

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 query 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 si vuole inserire un Rowid diverso da quello di Default in una tabella è necessario:

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

Dove 'Key' rappresenta il nome della nostra chiave.

Esempio:

| Nome Attributo | Valore Attributo |
|----------------|--|
| class | {cellEdit:false, readerId:'Key', caption: "Ditte", shrinkToFit:true, width:650 |

- Nell'applicazione, al momento della creazione della tabella, dovremo aggiungere all'array della grid il 'rowIndex'=>'Key', per specificare quale sarà il Rowid che dovrà utilizzare:

```
$sql=$this->CreaSql();  
$ita_grid01 = new TableView($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');  
....
```

Nota. Questa elaborazione è necessaria nel momento in cui la query che passiamo sarà il risultato di più JOIN a più tabelle.

Se invece si vuole creare un ROWID che racchiuda più campi, o semplicemente non riusciamo a farci tornare un determinato rowid, è necessario:

- Nell'applicazione, al momento della creazione della grid, dovremo rielaborare il risultato della query e passare alla tabella un Array.

Esempio:

In una tabella abbiamo: TABELLA: OPERAI JOIN PRODUZIONI

| CodiceDipendente | Cognome | Nome | Data | Totale Prodotti |
|------------------|---------|-------|------------|-----------------|
| 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 al doppioclick su una riga 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 Query.

Questa rielaborazione la possiamo fare tramite una funzione:

```
$sql=$this->CreaSql();
$ita_grid01 = new TableView($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)
```

La funzione **elaboraRecords**. In questa funzione provvederemo a rielaborare il nostro nuovo Rowid: 'Key' componendolo di "CODICEDIPENDENTE" - "DATA", in che quando faremo doppio click sulla riga selezionata il nostro rowid torni ad esempio: "001-20130122".

```
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;
}
```

Al doppio click per ottenere i dati che ci servono dal rowid potremo utilizzare la funzione: explode: L'explode ci restituirà quindi un array con le 2 chiavi che ci servono:

```
switch ($_POST['id']) {
    case $this->nameForm . '_grid0perai':
        $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=1358436865>

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

