

## Inserimento di un indice Specifico in Tabella : ReaderId

Il ReaderId è l'indice che identifica in modo univoco una riga in una Grid.

Se nella querry che viene passata alla Grid non è presente un Indice, il ReaderId viene automaticamente assegnato e assume valori in ordine Crescente.

Esempi:

1)

```
SELECT * FROM DIPENDENTI
```

In questo caso abbiamo preso tutto, compreso l'indice, e l'indice della griglia che verrà creata sarà lo stesso della tabella DIPENDENTI

2) Se invece prendiamo solo determinati campi dalla tabella DIPENDENTI:

```
SELECT CODICEDIPENDENTE,COGNOME,NOME FROM DIPENDENTI
```

L'indice della griglia che verrà creata sarà assegnato automaticamente in ordine crescente.

Se nella griglia si vuole utilizzare un Indice differente dall'indice definito nella tabella sarà necessario definire un ReaderId :

- 1) Nel model della grid inserire l'attributo: **readerId:'Key'**

Dove 'Key' rappresenta il nome della nostra chiave/indice.

Esempio:

Nome Attributo	Valore Attributo
class	{cellEdit:false, readerId:'KEY', caption: "Ditta", shrinkToFit:true, width:650 .....

- 2) Nel programma, al momento della creazione della griglia, dovremo aggiungere all'array il 'rowIndex'=>'Key',(Sempre utilizzando la stessa Key definita nel Model) per specificare quale sarà l'indice che dovrà utilizzare:

```
$sql=$this->CreaSql();
$ita_grid01 = new TableView($this->gridDitta,
                           array(
                                'sqlDB' => $this->DITTA_DB,
                                'sqlQuery' => $sql,
                                'rowIndex' => 'KEY'));
$ita_grid01->setPageNum(1);
$ita_grid01->setPageRows(20000);
$ita_grid01->setSortIndex('COGNOME');
$ita_grid01->setSortOrder('asc');
....
```

Se nella select quindi specificheremo ad esempio:

```
SELECT *, CODICEDIPENDENTE AS KEY FROM DIPENDENTI
```

Avremo come indice il Codice del Dipendente.

### Elabora Record

Nel Caso in cui vogliamo Elaborare il risultato della nostra tabella per un qualsiasi motivo, come ad esempio l'evidenziare in rosso il numero di telefono di un dipendente e in verde quello di cellulare, o una qualsiasi altro tipo di rielaborazione, possiamo ricorrere alla Funzione "Elabora Record". Questo tipo di funzione ci permette di **Rielaborare** l'Output che otterremo:

Posto	Mercato	Tipo	II. Ass.
	MERCATO POTENZA PICENA	NORMALE	5
34	MERCATO POTENZA PICENA	NORMALE	12
	MERCATO POTENZA PICENA	NORMALE	2

Per ottenere un'elaborazione di record dovremo:

- 1) Creare la Funzione di rielaborazione:

```
function elaboraRecords($Result_tab) {
    foreach ($Result_tab as $key => $Result_rec) {
        $Result_tab[$key]['TELEFONO'] = "<p style = 'background-color:#FF00FF;'>".$Result_tab[$key]['TELEFONO']."</p>";
        $Result_tab[$key]['CELLULARE'] = "<p style = 'background-color:#008800;'>".$Result_tab[$key]['CELLULARE']."</p>";
    }
    return $Result_tab;
}
```

In questa funzione viene passato l'array con il risultato della Querry: "SELECT \* FROM DIPENDENTI" e assegna ad ogni campo Telefono o Cellulare del record uno sfondo Rosso o Verde.

Alla fine possiamo notare il return di \$Risult\_tab in quanto ci servirà nella creazione della griglia.

- 2) Al momento della creazione della tabella, richiamiamo la funzione appena creata:

```
$sql="SELECT * FROM DIPENDENTI";
```

```

$ita_grid01 = new TableView($this->gridDitte,
    array(
        'sqlDB' => $this->DITTA_DB,
        'sqlQuery' => $sql,
        'rowIndex' => 'Key'));
$ita_grid01->setPageNum(1);
$ita_grid01->setPageRows(20000);
$ita_grid01->setSortIndex('COGNOME');
$ita_grid01->setSortOrder('asc');
// Elabora il risultato
$result_tab = $ita_grid01->getDataArray();
$result_tab = $this->elaboraRecords($result_tab);
$ita_grid01->getDataPageFromArray('json',
$result_tab)

```

E come possiamo notare la griglia viene creata utilizzando l'Array Rielaborato:  
`$ita_grid01->getDataPageFromArray('json', $Result_tab).`

---

Nel caso in cui la SELECT preveda una JOIN con più tabelle e non si ha la possibilità di selezionare un determinato indice da una TABELLA sarà necessario ricorrere ad un Funzione Elabora Record e alla creazione di un ReaderId composto da più Campi.

Questa funzione è necessaria soprattutto con la JOIN di più tabelle, perchè non sempre è possibile selezionare un solo campo:

Immaginiamo di avere 2 tabelle: ANAGRAFICA\_SOGGETTO e ANAGRAFICA\_IMMOBILE e che entrambe racchiudono tanti campi anagrafici importanti, troppi per essere selezionati uno ad uno. La soluzione più semplice sarebbe:

```

SELECT * FROM ANAGRAFICA_SOGGETTO JOIN ANAGRAFICA_IMMOBILE ON
ANAGRAFICA_SOGGETTO.CODICE = ANAGRAFICA_IMMOBILE.CODICESOGGETTO

```

Il risultato della tabella avrebbe quindi in comune anche gli indici delle 2 tabelle.

Come detto in precedenza la griglia prende automaticamente l'indice della tabella se si prendono tutti i campi presenti o comunque si seleziona anch'esso, ma nel caso di una JOIN come questa (in cui si prendono tutti i campi da entrambe) l'indice della griglia verrebbe assegnato automaticamente in ordine crescente, perchè nessuno dei 2 indici ha la priorità sull'altro.

Dovremo, quindi, nell'applicazione, al momento della creazione della grid, rielaborare il risultato della query e passare alla tabella un Array.

Esempio Guidato:

In una tabella abbiamo: TABELLA: OPERAI JOIN PRODUZIONI

<b>CodiceDipendente</b>	<b>Cognome</b>	<b>Nome</b>	<b>Data</b>	<b>Totale Prodotti</b>
001	Rossi	Luigi	22/01/2013	75
004	Verdi	Rosa	25/01/2013	15

Le tabelle senza JOIN :

## TABELLA: PRODUZIONI

CodiceProdotto	TipoProdotto	CodiceDipendente	Data	Numero
BX01	BulloneB	001	22/01/2013	20
BX01	BulloneA	001	25/01/2013	30
BX04	BulloneC	001	22/01/2013	25
BC9F	ChiodoB	004	22/01/2013	15

## TABELLA: OPERAI

CodiceDipendente	Cognome	Nome
001	Rossi	Luigi
004	Verdi	Rosa

Vogliamo che all'evento dbClickRow vengano elencati tutti i prodotti creati di quel operaio a quella data, portandoci dietro la data e codice del dipendente.

Dopo aver aggiunto nella grid del model il **ReaderId** e nell'applicazione il '**rowIndex**' ⇒ '**Key**' (come sopra descritto), nel programma dovremo anche provvedere alla Rielaborazione del risultato della Querry.

Questa rielaborazione può essere fatta sempre attraverso la funzione **ElaboraRecords**.

```
function elaboraRecords($Result_tab) {
    foreach ($Result_tab as $key => $Result_rec) {
        $Result_tab[$key]['Key'] = $Result_tab[$key]['CODICE'] . "-" .
$Result_tab[$key]['DATA'];
    }
    return $Result_tab;
}
```

In questa funzione provvederemo a rielaborare il nostro nuovo Rowid: 'Key' componendolo di "CODICEDIPENDENTE" - "DATA", in modo che al dbClickRow come rowid ci torni un indice come questo: "001-20130122".

Dopo di che, per poter utilizzare questo **Indice Composto**, all'evento dbClickRow potremo utilizzare la funzione: explode:

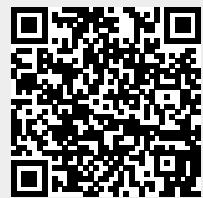
L'explode ci restituirà quindi un array con le 2 chiavi che ci servono, dopo di che possiamo assegnare a 2 variabili il Codice e la Data:

```
switch ($_POST['id']) {
    case $this->nameForm . '_gridOperai':
        $chiavi=explode("-", $_POST['rowid']);
        $Codice=$chiavi['0'];
        $Data=$chiavi['1'];
        $sql=" SELECT * FROM PRODOTTI WHERE CODICEDIPENDENTE
= $Codice AND DATA = '$Data'";
        $ita_grid01 = new TableView($this->gridPresenzeFiere,
            array(
                'sqlDB' => $this->DITTA_DB,
```

```
        'sqlQuery' => $sql));
$ita_grid01->setPageNum(1);
$ita_grid01->setPageRows(20000);
$ita_grid01->setSortIndex($_POST['sidx']);
$ita_grid01->setSortOrder($_POST['sord']);
$ita_grid01->getDataPage('json');
break;
}
break;
```

From:

<https://wiki.nuvolaitalsoft.it/> - **wiki**



Permanent link:

<https://wiki.nuvolaitalsoft.it/doku.php?id=sviluppo:readerid&rev=1358444040>

Last update: **2018/03/19 10:45**