

Database, SQL & MySQL

Dott. Paolo PAVAN

Maggio 2002



Introduzione a Linux

- Accesso alla consolle
- La shell di linux
 - I comandi essenziali
 - ls
 - cp, rm mkdir
 - editor pico



Input e output dei programmi

Tre canali predefiniti per ricevere e trasmettere dati da e verso i dispositivi di input e output:

lo standard input è il canale attraverso il quale il programma riceve i dati di ingresso; **lo standard output** è il canale di uscita del risultato dell'elaborazione del programma; **lo standard error** è dove il programma stampa eventuali errori occorsi durante l'esecuzione.

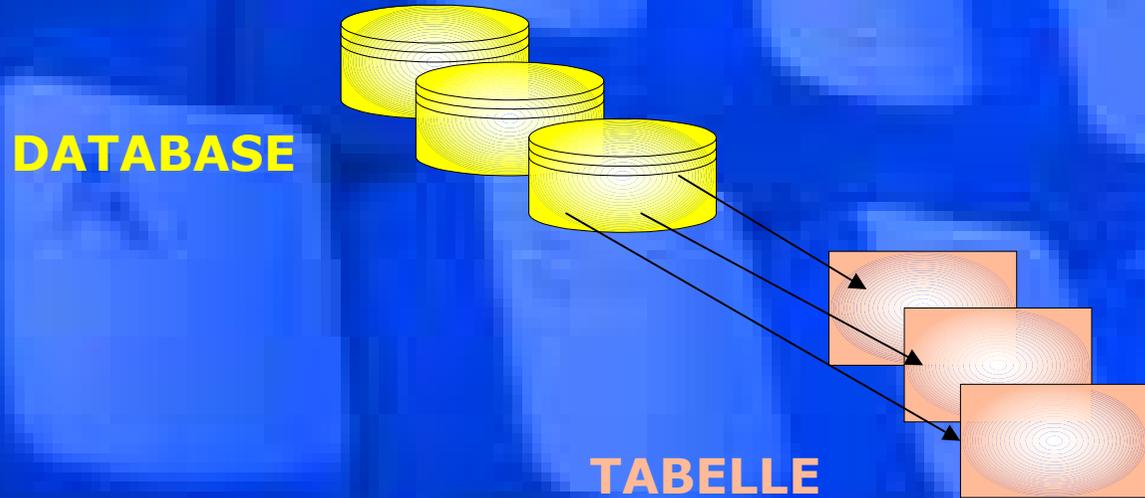
Gli operatori che gestiscono questi canali sono:

- > : redirige lo standard output;
- >> : redirige lo standard output verso un dispositivo aperto in modalità append;
- < : redirige lo standard input;
- 2> : redirige lo standard error;
- | : operatore pipe, concatena standard output e standard input di due programmi.

Struttura RDBMS

MYSQL - RDBMS

DATABASE



TABELLE



Introduzione ai DATABASE

- Database

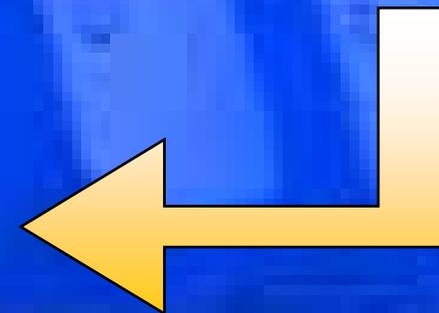
Indica in genere un insieme di dati rivolti alla rappresentazione di un preciso sistema informativo **di vario tipo, aziendale, scientifico, amministrativo o altro**

- Elementi essenziali

- **La struttura dei dati**
- **Le relazioni esistenti tra i dati**

- Rappresentazione schematica

- **Schema CONCETTUALE**
- **Schema LOGICO**
- **Schema FISICO**





Requisiti

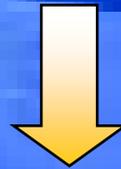
- **REQUISITI (coerenza database)**
 - **Ridondanza minima:** i dati non devono essere duplicati
 - **Multiutenza:** la base dati deve essere unica ma consultabile contemporaneamente da più utenti. In parole povere le informazioni devono essere integrate ma distribuite
 - **Permanenza dei dati:** I dati devono essere fissi e duplicati su memorie fisse (hard disk o stream tape)



Progettazione database

Primo Stadio

- **Schema concettuale**
 - Fase astratta
 - Modelli e teorizzazioni
 - Entry-relationship (E-R entità - relazione)



- **Modello entry-relationship**
 - Questo modello prevede l'individuazione delle entità base concrete ed astratte che andranno a formare insiemi di entità (entity set). Ogni entità avrà i suoi attributi

Entità base --> **TABELLE**

Attributi delle entità base --> **CAMPI**



Progettazione database

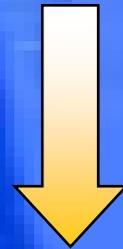
- Legami tra entità
 - Le entità possono avere dei legami (relazioni) rappresentate in base alla loro **CARDINALITA'**
- **CARDINALITA'** (relazioni)
 - Cardinalità 1:1 (uno a uno)
 - **Significa che ad un elemento di A può corrispondere uno ed uno solo elemento di B e viceversa**
 - **Esempio: due entità di un database (tabelle) nazioni e capitali: ad una nazione può corrispondere solo una capitale e viceversa**
 - Cardinalità 1:N (uno a molti)
 - **Significa che ad un elemento di A possono corrispondere più elementi di B ma non viceversa**
 - **Esempio: due entità di un database (tabelle) madri e figlio: ad una madre possono corrispondere più figli, ma un figlio può avere una sola madre**
 - Cardinalità N:N (molti a molti)
 - **Significa che ad un elemento di A possono corrispondere più elementi di B e viceversa**
 - **Esempio: due entità di un database (tabelle) ordini e articoli: un ordine può contenere più articoli e viceversa lo stesso articolo può apparire in più ordini**



Progettazione database

Secondo Stadio

- MODELLO SEMANTICO entry-relationship (schema concettuale) viene espresso attraverso uno **SCHEMA LOGICO**



- Stabilisce i modelli di relazione, di accesso e di interrogazione dei dati



Progettazione database

- **STRUMENTO DBMS (data base management system)**
 - Lo schema logico sarà vincolato dallo strumento DBMS, che verrà utilizzato per creare il database fisico, vero e proprio
- **Tipologie di database**
 - **Database piatto:** tabelle e record non sono collegati. E' impossibile creare strutture di dati relazionati
 - **Database gerarchico:** i dati sono collegati utilizzando puntatori complessi. Il sistema è difficilmente gestibile ed integrabile. Richiede l'opera di un programmatore esperto
 - **Database relazionale:** i dati possono essere collegati tra loro con esempi relazioni impostate attraverso il linguaggio SQL



Progettazione database

- E' nello schema logico che viene scelto lo strumento per la manipolazione dei dati.
- SQL (structured query language)
 - Consente di implementare lo schema logico di un database attraverso la creazione di tabelle, query, relazioni. Si può dire che l'sql è lo strumento per realizzare in pratica (schema fisico) ciò che viene progettato come schema logico



Progettazione database

Terzo Stadio

- **Schema fisico**
 - Prevede la costituzione del database su di una specifica architettura dotata di uno specifico sistema operativo
 - Normalizzazione dei dati
 - Integrità referenziali
- **SOFTWARE per DB RELAZIONALI**
 - Traduce sul computer lo schema concettuale attraverso l'applicazione delle regole imposte dallo schema logico:
 - ACCESS 95/97/2000: applicativo per database su piattaforma intel, con sistema operativo windows 95/98/2000
 - Altri software concorrenti: file maker, lotus approach



Strutturazione database

- Elementi fisici del database
 - **Campi:** sono le voci che costituiscono un record
 - **Record:** rappresenta la registrazione di un singolo dato, descritto da più campi
 - **Tabelle:** sono costituite dall'insieme dei record
 - **Database:** è costituito dalla singola tabella o da più tabelle eventualmente tra loro relazionate

Com_Cod	Com_Descrizione	Com_Costo	Lab_Cod
0010	Morriano laterale 100 cm	50000	0010
0020	Morriano laterale 200 cm	100000	0010
0030	Bianco 60 cm	50000	0050
0040	Pratic tavolo lungo	120000	0050
0050	Piano tavolo quadrato	120000	0050
0060	Gamba tavolo 60 cm	20000	0020
0070	Gamba tavolo 100 cm		Record
0080	Arma 100 cm	80000	0040
0090	Armi 200 cm	120000	0010
0100	Carrozzina da incasso	200000	0030
0110	Bussola 10 classe	10000	0010
0120	Bussola 10 viti	50000	0010
0130	Pannello di allacciatura	50000	0080
0140	Bastone appando al f	10000	0060
0150	Pannello posteriore 100 cm x 30	30000	0050

TABELLA

Campo



Concetti Base dei DB

Chiave Primaria

- Un RDBMS necessita sempre di un modo per identificare in maniera univoca ciascun record che viene inserito nella tabella.
- Questo elemento è conosciuto come **Chiave Primaria**
 - La chiave primaria viene identificata con un campo (o con un insieme di campi) che **rendono unico il record inserito**.
 - In genere il campo scelto come Chiave Primaria della tabella non può mai essere duplicato, quindi non si può ripetere (è unico).



Concetti Base dei DB

Indice

- Associare il valore indice ad un campo farà sì che questo venga indicizzato
- I dati vengono cercati e ritrovati più facilmente, se registrati (inseriti) secondo una sequenza alfabetica invece che con un ordine casuale.
 - Un indice può essere composto da uno o più campi
 - Conviene non indicizzare troppi campi pena il rallentamento dell'inserimento dei dati



Introduzione all'SQL

- Caratteristica essenziale del linguaggio SQL (structured query language) è sicuramente la sua relativa semplicità sintattica e la sua flessibilità
- Questo linguaggio è al tempo stesso semplice e potente
- E' un linguaggio universale, consente il dialogo con qualsiasi database moderno.



Concetti base di SQL

E' un linguaggio specializzato e non procedurale:

- *Specializzato*: è specializzato verso l'uso di tabelle e basi dati, non può essere usato per fare grafica, non è un linguaggio generico
- *Non procedurale*: con l'sql non vengono costruite procedure, non si spiega al sistema come devono esser fatte delle cose ma si indica chiaramente come ottenere un risultato, ad esempio il risultato di una query.



Istruzioni in SQL

- L'SQL è un linguaggio che consente di fornire *istruzioni* o comandi con i quali si formano enunciati. Alcuni comandi possiedono *clausole* e *predicati*.
- Questi comandi vengono suddivisi in 3 categorie:
 - **DDL** (*data definition language*): in questa categoria rientrano i comandi che servono per creare e modificare le tabelle e quindi per definire i dati. Ad esempio i comandi INSERT o UPDATE.
 - **DML** (*data manipulation language*): in questa invece ricadono i comandi con cui si manipolano i dati per ricavarne le informazioni necessarie. Ad esempio il comando SELECT.
 - **DCL** (*data control language*): in questa invece ricadono i comandi con cui stabiliscono priorità e accessi alle tabelle. Ad esempio i comandi GRANT o REVOKE.



Standard SQL

- E' un linguaggio standard ANSI, anche se i vari RDBMS lo implementano in modo più o meno completo.
- Tutti i vari RDBMS (oracle, informix, mysql, postgres, ingres) dovrebbero cercare di aderire il più possibile allo standar ANSI SQL



Dialetti SQL

- L'American National Standard Institute (ANSI) presidia e cura le specifiche del linguaggio SQL a cui tutti gli RDBMS devono adattarsi se vogliono definirsi tali. Tre specifiche SQL:
 - SQL1 emessa nel 1986
 - SQL2 emessa nel 1992
 - SQL3 emessa nel 1999: questo standard è stato fatto proprio anche dall'ISO (international standard organisation). Ha introdotto alcune novità come i cicli o le transazioni (ovvero la possibilità di eseguire più comandi in successione come se fossero un unico comando).

SQL Sintassi di un comando

- L'SQL è un linguaggio che consente di fornire *istruzioni* o comandi con i quali si formano enunciati. Alcuni comandi possiedono operazioni e *clausole*.
<comando>=<operazione> <clausola>[<clausola>.....]
- Più precisamente un comando si esprime in questo modo:
 - *COMANDO* <lista di selezione><espressione di tabella>[<clausola di ordinamento>]
 - *dove* <lista di selezione>
 - Elenco dei campi separati da virgole
 - *dove* <espressione di tabella>
 - <clausola FROM>
 - <clausola WHERE>
 - <clausola GROUP_BY>
 - <clausola HAVING>
 - [<clausola di ordinamento>]
 - ORDER BY ASC o ORDER BY DESC
- *Esempio*
 - *SELECT * from elenco WHERE Citta='Torino' ORDER BY nome ASC;*



Comandi base

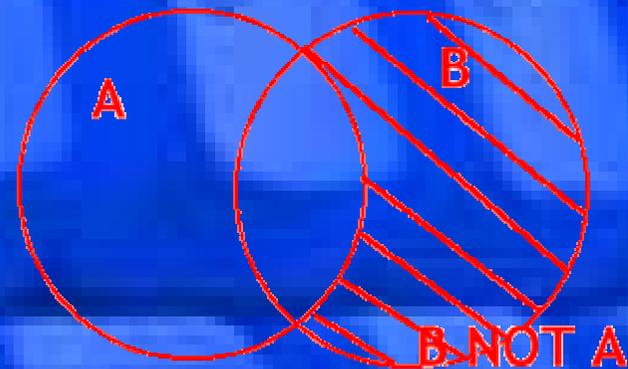
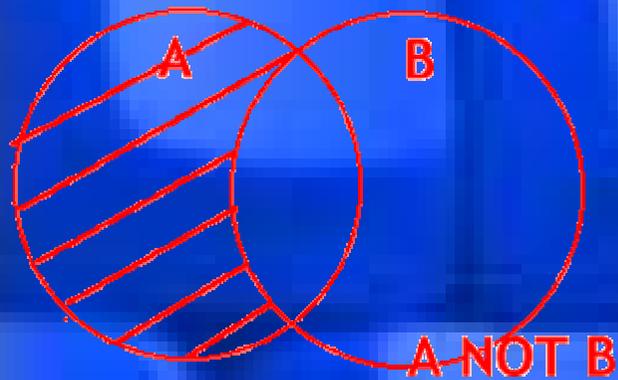
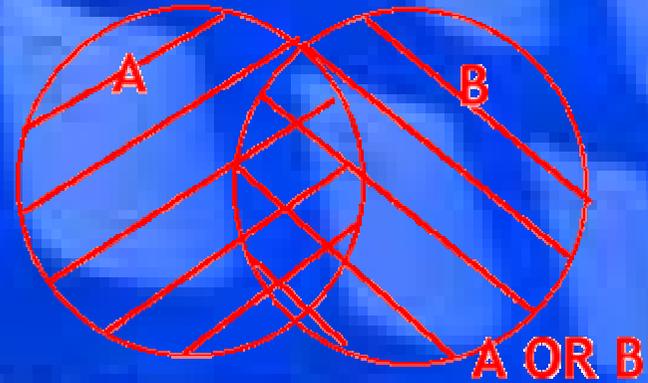
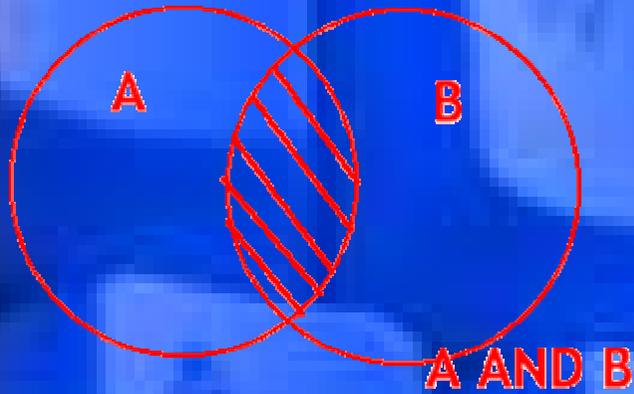
- Per inserire dati nel database useremo il comando INSERT
- Per modificare dati dal database useremo il comando UPDATE
- Per cancellare dati dal database useremo il comando DELETE
- Per richiedere dati dal database useremo il comando SELECT



Esempi di comandi

- `SELECT * FROM tabella WHERE campo = "condizione"`
- `INSERT INTO tabella (campo1,campo2,campon...)VALUES (valore campo1,valore campo2, valore campon)`
- `UPDATE tabella SET campo = valore campo WHERE campo = "condizione"`
- `DELETE FROM tabella WHERE campo = "condizione"`

Operatori Booleani





Privilegi sui database

- Per consentire ad un utente diverso dal proprietario di manipolare un database si può utilizzare il comando GRANT:
 - GRANT <privilegi> ON <banca dati>.* TO <utente>@<host> IDENTIFIED BY '<password>'\g
- Per revocare allo stesso modo si può usare il comando REVOKE



MySQL

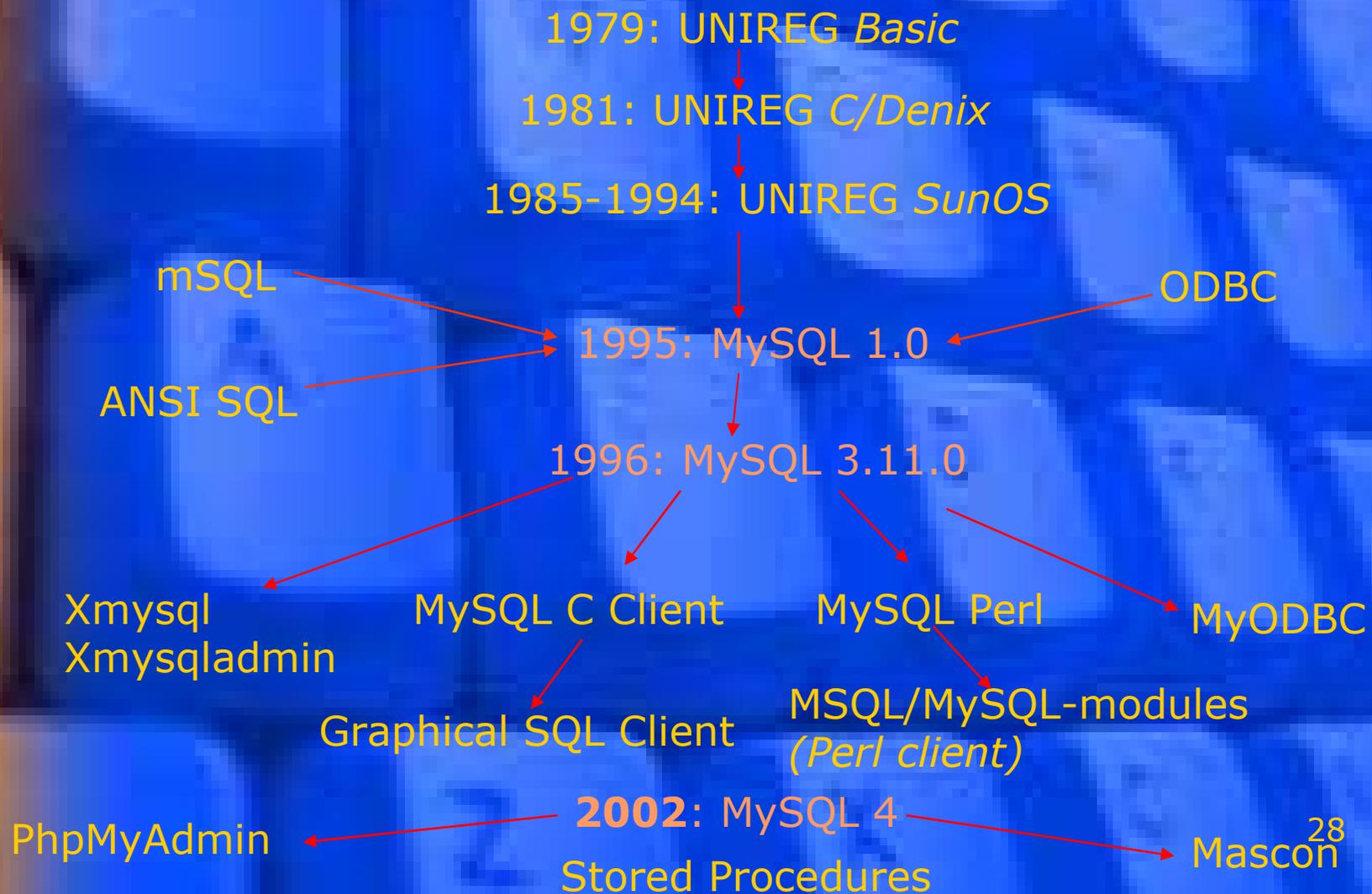
- E' un server/client SQL server
- Deriva dal progetto del suo progenitore mysql
- E' veloce e affidabile
- E' scalabile e multiplatforma
- E' stato progettato per la gestione di database di grandi dimensioni
- E' un RDBMS open-source
- Si pronuncia: my ess que ell

MySQL & RDBMS

- Mysql visti gli elementi coinvolti è **per definizione** (tabelle, colonne, righe, ecc.) Un SISTEMA **RELAZIONALE** per la gestione di basi di dati (in sigla, RDBMS)
- Per questo motivo possiamo dire che mysql è un RDBMS con supporto SQL.
- SQL è il linguaggio con cui vengono creati e manipolati contenitori e dati contenuti.



Storia di MySQL



Piattaforme supportate

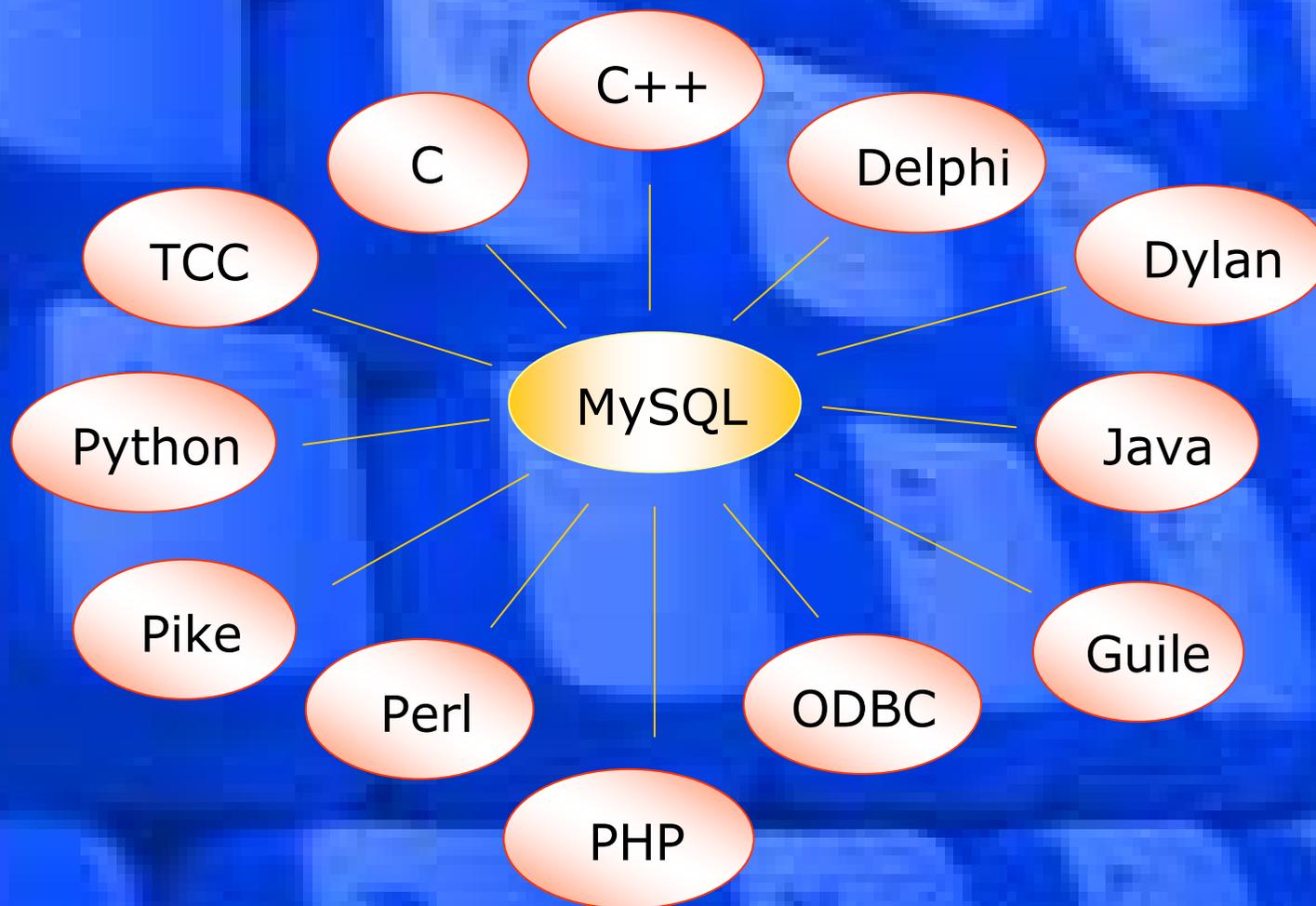
- Native
 - AIX 4.x
 - BSDI 3.0/3.1
 - DEC Tru64 Unix
 - FreeBSD 3.x
 - HP-UX 11.x
 - OS/2 Warp
 - SCO UnixWare 7.0.1
 - SGI Irix 6.x
 - Win95, Win98, NT
- Linux Threads or glibc
 - Linux 2.0
- MIT-Pthreads
 - BSDI 2.x
 - FreeBSD 2.x
 - HP-UX 10.20
 - NetBSD 1.3 Intel
 - NetBSD 1.3 Alpha
 - Open BSD 2.x
 - SunOS 4.x
- FSU-Pthreads
 - SCO open server



Ambienti di programmazione

- MySQL può essere usato come RDBMS in molti ambienti di programmazione
- Tantissime sono le application program interface ovvero gli ambienti verso cui MySQL si interfaccia
- Oggi è uno dei supporti per la gestione dei dati nei programmi più diffusi.

API supportate



Diffusione di MySQL

- Risulta diffusissimo perché protetto da licenza GPL e quindi utilizzabile senza costi
- Ha avuto negli ultimi due anni una diffusione davvero notevole
- Si stimano oramai in centinaia di migliaia le installazioni di mysql



Caratteristiche oggi

- Multithreaded, multiuser e velocissimo
- Mysql è dotato di sistemi per interfacciarsi ad altri DB tramite ODBC e sono oramai diffusi tool per la sua gestione via web (phpmyadmin) o client/server (mascon)
- E' portabile e scalabile
- Supporta le specifiche sancite dall' ANSI SQL 92 e le funzioni della versione 3.0 di ODBC 3.0
- Supporta molti linguaggi e tipologie di caratteri
- E' molto documentato e diffuso su internet
- E' molto robusto soprattutto perché diffusissimo e testato da migliaia o forse milioni di utenti.



Chi sostiene il progetto MySQL

- Allan Larsson: il boss di tcx
- David Axmark: si occupa di licenziarlo della documentazione e della sua distribuzione via web
- Antti Halonen: direttore commerciale
- Michael 'monty' Widenius: il suo principale sviluppatore
- Milioni di utenti su Internet che si appoggiano alla mailing list ed al sito ufficiale <http://www.mysql.com>



RDBMS in commercio

- Relational DBMS (RDBMS)
 - RDBMS commerciali
 - Oracle, informix, MSSQL server
 - GNU general public license software
 - MySQL, msql or postgresQL.

Futuro di MySQL

- Piena adesione agli standard ANSI SQL
- Mysql verso la licenza GPL
- Sacrificio della velocità per il supporto delle nuove features quali:
 - Transactions
 - Tecnologia ISAM e lock dei dati
 - Supporto tabelle INNODB: COMMIT e ROLLBACK
 - Evoluzione verso stored procedures e triggers



Versioni di MySQL

- La versione stabile oggi è la 3.23.X.
Root@box:~# mysql -V
Mysql ver 11.15 distrib 3.23.39, for
slackware-linux-gnu (i386)
- La versione 4.X è attualmente in fase di alfa testing.

Tipologie di dati per mysql

Tipi numerici

- **Int**
 - Un intero normale. Assume valori tra -2147483648 e 2147483647 se usato con segno, tra 0 e 4294967295 se usato senza segno
 - Int(3)
- **Real (double)**
 - Consente valori tra -1.7976931348623157E+308 e -2.2250738585072014E-308, 0, e tra 2.2250738585072014E-308 e 1.7976931348623157E+308

Tipologie di dati per MySQL

Tipi stringa

- **Char(length)**
 - Una stringa a lunghezza fissa, allineata a destra con spazi vuoti di riempimento. Il range di M è tra 1 e 255
 - Char(10)
- **Varchar(length)**
 - Una stringa a lunghezza variabile. Gli eventuali spazi vengono rimossi quando il valore viene memorizzato. Il range di M è tra 1 e 255
 - Varchar(10)

Tipologie di dati per mysql

tipi data

- **Date**
 - Y2K compliant
 - Una data, variabile nell'intervallo '1000-01-01' - '9999-12-31'. Mysql visualizza i valori DATE nel formato 'YYYY-MM-DD'
- **Time**
 - Un orario, nel range '-838:59:59' - '838:59:59'. Mysql visualizza questi valori nel formato 'HH:MM:SS'
- **Datetime**
 - Una combinazione di data e ora attuali, nel range '1000-01-01 00:00:00' - '9999-12-31 23:59:59'. Mysql visualizza i valori DATETIME nel formato 'YYYY-MM-DD HH:MM:SS'

Gestione del server MySQL

- L'accesso ad un server MySQL avviene sempre attraverso un software client che può essere rappresentato da programmi diversi:
 - mysql client
 - Xmysql
 - phpMyAdmin
 - Mascon
- Attraverso questi sistemi viene gestito il server MySQL e tutti i database in esso contenuti.

Accesso al server MySQL

- Per poter manipolare un database occorre possedere i privilegi di accesso.
- Prima di poter accedere al server sql occorre avere le permission di accesso ed essere dotati di:
 - user
 - password
- Di default il sistema accetta solo connessioni locali (**localhost**) ma può essere indotto ad accettare connessioni remote.



Riferimenti

Sito ufficiale

- [Http://www.Mysql.Com](http://www.Mysql.Com)

Documentazione ufficiale

- [Http://www.Mysql.Com/doc/](http://www.Mysql.Com/doc/)
- [Http://www.Mysql.Com/documentation/mysql/bychapter](http://www.Mysql.Com/documentation/mysql/bychapter)

Manualistica in italiano

<http://www.Linuxvalley.Com/columns/columns.Php?Idcol=114>

<http://www.Linuxvalley.Com/columns/columns.Php?Idcol=115&idpezzo=1>

Tool di gestione

Mascon per gestire in modalità client-server i DB con mysql

- [Http://www.Scibit.Com/products/software/utills/mascon.Htm](http://www.Scibit.Com/products/software/utills/mascon.Htm)

Phpmyadmin per la gestione di un DB mysql via web

- [Http://freshmeat.Net/redirect/phpmyadmin/8001/url_homepage/](http://freshmeat.Net/redirect/phpmyadmin/8001/url_homepage/)

I miti da sfatare sul mysql

- [Http://m.Bacarella.Com/research/mysqlmyths.Html](http://m.Bacarella.Com/research/mysqlmyths.Html)

PHP over MYSQL

- Il PHP (professional home page) è un linguaggio (ambiente) che si interfaccia che sfrutta a pieno la caratteristiche di mysql com RDBMS. Queste sono le sua caratteristiche:
 - Il PHP è un linguaggio di scripting
 - È un linguaggio "html-embedded"
 - Opera server-side, cioè lato server



Connessione ad un DB MySQL

```
<?Php
    $Host = 'localhost';
    $user = 'vostro_user';
    $password = 'vostra_password';
    Mysql_connect ($host,$user,$password) or die
    ("non riesco a connettermi");
    print "connessione eseguita";
?>
```

Creazione e selezione di un DB

```
<?Php
    $host = 'localhost';
    $user = 'vostro_user';
    $password = 'vostra_password';
    Mysql_connect($host,$user,$password) or die ("non
    riesco a connettermi");
    mysql_create_db("prova")or die ("non riesco a creare il
    database");
    mysql_select_db("prova") or die ("non riesco a
    selezionare il database");
    print "connessione, creazione, selezione del database
    eseguita";
```

```
?>
```

Creazione tabella

```
<?Php
    $host = 'localhost';
    $user = 'vostro_user';
    $password = 'vostra_password';
    Mysql_connect($host,$user,$password) or die ("non riesco a
    connettermi");
    print "connessione eseguita";
    Mysql_create_db("prova")or die ("non riesco a creare il database");
    mysql_select_db("prova") or die ("non riesco a selezionare il
    database");
    print "connessione, creazione, selezione del database eseguita";
    $sql= "CREATE TABLE test (
    id INT(10) NOT NULL,
    nome CHAR(100) NOT NULL,
    cognome CHAR(50) NOT NULL)";
    Mysql_query("$sql") or die ("non riesco a creare la tabella");
    print " la tabella test è stata creata con successo!";
```

```
?>
```



Inserimento dati in una tabella

```
<?Php
```

```
    Mysql_query("insert into mail (id,  
    nome,cognome) values  
    ('1','mario','rossi')");
```

```
?>
```

Selezione dei dati di una tabella

```
<?Php
//effettuare prima connessione e
selezione
$Dati = mysql_query("select * from
prova");
$array = mysql_fetch_array($dati);
?>
```