

КАТЕДРА: КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ
ДИСЦИПЛИНА: ИНФОРМАЦИОННИ СИСТЕМИ

ЛАБОРАТОРНО УПРАЖНЕНИЕ № 5

ТЕМА: Информационно търсене. Генериране на справки чрез php скрипт от MySQL база данни

ЦЕЛ:

Целта на упражнението е студентите да получат практически знания и умения за търсене на информация от MySQL база данни чрез PHP скрипт След упражнението студентите би следвало и да могат да реализират елементарни справки към базата данни чрез PHP.

I. ТЕОРЕТИЧНА ЧАСТ

При търсенето на информация се използва критерии за съответствие. Той се състои в необходимостта от определяне на последователност от правила за установяване на съответствието. Под това се разбира последователно усложняващ се анализ на обекта с постепенен обхват на все по-голям брой от неговите признаци

Извличането на данни от MySQL база данни става след като се изпращат определени SQL заявки към базата чрез функцията **mysql_query()**. Функцията връща ресурсен указател към всички записи които отговарят на подадената SQL заявка. В предходното упражнение показахме как може да бъде прочетена информацията чрез функцията **mysql_fetch_array()**. Сега ще разгледаме още няколко функции, които биха ни помогнали да разберем извлеченото съдържание от **mysql_query()**.

Функцията **mysql_fetch_row()** работи на абсолютно същия принцип като функцията **mysql_fetch_array()**. Има същия синтаксис:

Масив = **mysql_fetch_row** (връзкаКъмДанни);

Приема само един параметър – ресурса, получен от функцията **mysql_query()** и връща масив. Разликата е, че ключовете на масива не са имената на полетата от извлечения запис от базата данни а целочислени индекси 0,1,2 и т.н. Ако използваме функцията в цикъл, отново ще покажем всички извлечени записи.

```
While (Масив = mysql_fetch_row ( връзкаКъмДанни ) )  
  { echo Масив[0].“ “;  
  .....  
  echo Масив[N].“ “;}
```

Функцията **mysql_data_seek()** служи за позициониране на точно определен запис от извлеченията. Тя има два параметъра. Първият е ресурсния указател върнат от функцията **mysql_query()**, а вторият- целочислен индекс който показва номера (реда) на записа, който да бъде позициониран. Функцията връща булев резултат в зависимост от това дали е извършено успешно позиционирането или не. Нейният синтаксис е следния:

\$pos=mysql_data_seek(връзкаКъмДанни,0) or die('Възникна грешка');

Нулата като втори параметър във функцията означава да се локализира първия запис от ресурса с извлечени записи (върнатия резултат от **mysql_query()**). Ако е необходимо да се локализира 2рия запис то вместо нула ще пише 1. За n-тия ще пише n-1. Ако тази

функция се използва преди `mysqli_fetch_row()` и `mysqli_fetch_array()`, то ще бъде прочетен само позиционирания запис и всички след него (ако се използват в цикъл).

Функцията `mysqli_num_rows()` връща броя на извлечените записи. За нея се подава един параметър – ресурса върнат от `mysqli_query()`. Синтаксисът е следни:

`$number=mysqli_num_rows(връзкаКъмДанни) ,`

където `$number` е броя на извлечените записи.

II. ПРАКТИЧЕСКА ЧАСТ

За целите на упражнението с phpMyAdmin е създадена база данни “glasesrus”, в която са създадени следните таблици: “customers”, “products”, “purchases” и “purchasesproducts”. Таблица “customers” ще съдържа клиентите на даден магазин, “products” – продуктите които се продават, “purchases” – поръчките на клиентите, а таблица “purchasesproducts” е връзката между продуктите и поръчките. Структурата на таблиците е показана на фиг.1 а), б), в), г). а съдържанието им (записите им) на фиг.2 а), б), в), г). Задачата която е решена е следната:

ЗАДАЧА: Да се намерят всички поръчки на клиенти със Surname отговарящо на “Jones”. Съответно на коя дата какви продукти са закупили.

!ВАЖНО: За да можете да тествате упражнението ви е необходим следния софтуер. Web server, Php интерпретатор, MYSQL база данни. Всеки един от тези софтуери ги има в пакетите WAMP или XAMP. Настоящото упражнение е тествано върху WAMP Version 3.2.0.

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително	Действие
1	Id	int(11)			Не	Няма		AUTO_INCREMENT	Промяна Изтриване
2	Title	varchar(10)	latin1_swedish_ci		Не	Няма			Промяна Изтриване
3	Surname	varchar(100)	latin1_swedish_ci		Не	Няма			Промяна Изтриване
4	Firstname	varchar(100)	latin1_swedish_ci		Не	Няма			Промяна Изтриване

Действие	Име на ключ	Тип	Уникално	Опаковани	Колона	Кардиналност	Колация	Празно	Коментар
Редакция Изтриване	PRIMARY	BTREE	Да	Не	Id	12	A	Не	

фиг.1.а) Структура на таблица “customers”

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително	Действие
1	Id	int(11)			Не	Няма		AUTO_INCREMENT	Промяна Изтрив
2	Name	varchar(255)	latin1_swedish_ci		Не	Няма			Промяна Изтрив
3	Description	text	latin1_swedish_ci		Не				Промяна Изтрив
4	Quantity	int(11)			Не	Няма			Промяна Изтрив
5	Cost	float			Не	Няма			Промяна Изтрив

Маркиране всички Когато има отметка: Прелистване Промяна Изтриване Първичен Уникално

Индекс Пълнотекстово Пълнотекстово

Печат Анализ на таблицата Преместване колони Normalize

Добавяне 1 колона(и) след Cost Изпълнение

Индекси

Действие	Име на ключ	Тип	Уникално	Опаковани	Колона	Кардиналност	Колация	Празно	Коментар
Редакция Изтриване	PRIMARY	BTREE	Да	Не	Id	8	A	Не	

фиг.1.б). Структура на таблица "products"

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително	Действие
1	Id	int(11)			Не	Няма		AUTO_INCREMENT	Промяна Изтриване
2	customers_id	int(11)			Не	Няма			Промяна Изтриване
3	Day	int(11)			Не	Няма			Промяна Изтриване
4	Mount	int(11)			Не	Няма			Промяна Изтриване
5	Year	int(11)			Не	Няма			Промяна Изтриване

Маркиране всички Когато има отметка: Прелистване Промяна Изтриване Първичен Уникално

Индекс Пълнотекстово Пълнотекстово

Печат Анализ на таблицата Преместване колони Normalize

Добавяне 1 колона(и) след Year Изпълнение

Индекси

Действие	Име на ключ	Тип	Уникално	Опаковани	Колона	Кардиналност	Колация	Празно	Коментар
Редакция Изтриване	PRIMARY	BTREE	Да	Не	Id	7	A	Не	
Редакция Изтриване	customers_id	BTREE	Не	Не	customers_id	A		Не	

фиг.1.в) Структура на таблица „purchases“

#	Име	Тип	Колация	Атрибути	Празно	По подразбиране	Коментари	Допълнително	Действие
1	products_id	int(11)			Не	Няма			Промяна Изтриване
2	purchases_id	int(11)			Не	Няма			Промяна Изтриване
3	Quantity	int(11)			Не	Няма			Промяна Изтриване
4	Cost	float			Не	Няма			Промяна Изтриване

Маркиране всички Когато има отметка: Прелистване Промяна Изтриване Първичен Уникално

Индекс Пълнотекстово Пълнотекстово

Печат Анализ на таблицата Преместване колони Normalize

Добавяне 1 колона(и) след Cost Изпълнение

Индекси

Действие	Име на ключ	Тип	Уникално	Опаковани	Колона	Кардиналност	Колация	Празно	Коментар
Редакция Изтриване	products_id	BTREE	Не	Не	products_id	A		Не	
					purchases_id	A		Не	

фиг.1.г). Структура на таблица „purchasesproducts“

				Id	Title	Surname	Firstname
<input type="checkbox"/>		Редакция		Копиране		Изтриване	1 Mrs Smith Lunne
<input type="checkbox"/>		Редакция		Копиране		Изтриване	2 Miss Jones Ann
<input type="checkbox"/>		Редакция		Копиране		Изтриване	3 Mr Brown Simon
<input type="checkbox"/>		Редакция		Копиране		Изтриване	4 Mr Smith David
<input type="checkbox"/>		Редакция		Копиране		Изтриване	5 Mr Bell Peter
<input type="checkbox"/>		Редакция		Копиране		Изтриване	6 Mr Hal Elizabeth
<input type="checkbox"/>		Редакция		Копиране		Изтриване	7 Mr Smith Kevin
<input type="checkbox"/>		Редакция		Копиране		Изтриване	8 Mr Jones Jack
<input type="checkbox"/>		Редакция		Копиране		Изтриване	9 Mr Green Wiliam
<input type="checkbox"/>		Редакция		Копиране		Изтриване	10 Mr Bell Simon
<input type="checkbox"/>		Редакция		Копиране		Изтриване	11 Mr Brown Ian
<input type="checkbox"/>		Редакция		Копиране		Изтриване	16 Mr asd sda

фиг.2.а) Записи в таблица "customers"

					Id	Name	Description	Quantity	Cost
<input type="checkbox"/>		Редакция		Копиране		Изтриване	1 Beer Glass 600ml Beer Glas	345	3.99
<input type="checkbox"/>		Редакция		Копиране		Изтриване	2 Wine Glass 125ml Wine Glass	236	2.99
<input type="checkbox"/>		Редакция		Копиране		Изтриване	3 Wine Glass 175ml Wine Glass	436	3.5
<input type="checkbox"/>		Редакция		Копиране		Изтриване	4 Shot Glass 50ml small Glass	132	1.5
<input type="checkbox"/>		Редакция		Копиране		Изтриване	5 Spirit Glass 100 ml spirit Glass	489	2.5
<input type="checkbox"/>		Редакция		Копиране		Изтриване	6 Long Glass 200 ml Tall Glass	263	2.5
<input type="checkbox"/>		Редакция		Копиране		Изтриване	7 Beer Glass 300 ml Beer Glass	247	2.99
<input type="checkbox"/>		Редакция		Копиране		Изтриване	8 Wine Glass 225 ml Wine Glass	96	3.99

фиг.2.б). Записи в таблица „products“

					Id	customers_Id	Day	Mount	Year		
<input type="checkbox"/>		Редакция		Копиране		Изтриване	1	2	3	9	2006
<input type="checkbox"/>		Редакция		Копиране		Изтриване	2	4	6	9	2006
<input type="checkbox"/>		Редакция		Копиране		Изтриване	3	6	13	9	2006
<input type="checkbox"/>		Редакция		Копиране		Изтриване	4	2	22	9	2006
<input type="checkbox"/>		Редакция		Копиране		Изтриване	5	1	28	9	2006
<input type="checkbox"/>		Редакция		Копиране		Изтриване	6	9	1	10	2006
<input type="checkbox"/>		Редакция		Копиране		Изтриване	7	7	1	10	2006

фиг.2.в). Записи в таблица „purchases“

products_Id	purchases_Id	Quantity	Cost
2	1	20	2.99
3	2	10	3.5
8	2	30	4.5
6	3	25	2.5
3	4	10	3.5
4	4	100	1.5
5	4	40	3
1	5	22	3.99
1	6	6	3.99
3	7	15	3.5
4	7	25	2
5	7	10	2.5
7	7	55	2.5
8	7	1	3.99

Фиг.2.г). Записи в таблица „purchasesproducts“

1. Код за решение на задачата:

```
<?php
//----- Част 1 -----
    require_once("database.php");
//-----

//----- Част 2 -----
    $strSurname='Jones';
    $queryCust="SELECT * FROM customers WHERE Surname='$strSurname' ";
    $dbCustRecords=mysqli_query($con,$queryCust);
    if (!$dbCustRecords) die('Неуспешно запитване. Получи следната
грешка'.mysqli_error($con));
//-----

//----- Част 3 -----
    While ($CustRecords=mysqli_fetch_array($dbCustRecords))
    {
        $intCustId=$CustRecords["Id"];
        echo "<p> Customers: ";
        echo $CustRecords["Title"]." ";
        echo $CustRecords["Surname"]." ";
        echo $CustRecords["Firstname"]."</p>";
//-----

//----- Част 4 -----
        $queryPur="SELECT * FROM purchases WHERE
customers_Id='$intCustId'";
        $dbPurRecords=mysqli_query($con,$queryPur);
        if (!$dbPurRecords) die('Неуспешно запитване. Получи
следната грешка'.mysqli_error($con));
//-----

//----- Част 5 -----
        while ($PurRecords=mysqli_fetch_array($dbPurRecords))
        {
            $intPurId=$PurRecords["Id"];
            echo "<p> Purchases On: ";
            echo $PurRecords["Day"]." ";
            echo $PurRecords["Mount"]." ";
            echo $PurRecords["Year"]."</p>";
//-----

//----- Част 6 -----
            $queryPurPro="SELECT * FROM purchasesproducts WHERE
purchases_Id='$intPurId'";
            $dbPurProRecords=mysqli_query($con,$queryPurPro);
            if (!$dbPurProRecords) die('Неуспешно запитване. Получи
следната грешка'.mysqli_error($con));
//-----

//----- Част 7 -----
            while($PurProRecords=mysqli_fetch_array($dbPurProRecords))
            {
                $intProId=$PurProRecords["products_Id"];
                echo "<p>".$PurProRecords["Quantity"]." ";
//-----
```

```

//----- Част 8 -----
$queryPro="SELECT * FROM products WHERE
Id='$intProId'";
$dbProRecords=mysqli_query($con,$queryPro);
if (!$dbProRecords) die('Неуспешно запитване.
Получи следната грешка'.mysqli_error($con));
//-----

//----- Част 9 -----
$ProRecords=mysqli_fetch_array($dbProRecords);
echo $ProRecords["Name"]."
(".$ProRecords["Description"].") at &#163;";
echo $ProRecords["Cost"]." each. </p>";
}
}
}
//-----

//----- Част 10 -----
mysqli_close($con);
//-----
?>

```

2. Пояснение на кода

В част 1 от кода се осъществява достъп до файл “database.php” в който се намира кода който отваря връзка към сървъра и осъществява достъп до базата данни. Кода в този файл е подобен на предните упражнения. В началото на част две декларираме променлива \$strSurname, на която даваме стойност „Jones“. Това е фамилията на клиентите в която ще търсим в базата. Като използваме заявка към базата, записана в променлива \$queryCust, съхраняваме в ресурса \$dbCustRecords всички клиенти които имат фамилия „Jones“.

В част 3 с цикъл “while” изваждаме всички тези записи за клиенти и показваме на екрана техните данни. Копирме ключа Id на таблицата “customers” в променливата \$intCustId за да я използваме по късно.

С част 4 докато сме в цикъл “while” използваме нова заявка към базата данни. Тя избира всички покупки съдържащи идентификатора на клиента, който съхранихме в променливата \$intCustId в предната част.

В част 5 със нов цикъл „while“, който е вложен в първия, се извеждат тези покупки и показва кога са били направени. Идентификационния ключ на покупките съхраняваме в променливата \$intPurId.

С част 6, докато сме все още във вложения цикъл, изпращаме нова заявка към базата от данни, която избира всички записи от таблицата „purchasesproduct“ съдържащи идентификатора на покупката записан в \$intPurId.

В част 7 с нов цикъл „while“, който е вложен във втория „while“ започващ в част 5, зарежда продуктите от покупката и ги показва на екрана. Полето „products_Id“ съхраняваме в променливата \$intProId.

С част 8 изпращаме още една заявка към базата данни направена в последния “while” цикъл, която извлича всички продукти които съответстват на ключа, съхранен в \$intProId (винаги ще съответства само на един продукт). Тъй като само един запис на продукт съответства на ключа, в част 9 не е необходим нов “while” цикъл, а просто извикваме функцията mysqli_fetch_array(), която връща всички полета от записа.

В част 10 е затворена връзката към сървъра с базата данни. Резултатът от изпълнението на скрипта се вижда на фиг.3.



Фиг.3. Резултат от решението на задачата.

III. Задача за самостоятелна работа.

1. Осъществете връзка до базата данни "Firma", която би трябвало да сте създали в задачите за самостоятелна работа от предходните лабораторни упражнения
2. Създайте PHP скрипт който да извлече само описанието на 3тия запис от таблицата "Products".
Указание: За целта използвайте функциите `mysqli_data_seek()` и `mysqli_fetch_row()`.
3. Напишете допълнение към скрипта в практическата част на това лабораторно упражнение, което да генерира подходящи съобщения за грешка, ако на някои от етапите не бъде открит запис. Например „Не са направени покупки“, или „Няма клиенти с такова име“ и т.н.
Указание: За да проверявате дали има открити записи използвайте функцията `mysqli_num_rows()`.
4. Създайте потребителска форма със падащ списък който да предоставя възможност на потребителя да избере името на клиентите от таблица „customers“ и с натискането на бутон „ОК“ да се изведат всички поръчки за този клиент.