


CREARE UN SITO DINAMICO IN PHP

Vediamo come creare un semplice sito in php per la gestione di clienti in un pub, che consenta quindi ad un operatore di inserire, modificare o cancellare i clienti dal database.

Un esempio funzionante è mostrato [qui](#) ().

1. **Pannello di controllo riservato all'operatore**
 - Gestione utenti
 - Motore di ricerca interno

Ovviamente tale applicazione si appoggia su un database MySQL, costituito da una sola tabella: "*pub*", riservata ai dati dei clienti. [Qui](#) trovate il dump del db.

Il codice php necessario per realizzare l'applicazione si articola in diversi files, sia template html e sia files in php.

In questo tutorial, viene usata una logica ad oggetti, per cui tutti i files php saranno delle classi.

Vediamo ora l'elenco di tutte le classi dell'applicazione e contenute nell'apposita cartella ACTION:

- **index.php** : Printa la struttura della pagina e richiama tutte le classi ed i vari template
- **newuser.php** : Inserisce un nuovo utente
- **mod_ut_form.php** : Printa i dati dell'utente su un form di modifica
- **modifyut.php** : Modifica i dati di un utente
- **listall.php** : Stampa una tabella di tutti gli utenti/clienti registrati
- **del_ut.php** : Cancella un utente
- **cerca.php** : Effettua la ricerca su db di un cliente

Vediamo ora l'elenco di tutte le classi contenute nell'apposita cartella CLASSES:

- **MyConn.php** : Contiene le funzioni necessarie per gestire la connessione al database

Mentre le classi contenute nella cartella BEAN sono:

- **pub.php** : Contiene le funzioni necessarie per eseguire le varie query MySQL

Mentre la cartella TPL contiene dei semplici template html, necessari per printare messaggi all'utente ed il form di compilazione di un nuovo cliente.

Pagina "index.php"

Di seguito è riportato il suo codice:

```
<?php
define("ROOT_DIR", realpath("."));
define("TPL_DIR", ROOT_DIR."/tpl/");
define("ACT_DIR", ROOT_DIR."/action/");
define("CLASSES_DIR", ROOT_DIR."/classes/");
define("BEAN_DIR", ROOT_DIR."/bean/");
?>
<HTML>
<HEAD>
<LINK rel="stylesheet" type="text/css" href="img/stile2.css">
<script language="JavaScript" src="img/tigra_tables.js"></script>
<script LANGUAGE="JavaScript" src="img/valida_form.js"></script>
</HEAD>
<BODY>
<TABLE cellSpacing=0 cellPadding=0 width="100%" height="100%" border=0>
<TBODY>
<TR>
<TD class="bodyline">
<TABLE cellSpacing=2 cellPadding=5 width="100%" height="100%" border=0>
<TBODY>
<TR height=10%>
<TD valign=center align=left width="100%" colspan=2 bgcolor=black> <img SRC="img/logo.gif" WIDTH="224"
HEIGHT="57" BORDER=0 ALT=""></TD>
</TR>
<TR height="85%">
<TD valign=top width="15%" bgcolor=black>
<TABLE cellSpacing=1 cellPadding=4 width="100%" height="20%" border=0>
<TR>
<TD><A HREF="index.php?tpl=new_ut">Inserimento</A></TD>
</TR>
<TR >
<TD >
<A HREF="index.php?action=cerca">Ricerca</A></TD>
</TR>
<TR >
<TD><A HREF="index.php?action=listall">Listato</A></TD>
</TR>
<TR height=50>
<TD> </TD>
</TR>
<TR>
<TD valign=bottom align=middle><BR><BR><img SRC="img/centro2.jpg" WIDTH="150" HEIGHT="144" BORDER="0"
ALT=""></TD>
</TR>
<TR height=50>
<TD> </TD>
</TR>
<TR >
<TD><A HREF="index.php?tpl=home">Exit</A></TD>
</TR>
</TABLE>
</TD>
<TD class=tabline valign=top width="75%">
<?
if($_GET["action"])
    $act = $_GET["action"];
elseif($_POST["action"])
    $act = $_POST["action"];
elseif($_GET["tpl"])
    $tpl = $_GET["tpl"];
elseif($_POST["tpl"])
    $tpl = $_POST["tpl"];
else $tpl="home";
if($act)
{
    include_once(ACT_DIR.$act.".php");
    new $act();
}
if($tpl)
{
    include_once(TPL_DIR.$tpl.".html");
}
?>
</TD>
</TR>
```

```

<TR height="5%">
<TD bgcolor=black colspan=2 align=right><span class="text">Powered by Andrea Gentili</span> </TD>
</TR>
</TBODY>
</TABLE>
</TD>
</TR>
</TBODY>
</TABLE>
</BODY>
</HTML>

```

Il codice HTML serve per printare la tabella in cui vengono visualizzati l'header (semplicemente costituito dall'immagine), il menù laterale, il contenuto della pagina ed il footer. Le uniche porzioni di codice php sono quella iniziale e quella centrale: la prima parte serve per definire in maniera univoca i percorsi a partire dalla "index.php" ("**ROOT_DIR**" è il percorso assoluto fino ad essa), attraverso la definizione di costanti, chiamate ad esempio "**PHP_DIR**". *Così facendo, se in un futuro, si vuole cambiare il posto della cartella PHP o rinominarla, basta semplicemente cambiare la sua DEFINE per aggiornare l'applicazione.*

Nella figura sottostante è riportata l'alberatura di tale applicazione, che è da tenere in mente per la definizione dei vari percorsi relativi a quello della index.



Per capire meglio la seconda parte del codice php evidenziato, analizziamo gli HREF dei vari links del menù: ad esempio il primo link, quello dell'inserimento, ha un **HREF="index.php?tpl=new_ut"**, mentre il secondo ha un **HREF="index.php?action=cerca"**. Visto che usando una logica ad oggetti, ogni classe deve essere istanziata per poter essere eseguita, ogni HREF o ACTION presenti nelle varie pagine dovrà essere del tipo:

...="index.php?chiave=nome_file"

Dove la "chiave" può essere o *action* o *tpl* a seconda se si vuole istanziare una classe o richiamare un semplice template html; quindi, mentre la struttura della pagina sarà sempre fissa e stabilita solamente nella index, il contenuto della pagina verrà caricato ogni volta in base alle coppie **chiave=nome_file** che vengono ricevute dalla index o tramite \$_POST (cioè la coppia è presente nell'ACTION di un form) o tramite \$_GET (cioè tramite link). Viene stabilito anche un valore di default, che si verifica quando non sono settati né il \$_POST e né il \$_GET e cioè quando viene caricata per la prima volta la index; questa pagina sarà infatti un template html chiamato "*home.html*". Quindi se è settato \$_POST["action"] o \$_GET["action"], allora viene inclusa e poi istanziata la classe, mentre un template viene semplicemente incluso. Notate come vengono inclusi i file: **COSTANTE.\$tipo_file.estensione**.

Il controllo sugli if è fatto solamente sulle variabili \$act o \$tpl; **if(\$act)**, ad esempio, ritorna true se è settata \$act, altrimenti torna false.

Pagina "MyConn.php"

Di seguito è riportato il suo codice:

```

<?
class MyConn
{
    function closeConnect($conn)
    {
        mysql_close($conn);
    }

    function connessione()
    {
        $conn=mysql_connect("localhost","root","");
        mysql_select_db("pubs");
        return $conn;
    }
}

```

?>

Questa è una classe contenente due sole funzioni, una per connettersi al db ed una per chiudere la connessione. Poiché tale classe non possiede un costruttore, in quanto a noi servono solo le sue due sotto-funzioni, per richiamare, ad esempio, la funzione "connessione" da qualsiasi altra classe, potremmo usare direttamente l'operatore :: , e scrivere quindi **MyConn::connessione()** .

Pagina "pub.php"

Questo è una classe bean, cioè un file chiamato esattamente come la tabella del db a cui si riferisce; come si può vedere dal suo codice, essa contiene tutte le properties necessarie per gestire i relativi campi della tabella "pub" e vari metodi necessari per eseguire tutte le operazioni sulla tabella stessa, quali quelli per inserire, recuperare, modificare o cancellare i suoi dati.

```
<?
class pub
{
    // DEFINIZIONE DELLE PROPERTIES
    var $id;
    var $nome;
    var $cognome;
    var $indirizzo;
    var $cap;
    var $localita;
    var $data_nascita;
    var $data_registrazione;

    // DEFINIZIONE DEL COSTRUTTORE DELLA CLASSE
    function pub($arr=null)
    {
        if(is_array($arr)) $this->fill($arr);
    }

    // DEFINIZIONI DELLE FUNZIONI GET (ottieni un valore)/SET (imposta un valore)
    function getId(){ return $this->id;}
    function setId($value){$this->id=$value;}

    function getNome(){ return $this->nome;}
    function setNome($value){$this->nome=$value;}

    function getCognome(){ return $this->cognome;}
    function setCognome($value){$this->cognome=$value;}

    function getIndirizzo(){ return $this->indirizzo;}
    function setIndirizzo($value){$this->indirizzo=$value;}

    function getCap(){ return $this->cap;}
    function setCap($value){$this->cap=$value;}

    function getLocalita(){ return $this->localita;}
    function setLocalita($value){$this->localita=$value;}

    function getData_nascita(){ return $this->data_nascita;}
    function setData_nascita($value){$this->data_nascita=$value;}

    function getData_registrazione(){ return $this->data_registrazione;}
    function setData_registrazione($value){$this->data_registrazione=$value;}

    // FUNZIONE CHE RICHIAMA LE FUNZIONI SET
    function fill($value)
    {
        foreach($value as $k=>$v)
        {
            $func="set".ucfirst($k);
            if(method_exists($this,$func))
                $this->$func($v);
        }
    }

    // FUNZIONE PER L'INSERIMENTO DI DATI NEL DB
    function DbInsert($conn)
    {
        $query="INSERT INTO pub (nome,cognome,indirizzo,cap,localita,data_nascita,data_registrazione) VALUES('".$this->nome."','".$this->cognome."','".$this->indirizzo."','".$this->cap."','".$this->localita."','".$this->data_nascita."','".$this->data_registrazione."')";
        $result=mysql_query($query, $conn);
    }
}
```

```

// FUNZIONE CHE RECUPERA TUTTI I DATI DI UNA TABELLA
function DbGetAll($conn)
{
    $result = mysql_query("SELECT * FROM pub ORDER BY nome",$conn);
    while($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        $ret[]=$row;
    }
    return $ret;
}

// FUNZIONE CHE RECUPERA UN RECORD SOLO DI DATI
function dbGetOne($conn, $id)
{
    $result = mysql_query("SELECT * FROM pub WHERE id=".$id,$conn);
    while($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        $ret[]=$row;
    }
    return $ret;
}

// FUNZIONE PER LA RICERCA DI DATI
function cerca($conn,$key1,$val1)
{
    $result = mysql_query("SELECT * FROM pub WHERE ".$key1." LIKE ".$val1."%";,$conn);
    while($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        $ret[]=$row;
    }
    return $ret;
}

// CONTROLLO SUI DATI GIA' ESISTENTI
function esiste($conn)
{
    $result = mysql_query("SELECT * FROM pub WHERE nome='".$this->nome.'" AND cognome='".$this->cognome.'"
AND indirizzo='".$this->indirizzo.'" AND cap='".$this->cap.'" AND localita='".$this->localita.'" AND data_nascita='".$this-
>data_nascita.'" AND data_registrazione='".$this->data_registrazione.'"";,$conn);
    if($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        $ret[]=$row;
    }
    return $ret;
}

// FUNZIONA PER L'AGGIORNAMENTO DI DATI GIA' ESISTENTI
function dbUpdate($conn, $id)
{
    $query="UPDATE pub SET nome='".$this->nome."',cognome='".$this->cognome."',indirizzo='".$this-
>indirizzo."',cap='".$this->cap."',localita='".$this->localita."',data_nascita='".$this-
>data_nascita."',data_registrazione='".$this->data_registrazione.'" WHERE id=".$id;
    $result = mysql_query($query,$conn);
}

// FUNZIONE CHE CANCELLA UN RECORD DI DATI (cioè un singolo cliente)
function DbDeleteOne($id, $conn)
{
    $query="DELETE FROM pub WHERE id=".$id;
    mysql_query($query, $conn);
}
}
?>

```

Quindi tale classe possiede delle properties, che come già detto sono delle variabili chiamate con lo stesso nome dei campi della tabella, un costruttore chiamato ovviamente con lo stesso nome della classe e del file bean, il quale non fa altro che richiamare la funzione `fill()` attraverso l'oggetto `$this` solo nel caso in cui esso viene richiamato passandogli come parametro un vettore, quale ad esempio `$_POST`; mentre `$this` si riferisce alla classe stessa, il `"->"` serve per riferirsi alle sue funzioni o properties.

La funzione `fill()` non fa altro che richiamare le varie funzione `setNome_variabile()`, nel caso in cui nell'array che gli viene passato esiste una chiave coincidente con il `Nome_variabile`; ecco perchè questo ci torna molto utile soprattutto quando si effettua ad esempio l'inserimento dei dati sulla tabella. Ad esempio nel form di inserimento di un nuovo cliente, c'è una input del campo nome il cui NAME è proprio pari a "nome", e cioè a come è chiamata la variabile nel file bean (`$nome`) ed esattamente uguale al relativo campo sulla tabella del database. *Per cui quando si invieranno i dati da tale form, `$_POST` conterrà una chiave "nome", il cui valore verrà settato tramite la funzione `fill()` e successivamente dalla `setNome()` alla properties `$nome`.* Questo vale anche per tutte le altre properties; a tal punto, a meno di errori di

distrazione nello scrivere il nome della variabile, esse saranno tutte settate al valore inserito nelle loro relative caselle di testo del form di compilazione.

```

<FORM method="POST" action="index.php?action=NewUser" onSubmit="return validate(this);">
<TABLE width="100%" cellpadding=5 cellspacing=1 border=0>
<TR height=10>
  <TD colspan=2> </TD>
</TR>
<TR height=30>
  <TD width=30%> <SPAN class=text2><B><FONT COLOR="white">*</FONT>Nome :</B></SPAN></TD>
  <TD width=70%> <INPUT TYPE="text" NAME="nome" size=30> <input type="text" name="nomeError" size="30"
class=alert></TD>
</TR>
<TR height=30>
  <TD width=30%> <SPAN class=text2><B><FONT COLOR="white">*</FONT>Cognome :</B></SPAN></TD>
  <TD width=70%> <INPUT TYPE="text" NAME="cognome" size=30> <input type="text" name="cognomeError"
size="30" class=alert></TD>
</TR>
<TR height=30>
  <TD width=30%> <SPAN class=text2><B><FONT COLOR="white">*</FONT>Indirizzo :</B></SPAN></TD>
  <TD width=70%> <INPUT TYPE="text" NAME="indirizzo" size=30> <input type="text" name="indirizzoError"
size="30" class=alert></TD>
</TR>
<TR height=30>
  <TD width=30%> <SPAN class=text2><B><FONT COLOR="white">*</FONT>Cap :</B></SPAN></TD>
  <TD width=70%> <INPUT TYPE="text" NAME="cap" size=30> <input type="text" name="capError" size="30"
class=alert></TD>
</TR>
<TR height=30>
  <TD width=30%> <SPAN class=text2><B><FONT COLOR="white">*</FONT>Località :</B></SPAN></TD>
  <TD width=70%> <INPUT TYPE="text" NAME="localita" size=30> <input type="text" name="localitaError" size="30"
class=alert></TD>
</TR>
<TR height=30>
  <TD width=30%> <SPAN class=text2><B><FONT COLOR="white">*</FONT>Data Nascita :</B><BR> <FONT
COLOR="white"><input type="text" name="data_nascita" size="35" class=alert></TD>
  <TD width=70%> <INPUT TYPE="text" NAME="data_nascita" size=30> <input type="text" name="data_nascitaError"
size="35" class=alert></TD>
</TR>
<TR height=5>
  <TD colspan=2> </TD>
</TR>
<TR height=30>
  <TD><CENTER><INPUT TYPE="submit" value=" Invia " name=ok class=mainoption></CENTER></TD>
  <TD><CENTER><INPUT TYPE="reset" value=" Cancella " class=mainoption></CENTER></TD>
</TR>
</TABLE></FORM>

```

Se ci fosse un'altra casella di testo o un bottone con un NAME impostato e non presente come properties nel file bean, l'applicazione non darebbe comunque un messaggio di errore visto che la `setNome_variabibile` viene richiamata solo se `method_exists()` ritorna true, e quindi solo se esiste una property con quel nome.

Quando viene eseguita la `DbInsert()`, tutte le variabili sono già state settate al momento dell'istanza della classe, per cui la query viene scritta usando proprio tali valori delle properties, e quindi richiamandole attraverso il `$this`; notate come non ci sia bisogno di scrivere `$this->$nome_variabibile`, ma come il "\$" della variabile possa essere tranquillamente omissso.

A differenza della precedente, la `DbGetAll()` esegue la query MySQL e trasforma poi il suo risultato in un array associativo `$ret`, il quale sarà costituito da chiavi numeriche, i cui valori corrispondenti sono gli array riga della tabella in questione sul db. La parola chiave `return` consente a tale funzione di ritornare proprio l'array `$ret` alla classe che chiama la `DbGetAll()` stessa.

Pagina "`NewUser.php`"

```

<?php
include_once(CLASSES_DIR."MyConn.php");
include_once(BEAN_DIR."pub.php");
class NewUser
{
  function NewUser()
  {
    $conn = MyConn::connezzione();
    $pub = new pub($_POST);
    $pub->setData_registrazione(date("Y-d-m"));
    if(!$pub->esiste($conn))
    {
      $pub->DbInsert($conn);
      MyConn::closeConnect($conn);
    }
  }
}

```

```

        header("Location: index.php?tpl=InsertResult");
    }
    else
    {
        MyConn::closeConnect($conn);
        header("Location: index.php?tpl=ControlloResult");
    }
}
?>

```

Questa classe, come anche le altre "ModifyUt.php", "listall.php", "del_ut.php", sono semplicemente costituite da un costruttore, il quale in questa classe esempio, apre la connessione con il db, istanzia il bean passandogli come parametro \$_POST, effettua un controllo sull'esistenza o meno di un cliente già registrato su db con gli stessi dati (la funzione *esiste()* ritorna true se c'è già un utente registrato con gli stessi dati), e nel caso in cui la funzione ritorna False, effettua l'inserimento su db, attraverso la funzione già vista DbInsert(); nel caso contrario manda un header ad una pagina html che printa semplicemente "Cliente già registrato !!".

N.B.: Ricordatevi di chiudere sempre la connessione con il db prima di inviare un header, in quanto quando viene eseguito, tutto il codice ad esso sottostante non viene interpretato.

Pagina "cerca.php"

```

<?
include_once(CLASSES_DIR."MyConn.php");
include_once(BEAN_DIR."pub.php");
class cerca
{
    function cerca()
    {
        $conn = MyConn::connessione();
        $pub = new pub();
        $res=$pub->DbGetAll($conn);
        if($res==null)
        {
            print("<SPAN class=text><BR><BR><BR><CENTER>Non ci sono Clienti
registrati.</CENTER></SPAN><BR><BR>");
        }
        else
        {
            $this->echoform();
            if($_POST["valore"]!=" " && isset($_POST["opzione"]))
            {
                $appo=$pub->cerca($conn,$_POST["opzione"],$_POST["valore"]);
                if($appo==null)
                {
                    print("<SPAN class=text><CENTER>Non ci sono clienti con questo ".$_POST["opzione"].".</CENTER>
</SPAN><BR><BR>");
                }
                else
                {
                    $this->display($appo);
                }
            }
            ?>
            <script language=" JavaScript">
            <!--
            tigma_tables('demo_table', 2, 1, 'black', 'black', '#454545', 'black');
            // -->
            </script>
            <?
            }
        }
    }
    MyConn::closeConnect($conn);
}

function echoform()
{
    echo("<FORM METHOD=POST ACTION='index.php?action=cerca'><TABLE width=100% cellpadding=0
cellspacing=2 border=0 bgcolor=black><TR height=20><TD width=30% align=middle class=row1> <span
class=text2>Cerca un Cliente per:</span></TD><TD valign=top class=row1><TABLE><TR><TD width=20%
align=left><span class=text2> <B>Nome</B></span></TD><TD><INPUT TYPE='radio' NAME='opzione' value='nome'
class=radio></TD></TR><TR><TD><span class=text2> <B>Cognome</B></span></TD><TD><INPUT TYPE='radio'
NAME='opzione' value='cognome' class=radio></TD></TR></TABLE></TD><TD align=middle with=30%
class=row1><INPUT TYPE='text' NAME='valore' size=40></TD><TD align=middle with=20%class=row1><INPUT
TYPE='submit' value=' Visualizza ' class=mainoption></TD></TR></TABLE></FORM><BR>");
}
}

```

```

function display($appo)
{
    echo("<div class=div><TABLE id='demo_table' width=100% cellpadding=5 cellspacing=1 border=1
bordercolor=white><TR height=25><TD colspan=2 align=center bgcolor=#454545><SPAN
class=text> <B>Nome</B> </SPAN></TD><TD align=center bgcolor=#454545><SPAN
class=text> <B>Cognome</B> </SPAN></TD><TD align=center bgcolor=#454545><SPAN
class=text> <B>Indirizzo</B> </SPAN></TD><TD align=center bgcolor=#454545><SPAN
class=text> <B>Cap</B> </SPAN></TD><TD align=center bgcolor=#454545><SPAN
class=text> <B>Località</B> </SPAN></TD><TD align=center bgcolor=#454545><SPAN class=text> <B>Data
Nascita</B> </SPAN></TD><TD align=center bgcolor=#454545><SPAN class=text> <B>Data
Registrazione</B> </SPAN></TD><TD align=center bgcolor=#454545><SPAN class=text><B>MOD.</B></span></TD><TD
align=center bgcolor=#454545><SPAN class=text><B>DEL.</B></span></TD></TR><TR height=5><TD colspan=10
class=row5> </TD></TR>");
    foreach($appo as $k=>$v)
    {
        echo("<TR height=30><TR height=30><TD width=5% align=center><img SRC='img/icon_minipost.gif' WIDTH=12
HEIGHT=9 BORDER=0></TD><TD><SPAN class=text2>". $v["nome"]. "</SPAN></TD><TD><SPAN
class=text2>". $v["cognome"]. "</SPAN></TD><TD><SPAN class=text2>". $v["indirizzo"]. "</SPAN></TD><TD ><SPAN
class=text2>". $v["cap"]. "</SPAN></TD><TD><SPAN class=text2>". $v["localita"]. "</SPAN></TD><TD ><SPAN
class=text2>". $v["data_nascita"]. "</SPAN></TD><TD><SPAN
class=text2>". $v["data_registrazione"]. "</SPAN></TD> <TD align=middle><A
HREF='index.php?action=mod_ut_frm&id=" . $v["id"] . "'><img SRC='img/icon_builder.gif' WIDTH=18 HEIGHT=18
BORDER=0 ALT='Modifica'></A></TD><TD align=middle><A HREF='index.php?action=del_ut&id=" . $v["id"] . "'><img
SRC='img/1916.gif' WIDTH=16 HEIGHT=16 BORDER=0 ALT='Cancella'></A></TD></TR>");
    }
    echo("<TR height=5><TD colspan=10 class=row5> </TD></TR></TABLE></div>");
}
}
?>

```

Il costruttore di tale classe, dopo aver aperto la connessione con il db, recupera tutti i dati presenti nella tabella "pub", e solo nel caso in cui essa non sia vuota (condizione `$res!=NULL`), richiama la funzione esterna `echoForm()` (in quanto esterna al costruttore, si deve usare il `$this` per richiamarla) che printa il modulo di ricerca il cui ACTION punta a questa classe stessa. Quindi la prima volta che viene eseguita tale classe, `$_POST` non è settato e quindi l'esecuzione della classe finisce proprio con la stampa del modulo di ricerca; stessa cosa se si clicca su CERCA senza aver inserito tutti i valori nel modulo di ricerca. Se, invece, si sono scelti sia un valore dell'opzione (Nome o Cognome) ed inserito un valore, allora viene eseguita la ricerca e printato il risultato in una tabella, comprendendo anche i due soliti links per la modifica dei dati del cliente e la sua cancellazione dal db.

La ricerca sul db viene eseguita tramite una semplice query MySQL, di tipo SELECT ed usando l'opzione WHERE campo LIKE valore (`SELECT * FROM pub WHERE ".$key1." LIKE ".$val1."%";`); così facendo vengono ricercati tutti i nomi o cognomi che sono simili al valore immesso, nel senso che se io immetto il valore "a" e se nel db è presente un nome tipo "Andrea", la ricerca va a buon fine, printandomi quel record. Se avessi usato una query del tipo SELECT con l'opzione WHERE campo = valore, immettendo lo stesso valore "a", la ricerca non mi avrebbe printato nessun record a meno che esista una persona registrata su db il cui nome sia veramente "a".