Multiselezione con più pagine

Una Griglia Multiselect serve per rendere selezionabili più di una Riga alla volta.
E' possibile però che capiti di avere una Griglia con molte righe e che ciò renda i tempi di creazione/aggiornamento della tabella molto lunghi.
Il problema può essere risolto utilizzando una Multiselect con più Pagine, ma dato che non è ancra implementato un sistema di *cattura delle righe precedentemete selezionate* dovremo utilizzare la funzione **MultiSelezionePost**

MultiSelezionePost

Con questa funzione potremo:

- tenere memorizzate tutte le righe precedentemente selezionate.
- visualizzare in ogni pagina le righe già selezionate
- deselezionare le righe

Per poter utilizzare questa funzione avremo bisogno prima di tutto di una **variabile in Session** che conterrà tutte le righe precedentemente selezionate e il nome della griglia: Le dichiariamo:

```
    ..
    public $gridLicenze = "gfmConLicenze_gridLicenze";
    public $SelezioneTab = array();
    ..
```

Nel construct:

```
function __construct() {
        parent::__construct();
        try {
            $this->GAFIERE_DB = ItaDB::DBOpen('GAFIERE');
            $this->SelezioneTab = App::$utente->getKey($this->nameForm .
'_SelezioneTab');
            ...
```

Nel destruct:

```
function __destruct() {
        parent::__destruct();
        if ($this->close != true) {
            App::$utente->setKey($this->nameForm . '_SelezioneTab',
$this->SelezioneTab);
            ...
```

Nel close:

```
    public function close() {
        App::$utente->removeKey($this->nameForm . '_SelezioneTab');
        Out::closeDialog($this->nameForm);
    }
```

Dopo di che chiamiamo la funzione nel **onClickTablePager** :

```
...
$sql = $this->CreaSql();
$this->SelezioneTab=$this->gfmLib->MultiSelezionePost($this->SelezioneTab,$t
his->gridLicenze,$sql);
...
```

Notiamo:

- L'assegnazione dell'**sql**;
- L'assegnazione di **$this→SelezioneTab**
- Il richiamo della funzione **MultiSelezionePost** con l'utilizzo di 3 variabili:
  - $this→SelezioneTab: Utiliziamo la variabile in session in quanto contiene i campi già selezionati (ovviamente se si tratta del primo richiamo sarà vuota)
  - $this→gridLicenze: rappresenta il nome della griglia, in questo caso: **gfmConLicenze_gridLicenze**
  - $sql: la stessa sql che utilizziamo per creare la griglia, in modo che la funzione che stiamo chiamando si scorra gli stessi campi e possa controllare i ROWID selezionati.
    In questo caso noi utilizziamo un SELECT * FROM DITTELIC

La funzione:

```
public function MultiSelezionePost($SelezioneTab,$Griglia,$sql){
$Riga='jqg_'.$Griglia.'_';
// PRIMA PARTE
      foreach ($SelezioneTab as $key => $Result_rec) {
        if($_POST[$Riga.$key] == '0'){
             $_POST[$Riga.$key] = '0';
             $SelezioneTab[$key ] = '0';
            }else{
               if(!Out::valore($Riga.$key,'1')){
                  $_POST[$Riga.$key]='1';
                    }
                  }
                 }
// SECONDA PARTE
       $Griglia_tab=ItaDB::DBSQLSelect($this->getGAFIEREDB(),$sql,
true);//<- Cambiare il DB
        foreach ($Griglia_tab as  $Griglia_rec) {
            if( $_POST[$Riga.$Griglia_rec['ROWID']] == '1'){
                $NuovaSelezione[$Griglia_rec['ROWID']]='1';
                Out::valore($Riga.$Griglia_rec['ROWID'],'1');
             }else {
```

```
                if($SelezioneTab[$Griglia_rec['ROWID']] == '1'){
                          $NuovaSelezione[$Griglia_rec['ROWID']]='0';
                        }
                    }
                 }

       return $NuovaSelezione;
   }
```

La prima parte della funzione invia in **POST** anche le **row** selezionate nella pagina precedente (ovviamente se è la prima pagina non verrà eseguita)

Notiamo:

- le 3 variabili precedentemente descritte: **$SelezioneTab,$Griglia,$sql**
  - **$SelezioneTab:** $this→SelezioneTab
  - **$Griglia:** gfmConLicenze_gridLicenze
  - **$sql:** SELECT * FROM DITTELIC
- **jqg_'.$Griglia.'_'.N:** (*jqg_gfmConLicenze_gridLicenze_N*) rappresenta nel post le righe presenti nella pagina della tabella. Esso può assumere un valore tra 1 e 0 (Check o non Check). N rappresenta il numero della riga (Rowid)

Nella seconda parte della funzione:

- viene creato l'Array **$Griglia_tab**, che conterrà tutte le righe di **DITTELIC**. Dopo di che con un ciclo lo scorriamo e verifichiamo se tra i post della pagina sono presenti una o più righe che hanno valore 1 (Checked):
  - Se ci sono vengono aggiunte all'array provvisorio **$NuovaSelezione** e vengono valorizzate (Checked nella griglia)
  - Se non ci sono, controlla se tra le righe nella pagina sono presenti righe precedentemente selezionate (Quindi presenti in **$SelezioneTab**) e le assegna a 0. In questo modo non risulteranno più selezionate.
- In fine avremo il return di **$NuovaSelezione** che rappresenta l'array aggirnato con tutte le righe selezionate in precedenza.

Nel nostro programma avremo quindi la variabile in session $this→SelezioneTab costantemente aggiornato ad ogni evento legato alla tabella.

Nota

Con questa funzione è possibile selezionare oltre 2000 righe, ma nel caso in cui ce ne siano di più è opportuno inserire un bottone che permetta di stamparle direttamente tutte e inserire dei filtri nella ricerca(in modo da *smaltire* le righe in più).

Per l'output delle righe selezionate, possiamo utilizzare un semplice ciclo:

```
foreach ($SelezioneTab as $key => $Result_rec) {
                  $SelezioneGrid[]=$key;
               }
       $sql=" SELECT *
              FROM DITTELIC
```

```
              WHERE 1";
        $sql.=" AND(";
        for ($index = 0; $index < count($SelezioneGrid); $index++) {
             $sql.=" ROWID = $SelezioneGrid[$index] OR";
         }
        $sql=substr($sql,0,-3);
        $sql.=" )";
```

$SelezioneTab è l'Array che contiene le righe selezionate.
Ovviamente, se vogliamo passare una $SelezioneTab aggiornata, prima del ciclo, dovremo richiamare nuovamente la funzione MultiSelectPost, in modo che se nell'ultima pagina visualizzata sono state selezionate una o più righe, vengano aggiunte anch'esse all'array.

Con il ciclo, rielaboriamo l'array in modo da ottenere le varie $key,che sono l'indice(ROWID), in sequenza.
Dopo di che, in questo esempio, all'sql aggiungiamo la condizione *ROWID = N OR* per ogni elemento dell'array, in modo da selezionare tutti gli indici(ROWID) che corrispondono a quelli selezionati.
Avremo un risultato del genere:

SELECT * FROM DITTELIC WHERE 1 AND(ROWID = 2 OR ROWID = 3 OR ROWID = 35)