

The top-left portion of the slide features a series of thin, light-brown lines that intersect to form several overlapping, irregular polygons. These lines create a complex, abstract geometric pattern that tapers towards the right.

LABORATORIO DI BASI DI DATI

Vincenzo Calabrò

Introduzione (1)

- Scopo delle lezioni in laboratorio è quello di utilizzare i concetti e i contenuti appresi nelle lezioni teoriche su DBMS reali
- DBMS (**D**ata**B**ase **M**anagement **S**ystem): sistema software general-purpose che permette agli utenti di creare e mantenere una base di dati

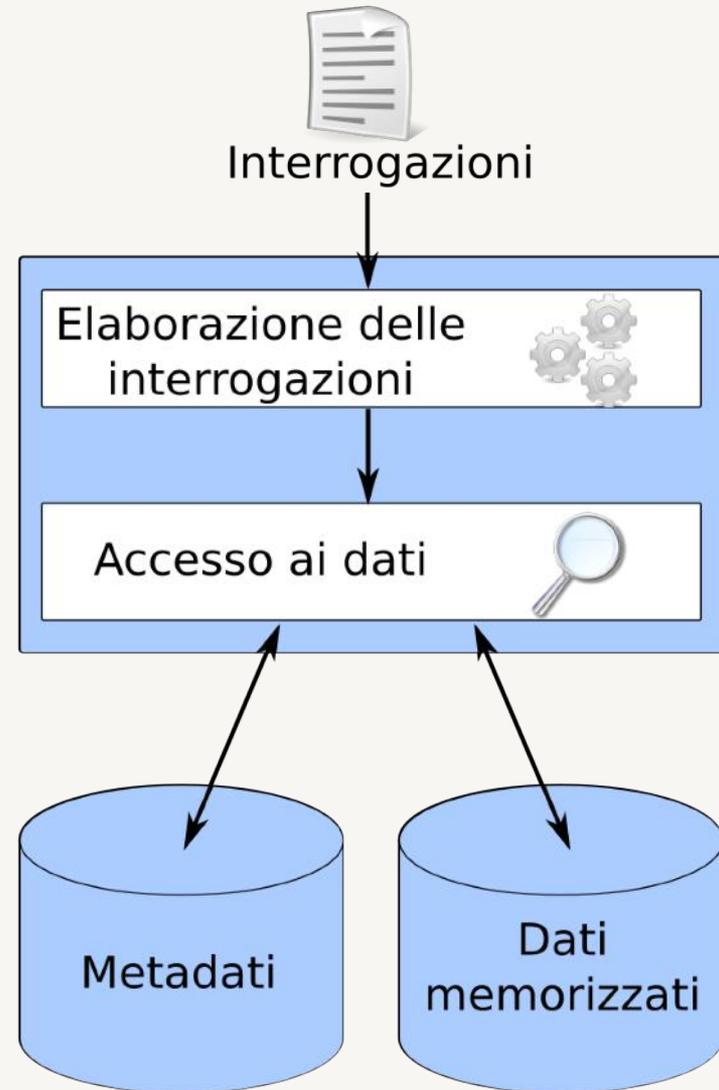
Introduzione (2)

- Un DBMS facilita il processo di **definizione, costruzione, manipolazione, e condivisione** di basi di dati (fra varie applicazioni)
 - **Definizione** implica specificare i tipi di dato, le strutture dati e i vincoli da memorizzare nella base di dati
 - **Costruzione** implica immagazzinare i dati su un mezzo di memorizzazione controllato dal DBMS (persistent storage)
 - **Manipolazione** include funzioni quali interrogazioni della base di dati, aggiornamenti, generazione di report
 - **Condividere** implica la possibilità che diversi utenti/applicazioni accedano alla stessa base di dati allo stesso tempo

Introduzione (3)

- Per gestire correttamente una base di dati, **non** è necessario usare un DBMS general-purpose
- È possibile scrivere il proprio set di programmi ad-hoc per produrre e mantenere una base di dati
 - realizzazione di un **DMBS special-purpose**
- L'insieme **base di dati + DMBS** viene chiamato **sistema di basi di dati**

Introduzione (4)



DBMS e utenti (1)

Amministratori



Utenti
(progettisti/programmatori, finali, casuali)



DBMS

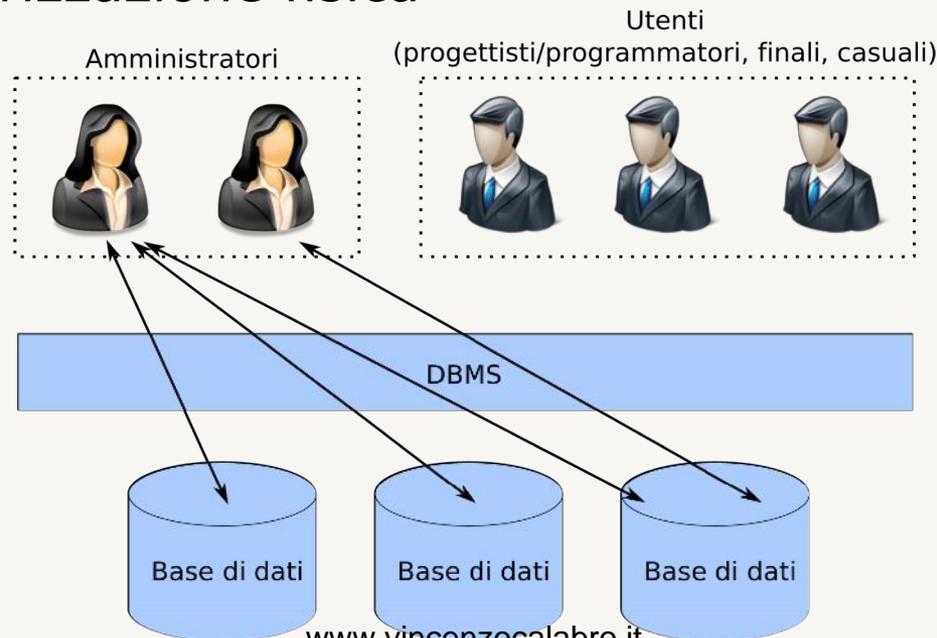
Base di dati

Base di dati

Base di dati

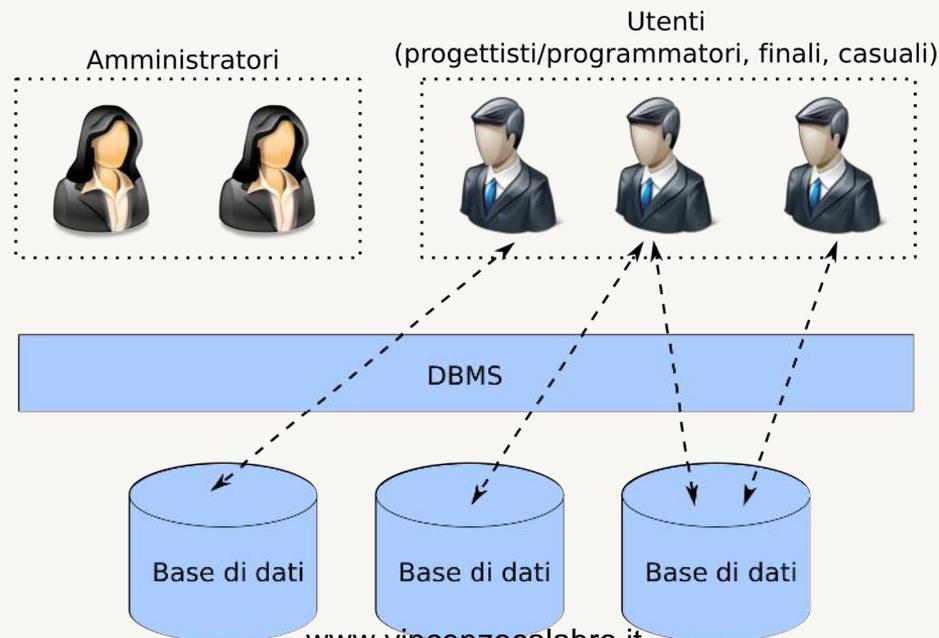
DBMS e utenti (2)

- Un **amministratore** è un utente del DBMS con **privilegi amministrativi** su **una o più** basi di dati ospitate (**superuser**: su **tutte** le basi di dati)
 - Funzionalità DDL
 - Gestione di privilegi
 - Sicurezza
 - Memorizzazione fisica



DBMS e utenti (3)

- Un **utente** (progettista/programmatore, utente finale, utente casuale) ha **privilegi specifici e limitati** su **una o più basi** di dati ospitate



DBMS e utenti (4)

- Un DBMS possiede **almeno un utente superuser** (inizialmente è l'unico utente del DBMS)
- Una base di dati possiede **almeno un utente administrator** (il proprietario – owner – della base di dati)
- I privilegi di un utente possono **variare** a seconda della base di dati

Schema o Database?

- La **relazione fra database e schema** dipende dal DBMS usato
- Esempi:
 - **PostgreSQL**: possono essere creati **più database** (CREATE DATABASE) e ogni database può contenere **più schemi** (CREATE SCHEMA)
 - **MySQL**: schemi e database sostanzialmente **coincidono** (CREATE DATABASE e CREATE SCHEMA sono sinonimi)

Architettura client/server (1)

- L'architettura di riferimento è di tipo **client/server**
- **DBMS Server**: software in esecuzione su un **elaboratore server**
 - generalmente implementato come servizio di sistema
 - configurato per accettare **connessioni locali** (pipe) e/o **connessioni remote** (TCP socket)

Architettura client/server (2)

- L'architettura di riferimento è di tipo **client/server**
- **Client**: usato dagli utenti per interagire con il server
 - client a **riga di comando**
 - client con **interfaccia grafica** (GUI)
 - client con **interfaccia web**
- Il client può essere logged sulla stessa macchina del server (**local client**), o su un'altra macchina (**remote client**)
 - local client: connessione tramite **pipe di comunicazione**
 - remote client: connessione tramite **TCP socket**

Scegliere il DMBS

- Il mercato offre un'ampia gamma di soluzioni
- Non esiste un prodotto migliore in assoluto
- La scelta deve risultare da una valutazione costi/benefici
 - anche in termini di complessità della soluzione rispetto all'ambito applicativo
- Tecnologia privilegiata al corso: **MariaDB**

DBMS Commerciali

- DBMS **commerciali**
 - MS SQL Server
 - Oracle
 - DB2
 - MySQL

- DBMS **open-source like**
 - MySQL
 - MariaDB
 - PostgreSQL
 - HSQLDB

MS SQL Server (1)

- DBMS relazionale sviluppato da [Microsoft](#)
- Sistemi operativi: [MS Windows](#), [Windows Server](#), [Linux](#)
- Cenni storici
 - 1989: [prima versione](#) per OS/2, base code acquistato da Sybase SQL Server (competizione con Oracle e IBM)
 - 2000: [SQL Server 2000](#), con performance migliori, IDE tools lato client, e diversi sistemi complementari (e.g., supporto per il data mining e OLAP)
 - 2005: [SQL Server 2005](#), con supporto nativo per XML e database mirroring
 - 2008: [SQL Server 2008](#), con full-text search
 - 2014: [SQL Server 2014](#), con supporto per Hekaton (in-memory database)
 - 2022: [SQL Server 2022](#), versione corrente

MS SQL Server (2)

- Diverse edizioni offerte



Express/web

Free, entry-level database for small web and mobile apps

Feature highlights

- Up to 16 cores of CPU
- Up to 64 GBs of memory
- In-memory OLTP and Columnstore
- End-to-end encryption with secure enclaves
- Full T-SQL surface area
- Support for Linux and Windows containers
- UTF-8 character encoding
- Data classification and auditing



Standard

Full featured database with Big Data Clusters for mid-tier applications and data marts

Feature highlights

- Up to 24 cores of CPU
- Up to 128 GBs of memory
- SQL Server 2019 Big Data Clusters
- Data virtualization through PolyBase
- Enhanced in-memory performance
- Automatic intelligent database tuning
- Azure Data Studio with notebook support

+ Express/web features



Enterprise

Mission-critical performance and intelligence for tier 1 databases

Feature highlights

- Unlimited cores of CPU
- Unlimited memory
- Industry-leading performance with unmatched scalability
- Unlimited virtualization benefits
- Petabyte scale data warehousing
- Business critical HA on Windows and Linux
- Low TCO with free DR replicas
- Access to Power BI Report Server

+ Standard features

+ Express/web features



Developer

Free to use with all the features of Enterprise Edition specifically for dev/test in non-production environments

- ORDBMS (DBMS relazionale e ad oggetti) sviluppato da [Oracle Corporation](#)
- Compatibile con diversi sistemi operativi
- Cenni storici
 - 1979: Oracle V2, supporto per funzionalità base di SQL
 - 1985: Oracle V5, supporto per interazioni client-server
 - 1992: Oracle V7, supporto per [stored procedure](#) e [trigger](#)
 - 1999: Oracle 8*i*, orientato alle interazioni tramite [Internet](#)
 - 2003: Oracle 10*g*, supporto per [grid](#) computing
 - 2013: Oracle 12*c*, supporto per [cloud](#) computing
 - 2023: [Oracle 23c](#), [versione corrente](#)

- Disponibile in diverse edizioni, fra cui:
 - **Express Edition**: versione entry level e limitata a 4GB di dati utente, 1GB di memoria, 1 CPU
 - **Standard Edition One**
 - **Standard Edition**: funzionalità di SE One + supporto per macchine più grandi e **clustering di servizi**
 - **Enterprise Edition**: tutte le funzionalità di Oracle Database + supporto per applicazioni di **online transaction processing (OLTP) ad alto volume, query-intensive data warehouses, applicazioni Internet onerose**
 - **Personal Edition**: quasi tutte le funzionalità di EE, limitato a **single-user development**

- DBMS relazionale sviluppato da **IBM**
- Compatibile con diversi sistemi operativi (fino al 1990 solo mainframe IBM)
- Cenni storici
 - 1983: **prima versione** di DB2, basata su un prodotto precedente di IBM, QBE
 - 2006: **DB2 9** che supporta XML
 - 2007: **DB2 9.5** orientato ad ambienti distribuiti
 - 2009: **DB2 9.7**, supporto alla frammentazione dei dati
 - 2013: **DB2 10.5**, migliorie di performance e funzionalità mobile
 - Versione corrente: **DB2 13** (e.g., supporto per AI)

- Disponibile in diverse edizioni, fra cui:
 - **Workgroup Server Edition (WSE)**: limits on processor and memory
 - **Enterprise Server Edition (ESE)**: funzionalità di WSE + materialized query tables, no limiti su processori, memorie, o dimensioni dei dati
 - **Advanced Enterprise Server Edition (AESE)**: funzionalità di ESE + column organized tables, in-memory database, data compression, workload management, replication, ...
 - **Advanced Workgroup Server Edition (AWSE)**: simile ad AESE, ma con limiti su processori e memoria
 - **Developer Edition**: supporto per un unico application developer

- DBMS relazionale
 - sviluppato in origine da [MySQL AB](#), ora parte di [Oracle](#)
 - disponibile con licenza GNU General Public License (alcune edizioni a pagamento per scopi commerciali)
- Compatibile con diversi sistemi operativi, molto diffuso per lo sviluppo di Web Application
- Cenni storici
 - Sviluppato a partire dal 1995 da Michael Widenius e David Axmark
 - 1998: prima versione per Windows
 - 2005: MySQL V5, supporto a viste e trigger
 - 2008: MySQL V5.1, supporto a frammentazione
 - Versione corrente: [MySQL 8](#)

- Disponibile in diverse edizioni, fra cui:
 - MySQL Standard Edition
 - MySQL Enterprise Edition: funzionalità standard +altre (e.g., sicurezza, monitor di performance)
 - MySQL Cluster Carrier Grade Edition (CGE): funzionalità enterprise +altre (e.g., storage engine addizionali, partizionamento automatico dei dati (*sharding*) con bilanciamento del carico)
 - MySQL Community Edition: licenza GNU

- "Cugino" molto stretto di MySQL
- DBMS nato nel 2009 come **fork** di **MySQL**
 - in seguito all'acquisizione da parte di Oracle
 - idea: garantire SW **free e open source** con licenza GNU
- Il **versioning** ha seguito il versioning di **MySQL** **fino alla versione 5.5**, successivamente nasce un versioning separato
 - nel 2012 si passa dalla versione 5.5 alla 10
 - versione corrente: 11

PostgreSQL (1)

- ORDBMS (relazionale e ad oggetti) rilasciato con licenza [PostgreSQL](#)
 - gratuito ed open-source
- Compatibile con diversi sistemi operativi
- Cenni storici
 - Nasce da un progetto condotto all'università della California@Berkley ([Ingres project](#))
 - 1988: viene rilasciato il primo prototipo
 - 1995: primo rilascio di [Postgres95](#)
 - 2000: supporto per chiavi esterne e sintassi SQL92 per i join
 - 2010-oggi (v15): miglioramenti a gestione di indici, trigger e viste, viste materializzate, meccanismi di lock avanzati, supporto JSON, ...

- Supporta molti dei costrutti previsti dallo standard
 - Trigger
 - Frammentazione
 - Replicazione
 - ...
- È fra i DBMS open-source più utilizzati
 - Facilmente estendibile in Python e Java
 - In continua evoluzione
- Supporta un linguaggio procedurale ([PL/pgSQL](#))

- Sistemi realizzati allo scopo di gestire **grosse moli** di dati per analizzarli in modo **aggregato**
- Strumento di **supporto alle decisioni**
- Non sono DBMS
 - svolgono un **ruolo** diverso
 - operano su dati **organizzati** in modo diverso
- Alcuni DBMS commerciali hanno di recente **integrato** funzioni OLAP

DBMS non relazionali

- Object-oriented

- I dati sono rappresentati sotto forma di oggetti
- Interesse negli anni '80 (molti DBMS commerciali hanno introdotto il supporto per gli oggetti)

- XML

- Gestione di dati semi-strutturati
- Molti DBMS commerciali ne hanno introdotto il supporto

- NoSQL

- Generalmente non hanno schema fisso, non supportano i join, non garantiscono le proprietà ACIDe
- Generalmente sono altamente scalabili e privilegiano l'accesso rapido ai dati alla consistenza

Il Web è:

“un sistema di risorse basati sul concetto di iper-testo collegati tra di loro su Internet”



Si basa su 4 semplici idee

1. Iper testo

un **formato** di documento che permette di **spostarsi** da un'informazione ad un'altra attraverso connessioni (**hyperlink**)

- permette di definire delle zone attive (e.g. parole chiave) che possono essere utilizzate sia per spostarsi all'interno di uno stesso documento che in altri documenti

2. Nome univoco per le risorse

affinché un documento sia reperibile su Internet è necessario che abbia un **nome univoco** (**U**niform **R**esource **I**dentifier)

- ogni informazione deve essere facilmente raggiungibile (posizione univoca)
- il W3C (**W**orld **W**ide **W**eb **C**onsortium) ha definito uno standard per localizzare in modo univoco le informazioni su internet: **URI**;
- un URI può essere o un **U**niform **R**esource **L**ocator o un **U**niform **R**esource **N**ame;
- per navigare in internet si usano le **URL**
<protocol>://<host>[:<port>][<path> [? <query>]
(e.g., <http://www.google.com/search?q=www>)

3. Sistema Client-Server

i documenti risiedono su dei **server** e i **client** **richiedono** e **visualizzano** questi documenti

- un Client (**web browser**) si connette ad un Server (**web server**) per recuperare le informazioni tramite il protocollo **HTTP** (**H**yper**T**ext **T**ransfer **P**rotocol)
- ogni richiesta avviene attraverso una URL
 - se un web browser richiede una pagina HTML contenente **5** immagini, verranno eseguite **6** richieste **indipendenti**
- ogni Server può servire **più** Client
- il protocollo HTTP è **stateless** (il Server non tiene traccia delle comunicazioni con i Client)

4. Linguaggi basati su marcatori

convenzione per rappresentare un documento che include informazioni e collegamenti ad altri documenti

- serve per realizzare ipertesti
- ogni elemento di controllo (che non rappresenta informazione) è rappresentato da un marcatore

Esempio

```
<html>
  <head>
    <title>Prova</title>
  </head>
  <body>
    <p>Hello world!
      <a ref="mailto:helloworld@mail.com">drop a line</a>
    </p>
  </body>
</html>
```

- Il primo linguaggio creato per il Web fu l'HyperText Markup Language ([HTML](#))
- Ispirato da [SGML](#) (**S**tandard **G**eneralized **M**arkup **L**anguage, ISO 8879:1986), usato per definire altri linguaggi basati su marcatori
- Dal 1996 l'HTML viene mantenuto dal [W3C](#)
- 1997: W3C rilascia le specifiche per la versione 4.0
- 2000: HTML diventa standard [ISO](#) (ISO/IEC 15445:2000)
- 2014: HTML5 rilasciato come W3C recommendation

- Non è un linguaggio di programmazione
 - non ha meccanismi che consentono di prendere delle decisioni
 - non è in grado di compiere delle iterazioni
- È un linguaggio di marcatura (Markup)
- Problema dell' HTML è la sua natura ibrida (lega l'**informazione** al modo di **rappresentarla**)

L'ultima versione di HTML è [HTML 5](#)

- idea: [separare](#) la struttura definita dal markup dal contenuto di una pagina Web
 - minore necessità di avere plugins (e.g., Flash)
 - elementi specifici per il controllo di contenuti multimediali
 - migliore gestione degli errori
 - indipendenza dalla piattaforma
 - supporto alla geolocalizzazione
 - sistema alternativo ai cookies ([HTML Local Storage](#))

- Dal 2000 il W3C propone **XHTML** (e**X**tensible **HTML**) come standard per Internet, in abbinamento con i **CSS** (**C**ascading **S**tyle **S**heets)
 - XHTML: usato per rappresentare il contenuto informativo
 - CSS: usati per la rappresentazione dell'informazione
- XHTML è basato su **XML** (e**X**tensible **M**arkup **L**anguage)
 - un documento XHTML è sostanzialmente una pagina HTML scritta secondo lo standard XML

- Definiscono **come** devono essere visualizzate gli elementi (tag) (X)HTML
- Sono stati aggiunti come estensione dalla versione 4.0 di HTML per **separare** il contenuto dalla forma
 - Originariamente l'HTML doveva solo rappresentare il contenuto
 - Dalla versione 3.0 di HTML sono stati **aggiunti tag per la formattazione** (e.g., ``)
 - Pensati per sostituire i tag di formattazione
 - **Nota:** XHTML in questo senso è un ritorno all'idea base dell'HTML

Problemi relativi ai CSS

- Non tutti i browser sono conformi alle specifiche
 - I browser vecchi non li supportano in modo completo
 - Ogni produttore tende ad aggiungere le proprie estensioni non compatibili con lo standard e con gli altri produttori
- Browser conformi alle specifiche possono implementare in modo leggermente diverso le specifiche stesse

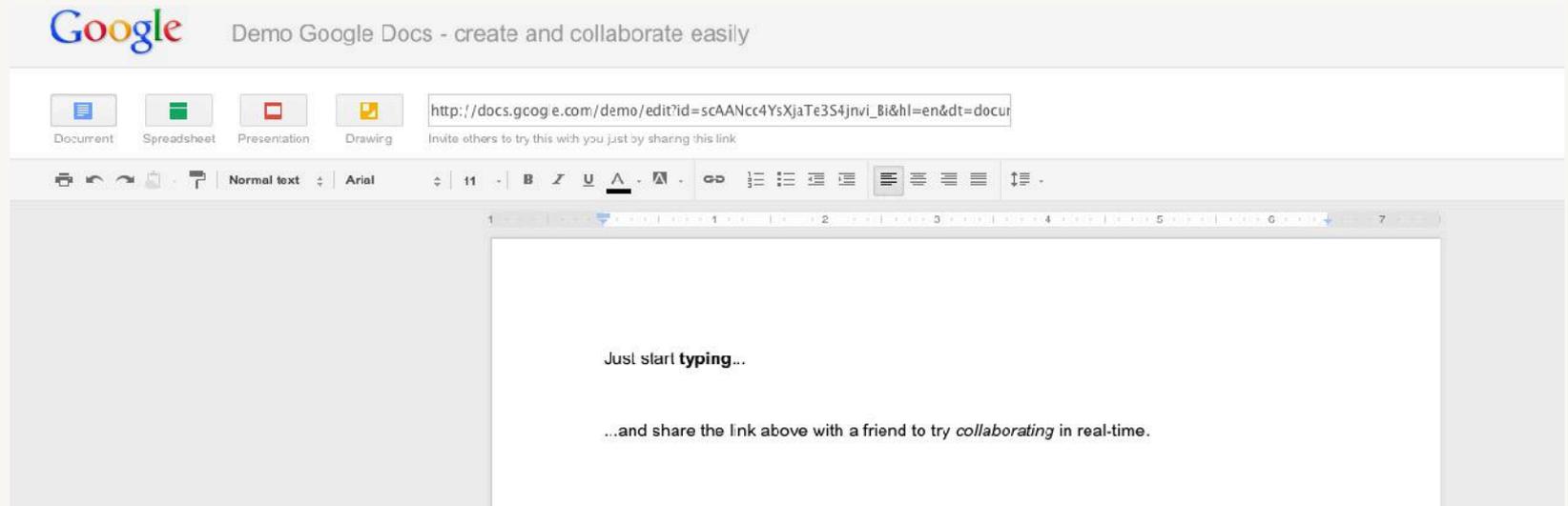
Il termine Web 2.0 è stato coniato per indicare un radicale cambiamento del modo di utilizzare lo strumento Web

- da fruitori passivi a attori attivi del Web.



- Semplifica l'iterazione dell'utente con il Web e lo rende in grado di **pubblicare** informazioni su internet senza essere un esperto
- Miglioramento dei **protocolli** di comunicazione per rendere più fruibile l'esperienza internet
- Si è passati dai **siti personali** ai **blog**...
 - Prima: necessario conoscere i linguaggi basati su tag sfruttare la propria `vena artistica'
 - Ora: possibile operare direttamente su web e sfruttare numerosi **template** per l'estetica
- ...dai **siti divulgativi** ai wiki
 - Prima: siti mantenuti da poche persone
 - Ora: tutti possono pubblicare nuovi contenuti

- ... dalle applicazioni **client side** a quelle **server side**
 - Prima: le applicazioni erano lato client ed era possibile solo salvare i dati su un server remoto
 - Ora: esistono applicazioni utilizzabili direttamente via web (senza bisogno di installarle sul client), i dati vengono salvati su un server e sono condivisi fra più utenti





Ambiente di sviluppo XAMPP

Pacchetti tutto incluso (1)

Il modo più semplice per avere un sistema completamente funzionante è usare un “pacchetto tutto incluso”

- **Easy-PHP**
<http://www.easyphp.org/>
- **XAMPP**
<http://www.apachefriends.org/>

I software che distribuiscono sono

- Open Source
- Multi-piattaforma

Pacchetti tutto incluso (2)

Concetto di piattaforma LAMP

- Sistema Operativo: **Linux**
- Web Server: **Apache**
- Database: **MySQL** (ora **MariaDB**)
- Linguaggio Scripting: **PHP** (e/o Perl e/o Python)

XAMPP: elaborazione di LAMP

- Sistema Operativo: **X** (Cross-platform)
- Web Server: **Apache**
- Database: **MySQL** (ora **MariaDB**)
- Linguaggio Scripting: **PHP, Perl**

Apache Software Foundation

<http://httpd.apache.org/>

- **Web Server** Open Source (sviluppato dal 1995)
- Ha una architettura modulare
 - *Core*: il programma principale che gestisce tutti i moduli
 - *Translation*: traduce la richiesta del client
 - *Access Control*: gestisce il controllo dell'accesso
 - *MIME Type*: gestisce i formati delle risorse
 - *Response*: gestisce la creazione della risposta per il client
 - *Logging*: tiene traccia di tutto ciò che è stato fatto
- Molto apprezzato nel mondo reale (#1 al mondo dal 1996)

The PHP Group

<http://www.php.net/>

- PHP significa: **PHP: Hypertext Preprocessor**
(è acronimo ricorsivo)
- Linguaggio di **scripting server-side** ed interpretato (sviluppato dal 1994)
- È Open Source (anche se la licenza non è compatibile con la GPL)
- Ha una grande libreria pronta all'uso
 - PEAR (**P**HP **E**xtension and **A**pplication **R**epository)
<http://pear.php.net/>

The phpMyAdmin Project

<http://www.phpmyadmin.net/>

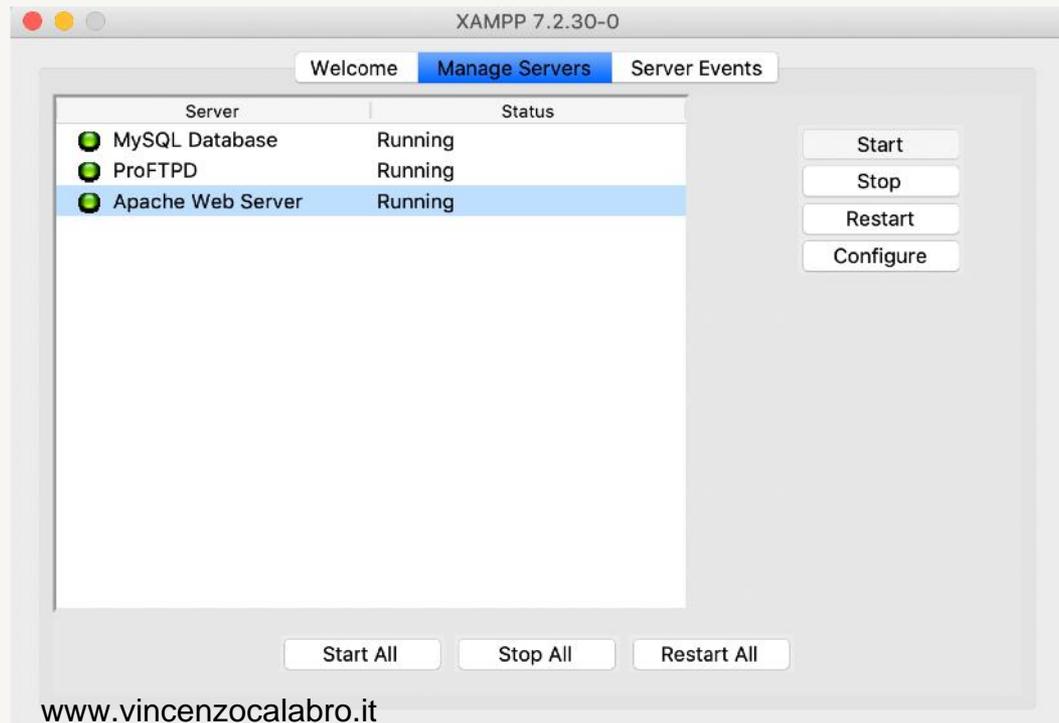
- È un'applicazione PHP Open-Source
 - Funziona attraverso un browser internet
- Serve per amministrare MySQL/MariaDB
 - Gestione Basi di Dati
 - Creazione/Distruzione/Modifica Relazioni
 - Esecuzione Query
 - Gestione/Backup dei dati
- È disponibile sotto licenza GPL

Installazione

- **Scaricare** dal sito web di XAMPP (<https://www.apachefriends.org/download.html>) la distribuzione di interesse a seconda del sistema operativo disponibile
- **Aprire** il file scaricato
- **Installare** il software nella locazione più comoda per l'utente

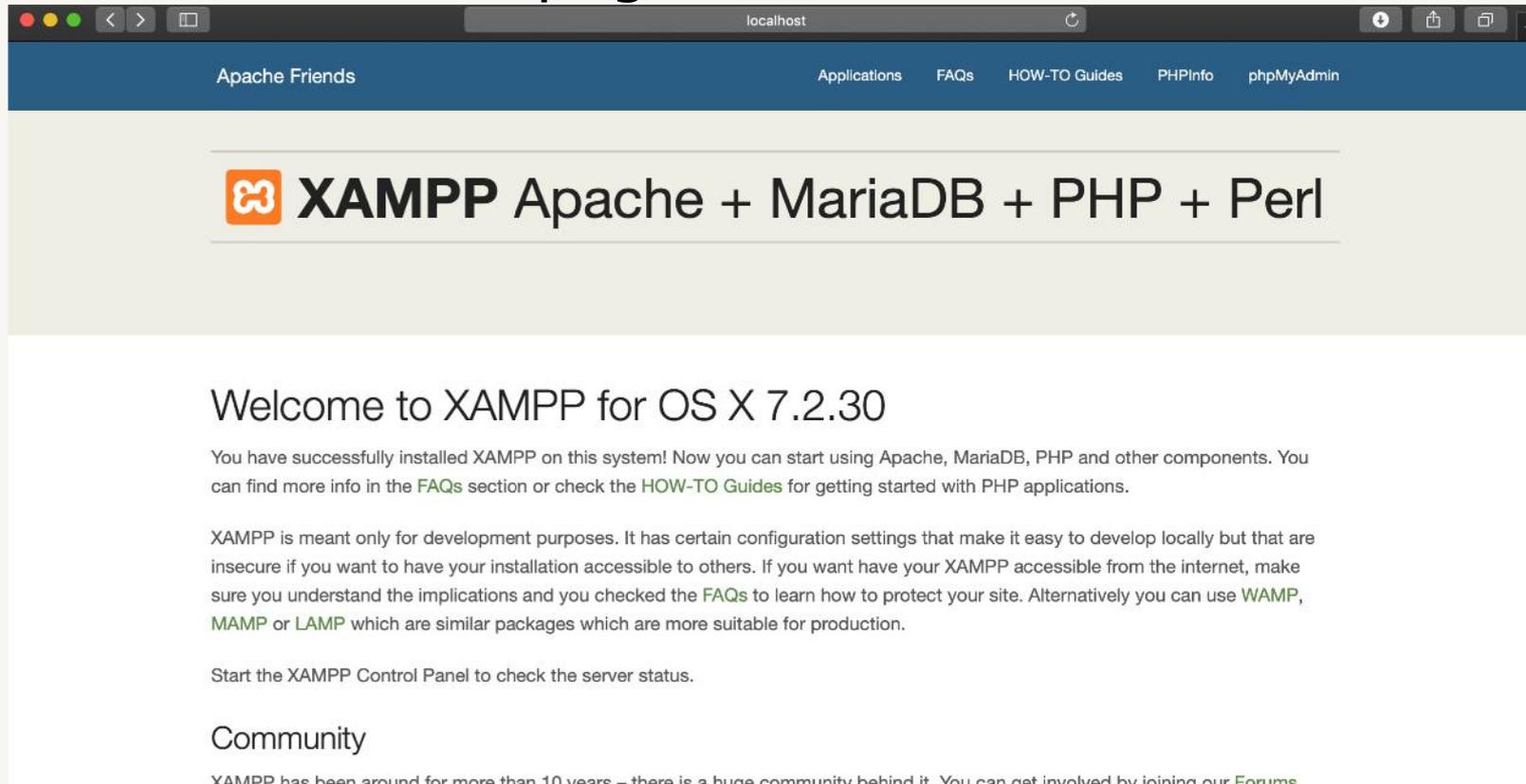
Per iniziare (1)

- Aprire il **pannello di controllo** di XAMPP
 - Windows: Start → Programs → XAMPP
 - Mac OS X: Applications → XAMPP → manager-osx
- **Avviare** i servizi di interesse (**Manage servers**)
 - Database
 - Apache



Per iniziare (2)

- All'URL <http://localhost>, da qualsiasi browser, si accede ad una pagina di welcome



Php MyAdmin (1)

- Per iniziare a lavorare su una base di dati, seguire il link a [phpMyAdmin](#)



The screenshot shows the XAMPP control panel website. The browser address bar shows 'localhost'. The navigation bar includes links for 'Apache Friends', 'Applications', 'FAQs', 'HOW-TO Guides', 'PHP info', and 'phpMyAdmin'. The 'phpMyAdmin' link is circled in red, and a red arrow points from the text in the bullet point above to this link. Below the navigation bar, the main heading reads 'XAMPP Apache + MariaDB + PHP + Perl'. The page content includes a welcome message for XAMPP for OS X 7.2.30, instructions on how to use the control panel, and a link to the community forums.

Welcome to XAMPP for OS X 7.2.30

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the [FAQs](#) to learn how to protect your site. Alternatively you can use [WAMP](#), [MAMP](#) or [LAMP](#) which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

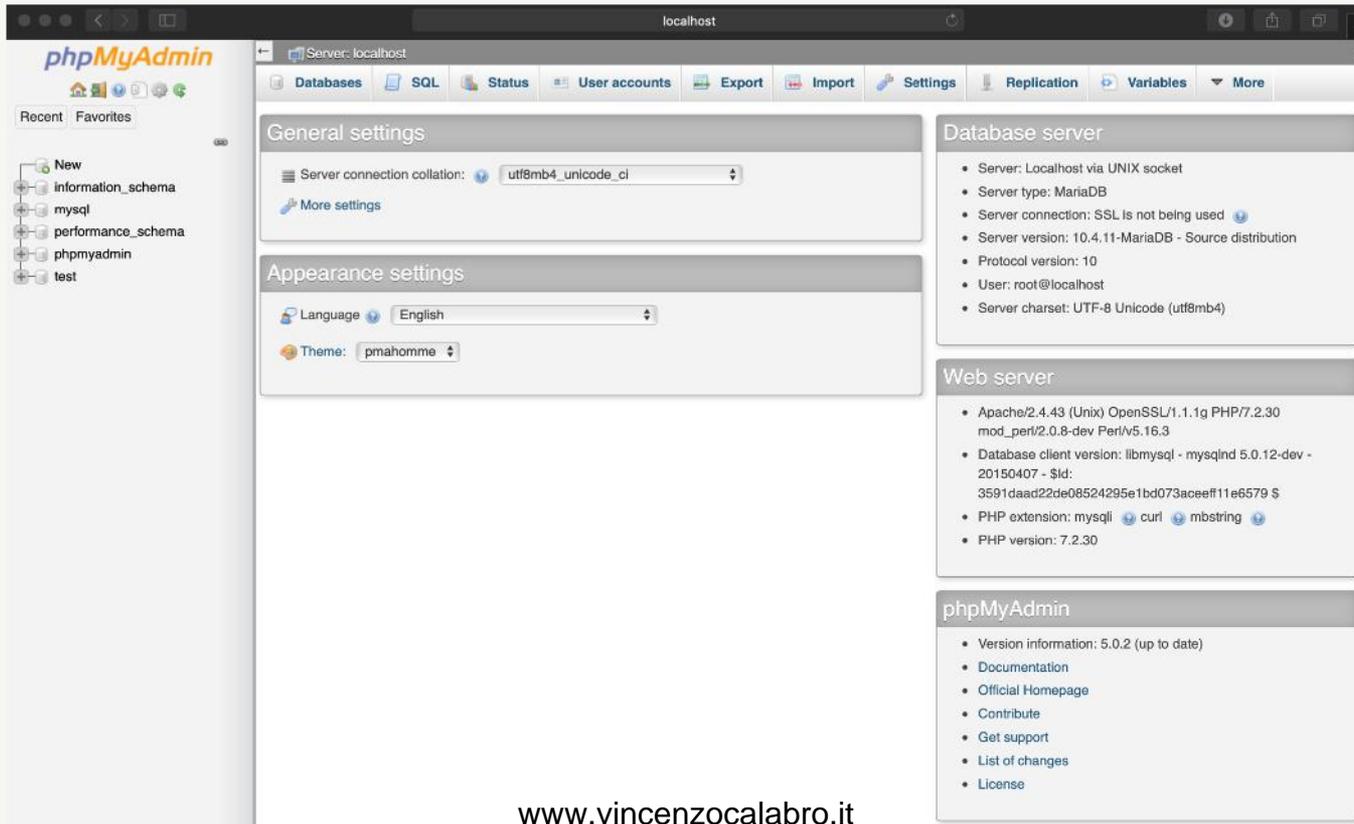
Community

www.vincenzocalabro.it

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our [Forums](#).

Php MyAdmin (1)

- Per iniziare a lavorare su una base di dati, seguire il link a [phpMyAdmin](#)
- Da qui possiamo lavorare sulla nostra base di dati



The screenshot displays the phpMyAdmin interface in a browser window. The top navigation bar includes tabs for Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables, and More. The left sidebar shows a tree view of databases: information_schema, mysql, performance_schema, phpmyadmin, and test. The main content area is divided into several panels:

- General settings:** Shows 'Server connection collation' set to 'utf8mb4_unicode_ci' with a 'More settings' link.
- Appearance settings:** Shows 'Language' set to 'English' and 'Theme' set to 'pmahomme'.
- Database server:** Lists server details: Localhost via UNIX socket, MariaDB, SSL not used, version 10.4.11-MariaDB, protocol 10, user root@localhost, and charset UTF-8 Unicode (utf8mb4).
- Web server:** Lists software versions: Apache/2.4.43, OpenSSL/1.1.1g, PHP/7.2.30, mod_perl/2.0.8-dev, Perl/v5.16.3, Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 3591daad22de08524295e1bd073aceeff11e6579 \$, PHP extension: mysqli, curl, mbstring, and PHP version: 7.2.30.
- phpMyAdmin:** Lists version information (5.0.2), documentation, official homepage, contribute, get support, list of changes, and license.

Php MyAdmin (2)

phpMyAdmin

Server: localhost

Databases SQL Status User accounts Export Import Settings Replication Variables More

Recent Favorites

New
information_schema
mysql
performance_schema
phpmyadmin
test

General settings

server connection collation: utf8mb4_unicode_ci

More settings

Appearance settings

Language: English

Theme: pmahomme

Database server

- Server: Localhost via UNIX socket
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.11-MariaDB - Source distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.43 (Unix) OpenSSL/1.1.1g PHP/7.2.30 mod_perl/2.0.8-dev Perl/v5.16.3
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 3591daad22de08524295e1bd073aceeff11e6579 \$
- PHP extension: mysqli curl mbstring
- PHP version: 7.2.30

phpMyAdmin

- Version information: 5.0.2 (up to date)
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

gestire la base di dati

scrivere interrogazioni

gestire i privilegi degli utenti

importare/esportare i dati dalla base di dati

Php MyAdmin (3)

- Fornisce **informazioni** rispetto alle versioni dei servizi e l'utente collegato

Database server

- Server: Localhost via UNIX socket
- Server type: MariaDB
- Server connection: SSL is not being used ⓘ
- Server version: 10.4.11-MariaDB - Source distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.43 (Unix) OpenSSL/1.1.1g PHP/7.2.30 mod_perl/2.0.8-dev Perl/v5.16.3
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 3591daad22de08524295e1bd073aceeff11e6579 \$
- PHP extension: mysqli ⓘ curl ⓘ mbstring ⓘ
- PHP version: 7.2.30

phpMyAdmin

- Version information: 5.0.2 (up to date)
- [Documentation](#)
- [Official Homepage](#)
- [Contribute](#)
- [Get support](#)
- [List of changes](#)
- [License](#)

Storage Engines (1)

- **DBMS Storage Engine**: componente software usato dal DBMS per creare, leggere, aggiornare, e cancellare (**CRUD - create, read, update and delete**) dati da una base di dati
- MariaDB supporta diversi **Storage Engine**
 - utilizzati per gestire diversi tipi di tabelle
- È possibile gestire **più storage engine** per una **singola base di dati**

Storage Engines (2)

- Gli Storage Engine a disposizione dipendono dalla **versione** del DBMS installato

Your SQL query has been executed successfully.

[SHOW ENGINES](#)

Pro

+ Options

Engine	Support	Comment	Transactions	XA	S
Aria	YES	Crash-safe tables with MyISAM heritage. Used for i...	NO	NO	M
MRG_MyISAM	YES	Collection of identical MyISAM tables	NO	NO	M
MEMORY	YES	Hash based, stored in memory, useful for temporary...	NO	NO	M
BLACKHOLE	YES	/dev/null storage engine (anything you write to it...	NO	NO	M
MyISAM	YES	Non-transactional engine with good performance and...	NO	NO	M
CSV	YES	Stores tables as CSV files	NO	NO	M
ARCHIVE	YES	gzip-compresses tables for a low storage footprint	NO	NO	M
InnoDB	DEFAULT	Supports transactions, row-level locking, foreign ...	YES	YES	Y

- Ricordate che MariaDB è fork di MySQL: alcuni engine sono (evoluzioni di) quelli di MySQL

Storage Engines (3)

- InnoDB

- Storage engine di **default**
- Presente anche in MySQL, ma dalla versione 10.3.7 le implementazioni divergono significativamente
- Supporta **chiavi esterne** e integrità referenziale (ma operazioni di delete/update attivate da foreign key **non** attivano trigger)
- Supporta le **transazioni**
- Garantisce il rispetto delle proprietà **ACIDe**
- Gestisce diverse modalità di **lock**

Storage Engines (3)

- MyISAM

- Leggero e performante
- Supporta indici FULLTEXT
- Non supporta transazioni, non supporta chiavi esterne
- MRG_MyISAM: collezione di tabelle MyISAM identiche

- Aria

- Evoluzione di MyISAM con migliorie di performance

- Archive

- Ottimizzato per gestire grandi archivi di dati
- Memorizza dati compressi, senza indici
- Nuove tuple sono inserite in un buffer di compressione, memorizzato sul disco quando necessario

Storage Engines (4)

- **BlackHole**

- Riceve ma **non memorizza** i dati
- Utile in ambienti distribuiti dove i dati non vengono memorizzati in locale ma replicati in automatico

- **CSV**

- Memorizza i dati come **file di testo** usando la virgola come separatore
- Utile ad esempio per lo scambio di dati fra sistemi diversi
- **Evoluzione è lo storage engine CONNECT**

- **Memory**

- Il contenuto viene mantenuto in **memoria (non su disco)**
- Utile per **cache read-only** di altre tabelle, o per **aree di lavoro temporanee**
- Non applicabile per memorizzazione permanente



Creazione di una base di dati e gestione del suo schema

Creare una base di dati

- Compilare la **form** fornita da phpMyAdmin



- Usare il **comando SQL**

```
CREATE [OR REPLACE] {DATABASE | SCHEMA} [IF NOT EXISTS]
db_name [create_specification] ...
```

create_specification:

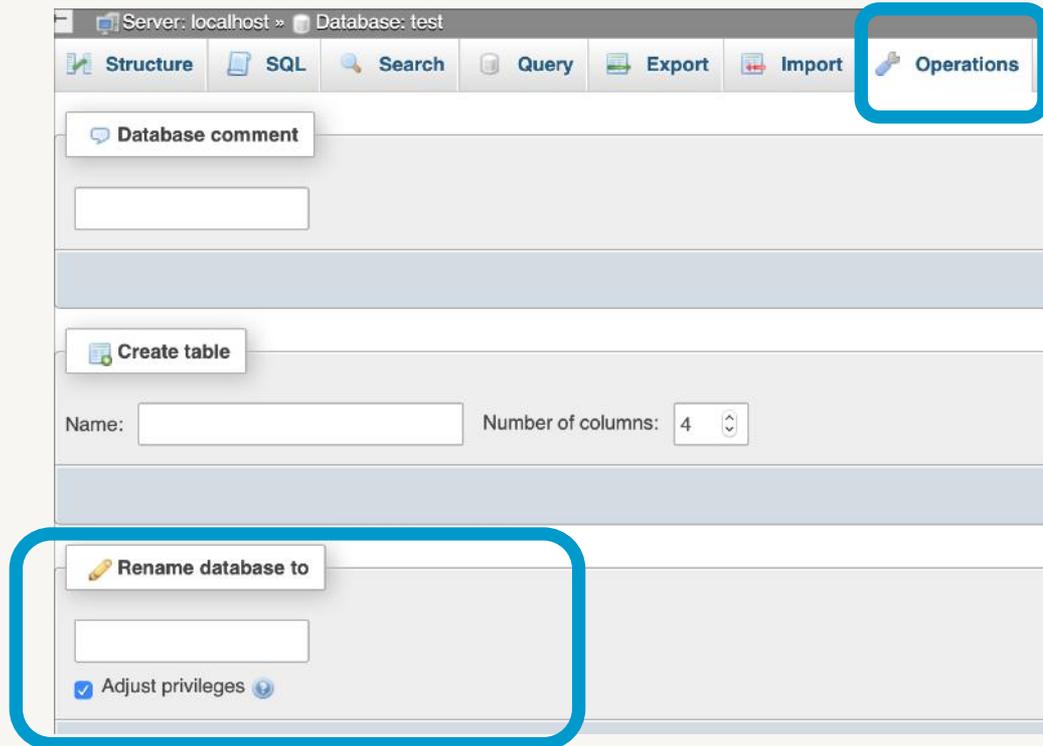
```
[DEFAULT] CHARACTER SET [=] charset_name |
```

```
[DEFAULT] COLLATE [=] collation_name |
```

```
COMMENT [=] 'comment'
```

Rinominare una base di dati

- Compilare la **form** sotto il tab Operation



The screenshot shows a database management interface with a toolbar at the top containing buttons for Structure, SQL, Search, Query, Export, Import, and Operations. The 'Operations' tab is selected and highlighted with a blue box. Below the toolbar, there are three main sections: 'Database comment', 'Create table', and 'Rename database to'. The 'Rename database to' section is highlighted with a blue box and contains a text input field, a checked checkbox labeled 'Adjust privileges', and a small help icon.

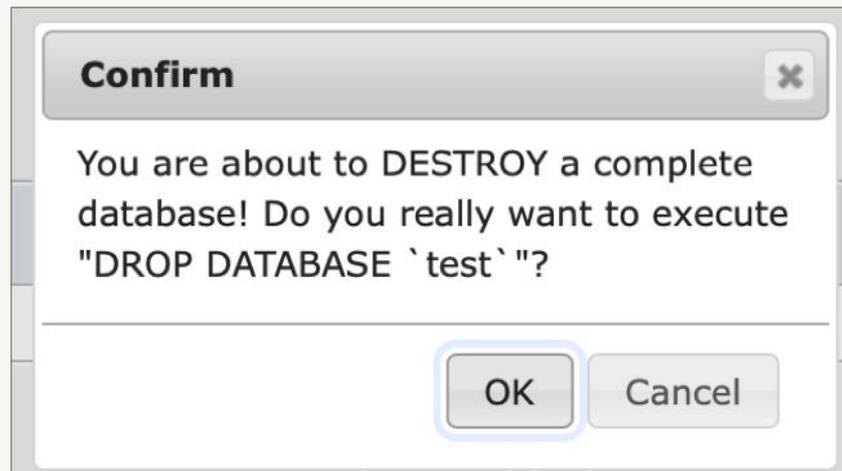
- Nota: il comando `RENAME DATABASE` non è più disponibile (rimosso per questioni di sicurezza)

Eliminare una base di dati

- Usare il tab **Drop**
- Usare il **comando SQL**

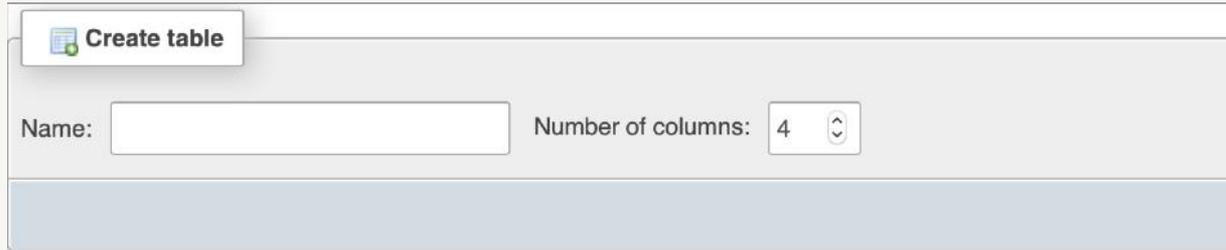
```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name
```

Attenzione: il comando DROP DATABASE **elimina** la base di dati e tutte le tue tabelle (inclusi i dati che memorizzano)!!



Creare una tabella (1)

- Compilare la **form** sotto il tab Structure



Create table

Name: Number of columns: 4

Compilare la form che compare specificando le caratteristiche di ciascun attributo

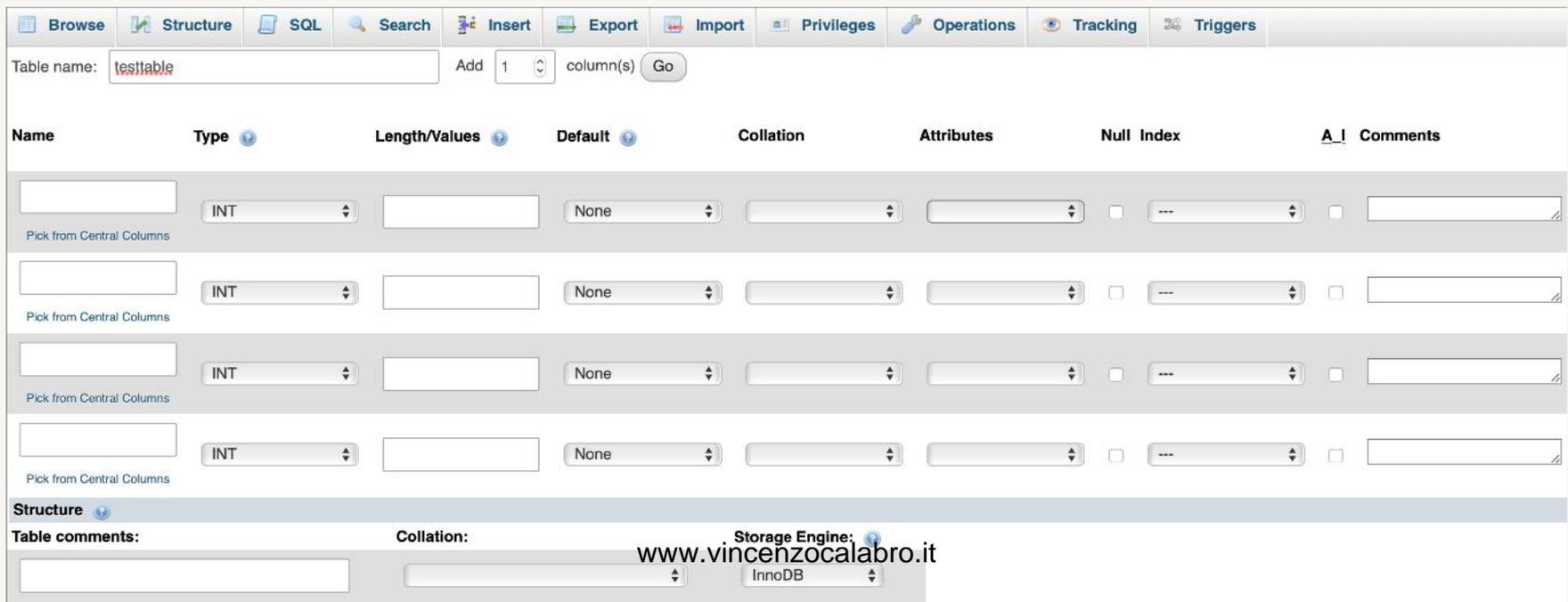


Table name: Add column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null Index	Comments
<input type="text"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
<input type="text"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
<input type="text"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
<input type="text"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>

Structure

Table comments:

Collation:

Storage Engine:

www.vincenzocalabro.it

Creare una tabella (2)

- Usare il comando SQL

```
CREATE [OR REPLACE] [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...) [table_options ]... [partition_options]
CREATE [OR REPLACE] [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)] [table_options ]... [partition_options]
    select_statement
CREATE [OR REPLACE] [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    { LIKE old_table_name | (LIKE old_table_name) }

select_statement:
    [IGNORE | REPLACE] [AS] SELECT ... (Some legal select statement)
```

- Esempio:

```
create table if not exists test (
a bigint auto_increment primary key,
name varchar(128) charset utf8,
key name (name(32))
) engine=InnoDB default_charset latin1;
```

Creare una tabella (3)

- Usare il comando SQL

```
create_definition:
  { col_name column_definition | index_definition | period_definition | CHECK (expr) }

column_definition:
  data_type
  [NOT NULL | NULL] [DEFAULT default_value | (expression)]
  [AUTO_INCREMENT] [ZEROFILL] [UNIQUE [KEY] | [PRIMARY] KEY]
  [INVISIBLE] [{WITH|WITHOUT} SYSTEM VERSIONING]
  [COMMENT 'string'] [REF_SYSTEM_ID = value]
  [COLUMN_FORMAT {FIXED|DYNAMIC|DEFAULT}]
  [reference_definition]
  | data_type [GENERATED ALWAYS]
  AS { { ROW {START|END} } | { (expression) [VIRTUAL | PERSISTENT | STORED] } }
  [UNIQUE [KEY]] [COMMENT 'string']

constraint_definition:
  CONSTRAINT [constraint_name] CHECK (expression)
```

Creare una tabella (4)

- Usare il comando SQL

```
index_definition:  
  {INDEX|KEY} [index_name] [index_type] (index_col_name,...) [index_option] ...  
  | {FULLTEXT|SPATIAL} [INDEX|KEY] [index_name] (index_col_name,...) [index_option] ...  
  | [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...) [index_option] ...  
  | [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index_name] [index_type] (index_col_name,...) [i  
  | [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (index_col_name,...) reference_definition
```

```
index_col_name:  
  col_name [(length)] [ASC | DESC]
```

```
index_type:  
  USING {BTREE | HASH | RTREE}
```

```
index_option:  
  KEY_BLOCK_SIZE [=] value  
  | index_type  
  | WITH PARSER parser_name  
  | COMMENT 'string'  
  | CLUSTERING={YES| NO}
```

```
reference_definition:  
  REFERENCES tbl_name (index_col_name,...)  
  [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]  
  [ON DELETE reference_option]  
  [ON UPDATE reference_option]
```

```
reference_option:  
  RESTRICT | CASCADE | SET NULL | NO ACTION
```

Tipi di dato (1)

- I tipi di dati (`data_type`) si distinguono in 4 categorie
 - Numerici
 - Stringhe
 - Date e ore
 - Altri tipi

- Tipi di dati numerici
 - **INT/INTEGER, TINYINT, SMALLINT, MEDIUMINT, BIGINT**: numeri interi signed/unsigned, supportano diversi range di valori (esempi:
TINYINT supporta [-128, 127] oppure [0, 255],
BIGINT supporta
[9.223.372.036.854.775.808, 9.223.372.036.854.775.807]
oppure [0, 18.446.744.073.709.551.615])
 - **BOOLEAN** (sinonimo di TINYINT(1))
0=FALSE, tutto il resto=TRUE
 - **DECIMAL[(M[,D])]** (sinonimi **DEC, NUMERIC, FIXED**): numeri a virgola fissa con M cifre prima e D cifre dopo la virgola
 - **FLOAT**: single-precision floating-point
 - **DOUBLE** (sinonimi **DOUBLE PRECISION, REAL**): double-precision floating-point
 - **BIT**

- Stringhe

- **CHAR(n)**: rappresenta una stringa di lunghezza n (anche se il valore è più corto)
- **VARCHAR(n)**: rappresenta una stringa di lunghezza al massimo n, lo spazio occupato dipende dall'istanza
- **BINARY e VARBINARY**: sono l'equivalente di CHAR e VARCHAR ma contengono stringhe binarie
- **BLOB**: permette di memorizzare grosse collezioni di dati come stringhe binarie, a seconda della dimensione, si utilizzano le varianti **TINYBLOB**, **BLOB**, **MEDIUMBLOB**, e **LOB**
- **TEXT**: è l'equivalente di BLOB, ma interpreta i dati memorizzati come stringhe di caratteri, è presente nelle varianti **TINYTEXT**, **TEXT**, **MEDIUMTEXT**, and **LONGTEXT**

- Date e ore
 - **DATE**: rappresenta una data in formato 'YYYY-MM-DD' fra '1000-01-01' e '9999-12-31'
 - **TIME**: rappresenta un orario oppure il tempo intercorso fra due eventi in formato 'HH:MM:SS.ssssss', può assumere valori fra '-838:59:59.999999' e '838:59:59.999999'
 - **DATETIME**: rappresenta data e ora in formato 'YYYY-MM-DD HH:MM:SS.ssssss' fra '1000-01-01 00:00:00.000000' e '9999-12-31 23:59:59.000000'
 - **TIMESTAMP**: rappresenta data e ora fra in formato 'YYYY-MM-DD HH:MM:SS.ffffff', fra 1970-01-01 00:00:01 e 2038-01-19 03:14:07
 - **YEAR**: rappresenta un anno in formato YYYY (da 1901 a 2155) oppure in formato YY (da 70 (=1970) a 69 (=2069))

- Altri tipi di dato

- ENUM

- permette all'utente di definire il dominio dell'attributo in modo esplicito
 - include sempre la stringa vuota ''
 - include il valore NULL se non diversamente indicato

- ESEMPIO

- ENUM('yes', 'no', 'maybe')

- Consente i valori NULL, '', 'yes', 'no', 'maybe'

- Per ciascun attributo definito è possibile esplicitare dei vincoli (**att_constraints**)
 - `NULL|NOT NULL` indica se l'attributo può o meno assumere il valore nullo
 - `DEFAULT default_value` definisce un valore di default
 - `AUTO_INCREMENT` indica che l'attributo è auto-incrementante, si applica solo a tipi numerici che possono solo assumere valori positivi, in ciascuna tabella ci può essere al più un attributo `AUTO_INCREMENT`
 - `UNIQUE` la relazione non può contenere valori duplicati
 - `PRIMARY KEY` l'attributo è chiave primaria per la relazione
 - `COMMENT 'string'` aggiunge un commento all'attributo

- I vincoli possono anche essere definiti a livello di relazione, dopo la definizione degli attributi
 - PRIMARY KEY [**idx_type**] (**idx_col_name**,...)
definisce una chiave primaria composta da uno o più attributi
 - {INDEX|KEY} [**idx_name**] [**idx_type**]
(**idx_col_name**,...)
definisce un indice su un attributo o un insieme di attributi
 - UNIQUE [INDEX|KEY] [**idx_name**] [**idx_type**]
(**idx_col_name**,...)
permette di definire l'opzione unique su uno o più attributi

- I vincoli possono anche essere definiti a livello di relazione, dopo la definizione degli attributi
 - `FULLTEXT [INDEX|KEY] [idx_name] (idx_col_name, ...)`
definisce un indice che permette la ricerca fulltext su uno o più attributi
 - `FOREIGN KEY [idx_name] (idx_col_name, ...)`
`REFERENCES tbl_name (index_col_name, ...)`
`[MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]`
`[ON DELETE RESTRICT|CASCADE|SET NULL|NO ACTION]`
`[ON UPDATE RESTRICT|CASCADE|SET NULL|NO ACTION]`
definisce un vincolo di integrità referenziale fra la tabella corrente e `tbl_name`
ATTENZIONE: funziona solo con InnoDB

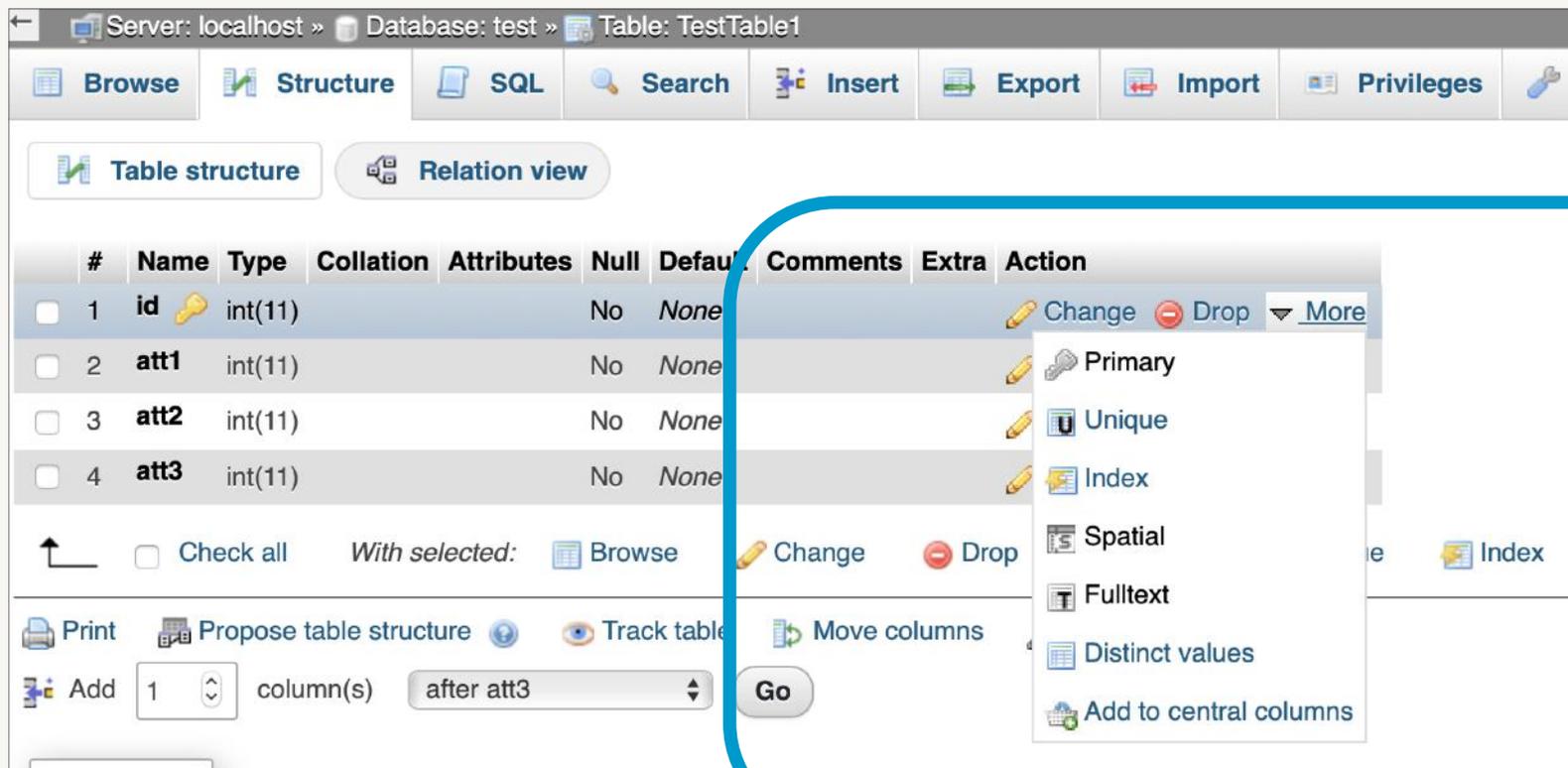
- MySQL permette anche di specificare delle opzioni specifiche per la tabella, fra queste la più rilevante è

ENGINE = **engine_name**

- permette di specificare lo Storage Engine da usare per la relazione
- va scelto con attenzione in base al supporto da offrire
- il default è **InnoDB**
- può essere **diverso** da quello di altre relazioni nella base di dati

Modificare una tabella (1)

- Usare le opzioni **Action** del tab **Structure** e modificare gli elementi di interesse



The screenshot shows the 'Structure' tab of a database management tool. The table 'TestTable1' is displayed with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None			Change Drop More
2	att1	int(11)			No	None			Change
3	att2	int(11)			No	None			Change
4	att3	int(11)			No	None			Change

The 'Action' column for the 'id' row is highlighted, and a context menu is open over it, showing the following options:

- Primary
- Unique
- Index
- Spatial
- Fulltext
- Distinct values
- Add to central columns

Modificare una tabella (2)

- Usare il comando SQL

```
ALTER TABLE tbl_name  
[alter_specification [, alter_specification] ...]
```

dove `alter_specification` permette di:

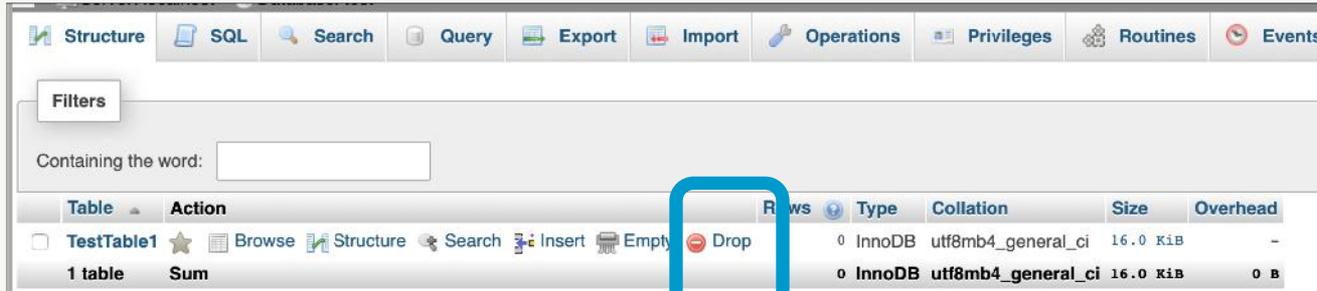
- Inserire `ADD`
- Eliminare `DROP`
- Modificare `ALTER`
- ...

diversi elementi dello schema come

- Colonne `COLUMN`
- Indici `INDEX`
- Vincoli (es. `FOREIGN KEY`)

Eliminare una tabella

- Usare l'opzione **Drop** del tab **Action**



- Usare il comando **SQL**

```
DROP [TEMPORARY] TABLE [IF EXISTS] [/*COMMENT TO SAVE*/]
tbl_name [, tbl_name] ...
[WAIT n|NOWAIT]
[RESTRICT | CASCADE]
```

Attenzione: il comando DROP TABLE **elimina** la tabella e il suo contenuto!



Inserimento e manipolazione di dati

Popolare una tabella

- Usare l'opzione **Insert** del tab **Action** e inserendo i valori degli attributi



- Usare il **comando SQL**

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] tbl_name [PARTITION (partition_list)] [(col,...)]
{VALUES | VALUE} ({expr | DEFAULT},...),(...),...
[ ON DUPLICATE KEY UPDATE
  col=expr
  [, col=expr] ... ] [RETURNING select_expr
  [, select_expr ...]]
```

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
[INTO] tbl_name [PARTITION (partition_list)] [(col,...)]
SELECT ...
[ ON DUPLICATE KEY UPDATE
  col=expr
  [, col=expr] ... ] [RETURNING select_expr
  [, select_expr ...]]
```

Modificare o cancellare una tupla (1)

- Visualizzare il contenuto della tabella ed editare o rimuovere la tupla



+ Options				
← T →				
	id	att1	att2	att3
<input type="checkbox"/>	10	20	20	20

Below the table, there are icons for **Edit** (pencil), **Copy** (plus sign), and **Delete** (minus sign).

- Per modificare, usare il **comando SQL**

```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference
  [PARTITION (partition_list)]
  SET col1={expr1|DEFAULT} [,col2={expr2|DEFAULT}] ...
  [WHERE where_condition]
  [ORDER BY ...]
  [LIMIT row_count]
```

Modificare o cancellare una tupla (1)

- Per eliminare, usare il comando SQL

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
      FROM tbl_name [PARTITION (partition_list)]
      [WHERE where_condition]
      [ORDER BY ...]
      [LIMIT row_count]
      [RETURNING select_expr
        [, select_expr ...]]
```

Esercizio 1 (1)

- Creare una base di dati **FIORISTA**

Esercizio 1 (2)

- Creare nella base di dati **FIORISTA** 3 relazioni con il seguente schema
 - **FIORE**(ID, Nome, Colore, Tipo, Prezzo)
 - FIORE.ID è autoincrementante
 - FIORE.Nome non può assumere il valore nullo
 - Se non è specificato il colore per un fiore, è perché è bianco
 - Non sono ammessi fiori con uguale nome e uguale colore
 - **CLIENTE**(CF, Nome, Cognome, NumTel)
 - CLIENTE.Cognome non può assumere il valore nullo
 - **CONSEGNA**(Numero, IDFiore, CFCliente, CittàConsegna, DataConsegna)
 - CONSEGNA.Numero è autoincrementante
 - CONSEGNA.IDFiore riferenzia FIORE.ID
 - CONSEGNA.CFCliente riferenzia CLIENTE.CF
 - update consentiti, cancellazioni no

Esercizio 1 (3)

- Popolare la base di dati inserendo le seguenti tuple

– **FIORE**(ID, Nome, Colore, Tipo, Prezzo)

- ID='1', Nome='Rosa', Colore='rosso', Tipo='Fiore reciso', Prezzo=12.50
- Nome='Giglio', Tipo='Fiore reciso', Prezzo=8.10
- Nome='Primula', Colore='giallo', Prezzo=2.80
- Nome='Rosa', Colore='rosa', Tipo='Fiore reciso', Prezzo=11.50
- Nome='Rosa', Colore='rosso', Tipo='Fiore reciso', Prezzo=15.50
- Colore='verde', Tipo='Ornamento', Prezzo=3.10

Esercizio 1 (4)

- Popolare la base di dati inserendo le seguenti tuple

– **CLIENTE**(CF, Nome, Cognome, NumTel)

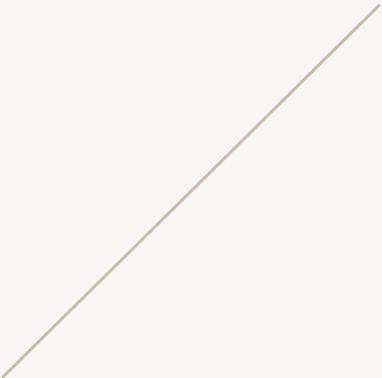
- CF='LRABNC70B52A247X', Nome='Laura', Cognome='Bianchi', NumTel='0373859562'
- CF='MRARSS65F12I935J', Nome='Mario', Cognome='Rossi'
- CF='NDRVRD80C25N864W', Nome='Andrea', Cognome='Verdi', NumTel='0363541298'

Esercizio 1 (5)

- Popolare la base di dati inserendo le seguenti tuple
 - **CONSEGNA**(Numero, IDFiore, CFCliente, CittàConsegna, DataConsegna)
 - IDFiore=1, CFCliente='LRABNC70B52A247X'
 - IDFiore=3, CFCliente='MRARSS65F12I935J', CittàConsegna='Milano', DataConsegna='2012-04-10'
 - IDFiore=3, CFCliente='MRARSS65F12I935J', CittàConsegna='Roma', DataConsegna='2012-04-10'
 - IDFiore=9, CFCliente='NDRVRD80C25N864W', CittàConsegna='Trieste', DataConsegna='2012-04-11'
- Aggiungere l'attributo sconto alla relazione CONSEGNA, il cui valore di default è 0

Esercizio 1 (6)

- Aggiornare le tabelle come segue
 - Modificare il CF del cliente 'LRABNC70B52A247X' in 'LRABNC70B52A247Y'
 - Eliminare il cliente 'MRARSS65F12I935J'
 - Cosa succede alla relazione CONSEGNA?



Gestione degli indici

Indici in MariaDB (1)

- Gli indici sono strutture dati che permettono al DBMS di ritrovare in modo **efficiente** i dati
- Essenziali quando la base di dati è **grande**
- MariaDB supporta diversi tipi di indice, principalmente:
 - indici **B-tree**
 - indici **hash** (supportati in modo esplicito solo dallo storage engine **Memory**)
 - indici **R-tree** (concetto simile a B-tree, ma usati per indicizzare spazi multi-dimensionali quali coordinate geografiche)

Indici in MariaDB (2)

- Storage engine diversi supportano indici diversi
- In MariaDB:
 - InnoDB supporta B-tree
 - Aria supporta B-tree e R-tree
 - MyISAM supporta B-tree e R-tree
 - Memory/Heap supportano B-tree e Hash
 - altri storage engine supportano altri indici meno comuni (e.g., TokuDB supporta la struttura dati **fractal trees**, ottimizzata per dati che eccedono la dimensione della memoria)

Indici in MariaDB (3)

- L'indice di **default** è quasi sempre basato su **B-trees**
- Un indice B-tree può essere usato per la valutazione di condizioni con operatori **>**, **>=**, **=**, **<=**, **<** oppure **BETWEEN**
- Può essere usato anche per la valutazione di condizioni di **LIKE** purché **inizino con un valore costante**
 - `SELECT * FROM Employees WHERE Name LIKE 'Maria%'`
può sfruttare l'indice
 - `SELECT * FROM Employees WHERE Name LIKE '%Maria'`
non può sfruttare l'indice

Definizione di indici (1)

- MariaDB genera **autonomamente** indici per gli attributi per i quali esiste un vincolo di:
 - Primary key
 - Unique
 - Foreign Key

```
PRIMARY KEY [index_type] (index_col_name,...) [index_option] ...
```

```
UNIQUE [INDEX|KEY] [index_name] [index_type](index_col_name,...)  
[index_option] ...
```

```
FOREIGN KEY [index_name] (index_col_name,...) ref_definition
```

- Fra le `index_option` è possibile specificare il tipo di indice, con **USING {BTREE | HASH | RTREE}**

Definizione di indici (2)

- È possibile dare un **nome** agli indici, **tranne che per un indice definito su primary key**
 - a seconda della versione del DBMS, dare un nome a un indice su primary key può essere **ignorato** dal DBMS senza notificare nulla, oppure può generare un **warning**
- MariaDB supporta indici full-text
- La presenza di un indice su un attributo può **rallentare** il processo di inserimento dei dati in una tabella
- È possibile **definire** indici durante la **creazione di una tabella**, oppure in un **secondo momento**

Definizione di indici (3)

- Per definire indici su una relazione **esistente**, sarebbe necessario effettuare un **alter table**
- **Ad esclusione degli indici primary key**, MariaDB mette a disposizione un comando apposito (**mappato su alter table**):

```
CREATE [OR REPLACE] [UNIQUE|FULLTEXT|SPATIAL] INDEX
[IF NOT EXISTS] index_name
  [index_type]
  ON tbl_name (index_col_name,...)
  [WAIT n | NOWAIT]
  [index_option]
  [algorithm_option | lock_option] ...

index_col_name:
  col_name [(length)] [ASC | DESC]

index_type:
  USING {BTREE | HASH | RTREE}

index_option:
  KEY_BLOCK_SIZE [=] value
  | index_type
  | WITH PARSER parser_name
  | COMMENT 'string'

algorithm_option:
  ALGORITHM [=] {DEFAULT|INPLACE|COPY|NOCOPY|INSTANT}

lock_option:
  LOCK [=] {DEFAULT|NONE|SHARED|EXCLUSIVE}
```

Rimozione di indici

- Per **rimuovere** un indice esistente, MariaDB offre un comando apposito (mappato ancora su un **alter table**)

```
DROP INDEX [IF EXISTS] index_name ON tbl_name  
[WAIT n |NOWAIT]  
[algorithm_option | lock_option] ...
```

algorithm_option:

```
ALGORITHM [=] {DEFAULT|INPLACE|COPY|NOCOPY|INSTANT}
```

lock_option:

```
LOCK [=] {DEFAULT|NONE|SHARED|EXCLUSIVE}
```

Visualizzazione di indici (2)

- Gli indici definiti su una tabella possono essere visualizzati con il comando **SHOW INDEX FROM table_name**

```
show index from provaindex
```

Profiling [\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#) [\[Refre](#)

+ Options

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
provaindex	0	PRIMARY	1	chiave	NULL	0	NULL	NULL		HASH		
provaindex	1	i_btree	1	indicebtree	A	NULL	NULL	NULL		BTREE		
provaindex	1	i_hash	1	indicehash	NULL	0	NULL	NULL		HASH		

Visualizzazione di indici (1)

```
show index from provaindex
```

Profiling [Edit inline] [Edit] [Create PHP code] [Refre

+ Options

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
provaindex	0	PRIMARY	1	chiave	NULL	0	NULL	NULL		HASH		
provaindex	1	i_btree	1	indicebtree	A	NULL	NULL	NULL		BTREE		
provaindex	1	i_hash	1	indicehash	NULL	0	NULL	NULL		HASH		

- **Table, Key_name, Column_name**: nomi della tabella, dell'indice, e dell'attributo
- **Non_Unique**: 1 se duplicati permessi
- **Seq_in_index**: posizione dell'attributo nell'indice
- **Collation**: A se ordinamento crescente, NULL altrimenti
- **Cardinality**: Stima dei valori unici nell'indice
- **Sub_part**: numero dei caratteri inclusi (NULL se tutti)
- **Packed**: metodo di compressione (NULL se nessuno)
- **Null**: vuoto se valori nulli non ammessi, NULL altrimenti
- **Index_type**: tipo dell'indice

Uso di indici (1)

- Il comando

EXPLAIN

permette di ottenere informazioni sull'uso degli indici nell'esecuzione di una query

Your SQL query has been executed successfully.

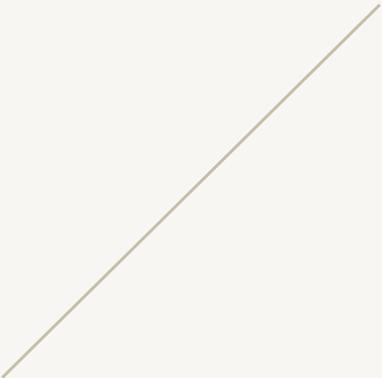
```
EXPLAIN SELECT P1.Codice, P2.Nome FROM Prodotti P1, Prodotti P2 WHERE P1.Codice > P2.Codice
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Skip Explain SQL\]](#) [\[Analyz](#)

+ Options

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	P1	index	PRIMARY	PRIMARY	64	NULL	12	Using index
1	SIMPLE	P2	ALL	PRIMARY	NULL	NULL	NULL	12	Range checked for each record (index map: 0x1)

- in questo caso, l'indice possibile per entrambe le tabelle è primary (**possible_keys = primary**)
- la tabella P1 viene scandita usando l'indice primary (**type = index**), mentre la tabella P2 viene scandita nel suo insieme (**type = ALL**)



Gestione dei trigger

- Un trigger è un **insieme di istruzioni** che vengono eseguite quando un **certo evento avviene** su una tabella
- L'evento `scatenante` può essere
 - **INSERT**
 - **UPDATE**
 - **DELETE**
- Le **istruzioni** del trigger possono essere eseguite
 - **(immediatamente) prima** dell'evento scatenante
 - **(immediatamente) dopo** l'evento scatenante
 - **al commit di una transazione**

Creazione di Trigger in MariaDB (1)

```
CREATE [OR REPLACE]
  [DEFINER = { user | CURRENT_USER | role | CURRENT_ROLE }]
  TRIGGER [IF NOT EXISTS] trigger_name trigger_time trigger_event
  ON tbl_name FOR EACH ROW
  [{ FOLLOWS | PRECEDES } other_trigger_name ]
  trigger_stmt
```

- **Trigger_time:**
 - BEFORE o AFTER
- **Trigger_event:**
 - INSERT (in seguito a INSERT, LOAD DATA, REPLACE), o
 - UPDATE, o
 - DELETE (in seguito a DELETE o REPLACE)
- **Attenzione!** DROP TABLE e TRUNCATE non causano attivazione di trigger (non usano DELETE)

Creazione di Trigger in MariaDB (2)

```
CREATE [OR REPLACE]
  [DEFINER = { user | CURRENT_USER | role | CURRENT_ROLE }]
  TRIGGER [IF NOT EXISTS] trigger_name trigger_time trigger_event
  ON tbl_name FOR EACH ROW
  [{ FOLLOWS | PRECEDES } other_trigger_name ]
  trigger_stmt
```

- **Trigger_stmt**: la lista delle azioni da intraprendere
 - qualora ci siano più istruzioni, sono racchiuse fra le parole chiave **BEGIN** e **END**
 - è possibile definire un **nuovo delimitatore temporaneo** in modo da poter utilizzare il simbolo ``;` come delimitatore per le istruzioni che compongono il trigger

Rimozione di Trigger in MariaDB

```
DROP TRIGGER [IF EXISTS] [schema_name.]trigger_name
```

- Il comando **DROP TRIGGER** elimina il trigger selezionato
- Se non viene specificato uno schema della base di dati, viene considerato quello di **default**
- **Attenzione!** La **rimozione di una tabella** (DROP) causa la rimozione di **tutti i trigger** definiti su di essa

Visualizzazione di Trigger in MariaDB (1)

- I trigger definiti su una base di dati possono essere visualizzati tramite il comando `SHOW TRIGGERS [FROM db_name]`

Your SQL query has been executed successfully.

[SHOW triggers](#)

Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Create\]](#)

+ Options

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	characte	collation	Database Collation
MaxATerraNegoziC	INSERT	RichiesteEsposizioni	BEGIN IF NEW.Tipologia = 'aTerra' AND NEW.Decisio...	BEFORE	2020-05-08 17:59:39.12	STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_...	root@localhost	utf8mb4	utf8mb4_	utf8mb4_genera

Visualizzazione di Trigger in MariaDB (2)

Your SQL query has been executed successfully.

SHOW triggers

Profiling [Edit inline] [Edit] [Cre

+ Options

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	characte	collation	Database Collation
MaxATerraNegozioC	INSERT	RichiesteEsposizioni	BEGIN IF NEW.Tipologia = 'aTerra' AND NEW.Decisio...	BEFORE	2020-05-08 17:59:39.12	STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_...	root@localho:	utf8mb4	utf8mb4_	utf8mb4_genera

- **Trigger, Event, Statement, Timing, Created**: nome, evento scatenante, statement, tempo di attivazione e data di creazione del trigger
- **Table**: tabella coinvolta dal trigger
- **sql_mode**: indicazione della modalità di utilizzo di MariaDB (usato per emulare altri DBMS server)
- **character_set_client, collation_connection, database_collation**: informazioni sui set di caratteri e collation

Si consideri la base di dati **RichiesteEsposizioni** usata nelle precedenti lezioni

- Creare un trigger che **limiti a 2 il numero massimo di richieste che hanno esito positivo nel NegozioC per tipologia 'a terra'**
 - se un inserimento viene bloccato, si generi il messaggio di errore **'NegozioC ha un massimo di 2 posti a terra!'**

Si consideri la base di dati **RichiesteEsposizioni** usata nelle precedenti lezioni

- Creare due tabelle aggiuntive
StoricoProdotti(Codice, Nome)
StoricoRichieste(Codice, NomeNeg, Tipologia, Decisione)
- Creare un trigger che, a seguito di **cancellazioni di un prodotto**:
 - **aggiunga una tupla in StoricoProdotti** con le informazioni sul prodotto
 - **sposti le relative richieste in StoricoRichieste**



Cenni sulla normalizzazione

Forme normali

- Una **forma normale** è una **proprietà** di una base di dati relazionale che ne garantisce (sotto certi aspetti) la **qualità**
- Quando una relazione non è normalizzata:
 - presenta **ridondanze**
 - si presta a **comportamenti poco desiderabili** durante le operazioni di **aggiornamento**
- Sono di solito definite sul **modello relazionale**, ma possono essere applicate in altri contesti, ad esempio il modello concettuale

Normalizzazione

- Procedura che permette di **trasformare schemi non normalizzati** in schemi che **soddisfano una forma normale**
- Forme normali e normalizzazione si utilizzano per verificare i risultati della progettazione di una base di dati (ed eventualmente migliorarli)
- La normalizzazione non è una metodologia (o una tecnica) di progetto

Relazione con anomalie

<u>Impiegato</u>	Stipendio	<u>Progetto</u>	Bilancio	Funzione
Rossi	20	Marte	2	tecnico
Verdi	35	Giove	15	progettista
Verdi	35	Venere	15	progettista
Neri	55	Venere	15	direttore
Neri	55	Giove	15	consulente
Neri	55	Marte	2	consulente
Mori	48	Marte	2	direttore
Mori	48	Venere	15	progettista
Bianchi	48	Venere	15	progettista
Bianchi	48	Giove	15	direttore

- Lo stipendio di ciascun impiegato è ripetuto in più tuple
 - ridondanza
- Se lo stipendio di un impiegato varia, è necessario modificarne il valore in tutte le tuple corrispondenti
 - anomalia di aggiornamento
- Se un impiegato esce da tutti tutti i progetti, non è possibile conservarne nome e stipendio
 - anomalia di cancellazione
- Non è possibile inserire informazioni su un impiegato che non partecipa a un progetto
 - anomalia di inserimento

- Questi fenomeni indesiderati sono causati dal fatto di aver **usato un'unica relazione per rappresentare informazioni eterogenee**:
 - gli **impiegati** con i relativi **stipendi**
 - i **progetti** con i relativi **bilanci**
 - le partecipazioni degli **impiegati** ai **progetti** con le relative **funzioni**

Dipendenza funzionale

- È un vincolo di integrità utile
- Date una relazione $r(\mathbf{U})$ e due sottoinsiemi \mathbf{X}, \mathbf{Y} di \mathbf{U} , si dice che r soddisfa una dipendenza funzionale $\mathbf{X} \rightarrow \mathbf{Y}$ da \mathbf{X} a \mathbf{Y} se, per ogni coppia di ennuple \mathbf{t}_1 e \mathbf{t}_2 di r con $\mathbf{t}_1[\mathbf{X}] = \mathbf{t}_2[\mathbf{X}]$, risulta anche $\mathbf{t}_1[\mathbf{Y}] = \mathbf{t}_2[\mathbf{Y}]$
- Nota: ad ogni chiave \mathbf{K} corrisponde una dipendenza funzionale da \mathbf{K} verso tutti gli attributi (e insiemi di attributi) della relazione
- Esempi:
 - Impiegato \rightarrow Stipendio
 - Progetto \rightarrow Bilancio
 - Impiegato, Progetto \rightarrow Funzione

Dipendenza funzionale non banale

- Sulla base della definizione sarebbe valida anche
 - Impiegato Progetto \rightarrow Progetto
- Si tratta però di una dipendenza “banale” (sempre soddisfatta)
- Dati due insiemi **X** e **Y** di attributi, e un attributo **A**:
 - **X** \rightarrow **A** è non banale se **A** non appartiene a **X**
 - **X** \rightarrow **Y** è non banale se nessuno degli attributi in **Y** appartiene a **X**

Dipendenze funzionale e anomalie (1)

- Non sempre le dipendenze funzionali causano anomalie

<u>Impiegato</u>	<u>Stipendio</u>	<u>Progetto</u>	<u>Bilancio</u>	<u>Funzione</u>
Rossi	20	Marte	2	tecnico
Verdi	35	Giove	15	progettista
Verdi	35	Venere	15	progettista
Neri	55	Venere	15	direttore
Neri	55	Giove	15	consulente
Neri	55	Marte	2	consulente
Mori	48	Marte	2	direttore
Mori	48	Venere	15	progettista
Bianchi	48	Venere	15	progettista
Bianchi	48	Giove	15	direttore

Impiegato → Stipendio causa anomalie

Progetto → Bilancio causa anomalie

Impiegato Progetto → Funzione non causa anomalie

Dipendenze funzionale e anomalie (2)

- **Impiegato Progetto** → **Funzione** non causa anomalie perché $\langle \text{Impiegato}, \text{Progetto} \rangle$ è **chiave** della relazione
 - impiegato da solo no
 - progetto da solo no
- Le anomalie sono causate dalla **presenza di concetti eterogenei**:
 - proprietà degli **impiegati** (lo stipendio)
 - proprietà di **progetti** (il bilancio)
 - proprietà **associate alla chiave**

Forma normale di Boyce e Codd (BCNF)

- Una relazione r è in forma normale di Boyce e Codd se per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di essa, X contiene una chiave K di r
- La forma normale richiede che i concetti siano omogenei (solo proprietà direttamente associate alla chiave)

Decomposizione in BCNF

- Quando una relazione non è in BCNF, può (in certi casi) essere **sostituita da altre relazioni in BCNF**
- Questo può essere fatto **decomponendo sulla base delle dipendenze funzionali**
 - con lo scopo di al fine di separare i concetti

Decomposizione in BCNF: esempio

- Prima della decomposizione:

<u>Impiegato</u>	<u>Stipendio</u>	<u>Progetto</u>	<u>Bilancio</u>	<u>Funzione</u>
Rossi	20	Marte	2	tecnico
Verdi	35	Giove	15	progettista
Verdi	35	Venere	15	progettista
Neri	55	Venere	15	direttore
Neri	55	Giove	15	consulente
Neri	55	Marte	2	consulente
Mori	48	Marte	2	direttore
Mori	48	Venere	15	progettista
Bianchi	48	Venere	15	progettista
Bianchi	48	Giove	15	direttore

Decomposizione in BCNF: esempio

- Dopo la decomposizione:

Impiegato	Progetto	Funzione
Rossi	Marte	tecnico
Verdi	Giove	progettista
Verdi	Venere	progettista
Neri	Venere	direttore
Neri	Giove	consulente
Neri	Marte	consulente
Mori	Marte	direttore
Mori	Venere	progettista
Bianchi	Venere	progettista
Bianchi	Giove	direttore

Impiegato	Stipendio
Rossi	20
Verdi	35
Neri	55
Mori	48
Bianchi	48

Progetto	Bilancio
Marte	2
Giove	15
Venere	15

Decomposizione in BCNF: problemi (1)

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

Impiegato → Sede
Progetto → Sede

Decomposizione in BCNF: problemi (2)

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Progetto	Sede
Marte	Roma
Giove	Milano
Venere	Milano
Saturno	Milano

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano
<i>Verdi</i>	<i>Saturno</i>	<i>Milano</i>
<i>Neri</i>	<i>Giove</i>	<i>Milano</i>

Decomposizione in BCNF: problemi (3)

Impiegato	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Venere	Milano
Neri	Saturno	Milano
Neri	Venere	Milano
<i>Verdi</i>	<i>Saturno</i>	<i>Milano</i>
<i>Neri</i>	<i>Giove</i>	<i>Milano</i>

- La relazione ricostruita è **diversa** dalla relazione originaria: la decomposizione **non** è **senza perdita**
- Le decomposizioni **devono essere senza perdita**

Decomposizione in BCNF: problemi (4)

- Una decomposizione differente, date le stesse FD

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Impiegato	Progetto
Rossi	Marte
Verdi	Giove
Verdi	Venere
Neri	Saturno
Neri	Venere

- Questa decomposizione è senza perdita, ma supponiamo di inserire una tupla che specifica che Neri partecipa a Marte

Decomposizione in BCNF: problemi (5)

Impiegato	Sede
Rossi	Roma
Verdi	Milano
Neri	Milano

Impiegato	Progetto
Rossi	Marte
Verdi	Giove
Verdi	Venere
Neri	Saturno
Neri	Venere
<i>Neri</i>	<i>Marte</i>

- Sembra ok sulle relazioni decomposte, ma **non conserva Progetto → Sede nella relazione originaria**

Impiegato	Progetto	Sede
Rossi	Marte	Roma
...
<i>Neri</i>	<i>Marte</i>	<i>Milano</i>

- Le decomposizioni **devono conservare le dipendenze**

Decomposizione in BCNF: problemi (6)

Dirigente	Progetto	Sede
Rossi	Marte	Roma
Verdi	Giove	Milano
Verdi	Marte	Milano
Neri	Saturno	Milano
Neri	Venere	Milano

Progetto Sede → Dirigente
Dirigente → Sede

Progetto Sede → Dirigente coinvolge tutti gli attributi,
non è possibile una decomposizione in BCNF (che
conservi questa dipendenza)

Terza Forma Normale

- Una relazione r è in **terza forma normale (3NF)** se, per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di essa, è verificata **almeno una delle seguenti condizioni**:
 - X contiene una chiave K di r
 - ogni attributo in Y è contenuto in almeno una chiave di r
- La terza forma normale è meno restrittiva della forma normale di Boyce e Codd, ma è **sempre** raggiungibile



VINCENZO CALABRÒ

LinkedIn vincenzocalbro

www.vincenzocalbro.it