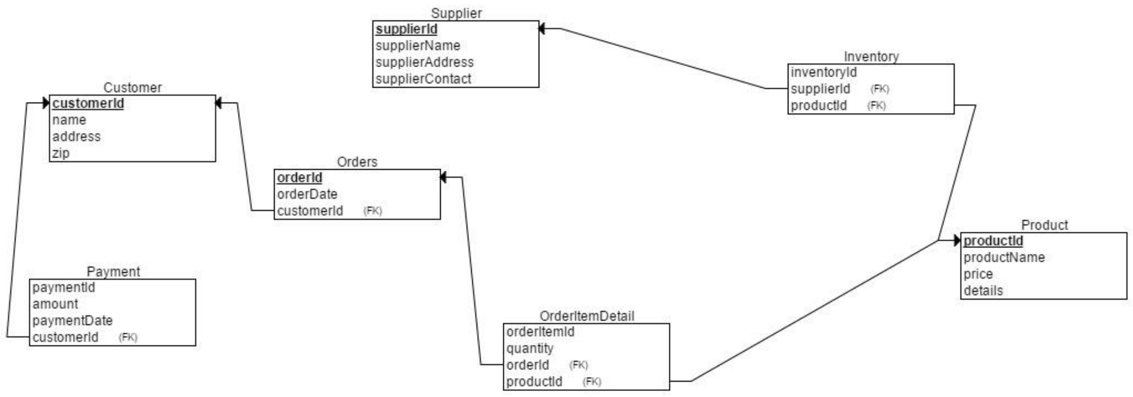
# SQLite DB ASSIGNMENT

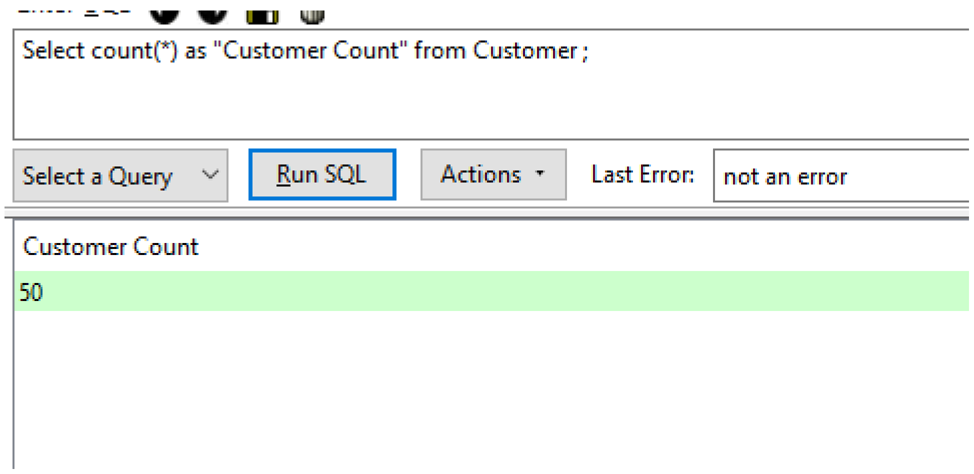
1. Installed SQLite Add-ons successfully for Firefox
2. Designed a database for Purchase Order Management System.
3. Created a schema with necessary tables from previous step.



4. Inserted sample data successfully.
5. Tried running different queries learnt in this chapter

5.1 Query to find the count of Customers

Select count(*) as "Customer Count" from Customer ;

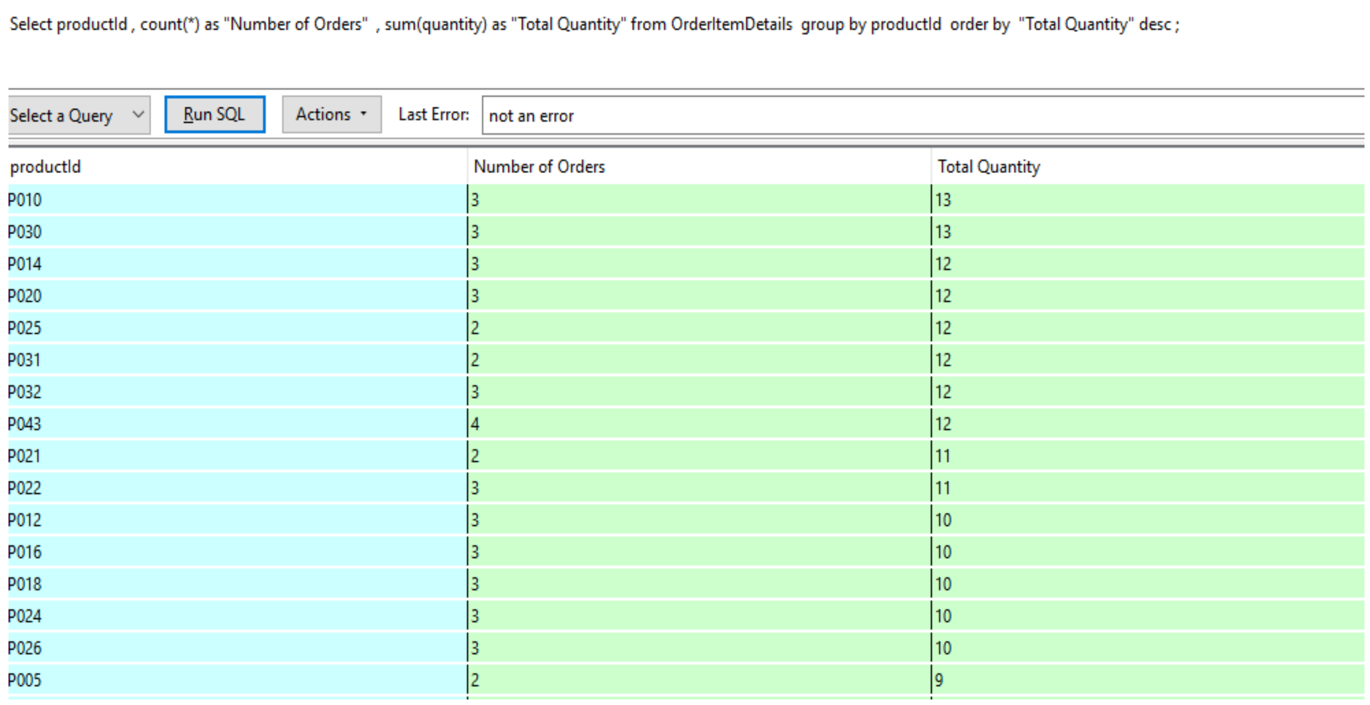


```
Select count(*) as "Customer Count" from Customer ;
```

```
Customer Count
50
```

5.2 Query to list maximum product sold

Select productId , count(*) as "Number of Orders" , sum(quantity) as "Total Quantity" from OrderItemDetails group by productId order by "Total Quantity" desc ;



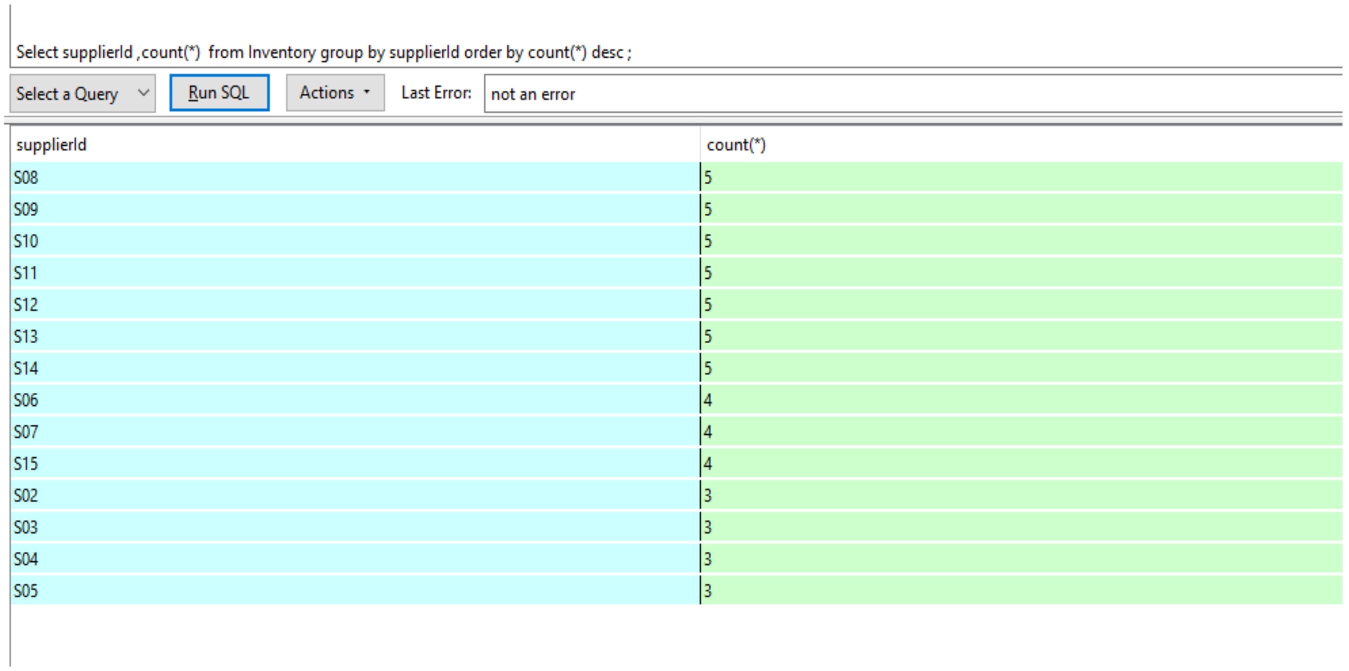
```
Select productId , count(*) as "Number of Orders" , sum(quantity) as "Total Quantity" from OrderItemDetails group by productId order by "Total Quantity" desc ;
```

Select a Query | Run SQL | Actions | Last Error: not an error

| productId | Number of Orders | Total Quantity |
|---|---|---|
| P010 | 3 | 13 |
| P030 | 3 | 13 |
| P014 | 3 | 12 |
| P020 | 3 | 12 |
| P025 | 2 | 12 |
| P031 | 2 | 12 |
| P032 | 3 | 12 |
| P043 | 4 | 12 |
| P021 | 2 | 11 |
| P022 | 3 | 11 |
| P012 | 3 | 10 |
| P016 | 3 | 10 |
| P018 | 3 | 10 |
| P024 | 3 | 10 |
| P026 | 3 | 10 |
| P005 | 2 | 9 |

5.3 Query to Count of Product Supplied by each Supplier

Select supplierId ,count(*)  from Inventory group by supplierId order by count(*) desc ;



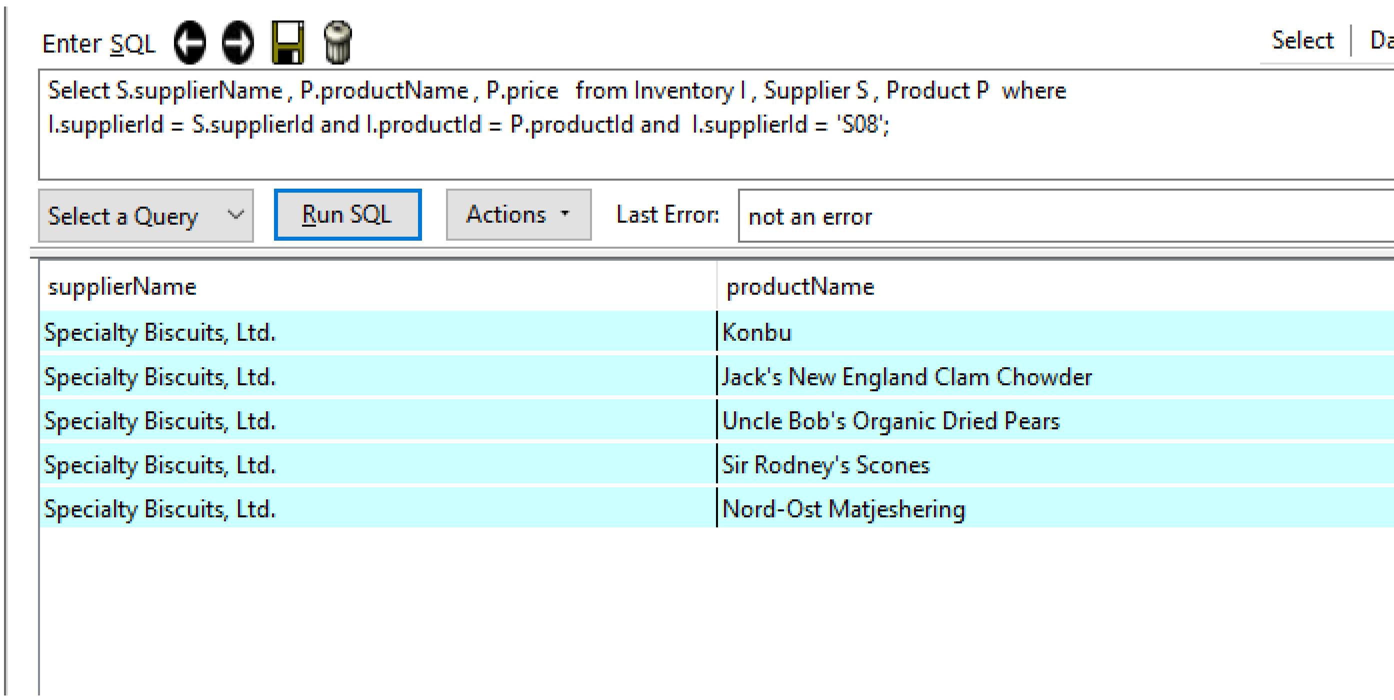
```
Select supplierId ,count(*)  from Inventory group by supplierId order by count(*) desc ;
```

Select a Query | Run SQL | Actions | Last Error: not an error

| supplierId | count(*) |
|---|---|
| S08 | 5 |
| S09 | 5 |
| S10 | 5 |
| S11 | 5 |
| S12 | 5 |
| S13 | 5 |
| S14 | 5 |
| S06 | 4 |
| S07 | 4 |
| S15 | 4 |
| S02 | 3 |
| S03 | 3 |
| S04 | 3 |
| S05 | 3 |

5.4 Query to find Supliername ,Product details sold by supplies S08

Select S.supplierName , P.productName , P.price  from Inventory I , Supplier S , Product P  where I.supplierId = S.supplierId and I.productId = P.productId and  I.supplierId = 'S08';



```
Select S.supplierName , P.productName , P.price   from Inventory I , Supplier S , Product P  where
I.supplierId = S.supplierId and I.productId = P.productId and  I.supplierId = 'S08';
```

Last Error: not an error

| supplierName | productName |
|---|---|
| Specialty Biscuits, Ltd. | Konbu |
| Specialty Biscuits, Ltd. | Jack's New England Clam Chowder |
| Specialty Biscuits, Ltd. | Uncle Bob's Organic Dried Pears |
| Specialty Biscuits, Ltd. | Sir Rodney's Scones |
| Specialty Biscuits, Ltd. | Nord-Ost Matjeshering |

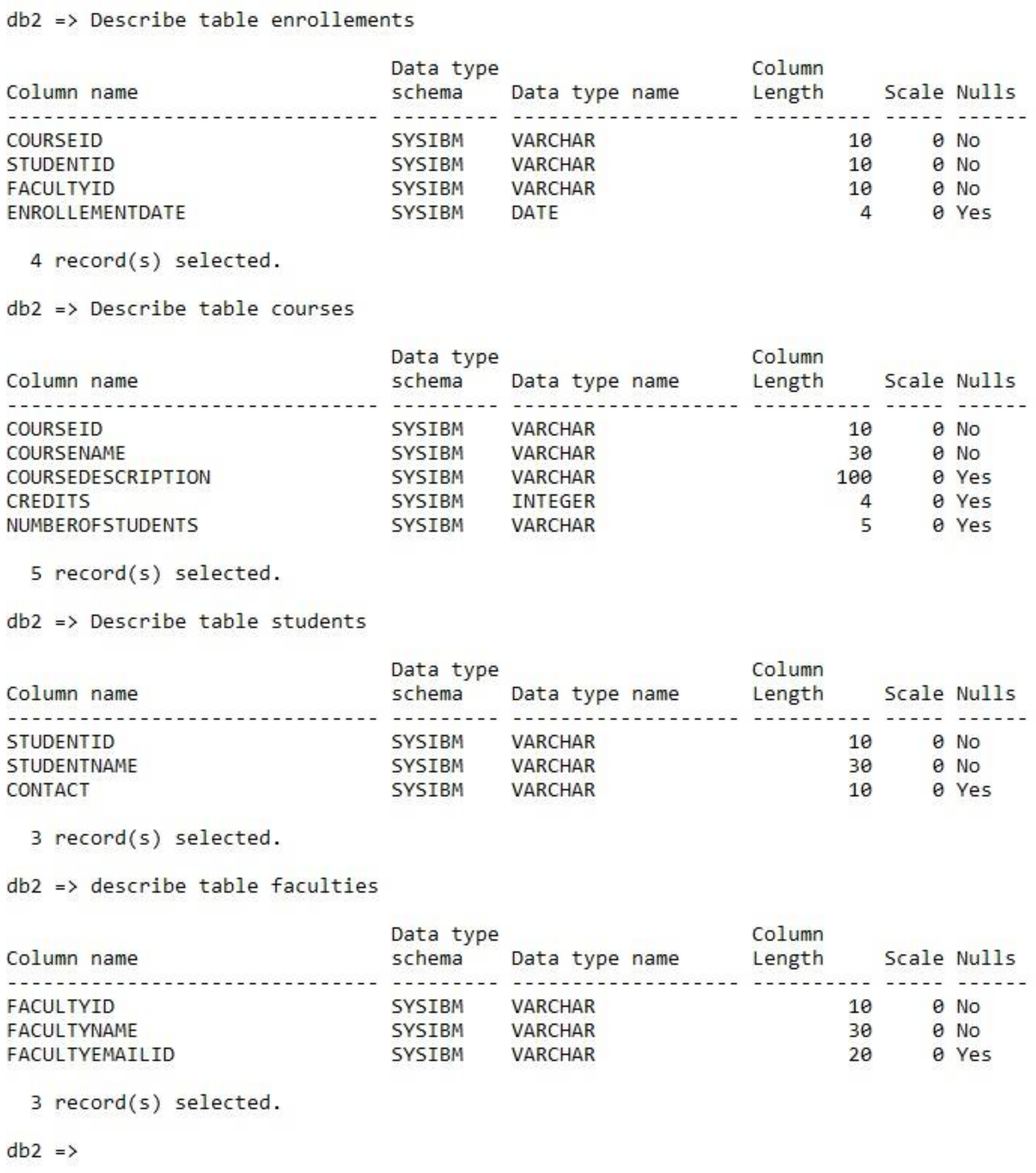
# DB2 EXPRESS C ASSIGNMENT

1. We created database name SAMPLE with force parameter (using: db2sampl command)

   Command used: write the following command in db2 command window

   ```
   db2sampl -force
   ```

2. We created a SAMLE database with schema name Department which contains tables named Courses, Faculties, Students and Enrollements. Below are the structures for each table in Department schema:



```
db2 => Describe table enrollements

                                Data type                     Column
Column name                     schema    Data type name      Length     Scale Nulls
------------------------------- --------- ------------------- ---------- ----- ------
COURSEID                        SYSIBM    VARCHAR                     10     0 No
STUDENTID                       SYSIBM    VARCHAR                     10     0 No
FACULTYID                       SYSIBM    VARCHAR                     10     0 No
ENROLLEMENTDATE                 SYSIBM    DATE                         4     0 Yes

  4 record(s) selected.

db2 => Describe table courses

                                Data type                     Column
Column name                     schema    Data type name      Length     Scale Nulls
------------------------------- --------- ------------------- ---------- ----- ------
COURSEID                        SYSIBM    VARCHAR                     10     0 No
COURSENAME                      SYSIBM    VARCHAR                     30     0 No
COURSEDESCRIPTION               SYSIBM    VARCHAR                    100     0 Yes
CREDITS                         SYSIBM    INTEGER                      4     0 Yes
NUMBEROFSTUDENTS                SYSIBM    VARCHAR                      5     0 Yes

  5 record(s) selected.

db2 => Describe table students

                                Data type                     Column
Column name                     schema    Data type name      Length     Scale Nulls
------------------------------- --------- ------------------- ---------- ----- ------
STUDENTID                       SYSIBM    VARCHAR                     10     0 No
STUDENTNAME                     SYSIBM    VARCHAR                     30     0 No
CONTACT                         SYSIBM    VARCHAR                     10     0 Yes

  3 record(s) selected.

db2 => describe table faculties

                                Data type                     Column
Column name                     schema    Data type name      Length     Scale Nulls
------------------------------- --------- ------------------- ---------- ----- ------
FACULTYID                       SYSIBM    VARCHAR                     10     0 No
FACULTYNAME                     SYSIBM    VARCHAR                     30     0 No
FACULTYEMAILID                  SYSIBM    VARCHAR                     20     0 Yes

  3 record(s) selected.

db2 =>
```

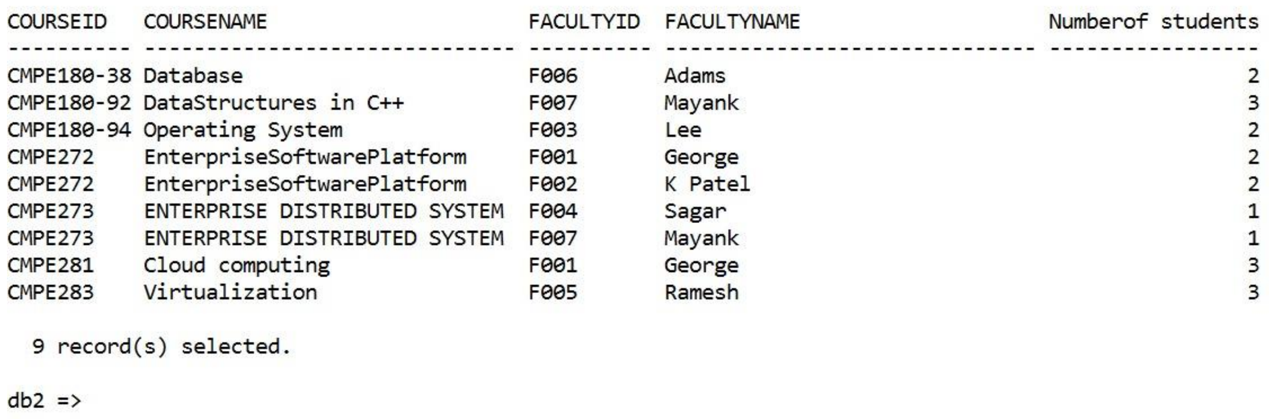
3. We ran two queries (use where clause and Group by).Blow are the snapshots of sample queries we ran:

   QUERY#1

   Query to print the courseid ,coursename,facultyid,facultyname grouped by the courseid and in ascending order of there courseid :

   select Enrollements.courseID,Courses.CourseName,Enrollements.facultyID,Faculties.facultyName, COUNT(Courses.CourseID) "Numberof students" From enrollements JOIN courses ON Enrollements.courseID = courses.courseID JOIN Faculties ON Faculties.FacultyID = Enrollements.FacultyID GROUP BY Enrollements.CourseID,Courses.CourseName,Enrollements.FacultyID,Faculties.FacultyName ORDER BY Enrollements.CourseID;

   OUTPUT:



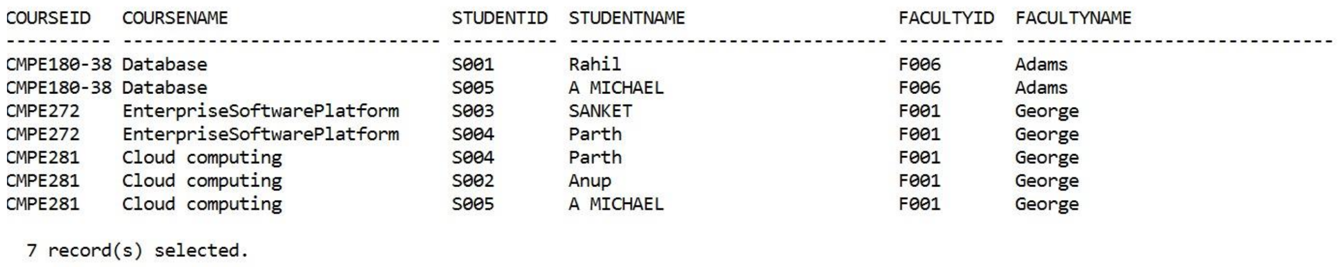
```
COURSEID   COURSENAME                     FACULTYID  FACULTYNAME                    Numberof students
---------- ------------------------------ ---------- ------------------------------ -----------------
CMPE180-38 Database                       F006       Adams                                          2
CMPE180-92 DataStructures in C++          F007       Mayank                                         3
CMPE180-94 Operating System               F003       Lee                                            2
CMPE272    EnterpriseSoftwarePlatform     F001       George                                         2
CMPE272    EnterpriseSoftwarePlatform     F002       K Patel                                        2
CMPE273    ENTERPRISE DISTRIBUTED SYSTEM  F004       Sagar                                          1
CMPE273    ENTERPRISE DISTRIBUTED SYSTEM  F007       Mayank                                         1
CMPE281    Cloud computing                F001       George                                         3
CMPE283    Virtualization                 F005       Ramesh                                         3

  9 record(s) selected.

db2 =>
```

   QUERY#2

   Query to print coursedid,CourseName,studentID,studentName,facultyID,facultyName for all the faculty with faculty as F001 or F006 in ascending order of there CourseID.

   SelectEnrollements.courseID,Courses.CourseName,Enrollements.studentID,Students.studentName,Enrollements.facultyID,Faculties.facultyName From enrollements JOIN courses ON Enrollements.courseID = courses.courseID JOIN students ON students.studentID = Enrollements.studentID JOIN Faculties ON Faculties.FacultyID = Enrollements.FacultyID where Enrollements.facultyID = 'F001' OR Enrollements.facultyID ='F006' ORDER BY Enrollements.CourseID;

   OUTPUT:



```
COURSEID   COURSENAME                     STUDENTID  STUDENTNAME                    FACULTYID  FACULTYNAME
---------- ------------------------------ ---------- ------------------------------ ---------- ------------------------------
CMPE180-38 Database                       S001       Rahil                          F006       Adams
CMPE180-38 Database                       S005       A MICHAEL                      F006       Adams
CMPE272    EnterpriseSoftwarePlatform     S003       SANKET                         F001       George
CMPE272    EnterpriseSoftwarePlatform     S004       Parth                          F001       George
CMPE281    Cloud computing                S004       Parth                          F001       George
CMPE281    Cloud computing                S002       Anup                           F001       George
CMPE281    Cloud computing                S005       A MICHAEL                      F001       George

  7 record(s) selected.
```

4. Generate query explain plan (use: db2exfmt tool)

   Command Used:
   db2 connect to SAMPLE(database name)
   db2 set schema department

```
db2 set current explain mode explain
db2 -tvf Query.sql (Query file with path if not in the current directory)
db2 set current explain mode no
db2exfmt -d SAMPLE -1 -o -t(for terminal output)
db2exfmt -d sample -1 -o C:\Users\R@HIL\db2exfmt_query_explain_output.txt
```

5. Content of the Explain Plan generated in the above text file

```
               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:      2017-02-27-21.41.31.054000
EXPLAIN_REQUESTER: R@HIL

Database Context:
----------------
        Parallelism:          None
        CPU Speed:            4.723442e-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:       2
        QUERYTAG:      CLP
        Statement Type:   Select
        Updatable:        No
        Deletable:        No
        Query Degree:     1

Original Statement:
------------------
select
  A.courseID,
  B.CourseName,
  A.facultyID,
  C.facultyName,
  A.studentID,
  D.studentName,
  A.enrollementDate
From
  enrollements A JOIN courses B
  ON A.courseID = B.courseID JOIN students D
  ON D.studentID = A.studentID JOIN Faculties C
  ON C.FacultyID = A.FacultyID
where
  (A.facultyID = 'F001' OR
   A.facultyID ='F006' OR
   A.facultyID = 'F003') AND
  (A.studentID = 'S001' OR
   A.studentID = 'S002' OR
   A.studentID = 'S004')
ORDER BY
  A.CourseID


Optimized Statement:
-------------------
SELECT
  Q6.COURSEID AS "COURSEID",
  Q5.COURSENAME AS "COURSENAME",
  Q6.FACULTYID AS "FACULTYID",
  Q8.FACULTYNAME AS "FACULTYNAME",
  Q6.STUDENTID AS "STUDENTID",
  Q7.STUDENTNAME AS "STUDENTNAME",
  Q6.ENROLLEMENTDATE AS "ENROLLEMENTDATE"
FROM
  DEPARTMENT.COURSES AS Q5,
  DEPARTMENT.ENROLLEMENTS AS Q6,
  DEPARTMENT.STUDENTS AS Q7,
  DEPARTMENT.FACULTIES AS Q8
WHERE
  (Q6.COURSEID = Q5.COURSEID) AND
  (Q7.STUDENTID = Q6.STUDENTID) AND
  (Q8.FACULTYID = Q6.FACULTYID) AND
  Q6.STUDENTID IN ('S001', 'S002', 'S004') AND
  Q6.FACULTYID IN ('F001', 'F006', 'F003')
ORDER BY
  Q6.COURSEID

Access Plan:
-----------
        Total Cost:             27.3662
        Query Degree:           1


                                         Rows
                                        RETURN
                                        (   1)
                                         Cost
                                          I/O
                                          |
                                        3.31579
                                        TBSCAN
                                        (   2)
                                        27.3662
                                           4
                                          |
                                        3.31579
                                        SORT
                                        (   3)
                                        27.3651
                                           4
                                          |
                                        3.31579
                                        ^HSJOIN
                                        (   4)
                                        27.3625
                                           4
                               /----------+----------\
                           3.31579                      3
                           ^HSJOIN                   TBSCAN
                           (   5)                    (  10)
                           20.525                    6.83667
                              3                         1
                   /---------+---------\               |
               3.31579                    3             7
               HSJOIN^                 TBSCAN   TABLE: DEPARTMENT
               (   6)                  (   9)       STUDENTS
               13.6876                 6.83667         Q7
                  2                       1
          /------+-------\               |
         8               3.31579          7
      TBSCAN             TBSCAN   TABLE: DEPARTMENT
      (   7)             (   8)       FACULTIES
      6.83622            6.84985         Q8
         1                  1
        |                  |
         8                 19
 TABLE: DEPARTMENT  TABLE: DEPARTMENT
      COURSES         ENROLLEMENTS
        Q5                 Q6


Operator Symbols :
------------------

   Symbol      Description
   ---------   ------------------------------------------
   >JOIN     : Left outer join
    JOIN<    : Right outer join
   >JOIN<    : Full outer join
   xJOIN     : Left antijoin
    JOINx    : Right antijoin
   ^JOIN     : Left early out
    JOIN^    : Right early out


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

Diagnostic Identifier:         1
Diagnostic Details:    EXP0256I  Analysis of the query shows that the
                       query might execute faster if an additional index
                       was created to enable zigzag join. Schema name:
                       "DEPARTMENT". Table name: "ENROLLEMENTS". Column
                       list: "(STUDENTID, FACULTYID, COURSEID)".

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


        1) RETURN: (Return Result)
```

```
                Cumulative Total Cost:                27.3662
                Cumulative CPU Cost:          309521
                Cumulative I/O Cost:          4
                Cumulative Re-Total Cost:     27.3636
                Cumulative Re-CPU Cost:       303956
                Cumulative Re-I/O Cost:       0
                Cumulative First Row Cost:    27.3658
                Estimated Bufferpool Buffers:  0

                Arguments:
                ---------
                BLDLEVEL: (Build level)
                        DB2 v11.1.1010.160 : s1612051900
                HEAPUSE : (Maximum Statement Heap Usage)
                        208 Pages
                PLANID  : (Access plan identifier)
                        44939729a14955e7
                PREPTIME: (Statement prepare time)
                               218 milliseconds
                SEMEVID : (Semantic environment identifier)
                        4465607c101ab241
                STMTHEAP: (Statement heap size)
                        8192
                STMTID  : (Normalized statement identifier)
                        8105e04819c65e82

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

                                Estimated number of rows:      3.31579
                                Number of columns:             7
                                Subquery predicate ID:                 Not
Applicable

                                Column Names:
                                ------------
                                +Q9.COURSEID(A)+Q9.ENROLLEMENTDATE
                                +Q9.STUDENTNAME+Q9.STUDENTID+Q9.FACULTYNAME
                                +Q9.FACULTYID+Q9.COURSENAME


        2) TBSCAN: (Table Scan)
                Cumulative Total Cost:                27.3662
                Cumulative CPU Cost:          309521
                Cumulative I/O Cost:          4
                Cumulative Re-Total Cost:     27.3636
                Cumulative Re-CPU Cost:       303956
                Cumulative Re-I/O Cost:       0
                Cumulative First Row Cost:    27.3658
                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:
                -------------
                        12) From Operator #3

                                Estimated number of rows:       3.31579
                                Number of columns:              7
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q6.COURSEID(A)+Q5.COURSENAME
                                +Q6.ENROLLEMENTDATE+Q6.FACULTYID+Q6.STUDENTID
                                +Q7.STUDENTNAME+Q8.FACULTYNAME


                Output Streams:
                --------------
                        13) To Operator #1

                                Estimated number of rows:       3.31579
                                Number of columns:              7
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q9.COURSEID(A)+Q9.ENROLLEMENTDATE
                                +Q9.STUDENTNAME+Q9.STUDENTID+Q9.FACULTYNAME
                                +Q9.FACULTYID+Q9.COURSENAME


        3) SORT  : (Sort)
                Cumulative Total Cost:                  27.3651
                Cumulative CPU Cost:            307155
                Cumulative I/O Cost:            4
                Cumulative Re-Total Cost:       27.3625
                Cumulative Re-CPU Cost:         301590
                Cumulative Re-I/O Cost:         0
                Cumulative First Row Cost:      27.3651
                Estimated Bufferpool Buffers:   1

                Arguments:
                ---------
                DUPLWARN: (Duplicates Warning flag)
                        FALSE
                KEYS    : (Key cardinality)
                        4
                NUMROWS : (Estimated number of rows)
                        4
                ROWWIDTH: (Estimated width of rows)
                        80.000000
                SORTKEY : (Sort Key column)
                        1: Q6.COURSEID(A)
                TEMPSIZE: (Temporary Table Page Size)
                        8192
                UNIQUE  : (Uniqueness required flag)
```

```
                        FALSE

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

                                Estimated number of rows:       3.31579
                                Number of columns:              8
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q5.COURSENAME+Q6.ENROLLEMENTDATE+Q6.FACULTYID
                                +Q6.STUDENTID+Q6.COURSEID+Q7.STUDENTNAME
                                +Q7.STUDENTID+Q8.FACULTYNAME


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

                                Estimated number of rows:       3.31579
                                Number of columns:              7
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q6.COURSEID(A)+Q5.COURSENAME
                                +Q6.ENROLLEMENTDATE+Q6.FACULTYID+Q6.STUDENTID
                                +Q7.STUDENTNAME+Q8.FACULTYNAME


        4) HSJOIN: (Hash Join)
                Cumulative Total Cost:                  27.3625
                Cumulative CPU Cost:            301590
                Cumulative I/O Cost:            4
                Cumulative Re-Total Cost:       27.3625
                Cumulative Re-CPU Cost:         301590
                Cumulative Re-I/O Cost:         4
                Cumulative First Row Cost:      27.3625
                Estimated Bufferpool Buffers:   1

                Arguments:
                ---------
                BITFLTR : (Hash Join Bit Filter used)
                        FALSE
                EARLYOUT: (Early Out flag)
                        LEFT
                HASHCODE: (Hash Code Size)
                        24 BIT
                HASHTBSZ: (Number of hash table entries)
                        3
                TEMPSIZE: (Temporary Table Page Size)
                        8192
                TUPBLKSZ: (Tuple Block Size (bytes))
                        4000

                Predicates:
                ----------
                7) Predicate used in Join,
                        Comparison Operator:            Equal (=)
                        Subquery Input Required:      No
                        Filter Factor:                0.333333

                        Predicate Text:
                        --------------
                        (Q7.STUDENTID = Q6.STUDENTID)



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

                                Estimated number of rows:      3.31579
                                Number of columns:             7
                                Subquery predicate ID:                 Not
Applicable

                                Column Names:
                                ------------
                                +Q5.COURSENAME+Q6.ENROLLEMENTDATE+Q6.FACULTYID
                                +Q6.STUDENTID+Q6.COURSEID+Q8.FACULTYNAME
                                +Q8.FACULTYID

                        10) From Operator #10

                                Estimated number of rows:      3
                                Number of columns:             2
                                Subquery predicate ID:                 Not
Applicable

                                Column Names:
                                ------------
                                +Q7.STUDENTNAME+Q7.STUDENTID


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

                                Estimated number of rows:      3.31579
                                Number of columns:             8
                                Subquery predicate ID:                 Not
Applicable

                                Column Names:
                                ------------
                                +Q5.COURSENAME+Q6.ENROLLEMENTDATE+Q6.FACULTYID
                                +Q6.STUDENTID+Q6.COURSEID+Q7.STUDENTNAME
                                +Q7.STUDENTID+Q8.FACULTYNAME


        5) HSJOIN: (Hash Join)
                Cumulative Total Cost:                20.525
                Cumulative CPU Cost:           232945
                Cumulative I/O Cost:           3
                Cumulative Re-Total Cost:      20.525
                Cumulative Re-CPU Cost:        232945
                Cumulative Re-I/O Cost:        3
                Cumulative First Row Cost:     20.525
                Estimated Bufferpool Buffers:  1

                Arguments:
```

```
                ---------
                BITFLTR : (Hash Join Bit Filter used)
                        FALSE
                EARLYOUT: (Early Out flag)
                        LEFT
                HASHCODE: (Hash Code Size)
                        24 BIT
                HASHTBSZ: (Number of hash table entries)
                        3
                JN INPUT: (Join input leg)
                        OUTER
                TEMPSIZE: (Temporary Table Page Size)
                        8192
                TUPBLKSZ: (Tuple Block Size (bytes))
                        4000

                Predicates:
                ----------
                8) Predicate used in Join,
                        Comparison Operator:            Equal (=)
                        Subquery Input Required:        No
                        Filter Factor:                  0.333333

                        Predicate Text:
                        --------------
                        (Q8.FACULTYID = Q6.FACULTYID)



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

                                Estimated number of rows:       3.31579
                                Number of columns:              6
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q5.COURSENAME+Q5.COURSEID+Q6.ENROLLEMENTDATE
                                +Q6.FACULTYID+Q6.STUDENTID+Q6.COURSEID

                        7) From Operator #9

                                Estimated number of rows:       3
                                Number of columns:              2
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q8.FACULTYNAME+Q8.FACULTYID


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

                                Estimated number of rows:       3.31579
                                Number of columns:              7
                                Subquery predicate ID:                  Not
Applicable
```

```
                        Column Names:
                        ------------
                        +Q5.COURSENAME+Q6.ENROLLEMENTDATE+Q6.FACULTYID
                        +Q6.STUDENTID+Q6.COURSEID+Q8.FACULTYNAME
                        +Q8.FACULTYID


        6) HSJOIN: (Hash Join)
                Cumulative Total Cost:                13.6876
                Cumulative CPU Cost:          164303
                Cumulative I/O Cost:          2
                Cumulative Re-Total Cost:     13.6876
                Cumulative Re-CPU Cost:       164303
                Cumulative Re-I/O Cost:       2
                Cumulative First Row Cost:    13.6876
                Estimated Bufferpool Buffers: 1

                Arguments:
                ---------
                BITFLTR : (Hash Join Bit Filter used)
                        FALSE
                EARLYOUT: (Early Out flag)
                        RIGHT
                HASHCODE: (Hash Code Size)
                        24 BIT
                HASHTBSZ: (Number of hash table entries)
                        3
                JN INPUT: (Join input leg)
                        OUTER
                TEMPSIZE: (Temporary Table Page Size)
                        8192
                TUPBLKSZ: (Tuple Block Size (bytes))
                        4000

                Predicates:
                ----------
                6) Predicate used in Join,
                        Comparison Operator:          Equal (=)
                        Subquery Input Required:      No
                        Filter Factor:                0.125

                        Predicate Text:
                        --------------
                        (Q6.COURSEID = Q5.COURSEID)



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

                                Estimated number of rows:     8
                                Number of columns:            2
                                Subquery predicate ID:                Not
Applicable

                                Column Names:
                                ------------
                                +Q5.COURSENAME+Q5.COURSEID

                        4) From Operator #8
```

```
                                Estimated number of rows:      3.31579
                                Number of columns:             4
                                Subquery predicate ID:                 Not
Applicable

                                Column Names:
                                ------------
                                +Q6.ENROLLEMENTDATE+Q6.FACULTYID+Q6.STUDENTID
                                +Q6.COURSEID


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

                                Estimated number of rows:      3.31579
                                Number of columns:             6
                                Subquery predicate ID:                 Not
Applicable

                                Column Names:
                                ------------
                                +Q5.COURSENAME+Q5.COURSEID+Q6.ENROLLEMENTDATE
                                +Q6.FACULTYID+Q6.STUDENTID+Q6.COURSEID


        7) TBSCAN: (Table Scan)
                Cumulative Total Cost:                 6.83622
                Cumulative CPU Cost:           66091
                Cumulative I/O Cost:           1
                Cumulative Re-Total Cost:      0.00732039
                Cumulative Re-CPU Cost:        15498
                Cumulative Re-I/O Cost:        0
                Cumulative First Row Cost:     6.82971
                Estimated Bufferpool Buffers:  1

                Arguments:
                ---------
                CUR_COMM: (Currently Committed)
                        TRUE
                JN INPUT: (Join input leg)
                        OUTER
                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
                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

                Input Streams:
                -------------
                        1) From Object DEPARTMENT.COURSES

                                Estimated number of rows:      8
                                Number of columns:             3
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q5.$RID$+Q5.COURSENAME+Q5.COURSEID


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

                                Estimated number of rows:      8
                                Number of columns:             2
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q5.COURSENAME+Q5.COURSEID


        8) TBSCAN: (Table Scan)
                Cumulative Total Cost:                 6.84985
                Cumulative CPU Cost:           94952
                Cumulative I/O Cost:           1
                Cumulative Re-Total Cost:      0.0209527
                Cumulative Re-CPU Cost:        44359
                Cumulative Re-I/O Cost:        0
                Cumulative First Row Cost:     6.83491
                Estimated Bufferpool Buffers:  1

                Arguments:
                ---------
                CUR_COMM: (Currently Committed)
                        TRUE
                JN INPUT: (Join input leg)
                        INNER
                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
                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:
                ----------
                11) Sargable Predicate,
                        Comparison Operator:            In List (IN), evaluated by
binary search (list sorted at compile-time)
                        Subquery Input Required:        No
                        Filter Factor:                  0.368421

                        Predicate Text:
                        --------------
                        Q6.STUDENTID IN ('S001', 'S002', 'S004')


                12) Sargable Predicate,
                        Comparison Operator:            In List (IN), evaluated by
binary search (list sorted at compile-time)
                        Subquery Input Required:        No
                        Filter Factor:                  0.473684

                        Predicate Text:
                        --------------
                        Q6.FACULTYID IN ('F001', 'F006', 'F003')



                Input Streams:
                -------------
                        3) From Object DEPARTMENT.ENROLLEMENTS

                                Estimated number of rows:       19
                                Number of columns:              5
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q6.$RID$+Q6.ENROLLEMENTDATE+Q6.FACULTYID
                                +Q6.STUDENTID+Q6.COURSEID


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

                                Estimated number of rows:       3.31579
                                Number of columns:              4
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q6.ENROLLEMENTDATE+Q6.FACULTYID+Q6.STUDENTID
                                +Q6.COURSEID
```

```
	9) TBSCAN: (Table Scan)
		Cumulative Total Cost: 		6.83667
		Cumulative CPU Cost: 		67039.5
		Cumulative I/O Cost: 		1
		Cumulative Re-Total Cost: 	0.00776841
		Cumulative Re-CPU Cost: 	16446.5
		Cumulative Re-I/O Cost: 	0
		Cumulative First Row Cost: 	6.83118
		Estimated Bufferpool Buffers: 	1

		Arguments:
		---------
		CUR_COMM: (Currently Committed)
			TRUE
		JN INPUT: (Join input leg)
			INNER
		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
		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:
		----------
		9) Sargable Predicate,
			Comparison Operator: 		In List (IN), evaluated by
binary search (list sorted at compile-time)
			Subquery Input Required: 	No
			Filter Factor: 			0.428571

			Predicate Text:
			--------------
			Q8.FACULTYID IN ('F001', 'F006', 'F003')



		Input Streams:
		-------------
			6) From Object DEPARTMENT.FACULTIES

				Estimated number of rows: 	7
				Number of columns: 		3
```

```
                              Subquery predicate ID:               Not
Applicable

                              Column Names:
                              ------------
                              +Q8.$RID$+Q8.FACULTYNAME+Q8.FACULTYID


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

                              Estimated number of rows:      3
                              Number of columns:             2
                              Subquery predicate ID:               Not
Applicable

                              Column Names:
                              ------------
                              +Q8.FACULTYNAME+Q8.FACULTYID


        10) TBSCAN: (Table Scan)
                Cumulative Total Cost:                6.83667
                Cumulative CPU Cost:            67039.5
                Cumulative I/O Cost:            1
                Cumulative Re-Total Cost:       0.00776841
                Cumulative Re-CPU Cost:         16446.5
                Cumulative Re-I/O Cost:         0
                Cumulative First Row Cost:      6.83118
                Estimated Bufferpool Buffers:   1

                Arguments:
                ---------
                CUR_COMM: (Currently Committed)
                        TRUE
                JN INPUT: (Join input leg)
                        INNER
                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
                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:
                ----------
                10) Sargable Predicate,
                        Comparison Operator:            In List (IN), evaluated by
binary search (list sorted at compile-time)
                        Subquery Input Required:        No
                        Filter Factor:                  0.428571

                        Predicate Text:
                        --------------
                        Q7.STUDENTID IN ('S001', 'S002', 'S004')



                Input Streams:
                -------------
                        9) From Object DEPARTMENT.STUDENTS

                                Estimated number of rows:       7
                                Number of columns:              3
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q7.$RID$+Q7.STUDENTNAME+Q7.STUDENTID


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

                                Estimated number of rows:       3
                                Number of columns:              2
                                Subquery predicate ID:                  Not
Applicable

                                Column Names:
                                ------------
                                +Q7.STUDENTNAME+Q7.STUDENTID


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

        Schema: DEPARTMENT
        Name:   COURSES
        Type:   Table
                        Time of creation:               2017-02-20-00.07.05.895001
                        Last statistics update:         2017-02-20-11.28.10.379000
                        Number of columns:              5
                        Number of rows:                 8
                        Width of rows:                  66
                        Number of buffer pool pages:    1
                        Number of data partitions:      1
                        Distinct row values:            No
                        Tablespace name:                IBMDB2SAMPLEREL
                        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

     Schema: DEPARTMENT
     Name:   ENROLLEMENTS
     Type:   Table
                    Time of creation:              2017-02-20-12.21.04.415001
                    Last statistics update:        2017-02-20-21.53.12.213000
                    Number of columns:             4
                    Number of rows:                19
                    Width of rows:                 39
                    Number of buffer pool pages:   1
                    Number of data partitions:     1
                    Distinct row values:           No
                    Tablespace name:               IBMDB2SAMPLEREL
                    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

     Schema: DEPARTMENT
     Name:   FACULTIES
     Type:   Table
                    Time of creation:              2017-02-20-11.27.45.674000
                    Last statistics update:        2017-02-20-21.22.47.596000
                    Number of columns:             3
                    Number of rows:                7
                    Width of rows:                 31
                    Number of buffer pool pages:   1
                    Number of data partitions:     1
                    Distinct row values:           No
                    Tablespace name:               IBMDB2SAMPLEREL
                    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

     Schema: DEPARTMENT
     Name:   STUDENTS
     Type:   Table
                    Time of creation:              2017-02-20-11.44.55.568000
                    Last statistics update:        2017-02-20-12.09.49.582000
                    Number of columns:             3
                    Number of rows:                7
                    Width of rows:                 32
                    Number of buffer pool pages:   1
                    Number of data partitions:     1
                    Distinct row values:           No
                    Tablespace name:               IBMDB2SAMPLEREL
```

```
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
```

# IBM GRAPH DB ASSIGNMENT

1.) Create an account on IBM Bluemix,navigate through “Data and analytics” section and go on Graph as a service Through the Graph Dashboard, capture the Service credentials from the left navigation Bar.

2.) We are now going to authenticate each of our requests made to our graph database by storing our credentials in temporary token and using it for authentication every time a request in made .

Commands used for Authentication:

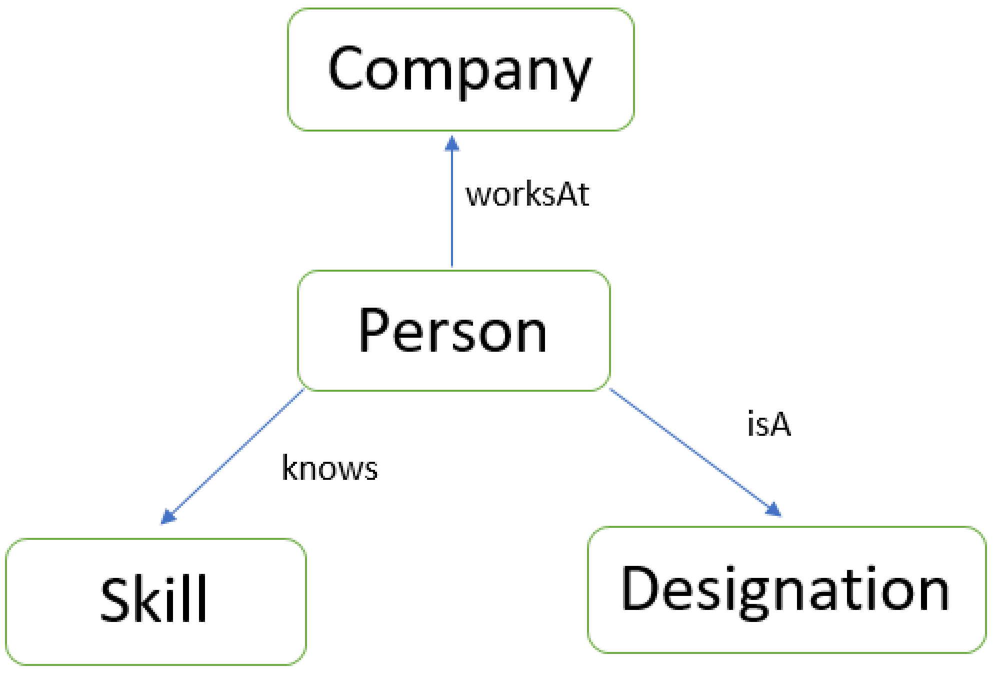
```
CREDS='
{
  "credentials": {
    "apiURL": "https://ibmgraph-alpha.ng.bluemix.net/a261eac3-6956-4185-a789-8e24c475e89b/g",
    "username": "575fd2b0-c011-4b94-82cf-43244b44f3a8",
    "password": "aca0d011-26f5-485d-a446-182750278a27"
  }
}
'
USER=$(echo $CREDS | jq -r '.credentials.username')
PASS=$(echo $CREDS | jq -r '.credentials.password')
URL=$(echo $CREDS | jq -r '.credentials.apiURL' | sed -E 's/(.*)\/.*/\1/' ) # remove the graph name from the apiURL
alias curl='curl --max-time 60 --connect-timeout 5 --silent --show-error' # set some defaults for curl
TOKEN=$(curl "${URL}/_session" -u "$USER:$PASS" | jq -r '.["gds-token"]')
echo "Your session token is $TOKEN"
```

3.) We created a graph with the name cmpe272gp23

Commands used

```
GRAPH="cmpe272gp23”
curl "$URL/_graphs/$GRAPH" \
   -X POST \
   -H "Authorization: gds-token $TOKEN" \
   -d '' | jq '.'
```

Now we are defining the schema for the graphdb cmpe272gp23
Below is the graph representation of cmpe272gp23



```
SCHEMA='
{
 "vertexLabels" : [
 {"name" : "Company"} ,
 {"name" : "Person"} ,
 {"name" : "Skill"} ,
 {"name" : "Position"}
         ],

 "edgeLabels": [
  {"name" : "worksAt" , "multiplicity":"MANY2ONE"},
  {"name": "knows" , "multiplicity" : "MULTI"},
  {"name" : "isA" ,"multiplicity" : "SIMPLE"}
         ],

 "propertyKeys" : [
  {"name" : "companyName","dataType":"String","cardinality":"SINGLE"} ,
  {"name" : "estbalished" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "firstName" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "lastName" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "age" , "dataType": "Integer" ,"cardinality" :"SINGLE"} ,
  {"name" : "salary" , "dataType": "Float" ,"cardinality" :"SINGLE"} ,
  {"name" : "address" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "gender" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "email" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "skillName" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "designation" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "joiningDate" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "employeeId" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "competencyLevel" , "dataType": "String" ,"cardinality" :"SINGLE"} ,
  {"name" : "since" , "dataType": "String" ,"cardinality" :"SINGLE"} ],

          "vertexIndexes" : [
          {"name":"userbyname" ,"propertyKeys" : ["firstName","lastName"] ,"composite":false , "unique":false } ,
            {"name":"userbyage" ,"propertyKeys" :["age"] ,"composite":false , "unique":false },
            {"name":"userbygender" ,"propertyKeys" :["gender"] ,"composite":false , "unique":false },
            {"name":"userbyskill" ,"propertyKeys" :["skillName"] ,"composite":false , "unique":false },
  {"name":"userbyemployeeid" ,"propertyKeys" :["employeeId"] ,"composite":true , "unique": true },
  {"name":"userbycompanyName" ,"propertyKeys" :["companyName"] ,"composite":false , "unique": false},
  {"name":"userbydesignation" ,"propertyKeys" :["designation"] ,"composite":false , "unique":false },
  {"name":"userbysalary" ,"propertyKeys" :["salary"] ,"composite":false , "unique":false }

  ],

   "edgeIndexes" : [
   {"name":"userbyjoiningDate" ,"propertyKeys" :["joiningDate"] ,"composite":false , "unique":false },
   {"name":"userbysince" ,"propertyKeys" :["since"] ,"composite":false , "unique":false },
   {"name":"userbycompetency" ,"propertyKeys" :["competencyLevel"] ,"composite":false , "unique":false}
   ]
}'


curl "$URL/$GRAPH/schema" \
    -X POST \
    -H "Authorization: gds-token $TOKEN" \
    -H 'Content-Type: application/json' \
    -d "$SCHEMA" | jq '.'
```

4.) We are now going to insert all the data into our schema.
  Commands used:

```
cat << ENDGREMLIN >gremlin.json # write everything until ENDGREMLIN into gremlin.json
{ "gremlin": "

def David =  graph.addVertex(T.label, 'Person', 'firstName', 'David',
'lastName','Morgan','age',35,'salary',70000.00,'address','San
Jose','gender','Male','email','xx@gmail.com','employeeId','G001');
def Vinayak =  graph.addVertex(T.label, 'Person', 'firstName', 'Vinayak',
'lastName','Patel','age',26,'salary',80000.00,'address','San
Fransisco','gender','Male','email','Vinayak@gmail.com','employeeId','G002');
def Akhilesh =  graph.addVertex(T.label, 'Person', 'firstName', 'Akhilesh',
'lastName','Doe','age',24,'salary',50000.00,'address','San
fernando','gender','Male','email','akhilesh@gmail.com','employeeid','G003');
def Rahil =  graph.addVertex(T.label, 'Person', 'firstName', 'Rahil',
'lastName','Modi','age',30,'salary',30000.00,'address','Santa
Cruiz','gender','Male','email','Rahil@gmail.com','employeeid','G005');
def Siddharth =  graph.addVertex(T.label, 'Person', 'firstName', 'Siddharth',
'lastName','Mewada','age',24,'salary',20000.00,'address','Santa
Cruiz','gender','Male','email','sid@gmail.com','employeeid','G006');

def Google = graph.addVertex(T.label, 'Company', 'companyName', 'Google', 'established', '1995');
def Apple = graph.addVertex(T.label, 'Company', 'companyName', 'Apple', 'established', '1997');

def Manager = graph.addVertex(T.label, 'Position', 'positionName','Project Manager');
def Director = graph.addVertex(T.label, 'Position', 'positionName','Director');
def Lead = graph.addVertex(T.label, 'Position', 'positionName','Technical lead');
def SystemsEngineer = graph.addVertex(T.label, 'Position', 'positionName','SystemsEngineer');

def Python =  graph.addVertex(T.label, 'Skill', 'skillName', 'Python');
def Java =  graph.addVertex(T.label, 'Skill', 'skillName', 'Java');
def Mainframe =  graph.addVertex(T.label, 'Skill', 'skillName', 'Mainframe');
def Javascript =  graph.addVertex(T.label, 'Skill', 'skillName', 'Javascript');
def Go  =  graph.addVertex(T.label, 'Skill', 'skillName', 'Go');

David.addEdge('knows',Python,'competencyLevel','good');
Vinayak.addEdge('knows',Java,'competencyLevel','poor');
Akhilesh.addEdge('knows',Mainframe,'competencyLevel','expert');
Rahil.addEdge('knows',Javascript,'competencyLevel','good');
Siddharth.addEdge('knows',Go,'competencyLevel','poor');
Siddharth.addEdge('knows',Python,'competencyLevel','expert');
Vinayak.addEdge('knows',Javascript,'competencyLevel','expert');

David.addEdge('worksAt' ,Google , 'joiningDate' , '12 Mar 2015' );
Vinayak.addEdge('worksAt' ,Google , 'joiningDate' , '29 Feb 2012' );
Siddharth.addEdge('worksAt' ,Google , 'joiningDate' , '31  Oct 2015' );
Rahil.addEdge('worksAt' ,Apple , 'joiningDate' , '20 June 2013' );
Akhilesh.addEdge('worksAt' ,Apple , 'joiningDate' , '01 Sep 2014' );

David.addEdge('isA',Manager,'since','Mar 2015');
Vinayak.addEdge('isA',SystemsEngineer,'since','Jun 2014');
Akhilesh.addEdge('isA',Director,'since','Dec 2014');
Rahil.addEdge('isA',SystemsEngineer,'since','Aug 2013');
Siddharth.addEdge('isA',Lead,'since','Nov 2015');


"
}
ENDGREMLIN

curl "$URL/$GRAPH/gremlin" \
    -X POST \
    -H "Authorization: gds-token $TOKEN" \
    -H 'Content-Type: application/json' \
    -d @gremlin.json | jq '.'
```

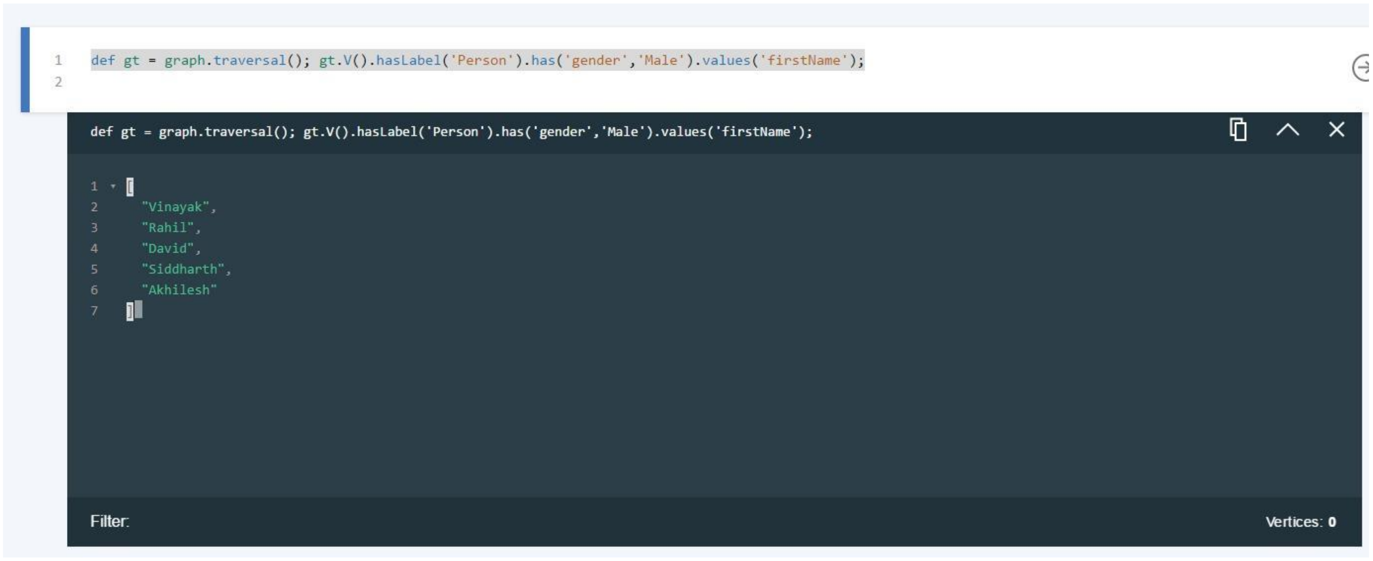
5.) Now we are going to use Gremlin queries to traverse through our graph database cmpe272gp23

QUERY#1

Query in GREMLIN to show all the person who has gender Male

```
def gt = graph.traversal();
gt.V().hasLabel('Person').has('gender','Male').values('firstName');
```

OUTPUT



```
def gt = graph.traversal(); gt.V().hasLabel('Person').has('gender','Male').values('firstName');
```

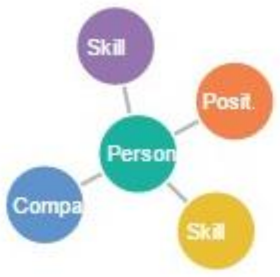
```
def gt = graph.traversal(); gt.V().hasLabel('Person').has('gender','Male').values('firstName');
```

```
[
  "Vinayak",
  "Rahil",
  "David",
  "Siddharth",
  "Akhilesh"
]
```

QUERY#2
Query in GREMLIN to show all the details of person with firstName : Siddharth

```
def gt = graph.traversal();
gt.V().hasLabel('Person').has('firstName','Siddharth').outE('knows','isA','worksAt').inV().path();
```

OUTPUT:



QUERY#3

Query in GREMLIN to find out all the details of people working at google with "competency level expert " of any languages .

```
  def gt = graph.traversal();
gt.V().hasLabel('Company').has('companyName','Google').inE('worksAt').outV().outE('knows').has('competencyLevel','
expert').inV().path();
```

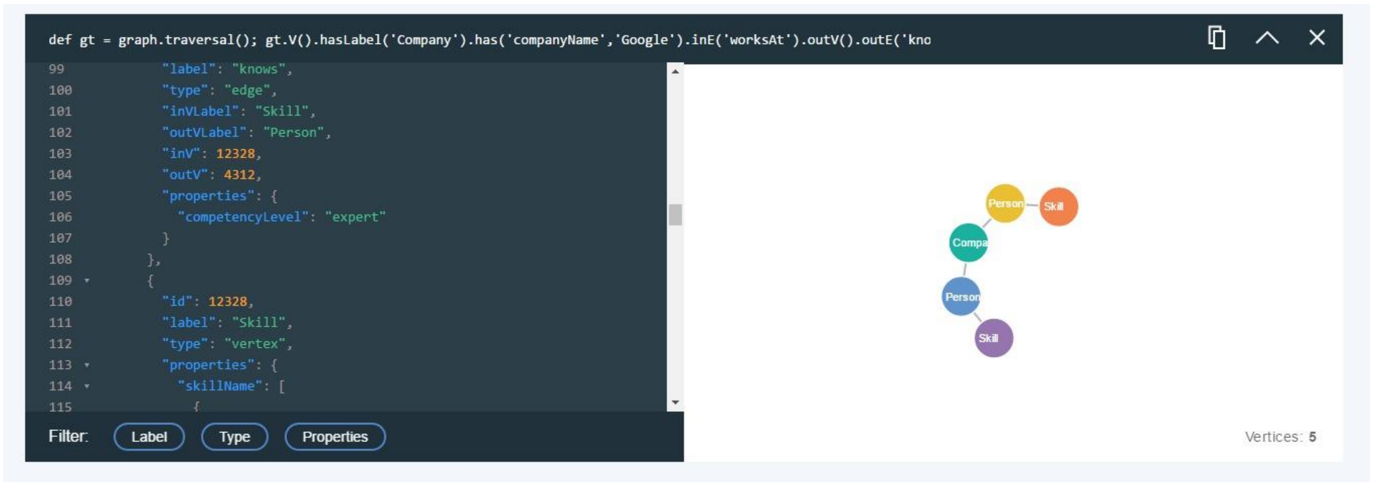
OUTPUT in jason :

```
[
 {
  "labels": [
   [],
   [],
  [],
  [],
  []
 ],
 "objects": [
  {
   "id": 4216,
   "label": "Company",
   "type": "vertex",
   "properties": {
    "established": [
     {
      "id": "1lb-394-2a6d",
      "value": "1995"
     }
    ],
    "companyName": [
     {
      "id": "173-394-sl",
      "value": "Google"
     }
    ]
   }
  },
  {
   "id": "e8b-388-fth-394",
   "label": "worksAt",
   "type": "edge",
   "inVLabel": "Company",
   "outVLabel": "Person",
   "inV": 4216,
   "outV": 4184,
   "properties": {
    "joiningDate": "29 Feb 2012"
   }
  },
  {
   "id": 4184,
   "label": "Person",
   "type": "vertex",
   "properties": {
    "firstName": [
     {
      "id": "16z-388-2dh",
      "value": "Vinayak"
     }
    ],
    "lastName": [
     {
      "id": "1l7-388-35x",
      "value": "Patel"
     }
    ],
    "address": [
     {
      "id": "2rv-388-5j9",
        "value": "San Fransisco"
      }
    ],
    "gender": [
      {
        "id": "363-388-6bp",
        "value": "Male"
      }
    ],
    "employeeId": [
      {
        "id": "3yj-388-a9x",
        "value": "G002"
      }
    ],
    "salary": [
      {
        "id": "2dn-388-4qt",
        "value": 80000
      }
    ],
    "age": [
      {
        "id": "1zf-388-3yd",
        "value": 26
      }
    ],
    "email": [
      {
        "id": "3kb-388-745",
        "value": "Vinayak@gmail.com"
      }
    ]
  }
},
{
  "id": "du3-388-glx-36g",
  "label": "knows",
  "type": "edge",
  "inVLabel": "Skill",
  "outVLabel": "Person",
  "inV": 4120,
  "outV": 4184,
  "properties": {
    "competencyLevel": "expert"
  }
},
{
  "id": 4120,
  "label": "Skill",
  "type": "vertex",
  "properties": {
    "skillName": [
      {
        "id": "16r-36g-7wl",
        "value": "Javascript"
```

```
            }
          ]
        }
      }
    ]
  },
  {
    "labels": [
      [],
      [],
      [],
      [],
      []
    ],
    "objects": [
      {
        "id": 4216,
        "label": "Company",
        "type": "vertex",
        "properties": {
          "established": [
            {
              "id": "1lb-394-2a6d",
              "value": "1995"
            }
          ],
          "companyName": [
            {
              "id": "173-394-sl",
              "value": "Google"
            }
          ]
        }
      },
      {
        "id": "emj-9js-fth-394",
        "label": "worksAt",
        "type": "edge",
        "inVLabel": "Company",
        "outVLabel": "Person",
        "inV": 4216,
        "outV": 12376,
        "properties": {
          "joiningDate": "31  Oct 2015"
        }
      },
      {
        "id": 12376,
        "label": "Person",
        "type": "vertex",
        "properties": {
          "firstName": [
            {
              "id": "93f-9js-2dh",
              "value": "Siddharth"
            }
    ],
    "lastName": [
     {
      "id": "9hn-9js-35x",
      "value": "Mewada"
     }
    ],
    "Email": [
     {
      "id": "bgr-9js-28lh",
      "value": "sid@gmail.com"
     }
    ],
    "address": [
     {
      "id": "aob-9js-5j9",
      "value": "Santa Cruiz"
     }
    ],
    "Gender": [
     {
      "id": "b2j-9js-27t1",
      "value": "Male"
     }
    ],
    "Employeeid": [
     {
      "id": "buz-9js-29dx",
      "value": "G006"
     }
    ],
    "salary": [
     {
      "id": "aa3-9js-4qt",
      "value": 20000
     }
    ],
    "age": [
     {
      "id": "9vv-9js-3yd",
      "value": 24
     }
    ]
  }
 },
 {
  "id": "dfv-9js-glx-cnc",
  "label": "knows",
  "type": "edge",
  "inVLabel": "Skill",
  "outVLabel": "Person",
  "inV": 16392,
  "outV": 12376,
  "properties": {
   "competencyLevel": "expert"
  }
```

```
      },
      {
        "id": 16392,
        "label": "Skill",
        "type": "vertex",
        "properties": {
          "skillName": [
            {
              "id": "7i9-cnc-7wl",
              "value": "Python"
            }
          ]
        }
      }
    ]
  }
]
```

OUTPUT:



```
def gt = graph.traversal(); gt.V().hasLabel('Company').has('companyName','Google').inE('worksAt').outV().outE('kno
```

```
          "label": "knows",
          "type": "edge",
          "inVLabel": "Skill",
          "outVLabel": "Person",
          "inV": 12328,
          "outV": 4312,
          "properties": {
            "competencyLevel": "expert"
          }
        },
        {
          "id": 12328,
          "label": "Skill",
          "type": "vertex",
          "properties": {
            "skillName": [
              {
```

REFRENCES :

1) www.w3schools.com