

Задача: запрограммировать логику работы страницы товара по схеме MVC



ASICS

Мужские кроссовки Gel-Kayano Trainer
Knit Reflective Knit Pack Silver/Black

13 890 ₽

Цвет



Размер

Выбрать размер ▼

Добавить в корзину

Артикул

H7S3N-9301

Материал

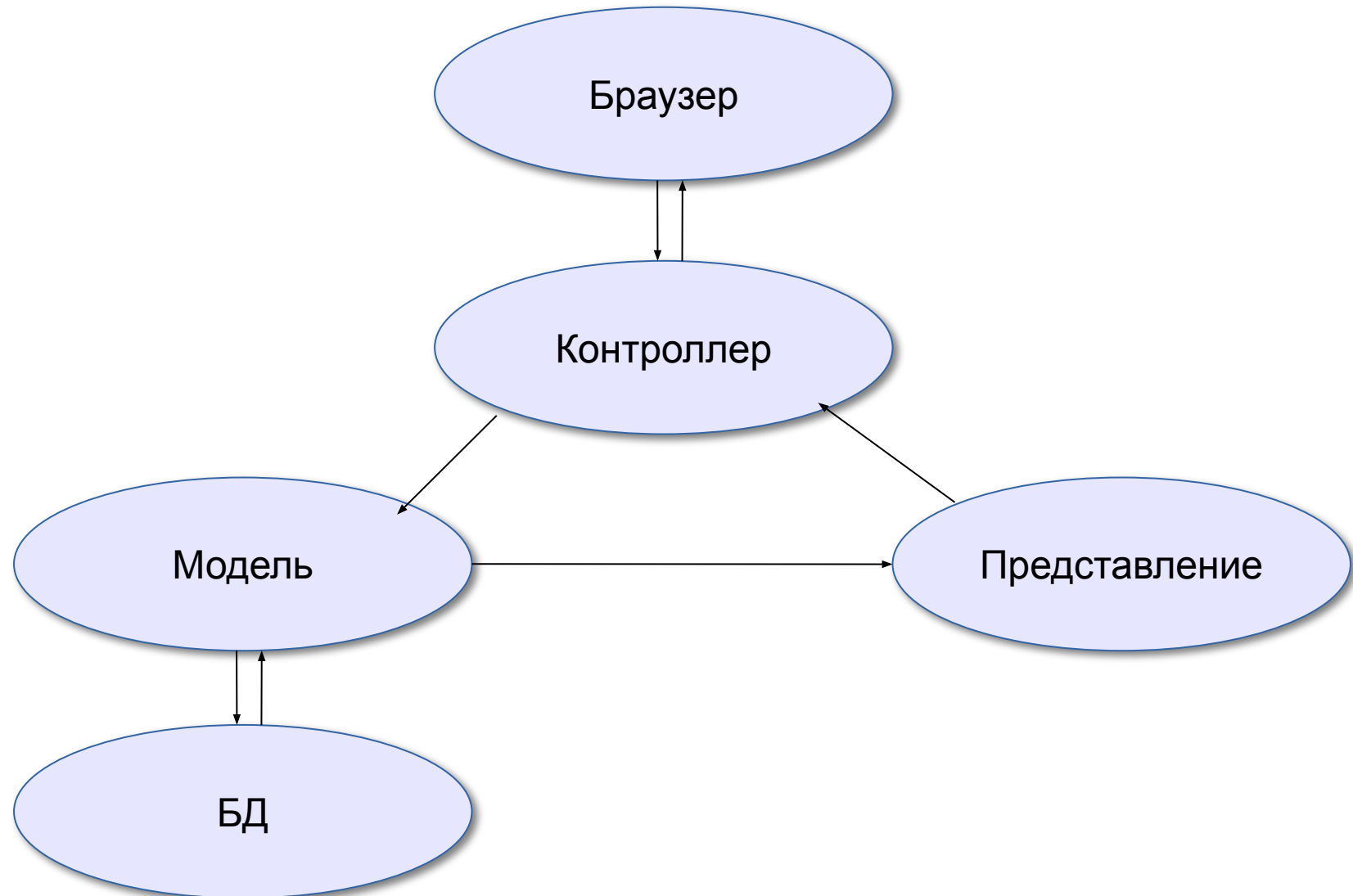
текстиль, резина

Страна

Индонезия

Инструменты: PHP, MySQL

Model-View-Controller



Model-View-Controller

- Браузер отправляет запрос Контроллеру
- Контроллер анализирует запрос и обращается к нужной Модели
- В Модели формируется набор данных и передаётся в Представление
- В Представлении формируется готовый ответ и возвращается Контроллеру
- Контроллер возвращает ответ браузеру.

Разбиваем задачу:

- Спроектировать БД:
 - Проанализировать требуемые сущности
 - Создать структуру таблиц
- Создать файловую структуру приложения
- Создать Контроллер
- Создать Модель
- Создать Представление
 - Создать шаблон на основе верстки

Анализируем макет, выделяем сущности

8 800 990-34-43



ASICS

Мужские кроссовки Gel-Kayano Trainer
Knit Reflective Knit Pack Silver/Black

13 890 ₽

Цвет



Размер

Выбрать размер

Добавить в корзину



Артикул

H7S3N-9301

Материал

текстиль, резина

Страна

Индонезия

ASICS Kayano Trainer Knit — современное прочтение кроссовок, смоделированных на основе классической беговой модели Gel-Kayano. Переосмысленная классика

Показать больше ↓

Основные команды SQL

CREATE TABLE ...

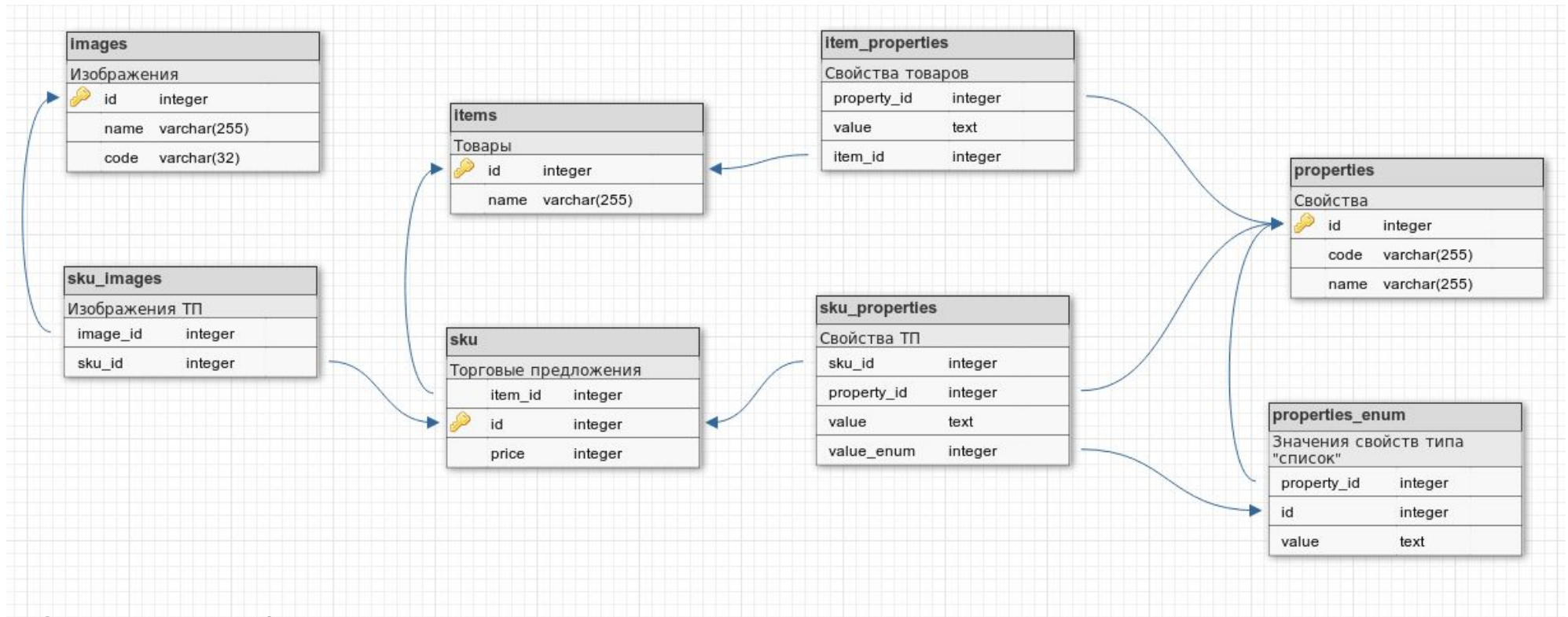
SELECT ... JOIN ...

INSERT ...

UPDATE ...

Практика создания запросов: www.sql-ex.ru

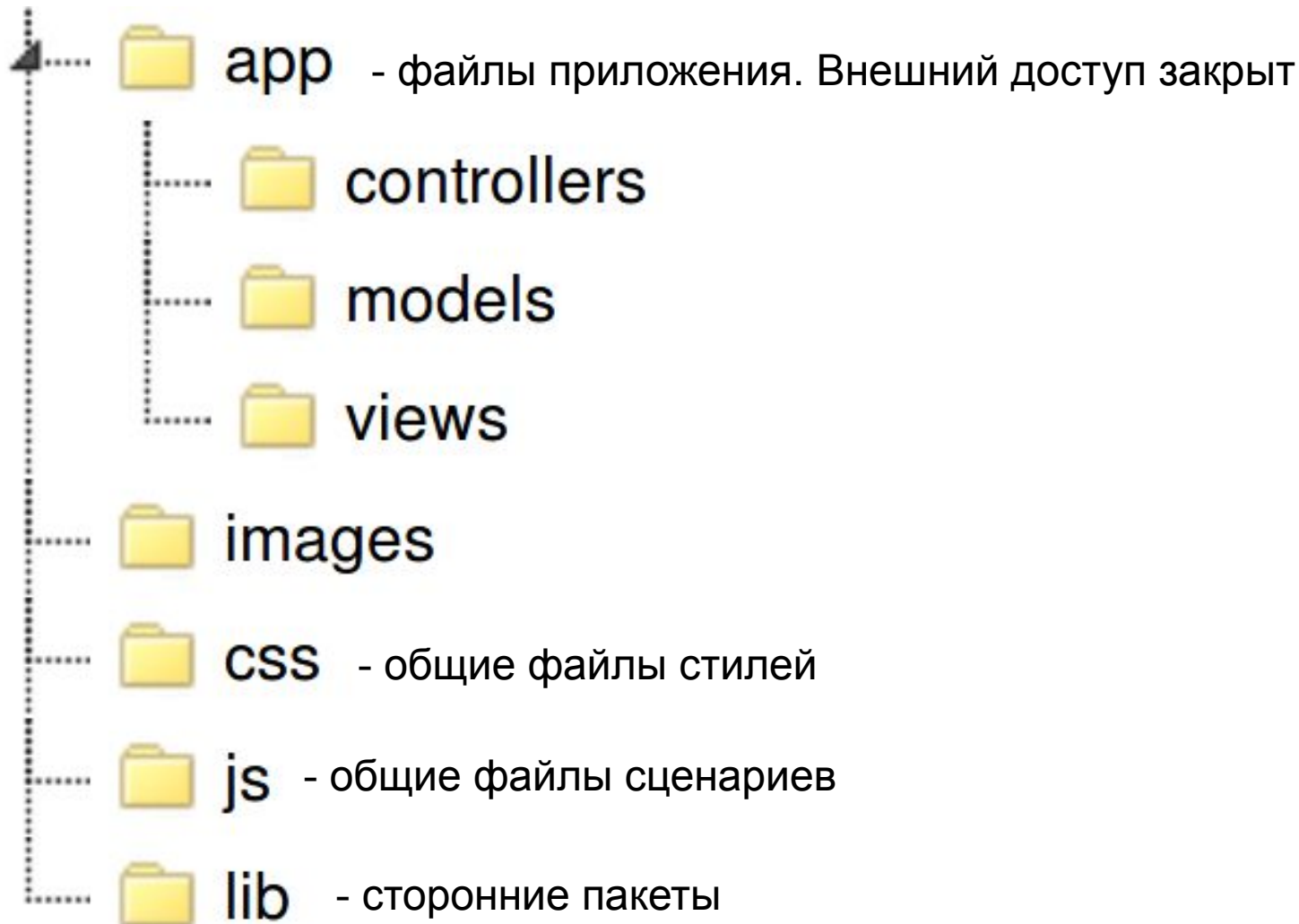
Структура таблиц



Создание таблицы (пример для товаров):

```
CREATE TABLE `items` (  
  `id` INT NOT NULL  
  AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

Создаем файловую структуру приложения



app / core.php - ядро

```
core.php  x
1  <?php
2
3  // регистрируем автозагрузку классов
4  spl_autoload_register(function ($className) {
5      if (strpos(strtolower($className), 'controller') !== false) {
6          $filePath = $_SERVER['DOCUMENT_ROOT'] . '/app/controllers/';
7      } elseif (strpos(strtolower($className), 'model') !== false) {
8          $filePath = $_SERVER['DOCUMENT_ROOT'] . '/app/models/';
9      } else {
10         $filePath = $_SERVER['DOCUMENT_ROOT'] . '/app/';
11     }
12
13     $filePath .= $className;
14     $filePath = str_replace('\\', '/', $filePath);
15     $filePath .= '.php';
16     if (file_exists($filePath)) {
17         include_once $filePath;
18     }
19 });
20
21 // стартуем сессию
22 if (!isset($_SESSION)) {
23     session_start();
24 }
25
26
```

app / View.php

```
View.php x
1  <?php
2
3  class View
4  {
5      public static function make($viewName, $data)
6      {
7          $filePath = $_SERVER['DOCUMENT_ROOT'] . '/app/views/';
8          $filePath .= $viewName;
9          $filePath = str_replace('\\', '/', $filePath);
10         $filePath .= 'template.php';
11         if (file_exists($filePath)) {
12             ob_start();
13             require($filePath);
14             return ob_get_clean();
15         }
16     }
17 }
18
```

app / controllers / CatalogController.php

```
CatalogController.php x
1  <?php
2
3  class CatalogController
4  {
5      public function getIndexPage()
6      {
7      }
8
9      public function getItemPage($itemId)
10     {
11         $viewData = CatalogModel::getItemData($itemId);
12
13         $view = View::make('catalog_item', $viewData);
14
15         return $view;
16     }
17
18     public function addToCart()
19     {
20     }
21 }
22
```

CatalogModel.php x

```
1  <?php
2
3  class CatalogModel
4  {
5      public static function getItemData($itemId)
6      {
7          $mysqli = new mysqli($DBHost, $DBLogin, $DBPassword, $DBName);
8
9          $query =
10             "SELECT i.id as ID, i.name as NAME, ip.value as PROP_VALUE , ps.code as PROP_CODE, ps.name as PROP_NAME ".
11             "FROM items as i " .
12             "JOIN item_properties as ip ON i.id = ip.item_id ".
13             "JOIN properties as ps ON ip.property_id = ps.id " .
14             "WHERE i.id = " . intval($itemId);
15
16             $dbResult = $mysqli->query($query);
17             $item = [];
18
19             while ($row = $dbResult->fetch_assoc()) {
20                 $item['ID'] = $row['ID'];
21                 $item['NAME'] = $row['NAME'];
22                 $item['PROP'][$row['PROP_CODE']]['NAME'] = $row['PROP_NAME'];
23                 $item['PROP'][$row['PROP_CODE']]['VALUE'] = $row['PROP_VALUE'];
24             }
25
26             return $item;
27         }
28     }
```

#	ID	NAME	PROP_VALUE	PROP_CODE	PROP_NAME
1	1	Botinky Rulf	чёрный	color	цвет
2	1	Botinky Rulf	кожа	material	материал

Переводим сырой набор данных в структурированный массив

```
Array (  
  [ID] => 1  
  [NAME] => Botinky Rulf  
  [PROP] => Array (  
    [color] => Array (  
      [NAME] => цвет  
      [VALUE] => чёрный  
    )  
    [material] => Array (  
      [NAME] => материал  
      [VALUE] => кожа  
    )  
  )  
) |
```

- В папке app/views/catalog_item создаём template.php
- Тут же создаём папки css, js, images и кладём туда специфические для данного представления стили, сценарии, картинки...
- В template.php копируем верстку и заменяем динамические блоки на вывод из массива \$viewData...

```
<div class="col small-1 medium-7 large-8">  
  <div class="item__data">  
    <b>ASICS</b>  
    <p><?= $viewData['NAME']; ?> </p>  
    <div class="price">13 890 ₺</div>  
  </div>
```


Точка входа. Файл index.php

```
index.php x
1  <?php
2
3  require ('app/core.php');
4
5  if (isset($_GET['itemId'])) {
6
7      echo CatalogController::getItemPage($_GET['itemId']);
8
9  } else {
10
11     echo CatalogController::getIndexPage();
12 }
13
14
15
16
```

Обратить внимание

Оформление кода: PSR

Приведение входных/выходных параметров: intval, htmlspecialchars...