

# Introduzione a MySQL

---

LABORATORIO DI BASI DI DATI  
A.A. 2019/2020

Dott. Marco Savi

Contenuti riadattati a partire da slides gentilmente concesse  
dai **Dott. Paolo Napoletano** e **Claudio Venturini**

# Informazioni utili

---

## × **Docente:**

- Dott. Marco Savi
  - Assistant Professor (Ricercatore) presso DISCo
  - Email: [marco.savi@unimib.it](mailto:marco.savi@unimib.it)

## × **Materiale didattico:**

- Portale e-learning: <http://elearning.unimib.it>
- Documentazione ufficiale di MySQL:  
<http://dev.mysql.com/doc/refman/5.7/en/index.html>
- Internet!

# Informazioni utili

---

- × **Orario** dei laboratori: martedì 8.30 – 11.30
- × **Laboratorio 14A1** - U14 – In teoria! → In pratica: **lab da remoto**
- × **Date** dei 7 laboratori:
  - **7 aprile**
  - 21 aprile
  - 28 aprile
  - 5 maggio
  - 12 maggio
  - 19 maggio
  - 26 maggio

**Prova finale**  
Da definire

# Prima di incominciare...

---

- × [https://www.linkedin.com/posts/aisent srl\\_coronavirus-ai-activity-6651510627309240320-v7ic](https://www.linkedin.com/posts/aisent srl_coronavirus-ai-activity-6651510627309240320-v7ic)
- × <https://www.cybersecurity360.it/legal/privacy-dati-personali/sito-inps-violazione-della-privacy-accessibili-i-dati-altrui/>

# Obiettivo del laboratorio

---

- ✗ Far acquisire allo studente un'adeguata capacità nell'utilizzo di strumenti informatici per la creazione e gestione delle basi di dati, detti anche DBMS.
- ✗ Trasferire nella pratica i concetti presentati durante le lezioni e le esercitazioni teoriche, mostrando un esempio concreto di utilizzo di un DBMS reale di uso comune.
- ✗ Studieremo ed utilizzeremo il DBMS **MySQL** e gli strumenti di creazione e gestione ad esso correlati, come **MySQL Workbench**.



# Prova finale

---

- × Al fine di accertare le conoscenze acquisite durante il laboratorio, è prevista una prova finale che consisterà in:
  - **Progettazione concettuale e logica** di una base di dati attraverso lo strumento di data modeling integrato in MySQL Workbench
  - **Implementazione fisica** del DB progettato, produzione di **script SQL** per il deploy e **popolamento** con un insieme minimo di tuple
  - **Sviluppo di alcune query SQL** che accertano la corretta implementazione del DB
  
- × Alla prova viene assegnato un punteggio da **-1 a 3 punti**, che si somma al voto finale dell'esame del corso
  - I punti ottenuti valgono per tutti gli appelli dell'anno accademico attuale
  
- × Essendo il laboratorio svolto **eccezionalmente per via telematica**, le modalità di valutazione non sono ancora state definite nel dettaglio

# Argomenti trattati

---

- × **Introduzione a MySQL**
- × Creazione di database in MySQL
- × Data Definition Language in MySQL
- × Data Manipulation Language in MySQL
- × Creazione di viste in MySQL
- × Accesso programmatico via JDBC

# Strumenti

---

## × **MySQL**

- Il DBMS su cui implementeremo ed eseguiremo le basi di dati progettate

## × **MySQL Workbench**

- Definizione (e modellazione grafica) di database, tabelle, viste e relazioni
- Manipolazione dei dati
- Gestione dei ruoli e dei privilegi degli utenti
- Amministrazione e monitoring del DBMS

## × **MySQL Command Line client**

- Client da riga di comando che consente di eseguire tutti gli statement SQL riconosciuti da MySQL per la definizione, manipolazione e amministrazione di basi di dati



# Data Base Management Systems

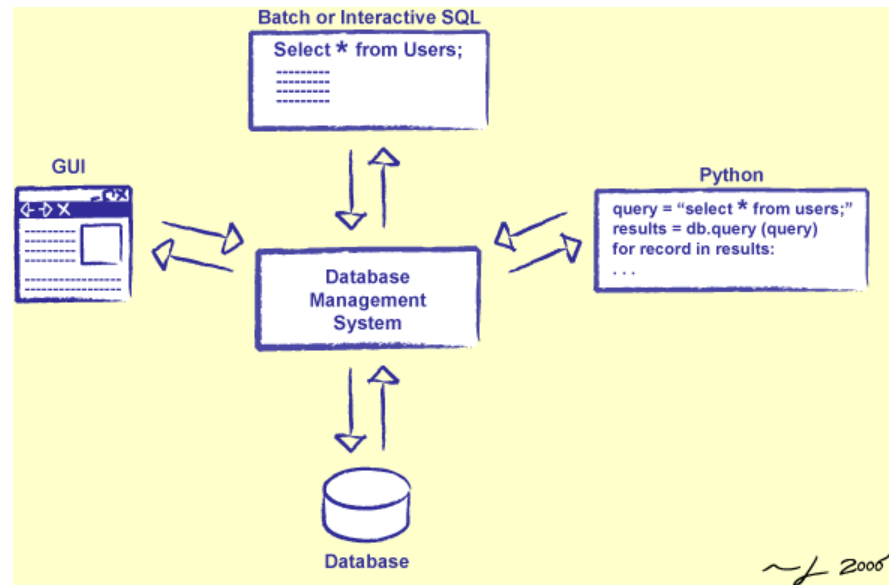
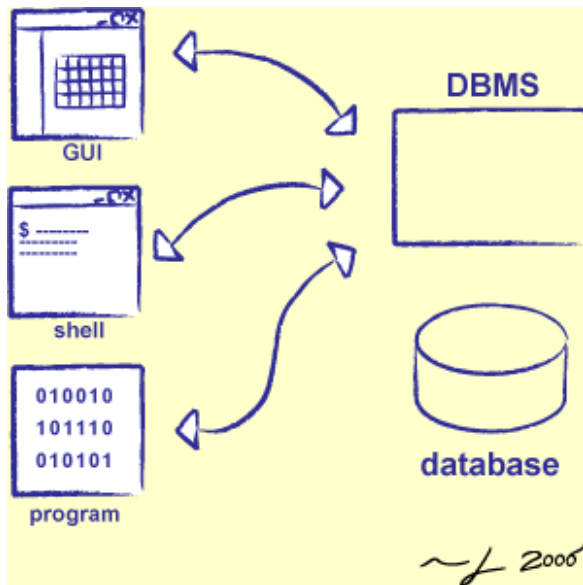
---

CARATTERISTICHE, TIPOLOGIE E ARCHITETTURA

# Data Base Management System

## × Definizione

- Un DBMS è un sistema software che facilita il processo di **definizione, costruzione e manipolazione di una base di dati**, garantendone la persistenza e consentendo l'**accesso concorrente ai dati** in essa contenuti da parte di **utenti e applicazioni**



# Tipologie di DBMS – Pre '70

---

- × **DBMS gerarchico** (hierarchical model): pre '70
  - Sviluppato in IBM durante gli anni 60'. Implementato per la prima volta nel 1968 da IBM Information Management System (IMS), il DBMS dei mainframe IBM.
  - Rappresentazione dei dati ad albero (segmento radice e segmenti figli), come in un file system
  - Rappresenta con efficacia strutture gerarchiche (ovvero con sole relazioni 1-N)
  - Non supporta relazioni M-N, che possono essere rappresentate solamente ridondando i dati
  - Ancora in uso per applicazioni specifiche (Windows Registry, GIS, ...)
  
- × **DBMS reticolare** (network model): pre '70
  - Introdotto dal CODASYL nel 1969, utilizzato dal linguaggio COBOL
  - Supera il modello gerarchico attraverso il concetto di reticolo, che permette la rappresentazione di relazioni M-N
  - Il database assume la forma di un grafo: ogni record è un nodo che può essere associato ad altri nodi tramite puntatori
  - Poco diffuso: superato dal modello relazionale

# Tipologie di DBMS – Post '70

---

- × **DBMS relazionale** (relational model) - **RDBMS**: '70
  - Struttura basata su **relazioni** (tabelle) caratterizzate da **attributi** aventi un **dominio** specifico (tipo di dato)
  - Dati organizzati in **tuple**, insieme non ordinati dei valori degli attributi
  - L'insieme di tuple di una relazione non è ordinato
  - Operazioni definite dall'**algebra relazionale**
  - Possibilità di definire **vincoli**:
    - Vincoli intra-relazionali: vincoli di dominio e vincoli di tupla
    - Vincoli di chiave
    - Vincoli inter-relazionali (chiavi esterne)
  - È il modello più diffuso (Oracle, IBM DB2, MySQL, MS SQL Server, Access, ...)
  
- × **DBMS orientato agli oggetti** (object database) - **OODBMS**: '80
  - Colmano il gap tra i RDBMS e linguaggi di programmazione ad oggetti
    - Gli oggetti utilizzati dall'applicazione possono essere salvati direttamente su memoria secondaria
    - Implementazione nativa del concetto di **ereditarietà**
  - Poco diffusi: esistono anche ibridi object-relational, come PostgreSQL

# Tipologie di DBMS – oggi

---

- × Il modello relazionale continua ad essere prevalente, ma...
- × DBMS **NoSQL**: abbandonano il modello relazionale
  - Concetto generico che racchiude vari modelli concepiti per applicazioni specifiche
  - Mirano a elevate performance per task specifici
  - Driver principali: **big data**, **social network**, **analytics**, **semantic web**
  - Maggiori dettagli nel corso di Basi di Dati Complementi
- × **Alcuni modelli** NoSQL:
  - Column store (HBase, Cassandra)
  - Document-oriented database (MongoDB, CouchDB, ...)
  - Key-value store (Redis, Riak, MemcacheDB, ...)
  - Graph database (Neo4J, Virtuoso)
  - Multi-model database (OrientDB, FoundationDB)

# Caratteristiche di un DBMS

---

- × Mantenimento della correttezza dei dati
  - Rispetto alla struttura
  - Nel tempo (persistenza)
- × Facilitare l'accesso ai dati da parte di utenti e applicazioni
- × Gestione degli accessi concorrenti ai dati
- × Controllo delle transazioni: **Transaction Processing**

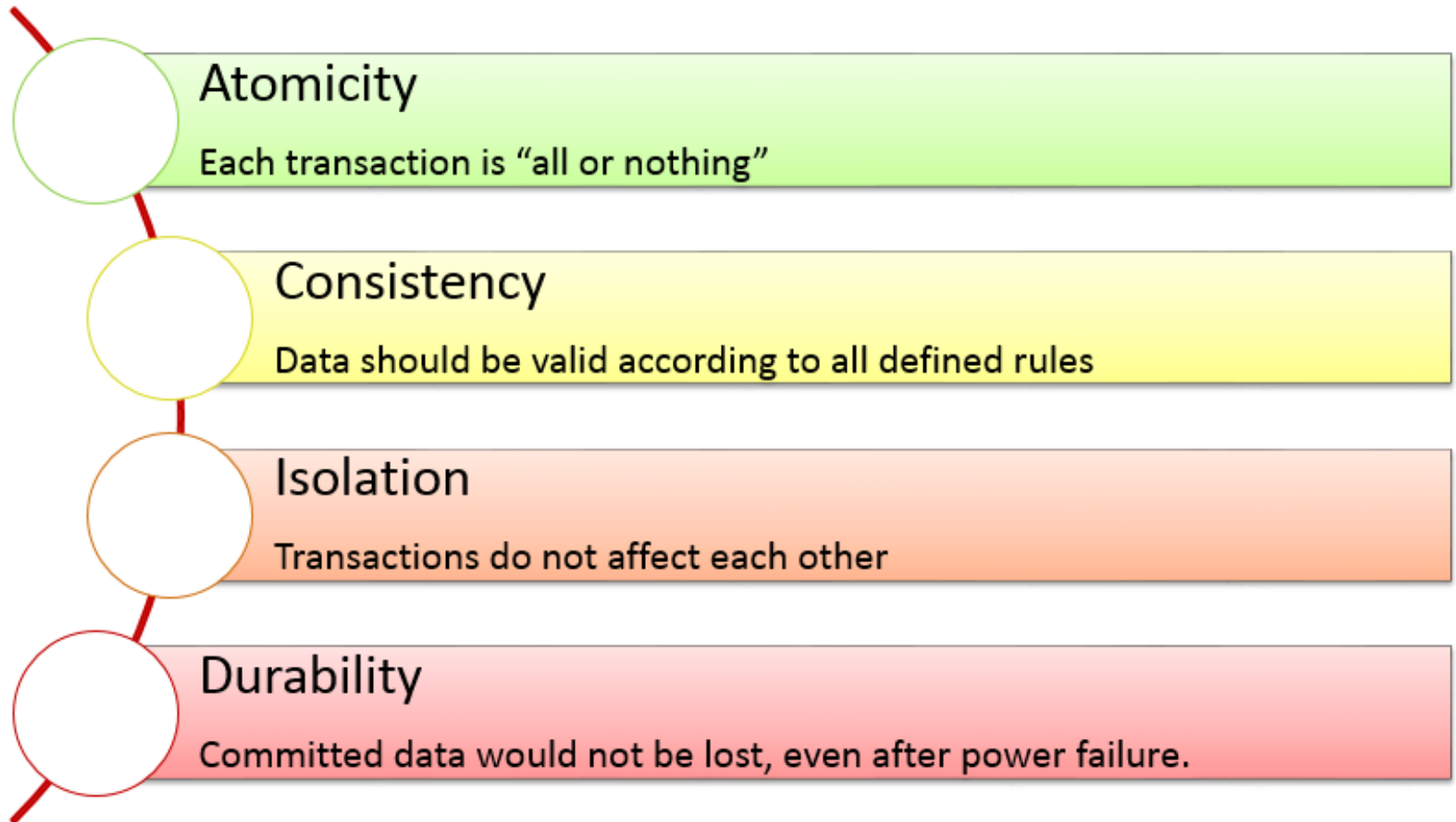
**ACID**

**Atomicity, Consistency, Isolation, Durability**

- × **Attenzione:** nella pratica esistono DBMS che non hanno tutte queste caratteristiche!

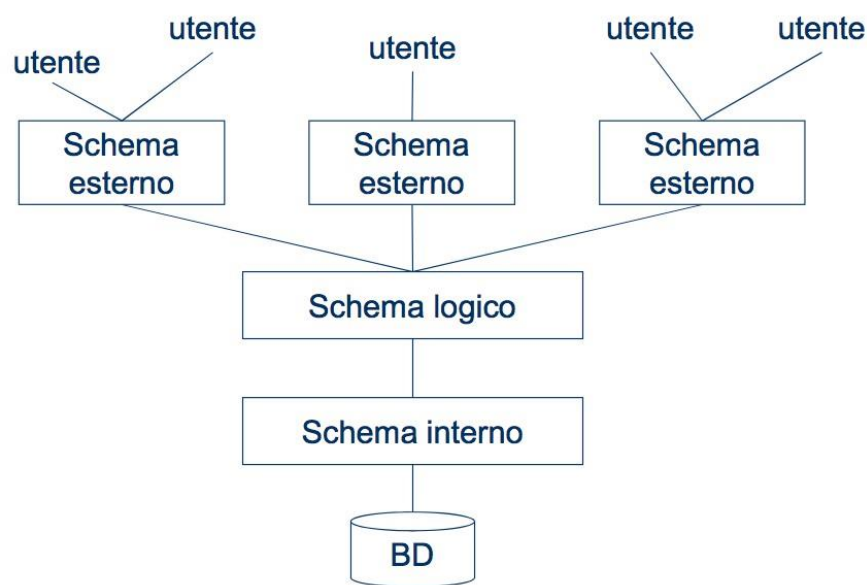
# Proprietà ACID

---



# Architettura di un DBMS

Un DBMS è un sistema software che facilita il processo di **definizione, costruzione e manipolazione di una base di dati**, garantendone la **persistenza** e consentendo l'**accesso ai dati** in essa contenuti da parte di **utenti e applicazioni**

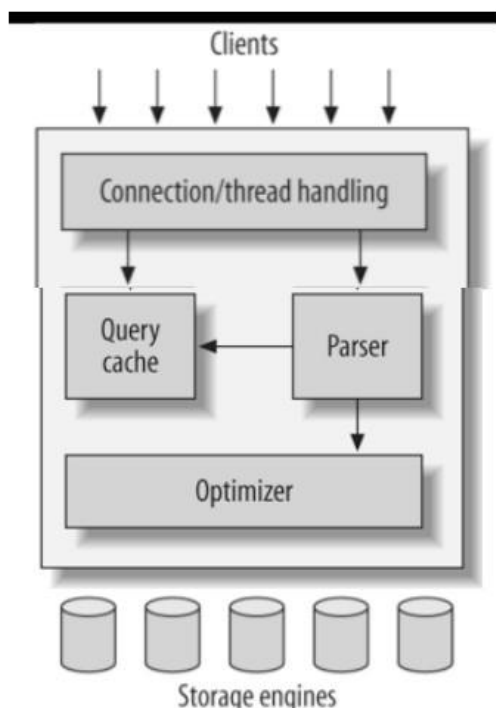


- × **Schema esterno:** porzione di schema logico visibile all'utente/applicazione
- × **Schema logico:** descrizione della struttura dei dati secondo il modello adottato dal DBMS (tabelle nel caso di un RDBMS)
- × **Schema interno:** rappresentazione degli elementi dello schema logico tramite strutture fisiche di memorizzazione (dipende dallo specifico DBMS)



# Architettura di un DBMS

Un DBMS è un sistema software che facilita il processo di **definizione, costruzione e manipolazione di una base di dati**, garantendone la **persistenza** e consentendo l'**accesso ai dati** in essa contenuti da parte di **utenti e applicazioni**



- × Parte **server** (il DBMS) e parte **client** (connettori per i linguaggi di programmazione, driver JDBC/ODBC)
- × **Alcune criticità!**
  - **Efficienza:** ottimizzazione delle richieste
  - Meccanismi per garantire:
    - l'**affidabilità** e la persistenza (*fault tolerance*)
    - il controllo di **concorrenza**
    - il controllo degli **accessi**

# Introduzione a MySQL

---

# Perché MySQL



- × È il DBMS più utilizzato in ambito web, parte anche dello stack LAMP (Linux, Apache, MySQL, PHP)
- × È il DBMS open source più utilizzato al mondo, e il 2° DBMS più utilizzato in assoluto (fonte: <http://db-engines.com/en/ranking>)
- × Ha una architettura che lo rende flessibile per diversi scopi e applicazioni
- × **Attenzione:** il DBMS perfetto non esiste! Ogni DBMS ha pro e contro, l'ampia diffusione non significa che MySQL sia adatto ad ogni esigenza!

# Cos'è MySQL

---

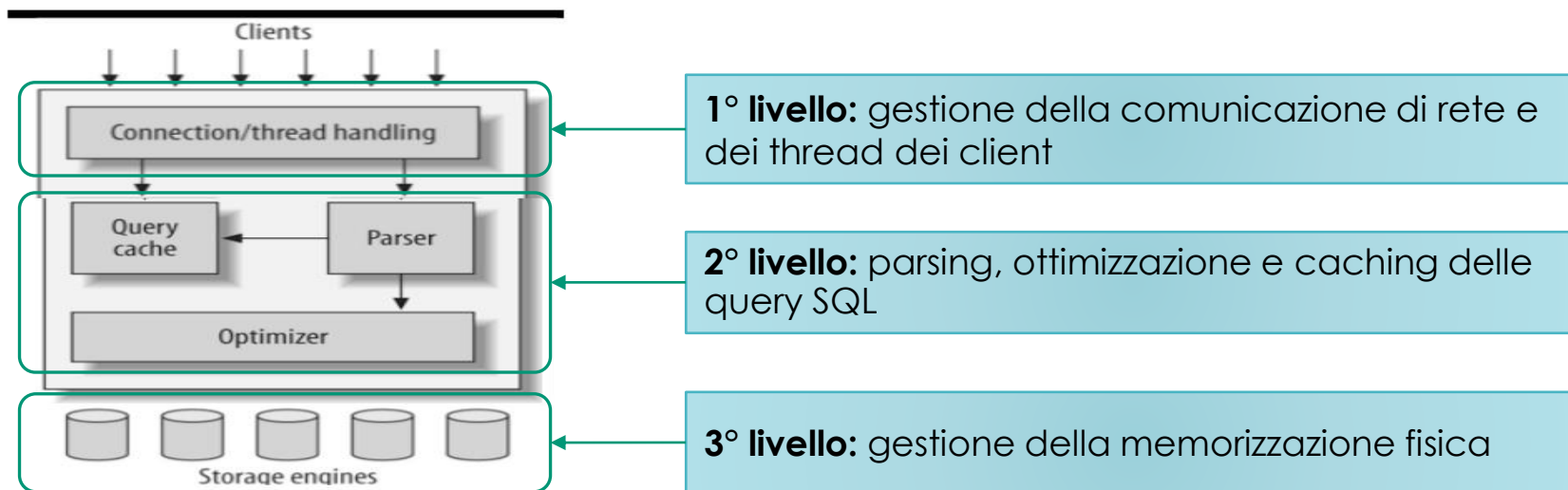
- × È un **RDBMS open source** di proprietà di Oracle
- × Due versioni:
  - **Versione Community**: distribuita con licenza GPL
  - **Versione Enterprise**: distribuita con una licenza commerciale proprietaria, aggiunge servizi di supporto e alcuni strumenti di gestione e amministrazione non open-source
- × Breve storia
  - Creato inizialmente per scopi personali da Michael "Monty" Widenius
  - 1995: prima versione rilasciata dalla società MySQL AB
  - 2001: introduzione dello storage engine InnoDB di Innobase, che supporta transazioni e proprietà ACID
  - 2005: Oracle acquisisce Innobase e intende rinegoziare le licenze per la fornitura di InnoDB. MySQL avvia lo sviluppo di uno storage engine alternativo: Falcon
  - 2008: MySQL AB viene acquisita da Sun Microsystems
  - 2010: Oracle acquisisce Sun Microsystems e MySQL AB: Falcon viene abbandonato

# Alcuni dettagli di MySQL

---

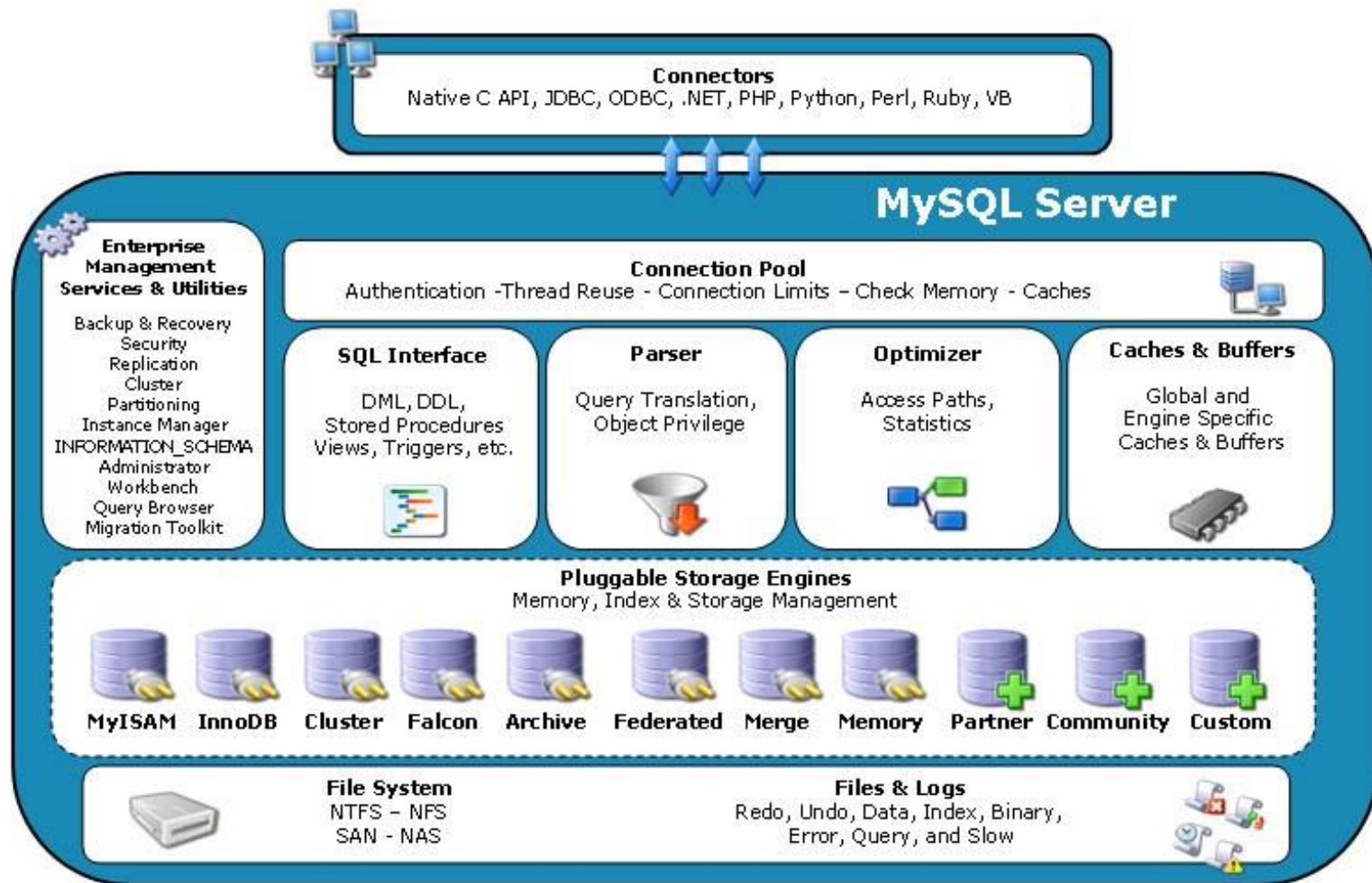
- × Supporta Linux, Windows e MacOSX
- × Supporta **standard SQL** ANSI
- × Dispone di connettori e driver ODBC (Open DataBase Connectivity), JDBC (Java DataBase Connectivity), e si interfaccia con i principali linguaggi di programmazione
  
- × Supporta **transazioni sia locali sia distribuite** (XA Transactions)
- × **ACID-compliant**: garantisce affidabilità e persistenza
  
- × Supporta inoltre:
  - Repliche master-slave (MySQL Replication)
  - Distribuzione (MySQL Cluster)
  - Federazione (Federated Tables)
  - Caching

# Architettura di MySQL



- × Un singolo **processo server** rimane in ascolto su una socket (di default sulla porta 3306) e attiva **un thread per ogni connessione** in ingresso
- × È possibile ospitare più server MySQL sullo stesso host (su porte differenti)
- × **Ogni server gestisce uno o più database** e fornisce meccanismi di **autenticazione e autorizzazione** degli utenti

# Pluggable Storage Engine Architecture



# MySQL Engines

- × MySQL viene distribuito con **8 storage engine**.
  - Possono essere **aggiunti, rimossi, attivati e disattivati a runtime**
  - Il comando `SHOW ENGINES` elenca gli engine supportati e disponibili

```
root@jeeg: ~  
mysql> show engines;  
+-----+-----+-----+-----+-----+-----+  
| Engine          | Support | Comment                                     | Transactions | XA   | Savepoints |  
+-----+-----+-----+-----+-----+-----+  
| MyISAM          | YES     | MyISAM storage engine                     | NO           | NO   | NO         |  
| MRG_MYISAM     | YES     | Collection of identical MyISAM tables     | NO           | NO   | NO         |  
| MEMORY         | YES     | Hash based, stored in memory, useful for  | NO           | NO   | NO         |  
| BLACKHOLE      | YES     | /dev/null storage engine (anything you   | NO           | NO   | NO         |  
| CSV            | YES     | CSV storage engine                       | NO           | NO   | NO         |  
| FEDERATED      | NO      | Federated MySQL storage engine           | NULL         | NULL | NULL       |  
| ARCHIVE        | YES     | Archive storage engine                   | NO           | NO   | NO         |  
| InnoDB         | DEFAULT | Supports transactions, row-level locking, | YES          | YES  | YES        |  
| PERFORMANCE_SCHEMA | YES    | Performance Schema                       | NO           | NO   | NO         |  
+-----+-----+-----+-----+-----+-----+  
9 rows in set (0.00 sec)  
mysql> █
```

- × Ognuno ha caratteristiche differenti, **la scelta dipende dalle esigenze dell'applicazione**.
- × L'engine è **definito a livello di tabella**



# Engine principali: InnoDB e MyISAM

---

## × InnoDB

- È l'engine di default a partire dalla versione 5.5
- **Transazionale**, DML **ACID-compliant**, supporto ai **vincoli di integrità referenziale**
- Consente letture e scritture concorrenti

## × MyISAM

- Era l'engine di default fino alla versione 5.1
- **Non transazionale**, **non ACID-compliant**, **non supporta vincoli di integrità referenziale**
- Elevate prestazioni in lettura, scritture concorrenti non consentite

# Altri engine

---

- × Maggiori informazioni: <http://dev.mysql.com/doc/refman/5.6/en/storage-engines.html>
- × Memory
  - Mantiene i **dati in memoria principale**: non è persistente!
  - Elevate prestazioni in lettura e scrittura
  - Utile per implementare buffer, cache, storage temporanei, in-memory processing
- × CSV
  - Memorizza i **dati in file di testo in formato CSV**
  - Non supporta gli indici
- × Archive
  - Consente la **memorizzazione compressa di grandi quantità di dati** per scopi di archiviazione
  - Non supporta gli indici
- × Merge
  - Consente di definire una tabella come **unione di tabelle MyISAM identiche**
  - Di default permette accessi read-only. Inserimenti possibili solo nella prima o nell'ultima tabella del set
- × Federated
  - Consente di **accedere a tabelle memorizzate fisicamente in un DBMS remoto** come se fossero in locale

# Strumenti (reprise...)

---

## × **Command Line Client**

- Client a riga di comando per eseguire statement DDL (Data Definition Language), DML (Data Manipulation Language) e DCL (Data Control Language)
- Varie utility di amministrazione da riga di comando
  - mysqladmin: operazioni di amministrazione del server MySQL
  - mysqlcheck: verifica dell'integrità e riparazione dei file di dati delle tabelle
  - mysqldump: backup di database
  - ... molte altre, anche per specifici engine

## × **MySQL Workbench**

- Include una interfaccia grafica per:
  - Modellazione e gestione di database MySQL
  - Amministrazione di server MySQL

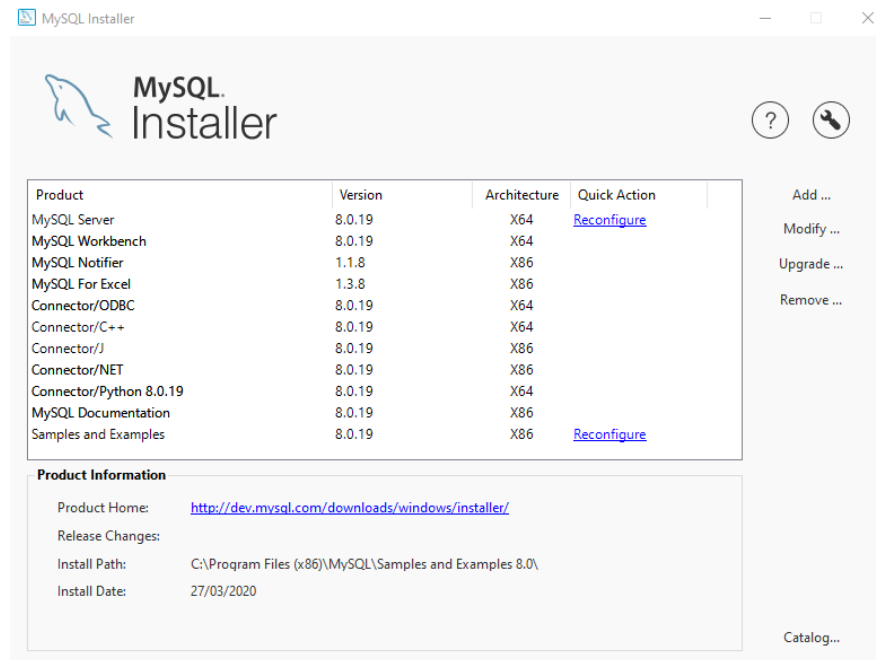
# Installare MySQL

---

- × Download (versione Community): <http://dev.mysql.com/downloads/>
  - Server, client, strumenti di gestione, e driver/connettori ufficiali per i principali linguaggi di programmazione
  - Download dei sorgenti
  
- × MySQL Installer per **Windows**: <http://dev.mysql.com/downloads/installer/>
  - Include anche MySQL Workbench e le utility per l'amministrazione e la gestione del server e dei database da riga di comando
  
- × Per **Linux** sono disponibili repository APT, Yum e SUSE
  
- × Per **Mac OSX** sono disponibili i binari di installazione (da riga di comando)

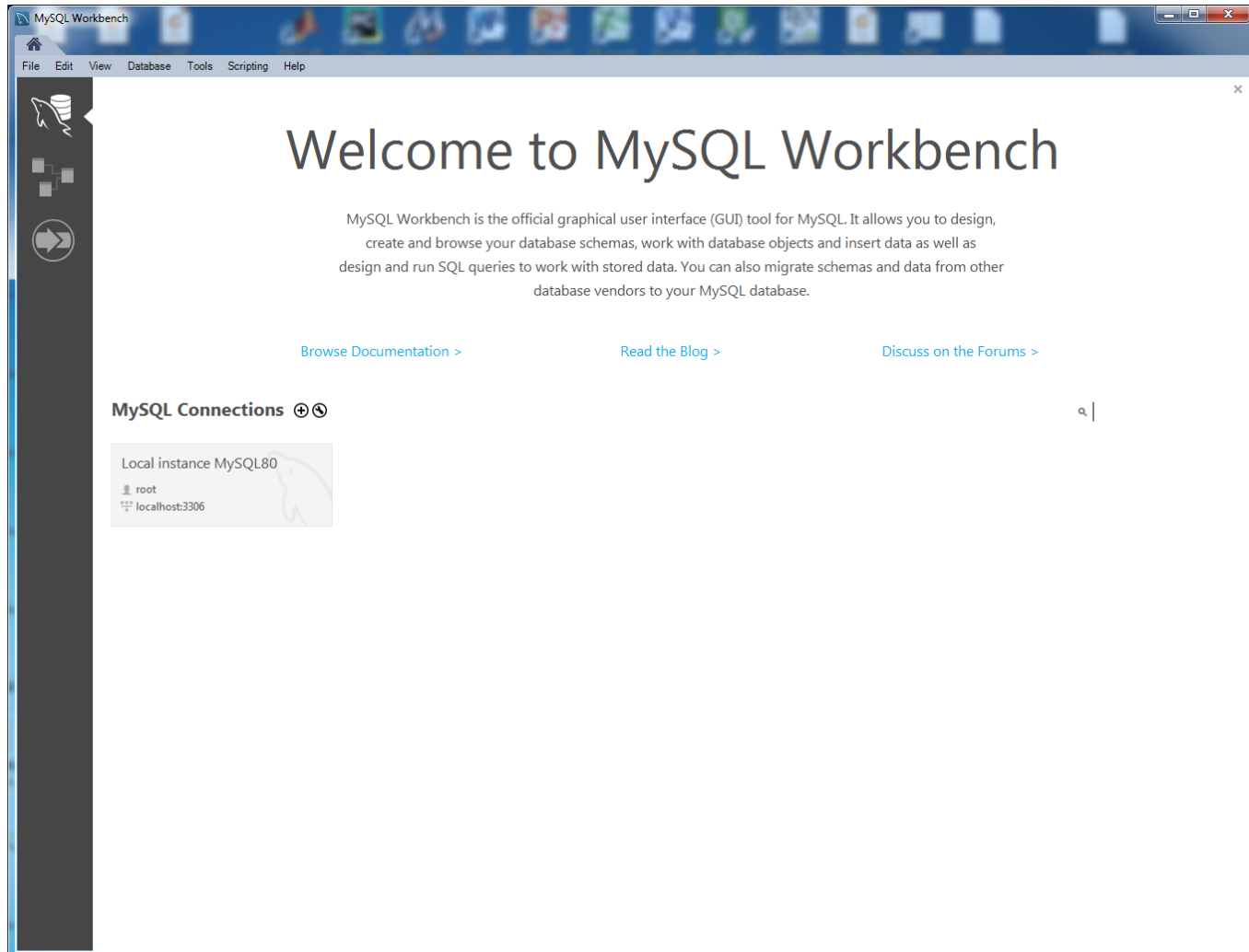
# Installare MySQL

- × Per il momento devono essere necessariamente installati i seguenti prodotti
  - MySQL Server (8.0.19)
  - MySQL Workbench (8.0.19)
- × E' possibile che dobbiate risolvere «a mano» dei problemi di dipendenza
- × Dovrete scegliere una password per accedere all'istanza locale di MySQL (MySQL80) per mezzo dell'interfaccia di loopback (localhost)



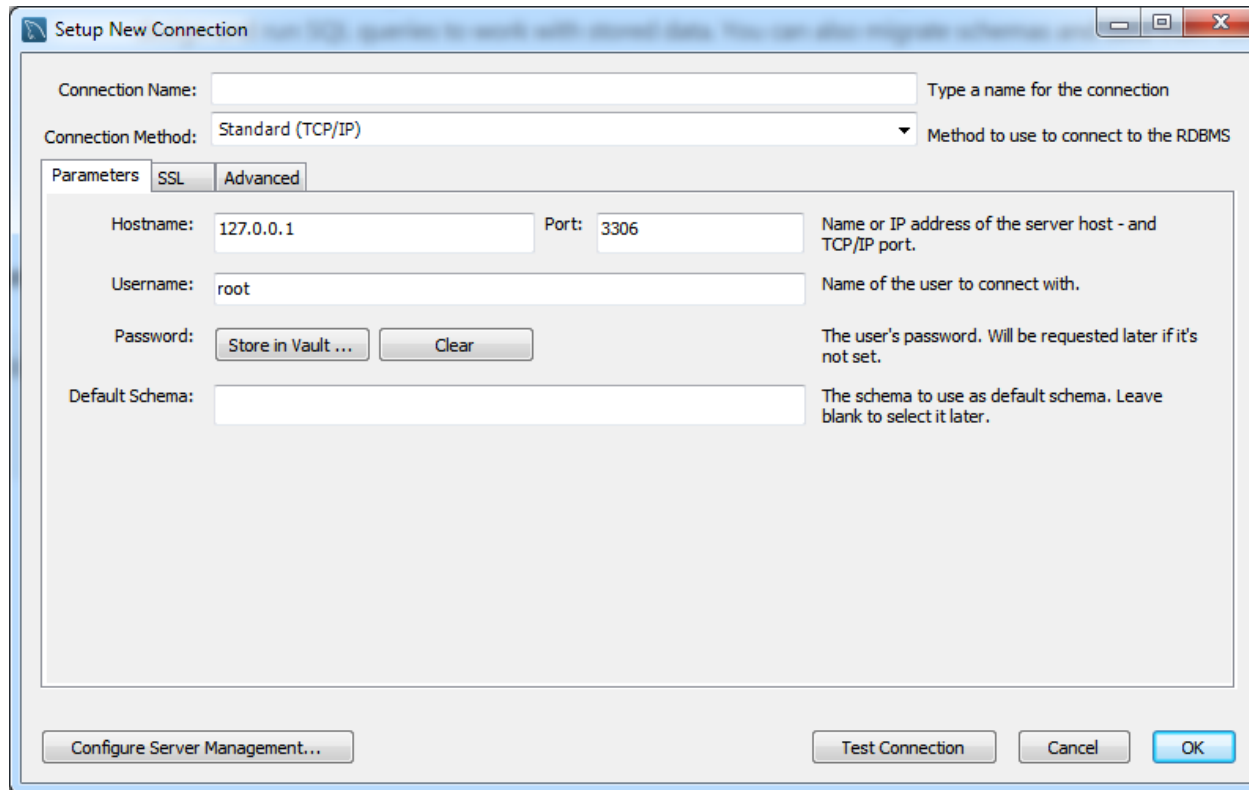
**MySQL  
installer e  
software  
installato**

# MySQL Workbench (8.0)



# MySQL Workbench (8.0)

## MySQL Connections



The screenshot shows the 'Setup New Connection' dialog box in MySQL Workbench. The window title is 'Setup New Connection'. The 'Connection Name' field is empty, with a placeholder text 'Type a name for the connection'. The 'Connection Method' is set to 'Standard (TCP/IP)'. Below this, there are three tabs: 'Parameters', 'SSL', and 'Advanced', with 'Parameters' selected. The 'Parameters' tab contains the following fields and options:

- Hostname:** 127.0.0.1
- Port:** 3306
- Username:** root
- Password:** A text field with a 'Store in Vault ...' button and a 'Clear' button.
- Default Schema:** An empty text field.

At the bottom of the dialog, there are four buttons: 'Configure Server Management...', 'Test Connection', 'Cancel', and 'OK'.

# MySQL Workbench (8.0)

## Administration

The screenshot displays the MySQL Workbench Administration - Server Status window for a local instance named 'Local instance MySQL80'. The interface is divided into several sections:

- MANAGEMENT:** Includes links for Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore.
- INSTANCE:** Includes links for Startup / Shutdown, Server Logs, and Options File.
- PERFORMANCE:** Includes links for Dashboard, Performance Reports, and Performance Schema Setup.

The main content area shows the following server details:

- Connection Name:** Local instance MySQL80
- Host:** gigi-PC
- Socket:** MySQL
- Port:** 3306
- Version:** 8.0.14 (MySQL Community Server - GPL)
- Compiled For:** Win64 (x86\_64)
- Configuration File:** C:\ProgramData\MySQL\MySQL Server 8.0\my.ini
- Running Since:** Fri Mar 22 08:33:43 2019 (2:02)

A 'Refresh' button is located below the server details.

The **Available Server Features** section shows the following settings:

Performance Schema:	<input checked="" type="radio"/> On	Windows Authentication:	<input type="radio"/> Off
Thread Pool:	<input type="radio"/> n/a	Password Validation:	<input type="radio"/> n/a
Memcached Plugin:	<input type="radio"/> n/a	Audit Log:	<input type="radio"/> n/a
Semisync Replication Plugin:	<input type="radio"/> n/a	Firewall:	<input type="radio"/> n/a
SSL Availability:	<input checked="" type="radio"/> On	Firewall Trace:	<input type="radio"/> n/a

The **Server Directories** section shows the following paths:

- Base Directory:** C:\Program Files\MySQL\MySQL Server 8.0\
- Data Directory:** C:\ProgramData\MySQL\MySQL Server 8.0\Data\
- Disk Space in Data Dir:** 77.86 GB of 232.79 GB available
- Plugins Directory:** C:\Program Files\MySQL\MySQL Server 8.0\lib\plugin\

The right-hand side of the window displays a dashboard with various performance metrics, all of which are currently showing 'No Data':

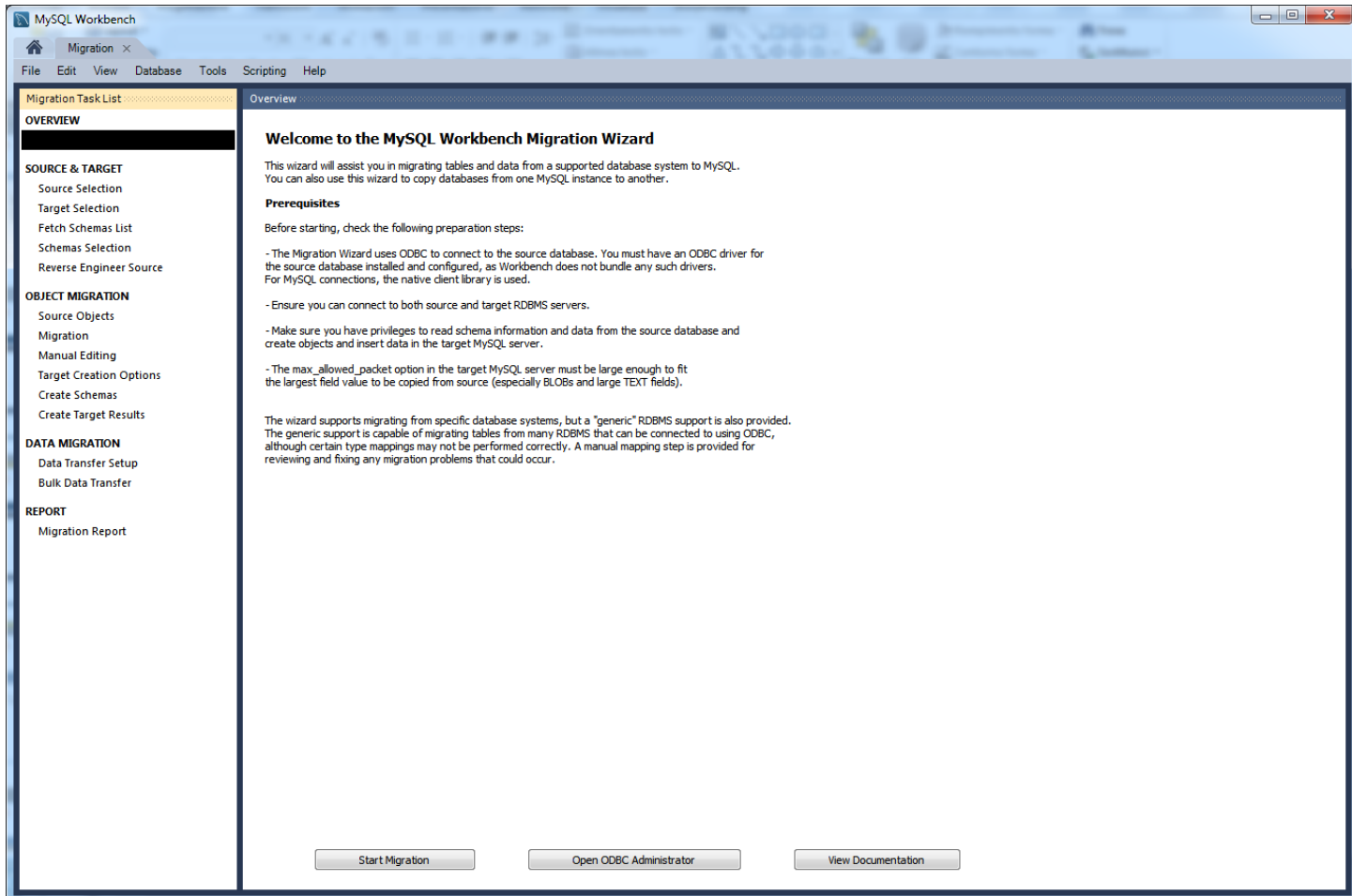
- Server Status: Running (indicated by a green play button icon)
- CPU/Load: ---
- Connections: ---
- Traffic: ---
- Key Efficiency: ---
- Selects per Second: ---
- InnoDB Buffer Usage: ---
- InnoDB Reads per Second: ---
- InnoDB Writes per Second: ---

The bottom of the window features an **Output** section with a dropdown menu set to 'Action Output' and a table with columns for '#', 'Time', 'Action', 'Message', and 'Duration / Fetch'.



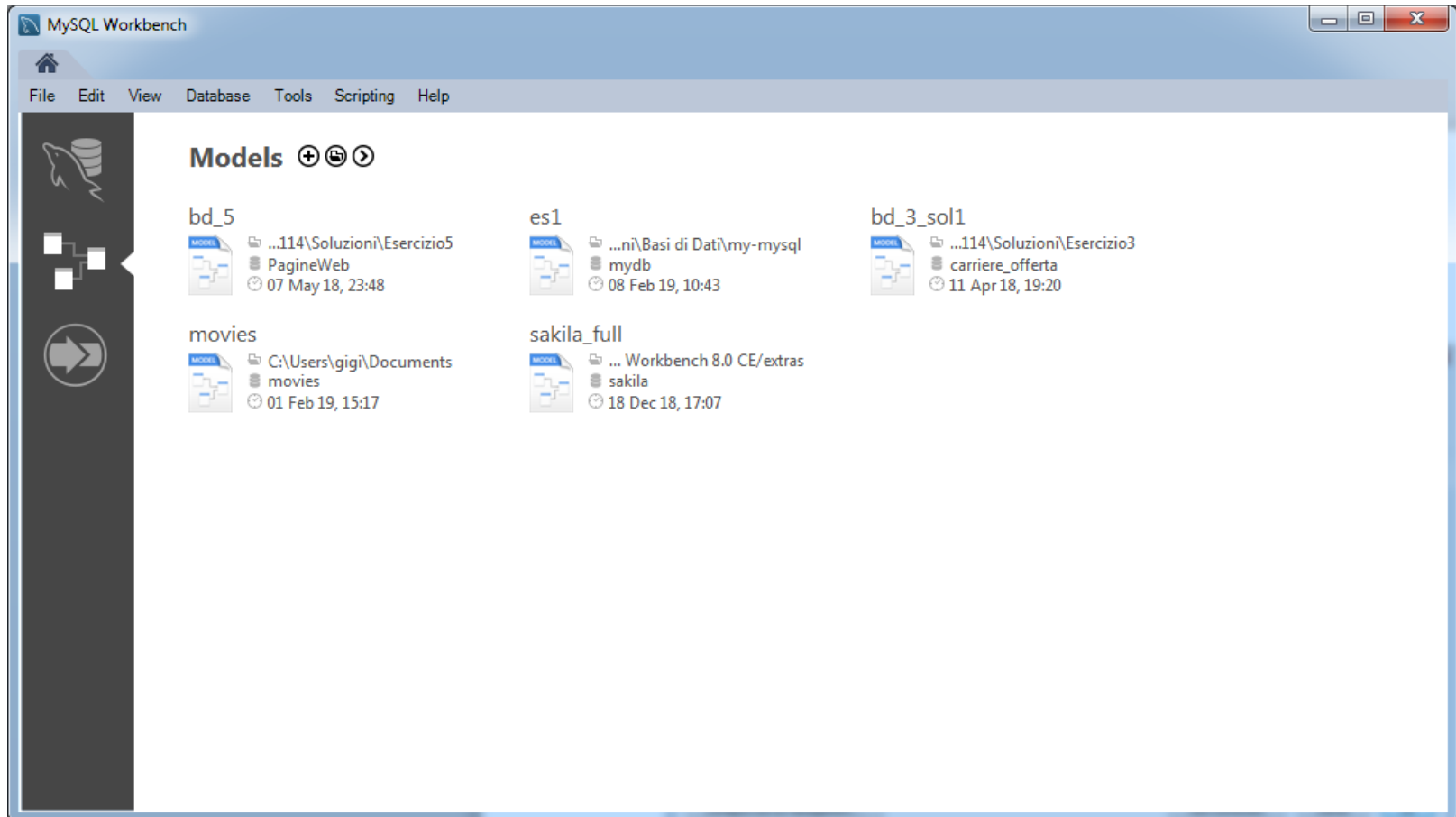
# MySQL Workbench (8.0)

## Migration Tool



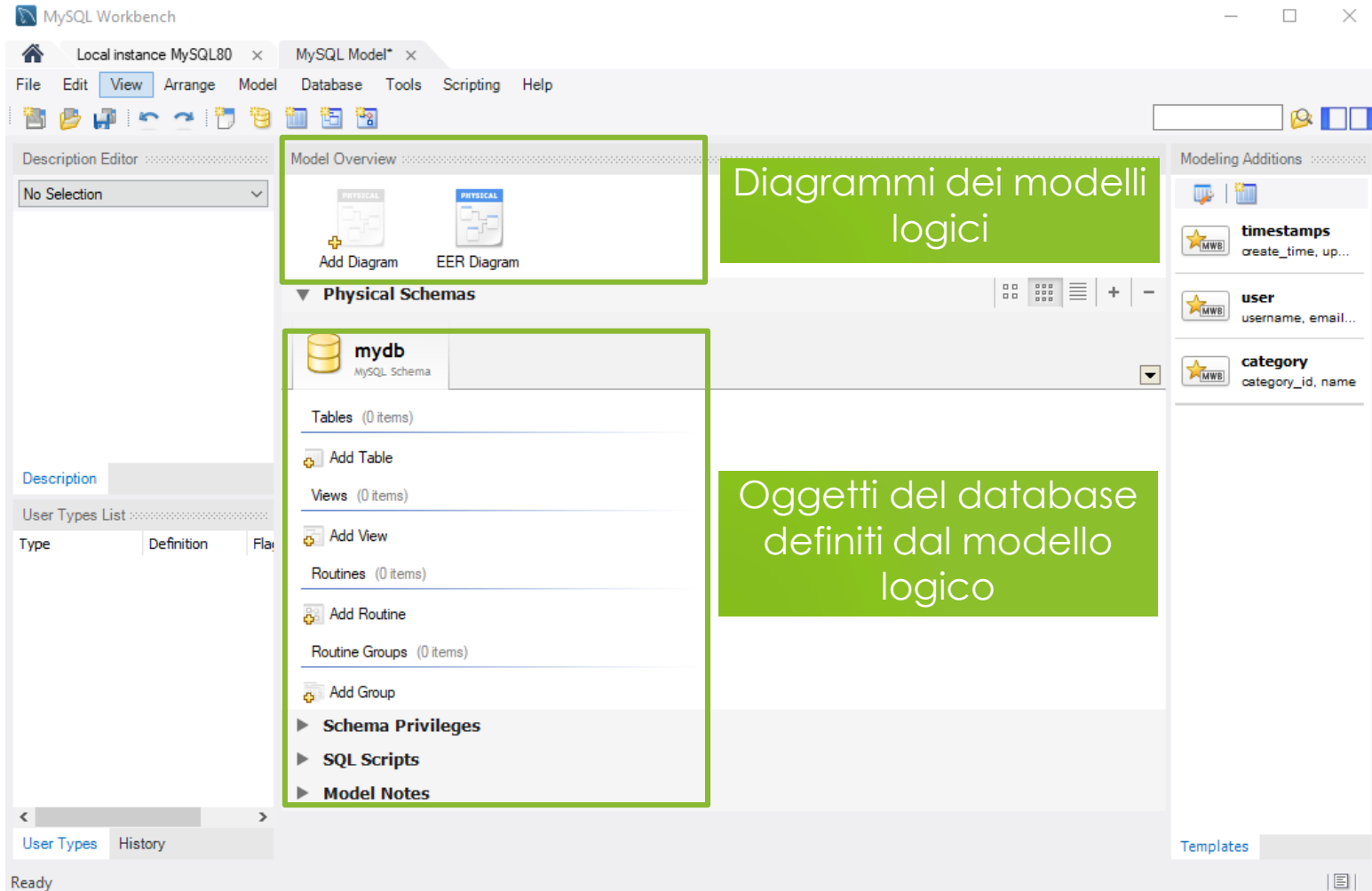
# MySQL Workbench (8.0)

## Data Modeling



# MySQL Workbench (8.0)

## Data Modeling



# MySQL Workbench (8.0)

## Data Modeling

MySQL Workbench

Local instance MySQL80 x MySQL Model\* x EER Diagram x

File Edit View Arrange Model Database Tools Scripting Help

Bird's Eye

Zoom: 100%

Diagram

Strumenti di modellazione

Modeling Additions

- timestamps  
create\_time, update\_time
- user  
username, email, passwor...
- category  
category\_id, name

table1

- idtable1 INT
- table1col VARCHAR(45)

Indexes

1:1

1:n

1:1

1:n

n:m

1:n

Description Editor

No Selection

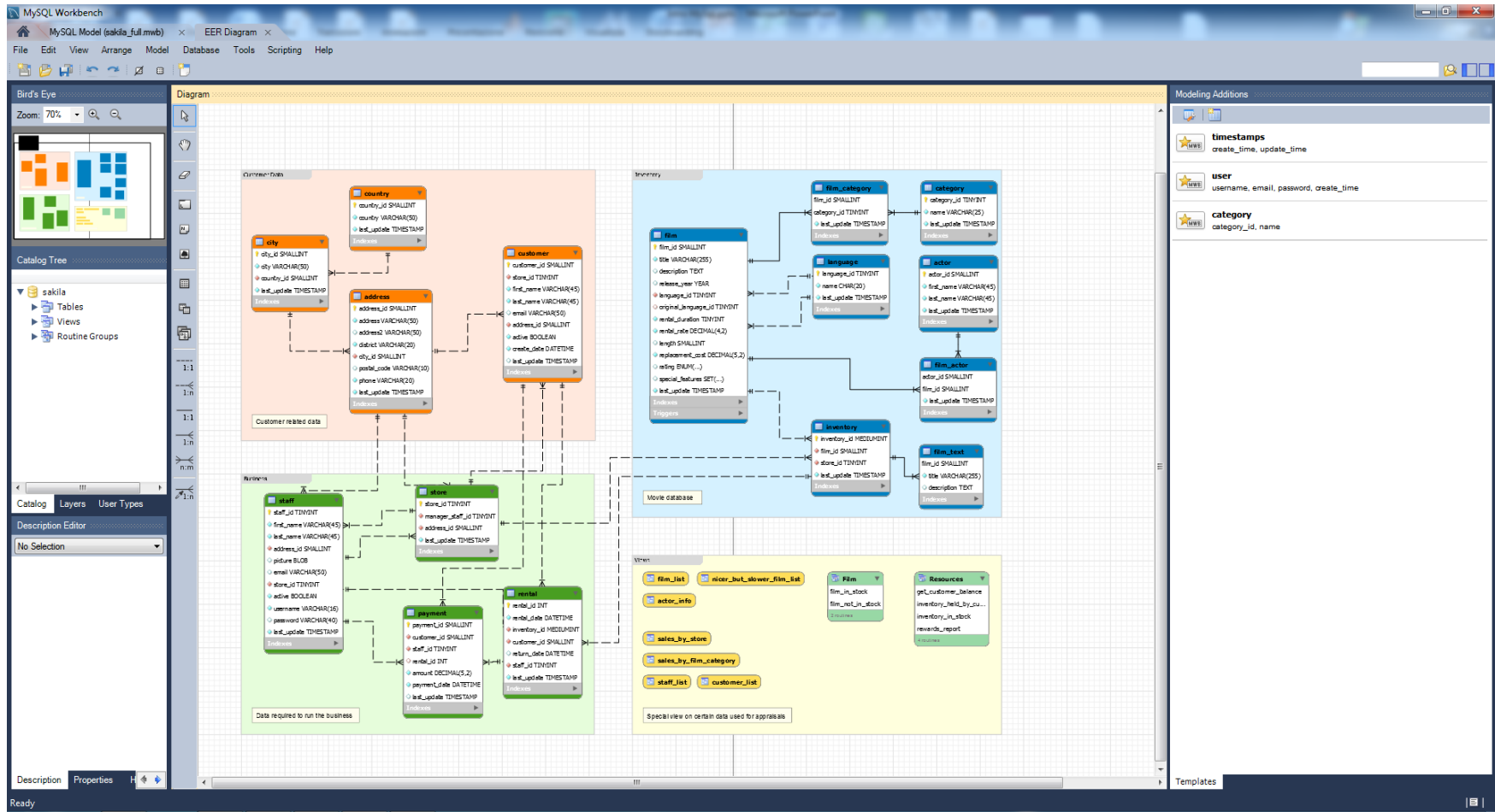
Description Properties

Placed table1

Templates

# MySQL Workbench (8.0)

## Data Modeling



# MySQL Workbench (8.0)

## Forward engineering: da modello (.mwb) a script SQL (.sql)

File → Export → Forward Engineer MySQL Create Script

**SQL Export Options**

Output SQL Script File:




Leave blank to view generated script but not save to a file.

SQL Options

- Generate DROP Statements Before Each CREATE Statement
- Generate DROP SCHEMA
- Skip Creation of FOREIGN KEYS
- Skip creation of FK Indexes as well
- Omit Schema Qualifier in Object Names
- Generate USE statements
- Generate Separate CREATE INDEX Statements
- Add SHOW WARNINGS After Every DDL Statement
- Do Not Create Users. Only Export Privileges
- Generate INSERT Statements for Tables

**SQL Object Export Filter**

To exclude objects of a specific type from the SQL Export, disable the corresponding checkbox. Press Show Filter and add objects or patterns to the ignore list to exclude them from the export.

	<input checked="" type="checkbox"/> Export MySQL Table Objects 4 Total Objects, 4 Selected	<input type="button" value="Show Filter"/>
	<input type="checkbox"/> Export MySQL View Objects 0 Total Objects, 0 Selected	<input type="button" value="Show Filter"/>
	<input type="checkbox"/> Export MySQL Routine Objects 0 Total Objects, 0 Selected	<input type="button" value="Show Filter"/>
	<input type="checkbox"/> Export MySQL Trigger Objects 0 Total Objects, 0 Selected	<input type="button" value="Show Filter"/>
	<input type="checkbox"/> Export User Objects 0 Total Objects, 0 Selected	<input type="button" value="Show Filter"/>

# MySQL Workbench (8.0)

**Reverse engineering:** da script SQL (.sql) a modello (.mwb)

File → Import → Reverse Engineer MySQL Create Script

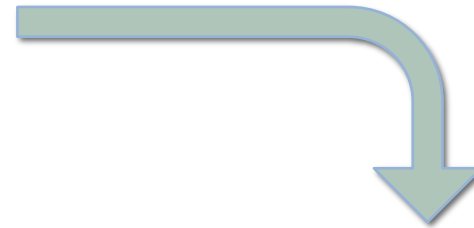
## Input and Options

### Select the script containing the schemata to reverse engineer

Select SQL script file:

File encoding:

Place imported objects on a diagram



## Reverse Engineering Progress

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

- Reverse Engineer SQL Script
- Verify Results
- Place Objects on Diagram

Import finished.

Finished parsing MySQL SQL script.

# Esercizi

---



# Esercizio 1

---

- × Progettare la seguente base di dati: **Esami**
  - Si rappresentino gli esami sostenuti dagli studenti dei corsi universitari
  - Devono essere rappresentati gli **Esami**, gli **Studenti** e i **Corsi**, con i relativi attributi
  
- × **1° parte**: progettazione
  1. Progettazione concettuale: modello ER
  2. Progettazione logica: modellare con il modello relazionale i dati rappresentati dal diagramma ER (tabelle, relazioni, e attributi, chiavi e vincoli di integrità)
  
- × **2° parte**: tramite lo strumento di modellazione di MySQL Workbench
  1. Creare un nuovo modello di database
  2. Definire le tabelle per la base di dati appena progettata

# Esercizio 2

---

- × Progettare la seguente base di dati: **Carriere Studenti**
  - I dati che riguardano gli studenti e i corsi che essi hanno superato attraverso esami.
  - Gli **studenti** sono descritti da una matricola, un nome, un cognome, il corso di Laurea a cui afferiscono.
  - Ogni **esame** è identificato da un codice e caratterizzato dal corso a cui si riferisce, dallo studente che l'ha sostenuto, dalla data con giorno, mese, anno, e dal voto ottenuto.
  - I **corsi** sono descritti mediante un codice, un nome, i **corsi di laurea** cui afferiscono (vi possono essere corsi interdipartimentali come ad esempio “Informatica”, “Teoria e tecnica della comunicazione”, ecc.) il numero di crediti di lezione, il numero di crediti di esercitazione. I corsi di laurea possono essere triennali o magistrali.
  
- × **Progettazione**
  1. Progettazione concettuale: modello ER
  2. Progettazione logica: modellare con il modello relazionale i dati rappresentati dal diagramma ER (tabelle, relazioni, e attributi, chiavi e vincoli di integrità)

# Esercizio 2

---

- × Usando MySQL Workbench, modellare le tabelle appena progettate in un nuovo diagramma
- × Generare uno script SQL di costruzione del database modellato attraverso il **forward engineering**
- × Tramite il **reverse engineering**, provare a ricostruire il modello a partire dallo script generato