Assignment – 1

Query Implementation using RDBMS system and Elastic Search

By:

Mr. Niravsinh Jadeja
(B00789139)

Task Description:

When it comes to building an application, it takes a lot effort and planning especially when dealing with databases. In this case, bus routes of Halifax city are provided. It is very helpful for creating a business application such as Transit application *[1]* which provides real-time buses information, utilizing such databases and query performed in back-end operation.

Here, the task is to perform some queries to fulfil the desired task such as finding all buses that go through a bus stop, finding list of bus details for a time frame, also finding of bus details such as stop sequence, and other route information when provided the bus name and route name, and find 3 busiest bus-stops through the day.

Relational Database Design:

This database is created and further executed on IBM Cloud platform *[2]*. It has used 'Db2 Warehouse on Cloud' service for performing Relational Database Management Systems (RDBMS) *[3]* query section of this assignment and 'Compose for Elasticsearch' for performing an elastic search query.

In RDBMS section, there are 3 tables named stops, stoptimes, and trip and each table contain the information related to bus routes, arrival time and so on. In 'stops' table, there are 3 columns stop_id, name_stop, lat, log is for stop identification number, the name of the stop, latitude, and longitude. The stoptimes holds trip_id, arrival_time, departure_time, stop_id, stop_sequence columns are respectively for trip identification number, arrival and departure time of the buses, with its bus-stop id number and stop sequence. The trips table contains following columns block_id, route_id, trip_headsign, service_id, shape_id, trip_id which are on block, route, service, shape, and trip identification number and headsign of the buses.

For the elastic search *[4]*, I have used Postman *[5]* service for uploading Jason format of the above datasets and for a search query.

I also have attached all queries for a relational database and an elastic section with its responses.

Application Queries & Test Result:

RDBM Section

a. Find all buses for a bus stop

Query:

SQL editor

```
1 select trip_headsign
2 from trips, stops
3 where name_stop like 'Ferry Stop - Woodside';
```

Response:

Jobs Clear All

Finished successfully

All (1) Failed (0)

select trip_headsign from trips, stops where name_stop like 'Ferry Stop - Wo...

Finished successfully

All (1) Failed (0)

select trip_headsign from trips, stops where name_stop like 'Ferry Stop - Woo...

Results Details

SQL statement

```
select trip_headsign
from trips, stops
where name_stop like 'Ferry Stop - Woodside'
```

Execution log

Run time: 0.016206232s

Status: Success

Rows affected: 0

Results Details

	TRIP_HEADSIGN
1	330 HALIFAX
2	53 NOTTING PARK TO HIGHFIELD TERMINAL
3	61 DOWNTOWN HALIFAX TO SCOTIA SQUARE
4	320 AIRPORT VIA FALL RIVER
5	10 WESTPHAL
6	54 MONTEBELLO
7	57 PORTLAND HILLS VIA PORTLAND ESTATES
8	320 DOWNTOWN HALIFAX VIA BRIDGE TERMINAL
9	88 BEDFORD COMMONS
10	22 ARMDALE TO EXHIBITION PARK
11	87 GLENDALE TO SACKVILLE TERMINAL
12	14 DOWNTOWN
13	2 WEDGEWOOD VIA MAIN
14	320 DOWNTOWN HALIFAX VIA BRIDGE TERMINAL
15	51 WINDMILL TO BRIDGE TERMINAL

1 2 3 4 5 ... 20 >

Here, I have provided a stop name ‘Ferry Stop – Woodside’ and it gives us a head sign of the buses for that bus-stop.

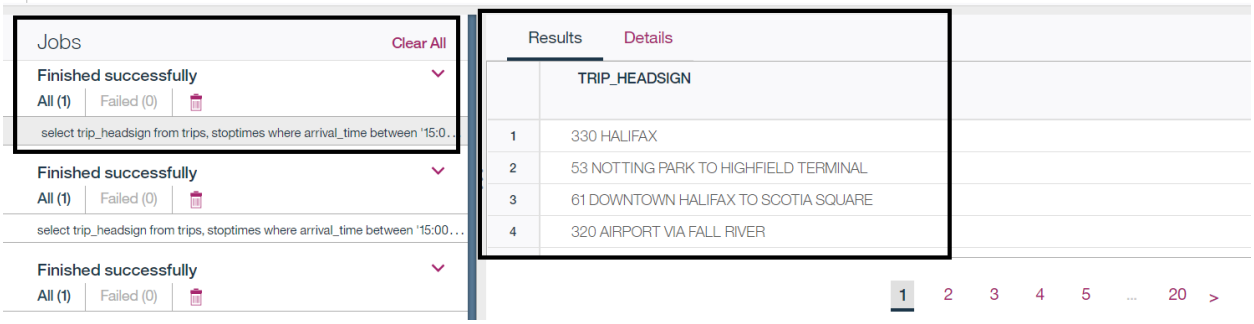
b. Find buses between two-time ranges

Query:

SQL editor

```
1 select trip_headsign
2 from trips, stoptimes
3 where arrival_time between '15:00:00' and '16:30:00';
```

Response:



Results Details

	TRIP_HEADSIGN
1	330 HALIFAX
2	53 NOTTING PARK TO HIGHFIELD TERMINAL
3	61 DOWNTOWN HALIFAX TO SCOTIA SQUARE
4	320 AIRPORT VIA FALL RIVER
5	10 WESTPHAL
6	54 MONTEBELLO
7	57 PORTLAND HILLS VIA PORTLAND ESTATES
8	320 DOWNTOWN HALIFAX VIA BRIDGE TERMINAL
9	88 BEDFORD COMMONS
10	22 ARMDALE TO EXHIBITION PARK
11	87 GLENDALE TO SACKVILLE TERMINAL
12	14 DOWNTOWN
13	2 WEDGEWOOD VIA MAIN
14	320 DOWNTOWN HALIFAX VIA BRIDGE TERMINAL
15	51 WINDMILL TO BRIDGE TERMINAL

1 2 3 4 5 ... 20 >

In this query I have provided a time frame from 15:00:00 to 16:30:00 and it gives us a head sign of those all buses which has arrival time from the mentioned time.

c. List all the bus-stops in the sequence of a bus on a particular route (Herein, route information signifies all the stops and their sequence made by the bus running on the provided route).

Query:

SQL editor

```
1 select stoptimes.stop_sequence, name_stop, route_id, trip_headsign, trips.trip_id
2 from trips, stops
3 inner join stoptimes on trips.trip_id = stoptimes.trip_id
4 where trip_headsign like '14 DOWNTOWN' and route_id like '14-114';
5
```

Response:

Jobs
Finished successfully
All (1) | Failed (0)
Clear All

Results | Details

	STOP_SEQUENCE	NAME_STOP	ROUTE_ID	TRIP_HEADSIGN	TRIP_ID
1	1	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BR
2	2	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BR
3	3	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BR
4	4	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BR

1 2 3 4 5 ... 20 >

Results | Details

STOP_SEQUENCE	NAME_STOP	ROUTE_ID	TRIP_HEADSIGN	TRIP_ID
1	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
2	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
3	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
4	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
5	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
6	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
7	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
8	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
9	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
10	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
11	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S
12	cow Bay Rd before Garrison Dr	14-114	14 DOWNTOWN	6529206-2012_08A-1208BRsu-S

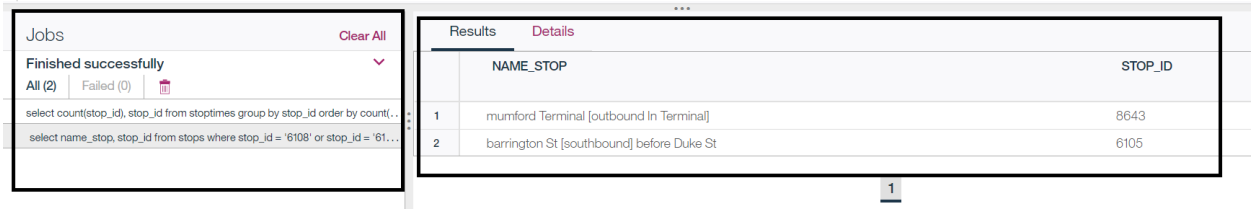
1 2 3 4 5 ... 20 >

d. Find top 3 bus stops that are the busiest throughout the day.

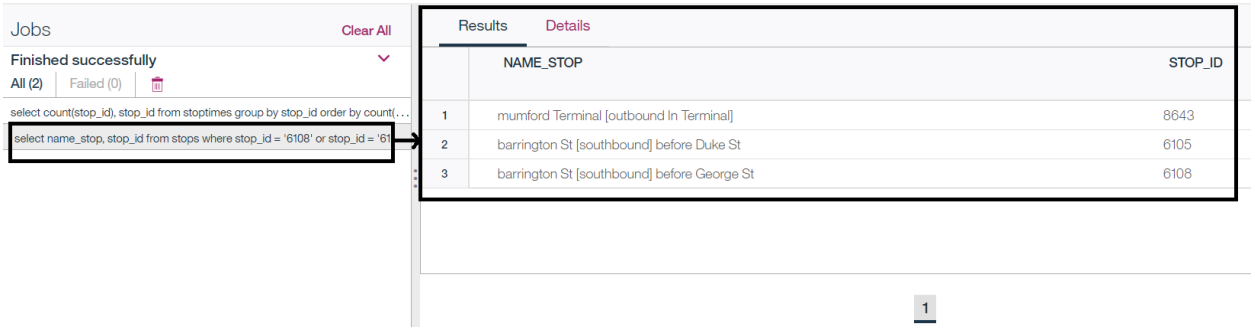
Query:

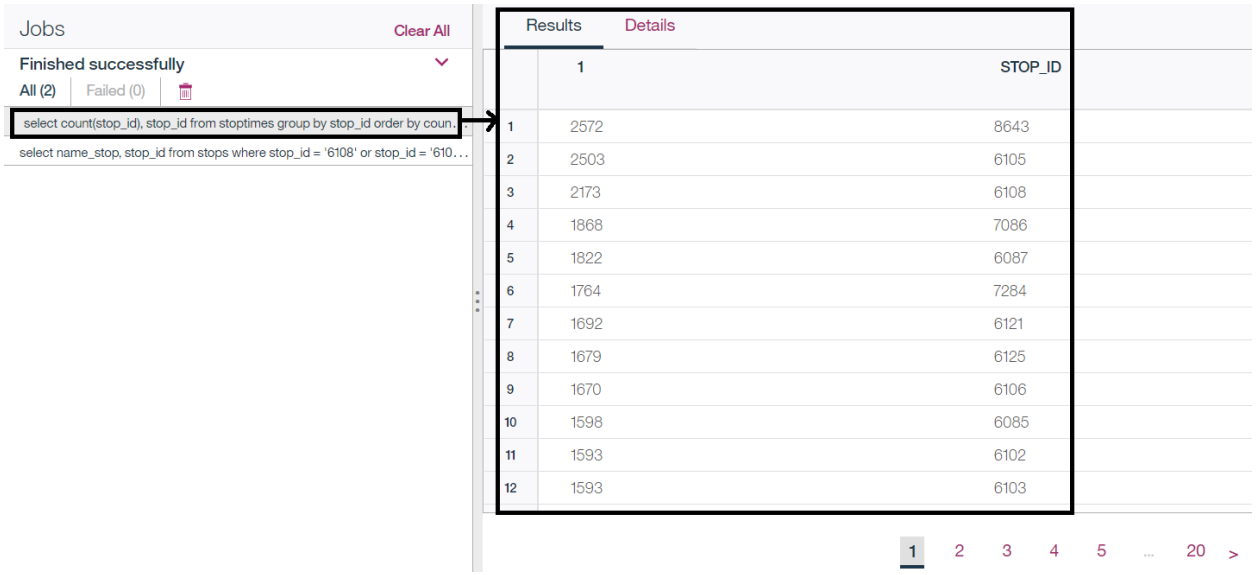
SQL editor

```
-- it contains 2 queries
-- 1st part is for finding which one is busy by counting stop_id from stoptimes

select count(stop_id), stop_id
from stoptimes
group by stop_id
order by count(stop_id) desc;

-- 2nd part is to assign the names of Bus-Stop with the help of stop_id
select name_stop, stop_id
from stops
where stop_id = '6108' or stop_id = '6105' or stop_id = '8643';

-- Result
-- 1   mumford Terminal [outbound In Terminal]      Stop_id: 8643
-- 2    barrington St [southbound] before Duke St   Stop_id: 6105
-- 3    barrington St [southbound] before George St Stop_id: 6108
```



Response:

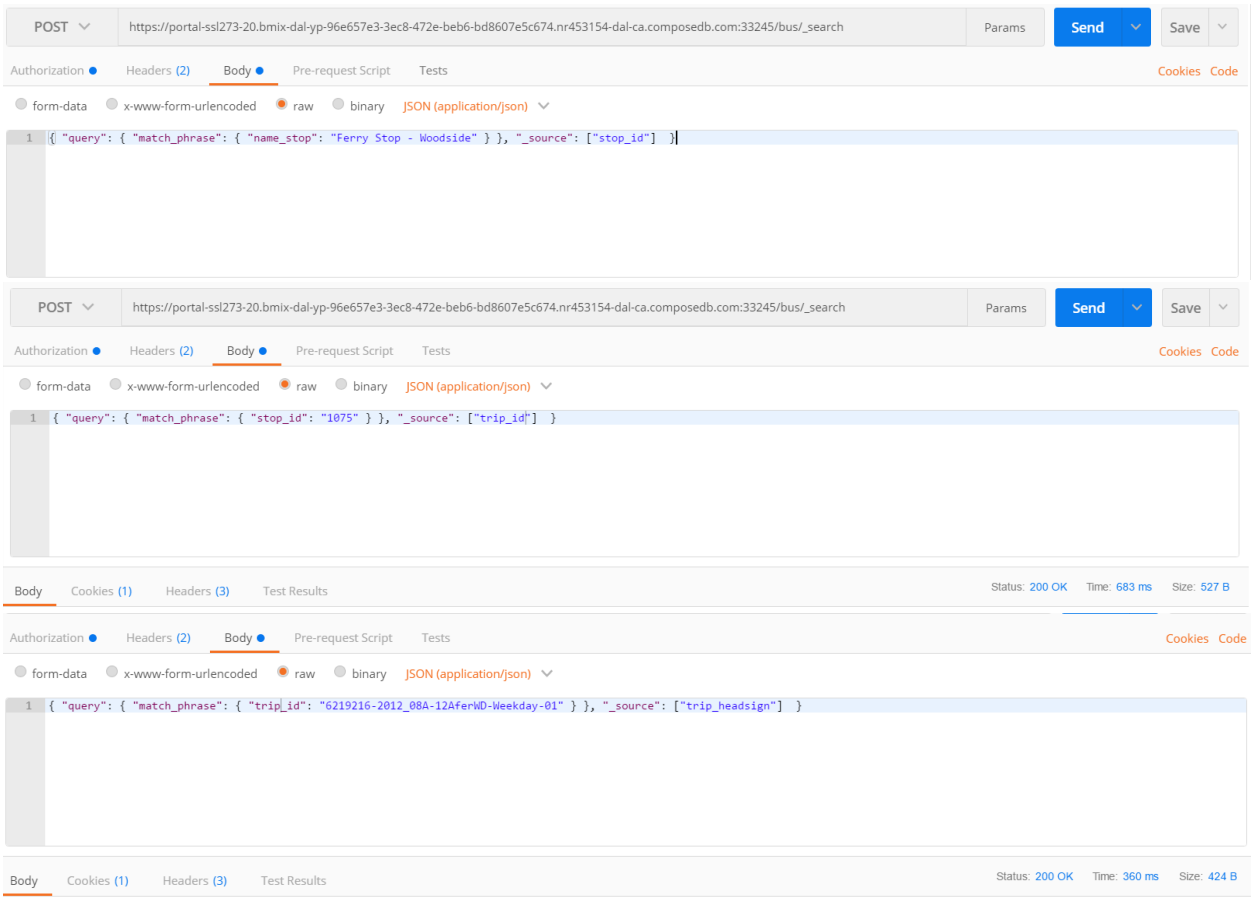




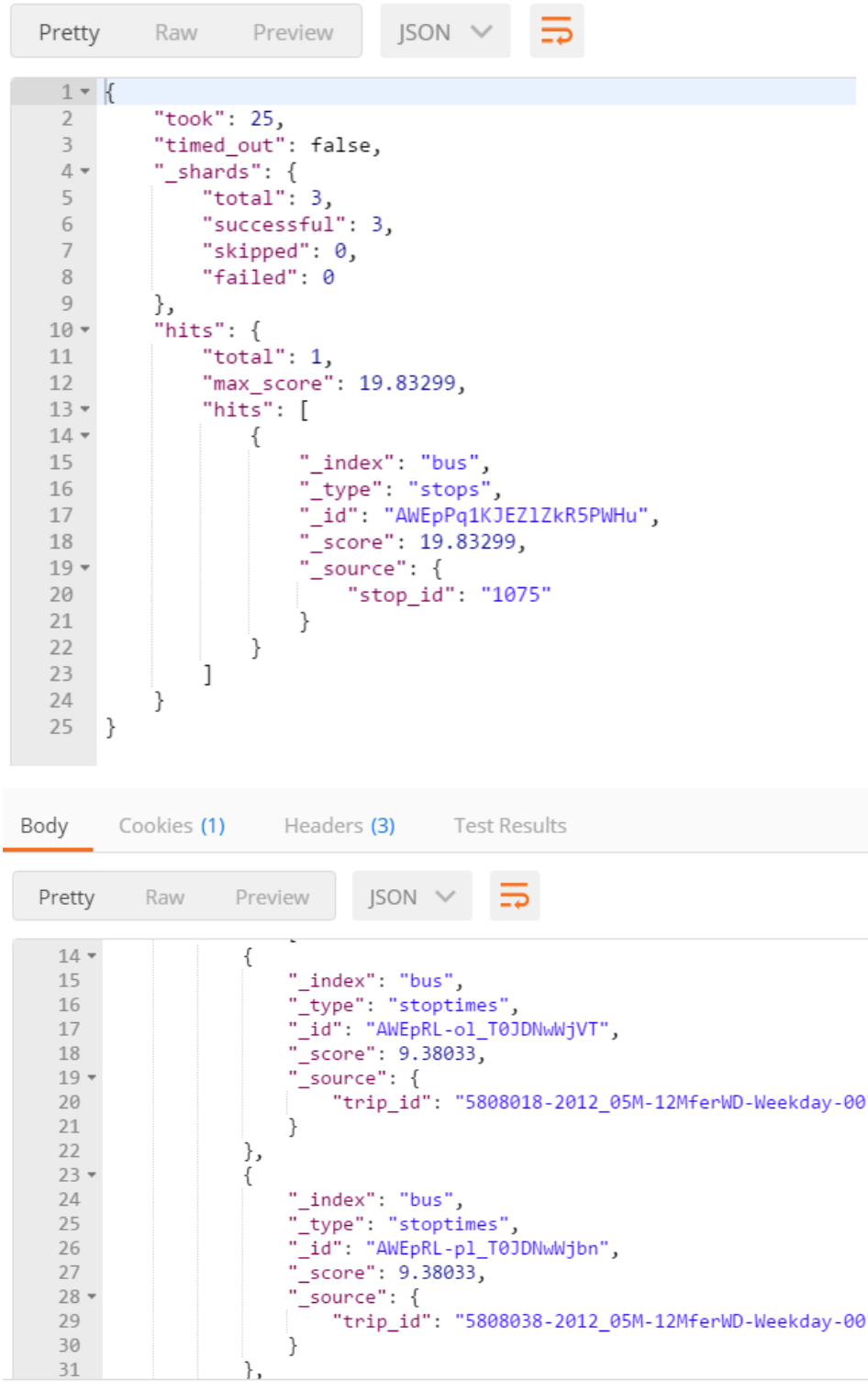
Elasticsearch Section:

a. Find all buses for a bus-stop

Query:

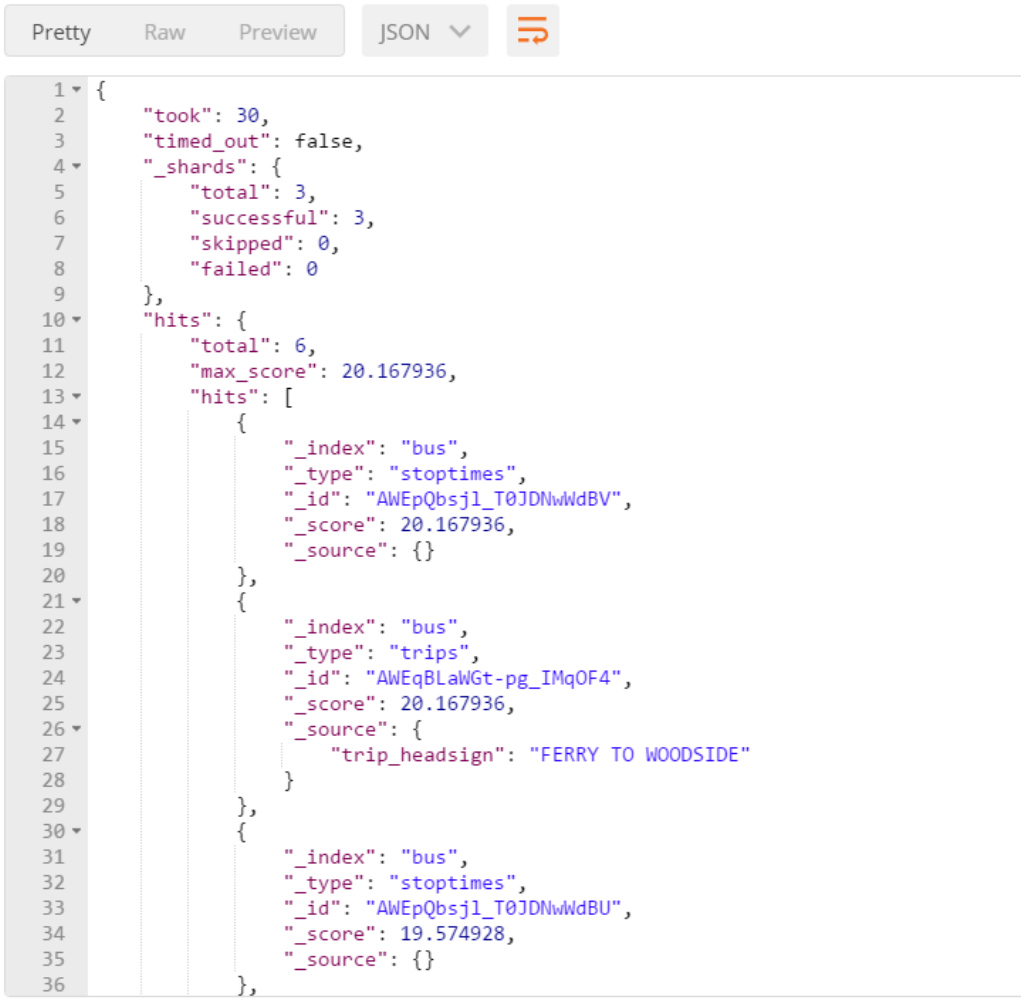


Response:



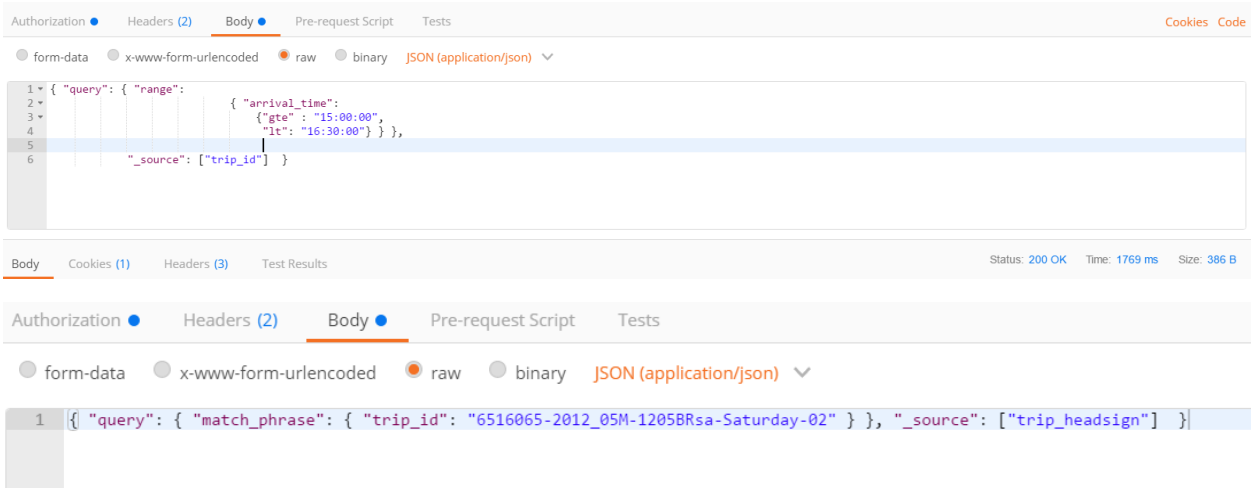

```
{
    "took": 25,
    "timed_out": false,
    "_shards": {
        "total": 3,
        "successful": 3,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": 1,
        "max_score": 19.83299,
        "hits": [
            {
                "_index": "bus",
                "_type": "stops",
                "_id": "AWEpPq1KJEZ1ZkR5PWHu",
                "_score": 19.83299,
                "_source": {
                    "stop_id": "1075"
                }
            }
        ]
    }
}
```

```
            {
                "_index": "bus",
                "_type": "stoptimes",
                "_id": "AWEpRL-ol_T0JDNwWjVT",
                "_score": 9.38033,
                "_source": {
                    "trip_id": "5808018-2012_05M-12MferWD-Weekday-00"
                }
            },
            {
                "_index": "bus",
                "_type": "stoptimes",
                "_id": "AWEpRL-pl_T0JDNwWjbn",
                "_score": 9.38033,
                "_source": {
                    "trip_id": "5808038-2012_05M-12MferWD-Weekday-00"
                }
            },
```



b. Find buses between two-time ranges

Query:



Response:


```
{
    "took": 12,
    "timed_out": false,
    "_shards": {
        "total": 3,
        "successful": 3,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": 62173,
        "max_score": 1,
        "hits": [
            {
                "_index": "bus",
                "_type": "stoptimes",
                "_id": "AWEpQbrpl_T0JDNwWWgv",
                "_score": 1,
                "_source": {
                    "trip_id": "6516065-2012_05M-1205BRsa-Saturday-02"
                }
            },
```



```
{
    "took": 14,
    "timed_out": false,
    "_shards": {
        "total": 3,
        "successful": 3,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": 3,
        "max_score": 13.39075,
        "hits": [
            {
                "_index": "bus",
                "_type": "trips",
                "_id": "AWEqBLahGt-pg_IMqQyI",
                "_score": 13.39075,
                "_source": {
                    "trip_headsign": "52 CROSSTOWN TO BURNSIDE"
                }
```

c. List all the bus-stops in the 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).

Query:

```
{
  "query": {
    "bool": {
      "must": [
        {"match_phrase": {"trip_headsign": "14 DOWNTOWN"}},
        {"match_phrase": {"route_id": "14-114"} }
      ]
    }
  }
}
```

Authorization ● Headers (2) Body ● Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
{ "query": { "match_phrase": { "trip_id": "6524119-2012_08A-1208BRwd-Weekday-01" } }, "_source": ["stop_sequence", "stop_id"] }
```

Authorization ● Headers (2) Body ● Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
{ "query": { "match_phrase": { "stop_id": "6900" } }, "_source": ["name_stop"] }
```

Response:


```
            "_index": "bus",
            "_type": "trips",
            "_id": "AWEp_-EgGt-pg_IMqLGV",
            "_score": 11.593606,
            "_source": {
                "block_id": "b_2153057",
                "route_id": "14-114",
                "trip_headsign": "14 DOWNTOWN",
                "service_id": "2012_08A-1208BRsu-Sunday-01",
                "shape_id": "140095",
                "trip_id": "6529207-2012_08A-1208BRsu-Sunday-01"
            }
        },
        {
            "_index": "bus",
            "_type": "trips",
            "_id": "AWEp_-EhGt-pg_IMqLW0",
            "_score": 11.593606,
            "_source": {
                "block_id": "b_2152465",
                "route_id": "14-114",
                "trip_headsign": "14 DOWNTOWN",
                "service_id": "2012_08A-1208BRwd-Weekday-01",
                "shape_id": "140095",
                "trip_id": "6524119-2012_08A-1208BRwd-Weekday-01"
            }
        }
    ]
  }
}
```

```
            {
                "_index": "bus",
                "_type": "stoptimes",
                "_id": "AWEpesXiGt-pg_IMoaPt",
                "_score": 12.58309,
                "_source": {
                    "stop_sequence": 35,
                    "stop_id": 6900
                }
```

```
    "hits": {
        "total": 1,
        "max_score": 7.156366,
        "hits": [
            {
                "_index": "bus",
                "_type": "stops",
                "_id": "AWEpPq1OJEZlZkR5PWVh",
                "_score": 7.156366,
                "_source": {
                    "name_stop": "herring Cove Rd before and opposite Melwood Ave"
                }
            }
        ]
    }
}
```


d. Find top 3 bus stops that are the busiest throughout the day.

Query:

```
{
  "size": 0,
  "aggs": {
    "group_by_stop_id": {
      "terms": {
        "field": "stop_id"
      }
    }
  }
}
```

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
{ "query": { "match_phrase": { "stop_id": "8643" } }, "_source": ["name_stop"] }
```

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
{ "query": { "match_phrase": { "stop_id": "6105" } }, "_source": ["name_stop"] }
```

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1  { "query": { "match_phrase": { "stop_id": "6108" } }, "_source": ["name_stop"]  }
```

Response:


```
"hits": {
    "total": 1,
    "max_score": 5.3270617,
    "hits": [
        {
            "_index": "bus",
            "_type": "stops",
            "_id": "AWEpPq1QJEZlZkR5PWok",
            "_score": 5.3270617,
            "_source": {
                "name_stop": "mumford Terminal [outbound In Terminal]"
            }
        }
    ]
}
```

```
"hits": {
    "total": 1,
    "max_score": 5.397235,
    "hits": [
        {
            "_index": "bus",
            "_type": "stops",
            "_id": "AWEpPq1KJEZlZkR5PWJW",
            "_score": 5.397235,
            "_source": {
                "name_stop": "barrington St [southbound] before Duke St"
            }
        }
    ]
}
```

```
"hits": {
    "total": 1,
    "max_score": 5.397235,
    "hits": [
        {
            "_index": "bus",
            "_type": "stops",
            "_id": "AWEpPq1KJEZlZkR5PWJW",
            "_score": 5.397235,
            "_source": {
                "name_stop": "barrington St [southbound] before Duke St"
            }
        }
    ]
}
```

Summary:

Here I have performed and compared the queries in RDBM and elastic search and so far, I have found that Relational database is easy for the low volume of data and its syntax is easy to grasp, and intuitive that one can perform the joint operation without many efforts.

Elastic search on the other side can handle a large volume of the data. It provides real-time search and data analysis. It is less expensive when it comes to big data handling and low hardware cost and provides accurate free text-search. But it can't handle joint query and it is complex for easy operations and not suitable for relational databases operation.

In addition, cloud platforms are more useful while dealing with the big amount of data, and for the small-scale operations, offline tools such as DB Brower for SQLite *[6]* are much more helpful for the beginners as they give less error for the action and there isn't any issue of server downtime.

References:

[1] Transit. Internet: https://transitapp.com/ [Jan 24, 2018]

[2] IBM Inc. ‘IBM Bluemix’. Internet: https://www.ibm.com/cloud/ [Jan 24, 2018]

[3] ‘Relational Database Management System’.

Internet: https://en.wikipedia.org/wiki/Relational_database_management_system [Jan 23, 2018]

[4] Elasticsearch BV.

Internet:https://www.elastic.co/guide/en/elasticsearch/reference/2.3/_introducing_the_query_language.html [Jan 24-25, 2018]

[5] Postdot Technologies Inc. ‘Postman’ Internet: https://www.getpostman.com/docs/ [Jan 24-25, 2018]

[6] DB Browser for SQLite. Internet: http://sqlitebrowser.org/ [Jan 24-25, 2018]