

Other - Feature #1867

tool to automate steps involved in adding affiliate programs

04/20/2024 09:56 AM - Kevin Zaleski

Status:	Assigned	Start date:	04/20/2024
Priority:	Normal	Due date:	
Assignee:	Alexander Zaleski	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

There are several steps to processing a new affiliate program that we have joined. The amount of time a human needs to spend could be reduced greatly by creating a tool that automates creation of the necessary data.

1. Create a database of subscribed affiliate programs.

After we create a database, <https://pickleballdiscountcodes.com/> and <https://pickleballdiscountcodes.com/aff.html> can be turned into dynamically-generated pages and humans will be spared from manually updating those pages every time a program is added.

TODO: DEFINE SCHEMA for affPrograms

- id int(10)
- affProgram varchar(128)
- affUrl varchar(255)
- refUrl varchar(255)
- PPCdoorway varchar(255)
- commission varchar(128)
- paidVia varchar(128)
- notes text utf8_general_ci

Above is needed for <https://pickleballdiscountcodes.com/aff.html>, below is needed for <https://pickleballdiscountcodes.com/> (disc, discText, and discCode are used to construct the "Discount Details" column in <https://pickleballdiscountcodes.com/>

- disc varchar(128)
- discText varchar(128)
- discCode varchar(128)
- discInstructions varchar(128)

I prompted <https://chat.openai.com/?model=gpt-4> with ...

give me SQL to create a table affPrograms and add 26 blank records using the following schema:

```
- id int(10) auto_increment
- affProgram varchar(128)
- affUrl varchar(255)
- refUrl varchar(255)
- PPCdoorway varchar(255)
- commission varchar(128)
- paidVia varchar(128)
- notes text utf8_general_ci
- disc varchar(128)
- discText varchar(128)
- discCode varchar(128)
- discInstructions varchar(128)
```

... and got ...

Below is the SQL code to create a table named affPrograms with the specified schema and to insert 26 blank records into it. This SQL assumes you are using a MySQL database given the datatype specifics like int(10) and varchar sizes, as well as the collation setting for the text field.

SQL to Create the Table

```
CREATE TABLE affPrograms (
```

```

id INT(10) AUTO_INCREMENT,
affProgram VARCHAR(128),
affUrl VARCHAR(255),
refUrl VARCHAR(255),
PPCdoorway VARCHAR(255),
commission VARCHAR(128),
paidVia VARCHAR(128),
notes TEXT CHARACTER SET utf8 COLLATE utf8_general_ci,
disc VARCHAR(128),
discText VARCHAR(128),
discCode VARCHAR(128),
discInstructions VARCHAR(128),
PRIMARY KEY (id)
);

```

SQL to Insert 26 Blank Records

Given that the id column is an auto-increment field, you do not need to specify the ID values when inserting new records. MySQL will automatically assign incrementing integers starting from 1. To insert blank records (with NULL values in all other fields), you can use the following:

```

INSERT INTO affPrograms
(affProgram, affUrl, refUrl, PPCdoorway, commission, paidVia, notes, disc, discText, discCode, discInstructions)
VALUES
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL),
(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);

```

Notes:

Primary Key: The id column is the primary key and is set to auto-increment.

Character Set and Collation: The notes column is set to use UTF-8 character encoding.

Inserting Blank Records: 26 insertions are batched into one command to streamline the operation. Since the id field is auto-increment, there's no need to specify it during insertion.

This setup ensures that you can add records to the table easily, and MySQL will manage the incremental IDs for each new entry.

2. Populate the database from our existing set of subscribed affiliate programs.

We have 26 programs to populate the dataabse, it is not clear if it will take less time to copy/paste manually or use AI.

If it needs to be done manually, just setup one record (s you can continue on with other parts of this task) and I will have someone else populate it.

If you want to try to use AI+python (perhaps more fun, probably more time) ...

I saved <https://pickleballdiscountcodes.com/aff.html> into Affiliate Programs.html

I prompted <https://chat.openai.com/?model=gpt-4> with ...

give me code to parse the values from the first column in the table in the attached table, list those values, and generate SQL to update the records, in order, in the affProgram field of the affPrograms table with those values

AI gave me: To extract the values from the first column of the table in the provided HTML file and generate SQL statements for updating records in the affProgram field of the affPrograms table, we can use the following approach:

Parse the HTML file using BeautifulSoup.

Extract the values from the first column of the table.

Generate SQL update statements to update the corresponding records.

Here's how you can do it in Python:

```
from bs4 import BeautifulSoup

# Load the HTML file content
with open('/mnt/data/Affiliate Programs.html', 'r') as file:
    html_content = file.read()

# Parse the HTML
soup = BeautifulSoup(html_content, 'html.parser')

# Find the table
table = soup.find('table')

# List to store the first column values
affiliate_programs = []

# Extract data from the first column, skipping the header row
for row in table.find_all('tr')[1:]: # The first row is the header
    first_column = row.find('td')
    if first_column:
        affiliate_programs.append(first_column.text.strip())

# Output the extracted values
print("Affiliate Programs:")
print(affiliate_programs)

# Generating SQL statements
print("\nSQL Update Statements:")
for program in affiliate_programs:
    sql = f"UPDATE affPrograms SET affProgram = '{program}' WHERE id = ?;"
    print(sql)
```

Tasks below are all optional, whichever ones you can do will help.

HTML to generate those two pages dynamically

AI can do it if you craft your prompt well.

Create HTML that will accept input fields and create the SQL to add a record to affPrograms.

give me HTML code that will accept the necessary input fields and display the SQL to add a record to the affPrograms table

Below is an example of a simple HTML form with JavaScript that captures input from the user for each field defined in your affPrograms table. The form will not actually insert data into a database; instead, it generates an SQL insert statement based on the

input and displays it on the same page.

HTML and JavaScript Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Add Record to affPrograms</title>
</head>
<body>
  <h1>Add a New Record to affPrograms</h1>
  <form id="dataForm">
    <label for="affProgram">Affiliate Program:</label>
    <input type="text" id="affProgram" name="affProgram"><br><br>

    <label for="affUrl">Affiliate URL:</label>
    <input type="text" id="affUrl" name="affUrl"><br><br>

    <label for="refUrl">Referral URL:</label>
    <input type="text" id="refUrl" name="refUrl"><br><br>

    <label for="PPCdoorway">PPC Doorway:</label>
    <input type="text" id="PPCdoorway" name="PPCdoorway"><br><br>

    <label for="commission">Commission:</label>
    <input type="text" id="commission" name="commission"><br><br>

    <label for="paidVia">Paid Via:</label>
    <input type="text" id="paidVia" name="paidVia"><br><br>

    <label for="notes">Notes:</label>
    <textarea id="notes" name="notes"></textarea><br><br>

    <label for="disc">Discount:</label>
    <input type="text" id="disc" name="disc"><br><br>

    <label for="discText">Discount Text:</label>
    <input type="text" id="discText" name="discText"><br><br>

    <label for="discCode">Discount Code:</label>
    <input type="text" id="discCode" name="discCode"><br><br>

    <label for="discInstructions">Discount Instructions:</label>
    <input type="text" id="discInstructions" name="discInstructions"><br><br>

    <button type="button" onclick="generateSQL()">Generate SQL</button>
  </form>

  <h2>Generated SQL Statement:</h2>
  <p id="sqlOutput"></p>

  <script>
    function generateSQL() {
      const form = document.forms['dataForm'];
      const affProgram = form['affProgram'].value;
      const affUrl = form['affUrl'].value;
      const refUrl = form['refUrl'].value;
      const PPCdoorway = form['PPCdoorway'].value;
      const commission = form['commission'].value;
      const paidVia = form['paidVia'].value;
      const notes = form['notes'].value;
      const disc = form['disc'].value;
      const discText = form['discText'].value;
      const discCode = form['discCode'].value;
      const discInstructions = form['discInstructions'].value;
```

```
        const sql = `
INSERT INTO affPrograms (affProgram, affUrl, refUrl, PPCdoorway, commission, paidVia, notes, disc,
discText, discCode, discInstructions) VALUES ('${affProgram}', '${affUrl}', '${refUrl}', '
${PPCdoorway}', '${commission}', '${paidVia}', '${notes}', '${disc}', '${discText}', '${discCode}'
, '${discInstructions}');`;
        document.getElementById('sqlOutput').innerText = sql;
    }
</script>
</body>
</html>
```

Explanation:

HTML Form: This form contains text inputs and a textarea for the fields of the affPrograms table. The form does not actually submit data to a server but instead triggers a JavaScript function when the user clicks the submit button.

JavaScript Function: generateSQL() is called when the form is submitted. It retrieves values from the form, constructs an SQL insert statement, and displays it in the pre element with the ID sqlOutput.

Preventing Form Submission: The onsubmit attribute in the form tag calls event.preventDefault() to prevent the form from submitting traditionally, which would reload the page.

This setup allows you to collect data via the HTML form and see the SQL statement generated live, which can then be used to manually insert data into a database. However, be cautious with directly using user input in SQL statements to avoid SQL injection vulnerabilities. In a production environment, you'd want to handle this securely on the server side, possibly with prepared statements or parameterized queries.

Setup Google PPC

ref: <https://support.google.com/google-ads/answer/10702525?hl=en#zippy=%2Cads%2Ckeywords>

Get AI to populate the template Create Responsive Search Ad(s).xlsx for import.

We will populate the template by replacing parameters with input values.

Headlines

Headline 1	10% Discount: ANIVIA10
Headline 1 position	-
Headline 2	10% Discount Code: ANIVIA10
Headline 2 position	-
Headline 3	10% Off Using Code ANIVIA10
Headline 3 position	-
Headline 4	10% Off - Aniviia Pickleball
Headline 4 position	-
Headline 5	10% Off, Aniviia Pickleball
Headline 5 position	-
Headline 6	Aniviia 10% Off Pickleball
Headline 6 position	-
Headline 7	Aniviia Pickleball - 10% Off
Headline 7 position	-
Headline 8	10% Off Aniviia with ANIVIA10
Headline 8 position	-
Headline 9	Use code ANIVIA10 for 10% off

If you decide to go ahead with this I will provide details.

Get AI to populate the template Add Keyword(s).xlsx for import.

I use <https://classygroundcovers.com/cgi-bin/cg/misspellings.pl> to generate keywords, but it might be simpler to have AI recreate the functionality of that tool.

Rules

Square braces: required element(s) "[a|b]"

Curly braces: optional element(s) "{c}"

Pipes inside braces are used to separate alternatives "{c|d|ef}"

Pipes outside braces are not allowed

Braces should not be nested

Characters can be escaped with a backslash "\", like: "[\{}]"

A tilde "~" will be converted into space. (Currently there is no escaping for it)

For example (replace BRAND with the brand name) {best} BRAND pickleball {coupon|discount|savings} {code|deal|promo}
generated 30 good keyword strings:

Displaying 32 of 32 keyword strings

Input was: {best} BRAND pickleball {coupon|discount|savings} {code|deal|promo}

Click on the box below to copy the list to the clipboard

```
BRAND pickleball
BRAND pickleball code
BRAND pickleball coupon
BRAND pickleball coupon code
BRAND pickleball coupon deal
BRAND pickleball coupon promo
BRAND pickleball deal
BRAND pickleball discount
BRAND pickleball discount code
BRAND pickleball discount deal
BRAND pickleball discount promo
BRAND pickleball promo
BRAND pickleball savings
BRAND pickleball savings code
BRAND pickleball savings deal
BRAND pickleball savings promo
best BRAND pickleball
best BRAND pickleball code
best BRAND pickleball coupon
best BRAND pickleball coupon code
best BRAND pickleball coupon deal
best BRAND pickleball coupon promo
best BRAND pickleball deal
best BRAND pickleball discount
best BRAND pickleball discount code
best BRAND pickleball discount deal
best BRAND pickleball discount promo
best BRAND pickleball promo
best BRAND pickleball savings
```

I also do brand models...

ProDrive [encounter|drive|Select|raw|carbon|bella|glasses|bag|backpack] {discount|sale|promo|coupon|deal}

History

#1 - 04/20/2024 10:08 AM - Kevin Zaleski

- Description updated

#2 - 04/20/2024 10:12 AM - Kevin Zaleski

- Description updated

#3 - 04/20/2024 11:12 AM - Kevin Zaleski

- File Screenshot_48.jpg added

- File Screenshot_47.jpg added

- File Add Keyword(s).xlsx added

- File Create Responsive Search Ad(s).xlsx added

#4 - 04/20/2024 05:28 PM - Kevin Zaleski

I created the affPrograms table with another field categoryId for future use (grouping the programs by category).

give me MySQL to create a table affCategories with fields:

id INT AUTO_INCREMENT,
cat VARCHAR,
name VARCHAR,
displayOrder INT,

The tables are setup at <https://liriope.net:2083/cpsess7417920226/3rdparty/phpMyAdmin/index.php>

#5 - 04/22/2024 04:22 PM - Kevin Zaleski

Need to add a 'type' field to identify the codes which are ours vs. someone else's.
To be used later to list programs that are candidates to re-apply to.

Files

Screenshot_47.jpg	87.8 KB	04/20/2024	Kevin Zaleski
Screenshot_48.jpg	235 KB	04/20/2024	Kevin Zaleski
Add Keyword(s).xlsx	14.4 KB	04/20/2024	Kevin Zaleski
Create Responsive Search Ad(s).xlsx	12.2 KB	04/20/2024	Kevin Zaleski