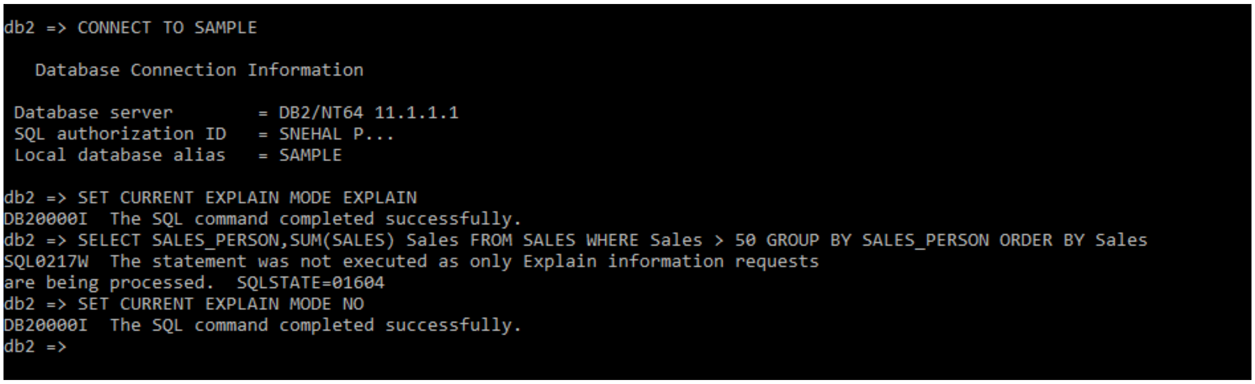
PROJECT TEAM 11
DATABASE TEAM BASED ASSIGNMENT

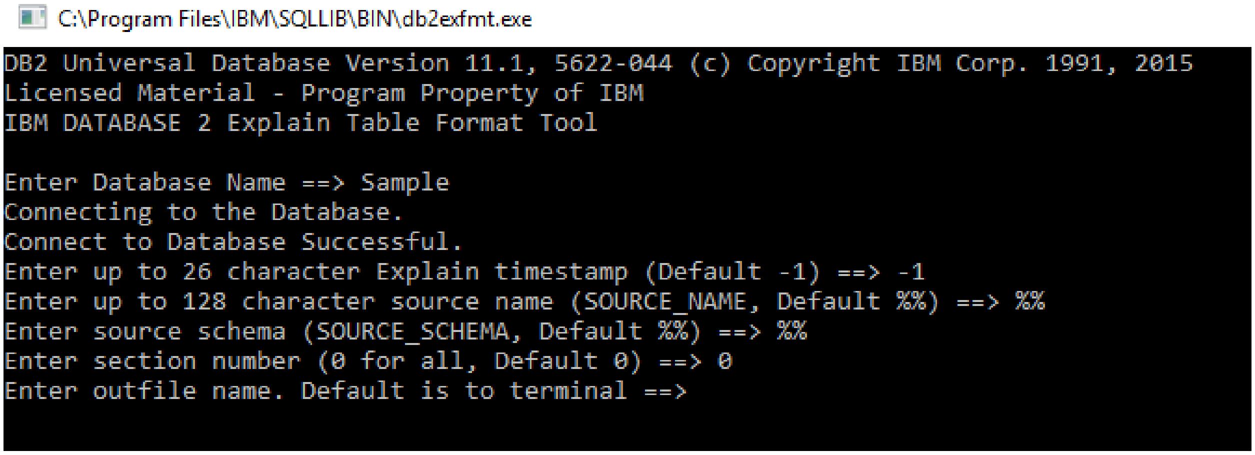
DB2 EXPRESS C

1.1 Running Sample query:



```
db2 => CONNECT TO SAMPLE

   Database Connection Information

 Database server        = DB2/NT64 11.1.1.1
 SQL authorization ID   = SNEHAL P...
 Local database alias   = SAMPLE

db2 => SET CURRENT EXPLAIN MODE EXPLAIN
DB20000I  The SQL command completed successfully.
db2 => SELECT SALES_PERSON,SUM(SALES) Sales FROM SALES WHERE Sales > 50 GROUP BY SALES_PERSON ORDER BY Sales
SQL0217W  The statement was not executed as only Explain information requests
are being processed.  SQLSTATE=01604
db2 => SET CURRENT EXPLAIN MODE NO
DB20000I  The SQL command completed successfully.
db2 =>
```

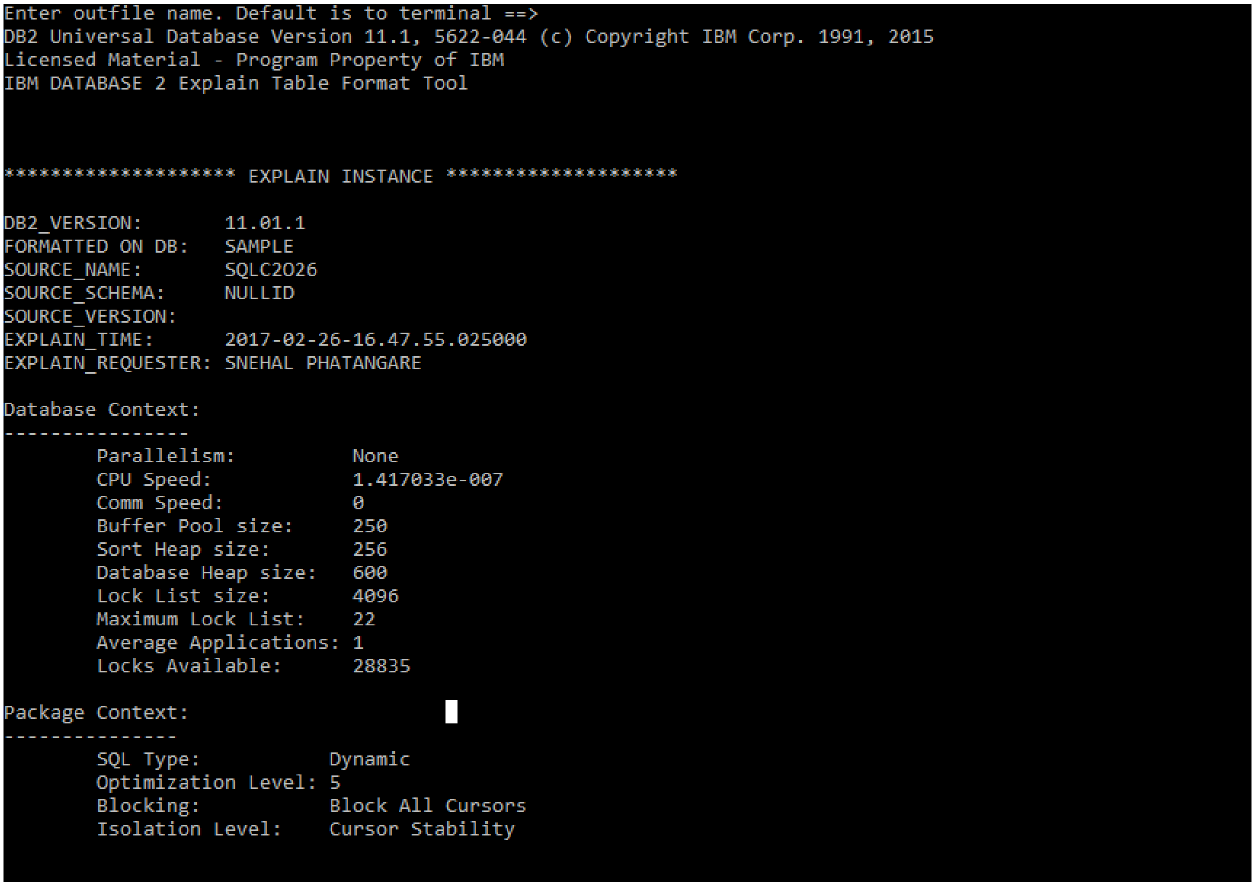
1.2 db2exfmt Command execution:



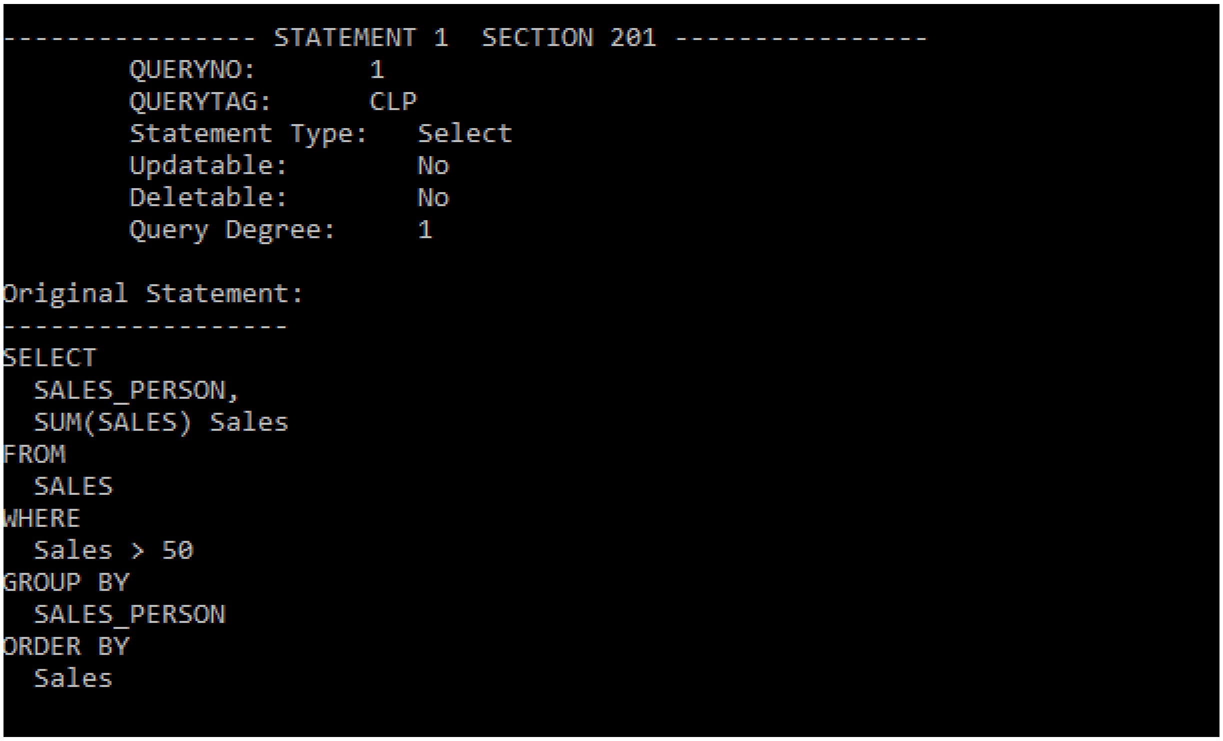
C:\Program Files\IBM\SQLLIB\BIN\db2exfmt.exe

```
DB2 Universal Database Version 11.1, 5622-044 (c) Copyright IBM Corp. 1991, 2015
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool

Enter Database Name ==> Sample
Connecting to the Database.
Connect to Database Successful.
Enter up to 26 character Explain timestamp (Default -1) ==> -1
Enter up to 128 character source name (SOURCE_NAME, Default %%) ==> %%
Enter source schema (SOURCE_SCHEMA, Default %%) ==> %%
Enter section number (0 for all, Default 0) ==> 0
Enter outfile name. Default is to terminal ==>
```

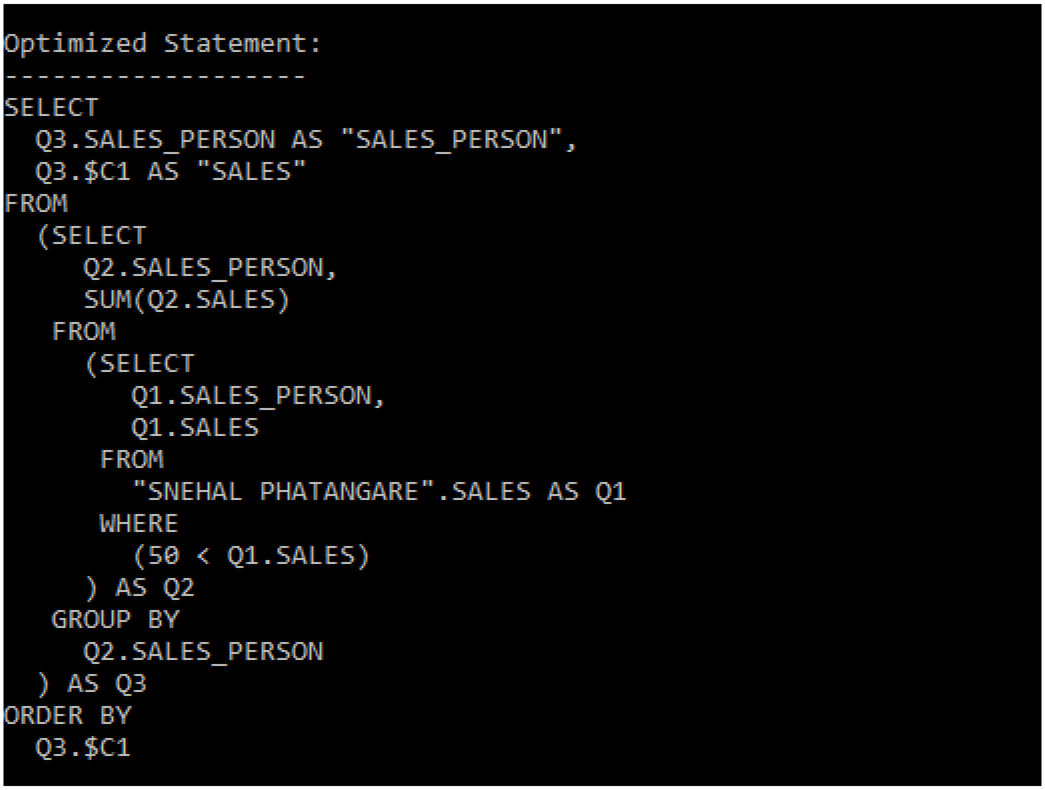
1.3 db2exfmt Command output:



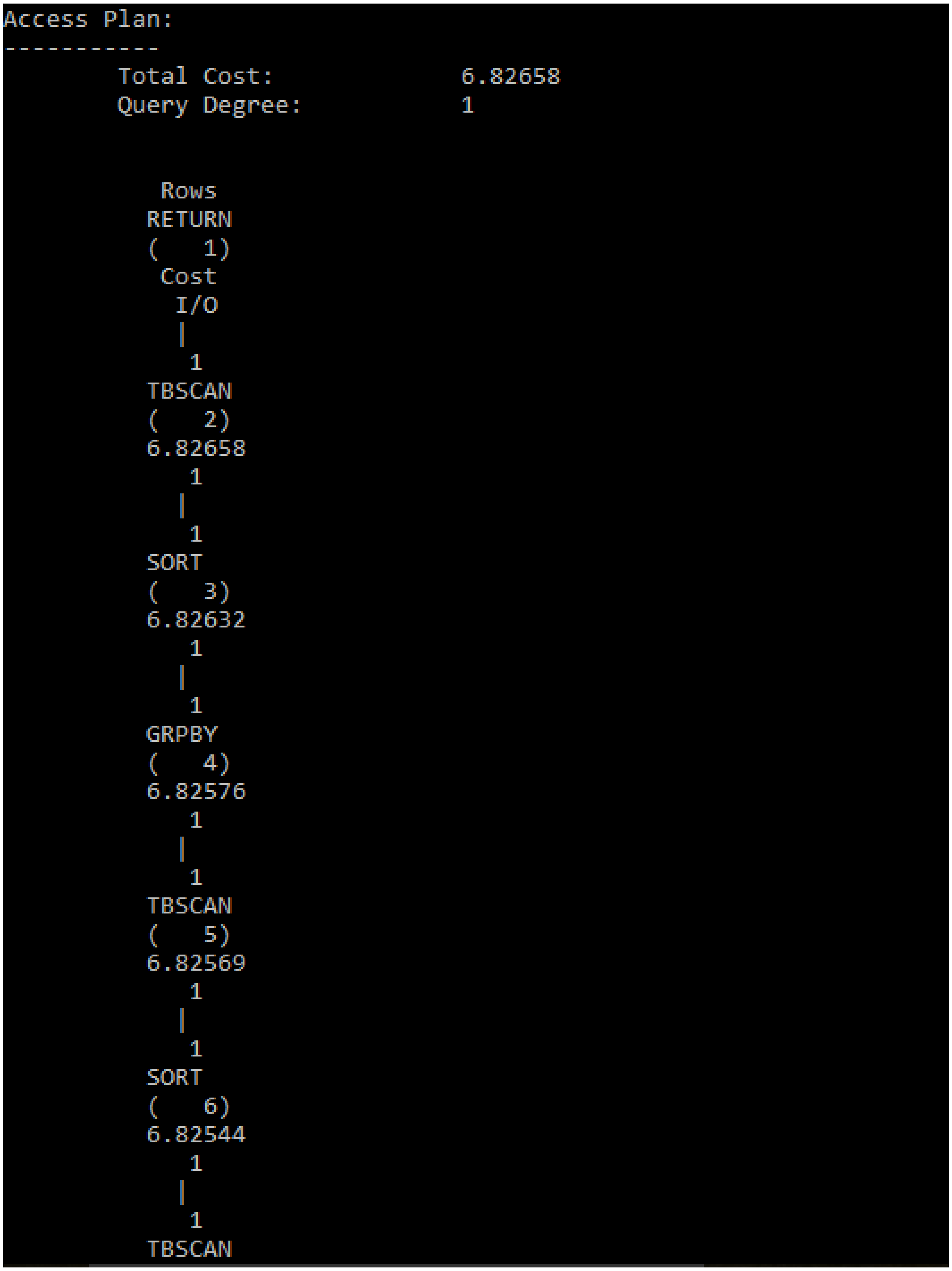
```
Enter outfile name. Default is to terminal ==>
DB2 Universal Database Version 11.1, 5622-044 (c) Copyright IBM Corp. 1991, 2015
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool



******************** EXPLAIN INSTANCE ********************

DB2_VERSION:       11.01.1
FORMATTED ON DB:   SAMPLE
SOURCE_NAME:       SQLC2O26
SOURCE_SCHEMA:     NULLID
SOURCE_VERSION:
EXPLAIN_TIME:      2017-02-26-16.47.55.025000
EXPLAIN_REQUESTER: SNEHAL PHATANGARE

Database Context:
----------------
        Parallelism:          None
        CPU Speed:            1.417033e-007
        Comm Speed:           0
        Buffer Pool size:     250
        Sort Heap size:       256
        Database Heap size:   600
        Lock List size:       4096
        Maximum Lock List:    22
        Average Applications: 1
        Locks Available:      28835

Package Context:
---------------
        SQL Type:           Dynamic
        Optimization Level: 5
        Blocking:           Block All Cursors
        Isolation Level:    Cursor Stability
```



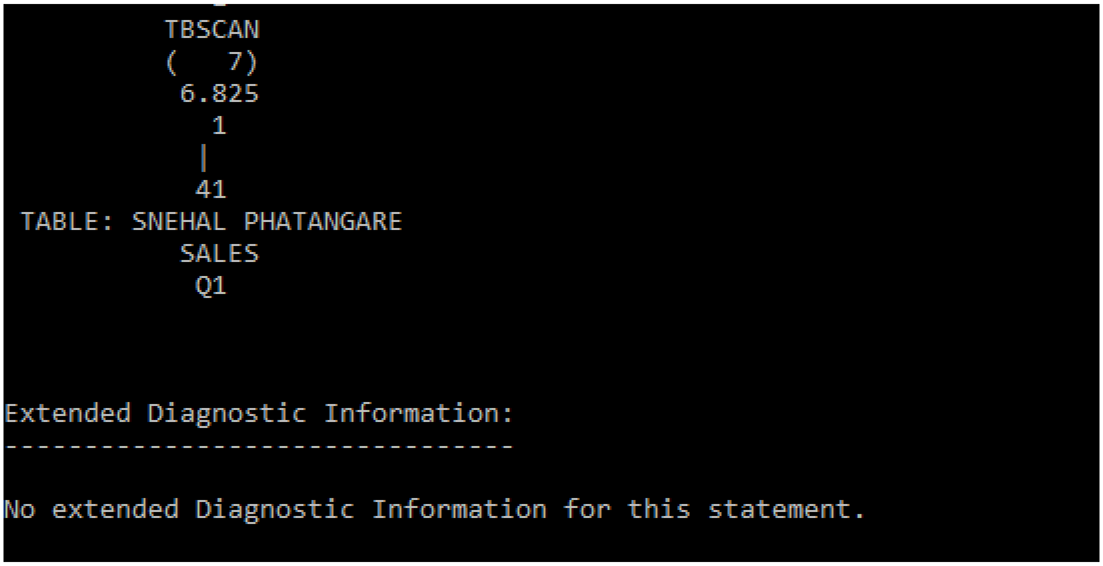
```
---------------- STATEMENT 1  SECTION 201 ----------------
        QUERYNO:        1
        QUERYTAG:       CLP
        Statement Type:   Select
        Updatable:        No
        Deletable:        No
        Query Degree:     1

Original Statement:
------------------
SELECT
  SALES_PERSON,
  SUM(SALES) Sales
FROM
  SALES
WHERE
  Sales > 50
GROUP BY
  SALES_PERSON
ORDER BY
  Sales
```



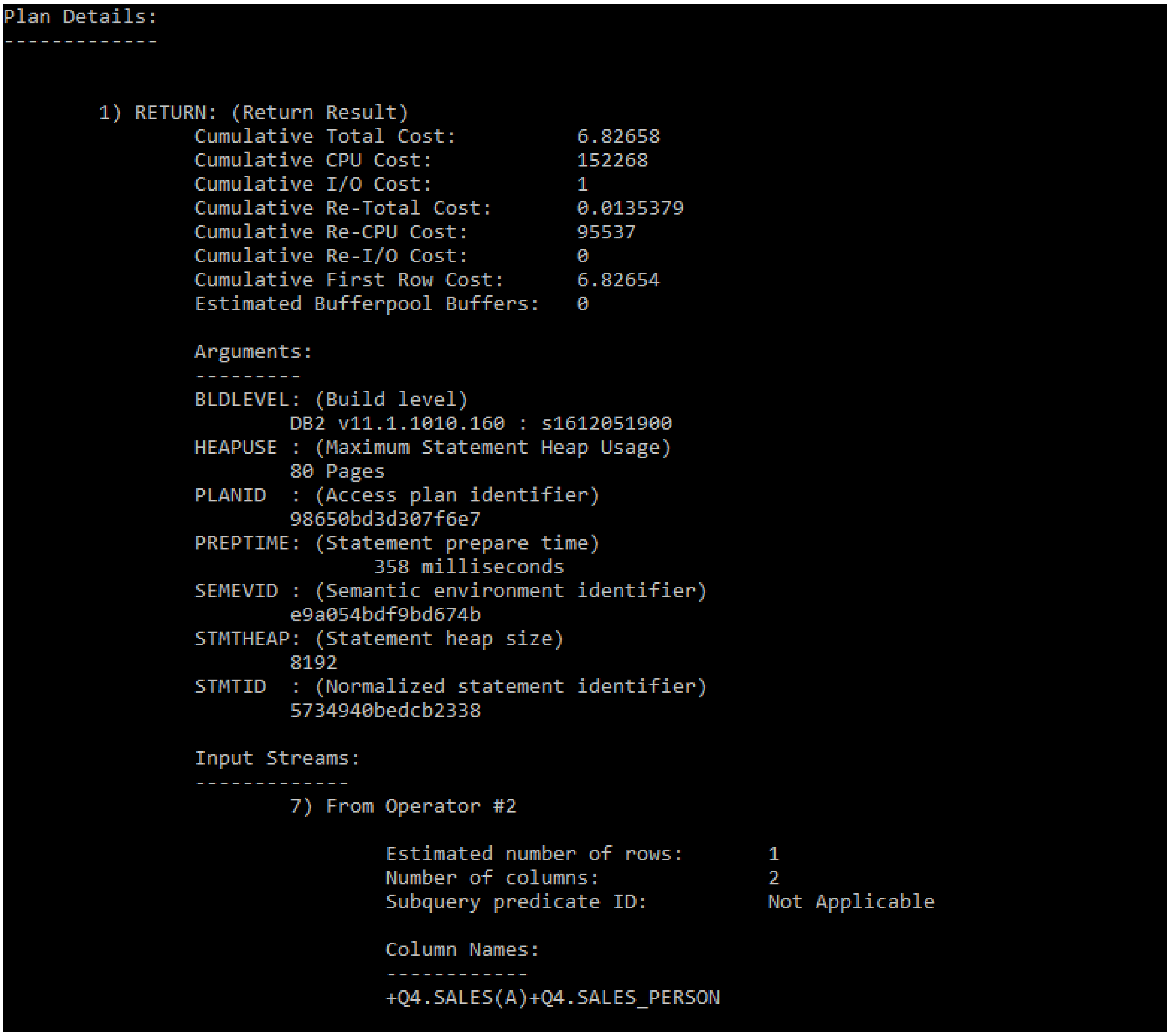
```
Optimized Statement:
-------------------
SELECT
  Q3.SALES_PERSON AS "SALES_PERSON",
  Q3.$C1 AS "SALES"
FROM
  (SELECT
     Q2.SALES_PERSON,
     SUM(Q2.SALES)
   FROM
     (SELECT
        Q1.SALES_PERSON,
        Q1.SALES
      FROM
        "SNEHAL PHATANGARE".SALES AS Q1
      WHERE
        (50 < Q1.SALES)
     ) AS Q2
   GROUP BY
     Q2.SALES_PERSON
  ) AS Q3
ORDER BY
  Q3.$C1
```



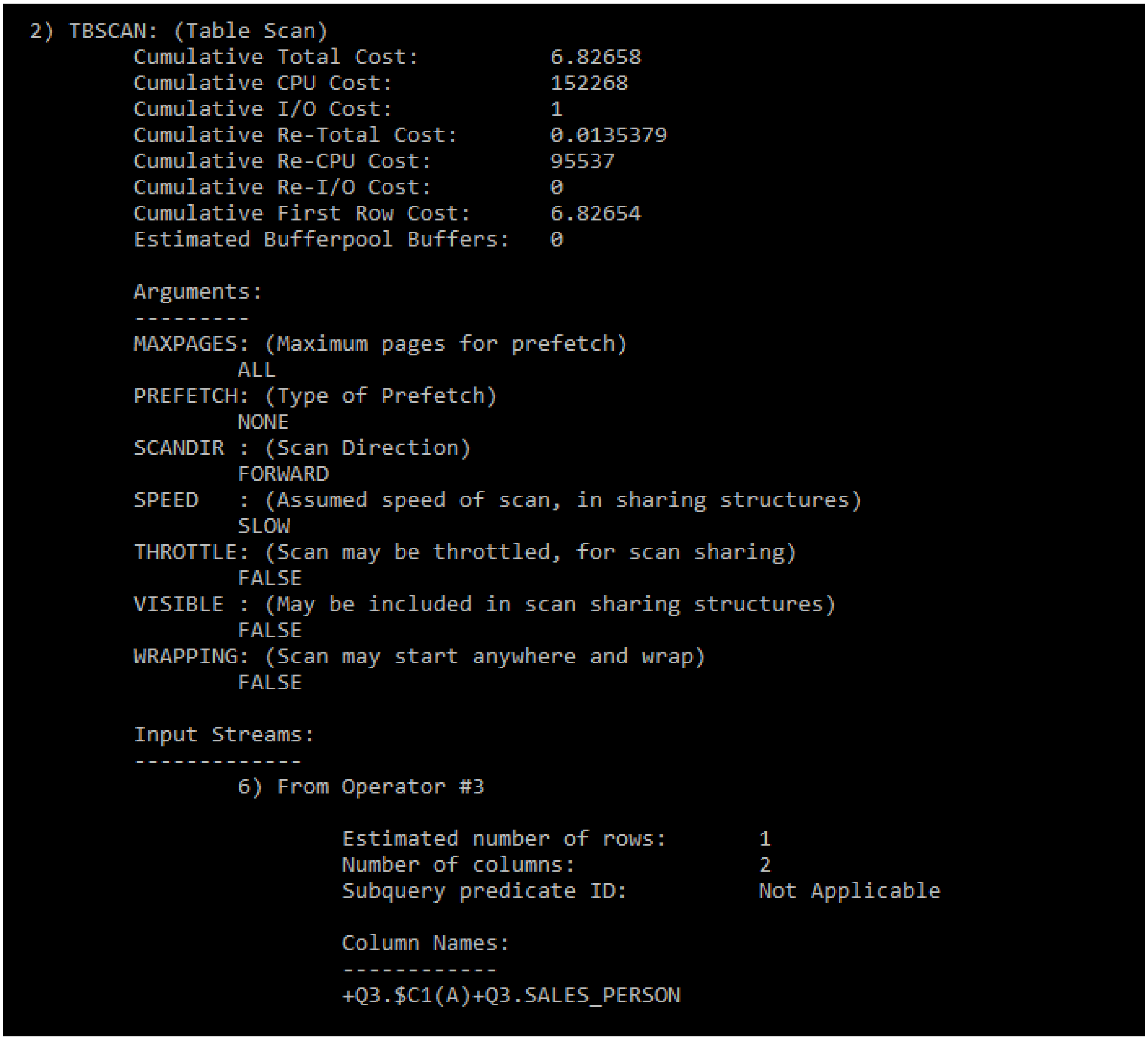
```
Access Plan:
-----------
        Total Cost:             6.82658
        Query Degree:           1


           Rows
          RETURN
          (   1)
           Cost
            I/O
            |
             1
          TBSCAN
          (   2)
          6.82658
             1
            |
             1
          SORT
          (   3)
          6.82632
             1
            |
             1
          GRPBY
          (   4)
          6.82576
             1
            |
             1
          TBSCAN
          (   5)
          6.82569
             1
            |
             1
          SORT
          (   6)
          6.82544
             1
            |
             1
          TBSCAN
```



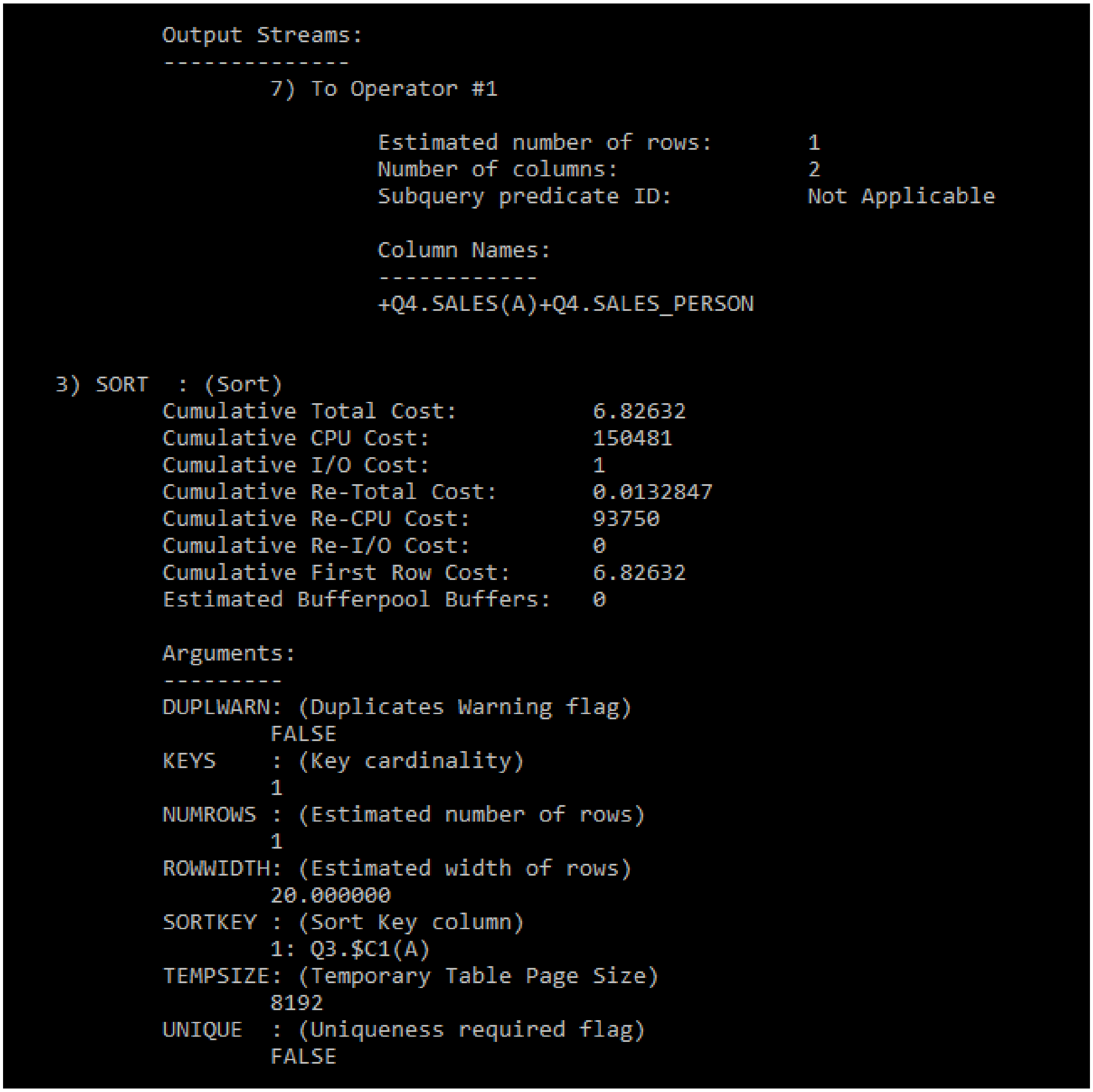
```
           TBSCAN
           (   7)
            6.825
              1
             |
             41
 TABLE: SNEHAL PHATANGARE
            SALES
             Q1


Extended Diagnostic Information:
--------------------------------

No extended Diagnostic Information for this statement.
```



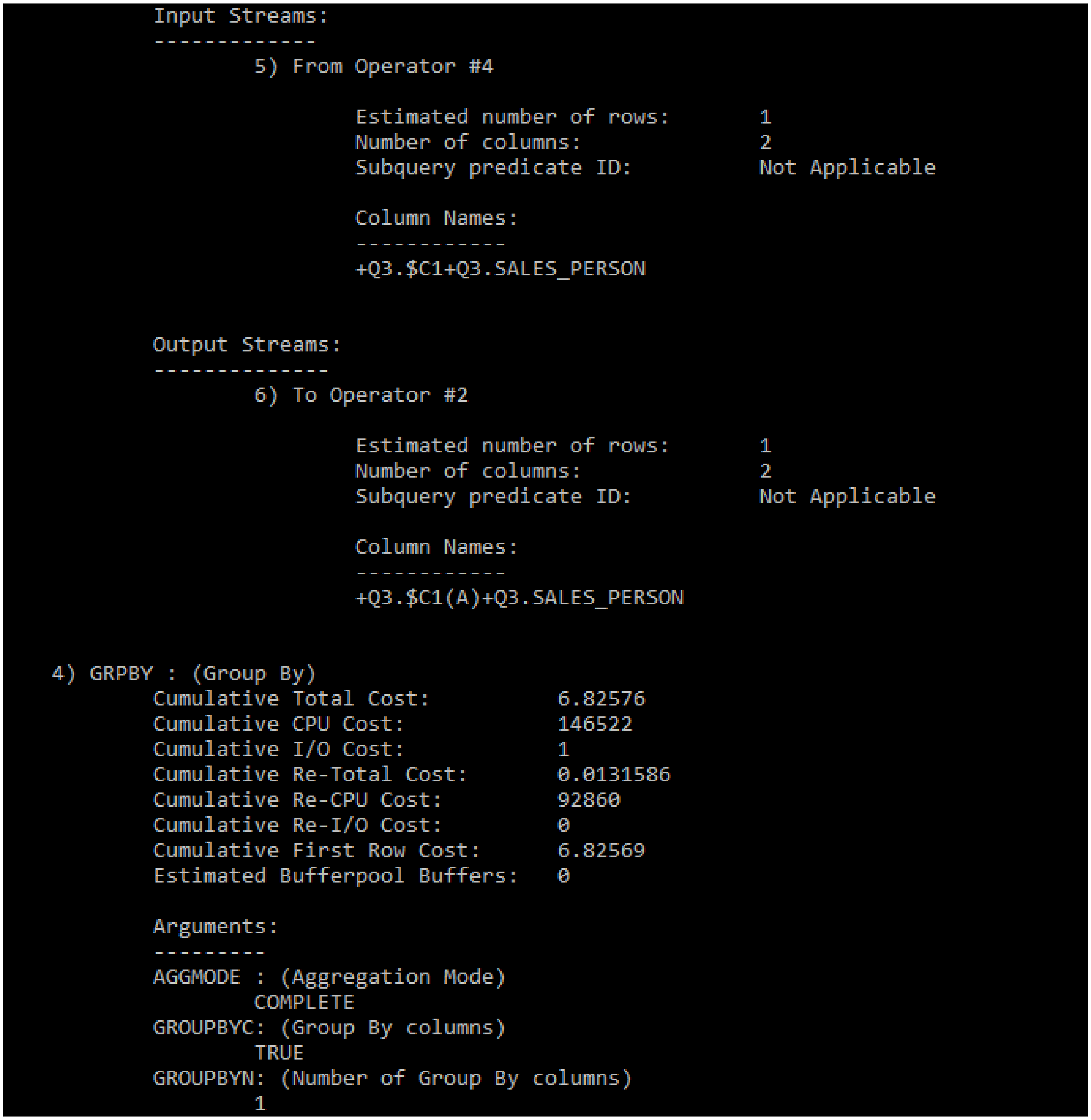
```
Plan Details:
-------------


        1) RETURN: (Return Result)
                Cumulative Total Cost:          6.82658
                Cumulative CPU Cost:            152268
                Cumulative I/O Cost:            1
                Cumulative Re-Total Cost:       0.0135379
                Cumulative Re-CPU Cost:         95537
                Cumulative Re-I/O Cost:         0
                Cumulative First Row Cost:      6.82654
                Estimated Bufferpool Buffers:   0

                Arguments:
                ---------
                BLDLEVEL: (Build level)
                        DB2 v11.1.1010.160 : s1612051900
                HEAPUSE : (Maximum Statement Heap Usage)
                        80 Pages
                PLANID  : (Access plan identifier)
                        98650bd3d307f6e7
                PREPTIME: (Statement prepare time)
                               358 milliseconds
                SEMEVID : (Semantic environment identifier)
                        e9a054bdf9bd674b
                STMTHEAP: (Statement heap size)
                        8192
                STMTID  : (Normalized statement identifier)
                        5734940bedcb2338

                Input Streams:
                -------------
                        7) From Operator #2

                                Estimated number of rows:       1
                                Number of columns:              2
                                Subquery predicate ID:          Not Applicable

                                Column Names:
                                ------------
                                +Q4.SALES(A)+Q4.SALES_PERSON
```



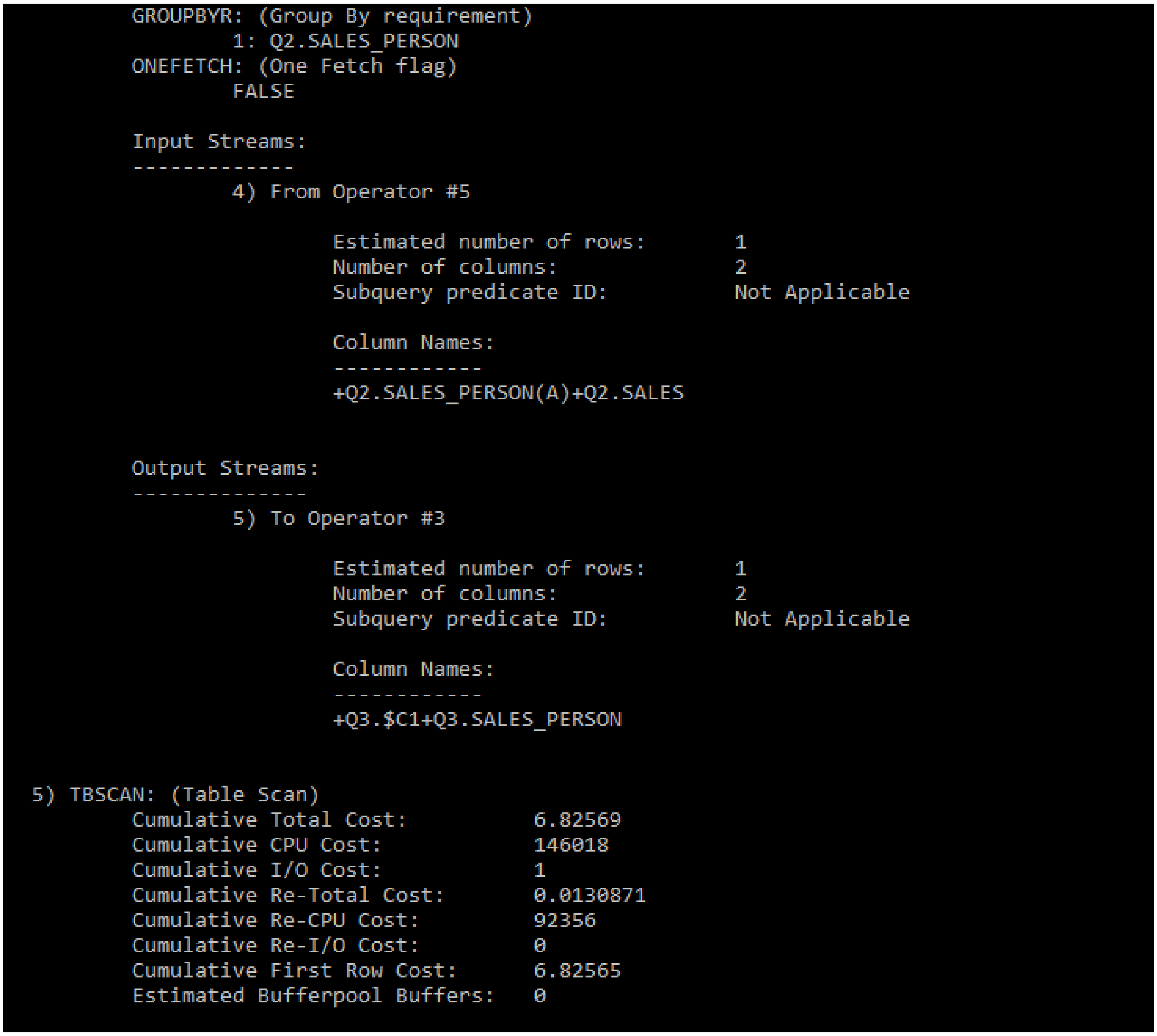
```
2) TBSCAN: (Table Scan)
        Cumulative Total Cost:          6.82658
        Cumulative CPU Cost:            152268
        Cumulative I/O Cost:            1
        Cumulative Re-Total Cost:       0.0135379
        Cumulative Re-CPU Cost:         95537
        Cumulative Re-I/O Cost:         0
        Cumulative First Row Cost:      6.82654
        Estimated Bufferpool Buffers:   0

        Arguments:
        ---------
        MAXPAGES: (Maximum pages for prefetch)
                ALL
        PREFETCH: (Type of Prefetch)
                NONE
        SCANDIR : (Scan Direction)
                FORWARD
        SPEED   : (Assumed speed of scan, in sharing structures)
                SLOW
        THROTTLE: (Scan may be throttled, for scan sharing)
                FALSE
        VISIBLE : (May be included in scan sharing structures)
                FALSE
        WRAPPING: (Scan may start anywhere and wrap)
                FALSE

        Input Streams:
        -------------
                6) From Operator #3

                        Estimated number of rows:       1
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q3.$C1(A)+Q3.SALES_PERSON
```



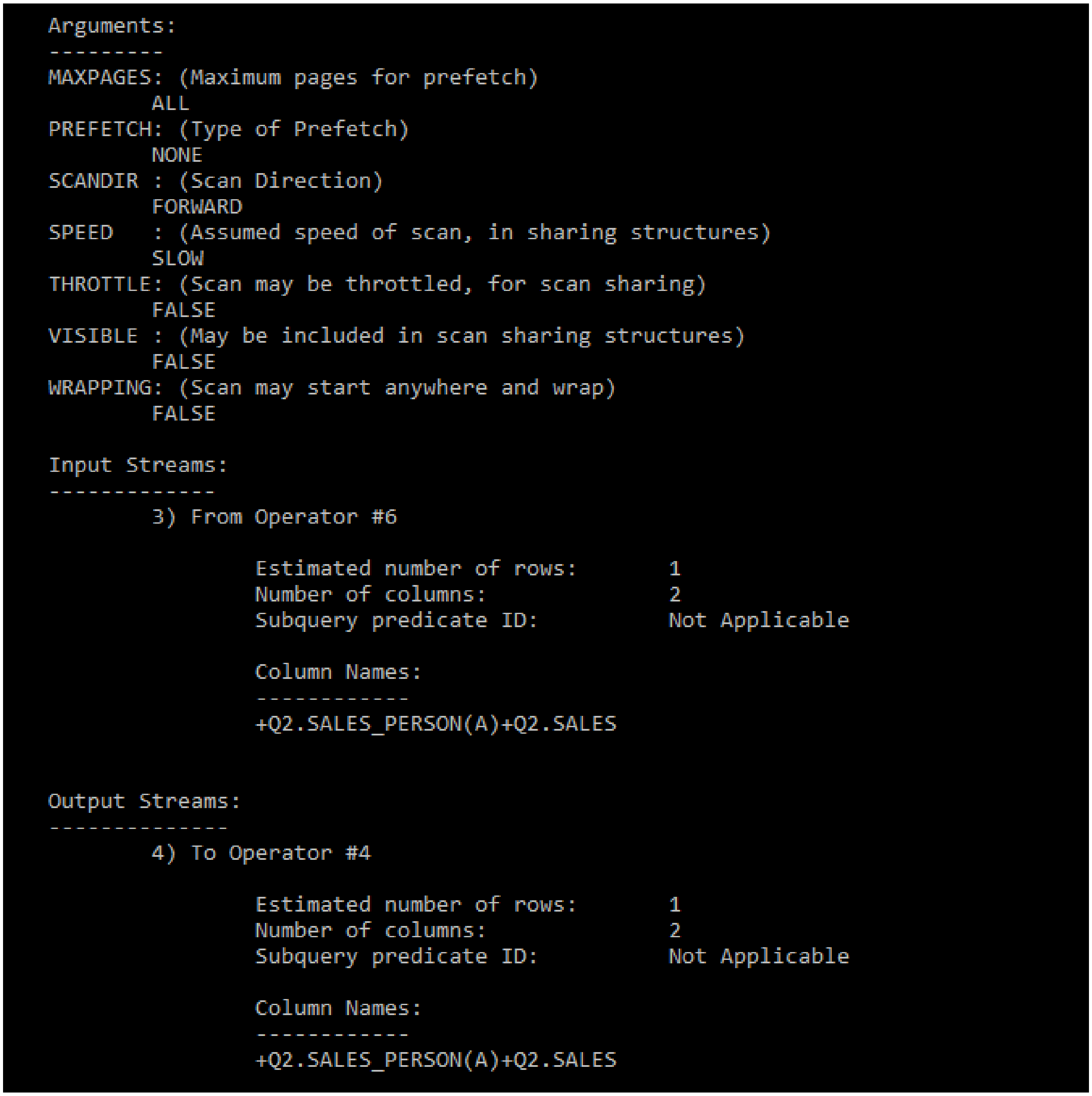
```
        Output Streams:
        --------------
                7) To Operator #1

                        Estimated number of rows:       1
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q4.SALES(A)+Q4.SALES_PERSON


3) SORT  : (Sort)
        Cumulative Total Cost:          6.82632
        Cumulative CPU Cost:            150481
        Cumulative I/O Cost:            1
        Cumulative Re-Total Cost:       0.0132847
        Cumulative Re-CPU Cost:         93750
        Cumulative Re-I/O Cost:         0
        Cumulative First Row Cost:      6.82632
        Estimated Bufferpool Buffers:   0

        Arguments:
        ---------
        DUPLWARN: (Duplicates Warning flag)
                FALSE
        KEYS    : (Key cardinality)
                1
        NUMROWS : (Estimated number of rows)
                1
        ROWWIDTH: (Estimated width of rows)
                20.000000
        SORTKEY : (Sort Key column)
                1: Q3.$C1(A)
        TEMPSIZE: (Temporary Table Page Size)
                8192
        UNIQUE  : (Uniqueness required flag)
                FALSE
```



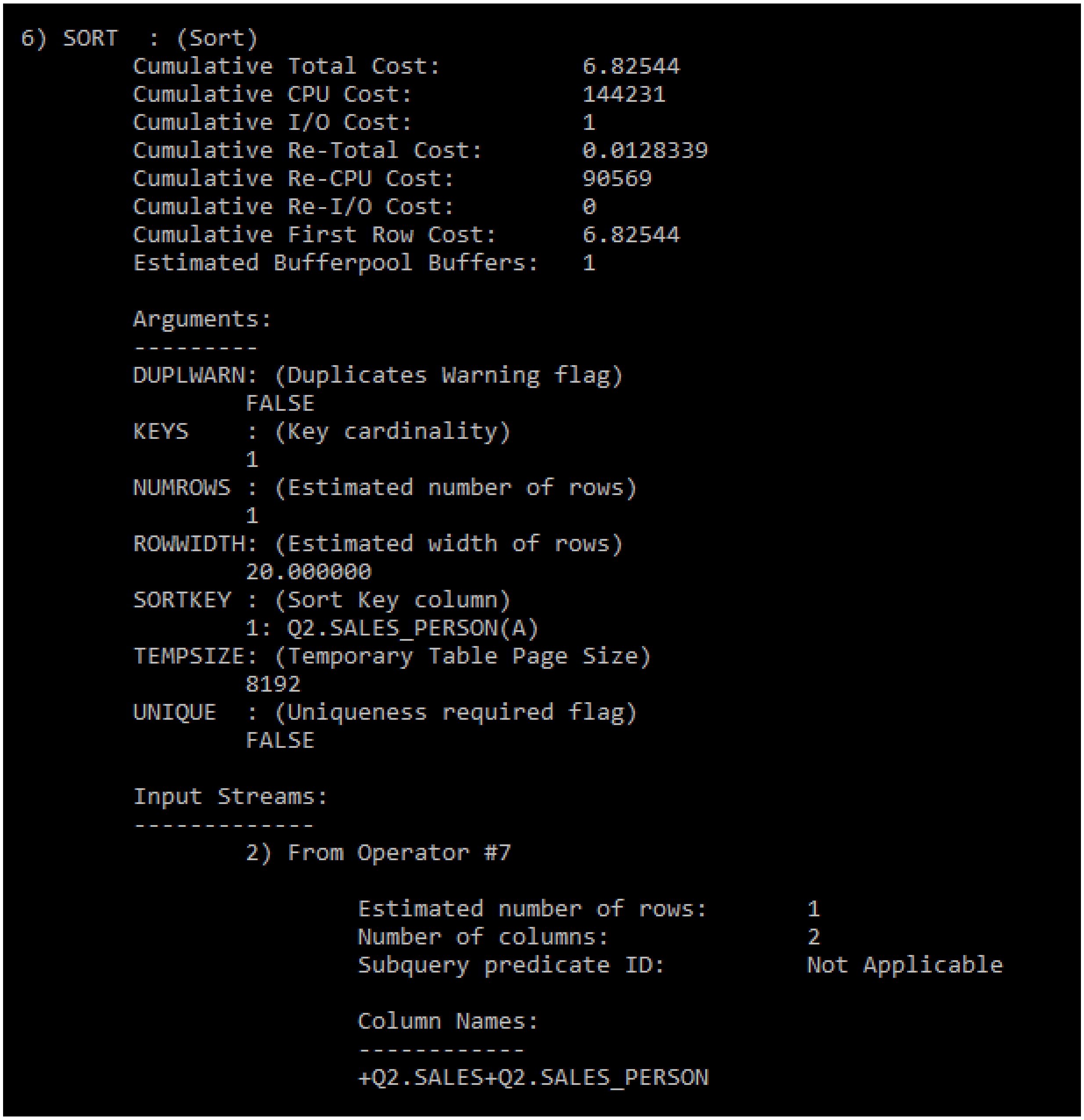
```
            Input Streams:
            -------------
                    5) From Operator #4

                            Estimated number of rows:       1
                            Number of columns:              2
                            Subquery predicate ID:          Not Applicable

                            Column Names:
                            ------------
                            +Q3.$C1+Q3.SALES_PERSON


            Output Streams:
            --------------
                    6) To Operator #2

                            Estimated number of rows:       1
                            Number of columns:              2
                            Subquery predicate ID:          Not Applicable

                            Column Names:
                            ------------
                            +Q3.$C1(A)+Q3.SALES_PERSON


    4) GRPBY : (Group By)
            Cumulative Total Cost:          6.82576
            Cumulative CPU Cost:            146522
            Cumulative I/O Cost:            1
            Cumulative Re-Total Cost:       0.0131586
            Cumulative Re-CPU Cost:         92860
            Cumulative Re-I/O Cost:         0
            Cumulative First Row Cost:      6.82569
            Estimated Bufferpool Buffers:   0

            Arguments:
            ---------
            AGGMODE : (Aggregation Mode)
                    COMPLETE
            GROUPBYC: (Group By columns)
                    TRUE
            GROUPBYN: (Number of Group By columns)
                    1
```



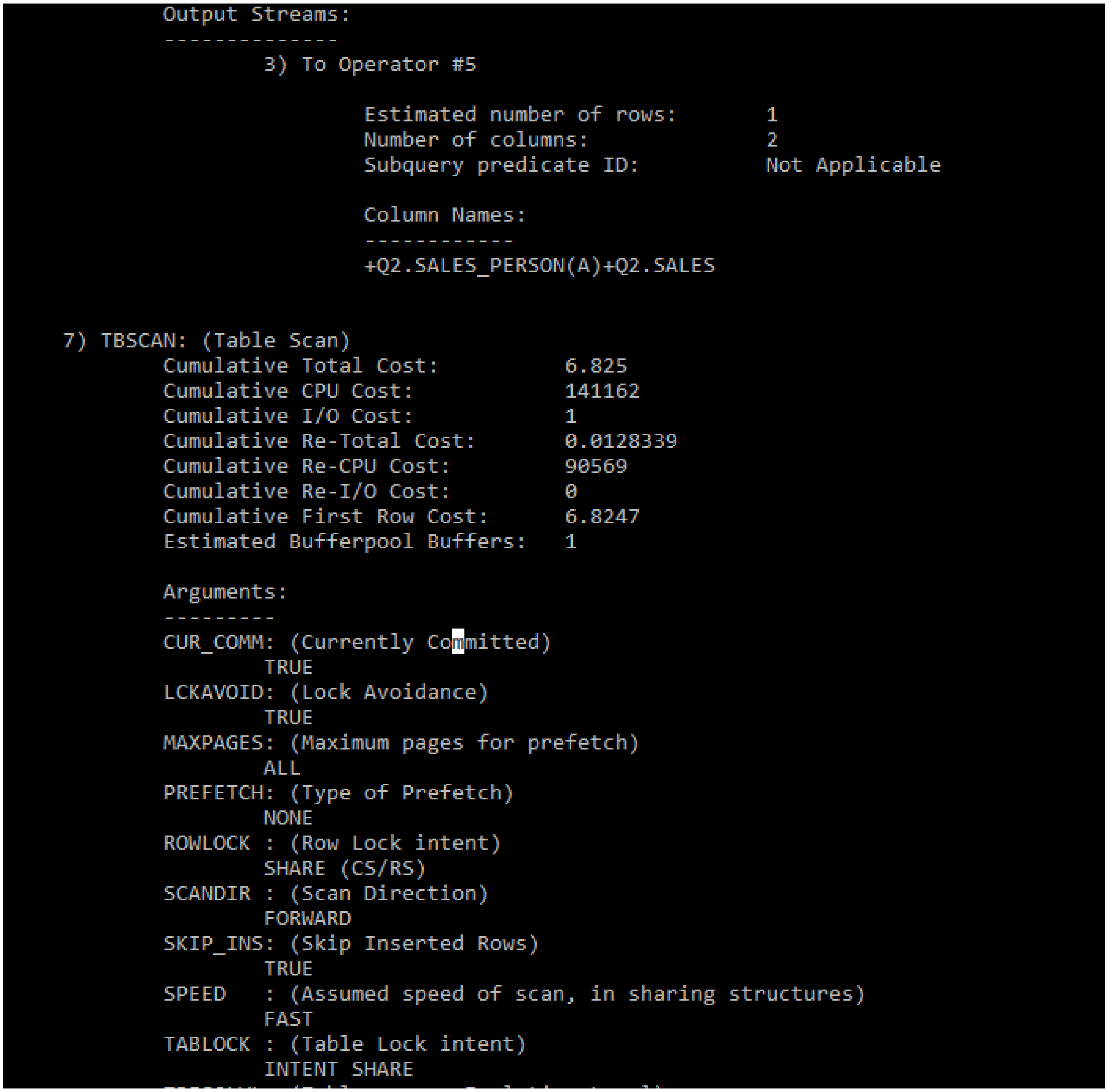
```
        GROUPBYR: (Group By requirement)
                1: Q2.SALES_PERSON
        ONEFETCH: (One Fetch flag)
                FALSE

        Input Streams:
        -------------
                4) From Operator #5

                        Estimated number of rows:       1
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q2.SALES_PERSON(A)+Q2.SALES


        Output Streams:
        --------------
                5) To Operator #3

                        Estimated number of rows:       1
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q3.$C1+Q3.SALES_PERSON


5) TBSCAN: (Table Scan)
        Cumulative Total Cost:          6.82569
        Cumulative CPU Cost:            146018
        Cumulative I/O Cost:            1
        Cumulative Re-Total Cost:       0.0130871
        Cumulative Re-CPU Cost:         92356
        Cumulative Re-I/O Cost:         0
        Cumulative First Row Cost:      6.82565
        Estimated Bufferpool Buffers:   0
```



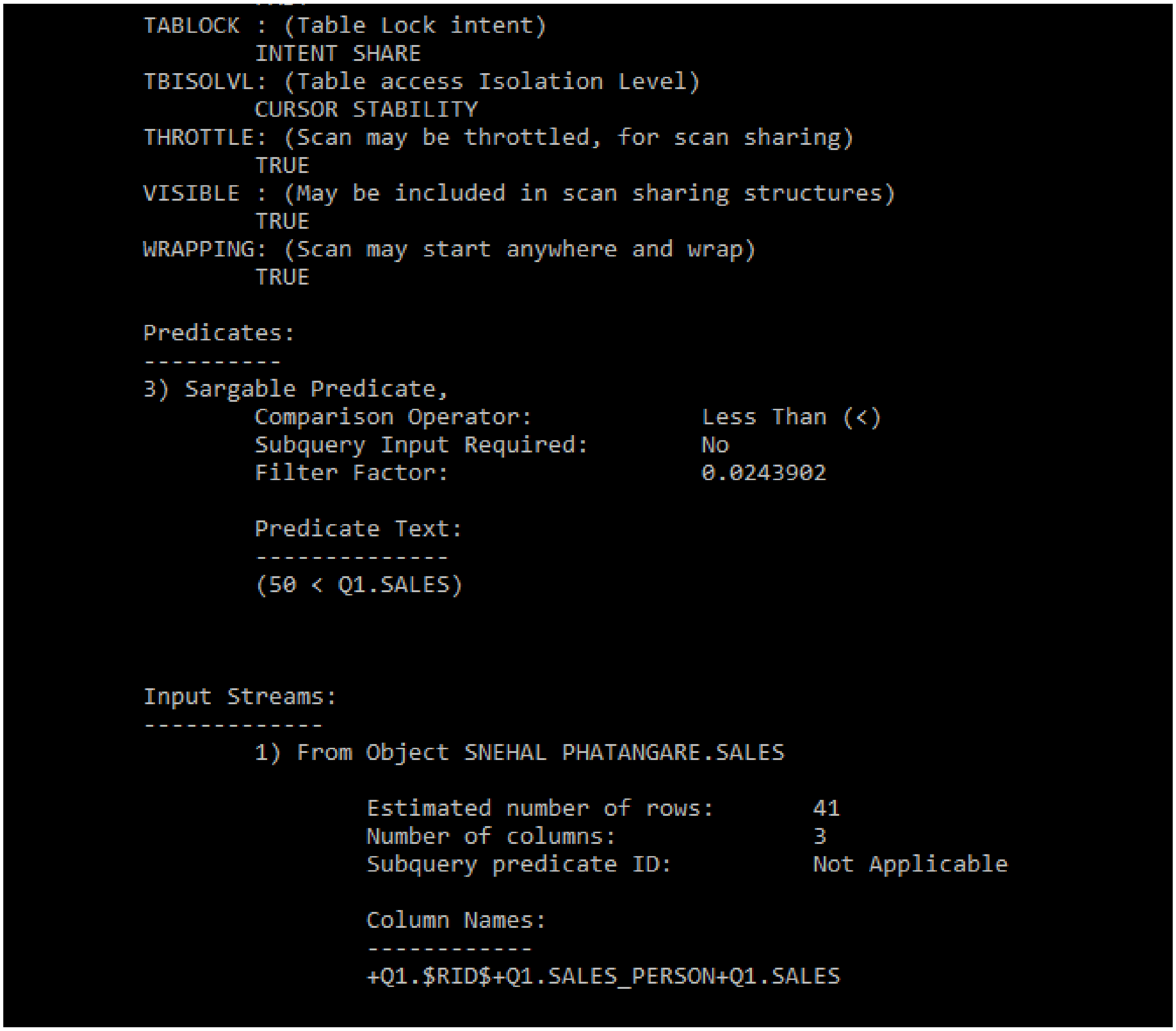
```
Arguments:
---------
MAXPAGES: (Maximum pages for prefetch)
        ALL
PREFETCH: (Type of Prefetch)
        NONE
SCANDIR : (Scan Direction)
        FORWARD
SPEED   : (Assumed speed of scan, in sharing structures)
        SLOW
THROTTLE: (Scan may be throttled, for scan sharing)
        FALSE
VISIBLE : (May be included in scan sharing structures)
        FALSE
WRAPPING: (Scan may start anywhere and wrap)
        FALSE

Input Streams:
-------------
        3) From Operator #6

                Estimated number of rows:       1
                Number of columns:              2
                Subquery predicate ID:          Not Applicable

                Column Names:
                ------------
                +Q2.SALES_PERSON(A)+Q2.SALES


Output Streams:
--------------
        4) To Operator #4

                Estimated number of rows:       1
                Number of columns:              2
                Subquery predicate ID:          Not Applicable

                Column Names:
                ------------
                +Q2.SALES_PERSON(A)+Q2.SALES
```



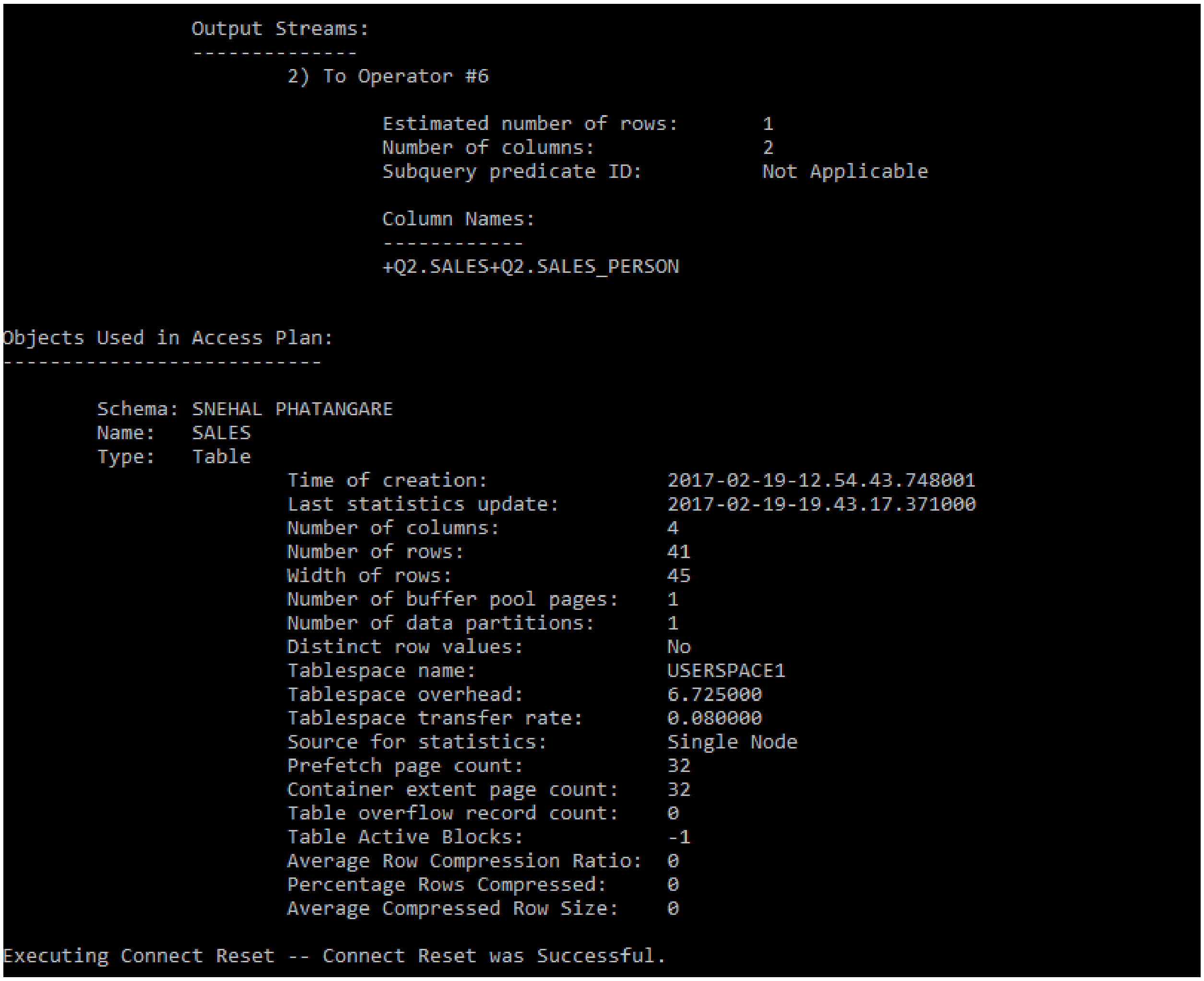
```
6) SORT  : (Sort)
        Cumulative Total Cost:          6.82544
        Cumulative CPU Cost:            144231
        Cumulative I/O Cost:            1
        Cumulative Re-Total Cost:       0.0128339
        Cumulative Re-CPU Cost:         90569
        Cumulative Re-I/O Cost:         0
        Cumulative First Row Cost:      6.82544
        Estimated Bufferpool Buffers:   1

        Arguments:
        ---------
        DUPLWARN: (Duplicates Warning flag)
                FALSE
        KEYS    : (Key cardinality)
                1
        NUMROWS : (Estimated number of rows)
                1
        ROWWIDTH: (Estimated width of rows)
                20.000000
        SORTKEY : (Sort Key column)
                1: Q2.SALES_PERSON(A)
        TEMPSIZE: (Temporary Table Page Size)
                8192
        UNIQUE  : (Uniqueness required flag)
                FALSE

        Input Streams:
        -------------
                2) From Operator #7

                        Estimated number of rows:       1
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q2.SALES+Q2.SALES_PERSON
```



```
        Output Streams:
        --------------
                3) To Operator #5

                        Estimated number of rows:       1
                        Number of columns:              2
                        Subquery predicate ID:          Not Applicable

                        Column Names:
                        ------------
                        +Q2.SALES_PERSON(A)+Q2.SALES


7) TBSCAN: (Table Scan)
        Cumulative Total Cost:          6.825
        Cumulative CPU Cost:            141162
        Cumulative I/O Cost:            1
        Cumulative Re-Total Cost:       0.0128339
        Cumulative Re-CPU Cost:         90569
        Cumulative Re-I/O Cost:         0
        Cumulative First Row Cost:      6.8247
        Estimated Bufferpool Buffers:   1

        Arguments:
        ---------
        CUR_COMM: (Currently Committed)
                TRUE
        LCKAVOID: (Lock Avoidance)
                TRUE
        MAXPAGES: (Maximum pages for prefetch)
                ALL
        PREFETCH: (Type of Prefetch)
                NONE
        ROWLOCK : (Row Lock intent)
                SHARE (CS/RS)
        SCANDIR : (Scan Direction)
                FORWARD
        SKIP_INS: (Skip Inserted Rows)
                TRUE
        SPEED   : (Assumed speed of scan, in sharing structures)
                FAST
        TABLOCK : (Table Lock intent)
                INTENT SHARE
```



```
TABLOCK : (Table Lock intent)
        INTENT SHARE
TBISOLVL: (Table access Isolation Level)
        CURSOR STABILITY
THROTTLE: (Scan may be throttled, for scan sharing)
        TRUE
VISIBLE : (May be included in scan sharing structures)
        TRUE
WRAPPING: (Scan may start anywhere and wrap)
        TRUE

Predicates:
----------
3) Sargable Predicate,
        Comparison Operator:            Less Than (<)
        Subquery Input Required:        No
        Filter Factor:                  0.0243902

        Predicate Text:
        --------------
        (50 < Q1.SALES)



Input Streams:
-------------
        1) From Object SNEHAL PHATANGARE.SALES

                Estimated number of rows:       41
                Number of columns:              3
                Subquery predicate ID:          Not Applicable

                Column Names:
                ------------
                +Q1.$RID$+Q1.SALES_PERSON+Q1.SALES
```

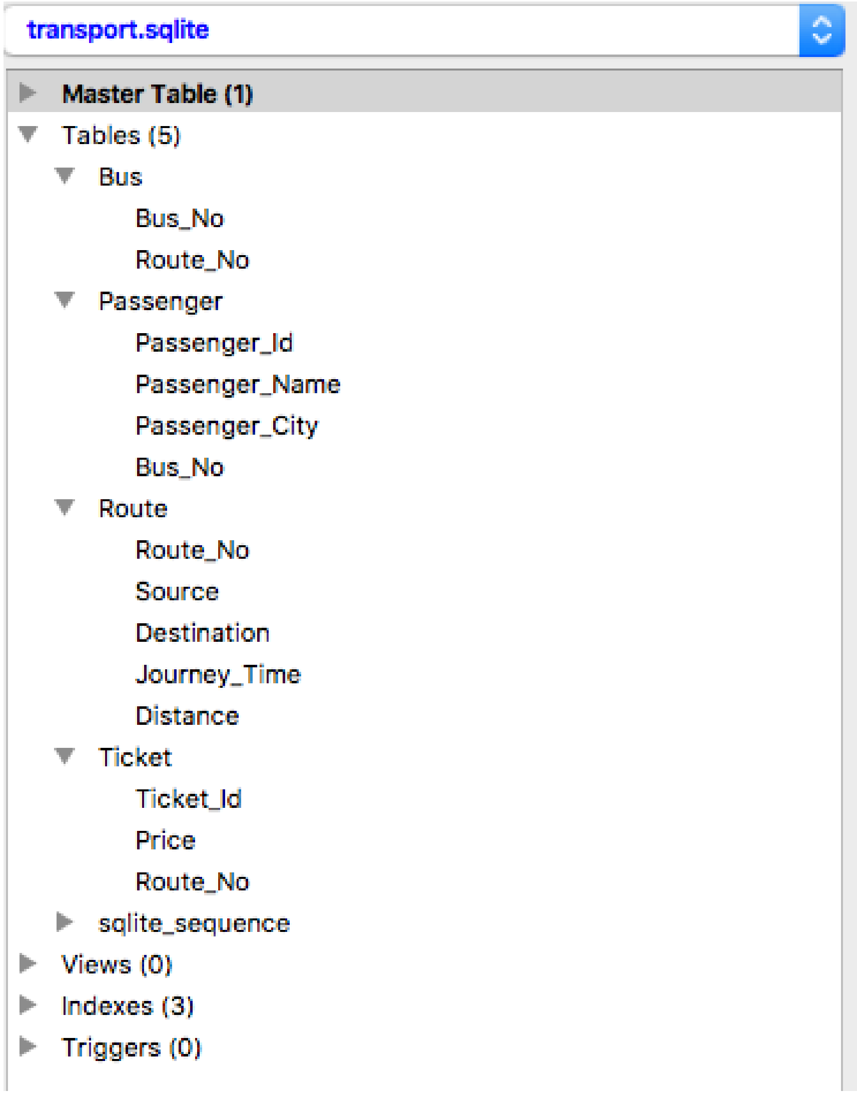


```
                Output Streams:
                --------------
                        2) To Operator #6

                                Estimated number of rows:       1
                                Number of columns:              2
                                Subquery predicate ID:          Not Applicable

                                Column Names:
                                ------------
                                +Q2.SALES+Q2.SALES_PERSON


Objects Used in Access Plan:
---------------------------

        Schema: SNEHAL PHATANGARE
        Name:   SALES
        Type:   Table
                        Time of creation:               2017-02-19-12.54.43.748001
                        Last statistics update:         2017-02-19-19.43.17.371000
                        Number of columns:              4
                        Number of rows:                 41
                        Width of rows:                  45
                        Number of buffer pool pages:    1
                        Number of data partitions:      1
                        Distinct row values:            No
                        Tablespace name:                USERSPACE1
                        Tablespace overhead:            6.725000
                        Tablespace transfer rate:       0.080000
                        Source for statistics:          Single Node
                        Prefetch page count:            32
                        Container extent page count:    32
                        Table overflow record count:    0
                        Table Active Blocks:            -1
                        Average Row Compression Ratio:  0
                        Percentage Rows Compressed:     0
                        Average Compressed Row Size:    0

Executing Connect Reset -- Connect Reset was Successful.
```

SQLite

TRAVEL MANAGEMENT SYSTEM

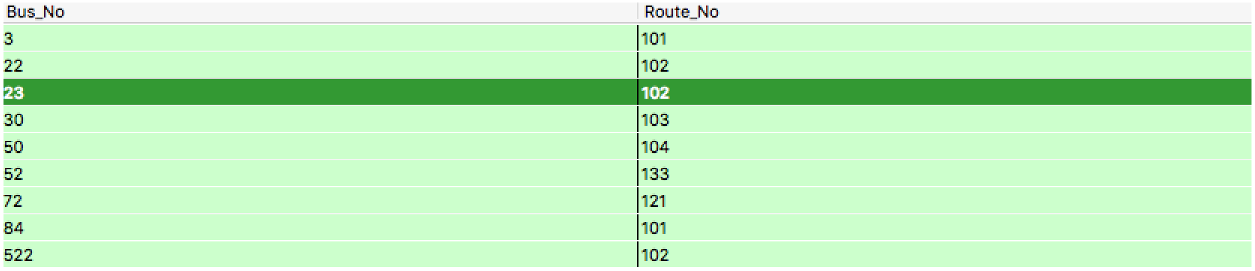
1.1 Database:



```
transport.sqlite
▶ Master Table (1)
▼ Tables (5)
    ▼ Bus
        Bus_No
        Route_No
    ▼ Passenger
        Passenger_Id
        Passenger_Name
        Passenger_City
        Bus_No
    ▼ Route
        Route_No
        Source
        Destination
        Journey_Time
        Distance
    ▼ Ticket
        Ticket_Id
        Price
        Route_No
    ▶ sqlite_sequence
▶ Views (0)
▶ Indexes (3)
▶ Triggers (0)
```

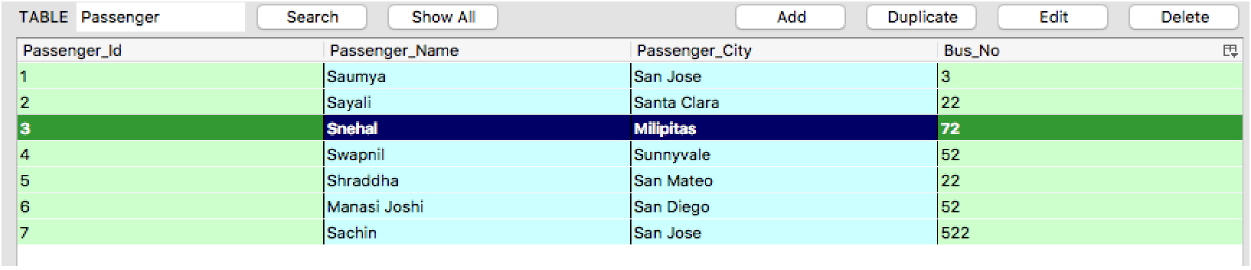
1.2 Database Tables:

1.2.1 Bus Table :



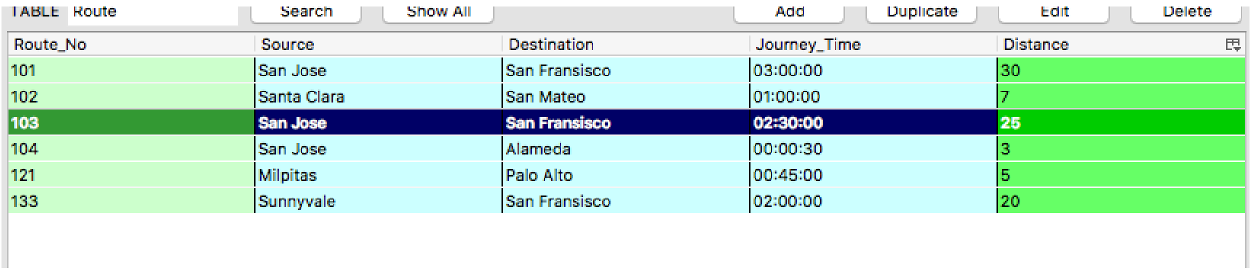
Bus_No	Route_No
3	101
22	102
23	102
30	103
50	104
52	133
72	121
84	101
522	102

1.2.2 Passenger Table:



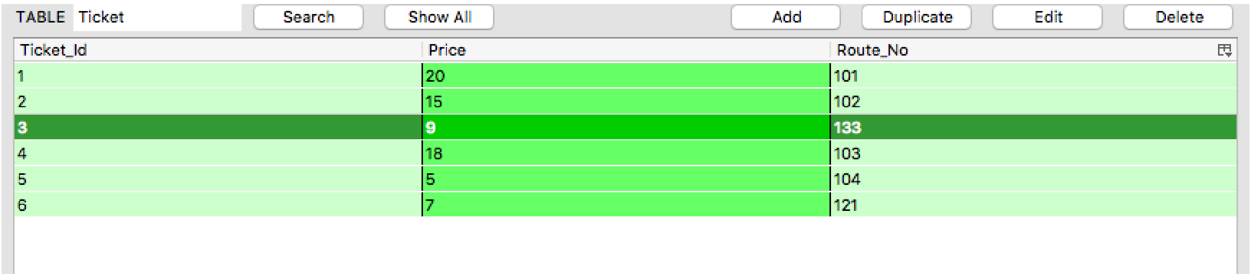
Passenger_Id	Passenger_Name	Passenger_City	Bus_No
1	Saumya	San Jose	3
2	Sayali	Santa Clara	22
3	Snehal	Milipitas	72
4	Swapnil	Sunnyvale	52
5	Shraddha	San Mateo	22
6	Manasi Joshi	San Diego	52
7	Sachin	San Jose	522

1.2.3 Route Table:



Route_No	Source	Destination	Journey_Time	Distance
101	San Jose	San Fransisco	03:00:00	30
102	Santa Clara	San Mateo	01:00:00	7
103	San Jose	San Fransisco	02:30:00	25
104	San Jose	Alameda	00:00:30	3
121	Milpitas	Palo Alto	00:45:00	5
133	Sunnyvale	San Fransisco	02:00:00	20

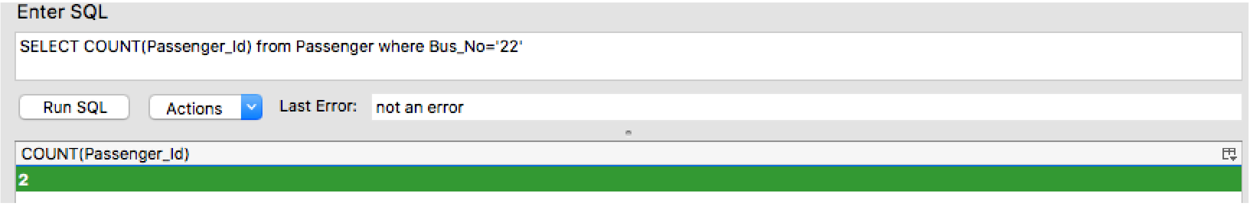
1.2.4 Ticket Table:



Ticket_Id	Price	Route_No
1	20	101
2	15	102
3	9	133
4	18	103
5	5	104
6	7	121

1.3 SQL Queries

Query 1. Get the number of passengers on Bus No 22.

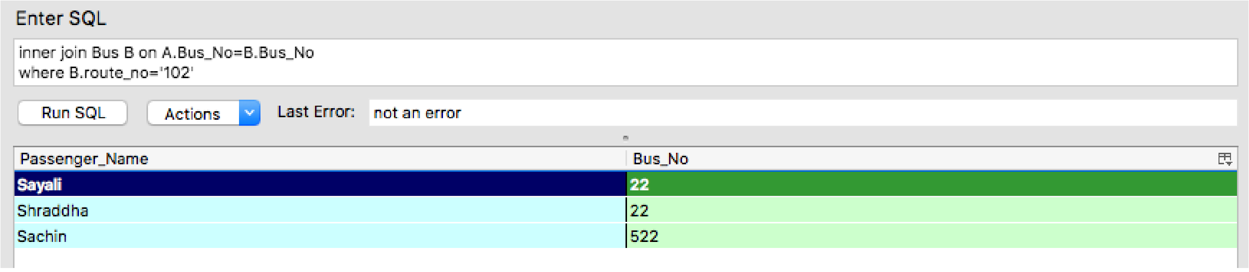


```
SELECT COUNT(Passenger_Id) from Passenger where Bus_No='22'
```

Last Error: not an error

COUNT(Passenger_Id)
2

Query 2. Show all the passengers boarding Bus No 102.

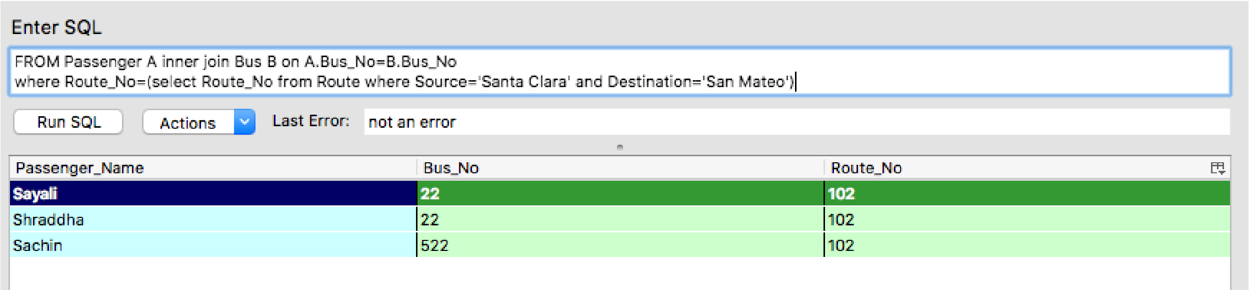


```
inner join Bus B on A.Bus_No=B.Bus_No
where B.route_no='102'
```

Last Error: not an error

Passenger_Name	Bus_No
Sayali	22
Shraddha	22
Sachin	522

Query 3. Show all the passengers travelling between Santa Clara and San Mateo.

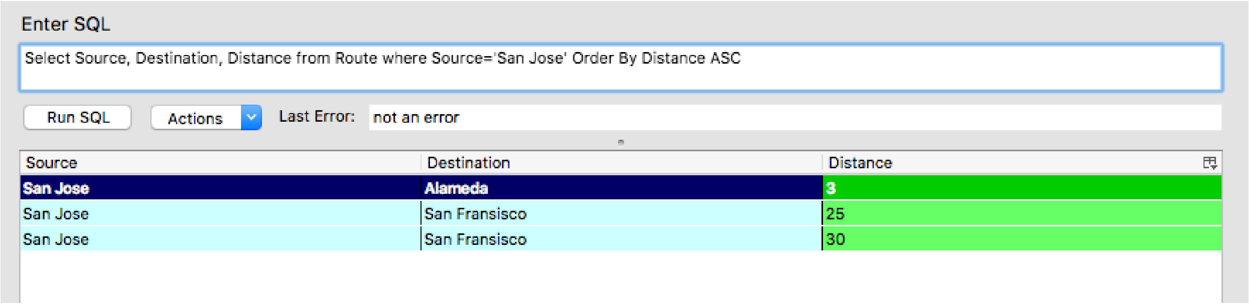


```
FROM Passenger A inner join Bus B on A.Bus_No=B.Bus_No
where Route_No=(select Route_No from Route where Source='Santa Clara' and Destination='San Mateo')
```

Last Error: not an error

Passenger_Name	Bus_No	Route_No
Sayali	22	102
Shraddha	22	102
Sachin	522	102

Query 4.Show the distance of all the stops from San Jose in ascending order.

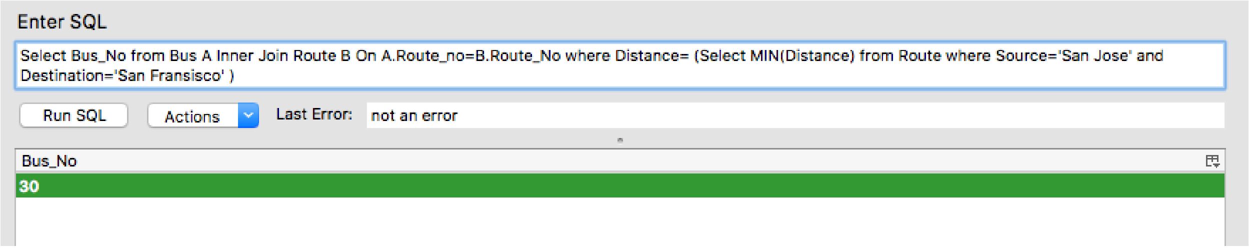


```
Select Source, Destination, Distance from Route where Source='San Jose' Order By Distance ASC
```

Last Error: not an error

Source	Destination	Distance
San Jose	Alameda	3
San Jose	San Fransisco	25
San Jose	San Fransisco	30

Query 5.Show the bus to take to go from San Jose to San Francisco in minimum time

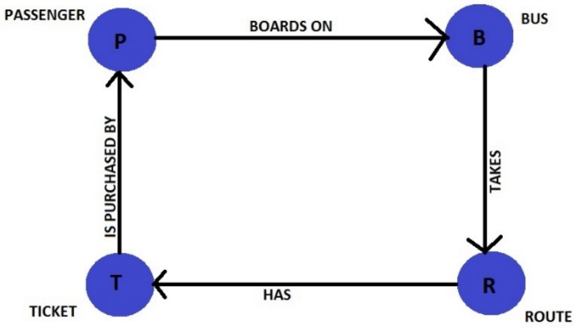


```
Select Bus_No from Bus A Inner Join Route B On A.Route_no=B.Route_No where Distance= (Select MIN(Distance) from Route where Source='San Jose' and Destination='San Fransisco' )
```

not an error

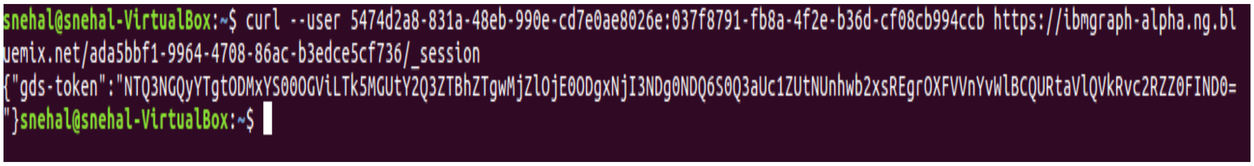
Bus_No
30

IBM BLUEMIX GRAPH



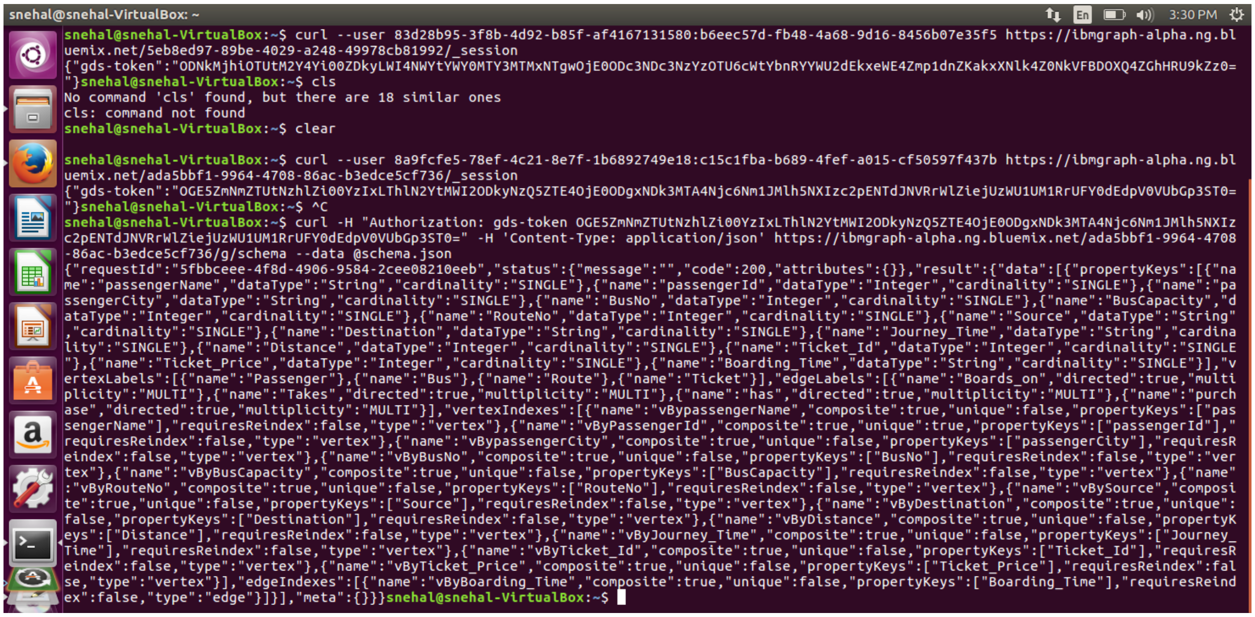
TRAVEL MANAGEMENT SYSTEM

1. Creating Access Token



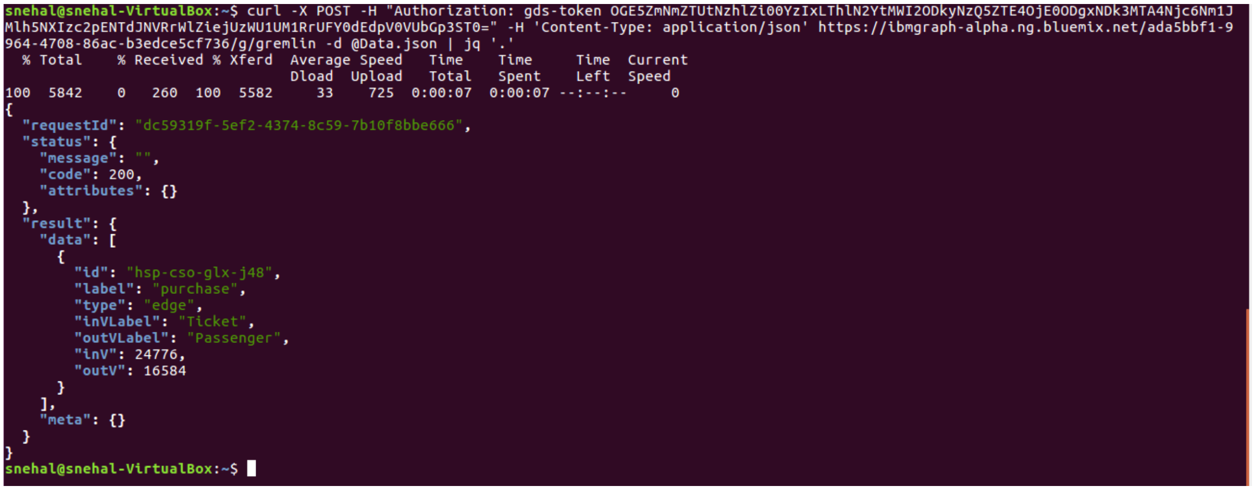
```
snehal@snehal-VirtualBox:~$ curl --user 5474d2a8-831a-48eb-990e-cd7e0ae8026e:037f8791-fb8a-4f2e-b36d-cf08cb994ccb https://ibmgraph-alpha.ng.bl
uemix.net/ada5bbf1-9964-4708-86ac-b3edce5cf736/_session
{"gds-token":"NTQ3NGQyYTgtODMxYS00OGViLTk5MGUtY2Q3ZTBhZTgwMjZlOjE0ODgxNjI3NDg0NDQ6S0Q3aUc1ZUtNUnhwb2xsREgrOXFVVnYvWlBCQURtaVlQVkRvc2RZZ0FIND0=
"}snehal@snehal-VirtualBox:~$ 
```

2.Creating Schema for Travel System Database using schema.json file

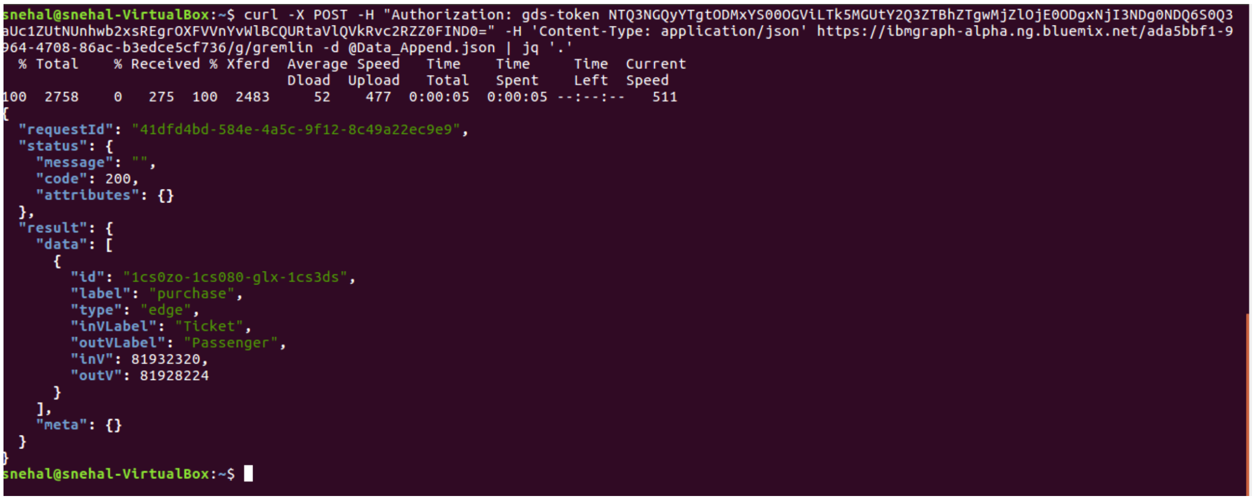


```
snehal@snehal-VirtualBox:~$ curl --user 83d28b95-3f8b-4d92-b85f-af4167131580:b6eec57d-fb48-4a68-9d16-8456b07e35f5 https://ibmgraph-alpha.ng.bl
uemix.net/5eb8ed97-89be-4029-a248-49978cb81992/_session
{"gds-token":"ODNkMjhiOTUtM2Y4Yi00ZDkyLWI4NWYtYWY0MTY3MTMxNTgwOjE0ODc3NDc3NzYzOTU6cWtYbnRYYWU2dEkxeWE4Zmp1dnZKakxXNlk4Z0NkVFBDOXQ4ZGhHRU9kZz0=
"}snehal@snehal-VirtualBox:~$ cls
No command 'cls' found, but there are 18 similar ones
cls: command not found
snehal@snehal-VirtualBox:~$ clear

snehal@snehal-VirtualBox:~$ curl --user 8a9fcfe5-78ef-4c21-8e7f-1b6892749e18:c15c1fba-b689-4fef-a015-cf50597f437b https://ibmgraph-alpha.ng.bl
uemix.net/ada5bbf1-9964-4708-86ac-b3edce5cf736/_session
{"gds-token":"OGE5ZmNmZTUtNzhlZi00YzIxLThlN2YtMWI2ODkyNzQ5ZTE4OjE0ODgxNDk3MTA4Njc6Nm1JMlh5NXIzc2pENTdJNVRrWlZiejUzWU1UM1RrUFY0dEdpV0VUbGp3ST0=
"}snehal@snehal-VirtualBox:~$ ^C
snehal@snehal-VirtualBox:~$ curl -H "Authorization: gds-token OGE5ZmNmZTUtNzhlZi00YzIxLThlN2YtMWI2ODkyNzQ5ZTE4OjE0ODgxNDk3MTA4Njc6Nm1JMlh5NXIz
c2pENTdJNVRrWlZiejUzWU1UM1RrUFY0dEdpV0VUbGp3ST0=" -H 'Content-Type: application/json' https://ibmgraph-alpha.ng.bluemix.net/ada5bbf1-9964-4708
-86ac-b3edce5cf736/g/schema --data @schema.json
{"requestId":"5fbbceee-4f8d-4906-9584-2cee08210eeb","status":{"message":"","code":200,"attributes":{}},"result":{"data":[{"propertyKeys":[{"na
me":"passengerName","dataType":"String","cardinality":"SINGLE"},{"name":"passengerId","dataType":"Integer","cardinality":"SINGLE"},{"name":"pa
ssengerCity","dataType":"String","cardinality":"SINGLE"},{"name":"BusNo","dataType":"Integer","cardinality":"SINGLE"},{"name":"BusCapacity","d
ataType":"Integer","cardinality":"SINGLE"},{"name":"RouteNo","dataType":"Integer","cardinality":"SINGLE"},{"name":"Source","dataType":"String"
,"cardinality":"SINGLE"},{"name":"Destination","dataType":"String","cardinality":"SINGLE"},{"name":"Journey_Time","dataType":"String","cardina
lity":"SINGLE"},{"name":"Distance","dataType":"Integer","cardinality":"SINGLE"},{"name":"Ticket_Id","dataType":"Integer","cardinality":"SINGLE
"},{"name":"Ticket_Price","dataType":"Integer","cardinality":"SINGLE"},{"name":"Boarding_Time","dataType":"String","cardinality":"SINGLE"}],"v
ertexLabels":[{"name":"Passenger"},{"name":"Bus"},{"name":"Route"},{"name":"Ticket"}],"edgeLabels":[{"name":"Boards_on","directed":true,"multi
plicity":"MULTI"},{"name":"Takes","directed":true,"multiplicity":"MULTI"},{"name":"has","directed":true,"multiplicity":"MULTI"},{"name":"purch
ase","directed":true,"multiplicity":"MULTI"}],"vertexIndexes":[{"name":"vBypassengerName","composite":true,"unique":false,"propertyKeys":["pas
sengerName"],"requiresReindex":false,"type":"vertex"},{"name":"vByPassengerId","composite":true,"unique":true,"propertyKeys":["passengerId"],"
requiresReindex":false,"type":"vertex"},{"name":"vBypassengerCity","composite":true,"unique":false,"propertyKeys":["passengerCity"],"requiresR
eindex":false,"type":"vertex"},{"name":"vByBusNo","composite":true,"unique":false,"propertyKeys":["BusNo"],"requiresReindex":false,"type":"ver
tex"},{"name":"vByBusCapacity","composite":true,"unique":false,"propertyKeys":["BusCapacity"],"requiresReindex":false,"type":"vertex"},{"name"
:"vByRouteNo","composite":true,"unique":false,"propertyKeys":["RouteNo"],"requiresReindex":false,"type":"vertex"},{"name":"vBySource","composi
te":true,"unique":false,"propertyKeys":["Source"],"requiresReindex":false,"type":"vertex"},{"name":"vByDestination","composite":true,"unique":
false,"propertyKeys":["Destination"],"requiresReindex":false,"type":"vertex"},{"name":"vByDistance","composite":true,"unique":false,"propertyK
eys":["Distance"],"requiresReindex":false,"type":"vertex"},{"name":"vByJourney_Time","composite":true,"unique":false,"propertyKeys":["Journey_
Time"],"requiresReindex":false,"type":"vertex"},{"name":"vByTicket_Id","composite":true,"unique":false,"propertyKeys":["Ticket_Id"],"requiresR
eindex":false,"type":"vertex"},{"name":"vByTicket_Price","composite":true,"unique":false,"propertyKeys":["Ticket_Price"],"requiresReindex":fal
se,"type":"vertex"}],"edgeIndexes":[{"name":"vByBoarding_Time","composite":true,"unique":false,"propertyKeys":["Boarding_Time"],"requiresReind
ex":false,"type":"edge"}]}],"meta":{}}}snehal@snehal-VirtualBox:~$
```

3.Inserting Data into schema using json file



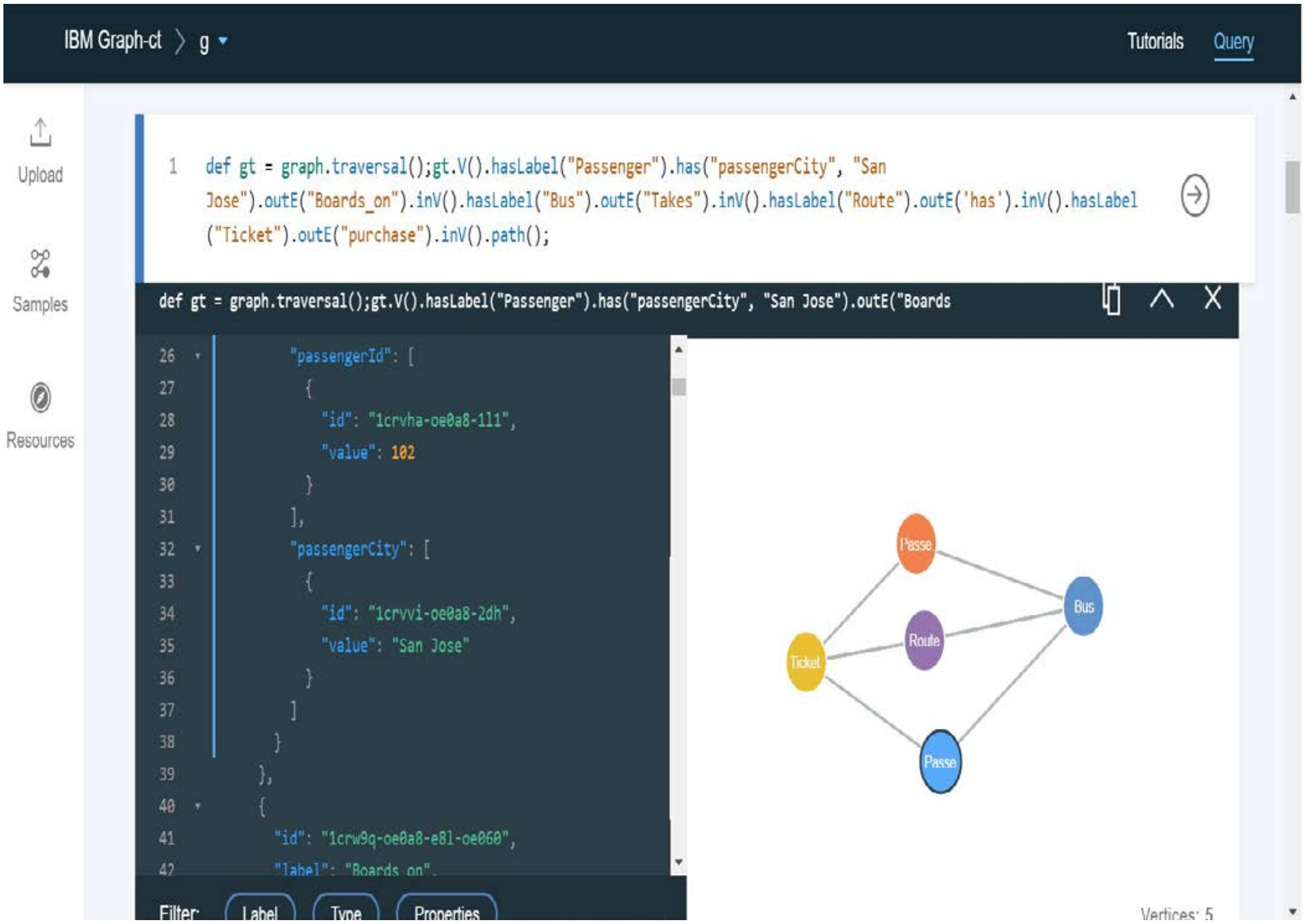
```
snehal@snehal-VirtualBox:~$ curl -X POST -H "Authorization: gds-token OGE5ZmNmZTUtNzhlZi00YzIxLThlN2YtMWI2ODkyNzQ5ZTE4OjE0ODgxNDk3MTA4Njc6Nm1J
Mlh5NXIzc2pENTdJNVRrWlZiejUzWU1UM1RrUFY0dEdpV0VUbGp3ST0=" -H 'Content-Type: application/json' https://ibmgraph-alpha.ng.bluemix.net/ada5bbf1-9
964-4708-86ac-b3edce5cf736/g/gremlin -d @Data.json | jq '.'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5842    0   260  100  5582     33    725  0:00:07  0:00:07 --:--:--     0
{
  "requestId": "dc59319f-5ef2-4374-8c59-7b10f8bbe666",
  "status": {
    "message": "",
    "code": 200,
    "attributes": {}
  },
  "result": {
    "data": [
      {
        "id": "hsp-cso-glx-j48",
        "label": "purchase",
        "type": "edge",
        "inVLabel": "Ticket",
        "outVLabel": "Passenger",
        "inV": 24776,
        "outV": 16584
      }
    ],
    "meta": {}
  }
}
snehal@snehal-VirtualBox:~$
```



```
snehal@snehal-VirtualBox:~$ curl -X POST -H "Authorization: gds-token NTQ3NGQyYTgtODMxYS00OGViLTk5MGUtY2Q3ZTBhZTgwMjZlOjE0ODgxNjI3NDg0NDQ6S0Q3
aUc1ZUtNUnhwb2xsREgrOXFVVnYvWlBCQURtaVlQVkRvc2RZZ0FIND0=" -H 'Content-Type: application/json' https://ibmgraph-alpha.ng.bluemix.net/ada5bbf1-9
964-4708-86ac-b3edce5cf736/g/gremlin -d @Data_Append.json | jq '.'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2758    0   275  100  2483     52    477  0:00:05  0:00:05 --:--:--   511
{
  "requestId": "41dfd4bd-584e-4a5c-9f12-8c49a22ec9e9",
  "status": {
    "message": "",
    "code": 200,
    "attributes": {}
  },
  "result": {
    "data": [
      {
        "id": "1cs0zo-1cs080-glx-1cs3ds",
        "label": "purchase",
        "type": "edge",
        "inVLabel": "Ticket",
        "outVLabel": "Passenger",
        "inV": 81932320,
        "outV": 81928224
      }
    ],
    "meta": {}
  }
}
snehal@snehal-VirtualBox:~$
```

4. Graph Traversal

4.1 Get travel information of passengers residing in San Jose city.

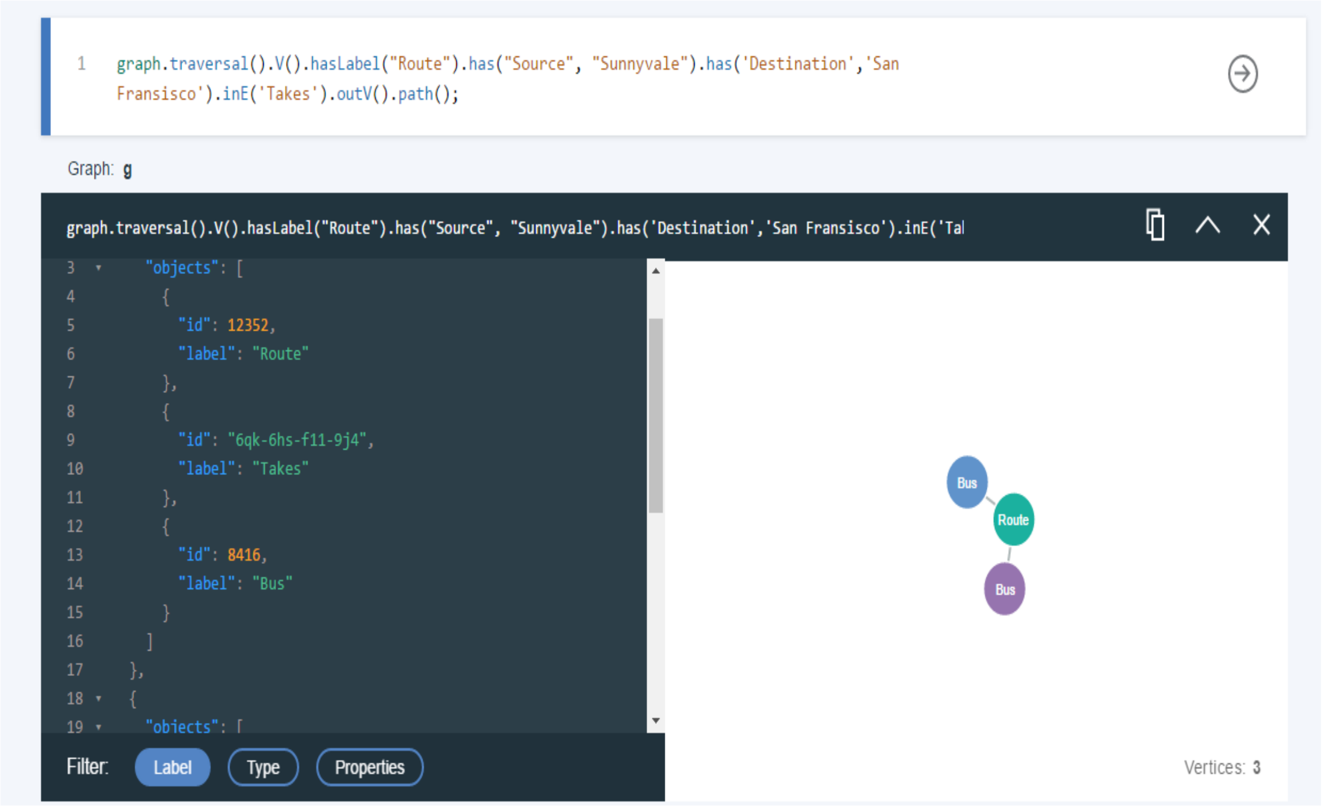


```
def gt = graph.traversal();gt.V().hasLabel("Passenger").has("passengerCity", "San Jose").outE("Boards_on").inV().hasLabel("Bus").outE("Takes").inV().hasLabel("Route").outE('has').inV().hasLabel("Ticket").outE("purchase").inV().path();
```

```
def gt = graph.traversal();gt.V().hasLabel("Passenger").has("passengerCity", "San Jose").outE("Boards
```

```
          "passengerId": [
            {
              "id": "1crvha-oe0a8-1l1",
              "value": 102
            }
          ],
          "passengerCity": [
            {
              "id": "1crvvi-oe0a8-2dh",
              "value": "San Jose"
            }
          ]
        }
      },
      {
        "id": "1crw9q-oe0a8-e8l-oe060",
        "label": "Boards_on",
```

4.2 Get all buses running between Sunnyvale and San Francisco



```
graph.traversal().V().hasLabel("Route").has("Source", "Sunnyvale").has('Destination','San
Fransisco').inE('Takes').outV().path();
```

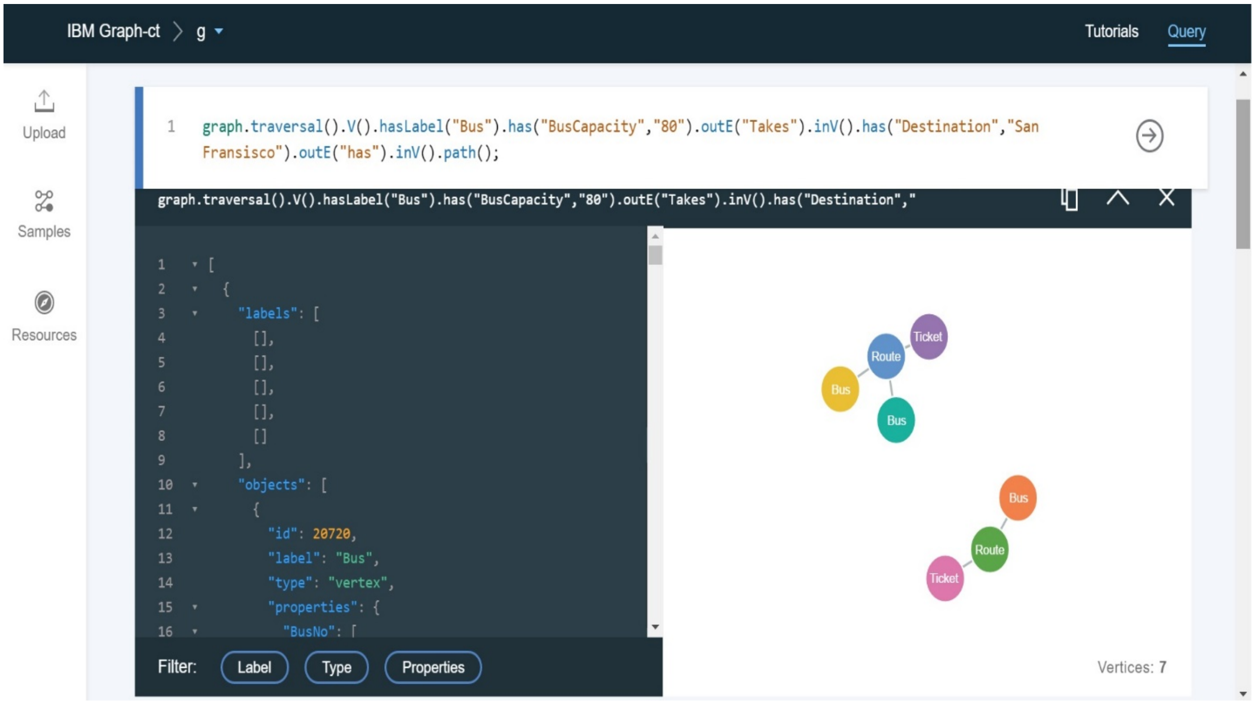
Graph: g

```
graph.traversal().V().hasLabel("Route").has("Source", "Sunnyvale").has('Destination','San Fransisco').inE('Tal
```

```
    "objects": [
      {
        "id": 12352,
        "label": "Route"
      },
      {
        "id": "6qk-6hs-f11-9j4",
        "label": "Takes"
      },
      {
        "id": 8416,
        "label": "Bus"
      }
    ]
  },
  {
    "objects": [
```

Vertices: 3

4.3 Get all buses with Bus Capacity 80, travelling to San Francisco.



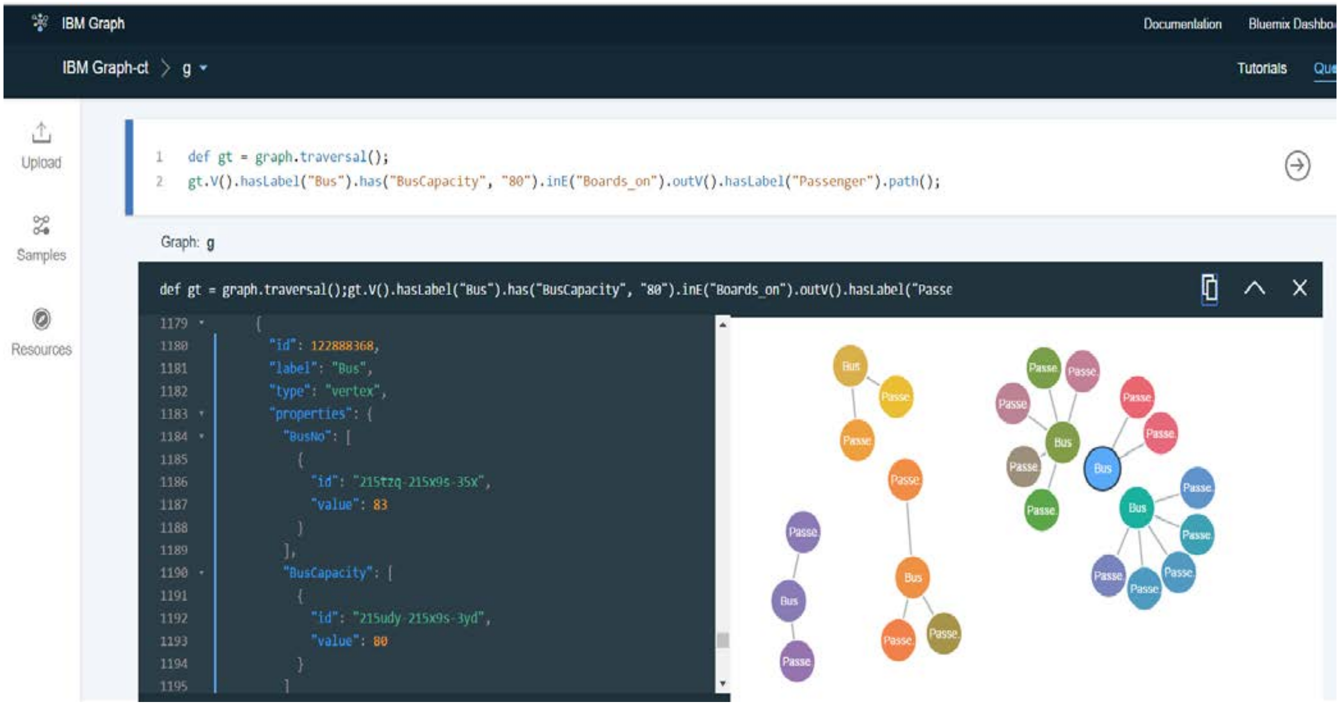
```
graph.traversal().V().hasLabel("Bus").has("BusCapacity","80").outE("Takes").inV().has("Destination","San
Fransisco").outE("has").inV().path();
```

```
graph.traversal().V().hasLabel("Bus").has("BusCapacity","80").outE("Takes").inV().has("Destination","
```

```
[
  {
    "labels": [
      [],
      [],
      [],
      [],
      []
    ],
    "objects": [
      {
        "id": 20720,
        "label": "Bus",
        "type": "vertex",
        "properties": {
          "BusNo": [
```

Vertices: 7

4.4 Get all passengers that boards on bus, having Bus capacity 80 .

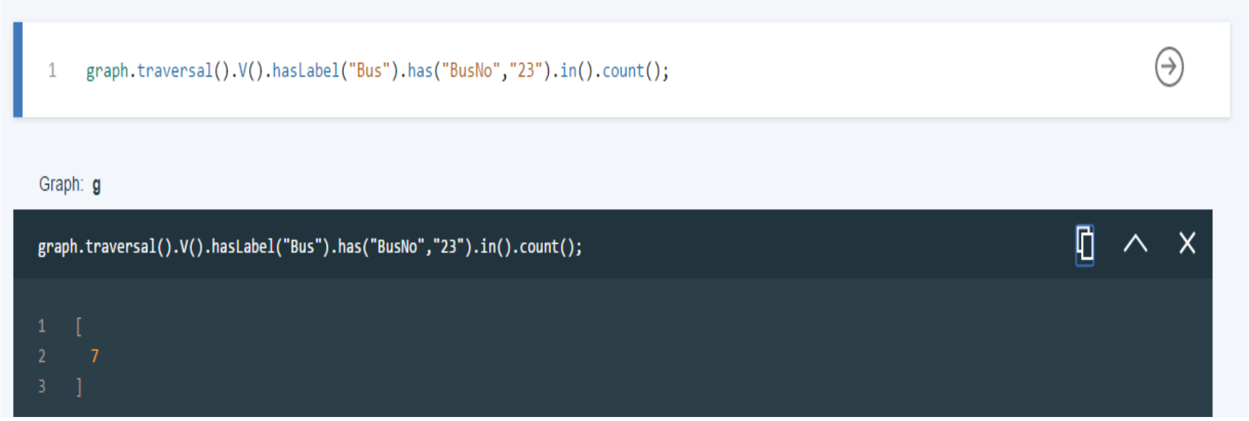


```
def gt = graph.traversal();
gt.V().hasLabel("Bus").has("BusCapacity", "80").inE("Boards_on").outV().hasLabel("Passenger").path();
```

```
def gt = graph.traversal();gt.V().hasLabel("Bus").has("BusCapacity", "80").inE("Boards_on").outV().hasLabel("Passe
```

```
        {
          "id": 122888368,
          "label": "Bus",
          "type": "vertex",
          "properties": {
            "BusNo": [
              {
                "id": "215tzq-215x9s-35x",
                "value": 83
              }
            ],
            "BusCapacity": [
              {
                "id": "215udy-215x9s-3yd",
                "value": 80
              }
            ]
```

4.5 Get the number of passengers travelling on Bus No. 23

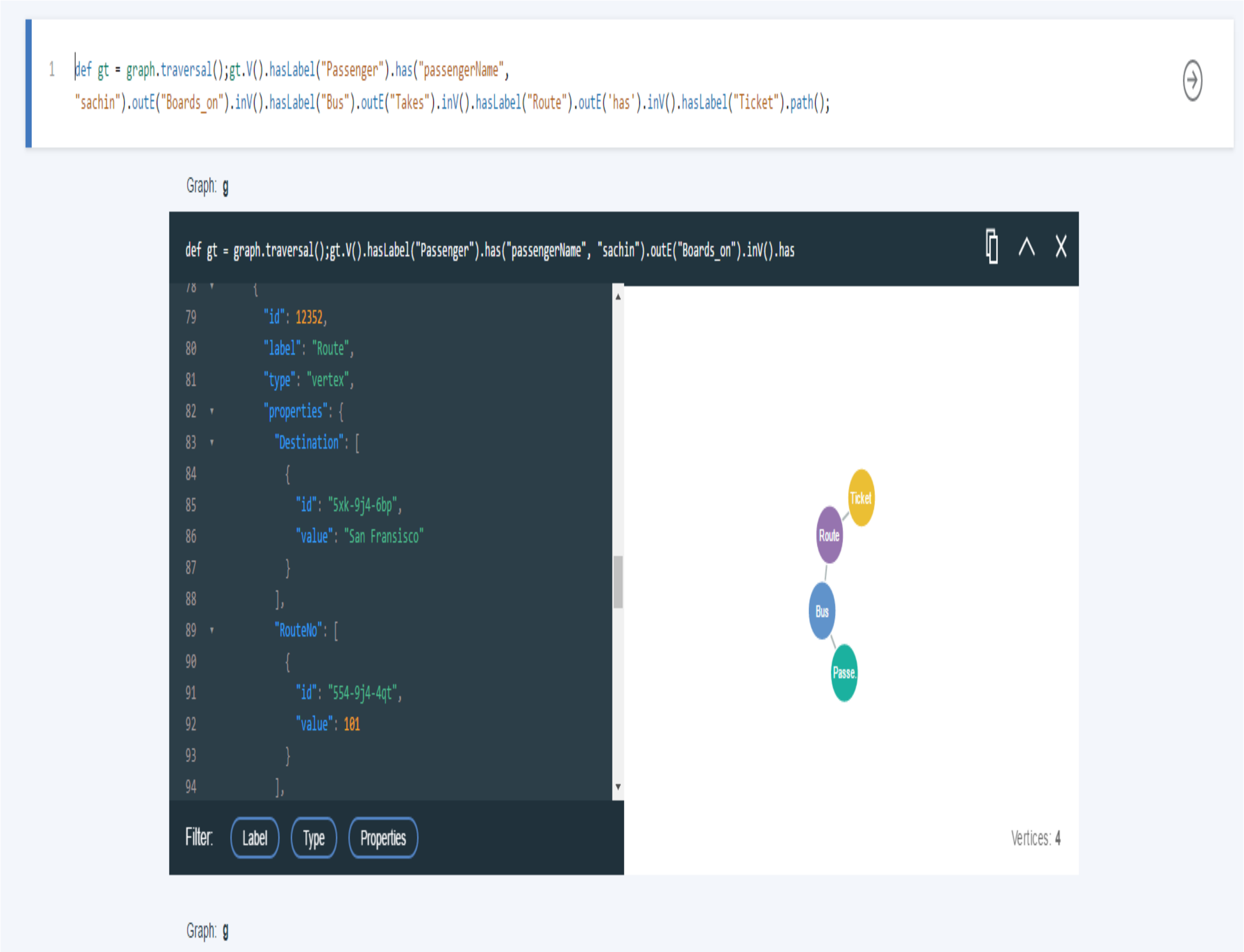


```
graph.traversal().V().hasLabel("Bus").has("BusNo","23").in().count();
```

```
graph.traversal().V().hasLabel("Bus").has("BusNo","23").in().count();
```

```
[
  7
]
```

4.6 Display travel details for Passenger, Sachin.



```
def gt = graph.traversal();gt.V().hasLabel("Passenger").has("passengerName",
"sachin").outE("Boards_on").inV().hasLabel("Bus").outE("Takes").inV().hasLabel("Route").outE('has').inV().hasLabel("Ticket").path();
```

Graph: g

```
def gt = graph.traversal();gt.V().hasLabel("Passenger").has("passengerName", "sachin").outE("Boards_on").inV().has
```

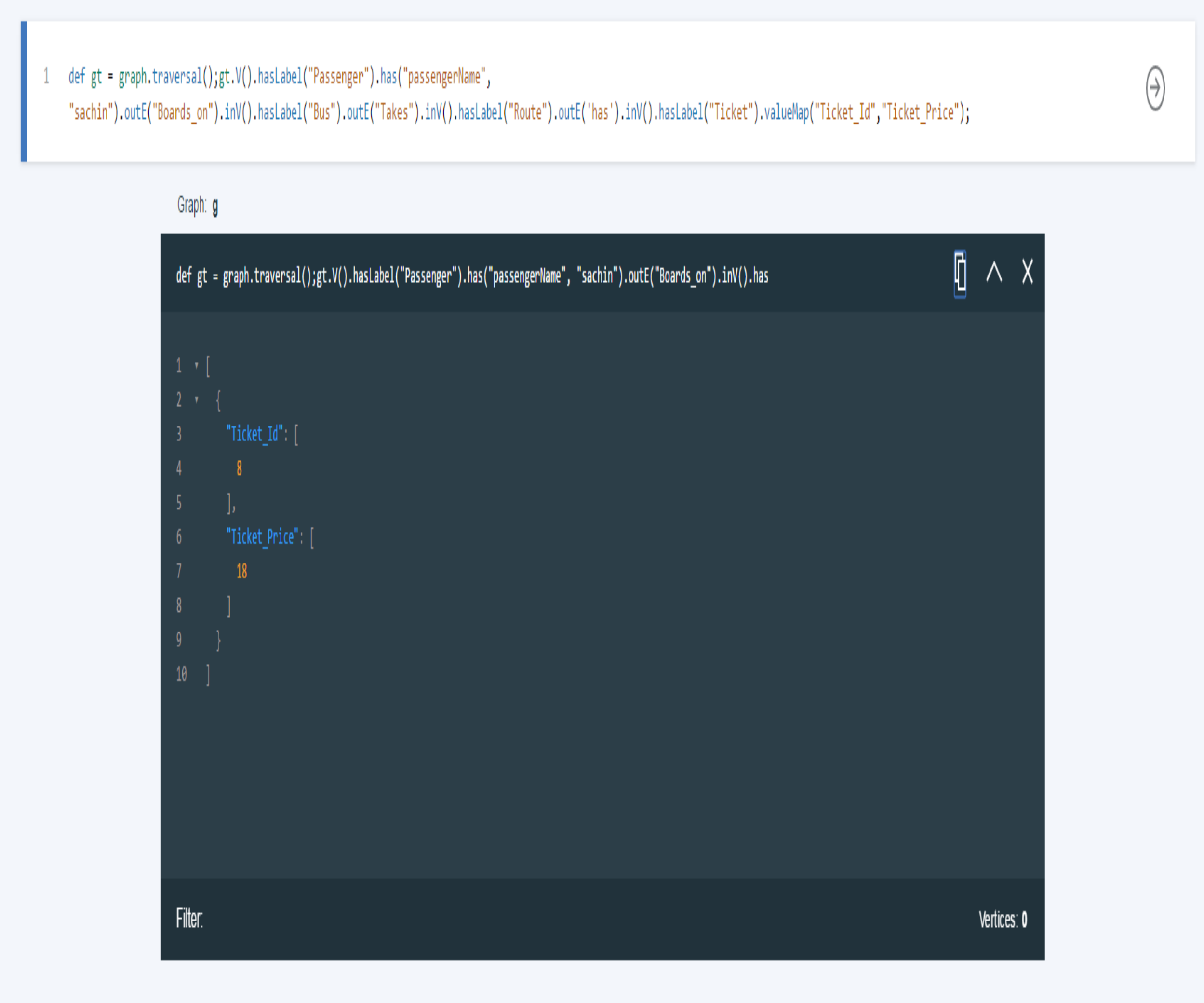
```
      {
        "id": 12352,
        "label": "Route",
        "type": "vertex",
        "properties": {
          "Destination": [
            {
              "id": "5xk-9j4-6bp",
              "value": "San Fransisco"
            }
          ],
          "RouteNo": [
            {
              "id": "554-9j4-4qt",
              "value": 101
            }
          ],
```

Filter: Label Type Properties

Vertices: 4

Graph: g

4.7 Display details of the ticket which Sachin bought.



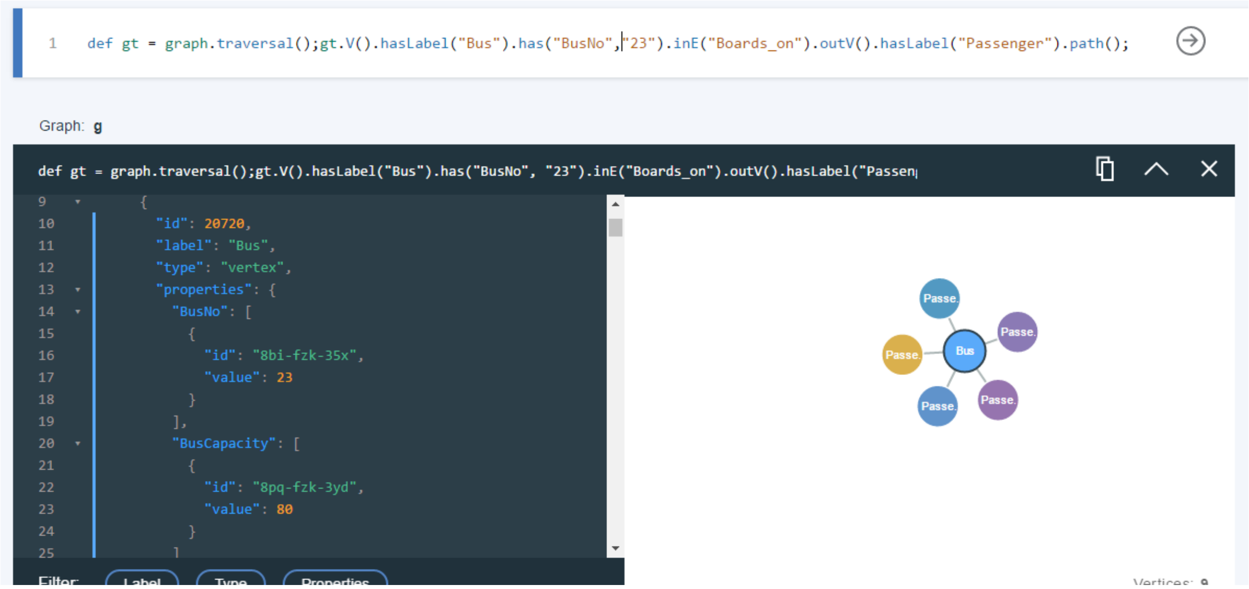
```
def gt = graph.traversal();gt.V().hasLabel("Passenger").has("passengerName",
"sachin").outE("Boards_on").inV().hasLabel("Bus").outE("Takes").inV().hasLabel("Route").outE('has').inV().hasLabel("Ticket").valueMap("Ticket_Id","Ticket_Price");
```

Graph: g

```
def gt = graph.traversal();gt.V().hasLabel("Passenger").has("passengerName", "sachin").outE("Boards_on").inV().has
```

```
[
  {
    "Ticket_Id": [
      8
    ],
    "Ticket_Price": [
      18
    ]
  }
]
```

4.8 Show all passengers boarding Bus No. 23



```
def gt = graph.traversal();gt.V().hasLabel("Bus").has("BusNo","23").inE("Boards_on").outV().hasLabel("Passenger").path();
```

Graph: **g**

```
def gt = graph.traversal();gt.V().hasLabel("Bus").has("BusNo", "23").inE("Boards_on").outV().hasLabel("Passen
```

```
      {
        "id": 20720,
        "label": "Bus",
        "type": "vertex",
        "properties": {
          "BusNo": [
            {
              "id": "8bi-fzk-35x",
              "value": 23
            }
          ],
          "BusCapacity": [
            {
              "id": "8pq-fzk-3yd",
              "value": 80
            }
          ]
```