

Introduzione a MySQL

Dott. Salvatore Alaimo

Studio 32 – Blocco 2

Email: alaimos@dmi.unict.it

- Introduzione a MySQL
 - Engine
 - Tipi di Dati
 - Installazione di MySQL
- Programmi Client
- Account e Privilegi
- Comandi MySQL
 - Alcuni esempi pratici
- Funzioni e Operatori
 - Altri esempi pratici

Parte I

Introduzione

- Creato dalla società MySQLAB sin dal 1979 soltanto dal 1996 supporta anche SQL.
- Sun Microsystems nel 2008 rileva la società per 1 miliardo di dollari



- Nel 2010 Oracle acquista Sun per 7,5 miliardi di dollari possedendo così anche MySQL.

ORACLE®

Cosa è MySQL

- MySQL, definito Oracle MySQL, è un Relational database management system (RDBMS), composto da un client con interfaccia a riga di comando e un server, entrambi disponibili sia per sistemi Unix o Unix-like come GNU/Linux che per Windows, anche se prevale un suo utilizzo in ambito Unix.
- Oggi l'ultima versione disponibile è la 5.6
- Dal 1996 supporta la maggior parte della sintassi SQL e si prevede in futuro il pieno rispetto dello standard ANSI. Possiede delle interfacce per diversi linguaggi, compreso un driver ODBC, due driver Java, un driver per Mono e .NET ed una libreria per python.

- MySQL mette a disposizione diversi **tipi di tabelle**,(ovvero "**storage engine**") per la memorizzazione dei dati.
- Ognuno presenta proprietà e caratteristiche differenti.
- Esiste una **API** che si può utilizzare per creare nuovi tipi di tabella che si possono installare senza necessità di riavviare il server
- Due sono i sistemi principali:
 - **Transazionali**: sono più sicuri, permettono di recuperare i dati anche in caso di crash, e consentono di effettuare modifiche tutte insieme;
 - **Non transazionali**: sono più veloci, occupano meno spazio su disco e minor richiesta di memoria.

Engine: MyIsam

- **MyISAM** era lo storage engine di default dal MySQL 3.23 fino al MySQL 5.4.
- **MyISAM** utilizza la struttura **ISAM** e deriva da un tipo più vecchio, oggi non più utilizzato, che si chiamava appunto **ISAM**.
- È un motore di immagazzinamento dei dati estremamente veloce e richiede poche risorse, sia in termini di memoria RAM, sia in termini di spazio su disco.
- Il suo limite principale rispetto ad alcuni altri SE consiste nel mancato supporto delle transazioni e alle foreign key.
- Ogni tabella MyISAM è memorizzata all'interno del disco con tre file:
 - un file **.frm** che contiene la definizione della tabella,
 - un file **.MYD** per i dati
 - un file **.MYI** per gli indici (149% dei dati indicizzati)

Engine: InnoDB

- **InnoDB** è un motore per il salvataggio di dati per MySQL, fornito in tutte le sue distribuzioni (Default dalla versione 5.5).
- La sua caratteristica principale è quella di supportare le transazioni di tipo **ACID (Atomicity, Consistency, Isolation, e Durability)**.
- Cosa differenzia InnoDB da MyISAM?
 - Minore tempo necessario per riparare le tabelle in caso di crash. (InnoDB: tempo costante; MyISAM: dipendente dalle dimensioni della tabella);
 - InnoDB ha un sistema interno per la gestione della cache, mentre MyISAM si affida all'OS;
 - MyISAM generalmente immagazzina i record di una tabella nell'ordine in cui sono state create, mentre InnoDB le immagazzina nell'ordine seguito dalla chiave primaria. Quando viene utilizzata la chiave per la lettura di una riga, l'operazione avviene più rapidamente;
 - InnoDB comprime i record molto meno rispetto a MyISAM nonostante ciò InnoDB richiede il 20% di spazio in meno;
 - In InnoDB, le ricerche fulltext sono supportate solo dalla versione 5.6 di MySQL.

- **InnoDB** mette a disposizione le seguenti funzionalità:
 - transazioni SQL con savepoint e transazioni XA;
 - lock a livello di record;
 - foreign key;
 - integrità referenziale;
 - colonne AUTOINCREMENT;
 - tablespace.
- InnoDB offre delle ottime performance in termini di prestazioni e utilizzo della CPU specialmente quando si ha a che fare con una grande quantità di dati.
- InnoDB può interagire tranquillamente con tutti gli altri tipi di tabelle in MySQL.

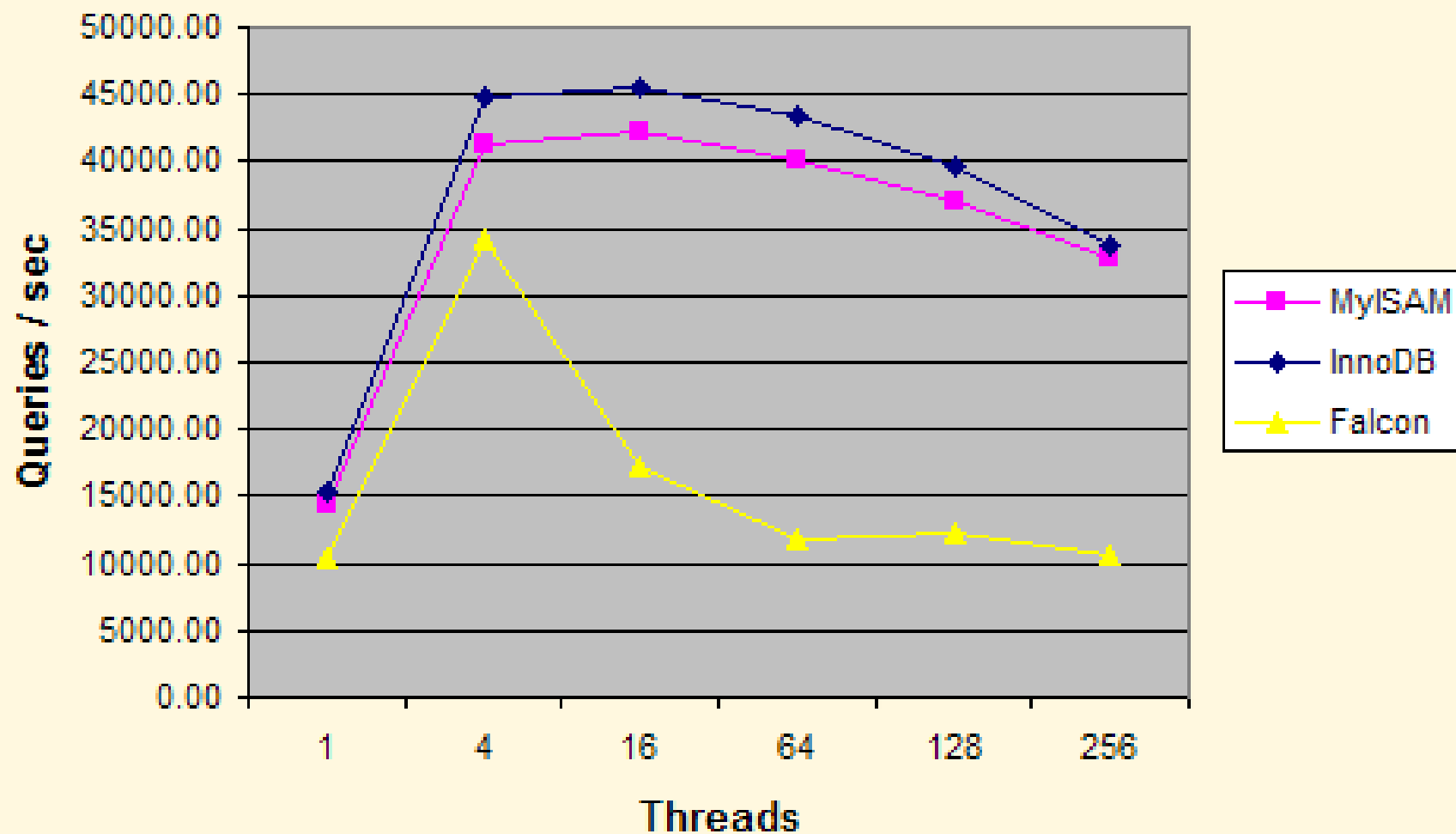
- Le tabelle InnoDB sono soggette alle seguenti limitazioni:
 - Non è possibile creare più di **1000 colonne** per tabella;
 - Su alcuni sistemi operativi le **dimensioni del tablespace** non possono superare i **2 Gb**;
 - La grandezza di tutti i **file di log** di InnoDB deve essere inferiore ai **4 Gb**;
 - La grandezza minima di un tablespace è di **10 MB**;
 - Non possono essere creati indici di tipo **FULLTEXT** con **MySQL 5.5 o precedente**;
 - Le **SELECT COUNT(*)** su tabelle di grandi dimensioni possono essere molto lente.

Engine: Altri

Storage Engine	Descrizione
BDB	Tabelle transaction-safe con page locking
HEAP	I dati della seguente tabella sono solamente memorizzati nella memoria principale
ISAM	Primo SE di MySQL
InnoDB	Tabella transaction-safe con lock di righe e chiavi esterne
MEMORY	Un alias di HEAP
MERGE	Una collezione di tabelle MyISAM usate come se fossero una singola tabella
MRG_MyISAM	Alias di MERGE
MyISAM	SE binario portabile, evoluzione di ISAM.

Engine: MyISAM vs InnoDB

READ_PK_POINT



Tipi di Dati: Numerici

Tipo	Byte	Min. Value (Signed/Unsigned)	Max. Value (Signed/Unsigned)
TINYINT	1	-128/0	127/255
SMALLINT	2	-32.768/0	32.767/65535
MEDIUMINT	3	-8.388.608/0	8.388.607/16.777.215
INT	4	-2.147.483.648/0	2.147.483.647/4.294.967.295
BIGINT	8	-9.223.372.036.854.775.808/0	9.223.372.036.854.775.807/18.446.744.073.709.551.615
FLOAT	4	+/-1.175494351E-38	+/-3.402823466E+38
DOUBLE	8	+/-2.225073858507201E-308	+/-1.7976931348623157E+308

Tipi di Dati: Numerici

Tipo	Byte	Min. Value (Signed/Unsigned)	Max. Value (Signed/Unsigned)
INTEGER	Equivale ad INT		
DOUBLE PRECISION	Equivale a DOUBLE		
REAL	Equivale a DOUBLE		
DECIMAL(M[,D])	M+2	Tutti i numeri di M cifre di cui D decimali.	
NUMERIC(M[,D])	Equivale a DECIMAL		
BIT(M)	M	una sequenza di M bit (MySQL 5.5)	

Tipi di Dati: Date e Tempi

Tipo	Byte	Range
DATE	3	dal '01/01/1000' al '31/12/9999'
DATETIME	8	dal '01/01/1000 00:00:00' al '31/12/9999 23:59:59'
TIMESTAMP[(M)]	4	dal '01/01/1970' al '31/12/2037'
TIME	3	da '-838:59:59' a '838:59:59'
YEAR[(M)]	1	per YEAR(4) dal '1901' al '2144'

Tipi di Dati: Testo e Altro

Tipo	Byte	Max Length
CHAR[(M)]/BINARY[(M)]	M	M
VARCHAR(M)/VARBINARY[(M)]	L+1	M
TINYBLOB/ TINYTEXT	L+1	255
BLOB/TEXT	L+2	65.535
MEDIUMBLOB/MEDIUMTEXT	L+3	16.777.215
LOB/TEXT	L+4	4.294.967.295
ENUM('value1','value2',...)	1 o 2 byte	65535 elementi
SET ('value1','value2',...)	1,2,3,4 o 8 byte	64 elementi

L è la lunghezza effettiva del testo memorizzato.

- Alcuni link per il download di software e tools:

- <http://dev.mysql.com/downloads/mysql/>
- <http://dev.mysql.com/downloads/tools/>
- <http://dev.mysql.com/downloads/installer/>
- <http://dev.mysql.com/downloads/connector/>
- <http://dev.mysql.com/downloads/mysql-proxy/>

Parte III

Client

PHPMyAdmin

The screenshot shows the PHPMyAdmin interface in a web browser. The browser's address bar displays `alpha.dmi.unict.it/phpmyadmin/index.php`. The interface includes a top navigation bar with tabs for Database, SQL, Stato, Processi, Privilegi, Esporta, Importa, Variabili, Set di caratteri, Motori, and Più. A left sidebar lists various databases and users, including `alaimos`, `dtweb`, `eshmun`, `mysql`, `performance_schema`, `phpmyadmin`, `rn209`, `sdibella`, `test`, and `vbonnici`.

The main content area is divided into three primary sections:

- Impostazioni Generali**: Contains options to change the password and select the MySQL connection collation, currently set to `utf8_general_ci`.
- Impostazioni di Presentazione**: Allows setting the language to `Italiano - Italian`, the theme to `pmahomme`, and the font size to `82%`. It also includes a link for further settings.
- MySQL**: Displays server information such as `Server: Localhost via UNIX socket`, `Versione del server: 5.5.35-0ubuntu0.12.04.2`, `Versione protocollo: 10`, `Utente: root@localhost`, and `Set di caratteri MySQL: UTF-8 Unicode (utf8)`.

Additional sections on the right include:

- Web server**: Lists `Apache/2.2.22 (Ubuntu)`, `Versione MySQL client: 5.5.35`, and `Estensioni PHP: mysqli`.
- phpMyAdmin**: Provides links for version information (`3.4.10.1deb1`), documentation, Wiki, the official homepage, contributing, receiving help, and a list of changes.

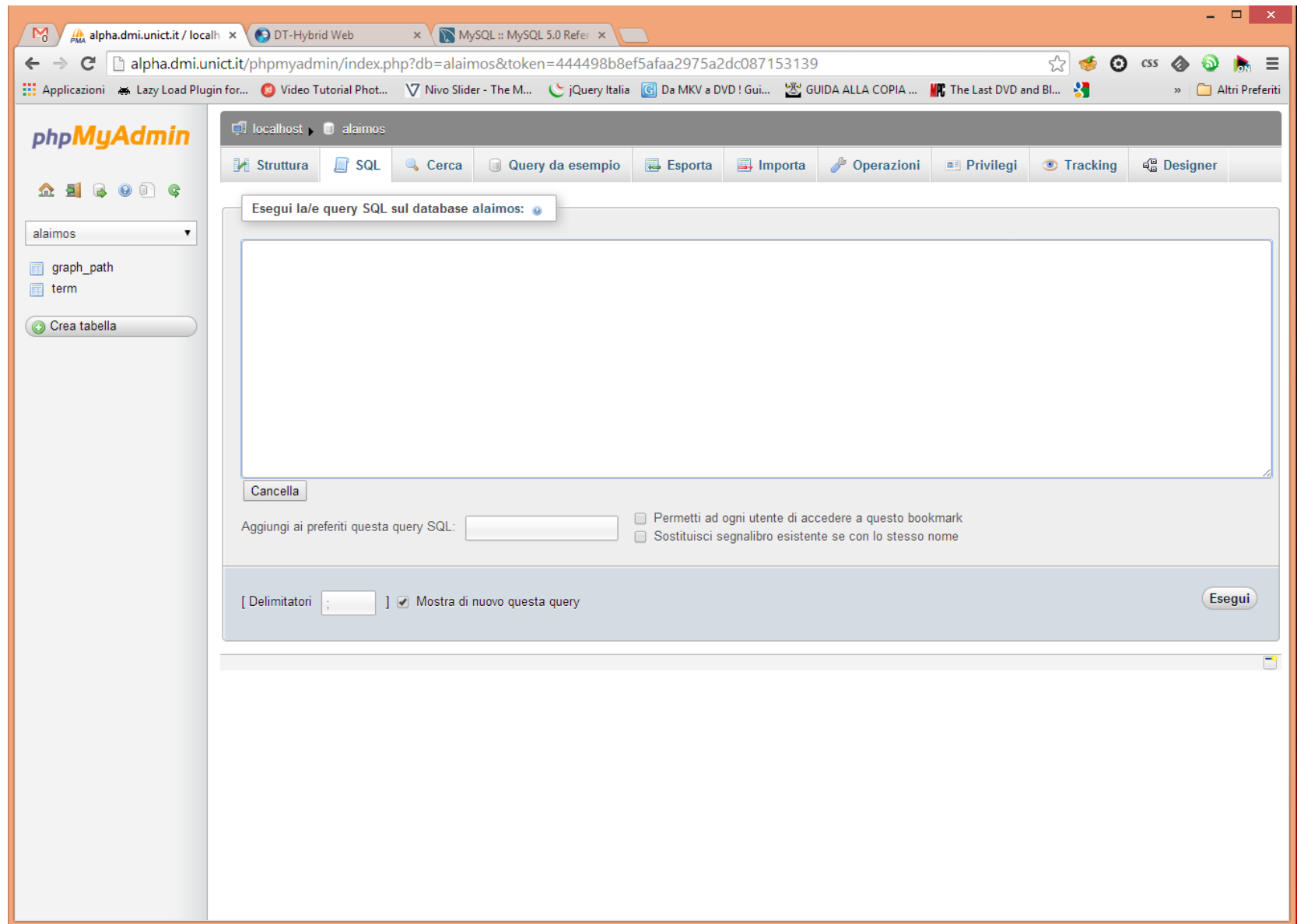
PHPMyAdmin

The screenshot shows the PHPMyAdmin interface in a web browser. The browser's address bar displays the URL: `alpha.dmi.unict.it/phpmyadmin/index.php?db=alaimos&token=444498b8ef5afaa2975a2dc087153139`. The interface includes a top navigation bar with tabs for 'Struttura', 'SQL', 'Cerca', 'Query da esempio', 'Esporta', 'Importa', 'Operazioni', 'Privilegi', 'Tracking', and 'Designer'. Below this, a table lists the tables in the 'alaimos' database:

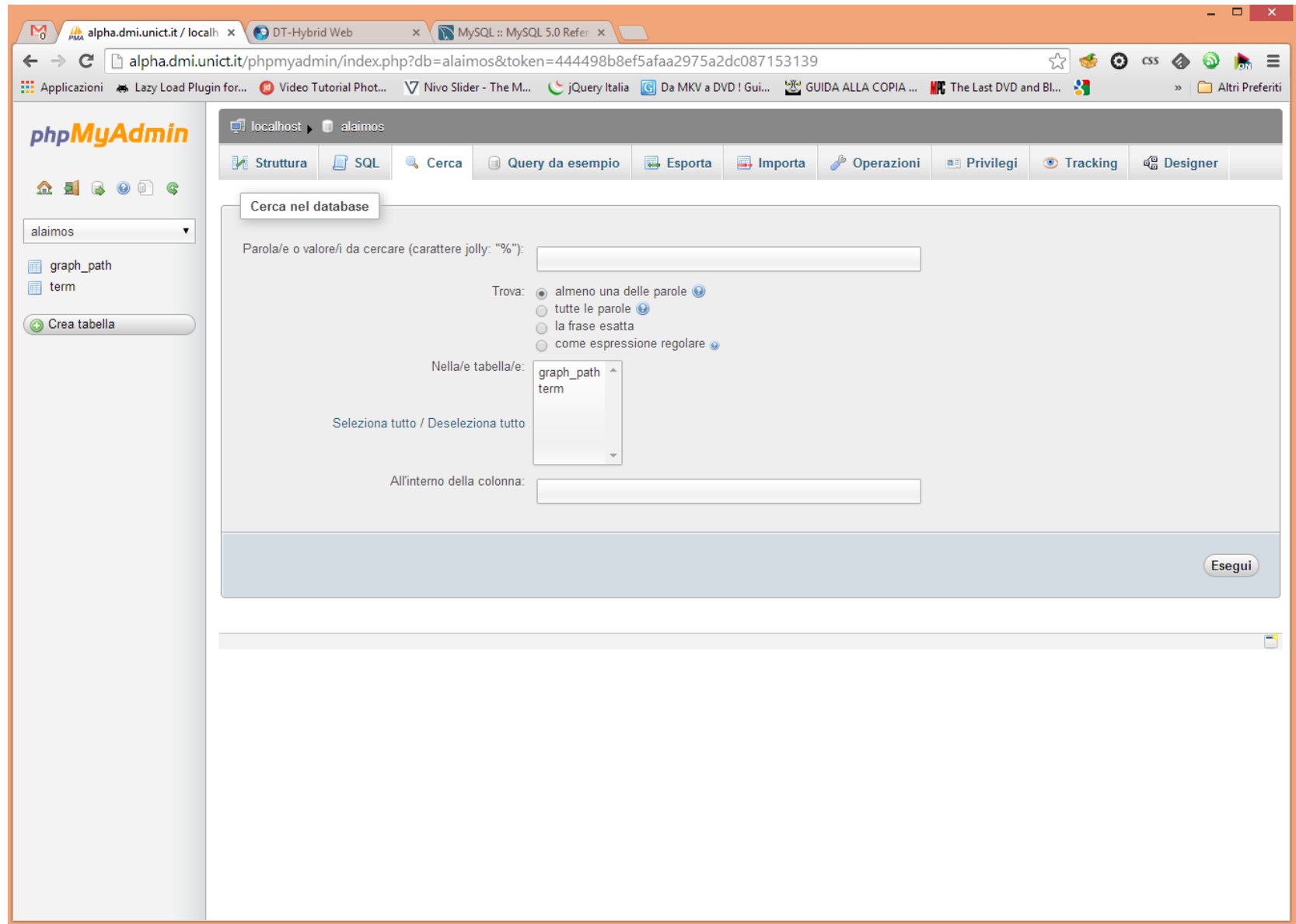
Tabella	Azione	Righe	Tipo	Collation	Dimensione	Overhead
<input type="checkbox"/> graph_path	Mostra Struttura Cerca Inserisci Svuta Elimina	1,043,029	MyISAM	latin1_swedish_ci	186.9 MiB	-
<input type="checkbox"/> term	Mostra Struttura Cerca Inserisci Svuta Elimina	40,248	MyISAM	latin1_swedish_ci	14.0 MiB	-
2 tabelle	Totali	1,083,277	InnoDB	latin1_swedish_ci	200.9 MiB	0 B

Below the table, there are links for 'Seleziona tutti / Deseleziona tutti' and 'Se selezionati:'. There are also links for 'Visualizza per stampa' and 'Data Dictionary'. A button labeled 'Crea una nuova tabella nel database alaimos' is visible. Below this button, there are input fields for 'Nome:' and 'Numero di colonne:'. An 'Esegui' button is located at the bottom right of the form.

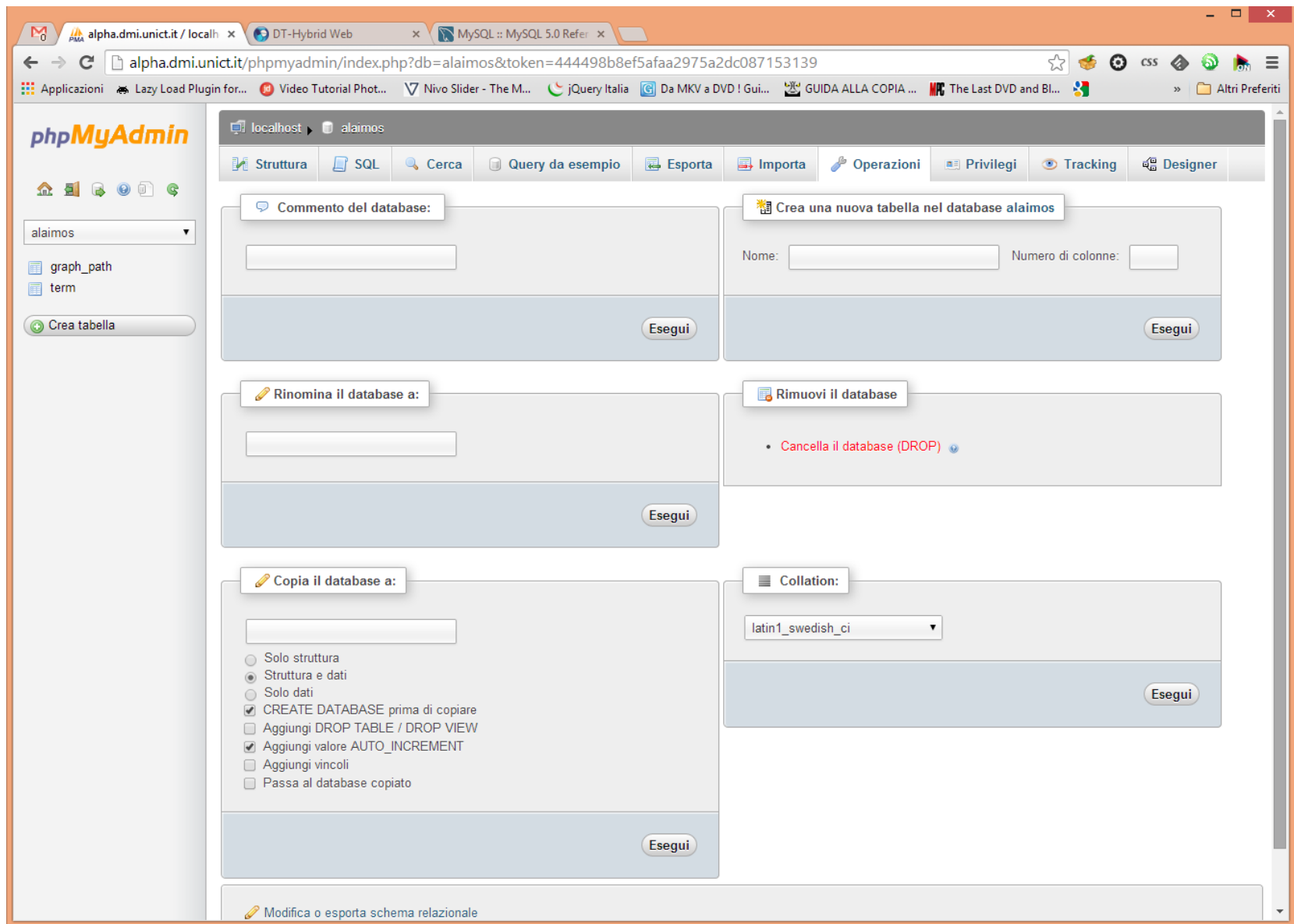
PHPMyAdmin



PHPMyAdmin



PHPMyAdmin



PHPMyAdmin

The screenshot shows the PHPMyAdmin interface in a web browser. The browser's address bar displays the URL: `alpha.dmi.unict.it/phpmyadmin/index.php?db=alaimos&token=444498b8ef5afaa2975a2dc087153139`. The interface is for the 'alaimos' database, specifically viewing the 'graph_path' table.

Left Sidebar: Contains the 'phpMyAdmin' logo, a home icon, and a dropdown menu showing 'alaimos'. Below the dropdown are links for 'graph_path' and 'term', and a 'Crea tabella' button.

Top Toolbar: Includes buttons for 'Mostra', 'Struttura', 'SQL', 'Cerca', 'Inserisci', 'Esporta', 'Importa', 'Operazioni', and 'Tracking'.

Main Content Area:

- A green status bar indicates: 'Mostrando i righi 0 - 29 (1,043,029 totale, La query ha impiegato 0.0270 sec)'.
- A SQL query is displayed: `SELECT * FROM `graph_path` LIMIT 0, 30`.
- Below the query are links: 'Profiling [Inline]', 'Modifica', 'Spiega SQL', 'Crea il codice PHP', and 'Aggiorna'.
- Navigation controls include 'Numero pagina: 1' and buttons for previous/next page.
- Display settings show 'Mostra: 30 righe a partire da # 30 in modalità orizzontale e ripeti gli headers dopo 100 celle'.
- A dropdown for 'Ordina per chiave:' is set to 'Nessuno'.
- A section titled '+ Opzioni' with a sort icon.
- A table with 9 columns: `id`, `term1_id`, `term2_id`, `relationship_type_id`, `distance`, and `relation_distance`. Each row includes action links for 'Modifica', 'Modifica in linea', 'Copia', and 'Elimina'.

	id	term1_id	term2_id	relationship_type_id	distance	relation_distance
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	1	22	19	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	2	22	21	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	3	25	16058	20	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	4	25	17957	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	5	26	13387	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	6	26	13388	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	7	26	13999	20	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	8	26	16512	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	9	26	16513	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	10	26	26807	20	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	11	29	28	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	12	29	35	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	13	29	69	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	14	29	78	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	15	29	234	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	16	29	285	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	17	29	289	1	1	1
<input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	18	29	292	1	1	1

PHPMyAdmin

The screenshot shows the PHPMyAdmin web interface in a browser window. The address bar displays the URL: `alpha.dmi.unict.it/phpmyadmin/index.php?db=alaimos&token=444498b8ef5afaa2975a2dc087153139`. The interface is divided into several sections:

- Left Sidebar:** Contains the PHPMyAdmin logo, a home icon, and a database selector set to 'alaimos'. Below this are icons for 'graph_path' and 'term', and a 'Crea tabella' button.
- Main Content Area:**
 - At the top, there are tabs for 'Mostra', 'Struttura', 'SQL', 'Cerca', 'Inserisci', 'Esporta', 'Importa', 'Operazioni', and 'Tracking'. The 'Struttura' tab is active.
 - Below the tabs is a table showing the structure of the 'term' table. The table has 7 columns: #, Campo, Tipo, Collation, Attributi, Null, Predefinito, Extra, and Azione.
 - Below the table structure, there are links for 'Visualizza per stampa', 'Vedi relazioni', 'Proponi la struttura della tabella', and 'Controlla tabella'.
 - Below these links is a form to 'Aggiungi' (Add) new columns, with a dropdown set to 'id' and an 'Esegui' button.
 - Below the form is a section titled 'Indici:' (Indexes).
- Bottom Section:**
 - On the left, there is a 'Spazio utilizzato' (Space used) section showing the size of the database and its components.
 - On the right, there is a 'Statistiche righe' (Row statistics) section showing various statistics for the table.

Table Structure (term):

#	Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione
1	id	int(11)		No	Nessuno	AUTO_INCREMENT		Modifica Elimina Più
2	name	varchar(255)	latin1_swedish_ci	No				Modifica Elimina Più
3	term_type	varchar(55)	latin1_swedish_ci	No	Nessuno			Modifica Elimina Più
4	acc	varchar(255)	latin1_swedish_ci	No	Nessuno			Modifica Elimina Più
5	is_obsolete	int(11)		No	0			Modifica Elimina Più
6	is_root	int(11)		No	0			Modifica Elimina Più
7	is_relation	int(11)		No	0			Modifica Elimina Più

Indexes:

Azione	Chiave	Tipo	Unica	Compresso	Campo	Cardinalità	Collation	Null	Commenti
Modifica Elimina	PRIMARY	BTREE	Si	No	id	40248	A		
Modifica Elimina	acc	BTREE	Si	No	acc	40248	A		
Modifica Elimina	t1	BTREE	No	No	name	40248	A		
Modifica Elimina	t2	BTREE	No	No	term_type	9	A		
Modifica Elimina	t4	BTREE	No	No	id	40248	A		
Modifica Elimina	t5	BTREE	No	No	acc	40248	A		
Modifica Elimina	t6	BTREE	No	No	id	40248	A		
Modifica Elimina	t6	BTREE	No	No	name	40248	A		
Modifica Elimina	t6	BTREE	No	No	term_type	40248	A		

Spazio utilizzato:

Tipo	Utilizzo
Dati	3,227.0 KIB
Indice	7,095.0 KIB
Totale	10,322.0 KIB

Statistiche righe:

Istruzioni	Valore
Formato	dinamico
Collation	latin1_swedish_ci
Righe	40,248
Lunghezza riga	82
Dimensione riga	263 B
Prossimo Autoindex	40,249
Creazione	Apr 15, 2014 alle 13:30
Ultimo cambiamento	Apr 15, 2014 alle 13:30
Ultimo controllo	Apr 15, 2014 alle 13:30

PHPMyAdmin

The screenshot displays the PHPMyAdmin interface in a web browser. The address bar shows the URL: `alpha.dmi.unict.it/phpmyadmin/index.php?db=alaimos&token=444498b8ef5afaa2975a2dc087153139`. The browser's tab bar includes "alpha.dmi.unict.it / localh...", "DT-Hybrid Web", and "MySQL :: MySQL 5.0 Refer...".

The interface features a sidebar on the left with the "phpMyAdmin" logo, a home icon, and a database selector set to "alaimos". Below the selector are links for "graph_path" and "term", and a "Crea tabella" button.

The main content area is titled "localhost > alaimos > term" and contains several functional panels:

- Altera tabella ordinata per:** A panel for changing the table's sort order. It shows "id" selected for sorting "singolarmente" in "Crescente" order. An "Esegui" button is at the bottom right.
- Sposta la tabella nel (database.tabella):** A panel for moving the table. It shows "alaimos" as the database and "term" as the table. The "Aggiungi valore AUTO_INCREMENT" checkbox is checked. An "Esegui" button is at the bottom right.
- Opzioni della tabella:** A panel for configuring table options. It includes fields for "Rinomina la tabella in" (set to "term"), "Commenti alla tabella", "Motore di Memorizzazione" (set to "MyISAM"), "Collation" (set to "latin1_swedish_ci"), "PACK_KEYS" (set to "DEFAULT"), "CHECKSUM", "DELAY_KEY_WRITE", "AUTO_INCREMENT" (set to "40249"), and "ROW_FORMAT" (set to "DYNAMIC"). An "Esegui" button is at the bottom right.
- Copia la tabella nel (database.tabella):** A panel for copying the table. It shows "alaimos" as the database and "term" as the table. Radio buttons are selected for "Struttura e dati". Checkboxes for "Aggiungi DROP TABLE", "Aggiungi valore AUTO_INCREMENT", and "Passa alla tabella copiata" are present. An "Esegui" button is at the bottom right.
- Amministrazione tabella:** A panel with a list of administrative actions: "Controlla tabella", "Analizza tabella", "Ripara tabella", "Ottimizza tabella", and "Ricarica la tabella (FLUSH)".
- Rimuovi la tabella o i dati:** A panel with two actions: "Svuota la tabella (TRUNCATE)" and "Elimina la tabella (DROP)".

PHPMyAdmin

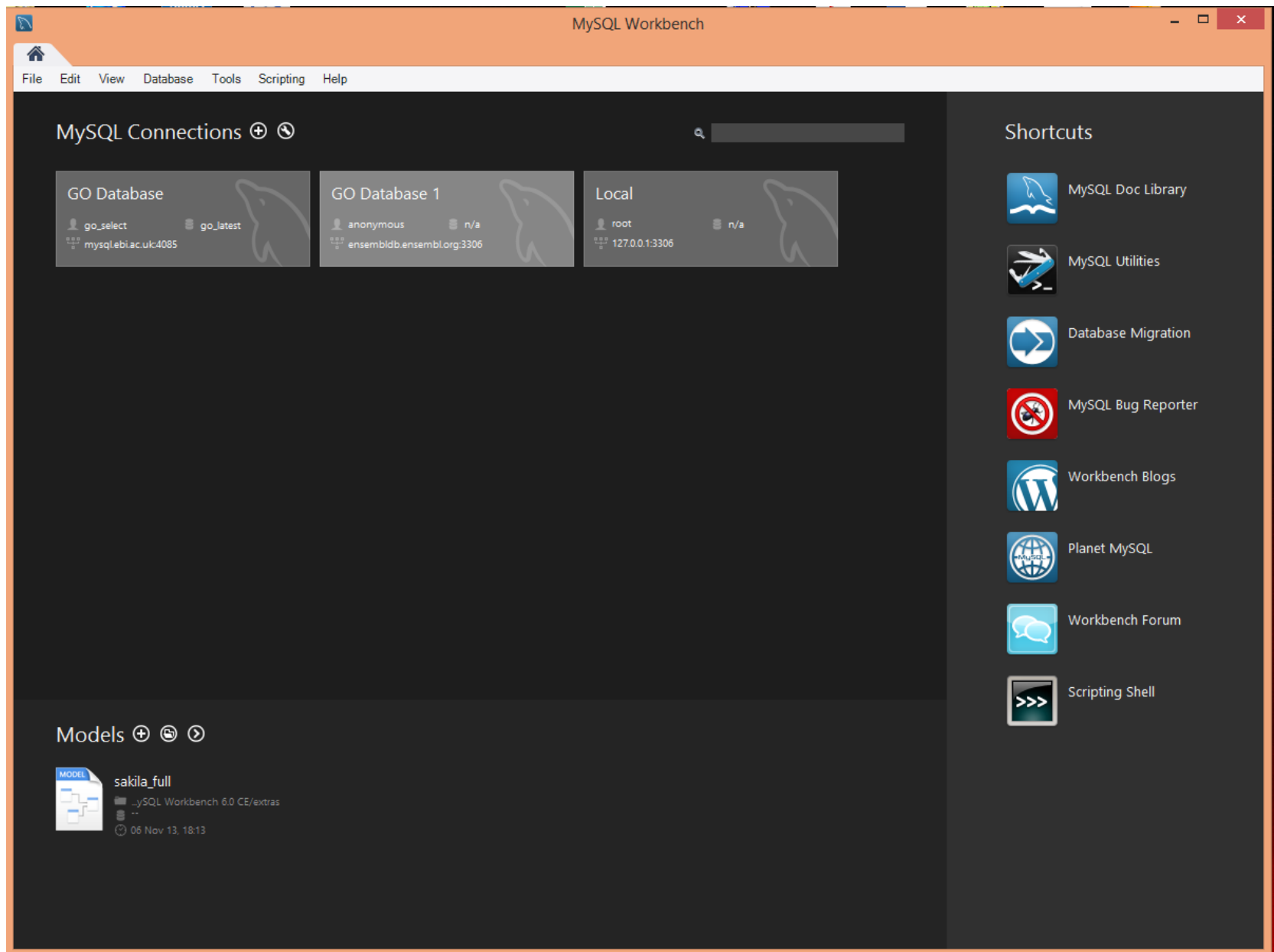
The screenshot shows the PHPMyAdmin interface in a web browser. The browser's address bar displays the URL: `alpha.dmi.unict.it/phpmyadmin/index.php?db=alaimos&token=444498b8ef5afaa2975a2dc087153139`. The browser's tab bar shows several open tabs, including 'alpha.dmi.unict.it / localh...', 'DT-Hybrid Web', and 'MySQL :: MySQL 5.0 Refer...'. The browser's toolbar includes various icons for applications, lazy load, video tutorial, nivo slider, jquery italia, da mkv a dvd, guida alla copia, the last dvd, and altri preferiti.

The PHPMyAdmin interface is displayed. The left sidebar shows the 'alaimos' database selected, with a list of tables: 'graph_path' and 'term'. A 'Crea tabella' button is visible. The main content area shows the 'Struttura' (Structure) tab for the 'alaimos' database. The 'Nome tabella:' field is empty. The 'Struttura' tab displays a table structure with two columns. The columns are defined by the following fields:

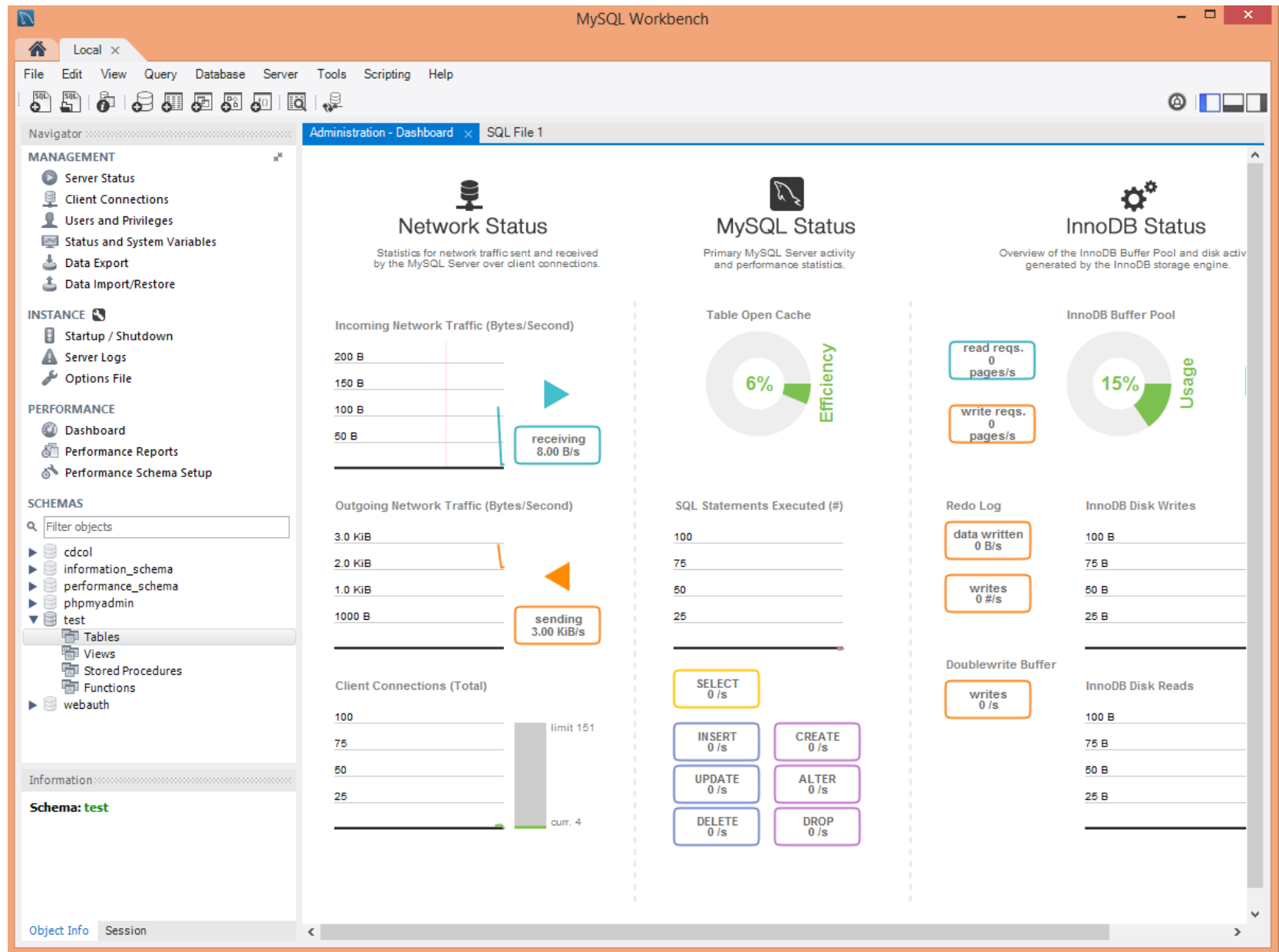
Campo	Tipo	Lunghezza/Set	Predefinito	Collation	Attributi	Null	Indice	AUTO_INCREMENT	Commenti	Tipo MIME	Trasformazione del browser	Opzioni di trasformazione
	INT		Nessuno			<input type="checkbox"/>	---	<input type="checkbox"/>				
	INT		Nessuno			<input type="checkbox"/>	---	<input type="checkbox"/>				

Below the table structure, there are fields for 'Commenti alla tabella:', 'Motore di Memorizzazione:' (set to InnoDB), and 'Collation:'. The 'Definizione PARTITION:' field is also present. At the bottom right, there are buttons for 'Salva', 'Oppure Aggiungi', and 'Esegui', along with a field for the number of columns (set to 1) and a label 'campo/i'.

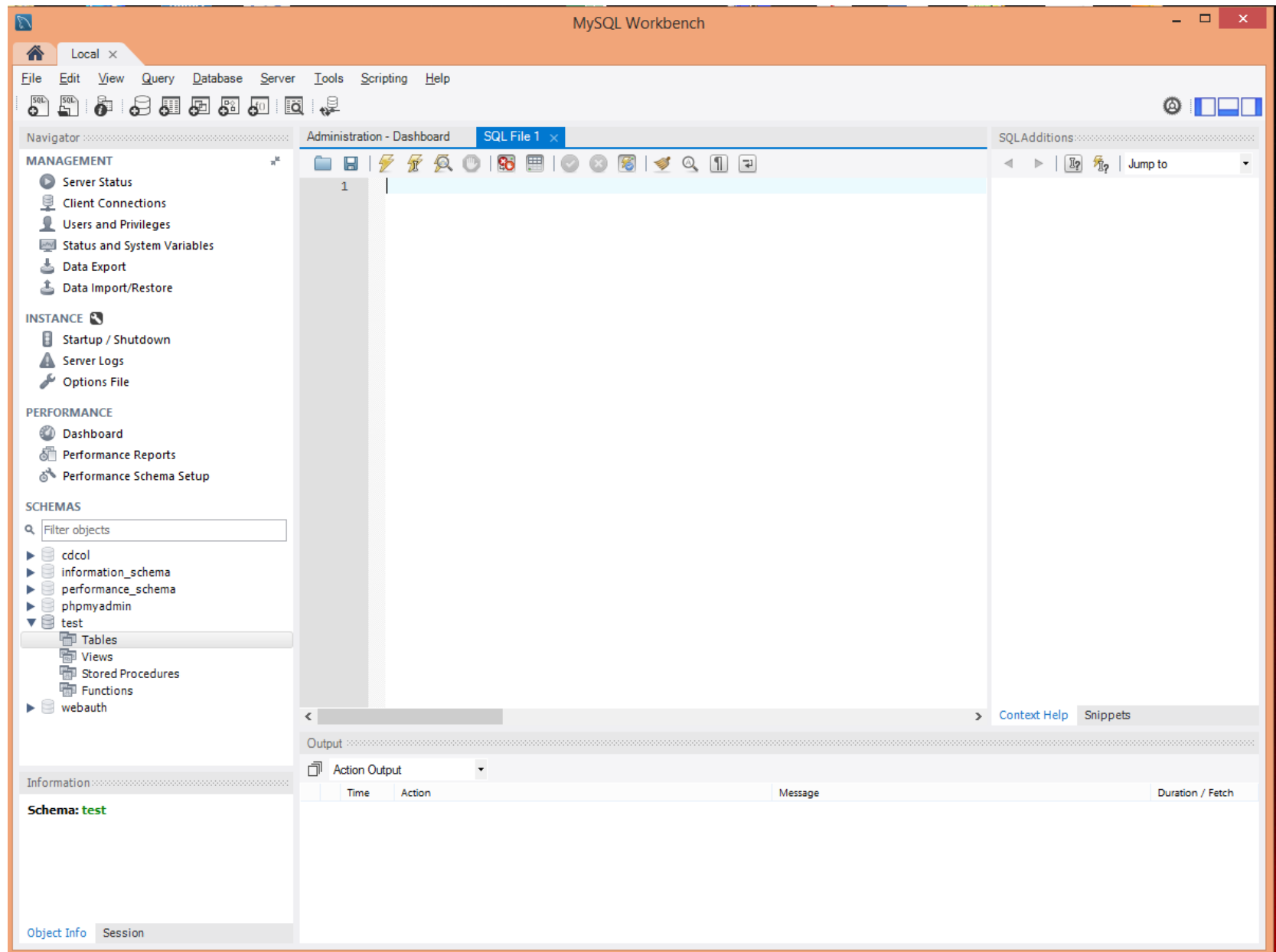
Mysql Workbench



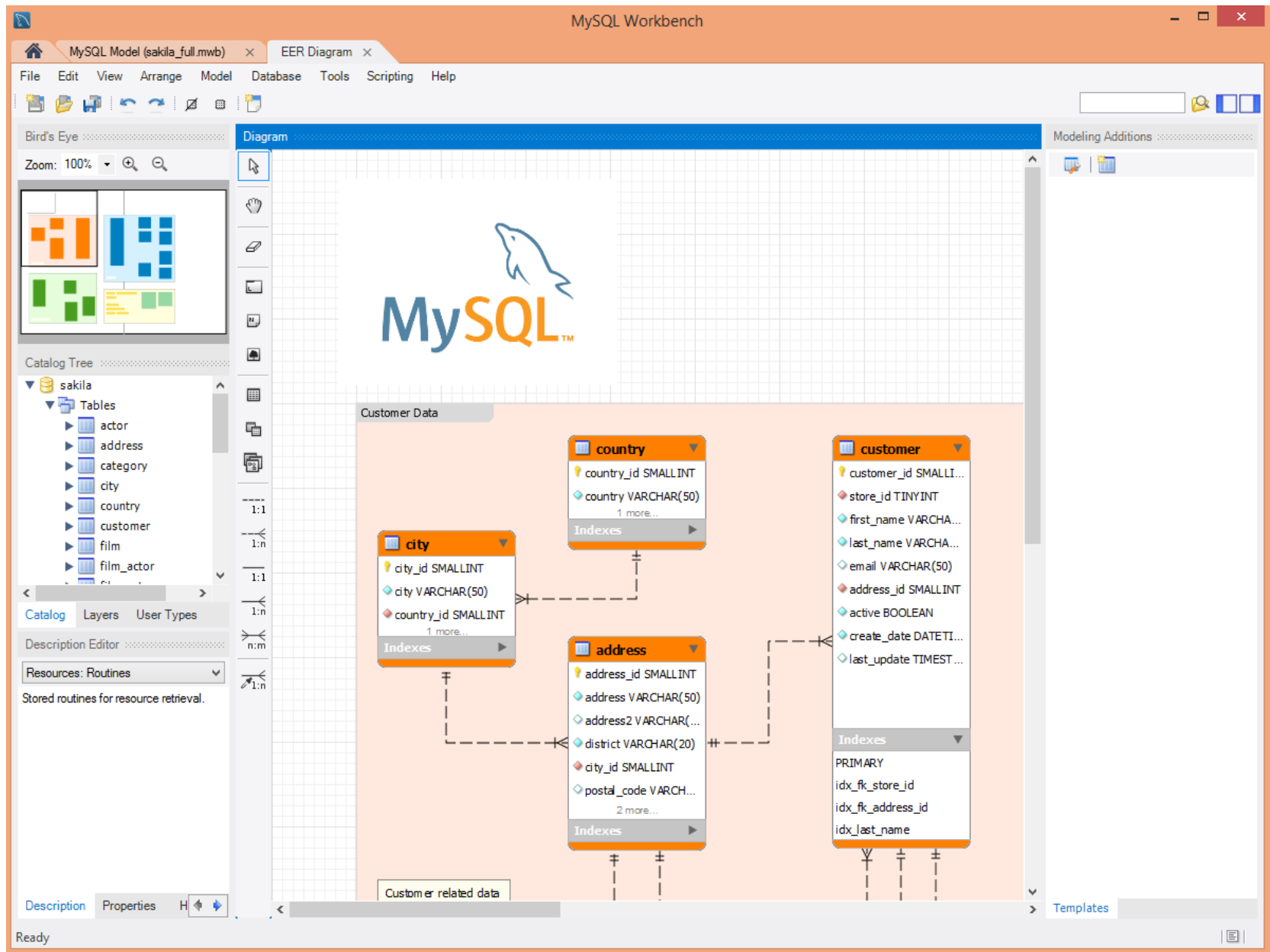
Mysql Workbench



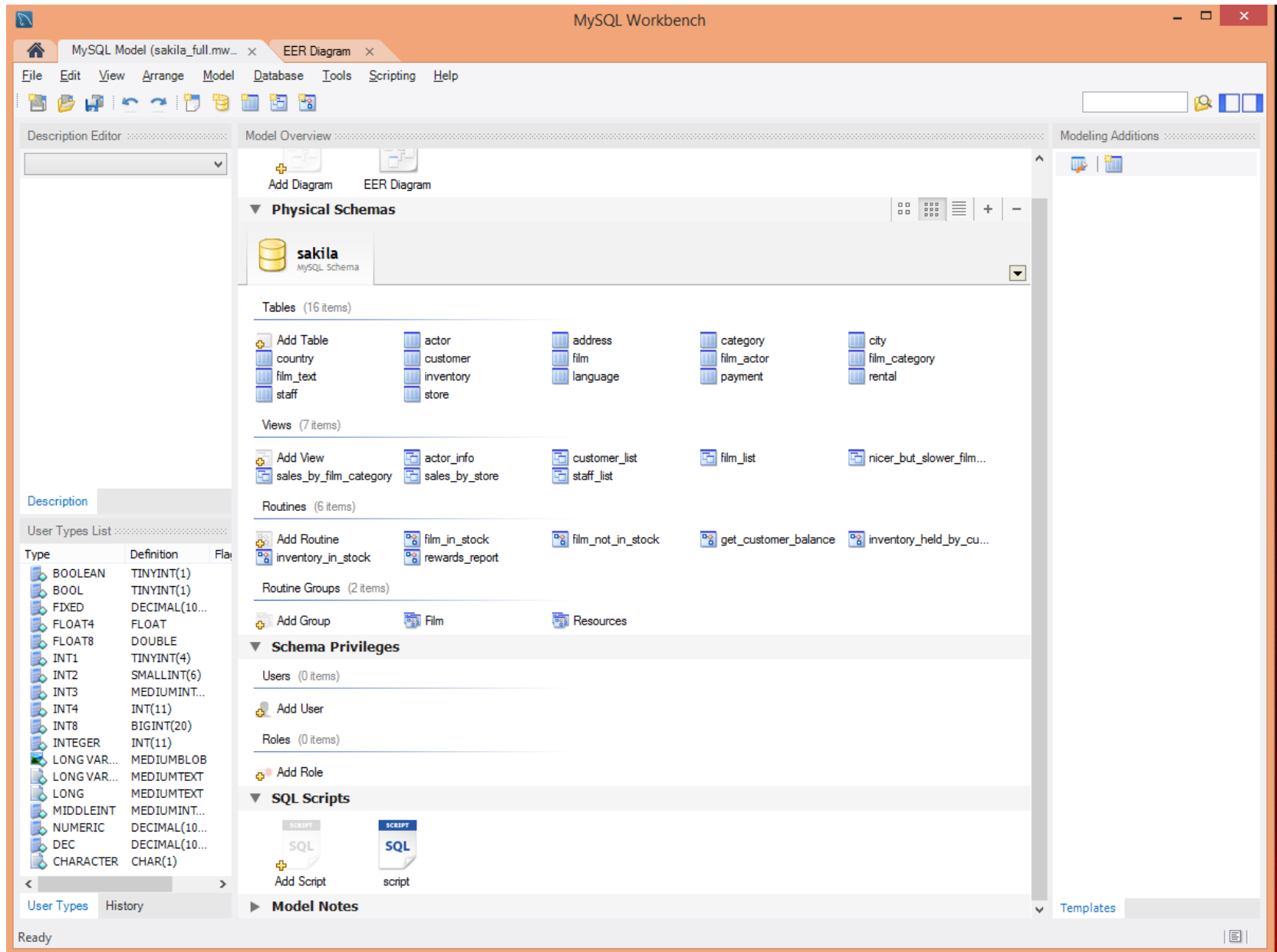
Mysql Workbench



Mysql Workbench



Mysql Workbench



Mysql Workbench

The screenshot displays the MySQL Workbench interface with the EER Diagram tab selected. The main workspace shows the 'Physical Schemas' section for the 'sakila' schema. It lists 16 tables and 7 views. The 'country' table is selected, and its details are shown in the 'Table Name' field and the 'Columns' tab.

Physical Schemas

Tables (16 items)

- country
- customer
- inventory
- store
- actor
- address
- film
- language
- category
- film_actor
- payment
- city
- film_category
- rental

Views (7 items)

- actor_info
- customer_list
- staff_list
- film_list
- nicer_but_slower_film...

Table Name: country

Schema: sakila

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF
country_id	SMALLINT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
country	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
last_update	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Collation:

Comments:

User Types List

Type	Definition	Flag
BOOLEAN	TINYINT(1)	
BOOL	TINYINT(1)	
FIXED	DECIMAL(10...	
FLOAT4	FLOAT	
FLOAT8	DOUBLE	
INT1	TINYINT(4)	
INT2	SMALLINT(6)	
INT3	MEDIUMINT...	
INT4	INT(11)	
INT8	BIGINT(20)	
INTEGER	INT(11)	
LONG VAR...	MEDIUMBLOB	
LONG VAR...	MEDIUMTEXT	
LONG	MEDIUMTEXT	
MIDDLEINT	MEDIUMINT...	
NUMERIC	DECIMAL(10...	
DEC	DECIMAL(10...	
CHARACTER	CHAR(1)	

User Types History

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

Templates

Ready

Parte IV

Account e Privilegi

- Come creare un utente?
 - **CREATE USER** *'name'@'host'* **IDENTIFIED BY** *'PASSWORD'*;
- Come assegnare privilegi ad un utente?
 - **GRANT** *privilege,...* **ON** **| 'db'. *| 'db'. 'table'* **TO** *'username'@'host', ...;*
 - **'privilege'** può essere uno dei seguenti valori:
 - **ALL**
 - **USAGE**
 - **SELECT, INSERT, UPDATE, DELETE**
 - **CREATE, ALTER, INDEX, DROP, CREATE VIEW, TRIGGER**

- Rimuovere un utente:
 - **DROP USER** *'name'@'host',...;*
- Rimuovere i privilegi di un utente:
 - **REVOKE** *privilege,...* **ON** **| 'db'. *| 'db'. 'table'* **FROM** *'username'@'host', ...;*
 - **'privilege'** può essere uno dei seguenti valori:
 - ***ALL PRIVILEGES***
 - ***USAGE***
 - ***SELECT, INSERT, UPDATE, DELETE***
 - ***CREATE, ALTER, INDEX, DROP, CREATE VIEW, TRIGGER***

Parte V

Comandi MySQL

Primi Esempi

– 1. Mostra il nome dell'utente corrente

```
SELECT user();
```

– 2. Mostra la versione, la data, e l'ora corrente

```
SELECT version(), current_date, current_time, current_timestamp;
```

– 3. Mostra l'elenco dei database

```
SHOW DATABASES;
```

- Creazione di un database:
 - **CREATE DATABASE [IF NOT EXISTS]** *nome*;
- Cancellazione di un database:
 - **DROP DATABASE [IF EXISTS]** *nome*;
- Accesso ad un database:
 - **USE** *nome*;

- **CREATE TABLE [IF NOT EXISTS]** *nome* (
 campo1 **TIPO1** **ALTRI PARAMETRI**,
 campo2 **TIPO2** **ALTRI PARAMETRI**,
 ...
 campoN **TIPON** **ALTRI PARAMETRI**,
 PRIMARY KEY(*campo1*, *campo2*)
) **ENGINE**=InnoDB;
- **IF NOT EXISTS**: crea la tabella solo se non esiste.
- Altri parametri:
 - **NOT NULL**: non permette valori NULL nella colonna;
 - **AUTO_INCREMENT**: una colonna il cui valore è calcolato automaticamente in base ad un contatore interno;
 - **DEFAULT** *valore*: specifica un valore di default per un campo;
 - **NULL**: specifica che un campo può contenere valori NULL.

Gestione Database e Tabelle

```
-- 1. Crea un database
CREATE DATABASE IF NOT EXISTS prova;

-- 2. Seleziona il database di prova
USE prova;

-- 3. Crea una tabella di esempio
CREATE TABLE country (
  country_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  country VARCHAR(50) NOT NULL,
  last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (country_id)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE city (
  city_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  city VARCHAR(50) NOT NULL,
  country_id SMALLINT UNSIGNED NOT NULL,
  last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (city_id),
  KEY idx_fk_country_id (country_id),
  CONSTRAINT `fk_city_country` FOREIGN KEY (country_id) REFERENCES country (country_id) ON DELETE RESTRICT ON UPDATE CASCADE
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

- **DESCRIBE** *nome*: mostra informazioni sui campi contenuti in una tabella.
- **INSERT INTO** *table* (*field1*, ..., *fieldN*) **VALUES** (*value1*, ..., *valueN*) [
ON DUPLICATE KEY UPDATE

field1=value1,

...

fieldN=valueN]

- **INSERT INTO** *table* (*field1*, ..., *fieldN*) **SELECT** ... [**ON DUPLICATE**
KEY UPDATE

field1=value1,

...

fieldN=valueN]

Gestione Database e Tabelle

-- 4. Uso del comando DESCRIBE

DESCRIBE city;

-- 5. Esempio Insert

INSERT INTO country (country_id, country) VALUES (1,'Afghanistan'), (NULL, 'Algeria');

INSERT INTO city (city_id, city, country_id) VALUES (1,'Kabul',1);

-- 6. Cancella la tabella di esempio

DROP TABLE city;

-- 7. Cancella un database

DROP DATABASE IF EXISTS prova;

Field	Type	Null	Key	Default	Extra
city_id	smallint(5) unsigned	NO	PRI	NULL	auto_increment
city	varchar(50)	NO		NULL	
country_id	smallint(5) unsigned	NO	MUL	NULL	
last_update	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

Select

SELECT

```
[ALL | DISTINCT | DISTINCTROW |
  [HIGH_PRIORITY]
  [STRAIGHT_JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
  [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr ...]
[FROM table_references
[WHERE where_condition]
[GROUP BY {col_name | expr | position}
  [ASC | DESC], ... [WITH ROLLUP]]
[HAVING where_condition]
[ORDER BY {col_name | expr | position}
  [ASC | DESC], ...]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[PROCEDURE procedure_name(argument_list)]
[INTO OUTFILE 'file_name'
  [CHARACTER SET charset_name]
  export_options
  | INTO DUMPFILE 'file_name'
  | INTO var_name [, var_name]]
[FOR UPDATE | LOCK IN SHARE MODE]]
```

Select

- **SELECT** seguita da una o più espressioni che saranno le colonne della tabella risultato;
- **FROM** seguita da i nomi di una o più tabelle dalle quali devono essere estratti i dati, per ogni tabella si può specificare un alias;
- **WHERE** che specifica le condizioni in base alle quali ogni riga sarà estratta oppure no dalle tabelle;
- **GROUP BY** che specifica le colonne sui cui valori devono essere raggruppate le righe nel risultato: tutte le righe con valori uguali saranno ridotte a una;
- **HAVING** che specifica ulteriori condizioni da applicare alle righe dopo il raggruppamento effettuato dalla **GROUPBY**;
- **ORDER BY** che specifica in quale ordine figureranno le righe del risultato;
- **LIMIT** che stabilisce il massimo numero di righe da estrarre.

Select

```
-- 1. Seleziona gli identificativi, i nomi e le email dei primi 10 clienti ordinati per cognome e nome
SELECT customer_id, last_name, first_name, email
      FROM customer
      ORDER BY last_name ASC, first_name ASC
      LIMIT 0,10;

-- 2. Seleziona gli utenti che sono stati aggiunti dopo le 22:04:37 del 14 aprile 2006
SELECT * FROM customer
      WHERE create_date >= '2006-02-14 22:04:37'
      ORDER BY last_name ASC;

-- 3. Cerca i film il cui nome inizia per "W"
SELECT * FROM film
      WHERE title LIKE 'W%'
      ORDER BY title ASC;

-- 4. Altro modo
SELECT * FROM film
      WHERE title REGEXP '^W'
      ORDER BY title ASC;

-- 5. Cerca i film che contengono "W*R"
SELECT * FROM film
      WHERE title LIKE '%W_R%'
      ORDER BY title ASC;
```

- Il valore **NULL** per un campo assume il significato di «mancante, sconosciuto»
 - Esso è trattato diversamente dagli altri valori.
- Per testare il valore di **NULL** non si possono usare i consueti operatori di confronto **=**, **<**, o **<>**
 - Esistono due operatori di confronto appositi per valori **NULL**:
 - **IS NULL**
 - **IS NOT NULL**
- Quando si usa **ORDER BY** i valori **NULL** sono inseriti all'inizio con **ASC** ed alla fine con **DESC**.

I Valori NULL

```
-- 7. Cerca i film che non sono ancora stati restituiti;
SELECT F.film_id, F.title, F.description, I.store_id, R.return_date
FROM film AS F, inventory AS I, rental AS R
WHERE
    I.film_id = F.film_id AND
    R.inventory_id = I.inventory_id AND
    R.return_date IS NULL
ORDER BY F.title ASC;
```

Select

-- 8. Quante volte un film è stato noleggiato e restituito?

```
SELECT F.film_id, F.title, COUNT(R.rental_id) AS Conteggio
FROM film AS F, inventory AS I, rental AS R
WHERE
    I.film_id = F.film_id AND
    R.inventory_id = I.inventory_id
GROUP BY F.film_id
ORDER BY Conteggio DESC;
```

-- 9. Perché non compaiono quelli con 0 noleggi e restituzioni? Bisogna costruire una query più complessa

```
SELECT F.film_id, F.title, IFNULL(C.Conteggio, 0) AS ConteggioCorretto
FROM film AS F
LEFT JOIN (
    SELECT I.film_id, COUNT(R.rental_id) AS Conteggio
    FROM inventory AS I
    LEFT JOIN rental AS R ON R.inventory_id = I.inventory_id
    GROUP BY I.film_id
) AS C ON F.film_id = C.film_id
ORDER BY Conteggio DESC;
```

Update

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference
  SET col_name1={expr1!DEFAULT} [, col_name2={expr2!DEFAULT}] ...
  [WHERE where_condition]
  [ORDER BY ...]
  [LIMIT row_count]

Multiple-table syntax:

UPDATE [LOW_PRIORITY] [IGNORE] table_references
  SET col_name1={expr1!DEFAULT} [, col_name2={expr2!DEFAULT}] ...
  [WHERE where_condition]
```

- **SET:** specifichiamo quali colonne modificare e quali valori assegnare;
- **WHERE:** le condizioni che determinano quali righe saranno modificate;
- **ORDER BY:** per decidere in che ordine effettuare gli aggiornamenti;
- **LIMIT:** per indicare il numero massimo di righe da modificare.

Delete

Single-table syntax:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name  
    [WHERE where_condition]  
    [ORDER BY ...]  
    [LIMIT row_count]
```

Multiple-table syntax:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]  
    tbl_name[.*] [, tbl_name[.*]] ...  
FROM table_references  
    [WHERE where_condition]
```

Or:

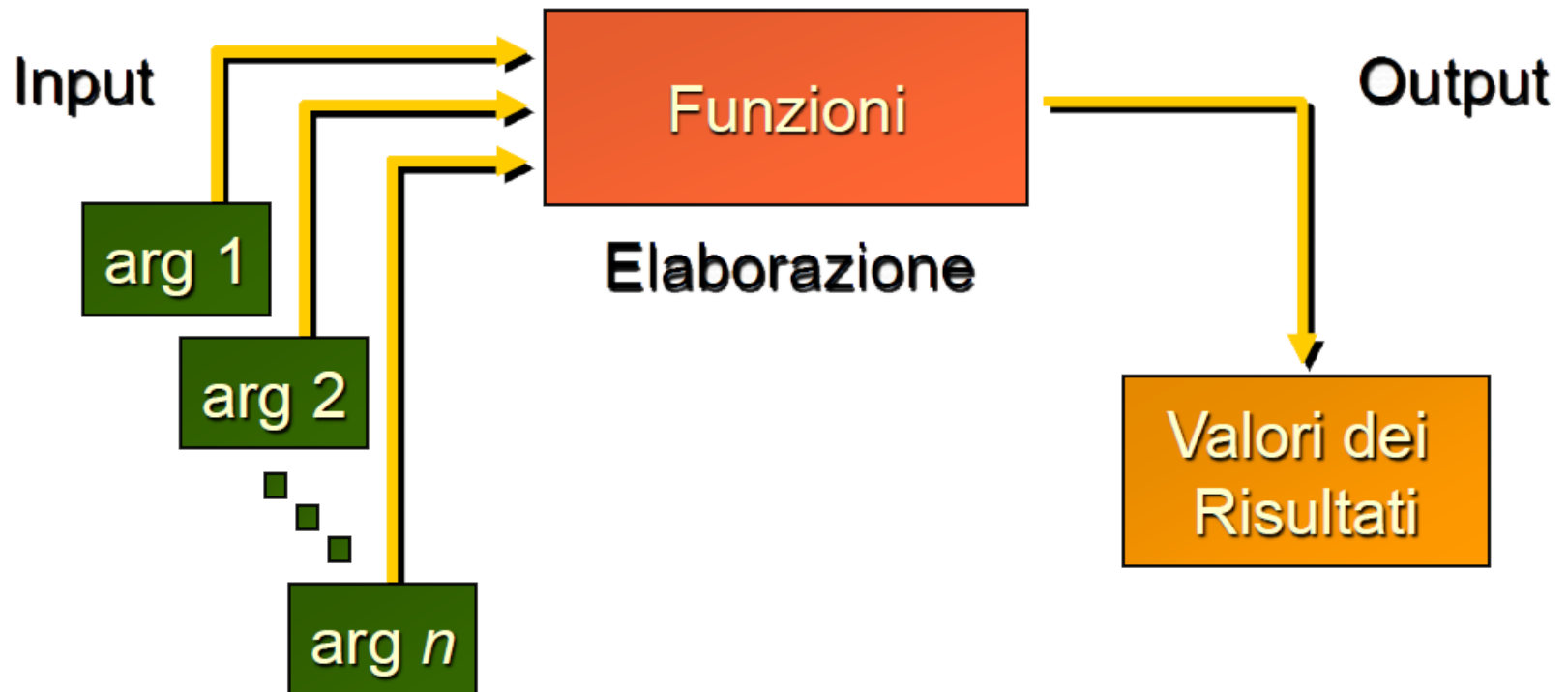
```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]  
    FROM tbl_name[.*] [, tbl_name[.*]] ...  
    USING table_references  
    [WHERE where_condition]
```

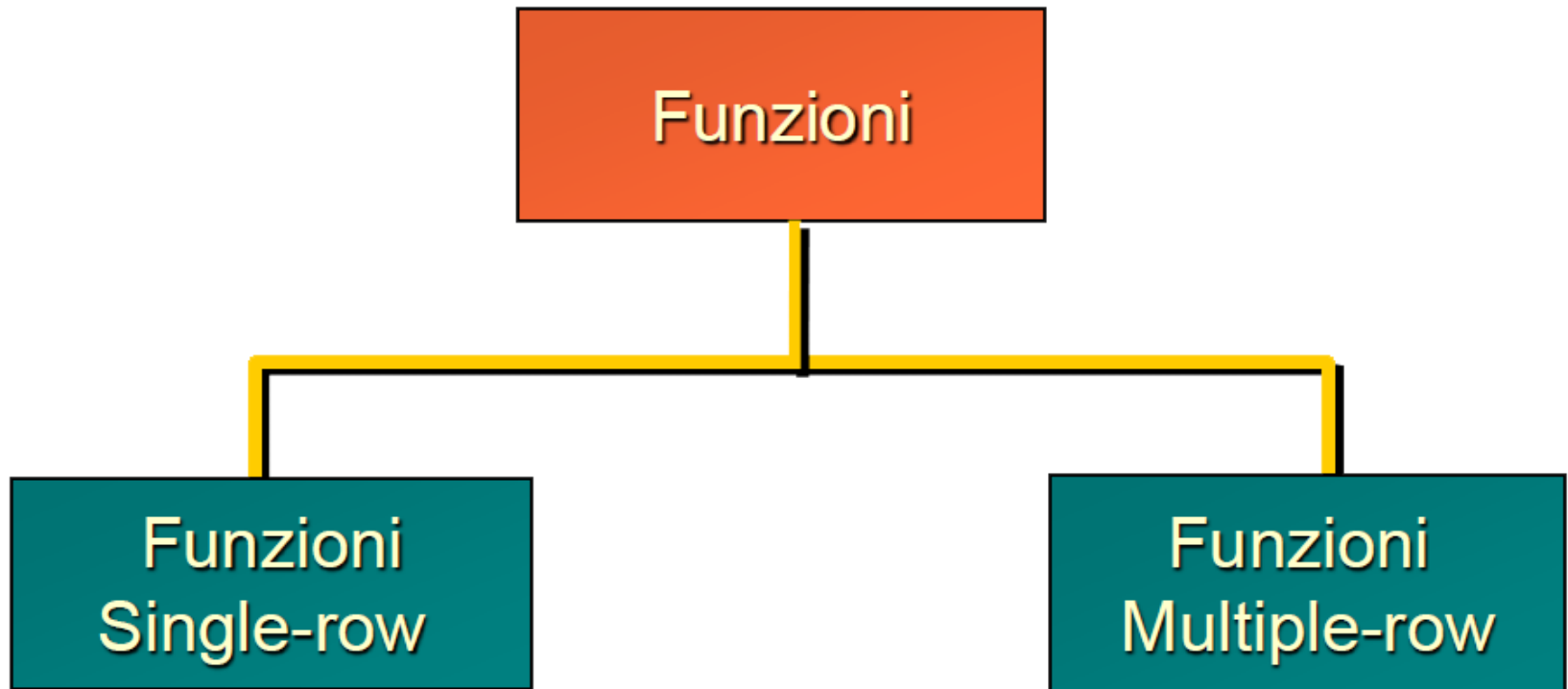
- **ORDER BY** e **LIMIT**: funzionano come in **UPDATE**;
- **WHERE**: stabilisce le condizioni in base alle quali le righe verranno eliminate .

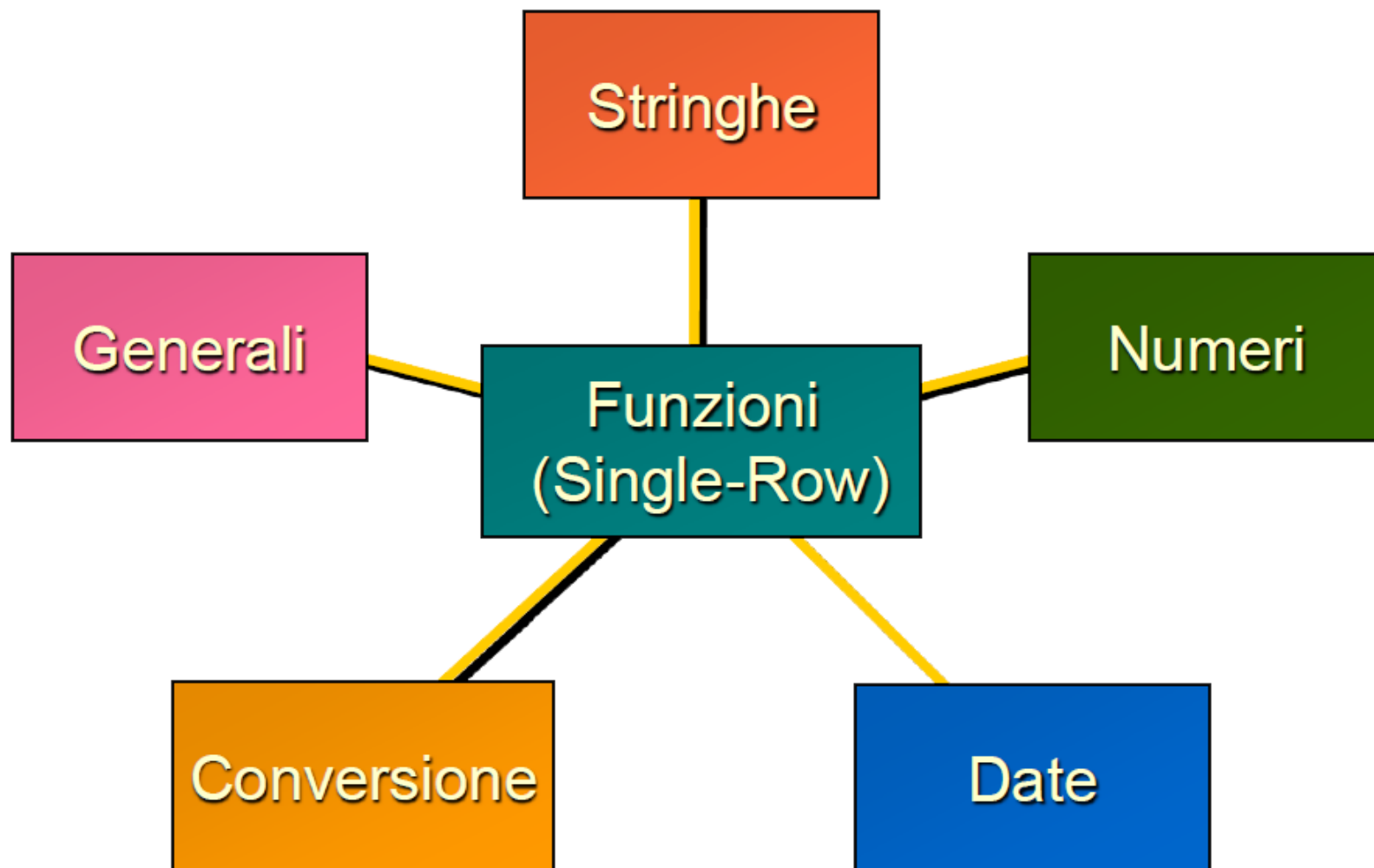
Parte VI

Funzioni e operatori

Funzioni







- **Aritmetici:**
 - "+" (**addizione**);
 - "-" (**sottrazione**);
 - "*" (**moltiplicazione**);
 - "/" (**divisione**);
 - "%" (**modulo**);
- **Matematici:**
 - **ABS**(X)
 - **FLOOR**(X)
 - **CEILING**(X)
 - **SIN**(X)
 - **COS**(X)
 - **LN**(X)
 - **LOG**(X)
 - **LOG**(B,X)
- **Logici:**
 - **NOT**(!);
 - **AND**(&&);
 - **OR**(||)
 - **XOR**;
- **Confronto**
(Risultato 1 o 0):
 - =;
 - <> , !=;
 - <= , < , >= ;
 - > , <=> (NULL-safe);
 - **IS NULL**;
 - **IS NOT NULL**.

- Per controllare se un numero è all'interno di un range di valori si può usare una delle seguenti espressioni:
 - *expr* **BETWEEN** *min* **AND** *max*
 - *expr* **NOT BETWEEN** *min* **AND** *max*
- Per confrontare rispetto ad una lista fissata di valori:
 - *expr* **IN** (*value*, ...)
 - *expr* **NOT IN** (*value*, ...)
- **COALESCE**(*val*, ...): restituisce il primo elemento non-NULL di una lista;
- **INTERVAL**(*N*,*N1*,*N2*,*N3*,...):
Ritorna:
 - 0 se *N* < *N1*;
 - 1 se *N* < *N2*;
 - ecc...
 - -1 se *N* è NULL.

Controllo del flusso

```
CASE value WHEN [compare_value] THEN result [WHEN [compare_value] THEN  
result ...] [ELSE result] END
```

```
CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...]  
[ELSE result] END
```

```
IF(expr1,expr2,expr3)
```

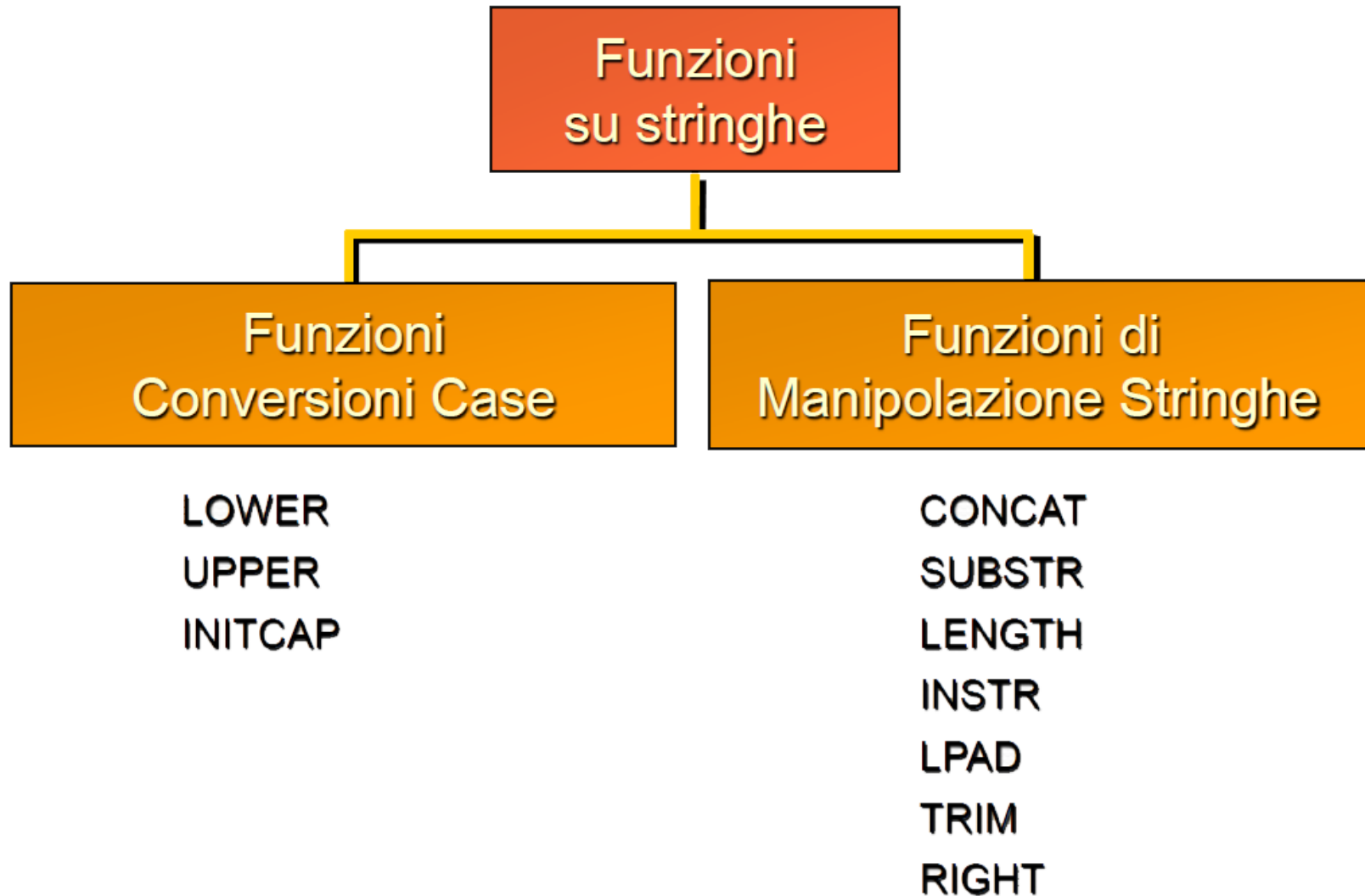
Syntax:

```
IFNULL(expr1,expr2)
```

Controllo del flusso

```
SELECT film_id, title,  
       CASE rating  
       |   WHEN 'G' THEN 'General Audiences'  
       |   WHEN 'PG' THEN 'Parental Guidance Suggested'  
       |   WHEN 'PG-13' THEN 'Parents Strongly Cautioned'  
       |   WHEN 'R' THEN 'Restricted'  
       |   WHEN 'NC-17' THEN 'No one 17 and under admitted'  
       |   ELSE 'There is nothing else'  
       |  
       END AS ExplainedRating,  
       IF(STRCMP(rating, 'NC-17')=0, 'YES', 'no') AS RequireID FROM film;
```

Funzioni su stringhe



Funzioni su stringhe

- Funzione conversioni case:
 - **LOWER**(*str*) , **LCASE**(*str*)
 - **UPPER**(*str*) , **UCASE**(*str*)
- Funzioni manipolazione stringhe:
 - **ASCII**(*str*): ritorna il valore numerico del carattere più a sinistra di *str*;
 - **BIN**(*N*): Ritorna una stringa che rappresenta il valore binario di *N*;
 - **CONCAT_WS**(*separator, str1, str2, ...*): il primo argomento è il separatore il resto gli argomenti;
 - **CONV**(*N, from_base, to_base*): converte i numeri tra differenti basi;
 - **BIT_LENGTH**(*str*): Ritorna la lunghezza della stringa *str* in bit;
 - **CHAR**(*N, ...*): interpreta ogni argomento *N* come intero e ritorna una stringa consistente dei caratteri dati dal codice numerico degli interi;
 - **CHAR_LENGTH**(*str*): ritorna la lunghezza della stringa misurata in caratteri;
 - **CONCAT**(*str1, str2, ...*): ritorna la stringa che si ottiene concatenando gli argomenti. Ritorna NULL se un argomento è NULL.

Funzioni su stringhe

- Funzioni manipolazione stringhe:

- **ELT**(*N, str1, str2, str3, ...*): Ritorna *str1* se *N*=1, *str2* se *N*=2 ect...;
- **FIELD**(*str, str1, str2, str3, ...*): Ritorna la posizione di *str* in *str1, str2, ect...*;
- **FIND_IN_SET**(*str, strlist*): Ritorna un valore nel range tra 1 a *N* se la stringa *str* è nella lista delle stringhe *strlist* (ovvero *N* sottostringhe, separate da ",");
- **HEX**(*N_o_S*): ritorna il valore esadecimale della stringa;
- **LEFT**(*str, len*): Ritorna i *len* caratteri più a sinistra di *str*;

- **INSTR**(*str, substr*): Ritorna la posizione della prima occorrenza della *substr* in *str*;
- **LENGTH**(*str*): Ritorna la lunghezza della stringa *str* in bytes;
- **LOCATE**(*substr, str*) / **LOCATE**(*substr, str, pos*): La prima sintassi ritorna la posizione delle prima occorrenza di *substr* in *str*, la seconda inizia la ricerca dalla posizione *pos*;
- **LTRIM**(*str*): Ritorna *str* con gli spazi iniziali rimossi.

Funzioni su stringhe

- **REPEAT**(*str,count*): Ritorna una stringa formata da *str* ripetuta *count* volte;
- **REPLACE**(*str,from,to*): Rimpiazza tutte le occorrenze di *from* in *to* da *str*;
- **REVERSE**(*str*): Ritorna la stringa invertita;
- **RIGHT**(*str,len*): Ritorna i *len* caratteri più a destra di *str*;
- **RTRIM**(*str*): Ritorna la stringa *str* con gli spazi finali rimossi;
- **STRCMP**(*expr1,expr2*): Ritorna **0** (**zero**) se le due stringhe sono uguali, **-1** se il primo argomento è più piccolo del secondo, **1** altrimenti.

Funzioni su stringhe

```
SELECT C.customer_id,  
       CONCAT(C.first_name, ' ', C.last_name) AS name,  
       CONCAT_WS(', ', A.address, IF(STRCMP(A.address2,'')=0, NULL, A.address2)) AS address,  
       A.postal_code, A.phone, CI.city, CO.country  
FROM  
       customer AS C  
       JOIN address AS A ON A.address_id = C.address_id  
       JOIN city AS CI ON CI.city_id = A.city_id  
       JOIN country AS CO ON CO.country_id = CI.country_id  
WHERE active <> 0;
```

- **MATCH** (*col1,col2,...*) **AGAINST** (*expr* [**IN BOOLEAN MODE** | **WITH QUERY EXPANSION**]):

- **MATCH ... AGAINST** è utilizzata per ricerche full text:
 - ritorna la rilevanza tra il testo che si trova nelle colonne (*col1,col2,...*) e la query *expr*.
 - La similarità è un valore positivo in virgola mobile.
- La funzione **match** esegue una ricerca in linguaggio naturale per una stringa contro un testo che è rappresentato da una o più colonne incluse in un indice FULL TEXT.
- La ricerca di default è case-insensitive.

```
SET @WORD = 'circus';
SELECT film_id, title,
       MATCH(title, description) AGAINST (@WORD) AS score
FROM film_text
WHERE MATCH(title, description) AGAINST (@WORD);
```

- È possibile eseguire una ricerca full-text in boolean mode, i segni + e – indicano le parole che devono essere presenti o assenti, rispettivamente per un match che occorre.
- A volte la stringa di ricerca è troppo corta e potrebbe tralasciare dei risultati significativi, è possibile utilizzare **query expansion** per ovviare a questo problema.

```
SET @WORD = 'circus';
SELECT film_id, title,
       MATCH(title, description) AGAINST (@WORD WITH QUERY EXPANSION) AS score,
       MATCH(title, description) AGAINST (@WORD) AS score2
FROM film_text
WHERE MATCH(title, description) AGAINST (@WORD WITH QUERY EXPANSION);
```

Stored Procedures

```
DELIMITER //
```

```
CREATE PROCEDURE nome ([IN|OUT|INOUT nomeParametro tipo, ...])
```

```
BEGIN
```



```
END; //
```

```
DELIMITER ;
```

```
CALL nomeProc(...);
```

```
CREATE PROCEDURE nomeProc (IN param1 INT, OUT param2 INT)
```

```
SELECT COUNT(*) INTO param2 FROM tabella
```

```
WHERE campo1 = param1;
```

```
CALL nomeProc (5, @a);
```

```
SELECT @a;
```

Stored Procedures

```
DELIMITER //
CREATE PROCEDURE procedura1 (param1 INT, param2 CHAR(3), OUT param3 INT)
BEGIN
    DECLARE finito INT default 0;
    DECLARE a INT;
    DECLARE b CHAR(50);
    DECLARE cur1 CURSOR FOR SELECT id,nome
        FROM clienti WHERE cat = param2;
    DECLARE CONTINUE HANDLER FOR SQLSTATE '02000'
        SET finito = 1;
    OPEN cur1;
    SET param3 = 0;
    FETCH cur1 INTO a,b;
    ciclo: WHILE NOT finito DO
        IF param3 < param1 THEN
            SET param3 = param3 + 1;
            FETCH cur1 INTO a,b;
        ELSE
            LEAVE ciclo;
        END IF;
    END WHILE ciclo;
END; //
DELIMITER ;
```

Stored Procedures

```
DECLARE a INT default 0;
DECLARE cond1 CONDITION FOR 1045;
DECLARE cond2 CONDITION FOR SQLSTATE '02000';
DECLARE cur1 CURSOR FOR query;
DECLARE EXIT HANDLER FOR cond1 SET variabile = valore;
DECLARE CONTINUE HANDLER FOR cond2 SET variabile = valore;
DECLARE CONTINUE HANDLER FOR SQLWARNING;
OPEN cur1;
FETCH cur1 INTO variabile1 variabile2;
CLOSE cur1;
IF condizione THEN istruzioni
    [ELSEIF condizione THEN istruzioni] ...
    [ELSE istruzioni]
END IF;
[nome:] LOOP
    istruzioni
END LOOP [nome];
[nome:] REPEAT
    istruzioni
    UNTIL condizione
END REPEAT [nome];
[nome] WHILE condizione DO
    istruzioni
END WHILE [nome];
ITERATE nomeCiclo;
LEAVE nomeCiclo;
```

Stored Functions

```
DELIMITER //
```

```
CREATE FUNCTION nome ([parametro1 tipo1, parametro2 tipo2, ...])  
RETURNS tipoRitorno  
BEGIN  
  
    RETURN qualcosa  
  
END; //
```

```
DELIMITER ;
```



```
SELECT ..., nome(...)
```