

VINCOLI NON IMPLMENTBILI DIRETTAMENTE IN MYSQL

Risoluzione vincoli V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V15, V27, V19, V20, V21

V1: L'attributo email_utente della relazione utente ha una lunghezza massima di 70 caratteri ed ha il seguente formato '<utente>@<provider>.<dominio>' e può contenere lettere numeri e i seguenti caratteri [- _ .] è case insensitive. [php]

V2: L'attributo password dell'entità utente ha una lunghezza 1 e 50 caratteri e contiene solo caratteri alfanumerici e i seguenti caratteri [_ - ! * .] [php]

V3: L'attributo nickname della relazione è univoco ha una lunghezza tra 1 e 20 caratteri contiene solo caratteri alfanumerici e seguenti caratteri [' " - _ * !.] e spazi [php]

V4: L' attributo nome della relazione info_utente ha una lunghezza massima di 20 caratteri e può contenere solo lettere il carattere di spazio e il carattere di apostrofo. [php]

V5: L' attributo cognome della relazione info_utente ha una lunghezza massima di 20 caratteri e può contenere solo lettere il carattere di spazio e il carattere di apostrofo. [php]

V6: L'attributo data_nascita della relazione info_utente deve essere minore o uguale alla data corrente nel momento in cui viene inserita. [php]

V7: L'attributo nazione_nascita della relazione info_utente ha una lunghezza massima di 50 caratteri e contiene solo lettere il carattere di spazio e il carattere di apostrofo. [php]

V8: L'attributo regione_nascita della relazione info_utente ha una lunghezza massima di 50 caratteri e contiene solo lettere il carattere di spazio e il carattere di apostrofo. [php]

V9: L'attributo sesso della relazione info_utente può assumere uno tra i seguenti valori {'f', 'm'} dove 'm' indica che si tratta di un utente di sesso maschile mentre 'f' indica un utente di sesso femminile. [php]

V10: L'attributo tipo_hobby della relazione hobby può assumere i seguenti valori {'sport','musica','arte','tecnologia','film e serie tv','fotografia','moda e design','videogames', 'viaggiare'} l'utente può inserire un hobby aggiuntivo tramite

una form testuale che non superi i 20 caratteri e che contiene solo lettere il carattere di spazio e il carattere di apostrofo. [php]

V11: L'attributo citta_residenza della relazione info_utente ha una lunghezza massima di 50 caratteri e contiene solo lettere il carattere di spazio e il carattere di apostrofo. [php]

V15: L'attributo messaggio della relazione testo ha una lunghezza massima di 50 caratteri e minima di 1 carattere. [php]

V19: L'attributo latitudine della relazione luogo è un valore a virgola mobile compreso tra -180.000000 e 180.000000. [php]

V20: L'attributo latitudine della relazione luogo è un valore a virgola mobile compreso tra -180.000000 e 180.000000. [php]

V21: L'attributo descrizione_foto della relazione foto ha una lunghezza massima di 20 caratteri. [php]

V27: L'attributo commento della relazione giudizio_foto ha una lunghezza massima di 50 caratteri e minima di 1 carattere. [php]

I vincoli sopra elencati sono stati risolti a livello applicativo con l'utilizzo di una serie di funzioni definite nella cartella nel file:

- /applicativo_php_cinguettio/function/funCheckInput

All'interno di questa cartella si possono trovare tutte le funzioni per il controllo dell'input . Qui si riporta un breve elenco di alcune funzioni. Per vedere l'elenco completo di funzioni utilizzate vedere il file sopra indicato.

Vincolo 1:

```
/**
 * 1) Questa funzione prende in ingresso una stringa e verifica che sia un indirizzo email
 * nel corretto formato In caso negativo restituisce false
 *
 * @param string $value
 * @return boolean
 */
function emailFormat ($value){

    $len = strlen($value);
    if(!preg_match(VALID_EMAIL_FORMAT,$value) || $len > 70){
        return false;
    }

    return true;
}
```

Vincolo 2:

```
/**
 * Questa funzione prende in ingresso una stringa e verifica che la password sia in un
 * formato corretto In caso negativo restituisce false
 *
 * @param string $value
 * @return boolean
 */
function pwdFormat ($value){

    if(!preg_match(VALID_PWD_FORMAT,$value)){
        return false;
    }

    return true;
}
```

Vincolo 3:

```
/**
 * Questa funzione prende in ingresso una stringa e verifica che il nickname sia in un
 * formato corretto in caso negativo restituisce false
 *
 *
 * @param string $value
 * @return boolean
 */
function nickFormat ($value){

    if(!preg_match(VALID_NICK_FORMAT,$value)){
        return false;
    }

    return true;
}
```

Vincolo 4 e 5:

```
/**
 * Questa funzione prende in ingresso una stringa e verifica che il nome o il cognome di un
 * utente siano
 * in un formato corretto
 * In caso negativo restituisce false
 *
 * @param string $value
 * @return boolean
 */
function nameFormat ($value){

    if(strlen($value) == 0){
        return true;
    }

    else if(!preg_match(VALID_NAME_FORMAT,$value)){
        return false;
    }

    return true;
}
```

Vincolo 6:

```
/**
 * Questa funzione prende in ingresso una data di nascita e verifica che tutti i
 * campi della data che siano a null minore della data corrente in caso positivo restituisce
 * true altrimenti false
 *
 * @param string $data_nascita
 * @return boolean
 */
function checkDataNascita($data_nascita){
    $current_data = date('Y/m/d');
    if($data_nascita > $current_data && $data_nascita != null){
        return false;
    }
    return true;
}
```

Vincolo 7, 8, 11:

```
/**
 * Questa funzione prende in ingresso una stringa e verifica che nazione_nascita, *
 * citta_residenza o
 * regione_nascita siano in un formato corretto
 * in caso negativo restituisce false
 *
 * @param string $value
 * @return boolean
 */
function placeFormat ($value){

    if(strlen($value) == 0){
        return true;
    }

    else if(!preg_match(VVALID_PLACE_FORMAT,$value)){
        return false;
    }

    return true;
}
```

Vincolo 9:

```
/**
 * Questa funzione prende in ingresso una stringa e verifica che il sesso inserito
 * dall'utente
 * sia in un formato corretto
 * in caso negativo restituisce false
 *
 * @param string $value
 * @return boolean
 */
function sexFormat ($value){

    if(strlen($value) == 0){
        return true;
    }

    else if(!preg_match(VALID_SEX_FORMAT,$value)){
        return false;
    }

    return true;
}
```

Vincolo 10:

```
/**
 * Questa funzione prende in ingresso una stringa e una regex e controlla che l'hobby sia
 * in un formato corretto definito dalla regex in caso negativo restituisce false
 *
 * @param string $value
 * @param string $regex
 * @return boolean
 *
 */
function hobbyFormat ($value, $regex){

    if(strlen($value) == 0){
        return true;
    }

    else if(!preg_match($regex,$value)){
        return false;
    }

    return true;
}
```

Vincolo 15, 27:

```
/**
 * Questa funzione prende in ingresso una stringa e verifica che il testo
 * inserito dall'utente
 * sia in un formato corretto
 * in caso negativo restituisce false
 *
 * @param string $value
 * @return boolean
 */
function textFormat ($value){

    if(!preg_match(VALID_TEXT_FORMAT,$value)){
        return false;
    }

    return true;
}
```

Vincolo 19, 20:

```
/**
 * 12) Questa funzione prende in ingresso un coordinata espressa in GMS(gradi minuti e
 * secondi)
 * e la converte in decimali
 *
 * @param string $gradi
 * @param string $minuti
 * @param string $secondi
 * @param string $direzione
 * @return float
 */
function GMStoDecimal($gradi, $minuti, $secondi, $direzione){

    $decimal = $gradi + ($minuti / 60.0) + ($secondi / 3600.0);

    if($direzione == 's' || $direzione == 'w'){
        $decimal *= -1;
    }

    return $decimal;
}
```

Vincolo 21:

```
/**
 * Questa funzione prende in ingresso una stringa e verifica che il testo di
 * descrizione_foto
 * sia in un formato corretto
 * in caso negativo restituisce false
 *
 * @param string $value
 * @return boolean
 */
function descrFormat ($value){

    if(strlen($value) == 0){
        return true;
    }

    else if(!preg_match(VALID_DESCR_FORMAT,$value)){
        return false;
    }

    return true;
}
```

Tutti le funzioni sopra indicate utilizzano diverse costanti, le quali sono reperibili nel file

- `/applicativo_php_cinguettio/config/config.php`

Alcune delle costanti definite si possono trovare qui in descrizione per un elenco completo delle costanti vedere il file sopra indicato

```
/*
 * COSTANTI PER LA VALIDAZIONE DELL'INPUT
 */
// definizione della regex per il corretto formato della mail
define('VALID_EMAIL_FORMAT','/^[A-Za-z0-9_\-.]+@([A-Za-z0-9]+)[.]( [A-Za-z]{2,4})$/');
// definizione della regex per il corretto formato della password
define('VALID_PWD_FORMAT','/^[A-Za-z0-9\p{L}\à\è\ù\ò\é\i\_-!*.]{1,50}$/');
// definizione della regex per il corretto formato del nickname
define('VALID_NICK_FORMAT','/^[A-Za-z0-9\p{L}\à\è\ù\ò\é\i\_-\'\"s!*.]{1,20}$/');
// definizione della regex per il corretto formato di nome e cognome
define('VALID_NAME_FORMAT','/^[A-Za-z\p{L}\à\è\ù\i\ò\é\'s]{1,20}$/');
// definizione della regex per il corretto formato di nazione_nascita, regione_nascita,
citta_residenza
define('VALID_PLACE_FORMAT','/^[A-Za-z\p{L}\à\è\ù\i\'s]{1,50}$/');
// definizione della regex per il corretto formato degli hobby
define('VALID_HOBBY_1_FORMAT','/^sport$/');
define('VALID_HOBBY_2_FORMAT','/^musica$/');
define('VALID_HOBBY_3_FORMAT','/^arte$/');
define('VALID_HOBBY_4_FORMAT','/^tecnologia$/');
define('VALID_HOBBY_5_FORMAT','/^film e serie tv$/');
define('VALID_HOBBY_6_FORMAT','/^fotografia$/');
define('VALID_HOBBY_7_FORMAT','/^moda e design$/');
define('VALID_HOBBY_8_FORMAT','/^videogames$/');
define('VALID_HOBBY_9_FORMAT','/^viaggiare$/');
define('VALID_HOBBY_10_FORMAT','/^[A-Za-z\p{L}\à\è\ù\ò\é\i\'s]{1,20}$/');
// definizione della regex per il corretto formato del sesso
define('VALID_SEX_FORMAT','/^[mf]$/');
// definizione della regex per il corretto formato dei messaggi di testo e dei commenti alle foto
define('VALID_TEXT_FORMAT','/^(.|\n){1,50}$/');
// definizione della regex per il corretto formato della descrizione della foto
define('VALID_DESCR_FORMAT','/^(.|\n){1,20}$/');
```

Risoluzione vincolo V18

V18: Ogni utente può segnalare un cinguettio di tipo testo di un altro utente che segue come “inappropriato”. Un utente non può segnalare un proprio cinguettio di tipo testo come “inappropriato” né un cinguettio che ha già segnalato come “inappropriato”.

```
/**
 * Questa funzione prende in ingresso un riferimento ad un array e una connessione al
 * database
 * e stampa il post di testo
 *
 * @param array $row
 * @param mixed $conn
 * @return no return
 */
function printTesto(&$row, $conn){
    .
    .
    .
    // verifico se il post è stato pubblicato dall' utente della sessione in caso non lo
    // sia inserisco i pulsanti
    // per la segnalazione del testo
    if($row['email_utente'] != $_SESSION['email_utente']) {

        //query per verificare se l'utente ha già segnalato il messaggio di testo
        $sql = queryTestoSegnalato($row['email_utente'], $row['id_cinguettio']);
        $result = mysqli_query($conn, $sql);

        // se il messaggio è stato già segnalato dall'utente stampo un bottone che
        // se premuto fa visualizzare un messaggio di errore
        // verifico se il post è stato pubblicato dall utente della sessione in caso
        //non lo sia inserisco i pulsanti
        // per la segnalazione del testo
        if($row['email_utente'] != $_SESSION['email_utente']) {

            //query per verificare se l'utente ha già segnalato il messaggio di testo
            $sql = queryTestoSegnalato($row['email_utente'], $row['id_cinguettio']);
            $result = mysqli_query($conn, $sql);

            // se il messaggio è stato già segnalato dall'utente stampo un bottone che
            // se premuto
            // fa visualizzare un messaggio di errore
            if(mysqli_num_rows($result) == TRUE){
                echo '<button class="segnala-testo" onclick="showModal(';
                echo '"testo-errore-max-segnalazioni','post-container'";
                echo ')"> Segnala testo</button>';
            }
            else{
                . . . . .
            }
        }
    }
}
```

La funzione completa e la funzione per le query si trovano nei file

- /applicativo_php_cinguettio/function/funBacheca.php
- /applicativo_php_cinguettio/query/queryBacheca.php

Risoluzione vincolo V26

V26: Un utente esperto può esprimere un giudizio di apprezzamento per i cinguettii di tipo foto pubblicati solo da utenti che segue o per i propri cinguettii di tipo foto. L'utente esperto non può esprimere un giudizio di apprezzamento più di 3 volte per uno stesso cinguettio di tipo foto.

```
/**
 * Questa funzione prende in ingresso un riferimento ad un array e una connessione
 * al database e stampa il post di foto
 *
 * @param array $row
 * @param mixed $conn
 * @return no return
 */
function printFoto(&$row, $conn, $utente_esperto){

    //elenco dei commenti
    $sql = queryCommenti($row['email_utente'], $row['id_cinguettio']);

    //effettuo l'interrogazione per estrarre i commenti alla foto
    // memorizzo il numero di commenti effettuati
    $commenti = mysqli_query($conn, $sql);
    $num_commenti = mysqli_num_rows($commenti);

    //query per il numero massimo di commenti di commenti effettuati dall'utente per
    //il cinguettio
    $sql = queryNumeroMaxCommenti($row['email_utente'], $row['id_cinguettio']);

    //effettuo l'interrogazione e memorizzo il numero massimo di commenti effettuati
    //dall'utente
    //per il cinguettio
    $result = mysqli_query($conn, $sql);
    $sql_result = mysqli_fetch_assoc($result);
    $max_commenti_utente = $sql_result['num_commenti'];

    . . . . .

    // verifico se l'utente ha già inserito più di tre commenti
    if($max_commenti_utente >= 3){
        echo '<button onclick="showModal(' . "'ins-errore-max-commenti','post-container'"
            . ')" class="inserimento-commento">';
        echo 'Inserisci un commento </button>';
    }
    // verifico se l'utente è esperto
    else if($utente_esperto == FALSE){
        echo '<button onclick="showModal(' . "'ins-errore-no-esperto','post-container'"
            . ')" class="inserimento-commento">';
        echo 'Inserisci un commento </button>';
    }

    . . . . .
}
```

La funzione completa e la funzione per le query si trovano nei file

- /applicativo_php_cinguettio/function/funBacheca.php
- /applicativo_php_cinguettio/query/queryBacheca.php

Risoluzione vincolo V26

V28: L'attributo data_ora_commento della relazione giudizio_foto corrisponde alla data e all'ora corrente nel momento in cui viene commentata una foto. [php]

```
// query per l'inserimento di un commento di un utente ad uno specifico post
function queryInsCommento($utente_giudicato, $id_cinguettio, $commento){

    $sql = 'INSERT INTO giudizio_foto(email_utente, utente_giudicato, id_cinguettio, ' .
        'data_ora_commento, commento)' .
        'VALUES(' . "'" . $_SESSION['email_utente'] . "', " .
        '$utente_giudicato . "', " . $id_cinguettio . "', " . 'DEFAULT, ' .
        '$commento . '))';

    return $sql;
}
```

Risoluzione vincolo V29

V29: Un utente può definire come luogo preferito un proprio cinguettio di tipo luogo o un cinguettio di tipo luogo pubblicato da un altro utente che segue ma non può definire uno stesso cinguettio di tipo luogo come luogo preferito più di 1 volta. [php]

```
// query per sapere se l'utente ha già definito il luogo come luogo preferito
function queryLuogoPreferito ($utente_luogo, $id_cinguettio){

    $sql = 'SELECT * ' .
        'FROM luogo_preferito ' .
        'WHERE utente_luogo=' . "'" . $utente_luogo . "'" . ' AND ' .
        'email_utente=' . "'" . $_SESSION['email_utente'] . "'" . ' ' .
        'AND ' .
        'id_cinguettio=' . $id_cinguettio;

    return $sql;
}
```

Risoluzione vincolo V30

V30: Un utente non può seguire sé stesso né può inviare una richiesta di seguire un utente che già segue

Il fatto che un utente non possa seguire se stesso lo gestiamo impedendo che nella ricerca utente abbia un qualsiasi tipo di pulsante per. Se un utente loggato in sessione prova a cercarsi viene rediretto nel proprio profilo

```
// stampo elementi solo se ci il numero di risultati
// è maggiore di 0 e non stampo nel caso
// dell'utente stesso
while($row = mysqli_fetch_assoc($result)){
    if(($row['utente_esperto'] == TRUE) &&
        ($row['email_utente'] == $_SESSION['email_utente'])){
        printUtenteEsperto($row);
    }
    else if (($row['utente_esperto'] == FALSE) &&
        ($row['email_utente'] != $_SESSION['email_utente'])){
        printUtente($row);
    }
}
```

Per una descrizione più dettagliata vedere i file

- /applicativo_php_cinguettio/Visual/ricercaUtente.php
- /applicativo_php_cinguettio/function/funRicerca.php
- /applicativo_php_cinguettio/query/queryRicercaUtente.php

Risoluzione vincolo V31

V31: L'attributo data_ora_richiesta della relazione seguaci_seguiti corrisponde alla data e all'ora corrente nel momento in cui viene inviata una richiesta di seguire un utente.

```
// query per iniziare a seguire un utente
function querySegui($utente_seguito){

    $sql = 'INSERT INTO seguaci_seguiti(utente_seguace, utente_seguito) ' .
        'VALUES(' . "''" . $_SESSION['email_utente'] . "''" . ',' .
        $utente_seguito . ')';

    return $sql;
}
```

Risoluzione vincolo V32, V33, V34, V35

V32: L'attributo data_ora_richiesta della relazione seguaci_seguiti corrisponde alla data e all'ora corrente nel momento in cui viene inviata una richiesta di seguire un utente.

V33: L'attributo richiesta_accettata della relazione seguaci_seguiti è un valore booleano che assume valore 'TRUE' se la richiesta è stata accettata altrimenti assume valore 'FALSE' se la richiesta non è ancora stata accettata

V34: Un utente per poter vedere i cinguettii pubblicati da un altro utente accedere alle sue informazioni personali deve essere un suo seguace

V35: Un utente non può chiedere ad un altro utente di seguirlo se ha già inviato in precedenza una richiesta a cui non è stata data ancora una risposta

Vincolo 32

```
// query per iniziare a seguire un utente
function querySegui($utente_seguito){

    $sql = 'INSERT INTO seguaci_seguiti(utente_seguace, utente_seguito) ' .
           'VALUES(' . "''" . $_SESSION['email_utente'] . "''" . ',' .
           $utente_seguito . ')';

    return $sql;

}
```

Vincolo 33

```
if ($richiesta == 1 ){

    $sql = "UPDATE seguaci_seguiti SET richiesta_accettata = 1 ,
data_ora_risposta = now()
           WHERE utente_seguace = '{$utente_seguace}' and utente_seguito =
 '{$utente}'";

    mysqli_query($conn, $sql);
    $page = 'richieste.php';
    redirect($page);

}
```

Vincolo 34

Guardare la funzione queryPostList() nel file

- **/applicativo_php_cinguettio/query/queryBacheca.php**

Vincolo 35

Viene risolto a livello grafico togliendo il pulsante di invio richiesta e viene inserito al suo posto un messaggio che segnala la data di invio della richiesta vedere per maggiori informazioni il file

- **/applicativo_php_cinguettio/visual/ricercaUtente.php**