



## ***GIS e Geo WEB: piattaforme e architetture***

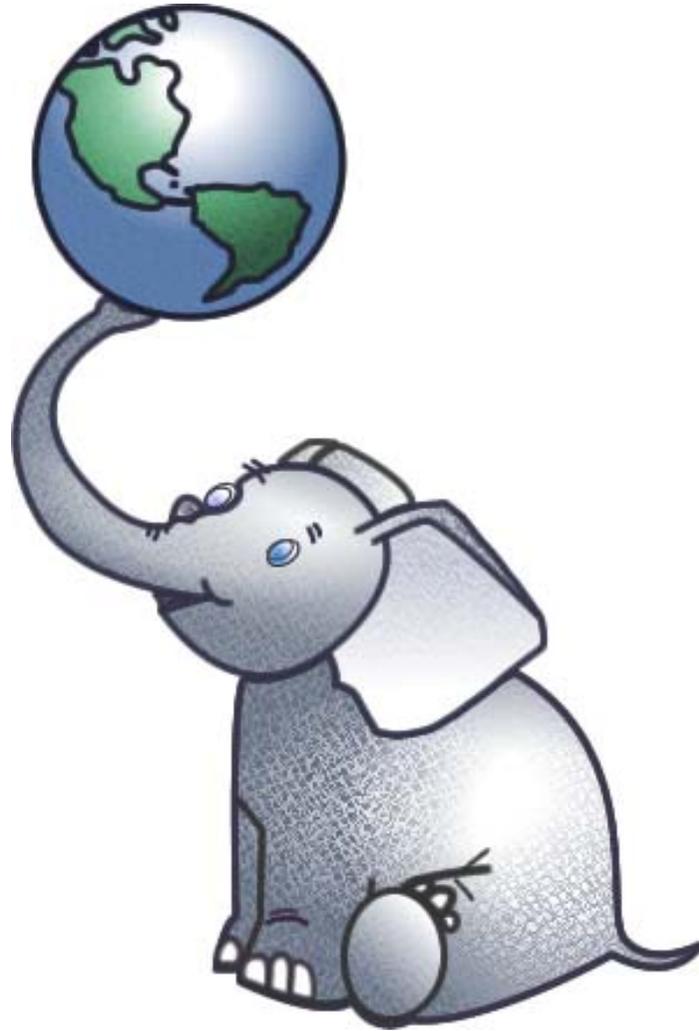
***Docente: Cristoforo Abbattista  
eMail: [abbattista@planetek.it](mailto:abbattista@planetek.it)***



# III lezione

## Componenti e linguaggi

*GIS e Geo WEB: piattaforme e architetture*



# **Vediamo un po'**

## **Demo PostgreSQL & PostGIS**

## PostGIS

### PostGIS

- Estensione spaziale del database PostgreSQL
- Supporta le librerie:
  - GEOS (Geometry Engine - Open Source )
  - PROJ4 (Cartographic projections)

## Installazione

### Fasi dell'installazione

- Nell'installazione di postgis sarà creato il database *postgis* che potrà essere assunto come template per tutti i database con estensione spaziale
- Nel database *postgis* ci saranno due tabelle:
  - *geometry\_columns* contiene l'associazione di una tabella al proprio sistema di riferimento geografico
  - *spatial\_ref\_sys* contiene tutti i sistemi di riferimento gestiti da postGIS

## Geometrie

### Tipologia di geometrie gestite

- POINT
- LINESTRING
- POLYGON
- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON
- GEOMETRYCOLLECTION

## Inserimento dati

### Comando SQL

- `INSERT INTO roads (road_id, roads_geom, road_name) VALUES (1,GeomFromText('LINESTRING(191232 243118,191108 243242)',-1),'Jeff Rd');`

### Da shape

- Per caricare i dati da uno shape ESRI in una tabella in postGIS si utilizza il comando *shp2pgsql*

## shp2pgsql

### Opzioni del comando shp2pgsql

Il comando shp2pgsql ha le seguenti opzioni:

- **-d** Cancella la tabella prima di crearne una nuova
- **-a** Aggiunge i dati in una tabella con la stessa struttura dello shape
- **-c** (default) Crea una nuova tabella
- **-p** Genera solo il codice di creazione della tabella, senza l'inserimento dei dati
- **-D** Usa il comando *dump* di PostgreSQL invece dell'insert. Si usa con una mole eccessiva di dati da caricare
- **-s** <SRID> Crea e popola la tabella con questo specifico sistema di riferimento.
- **-k** Mantiene il nome originale degli attributi conservando anche le condizioni di Upper Case.
- **-i** Forza tutti gli interi a 32-bit
- **-I** Crea un indice GiST sulla colonna della geometria della tabella
- **-w** Crea l'uscita nel formato WKT (per usarlo con le vecchie versioni di PostGIS 0.x )
- **-W** <encoding> Specifica l'encoding del dbf. Se usato, tutti gli attributi saranno convertiti da questo encoding all'UTF-8

## pgsql2shp

### pgsql2shp

- Esporta una tabella di PostGIS in uno shape

### Opzioni del comando pgsq2shp

Il comando pgsq2shp ha le seguenti opzioni:

- -f <filename> Scrive il file di uscita con questo nome.
- -h <host> Il server su cui c'è il database a cui connettersi.
- -p <port> La porta su cui ci si connette.
- -P <password> La password per connettersi al database.
- -u <user> L'username con cui ci si connette.
- -g <geometry column> Il nome delle colonna che contiene la geometria che si vuole esportare (nel caso di tabelle con multigeometrie).
- -b Usa un cursore binario
- -r Non cancella il campo gid
- -d Permette la compatibilità con versioni precedenti di postGIS.

## Indici

### Gestione degli indici

- Gli indici migliorano la gestione delle ricerche su database con un eccessivo numero di dati.
- Senza gli indici le ricerche richiederebbero un'analisi sequenziale dei record delle tabelle nel database.
- L'indicizzazione organizza i dati in strutture ad albero che possono essere facilmente attraversate per raggiungere un particolare dato.
- PostgreSQL supporta diversi tipi di indici:
  - B-Tree usato per dati che possono essere ordinati lungo un asse come numeri, lettere e date; (i dati cartografici non sono di questo tipo)
  - R-Tree suddivide i dati in rettangoli e sotto rettangoli. Può essere usato per dati cartografici ma non ha le stesse performance dell'indice GiST
  - GiST suddivide in dati e li raggruppa considerando le adiacenze, le sovrapposizioni e le inclusioni.
- PostGIS usa l'indice GiST

## Ottimizzazione

### Creazione indice spaziale

Per creare un indice GiST su una tabella si usa il seguente comando:

- `CREATE INDEX [indexname] ON [tablename] USING GIST ( [geometryfield] );`

### Creazione Cluster

- Nel caso ci siano tabelle principalmente di sola lettura che utilizzano un solo indice si può usare il metodo `CLUSTER`
- Questo metodo riordina tutte le righe seguendo l'ordine dell'indice
- Il comando per aggiungere questo metodo è il seguente:
  - `CLUSTER my_geom_index ON my_table;`

## Funzioni

### Funzioni di analisi spaziale

- Alcune funzioni di analisi spaziale sono le seguenti:
  - ST\_Distance(geometry, geometry) non usa gli indici.
  - ST\_DWithin(geometry, geometry, float)
  - ST\_Equals(geometry, geometry)
  - ST\_Disjoint(geometry, geometry)
  - ST\_Intersects(geometry, geometry)
  - ST\_Touches(geometry, geometry)
  - ST\_Crosses(geometry, geometry)
  - ST\_Within(geometry A, geometry B)
  - ST\_Overlaps(geometry, geometry)

## Esercitazione

### Esercitazione

- Installare postGIS
- Creare un database con estensione spaziale
- Importare uno shape nel database creato
- Testare alcune query o funzioni sulla tabella creata
- Esportare la tabella in uno shape



**Ora comandiamo noi!**

**I linguaggi di programmazione**



# Sempre più in alto

## HTML & CSS & Javascript

# HTML, Javascript e CSS

## La pagina web

- Ogni pagina web è l'insieme di codice
  - HTML, Javascript e CSS
- Per tutte le applicazioni web geografiche e non conoscere queste tre tecnologie è importante

## Come si scrive una pagina web

- Con un CMS
- Con un editor di testo evoluto che aiuti la scrittura del codice
- Con un editor di testo semplice per scrivere e personalizzare spezzoni di codice tratti da tutorial e guide;



## Demo HTML & CSS & Javascript

## Linguaggi

### HTML

- **HTML (HyperText Markup Language)** è il primo linguaggio ideato (e il più utilizzato) per realizzare le pagine internet.

### CSS

- **CSS (Cascading Style Sheet)** è il linguaggio usato per definire la rappresentazione delle pagine WEB

### JavaScript

- Linguaggio di programmazione interpretato dal Browser client

## HTML

### HTML

- L'HTML è un linguaggio che descrive il **contenuto** di una pagina web ma non la forma, che viene descritta dal CSS
- L'estensione dei file è .htm o .html
- La sintassi è stabilita dal World Wide Web Consortium (W3C)

### Componenti

- Tag: unità fondamentale che permette l'interpretazione da parte del browser.
- Attributi: proprietà del tag
- Valori: valore dell'attributo

## Struttura HTML

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>PROVA</title>
  </head>
  <body>
    <!-- Scriveremo qui -->
    Qui il nostro contenuto

  </body>
</html>
```

## Esercitazione HTML

- Modificare la pagina HTML iniziale
- Inserire immagini
- Creare un pagina di un form

## CSS

### CSS

- Ha permesso di separare all'interno di una pagina Web i contenuti dalla formattazione
- Migliora le performance dell'applicazione Web

### Richiamare i CSS

- richiamarli direttamente nelle pagine come stile dell'elemento usando l'attributo style
- utilizzare l'elemento `<style>` nell'head dei nostri documenti XHTML
- richiamando una pagina di stili esterna attraverso l'elemento `<link>`
- utilizzando la direttiva `@import` in `<style>`.

## Struttura CSS

```
div.olMap {  
  z-index: 0;  
  padding: 0px!important;  
  margin: 0px!important;  
}  
  
div.olMapViewport {  
  text-align: left;  
}  
  
div.olLayerDiv {  
  -moz-user-select: none;  
}  
  
.olLayerGoogleCopyright {  
  left: 2px;  
  bottom: 2px;  
}  
  
.olLayerGooglePoweredBy {  
  left: 2px;  
  bottom: 15px;  
}  
  
.olControlAttribution {  
  font-size: smaller;  
  right: 3px;  
  bottom: 4.5em;  
  position: absolute;  
  display: block;  
},
```

## Esercitazione CSS

- Inserire CSS
- Modificare e aggiungere nuovi elementi al CSS

# JavaScript

## JavaScript

### JavaScript

- Linguaggio interpretato quindi non compilato.
- Sintassi analoga a quello compilato, quindi con la possibilità di utilizzare strutture di controllo, cicli ecc.

### Richiamare Javascript

- Inserire il codice direttamente nel tag script
- Richiamare un file javascript esterno

## Struttura JavaScript

---

```
* {Object} symbolizer hash
*/
createSymbolizer: function(feature) {
  var style = this.createLiterals(
    OpenLayers.Util.extend({}, this.defaultStyle), feature);

  var rules = this.rules;

  var rule, context;
  var elseRules = [];
  var appliedRules = false;
  for(var i=0, len=rules.length; i<len; i++) {
    rule = rules[i];
    // does the rule apply?
    var applies = rule.evaluate(feature);

    if(applies) {
      if(rule instanceof OpenLayers.Rule && rule.elseFilter) {
        elseRules.push(rule);
      } else {
        appliedRules = true;
        this.applySymbolizer(rule, style, feature);
      }
    }
  }

  // if no other rules apply, apply the rules with else filters
  if(appliedRules == false && elseRules.length > 0) {
    appliedRules = true;
  }
}
```

## Esercitazione JavaScript

- Creare file JavaScript
- Modificare le pagine HTML con i medesimi file
- Creare pagine web complete di CSS e JavaScript



# Fine III lezione

## Componenti e linguaggi

*GIS e Geo WEB: piattaforme e architetture*