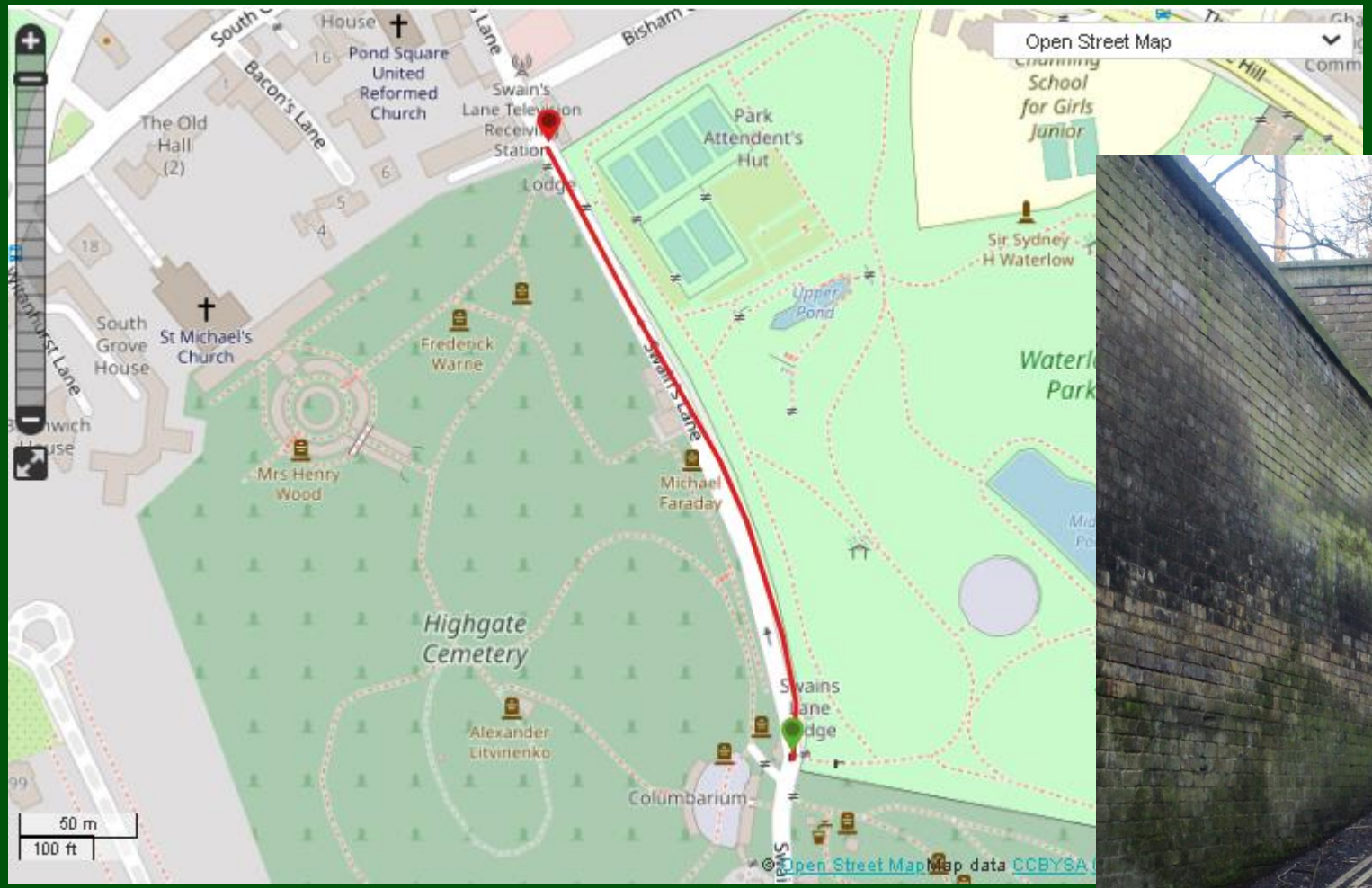
Extract points from GPX using correct namespace
Convert them to a multiset.
Cast them as a spatial ordinate array

```
…
 UPDATE activities a
    SET geom = mdsys.sdo_geometry(2002,4326,null,mdsys.sdo_elem_info_array(1,2,1),
    cast(multiset(
      select CASE n.rn WHEN 1 THEN pt.lng WHEN 2 THEN pt.lat END ord
      from (
        SELECT rownum rn
        ,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lon')) as lng
        ,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lat')) as lat
        FROM   TABLE(XMLSEQUENCE(extract(a.gpx,'/gpx/trk/trkseg/trkpt', 'xmlns="http://www.topografix.com/GPX/1/1"'))) t
        ) pt,
        (select 1 rn from dual union all select 2 from dual) n
        order by pt.rn, n.rn
      ) AS mdsys.sdo_ordinate_array))
    , xmlns = 'xmlns="http://www.topografix.com/GPX/1/1"'
    WHERE  a.gpx IS NOT NULL
    And    activity_id = p_activity_id;
…
```

DEFINE BOUNDS FOR INDEXED GEOMETRIES

```
insert into user_sdo_geom_metadata (table_name,column_name,diminfo,srid)
values (
  'MY_GEOMETRIES' , 'GEOM_27700',
  sdo_dim_array(
    sdo_dim_element('Easting',-1000000,1500000,0.05),
    sdo_dim_element('Northing', -500000,2000000,0.05)
  ),
  27700
);

insert into user_sdo_geom_metadata (table_name,column_name,diminfo,srid)
values (
  'MY_GEOMETRIES' , 'GEOM',
  sdo_dim_array(
    sdo_dim_element('Longitude',-180,180,0.05),
    sdo_dim_element('Latgitude',-90,90,0.05)
  ),
  4326
);

CREATE INDEX my_geometries_geom ON my_geometries (geom) INDEXTYPE IS MDSYS.SPATIAL_INDEX_v2;
CREATE INDEX my_geometries_geom_27700 ON my_geometries (geom_27700) INDEXTYPE IS MDSYS.SPATIAL_INDEX_v2;
```

SWAIN'S LANE, HIGHGATE



GPX ROUTE

```
<?xml version="1.0" encoding="UTF-8"?>
<gpx xmlns="http://www.topografix.com/GPX/1/1" version="1.1" creator="ViewRanger -
//www.viewranger.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/GPX/1/1/gpx.xsd">
  <rte>
    <name><![CDATA[Swain's World]]></name>
    <rtept lat="51.567076879686" lon="-0.146511140744 74"></rtept>
    <rtept lat="51.567297957247" lon="-0.146492365281 62"></rtept>
    <rtept lat="51.567563048851" lon="-0.146578195970 11"></rtept>
    <rtept lat="51.568016535308" lon="-0.146806183736 36"></rtept>
    <rtept lat="51.568253280116" lon="-0.146975162904 3"></rtept>
    <rtept lat="51.568870144602" lon="-0.147543791215 45"></rtept>
    <rtept lat="51.569488667885" lon="-0.148072186391 39"></rtept>
  </rte>
</gpx>
```

LOAD GPX ROUTE INTO TABLE

```
INSERT INTO my_geometries (geom_id, descr, gpx)
VALUES (2,'Swains World Route', XMLTYPE(getClobDocument('STRAVA','swainsworldroute.gpx')));

UPDATE my_geometries
SET geom = mdsys.sdo_geometry(2002,4326,null,mdsys.sdo_elem_info_array(1,2,1),
cast(multiset(
  select CASE n.rn WHEN 1 THEN pt.lng WHEN 2 THEN pt.lat END ord
  from (
    SELECT /*+MATERIALIZE*/ rownum rn
    ,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'rtept/@lon')) as lng
    ,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'rtept/@lat')) as lat
    FROM   my_geometries g,
           TABLE(XMLSEQUENCE(extract(g.gpx,'/gpx/rte/rtept','xmlns="http://www.topografix.com/GPX/1/1"'))) t
    where g.geom_id = 2
    ) pt,
    (select 1 rn from dual union all select 2 from dual) n order by pt.rn, n.rn) AS mdsys.sdo_ordinate_array
))
WHERE gpx IS NOT NULL
AND   geom IS NULL
/
UPDATE my_geometries
SET mbr = sdo_geom.sdo_mbr(geom)
,   geom_27700 = sdo_cs.transform(geom,27700)
/
```

WHICH ACTIVITIES GO UP SWAIN'S LANE?

```
With a as (
SELECT a.activity_id, a.activity_date, a.activity_name
,      a.distance km
,      SDO_GEOM.RELATE(a.geom,'anyinteract',g.geom,25) geom_relate
FROM   activities a
,      my_geometries g
WHERE  a.activity_type = 'Ride'
And    a.activity_date >= TO_DATE('01122020','DDMMYYYY')
and    g.geom_id = 2 /*Swains World Route*/
)
Select *
From   a
Where  geom_relate = 'TRUE'
Order by activity_date
```

Activity ID	ACTIVITY_DATE	ACTIVITY_NAME	Distance Km	geom relate
4419821750	08:44:45 02.12.2020	Loop	39.57	TRUE
4428307816	10:49:25 04.12.2020	Loop	24.68	TRUE
4431920358	09:41:13 05.12.2020	Loop	24.87	TRUE
4441231595	09:27:55 07.12.2020	Loop	25.08	TRUE
4445171764	09:23:47 08.12.2020	Loop	24.79	TRUE
4449695745	09:33:03 09.12.2020	Loop	35.04	TRUE
4454062323	10:53:14 10.12.2020	Loop	24.53	TRUE
4457817695	10:05:05 11.12.2020	Loop	25.19	TRUE
4461572963	09:27:08 12.12.2020	Loop	23.21	TRUE
4468006769	14:31:13 13.12.2020	Loop	23.25	TRUE

…

YOU CANNOT PASS A SPATIAL GEOMETRY TO AN ANALYTIC FUNCTION

```
with x as (
SELECT activity_id
,      TO_DATE(EXTRACTVALUE(VALUE(t), 'trkpt/time'),'YYYY-MM-DD"T"HH24:MI:SS"Z"') time
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lat')) lat
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lon')) lng
FROM   activities a,

TABLE(XMLSEQUENCE(extract(a.gpx,'/gpx/trk/trkseg/trkpt','xmlns="http://www.topografix.com/GPX/1/1"'))) t
WHERE  a.activity_id IN(4468006769)
), y as (
select x.*, strava_pkg.make_point(lng,lat) loc
from x
)
select lag(loc,1) over (partition by activity_id order by time) last_loc
from   y
/
```

```
select lag(loc,1) over (partition by activity_id order by time) last_loc
           *
ERROR at line 13:
ORA-22901: cannot compare VARRAY or LOB attributes of an object type
```

LOOK AT EACH ASCENT OF SWAIN'S LANE IN ONE ACTIVITY (1)

```
WITH … a as ( /*extract all points in activity*/
SELECT a.activity_id, g.geom g_geom, g.tol, g.geom_length
,      TO_DATE(EXTRACTVALUE(VALUE(t), 'trkpt/time'),'YYYY-MM-DD"T"HH24:MI:SS"Z"') time
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lat')) lat
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/@lon')) lng
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/ele')) ele
,      TO_NUMBER(EXTRACTVALUE(VALUE(t), 'trkpt/extensions/gpxtpx:TrackPointExtension/gpxtpx:hr'
       ,'xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1"')) hr
FROM   activities a,
       geo g,
       TABLE(XMLSEQUENCE(extract(a.gpx,'/gpx/trk/trkseg/trkpt'
       ,'xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1"'))) t
Where  a.activity_type = 'Ride'
And    a.activity_id IN(4468006769)
and    SDO_GEOM.RELATE(a.geom,'anyinteract',g.geom,g.tol) = 'TRUE' /*activity has relation to reference geometry*/
), b as ( /*smooth elevation*/
Select a.*
,      avg(ele) over (partition by activity_id order by time rows between 2 preceding and 2 following) avg_ele
From   a
), c as ( /*last point*/
Select b.*
,      row_number() over (partition by activity_id order by time) seq
,      lag(time,1) over (partition by activity_id order by time) last_time
,      lag(lat,1) over (partition by activity_id order by time) last_lat
,      lag(lng,1) over (partition by activity_id order by time) last_lng
,      lag(avg_ele,1) over (partition by activity_id order by time) last_avg_ele
From   b
)…
```

LOOK AT EACH ASCENT OF SWAIN'S LANE IN ONE ACTIVITY (2)

```
), d as ( /*make points*/
SELECT c.*
,      strava_pkg.make_point(lng,lat) loc
,      strava_pkg.make_point(last_lng,last_lat) last_loc
FROM   c
), e as ( /*determine whether point is inside the polygon*/
select d.*
,      86400*(time-last_time) secs
,      avg_ele-last_avg_ele ele_diff
,      sdo_geom.sdo_distance(loc,last_loc,0.05,'unit=m') dist
,      SDO_GEOM.RELATE(loc,'anyinteract', g_geom, tol) sdo_relate
FROM   d
), f as (
select e.*
,      CASE WHEN sdo_relate != lag(sdo_relate,1) over (partition by activity_id order by time) THEN 1 END sdo_diff
from   e
), g as (
select f.*
,      SUM(sdo_diff) over (partition by activity_id order by time range between unbounded preceding and current row) sdo_seq
from f
where  sdo_relate = 'TRUE'
)…
```

LOOK AT EACH ASCENT OF SWAIN'S LANE IN ONE ACTIVITY (3)

```
…
select activity_id, min(time), max(time)
, sum(dist)/1000 sum_dist
, sum(secs) sum_secs
, 3.6*sum(dist)/sum(secs) avg_speed
, sum(greatest(0,ele_diff)) ele_gain
, sum(least(0,ele_diff)) ele_loss
, 100*sum(ele_diff*dist)/sum(dist*dist) avg_grade
, min(ele) min_ele
, max(ele) max_ele
, sum(hr*secs)/sum(secs) avg_Hr
, max(hr) max_hr
, count(*) num_pts
from   g
group by activity_id, sdo_seq, g.geom_length
having sum(dist)>= g.geom_length/2 /*make sure line we find is longer than half route to prevent fragmentation*/
order by 2
/
```

Activity ID	MIN(TIME)	MAX(TIME)	Dist. (km)	Secs	Avg Speed (kph)	Ele Gain (m)	Ele Loss (m)	Avg Grade %	Min Ele (m)	Max Ele (m)	Avg HR	Max HR	Num Pts
4468006769	14:55:51 13.12.2020	14:58:17 13.12.2020	.372	147	9.1	36.1	.0	8.6	86.8	122.7	141	153	147
	15:08:13 13.12.2020	15:10:28 13.12.2020	.374	136	9.9	36.2	.0	8.4	86.8	122.8	147	155	136
	15:22:49 13.12.2020	15:25:18 13.12.2020	.369	150	8.9	36.2	.0	8.2	86.8	122.7	147	155	150
**********			--------	-----									
sum			1.116	433									

COMMENTS

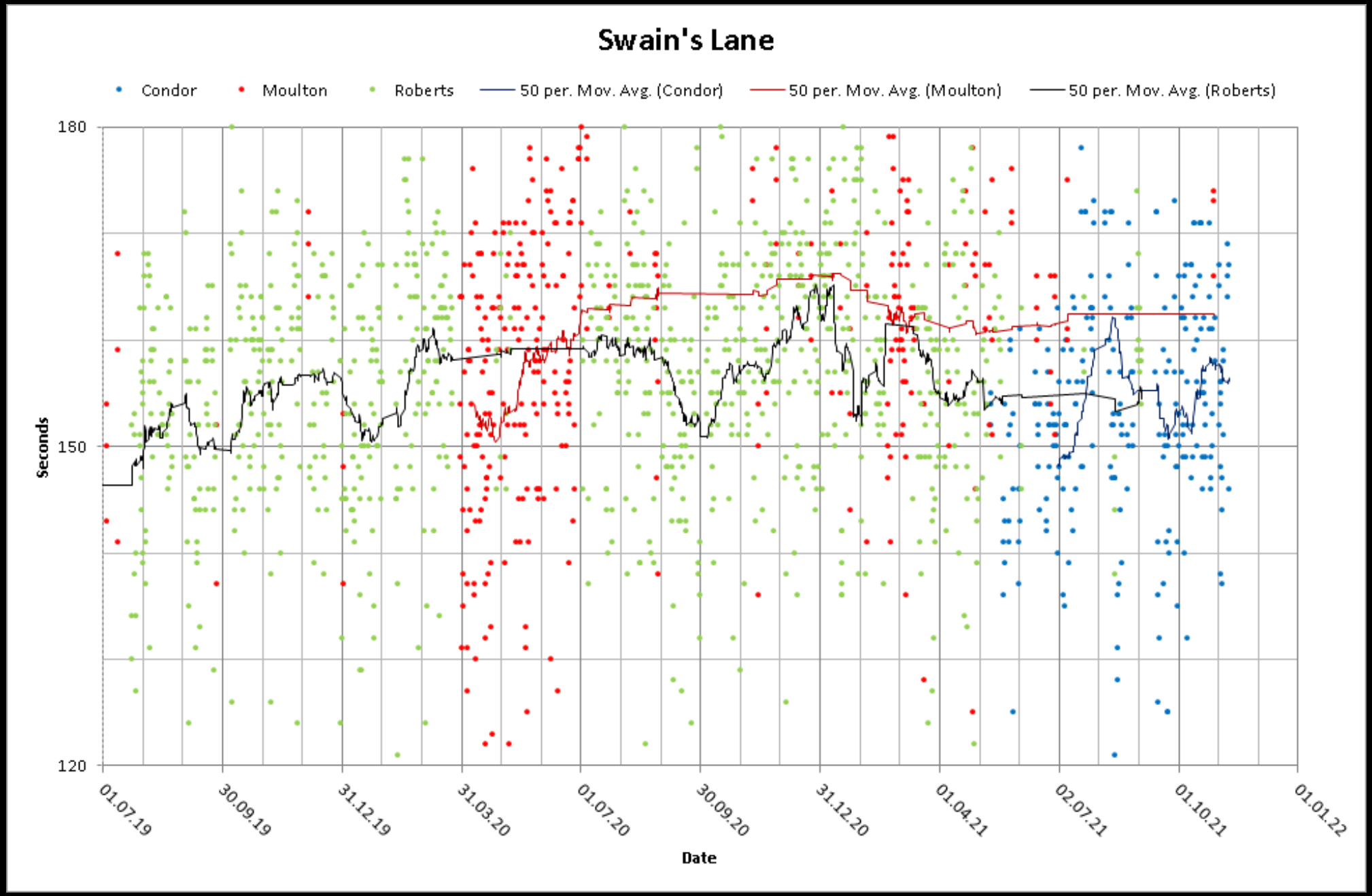
This turned out to be quite CPU intensive

To process one activity took 14 seconds (on my laptop)

Most of the time is on the analytic functions rather than the spatial operators.

To process many activities

- 591 activities with 1729 ascents of Swain's Lane
- I found it easier to build up results in a reporting table.


Swain's Lane
Condor
Moulton
Roberts
50 per. Mov. Avg. (Condor)
50 per. Mov. Avg. (Moulton)
50 per. Mov. Avg. (Roberts)
180
150
120
Seconds
01.07.19
30.09.19
31.12.19
31.03.20
01.07.20
30.09.20
31.12.20
01.04.21
02.07.21
01.10.21
01.01.22
Date

COMPARING GEOMETRIES

SDO_UTL.SIMPLIFY

"Simplifies the input geometry, based on a threshold value, using the Douglas-Peucker algorithm."

- Tolerance 1m
- Wikipedia: Ramer–Douglas–Peucker algorithm
- Iterative end-point fit algorithm that 'decimates a curve composed of line segments to a similar curve with fewer points'

```
…
 UPDATE activities
    SET    geom = sdo_util.simplify(geom,1)
    WHERE  geom IS NOT NULL
    And    activity_id = p_activity_id;
…
```

KEEP TRACK OF YOUR COORDINATE SYSTEMS

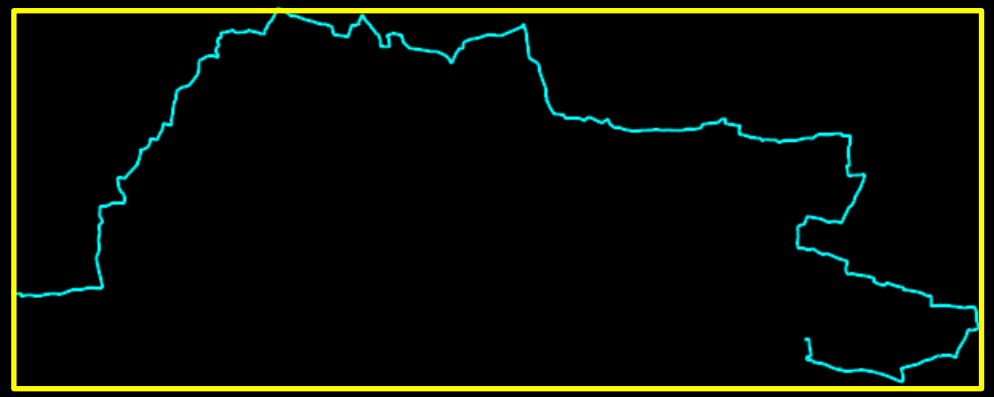
Count the number of points (or vertices) in a geometry.

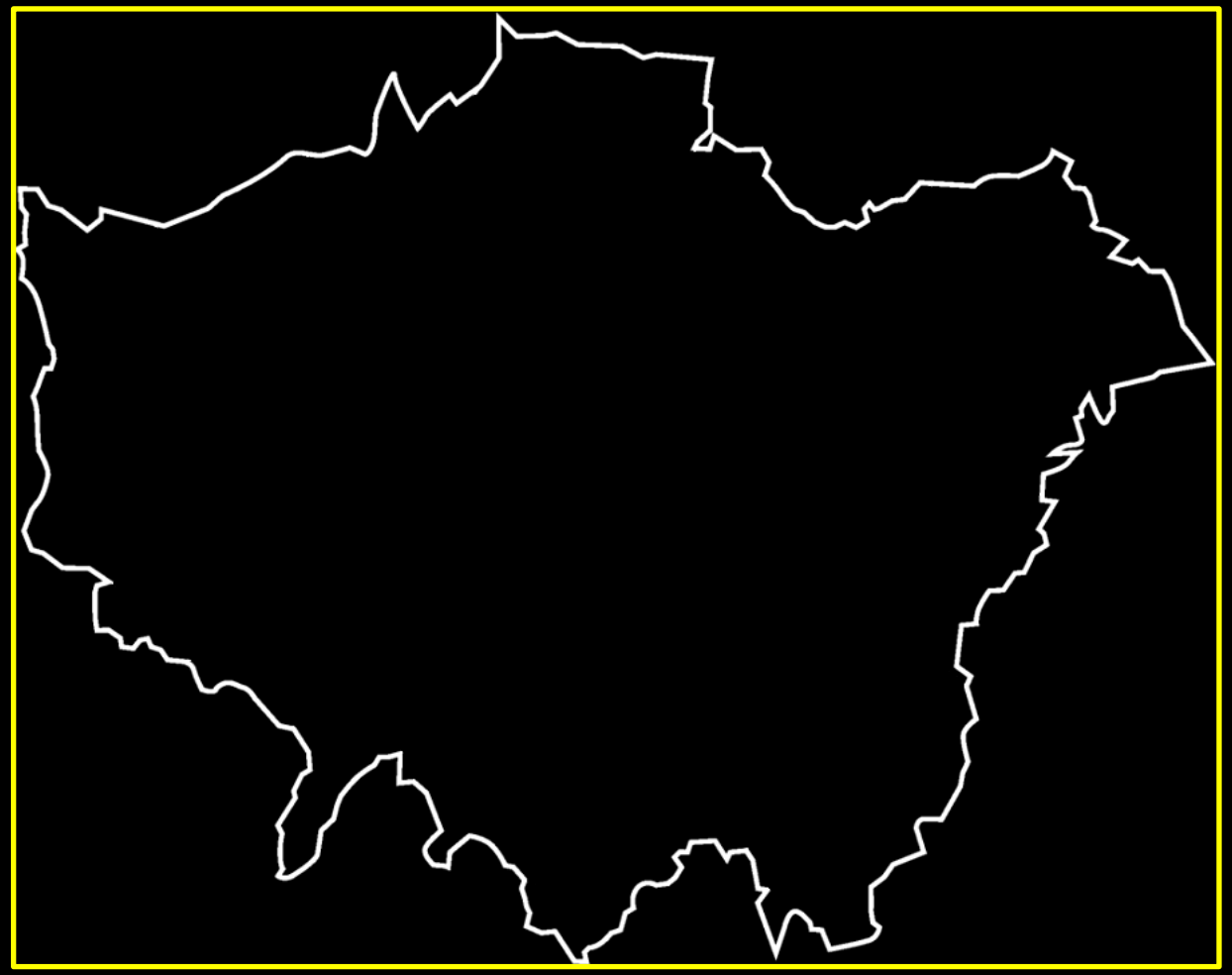
Cannot compare geometries of different co-ordinate systems.

- But you can convert between them
- I am going to store the geometry for each activity in two coordinate systems

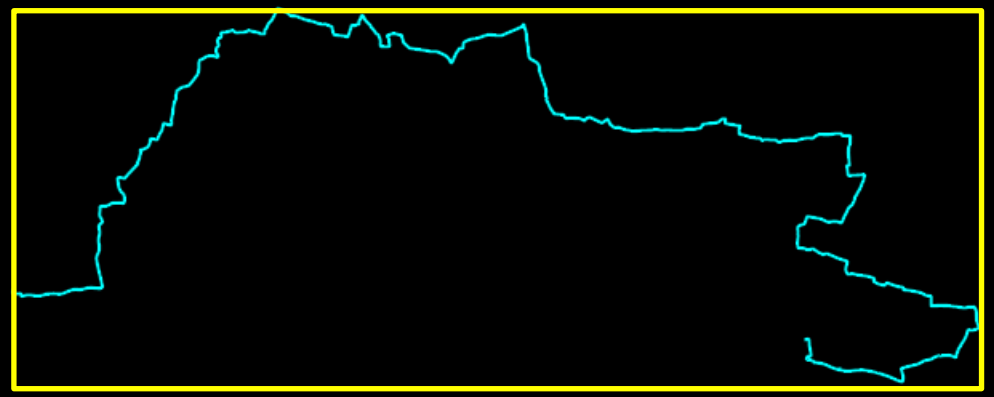
```
…
 UPDATE activities
  SET    num_pts = SDO_UTIL.GETNUMVERTICES(geom)
  ,      geom_27700 = sdo_cs.transform(geom,27700)
…
  WHERE  geom IS NOT NULL
  And    activity_id = p_activity_id
…
```

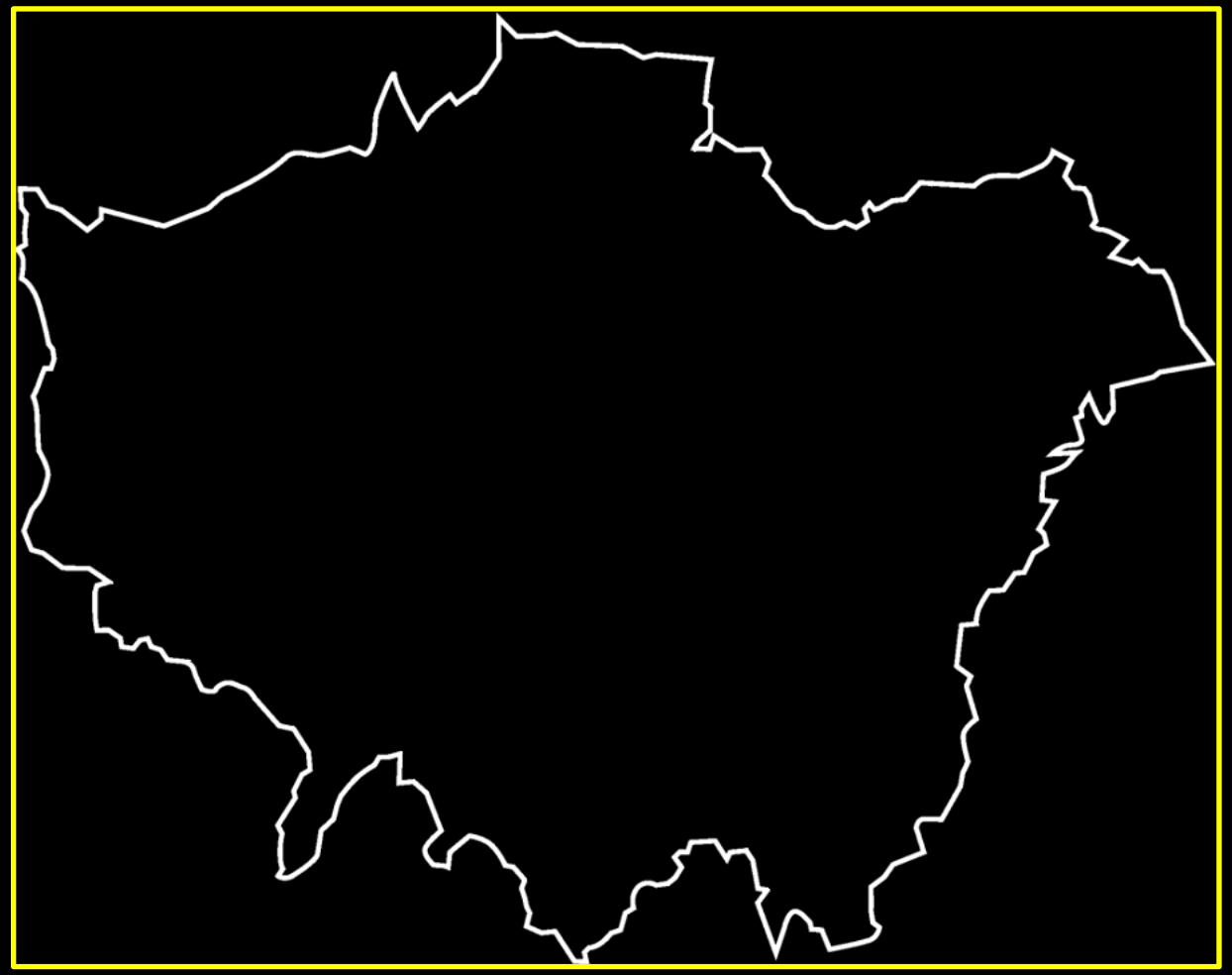
MINIMUM BOUNDED RECTANGLE





MINIMUM BOUNDED RECTANGLE





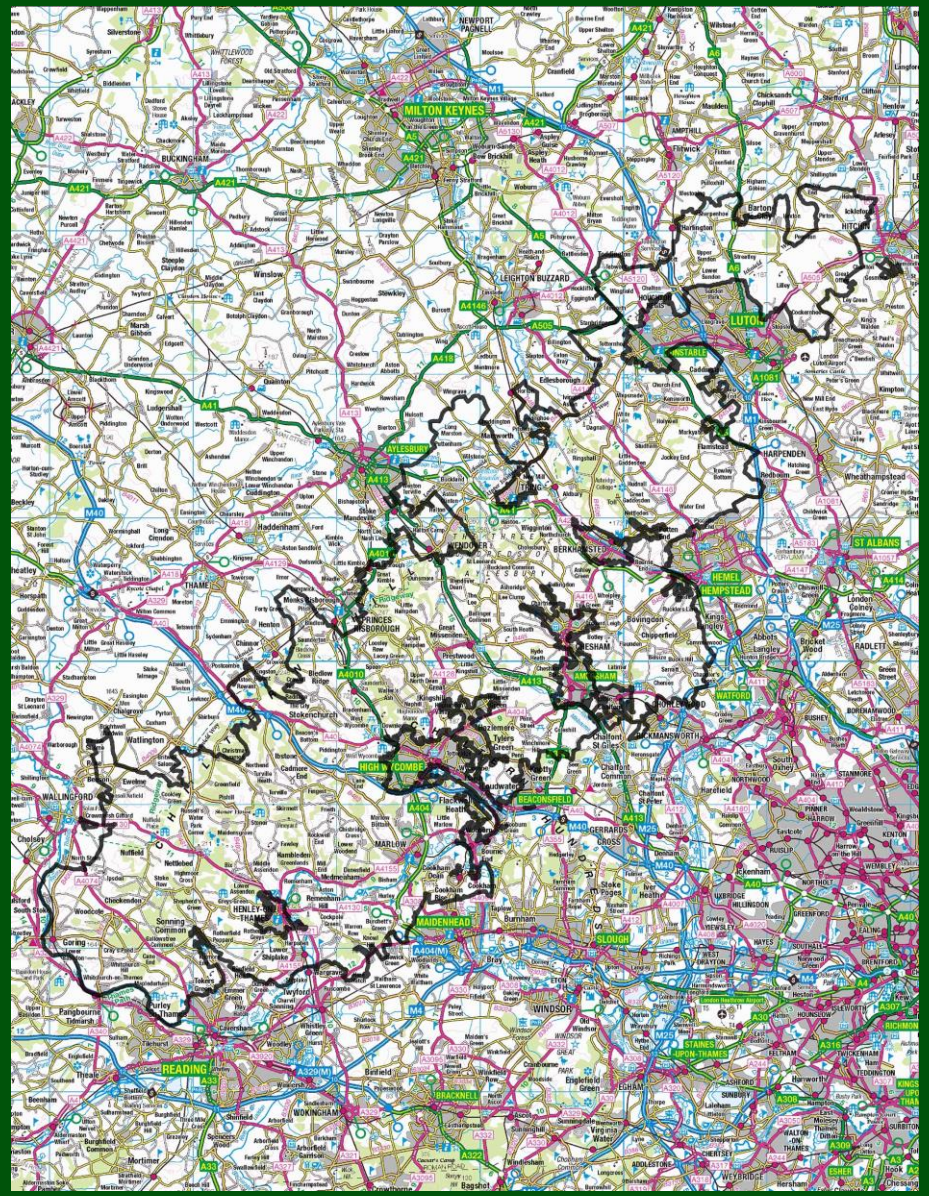
MINIMUM BOUNDED RECTANGLE

Minimum Bounded Rectangle

- Draw a box around your geometry.
  - 5 points (5th point same as 1st to close the box)
- Quick rough comparison of geometries with false positives to save time later not doing a full comparison when not necessary

```
…
 UPDATE activities
  SET     …
  ,       mbr = sdo_geom.sdo_mbr(geom)
  WHERE   geom IS NOT NULL
  And     activity_id = p_activity_id
…
```

CHILTERN AONB



Chilterns Cycleway

OBTAINING GEOGRAPHICAL DATA

- **I want to see previous routes that go to the Chilterns**
  - **So I need a working definition of 'The Chilterns'**
  - **More generically, I need a definition of an area for any place/area name that I might be interested in.**

- **Political Boundaries**
- **Administrative Boundaries**

- Natural Earth: 1:10m cultural vectors - countries/sovereign states
- DIVA-GIS - Administrative areas

The world is divided into 206 sovereignties

- **(including independent and leased areas)**

Different countries use different subdivisions

Netherlands: Province, Municipality

Germany: States, Region, Urban/Rural District

France: Region, Department, Arrondissment, Canton, Commune

OBTAINING UK GEOGRAPHICAL DATA

The United Kingdom consists of the 4 'home' countries.

- **These are divided down into counties, authorities, districts, wards, parishes, Scottish communities**

Guernsey, Jersey and the Isle of Man are "Crown Dependencies".

There are 14 dependent territories

- **Some broken down into separate islands.**

- Ordnance Survey: Boundary-Line - boundaries
- Spatial Hub: Scottish Community Council Boundaries
- DEFRA: Natural England, Areas of Outstanding Natural Beauty (AONB)

United Kingdom

- **England**
- **Northern Ireland**
- **Scotland**
- **Wales**

- **Guernsey**
  - **Alderney**
  - **Guernsey**
  - **Herm**
  - **Sark**
- **Isle of Man**
- **Jersey**

Anguilla
Bermuda
Cayman Islands
Dhekelia Sovereign Base Area
Falkland Islands
Gibraltar
British Indian Ocean Territory
- **Diego Garcia Naval Support Facility**

Montserrat
Pitcairn Islands
South Georgia and the Islands
- **South Georgia**
- **South Sandwich Islands**

Saint Helena
- **Ascension**
- **Saint Helena**
- **Tristan da Cunha**

Turks and Caicos Islands
British Virgin Islands
Akrotiri Sovereign Base Area

ESRI SHAPEFILES

"Esri's somewhat open, hybrid vector data format using SHP, SHX and DBF files. Originally invented in the early 1990s, it is still commonly used as a widely supported interchange format".

- **Oracle provides a java shapefile converter that transforms shapefiles into database tables.**
- **19c Developers Guide: Loading ESRI Shapefiles into Spatial and Graph**

Zip archives including but not limited to, but containing at least always the first 3:

- **.shp - the main file that contains the geometry itself,**
- **.shx - an index file,**
- **.dbf - a DBase file containing other attributes to describe the spatial data. DBF file contains all the other columns in the table. Can be opened with Microsoft Excel, so you can see the data.**
- .prj - contains the projection description of the data,
- .csv -the same data as in the .dbf file, but as a comma-separated data file,
- .cpg - the code page of the data in the .dbf file.

ORACLE SHAPEFILE CONVERTER

Java Utility to import spatial geometries and other related data.

- **Each ESRI archive will create a different table**
- **I treat that as a staging table, and I will copy them into a consolidated table of areas**

- **Most ESRI archives seem to be defined with longitude/latitude coordinates – i.e. WGS84, SRID 4326**
  - **UK government use OSGB36, SRID 27700**
- **The converter will also convert data between SRIDs, but that seems to slow.**
  - **Faster to load delivered SRID, and convert it in SQL later.**

- load_countries.sql - all countries
- load_XXX.sql, - load administrative areas for a country where XXX is the 3-letter ISO code for that country (eg. load_FRA.sql for France).
- load_uk.sql - OS data
  - Includes DML to determine which wards and parishes are in which districts and boroughs.

- fix_names.sql - to simplify names
  - stripping common suffixes.
- fix_my_areas.sql - collect statistics, count children, look for children with same name, simplify areas > 10,000 points.

HIERARCHY OF AREAS

Most political areas fit into a hierarchy.

- Country → Country/Authority → District → Ward / Parish
- Some of that hierarchy is in the non-spatial data
  - But sometimes I have had to work it out with spatial comparisons
  - E.g. Berkshire is now a ceremonial country – since split into 6 unitary authorities
  - It is convenient to use it as a shorthand, and so fit it into the hierarchy.

Ultimately

- Search for an area by name
- Find the activities that enter that area

Search hierarchically

- Start with root areas (sovereignties)
- Recursively search children of matches

PREPROCESSING

But even so it is slow

- **Too slow for an on-line query tool**

My Spatial Data is static

- **Activities don’t change**
- **Areas only change occasionally**
  - **UK boundary reorganisations 1972, 1992**

Pre-process search results

- **Every area for every activity**
- **81810 areas for 3227 acitivities**

62682 areas

- **France 40825**
- **UK 20328**
- **NL 511**
- **Germany 493**

RECURSIVE SERARCH

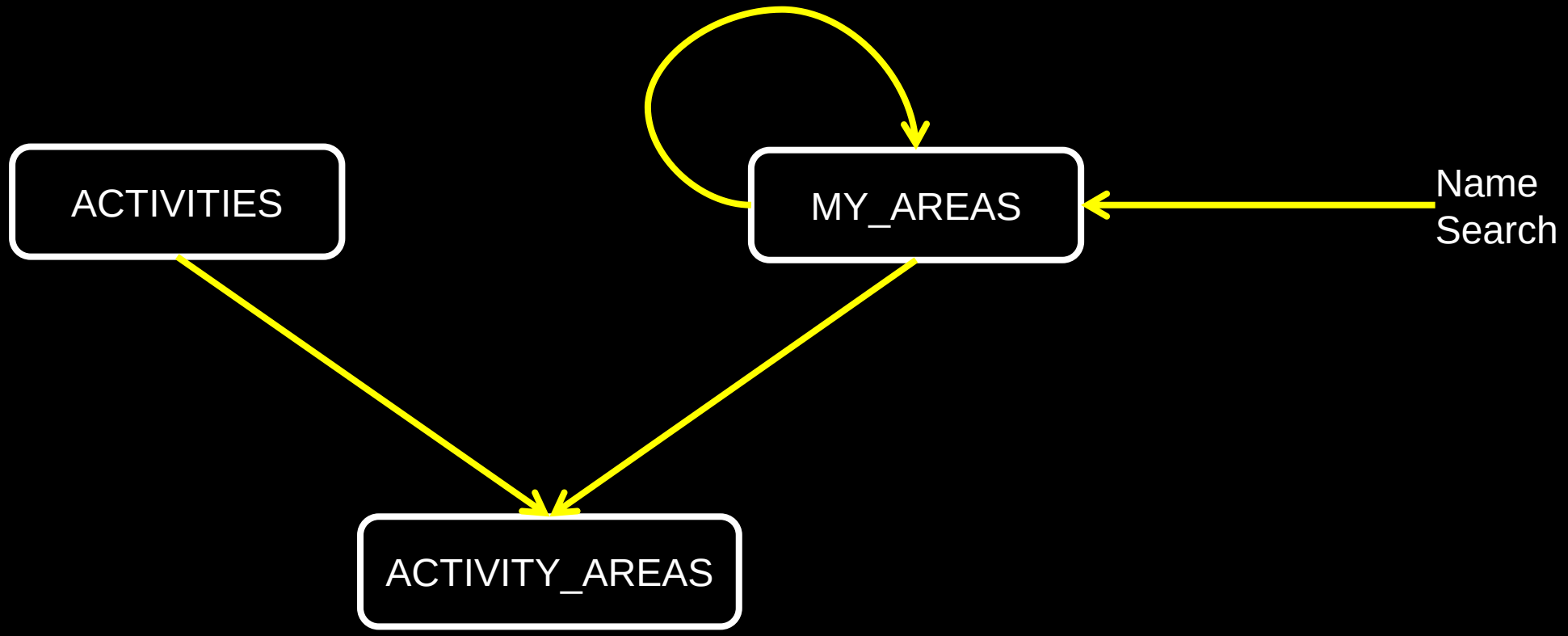
```
PROCEDURE activity_area_search
(p_activity_id INTEGER
,p_area_code   my_areas.area_code%TYPE   DEFAULT NULL
,p_area_number my_areas.area_number%TYPE DEFAULT NULL
,p_query_type VARCHAR2 DEFAULT 'P'
,p_level INTEGER DEFAULT 0) IS
BEGIN
  FOR i IN(
   SELECT m.*
   ,      CASE WHEN m.geom_27700 IS NOT NULL THEN sdo_geom.sdo_length(SDO_GEOM.sdo_intersection(m.geom_27700,a.geom_27700,5), unit=>'unit=km')
               WHEN m.geom       IS NOT NULL THEN sdo_geom.sdo_length(SDO_GEOM.sdo_intersection(m.geom,a.geom,5), unit=>'unit=km') END geom_length
   FROM   my_areas m, activities a
   WHERE  (  (p_query_type = 'P' AND parent_area_code = p_area_code AND parent_area_number = p_area_number)
          OR (p_query_type = 'A' AND area_code        = p_area_code AND area_number        = p_area_number)…
          OR (p_area_code IS NULL AND p_area_number IS NULL AND parent_area_code IS NULL AND parent_area_number IS NULL))
   AND    a.activity_id = p_activity_id
   and    SDO_GEOM.RELATE(a.mbr,'anyinteract',m.mbr) = 'TRUE'
   and    SDO_GEOM.RELATE(a.geom,'anyinteract',m.geom) = 'TRUE'
  ) LOOP
    IF i.area_level>0 OR i.num_children IS NULL THEN
      BEGIN
        INSERT INTO activity_areas (activity_id, area_code, area_number, geom_length)
        VALUES                      (p_activity_id, i.area_code, i.area_number, i.geom_length);
      EXCEPTION
        WHEN dup_val_on_index THEN
          UPDATE activity_areas
          SET    geom_length = i.geom_length
          WHERE  activity_id = p_activity_id AND area_code = i.area_code AND area_number = i.area_number;
      END;
    END IF;

    IF i.num_children > 0 THEN
      strava_pkg.activity_area_search(p_activity_id, i.area_code, i.area_number, 'P', p_level+1);
    END IF;
  END LOOP;
END activity_area_search;
```

RECURSIVE SEARCH FOR ONE ACTIVITY

```
Searching 4372796838:-
Found SOV-1159320701:United Kingdom,    2.895 km
.Found GEOU-1159320743:England,    2.851 km
..Searching 4372796838:GEOU-1159320743
..Found GLA-117537:Greater London,    2.851 km
...Found LBO-50724:City of Westminster,    1.732 km
....Found LBW-117484:Abbey Road,    1.435 km
....Found LBW-50639:Maida Vale,    0.298 km
...Found LBO-50632:Camden,    1.119 km
....Searching 4372796838:LBO-50632
....Found LBW-117286:Kilburn,    0.273 km
....Found LBW-117288:Swiss Cottage,    1.033 km
....Found LBW-117287:West Hampstead,    0.084 km
...Done 4372796838:GLA-117537:    3.894 secs).
..Done 4372796838:GEOU-1159320743:    5.133 secs).
.Done 4372796838:SOV-1159320701:    5.579 secs).
Done 4372796838:-:    6.626 secs).
```

TEXT SEARCHING

TEXT SEARCH AREAS BY NAME



NAME SEARCH

“Streatley%”
- **Berkshire?**
- **Bedfordshire?**

“%Twyford%”
- **Twyford, Wokingham**
- **Colden Common & Twyford, Winchester**
- **And 6 others**

“%York%”
- **North/South/West Yorkshire, England**
- **East Riding of Yorkshire**
- **York, England**
- **Rural West York, York, England**
- **New York, United States of America**
- **Anderston/City/Yorkhill, Glasgow City, Scotland**
- **Yorkhill and Kelvingrove, Glasgow City, Scotland**

Need hierarchy
- **Streatley, West Berkshire, England, United Kingdom**
- **Streatley, Central Bedfordshire, England, United Kingdom**

Text Index on the hierarchy

WALK UP HIERARCHY TO ROOT BUILD PATH

```
FUNCTION name_hierarchy_fn
(p_area_code   my_areas.area_code%TYPE DEFAULT NULL
,p_area_number my_areas.area_number%TYPE DEFAULT NULL
,p_type VARCHAR2 DEFAULT 'C' /*(C)umulative, (R)oot*/
) RETURN CLOB DETERMINISTIC is
  l_name_hierarchy CLOB;
  l_last_name my_areas.name%TYPE := '';
  l_count INTEGER := 0;
BEGIN
  FOR i IN (
    SELECT area_code, area_number, name, matchable
    FROM   my_areas m
    START WITH area_code = p_area_code AND area_number = p_area_number
    CONNECT BY NOCYCLE prior m.parent_area_code   = m.area_code
                   AND prior m.parent_area_number = m.area_number
  ) LOOP
    IF i.matchable >= 1 THEN
      l_count := l_count + 1;
      IF l_count > 1 AND p_type = 'C' THEN
        IF i.name != l_last_name THEN --supress repeated names
          l_name_hierarchy := l_name_hierarchy ||', '|| i.name;
        END IF;
      ELSE
        l_name_hierarchy := i.name;
     END IF;
     l_last_name := i.name;
    END IF;
  END LOOP;
RETURN l_name_hierarchy;
END name_hierarchy_fn;
```

OPTION 1: STORE HIERARCHY ON TABLE, CREATE A MULTI-COLUMN TEXT INDEX

To avoid mutation error

```
ALTER TABLE my_areas add name_hierarchy VARCHAR(4000)
/
CREATE GLOBAL TEMPORARY TABLE my_areas_temp ON COMMIT PRESERVE ROWS AS
SELECT area_code, area_number
,      strava_pkg.name_hierarchy_fn(parent_area_code,parent_area_number) name_hierarchy
FROM my_areas WHERE parent_area_code IS NOT NULL AND parent_area_number IS NOT NULL
/

MERGE INTO my_areas u
USING (SELECT * FROM my_areas_temp) s
ON (u.area_code = s.area_code AND u.area_number = s.area_number
AND (u.name_hierarchy IS NULL OR u.name_hierarchy != s.name_hierarchy))
WHEN MATCHED THEN UPDATE
SET u.name_hierarchy = s.name_hierarchy
/
```

OPTION 1: STORE HIERARCHY ON TABLE, CREATE A MULTI-COLUMN TEXT INDEX

Create Multi-column text index

```
begin
 ctx_ddl.create_preference('my_areas_lexer', 'BASIC_LEXER');
 ctx_ddl.set_attribute('my_areas_lexer', 'mixed_case', 'NO');
 ctx_ddl.create_preference('my_areas_datastore', 'MULTI_COLUMN_DATASTORE');
 ctx_ddl.set_attribute('my_areas_datastore', 'columns', 'name, name_hierarchy');
end;
/
CREATE INDEX my_areas_name_txtidx ON my_areas (name) INDEXTYPE IS ctxsys.context
PARAMETERS ('datastore my_areas_datastore lexer my_areas_lexer sync(on commit)')
/
```

OPTION 2: INDEX A *USER_DATASTORE* BASED ON THE RESULT OF A PL/SQL FUNCTION

Created a procedure that returns a string containing the hierarchy of a given area, and then create a text index using that.

The format of the parameters must be exactly as follows:

- **The rowid of the row being indexed is passed to the procedure;**
- **The string to be indexed is passed back as a CLOB parameter.**

OPTION 2: INDEX A *USER_DATASTORE* BASED ON THE RESULT OF A PL/SQL FUNCTION

```
PROCEDURE name_hierarchy_txtidx
(p_rowid   in rowid
,p_dataout IN OUT NOCOPY CLOB
) IS
  l_count INTEGER := 0;
BEGIN
  FOR i IN (
    SELECT area_code, area_number, name, matchable
    FROM   my_areas m
    START WITH rowid = p_rowid
    CONNECT BY NOCYCLE prior m.parent_area_code   = m.area_code
                   AND prior m.parent_area_number = m.area_number
  ) LOOP
    IF i.matchable >= 1 THEN
      l_count := l_count + 1;
      IF l_count > 1 THEN
        p_dataout := p_dataout ||', '|| i.name;
      ELSE
        p_dataout := i.name;
      END IF;
    END IF;
  END LOOP;
END name_hierarchy_txtidx;
```

NAME_HIERARCHY_TXTIDX(L_ROWID, L_CLOB)

```
set serveroutput on
DECLARE
  l_rowid ROWID;
  l_clob CLOB;
BEGIN
  select rowid
  into   l_rowid
  FROM   my_areas m
  WHERE  area_code = 'CPC'
  And    area_number = '40307';

  strava_pkg.name_hierarchy_txtidx(l_rowid, l_clob);
  dbms_output.put_line(l_clob);
END;
/

Streatley, West Berkshire, England, United Kingdom

PL/SQL procedure successfully completed.
```

OPTION 2: INDEX A *USER_DATASTORE* BASED ON THE RESULT OF A PL/SQL FUNCTION

The procedure is an attribute of the data store.
I have not been able to combine a multi-column data store with a user data store.

```
BEGIN
  ctx_ddl.create_preference('my_areas_lexer', 'BASIC_LEXER');
  ctx_ddl.set_attribute('my_areas_lexer', 'mixed_case', 'NO');
  ctx_ddl.create_preference('my_areas_datastore', 'user_datastore');
  ctx_ddl.set_attribute('my_areas_datastore', 'procedure'
                      , 'strava_pkg.name_hierarchy_txtidx');
  ctx_ddl.set_attribute('my_areas_datastore', 'output_type', 'CLOB');
END;
/

CREATE INDEX my_areas_name_txtidx on my_areas (name) INDEXTYPE IS ctxsys.context
PARAMETERS ('datastore my_areas_datastore lexer my_areas_lexer')
/
```

TEXT SEARCH EXAMPLES
1 TERM

I get the two Streatleys, one in Berkshire, and the other in Bedfordshire.

```
SELECT score(1), area_Code, area_number, name, suffix, name_hierarchy
FROM   my_areas m
WHERE  CONTAINS(name,'streatley',1)>0
/
```

```
  SCORE(1) AREA AREA_NUMBER NAME                 SUFFIX     NAME_hierarchy
---------- ---- ----------- -------------------- ---------- ------------------------------------------------------------
        16 CPC        41076 Streatley            CP         Streatley, Central Bedfordshire, England, United Kingdom
        16 CPC        40307 Streatley            CP         Streatley, West Berkshire, England, United Kingdom
```

TEXT SEARCH EXAMPLES
2 TERMS

Search for both the village and the county
Now I just get one result. The Streatley in Berkshire.

```
SELECT score(1), area_Code, area_number, name, suffix, name_hierarchy
FROM   my_areas m
WHERE  CONTAINS(name,'streatley and berks%',1)>0
/
```

```
  SCORE(1) AREA AREA_NUMBER NAME               SUFFIX     NAME_hierarchy
---------- ---- ----------- ------------------ ---------- ------------------------------------------------------------
        11 CPC        40307 Streatley          CP         Streatley, West Berkshire, England, United Kingdom
```

TEXT SEARCH EXAMPLES HIGHER UP THE HIERARCHY

I get 184 areas, of different types within the areas called Berkshire
because the name of the parent area appears in the hierarchy of all the children

```
SELECT score(1), area_Code, area_number, name, suffix, name_hierarchy
FROM   my_areas m
WHERE  CONTAINS(name,'berkshire',1)>0
/
```

SCORE(1)	Area Code	Area Number	NAME	SUFFIX	NAME_hierarchy
11	UTA	101678	Windsor and Maidenhead	(B)	Windsor and Maidenhead, Berkshire, England, United Kingdom
11	UTA	101680	Wokingham	(B)	Wokingham, Berkshire, England, United Kingdom
11	UTA	101681	Reading	(B)	Reading, Berkshire, England, United Kingdom
11	UTA	101685	West Berkshire		West Berkshire, England, United Kingdom
11	UTW	40258	Norreys	Ward	Norreys, Wokingham, Berkshire, England, United Kingdom
11	UTW	40261	Barkham	Ward	Barkham, Wokingham, Berkshire, England, United Kingdom

…

TEXT SEARCH EXAMPLES
TOP OF THE HIERARCHY

Exclude any result where its parent is also in the result set,
- because the name of the parent area appears in the hierarchy of all the children

```
WITH x AS (
SELECT area_code, area_number, parent_area_code, parent_area_number, name,
name_hierarchy
FROM   my_areas m
WHERE  CONTAINS(name,'berkshire',1)>0)
SELECT * FROM x WHERE NOT EXISTS (
  SELECT 'x' FROM x x1
  WHERE  x1.area_code = x.parent_area_code
  AND    x1.area_number = x.parent_area_number
)
/
```

```
Area         Area
Code       Number      SCORE NAME_hierarchy
---- ----------- ---------- ----------------------------------------------------
UTA        101685         11 West Berkshire, England, United Kingdom
CCTY            7         11 Berkshire, England, United Kingdom
```

TEXT SEARCH EXAMPLES
TOP OF THE HIERARCHY

Join the pre-processed areas through which activities pass
- to the areas found by the text search
- and then exclude areas whose parent was also found in the same activity.

```
WITH x AS (
SELECT aa.activity_id, m.area_code, m.area_number, m.parent_area_code, m.parent_area_number, m.name, m.name_hierarchy
FROM   my_areas m, activity_areas aa
WHERE  m.area_Code = aa.area_code
AND    m.area_number = aa.area_number
AND    CONTAINS(name,'berkshire',1)>0
)
SELECT a.activity_id, a.activity_date, a.activity_name, a.activity_type, a.distance_km, x.area_Code, x.area_number, x.name, x.name_hierarchy
FROM   x, activities a
WHERE  x.activity_id = a.activity_id
AND    a.activity_date between TO_DATE('01022019','DDMMYYYY') and TO_DATE('28022019','DDMMYYYY')
AND NOT EXISTS (SELECT 'x' FROM x x1
                WHERE  x1.area_code = x.parent_area_code
                AND    x1.area_number = x.parent_area_number
                AND    x1.activity_id = x.activity_id)
ORDER BY a.activity_date;
```

Activity ID	Activity Date	ACTIVITY_NAME	Activity Type	Distance (km)	Area Code	Area Number	NAME	NAME_hierarchy
2156308823	17-FEB-19	MV - Aldworth, CLCTC Aldworth-Reading	Ride	120.86	CCTY	7	Berkshire	England, United Kingdom
2156308823	17-FEB-19	MV - Aldworth, CLCTC Aldworth-Reading	Ride	120.86	UTA	101685	West Berkshire	England, United Kingdom
2172794879	24-FEB-19	MV - Maidenhead	Ride	48.14	CCTY	7	Berkshire	England, United Kingdom
2173048214	24-FEB-19	CLCTC: Maidenhead - Turville Heath	Ride	53.15	CCTY	7	Berkshire	England, United Kingdom
2173048406	24-FEB-19	Maidenhead - Burnham Beeches - West Drayton	Ride	27.92	CCTY	7	Berkshire	England, United Kingdom

…

TEXT SEARCH EXAMPLES
TOP OF THE HIERARCHY

Join the pre-processed areas through which activities pass

- to the areas found by the text search
- and then exclude areas whose parent was also found in the same activity.

```
WITH x AS (
SELECT aa.activity_id, m.area_code, m.area_number, m.parent_area_code, m.parent_area_number, m.name, m.name_hierarchy
FROM   my_areas m, activity_areas aa
WHERE  m.area_Code = aa.area_code
AND    m.area_number = aa.area_number
AND    CONTAINS(name,'Nuremberg',1)>0
)
SELECT a.activity_id, a.activity_date, a.activity_name, a.activity_type, a.distance_km, x.area_Code, x.area_number, x.name, x.name_hierarchy
FROM   x, activities a
WHERE  x.activity_id = a.activity_id
AND NOT EXISTS (SELECT 'x' FROM x x1
                WHERE  x1.area_code = x.parent_area_code
                AND    x1.area_number = x.parent_area_number
                AND    x1.activity_id = x.activity_id)
ORDER BY a.activity_date;
```

Activity ID	Activity Date	ACTIVITY_NAME	Activity Type	Distance (km)	Area Code	Area Number	NAME	NAME_HIERARCHY
1974863171	19-NOV-18	Nuremberg	Ride	2.01	UDIS	863053	Nuremberg Städte	Mittelfranken, Bayern, Germany
1978383889	22-NOV-18	Nuremburg Centre - Messe	Ride	6.96	UDIS	863053	Nuremberg Städte	Mittelfranken, Bayern, Germany
1978625587	22-NOV-18	Nuremburg Messe - Hauptbahnhof	Ride	5.09	UDIS	863053	Nuremberg Städte	Mittelfranken, Bayern, Germany

APEX FRONTEND

APEX FRONT END DEMO

Oracle 19c

- **Tim Hall's Blog:** Oracle REST Data Services (ORDS) : Standalone Mode

- Apex Admin
- Apex Login

Strava Activities App

- Apex Strava Activities App

SUMMARY

Loaded GPX

- **Queried GPX**

Converted GPX → Geometries

Loaded Geographical Data

- **Pre-processed comparison of GPX Geometrics to Geographical Geometries**

Text Indexing and Searching of hierarchies of Geographical Areas

Apex

REFERENCES

Go-Faster Blog: First Steps in Spatial Data

- https://blog.go-faster.co.uk/2021/02/spatialindex.html

GitHub repository: Strava

- https://github.com/davidkurtz/strava

QUESTIONS