

Тестовое задание: PHP Разработчик

Благодарим за проявленный интерес!



Наша вакансия на сайте HeadHunter

1 Тестовое задание

Данная задача в общем и целом представляет собой упрощённую версию одной из задач, с которой мы часто сталкивались в процессе работы.

Тестовое задание разбито на две части. Выполнение второй части будет плюсом, выполнение же первой обязательно.

Языком для реализации является PHP, базой данных - PostgreSQL или MySQL, остальные значения не имеет.

1.1 Часть 1

Необходимо написать скрипт импорта данных, обрабатывающий директорию с несколькими файлами в формате xml, и складывающий получившиеся данные в базу данных (БД).

Файлы представляют собой немного видоизменённый формат типовой выгрузки системы 1С, содержащей в себе информацию о товарных предложениях.

В директории располагаются файлы importX_1.xml и offersX_1.xml. В названиях файлов X обозначает город, информация по товарам которого представлена в файле. Каждая пара файлов (import и offers) в совокупности представляет собой информацию о товарах для конкретного города. Файлы дополняют друг друга, то есть часть информации находится в одном файле, часть - в другом.

Название города также указано в содержимом тега **Классификатор->Наименование** в начале файла, например:

`<Наименование>Классификатор (Москва)</Наименование>`

Это может использоваться для определения названия города, для которого применяется данный файл.

Нужная для задания информация - список товаров и торговых предложений, которые сопоставляются по значению поля **Код**.

В файле **import** это:

`<Каталог>`

```

<Товары>

<Товар>
  <!-- поля товара -->
  <Код>1234567</Код>          <!--Код товара-->
  <Вес>0.035</Вес>           <!--Вес товара-->
  <Наименование>Наименование товара</Наименование> <!--Название товара-->
  <Взаимозаменяемости>
    <Взаимозаменяемость>
      <Марка>CUMMINS</Марка>
      <Модель>ISBe6.7 (ISDe6.7)</Модель>
      <КатегорияТС>Двигатели</КатегорияТС>
    </Взаимозаменяемость>
    <!-- ... ещё взаимозаменяемости -->
  </Взаимозаменяемости>
  <!-- остальные поля можно игнорировать -->
</Товар>

<Товар>
  <!-- информация о другом товаре -->
</Товар>

<!-- ... ещё товары ... -->

</Товары>
</Каталог>

```

В файле **offers**:

```

<ПакетПредложений>
<Предложения>

<Предложение>
  <Код>1234567</Код>          <!--Код товара-->
  <Наименование>Наименование товара</Наименование> <!--Название товара-->
  <Количество>1</Количество>          <!--Количество товара в
<Цены>
  <!-- В качестве цены товара берётся первая по порядку цена из блока, ос
  <Цена>
    <ЦенаЗаЕдиницу>11891</ЦенаЗаЕдиницу> <!--Цена товара-->
  </Цена>
  <!-- ... -->
</Цены>
  <!-- остальные поля можно игнорировать -->
</Предложение>

<Предложение>
  <!-- информация о другом предложении -->

</Предложение>

<!-- ... ещё предложения ... -->

</Предложения>
</ПакетПредложений>

```

Соответственно, блоки **Товар** и **Предложение** с одинаковым полем **Код** относятся к одному товару. Не все товары представлены во всех файлах, то есть в одном городе может быть больше товаров, чем в другом.

Необходимо написать скрипт на PHP, который бы разбирал содержимое всех файлов, собирал их вместе и складывал бы в таблицу в БД следующего формата:

- id - ID товара в БД (autoincrement, primary key)
- name - название товара
- code - код товара
- weight - вес товара
- quantity_CITY - количество товара в конкретном городе. Несколько колонок, по одной для каждого города, например quantity_msk, quantity_kazan и так далее. Коды для городов можно взять любые
- price_CITY - цена товара в конкретном городе. Аналогично quantity_CITY
- usage - перечисленные через символ "|" взаимозаменяемости для данного товара в виде "Марка-Модель-КатегорияТС,Марка-Модель-КатегорияТС,...". Пример: "Foton-1039 E4-Грузовые автомобили|Cummins-ISLe8.9-Двигатели|...".

Если в каком-либо городе нет количества или цены для данного товара, или он отсутствует вообще, в соответствующие поля ставится 0. Если в разных файлах у одного и того же товара имеются несовпадающие поля, например, название или вес, можно использовать любое из них для записи в БД.

При последующем запуске скрипт должен обновлять данные по товарам, находя соответствие по полю **Код** (то есть обновлять то, что уже есть, а не добавлять новые).

Скрипт желательно реализовать в виде программы для командной строки UNIX-образной ОС.

1.2 Часть 2

Вывести данные из БД на веб-странице в постраничном виде в виде таблицы, упорядоченные в порядке убывания id. Язык реализации - PHP, используемые фреймворки, инструменты и внешний вид не важны. Количество товаров на странице тоже не имеет значения (то есть можно взять любое адекватное число, например, 20 или 50).

1.3 Данные

Данные для работы: [data.tar.gz](#)

Created: 2019-12-17 Вт 13:55

Уточнения:

1. К заданию должна быть приложена инструкция, которая позволит развернуть проект с нуля. Для проверки задания используется тестовая машина с php 7.4, mariadb 10.4 (Вы можете использовать MySQL) и postgresql 12. Соответственно, должны быть пояснения как создаётся база данных и запускается php-код.
2. Задание должно выполняться относительно быстро, использовать только XMLReader и ООП. Производительность компьютера у всех разная, но на условно средней машине задание не должно выполняться больше 5 минут. Обращаем внимание что размер файла для импорта может превышать кратно размер оперативной памяти, для тестирования используются файлы более 100 Гб.
3. Мы рассматриваем аккуратный код, как будто это настоящее решение.
4. Мы выбираем человека с самым качественным решением - быстрые решения, которые в реальных условиях показывали бы минимальное количество ошибок и проблем у коллег на этапе code-review.

Результат ждём по e-mail

hh@sionic.ru

Тема сообщения: "PHP Программист"