



# TINY CODERS (WEB SYSTEMS)

## MEMBERS:

- AMISTOSO, CLARIES ANN B.
- BATA, MARIZ S.
- FETIL, JAENNIE CYRELL N.
- RESENTES, SHANLEY R.
- SABLAYAN, PENELOPE R.

## 1. DISPLAY ALL ITEMS:

The screenshot shows the phpMyAdmin interface with the 'items' table selected. The table contains 10 rows of data, including item\_id, item\_name, item\_desc, and item\_price. The SQL query 'SELECT \* FROM items;' is entered in the query box.

item_id	item_name	item_desc	item_price
1	caramel macchiato	freshly steamed milk with vanilla-flavoured syrup...	99.00
2	latte	a milk coffee that boast a silky layer of foam as ...	88.75
3	espresso	strong black coffee that's created when hot water ...	50.00
4	cappuccino	a perfect balance of espresso steamed milk and foa...	95.00
5	iced classic chocolate	milk and mocha sauce topped with sweetened and whi...	105.00
6	iced caffe mocha	espresso combined with bittersweet mocha sauce and...	105.75
7	french butter croissant	a beautiful golden color, crisp, moist, and butter...	125.00
8	dark chocolate macadamia cookie	a decadent combination of belgian chocolate chunks...	129.00
9	bacon belgian waffle	yeast-raise dough with sugar pearls and loaded wit...	99.00
10	classic lasagna	this classic lasagna is the perfect italian comfor...	159.00

Here, we inserted the item name, description as well as the item price and we display the information using the SQL “SELECT \* FROM items.

## 2. DISPLAY ALL THE USERS:

The screenshot shows the phpMyAdmin interface with the 'user' table selected. The table contains 4 rows of data, including ua\_id, username, and password. The SQL query 'SELECT \* FROM user;' is entered in the query box.

ua_id	username	password
1	JB	jbb
2	Claries	Claries123
3	JM	jimm
4	Ana	aan

Same with displaying the items we also used “SELECT” to display all the users in our database.



3. USING WHERE CLAUSE

Server: 127.0.0.1 » Database: sample » Table: customer\_profile

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

SELECT \* FROM customer\_profile WHERE gender = 'M';

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

cus\_id

name

gender

ua\_id

1 JB M 1

3 JM M 3

Check all

With selected:

Edit

Copy

Delete

Export

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Query results operations

Print

Copy to clipboard

Export

Display chart

Create view

Here, we view the customer\_profile table and run a SQL query to display all customers whose gender is “M”. The results show two rows, meaning there are two customers in the table with a gender of “M”.

4. DISPLAY ALL ORDERS OF CUS\_ID 1

Server: 127.0.0.1 » Database: sample » Table: orders

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)

SELECT \* FROM `orders` WHERE cus\_id = '1';

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

order\_id

item\_id

item\_qty

cus\_id

1 4 2 1

3 9 2 1

Check all

With selected:

Edit

Copy

Delete

Export

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Query results operations

Print

Copy to clipboard

Export

Display chart

Create view

Bookmark this SQL query

Console

Here, we're examining the "orders" table and ran a SQL query to retrieve all orders associated with customer ID 1. The results display two rows, confirming that this customer has placed two orders.



5. DISPLAYING TOTAL QUANTITY SOLD PER ITEMS:

phpMyAdmin

Recent Favorites

New

information\_schema

mysql

performance\_schema

phpmyadmin

sample

New

customer\_profile

cus\_contact\_info

items

orders

user

test

Server: 127.0.0.1 » Database: sample » Table: items

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Triggers

Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total, Query took 0.0006 seconds)

SELECT items.item\_name, SUM(orders.item\_qty) AS total\_qty\_sold FROM items JOIN orders ON items.item\_id= orders.item\_id GROUP BY items.item\_name;

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

item_name	total_qty_sold
bacon belgian waffle	2
cappuccino	2
classic lasagna	1
french butter croissant	1
iced classic chocolate	1

Show all | Number of rows: 25 | Filter rows: Search this table

Here, the total quantity sold for each product were presented, displaying only descriptive columns and total quantity sold. This enables us to present our information in a more concise and clearer perspective while putting emphasis on the quantities.

phpMyAdmin

Recent Favorites

New

information\_schema

mysql

performance\_schema

phpmyadmin

sample

New

customer\_profile

cus\_contact\_info

items

orders

user

test

Server: 127.0.0.1 » Database: sample » Table: items

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Triggers

Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total, Query took 0.0005 seconds)

SELECT items.item\_name, SUM(orders.item\_qty) AS total\_qty\_sold FROM items JOIN orders ON items.item\_id= orders.item\_id GROUP BY items.item\_name HAVING SUM(orders.item\_qty)>0;

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

item_name	total_qty_sold
bacon belgian waffle	2
cappuccino	2
classic lasagna	1
french butter croissant	1
iced classic chocolate	1

Show all | Number of rows: 25 | Filter rows: Search this table

6. DISPLAYING ITEMS HAVING MORE THAN 0 SALES:

Here, we displayed all the items that contains more than 0 sales using the HAVING. This enables to filter out informations particularly the items that contains more than 0 sales.



phpMyAdmin interface showing the 'items' table. The query executed is:

```
SELECT i.item_name, SUM(o.item_qty * i.item_price) AS sales_amount FROM items AS i JOIN orders AS o ON i.item_id = o.item_id GROUP BY i.item_name;
```

The results show the sales amount for each item:

item_name	sales_amount
bacon belgian waffle	198.00
cappuccino	190.00
classic lasagna	159.00
french butter croissant	125.00
iced classic chocolate	105.00

7.DISPLAY SALES

We printed the sales per item. In this case we only needed to print the item name along with the sales amount. Through this, we were able to visualize the total income that the sellers receive. Hence, being able to narrow down the information into just two columns allowed us to easily analyze how the sales were doing per item, which enables us to identify which among the products are mostly patronized by customers.

phpMyAdmin interface showing the 'orders' table. The query executed is:

```
SELECT DATE (order_date) AS order_day, SUM(orders.item_qty * items.item_price) AS sales_per_day FROM orders JOIN items ON items.item_id=orders.item_id GROUP BY DATE (orders.order_date);
```

The results show the sales per day:

order_day	sales_per_day
2024-02-10	349.00
2024-04-04	198.00
2024-05-05	125.00
2024-05-15	105.00

8. DISPLAYING THE TOTAL SALES PER DAY:

Here, we displayed the order date along with the sales per day, allowing us to visualize and easily identify the total number of sales per day.





localhost/phpmyadmin/index.php?route=/table/sql&db=sample&table=orders

phpMyAdmin

Server: 127.0.0.1 > Database: sample > Table: items

Showing rows 0 - 4 (5 total. Query took 0.0005 seconds.)

```
SELECT items.item_name, SUM(orders.item_qty * items.item_price) AS tota_sales, SUM(orders.item_qty) as quantity_per_item FROM orders JOIN items ON items.item_id = orders.item_id GROUP BY items.item_name;
```

item_name	tota_sales	quantity_per_item
bacon belgian waffle	198.00	2
cappuccino	190.00	2
classic lasagna	159.00	1
french butter croissant	125.00	1
iced classic chocolate	105.00	1

9. DISPLAYING THE TOTAL SALES AND QUANTITY PER ITEMS AND DAY:

Here, we displayed the item name along with quantity per item and the total sales, visualizing and displaying a more detailed perspective of the items, quantity and item sales.

10. DISPLAYING THE TOTAL SALES TOTAL QUANTITY SOLD PER ITEM AND DAY

localhost/phpmyadmin/index.php?route=/table/sql&db=sample&table=items

phpMyAdmin

Server: 127.0.0.1 > Database: sample > Table: items

Showing rows 0 - 4 (5 total. Query took 0.0006 seconds.)

```
SELECT items.item_name, SUM(items.item_price * orders.item_qty) AS Total_sales, DATE (orders.order_date) AS Order_day, SUM(orders.item_qty) AS Total_qty FROM orders JOIN items ON items.item_id=orders.item_id GROUP BY items.item_name, DATE(orders.order_date) ORDER BY 'order_day' DESC;
```

item_name	Total_sales	Order_day	Total_qty
cappuccino	190.00	2024-02-10	2
classic lasagna	159.00	2024-02-10	1
bacon belgian waffle	198.00	2024-04-04	2
french butter croissant	125.00	2024-05-05	1
iced classic chocolate	105.00	2024-05-15	1

11. DISPLAYING THE TOTAL SALES AND TOTAL QUANTITY SOLD PER FULLNAME

localhost/phpmyadmin/index.php?route=/table/sql&db=sample&table=items

phpMyAdmin

Server: 127.0.0.1 > Database: sample > Table: customer\_profile

Showing rows 0 - 3 (4 total. Query took 0.0006 seconds.)

```
SELECT customer_profile.name AS fullname, SUM(orders.item_qty * items.item_price) AS total_sales, SUM(orders.item_qty) AS total_qty_sold FROM orders JOIN items ON items.item_id=orders.order_id JOIN customer_profile ON customer_profile.cus_id=orders.cus_id GROUP BY customer_profile.name;
```

fullname	total_sales	total_qty_sold
Ana	105.00	1
Claries	88.75	1
JB	298.00	4
JM	95.00	1



12. DISPLAYING THE NUMBERR OF ITEMS BROUGHT BY CUSTOMERS PROFILE

The screenshot shows the phpMyAdmin interface with a SQL query executed. The query is: `SELECT user.username, COUNT(*) AS items_bought FROM orders JOIN customer_profile ON customer_profile.ua_id=orders.cus_id JOIN user ON user.ua_id=customer_profile.ua_id GROUP BY user.username;` The result table shows the following data:

username	items_bought
Ana	1
Claries	1
JB	2
JM	1

USING COUNT (\*)

Here, we displayed the number of items purchased by each customer username through using the COUNT (\*) function in the SQL Query, allowing us to count and identify the total number of purchases associated with each customer/user

The screenshot shows the phpMyAdmin interface with a SQL query executed. The query is: `SELECT items.item_name FROM items WHERE items.item_name LIKE 'C%';` The result table shows the following data:

item_name
caramel macchiato
cappuccino
classic lasagna

13. THE USE OF WILDCARD:

In this case, we used the wildcards to display the items from particular string starting. Additionally, we also used the SQL LIKE operator together with a wildcard character in order to filter out the result, to showcase only the items that matches and corresponds with our search criteria. Through this, displaying and picking relevant items are made easier.