#1093 - You can't specify target table 'XXX' for update in FROM claus

If you do this:

```
DELETE FROM story_category
WHERE category_id NOT IN (
        SELECT DISTINCT category.id AS cid FROM category 
        INNER JOIN story_category ON category_id=category.id
)
```

you a going to get an error.

But if you wrap the condition in one more select

```
DELETE FROM story_category
WHERE category_id NOT IN (
    SELECT cid FROM (
        SELECT DISTINCT category.id AS cid FROM category 
        INNER JOIN story_category ON category_id=category.id
    ) AS c
)
```

it would do the right thing!!

The problem is that MySQL, for whatever inane reason, doesn't allow you to write queries like this:

```
UPDATE myTable
SET myTable.A =
(
    SELECT B
    FROM myTable
    INNER JOIN ...
)
```

That is, if you're doing an `UPDATE`/`INSERT`/`DELETE` on a table, you can't reference that table in an inner query *(you **can** however reference a field from that outer table...)*

The solution is to replace the instance of `myTable` in the sub-query with `(SELECT * FROM myTable)`, like this

```
UPDATE myTable
SET myTable.A =
(
    SELECT B
    FROM (SELECT * FROM myTable) AS something
    INNER JOIN ...
)
```

This apparently causes the necessary fields to be implicitly copied into a temporary table, so it's allowed.

I found this solution here. A note from that article:

> You don’t want to just `SELECT * FROM table` in the subquery in real life; I just wanted to keep the examples simple. In reality, you should only be selecting the columns you need in that innermost query, and adding a good `WHERE` clause to limit the results, too.