Платформа ОФД. Задание для кандидатов на позицию Data Engineer

# Задание

Разработать шаблон для отчета, который принимает на вход несколько параметров и файл с разметкой и, используя данные о кассах и продажах в базе SQLite, формирует файл с отчетом.

# Исходные данные

Исходные данные включают:

* Файл с разметкой двух брендов в формате csv с полями:
  + brand
  + product\_name\_hash (хэш от названия товара, по которому можно найти интересующие продажи в чеках)
* База данных de\_test.db SQLite (<https://www.sqlite.org/>) со следующими таблицами:
  + kkt\_activity – данные о первом и последнем чеке для всех касс
  + kkt\_categories – категории касс по виду деятельности
  + kkt\_info – основная информация о кассах, включая данные организации и торговой точки
  + sales – продажи касс (чеки)

***Примечание***. Все названия полей в таблицах унифицированы. Таким образом, поля с одинаковым названием в разных таблицах ссылаются на одни и те же сущности, и по ним можно джойнить таблицы (например, kkt\_number во всех таблицах указывает на конкретную уникальную кассу).

## Таблица kkt\_activity

Описание полей:

| Поле | Описание |
| --- | --- |
| kkt\_number | Регистрационный номер кассы |
| receipt\_date\_min | Дата первого чека, полученного от кассы |
| receipt\_date\_max | Дата последнего чека, полученного от кассы |

## Таблица kkt\_categories

Описание полей:

| Поле | Описание |
| --- | --- |
| kkt\_number | Регистрационный номер кассы |
| category | Категория кассы по виду деятельности |
| date\_from | Дата начала действия записи (включается) |
| date\_till | Дата начала действия записи (не включается) |
| version | Версия классификации |

Примечания к таблице:

1. Таблица содержит несколько версий классификации. Когда появляется новая версия классификатора торговых точек, то все кассы классифицируются за все время их работы и в таблице появляются новые записи для новой версии, при этом старые версии не удаляются и могут использоваться в старых отчетах.
2. Классификация касс производится каждую неделю (так как касса могла начать торговать другими товарами). Таким образом, каждую неделю для каждой работающей кассы появляется новая запись в таблице для каждой версии классификатора.
3. В таблице хранятся только записи для работающих касс, которые удалось классифицировать. Остальных касс в таблице нет.

## Таблица kkt\_info

Описание полей:

| Поле | Описание |
| --- | --- |
| org\_inn | ИНН организации |
| shop\_id | Идентификатор торговой точки (уникален в рамках конкретной организации) |
| kkt\_number | Регистрационный номер кассы |
| region | Регион, в котором работает касса |
| date\_from | Дата начала действия записи (включается) |
| date\_till | Дата начала действия записи (не включается) |

Примечания к таблице:

1. Таблица содержит основную информацию о кассе за всю историю ее работы.
2. Для каждой кассы в таблице может быть несколько записей, которые не пересекаются по периоду действия, обозначенному датами date\_from и date\_till.
3. Данные таблицы являются выжимкой из более полной таблицы, поэтому в ней могут быть полностью идентичные записи с разными периодом действия. Это значит, что в исходной таблице изменились какие-то параметры кассы, которые в итоге не попали в выжимку для текущего задания. Это никак не влияет на принципы работы с таблицей.

## Таблица sales

Описание полей:

| Поле | Описание |
| --- | --- |
| org\_inn | ИНН организации |
| receipt\_date | Дата продажи (чека) |
| receipt\_id | Идентификатор продажи (чека) |
| product\_name\_hash | Хэш от названия товара |
| total\_sum | Сумма продажи товара |

# Требования к отчету

## Входящие параметры

Должна быть возможность задать следующие параметры отчета:

1. Путь к файлу product\_names.csv.
2. Период, задается двумя датами в формате yyyy-MM-dd (обе даты включаются в отчет):
   1. date\_from
   2. date\_to
3. Фильтр kkt\_category, принимающий на вход список категорий: может быть пустым или содержать любое кол-во категорий (например: FMCG, HoReCa).
4. Признак необходимости группировки данных отчета по следующим полям:
   1. receipt\_date
   2. region
   3. channel (данное поле вычисляется, описание ниже)

## Содержание отчета

Отчет всегда формируется по продажам брендов из входящего файла product\_names.csv в соответствии с разметкой. То есть для каждого бренда (brand) из файла нужно найти все продажи, имеющие соответствующие названия товаров (product\_name\_hash). Все прочие продажи в отчет попадать не должны.

### Фильтры

У отчета должен быть только один опциональный фильтр kkt\_category. Если он пустой, то параметр игнорируется. В противном случае при формировании отчета должны учитываться только продажи касс указанных категорий. В данном отчете используется всегда наиболее актуальная версия классификации (максимальная дата в поле version).

### Разрезы

У отчета есть один обязательные разрез, который должен быть всегда – brand. Остальные разрезы опциональны и зависят от указанных признаков необходимости группировки во входящих параметрах отчета:

* receipt\_date: если признак указан, то отчет должен быть дополнительно сгруппирован по полю receipt\_date из таблицы sales
* region: если признак указан, то отчет должен быть дополнительно сгруппирован по полю region из таблицы kkt\_info
* channel: если признак указан, то отчет должен быть дополнительно сгруппирован по полю channel, которое необходимо вычислить

Значение поля channel указывает на то, является ли организация торговой сетью, и вычисляется в соответствии со следующими правилами:

1. Значение вычисляется для организации целиком, то есть привязано к уникальному значению org\_inn.
2. Значения зависят от количества активных торговых точек (shop\_id из таблицы kkt\_info в периоде, за который строится отчет (входящие параметры date\_from и date\_to).
3. Торговая точка считается активной, если есть хотя бы одна касса, у которой период активности (receipt\_date\_min и receipt\_date\_max из таблицы kkt\_activity) пересекается с периодом отчета.
4. Возможные значения:
   1. nonchain: у организации < 3 активных торговых точек в периоде, за который формируется отчет
   2. chain: у организации >= 3 активных торговых точек в периоде, за который формируется отчет

### Показатели

В рамках отчета рассчитываются два показателя:

1. total\_sum – сумма продаж бренда (по полю total\_sum из таблицы sales)
2. total\_sum\_pct – процент продаж бренда от продаж всех брендов (total\_sum бренда, разделенный на сумму total\_sum всех брендов в отчете, округленный до 2-х знаков после запятой)

### Формат

Отчет должен формироваться в виде файла:

* Название: report
* Формат: csv
* Заголовки: названия полей
* Разделитель: запятая

## Примеры отчета

Данные в примерах не являются реальными результатами, а приведены

### Пример №1

Входящие параметры:

1. Период:
   1. date\_from: “2019-08-01”
   2. date\_to: “2019-08-02”
2. Фильтр kkt\_category: не указан.
3. Разрезы:
   1. receipt\_date: false
   2. region: false
   3. channel: false

Результат:

| brand | total\_sum | total\_sum\_pct |
| --- | --- | --- |
| parliament | 200 | 0.67 |
| marlboro | 100 | 0.33 |

### Пример №2

Входящие параметры:

1. Период:
   1. date\_from: “2019-08-01”
   2. date\_to: “2019-08-02”
2. Фильтр kkt\_category: FMCG.
3. Разрезы:
   1. receipt\_date: true
   2. region: true
   3. channel: true

Результат:

| receipt\_date | region | channel | brand | total\_sum | total\_sum\_pct |
| --- | --- | --- | --- | --- | --- |
| 2019-08-01 | г. Москва | chain | parliament | 200 | 0.67 |
| 2019-08-01 | г. Москва | chain | marlboro | 100 | 0.33 |
| 2019-08-02 | г. Москва | chain | parliament | 200 | 0.67 |
| 2019-08-02 | г. Москва | chain | marlboro | 100 | 0.33 |
| 2019-08-01 | г. Москва | nonchain | parliament | 200 | 0.67 |
| 2019-08-01 | г. Москва | nonchain | marlboro | 100 | 0.33 |
| 2019-08-02 | г. Москва | nonchain | parliament | 200 | 0.67 |
| 2019-08-02 | г. Москва | nonchain | marlboro | 100 | 0.33 |
| 2019-08-01 | Санкт-Петербург | chain | parliament | 200 | 0.67 |
| 2019-08-01 | Санкт-Петербург | chain | marlboro | 100 | 0.33 |
| … | … | … | … | … | … |

***Примечание***. Так как был указан фильтр, то в отчете отображаются только данные категории FMCG.

# Требования к результатам

В качестве решения задания необходимо предоставить:

1. Исходный код на одном из языков: Scala, Python, Java.
2. Указания по использованию шаблона (чтобы проверяющий мог сам запустить решение и убедится, что оно работает).
3. Три варианта готового отчета в соответствии с таблицей ниже:

| date\_from | date\_to | categories | receipt\_date | region | channel |
| --- | --- | --- | --- | --- | --- |
| 2019-08-01 | 2019-08-31 |  | false | false | false |
| 2019-08-01 | 2019-08-31 |  | true | true | true |
| 2019-08-05 | 2019-08-11 | FMCG | false | true | false |

Примечания к решению:

1. В рамках задания можно использовать SQL-запросы в любом объеме.
2. Файл product\_names.csv можно загружать в базу при условии, что загрузка будет реализована в рамках решения (то есть пользователь в любом случае указывает путь к файлу в исходных параметрах).
3. Можно использовать любую базу данных при условии, что проверяющий сможет быстро воспроизвести результаты (ему не потребуется разворачивать собственную базу данных для проверки решения).

# Вопросы по решению

На втором этапе собеседования необходимо будет ответить на несколько вопросов относительно данного задания:

1. В исходной базе данных SQLite таблицы никак не оптимизированы. Какие рекомендации вы можете дать по оптимизации таблиц, чтобы можно было быстро работать с миллионами записей в таблицах с данными касс и миллиардами записей в таблице sales?
2. Как вы оцениваете оптимизацию вашего решения? Что можно было бы улучшить, при условии, что шаблон отчета будет использоваться постоянно на больших объемах данных?
3. Если один из менеджеров попросит вас доработать отчет, чтобы можно было запускать его без файла product\_names.csv по всем продажам, какое решение вы ему предложите?