Duplicates and Sub Selects

Posted At : November 16, 2006 1:45 PM | Posted By : Mark Kruger
Related Categories: SQL tips

Here's a sticky problem. How do you build a query that gives you distinct records from one table based on multiple records from another table and order by a date found in the second table. In my real world example, I have project tracks in "Items" and comments or notes in "events". I want a distinct list of "items" that have been updated in the past 10 days. That seems easy right? Well... not as easy as it seems on the surface.

The Problem

Let's say in a project has been "commented on" ("events" table) several times during the 10 days. If I do something like:

```
<cfquery ....>
        SELECT DISTINCT I.itemid, i.title
        FROM items i join events e
        ON i.itemid = e.itemid
        ORDER by e.datecreated
    </cfquery>
```

The DB will complain that I cannot order by a column that is not in the SELECT list when using "DISTINCT". If I put the column in the select list I will get duplicates.

Real World Example

Let's a look at the problem a bit closer. *Items* contains the master record for a particular track. For example, Items are related to individual bugs, feature requests, work orders, or change requests. For example, the Items table might look like:

```
ItemId Description

17 Wash Your Wife

22 Take out the dry cleaning

4 Pick up the dog
```

Events contains comments or updates mapped to the itemid. So events might look lik this:

```
ItemId event    Date

17 She Didn't like it   11-15-2006

17 Are you sure about this  11-12-2006

22 My Suite didn't like desert  11-14-2006

4 I should have bought a smaller dog 11-11-2006
```

If we order by date we will end up with:

```
ItemId event    Date

17 She Didn't like it   11-15-2006

22 My Suite didn't like desert  11-14-2006

17 Are you sure about this  11-12-2006

4 I should have bought a smaller dog 11-11-2006
```

...so any "select distinct" join that includes the date will end up showing

```
ItemId Description   date

17 Wash Your Wife   11-15-2006

22 Take out the dry cleaning  11-14-2006

17 Wash Your Wife   11-12-2006

4 Pick up the dog   11-11-2006
```

What I want is to remove that third line. I already "know" that itemid 17 has been updated recently. I don't need to "know" that it was updated previously - even if it is inside my 10 day window. Sure, I can simply keep track of the itemIds in my loop and not display the ones that have already been handled. But surely there is another way? Here's the solution:

```
<Cfquery ...>
SELECT     v.itemid, v.title, e.datecreated, e.eventid
FROM     items v
    INNER JOIN Events e
        ON v.itemid = e.itemid
WHERE     e.eventid in (
    SELECT     top 1 e2.eventid
    FROM     events e2
    WHERE     e2.itemid = v.itemid
    ORDER BY e2.datecreated desc
    )
ORDER BY e.datecreated desc
</CFQUERY>
```

What's the secret? The subsect filters out all but the latest eventid - and that effects the date that is pulled in from the eventID table. This has the effect of removing that second level of items.

NOTE: Thanks to my good friend and SQL guru Mike Klostermeyer for this tip.