Задача

1). Сделать настройки программы Datacol для парсинга объявлений с сайта Avito в таблицу MySQL по заданным требованиям:

- Регион :: **Санкт-Петербург, Ленинградская область, Татарстан**
- Тип объявления :: **только “Частные”**
- Категории/подкатегории для **стартовых** URLов ::
    - Транспорт > Автомобили
    - Недвижимость > Квартиры
    - Недвижимость > Комнаты
    - Недвижимость > Дома, дачи, коттеджи
    - Недвижимость > Земельные участки
    - Недвижимость > Гаражи и машиноместа
    - Недвижимость > Коммерческая недвижимость
    - Недвижимость > Недвижимость за рубежом
    - Личные вещи > Часы и украшения
    - Личные вещи > Красота и здоровье
    - Транспорт > Мотоциклы и мототехника
- Категории/подкатегории, с которыми работаем **в принципе** :: Все, кроме “Работа > Вакансии” и “Работа > Резюме” и “Хобби и отдых > Знакомства”

2). Спроектировать таблицу MySQL, в которую будут складываться данные. Примерный формат:

- все значащие поля (см. описание ниже);
- текстовое поле для всего объявления;
- дата создания объявления;
- дата обновления объявления.

Номер объявления первичный или уникальный ключ.

Формат URL запроса на поиск нужных нам объявлений. Проверки.

Соответственно, URLы разделов, к которым мы станем обращаться, будут вида:

```
https://www.avito.ru/tatarstan/kvartiry?user=1
https://www.avito.ru/sankt-peterburg/avtomobili?user=1
https://www.avito.ru/leningradskaya_oblast/chasy_i_ukrasheniya?user=1
```

Где:

- **`user=1`** – это “частные” объявления
- **`tatarstan, sankt-peterburg, leningradskaya_oblast`** – это искомые регионы
- `kvartiry и т.п.` – это категории/подкатегории

Допустимо использование мобильной версии сайта для решения нашей задачи, если это будет проще без ущерба качеству итоговых данных.

При написании настроек необходимо учесть, что Datacol очень часто при парсинге “уходит” в другие категории, регионы и типы объявлений. Причем “уйти” он может как на страницу со *списком* объявлений, так и напрямую на страницу *конкретного* объявления (через блок “похожие объявления”).

Для нас *недопустим* уход в иные регионы кроме заданных, а также *критически недопустим* переход от “частных” объявлений к объявлениям компаний.

Поэтому необходимо сделать в настройках следующие проверки:

- проверку на нужный регион - в URLe должны содержаться **`tatarstan/sankt-peterburg/leningradskaya_oblast`**
- проверку на нужный тип (“частные”) объявления
    - для страницы со *списком* объявлений - в URLe должен содержаться **`user=1`**
    - на странице *конкретного* объявления в URLe тип юзера не указан, поэтому проверять надо как-то иначе.
      Насколько я понимаю, логика обозначения НЕчастного объявления на его странице такая:
        - либо в строке `<div class="description_seller"> ... </div>` указан в значении НЕ “`Продавец`”, а что-то иное - “`Автодилер`” для автомобилей (**пример**), “`Агенство`” для недвижимости (**пример**) и т.п.
        - либо в строке `<div class="description_seller"> ... </div>` указан все-таки “`Продавец`”, но при этом справа от его имени есть бейдж вида `<span class="c-2"> (...) </span>` со значениями типа “`компания`”, “`магазин`” и т.п. (**пример**)
        - **возможно, есть и иные признаки НЕчастного объявления, о которых я не упомянул**

Перечень полей для парсинга. Формат значений.

ПАРАМЕТР(Ы)	КАК СПАРСИТЬ / ПРИМЕР ЗНАЧЕНИЯ

Категория - Подкатегория объявления	Берем из каталога. Примеры:
	`Категория :: Недвижимость`
	`Подкатегория :: Комнаты`
	
	`Категория :: Транспорт`
	`Подкатегория :: Автомобили`

Тип объявления	Варианты значений:
	`Продам`
	`Сдам`
	`Куплю`
	`Сниму`
	Если не указано - присваивать значение по умолчанию “`Продам`”

Номер (ID) объявления	Брать как со страницы самого объявления… `Номер объявления: 587866915` **Важно! Для случаев, когда Datacol повторно парсит объявление с уже существующим в нашей базе ID:** Таблица у нас в MySQL одна. В ней номер объявления является первичным или уникальным ключом. При вставке объявления первый раз проставляется дата вставки. При повторной вставке этого же объявления по номеру, идет запрос `insert on duplicate key update` и проставляется дата обновления.
URL объявления	`https://www.avito.ru/moskva/avtomobili/honda_cr-v_2011_587866915`

Дата размещения объявления на сайте	Можно брать из списка объявлений или со страницы самого объявления.
	Формат:
	`(Размещено) сегодня в ЧЧ:ММ`
	`(Размещено) вчера в ЧЧ:ММ`
	`(Размещено) ДД месяца в ЧЧ:ММ`
	Соответственно, для нашей БД конвертим в формат нормальной даты /времени `ДД.ММ.ГГГГ ЧЧ:ММ`

Наименование товара	Берется из `title` объявления из списка объявлений или со страницы объявления.
Цена (руб.)	Берем из соответствующего поля `Цена` конкретного объявления **Важно!** Цену нужно доставать одну (их там может быть две) и сразу в виде числа (удаляя пробелы и "руб.")

Имя пользователя	В зависимости от категории, лейбл у поля на странице объявления может различаться (“Меня зовут”, “Продавец”, …), но в коде это, похоже, всегда
	
	`<div class="description_content" id="seller" itemprop="seller" itemscope="" itemtype="http://schema.org/Person">`
	`<strong itemprop="name">`
	`Вазген`
	`</strong> </div>`

Телефон пользователя	Берем значение со страницы конкретного объявления - расположен под именем

	Телефон частично скрыт от визуального считывания за кнопкой “Показать телефон” - придется, видимо, эмулировать клик по кнопке: `<span class="description__phone-insert js-phone-show__insert"><span class="btn__text">`**`Показать телефон`**`</span></span>` **Важно!** Телефон необходимо в базу класть в виде числа (удаляя пробелы и дефисы)
Страна	Так как avito.ru работает только с россиянами, для всех пишем по умолчанию - `Россия`

Область - Город - Метро	Если смотрим в поле “`Город`” на странице конкретного объявления, то логика такова:
	`Просто текст` = <`Город`> (областной центр)
	`Текст, Текст` = <`Область/регион`>, <`Город`> (не центральный) **ПРИМЕР**
	`Текст, м. Текст` = <`Город`> (областной центр), м. <`Станция метро`> **ПРИМЕР**
	
	Иных вариантов я не встречал. Может быть вы сможете парсить более изящным способом, но на выходе нам нужны значения трех полей:
	`Область`
	`Город`
	`Метро` (если есть)

HTML-текст всего объявления	Весь значащий текст объявления.