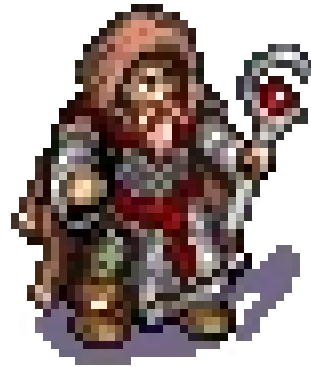


# Topic : Spatial Database



Go Frendi Gunawan

[gofrendiasgard@gmail.com](mailto:gofrendiasgard@gmail.com)

<https://github.com/goFrendiAsgard>

# About the topic

- The benefit of using database
- The spatial database standard
- Import your shp into database

# Shp format

- Used by ESRI
- Widely used
- Binary
- Not the only format available
- Not friendly to open source

# Shp alternatives

- Geojson
  - <http://geojson.org>
- Database
  - <http://www.opengeospatial/standards/sfs>
  - Real world RDBMS:
    - Post-gis (Postgree)
    - Spatialite (SQLite)
    - MySQL-spatial-extension (MySQL)

# The benefit of GeoJSON

- Greatly compatible with javascript
  - <http://leaflet.cloudmade.com>
- Can be produced easily by php (e.g: fetch from database and present it as json)
  - json\_encode
  - json\_decode
- Human readable
- Fat-free (compare to xml)

# The benefit of using database

- Platform independent
- Can be used in many application (desktop & web)
- The freedom to perform “acrobatic SQL-query”

# How about “normal” database?

- It is not standardized
- Need your own logic to:
  - Calculate the area of polygon
  - Selecting every point in a polygon
  - Spatial things
- It is just not easy for spatial tasks

# Who make spatial-database?

- OGC (Open Geo Spatial)
  - An open source organization
  - Making standard
    - <http://www.opengeospatial.org/standards/sfs>
  - Good guys
  - Have too much time :)



# Special things with spatial database

- Geom Field
- Spatial relation
- Standardized

# Geomfield

- Define it
  - Point, Polyline, Polygon, Geometry
- Fetch it
  - AsBinary
    - not readable
  - AsText
    - **POINT(12.34 15.75)**
- Insert it
  - **GeomFromText('POLYGON ((0 0, 0 3, 3 3, 3 0, 0 0))')**

# Spatial Relation

- Contains(g1, g2)
- Crosses(g1, g2)
- Disjoint(g1, g2)
- Equals(g1, g2)
- Intersects(g1, g2)
- Overlaps(g1, g2)
- Touches(g1, g2)
- Within (g1, g2)
- More here :  
<http://dev.mysql.com/doc/refman/5.1/en/functions-for-testing-spatial-relati>

# Convert shp to database

- Use ogr2ogr (don't worry there is a GUI for this)
  - `ogr2ogr -f "MySQL" MySQL:"geo,user=root,host=localhost,password=toor" -lco engine=MYISAM airports.shp`
- Use qgis plugin (works for spatialite and postgis)

# Convert database to geoJSON

- `SELECT ASTEXT(your_geom_field) FROM your_table`
  - `POINT(123.45 67.89)`
- Perform a little logic to change that into this:
  - `{Feature:point, coordinate:[123.45, 67.89]}`
- Or simply use `geophp` (only for php)

# Best practice (MySQL)

- Importing and exporting binary data into plain-text SQL is risky
- Use mysqldump instead
  - Mysqldump -u root -p password database > filename.sql
  - Mysql -u root -p password database < filename.sql

# Let's get your hand dirty

- Transform shp into mySQL with ogr2ogr
  - Alaska
  - Grassland
  - airports
- Do some “spatial SQL”
  - Airports within grassland
- Show it on cloudmade's leaflet platform (require a bit php)