MYSQL IN CLAUSE

You can use **IN** clause to replace many **OR** conditions

To understand **IN** clause, consider an **employee_tbl** table, which is having the following records:

```
mysql> SELECT * FROM employee_tbl;
+------+------+------------+--------------------+
| id   | name | work_date  | daily_typing_pages |
+------+------+------------+--------------------+
|    1 | John | 2007-01-24 |                250 |
|    2 | Ram  | 2007-05-27 |                220 |
|    3 | Jack | 2007-05-06 |                170 |
|    3 | Jack | 2007-04-06 |                100 |
|    4 | Jill | 2007-04-06 |                220 |
|    5 | Zara | 2007-06-06 |                300 |
|    5 | Zara | 2007-02-06 |                350 |
+------+------+------------+--------------------+
7 rows in set (0.00 sec)
```

Now, suppose based on the above table you want to display records with daily_typing_pages equal to 250 and 220 and 170. This can be done using **OR** conditions as follows

```
mysql>SELECT * FROM employee_tbl
    ->WHERE daily_typing_pages= 250 OR
    ->daily_typing_pages= 220 OR  daily_typing_pages= 170;
+------+------+------------+--------------------+
| id   | name | work_date  | daily_typing_pages |
+------+------+------------+--------------------+
|    1 | John | 2007-01-24 |                250 |
|    2 | Ram  | 2007-05-27 |                220 |
|    3 | Jack | 2007-05-06 |                170 |
|    4 | Jill | 2007-04-06 |                220 |
+------+------+------------+--------------------+
4 rows in set (0.02 sec)
```

Same can be achieved using **IN** clause as follows:

```
mysql> SELECT * FROM employee_tbl
    -> WHERE daily_typing_pages IN ( 250, 220, 170 );
+------+------+------------+--------------------+
| id   | name | work_date  | daily_typing_pages |
+------+------+------------+--------------------+
|    1 | John | 2007-01-24 |                250 |
|    2 | Ram  | 2007-05-27 |                220 |
|    3 | Jack | 2007-05-06 |                170 |
|    4 | Jill | 2007-04-06 |                220 |
+------+------+------------+--------------------+
4 rows in set (0.02 sec)
```