

***DBMS (Data Base Management System)
nei progetti astronomici:
Motivazioni e modalità di utilizzo***



DBMS (Data Base Management System)

Motivazioni:

I DBMS rendono più semplice:

- Archiviazione;
- Accesso;
- Condivisione;
- Protezione dei dati;

... ma per usare i DBMS è necessario :

- Imparare ad usare un nuovo software;
- Sforzarsi di seguire la logica dei database;

DBMS (Data Base Management System)

I dati:

1	BN	Tau	8.2	05	34.5	+22	01.0	6.3 kly	6'x4'
2	GC	Aqu	6.3	21	33.5	00	-49.0	36.2 kly	12.9'
3	GC	CVn	6.3	13	42.2	+28	23.0	30.6 kly	16.2'
4	GC	Sco	6.4	16	23.5	-26	31.5	6.8 kly	26.3'
5	GC	Ser	6.2	15	18.6	+02	05.0	22.8 kly	17.4'
6	OC	Sco	4.2	17	40.4	-32	13.8	2 kly	33'
7	OC	Sco	4.1	17	53.9	-34	47.0	800 ly	80.0'
8	BN	Sag	6.0	18	04.1	-24	18.0	5200 ly	90'x40'
9	GC	Oph	7.3	17	19.2	-18	31.0	26.4 kly	9.3'

... aggiungiamo i "meta-data" !

DBMS (Data Base Management System)

Dati + metadata = file FITS, VOTable, etc...

Contenuto: catalogo oggetti Messier

Data: 15/12/2006

Strumento: ...

Id	Type	Const	Mag	Ra_h	Ra_m	Dec	Dec_p	Dist	App_size
(int)	(string)	(string)	(float)	(float)	(float)	(float)	(float)	(float)	(string)
1	BN	Tau	8.2	05	34.5	+22	01.0	6.3 kly	6'x4'
2	GC	Aqu	6.3	21	33.5	00	-49.0	36.2 kly	12.9'
3	GC	CVn	6.3	13	42.2	+28	23.0	30.6 kly	16.2'
4	GC	Sco	6.4	16	23.5	-26	31.5	6.8 kly	26.3'
5	GC	Ser	6.2	15	18.6	+02	05.0	22.8 kly	17.4'
6	OC	Sco	4.2	17	40.4	-32	13.8	2 kly	33'
7	OC	Sco	4.1	17	53.9	-34	47.0	800 ly	80.0'
8	BN	Sag	6.0	18	04.1	-24	18.0	5200 ly	90'x40'
9	GC	Oph	7.3	17	19.2	-18	31.0	26.4 kly	9.3'

... ora aggiungiamo un software che nasconda i dettagli di memorizzazione dei dati

DBMS (Data Base Management System)

Dati + metadata + software

Contenuto: catalogo oggetti Messier

Data: 15/12/2006

Strumento: ...

Id	Type	Const	Mag	Ra_h	Ra_m	Dec	Dec_p	Dist	App_size	
(int)	(string)	(string)	(float)	(float)	(float)	(float)	(float)	(float)	(string)	(string)
1	BN	Tau	8.2	05	34.5	+22	01.0	6.3 kly	6'x4'	
2	GC	Aqu	6.3	21	33.5	00	-49.0	36.2 kly	12.9'	
3	GC	CVn	6.3	13	42.2	+28	23.0	30.6 kly	16.2'	
4	GC	Sco	6.4	16	23.5	-26	31.5	6.8 kly	26.3'	
5	GC	Ser	6.2	15	18.6	+02	05.0	22.8 kly	17.4'	
6	OC	Sco	4.2	17	40.4	-32	13.8	2 kly	33'	
7	OC	Sco	4.1	17	53.9	-34	47.0	800 ly	80.0'	
8	BN	Sag	6.0	18	04.1	-24	18.0	5200 ly	90'x40'	
9	GC	Oph	7.3	17	19.2	-18	31.0	26.4 kly	9.3'	

DBMS (Data Base Management System)

Dati + metadata + software = DATA BASE !

Contenuto: catalogo oggetti Messier

Data: 15
Strumen

Id Type
(int) (s

1	BN
2	GC
3	GC
4	GC
5	GC
6	OC
7	OC
8	BN
9	GC

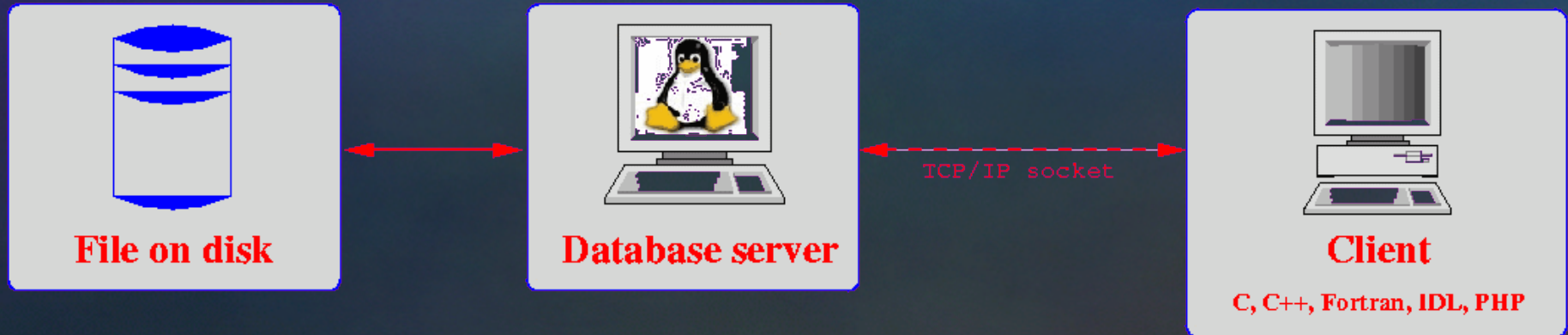
The screenshot shows the MySQL Query Browser interface. The query executed is `SELECT * FROM Messier M LIMIT 0,1000`. The result set contains 110 rows, with the first 9 rows displayed in the table below. The status bar indicates "110 rows fetched in 0:00.0671".

M	Type	Const	Mag	Ra	Decl	Dist	App_size
1	BN	Tau	8.2	5.575	22.0167	6.3 kly	6'x4'
2	GC	Aqu	6.3	21.5583	-0.816667	36.2 kly	12.9'
3	GC	CVn	6.3	13.7033	28.3833	30.6 kly	16.2'
4	GC	Sco	6.4	16.3917	-25.475	6.8 kly	26.3'
5	GC	Ser	6.2	15.31	2.08333	22.8 kly	17.4'
6	OC	Sco	4.2	17.6733	-31.77	2 kly	33'
7	OC	Sco	4.1	17.8983	-33.2167	800 ly	80.0'
8	BN	Sag	6	18.0683	-23.7	5200 ly	90'x40'
9	GC	Oph	7.3	17.32	-17.4833	26.4 kly	9.3'

ng)

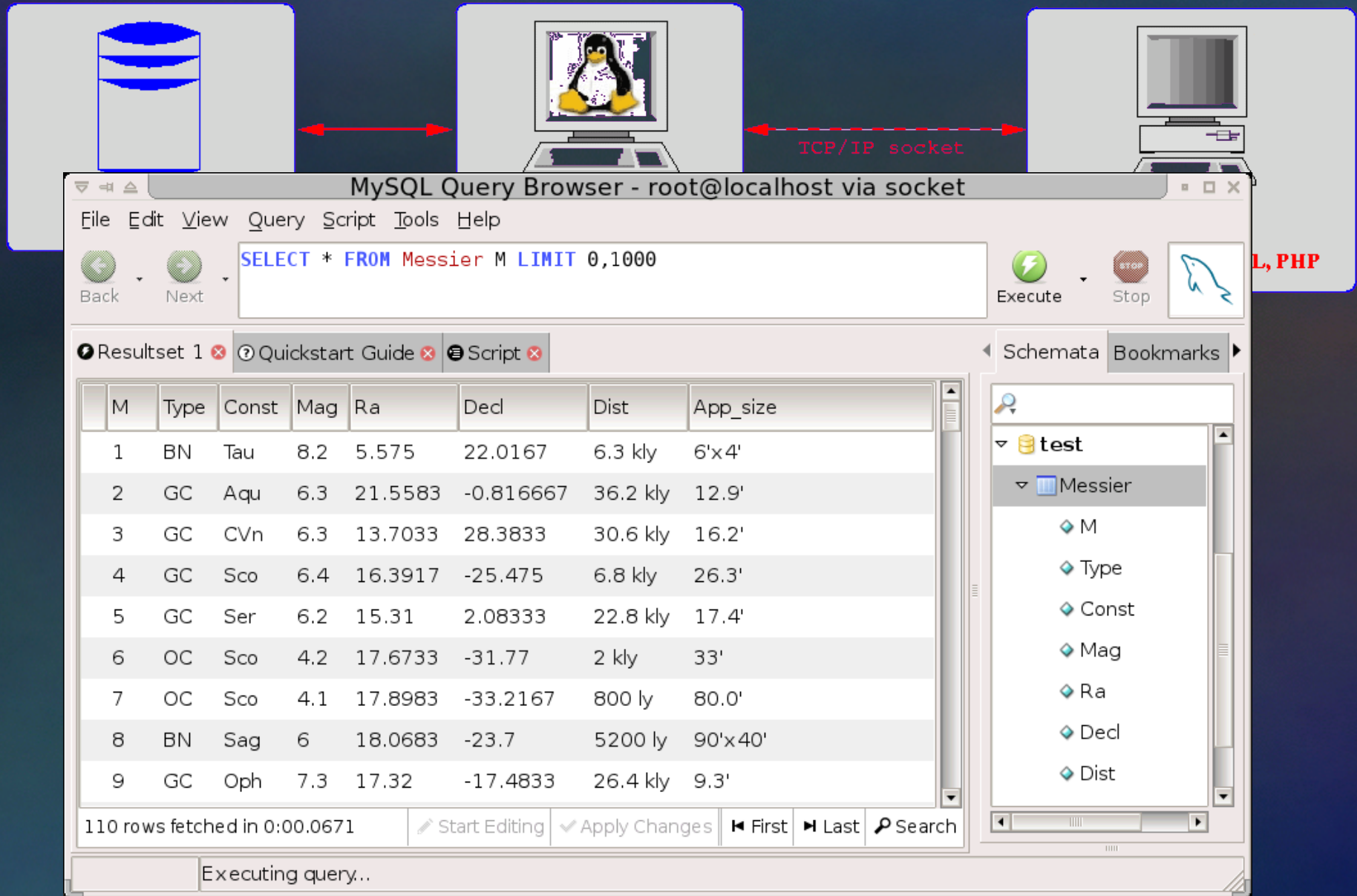
DBMS (Data Base Management System)

Architettura di un DBMS:



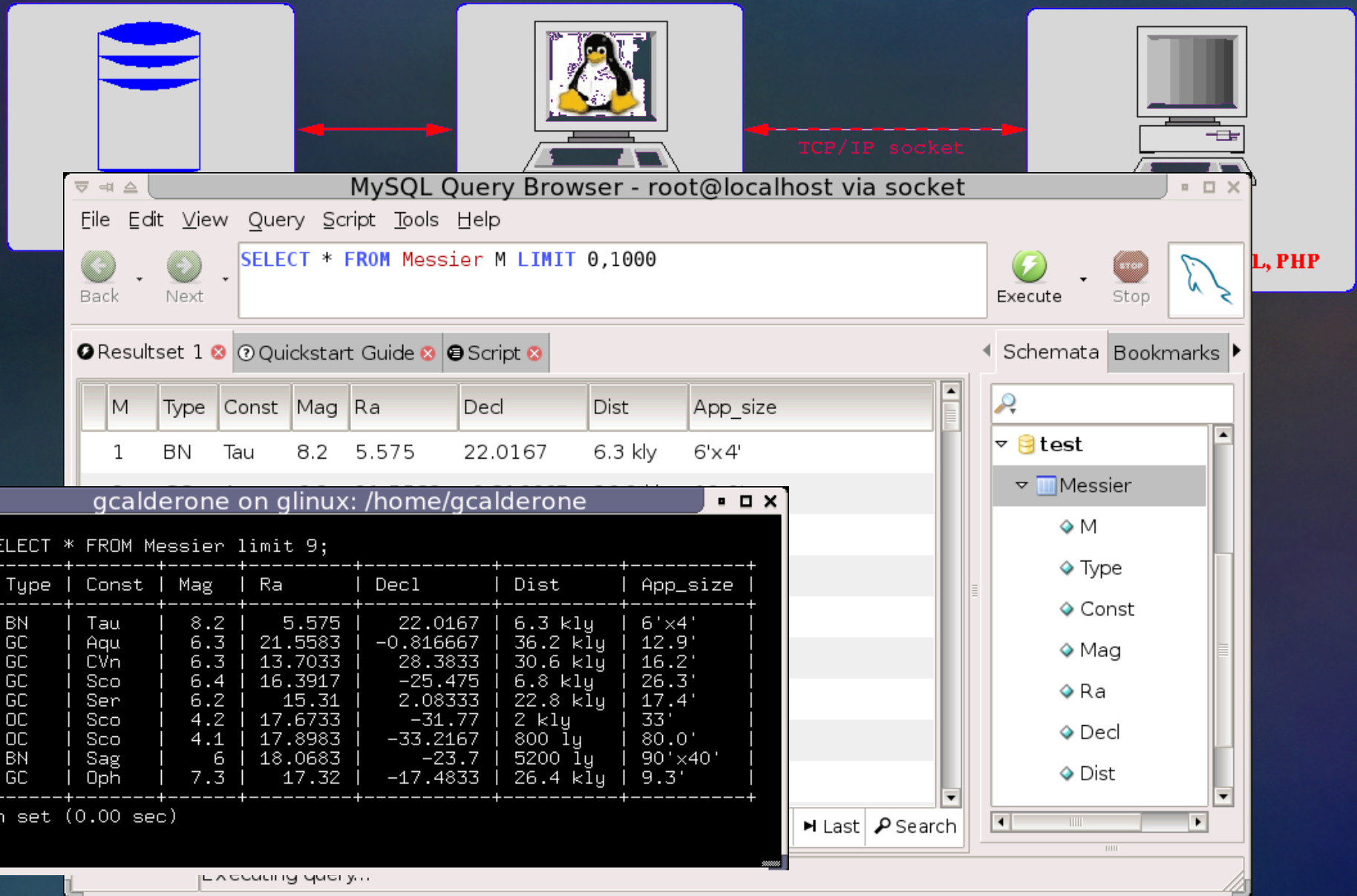
DBMS (Data Base Management System)

Architettura di un DBMS:



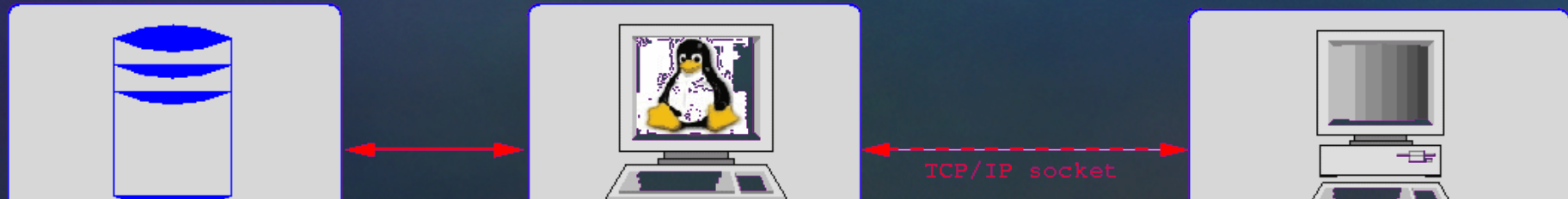
DBMS (Data Base Management System)

Architettura di un DBMS:



DBMS (Data Base Management System)

Architettura di un DBMS:



MySQL Query Browser - root@localhost via socket

gcalderone on glinux: /home/gcalderone

```
File Edit Options Buffers Tools Debug IDLWAVE Help
; The user contributed library
@mcs_usrlib ; User contributed Functions and Pro
PRO simplesql
; This is the common section for the "mcs_usrlib"
common DBinfo, c_user, c_password, c_dbname, c_query
common TABinfo, mcs_types, is_rowstru, n_fields, n_rows, f_name, $
f_atype, f_itype, tab_stru
;--
; Required info
c_user = 'testusr' & c_password = 'testpwd' & c_dbname = 'testdb'
c_query = "SELECT * FROM Messier_view"
; All in one function: open, execute query, show the result, close
status = dbexec_showqry(Debug=debug) ; (return 0 if OK)
if (status ne 0) then return ; Error?
; The result is stored in a structure...
print, 'Info on the returned structure:'
help, /stru, tab_stru
END
```

Resultset 1

M	Type	Const	Mag	Ra
1	BN	Tau	8.2	5.575

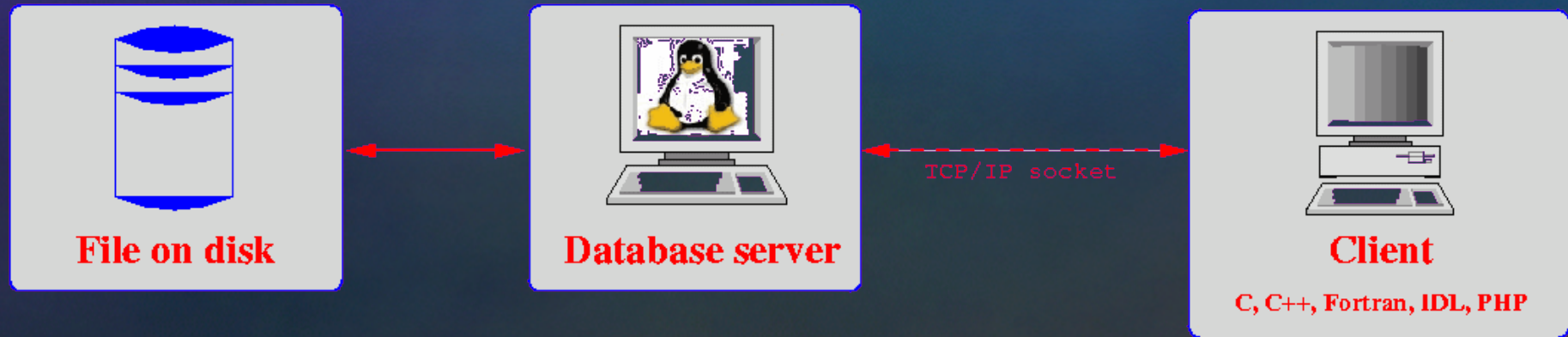
```
mysql> SELECT * FROM Messier limit 9;
+----+-----+-----+-----+-----+-----+
| M  | Type | Const | Mag  | Ra    | Decl  |
+----+-----+-----+-----+-----+-----+
| 1  | BN   | Tau   | 8.2  | 5.575 | 22    |
| 2  | GC   | Aqu   | 6.3  | 21.5583 | -0.8  |
| 3  | GC   | CVn   | 6.3  | 13.7033 | 28    |
| 4  | GC   | Sco   | 6.4  | 16.3917 | -2    |
| 5  | GC   | Ser   | 6.2  | 15.31   | 2.    |
| 6  | OC   | Sco   | 4.2  | 17.6733 | -    |
| 7  | OC   | Sco   | 4.1  | 17.8983 | -33   |
| 8  | BN   | Sag   | 6    | 18.0683 | -    |
| 9  | GC   | Oph   | 7.3  | 17.32   | -17   |
+----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
```

Executing query...

--u-:---F1 simplesql.pro (IDLWAVE Mouse Abbrev Fill)--L1--All-----

DBMS (Data Base Management System)



MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Back Next `SELECT * FROM Messier M LIMIT 0,1000` Execute Stop

Resultset 1 x Quickstart Guide x Script x

M	Type	Const	Mag	Ra	Decl	Dist	App_size
1	BN	Tau	8.2	5.575	22.0167	6.3 kly	6'x4'
2	GC	Aqu	6.3	21.5583	-0.816667	36.2 kly	12.9'
3	GC	CVn	6.3	13.7033	28.3833	30.6 kly	16.2'
4	GC	Sco	6.4	16.3917	-25.475	6.8 kly	26.3'
5	GC	Ser	6.2	15.31	2.08333	22.8 kly	17.4'
6	OC	Sco	4.2	17.6733	-31.77	2 kly	33'
7	OC	Sco	4.1	17.8983	-33.2167	800 ly	80.0'
8	BN	Sag	6	18.0683	-23.7	5200 ly	90'x40'
9	GC	Oph	7.3	17.32	-17.4833	26.4 kly	9.3'

110 rows fetched in 0:00.0671 Start Editing Apply Changes First Last Search

Executing query...

Schemata Bookmarks

- test
 - Messier
 - M
 - Type
 - Const
 - Mag
 - Ra
 - Decl
 - Dist

Motivazioni:

- Archiviazione;
- Accesso;
- Condivisione;
- Protezione dei dati;

DBMS (Data Base Management System)

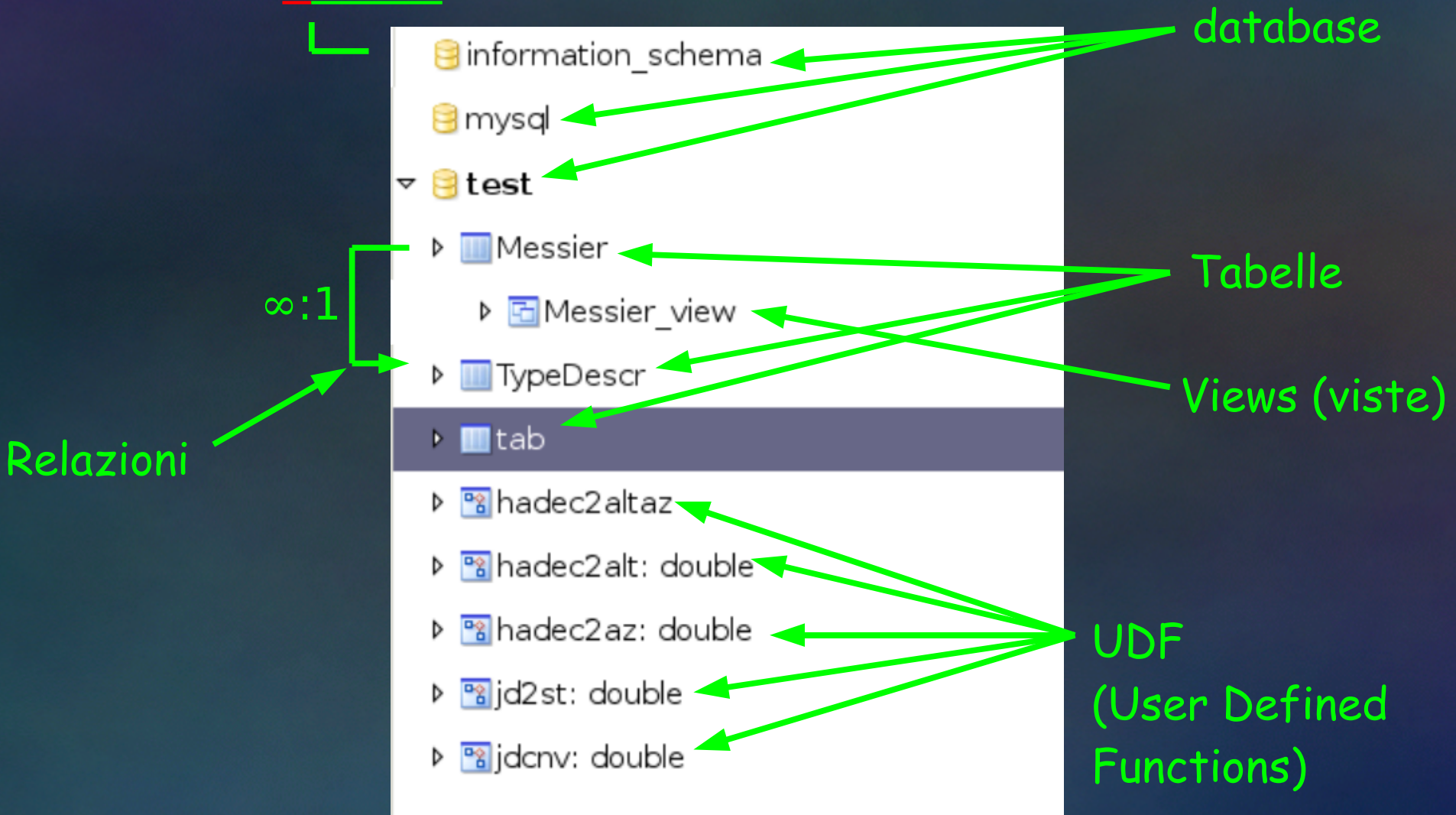
Database models:

- **Flat:** tabella bidimensionale senza constraints relativi al tipo di dato (Es. Fogli di lavoro..)
- **Gerarchico:** struttura ad albero (Es. File system, XML, struttura di un database relazionale, ecc...)
- **Network:** struttura a grafo (Es. Problemi di percorso minimo, ecc...)
- **Relazionale:** Tabelle e relazioni (Es. MySQL, Postgres, Oracle, SQLServer, ecc...)

DBMS (Data Base Management System)

Struttura di un database relazionale:

RDBMS



DBMS (Data Base Management System)

Struttura di una tabella:

M	Type	Const	Mag	Ra	Decl	Dist	App_size
1	BN	Tau	8.2	5.575	22.0167	6.3 kly	6'x4'
2	GC	Aqu	6.3	21.5583	-0.816667	36.2 kly	12.9'
3	GC	CVn	6.3	13.7033	28.3833	30.6 kly	16.2'
4	GC	Sco	6.4	16.3917	-25.475	6.8 kly	26.3'
5	GC	Ser	6.2	15.31	2.08333	22.8 kly	17.4'
6	OC	Sco	4.2	17.6733	-31.77	2 kly	33'
7	OC	Sco	4.1	17.8983	-33.2167	NULL	80.0'
8	BN	Sag	6	18.0683	-23.7	5200 ly	90'x40'
9	GC	Oph	7.3	17.32	-17.4833	26.4 kly	9.3'

Field
(campo,
attribute)

Record
(row, riga,
entry,
observation,
tuple)

Valore NULL
(diverso da 0 e stringa vuota)

DBMS (Data Base Management System)

Struttura di una tabella:

M	Type	Const	Mag	Ra	Decl	Dist	App_size
1	BN	Tau	8.2	5.575	22.0167	6.3 kly	6'x4'
2	GC	Aqu	6.3	21.5583	-0.816667	36.2 kly	12.9'
3	GC	CVn	6.3	13.7033	28.3833	30.6 kly	16.2'

Tipo di dato

```
mysql> DESCRIBE Messier;
```

Field	Type	Null	Key	Default	Extra
M	int(11)	NO	PRI		
Type	char(2)	YES	MUL	**	
Const	char(3)	YES		***	
Mag	float	YES		NULL	
Ra_h	float	YES		NULL	
Ra_m	float	YES		NULL	
Dec_d	float	YES		NULL	
Dec_p	float	YES		NULL	
Dist	char(20)	YES		NULL	
App_size	char(20)	YES		unknown	

Foreign key
(FK)

Primary key (PK)

Valore di default

DBMS (Data Base Management System)

Relazioni:

▼ Messier

🔑 M

◆ Type

◆ Const

◆ Mag

◆ Ra_h

◆ Ra_m

◆ Dec_d

◆ Dec_p

◆ Dist

◆ App_size

M	Type	Const	Mag	Ra	Decl	Dist	App_size
1	BN	Tau	8.2	5.575	22.0167	6.3 kly	6'x4'
2	GC	Aqu	6.3	21.5583	-0.816667	36.2 kly	12.9'
3	GC	CVn	6.3	13.7033	28.3833	30.6 kly	16.2'
4	GC	Sco	6.4	16.3917	-25.475	6.8 kly	26.3'
5	GC	Ser	6.2	15.31	2.08333	22.8 kly	17.4'
6	OC	Sco	4.2	17.6733	-31.77	2 kly	33'
7	OC	Sco	4.1	17.8983	-33.2167	NULL	80.0'
8	BN	Sag	6	18.0683	-23.7	5200 ly	90'x40'
9	GC	Oph	7.3	17.32	-17.4833	26.4 kly	9.3'

▼ TypeDescr

🔑 Type

◆ Descr

🔑 Type	Descr
OC	Open Cluster
GC	Globular Cluster
GX	Galaxy

DBMS (Data Base Management System)

Relazioni:

▼ Messier

- 🔑 M
- FK 🔹 Type
- 🔹 Const
- 🔹 Mag
- 🔹 Ra_h
- 🔹 Ra_m
- 🔹 Dec_d
- 🔹 Dec_p
- 🔹 Dist
- 🔹 App_size

Relazione ∞:1

▼ TypeDescr

- 🔑 Type PK
- 🔹 Descr

M	Descr	Const	Mag	Ra	Decl
1	Bright Nebula	Tau	8.2	5.575	22.0167
2	Globular Cluster	Aqu	6.3	21.5583	-0.816667
3	Globular Cluster	CVn	6.3	13.7033	28.3833
4	Globular Cluster	Sco	6.4	16.3917	-25.475
5	Globular Cluster	Ser	6.2	15.31	2.08333
6	Open Cluster	Sco	4.2	17.6733	-31.77

DBMS (Data Base Management System)

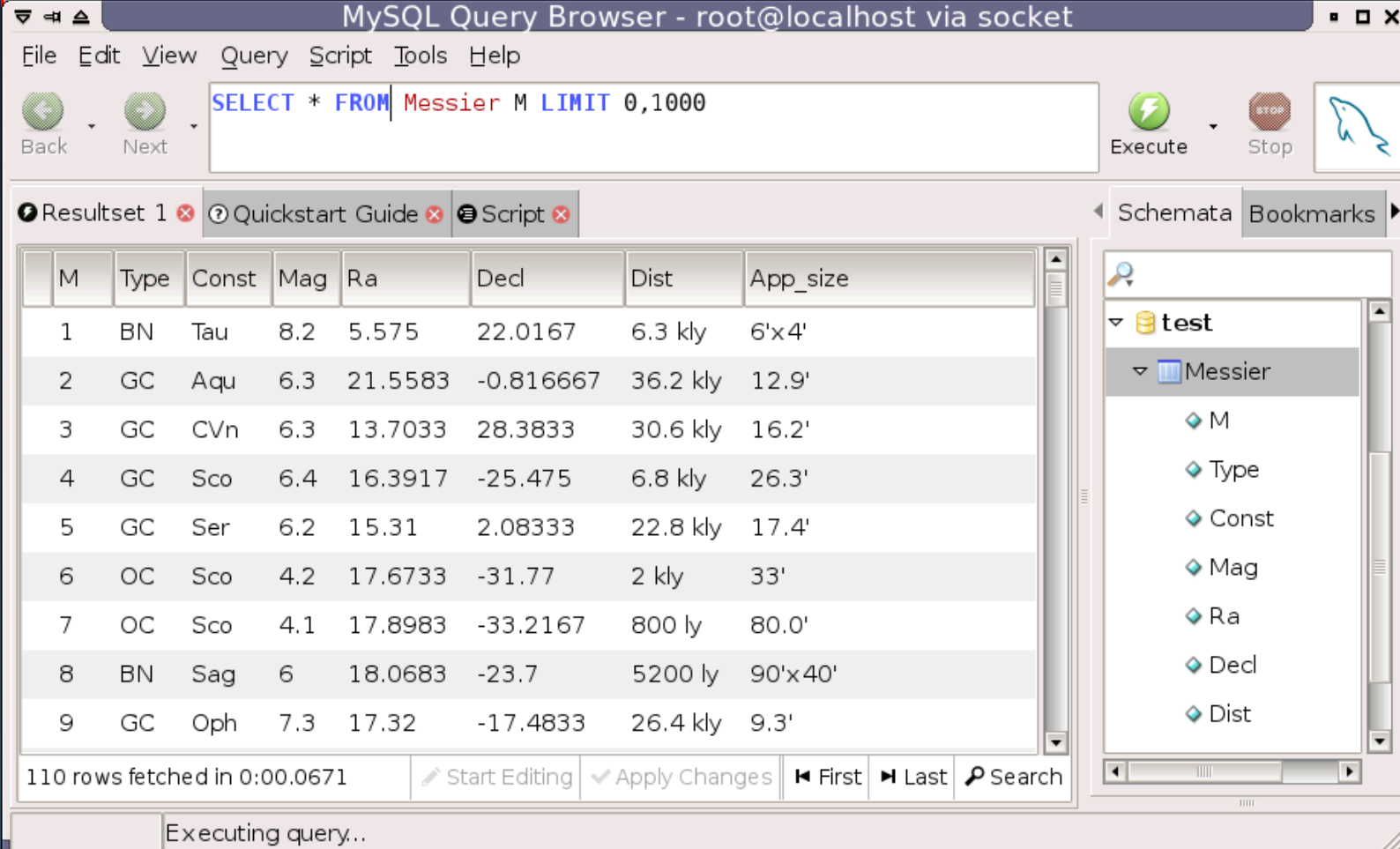
SQL (Structured Query Language):

- Implementato per la prima volta nel 1970 (con il nome di SEQUEL, Structured English Query Language) per il "System R" di IBM;
- Adottato come standard da ANSI (1986) e ISO (1987) con il nome di SQL (la cui pronuncia corretta é "es-q-el");
- Ne esistono varie versioni:
 - SQL-86, SQL-87 (adottato anche da ISO);
 - SQL-89;
 - SQL-92 (il più usato);
 - SQL-99 (aggiunta di espressioni regolari, subqueries, triggers, etc...);
 - SQL-2003 (XML e altre funzionalità avanzate, ma non é distribuito gratuitamente!!!);

DBMS (Data Base Management System)

Come si usa SQL:

- Da terminale:
- Da interfaccia grafica:



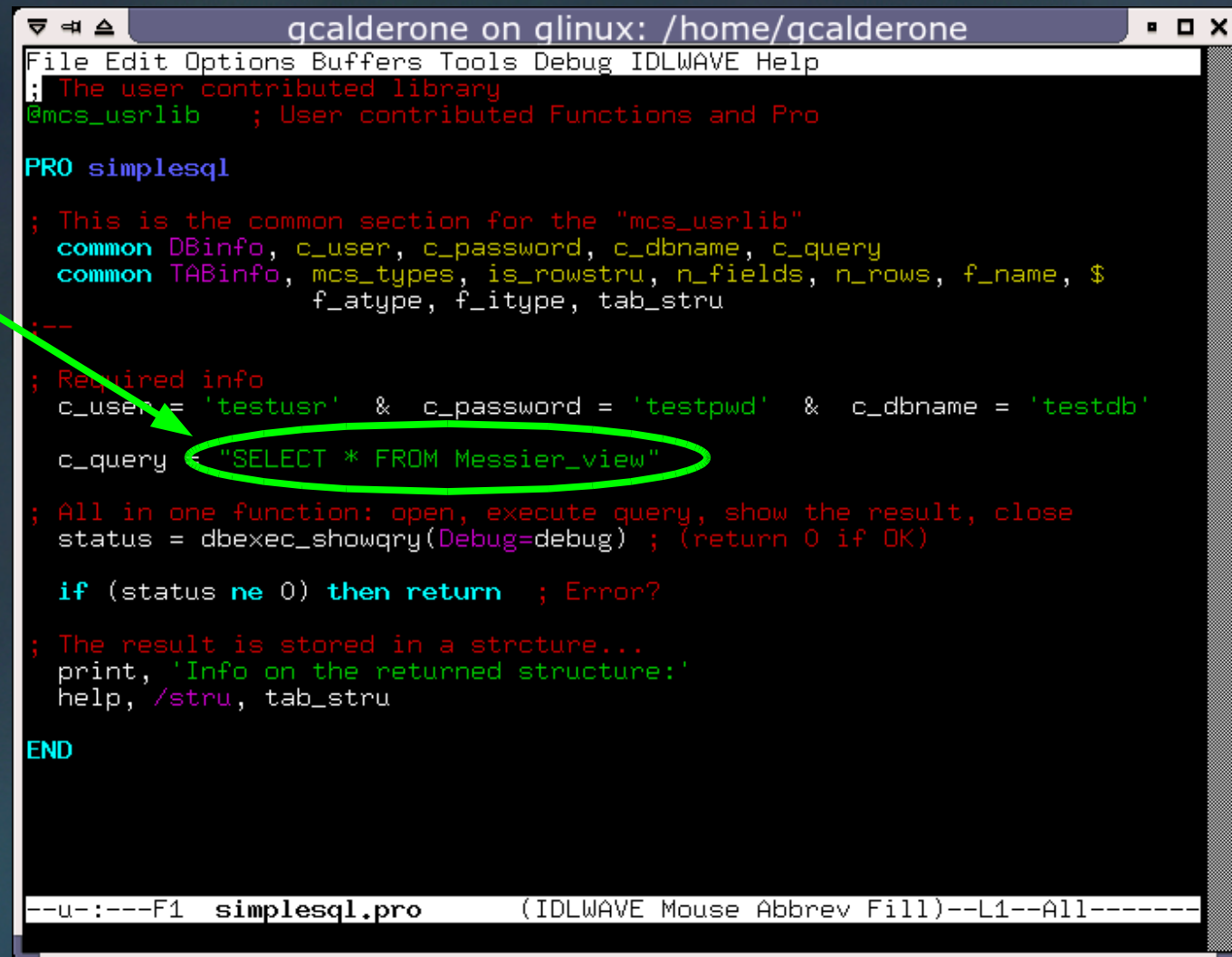
The screenshot shows the MySQL Query Browser interface. The title bar reads "MySQL Query Browser - root@localhost via socket". The menu bar includes File, Edit, View, Query, Script, Tools, and Help. The query editor contains the SQL statement: `SELECT * FROM Messier M LIMIT 0,1000`. Below the query editor are "Back" and "Next" buttons, and "Execute" and "Stop" buttons. The main area displays a table with 9 rows of data. The table has columns: M, Type, Const, Mag, Ra, Decl, Dist, and App_size. The status bar at the bottom indicates "110 rows fetched in 0:00.0671" and includes buttons for "Start Editing", "Apply Changes", "First", "Last", and "Search". A "Schemata" panel on the right shows the database structure for "test", including a "Messier" table with columns M, Type, Const, Mag, Ra, Decl, and Dist.

M	Type	Const	Mag	Ra	Decl	Dist	App_size
1	BN	Tau	8.2	5.575	22.0167	6.3 kly	6'x4'
2	GC	Aqu	6.3	21.5583	-0.816667	36.2 kly	12.9'
3	GC	CVn	6.3	13.7033	28.3833	30.6 kly	16.2'
4	GC	Sco	6.4	16.3917	-25.475	6.8 kly	26.3'
5	GC	Ser	6.2	15.31	2.08333	22.8 kly	17.4'
6	OC	Sco	4.2	17.6733	-31.77	2 kly	33'
7	OC	Sco	4.1	17.8983	-33.2167	800 ly	80.0'
8	BN	Sag	6	18.0683	-23.7	5200 ly	90'x40'
9	GC	Oph	7.3	17.32	-17.4833	26.4 kly	9.3'

DBMS (Data Base Management System)

Come si usa SQL:

- Da terminale:
- Da interfaccia grafica:
- Da linguaggio di programmazione:



```
gcalderone on qlinux: /home/gcalderone
File Edit Options Buffers Tools Debug IDLWAVE Help
; The user contributed library
@mcs_usrlib ; User contributed Functions and Pro

PRO simplsql

; This is the common section for the "mcs_usrlib"
common DBinfo, c_user, c_password, c_dbname, c_query
common TABinfo, mcs_types, is_rowstru, n_fields, n_rows, f_name, $
                f_atype, f_itype, tab_stru
---
; Required info
c_user = 'testusr' & c_password = 'testpwd' & c_dbname = 'testdb'
c_query = "SELECT * FROM Messier_view"

; All in one function: open, execute query, show the result, close
status = dbexec_showqry(Debug=debug) ; (return 0 if OK)

if (status ne 0) then return ; Error?

; The result is stored in a structure...
print, 'Info on the returned structure:'
help, /stru, tab_stru

END

---u-:---F1 simplsql.pro (IDLWAVE Mouse Abbrev Fill)--L1--All-----
```


DBMS (Data Base Management System)

SQL, istruzioni base:

- Lettura record: `SELECT`
- Inserimento record: `INSERT`
- Aggiornamento record: `UPDATE`
- Eliminazione record: `DELETE`

NOTA:

tutte le operazioni di scrittura sono "record-based", cioè non si può cancellare un solo campo.

DBMS (Data Base Management System)

SQL, esempi di lettura:

- Lettura di tutti i record:

```
SELECT * FROM Messier;
```

- Lettura di alcuni campi, tutti i record:

```
SELECT M, Ra, Decl FROM Messier;
```

- Lettura di alcuni record:

```
SELECT * FROM Messier WHERE Type = 'GC';
```

- Lettura di tutti i record ordinati per magnitudine:

```
SELECT * FROM Messier ORDER BY Mag;
```


DBMS (Data Base Management System)

SQL, esempi di scrittura:

- Inserimento di un nuovo record:

```
INSERT INTO Messier(M, Ra, Decl) VALUES(111, 1.2, 3.4);
```

- Aggiornamento di un record:

```
UPDATE Messier SET Mag=6.0 WHERE M = 111;
```

- Eliminazione di un record:

```
DELETE FROM Messier WHERE M = 111;
```

NOTA:

non bisogna mai dimenticare le clausole **WHERE** nelle operazioni di **UPDATE** e **DELETE**, altrimenti tutti i record della tabella verranno modificati.

DBMS (Data Base Management System)

SQL, esempi di lettura con funzioni di aggregazione:

- Quanti record contiene la tabella:

```
SELECT count(*) FROM Messier;
```

- Subtotale per tipo di oggetto:

```
SELECT type, count(type) FROM Messier GROUP BY type;
```

- Media delle magnitudini per tipo di oggetto:

```
SELECT type, avg(Mag) FROM Messier GROUP BY type;
```


DBMS (Data Base Management System)

SQL, esempio di join:

- Lettura dei record e della relazione con tabella TypeDescr:

```
SELECT Messier.M, Messier.Type, TypeDescr.Descr  
FROM Messier LEFT JOIN TypeDescr  
ON Messier.Type = TypeDescr.Type;
```


DBMS (Data Base Management System)

SQL, esempio di view

Join automatica con tabella TypeDescr

```
CREATE VIEW Messier_view AS
SELECT
    Messier.M,TypeDescr.Descr AS Descr,
    Messier.Const,
    Messier.Mag,
    Messier.Ra,
    Messier.Decl,
    Messier.Dist,
    Messier.App_size
FROM Messier LEFT JOIN TypeDescr ON
    Messier.Type=TypeDescr.Type;

SELECT * FROM Messier_view;
```


DBMS (Data Base Management System)

SQL, creazione di funzioni (UDF):

Conversione di data civile in giorno Giuliano

```
CREATE FUNCTION jdcnv(yr INT, mn INT, day INT, hr DOUBLE)
  RETURNS DOUBLE
  BEGIN
    DECLARE L DOUBLE;
    DECLARE julian DOUBLE;

    SET L = (mn-14)/12;
    SET julian = day - 32075 + 1461*(yr+4800+L)/4 +
                367*(mn - 2-L*12)/12 - 3*((yr+4900+L)/100)/4;

    SET julian = julian + hr/24.0 - 0.5;

    RETURN julian;
  END//

SELECT jdcnv(2006, 12, 15, 0);
```


DBMS (Data Base Management System)

SQL, esempio di view (2):

Una view più sofisticata:

```
CREATE VIEW Messier_view AS
    SELECT Messier.M,
           TypeDescr.Descr AS Descr,
           Messier.Const,
           Messier.Mag,
           Messier.Ra,
           Messier.Decl,
           hadec2alt(Ra, Decl, 38) AS Alt,
           hadec2az(Ra, Decl, 38) AS Az,
           Messier.Dist,
           Messier.App_size
    FROM Messier LEFT JOIN TypeDescr
    ON Messier.Type = TypeDescr.Type
    ORDER BY Descr, M;
```

```
SELECT * FROM Messier_view;
```


DBMS (Data Base Management System)

Privilegi degli utenti:

- Account utente con username e password;
- Connessioni sicure tramite SSL;
- Diversi tipi di diritti:
 - Accesso;
 - Lettura;
 - Scrittura;
 - Creazione tabelle, funzioni, ecc...
- Diritti applicabili su diversi livelli:
 - Database;
 - Tabella;
 - Colonna;
 - Record (su MySQL, tramite MyRO);

DBMS (Data Base Management System)

Supporto transazioni:

- Alcune operazioni di scrittura necessitano di più istruzioni SQL;
- Ma l'atomicità di scrittura (meccanismo di LOCKING delle tabelle) è attivo soltanto durante l'esecuzione di UNA istruzione SQL;

=> si possono verificare casi di inconsistenza dei dati.
- Il meccanismo di transazione permette di risolvere il problema effettuando la scrittura in UNICA SOLUZIONE dopo l'ultima query SQL;
 - BEGIN TRANSACTION;
 - COMMIT;
 - ROLLBACK;

DBMS (Data Base Management System)

Ottimizzare le query con gli indici:

- Vengono usati per ottimizzare:
 - Operazioni di ricerca di record;
 - Join di tabelle;
- Nei casi più favorevoli (hash) il tempo di ricerca é costante e indipendente dalla dimensione del database;
- Alcuni fra gli indici più diffusi sono:
 - **Hash**: il più veloce, ma funziona solo con operatori di uguaglianza (=) o disuguaglianza (<>);
 - **B-tree**: il più diffuso, funziona anche con operatori di confronto (<, <=, >, >=, BETWEEN, LIKE), (complessità $O(\log n)$);
- Rendono più lente le operazioni di scrittura su database e richiedono memoria di massa aggiuntiva;

DBMS (Data Base Management System)

Indicizzazione su geometria sferica:

- La ricerca più semplice é del tipo:

```
SELECT ... WHERE  
lat > LatInf AND lat < LatSup AND  
lon > LonInf AND lon < LonInf;
```

In questo caso l'indice sui campi lat e long viene utilizzato ma la regione selezionata non é rettangolare (distorsioni ai poli);

- Una ricerca più sofisticata potrebbe essere la selezione circolare:

```
SELECT ... WHERE  
Angular_Distance(Clat, Clon, lat, lon) < Radius;
```

Ma in questo caso non verrebbero utilizzati gli indici;



Non c'é un modo efficace di fare questa query utilizzando soltanto SQL !

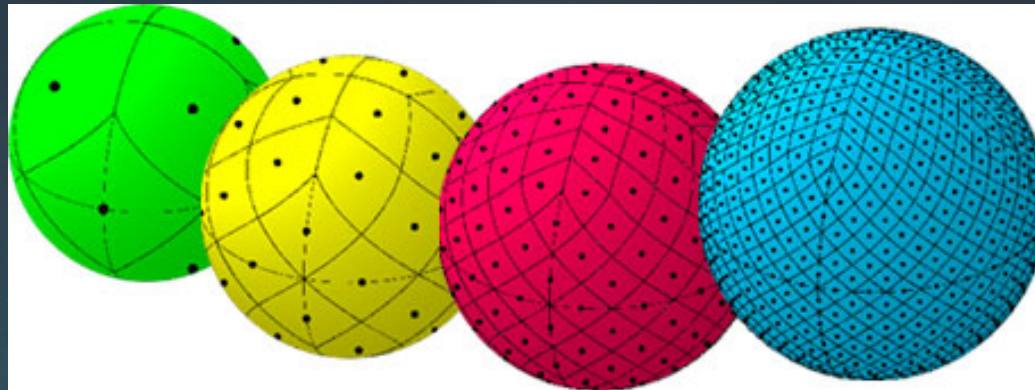
DBMS (Data Base Management System)

Indicizzazione alternativa:

- HTM (Hierarchical Triangular Mesh):



- HealPix (Hierarchical Equal Area isoLatitude Pixelization):

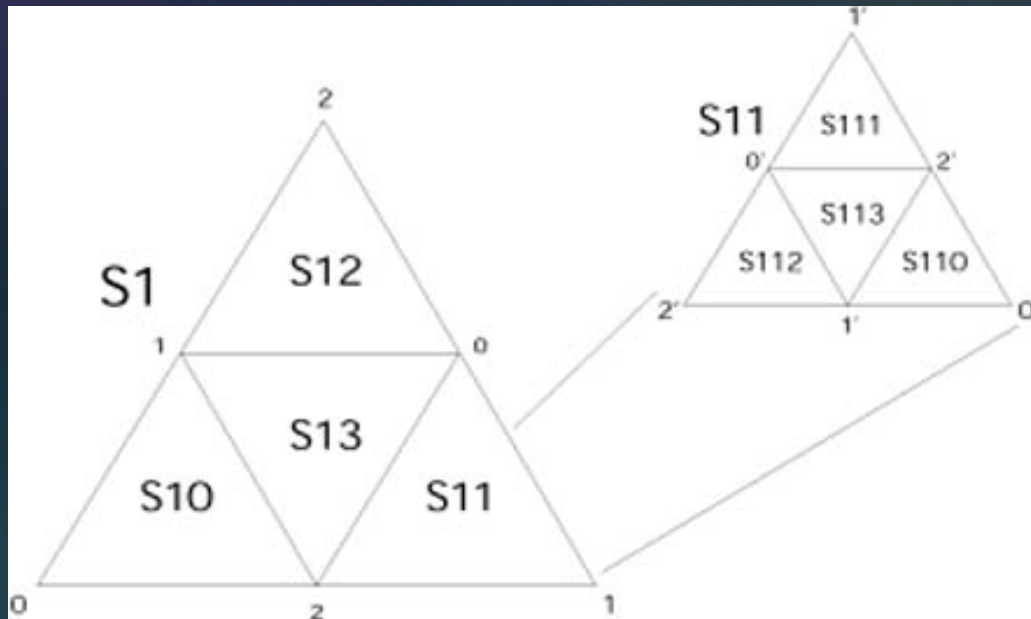


DBMS (Data Base Management System)

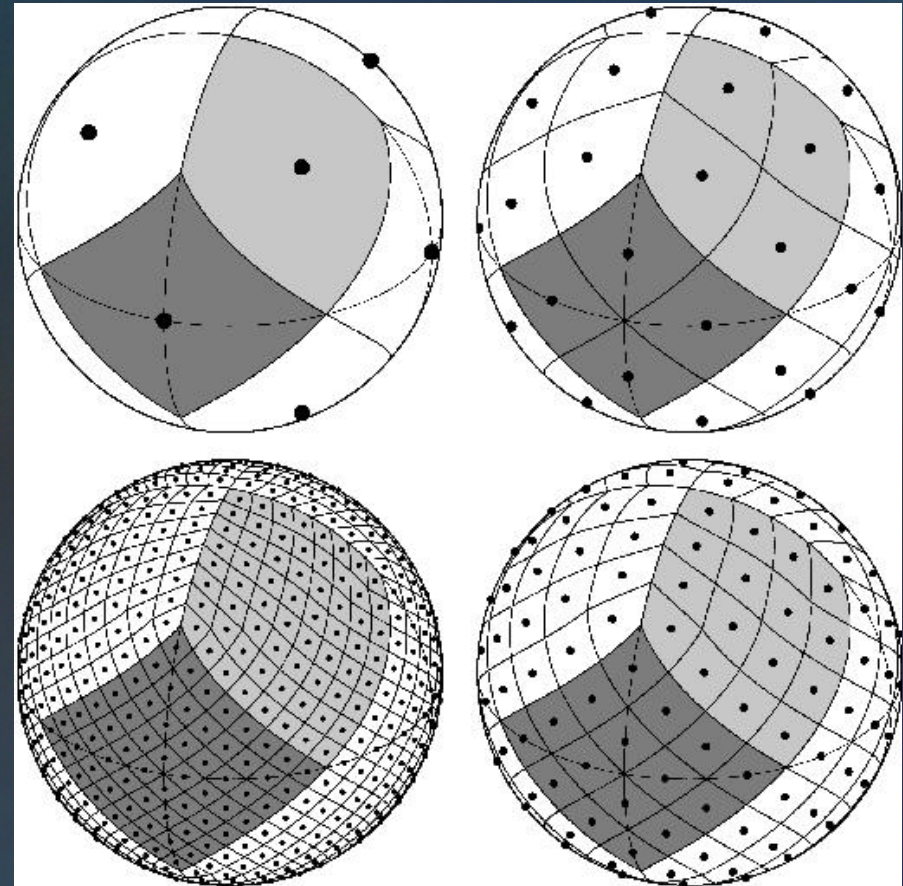
Indicizzazione alternativa:

- Che significa "hierarchical" ?

HTM



HealPix



DBMS (Data Base Management System)

Indicizzazione su geometria sferica con HTM:

- Aggiungiamo una colonna contenente l'HTM id alla tabella Messier:

```
ALTER TABLE Messier
ADD COLUMN htmID INT,
ADD INDEX(htmID);
```

- Definiamo una nuova funzione:

```
CREATE FUNCTION HTM_Lookup(Ra DOUBLE, Decl DOUBLE)
RETURNS INT ...
```

Questa funziona ritorna l'htmID del pixel in cui cadono le coordinate indicate.

- Una semplice ricerca é ora:

```
SELECT ... WHERE htmID = HTM_Lookup(CRa, CDecl);
```

- Query più complesse (selezioni su domini di qualsiasi forma) sono possibili utilizzando il DIF (Dynamical Index Facility)

DBMS (Data Base Management System)

... e in 3 dimensioni ?

- Un sistema a 3 dimensioni può essere realizzato aggiungendo alle coordinate lat/long una terza coordinata: la distanza;
- Per ottenere una indicizzazione completa in questo sistema di riferimento basta aggiungere un indice sul campo relativo alla distanza;

```
WHERE ... AND Dist BETWEEN 3.0 AND 3.5;
```

E' possibile valorizzare automaticamente il campo htmID utilizzando i TRIGGER:

```
CREATE TRIGGER trig_Messier
BEFORE INSERT ON Messier FOR EACH ROW
BEGIN
    SET NEW.htmID = HTM_Lookup(NEW.Ra, NEW.Decl);
END//;
```


DBMS (Data Base Management System)

Database Engine:

- Mysql offre l'opportunità di creare database Engine:
- Si tratta di software che permette di integrare qualsiasi tipo di file all'interno del DBMS;
- Si possono quindi fare i operazioni di lettura e scrittura su questi file come se fossero normali tabelle, in maniera completamente trasparente;
- Stiamo lavorando ad un database Engine per l'accesso a file FITS e VOTable;

DBMS (Data Base Management System)

Catalogo astronomico su RDBMS:

- I moderni RDBMS hanno le seguenti caratteristiche:
 - Numero di record: 10^9 - 10^{19} e oltre;
 - Dimensione dei file: Tbytes ($\sim 10^{12}$ bytes) - Exabytes ($\sim 10^{18}$ bytes)
- L'indicizzazione permette la creazione di relazioni fra tabelle così grandi (correlazioni, database multiwavelength, etc...);
- Facilmente espandibili tramite UDF, Stored procedure, etc...;
- Accesso semplice tramite le interfacce SQL:
 - ==> le istruzioni per accedere sono sempre le stesse qualunque sia il DBMS e le dimensioni del DB;

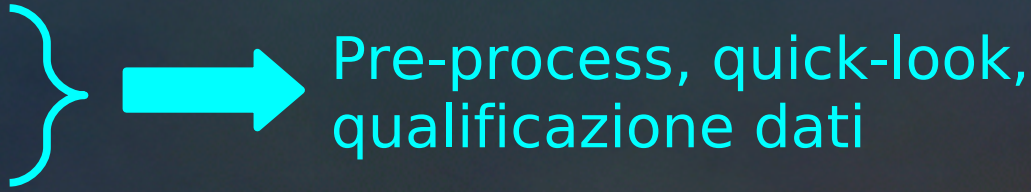
DBMS (Data Base Management System)

Esempio di cataloghi astronomici su RDBMS:

- *GSC 2.2, 2.3* - the Guide Star Catalogue II (thanks Turin Astronomical Observatory);
- *2MASS* - The Two Micron All Sky Survey at IPAC
- *UCAC 2* - The Second U.S. Naval Observatory CCD Astrograph Catalog
- *ASCC 2.5* - The All-Sky Compiled Catalogue V.2: Kharchenko N.V., Kinematics and Physics of Celestial Bodies. 17, 409 (2001);
- *GSPC2* - Guide Star Photometric Catalogue V.2
- *BSC* - Bright Star Catalogue 5th Revised Ed.: Hoffleit D., Warren Jr W.H. (1991).
- *USNO A2.0* - USNO A2.0 at ESO/ST-ECF Dave Monet, US Naval Observatory Flagstaff Station (1998)
- *USNO B1.0* - USNO B1.0 at CDS-VizieR USNO-B1.0 catalog was created by Dave Monet and collaborators at USNO Flagstaff Station Monet D.G., Levine S.E., Casian B., et al. Astron. J. 125, 984 (2003)

DBMS (Data Base Management System)

RDBMS a supporto di progetti in astronomia:

- Oltre i dati scientifici (che tipicamente vanno in file FITS) vengono prodotti una serie di altri dati:
 - House-keeping;
 - Event log;
 - Observation log;

Pre-process, quick-look, qualificazione dati
- Tutti questi dati possono essere prodotti automaticamente (tramite un sistema informativo, ad esempio basato su MCS) e memorizzati in un RDBMS;
- Il RDBMS diventa quindi un repository centralizzato cui possono accedere diversi utenti (con diversi diritti d'accesso):
 - Ricercatori per i dati scientifici;
 - Tecnici per il controllo remoto della strumentazione;
 - Altri utenti (outreach);

DBMS (Data Base Management System)

- E.F.Codd, "A relational model for large shared data banks":
 - <http://www.acm.org/classic/nov95>
- Database e-Learning:
 - <http://db.grussels.org>
- On wikipedia:
 - <http://en.wikipedia.org/wiki/Database>
- MySQL documentation:
 - <http://www.mysql.org>
 - <http://dev.mysql.com/doc/refman/5.1/en/index.html>
- HTM e HealPix
 - <http://www.sdss.jhu.edu/htm>
 - <http://healpix.jpl.nasa.gov>
- MCS, Myro
 - <http://ross.iasfbo.inaf.it/mcs/>
- Materiale su questi seminari:
 - <http://ross.iasfbo.inaf.it/seminariDB>