



Διαχείριση Χωρικών Δεδομένων

ΔΙΑΛΕΞΗ 4^η: Χωρικά δεδομένα σε Σχεσιακές Βάσεις Δεδομένων

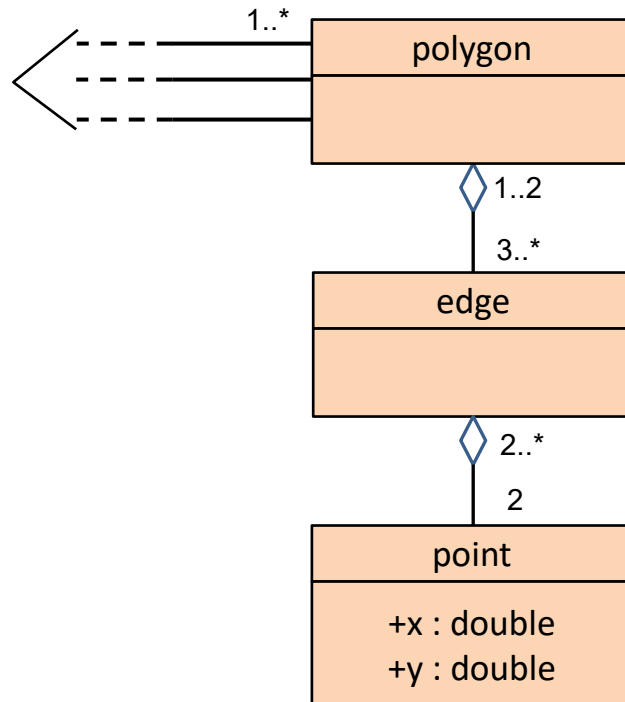
- Εισαγωγή και Διαχείριση Χωρικών Δεδομένων
- Υποτυπώδες Κτηματολόγιο
- Η ανάγκη για ενσωμάτωση της διαχείρισης χωρικών δεδομένων στα συστήματα ΒΔ

Νικόλαος Μήτρου
Καθ. ΕΜΠ



Εισαγωγή και Διαχείριση Χωρικών Δεδομένων

Επίπεδα γεωμετρικά ή
γεωγραφικά στοιχεία

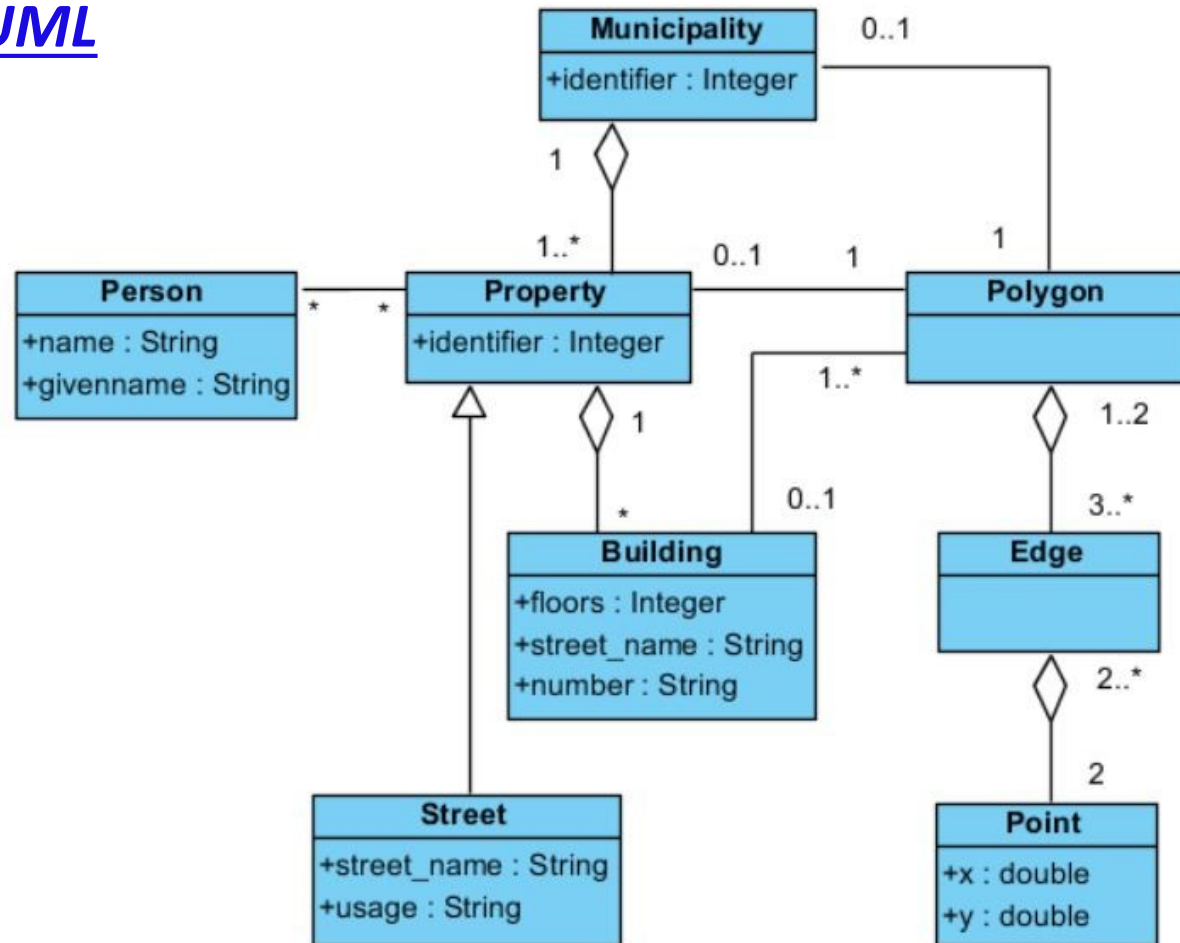


- Επίπεδα γεωμετρικά ή γεωγραφικά αντικείμενα συσχετίζονται με ένα ή περισσότερα πολύγωνα
- Κάθε πολύγωνο απαρτίζεται από ένα σύνολο ακμών (τρεις ή περισσότερες). Κάθε ακμή συμμετέχει στην περίμετρο ενός ή δύο πολυγώνων
- Κάθε ακμή ορίζεται από ακριβώς δύο σημεία. Κάθε σημείο συμμετέχει στον ορισμό δύο ή περισσότερων ακμών



Παράδειγμα: Υποτυπώδες Κτηματολόγιο

Μοντέλο UML





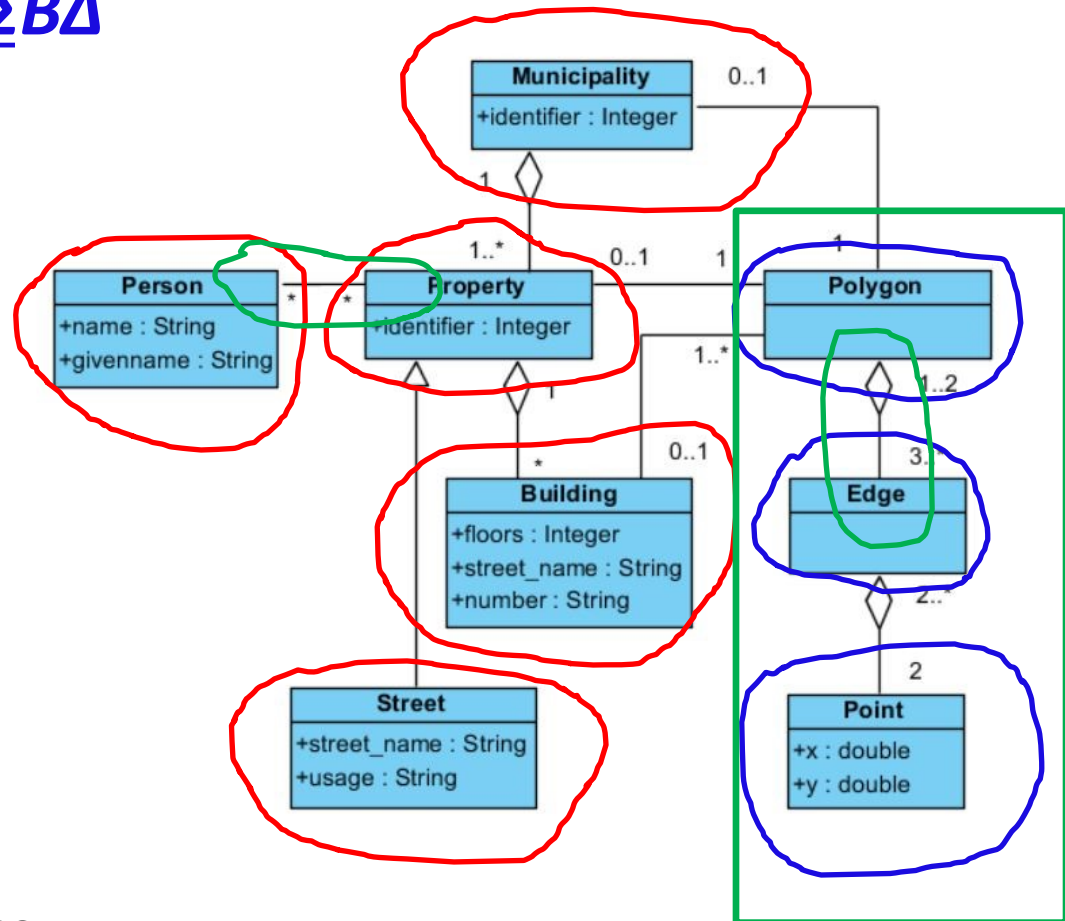
Υποτυπώδες Κτηματολόγιο (συνέχεια)

Αντιστοίχιση σε ΣΒΔ

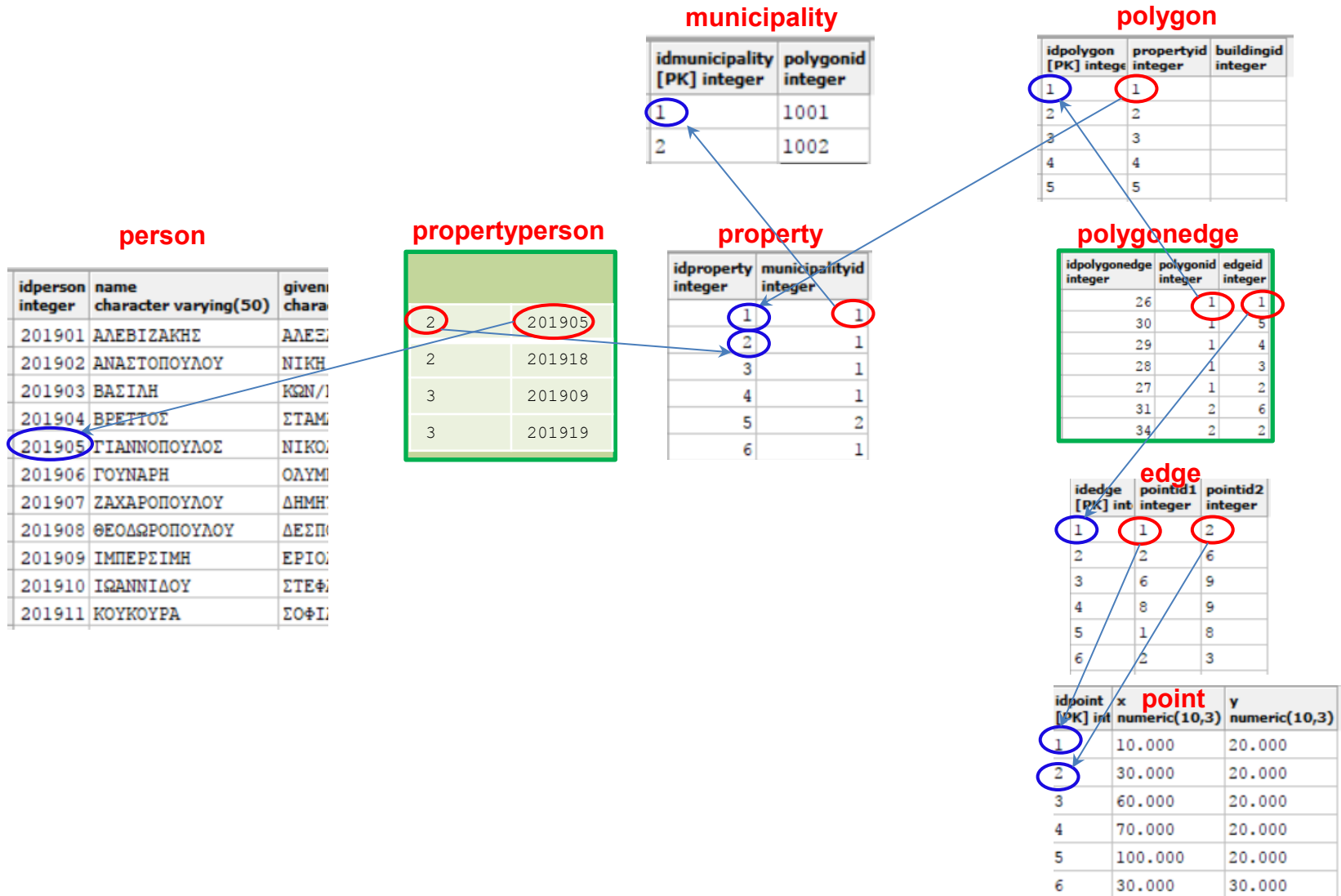
Δημιουργία 10 πινάκων

- ✓ Municipality
- ✓ property
- ✓ Person
- ✓ Building
- ✓ Street
- ✓ Polygon
- ✓ Edge
- ✓ Point
- ✓ Polygonedge
- ✓ Propertyperson

Χωρικά

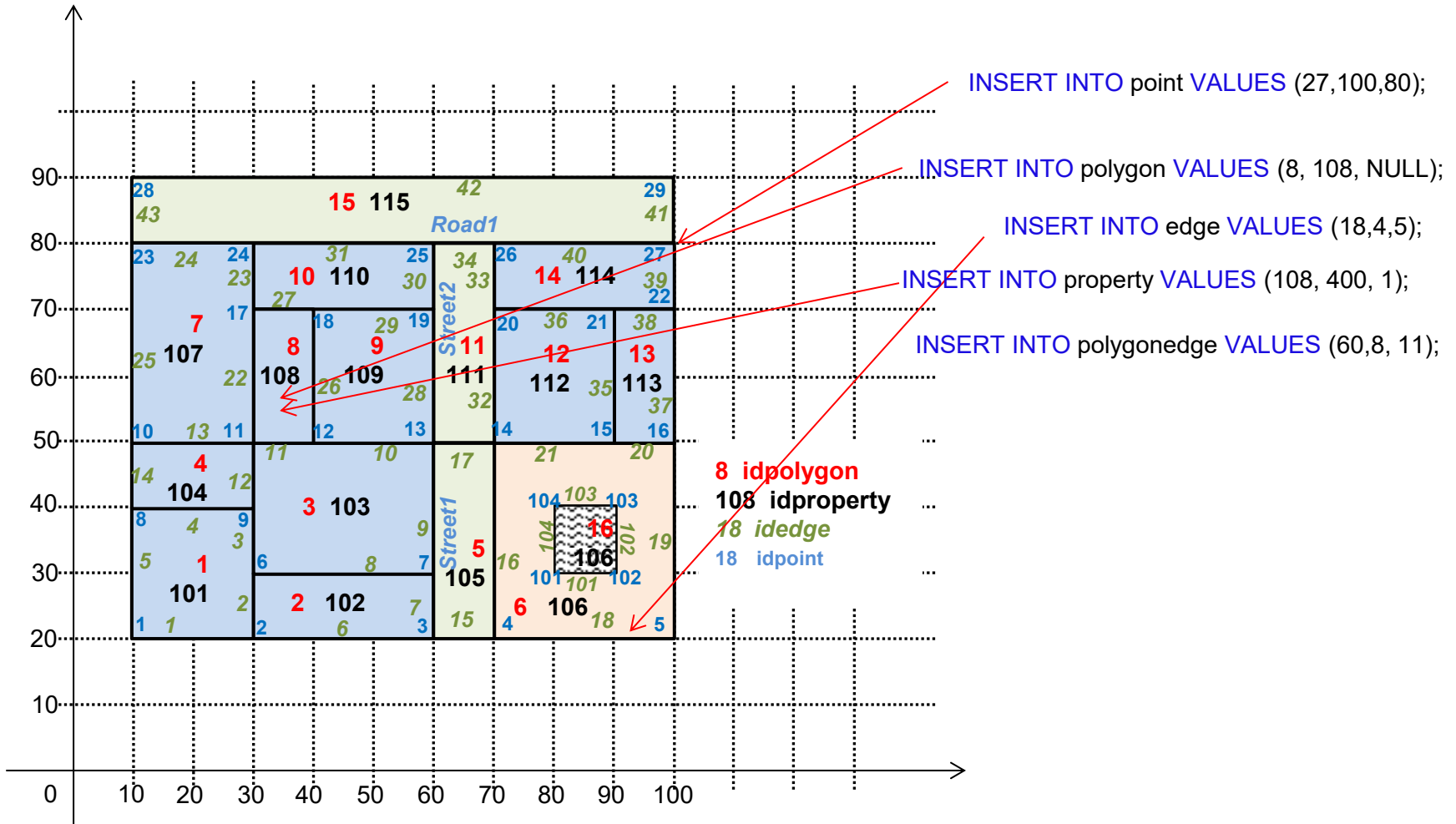


Γιατί όχι και edgepoint ???





Υποτυπώδες Κτηματολόγιο (συνέχεια)





Υποτυπώδες Κτηματολόγιο (συνέχεια)

Υπολογισμός μήκους ακμών

```
SELECT e.IDedge,  
       ROUND(SQRT( POWER((p1.x - p2.x),2) +  
                   POWER((p1.y - p2.y),2)) ,2)  
       AS length  
FROM exercisel.edge e, exercisel.point p1,  
     exercisel.point p2  
WHERE e.pointID1 = p1.IDpoint  
AND   e.pointID2 = p2.IDpoint  
ORDER BY length, e.IDedge
```



idedge integer	length numeric
33	10.00
34	10.00
38	10.00
39	10.00
41	10.00
43	10.00
101	10.00
102	10.00
103	10.00
104	10.00
1	20.00
4	20.00
5	20.00
9	20.00
10	20.00
13	20.00



Υποτυπώδες Κτηματολόγιο (συνέχεια)

Εύρεση ακμών ανά ιδιοκτησία (*propertyedge*)

```
SELECT propertyid, edgeid
FROM exercise1.polygon, exercise1.polygonedge
WHERE
    polygon.idpolygon=polygonedge.polygonid
AND polygon.propertyid > 0
```




propertyid integer	edgeid integer
101	1
101	2
101	3
101	4
101	5
102	6
102	7
102	8
102	2
103	8
103	9
103	10
103	11
103	12
103	3
104	4



Υποτυπώδες Κτηματολόγιο (συνέχεια)

Υπολογισμός περιμέτρου πολυγώνων I: με SQL query

```
SELECT C1.propertyid, C1.circumference FROM
(SELECT propertyedge.propertyid,
  ROUND(SUM(A1.length),2) AS circumference
FROM
  (SELECT propertyid, edgeid
  FROM exercisel.polygon, exercisel.polygonedge
  WHERE
    polygon.idpolygon=polygonedge.polygonid
  AND polygon.propertyid > 0) propertyedge,
  (SELECT e.IDedge, SQRT( POWER((p1.x - p2.x), 2)
    + POWER ((p1.y - p2.y), 2)) AS length
  FROM exercisel.edge e, exercisel.point p1,
  exercisel.point p2
  WHERE
    e.pointID1 = p1.IDpoint AND
    e.pointID2 = p2.IDpoint ) A1
WHERE propertyedge.edgeid = A1.idedge
GROUP BY propertyedge.propertyid) C1
ORDER BY C1.circumference, C1.propertyid;
```



propertyid integer	circumference numeric
104	60.00
108	60.00
113	60.00
101	80.00
102	80.00
105	80.00
109	80.00
110	80.00
111	80.00
112	80.00
114	80.00
103	100.00
107	100.00
106	160.00
115	200.00



Υποτυπώδες Κτηματολόγιο (συνέχεια)

Υπολογισμός περιμέτρου πολυγώνων II - ως συνάρτηση

```
DROP FUNCTION IF EXISTS exercisel.circumference(integer) CASCADE;
CREATE FUNCTION exercisel.circumference(polygID integer)
    RETURNS NUMERIC AS $$
    DECLARE circ NUMERIC :=0;
    p1ID exercisel.point.IDpoint%TYPE;
    p2ID exercisel.point.IDpoint%TYPE;
    p1x exercisel.point.x%TYPE;
    p1y exercisel.point.y%TYPE;
    p2x exercisel.point.x%TYPE;
    p2y exercisel.point.y%TYPE;
BEGIN
    FOR p1ID, p2ID IN (SELECT edge.pointID1, edge.pointID2
        FROM exercisel.edge, exercisel.polygonedge
        WHERE edge.IDedge=polygonedge.edgeID
        AND polygonedge.polygonid=polygID)
    LOOP
        p1x = (SELECT x FROM exercisel.point WHERE point.IDpoint=p1ID);
        p1y = (SELECT y FROM exercisel.point WHERE point.IDpoint=p1ID);
        p2x = (SELECT x FROM exercisel.point WHERE point.IDpoint=p2ID);
        p2y = (SELECT y FROM exercisel.point WHERE point.IDpoint=p2ID);
        circ = circ + SQRT( POWER((p1x - p2x), 2) +
            POWER ((p1y - p2y), 2));
    END LOOP;
    RETURN ROUND(circ,2);
END;
$$ LANGUAGE plpgsql;
```



Υποτυπώδες Κτηματολόγιο (συνέχεια)

Υπολογισμός περιμέτρου πολυγώνων II – (συνέχεια)

```
CREATE VIEW exercisel.circumferences AS
  SELECT IDpolygon AS ΠΟΛΥΓΩΝΟ,
         exercisel.circumference(IDpolygon) AS ΠΕΡΙΜΕΤΡΟΣ
  FROM exercisel.polygon pol
  WHERE pol.propertyID>0
  ORDER BY ΠΕΡΙΜΕΤΡΟΣ, ΠΟΛΥΓΩΝΟ;
```

```
SELECT * FROM exercisel.circumferences;
```



ΠΟΛΥΓΩΝΟ integer	ΠΕΡΙΜΕΤΡΟΣ numeric
16	40.00
4	60.00
8	60.00
13	60.00
1	80.00
2	80.00
5	80.00
9	80.00
10	80.00
11	80.00
12	80.00
14	80.00
3	100.00
7	100.00
6	120.00
15	200.00

Άσκηση:

Πώς πρέπει να κληθεί η `circumference()` για να δώσει την περίμετρο των ιδιοκτησιών, όταν μπορούν αυτές να απαρτίζονται από πολλά πολύγωνα;

Να ελεγχθεί, αφού δώσετε, π.χ.:

```
UPDATE exercisel.polygon SET propertyID=1
WHERE IDpolygon=6;
```



PL/PgSQL (συνέχεια)

Επιστροφή σύνθετων τύπων (records)

- **RETURN NEXT** expression;

```
CREATE TYPE exercisel.neighbours
AS (pers INT, nam VARCHAR, prop INT, border INT);

CREATE OR REPLACE FUNCTION exercisel.find_neighbours(text)
RETURNS SETOF exercisel.neighbours AS $$
DECLARE
    name ALIAS FOR $1;
    ...
    neighbs exercisel.neighbours;
BEGIN
    ...
    FOR neighbs IN (SELECT ... )
    LOOP
        RETURN NEXT neighbs
    END LOOP;
    RETURN;
END;
$$ LANGUAGE plpgsql;
```



PL/PgSQL (συνέχεια)

Επιστροφή σύνθετων τύπων (records) -συνέχεια

- **RETURN QUERY** query;

```
CREATE TYPE exercisel.neighbours
AS (pers INT, nam VARCHAR, prop INT, border INT);

CREATE OR REPLACE FUNCTION exercisel.find_neighbours(text)
RETURNS SETOF exercisel.neighbours AS $$
DECLARE
    name ALIAS FOR $1;
    ...
BEGIN
    ...
    RETURN QUERY SELECT personID, name, propertyID, edge
                  FROM ...
END;
```



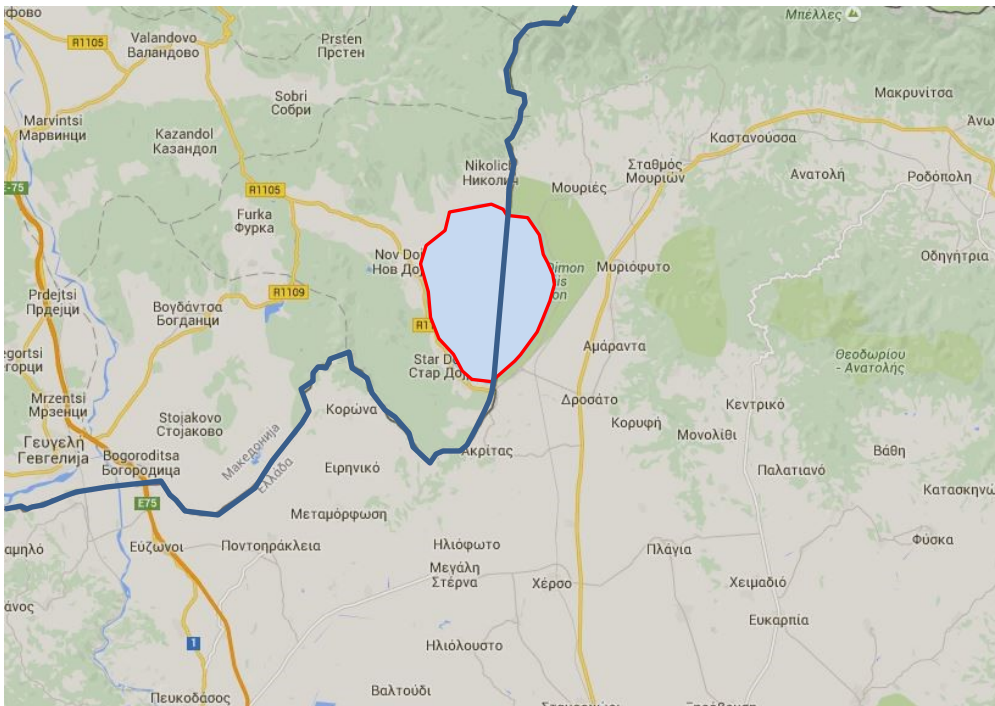
Ενσωματωμένη διαχείριση χωρικών δεδομένων

- Η διαχείριση χωρικών δεδομένων σε ΣΒΔ, ως συμβατικών πινάκων της ιεραρχίας <πολύγωνο><ακμή><σημείο>, όπως στα προηγούμενα παραδείγματα, είναι πολύπλοκη και επιρρεπής σε λάθη:
 - Τα ερωτήματα ενημέρωσης και αναζήτησης είναι πολύπλοκα
 - Όλοι οι υπολογισμοί πρέπει να γίνουν στο πρόγραμμα εφαρμογής (εκτός συστήματος διαχείρισης)
 - Οποιαδήποτε αναδόμηση των δεδομένων επιφέρει αλλαγές στα προγράμματα εφαρμογών
- Αντ' αυτών, εισάγονται ειδικές στήλες «γεωμετρίας» σε πίνακες περιγραφικών δεδομένων γεωμετρικών ή γεωγραφικών οντοτήτων (**abstract data types**)
- Διατίθεται ένα πλούσιο ρεπερτόριο συναρτήσεων για τις συνηθέστερες λειτουργίες και τους αναγκαίους υπολογισμούς, εντός του συστήματος διαχείρισης.
- Τα γνωστά συστήματα διαχείρισης βάσεων δεδομένων (DBMS) προσφέρουν επεκτάσεις με μια τέτοια λειτουργικότητα



Ενσωματωμένη διαχείριση χωρικών δεδομένων (συνέχεια)


Παράδειγμα γενικευμένων χωρικών δεδομένων (abstract data types)



```
CREATE TABLE lakes (
```

```
  IDlake VARCHAR(20) PRIMARY KEY,  
  name VARCHAR(50));
```


```
SELECT AddGeometryColumn('lakes',  
  'lake_geom', 4326, 'POLYGON', 2);
```

IDlake	name	lake_geom
GR_L021	Δοϊράνη	

```
CREATE TABLE borders (
```

```
  IDborders VARCHAR(20) PRIMARY KEY);
```

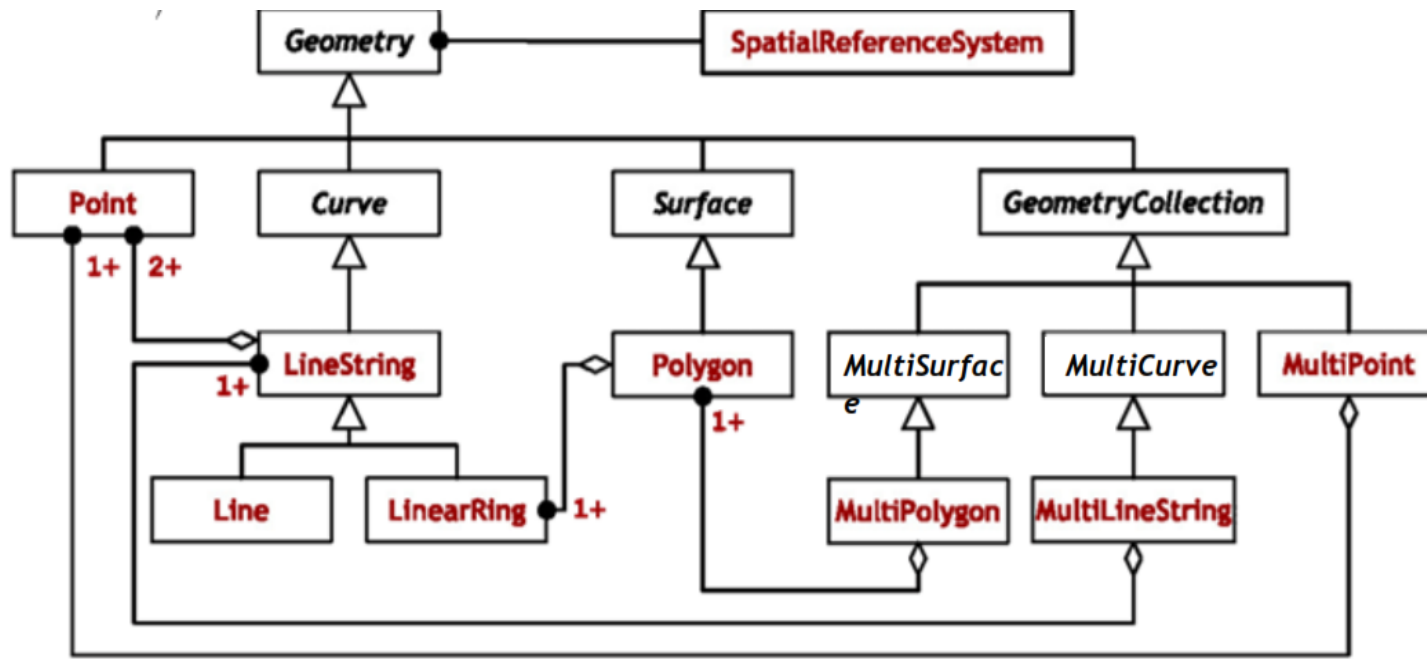
```
SELECT AddGeometryColumn('borders',  
  'geometry', 4326, 'LINESTRING', 2);
```

IDborders	geometry
GR_B042	



Απλές οντότητες κατά OGC

Η ιεραρχία απλών (γεωμετρικών) οντοτήτων
(*simple features*) κατά OGC





Ερωτήματα επί χωρικών δεδομένων

Παράδειγμα εκτύπωσης γεωμετρίας

Ερώτημα

```
SELECT name, ST_astext(lake_geom) FROM lakes;  
WHERE name = 'Δοϊράνη';
```

Αποτέλεσμα

name	lake_geom
Δοϊράνη	POLYGON ((22.7209 41.2390, 22.7109 41.2324, 22.7082 41.2273, 22.7209 41.2390));

Παράδειγμα ελέγχου τοπολογικής σχέσης

```
SELECT lakes.name  
FROM lakes, borders  
WHERE CROSSES (lakes.lake_geom, borders.geometry)=1;
```