



POLITECNICO
MILANO 1863

SISTEMI INFORMATIVI

A.A. 2018-19

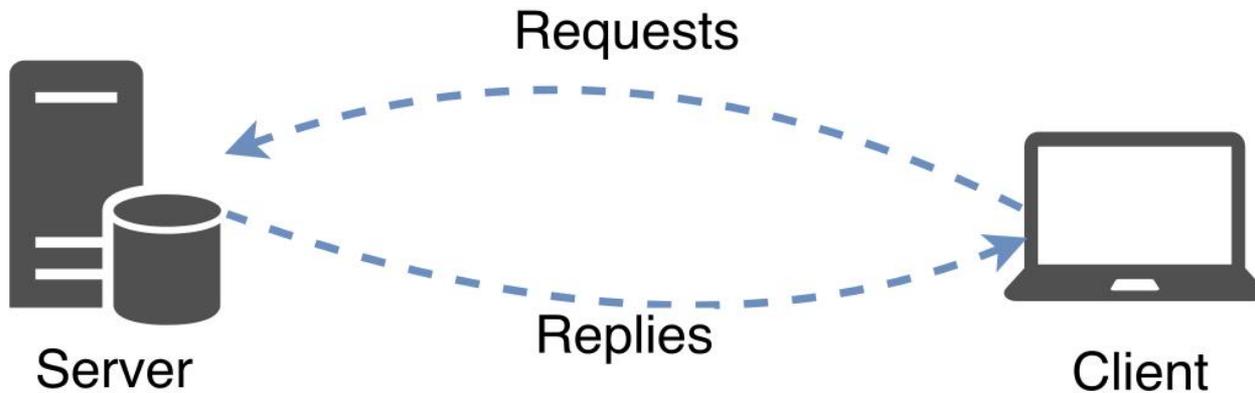
Esercitazione 5 - MySQL

Dott. Michele Zanella

- Introduzione a MySQL
- Download & Installazione
- MySQL Workbench
- Interazione Command Line Client (Approfondimento)

- Open source Relational Database Management System (RDBMS) sviluppato da Oracle
- Basato su basi di dati *relazionali*
- Supporta ANSI/ISO SQL standard
- Suite integrata con programmi client, server e utility
- **Obiettivi**
 - Conoscenza dei tools disponibili
 - Configurazione di MySQL
 - Gestione di basi di dati (creazione, modifica, etc...)
 - Effettuare interrogazioni e recuperare informazioni tramite i tools a riga di comando e grafici

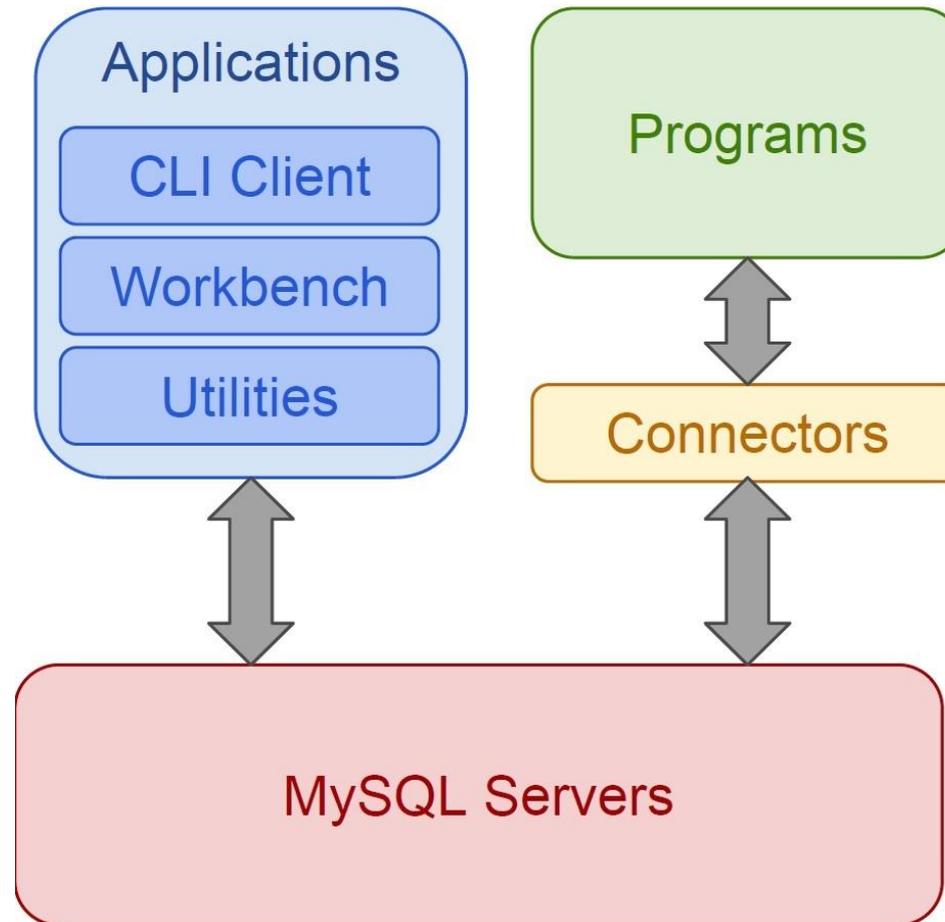
- Architettura Client-Server



- La suite è principalmente composta da 3 categorie di prodotti:
 - **MySQL Servers**: software che operano direttamente sulle basi di dati.
Nelle nostre applicazioni saranno in esecuzioni in *background*. In applicazioni reali sono ospitati in macchine dedicate (*server*) che sono in grado di gestire molte richieste provenienti dalle applicazioni.
 - ***Applications***: software e gli strumenti necessari per interfacciarsi (e.g., *clients*) e gestire i servers (e.g., *utilities*).
 - **Connectors**: sono delle librerie utilizzate dagli sviluppatori per utilizzare ed integrare i servizi offerti dal DBMS nei propri software.

- **Applications:**
 - [CLI Client](#): client a riga di comando con il quale è possibile:
 - Connettersi e gestire un MySQL Server
 - Gestire le basi di dati (e.g., creare o modificare DBs, tabelle, inserire valori, etc..)
 - Eseguire interrogazioni sui DB
 - [Workbench](#): client ad interfaccia grafica, estende le funzionalità del client testuale:
 - Gestione grafica delle basi di dati
 - Migrazione e backup dei dati
 - *Notifier*
 - *Utilities*

Introduzione: MySQL



- **Leggere slides dal titolo “MySQL – Installazione”**

Download & Installazione

- Guide online:
 1. MySQL Installer:
 - Windows: <https://dev.mysql.com/doc/refman/5.7/en/mysql-installer-setup.html>
 - Linux: <https://dev.mysql.com/doc/refman/5.7/en/linux-installation.html>
 - OS X: <https://dev.mysql.com/doc/refman/5.7/en/osx-installation.html>
 2. MySQL Tutorial:
 - CLI Shell: <https://dev.mysql.com/doc/refman/5.7/en/tutorial.html>
 - Workbench: <http://dev.mysql.com/doc/workbench/en/>

- Tipi di dato (<https://dev.mysql.com/doc/refman/5.7/en/data-types.html>):
 - **Numerici:** BIT, INT, BOOL, DECIMAL, FLOAT
 - **Temporal:** DATE, DATETIME, TIMESTAMP, TIME, YEAR
 - **Stringhe:** CHAR, VARCHAR, TEXT
- Alcuni tipi permettono di specificare la lunghezza del dato:
 - INT (2): intero di due cifre
 - VARCHAR (20): stringa di lunghezza massima 20

- Indici e chiavi esterne:
 - **PRIMARY KEY (PK):** Indice univoco dove tutte le colonne che compongono la chiave sono definiti come NOT NULL
<https://dev.mysql.com/doc/refman/5.7/en/create-table.html>
 - **UNIQUE (UQ):** vincolo per cui tutti i valori della colonna devono essere univoci.
 - **FOREIGN KEY (FK):** identificatore esterno che permette di referenziare dati tra diverse tabelle. Normalmente si riferisce alla chiave della tabella esterna.
https://www.w3schools.com/sql/sql_foreignkey.asp
 - **REFERENCES:** identificatore della tabella esterna da referenziare con la FOREIGN KEY
 - **ON DELETE/ ON UPDATE:** specificare l'azione da intraprendere quando il dato viene cancellato o modificato (NO ACTION, CASCADE, SET NULL)

- Ambiente grafico che permette di gestire ed interrogare il database
- Esegue in maniera trasparente all'utente i comandi SQL
- Più intuitivo e diretto per alcune applicazioni rispetto alla CLI
- Permette di visualizzare e modificare i comandi che vengono eseguiti ad ogni operazione.
- **Guide e tutorial online ufficiali:**
 - <http://dev.mysql.com/doc/workbench/en/>
 - <https://dev.mysql.com/doc/workbench/en/wb-home.html>
 - <https://dev.mysql.com/doc/workbench/en/wb-admin.html>
 - <https://dev.mysql.com/doc/workbench/en/wb-sql-editor.html>

- Funzionalità:
 - **Esecuzione comandi SQL:** creare e gestire connessioni con i server db.
Possibilità di eseguire query SQL sui db connessi.
 - **Modellizzazione dei dati:** creare e modificare graficamente lo schema dei db attraverso il *Table Editor* (tabelle, colonne, indici,...)
 - **Amministrazione MySQL Servers:** possibilità di effettuare backup e recovery e analisi sulle performance
 - **Migrazione dei dati:** possibilità di esportare ed importare dati da e verso Microsoft SQL Server, Access, SQLite e altri RDBMS
 - **Supporto per applicazioni MySQL Enterprise**

Workbench: Home Screen

The screenshot shows the MySQL Workbench Home Screen. The interface includes a top menu bar with 'File', 'Edit', 'View', 'Database', 'Tools', 'Scripting', and 'Help'. The main content area features a 'Welcome to MySQL Workbench' message, a description of the tool's capabilities, and three links: 'Browse Documentation >', 'Read the Blog >', and 'Discuss on the Forums >'. On the left sidebar, there is a 'Migration Wizard' icon and a 'MySQL Connections' section. The 'MySQL Connections' section lists 'Local instance MySQL56' with sub-entries for 'root' and 'localhost:3306'. Red circles and arrows highlight these elements, with corresponding text labels.

Istanze connesse ed aperte

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

[Browse Documentation >](#) [Read the Blog >](#) [Discuss on the Forums >](#)

MySQL Connections ⓘ ⓘ

Local instance MySQL56

- root
- localhost:3306

Filter connections

Migration Wizard

Istanze MySQL Server disponibili

Workbench: Connection page

The screenshot displays the MySQL Workbench interface. On the left, the 'MANAGEMENT' sidebar is circled in red and labeled 'Gestione del MySQL Server'. It includes sections for Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore, INSTANCE (Startup / Shutdown, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup), and SCHEMAS. Below this, the 'Information' sidebar is also circled in red and labeled 'Informazioni sulla sessione corrente', showing connection details for 'Local instance MySQL56'. The main workspace shows a query window with a query: `SELECT * FROM sivol1.biglietto;` and a 'Result Grid' displaying the following data:

| idvolo | data_partenza | idcliente | posto | prezzo |
|--------|---------------|-----------|-------|--------|
| 1 | 1 | 1 | NULL | 600 |
| 2 | 1 | 2 | NULL | 600 |
| 3 | 2 | 2 | NULL | 300 |
| 4 | 2 | 2 | NULL | 300 |
| 2 | 3 | 1 | NULL | 600 |
| 2 | 3 | 1 | NULL | 300 |
| NULL | NULL | NULL | NULL | NULL |

The 'Output' window at the bottom shows the execution log with three rows, each with a green checkmark, indicating successful execution of the queries. The first row shows the query execution with 8 rows returned. The second row shows a JOIN query with 3 rows returned. The third row shows the same query as the first, with 7 rows returned.

- Gestione del db Server:
 - **Server status**
 - **Connessioni con Client**
 - **Utenti e privilegi**
 - **Importazione/Esportazione dati**
 - **Start-up e Spegnimento del MySQL Server**
 - **Logs del Server** (Errori, operazioni effettuate,...)
 - **Dashboard** (Informazioni sulla rete, sulle presetazioni e l'attività del server, volumi di lettura e scrittura)

Workbench: Connection page

The screenshot shows the MySQL Workbench interface with several key components highlighted by red circles and arrows:

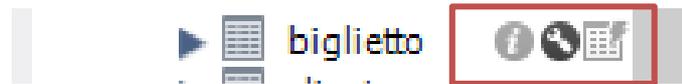
- SQL Query Menu:** A toolbar above the SQL Editor containing various icons for query execution and editing.
- SQL Editor (interrogazioni, editing DB...):** The central area where SQL queries are written and executed.
- Aiuto contestuale:** A panel on the right side providing context-sensitive help for the current query.
- Object Browser:** A tree view on the left side showing the database schema structure.
- Griglia risultato interrogazioni:** A table displaying the results of the executed query.
- Opzioni di visualizzazione del risultato:** A vertical menu on the right side of the result grid, allowing users to switch between different result display options like Result Grid, Form Editor, Field Types, and Query Stats.
- Informazioni oggetti:** A panel at the bottom left showing details about the selected object in the Object Browser.
- Cronologia interrogazioni/operazioni:** A panel at the bottom right showing a log of executed queries and their execution times.

| idvolo | data_partenza | idcliente | posto | prezzo |
|--------|---------------|-----------|-------|--------|
| 1 | 1 | 1 | NULL | 600 |
| 2 | 1 | 2 | NULL | 600 |
| 3 | 2 | 2 | NULL | 300 |
| 4 | 2 | 2 | NULL | 300 |
| 2 | 3 | 1 | NULL | 600 |
| 2 | 3 | 1 | NULL | 300 |
| NULL | NULL | NULL | NULL | NULL |

| # | Time | Action | Message | Duration / Fetch |
|---|----------|--|-------------------|-----------------------|
| 1 | 17:02:02 | SELECT * FROM sivoli.volo LIMIT 0, 1000 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 17:02:02 | SELECT V1.idvolo, V2.idvolo FROM volo as V1 JOIN volo as V2 ON V1.aeroporto_arivo=V2.aeroporto_partenza WHERE V1.aeroporto_partenza=MXP... | 3 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 17:02:09 | SELECT * FROM sivoli.biglietto LIMIT 0, 1000 | 7 row(s) returned | 0.000 sec / 0.000 sec |

Workbench: Object Browser

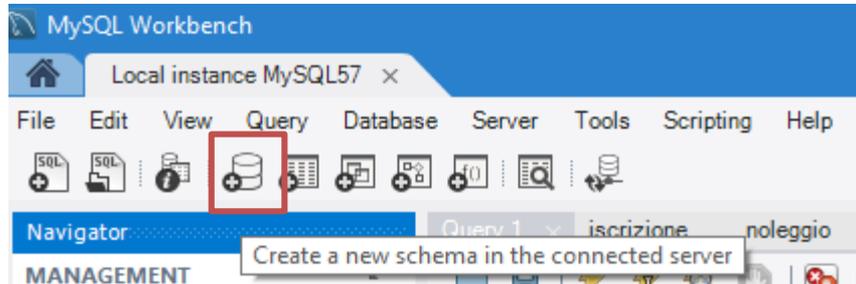
- Diverse opzioni per le tabelle



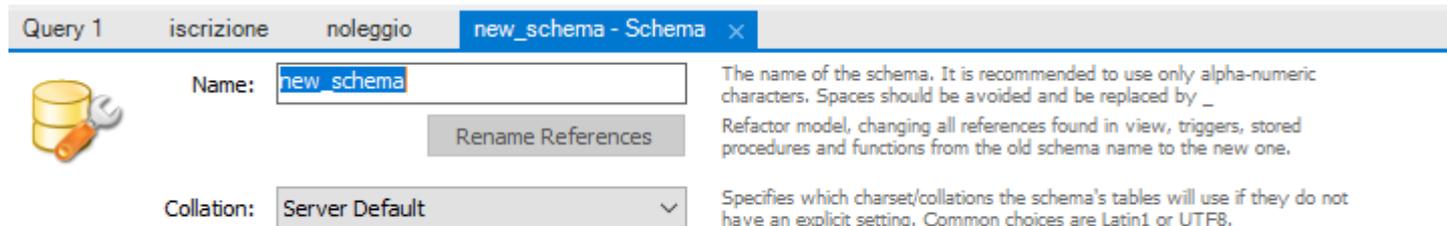
- **Info:** Apre il *Table Inspector* per la tabella specifica (mostra struttura e informazioni della tabella)
- **Modifica:** Apre il *Table Editor*
- **SELECT:** Esegue una `SELECT *` sulla tabella e mostra il risultato

Workbench: Creazione di un DB

- Cliccare sull'icona "New schema" in alto a sinistra

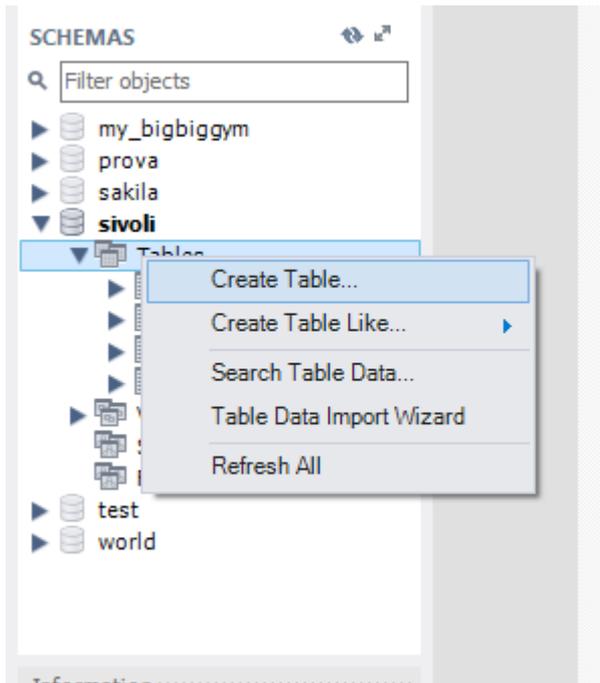


- Si apre lo "Schema Tab" in cui inserire il nome del DB: *prova*.



- Nella finestra di dialogo che compare cliccare su *Apply* e poi *Finish*.
- Il nuovo DB comparirà nell'*Object Browser* a sinistra

Workbench: Aggiunta di una tabella



- Tasto destro sul nome DB nel quale si vuole creare la tabella
 - È possibile anche creare una tabella partendo da un template: *Create Table like..*
 - Oppure importarla da un file esterno: *Table Data Import Wizard*
- Cliccando *Create Table...* si apre l'*Editor delle tabelle*

Workbench: Editor delle tabelle (Column Tab)

Table Name: ← schema: ~~civoli~~ Nome della tabella

Collation: Schema Default Engine: InnoDB

Comments:

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------|----------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| | | <input type="checkbox"/> | |

Gestione dei campi (Nome, Tipo di dato, Opzioni)

Applica/Annulla modifiche

Column Name:
Collation:
Comments:

Data Type:
Default:

Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options ← Editor Tabs

Apply Revert

- Editor tabs:

Columns Indexes Foreign Keys Triggers Partitioning Options

- **Columns:** Gestione delle colonne
- **Indexes:** Gestione degli indici
- **Foreign Keys:** Gestioni chiavi esterne
- **Triggers:** Gestione dei triggers
- **Partitioning:** Gestione partizionamento tabella
- **Options:** Altre opzioni

- Opzioni dei campi:
 - **PK** : Primary Key
 - **NN**: Not Null
 - **UQ**: Unique Index
 - **BIN**: Binary
 - **UN**: Unsigned
 - **ZF**: Zerofill
 - **AI**: Auto-Increment
 - **G**: Generated column

Workbench: Editor delle tabelle (Foreign Keys)

The screenshot shows the MySQL Workbench interface for editing foreign keys. At the top, the 'Table Name' is 'new_table', 'Schema' is 'sivoli', 'Collation' is 'Schema Default', and 'Engine' is 'InnoDB'. Below this is a table with columns 'Foreign Key Name' and 'Referenced Table'. To the right are columns 'Column' and 'Referenced Column'. Further right is the 'Foreign Key Options' section with 'On Update' and 'On Delete' dropdowns, and a 'Skip in SQL generation' checkbox. At the bottom right are 'Apply' and 'Revert' buttons. Red boxes highlight these key areas, and red arrows point from Italian text labels to them.

Table Name: new_table Schema: **sivoli**

Collation: Schema Default Engine: InnoDB

Comments:

| Foreign Key Name | Referenced Table | Column | Referenced Column | Foreign Key Options |
|------------------|------------------|--------|-------------------|---|
| | | | | On Update: On Delete: <input type="checkbox"/> Skip in SQL generation |

Foreign Key Comment

Apply Revert

Tabella esterna referenziata

Elenco delle chiavi esterne (Etichetta identificativa)

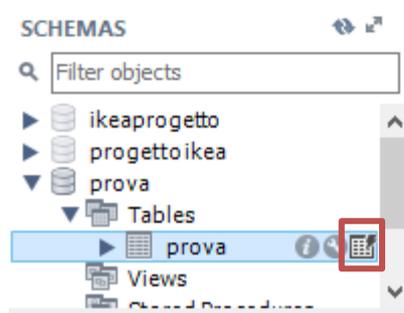
Colonna relativa alla chiave

Colonna referenziata della tabella esterna

Opzioni della chiave esterna

Workbench: Popolare le tabelle

- Cliccare sull'icona *SELECT* a fianco della tabella da popolare nell'Object Browser

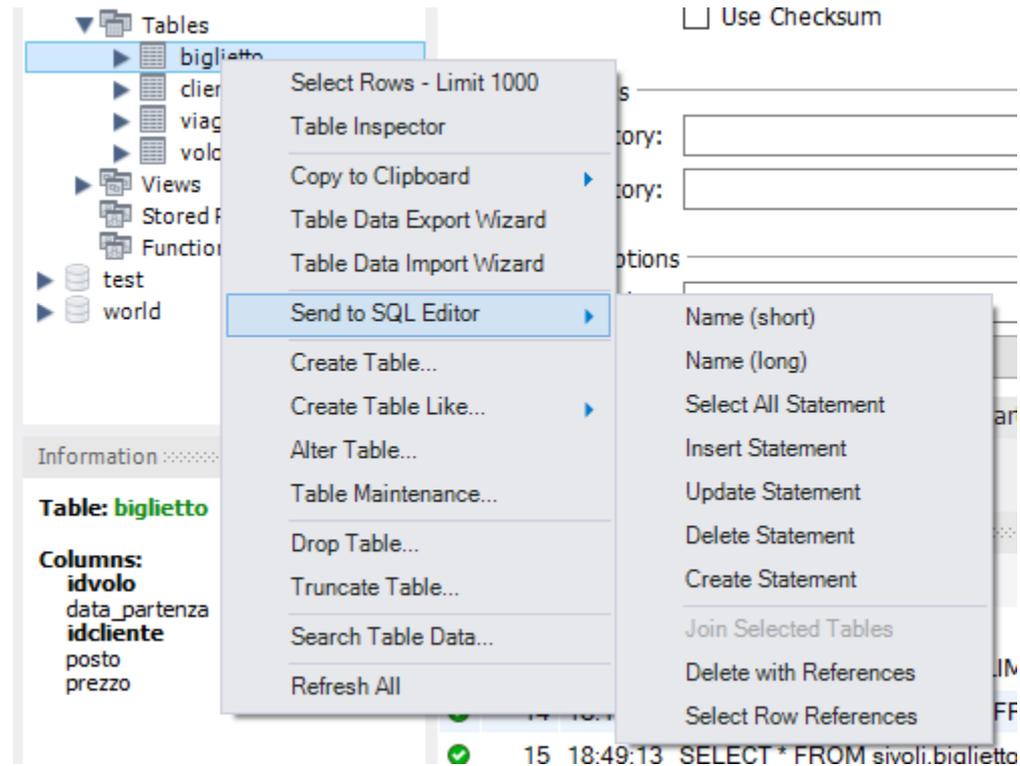


- Si apre la griglia del risultato, contenente tutti i dati di quella tabella (all'inizio sarà vuota)
- Inserire i dati necessari all'interno della griglia (attenzione alla coerenza tra le chiavi esterne e i dati già presenti in altre tabelle)
- Una volta inseriti i dati cliccare su *Apply* in basso a destra, verificare la query proposta quindi ancora *Apply* e infine *Finish*

- Buone norme per la creazione delle tabelle
 1. **Creare le tabelle**
 2. **Aggiungere le chiavi esterne**
 3. **Se tutto funziona come deve funzionare**
 4. **Popolare le tabelle (iniziando da quelle senza vincoli esterni)**

Workbench: SQL Editor

- Generare costrutti SQL



- Query Menu (icone da sinistra):



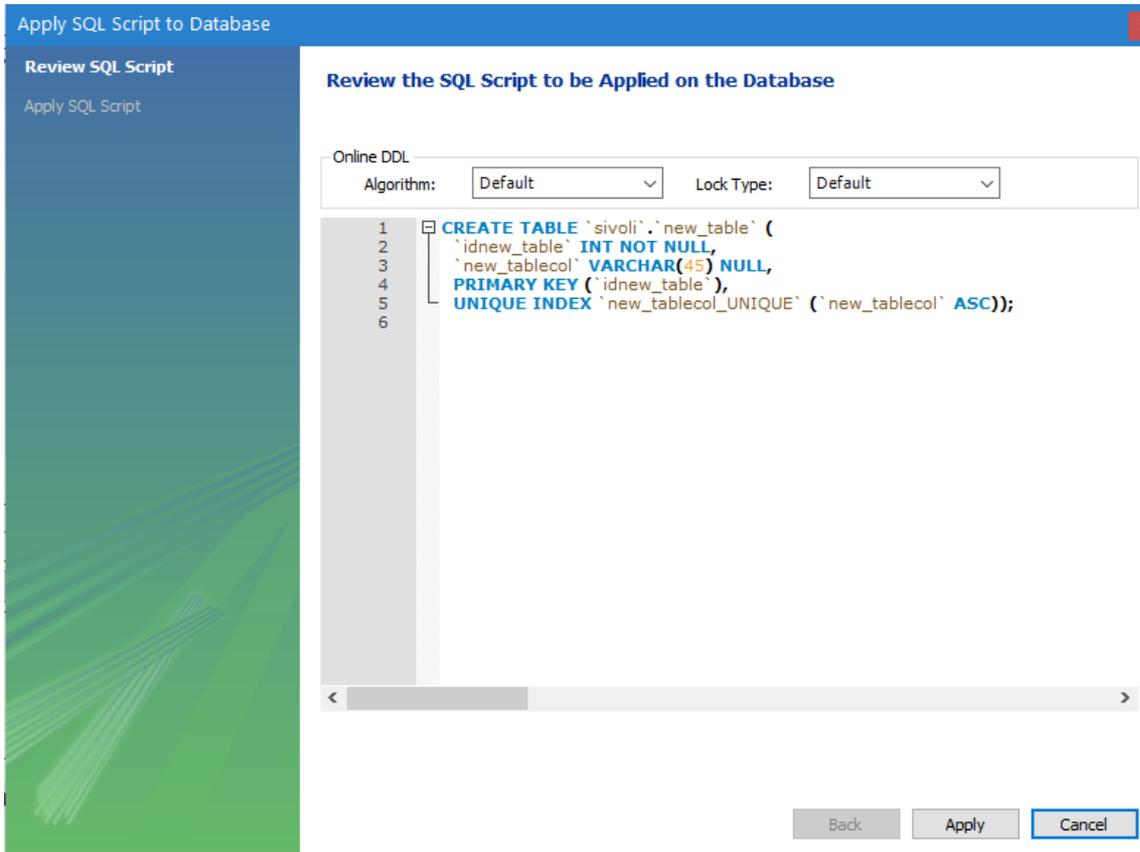
- **Aprire uno script SQL da file**
- **Salvare uno script SQL su file**
- **Eseguire script SQL:** solo una selezione o l'intera query
- **Eseguire script SQL corrente:** solo la query dove si trova il puntatore
- **Esegui comando EXPLAIN sullo script SQL:** mostra l'*execution plan* per l'esecuzione della query
- **Fermare la query in esecuzione**

- Griglia risultati (icone da sinistra):



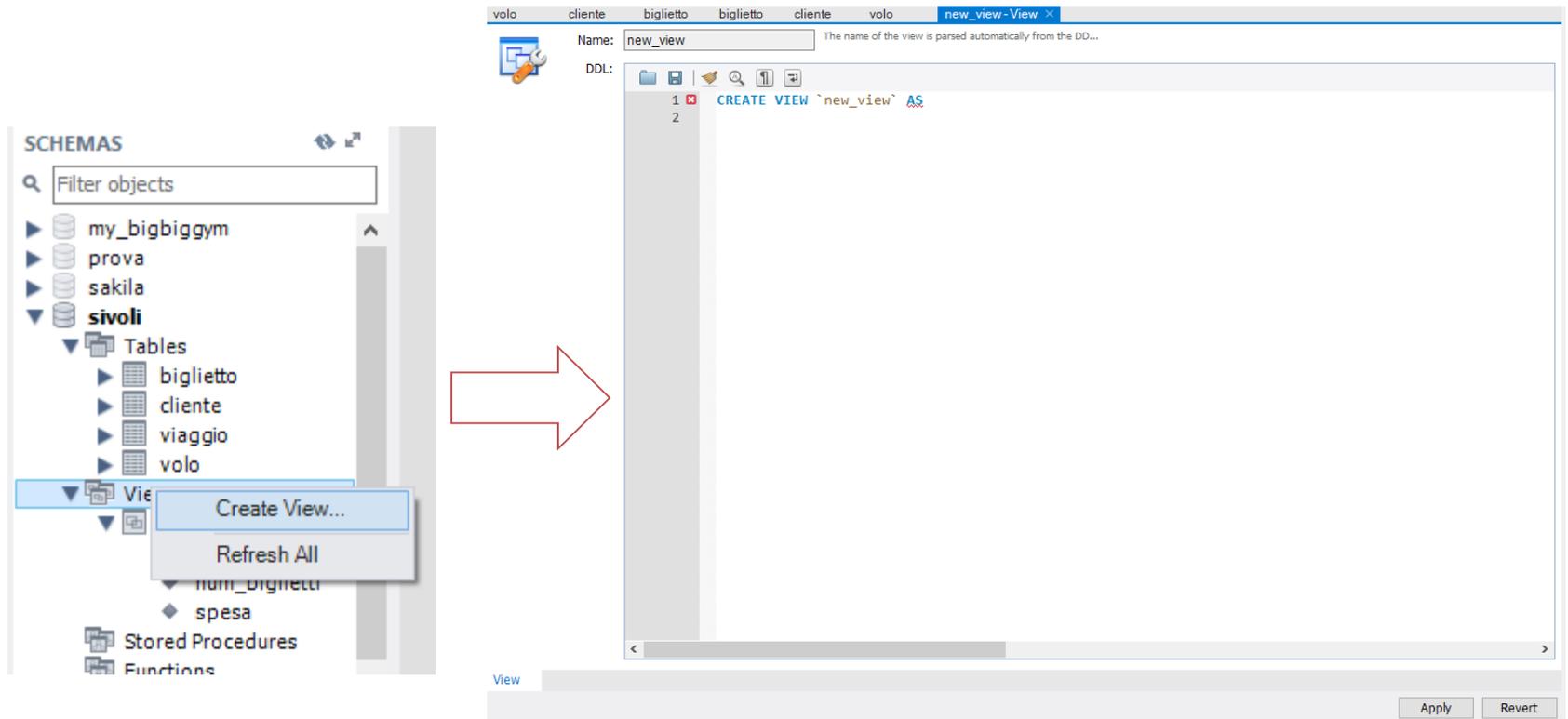
- **Reset:** azzera l'ordinamento delle colonne
- **Aggiorna**
- **Filtra righe:** ricerca case-insensitive tra tutte le celle
- **Modifica riga corrente**
- **Aggiungi riga:** aggiunge una riga vuota
- **Cancella riga selezionata**
- **Esporta:** i risultati in un file CSV, HTML, JSON o SQL
- **Importa:** da un file CSV
- **Aggiusta dimensioni cella in base al contenuto**

Workbench: Visualizzatore Script SQL



- Permette di visualizzare e modificare lo script SQL da eseguire quando si creano/modificano parametri delle tabelle o dello schema prima che vengano effettivamente applicate

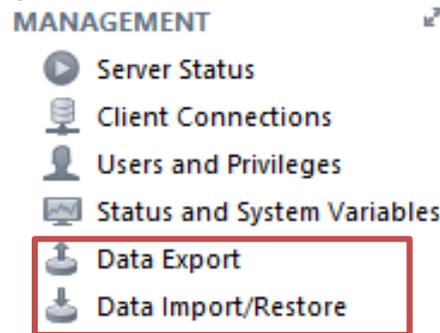
Workbench: Creazione di una View



Poi si tratta solo di scrivere la view in SQL

Workbench: Esportazione/Importazione dei dati

- **Data Export:** Esporta il corrente db, con tabelle e dati inseriti in un file *dump* .sql
 - Esportazione in una cartella dump: ogni tabella avrà un singolo file specifico
 - Esportazione in un file self-contained: tutto il db e i dati vengono salvati in un unico file.
- **Data Import/Restore:** Importa da una cartella o file *dump* .sql un db creato precedentemente



Approfondimento: CLI Client

- Aprire il *Command Line client MySQL* e digitare la password di root
OPPURE
- Aprire il terminale/shell (per OS X) e digitare *mysql*.
Se il comando non viene riconosciuto utilizzare la stringa
/usr/local/mysql/bin/mysql oppure impostare la variabile d'ambiente
PATH come spiegato qui: <https://didatticamillu.blogspot.it/2013/03/creazione-variabili-dambiente-mac-os-x.html>
- *mysql >* indica che siamo nella shell interattiva di MySQL
- Digitando *h* viene mostrato l'help con i comandi che è possibile eseguire.

```
mysql >
```

- Un comando è sempre seguito da “;”.

- Connessione e disconnessione dal Server

```
$ mysql -u user -p
Enter Password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 25338 to server version:
5.7.21-standard Type 'help;' or '\h' for help.

Type '\c' to clear the buffer.

mysql> quit
Bye
```

- Noi utilizzeremo l'utente *'root'*
- Se non viene specificata nessuna opzione si accede in modalità *anonima*

Approfondimento: CLI Client

- Verifica della versione in uso: *select version();*

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.6.21    |
+-----+
1 row in set (0,00 sec)
```

- Verificare quali databases sono presenti sul server: *show databases;*

```
mysql> show databases;
+-----+
| Database |
+-----+
| test     |
+-----+
1 row in set (0,00 sec)
```

Approfondimento: CLI Client

- Creazione di un database: *create database*

```
mysql> create database prova;  
Query OK, 1 row affected (0,00 sec)
```

- Utilizzare un database: *use*

```
mysql> use prova;  
Database changed
```

- È possibile impostare il database da utilizzare direttamente alla connessione

```
$ mysql -u root -p nome_db
```

Approfondimento: CLI Client

- Mostrare le tabelle presenti in un database: *show tables;*

```
mysql> show tables;  
...
```

- Creazione di una tabella: *create table*

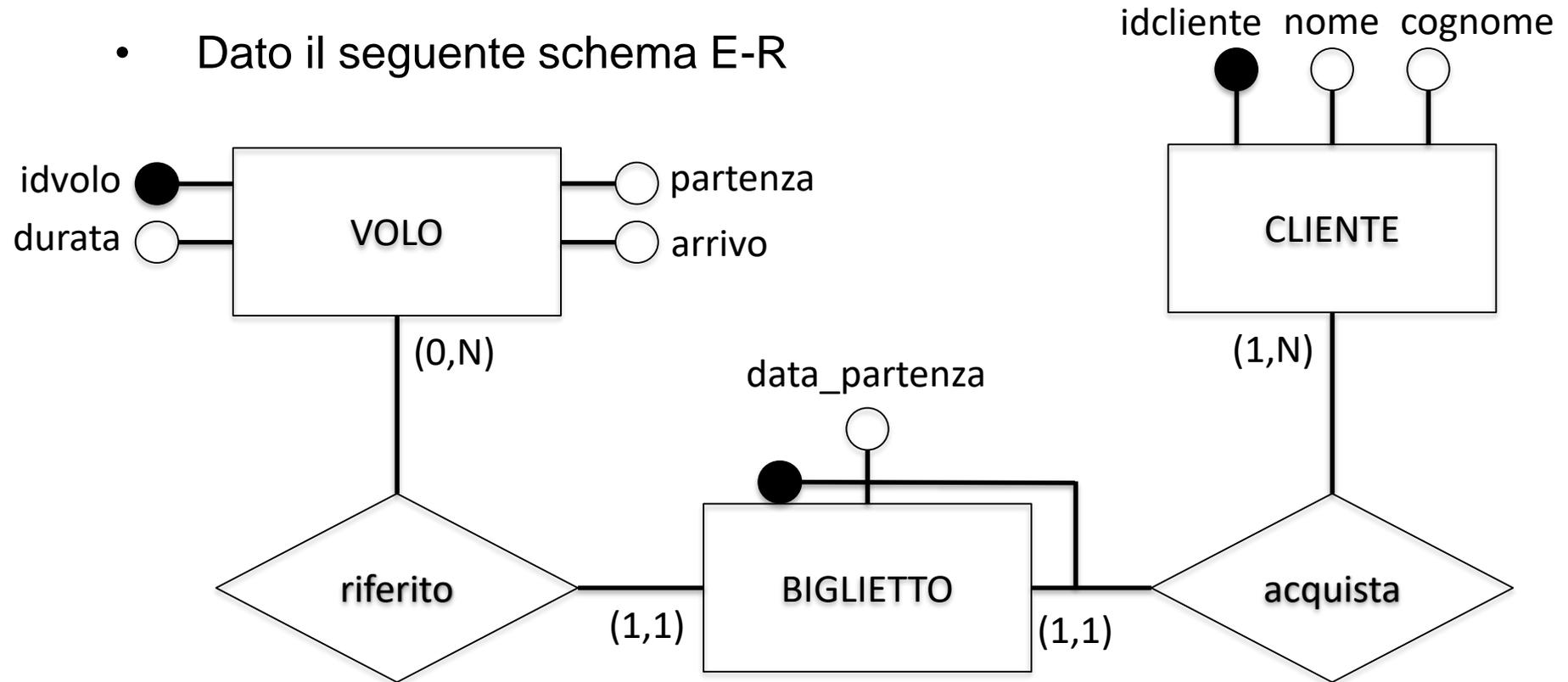
```
mysql> create table tab_name (col_name TYPE,...);
```

- Mostrare gli attribute di una tabella e le relative informazioni (chiavi, tipo, valore di default...): *describe*

```
mysql> describe tab_name;
```

Approfondimento: Creazione di un DB tramite CLI

- Dato il seguente schema E-R



Approfondimento: Creazione di tabelle tramite CLI

- Con riferimento allo schema E-R di prima, creiamo le tabelle **cliente** e **volo** secondo il seguente schema logico:

CLIENTE(idcliente, nome, cognome)

VOLO(idvolo, aeroporto_partenza, aeroporto_arrivo, durata)

- Cliente.idcliente: INT e PK
- Cliente.(nome, cognome): VARCHAR(25)
- Volo.idvolo: INT e PK
- Volo.(Aeroporto_partenza, Aeroporto_arrive): VARCHAR(3)
- Volo.durata: INT(4)

Approfondimento: Creazione di tabelle tramite CLI

```
mysql> CREATE TABLE `si`.`cliente` (  
  `idcliente` INT NOT NULL,  
  `nome` VARCHAR(25) NOT NULL,  
  `cognome` VARCHAR(25) NOT NULL,  
  PRIMARY KEY (`idcliente`)  
);
```

```
mysql> CREATE TABLE `si`.`volo` (  
  `idvolo` INT NOT NULL,  
  `aeroporto_partenza` VARCHAR(3) NOT NULL,  
  `aeroporto_arrivo` VARCHAR(3) NOT NULL,  
  `durata` INT(4) NOT NULL,  
  PRIMARY KEY (`idvolo`)  
);
```

Approfondimento: Creazione di tabelle tramite CLI

- Creiamo una tabella **biglietto** secondo il seguente schema logico:
- BIGLIETTO(data_partenza, id_cliente, id_volo, posto)
Con i seguenti vincoli di integrità referenziale:
 - Biglietto.id_volo -> Volo.idvolo
 - Biglietto.id_cliente -> Cliente.idcliente
- Biglietto.data_partenza: DATE e PK
- Biglietto.id_cliente: INT, FK e PK
- Biglietto.id_volo: INT e FK (*uguale al campo id_volo della tabella Volo*)
- Biglietto.posto: VARCHAR(3)

Approfondimento: Creazione di tabelle tramite CLI

```
mysql> CREATE TABLE `sivoli`.`biglietto` (  
  `idvolo` INT NOT NULL,  
  `data_partenza` INT NOT NULL,  
  `id_cliente` INT NOT NULL,  
  `posto` VARCHAR(3) NOT NULL,  
  PRIMARY KEY (`data_partenza`, `id_cliente`),  
);
```

Approfondimento: Aggiunta di vincoli tramite CLI

```
mysql> ALTER TABLE `biglietto`  
ADD CONSTRAINT `idvolo`  
  FOREIGN KEY (`idvolo`)  
  REFERENCES `sivoli`.`volo` (`idvolo`)  
  ON DELETE NO ACTION  
  ON UPDATE CASCADE,  
ADD CONSTRAINT `id_cliente`  
  FOREIGN KEY (`id_cliente`)  
  REFERENCES `sivoli`.`cliente` (`idcliente`)  
  ON DELETE NO ACTION  
  ON UPDATE CASCADE  
);
```

Approfondimento: CLI Client (Esercizio 1)

1. Connettersi al MySQL Server come 'root'
2. Creare il database *shell_db* dopo aver verificato che non sia già presente
3. Creare tre tabelle: *dipendente* (id, cognome, nome), *sede* (id, città), *incarico* (dip, data, *sede*, salario)
Sugg.: id e salario sono interi, data è un campo data e gli altri campi tutte stringhe.
4. Verificare la presenza delle tre tabelle

Approfondimento: CLI Client

- Inserire valori: *insert into*

```
mysql> insert into tab_name [(column1, column2,...)]  
-> values ("val1", "val2",...);
```

- Interrogazioni sulle tabelle: *select*

```
mysql> select what_to_select  
-> from table  
-> where conditions;
```

- Si utilizza SQL per specificare le interrogazioni come abbiamo visto nelle esercitazioni

Approfondimento: CLI Client

- Eliminare valori: *delete from*

```
mysql> delete from tab_name  
-> where conditions;
```

- Aggiornare valori: *update*

```
mysql> update tab_name  
-> set col_name = "value"  
-> where conditions;
```

Approfondimento: CLI Client (Esercizio 2)

1. Inserire alcuni valori nelle tre tabelle create precedentemente
2. Mostrare tutti i dipendenti
3. Mostrare tutte le sedi
4. Mostrare tutti i dipendenti che hanno avuto un incarico in una specifica sede
5. Mostrare tutti i dipendenti che lavorano o hanno lavorato in una specifica città
6. Mostrare il salario minimo per ogni sede