

## Spatial Databases and GIS

### Solution for Sheet 5

#### Exercise 1 (Spatial Queries and Core SQL)

Give one SQL-Statement that retrieves the point having the shortest distance to the straight line with id 1 given the relations below:

POINT( id : integer, x : double, y : double)

LINE( id : integer, xStart : double, yStart : double, m : double, b : double, xEnd : double, yEnd : double)

m and b are the values for the equation:  $y = mx + b$

*Shortened version without equations:*

```
SELECT pointid, lineid
FROM (SELECT pointid, lineid, min(dist) as distance
      FROM ((SELECT p.id as pointid, l.id as lineid,
                  (Entfernung Lotfußpunkt-Punkt) as dist
              FROM Point p, Line l
              WHERE (Schnittpunkt liegt auf Segment) AND l.id=1
              UNION
              (SELECT ps.id as pointid, ls.id as lineid,
                  (Entfernung Startpunkt-Punkt) as dist
              FROM Point ps, Line ls
              WHERE ls.id=1)
              UNION
              (SELECT pe.id as pointid, le.id as lineid,
                  (Entfernung Endpunkt -Punkt) as dist
              FROM Point pe, Line le
              WHERE le.id=1)) as apld
      GROUP BY pointid, lineid) as mpld
WHERE distance = (SELECT min(dist) FROM (apld))
```

*Detailed version:*

```

SELECT pointid, lineid
FROM (SELECT pointid, lineid, min(dist)as distance
      FROM ((SELECT p.id as pointid, l.id as lineid,
                  (sqrt(((p.y+l.m*p.x - l.b)/(2*l.m)-p.x)^2 +
                        ((l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)-p.y)^2))
                  as dist
      FROM Point p, Line l
      WHERE ((p.y+l.m*p.x - l.b)/(2*l.m) <= l.xStart
            AND (p.y+l.m*p.x - l.b)/(2*l.m) >= l.xEnd )
            OR (p.y+l.m*p.x - l.b)/(2*l.m) >= l.xStart
            AND (p.y+l.m*p.x - l.b)/(2*l.m) <= l.xEnd )
            AND((l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)-p.y) <= l.yStart
            AND (l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)-p.y) >= l.yEnd )
            OR (l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)-p.y) >= l.yStart
            AND ((l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)-p.y)<= l.yEnd )
      UNION
      (SELECT ps.id as pointid, ls.id as lineid,
            sqrt((ls.xStart-ps.x)^2 + (ls.yStart-ps.y)^2) as dist
      FROM Point ps, Line ls)
      UNION
      (SELECT pe.id as pointid, le.id as lineid,
            sqrt((le.xEnd-pe.x)^2 + (le.yEnd-pe.y)^2) as dist
      FROM Point pe, Line le)) as apld
GROUP BY pointid, lineid) as mpld
WHERE distance = (SELECT min(dist)
                  FROM ((SELECT p.id as pointid, l.id as lineid,
                              (sqrt(((p.y+l.m*p.x - l.b)/(2*l.m)-p.x)^2 +
                                    ((l.m* (p.y+l.m*p.x - l.b)/
                                          (2*l.m) + l.b)-p.y)^2)) as dist
                  FROM Point p, Line l
                  WHERE ((p.y+l.m*p.x - l.b)/(2*l.m) <= l.xStart
                        AND (p.y+l.m*p.x - l.b)/(2*l.m) >= l.xEnd )
                        OR (p.y+l.m*p.x - l.b)/(2*l.m) >= l.xStart
                        AND (p.y+l.m*p.x - l.b)/(2*l.m) <= l.xEnd )
                        AND((l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)
                              -p.y) <= l.yStart
                        AND (l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)
                              -p.y) >= l.yEnd )
                        OR (l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)
                              -p.y) >= l.yStart
                        AND ((l.m* (p.y+l.m*p.x - l.b)/(2*l.m) + l.b)
                              -p.y)<= l.yEnd )
                  UNION
                  (SELECT ps.id as pointid, ls.id as lineid,

```

```

sqrt((ls.xStart-ps.x)^2
      + (ls.yStart-ps.y)^2) as dist
FROM   Point ps, Line ls)
UNION
(SELECT pe.id as pointid, le.id as lineid,
      sqrt((le.xEnd-pe.x)^2
            + (le.yEnd-pe.y)^2) as dist
FROM   Point pe, Line le))as apld)
  
```

## Exercise 2 (Overlay Operations)

Which geometry type(s) does the result of the following operations have?

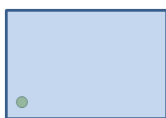
1. Intersection of two lines

*Point, Multipoint, Line, Multiline, Collection*



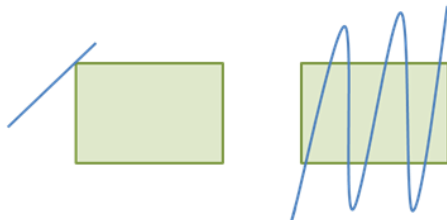
2. Intersection of a point and a polygon

*Point*



3. Subtraction of a line from a polygon

*Polygon, Multipolygon*



4. Subtraction of a point from a line

*Line, Multiline*



5. Union of two polygons

*Polygon, Multipolygon*



6. Union of a polygon and a line

*Polygon, Collection, Undefined*

