МИНИСТЕРСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ»

Обособленное подразделение

«КОЛЛЕДЖ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ»

ФАКУЛЬТЕТ ЭЛЕКТРОСВЯЗИ

КАФЕДРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СЕТЕЙ ТЕЛЕКОММУНИКАЦИЙ

Отчёт

к лабораторной работе №13

ИСПОЛЬЗОВАНИЕ СРЕДСТВА ЯЗЫКА PHP ДЛЯ РАБОТЫ С СУБД

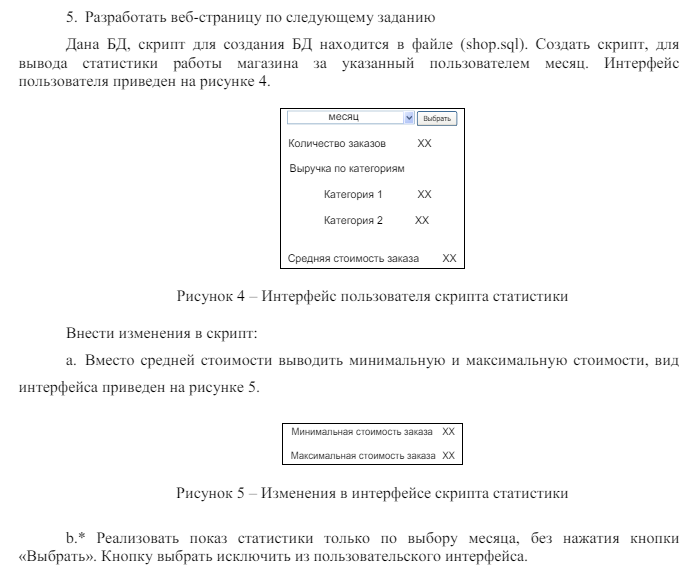
по дисциплине

«Конструирование программ и языки программирования»

Выполнил учащийся гр. ПО912 С. П. Шепелев

Руководитель Н. И. Янович

Минск 2022

****

**ПРИМЕР**

<?php

//Определяем параметры MySQL.

$DB\_HOST = 'localhost'; // MySQL server hostname

$DB\_PORT = '3306'; // MySQL server port number (default 3306)

$DB\_NAME = 'test'; // MySQL database name

$DB\_USER = 'root'; // MySQL username

$DB\_PASS = ''; // password

try {

// Устанавливаем соединение с MySQL сервером.

$dbConn = new PDO("mysql:host=$DB\_HOST; port=$DB\_PORT;dbname=$DB\_NAME", $DB\_USER, $DB\_PASS);

$dbConn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

//Установить режим обработки ошибок

echo 'Connected', '<br />';

//Создаем подготовленный оператор для удаления существующей таблицы

$pstmt = $dbConn->prepare('DROP TABLE IF EXISTS `test`');

$pstmt->execute(); // запускаем подготовленный оператор

echo 'DROP TABLE: ', $pstmt->rowCount(),' rows', '<br />';

//Создаем подготовленный оператор для создания тестовой q

$pstmt = $dbConn->prepare(

'CREATE TABLE IF NOT EXISTS `test` (

`id` INT AUTO\_INCREMENT,

`name` VARCHAR(20),

PRIMARY KEY (`id`))');

$pstmt->execute(); // запускаем подготовленный оператор

echo 'CREATE TABLE: ', $pstmt->rowCount(),' rows', '<br />';

// Создаем подготовленный запрос с именованными параметрами

// в форме :paramName

$pstmt = $dbConn->prepare('INSERT INTO `test` (`id`, `name`)

VALUES (:id, :name) ');

// Привязываем параметры к программным переменным

// с указанным типом

$pstmt->bindParam(':id', $id, PDO::PARAM\_INT);

$pstmt->bindParam(':name', $name, PDO::PARAM\_STR);

$id = 2001;

$name = 'john';

$pstmt->execute(); // запускаем подготовленный запрос

echo 'INSERT INTO: ', $pstmt->rowCount(),' rows', '<br />';

echo 'LAST\_INSERT\_ID (of the AUTO\_INCREMENT column) is ',

$dbConn->lastInsertId(),'<br />';

// Изменяем значения переменных-параметров запроса

$id = 2002;

$name = 'jane';

$pstmt->execute();

// Запускаем подготовленный оператор снова с новыми значениями

// Выводим статистику по добавленным записям

// количество вставленных записей

echo 'INSERT INTO: ', $pstmt->rowCount(),'rows', '<br />';

// Значение уникального ключа для последней вставленной записи

echo 'LAST\_INSERT\_ID (of the AUTO\_INCREMENT column) is ',

$dbConn->lastInsertId(),'<br /><br />';

// Создадим подготовленный запрос

// с позиционными (неименованными) параметрами в форме ?

$pstmt = $dbConn->prepare('SELECT \* FROM `test` WHERE `id` = ?

AND `name` LIKE ? ');

$pstmt->bindParam(1, $id, PDO::PARAM\_INT); //Индекс id начать с 1

$pstmt->bindParam(2, $name, PDO::PARAM\_STR);

$name = 'j%';

$id = 2001;

$pstmt->execute();

echo 'SELECT: ', $pstmt->rowCount(), ' rows', '<br />';

// Результат выборки будем извлекать построчно.

// По умолчанию в результирующей строке столбцы индексируются

// как именами столбцов в таблице данных, так и индексными значени-ями

// начиная с 0.

while ($row = $pstmt->fetch()) {

echo 'Retrieve via column name: id=', $row['id'],

' name=', $row['name'], '<br />';

echo 'Retrieve via column number: id=', $row[0],

' name=', $row[1], '<br />';

print\_r($row); // вывести содержание результирующей строки

echo '<br />';

}

echo '<br />';

// Выполним запрос снова, но с другим значением $id.

$id = 2002;

$pstmt->execute();

echo 'SELECT: ', $pstmt->rowCount(),' rows', '<br />';

// Извлечем полный набор результатов (все строки) в буфер програм-мы

// с опцией «FETCH\_ASSOC», которая означает что данные будут

// извлечены только в ассоциативный массив с индексами,

//соответствующими названиям столбцов в таблице базы данных

$resultset = $pstmt->fetchAll(PDO::FETCH\_ASSOC);

// print\_r($resultset);

foreach ($resultset AS $row) {

echo 'Retrieve via column name: id=', $row['id'],

'name=', $row['name'], '<br />';

print\_r($row); // for showing the contents of resultset’s row

echo '<br />';

}

echo '<br />';

$pstmt->execute();

echo 'SELECT: ', $pstmt->rowCount(),' rows', '<br />';

// Извлечем полный набор результатов (все строки) в буфер програм-мы

// с опцией «FETCH\_OBJ», которая означает что данные будут

// извлечены объект со значениями свойств, соответствующими

// названиям столбцов в таблице базы данных СУБД

$resultset = $pstmt->fetchAll(PDO::FETCH\_OBJ);

foreach ($resultset AS $row) {

// Обрабатываем каждую строку в буфере

echo 'Retrieve via column name: id=', $row->id,

'name=', $row->name, '<br />';

print\_r($row);

echo '<br />';

}

// Закрыть соединение с СУБД

$dbConn = NULL;

} catch (PDOException $e) {

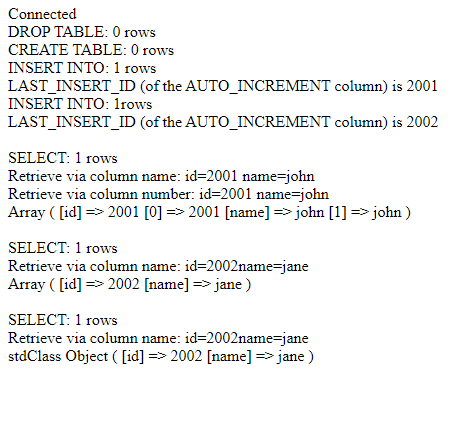
$fileName = basename($e->getFile(),".php");

$lineNumber = $e->getLine();

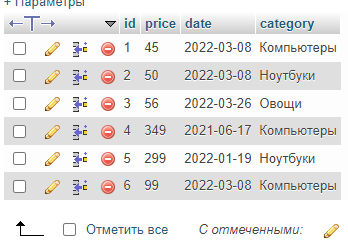
die("[$fileName][$lineNumber] Database error: ".$e->getMessage().'<br />');

}

?>



**ЗАДАНИЕ**

**

*config.php*

<?php

$dsn = 'mysql:host=localhost; port=3306; dbname=shop';

$pdo = new PDO($dsn, 'root', '');

?>

*1.php*

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Магазин</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<table>

<tr><td colspan=2>

<form name='f' action='' method="POST">

<select name='s' onchange="this.form.submit()">

<?php

$months = array('Январь','Февраль','Март','Апрель','Май','Июнь','Июль','Август','Сентябрь','Октябрь','Ноябрь','Декабрь');

for($i=0;$i<count($months);$i++)

{

echo "<option value=",($i+1);

if($\_POST['s']==($i+1)){

echo ' selected';

}

echo ">",$months[$i],"</option>";}

?>

</select>

</form>

</td></tr>

<?php

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

}

else{

$\_POST['s']=1;

}

setcookie('month',$\_POST['s'], time()+600);

require 'config.php';

$sql = "SELECT category, sum(price) FROM `orders` where month(date) = :mon GROUP BY `category`";

$sql2 = "SELECT min(price), max(price), round(avg(price)), count(\*) FROM `orders` where month(date) = :mon";

$query = $pdo->prepare($sql);

$query->execute(['mon'=>$\_POST['s']]);

$query2 = $pdo->prepare($sql2);

$query2->execute(['mon'=>$\_POST['s']]);

$roww = $query2->fetchALL(PDO::FETCH\_ASSOC)[0];

if($roww['count(\*)']==0){

echo "<tr><td colspan=2 style='text-align:center'>Заказов в этом месяце не было</td></tr>";

}

else{

echo "<tr><td>Количество заказов </td><td style='text-align:right'>".$roww['count(\*)']."</td></tr>";

echo "<tr><td colspan=2>Выручка по категориям</td></tr><tr><td colspan=2><ul>";

foreach(($query->fetchALL(PDO::FETCH\_ASSOC)) as $row){

echo '<li>'.$row['category'].' - '.$row['sum(price)'].' руб.</li>';

}

echo "</ul></td></tr>";

echo "<tr><td>Средняя стоимость заказов </td><td>".$roww['round(avg(price))'].' руб.</td></tr>';

echo "<tr><td>Минимальная стоимость заказа </td><td>".$roww['min(price)'].' руб.</td></tr>';

echo "<tr><td>Максимальная стоимость заказа </td><td>".$roww['max(price)'].' руб.</td></tr>';

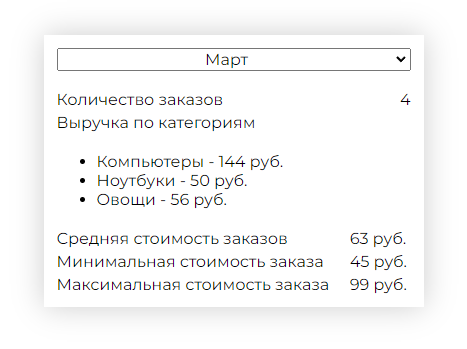
}

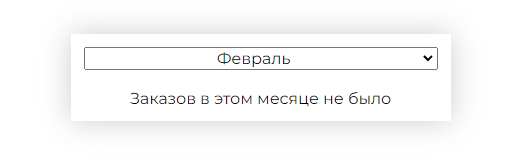
?>

</table>

</body>

</html>





**Контрольные вопросы**

1. Какие функции в языке PHP предназначены для работы с MySQL и каково их назначение?

**Методы класса PDO**

**beginTransaction** – Инициализация транзакции

**commit** – Фиксирует транзакцию

**construct** – Создает экземпляр PDO, предоставляющий соединение с базой данных

**errorCode** – Возвращает код SQLSTATE результата последней операции с базой данных

**errorInfo** – Получает расширенную информацию об ошибке, произошедшей в ходе последнего обращения к базе данных

**exec** – Выполняет SQL-запрос и возвращает количество затронутых строк

**getAttribute** – Получить атрибут соединения с базой данных

**getAvailableDrivers** – Возвращает массив доступных драйверов PDO

**inTransaction** – Проверяет, начата ли транзакция

**lastInsertId** – Возвращает ID последней вставленной строки или значение последовательности

**prepare** – Подготавливает запрос к выполнению и возвращает связанный с этим запросом объект

**query** – Выполняет SQL-запрос и возвращает результирующий набор в виде объекта PDOStatement

**quote** – Заключает строку в кавычки для использования в запросе

**rollBack** – Откат транзакции

**setAttribute** – Установка атрибута

1. Сформулируйте алгоритм построения SQL-запроса на основании данных HTML-формы.

$query = $pdo->prepare("SELECT category, sum(price) FROM `orders` where month(date) = **:mon** GROUP BY `category`");

$query->execute**(['mon'=>$\_POST['s']]);**