

# appunti di info sintassi php connessione al database

## Inserimento dati

```
inserireutenti.php
1  <?php
2
3  // Recupero i dati inviati dal form tramite il metodo POST
4
5  $id = $_POST['id'];
6  $nome = $_POST['nome'];
7  $cognome = $_POST['cognome'];
8
9
10 $servername = "localhost";
11 $username = "root";
12 $password = "";
13 $dbname = "mydb";
14
15 // creazione connessione
16
17 $conn = mysqli_connect($servername, $username, $password, $dbname);
18
19 // controllo connessione
20
21 if (!$conn) {
22     die("connessione fallita: " . mysqli_connect_error());
23 }
24
25 //Definizione della Query SQL
26
27 $sql = "INSERT INTO utenti (id, nome, cognome)
28 VALUES ('$id', '$nome', '$cognome')";
29
30 //Esecuzione e verifica del risultato
31 //invia effettivamente il comando al database
32 if (mysqli_query($conn, $sql)) {
33
34     //Se l'operazione va a buon fine, stampa un messaggio di successo.
35
36     echo "dati inseriti con successo";
37 } else {
38
39     //Se fallisce (ad esempio se la tabella non esiste), stampa l'errore.
40
41     echo "Errore: " . $sql . "<br>" . mysqli_error($conn);
42 }
43
44 mysqli_close($conn);
45
46 echo "<br><a href='index.html'> Torna al menu";
47
48 >>
```

### 1. Configurazione delle credenziali

Le prime righe definiscono le variabili necessarie per "bussare alla porta" del database.

- **\$servername**: L'indirizzo del server (di solito **localhost** se il database è sullo stesso computer del sito).
- **\$username** e **\$password**: Le tue chiavi di accesso.
- **\$dbname**: Il nome specifico del database su cui vuoi lavorare.

### 2. Apertura della connessione

## PHP

```
$conn = mysqli($servername, $username, $password, $dbname);
```

Qui viene invocata la funzione **mysqli()**. Se tutto è corretto, crea un "ponte" (salvato nella variabile **\$conn**) tra PHP e il database.

### 3. Controllo degli errori

#### PHP

```
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}
```

È fondamentale verificare se la connessione è riuscita. Se **\$conn** è falso, il comando **die()** interrompe immediatamente l'esecuzione del programma e stampa l'errore specifico restituito dal sistema.

### 4. Definizione della Query SQL

#### PHP

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('John', 'Doe', 'john@example.com')";
```

Questa è una stringa di testo scritta in linguaggio SQL. Dice al database: *"Inserisci nella tabella 'MyGuests' un nuovo record con questi dati (John, Doe, email...)"*. In questa fase, l'istruzione è solo una stringa, non è ancora stata eseguita.

### 5. Esecuzione e verifica del risultato

#### PHP

```
if (mysqli_query($conn, $sql)) {  
    echo "New record created successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);  
}
```

La funzione `mysqli_query($conn, $sql)` è il cuore dell'operazione: invia effettivamente il comando al database attraverso il "ponte" creato prima.

- Se l'operazione va a buon fine, stampa un messaggio di successo.
- Se fallisce (ad esempio se la tabella non esiste), stampa l'errore.

6. Chiusura della connessione

**PHP**

```
mysqli_close($conn);
```

Infine, è buona norma chiudere sempre la connessione per liberare risorse sul server una volta terminato il lavoro.

# Stampa dei dati

```
visioneutenti.php
1  <?php
2
3  $servername = "localhost";
4  $username = "root";
5  $password = "";
6  $dbname = "mydb";
7
8  // creazione connessione
9
10 $conn = mysqli_connect($servername, $username, $password, $dbname);
11
12 // controllo connessione
13
14 if (!$conn) {
15     die("connessione fallita: " . mysqli_connect_error());
16 }
17 //richiesta dei dati
18
19 $sql = "SELECT id, nome, cognome FROM utenti";
20
21 //comando viene inviato al database. La risposta (l'elenco dei nomi) viene salvata nella variabile $result
22
23 $result = mysqli_query($conn, $sql);
24
25 //i dati diventano visibili all'utente
26
27 if (mysqli_num_rows($result) > 0) {
28
29     // Controlla se il database ha trovato almeno una persona. Se è vuoto, andrà direttamente a scrivere "0 results".
30     //finché ci sono righe da leggere, il codice ne prende una alla volta e la mette dentro $row
31     while($row = mysqli_fetch_assoc($result)) {
32
33         //stampa a video il contenuto
34
35         echo "id: " . $row["id"]. " - Nome: " . $row["nome"]. " " . $row["cognome"]. "<br>";
36     }
37 } else {
38     echo "0 results";
39 }
40
41 //chiude la connessione
42
43 mysqli_close($conn);
44
45 echo "<br><a href='index.html'> Torna al menu";
46
47 ?>
```

## 1. Preparazione delle credenziali

Le prime righe servono a "presentarsi" al database. È come fornire le chiavi e l'indirizzo di una casa.

- **\$servername = "localhost";**: Dice al codice che il database si trova sullo stesso computer dove gira il sito.
- **\$username = "username";** e **\$password = "password";**: Sono il nome utente e la password per accedere.

- **`$dbname = "myDB";`**: Specifica il nome del database esatto da aprire.

## 2. Apertura della connessione

- **`$conn = mysqli_connect(...)`**: Questo comando prova effettivamente ad aprire la porta del database usando le credenziali scritte sopra. Il risultato viene salvato in **`$conn`**.
- **`if (!$conn) { die(...); }`**: È un controllo di sicurezza. Se la connessione fallisce (es. password errata), il programma si ferma subito e scrive "Connection failed" spiegando l'errore.

## 3. La richiesta dei dati (Query)

- **`$sql = "SELECT id, firstname, lastname FROM MyGuests";`**: Questa è una frase in linguaggio SQL. Significa: "Prendi l'ID, il nome e il cognome di tutti quelli che stanno nella tabella MyGuests".
- **`$result = mysqli_query($conn, $sql);`**: Qui il comando viene inviato al database. La risposta (l'elenco dei nomi) viene salvata nella variabile **`$result`**.

## 4. Mostrare i risultati a video

Questa è la parte dove i dati diventano visibili all'utente.

- **`if (mysqli_num_rows($result) > 0)`**: Controlla se il database ha trovato almeno una persona. Se è vuoto, andrà direttamente a scrivere "0 results".
- **`while($row = mysqli_fetch_assoc($result))`**: Questo è un ciclo. Immagina di avere una lista della spesa: finché ci sono righe da leggere, il codice ne prende una alla volta e la mette dentro **`$row`**.
- **`echo "id: " . $row["id"] . ...;`** Qui il codice stampa a video il contenuto. Prende il valore della colonna "id", "firstname" e

"lastname" e li scrive sulla pagina web, aggiungendo un `<br>` per andare a capo dopo ogni persona.

## 5. Pulizia finale

- `mysqli_close($conn);`: È buona norma chiudere sempre la connessione una volta finito il lavoro, per non sprecare risorse del server.