ORACLE®

Introduction to MySQL GIS: A Crash Course

Norvald H. Ryeng
Software Engineer

norvald.ryeng@oracle.com

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

1. About MySQL
2. GIS basics
3. Example
4. Future directions

About MySQL

- “The world's most popular open source DBMS”
  - Very popular in web backends
- Has had GIS support for many years
- Released version 5.7 in October 2015
  - Replaced old GIS engine with Boost.Geometry
  - The start of a GIS initiative at MySQL
  - Staffed up a GIS team
  - Main GIS feature: InnoDB spatial indexes

GIS Implementation

- Using Boost.Geometry (since MySQL 5.7)
  - Extending Boost.Geometry
  - Fixing bugs in Boost
  - Contributing back to Boost
- Built-in functionality
  - No plugins needed

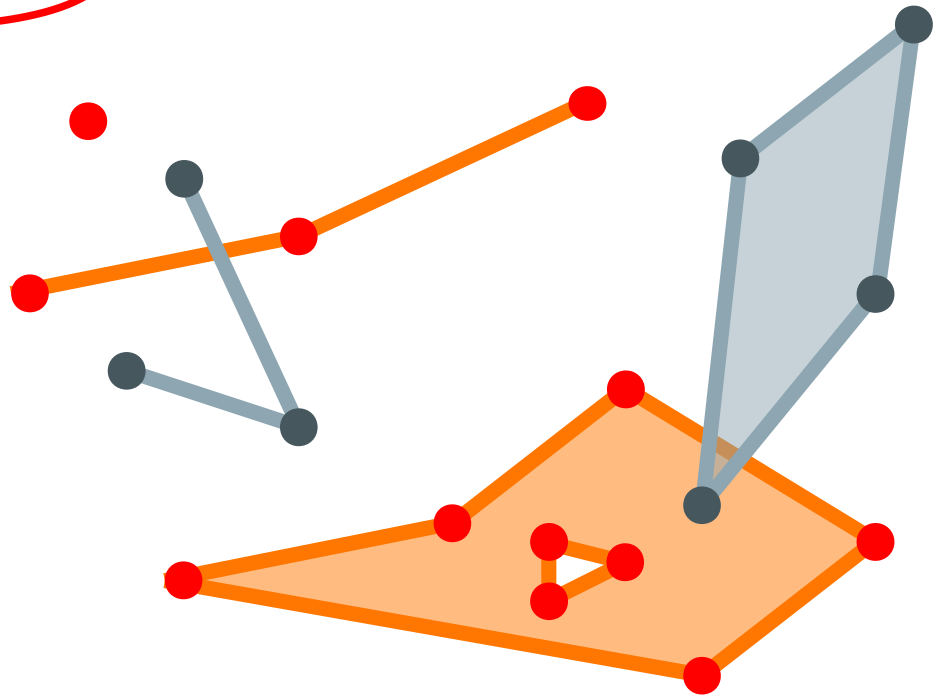
GIS basics

> “Geography is just physics slowed down, with a couple of trees stuck in it.”
>
> — Terry Pratchett, in *The Last Continent*

Non-instantiable, but can be used as column type

Data types

- Geometry
  - Point
  - LineString
  - Polygon
  - GeometryCollection
    - MultiPoint
    - MultiLineString
    - MultiPolygon



ORACLE

Spatial reference systems

- Each geometry is in a spatial reference system (SRS)
  - Specified by SRID (integer)
  - Geometries in different SRSs can't be compared
- MySQL supports a 2d Cartesian system (infinite plane)
  - SRID 0 is default if no SRID is specified
    - Unless the import format defaults to another SRS, e.g., WGS 84 for GeoJSON
- Other reference systems
  - MySQL doesn't have a spatial reference system database
  - Computations are always done in a 2d Cartesian system
    - Will be correct for SRID 0 and projected SRSs
  - Exception: ST_Distance_Sphere

Point

Point(0, 0)

ST_GeomFromText('POINT(0 0)')

ST_GeomFromGeoJSON('{"type":"Point","coordinates":[0,0]}')

ST_PointFromGeohash('s000', 0)

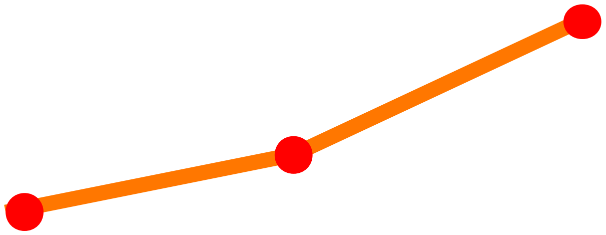
ST_GeomFromWKB(0x010100000000000000000000000000000000000000)

ST_GeomFromWKB(0x010100000000000000000000000000000000000000, 0)

Spatial reference system identifier (SRID)

LineString

- Two or more points



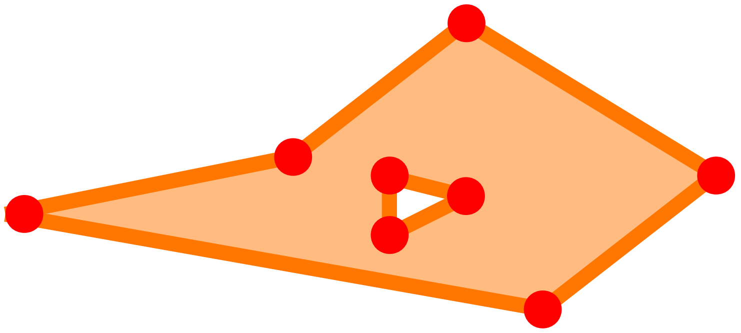
LineString(Point(0, 0), Point(1, 1))

ST_GeomFromText('LINESTRING(0 0, 1 1)')

ST_GeomFromGeoJSON('{"type":"LineString","coordinates":[[0,0],[1,1]]}')

Polygon

- One exterior ring
- Zero or more inner rings (holes)
- At least four points in each ring
  - Start and end point is the same
- Clockwise and couterclockwise rings can be mixed

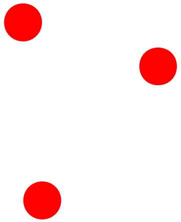


Inner ring

```
Polygon(LineString(Point(0, 0), Point(1, 0), Point(1, 1), Point(0, 0)))
ST_GeomFromText('POLYGON((0 0, 1 0, 1 1, 0 0), (0.2 0.1, 0.9 0.8, 0.9 0.1, 0.2 0.1))')
ST_GeomFromGeoJSON('{"type":"Polygon","coordinates":[[[0,0],[1,0],[1,1],[0,0]]]}')
```

MultiPoint

- One or more Point
  - Can't be empty



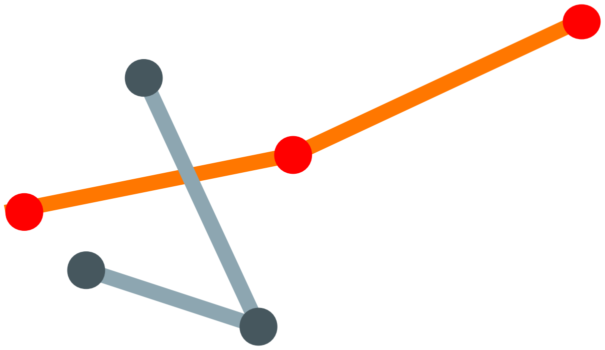
```
MultiPoint(Point(0, 0), Point(1, 1))

ST_GeomFromText('MULTIPOINT(0 0, 1 1)')

ST_GeomFromGeoJSON('{"type":"MultiPoint","coordinates":[[0,0],[1,1]]}')
```

MultiLineString

- One or more LineString
  - Can't be empty



MultiLineString(LineString(Point(0, 0), Point(1, 1)), LineString(Point(2, 2), Point(3, 3)))

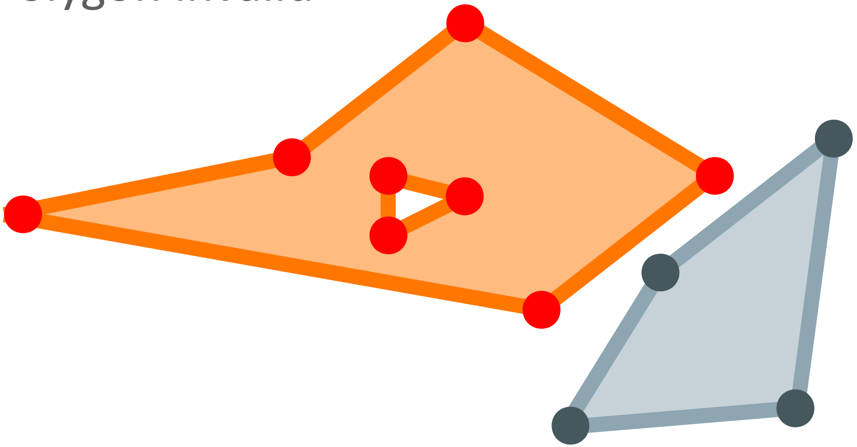
ST_GeomFromText('MULTILINESTRING((0 0, 1 1),(2 2, 3 3))')

ST_GeomFromGeoJSON('{"type":"MultiLineString","coordinates":[[[0,0],[1,1]],[[2,2],[3,3]]]}')

MultiPolygon

Will make the MultiPolygon invalid

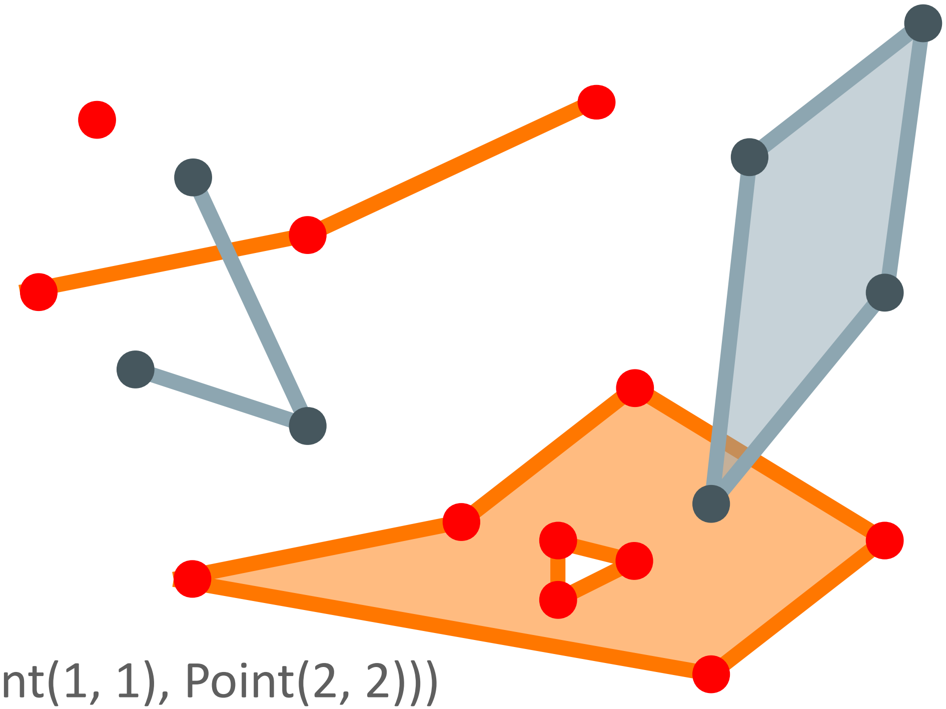
- One or more Polygon
  - Can't be empty
- Polygons shouldn't overlap
  - May touch in a finite number of points



```
MultiPolygon(Polygon(LineString(Point(0, 0), Point(1, 0), Point(1, 1), Point(0, 0))))
ST_GeomFromText('MULTIPOLYGON(((0 0, 1 0, 1 1, 0 0)))')
ST_GeomFromGeoJSON('{"type":"MultiPolygon","coordinates":[[[[0,0],[1,0],[1,1],[0,0]]]]}')
```

GeometryCollection

- Zero or more geometries
  - May be empty
- No restrictions on overlapping



GeometryCollection(Point(0, 0), Linestring(Point(1, 1), Point(2, 2)))

ST_GeomFromText('GEOMETRYCOLLECTION(POINT(0 0), LINESTRING(1 1, 2 2))')

ST_GeomFromGeoJSON('{"type":"GeometryCollection","geometries":
[{"type":"Point","coordinates":[0,0]},{"type":"LineString","coordinates":[[1,1],[2,2]]}]}')

ORACLE

Data types summary

- MySQL stores shape and spatial reference system identificator (SRID)
- MySQL can store invalid geometries
  - Check with ST_IsValid
- Only GeometryCollections can be empty (not even subclasses)
  - No POINT EMPTY, MULTILINESTRING EMPTY, etc.

GIS functions

- Only defined for valid geometries
- The result is undefined for invalid geometries
  - A best effort result
  - A weird result
  - An error
- Check with ST_IsValid if unsure
- Mixing SRIDs is not allowed
  - Results in an error

Functions

- Comparison
  - ST_Contains, ST_Crosses, ST_Disjoint, ST_Equals, ST_Intersects, ST_Overlaps, ST_Touches, ST_Within
- Set operations
  - ST_Difference, ST_Intersection, ST_SymDifference, ST_Union
- Measures
  - ST_Area, ST_Distance, ST_Distance_Sphere, ST_Length
- Extract properties
  - ST_Dimension, ST_EndPoint, ST_ExteriorRing, ST_GeometryN, ST_GeometryType, ST_InteriorRingN, ST_IsClosed, ST_IsEmpty, ST_IsSimple, ST_IsValid, ST_PointN, ST_SRID, ST_StartPoint, ST_X, ST_Y

Functions

- Analysis
  - ST_Buffer, ST_Centroid, ST_ConvexHull, ST_Envelope, ST_MakeEnvelope, ST_Simplify,
- Helper functions
  - ST_LatFromGeohash, ST_LongFromGeohash, ST_Validate
- Import
  - ST_GeomCollFromTxt/ST_GeomCollFromText, ST_GeomCollFromWKB, ST_GeomFromGeoJSON, ST_GeomFromText, ST_GeomFromWKB, ST_LineFromText, ST_LineFromWKB, ST_MLineFromText, ST_MLineFromWKB, ST_MPointFromText, ST_MPointFromWKB, ST_MPolyFromText, ST_MPolyFromWKB, ST_PointFromGeohash, ST_PolyFromText, ST_PolyFromWKB
- Export
  - ST_AsBinary, ST_AsGeoJSON, ST_AsText, ST_Geohash

ORACLE

Optimization

- The optimizer automatically uses an R-tree index if it thinks it's beneficial
- The query must have a suitable WHERE clause
  - ST_Contains, ST_Crosses, ST_Disjoint, ST_Equals, ST_Intersects, ST_Overlaps, ST_Touches, ST_Within
  - MBRContains, MBRDisjoint, MBREquals, MBRIntersects, MBROverlaps, MBRTouches, MBRWithin

Basic GIS Coordinates
Jan Van Sickle
Van Sickle
Basic GIS Coordinates
GIS A Computing Perspective
Second Edition
GIS: A Computing Perspective
Worboys and Duckham
Weighing the World
The Hotel
United Kingdom
Ireland
Danmark
Nederland
Deutschland
lenovo
ThinkPad

Example

Example

- Sightseeing in Trondheim, Norway
- Database of popular places to visit
  - A unique ID
  - Position (point)
  - Descriptive text

Creating the table

```
CREATE TABLE sights (
   id INT AUTO_INCREMENT PRIMARY KEY,
   pos POINT NOT NULL,
   description VARCHAR(200),
   SPATIAL KEY my_r_tree(pos)
) ENGINE=InnoDB;
```

```
INSERT INTO sights (pos, description) VALUES (
   Point(10.3958, 63.4269), 'Nidaros Cathedral'
);
```

```
SELECT ST_AsText(pos), description FROM sights;
```

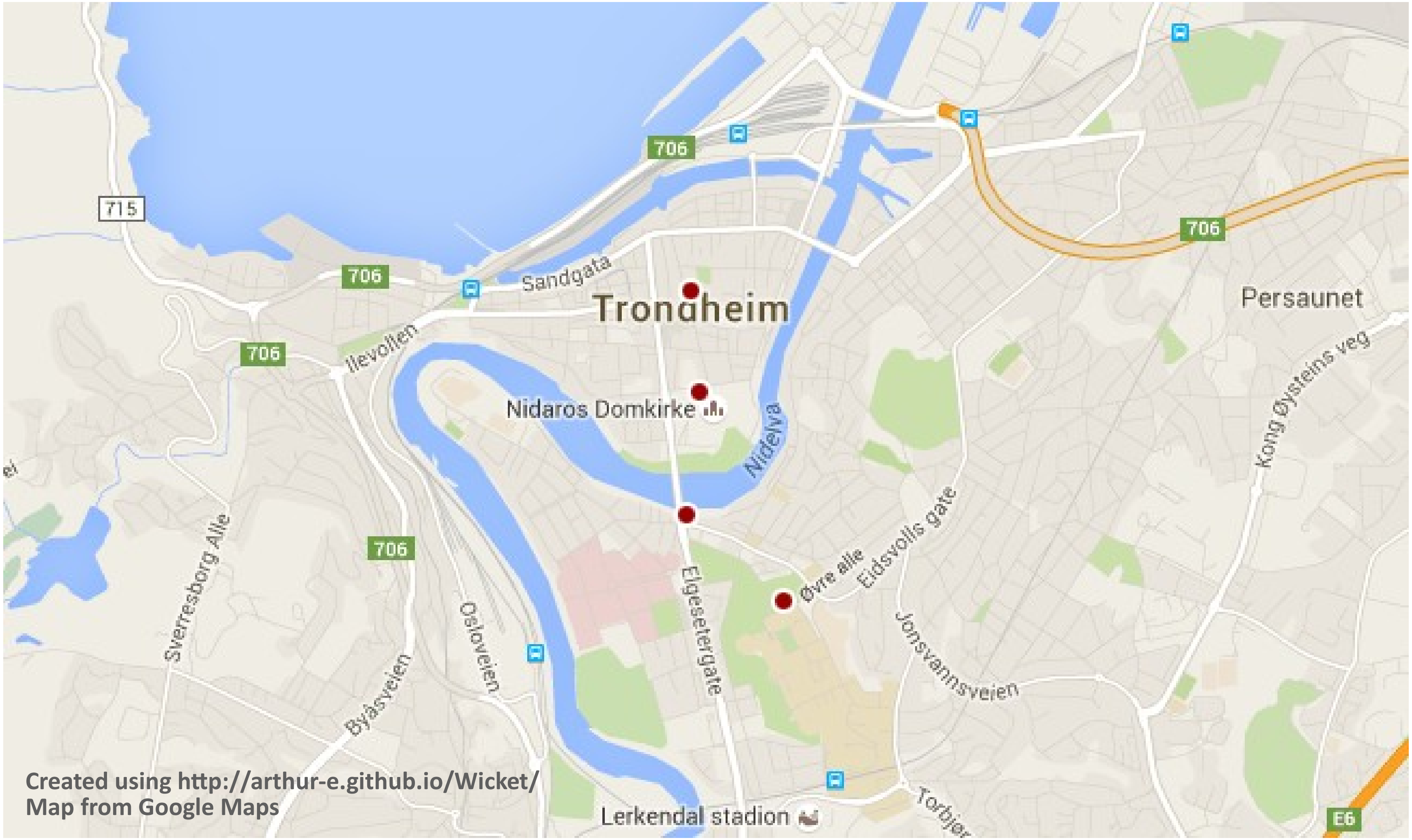
```
ST_AsText(pos)          description
POINT(10.3958 63.4269)  Nidaros Cathedral
```

Inserting data

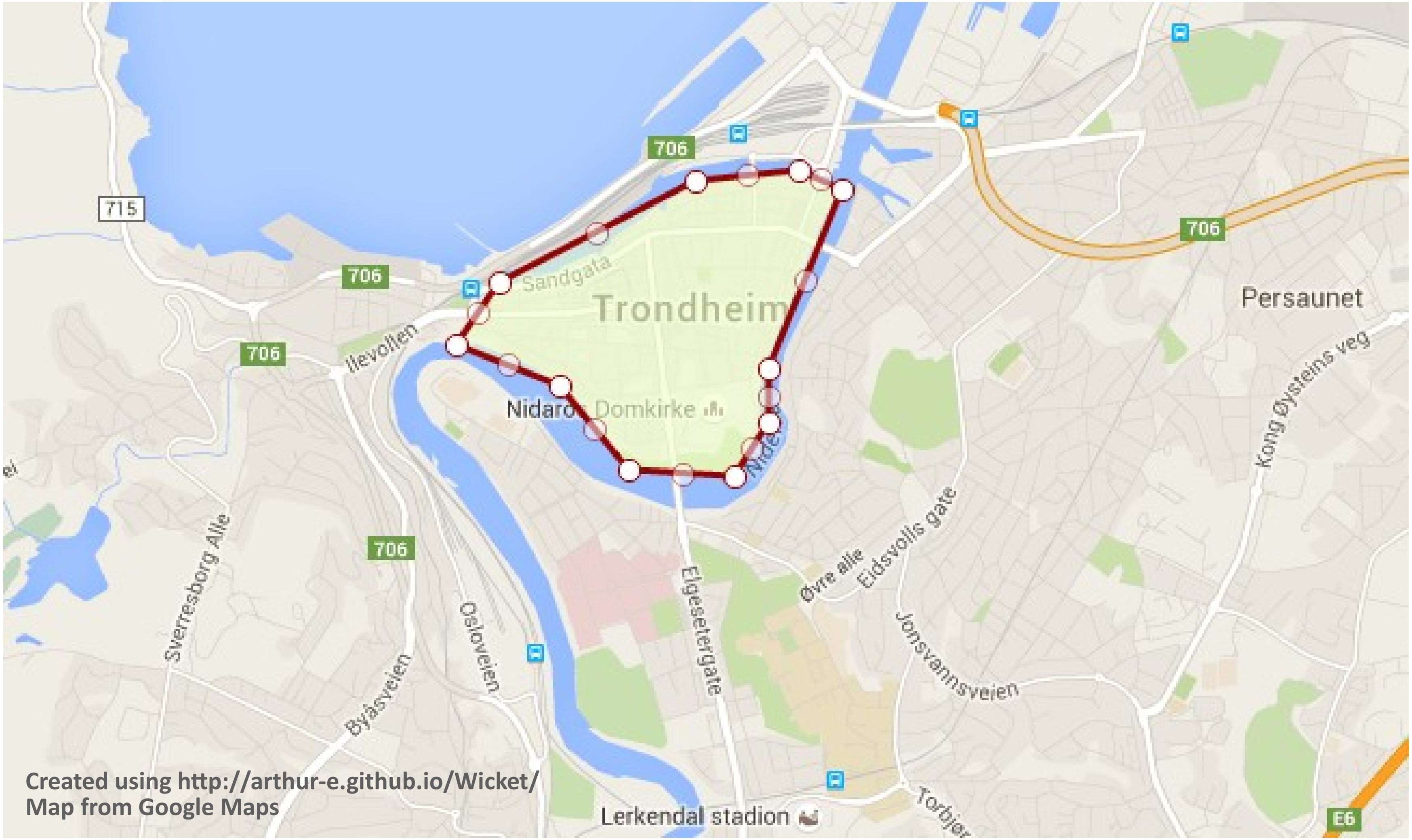
```
INSERT INTO sights (pos, description) VALUES (
   ST_GeomFromGeoJSON('{"type":"Point","coordinates":[10.4025,63.4194]}'),
   'Norwegian University of Science and Technology'
);

INSERT INTO sights (pos, description) VALUES (
   ST_GeomFromText('POINT(10.3948 63.4225)', 4326),
   'Student Society Building'
);

INSERT INTO sights (pos, description) VALUES (
   ST_GeomFromText('POINT(10.3951 63.4305)'),
   'Olav Tryggvason Monument'
);
```



Created using http://arthur-e.github.io/Wicket/
Map from Google Maps



Created using http://arthur-e.github.io/Wicket/
Map from Google Maps

Query

```
SET @city_center = ST_GeomFromText(
   'POLYGON((10.3765 63.4292, 10.3847 63.4277, 10.3902 63.4247, 10.3986 63.4245,
   10.4013 63.4264, 10.4013 63.4283, 10.4072 63.4347, 10.4037 63.4354,
   10.3954 63.4350, 10.3799 63.4314, 10.3765 63.4292))'
);

SELECT description FROM sights
   WHERE ST_Within(pos, @city_center);
```

Query

```
SET @city_center = ST_GeomFromText(
   'POLYGON((10.3765 63.4292, 10.3847 63.4277, 10.3902 63.4247, 10.3986 63.4245,
   10.4013 63.4264, 10.4013 63.4283, 10.4072 63.4347, 10.4037 63.4354,
   10.3954 63.4350, 10.3799 63.4314, 10.3765 63.4292))'
);

SELECT description FROM sights
   WHERE ST_Within(pos, @city_center);
```

```
ERROR 3033 (HY000): Binary geometry function st_within given two geometries of
different srids: 4326 and 0, which should have been identical.
```

The WKB format doesn't preserve the SRID

Query

```
UPDATE sights SET pos = ST_GeomFromWKB(ST_AsBinary(pos));

SELECT description FROM sights
   WHERE ST_Within(pos, @city_center);
```

```
description
Nidaros Cathedral
Olav Tryggvason Monument
```

Future directions

Future enhancements

- A non-flat Earth
  - Ellipsoidal Earth model
  - Projections
  - Transformations between these
- OGC and SQL/MM standard metadata tables (spatial reference systems, etc.)
- 3d and 4d support
  - 3dm, 3dz, 3dzm
- What else would **you** like to see?
  - Let us know!

> “GIS is a form of digital mapping technology. Kind of like Google Earth, but better.”
>
> — Arnold Schwarzenegger, Governor of California

Hardware and Software

Engineered to Work Together

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE®