

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА**



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до практичної роботи №4

з курсу

«Бази Даних»

студента 2 курсу

групи ПП-22

спеціальності 122 «Комп'ютерні науки»

ОП «Прикладне програмування»

Шевлюк Вікторії Віталіївни

Викладач:

асистент

Криволапов Я. В.

Київ – 2022

Тема: Захист інтерфейсу користувача від SQL-ін'єкцій. Виконання складних запитів та процедур.

Мета: Набуття навичок захисту інтерфейсу користувача від SQL-ін'єкцій.

Завдання:

За допомогою PHP-скриптів реалізувати виконання запитів 1.4, 1.6, 2.1, 2.2. Запити мають бути захищеними від SQL-ін'єкцій, а результат виконання виводиться у вигляді таблиці.

1.4 Інформацію про реалізовану за певний період часу та не сплачену продукцію кондитерської фабрики. Дата впровадження (початковий та кінцевий термін) повинна задаватись під час виконання запиту у вигляді параметра, передбачити також можливість отримання інформації для всього періоду часу. Динамічний набір записів повинен мати наступні поля: Повна назва продукції, Кількість продукції, Дата реалізації, Дата сплати, Прогноз ціни, Реалізація.

1.6 Визначити дані про реалізацію кондитерських виробів за останні дні, за основу взяти кінцеву дату реалізації. Кількість останніх днів має вводиться у вигляді параметра. Динамічний набір записів складається з таких полів: Технолог цеху, Повна назва продукції, Кількість продукції, Дата реалізації, Дата сплати.

2.1 Для кондитерської продукції загальну кількість виробництва та загальну реалізацію (назва повинно задаватися під час виконання запиту у вигляді параметра, передбачити можливість отримання інформації про всю продукцію одночасно);

2.2 Для технологів загальну кількість виробництва кондитерської продукції та загальну реалізацію готової продукції за деякий місяць деякого року (значення параметрів для розрахункових полів Рік та Місяць, що будуються по полю Дата впровадження, повинні вводиться під час виконання запиту).

Хід роботи:

Для того, щоб захистити користувача від SQL-ін'єкцій я використала можливості такі СУБД як процедури.

Роботу починаємо з того, що створюємо процедури у нашій базі даних.

Нижче наведені скріншоти моїх процедур:

► Процедура 1_4

The screenshot shows a window titled 'Изменить' (Edit) for a database procedure. It has a tabbed interface with 'Детали' (Details) selected. The 'Имя процедуры' (Procedure name) is 'proc1_4' and the 'Тип' (Type) is 'PROCEDURE'. Below this is a table for 'Параметры' (Parameters) with columns: 'Направление' (Direction), 'Имя' (Name), 'Тип' (Type), 'Длина/Значения' (Length/Values), and 'Параметры' (Parameters). There are two parameters: 'prod' and 'splt', both with direction 'IN' and type 'VARCHAR'. At the bottom is a text area for 'Определение' (Definition) containing a SQL query. The window has 'Вперёд' (Next) and 'Заккрыть' (Close) buttons at the bottom right.

Направление	Имя	Тип	Длина/Значения	Параметры
IN	prod	VARCHAR	---	Удалить
IN	splt	VARCHAR	---	Удалить

Добавить параметр

```
1 SELECT asort.NAME, realis.KIL, realis.DATAOPL, realis.DATAPROPL,
2 realis.KIL*(if (realis.KIL<20, product_per_day.COBIV*1.27,
3 product_per_day.COBIV+1.25))
4 FROM asort, realis, product_per_day, tech, product
5 WHERE
6 realis.DATAOPL >= prod and realis.DATAOPL <= splt
7 and (realis.DATAPROPL > splt or realis.DATAPROPL = '0000-00-00')
8 and product_per_day.KPROD = asort.KPROD
9 and realis.KPROD = asort.KPROD
10 and product.K_TPR = asort.K_TPR
and product_per_day.KCEX = tech.KCEX
```

Вперёд Заккрыть

► Процедура 1_6

Изменить

Детали

Имя процедуры

proc1_6

Тип

PROCEDURE

Параметры

Направление

Имя

Тип

Длина/Значения

Параметры

↕

IN

last

I

Удалить

Добавить параметр

Определение

```
1 SELECT concat(tech.name, " ", tech.ima, " ", tech.pobat),
2      concat(product.TUP_PROD, " ", asort.name) , realis.KIL,
3      realis.DATAOPL, realis.DATAPROPL
4 FROM asort, product, product_per_day, realis, tech
5 WHERE(
6 ((select max(DATAOPL) from realis) - interval lastdays day)<=
7   DATAOPL
8   and product_per_day.KPROD =asort.KPROD
9   and product_per_day.KCEX = tech.KCEX
10  and asort.K_TPR = product.K_TPR
11  and realis.KPROD = asort.KPROD
12 )
```

Вперёд

Закрыть

► Процедура 2_1

Изменить

Детали

Имя процедуры

proc2_1

Тип

PROCEDURE

Параметры

Направление

Имя

Тип

Длина/Значения

Параметры

↑

IN

myf

↓

45

utf8mb4

Удалить

Добавить параметр

Определение

```
1 SELECT asort.NAME,sum(realis.KIL), sum(realis.KIL*if
  (realis.KIL<20, product_per_day.COBIV*1.27,
  product_per_day.COBIV+1.25))
2 FROM asort, product_per_day, product, realis, tech
3 WHERE(
4   product_per_day.KPROD =asort.KPROD
5     and product_per_day.KCEX = tech.KCEX
6     and asort.K_TPR = product.K_TPR
7     and realis.KPROD = asort.KPROD
8 )
9 GROUP BY asort.NAME
10 HAVING asort.NAME like concat(myprod,'%')
```

Вперёд

Закреть

► Процедура 2_2

Изменить

Детали

Имя процедуры

proc2_2

Тип

PROCEDURE

Параметры

	Направление	Имя	Тип	Длина/Значения	Параметры	
↑	IN	myr	I	11		Удалить
↑	IN	my	I	11		Удалить

Добавить параметр

Определение

```

1 SELECT concat(mymonth, 'місяць', myyear, 'рік'),
2         concat(tech.name, " ", tech.ima, " ", tech.pobat),
3         sum(realis.KIL) ,
4         sum(realis.KIL*if (realis.KIL<20, product_per_day.COBIV*1.27,
5                             product_per_day.COBIV+1.25))
5 FROM asort, product_per_day, product, realis, tech
6 WHERE(
7 product_per_day.KPROD =asort.KPROD
8   and product_per_day.KCEX = tech.KCEX
9   and asort.K_TPR = product.K_TPR
10  and realis.KPROD = asort.KPROD
11  and (month(realis.DATAOPL) = mymonth and year(realis.DATAOPL)

```

Вперёд

Закреть

Тепер напишемо скрипти, що будуть виконувати наші процедури через графічний інтерфейс. У кожному скрипті будуть створюватись поля для введення інформації від користувача та кнопки для підтвердження запиту, після натискання якої будуть виводитись відповідні таблиці.

Розглянемо код скрипту для першої процедури:

```
<form method="get">

DATAOPL:<br>

<input type="text" name="rel"><br>

<br>

DATAPROPL:<br>

<input type="text" name="splt"><br>

<br>

<input type="submit" value="GO">

</form>

<br> <br>

<?php
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
if(isset($_GET['rel'], $_GET['splt']))
{
include ("config.php");
$query = "call proc1_4 ('$_GET[rel]', '$_GET[splt]')";
echo "<br><br>";

$ver=mysqli_query($dbcon,$query);

if(!$ver){
echo "<P>Не вдалося виконати запит</P>";
exit(mysqli_error($dbcon));
```

```
}
```

```
echo "<P><B> Запит 1.4 </B></P>";
```

```
echo "<table border=1>";
```

```
while(list($NAME, $TUP_PROD, $DATAOPL, $DATAPROPL) = mysqli_fetch_row($ver))
```

```
{
```

```
echo "<tr>
```

```
    <td> $NAME </td>
```

```
    <td> $TUP_PROD </td>
```

```
    <td> $DATAOPL </td>
```

```
    <td> $DATAPROPL </td>
```

```
    </td>";
```

```
}
```

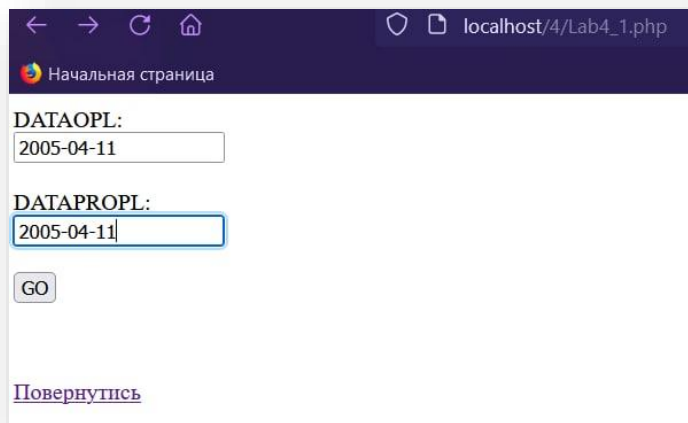
```
echo "</table>";
```

```
echo "<P>  </P>";
```

```
}
```

```
?>
```


Результат виконання цього скрипту виглядає наступним чином:



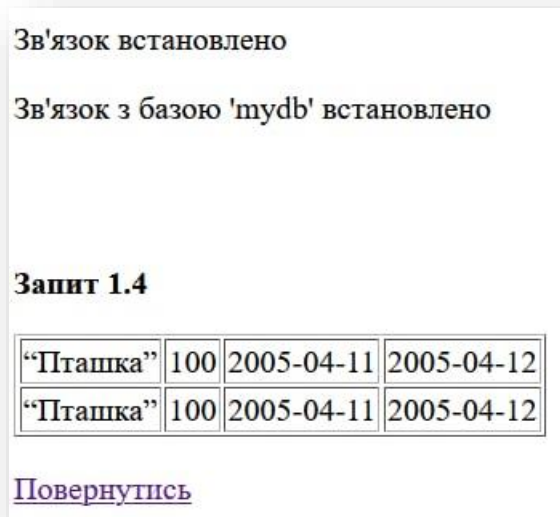
← → ↻ 🏠 localhost/4/Lab4_1.php

Начальная страница

DATAOPL:

DATAPROPL:

[Повернутись](#)



Зв'язок встановлено

Зв'язок з базою 'mydb' встановлено

Запит 1.4

"Пташка"	100	2005-04-11	2005-04-12
"Пташка"	100	2005-04-11	2005-04-12

[Повернутись](#)

Оскільки за варіантом дедлайн сплати має бути в день реалізації, в полях я вказую однакову дату.

Тепер переглянемо скрипт для другої процедури:

```
<form method="get">
  Lastdays:<br>
  <input type="text" name="lastdays"><br>
  <br>
  <input type="submit" value="GO">
</form>
<br> <br>
```

```

<?php
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
if(isset($_GET['lastdays']))
{
include ("config.php");
$query = "call proc1_6 ('".$_GET['lastdays'])";
echo "<br><br>";

$ver=mysqli_query($dbcon,$query);

if(!$ver){
echo "<P>Не вдалося виконати запит</P>";
exit(mysqli_error($dbcon));
}

echo "<P><B> Запит 1.6 </B></P>";
echo "<table border=1>";
while(list($fullTechName, $fullProdName, $KIL, $DATAOPL, $DATAPROPL) =
mysqli_fetch_row($ver))
{
echo "<tr>
        <td> $fullTechName </td>
        <td> $fullProdName </td>
        <td> $KIL </td>
        <td> $DATAOPL </td>
        <td> $DATAPROPL </td>
        </td>";
}

```

```

echo "</table>";

echo "<P> </P>";

}

?>

```

Результат роботи другої процедури:

← → ↻ 🏠 localhost/4/Lab4_2.php

Начальная страница

Lastdays:

[Повернутись](#)

Зв'язок встановлено

Зв'язок з базою 'mydb' встановлено

Запит 1.6

“Проценко” “Іван” “Семенович”	Крамель “Пташка”	58	2005-06-22	2005-06-23
“Проценко” “Іван” “Семенович”	Крамель “Молочна”	56	2005-06-26	2005-06-27
“Проценко” “Іван” “Семенович”	Крамель “Театральна”	113	2005-06-30	2005-07-01
“Кравченко” “Кирило” “Сергійович”	Цукерки “Мулатка”	118	2005-07-05	2005-07-15
“Кравченко” “Кирило” “Сергійович”	Цукерки “Хід королеви”	129	2005-07-08	2005-07-09
“Кравченко” “Кирило” “Сергійович”	Цукерки “Ромашка”	201	2005-07-12	0000-00-00
“Кравченко” “Кирило” “Сергійович”	Цукерки “Ромашка”	201	2005-07-12	0000-00-00
“Жовніров” “Юрій” “Петрович”	Шоколад “Чайка”	203	2005-07-16	2005-07-27
“Жовніров” “Юрій” “Петрович”	Шоколад “Оленка”	90	2005-07-20	2005-07-21
“Жовніров” “Юрій” “Петрович”	Шоколад “Гвардейский”	92	2005-06-18	2005-06-19

[Повернутись](#)

Тепер до скрипту під третю процедуру:

```
<form method="get">
  Production_name:<br>
  <input type="text" name="myprod"><br>
  <br>
  <input type="submit" value="GO">
</form>
<br> <br>

<?php
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
if(isset($_GET['myprod']))
{
include ("config.php");
$query = "call proc2_1 ('$_GET[myprod]')";
echo "<br><br>";

$ver=mysqli_query($dbcon,$query);

if(!$ver){
echo "<P>Не вдалося виконати запит</P>";
exit(mysqli_error($dbcon));
}

echo "<P><B> Запит 2.1 </B></P>";
echo "<table border=1>";
while(list($NAME, $fullAMOUNT, $fullREALISATION) = mysqli_fetch_row($ver))
{
echo "<tr>
      <td> $NAME </td>
      <td> $fullAMOUNT </td>
      <td> $fullREALISATION </td>
    </td>";
}

echo "</table>";
echo "<P>  </P>";

}
?>
```

Результат роботи цього скрипту:

← → ↻ 🏠 localhost/4/Lab4_3.php

🦊 Начальная страница

Production name:

[Повернутись](#)

Зв'язок встановлено

Зв'язок з базою 'mydb' встановлено

Запит 2.1

"Пташка"	383	2401.4100
----------	-----	-----------

[Повернутись](#)

I останній скрипт для четвертої процедури:

```
<form method="get">
  MONTHL:<br>
  <input type="text" name="month"><br>
  <br>
  YEAR:<br>
  <input type="text" name="year"><br>
  <br>
  <input type="submit" value="GO">
</form>
<br> <br>

<?php
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
if(isset($_GET['month'], $_GET['year']))
{
include ("config.php");
$query = "call proc2_2 ('$_GET'month]', '$_GET[year]')";
echo "<br><br>";

$ver=mysqli_query($dbcon,$query);

if(!$ver){
echo "<P>Не вдалося виконати запит</P>";
exit(mysqli_error($dbcon));
}

echo "<P><B> Запит 1.4 </B></P>";
echo "<table border=1>";
while(list($NAME, $sec, $third, $forth) = mysqli_fetch_row($ver))
{
echo "<tr>
      <td> $NAME </td>
      <td> $sec </td>
      <td> $third </td>
      <td> $forth </td>
    </td>";
}

echo "</table>";
echo "<P> </P>";

}
?>
```

Результат його роботи:

MONTH:

YEAR:

Зв'язок встановлено

Зв'язок з базою 'mydb' встановлено

Запит 2.2

5місяць2005рік	"Проценко" "Іван" "Семенович"	917	4388.4900
----------------	-------------------------------	-----	-----------

[Повернутись](#)

Висновок: у ході даної лабораторної роботи я набула навичок захисту інтерфейсу користувача від SQL-ін'єкцій за допомогою створення процедур, а також покращила свої навичку у створюванні графічних інтерфейсів для користувачів за допомогою php-скриптів.

Контрольні питання:

1. Що таке SQL-ін'єкція?

SQL ін'єкція — один з поширених способів злому сайтів та програм, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду.

2. Які види та способи SQL-ін'єкцій ви знаєте?

Використання UNION, Використання UNION + group_concat(), Екранування хвоста запиту, розщеплення запиту.

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

Метод POST використовується для запису даних із елементів графічного інтерфейсу.

Метод GET відображає записані дані з параметрами у пошуковому рядку, що саме по собі є не дуже безпечним.

4. Яке призначення функції mysqli_escape_string()?

Ця функція допомагає уникати спеціальних символів у рядках.

5. Що таке регулярні вирази?

Регулярні вирази це спеціальні правила що визначають символи, які можуть з'являтися у виразах.

6. Що таке інтерфейс користувача.

Інтерфейс користувача — засіб зручної взаємодії користувача з інформаційною системою.

7. Основні елементи керування інтерфейсу користувача.

Текстові поля, кнопки, чекбокси, радіо-кнопки, випадаючі менюшки

8. Значення тегу <form> ...</form>.

Тег для створення форми

9. Значення тегу <table> ...</table>.

Тег для створення таблиці.

10. Що таке повнотекстний пошук?

Повнотекстовий пошук - це комплексний метод пошуку, який порівнює кожне слово запиту пошуку з кожним словом у документі чи базі даних.

11. Що таке релевантність?

Релевантність — міра відповідності отриманого результату бажаному.

12. Види повнотекстного пошуку.

Існує три типи повнотекстового пошуку:

- ▶ Повнотекстовий пошук природною мовою.
- ▶ Логічний повнотекстовий пошук.
- ▶ Пошуки розширення запитів.

13. Значення оператора MATCH (...) AGAINST (...).

Для повнотекстового пошуку в MySQL використовується конструкція `MATCH (filelds) ... AGAINST (words)`. Вона може працювати в різних режимах, які досить сильно відрізняються між собою. Для всіх діє таке правило: дана конструкція повертає умовну релевантність, але спосіб обчислення якої може бути різним залежно від режиму.