Адаптивная верстка

<div class=”container”>

<div class=”column”>

Column 1

</div>

<div class=”column”>

Column 1

</div>

<div class=”column”>

Column 1

</div>

</div>

Необходимо написать css код, и расположить блоки .column горизонтально. Ширина блоков при разрешении экрана > 1024 px должна быть 1/3 размеры экрана.

При разрешении экрана < 500px блоки должны расположиться вертикально один под другим, и занимать 100% ширины браузера.

**Решение**

.column {

width: 33.33%;

height: auto;

}

@media screen and (max-width: 500px) {

.column {

width: 100%;

height: auto;

}

}

База данных: SQL Запрос

Есть две таблицы

**Таблица products**

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **name** | **price** | **category** |
| 1 | Product 1 | 100 | 1 |
| 2 | Product 2 | 200 | 2 |
| 3 | Product 3 | 150 | 1 |
| 4 | Product 4 | 200 | 3 |

**Таблица categories**

|  |  |  |
| --- | --- | --- |
| **id** | **slug** | **name** |
| 1 | pc | Компьютеры |
| 2 | phone | Телефоны |
| 3 | notebook | Ноутбуки |

Необходимо написать запрос, чтобы получить список товаров, зная только значение **slug** категории.

Также плюсом будет реализовать схем таблица БД когда один товар может иметь несколько категорий, но всегда должен иметь одну главную

То есть продукт “рубашка в клетку” может иметь главную категорию “**Рубашки**” но также иметь дополнительные категории “**Одежда**”, “**Для мужчин**”, “**Акции и скидки**”

Написать запрос и в этом случае, то-есть зная только slug получить список торов которые размещены не только в своих главных категориях, и в дополнительных.

**Решение**

SELECT \* FROM products p1

LEFT JOIN categories c1 ON p1.category = c1.id

WHERE c1.slug = "pc"

**Решение2**

Удалить колонку **category** в таблице продуктов, а вместо этого создать дополнительную таблицу **product\_categories** с полями **id,product\_id,category\_id** где и будет привязка продуктов к категориям.

**Запрос:**

**Только в одной категории**

SELECT p1.id,p1.name,p1.price FROM products p1

LEFT JOIN product\_categories pc1 ON p1.id = pc1.product\_id

LEFT JOIN categories c1 ON c1.id = pc1.category\_id

WHERE c1.slug = "notebook"

**В нескольких категориях**

SELECT prd1.id, prd1.name, prd1.price FROM categories ctg1

LEFT JOIN product\_categories prc1 ON ctg1.id = prc1.category\_id

LEFT JOIN products prd1 ON prd1.id = prc1.product\_id

WHERE prc1.product\_id

IN

(SELECT p1.id FROM products p1

LEFT JOIN product\_categories pc1 ON p1.id = pc1.product\_id

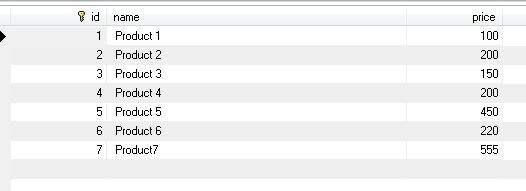
LEFT JOIN categories c1 ON c1.id = pc1.category\_id

WHERE c1.slug = "notebook")

GROUP BY prd1.id

**Таблицы:**

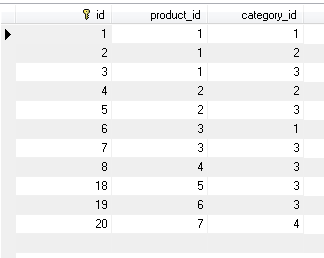
**Products**



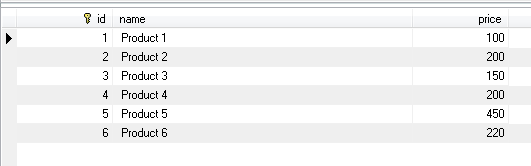
**Categories**



**Product\_categories**

****

**Выборка с запроса:**

****

Фреймворки и работа с данным

Взять любой знающий фреймворк yii/laravel/symfony/silex и сделать простое приложение которое будет получить запросы по API и отправлять письмо на почту по шаблоны, так же сделать авторизацию по api токену. Почту, куда отправлять почту необходимо указать в конфигурационном файле фреймворка.

Есть  роут (url)

**POST /api/v1/{event}**

**{event}** - это название события, например new\_order, new transaction, new\_message возьмем пока-что только **new\_order**

Входящие параметр только один, это номер заказа  ну и токен **api\_token**

**Например:**

{“order”: 1, “api\_token”: “...”}

каждое событие имеет свой шаблона, в который подставляются данные из запроса, в нашем случае это значение order

но заложить механизм таким образом, что бы мы могли создавать еще события которые нам будут нужны в будущем, и они так же будут иметь свои шаблоны и свой набор обязательных данных.

**Например событие new\_transaction**

Должно иметь два параметра **{card}** и **{amount}**

Результат на почту отправляем либо просто функцией mail либо установить одно из популярных решений через  composer, PHPMailer или SwiftMailer

**И финальным этапом, залить это приложение в ваш личный github репозиторий.**

Куда слать резюме

Резюме слать на почту [vk@etradeua.com](mailto:vk@etradeua.com)

Тема письма: **Резюме php-developer**

В письме пришлите ваше резюме,  ссылку на github репозиторий с выполненным заданием. Решение первых двух заданий можете прислать архивом в письме, или другим удобным способом.