

Laboratorio di Basi di dati

Introduzione a MySQL

Anno accademico 2014-15

Marco Mesiti

MySQL

- MySQL è il database relazionale open source più diffuso al mondo
- MySQL è stabile e veloce
- MySQL comprende le principali caratteristiche di un moderno sistema di gestione dati:
 - Transazioni
 - Vincoli a livello di tupla
 - Chiavi esterne
 - Interrogazioni annidate
- Ultima versione è MySQL 5.5 scaricabile all'indirizzo:
 - www.mysql.org

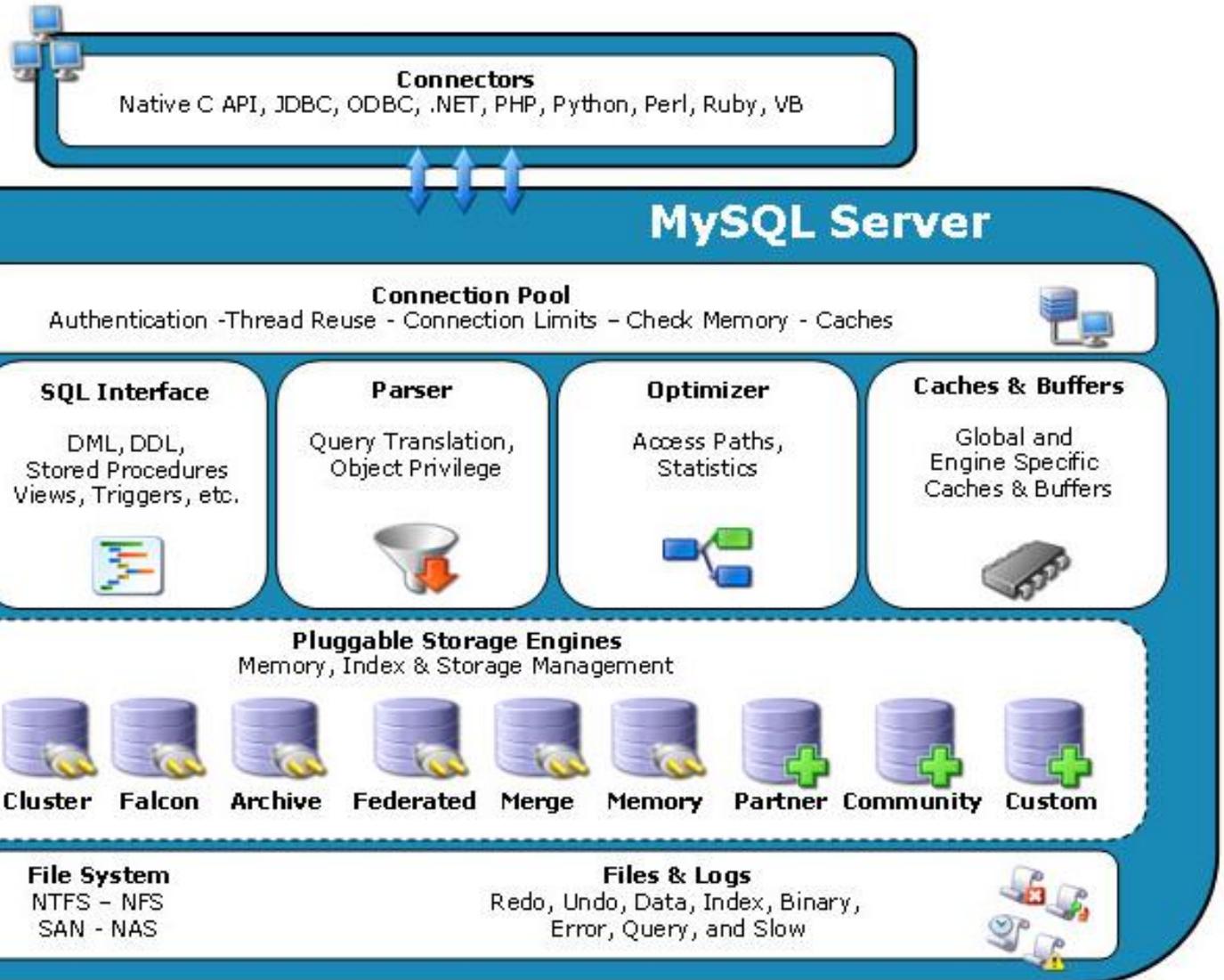


Caratteristiche di MySQL

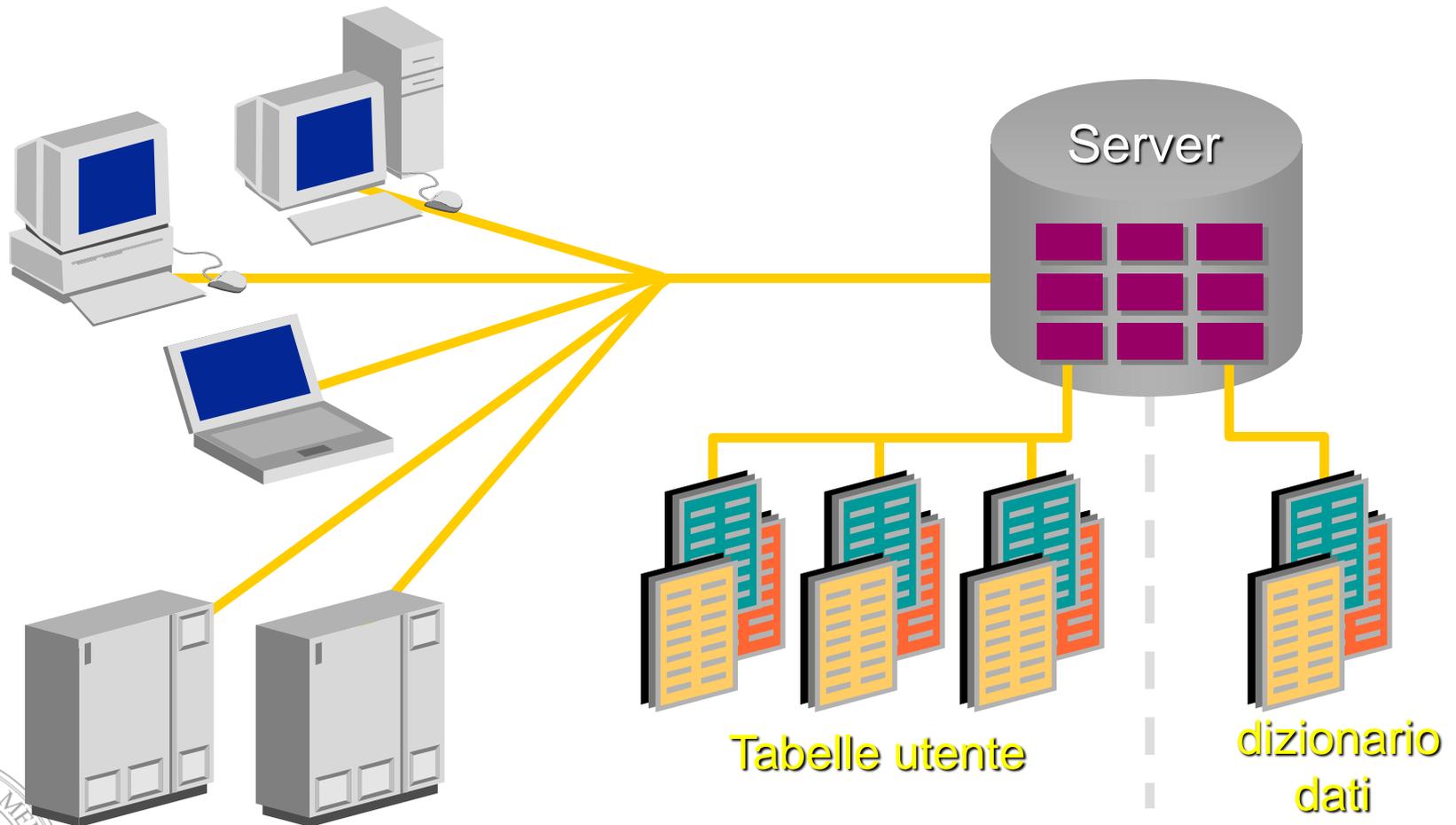
- MySQL NON va confuso con SQL
 - SQL è un linguaggio di interrogazione
 - MySQL sfrutta SQL per interagire con i dati.
 - SQL è un linguaggio, MySQL è un programma
- MySQL NON è un database
 - MySQL è un sistema di gestione DBMS
 - Serve a gestire più database
- MySQL è un prodotto multi-piattaforma
 - Utile sia in ambiente Linux che Windows
- MySQL è open-source (gratis + codice)
- MySQL è cresciuto insieme a PHP (LAMP)
 - Aperto comunque anche ad altri linguaggi



Architettura di MySQL



RDBMS client-server



Tipi di dati: Numerici (1/2)

Tipo	Byte	Minimum Value	Maximum Value
		(Signed/Unsigned)	(Signed/Unsigned)
TINYINT[(M)]	1	-128	127
		0	255
SMALLINT [(M)]	2	-32768	32767
		0	65535
MEDIUMINT [(M)]	3	-8388608	8388607
		0	16777215
INT [(M)]	4	-2147483648	2147483647
		0	4294967295
BIGINT [(M)]	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615
FLOAT[(M,D)]	4	+/-1.175494351E-38	+/- 3.402823466E+38
DOUBLE [(M,D)]	8	+/-2.2250738585072014E-308	+/-1.7976931348623157E+308

Tipi di dati: Numerici (2/2)

- INTEGER
 - DOUBLE PRECISION
 - REAL
 - DECIMAL
 - NUMERIC
-
- Se viene passato il numero 12.345678 ad una colonna di tipo `FLOAT(2,4)`, tale numero diverrà 12.3456 (in quanto si accettano 2 interi ma solo 4 decimali).



Tipi di dati: testo

Tipo	Max size	Byte
CHAR[(M)]	M	M
VARCHAR(M)	M	L+1
TINYBLOB/ TINYTEXT	2^8-1	L+1
BLOB/TEXT	$2^{16}-1$	L+2
MEDIUMBLOB/MEDIUMTEXT	$2^{24}-1$	L+3
LOB/TEXT	$2^{32}-1$	L+4
ENUM('value1','value2',...)	65535 elementi	1 o 2 byte
SET ('value1','value2',...)	64 elementi	1,2,3,4 o 8 byte

- L rappresenta la lunghezza effettiva del testo

Tipi di dati: date e time

Tipo	Range	Byte
DATE	'1000-01-01' '9999-12-31'	3
DATETIME	'1000-01-01 00:00:00' '9999-12-31 23:59:59'	8
TIMESTAMP[(M)]	Da 19700101000000 ad una data nel 2037	4
TIME	'-838:59:59' - '838:59:59'	3
YEAR [(M)]	1901 al 2144 per YEAR(4)	1



Formati di Date e Timestamp

Tipo	Formato
DATETIME	AAAA-MM-GG HH:MM:SS
DATE	AAAA-MM-GG
TIME	HH:MM:SS
YEAR	AAAA
TIMESTAMP	Può avere diversi formati (vedi sotto)

Tipo	Formato
TIMESTAMP(14)	AAAAMMGGHHMMSS
TIMESTAMP(12)	AAMMGGHHMMSS
TIMESTAMP(10)	AAMMGGHHMM
TIMESTAMP(8)	AAAAMMGG
TIMESTAMP(4)	AAMM
TIMESTAMP(2)	AA

Funzione	Formato della risposta
NOW()	AAAA-MM-GG HH:MM:SS
CURDATE()	AAAAA-MM-GG
CURTIME()	HH:MM:SS

Operazioni con le date

Funzione	Formato della risposta
NOW()	AAAA-MM-GG HH:MM:SS
CURDATE()	AAAAA-MM-GG
CURTIME()	HH:MM:SS

Funzione	Utilizzo
DATE_ADD()	DATE_ADD(data, INTERVAL espressione tipo)
DATE_SUB()	DATE_SUBB(data, INTERVAL espressione tipo)
PERIOD_ADD()	PERIOD_ADD(Periodo, Mesi) il Periodo va espresso informato AAAAMM oppure AAMM
PERIOD_SUBB())	PERIOD_SUB(Periodo1, Periodo2) il Periodo va espresso informato AAAAMM oppure AAMM

1- Sommare 10 giorni alla data corrente:

```
SELECT DATE_ADD(CURDATE(), INTERVAL 10 DAYS);
```

2- Sottrarre 2 mesi da una data specifica:

```
SELECT DATE_SUB('2005-01-23', INTERVAL 2 MONTHS);
```

3- Aggiungere 7 mesi al mese di gennaio 2005:

```
SELECT PERIOD_ADD(200501, 7);
```

4- Sottrarre 3 mesi al mese di gennaio 2005:

```
SELECT PERIOD_ADD(200501, -3);
```

5- Trovare la differenza (espressa in mesi) tra il gennaio 2005 ed ottobre 2002:

```
SELECT PERIOD_SUB(200501, 200210);
```

MySQL ed XAMPP

- MySQL è già incluso in XAMPP e non occorre effettuare l'installazione
- XAMPP mette a disposizione **phpMyAdmin**
- E' una interfaccia Web di amministrazione che permette di interagire con il DBMS
 - Creare e cancellare database
 - Creare, modificare e cancellare tabelle
 - Specificare interrogazioni
 - Operare sul livello «fisico» del database
 -
- Esiste anche il programma **MySQL Workbench** scaricabile dal sito di MySQL (non ha un'interfaccia Web, ma è molto comodo)



XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

XAMPP Control Panel v3.2.1

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	12504 12236	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	13636	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

20:40:54 [Apache] Attempting to stop Apache (PID: 6024)
20:40:54 [Apache] Attempting to stop Apache (PID: 6348)
20:40:54 [Apache] Status change detected: stopped
20:40:55 [mysql] Attempting to stop MySQL (PID: 13884)
20:40:55 [mysql] Status change detected: stopped
20:40:57 [Apache] Attempting to start Apache app...
20:40:57 [Apache] Status change detected: running
20:40:58 [mysql] Attempting to st...
20:40:58 [mysql] Status change d...

localhost / 127.0.0.1 | php | x

localhost/phpmyadmin/index.php?db=&table=&server=1&target=&lang=it&collation_connection=utf8_general_ci&tok

phpMyAdmin

(Tabelle recenti) ...

- cdcol
- information_schema
- mysql
- nett
- performance_schema
- phpmyadmin
- ps270a2014
- test
- webauth

Impostazioni Generali

Collation della connessione del server :
Codifica caratteri

Impostazioni di Presentazione

Lingua - Language : Italiano - Italian
Tema: pmahomme
Dimensione font: 82%
Ulteriori impostazioni

Server del Database

- Server: 127.0.0.1 via TCP/IP
- Tipo di server: MySQL
- Versione del server: 5.5.32 - MySQL Community Server (GPL)
- Versione protocollo: 10
- Utente: root@localhost
- Codifica caratteri del server: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.4 (Win32) OpenSSL/0.9.8y PHP/5.4.16
- Versione del client del database: libmysql - mysqlnd 5.0.10 - 20111026 - \$Id: e707c415dh32080h3752h232487a435ee037



Cosa vediamo...

- Ci occupiamo del livello logico del DBMS
 - Studiamo come inserire, creare, modificare, popolare e interrogare database e tabelle
 - Facciamo vedere le istruzioni che possono essere eseguite da SQL (si può usare l'interfaccia per eseguire gli stessi comandi)
 - Nelle ultime lezioni affronteremo alcuni aspetti che riguardano il modello fisico del database
 - Come vengono memorizzate le tabelle
 - Come sviluppare strutture ausiliarie di accesso
- Vedremo anche la gestione degli utenti



Database «corsiInglese»

- Creiamo in questo database le tabelle:
Livello(nome, libroTesto, n_max_allievi, esameFinale)
Insegnante(matricola, nome_U, cognome_U, dataNascita,
nazionalita, via, citta, cap)
Corso(livello^{LIVELLO}, numero, dataAttivazione,
insegnante^{INSEGNANTE})
Allievo(matricola, nome)
Iscrizione(livello^{CORSO}, numero^{CORSO}, allievo^{ALLIEVO},
dataIscrizione, n_assenze, costo)
- Le altre tabelle le potete creare a casa per esercizio



Creazione di una base di dati

- Per creare un database si usano le istruzioni:
 - `create database nome_database;`
- Nel nostro esempio
 - `Create database corsiInglese;`
- Si può controllare che il comando abbia funzionato digitando:
 - `show databases;`
- Nella lista delle basi di dati presenti nel sistema dovrete vedere anche il database `corsiInglese`
- Per rimuovere un database si usano le istruzioni:
 - `drop database nome_database;`



Selezione di una base di dati

- Prima di creare le tabelle, o fare qualsiasi operazione sul database è opportuno segnalare a MySQL che si intende utilizzare la base di dati creata. E' necessario digitare:
 - `use nome_database;`
- Nel nostro esempio digitare il comando:
 - `use corsiInglese;`
- In questo modo la base di dati risulta selezionata e tutto quello che si fa da questo momento andrà a cambiare la base di dati selezionata



Creazione tabella 'Livello'

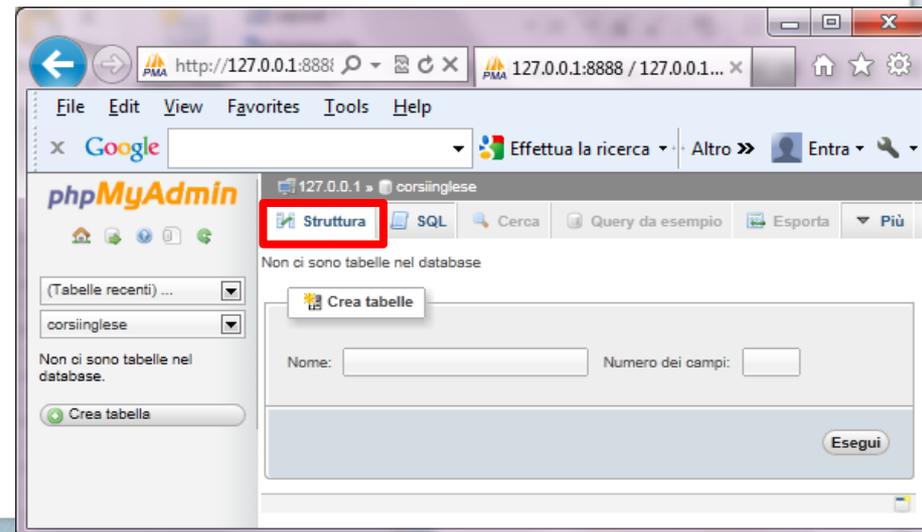
campo	tipo	vincolo
nome	VARCHAR(20)	PRIMARY KEY
libroTesto	VARCHAR(50)	NOT NULL
n_max_allievi	INT	DEFAULT 10
esameFinale	BOOLEAN	NOT NULL (si,no)

Usare SQL

Usare interfaccia

(stessa sintassi vista a lezione)

```
CREATE TABLE nomeTabella(  
  Campo1 tipo1 vincolo1,  
  
  CampoN tipoN vincoloN,  
)
```



Specifica della tabella 'Livello'

The screenshot shows the phpMyAdmin interface for creating a table named 'livello'. The table structure is defined as follows:

Nome	Tipo	Lunghezza/Valori	Predefinito	Codifica caratteri	Attributi	Null	Indice	A_I	Commenti
nome	VARCHAR	20	Nessuno			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>	
libroTesto	VARCHAR	50	Nessuno			<input type="checkbox"/>	---	<input type="checkbox"/>	
n_max_allievi	INT	3	Come definito: 10			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>	
esameFinale	BOOLEAN		Come definito: true			<input type="checkbox"/>	---	<input type="checkbox"/>	

```
CREATE TABLE livello(  
  nome varchar(20) PRIMARY KEY,  
  libroTesto varchar(50) NOT NULL,  
  n_max_allievi int(3) DEFAULT '10',  
  esameFinale tinyint(1) NOT NULL DEFAULT '1'  
);
```

Comandi per tabelle

- Per vedere le tabelle presenti nel database
 - `show tables;`
- Per vedere le caratteristiche di una tabella
 - `describe nome_tabella;`
- Per cancellare una tabella
 - `drop table nome_tabella;`



Tipo Boolean

- Il tipo boolean, viene implementato con un intero in MySQL
- Si può utilizzare il tipo ENUM per indicare i valori che può assumere
 - `esameFinale enum('1','0') NOT NULL DEFAULT '1'`

The screenshot shows the phpMyAdmin interface in a browser window. The main content area displays the 'Modifica' (Modify) window for the 'esameFinale' table. The table structure is shown as follows:

Nome	Tipo	Lunghezza/Valori	Predefinito	Codifica caratteri	Attributi
esameFinale	ENUM	'1','0'	Come definito: 1	utf8_bin	

Below the table, there is a warning: "Dati in ENUM o SET troppo lunghi? Più spazio per la modifica". A "Salva" (Save) button is visible at the bottom right of the modification window. The left sidebar shows the database structure for 'corsinglese' with tables: 'nome', 'libroTesto', 'n_max_allievi', and 'esameFinale'.

Inserire i seguenti dati

Nome	libroTesto	n_max_allievi	esameFinale
base	English base	10	1
medium	English plus	5	0
advance	English extra	25	1

Usare SQL

Usare interfaccia

(stessa sintassi vista a lezione)

```
Insert into nomeTabella  
values(valCol1,..., calColN)
```

The screenshot shows the phpMyAdmin web interface. The browser address bar displays '127.0.0.1:8888 / 127.0.0.1 / c...'. The interface includes a navigation menu with options like 'Mostra', 'Struttura', 'SQL', 'Cerca', 'Inserisci', 'Esporta', 'Importa', 'Operazioni', and 'Più'. The 'Inserisci' button is highlighted with a red rectangle. Below the menu, a table structure is visible with columns: 'nome' (varchar(20)), 'libroTesto' (varchar(50)), 'n_max_allievi' (int(3)), and 'esameFinale' (enum). The 'nome' field contains 'base', 'libroTesto' contains 'English base', and 'n_max_allievi' contains '10'. The 'esameFinale' field has radio buttons for '1' and '0', with '1' selected. An 'Esegui' button is at the bottom of the form.



Proviamo a modificare i tipi della tabella livello

- Cambiare il tipo di nome
 - `varchar(20)` → `varchar(30)`
- Il sistema protesta? L'operazione va a buon fine?
- Cambiare il tipo di libroTesto
 - `varchar(50)` → `varchar(5)`
- Il sistema protesta? L'operazione va a buon fine?

NOTA SUGLI INSERIMENTI CON IL COMANDO INSERT

I valori di tipo stringa e di tipo data vanno inseriti tra apici, i valori numerici non vanno inclusi tra apici



Inserimenti, cancellazioni e modifiche

- Per verificare che il sistema abbia inserito in modo corretto i dati, digitare il comando:
 - `SELECT * FROM nome_tabella;`
- Si può rimuovere uno o più tuple da una tabella
 - `DELETE FROM nome_tabella [WHERE cond];`
 - `DELETE FROM livello WHERE esameFinale='1';`
- Per modificare i dati di una tabella si può usare il comando di update:
 - `UPDATE nome_tabella
SET nome_col1=espressione1 [,
nome_col2=espressione2,...]
[WHERE condizione];`
 - **Esempio:** `UPDATE Livello SET LibroTesto='';`
 - Cosa è successo dopo l'esecuzione di questo comando?

Creazione e popolamento tabella 'Insegnante'

campo	tipo	vincolo
matricola	VARCHAR(5)	PRIMARY KEY
nome	VARCHAR(20)	UNIQUE
cognome	VARCHAR(20)	UNIQUE
dataNascita	date	NOT NULL
nazionalita	VARCHAR(20)	NOT NULL
via	VARCHAR(20)	NOT NULL
citta	VARCHAR(20)	NOT NULL
cap	VARCHAR(5)	NOT NULL

matricola	nome	cognome	dataNascita	nazionalita	via	citta	cap
1	Jane	Brown	16/4/1979	UK	Via Platini	Milano	20100
2	Mark	Sloan	18/12/1972	Ireland	Via C. Fo'	Rapallo	16035

Creazione e popolamento tabella 'Corso'

campo	tipo	vincolo
livello	???	PRIMARY KEY
numero	INT(5)	
dataAttivazione	date	NOT NULL
insegnante	???	NOT NULL

Come si fa a determinare il tipo dei campi con tipo ???

livello	numero	dataAttivazione	insegnante
base	1	1/1/2012	???
medium	1	1/3/2012	???
base	2	5/7/2012	???

L'insegnante dei corsi base è Jane, mentre per gli altri è Mark



Creazione e popolamento tabella 'Allievo'

campo	tipo	vincolo
matricola	VARCHAR(5)	PRIMARY KEY
nome	VARCHAR(20)	UNIQUE

livello	nome
1	Rossi Antonio
2	Della Ca' Antonio Maria
3	De Benedetti Maria
4	Marcovaldo Ernesto
5	Ferrante Marina



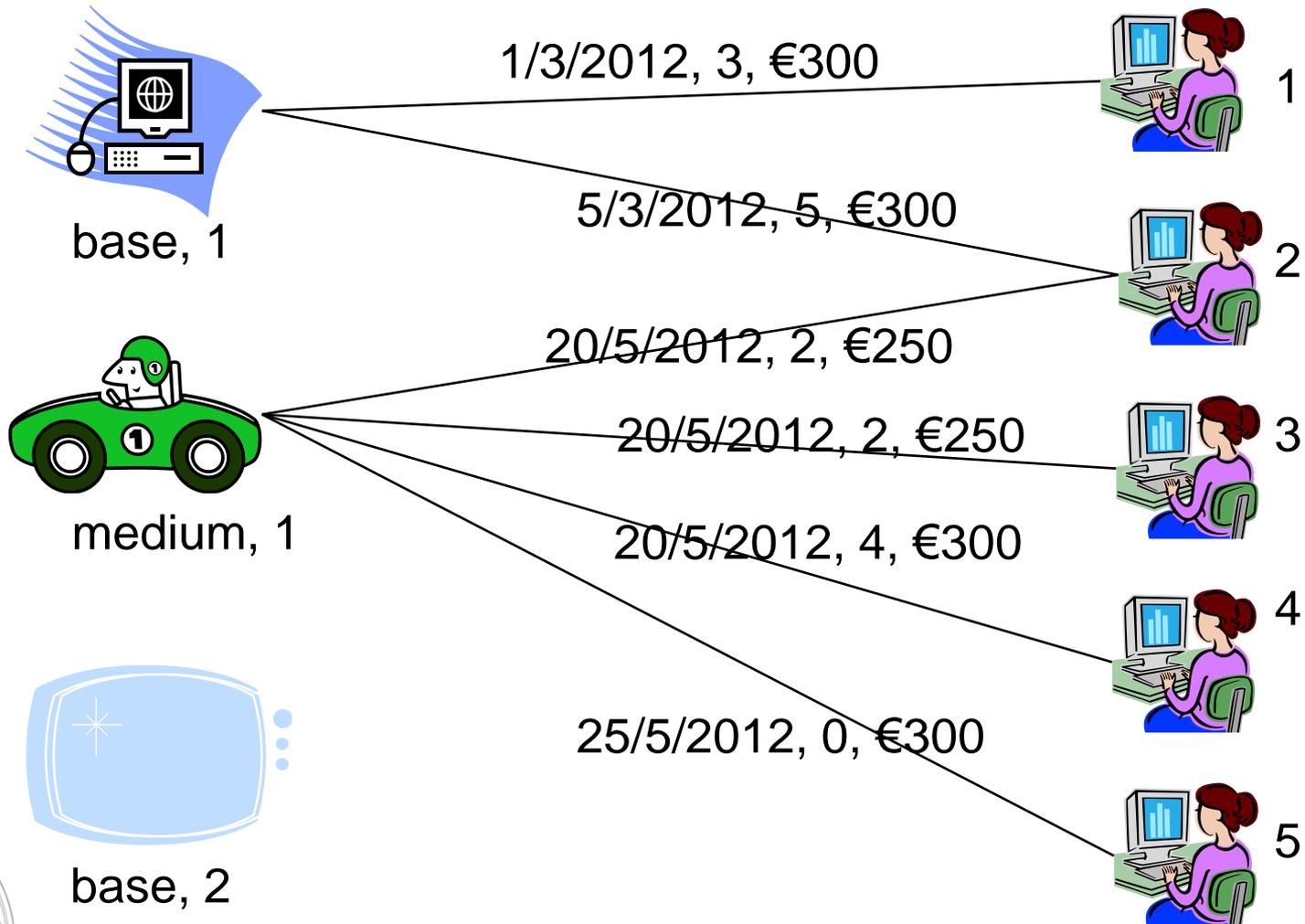
Creazione e popolamento tabella 'Iscrizione'

campo	tipo	vincolo
livello	???	PRIMARY KEY
numero	???	
allievo	???	
dataIscrizione	date	NOT NULL
n_assenze	INT	NOT NULL
costo	DECIMAL(8,2)	NOT NULL

Come si fa a determinare il tipo dei campi con tipo ???



Popolamento



Script completo

- Nel file **corsInglese.sql** è riportato lo script completo per la creazione della base di dati.
- Si noti che attraverso questo script sono stati imposti i vincoli di identificazione e di riferimento imposti dal diagramma ER iniziale
- Sono stati imposti anche i seguenti vincoli (della tabella dei vincoli riportati nella seguente pagina):
 - V4
 - V7: `Recapito.tipo ∈ {'casa','cell','skype'}`
 - V8
 - V12
- Gli altri vincoli non sono ancora specificati. Vedremo più avanti come specificare tali vincoli



Vincoli sul modello relazionale

V1: Un insegnante non può avere assegnato più corsi che si tengono negli stessi orari

V2: Il numero di allievi di un corso deve essere minore del numero massimo di allievi previsti per i corsi di quel livello

V4: il giorno di un corso deve essere un valore tra LUN,MAR,MER,GIO,VEN

V5: se un allievo prende lezioni private da un certo insegnante, allora quell'insegnante deve tenere almeno un corso tra quelli a cui l'allievo è iscritto

V6: Il giorno di attivazione di un corso deve precedere le date di iscrizione degli allievi per quel corso

V7: costo, n_max_allievi ≥ 1 ; n_assenze ≥ 0 , Recapito.tipo $\in \{\text{casa, cell, skype}\}$

V8: tipo in AttivitaCulturali può essere nullo o assumere un valore in {'proiezione', 'conferenza'}

V9: se AttivitaCulturali .tipo=NULL gli attributi opzionali in AttivitaCulturali sono NULL

V10: se AttivitaCulturali . tipo='proiezione' gli attributi regista e film in AttivitaCulturali sono NOT NULL e gli attributi relatore e argomento sono NULL

V11: se AttivitaCulturali . tipo='conferenza' gli attributi regista e film in AttivitaCulturali sono NULL e gli attributi relatore e argomento sono NOT NULL

V12: Vincoli di unique su nome e cognome in Insegnante, allievo, data e ora in LezionePrivata

V13: Non possono esistere allievi senza almeno un recapito telefonico e che non siano iscritti ad almeno un corso, Ogni corso si deve svolgere in almeno un giorno della settimana

Alcuni aspetti del modello fisico

- MySQL supporta diverse strategie di memorizzazione delle tabelle a livello fisico
 - **MyISAM**: tipo di tabelle standard, supportano la compressione ma non gestiscono le transazioni
 - **InnoDB**: sono transaction-safe (proprietà ACIDe), supportano le transazioni e i lock di riga
 - **BDB** (Berkeley DB): motore di memorizzazione che supporta transazioni e lock a livello di pagina
 - **MEMORY**: tabelle che vengono salvate interamente in memoria locale e mai su disco, sono molto veloci ma hanno dimensione limitata e non possono essere recuperate in caso di guasto del sistema
- Di default viene utilizzato **InnoDB**. Verificate sempre che questa sia l'opzione di memorizzazione delle tabelle



Chiavi esterne

- Quanto visto finora a lezione è il modo standard (SQL:2003) per la creazione delle tabelle e stabilire i vincoli di identificazione e di riferimento
- Bisogna però precisare che solo nelle tabelle di tipo **InnoDB** è possibile specificare i vincoli di chiave esterna
 - Le colonne riferite devono essere indicate
 - Occorre specificare un indice sulle colonne «chiave esterna»
 - Non si possono effettuare specifiche «in-line»
- **Esempio:**

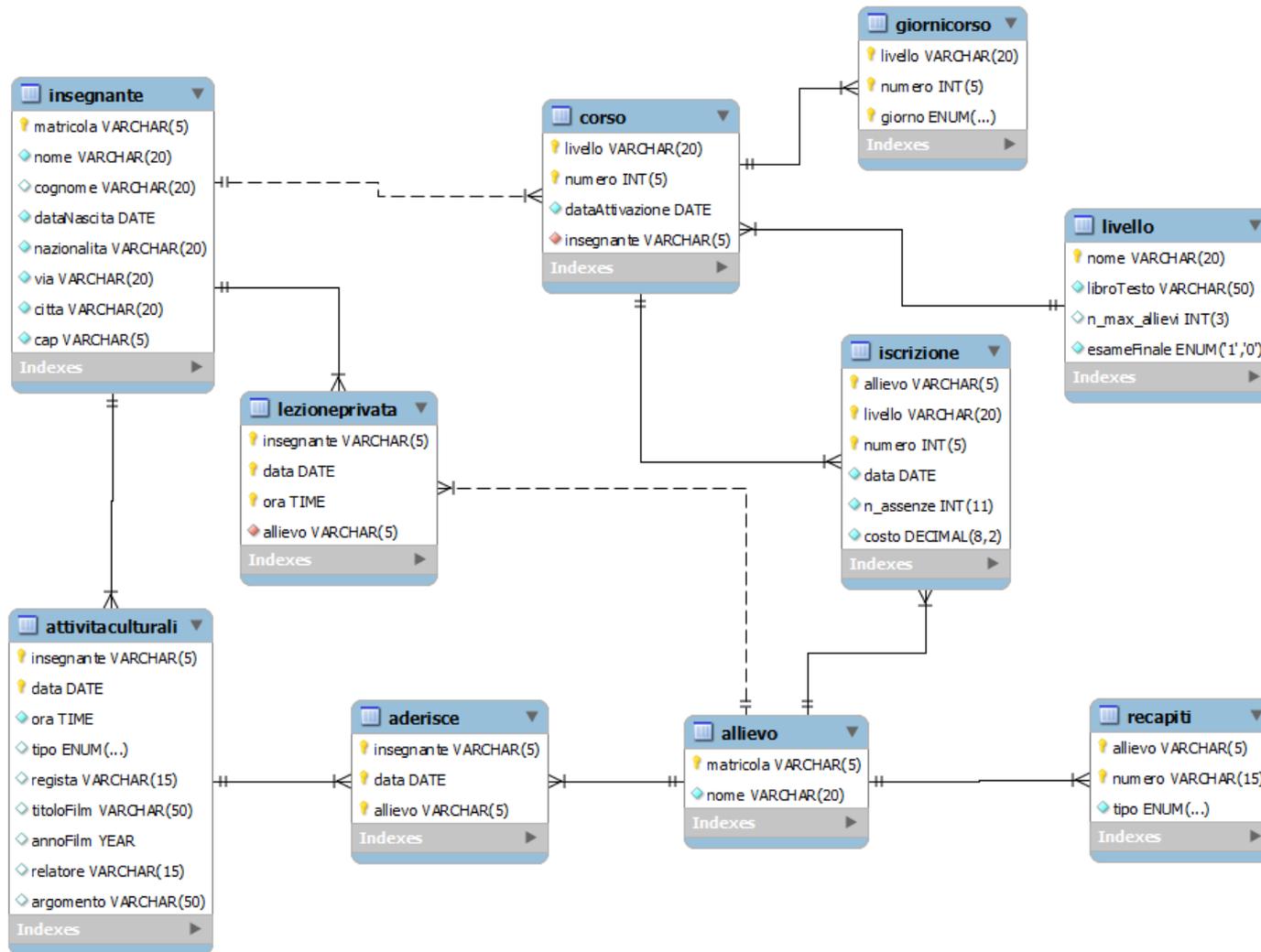
```
CREATE TABLE parent (id INT NOT NULL,  
                      PRIMARY KEY (id));  
  
CREATE TABLE child (id INT, parent_id INT,  
                    INDEX (parent_id),  
                    FOREIGN KEY (parent_id) REFERENCES parent(id)  
                      ON DELETE CASCADE);
```

Chiave esterne e vincoli CHECK

- A lezione avete visto anche diversi tipi di MATCH sulla chiave esterna
 - MySQL non li supporta. Essenzialmente supporta la semantica di MATCH SIMPLE
- Lo script completo è in **corsIngleseMySQL.sql**
- Lo standard prevede vincoli di **CHECK** a livello di tupla
 - CHECK (dataAssunzione > '2012-01-01')
 - MySQL non li supporta (potete specificarli a livello di sintassi, ma vengono ignorati)
 - Nelle ultime lezioni vedremo come specificare questi vincoli



Schema relazionale



Schema **relazionale**

- Lo schema precedente è stato realizzato con uno strumento di MySQL Workbench
- Mette in evidenza la struttura dello schema relazionale, i vincoli di identificazione e i vincoli di chiave esterna
- Esiste uno strumento analogo anche in phpMyadmin
- Fate attenzione! Questo diagramma **non** corrisponde al diagramma ER!! Sono due formalismi diversi che si basano su modelli distinti e non vanno confusi



Conclusioni

- In questa lezione abbiamo visto il funzionamento di uno specifico DBMS: MySQL
- Questo strumento è molto utilizzato, ma non è pienamente conforme allo standard SQL
 - Non sono supportate tutte le funzionalità
 - Alcune volte viene supportata la sintassi, ma poi manca l'implementazione
- E' importante tenere sempre a mente lo standard e poi verificare quali sono le «semplificazioni» adottate nello specifico DBMS
- Strumenti come Oracle, DB2, postGRES aderiscono di più allo standard

