

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:      SQLC2026
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

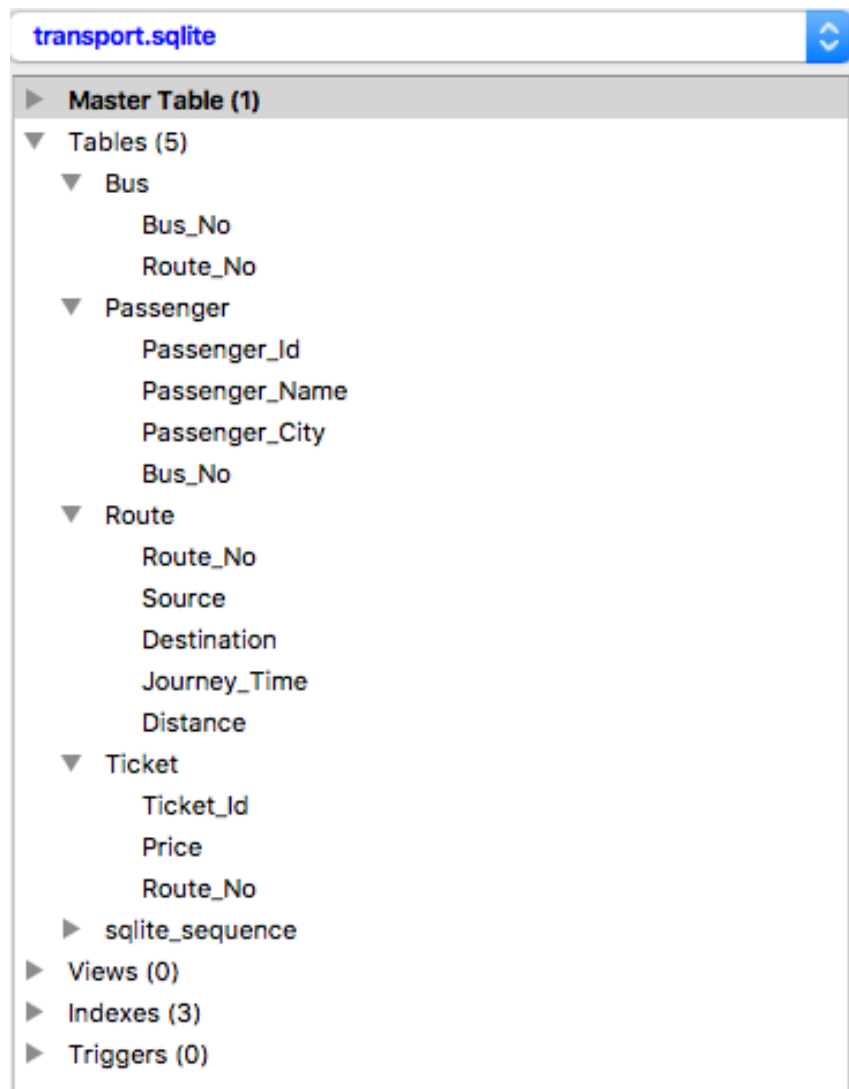
Time of creation: 2017-02-19-12.54.43.748001
Last statistics update: 2017-02-19-19.43.17.371000
Number of columns: 4
Number of rows: 41
Width of rows: 45
Number of buffer pool pages: 1
Number of data partitions: 1
Distinct row values: No
Tablespace name: USERSPACE1
Tablespace overhead: 6.725000
Tablespace 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:



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:

TABLE	Passenger	Search	Show All	Add	Duplicate	Edit	Delete
Passenger_Id	Passenger_Name	Passenger_City	Bus_No				
1	Saumya	San Jose	3				
2	Sayali	Santa Clara	22				
3	Snehal	Milpitas	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:

TABLE	Route	Search	Show All	Add	Duplicate	Edit	Delete
Route_No	Source	Destination	Journey_Time	Distance			
101	San Jose	San Francisco	03:00:00	30			
102	Santa Clara	San Mateo	01:00:00	7			
103	San Jose	San Francisco	02:30:00	25			
104	San Jose	Alameda	00:00:30	3			
121	Milpitas	Palo Alto	00:45:00	5			
133	Sunnyvale	San Francisco	02:00:00	20			

1.2.4 Ticket Table:

TABLE

Ticket

Search

Show All

Add

Duplicate

Edit

Delete

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.

Enter SQL

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

Run SQL Actions Last Error: not an error

COUNT(Passenger_Id)
2

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

Enter SQL

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

Run SQL Actions 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.

Enter SQL

```
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')
```

Run SQL Actions 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.

Enter SQL

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

Run SQL Actions Last Error: not an error

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

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

Enter SQL

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')

Run SQL

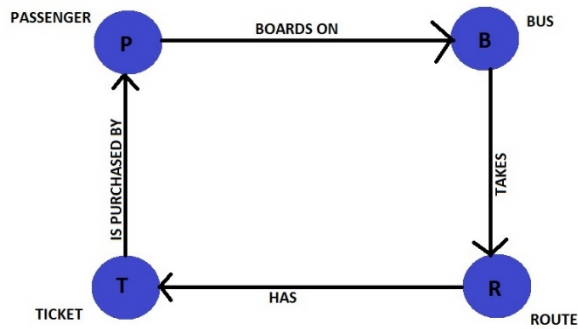
Actions

Last Error: 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.bluemix.net/ada5bbf1-9964-4708-86ac-b3edce5cf736/_session
{"gds-token": "NTQ3NGQyYTgtODMxYS00OGVlLTk5MGUtY2Q3ZTBhZTgwMjZlOjE0ODgxNjI3NDg0NDQ6S0Q3aUc1ZUtNUnhwb2xsREgrOXFVVnYyVWlBCQURtaVlQVkrVc2RZZ0FIND0="}
snehal@snehal-VirtualBox:~$
```

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

```
snehal@snehal-VirtualBox: ~  
snehal@snehal-VirtualBox:~$ curl --user 83d28b95-3f8b-4d92-b85f-af4167131580:b6eec57d-fb48-4a68-9d16-8456b07e35f5 https://lbmgraph-alpha.ng.bl  
uenix.net/5eb8ed97-89be-4029-a248-49978cb81992/_session  
{  
  "gds-token": "ODNMjhhOTUTM2Y4Yl00ZDkyLWI4NWYyYVY0MTY3MTXhNTgwOjE0ODc3NDc3ZnZyZTU6cWtYbnRYVWU2dEkxeWE4Zmp1dnZKakxXNlRk4Z0NkVFBD0XQ4ZGhRU9KzZ0=  
"}  
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://lbmgraph-alpha.ng.bl  
uenix.net/ada5bbf1-9964-4708-86ac-b3edce5cf736/_session  
{  
  "gds-token": "OGESZmNmZTUtNzhZl00YzIxLThlN2YtMWI2ODkyNzQ5ZTE4OjE0ODgxNDk3MTA4Njc6Nm1JMH5NXIzc2pENTdJNVRRWlZleJUZuWU1UM1RrUFY0dEdpV0VubGp3ST0=  
"}  
snehal@snehal-VirtualBox:~$ curl -H "Authorization: gds-token OGESZmNmZTUtNzhZl00YzIxLThlN2YtMWI2ODkyNzQ5ZTE4OjE0ODgxNDk3MTA4Njc6Nm1JMH5NXIz  
c2pENTdJNVRRWlZleJUZuWU1UM1RrUFY0dEdpV0VubGp3ST0=" -H 'Content-Type: application/json' https://lbmgraph-alpha.ng.bluenix.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": [{"name": "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"}, {"  
vertexLabels": [{"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 OGESZmNmZTUtNzhZl00YzIxLThlN2YtMWI2ODkyNzQ5ZTE4OjE0ODgxNDk3MTA4Njc6Nm1J  
MLh5NXIzc2pENTdJNVRRWlZleJUZuWU1UM1RrUFY0dEdpV0VubGp3ST0=" -H 'Content-Type: application/json' https://lbmgraph-alpha.ng.bluenix.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-5ef2-4374-8c59-7b10f8bbe660",  
  "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 NT03NCGyVTgt0DMxYS00CGVLLTk5MGUyV2Q3ZTBhZTgwMjZlQjE0ODgxNjI3NDg0NDQ6S0Q3bUc1ZUENUnhwb2xsREgrOXFVYmVhLBCQURtaVlQYkRvc2RZZ0FIND0=" -H 'Content-Type: application/json' https://ibmgraph-alpha.ng.bluemix.net/ada5bbf1-964-4708-86ac-b3edce5cf736/g/gremlin -d @data_append.json | jq '.'
```

% Total	% Received	% Xferd	Average Dload	Average Upload	Speed	Time Total	Time Spent	Time Left	Current 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:~$
```

```
{
  "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": {}
  }
}
```

4. Graph Traversal

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

IBM Graph-ct > g

Tutorials [Query](#)

Upload

Samples

Resources

```

1  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

```

```

26  "passengerId": [
27    {
28      "id": "1crvha-oe0a8-111",
29      "value": 102
30    }
31  ],
32  "passengerCity": [
33    {
34      "id": "1crvvi-oe0a8-2dh",
35      "value": "San Jose"
36    }
37  ]
38  }
39  },
40  {
41    "id": "1crw9q-oe0a8-e81-oe060",
42    "label": "Boards on".

```

Filter

Label

Type

Properties

→

Vertices: 5

4.2 Get all buses running between Sunnyvale and San Francisco

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

Graph: g

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

```
3  "objects": [  
4    {  
5      "id": 12352,  
6      "label": "Route"  
7    },  
8    {  
9      "id": "6qk-6hs-f11-9j4",  
10     "label": "Takes"  
11   },  
12   {  
13     "id": 8416,  
14     "label": "Bus"  
15   }  
16 ]  
17 },  
18 {  
19   "objects": [  
20     {  
21       "id": 20720,  
22       "label": "Bus",  
23       "type": "vertex",  
24       "properties": {  
25         "BusNo": 1  
26       }  
27     },  
28     {  
29       "id": 20721,  
30       "label": "Route",  
31       "type": "vertex",  
32       "properties": {  
33         "RouteNo": 1  
34       }  
35     },  
36     {  
37       "id": 20722,  
38       "label": "Ticket",  
39       "type": "vertex",  
40       "properties": {  
41         "TicketNo": 1  
42       }  
43     },  
44     {  
45       "id": 20723,  
46       "label": "Bus",  
47       "type": "vertex",  
48       "properties": {  
49         "BusNo": 2  
50       }  
51     },  
52     {  
53       "id": 20724,  
54       "label": "Route",  
55       "type": "vertex",  
56       "properties": {  
57         "RouteNo": 2  
58       }  
59     },  
60     {  
61       "id": 20725,  
62       "label": "Ticket",  
63       "type": "vertex",  
64       "properties": {  
65         "TicketNo": 2  
66       }  
67     },  
68     {  
69       "id": 20726,  
70       "label": "Bus",  
71       "type": "vertex",  
72       "properties": {  
73         "BusNo": 3  
74       }  
75     },  
76     {  
77       "id": 20727,  
78       "label": "Route",  
79       "type": "vertex",  
80       "properties": {  
81         "RouteNo": 3  
82       }  
83     },  
84     {  
85       "id": 20728,  
86       "label": "Ticket",  
87       "type": "vertex",  
88       "properties": {  
89         "TicketNo": 3  
90       }  
91     },  
92     {  
93       "id": 20729,  
94       "label": "Bus",  
95       "type": "vertex",  
96       "properties": {  
97         "BusNo": 4  
98       }  
99     },  
100    {  
101      "id": 20730,  
102      "label": "Route",  
103      "type": "vertex",  
104      "properties": {  
105        "RouteNo": 4  
106      }  
107    },  
108    {  
109      "id": 20731,  
110      "label": "Ticket",  
111      "type": "vertex",  
112      "properties": {  
113        "TicketNo": 4  
114      }  
115    },  
116    {  
117      "id": 20732,  
118      "label": "Bus",  
119      "type": "vertex",  
120      "properties": {  
121        "BusNo": 5  
122      }  
123    },  
124    {  
125      "id": 20733,  
126      "label": "Route",  
127      "type": "vertex",  
128      "properties": {  
129        "RouteNo": 5  
130      }  
131    },  
132    {  
133      "id": 20734,  
134      "label": "Ticket",  
135      "type": "vertex",  
136      "properties": {  
137        "TicketNo": 5  
138      }  
139    },  
140    {  
141      "id": 20735,  
142      "label": "Bus",  
143      "type": "vertex",  
144      "properties": {  
145        "BusNo": 6  
146      }  
147    },  
148    {  
149      "id": 20736,  
150      "label": "Route",  
151      "type": "vertex",  
152      "properties": {  
153        "RouteNo": 6  
154      }  
155    },  
156    {  
157      "id": 20737,  
158      "label": "Ticket",  
159      "type": "vertex",  
160      "properties": {  
161        "TicketNo": 6  
162      }  
163    },  
164    {  
165      "id": 20738,  
166      "label": "Bus",  
167      "type": "vertex",  
168      "properties": {  
169        "BusNo": 7  
170      }  
171    },  
172    {  
173      "id": 20739,  
174      "label": "Route",  
175      "type": "vertex",  
176      "properties": {  
177        "RouteNo": 7  
178      }  
179    },  
180    {  
181      "id": 20740,  
182      "label": "Ticket",  
183      "type": "vertex",  
184      "properties": {  
185        "TicketNo": 7  
186      }  
187    },  
188    {  
189      "id": 20741,  
190      "label": "Bus",  
191      "type": "vertex",  
192      "properties": {  
193        "BusNo": 8  
194      }  
195    },  
196    {  
197      "id": 20742,  
198      "label": "Route",  
199      "type": "vertex",  
200      "properties": {  
201        "RouteNo": 8  
202      }  
203    },  
204    {  
205      "id": 20743,  
206      "label": "Ticket",  
207      "type": "vertex",  
208      "properties": {  
209        "TicketNo": 8  
210      }  
211    },  
212    {  
213      "id": 20744,  
214      "label": "Bus",  
215      "type": "vertex",  
216      "properties": {  
217        "BusNo": 9  
218      }  
219    },  
220    {  
221      "id": 20745,  
222      "label": "Route",  
223      "type": "vertex",  
224      "properties": {  
225        "RouteNo": 9  
226      }  
227    },  
228    {  
229      "id": 20746,  
230      "label": "Ticket",  
231      "type": "vertex",  
232      "properties": {  
233        "TicketNo": 9  
234      }  
235    },  
236    {  
237      "id": 20747,  
238      "label": "Bus",  
239      "type": "vertex",  
240      "properties": {  
241        "BusNo": 10  
242      }  
243    },  
244    {  
245      "id": 20748,  
246      "label": "Route",  
247      "type": "vertex",  
248      "properties": {  
249        "RouteNo": 10  
250      }  
251    },  
252    {  
253      "id": 20749,  
254      "label": "Ticket",  
255      "type": "vertex",  
256      "properties": {  
257        "TicketNo": 10  
258      }  
259    },  
260    {  
261      "id": 20750,  
262      "label": "Bus",  
263      "type": "vertex",  
264      "properties": {  
265        "BusNo": 11  
266      }  
267    },  
268    {  
269      "id": 20751,  
270      "label": "Route",  
271      "type": "vertex",  
272      "properties": {  
273        "RouteNo": 11  
274      }  
275    },  
276    {  
277      "id": 20752,  
278      "label": "Ticket",  
279      "type": "vertex",  
280      "properties": {  
281        "TicketNo": 11  
282      }  
283    },  
284    {  
285      "id": 20753,  
286      "label": "Bus",  
287      "type": "vertex",  
288      "properties": {  
289        "BusNo": 12  
290      }  
291    },  
292    {  
293      "id": 20754,  
294      "label": "Route",  
295      "type": "vertex",  
296      "properties": {  
297        "RouteNo": 12  
298      }  
299    },  
300    {  
301      "id": 20755,  
302      "label": "Ticket",  
303      "type": "vertex",  
304      "properties": {  
305        "TicketNo": 12  
306      }  
307    },  
308    {  
309      "id": 20756,  
310      "label": "Bus",  
311      "type": "vertex",  
312      "properties": {  
313        "BusNo": 13  
314      }  
315    },  
316    {  
317      "id": 20757,  
318      "label": "Route",  
319      "type": "vertex",  
320      "properties": {  
321        "RouteNo": 13  
322      }  
323    },  
324    {  
325      "id": 20758,  
326      "label": "Ticket",  
327      "type": "vertex",  
328      "properties": {  
329        "TicketNo": 13  
330      }  
331    },  
332    {  
333      "id": 20759,  
334      "label": "Bus",  
335      "type": "vertex",  
336      "properties": {  
337        "BusNo": 14  
338      }  
339    },  
340    {  
341      "id": 20760,  
342      "label": "Route",  
343      "type": "vertex",  
344      "properties": {  
345        "RouteNo": 14  
346      }  
347    },  
348    {  
349      "id": 20761,  
350      "label": "Ticket",  
351      "type": "vertex",  
352      "properties": {  
353        "TicketNo": 14  
354      }  
355    },  
356    {  
357      "id": 20762,  
358      "label": "Bus",  
359      "type": "vertex",  
360      "properties": {  
361        "BusNo": 15  
362      }  
363    },  
364    {  
365      "id": 20763,  
366      "label": "Route",  
367      "type": "vertex",  
368      "properties": {  
369        "RouteNo": 15  
370      }  
371    },  
372    {  
373      "id": 20764,  
374      "label": "Ticket",  
375      "type": "vertex",  
376      "properties": {  
377        "TicketNo": 15  
378      }  
379    },  
380    {  
381      "id": 20765,  
382      "label": "Bus",  
383      "type": "vertex",  
384      "properties": {  
385        "BusNo": 16  
386      }  
387    },  
388    {  
389      "id": 20766,  
390      "label": "Route",  
391      "type": "vertex",  
392      "properties": {  
393        "RouteNo": 16  
394      }  
395    },  
396    {  
397      "id": 20767,  
398      "label": "Ticket",  
399      "type": "vertex",  
400      "properties": {  
401        "TicketNo": 16  
402      }  
403    },  
404    {  
405      "id": 20768,  
406      "label": "Bus",  
407      "type": "vertex",  
408      "properties": {  
409        "BusNo": 17  
410      }  
411    },  
412    {  
413      "id": 20769,  
414      "label": "Route",  
415      "type": "vertex",  
416      "properties": {  
417        "RouteNo": 17  
418      }  
419    },  
420    {  
421      "id": 20770,  
422      "label": "Ticket",  
423      "type": "vertex",  
424      "properties": {  
425        "TicketNo": 17  
426      }  
427    },  
428    {  
429      "id": 20771,  
430      "label": "Bus",  
431      "type": "vertex",  
432      "properties": {  
433        "BusNo": 18  
434      }  
435    },  
436    {  
437      "id": 20772,  
438      "label": "Route",  
439      "type": "vertex",  
440      "properties": {  
441        "RouteNo": 18  
442      }  
443    },  
444    {  
445      "id": 20773,  
446      "label": "Ticket",  
447      "type": "vertex",  
448      "properties": {  
449        "TicketNo": 18  
450      }  
451    },  
452    {  
453      "id": 20774,  
454      "label": "Bus",  
455      "type": "vertex",  
456      "properties": {  
457        "BusNo": 19  
458      }  
459    },  
460    {  
461      "id": 20775,  
462      "label": "Route",  
463      "type": "vertex",  
464      "properties": {  
465        "RouteNo": 19  
466      }  
467    },  
468    {  
469      "id": 20776,  
470      "label": "Ticket",  
471      "type": "vertex",  
472      "properties": {  
473        "TicketNo": 19  
474      }  
475    },  
476    {  
477      "id": 20777,  
478      "label": "Bus",  
479      "type": "vertex",  
480      "properties": {  
481        "BusNo": 20  
482      }  
483    },  
484    {  
485      "id": 20778,  
486      "label": "Route",  
487      "type": "vertex",  
488      "properties": {  
489        "RouteNo": 20  
490      }  
491    },  
492    {  
493      "id": 20779,  
494      "label": "Ticket",  
495      "type": "vertex",  
496      "properties": {  
497        "TicketNo": 20  
498      }  
499    },  
500    {  
501      "id": 20780,  
502      "label": "Bus",  
503      "type": "vertex",  
504      "properties": {  
505        "BusNo": 21  
506      }  
507    },  
508    {  
509      "id": 20781,  
510      "label": "Route",  
511      "type": "vertex",  
512      "properties": {  
513        "RouteNo": 21  
514      }  
515    },  
516    {  
517      "id": 20782,  
518      "label": "Ticket",  
519      "type": "vertex",  
520      "properties": {  
521        "TicketNo": 21  
522      }  
523    },  
524    {  
525      "id": 20783,  
526      "label": "Bus",  
527      "type": "vertex",  
528      "properties": {  
529        "BusNo": 22  
530      }  
531    },  
532    {  
533      "id": 20784,  
534      "label": "Route",  
535      "type": "vertex",  
536      "properties": {  
537        "RouteNo": 22  
538      }  
539    },  
540    {  
541      "id": 20785,  
542      "label": "Ticket",  
543      "type": "vertex",  
544      "properties": {  
545        "TicketNo": 22  
546      }  
547    },  
548    {  
549      "id": 20786,  
550      "label": "Bus",  
551      "type": "vertex",  
552      "properties": {  
553        "BusNo": 23  
554      }  
555    },  
556    {  
557      "id": 20787,  
558      "label": "Route",  
559      "type": "vertex",  
560      "properties": {  
561        "RouteNo": 23  
562      }  
563    },  
564    {  
565      "id": 20788,  
566      "label": "Ticket",  
567      "type": "vertex",  
568      "properties": {  
569        "TicketNo": 23  
570      }  
571    },  
572    {  
573      "id": 20789,  
574      "label": "Bus",  
575      "type": "vertex",  
576      "properties": {  
577        "BusNo": 24  
578      }  
579    },  
580    {  
581      "id": 20790,  
582      "label": "Route",  
583      "type": "vertex",  
584      "properties": {  
585        "RouteNo": 24  
586      }  
587    },  
588    {  
589      "id": 20791,  
590      "label": "Ticket",  
591      "type": "vertex",  
592      "properties": {  
593        "TicketNo": 24  
594      }  
595    },  
596    {  
597      "id": 20792,  
598      "label": "Bus",  
599      "type": "vertex",  
600      "properties": {  
601        "BusNo": 25  
602      }  
603    },  
604    {  
605      "id": 20793,  
606      "label": "Route",  
607      "type": "vertex",  
608      "properties": {  
609        "RouteNo": 25  
610      }  
611    },  
612    {  
613      "id": 20794,  
614      "label": "Ticket",  
615      "type": "vertex",  
616      "properties": {  
617        "TicketNo": 25  
618      }  
619    },  
620    {  
621      "id": 20795,  
622      "label": "Bus",  
623      "type": "vertex",  
624      "properties": {  
625        "BusNo": 26  
626      }  
627    },  
628    {  
629      "id": 20796,  
630      "label": "Route",  
631      "type": "vertex",  
632      "properties": {  
633        "RouteNo": 26  
634      }  
635    },  
636    {  
637      "id": 20797,  
638      "label": "Ticket",  
639      "type": "vertex",  
640      "properties": {  
641        "TicketNo": 26  
642      }  
643    },  
644    {  
645      "id": 20798,  
646      "label": "Bus",  
647      "type": "vertex",  
648      "properties": {  
649        "BusNo": 27  
650      }  
651    },  
652    {  
653      "id": 20799,  
654      "label": "Route",  
655      "type": "vertex",  
656      "properties": {  
657        "RouteNo": 27  
658      }  
659    },  
660    {  
661      "id": 20800,  
662      "label": "Ticket",  
663      "type": "vertex",  
664      "properties": {  
665        "TicketNo": 27  
666      }  
667    },  
668    {  
669      "id": 20801,  
670      "label": "Bus",  
671      "type": "vertex",  
672      "properties": {  
673        "BusNo": 28  
674      }  
675    },  
676    {  
677      "id": 20802,  
678      "label": "Route",  
679      "type": "vertex",  
680      "properties": {  
681        "RouteNo": 28  
682      }  
683    },  
684    {  
685      "id": 20803,  
686      "label": "Ticket",  
687      "type": "vertex",  
688      "properties": {  
689        "TicketNo": 28  
690      }  
691    },  
692    {  
693      "id": 20804,  
694      "label": "Bus",  
695      "type": "vertex",  
696      "properties": {  
697        "BusNo": 29  
698      }  
699    },  
700    {  
701      "id": 20805,  
702      "label": "Route",  
703      "type": "vertex",  
704      "properties": {  
705        "RouteNo": 29  
706      }  
707    },  
708    {  
709      "id": 20806,  
710      "label": "Ticket",  
711      "type": "vertex",  
712      "properties": {  
713        "TicketNo": 29  
714      }  
715    },  
716    {  
717      "id": 20807,  
718      "label": "Bus",  
719      "type": "vertex",  
720      "properties": {  
721        "BusNo": 30  
722      }  
723    },  
724    {  
725      "id": 20808,  
726      "label": "Route",  
727      "type": "vertex",  
728      "properties": {  
729        "RouteNo": 30  
730      }  
731    },  
732    {  
733      "id": 20809,  
734      "label": "Ticket",  
735      "type": "vertex",  
736      "properties": {  
737        "TicketNo": 30  
738      }  
739    },  
740    {  
741      "id": 20810,  
742      "label": "Bus",  
743      "type": "vertex",  
744      "properties": {  
745        "BusNo": 31  
746      }  
747    },  
748    {  
749      "id": 20811,  
750      "label": "Route",  
751      "type": "vertex",  
752      "properties": {  
753        "RouteNo": 31  
754      }  
755    },  
756    {  
757      "id": 20812,  
758      "label": "Ticket",  
759      "type": "vertex",  
760      "properties": {  
761        "TicketNo": 31  
762      }  
763    },  
764    {  
765      "id": 20813,  
766      "label": "Bus",  
767      "type": "vertex",  
768      "properties": {  
769        "BusNo": 32  
770      }  
771    },  
772    {  
773      "id": 20814,  
774      "label": "Route",  
775      "type": "vertex",  
776      "properties": {  
777        "RouteNo": 32  
778      }  
779    },  
780    {  
781      "id": 20815,  
782      "label": "Ticket",  
783      "type": "vertex",  
784      "properties": {  
785        "TicketNo": 32  
786      }  
787    },  
788    {  
789      "id": 20816,  
790      "label": "Bus",  
791      "type": "vertex",  
792      "properties": {  
793        "BusNo": 33  
794      }  
795    },  
796    {  
797      "id": 20817,  
798      "label": "Route",  
799      "type": "vertex",  
800      "properties": {  
801        "RouteNo": 33  
802      }  
803    },  
804    {  
805      "id": 20818,  
806      "label": "Ticket",  
807      "type": "vertex",  
808      "properties": {  
809        "TicketNo": 33  
810      }  
811    },  
812    {  
813      "id": 20819,  
814      "label": "Bus",  
815      "type": "vertex",  
816      "properties": {  
817        "BusNo": 34  
818      }  
819    },  
820    {  
821      "id": 20820,  
822      "label": "Route",  
823      "type": "vertex",  
824      "properties": {  
825        "RouteNo": 34  
826      }  
827    },  
828    {  
829      "id": 20821,  
830      "label": "Ticket",  
831      "type": "vertex",  
832      "properties": {  
833        "TicketNo": 34  
834      }  
835    },  
836    {  
837      "id": 20822,  
838      "label": "Bus",  
839      "type": "vertex",  
840      "properties": {  
841        "BusNo": 35  
842      }  
843    },  
844    {  
845      "id": 20823,  
846      "label": "Route",  
847      "type": "vertex",  
848      "properties": {  
849        "RouteNo": 35  
850      }  
851    },  
852    {  
853      "id": 20824,  
854      "label": "Ticket",  
855      "type": "vertex",  
856      "properties": {  
857        "TicketNo": 35  
858      }  
859    },  
860    {  
861      "id": 20825,  
862      "label": "Bus",  
863      "type": "vertex",  
864      "properties": {  
865        "BusNo": 36  
866      }  
867    },  
868    {  
869      "id": 20826,  
870      "label": "Route",  
871      "type": "vertex",  
872      "properties": {  
873        "RouteNo": 36  
874      }  
875    },  
876    {  
877      "id": 20827,  
878      "label": "Ticket",  
879      "type": "vertex",  
880      "properties": {  
881        "TicketNo": 36  
882      }  
883    },  
884    {  
885      "id": 20828,  
886      "label": "Bus",  
887      "type": "vertex",  
888      "properties": {  
889        "BusNo": 37  
890      }  
891    },  
892    {  
893      "id": 20829,  
894      "label": "Route",  
895      "type": "vertex",  
896      "properties": {  
897        "RouteNo": 37  
898      }  
899    },  
900    {  
901      "id": 20830,  
902      "label": "Ticket",  
903      "type": "vertex",  
904      "properties": {  
905        "TicketNo": 37  
906      }  
907    },  
908    {  
909      "id": 20831,  
910      "label": "Bus",  
911      "type": "vertex",  
912      "properties": {  
913        "BusNo": 38  
914      }  
915    },  
916    {  
917      "id": 20832,  
918      "label": "Route",  
919      "type": "vertex",  
920      "properties": {  
921        "RouteNo": 38  
922      }  
923    },  
924    {  
925      "id": 20833,  
926      "label": "Ticket",  
927      "type": "vertex",  
928      "properties": {  
929        "TicketNo": 38  
930      }  
931    },  
932    {  
933      "id": 20834,  
934      "label": "Bus",  
935      "type": "vertex",  
936      "properties": {  
937        "BusNo": 39  
938      }  
939    },  
940    {  
941      "id": 20835,  
942      "label": "Route",  
943      "type": "vertex",  
944      "properties": {  
945        "RouteNo": 39  
946      }  
947    },  
948    {  
949      "id": 20836,  
950      "label": "Ticket",  
951      "type": "vertex",  
952      "properties": {  
953        "TicketNo": 39  
954      }  
955    },  
956    {  
957      "id": 20837,  
958      "label": "Bus",  
959      "type": "vertex",  
960      "properties": {  
961        "BusNo": 40  
962      }  
963    },  
964    {  
965      "id": 20838,  
966      "label": "Route",  
967      "type": "vertex",  
968      "properties": {  
969        "RouteNo": 40  
970      }  
971    },  
972    {  
973      "id": 20839,  
974      "label": "Ticket",  
975      "type": "vertex",  
976      "properties": {  
977        "TicketNo": 40  
978      }  
979    },  
980    {  
981      "id": 20840,  
982      "label": "Bus",  
983      "type": "vertex",  
984      "properties": {  
985        "BusNo": 41  
986      }  
987    },  
988    {  
989      "id": 20841,  
990      "label": "Route",  
991      "type": "vertex",  
992      "properties": {  
993        "RouteNo": 41  
994      }  
995    },  
996    {  
997      "id": 20842,  
998      "label": "Ticket",  
999      "type": "vertex",  
1000     "properties": {  
1001       "TicketNo": 41  
1002     }  
1003   }  
1004 ]  
1005 }  
1006 }
```

Filter:

Vertices: 3



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

IBM Graph-ct > g

Tutorials [Query](#)

Upload

Samples

Resources

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

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

```
1  [  
2    {  
3      "labels": [  
4        [],  
5        [],  
6        [],  
7        [],  
8        []  
9      ],  
10     "objects": [  
11       {  
12         "id": 20720,  
13         "label": "Bus",  
14         "type": "vertex",  
15         "properties": {  
16           "BusNo": 1  
17         }  
18       },  
19       {  
20         "id": 20721,  
21         "label": "Route",  
22         "type": "vertex",  
23         "properties": {  
24           "RouteNo": 1  
25         }  
26       },  
27       {  
28         "id": 20722,  
29         "label": "Ticket",  
30         "type": "vertex",  
31         "properties": {  
32           "TicketNo": 1  
33         }  
34       },  
35       {  
36         "id": 20723,  
37         "label": "Bus",  
38         "type": "vertex",  
39         "properties": {  
40           "BusNo": 2
```

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

IBM Graph

Documentation Bluemix Dashboard

IBM Graph-ct > g

Tutorials Que

Upload

Samples

Resources

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

Graph: g

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

```
1179 [
1180   "id": 122888368,
1181   "label": "Bus",
1182   "type": "vertex",
1183   "properties": {
1184     "BusNo": [
1185       {
1186         "id": "215tzq.215x9s-35x",
1187         "value": 83
1188       }
1189     ],
1190     "BusCapacity": [
1191       {
1192         "id": "215udy.215x9s-3yd",
1193         "value": 80
1194       }
1195     ]
1196   }
1197 ]
```

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

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

Graph: g

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

```
1 [
2  7
3  ]
```

4.6 Display travel details for Passenger, Sachin.

```
1 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
```

```
78 {  
79   "id": 12352,  
80   "label": "Route",  
81   "type": "vertex",  
82   "properties": {  
83     "Destination": [  
84       {  
85         "id": "5xk-9j4-6bp",  
86         "value": "San Francisco"  
87       }  
88     ],  
89     "RouteNo": [  
90       {  
91         "id": "554-9j4-4qt",  
92         "value": 101  
93       }  
94     ],  
95   }
```

Filter: ☐ Label ☒ Type ☐ Properties



Vertices: 4

Graph: g

4.7 Display details of the ticket which Sachin bought.

```
1 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
```



```
1  ▾ [  
2  ▾ {  
3    "Ticket_Id": [  
4      8  
5    ],  
6    "Ticket_Price": [  
7      18  
8    ]  
9  }  
10 ]
```

Filter:

Vertices: 0

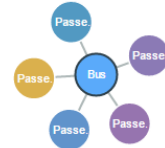
4.8 Show all passengers boarding Bus No. 23

```
1 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
```

```
9 {
10   "id": 20720,
11   "label": "Bus",
12   "type": "vertex",
13   "properties": {
14     "BusNo": [
15       {
16         "id": "8bi-fzk-35x",
17         "value": 23
18       }
19     ],
20     "BusCapacity": [
21       {
22         "id": "8pq-fzk-3yd",
23         "value": 80
24       }
25     ]
26   }
27 }
```



Vertices: 6