

Introduzione alle Basi di Dati

Claudio Rocchini

Istituto Geografico Militare

2018

Introduzione

Vediamo brevemente alcuni aspetti che riguardano i database, con particolare riferimento al loro utilizzo nei GIS.

Una introduzione (anche breve) alle Basi di Dati, richiederebbe un intero corso universitario annuale: si presenteranno i concetti minimi necessari alla comprensione delle basi di dati utilizzate nei sistemi geografici.

Caratteristiche delle Basi di Dati

Sistema (prodotto software) in grado di gestire collezioni di dati che siano:

- **Grandi:** di dimensioni molto maggiori della memoria centrale dei sistemi di calcolo utilizzati (RAM).
- **Persistenti:** con un periodo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano (anni, decenni).
- **Condivise:** utilizzate da applicazioni ed utenti diversi anche contemporaneamente (decine, centinaia di utenti, software di visualizzazione, software di editing, ...).

Caratteristiche delle Basi di Dati

Le Basi di dati devono assicurare:

- **Affidabilità:** resistenza a malfunzionamenti hardware e software, a manovre errate, colpose o dolose.
- **Privatezza:** con una disciplina e un controllo degli accessi (utenti e permessi).
- **Efficienza:** utilizzare al meglio le risorse di spazio e tempo del sistema (es. ricerca veloce all'interno di una grande mole di dati).
- **Efficacia:** rendere produttive le attività dei suoi utilizzatori.

Il Modello Relazionale

Le basi di dati moderne sono rappresentate dal cosiddetto *Modello Relazionale*. In passato esistevano altri modelli (gerarchico, reticolare).

Il Modello Relazionale è costituito da:

Entità	Sono le classi di oggetti distinguibili del nostro database, sono identificati dai loro attributi.
Relazioni	Legano fra loro le entità. Possono essere di tre tipi: uno a uno, uno a molti (o molti a uno), molti a molti.
Attributi	sono le proprietà delle entità, sono associati ad un particolare tipo di dato.

Esempio 1

Si vuole realizzare un Database per la gestione del personale. Il database deve memorizzare i dati dei dipendenti e degli uffici. Bisogna memorizzare anche in quale ufficio lavora ogni dipendente.

Entità	Dipendenti, uffici
Relazioni	Una relazione lega il dipendente e l'ufficio in cui lavora.
Attributi	Per i dipendenti: nome, cognome, codice fiscale, qualifica ... Per gli uffici: denominazione, localizzazione, ...

Esempio 2

Si vuole memorizzare una Base di Dati per memorizzare le strade ed i comuni d'Italia.

Entità	Strade, comuni.
Relazioni	Una relazione potrebbe collegare una strada ed il comune in cui si trova.
Attributi	Per le strade: nome, classifica (viale, via, ...), localizzazione geografica. Per i comune: denominazione, codice istat, localizzazione geografica, ...

Cardinalità delle relazioni

- **Uno a Uno**

- Ad ogni entità ne corrisponde una ed una sola: es. capo ufficio ed ufficio (o viceversa).

- **Uno a Molti**

- (La più diffusa) ad ogni entità ne corrispondono una o più di una: impiegati ed ufficio in cui lavorano (ogni impiegato lavora in un ufficio, mentre ad un ufficio corrispondono più impiegati).

- **Molti a Molti**

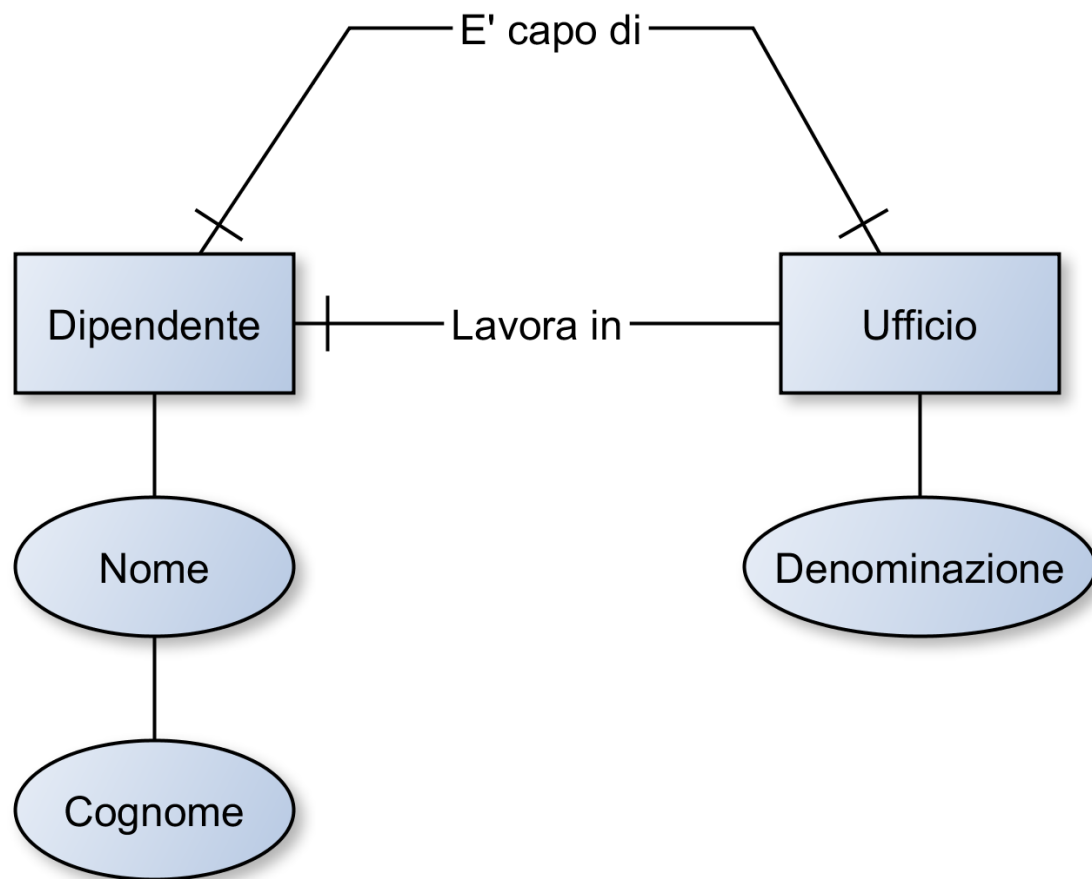
- Ad ogni entità ne corrispondono molte e viceversa: una strada può passare per più comuni, mentre ovviamente per un comune passano molte strade.

Attributi e tipi di dato

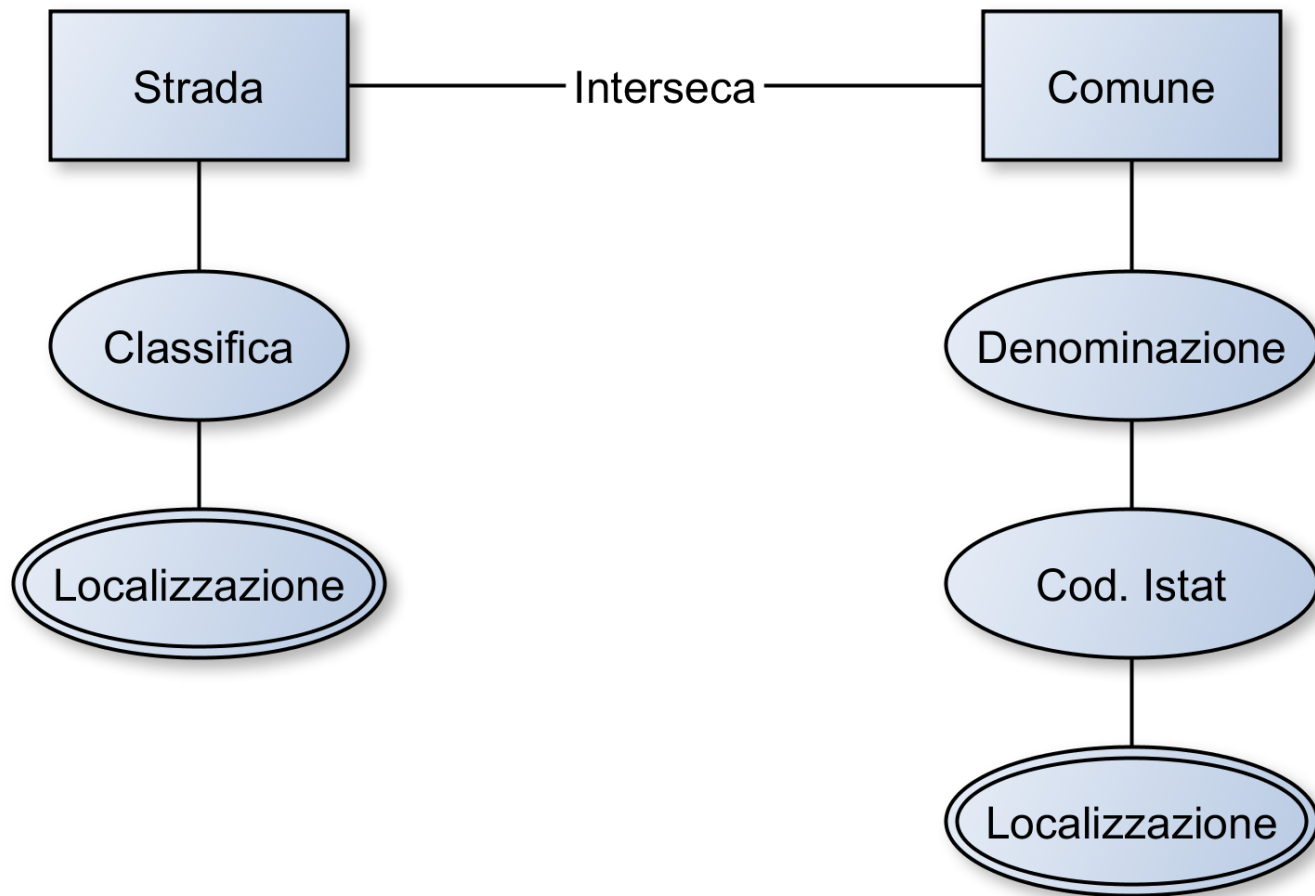
Gli attributi di una base dati sono associati ad un tipo di dato. Se un attributo è associato ad un particolare tipo di dato, non può contenere valori di altri tipi. Esempi di tipi di dato:

- **Testo** (Stringa): es. nome, cognome, ...
- **Numeri Interi**: es. età anagrafica, numero di corsie di una strada, ...
- **Numeri decimali**: es. altezza di una persona, larghezza di una strada
- *Date/ore*: es. data di nascita, ...
- Valori di verità vero/falso (**Boolean**), es. coniugato si/no, ...
- *Localizzazioni geografiche*, ...

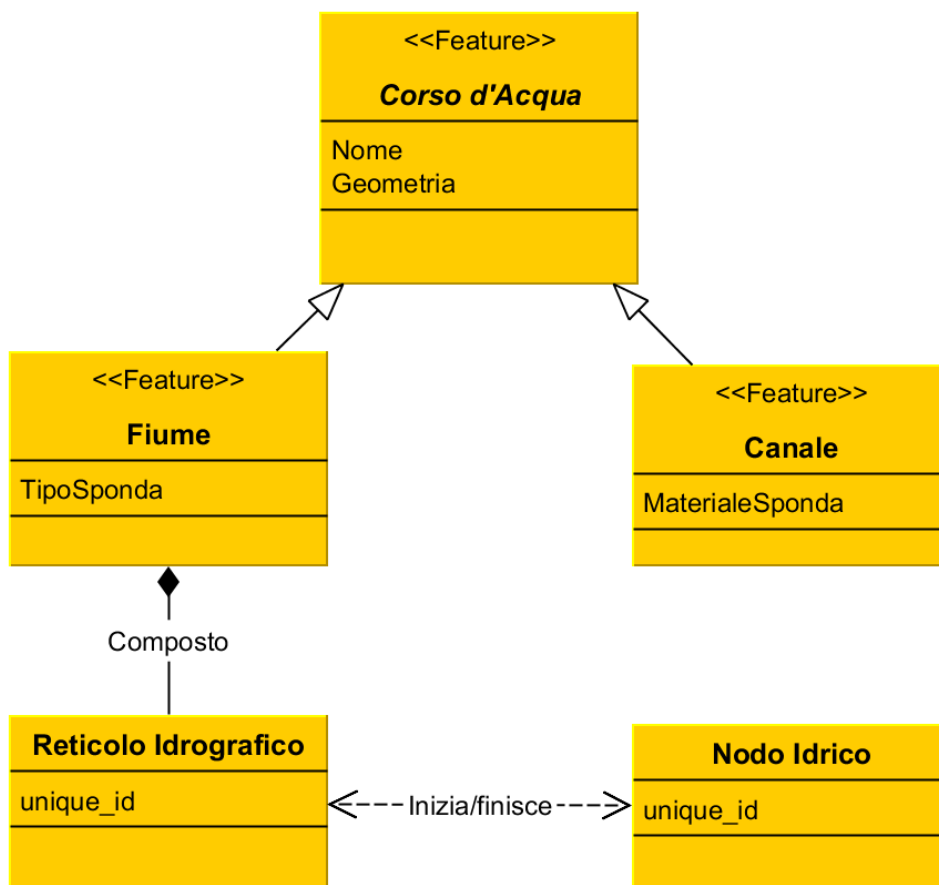
Schema ER (Entità relazioni)



Schema ER (Entità relazioni) «geografico»



Schema UML (ha sostituito ER)



Implementazione del Modello Relazionale

- E' realizzato in pratica attraverso una serie di **tabelle**.
- Ogni entità è associata ad una tabella di dati;
 - le **righe** rappresentano le **istanze** dell'entità (i singoli oggetti, es. ogni dipendente);
 - le **colonne** rappresentano gli **attributi** (nome, cognome,...).
- Le **relazioni** sono realizzate:
 - Dalla corrispondenza di valori identici in opportune tabelle;
 - Dalla definizione di vincoli di integrità.
- Le relazioni **n:n** hanno bisogno di una tabella dedicata alla relazione.
- Nota: non c'è un *ordine* prestabilito delle righe di una tabella.

Es 1: tabella

Tabelle

Colonne (attributi)

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays a tree of database objects, including 'Sequences' and 'Tables (53)'. The 'Tables' folder is expanded, showing a list of tables. The table 'lap030' is selected and highlighted with a blue dashed box. An arrow points from the label 'Tabelle' to this table. The main pane shows the 'SQL' tab with a query:

```
1 SELECT * FROM dati.lap030
2 ORDER BY ogc_fid ASC
```

 Below the query, the 'Data Output' tab displays a table with 8 columns and 15 rows. The columns are: 'ogc_fid [PK] integer', 'uid character varying (36)', 'con numeric (6)', 'fun numeric (6)', 'hct numeric (6)', 'loc numeric (6)', and 'ltn numeric (6)'. The rows contain numerical values and UUIDs. An arrow points from the label 'Colonne (attributi)' to the column headers. Another arrow points from the label 'Righe' to the first row of data. The label 'Tabelle' also has an arrow pointing to the 'lap030' table in the tree.

	ogc_fid [PK] integer	uid character varying (36)	con numeric (6)	fun numeric (6)	hct numeric (6)	loc numeric (6)	ltn numeric (6)
1	1	bb39a77d-4447-11e7-a44...	998	6	4	44	
2	2	bb39a77d-4447-11e7-bd0...	998	6	4	44	
3	3	bb39a77d-4447-11e7-b81...	998	6	4	44	
4	4	bb3c08db-4447-11e7-b15...	998	6	4	44	
5	5	bb3c08db-4447-11e7-b1c...	998	6	4	44	
6	6	bb3c08db-4447-11e7-bbd...	998	6	4	44	
7	7	bb3e6a39-4447-11e7-bec...	998	3	4	45	
8	8	bb3e6a39-4447-11e7-8bb...	998	6	4	44	
9	9	bb3e6a39-4447-11e7-a45...	998	6	1	44	
10	10	bb40cb97-4447-11e7-9a2...	998	6	4	44	
11	11	bb40cb97-4447-11e7-be0...	998	3	4	44	
12	12	bb40cb97-4447-11e7-b7d...	998	3	4	45	
13	13	bb432cf5-4447-11e7-b073...	998	6	4	44	
14	14	bb432cf5-4447-11e7-aae4...	998	6	4	44	
15	15	bb432cf5-4447-11e7-bf5...	998	6	4	44	

Righe

Es: relazione (fra strade e descrizione della localizzazione)

The image shows a database interface with two tables. A red arrow points from the value '44' in the 'ltn numeric (6)' column of the first table to the row with '44' in the 't5_3_1_code' column of the second table.

Table 1: Data Output

	ogc_fid [PK] integer	uid character varying (36)	con numeric (6)	fun numeric (6)	hct numeric (6)	loc numeric (6)	ltn numeric (6)
1	1	bb39a77d-4447-11e7-a44...	998		6	4	44
2	2	bb39a77d-4447-11e7-bd0...	998		6	4	44
3	3	bb39a77d-4447-11e7-b81...	998		6	4	44
4	4	bb3c08db-4447-11e7-b15...	998		6	4	44
5	5	bb3c08db-4447-11e7-b1c...	998		6	4	44
6	6	bb3c08db-4447-11e7-bbd...	998		6	4	44
7	7	bb3e6a39-4447-11e7-bec...	998		3	4	45
8	8	bb3e6a39-4447-11e7-8bb...	998		6	4	44
9	9	bb3e6a39-4447-11e7-a45...	998		6	1	44
10	10	bb40cb97-4447-11e7-9a2...	998		6	4	44
	11	bb40cb97-4447-11e7-be0...	998		3	4	44
		bb40cb97-4447-11e7-b7d...	998		6	4	44
		bb40cb97-4447-11e7-b7d...	998		6	4	44
		bb40cb97-4447-11e7-b7d...	998		6	4	44

Table 2: Data Output

ort_name character varying	t5_3_1_code character varying	name character varying	definition character varying	description character varying	alias_es_ character var
WaterbodyBottom	17	On Waterbody Bottom	Located on the water...	NULL	Sunken, On Sea
owWaterbodyBott...	23	Below Waterbody Bot...	Buried below the wat...	NULL	Below Sea Botto
owGroundSurface	40	Below Ground Surface	Buried below the gro...	NULL	Underground
Surface	44	On Surface	Located on the groun...	NULL	NULL
oveSurface	45	Above Surface	Located above the gr...	NULL	NULL
oveWaterbodyBott...	46	Above Waterbody Bo...	Located above the w...	NULL	NULL
owWaterbodySurf...	47	Below Waterbody Sur...	Located at all times b...	NULL	NULL

Basi di dati e documenti

Applicazione orientata al «documento»

- Es. lettere di Word, immagini Photoshop
- Creazione di un nuovo elemento.
- Caricamento documento da disco in memoria computer.
- Operazioni di modifica in memoria.
- Salvataggio (oppure no) da memoria a disco.
- Manipolazione diretta del file su disco (cancellazione, copia, spedizione, ...).

Applicazione orientata alla «base di dati»

- Es. Gestione amministrativa, dati geografici
- Lettura dei dati tramite connessione senza caricamento.
- Registrazione istantanea delle modifiche, non si possono caricare/salvare i dati.
- Il file non è direttamente accessibile (no copia, cancellazione, diretta).
- Utilizzo condiviso fra più utenti contemporanei.

Concetti: Chiavi e Chiave primaria

- **Chiave**

- Una chiave è un attributo (od un insieme di attributi), il cui valore è utilizzato nelle relazioni fra tabelle.

- **Chiave Primaria**

- La Chiave Primaria è un attributo (od un insieme di attributi) che identifica univocamente un oggetto (una riga, si ricorda che l'ordine è casuale).

- **Chiave esterna (Foreign key)**

- E' il codice di partenza di una relazione

- **Esempi**

- Per i dipendenti: codice fiscale (oppure nome+cognome+data_di_nascita)
- Per i comuni: codice Istat
- Per i boschi: numero progressivo o codice esadecimale casuale

Indici

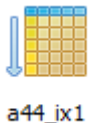
- Gli indici sono strutture aggiuntive che permettono di **velocizzare** le interrogazioni sui campi dati.
- Un indice è di solito associato ad un **attributo** (colonna di una tabella), oppure ad un insieme di colonne (es. cognome o cognome + nome).
- Se si prevede di fare molte **ricerche** su di una colonna od un insieme di colonne, è opportuno costruire un indice.
- Gli indici velocizzano enormemente le query (ma occupano **spazio disco** e rallentano leggermente la modifica dei dati).
- Gli indici standard si riferiscono a parole, numeri e date.
- Esistono poi gli **indici spaziali** (sulle geometrie).

Indici (Esempio)

```
SELECT *  
FROM dbsn.edifc  
WHERE  
    shape &&  
    'SRID=6875;POINT(6723303.2824 4939282.4087)'::GEOMETRY
```

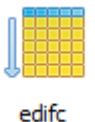
- Con indice

- "Index Scan using a44_ix1 on edific (cost=0.42..8.44 rows=1 width=425)"
- 32 millisecondi = meno di un centesimo di secondo



- Senza indice

- "Seq Scan on edific (cost=0.00..1858574.00 rows=1 width=425)"
- 153773 millisecondi = più di due minuti e mezzo

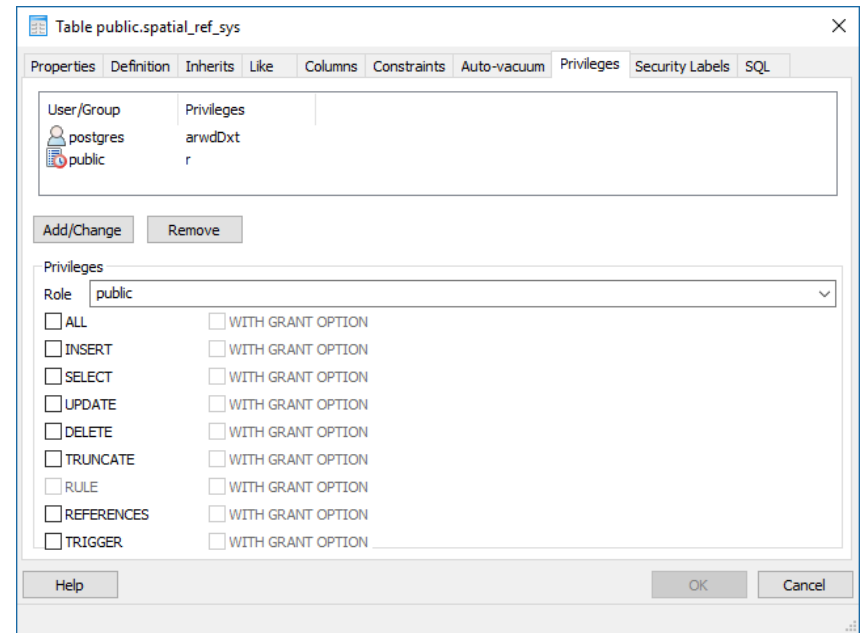


Schemi

- Scopo
 - I database reali possono essere molto **complessi** e contenere migliaia di oggetti (tabelle).
- Definizione
 - Gli schemi rappresentano la possibilità di **suddividere** il database in sottoparti (per fare ordine). Ogni oggetto del database apparterrà ad uno schema.
- Similitudini
 - Gli schemi di database hanno la stessa funzione delle **cartelle** per i dischi (ma uno schema non può contenere un altro schema!).
- Esempi
 - In Oracle ad ogni utente è associato un omonimo schema.
 - Il database geodetico IGM è diviso negli schemi: punti trigonometrici, livellazione, IGM95.

Utenti e permessi

- Come per Windows e i suoi files, i database hanno **utenti**, **gruppi** e **permessi**.
- Gli utenti sono muniti di **password** per accedere al database.
- Gli elementi del database possono avere dei permessi diversi per utenti o gruppi di utenti diversi (es. *accesso, modifica, lettura*).
- Nell'esempio postgres (l'amministratore) ha tutti i privilegi, public (qualsiasi altro utente) può solo leggere i dati di questa tabella.

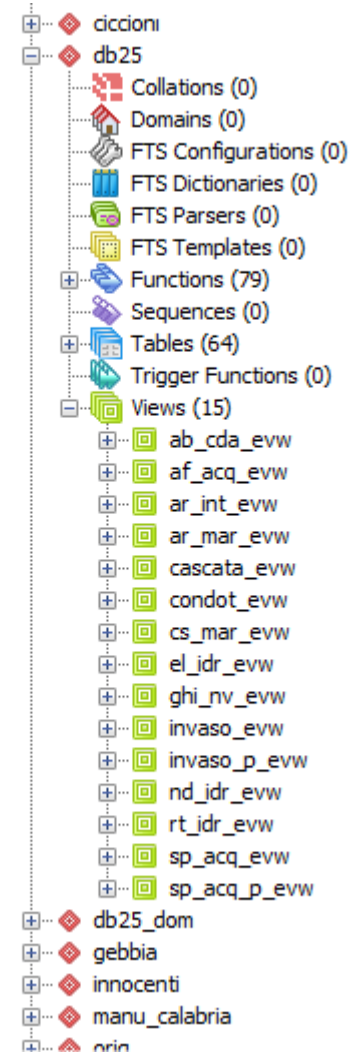


Query (interrogazione)

- Le basi di dati contengono un'enorme quantità di dati.
- Scopo delle Query:
 - **Estrarre** un sottoinsieme di dati che rispetta particolari condizioni.
 - Utilizzare le **relazioni** fra entità.
 - **Correlare** i dati in modo da estrarre nuove informazioni.
- Esempi:
 - I nomi e cognomi (e non gli altri attributi) di tutti i dipendenti.
 - I dipendenti nati nel 1967.
 - I dipendenti che si chiamano «Rocchini» di cognome.
 - I dipendenti del Servizio Informatico.
 - Le strade principali.
 - Le strade secondarie vicine ad un uliveto.
 - I laghi con superficie minore di 1000 mq della Lombardia.

Viste

- Alcune query sono molto importanti, tanto che è possibile «battezzarle» e salvarle all'intero del database.
- Una query battezzata e salvata prende il nome di «*vista*» (**view** in inglese), perché è un modo diverso di *vedere* i dati.
- Le viste possono essere utilizzate come tabelle, per **leggere** o addirittura **modificare** i dati.
- Il contenuto delle viste non è fissato durante la loro creazione, ma varia al variare dei dati sottostanti; il contenuto è quindi dinamico.



Transazioni

- Le transazioni sono formate da un blocco indivisibile di operazioni di modifica; se una parte delle modifiche non va a buon fine, nessuna modifica viene effettuata.
- Si pensi all'operazione di prenotare due posti vicini sul *Freccia Rossa*; voglio essere sicuro di prenotare entrambi i posti, prima che qualcun altro ne prenoti uno al mio posto.
- Un tentativo di transazione può terminare con:
 - Successo, le modifiche vengono effettuate (**COMMIT**)
 - Fallimento, nessuna modifica effettuata (**ROLLBACK**)

Forme normali

- Esiste una teoria sulla progettazione delle basi di dati.
- **Prima forma:** tutte le righe hanno lo stesso numero di colonne e ogni colonna lo stesso tipo di dato, gli attributi sono valori semplici, esiste una chiave primaria (l'ordine delle righe è irrilevante).
- **Seconda forma:** tutti gli attributi non chiave dipendono dall'intera chiave primaria.
- **Terza forma:** tutti gli attributi non chiave dipendono solo dalla chiave primaria (e non da altri attributi).

SQL

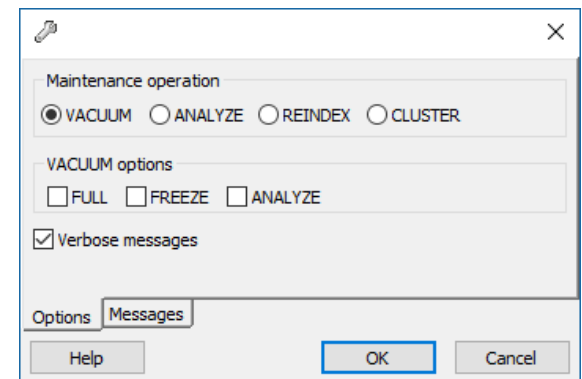
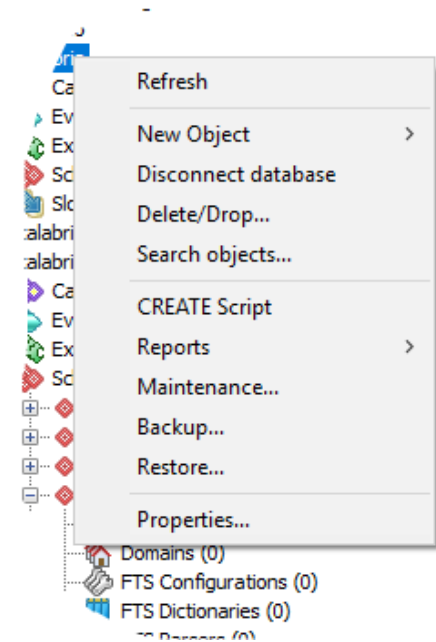
- Ci sono vari software per database (professionali: Oracle, Postgres, MS SQL Server, MySQL; casalinghi: Access, DB3/Clipper, SQLite).
- Esiste però un linguaggio standard per la manipolazione di database: Standard Query Language.
- Esempi:

```
SELECT nome, cognome  
FROM dipendenti  
WHERE anno_assunzione < 2005;
```

```
SELECT MIN(quota)  
FROM punti_quota  
WHERE nome_regione='sardegna' ;
```

Operazioni di amministrazione

- I database professionali hanno bisogno di operazioni routinarie di gestione:
 - **Vacuum**: per motivi di efficienza i dati non vengono mai cancellati veramente, ma «nascosti», questa operazione *recupera spazio* dai dati cancellati. Molti database possono automatizzare questa operazione (Autovacuum)
 - **Analyze**: per utilizzare efficientemente gli indici è necessario creare degli istogrammi di *distribuzione statistica* dei dati. E' necessario soprattutto dopo modifica sostanziali della base di dati (inserimento o cancellazione di un grande numero di dati)
 - **Backup/Restore**: i file del database non sono direttamente accessibili, per creare copie di sicurezza è necessario effettuare queste operazioni.



Appendice: software per geodatabase

- Su file (Documento)
 - Microsoft Access (ArcGIS Personal Geodatabase, Geomedia)
 - SQLite (Spatialite)
 - File e cartelle (Insieme di shapefile, ArcGIS FileGeodatabase)
 - dBase IV (es. i file .DBF a corredo degli shapefile)
- Su server, scatola nera (in pratica è un «servizio» del server)
 - PostgreSQL
 - Oracle
 - Microsoft SQLServer
 - MySQL (di Oracle)
- Su cluster di server (NoSQL, BigData)
 - MongoDB, ElasticSearch
 - Hadoop + Spark + Spark SQL, ...