



GIS e Geo WEB: piattaforme e architetture

***Docente: Cristoforo Abbattista
eMail: abbattista@planetek.it
Esercitazioni: Marianna Carbone***



III lezione

Componenti e linguaggi

GIS e Geo WEB: piattaforme e architetture



PostgreSQL & PostGIS

PostgreSQL/PostGIS

PostgreSQL/PostGIS

PostgreSQL

- Cosa è
- Installazione
- Esercitazione

PostGIS

- Cosa è
- Installazione
- esercitazione

PostgreSQL

PostgreSQL

Storia

- Basato su un progetto dell'Università di Berkeley
- Dalla versione 6, il software viene sviluppato da volontari di tutto il mondo

Caratteristiche principali

- Database relazionale ad oggetti (ORDBMS)
- OpenSource
- Supporta gran parte dello standard SQL (query, viste, trigger)
- Permette di definire nuovi tipi di dati
- Supporta molti linguaggi: PL/pgsql, Perl, Python, R, C, C++

PostgreSQL

Architettura

- Una sessione di PostgreSQL consiste nei seguenti processi:
 - Un processo server che gestisce i file del database, accetta le connessioni dei vari client e effettua le azioni sul database. Questo processo si chiama *postgres*
 - Un'applicazione client che richiede le operazioni

Sistemi Operativi

- Linux
- Unix
- Windows
- Download <http://www.postgresql.org>

Installazione

Fasi dell'installazione

- Scaricare il pacchetto relativo al proprio sistema operativo
- Creazione dell'utente amministratore di PostgreSQL
- Creazione del database cluster (collezione di database amministrati da una singola istanza di PostgreSQL)
- Creazione del database *postgres* e del database *template1*
- Avvio del servizio PostgreSQL

Configurazioni

- postgresql.conf
 - Path dei file di configurazione
 - Proprietà della connessione
 - Memoria e log
- pg_hba.conf
 - Gestione autenticazione client
- pg_ident.conf
 - Gestione autenticazione user

postgresql.conf

```
#  
# CONNECTIONS AND AUTHENTICATION  
#-----  
  
# - Connection Settings -  
  
#listen_addresses = 'localhost' # what IP address(es) to listen on;  
# comma-separated list of addresses;  
# defaults to 'localhost', '*' = all  
# (change requires restart)  
port = 5432 # (change requires restart)  
max_connections = 100 # (change requires restart)  
# Note: Increasing max_connections costs ~400 bytes of shared memory per  
# connection slot, plus lock space (see max_locks_per_transaction). You might  
# also need to raise shared_buffers to support more connections.  
#superuser_reserved_connections = 3 # (change requires restart)  
#unix_socket_directory = '' # (change requires restart)  
#unix_socket_group = '' # (change requires restart)  
#unix_socket_permissions = 0777 # begin with 0 to use octal notation  
# (change requires restart)  
#bonjour_name = '' # defaults to the computer name  
# (change requires restart)  
  
# - Security and Authentication -  
  
#authentication_timeout = 1min # 1s-600s  
#ssl = off # (change requires restart)  
#ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers  
# (change requires restart)  
#password_encryption = on
```


pg_hba.conf

Autenticazione Client

- Pg_hba (host-based authentication) costituito da un set di righe che rappresentano le connessioni. Ci sono diversi tipi di forme:
 - local database user auth-method [auth-option]
 - host database user CIDR-address auth-method [auth-option]
 - hostssl database user CIDR-address auth-method [auth-option]
 - hostnossl database user CIDR-address auth-method [auth-option]
 - host database user IP-address IP-mask auth-method [auth-option]
 - hostssl database user IP-address IP-mask auth-method [auth-option]
 - hostnossl database user IP-address IP-mask auth-method [auth-option]

Pg_hba.conf

Metodi di autenticazione

- **Trust**
- **Reject**
- **Md5**
- **Password**
- **Gss**
- **Sspi**
- **Krb5**
- **Ident**
- **Ldap**
- **Radius**
- **Cert**
- **Pam**

Pg_ident.conf

```
#
# This file controls PostgreSQL ident-based authentication. It maps
# ident user names (typically Unix user names) to their corresponding
# PostgreSQL user names. Records are of the form:
#
# MAPNAME IDENT-USERNAME PG-USERNAME
#
# (The uppercase quantities must be replaced by actual values.)
#
# MAPNAME is the (otherwise freely chosen) map name that was used in
# pg_hba.conf. IDENT-USERNAME is the detected user name of the
# client. PG-USERNAME is the requested PostgreSQL user name. The
# existence of a record specifies that IDENT-USERNAME may connect as
# PG-USERNAME. Multiple maps may be specified in this file and used
# by pg_hba.conf.
#
# This file is read on server startup and when the postmaster receives
# a SIGHUP signal. If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect. You can use
# "pg_ctl reload" to do that.
#
# Put your actual configuration here
# -----
#
# No map names are defined in the default configuration. If all ident
# user names and PostgreSQL user names are the same, you don't need
# this file. Instead, use the special map name "sameuser" in
# pg_hba.conf.
#
# MAPNAME IDENT-USERNAME PG-USERNAME
```

Configurazione

Parametri

LC_COLLATE	Ordine delle stringhe
LC_CTYPE	Classificazione dei caratteri
LC_MESSAGES	Linguaggio dei messaggi
LC_MONETARY	Formattazione delle cifre
LC_NUMERIC	Formattazione dei numeri
LC_TIME	Formattazione delle date

Character Set

- Permette di memorizzare all'interno del database molti set di caratteri denominati encodings
- Ogni encoding deve essere compatibile con LC_CTYPE e LC_COLLATE definito sul server

Encoding

Name	Description	Language	Server?	Bytes/Char	Aliases
LATIN6	ISO 8859-10, ECMA 144	Nordic	Yes	1	ISO885910
LATIN7	ISO 8859-13	Baltic	Yes	1	ISO885913
LATIN8	ISO 8859-14	Celtic	Yes	1	ISO885914
LATIN9	ISO 8859-15	LATIN1 +Euro + accents	Yes	1	ISO885915
LATIN10	ISO 8859-16, ASRO SR 14111	Romanian	Yes	1	ISO885916
SJIS	Shift JIS	Japanese	No	1-2	Mskanji, ShiftJIS, WIN932, Windows932
SHIFT_JIS_2004	Shift JIS, JIS X 0213	Japanese	No	1-2	
SQL_ASCII	unspecified (see text)	any	Yes	1	
UTF8	Unicode, 8-bit	all	Yes	1-4	Unicode
WIN866	Windows CP866	Cyrillic	Yes	1	ALT
WIN874	Windows CP874	Thai	Yes	1	
WIN1250	Windows CP1250	Central European	Yes	1	
WIN1251	Windows CP1251	Cyrillic	Yes	1	WIN
WIN1252	Windows CP1252	Western European	Yes	1	
WIN1253	Windows CP1253	Greek	Yes	1	
WIN1254	Windows CP1254	Turkish	Yes	1	
WIN1255	Windows CP1255	Hebrew	Yes	1	
WIN1256	Windows CP1256	Arabic	Yes	1	
WIN1257	Windows CP1257	Baltic	Yes	1	
WIN1258	Windows CP1258	Vietnamese	Yes	1	ABC, TCVN, TCVN5712, VSCII

Indici

Indici

- Indici permettono di velocizzare le ricerche all'interno del database.
 - 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)
 - Hash: gestisce solo semplici confronti di uguaglianza.
 - GIST (Generalized Search Tree): costruisce una struttura di accesso al dato ad albero bilanciata
 - GIN (Generalized Inverted Index): costruisce una struttura memorizzando un set di valori key e posting list, in cui posting list è una serie di righe in cui è presente la chiave. Ogni indice può contenere molte chiavi così come molte chiavi possono essere in multiple posting list.

Ottimizzazione

Metodi

- **Le operazioni per migliorare le performance del database sono le seguenti:**
 - **Analyze**
 - **Vacuum**
 - **Reindex**

Ottimizzazione

Analyze

- Aggiorna le statistiche usate dal sistema per gestire al meglio le query (ANALYZE)

Vacuum

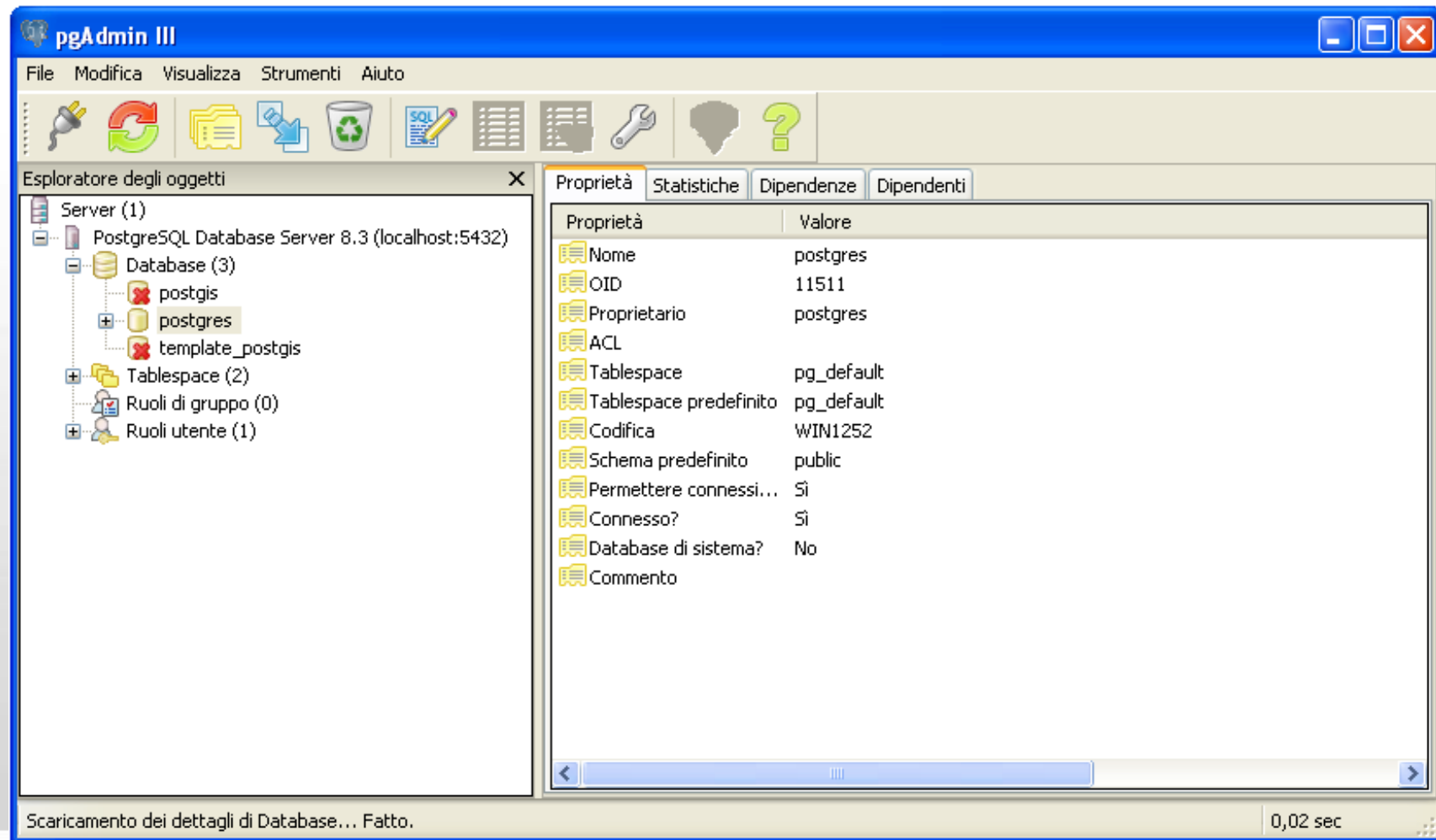
- Recupera e riutilizza lo spazio del disco occupato da righe aggiornate o cancellate
- Aggiorna le statistiche usate dal sistema per gestire al meglio le query (ANALYZE)

Reindex

- Ricostruisce gli indici del database

PostgreSQL

PgAdmin III



PostgreSQL

Command line interface

```
C:\Programmi\PostgreSQL\8.3\bin>psql -U postgres -W postgres
Password for user postgres:
Welcome to psql 8.3.7, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

Warning: Console code page (850) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.

postgres=# _
```

Esercitazione

Esercitazione

- Installare postgresQL
- Creare un database sia con pgAdmin III che da command line
- Creare delle tabelle
- Fare query SQL
- Gestire il server

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 (nelle versioni precedenti alla 9) 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



Comandiamo noi!

I linguaggi di programmazione



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