

**МІНІСТЕРСТВО АГРАРНОЇ ПОЛІТИКИ УКРАЇНИ
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ
УНІВЕРСИТЕТ**

Кафедра економічної кібернетики

**ТЕХНОЛОГІЯ СТВОРЕННЯ ПРОГРАМНИХ
ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ**

МЕТОДИЧНІ ВКАЗІВКИ

**для виконання лабораторних робіт
бакалаврами напрямку 6.050100– «Економічна кібернетика»**

Вінниця 2010

Технологія створення програмних інтелектуальних систем. Методичні вказівки для виконання лабораторних робіт бакалаврами напряму 6.050100– «Економічна кібернетика» / Паламарчук Є.А., Яцковська Р.О. - Вінниця: ВНАУ, 2010.- 68с.

Укладачі: Паламарчук Є.А., доцент, к.т.н.
Яцковська Р.О., асистент

Рецензенти: Кравченко Ю.С., к.ф.-м.н., доцент кафедри електроніки ВНТУ

Гудзенко Н.М. к.е.н., доцент кафедри організації обліку та звітності ВНАУ

Коротка анотація

Методичні вказівки для виконання лабораторних робіт з дисципліни "Технологія створення програмних інтелектуальних систем" містять структурно-модульну схему курсу, розбиття дисципліни на модулі та розподіл балів за категоріями діяльності студента, 10 лабораторних робіт, вимоги до виконання та оформлення лабораторних робіт та список рекомендованої літератури.

Призначені для використання бакалаврами напряму 6.050100– «Економічна кібернетика».

НАУКОВО - МЕТОДИЧНЕ ВИДАННЯ

**Рекомендовано науково-методичною радою
Вінницького національного аграрного університету
протокол № 9 від „31” травня 2010 р.**

НАУКОВО - МЕТОДИЧНЕ ВИДАННЯ

Паламарчук Євген Анатолійович,
Яцковська Римма Олександрівна,

ТЕХНОЛОГІЯ СТВОРЕННЯ ПРОГРАМНИХ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

Набір і редагування авторські

Технічний редактор *Олександр Романов*

Верстка

Підписано до друку Формат 60x84/16.
Папір офсетний. Друк різнографічний.
Тираж прим.

Віддруковано у редакційно-видавничому відділі
Вінницького державного аграрного університету
21008, м. Вінниця, вул. Сонячна, 3

ДЖЕРЕЛА В INTERNET

1. <http://programer.org.ua/index.php?mainpart=2>
2. <http://www.znannya.org/>
3. <http://uk.wikipedia.org>
4. www.diasoft.ru
5. www.dtkr.ua
6. www.knigka.org.ua/category/office_applications
7. www.programbank.ru
8. www.softlab.ru
9. www.vsau.vin.ua
10. www.williamspublishing.com

ЗМІСТ

Вступ.....	4
Структура орієнтовного розподілу балів за модулями навчальної дисципліни, що присвоюється студентам.....	5
Лабораторна робота №1.....	7
Лабораторна робота № 2.....	14
Лабораторна робота № 3.....	29
Лабораторна робота № 4.....	33
Лабораторна робота № 5.....	37
Лабораторна робота № 6.....	43
Лабораторна робота № 7.....	45
Лабораторна робота № 8.....	48
Лабораторна робота № 9.....	56
Лабораторна робота №10.....	59
Список рекомендованої літератури	65

ВСТУП

Мета: надання знань з теоретичних засад і принципів побудови структур та моделей даних в сучасних програмних продуктах; набуття вмінь роботи з структурами та моделями даних за допомогою розробки власних прикладних програм на мові PHP; формування навиків налагодження програмних додатків з метою доступу до даних.

Завдання: вивчення теоретичних основ представлення інформації в пам'яті ЕОМ та характеристик прикладного програмного забезпечення по управлінню даними, формування уявлення у студентів про реляційну модель даних, застосування набутих знань, умінь та навиків для побудови власних програмних додатків маніпулювання даними.

Предмет: система засобів автоматизації для роботи з даними, а саме:

- засоби створення структур даних;
- засоби моделювання великих обсягів даних;
- засоби доступу до даних за допомогою сучасного програмного забезпечення.

Викладений матеріал методичних вказівок для виконання лабораторних робіт з дисципліни "Технологія створення програмних інтелектуальних систем" розрахований на студентів 3-го курсу факультету економіки та підприємництва.

У лабораторних роботах наведено завдання, короткі теоретичні відомості для закріплення знань студентів та пояснення практичних задач, що полегшує роботу при самостійному виконанні практичних завдань.

Оцінювання знань студентів здійснюється з використанням модульно-рейтингової системи оцінювання знань, при якій викладач оцінює знання і роботу студента у балах. Студент в свою чергу має ознайомитися з виконанням лабораторної роботи (кількість заходів –10), виконати завдання, дати відповіді на контрольні запитання, які наведені в кінці лабораторної роботи, оформити звіт в зошиті і захистити лабораторну роботу.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Максим Кузнецов, Ігор Симдянов, Сергій Голишев. PHP 5. Практика створення Web-сайтів: БХВ-Петербург, - 2005р., - 948 с.
2. Джордж Шлоснейгл. Професійне програмування на PHP: Вільямс, 2006 р., - 624 с.
3. Люк Веллинг, Лора Томсон. Розробка Web-додатків за допомогою PHP і MySQL: Вільямс, 2007р., - 880 с.
4. Зольників Д. З. PHP 5: НТ Пресс, 2007 р., 256 с.
5. Ларри Ульман. Основи програмування на PHP: ДМК прес, 2001 р., 288 с.
6. Баженов В. А., Венгерським П. С. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник. К.: Каравела, 2007. –640 с.
7. Глушков С.В., Сурядний А.С. Персональний комп'ютер. Харьков. ФОЛІО, - 2004 г., - 282 с.
8. Дибкова Л. М. "Інформатика та комп'ютерна техніка", К., "Академія", 2002р. - 325 с.
9. Економічна інформатика Практикум: Навчальний посібник/ С.Д. Мамченко, В.А. Одинець. - К.: Знання, 2008. - 710с.
10. Економічна інформатика: Навчальний посібник/ В.С. Григорків, Л.Л. Маханець, Р.Р. Білоскурський и др.; Чернівці: Книги-XXI, 2008. - 464с.
11. Інформатика та комп'ютерна техніка: Навчальний посібник/ М.В. Макарова, Г.В. Карнаухова, С.В. Запара; Ред. М.В. Макарова. - Суми: Університетська книга, 2008. - 665с.
12. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: підручник/ В.А. Баженов, П.С. Венгерський, В.М. Горлач, І.М. Дудзяний. - К.: Каравела, 2008. - 639 с.
13. Олександр Качанов. Буквар по PHP і MySQL, 2000 р, 27 с.
14. Энди Харрис. PHP/MySQL для початківців: Кудиц-образ, 2005р., 284 с.
15. К Косентіно. PHP web професіоналам: Посібник. К.: Видавнича група BHV, 2001.

Зміст звіту:

1. Тема
2. Мета
3. Текст скрипту
4. Детальні пояснення до принципу його роботи

Контрольні запитання:

1. Завдяки чому в технології AJAX можна змінювати зміст WEB-сторінки без її перевантаження?
2. Для чого використовується модель DOM при роботі AJAX?
3. Яку роль відіграють теги <div> в AJAX?
4. Які додаткові функції мають теги <div> (знайдіть інформацію самостійно і занотуйте у звіт)?
5. Що представляє собою цей вираз : `document.getElementById('myform').innerHTML` ?
6. Поясніть принцип роботи WEB-сторінки `index.html`.
7. Яким чином JavaScript надсилає інформацію до php-скриптів?
8. Яким чином JavaScript одержує інформацію від php-скриптів?

Структура орієнтовного розподілу балів за модулями навчальної дисципліни, що присвоюється студентам

Вид контролю	Модуль	Тема	Навч. заняття (підготовка до виконання)	Виконання індивід. завдань (ОР, реферат, РІР, РР та ін.)	Модульний (змістово-модульний контроль)	Всього балів (сума 4+5+6)
1	2	3	4	5	6	7
Поточний контроль	I Використання PHP-технологій	Лекції: 1. Введення до курсу. Теорія реляційних баз даних. 2. Типи баз даних. Поля. Записи. Таблиці. 3. Пошук даних і індексні файли. 4. Цілісність даних. Нормалізація. 5. Сервери баз даних. Запити. Транзакції. 6. Модель управління SQL-сервером через технології http-протоколу. 7. Взаємодія web-форм із сервером баз даних MySQL. 8. Використання об'єктів web-форм для організації командного обміну з web-сервером.	8			8
		Лаб. роботи: 1. Дослідження операторів приєднання та аналіз бази даних на прикладі PHP-скрипту. 2. Створення бази даних. Створення таблиць. Заповнення таблиць. Створення запитів БД. 3. Технологія видалення записів із таблиць. 4. Технологія редагування записів у таблицях. 5. Реляційні бази даних (РБД). Дослідження основ роботи із записами у РБД з використанням web-технологій	5	4	5	14
		СРС: 1. Використання php-операторів для роботи з БД. 2. Технологія створення БД, таблиць, запитів. 3. Основи роботи із записами у РБД з використанням web-технологій		3		3
		Зах. мод. 1			10	10
Всього за модуль 1						35

Продовження таблиці

1	2	3	4	5	6	7
Поточний контроль	2 Робота з РБД	Лекції 1. Оператори та скрипти створення баз даних, таблиць та їх структур. Права доступу користувачів та захист інформації. 2. Оператори та скрипти створення та редагування записів. 3. Оператори та скрипти для вибірання інформації і її обробки. 4. Методи організації групової обробки даних. 5. Методи зберігання та обробки бінарних даних. - 4 години. 6. Використання Java Script у WEB-формах. 7. Ajax. Технологія створення інтерактивних інтерфейсів. Прості WEB-форми з AJAX. 8. Ajax. Взаємодія AJAX-форм із сервером баз даних.	8			8
		Лаб. роб. 1. Навігація у РБД. Дослідження основ створення довідникових структур в РБД. 2. Виконання групових операцій над записами 3. Оперування бінарними даними в СУБД з використанням web-форм. 4. Оперування бінарними даними в СУБД. Вивантаження та відкриття їх програмами. 5. WEB-інтерфейс з інтерактивними функціями. JavaScript. AJAX	5	4	5	14
		СРС 1. Структура та навігація РБД. 2. Робота з бінарними файлами. 3. JavaScript. AJAX та РНР-технології		3		3
		Зах. мод. 2			10	10
		Всього за модуль 2				
Всього за поточний контроль*						70
Підсумковий контроль (екзамен)						30
Разом						100

ВСТУП

Мета: надання знань з теоретичних засад і принципів побудови структур та моделей даних в сучасних програмних продуктах; набуття вмінь роботи з структурами та моделями даних за допомогою розробки власних прикладних програм на мові РНР; формування навиків налагодження програмних додатків з метою доступу до даних.

Завдання: вивчення теоретичних основ представлення інформації в пам'яті ЕОМ та характеристик прикладного програмного забезпечення по управлінню даними, формування уявлення у студентів про реляційну модель даних, застосування набутих знань, умінь та навиків для побудови власних програмних додатків маніпулювання даними.

Предмет: система засобів автоматизації для роботи з даними, а саме:

- засоби створення структур даних;
- засоби моделювання великих обсягів даних;
- засоби доступу до даних за допомогою сучасного програмного забезпечення.

Викладений матеріал методичних вказівок для виконання лабораторних робіт з дисципліни "Технологія створення програмних інтелектуальних систем" розрахований на студентів 3-го курсу факультету економіки та підприємництва.

У лабораторних роботах наведено завдання, короткі теоретичні відомості для закріплення знань студентів та пояснення практичних задач, що полегшує роботу при самостійному виконанні практичних завдань.

Оцінювання знань студентів здійснюється з використанням модульно-рейтингової системи оцінювання знань, при якій викладач оцінює знання і роботу студента у балах. Студент в свою чергу має ознайомитися з виконанням лабораторної роботи (кількість заходів –10), виконати завдання, дати відповіді на контрольні запитання, які наведені в кінці лабораторної роботи, оформити звіт в зошиті і захистити лабораторну роботу.

```
}  
?>
```

1. Розберіть, як працює скрипт **translate.php** і файл **ajax-php.html**. Зверніть увагу на те, яким чином скрипт повертає результати своєї діяльності. Очевидно, що словниковий запас такого перекладача закладений безпосередньо у текст програми самого скрипту! З цього напрошується винести словниковий запас у окрему таблицю-довідник і заставити скрипт `translate.php` шукати слова у таблиці. Тоді дані будуть відділені від програми. Як це зробити? Очевидно, що потрібно мати таблицю-словник (**ekXXdictionary** (XX — номер ftp-акаунту)), в якій відповіді буде шукати `php`-скрипт-перекладач. Структура таблиці може бути така:

```
id : integer  
english : varchar(255)  
ukrainian : varchar(255)
```

7. Створіть скрипт `create_db.php` для створення таблиці `ekXXdictionary`. Окрім створення структури, у скрипті для початку роботи зробіть декілька записів у таблицю `ekXXdictionary`. Додайте до вже існуючого скрипту `inc.php` у вашому кореневому каталозі рядок `$ekXXdictionary = ekXXdictionary;`

8. Самостійно створіть скрипт `mysqltrans.php`. У цьому скрипті методом GET мають одержуватись слова для перекладу і в таблиці `ekXXdictionary` у полі `english` шукатись слово і знайдений результат із поля `ukrainian` передаватись назад (яким чином?) до `ajax-php.html`.

9. Замініть у `ajax-php.html` `translate.php` на `mysqltrans.php` і перевірте роботу перекладача.

10. Створіть скрипт `edit.php` для редагування словника, використовуючи напрацювання попередніх лаб. робіт. Скрипт повинен мати функції додавання нових записів, редагування і видалення.

11. Додайте до форми в скрипті `edit.php` перевірку заповненості полів (валідатор). Для цього спочатку створіть допоміжну WEB-сторінку `1.html` з текстом нижче і дослідіть його роботу:

```
<html>
```

ЛАБОРАТОРНА РОБОТА №1

Тема: Дослідження операторів приєднання та аналіз бази даних на прикладі PHP-скрипту.

Мета: Набуття навичок створення простих серверних скриптів та аналізу баз даних і таблиць.

Теоретичні відомості

Для проведення досліджень будуть використані наступні SQL-функції, а також скриптова мова програмування PHP. Функції наведені в синтаксисі мови програмування PHP.

Функція `mysql_connect()`

Функція `mysql_connect` - Відкриває з'єднання з сервером MySQL.

Повертає покажчик на з'єднання з MySQL у разі успішного виконання, або FALSE при невдачі.

При її виклику, в якості параметра повинні бути вказані місцезнаходження сервера, реєстраційне ім'я і пароль користувача MySQL. Значення, яке повертає `mysql_connect()`, зазвичай привласнюється змінній, яка потім використовується функцією `mysql_select_db()`

```
$server_connection = mysql_connect ("localhost",  
"username", "password ");
```

Localhost - назва сервера

Username - назва користувача власника процесу сервера

Password - пароль

З'єднання з сервером буде закрито при завершенні виконання скрипту, якщо до цього воно не буде закрито за допомогою функції `mysql_close()`

Функція `mysql_select_db()`

Функція `mysql_select_db()` - вибирає базу даних MySQL `mysql_select_db("news", "$server_connection")`.

Повертає TRUE у разі успішного завершення, FALSE у разі виникнення помилки.

`mysql_select_db()` вибирає для роботи вказану базу даних на сервері, на який посилається переданий покажчик.

Якщо параметр покажчика опущений, використовується останнє відкрите з'єднання. Якщо немає жодного відкритого з'єднання, функція спробує з'єднатися з сервером аналогічно функції `mysql_connect()`, викликаною без параметрів.

Функція `mysql_query()`

`mysql_query()` надсилає запит до активної бази даних сервера.

Тільки для запитів **SELECT**, **SHOW**, **EXPLAIN**, **DESCRIBE**, `mysql_query()` повертає покажчик на результат запиту, або `FALSE` якщо запит не був виконаний. У решті випадків, `mysql_query()` повертає `TRUE` у разі успішного запиту і `FALSE` у разі помилки. Значення не рівне `FALSE` говорить про те, що запит було виконано успішно.

```
$result = mysql_query ("SELECT * FROM news", $db);
```

Функція `mysql_fetch_array`

Функція `mysql_fetch_array` приймає перший рядок результату виконання `MYSQL` і поміщає його в масив. При кожному виклику цієї функції повертається наступний рядок даних з результатів запиту до тих пір, поки не будуть оброблені всі рядки. На масив можна посилатися по номеру або по імені стовпця:

```
$myrow = mysql_fetch_array($result);
```

Запит **SHOW DATABASES** призначений для одержання інформації про доступні користувачеві бази даних:

```
$result = mysql_query("SHOW DATABASES", $db);
```

Для того, щоб почати роботу з таблицями, необхідно повідомити `MySQL` з якою базою даних ви маєте намір працювати. Це здійснюється за допомогою команди `USE`:

```
mysql_select_db( 'my_db', $db );
```

Тут `my_db` - назва вибраної бази даних.

Запит **DESCRIBE** дозволяє одержати інформацію про структуру вказаної таблиці і має наступний синтаксис:

```
$x = mysql_query("DESCRIBE table_name", $db);
```

Тут `table_name` - назва таблиці, структуру якої треба одержати. Наприклад, може бути одержана структура

Field	Type	Null	Key	Default	Extra
news_id	int(11)		PR	0	auto_increment

```

}
http.send(null);
}

</script>
<h1>Перекладач. Введіть слово англійською:</h1>
<form>
<input type="text" onkeyup="validate(this.value)" />
<div id="myform"></div>
</form>

</body>
</html>

```

5. Як слідує із тексту цієї програми, вона в процесі своєї роботи використовує `php`-скрипт, який очевидно і має перекладати слова. Створіть файл `translate.php` в цьому ж каталозі:

```

<?php
$x = $_GET['word'];
$x = strtolower($x); // змінимо регістр букв на нижній

switch($x) {
case 'cat' :
print "кішка";
break;
case 'dog' :
print "собака";
break;
case 'rat' :
print "пацюк";
break;
case 'cow' :
print "корова";
break;
case 'hen' :
print "курка";
break;
case 'goat' :
print "коза";
break;
default: print "?";
}

```



```

<script type="text/javascript">
function replace() {
document.getElementById('abc').innerHTML = "Це, я
<b>AJAX</b>!";
}
</script>
</head>
<body>
<div style="background-color:#00FF00"
id="abc">Привіт!</div>
<p><a href="javascript:replace()">Хто ти?</a></p>
</body>
</html>

```

2. Відкрийте у WEB-браузері файл *index.html*.
Поспостерігайте, як він працює. Уважно розберіть його код.

3. Створіть в каталозі **lr10** файл **ajax-php.html** для дослідження роботи AJAX і і php. Цей файл буде виконувати функцію простого інтерфейсу для перекладу слів з англійської мови на українську:

```

<html>
<head>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=UTF-8">
<script type="text/javascript">
var http = false;

if (navigator.appName == "Microsoft Internet Explorer") {
http = new XMLHttpRequest("Microsoft.XMLHTTP");
} else {
http = new XMLHttpRequest();
}

function validate(user) {
http.abort();
http.open("GET", "translate.php?word=" + user, true);
http.onreadystatechange=function() {
if(http.readyState == 4) {
document.getElementById('myform').innerHTML =
http.responseText;
}
}
}

```

Heading	varchar(48)	YES		NULL	
Body	Text	YES		NULL	
Date	Date	YES		NULL	
author_name	varchar(48)	YES		NULL	
author_email	varchar(48)	YES		NULL	

Короткий опис елементів таблиці:

- Кожна стрічка в графі Field повідомляє про назву поля (стовпця таблиці), в якому містяться дані.
- В графі Type розміщена інформація про те, які дані і якого типу можуть зберігатися в кожному конкретному полі таблиці.
- Якщо в графі NULL відповідного поля стоїть слово YES, то значення цього поля може бути рівним NULL.
- Наявність напроти конкретного поля в графі Key позначення PRI (скорочення від слова primary – первинний ключ) вказує на те, що дане поле є індексним. У стовпці первинного ключа не можуть знаходитися два однакових значення.
- Графа Default вказує на те, які значення вводяться в це поле за замовчуванням.
- І, нарешті Extra, описує будь-які додаткові атрибути, які може мати відповідна графа таблиці. Наприклад, атрибут auto_increment, призначений первинному ключу, вказує, що значення збільшується на 1 кожного разу при додаванні нового рядка до таблиці.

PHP-функція include()

Функція **include(filename);**

Функція include() виконує саме те, про що повідомляє її назва: тобто включає в текст скрипту інший файл. Вона вельми зручна, оскільки дозволяє використовувати заздалегідь підготовлені фрагменти коду, які можуть бути вставлені в різні сторінки, але зберігаються в одному місці. Синтаксис функції:

include("file.inc");

Якщо ви помістите вище приведений рядок до одного з своїх скриптів, то зможете застосувати будь-які функції, визначені у файлі file.inc. В дійсності можна використовувати не

тільки функції, але й будь-яку інформацію, що міститься у вказаному файлі. Ви можете вставити файл майже будь-якого типу, наприклад, що містить текст, код HTML або код PHP. Файли, що включаються в сценарій, як правило, мають розширення **.inc**, що спрощує їх пошук.

Завдання

1. Користуючись власним ftp-обліковим записом, створити на сервері PHP-скрипт згідно інструкцій, що наведені нижче, для дослідження функцій MySQL.

2. Провести дослідження основних функцій створення, перегляду та зміни баз даних та структури таблиць з використанням LAMP (Linux-Apache-Mysql-PHP) технології.

Хід виконання роботи:

1. Завантажити операційну систему Linux.
2. Запустити файловий менеджер (ФМ) Krusader і приєднатись по ftp до сервера, використовуючи логін і пароль свого власного ftp-облікового запису. В з'єднанні використати сервер **192.168.1.5** або **vsau.vin.ua** (Нагадаємо, що для створення ftp-з'єднання в ФМ Krusader і треба або натиснути Ctrl_N або ж скористатись його меню).

3. Після успішного з'єднання із сервером, користуючись меню ФМ, створіть новий файл на ftp-панелі з назвою **lr1.php**. Одразу після цього має відкритись текстовий редактор. Збережіть цей порожній текст (Ctrl_S) і далі внесіть до нього наступний PHP-код:

```
<?php  
include("../inc.php");
```

```
$db = mysql_connect($cServname, $cUsername, $cPassword) or  
die ('Не можу приєднатись до MySQL-серверу.');
```

print "Переглянемо доступні нам бази даних. (SHOW DATABASES)<hr>";

```
$x = mysql_query("SHOW DATABASES",$db) or die  
(mysql_error());
```

ЛАБОРАТОРНА РОБОТА №10

Тема: WEB-інтерфейс з інтерактивними функціями. JavaScript. AJAX.

Мета: Набуття вмінь та навичок у створення скриптів з використанням технології AJAX.

Теоретичні відомості

AJAX (Asynchronous JavaScript and XML).

AJAX описує, як JavaScript може бути використаний для отримання даних з сервера, використовуючи XML HTTP запит об'єкта, а потім вставити ці дані в сайт за допомогою DOM (*Document Object Model*). Це відбувається асинхронно - тобто, у фоновому режимі, без необхідності перезавантаження всієї сторінки.

Використовуючи JavaScript, DOM надає можливість програмно вставляти, видаляти і редагувати теги на веб-сторінці. Головна ідея DOM полягає у представленні WEB-сторінки як набору окремих об'єктів (полів, тегів, форм тощо), з кожним з яких може працювати JavaScript.

Завдання

Дослідити роботу найпростішого AJAX-скрипта.

Розробити і створити скрипт для пошуку записів у БД, використовуючи технологію AJAX. Робота базується на скриптах і даних, що були створені в лаб. роботі №9.

Хід виконання роботи

1. Створіть на своєму ftp-акаунті каталог **lr10**.
2. Створіть в каталозі **lr10** скрипт **index.html** для дослідження роботи AJAX і DOM:

```
<html>  
<head>  
<META HTTP-EQUIV="Content-Type" CONTENT="text/html";  
charset=UTF-8">
```

7. Спробуйте замість зображення завантажити файл іншого типу, наприклад *.doc або *.pdf. Перевірте, яка відбувається реакція при кліку на комірки таблиці, де мають розташовуватись зображення. Висновки занотуйте.

Зміст звіту:

- 1. Тема
- 2. Мета
- 3. Текст скрипту
- 4. Детальні пояснення до принципу його роботи

Контрольні запитання:

- 1. Який принцип роботи скрипту, що витягає бінарні дані із БД?
- 2. Який оператор в скрипті getfile.php генерує зображення, документ тощо?

```
while($a = mysql_fetch_row($x)) {  
print "$a[0]<br>";  
}
```

```
// приєднуємось до бази даних $cDatabase (у вашому випадку це БД "student")  
mysql_select_db( $cDatabase, $db ) or die ("Не могу активізувати базу даних");
```

```
// і з'ясуємо, що там є ...  
print "<br>Переглянемо список наявних таблиць в базі даних $cDatabase. (Show Tables)<br>";  
$x = mysql_query("SHOW TABLES", $db) or die (mysql_error());  
while($a = mysql_fetch_row($x)) {  
print "$a[0]<br>";  
}
```

```
// а тепер подивимось, яку структуру має таблиця  
$t='news'; // надамо змінній $t назву однієї з наявних в базі ба-  
них таблиць  
print "<br>Переглянемо структуру таблиці \"$t\" (DESCRIBE TABLE)<br>";  
print "<table border='1'>  
<th>Назва поля</th>  
<th>Тип поля</th>  
<th>Нуль</th>  
<th>Первинний ключ</th>  
<th></th>  
<th>Спосіб зміни ключа таблиці</th>  
<tr>  
<td>&nbsp;$a[0]</td>  
<td>&nbsp;$a[1]</td>  
<td>&nbsp;$a[2]</td>  
<td>&nbsp;$a[3]</td>  
<td>&nbsp;$a[4]</td>  
<td>&nbsp;$a[5]</td></tr>";  
};  
$x = mysql_query("DESCRIBE $t", $db) or die (mysql_error());  
while($a = mysql_fetch_row($x)) {  
print "<tr>  
<td>&nbsp;$a[0]</td>  
<td>&nbsp;$a[1]</td>  
<td>&nbsp;$a[2]</td>  
<td>&nbsp;$a[3]</td>  
<td>&nbsp;$a[4]</td>  
<td>&nbsp;$a[5]</td></tr>";  
}
```

```
print "</table>";
// закриємо з'єднання із сервером
mysql_close($db);
?>
```

Пояснення до роботи скрипту:

В цьому скрипті після з'єднання із сервером в змінній `$db` буде знаходитись код з'єднання. За одержаним кодом ми надалі будемо спілкуватись із MySQL-сервером

Функція `mysql_query` призначена для надсилання команд MySQL-серверу. Після її виконання маємо результат, який надається змінній (*наприклад* `$x`). У більшості випадків результатом є масив стрічок для того що би їх виділити використовується функція `mysql_fetch_row`, яка також дає масив стрічок. Але їх вже можна використовувати для роботи.

На операторі `while` створений цикл, який повторює дію стільки разів, скільки в масиві (*в нашому прикладі це* `$a`) буде стрічок. Зверніть увагу, що в його круглих дужках `while($a = mysql_fetch_row($x))` не тільки працює функція `mysql_fetch_row`, але й змінні `$a` надається значення і на останок оператор `while` перевіряє, чи не порожній масив повернула функція `mysql_fetch_row`! До речі, в процесі кожного `while`-повторення запускається функція `mysql_fetch_row`, яка отримує **наступний** елемент масиву `$x`.

До кожного одержаного елемента стрічкового масиву треба звертатись за індексом, наприклад `$a[1]` або `$a[0]`.

Функція `die` використовується для формування повідомлень, якщо з тих чи інших причин не спрацює оператор запиту до MySQL-серверу.

Після успішного набору скрипта і його збереження, перевірте його роботу web-браузером. Для цього відкрийте такий url: <http://vsau.vin.ua/students/labor/lamp/XXX/lr1.php>, де XXX це назва вашого ftp-облікового запису, наприклад, **ek1**. В разі відсутності помилок в скрипті, ви побачите WEB-сторінку, яку сгенерував скрипт.

```
} else {
// А ці 2 поля дозволяють кеширувати переданий файл.
// Дозволити кеширування
header("Cache-Control: public");
// Передати MIME-тип файлу
// Клієнт має знати, що саме йому передають.
header("Content-type: ".$b[mimetype]);
// Виводимо тіло файлу
print $b[filebody];
}
mysql_close($db);
?>
```

2. *Уважно перевірте скрипт `inc_tab.php`, щоб в ньому не було порожніх рядків ні на початку ні в середині, ні після останнього оператора. Наявність них призведе до блокування роботи скрипту `getfile.php` !*

3. Перевірити роботу гіперпосилання на зображення в базі даних. Для цього в таблиці телефонних абонентів скрипту `edit_phone.php` до дати колонку із словом-гіперпосиланням “Фото”. В гіперпосилання вказати скрипт `getfile.php` і передати йому в командному рядку значення поля `phone_id` із таблиці `$streets_photo`. В разі правильного виконання завдання при переході по гіперпосиланню має завантажитись сторінка із фотографією.

4. Замінити код цієї комірки на код зображення, де джерелом зображення виступає той самий скрипт `getfile.php`. Замість “...” використати відповідну змінну або елемент масиву із власного скрипту.

```
<td><img src='getfile.php?phone_id=...' Alt='Фото' border=1 width=60></td>
```

5. Переконайтесь, що в таблиці абонентів в полі “Фото” з'явилися відповідні фотографії.

6. Додайте до тегів зображень фотографій ... гіперпосилання на той самий скрипт із тим самим командним рядком. Після цього при кліку на фотографії буде завантажуватись це зображення у повний розмір файлу.

ЛАБОРАТОРНА РОБОТА № 9

Тема: Оперування бінарними даними в СУБД. Вивантаження та відкриття їх програмами.

Мета: Набуття вмінь та навичок у створення скриптів для одержання, переглядання і вивантаження бінарних даних із СУБД.

Завдання

Розробити і створити скрипт вивантаження і переглядання бінарних даних з бази даних. Робота базується на скриптах і даних, що були створені в лаб. роботі №8.

Хід виконання роботи

1. Скопіювати каталог на своєму ftp-акаунті lr8 разом із скриптами у ньому у каталог **lr9**.
2. Створити в каталозі **lr9** скрипт **getfile.php** для одержання бінарних даних із таблиці **\$phones_photo** за переданим методом **GET id** запису.

```
<?php
include('../inc.php');//підключаємо файл з змінними про
підключення
include('../inc_tab.php');// підключаємо файл з змінним про
таблиці
$db = mysql_connect($cServname,$cUsername,$cPassword)
or die('Не можу приєднатись
до MySQL-сервера. Перевірте чи не порожні значення
змінних.');
```

```
mysql_select_db( $cDatabase, $db ) or die ("Не можу
приєднатись до бази даних: $cDatabase.");

$phone_id = $_GET['phone_id'];

$a = mysql_query("SELECT * FROM $phones_photo WHERE
phone_id='$phone_id',$db) or die(mysql_error());
$b = mysql_fetch_array($a);
if ( !$b ) {
// Якщо такого файлу немає, тоді повертаємо до клієнта
код помилки:
header("HTTP/1.0 404 Not Found");
```

Контрольні запитання:

1. Для чого існує функція include(). Де і в яких сценаріях вона використовується?
2. Яка інформація міститься у файлі inc.php?
3. Яка інформація міститься у файлі inc_tab.php?
4. Яке призначення функції mysql_connect()?
5. Яке призначення функції mysql_select_db()?
6. Яке призначення функції mysql_select_db()?
7. Яке призначення функції mysql_query()?
8. Яке призначення функції mysql_fetch_array ()?
9. Як працює цикл в операторі while(...) і які при цьому виконуються функції?
10. Що таке асоціативний масив? Де в скрипті вони використовуються?
11. Що може виступати у якості індексу асоціативного масиву?
12. Як працює та частина скрипту, де створюється таблиця з призначеннями полів? Які HTML-теги створюють таблицю, її заголовок, рядки, колонки?
13. Чому в скрипті, де виводиться таблиця ніде не вказана кількість її рядків? Чим вона тоді визначається?

ЛАБОРАТОРНА РОБОТА № 2

Тема: Створення бази даних. Створення таблиць. Заповнення таблиць. Створення запитів БД.

Мета: Набуття навичок створення та редагування БД, таблиць.

Теоретичні відомості

Типи полів таблиць

Як відомо, бази даних складаються як правило з таблиць. Кожна з них має свою назву і структуру, тобто певний набір полів і їх назв. Кожне поле, як і змінні в програмуванні має свій тип. Типізація полів призначена для прискорення обробки інформації сервером. Адже знаючи тип даних в полі, наприклад цілі числа, можна одразу застосовувати до них цілочисельну математику, не витрачаючи час на визначення типу даних.

Список типів полів, що найчастіше зустрічаються, наведений в таблицях 2.1 – 2.3. Для багатьох з них надається максимальна розрядність або ширина поля, що вказується в дужках, яку ми далі позначатимемо символом як max. Наприклад, запис INT(2) означає, що значення даного поля не може перевищувати 99.

Таблиця 2.1 – Числові типи

Тип	Опис
TINYINT[(max)]	Дуже маленькі цілі числа діапазону -127...128.
SMALLINT[(max)]	Малі цілі числа діапазону -32768...32767.
MEDIUMINT[(max)]	Середні цілі числа.
INT[(max)]	Звичайні цілі числа.
FLOAT[(max,P)]	Числа з плаваючою точкою.
DOUBLE[(max,P)]	Числа з плаваючою точкою подвійної точності.
DECIMAL[(max,P)]	Числа з плаваючою крапкою, що приведені до типу char.

Контрольні запитання:

1. Що таке дані бінарного типу? Чи може рядок бути збереженим у вигляді бінарних даних?
2. Які типи полів в БД дозволяють зберігати бінарні дані?
3. Що необхідно додатково зберігати про бінарні дані? Чи є ця інформація обов'язковою?
4. Що треба зробити, щоби при відкриванні бінарних даних автоматично вибиралась програма для роботи із ними?
5. Чи треба вказувати розмір поля для збереження бінарних даних?
6. Який максимальний об'єм інформації можна зберегти у полі типу BINARY, BLOB, MEDIUMBLOB, LONGBLOB?
7. Які дії мають буди додані до скрипта, який одержує бінарні дані від web-браузера, щоби їх можна було записати до бази даних?
8. Для чого використовується функція addslashes?
9. Що треба зробити в скрипті при редагуванні полів про об'єкт, щоби випадково не стерти попередньо внесені до БД бінарні дані, якщо файл на web-формі не вибирався?
10. Яке поле на web-формі відповідає про передачу бінарних даних до сервера?

```

phone = '$phone',
pib = '$pib',
str_id = '$str_id',
apart = '$apart',
house = '$house',
mimetype = '$mimetype',
filebody = '$filebody'
WHERE phone_id = '$phone_id'
", $db);
} else {
mysql_query(" UPDATE $phones_photo
SET
phone = '$phone',
pib = '$pib',
str_id = '$str_id',
apart = '$apart',
house = '$house'
WHERE phone_id = '$phone_id'
", $db);
}

```

1. Перевірте працездатність скрипта на відсутність повідомлень про помилки при додаванні нових записів, при внесенні змін з фотографіями і без них.

2. Виведіть значення змінної mimetype і дослідіть, що передається при виборі файлів з розширенням *.jpg, *.doc, *.txt, *.html, *.pdf? Одержані дані занотуйте у звіт.

3. Що спільного і у чому різниця у полях типу BINARY, BLOB, MEDIUMBLOB, LONGBLOB. Дані знайдіть в інтернеті і занотуйте у звіт.

Зміст звіту:

1. Тема.
2. Мета.
3. Текст скрипту.
4. Детальні пояснення до принципу його роботи.
5. Результати дослідження.

До числових типів відносяться цілі числа і дійсні числа (числа з плаваючою крапкою). Для чисел з плаваючою крапкою, окрім максимальної ширини розрядності або ширини поля можна також вказувати число значущих цифр після коми, що в табл. 2.1 позначається символом P.

Стрічкові типи полів наведені в таблиці 2.2.

Таблиця 2.2 – Стрічкові типи

Тип	
CHAR (len)[BINARY]	Стрічки з довжиною len, яка не перевищує 255 символів. Ключове слово BINARY вказує на те, що дані повинні оброблятися незалежно від регістра.
CHAR	Синонім CHAR(1).
VARCHAR (len)[BINARY]	Синонім CHAR(len) за винятком того, що стрічки можуть бути довільної довжини.
TEXT	Стрічки з максимальною довжиною – 65535 байтів. Дані цього типу чутливі до регістру.
BLOB	Бінарні (двійкові) дані з об'ємом до 65535 байтів. Тип BLOB (binary large object - великий двійковий об'єкт) призначений для зберігання будь-яких двійкових даних, зокрема зображень і звукових послідовностей.

Різні типи для зберігання **дати і часу** наведені в таблиці 2.3.

Таблиця 2.3 – Типи дати і часу

Тип	Опис
DATE	Дата у форматі PPPP-ММ-ДД.
TIME	Час у форматі ЧЧ-ММ-СС.
DATETIME	Дата і час у форматі PPPP-ММ-ДД ЧЧ-ММ-СС.
YEAR	Рік у форматі PP або PPPP.
TIMESTAMP	Мітка часу для відліку за транзакціями у форматі PPPP-ММ-ДД ЧЧ-ММ-СС.

Створення таблиць

(команди нижче надаються в синтаксисі PHP)

CREATE TABLE table_name – створити таблицю із заданою назвою (тут **table_name** – назва створюваної таблиці). В цій команді також треба вказати структуру майбутньої таблиці, тобто назви полів та їх типи. Додатково можуть бути вказані ключові поля, тип кодування символів, а також коментарі. Наприклад, створимо таблицю news з полями news_id, heading, body, date, author_name, author_email і ключовим полем (PRIMARY KEY) news_id:

```
mysql_query("CREATE TABLE news (  
news_id INT NOT NULL auto_increment,  
heading VARCHAR(48),  
body TEXT,  
date DATE,  
author_name VARCHAR(48),  
author_email VARCHAR(48),  
PRIMARY KEY(news_id)", $db);
```

(Для зручності запису і читання цей рядок з командою записаний у декілька стрічок). Первинному ключу (PRIMARY KEY) можна наданий атрибут **auto_increment**, який буде в майбутньому *автоматично* генерувати унікальний ключ до кожного нового запису. Цей ключ представляє собою ціле неповторюване в інших записах цієї таблиці число. auto_increment – означає, що воно формується шляхом збільшення на одиницю значення останнього ключа. При вставці першого запису до таблиці значення ключа виставляється рівним нулю.

Зміна структури вже існуючих таблиць ALTER TABLE

Команда ALTER TABLE має три методи:

Додавання нового поля до таблиці (Add). Наприклад:

```
mysql_query("ALTER TABLE birthdays ADD age  
CHAR(3)", $db);
```

 Тут до структури таблиці буде додане ще одне поле **age** типу CHAR довжиною 3 байта.

Змінити поле (CHANGE)

Наприклад, встановимо в таблиці news довжину поля

```
// Конвертувати символи, що не розуміє MySQL  
$filebody = addslashes($filebody);  
}
```

12. Вставте у порожні поля оператора **INSERT** змінні **\$mimetype** і **\$filebody** :

```
.. mysql_query("INSERT INTO $phone_table VALUES (  
NULL,  
'$phone',  
'$pib',  
'$str_id',  
'$house',  
'$apart',  
'$mimetype',  
'$filebody'  
)", $db) ..
```

Відтепер оператор **INSERT** буде додавати до таблиці не тільки стрічкові дані, але й бінарні (файл із фотографією абонента), що містяться у змінній \$filebody.

14. Для повноцінної роботи оператора UPDATE його також можна змінити аналогічно до попереднього:

```
...  
mimetype = '$mimetype',  
filebody = '$filebody'  
WHERE ...
```

15. Але ж треба зробити так, щоби при редагуванні даних абонента у випадку, якщо файл із фотографією не вибирається на формі, то раніше внесені бінарні дані в таблиці зберігались. Як це зробити? Дуже просто! Перед **UPDATE** Будемо аналізувати будь-яку змінну від файлу із фотографією на її “непорожність”. Якщо вона буде непорожньою, то в оператором UPDATE вносимо зміни у поля **mimetype** і **filebody**. А у протилежному випадку, виконаємо інший варіант UPDATE але без цих полів:

```
...  
if ($filebody != "")  
{  
mysql_query(" UPDATE $phones_photo  
SET
```



```
''  
)", $db) ) ...
```

7. Переконайтесь в працездатності скрипта.
8. Додайте до заголовочної частини форми такі зміни:

```
<form action='$PHP_SELF' method='post'  
ENCTYPE=multipart/form-data name='myform'>
```

9. Через те, що цей скрипт тепер буде працювати, використовуючи методи **POST** (надсилання даних із форми) і **GET** — надсилання даних із командного рядку, то необхідно забезпечити роботу цих обидвох методів одночасно. Для цього “склеїмо” два рядки з різними методами разом, наприклад, `$str_id = $_POST['str_id'].$_GET['str_id'];` Дублювання інформації у нас не відбудеться! Адже дані надсилаються або методом **GET** (тоді всі `$_POST[...]` порожні), або методом **POST** (тоді, навпаки, всі `$_GET[...]` порожні). Всі подібні рядки перенесіть на початок скрипту і розташуйте одразу після операторів `include`. Запишіть ці оператори за зразком:

```
$phone_id = $_POST['phone_id'].$_GET['phone_id'];  
$phone = $_POST['phone'].$_GET['phone'];
```

Переконайтесь у працездатності скрипту.

10. Тепер додайте до вашої форми поле для вибору файлу із фотографією, наприклад :

```
Файл з фото: <input type='file' name='myfile'>
```

11. На початку скрипта після операторів типу `$phone_id = $_POST["phone_id"];` (див. Вище) розташуйте оператори для одержання інформації про файл із фотографією:

```
$filesize = $HTTP_POST_FILES[myfile][filesize];  
$mimetype = $HTTP_POST_FILES[myfile][type];  
$tmp_name = $HTTP_POST_FILES[myfile][tmp_name];  
if ( $file = @fopen( $tmp_name, "rb" ) ) {  
// Прочитати вміст тимчасового файлу  
$filebody = fread($file, $filesize);  
fclose( $file );  
// Видалити тимчасовий файл - сам сервер може про це забути  
unlink( $tmp_name );
```

author_name у 15 байтів:

```
mysql_query("ALTER TABLE news CHANGE author_name  
author_name VARCHAR(15)", $db);
```

Або ж змінимо назву поля body в таблиці news на нову - description:

```
mysql_query("ALTER TABLE news CHANGE body  
description");
```

Видалити поле (DROP)

В прикладі показано видалення поля date в таблиці news.

```
mysql_query("ALTER TABLE news DROP date", $db);
```

Змінити назву таблиці (RENAME TABLE)

Наприклад,

```
mysql_query("RENAME TABLE news TO story ", $db);
```

Видалити таблицю DROP TABLE

```
mysql_query("DROP TABLE table_name", $db);
```

Видаляється таблиця із вказаною назвою, наприклад table_name. Будьте обережні! Відновлення видаленої таблиці неможливе!

Робота із записами в таблицях

Додавання нового запису (INSERT)

В цій команді треба вказати назву таблиці (наприклад news), до якої потрібно вставити величини, а також вказати значення BCIX (це обов'язково) полів цього запису. В разі, якщо значення немає, то треба передати порожнє значення, наприклад " ". Значення полів передаються оператором **VALUES**. В його круглих дужках через кому перераховуються значення цього запису. Кількість значень в операторі **VALUES** обов'язково має збігатись із кількістю полів в таблиці. Вставимо, наприклад порожній запис в таблицю news. Там 6 полів. Тоді маємо виконати :

```
mysql_query("INSERT INTO news VALUES(",",",",",",",",  
$db ");
```

Цікаво ось що. Після додавання нового запису в першому полі (насправді їх нумерація йде з 0) буде не порожній рядок, а число 0! Чому? Вище, коли ми створювали структуру таблиці

news це поле було вказано як **news_id** INT NOT NULL auto_increment. Сам auto_increment при додаванні нових записів до цього поля буде самостійно буде вносити ключове число.

А тепер додамо до цієї ж таблиці ще один запис з даними (для зручності читання запишемо оператор декількома рядками):

```
mysql_query("INSERT INTO news VALUES(
',
'Назва',
'Текст новин',
'11-16-2006',
'Загоруйко В.',
'sergiy@ukr.net'
), $db ");
```

Замість конкретних значень, можна підставляти і змінні, наприклад:

```
mysql_query("INSERT INTO news VALUES(
',
'$a',
'Текст новин',
'$dt',
'Загоруйко В.',
'$qwerty'
), $db ");
```

Зміна запису (ів) (UPDATE)

Зрозуміло, що, якщо уявити собі таблицю із записами, представити, що в ній треба щось змінити, то виникають такі питання:

1. В якому записі зробити зміни?
2. В якому полі зробити зміни?
3. Які дані треба занести?

Відбір записів SELECT

Це – мабуть найголовніший оператор, який використовується при роботі з базами даних. Головна його функція – це вибірка даних з таблиць за вказаними умовами, наприклад:

```
$x=mysql_query("SELECT field_name FROM table_name",
$db);
```

```
mysql_query("INSERT INTO $streets_photo
VALUES(NULL,'Гарпіна'),$db);
mysql_query("INSERT INTO $streets_photo
VALUES(NULL,'Садова'),$db);
mysql_query("INSERT INTO $streets_photo
VALUES(NULL,'Стуса'),$db);
mysql_query("INSERT INTO $streets_photo
VALUES(NULL,'Квітнева'),$db);
```

```
mysql_close($db); // закриваємо з'єднання
?>
```

Зверніть увагу на те, що до таблиці з абонентами ми додали два поля для зберігання фотографій (**mimetype** і **filebody**).

5. Скопіюйте у каталог **lr8** скрипт для переглядання, додавання і редагування записів з телефонного довідника з лаб. роботи №5. Змініть назву цього скрипту на **edit_phone.php**.

6. В скрипті **edit_phone.php** уважно у всіх місцях замініть назви змінних **Sphones** на **\$phones_photo** і **\$streets** на **\$streets_photo**.

7. В частині скрипту, де додаються записи, тимчасово додайте два порожніх поля для підтримки формату. Наприклад, якщо раніше було:

```
... mysql_query("INSERT INTO $phones_photo VALUES (
NULL,
'$phone',
'$pib',
'$str_id',
'$house',
'$apart'
) ", $db) ) ...
```

то внесіть такі зміни :

```
... mysql_query("INSERT INTO $phones_photo VALUES (
NULL,
'$phone',
'$pib',
'$str_id',
'$house',
'$apart',
',
',
```

```

$db =
mysql_connect($cServname,$cUsername,$cPassword)//
під'єднуємось до mysql- сервера
    or die('Unable to connect to MySQL server.');//або вивести
повідомлення про не можливість приєднання

    mysql_select_db($cDatabase,$db) or die ("Could not activate
the database");
// створюємо таблицю телефонних абонентів
echo "<h2>Створюємо таблицю : $phones_photo</h2>";

mysql_query("DROP TABLE $phones_photo", $db);
mysql_query("CREATE TABLE $phones_photo (
phone_id int not null auto_increment,
phone VARCHAR(30),
pib VARCHAR(250),
str_id int (10),
house VARCHAR(10),
apart VARCHAR(10),
mimetype VARCHAR(200),
filebody BLOB,
PRIMARY KEY(phone_id))",
$db) or die(mysql_error());//якщо запит не виконано-
виводимо зміст помилки

echo "<h4>Таблиця \"\$phones_photo\" була успішно ство-
рена </h4>";
echo "<h2>Створюємо таблицю : $streets_photo</h2>";
mysql_query("DROP TABLE $streets_photo", $db);
mysql_query("CREATE TABLE $streets_photo (
str_id int not null auto_increment,
street varchar(250),
PRIMARY KEY(str_id))",
$db) or die(mysql_error());

echo "<h4>Таблиця \"\$streets_photo\" була успішно створе-
на </h4>";

// додамо делька вулиць для початку роботи
mysql_query("INSERT INTO $streets_photo
VALUES(NULL,'Сонячна')",$db);
mysql_query("INSERT INTO $streets_photo
VALUES(NULL,'Соборна')",$db);

```

Запис означає “відібрати з таблиці table_name всі дані, що знаходяться в полі field_name”. Чому всі? Тому що не вказано умови відбору (оператор WHILE). Теж саме, але з умовою і відбір будемо робити за двома полями:

```

$x = mysql_query("SELECT news
body,
author_name
WHERE      author_email      ='my@asd.com'      AND
heading='Рослини' ",
$db);

```

Звучить так: “З таблиці news відібрати інформацію з полів body і author_name з тих записів, де в полях **author_email** знаходиться стрічка 'my@asd.com' і в **heading** стрічка 'Рослини'.

А можна відбирати одразу з усіх полів і їх назви не вказувати? Звичайно, тільки замість назв треба розмістити символ * :

```

$x = mysql_query("SELECT *
WHERE      author_email      ='my@asd.com'      AND
heading='Рослини' ",
$db);

```

Залишилось лише розібратись, що ж потрапляє до змінної \$x? Відповідь буде така: “Двовимірний масив стрічок” або просто таблиця. Тобто при відбиранні даних у вказану змінну потрапляє таблиця, в якій буде стільки полів, скільки було вказано у запиті (в перших двох випадках їх було 2 (body і author_name), а в останньому – 5). І ця таблиця буде мати стільки записів, скільки “відібрав” оператор **SELECT**.

Як же одержати кожне значення окремо?

Виділення окремих значень із масиву запиту **mysql_fetch_row**.

Оператор **mysql_fetch_row** дозволяє виділяти окремі рядки(записи) із двовимірного масиву. Якщо один раз виконати його, наприклад \$y=mysql_fetch_row(\$x); , то до змінної \$y потрапить перший рядок таблиці \$x. Повторне виконання цього ж оператору призведе до того, що до змінної \$y потрапить другий рядок таблиці \$x.

Добре, а як же тоді одержати значення кожного елементу

цього рядку, наприклад **body**? Дуже просто! З індексом цього елементу в масиві. Індекси нумеруються по порядку від 0 і далі. Тому, якщо ми зробимо запит, наприклад по 2-х полях, то таблиця \$x, буде мати лише 2 поля з індексами 0 і 1.

Тобто значення поля body будуть мати індекс 0. Виведемо перше знайдене на web-сторінку:

```
$x = mysql_query("SELECT news
body,
author_name
WHERE author_email      ='my@asd.com'      AND
heading='Послини' ",
$db);
$y=mysql_fetch_row($x);
print "$y[0]";
```

А якщо відібралось декілька записів? Як визначити їх кількість? Один із способів полягає у створенні циклу з повторенням оператора **mysql_fetch_row**. Повторення буде продовжуватись до тих пір, поки масив \$y не стане порожнім:

```
while($myr=mysql_fetch_row($x))
{
print "$y[0] $y[1]<br> ";
}
```

Завдання:

1. Користуючись власним ftp-обліковим записом, створити на сервері PHP-скрипт згідно інструкцій, що наведені нижче, для дослідження функцій MySQL.

2. Провести дослідження основних функцій створення, перегляду та зміни баз даних та структури таблиць з використанням LAMP (Linux-Apache-MySQL-PHP) технології.

Хід виконання роботи

1. Завантажити операційну систему Linux.
2. Запустити файловий менеджер (ФМ) Krusader і приєднатись по ftp до сервера, як описано в попередній лабораторній роботі.

3. Після успішного з'єднання із сервером, користуючись

Графічна програма зазвичай читає інформацію порціями мінімум по 3 байта (24 біта = 16 777 216 значень!), знаходить до їх значення відповідний відтінок кольору точки (піксела) й видає малює її на екрані.

Для зберігання файлів у базі даних як правило використовуються поля абстрактного типу (BLOB (block of binary) або LONGBLOB (long block of binary)). Перший з них дозволяє зберігати блоки довжиною у 65 538 байт, а другий — у 4 294 967 300 байт. Додатково окрім самого файлу потрібно запам'ятати також його назву, довжину і тип даних (mimetype). Останнє потрібно, щоб при відкритті таких даних автоматично знаходилась програма, що з ними працює, наприклад, якщо це *.html, то це браузер, якщо це *.doc, то це текстовий редактор і т.д. Ці додаткові дані як правило записуються у простих текстових полях типу, наприклад, varchar (див. далі).

Завдання

1. Розробити і створити скрипт для внесення і редагування бінарних даних (файлів із зображеннями) до таблиці телефонного довідника.

Хід виконання роботи

1. Створити на своєму ftp-акаунті каталог **lr8**.
2. Додати до скрипту inc_tab.php, що знаходиться рівнем вище від каталогу lr8 два рядки (XX — номер ftp-акаунту студента):

```
$phones_photo='ekXXphonesphoto';
$streets_photo='ekXXstreetsphoto';
```

1. Скопіювати до каталогу **lr8** скрипт для створення таблиць телефонного довідника з лабораторної роботи №5. Замінити в ньому назви змінних \$phones на \$phones_photo і \$streets на \$streets_photo.

2. Створити в каталозі **lr8** скрипт **create_db.php** для створення двох таблиць цього проекту з таким змістом:

```
<?php
include('../inc.php');//підключаємо файл з змінними про
підключення
include('../inc_tab.php');// підключаємо файл з змінним про
таблиці
include('../header.php');// підключаємо файл з заголовочною
частиною web-сторінки. тут також і кодування.
```

ЛАБОРАТОРНА РОБОТА № 8

Тема: Оперування бінарними даними в СУБД з використанням web-форм.

Мета: Набуття вміння та навичок у створення скриптів внесення і редагування бінарних даних в СУБД

Теоретичні відомості

Будь-який файл, за суттю, бінарний, тому що містить двійкові дані. Навіть текстові файли – й ті містять двійкові дані. Всі двійкові дані складаються з байтів.

Байт – це одиниця вимірювання інформації, яка складається з 8 біт. А що ж таке біт? Біт – це теж одиниця вимірювання інформації, але в неї можливі тільки два варіанти, які вона може приймати (1 або 0). Це обумовлено тим, що фізично біт це найменша ділянка диску, яка може бути намагніченою, розмагніченою (умовно). В одному випадку біт вважається рівним 1, а в іншому – 0. Третього не буває.

8 бітів і складають 1 байт. Це, так би мовити, комп'ютерне слово мінімальної довжини. Всього можливих комбінацій біт у байті – 256 (2^8). Отже числове значення байта може бути рівним від 0 до 255 включно.

Будь-яка інформація як правило записується на диск послідовністю (ланцюжком) байтів. Вона утворює файл. Файл також має і службову інформацію типу назви файлу, дати і часу його створення, довжини тощо. Головне у тому, що у файлі можна зберігати будь-яку інформацію - графічну, звукову, текстову, архівну тощо.

Інтерпретація байтів файлу визначається тільки програмою, якою він відкривається. Якщо це текстовий редактор, то він зчитує кожний байт (код символу) і за спеціальною таблицею знаходить відповідне його графічне зображення й виводить його на екран.

Якщо це звукова програма, то вона використовує послідовність байтів для управління електронікою звукової карти і ми чуємо відповідні до записаних у файл кодів звуку.

меню ФМ, створіть новий файл на ftp-панелі з назвою lg2.php.

Введення інформації в базу даних з використанням PHP

Для роботи з базою даних застосування PHP є менш трудомістким, гнучкішим і, що найважливіше, легше здійснюваним за допомогою Web-браузера.

У основі взаємодії PHP та бази даних покладено прості принципи:

- спочатку необхідно під'єднатись до серверу БД та зареєструватись;
- потім слід вибрати базу даних, яка стане використовуватися;
- і, нарешті, - відправляти запити SQL на сервер для додавання, видалення і зміни даних.

Вам навіть не треба пам'ятати про те, що в кінці роботи потрібно закрити з'єднання з сервером. PHP виконає це за вас.

Зараз ви повинні побудувати таблицю news за допомогою PHP.

Створюємо файл inc_tab.php, який теж приєднується до скрипту, за допомогою функції **include**:

```
<?
$news_table = 'news';
?>
```

Ви бачите, що тут створюється змінна \$news_table, яка запам'ятовує назву таблиці. **Кожен студент повинен дати свою індивідуальну назву, яка б більше не повторювалась. Наприклад, \$news_table = 'ek21news';**

Для зручності роботи в майбутньому створіть ще один скрипт **header.php** із тегами заголовку сторінок, які будуть генерувати наші скрипти:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
</head>
```

1. Створюємо скрипт **news.php**:

Наша таблиця Новин повинна мати такі поля:

- Порядковий номер - news_id;
- Заголовок новини – heading;
- Зміст новини – body;
- Дату створення – date;
- Ім'я автора - author_name;
- Email автора - author_email.

Типи цих полів, їх розмір та сама структура таблиці була розглянута [раніше](#). Також ви тут бачите застосування [функцій](#), розглянутих на початку лабораторної роботи.

```
<?php
include('../inc.php'); //підключаємо файл з інформацією про
підключення
include('inc_tab.php'); // підключаємо файл з інформацією
про таблиці
include('header.php'); // підключаємо файл з заголовочною
частиною web-сторінки
$db = mysql_connect($cServname,$cUsername,$cPassword)
// під'єднатись до MySQL сервера
or die('Unable to connect to MySQL server.');// або вивести
повідомлення про неможливість під'єднання
mysql_select_db( $cDatabase, $db ) or die ("Could not activate
the database");
// створюємо таблицю News, яка збирає замітки новин і має
такі поля
echo"<h4>Створюємо таблицю \"$news_table\" ...</h4>";
mysql_query("CREATE TABLE $news_table (
news_id INT NOT NULL auto_increment,
heading VARCHAR(48),
body TEXT,
dt VARCHAR(10),
author_name VARCHAR(48),
author_email VARCHAR(48),
PRIMARY KEY(news_id))",
$db) or die(mysql_error()); //якщо запит не виконано - виводимо
зміст помилки
echo"<h4>Таблиці успішно створені</h4>";
mysql_close($db);
?>
```

6. Перенести до каталогу Iг7 скрипт-редактор записів **edit.php** із лаб. роботи №5.

7. Використати одержані вище напрацювання із скрипту **del_phones.php** для модифікації скрипту **edit.php**. Спочатку ньому прибрати старе гіперпосилання у полі для видалення рядків, а також фрагмент коду, що видаляв записи за цим гіперпосиланням. На заміну додати поле із **checkbox**`ами, кнопку “Видалити позначені записи” і код обробки роботи цієї кнопки.

8. *Додатково.* Використовуючи ідеї, що викладені вище, створити скрипт **edit_all.php** для групового редагування прізвищ абонентів. Для цього у відповідних комірках таблиці вивести в прізвища абонентів не просто як текст (**<td>Сердюк А.Б.</td>...**), а скористатись об'єктами типу text (**<td><input type='text' name='y4' value='Сердюк А.Б.'></td>..**). В коді для внесення відредагованих значень використати оператор **update**

```
if (@$edit)
{
$a=mysql_query("SELECT * from $phones",$db);
while($b=mysql_fetch_row($a)){
$x=$HTTP_GET_VARS[y.$b[0]]; //одержуємо надіслане значення checkbox`а
mysql_query("UPDATE $phones SET
pib = $x
WHERE id=$b[0]",$db);
}
```

Зміст звіту:

1. Тема
2. Мета
3. Текст скрипту
4. Детальні пояснення до принципу його роботи

```

<td><input type='checkbox' name='x$b[0]' value='on'>
...
}
...
<tr><td colspan='3'><input type='submit' name='del'
value='Видалити позначені записи'></td></tr>
</table>
</form>";

```

4. Додати на початку скрипту **del_phones.php** код, який би видаляв позначені на у списку на формі записи. Для цього скористаємось властивістю позначення об'єктів **checkbox** на формі. Вона полягає у тому, що всі позначені **checkbox**'и на web-формі (з "пташками"), при надсиланні мають непорожнє значення, а не позначені — порожнє. Виходячи з цього частина скрипту, що має видаляти записи, повинна перевірити, чи змінна із назвою **checkbox**'у порожня чи ні. Якщо змінна є непорожньою, то такий запис видаляється із таблиці. А у протилежному випадку — з ним нічого не робиться. Як це зробити?

Звичайно, що нам буде потрібен цикл по всіх записах таблиці **phones**. В ньому треба зробити такі дії:

- а. Одержати ID поточного запису.
- б. Одержати назву змінної **checkbox**'у, що відповідає на цей запис.
- в. Перевірити, чи вона не порожня. Якщо так, то видалити запис із поточним ID.

А ось один із варіантів такого коду:

```

if (@$do)
{
$a=mysql_query("SELECT * from $phones",$db);
while($b=mysql_fetch_row($a)){
$x=$HTTP_GET_VARS[x.$b[0]]; //одержуємо надіслане значення checkbox`а
if(!empty($x))
{
mysql_query("DELETE FROM $phones WHERE
id=$b[0]",$db);
}
}
}

```

1. Створюємо скрипт **database.php**, який буде заносити новини у БД.

Цей скрипт демонструє, наскільки легко розмістити інформацію в БД, використовуючи PHP. Достатньо ввести лише декілька коротких стрічок. З тридцяти стрічок коду в цьому скрипті, лише біля десяти відносяться до PHP. У інших звичайна розмітка HTML. Зразки форм, які з'являються при виконанні скрипту, показані на рис. 2.1 та 2.2

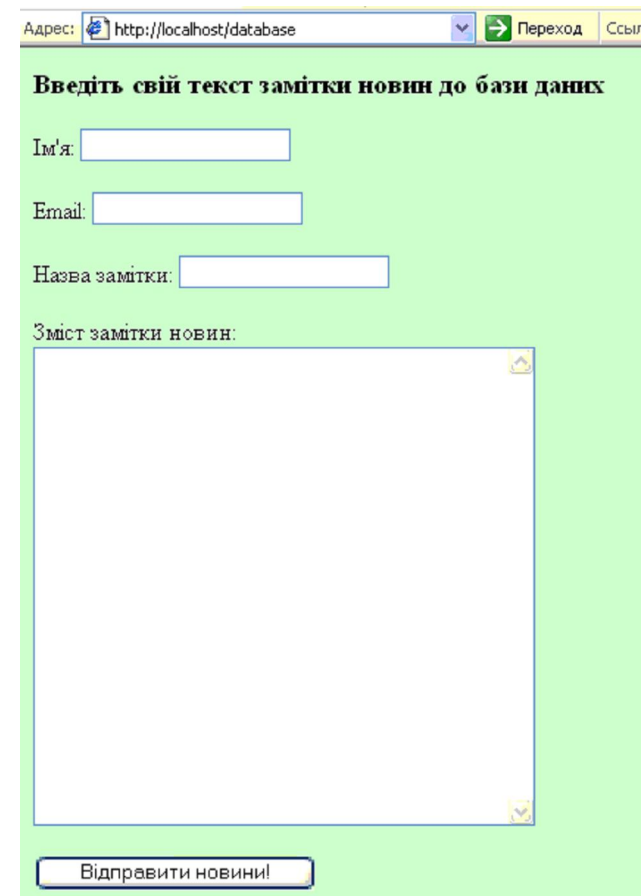


Рис. 2.1 Початкова форма введення інформації про новини в таблицю news з скрипту **database.php**

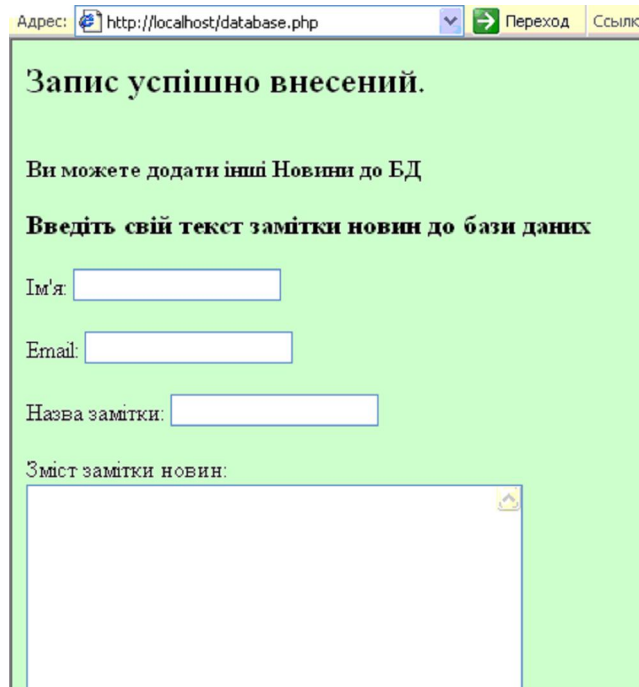


Рис. 2.2 Результат надсилання до сервера даних з форми при виконанні скрипту **database.php**

А це текст самого скрипту **database.php**:

```

1. <?php
2. include('../inc.php');
3. include('inc_tab.php');
4. include('header.php');
5. print"<body bgcolor='#C4FFC4'"
6. /* Ця програма дозволяє додавати Новини у базу да-
них */
7. if (isset ($submit)) { //якщо була натиснута кнопка
submit на формі
8. $db =
mysql_connect($cServname,$cUsername,$cPassword) or die ("Не
можу приєднатись до MySQL-сервера. Перевірте чи не порожні
значення змінних.");

```

ЛАБОРАТОРНА РОБОТА № 7

Тема: Виконання групових операцій над записами.

Мета: Набуття вмінь та навичок у створення скриптів для групового видалення і редагування записів

Завдання

Для дослідження використати просту двотабличну РБД телефонного довідника, що була розроблена у лабораторній роботі №5.

1. Розробити і створити скрипт групового видалення записів із таблиці **phones** абонентів телефонного довідника.

2. Розробити і створити скрипт групового редагування прізвищ абонентів у таблиці **phones** телефонного довідника.

Хід виконання роботи

1. Створити на своєму ftp-акаунті каталог **lr7**.

2. Створити в каталозі **lr7** скрипт **del_phones.php**, який би з таблиці (**\$phones**) виводив на web-сторінку список вулиць у вигляді таблиці з трьома полями “№ п/п” і “Назва вулиці” і “Видалити”.

3. Код виведення цієї таблиці розташувати на формі з методом передачі **GET** на саму себе (**action='\$PHP_SELF'**). В нижній частині форми розташувати кнопку типу **submit** з відписом “Видалити позначені записи”.

4. В полі “Видалити” розташувати об'єкти типу **checkbox**. У їх назвах закодувати id записів, до якого кожен з них відноситься. Припустимо, що назва кожного з них буде представляти якоюсь буквою, наприклад **x** і коду **id**. Тоді одержимо назви типу **x3, x12, x14...** Ось приклад одного із варіантів фрагменту коду:

```

$db=mysql_query("SELECT * FROM $phones ORDER BY
pib",$db);
$i=1;
while($b=mysql_fetch_row($a)){
...
print "<tr><td>$i</td>
...

```


7. Використати браузером скрипт **view_streets.php** і переконатись, що гіперпосилання відкривають скрипт **view_phones.php**, в якому виводиться список абонентів обраної вулиці.

8. Додати до запитів у скриптах команди для впорядкування записів за абеткою (**ORDER BY**) і переконатись у їх роботі.

9. *(Додатково). Об'єднати створені скрипти до одного.*

Зміст звіту:

1. Тема
2. Мета
3. Текст скрипту
4. Детальні пояснення до принципу його роботи

9. `mysql_select_db($cDatabase, $db) or die ("Не можу приєднатись до бази даних: $cDatabase.");`

10. `$date=date("Y-m-d");`

11. `if (mysql_query("INSERT INTO $news_table VALUES (`

`NULL,`

`'$heading',`

`'$body',`

`'$date',`

`'$auth',`

`'$auth_email'`

`)", $db)) {`

12. `echo "<h2>Запис успішно внесений.</h2>";`

13. `echo "
Ви можете додати інші Новини до`

`БД";`

14. `} else {`

15. `echo "
Трапилась помилка. Перевірте`

`скрипт.";`

16. `}`

17. `}`

18.

19. `?>`

20. `<p><h3>Введіть свій текст замітки новин до бази даних</h3></p>`

20. `<form action="database.php" method="post">`

21. `Ім'я: <input name="auth" type="text" ><p></p>`

22. `Email: <input name="auth_email" type="text"><p></p>`

23. `Назва замітки: <input name="heading" type="text"><p></p>`

24. `Зміст замітки новин:
`

25. `<textarea name="body" rows=20 cols=40></textarea><p></p>`

26. `<input type="submit" name="submit" value="Надіслати новини!">`

27. `</form>`

28. `</body>`

29. `</html>`

ЛАБОРАТОРНА РОБОТА № 6

Пояснення до скрипту database.php
В стрічці 2 під'єднуємо файл inc.php, в якому налаштована вся інформація для з'єднання з сервером БД.

В стрічці 3 під'єднуємо файл inc_tab.php, в якому зберігається назва таблиці.

В стрічці 4 під'єднуємо файл header.php, в якому зберігається назва таблиці.

В стрічці 7 перевіряється чи була натиснута кнопка Submit. Якщо так, то так, то виконуються рядки у фігурних дужках. Тобто додати до своєї таблиці новий запис.

В стрічці 8 результат виконання функції mysql_connect(...) надається змінній \$db. Надалі значення цієї змінної (“канал зв'язку”) використовується для діалогу скрипту із сервером баз даних MySQL.

В стрічці 9 функція mysql_select_db обирає для подальшої роботи базу даних, назва якої знаходиться у змінній \$cDatabase.

У стрічках 11-16 виконується запит, згідно якого до таблиці має бути доданий новий запис із величинами в полях, що визначаються змінними \$heading, \$body, \$date, \$auth, \$auth_email.

Починаючи із стрічки 19 за допомогою коду HTML створюється форма для введення інформації і надсилання її цьому ж самому скриптові.

Цей скрипт викликає сам себе (стрічка 26). Тому в одному сеансі може бути введено декілька заміток з новинами. Форма HTML виводиться на екран при кожному виконанні скрипту, та після чергового натискання користувачем кнопки Submit.

2. Самостійно додати засоби контролю помилок, описані у попередній роботі.

3. Додайте 8-10 записів до БД.

4. Для того, щоб переглянути зміст таблиці у БД, напишіть наступний скрипт **view_news.php**:

```
<?php  
include('../inc.php');  
include('inc_tab.php');  
include('header.php');
```

Тема: Навігація у РБД. Дослідження основ створення довідникових структур в РБД.

Мета: Набуття вмінь навичок у створенні списків з пов'язаних таблиць.

Завдання

Для дослідження використати просту двотабличну РБД телефонного довідника, що була розроблена у лабораторній роботі №5. Розробити і створити скрипт одержання списку вулиць (підпорядкована таблиця), гіперпосилання якого б дозволяли одержувати списки абонентів, що мешкають на обраній вулиці.

Хід виконання роботи

1. Користуючись скриптами з лабораторної роботи №5, наповнити таблиці телефонного довідника записами. Кількість вулиць має бути не менше 5 і кількість телефонних абонентів на кожній із них не менше 3.

2. Створити на своєму ftp-акаунті каталог **lr6**.

3. Створити в каталозі **lr6** скрипт **view_streets.php**, який би з таблиці (**\$phones**) виводив на web-сторінку список вулиць у вигляді таблиці з двома полями “№ п/п” і “Назва вулиці”.

4. Після успішного виконання п.3 додати до назв вулиць гіперпосилання на скрипт **view_phones.php**, передавши йому у командному рядку **ID** вулиці і її назву, наприклад ... **view_phones.php?id=\$b[0]&street=\$b[1]** .

5. Створити скрипт для виведення у вигляді таблиці списку **vcix** телефонних абонентів. У таблиці на цій web-сторінці має показуватись всі поля (**\$streets**) окрім поля **str_id**. Перевірити працездатність скрипту, відкривши його браузером.

6. Додати на початку скрипту **view_phones.php** цього балиціВ файлі inc_tab.php додати два рядки для одержання із командного рядку **ID** вулиці і її назву (див. п.4) у відповідні змінні. Вивести значення цих змінних у заголовку до таблиці, що генерує цей web-скрипт, наприклад, “Список телефонних абонентів, що проживають на вулиці <.назва вулиці.>”.

Контрольні запитання:

1. Що таке реляційна база даних (РБД)?
2. Які існують типи реляційних зв'язків. Наведіть приклади.
3. Які типи серверів підтримують цілісність реляційних зв'язків? Які мають переваги і недоліки такі сервери?
4. Які типи серверів не підтримують цілісність реляційних зв'язків? Які мають переваги і недоліки такі сервери?
5. Що таке індексні файли? Для чого вони використовуються? Наведіть приклад індексного файлу.
6. Чи існують випадки, коли можна відмовитись від індексних файлів?
7. В чому полягає принцип цілісності даних у БД?
8. В чому полягає принцип нормалізації даних? Наведіть власний приклад.

```
print"<body bgcolor='#C4FFC4'>"
/* Ця програма дозволяє переглядати зміст таблиці Новин
у базі даних */

$db = mysql_connect($cServname,$cUsername,$cPassword)
or die ("Не можу приєднатись до MySQL-сервера. Перевірте чи
не порожні значення змінних.");

mysql_select_db( $cDatabase, $db ) or die ("Не можу
приєднатись до бази даних: $cDatabase.");

$date=date("Y-m-d");

echo "<h2>Таблиця $news_table $date</h2>";

/* Тут запам'ятовуємо у змінну $rezult запит до серверу */
$rezult= mysql_query("SELECT * FROM $news_table ");

/* Змінна $items приймає перший рядок результату вико-
нання MYSQL і розміщує його в масиві рядків. Щоб доступитись
до елементів масиву використовуємо цикл while */
while ($items = mysql_fetch_row($rezult))

/* Виводимо результат у вигляді таблиці на екран, де у
кожну комірку прописуємо значення елемента масиву за його
номером */
{
echo"<table border=1> <tr align=center>
<td width=20>$items[0]</td>
<td width=60>$items[1]</td>
<td width=500>$items[2]</td>
<td width=100>$items[3]</td>
<td width=80>$items[4]</td>
<td width=80>$items[5]</td>
</tr></table>";
}
?>
</body>
</html>
```

Контрольні запитання:

1. Для чого існує функція include(). Де і в яких сценаріях вона використовується?
2. Яка інформація міститься у файлі inc.php?
3. Яка інформація міститься у файлі inc_tab.php?
4. Яке призначення функції mysql_connect()?
5. Яке призначення функції mysql_select_db()?
6. Яке призначення функції mysql_query()?
7. Яке призначення функції mysql_fetch_array()?
8. Які SQL-запити ви знаєте?
9. Наведіть приклад запиту SELECT у зроблених вами скриптів?
10. Наведіть приклад запиту DELETE у зроблених вами скриптів?
11. Наведіть приклад запиту INSERT у зроблених вами скриптів?
12. Для чого призначена команда UPDATE?
13. Для чого призначена команда DESCRIBE?
14. Які є типи полів бази даних?
15. Як створити базу даних?
16. Як показати всі БД?
17. Як вибрати потрібну БД?
18. Як створити таблицю?
19. Що таке первинний ключ?
20. Яким чином вводяться дані у таблицю?
21. Яке призначення скрипту database.php? Яка послідовність створення цього скрипту? Як він працює?
22. Яке призначення скрипту news.php? Як він працює?

```
<select name='str_id'>
<?php
$st = mysql_query("SELECT * FROM $streets ORDER BY
street", $db) or die();
while($s = mysql_fetch_row($st))
{
if ($str_id == $st[0]) $t = 'selected';
else $t = '';
print "<option value='$s[0]' $t>$s[1]</option>";
}
<option>
?>
</select>
```

1. Переконайтесь, що внесені зміни призвели до показу селектором назв редагованих вулиць. Уважно розберіть принцип роботи запропонованого коду. Пояснення занотуйте у звіт.

2. Залишилось зробити так, щоби у списку телефонних абонентів, що виводяться у таблиці замість коду вулиці виводилась її назва. Для цього в процесі побудови кожного рядку цієї таблиці зробити запит до таблиці **streets** і одержати назву вулиці. Цю назву треба вивести у полі, де перед цим показувався код вулиці. Для цього в існуючому циклі, де виводяться рядки таблиці з абонентами, додайте команди **...mysql_query("SELECT ...", \$db) or die();**

i ...mysql_fetch_row(...); самостійно. (назви змінних із масивами у цих командах має обов'язково відрізнятися від вже використаних у циклі!).

3. Переконайтесь, що виконані вами зміни у кодї дозволили одержувати в таблиці назви вулиць. Після успішного завершення завдання ви маєте просту реляційну базу даних із скриптовою зв'язкою записів у таблицях.

Зміст звіту:

1. Тема
2. Мета
3. Текст скрипту
4. Детальні пояснення до принципу його роботи

ЛАБОРАТОРНА РОБОТА № 3

Тема: Технологія видалення записів із таблиць.

Мета: Набуття навичок у створенні скриптів для видалення записів у таблицях БД.

Теоретичні відомості

Видалення записів DELETE

Команда DELETE просто видаляє запис (або записи), що відповідають вказаному критерію. Для видалення запису(ів) з таблиці треба вказати назву цієї таблиці і умову, які записи треба видалити, наприклад:

```
mysql_query("DELETE FROM news WHERE news_id > '2' , $db ");
```

У розглянутому випадку з таблиці видалений рядок, де значення **news_id** дорівнює 1. У таблиці можна також, наприклад, видалити рядки, в яких поле **author_name** містить значення «Загоруйко В.» і т.д.

```
mysql_query("DELETE FROM news WHERE author_name = 'Загоруйко В' , $db ");
```

Якщо не вказати ніякої умови, то в таблиці будуть видалені всі рядки:

```
mysql_query("DELETE FROM news , $db ");
```

Завдання

Маючи у своєму розпорядженні скрипти, що створені у лабораторній роботі №2 для внесення записів до таблиці, переглядання вже внесених записів, а також скрипт для видалення записів, створити один комбінований скрипт-редактор записів, який би дозволяв виконувати ці дії.

Хід виконання роботи

1. Створити каталог **lr3**.
2. Допоміжні скрипти **header.php**, **inc_tab.php** розташувати над каталогом **lr3** (в кореновому каталозі **ftp-акаунту**).
3. Скопіювати в каталог скрипт **database.php**.
4. Змінити назву **database.php** на **edit.php**.

дозволяв редагувати таблицю **phones**. Для цього замініть в ньому назву редагованої таблиці (використайте змінну **\$phones**) і приведіть назви і кількість полів до структури таблиці **phones**. Для поля таблиці **str_id** тимчасово для перевірки працездатності скрипту використайте на формі аналогічне поле типу **text**.

4. Переконайтесь, що за допомогою створеного скрипта можна виконувати додавання нових записів до таблиці **phones**, їх переглядання, видалення і редагування.

5. Поєднайте таблиці **streets** і **phones** реляційним зв'язком таким чином, щоби при додаванні або редагуванні записів замість кодів вулиць можна було використовувати їх назви. Для цього на формі замініть текстове поле для **str_id** на поле множинного вибору із списку, використовуючи для цього об'єкт **<select ...>**. Вставте такий код:

```
<select name='str_id'>
<?php
$st = mysql_query("SELECT * FROM $streets ORDER BY
street", $db) or die();
while($s = mysql_fetch_row($st)) print "option
value='$s[0]'>$s[1]</option>";
<option>
?>
</select>
```

6. Відкрийте скрипт браузером і переконайтесь, що на формі з'явився об'єкт із списком вулиць. Переконайтесь, що при виборі вулиці і натисканні кнопки додати або редагувати в таблиці **phones** відповідно змінюється код вулиці.

7. Доопрацюйте скрипт таки чином, щоби при редагуванні запису, в селекторі автоматично показувалась та назва вулиці, код якої вказаний у обраному записі з таблиці **phones**. (Адже зараз при редагуванні в селекторі завжди показується 1-я у списку вулиця!). Для цього використаємо властивість об'єкту **<option...>**, яка називається **selected**. Той об'єкт **<option...>**, який буде мати запис **selected**, буде автоматично показуватись таким, який обраний. Змінимо код наступним чином:

5. Змініть шляхи у командах **include** таким чином, щоби вони вказували на файли, що знаходяться рівнем вище:

```
include('../inc.php');
include('../inc_tab.php');
include('../header.php');
```

6. Запустіть скрипт і переконайтесь, що разом із формою виводиться таблиця із записами. Спробуйте додати новий запис. Після додавання він має з'явитись у цій таблиці.

7. Створити налаштування для видалення записів. Для цього зробимо так, щоби в одній з колонок таблиці існуючих записів виводилось не тільки значення, але й до нього було приєднано гіперпосилання для видалення цього запису із таблиці.

Що повине робити це гіперпосилання? Воно має викликати скрипт для видалення і передати в командному рядку код запису, який треба видалити. Наприклад, `del.php?id=5`. Цей скрипт одержить код і видалить запис.

Але спочатку створимо такий допоміжний скрипт **del.php**:

```
include('../inc.php');
include('../inc_tab.php');
include('../header.php');
print"<body bgcolor='#C4FFC4'>"
$id = $_GET['id']; // одержимо із командного рядку значення параметру id і надамо його змінній $id. Тут співпадіння назв зроблено тільки для спрощення читання
$db = mysql_connect($cServname, $cUsername, $cPassword) or die ("Не можу приєднатись до MySQL-сервера. Перевірте чи не порожні значення змінних.");
mysql_select_db( $cDatabase, $db ) or die ("Не можу приєднатись до бази даних: $cDatabase.");
if (mysql_query("DELETE FROM $news_table WHERE news_id='$id' ", $db) )
{
print "запис із номером $x із таблиці $news_table був видалений";
} else {
print "не можу знайти запис із номером $x в таблиці $news_table";
}
?>
```

Спробуйте, набираючи у командному рядку у веб-браузері видаляти записи, наприклад

```
$phones='ekXXphones';
$streets='ekXXstreets';
де XX – номер ftp акаунту
```

5. Відредагувати скрипт для створення таблиць таким чином, щоби він створював дві таблиці простого телефонного довідника з такою структурою:

Таблиця phones

№	Назва поля	Тип поля
1	id	int(10) autoincrement
2	phone	varchar(30)
3	pib	varchar(250)
4	str_id	int(10)
5	house	varchar(10)
6	apart	varchar(10)

Таблиця streets

№	Назва поля	Тип поля
1	str_id	int(10) autoincrement
2	street	varchar(250)

Не забудьте використати у скрипті оператори **include** для одержання змінних з'єднання, назв таблиць і заголовочної частини веб-сторінки. Для створення кожної таблиці використайте окремі запити **mysql_query("CREATE TABLE ... без використання команд обробки помилок ... or die(mysql_error());**. В кінці цього скрипту розташуйте декілька команд для додавання 5-6 записів із назвами вулиць до таблиці **streets**, наприклад:

```
mysql_query("INSERT INTO $streets
VALUES(NULL,'Сонячна'),$db);
mysql_query("INSERT INTO $streets
VALUES(NULL,'Соборна'),$db);
```

- ...
- 1. Запустіть скрипт браузером.
- 2. Відредагуйте скрипт для переглядання структури таблиць і переконайтесь, що у вас створились таблиці **phones** і **streets**, а в таблиці **streets** є записи із назвами вулиць.
- 3. Відредагуйте скрипт `edit.php` таким чином, щоб він

гам третьої нормальної формі і, крім того, забороняється зберігати незалежні елементи в одній і тій же таблиці, коли між цими елементами існує зв'язок (зв'язки) багато до багатьох.

5. П'ята нормальна форма: таблиця задовольняє вимогам четвертої нормальної форми та існує можливість модернізації даних таблиці. Передбачається, що при розробці баз даних і особливо при проведенні нормалізації таблиць потрібно звертати велику увагу на те, щоб випадково не проігнорувати суттєву характеристику або параметри об'єкта.

Нормалізація збільшує число відношень в базах даних і тим самим час обробки. Але за рахунок коректності і усунення дублювання відбувається прискорення виконання доступу до даних.

Індексні файли.

При індексуванні що виконується за командою INDEX, також створюється новий файл, який називається індексним, але в цьому файлі містяться лише номери записів, розміщених у тій послідовності, яку вони б мали як би були відсортовані. Індексний файл має розширення IDX. Він не може використовуватись окремо, а використовується тільки разом з його файлом даних. Для кожного файлу даних може бути створено декілька індексних файлів. Максимальна кількість таких файлів може бути сім. Індексні файли дозволяють просто і швидко одержувати базу даних за різними полями.

Завдання

Створити просту двохтабличну РБД. Розробити і створити скрипт для внесення, редагування, переглядання і видалення в головній таблиці цієї РБД.

Хід виконання роботи

1. Створити на своєму ftp-акаунті каталог **lr5**.
2. Скопіювати в каталог **lr5** скрипт **edit.php**, виконаний в лаб. роботі №4.
3. Скопіювати в каталог **lr5** скрипти з лаб. роботі №1 для створення таблиць і переглядання їх структури.
4. В файлі `inc_tab.php` додати два рядки :

`http://vsau.vin.ua/...del.php?id=2`. Результати роботи цього скрипту перевіряйте, запускаючи скрипт `edit.php` в іншій вкладці веб-браузера.

8. Тепер приєднаємо до рядків таблиці записів в скрипті **edit.php** необхідне гіперпосилання. Питання в одному. Де брати код рядку, що треба видалити? З поля `news_id!` Воно у нас йде першим у структурі і тому його номер буде 0 (в php нумерація елементів йде з нуля, а не з 1). Гіперпосилання для видалення встановимо, наприклад, на поле **author_name**. Враховуючи, що воно 5, маємо такі зміни :

```
echo"<table border=1> <tr align=center>
<td width=20>${items[0]}</td>
<td width=60>${items[1]}</td>
<td width=500>${items[2]}</td>
<td width=100>${items[3]}</td>
<td width=80><a
href='del.php?id=${items[0]}${items[4]}</a></td>
<td width=80>${items[5]}</td>
</tr></table>";
```

Зверніть увагу на виділений рядок і розберіть його зміст. Запустіть скрипт `edit.php` і переконайтесь, що записи в цій колонці одержали гіперпосилання. Наведіть курсор мишки на кожне із них і переконайтесь, що параметр `id` має індивідуальні значення.

Переконайтесь, що ці гіперпосилання запускають скрипт **del.php** і він видаляє ці записи.

9. Самостійно доопрацюйте тепер скрипт **edit.php** таким чином, щоби гіперпосилання для видалення записів запускали саме цей скрипт таким чином, щоб на початку він при необхідності видаляв запис, а далі показував форму і список існуючих записів.

Для цього на його початку треба одержати значення параметру `id` і перевірити, чи він не є порожнім. Для не порожнього параметру (це означає, що є що видаляти) запустити оператори

видалення (використайте ідеї із скрипту del.php) і далі продовжити виконання скрипту. Якщо ж параметр порожній, то оператори видалення треба просто оминати і так само продовжити виконання скрипту. Допомога: оператор if (\$x != "...")... (чи в змінній \$x непорожній рядок? " це два апострофи, а не лапки).

Контрольні запитання:

1. Поясніть принцип роботи всіх скриптів цієї лабораторної роботи.
2. Що означає позначення ".../?" ?
3. Для чого використовується оператор **WHERE** і як він працює у створених скриптах?
4. Що виконає цей оператор: `mysql_query("DELETE FROM $news_table ", $db);` ?

ЛАБОРАТОРНА РОБОТА № 5

Тема: Реляційні бази даних (РБД). Дослідження основ роботи із записами у РБД з використанням web-технологій.

Мета: Набуття вмінь навичок у створенні скриптів для редагування записів у РБД.

Теоретичні відомості

При проектуванні реляційних баз даних застосовується метод *декомпозиції*. Метою декомпозиції є вилучення з таблиці полів (стовбців), дані яких можуть повторюватися. Такі поля повинні бути розташовані в новій таблиці. Після цього на дані в новій таблиці повинні вказувати посилання, які містяться в первинній таблиці бази даних.

Умови нормалізації баз даних.

Процес трансформації даних в реляційну форму називається нормалізацією баз даних. Нормалізація - це видалення надлишкових даних з кожної таблиці бази даних. Процес нормалізації переслідує подвійну мету: видалити надлишкові копії даних і забезпечити максимальну гнучкість, як в структурах таблиць, так і в її інтерфейсних програмах на випадок можливих майбутніх змін баз даних. Найчастіше зустрічаються п'ять форм нормалізації, що означає п'ять різних установок реляційного критерію нормалізації баз даних.

1. Перша нормальна форма: кожне з полів заданої таблиці повинно бути неподільним по змісту використання і не містити повторюваних груп. Поле вважається неподільним, якщо воно містить тільки один елемент даних.

2. Друга нормальна форма: задана таблиця задовольняє вимогам першої нормальної форми і, крім того, всі неключові поля таблиці повністю залежать від первинного ключа таблиці і від кожного поля в первинному ключі.

3. Третя нормальна форма: таблиця задовольняє вимогам другої нормальної форми і, крім того, всі неключові поля повністю залежать від первинного ключа таблиці і не залежать один від одного.

4. Четверта нормальна форма: таблиця задовольняє вимо-

команду зміни запису у таблиці — **UPDATE** ... Окрім надання нових значень вказаним полям, не забудьте вказати умову, за якою будуть відбуватись зміни у таблиці. Нею має бути збіжність **id** редагованого запису із вказаним:

... **WHERE news_id = \$news_id** ...

14. Відкрийте у браузері скрипт **edit.php** і переконайтесь у працездатності режиму редагування.

15. Об'єднайте напрацювання двох скриптів в один. Для цього доопрацюйте форму в **edit.php**. Додайте до неї тег із кнопкою, який використовувався в скрипті **update.php**

```
<input type="submit" name="update" value="Змінити дані">
```

Тепер їх на формі буде дві. Одна для додавання записів, а друга для їх редагування.

16. Додайте до форми невидиме поле із **news_id**.

17. Виведіть значення одноіменних змінних у всіх полях форми за зразком **update.php**.

18. Замініть гіперпосилання у таблиці з **update.php** на **edit.php**.

19. Відкрийте скрипт **edit.php** і переконайтесь, що він правильно працює у режимах редагування, додавання і видалення записів.

20. Наведіть у звіті до лабораторної роботи тексти створених скриптів і детальні пояснення їх роботи.

Контрольні запитання:

1. Поясніть принцип роботи всіх скриптів цієї лабораторної роботи.

2. Як працює оператор **UPDATE** ? Наведіть власні приклади його використання.

3. Яким чином можна одержати значення параметрів з командного рядку?

4. Для чого використовуються невидимі поля у формах? Наведіть власні приклади.

ЛАБОРАТОРНА РОБОТА № 4

Тема: Технологія редагування записів у таблицях.

Мета: Набуття навичок у створенні скриптів для редагування записів у таблицях БД.

Теоретичні відомості

Команда **UPDATE** дозволяє змінювати значення в таблиці. Для цього їй треба вказати назву таблиці, назву поля(ів) і значення, яке треба в ньому (у них) замінити.

Наприклад,

```
mysql_query("UPDATE news SET  
body = 'В Україні настала гарна погода ',  
author_name = 'Василенко М.А.' ",  
$db);
```

Із наведеного зрозуміло, що до вказаних полів (**body** і **author_name**) маю внести відповідні значення. Але ж децо ми забули! Тут не вказано, в ЯКОМУ записі треба зробити ці зміни? Помилка? Ні. Просто ці значення внесуться до ВСІХ записів, що є у таблиці.

Як вказати серверу, ДЕ треба робити зміни. Для цього існує оператор умови **WHERE** (перекладається “в тому випадку коли”). В мові PHP на ньому створюються цикли. А в мові запитів від використовується для вказування умови:

```
mysql_query("UPDATE news SET  
body = 'Зібраний гарний урожай ',  
author_name = 'Хлопецька О.Ф.'  
WHERE news_id='1' ",  
$db);
```

Тепер зміни будуть внесені лише до запису, який має в полі **news_id** значення 1.

Кількість одночасно змінюваних полів і записів може бути будь-яка. А сама умова може бути і складною, як в операторі IF, наприклад

```
WHERE author_email = 'my@asd.com' AND  
heading='Послини'
```

Завдання

Маючи у своєму розпорядженні скрипт, що був створений у лабораторній роботі №3 для внесення і видалення записів в таблиці, переглядання вже внесених записів, створити скрипт, який би дозволив редагувати існуючі записи.

Хід виконання роботи

1. Створити на своєму ftp-акаунті каталог lr4.
2. Скопіювати туди скрипт **edit.php**.
3. Створити копію скрипту **edit.php** в цьому ж каталозі під назвою **update.php**.
4. Підготувати скрипт **update.php** до роботи. Для цього залишити в ньому лише include-рядки:

```
include('../inc.php');  
include('../inc_tab.php');  
include('../header.php');
```

а також ті, які відповідають за виведення форми на екран.

5. В скрипті **edit.php** додати до будь-якої колонки таблиці, де виводиться список записів, гіперпосилання на скрипт **update.php** аналогічно до раніше створеного гіперпосилання для видалення записів. В командному рядку цього гіперпосилання передати значення всіх полів таблиці до скрипту **update.php**. Одразу визначте для себе єдину систему назв змінних і полів таблиці для редагування. Для спрощення рекомендуємо використати назви полів із таблиці (**news_id**, **auth**, **auth_email**, **heading**, **body**). Гіперпосилання створіть наприклад:

```
<a  
href='update.php?news_id=$a[0]&auth=$a[1]&auth_email=...&heading=...&body=...>$a[...]</a>
```

(наведений масив \$a є умовним. У власному проекті використайте той, що вже є у скрипті **edit.php** і правильно вкажіть його індекси в командному рядку).

6. Додайте до скрипту **update.php** рядки, за допомогою яких можна одержати значення всіх змінних із командного рядку, наприклад

```
$news_id=$_GET['news_id'];  
...
```

7. Зробіть так, щоб на формі, яку створює **update.php** з'являлись значення змінних, що одержані із командного рядку. Для цього в тегах полів (`<input type='text' ...>`) надайте властивостям **value** значення відповідних змінних. Наприклад,

```
<input name="auth_email" type="text" value="<? echo $auth_email; ?>">
```

8. Додайте до форми скрипту **update.php** невидиме поле для зберігання значення **id**-редагованого рядку таблиці:

```
<input name="news_id" type="text" value="<? echo $news_id; ?>">
```

9. Відкрийте у браузері скрипт **edit.php**, наведіть вказівник мишки на гіперпосилання в таблиці із існуючими записами на поле для редагування і переконайтесь, що командний рядок і його зміст, який покаже браузер, відповідають вашому проекту. Клацніть на гіперпосилання і переконайтесь, що у всіх полях форми скрипту **update.php** з'являться значення із рядку таблиці скрипту **edit.php**.

10. Відкрийте у браузері код одержаної WEB-сторінки (CTRL_U) і переконайтесь, що у невидимому полі зберігається правильне значення **news_id**.

11. Переконайтесь, що у якості **action** у цій формі буде виконуватись скрипт **edit.php**. Замініть назви кнопки цього скрипту на іншу і в подальшому використайте її для визначення, чи була натиснута кнопка редагування на формі, наприклад

```
<input type="submit" name="update" value="Змінити дані">
```

12. Знову відкрийте скрипт **edit.php** і додайте після операторів з'єднання з базою даних оператори для одержання значень полів із форми скрипту **update.php** за зразком :

```
$news_id=$_GET['news_id'];
```

13. Додайте оператор умовного переходу, який би аналізував, чи була натиснута кнопка на формі в скрипті для редагування. (Використайте свою назву і ідею визначення, яка вже закладена в цьому скрипті). Цей оператор має виконувати