**Практическая работа № 12.**

**Безопасность SQL в PHP**

В предыдущих уроках, когда мы изучали работу с SQL в PHP, мы не изучали такой важный раздел, как безопасность. Дело в том, что по умолчанию SQL запросы, которые мы делаем через PHP имеют уязвимости, которыми может воспользоваться хакер, чтобы взломать наш сайт.

В данном разделе мы начнем изучать основные проблемы, которые ведут к уязвимостям, а также защиту от них.

Для дальнейшей работы нам нужно сделать ряд таблиц с данными. Эти таблицы будут фигурировать дальше в практических.

Таблица users:

|  |  |  |  |
| --- | --- | --- | --- |
| id | login | password | role |
| 1 | admin | abcde | 1 |
| 2 | user | 12345 | 2 |

Таблица prods:

|  |  |  |
| --- | --- | --- |
| id | name | cost |
| 1 | prod1 | 500 |
| 2 | prod2 | 600 |
| 3 | prod3 | 700 |

Таблица notes:

|  |  |  |
| --- | --- | --- |
| id | text | user\_id |
| 1 | 111 111 111 | 1 |
| 2 | 222 222 222 | 1 |
| 3 | 333 333 333 | 2 |
| 4 | 444 444 444 | 2 |

№1

Создайте описанные таблицы.

Комментарии в SQL запросе в PHP

Для дальнейшей работы вам необходимо знать, как ставятся комментарии в SQL коде. Давайте посмотрим на примере. Пусть у нас есть следующий запрос:

<?php

$query = 'SELECT \* FROM users';

$res = mysqli\_query($link, $query) or die(mysqli\_error($link));

?>

Давайте поставим однострочный комментарий. Это делается с помощью двух дефисов:

<?php

$query = 'SELECT \* FROM users --комментарий';

$res = mysqli\_query($link, $query) or die(mysqli\_error($link));

?>

А теперь сделаем многострочный комментарий:

<?php

$query = 'SELECT \* FROM users /\*

комментарий

\*/';

$res = mysqli\_query($link, $query) or die(mysqli\_error($link));

?>

Кавычки в SQL запросах в PHP

Бывают случаи, когда в SQL запросе мы используем текстовое значение, содержащее внутри себя кавычку. В этом случае мы получим ошибку синтаксиса SQL:

<?php

$query = "SELECT \* FROM users WHERE login='Д'Артаньян'";

$res = mysqli\_query($link, $query)

or die(mysqli\_error($link)); // выведет ошибку

?>

Для решения проблемы нужно эту кавычку экранировать с помощью обратного слеша:

<?php

$query = "SELECT \* FROM users WHERE login='Д\'Артаньян'";

$res = mysqli\_query($link, $query) or die(mysqli\_error($link));

?>

Как правило, однако, текст с кавычкой мы не пишем в явном виде, а берем из переменной. Например, вот так:

<?php

$login = 'Д\'Артаньян';

$query = "SELECT \* FROM users WHERE login='**$login**'";

$res = mysqli\_query($link, $query) or die(mysqli\_error($link));

?>

Также наш текст может прийти из формы, которую заполняет пользователь. В этом случае мы получаем место, потенциально подверженное ошибке. В следующем примере будет ошибка, если пользователь введет данные с кавычкой:

<?php

$login = **$\_POST**['login'];

$query = "SELECT \* FROM users WHERE login='**$login**'";

$res = mysqli\_query($link, $query) or die(mysqli\_error($link));

?>

Для решения проблемы нам необходимо воспользоваться функцией с очень длинным именем: mysqli\_real\_escape\_string. Эта функция сама заэкранирует кавычки в тексте (и некоторые другие проблемные символы тоже):

<?php

$login = mysqli\_real\_escape\_string($link, **$\_POST**['login']);

$query = "SELECT \* FROM users WHERE login='**$login**'";

$res = mysqli\_query($link, $query) or die(mysqli\_error($link));

?>

№1

Пусть у вас есть некоторая таблица articles со статьями. Пусть есть также следующая форма:

<form action="" method="POST">

<input name="title">

<textarea name="text"></textarea>

<input type="submit">

</form>

Убедитесь, что эта форма имеет проблему с кавычками. Избавьтесь от проблемы.