



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

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



## II lezione Componenti e linguaggi

*GIS e Geo WEB: piattaforme e architetture*





**Conservare in un luogo sicuro**

**Basi di Dati**

# Il nostro mondo



## Classifichiamola



# Il Database

## DBMS – Database Management System

- Edgar CODD alla fine degli anni 60
  - PostgreSQL, MS SQL Server, ORACLE, Informix, ACCESS, SYBASE, MySQL, etc.
- È un archivio elettronico di dati
  - Prevede un programma di interfaccia per la definizione (DDL) e la manipolazione (DML) dei dati.
  - Esempio: Rubrica Telefonica
- I dati sono organizzati in tabelle costituite da:
  - colonne o campi
  - righe o record
- Ogni campo ha:
  - Un nome
  - Un tipo di dato (numerico, testo,...)
  - Degli attributi per il tipo scelto (dimensione, formato,...)

Nome	Cognome	Telefono	Indirizzo
Mario	Rossi	06942873	Via Conte68
Paolo	Bianchi	09247362	Corso Italia,15
Carlo	Verdi	02834472	Piazza Indipendenza,4

# Il Database


## Com'è organizzato

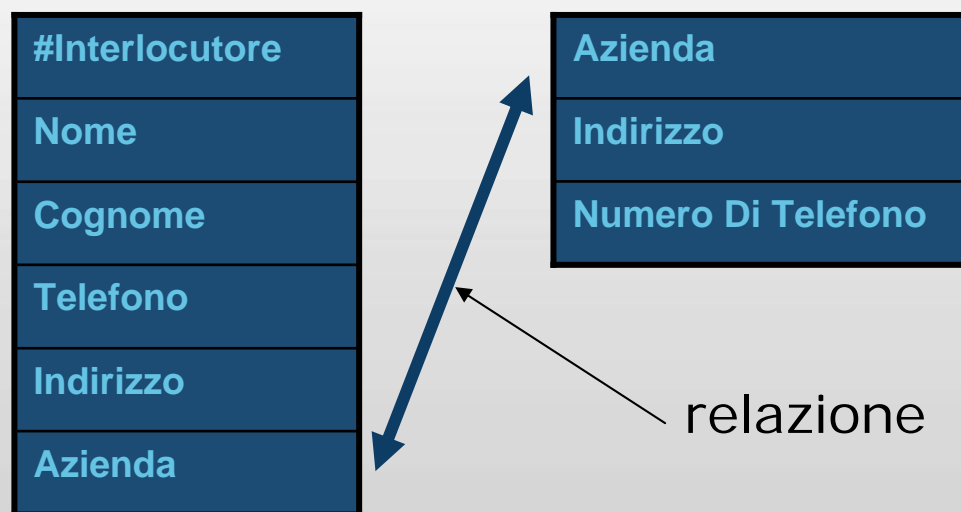
- È opportuno che ogni record abbia **chiave di ricerca primaria**
  - un campo o una combinazione di campi
  - Attenti alla scelta!
    - Un contatore?
- Un **indice** per il recupero veloce dei dati mediante la ricerca random invece di quella sequenziale (TABLE SCAN).
- La chiave primaria è un indice
- Un indice può anche insistere su campi diversi.

Nome	Cognome	Telefono	Indirizzo
Mario	Rossi	06942873	Via Conte68
Paolo	Bianchi	09247362	Corso Italia,15
Carlo	Verdi	02834472	Piazza Indipendenza,4

# Il Database

## Relazionale

- Le tabelle sono correlate mediante relazioni tra campi.
- Aggregazione di dati attraverso le relazioni
- Tipi di relazione
  - Uno a uno
  - Uno a molti (chiave esterna o Foreign Key)
  - Molti a molti
- Come si descrivono a livello fisico? 
  - Attraverso un campo in più





# Il Database

## Integrità referenziale

- Non vogliamo che il database abbia dati non congruenti
  - Interlocutori collegati ad aziende non registrate
  - Aziende senza interlocutori
  - Eliminazione di aziende con interlocutori
- Il sistema si preoccupa di mantenere l'integrità referenziale
  - Vengono posti dei vincoli che possono complicare la gestione del sistema



# Disegnare un Database

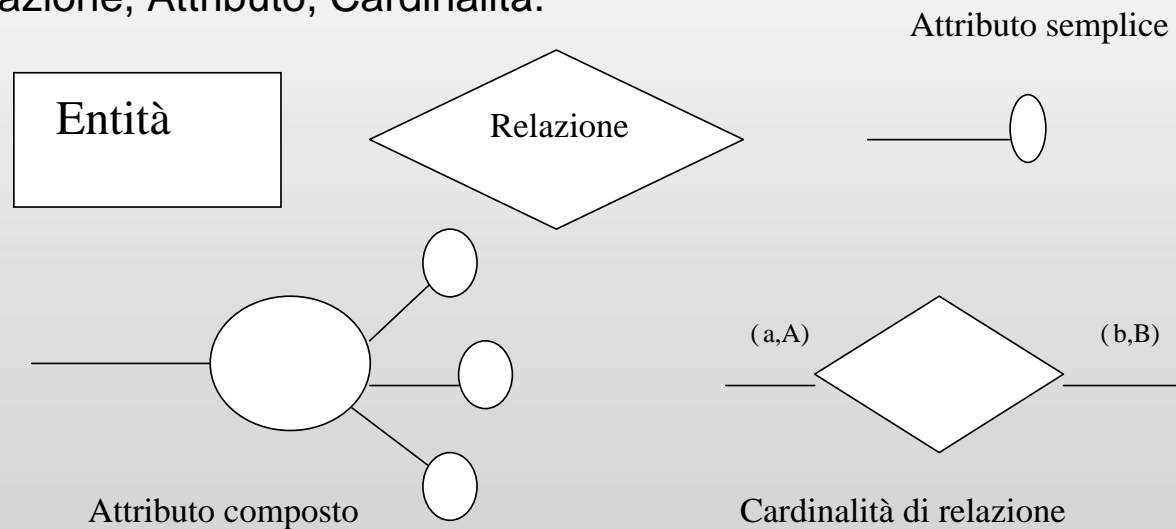
## Le domande

- Quale è lo scopo del database?
- Quali sono le entità principali?
- Quali sono i dati che caratterizzano tali entità?
- In che modo sono correlate tali entità?
  - Verifica della struttura ed eventuali correzioni

# Il modello Entità-Relazione

## Un modello concettuale

- Il modello ER e' un modello concettuale di dati
  - descrive la realtà in maniera semplice
  - indipendente dalla organizzazione dei dati nel computer.
- Gli elementi principali del modello sono
  - Entità, Relazione, Attributo, Cardinalità.



# Il modello Entità-Relazione

## Le Entità

- Le Entità sono classi di oggetti distinguibili tra loro ai fini del mondo di interesse che si intende modellare.
- Un'occorrenza (o istanza) di un'entità è un oggetto della classe

STUDENTE

CORSO

PROFESSORE

## Le Relazioni

- Le Relazioni (o Associazioni) rappresentano legami logici fra due o più entità
- Ci possono essere più relazioni fra diverse entità o relazioni ricorsive

STUDENTE

ESAMI

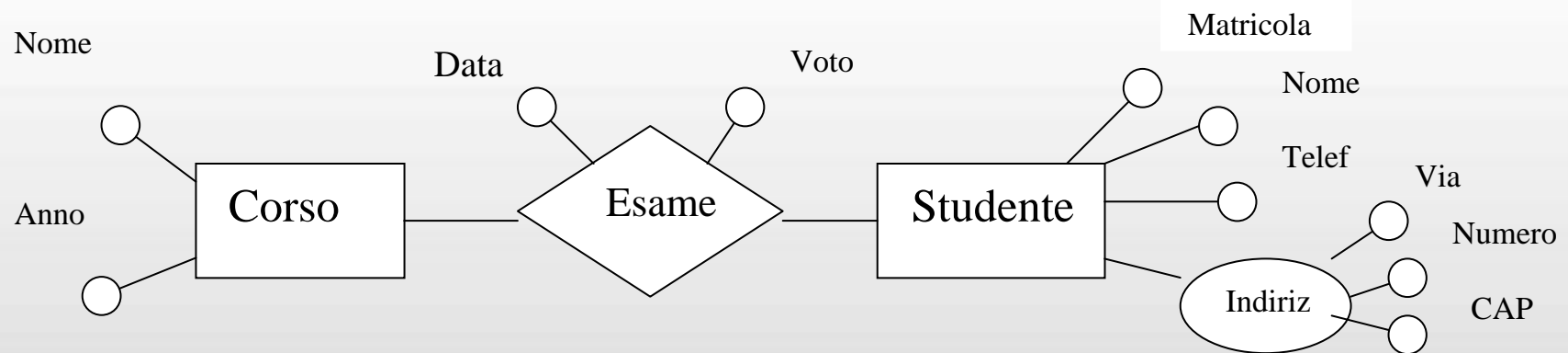
CORSO



# Il modello Entità-Relazione

## Gli attributi

- Sono le proprietà delle entità e delle relazioni



# Il modello Entità-Relazione

## Cardinalità delle relazioni

- Determina il numero minimo e massimo di istanze di relazione per ogni entità a cui la relazione si riferisce
- Nell'esempio
  - ad ogni impiegato si possono assegnare da 1 a 5 incarichi
  - ogni incarico può essere assegnato al più a 50 impiegati
- La cardinalità può essere
  - uno a uno (1-1)
  - uno a molti (1-N)
  - molti a molti (M-N).



# Disegnare un Database

## I tipi di dati

- **Testo:** combinazione di caratteri e numeri fino a 255 caratteri
- **Numerico:** valori numerici tra cui (Byte, Intero, Intero lungo, Precisione singola, Precisione doppia, ecc.)
- **Data/Ora:** valori di data e ora in vari formati
- **Contatore:** numero sequenziale incrementato automaticamente
- **Boolean**
- **BLOB, Text:** Immagine o campi lunghi

# Disegnare un Database

## Gli attributi

- **Dimensione:** numero max di caratteri
- **Formato:** memorizzare i dati in modi differenti
- **Valore predefinito:** specifica un valore iniziale
- **Valido se..:** intervalli di validità dei valori immessi
- **Richiesto**
- **Indicizzato:** sul campo esiste un indice
- **Consenti lunghezza zero:** permette di avere valore NULL



## Standard (Structured) Query Language

- **DDL (Data Definition Language)**
  - Creare la struttura del database
    - Tabelle con i relativi campi e i relativi attributi
    - Indici
    - Trigger
    - Stored Procedure
- **DML (Data Manipulation Language)**
  - Ricercare i dati nel database
  - Inserire i dati nel database
  - Modificare i dati nel database
  - Eliminare i dati dal database
  - Eseguire Stored Procedure

## Esempi

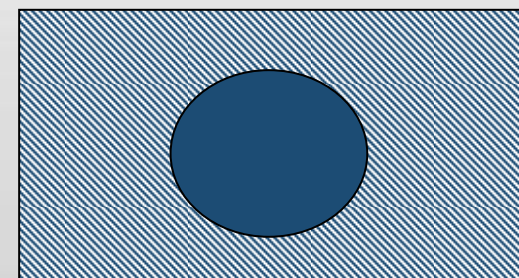
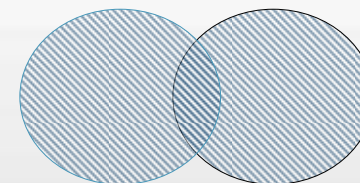
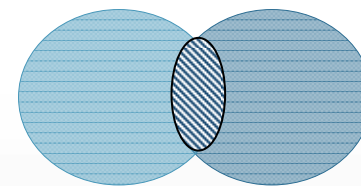
- **DML (Data Manipulation Language)**

- Trovare il nome e il cognome dell'interlocutore e il numero di telefono dell'azienda in cui lavora



## Operatori

- Su una sola tabella
  - Proiezione
  - selezione
- Su più tabelle
  - Unione
  - Intersezione
  - Join
- Logici
  - AND, OR, NOT
- Confronto
  - >, <, =, !=, LIKE
- NULL, NOT NULL



## Gestire tabelle nel database

- CREATE TABLE tabella (campoX tipo [(dimensioni)] [NOT NULL])
- ALTER TABLE tabella {ADD {COLUMN tipo campo[(dimensioni)] [NOT NULL] |  
ALTER COLUMN tipo campo[(dimensioni)] |  
DROP {COLUMN campo} }
- DROP TABLE tabella

## Gestire indici nel database

- CREATE [UNIQUE] INDEX indice ON tabella (campo)
- DROP INDEX indice ON tabella

## Gestire viste nel database

- CREATE VIEW vista [lista campi] AS istruzioneselect
- DROP VIEW vista



# SQL - DML

## Ricerca i dati nel database

- SELECT lista colonne FROM lista tabelle utilizzate
- [WHERE lista di condizioni] [GROUP BY... ] [HAVING... ] [ORDER BY... ]

## Inserire i dati nel database

- INSERT INTO tabella utilizzata [lista campi] VALUES lista valori

## Modificare i dati nel database

- UPDATE tabella SET (lista coppie campi=valori) WHERE lista condizioni

## Eliminare i dati dal database

- DELETE FROM tabella utilizzata WHERE lista condizioni



# **Vediamo un po'**

## **Demo PostgreSQL & PostGIS**

# PostgreSQL/PostGIS

## PostgreSQL/PostGIS

### PostgreSQL

- Cosa è
- Installazione
- Esercitazione

### PostGIS

- Cosa è
- Installazione
- esercitazione

## 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**
- **Crypt**
- **Password**
- **Gss**
- **Sspi**
- **Krb5**
- **Ident**
- **Ldap**
- **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 (non configurabile)
- LC\_CTYPE (non configurabile)
- LC\_MESSAGES
- LC\_MONETARY
- LC\_NUMERIC
- LC\_TIME

### Character Set

- Permette di memorizzare all'interno del database molti set di caratteri denominati encodings
- Ogni encoding deve essere compatibile con LC\_TYPE 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	01-feb	Mskanji, ShiftJIS, WIN932, Windows932
SHIFT_JIS_2004	Shift JIS, JIS X 0213	Japanese	No	01-feb	
SQL_ASCII	unspecified (see text)	any	Yes	1	
UTF8	Unicode, 8-bit	all	Yes	01-apr	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)
  - 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 (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

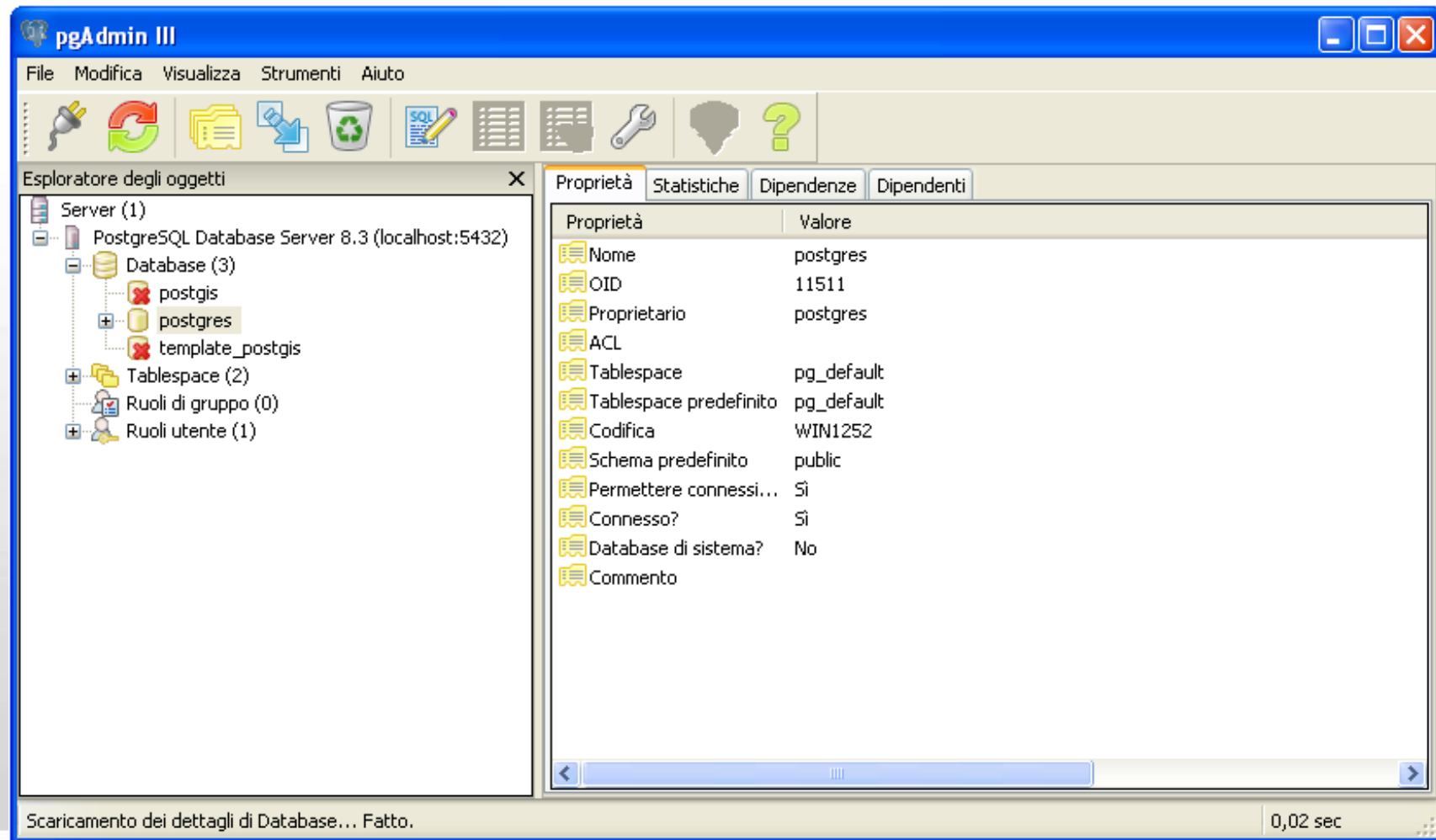
- Recupera e riutilizza lo spazio del disco occupato da righe aggiornate o cancellate
- Evita la perdita di dati molto vecchi
- Aggiorna le statistiche usate dal sistema per gestire al meglio le query (ANALYZE)

### 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



# Fine II lezione

## Componenti e linguaggi

*GIS e Geo WEB: piattaforme e architetture*