Задания для стажеров

<table>
<tr><th>Уровень сложности</th><th>Задачи по SQL</th></tr>
<tr><td>I</td><td>Table: daily_weather
<table>
<tr><th>month</th><th>day</th><th>temperature</th></tr>
<tr><td>1</td><td>1</td><td>-15</td></tr>
<tr><td>1</td><td>2</td><td>-19</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>2</td><td>1</td><td>-5</td></tr>
<tr><td>2</td><td>2</td><td>0</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
</table>
1. Написать запрос который посчитает среднюю, минимальную и максимальную температуру
2. Написать запрос который посчитает среднюю, минимальную и максимальную температуру в разрезе месяцев</td></tr>
<tr><td>I</td><td>Какое максимальное и минимальное количество строк может быть получено при JOINe двух таблиц по 5 строк каждая (поля not NULL).
<table>
<tr><th></th><th>MIN</th><th>MAX</th></tr>
<tr><td>INNER</td><td></td><td></td></tr>
<tr><td>LEFT</td><td></td><td></td></tr>
<tr><td>RIGHT</td><td></td><td></td></tr>
<tr><td>FULL</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>II</td><td>Table: mon_salary
<table>
<tr><th>id</th><th>month</th><th>salary</th></tr>
<tr><td>1</td><td>1</td><td>100</td></tr>
<tr><td>1</td><td>2</td><td>150</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>2</td><td>5</td><td>200</td></tr>
<tr><td>2</td><td>6</td><td>180</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
</table>
Написать sql-запрос отбирающие все id и month у которых salary > avg salary для id</td></tr>
</table>

<table>
<tr><th>Уровень сложности</th><th>Задачи по SQL</th></tr>
<tr><td>II</td><td>
Table: VSP_oper_data
<table>
<tr><th>Client_id</th><th>Report_date</th><th>VSP_Number</th><th>Txn_type</th><th>Txn_amount</th></tr>
<tr><td>1233455</td><td>2017.05.02</td><td>1234/0123</td><td>debit</td><td>10000</td></tr>
<tr><td>1233455</td><td>2017.05.03</td><td>1236/0123</td><td>credit</td><td>1000</td></tr>
<tr><td>1233455</td><td>2017.05.04</td><td>1234/0123</td><td>credit</td><td>1000</td></tr>
<tr><td>1233455</td><td>2017.05.07</td><td>1235/0123</td><td>debit</td><td>15000</td></tr>
<tr><td>1233456</td><td>2017.05.02</td><td>1234/0123</td><td>debit</td><td>11000</td></tr>
<tr><td>1233456</td><td>2017.05.03</td><td>1236/0123</td><td>credit</td><td>10000</td></tr>
<tr><td>1233456</td><td>2017.05.04</td><td>1234/0123</td><td>credit</td><td>10000</td></tr>
<tr><td>1233456</td><td>2017.06.07</td><td>1237/0123</td><td>debit</td><td>5000</td></tr>
</table>
В таблице VSP_oper_data txn_type принимает значения debit, credit

Задание: напишите sql запрос, который для каждого клиента выводит сумму debit, credit операций и последнее посещенное VSP по месяцам. Результат представьте в виде:
<table>
<tr><th>Client_id</th><th>Report_date</th><th>Debit_amount</th><th>Credit_amount</th><th>Last_VSP</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>III</td><td>
1. Реализовать задачу из п. 2 тремя различными способами

Table: daily_weather
<table>
<tr><th>month</th><th>day</th><th>temperature</th></tr>
<tr><td>1</td><td>1</td><td>-15</td></tr>
<tr><td>1</td><td>2</td><td>-19</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>2</td><td>1</td><td>-5</td></tr>
<tr><td>2</td><td>2</td><td>0</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
</table>
2. Написать sql запрос который считает для каждого дня среднюю температуру за 5 предыдущих дней
</td></tr>
<tr><td>III</td><td>
Задание: напишите sql запрос, который для каждого клиента из VSP_oper_data выведет долю debit операций клиента к debit операциям всех клиентов по месяцам. Результат в виде таблицы:
<table>
<tr><th>Client_id</th><th>Report_date</th><th>Ratio</th></tr>
<tr><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

Уровень сложности	Scala или python (используя spark, pandas или стандартные коллекции)
I	**Необходимо на любом знакомом языке программирования написать алгоритм:** Вывести построчно целые числа от 1 до N не более C чисел в строке
II	## Задача Написать приложение на scala или python, которое в локальном режиме выполняет следующее: По имеющимся данным о рейтингах книг посчитать агрегированную статистику по ним. 1. Прочитать csv файл: **book.csv** 2. Вывести схему для dataframe полученного из п.1 3. Вывести количество записей 4. Вывести информацию по книгам у которых рейтинг выше 4.50 5. Вывести средний рейтинг для всех книг. 6. Вывести агрегированную инфорацию по количеству книг в диапазонах: 0 - 1 1 - 2 2 - 3 3 - 4 4 - 5

III	## Задача
	Написать приложение (опционально spark приложение), которое в локальном режиме выполняет следующее:
	По имеющимся данным о рейтингах фильмов (MovieLens: 100 000 рейтингов) посчитать агрегированную статистику по ним. Использовать архив **data.zip**
	
	## Описание данных
	Имеются следующие входные данные:
	Архив с рейтингами фильмов.
	Файл README содержит описания файлов.
	Файл u.data содержит все оценки, а файл u.item — список всех фильмов. (используются только эти два файла)
	id_film=32
	
	1. Прочитать данные файлы.
	2. создать выходной файл в формате json, где
	Поле `“Toy Story ”` нужно заменить на название фильма, соответствующего id_film и указать для заданного фильма количество поставленных оценок в следующем порядке: `"1", "2", "3", "4", "5"`. То есть сколько было единичек, двоек, троек и т.д.
	В поле `“hist_all”` нужно указать то же самое только для всех фильмов общее количество поставленных оценок в том же порядке: `"1", "2", "3", "4", "5"`.
	
	Пример решения:
	
	{
	"Toy Story": [
	134,
	123,
	782,
	356,
	148
	],
	"hist_all": [
	134,
	123,
	782,
	356,
	148
	]
	}

Уровень сложности	Scala или python (используя spark, pandas или стандартные коллекции)
III	**Входные данные:** **Customer.csv** – информация о клиентах Имя поля: формат id: Int, name: String, email: String, joinDate: Date, status: String **Product.csv** – информация о товарах id: Int name: String price: Double numberOfProducts: Int **Order.csv** – информация о заказах customerID: Int orderID: Int productID: Int numberOfProduct: Int – кол-во товара в заказе orderDate: Date status: String Необходимо определить самый популярный продукт у клиента Итоговое множество должно содержать поля: customer.name, product.name Результат записать в csv-файл