

Re: mysql update/replace syntax

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2004-08/2390.html>

From: Gordon Burditt (gordonb.pilrf_at_burditt.org)

Date: 08/30/04

Date: 30 Aug 2004 04:40:03 GMT

>> > *You are approaching this from the wrong angle. You should be quering the customer sales records and producing summary information from that.*

This is a hospital kitchen. Recording Personally Identifiable Information about a patient in a database makes that database legally unusable for the purpose intended for it (what kind of food they should order and how much they should keep on hand), and that's why your predecessor is in jail now: violating HIPAA requirements.

>> > *INSERT INTO sales (customer, product, qty) VALUES (\$customer, \$fruit, \$qty)*

>> > *SELECT fruit, sum(qty) AS numsales FROM sales GROUP BY fruit ORDER BY numsales*

>> > <http://dev.mysql.com/doc/mysql/en/GROUP-BY-Functions.html>

>> > *No, I am wanting to *record* what visitor does. If they buy 5 apples, I want MySQL to find the row with "apple" in it, and increment it by 5. If there is *no* row with "apple", then create a row and enter a "5" there.*

It is possible to insert or update a row with one query:

```
INSERT INTO sales_summary SET product = 'apple', qty = 3
ON DUPLICATE KEY UPDATE qty = qty + 3;
```

This requires that the product column has a unique key on it.

It has the advantage that it's atomic: you don't have to do explicit locking but you can't get fouled up by different ordering of requests. (The problem with this kind of query is that you need a minimum version of MySQL of about 4.1 (not sure exactly which version), and I don't know that any other database accepts this syntax.).

You might be able to add a "date" column to get daily totals, but

only if you can convince management that asking the judge for permission is worth the risk of having him prohibit the existence of the database entirely.

I use this sort of thing a lot with email white/black lists. You want to record, say, the sender, the number of emails from this sender to good addresses (but NOT what the good addresses are), the number of emails from this sender to bad addresses (but NOT what the bad addresses are), and the time of the latest email from that sender. If there's no entry, add one. If there is an entry, increment one of the counters. There is a high probability of simultaneous SPAMs from the same sender arriving at the same time. You DO NOT want to record each email: this allows spammers to conduct a denial-of-service attack against you by running your database out of disk space.

Another process can later classify the sender as one to be blocked or not, in part based on the assumption that a sender who sends a lot of mail to mostly invalid addresses is a spammer. Other fields can store manual settings.

*>Yes you are. The above will tell you exactly how many apples or bananas have
>been sold without resorting to increamenting a seperate table/record.*

Yes, but you use a lot more storage, and the personally identifiable information in it means you're not allowed to do any SELECTs at all, and has a high probability of getting the whole project cancelled.

*>Further more, with a little creativity, you can find out things like, on
>what day which fruit sells best. How much of each fruit do you sell each
>month? Is there a peek period for selling oranges?*

Keeping this kind of marketing information around can kill your business if it gets out that you're keeping it, say, because someone managed to steal it.

Gordon L. Burditt