



# PostgreSQL/PostGIS

## Database for Geospatial Applications

*Ahmad Zaenun Faiz*

10 - 17 September 2023

# Biodata Diri



Ahmad Zaenun Faiz  
**GIS & Data Specialist**



Sarjana Sains  
Fakultas Geografi UGM



Data Engineer  
Kaleka Indonesia



GIS related projects

- WebGIS – Indonesia Tourism Development Project
- Geoportal Cagar Budaya - Ertim Conservation Institute



<https://www.linkedin.com/in/ahmad-zaenun-faiz/>



<https://github.com/ahmadzfaiz>



# Full-stack WebGIS Course

1. GIS fundamental
2. Introduction to HTML & CSS
3. Front-end Development with JavaScript
4. JavaScript Mapping Library
5. Build a Geo API
6. Working with Spatial Database
7. Python Fundamental
8. Backend Development for Geospatial Applications
9. WebGIS Project: Tourism WebMap in Bikini Bottom

*From A to Z*

## Fullstack WebGIS Course

*Geocourse.id Special Course for All!*



*Are You Interested?*



# Full-stack WebGIS course roadmap

## GIS Fundamental

2 sessions ~ QGIS

## HTML & CSS

2 sessions ~ HTML & CSS

## JavaScript

4 sessions ~ JavaScript

## Spatial Database

3 sessions ~ PostgreSQL & PostGIS

## Geospatial API

3 sessions ~ GeoServer

## JavaScript for Map

3 sessions ~ Openlayers & Leaflet

## Python

3 sessions ~ Python

## Backend Framework

4 sessions ~ Django

## Final Project

6 sessions







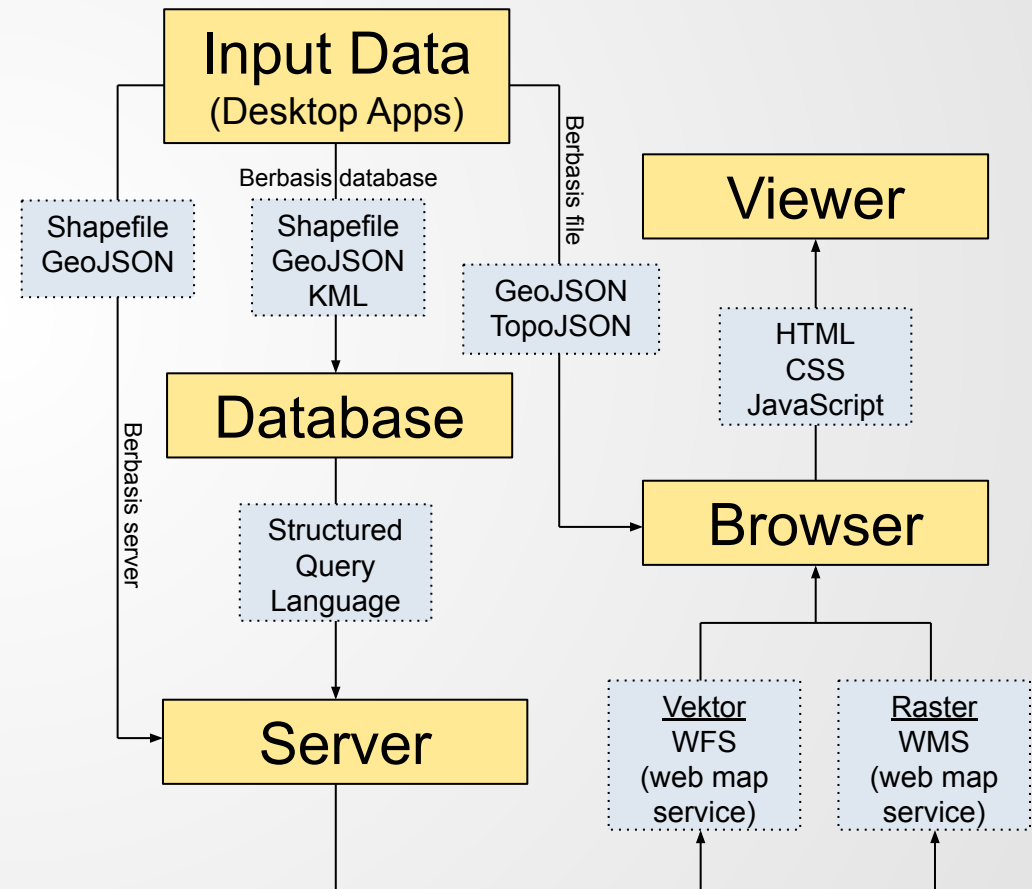
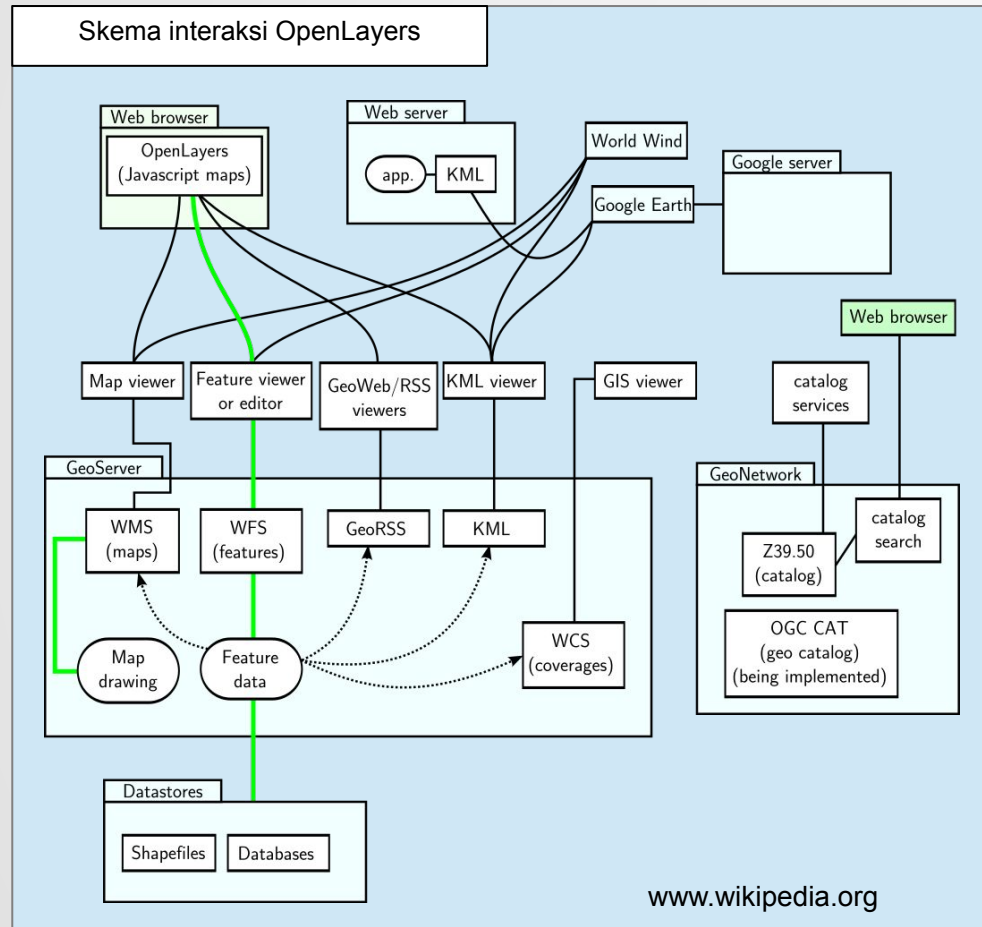
# Kenapa harus belajar Database?

- Membuat agar user mudah mendapatkan data
- Mampu dipanggil secara cepat
- Bekerja dengan banyak data dengan berbagai macam format dan variabel
- Menyediakan tempat penyimpanan data yang relevan
- Memungkinkan untuk pengembangan aplikasi ataupun analisis data yang kompleks





# Cara Kerja WebGIS





# Contoh aplikasi pengguna database



**Sosial Media**



**Marketplace**



**WIKIPEDIA**

**Web Pages**



**Google Maps**

**Map Services**







# Spreadsheet vs Database



- Informasi tersusun dalam sel-sel
- Akses data dalam jumlah terbatas
- Satu user dalam satu waktu
- Dikelola oleh user
- Data entry secara manual



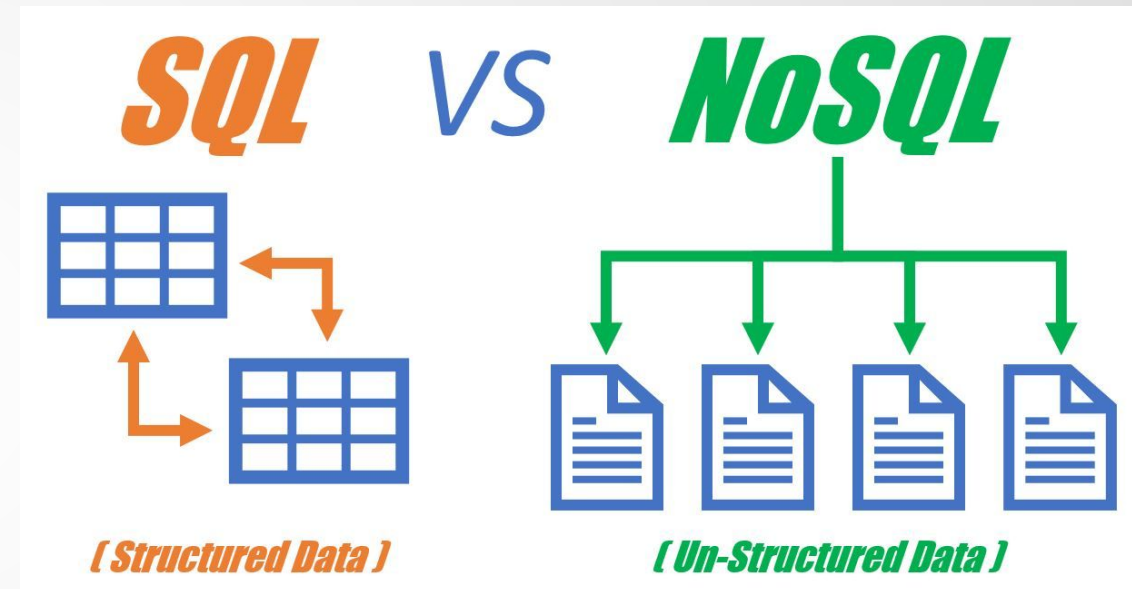
- Informasi tersusun dalam koleksi data yang kompleks
- Akses data dalam jumlah besar
- Banyak user dalam satu waktu
- Dikelola oleh DMS (*Data Management System*)
- Data entry tegas dan konsisten





# Tipe Database

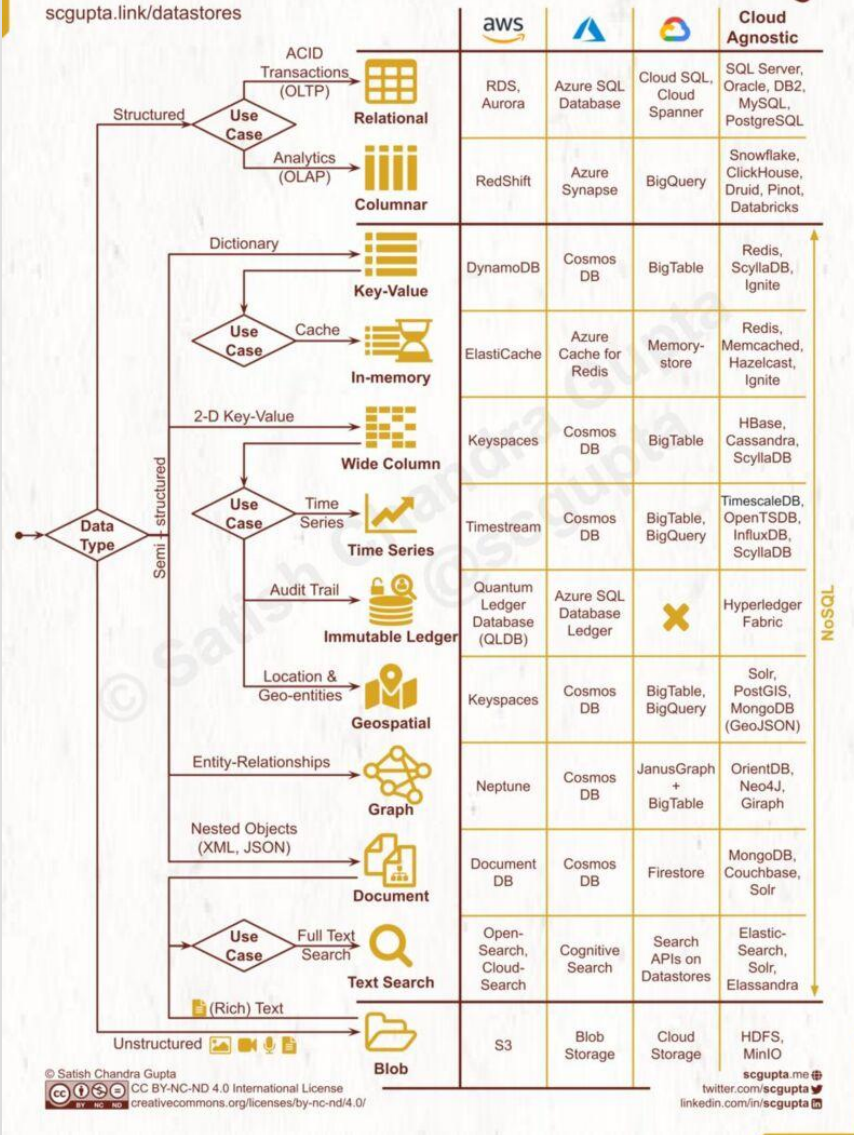
SQL	NOSQL
Relational Database management system	Distributed Database management system
Vertically Scalable	Horizontally Scalable
Fixed or predefined Schema	Dynamic Schema
Not suitable for hierarchical data storage	Best suitable for hierarchical data storage
Can be used for complex queries	Not good for complex queries





## SQL vs. NoSQL: Cheatsheet for AWS, Azure, and Google Cloud

scgupta.link/datastores



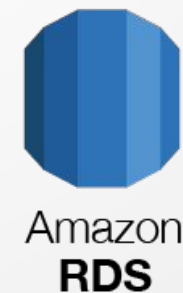
# Tipe-tipe Data

- Terstruktur
- Semi-terstruktur
- Tidak terstruktur





# SQL Database





# Dasar-dasar Bahasa Kueri SQL

Structured Query Language atau dikenal sebagai **SQL** adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional.

Sejarah SQL dimulai oleh seorang peneliti dari IBM, Donald D. Chamberlin dan Raymond F. Boyce pada awal tahun 1970 setelah mempelajari model relasi data dari Edgar F. Codd

Standardisasi SQL dimulai pada tahun 1986, ditandai dengan dikeluarkannya standar SQL oleh ANSI.





# SQL Syntax

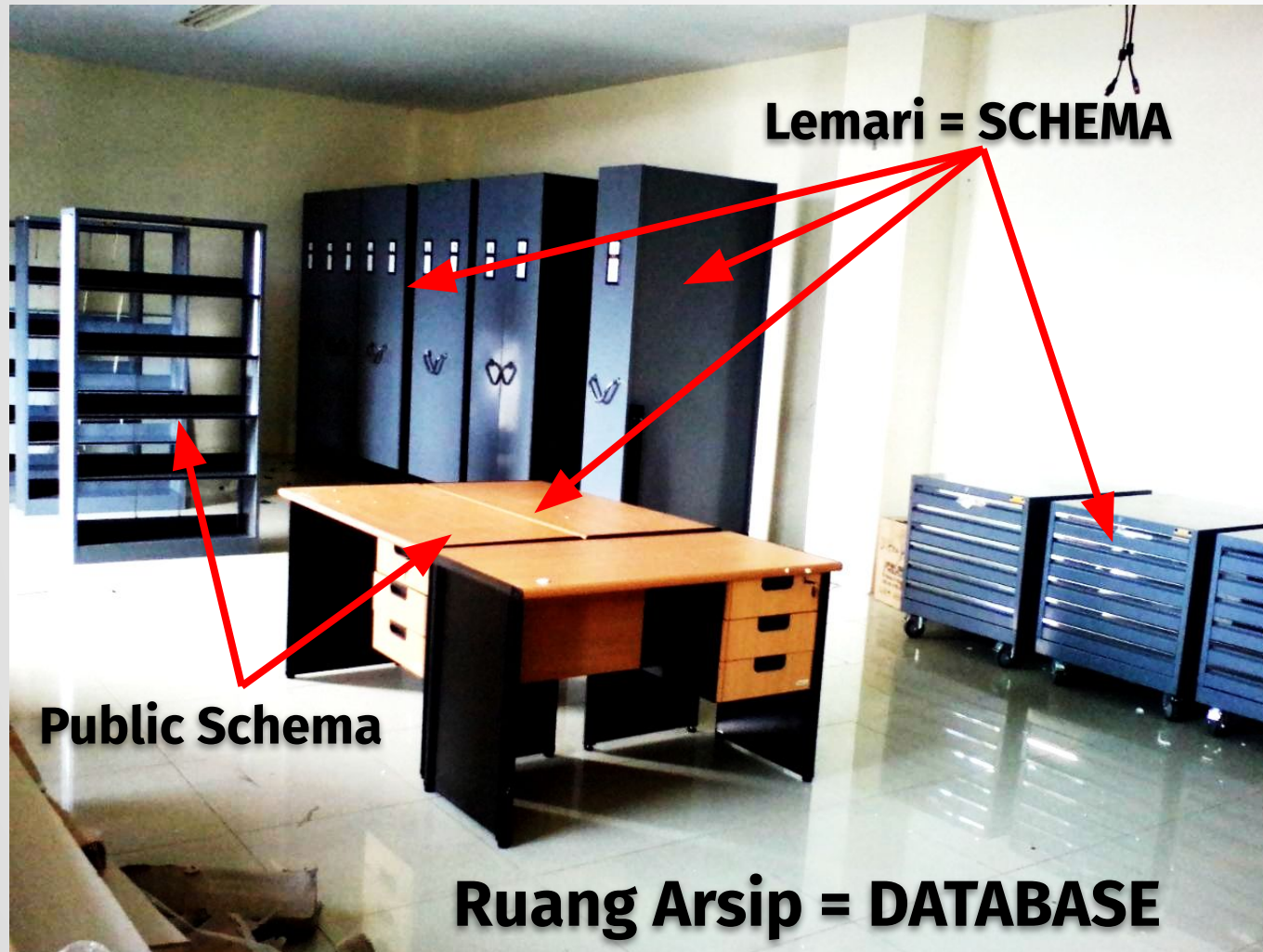
- **Clause:** komponen konstituen atas statement dan query
- **Expression:** komponen yang memproduksi nilai
- **Predicate:** komponen kondisional untuk menghasilkan logika matematika
- **Query:** syntax untuk memanggil data dengan kriteria tertentu
- **Statement:** sekumpulan syntax untuk menghasilkan perintah tertentu

```
UPDATE clause {UPDATE country  
               SET clause {SET population = expression population + 1  
               WHERE clause {WHERE expression name = 'USA';  
                               predicate  
                               }  
               } statement
```





# Komponen Database



**File = TABLE**



**Nomor Inventarisasi  
= INDEX**

## Komponen Database

- **DATABASE** ruang penyimpanan data
- **SCHEMA** lemari penyimpanan pada DATABASE
- **TABLE** data yang disimpan pada SCHEMA tertentu
- **COLUMN** variabel pada suatu tabel
- **INDEX** inventarisasi data agar mudah ditemukan





# SQL Statement

## Data Definition Language

- **CREATE** statement digunakan untuk membuat komponen database
  - **CREATE TABLE** statement untuk membuat TABLE
    - **NOT NULL** kolom tidak boleh memiliki nilai NULL
    - **UNIQUE** kolom harus selalu memiliki penulisan yang berbeda
    - **CHECK** memastikan isian sesuai dengan kondisi tertentu
    - **DEFAULT** menentukan nilai dasar pada tabel
    - **PRIMARY KEY** kolom sebagai identitas, harus NOT NULL dan UNIQUE
    - **FOREIGN KEY** kolom untuk menghasilkan relasi dengan tabel lain
    - **CREATE INDEX** menambahkan penanda, agar proses query lebih cepat
- **DROP** statement digunakan untuk menghilangkan komponen database
- **ALTER** statement digunakan untuk menambah, mengurangi ataupun memodifikasi data eksisting

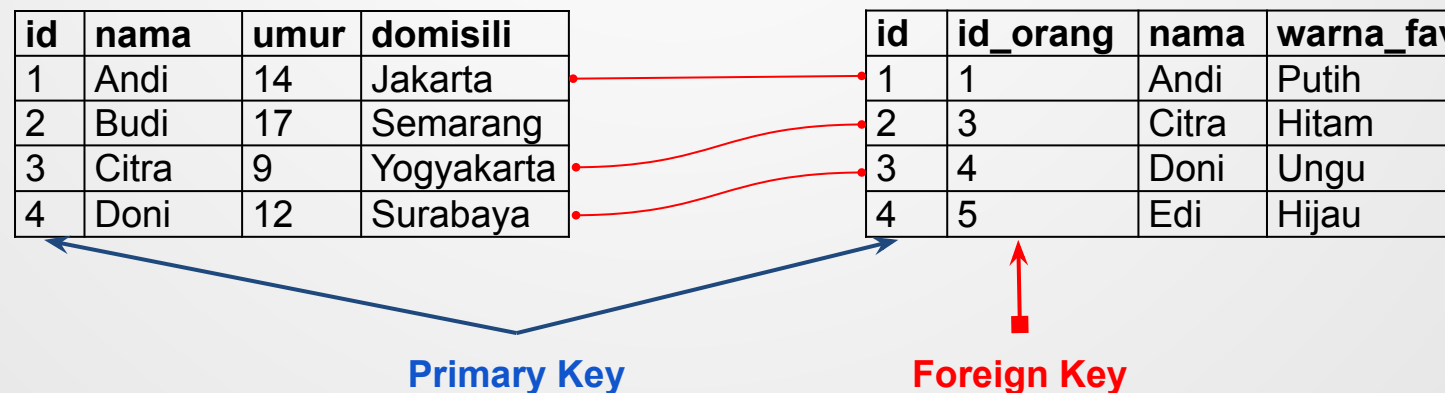
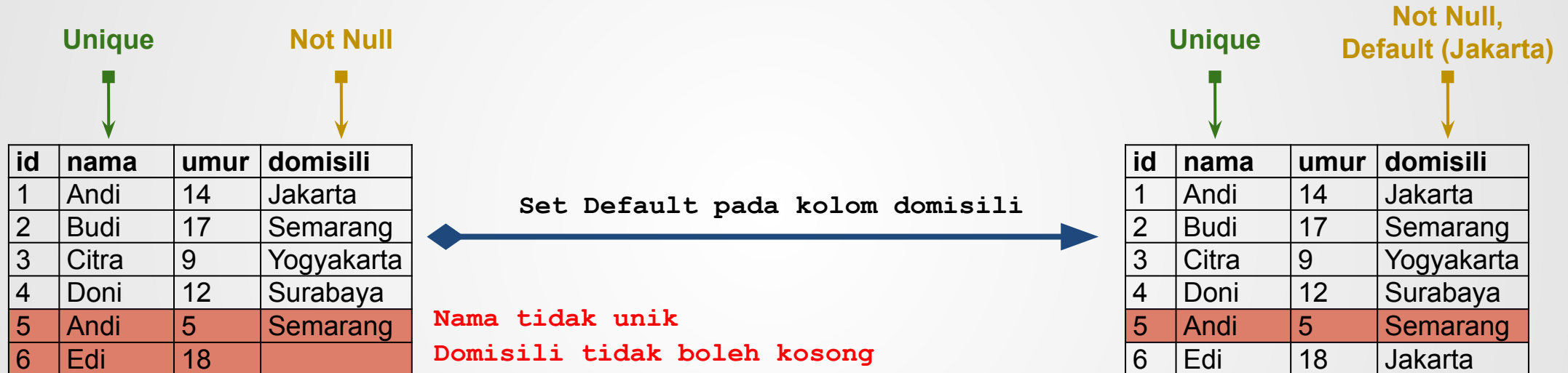






# SQL Statement

## Data Definition Language: Constraint





# SQL Statement

## Data Manipulation Language

- **SELECT** statement untuk memilih data dari database
- **INSERT INTO** statement untuk menambahkan baris data dalam tabel
- **UPDATE** statement untuk memodifikasi baris data dalam tabel
- **DELETE** statement untuk menghapus baris data dalam tabel
- **WHERE** clause untuk filter data
- **JOIN** clause untuk menggabungkan data dari beberapa tabel
- **ORDER BY** statement untuk mengurutkan data
  - **ASC** untuk mengurutkan dari nilai teratas ke bawah
  - **DESC** untuk mengurutkan dari nilai terbawah ke atas
- **GROUP BY** statement untuk mengelompokkan data pada operasi matematika
- **HAVING** clause merupakan WHERE clause yang dapat digunakan pada operasi matematika





# Join Table

nama	umur	domisili
Andi	14	Jakarta
Budi	17	Semarang
Citra	9	Yogyakarta
Doni	12	Surabaya

nama	warna_fav
Andi	Putih
Citra	Hitam
Doni	Ungu
Edi	Hijau

## (INNER) JOIN

nama	umur	domisili	warna_fav
Andi	14	Jakarta	Putih
Citra	9	Yogyakarta	Hitam
Doni	12	Surabaya	Ungu

## LEFT JOIN

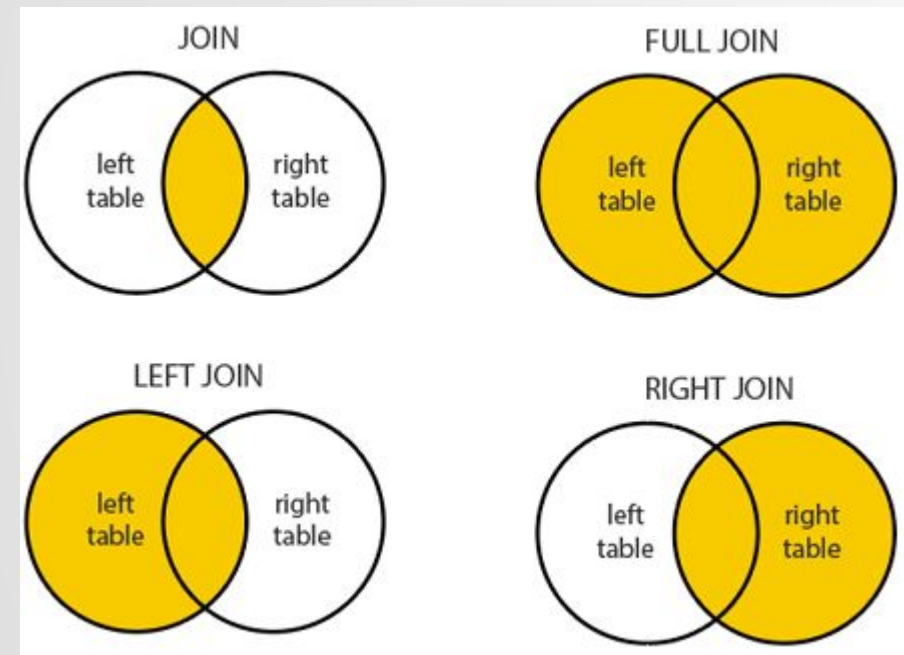
nama	umur	domisili	warna_fav
Andi	14	Jakarta	Putih
Budi	17	Semarang	NULL
Citra	9	Yogyakarta	Hitam
Doni	12	Surabaya	Ungu

## RIGHT JOIN

nama	umur	domisili	warna_fav
Andi	14	Jakarta	Putih
Citra	9	Yogyakarta	Hitam
Doni	12	Surabaya	Ungu
Edi	NULL	NULL	Hijau

## FULL (OUTER) JOIN

nama	umur	domisili	warna_fav
Andi	14	Jakarta	Putih
Budi	17	Semarang	NULL
Citra	9	Yogyakarta	Hitam
Doni	12	Surabaya	Ungu
Edi	NULL	NULL	Hijau



# Spatial Database

**Database spasial** merupakan *general-purpose database* (biasanya berupa *relational database/SQL*) yang diperkaya dengan **data spasial** yang merepresentasikan objek yang didefinisikan secara geometri, bersamaan dengan alat-alat untuk kueri dan analisis datanya,







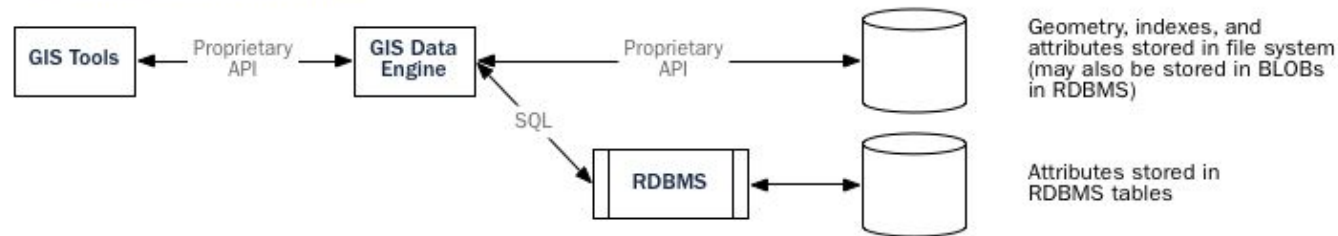
# Database Spasial

## Evolution of GIS Architectures

### First-Generation GIS:



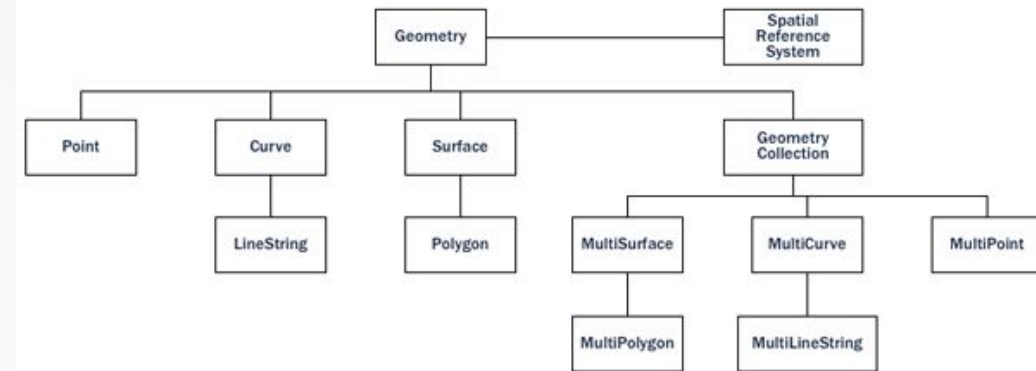
### Second-Generation GIS:



### Third-Generation GIS:



## Geometry Hierarchy

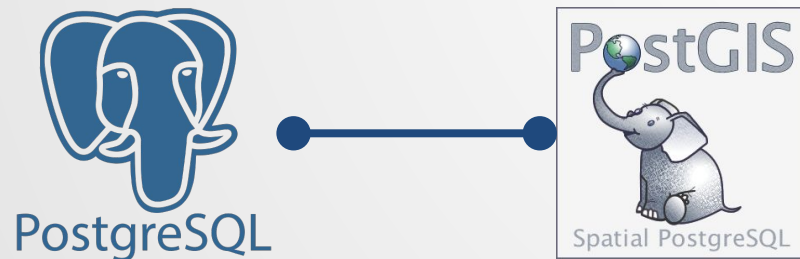


Open/Free	Closed/Proprietary
<ul style="list-style-type: none"><li>• Loading/Extracting<ul style="list-style-type: none"><li>◦ Shp2Pgsql</li><li>◦ ogr2ogr</li><li>◦ Dxf2PostGIS</li></ul></li><li>• Web-Based<ul style="list-style-type: none"><li>◦ Mapserver</li><li>◦ GeoServer (Java-based WFS / WMS -server )</li><li>◦ SharpMap SDK - for ASP.NET 2.0</li><li>◦ MapGuide Open Source (using FDO)</li></ul></li><li>• Desktop<ul style="list-style-type: none"><li>◦ uDig</li><li>◦ QGIS</li><li>◦ mezoGIS</li><li>◦ OpenJUMP</li><li>◦ OpenEV</li><li>◦ SharpMap SDK for Microsoft.NET 2.0</li><li>◦ ZigGIS for ArcGIS/ArcObjects.NET</li><li>◦ GvSIG</li><li>◦ GRASS</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Loading/Extracting<ul style="list-style-type: none"><li>◦ Safe FME Desktop Translator/Converter</li></ul></li><li>• Web-Based<ul style="list-style-type: none"><li>◦ Ionic Red Spider (now ERDAS)</li><li>◦ Cadcorp GeognoSIS</li><li>◦ Iwan Mapserver</li><li>◦ MapDotNet Server</li><li>◦ MapGuide Enterprise (using FDO)</li><li>◦ ESRI ArcGIS Server</li></ul></li><li>• Desktop<ul style="list-style-type: none"><li>◦ Cadcorp SIS</li><li>◦ Microimages TNTmips GIS</li><li>◦ ESRI ArcGIS</li><li>◦ Manifold</li><li>◦ GeoConcept</li><li>◦ MapInfo (v10)</li><li>◦ AutoCAD Map 3D (using FDO)</li></ul></li></ul>

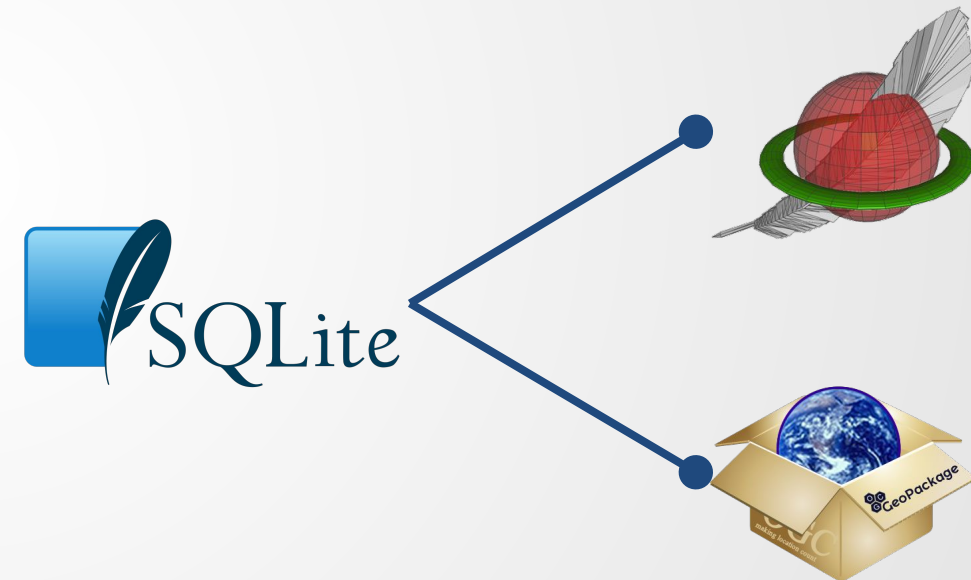


# 3<sup>rd</sup> Generation GIS

Server Database



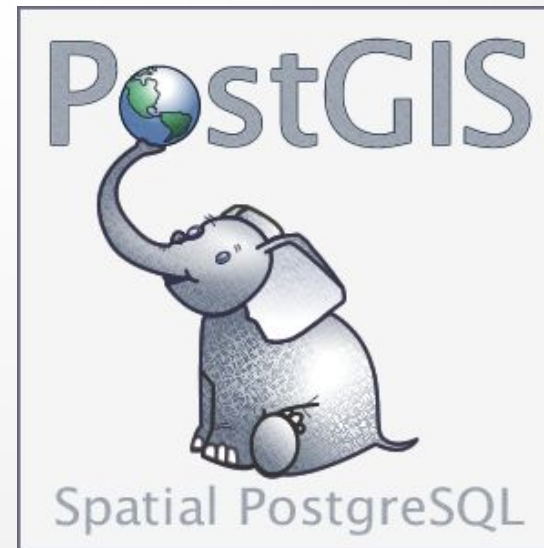
Non-Server Database





# Data Spasial di PostGIS

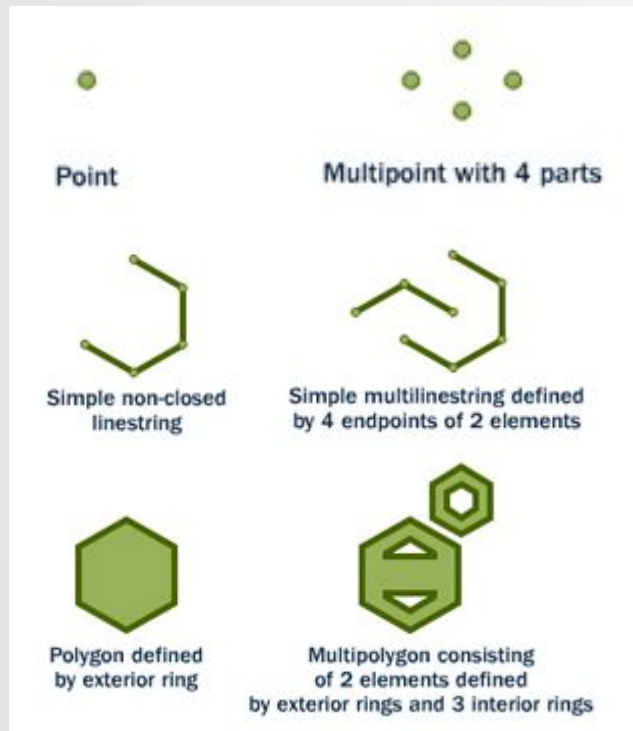
PostGIS adalah spatial database extender untuk PostgreSQL. PostGIS dikembangkan oleh The Open Source Geospatial Foundation (OSGeo), lembaga pengembang software *open source* seperti QGIS, GeoServer, MapServer, OpenLayers, GeoNode dll.





# PostGIS:

## Tipe-tipe data spasial di PostGIS



Penyimpanan data spasial di PostGIS berupa Geometri seperti Point, MultiPoint, LineString, MultiLineString, Polygon, MultiPolygon dan GeometryCollection. Setiap geometri di PostGIS memiliki Spatial Reference ID (SRID)

### Spatial Reference ID

- Sistem Koordinat
- Proyeksi
- Zona
- Datum







# PostGIS:

## Geometri & Spatial Reference ID

### Query Dasar

- ST\_GeometryType(geom)
- ST\_CoordDim(geom)
- ST\_SRID(geom)

### Query: Geometri

- ST\_IsCollection(geom)
- ST\_Dimension(geom)
- ST\_NumGeometries(geom)
- ST\_NumInteriorRings(geom)
- ST\_NPoints(geom)

### Query Validasi

- ST\_IsSimple(geom)
- ST\_IsEmpty(geom)
- ST\_IsClosed(geom)
- ST\_IsRing(geom)
- ST\_IsValid(geom)
- ST\_IsValidReason(geom)

### Query Konversi

- ST\_Transform(geom, SRID)





# PostGIS: Ekstraksi Geometri

- ST\_AsText(geom)
- ST\_AsEWKT(geom)
- ST\_AsGeoJSON(geom)
- ST\_AsGML(geom)
- ST\_AsKML(geom)

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Study site</name>
    <description>Forest inventory study</description>
    <Polygon>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            -94.765829,31.505884,0
            -94.762480,31.506556,0
            -94.763288,31.509076,0
            -94.766736,31.508471,0
            -94.765829,31.505884,0
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </Placemark>
</kml>
```

KML

```
BODY
1 {
2   "type": "FeatureCollection",
3   "features": [
4     {
5       "type": "Feature",
6       "geometry": {
7         "type": "Point",
8         "coordinates": [-111.125, 33.375]
9       },
10      "properties": {
11        "trackid": "AA-1234",
12        "reported_dt": "12/31/2019 23:59:59"
13      }
14    },
15    {
16      "type": "Feature",
17      "geometry": {
18        "type": "Point",
19        "coordinates": [-113.675, 35.875]
20      },
21      "properties": {
22        "trackid": "AA-1234",
23        "reported_dt": "12/31/2019 23:59:59"
24      }
25    }
26  ]
27 }
```

GeoJSON



Google Maps



Google Earth Engine



# Get in touch with me at



Ahmad Zaenun Faiz  
**GIS & Data Specialist**



Ahmad Zaenun Faiz



[github.com/ahmadzfaiz](https://github.com/ahmadzfaiz)



[azfaiz.com](http://azfaiz.com)

