#### Санкт-Петербургский Государственный Политехнический Университет

Институт прикладной математики и механики

Кафедра «Прикладная математика»

# 

# 

# 

# Отчет по курсу «базы данных»

##### 

##### 

##### 

##### 

##### 

##### 

##### 

##### 

##### 

##### 

##### Студент: Цветков А.Д.

##### Группа: 23631/1

##### Преподаватель: Крашенинников С.В.

## Формулировка задания

#### Исходное задание (вариант 10 – картотека уголовного розыска):

Реализовать базу данных для картотеки уголовного розыска, содержащую в себе уголовный кодекс (номер статьи, статья, наказание, наказание в случае рецидива), список попавшихся в течение дня (ФИО, отпечатки пальцев (ID), дата ареста, статья), архив (история “ходок” всех клиентов) и список федерального розыска.

Требуется поддержка:

* Оформление задержанного
* Сличение по ID со списком федерального розыска и коррекция федерального списка в случае успеха
* Архив (преступники могут иметь несколько разных ФИО для одного ID)
* Вынесение приговора (в тюрьму, в федеральную тюрьму (если проходит по федеральным спискам), штраф и на свободу, расстрел)
* Поиск по архиву (по ФИО, по ID)

Отчёты: Архив (данные за последний месяц, год), Федеральный розыск (текущее состояние), список отпускаемых на свободу (у кого в ближайший месяц закончится срок), статистика преступлений по статьям и сезонам.

#### Дополнительные соглашения предметной области:

* Статистика представляет собой количество преступлений по определенной статье за сезоны: весна, лето, осень, зима, курортный сезон (с 01.07 по 14.09), новогодние праздники (с 01.01 по 08.01);
* Если преступник был расстрелян, оформить преступника с таким же ID нельзя;
* Если пойманный преступник числится в федеральном списке и должен быть отправлен в тюрьму (по статье при задержании или по статье(-ям) из федерального списка), он отправляется в федеральную тюрьму;
* Пользователь вручную задает дату ареста. Оформление преступника с датой, которая еще не наступила, не допускается;
* Изменение существующих записей в архиве не допускается;
* Допускается совершение преступления преступником, отбывающим наказание в тюрьме;
* Рецидив – повторное и более задержание независимо от статьи (если ID преступника есть в архиве, ему присваивается наказание, предусмотренное при рецидиве);
* Уголовный кодекс содержит 361 статью, ID преступника не может быть длиннее 8 символов, значение отчества может быть пустым, длина фамилии и имени не должна превышать 50 символов;
* Если преступник получил наказание, находясь в тюрьме, сроки отбывания наказания суммируются;
* При выводе статистики настоящим именем преступника считается то, с которым он был оформлен первый раз;
* Срок заключения выражается в годах, размер штрафа – в рублях;
* В федеральном списке преступник может встречаться несколько раз.

#### 

#### Итоговое задание:

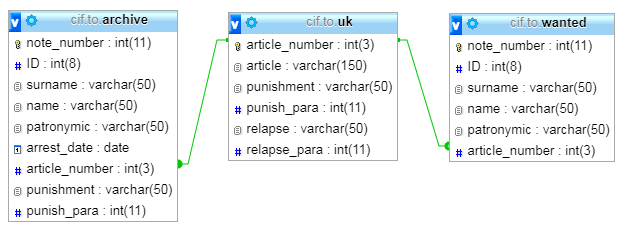
* Спроектировать и реализовать информационную систему, соответствующую исходному заданию с учетом дополнительных соглашений предметной области
* Добавить поддержку стандартных операций:
  1. Поиск записи по ключевому полю
  2. Поиск записи по не ключевому полю
  3. Поиск записи по маске
  4. Добавление записи
  5. Добавление группы записей
  6. Изменение записи (определение изменяемой записи по ключевому полю)
  7. Изменение записи (определение изменяемой записи по не ключевому полю)
  8. Удаление записи (определение удаляемой записи по ключевому полю)
  9. Удаление записи (определение удаляемой записи по не ключевому полю)
  10. Удаление группы записей
  11. Сжатие базы данных (после удаления из БД 200 строк)
  12. Сжатие базы данных (после удаления, в результате которого в БД остается 200 строк)
* Замерить время выполнения стандартных операций для таблиц, содержащих 1000, 10000, 100000 записей.
* Сделать выводы.

## 

## Используемый программный инструментарий

* Локальный сервер Denwer для работы с PHP и SQL:
  1. PHP 5.3.13
  2. MySQL 5.5
  3. Apache 2.2.22
  4. phpMyAdmin 3.5.1
* Текстовый редактор Notepad++ v7.6 для работы с PHP-файлами

***Схема базы данных***



*Таблица archive (архив):*

* **note\_number** – первичный ключ, является номером записи в таблице, AUTO\_INCREMENT
* **ID** – отпечатки преступника
* **surname** – фамилия преступника
* **name** – имя преступника
* **patronymic** – отчество преступника (по умолчанию NULL)
* **arrest\_date** – дата ареста/оформления
* **article\_number** – номер статьи в уголовном кодексе, внешний ключ, ссылается на поле **article\_number** таблицы *uk*
* **punishment** – вид наказания (тюрьма, федеральная тюрьма, штраф, расстрел)
* **punish\_para** – параметр наказания (рубли/года)

*Таблица uk (уголовный кодекс):*

* **article\_number** – первичный ключ, номер статьи
* **article** – название статьи
* **punishment** – вид наказания (тюрьма, штраф, расстрел)
* **punish\_para** – параметр наказания (рубли/года)
* **relapse** – вид наказания в случае рецидива
* **relapse\_para** – параметр наказания в случае рецидива

*Таблица wanted (федеральный розыск):*

* **note\_number** – первичный ключ, является номером записи в таблице, AUTO\_INCREMENT
* **ID** – отпечатки преступника
* **surname** – фамилия преступника
* **name** – имя преступника
* **patronymic** – отчество преступника (по умолчанию NULL)
* **article\_number** – номер статьи в уголовном кодексе, внешний ключ, ссылается на поле **article\_number** таблицы *uk*

***Основные запросы предметной области***

* *Поиск по архиву (по ФИО):*

<form method="post" action="archive.php">

<i> Введите ФИО для поиска: </i>

</br>

</br>

<div>

<label for="surname">Фамилия</label>

<input type="text" name="surname" required>

<label for="name">Имя</label>

<input type="text" name="name" required>

<label for="patronymic">Отчество</label>

<input type="text" name="patronymic">

</br>

</br>

<button type="submit" name = "find1">Найти!</button>

<div>

<?php

if(isset($\_POST['find1']))

{

$surname1 = $\_POST['surname'];

$name1 = $\_POST['name'];

$patronymic1 = $\_POST['patronymic'];

if ($patronymic1 == NULL)

{

$query ="SELECT \* FROM archive WHERE surname = '$surname1' AND name = '$name1' ORDER BY arrest\_date DESC";

}

else

{

$query = "SELECT \* FROM archive WHERE surname = '$surname1' AND name = '$name1' AND patronymic = '$patronymic1' ORDER BY arrest\_date DESC";

}

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$num\_rows = mysqli\_num\_rows($result);

if ($num\_rows == NULL)

{

echo "<i> <font color=\"red\"> К сожалению, ничего не нашлось. </font> </i>";

}

else {

//вывод на экран}

}

* *Поиск по архиву (по ID):*

<form method="post" action="archive.php">

<i> Введите ID для поиска: </i>

</br>

</br>

<label for="ID">ID задержанного</label>

<input type="number" maxlength="8" min="1" max="99999999" name="ID" required>

</div>

</br>

<button type="submit" name = "find2">Найти!</button>

</br>

<?php

if(isset($\_POST['find2']))

{

$ID1 = $\_POST['ID'];

$query = "SELECT \* FROM archive WHERE ID = '$ID1' ORDER BY arrest\_date DESC";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$num\_rows = mysqli\_num\_rows($result);

if ($num\_rows == NULL)

{

echo "<i> <font color=\"red\"> К сожалению, ничего не нашлось. </font> </i>";

}

else

{

//вывод на экран }

}

* *Просмотр архива за определенный период*

<form method="post" action="archive.php">

<div>

<label for="searchtype">Парметры отображения</label>

<select name="searchtype">

<option>Выберите тип поиска</option>

<option value="lastmonth"> Последний месяц </option>

<option value="lastyear">Последний год</option>

</select>

</div>

</br>

<button type="submit" name = "show">Показать!</button>

<?php

if(isset($\_POST['show']))

{

$name = $\_POST['searchtype'];

echo "</br>";

if ($name == "lastmonth")

{

$query ="SELECT \* FROM archive WHERE arrest\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 30 DAY) ORDER BY arrest\_date DESC";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$num\_rows = mysqli\_num\_rows($result);

if ($num\_rows == NULL)

{

echo "<i> <font color=\"red\"> К сожалению, ничего не нашлось. </font> </i>";

}

else

{ //вывод на экран}

if ($name == "lastyear")

{

$query ="SELECT \* FROM archive WHERE arrest\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 365 DAY) ORDER BY arrest\_date DESC";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$num\_rows = mysqli\_num\_rows($result);

if ($num\_rows == NULL)

{

echo "<i> <font color=\"red\"> К сожалению, ничего не нашлось. </font> </i>";

}

else

{ //вывод на экран}

}

}

* *Запрос для просмотра текущего состояния федерального розыска*

$query ="SELECT wanted.id, wanted.surname, wanted.name, wanted.patronymic, wanted.article\_number, uk.article

FROM wanted

INNER JOIN uk

ON uk.article\_number = wanted.article\_number

ORDER BY surname, name, patronymic";

* *Оформление задержанного*

<form method="post" action="reg.php">

<div>

<label for="surname">Фамилия</label>

<input type="text" name="surname" required placeholder="Иванов" maxlength = "50">

<label for="name">Имя</label>

<input type="text" name="name" required placeholder="Иван" maxlength = "50">

<label for="patronymic">Отчество</label>

<input type="text" name="patronymic" placeholder="не обязательно" maxlength = "50">

</br>

<div>

</br>

<label for="date">Дата ареста</label>

<input type="date" name="date" required>

</div>

</br>

<div>

<label for="ID">ID задержанного</label>

<input type="number" maxlength="8" min="1" max="99999999" name="ID" required>

</div>

</br>

<div>

<label for="clause">№ статьи </label>

<input type="number" maxlength="3" min="1" max="361" name="clause" required>

</div>

</br>

<button type="submit" name = "add">Добавить</button>

<button type="reset">Сбросить</button>

</form>

<?php

if (isset($\_POST['add']))

{

$surname = $\_POST['surname'];

$name = $\_POST['name'];

$patronymic = $\_POST['patronymic'];

$date = $\_POST['date'];

$ID = $\_POST['ID'];

$st\_num = $\_POST['clause'];

$date1 = date('Y-m-d');

if ($date > $date1) //проверка на корректность даты

{

echo "<i> <font color=\"red\">Произошла ошибка: дата не может быть больше текущей </font> </i>";

}

else

{

$query = "SELECT \* FROM archive WHERE id = '$ID' AND punishment = 'расстрел'"; //проверка, жив ли преступник

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$num\_rows = mysqli\_num\_rows($result);

$query1 = "SELECT \* FROM archive WHERE id = '$ID'"; //запрос на наличие преступника в архиве

$result1 = mysqli\_query($link, $query1) or die("Ошибка " . mysqli\_error($link));

$num\_rows1 = mysqli\_num\_rows($result1);

$query2 = "SELECT \* FROM wanted WHERE id = '$ID'"; //запрос на наличие преступника в федеральном списке

$result2 = mysqli\_query($link, $query2) or die("Ошибка " . mysqli\_error($link));

$num\_rows2 = mysqli\_num\_rows($result2);

if ($num\_rows !=NULL)

{

if ($num\_rows2!=NULL)

{

$querydel = "DELETE FROM wanted WHERE id = '$ID'";

mysqli\_query($link, $querydel) or die("Ошибка " . mysqli\_error($link));

}

echo "<i> <font color=\"red\"> Произошла ошибка: попавшийся был расстрелян </font> </i>";

}

else

{

if (($num\_rows1 == NULL) && ($num\_rows2 == NULL)) //если нет ни в архиве, ни в федеральном списке

{

$query3 = "SELECT \* FROM uk WHERE article\_number = '$st\_num'"; //запрос для определения наказания

$result3 = mysqli\_query($link, $query3) or die("Ошибка " . mysqli\_error($link));

$row3 = mysqli\_fetch\_assoc($result3);

$punishment3 = $row3['punishment'];

$punish\_para3 = $row3['punish\_para'];

mysqli\_query($link, "INSERT INTO archive (ID, surname, name, patronymic, arrest\_date, article\_number, punishment, punish\_para) VALUES ('$ID', '$surname', '$name', '$patronymic', '$date', '$st\_num', '$punishment3', '$punish\_para3')");

echo "</br></br>";

echo "<i> <font color=\"green\"> Успешно! Приговор: </font> </i>";

if ($punishment3 == 'расстрел') echo "<i> <font color=\"green\"> расстрел. </font> </i>";

if ($punishment3 == 'тюрьма') echo "<i> <font color=\"green\"> в тюрьму. Количество лет: $punish\_para3 </font> </i>";

if ($punishment3 == 'штраф') echo "<i> <font color=\"green\"> штраф в размере $punish\_para3 рублей.</font> </i>";

echo "</br></br>";

}

if (($num\_rows1 != NULL) && ($num\_rows2 == NULL)) //если есть в архиве, но нет в федеральном списке

{

$query3 = "SELECT \* FROM uk WHERE article\_number = '$st\_num'";

$result3 = mysqli\_query($link, $query3) or die("Ошибка " . mysqli\_error($link));

$row3 = mysqli\_fetch\_assoc($result3);

$punishment3 = $row3['relapse']; //наказание как рецидив

$punish\_para3 = $row3['relapse\_para'];

mysqli\_query($link, "INSERT INTO archive (ID, surname, name, patronymic, arrest\_date, article\_number, punishment, punish\_para) VALUES ('$ID', '$surname', '$name', '$patronymic', '$date', '$st\_num', '$punishment3', '$punish\_para3')");

echo "</br></br>";

echo "<i> <font color=\"green\"> Успешно! Приговор: </font> </i>";

if ($punishment3 == 'расстрел') echo "<i> <font color=\"green\"> расстрел. </font> </i>";

if ($punishment3 == 'тюрьма') echo "<i> <font color=\"green\"> в тюрьму. Количество лет: $punish\_para3 </font> </i>";

if ($punishment3 == 'штраф') echo "<i> <font color=\"green\"> штраф в размере $punish\_para3 рублей.</font> </i>";

echo "</br></br>";

}

if (($num\_rows1 == NULL) && ($num\_rows2 != NULL)) //если нет в архиве, но есть в федеральном списке

{

$prison\_count = 0;

$rustrell\_count = 0; //счетчики для вынесения приговора

$tax\_count = 0;

$query3 = "SELECT \* FROM uk WHERE article\_number = '$st\_num'";

$result3 = mysqli\_query($link, $query3) or die("Ошибка " . mysqli\_error($link));

$row3 = mysqli\_fetch\_assoc($result3);

$punishment3 = $row3['punishment'];

$punish\_para3 = $row3['punish\_para'];

if ($punishment3 == 'тюрьма')

{

$punishment3 = 'федеральная тюрьма';

$prison\_count += $punish\_para3;

}

if ($punishment3 == 'штраф') $tax\_count+= $punish\_para3;

if ($punishment3 == 'расстрел') $rustrell\_count = 1;

mysqli\_query($link, "INSERT INTO archive (ID, surname, name, patronymic, arrest\_date, article\_number, punishment, punish\_para) VALUES ('$ID', '$surname', '$name', '$patronymic', '$date', '$st\_num', '$punishment3', '$punish\_para3')");

while ($row2 = mysqli\_fetch\_assoc($result2)) //поиск по ID в федеральном списке, в цикле добавляем все записи в архив

{

$surname2 = $row2['surname'];

$name2 = $row2['name'];

$patronymic2 = $row2['patronymic'];

$st\_num2 = $row2['article\_number'];

$query3 = "SELECT \* FROM uk WHERE article\_number = '$st\_num2'";

$result3 = mysqli\_query($link, $query3) or die("Ошибка " . mysqli\_error($link));

$row3 = mysqli\_fetch\_assoc($result3);

$punishment3 = $row3['punishment'];

$punish\_para3 = $row3['punish\_para'];

if ($punishment3 == 'тюрьма')

{

$punishment3 = 'федеральная тюрьма'; //поскольку проходит по федеральным спискам

$prison\_count += $punish\_para3;

}

if ($punishment3 == 'штраф') $tax\_count+= $punish\_para3;

if ($punishment3 == 'расстрел') $rustrell\_count = 1;

mysqli\_query($link, "INSERT INTO archive (ID, surname, name, patronymic, arrest\_date, article\_number, punishment, punish\_para) VALUES ('$ID', '$surname2', '$name2', '$patronymic2', '$date', '$st\_num2', '$punishment3', '$punish\_para3')");

}

echo "</br></br>";

echo "<i> <font color=\"green\"> Успешно! Приговор: </font> </i>";

if ($rustrell\_count!= NULL) echo "<i> <font color=\"green\"> Расстрел. </font> </i>";

else

{

if ($prison\_count != NULL) echo "<i> <font color=\"green\"> В федеральную тюрьму. Количество лет: $prison\_count </font> </i> </br>";

}

if ($tax\_count != NULL) echo "<i> <font color=\"green\"> Штраф в размере $tax\_count рублей.</font> </i>";

echo "</br></br>";

$querydel = "DELETE FROM wanted WHERE id = '$ID'";

mysqli\_query($link, $querydel) or die("Ошибка " . mysqli\_error($link));

}

if (($num\_rows1!= NULL) && ($num\_rows2 != NULL)) //если есть в федеральном списке и в архиве

{

$prison\_count = 0;

$rustrell\_count = 0;

$tax\_count = 0;

$query3 = "SELECT \* FROM uk WHERE article\_number = '$st\_num'";

$result3 = mysqli\_query($link, $query3) or die("Ошибка " . mysqli\_error($link));

$row3 = mysqli\_fetch\_assoc($result3);

$punishment3 = $row3['relapse'];

$punish\_para3 = $row3['relapse\_para']; //оформляем как рецидив

if ($punishment3 == 'тюрьма')

{

$punishment3 = 'федеральная тюрьма';

$prison\_count += $punish\_para3;

}

if ($punishment3 == 'штраф') $tax\_count+= $punish\_para3;

if ($punishment3 == 'расстрел') $rustrell\_count = 1;

mysqli\_query($link, "INSERT INTO archive (ID, surname, name, patronymic, arrest\_date, article\_number, punishment, punish\_para) VALUES ('$ID', '$surname', '$name', '$patronymic', '$date', '$st\_num', '$punishment3', '$punish\_para3')");

while ($row2 = mysqli\_fetch\_assoc($result2)) //поиск по ID в федеральном списке

{

$surname2 = $row2['surname'];

$name2 = $row2['name'];

$patronymic2 = $row2['patronymic'];

$st\_num2 = $row2['article\_number'];

$query3 = "SELECT \* FROM uk WHERE article\_number = '$st\_num2'";

$result3 = mysqli\_query($link, $query3) or die("Ошибка " . mysqli\_error($link));

$row3 = mysqli\_fetch\_assoc($result3);

$punishment3 = $row3['relapse'];

$punish\_para3 = $row3['relapse\_para'];

if ($punishment3 == 'тюрьма')

{

$punishment3 = 'федеральная тюрьма';

$prison\_count += $punish\_para3;

}

if ($punishment3 == 'штраф') $tax\_count+= $punish\_para3;

if ($punishment3 == 'расстрел') $rustrell\_count = 1;

mysqli\_query($link, "INSERT INTO archive (ID, surname, name, patronymic, arrest\_date, article\_number, punishment, punish\_para) VALUES ('$ID', '$surname2', '$name2', '$patronymic2', '$date', '$st\_num2', '$punishment3', '$punish\_para3')");

}

echo "</br></br>";

echo "<i> <font color=\"green\"> Успешно! Приговор: </font> </i>";

if ($rustrell\_count!= NULL) echo "<i> <font color=\"green\"> Расстрел. </font> </i>";

else

{

if ($prison\_count != NULL) echo "<i> <font color=\"green\"> В федеральную тюрьму. Количество лет: $prison\_count </font> </i> </br>";

}

if ($tax\_count != NULL) echo "<i> <font color=\"green\"> Штраф в размере $tax\_count рублей.</font> </i>";

echo "</br></br>";

$querydel = "DELETE FROM wanted WHERE id = '$ID'";

mysqli\_query($link, $querydel) or die("Ошибка " . mysqli\_error($link));

}

}

}

}

?>

* *Запрос для отображения попавшихся в течение дня $date:*

$query = "SELECT archive.id, archive.surname, archive.name, archive.patronymic, archive.arrest\_date, archive.article\_number, archive.punishment, archive.punish\_para, uk.article

FROM archive

INNER JOIN uk

ON uk.article\_number = archive.article\_number WHERE arrest\_date = '$date'

ORDER BY archive.surname, archive.name, archive.patronymic";

* *Список отпускаемых на свободу*

<form method="post" action="report.php">

<button type="submit" name = "free">Показать список отпускаемых на свободу</button>

<?php

if (isset($\_POST['free']))

{

$query ="SELECT \* FROM

(SELECT \* FROM ARCHIVE WHERE ID NOT IN

(SELECT ARCHIVE.ID FROM ARCHIVE WHERE punishment = 'расстрел')) AS NORUSTRELL

WHERE punishment = 'тюрьма' OR punishment = 'федеральная тюрьма' ORDER BY id, arrest\_date"; //отсеивает расстрелянных, выбирает только тех, кто сидит/сидел в тюрьме

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$num\_rows = mysqli\_num\_rows($result);

if ($num\_rows == NULL)

{

echo "</br></br>";

echo "<i> <font color=\"red\"> К сожалению, ничего не нашлось. </font> </i>";

echo "</br>";

}

else

{

$table = "CREATE TABLE free

(

num INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

ID INT(8) NOT NULL,

surname VARCHAR(50) NOT NULL,

name VARCHAR(50) NOT NULL,

patronymic VARCHAR(50) DEFAULT NULL,

free\_date DATE NOT NULL

)"; //создаем новую таблицу, чтобы записывать в нее подходящие значения

$tableres = mysqli\_query($link, $table) or die("Ошибка " . mysqli\_error($link));

$row = mysqli\_fetch\_assoc($result);

while ($row != NULL) //пробегаемся по всем строчкам

{

$ID = $row['ID'];

$surname = $row['surname']; //запоминаем имя, с которым преступник попался первый раз

$name = $row['name'];

$patronymic = $row['patronymic'];

$realdate = NULL; //показатель реальной даты начала отсидки

$realsrok = NULL; //реальный срок

$currentdate = NULL; //текущая считанная дата

$currentsrok = NULL; //текущий считанный срок

$n = 0;

do

{

$currentdate = $row['arrest\_date'];

$currentsrok = $row['punish\_para'];

if ($realsrok == NULL) $realsrok = $currentsrok; //если первая строчка, заполняем

if ($realdate == NULL) $realdate = $currentdate;

if ($n!=0)

{

if (strtotime("$realdate + $realsrok year") <= strtotime("$currentdate")) //если условие выполняется, преступник уже вышел до даты следующей отсидки

{

$realdate = $currentdate;

$realsrok = $currentsrok;

}

else $realsrok+=$currentsrok; //иначе плюсуем срок

}

$n++;

$row = mysqli\_fetch\_assoc($result);

} while ($row['ID'] == $ID);

$free\_datesec = strtotime("$realdate + $realsrok year");

if (($free\_datesec > strtotime("now")) && ($free\_datesec<= strtotime("now+1 month"))) //если дата выхода находится в интервале от сегодняшнего дня до сегодня + месяц

{

$free\_date = date("Y-m-d", $free\_datesec);

$filltable = "INSERT INTO free (ID, surname, name, patronymic, free\_date) VALUES ('$ID', '$surname', '$name', '$patronymic', '$free\_date')";

$filltableres = mysqli\_query($link, $filltable) or die("Ошибка " . mysqli\_error($link));

}

}

$query1 = "SELECT \* FROM free ORDER BY free\_date"; //выводим получившуюся таблицу с сортировкой по дате

$result1 = mysqli\_query($link, $query1) or die("Ошибка " . mysqli\_error($link));

$num\_rows1 = mysqli\_num\_rows($result1);

if ($num\_rows1 == NULL)

{

echo "</br></br>";

echo "<i> <font color=\"red\"> К сожалению, ничего не нашлось. </font> </i>";

echo "</br>";

}

else

{

//выводим на экран

}

$query2 = "DROP TABLE free"; //удаляем таблицу

$result2 = mysqli\_query($link, $query2) or die("Ошибка " . mysqli\_error($link));

}

}

?>

</form>

* *Запросы для составления статистики за год (аналогично для всего времени, без запроса “selected”)*

$query = "SELECT \* FROM (SELECT \* FROM archive WHERE arrest\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 365 DAY)) AS selected WHERE article\_number = '$st\_num' AND (arrest\_date LIKE '%-03-%' OR arrest\_date LIKE '%-04-%' OR arrest\_date LIKE '%-05-%')"; //весна

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$num\_rows = mysqli\_num\_rows($result); //считаем количество строк == количество преступлений

$query = "SELECT \* FROM (SELECT \* FROM archive WHERE arrest\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 365 DAY)) AS selected WHERE article\_number = '$st\_num' AND MONTH(arrest\_date)>=6 AND MONTH(arrest\_date)<=8"; //лето

$query = "SELECT \* FROM (SELECT \* FROM archive WHERE arrest\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 365 DAY)) AS selected WHERE article\_number = '$st\_num' AND MONTH(arrest\_date)>=9 AND MONTH(arrest\_date)<=11"; //осень

$query = "SELECT \* FROM (SELECT \* FROM archive WHERE arrest\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 365 DAY)) AS selected WHERE article\_number = '$st\_num' AND (arrest\_date LIKE '%-12-%' OR arrest\_date LIKE '%-01-%' OR arrest\_date LIKE '%-02-%')"; //зима

$query = "SELECT \* FROM (SELECT \* FROM archive WHERE arrest\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 365 DAY)) AS selected WHERE article\_number = '$st\_num' AND (EXTRACT(MONTH FROM arrest\_date)\*100+EXTRACT(DAY FROM arrest\_date)) >= 701 AND (EXTRACT(MONTH FROM arrest\_date)\*100+EXTRACT(DAY FROM arrest\_date))<=914"; //курортный сезон

$query = "SELECT \* FROM (SELECT \* FROM archive WHERE arrest\_date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 365 DAY)) AS selected WHERE article\_number = '$st\_num' AND (EXTRACT(MONTH FROM arrest\_date)\*100+EXTRACT(DAY FROM arrest\_date)) >= 101 AND (EXTRACT(MONTH FROM arrest\_date)\*100+EXTRACT(DAY FROM arrest\_date))<=108"; //новогодние праздники

***Реализация стандартных операций***

*\*table – одна из трех таблиц, содержащих 1000, 10000 и 100000 записей  
\*$value – количество записей в текущей таблице*

*\*функция random\_string возвращает рандомную строку заданной длины*

##### Поиск по ключевому полю

##### $time = NULL;

##### for ($i = 0; $i < 100; $i++)

##### {

##### $id = rand(1, $value);

##### $time\_start = microtime(TRUE);

##### $query = "SELECT \* FROM table WHERE id = $id;";

##### $result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

##### $time\_end = microtime(TRUE);

##### $time += ($time\_end - $time\_start);

##### }

##### $time /= 100; //усредняем результат

##### echo "</br> <i> $value: </i>";

##### echo "$time";

##### Поиск по не ключевому полю

##### $time = NULL;

##### for ($i = 0; $i < 100; $i++)

##### {

##### $num = rand(1, 10000); //в значении no1 лежит число от 1 до 10000

##### $time\_start = microtime(TRUE);

##### $query = "SELECT \* FROM table WHERE no1 = $num;";

##### $result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

##### $time\_end = microtime(TRUE);

##### $time += ($time\_end - $time\_start);

##### }

##### $time /= 100; //усредняем время

##### echo "</br> <i> $value: </i>";

##### echo $time;

##### Поиск по маске

$time = NULL;

for ($i = 0; $i < 100; $i++)

{

$mask = random\_string(5);

$time\_start = microtime(TRUE);

$query = "SELECT \* FROM table WHERE text LIKE '%$mask%'";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time\_end = microtime(TRUE);

$time += ($time\_end - $time\_start);

}

$time /= 100; //усредняем результат

echo "</br> <i> $value: </i>";

echo $time;

##### Добавление записи

$time = NULL;

$num1 = rand(1, 10000);

$num2 = rand(1, 10000);

$str = random\_string(15); //длина строки в text

for ($i = 0; $i < 100; $i++)

{

$time\_start = microtime(TRUE);

$query = "INSERT INTO table VALUES (NULL, '$str', '$num1', '$num2')";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time\_end = microtime(TRUE);

$time += ($time\_end - $time\_start);

}

$time /= 100; //усредняем время

echo "</br> <i> $value: </i>";

echo $time;

##### Добавление группы записей

//добавляем 100 записей

$time=NULL;

$time\_start = microtime(TRUE);

for ($i = 0; $i < 100; $i++)

{

$num1 = rand(1, 10000);

$num2 = rand(1, 10000);

$str = random\_string(15);

$query = "INSERT INTO table VALUES (NULL, '$str', '$num1', '$num2')";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

}

$time\_end = microtime(TRUE);

$time = ($time\_end - $time\_start);

echo "</br> <i> $value: </i>";

echo $time;

##### Изменение записи по ключевому полю

$time=NULL;

for ($i = 0; $i < 100; $i++)

{

$num1 = rand(1, 10000);

$num2 = rand(1, 10000);

$str = random\_string(15);

$id = rand(1, $value);

$time\_start = microtime(TRUE);

$query = "UPDATE table SET text = '$str', no1 = '$num1', no2 = '$num2' WHERE id = $id";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time\_end = microtime(TRUE);

$time += ($time\_end - $time\_start);

}

$time /= 100;

echo "</br> <i> $value: </i>";

echo $time;

##### Изменение записи по не ключевому полю

$time=NULL;

for ($i = 0; $i < 100; $i++)

{

$num1 = rand(1, 10000);

$num2 = rand(1, 10000);

$str = random\_string(15);

$id = rand(1, $value);

$time\_start = microtime(TRUE);

$query = "UPDATE table SET text = '$str', no1 = '$num1', no2 = '$num2' WHERE no1 = $num1";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time\_end = microtime(TRUE);

$time += ($time\_end - $time\_start);

}

$time /= 100;

echo "</br> <i> $value: </i>";

echo $time;

##### Удаление записи по ключевому полю

$mysqli = new mysqli("localhost", "root", "", "test");

$time = NULL;

$num1 = rand(1, 10000);

$num2 = rand(1, 10000);

$str = random\_string(15);

for ($i = 0; $i < 100; $i++)

{

$mysqli->query("INSERT INTO table VALUES (NULL, '$str', '$num1', '$num2')");

$id = mysqli\_insert\_id($mysqli); //запоминаем id вставленной строки, чтобы потом ее удалить

$time\_start = microtime(TRUE);

$mysqli->query("DELETE FROM table WHERE id = $id");

$time\_end = microtime(TRUE);

$time += ($time\_end - $time\_start);

}

$time /= 100;

echo "</br> <i> $value: </i>";

echo $time;

##### Удаление записи по не ключевому полю

$time = NULL;

$num1 = rand(1, 10000);

$num2 = rand(1, 10000);

$str = random\_string(15);

for ($i = 0; $i < 100; $i++)

{

$query = "INSERT INTO table VALUES (NULL, '$str', '$num1', '$num2')";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time\_start = microtime(TRUE);

$query = "DELETE FROM table WHERE text = '$str'";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time\_end = microtime(TRUE);

$time += ($time\_end - $time\_start);

}

$time /= 100;

echo "</br> <i> $value: </i>";

echo $time;

##### Удаление группы записей (200 строк)

$time = NULL;

for ($i = 301; $i < 501; $i++)

{

$time\_start = microtime(TRUE);

$query = "DELETE FROM table WHERE id=$i";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time\_end = microtime(TRUE);

$time += ($time\_end - $time\_start);

}

echo "</br> <i> $value: </i>";

echo $time;

##### Сжатие базы данных (после удаления 200 строк)

$time = NULL;

$start = microtime(true);

$query = "OPTIMIZE TABLE table";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time = (microtime(true) - $start);

echo "</br> <i> $value: </i>";

echo $time;

##### Сжатие базы данных (после того, как в базе осталось 200 строк)

$time=NULL;

$query = "DELETE FROM table WHERE id < 500 OR id >= 700";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$start = microtime(true);

$query = "OPTIMIZE TABLE table";

$result = mysqli\_query($link, $query) or die("Ошибка " . mysqli\_error($link));

$time = (microtime(true) - $start);

echo "</br> <i> $value: </i>";

echo $time;

## Сравнение временных затрат при реализации стандартных операций

|  |  |  |  |
| --- | --- | --- | --- |
| Число записей | 1000 | 10000 | 100000 |
| Поиск по ключевому полю | 0.00016802787780762 | 0.0001811408996582 | 0.00019845724105835 |
| Поиск по не ключевому полю | 0.00071598768234253 | 0.0047337198257446 | 0.045487468242645 |
| Поиск по маске | 0.0014071154594421 | 0.0092597556114197 | 0.092726683616638 |
| Добавление записи | 0.035849208831787 | 0.032016913890839 | 0.040334143638611 |
| Добавление группы записей | 2.8819501399994 | 2.995316028595 | 2.9813828468323 |
| Изменение записи (определение изменяемой записи по ключевому полю) | 0.024463918209076 | 0.025733873844147 | 0.025272123813629 |
| Изменение записи (определение изменяемой записи по не ключевому полю) | 0.023272418975836 | 0.034905142784119 | 0.11232304811478 |
| Удаление записи (определение удаляемой записи по ключевому полю) | 0.030173079967499 | 0.030814440250397 | 0.031547908782959 |
| Удаление записи (определение удаляемой записи по не ключевому полю) | 0.024585981369019 | 0.045209622383118 | 0.12778278112411 |
| Удаление группы записей (200 строк) | 5.8140592575073 | 5.969313621521 | 5.3879413604736 |
| Сжатие базы данных (после удаления из БД 200 строк) | 0.2546489238739 | 0.46516704559326 | 2.9440541267395 |
| Сжатие базы данных (после удаления, в результате которого в БД остается 200 строк) | 0.21185898780823 | 0.23289084434509 | 0.23236918449402 |

## Выводы

|  |  |  |
| --- | --- | --- |
| Операция | Сложность | Пояснение |
| Поиск по ключевому полю |  | Существует только один результат поиска, поиск в сбалансированном дереве |
| Поиск по не ключевому полю |  | Требуется просмотреть все строки таблицы |
| Поиск по маске |  | Проверяются все строки таблицы |
| Добавление записи |  | Добавление в конец – константа, операция требует заполнения всех полей за фиксированное время |
| Добавление группы записей |  | Добавление в конец таблицы k строк |
| Изменение записи (определение изменяемой записи по ключевому полю) |  | Сначала выполняем поиск по ключу, затем изменяем значение за фиксированное время. |
| Изменение записи (определение изменяемой записи по не ключевому полю) |  | Поиск по не-ключевому полю + константное время изменения |
| Удаление записи (определение удаляемой записи по ключевому полю) |  | Поиск по ключу + удаление за фиксированное время |
| Удаление записи (определение удаляемой записи по не ключевому полю) |  | Время поиска по не ключевому полю + удаление за константу |
| Удаление группы записей |  | Время на поиск одной записи + константа удаления одной записи, умноженные на количество строк |
| Сжатие базы данных (после удаления из БД 200 строк) | Зависит от размера таблицы | Время увеличивается с увеличением количества строк в таблице |
| Сжатие базы данных (после удаления, в результате которого в БД остается 200 строк) |  | Время константное, так как в базе постоянное количество строк |