



ConcertEdia

Progetto finale del corso Basi di Dati & Web.

github.com/tiaringhio/ConcertEdia

Realizzato da:

[Filippo Botti](#) - 287065

[Mattia Ricci](#) - 285237



Table of contents

[Table of contents](#)

[Descrizione](#)

[Tecnologie usate](#)

[Pagine](#)

[Home](#)

[Concert](#)

[Band](#)

[Curiosity](#)

[SingolaBand](#)

[Concerto](#)

[Header & Footer](#)

[TOP Header](#)

[Footer](#)

[Diagramma ER](#)

[Schema Logico](#)



Query

[Band.php](#)
[Concerti.php](#)
[Concerto.php](#)
[Curiosity.php](#)
[Search.php](#)
[SingolaBand.php](#)



Descrizione

Il sito web ha lo scopo di mostrare i dati di un Database relativo ai concerti che si sono svolti nel passato. La progettazione e il versioning sono stati gestiti attraverso la piattaforma GitHub.

!! Una descrizione più completa del progetto è presente nel README trovabile nella repository GitHub associata al progetto.



Tecnologie usate

Il sito web è stato sviluppato utilizzando **php**, **HTML**, **CSS** e una minima parte di **JavaScript**.

- Per tutto ciò che riguarda la grafica è stato utilizzato il framework **Bootstrap**, dal quale sono state prese le basi grafiche, successivamente modificate tramite **CSS**.
- Per quanto riguarda la connessione tra sito e Database è stato utilizzato **php**.
- Per implementare il Database è stato utilizzato **PhpMyAdmin**, fornito da **xampp**, così come l'hosting del sito in locale.



Pagine

Seguono delle brevi descrizioni delle pagine presenti nel sito.

È composto da quattro pagine principali:



Home

Una prima pagina di presentazione del sito, che non mostra alcun dato presente nel Database, ma spiega al visitatore cos'è il sito e chi siamo.

Concert

Una pagina che, comunicando con il Database, mostra a scorrimento tutti i concerti presenti, con foto, titolo e data. Lo scorrimento dei concerti può essere effettuato dal visitatore, attraverso gli appositi comandi o con le frecce tramite tastiera, altrimenti è automatico ogni sette secondi.
Infine, cliccando su ogni concerto si passerà alla pagina specifica del concerto in questione.

Band

Una pagina che, comunicando con il Database, mostra tutte le band presenti, con foto e nome della band mostrato solo al passaggio del cursore sopra la foto, con un piacevole effetto visivo.

Anche in questo caso, le band sono cliccabili e riportano alla rispettiva pagina specifica.

? Curiosity

Una pagina che, comunicando con il Database, mostra alcune curiosità riguardo i concerti e le band, come il concerto più visto o il musicista più anziano. Ogni curiosità è ricavata grazie ad un'interrogazione del Database, di complessità varia a seconda di ciò che si vuole mostrare.

Entrando nello specifico sono inoltre presenti due successive pagine:

SingolaBand

Una pagina che, comunicando con il Database, mostra tutte le informazioni riguardo una band: nome, genere, foto, breve descrizione e la formazione.

Concerto

Una pagina che, comunicando con il Database, mostra tutte le informazioni riguardo un concerto: nome, genere, un player di YouTube collegato al link presente nel Database, descrizione, scaletta e band che hanno partecipato al concerto, con i loro componenti e gli strumenti da loro suonati.

Header & Footer

Ogni pagina sopra elencata comprende un "Header" e un "Footer".

Header

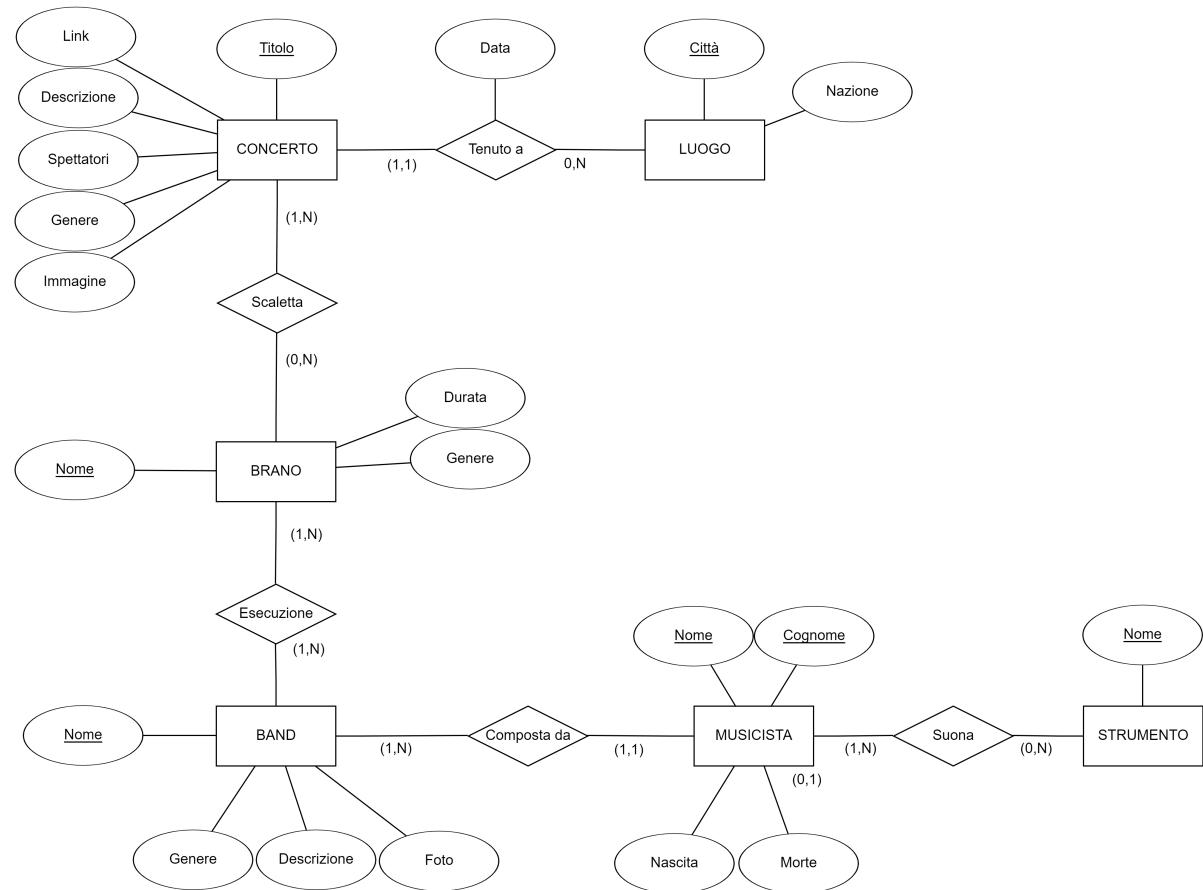
In questo elemento è mostrata una barra di navigazione per muoversi tra le varie pagine del sito e una barra di ricerca attraverso la quale è possibile cercare una band o un concerto specifico. Nel caso in cui si cercasse un elemento non presente nel Database si arriverà ad una pagina di errore che comunica la mancanza di quel dato e permette di tornare alla home o di effettuare una nuova ricerca.

⇓ Footer

Questo elemento è una breve presentazione del sito, con i rispettivi collegamenti alle pagine GitHub degli studenti responsabili del progetto e i loro indirizzi email. È inoltre presente un collegamento alla repository GitHub del progetto.

Diagramma ER

Il diagramma ER è stato studiato e progettato a mano e successivamente implementato tramite il tool online [ERDPlus](#).



Schema Logico

LUOGHI(Città, Nazione)

CONCERTI(Titolo, Descrizione, Genere, Spettatori, Link, Città, Data, Immagine)

!! Con vincolo di integrità referenziale tra Concerti.Città e Luoghi.Città.

SCALETTE(Concerto, Brano)

**!! Con vincolo di integrità referenziale tra Scalette.Concerto e
Concerti.Titolo & Scalette.Brano e Brani.Nome.**

BRANI(Nome, Genere, Durata)

ESECUZIONI(Brano, Band)

**!! Con vincolo di integrità referenziale tra Esecuzioni.Brano e Brano.nome &
Esecuzioni.Band e Band.Nome.**

BAND(Nome, Genere, Foto, Descrizione)

MUSICISTI(Nome, Cognome, Nascita, Morte, Band)

!! Con vincolo di integrità referenziale tra Musicisti.Band e Band.Nome.

SUONA(Nome_Musicista, Cognome_Musicista, Strumento)

**!! Con vincolo di integrità referenziale tra Suona.Nome_Musicista e
Musicisti.Nome & Suona.Cognome_Musicista e Musicisti.Cognome &
Suona.Strumento e Strumenti.Nome.**

STRUMENTI(Nome)



Query

Elenco delle query utilizzate con rispettiva descrizione, separate per pagina in cui sono presenti.

```
$conn = mysqli_connect($servername, $username, $password, $dbname)
or die("Connection failed: " . mysqli_connect_error());
```

Permette la connessione ad un server MySQL, le cui specifiche sono inserite nelle variabili \$servername, \$username, \$password, \$dbname.

```
$result = mysqli_query($conn, $query);
```

Permette di interrogare il Database \$conn con la query \$query scritta precedentemente.

Restituisce una struttura dati di tipo mysqli_result contenente il risultato della query.

```
$record = mysqli_fetch_assoc($resultset)
```

Recupera una riga dal risultato come array associativo.

Posta all'interno di un ciclo while recupera tutte le righe del risultato

Band.php

```
$sql = "SELECT * FROM band";
```

Questa interrogazione permette di selezionare ogni valore presente nella tabella band.

Successivamente, attraverso opportune operazioni per accedere al Database, verranno mostrati solo i campi 'Nome' e 'Foto'.

Concerti.php

```
$sql = "SELECT * FROM concerti";
```

Questa interrogazione permette di selezionare ogni valore presente nella tabella concerti.

Successivamente, attraverso opportune operazioni per accedere al Database, verranno mostrati solo i campi 'Titolo', 'Data' e 'Immagine'.

Concerto.php

```
$sql = "SELECT * FROM concerti  
WHERE Titolo = '" . $_GET['varname'] . "'";
```

Questa interrogazione permette di selezionare tra i valori della tabella concerti, il valore avente il titolo uguale alla stringa salvata in `$_GET['varname']`, determinato dal click su un concerto nella pagina precedente (Concerti.php).

Successivamente, attraverso opportune operazioni per accedere al Database, verranno mostrati solo i campi 'Titolo', 'Città', 'Link' e 'Descrizione'.

```
$scaletta
```

```
SELECT Brano FROM scalette  
WHERE scalette.Concerto = '" . $concerto . "'
```

Questa query permette di selezionare dalla tabella Scalette tutti i Brani che hanno come valore del campo concerto il titolo del concerto in questione, salvato nella variabile `$concerto`.

Successivamente, attraverso opportune operazioni per accedere al Database, verranno mostrati il nome di ogni brano.

```
$queryBand
```

```
SELECT DISTINCT band.nome  
FROM brani, concerti, band, esecuzioni, scalette  
WHERE concerti.Titolo = scalette.Concerto AND  
scalette.Brano = brani.Nome AND  
brani.Nome = esecuzioni.Brano AND  
esecuzioni.Band = band.Nome AND  
concerti.titolo = '" . $concerto . "'
```

Questa query permette di selezionare tutte le band che hanno partecipato al concerto in questione, salvato nella variabile `concerto`.

Per ottenerle occorre fare un JOIN tra le tabelle concerti e scalette, scalette e brani, brani ed esecuzioni, esecuzioni e concerti.

Successivamente, attraverso opportune operazioni per accedere al Database, verranno mostrati il nome di ogni band.

```
$query_musicista
```

```
SELECT DISTINCT musicisti.nome, musicisti.cognome
FROM musicisti, brani, concerti, band, esecuzioni, scalette
WHERE concerti.Titolo = scalette.Concerto
AND scalette.Brano = brani.Nome
AND brani.Nome = esecuzioni.Brano
AND esecuzioni.Band = band.Nome
AND musicisti.Band = band.Nome
AND band.nome = '" . $rowBand['nome'] . "'
```

Questa query permette di selezionare tutti i musicisti che fanno parte della band in questione,
accessibile nella \$rowBand['nome'].

Per poterlo ottenere occorre fare un JOIN tra le tabelle concerti e scalette,
scalette e brani, brani e esecuzioni, esecuzioni e band, band e musicisti.

Successivamente, attraverso opportune operazioni per accedere al Database,
verranno mostrati il 'Nome' e il 'Cognome' di ogni musicista.

```
$queryStrumento
```

```
SELECT DISTINCT strumenti.Nome
FROM strumenti, musicisti, suona
WHERE musicisti.Nome = suona.Nome_Musicista
AND musicisti.Cognome = suona.Cognome_Musicista
AND strumenti.Nome = suona.Strumento
AND musicisti.Nome = '" . $row_musicista['nome'] . "'"
AND musicisti.Cognome = '" . $row_musicista['cognome'] . "'"
```

Questa query permette di selezionare gli strumenti suonati dal musicista in
questione,

accessibile nella variabile \$row_musicista['nome'] e \$row_musicista['cognome'].

Per poterlo ottenere occorre fare un JOIN tra le tabelle musicisti e suona, suona e
strumenti.

Successivamente, attraverso opportune operazioni per accedere al Database,
verranno mostrati il 'Nome' dello strumento.

Curiosity.php

```
$longConcert
```

```

SELECT concerti.Titolo, SEC_TO_TIME(SUM(TIME_TO_SEC(brani.Durata))) AS duratatot
FROM concerti,scalette,brani
WHERE concerti.Titolo = scalette.Concerto
AND brani.Nome = scalette.Brano
GROUP BY concerti.Titolo
HAVING duratatot > ALL (SELECT SEC_TO_TIME(SUM(TIME_TO_SEC(brani.Durata))) AS duratatot
FROM concerti,scalette,brani
WHERE concerti.Titolo = scalette.Concerto
AND brani.Nome = scalette.Brano
GROUP BY concerti.Titolo)

```

Questa query permette di ottenere il titolo del concerto più lungo.

Per ottenerlo occorre fare un JOIN tra le tabelle concerti e scalette, scalette e brani.

E' ottenuto tramite un raggruppamento, infatti vengono sommate le durate di ogni brano, raggruppandole in base al concerto a cui appartengono.

Successivamente attraverso la clausola **HAVING** si cerca tra tutte le durate quale sia quella maggiore. Successivamente, attraverso opportune operazioni per accedere al Database,

verrà mostrato il 'Titolo' del concerto.

\$polystrum

```

SELECT musicisti.Nome, musicisti.Cognome, count() AS numerostrum
FROM musicisti, suona, strumenti
WHERE musicisti.Nome = suona.Nome_Musicista
AND musicisti.Cognome = suona.Cognome_Musicista
AND strumenti.Nome = suona.Strumento
GROUP by musicisti.Nome, musicisti.Cognome
HAVING count() >= ALL (SELECT count(*) FROM musicisti, suona, strumenti
WHERE musicisti.Nome = suona.Nome_Musicista
AND musicisti.Cognome = suona.Cognome_Musicista
AND strumenti.Nome = suona.Strumento
GROUP by musicisti.Nome, musicisti.Cognome)

```

Questa query permette di trovare i musicisti che suonano il maggior numero di strumenti.

Per ottenerli occorre fare un JOIN tra le tabelle musicisti e suona, suona e strumenti.

Il conteggio è ottenuto raggruppando il numero di strumenti suonati per ogni musicista.

Successivamente attraverso la clausola HAVING si cerca tra tutti i musicisti quelli con il numero maggiore.

Successivamente, attraverso opportune operazioni per accedere al Database, verrà mostrato il 'Nome' e il 'Cognome' del musicista.

```
$strum
```

```
SELECT strumenti.nome FROM strumenti, suona, musicisti  
WHERE musicisti.Nome = suona.Nome_Musicista  
AND musicisti.Cognome = suona.Cognome_Musicista  
AND strumenti.Nome = suona.Strumento  
AND musicisti.nome = '" . $row['Nome'] . "'  
AND musicisti.cognome = '" . $row['cognome'] . "'
```

Questa query permette di trovare il nome degli strumenti suonati dal musicista in questione, accessibile in \$row['Nome'] e \$row['cognome']. Per ottenerlo è necessario un JOIN tra musicisti e suona, suona e strumenti. Successivamente, attraverso opportune operazioni per accedere al Database, verrà mostrato il 'Nome' dello strumento.

```
$oldMusician
```

```
SELECT nome, cognome, (2020-YEAR(Nascita)) AS età FROM musicisti  
WHERE Morte IS NULL  
AND (2020-YEAR(Nascita)) >= ALL (SELECT (2020-YEAR(Nascita)) AS età FROM musicisti  
WHERE Morte IS NULL)
```

Questa query permette di trovare il musicista più anziano. Per ottenerlo occorre verificare che il campo morte sia null, successivamente occorre confrontare le età di tutti i musicisti. Successivamente, attraverso opportune operazioni per accedere al Database, verrà mostrato il 'Nome', il 'Cognome' e l'età (ricavata dalla query sottraendo a 2020 l'anno di nascita) del musicista.

```
$band
```

```
SELECT band.nome FROM band, musicisti  
WHERE
```

```
musicisti.band = band.nome  
AND musicisti.cognome = '" . $row['cognome'] . "'  
AND musicisti.nome = '" . $row['nome'] . "'
```

Questa query permette di trovare la band del musicista in questione.
Per ottenerla occorre fare un JOIN tra le tabelle musicisti e band.
Successivamente occorre verificare che il cognome e il nome del musicista siano uguali ai valori presenti in \$row['nome'] e \$row['cognome'].
Successivamente, attraverso opportune operazioni per accedere al Database, verrà mostrato il 'Nome' della band.

```
$diedMusician
```

```
SELECT nome, cognome, (YEAR(Morte)-YEAR(Nascita)) AS età FROM musicisti  
WHERE Morte IS NOT NULL  
AND (YEAR(morte) - YEAR(Nascita)) <= ALL (SELECT (YEAR(morte)-YEAR(Nascita)) AS età  
FROM musicisti  
WHERE Morte IS NOT NULL)
```

Questa query permette di trovare il musicista morto più giovane.
Per ottenerlo occorre verificare che il campo morte sia diverso da null,
successivamente occorre confrontare le età di tutti i musicisti morti.
Successivamente, attraverso opportune operazioni per accedere al Database,
verrà mostrato il 'Nome', il 'Cognome' del musicista.

```
$greatConcert
```

```
SELECT concerti.titolo, concerti.Spettatori FROM concerti  
WHERE concerti.spettatori IS NOT NULL  
AND concerti.Spettatori >= ALL(SELECT spettatori FROM concerti  
WHERE Spettatori IS NOT NULL)
```

Questa query permette di ottenere il titolo e gli spettatori del concerto con più spettatori.
Occorre verificare che il campo spettatori non sia nullo.
Successivamente, attraverso opportune operazioni per accedere al Database,
verrà mostrato il 'Titolo' del concerto e il numero di 'Spettatori'.

Search.php

```
$sql
```

```
SELECT concerti.titolo  
FROM concerti  
WHERE concerti.titolo = '" . $tosearch . "'
```

Questa query permette di selezionare la band avente nome uguale alla stringa \$tosearch.

Successivamente verrà verificato che il risultato restituisca un valore.

SingolaBand.php

```
$sql
```

```
SELECT * FROM band WHERE nome = '" . $_GET['varname'] . "'
```

Questa query permette di selezionare tra le band quella avente nome pari al valore salvato in \$_GET['varname'], determinata dal click su una band nella pagina precedente (band.php).

Successivamente, attraverso opportune operazioni per accedere al Database, verrà mostrato il 'Nome', il 'Genere', la 'Foto' e la 'Descrizione' della band.

```
$query_musicista
```

```
SELECT DISTINCT musicisti.nome, musicisti.cognome  
FROM musicisti, band  
WHERE musicisti.Band = band.Nome  
AND band.nome = '" . $_GET['varname'] . "'
```

Questa query permette di ricavare il nome e il cognome dei musicisti appartenenti alla band in questione.

Per ottenerli occorre fare un JOIN tra band e musicisti e verificare che il nome della band sia uguale al valore presente in &_GET['varname'].

Successivamente, attraverso opportune operazioni per accedere al Database, verrà mostrato il 'Nome' e il 'Cognome' dei musicisti.

```
$queryStrumento
```

```
SELECT distinct strumenti.Nome FROM strumenti, musicisti, suona
WHERE musicisti.Nome = suona.Nome_Musicista
AND musicisti.Cognome = suona.Cognome_Musicista
AND strumenti.Nome = suona.Strumento
AND musicisti.Nome = '" . $row_musicista['nome'] . "'"
AND musicisti.Cognome = '" . $row_musicista['cognome'] . "'"
```

Questa query permette di ricavare lo strumento suonato dal musicista in questione.

Per ottenerlo occorre fare un JOIN tra le tabelle musicisti e suona e suona e strumenti e verificare che nome e cognome musicista siano uguali ai valori resenti in \$row_musicista['Nome'] e \$row_musicista['Cognome'].

Successivamente, attraverso opportune operazioni per accedere al Database, verrà mostrato il 'Nome' dello strumento.