

## INTRODUZIONE AI DATABASE

Un database è un sw che consente di gestire salvataggio, recupero, analisi, aggiornamento dei dati.

Tipi di DBMS (Database ManagementSystem):

- RDBMS: Relational DBMS
- NoSQL database

### RDBMS

I dati sono organizzati in tabelle.

Tipi di RDBMS:

- A pagamento: Oracle, SQLServer
- Free: MySQL, Postgresql

Operazioni **CRUD**: Create, Read, Update, Delete

## MYSQL

Database open-source più diffuso al mondo.

- MySQL CommunityEdition:
  - InnoDB: motore più performante

Dal sito:

The MySQL Community Edition includes:

- **SQL and NoSQL** for developing both relational and NoSQL applications
- **MySQL Document Store** including X Protocol, XDev API and MySQL Shell
- **Transactional Data Dictionary** with Atomic DDL statements for improved reliability
- **Pluggable Storage Engine Architecture** (InnoDB, NDB, MyISAM, etc)
- **MySQL Replication** to improve application performance and scalability **[Utile quando si hanno molti dati e molti accessi al DB]**
- **MySQL Group Replication** for replicating data while providing fault tolerance, automated failover, and elasticity
- **MySQL InnoDB Cluster** to deliver an integrated, native, high availability solution for MySQL
- **MySQL Router** for transparent routing between your application and any backend MySQL Servers
- **MySQL Partitioning** to improve performance and management of large database applications **[Migliora l'accesso ai file del file system]**
- **Stored Procedures** to improve developer productivity
- **Triggers** to enforce complex business rules at the database level
- **Views** to ensure sensitive information is not compromised **[Query che fanno vedere all'utente dei dati, ma che non gli permettono di modificarli]**
- **Performance Schema** for user/application level monitoring of resource consumption
- **Information Schema** to provide easy access to metadata
- **MySQL Connectors** (ODBC, JDBC, .NET, etc) for building applications in multiple languages **[Fa interagire il DB con altri linguaggi di programmazione. In Java JDBC Connector]**
- **MySQL Workbench** for visual modeling, SQL development and administration **[Client che consente di accedere al DB]**

## NOTA:

Quando si lavora con un DB è conveniente non lavorare mai direttamente come utente root, ma creare un utente che abbia gli stessi privilegi di un utente root e fare interfacciare lui con l'applicazione

- Si evita di passare da root, perché se qualcuno buca il DB, ha accesso a tutti i DB presenti su mysql
- Creando un utente apposito, eventuali attacchi saranno limitati soltanto al DB specifico cui quell'utente ha accesso.

## DOWNLOAD MYSQL:

Due possibilità:

1. MySQL: <https://www.mysql.com/it/products/community/>
  - Download MySQLCommunityServer
  - MySQLCommunityServer
2. XAMPP: <https://www.apachefriends.org/it/index.html>
  - Contiene un set di applicazioni, tra cui MySQL


### MYSQL

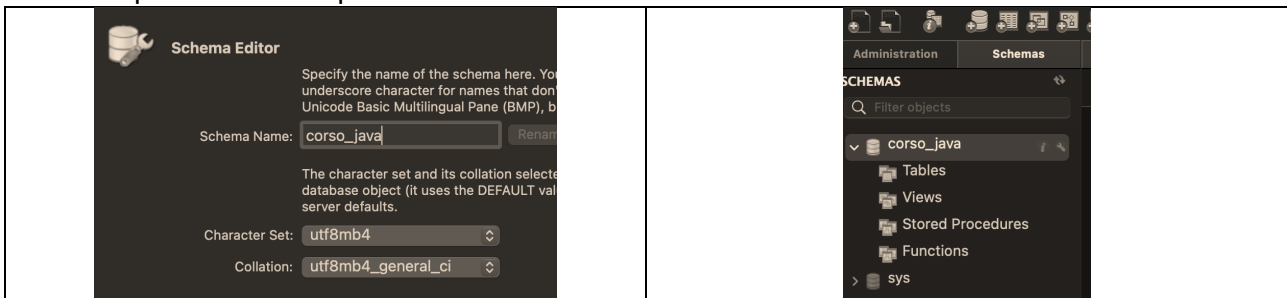
Preferenze di sistema – MYSQL – Start MySQL Server

### MARIADB

mysql -u root --password=pao1oMYSQL1990

## CREARE DB CON MYSQL

- Assicurarsi che MySQL sia connesso
- Tasto “create new schema” 
- Impostare nomi e parametri:



- Cliccando su apply compare la query standard per la creazione dello schema:

```
CREATE SCHEMA `corso_java` DEFAULT CHARACTER SET utf8mb4 ;
```

- Si può fare la stessa cosa anche da riga di comando [terminale]:

```
mysql -u root -p [oppure: mysql -u root --password=pao1oMYSQL1990]
```

→ posso scrivere le istruzioni mysql

```
CREATE SCHEMA `corso_java_2` DEFAULT CHARACTER SET utf8mb4 ;
```

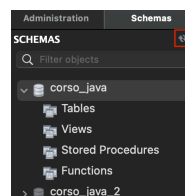
```
paolo francescosciammarella ~ mysql -u root -p -- 80x24
Last login: Thu Feb 17 11:51:22 on ttys000
paolo francescosciammarella@MacBook-Pro-di-Paolo ~ % mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> CREATE SCHEMA `corso_java_2` DEFAULT CHARACTER SET utf8mb4 ;
Query OK, 1 row affected (0.002 sec)

MySQL [(none)]>
```



## CREARE TABELLA CON MYSQL

- Esplodo lo schema in cui inserire la tabella
- Tables – tasto dx – Create table
  - PK = PrimaryKey
  - NN = Non null
  - UQ = Unique (index for the column)
  - B = Binary
  - UN = Unsigned
  - ZF = Zero fill
  - AI = Auto-increment (genera da solo ed incrementa)
  - G = Generated column
- Inserisco le colonne e clicco su apply: verrà generato lo script per la creazione della tabella:

The screenshot shows the 'Table' editor for a table named 'clienti'. The columns are: idclienti (INT, PK, NN, AI), nome (VARCHAR(255), NN), cognome (VARCHAR(255), NN), email (VARCHAR(100), NN), and telefono (VARCHAR(20), NN). The 'Apply' button is highlighted. To the right, the generated SQL script is shown:

```

1 CREATE TABLE `corso_java`.`clienti` (
2   `idclienti` INT NOT NULL AUTO_INCREMENT,
3   `nome` VARCHAR(255) NOT NULL,
4   `cognome` VARCHAR(255) NOT NULL,
5   `email` VARCHAR(100) NOT NULL,
6   `telefono` VARCHAR(20) NOT NULL,
7   PRIMARY KEY (`idclienti`));
8

```

- Creazione chiave esterna (il dominio dei di una colonna è preso da un'altra colonna: posso inserire solo i dati presenti in colonna di altra tabella):

Esempio: creazione tabella ordini

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Defa
idordini	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
id_cliente	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
data_consegna	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
importo	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

- Nome della ForeignKey
- Tabella da cui prendere i valori
- Settaggio vincolo:
  - Column: colonna tabella attuale da vincolare
  - Ref column: colonna da cui prendere i dati da usare come vincolo

Foreign Key	Referenced Table	Column	Referenced Column	On Update:	On Delete:	Comment:
id_cliente	corso_java`.`clienti`	id_cliente	idclienti	NO ACTION	NO ACTION	

SQL Generato:

```

CREATE TABLE `corso_java`.`ordini` (
  `idordini` INT NOT NULL AUTO_INCREMENT,
  `id_cliente` INT NULL,
  `data_consegna` DATETIME NULL,
  `importo` DOUBLE NULL,
  PRIMARY KEY (`idordini`),
  INDEX `id_cliente_idx` (`id_cliente` ASC) VISIBLE,
  CONSTRAINT `id_cliente`
    FOREIGN KEY (`id_cliente`)
    REFERENCES `corso_java`.`clienti` (`idclienti`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

```

## INSERIRE RIGHE IN TABELLA:

- Tasto dx: insert new row
- Oppure: dal lato cliccando sul bottone +
- Inserendo i dati a mano e cliccando poi su Apply, verranno effettuate le insert

```

INSERT INTO `corso_java`.`clienti` (`nome`,
`cognome`, `email`, `telefono`) VALUES
('Avril', 'Lavigne', 'test@test.com', '987');

```

The screenshot shows the 'clienti' table in the 'corso\_java' database. The columns are: idclienti, nome, cognome, email, and telefono. The data inserted is:

idclienti	nome	cognome	email	telefono
1	Avril	Lavigne	test@test.com	987

•