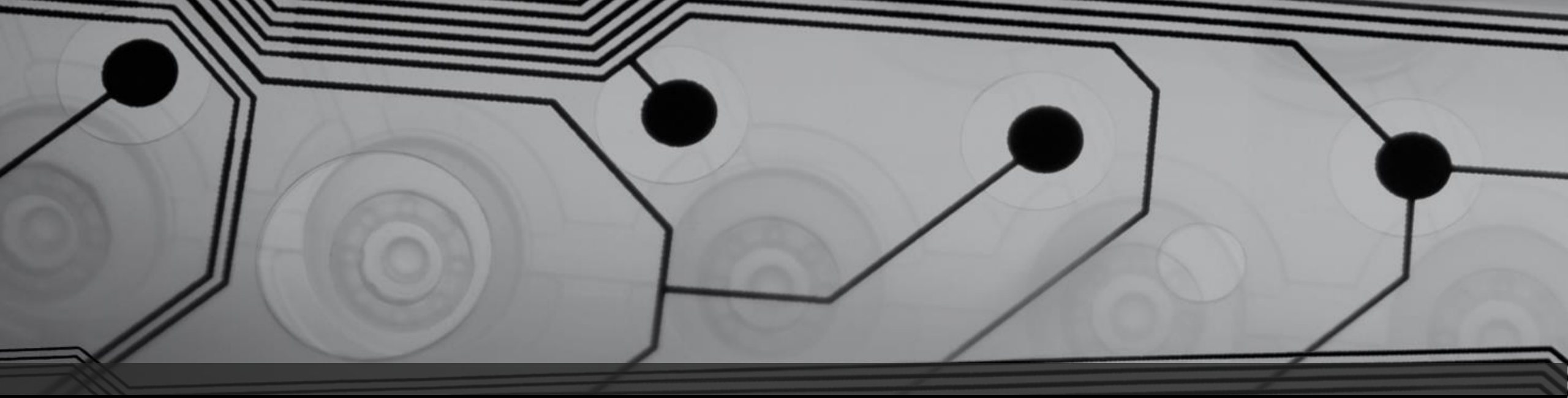


# Interazione con i Database

Prof. Ing. Ivan Gentile

# Possibili interazioni

- Procedurale
- A oggetti
- Le librerie disponibili
  - mysql
  - **mysqli**
  - PDO

https://www.php.net/manual/it/book.mysqli.php

# Lo stile di mysqli

- Basato su 3 classi
- **mysqli**
  - rappresenta la connessione col DBMS MySQL
- **mysqli_result**
  - rappresenta il risultato di una query;
- **mysqli_stmt**
  - rappresenta un comando SQL “parametrico”

# Apertura e chiusura connessione

- Approccio procedurale
  - **resource mysqli_connect(server, user, password, [DB])**
  - **bool mysql_close ([ resource $link_identifier = NULL ] )**
- Approccio ad oggetti
  - **resource new mysqli(server, user, password, [DB])**
  - **bool mysqli::close ( void )**

https://www.php.net/manual/it/mysqli.construct.php

```php
try {
    $mysqli = new mysqli("localhost", 'root', '', 'calcio');
}
catch (mysqli_sql_exception $e) {
  echo "Numero di errore: " . $e->getCode() . "<br>";
  echo "Stringa di errore: " .  $e->getMessage();
  die();
}
echo "OK Connessione stabilita";
```

https://www.php.net/manual/it/mysqli.close.php

```php
$mysqli->close();
```

# Esecuzione Query

- Procedurale
  - **mixed mysqli_query ( mysqli $link , string $query [, int $resultmode = MYSQLI_STORE_RESULT ] )**
- Oggetti
  - **mixed mysqli::query ( string $query [, int $resultmode = MYSQLI_STORE_RESULT ] )**
- Per un successo restituisce
  - un oggetto `mysqli_result (SELECT) o TRUE in caso di successo`
  - `FALSE` in caso di insuccesso

https://www.php.net/manual/it/mysqli.query.php

```php
$query = "Select * from squadre ";

$risultato = @$mysqli->query($query);
```

# Riga da una query

- Stile procedurale
  - `array mysqli_fetch_assoc ( mysqli_result $result )`
  - `mixed mysqli_fetch_row ( mysqli_result $result )`
  - `mixed mysqli_fetch_array ( mysqli_result $result [, int $resulttype = MYSQLI_BOTH ] )`
- Stile ad oggetti
  - `array mysqli_result::fetch_assoc ( void )`
  - `mixed mysqli_result::fetch_row ( void )`
  - `mixed mysqli_result::fetch_array ([ int $resulttype = MYSQLI_BOTH ] )`
- https://www.php.net/manual/it/mysqli-result.fetch-array.php

# Riga da una query: esempio

```php
for ($i = 0; $i < $risultato->num_rows; $i++) {
  //print_r($risultato->fetch_array(MYSQLI_ASSOC));
  $riga = $risultato->fetch_array(MYSQLI_ASSOC);
  echo "<tr>";
  echo "<td>" . $riga['Nome'] . "</td>";
  echo "<td>" . $riga['Città'] . "</td>";
  echo "<td>" . $riga['Indirizzo'] . "</td>";
  echo "<td>" . $riga['CAP'] . "</td>";
  echo "<td>" . number_format($riga['Budget']) . "</td>";
  echo "<td>" . $riga['Abbonati'] . "</td>";
  echo "<td>" . $riga['Serie'] . "</td>";
  echo "</tr>";
}
```

# Rilascio memoria

- Stile procedurale
  - **`void mysqli_free_result ( mysqli_result $result )`**
- Stile ad oggetti
  - **`void mysqli_result::free ( void )`**
  - **`void mysqli_result::close ( void )`**
  - **`void mysqli_result::free_result ( void )`**

# Prepared statements

- Query parametrica (perapred statements)
- Di solito la soluzione da preferire per vari motivi
  - Maggiori performance (quando dobbiamo eseguire la stessa query con parametri diversi)
  - Maggiore chiarezza: non si devono costruire stringhe incastrando stringhe, real_escape_string, variabili
  - Minore rischio di SQL Injection: non c'è bisogno di preoccuparsi di SQL Injection perché i valori recuperati per esempio da una form non vengono incastrati in una stringa ma vengono passati come valori allo statement

# Prepared statement

- Stile ad oggetti: classe **`mysqli_stmt`**
- Stile procedutale: funzioni del tipo **`mysqli_stmt_<nome>`**
  - **`<nome>`** sono gli stessi dei metodi di mysqli_stmt
- Facciamo riferimento allo stile ad oggetti
- Un oggetto della classe `mysqli_stmt` è restituito dal metodo **`prepare`** di `mysqli`

# Attributi e metodi di mysqli_stmt

- https://www.php.net/manual/en/class.mysqli-stmt.php

## The mysqli_stmt class

(PHP 5, PHP 7, PHP 8)

### Introduction

Represents a prepared statement.

### Class synopsis

```
class mysqli_stmt {

   /* Properties */
   public readonly int|string $affected_rows;
   public readonly int|string $insert_id;
   public readonly int|string $num_rows;
   public readonly int $param_count;
   public readonly int $field_count;
   public readonly int $errno;
   public readonly string $error;
   public readonly array $error_list;
   public readonly string $sqlstate;
   public int $id;

   /* Methods */
   public __construct(mysqli $mysql, ?string $query = null)
   public attr_get(int $attribute): int
   public attr_set(int $attribute, int $value): bool
   public bind_param(string $types, mixed &$var, mixed &...$vars): bool
   public bind_result(mixed &$var, mixed &...$vars): bool
   public close(): true
   public data_seek(int $offset): void
   public execute(?array $params = null): bool
   public fetch(): ?bool
   public free_result(): void
   public get_result(): mysqli_result|false
   public get_warnings(): mysqli_warning|false
   public more_results(): bool
   public next_result(): bool
   public num_rows(): int|string
   public prepare(string $query): bool
   public reset(): bool
   public result_metadata(): mysqli_result|false
   public send_long_data(int $param_num, string $data): bool
   public store_result(): bool
}
```

# Parametri di output

- https://www.php.net/manual/en/mysqli-stmt.bind-result.php
- `public mysqli_stmt::bind_result(mixed &$var, mixed &...$vars): bool`
- **`fetch`**: legge una riga alla volta (https://www.php.net/manual/en/mysqli-stmt.fetch.php)
- **`get_result`**: restituisce un oggetto mysqli_result (https://www.php.net/manual/en/mysqli-stmt.get-result.php)

# Parametri di output:fetch

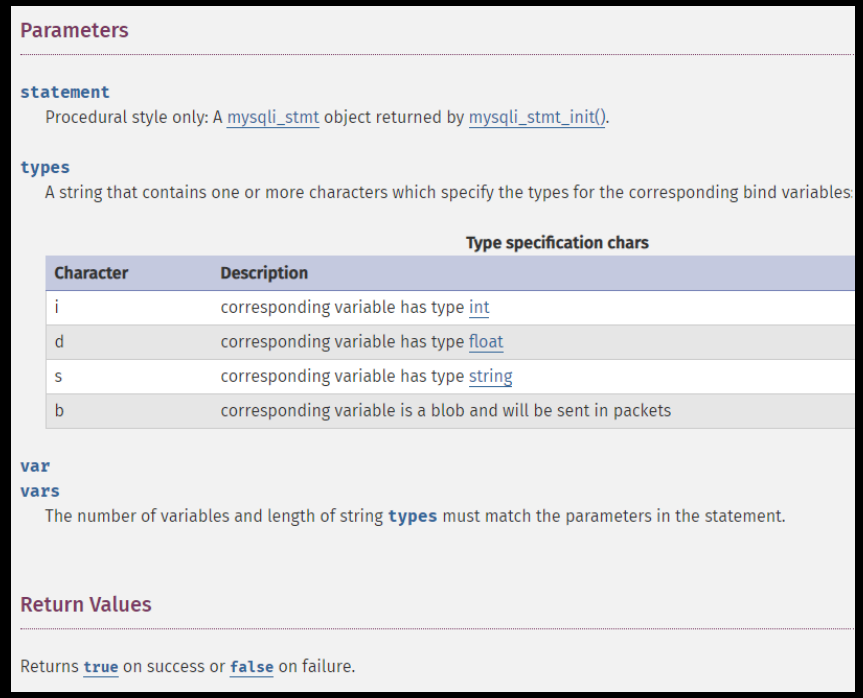
```php
<?php
$mysqli = new mysqli("localhost", "my_user", "my_password", "world");

/* check connection */
if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}

$query = "SELECT Name, CountryCode FROM City ORDER by ID DESC LIMIT 150,5";

if ($stmt = $mysqli->prepare($query)) {

    /* execute statement */
    $stmt->execute();

    /* bind result variables */
    $stmt->bind_result($name, $code);

    /* fetch values */
    while ($stmt->fetch()) {
        printf ("%s (%s)\n", $name, $code);
    }

    /* close statement */
    $stmt->close();
}

/* close connection */
$mysqli->close();
?>
```

# Parametri di output: get_result

```php
<?php

mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
$mysqli = new mysqli("localhost", "my_user", "my_password", "world");

$query = "SELECT Name, Population, Continent FROM Country WHERE Continent=? ORDER BY Name LIMIT 1";

$stmt = $mysqli->prepare($query);
$stmt->bind_param("s", $continent);

$continentList = array('Europe', 'Africa', 'Asia', 'North America');

foreach ($continentList as $continent) {
    $stmt->execute();
    $result = $stmt->get_result();
    while ($row = $result->fetch_array(MYSQLI_NUM)) {
        foreach ($row as $r) {
            print "$r ";
        }
        print "\n";
    }
}
```

# Parametri di input:

- https://www.php.net/manual/en/mysqli-stmt.bind-param.php
- **`public mysqli_stmt::bind_param(string $types, mixed &$var, mixed &...$vars): bool`**



### Parameters

**`statement`**
Procedural style only: A mysqli_stmt object returned by mysqli_stmt_init().

**`types`**
A string that contains one or more characters which specify the types for the corresponding bind variables

**Type specification chars**

| Character | Description |
|---|---|
| i | corresponding variable has type int |
| d | corresponding variable has type float |
| s | corresponding variable has type string |
| b | corresponding variable is a blob and will be sent in packets |

**`var`**
**`vars`**
The number of variables and length of string **`types`** must match the parameters in the statement.

### Return Values

Returns **`true`** on success or **`false`** on failure.

# Parametri di input

```php
<?php

mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
$mysqli = new mysqli('localhost', 'my_user', 'my_password', 'world');

$stmt = $mysqli->prepare("INSERT INTO CountryLanguage VALUES (?, ?, ?, ?)");
$stmt->bind_param('sssd', $code, $language, $official, $percent);

$code = 'DEU';
$language = 'Bavarian';
$official = "F";
$percent = 11.2;

$stmt->execute();

printf("%d row inserted.\n", $stmt->affected_rows);

/* Clean up table CountryLanguage */
$mysqli->query("DELETE FROM CountryLanguage WHERE Language='Bavarian'");
printf("%d row deleted.\n", $mysqli->affected_rows);
```

# Liberare risorse e chiudere lo statement

- **`public free_result(): void`**
  - https://www.php.net/manual/it/mysqli-stmt.free-result.php
- **`public close(): true`**
  - https://www.php.net/manual/it/mysqli-stmt.close.php