Basi di dati e GIS

Paolo Zatelli

Dipartimento di Ingegneria Civile, Ambientale e Meccanica

Università di Trento

Outline

UNIVERSITÀ DEGLI STUDI DI TRENTO
Dipartimento di Ingegneria Civile, Ambientale e Meccanica

Basi di dati

Per base di dati o database si intende un insieme di informazioni organizzate in funzione delle necessità di acquisizione, gestione ed utilizzo.

Per estensione spesso si usa il termine “database” anche per indicare un sistema di gestione di basi di dati o l’insieme di base di dati e sistema di gestione.

Il sistema di gestione di una base di dati si indica con DataBase Management System (DBMS).

Database e GIS

In ambito GIS i database sono tipicamente utilizzati per la gestione degli attributi collegati direttamente (es. nomi di vie) o indirettamente (es. codici fiscali di proprietari di particelle in un catasto) alle primitive geometriche.

La maggior parte dei GIS associa database solo a primitive vettoriali, alcuni sono in grado di utlizzarli anche con mappe raster.

I software GIS forniscono un servizio di gestione di database interno, spesso con capacità limitate, ma sono solitamente in grado di collegarsi a DBMS esterni.

Categorie

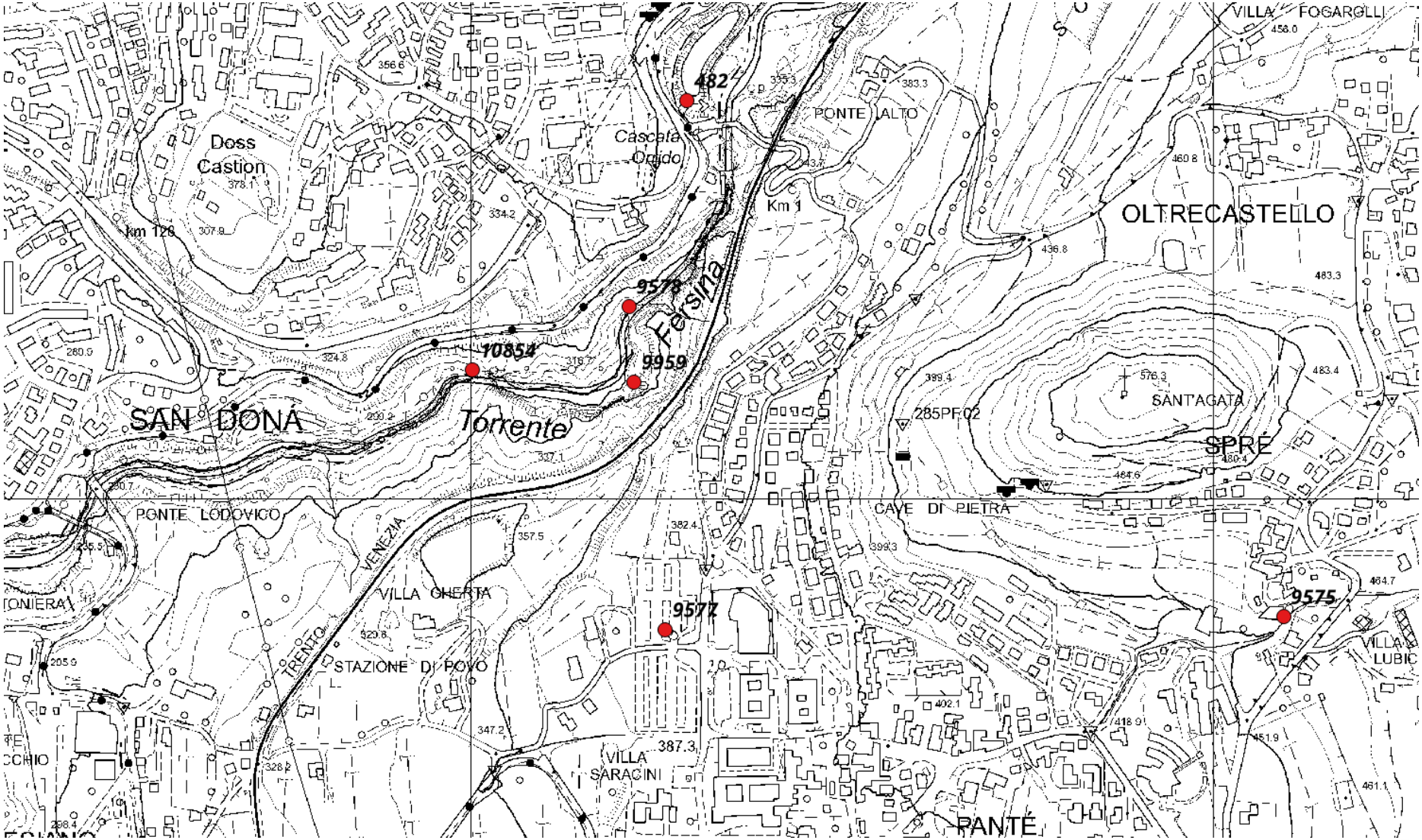
La connessione tra attributi e primitive geometriche è realizzata assegnando loro identificativi numerici interi detti categorie o feature ID.

Le categorie possono essere uniche (es. una categoria per ogni punto) oppure uguali per diverse primitive (es. punti con le stesse caratteristiche hanno stessa categoria).

Alcuni GIS sono in grado di gestire più categorie, e quindi connessioni al database, per le stesse primitive geometriche.

UNIVERSITÀ DEGLI STUDI DI TRENTO
Dipartimento di Ingegneria Civile, Ambientale e Meccanica

Categorie - sorgenti attorno a Mesiano



Mappa delle sorgenti attorno a Mesiano e categorie.

Tabelle

Le informazioni in un database sono memorizzate in tabelle, spesso collegate fra loro.

Le tabelle dovrebbero raccogliere informazioni omogenee che sono lette e scritte assieme.

E' possibile progettare un database per ottimizzare la distribuzione dei dati in tabelle.

Parti di una tabella

Una tabella di database è organizzata per righe e colonne.

Le righe sono dette record o *tuple* ed in ambito GIS corrispondono ad una o più primitive geometriche.

Le colonne sono dette campi o *fields* e sono tipizzate (int, float, double, char, ecc.). I dati in una colonna devono essere delle stesso tipo[1] e rappresentano gli attributi delle primitive.

Un record è quindi composto da un insieme fissato di campi per ogni tabella.

[1] sono comunque *interpretati* come fossero dello stesso tipo.

Parti di una tabella - esempio

Colonna = Campo Field

Riga = Record

CODSOR	NOME_UFF	PORTATA	azimuth180	ENT_VS
9959	pegoretti	4	111	1
10854	maestranzi - tomasi	4	317	0
9578	ponte alto uscita	50.58	318	0
9577	pante' 2	0	125	1
9575	oltrecastello	2.25	34	1
482	ponte alto origine	0	210	1

Tabella della mappa delle sorgenti vicino a Mesiano.

Campi chiave

In un DBMS è solitamente possibile imporre condizioni e vincoli sui campi, ad es. il campo non può essere vuoto, il valore deve essere in un certo intervallo, ecc.

Un campo chiave è un campo che può essere utilizzato per identificare in modo univoco un record.

Un campo chiave:

- deve contenere un valore univoco e non nullo (NULL)
- può consistere in uno o più campi combinati
- consente il collegamento (join) di tabelle tra loro
- è fondamentale per il collegamento con le primitive geometriche nei GIS

Ogni tabella può avere una o più chiavi.

Campi chiave - esempio

Possibile campo chiave

Possibile campo chiave*

CODSOR	NOME_UFF	PORTATA	azimuth180	ENT_VS
9959	pegoretti	4	111	1
10854	maestranzi - tomasi	4	317	0
9578	ponte alto uscita	50.58	318	0
9577	pante' 2	0	125	1
9575	oltrecastello	2.25	34	1
482	ponte alto origine	0	210	1

Possibili campi chiave.

(* solo se non esistono sorgenti con lo stesso nome)

Campi chiave - collegamento tra tabelle

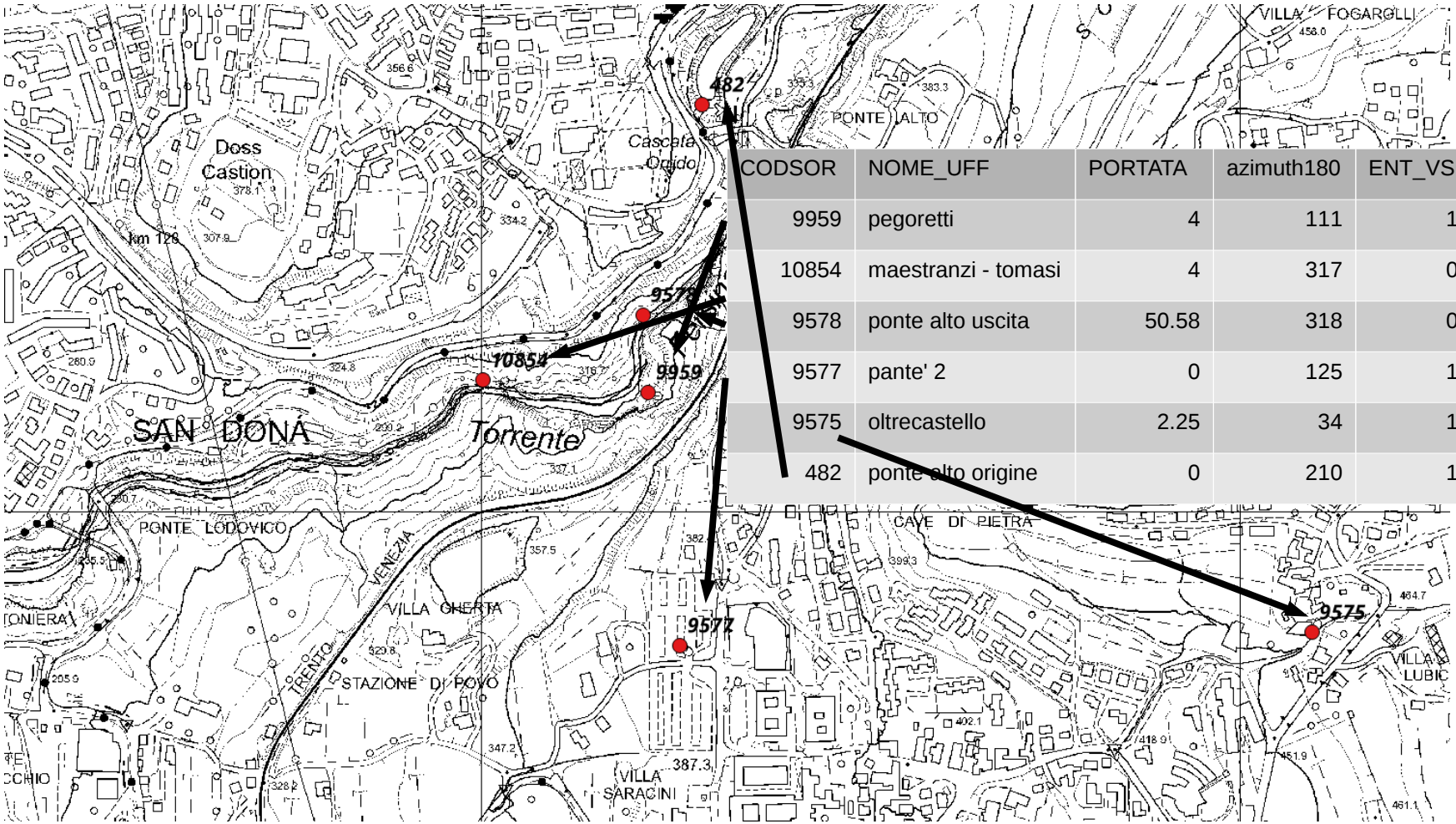
CODSOR	NOME_UFF	PORTATA	azimuth180	ENT_VS
9959	pegoretti	4	111	1
10854	maestranzi - tomasi	4	317	0
9578	ponte alto uscita	50.58	318	0
9577	pante' 2	0	125	1
9575	oltrecastello	2.25	34	1
482	ponte alto origine	0	210	1

CODSOR	pH	durezza	Residuo fisso 180°	Data prelievo
9959	7.8	22.1	296.96	04-02-2014
10854	7.6	21.8	298.71	30-01-2014
9578	7.9	22.4	295.86	01-02-2014
9577	7.7	22.3	296.75	03-02-2014
9575	7.8	21.7	294.97	02-02-2014
482	7.6	22.1	296.31	31-01-2014

Collegamento (join) fra tabelle con campi chiave[2].

[2] i valori della seconda tabella sono fittizzi

Campi chiave - collegamento tabella - mappa



CODSOR	NOME_UFF	PORTATA	azimuth180	ENT_VS
9959	pegoretti	4	111	1
10854	maestranzi - tomasi	4	317	0
9578	ponte alto uscita	50.58	318	0
9577	pante' 2	0	125	1
9575	oltrecastello	2.25	34	1
482	ponte alto origine	0	210	1

Collegamento fra tabella e mappa con campo chiave.

Indici su tabelle

Le operazioni su tabelle, in particolare la ricerca e la selezione, possono essere molto più veloci se i dati sono ordinati.

Intuitivamente è come effettuare manualmente una ricerca per leggere o scrivere su un elenco in ordine alfabetico anzichè casuale.

Si potrebbero ordinare i record secondo un campo, ma in generale è poco efficiente e limitato ad un solo campo per tabella.

Si aggiungono campi fittizzi, detti indici, che indicano l'ordine dei record riferiti a specifici campi. I campi si dicono *indicizzati*.

Una tabella può avere nessuno, uno o molti campi indicizzati.

Structured Query Language

L'interrogazione e la selezione su tabelle sono fatte usando lo Structured Query Language (SQL).

L'SQL è uno standard ISO dal 1987, ma la maggior parte dei gestori di database implementano anche caratteristiche non standard.

Lo stesso linguaggio può essere usato per:

- inserire dati
- modificare dati
- creare e cancellare tabelle
- creare e cancellare record
- creare e cancellare campi

SQL - selezione

Una operazione su un database è in generale indicata con query, termine che indica più specificamente una interrogazione.

In SQL una query ha la forma: Select-From-Where

- SELECT indica i campi che si vogliono estrarre (* per tutti)
- FROM la tabella/e su cui si effettua l'interrogazione
- WHERE le condizioni usate per la selezioni

Esempio:

```
SELECT NOME_UFF,PORTATA FROM sorgenti WHERE PORTATA>3
```

Molti programmi di gestione di DB e GIS forniscono interfaccia per la creazione assistita di query.

SQL - esempio di query

```
SELECT NOME_UFF,PORTATA FROM sorgenti WHERE PORTATA>3
```

CODSOR	NOME_UFF	PORTATA	azimuth180	ENT_VS
9959	pegoretti	4	111	1
10854	maestranzi - tomasi	4	317	0
9578	ponte alto uscita	50.58	318	0
9577	pante' 2	0	125	1
9575	oltrecastello	2.25	34	1
482	ponte alto origine	0	210	1

Risultato della query sulla tabella sorgenti.

SQL - creazione di un campo

```
ALTER TABLE sorgenti ADD COLUMN port_mc double
UPDATE sorgenti SET port_mc=PORTATA/1000
```

CODSOR	NOME_UFF	PORTATA	azimuth180	ENT_VS	port_mc
9959	pegoretti	4	111	1	0.004
10854	maestranzi - tomasi	4	317	0	0.004
9578	ponte alto uscita	50.58	318	0	0.05058
9577	pante' 2	0	125	1	0
9575	oltrecastello	2.25	34	1	0.00225
482	ponte alto origine	0	210	1	0

Aggiunta e riempimento della colonna port_mc.

Data Base Management System

Un DBMS (Data Base Management System) è un sistema software per la gestione di database.

Controlla l'acquisizione, l'organizzazione, l'interrogazione, la sicurezza e l'integrità di un database.

Risponde a richieste di *client* (es. GIS) ed esegue le relative operazioni, in particolare il trasferimento di dati.

In particolare permettono di:

creare DB strutture di tabelle, vincoli d'integrità, ...

gestire dati inserire dati, effettuare query, definire trigger, ...

gestire accessi concorrenti, back up, roll back, ...

Requisiti di un DBMS

Un DBMS moderno deve essere in grado di garantire:

ridondanza dei dati ogni informazione dovrebbe essere presente una volta sola, per evitare incoerenze, soprattutto in caso di aggiornamento

uniformità dei dati informazioni omogenee devono essere inserite in modo coerente (es. “via” o “Via” o “V.”)

indipendenza dalla piattaforma il database deve essere indipendente dal sistema utilizzato per accederci

sicurezza delle transazioni le operazioni sul database non devono mai lasciare il database in uno stato non coerente

multiutenza più utenti (client) devono poter accedere al database contemporaneamente

Protocollo ACID

I DBMS assicurano la possibilità di multiutenza attraverso il protocollo ACID: Atomicity, Consistency, Isolation e Durability.

Atomicità ogni operazione (transazione) sul database è indivisibile nella sua esecuzione, la modifica è completamente effettuata o non lo è affatto

Consistenza all'inizio ed al termine di una transazione il database è in uno stato consistente, non sono mai violati eventuali vincoli di integrità

Isolamento ogni transazione deve essere eseguita in modo isolato e indipendente dalle altre, l'eventuale fallimento di una transazione non deve interferire con le altre transazioni

Durabilità o persistenza, le modifiche al database causate da una transazione non devono in nessun caso essere perse; per assicurare questa condizione si usano *log* che registrano le operazioni fatte

DBMS

Esistono diversi approcci alla gestione di un database:

file based ogni tabella è memorizzata in un file separato (CSV, DBF, ecc.)

embedded le tabelle e le informazioni accessorie sono salvate in un unico file a cui una applicazione accede tramite una libreria dedicata (SQLite)

cliend-server il database è archiviato in modo trasparente all'utente, che vi accede attraverso un server

File based

In questo tipo d database ogni tabella è memorizzata in un file separato.

Eventuali informazioni ausiliarie (vincoli, indici, ecc.) sono memorizzate in file separati, di solito specifici per il programma in uso.

I vantaggi di questo approccio sono dati da semplicità (per qualunque operazione su una tabella si lavora direttamente su un solo file) e standardizzazione (data la struttura semplice, quasi tutti i programmi sono in grado di leggere e scrivere questi formati); si usano perciò come formato di interscambio.

Gli svantaggi sono legati al limitato numero di tipi di dati, alla scarsa efficienza dal punto di vista informatico ed alla limitazione di memorizzazione di singole tabelle.

I formati più usati sono CSV e DBF.

CSV

Nel formato CSV (Comma Separated Values) i dati sono salvati in formato ASCII, i record sono individuati da righe separate, i campi da un carattere *separatore di campi*, di solito la virgola.

```
CODSOR,NOME_UFF,PORTATA,azimuth180,ENT_VS,port_mc
9959,pegoretti,4,111,1,0.004
10854,maestranzi - tomasi,4,317,0,0.004
9578,ponte alto uscita,50.58,318,0,0.05058
9577,pante' 2,0,125,1,0
9575,oltrecastello,2.25,34,1,0.00225
482,ponte alto origine,0,210,1,0
```

Tabella delle sorgenti in formato CSV.

DBF

Il formato DBF (dBASE database file) è un formato binario introdotto nel 1983 che memorizza una singola tabella.

Può contenere i tipi di campi: C (Character), D (Date), F (Floating point), L (Logical), M (Memo) e N (Numeric). Di fatto si usano Character e Numeric.

Ha alcuni importanti limiti:

- massimo di 10 caratteri per i nomi dei campi[3]
- la maggior parte dei programmi tratta i nomi dei campi come *case insensitive*
- massimo di 256 campi
- massima dimensione di un campo Character o Numeric di 20

[3] un numero maggiore di caratteri può portare a campi con lo stesso nome

DBF e GIS

In ambito GIS sono usati nel formato di interscambio per mappe vettoriali ESRI Shape per memorizzare gli attributi.

Alcuni programmi GIS ignorano i limiti del formato DBF e spesso si generano incompatibilità.

È possibile modificare il DBF di un shape separatamente con programmi non GIS (es. fogli elettronici) senza però cambiare l'ordine dei record, che corrisponde all'ordine delle primitive geometriche sulla mappa.

SQLite

SQLite è un DBMS realizzato da una libreria usata dalle applicazioni per gestire una base di dati.

Il database è salvato in un unico file che contiene sia le tabelle che tutte le informazioni accessorie.

È molto più efficiente e flessibile del DBF e molto più semplice dei DBMS con architettura client-server.

Si preferisce quando non si usano dati e query complesse e non c'è uso concorrente.

DBMS client-server

Sono costituiti da un server che effettua le operazioni richieste da client.

I vantaggi di questo approccio sono:

- scalabilità
- uso concorrente
- prestazioni elevate

Gli svantaggi sono:

- uso di risorse computazionali
- uso di client specifici
- necessità di configurazione e gestione

Esempi di DBMS client-server

I DBMS più diffusi sono:

- IBM DB2
- Microsoft SQLServer
- MySQL
- Oracle
- PostgreSQL
- SmallWorld
- SyBase ASE

ODBC

I diversi DBMS utilizzano protocolli specifici per la connessione con i client, che devono avere i relativi driver per connettersi.

L'Open DataBase Connection (ODBC) fornisce uno standard per la connessione a DBMS anche senza driver specifici.

Rispetto ai protocolli specifici l'ODBC ha limitazioni sia relative alle prestazioni sia rispetto alle operazioni effettuabili.

Database relazionali

I Database Relazionali (RDB) sono basati sul paradigma oggetto-relazione.

Le tabelle sono dette *relazioni* perchè legano gli oggetti rappresentati dalle righe (record) ai loro attributi rappresentati dalle colonne (campi).

Usano l'SQL come linguaggio per l'interrogazione e la gestione del database.

Sono usati per gli attributi in ambito GIS per la loro efficienza e soprattutto per la flessibilità nella strutturazione dei dati e nella loro interrogazione.

NoSQL

Nella gestione di enormi moli di dati con struttura semplice l'uso di database relazionale può dare prestazioni non sufficienti.

Sono stati sviluppati modelli di database e relativi sistemi di gestione che rinunciano alla flessibilità dei DB relazionali per guadagnare efficienza, soprattutto quando si usano piccole transazioni in scrittura e grandi in lettura.

La struttura dei dati è fissa e conosciuta a priori.

I database NoSQL gestiscono i dati non in tabelle, ma usando grafi o array associativi con configurazione molto ridondante.

Esempi di DBMS NoSQL: BigTable di Google, Cassandra di Apache, Dynamo di Amazon, MongoDb, Project Voldemort di LinkedIn.

DBMS ed estensioni spaziali

Le estensioni spaziali di database e DBMS nascono dall'idea di gestire in un database oltre agli attributi la geometria e la topologia di un insieme di dati spaziali.

Il livello base è il cosiddetto *geocoding* in cui 2 o 3 colonne di una tabella sono interpretate come coordinate. Si possono rappresentare solo punti e sistema di riferimento, proiezione ed unità di misura devono essere specificati a parte.

Le estensioni spaziali dei database aggiungono i tipi di dati per rappresentare le primitive geometriche e topologiche, i sistemi di riferimento e proiezione e le funzioni per elaborarli.

Tipi di dati spaziali

La base per i tipi di dati spaziali in un database ѐata allo standard ISO 19125 *Simple feature access*.

Le primitive vettoriali sono rappresentate nel formato *Well Known Text* (WKT) o *Well Known Binary* (WKB), il primo leggibile da un operatore, il secondo illeggibile ma più efficiente dal punto di vista informatico.

Formato	Codifica
WKT	POINT(0 0)
WKB	0101000020E61

Tabella: Codifica WKT e WKB di un punto.

Le coordinate possono essere 2D, 3D o 4D (la quarta coordinata è su un riferimento curvilineo).

Tipi di dati spaziali

I tipi di dati che compongono la cosiddetta *geometry* sono:

- POINT(0 0)
- LINESTRING(0 0,1 1,1 2)
- POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))
- MULTIPOINT(0 0,1 2)
- MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))
- MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)), ((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))
- GEOMETRYCOLLECTION(POINT(2 3),LINESTRING((2 3,3 4)))
- TIN (Triangulated Irregular Network)
- POLYHEDRALSURFACE

Esistono anche le primitive specificate dallo standard ISO 13249-3:2006 per descrivere curve, ma sono poco usate bei GIS.

Well Known Text

wkt_geom	CODSOR	NOME_UFF	PORTATA	azimuth180	ENT_VS
Point (666220.08999999996740371 5104156.21000000182539225)	9959	pegoretti	4	111	1
Point (666002.09000000054948032 5104172.21000000182539225)	10854	maestranzi - tomasi	4	317	0
Point (666213.0899999993853271 5104258.2099999999627471)	9578	ponte alto uscita	50.58	318	0
Point (666261.78999999957159162 5103822.77999999932944775)	9577	pante' 2	0	125	1
Point (667095.000000000023283064 5103840.13999999966472387)	9575	oltrecastello	2.25	34	1
Point (666291.09000000054948032 5104535.20000000018626451)	482	ponte alto origine	0	210	1

Tabella delle sorgenti con geometria in WKT.

Datum

Le informazioni riguardo a sistemi di riferimento, proiezioni e loro trasformazioni secondo lo standard OGC “Geographic information - Well known text representation of coordinate reference systems”.

Ad es.

```
PARAM_MT["Mercator_2SP",
    PARAMETER["semi_major",6370997.0],
    PARAMETER["semi_minor",6370997.0],
    PARAMETER["central_meridian",180.0],
    PARAMETER["false_easting",-500000.0],
    PARAMETER["false_northing",-1000000.0],
    PARAMETER["standard parallel 1",60.0]]
```

Indicizzazione spaziale

Per avere buone performance è fondamentale l'utilizzo di indicizzazione spaziale.

A seconda dei diversi DBMS sono disponibili diversi tipi di indicizzazione spaziale (Grid, R-tree, Quadtree, GIST, Z-order, ecc.), differiscono per efficienza e complessità.

L'uso di indici può ridurre i tempi di selezione ed elaborazione di dati spaziali anche di diversi ordini di grandezza.

DBMS ed estensioni spaziali

DBMS	Estensione spaziale
IBM DB2	Spatial Extender
Microsoft SQLServer	Microsoft SQLServer
MySQL	MySQL (MySpatial)
Oracle	Oracle Spatial
PostgreSQL	PostGIS
SmallWorld	SmallWorld VMDS
Sqlite	SpatialLite
SyBase ASE	Boeing's SQS

Tabella: DBMS più diffusi e loro estensioni spaziali.

DBMS NoSQL con estensione spaziale: Neo4j e AllegroGraph.

Scelta del DBMS

La maggior parte dei GIS consentono l'uso di un DBMS integrato o uno esterno.

Il DBMS interno ha il vantaggi di non richiedere installazioni aggiuntive e configurazioni, lo svantaggio di essere implementato per un uso da parte del solo GIS, con capacità spesso limitate.

Un DBMS esterno richiede la sua installazione e configurazione, ma è solitamente molto più flessibile.

Un DBMS esterno consente:

- l'uso concorrente da parte di più utenti dello stesso database
- l'accesso al database da parte di applicazioni diverse (GIS, Mapserver, SOS, ecc.)
- la gestione e l'elaborazione efficiente di grandi moli di dati

UNIVERSITÀ DEGLI STUDI DI TRENTO
Dipartimento di Ingegneria Civile, Ambientale e Meccanica

Questa presentazione è