

Задание (2):

1. Определить **сущности** (таблицы), **атрибуты** сущности (характеристики), **связи** между сущностями, таблицы должны быть в **3НФ**.
2. Создать **ER-диаграмму**, определить типы данных.
3. Сгенерировать **SQL-код**
4. Создать **базу данных** в MS SQL Server
5. При помощи запроса **создать таблицы и связи между ними**, используя полученный SQL-код
6. Создать **диаграмму базы данных** в MS SQL Server и сохранить.
7. Если необходимо, то создать **словарь данных**.
8. **Заполнить БД** (перед заполнением привести таблицы в соответствующий вид, т.е. порядок столбцов в .xlsx и в таблице SQL должен быть одинаковым)
9. Перейти к **запросам**

Теоретический материал (шпаргалка)

Для **нормализации таблиц (3НФ)** придерживаться 5 критериев:

- В каждой клеточке таблицы должно быть только одно значение
- Не должно быть повторяющихся строк
- Есть первичный ключ
- Все атрибуты зависят от первичного ключа целиком, а не от какой-то его части
- Все атрибуты зависят только от первичного ключа, но не от других атрибутов

Типы данных:

Значение	Обозначение в MS SQL	Описание
Строка переменной длины	varchar(N)	Можно указать максимально возможную длину строки для соответствующего столбца.
Целое число	int	Данный тип позволяет нам использовать в столбце только целые числа, как положительные, так и отрицательные.
Вещественное число	float	Числа, в которых может присутствовать десятичная точка (запятая).
Дата	date	Если в столбце необходимо хранить только Дату, которая состоит из трех составляющих: Числа, Месяца и Года.
Время	time	Данный тип можно использовать, если в столбце необходимо хранить только данные о времени, т.е. Часы, Минуты, Секунды и Миллисекунды.
Дата и время	datetime	Данный тип позволяет одновременно сохранить и Дату, и Время. Например, 15.02.2014 17:38:31.323
Логический	bit	Данный тип удобно применять для хранения значений вида «Да»/«Нет», где «Да» будет сохраняться как 1 - Истина, а «Нет» будет сохраняться как 0 - Ложь.

В ER-диаграмме указать **Nullable**:

NOT NULL — колонка не может иметь нулевое значение (все должно быть заполнено)

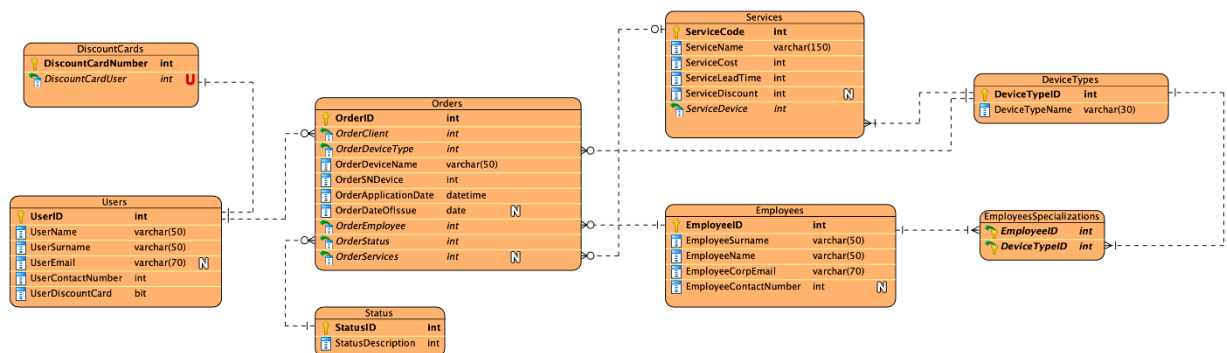
NULL – колонка может иметь нулевое значение (пустое, незаполненное)

Генерация SQL-кода: ПКМ по диаграмме – Utilites – Generate SQL – выбираем в списке MS SQL Server – ОК – копируем код.

Пример словаря данных:

Data Dictionary				
Имя БД				
Role – имя таблицы				
KEY-ключ	FIELD NAME – имя столбца	DATA TYPE / FIELD SIZE – тип данных	REQUIRED? - обязательный	NOTES - заметки
PK	Role	int	Y	IDENTITY
	RoleName	varchar(100)	Y	Имя роли

Пример диаграммы:



Запросы:

SELECT — основной оператор SQL, он нужен для извлечения данных. (то, что выводим)

FROM указывает, откуда эти данные нужно выбрать (таблица, откуда выводим данные)

WHERE уточняет выборку, сформированную с помощью FROM. Он возвращает только те данные, которые соответствуют требованиям запроса

Операторы:

CREATE — создание нового объекта в существующей базе.

ALTER — изменение существующего объекта.

DROP — удаление объекта из базы.

SELECT — позволяет выбрать данные в соответствии с необходимым условием.

INSERT — осуществляют добавление новых данных.

UPDATE — производит замену существующих данных.

DELETE — удаление информации.

ALL	SQL оператор ALL осуществляет вывод всех значений из таблицы.
AND	SQL оператор AND представляет собой получение результата при соблюдении двух поставленных условий.
BETWEEN	SQL оператор BETWEEN. Для этого оператора SQL условия можно установить в определённом диапазоне. Для корректной работы нужно задать минимальное и максимальное значение.
IN	Простой SQL оператор IN указывает, с какими значениями нужно вывести строки.
LIKE	Популярный SQL оператор LIKE. Данный оператор позволяет осуществлять поиск подстроки в тексте и, если подстрока найдена, то она выводится.
NOT	SQL оператор отрицания NOT. Аннулирует любые условия.

OR	SQL оператор «ИЛИ». Дает результат в том случае, когда значение TRUE есть хотя бы в одном из операндов.
----	---

Пример реализации БД средствами PhpMyAdmin (вариант 2 – IT – Сервис)

Разработка БД

По техническому заданию определяем базовые сущности. У нас это будут:

- Пользователь
- Роль пользователя
- Заявка
- Статус заявки

Рассмотрим подробнее каждую сущность:

Сущность	Поле	Тип
user	id	счетчик
	login	строка, уникальный
	password	строка
	fio	строка
	phone	строка
	role_id	идентификатор роли (индекс)

Сущность	Поле	Тип
role	id	счетчик
	name	строка

Сущность	Поле	Тип
order	id	счетчик
	user_id	идентификатор пользователя (индекс)
	oborud	строка
	model	строка
	description	текст
	data	таймстамп
	status_id	идентификатор статуса (индекс)

Сущность	Поле	Тип
status	id	счетчик
	name	строка

Реализуем БД через PhpMyAdmin.

Не забываем, как мы ее назвали в проекте и установить нужную кодировку для русских символов.

В данном случае назвала **servis**, кодировка **utf8mb3_general_ci**

1. Создаем следующие таблицы:

Структура таблицы **user**. Комментарии: поля **login** делаем *уникальными*, а поле **role_id** - *индексом*.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id 🔑	int			Нет	Нем		AUTO_INCREMENT
<input type="checkbox"/> 2	login 🔑	varchar(255)	utf8mb3_general_ci		Нет	Нем		
<input type="checkbox"/> 3	password	varchar(255)	utf8mb3_general_ci		Нет	Нем		
<input type="checkbox"/> 4	fio	varchar(255)	utf8mb3_general_ci		Нет	Нем		
<input type="checkbox"/> 5	phone 🔑	varchar(255)	utf8mb3_general_ci		Нет	Нем		
<input type="checkbox"/> 6	role_id 🔑	int			Нет	1		

Структура таблицы **role**:

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id 🔑	int			Нет	Нем		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(255)	utf8mb3_general_ci		Нет	Нем		

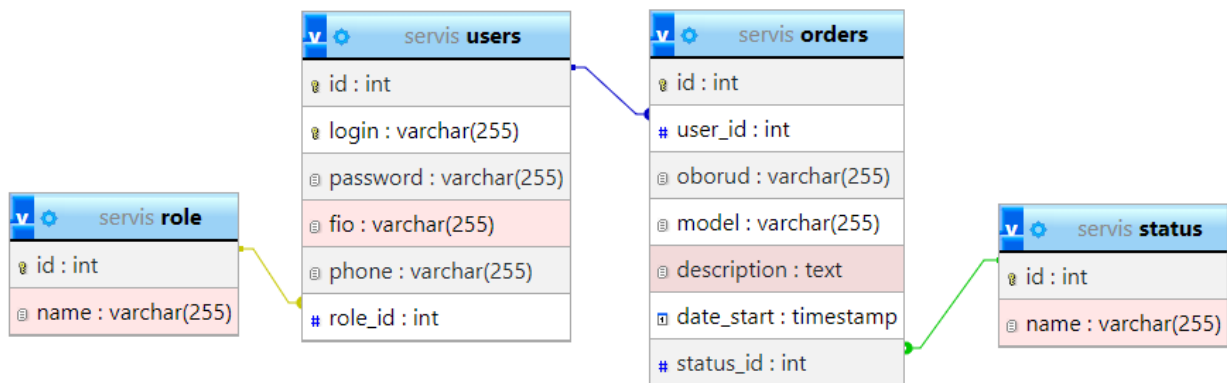
Структура таблицы **order**:

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id 🔑	int			Нет	Нем		AUTO_INCREMENT
<input type="checkbox"/> 2	user_id 🔑	int			Нет	Нем		
<input type="checkbox"/> 3	oborud	varchar(255)	utf8mb3_general_ci		Нет	Нем		
<input type="checkbox"/> 4	model	varchar(255)	utf8mb3_general_ci		Нет	Нем		
<input type="checkbox"/> 5	description	text	utf8mb3_general_ci		Нет	Нем		
<input type="checkbox"/> 6	date_start	timestamp			Да	NULL		
<input type="checkbox"/> 7	status_id 🔑	int			Нет	Нем		

Структура таблицы **status**:

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1	id 🔑	int		Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2	name	varchar(255)	utf8mb3_general_ci	Нет	Нет		

2. Создаем связи между таблицами (через ограничения, либо в **Дизайнере**). Вид из Дизайнера представлен ниже:



3. Заполним таблицу **role** (вкладка **Вставка**):

← T →	id	name
<input type="checkbox"/>	1	Пользователь
<input type="checkbox"/>	2	Администратор
<input type="checkbox"/>	3	Мастер

4. А затем, **user**. При вставке данных не забываем пароль кодировать через функцию **md5**
Обязательно для администратора вносите данные из ТЗ!!!

id	int	<input type="text"/>	<input type="text"/>
username	varchar(255)	<input type="text"/>	<div>capp</div>
password	varchar(255)	<div>MD5</div>	<div>password</div>

Вводим одного пользователя и администратора (*для администратора выдан логин и пароль*)

		id	login	password	fio	phone	role_id
<input type="checkbox"/>		1	user1	24c9e15e52afc47c225b757e7bee1f9d	Иванов Иван Иванович	89277756682	1
<input type="checkbox"/>		2	admin	21232f297a57a5a743894a0e4a801fc3	Смирнов Павел Сергеевич	89228889965	2

Также можно заполнить таблицу со статусами заказов:

		id	name
<input type="checkbox"/>		1	новая
<input type="checkbox"/>		2	в процессе ремонта
<input type="checkbox"/>		3	завершена
<input type="checkbox"/>		4	готов к выдаче
<input type="checkbox"/>		5	ожидание запчастей

Таблицу с заявками будете заполнить через программирование. По умолчанию в структуре таблицы с заявками поставьте статус 1 (новая заявка)

<input type="text" value="status_id"/>	<input type="text" value="INT"/>	<input type="text"/>	<div>Как определено: <input type="text" value="1"/></div>
--	----------------------------------	----------------------	---

Решение задания по оформлению заявок на ремонт оргтехники

1. Переходим в папку **OSPanel – domains**. Организуем свою папку, например, **servis**. Внутри папки организуем папку **css** – для оформления веб – сервиса. Также у нас будут организованы следующие файлы:

- **index.php** – главная страница с формой авторизации
- **db.php** – подключение к базе данных
- **orders.php** – страница с формой оформления заявок
- **orders_user.php** – страница с выводом данных по заявкам из БД
- **logout.php** – страница, осуществляющая выход из аккаунта

Оформление страницы **index.php**

1. Вставляем в текстовом редакторе стандартную заготовку (! + tab). Добавляем форму авторизации:

```
<body>
  <header>
    <h1>IT - сервис</h1>
  </header>
  <main>
    <div class = "wrap"> <p class = "error"></p>
    <h2>Авторизация</h2>
    <table>

      <form action = "" method = "POST">
        <tr><td><label>Введите логин</label></td>
          <td><input type="text" name = "login"></td></tr>
        <tr><td><label>Введите пароль</label></td>
          <td><input type="password" name = "password"></td></tr>
        <tr><td></td><td><button> Авторизоваться </button></td></tr>
      </form>
    </table>
  </div>
</main>
</body>
```

2. Пропишем стили для веб – страницы. В папке **css** организуем файл **main.css**. **Оформляете самостоятельно!!!!**

Примерный вид в браузере:

IT - сервис

Авторизация

Введите логин

Введите пароль

3. В файле **db.php** прописываем подключение к созданной нами БД (в моем случае **servis**)

```
db.php
1  <?php
2  $link = mysqli_connect('localhost', 'root', '', 'servis');
3  if(!$link){
4      die('Ошибка: ' . mysqli_connect_error());
5  }
6  ?>
```

4. Програмируем на странице **index.php**.

- Организуем сессию **session_start()**, чтобы запомнить данные авторизованного пользователя;
- Осуществляем подключение к серверу и к БД с помощью **include_once()**;
- С помощью функции **empty** проверяем поля формы на пустоту, если они заполнены, то заносим значения в переменные **\$login** и **\$password**;
- Далее делаем запрос на выборку строки из таблицы **users** с совпадающими значениями логина и пароля. Итог записываем в переменную **\$user**;
- Далее происходит проверка на совпадение логина и пароля (те, которые у нас уже существуют в таблице БД), если все верно, то организуем сессию и осуществляем переход на страницу **orders.php** (если это пользователь с ролью Пользователь) или на странице **admin.php** (если пользователь с ролью Администратора). В противном случае выведется ошибка (оформление ошибки через css – вывод красным цветом).

```

index.php
1  <?php
2  session_start();
3  include_once("db.php");
4
5  if(!empty($_POST['login']) && !empty($_POST['password'])){
6      $login = $_POST['login'];
7      $password = md5($_POST['password']);
8
9      $result = mysqli_query($link, "SELECT * FROM users WHERE login = '$login' AND password = '$password'");
10     $user = mysqli_fetch_assoc($result);
11
12     if(!empty($user)) {
13         $_SESSION['auth'] = true;
14         $_SESSION['login'] = $user['login'];
15         $_SESSION['id'] = $user['id'];
16         $_SESSION['role_id'] = $user['role_id'];
17         if($_SESSION['role_id'] == '1'){
18             header("Location: orders.php");
19         }
20         elseif($_SESSION['role_id'] == '2'){
21             header("Location: admin.php");
22         }
23     } else {
24         echo "<p class = 'error' id = 'error' align='center'>Неверный логин или пароль</p>";
25     }
26 }
27 ?>

```

Тестируем!!! Все должно работать

Оформление страницы orders.php

1. Вставляем html – шаблон. В меню навигации добавится пункты

```

<nav>
    <a href = "orders.php">Создать заявку</a>
    <a href = "orders_user.php">Все заявки</a>
    <a href = "logout.php">Выход</a></p>
</nav>

```

2. В основном контенте оформляем форму для формирования заявок.

```

<main>
    <h2>Подача заявления</h2>
    <table>
        <form action = "" method = "POST">
            <tr><td><label>Тип оборудования</label></td>
                <td><input type="text" name = "oborud"></td></tr>
            <tr><td><label>Модель</label></td>
                <td><input type="text" name = "model"></td></tr>
            <tr><td><label>Опишите проблемы</label></td>
                <td><textarea name = "description" rows = "5" cols = "50"></textarea></td></tr>
            <tr><td></td><td><button>Подать заявку </button></td></tr>
        </form>
    </table>
</main>

```

В css – файле добавим оформление.

Вид из браузера:

IT - сервис

[Создать заявку](#) [Все заявки](#) [Выход](#)

Подача заявления

Тип оборудования

Модель

Опишите проблемы

3. Програмируем:

- В верхней части стартуем сессию и осуществляем проверку на авторизацию, если пользователь прошел авторизацию, то ему можно приступить к заполнению форму, в противном случае произойдет редирект на **index.php**.

```
1 <?php
2 session_start();
3 if(empty($_SESSION['auth'])){
4     header("Location: index.php");
5 }
6 ?>
```

- После блока **main** прописываем код для отправки данных на сервер в таблицу **orders**:

```
<?php
include_once("db.php");
if(!empty($_POST['oborud']) && !empty($_POST['model']) && !empty($_POST['description'])){
    $oborud = $_POST['oborud'];
    $model = $_POST['model'];
    $description = $_POST['description'];
    $user_id = $_SESSION['id'];

    $result = mysqli_query($link, "INSERT INTO orders(user_id, oborud, model, description)
                                   VALUES ('$user_id', '$oborud', '$model', '$description')");
    if($result == 'true'){
        header("Location: orders_user.php");
    } else
    echo "<p class = 'error'>Информация не добавлена!</p>";
}
?>
```

- В таблице **orders** отредактируйте поле **date_start**, чтобы по умолчанию проставлялось текущее время (если забыли сделать сразу):

<input type="checkbox"/>	6	date_start	timestamp	Да	CURRENT_TIMESTAMP	DEFAULT_GENERATED
--------------------------	---	------------	-----------	----	-------------------	-------------------

Тестируем! Если все выполнено верно, то в таблицу БД добавятся данные.

Оформление страницы orders_user.php

1. Делаем дубль страницы **orders.php**. Изменения будут только в блоке **main**. Вывод данных будем осуществлять в виде карточек (можно оформить вывод в виде таблицы, на ваше усмотрение):

```
<main>
  <h2>Заявки пользователя</h2>

  <?php
    include_once("db.php");
    $result = mysqli_query($link, "SELECT * FROM orders LEFT JOIN users ON orders.user_id = users.id
                                  LEFT JOIN status ON orders.status_id = status.id
                                  WHERE user_id = '$_SESSION[id]'");

    while($row = mysqli_fetch_assoc($result)){
      echo "<div class='cards'>
        <p><b>Дата подачи заявки:</b> $row[date_start]</p>
        <p><b>Тип оборудования:</b> $row[oborud]</p>
        <p><b>Модель:</b> $row[model]</p>
        <p><b>Описание неисправности:</b> $row[description]</p>
        <p><b>Статус:</b> $row[name]</p>
      </div>";
    }
  ?>
</main>
```

В верхней части не забыть скрипт по проверке авторизации!!!

!!! Обратите внимание на сложный запрос, который связывает между собой три таблицы по индексированным полям.

Примерный вид из браузера:

IT - сервис

[Создать заявление](#) [Все заявления](#) [Выход](#)

Заявления пользователя

Дата подачи заявки: 2024-05-27 16:05:47

Тип оборудования: Принтер

Модель: aaa456

Описание неисправности: не захватывает бумагу

Статус: новая

Оформление страницы logout.php

1. Авторизованный пользователь должен иметь возможность осуществить выход из аккаунта. Для этого нужно обнулить сессию и перенаправить на страницу авторизации:

```

logout.php
1  <?php
2  session_start();
3  $_SESSION['auth'] = null;
4  header("Location: index.php");
5  ?>

```

Тестируем. Если все верно, то сервис со стороны пользователя полностью готов!

Функционал администратора

1 Создаем файл **admin.php**. Авторизовываясь под логином и паролем администратора (прописан в ТЗ), попадаем в панель администратора где будут сразу видны все заявки от пользователей. Можно сделать дубль страницы **orders_user.php**. Немного подправить меню навигации и запрос на вывод данных.

```

<nav>
  <a href = "admin.php">Панель администратора</a>
  <a href = "update.php">Сменить статус</a>
  <a href = "logout.php">Выход</a></p>
</nav><br>
<main>
  <h2>Все заявки пользователей</h2>
  <?php
    include_once("db.php");
    $result = mysqli_query($link, "SELECT * FROM orders LEFT JOIN users ON orders.user_id = users.id
                                  LEFT JOIN status ON orders.status_id = status.id
                                  ORDER BY date_start");
    while($row = mysqli_fetch_assoc($result)){
      echo "<div class='cards'>
        <p><b>ФИО клиента:</b> $row[fio]</p>
        <p><b>Телефон клиента:</b> $row[phone]</p>
        <p><b>Дата подачи заявки:</b> $row[date_start]</p>
        <p><b>Тип оборудования:</b> $row[oborud]</p>
        <p><b>Модель:</b> $row[model]</p>
        <p><b>Описание неисправности:</b> $row[description]</p>
        <p><b>Статус:</b> $row[name]</p>
      </div>";
    }
  ?>
</main>

```

При тестировании в браузере выводятся все заявки от пользователей, добавляются еще поля ФИО клиента и его телефон.

Добавление поиска заявки (по ФИО клиента)

1. На странице **admin.php** добавляем строку поиска:

IT - сервис

[Панель администратора](#) [Сменить статус](#) [Выход](#)

Поиск заявки

Введите ФИО клиента

Все заявки пользователей

Для этого вносим изменение в код, добавив форму поиска. В нашем примере поиск будет осуществляться по ФИО клиента. Можно сделать по другому параметру.

```
<main>
  <h2>Поиск заявки</h2>
  <form method = "POST" >
    <label>Введите ФИО клиента</label>
    <input type = "text" name = "fio">
    <button>Найти заявку</button>
  </form><br><br>
  <h2>Все заявки пользователей</h2>
```

2. Програмируем: перед кодом, где осуществляется вывод всех заявок, прописываем условие. Если в поле поиска вводится данные существующего клиента, то выводятся только его заявки, в противном случае – заявки всех пользователей.

```
<?php
include_once("db.php");
if(!empty($_POST['fio'])){
    $search = mysqli_query($link, "SELECT * FROM orders LEFT JOIN users ON orders.user_id = users.id
                                  LEFT JOIN status ON orders.status_id = status.id
                                  WHERE fio = '$_POST[fio]'");
    while($res = mysqli_fetch_assoc($search)){
        echo "<div class='cards'>
        <p><b>ФИО клиента:</b> $res[fio]</p>
        <p><b>Телефон клиента:</b> $res[phone]</p>
        <p><b>Дата подачи заявки:</b> $res[date_start]</p>
        <p><b>Тип оборудования:</b> $res[oborud]</p>
        <p><b>Модель:</b> $res[model]</p>
        <p><b>Описание неисправности:</b> $res[description]</p>
        <p><b>Статус:</b> $res[name]</p>
        </div>";
    }
}else {
```

Результат:

IT - сервис

[Панель администратора](#) [Сменить статус](#) [Выход](#)

Поиск заявки

Введите ФИО клиента

Все заявки пользователей

ФИО клиента: Иванов Иван Иванович	ФИО клиента: Иванов Иван Иванович
Телефон клиента: 89277756682	Телефон клиента: 89277756682
Дата подачи заявки: 2024-05-27 16:05:47	Дата подачи заявки: 2024-05-27 16:12:54
Тип оборудования: Принтер	Тип оборудования: Компьютер
Модель: aaa456	Модель: gvdf555
Описание неисправности: не захватывает бумагу	Описание неисправности: Нагревается и шумит
Статус: новая	Статус: новая

1. Программный код для подсчета количества заявок. Для тестирования, можно провести изменение статуса в таблице БД.

```
<?php
session_start();
if(!empty($_SESSION['auth'])){
}
include_once("db.php");

$count = mysqli_query($link, "SELECT COUNT(*) as count FROM orders
                                LEFT JOIN users ON orders.user_id = users.id
                                LEFT JOIN status ON orders.status_id = status.id
                                WHERE name = 'завершена'");

$res1 = mysqli_fetch_assoc($count);
$count = $res1['count'];

?>
<!DOCTYPE html>
```

Для вывода статистики пропишем над формой поиска:

```
<main>
<h4>Количество выполненных заявок: <?php echo $count ?></h4>

<h2>Поиск заявки</h2>
```

Тестируем:

IT - сервис

[Панель администратора](#) [Сменить статус](#) [Выход](#)

Количество выполненных заявок: 1

Изменение статуса заявки (по указанию модели оборудования)

1. За ссылкой «Изменить статус заявки» на панели навигации закрепим файл **update.php**.
2. В блок **main** добавим форму на изменение статуса заявки. В форме будет поле для ввода модели устройства, список вариантов изменения статуса и кнопка.

```
<main>
  <h2>Сменить статус заявки</h2>
  <form action = "" method = "POST">
    <table>
      <tr>
        <td>Модель</td>
        <td><input type = "text" name="model"></td>
      </tr>
      <tr>
        <td>Сменить статус</td>
        <td><select name = "status">
          <option value = "в процессе ремонта">в процессе ремонта</option>
          <option value = "завершена">завершена</option>
          <option value = "готов к выдаче">готов к выдаче</option>
          <option value = "ожидание запчастей">ожидание запчастей</option>
        </select></td>
      </tr>
      <tr>
        <td></td>
        <td><button>Изменить</button></td>
      </tr>
    </table>
  </form>
```

Вид из браузера:

IT - сервис

[Панель администратора](#) [Сменить статус](#) [Выход](#)

Сменить статус заявки

Модель	<input type="text"/>
Сменить статус	<div>в процессе ремонта ▾</div>
	<div>Изменить</div>

3. Под формой разместим программный код:

```

</form>
<?php
    include_once("db.php");

    if(!empty($_POST['model']) && !empty($_POST['status'])){
        $model = $_POST['model'];
        $status = $_POST['status'];

        $result = mysqli_query($link, "UPDATE orders LEFT JOIN status ON orders.status_id = status.id
                                         SET name = '$status'
                                         WHERE model = '$model'");
        if($result == 'true'){
            header("Location: admin.php");
        } else{
            "<p class = 'error'>Статус не изменен</p>";
        }
    }
?>

```

4. Не забываем в верхней части программного кода прописывать авторизацию пользователя!

```

<?php
session_start();
if(!empty($_SESSION['auth'])){
}
?>

```

4. При тестировании кода, если происходит изменение статуса заявки, то осуществляется переход на главную страницу административной панели. Изменение статуса также происходит и в БД.

Этого функционала вполне достаточно!!!