Data Management, Warehousing, and Analytics Assignment 1

TEAM MEMBERS:

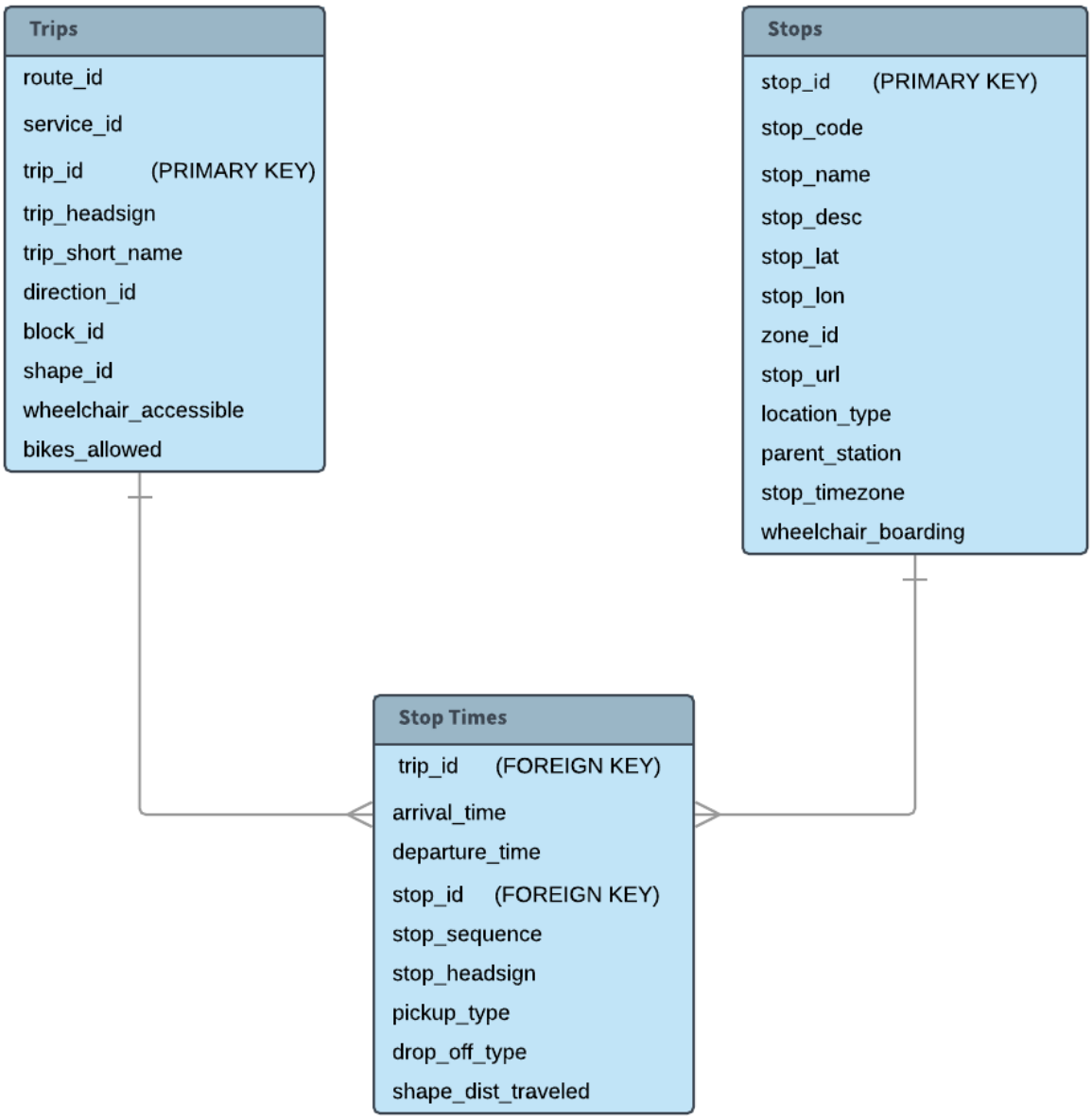
Aravind Govindarajan (B00803336)

Haritha Desikan (B00801722)

TASK DESCRIPTION

To get ourselves acquainted with a cloud computing platform and database concepts by running queries for a Transit dataset in local RDBMS (MySQL), IBM cloud RDBMS and performing Elasticsearch on Cloud using the Postman software and then compare the efficiency of the three methods with the execution time taken for each query.

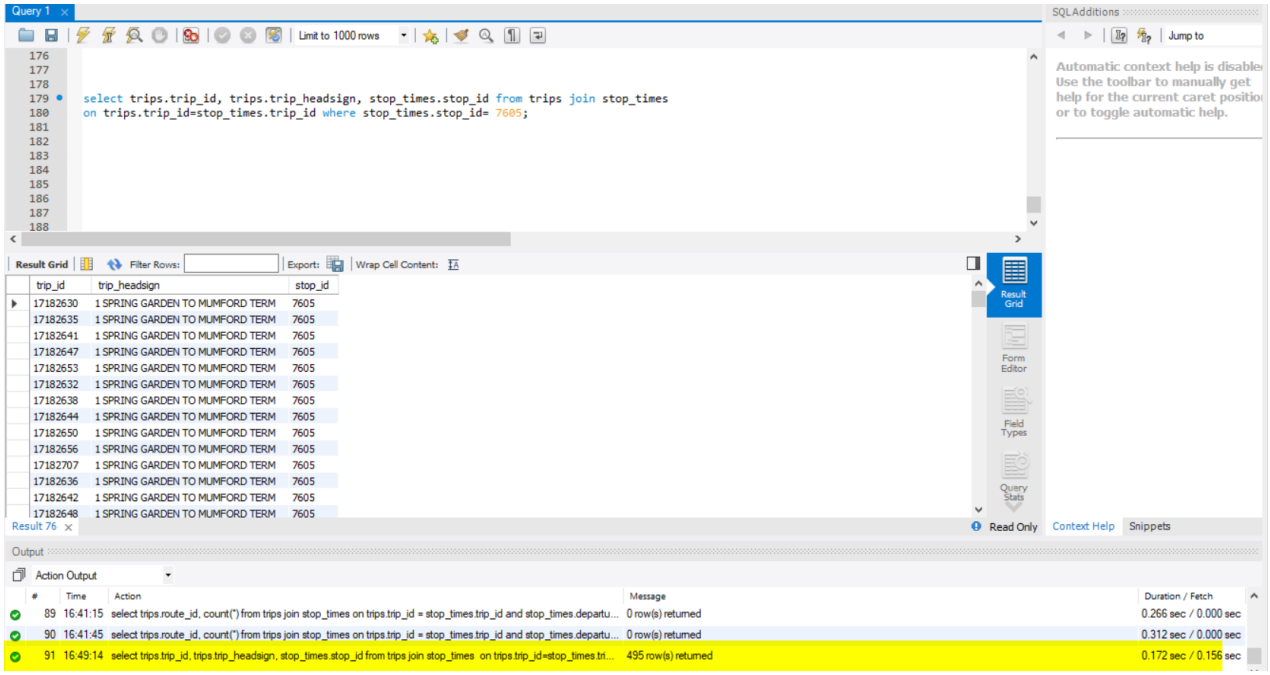
RELATIONAL DATABASE DESIGN (ENTITY RELATIONSHIP DIAGRAM)



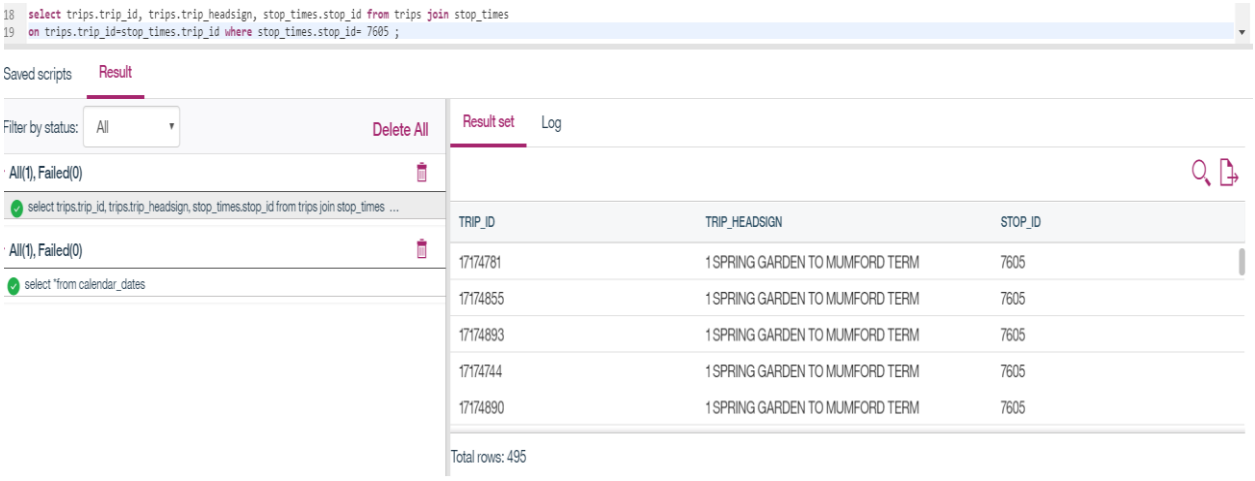
QUERIES

1. We are asked to identify all routes and buses that goes via a specific stop from the Transit database in this question. For this, we have given a stop id as input with the where clause and obtained trip id, trip head sign along with the stop id. We have made use of joins to combine trips and stop_times tables. The Response time taken for this query to run in local RDBMS and cloud are 172 millisecs and 53 millisecs.

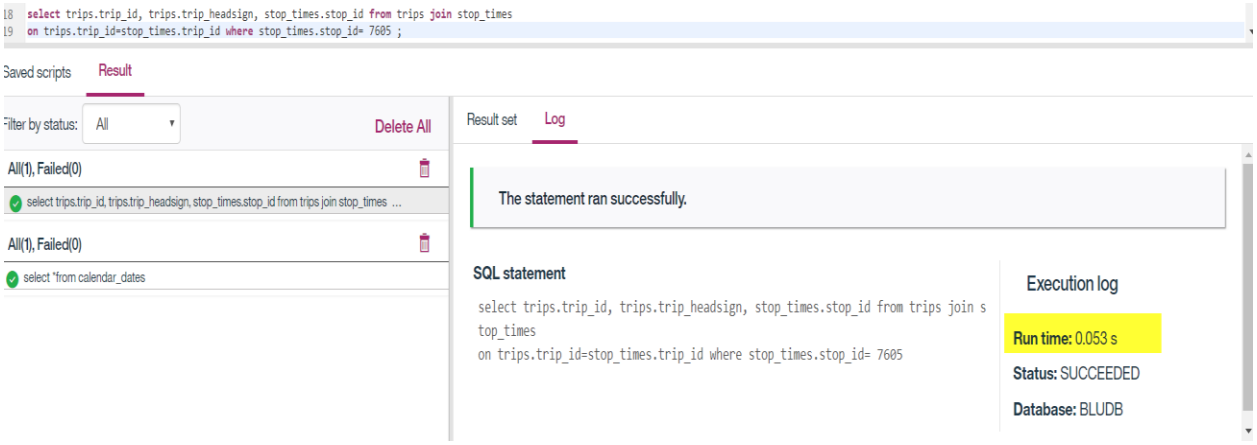
USING LOCAL RDBMS



USING CLOUD RDBMS (OUTPUT SCREEN)

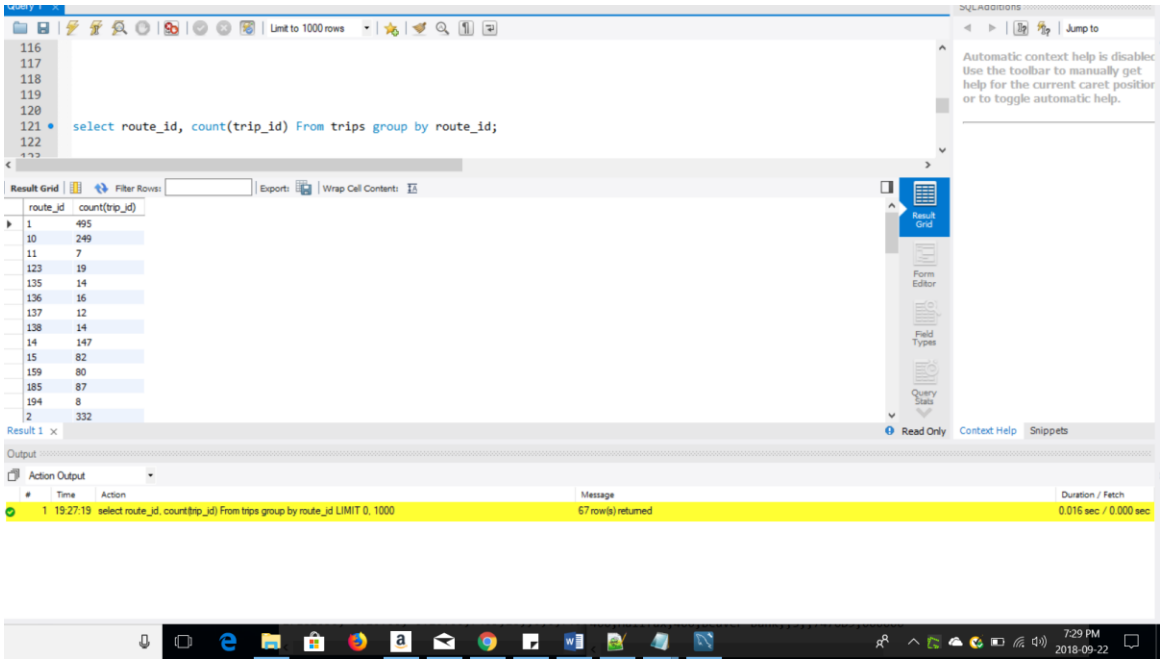


USING CLOUD RDBMS (RESPONSE TIME)

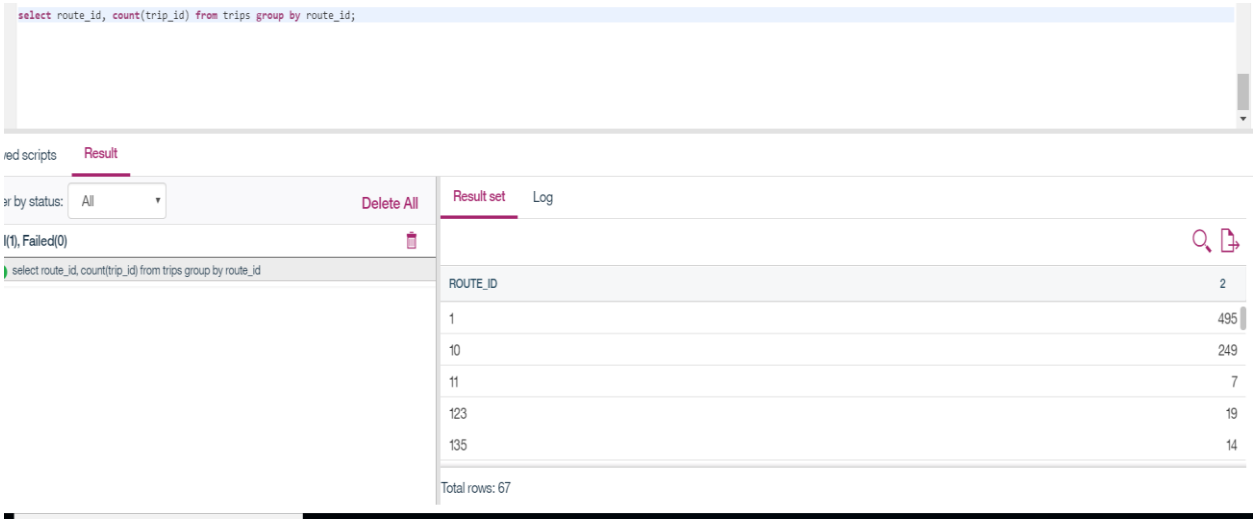


2. We are asked to find the total number of trips on each route in this question. We have used trips table in the query and obtained the count of total trips in each route. Response time taken to run this query in local RDBMS and cloud RDBMS are 160 millisecs and 18 millisecs.

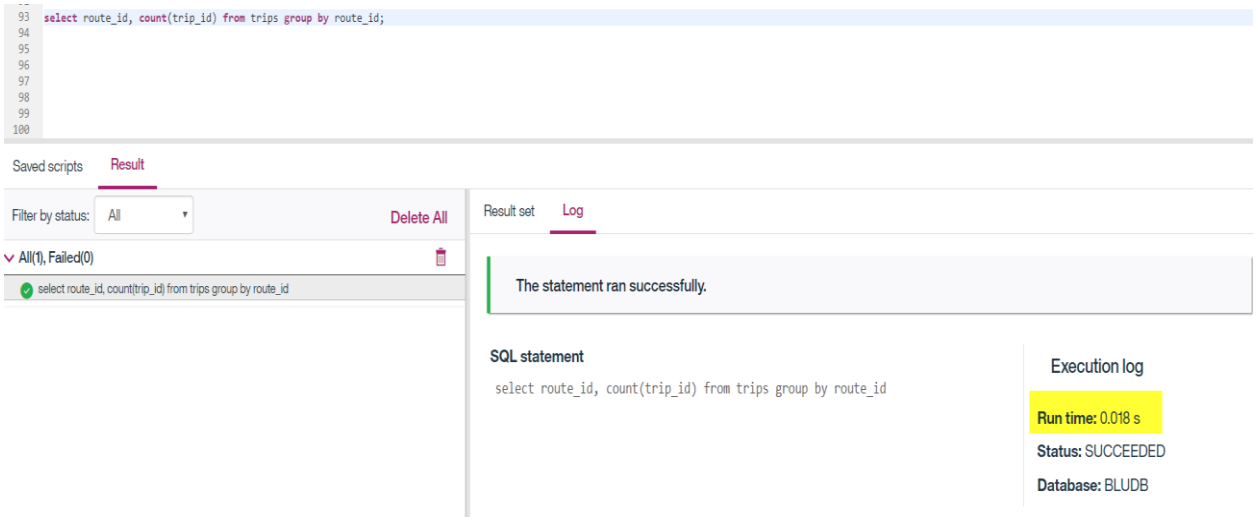
USING LOCAL RDBMS



USING CLOUD RDBMS (OUTPUT SCREEN)

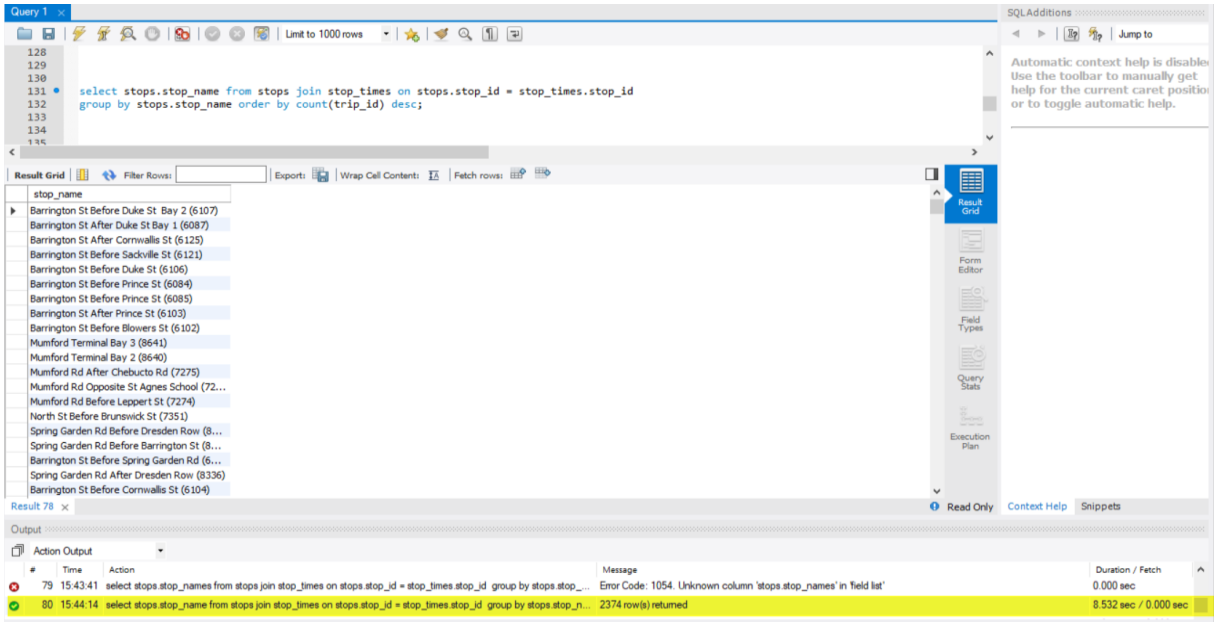


USING CLOUD RDBMS (RESPONSE TIME)

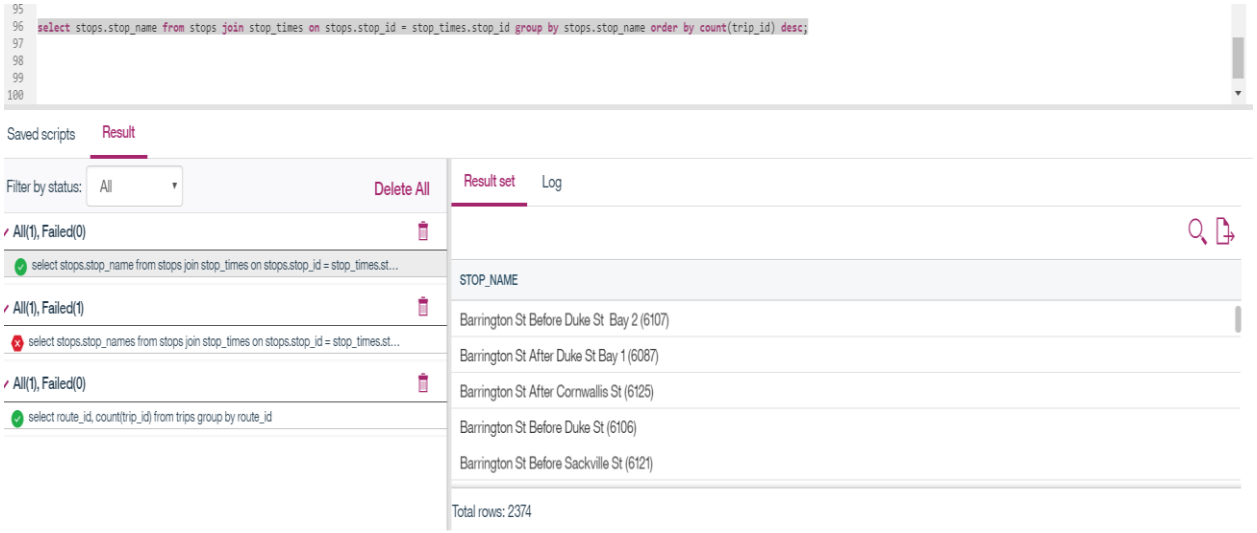


3. We are asked to find the most and least busy bus stops in a day in this question. For this, we sort the stop names in descending order of the count of trips. We use join in this query to combine stops and stop_times tables. Response time for executing this query in local RDBMS and cloud RDBMS are 853.2 millisecs and 63 millisecs.

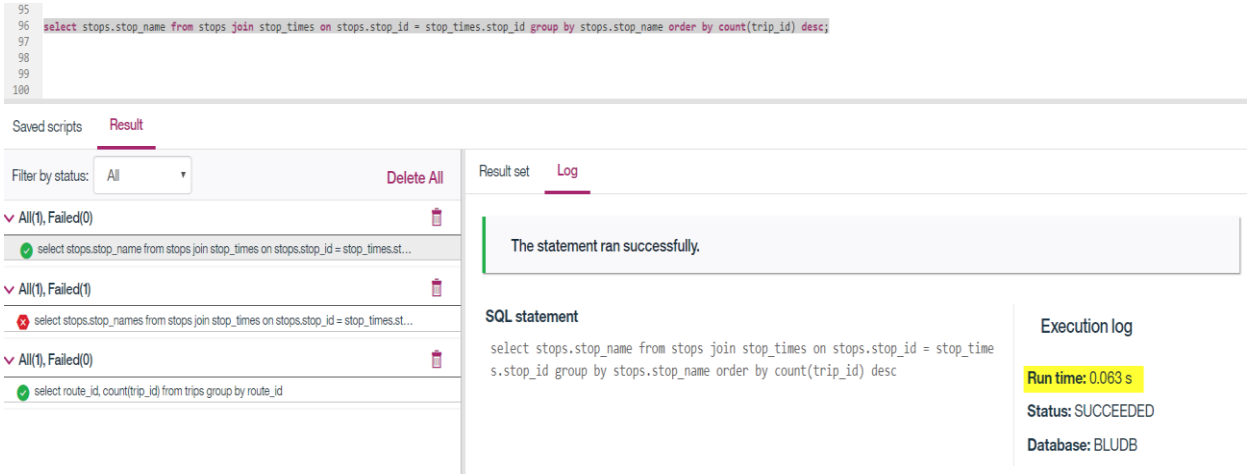
USING LOCAL RDBMS



USING CLOUD RDBMS (OUTPUT SCREEN)

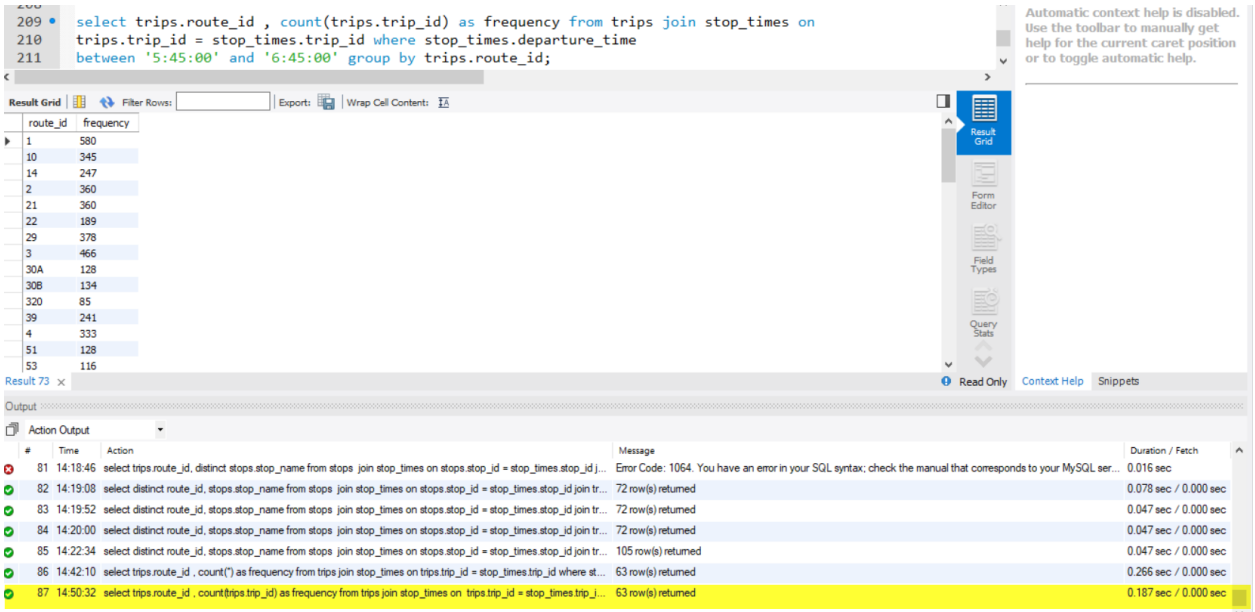


USING CLOUD RDBMS (RESPONSE TIME)

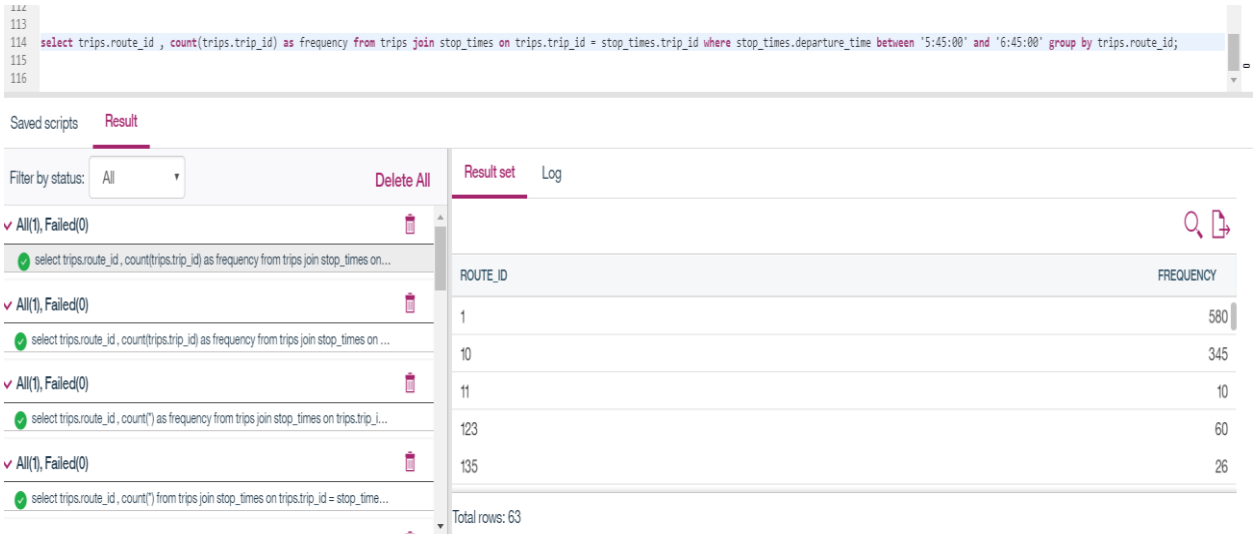


4. We must find the frequency of a bus service over a time range in this question. For this purpose, we give a route identifier and time range as input and get the number of times a bus route service is available in that time range. The query is written with a join combining trips and stop_times tables. Response time for running this query in local RDBMS and cloud RDBMS are 31 millisecs and 26 millisecs.

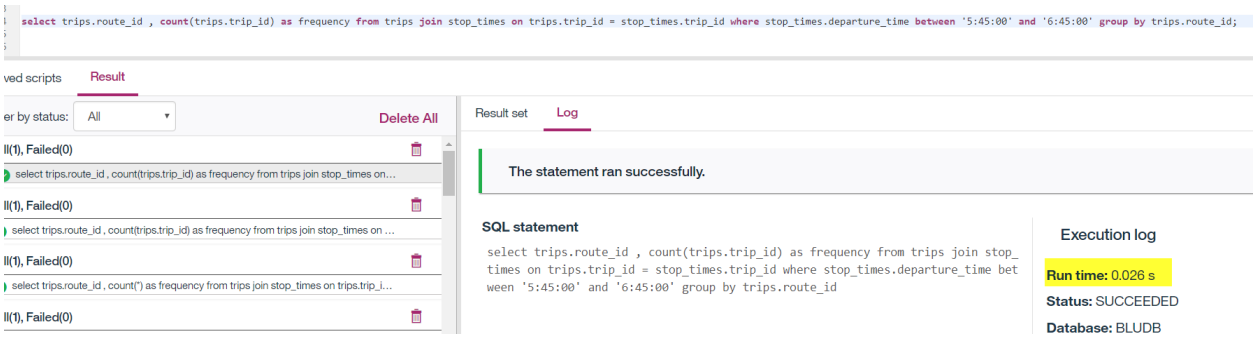
USING LOCAL RDBMS



USING CLOUD RDBMS (OUTPUT SCREEN)

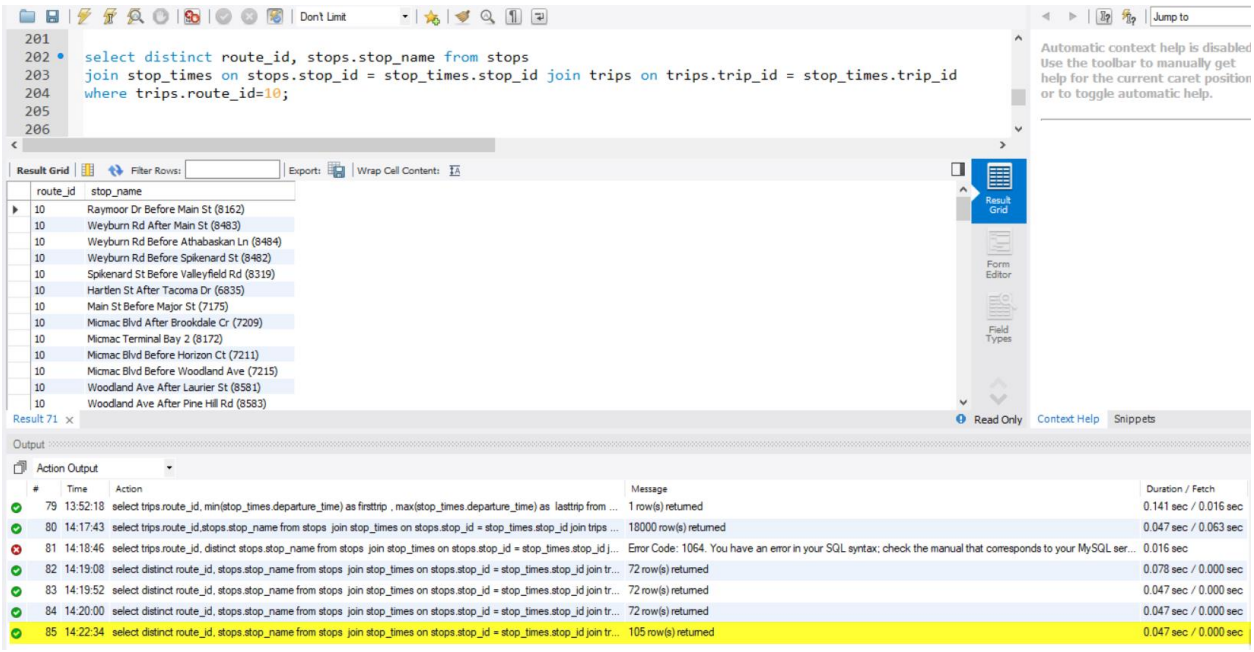


USING CLOUD RDBMS (RESPONSE TIME)

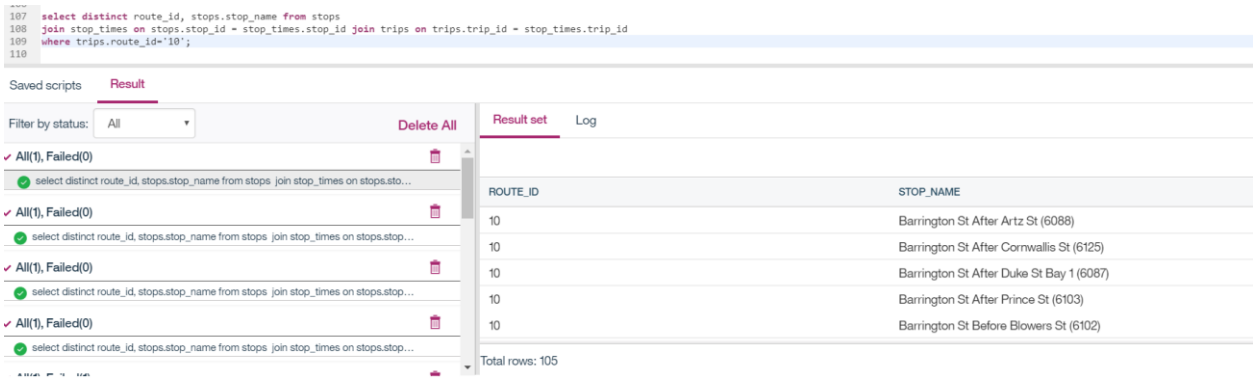


5. We are to find the actual route of a bus service in this question. For this purpose, we get route identifier as input dynamically and display stop names along with their route identifier with the help of joins to combine stops with stop_times table and another join to combine trips and stop_times. Response time for running this query in local RDBMS and cloud RDBMS are 47 millisecs and 54 millisecs.

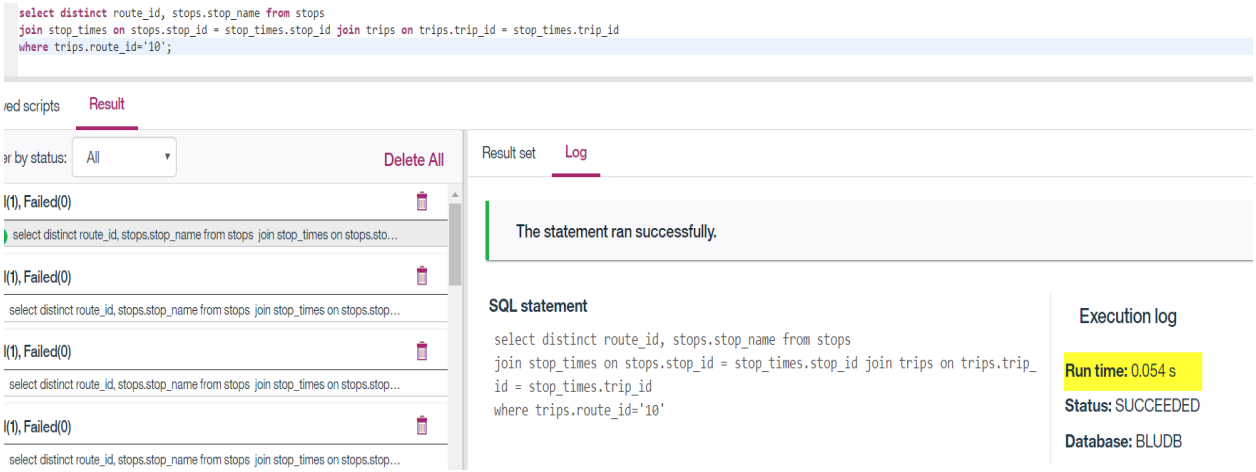
USING LOCAL RDBMS



USING CLOUD RDBMS (OUTPUT SCREEN)

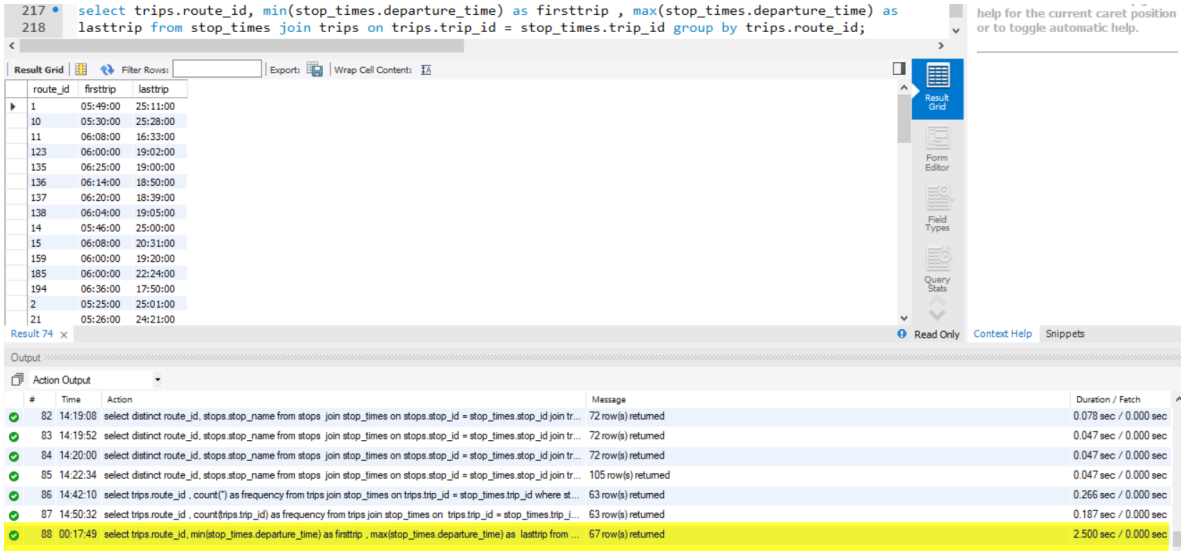


USING CLOUD RDBMS (RESPONSE TIME)

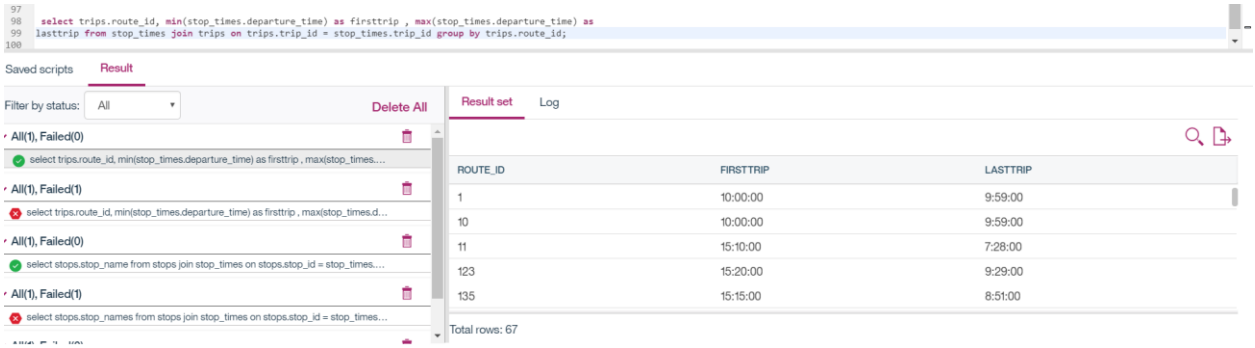


6. In this question, we must get the first and last trip for a particular route. By giving route identifier as input, we run a query to find the minimum and maximum of departure times for that route. We have combined stop_times and trips tables using join. Response time for this query to execute in local RDBMS and cloud RDBMS are 2500 millisecs and 33 millisecs.

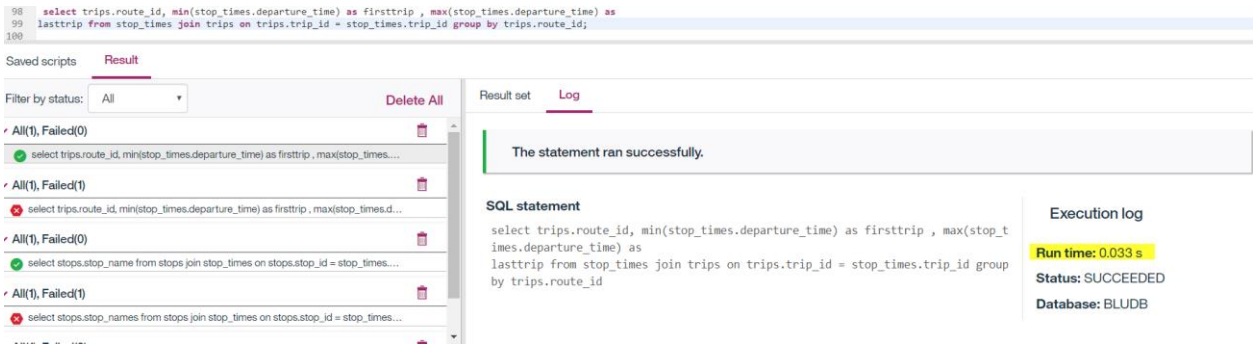
USING LOCAL RDBMS



USING CLOUD RDBMS (OUTPUT SCREEN)



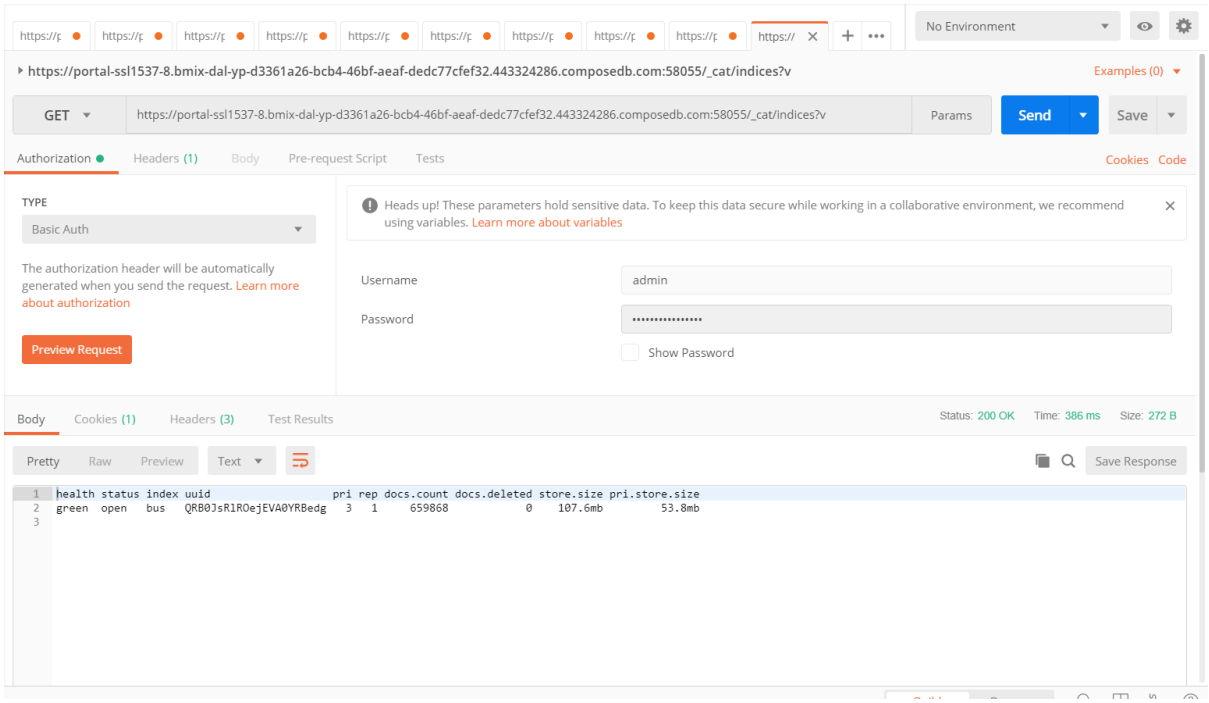
USING CLOUD RDBMS (RESPONSE TIME)



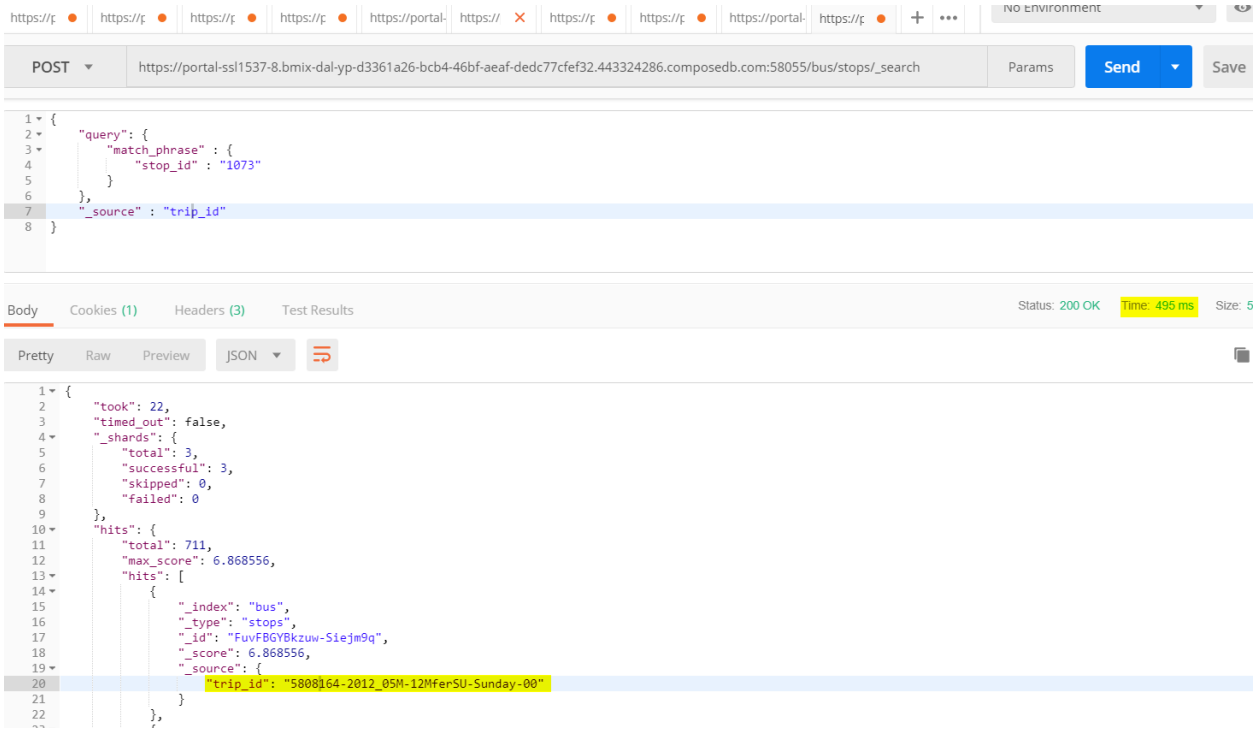
CARRYING OUT QUERIES IN ELASTICSEARCH

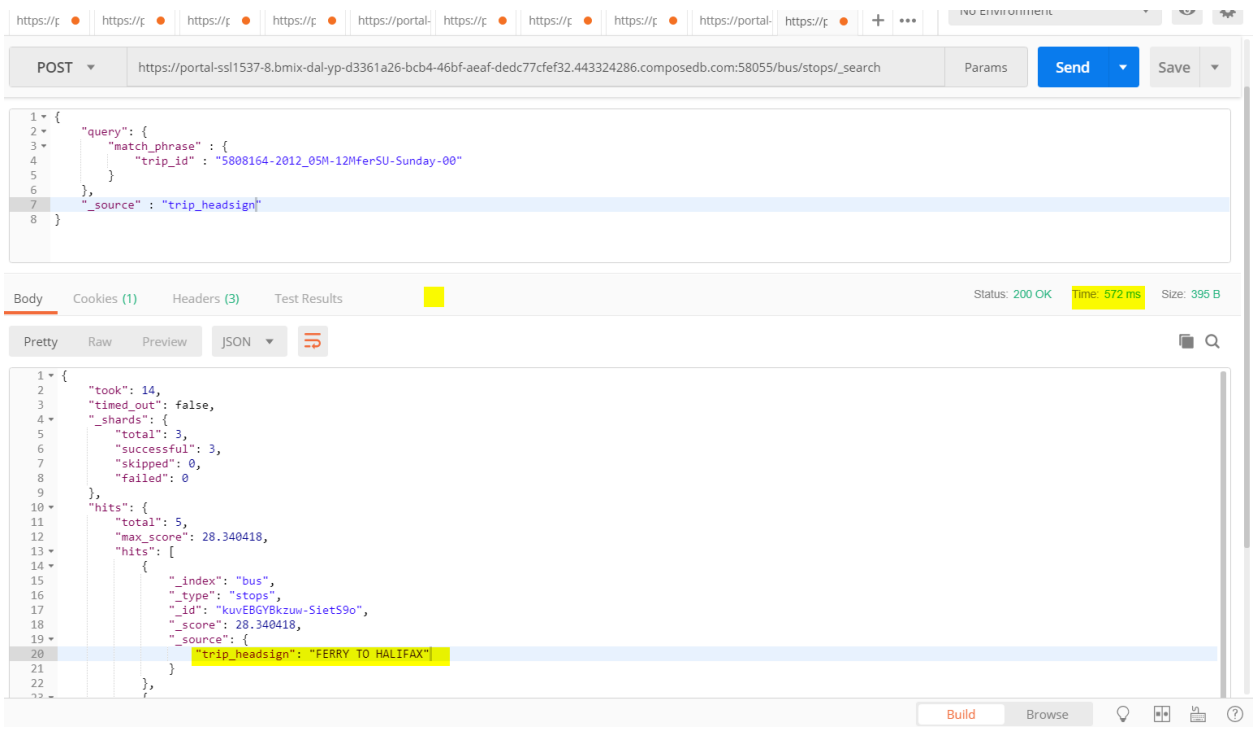
We use Elasticsearch to run the same set of queries in NOSQL. Elastic search is a search engine which makes use of indices to search for a specific pattern in the dataset. Below are screenshots of queries taken with Elasticsearch.

LOADING BULK DATA INTO ELASTICSEARCH

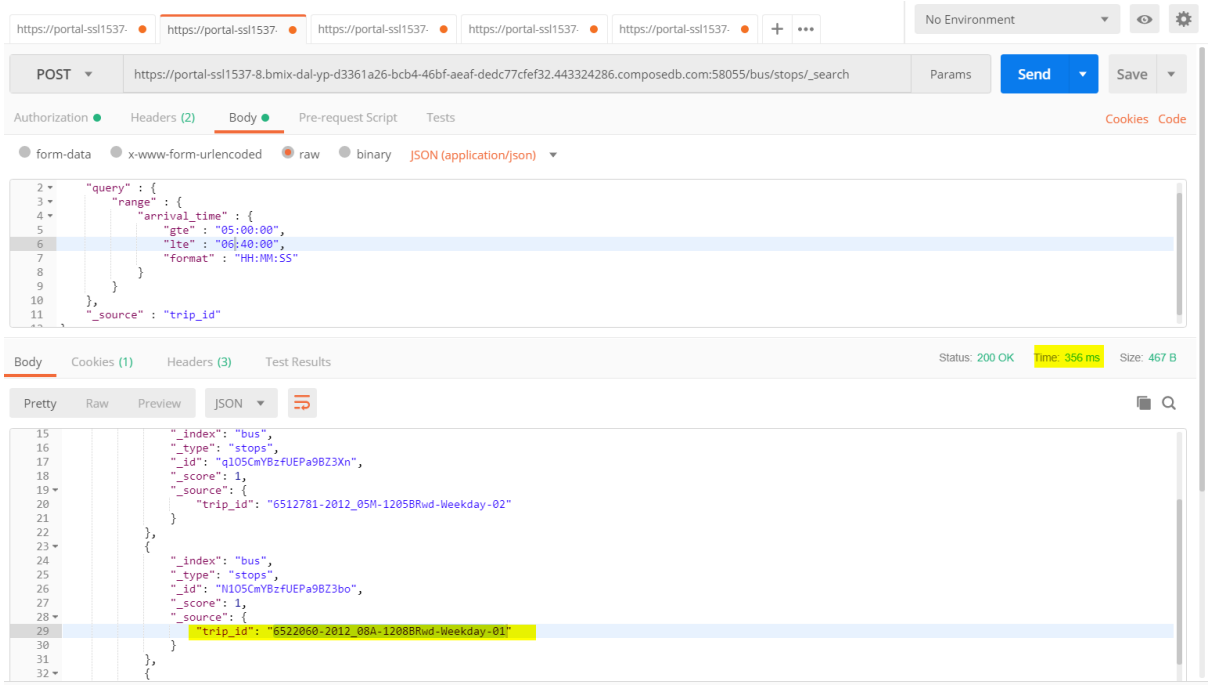


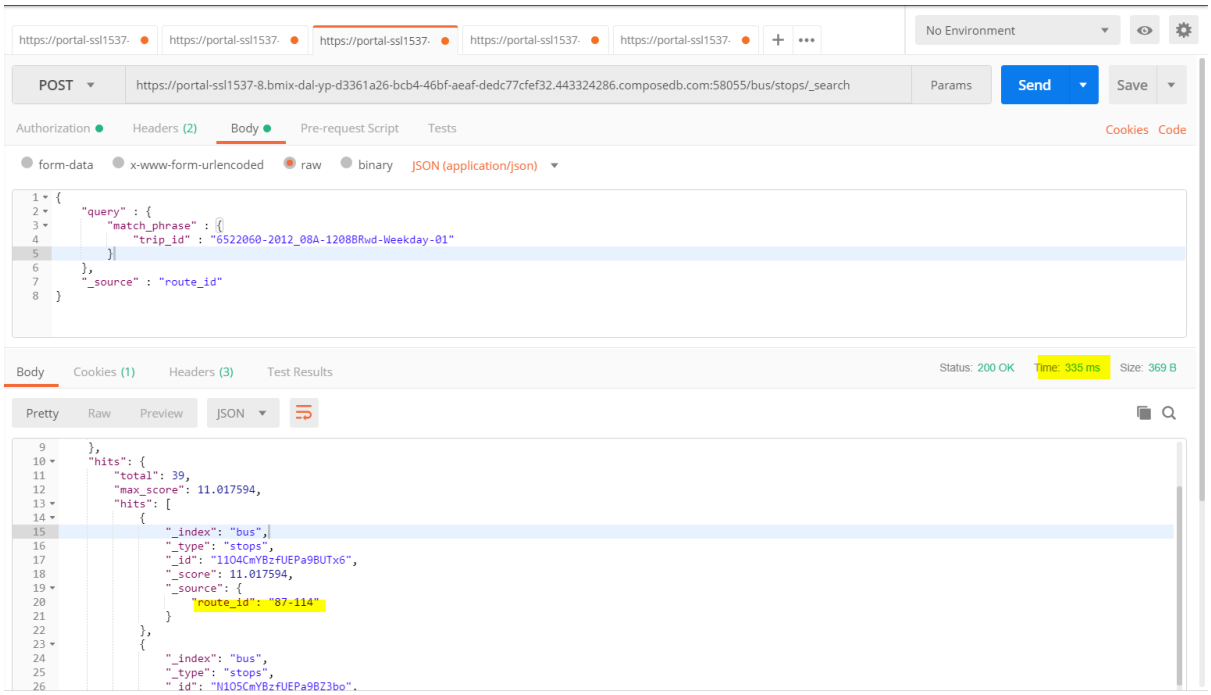
1. We are to identify routes and busses that pass through a specific stop from the Transit database in this question.

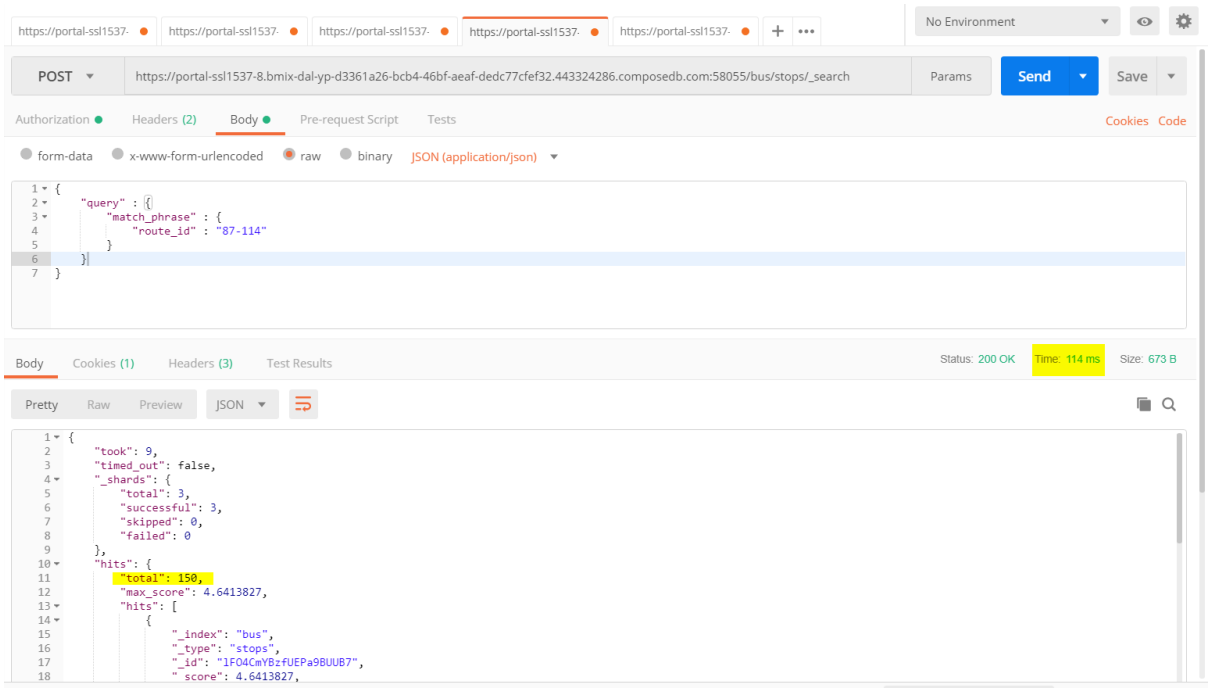




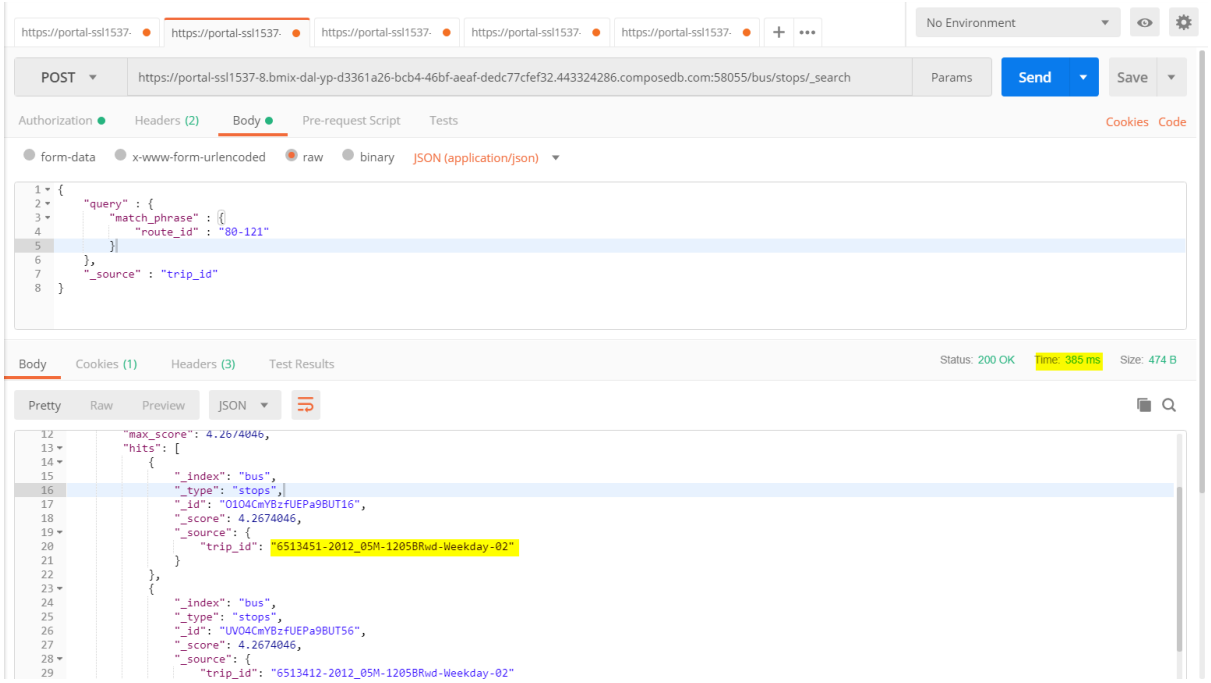
2. We are to find the first and last trip for a specific route here.

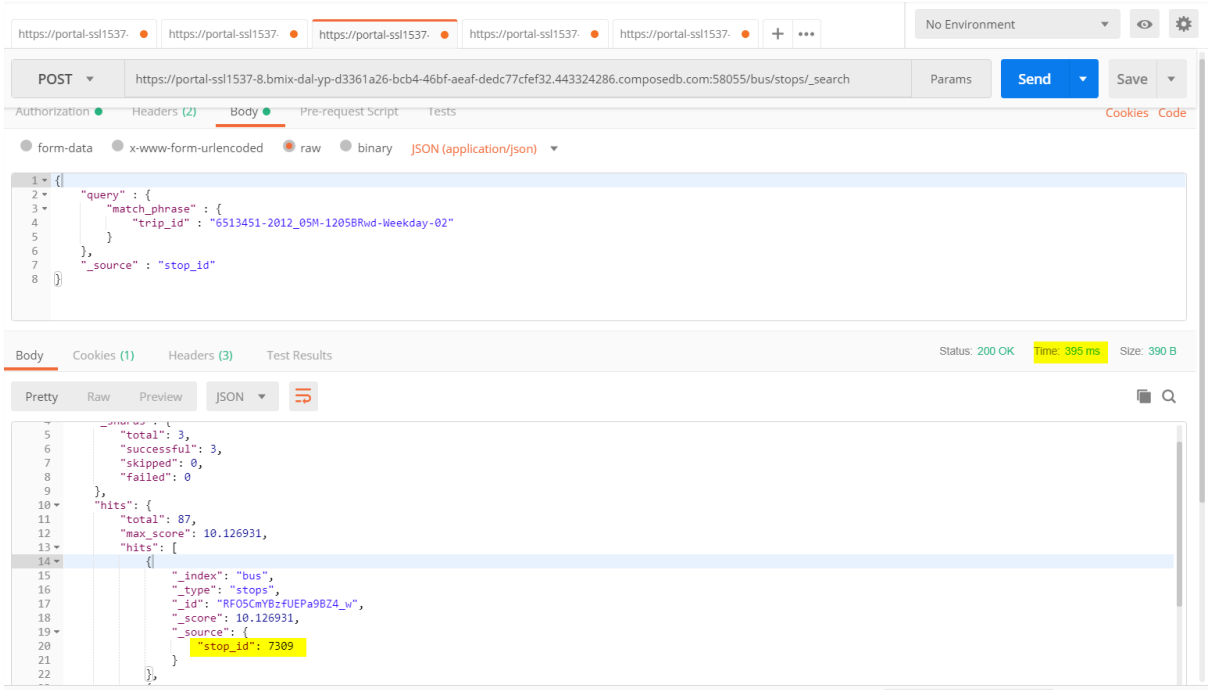


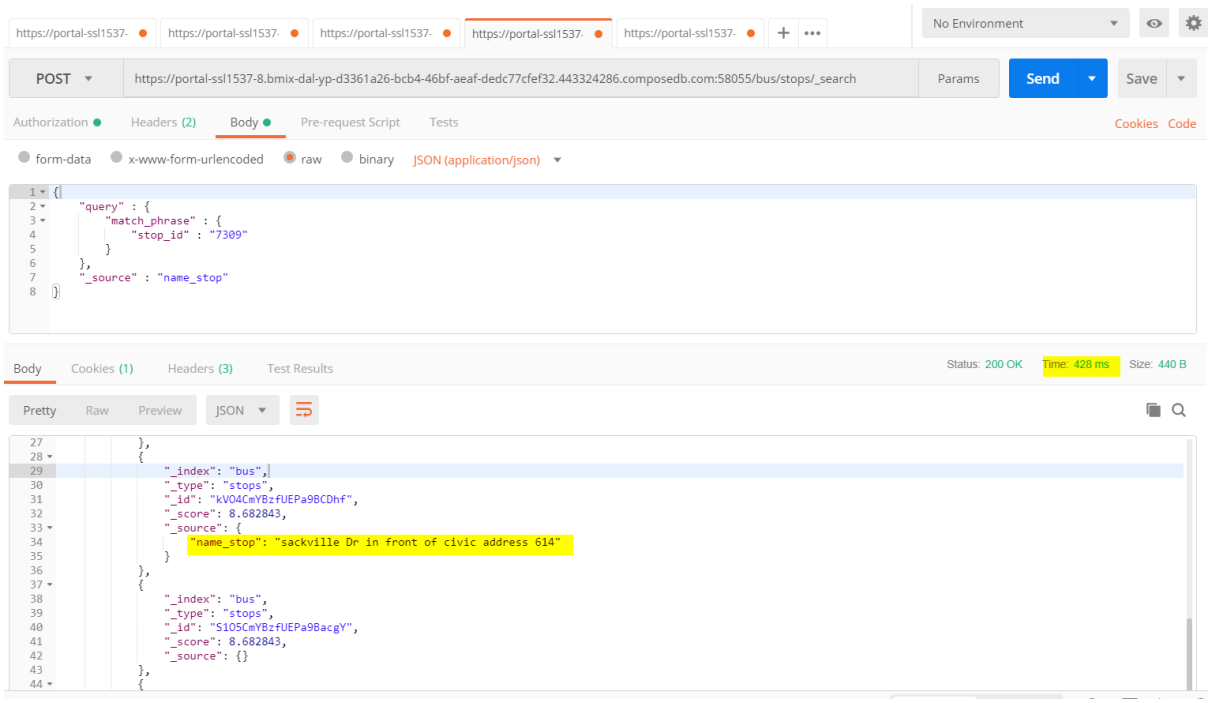




3. We are to find the actual route of a bus service.

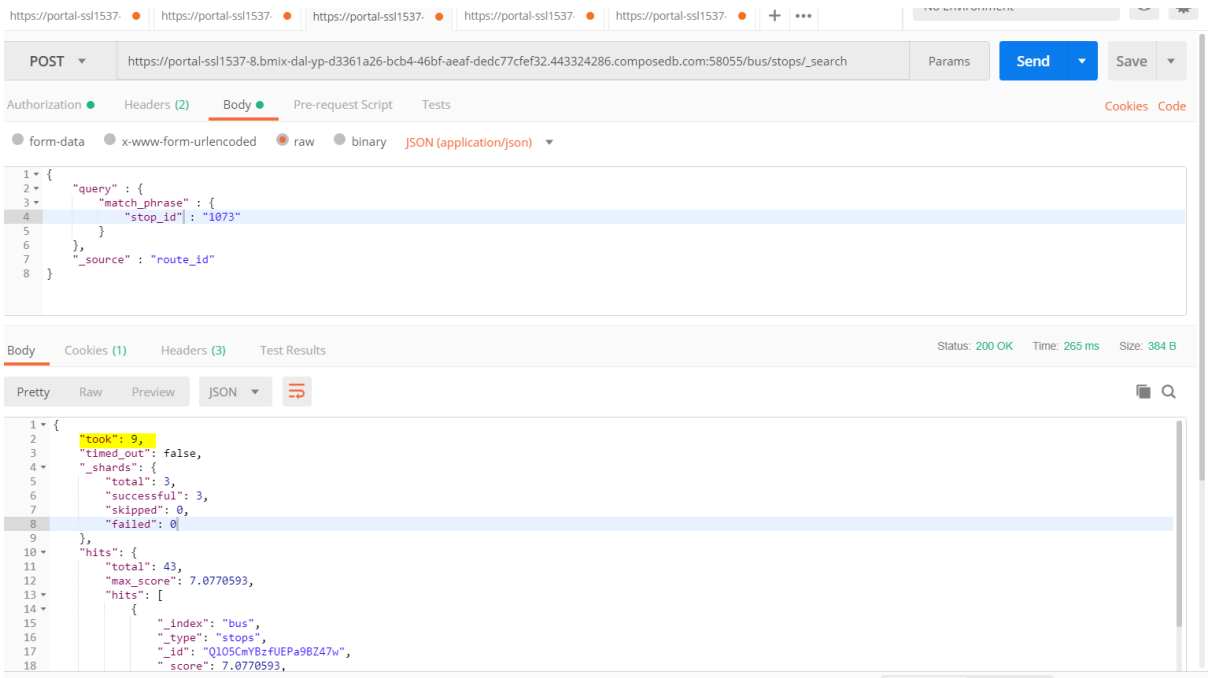


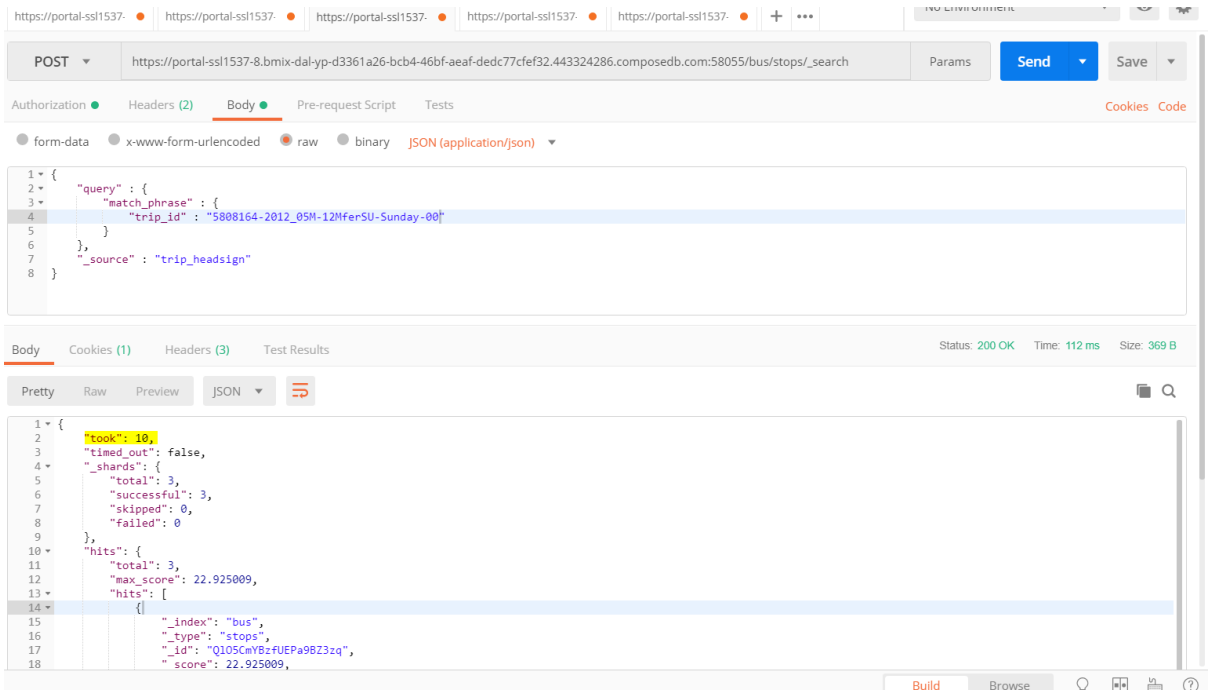




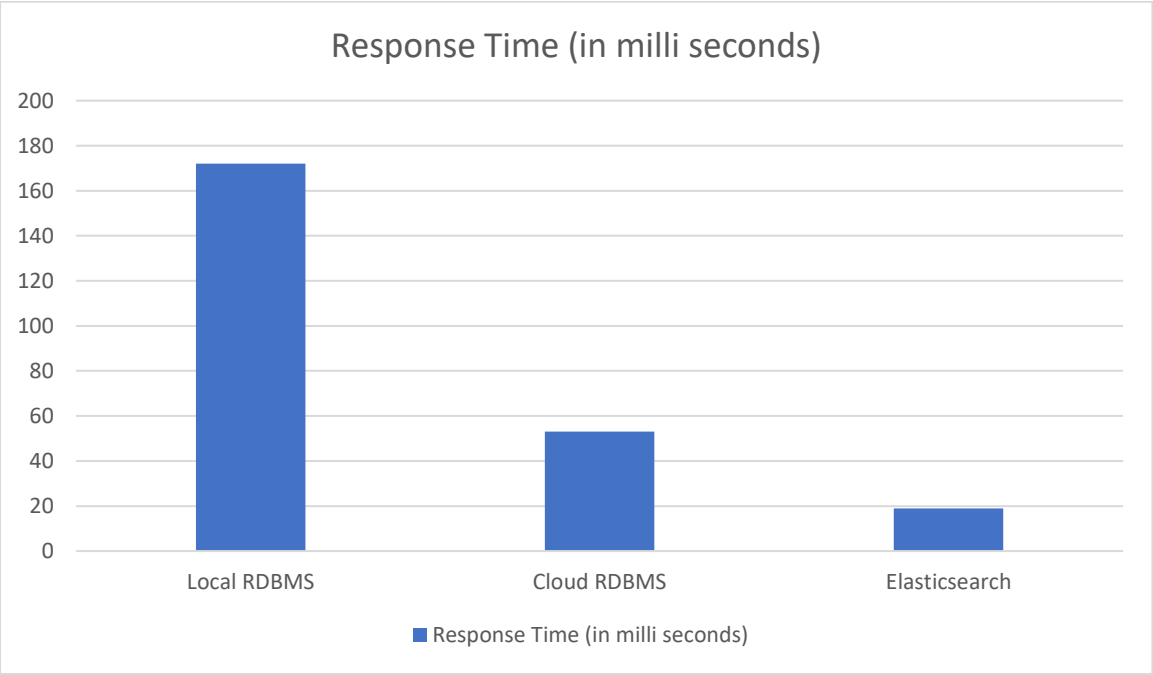
TEST RESULTS

Comparing the response time taken for the first query to run in local MySQL, cloud and Elasticsearch, we notice that Elasticsearch is more efficient of the three methods. This could be understood by the below screenshot where it has taken 9+10 = 19 millisecs for the query.


POST
https://portal-ssl1537-8.bmix-dal-yp-d3361a26-bcb4-46bf-aeaf-dedc77cfef32.443324286.composedb.com:58055/bus/stops/_search
Params
Send
Save
Authorization
Headers (2)
Body
Pre-request Script
Tests
Cookies
Code
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
{
"query" : {
"match_phrase" : {
"stop_id" : "1073"
}
},
"_source" : "route_id"
}
Body
Cookies (1)
Headers (3)
Test Results
Status: 200 OK
Time: 265 ms
Size: 384 B
Pretty
Raw
Preview
JSON
{
"took": 9,
"timed_out": false,
"_shards": {
"total": 3,
"successful": 3,
"skipped": 0,
"failed": 0
},
"hits": {
"total": 43,
"max_score": 7.0770593,
"hits": [
{
"_index": "bus",
"_type": "stops",
"_id": "Q1O5CmYBzfUEPa9BZ47w",
"_score": 7.0770593,


POST
https://portal-ssl1537-8.bmix-dal-yp-d3361a26-bcb4-46bf-aeaf-dedc77cfef32.443324286.composedb.com:58055/bus/stops/_search
Params
Send
Save
Authorization
Headers (2)
Body
Pre-request Script
Tests
Cookies
Code
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
{
"query" : {
"match_phrase" : {
"trip_id" : "5808164-2012_05M-12MferSU-Sunday-00"
}
},
"_source" : "trip_headsign"
}
Body
Cookies (1)
Headers (3)
Test Results
Status: 200 OK
Time: 112 ms
Size: 369 B
Pretty
Raw
Preview
JSON
{
"took": 10,
"timed_out": false,
"_shards": {
"total": 3,
"successful": 3,
"skipped": 0,
"failed": 0
},
"hits": {
"total": 3,
"max_score": 22.925009,
"hits": [
{
"_index": "bus",
"_type": "stops",
"_id": "Q1O5CmYBzfUEPa9BZ3zq",
"_score": 22.925009,
Build
Browse

The response time for the first query in local RDBMS, cloud RDBMS and elastic search is represented as a chart:



DISCUSSION AND SUMMARY

We used MySQL Workbench for querying on our local machine. Since the software is a GUI version, we found it to be more appealing and easier to use unlike the command line MySQL server. Loading bulk data into the local database using MySQL workbench also took only a decent amount of time. Since the server runs locally we had no issues with its availability as well. On the contrary, we faced a lot of availability issues with IBM DB2 cloud service as services were down most of the time due to a massive number of users trying to reach the cloud. Apart from this only issue we found the cloud RDBMS to be more efficient than local RDBMS in terms of response time taken for each query. Also, loading huge data sets was comparatively fast.

Elasticsearch was very efficient for queries that search for a matching data/pattern from the database. But loading bulk data into Elasticsearch using Postman REST API was quite tedious as most of the time there were gateway failures. To sum up, elastic search has outplayed both the local and cloud RDBMS in terms of efficiency.