Bazaar – php example code – part 14 – handling sold item in database

Article focus on way how to mark sold item in the bazaar_item table for further deletion. Successfully sold item is marked with -1 in cart_number, that is displayed in listing on admin page of bazaar. Admin can see it and delete them. Separate article will focus on way how to order items in lists for better manipulation.

Requirements for item handling

During lifecycle of item we must to manage marking different states of item. When user publish item for sell, item is marked by cart_number field in a database table bazaar_item as 0 and published gain default value of 0 (waiting for admin to make item visible in listening's).

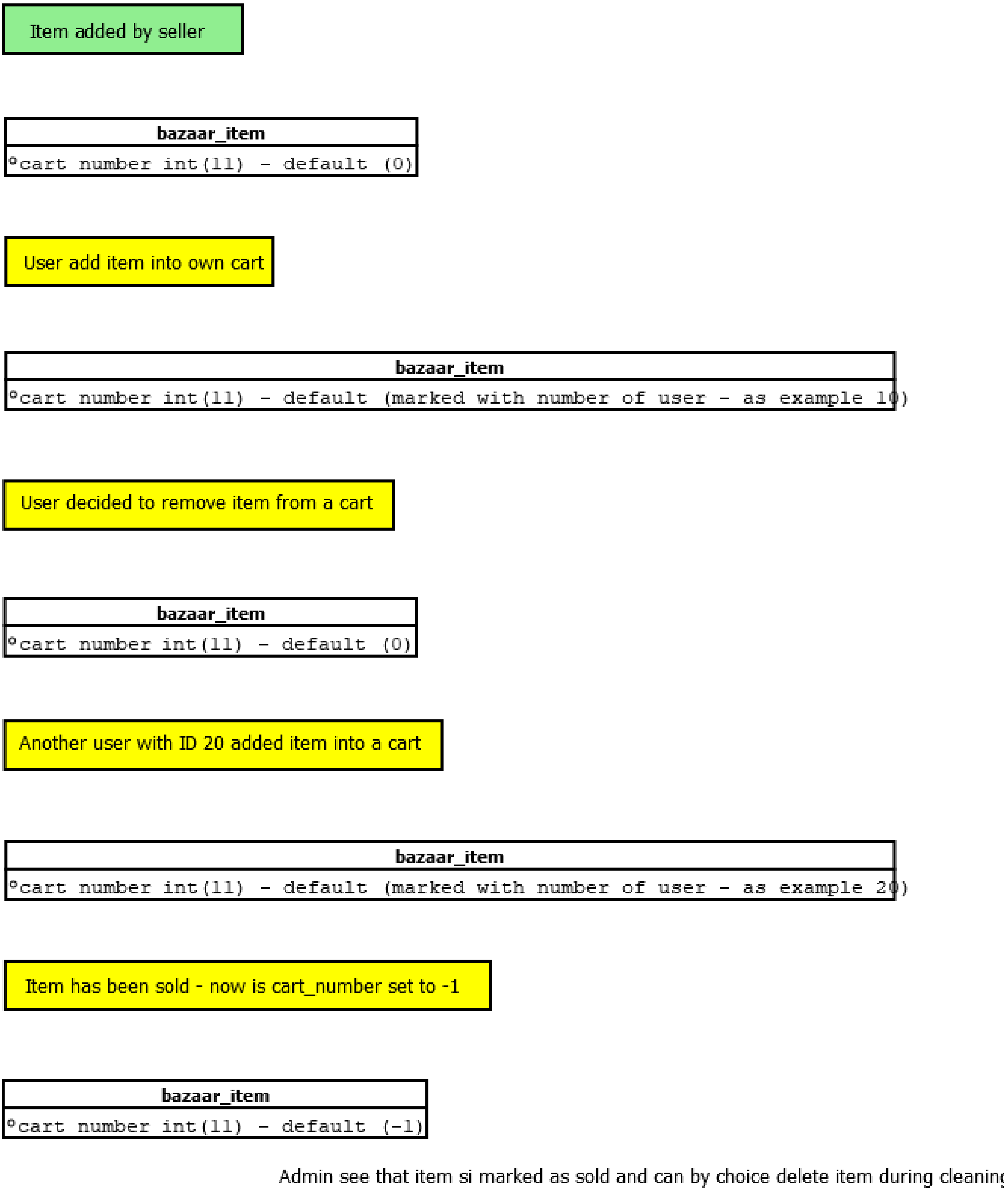
After admin approval, visible field change its value from 0 to 1. Item now waiting for adding into a cart of buyer. After selecting and adding item into a cart cart_number is changed in to a number representing ID of user that make buy (added item into their own cart).

If usere remove item from cart, cart_number value returns to 0. Another user can add item into a cart and mark them with own ID number.

If user commit cart for final buy, all successfully sold items are marked by -1 in cart_number field.

On admin page, sold item marked with -1 are displayed and after admins decision can be finally removed from a database. Optional functionality for further implementation is option for archiving of soled item by site admin. Another way for automation of removing unused and sold item is creating automated script associated with button for removing any sold item from database table before a some date (as example one month after sold).

Full process of altering of cart_number field is shown on next picture.



Item added by seller

bazaar_item

°cart number int(11) - default (0)

User add item into own cart

bazaar_item

°cart number int(11) - default (marked with number of user - as example 10)

User decided to remove item from a cart

bazaar_item

°cart number int(11) - default (0)

Another user with ID 20 added item into a cart

bazaar_item

°cart number int(11) - default (marked with number of user - as example 20)

Item has been sold - now is cart_number set to -1

bazaar_item

°cart number int(11) - default (-1)

Admin see that item si marked as sold and can by choice delete item during cleaning

Process of handling cart_number filed during lifetime of item
In further rows you can see how our page codes handle this changes during whole lifetime of item for sell.

1. sellitem.php – item added for listening

Because default values for fields published and cart_number set to 0. There is no need for any specific write values into a database table. As you can see in next sql statement:

```
$sql = „INSERT INTO bazaar_item (name_of_item, price_eur, subc
ategory_id, users_id, item_add_date, screenshot1, screenshot2,
 screenshot3, item_description)
                              VALUES (‚$name_of_item‘, $price_eu
r , ‚$subcategory_id‘ , ‚$users_id‘ , now(), ‚$screenshot1‘, ‚
$screenshot2‘, ‚$screenshot3‘, ‚$item_description‘ )“;
```

2. cart.php – mark item as sold after successful commit to buy

Cart.php script is responsible for marking cart_number with -1 after successful buy. This is how it is implemented in our code

```
 /************************************************************
*
             *   obtain data buyed items from this buyer with
users_id defined by current SESSION
             */
            //get info about sold items – we must go through a
ll buyed items and send emaily one by one for all diferent sel
ers of item (first approach for all item one)
            $sql = „SELECT * FROM bazaar_item WHERE cart_numbe
r = „.“,$users_id'“  ;
            if($output = mysqli_query($dbc, $sql)){
                if(mysqli_num_rows($output) > 0){  // if any r
ecord obtained from SELECT query

                    while($row = mysqli_fetch_array($output)){
 //next rows outputed in while loop
```

```
                           $item_id = $row[‚item_id‘];
                            $name_of_item = $row[‚name_of_item
‘];
                           $price_eur = $row[‚price_eur‘];
                            $users_id_of_seller = $row[‚users_
id‘];
                             /********************************
************************
                             *   if item with item_ide was bou
ght, tgen set cart_number to -1 and published to -1 – mean sol
d
                            */
                             // update cart_number and publish
ed to -1 sold
                            $sql_update = „UPDATE bazaar_item
SET cart_number = ‚-1‚, published = ‚-1‚ WHERE item_id = $item
_id LIMIT 1″;
                           // execute SQL
                           mysqli_query($dbc, $sql_update);
```

3. buyitem.php – mark added item into a cart by users ID

When a user add item into a cart, buyitem.php script mark cart_number field with ID of that user. This script work as you now in a two ways. After first call from index.php obtaing get data about item of interest and show info about product of interest, user can chose if will add item into a cart and post submit result on themself. Only post handling part (not get handling part) of code will make operation that is shown in next rows:

```
             // conect to the database
              $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PW, D
B_NAME);
              // get user id from selected session
              $users_id = $_SESSION[‚users_id‘];
              //create sql query along selected operation

                    $sql = „UPDATE bazaar_item SET cart_number
 =“.“‚$users_id‚“.“ WHERE item_id = $item_id LIMIT 1″;
```

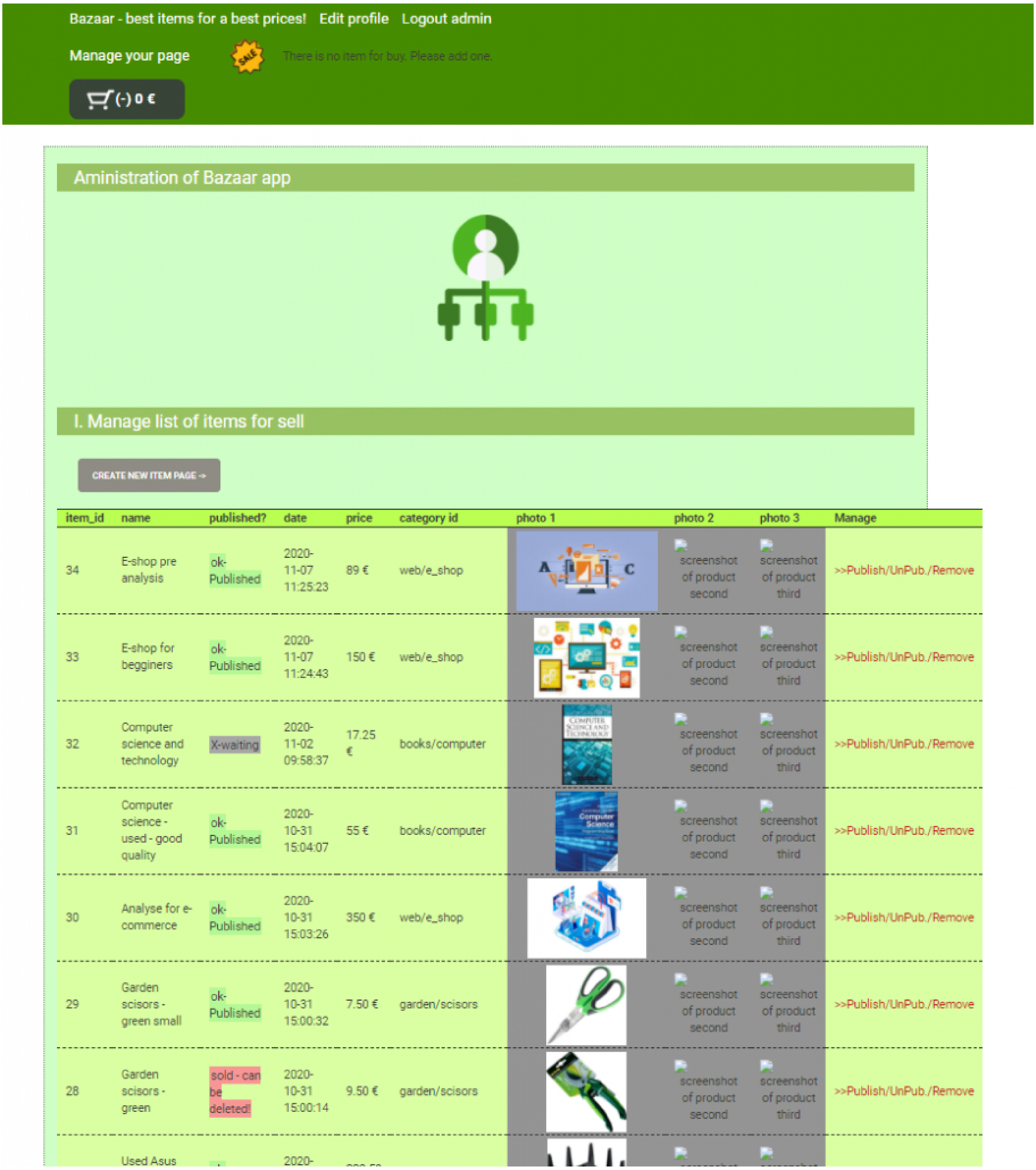
```
                    // execute SQL
                    mysqli_query($dbc, $sql);
                    // confirm executed command
                    echo ‚<p> The item <strong>‘ . $name_of_it
em . ‚</strong> with id <strong>‘ . $item_id . ‚</strong> was
succesfully added into a cart. </p>‘;

              // close database connection
              mysqli_close($dbc);
```

4. admin.php – show status of item and give option for deletion

Admin page show items in a database and allow ordering (it will be explained in one of the next articles, how we obtained these functionality).

Next picture show how it looks like without ordering functionality.



Bazaar - best items for a best prices! Edit profile Logout admin

Manage your page

There is no item for buy. Please add one.

Aministration of Bazaar app

I. Manage list of items for sell

item_id	name	published?	date	price	category id	photo 1	photo 2	photo 3	Manage
34	E-shop pre analysis	ok-Published	2020-11-07 11:25:23	89 €	web/e_shop		screenshot of product second	screenshot of product third	>>Publish/UnPub./Remove
33	E-shop for begginers	ok-Published	2020-11-07 11:24:43	150 €	web/e_shop		screenshot of product second	screenshot of product third	>>Publish/UnPub./Remove
32	Computer science and technology	X-waiting	2020-11-02 09:58:37	17.25 €	books/computer		screenshot of product second	screenshot of product third	>>Publish/UnPub./Remove
31	Computer science - used - good quality	ok-Published	2020-10-31 15:04:07	55 €	books/computer		screenshot of product second	screenshot of product third	>>Publish/UnPub./Remove
30	Analyse for e-commerce	ok-Published	2020-10-31 15:03:26	350 €	web/e_shop		screenshot of product second	screenshot of product third	>>Publish/UnPub./Remove
29	Garden scisors - green small	ok-Published	2020-10-31 15:00:32	7.50 €	garden/scisors		screenshot of product second	screenshot of product third	>>Publish/UnPub./Remove
28	Garden scisors - green	sold - can be deleted!	2020-10-31 15:00:14	9.50 €	garden/scisors		screenshot of product second	screenshot of product third	>>Publish/UnPub./Remove
	Used Asus		2020-						

admin.php – sold item marked in published field

Next code snippet show, how are information depicted in published filed in our table:

```
 while($row = mysqli_fetch_array($output)){ //next rows output
ed in while loop
                echo “ <div class=\“mailinglist\“> “ ;
                echo „<tr>“;
                    echo „<td>“ . $row[‚item_id‘] . „</td>“;
                    echo „<td>“ . $row[‚name_of_item‘] . „</td
```

```
>“;
                    if ($row[‚published‘]==1) { // show if published – set 1 or waiting set to 0
                          echo ‚<td><span class=“green“> ok-Published </spann></td>‘;
                    } else if ($row[‚published‘]== 0){ // 1 published, 0 unpublished, -1 sold ready for deletion
                           echo ‚<td><span class=“gray“> X-waiting </spann></td>‘;
                    } else if ($row[‚published‘]== -1){ // 1 published, 0 unpublished, -1 sold ready for deletion
                        echo ‚<td><span class=“red“> sold – can be deleted! </spann></td>‘;
                    }
```

… next part is omited

Adin can now see all itms marked as sold and can manually delete them. For this operation is requested removeitem.php script by clicking link generating link in row manage from upper picture (on admin page). This way is provided call with GET data.

Our removeitem.php script work also as it was mentioned before for buyitem.php in two ways. GET data from calling ling and next submit user decision for removing item from database. For deletion user must change radiobutton from no to yes and submit deletion commit.

Main part enable publishing, unpublishing and deletion of our item. Next code snippet show main logic of this script.

… omitted part of removeitem.php …

```
 //create sql query along selected operation
              switch ($operation) {
                case „publish“:
                    $sql = „UPDATE bazaar_item SET published = ‚1‘ WHERE item_id = $item_id LIMIT 1“;
                    // execute SQL
                    mysqli_query($dbc, $sql);
                    // confirm executed command
                    echo ‚<p> The item <strong>‘ . $name_of_it
```

```
em . ‚</strong> with id <strong>‘ . $item_id . ‚</strong> was
succesfully published. </p>‘;
                    break;
                case „unpublish“:
                    $sql = „UPDATE bazaar_item SET published =
 ‚0‘ WHERE item_id = $item_id LIMIT 1“;
                    // execute SQL
                    mysqli_query($dbc, $sql);
                    // confirm executed command
                    echo ‚<p> The item <strong>‘ . $name_of_it
em . ‚</strong> with id <strong>‘ . $item_id . ‚</strong> was
succesfully unpublished. </p>‘;
                    break;
                case „delete“:
                    $sql = „DELETE FROM bazaar_item WHERE item
_id = $item_id LIMIT 1“;
                    // execute SQL
                    mysqli_query($dbc, $sql);
                    // confirm executed command
                    echo ‚<p> The item <strong>‘ . $name_of_it
em . ‚</strong> with id <strong>‘ . $item_id . ‚</strong> was
succesfully deleted from listening on bazaar. </p>‘;
                    @unlink(IMAGE_PATH . $screenshot1); //dele
te image file
                    @unlink(IMAGE_PATH . $screenshot2);
                    @unlink(IMAGE_PATH . $screenshot3);
                    break;
            }
… omitted part …
```

Conclusion

We have now a better view on to how is lifecycle of item managed from moment of publishing for sell up to time of removing item from a database.

Now we can see further ways for improvement mostly in last phases of item lifecycle. Our first approach rely on manual handling by site admin, we can add much more automation into a this process by adding code for batch deletion of all sold items or only items sold before a specific time. Other thoughts rely on way how to run script in regulary base without need for intervention. Or how to archive all or only a specific items.

Full application code can be obtained from github from there.