# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №4**

# з курсу

**«Бази даних»**

*Студента 2 курсу*

*групи ПП-21 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

Краснощок В.М.

## Київ – 202

**1.Назва роботи**

Захист інтерфейсу користувача від SQL-ін’єкцій. Виконання складних запитів та процедур.

1. **Тема роботи**

Захист інтерфейсу користувача від SQL-ін’єкцій. Виконання складних запитів та процедур.

1. **Мета роботи**

Набуття навичок захисту інтерфейсу користувача від SQL-ін’єкцій.

1. **Умова завдання**

Відповідно до обраного варіанту та індивідуального завдання:

1) створити форми для виводу результатів запитів;

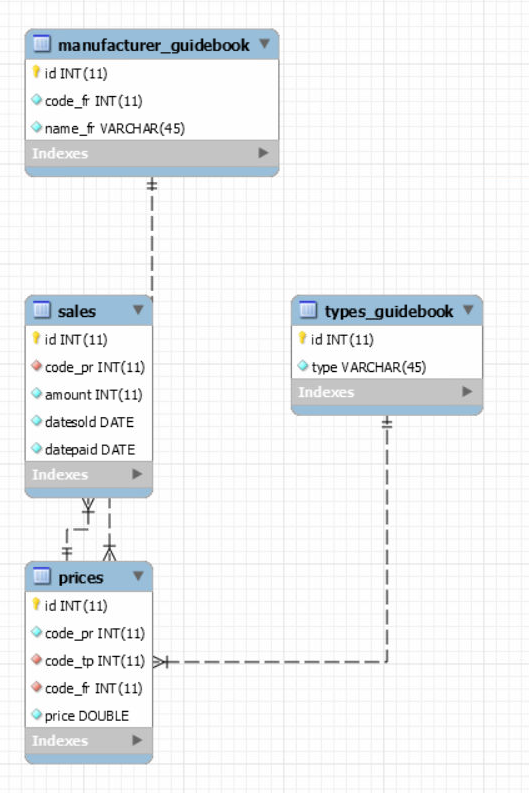
2) виконати екранування полів вводу даних;

3) розробити шлаблони перевірки даних, які мають певну структуру;

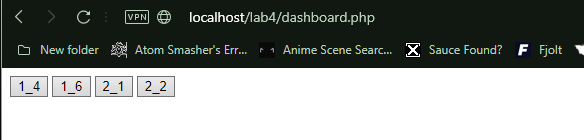
4) захистити форми з передачою даних від ін’єкцій об’єднання та екранування «хвоста» запиту;

1. **Рішення**

Діаграма:



Головна сторінка:



<html>

<head>

<meta charset="UTF-8">

<title>Seletskiy (modul)</title>

</head>

<body>

<form method="post" id="contact-form">

<input type="button" name="Insert\_Ad" value="1\_4" onclick="location.href='1\_4.php'">

<input type="button" name="Insert\_Ad" value="1\_6" onclick="location.href='1\_6.php'">

<input type="button" name="Insert\_Ad" value="2\_1" onclick="location.href='2\_1.php'">

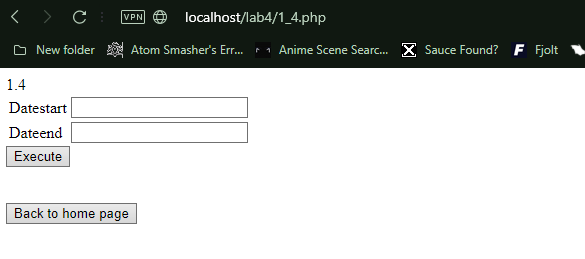
<input type="button" name="Insert\_Ad" value="2\_2" onclick="location.href='2\_2.php'">

</form>

</body>

</html>

Запит 1.4:



<form>

1.4

<table>

<tr>

<td>Datestart</td>

<td><input type="text" name="t1"></td>

</tr>

<tr>

<td>Dateend</td>

<td><input type="text" name="t2"></td>

</tr>

</table>

<input name="button" type="submit" formmethod="post" value="Execute">

<br><br><br><input type="submit" formaction="dashboard.php" value="Back to home page">

</form>

<?php

require "connection.php";

if(isset($\_POST['button'])){

if (empty($\_POST['t1'])) $\_POST['t1']=date('2000-01-01') ;

if (empty($\_POST['t2'])) $\_POST['t2']=date('2012-12-31');

$query = "call 1\_4(\"".$\_POST['t1']."\", \"".$\_POST['t2']."\")";

$ver = mysqli\_query($connection, $query);

if(!$ver) exit(mysql\_error());

echo "<table border = 1>";

echo "<tr>

<td> Type </td>

<td> Firm's name </td>

<td> Cost </td>

<td> Date sold </td>

<td> Date paid </td>

</tr>";

while(list($type, $name\_fr, $cost, $datesold, $datepaid) = mysqli\_fetch\_row($ver)){

echo "<tr>

<td> $type </td>

<td> $name\_fr </td>

<td> $cost </td>

<td> $datesold </td>

<td> $datepaid </td>

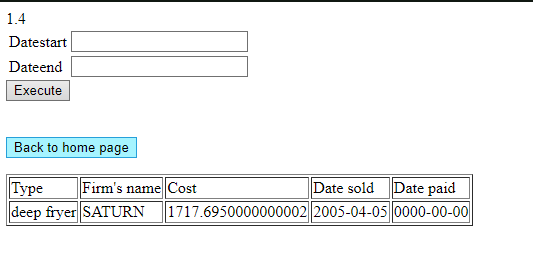
</tr>";

}

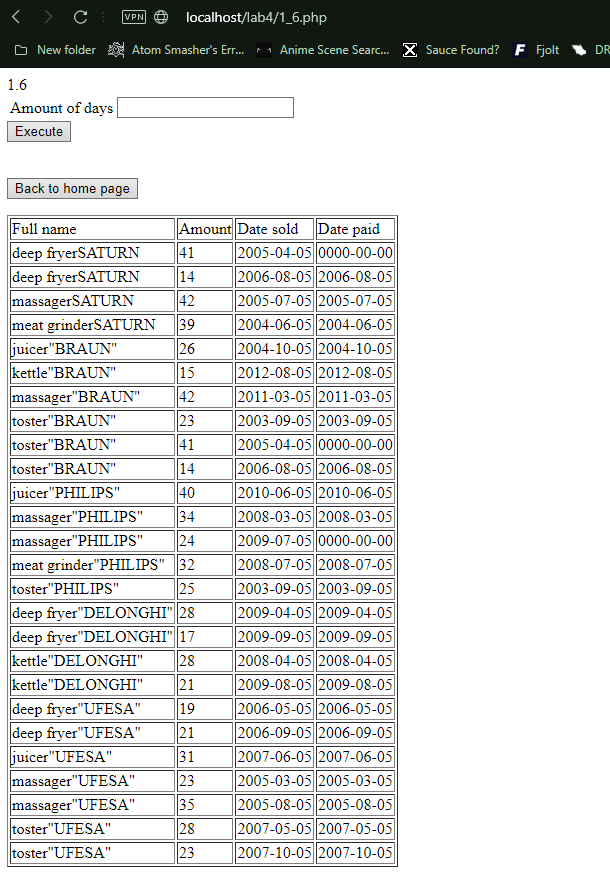
echo "</table>";

}

?>



Запит 1.6:



<form>

1.6

<table>

<tr>

<td>Amount of days</td>

<td><input type="text" name="t1"></td>

</tr>

</table>

<input name="button" type="submit" formmethod="post" value="Execute">

<br><br><br><input type="submit" formaction="dashboard.php" value="Back to home page">

</form>

<?php

require "connection.php";

if(isset($\_POST['button'])){

if (empty($\_POST['t1'])) $\_POST['t1']='2000';

$query = "call 1\_6(\"".$\_POST['t1']."\")";

$ver = mysqli\_query($connection, $query);

if(!$ver) exit(mysql\_error());

echo "<table border = 1>";

echo "<tr>

<td> Full name</td>

<td> Amount </td>

<td> Date sold </td>

<td> Date paid </td>

</tr>";

while(list($fullname, $amount, $datesold, $datepaid) = mysqli\_fetch\_row($ver)){

echo "<tr>

<td> $fullname </td>

<td> $amount </td>

<td> $datesold </td>

<td> $datepaid </td>

</tr>";

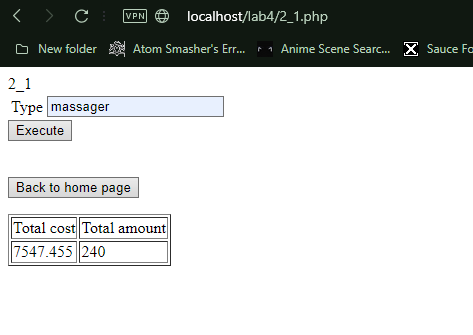
}

echo "</table>";

}

?>

Запит 2.1:



<form>

2\_1

<table>

<tr>

<td>Type</td>

<td><input type="text" name="t1"></td>

</tr>

</table>

<input name="button" type="submit" formmethod="post" value="Execute">

<br><br><br><input type="submit" formaction="dashboard.php" value="Back to home page">

</form>

<?php

require "connection.php";

if(isset($\_POST['button'])){

if (empty($\_POST['t1'])) $\_POST['t1']=date('2000-01-01');

$query = "call 2\_1(\"".$\_POST['t1']."\")";

$ver = mysqli\_query($connection, $query);

if(!$ver) exit(mysql\_error());

echo "<table border = 1>";

echo "<tr>

<td> Total cost </td>

<td> Total amount </td>

</tr>";

while(list($cost, $total) = mysqli\_fetch\_row($ver)){

echo "<tr>

<td> $cost </td>

<td> $total </td>

</tr>";

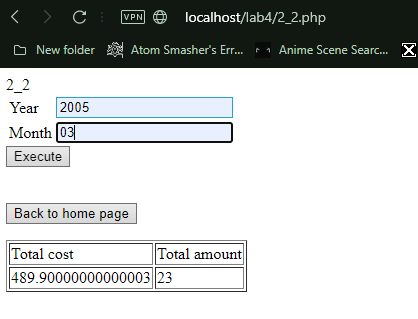
}

echo "</table>";

}

?>

Запит 2.2:



<form>

2\_2

<table>

<tr>

<td>Year</td>

<td><input type="text" name="t1"></td>

</tr>

<tr>

<td>Month</td>

<td><input type="text" name="t2"></td>

</tr>

</table>

<input name="button" type="submit" formmethod="post" value="Execute">

<br><br><br><input type="submit" formaction="dashboard.php" value="Back to home page">

</form>

<?php

require "connection.php";

if(isset($\_POST['button'])){

if (empty($\_POST['t1'])) $\_POST['t1']='2000';

if (empty($\_POST['t2'])) $\_POST['t2']='12';

$query = "call 2\_2(\"".$\_POST['t1']."\", \"".$\_POST['t2']."\")";

$ver = mysqli\_query($connection, $query);

if(!$ver) exit(mysql\_error());

echo "<table border = 1>";

echo "<tr>

<td> Total cost </td>

<td> Total amount </td>

</tr>";

while(list($cost, $total) = mysqli\_fetch\_row($ver)){

echo "<tr>

<td> $cost </td>

<td> $total </td>

</tr>";

}

echo "</table>";

}

?>

**Контрольні питання**

1. Що таке SQL-ін’єкція?

Один з поширених способів злому сайтів та програм, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду.

1. Які види та способи SQL-ін’єкцій ви знаєте?

Даними, переданими через методи POST і GET

Значеннями [HTTP-Cookie]

HTTP\_REFERER (для скриптів)

AUTH\_USER та AUTH\_PASSWORD (при використанні аутентифікації)

1. Особливості методів POST та GET? Який з цих методів більш захищений від SQL-ін’єкцій?

POST використовується для відправки даних, GET - для отримання. Обидва методи вразливі для SQL-ін’єкцій, тому що вони не залежать від POST, GET та навіть HTTP

1. Яке призначення функції mysqli\_escape\_string()?

Замінити спеціальні символи на символьні сполучення, що репрезентують дані символи в SQL.

1. Як задається шаблон перевірки даних?

LIKE або NOT LIKE

1. Що таке регулярні вирази?

Вираз, заданий одним з вищевказаних шаблонів

1. Що таке підготовлений вираз?

Вирази екземпляру інтерфейса PreparedStatement: вони "пам'ятають" скомпільовані SQL-вирази. Саме тому вони називаються "prepared" ("підготовлені").

1. Що таке інтерфейс користувача.

Засіб зручної взаємодії користувача з інформаційною системою

1. Основні елементи керування інтерфейсу користувача.

Кнопки, форми, поля вводу, текст

1. Значення тегу <form> …</form>.

Створює форму (з кнопками, полями вводу, значення яких можна використати в методі POST, наприклад)

1. Значення тегу <table> …</table>.

Створює таблицю

1. Що таке повнотекстний пошук?

У повнотекстовому пошуці, пошуковик аналізує всі слова в кожному збереженому документі, та намагається відповідати критеріям пошуку (наприклад, тексту визначеному користувачем)

1. Що таке релевантність?

Відповідність результатів пошуку заданим критеріям

1. Значення оператора MATCH (…) AGAINST (…).

В першу змінну записуються колонки, в яких буде виконано пошук тексту, записаного в другу змінну.

1. **Висновки**

В результаті виконання даної лабораторної роботи я набув навичок захисту інтерфейсу користувача від SQL-ін’єкцій. Створив потрібні скрипти, що виконують процедуру, захищену від SQL-ін’єкцій через вказані параметри та виводять результат виконання вищевказаних процедур на екран. Вважаю дану лабораторну роботу виконаною в повному обсязі.