**Ф И Л Ь Т Р А Ц И Я Д А Н Н Ы Х Б Д**

 На этапе проектирования приложения БД часто приходится использовать фильтры. Например, фильтры вы можете наблюдать на сайтах интернет магазинов, когда фильтруете данные по цвету, размеру, цене и т.д. Стоит отметить, что пользователем могут быть заданы не все возможные критерии для фильтров, что несколько усложняет код для обработки данных, пример такой обработки

|  |
| --- |
| //фильтрация данных//Ниже приведем код для фильтрации данных таблицы `orders`, возможные поля для фильтров 'd\_i', ' d\_out', //'id\_room', однако отметит, что некоторые поля для фильтра могут быть и не заданы. Код, который приведен //ниже автоматически отбирает те фильтры, которые были заданы require 'connect.php';$query = "SELECT \* FROM `orders` ";$where\_arr = array(); //определяем массив возможных фильтров для where $fields = array( 'd\_i', ' d\_out', 'id\_room' ); // возможные поля фильтров !!!важно эти поля должны совпадать с названиями полей в исходной таблице foreach($fields as $v) //цикл foreach для обхода по всем возможным полям { if (isset($\_GET[$v])) // если фильтр задан, то заносим его в массив возможных фильтров { array\_push( $where\_arr, $v.'='.$\_GET[$v] );  }}$result = mysqli\_query( $db, $query . implode(' AND ', $where\_arr) ); //вызываем запрос для фильтрации  |

 Рассмотрим с вами основные этапы фильтрации данных, на примере сайта гостиницы.

Работу приложения можно организовать следующим образом. Разбить на два функционала. Первый функционал - основная часть приложения, которая реализует пользовательскую часть приложения. Например, онлайн бронирование номеров, просмотр номерного фонда гостиницы, цен (Гиперссылки на Рис.1, так называемый header сайта ) и , самое главное, просмотр и бронирование свободных номеров на интересующие даты (фильтры по датам и количеству отдыхающих на Рис.1).

 Второй функционал - работа приложения на стороне администратора, который может заполнять таблицы БД, редактировать и удалять данные (Админская часть справа - right sidebar).



Рис. 1

 На странице для фильтрации данных предлагаются поля для фильтров, как правило, в виде выпадающих списков. В случае, когда надо ввести дату, можно воспользоваться следующим html фрагментом

|  |
| --- |
| <form action="find.php" method="post"> //обработчик find.php получит данные $\_POST['date\_in'], // $\_POST['date\_out'], $\_POST[' count '] <p> <label for="date\_in">Date in: </label> <input type="date" name="date\_in"/> <label for="date\_out">Date out: </label> <input type="date" id="date\_out" name="date\_out"/><input type=text name="count" size=4 value="1"> Cout </input> ;<input name="find" type="submit" value="FindЮ</p></form>  После выбора пользователем фильтров нажимаем на кнопку Find и отправляем выбранные данные обработчику find.php, который далее уже работает с ними по своему усмотрению. Например, в нашем случае, обращается к таблицам БД и выбирает те номера, которые не заняты на указанные даты и выдает их на экран.  <?php// обработка запроса POST, ($\_POST - ассоциативный массив)if (isset($\_POST['date\_in']) && isset($\_POST['date\_out'])) require 'connect.php'; $date\_in\_=$\_POST['date\_in']; $date\_out\_=$\_POST['date\_out']; $date\_out=strtotime($\_POST['date\_out']); //переводим в формат даты $date\_in=strtotime($\_POST['date\_in']); //переводим в формат даты $m\_out = date('m', $date\_out); //выщепляем месяц от даты $m\_in=date('m', $date\_in); $d\_out = date('d', $date\_out); //выщепляем день от даты $d\_in=date('d', $date\_in); //создаем сложный запрос - запрос от запроса, здесь вы можете увидеть вариант передачи переменных в часть WHERE запроса $query="SELECT DISTINCT `room`.`ID Room`, `room`.`Room number` , `category`.`Category type` FROM `orders`, `room`, `category`  WHERE (`category`.`ID Category`=`room`.`ID Category`) and `room`.`ID Room` not in  (SELECT `orders`.`ID Room` FROM `orders` WHERE MONTH(`orders`.`Check in`)= '".$m\_out."' and ((DAY(`orders`.`Check in`)  BETWEEN '".$d\_in."' and '".$d\_out."') or (DAY(`orders`.`Check out`) BETWEEN '".$d\_in."' and '".$d\_out."')))";  $result = mysqli\_query($db, $query); // вызываем метод для отображения строк //далее обрабатывает запрос построчно и выводим на экран if (!$result) {echo " Sorry, bun there are no free rooms for your date :( ";}if ($result)  { echo "<p align=center><b> F R E E &nbsp; &nbsp; &nbsp; &nbsp; R O O M S &nbsp; &nbsp; :) </b></p>"; $rows = $result->num\_rows; // количество строк echo "<table align=center>";//первая строка таблица - строка заголовков echo "<tr>"; echo "<th>ID</th>"; //имя первого поля echo "<th>N</th>"; //имя второго поля echo "<th>Category</th>"; //имя третьего поля echo "<th></th>"; echo "<th></th>"; echo "</tr>";//последующие строки таблицы - результат запросов for ($i = 0; $i < $rows; ++$i) { $row = $result->fetch\_row(); // получение строки из массива результирующего запроса  echo "<tr>"; for ($j = 0; $j < count($row); $j++)  { echo "<td align=center>$row[$j]</td>"; } //на следующей строчке приведен пример передачи переменных через адресную строку обработчику // book.php, который получает их в виде $\_GET['id\_room']  echo "<td><a href=\"book.php?id\_room=$row[0] & d\_i=$date\_in\_ & d\_out=$date\_out\_\" > BOOK </a></td>";  echo "</tr>"; } echo "</table>";}$db->close();?> |

Результатом будет следующая страница



Выбрав интересующий нас номер, т.е. нажав на BOOK, попадаем на страницу book.php, на которой идет дальнейшая обработка данных

|  |
| --- |
| <?php// обработка запроса POST, ($\_POST - ассоциативный массив) if (isset($\_GET['id\_room']) && isset($\_GET['d\_i']) && isset($\_GET['d\_out'])) { //получили данные и далее работаете с ними echo $\_GET['d\_out'] ; echo "<br>"; echo $\_GET['d\_i'] ; echo "<br>"; echo $\_GET['id\_room']; echo "<br>"; }else echo "not find";?> |