Практические задания

**SQL**

Есть 3 таблицы

Направления - class\_types

Группы - groups

Посетители - visitors

creationDate - это unsigned int, в котором хранится время в unix timestamp.

lastName, name, middleName - это фамилия, имя и отчество посетителя.



**Уровень 1**

* Вывести только поля id и lastName по всем посетителям.
* Выбрать всех посетителей с именем “Саша”.
* Выбрать всех посетителей у которых в имени, фамилии или отчестве встречается буквосочетание “иван”.
* Отсортировать всех посетителей по фамилии, имени, отчеству в обратном порядке от Я до А.
* Вывести id, фамилию, имя посетителя, который был добавлен в базу последним.
* Вывести список посетителей, добавленных в базу за последние 2 дня.

**Уровень 2**

* Вывести поля id, фамилию, имя посетителей, которые записаны в группу с id = 52. Отсортировать в алфавитном порядке по фамилии, имени.
* Вывести id, фамилию, имя посетителя, который был последним добавлен в группу с id = 52.
* Вывести количество посетителей в каждой группе. Список нужно отсортировать по количеству посетителей в группе по спаданию,

*Например:*

|  |  |
| --- | --- |
| Йога продвинутые | 7 |
| Йога начинающие | 5 |
| Йога средний уровень | 2 |

* Вывести список групп в алфавитном порядке и посетителей, которые записаны в эти группы также в алфавитном порядке. Результат должен состоять из трех колонок: id группы, название группы и фамилия+имя посетителя.

Например:

|  |  |  |
| --- | --- | --- |
| id | Название Группы | Фамилия Имя посетителя |
| 4 | Йога начинающие | Алексеева Дарья |
| 8 | Йога начинающие | Волошин Сергей |
| 7 | Йога продвинутые | Антонов Павел |
| 6 | Йога продвинутые | Коновалов Андрей |
| 10 | Йога продвинутые | Шишков Александр |
| 3 | Йога средний уровень | Богун Владимир |
| 1 | Йога средний уровень | Смолянюк Денис |

* Отобразить список групп, в которых нет ни одного посетителя. id и название группы.
* Выбрать всех посетителей из Направления с id = 4. Сортировать посетителей по ФИО по алфавиту.

**Уровень 3**

* Вывести список Направлений, Групп и Посетителей каждой группы. Направления, Группы и Посетители должны быть отсортированы по алфавиту.

*Например:*

|  |  |  |
| --- | --- | --- |
| Название направления | Название Группы | Фамилия Имя посетителя |
| Йога | Йога начинающие | Алексеева Дарья |
| Йога | Йога начинающие | Волошин Сергей |
| Йога | Йога продвинутые | Антонов Павел |
| Йога | Йога продвинутые | Коновалов Андрей |
| Йога | Йога продвинутые | Шишков Александр |
| Йога | Йога средний уровень | Богун Владимир |
| Йога | Йога средний уровень | Смолянюк Денис |
| Фитнес | Фитнес взрослые | Аладин Ахмед |
| Фитнес | Фитнес взрослые | Волошкина Алла |
| Фитнес | Фитнес дети 3-5 лет | Иванов Петр |
| Фитнес | Фитнес дети 3-5 лет | Шишлинов Павел |

* Вывести список посетителей, которые посещают 2 и более групп и кол-во групп, которые они посещают.
* Вывести посетителей и список групп, которые они посещают через запятую.

Посетители должны быть отсортированы по алфавиту. Отобразить только первых 10 посетителей. То есть нужно сделать так, чтобы скрипт в результате давал две колонки и информацией.

*Например:*

Иванов Иван Иванович Группа1, Группа 2, Группа 5

Иранская Валентина Дмитриевна Группа 5

Петров Александр Владимирович Группа 2

Салова Анастасия Петровна Группа 4, Группа 6

...

**Уровень 4**

* Есть таблицы:

**info\_sources** - источники обращений (откуда посетитель узнал об организации, в которую обратился).

**visitors** - посетители. (в поле **balance** записана сумма денег, которая есть на депозите посетителя).

**payments** - платежи посетителей (поле **paidMoneyAmount** - сумма денег, которую внес посетитель за один товар или услугу или за пакет товаров или услуг, таких товаров и услуг может быть куплено любое количество, при каждой оплате создается новая запись в таблице payments).



Задача: Вывести список источников обращений со следующими данными:

* кол-во посетителей, которые узнали об организации из этого источника
* Сумма денег на балансе всех посетителей, которые узнали об организации из этого источника
* сумма денег, которую заплатили суммарно посетители, которые пришли из данного источника обращений.

*Например:*

|  |  |  |  |
| --- | --- | --- | --- |
| *Название источника* | *Всего на депозитах* | *Кол-во посетителей* | *Сумма денег* |
| Реклама в метро | 400 | 10 | 1008 |
| Google | 540 | 56 | 5904 |
| VK | 180 | 23 | 985 |

* Есть простая программа управления товарами и складом.

В базе данных присутствуют следующие таблицы.



**products**

Таблица с товарами. Это просто справочник всех товаров, с которыми работает компания.

**warehouses**

Таблица со складами. Склад - это некое помещение, в котором товары хранятся.

**products\_movements**

Операции движений товаров. В этой таблице хранится список всех движений товаров: закупок, продаж и перемещений между складами.

Поля в таблице:

date - дата совершения операции

productId - товар

amount - количество товара, которое принимало участие в данном движении

pricePerItem - стоимость единицы товара

priceTotal - стоимость всей парти товаров, которые принимали участие в данном движении

operationType - тип операции. Может быть одним из значений:

income - приход товара, закупка

sale - продажа товара

move - перемещение товара со склада на склад

warehouseId - склад, по которому происходило движение товара

**warehouses\_products**

таблица со сводными данными, в которой хранится информация о том, сколько остатков товаров есть на том или ином складе.

warehouseId - склад

productId - товар

amountLeft - сколько указанного товара находится на указанном складе.

Некоторые детали того, как организованы данные в базе.

* Закупки товаров добавляются в таблицу products\_movements с количеством товаров, которое больше нуля.
* Продажи товаров указываются с количеством товаров, которое меньше нуля.
* Перемещение товаров в таблицу добавляется двумя операциями, у каждой из которой operationsType = move.
* первая операция является операцией списания товаров и в ней указывается склад, с которого товар списывается и amount меньше нуля
* вторая операция - операция прихода товара на другой склад, в ней указывается склад, куда товар привезли и amount больше нуля.

Другие условия

Новые товары в справочник могут быть добавлены в любой момент времени, независимо от того, была ли произведена закупка этого товара.

Что нужно сделать

Необходимо написать sql-запрос, который обновит информацию в таблице **warehouses\_products** об остатках товаров на складах. Если товара нет в этой таблице для одного или нескольких складов, то запись об этом товаре нужно добавить в таблицу с указанием остатков на складах.

Если для того, чтобы все работало, нужно внести в базу дополнительные изменения - укажите, какие именно.

В результате выполнения задания должен получиться один sql-запрос.

**PHP, JS, Jquery**

**1.** На сервере установлено время GMT, даты в базе данных хранятся в виде unix timestamp так же в GMT.

Пользователь находится в часовом поясе GMT+3 и в запросе от пользователя пришла дата+время в виде строки “14.07.2016 15:35” (то есть в его часовом поясе).

* Присвоить переменной значение этого времени в unix timestamp для сохранения в базу (то есть для сохранения в часовом поясе GMT).
* Отобразить пользователю сохраненное время в его часовом поясе в том же формате, в котором оно пришло.

**2**. Есть массив с персиками разных цветов и размеров.

$peaches = array(

array(

‘color’ => ‘pink’,

‘size’ => 8

),

array(

‘color’ => ‘red’,

‘size’ => 10

),

array(

‘color’ => ‘orange’,

‘size’ => 6

),

array(

‘color’ => ‘green’,

‘size’ => 4

),

);

В цикле необходимо кажому персику добавить новое свойство, которое бы указывало на то, что персик относится к категории больших или маленьких. Большой персик - тот, у которого размер больше или равен 8. Свойство нужно назвать “isBig”, значение - булевское.

**3.** Есть дерево категорий и подкатегорий. Глубина вложенности может быть любой. Нужно вывести все категории и подкатегории в виде наглядного дерева (с отступами слева для каждой вложенной сущности).

В качестве исходных данных есть массив категорий в виде:

$cats = [

"Electronics" => [

"Phones" => ["Mobile Phones", "Satellite Phones"],

"Tablets" =>["6 inch", "7 inch", "9 inch"],

"Kids Toys",

...

],

...,

"Spare Parts" => [

"Wheels" => [...],

"Brakes" => [...],

"Discs" => [...],

"Transmission"

],

...

];

**Jquery**

**4.** Написать html и js код для следующей функциональности:

На странице есть checkbox и два текстовых поля (id=”field1” и id=”field2”).

<input type=”checkbox”>

<input type=”text” id=”field1”>

<input type=”text” id=”field2”>

Нужно сделать так, чтобы при включении чекбокса поле field1 становилось неактивно, а field2 активно, а при снятии галочки - наоборот.

**5.** На странице есть элементы:

<input class=”js-values” type=”text” value=”1”>

<input class=”js-values” type=”text” value=”6”>

<input class=”js-values” type=”text” value=”3”>

<input class=”js-values” type=”text” value=”19”>

<input class=”js-values2” type=”text” value=”54”>

<input class=”js-values2” type=”text” value=”5”>

<input class=”js-values2” type=”text” value=72”>

<input class=”js-values2” type=”text” value=”7”>

Нужно при помощи jquery сделать так, чтобы значения из каждой серии элементов были разбросаны по разным массивам.

**6.** Необходимо разработать архитектуру интерфейсов и классов для реализации описанной ниже задачи.

Стоит задача получать текущие курсы валют из разных источников, находить среднее значение курса для каждой валютной пары и сохранять в базу это среднее значение.

Все источники имеют разный формат предоставляемых ими данных, а также разные названия полей для одних и тех же данных.

Например,

* в одном источнике данные отдаются в формате json; поле, которое содержит в себе название валютной пары, называется pairName, а значение курса валют price.
* в другом источнике - формат xml; поле с названием валютной пары называется currency, а значение курса валют course.

Нужно создать архитектуру интерфейсов и классов (их описание, без реализации), которая позволит выполнять следующие действия:

* получать данные из разных источников и в разных форматах
* проводить вычисления с полученной информацией
* сохранять результаты расчетов в базу.

В результате выполнения задач должны быть представлены описания интерфейсов/классов, наследственные связи, сигнатуры методов. А также краткое описание того, что за что отвечает и как эти сущности между собой взаимодействуют.