

## CSCI 5408

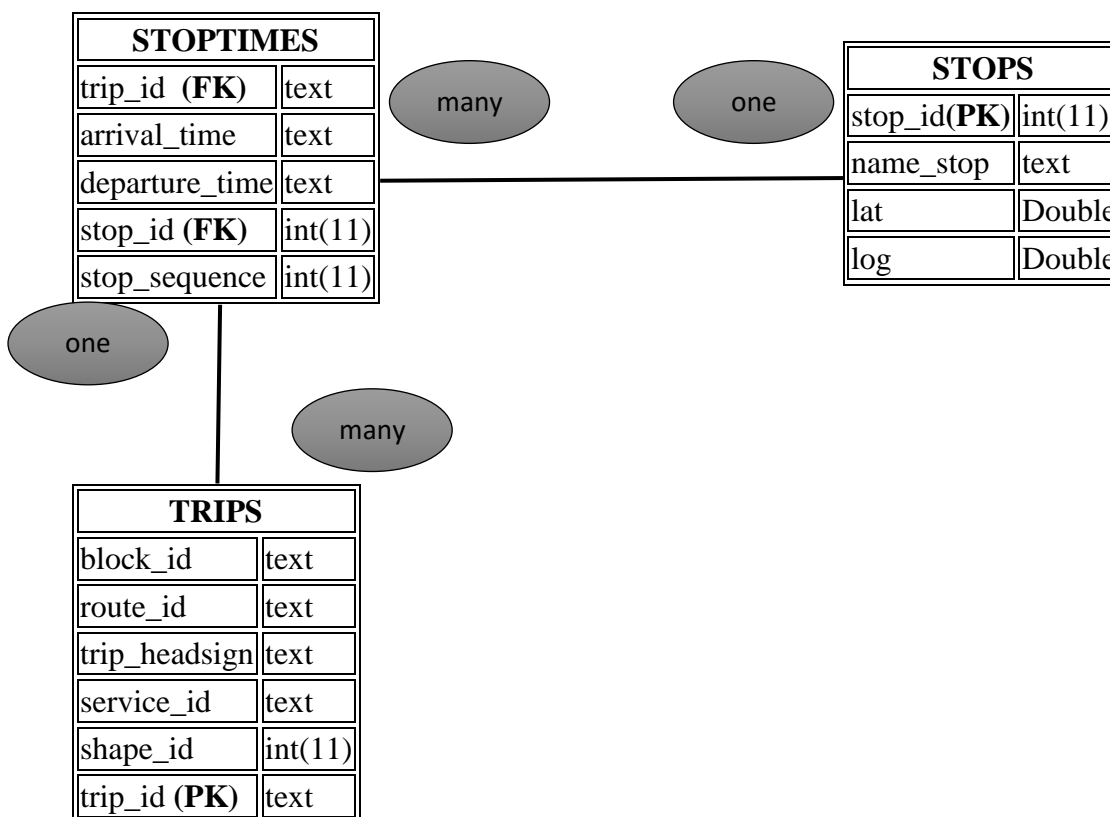
### Assignment 1: Query Implementation using Relational Database and Elastic Search

#### 1. TASK DESCRIPTION

The task is to get the bus routes dataset and perform set of queries using MySQL in local machine and Elastic search in AWS EC2 instance.

This task is mainly to experiment and understand the difference between SQL and NoSQL queries in terms of performance, efficiency, time consumption, resource utilization and other performance categories.

#### 2. RELATIONAL DATABASE DESIGN



#### Normalization

The above tables are already decomposed and there are no redundant data in the tables, hence the tables cannot be normalised further.

#### 3. MYSQL QUERIES

- Find all buses for a particular Bus Stop

Input: Bus Stop Name

Output: List of all buses, response time for the search query

*Fetching all Columns*

```
select * from stops INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop_id INNER JOIN trips
```

**ON stoptimes.trip\_id=trips.trip\_id where stops.name\_stop like ' Ferry Stop - Halifax';**

12 `select * from stops INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop_id INNER JOIN trips`  
 13 `ON stoptimes.trip_id=trips.trip_id where stops.name_stop like ' Ferry Stop - Halifax';`

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

stop_id	name_stop	lat	log	trip_id	arrival_time	departure_time	stop_id	stop_sequence	block_id	route_id
1073	Ferry Stop - Halifax	44.6495	-63.5729	6219236-2012_08A-12AferSA-Saturday-01	11:12:00	11:12:00	1073	2	07:37:00	6522959-2012_08A-1208E
1073	Ferry Stop - Halifax	44.6495	-63.5729	6219239-2012_08A-12AferSA-Saturday-01	12:42:00	12:42:00	1073	2	21:44:00	6528151-2012_08A-1208E
1073	Ferry Stop - Halifax	44.6495	-63.5729	6219188-2012_08A-12AferWD-Weekday-01	18:45:00	18:45:00	1073	1	22:25:00	6523372-2012_08A-1208E
1073	Ferry Stop - Halifax	44.6495	-63.5729	5808021-2012_05M-12MferWD-Weekday-00	08:49:00	08:49:00	1073	2	21:10:00	6515902-2012_05M-1205E
1073	Ferry Stop - Halifax	44.6495	-63.5729	5808054-2012_05M-12MferSA-Saturday-00	11:12:00	11:12:00	1073	2	12:57:00	6523001-2012_08A-1208E
1073	Ferry Stop - Halifax	44.6495	-63.5729	6219340-2012_08A-12AferSU-Sunday-01	08:15:00	08:15:00	1073	1	08:44:00	6516550-2012_05M-1205E
1073	Ferry Stop - Halifax	44.6495	-63.5729	6219159-2012_08A-12AferWD-Weekday-01	10:15:00	10:15:00	1073	1	09:12:00	6516550-2012_05M-1205E
1073	Ferry Stop - Halifax	44.6495	-63.5729	6219131-2012_08A-12AferWD-Weekday-01	16:57:00	16:57:00	1073	2	15:32:00	6530381-2012_08A-1208E

Result 41 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
69	16:10:25	select * from stops where name_stop like ' Ferry Stop - Halifax' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
70	16:10:48	select * from stops INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop_id INNER J...	63 row(s) returned	594.187 sec / 0.000 sec

**select trips.trip\_headsign from stops INNER JOIN stoptimes ON stops.stop\_id=stoptimes.stop\_id  
INNER JOIN trips ON stoptimes.trip\_id=trips.trip\_id where stops.name\_stop like ' Ferry Stop - Halifax';**

*Fetching only the list of buses for a particular bus\_stop the execution time shown in snapshot is 370.766 sec.*

14 `select trips.trip_headsign from stops INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop_id INNER JOIN trips`  
 15 `ON stoptimes.trip_id=trips.trip_id where stops.name_stop like ' Ferry Stop - Halifax';`  
 16  
 17

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

trip_headsign
41
19
69
44
4
40
87
5
10

Result 42 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
70	16:10:48	select * from stops INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop_id INNER J...	63 row(s) returned	594.187 sec / 0.000 sec
71	16:30:29	select trips.trip_headsign from stops INNER JOIN stoptimes ON stops.stop_id=stoptimes...	63 row(s) returned	370.766 sec / 0.000 sec

b. Find buses between two time ranges

1. Input: Time Range 1 (hh:mm:ss), Time Range 2 (hh:mm:ss)
2. Output: List of all buses, response time for the search query

**Select trips.trip\_headsign, trips.trip\_id from trips**

**inner join stoptimes on trips.trip\_id=stoptimes.trip\_id**

**where stoptimes.arrival\_time between '13:00:00' and '20:00:00';**

The screenshot shows a database query interface. At the top, a SQL query is entered: `Select trips.trip_headsign, trips.trip_id from trips inner join stoptimes on trips.trip_id=stoptimes.trip_id where stoptimes.arrival_time between '13:00:00' and '20:00:00';`. Below the query, a 'Result Grid' displays the results. The grid has two columns: 'trip\_headsign' and 'trip\_id'. The results are as follows:

trip_headsign	trip_id
330 HALIFAX	6513332-2012_05M-1205BRwd-Weekday-02
330 HALIFAX	6513332-2012_05M-1205BRwd-Weekday-02
61 DOWNTOWN HALIFAX TO SCOTIA SQUARE	6518384-2012_05M-1205BRsu-Sunday-02
61 DOWNTOWN HALIFAX TO SCOTIA SQUARE	6518384-2012_05M-1205BRsu-Sunday-02
61 DOWNTOWN HALIFAX TO SCOTIA SQUARE	6518384-2012_05M-1205BRsu-Sunday-02
61 DOWNTOWN HALIFAX TO SCOTIA SQUARE	6518384-2012_05M-1205BRsu-Sunday-02

Below the result grid, an 'Output' section shows the execution log. It includes a table with columns: '#', 'Time', 'Action', 'Message', and 'Duration / Fetch'.

#	Time	Action	Message	Duration / Fetch
78	16:47:57	select count(*) from stoptimes where arrival_time between '13:00:00' and '21:00:00' LIM...	1 row(s) returned	0.516 sec / 0.000 sec
79	16:49:03	Select trips.tripheadsign, trips.trip_id from trips inner join stoptimes on trips.trip_id=stopti...	Error Code: 1054. Unknown column 'trips.tripheadsign' in 'field list'	0.000 sec
80	16:49:26	Select trips.trip_headsign, trips.trip_id from trips inner join stoptimes on trips.trip_id=stopti...	1000 row(s) returned	0.282 sec / 0.453 sec

**Select trips.trip\_headsign, stoptimes.arrival\_time from trips**

**inner join stoptimes on trips.trip\_id=stoptimes.trip\_id**

**where stoptimes.arrival\_time between '13:00:00' and '20:00:00';**

*This query is to select list of all buses between time interval as shown in the query, it takes 0.359 seconds to execute the query as shown below.*

The screenshot shows a database query interface. At the top, a SQL query is entered: `Select trips.trip_headsign, stoptimes.arrival_time from trips inner join stoptimes on trips.trip_id=stoptimes.trip_id where stoptimes.arrival_time between '13:00:00' and '20:00:00';`. Below the query, a 'Result Grid' displays the results. The grid has two columns: 'trip\_headsign' and 'arrival\_time'. The results are as follows:

trip_headsign	arrival_time
330 HALIFAX	13:00:00
330 HALIFAX	13:32:00
61 DOWNTOWN HALIFAX TO SCOTIA SQUARE	18:27:00
61 DOWNTOWN HALIFAX TO SCOTIA SQUARE	18:27:00
61 DOWNTOWN HALIFAX TO SCOTIA SQUARE	18:28:00
61 DOWNTOWN HALIFAX TO SCOTIA SQUARE	18:28:00

Below the result grid, an 'Output' section shows the execution log. It includes a table with columns: '#', 'Time', 'Action', 'Message', and 'Duration / Fetch'.

#	Time	Action	Message	Duration / Fetch
79	16:49:03	Select trips.tripheadsign, trips.trip_id from trips inner join stoptimes on trips.trip_id=stopti...	Error Code: 1054. Unknown column 'trips.tripheadsign' in 'field list'	0.000 sec
80	16:49:26	Select trips.trip_headsign, trips.trip_id from trips inner join stoptimes on trips.trip_id=stopti...	1000 row(s) returned	0.282 sec / 0.453 sec
81	16:51:58	Select trips.trip_headsign, stoptimes.arrival_time from trips inner join stoptimes on trips.tri...	1000 row(s) returned	0.359 sec / 0.235 sec

**Select count(\*) from trips**

**inner join stoptimes on trips.trip\_id=stoptimes.trip\_id**

**where stoptimes.arrival\_time between '13:00:00' and '20:00:00';**

11  
12 Select count(\*) from trips inner join stoptimes on trips.trip\_id=stoptimes.trip\_id  
13 where stoptimes.arrival\_time between '13:00:00' and '20:00:00';  
14

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

count(*)
133912

Result 52 x Read On

Output

#	Time	Action	Message	Duration / Fetch
80	16:49:26	Select trips.trip_headsign, trips.trip_id from trips inner join stoptimes on trips.trip_id=stopti...	1000 row(s) returned	0.282 sec / 0.453 sec
81	16:51:58	Select trips.trip_headsign, stoptimes.arrival_time from trips inner join stoptimes on trips.tri...	1000 row(s) returned	0.359 sec / 0.235 sec
82	16:53:53	Select count(*) from trips inner join stoptimes on trips.trip_id=stoptimes.trip_id where stop...	1 row(s) returned	213.891 sec / 0.000 sec

*It takes 213 seconds to fetch all the records, the time difference between different execution of queries is because MYSQL is set to display only 1000 records at a time so it does not fetch all the matching records. This should be considered while comparing the performance.*

c. Find route information of a particular bus on a particular route

1. Input: Bus Name, Route Name

2. Output: List of all routes, response time for the search query

**select trips.route\_id, stops.name\_stop from stops**

**INNER JOIN stoptimes ON stops.stop\_id=stoptimes.stop\_id**

**INNER JOIN trips ON stoptimes.trip\_id=trips.trip\_id**

**where trips.trip\_headsign like '1' and trips.route\_id like '6522264-2012\_08A-1208BRwd-Weekday-01';**

*Below query fetches list of all the routes given the bus name and route name (route\_id) as given in the question, time taken is 86 seconds as shown below.*

```

28 select trips.route_id, stops.name_stop from stops INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop_id INNER JOIN trips
29 ON stoptimes.trip_id=trips.trip_id where trips.trip_headsign like '1' and trips.route_id like '6522264-2012_08A-1208BRwd-Weekday-01';

```

route_id	name_stop
6522264-2012_08A-1208BRwd-Weekday-01	water St Terminal
6522264-2012_08A-1208BRwd-Weekday-01	duke St [westbound] before Market St
6522264-2012_08A-1208BRwd-Weekday-01	cogswell St [westbound] after Gottingen St
6522264-2012_08A-1208BRwd-Weekday-01	cogswell St [westbound] before North Park St
6522264-2012_08A-1208BRwd-Weekday-01	cogswell St [westbound] opposite Common

Result 60

#	Time	Action	Message	Duration / Fetch
88	17:20:01	select route_id from trips where trip_headsign like '1' LIMIT 0, 1000	146 row(s) returned	0.031 sec / 0.000 sec
89	17:22:10	select route_id from trips where trip_headsign like '1' and route_id like '6522264-2012_0...	1 row(s) returned	0.032 sec / 0.000 sec
90	17:35:58	select trips.route_id, stops.name_stop from stops INNER JOIN stoptimes ON stops.stop...	37 row(s) returned	86.375 sec / 0.000 sec

**select stops.name\_stop from stops**

**INNER JOIN stoptimes ON stops.stop\_id=stoptimes.stop\_id**

**INNER JOIN trips ON stoptimes.trip\_id=trips.trip\_id**

**where trips.trip\_headsign like '1' and trips.route\_id like '6522264-2012\_08A-1208BRwd-Weekday-01';**

```

28 select stops.name_stop from stops
29 INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop_id
30 INNER JOIN trips ON stoptimes.trip_id=trips.trip_id
31 where trips.trip_headsign like '1' and trips.route_id like '6522264-2012_08A-1208BRwd-Weekday-01';
32

```

name_stop
water St Terminal
duke St [westbound] before Market St
cogswell St [westbound] after Gottingen St
cogswell St [westbound] before North Park St
cogswell St [westbound] opposite Common

Result 61

#	Time	Action	Duration / Fetch	Message
89	17:22:10	select route_id from trips where trip_headsign like '1' and route_id like '6522264-2012_08A-...	0.032 sec / 0.000 sec	1 row(s) returned
90	17:35:58	select trips.route_id, stops.name_stop from stops INNER JOIN stoptimes ON stops.stop_id=...	86.375 sec / 0.000 sec	37 row(s) returned
91	17:49:40	select stops.name_stop from stops INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop...	75.797 sec / 0.000 sec	37 row(s) returned

**select stoptimes.stop\_id, stoptimes.stop\_sequence, stops.name\_stop from stops**

**INNER JOIN stoptimes ON stops.stop\_id=stoptimes.stop\_id**

**INNER JOIN trips ON stoptimes.trip\_id=trips.trip\_id**

**where trips.trip\_headsign like '1' and trips.route\_id like '6522264-2012\_08A-1208BRwd-Weekday-01';**

```

22 #c query
23 select stoptimes.stop_id, stoptimes.stop_sequence, stops.name_stop from stops
24 INNER JOIN stoptimes ON stops.stop_id=stoptimes.stop_id
25 INNER JOIN trips ON stoptimes.trip_id=trips.trip_id
26 where trips.trip_headsign like '1' and trips.route_id like '6522264-2012_08A-12088Rwd-Weekday-01';
27
28

```

stop_id	stop_sequence	name_stop
8435	1	water St Terminal
6590	2	duke St [westbound] before Market St
6456	3	cogswell St [westbound] after Gottingen St
6460	4	cogswell St [westbound] before North Park St

Result 1 x

Output

#	Time	Action	Duration / Fetch	Message
3	19:17:34	select stops.name_stop, trips.trip_headsign, trips.route_id, count(trips.route...	600.000 sec	Error Code: 2013. Lost connection to MySQL server during query
4	19:28:23	select stops.name_stop, trips.trip_headsign, trips.route_id, count(trips.route...	5.953 sec	OK
5	19:28:29	INTERRUPT	0.000 sec	OK - Query cancelled
6	19:28:41	select stops.name_stop, trips.trip_headsign, trips.route_id, count(trips.route...	600.047 sec	Error Code: 2013. Lost connection to MySQL server during query
7	19:45:54	select stops.name_stop, trips.trip_headsign, trips.route_id, count(trips.route...	600.047 sec	Error Code: 2013. Lost connection to MySQL server during query
8	21:07:53	select stoptimes stop_id, stoptimes stop_sequence, stops.name_stop from s...	235.422 sec / 0.000 sec	37 row(s) returned

Time duration: 235 seconds.

d. Find top 3 bus stops that are the busiest throughout the day in terms of bus routes. (Hint: The bus stops with high volume of bus routes and close time gaps would be considered as busiest).

1. Input: None

2. Output: List of Bus Name, response time for the search query.

**select stops.name\_stop, trips.trip\_headsign, trips.route\_id, count(trips.route\_id) from trips**

**INNER JOIN stoptimes ON trips.trip\_id=stoptimes.trip\_id**

**inner join stops on stops.stop\_id=stoptimes.stop\_id**

**group by(trips.route\_id) order by count(trips.route\_id) desc LIMIT 3;**

*The above query as a single query is not getting executed. The query runs for 600 seconds after which it says connection lost. Hence, I am splitting the above query as two parts as shown below*

**select trip\_id, route\_id, count(route\_id) from trips**

**group by route\_id order by count(route\_id) desc LIMIT 1;**

*The above query is collecting the trip\_id which has the maximum number of routes. This query takes 0.250 seconds to execute.*

```

6 select trip_id, route_id, count(route_id) from trips
7 group by route_id order by count(route_id) desc LIMIT 1;

```

trip_id	route_id	count(route_id)
6514048-2012_05M-1205BRwd-Weekday-02	1-121	169

Result 13 x

Output

Action Output

#	Time	Action	Message	Duration /
33	20:25:44	select stops.name_stop from stops INNER JOIN stoptimes on stoptimes.stop_id=s...	3 row(s) returned	0.047 sec
34	21:05:27	select trip_id, route_id, count(route_id) from trips group by route_id order by count(...	1 row(s) returned	0.250 sec

**select trip\_id, route\_id, count(route\_id) from trips**  
**group by route\_id order by count(route\_id) desc LIMIT 1;**

Three bus stops that are part of the trip\_id with maximum number of routes is selected below. The query takes 0.047 seconds to execute as shown below.

```

9 select stops.name_stop from stops
10 INNER JOIN stoptimes on stoptimes.stop_id=stops.stop_id
11 where trip_id like '6514048-2012_05M-1205BRwd-Weekday-02' LIMIT 3;

```

name_stop
mumford Terminal Downtown [in Terminal]
east Perimeter Rd after Mumford Rd
east Perimeter Rd 2nd Stop

Result 12 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
32	20:25:16	select trip_id, route_id, count(route_id) from trips group by route_id order by count(...	3 row(s) returned	0.250 sec / 0.000 sec
33	20:25:44	select stops name_stop from stops INNER JOIN stoptimes on stoptimes stop_id=s...	3 row(s) returned	0.047 sec / 0.000 sec

## 4. ELASTICSEARCH QUERIES

### CLUSTER STATE - USING SENSE EXTENTION

Server  History

```
1 GET _cluster/state
2
3 POST /bustracks/_search
4 {
5   "query": {
6     "match": {
7       "name": "dartmouth Bridge Terminal - Bay 15"
8     }
9   }
10 }
11 POST /bustracks/trips/_search
12 {
13   "query": {
14     "match": {
15       "route_id": "1-114"
16     }
17   }
18 }
19 POST /bustracks/trips/_search
20 {
21   "query": {
22     "match": {
23       "trip_id": "6525359-2012_08A-1208BRwd-Weekday-01"
24     }
25   }
26 }
27 POST /bustracks/trips/_search
28 {
```

```
1 {
2   "cluster_name": "my-application",
3   "version": 9,
4   "state_uuid": "RwqajCaIQwiBrH5Q60PSPA",
5   "master_node": "RDSezw-XR-W6E_dNmpjYw",
6   "blocks": {},
7   "nodes": {
8     "RDSezw-XR-W6E_dNmpjYw": {
9       "name": "node-1",
10      "transport_address": "172.31.43.95:9300",
11      "attributes": {}
12    }
13  },
14  "metadata": {
15    "cluster_uuid": "9Gu70qpWSMciQoGxeqa1Fg",
16    "templates": {},
17    "indices": {
18      "bustrack": {
19        "state": "open",
20        "settings": {
21          "index": {
22            "creation_date": "1485276284393",
23            "number_of_shards": "5",
24            "number_of_replicas": "1",
25            "uuid": "ruF08aVeTkG4FpPJ0eK6Vg",
26            "version": {
27              "created": "2040499"
28            }
29          }
30        }
31      }
32    }
33  }
```

### SAMPLE QUERY

Server

```
1 GET _cluster/state
2
3 POST /bustracks/_search
4 {
5   "query": {
6     "match": {
7       "name": "dartmouth Bridge Terminal - Bay 15"
8     }
9   }
10 }
11 POST /bustracks/trips/_search
12 {
13   "query": {
14     "match": {
15       "route_id": "1-114"
16     }
17   }
18 }
19 POST /bustracks/trips/_search
20 {
21   "query": {
22     "match": {
23       "trip_id": "6525359-2012_08A-1208BRwd-Weekday-01"
24     }
25   }
26 }
27 POST /bustracks/trips/_search
28 {
```

```
1 {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "failed": 0
8   },
9   "hits": {
10    "total": 191,
11    "max_score": 9.621441,
12    "hits": [
13      {
14        "_index": "bustracks",
15        "_type": "stops",
16        "_id": "7615",
17        "_score": 9.621441,
18        "_source": {
19          "stop_id": "7615",
20          "name": "dartmouth bridge terminal - bay 15",
21          "latt": "44.67075",
22          "log": "-63.575199"
23        }
24      },
25      {
26        "_index": "bustracks",
27        "_type": "stops",
28        "_id": "7602",
```



The queries are written and executed in python. Python directly connected to instance using below code snippet. Hence the code and output are stored in file and embedded below.

```
from elasticsearch import Elasticsearch
```

```
es = Elasticsearch(['ec2-52-27-28-3.us-west-2.compute.amazonaws.com:9200'])
```

#### Query A – Python code with elastic search embedded

Execution time – 4 seconds



Query A - text file.txt



Query A Python  
Code.py

#### Output for Query A



Output of Query A.txt

#### Query B – Python code with elastic search embedded

Execution time – 5 seconds



Query B - text file.txt



Query B Python  
Code.py

#### Output for Query B



Output of Query B.txt

#### Query C – Python code with elastic search embedded

Execution time – 10 seconds



Query C - text file.txt



Query C Python  
Code.py

### Output for Query C



Output of Query C.txt

### Query D – Python code with elastic search embedded

Execution time – 400 seconds



Query D - text file.txt



Query D Python  
Code.py

### Output for Query D



Output of Query D.txt

## 5. COMPARISON OF PERFORMANCE AND EVALUATION

Time duration in seconds		
Query	SQL	NoSQL
A	370	4
B	213	5
C	235	10
D	0.29/600	400

*The time difference between execution of some MYSQL and NOSQL queries is because, MYSQL is set to display only 1000 records at a time so it does not fetch all the matching records. This should be considered while comparing the performance.*

*Taking this onto consideration, the NOSQL gives better performance*

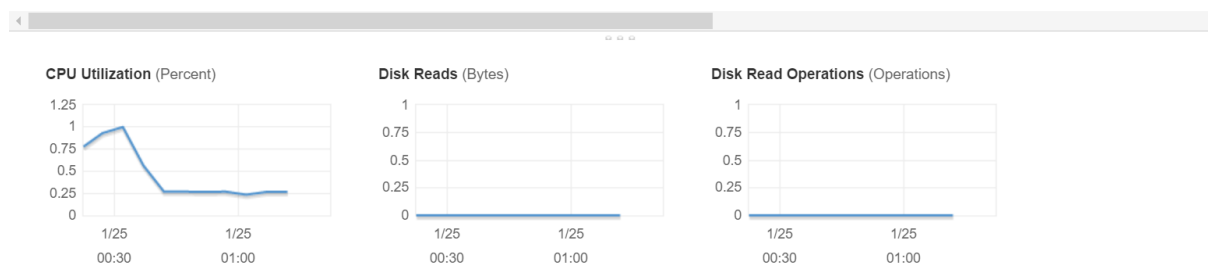
## LOCAL MACHINE CPU UTILIZATION FOR MYSQL WHILE EXECUTING A COMPLEX QUERY

Windows Start-Up Application	0%	0.1 MB	0 MB/s	0 Mbps
Client Server Runtime Process	0.2%	0.6 MB	0 MB/s	0 Mbps
Client Server Runtime Process	0%	0.5 MB	0 MB/s	0 Mbps
Windows Session Manager	0%	0.1 MB	0 MB/s	0 Mbps
> Task Manager	0.8%	12.8 MB	0 MB/s	0 Mbps
> Service Host: Local System (Net...	0.8%	69.5 MB	0 MB/s	0 Mbps
> Service Host: Local Service (Net...	0%	6.2 MB	0 MB/s	0 Mbps
> appmodel (2)	0%	2.5 MB	0.1 MB/s	0 Mbps
> Service Host: Windows Image A...	0%	0.5 MB	0.1 MB/s	0 Mbps
AVG Scanning Core Module - S...	0.2%	19.2 MB	0.1 MB/s	0 Mbps
AVG Resident Shield Service	0%	6.2 MB	0.1 MB/s	0 Mbps
System and compressed memory	1.0%	406.4 MB	0.1 MB/s	0 Mbps
> MySQL Workbench	0.1%	21.2 MB	0.1 MB/s	0 Mbps
> Google Chrome	0%	61.3 MB	0.1 MB/s	0 Mbps
VProtect Application (32 bit)	0%	6.9 MB	0.1 MB/s	0 Mbps
Cortana	5.5%	88.0 MB	0.1 MB/s	0 Mbps
> mysqld	77.2%	14.7 MB	0.2 MB/s	0 Mbps
> Microsoft Windows Search Inde...	6.8%	38.1 MB	0.3 MB/s	0 Mbps

Here the CPU utilization is 77% and mysqld utilizes 14 MB of memory and MySQL Workbench utilizes 21 MB of memory.

## AWS INSTANCE CPU UTILIZATION GRAPH SHOWN BELOW

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-07d1bf5a660da2b4c	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-52-27-28-3.t



## 6. SUMMARY OF OBSERVATIONS:

- Elasticsearch is efficient while searching indexed text and it goes slow while used inside a script to join the output of one search inside another.
- In short, this is not very efficient for performing joins which needs extra support of scripts or scripting languages like python.

- Using scripts adds a lot of additional overhead apart from searching.
- MySQL is efficient in performing joins and is less time consuming only with respect to joins.

**TOOLS:** MySQL Workbench, Python (import elasticsearch package)

## **7. AWS CONFIGURATION**

**PUBLIC IP of EC2 instance created: 52.27.28.3**

**DNS of EC2 instance created: ec2-52-27-28-3.us-west-2.compute.amazonaws.com**

### **TEAM:**

ANURAG TENKOTI (B00738753)

YAMUNA JAYABALAN (B00741912)

## **8. TASK BREAKDOWN**

ANURAG: SQL Queries - A, B, NOSQL , AWS Configuration

YAMUNA: SQL Queries – C, D, NOSQL Queries – A, B, C, D

REPORT: YAMUNA