3. b Find buses between two time ranges

→ Use Arrival time

3. c List all the bus stops in sequence of a particular bus on a particular route (Herein, route information signifies all the stops and their sequence made by the bus running on the provided route). (Hint: For particular route use route_id and for particular bus use trip_headsign)

It should give **all bus stop name in sequence** (routes) for given trip_headsign and route id

For example - input:
trip_headsign = 159 Halifax and route_id = 159-221

Sample output :

Stop_Sequence	Name_stop	Route_id	Trip_headsign	Trip_id
1	Portland Hills Terminal	159-121	159 Halifax	6511549-..
2	Pehnom Terminal	159-121	159 Halifax	6511549-..
3	Alderney Dr opposite Terminal	159-121	159 Halifax	6511549-..
4	Wyse Rd Sportsplex Metrolink	159-121	159 Halifax	6511549-..
5	Barrington St in front of Scotia Square	159-121	159 Halifax	6511549-..

(Stop sequence 1-2-3-4-5) --- (Portland Hills Terminal –> Pehnom Terminal → Alderney Dr opposite Terminal → Wyse Rd Sportsplex Metrolink → Barrington St in front of Scotia Square)

is one route for trip_headsign = 159 Halifax and route_id = 159-221 . It may have have more than one route. Your output should be similar to the table shown above.

3.d Find top 3 bus stops that are the busiest throughout the day. (Hint:The bus stops with high volume of bus routes).

Typo in output: you need to **find 3 bus stops that are the busiest throughout the day**. **Not bus names**

Notes:

- As I mentioned in the assignment that some rows might be rejected (some records have different date format) so ignore it.
- Marking of assignment will be based on your query. The output might vary - for example. if someone is using not null for some columns then chances of record rejection will be more and vice versa. So Don't worry about number of rows.
- If you try to upload stoptimes.json, then it may fail because it is a large file. So, I have divided it into multiple parts (Given in a separate folder). Curl will append data for example if you want to upload 2 json files – stoptimes1.json, stoptimes2.json in bus/stoptimes. So first upload stoptimes1.json in bus/stoptimes then stoptimes2.json. When you upload stoptimes2.json, it will append those data (not overwrite so don't worry).

  So bus/stoptimes – stoptimes1.json – stoptimes14.json (total 14 json files)
  bus/stops – stops.json
  bus/trips – trips.json
- You don't need to use joins in elastic search queries. Example

For ExampleYou have 2 tables

EMPLOYEE TABLE

Emp_ID	EMP_FIRST_NAME	EMP_LAST_NAME	DEP_ID
1	Trishla	Shah	3
2	Trishla	abc	2
3	Trishla	xyz	3
4	Akash	Abc	1
5	Trishla	efg	1

DEPARTMENT TABLE

DEP_ID	DEP_NAME
1	Engineering
2	Marketing
3	Sales

Query : Find out department name of employees whose first name is Trishla

In RDBMS You can perform join between EMPLOYEES and DEPARTMENT TABLE using dep_id but for elastic search you can write 2 separate query.

Elastic search – example query

First,we need to find department id from the employees table

First query –

```
{

"query": { "match_phrase": { "EMP_FIRST_NAME": " TRISHLA" } },

"_source": ["DEP_ID"]

}
```

Here you will get 4 results. For example-

1.

EMP_FIRST_NAME: Trishla

DEP_ID: 3

2.

EMP_FIRST_NAME: Trishla

DEP_ID: 2

3.

EMP_FIRST_NAME: Trishla

DEP_ID: 3

4.

EMP_FIRST_NAME: Trishla

DEP_ID: 1

Then, using DEP_ID, you need to find department name. **ONLY TAKE 1 RESULT FROM ABOVE AND HARDCODE IT INTO SECOND QUERY. IGNORE OTHER RESULTS**

So 2nd Query will be :

```
{

"query": { "match_phrase": { " DEP_ID ": " 3" } },

"_source": ["DEP_NAME"]

}
```

it will give 1 result:

1.

DEP_ID: 3

DEP_NAME: Sales

Similarly, you can do for 3 tables. **Just show sample output.** Elasticsearch section is just to teach, how you can write simple queries in elasticsearch. So, Don’t worry ☺